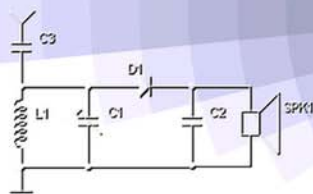


Компьютер *на* любительской радиостанции

- Виды цифровой радиосвязи
- Аппараты для цифровой радиосвязи
- Радилюбительские искусственные спутники Земли
- Программы для радилюбителей



+ Дискета



MACI EP

Геннадий Тяпичев

Компьютер *на* *любительской* *радиостанции*

Санкт-Петербург

«БХВ-Петербург»

2002

УДК 681.3.06
ББК 32.884.19
Т99

Тяпичев Г. А.

Т99 Компьютер на любительской радиостанции. — СПб.:
БХВ-Петербург, 2002. — 448 с.: ил.

ISBN 5-94157-172-0

В книге рассматриваются разнообразные аспекты применения компьютера на любительской радиостанции — для организации различных видов цифровой радиосвязи, ведения журнала радиосвязей, расчета орбит радиолюбительских искусственных спутников Земли (ИСЗ). Отдельная глава посвящена самостоятельному созданию простых в настройке и эффективных в работе конструкций модемов и других полезных для радиолюбителя аппаратов.

Для широкого круга радиолюбителей

УДК 681.3.06
ББК 32.884.19

Группа подготовки издания:

Главный редактор	<i>Екатерина Кондукова</i>
Зам. главного редактора	<i>Евгений Рыбаков</i>
Зав. редакцией	<i>Анна Кузьмина</i>
Редактор	<i>Дмитрий Локай</i>
Компьютерная верстка	<i>Нatalы Смирновой</i>
Корректор	<i>Нatalия Першакова</i>
Дизайн обложки	<i>Игоря Цырульников</i>
Зав. производством	<i>Николай Тверских</i>

Лицензия ИД № 02429 от 24.07.00. Подписано в печать 14.05.02.
Формат 70×100¹/₁₆. Печать офсетная. Усл. печ. л. 36,12.
Тираж 3000 экз. Заказ №
"БХВ-Петербург", 198005, Санкт-Петербург, Измайловский пр., 29.

Отпечатано с готовых диапозитивов
в Академической типографии "Наука" РАН
199034, Санкт-Петербург, 9 линия, 12.

ISBN 5-94157-172-0

© Тяпичев Г. А., 2002
© Оформление, издательство "БХВ-Петербург", 2002

Содержание

Предисловие.....	1
Глава 1. Радиолюбители и виды любительской радиосвязи	5
Как это начиналось	5
Кто такие радиолюбители-коротковолновики.....	10
Различные виды радиосвязи, используемые диапазоны	15
Телеграф.....	15
SSB-связь на одной боковой полосе	16
Телетайп и другие виды связи.....	16
Любительские диапазоны.....	17
Компьютер и цифровые виды радиосвязи.....	18
Компьютеры — как это все началось	18
Телеграф как цифровой вид связи.....	20
Телетайп.....	21
AMTOR	21
PACTOR.....	22
Packet Radio	22
Другие виды цифровой радиосвязи	22
Глава 2. Цифровые виды связи и программы для них	23
Предисловие	23
Телеграф как цифровой вид связи.....	24
Принципы построения телеграфного сигнала.....	24
Программа CW_QSO работает с кодом Морзе.....	30
Общие положения.....	32
Перечень команд.....	33
Подключение.....	35
Рабочий экран.....	35
Работа на передачу.....	36
Журнал	36
Проверка скоростей передачи	37
Как проводятся радиосвязи	37
Программа QSO_LOG как дополнение к CW_QSO	39
ПК становится "секретарем коротковолновика".....	40
Аппаратный журнал в других программах.....	43
Компьютер и телетайп.....	43
Как построен телетайпный сигнал	43
Как передать телетайпом символ	46
Программа RTTY_QSO — один из вариантов телетайпной программы	47
Общие положения.....	47
Конфигурационный файл	48
Перечень команд.....	49
Подключение.....	51

Рабочий экран	51
Работа на передачу	52
Журнал	52
Как выбрать лучшую из нескольких программ	54
Как выбрать программу для телетайпа	54
Программа RTT_BBS — еще один из примеров применения телетайпа	57
RTTY_BBS — электронная доска объявлений	57
Как работать с RTTY_BBS	58
Коротко о самой программе	61
И еще немного о телетайпе	63
AMTOR — модернизированный телетайп	64
Принципы построения сигналов в AMTOR	64
Кратко о рабочих режимах AMTOR	67
И еще немного о режиме ARQ	69
Программа Terman93 работает в режиме AMTOR	71
О программе Terman93	71
Как подключить программу	71
Конфигурационный файл	72
Наладка часов компьютера	75
Работа с программой	76
Как передавать файлы	77
Цифровой вид связи RACTOR	78
Принципы построения сигнала в RACTOR	78
Программа Terman93 работает в режиме RACTOR	81
О программе Terman93	81
Как подключить программу	81
Конфигурационный файл	82
Наладка часов компьютера	85
Работа с программой	86
Некоторые особенности RACTOR	87
Как передавать файлы	88
Как осваивать работу в RACTOR	88
Packet Radio — один из главных видов цифровой связи	89
Основные принципы построения сигнала в Packet Radio	89
Что такое "пакет"	90
Программа BayCom Terminal — одна из лучших для Packet Radio	94
Программа BayCom и другие	94
Коротко о программе BayCom Terminal	96
Команды программы BayCom Terminal	97
Конфигурационный файл BayCom Terminal	101
Всемирная любительская радиосеть	106
Основные принципы работы радиосети	106
Радиолубительская сеть Западной Европы	110
Перечень RACTOR BBS	112
Сеть AMPR	113
Программа F6FBB — самая распространенная из MBBS	116
О самой программе и как ее запустить	116
Основные команды MBBS	120
Краткий перечень команд программы F6FBB — MBBS	120
Описание команд BBS/Mailbox	122

FlexNet работает в сети Packet Radio	126
Основные принципы работы и состав FlexNet.....	126
Примеры применения FlexNet	127
Программа ВСМ в роли MBBS	130
Любительская сеть КАВМИНВОД	136
Лучшая региональная любительская сеть.....	136
Команды AMPR узла BayBOX.....	138
Новые виды цифровой связи.....	140
PSK31 — работа с клавиатурой	140
Программа PSK31SBW.....	145
Что такое PSK31SBW.....	145
Основные приемы работы.....	147
Передача сигналов PSK31	151
Передача текстовых файлов.....	153
Обобщение о PSK31	154
Немного о MT63	155
Любительское телевидение на коротких волнах.....	157
Связь через метеорные потоки.....	157
Заключение	158
Глава 3. Аппараты для цифровой связи.....	159
Предисловие	159
Модем для CW	160
Схема модема	161
Отдельные узлы модема	164
Тональный генератор.....	164
Полосовой фильтр	164
Специальный частотный детектор	165
Выходное согласующее устройство.....	167
Настройка модема.....	168
Модем универсальный	170
Схема модема	171
Отдельные узлы модема	173
Тональный генератор.....	173
Полосовой фильтр	174
Выходное согласующее устройство.....	176
Настройка модема.....	177
Другие варианты универсального модема	179
Модем MODEM21	180
Модем MODEM23	185
Прочие аппараты	185
Модем для частоты 215 кГц.....	185
Схема модема	186
Отдельные узлы модема	188
Настройка модема.....	193
Модем на 564ГГц	195
Схема модема	195
Отдельные узлы модема	195
Настройка модема.....	197
Преселектор.....	199

Модем и звуковая карта компьютера.....	203
Три технологии в цифровой радиосвязи.....	203
Что такое TNC	205
Технические характеристики	206
Состав контроллера.....	206
Команды TNC-2.....	207
Звуковая карта вместо модема.....	228
Программа RITTY и звуковая карта	231
Звуковая карта	232
FSK-, PTT-, и TTY-выходы.....	233
Рабочий экран	233
Использование RITTY.....	234
Обработка сигнала	236
Меню <i>SETUP</i>	239
Меню <i>RECEIVE</i>	240
Меню <i>TRANSMIT</i>	242
Цвета экрана.....	245
Примечания.....	245
Использование совместно с RTTY от WF1B	245
И еще о RITTY.....	246
Глава 4. Компьютер и ИСЗ	247
Предисловие	247
Первые спутники	247
Первый в мире ИСЗ	247
Работаем через спутники РС	249
Радиолюбительские ИСЗ	252
Спутники с аналоговой аппаратурой.....	255
Спутники с аппаратурой для цифровой связи	258
Спутники, которые уже не слышны.....	261
Расчет элементов орбит ИСЗ.....	265
Кеплеровские данные.....	266
Как расшифровать кеплеровские данные	269
Программа ORBITA.....	271
Как принимать информацию с RS	276
Глава 5. Как сделать программу	279
Предисловие	279
Как сделана программа ORBITA.....	280
Вспомогательная программа.....	290
Как сделана программа CW_QSO	300
Как сделана программа RTTY_QSO	359
Как сделана программа QSO_LOG	421
Как сделать программу для PSK31.....	436
Заключение	439
Приложение. Описание дискеты.....	441
Список литературы	442

*Раисе И. Тяпичевой, моей жене
и помощнице, посвящается эта книга.*

Автор

Предисловие

В настоящее время каждому человеку хочется иметь максимум информации. Интернет дает доступ к неограниченному источнику информации. Теоретически каждый может найти в нем все необходимое, но для начала, как минимум, нужно иметь доступ в Интернет.

Наша страна имеет уникальные размеры, а потому далеко не в каждом населенном пункте в ближайшие годы будет доступен Интернет. В то же время радиосвязь в деле обеспечения информацией отдаленных городов и сел может сыграть огромную роль. Применение компьютеров позволило создать такие виды радиосвязи, которые даже при скорости 300 Бод способны удовлетворить потребности многих людей в деле получения нужной информации. Так, например, в радиолюбительской сети на базе видов связи Packet Radio и PACTOR можно найти исчерпывающий материал по любой радиолюбительской тематике. Возможности такой сети особенно возрастают при использовании искусственных спутников Земли (ИСЗ).

Твердая уверенность в том, что только так называемые "цифровые виды связи", в которых главную роль играет компьютер, могут предоставить многим из нас доступ к необходимой информации, побудила меня взяться за написание этой книги. В ней я хочу передать заинтересованным людям накопленный за много лет опыт применения компьютера на любительской радиостанции. В течение многих лет я занимался на любительском уровне созданием программ и модемов для цифровых видов связи, разрабатывал программы для расчета элементов орбит ИСЗ и многими иными радиолюбительскими направлениями. Большинство из них приведены в этой книге.

Предлагаемая книга рассчитана на широкий круг читателей. Даже далекий от любительской радиосвязи человек сможет найти в ней интересные сведения о любительских ИСЗ, о работе компьютерных программ совместно с радиостанцией или просто уяснить для себя какие-то интересные особенности работы компьютера. Описания видов связи AMTOR, PACTOR и PSK31 ранее были опубликованы только в специальной литературе. Так что здесь впервые массовому читателю предоставляется возможность познакомиться с их замечательными возможностями.

Материал книги распределен по главам следующим образом.

□ **Глава 1. Радиолюбители и виды любительской радиосвязи.**

В этой главе, рассчитанной на массового читателя, рассказывается об открытии радиосвязи, описываются этапы развития радиосвязи, роль радиосвязи в нашей жизни. Далее рассказывается о радиолюбителях-коротковолновиках, любительских диапазонах и кратко об основных видах любительской радиосвязи, в том числе и о появлении цифровых видов связи, главная роль в которых принадлежит компьютеру.

□ **Глава 2. Цифровые виды связи и программы для них.**

Материал этой главы представляет интерес для радиолюбителя, который решает приобрести компьютер для своей радиостанции, а также для любознательного массового читателя, который хочет узнать что-то новое. В *главе 2* поочередно рассказывается об особенностях каждого из цифровых видов связи, о кодировании и декодировании символов в каждом из них, что должно заинтересовать и начинающих программистов, и студентов учебных заведений связи. Далее приводятся сведения о компьютерной программе, предлагаемой для использования. Большинство описанных в книге программ разработано мною, вся документация к программам выполнена на русском языке. К программам зарубежных авторов мною выполнен перевод необходимой документации и пояснительных текстов на русский язык.

Хочу отметить особенности программы проведения радиосвязей посредством кода Морзе. Кроме основной функции проведения телеграфных радиосвязей, она может использоваться также как тренажер для изучения приема на слух и передачи на ручном ключе сигналов кода Морзе, при этом имеется возможность точного расчета скорости передачи. Или быть просто интересной игрошкой для самого юного радиолюбителя.

□ **Глава 3. Аппараты для цифровой связи.**

Материал этой главы должен заинтересовать радиолюбителя-конструктора. Здесь приведены описания конструкций и принципиальные электрические схемы модемов, доступных для изготовления в домашних условиях. Хорошая работа этих аппаратов подтверждается их многолетней эксплуатацией. Кроме того, в разделе *"Прочие аппараты"* приведены описания и схемы оригинальных экспериментальных аппаратов, предназначенных для повышения эффективности приема сигналов на радиостанции.

□ **Глава 4. Компьютер и ИСЗ.**

В этой главе приведен перечень ВСЕХ любительских ИСЗ, функционировавших в период написания книги, а также и не функционирующих в нормальном режиме, но находящихся на орбите. Приведены все доступные данные по каждому из спутников, даны ссылки на страницы Интернета, где можно найти дополнительную информацию по данному спутнику. Описаны методы использования ИСЗ для связи в любительской сети.

□ Глава 5. Как сделать программу.

Материал предназначен для любителей программировать. Здесь размещены исходные коды основных компонентов связанных компьютерных программ, используя которые программист-любитель может сам добавить нужные ему функции и подпрограммы и создать практически свою собственную программу. Например, в исходных кодах отрывка из программы ORBITA приведен полностью математический аппарат, необходимый для расчета элементов орбит любительского ИСЗ. Файл с исходными кодами из программы CW_QSO позволит разработать свою собственную программу для приема и передачи сигналов кода Морзе, или тренажер для изучения сигналов кода Морзе.

К книге прилагается дискета с программами.

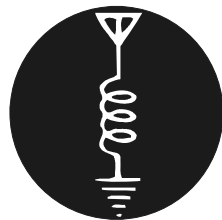
Желаю читателю успехов в расширении кругозора!

Геннадий А. Тяпичев — RA3XB

<mailto:ra3xb@kaluga.ru>

Глава 1

Радиолюбители и виды любительской радиосвязи



Как это начиналось

7 мая 1895 года русский инженер, преподаватель Минного офицерского класса в Санкт-Петербурге, Александр Степанович Попов, впервые в истории человечества на заседании физического отделения Русского физико-химического общества продемонстрировал похожее на телеграфный аппарат приемное устройство, которое, в отличие от обычного телеграфа, не имело никаких подводящих проводов. К всеобщему изумлению этот необычный аппарат без какой-либо видимой команды вдруг начал печатать телеграмму. Текст этой телеграммы состоял из двух слов: "Генрих Герц". Телеграмма была послана специальным передающим устройством, удаленным от аудитории, где проходило заседание ученого совета, на значительное расстояние. Это событие было первой публичной демонстрацией радиосвязи, когда первый радиоприемник принял и напечатал первую в мире радиограмму, полученную от находящегося в удалении передатчика.

Немецкий ученый Генрих Герц открыл явление передачи энергии, возникавшей при искровом электрическом разряде, на значительное расстояние. Но, как пишут историки, Герц считал, что открытое им явление не найдет какого-либо полезного применения.

Новый вид связи не нуждался в длинных линиях из проводов и быстро завоевал популярность. Над Европой витал призрак предстоящей войны, и государства не жалели средств для такого удобного, с военной точки зрения, вида связи. Всюду на планете возникали коммерческие и военные радиостанции, появлялись сотни энтузиастов этого вида связи среди интеллигенции. Постепенно из этих энтузиастов стали организовываться различные кружки и общества, члены которых стали называть себя "любителями радио". Первая официальная радиолубительская организация была создана в 1914 году в США. Она существует и до настоящего времени, являясь самой многочисленной радиолубительской организацией в мире. Организация носит название ARRL — Американская Радиолубительская Лига. На рис. 1.1 изображен официальный значок члена Лиги, взятый мною из архива радиолубительских рисунков.



Рис. 1.1. Значок члена ARRL

Военные и государственные радиоспециалисты того времени решили, что для целей военной, государственной и коммерческой радиосвязи пригодны только радиоволны длиной более 200 метров. Радиоволны более короткие были отданы, за ненадобностью, в распоряжение радиолюбителей. Как показало будущее развитие радиосвязи, именно короткие волны стали играть в радиосвязи главную роль.

В нашей стране важнейшую роль в деле развития радио сыграла организованная в начале 20-х годов прошлого столетия Нижегородская радиолaborатория, руководимая М. А. Бонч-Бруевичем. В этой лаборатории были разработаны теоретические основы радиосвязи, многие конструкции радиоаппаратов, электронные приборы, а также самые мощные для того времени радиолампы для передатчиков. Ученые этой лаборатории были инициаторами разрешения частным лицам иметь дома радиостанции ограниченной мощности. После принятия Правительством соответствующего постановления, именно сотрудники Нижегородской радиолaborатории стали первыми официальными радиолюбителями-коротковолновиками. Федор Лбов, один из сотрудников Нижегородской лаборатории, на своей любительской радиостанции установил рекордную для того времени радиосвязь на коротких волнах. Сигналы его радиостанции были зафиксированы на территории нынешнего Ирака, расстояние по прямой линии до станции приема составляло 3000 км.

На рис. 1.2, взятом из радиолюбительского архива, изображен радист, работающий на телеграфном ключе.

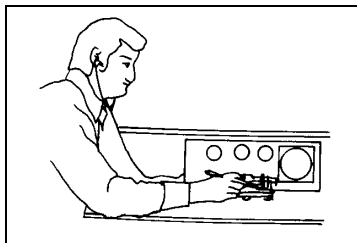


Рис. 1.2. Радист за работой

В те годы радиостанции уже умели передавать человеческую речь и музыку, но для радиолюбителей пока оставалась возможность вести передачу только телеграфным ключом, используя код Морзе. В нашей стране это была эпоха освоения Арктики, эпоха челюскинцев и дрейфующей на льдине метеорологической станции "Северный полюс". Во всех перечисленных событиях радиосвязь играла исключительно важную роль.

Радист Э. Т. Кренкель стал легендарным среди радиолюбителей нашей страны. Многие юноши тех лет строили детекторные приемники по описаниям более опытных любителей и часами вслушивались в эфир, ожидая начала работы станции этого радиста из далекой Арктики.

На рис. 1.3 изображена принципиальная электрическая схема простого детекторного радиоприемника, которым пользовались большинство радиолюбителей той поры.

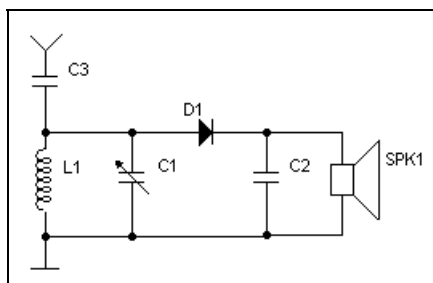


Рис. 1.3. Принципиальная схема детекторного приемника

Конференция радиолюбителей всего мира состоялась в Париже в 1925 году. На этой конференции было принято решение об организации Международного радиолюбительского союза — IARU. Эта организация продолжает работу и в настоящее время. Радиолюбительские организации всех стран считают своим долгом быть членами IARU.

Годы войны стали суровым испытанием для всех народов нашей страны. Радиолюбители быстрее других осваивали сложное армейское радиооборудование, в сложнейших условиях обеспечивали надежную связь, чем внесли большой вклад во всенародное дело Великой Победы.

Послевоенные годы, несмотря на сожженные города, деревни и разрушенное народное хозяйство, отличались огромным энтузиазмом наших людей. После напряженного трудового дня молодые люди шли на занятия в школы рабочей молодежи, в кружки различных оборонных обществ, где очень популярными были секции изучения радио. Издательством оборонных обществ печаталось много различной литературы по изготовлению радиоприемников, простых передатчиков для любительских радиостанций. Примерно

в 1949 году в газете для школьников "Пионерская правда" было напечатано объявление о том, что вышла в свет книга автора Борисова "Юный радиолюбитель" и сообщался адрес, по которому эту книгу можно было заказать. Для меня в ту пору это была самая замечательная книга, в которой описывались даже мельчайшие подробности изготовления не только простых приемников, но и таких радиодеталей, как конденсаторы и резисторы, купить которые в те годы было можно только в крупнейших городах. Трудно забыть волнение и восторг, когда в наушниках, подключенных к только что изготовленному из самодельных катушек, конденсаторов и детектора простому детекторному приемнику, раздались сигналы далекой радиостанции.

В те далекие годы в любительском эфире также господствовал телеграф. Для передачи сигнала от микрофона использовались или амплитудная, или частотная модуляция несущей частоты звуковыми сигналами.

На рис. 1.4 изображен телеграфный ключ несколько необычной для нас конструкции. Картинка взята из архива. Примерно о такой конструкции телеграфного ключа будет рассказано в *главе 2* этой книги.

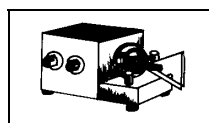


Рис. 1.4. Телеграфный ключ

Амплитудную модуляцию использовали все широкоэвещательные государственные и коммерческие радиостанции, а также некоторые радиолюбители. Для этого требовался сложный аппарат — модулятор. Наиболее сложным изделием при изготовлении модулятора был модуляционный трансформатор, от конструкции, аккуратности изготовления и способа экранирования которого зависело качество сигнала. Модулятор потреблял мощность, соизмеримую с потребляемой мощностью всей остальной части передатчика. Поэтому передатчики с амплитудной модуляцией были очень неэкономичны.

Частотная модуляция применялась для связи на ультракоротких волнах и в телевидении. Дело в том, что передатчик с частотной модуляцией занимал слишком широкую полосу частот. Для коротковолновых диапазонов, которые к тому времени были заполнены уже достаточно густо, это было недопустимо. В те же годы были сделаны и основные изобретения и разработки в области приема и передачи однополосных сигналов. В 30-е годы прошлого столетия в России уже было построено несколько магистральных линий дальней связи, работавших на принципе приема и передачи однополосных сигналов.

В радиолюбительскую практику прием и передача однополосных сигналов пришли в начале 60-х годов. Большую роль в формировании взглядов на

этот вид связи сыграла книга "Техника любительской однополосной радиосвязи" авторов С. Бунимовича и Л. Яйленко. Впервые книга была издана в 1964 году, переиздана — в 1970 году.

Для радиолюбителей освоение однополосной телефонии давало экономию электроэнергии, улучшение качества сигналов, уменьшение помех и значительное увеличение дальности радиосвязей. Однополосную модуляцию радиолюбители обозначают сочетанием букв — SSB.

Первая любительская радиостанция на все коротковолновые диапазоны, которая обеспечивала отличное качество сигнала с однополосной модуляцией при простоте изготовления и настройки, была разработана Г. Джунковским и Я. Лаповком. Радиостанция называлась "ДЛ-66" и была описана в журнале "Радио" № 5, 6 и 7 за 1967 год. Статья называлась "Радиостанция первой категории". Изготовленная мною в 1969 году по этому описанию радиостанция проработала у меня дома более десяти лет, при этом все корреспонденты отмечали хорошее качество сигнала.

Однако самой популярной среди радиолюбителей нашей страны и стран ближнего зарубежья в те годы была конструкция радиостанции, называемой среди радиолюбителей "Трансивер UW3DI".

Словом "трансивер" радиолюбители называют радиостанцию, в которой одни и те же электронные усилительные приборы (транзисторы или радиолампы) используются попеременно то в режиме передачи, то в режиме приема. В такой конструкции используется заметно меньшее число электронных приборов, уменьшаются габариты и вес. Сочетание букв UW3DI является позывным любительской радиостанции, принадлежащей радиолюбителю-коротковолновику Ю. Кудрявцеву, который описал конструкцию разработанной им радиостанции в журнале "Радио" № 5 за 1970 год. Статья называлась "Коротковолновый трансивер".

За последующие годы радиолюбителями были созданы многие новые конструкции, но работу радиостанции на "Трансивер UW3DI" часто можно слышать и в настоящее время.

Разработку самой современной любительской радиостанции сделал В. В. Дроздов. Радиостанция была описана и в журнале "Радио" и в брошюрах издательства "Радио и связь" в конце 1980-х годов. В любительских кругах эта радиостанция называется "Трансивер Дроздова".

Освоение любителями работы в режиме однополосной модуляции сигнала сыграло огромную роль в развитии любительской радиосвязи. Но в 1970-х годах ведущие фирмы — изготовители аппаратуры для радиосвязи, стали применять в своих конструкциях процессоры, которые до этого применялись только в электронных вычислительных машинах. В этот период стали появляться новые виды радиосвязи, которые могли передавать и принимать радиосигналы в автоматическом режиме и с огромными, немыслимыми до

тех времен, скоростями. В это же время у радиолюбителей стали появляться первые конструкции бытовых компьютеров. В нашей стране процесс появления бытовых компьютеров шел с опозданием, по сравнению со странами Запада, более чем на 10 лет. Первой доступной моделью простого бытового компьютера стал "Радио-86РК", подробный процесс изготовления которого был описан в нескольких номерах журнала "Радио" за 1986 год. В тот же период в нашей стране некоторые фирмы стали изготавливать аналогичные по характеристикам миникомпьютеры "Спектрум".

На рис. 1.5 изображен один из первых вариантов компьютера, который стал применяться в любительской радиосвязи. Рисунок взят из архива.

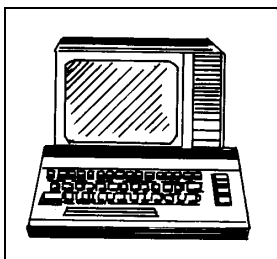


Рис. 1.5. Один из первых компьютеров

Так в нашей стране начался процесс компьютеризации любительской радиостанции, процесс развития "цифровых" видов связи. По моему мнению, этот процесс является новым поворотным пунктом в качественном развитии любительской радиосвязи.

Целью этой книги как раз и является показать читателю основные из известных на текущий день достоинств применения компьютера на любительской радиостанции и познакомить его с любительскими цифровыми видами связи, главную роль в которых выполняет компьютер. Это все будет описано в следующих главах книги. А пока продолжим знакомство с деятельностью радиолюбителей-коротковолновиков.

Кто такие радиолюбители-коротковолновики

Каждый из нас имеет какое-либо любимое занятие, которому он посвящает большую часть своего свободного времени.

Одни предпочитают каждую свободную минуту посидеть в кругу друзей за кружкой пива, другие бегут в гараж к своему ненаглядному четырехколесному идолу, третьи мчатся на собрание коллекционеров, четвертые готовятся к

покорению очередной горной вершины и т. д. Но есть и такие люди, которые всем прочим видам досуга предпочитают радиолулюбительскую деятельность. Понятие "радиолулюбитель" исключительно многогранно. Оно предполагает, что этот человек может часами сидеть за радиоприемником, чтобы отыскать в безбрежном эфире сигналы заветной радиостанции, что он может серьезно заняться изучением иностранного языка, чтобы свободно общаться с новыми друзьями из далекой страны, что он может взяться за изготовление какой-то очень необходимой ему конструкции электронного аппарата.

Радиолулюбительство позволяет приобрести ценные практические навыки и знания в области электро- и радиотехники, а также географии и иностранных языков. Такие знания могут оказаться весьма полезными и для специалиста, и для воина.

Коротковолновым радиолулюбительством занимаются сотни тысяч людей в мире. Самыми молодыми представителями армии коротковолнников являются школьники.

При многих школах и других учебных заведениях работают радиостанции коллективного пользования, так что возраст младших коротковолнников десять-двенадцать лет. Возраст многих старейших коротковолнников — больше восьмидесяти лет.

Короткими волнами увлекаются люди техники и искусства, военнотружущие и домашние хозяйки, школьники, ученые, рабочие.

В истории нашей страны яркой страницей является легендарный дрейф полярной станции "Северный полюс" с четверкой папанинцев. Радиот этой станции Эрнст Теодорович Кренкель, впоследствии Герой Советского Союза, доктор географических наук, был коротковолнником. Он же был радиотом ледокола "Челюскин", и его четкая работа на радиостанции сыграла важнейшую роль в быстрой организации спасения терпящих бедствие полярников.

Всему миру известен отважный путешественник — норвежский ученый Тур Хейердал. Трижды выходил он в открытый океан на утлых суденышках — плоте "Кон-Тики" и папирусной лодке "Ра". И всегда в составе его экипажа находились коротковолнники, успешно осуществлявшие связь со всем миром.

Отважные одиночки довольно часто отправляются в опасные морские путешествия, при этом любительская радиостанция позволяет им при необходимости связываться с наземными станциями. На рис. 1.6, взятом из архива, изображен яхтсмен, на парусе которого видны цифры "73", обозначающие на радиолулюбительском языке самые наилучшие пожелания.

Многие государственные деятели активно участвовали в радиолулюбительской деятельности. Бывший король Иордании Хусейн в молодости был очень активным коротковолнником. Известного сенатора из США Барри Голдуоте-

ра в 70-е годы прошлого столетия можно было часто слышать на телеграфных участках диапазонов.

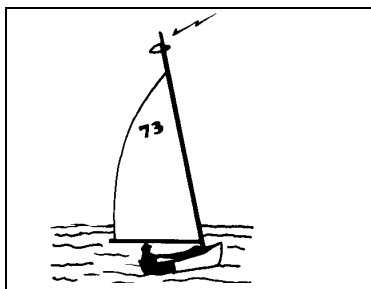


Рис. 1.6. Радиолучитель в море

Чем же привлекают короткие волны многочисленную армию столь не схожих друг с другом людей? Тем, что, как уже говорилось, коротковолновое радиолучительство многообразно, и каждый находит в нем что-то наиболее для себя привлекательное.

Вы любите мастерить, конструировать? Коротковолновики, как правило, сами конструируют свои передатчики, антенны, иногда и приемники, создавая подчас конструкции, не уступающие лучшим промышленным образцам.

Вы любите разрабатывать программы для компьютера — перед вами открываются безграничные возможности создавать новые виды цифровой связи или разрабатывать новые программы для старых видов. Вас с детства влекла романтика дальних путешествий, вы увлекались географией и зачитывались романами Жюль Верна? Коротковолновик может в течение часа "побывать" на всех континентах — в Африке и Америке, Азии и Антарктиде.

Вы страстный коллекционер, собираете марки, значки, открытки? Коротковолновики по традиции каждую проведенную связь подтверждают особыми карточками-квитанциями, которые высылают друг другу по почте. Эти карточки обычно бывают красочно оформлены, содержат фотографии, пейзажи, забавные рисунки. Коллекция таких карточек — это не рядовая коллекция обычных почтовых открыток! К этому можно прибавить еще коллекции значков, посвященных различным событиям в области коротковолнового любительства, и дипломов, выдаваемых коротковолновикам за различные достижения.

Вас привлекает изучение иностранных языков? Легче всего изучить язык, регулярно разговаривая на нем. Только коротковолновики имеют возможность, даже не выходя из дома, практиковаться в любом иностранном языке, разговаривая с собеседником, знающим этот язык в совершенстве.

Вам по душе спортивная борьба, соревнование? Существует много различных соревнований коротковолновиков — от местных до международных. Для того чтобы померяться мастерством со спортсменами других стран, коротковолновику даже не надо выезжать из своего города. А во многих соревнованиях он может выполнить нормативы спортивных разрядов или даже мастера спорта.

Привожу характеристики основных видов радиоспорта.

- ❑ **Спортивная телеграфия.** Это наиболее "камерный" вид спорта. Казалось бы, здесь все предельно просто — передавай и принимай телеграфные сигналы так быстро, как сможешь. Но эта простота кажущаяся: все легко и понятно лишь при небольших скоростях приема и передачи, до 100—110 знаков в минуту. Любой спортсмен, выступающий в видах, где требуется специальное оборудование, тщательно готовит его к соревнованиям. Спортсмены-телеграфисты часто изготавливают сами свои ключи, в этом им помогают и профессиональные конструкторы, и механики.
- ❑ **Спортивная радиопеленгация.** Этот вид радиоспорта родился в Голландии в первые годы после Второй мировой войны. Однажды поздним вечером на улицах Гааги появилась группа велосипедистов, на машинах которых были установлены странные аппараты с антеннами. На одном из перекрестков они по команде разъехались в разные стороны. Медленно двигаясь по пустынным улицам, они время от времени останавливались, сверялись с картой и прослушивали эфир, вращая специальные антенны. В головных телефонах раздавался обычный шум эфира, разноголосица станций, как вдруг послышался странный позывной: "Я — лиса, я — лиса". Это и был нужный велосипедисту позывной. Вращая антенну, велосипедист определил направление, откуда сигнал был слышен наиболее громко. Взяв пеленг, он отправился на поиски дома, где работал передатчик со странным позывным — "лиса". Так в 1947 году в Европе состоялись соревнования, которые вскоре завоевали громадную популярность среди радиолюбителей и получили название "охота на лис". Стали проводиться национальные первенства "охотников", а затем и международные. До 1954 года поиск "лис" велся на 80-метровом диапазоне, только на традиционных соревнованиях в Югославии в этом же году был применен УКВ-диапазон 144 МГц. В радиолубительской литературе все чаще стали публиковаться материалы о передатчиках, а главное, портативных приемниках для "охоты на лис". Здесь открылось широкое поле деятельности для конструкторов.

Соревнования по спортивной радиопеленгации в их современном виде включают поиск радиопередатчиков ("лис") на диапазонах 3,5 и 144 МГц и метание гранат. Накануне соревнований участники получают карты местности предстоящих забегов. После старта "лисы" (передатчики) начинают подавать сигналы: 30 секунд — сигнал, 30 секунд — пауза. Спорт-

смен должен найти передатчик в лесу, проставить отметку о прохождении "лисы" и искать следующую. Кто сделает это быстрее, тот и станет победителем. Спортсмену-"лисолову" нужно не только иметь отличную аппаратуру (приемники-пеленгаторы), но и быть хорошим кроссменом, ведь трассы имеют протяженность 9—10 км. Это при идеальном поиске, а в ходе соревнований спортсмены пробегают гораздо больше.

- **Многоборье радистов.** Уже само название этого вида спорта достаточно красноречиво. Спортсмен-радиомногоборец должен мастерски владеть ключом, работать на радиостанции с различными корреспондентами, метко стрелять и бросать гранаты, ориентироваться на местности с помощью карты и компаса, т. е. подготовка его должна быть комплексной и разносторонней.
- **Радиосвязь на коротких и ультракоротких волнах.** Один из самых любимых радиолюбителями видов спорта. Коротковолновик, находясь на своей радиостанции, за определенное время должен провести как можно больше радиосвязей с другими корреспондентами, участвующими в этих соревнованиях. При проведении связи участники должны обмениваться специальными контрольными номерами, состоящими, как правило, из нескольких цифр. За каждую проведенную радиосвязь начисляется определенное количество очков, при этом связь с более удаленным корреспондентом оценивается большим количеством очков. После окончания соревнований участник обязан выслать в судейскую коллегию отчет об участии в соревнованиях. В этом отчете должно быть указано время проведения каждой связи, позывной корреспондента, переданный корреспонденту и полученный от корреспондента контрольные номера. Судейская коллегия сверяет между собой отчеты всех участников на предмет совпадения всех записанных в отчете данных. Если какая-то запись не совпадает, то связь аннулируется у обоих участников. Судейская коллегия суммирует в отчете каждого участника очки за оставшиеся радиосвязи, победителем назначается участник, набравший наибольшую сумму очков за проведенные связи.

Участие в соревнованиях повышает мастерство радиолюбителя, закаляет волю и стимулирует совершенствование аппаратуры. Одним из любимейших соревнований по радиосвязи для многих радиолюбителей является "Полевой день". Это бывает каждый год в конце июля, когда группа радиолюбителей вместе с радиостанцией выезжает в полевые условия. Как правило, для расположения радиостанции выбирается самая большая из окрестных высот, чтобы обеспечить возможность проведения дальних связей. Связи проводятся на УКВ-диапазонах в течение суток непрерывно, поэтому соревнования командные.

На рис. 1.7, взятом из архива, изображена палатка и антенна УКВ-диапазона. Похоже, что соревнования уже закончились и радиолюбители отдыхают.

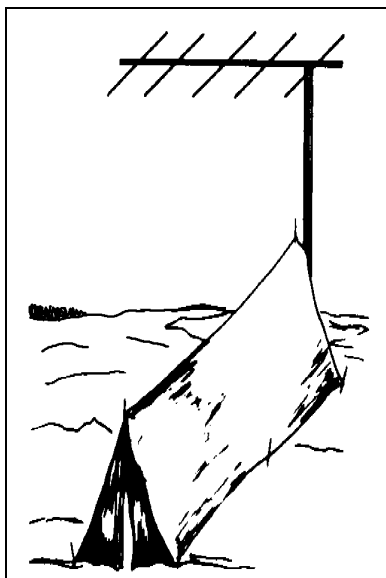


Рис. 1.7. Радиостанция на "Полевом дне"

Коротковолновик — это специалист, имеющий знания в области электротехники, радиотехники, радиосвязи. К тому же он и квалифицированный радист, способный вести радиосвязь даже в таких условиях, в которых отказываются работать профессиональные радисты, а в случае необходимости способный быстро найти и устранить неисправность в своей радиостанции.

Основой работы коротковолновика, сутью коротковолнового любительства, является установление двусторонней радиосвязи на коротких волнах.

Различные виды радиосвязи, используемые диапазоны

Телеграф

Как ни странно, но наиболее популярным видом радиосвязи до последнего времени был и пока остается телеграф — проведение связей при помощи кода Морзе. Самые интересные радиосвязи, связи с самыми неожиданными и удаленными корреспондентами проводятся, как правило, только телеграфом. Международная радиолубительская организация даже считает знание приема на слух и передачи на ручном ключе сигналов кода Морзе необходимым условием при переводе радиостанции, принадлежащей радиолубителю, в более высокую категорию. В нашей стране знания и опыт коротковолновика оцениваются по четырем категориям. К четвертой категории

относятся начинающие, для которых знание телеграфа необязательно. Далее идут третья, вторая и первая категории, для которых знание телеграфа обязательно.

При этом коротковолновик каждой из указанных категорий должен уметь принимать и передавать телеграфные сигналы не хуже установленной для данной категории нормы.

SSB-связь на одной боковой полосе

Большое число коротковолновиков предпочитает проводить радиосвязи при помощи микрофона, используя при этом SSB-радиостанции с однополосной модуляцией сигнала. Используя эти радиостанции, коротковолновики получают возможность непосредственного общения голосом с удаленным корреспондентом. Этим пользуются студенты институтов иностранных языков и простые граждане, желающие усовершенствовать свои знания иностранного языка.

На рисунке из старого архива изображен микрофон — самый необходимый аппарат при проведении SSB-радиосвязи.

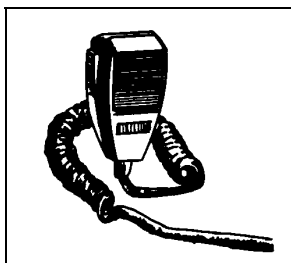


Рис. 1.8. Микрофон

Телетайп и другие виды связи

В последние годы на любительских диапазонах постоянно растет число радиостанций, работающих цифровыми видами связи, особенно телетайпом. Этот вид связи коротковолновики используют как при обычных связях, так и в соревнованиях. Более подробно о цифровых видах связи читайте в *главе 2*.

Для проведения любительских радиосвязей отведено довольно много диапазонов, как на коротких, так и на ультракоротких волнах. Приведу описания только некоторых, наиболее любимых радиолюбителями, коротковолновых диапазонов.

Любительские диапазоны

- **Диапазон 1,9 МГц.** Иначе этот диапазон называется "диапазон 160 метров". Дело в том, что диапазоны радиоволн могут иметь единицу измерения либо "метр", либо "Герц". Единица измерения частоты радиоволн "Герц" очень мала, поэтому в реальном радиоприемнике диапазон измеряется величинами "килоГерц" (сокращенно кГц) или "МегаГерц" (сокращенно МГц). Приставка "кило" обозначает тысячу, приставка "Мега" обозначает миллион, а слово "Герц" — это фамилия выдающегося ученого-физика, который открыл эффект распространения радиоволн.

Этот диапазон является чисто "ночным" диапазоном. Днем на этом диапазоне можно услышать только очень близко расположенные радиостанции, а ночью трудно найти свободный от работающих радиостанций участок диапазона. Здесь можно встретить и начинающего коротковолновика и известного "аса эфира". После полуночи число работающих радиостанций сокращается и на свободных участках диапазона можно услышать вдруг слабый сигнал радиостанции из другого континента. Работу на этом диапазоне разрешают начинающим коротковолновикам.

- **Диапазон 3,5 МГц** тоже относится к "ночным" диапазонам, но до полудня, а затем и с 15-ти часов можно связаться с корреспондентами, удаленными до 500—600 км. Зато ночью на этом диапазоне в европейской части России хорошо слышны станции Европы и Азии, а при очень хорошем прохождении радиоволн можно услышать даже станции Америки. Этот диапазон широко используется радиолюбителями для различных собраний радиолюбителей какого-либо региона, различного рода "круглых столов", когда на определенном участке диапазона собираются в назначенное время радиолюбители, объединенные общими интересами.

- **Диапазон 7 МГц** плотно забит работающими радиостанциями круглые сутки. Днем здесь слышны станции близлежащих регионов, удаленные на расстояние от 500 до 1000 км, причем зимой прохождения радиоволн лучше. Ночью, особенно после полуночи, на этом диапазоне часто встречаются очень редкие и удаленные станции, вызывающие всеобщий интерес коротковолновиков. Единственным недостатком этого диапазона является большое количество помех от мощных вещательных радиостанций, потому что чуть выше по частоте располагается широковещательный сорокаметровый диапазон.

- **Диапазон 14 МГц** — самый любимый диапазон для большинства коротковолновиков. Прохождение радиоволн на этом диапазоне хорошее почти круглые сутки. Ночами, особенно зимними, прохождение значительно хуже. В начале лета в утренние часы радиостанции Северной Америки зачастую слышны очень здорово. На европейской части России в течение всего дня хорошо слышны станции Западной Европы и Азии, но прохождение здесь временами бывает очень нестабильным. Громко слышимая

станция вдруг становится слышна все хуже, порой пропадает совсем, но потом снова появляется. Такие явления называются "замирениями". Иногда прохождение становится совсем плохим, и такое состояние может длиться несколько дней. Связаны подобные явления с выбросами солнечной энергии и так называемыми магнитными бурями.

- **Диапазон 21 МГц** также привлекает многих коротковолновиков. В ночные часы, как правило, никого не слышно, зато днем и особенно в годы повышенной солнечной активности на этом диапазоне можно запросто провести связь с радиостанциями Японии, Америки и Африки. При хорошем прохождении можно работать со станциями Австралии. Рано утром, а иногда и поздно вечером бывают очень громко слышны радиостанции американского континента. Прохождение здесь еще менее устойчивое, чем на диапазоне 14 МГц, но помехи от ведомственных и широкоэмиттерных радиостанций практически отсутствуют.
- **Диапазон 28 МГц** — это самый капризный из коротковолновых диапазонов, зато здесь можно проводить самые удаленные радиосвязи. Диапазон характеризуется очень глубокими замирениями, а зачастую прием сигналов радиостанции прекращается совсем, зато рядом вдруг появляются сигналы совершенно другой станции. Этот диапазон интересен также тем, что на нем выделен довольно большой участок, предназначенный для проведения радиосвязей через искусственные спутники Земли (ИСЗ). Проведение радиосвязей через ИСЗ является очень интересным занятием, которым увлекаются многие тысячи радиолюбителей-коротковолновиков. О том, как проводить такие радиосвязи, я расскажу после того, как познакомлю вас с так называемыми "цифровыми видами радиосвязи". Так называются виды радиосвязи, где главную роль играет компьютер.

В библиотеках можно найти много литературы по вопросам любительской радиосвязи, в том числе и описание особенностей различных любительских диапазонов.

Компьютер и цифровые виды радиосвязи

Компьютеры — как это все началось

С давних времен люди задумывались над созданием "вычислительных машин", механизмов, которые оказывали бы какую-то помощь человеку в проведении вычислительных действий. История сохранила для нас имена таких людей: Леонардо да Винчи (приблизительно 1500 год), Блез Паскаль (1653 год), Готфрид Лейбниц (1673 год).

Но самым значительным шагом в деле создания современного компьютера было создание группой ученых в Англии умной машины для расшифрования секретных кодов вражеских сообщений. Шла вторая мировая война и

Англия терпела массу неприятностей от немецких кораблей и подводных лодок. Правительство Англии поручило группе самых выдающихся ученых страны разработать мероприятия по защите страны. В начале 1940 года этой группой ученых и математиков был разработан целый ряд аппаратов, включая "Colossus", для декодирования секретных сообщений, используемых немцами в военных целях.

Первые подобные аппараты делались на основе электромагнитных реле, создавали много шума и потребляли большое количество электроэнергии.

Затем были разработаны специальные диоды и радиолампы. Новые электронные аппараты позволили создать новую технологию для этой техники, позволили избавиться от медленных и шумных переключающих реле и формировать сигналы намного быстрее и надежнее. Это было реальным началом создания нынешних вычислительных машин.

В 1944 году в Гарвардском университете в США был создан первый универсальный *программируемый* цифровой компьютер.

В 1946 году была включена в работу первая полностью электронная вычислительная машина, состоящая приблизительно из 18 000 диодов и радиоламп. Веса она около 30 тонн, занимала площадь около 150 квадратных метров, и нуждалась в собственной подстанции мощностью 150 кВт! Стоимость машины была, как сообщалось, \$ 600 000, по тем временам огромная сумма.

Первый коммерчески доступный компьютер UNIVAC-1 был введен в действие в 1951 году и использовался для обработки результатов голосования на президентских выборах в США 1952 года.

Транзистор был открыт в 1947 году и сразу же стал использоваться во всех типах электронных устройств.

Примерно в 1958 году в штате Техас была разработана первая интегральная схема или "IC", иногда называемая также "чипом". Эти миниатюрные схемы, подробности которых могут быть рассмотрены только при помощи микроскопа, содержат много тысяч транзисторов и других компонентов в схеме размером меньше, чем размер ногтя мизинца.

Компьютеры, разработанные до середины шестидесятых годов, были все очень громоздкие и дорогие. Но в 1965 Digital Equipment Corporation (DEC) создала первый мини-компьютер, PDP-8. Компьютер имел малую вычислительную мощность, но продавался менее чем за \$ 20 000. Это позволило, наконец, обычным ученым и инженерам получать доступ к компьютерам не только для вычислений, но и для управления и текущего контроля экспериментов лаборатории.

В 1973 году был разработан первый микропроцессор. Появилась возможность создать универсальный микрокомпьютер, который был бы много мощнее, чем первые малые компьютеры при стоимости меньше \$ 1000.

Хотя стоимость процессора стала доступной, купить компьютеры было невозможно из-за их отсутствия. Появились энтузиасты, которые на базе покупного процессора стали строить свои компьютеры. Программ никаких не было, создаваемые вновь программы приходилось вводить в машину вручную, выбивая на клавиатуре один байт за другим. Мне тоже пришлось заниматься такой "технологией", но только на десять-двенадцать лет позднее.

В середине семидесятых годов прошлого века фирма IBM начала использование дискеты (диаметром 8 дюймов, односторонняя, 128 Кбайт) в некоторых изготавливаемых машинах.

В это время стали создаваться операционные системы.

В начале 1980-х годов фирма IBM, гигант компьютерной промышленности, создала свою собственную версию персонального компьютера IBM PC. Стали создаваться "умные" программы как для производственных или коммерческих целей, так и для игр.

Есть и другие типы персональных компьютеров. Самый большой конкурент моделям IBM — Apple Macintosh.

Существуют компьютеры и других фирм, менее известные в нашей стране.

Процессоры, зародившиеся как основные элементы вычислительной техники, применили в радиосвязи профессиональные связисты. Именно в институтах и экспериментальных лабораториях были разработаны основные виды цифровой радиосвязи. Радиолюбители сначала пользовались уже готовыми разработками, а также разрабатывали для них новые компьютерные программы. Каждая из таких программ несла в себе что-то новое, более совершенное. Затем, по мере накопления опыта, стали создавать свои, новые, виды связи.

Заканчивая первую главу этой книги, привожу краткий перечень цифровых видов связи, применяющихся в настоящее время радиолюбителями. Подробное описание основных видов цифровой радиосвязи приводится в *главе 2*.

Телеграф как цифровой вид связи

Разумеется, если передача телеграфных сигналов осуществляется рукой радиста и телеграфным ключом, а прием этих сигналов на приемной радиостанции проводит "на слух" другой радист, то ни о какой цифровой связи не может быть и речи. Телеграф как цифровой вид связи существует тогда, когда на передающей станции формирование телеграфных сигналов осуществляет компьютер, а на приемной радиостанции дежурный радист только наблюдает за тем, как компьютер в автоматическом режиме ведет прием телеграфных сигналов. Телеграф как цифровой вид связи имеет существенный недостаток. Дело в том, что телеграфные сигналы состоят из посылок, когда передатчик посылает в эфир электромагнитную энергию — этими по-

сылками являются точки и тире — и паузами между этими посылками, когда передатчик не излучает в эфир энергию. При этом на приемной радиостанции во время приема точек или тире звучит сигнал далекого передатчика, а во время пауз слышны только шумы и трески эфира. Если эти шумы и трески бывают очень сильными, а такое явление наблюдается очень часто, компьютер принимает помехи за полезный сигнал и искажает принимаемый текст. Поэтому "компьютерный" телеграф часто используют комбинированным способом — передачу текста ведет компьютер, а при приеме радиолубитель контролирует правильность приема компьютером.

Компьютерные программы для телеграфа бывают очень полезны в качестве тренажеров для совершенствования приема на слух и передачи на ручном ключе телеграфных сигналов.

Телетайп

Телетайп является самым первым из цифровых видов связи, освоенных радиолюбителями. Еще до появления на радиостанциях компьютеров некоторые из наиболее подготовленных радиолубителей проводили радиосвязи телетайпом. Приемные и передающие телетайпные аппараты в ту пору представляли собой очень громоздкие сооружения, состоящие из электроники и реле. Громоздкими были и клавиатуры у таких аппаратов.

С появлением компьютера на любительской радиостанции телетайп стал самым простым и удобным из всех цифровых видов связи. При этом он применяется как для проведения коротких, обычных, радиосвязей, так и для передачи довольно больших по объему сообщений. В радиолубительской практике телетайп обычно обозначается как RTTY — Radio Tele Type.

Телетайп на сегодня является единственным видом цифровой радиосвязи, который используется в международных соревнованиях по радиосвязи.

AMTOR

Название этого вида связи расшифровывается как AMateur Telex Over Radio — любительская передача посредством радио. В 1970 году был разработан для применения в морском флоте новый вид телетайпа, названный SITOR (Simple Telex Over Radio). Затем этот вид связи разрешили использовать радиолюбителям, но с условием некоторой переработки. Переработку выполнил англичанин Peter Martinez в 1976 году. Получившийся новый вид связи был назван AMTOR. Целью создания этого вида связи послужило, по моему мнению, желание приобщить радиолубителей к новому виду связи и привлечь их к работе над недостатками телетайпа. Поэтому без большой ошибки можно сказать, что AMTOR — это модернизированный телетайп. Несколько лет тому назад этот вид связи исключительно широко использо-

вался западными коротковолновиками. Сегодня его популярность несколько снизилась в связи с появлением более новых видов.

РАCTOR

Название этого вида связи расшифровывается как PACeted Telex Over Radio — пакетная передача посредством радио. РАCTOR является большим шагом вперед по сравнению с АМTOR. Основным назначением этого вида связи является безошибочная передача больших по размеру сообщений на низкочастотных участках коротковолновых диапазонов. В настоящее время используется очень многими западными радиолюбителями для работы в радиолюбительской сети.

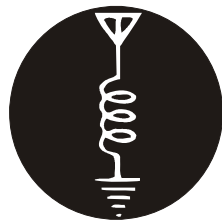
Packet Radio

В настоящее время это один из важнейших видов любительской цифровой связи, на базе которого построена Всемирная радиолюбительская сеть. Для многих коротковолновиков только этот вид связи предоставляет доступ к необходимой информации. По мере роста числа компьютеров будет расти и популярность Packet Radio.

Другие виды цифровой радиосвязи

Кроме перечисленных выше существует еще целый ряд цифровых видов связи, которые разработаны уже давно, но не пользуются большой популярностью. Особенно в нашей стране. Кроме того, за последние годы появились такие виды связи, как PSK31, MT63 и многие другие. Хотя некоторые из этих видов связи пока пользуются определенной популярностью, но, по моему мнению, широкого распространения они не получают из-за сложности в реализации или из-за повышенного создания помех другим станциям. Подробное описание каждого из основных видов цифровой связи приведено в *главе 2*.

Глава 2



Цифровые виды связи и программы для них

Предисловие

Один из моих знакомых, тоже радиолюбитель-коротковолновик, однажды задал мне вопрос примерно следующего содержания.

"По данным опроса владельцев компьютеров, более 70% от общего числа опрошенных постоянно работают только с системой Windows. Так почему же тогда ты не пишешь свои программы только для Windows с использованием новейших компиляторов, которые специально предназначены для программ, работающих под Windows? Таких как Visual C++ 6, C++ Builder 5, Delphi 6 и других?"

Вполне возможно, что подобный вопрос может возникнуть и у некоторых читателей этой книги.

Попробую ответить на этот вопрос.

Во-первых, среди указанных выше 70% владельцев компьютеров наверняка очень мало радиолюбителей-коротковолновиков. Скорее всего, коротковолновики входят в число оставшихся 30%. Для серьезного коротковолновика главную роль играют не привлекательные экранные заставки, которые предлагает нам Windows, а качество приема и передачи сигналов, которое лучше других операционных систем пока могут обеспечить нам Linux и MS-DOS.

Во-вторых, дело в том, что на таких замечательных компиляторах, как Visual C++ очень удобно писать так называемые "офисные" программы, которые работают с базами данных, и не совсем удобно писать программы, которые работают с различными внешними устройствами, подключенными к управляемым портам.

Программа для любого вида радиосвязи состоит из двух частей — первая часть предназначена для обслуживания экрана, для открытия и закрытия нужных файлов, и для других подобных нужд, вторая часть программы служит только одной цели — она обслуживает порт, к которому подключено какое-то устройство. Таким устройством может быть или контроллер TNC, или модем. Эта часть программы, которая предназначена для работы с внешним устройством, называется драйвером.

Драйвер всегда нуждается в специальных условиях, которые должны способствовать быстрой работе выполняемых программой операций. Поэтому драйверы стараются писать на языках самого низкого уровня, таких как Assembler, С или, в крайнем случае, Pascal.

Поэтому зачастую программа для цифрового вида связи разбивается на две отдельные части — программу для управления файлами и экраном и совершенно отдельную программу-драйвер. Зачастую драйверов для одной программы бывает даже несколько. Наглядным подтверждающим примером является широко известная программа для пакетной связи WINPACK 6.42. Собственно сама эта замечательная программа, образно говоря, не знает, как и подойти к СОМ-порту. Зато к этой программе существуют более десятка различных драйверов. Это драйверы для работы с различными контроллерами TNC, эмуляторы TNC и драйверы AGWPE, TFPCX, BPQ и много других.

Если взять программу для пакета WINTNC, разработанную G7JF, то эта программа не нуждается в дополнительных драйверах, потому что внутри себя содержит подпрограмму эмулятора TNC, в составе которой и содержится необходимый драйвер. Программисту видно, что эта программа, предназначенная для работы под управлением Windows, написана на языке программирования С с элементами С++. У этого же автора есть точно такая же программа, написанная для работы под управлением MS-DOS. Этот автор пошел на то, что одну и ту же программу сделал в двух вариантах — один вариант работает только под управлением MS-DOS, второй вариант работает только под управлением Windows.

При разработке описанных в этой книге моих программ я пошел по третьему пути — сделал свои программы таким образом, чтобы они могли одинаково хорошо работать и под управлением MS-DOS и под управлением Windows. Дело в том, что в нашей стране пока не каждый может приобрести современный компьютер, и нужны программы, по моему мнению, которые могли бы одинаково хорошо работать и на Pentium III и на 386 процессорах, как под управлением MS-DOS, так и Windows 95/98.

Телеграф как цифровой вид связи

Принципы построения телеграфного сигнала

В 1832 году на борту морского судна "Салли" один из пассажиров показывал опыт, как магнитная стрелка компаса начинает двигаться, когда к ней подносят кусок проволоки, присоединенной обеими концами к электрической батарее. За опытом внимательно наблюдал пассажир по имени Самюэль Морзе. Опыт натолкнул его на мысль попытаться создать систему передачи сигналов по проводам, и после пяти лет экспериментов в 1837 году ему уда-

лось это сделать. Для передачи он использовал ключ, изобретенный русским ученым Б. С. Якоби, а для приема — автоматическое устройство записи сигналов. Работая над созданием нового телеграфа, Морзе попутно изобрел и код, который успешно находит применение до сих пор. Основу кода Морзе составляют два знака — точка и тире.

После этого многие изобретатели пытались придумывать иные различные коды, из которых один был хитроумнее другого, но время выбрало лучший из них, который был создан американским инженером С. Морзе.

В телеграфе точки и тире — это посылки, отличающиеся друг от друга только длительностью и разделенные между собой паузами. Тире длиннее точки в три раза, пауза между точками и тире внутри буквы равна длительности одной точки, а длительность паузы между отдельными буквами равна длительности тире. Разными комбинациями точек и тире образованы все буквы латинского алфавита, цифры, знаки препинания и раздела. В русском алфавите есть буквы, которых нет в латинском, поэтому в русском варианте азбуки Морзе для них созданы свои комбинации точек и тире. Это буквы Ч, Ш, Э, Ю, Я.

Обучение приему на слух и передаче на ручном ключе сигналов кода Морзе для профессиональных радистов длится несколько месяцев. Хотя процесс этот очень сложный, но применение специального тренажера позволяет научиться принимать и передавать сигналы кода Морзе самостоятельно.

Надо сказать, что о методике самостоятельного изучения телеграфной азбуки до сих пор нет единого мнения. Одна из старых методик предлагает разбить весь алфавит на определенные группы из нескольких букв и проводить изучение по этим группам.

Следующие группы букв предназначены для изучения приема на слух телеграфных сигналов кода Морзе. Сигналы нужно слушать и обязательно тут же записывать принятую букву на бумагу.

- Первая группа — Т, М, Г, Ш, Е, И, С, Х.
- Вторая группа — А, Г, Ф, Б, З.
- Третья группа — У, К, Р, Щ, Й.
- Четвертая группа — Н, Ъ, Ж, Ю, Л.
- Пятая группа — В, Ы, Я, Ч, запятая, точка, восклицательный и вопросительный знаки.
- Шестая группа — Э, Ц, Д, П.
- Седьмая группа — 1, 3, 5 7, 0.
- Восьмая группа — 2, 4, 6, 8, 9.

Существует мнение, что азбуку Морзе могут изучить лишь люди, обладающие музыкальным слухом. Музыкальный слух, конечно, помогает, однако и

без него можно выучиться принимать и передавать телеграфные сигналы, только потребуются немного больше времени и терпения. Описанная ниже программа CW_QSO позволяет самостоятельно изучать прием на слух сигналов кода Морзе. Для этого сначала следует записать в файл буквы из какой-то одной изучаемой группы блоками (по пять букв в блоке и пробел между блоками), при этом буквы в разных блоках должны быть написаны в различных сочетаниях. Затем этот файл прослушивать, используя программу CW_QSO, на определенной скорости при одновременной записи на бумагу каждой прослушанной буквы. Начинать изучение следует с самой низкой скорости.

Кое-кто считает, что лишь после того, как новичок научится принимать 45 знаков в минуту, он может слушать "живой" эфир. Это неверно. Эфир надо слушать и пытаться записать услышанные буквы параллельно с изучением азбуки, с первых занятий. Не беда, если из сотен букв вы запишете лишь несколько, зато цена их намного выше, чем учебных.

Когда скорость приема станет стабильной и не хуже 45 знаков в минуту, можно приступать к разучиванию передачи на ключе.

Для обучения передаче рекомендуют разбить алфавит на следующие группы.

- Первая группа — Т, М, Щ, Ш, Е, И, С, Х.
- Вторая группа — 0, 5, 1, 9, 8.
- Третья группа — 7, 6, 2, 3, 4.
- Четвертая группа — А, Я, У, Ж, В, Й, Н, Д.
- Пятая группа — Б, Г, З, Ю, Р, Л, П.
- Шестая группа — К, Ь, Ф, Э, Щ, Ц, Ы.

Главное условие при разучивании передачи знаков — не торопиться с наращиванием скорости передачи. Любая спешка ведет к сокращению длительности каких-либо элементов, от чего сигнал становится непонятным для принимающего радиста.

В описываемой ниже программе CW_QSO имеется возможность подключения к игровому порту (порту джойстика) двухстороннего телеграфного ключа. При этом компьютер при передаче принимает на себя роль формирователя длительностей тире, точек и пауз. Оператору остается только примерно выдерживать необходимые длительности. Компьютер в этом случае превращается в нормальный полуавтоматический телеграфный ключ, позволяющий проводить тренировку или работать в эфире с разными скоростями. Но еще раз напоминаю, что ни в коем случае нельзя спешить с наращиванием скорости передачи. Простейшая конструкция такого телеграфного ключа может состоять из куска ножовочного полотна (от ножовки по металлу) длиной примерно 120 мм, деревянной дощечки размером 120×60×20 мм, двух металлических уголков 25×25 и длиной 25 мм, а также нескольких шу-

рупов по дереву и четырех винтов с гайками М3×10. На дощечке необходимо карандашом провести центральную линию, параллельную длинным сторонам. Один конец ножовочного полотна зажимается между двумя уголками четырьмя винтами, затем эта конструкция из уголков шурупами крепится к одному концу дощечки так, чтобы ножовочное полотно расположилось по центральной линии на расстоянии примерно 8 мм от дощечки. Незакрепленный конец полотна должен иметь возможность свободно вибрировать в горизонтальном положении параллельно дощечке. На расстоянии примерно 3 см от свободного конца полотна в дощечку по обе стороны полотна следует ввернуть два шурупа таким образом, чтобы расстояние между полотном и головкой шурупа было не более 2-х миллиметров. Теперь, если зажать свободный конец ножовочного полотна между большим и указательным пальцами правой руки, то при легком надавливании на полотно то одним, то другим пальцем, полотно будет касаться головок ввернутых в дощечку шурупов. Теперь остается к закрепленному концу ножовочного полотна припаять провод — это будет заземляющий провод, к головкам шурупов также следует припаять по проводу — это будут провода для точек и тире. О том, как подключать такой самодельный и простейший телеграфный ключ к порту джойстика, сказано ниже, в описании программы CW_QSO. Это только одна из возможных простейших конструкций манипулятора для полув автоматического телеграфного ключа. По мере приобретения навыков вы самостоятельно сделаете себе намного лучшую конструкцию, которая будет удовлетворять всем вашим требованиям. Нужно только предусмотреть возможность сделать основание ключа более тяжелым.

На рис. 2.1 схематично изображен один из возможных вариантов простого телеграфного ключа.

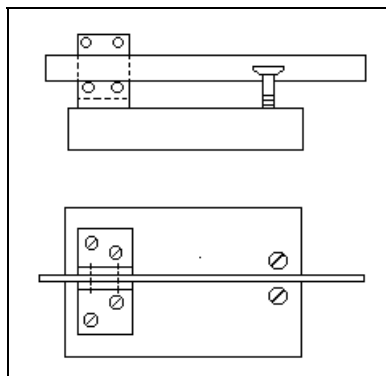


Рис. 2.1. Схема телеграфного ключа

Для желающих заняться изучением телеграфа в табл. 2.1 размещен код Морзе.

Таблица 2.1. Телеграфный код

Латинский алфавит	Русский алфавит	Код Морзе	Цифры, знаки препинания	Код Морзе
A	А	. —	1	. — — — —
B	Б	— ...	2	.. — — —
C	Ц	— . . .	3	... — —
D	Д	— ..	4 —
E	Е	.	5
F	Ф	.. — .	6	—
G	Г	— — .	7	— — ...
I	И	..	8	— — — ..
J	Й	. — — — —	9	— — — — .
K	К	— . —	0	— — — — —
L	Л	. — ..	9 (сокращенно)	— .
M	М	— —	0 (сокращенно)	—
N	Н	— .	Точка
O	О	— — —	Запятая	. — . — . —
P	П	. — — .	Дробная черта	. —
Q	Щ	— — — —	Вопросительный знак	.. — — ..
R	Р	. — .	Двоеточие	— — — ...
S	С	. . .	Восклицательный знак	— — . — —
T	Т	—	Точка с запятой	— . — . — .
U	У	. . —	Кавычки	. —
V	Ж	. . . —	Начало передачи	— . — — —
W	В	. — —	Знак раздела	— ... —
X	Ь	— .. —	Конец передачи	. —
Y	Ы	— . — —		
Z	З	— — ..		
	Ч	— — — .		
	Ш	— — — —		
	Э	.. — ..		
	Ю	.. — .		
	Я	. — . —		

Сначала код Морзе применялся на телеграфных линиях, которые по длинным проводам передавали сообщения от одного города к другому. После изобретения радио этот код стал использоваться для радиосвязей и пользуется большой популярностью среди профессиональных радистов и радиолюбителей-коротковолновиков вплоть до наших дней.

Передача на ручном телеграфном ключе и прием сигналов на слух — сложное и утомительное дело. Для облегчения труда радистов стали создаваться специальные сокращенные слова, которые содержали в себе определенную зашифрованную фразу. Так был создан международный радиолюбительский Q-код. Каждое слово этого кода состоит из трех букв, причем первой буквой всегда является буква Q. В настоящее время Q-код содержит примерно 70 фраз. В табл. 2.2 приведены некоторые из них, которые могут встречаться в дальнейшем в этой книге. Если кодовая фраза передана с последующим знаком вопроса, то это означает вопрос, иначе является ответом.

Таблица 2.2. Q-код

Q-код	Зашифрованная фраза
QRA?	Каков Ваш адрес?
QRA	Мой адрес...
QRB?	Каково расстояние между нашими радиостанциями?
QRB	Между нашими радиостанциями... километров
QRM?	Есть ли помехи от других радиостанций?
QRM	На этой частоте имеются помехи от других радиостанций
QRN?	Есть ли атмосферные помехи?
QRN	На этой частоте имеются атмосферные помехи
QRO?	Следует ли мне увеличить мощность?
QRO	Увеличьте мощность своей радиостанции
QRP?	Следует ли мне уменьшить мощность?
QRP	Уменьшите мощность своей радиостанции
QRZ?	Кто меня вызывает? Повторите вызов.
QRZ	Вас вызывает...
QSL?	Подтверждаете прием сигналов моей радиостанции?
QSL	Прием сигналов Вашей радиостанции подтверждаю
QSO?	Имеете ли Вы связь с...
QSO	1. Я имею связь с... 2. Любительская радиосвязь
QTH?	В каком населенном пункте находится Ваша радиостанция?
QTH	Моя станция находится в городе...

Кроме международного Q-кода существует также международный радиолобительский код, в котором употребляются сокращенные до двух-трех букв известные слова радиолобительского жаргона английского языка.

Все эти ухищрения направлены на повышение скорости приема и передачи телеграфных сообщений. Но все они ни в коей мере не могут сравниться с тем ускорением телеграфной передачи, которое может предоставить компьютер.

Как уже упоминалось ранее, телеграф может существовать в наше время в двух видах. Традиционный телеграф, когда на передающей радиостанции радист ведет передачу на ручном ключе, а на приемной радиостанции другой радист ведет прием поступающих сигналов на слух, никак не может называться цифровым видом связи.

Телеграф как цифровой вид связи существует тогда, когда на передающей станции формирование телеграфных сигналов осуществляет компьютер, а на приемной радиостанции дежурный радист только наблюдает за тем, как компьютер в автоматическом режиме ведет прием телеграфных сигналов. Телеграф как цифровой вид связи имеет существенный недостаток. Дело в том, что телеграфные сигналы состоят из посылок, во время которых передатчик посылает в эфир электромагнитную энергию (этими посылками являются точки и тире), и паузами между этими посылками, когда передатчик не излучает в эфир энергию. При этом на приемной радиостанции во время приема точек или тире звучит сигнал далекого передатчика, а во время пауз слышны только шумы и трески эфира. Если эти шумы и трески бывают очень сильными, а такое явление бывает очень часто, компьютер принимает помехи за полезный сигнал и искажает принимаемый текст.

Поэтому "компьютерный" телеграф часто используют комбинированным способом — передачу текста ведет компьютер, а при приеме радиолобитель на слух контролирует правильность приема компьютером телеграфных сигналов.

Программа CW_QSO работает с кодом Морзе

Когда компьютер появился на любительской радиостанции, ему сначала было предложено освоить роль секретаря, первого помощника радиолобителя. Затем многие радиолобители стали задумываться, а не заставить ли компьютер поработать с кодом Морзе? Эта идея пришлась по душе не всем радиолобителям.

Очень многие считали и считают в настоящее время, что не стоит отбирать у них романтику телеграфного ключа и приема на слух. Но дело в том, что никто ни у кого никакой романтики не отбирает! Каждый из коротковолновиков волен сам делать свой выбор. Эфир велик и места всем достаточно.

При освоении телеграфа компьютеру были поставлены жесткие условия при формировании символов (букв) из точек и тире. Было установлено, что если принять длительность звучания точки за единицу, то длительность тире

должна составлять три единицы, длительность паузы между точками или тире внутри символа должна быть равна длительности точки, а длительность паузы между двумя символами должна составлять три единицы, т. е. равняться длительности тире. Многие радиолюбители, начиная с 80-х годов прошлого столетия, взялись за создание компьютерных программ для использования в радиосвязи кода Морзе. Особенно много было создано различных тренажеров для изучения приема на слух телеграфных сигналов, для наращивания спортсменами скорости приема на слух телеграфных сигналов. Были созданы также и программы, которые позволяли компьютеру самостоятельно проводить телеграфные радиосвязи, т. е. самостоятельно проводить и прием, и передачу текстов, составляющих телеграфную радиосвязь. Основными из подобных (типовых) текстов являются следующие тексты.

1. Текст общего вызова, когда радиостанция выходит в эфир и сообщает всем, кто в данный момент принимает ее сигналы, о своей готовности принимать сигналы любой другой радиостанции, заинтересованной в проведении совместной радиосвязи.
2. Текст знакомства. Радиостанция сообщает корреспонденту имя оператора и местонахождение радиостанции, передает кодовое сообщение о качестве приема сигналов корреспондента и, зачастую, сообщает о погодных условиях в своем регионе.
3. Текст с просьбой повторить то или иное сообщение корреспондента.
4. Текст сообщения об аппаратуре, используемой в данный момент на станции.
5. Текст с подтверждением приема всех переданных корреспондентом сообщений.
6. Текст прощания с корреспондентом и передача наилучших пожеланий.

В пункте 2 сказано, что корреспонденты обмениваются кодовыми сообщениями о качестве приема сигналов корреспондента. Для такого случая существуют специальные кодовые фразы — RST, или RSM, или RS.

RST расшифровывается как "разбираемость — слышимость — тон". Каждый из этих параметров оценивается определенным количеством баллов. Так, разбираемость оценивается исходя из пяти баллов, слышимость и тон — из девяти баллов. RST 599 — отличная разбираемость при хорошей слышимости и хорошем тоне (без хрипоты) сигнала. RST характерен для телеграфа.

RSM — расшифровывается как "разбираемость — слышимость — модуляция".

Характерен для работающих модулированным сигналом, т. е. амплитудной, частотной или однополосной модуляцией сигнала. RSM 595 — отличная разбираемость при хорошей слышимости и нет искажений сигналов. Последние 30 лет вместо RSM всегда передают только RS, считая, что с плохой модуляцией в эфире делать нечего. Никто не станет проводить связи, если у станции отвратительный сигнал.

Кроме того, компьютер должен давать возможность оператору вести передачу непосредственно с клавиатуры или подключенного к компьютеру телеграфного ключа. Это очень важно для повышения оперативности в работе станции.

На различных страницах в Интернете можно найти самые разнообразные версии подобных программ, но все они имеют описания на английском языке и могут работать только с буквами латинского алфавита. На приложенной к данной книге дискете находится программа CW_QSO. Сочетание букв CW означает на радиолюбительском жаргоне телеграф, сочетание QSO является словом из международного Q-кода и обозначает проведение радиосвязи. Из названия понятно, что программа предназначена для проведения любительских радиосвязей телеграфом.

Эта программа может свободно распространяться среди радиолюбителей, для использования в коммерческих целях необходимо согласие автора.

Программа предназначена для проведения любительских радиосвязей посредством компьютера IBM PC под управлением MS-DOS или Windows 95/98. Кроме того, программа может служить хорошим тренажером для изучения приема и передачи телеграфных сигналов кодом Морзе. Для работы под управлением Windows 95/98 все файлы, входящие в пакет программы, следует разместить в одном подкаталоге, затем файловым менеджером **Мой компьютер** выбрать строку с исполняемым файлом cw_qso.exe и нажать правую кнопку мышки. В появившемся меню выбрать строку **Создать ярлык**. Образуется файл с расширением .pif, который можно переместить на рабочий стол и запускать программу непосредственно с рабочего стола компьютера.

Общие положения

Для нормальной работы компьютера с этой программой в эфире обязательно необходим отдельный аппарат, называемый "модем". Модем необходим для согласования компьютера с передатчиком радиостанции. Если использовать компьютер и программу в качестве тренажера или просто игрушки, то модем иметь необязательно. Для работы в эфире рекомендуется использовать MODEM3, файл с описанием которого можно скачать с приложенной к книге дискеты. Описание работы этого и других модемов будет выполнено в *главе 3*. Можно использовать модемы и других аналогичных конструкций.

Программа CW_QSO не имеет зарубежных аналогов и рассчитана на пользователей, предпочитающих при проведении радиосвязей использовать русский язык. Хорошо работает и с текстами, набранными латинскими буквами.

Перечень команд

Все задействованные в программе CW_QSO команды подаются нажатием на отдельные клавиши или сочетания из нескольких клавиш. Перечень команд приведен в табл. 2.3.

Таблица 2.3. Перечень команд

Клавиша	Действие команды
<F1>	Выдает на экран текст файла документации с описанием всех команд
<F2>	Увеличивает скорость передачи на одну ступень
<F3>	Уменьшает скорость передачи на одну ступень
<F4>	Увеличивает скорость приема сигналов корреспондента
<F5>	Уменьшает скорость приема сигналов корреспондента
<F6>	Разрешает ввод позывного корреспондента в таблицу данных
<F7>	Разрешает ввести с клавиатуры в таблицу имя корреспондента
<F8>	Разрешает ввести с клавиатуры в таблицу название города
<F9>	Разрешает ввести с клавиатуры в таблицу величину RST
<F10>	Разрешает ввести с клавиатуры величину температуры воздуха
<F11>	Вносит все необходимые данные в аппаратный журнал
<F12>	Выполняет обновление экрана
<Alt>+<F1>	Не используется
<Alt>+<F2>	Выдает на передачу текст общего вызова
<Alt>+<F3>	Выдает на передачу текст с просьбой повторить вызов (QRZ?)
<Alt>+<F4>	Производит вызов корреспондента по позывному из таблицы
<Alt>+<F5>	Выдает на передачу типовой текст начала радиосвязи
<Alt>+<F6>	Выдает на передачу заготовленный текст сообщения о своей аппаратуре
<Alt>+<F7>	Выдает на передачу типовой текст окончания радиосвязи
<Alt>+<F8>	Разрешает производить передачу непосредственно с клавиатуры
<Alt>+<F9>	Выдает типовой текст с просьбой повторить имя корреспондента
<Alt>+<F10>	Разрешает выдать на передачу текстовый файл с заданным именем
<Alt>+<X>	Выход в DOS
<Ctrl>+<F1>	Не используется
<Ctrl>+<F2>	Включает или отключает громкоговоритель компьютера
<Ctrl>+<F3>	Очищает окно передачи

Таблица 2.3 (окончание)

Клавиша	Действие команды
<Ctrl>+<F4>	Очищает окно приема
<Ctrl>+<F5>	Выдает на передачу текст из файла t5.msg
<Ctrl>+<F6>	Выдает на передачу текст из файла t6.msg
<Ctrl>+<F7>	Выдает на передачу текст из файла t7.msg
<Ctrl>+<F8>	Выдает на передачу текст из файла t8.msg
<Ctrl>+<F9>	Выдает на передачу текст из файла t9.msg
<Shift>+<F1>	Не используется
<Shift>+<F2>	Повышает тон звучания сигнала в громкоговорителе компьютера
<Shift>+<F3>	Понижает тон звучания сигнала в громкоговорителе компьютера
<Shift>+<F4>	Просмотр всех записей аппаратного журнала
<End>	Выдает на передачу позывные и переводит станцию в режим приема
<Home>	Выдает на передачу позывные и остается в режиме передачи
<Esc>	Срочный выход
<PgUp>	Переключает прием RUS/LAT символов

Для ввода команд в программе задействованы функциональные клавиши <F1>—<F12>, а также сочетания нажатия функциональной клавиши с клавишами управляющими — <Alt> и <Ctrl>. Так, после нажатия клавиши <F1> компьютер выдает на экран файл помощи с описанием всех задействованных команд. Клавишами <F2> и <F3> можно увеличить или уменьшить скорость передачи, клавишами <F4> и <F5> увеличивается или уменьшается скорость приема. После нажатия на клавиши <F6>—<F10> вводятся с клавиатуры в память компьютера позывной и другие данные станции-корреспондента. Клавиша <F8> предназначена для передачи текста непосредственно с клавиатуры. После нажатия <F8> компьютер ожидает ввод с клавиатуры очередного слова. Введенное слово отображается на экране. Затем следует нажать клавишу <Пробел> и введенное слово начинает передаваться. Таким же образом вводятся все последующие слова. Для прекращения работы с клавиатуры следует нажать <Enter>. Клавиша <F10> позволяет выполнить передачу приготовленного заранее файла величиной до 4 Кбайт. После нажатия на клавиши <Alt>+<F2>—<Alt>+<F10> компьютер начинает выдавать на передачу соответствующие сообщения или файлы. Все эти команды подробно описаны в документации к программе. Сочетания <Ctrl>+<Fx> служат таким же целям. Нажатием <Ctrl>+<F2> включается или отключается контроль передающихся сигналов через динамик компью-

тера, что особенно важно, если вы решили заняться изучением приема телеграфных сигналов, или просто для контроля за работой собственной радиостанции. Программа позволяет устанавливать наиболее приятный для слуха тон сигнала.

Подключение

Подключение модема к порту RS-232-C компьютера выполняется по следующей, обычно принятой, схеме, приведенной в табл. 2.4.

Таблица 2.4. Схема подключения

Название линии	Разъем DB25	Разъем DB9	Выполняемые действия
RTS	4	7	Включение прием/передача
DTR	20	4	Модуляция сигнала (передача)
CTS	5	8	Принимаемый сигнал (прием)
SG	7	5	Сигнальное заземление

Рабочий экран

Экран разбит на три части. Верхняя часть состоит из трех строк, в которых располагаются данные о вашем корреспонденте, текущая дата и системное время компьютера, вид работы (CW), температура воздуха за окном. В этой же строке рядом с температурой находится и величина в герцах тона компьютерного динамика, в строке скоростей первая цифра показывает скорость передачи, а вторая цифра — скорость приема. При включении компьютера эти две величины совпадают, но в дальнейшем можно изменять любую из них в больших пределах. В верхнем правом углу экрана при переключениях RUS/LAT появляются соответствующие надписи.

Справа и слева от слова **CW** располагаются символы индикатора настройки **о**. Слева символ **о** появляется при приеме тонового сигнала, в это же время символ справа исчезает. При отсутствии сигнала на экране находится только символ справа. В средней части экрана располагается окно принимаемой информации. В этой части экрана появляются окна с вспомогательной информацией.

В нижней части экрана расположено окно передаваемой информации. В этом окне сначала одним цветом записывается информация, предназначенная для передачи, а затем, во время передачи, каждая переданная буква закрашивается другим цветом. Это также предоставляет дополнительные удобства пользователям, изучающим прием телеграфных сигналов. После

передачи соответствующего текста окно очищается. В этом окне также появляются, при необходимости, информационные окна с соответствующими подсказками.

Кроме того, имеются две информационных строки — одна в самом низу экрана, вторая — между окнами приема и передачи. В строках находятся подсказки, которые могут понадобиться на первых порах при освоении программы.

Работа на передачу

В программе предусмотрены широкие возможности по выбору различных, заранее подготовленных текстов, а также существует возможность работы непосредственно с клавиатуры.

Для работы с клавиатуры следует нажать <Alt>+<F8>, дождаться пока программа выдаст в эфир позывные, затем как можно быстрее набрать на клавиатуре первое слово текста и нажать <Пробел>, при этом программа начинает передавать введенное слово. Затем таким же образом вводятся и передаются все слова необходимого для передачи текста. Для окончания работы с клавиатуры следует нажать <Enter>.

Особенностью программы является возможность передачи содержимого текстового файла размером до 4 Кбайт. Для этого следует нажать сочетание клавиш <Alt>+<F10>, появится информационное окно, которое подскажет, что нужно ввести имя файла. После ввода с клавиатуры имени нужного файла программа включается на передачу и нужный текст идет в эфир.

Передачу также можно вести и от руки с помощью телеграфного ключа. Для этого к игровому порту (порту джойстика) можно подключить простой двухсторонний телеграфный ключ, который позволяет работать в режиме полуавтоматического телеграфного ключа — при нажатии в одну сторону выдаются точки, при нажатии в другую — тире. Схема подключения ключа к игровому порту следующая:

- ☐ вывод 4 порта — GND (земля, корпус);
- ☐ вывод 2 порта — точки;
- ☐ вывод 7 порта — тире.

Этот вариант дает возможность совершенствоваться в передаче на электронном полуавтоматическом ключе и оперативно работать в эфире.

Журнал

Нажатием на клавишу <F11> открывается возможность сохранить данные по QSO в аппаратном журнале. При этом в аппаратный журнал *автоматически* вводятся все данные по корреспонденту из экранной таблицы данных —

позывной корреспондента, его имя, город и переданный от вас RST. Поэтому сразу же после нажатия на клавишу всплывает окно подсказки. Если вы действительно что-то упустили, то следует нажать <Esc> и команда будет отменена, если все нормально, то нажимайте <Enter> и вводите недостающие данные. Все введенные в журнал QSO можно просмотреть при помощи нажатия комбинации клавиш <Shift>+<F4>, при этом на экран выводятся не все, а только самые необходимые данные из журнала. Полностью всю информацию из журнала можно просмотреть, используя программу qso_log.exe. Это программа простого аппаратного журнала и должна находиться в одном каталоге с программой sw_qso.exe. Всю информацию, введенную в файл базы данных одной программой, можно просмотреть другой программой и наоборот.

Проверка скоростей передачи

Проверка указанных на экране скоростей передачи выполняется нажатием комбинации клавиш <Alt>+<F10> с использованием обычного секундомера. Для целей проверки в комплекте имеется файл G, в котором содержатся две строки по 10 слов "PARIS" в каждой, итого 100 символов с пробелами. Для проверки следует одновременно с запуском на передачу файла G включить секундомер и выключить секундомер одновременно с окончанием передачи последнего символа. Расчет выполняем по формуле:

$$(100 / t) \times 60 = A,$$

где 100 — число символов (знаков) в файле G; t — число секунд (по секундомеру); 60 — число секунд в одной минуте; A — скорость передачи (знаков в минуту).

Описанная в данной статье программа позволяет не только проводить связи телеграфом с высоким качеством приема и передачи, но и может служить очень удобным и надежным тренажером при обучении приему и передаче телеграфных сигналов кода Морзе. Или просто быть интересной игрушкой. Но работа этой программы в эфире невозможна без аппарата, называемого модемом. Конструкции нескольких таких аппаратов будут описаны в *главе 3*.

Как проводятся радиосвязи

В начале каждого из описанных в *главе 1* диапазонов имеется участок, специально выделенный для проведения телеграфных радиосвязей. Никакими другими видами связи здесь работать нельзя. Такие привилегии предоставляются только телеграфу. Чтобы начать работу в эфире, сначала по имеющимся приборам необходимо настроить передатчик, проверить качество согласования передатчика с антенной. Эти процедуры особенно важны для того, чтобы быть уверенным в надежной работе радиостанции и отсутствии помех от ее работы. Само собой разумеется, что задолго до этого была опробована работоспособность компьютера и модема. Сначала можно поработать

в эфире "на общий вызов". Для этого на выбранном для работы диапазоне находим свободную частоту, некоторое время наблюдаем за этой частотой, чтобы удостовериться в ее занятости. Затем нажимаем на клавиатуре компьютера сочетание клавиш <Alt>+<F2>. По этой команде компьютер тут же включает радиостанцию на передачу и в эфир идет текст общего вызова: "CQ CQ CQ de RA3XB RA3XB pse K". Этот текст полностью составлен из кодовых выражений. Если перевести эту фразу на "нормальный язык", то получится следующее: "ВСЕМ ВСЕМ ВСЕМ здесь радиостанция с позывным RA3XB RA3XB пожалуйста вызывайте". Если этот вызов услышит оператор радиостанции, который давно мечтает поработать с радиостанцией RA3XB, то он тут же начинает эту станцию вызывать. Услышав ответный вызов нашего корреспондента, нажимаем клавишу <F6> и вводим с клавиатуры позывной вызвавшей нас радиостанции. Начиная с этого момента позывной корреспондента будет постоянно находиться в памяти компьютера, пока мы не введем другой позывной или не выключим компьютер. В документации к программе CW_QSO довольно подробно описаны все возможные для этой программы команды и режимы работы. Происходит обмен информацией, после чего корреспондент уходит с частоты нашей радиостанции, и мы можем снова выдать в эфир общий вызов.

Большинство коротковолновиков работе "на общий вызов" предпочитают "охоту на DX".

DX на любительском жаргоне обозначает очень удаленную и очень редкую радиостанцию. "Охотник за DX" внимательно вслушиваясь в пiski, трески и другие шумы эфира, выискивает интересного корреспондента. Обычно связь с DX-станцией интересует очень многих и потому на частоте этой станции начинается настоящая свалка. Выигрывает тот, кто имеет больше опыта и лучшую антенную систему.

После успешного проведения связи с интересным корреспондентом нажимаем на клавиатуре клавишу <F11>. Компьютер автоматически делает в аппаратном журнале запись о проведенной радиосвязи. Просмотреть все записи аппаратного журнала можно сочетанием клавиш <Shift>+<F4>. Также это можно сделать и программой QSO_LOG, если эта программа находится в том же подкаталоге.

Нажатием комбинации клавиш <Ctrl>+<F2> можно включить или выключить контрольный динамик компьютера, который при работе на передачу воспроизводит передаваемый код Морзе. Сочетание клавиш <Shift>+<F2> и <Shift>+<F3> повышает или понижает тон этих сигналов. Клавишами <Ctrl>+<F5>—<Ctrl>+<F9> можно передать один из приготовленных заранее текстов. Если к игровому порту компьютера подсоединен телеграфный ключ, то передачу можно вести непосредственно с этого ключа. Нажатие на клавишу <PageUp> переводит компьютер в режим приема либо букв русского алфавита, либо латинского. По нажатию клавиши <F1> в любой момент можно получить перечень всех задействованных команд и другую информацию.

Компьютер часто бывает очень полезен при работе телеграфом, особенно начинающим коротковолновикам. Но этот вид связи — телеграф — не дает возможности компьютеру раскрыть все свои скоростные возможности. Дело в том, что телеграфный сигнал состоит из сочетаний наполненных сигналом точек и тире с паузами, во время которых сигнал должен отсутствовать. Но в реальном эфире полного отсутствия сигнала, полной тишины никогда не бывает! В эфире постоянно присутствуют всевозможные шумы и шорохи, к которым еще добавляются шумы приемных каскадов радиоприемника. Поэтому при больших уровнях шумов или помех компьютер начинает реагировать на эти помехи точно так же, как и на полезный сигнал. Происходит сбой, прием неверной информации. Чтобы избежать подобного явления, инженеры-связисты придумали оригинальный способ. Они во время пауз стали также передавать сигнал, но только совершенно другого тона. Так, при передаче полезного сигнала (я буду называть это "токовой посылкой") передается звуковой тон частотой 1000 Гц, а при передаче паузы (я буду называть это "бестоковой посылкой") передается звуковой тон частотой 1200 Гц. Величины 1000 Гц и 1200 Гц я выбрал условно. На деле они могут иметь совершенно другие значения.

Придуманный способ передачи токовых и бестоковых посылок звуковыми сигналами различных тонов позволил компьютеру совершать гораздо меньше ошибок при приеме сигналов из эфира. Появилась возможность увеличить скорость передачи информации, а для этой цели был разработан новый вид радиосвязи — телетайп. Принципы приема и передачи телетайпных сигналов рассмотрим в следующем разделе.

Программа QSO_LOG как дополнение к CW_QSO

Информационные данные по каждой проведенной радиосвязи радиолюбитель-коротковолновик должен записывать в аппаратный журнал. Во-первых, таково требование органов надзора, а во-вторых, каждый радиолюбитель заинтересован в сохранении этих данных. Кроме того, каждая радиосвязь должна подтверждаться с обеих сторон специальными почтовыми карточками. Это так называемые среди радиолюбителей "QSL-карточки". На красочной оформленной лицевой стороне располагается позывной принадлежавшей владельцу карточки любительской радиостанции, на обратной стороне — позывной корреспондента, дата и время проведения радиосвязи и т. д. Особенно нарядными и красочными бывают карточки у радиолюбителей из экзотических стран. Слово QSL взято из международного радиолюбительского Q-кода. Этот код был разработан очень давно и предназначался для сокращения передаваемых телеграфных текстов, но широко применяется радиолюбителями и в настоящее время как в обычных разговорах, так и в различных текстах.

QSL-карточки не только являются красочными "трофеями" радиолюбителя, но и дают право на получение различных как отечественных, так и зарубежных дипломов. Диплом представляет собой красочную своего рода "Почетную грамоту", в которую вписывается имя и позывной награждаемого и его достижения в радиоспорте. Например, Калужская федерация радиоспорта уже давно учредила и выдает диплом "К. Э. Циолковский". Для получения диплома следует набрать определенное количество очков за проведенные радиосвязи и подтвердить эти радиосвязи QSL-карточками.

Если при проведении обычной радиосвязи не возникает проблем с заполнением аппаратного журнала, то во время заочных соревнований по радиосвязи, когда в течение суток проводится несколько тысяч связей, заполнение журнала становится проблемой, но еще более неприятной проблемой является составление отчета о работе в соревнованиях. Это значит, что все записанные в журнале связи следует еще раз переписать на бланки установленной формы.

ПК становится "секретарем коротковолновика"

Компьютер легко справляется со всеми трудностями по ведению аппаратного журнала и составлению различных отчетов. В настоящее время в Интернете можно найти массу различных вариантов программ аппаратных журналов. В названии этих программ почти всегда встречается слово "log" — оно взято из английского морского жаргона и означает корабельный вахтенный журнал. Некоторая сложность в том, что все эти программы имеют документацию на английском языке и могут вызвать определенные трудности при освоении.

На приложенной к книге дискете можно свободно взять программу очень простого аппаратного журнала, которая может работать практически на любом из широко используемых компьютеров, начиная от компьютеров на базе 386SX до современных компьютеров на базе Pentium III.

Исполняемый файл этой программы называется qso_log.exe и находится в папке QSO_LOG.

Программа может работать и под управлением MS-DOS, и под управлением Windows 95/98. При работе с Windows 95/98 все входящие в состав программы файлы следует разместить в отдельном подкаталоге, затем файловым менеджером **Мой компьютер** отыскать исполняемый файл qso_log.exe и нажать правую кнопку мыши. В появившемся меню выбрать строку **Создать ярлык**, при этом образуется новый файл qso_log.pif. Для запуска программы в работу теперь нужно будет задействовать этот файл. Его можно также вывести на рабочий стол компьютера. Под управлением Windows 95/98 при запуске pif-файла программа фактически работает с MS-DOS и с очень хорошими результатами, не работает только принтер. Все меню и вся документация программы написаны на русском языке, программа имеет малые размеры, хотя

выполняет практически все функции, которые имеются в программах с размерами порядка мегабайта и более.

При первом включении программы появляется заставка с текстовым меню, которое выглядит следующим образом:

- 0 — информация о программе;
- 1 — добавить запись;
- 2 — выводить запись на экран;
- 3 — выводить запись как для отчета;
- 4 — изменить номер записи;
- 5 — сортировка записей по номерам;
- 6 — найти запись радиосвязи;
- 7 — вывести на печать;
- 8 — печатать как отчет;
- 9 — выйти из программы.

Если выбрать первую строку меню (команда 0), то на экран компьютера будет выведена вся необходимая для работы с программой информация.

Команда 1 позволяет вводить в память компьютера все необходимые по данной радиосвязи информационные данные. А именно: позывной радиостанции корреспондента, диапазон, дату и время проведения связи, информацию о разбираемости и слышимости радиосигналов корреспондента, информацию о разбираемости и слышимости сигналов вашей радиостанции корреспондентом, имя корреспондента, местонахождение радиостанции корреспондента (название города или села), вид модуляции сигнала при проведении радиосвязи, информацию по диплому (если QSL-карточка корреспондента нужна для получения диплома) и другую информацию или свои замечания.

Команда 2 выводит на экран постранично все записи аппаратного журнала. Эта процедура бывает иногда нужна для обзора всех проведенных радиосвязей.

Команда 3 выводит на экран также постранично все записи аппаратного журнала, но в формате отчета о соревнованиях по радиосвязи. Радиолюбители любят участвовать в различного рода заочных соревнованиях по радиосвязи. Такие соревнования организуют республиканские, областные или городские федерации радиоспорта или радиоклубы. Правда, для участников республиканских или международных соревнований нужна программа журнала с несколько большими возможностями.

Команда 4 позволяет изменить номер какой-либо записи. Следует заметить, что этой строкой меню следует пользоваться очень осторожно, потому что случайные ошибки при изменении данных могут вызвать неприятные явления.

Команда 5 позволяет провести сортировку записей строго по номерам, если перед этим приходилось изменять номера записей. Обычно эти две процедуры никогда не делаются, чтобы не внести случайных негативных изменений в журнал. Дело в том, что компьютер сам выполняет все записи четко и аккуратно и какое-то неправильное вмешательство может дать нежелательный результат. Пользоваться командами 4 и 5 следует только в крайних случаях, предварительно перед этим нужно сделать резервную копию файла qso_log.dat.

Команда 6 позволяет проводить поиск необходимой записи по позывному радиостанции корреспондента. Компьютер запрашивает искомый позывной и после ввода позывного с клавиатуры выдает на экран всю имеющуюся по данной радиостанции информацию. Если вы по каким-то причинам не помните необходимый позывной, то можно задать одну или несколько первых букв, которые могут быть в составе нужного позывного. В таком случае компьютер будет поочередно выводить на экран все записи, в которых имеется заданное сочетание букв. Например, если для поиска ввести только префикс UA3X, то программа начнет выводить на экран все записи с позывными от UA3XAA до UA3XZZ, если записи для этих позывных присутствуют в файле базы данных. Введем для поиска UA3X и увидим на экране:

Введите позывной: UA3X

Номер п/п: 2

Диапазон: 3,5

Дата: 12-12-1999

Время: 12:12

Позывной: ua3xfa

RST от Вас: 599

RST к Вам: 589

Имя корр.: Vlad

QTH корр.: Ludinovo

Модуляция: ssb

Диплом: net

Доп. Инфо.: net

Далее программа предложит нажать на клавишу <Enter>, после чего выдаст на экран всю информацию по следующему позывному.

Команда 7 позволяет вывести на принтер для печати все имеющиеся в журнале записи. Каждая запись печатается одной строкой, при этом выводятся только основные данные. Вспомогательная информация не печатается. Чтобы просмотреть всю запись по какой-то связи полностью, следует воспользоваться предыдущей строкой меню.

Команда 8 позволяет вывести на принтер для печати все записи в форме отчета о соревнованиях. В табл. 2.5 привожу пример одной строки подобной распечатки.

Таблица 2.5. Пример записи

№	Диа-пазон	Дата	Время	Позывной	Принят RST	Передан RST	Информация
105	14.1	23-12-2001	23:10	UA3XBI	588	599	д. Циолковский

Последняя строка меню служит для выхода из программы.

Программа QSO_LOG благодаря своей простоте и доступности позволит многим радиолюбителям в дальнейшем осваивать и более сложные программы ведения аппаратных журналов для любительских радиостанций.

Аппаратный журнал в других программах

В специальных программах для проведения любительских радиосвязей телеграфом CW_QSO и телетайпом — RTTY_QSO, одна из которых будет описана в следующем разделе, основная часть приведенной выше программы QSO_LOG встраивается как дополнительный модуль и используется для автоматического ввода в журнал записи о проводимой радиосвязи во время работы в эфире. Все записи о проведенных связях запоминаются в файле qso_log.dat, который является своеобразной базой данных. Этот файл имеет один и тот же формат с аналогичным файлом программы QSO_LOG и может, при необходимости, обрабатываться этой программой.

Главная задача программы QSO_LOG заключается в выводе на экран всей подробной информации о каждой записи и возможности распечатать на принтере все необходимые записи в формате отчета о соревнованиях по радиосвязи.

Компьютер и телетайп

Как построен телетайпный сигнал

Любительский телетайп (RTTY — Radio Tele Type) является, по-видимому, самым старым из всех видов цифровой радиосвязи и остается пока единственным из этих видов, на котором проводятся международные соревнования по радиосвязи. Взгляните на календарь международных соревнований, и вы в этом удостоверитесь. Этот вид связи не требует громоздких и сложных программ и аппаратов, дорогих компьютеров и дорогих принадлежностей к компьютерам. В то же время он позволяет быстро переходить от приема к передаче и наоборот, допускает возможность работать на передачу непосредственно с клавиатуры, достаточно хорошо (в смысле для радиолюбите-

ля) работает в условиях помех и сам создает гораздо меньше помех, чем некоторые новые виды цифровой связи. Ведь при любительской радиосвязи необязательно нужно принимать 100% информации, достаточно принять основную информацию, необходимую для оформления либо QSL-карточки, либо любительского диплома.

Практически каждые полгода появляются все новые и новые виды связи, но телетайп остается основным видом радиосвязи для очень многих радиолюбителей-коротковолновиков во всем мире.

Для каждого вида цифровой радиосвязи имеется соответствующий *протокол* — это принятый и утвержденный авторитетными международными организациями подробный перечень всех основных условий, которым должны безусловно соответствовать все параметры передаваемых в этом виде связи сигналов. При дальнейших усовершенствованиях *протокол* может только расширяться, но ни в коем случае не изменяет основных параметров. Если стоит вопрос об изменении основных параметров, то обновленный вид связи получает совершенно другое название.

В табл. 2.6 приводятся основные параметры телетайпного сигнала.

Таблица 2.6. Коды телетайпа

№	Латинский алфавит	Русский алфавит	Цифры и дополнительные символы	Информационные посылки	Шестнадцатеричный номер
00	Русский регистр, код 93			00000	00
01	T	Т	5	00001	01
02	Возврат каретки, код 10			00010	02
03	O	О	9	00011	03
04	Пробел, код 32			00100	04
05	H	Х	Я	00101	05
06	N	Н	,	00110	06
07	M	М	.	00111	07
08	Перевод строки, код 13			01000	08
09	L	Л)	01001	09
10	R	Р	4	01010	0A
11	G	Г	Ш	01011	0B
12	I	И	8	01100	0C
13	P	П	0	01101	0D
14	C	Ц	:	01110	0E

Таблица 2.6 (окончание)

№	Латинский алфавит	Русский алфавит	Цифры и дополнительные символы	Информационные посылки	Шестнадцатеричный номер
15	V	Ж	=	01111	0F
16	E	Е	3	10000	10
17	Z	З	+	10001	11
18	D	Д	Ч	10010	12
19	B	Б	?	10011	13
20	S	С	'	10100	14
21	Y	Ы	6	10101	15
22	F	Ф	Э	10110	16
23	X	Ь	/	10111	17
24	A	А	—	11000	18
25	W	В	2	11001	19
26	J	Й	Ю	11010	1A
27	Регистр цифр, код 124			11011	1B
28	U	У	7	11100	1C
29	Q	О	1	11101	1D
30	K	К	(11110	1E
31	Латинский регистр, код 91			11111	1F

Телетайп предназначен для передачи текстовой информации. Каждая буква текста перед выдачей в эфир кодируется специальным пятибитовым кодом, т. е. каждой букве назначается определенная последовательность токовых и бестоковых посылок. Например, в коде буквы Т имеются четыре бестоковых (1-я, 2-я, 3-я и 4-я) посылки и одна токовая (5-я), букве W соответствуют три токовых посылки (1-я, 2-я и 5-я) и две бестоковых посылки (3-я и 4-я). Эта кодировка закреплена *протоколом* и изменена быть не может.

Из пяти посылок оказалось возможным создать только 32 различных комбинации, поэтому в телетайпе используются только 32 символа.

Но для нормальной работы нужно было также использовать шесть служебных символов, так что на долю букв осталось только 26 мест. Этого количества хватало только для кодирования, а значит и для передачи в эфир лишь заглавных букв латинского алфавита. Для цифр и строчных букв этого же алфавита кодов не хватало. Вышли из положения вводом понятия "регистр". Были введены в практику и узаконены регистр заглавных букв, регистр

строчных букв и регистр цифр. Для того чтобы начать передачу текста из латинских заглавных букв, следует сначала передать на принимающую информацию станцию сигнал включения в работу регистра заглавных букв. Только после этого появляется возможность начать передачу заглавных букв. Чтобы начать передачу цифр, сначала на принимающую станцию отправляется команда включить в работу регистр цифр. При передаче строчных букв сначала передается сигнал включения регистра строчных букв.

Если для латинского алфавита 26 букв оказалось достаточным, то для работы с русским алфавитом пришлось отобрать несколько мест у цифр.

Пятью битами (посылками) можно закодировать только 32 символа (буквы), семью битами можно закодировать 128 символов. Для кодирования 256 символов необходимы восемь посылок. Все это находит применение в любом из компьютеров. Только из восьми посылок можно создать 256 кодовых комбинаций, достаточных для кодирования 256 стандартных символов, составляющих кодовую таблицу любого компьютера.

Теперь для понятливого читателя становится ясно, что нормальной телетайпной программой передать или принять бинарный файл невозможно при любых ухищрениях.

Посылки названы мною токовыми и бестоковыми условно. На самом деле все посылки передаются промодулированными звуковыми тонами с разными частотами. Так, если посылки, названные мною бестоковыми, посылаются в эфир промодулированными звуковой частотой 1000 Гц, то посылки токовые выдаются промодулированными с частотой 1170 Гц. *Протокол* требует, чтобы разница в частотах токовых и бестоковых посылок составляла 170 Гц. Создавая свои программы, многие программисты допускают возможность передавать бестоковые посылки тонами более высокой частоты (1170 Гц), а токовые — тонами низкой частоты (1000 Гц), т. е. допускают так называемый "реверс" сигналов. Приведенные мною цифры 1170 Гц и 1000 Гц условные. Величины частот могут быть иными, но разница между частотами для телетайпа должна всегда быть равной 170 Гц.

Как передать телетайпом символ

Каждая буква (символ) в телетайпном коде состоит из семи составных частей — посылок. Первой идет стартовая посылка, затем — пять информационных посылок, замыкает код символа стоповая посылка.

Протокол установил, что перед выдачей в эфир кода какой-то буквы непременно должна быть передана так называемая "стартовая" токовая посылка.

Принимающая станция после получения стартовой посылки становится готовой принять все последующие пять "информационных" посылок, составляющих код той или иной буквы. Кроме того, после передачи всех информационных посылок передающая станция передает в эфир токовую посылку, которая называется "стоповой посылкой". При этом по длительности стоповая посылка должна быть в полтора раза больше посылки информаци-

онной. *Протокол* устанавливает, что длительность каждой из перечисленных мною выше посылок должна подчиняться формуле:

$$T = 1000 / N,$$

где T — длительность информационной и стартовой посылок в миллисекундах; 1000 — число миллисекунд в одной секунде; N — скорость передачи RTTY в бодах.

Протокол также устанавливает скорость передачи сигналов в любительском телетайпе равную 45,45 Бод. При такой скорости длительность стартовой и каждой из информационных посылок должна составлять 22 мс, а длительность стоповой посылки получается равной 33 мс. Некоторые программисты предусматривают в своих программах возможность работать и на более высоких скоростях, но эти повышенные скорости можно использовать только для каких-то экспериментов. Например, для проверки работоспособности модема на больших скоростях и т. д. Чем выше скорости передачи телетайпного сигнала, тем короче становится информационная посылка, и тем более становится связь подвержена воздействию помех.

На рис. 2.2 приведена схема посылок, из которых состоит телетайпный сигнал при передаче символа.



Рис. 2.2. Структура телетайпного сигнала

Из схемы видно, что передача символа начинается с передачи стартовой посылки, затем следуют пять информационных посылок и завершает передачу символа посылка стоповая, длительность которой примерно в полтора раза больше. Такая схема передачи символа характерна для асинхронных видов цифровой связи.

Программа RTTY_QSO — один из вариантов телетайпной программы

Общие положения

Программа RTTY_QSO находится на прилагаемой дискете и предназначена для проведения любительских радиосвязей телетайпом посредством IBM PC или совместимого компьютера под управлением MS-DOS или Windows 95/98.

Программа может свободно распространяться среди радиолюбителей России по принципу "как есть", т. е. без гарантий со стороны автора и без претензий со стороны пользователей.

Программа не имеет зарубежных аналогов и рассчитана на пользователей, предпочитающих при проведении радиосвязей использовать русский язык. Хорошо работает и с текстами, написанными латинскими буквами.

Комплект состоит из следующих файлов, которые должны постоянно находиться в одном и том же каталоге:

- ☐ rty_qso.exe — основной исполняемый файл программного комплекта;
- ☐ rty_qso.cfg — конфигурационный файл;
- ☐ qso_log.dat — файл с данными аппаратного журнала;
- ☐ cq.msg — файл с текстом для передачи общего вызова;
- ☐ qrz.msg — файл с текстом просьбы повторить вызов;
- ☐ rpt.msg — файл с текстом просьбы повторить имя и город;
- ☐ eqp.msg — файл с описанием собственной аппаратуры;
- ☐ my_name.msg — файл с текстом о собственном имени и городе;
- ☐ konec.msg — файл окончания типовой радиосвязи;
- ☐ f5.msg,... , f9.msg — файлы с любыми дополнительными текстами.

Тексты всех файлов с расширением .msg могут изменяться пользователем по своему усмотрению, но при условии, что размер файла не должен превышать 380 знаков и не должен содержать более 5 текстовых строк. Размер файла для передачи при помощи клавиш <Alt>+<F10> не должен превышать 4000 байт. Каждая строка должна заканчиваться нажатием клавиши <Enter>. (Для последней 5-й строки этого делать не следует.) При подготовке текста, предназначенного для зарубежных корреспондентов, помните, что программы многих из них работают только с заглавными буквами.

Для работы в эфире рекомендую использовать модем MODEM22, описание которого дано в *главе 3*.

В программе применен оригинальный индикатор настройки на частоту корреспондента.

Программа допускает реверс (нормальный/обратный) как при приеме, так и при передаче.

Конфигурационный файл

Конфигурационный файл должен иметь определенное количество строк, при этом каждая строка должна начинаться с первой позиции, никакие пустые строки перед первой строкой и между остальными строками не допускаются.

Строки sfg-файла имеют следующие значения:

- ☐ 1-я строка — используемый COM-порт;
- ☐ 2-я строка — позывной собственной радиостанции;
- ☐ 3-я строка — собственное имя;
- ☐ 4-я строка — собственный населенный пункт;
- ☐ 5-я строка — адрес и наименование файла радиожурнала;
- ☐ 6-я строка — адрес и наименование файла, сохраненного QSO;
- ☐ 7-я строка — температура наружного воздуха в вашем городе;
- ☐ 8-я строка — позывной наиболее вероятного корреспондента;
- ☐ 9-я строка — имя наиболее вероятного корреспондента;
- ☐ 10-я строка — город наиболее вероятного корреспондента;
- ☐ 11-я строка — наиболее вероятный вариант переданного RST.

Каждая из этих строк может редактироваться самым простым текстовым редактором при соблюдении вышеописанных требований. Строки конфигурационного файла нельзя менять местами или исключать какую-то из строк. Например, если вы не хотите помещать позывной или имя наиболее вероятного корреспондента, то это не значит, что нужно выбрасывать всю строку. В таком случае достаточно просто вместо позывного или имени ввести пробелы.

Перечень команд

Все команды в программе RTTY_QSO подаются нажатием на отдельные клавиши или сочетания из нескольких клавиш. Перечень всех задействованных в программе команд приведен в табл. 2.7.

Таблица 2.7. Перечень команд

Клавиша	Действие команды
<F1>	Выдает на экран текст с описанием всех команд
<F2>	Увеличивает скорость передачи на одну ступень
<F3>	Уменьшает скорость передачи на одну ступень
<F4>	Выполняет реверс сигналов при приеме (прямой/обратный)
<F5>	Выполняет реверс сигналов при передаче (прямой/обратный)
<F6>	Разрешает ввод позывного корреспондента в таблицу данных
<F7>	Разрешает ввести с клавиатуры в таблицу имя корреспондента
<F8>	Разрешает ввести с клавиатуры в таблицу название города

Таблица 2.7 (продолжение)

Клавиша	Действие команды
<F9>	Разрешает ввести с клавиатуры в таблицу величину RST
<F10>	Разрешает ввести с клавиатуры величину температуры воздуха
<F11>	Вносит все необходимые данные в аппаратный журнал
<F12>	Выполняет обновление экрана
<Alt>+<F1>	Дополнительный текст помощи
<Alt>+<F2>	Выдает на передачу текст общего вызова
<Alt>+<F3>	Выдает на передачу текст с просьбой повторить вызов (QRZ?)
<Alt>+<F4>	Производит вызов корреспондента по позывному из таблицы
<Alt>+<F5>	Выдает на передачу типовой текст начала радиосвязи
<Alt>+<F6>	Выдает на передачу заготовленный текст сообщения о своей аппаратуре
<Alt>+<F7>	Выдает на передачу типовой текст окончания радиосвязи
<Alt>+<F8>	Разрешает производить передачу непосредственно с клавиатуры
<Alt>+<F9>	Выдает типовой текст с просьбой повторить имя корреспондента
<Alt>+<F10>	Разрешает выдать на передачу текстовый файл с заданным именем
<Alt>+<X>	Выход в DOS
<Ctrl>+<F1>	Информация о программе
<Ctrl>+<F2>	Включает или отключает громкоговоритель компьютера
<Ctrl>+<F3>	Очищает окно передачи
<Ctrl>+<F4>	Очищает окно приема
<Ctrl>+<F5>	Выдает на передачу текст из файла t5.msg
<Ctrl>+<F6>	Выдает на передачу текст из файла t6.msg
<Ctrl>+<F7>	Выдает на передачу текст из файла t7.msg
<Ctrl>+<F8>	Выдает на передачу текст из файла t8.msg
<Ctrl>+<F9>	Выдает на передачу текст из файла t9.msg
<Shift>+<F1>	Информация о программе
<Shift>+<F2>	Повышает тон звучания сигнала в громкоговорителе компьютера
<Shift>+<F3>	Понижает тон звучания сигнала в громкоговорителе компьютера
<Shift>+<F4>	Просмотр всех записей аппаратного журнала
<End>	Выдает на передачу позывные и переводит станцию в режим приема

Таблица 2.7 (окончание)

Клавиша	Действие команды
<Home>	Выдает на передачу позывные и остается в режиме передачи
<Esc>	Срочный выход
<PgUp>	Переключает прием RUS/LAT символов
<PgDN>	Ввести позывной корреспондента из информационной строки

Подключение

Подключение модема к порту RS-232-C компьютера выполняется по следующей, обычно принятой, схеме, приведенной в табл. 2.8.

Таблица 2.8. Схема подключения

Название линии	Разъем DB25	Разъем DB9	Выполняемые действия
RTS	4	7	Включение прием/передача
DTR	20	4	Модуляция сигнала (передача)
CTS	5	8	Принимаемый сигнал (прием)
SG	7	5	Сигнальное заземление

Рабочий экран

Экран разбит на три части. Верхняя часть состоит из трех строк, в которых располагается таблица переменных данных — данные о вашем корреспонденте, текущая дата и системное время компьютера, вид работы (RTTY), температура воздуха за окном, в этой же строке рядом с температурой находится и величина в герцах тона компьютерного динамика, в строке скоростей после величины скорости (в конце строки) находится символ включения контрольного динамика компьютера. Справа от надписи **RTTY** на черном фоне находится символ **o**. Это символ настройки на частоту корреспондента. При изменении тона символ перемещается в левую сторону от надписи. При точной настройке символы видны с обеих сторон от надписи **RTTY**. Идеальным можно считать тот случай, когда при приеме на экране видны только наиболее удаленные от слова **RTTY** символы **o**.

Слева от слова **ПРИЕМ** появляются символы, указывающие на прием нормальных или обратных символов, справа — символы состояния передачи.

Символы **нор** и **обр** появляются при нажатии <F4> и <F5> и соответствуют нормальному или обратному приему или передаче.

В верхнем правом углу экрана при переключениях RUS/LAT появляются соответствующие надписи.

В средней части экрана располагается окно принимаемой информации. В этой части экрана появляются окна со вспомогательной информацией.

В нижней части экрана расположено окно передающейся информации. Кроме того, имеются две информационных строки — одна в самом низу экрана, вторая — между окнами приема и передачи. В верхней информационной строке выдается информация об использовании принтера и файла сохранения, позывной вызывающей радиостанции, сведения о последнем QSO, проведенном с данным позывным.

Работа на передаче

Программа предусматривает подключение на передачу через последовательный порт RS-232 (COM1—COM4) к модулятору модема. При этом используется сигнал DTR (штырек 20 на DB25 или штырек 4 на DB9).

Возможен и еще вариант — взять сигнал от контрольного динамика через переходные конденсаторы и подать его на микрофонный вход передатчика.

В программе имеются большие возможности по выбору различных, заранее подготовленных текстов, а также возможность работы непосредственно с клавиатуры.

Для работы с клавиатуры следует нажать сочетание клавиш <Alt>+<F8>, дождаться пока программа выдаст в эфир позывные, затем как можно быстрее набрать на клавиатуре первое слово текста и нажать клавишу <Пробел>, при этом программа начинает передавать введенное слово. Затем таким же образом вводятся и передаются все слова необходимого для передачи текста. Для окончания работы с клавиатуры следует нажать клавишу <Enter>. Следует помнить, что клавиатура хранит несколько введенных символов в своей памяти пока идет передача ранее введенного слова, поэтому можно начинать ввод нового слова не дожидаясь конца передачи предыдущего.

Журнал

Нажатием на клавишу <F11> открывается возможность сохранить данные по QSO в аппаратном журнале. При этом в аппаратный журнал *автоматически* вводятся все данные по корреспонденту из экранной таблицы данных — позывной корреспондента, его имя, город и переданный от вас RST. Поэтому сразу же после нажатия на клавишу всплывает окно подсказки. Если вы действительно что-то упустили, то следует нажать клавишу <Esc> и команда будет отменена, если все нормально, то нажимайте <Enter> и вводите недостающие данные. Все введенные в журнал QSO можно просмотреть путем

нажатия сочетания клавиш <Shift>+<F4>, при этом на экран выводятся не все, а только самые необходимые данные из журнала. Полностью всю информацию из журнала можно просмотреть, используя программу qso_log.exe. Эта программа простого аппаратного журнала и должна находиться в одном каталоге с программой rttu_qso.exe. Все введенное одной программой можно просмотреть другой программой и наоборот.

База данных журнала хранится в файле qso_log.dat, при этом в одном таком файле может храниться только информация о 250 радиосвязях, т. е. при размещении в одном файле данных о 250 QSO этот файл нужно переименовать так, чтобы в имени была зашифрована информация о том, за какое время собрана информация, имеющаяся в файле. При этом для дальнейшей работы следует в текстовом редакторе создать абсолютно чистый файл qso_log.dat.

Пример нового имени: файл с именем 4_10_99.dat содержит записи QSO начиная с 4-го месяца (апрель) по 10-й месяц (октябрь) 1999 года.

Программа qso_log.exe также позволяет все записи из файла qso_log.dat как отчет о проведенных радиосвязях (соревнованиях) распечатать на бумагу.

В этой программе впервые применено довольно интересное новшество, аналогичного которому я не встречал ни в одной программе. Речь идет о задействованном в данной версии программы автоматическом вводе из эфира в информационную строку позывного вызывающей радиостанции.

Как только вас начинает вызывать какая-либо радиостанция, программа в автоматическом режиме принимает этот позывной и заносит его в информационную строку. Чтобы начать работу с этой станцией, следует нажать клавишу <PageDown>. При этом программа вводит позывной из информационной строки в таблицу переменных данных, проверяет по журналу наличие записи о связи с этим позывным и, если запись о такой связи имеется, заносит автоматически в таблицу переменных данных имя и город корреспондента и выдает в информационной строке сведения о последнем проведенном с этим позывным QSO. Это новшество значительно упрощает работу оператора при проведении связей.

Также при ручном вводе в таблицу переменных данных позывного корреспондента (по нажатию клавиши <F6>) программа просматривает все записи аппаратного журнала, вводит автоматически все необходимые данные по радиосвязям, проведенным ранее с этой станцией, и выдает соответствующую информацию. Если вместо полного позывного будет введен только префикс или часть префикса, то программа выдает информацию о последней связи с радиостанцией, позывной которой имеет данный префикс (или часть префикса).

Программа RTTY_QSO содержит подпрограммы приема и передачи, которые работают намного эффективнее аналогичных подпрограмм в разработках других авторов. Это проверено на практике. Можете убедиться сами по рекомендациям из следующего раздела.

На сегодняшний день среди радиолюбителей и в сети Интернет можно найти большое количество самых разнообразных программ для RTTY. Среди радиолюбителей ходят различные истории о чудодейственности той или иной программы, одни утверждают, что нет ничего лучше какой-то сверхдорогой звуковой карты, другие утверждают, что все чемпионы работают в режиме RTTY только через модемы фирмы N, третьи восхваляют модемы фирмы Z. Как правило, все подобные басни создают и распространяют не очень компетентные люди, а им вторят легковверные радиолюбители, готовые утверждать, что они лично видели программу RTTY, способную передавать и принимать бинарные файлы. Поэтому становится актуальным вопрос о том, как выбрать необходимую для вас RTTY-программу. Мое мнение таково — работа через правильно настроенный модем ничуть не хуже, чем работа через самую дорогую звуковую карту, и наоборот, работа через хорошую звуковую карту ничуть не хуже, чем работа через правильно настроенный модем.

Как выбрать лучшую из нескольких программ

Как выбрать программу для телетайпа

Если прослушать внимательно участки для цифровой радиосвязи на различных диапазонах, то оказывается, что наиболее применяемыми видами связи являются Packet Radio, PACTOR и RTTY. Старый и удобный телетайп продолжает во многом удовлетворять потребности радиолюбителей в повседневных коротких радиосвязях благодаря своей простоте в обслуживании, надежности и нетребовательности к аппаратным средствам. И это несмотря на постоянно появляющиеся новые виды связи. В это же время Packet Radio и PACTOR используются, в основном, для работы в радиолюбительской сети.

Существует большое количество различных программ для работы телетайпом посредством компьютеров, и добыть в Интернете любую из них труда не составляет. Но все эти программы, как правило, разработаны за пределами нашей страны, не учитывают особенности нашего языка, нашего алфавита. С этой проблемой я столкнулся сразу же после получения разрешения на работу телетайпом в 80-е годы. Пришлось самому садиться за учебники по программированию и пытаться создать программу, удобную для российского радиолюбителя. Такая программа была создана в 1990 году совместно с программой для проведения телеграфных радиосвязей. Эти программы предназначались для работы с единственно доступными тогда компьютерами "Радио-86РК". Потом доступными стали IBM PC и пришлось долго переучиваться и приспосабливаться к новым условиям, новым языкам программирования.

За период длительной работы в эфире различными видами цифровой радиосвязи и одновременной работой по созданию радиолюбительских программ

и модемов для цифровых видов связи, у меня сложились определенные критерии пригодности программы для российского радиолюбителя. Основные моменты этих критериев и применяемый мною уже много лет метод тестирования программ для цифровой радиосвязи я хочу описать в этом разделе.

Так каким же требованиям должна удовлетворять "хорошая" программа, предназначенная для массового российского радиолюбителя?

- ☐ Программа должна быть как можно короче и работать на самых простых компьютерах, потому что не все радиолюбители имеют возможность приобретать новейшие модели компьютеров. Программа должна как можно меньше загружать память.
- ☐ Программа должна нормально работать как под управлением MS-DOS, так и под управлением Windows 95/98.
- ☐ Программа должна включать в себя встроенный, пускай даже самый простой, журнал учета проведенных радиосвязей. Зачастую для радиолюбителя совершенно не нужен громоздкий, размером в мегабайты, журнал, напичканный никогда не востребованными функциями.
- ☐ Программа должна сохранять в специальном файле всю принимаемую из эфира информацию.
- ☐ Программа должна иметь возможность передать в эфир отдельный текстовый файл, пусть даже и не очень большой.
- ☐ Программа должна работать с самыми разнообразными конструкциями простых самодельных модемов, потому что многие из нас не имеют возможности приобрести дорогостоящие изделия известных фирм. При этом качество работы программы на прием и передачу должно быть не хуже, чем при работе с дорогостоящими фирменными аппаратами.
- ☐ Программа должна иметь всю документацию на русском языке, допускать использование в документации только слов международного радиолюбительского кода и жаргона.
- ☐ Программа должна работать как с заглавными, так и со строчными буквами и русского алфавита, и латинского алфавита.
- ☐ Программа должна иметь эффективный визуальный индикатор настройки на частоту корреспондента (желательно).
- ☐ На экране компьютера, перед глазами радиолюбителя, всегда должны находиться переменные данные — сведения о корреспонденте, т. е. позывной, имя и город корреспондента, передаваемый ему RST и прочие данные.

Уже длительное время для определения эффективности работы различных программ для различных видов цифровой радиосвязи, в том числе и для RTTY, мною используется простой метод, позволяющий с достаточной точ-

ностью выбрать наиболее эффективную из нескольких подобных программ. Метод очень простой и может быть повторен любым радиолюбителем.

При экспериментах я использую старый кассетный магнитофон и модем типа MODEM22. При этом магнитофон может иметь "не совсем чистые" головки и допускать большую детонацию звука. Это будет способствовать созданию (или имитации) помех. Заранее мною создан текстовый файл, который состоит из строк с заглавными буквами латинского алфавита от А до Z включительно. Назовем этот файл test. Выглядит этот файл следующим образом: первая строка состоит из двадцати расположенных подряд букв А; вторая строка состоит из двадцати расположенных подряд букв В и т. д. Последняя строка состоит из двадцати расположенных подряд букв Z.

Испытания провожу в следующем порядке.

1. На магнитофон от одной из испытуемых программ (в режиме передачи) через модем записываю текст файла test. Затем эту запись по очереди считываю каждой из испытуемых программ и фиксирую все ошибки приема. Полученные данные заносу в специальную таблицу.
2. Затем на магнитофон записываю тот же самый файл test второй из испытуемых программ, считываю эту запись по очереди каждой из испытуемых программ, все ошибки также заносу в таблицу.
3. Затем на магнитофон записываю тот же самый файл test третьей программой и т. д.

Испытания провожу на скорости 45,45 Бод. Если "однозначного" победителя выявить не удастся, то следующий этап испытаний провожу на повышенной скорости, например, 100 Бод. Но, как правило, второго этапа проводить не приходится.

Естественно, проводить такие испытания следует только при необходимости выявить программу с наилучшими показателями по приему и передаче сигналов. Если вы на первое место ставите наличие красивой картинка-заставки или наличие в программе нескольких видов цифровой связи, то читать далее эту книгу не стоит. Как показал опыт, в программах, написанных несколькими видами связи, применяются весьма примитивные подпрограммы приема и передачи, получить хорошее качество в таких случаях бывает трудно, а порой и невозможно.

Радиолюбители, заинтересованные в хороших результатах своей работы в соревнованиях, должны меньше доверять всевозможным распространяемым слухам, а должны самостоятельно выбирать себе программу для повседневной работы, а тем более для участия в соревнованиях.

Для того чтобы заинтересованный читатель мог сам поэкспериментировать в выборе нужной программы, я поместил на прилагаемую к книге дискету

еще одну из своих программ — программу RTTY_GT1. При проведении сравнения и выбора лучшей программы советую взять три программы: RTTY_QSO, RTTY_GT1 и программу Terman93, которая также может работать в режиме телетайпа.

Программа RTT_BBS — еще один из примеров применения телетайпа

RTTY_BBS — электронная доска объявлений

Длительное время мне пришлось быть руководителем самостоятельного радиоклуба и начальником коллективной радиостанции. Радиолюбители старшего поколения помнят организованные нашим клубом радиоэкспедиции по местам действий комсомольцев — Героев Людиновского партизанского подполья, радиоэкспедицию на Ильинский оборонительный рубеж, где курсанты подольских военных училищ своими сердцами защищали от вражеских полчищ столицу нашей Родины Москву, радиоэкспедицию на широко известную по популярной песне "Безымянную высоту" и много других мероприятий. При организации и проведении этих мероприятий и вообще в повседневной жизни клуба важную роль играла текущая информация, и один из важных инструментов распространения этой информации — доска объявлений. Этот "инструмент распространения информации" располагался в помещении клуба на видном месте. Кроме обязательных инструкций там находились объявления о сроках проведения радиосоревнований, о предстоящих собраниях, планы работы клуба, информация отдельных радиолюбителей об имеющихся у них проблемах или избытках каких-то радиодеталей. Постоянно вывешивалась информация из рубрики "На любительских диапазонах" газеты "Советский патриот".

Прошли годы, изменились люди, изменились страны, изменились и "инструменты распространения информации". В радиолюбительском мире действует широко разветвленная сеть на основе Packet Radio, основу которой составляют так называемые BBS и MBBS — своеобразные "почтовые ящики", которые способны переработать и доставить до адресатов огромное количество важной и интересной информации. Но проблему распространения местной информации, которая интересна только небольшому кругу радиолюбителей какого-то региона, глобальная пакетная сеть не решает, т. к. далеко не каждый радиолюбитель имеет возможность стать владельцем пакетной станции.

BBS (Bulletin Board System) расшифровывается как "доска объявлений для бюллетеней", MBBS (Mail and Bulletin Board System) расшифровывается как "доска объявлений для бюллетеней и почты". Эти понятия используются во Всемирной радиолюбительской сети.

По моему мнению, в большей степени задачу распространения местной информации может решить BBS на базе более распространенного вида связи — телетайпа. Для работы телетайпом достаточно иметь даже самый простейший компьютер типа "Радио-86РК" и несложный модем. В конце 80-х годов мне несколько лет пришлось работать телетайпом посредством этого компьютера и хочу заверить, что это почти то же самое, что и работа с помощью современных компьютеров и программ. Разработанная мною в те годы программа для проведения телетайпных радиосвязей посредством "Радио-86РК" могла одинаково хорошо работать с текстами и русского, и латинского алфавитов, имела встроенный аппаратный журнал, работала на скоростях до 150 Бод. Исходя из вышеперечисленных соображений, недавно мною была разработана программа "электронной доски объявлений" (или почтового ящика) RTTY_BBS.

Как работать с RTTY_BBS

Программа RTTY_BBS может применяться в качестве местной "электронной доски объявлений" и местными радиоклубами, и коллективными радиостанциями, вокруг которых формируются радиолюбители, и отдельными радиолюбителями, которые обладают интересной для других информацией. Практически RTTY_BBS обладает всеми функциями нормальной пакетной BBS, кроме функции проведения форвардинга. Перечень всех команд, доступных удаленному пользователю, приведен в табл. 2.9.

Таблица 2.9. Перечень команд BBS

Команда	Действие команды
HELP	BBS выдает файл помощи с перечнем всех команд
H	BBS выдает файл помощи с перечнем всех команд
I	BBS выдает информационный файл о своей радиостанции
K n	Уничтожить сообщение с номером n
KM	Уничтожить все прочитанные сообщение на ваш позывной
L	Получить список всех имеющихся бюллетеней
LM	Получить список всех имеющихся на ваш позывной сообщений
LN	Получить список всех имеющихся на ваш позывной новых сообщений
LL n	Получить список числа n последних бюллетеней
M	Получить список всех имеющихся для вас бюллетеней
R n	Прочитать сообщение номер n с заголовком
V n	Прочитать сообщение номер n без заголовка
RM	Прочитать все сообщения на ваш позывной с заголовками

Таблица 2.9 (окончание)

Команда	Действие команды
VM	Прочитать все сообщения на ваш позывной без заголовков
RN	Прочитать все новые сообщения на ваш позывной с заголовками
VN	Прочитать все новые сообщения на ваш позывной без заголовков
S	Послать сообщение
SB	Послать на BBS бюллетень
SP	Послать персональное сообщение на позывной
V	Запросить версию программы BBS
?	Получить файл помощи с перечнем всех команд

Радиостанция, оснащенная компьютером с программой RTTY_BBS и модемом, должна работать в заранее назначенное и известное всем заинтересованным пользователям время на строго определенной частоте. Предположим, что BBS имеет позывной UA3XBI, а удаленный пользователь имеет позывной RA3XB. Удаленный пользователь для работы с RTTY_BBS настраивается на заданную частоту и производит вызов:

```
RYRYRY UA3XBI UA3XBI DE RA3XB RA3XB K
```

Если BBS четко принимает и декодирует свой позывной и позывной вызывающего корреспондента, то сразу же начинает работать на передачу. Первое сообщение от BBS будет следующее:

```
"Привет, RA3XB... Здесь персональный BBS UA3XBI"
```

Далее BBS находит позывной вызвавшего корреспондента в аппаратном журнале, запоминает имя и другие данные этого корреспондента и ищет имеющуюся для него персональную информацию. Если персональная информация для данного корреспондента имеется, то BBS выдает перечень заголовков этой информации, иначе выдает сообщение:

```
"Для Вас ничего нового нет. Вам доступны все команды.  
BBS..."
```

Последняя строчка "BBS..." говорит о том, что BBS ждет команду и это буквосочетание является приглашением для ввода какой-либо из команд. Например, если вы желаете получить информацию обо всех доступных командах, следует выдать команду `HELP` или `h` или `?`. Для ввода команды существует определенная закономерность. Команда должна подаваться следующей строкой:

```
RYRY CMD HELP K
```

В этой строке непременно перед командой должно находиться буквосочетание `CMD` и один пробел, после командного буквосочетания также обязательно должен быть хотя бы один пробел. Начинаться командная строка должна буквосочетанием `RY`, повторенным хотя бы два раза (как в примере).

На праздничный день можно каждому вызывающему передать поздравление. Для этой цели существует специальный подключаемый приветственный файл.

При получении команды `HELP` или `h` или `?` программа выдаст файл помощи — перечень всех доступных пользователю команд. Следует учитывать, что вся выдаваемая программой информация идет буквами русского алфавита (на русском языке), но подавать команды на BBS следует только буквами латинского алфавита.

По команде `I` программа выдает информационный файл, в котором может располагаться информация о радиостанции BBS, об операторе станции, городе и т. д.

По команде `K A2` программа уничтожит персональное сообщение `A2`, если это сообщение адресовано вам. По команде `KM` программа уничтожит все персональные сообщения, адресованные на ваш позывной.

По команде `LM` программа выдаст перечень всех персональных сообщений на ваш позывной, по команде `LN` программа выдаст перечень только новых персональных сообщений на ваш позывной, еще не просмотренных вами.

По команде `L` программа выдает перечень всех имеющихся на данный момент бюллетеней, по команде `LL 4` выдаст перечень четырех последних бюллетеней.

Команды `SB`, `s` и `SP` служат для размещения в памяти BBS переданных вами бюллетеней или персональных сообщений. Различие между бюллетенем и персональным сообщением в том, что бюллетень доступен для чтения ВСЕМ пользователям BBS и адресуется, как правило, `wSEM`, а персональное сообщение адресуется на какой-то позывной и может быть прочитано и уничтожено только этим корреспондентом.

По команде `SB wSEM` программа запрашивает название бюллетеня, например, "Имею излишки радиодеталей". После ввода вами строки названия программа предлагает ввести текст бюллетеня. Текст обязательно должен заканчиваться нажатием клавиши `<Enter>` и вводом буквосочетания `/EX`. Это принято во всех пакетных сообщениях и я не стал нарушать установленное правило. Приняв заключительное буквосочетание `/EX`, программа сообщает о приеме от вас бюллетеня и предлагает ввести новую команду. Точно таким же образом по команде `SP RA3XB` вы можете передать на BBS персональное сообщение для `RA3XB`, которое сможет прочитать ваш корреспондент при первом же контакте с BBS. Команду `s` можно применять в сочетании с `wSEM` для бюллетеней.

По команде `RM` можно прочитать всю почту, адресованную на ваш позывной, при этом все сообщения будут выдаваться с расширенными сообщениями об отправителе, по команде `VM` можно прочитать всю ту же самую почту, но с краткой информационной строчкой об отправителе. По команде `RN` вы можете прочитать всю новую почту на ваш позывной с расширенной информацией об отправителе, по команде `VN` — то же, но с краткой информацией.

По команде `R A4` вы сможете прочитать сообщение с номером `A4`, по команде `R 2` — прочитать бюллетень с номером `2`.

Для прекращения связи с BBS служит команда `V`.

Коротко о самой программе

Программа `RTTY_BBS` имеет три рабочих режима.

Первый режим — режим собственно электронной доски объявлений (или почтового ящика, если это название больше вам нравится). В этом режиме программа ждет вызова от удаленного корреспондента и отвечает на его команды в автоматическом режиме. Это главный режим программы.

Второй режим — режим работы под управлением системного оператора (режим `sysop`). В этом режиме `sysop` может проверить действие всех команд BBS, подготовить бюллетени с нужной информацией и провести какие-то профилактические мероприятия.

Третий режим — режим терминала. В этом режиме программа работает как обычная `RTTY`-программа для проведения обычных телетайпных радиосвязей. Эта программа имеет много общего с описанной выше программой `RTTY_QSO`. В табл. 2.10 приведены команды терминальной программы. Следует обратить внимание на сочетание клавиш `<Alt>+<F9>`. По этой команде программа выдает на передачу не отдельное слово, а целую строку, в которой могут содержаться пробелы.

Таблица 2.10. Перечень команд

Клавиша	Действие команды
<F1>	Выдает на экран текст с описанием всех команд
<F2>	Увеличивает скорость передачи на одну ступень
<F3>	Уменьшает скорость передачи на одну ступень
<F4>	Выполняет реверс сигналов при приеме (прямой/обратный)
<F5>	Выполняет реверс сигналов при передаче (прямой/обратный)
<F6>	Разрешает ввод позывного корреспондента в таблицу данных
<F7>	Разрешает ввести с клавиатуры в таблицу имя корреспондента

Таблица 2.10 (продолжение)

Клавиша	Действие команды
<F8>	Разрешает ввести с клавиатуры в таблицу название города
<F9>	Разрешает ввести с клавиатуры в таблицу величину RST
<F10>	Разрешает ввести с клавиатуры величину температуры воздуха
<F11>	Вносит все необходимые данные в аппаратный журнал
<F12>	Выполняет обновление экрана
<Alt>+<F1>	Текст с дополнительной информацией
<Alt>+<F2>	Выдает на передачу текст общего вызова
<Alt>+<F3>	Выдает на передачу текст с просьбой повторить вызов (QRZ?)
<Alt>+<F4>	Производит вызов корреспондента по позывному из таблицы
<Alt>+<F5>	Выдает на передачу типовой текст начала радиосвязи
<Alt>+<F6>	Выдает на передачу заготовленный текст сообщения о своей аппаратуре
<Alt>+<F7>	Выдает на передачу типовой текст окончания радиосвязи
<Alt>+<F8>	Разрешает производить передачу непосредственно с клавиатуры
<Alt>+<F9>	Разрешает выдать на передачу строку текста с клавиатуры
<Alt>+<F10>	Разрешает выдать на передачу текстовый файл с заданным именем
<Alt>+<X>	Выход в DOS
<Ctrl>+<F1>	Дополнительная информация о программе
<Ctrl>+<F2>	Включает или отключает громкоговоритель компьютера
<Ctrl>+<F3>	Очищает окно передачи
<Ctrl>+<F4>	Очищает окно приема
<Ctrl>+<F5>	Выдает на передачу текст из файла t5.msg
<Ctrl>+<F6>	Выдает на передачу текст из файла t6.msg
<Ctrl>+<F7>	Выдает на передачу текст из файла t7.msg
<Ctrl>+<F8>	Выдает на передачу текст из файла t8.msg
<Ctrl>+<F9>	Выдает на передачу текст из файла t9.msg
<Shift>+<F1>	Информация о программе
<Shift>+<F2>	Повышает тон звучания сигнала в громкоговорителе компьютера
<Shift>+<F3>	Понижает тон звучания сигнала в громкоговорителе компьютера
<Shift>+<F4>	Просмотр всех записей аппаратного журнала

Таблица 2.10 (окончание)

Клавиша	Действие команды
<End>	Выдает на передачу позывные и переводит станцию в режим приема
<Home>	Выдает на передачу позывные и остается в режиме передачи
<Esc>	Срочный выход
<PgUp>	Переключает прием RUS/LAT символов
<PgDN>	Ввести позывной корреспондента из информационной строки

В программе есть интересная особенность — она автоматически принимает и записывает в информационной строке позывной вызывающей станции. Как только в эфире прозвучит вызов, сразу же позывной отображается в информационной строке. Если вы пожелаете провести связь с этой станцией, то стоит только нажать определенную клавишу и программа отыщет во встроенном аппаратном журнале последнюю связь с этим позывным и выдаст всю необходимую информацию об этой связи в информационной строке, а имя и другие важные данные запомнит. Так что оператору не нужно будет вручную вводить имя и город своего корреспондента. Задействованный в программе аппаратный журнал также просматривается программой при вводе в окно **call** с клавиатуры любого позывного.

Работа программы RTTY_BBS в режиме терминала мало отличается от работы с программой RTTY_QSO. В этом режиме можно проводить обычные телетайпные радиосвязи.

Файл с базой данных встроенного аппаратного журнала можно просмотреть полностью программой RTTY_BBS, а также специальной программой QSO_LOG, которая позволяет, кроме того, проводить различные поиски в файле базы данных и распечатывать записи аппаратного журнала на принтере в форме отчета участника соревнований.

Программа RTTY_BBS предназначена для компьютера IBM PC. Может работать под управлением и MS-DOS, и Windows 95/98. Работа программы проверена на компьютерах с процессорами 386SX 33 МГц и Pentium 100 в режимах испытательного стенда.

RTTY_BBS является программой экспериментальной, распространяется среди радиолюбителей свободно по принципу "как есть", т. е. без гарантий автора и претензий от пользователей. Вся документация к программе на русском языке. Скачать программу можно с прилагаемой к данной книге дискеты.

И еще немного о телетайпе

Историки свидетельствуют, что после изобретения С. Морзе телеграфного кода и внедрения этого кода в работу на телеграфных линиях связи стали

появляться разработки новых, более хитроумных кодов. Но время всем хитроумным изощрениям предпочло простоту и надежность. Код Морзе продолжает служить людям и сегодня, в то время как о более сложных и хитроумных изобретениях все давно уже забыли.

То же самое, по моему мнению, произойдет и с телетайпом. За последние время почти каждые полгода появляется новый вид цифровой связи, более изощренный, более напичканный всякими сложностями. Но наибольшей популярностью среди радиолюбителей все равно продолжает пользоваться простой и надежный телетайп. Так будет продолжаться и далее, пока не будут разработаны совершенно новые принципы цифровой радиосвязи.

AMTOR — модернизированный телетайп

Принципы построения сигналов в AMTOR

Долгое время телетайп оставался единственным широко распространенным видом цифровой связи, несмотря на имеющиеся в этом виде связи недостатки. В результате развития одного из вариантов модернизации традиционного телетайпа появилась новая система связи AMTOR. Как сообщил в своем ответе на одно из моих писем "отец" AMTOR англичанин Peter Martinez, этот вид связи был разработан в 1976 году. Название AMTOR произошло от фразы AMateur Teleprinting Over Radio. Вольный перевод этой фразы выглядит примерно так — "любительская передача печатных символов посредством радио".

Довольно длительное время AMTOR пользовался популярностью среди радиолюбителей. Существовали даже небольшие локальные сети, "почтовые ящики" и "информационные доски". Но с развитием других видов связи популярность AMTOR падает.

Несмотря на это, AMTOR заслуживает того, чтобы о нем не забывали.

Как и обычный телетайп, AMTOR использует для передачи токовых и бестоковых посылок звуковые частоты с разницей в 170 Гц. Так, например, если бестоковая посылка передается звуковым сигналом частотой 1000 Гц, то токовая посылка — 1170 Гц.

Имеются и существенные отличия.

- Каждый символ кодируется исходя из семибитового кода, в то время как в RTTY кодирование выполнялось по пятибитовому коду. Это значит, что каждый символ имеет семь информационных посылок. Стартовые и стоповые посылки отсутствуют. При этом каждый символ обязательно должен иметь в своем составе четыре единицы и три нуля, т. е. четыре бестоковые и три токовые посылки. Этот принцип используется для обнаружения ошибки.

- AMTOR является синхронной системой, в которой через определенные промежутки времени передаются специальные синхронизирующие символы. Эта особенность накладывает серьезные ограничения на скорость переключения с приема на передачу и наоборот. Время переключения не должно превышать 20 мс.
- Каждая информационная посылка имеет длительность 10 мс. Тогда длительность символа составит 70 мс. При этом скорость передачи постоянная и равна 100 Бод.
- Символы передаются блоками (пакетами), каждый из которых содержит три символа. Длительность передачи одного блока составляет 210 мс.
- Вместо нормального позывного AMTOR-станция использует при первом установлении контакта так называемый SelCall — часть позывного, состоящую из четырех символов. Например, обладатель позывного UA3XBI может для своей станции назначить SelCall UXBI.
- AMTOR имеет три основных рабочих режима — режимы А (ARQ), В (FEC) и L (Listen).

В табл. 2.11 приведены коды символов AMTOR.

Таблица 2.11. Коды символов AMTOR

№	Латинский алфавит	Русский алфавит	Цифры и дополнительные символы	Информационные посылки	Шестнадцатеричный номер
00	Русский регистр, код 93			0101011	2B
01	T	Т	5	0010111	17
02	Возврат каретки, код 10			0001111	0F
03	O	О	9	1000111	47
04	Пробел, код 32			0011101	1D
05	H	Х	Я	1001011	4B
06	N	Н	,	1001101	4D
07	M	М	.	1001110	4E
08	Перевод строки, код 13			0011011	1B
09	L	Л)	1010011	53
10	R	Р	4	1010101	55
11	G	Г	Ш	1010110	56
12	I	И	8	1011001	54
13	P	П	0	1011010	5A
14	C	Ц	:	1011100	5C

Таблица 2.11 (окончание)

№	Латинский алфавит	Русский алфавит	Цифры и до- полнительные символы	Информацион- ные посылки	Шестнадцатерич- ный номер
15	V	Ж	=	0011110	1E
16	E	Е	3	0110101	35
17	Z	З	+	1100011	63
18	D	Д	Ч	1100101	65
19	B	Б	?	0100111	27
20	S	С	'	1101001	69
21	Y	Ы	6	1101010	6A
22	F	Ф	Э	1101100	6C
23	X	Ь	/	0101110	6E
24	A	А	-	1110001	71
25	W	В	2	1110010	72
26	J	Й	Ю	1110100	74
27	Регистр цифр, код 124			0110110	36
28	U	У	7	0111001	39
29	Q	О	1	0111010	3A
30	K	К	(0111100	3C
31	Латинский регистр, код 91			0101101	2D
Служебные символы					
<RQ>				0110011	33
alpha				1111000	78
beta				1100110	66
CS1				1010011	53
CS2				0101011	2B
CS3				1001101	4D

Во многих литературных источниках для обозначения информационных посылок (битов) применяются обозначения Y и B, при этом $Y = 0$, $B = 1$.

Буквы русского алфавита я привел на всякий случай, возможно, они кому-то могут потребоваться. Во всех известных мне программах AMTOR используется только латинский алфавит.

Кратко о рабочих режимах AMTOR

□ **Режим L (Mode L — Listen).** В этом режиме проводится наблюдение за работающими на данной частоте станциями. Если другая работающая на данной частоте станция ведет передачу в режиме В (FEC), то вся принимаемая информация выводится на экран компьютера. Если другая станция работает в режиме А (ARQ), то на экране появляются некоторые символы, проконтролировать передающую станцию в этом случае невозможно. В некоторых программах предусмотрена возможность отвечать в автоматическом режиме на свой SelCall, если вашу станцию вдруг начинает вызывать другая станция, работающая в режиме ARQ. В этом случае появляется возможность провести нормальную радиосвязь с синхронизацией.

Если другая станция после работы на передачу в режиме FEC сообщает о том, что переходит на прием, то с этой станцией можно провести нормальную радиосвязь без синхронизации, подобно радиосвязи RTTY.

□ **Режим В (FEC — Forward Error Correction).** Применяется для передачи каких-либо сообщений для большого числа станций. Также позволяет проводить радиосвязи без синхронизации с другой станцией, при этом одна из станций ведет передачу в режиме FEC, вторая в это время слушает в режиме Listen. И наоборот. В названии режима говорится о системе выявления ошибок. Имеется в виду, что для этих целей каждый символ передается два раза со сдвигом через четыре символа. При первом приеме символ заносится в специальный буфер, а через четыре символа программа сравнивает только что принятый символ с содержимым буфера. Если эти два символа совпадают, то изображение символа выводится на экран. В случае несовпадения на экране появляется или пробел, или какой-то экзотический символ.

□ **Режим А (ARQ — Auto ReQuest).** Применяется при работе двух станций в режиме синхронного соединения. Работа ведется в полудуплексном режиме, причем одна из станций всегда является главной (передающая информацию), вторая — вспомогательной (подтверждает прием или просит повторить). Передающая станция передает пакет из трех символов за 210 мс и тут же переходит на прием для получения подтверждения от приемной станции. После получения подтверждения она должна передавать следующий пакет и т. д. Время от начала передачи одного пакета до начала передачи следующего пакета составляет 450 мс. Следовательно, для двух переключений и приема подтверждающего символа (70 мс) остается $(450 - 210) = 240$ мс. Если считать, что на два переключения пойдет $(20 + 20) = 40$ мс, то на ожида-

ние подтверждающего символа остается 200 мс. Это время ограничивает максимальную дальность связи для AMTOR.

Для вызова корреспондента оператор передающей станции должен ввести в программу SelCall вызываемой станции и включить передачу в режиме ARQ.

Как только принимающая станция начнет принимать и правильно дешифровать полученный вызов, она начинает в промежутки между пакетами передающей станции вставлять подтверждающие символы. Передающая станция, в свою очередь, как только начнет принимать эти подтверждающие символы, сразу же переходит к нормальному обмену информацией в режиме синхронизации. Подтверждающие символы, которые передает принимающая станция, называются CS1, CS2, CS3, alpha, beta и RQ. При успешном приеме пакетов в качестве ответов идут символы CS1 и CS2. Если произошла ошибка, в ответ повторяется предыдущий символ. Если вдруг главная станция не приняла подтверждающий символ, она передает пакет из трех символов RQ вместо информационного пакета, в ответ вспомогательная станция должна повторить предыдущий символ. Если вспомогательная станция хочет "перебить" главную и поменяться ролями, то она передает символ CS3, после чего станции меняются ролями. Для окончания связи главная станция передает пакет из трех символов alpha, после чего происходит разъединение. Если инициатива разъединения исходит от вспомогательной станции, то она вначале должна передать запрос, чтобы стать главной, а уж затем выдать пакет из трех символов alpha.

В процессе работы в режиме ARQ обе станции должны вести непрерывный учет количеству ошибок, следующих одна за другой. Если счетчик ошибок дойдет до заранее установленной величины (например, 32), то должна произойти процедура повторной синхронизации станций с сохранением текущего статуса.

Необходимость такой процедуры вызвана тем, что в системе AMTOR, в отличие от RTTY, нет стартовых и стоповых посылок и данные передаются в синхронном режиме при строгом совпадении показания "часов" на одной и другой станции. При большом несовпадении показания часов накапливается ошибка и связь становится невозможной.

Главная станция должна корректировать свои "часы", отслеживая ответы вспомогательной, и наоборот. После повторной синхронизации возобновляется передача данных из буфера с того места, где она была прервана. Собственно процедура синхронизации такова: главная станция в течение примерно 5 секунд передает пакеты синхроблоков, а вспомогательная принимает их в сдвиговый регистр и сравнивает с эталоном. При успешном сравнении начинает отвечать чередующимися кодами CS1/CS2. Главная станция при этом прекращает передачу синхроблоков и продолжает передачу нужной информации.

Существует два типа пакетов синхроблоков. Первый тип содержит RQ на втором месте, а первое и третье места в пакете занимают обычные символы. Второй тип имеет RQ на третьем месте, а первое и второе места в пакете занимают обычные символы. Например, при передаче CQ первый тип синхроблока будет: "C — RQ — Q", второй тип — "C — Q — RQ".

Если в буфере главной станции уже нет информационных символов для передачи, то она, для поддержания линии связи, передает пакеты из трех вспомогательных символов beta. Принимающая станция при этом должна брать роль главной станции на себя.

И еще немного о режиме ARQ

Для более любознательных читателей привожу некоторые дополнительные сведения о режиме ARQ.

Этот режим более других устойчив к искажениям информации, но он и сложнее других режимов в реализации, т. к. на каждую порцию информации, посланной передающей станцией, принимающая станция передает подтверждающий сигнал. В режиме ARQ могут работать только две станции. В зависимости от ситуаций, складывающихся как в начале, так и в течение контакта, станциям присваиваются определенные статусы, в соответствии с которыми регламентируется их работа.

- ☐ Статус **"ведущая станция"** присваивается вызывающей станции в начале контакта. Все действия участников контакта синхронизируются с передачами ведущей станции.
- ☐ Статус **"ведомая станция"** присваивается вызываемой станции в начале контакта и, как и предыдущий статус, не переприсваивается в течение всего контакта.
- ☐ Статус **"передающая информацию станция"** (ISS) присваивается станции, которая в текущий момент передает сообщения пользователя. Этот статус может переприсваиваться в течение контакта.
- ☐ Статус **"принимающая информацию станция"** (IRS) присваивается станции, которая в текущий момент принимает и печатает сообщение пользователя. Этот статус также может переприсваиваться в течение контакта.

Так как обе станции активны в течение всего контакта, два последних статуса указывают текущее направление потока информации. При изменении направления трафика эти два статуса меняются на противоположные.

Как же происходит обмен информацией? ISS разбивает передаваемое сообщение на блоки (пакеты) по три символа в каждом и последовательно их передает. Блоки попеременно нумеруются "Блок 1" и "Блок 2", причем нумерация блоков ведется синхронно на обеих станциях. После передачи одного блока ISS ожидает ответ IRS, которая сообщает, принят блок или нет, передавая соответствующий управляющий символ. Если "Блок 1" был при-

нят без ошибок, то IRS передает управляющий символ CS2, если "Блок 1" принят с искажением, то IRS передает символ CS1. Для "Блока 2" управляющие символы меняются местами. Это легко запомнить с помощью следующего правила: ответом IRS служит управляющий символ с номером требуемого информационного блока. Таким образом, если ISS приняла управляющий сигнал с номером только что переданного блока, она повторяет его передачу. Если ISS не получила подтверждающего символа, то она передает специальный блок "Требование повтора", состоящий из трех символов RQ. Получив такой блок, IRS повторяет переданный символ.

Теперь рассмотрим, как происходит изменение направления потока информации. Здесь возможны два случая: IRS прерывает работу ISS для передачи своего сообщения, либо ISS изменяет направление трафика после окончания передачи сообщения. В первом случае для изменения направления трафика IRS в ответ на очередной принятый без ошибок информационный блок передает управляющий символ CS3. ISS, получив символ CS3, передает так называемый блок "Over", состоящий из служебных символов "beta" — "alpha" — "beta", после чего получает статус IRS. После приема блока "Over" IRS получает статус ISS и передает либо блок из трех символов RQ, если она имеет статус ведомой станции, либо просто один символ RQ, если имеет статус ведущей станции. Это различие объясняется необходимостью сохранения периодичности циклов обмена по времени.

Во втором случае ISS передает три последовательных символа "Регистр цифр" — "+" — "?". Эта последовательность должна быть последней в передаваемом сообщении и необязательно должна передаваться одним блоком. Для дополнения блока, в случае необходимости, используется символ beta. После приема указанной последовательности IRS передает управляющий символ CS3, и дальнейшие действия аналогичны первому случаю.

Для окончания связи ISS передает блок "Конец связи", состоящий из трех символов alpha. Получив такой блок, IRS подтверждает его прием, и обе станции отключаются, переходя в режим ожидания. Кроме этого, прерывание связи может наступить, если корреспонденты длительное время друг друга не принимают или когда количество ошибок превышает определенное значение. В этих случаях ведущая станция сразу начинает проводить попытки восстановления контакта. Если контакт восстановлен, то передача информации продолжается с прерванного блока. Если перед прерыванием статус ISS имела ведомая станция, то сразу после восстановления контакта она проводит процедуру изменения направления трафика.

Установление контакта, а также его восстановление происходит следующим образом. Ведущая станция попеременно передает два блока с SelCall ведомой станции. SelCall-сигнал состоит из четырех алфавитных символов и обычно формируется из позывного станции. Рекомендуется использовать первый и три последних алфавитных символа позывного станции. Так, позывному

UA3XCE будет соответствовать позывной сигнал UXCE, а общему вызову — CQCQ. Чтобы принимающая станция могла отличить блоки с позывным от информационных блоков, в позывные блоки добавляется служебный символ RQ: в первом блоке символ RQ вставляется во вторую позицию, а во втором — в третью позицию. Таким образом, позывным сигналам в приведенном выше примере будут соответствовать следующие блоки: "U-RQ-B", "C-E-RQ" и "C-RQ-Q", "C-Q-RQ". Подтверждение правильного приема обоих позывных блоков осуществляется передачей одного и того же управляющего сигнала — CS1 или CS2. Если принятый позывной сигнал совпадает с позывным сигналом станции, то вызываемая станция получает статусы "ведомая" и IRS, и начинается процесс обмена информацией.

Проведенные многими радиолюбителями эксперименты показывают, что, AMTOR в режиме ARQ по помехозащищенности превосходит пакетную связь на средневолновом и низкочастотных коротковолновых диапазонах. Это превосходство объясняется передачей за один раз более короткой порции информации, чем в пакетной связи.

Программа Terman93 работает в режиме AMTOR

О программе Terman93

Программа Terman93 разработана известным программистом из Швейцарии Томом Сайлером (Tom Sailer, HB9JNX) в 1994 году. Программа имеет несколько вариантов исполнения, но в этой книге будет рассматриваться только вариант, предназначенный для работы совместно с модемом типа AN93. Вместо модема AN93 может с успехом использоваться описанный в главе 3 данной книги модем типа MODEM22. На прилагаемой к книге дискете имеется документация на русском языке к этой программе и некоторые необходимые для работы на русском языке файлы. Непосредственно саму программу следует скачать с Web-страницы Тома Сайлера, которая находится по адресу <http://www.baycom.org/~tom/ham/ham.html>, или взять ее из другого, более удобного для вас источника. Взятую в Интернете программу следует распаковать в предназначенный для этой программы подкаталог, после чего скопировать туда файлы с прилагаемой дискеты.

Программа Terman93 предназначена для работы в режимах RTTY, AMTOR и PACTOR. В режиме RTTY качество приема сигналов неудовлетворительное, в режимах AMTOR и PACTOR все в норме.

В данном разделе будет рассматриваться только работа программы в режиме AMTOR.

Как подключить программу

Программа может работать на компьютерах с процессором 286 и выше с использованием модема типа MODEM22, описание конструкции которого на-

ходится в *главе 3*. Подключение модема к порту RS-232-C компьютера выполняется по следующей схеме, приведенной в табл. 2.12.

Таблица 2.12. Схема подключения

Название линии	Разъем DB25	Разъем DB9	Выполняемые действия
RTS	4	7	Включение прием/передача
DTR	20	4	Модуляция сигнала (передача)
DCD	8	1	Принимаемый сигнал (прием)
SG	7	5	Сигнальное заземление

Конфигурационный файл

Конфигурация находится в файле allmode.ini. Этот файл — обычный текстовый ASCII-файл, разделенный на секции. Названия секций заключены в квадратные скобки. Редактировать файл нужно простым редактором в системе MS-DOS, соблюдая указанные ниже следующие требования при выборе величин.

Если назначаемая величина — логическая (boolean), то это может быть записано как 0, F, 1 или T. Целые числа могут быть или десятичные, или шестнадцатеричные (hexadecimal), при этом шестнадцатеричные числа должны оканчиваться на "h".

□ Секция [AN93]

- ConvAdjust=23

В версии AN93 эта строка управляет выбором времени часов компьютера. Этот параметр используется, чтобы отрегулировать часы компьютера в соответствии с быстродействием процессора, установленного на компьютере. Величину следует уменьшать, если часы идут с замедлением. Признак замедления — величина dt в окне статуса в этом случае главным образом отрицательная во время связи в режиме AMTOR SLAVE или PACTOR SLAVE.

- ComRTSInv=F

Линия RTS COM-порта служит как линия РТТ для переключения прием/передача. Если РТТ-логика инвертирована, вы можете исправить это, установив вместо F значение T или 1.

- ComDTRInv=F

Линия DTR COM-порта работает при передаче (TxD) и подает сигналы для управления частотой тонального генератора. Если произведен-

ные тоны должны быть инвертированы, вы можете добиться этого, установив ComDTRInv=T.

- ComDCDInv=F

Вход данных на компьютер идет через DCD-линию. Если DCD-логика обратная, вы можете исправить это, установив ComDCDInv=T.

- ComAddr=3F8H

Адрес COM-порта должен быть 3F8H для COM1, 2F8H для COM2, 3E8H для COM3, и 2E8H для COM4.

- LptAddr=0H

Это — адрес LPT-порта. LPT1 имеет адрес 378H, LPT2 — 278H. Величина 0 означает, что LPT не используется.

□ Секция [AMTOR]

- RxAfterTx=20

Эта величина определяет, сколько миллисекунд программное обеспечение должно ждать перед стартом в режиме SLAVE. Для работы с DX установите этот параметр равным TxDelay. Если вы предпочитаете ближние QSO, установите этот параметр равным приблизительно 50.

- TxDelay=20

Это — задержка вашего приемопередатчика при включении в миллисекундах.

- SelCall=NONE

Это — ваш SelCall. Он состоит ровно из 4 символов. Обычно это один или два первых и два или три последних символа вашего позывного. Если в позывном только три символа, вы можете использовать первый символ дважды. Например, HB9JNX — HJNX, HB9ZZ — HBZZ, ZS5S — ZZSS.

- CrLf=T

Должен ли выдаваться символ LF (Line Feed — перевод строки) после символа CR (Carriage Return — перевод каретки).

- CapsOnly=F

Нужно ли позволить строчные символы.

- Alphabet=0

Определяет, какой алфавит должен использоваться. В настоящее время эти алфавиты определены следующими цифрами:

- ◇ 0 — американский национальный алфавит ИТА 2;
- ◇ 1 — национальный алфавит Великобритании ИТА 2;
- ◇ 2 — немецкий национальный алфавит ИТА 2.

- TxReverse=F

Должен ли TX быть инвертирован.

- Reverse=F

Должны ли mark и space поменяться местами.

□ Секция [AMTOR CALLS]

- Call##=CALL

Вы можете определить список AMTOR SelCalls, которые вы используете часто. Позывные SelCalls, введенный здесь, появляются в меню Запроса AMTOR, где вы их можете выбрать.

являются порядковыми числами, начиная с 00.

□ Секция [FILES]

- Desc##=Title
- Name##=Filename

Это — список текстовых файлов, в которых вы нуждаетесь часто. Эти файлы появляются в меню передачи.

являются порядковыми числами, начиная с 00.

□ Секция [CWID]

- Speed=180

Это — скорость передачи телеграфом (CW) позывного вашей станции. Должна устанавливаться высокая величина (например, от 180 до 240), чтобы передавать телеграфом позывной сразу же после включения на передачу.

- Interval=600

Это — интервал в секундах CW-передач в течение ARQ QSO. Установите эту величину как можно больше.

Величина 0 запрещает CW-передачу.

- Call=MYOCALL

Текст, который должен быть послан как передача CWID. Записать 0, если вы не хотите передачи CWID.

□ Секция [Windows]

- ScrollBackLines=400

Позволяет вам определять число строк в буфере прокрутки текста.

Могут быть величины от 50 до 400, или 0, если вы не хотите использовать буфер прокрутки.

- NoStatus=F

Позволяет вам скрывать статус и окно монитора.

□ Секция [Colours]

Общий цветовой формат: `xyH`, где

- `x` — цвет фона;
- `y` — цвет переднего плана.

Цвета фона (цвета переднего плана):

◇ 0 — черный;	◇ 8 — темно-серый;
◇ 1 — синий;	◇ 9 — светло-синий;
◇ 2 — зеленый;	◇ A — светло-зеленый;
◇ 3 — голубой;	◇ B — ярко-голубой;
◇ 4 — красный;	◇ C — ярко-красный;
◇ 5 — сиреневый;	◇ D — ярко-сиреневый;
◇ 6 — коричневый;	◇ E — желтый;
◇ 7 — светло-серый;	◇ F — ярко-белый.

- `Cursor=xyH`

Устанавливает цвет курсора.

- `Frame=xyH`

Устанавливает цвет рамок между окнами.

- `Monitor=xyH`

Устанавливает цвет монитора и окна статуса.

- `Tx=xyH`

Устанавливает цвет окна передачи.

- `RxMessages=xyH`

Устанавливает цвета программных сообщений, появляющихся в окне приема.

- `RxSent=xyH`

Устанавливает цвет переданного текста.

- `RxReceived=xyH`

Устанавливает цвет полученного текста.

Наладка часов компьютера

Сигнал с выхода (вывод 20 при DB25) подать на частотомер, способный измерять тысячные доли герца. При этом программа должна находиться в режиме `AdjustClock`. В этом режиме компьютер должен генерировать сигнал частотой ровно 400 Гц. Изменять частоту генерации следует величиной зна-

чения `AdjustClock`. Начальное значение, записанное в файле `allmode.ini`, составляет `AdjustClock=23`. Следует изменять эту величину так, чтобы показания частотомера были не хуже $400 \pm 0,3$ Гц (± 30 ppm).

Вы можете также проверить чистоту генерируемого компьютером тона. Для этого можно подключить громкоговоритель через последовательный резистор 1 кОм к выводу 20 СОМ-порта компьютера (для разъема DB25), второй вывод громкоговорителя подключить к земле. Тон, который вы услышите, будет не очень чист. Это является нормальным. Но если степень загрязнения высока, работа системы будет плохая! Это может зависеть от работы какой-то другой TSR (резидентной в памяти) программы, прерывания которой вызывают помехи. Удалите из памяти все резидентные программы, насколько это возможно!

Если вы не имеете такого точного частотомера, вы можете регулировать часы во время работы в эфире. Попросите друга-радиолюбителя вызвать вас в режиме AMTOR или PACTOR и следите во время установления связи за величиной dt в окне статуса и появлением символов **o** и **n** в окне монитора. Эти символы и величина dt показывают количество случаев подгонки частоты программой в автоматическом режиме. В лучшем случае величина $dt = 0$ — самый приемлемый вариант.

Изменяя величину `AdjustClock` добейтесь того, чтобы $dt = 0$, а символы **o** и **n** в окне монитора появлялись очень редко.

Работа с программой

Не используйте приемник с узкополосным фильтром, особенно с DSP-версией. Эти фильтры обычно вызывают очень большое искажение сигналов, особенно на краях полосы пропускания фильтра.

Не используйте фильтры с полосой пропускания меньше 500 Гц. Даже диапазон 500 Гц может оказаться слишком узким, так что вы должны попробовать начать работу совсем без фильтра.

(Помните: полоса сигналов при приеме 200 Бод PACTOR-сигнала — 400 Гц). Подрегулируйте центральную частоту вашего конвертера к центральной частоте фильтра приемника.

Управление программой выполняется через меню.

Экран разделен на 3 области:

- ☐ Окно **Rx** — самая большая область. Здесь отображается белыми буквами все то, что вы получили от партнера по QSO, и синими буквами все то, что вы напечатали и что уже передано.
- ☐ Окно **Tx** — составляет всего несколько строк, в этом окне показывается все, что вы в настоящее время печатаете.

Обратите внимание, что напечатанное слово не будет послано до тех пор, пока вы не напечатаете небуквенный символ или не нажмете клавишу <Enter>.

Такое положение не позволяет удалить из слова неверно напечатанный символ в AMTOR и RTTY.

(В PACTOR вы можете исправлять столько, сколько хотите, нажимая клавишу <Backspace>, сигнал этой клавиши не передается по эфиру.)

- Окно **Монитор** показывает вам все, что происходит в настоящее время в эфире. Вы должны знать основы протокола этих видов связи, чтобы понимать то, что все эти символы означают. То же самое относится и к окну состояния связи, расположенному справа внизу.

Меню вызывается клавишей <F2>. Вы можете выбирать любой пункт при помощи клавиш-стрелок, после чего нужно нажать клавишу <Enter>.

Некоторые пункты меню имеют подменю, которые обозначены стрелкой, указывающей направо. При выборе такого пункта появляется окно с пунктами подменю.

Я не буду объяснять каждый пункт меню, т. к. все действия предельно простые. Существует строка подсказок, где указано большинство команд.

Одно примечание: сразу же после запуска программы ничто не включается. Если вы хотите слушать QSO в AMTOR или PACTOR, вы должны нажать клавишу <F4>.

RTTY и AMTOR обычно имеют сдвиг 170 Гц, PACTOR имеет сдвиг 200 Гц, однако AMTOR и RTTY большинства западных станций, которые работают с контроллером PTC, имеют сдвиг также 200 Гц. Так что 170 Гц и 200 Гц в любительском эфире перемешаны.

В течение QSO или для быстрой настройки вы должны постоянно использовать индикатор настройки в верхней части экрана.

Как передавать файлы

Дать команду на передачу файла можно через меню. Файлы, которые используются часто, могут быть введены в секцию [Files] файла allmode.ini. Эти файлы появляются впоследствии в меню. Кроме того, они могут быть переданы путем нажатия на "горячие" клавиши. При нажатии на клавиши <Shift>+<F1>—<Shift>+<F10> передаются файлы от 00 до 09, при нажатии на клавиши <Ctrl>+<F1>—<Ctrl>+<F10> посылаются на передачу файлы от 10 до 19, а клавиши <Alt>+<F1>—<Alt>+<F10> посылают на передачу файлы от 20 до 29.

Эти файлы могут содержать следующие специальные символы:

- \ — послать одну наклонную влево черту;
- \q — закончить передачу (и файл) в этом пункте;

- \t — вставить время;
- \d — вставить дату;
- \w — вставить день недели.

Программа TERNAN93 очень удобна и надежна в работе. Автор этой программы является известным программистом, разработчиком многих драйверов к различным видам цифровой радиосвязи. Настоятельно рекомендую приобрести и освоить эту программу.

Цифровой вид связи РАСТОР

Принципы построения сигнала в РАСТОР

Цифровой вид связи РАСТОР появился в начале 1990-х годов. В странах Западной Европы и США уже много лет существует большая радиолобительская сеть, основу которой составляют BBS, работающие в этом режиме.

Как показали многочисленные опыты, РАСТОР лучше других цифровых видов связи работает в условиях помех, присущих низкочастотным коротковолновым диапазонам.

В нашей стране любительская радиосеть еще только начинает развиваться, и при этом основу этой сети составляют BBS-станции, работающие в режиме Packet Radio. Режим РАСТОР еще не нашел широкого распространения.

Основной причиной такого положения, на мой взгляд, являются большая сложность создания чисто программных реализаций этого вида связи, в отличие, например, от RTTY, а также отсутствие в продаже дешевых аппаратных модемов и контроллеров.

Главной особенностью режима РАСТОР является необходимость организации постоянно действующих прерываний через интервалы времени ровно в одну миллисекунду и корректировка (при необходимости) длительности этого интервала программным путем, т. е. программа должна работать в режиме реального времени.

Западные фирмы разработали специальные аппаратные контроллеры — TNC — и много различных компьютерных программ для работы с этими контроллерами. Контроллеры для РАСТОР западного производства слишком дорогие, и это второй фактор, сдерживающий применение этого вида цифровой связи в нашей стране.

Каждый передаваемый в режиме РАСТОР символ кодируется восьмибитовым кодом и состоит из восьми токовых и бестоковых посылок, длительность одной посылки при скорости 200 Бод составляет 5 мс, а при скорости 100 Бод — 10 мс, при этом каждая посылка передается серией импульсов длительностью в 1 мс.

Коды символов PACTOR полностью соответствуют стандартной кодировке IBM, используемой в MS-DOS. Поэтому никакой дополнительной перекодировки символов не требуется.

Разница между частотами сигналов "mark" (высокая звуковая частота, например, 1200 Гц) и "space" (низкая звуковая частота, например, 1000 Гц) должна составлять 200 Гц. Кроме служебных и информационных символов, кодированных восьмибитовым кодом, имеются также четыре подтверждающих символа (символы синхронизации), каждый из которых состоит из 12 битов.

PACTOR дает возможность проводить радиосвязи в трех основных режимах.

- ❑ **Режим FEC (Forward Error Correction)** позволяет передавать в эфир файлы с информацией, доступной любой из находящихся в данный момент на частоте радиостанций. В радиолюбительской практике таким файлом может быть файл с текстом общего вызова. Свободно принимать эти передачи может любая из находящихся на этой частоте PACTOR-радиостанций, если эта радиостанция в данный момент находится в режиме LISTEN.
- ❑ **Режим LISTEN (MONITOR)** позволяет четко принимать на экране дисплея своего компьютера сигналы другой PACTOR-станции, которая ведет передачу в режиме FEC. Но если эта же станция настроится на частоту двух других радиостанций, которые обмениваются между собой информацией в режиме ARQ (MARQ), то она сможет получать на экране своего дисплея только отдельные фрагменты этого обмена, т. е. сможет принимать информацию только в очень непродолжительные моменты случайной синхронизации.
- ❑ **Режим ARQ (Auto ReQuest)** предполагает работу двух радиостанций по обмену информации в полной зависимости друг от друга. Эта зависимость устанавливается специальными сигналами — синхросимволами, при этом ведущая передачу станция полностью берет на себя управление приемной радиостанцией, т. е. устанавливается режим синхронной работы двух радиостанций. На деле это выглядит следующим образом. Вы настраиваете свою PACTOR-станцию точно на частоту своего корреспондента, вводите в программу позывной этого корреспондента и включаете режим ARQ. Ваша станция сразу же начинает выдавать пакеты длительностью примерно в 1 секунду, в которых содержатся команды вызова определенного корреспондента. Каждый такой пакет отделяется один от другого промежутком в несколько миллисекунд. Как только станция корреспондента начнет принимать и правильно декодировать сигналы вашей радиостанции, она в промежутки между пакетами вашей радиостанции начнет передавать ответные сигналы длительностью в несколько миллисекунд. Ваша станция должна принять ответы корреспондента, правильно их декодировать и дать положительный ответ корреспонденту о готовности установить связь. Если и одна, и другая станции четко принимают

сигналы друг друга и правильно их декодируют, возникает достаточно прочная связь между этими радиостанциями, которая называется синхронным режимом работы. При этом любая третья радиостанция из вашего обмена будет принимать и отображать на экране своего дисплея только отдельные несвязные фрагменты в моменты случайного совпадения синхронизации.

В радиолюбительской практике возможен такой случай, когда в ответ на общий вызов вашей радиостанции в режиме FEC вас начинает вызывать какая-то станция в режиме ARQ, между двумя вашими станциями устанавливается нормальный режим синхронизации и нормальный обмен информацией. Если вдруг резко ухудшается прохождение, то работа в режиме ARQ становится невозможной, но все еще можно будет попытаться продолжить работу в режиме FEC с частичной потерей информации.

Работа в режиме FEC всегда осуществляется со скоростью 100 Бод при длине пакета передаваемых информационных символов от 6 до 9. Начинает работать на передачу при длине пакета 9 символов, но в случае плохого прохождения количество символов в пакете может уменьшиться до 6. Кроме того, в зависимости от прохождения каждый из передаваемых пакетов может транслироваться несколько раз подряд, в зависимости от прохождения число повторов может быть 2 или 3. Одни программы требуют установки числа повторов в конфигурационном файле, другие выбирают это число сами.

Режим LISTEN (в некоторых программах — MONITOR) позволяет принимать и отображать на дисплее своего компьютера всю информацию, которую на данной частоте другая PACTOR-станция передает в режиме FEC. Если ваша станция находится в режиме LISTEN и начинает принимать сигналы другой станции, которая вызывает вас в режиме ARQ, то она тут же начинает отвечать вызываемой радиостанции короткими подтверждающими импульсами. При хорошем приеме и правильном декодировании сигналов обеими станциями, они автоматически входят в синхронный режим обмена информацией — режим ARQ.

Следует знать, что режим FEC на одной станции и режим LISTEN на другой позволяют проводить QSO по типу телетайпной связи в асинхронном режиме.

Режим ARQ начинается с вызова определенной станции. Сначала следует ввести позывной корреспондента, настроиться на его частоту и только после этого включать режим ARQ. Ваша станция начинает тут же выдавать пакеты длительностью чуть меньше 1 секунды с интервалами между пакетами длительностью в несколько миллисекунд. Если вызываемая станция принимает пакеты вашей станции и правильно их декодирует, то она начинает во время интервалов между пакетами вызываемой станции вставлять свои ответные подтверждающие импульсы. Если ваша станция правильно примет и декодирует ответные подтверждающие импульсы, то между двумя станциями устанавливается надежная связь в синхронном режиме, во время которой

синхронный режим постоянно поддерживается взаимным обменом подтверждающими и синхронизирующими символами. В этом режиме передающая станция является главной (MASTER) и заставляет принимающую станцию (SLAVE) четко выполнять все необходимые команды.

В режиме ARQ передача информации может происходить или при скорости 200 Бод, или при 100 Бод. Других скоростей нет. Начинают работу со скорости 200 Бод и при ухудшении прохождения скорость уменьшается до 100 Бод. Переменной величиной является также и число информационных символов в пакете. Наибольшее число символов — длина пакета — равно 20 и может быть установлено программой только при скорости 200 Бод. При скорости 100 Бод наибольшая длина пакета составляет 9 символов. При ухудшении прохождения эти цифры могут уменьшаться.

В начале каждого пакета находится цифра, означающая длину пакета, так что принимающая сторона заранее знает, какой длины пакет ей предстоит принять.

Программа TERNAN93 работает в режиме PACTOR

О программе TERNAN93

Программа TERNAN93 разработана известным программистом из Швейцарии Томом Сайлером (Tom Sailer, HB9JNX) в 1994 году. Программа имеет несколько вариантов исполнения, но в этой книге будет рассматриваться только вариант, предназначенный для работы совместно с модемом типа AN93. Вместо модема AN93 может с успехом использоваться описанный в *главе 3* данной книги модем типа MODEM22. На прилагаемой к книге дискете имеется документация на русском языке к этой программе и некоторые необходимые для работы на русском языке файлы. Непосредственно саму программу следует скачать с Web-страницы Тома Сайлера, которая находится по адресу <http://www.baycom.org/~tom/ham/ham.html>, или взять ее из другого, более удобного для вас источника. Взятую в Интернет программу следует распаковать в предназначенный для этой программы подкаталог, после чего скопировать туда файлы с прилагаемой дискеты.

Программа TERNAN93 предназначена для работы в режимах RTTY, AMTOR и PACTOR. В режиме RTTY качество приема сигналов неудовлетворительное, в режимах AMTOR и PACTOR все в норме.

В данном разделе будет рассматриваться только работа программы в режиме PACTOR.

Как подключить программу

Программа может работать на компьютерах с процессором 286 и выше с использованием модема типа MODEM22, описание конструкции которого на-

ходится в *главе 3*. Подключение модема к порту RS-232-C компьютера выполняется по следующей схеме, приведенной в табл. 2.13.

Таблица 2.13. Схема подключения

Название линии	Разъем DB25	Разъем DB9	Выполняемые действия
RTS	4	7	Включение прием/передача
DTR	20	4	Модуляция сигнала (передача)
DCD	8	1	Принимаемый сигнал (прием)
SG	7	5	Сигнальное заземление

Конфигурационный файл

Конфигурация выполнена в файле allmode.ini. Этот файл — обычный текстовый ASCII-файл, разделенный на секции. Названия секций заключены в квадратные скобки. Редактировать файл нужно простым редактором в системе MS-DOS, соблюдая указанные ниже требования при выборе величин.

В каждой строке нужно просмотреть записанную после знака равенства величину, и, в случае необходимости, отредактировать ее. Если назначаемая величина — логическая (boolean), то в таком случае может быть записано 0, F, 1 или T. Целые числа могут быть или десятичные или шестнадцатеричные (hexadecimal), при этом шестнадцатеричные числа должны оканчиваться на "h".

□ Секция [AN93]

- ConvAdjust=23

В версии AN93 эта строка управляет выбором времени часов компьютера. Этот параметр используется, чтобы отрегулировать часы компьютера в соответствии с быстродействием процессора, установленного на компьютере. Величину следует уменьшать, если часы идут с замедлением. Признак замедления — величина dt в окне статуса в этом случае главным образом отрицательная во время связи в режиме AMTOR SLAVE или PACTOR SLAVE.

- ComRTSInv=F

Линия RTS COM-порта служит как РТТ линия переключения прием/передача. Если РТТ-логика инвертирована, вы можете исправить это, установив вместо F значение T или 1.

- ComDTRInv=F

Линия DTR COM-порта работает при передаче (TxD) и подает сигналы для управления частотой тонального генератора. Если произведен-

ные тоны должны быть инвертированы, вы можете добиться этого, установив `ComDTRInv=T`.

- `ComDCDInv=F`

Вход данных на компьютер идет через DCD-линию.

Если DCD-логика обратная, вы можете исправить это, установив `ComDCDInv=T`.

- `ComAddr=3F8H`

Адрес COM-порта должен быть 3F8H для COM1, 2F8H — для COM2, 3E8H — для COM3, и 2E8H для — COM4.

- `LptAddr=0H`

Это — адрес LPT-порта. LPT1 имеет адрес 378H, LPT2 — 278H. Величина 0 означает, что LPT не используется.

□ Секция [PACTOR]

- `RxAfterTx=50`

Эта величина определяет, сколько миллисекунд программное обеспечение должно ждать начала передачи в режиме SLAVE.

Для DX установите этот параметр равным `TxDelay`. Если вы хотите проводить ближние связи, установите величину приблизительно 50. Вы можете задать весьма большую величину, если PACTOR-связь должна идти по длинному пути прохождения (LongPath mode) для DX связей.

- `TxDelay=50`

Это — число миллисекунд задержки вашего приемопередатчика для переключения с приема на передачу и наоборот.

- `MyCall=MY0CALL`

Ваш позывной (до 8 символов).

- `StreamFEC=T`

Разрешает поток в режиме FEC. Если поток запрещен, используется нормальный выбор времени для FEC, иначе — сжатый выбор времени.

- `FECRetrans=2`

Определяет, сколько раз должен повторяться пакет при передаче в FEC-режиме. Обычно 2, но если прохождение очень хорошее, вы можете установить 1.

□ Секция [LISTEN]

- `CheapPactor=F`

Если у вас медленный компьютер, то могут быть проблемы при переходе в режим приема, о чем свидетельствует сообщение `LOST INTS`. Тогда вы можете попробовать установить этот параметр в T.

□ Секция [PACTOR_CALLS]

- Call##=CA0LL

Вы можете определить список позывных, которые используете часто. Позывные, введенные здесь, появятся в меню **Call**, где их можно брать.

являются порядковыми числами, начиная с 00.

□ Секция [FILES]

- Desc##=Title
- Name##=Filename

Это — список текстовых файлов, в которых вы часто нуждаетесь. Эти файлы появляются в меню передачи.

являются порядковыми числами, начиная с 00.

□ Секция [CWID]

- Speed=180

Это — скорость передачи телеграфом (CW) позывного вашей станции. Должна устанавливаться высокая величина (например, от 180 до 240), чтобы передавать телеграфом позывной сразу же после включения на передачу.

- Interval=600

Это — интервал в секундах CW-передач в течение ARQ QSO. Установите эту величину как можно больше.

Величина 0 запрещает CW-передачу.

- Call=MYOCALL

Текст, который должен быть послан как передача CWID. Записать 0, если вы не хотите передачи CWID.

□ Секция [Windows]

- ScrollBackLines=400

Позволяет вам определять число строк в буфере прокрутки текста.

Могут быть величины от 50 до 400, или 0, если вы не хотите использовать буфер прокрутки.

- NoStatus=F

Позволяет вам скрывать статус и окно монитора.

□ Секция [Colours]

Общий цветовой формат: xуN, где:

- x — цвет фона;
- y — цвет переднего плана.

Цвета фона (цвета переднего плана):

- | | |
|---------------------|-----------------------|
| ◇ 0 — черный; | ◇ 8 — темно-серый; |
| ◇ 1 — синий; | ◇ 9 — светло-синий; |
| ◇ 2 — зеленый; | ◇ A — светло-зеленый; |
| ◇ 3 — голубой; | ◇ B — ярко-голубой; |
| ◇ 4 — красный; | ◇ C — ярко-красный; |
| ◇ 5 — сиреневый; | ◇ D — ярко-сиреневый; |
| ◇ 6 — коричневый; | ◇ E — желтый; |
| ◇ 7 — светло-серый; | ◇ F — ярко-белый. |

- `Cursor=xyH`

Устанавливает цвет курсора.

- `Frame=xyH`

Устанавливает цвет рамок между окнами.

- `Monitor=xyH`

Устанавливает цвет монитора и окна статуса.

- `Tx=xyH`

Устанавливает цвет окна передачи.

- `RxMessages=xyH`

Устанавливает цвета программных сообщений, появляющихся в окне приема.

- `RxSent=xyH`

Устанавливает цвет переданного текста.

- `RxReceived=xyH`

Устанавливает цвет полученного текста.

Наладка часов компьютера

Сигнал с выхода (вывод 20 при DB25) подать на частотомер, способный измерять тысячные доли герца. При этом программа должна находиться в режиме `AdjustClock`. В этом режиме компьютер должен генерировать сигнал частотой ровно 400 Гц. Изменять частоту генерации следует величиной значения `AdjustClock`. Начальное значение, записанное в файле `allmode.ini`, составляет `AdjustClock=23`. Следует изменять эту величину так, чтобы показания частотомера были не хуже $400 \pm 0,3$ Гц (± 30 ppm).

Вы можете также проверить чистоту генерируемого компьютером тона. Для этого можно подключить громкоговоритель через последовательный резистор 1 кОм к выводу 20 СОМ-порта компьютера (для разъема DB25), второй

вывод громкоговорителя подключить к земле. Тон, который вы услышите, будет не очень чист. Это является нормальным. Но если степень загрязнения высока, работа системы будет плохая! Это может зависеть от какой-то другой TSR (резидентной в памяти) программы, прерывания которой вызывают помехи. Удалите из памяти все резидентные программы, насколько это возможно!

Если вы не имеете такого точного частотомера, вы можете регулировать часы во время работы в эфире. Попросите друга-радиолюбителя вызвать вас в режиме AMTOR или RACTOR и следите во время установления связи за величиной dt в окне статуса и появлением символов **o** и **n** в окне монитора. Эти символы и величина dt показывают количество случаев подгонки частоты программой в автоматическом режиме. В лучшем случае величина $dt = 0$ — самый приемлемый вариант.

Изменяя величину AdjustClock, добейтесь того, чтобы $dt = 0$, а символы **o** и **n** в окне монитора появлялись очень редко.

Работа с программой

Не используйте приемник с узкополосным фильтром, особенно с DSP-версией. Эти фильтры обычно оказывают очень большое искажение сигналов, особенно на краях полосы пропускания фильтра.

Не используйте фильтры с полосой пропускания меньше 500 Гц. Даже 500 Гц может быть слишком узким диапазоном, так что вы должны попробовать начинать работу совсем без фильтра.

(Помните: полоса сигналов при приеме 200 Бод RACTOR-сигнала — 400 Гц. Подрегулируйте центральную частоту вашего конвертера к центральной частоте фильтра приемника.

Управление программой выполняется через меню.

Экран разделен на 3 основных области:

- ☐ Окно **Rx** — самая большая область. Здесь отображается белыми буквами все то, что вы получили от партнера по QSO, и синими буквами все то, что вы напечатали и что уже передано.
- ☐ Окно **Tx** — составляет всего несколько строк, в этом окне показывается все, что вы в настоящее время печатаете.

Обратите внимание, что напечатанное слово не будет послано до тех пор, пока вы не напечатаете небуквенный символ или не нажмете клавишу <Enter>.

Такое положение не позволяет удалить из слова неверно напечатанный символ в AMTOR и RTTY.

(В RACTOR, вы можете исправлять столько, сколько хотите, нажимая клавишу <Backspace>, сигнал этой клавиши не передается по эфиру.)

□ Окно **Монитор** показывает вам все, что происходит в настоящее время в эфире. Вы должны знать основы протокола этих видов связи, чтобы понимать то, что все эти символы означают. То же самое относится и к окну состояния связи, расположенному справа внизу.

Меню вызывается клавишей <F2>. Вы можете выбирать любой пункт при помощи клавиш-стрелок, после чего нужно нажать клавишу <Enter>.

Некоторые пункты меню имеют подменю, которые обозначены стрелкой, указывающей направо. При выборе такого пункта появляется окно с пунктами подменю.

Я не буду объяснять каждый пункт меню, т. к. все действия предельно простые. Существует строка подсказок, где указано большинство команд.

Одно примечание: сразу же после запуска программы ничего не включается. Если вы хотите слушать QSO в AMTOR или PACTOR, вы должны будете нажать клавишу <F4>.

RTTY и AMTOR обычно имеют сдвиг рабочих частот 170 Гц, PACTOR имеет сдвиг 200 Гц, однако AMTOR и RTTY большинства западных станций, которые работают с контроллером PTC, имеют сдвиг также 200 Гц. Так что 170 Гц и 200 Гц в любительском эфире перемешаны. Некоторые радиолюбители в своих модемах, чтобы без перестройки работать и в режиме RTTY, и в режиме PACTOR, устанавливают величину сдвига 185 Гц, как говорится, ни вашим, ни нашим.

В течение QSO или для быстрой настройки вы должны постоянно использовать индикатор настройки в верхней части экрана.

Некоторые особенности PACTOR

В режиме PACTOR имеется несколько необычных вещей.

Так, не имеется никакого автоматического алгоритма для увеличения скорости. То есть вы должны включать вручную повышение скорости до 200 Бод. Уменьшение скорости происходит автоматически, если пакет не может быть правильно получен в течение 4 циклов. Обратите внимание, что вы можете запрашивать повышение скорости только в том случае, если вы — станция получения информации (IRS).

Причина этого в том, что программа в полностью автоматическом режиме будет затрачивать слишком много времени на безуспешное переключение скорости то вверх, то вниз. Наблюдая информацию в окне статуса, можно всегда сделать вывод о необходимости увеличения скорости и выполнить это вручную. Это будет легко делать после приобретения некоторого опыта.

Иногда PACTOR не может разъединиться, т. е. если вы хотите разъединиться с корреспондентом, но отсутствует подтверждение от него, то пребывание в состоянии связи будет продолжаться до получения подтверждения. Это будет устранено в новой версии программы. Вы можете прекратить передачу пакетов вручную, нажав на клавишу <F4>.

Как передавать файлы

Дать команду на передачу файла можно через меню. Файлы, которые используются часто, могут быть введены в секцию [Files] файла allmode.ini. Эти файлы появляются впоследствии в меню. Кроме того, они могут быть переданы путем нажатия на "горячие" клавиши. При нажатии на клавиши <Shift>+<F1>—<Shift>+<F10> передаются файлы от 00 до 09, при нажатии на клавиши <Ctrl>+<F1>—<Ctrl>+<F10> посылаются на передачу файлы от 10 до 19, а клавиши <Alt>+<F1>—<Alt>+<F10> посылают на передачу файлы от 20 до 29.

Эти файлы могут содержать следующие специальные символы:

- ☐ \ — послать одну наклонную влево черту;
- ☐ \q — закончить передачу (и файл) в этом пункте;
- ☐ \t — вставить время;
- ☐ \d — вставить дату;
- ☐ \w — вставить день недели.

Как осваивать работу в PACTOR

Основной особенностью работы в PACTOR и AMTOR является необходимость исключительно быстрого, в пределах единиц миллисекунд, переключения радиостанции с приема на передачу и наоборот. Никакие электромагнитные реле здесь не обеспечат необходимого быстродействия. Нужно применять электронные (транзисторные) переключатели. Подобная схема работает в трансивере RA3AO.

Для начала нужно установить транзисторный переключатель прием/передача на свой трансивер. Затем сделать необходимые переводы документации к программе Terman93 и выписать на отдельную шпаргалку все команды, задействованные в этой программе. Это очень удобно при освоении — иметь постоянно перед глазами перечень используемых в программе команд. Программа очень хороша тем, что имеет слева внизу окно монитора, в котором отображается вся принимаемая информация. Модем нужно использовать универсальный, чтобы обеспечить работу во всех видах цифровой связи. Это модем типа MODEM22, описание которого размещено на прилагаемой к книге дискете. Необходимо только на линии приема сигналов между модемом и компьютером поставить переключатель. Дело в том, что для приема пакета на COM-порту компьютера используется вывод CTS (номер 5 для DB25), а для приема сигналов программы Terman93 необходимо использовать вывод DCD (номер 8 для DB25).

Прием сигналов AMTOR и PACTOR в окне монитора никаких проблем вызывать не должен. Введите свой позывной и все другие необходимые данные в файл allmode.ini. Выбирайте самую громкую BBS-станцию, настраивайтесь

на ее частоту и начинайте вызывать. После некоторых неудачных попыток начнете получать ответные сигналы, это значит, что между вашими станциями установилась нормальная синхронизация и вы в окне приема начнете получать пакеты с информацией, из которых будут складываться текстовые строки. Но после приема 1—2 строк BBS перестанет принимать сигналы вашей радиостанции и связь прекратится. Это говорит о том, что нужно проводить коррекцию часов программы. Для целей проведения коррекции в инициализационном файле программы `allmode.ini`, вернее в разделе `[FACTOR]` этого файла, находится строка `ConvAdjust=23`. Можно увеличить цифру 23 на 4 единицы и снова вызывать эту же станцию. Результаты могут оказаться тоже плохими. Установите `ConvAdjust=30`. Пробуйте вызывать BBS снова. Повторяйте эти процедуры до тех пор, пока связь с BBS не станет непрерывной.

Все зависит от быстродействия процессора на вашем компьютере. Мне известны случаи, когда для Pentium 100 использовалась величина `ConvAdjust=32`, а для Celeron 330 — величина `ConvAdjust=49`.

Таким путем вы становитесь полноправным участником работы радиолюбительской сети, можете отправлять и получать через свою "home BBS" любую корреспонденцию для любого корреспондента. Не забывайте только уничтожать все адресованные вам и прочитанные сообщения.

Packet Radio — один из главных видов цифровой связи

Основные принципы построения сигнала в Packet Radio

Пакетная радиосвязь Packet Radio — это цифровая документальная безошибочная связь, осуществляемая с помощью компьютеров, подключаемых к радиостанции через пакетные контроллеры, их называют TNC — Terminal Node Controller. Такой контроллер представляет собой небольшую ЭВМ с процессором типа Z80 или аналогичным. Назначение контроллера:

- ☐ получать от приемника радиостанции сигналы, декодировать эти сигналы и передавать их на COM-порт компьютера;
- ☐ получать от компьютера предназначенный для передачи текст, разбивать его на пакеты нужной длины, кодировать символы пакета и передавать их на передатчик радиостанции.

Контроллеры TNC аппаратного исполнения являются связующим звеном между компьютером и радиостанцией. Как правило, это достаточно дорогие изделия, но качество приема и передачи с помощью этих аппаратов всегда хорошее.

Кроме того, существуют пакетные контроллеры, выполненные программно. В качестве примера назову такие программные контроллеры (их часто называют эмуляторами TNC): L2, TFPCX, TFKISS, AGWPE и многие другие. При использовании таких программных контроллеров в качестве связующего звена между компьютером и радиостанцией необходим аппаратный модем, который будет передавать сигналы от компьютера к радиостанции и наоборот. Описание некоторых конструкций таких модемов приведено в главе 3.

В основе пакетной связи находится понятие "кадр", зачастую также называемое "пакет".

Что такое "пакет"

Пакетная передача данных по каналу связи осуществляется небольшими блоками данных, называемых кадрами. Каждый кадр состоит из меньших по размеру групп, называемых полями. В табл. 2.14 приведена структура служебного кадра. Имеется несколько типов служебных кадров, но все они имеют одинаковую структуру. Служебные кадры, так называемые "супервизорные", обозначаются буквой S, служебные кадры "нenumерованные" обозначаются буквой U. Вся передаваемая информация идет через "информационные" кадры, обозначаемые буквами I или UI. Структура информационного кадра приведена в табл. 2.15.

Каждый кадр делится на отдельные области, называемые "полями". Размеры полей различны, задачи для каждого поля строго определены *протоколом*.

Следует обратить внимание на то, что первый посылаемый бит расположен слева.

Таблица 2.14. Служебный кадр (S, U)

Флаг	Адрес	Управление	Контрольная сумма (FCS)	Флаг
01111110	112 / 560 бит	8 бит	16 бит	01111110

Таблица 2.15. Информационный кадр (I)

Флаг	Адрес	Управление	PID	Информационное поле	Контрольная сумма (FCS)	Флаг
01111110	112 / 560 бит	8 бит	8 бит	N × 8 бит	16 бит	01111110

Каждое поле состоит из целого числа байтов и выполняет специальные функции, описанные ниже.

❑ Поле флага.

Поле флага имеет длину в один байт. Поскольку флаг используется для разграничивания кадров, он присутствует как в начале, так и в конце каждого кадра. Два кадра могут иметь один общий флаг, обозначающий конец первого кадра и начало следующего. Флаг состоит из нуля, за которым следуют шесть единиц и опять ноль, или 01111110 (7E в шестнадцатеричной системе исчисления). В результате вставки битов (см. ниже) такая последовательность не может появиться больше нигде внутри полного кадра.

❑ Поле адреса.

Поле адреса используется для идентификации как отправителя кадра, так и его получателя. Кроме того, поле адреса содержит информацию типа команда/ответ, а также средства, обеспечивающие работу ретранслятора 2-го уровня.

❑ Поле управления.

Поле управления используется для идентификации типа посылаемого кадра и для управления различными признаками соединения 2-го уровня. Его длина составляет один байт.

❑ Поле PID.

Поле идентификатора протокола (PID) присутствует только в информационных (I и UI) кадрах. Оно идентифицирует тип протокола 3-го уровня, если он используется.

❑ Информационное поле.

Информационное поле используется для переноса данных из одного конца канала связи в другой. Информационные поля разрешены только в трех связках кадров: I-кадр, UI-кадр, и FRMR-кадр. I-поле может иметь длину до 256 байтов и должно содержать целое число байтов. Эти ограничения должны применяться до вставки нулевых битов. Любая информация в I-поле будет посылаться по каналу связи прозрачно, за исключением вставки нулевого бита, необходимого, чтобы воспрепятствовать случайному появлению флагов в I-поле.

❑ Вставка битов.

Чтобы гарантировать, что последовательность битов флага случайно не появилась нигде более в кадре, посылающая станция должна контролировать последовательность битов на присутствие в ней группы из пяти и более подряд стоящих единичных битов. Каждый раз, когда посылаются пять подряд стоящих единичных битов, посылающая станция должна вставлять нулевой бит после пятого единичного бита. При получении

кадра, каждый раз, когда принимаются подряд пять единичных битов, нулевой бит, непосредственно следующий за этими пятью единичными битами, должен отбрасываться.

❑ Проверочная последовательность кадра.

Проверочная последовательность кадра (или FCS — контрольная сумма) — это шестнадцатибитное число, вычисляемое отправителем и получателем кадра. Проверочная последовательность используется, чтобы удостовериться в том, что кадр не был искажен средой, используемой для передачи кадра от отправителя к получателю. Она вычисляется в соответствии с рекомендациями ISO-3309 (HLDC).

❑ Порядок передачи битов.

За исключением поля FCS, все поля кадра AX.25 должны посылаться, начиная с младшего значащего бита. FCS посылается начиная со старшего значащего бита.

❑ Недействительные кадры.

Любой кадр, состоящий менее чем из 136 битов (включая открывающий и закрывающий флаги), не ограниченный открывающим и закрывающим флагами или не отвечающий требованию по числу байтов (целому числу байтов), должен рассматриваться в данном канальном уровне как недействительный кадр.

Поле адреса содержит адреса назначения и отправителя, т. е. позывные сигналы радиостанций в коде ASCII (КОИ-7), а также позывные станций-ретрансляторов, если таковые применяются.

Поле управления служит для определения типа кадра. Дело в том, что, кроме информационных кадров (I), предусмотрена передача и служебных кадров — так называемых супервизорных и нумерованных кадров, формат которых показан в табл. 2.14. Эти кадры необходимы для выполнения процедур протокола AX.25. Так, например, супервизорные кадры (S) служат для подтверждения приема неискаженных помехами кадров или для запроса повторной передачи искаженных кадров. Нумерованные кадры (U) служат для установления логического соединения и других случаев управления обменом в сети. Поле управления во всех типах кадров определяет тип кадра и функцию, которую должна выполнить приемная сторона при получении этого кадра.

Поле определения протокола (поле PID) служит для определения приемной стороной конкретной версии протокола, применяемого передающей стороной.

Это устраняет различные недоразумения, которые могут возникнуть в процессе развития и совершенствования протоколов любительской пакетной радиосвязи. В настоящее время в ходу у радиолюбителей по меньшей мере три версии протоколов канального уровня, и необходимо точно знать, какого из них придерживается корреспондент.

Длина информационного поля, т. е. поля, содержащего передаваемую информацию (пакет), ограничивается величиной 2048 битов (256 байтов). Важно, чтобы число битов в этом поле было кратно восьми (количеству полных символов в коде КОИ-7 с контрольным разрядом). Однако с увеличением длины кадра увеличивается время его передачи, что повышает вероятность поражения его помехой, а также увеличивает время ожидания передачи других абонентов, работающих в данной сети. Поэтому радиолюбители редко передают кадры с длиной информационной части более 1024 битов (128 байтов).

Контрольная сумма необходима для обнаружения ошибок в кадре при его приеме. При передаче вся битовая последовательность кадра подвергается подсчету в соответствии с определенным правилом. Результаты подсчета и представляют контрольную сумму. При приеме также выполняется подсчет, результаты которого сравниваются с принятой контрольной суммой. Если они не совпадают — в принятом кадре имеется ошибка и кадр необходимо повторить.

Процедура компоновки кадра при передаче и его распаковка до уровня пакета относятся ко второму уровню эталонной модели МОС — протоколу управления информационным каналом и обычно выполняются в пакетном адаптере аппаратно-программными средствами.

Благодаря этому осуществляется безошибочная связь между корреспондентами непосредственно или же через цифровой ретранслятор (Repeater).

Цифровые ретрансляторы — это специальные радиостанции, расположенные на возвышенных местах или высоких сооружениях. Работают круглосуточно в автоматическом режиме. Используют для связи ультракороткие волны. Эти радиостанции ведут прием цифровых (или аналоговых) сигналов, затем переносят принятые сигналы на другую частоту и передают их в эфир уже на новой частоте. Ретрансляторы могут устанавливаться на искусственных спутниках Земли. Заметим, что ретрансляторы можно объединить в цепочки. На каждом этапе происходит проверка кадра на ошибки при приеме и подтверждается безошибочный прием.

Благодаря адресной части кадра возможен обмен между двумя или группой корреспондентов в сетях пакетных радиостанций, т. е. когда на одной частоте работает большое число логически связанных или не связанных между собой корреспондентов. Понятие логической связи, или, как говорят специалисты, виртуального соединения, состоит в том, что пакеты между парой любых абонентов передаются в общем канале связи, используемом множеством других абонентов, и селективируются по адресам (в любительском варианте — позывным), имеющимся в заголовке кадров, только теми абонентами сети, с которыми установлено логическое соединение.

Для того чтобы пакеты, по возможности, не накладывались друг на друга во времени, существует дисциплина, или, как чаще говорят, протокол доступа в канал (на частоту) коллективного пользования. Любители применяют протокол множественного доступа с контролем занятости канала.

Как это осуществляется практически? Корреспондент перед передачей проверяет занятость канала и немедленно начинает передачу, если канал свободен. Если же канал занят, то передача кадра откладывается на некоторое время, устанавливаемое оператором перед вхождением в сеть (в зависимости от ее загрузки). По истечении этого времени снова проверяется занятость канала и процедура повторяется. Чем больше загружена данная частота, тем больше среднее время ожидания передачи кадра.

Любительская пакетная радиосвязь, кроме связи типа "корреспондент — корреспондент", обращения к PBBS, почтовым ящикам (малым по объему памяти PBBS) и маякам (Beacons — почтовым ящикам для срочных сообщений, периодически сообщаящим о наличии в них телеграмм), позволяет осуществлять передачу через станции-шлюзы (Gateways), принимающие пакеты на одной частоте и передающие их на другой (например, из одной КВ-сети в другую, с УКВ на КВ, с КВ через спутниковый УКВ-канал и т. д.). Причем, можно передавать не только смысловые и графические сообщения, но и программы для компьютеров. Для этого в каждом контроллере предусмотрен "прозрачный" режим передачи, при котором передача осуществляется не символами семизначного кода КОИ-7, как при обмене смысловой информацией, а непосредственно натуральным двоичным кодом.

Пакетная радиосвязь может осуществляться и через любительские спутники. Так, например, некоторые спутники позволяют обмениваться пакетами через специальный цифровой ретранслятор. Другие спутники позволяют получить от них информационные бюллетени в пакетном режиме. Вообще говоря, пакетная связь чрезвычайно удобна для связи через низкоорбитальные спутники, время радиовидимости которых ограничено минутами. Несмотря на это, за счет достаточно высокой скорости передачи (1200 Бод) можно успеть обменяться информацией со многими корреспондентами.

Программа BayCom Terminal — одна из лучших для Packet Radio

Программа BayCom и другие

Программа BayCom 1.5a была опубликована немецкими программистами-радиолюбителями DL8MBT и DG6RBU в 1992 году. В нашей стране она появилась примерно в 1994 году и сразу стала довольно популярной среди радиолюбителей. Основная причина популярности заключалась в том, что программе не требовался дорогой и дефицитный аппарат TNC. Программа имела в своем составе так называемый "встроенный программный TNC", роль которого выполняла программа L2. В некоторых источниках эта программа называлась "эмулятор TNC". Также популярности программы способствовал тот факт, что программа кроме основного режима работы, т. е. работы с нормальным, достаточно сложным, модемом, могла работать и с так называемым "мини-модемом", состоящим из одного операционного

усилителя типа K140УД7. Такое построение программ с двумя режимами работы — нормальным режимом и режимом упрощенным — было в тот период своеобразной модой. Аналогичным образом была выполнена программа НАМСОМ и некоторые другие программы. Очень многие польстились на простоту изготовления "мини-модема", но так и не добившись хорошей работы с этим аппаратом, прекратили эксперименты с Packet Radio, так и не попробовав очень хорошие программы в нормальном режиме работы с нормальным модемом.

По моему мнению, увлечение "мини-модемом" сыграло отрицательную роль в развитии пакетной связи в нашей стране. Основываясь на неудачных опытах с этим аппаратом, некоторые горе-пакетчики стали распространять слухи о плохой работе программ, о том, что работать пакетом можно только на импортной аппаратуре.

К счастью, таких "мини-пакетчиков" было не так уж и много. Основная масса увлекающихся пакетной связью коротковолновиков продолжала работу по совершенствованию отечественных аппаратов для пакетной связи. Ими создавались и новые конструкции аппаратных TNC, и новые конструкции модемов для программных "эмуляторов TNC".

В 1994 году этими же авторами программа BayCom была усовершенствована и получила наименование BayCom Terminal 1.60. Программа данной версии имела несколько модификаций, из которых необходимо отметить BayCom Terminal 1.60u для PC/FlexNet. Этот вариант программы вышел в 1997 году.

Указанные выше программы BayCom и много других программ можно найти в Интернете по адресу <http://www.baycom.org>.

Кроме BayCom, имелось большое число прочих пакетных программ. Большинство из этих программ предназначались для работы в качестве BBS (Bulletin Board System), MBBS (Mail and Bulletin Board System) — в качестве различного рода программ электронной почты и электронных досок объявлений. И очень мало было программ, которые в своем наименовании имели слово "Terminal" — такие программы предназначались или для проведения обычных радиосвязей, или для одноразового получения от BBS необходимой информации.

Чтобы рассказать хотя бы понемногу обо всех популярных программах для Packet Radio, надо бы написать еще одну книгу. Ограничусь только перечислением самых популярных программ.

- ❑ Программы для MBBS, которые разработал французский радиолобитель Jean Paul Roubelat (F6FBB).
- ❑ Программа WINPACK 6.XX, разработал Roger Barker (G4IDE).
- ❑ Программы BayCom от DL8MBT и группы.
- ❑ Программы, которые разработал англичанин Jon Welch (G7JFF).

- ☐ Программа ТРК, разработал француз Gerhard Regnard (F1EBN).
 - ☐ Программы и драйверы к программам, которые разработал T. Sailer, HB9JNX.
 - ☐ Программа НС, разработанная при Центре подготовки космонавтов группой программистов во главе с RV3DGA.
 - ☐ Драйверы и программы от G8BPQ — John Wiseman.
 - ☐ Программы для связи пакетной сети с Интернетом, автор Phil Karn (K9Q).
- И много, много других очень хороших программ!

Коротко о программе BayCom Terminal

Программа может работать с модемами типа MODEM22 или MODEM3, описания которых даны в *главе 3*. Подключение модема к COM-порту компьютера и к радиостанции выполняется аналогично подключению модема в программах для работы RTTY (телетайпом) (см. табл. 2.8).

Главные особенности и свойства программы:

- ☐ программа состоит из двух основных частей — эмулятора TNC (L2) и собственно терминальной части (SCC). TNC-часть работает резидентно и поэтому незаметна при работе других программ под DOS. После ее запуска уже возможны соединения с другими станциями. Но их сообщения будут выданы на дисплей только после старта терминальной программы;
- ☐ в терминальной части задействован редактор текстовых строк, поэтому каждое задание программе с помощью перемещения курсора можно использовать повторно;
- ☐ экран разделен на три окна: окно для передачи (Tx), окно для приема (Rx) и окно монитора. Можно переводить курсор в любое окно, изменять их размеры, "перелистывать" текст в окнах вперед и назад;
- ☐ задействованы обычные свойства многоканального соединения, включая прием и передачу файлов;
- ☐ накопление и повторное воспроизведение содержащегося на дисплее текста после кратковременного выхода из программы.

Взять эту и другие программы известной фирмы можно в Интернете по адресу <http://www.baycom.org/>.

Пакет программы BayCom Terminal состоит из следующих основных файлов.

- ☐ L2.EXE — уже упоминавшийся эмулятор TNC Baycom-системы. Программа имеет важную особенность — работает резидентно. Это означает, что после старта программы возможна обычная работа в среде DOS (обработка текстов, программирование и т. д., только вы лишаетесь временно около 90 Кбайт памяти). Индикатором включения является маленький квадратик в правом верхнем углу экрана. Теперь можно связываться с

другими станциями и передавать им информацию. Вывод на дисплей принятой информации возможен только после старта терминальной части (SCC.EXE).

- ❑ SCC.EXE — терминальная часть. После старта программы дисплей разделяется на три окна, описанных выше.
- ❑ OFF.COM — служит только для удаления L2.EXE из памяти PC.
- ❑ SCC.INI — конфигурационный файл. По оформлению это обычный ASCII-файл, содержащий все необходимые параметры пользователя. Он обрабатывается любым текстовым редактором. До начала работы в него должны быть занесены соответствующие позывные и параметры.
- ❑ PARA.EXE — служит для преобразования заданных пользователем параметров в понятный для компьютера формат. Стартует автоматически после вызова L2, если были внесены изменения в SCC.INI. Возможен и самостоятельный запуск. Программа выдает сообщение в случае некорректного задания параметров.
- ❑ SCC.PAR — машиночитаемый файл параметров, созданный PARA.EXE после первого вызова из SCC.INI.
- ❑ TERMHELP.SCC — Help-файл, содержащий тексты, которые могут быть вызваны нажатием комбинации клавиш <Alt>+<F1>.
- ❑ SCC.VID — копия дисплея, создаваемая при выходе из программы. Создается после первого старта BayCom и выводится при следующем старте.
- ❑ SCC.PWD — файл с паролями.
- ❑ SCC.LOG — файл простейшего "вахтенного" журнала, создаваемый в случае установки LOG-команды.
- ❑ BAYCOM.BAT — командный файл для запуска программы. При вызове стартует L2 и включается терминал. При окончании пакет радиоэксплуатации L2 автоматически удаляется.
- ❑ LOG.SCC — файл, содержащий вахтенный журнал станции, если функция LOG в положении ON. Этот файл может обрабатываться любым текстовым редактором.

Команды программы BayCom Terminal

Короткое обозрение команд, применяемых в программе, приведено в табл. 2.16.

Таблица 2.16. Основные команды

Команда	Выполнение	Функция команды
ANSWER	ON/OFF	Персональное приветствие вкл/откл
BADRESS	<позывной>	Установка адреса для маяка
BEACON	<число>	Период передачи маяка

Таблица 2.16 (продолжение)

Команда	Выполнение	Функция команды
BT	<текст>	Написать текст маяка
C	<позывной> [<ретранслят>]	Установить связь, обычная запись
CA	<канал><x>	Программное шумоподавление с SCC-картой
CA	<0/1>	Программное шумоподавление (0 — вкл, 1 — откл)
CBELL	ON/OFF	Сигнал при connect вкл/откл
CL	[номер]	Очистка дисплея
CNOT	<позывные>	Перечень игнорируемых позывных
COMBASE		Адрес COM-порта в случае отклонения
COMINT		Номер прерывания COM-порта
COMMAND	ON/OFF	Автоматическое двоеточие после команды
CRTSAVE	<минуты>	Автоматическое отключение дисплея
CS		Выдает список пользователей
CT	<текст>	Установка текста CTEXT
D		Разрыв связи
DAMA	<X>	Альтернативная работа репитера
DC	<позывной>	Установка собственного Digi-позывного
DISCT	<число>	Вывод времени Timer при разрыве связи
DW	<X>	Время ожидания сигнала
E	<имя>	Файл с клавиатуры
EN	ON/OFF	Переключение англ/нем языков сообщений
EXITLOG	ON/OFF	Запись в журнал сразу или при выходе
FR	<X>	Время ожидания подтверждения
GERMAN	ON/OFF	Включение немецких специальных знаков
H		Вызов файла помощи
HEN	<канал><0/1>	Переключение 9600 Бод модема по DF9IC
HB	<X>	Скорость
I		Выдать текст с информацией
IP	<X>	Максимальная длина информационного поля
KN	ON/OFF	Подавитель шума вкл/откл

Таблица 2.16 (продолжение)

Команда	Выполнение	Функция команды
L		Выдать стандартный текст
LINKT	<X>	Лимит времени при связи при отсутств. информации
LOG	ON/OFF	Запись в журнал вкл/откл
MAX	<X>	Количество пакетов до подтверждения
MH	<опция>	Выдать MHEARD-список
MHEAD		Управление адресной строкой в окне
MC	<позывные>	Выбор позывных, выводимых в окне
MO	<номера каналов>	Установка выводимых каналов в окне
MOUSE	ON/OFF	Подключение мыши
MY	<позывной>	Собственный позывной
O	<команда>	Ввод DOS-команды
PAC	<X>	Максимальная длина пакета
Q		Прекратить связь по условию
QT	<буква>	Передача Quit-текста
R	<имя файла>	Передача текстового файла
RCMD	<команды>	Команды, разрешенные для телеуправления
REM	ON/OFF	Разрешение телеуправления
RES	<X>	Задержка выдачи подтверждения
RET	<X>	Число повторений передачи
RP	<имя файла>	Передача двоичного файла
SA	<атрибуты>	Установка цветов дисплея
SCCBASE		Базисный адрес SCC-карты
SCCINT		Номер прерываний SCC-карты
ST		Задание стандартных текстов
SY		Окончание программы
TA	<X>	Время задержки выключения передатчика
TI	<X>	Таймер Info-передачи
TQ	<X>	Максимальное время после Quit
TX	<X>	Время задержки включения передатчика

Таблица 2.16 (окончание)

Команда	Выполнение	Функция команды
U		Выдать список всех пользователей
V	<имя файла>	Просмотр текстового файла
VE		Выдать номер версии программы
W	<имя файла>	Прием текстового файла
WP	<имя файла>	Прием двоичного файла
WW	<X>	Позиция автоматического конца строки
X	<my> [<ретрансл.>]	Связь под любым Mucall <my>

Команды, передаваемые нажатием на клавиши, приведены в табл. 2.17.

Таблица 2.17. Команды при нажатии на клавиши

Клавиши	Выполняемая команда
<F1>—<F8>	Переключение портов
<F9>	Переход в другое окно
<F10>	Monitor + Unproto-окно
<Esc> или <Tab>	Переключение команда/информация
<Shift>+<Tab>	Остановить выдачу из выбранного порта
<Ctrl>+<Return>	Переход в другую строку без ее использования
<PageDown>	Переход по тексту вниз
<PageUp>	Переход по тексту вверх
<Ctrl>+<PageUp>	Увеличение окна монитора, текст сдвигается
<CTRL>+<PageDown>	Уменьшение окна монитора, текст сдвигается
<Ctrl>+<Home>	Уменьшение Tx-окна, текст сдвигается
<Ctrl>+<END>	Увеличение Tx-окна, текст сдвигается
<Ctrl>+<F1>	Help-текст на слово под курсором
<Alt>+<X>	Окончание SCC.EXE, выход в DOS
<Alt>+<1>	Перевод курсора в Tx-окно
<Alt>+<2>	Перевод курсора в Rx-окно
<Alt>+<3>	Перевод курсора в окно монитора

Таблица 2.17 (окончание)

Клавиши	Выполняемая команда
<Alt>+<U>	Переключение умляутов
<Alt>+<F1>	Повторная передача последнего Help-текста
<Insert>	Переключение Insert-режима

Конфигурационный файл BayCom Terminal

Листинг 2.1. Файл SCC.INI

```
; Инициализационный файл для BayCom-Terminal Version 1.60
;
; Выбор командных установок (данных).
; Слова и строки после точки с запятой (;), программой игнорируются.
; Если данные для установки отсутствуют, то вместо них ставить 0.
;
combase 2F8h ; Адрес COM-порта для модема
; COM-адреса: COM1-3F8h, COM2-2F8h, COM3-3E8h, COM4-2E8h
; COM-прерывания: COM1 - 4, COM2 - 3, COM3 - 4, COM4 - 3
comint 3 ; Номер прерывания для порта с модемом
kissbase 0h ; Адрес COM-порта для KISS-Mode
kissint 0 ; Номер прерываний для порта KISS-Mode
;
fskbase 0h ; Адрес LPT-порта для PAR96-Modem
fskint 0 ; Номер прерываний LPT-порта с PART-Modem
;
sccbase 0h ; Адрес USCC-Karte
sccint 0 ; Номер прерываний USCC-Karte
;
;
; Данные по распределению памяти
tports 5 ; 2..8 Число ЭКРАНОВ (1 для мониторинга)
buffer 40 ; 40..150 Общее число буферов AX.25 в ОЗУ
mbuffer 5 ; 2..50 Общее число Monitor-Buffer
rbuffer 10 ; 4..20 Общее число приемных (RX) буферов
;
; Установка ВИДЕО
```


; EIN – аналогично английскому ON – означает включение данной опции
 ; AUS – аналогично английскому OFF – означает выключение данной опции
 color EIN ; вкл/откл EIN – цветное изображение, AUS – монохромное
 vgalines AUS ; вкл/откл При применении назначает 43/50 строк
 dllines 6 ; 2..20 Число строк в передающем (TX) окне
 d2lines 20 ; 4..23 Номер строки начала окна мониторинга
 vcolumns 80 ; 80..132 Количество букв (столбцов) в строке
 vlins 25 ; 25..75 Количество задействованных строк на экране
 saveports 2 ; 0..8 Количество задействованных портов
 wwrap 80 ; 40..100 Позиция для автоматического перевода строки
 crtsave 10 ; 0..100 Число минут бездействия до погасания экрана
 insmode EIN ; вкл/откл Переключение режима ВСТАВКА/ЗАМЕНА
 dosclear AUS ; вкл/откл Необходимость запоминания экрана при перезапуске
 ; EIN-экран очищается, AUS-экран снова восстанавливается
 mheader EIN ; вкл/откл Перевод строки перед информацией
 german EIN ; вкл/откл Немецкие буквы с умлаутиами
 remote EIN ; вкл/откл Допустить удаленное управление
 ; программой вашему корреспонденту
 command EIN ; вкл/откл Вводить автоматически двоеточие после ввода
 ; каждой команды (быть в командном режиме)
 echo AUS ; вкл/откл Вывод собственного передаваемого
 ; текста в RX-окне
 log EIN ; вкл/откл Вести журнал учета работы в файле SCC.LOG
 exitlog AUS ; вкл/откл Запись в журнал вести только перед выключением
 knax EIN ; вкл/откл Звуковой сигнал при удачном приеме инфо
 cbell EIN ; вкл/откл Звуковой сигнал при ;Connect/Disconnect/CTRL-G
 mouse AUS ; вкл/откл Работа манипулятора "мышь"
 7psave EIN ; вкл/откл Автоматическое записывание 7plus-файлов
 autobin EIN ; вкл/откл Автоматический перенос Bin-файлов
 blink 1 ; 0..2 L2-Blinken: 0=AUS, 1=EIN, 2=EIN если поданы новые
 ; данные

; Здесь представлены доступные для удаленного пользователя команды
 rcmd cstatus info mheard quit rtt read write rprg wprg dir
 ;

; Файлы, которые автоматически открываются с запуском программы, и в
 ; которые вносятся все принимаемые данные, проходящие по данному каналу.
 ; Данные каждой новой работы добавляются к предыдущим.
 write 1 port1.scc ;Для файлов, расположенных в других каталогах,
 write 2 port2.scc ; представить полный маршрут

```
write 3 AUS
write 4 AUS
write 5 AUS
write 6 AUS
write 7 AUS
write 0 AUS
;
; Назначение позывных
mycall RA3XB RA3XB-1 RA3XB-2 RA3XB-3 ; собственные позывные (max. 4)
dcall RA3XB-10 ; Собственный ретранслятор (можно и как MYCALL)
connect GT-1 ; Спецпозывной для 'квази connect' на порте 0
cnot NOCALL TNC2C ; Перечень игнорируемых для связи позывных
;
; Назначение служебных текстов
;
; Текст при CONNECT с неработающей программе SCC.EXE
l2ctext PC AT present BayCom \v – Terminal not Active ;собственно
; программа не активна, но сообщение можно оставить
;
;name Gennady ;отмечается при инсталляции
;qth KO73fu, Ljudinovo, nr.Kaluga
;
; Назначается буква, по которой выбирается Connecttext
ctext T
;
; Назначается буква, по которой выбирается текст в случае окончания
; связи по команде Q
qtext q
;
answer EIN ; вкл/откл Приветствие при CONNECT, берется из файла CALL.CTX
;
; Назначение служебных текстов, вызываемых нажатием определенной буквы.
; Каждая строчка текста обозначается буквой с цифрой-номером строки.
st n \xGENNADY
st g \xLudinovo
st i0 \x My Rig – PC AT, PCU – Pentium-100
st i1 \x Soft BayCom \v
st i2 \x TRNSVR 200 Watts
st i3 \x Modem Analog – Digi Filter
```

```

st i4 \x \c de \m >
st q0 \x Time of a Connection \a
st q1 Good bye dear friend. Till we meet again, 73's BY!!!
st t0 Hello, dear \c !!!
st t1 Here is Genady - \m, QTH-Ljudinovo/KO73fu, with BayCom \v
st t2 It is Time - \t. Date - \d. Port - \k
st t3 \x \c de \m >
st z \x:users
;
; Параметры, допустимые для всех каналов
;
monitor 0 1 2 3 4 5 6 7 8 ; Перечень отображаемых на мониторе каналов
mselect 0 ; Выводимая информация: 0=вся, 1=QSO,
; 2=только от вас(MFROM), 3=только к вам(MTO), +10=ничего(NOT)
mcalls RA3XB RA3XB-1 RA3XB-2 RA3XB-3 ;ваши позывные
;
frack 40 ; 10..200 Время на рестарт фрейма в 100 ms на шаг (Timer 1)
resptime 2 ;1..10 RESPONSE-таймер в 100 ms на шаг(Timer 2)
linktime 120 ;6..120 Неактивность абонента в 10 сек на шаг (Timer 3)
paclen 80 ;1..256 Длина пакета на передачу. Выбирать из 1...256
retry 20 ;1..100 Число попыток без подтверждения приема
ipoll 256 ;1..100 max длина принимаемого пакета
disctime 360 ;0..600 Время для неактивного порта до разъединения
; в 10 сек на шаг
; Установка цветов экрана
; Во всех случаях цвета фона и главные вводятся в шестнадцатеричном формате
cselect 0 ; Атрибут для Monochrom (MDA, Hercules, Laptop LCD)
sattrib 0 07 ; TX окно (на передачу)
sattrib 1 70 ; Верхняя строка состояния (между RX- и TX-окнами)
sattrib 2 f0 ; STOP-индикатор в верхней строке состояния (мерцает!!)
sattrib 3 07 ; RX-окно (окно приема)
sattrib 4 70 ; нижняя строка состояния: Conn-CALL+Port-Nummer
sattrib 5 70 ; нижняя строка состояния: нет каналов, нет Connect
sattrib 6 0f ; нижняя строка состояния: канал задействован
sattrib 7 07 ; Monitor: адресное поле
sattrib 8 0f ; Monitor: информационное поле
sattrib 9 70 ; Текст справочного окна
sattrib 10 07 ; Отмеченное место в справочном окне

```

```
sattrib 11 8f ; Задействованное место в справочном окне
sattrib 12 70 ; Контрольные знаки (не применять 0,3,7,8 !)
;
cselect 13 ; Атрибут для цвета индикатора (CGA, EGA, VGA)
;
sattrib 0 02 ; TX-окно (на передачу)
sattrib 1 47 ; Верхняя строка состояния (между RX- и TX-окнами)
sattrib 2 c7 ; STOP-индикатор в верхнем окне состояния (мерцает!!)
sattrib 3 07 ; RX-окно (окно приема)
sattrib 4 47 ; нижняя строка состояния: индикация Conn-CALL+Port - Nummer
sattrib 5 40 ; нижняя строка состояния: не задействованный канал, нет
; Connect
sattrib 6 57 ; нижняя строка состояния: канал задействован
sattrib 7 03 ; Monitor: адресное поле
sattrib 8 02 ; Monitor: информационное поле
sattrib 9 17 ; Текст справочного окна
sattrib 10 1e ; Отмеченное место в справочном окне
sattrib 11 5e ; Задействованное место в справочном окне
sattrib 12 04 ; Контрольные знаки (не применять 0,3,7,8 !)
; Первая цифра — цвет фона, вторя — цвет букв.
; Цифры, соответствующие цветам
; 0 = черный 8 = темносерый
; 1 = синий 9 = светлосиний
; 2 = зеленый A = светлозеленый
; 3 = голубой B = светлоголубой
; 4 = красный C = светлокрасный
; 5 = фиолетовый D = сиреневый
; 6 = коричневый E = желтый
; 7 = белый F = яркобелый
;
; Установки для разделение работы каналов
; Здесь для каждого канала с индивидуальным TNC (или модемом)
; предназначен индивидуальный блок параметров. Номер канала
; будет определяться по ряду параметров.
; Параметры незадействованных каналов будут игнорироваться,
; удалять их не следует.
assign scc0 ; Блок параметров канала 0 для первой USCC-Karte
mode 300 ; показатель для этого канала (скорость БОД)
dwait 30 ; Канал свободный, ожидает (умножить на 10 ms)
```

```
txdelay 25 ; Задержка на передачу PTT (умножить на 10 ms)
maxfram 3 ; Мах число переданных пакетов (1..7)
beacon 2400 ; Время между посылками маяка, с (0 – маяк отключен)
badress CQ RA3XB ; Адресат для маяка <ziel> <mycall> <digi>
btext CQ de RA3XB,QTH=KO73fu , BayCom \v ; текст маяка
;
;
assign rs232 ; Блок параметров для модема на порту RS232
mode 300 ; показатель этого канала (скорость БОД)
dwait 30 ; Канал свободный, ожидает (умножить на 10 ms)
txdelay 25 ; Задержка на передачу сигнала PTT, (умножить на 10 ms)
maxfram 1 ; Мах число переданных пакетов (1..7)
beacon 2400 ; Время между посылками маяка, с (0 – маяк отключен)
badress CQ RA3XB ; Адресат для маяка <ziel> <mycall> <digi>
btext CQ de RA3XB, QTH=KO73fu, BayCom v1.60 ; текст маяка.
; Далее должны находиться блоки параметров для других портов и TNC
; Конец файла
```

Программа BayCom Terminal является одной из лучших среди множества других программ-терминалов по следующим причинам:

- ☐ программа может работать с большим количеством разнообразных драйверов и устройств;
- ☐ вариант программы BCT 1.60u предназначен для работы в системе FLEXNET, что еще больше расширяет возможности программы;
- ☐ существует вариант BCT 1.60u для работы в операционной системе Linux;
- ☐ программа имеет расширенный файл документации, который можно редактировать. Документацию можно просматривать непосредственно при работе программы;
- ☐ программа позволяет проконтролировать все параметры и внести необходимые изменения в процессе работы.

Всемирная любительская радиосеть

Основные принципы работы радиосети

С 1 марта 1990 года радиолюбители Советского Союза получили возможность использовать цифровую радиосвязь — работать в радиосетях с пакетной коммутацией. Радиолюбители США и стран Западной Европы заимели такую возможность примерно на 12—15 лет раньше. При пакетной радиосвязи груп-

па радиостанций обменивается сообщениями на одной частоте, передавая информационные сообщения (пакеты) друг другу поочередно в различные отрезки времени. Благодаря коллективному использованию одной полосы частот многими станциями такой метод передачи очень эффективно и экономно расходует частотные ресурсы узких любительских диапазонов.

Каждый из передаваемых пакетов содержит контрольную сумму, проверка которой при приеме позволяет обнаружить ошибки, возникающие из-за помех, возможного взаимного наложения пакетов во времени и других причин. При обнаружении ошибки принимающая станция не передает пакет-квитанцию, в результате чего передающая повторяет передачу этого же пакета. Таким образом, достигается идентичность переданной и принятой информации, или, как говорят специалисты, надежный прием в ненадежных каналах.

Пакетная радиосвязь — высокоавтоматизированный вид любительского обмена: все процедуры (протоколы) выполняются без участия оператора с помощью аппаратного контроллера пакетной связи (TNC) или программного "эмулятора TNC" совместно с аппаратным модемом.

Скорость передачи данных на КВ-диапазонах принята 300 Бод, на УКВ-диапазонах — 1200 Бод и выше. Такое различие в рабочих скоростях зависит от наличия эфирных помех. По этой причине дальние связи на диапазонах ниже 7 МГц затруднены.

Пакетная радиосвязь чрезвычайно удобна для передачи различного рода подготовленной заранее документальной информации: сообщений, документов, бюллетеней, программ. Она же совершенно неэффективна при непосредственном обмене сообщениями, особенно на низкочастотных коротковолновых диапазонах. Для персонального общения лучше применить телетайп или неавтоматические виды связи CW, SSB. Дело в том, что хотя физическая скорость передачи пакетных данных в шесть раз выше, чем при радиотелетайпной любительской связи, суммарное время печатания сообщения и его пакетной передачи с переповторами в КВ-диапазоне часто оказывается невысокой.

Наибольшей эффективности использования пакетной связи удастся добиться, применяя этот вид связи в работе Всемирной любительской радиосети.

Весь мир охвачен сетью любительских радиостанций, работающих цифровой радиосвязью. Основу сети составляют MBBS и BBS — специальные станции, работающие круглосуточно в автоматическом режиме. Каждая из них имеет в сети свой специальный позывной, в котором указываются страна, область и город. Например, позывной RK6HWW.#PTG.STA.RUS.EU принадлежит очень известной в нашей стране станции RK6HWW, расположенной в городе Пятигорске Ставропольского края. Страна — РОССИЯ (RUS), континент — ЕВРОПА (EU). Основой MBBS является мощный компьютер, работающий под управлением специальной программы. BBS в

автоматическом режиме в установленное время и на установленных частотах проводит сеансы по обмену информацией со своими постоянными партнерами — другими MBBS. Многие MBBS и BBS позволяют пользоваться имеющейся в их памяти информацией всем желающим. Вся информация разделяется по отдельным темам (рубрикам) в виде бюллетеней и персональных сообщений. Кроме того, на многих BBS имеются файловые серверы, в которых хранятся файлы наиболее интересных радиолюбительских программ. Каждый радиолюбитель может создать и отправить в сеть на всеобщее обозрение бюллетень по любой теме. Главное условие для любого бюллетеня — этот бюллетень должен быть интересен хотя бы небольшой группе радиолюбителей. Каждый радиолюбитель может отправить через сеть персональное сообщение (письмо) любому из участвующих в работе сети радиолюбителю из любой страны. Главное условие — и пославшая сообщение станция и ее корреспондент должны быть "привязаны" к определенному BBS, как говорят радиолюбители, должна иметь свою "home BBS". Например, радиостанция RA3XB, расположенная в Калужской области, в качестве "home BBS" имеет радиостанцию BBS RW6HQN, расположенную в Ставропольском крае, при этом почтовый адрес радиостанции RA3XB во всемирной сети будет RA3XB@RW6HQN.STA.RUS.EU и вся поступающая в адрес RA3XB корреспонденция будет храниться в памяти BBS RW6HQN.

Всемирная радиолюбительская сеть имеет возможность передавать и получать свои сообщения через всемирную сеть Интернет.

Всемирная радиолюбительская сеть, в свою очередь, состоит из локальных сетей. Наибольшими по количеству станций являются локальные сети Северной Америки, Западной Европы, Японии и т. д. Радиолюбительская сеть России пока еще находится в зачаточном состоянии, не удовлетворяет запросам многих радиолюбителей, особенно проживающих в отдаленных от больших городов населенных пунктах. Ближайшей и наиболее доступной для проживающих в европейской части России радиолюбителей является радиолюбительская сеть Западной Европы.

Любительские сети пакетной радиосвязи обычно строятся по следующей схеме. Большинство абонентов обменивается информацией на УКВ при небольшой мощности своих передатчиков (обычно единицы или доли ватта), создавая локальные сети на определенных частотах в пределах 2-метрового или 70-сантиметрового диапазона. Для увеличения зоны действия сети используются полудуплексные ретрансляторы digipeaters или дуплексные ретрансляторы repeaters. Понятие "полудуплекс" подразумевает поочередную передачу и прием сообщений, а "дуплекс" позволяет вести передачу и принимать сообщения в одно и то же время. В качестве digipeaters могут быть использованы любые из абонентов сети, в качестве repeaters — специальные ретрансляторы, мгновенно ретранслирующие пакеты на другой частоте.

Для перехода из одной сети в другую используются межсетевые шлюзы (GATES) и узлы (NODES). Шлюзы осуществляют прямую ретрансляцию пакетов из одной сети в другую, т. е. с одной сетевой частоты на другую, например, КВ-диапазона в УКВ или УКВ в УКВ, но на разных частотах.

Узлы выполняют более сложную функцию. При установлении логического соединения с узлом абонент дает ему указания на выполнение операционных команд, например, соединить его с определенной станцией, дать общий вызов CQ. Узел самостоятельно выполняет эти команды многократно, после чего сообщает абоненту о результатах их выполнения.

Кроме того, узел собирает и передает по запросу абонентов сведения о сетях, к которым он имеет доступ, о ближайших к нему других узлах, о состоянии линий связи с этими узлами (надежности прохождения пакетов), об абонентах, которых узел слышал в последнее время, о состоянии узла в данный момент (с какими абонентами он соединен), о версиях узловых программ, владельце узла, географическом местонахождении.

Абоненты УКВ-сети, подключенной к данному BBS, обращаются к нему в удобное для себя время и, просматривая каталоги всех сообщений, выбирают и считывают интересующие их сообщения. Это избавляет абонентов сети от необходимости непосредственной связи со своими корреспондентами на КВ, т. е. от необходимости иметь достаточно мощную и дорогую КВ-радиостанцию, антенны. К тому же обмен данными на УКВ надежен и в четыре раза быстрее, чем на КВ, что экономит время абонентов.

Основное развитие любительских сетей в небольших по размерам странах Западной Европы, в Японии и других идет по линии создания локальных УКВ-сетей, наращиваемых вокруг региональных и других BBS. Это очень удобно и выгодно, потому что частотные полосы УКВ-диапазонов существенно шире, чем КВ-диапазонов, механизм распространения прямой видимости позволяет осуществлять передачу со скоростями 1200 Бод и более, антенны этого диапазона имеют малые габариты и, наконец, УКВ ЧМ-трансиверы малой мощности, работающие на одной-двух фиксированных частотах, могут быть малогабаритными, недорогими устройствами. Отдельные локальные УКВ-сети соединяются между собой через ретрансляторы, шлюзы или узлы как на УКВ, так и КВ-частотах. Конкретные формы взаимосоединения зависят как от взаимного территориального расположения сетей, так и от предпочтений связи между теми или иными сетями.

Наша страна раскинулась на огромном пространстве, где порой радиолюбителя от его ближайших коллег отделяют сотни километров. В таких условиях большое значение имеет создание надежных аппаратов и устройств для работы на коротковолновых диапазонах со скоростью 300 Бод и для работы пакетом через искусственные спутники Земли.

Несмотря на то, что цифровая связь Packet Radio выполняет основную работу в радиолюбительской сети, за последние годы в сети значительно возрос-

ло число MBBS и BBS, работающих цифровым видом связи PACTOR. По мнению некоторых радиолюбителей, число таких станций на коротковолновых диапазонах превосходит число пакетных станций. Это можно объяснить лучшей помехозащищенностью PACTOR на коротковолновых диапазонах. Наглядным примером служит радиолюбительская сеть Западной Европы.

Радиолюбительская сеть Западной Европы

Сеть состоит из очень большого числа BBS. В работе сети используются практически все виды цифровой радиосвязи. Особенностью этой сети является то, что режим Packet Radio используется только для работы на диапазонах УКВ со скоростями 1200 Бод и более. Это обусловлено большим количеством УКВ-ретрансляторов в странах Западной Европы, широким использованием спутниковой радиосвязи и малыми расстояниями между радиостанциями. С корреспондентами из стран Восточной Европы и более удаленных стран эти BBS предпочитают работать в режиме PACTOR.

Наиболее удобным диапазоном для цифровых видов связи является 20-метровый диапазон. От начала этого диапазона до частоты примерно 14,060 МГц безраздельно господствует телеграф, затем до частоты примерно 14,090 МГц располагается масса BBS, работающих в режиме PACTOR. Но наблюдать работу этих радиостанций можно только во время их работы с другими корреспондентами т. к. общих вызовов они не дают, а могут изредка включать маяк. Для работы с этими BBS необходимо знать расписание их рабочих частот. В качестве примера в табл. 2.18 приведено расписание рабочих частот BBS — радиостанции DA5UDK.

Таблица 2.18. Рабочие частоты DA5UDK

Частота (МГц)	Время (UTC)	Тип антенны
7,035	04 ... 06, 17 ... 23	Вертикальная
7,042	04 ... 06, 17 ... 23	Вертикальная
10,142	04 ... 23	Вертикальная
14,072	04 ... 23	Вертикальная
14,029, 14,102	04 ... 23	Двухэлементная
18,102	04 ... 23	Вертикальная
21,072	08 ... 17	Вертикальная
21,095	08 ... 17	Двухэлементная
28,142	08 ... 13	Двухэлементная

Практически все PACTOR BBS используют управляющую программу WINLINK. В табл. 2.19 привожу перечень команд программы WINLINK.

Полную информацию по этому вопросу можно посмотреть на Web-странице по адресу <http://www.scs-ptc.com/>.

Таблица 2.19. Команды программы WINLINK

Команда	Наименование	Функция команды
A	ABORT	Передающая станция должна прекратить передачу
B	BYE	Выход из WINLINK
CANCEL #		Удалить сообщение под номером #
CHANGExx		Контроль удаленных команд (для удаленного Sysop)
D #	DELETE	Удалить свое сообщение под номером #
E	EXPERT	Переключить состояние, показать состояние
H	HELP	Выдать на экран файл помощи (этот файл)
K #	KILL	Удалить сообщение под номером #
I	INFO	Выдать на экран файл информации
L	LIST	Список всех сообщений
LM	LIST	Список всех сообщений для себя
LB		Список новых бюллетеней (не просмотренных)
LB #		Список бюллетеней в системе, начиная с номера #
LLB # (loc)		Список последних бюллетеней с номером # (loc)
LF		Список файлов для FORWARDING
LH		Список всех файлов помощи
LI		Список INTERCEPT-файлов
LL		Список новых сообщений, после последнего запроса
LL # (loc)		Список последних новых сообщений, начиная с номера #
LN		Список всех своих непрочитанных сообщений
LOGOUT, QRT		Выход из WINLINK
LR		Список пользователей за последние 24 часа
LT		Список всех NTS-сообщений
LY		Список используемой станцией статистики
L>CALL	LTO	Список сообщений для CALL
L<CALL	LFM	Список сообщений от CALL

Таблица 2.19 (окончание)

Команда	Наименование	Функция команды
L@ CALL		Список сообщений, адресованных как @CALL
NTS		Список ожидаемых NTS-сообщений
PAGE		Включить PageMode; показывает параметр
R #	READ #	Читать сообщение с номером # без заголовка
RC		Читать все свои ожидаемые сообщения
RH #		Читать сообщение под номером # с заголовком
RM, RN		Читать все свои непрочитанные сообщения
RMM # # # #		Читать много сообщений по номерам с остановками
RMMC # # # #		Читать много сообщений по номерам без остановок
SP CALL, S		Послать частное сообщение для CALL
SP CALL@BBS		Послать частное сообщение для CALL@BBSCALL
SB TOPIC		Послать бюллетень для рубрики TOPIC
SB TOPIC@DES		Послать бюллетень для рубрики TOPIC@DES
SFM		Послать файл с сообщением
ST		Послать NTS-сообщение
TIME		Показать ежедневный лимит времени
T	TALK	Запросить общение с системным оператором (Sysop)
V	VERSION	Запросить текущую версию программы WINLINK

Все посылаемые сообщения должны заканчиваться новой строкой, состоящей из трех напечатанных символов /EX и последующего нажатия клавиши <Enter>.

Перечень РАСТОР BBS

В табл. 2.20 привожу очень краткий перечень BBS, которые можно часто слышать в европейской части России.

Таблица 2.20. Перечень BBS

Позывной BBS	Рабочие частоты, кГц
9A0APL.HRV.EU	7037, 14069, 14072, 14079
DK0MUN.#BAY.DEU.EU	14066, 14069
F3KT.FPDL.FRA.EU	7039, 14071. 14073, 18102, 21072, 28103

Таблица 2.20 (окончание)

Позывной BBS	Рабочие частоты, кГц
GB7SIG	3580, 7038, 10145, 14075, 21080
HB9AK.CHE.EU	3581, 7039, 14071, 14097, 21070, 28079
OH2BAW.FIN.EU	3587, 7037, 10145, 14169, 14075
OH2NPE.FIN.EU	14074, 14078
ON0BEL.#LG.BEL.EU	3578, 7037, 10145, 14072, 21075
OZ2AMT.ISH.SJL.DNK.EU	3580, 7037, 10133, 14072, 14117
OZ4SCA.OEL.JYN.DNK.EU	3580, 7037, 14068, 14072, 21073, 28073
PA3GMZ.#ZHI.NDL.EU	7037, 14069, 14079, 28079, 28085
SM7TDC.M.SWE.EU	3584, 3593, 7036, 14065, 14072, 14075
SV1IW.GRC.EU	14068, 14072, 21070, 21073, 21077
TA7TRA.#TRA.TUR.AS	14068
TJ1GD.CMR.AF	14068, 21073
UA6ADV	14063, 14075
VK2AAB.#SYD.NSW.AUS.OC	14065, 14070, 14076, 14078
VK2AGE.#NE.NSW.AUS.OC	7045, 10108, 14077, 21076

Сеть AMPR

Интернет прочно вошел в жизнь нашего общества. Радилюбительская сеть получила новые возможности — войти в Интернет и стать как бы его составной частью благодаря протоколу TCP/IP.

TCP/IP (Transport Control Protocol/Internet Protocol) — это профессиональный коммуникационный протокол, который позволяет связать различные компьютерные системы, использующие разные сетевые протоколы (в том числе связать локальные сети пакетной радиосвязи с сетью Интернет), в единую глобальную компьютерную сеть.

Для работы в TCP/IP Фил Карн (Phil Karn, KA9Q) в 1989 году выпустил свою знаменитую программу NET, впоследствии доработанную и названную NOS (Network Operating System). В настоящий момент существует более десятка версий NOS, доступных в исходных кодах, для различных машин (IBM, Sun, Macintosh и т. д.) и различных операционных систем (MS-DOS, Windows, UNIX, OS/2). Это: KA9Q, JNOS, PA0GRI, GRINOS, WNOS, GPSNOS, GRACILIS, WAMPES, PMNOS, G1EMM, TNOS. Наиболее разви-

той, популярной и поддерживаемой версией является JNOS, разработанная Johan Reinalda, WG7J/PA3DIS.

Для персональной идентификации каждая TCP/IP-станция имеет свой уникальный IP-адрес. В соответствии с принятым в профессиональной сети Интернет стандартом, IP-адрес состоит из четырехбайтной последовательности, где каждый байт дает определенную информацию о том, к какой сети и подсети относится данная станция. IP-адрес имеет иерархическую структуру, т. е. крайний левый байт имеет максимальную значимость (определяет всю сеть), а крайний правый минимальную (определяет конкретную станцию в сети). Такое построение необходимо для маршрутизации информации как внутри одной сети, так и между разными компьютерными сетями. Все адреса радиолучительских TCP/IP-станций начинаются с номера 44. Например, адрес TCP/IP-станции RZ6HXA 44.178.92.2 расшифровывается так:

- 44 — радиолучительская TCP/IP-сеть;
- 178 — Россия;
- 92 — Ставропольский край;
- 2 — персональный номер.

Теперь рассмотрим основные протоколы, выполняемые на TCP/IP-станции.

- Telnet — протокол эмуляции терминала. Этот протокол позволяет осуществить терминальное соединение с удаленной TCP/IP-станцией и очень похож на соединение одной пакетной станции с другой (пользователя с BBS).
- FTP (File Transfer Protocol) — протокол передачи файлов — позволяет организовать пересылку ASCII и бинарных файлов (в том числе и программ) между TCP/IP-станциями.
- SMTP (Simple Mail Transfer Protocol) — протокол передачи почтовых сообщений — позволяет переслать почтовое сообщение на указанный почтовый адрес.
- POP (Post Office Protocol) — разновидность почтового протокола. Для работы SMTP-протокола необходимо, чтобы удаленная станция работала 24 часа в сутки. Это не всегда удобно. Поэтому при POP-протоколе почта, предназначенная вам, складывается на почтовом сервере (Mailbox), а при появлении вашей станции в эфире — пересылается вам.
- Ping (Packet InterNet Groper) — протокол, служащий для диагностики канала. Протокол инициирует посылку пакета-запроса на удаленную станцию, ждет ответа и показывает время соединения с удаленной станцией.
- FINGER — данный протокол позволяет получить информацию о пользователях на удаленной станции.

В результате "слияния" радиолучительской сети с Интернетом возникла так называемая "сеть AMPR", которая является частью Всемирной радиолучительской сети, связующим звеном между этой сетью и Интернетом.

Все возрастающий объем передаваемой информации по сетям пакетного радио остро ставит вопрос об использовании скоростей передачи значительно больших, чем 300 Бод на диапазонах КВ, 1200 и 9600 Бод на УКВ. И если создание высокоскоростной сети в пределах большого города уже не представляет сложности, то в удаленных от больших городов регионах встречаются большие трудности в связи с ограниченной зоной действия УКВ-сети.

Выйти из создавшегося затруднения можно путем использования искусственных спутников Земли (ИСЗ). В настоящее время на круговых полярных и эллиптических орбитах находится большое число радиолюбительских спутников, на которых установлена специальная аппаратура, поддерживающая пакетную радиосвязь.

Если любительская радиостанция имеет аппаратуру и антенные системы для приема и передачи информации от аппаратуры, установленной на ИСЗ, то эта станция может существенно повысить качество связи. Обмен бюллетенями и персональной почтой через низкоорбитальные радиолюбительские спутники как "летающие почтовые ящики" возможен с периодом примерно 100 минут. Каждый из таких сеансов связи позволяет получить до 900 Кбит информации за один виток, что однозначно решает вопрос эффективного получения информации из любой точки мира.

Самым эффективным средством повышения скорости в пакетной сети является использование сети AMPR (Internet \longleftrightarrow Packet Radio). Но при этом радиосеть превращается в сеть ... проводную, телефонную.

В основе этого способа лежит использование протоколов TCP/IP глобальной компьютерной сети Интернет, которые используют AX.25 в качестве транспортной среды, передавая собственные данные "поверх" него. Главными достоинствами являются высокая скорость передачи и стабильность канала. В Интернете радиолюбителям выделена сеть 44 (класс А), что позволяет использовать 16 387 064 адресов во всем мире. Российская часть сети 44.178/16 позволяет обращаться к 65 536 адресам. Системы, поддерживающие транспорт и сервис TCP/IP, в состоянии использовать практически любые из существующих каналов связи: выделенные и коммутируемые телефонные линии, локальные компьютерные сети, системы цифровой радиосвязи и т. д.

Информационный сервис систем TCP/IP представляет набор готовых унифицированных служб: электронная почта, система терминального доступа, система доступа к файлам на специальных серверах, телеконференции (заочные и интерактивные), системы доступа к информационно-поисковым системам и базам данных и т. п.

Также важным достоинством TCP/IP-систем является то, что для установления связи с корреспондентом вам не обязательно знать маршрут до него (что является необходимым условием в AX.25), система сама выбирает оптимальный маршрут и устанавливает соединение.

Основным элементом сети AMPR является компьютер, объединяющий в единую логическую сетевую структуру сети Интернет и пакетного радио, называющийся AMPR-шлюз (AMPR gateway). Такой компьютер имеет, по крайней мере, два интерфейса — один для сети Интернет, а другой для пакетного радио. Причем каждый интерфейс имеет свой собственный IP-адрес.

Упрощенно AMPR-шлюз работает так: приняв данные по одному из интерфейсов (из Интернета или из пакетного радио) шлюз проверяет адрес назначения и в соответствии с таблицей маршрутизации отправляет данные на соответствующий интерфейс.

Наиболее популярным программным обеспечением для AMPR-шлюза являются:

- для операционной системы (ОС) MS-DOS — JNOS и TNOS;
- для операционной системы (ОС) Linux — JNOS и TNOS.

Возможно использование только ОС Linux версии 2.0 и выше, имеющей поддержку всех необходимых протоколов (AX.25, IP) в ядре.

Программа F6FBB — самая распространенная из MBBS

О самой программе и как ее запустить

Автор программы — Jean-Paul Reubelat (F6FBB). Программа создана в конце 80-х годов прошлого столетия, с тех пор претерпела массу изменений и переработок, имеет варианты для работы почти во всех операционных системах и работает со всеми известными TNC и модемами.

Наиболее распространенными являются варианты программы для MS-DOS и для Windows. Основные функции того и другого вариантов мало отличаются один от другого, но вариант для MS-DOS — F6FBB 5.15c — более привлекателен для начального освоения тем, что он отлично согласуется и работает с эмулятором терминала TFPCX, т. е. не требует аппаратного контроллера TNC. Особенности этой программы и будут описаны ниже.

Важным моментом при использовании программы является правильное оформление главного инициализационного файла, который носит название `appel.bat`.

Ниже приведен текст этого файла, при этом предполагается, что сама программа и все другие, необходимые для ее нормальной работы программы, находятся на диске D:. На этом же диске в подкаталоге D:\TFPCX210\ должны находиться файлы `tfpcx.286`, `tfpcx.ini` и другие нужные файлы.

Листинг 2.2. Командный файл appel.bat

```
REM Начало файла
@echo off
path=d:\dos;d:\arc;d:\drv;d:\fbb;d:\fbb\system;d:\fbb\bin
cd \fbb\system
rem clr_user
cd \fbb
d:\tfpcx210\tfpcx286 -n -pcom2 -b300 -c30 -Fd:\tfpcx210\tfpcx.ini
break off
echo Программа запускается, нажать Ctrl-C для останова в течение 3 сек.
sleep 3
serv %1
if errorlevel 2 goto direct
if errorlevel 1 goto suite
goto fin
:suite
cls
echo Sorting HROUTE.SYS...
copy \fbb\system\hroute.sys temp.$$$
type temp.$$$ | sort | find /v "#" " > \fbb\system\hroute.sys
del temp.$$$
echo Backup of important files...
copy \fbb\system\dirmes.sys \fbb\backup\dirmes.sl
copy \fbb\system\inf.sys \fbb\backup\inf.sl
echo Cleaning up message-files...
del epurmess.res
epurmess
echo Deleting old 7plus files parts (older than 30 days)
AUTO7P DELETIME 30
cd \fbb\system
MAINTINF 3 /a
COPY D:\FBB\SYSTEM\INF.NEW D:\FBB\SYSTEM\INF.SYS
cd \fbb
del d:\fbb\system\dirmes.new
:direct
cd \fbb\system
tlabel
cd \fbb\system\sat
```



```

if not exist amsat.txt goto appel
satupdat amsat.txt /n /s > satupdat.res
satupdat amsat.txt >> satupdat.res
del amsat.txt
:appel
cd \fbb
echo Программа запускается, нажмите Ctrl-C для останова в течение 7 сек.
sleep 7
serv -t
if errorlevel 2 goto direct
if errorlevel 1 goto suite
:fin
D:\tfpcx210\tfpcx286 -n -u
echo Программа остановлена и закрыта.
break on
Echo on
Rem Конец файла

```

Следующим важным конфигурационным файлом является файл port.sys, который выполняет конфигурирование COM-порта. Ниже привожу текст всего файла, необходимого для организации работы программы через TFCPX, порт COM2, на одном диапазоне.

Листинг 2.3. Файл port.sys

```

#####
# PORT.SYS for FBB 5.15 #
# for RA3XB #
#####
# Файл для программирования каналов и TNC.
#
# Ports: используемые порты (COM1, COM2, и т. д.)
# TNCs: число используемых TNC или модемов. С multiplexer
# можно использовать до 4-х TNC на порт.
#
#Ports TNCs
1 1
# 1 2
#

```

```
# Com: COM-номер (1,2,...8)
# Interface: 1 = Внешний ESS/COMBIOS драйвер
# 2 = BPQ-node V 4.05 или выше.
# 3 = Telephone-modem совместно с FBBIOS
# 4 = DRSI карта с драйвером (WA8DED host-mode)
# 5 = TFPCR/TFPCX интерфейс (WA8DED host-mode)
# Adress: адрес порта (hexadecimal). Игнорируется при
# COMBIOS/BPQ/MODEM.
# Baud: Скорость в Бод. Игнорируется при BPQ.
#
# Число строк (ниже) должно быть равно числу портов.
#
# Com Interface Adress (Hex) Baud
# 7 4 0 1200
2 5 0x2f8 300
#
# TNC: Число используемых TNC.
# NbCh: Число каналов, которое я хочу использовать с TNC.
# Max задействованных каналов возможно с firmware.
# Com: Номер COM-порта. Com1, Com2 и т. д.
# MultCh: Число каналов при использовании port-multiplexer, иначе 1.
# При DRSI использовать число 0 ... 7, при KAM — 1/VHF и 2/HF.
# PacLen: PACLEN (длина пакета в байтах) на этом TNC.
# Maxframe: max число frames TNC было передано за время.
# NbFwd: Число каналов для OUTGOING forward за это же время.
# MxBloc: Размер forward-block в килобайтах.
#
# M/P-Fwd: Минуты времени для старта forward, и период
# (число минут до начала forward-start).
# Mode: Одна из следующих:
# B: BBS-mode.
# G: "Guest"-mode.
# U: Normal-mode.
# Тип host-mode (главного режима), один из следующих:
# D: WA8DED
# K: KAM host-mode.
# P: PK-232
# Q: BPQ v 4.x
```

```
# Addition: Одна из этих букв используется при:
# L: Передача unproto маяка перед каждым поступлением mail.
# M: Telephone-modem.
# Y: Yapp задействован на этом QRG.
# W: Gateway задействован для этого QRG.
# Freq.: Текстовое описание этого порта (max 9 символов, без пробелов)
# Число строк должно соответствовать числу TNC.
#
# TNC NbCh Com MultCh Pacln Maxfr NbFwd MxBloc M/P-Fwd Mode Freq
1 4 2 0 80 2 1 10 00/60 UDYW HF
# 1 5 7 0 80 2 1 10 00/60 UDYW VHF
# 2 5 7 1 80 2 1 10 00/60 UDYW VHF
#
# Специальный позывной и modes для этих каналов.
#
#TNC Nbs Callsign-SSID Mode
#1 1 UA3XFA B
#
# Конец файла.
#
```

Основные команды MBBS

Краткий перечень команд программы F6FBV — MBBS

Программа имеет большое количество команд, которые могут быть доступны удаленному пользователю. В табл. 2.21 приведен краткий перечень только основных команд.

Таблица 2.21. Перечень основных команд

Команда	Обозначение	Функция команды
?, HELP	СПРАВКА	Выдает текст файла справок
A	Abort	Аварийное прекращение передачи файла
B	Bay	Выключение, прекращение связи
C	Conference	Доступ к конференции за круглым столом
D	DOS	Выход в MS-DOS
F		Перейти в режим сервера
FP	Find Path	Поиск диска по специфическому имени файла

Таблица 2.21 (продолжение)

Команда	Обозначение	Функция команды
G	Gateway	Переход на режим шлюза
I	Info	Информация по аппаратным средствам системы
I <позыв- ной>		Показать информацию по этому позывному
I@destinati on		Список всех зарегистрированных в Mailbox
ID		Показывает число существующих записей
IN <позыв- ной>		Показывает полный адрес позывного
IZ	Postal Area	Список зарегистрированных в почтовом сервере
J	Jheard	Список последних 20 соединений
K	Kill	Уничтожить сообщение
L	List	Список сообщений
M	Make	Копировать сообщение в файл
N	Name	Ввести в память ваше имя
NH	Home BBS	Ввести ваш Home BBS
NL	Location	Ввести название вашего города
NQ	QRA	Ввести ваш QRA-локатор
NZ	Postal area	Ввести ваш почтовый индекс
O	Options	Выбрать/изменить опции
PS	Servers	Показать доступные серверы
PG	Program	Показать доступные DOS-программы
R	Read	Чтение сообщений
S	Send	Передать сообщение
T	Talk	Разговаривать с системным оператором
U	Upload	Загрузить файл Mailbox
V	Verbose	Подробное чтение сообщений
W	What	Перечень доступных файлов
X	Expert	Изменить доступ Normal на Expert
Y	Yapp	Передача бинарных файлов по YAPP-протоколу
Z	Delete	Удалить файл

Таблица 2.21 (окончание)

Команда	Обозначение	Функция команды
>	Text	Передать текст станции, соединенной с BBS
=	Connect	Соединиться со станцией, соединенной с BBS
!	BBS Status	Получить на дисплей статус (состояние) системы
#	Mail for	Показать список позывных, для которых есть почта
*	Wildcard	Используется для ускорения ввода команд и поисков

Описание команд BBS/Mailbox

Справка по отдельной команде — ? *x*, где *x* является символом или именем команды.

Пример. Для получения справки по почтовым сообщениям введите: ? L <RETURN>.

Для справки относительно команды REQDIR: ? REQDIR <RETURN>.

Описание ключей *x*:

- ☐ ? — запрашивает справочный файл;
- ☐ ! — показывает всех соединенных сейчас с Mailbox;
- ☐ # — показывает список позывных тех, кто не читал почту;
- ☐ A — прерывает вывод из Mailbox в любое время. Однако не забудьте, чтобы в буфере TNC не было данных для передачи;
- ☐ B — отсоединяет вас от Mailbox;
- ☐ D — переместиться из Mailbox режима в режим FBB-DOS, и получить доступ к использованию собственной области файлов. Управление FBB-DOS подобно управлению в MS-DOS, допускаются wildcards и иерархические каталоги. Если после команды D будет стоять имя файла, этот файл будет загружен;
- ☐ F — переключение между режимами BBS и файлового сервера;
- ☐ FR — ищет путь к любому файлу в заданном каталоге. При этом можно применять wildcards (заменять слово звездочкой или другим значком);
- ☐ I — выдает информацию относительно аппаратных средств BBS, и как войти в контакт с Sysop (системным оператором);
- ☐ J — выдает список позывных, которые были услышаны или соединились с Mailbox:
 - JK: показывает 20 последних, соединенных позывных;
 - JA: соединенные позывные на порту A;

- JВ: соединенные позывные на порту В, и т. д.;
- J1: позывные, услышанные на порту А;
- J2: позывные, услышанные на порту В, и т. д.

Учитываются текущие 6 активных портов, маркированные от А до F;

❑ К — уничтожает сообщения, посланные или для, или от вашего позывного. Только вы или Sysop можете уничтожать сообщения, адресованные для вашего позывного:

- К msg#: уничтожить сообщение под номером #;
- км: уничтожить все адресованные на ваш позывной и прочитанные сообщения. Сообщения, которые вы не читали, не будут удалены;

❑ L — получить списки заголовков сообщений:

- L: списки любых новых сообщений с тех пор, когда вы использовали эту команду;
- LV: списки бюллетеней;
- LC: здесь пользователи могут определять маску, чтобы получить только те заголовки сообщения, которые соответствуют нужному требованию. Например, ввод команды LC AMSAT позволит получить список сообщений, адресованных AMSAT;
- LC ?: списки активных в данный момент рубрик;
- LC *: списки "Все типы сообщений";
- LD: списки с датой:
 - ◇ список почты со сроком раньше 1 января 1993 года;
 - ◇ список почты со сроком позже 1 января 1992 года;
- LM: список всех сообщений на ваш позывной;
- LN: список любых новых (нечитанных) сообщений к вам;
- LL #: список последних # сообщений, например, LL 30 дает список заголовков 30 последних сообщений;
- LS xxx: поиск текстовой строки во всех заголовках сообщений. Например, если вы хотели получить список всех сообщений, в заголовке которых содержится слово "радио", нужно просто напечатать LS РАДИО;
- LR: подобно команде L, но список сообщений будет распечатан в обратном порядке;
- L<Call: список сообщений от callsign (позывного);
- L>Call: список сообщений, адресованных к callsign;

- L@ BBS: список сообщений, полученных от BBS;
 - L Msg#: списки всех сообщений выше определенного номера;
 - L Msg#-#: списки заголовков всех сообщений между двумя числами;
- M — сделать сообщение в файл из области файлов FBB-DOS. Используйте синтаксис:
- M msg# path\filename: обратите внимание, если никакое утверждение пути не дано, сообщение будет помещено в корневой каталог FBB-DOS;
 - MA msg# path\filename: добавить сообщение к существующему файлу;
 - MV msg# path\filename: как в строке выше, но будет включена расширенная информация маршрутизации;
- N — эти команды используются для ввода вашей личной информации относительно регистрации в Mailbox, которая бывает необходима для расчета орбит ИСЗ и места нахождения радиостанции:
- N: ввести/изменить ваше имя;
 - NH: ввести/изменить ваш HomeBBS, через который намерены получать почту;
 - NL: ввести/изменить ваш город (месторасположения или локатор);
 - NP: изменить ваш пароль (только для телефонного доступа);
 - NQ: ввести/изменить ваш QTH-локатор;
 - NZ: ввести/изменить ваш почтовый индекс;
- NEW — выдается распечатка всех новых файлов, которые были загружены или созданы начиная с последнего раза, когда вы использовали эту команду. Этот список покрывает все области каталога, к которым вам разрешен доступ;
- O — выдает на дисплей различные зарегистрированные и доступные опции удаленного пользователя:
- OL: список доступных языков;
 - OL #: выбор языка;
 - OM: переключение on/off служит для выдачи на дисплей вашей почты дополнительных сообщений при входе в систему;
 - ON #: назначает номер последних трех цифр из номеров сообщений для сокращенного ввода пользователем. Например, ON 37 определит сообщение # 37,000 как основной номер сообщения, а команда R 25 будет означать, что следует читать сообщение #37,025;
 - OP: переключает режим листания on/off;

- OP #: определяет число строк, посланных в режиме листания прежде, чем появится подсказка "Send CR to Continue or A to Abort" в конце страницы;
- OR: переключение между разрешением доступа только для чтения почты и бюллетеней и полной распечаткой почты;
- PG — передает список программ пользователя, доступных для выполнения;
- PS — передает список удаленных средств Server, доступных в настоящее время;
- R — чтение сообщений:
 - RM: читает все имеющиеся сообщения, которые адресованы вам;
 - RN: читает все новые (нечитанные) сообщения, адресованные вам;
 - R msg# msg#: читает до шести сообщений в одном блоке;
- S — передача сообщений или бюллетеней:
 - S callsign: передача сообщения или бюллетеня. Если тип не определен, то предполагается, что это сообщение частное;
 - SP callsign: передается частное сообщение;
 - SB ALL: передать бюллетень, который может читаться *всеми* пользователями в сети;
 - SB ALL @xxx: послать сообщение, которое может читаться *всеми* пользователями сети и которое будет находиться в рубрике xxx;
 - SB ALL @GBR: напечатайте это, чтобы послать сообщение, которое может читаться *всеми* пользователями в Великобритании;
 - SB ALL @DEU: напечатайте это, чтобы послать сообщение, которое может читаться *всеми* пользователями в Германии;
 - SR Msg#: дать ответ на сообщение с номером #. При этом заголовок ответа создается автоматически и начинается с сочетания букв Re: xxx;
 - SC Msg# CALL @ BBS: послать копию сообщения или бюллетеня к определенному callsign;
- T — вызвать Sysop для общения в данное время. Если Sysop недоступен в течение одной минуты, программа выдает подсказку и автоматически возвращается к нормальной работе;
- U — загружает текстовый файл от пользователя к корневом каталоге FBB-DOS;
- V — читает сообщения с заголовками маршрутизации. Используется как и команды R, например: VM или V 1234;
- W — выдает список файлов, которые имеются внутри корневом каталоге FBB-DOS. Эта команда может сопровождаться маской, определяющей

критерии поиска или имя каталога; в последнем случае содержание каталога будет перечислено;

- ❑ X — переключение между "Нормальным" или "Экспертным" состоянием, которое дает короткую или полную подсказку;
- ❑ Y — двоичные передачи должны быть выполнены с использованием YAPP-протокола в оба конца связи (у вас должна быть программа, управляющая YAPP-протоколом в вашем компьютере):
 - YD File: получить бинарный файл от BBS к вам;
 - YU File: передать бинарный файл от вас к BBS. Вы не можете заменять или модифицировать существующий бинарный файл;
 - YZ File: удалить файл из области двоичных файлов;
- ❑ Z — удаляет файл из области файлов FBB-DOS.

Перечисленные команды дают представление об огромных возможностях этой программы. Скачать программу можно в Интернете по адресу <http://www.f6fbb.org> или с множества других зеркальных страниц. Некоторые вспомогательные файлы к этой программе можно взять на прилагаемой к книге дискете.

Надежная работа программ, разработанных F6FBB, признана всеми. Но наряду с этой программой существуют другие, не менее надежные в работе и обладающие иными очень важными достоинствами.

FlexNet работает в сети Packet Radio

Основные принципы работы и состав FlexNet

FlexNet — гибкая модульная система для организации эффективных пакетных радиосетей. Система имеет большой набор различных драйверов, который позволяет вам создать узел или пользовательский терминал. Все уровни работающей в сети FlexNet системы (включая каналы доступа пользователя) сами настраиваются согласно текущему состоянию канала связи. Единственное исключение составляет параметр TXDelay, который устанавливает сам пользователь. Для FlexNet написано большое количество драйверов, которые позволяют создавать высокоэффективные пакетные системы практически на любом оборудовании от звуковых карт до модемов и современных TNC контроллеров.

В настоящее время узлы FlexNet составляют основу всей европейской любительской пакетной сети.

Автором программного обеспечения FlexNet является Gunter Jost (DK7WJ), который сохраняет все авторские права.

Первые идеи для разработки этого программного обеспечения появились в 1987 году, а первая версия FlexNet была разработана DK7WJ в 1988 году. Начиная с 1991 года работа проводилась для MS-DOS, испытывалась только на внутренних сетях и не распространялась между радиолюбителями. Однако после многочисленных просьб любителей в 1994 году был выпущен PC/FlexNet в модульном варианте. Драйвера для PC/FlexNet были разработаны в сотрудничестве с DL8MBT, автором программного обеспечения BayCom. Это позволило сделать PC/Flexnet пригодным для использования как опытными, так и начинающими радиолюбителями.

Цель FlexNet — предоставить эффективное и простое программное обеспечение, которое можно использовать как для простого терминала, так и для построения сложных узлов с BBS, TCP/IP, DXC и т. д.

Программное обеспечение FlexNet может свободно копироваться и использоваться для некоммерческих любительских радиосетей.

В настоящий момент применяются две основные версии FlexNet-драйверов:

- ❑ PC/FlexNet 3.3g — 16-разрядная версия для работы под MS-DOS или Windows 95/98;
- ❑ FLEX32 — 32-разрядная версия для работы под Windows 95/98/ME/2000/NT.

Подробную информацию, программы и наборы драйверов можно взять в Интернете по адресу <http://www.afthd.tu-darmstadt.de/~flexnet/> или на домашней страничке BayCom (<http://www.baycom.org/>), или на файловом сервере Пятигорского радиоклуба <ftp://193.124.120.49/flexnet/>.

Примеры применения FlexNet

Ниже приведены несколько примеров использования FlexNet для совместной работы с некоторыми пакетными программами.

Терминальные программы Packet Radio, которые поддерживают систему команд WA8DED (большинство современных программ), могут успешно работать через звуковую плату SoundBlaster (SB16, SB PRO или совместимую), при этом отпадает необходимость и в TNC, и в модеме. В этом случае на микрофонный вход платы через конденсатор 0,1 мкФ и регулирующий потенциометр подаются сигналы с радиоприемника, а с линейного выхода платы также через конденсатор 0,1 мкФ и регулирующий потенциометр сигналы подаются на микрофонный вход SSB-передатчика. Сигналы управления передатчиком (РТТ) берутся с COM-порта (или LPT, по желанию). Такими программами являются F6FBB 5.15, HostCom, GP и многие другие. Может работать через звуковую плату и очень популярная в нашей стране программа BayCom, вернее, не сама программа, а несколько переработанный аналог этой программы, который имеет наименование BCT.EXE.

Необходимым условием для успешной работы через звуковую плату является наличие пакета программ PC/FlexNet 3.3G, разработанного радиолюбителя-

ми DK7WJ, HB9JNX и группой FLEXNET. Этот пакет включает в себя несколько программ, среди которых имеются и основополагающие программы, и драйверы, и эмуляторы модемов, и эмулятор TNC, и различные утилиты.

Ниже привожу практически используемые командные файлы, по которым можно запускать в работу терминальные пакетные программы. При этом подразумевается, что пакет программ PC/FlexNet 3.3g располагается по адресу D:\FLEX\, а сами bat-файлы находятся в составе пакета терминала, также на диске D:.

Листинг 2.4. Файл appel_f.bat для программы F6FBB 5.15c

```
(начало appel_f.bat файла)
@echo off path=d:\dos;d:\drv;d:\fbb;d:\fbb\system;d:\fbb\bin
@echo off
LH D:\FLEX\FLEXNET 20
if errorlevel 1 goto fin
LH D:\FLEX\SBLAST -b:0x220 -i:5 -d:0 -tc:2 -ts:0x2f8
rem LH D:\FLEX\SB_9k6 -b:0x220 -i:5 -d:0 -tc:2 -ts:0x2f8
if errorlevel 1 goto fin
D:\FLEX\FLEX
D:\FLEX\FSET mode 0 1200c
D:\FLEX\FSET txd 0 15
D:\FLEX\FSET digi RA3XB-3
LH D:\FLEX\TFEMU
cd \fbb\system
rem clr_user
cd \fbb
break off
echo Program is started, use ^C to interrupt within 3 seconds
sleep 3
serv %1 .....
.....
середина файла не меняется и полностью соответствует
стандартной программе appel.bat
.....
.....
:fin
D:\FLEX\FLEX /u
```

```
echo Program is terminated.  
break on  
Echo on  
(конец appel_f.bat файла)
```

Как видно из текста, изменения стандартного файла проводятся только в самом начале и в самом конце. Все добавления выполнены заглавными буквами.

Листинг 2.5. Файл flex_hc.bat для запуска HostCom

```
@echo off  
LH d:\FLEX\FLEXNET 20  
if errorlevel 1 goto error  
LH D:\FLEX\SBLAST -b:0x220 -i:5 -d:0 -tc:2 -ts:0x2f8  
rem LH D:\FLEX\SB_9k6 -b:0x220 -i:5 -d:0 -tc:2 -ts:0x2f8  
if errorlevel 1 goto error  
D:\FLEX\FLEX  
D:\FLEX\FSET mode 0 1200c  
D:\FLEX\FSET txd 0 15  
D:\FLEX\FSET digi RA3XB-3  
LH D:\FLEX\TFEMU  
echo Программа стартовала, используйте ^C для прерывания в течение 3 сек.  
d:\hc\bin\sleep 3  
d:\hc\bin\hc -1  
:error  
D:\FLEX\FLEX /u  
:end
```

Листинг 2.6. Файл fl_bauc.bat для запуска программы ВСТ

```
(Программа ВСТ.EXE и fl_bauc.bat находится в составе пакета D:\FLEX)  
@echo off  
LH FLEXNET 20  
if errorlevel 1 goto error  
LH SBLAST -b:0x220 -i:5 -d:0 -tc:2 -ts:0x2f8  
if errorlevel 1 goto error  
FLEX  
FSET mode 0 1200c  
FSET txd 0 15
```

```
FSET digi RA3XB-3 bct RA3XB:error  
FLEX /u  
:end
```

Листинг 2.7. Файл fl_emu.bat для запуска других программ

```
@echo off  
LH FLEXNET 20  
if errorlevel 1 goto error  
LH SBLAST -b:0x220 -i:5 -d:0 -tc:2 -ts:0x2f8  
if errorlevel 1 goto error  
FLEX  
FSET mode 0 1200c  
FSET txd 0 15  
FSET digi RA3XB-3  
LH TFEMU goto end  
:error FLEX /u  
:end
```

Особенность данного командного файла заключается в том, что после его выполнения все имеющиеся в нем компоненты остаются загруженными резидентно, после чего можно запускать обычным способом любую терминальную программу и, если она поддерживает систему команд WA8DED, то станет работать через звуковую карту. Для выгрузки резидентных элементов следует еще раз запустить fl_emu.bat.

Обратите внимание, что все программы предназначены для работы со скоростью 1200 Бод.

В конце этой главы есть короткое описание лучшей, по моему мнению, региональной пакетной сети нашей страны, приведена схема этой сети. Обратите внимание, что все узлы в этой сети являются Flex-узлами, и это позволяет им свободно взаимодействовать с пакетной сетью Западной Европы. Основную роль при этом играет программа BayCom-Mailbox (BCM).

Программа BCM в роли MBBS

Универсальная почтовая программа BCM (BayCom-Mailbox) предназначена для организации и работы MBBS в любительской пакетной сети. Программа имеет разработанные варианты для операционных систем DOS, Linux, Windows NT. BCM имеет HTTP-, FTP-, POP3-, SMTP-, NNTP-интерфейсы и отличается исключительной надежностью в работе, простотой настройки, хо-

рошим сервисом, высокой скоростью работы. Файловый сервер, кроме YAPP и AUTOBIN, поддерживает новый протокол передачи данных DIDADIT.

Автор программы Florian Radlherr (Flori, DL8MBT) начал разработку BayCom-Mailbox в 1991 году. Много лет дальнейшим совершенствованием программы занимался Dietmar (OE3DZW) и другие радиолюбители. На сегодняшний день наиболее стабильными считаются версии BCM 1.42 и 1.43.

Требования к компьютеру, на котором должна быть установлена программа BayCom-Mailbox:

- для DOS: 386SX, RAM 2—4 Мбайт, HDD от 80 Мбайт;
- для Linux: 486—33 МГц, RAM 8—16 Мбайт, HDD от 250 Мбайт;
- для Windows NT: 486—66 МГц, RAM 32 Мбайт, HDD — 600 Мбайт и более.

Здесь приведены минимальные требования, при которых BCM работает без задержек одновременно с 70 пользователями!

Документацию для пользователя и для системного оператора (Sysop) на русском языке можно почитать (и скачать!) в Интернете по адресу <http://www.hamradio.cmw.ru/>.

Перевод документации к BCM на русский язык очень квалифицированно выполнил Игорь Лаврушов (UA6HJQ), организатор великолепной пакетной сети региона КАВМИНВОД.

Некоторую часть из переведенной на русский язык документации можно взять на прилагаемой к книге дискете. Поэтому я не буду приводить здесь полный перечень команд, используемых программой BayCom-Mailbox. Весь перечень очень большой, составляет более сотни команд, поэтому приведу перечень сокращенный.

Команды пользователя (BayCom-Mailbox версия 1.42n для DOS)

Персональные установки пользователя:

- ALTER — показать все персональные текущие установки:
 - CHECK <опции> — изменить опции для команды CHECK;
 - COMMAND <команда> — установить команду, которая выполняется при соединении;
 - DEFAULT — стереть все данные о себе и перерегистрироваться заново;
 - DIRFORMAT <0 или 1> — установить формат показа даты при командах Read/Dir/List;
 - ECHO <0 или 1> — установить режим ЭХО из Mailbox;
 - FORWARD <mybbs> — установить свой домашний BBS (как MYBBS или NH);
 - FDELAY <минуты> — установить периодичность вызова терминала пользователя;

- `HELPLEVEL 0|1|2` — установить уровень помощи;
 - `IDIR <опции>` — изменить опции для Info-DIR;
 - `ILIST <опции>` — изменить опции для Info-LIST;
 - `IREAD <опции>` — изменить опции для Info-READ;
 - `LF <число>` — разделять строки между абзацами текстов или нет;
 - `LINES <линии>` — количество строк, после которых останавливается листинг;
 - `LOGINPWTYPE` — установить тип пароля baycom/md2/md5/inactive;
 - `NAME <имя>` — установить имя пользователя (как NAME);
 - `NEWCALL <позывн.>` — установить новый позывной, после изменения;
 - `PROMPT <текст>` — изменить стандартную строку приглашения;
 - `PW <пароль>` — установить пароль пользователя;
 - `PWLINE 0|1` — показать приглашение сразу (0), не показывать сразу (1);
 - `READLOCK 0|1|2` — уровень доступа к вашей персональной почте;
 - `REJECT<рубрика>` — установить игнорируемые рубрики;
 - `SPEECH <язык>` — установить язык общения с Mailbox;
 - `SFPWTYPE<пароль>` — установить пароль для User-FWD;
 - `UDIR <опции>` — изменить опции для User-DIR;
 - `UFWD <позывной>` — активизировать User-FWD;
 - `ULIST <опции>` — изменить опции для User-LIST;
 - `UREAD <опции>` — изменить опции для User-READ.
- ☐ `MYBBS <homebbs>` — зарегистрировать ваш домашний BBS;
- ☐ `NH <homebbs>` — как MYBBS или Alter Forward (команда из F6FBB);
- ☐ `NAME <имя>` — зарегистрировать ваше имя.
- Листать и искать сообщения:
- ☐ `CD [<рубрика>]` — изменить текущую рубрику;
- ☐ `CHECK <опции>` — листать бюллетени независимо от рубрик;
- ☐ `BIDLIST [<опция>] <id>` — показать Bulletin-ID (возможны ключи: bid-i);
- ☐ `LIST [<рубрика>] <номер>` — листать сообщения в Mailbox (похожа на DIR);
- ☐ `DIR` — показать все сообщения из текущей рубрики:
- `AFTER <дата>` — показать все бюллетени, пришедшие в указанный день;

- **BOARDS** <рубрика> — показать информацию о рубриках;
- **MESSAGES** <текст> — вывести список всех бюллетеней из Mailbox (очень много!);
- **NEWS** <рубрика> — показать все новые бюллетени;
- **OUTSTANDING** <позывной> — список неотправленной почты для fwd-партнеров;
- **PATH** <позывной> — список fwd-партнеров и их установки из файла fwd.bcm;
- **SENT** <позывной> — показать какие сообщения пришли от указанного fwd-партнера;
- **HOLDUSER** — показать задержанные сообщения пользователей;
- **USERS** — позывные станций, для которых имеется почта;
- **ALL** <позывной> — показать всех пользователей (от которых были сообщения);
- **LOCAL**<позывной> — список пользователей, соединявшихся с Mailbox локально;
- **MSG** <текст> — листать сообщения, находящиеся в User-каталогах.

Передать короткий текст другому пользователю:

- ☐ **CHAT** <позывной> <текст>;
- ☐ **CONVERS** <позывной> <текст>;
- ☐ **MSG** <позывной> <текст>;
- ☐ **TALK** <позывной> <текст>;
- ☐ **WRITE** <позывной> <текст>;
- ☐ **WALL** <текст> — послать короткое объявление сразу всем пользователям.

Уничтожение сообщений:

- ☐ **ERASE** <номер> — удалить сообщение под номером;
- ☐ **PURGE** <номер> — уничтожить сообщение под номером;
- ☐ **UNERASE** — восстановить удаленное случайно сообщение.

Разорвать связь с MBBS:

- ☐ **BYE**;
- ☐ **EXIT**;
- ☐ **QUIT**;
- ☐ **LOGOUT**.

Вызвать подсказку или файл помощи:

- ☐ ? — справка по основным командам, как и команда `HELP`;
- ☐ `CMDLIST` — показать список команд без их описания;
- ☐ `HELP` — список команд первой необходимости;
- ☐ `HELP I` — список всех команд без описания;
- ☐ `HELP CMD` — список команд с кратким описанием;
- ☐ `HELP <команда>` — показать помощь по конкретной команде;
- ☐ `MAN` — помощь, как и команда `HELP`.

Читать сообщения:

- ☐ `READ [<рубрика>] <номер>` — читать сообщение;
- ☐ `HEADER <рубрика> <номер>` — показать только "шапку" сообщения (как `KOPF`);
- ☐ `KOPF <рубрика> <номер>` — показать только "шапку" сообщения (как `HEADER`);
- ☐ `NEXT` — прочесть следующее сообщение.

Послать сообщение:

- ☐ `S <позывн> <bbs> [<#LT>] <заголовок>` — послать сообщение;
- ☐ `S <рубрика> <страна> <заголовок>` — послать сообщение в рубрику;
- ☐ `SB <рубрика>@<страна> <заголовок>` — послать бюллетень в рубрику;
- ☐ `SP <позывной>@ [<bbs>] [<#LT>] <заголовок>` — послать персональное сообщение;
- ☐ `SETLIFETIME <рубрика> <номер> <#LT>` — изменить "время жизни" (LT) вашего сообщения;
- ☐ `REPLY [<опции>]` — послать персональный ответ на последнее прочитанное сообщение;
- ☐ `TRANSFER <позывной1> <номер> <позывной2>` — перенаправить сообщение другому пользователю;
- ☐ `CP` — перенаправить сообщение (как `TRANSFER`);
- ☐ `FORWARD <опции>` — перенаправить сообщение в другой Mailbox;
- ☐ `FOLLOWUP` — послать комментарий как бюллетень (как `COMMENT`);
- ☐ `COMMENT [<опции>]` — прокомментировать бюллетень (как `FOLLOWUP`).

Статистика о работе системы:

- ☐ `SEMAPHORES` — показать используемые данные;
- ☐ `PS` — показать список текущих процессов в MBBS;

- ❑ MEM — показать распределение памяти;
- ❑ STATUS — показать различную статистику:
 - CPU — нагрузка на процессор;
 - FORWARD — статистика по партнерам;
 - LIMITS — информация о свободных ресурсах.

Разные прочие команды:

- ❑ AKTUELL — прочитать файл с актуальной информацией;
- ❑ CONVAT — показать файл CONVAT.BCM (переименование территорий);
- ❑ CRONTAB — просмотреть файл CRONTAB.BCM;
- ❑ TIME — показать дату и время;
- ❑ UNKNOWN — показать потерянные сообщения из файла UNKNOWN.BCM;
- ❑ QTH <опции> — расчет локатора, координат, расстояния, азимута;
- ❑ RUNUTILS — показать список дополнительных команд;
- ❑ VERSION — показать версию программы;
- ❑ INFO — вывести информацию об этом Mailbox;
- ❑ SF — начать форвардинг;
- ❑ SFHOLD — показать почту, которую отвергли FWD-партнеры из файла SFHOLD.BCM;
- ❑ SLEEP <секунды> — остановить Mailbox на указанное количество секунд;
- ❑ REJECT — показать содержимое файла REJECT.BCM;
- ❑ RLOG — почта, от которой отказался этот Mailbox, из файла REJLOG.BCM;
- ❑ F> — перевести Mailbox в режим пользовательского форвардинга;
- ❑ FS — перейти в режим файлового сервера;
- ❑ IMPDEL — удалить файл Import;
- ❑ MAILSERVER — почтовый сервер для рассылки почты определенным группам пользователей;
- ❑ PARAMETER <команда> — показать список установок из файла INIT.BCM;
- ❑ PATH <mbox/addr> — поиск адресов и других данных, как и команда FIND;
- ❑ FIND <mbox/addr> — как и команда PATH.

Информация о пользователях и соединениях:

- ❑ ALTER <позывной> — показать информацию о пользователе;
- ❑ USERS — показать позывные всех радиостанций, соединенных с Mailbox (как и команда FINGER);

- ❑ FINGER <позывной> — показать список соединенных пользователей, как и команда USERS;
- ❑ LOG <опции> <дата> — показать аппаратный журнал (возможны разные ключи);
- ❑ SLOG — показать системные сообщения из файла SYSLOG.BCM;
- ❑ PRIV <...> — последний вход в систему системного оператора;
- ❑ PWLOG <spec> — показать, кто соединялся с Mailbox по паролю;
- ❑ WHOAMI — "кто я такой в этой системе?";
- ❑ QUOTA — показать сколько Кбайт перекачано и сколько еще можно скачать.

Программа BayCom-Mailbox имеет файловый сервер, расположенный в одном из своих каталогов. В этом каталоге находятся многие из самых интересных для любого радиолюбителя программ. BCM дает удаленному пользователю возможность просмотреть имеющиеся в файловом сервере программы и, в случае необходимости, скачать их. Привожу команды для работы с файловым сервером.

- ❑ FILESERV — зайти в файловый сервер, как и по команде FS:
 - BGET <файл> — читать из сервера файл в формате AutoBin;
 - BPUT <файл> — записать на сервер файл в формате AutoBin (если это разрешено программой);
 - CD <путь> — изменить каталог;
 - DIR <путь> — показать список всех каталогов и файлов;
 - GET <файл> — читать из сервера текстовый файл;
 - PATH — показать все доступные диски и каталоги;
 - PUT <файл> — записать на сервер текстовый файл (если это разрешено программой);
 - QUIT — выйти из файлового сервера и вернуться в режим Mailbox;
 - YGET <файл> — считать из сервера файл в YAPP-формате;
 - YPUT <файл> — записать на сервер файл в YAPP-формате (если это разрешено программой).

Практическому применению программы BCM посвящен следующий раздел книги — описание любительской сети КАВМИНВОД.

Любительская сеть КАВМИНВОД

Лучшая региональная любительская сеть

На Северном Кавказе есть удивительный регион — район Кавказских Минеральных Вод (КАВМИНВОД). Природа наградила этот район теплым

климатом, в недрах этого района содержатся целебные минеральные воды, на дне озер — целебные минеральные грязи. Здесь проживают приветливые и доброжелательные люди, при необходимости готовые прийти на помощь тысячам приезжающих на лечение больных людей. Здесь также много энтузиастов-радиолюбителей.

Несколько лет тому назад мне потребовался большой объем радиолубительской информации, которую можно было получить только через пакетную сеть. Нужно было выбрать такую "home BBS", через которую было бы удобно эту необходимую информацию получать. Дело в том, что в то время самая близкая пакетная BBS находилась от меня на расстоянии 350 километров. О работе на УКВ речи быть не могло, приходилось рассчитывать только на коротковолновые диапазоны. Начал поиски с диапазона 3,5 МГц. BBS было много, но помех они создавали сами себе тоже огромное количество. Диапазон 14 МГц оказался более удобным. После непродолжительной работы с каждой из нескольких BBS, выбор пал на BBS с позывным RK6HWW. Станция работала из г. Пятигорска Ставропольского края, принадлежала Пятигорскому городскому радиоклубу, системным оператором станции был Игорь Лаврушов (UA6HJQ).

С тех пор прошло немало времени, но у меня ни разу не появилось желание сменить "home BBS". Так чем же меня привлекает работа с этой станцией?

- ❑ Станция RK6HWW является составной частью региональной радиолубительской пакетной сети, информацию для которой собирает целый "коллектив" из нескольких радиостанций, используя при этом и сети AMPR, и спутниковую связь, постоянный обмен информацией с многими отечественными и зарубежными BBS.
- ❑ Станция имеет два канала с фиксированными частотами на КВ-диапазонах, что делает ее привлекательной для корреспондентов, не имеющих возможности работать в УКВ-пакетной сети.
- ❑ Организаторы и руководители станции не останавливаются на достигнутом, постоянно совершенствуют и оборудование, и программное обеспечение, стараясь соответствовать уровню лучших западных станций. Использование узлов FlexNet позволяет сети быть полностью интегрированной с Всемирной любительской радиосетью.

О привлекательности этой сети, в смысле получения информации, говорит тот факт, что ее услугами пользуются 76 любительских радиостанций из других регионов. В том числе и принадлежащая мне любительская радиостанция.

Схема радиолубительской сети КАВМИНВОД приведена на рис. 2.3.

Радиолубители из европейской части России и областей Западной Сибири могут работать с RK6HWW на частоте 7,029 МГц (диапазон 40 метров) или на частоте 14,088 МГц (диапазон 20 метров) в зависимости от наличия прохождения в нужную сторону. Станция RK6HWW является узлом (BQ-

NODE) и позволяет соединенному с ней пользователю либо подключиться к мощному серверу RW6HQN, либо связаться с другим абонентом этого узла. При этом узел имеет связь с сервером и другими компонентами сети на частотах диапазона 432 МГц.

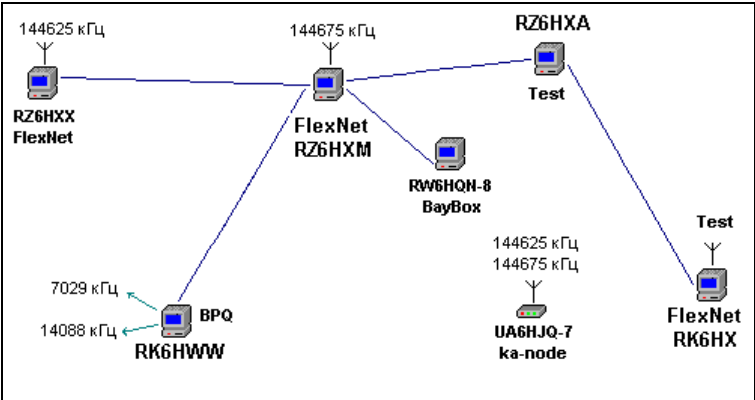


Рис. 2.3. Схема любительской радиосети КАВМИНВОД

Радиолюбители городов Ставрополь, Невинномысск, Черкесск, Нальчик, Армавир, Владикавказ и многих других городов Северного Кавказа, не относящихся к району КАВМИНВОД, могут работать с сетью на УКВ через входящие в сеть радиостанции RZ6HXX, RZ6HXM и RK6HX на частоте 144,625 МГц или 144,675 МГц (УКВ-диапазон 2 метра). Или через узел UA6HJQ-7 на частотах 144,625 и 144,675 МГц.

Особо следует отметить наличие в составе сети AMPR-узла, который установлен на RZ6HXA. Выйти на этот узел со стороны Интернета можно, используя адрес <http://baybox.narod.ru/>.

Войдя на указанную страничку узла, следует выбрать **HTTP ВХОД**. Для вас откроется окно ввода информации **login**, в которое вы должны ввести позывной **rz6hdd**, а на запрос **password** ввести пароль — **test1234**.

Команды AMPR узла BayBOX

В табл. 2.22 приведены команды, используемые в узле BayBOX.

Таблица 2.22. Команды узла BayBOX

Команда	Описание команды	Пример команды
ALTER FWD	Зарегистрировать ваш позывной	A F RA3XB
ALTER NAME	Зарегистрировать ваше имя	A N Игорь

Таблица 2.22 (окончание)

Команда	Описание команды	Пример команды
ALTER	Показать ваши персональные уставки	A
CHECK	Перечень 20-ти самых свежих сообщений	C -20
DIR BULLETIN	Показать все имеющиеся рубрики	D B *
LIST	Читать 10 новых сообщений из WSEM	L WSEM -10
READ	Читать сообщение № 4 из рубрики WSEM	R WSEM 4
REPLY	Послать ответ на прочитанное сообщение	REP
SEND	Послать персональное сообщение	S UA6HJQ Письмо
ERASE	Удалить прочитанное сообщение под № 5	E 5
QUIT	Закончить сеанс связи с MBBS	Q или B
INFO	Прочитать файл с информацией о системе	I
TALK	Послать короткий текст другой станции	T RK6HX Привет, я жду
USERS	Выдать список всех соединенных позывных	U
RENUTILS	Показать список внешних команд и серверов	RUN
FILESURE	Зайти в файловый сервер	FS
HELP CMDLIST	Список всех команд с кратким описанием	H CMD
HELP<команда>	Подробная помощь по конкретной команде	H LIST или H SB
HELP INDEX	Показать список всех команд без описания	H I

Для того чтобы стать постоянным пользователем радиолобительской пакетной сети КАВМИНВОД со стороны Интернета через узел BagBOX, следует написать письмо системному оператору узла по адресу **rz6hxa@rz6hxa.sta.rus.eu** с просьбой открыть доступ и выдать пароль. Письмо следует отправить по электронной почте.

Получить большое количество информации по организации сети КАВМИНВОД можно в Интернете по адресу **<http://www.hamradio.cmw.ru/>**.

Новые виды цифровой связи

PSK31 — работа с клавиатурой

PSK31 является цифровым видом связи, который способен работать при очень незначительной полосе излучаемой частоты. Программа существует во многих вариантах, наиболее распространенными являются варианты программы, работающие под управлением Windows 95 и выше, используя вместо модема звуковую карту компьютера. PSK31 разрабатывался как способ связи между двумя или несколькими коротковолновиками при очень малой, по сравнению с другими видами цифровой связи, скорости передачи информации. Скорость передачи в режиме PSK31 всегда постоянна и равна 31,25 Бод, что соответствует скорости печатания на клавиатуре радиолюбителем средних способностей. При этом сигнал несущей частоты имеет постоянную величину частоты, а токовые и бестоковые посылки отличаются друг от друга только поворотом фазы на 180 градусов.

Идея такого вида связи принадлежит SP9VRC (Pawel Jalocho) и была реализована в середине 90-х годов в разработанной им программе SLOWBPSK. Далее разработку этого вида связи и программы для обработки сигналов PSK31 выполнил Peter Martinez (G3PLX), "отец" и создатель AMTOR.

Разработчики этого вида связи исходили из того, что такие виды связи, как RTTY, а тем более AMTOR и PACTOR, работающие при условии синхронной связи, не могут удовлетворять потребности радиолюбителей в коротком обмене информацией с быстрыми переключениями.

Привлекательным условием также является очень узкая полоса излучаемых частот и возможность еще раз "поколдовать" над созданием новых кодов для символов. С момента создания кодов Морзе многим хочется создать какие-то свои, ну уж очень необыкновенные, коды. Была создана новая таблица кодирования символов, отличная от таблиц кодирования RTTY и AMTOR. При создании этой таблицы наиболее употребляемые, с точки зрения разработчиков, символы создавались из меньшего числа посылок, а менее употребляемым предназначалось самое большое число посылок.

Вполне естественно, что буквы русского алфавита, как совсем не употребляемые разработчиками, создавались из большего числа посылок. Например, если одно и то же слово передать латинскими и русскими символами, то передача русскими символами будет почти в два раза длиннее.

PSK31 имеет одну очень неприятную проблему. Дело в том, что изменение фазы несущей частоты на противоположную равносильно мгновенному выключению одного передатчика и столь же мгновенному включению другого передатчика, фаза частоты которого изменена на 180 градусов. Такие резкие изменения создают в эфире сильную помеху в виде громкого шелчка на очень большом частотном участке.

Радиолюбители, использующие PSK31, должны знать о наличии этой проблемы и всегда принимать меры для предотвращения помех соседним радиостанциям.

Одной из таких мер является правильный выбор величины PSK31-сигнала, подаваемого на микрофонный вход передатчика. Эта величина должно быть как можно меньше, чтобы уменьшить помехи. Peter Martinez предлагает выполнять соединение компьютера с радиостанцией по схеме, изображенной на рис. 2.4. Обратите внимание, что схема предусматривает снижение сигнала в 10 раз. Малая величина подаваемого сигнала в то же время должна быть достаточной для обеспечения расчетного режима работы передатчика.

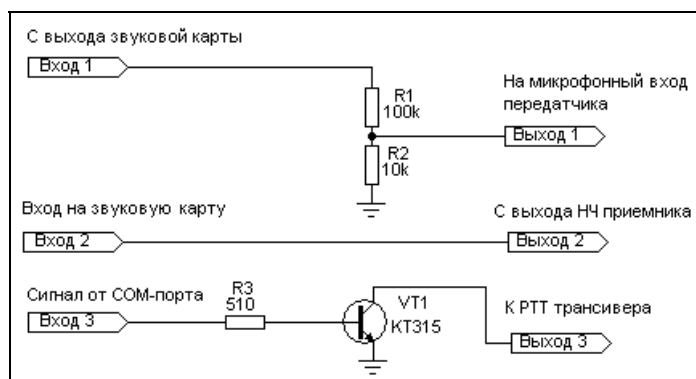


Рис. 2.4. Схема соединения компьютера с радиостанцией

Большую роль при приеме сигналов PSK31 играет настройка фильтров. Достаточно узкополосный фильтр позволяет избавиться от помех соседних радиостанций.

PSK31 существует в двух видах — BPSK и QPSK.

BPSK осуществляет изменение фазы модулирующего сигнала только от 0 градусов до 180 градусов и наоборот.

В декабре 1997 года был обнародован новый вид PSK31, получивший название QPSK. В этом виде кроме указанного выше изменения фазы от 0 до 180 градусов (и наоборот), было добавлено одновременное изменение фазы от 90 до 270 градусов (и наоборот). Это можно представить, если предположить, что имеется два одновременно работающих передатчика, у первого из них фаза изменяется от 0 до 180, а у второго — от 90 до 270 градусов.

Разработчики уверяют, что при приеме QPSK имеет лучшее на 3 дБ соотношение "сигнал/шум". И при этом не возрастает число ошибок.

Считается, что программа для PSK31 должна иметь в своем составе и BPSK, и QPSK, чтобы можно было пробовать и один, и другой виды в поисках лучшего варианта для приема сигналов в данной ситуации.

Примененное в PSK31 кодирование символов имеет следующие особенности:

- ☐ коды символов состоят из разного числа посылок;
- ☐ каждый символ отделен от другого сочетанием из двух нулей (00);
- ☐ ни один код символа не содержит в своем составе двух нулей подряд;
- ☐ код каждого символа начинается и заканчивается единицей (1).

Если в асинхронных видах связи (RTTY) код каждого символа начинается со стартового бита и заканчивается стоповым битом, то в PSK31 роль такого стартового элемента выполняет сочетание из двух рядом стоящих нулей. Переменная длина кодов вызвана желанием уменьшить время передачи текстов на английском языке. В табл. 2.23 приведены коды первых 128 символов из кодовой таблицы IBM.

Таблица 2.23. Коды символов PSK31

Символ	Код символов	Символ	Код символов
NUL	1010101011	A	1111101
SOH	1011101101	B	11101011
STX	1011101101	C	10101101
ETX	1101110111	D	10110101
EOT	1011101011	E	1110111
ENQ	1101011111	F	11011011
ACK	1011101111	G	11111101
BEL	1011111101	H	101010101
BS	1011111111	I	1111111
HT	11101111	J	111111101
LF	11101	K	101111101
VT	1101101111	L	11010111
FF	1011011101	M	10111011
CR	11111	N	11011101
SO	1101110101	O	10101011
ST	1110101011	P	11010101
DLE	1011110111	Q	111011101
DC1	1011110101	R	10101111
DC2	1110101101	S	1101111
DC3	1110101111	T	1101101

Таблица 2.23 (продолжение)

Символ	Код символов	Символ	Код символов
DC4	1101011011	U	101010111
NAK	1101101011	V	110110101
SYN	1101101101	X	101011101
ETB	1101010111	Y	101110101
CAN	1101111011	Z	101111011
EM	1101111101	[1010101101
SUB	1110110111	\	111110111
ESC	1101010101]	111101111
FS	1101011101	^	111111011
GS	1110111011	-	1010111111
RS	1011111011	.	101101101
US	1101111111	/	1011011111
SP	1	a	1011
	111111111	b	1011111
“	101011111	c	101111
#	111110101	d	101101
\$	111011011	e	11
%	1011010101	f	111101
&	1010111011	g	1011011
'	101111111	h	101011
(11111011	i	1101
)	11110111	j	111101011
*	101101111	k	10111111
+	111011111	l	11011
,	1110101	m	111011
-	110101	n	1111
.	1010111	o	111
/	110101111	p	11111
0	10110111	q	110111111
1	10111101	r	10101

Таблица 2.23 (окончание)

Символ	Код символов	Символ	Код символов
2	11101101	s	10111
3	11111111	t	101
4	101110111	u	110111
5	101011011	v	1111011
6	101101011	w	1101011
7	110101101	x	11011111
8	110101011	y	1011101
9	110110111	z	111010101
:	11110101	{	1010110111
;	110111101		110111011
<	111101101	}	1010110101
=	1010101	~	1011010111
>	111010111	DEL	1110110101
?	1010101111		
@	1010111101		

Реальные программы могут иметь коды всех 256 символов из кодовой таблицы IBM, но при этом коды второй половины могут иметь какие-то различия. Дело в том, что во второй половине кодовой таблицы располагаются национальные символы многих государств, которые имеют значительные различия в написании.

Первое появление нового вида связи вызвало бурный восторг у радиолюбителей. Прочувствовать это можно, просмотрев материалы на страничке Интернета по адресу <http://www.ua1aco.narod.ru/> или <http://www.psk31.newmail.ru/>. Много ссылок на другие страницы Интернета находятся по адресу <http://www.ua1aco.narod.ru/arhiv.htm>.

Первой программой для работы PSK31 была программа PSK31SBW. К настоящему времени имеется уже несколько версий этой программы. Особенно привлекательна эта программа тем, что к ней имеется отлично выполненный файл помощи на русском языке. Создан этот документ в сентябре 1999 года радиолюбителями RV3DBL и RA3DOA.

В настоящее время в Интернете можно найти множество вариантов программ для работы PSK31, но я настоятельно советую начать освоение этого вида связи именно с программы PSK31SBW 1.08.

Программа PSK31SBW

Что такое PSK31SBW

Рабочий экран программы состоит из большого окна принимаемой информации, ниже которого располагается довольно узкое окно для печатания команд. В левом нижнем углу располагаются индикаторы настройки — выполненный в виде круга *индикатор фазы*, а ниже — узкая полоса индикатора, который получил название "*Водопад*".

В присутствии только фоновых шумов на индикаторе фазы видны лишь беспорядочные красные всполохи и на основном экране ничего не выдается. Каждый раз с обнаружением синхросигнала PSK31 на индикаторе фазы появляется вертикальная линия вниз из центра, а на дисплее "Водопад" — пара близко расположенных друг к другу линий по центру дисплея (при точной настройке на сигнал).

Справа от индикаторов располагаются кнопки управления шумоподавителем и частотами приема и передачи.

Если в поле **Squelch** (Шумоподаватель) отмечен режим **Auto** (Автоматический), то изображение на индикаторе фазы из красного становится желтым, индицируя тем самым обнаружение синхросигнала. С началом передачи текста линия на индикаторе фазы станет показывать скачки фазы, а на дисплее "Водопад" спектр станет "промодулированным", и в основном окне экрана появится текст. Если сигнал слишком слаб для переключения шумоподавителя и вы все же хотите что-то принимать, то можно переключить **Squelch** (Шумоподаватель) в положение **Off** (Отключен). При этом изображение сигнала на индикаторе фазы будет постоянно желтым и программа будет пытаться обрабатывать (преобразовывать в символы — "принимать") даже шумы в полосе приема. Функциональная клавиша <F2> переключает **Squelch** (Шумоподаватель) между состояниями **Auto** (Автоматический) и **Off** (Отключен).

Еще правее располагается окно настройки *частоты приема (Rx Freq)* и *частоты передачи (Tx Freq)*. Далее следуют кнопки:

- ☐ **TX (on/off)** — передача (включено/отключено);
- ☐ **CWID** — передать свой позывной телеграфом;
- ☐ **CQ** — передать текст общего вызова;
- ☐ **Tune** — выдать на передачу чистый тоновый сигнал.

Вы можете вручную подстраивать частоту приема программы, удерживая в строго вертикальном положении отображение сигнала на индикаторе фазы, или же вы можете задействовать систему АПЧ (AFC) программы, которая будет осуществлять точную автоматическую подстройку на сигнал в периоды срабатывания шумоподавителя и отслеживать любой (медленный) дрейф частоты передаваемого сигнала.

В основной массе PSK31-связей задействуется режим BPSK, при котором небольшая вертикальная полоска, отображающая сигнал на индикаторе фазы, перемещается снизу вверх и наоборот. Но если вдруг на индикаторе фазы появятся перекрещивающиеся линии вместо одной вертикальной, то это означает присутствие сигнала QPSK. Следует переключиться в режим работы QPSK.

Любые сбои в приеме данных проявляются в виде ошибок в декодируемом тексте. Программа PSK31SBW способна также работать в режиме коррекции ошибок за счет передачи избыточных данных. При этом вместо непосредственной передачи избыточных битов и соответствующего снижения скорости передачи текста за каждые те же 32 мс передаются два бита данных с использованием четверичной фазовой манипуляции несущей вместо двоичной. Соответственно на индикаторе фазы отображение сигнала принимает вид креста вместо переключающейся линии. Это и есть режим QPSK.

Итак, если вы видите крест на индикаторе фазы, то принимаемый вами сигнал является сигналом QPSK. Но если при этом программа находится в режиме BPSK, то на экран будут выводиться только всякие случайные символы. Для переключения в режим QPSK воспользуйтесь меню **Mode** (Режимы) программы или функциональной клавишей <F1> на клавиатуре. Состояние синхронизации для режима QPSK аналогично таковому для режима BPSK и на индикаторе фазы сигнал синхронизации режима QPSK по-прежнему отображается линией из центра вниз. Автоматическое переключение режимов программы не предусмотрено: вы должны сами идентифицировать сигнал (на индикаторе фазы) и вручную установить режим BPSK/QPSK. Для режима QPSK характерен меньший коэффициент ошибок в условиях федингов, но настройка на сигналы требует большей аккуратности. Маркеры по кругу индикатора фазы (два или четыре, светлые сегменты окружности) будут напоминать вам о текущем режиме работы программы. И если вы установили режим QPSK для приема сигнала, который на индикаторе фазы имеет вид перекрещивающихся линий, но на экран по-прежнему выдаются бессмысленные символы, то возможно, что передача сигнала ведется в неверной полярности.

При приеме сигналов BPSK из эфира использование конкретно верхней или нижней боковой полосы частот трансивера не имеет значения. Однако это принципиально важно для сигналов QPSK. По умолчанию программа PSK31SBW настроена на корректный прием сигналов QPSK с использованием *верхней* боковой полосы частот. Однако в меню **Setup** (Установки) программы при помощи опции **Inverted QPSK** (Инверсия сигналов QPSK) вы можете изменить эту установку программы на противоположную, если захотите правильно принимать сигнал QPSK с использованием нижней боковой полосы трансивера. Проверьте, что ваш приемник использует правильную боковую полосу частот и состояние опции **Inverted QPSK** (Инверсия сигналов QPSK) в меню **Setup** (Установки) программы.

Основные приемы работы

Для приема сигналов PSK31 вам потребуется SSB-приемник. Реальная частота, на которой вы будете принимать сигнал, смещена относительно частоты настройки приемника (который показывает частоту подавленной несущей SSB-сигнала). Например, для того чтобы настроиться на сигнал PSK31 на частоте 14 070,15 кГц, вы можете установить в приемнике режим приема верхней боковой полосы и настроить его на частоту 14 069,0 кГц. В результате на выходе приемника будет слышен звуковой сигнал PSK31 с частотой 1,15 кГц.

В режиме PSK31 необходимо настраиваться на частоту сигнала с точностью до единиц герц, однако шаг перестройки большинства современных трансиверов обычно слишком груб для этого. По этой причине программа для приема сигналов PSK31 имеет узкополосный НЧ-фильтр, который может перестраиваться в звуковом диапазоне (30—3 001 Гц с шагом в 1 Гц). В упомянутом выше примере звуковой фильтр программы PSK31SBW должен быть настроен на частоту 1150 Гц с помощью окна **Rx Freq** (Частота приема) программы. Вы можете щелкнуть мышкой в этом окне и изменять номинал частоты или непосредственно (вводя номинал частоты с клавиатуры), или, используя кнопки вверх/вниз рядом с окном, или левую и правую клавиши управления курсором на клавиатуре.

Существует и другой способ подстройки частоты приема программы. Вы можете поместить курсор мышки по центру спектра сигнала, расположенного не по центру дисплея "Водопад", и щелкнуть левой кнопкой мышки — это приведет к автоматическому переносу спектра сигнала в центр дисплея.

Настройка на сигнал PSK31 заметно отличается от настройки на любой другой сигнал, например, на сигнал RTTY, по двум причинам. Во-первых, сигнал PSK31 иногда представляется одиночным тоном в центре дисплея "Водопад", а иногда и двумя тонами по обе стороны от центра. Если вы настроитесь со смещением в 15 Гц от центра, то один из двух тонов будет находиться в центре, и другой — снаружи полосы приема, так что вы можете получить неверную точку настройки даже при точной настройке на один из тонов. Во-вторых, настройка отличается также и потому, что спектр частот сигнала настолько узок, что прием сигнала PSK31 возможен при отстройке всего лишь на 50 Гц от другого сигнала, и при таком небольшом разnose частот ухо человека уже не способно достаточно хорошо разделить эти два сигнала и обеспечить вам правильную настройку.

Необходимо освоить несколько непривычную технологию настройки на сигнал. Для упрощения этой процедуры программа PSK31SBW снабжена двумя индикаторами настройки — дисплеем "Водопад" и индикатором фазы несущей частоты сигнала. Чтобы настроиться на сигнал PSK31, вы должны научиться на слух настраиваться (приемником трансивера) с точностью до 250 Гц от действительной частоты сигнала, а затем с помощью дисплея "Во-

допад" — с точностью до 15 Гц. Наконец, с помощью индикатора фазы — с точностью до 1—2 Гц, необходимой для нормального приема сигналов PSK31. Кроме того, для того чтобы вы могли своевременно переключить режим работы программы, индикатор фазы сообщает вам о виде принимаемого сигнала (BRSK или QPSK) и дает достаточно удобную и точную оценку качества связи.

Дисплей "Водопад" в левой нижней части экрана программы подобен миниатюрному анализатору спектра, в котором уровни сигналов в полосе ± 250 Гц от центральной частоты настройки отображаются яркостью их откликов. Когда вы будете перестраиваться по частоте приемником трансивера или с помощью окна **Rx Freq** (Частота приема) программы, принимаемые сигналы будут смещаться по окну дисплея. Если отклик сигнала находится в правой части дисплея, то его частота выше частоты приема программы. Изображение спектра на дисплее постоянно прокручивается вниз подобно водопаду, благодаря чему вы можете наблюдать сигналы в полосе частот анализатора в течение последних трех секунд. Чтобы настроиться на сигнал PSK31, перестраивайте приемник до тех пор, пока сигнал не окажется как можно ближе к центру дисплея.

Если поперек дисплея "Водопад" вы увидите красные горизонтальные линии, то это свидетельствует о том, что уровень сигнала на входе звуковой платы слишком высок. Уменьшите входное усиление звуковой платы с помощью программы микшера или воспользуйтесь аттенуатором для снижения уровня сигнала от приемника.

Дисплей "Водопад" сам по себе уже позволяет настраиваться достаточно точно, чтобы обеспечить правильный прием и вывод символов на экран. Однако индикатор фазы дает еще более точную оценку состояния настройки и качества сигнала и служит средством распознавания вида сигналов (BPSK или QPSK), чтобы можно было включить соответствующий режим работы программы и правильно обрабатывать сигнал.

Индикатор фазы показывает относительное изменение фазы несущей частоты манипулированного сигнала и амплитуду последнего. Обычная (не манипулированная) несущая не имеет скачкообразных фазовых сдвигов и отображается вертикальной линией вверх из центра индикатора (при точной настройке), при этом чем мощнее сигнал, тем длиннее эта линия. Сигнал синхронизации в режиме PSK31, являющийся несущей, проманипулированной относительным сдвигом фазы частоты на 180 градусов, имеет вид желтой вертикальной линии вниз от центра индикатора (при точной настройке).

Если настройка на сигнал неточная, то линия на индикаторе фазы будет смещаться по кругу относительно своего вертикального положения. Так, если частота сигнала немного выше центральной частоты полосового НЧ-фильтра программы, то линия смещается по кругу вправо (по часовой стрелке). Вы можете установить линию точно в вертикальное положение

клавишами влево/вправо управления курсором на клавиатуре или щелчками мышкой по кнопкам вверх/вниз рядом с окном **Rx Freq** (Частота приема).

Если вы настроитесь на немодулированную несущую со сдвигом по частоте на 15 Гц, то линия на индикаторе фазы развернется на 180 градусов из положения вертикально вверх от центра в положение вертикально вниз от центра — точно так же, как при точной настройке на сигнал синхронизации, однако на дисплее "Водопад" вы сможете заметить разницу в том, что сигнал синхронизации состоит из двух тонов по одному в каждую сторону (от центра дисплея), а несущая при неточной настройке на нее — только из одного (смещенного влево или вправо от центра дисплея). Аналогичным образом, сигнал синхронизации, смещенный на 15 Гц при неточной настройке, выглядит на индикаторе фазы точно так же, как немодулированная несущая при точной настройке на нее, однако на дисплее "Водопад" вы сможете заметить разницу в сигналах. Если вы настроитесь на сигнал со смещением более 15 Гц, то сигнал перестанет отображаться на индикаторе фазы, и последний переключится на отображение шумов, но вы по-прежнему сможете наблюдать спектр сигнала на дисплее "Водопад".

Хорошую помощь в практическом освоении настройки на сигнал окажут вам эксперименты с немодулированными несущими. Вращайте ручку основной настройки точно по несущей, после чего окончательно подстройтесь на нее, нажимая клавиши управления курсором до тех пор, пока линия на индикаторе фазы не займет вертикальное положение из центра вверх. Мышкой старайтесь щелкать так, чтобы линия на индикаторе фазы сразу устанавливалась точно вертикально вверх из центра индикатора.

Когда вы проделаете то же самое при настройке на сигнал PSK31, то, щелкнув мышкой по центру сигнала, вы вместо единственной линии вверх от центра индикатора фазы увидите или линию из центра вниз если это сигнал синхронизации, или линию по диаметру индикатора (для BPSK) если в этот момент передаются символы текста.

При приеме только сигнала синхронизации (когда корреспондент не нажимает на клавиши) на индикаторе фазы вместо линии, характеризующей фазу сигнала, всегда появляется двузначный цифровой дисплей, показывающий относительное (в дБ) ослабление побочных составляющих спектра сигнала. На этот дисплей выдается оценка качества принимаемого сигнала (PSK31).

В режиме BPSK система шумоподавителя ориентируется по "чистоте" сигнала на индикаторе фазы, а в режиме QPSK — по числу коррекций ошибок, выполняемых декодером. Следует иметь в виду, что если вы в режиме QPSK программы настроитесь на сигнал BPSK или инвертированный сигнал QPSK, или, находясь в режиме BPSK, настроитесь на сигнал QPSK, то шумоподаватель в момент передачи текста не сработает. Шумоподаватель сра-

батывает с задержкой около 1,5 сек. Кроме того, шумоподаватель срабатывает при обнаружении синхросигнала и отключает прием, если "уцепится" за окончание передачи сигнала PSK31. Поэтому никогда не будут потеряны первые передаваемые символы и не будет печататься "мусор" на экране (по завершении передачи), т. к. передача сигнала PSK31 всегда начинается с выдачи синхросигнала и заканчивается специальным сигналом окончания передачи.

Система АПЧ (AFC) будет отслеживать этот дрейф. В обоих случаях изменение частоты приема отображается в окне **Rx Freq** (Частота приема) программы. Когда вы захотите ответить на этот сигнал, то убедитесь, что задействована опция **Net** (Сеть) в окне **Tx Freq** (Частота передачи) программы. Если это так, то с началом ввода символов и включения передатчика частота последнего будет точно соответствовать предшествующей частоте приема. Использование опции **Net** (Сеть) соответствует работе трансивера с выключенной расстройкой. Вы всегда будете отвечать на частоте корреспондента, даже если ваша станция или станция корреспондента "плывет" по частоте, но у каждого из вас задействована система АПЧ (AFC) для компенсации такого дрейфа. Функциональная клавиша <F3> включает/отключает систему АПЧ (AFC). Вы можете отключить эту систему, если желаете вручную подстраиваться на частоту сигнала.

Единственно, когда не следует задействовать опцию **Net** (Сеть), так это тогда, когда вы работаете на преднамеренно разнесенных рабочих частотах.

Если вам заранее известно наличие подобного смещения рабочих частот приема и передачи вашего трансивера, то вы не должны в своей работе использовать опцию **Net** (Сеть). Это обусловит уверенность в том, что по крайней мере другая станция сможет отвечать на вашей частоте, и любая третья станция сможет легко принимать вас обоих без дополнительной подстройки. Переключение опции **Net** (Сеть) можно осуществить и функциональной клавишей <F4>.

Если вы хотите перестроиться на соседний канал в пределах боковой полосы частоты трансивера, то можете щелкнуть мышкой в окне **Rx Freq** (Частота приема) и затем с помощью клавиатуры изменить частоту. Это также применимо и в отношении окна **Tx Freq** (Частота передачи), если не задействована опция **Net** (Сеть), причем сделать это можно даже в процессе передачи. В обоих случаях действительное изменение частоты происходит лишь при выводе курсора из окна. Для этого можно нажать клавишу <Enter> или щелкнуть мышкой в окне передачи, или нажать клавишу <Tab>.

Кроме того, вы можете сменить частоту в пределах ± 250 Гц, щелкнув мышкой по дисплею "Водопад" в области свободной частоты, перестроив тем самым приемник на данную рабочую частоту. В этой ситуации для обеспе-

чения синхронной смены частоты передачи достаточно задействовать опцию **Net** (Сеть).

Так же как и для организации передачи кодом Морзе своего позывного, по завершении цикла передачи путем щелчка мышью по кнопке **CWID** (или нажатия на клавишу <F6>) можно воспользоваться пунктом меню **Mode** (Режимы) для выбора в нем режима **Send CW** (Передача телеграфом) передачи кода Морзе. После этого просто вводите символы с клавиатуры. Используйте кнопку **Tx Off** в программе или функциональную клавишу <F5> для возврата в режим приема по окончании передачи.

Для передачи общего вызова с последующей передачей в конце вашего позывного кодом Морзе, щелкните мышкой по кнопке **CQ** программы или нажмите клавишу <F7>.

Для экстренного прерывания передачи нужно щелкнуть мышкой по кнопке **Tx Off** или нажать клавишу <F5>.

Передача сигналов PSK31

Для того чтобы переключить программу PSK31SBW в режим передачи, достаточно простого нажатия любой символьной клавиши. Лучше всего пользоваться клавишей <Enter>, т. к. при этом на принимающей стороне происходит переход на новую строку выдачи символов на экран. Если перед этим вы правильно установили уровень сигнала, подаваемого на вход передатчика с выхода звуковой платы, то передатчик начнет излучать чистый двухтональный сигнал, причем индицируемая выходная мощность должна быть на уровне приблизительно 50% от максимальной. В отличие от сигнала FSK с огибающей постоянной амплитуды, ограничение которой в процессе передачи ничем не грозит, сигнал PSK31 имеет изменяющуюся амплитуду, всякое искажение которой при передаче неминуемо приводит к появлению нежелательных побочных излучений.

С началом передачи текста мощность передатчика будет колебаться до отметки максимального уровня. Изображение на индикаторе фазы станет зеленым и печатаемые символы по мере их ввода будут появляться в окне передачи. Вы не сможете корректировать уже введенные ошибочные символы до того, как они будут переданы. Нажатие клавиши возврата на один символ (<Backspace>) приводит к выдаче в канал соответствующего управляющего кода, благодаря которому стирается копия этого символа и на приемной стороне, но только в пределах (до начала) текущей строки передачи. Коды ANSI расширенного набора символов, если они есть на клавиатуре, например, *umlauts* и другие (национальные) письменные знаки, также могут быть переданы, но правильно приняты они могут быть лишь при условии, что оба корреспондента используют один и тот же расширенный набор символов ANSI. По мере передачи текста (которая может притормаживаться, если

вы быстро вводите символы) он появляется в основном окне. Не забывайте, что эффективность режима PSK31 основана в некоторой степени и на том, что для наиболее часто встречающихся символов текста используются наиболее короткие коды. Это означает также, что текст из символов нижнего регистра передается быстрее текста, составленного из символов верхнего регистра. По этой причине передача текста заметно замедлится, если вы нажмете клавишу <Caps Lock> и станете передавать только символы верхнего регистра. Если вам неудобно печатать в смешанном (верхнем и нижнем) регистре, то передавайте весь текст символами только нижнего регистра. (Инверсия сигналов QPSK).

Как и в процессе приема, при передаче сигнала BPSK выбор боковой полосы частот трансивера не имеет значения, но при передаче сигнала QPSK это следует учитывать. По умолчанию программа PSK31SBW настроена на передачу сигналов QPSK с использованием верхней боковой полосы частот, но опцией **Inverted QPSK** (Инверсия сигналов QPSK) меню **Setup** (Установки) программы вы можете изменить эту настройку программы на противоположную, если захотите правильно передавать сигнал QPSK с использованием нижней боковой полосы частот трансивера. Применяйте опцию **Inverted QPSK** (Инверсия сигналов QPSK) в меню **Setup** (Установки) программы, если вы вынуждены использовать именно нижнюю боковую полосу частот.

В режиме передачи нажатие функциональной клавиши <F5> на клавиатуре или если необходимо, то функциональной клавиши <F8> может быть использовано для инициирования передачи чистого тонального сигнала для отладочных целей, а нажатие клавиши <F7> приведет к передаче текста CQ (в режиме PSK31) с последующей передачей кодом Морзе вашего позывного.

Текст, принятый на экране, может быть сохранен в файле. Выберите меню **File** (Файл) и в нем пункт **Log to File** (Запись в файл), а затем в окне выбора файла или подтвердите предлагаемое имя файла PSK31.LOG, или введите любое другое имя по вашему выбору. По мере приема текста последний будет одновременно с выдачей на экран записываться в указанный файл. Если после этого вы снова откроете меню **File** (Файл) программы, то увидите, что пункт **Log to File** (Запись в файл) этого меню помечен "птичкой" — это свидетельствует о наличии в данный момент файла, открытого для сохранения принимаемого текста. Для закрытия этого файла достаточно еще раз выбрать в меню **File** (Файл) пункт **Log to File** (Запись в файл). Если впоследствии вы снова задействуете для сохранения принимаемого текста тот же самый файл, то новый текст будет просто добавляться в конец уже существующего файла. Так будет происходить до тех пор, пока вы не удалите этот файл за ненадобностью.

Передача текстовых файлов

Если выбрать пункт **Send File** (Передача файла) в меню **File** (Файл), то заранее подготовленный файл с текстом может быть передан по эфиру. При этом появляется окно выбора файла, в котором вы сможете выделить требуемый файл. По умолчанию в окне выбора выдается перечень файлов с расширением .txt или с расширением, выбранным вами, но вы должны быть уверены в том, что выбираете для передачи именно текстовый файл, а не двоичный файл данных. После щелчка мышкой по кнопке **ОК** это окно закроется и начнется передача файла. В процессе передачи файла на экране присутствует соответствующее информационное окно (с именем передаваемого файла) и вы можете прервать передачу текста с помощью кнопки **Abort** (Сброс) в этом окне. В противном случае по окончании передачи всего файла окно закрывается автоматически.

Оценка качества сигналов

Как и в SSB-режиме, для правильной передачи сигналов PSK31-передатчик должен работать в режиме с высокой линейностью усиления. При работе же в режиме RTTY высокая линейность передатчика не требуется. Отчасти поэтому некоторые начинающие пользователи режима PSK31 зачастую пренебрегают необходимостью обеспечения линейности работы своего передатчика. В программу PSK31SBW (начиная с версии 1.08) введен режим автоматической оценки линейности (качества) принимаемого сигнала, что позволяет без особых хлопот предоставлять корреспонденту оценку качества сигнала его передатчика.

Значение качества принимаемого сигнала появляется в круге индикатора фазы в те моменты, когда не передается текст. Эта оценка дается в единицах дБ и характеризует уровень побочных спектральных составляющих сигнала (отстоящих не менее чем на 46 Гц от центральной частоты сигнала) относительно основной пары спектральных составляющих сигнала (имеющих частоты ± 15 Гц от центральной частоты сигнала). Значение качества сигнала выдается программой только при приеме сигнала синхронизации, т. е. тогда, когда излучаемый передатчиком корреспондента сигнал представляет собой двухтональный сигнал, который обычно используется для оценки линейности (качества) SSB-передатчиков. Фактически число, представляющее качество сигнала в дБ, является оценкой уровня интермодуляционных составляющих третьего порядка в спектре частот, излучаемых SSB-передатчиками.

Для оценки качества сигнала PSK31 дождитесь момента начала передачи сигнала синхронизации от передатчика корреспондента или попросите его самого некоторое время ничего не передавать (и не нажимать на клавиши), чтобы считать с дисплея и затем передать ему это число в дБ. Конечно, если ваш корреспондент тоже пользуется программой PSK31SBW (версии 1.08 или более поздней), то вы можете запросить у него аналогичную оценку качества вашего сигнала, сделав для этого некоторую паузу в своей передаче.

Возможны ситуации, в которых показания индикатора могут оказаться ошибочными:

- ❑ при наличии очень близко по частоте сигнала другой станции, который способен исказить оценку качества требуемого сигнала, хотя при этом и не мешает его приему;
- ❑ очень слабый сигнал может оцениваться как сигнал низкого качества, если его побочные составляющие скрыты шумовым фоном в обрабатываемой полосе частот;
- ❑ не забывайте также о том, что низкая оценка качества принимаемого сигнала может быть обусловлена и искажениями в вашем собственном приемнике. К примеру, быстродействующая система АПЧ способна иногда вызывать искажения принимаемого сигнала, иногда же сказывается недостаточное качество основного НЧ-выхода приемника (на динамик), в то время, как выход на наушники (или линейный выход НЧ) может оказаться вполне приемлемого качества. Низкое, по показаниям дисплея, качество всех принимаемых сигналов обычно свидетельствует о наличии проблем в самом приемнике, в котором, например, мощные сигналы в диапазоне частот создают нечто подобное побочным продуктам преобразования.

Качество сигнала должно быть не менее 25 дБ. Если вы получили рапорт с более низкой оценкой, есть основание насторожиться. Некоторые SSB-передатчики способны обеспечить в режиме полной мощности не более 20 дБ, в связи с чем для обеспечения лучшего качества сигнала следует использовать их в режиме пониженной выходной мощности. Значения же менее 10 дБ соответствуют подаче на вход передатчика прямоугольных импульсов вместо синусоидального напряжения.

Обобщение о PSK31

Несмотря на применение в цифровой любительской связи новой технологии с использованием DSP (компьютерных звуковых карт), применяемых для передачи данных на КВ-диапазонах, по-прежнему сохраняет популярность традиционный режим RTTY, обладающий очевидными достоинствами в сравнении с другими, более современными режимами. По-видимому, это обусловлено тем, что режим RTTY проще в использовании и в большей степени является "ручным" и "общительным" способом связи. Вы можете слышать корреспондентов, ведущих связь между собой, и присоединяться к их беседе, что, например, невозможно сделать при проведении связи в режиме ARQ (AMTOR, PACTOR). Кроме того, режим RTTY обеспечивает наиболее простую и дешевую цифровую связь.

Новый вид цифровой связи, появившийся вдруг у радиолюбителей в 1998 году, способен восполнить пробел между RTTY с одной стороны и другими высокотехнологичными режимами с другой стороны. Это режим PSK31, ос-

нованный на идеях SP9VRC и ориентированный на использование дешевых комплектов DSP и общедоступного программного обеспечения. В нем современная DSP-технология используется для обеспечения возможности проведения оживленных QSO, связанных с передачей/приемом данных по типу "клавиатура-экран". Полоса частот сигнала в режиме PSK31 намного уже, чем в любом другом режиме передачи данных, благодаря чему он обеспечивает возможность связи с более низкими уровнями сигналов в перегруженных на сегодняшний день любительских диапазонах.

Основная идея режима PSK31 состоит в том, что манипуляция сигналом выполняется сдвигом фазы несущей вместо сдвига ее частоты. Полоса частот сигнала при этом определяется только скоростью манипуляции, а не значением сдвига частоты в сумме со скоростью манипуляции. Выбранная скорость манипуляции в 31,25 Бод обуславливает полосу частот всего в 31 Гц вместо 300—500 Гц для других режимов. За счет использования алфавита кода, подобного алфавиту кода Морзе с короткими кодами для наиболее часто встречающихся символов (в текстах на английском языке), режим PSK31 обеспечивает скорость передачи текста порядка 50-ти слов в минуту. При использовании в передатчике и приемнике самых узкополосных фильтров, какие только возможны, эффективность режима PSK31 даже без автоматической коррекции ошибок, безусловно, превосходит эффективность любого другого режима и обеспечивает дополнительное преимущество для оживленных QSO, состоящее в том, что даже при "провалах" в уровне сигнала эффективность режима снижается очень незначительно. На радиотрассах, где чаще случаются ошибки пакетного типа, чем равномерно распределенные одиночные сбои, программа PSK31SBW может быть переключена в режим использования специального кода при передаче и соответствующего декодера при приеме. Это режим QPSK с четырьмя фазовыми сдвигами вместо двух, но полоса частот, занимаемая сигналом, и скорость передачи символов остаются прежними (как с сигналами BPSK). Цена обеспечиваемого при этом повышения помехоустойчивости — более жесткие требования к стабильности несущей частоты и задержка в 640 мс при декодировании символов.

Немного о MT63

Автором этого вида связи является Pawel Jalocho (SP9VRC). Им также разработан один из вариантов программы MT63 для системы Linux. Программу для Windows 95/98 разработал Nino (IZ8BLY), его страница в Интернете находится по адресу <http://ninopo.freeweb.org/>.

MT63 предназначен для проведения диалоговой радиосвязи между любительскими радиостанциями, используя процессы (FEC) корректировки ошибок.

Модем MT63 может быть построен на высокоскоростном DSP-процессоре, или на внешнем DSP-аппарате, подобно Motorola EVM, или с помощью

программного обеспечения РС, использующем звуковую карту. Модем выдает на передачу сигнал, состоящий из 64 отдельных тонов с шириной полосы 15,625 Гц каждый, что в сумме составляет полосу частот шириной 1 кГц. Сигнал занимает частоты от 500 до 1500 Гц. При этом скорость передачи символов для телеграфа составит примерно 10 символов в секунду.

Спектр сигнала МТ63 при ширине полосы 1 кГц представлен на рис. 2.5.

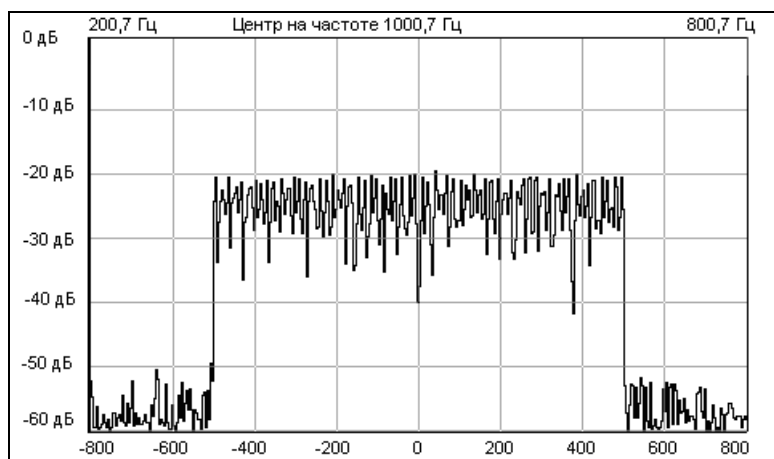


Рис. 2.5. Диаграмма полосы сигнала МТ63

Кроме полосы шириной в 1 кГц, МТ63-сигнал может использоваться при ширине полосы в 500 Гц и 2 кГц. По умолчанию принимается сигнал с полосой 1 кГц. В каждом из трех указанных сигналов низшая частота модулирующего тона всегда остается равной 500 Гц. При этом для сигнала с меньшей шириной полосы (500 Гц) длительность сигнальной посылки увеличивается в два раза, а для сигнала с большей шириной полосы (2 кГц) длительность сигнальной посылки уменьшается в два раза. Соотношения ширины полосы и скоростей передачи сигналов МТ63 представлено в табл. 2.24.

Таблица 2.24. Соотношения в сигналах МТ63

Полоса	Диапазон	Скорость	Символ/сек	Время/символ
500 Гц	500–1000 Гц	5 Бод	5	6,4 или 12,8
1 000 Гц	500–1500 Гц	10 Бод	10	3,2 или 6,4
2 000 Гц	500–2000 Гц	20 Бод	20	1,6 или 3,2

Данные пользователя от клавиатуры или из файла (код данных — ASCII с 7 битами) далее разделяются на 64 частицы, используя функцию Walsh для обеспечения FEC-корректировки ошибок. Функция Walsh гарантирует, что при повреждении 16 из этих 64 элементарных посылок, расшифровка будет давать однозначный результат.

MT63-сигнал распространяется и во времени, и в пространстве. Чтобы гарантировать минимальное нарушение сигнала от помех, каждый кодируемый символ разбит более чем на 32 последовательные элементарные посылки (3,2 секунды). Чтобы гарантировать отсутствие ошибок при ухудшении условий прохождения сигнала, символ разбивается на 64 элементарные посылки, при этом время увеличивается до 6,4 секунды.

Сигнал радиостанции, работающей в режиме MT63, напоминает какое-то рычание. Совершенно не верится, что такой сигнал сформирован из чистых звуковых тонов.

Дополнительную информацию по этому виду цифровой радиосвязи можно получить в Интернете по адресу <http://www.qsl.net/zl1bpu/MT63/MT63.html>.

Любительское телевидение на коротких волнах

Следует упомянуть еще один вид цифровой радиосвязи, который пользуется популярностью у некоторой группы коротковолновиков, — SSTV (Slow Scan TeleVision) — телевидение с медленной разверткой. SSTV представляет собой передачу и прием статического изображения (картинки) посредством радиостанции и компьютера. Дело в том, что отечественное телевидение использует радиоканал шириной 6,5 МГц. Только такая ширина канала может создать на экране иллюзию движения и быструю смену цветов. На любительских коротковолновых диапазонах допускается ширина канала 3 кГц, так что при такой величине можно передавать и получать только статические изображения. Такое телевидение отдельные радиолюбители используют для передачи интересных, красочно оформленных сообщений, фотографий, QSL-карточек.

Связь через метеорные потоки

Среди радиолюбителей многие увлекаются связью на УКВ (ультракоротковолновый диапазон радиоволн). Радиосвязь на УКВ имеет свои особенности и разновидности. Одной из таких разновидностей как раз и является радиосвязь за счет отражения от метеорных потоков. Из астрономии известно, что наша Земля во время своего годового вращения поворачивает нас к такому участку небесного пространства, где располагаются различные по интенсивности метеорные потоки. Все эти потоки имеют четкое местопо-

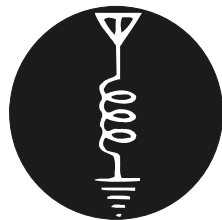
ложение в пространстве, определенные названия. Имеются специальные таблицы, по которым подготовленный человек может рассчитать время, когда Земля повернет нас к тому или иному потоку. Метеорный поток — это масса метеоров, которые летят в сторону Земли. Попадая в земную атмосферу, метеоры сгорают, оставляя за собой ионизированный след. Ионизированный след живет несколько мгновений, и этого бывает достаточно, чтобы радиосигнал отразился от следа и вернулся на Землю.

Радиосвязь осуществляется следующим образом. Один из радиолюбителей направляет антенну в сторону метеорного потока и с определенными интервалами посылает в сторону потока радиосигналы с очень большой скоростью. Если другой радиолюбитель, удаленный от первого на большое расстояние, направит свою антенну на тот же метеорный поток, то, при удачном стечении обстоятельств, может принять переданный первым радиолюбителем сигнал, который будет слышен как некий щелчок. Чтобы расшифровать полученный сигнал, он должен быть во время приема записан на специальное приемное устройство, например, магнитофон, работающий на запись при очень большой скорости. Чтобы полученный сигнал расшифровать, магнитофонную ленту с записью сигнала прослушивают на очень малой скорости. Пример с магнитофоном я привел только для лучшего понимания процесса, хотя несколько лет тому назад для подобных связей действительно применяли магнитофоны. Сейчас для этих целей применяются компьютеры, оснащенные специальными программами.

Заключение

В этой главе я рассказал только о самых основных, наиболее распространенных видах цифровой радиосвязи. За последние годы появилось много новых видов цифровой связи, но среди них пока нет такого вида, который бы стал достойной заменой ныне существующим и описанным в этой главе.

Глава 3



Аппараты для цифровой связи

Предисловие

С самого зарождения любительских цифровых видов связи и до наших дней неизменным атрибутом этих видов связи является аппарат, связывающий и согласующий радиостанцию и компьютер в единый рабочий орган. На выходе приемника мы всегда имеем сигналы с переменной амплитудой и частотой, которые в таком виде непонятны для компьютера, а на выходе портов компьютера имеем сигналы в виде импульсов прямоугольной формы, в виде токовых и бестоковых посылок различной длительности, которые не могут быть понятны непредназначенному для приема таких сигналов радиопередатчику. В этой книге речь не идет о специальных радиостанциях, которые имеют в своем составе встроенный процессор и не нуждаются ни в каких дополнительных аппаратах. Речь идет об обычных радиолюбительских приемопередатчиках — SSB-трансиверах, которыми оснащены в настоящее время большинство любительских радиостанций.

В этой главе я описываю конструкции различных аппаратов, необходимых для проведения различных видов цифровых радиосвязей, описанных в *главе 2*, с применением компьютера IBM PC и любительского приемопередатчика с однополосной модуляцией сигналов (SSB-трансивера).

Большинство из описанных в этой главе конструкций различных аппаратов разработано мною в процессе многолетней радиолюбительской деятельности. Все конструкции прошли испытание на моей домашней радиостанции и испытательном стенде, а также на радиостанциях других радиолюбителей-коротковолновиков.

Кроме того, в этой главе будут приведены схемы экспериментальных аппаратов, которые еще не являются законченными конструкциями и предлагаются радиолюбителям как материал для дальнейшего их совершенствования.

При разработке описанных в книге своих конструкций мною не ставилась задача конкурировать с известными фирмами — изготовителями модемов и другой электронной техники. Разработанные мною конструкции предназначены для любознательного читателя-радиолюбителя, который хотел бы своими руками сделать приличный аппарат из широкодоступных радиодеталей, способный передавать и принимать сигналы не хуже многих фирмен-

ных. Поэтому критикам-любителям не стоит сравнивать разработанные мною конструкции с изделиями известных фирм!

Модем для CW

Слово "модем" является сложным словом и состоит из двух составляющих — из двух начальных букв слова "модулятор" и трех начальных букв слова "де-модулятор". Сразу хочу предупредить, что используемый для работы в сети Интернет модем и модем для любительской радиосвязи имеют только одинаковое название. Внутреннее устройство и многие принципы работы этих двух модемов совершенно разные. Чтобы понять значение слов "модулятор" и "демодулятор", обратимся к примеру с двумя радиостанциями, когда на одной из них находится радист, который телеграфным ключом передает в эфир сигналы кода Морзе, а радист на другой радиостанции принимает эти сигналы, переводит сигналы кода Морзе в буквы и записывает эти буквы на бумагу. Как только передающая радиостанция включается на передачу, она сразу же начинает излучать в эфир электромагнитную энергию. Радисты говорят, что передатчик радиостанции начинает излучать "несущую частоту". В этот момент радист на принимающей радиостанции обнаруживает факт включения передающей станции по наличию несущей частоты, но никакого полезного сигнала пока нет. Когда же радист передающей радиостанции нажимает на рычаг телеграфного ключа, то на эту электромагнитную энергию (несущую частоту) накладывается сигнал звуковой частоты. Этот сигнал звуковой частоты вырабатывается специальным генератором. Радист на принимающей радиостанции тут же начинает слышать в наушниках приятный звуковой сигнал. Как только радист передающей станции прекращает нажатие на рычаг телеграфного ключа, радист принимающей станции в своих наушниках уже никакого полезного сигнала не слышит. Процесс наложения звукового сигнала на излучаемую передатчиком электромагнитную энергию называли "модуляция", а процесс периодического нажатия и отпускания рычага телеграфного ключа называли словом "манипуляция". Позже на больших радиостанциях процесс модуляции вместо радиста стали выполнять специальные аппараты, которые получили название "модуляторы". Аналогичный процесс замены радиста на соответствующий аппарат произошел и на приемной радиостанции. Там появился аппарат, который в автоматическом режиме принимал модулированные звуковой частотой сигналы кода Морзе и переводил полученные сигналы кода Морзе в буквы. Такой аппарат стали называть "демодулятор". Так что наш модем является аппаратом, который с одной стороны может "накладывать" на излучаемую передатчиком электромагнитную энергию полезные сигналы звуковой частоты, т. е. быть модулятором, а с другой стороны может полученные от приемника сигналы звуковой частоты превращать в понятные для компьютера импульсы положительного или отрицательного напряжения, т. е. быть демодулятором.

В нашей стране промышленность не изготавливает модемы для любительской радиосвязи. При большом желании и наличии соответствующей суммы денег можно найти и приобрести такой модем, изготовленный за рубежом. Для всех прочих любителей остается только путь самостоятельного изготовления модема. Полное описание конструкций разных модемов и принципиальные электрические схемы отдельных узлов этих модемов можно скачать в Интернете со странички, расположенной по адресу <http://users.kaluga.ru/ra3xb/>.

Далее в этом разделе я привожу принципиальные электрические схемы отдельных узлов модема, описание которого также располагается на приложенной к книге дискете под названием MODEM3. Модем рассчитан на его использование совместно с IBM PC или полностью совместимым компьютером. Практически всю конструкцию, после незначительной переделки, можно использовать с любым компьютером. По своему назначению модем аналогичен сложному VayCom и внешнему HamCom модемам и использует те же выводы COM-порта.

Конструкции монтажных плат не описываю, чтобы не навязывать любителям свои технологические приемы. Каждый может делать любой блок по своей собственной технологии, придерживаясь только основных общепринятых правил расположения и крепления радиодеталей.

Конструктивно модем разделен на отдельные блоки. Каждый блок имеет свою принципиальную электрическую схему и ее краткое описание. Предполагается, что каждый блок будет изготовлен на отдельной плате, и это позволит в дальнейшем легко проводить модернизацию модема путем замены одного из блоков новым, усовершенствованным блоком.

Схема модема

На рис. 3.1 представлена блок-схема модема, все детали различных узлов модема рассчитаны на его работу со средней частотой примерно 1000 Гц. Модем работает на звуковых (аудио) частотах и совмещает в себе две основных составных части — передающую часть (модулятор) и приемную часть (демодулятор).

Модулятор включает в себя устройство для включения и выключения передатчика и собственно модулятор — устройство для подачи на вход передатчика посылок от тонального генератора (U1). Демодулятор включает в себя полосовой фильтр на операционных усилителях (U2), специальный частотный детектор (U3) и выходной узел (U4).

Подключение модема к компьютеру должно выполняться через стандартный COM-порт с интерфейсом RS-232-C. Официальное ограничение по длине для соединения экранированным кабелем по стандарту RS-232-C составляет 15,2 м. На практике это расстояние должно быть как можно короче. Уровни напряжений на линиях разъема для логического нуля следует считать

−12...−3 В, для логической единицы +3...+12 В. Промежуток от −3 до +3 В соответствует неопределенному значению. Каждый COM-порт имеет свой собственный разъем, который может иметь либо 25 контактов (DB25), либо 9 контактов (DB9).

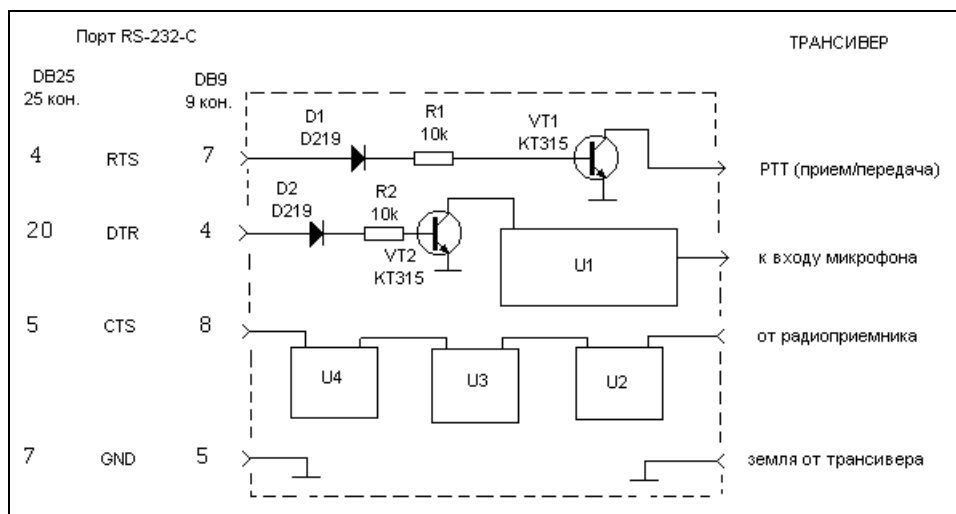


Рис. 3.1. Блок-схема модема

На блок-схеме слева указаны номера контактов разъема COM-порта для вариантов применения DB25 и DB9, справа указаны гнезда приемопередатчика, к которым подводится или от которых берется сигнал. Компьютер через контакты 4 (7) разъема COM-порта (здесь и далее первая цифра относится к разъему с 25 контактами, а цифра в скобках — к разъему с 9 контактами) подает сигнал для управления переключением передатчика прием/передача. Назначение этого контакта в системе RS-232-C — запрос для передачи, наименование — RTS (Request to send). Если компьютер установит на этом выводе положительное напряжение, то передатчик радиостанции включится для работы на передачу и пока положительная величина напряжения не будет отключена, передатчик будет работать на передачу. Чтобы выключить передатчик, компьютер должен подать на этот вывод отрицательное напряжение. При команде на передачу сигнал через диод VD1 и резистор R1 поступает на транзисторный переключатель, выполненный на транзисторе VT1. К цепи коллектора этого транзистора подключается устройство, предназначенное для переключения "прием/передача", установленное на передатчике. При подаче на базу транзистора VT1 положительного напряжения устройство переключения срабатывает и включает радиостанцию на передачу.

Блок U1 представляет собой тональный генератор (рис. 3.2). Сигналы для манипуляции тонального генератора берутся с контакта 20 (4) разъема.

Чтобы наложить сигналы кода Морзе на несущую частоту, компьютер использует контакт 20 (4) разъема COM-порта. Назначение этого контакта — готовность выходных данных — DTR (Data Terminal Ready). Если компьютер устанавливает на этом контакте (выводе) положительное напряжение, то сигнал от контакта 20 (4) через диод VD2 и резистор R7 поступает на базу транзисторного ключа на VT2, который в тот же момент включает в работу на определенной звуковой частоте устройство U1, являющееся тональным генератором. Генерируемая этим тональным генератором звуковая частота накладывается на несущую частоту передатчика и в эфир идет полезный сигнал. Получается так, что транзистор VT2 выполняет роль телеграфного ключа и выполняет процесс модуляции путем манипуляции тонального генератора. Далее сигналы манипулируемого тонального генератора подаются на микрофонный вход передатчика.

С контакта 5 (8) разъема COM-порта компьютер принимает сигнал от демодулятора. Название контакта — сброс для передачи — CTS (Clear to Send). Чтобы стать понятным компьютеру, сигнал от приемника должен пройти несколько обработок. Сначала сигнал от приемника поступает на устройство U2 (рис. 3.3) — полосовой фильтр, основное назначение которого состоит в создании достаточно узкой полосы пропускания для поступившего от приемника полезного сигнала и отфильтровывании помех от сигналов соседних работающих радиостанций. Затем сигнал поступает на устройство U3 (рис. 3.4) — специальный частотный детектор. Если на детектор поступает полезный звуковой сигнал, то на выходе детектора устанавливается положительное напряжение, если полезного сигнала в данный момент нет, то на выходе детектора напряжение равно нулю. Далее сигналы от детектора перед поступлением на компьютер проходят еще одно устройство — U4 (рис. 3.5), которое отфильтровывает выходящий сигнал от продуктов преобразования и выдает на разъем COM-порта электрический сигнал в точном соответствии с требованиями стандарта RS-232-C.

Контакт 7 (5) разъема COM-порта является общим для всех остальных контактов заземляющим. Он носит название — сигнальное заземление GND или SG (Signal Ground). Разъем DB25 имеет еще один заземляющий контакт 1 (в DB9 такого контакта нет). Этот контакт называется — защитное заземление — FG (Frame Ground) и служит только для соединения экранирующего корпуса модема с экранирующим корпусом компьютера. Контакты 7 (5) и 1 соединять между собой ни в коем случае нельзя!

Кроме перечисленных выше вводов и выводов, на модем следует подать от источника питания напряжение +5 В, +12 и -12 В, не забудьте общий провод (заземление) от источника питания для напряжения +5 В и двуполярного выпрямителя.

Отдельные узлы модема

Тональный генератор

Тональный генератор U1 представлен на рис. 3.2. Генератор выполнен на транзисторах VT1 и VT2 типа КТ315Б по схеме с обратной связью через двойной Т-мост, обладает высокой стабильностью и достаточно хорошим качеством сигнала при питании от стабилизированного источника.

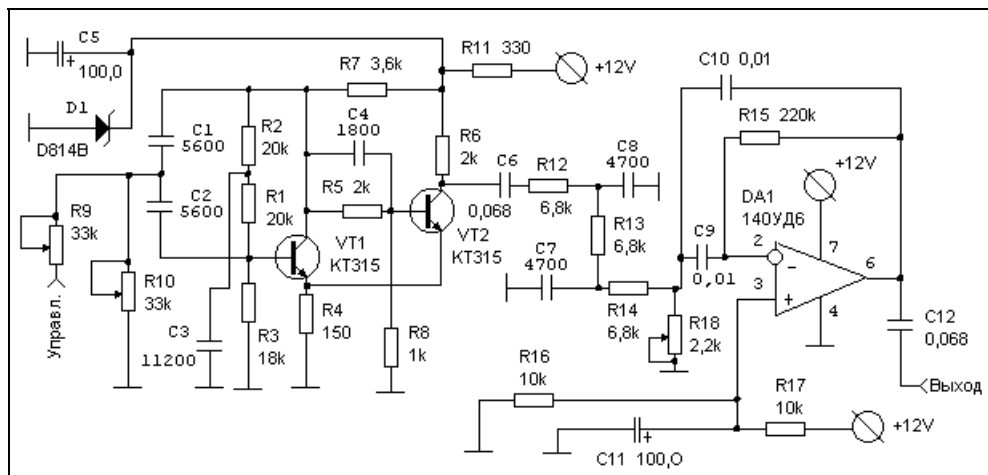


Рис. 3.2. Принципиальная электрическая схема тонального генератора

Конденсаторы C1, C2 и C3 должны иметь допуск не хуже 10 процентов. На операционном усилителе DA1 типа К140УД6 выполнен полосовой фильтр, назначением которого является улучшение синусоидальности выходного сигнала. Для настройки частотомер подключается к точке выхода, регулировкой R10 устанавливается величина нижней частоты, а регулировкой R9 (при замкнутой на землю точке входа "управление") устанавливается величина верхней частоты. Изменением величины резистора R7 можно корректировать качество синусоиды генерируемого сигнала (только при осциллографическом контроле). Резистором R18 добиваются равной амплитуды для сигналов высокой и низкой частоты.

Полосовой фильтр

Полосовой фильтр U2 может иметь множество вариантов исполнения. Заранее трудно предсказать наилучший вариант. Предлагаю самый простой одноканальный вариант активного полосового аудиофильтра. Следует начать с изготовления первого, самого простого, варианта на отдельной плате и если он не будет удовлетворять требованиям приема, то в дальнейшем следует его

дит сигналы в остроконечные импульсы малой длительности, поступающие на DD2 и DD3.

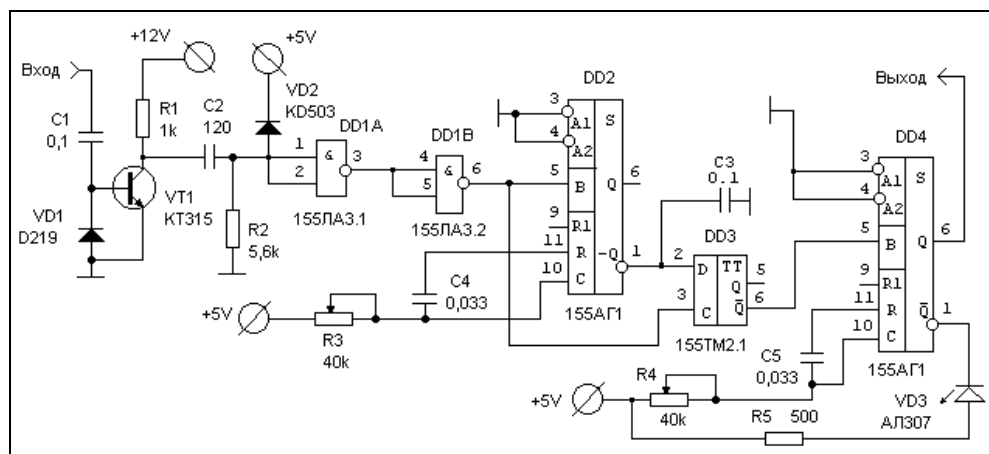


Рис. 3.4. Специальный частотный детектор

Резистором R3 детектор настраивается на среднюю частоту фильтра U2. Для этого, перемещая движок резистора R4, установить такое положение движка, при котором произойдет резкое срабатывание светодиода VD2.

Для настройки детектора выполните следующие пункты:

1. Подключите вольтметр постоянного напряжения на выход детектора и подайте на вход модема сигнал от звукового генератора.
2. Установите 3Г на частоту срабатывания детектора, которая примерно должна соответствовать центральной частоте настройки входного полосового фильтра, изменяя величину сопротивления настройки DD2 (R3) от минимума, остановите движок резистора при резком уменьшении показаний вольтметра.
3. Проверьте срабатывание детектора при уменьшении и увеличении частоты 3Г выше или ниже частоты срабатывания детектора, при этом регулирующий потенциометр на DD4 должен иметь минимальную величину сопротивления.
4. Изменяя величину сопротивления резистора настройки DD4 (R4), наблюдайте постоянное увеличение показаний вольтметра и, в какой-то момент, резкое изменение показаний. В этот момент следует прекратить вращение движка резистора и на небольшую величину угла повернуть движок резистора R4 в обратную сторону.

5. изменяя частоту ЗГ в пределах полосы пропускания фильтра, наблюдайте скачкообразное изменение показаний вольтметра от нуля до максимума (и наоборот) ТОЛЬКО при переходе через установленную ранее частоту настройки детектора. Если будут наблюдаться и другие, пускай даже и незначительные, скачки, то следует уменьшить величину сопротивления второго резистора R4 (настроить DD4).

Выходное согласующее устройство

На рис. 3.5 представлена принципиальная электрическая схема согласующего выходного устройства U4.

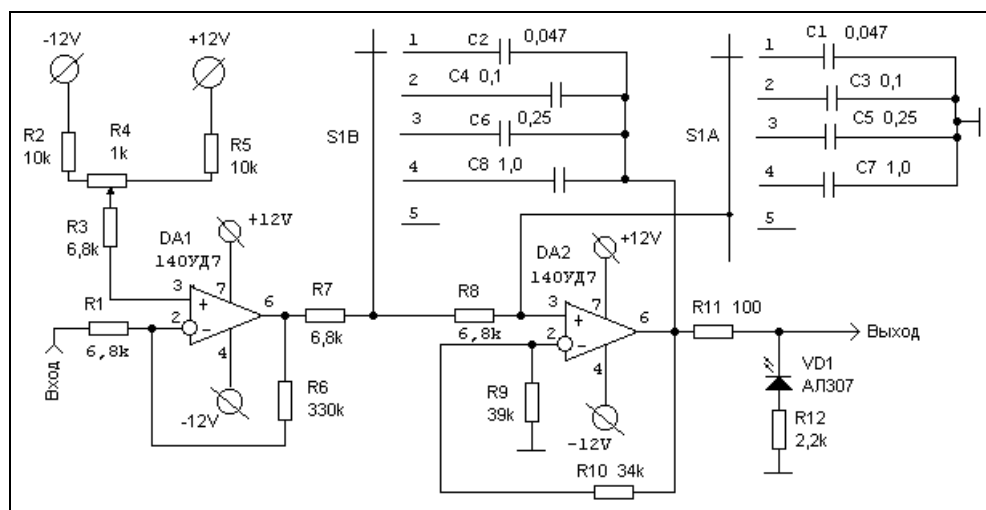


Рис. 3.5. Выходное согласующее устройство

На операционных усилителях DA1 и DA2 собран фильтр, который очищает полезный сигнал от продуктов преобразования и, одновременно, выдает для СОМ-порта полезные сигналы необходимой амплитуды и полярности. Практика показала достаточность такого согласователя практически для любых видов детекторов. Поскольку радиолюбители зачастую вынуждены использовать некондиционные детали, в схему добавлен потенциометр, которым необходимо установить величины напряжений выходного сигнала при наличии входного сигнала (от +10 до +12 В) и при отсутствии сигнала (от -10 до -12 В). Переключатель S1 может быть любой конструкции, при этом конденсаторы величиной 0,047 (положение 1) должны использоваться для приема пакетных сигналов 300 Бод, конденсаторы 1,0 мкФ используются при телеграфе, а остальные положения переключателя должны использоваться при других видах и скоростях. Подбирайте другие, более удобные для вас величины емкостей этих конденсаторов.

Настройка модема

Изготовленный модем следует тщательно настроить. Особенно внимательно нужно выполнить настройку детектора, от нормальной работы которого зависит работа всего устройства. Настройка выполняется за несколько этапов.

Начать настройку модема следует с выбора величин рабочих частот. Поскольку модем может работать с узкополосным сигналом, следует за основу при расчетах взять характеристики узкополосного (телеграфного) фильтра. Работать предполагается всеми видами цифровой связи. Предположим, что радиоприемник на вашей станции имеет узкополосный фильтр с шириной полосы 300 Гц и пропускает частоты от 1000 до 1300 Гц. В этом случае величина средней частоты будет равна 1150 Гц. Учитывая величину сдвига частот, равную 200 Гц, определяем, что нижняя частота должна быть 1050 Гц, а верхняя — 1250 Гц.

Далее проводим настройку тонального генератора (см. рис. 3.2). Для настройки частотомер подключается к точке выхода, регулировкой R10 устанавливается величина нижней частоты (например, 1050 Гц), а регулировкой R9 (при замкнутой на землю точке входа "управление") устанавливается величина верхней частоты (например, 1250 Гц). Изменением величины резистора R7 можно корректировать качество синусоиды генерируемого сигнала (только при осциллографическом контроле). Резистором R18 добиваются равной амплитуды для сигналов высокой и низкой частоты. Величина амплитуды выходного сигнала должна измеряться вольтметром, подключенным к выходу тонального генератора.

Затем следует от звукового генератора подать на вход фильтра (см. рис. 3.3) сигнал с частотой 1150 Гц. Каскады на DA1 и DA2 резисторами R5 и R6 настраиваются на эту среднюю частоту поочередно.

Далее следует подключить вольтметр постоянного напряжения на выход специального частотного детектора (см. рис. 3.4) и подать на вход модема сигнал от звукового генератора 1150 Гц. При этом будем считать, что именно эта частота должна быть частотой срабатывания детектора. После того как вы наберетесь побольше опыта в настройке детектора, можно будет по своему желанию изменять величину частоты срабатывания детектора.

Установите движок резистора R3 на минимальное сопротивление. Медленно изменяя величину сопротивления резистора R3 от минимума в сторону увеличения, следует сразу же остановить движок резистора при резком уменьшении показаний вольтметра. Можно считать, что нами установлена нужная частота срабатывания детектора, равная величине 1150 Гц.

Следующим этапом должна быть проверка правильности выполненных действий по установке частоты срабатывания детектора. Это легко проверить, если изменять частоту звукового генератора, подключенного на вход модема. При изменении частоты звукового генератора от 1050 до 1250 Гц пока-

зания вольтметра должны резко изменяться при переходе через частоту 1150 Гц. При этом регулирующий потенциометр на DD4 должен иметь минимальную величину сопротивления.

Резистором R4 можно изменять "наполняемость" сигнала. Изменяя величину сопротивления резистора настройки DD4 (R4) наблюдать постоянное увеличение показаний вольтметра и, в какой-то момент, резкое изменение показаний. В этот момент следует прекратить вращение движка резистора и на небольшую величину угла повернуть движок резистора R4 в обратную сторону. На рис. 3.6 показаны осциллограммы сигналов в отдельных точках схемы детектора.

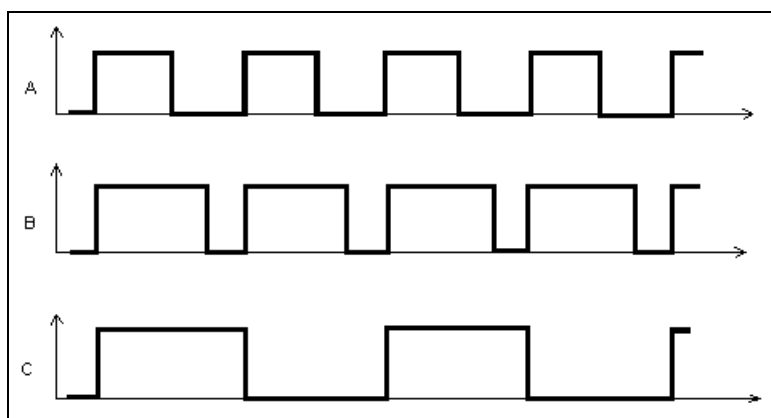


Рис. 3.6. Осциллограммы сигналов детектора

В положении А изображена осциллограмма, полученная на выводе 6 микросхемы DD3, в случае, если частота сигнала на входе выше частоты срабатывания детектора. На графике видно, что положительные импульсы составляют половину периода, т. е. имеем вид колебаний, называемый "меандр". Если частота сигнала будет меньше частоты срабатывания детектора, то в этой точке напряжение будет равно нулю. Далее сигнал поступает на микросхему DD4, назначение которой только одно — увеличить продолжительность положительного импульса, как бы растянуть его. Казалось бы, что можно обойтись и без этого каскада, но практика показала, что каскад на DD4 улучшает сигнал на выходе модема. Увеличение длительности положительного импульса выполняется резистором R4. Нормальным можно считать положение, когда длительность положительного импульса составляет не более $3/4$ от длительности периода. Такой вариант осциллограммы на выходе микросхемы DD4 показан в положении В. В случае увеличения длительности сверх указанной величины возможно скачкообразное изменение формы сигнала и увеличение периода колебаний в два раза. Момент такого резкого

изменения периода отмечается вольтметром как резкое уменьшение величины напряжения на выходе. Состояние сигнала на выходе микросхемы DD4 будет соответствовать положению С на рис. 3.6. Такое явление считается нежелательным, поэтому будьте внимательны при регулировке величины резистора R4.

Заключительную проверку можно сделать, изменяя частоту 3Г в пределах полосы пропускания фильтра, при этом наблюдать скачкообразное изменение показаний вольтметра от нуля до максимума (и наоборот) ТОЛЬКО при переходе через установленную ранее частоту настройки детектора (1150 Гц). Если будут наблюдаться и другие, пускай даже и незначительные, скачки то следует уменьшить величину сопротивления резистора R4 (настроить DD4).

Предложенный вариант конструкции модема хорошо согласуется с COM-портом компьютера и показал очень хорошие результаты при работе телеграфом и удовлетворительные результаты при работе с другими видами цифровой связи — RTTY, AMTOR, PACTOR и Packet Radio. Основное достоинство этой конструкции — возможность принимать сигналы при очень узкой полосе входного фильтра, что исключительно важно в условиях больших помех. Ни одна другая из известных мне конструкций этого делать не может. Возможность работы с очень узкой полосой получается только благодаря применению специального частотного детектора.

Если кто-то решил взяться за изготовление этой конструкции, то не стоит откладывать начатое дело. Блочная конструкция позволяет заменять любой неудачный блок более надежным и более удобным.

Модем универсальный

Описываемая в этом разделе конструкция модема MODEM22 рассчитана на ее использование совместно с IBM PC или полностью совместимым компьютером и является последующей доработкой разработанного в 1990 году модема MODEM2. Многочисленные тесты модема MODEM22 и его аналогов показали, что этот модем является на сегодняшний день ЛУЧШИМ среди других любительских разработок, превосходит известные мне любительские модемы, выполненные с применением детекторов на микросхемах 564ГГ1 и 155АГ1, по качеству приема сигналов и по простоте настройки во много раз. Модем опробовался мною в работе со следующими программами Packet Radio при скорости 300 Бод:

- ☐ BayCom 1.60;
- ☐ HC (аналог SP) + TFPCX;
- ☐ F6FBB 1.15c + TFPCX;
- ☐ mubay102 + TFPCX;
- ☐ wintnc11;

□ WinPack 6.42(6.51) + AGWpe;

□ AGWterm + AGWpe

и некоторыми другими, менее известными.

А также в других видах цифровой связи — CW, RTTY, AMTOR и PACTOR. При этом качество сигнала было превосходным.

Схема модема

На рис. 3.7 представлена блок-схема модема, все детали различных узлов модема рассчитаны на его работу со средней частотой около 1000 Гц. Модем работает на звуковых частотах и совмещает в себе две основных составных части — передающую часть (модулятор) и приемную часть (демодулятор). Модулятор, в свою очередь, включает в себя устройство для включения и выключения передатчика и собственно модулятор — устройство для подачи на микрофонный вход SSB-передатчика посылок от тонального генератора (U1). Демодулятор включает в себя полосовой фильтр на операционных усилителях (U2), специальный частотный детектор (U3) и выходной узел (U4). Предполагается изготовление каждого из узлов модема на отдельной плате, что позволит в дальнейшем безболезненно заменять неудачно выполненные узлы.

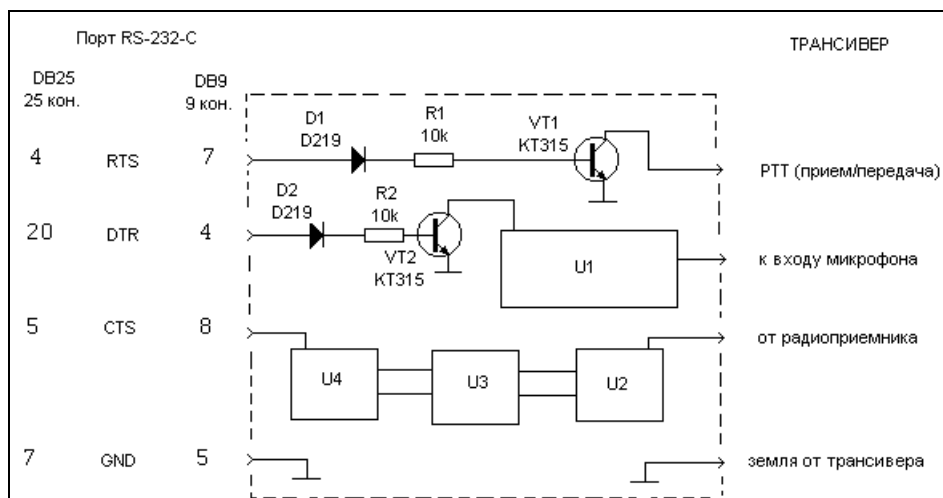


Рис. 3.7. Блок-схема модема

Подключение модема к компьютеру должно выполняться через стандартный СОМ-порт с интерфейсом RS-232-С. Официальное ограничение по длине для соединения экранированным кабелем по стандарту RS-232-С составляет

15,2 м. На практике это расстояние должно быть как можно короче. Уровни напряжений на линиях разъема для логического нуля следует считать от -12 до -3 В, для логической единицы от $+3$ до $+12$ В. Промежуток от -3 до $+3$ В соответствует неопределенному значению. Каждый СОМ-порт имеет свой собственный разъем, который может иметь либо 25 контактов (DB25), либо 9 контактов (DB9).

На блок-схеме слева указаны номера контактов разъема СОМ-порта для вариантов применения DB25 и DB9, справа указаны гнезда приемопередатчика, к которым подводится или от которых берется сигнал.

С контакта 4 (7) разъема СОМ-порта (здесь и далее первая цифра относится к разъему с 25 контактами, а цифра в скобках — к разъему с 9 контактами) берется сигнал для управления переключением передатчика прием/передача. Назначение этого контакта в системе RS-232-C — запрос для передачи, наименование — RTS (Request to send). Далее через диод VD1 и резистор R1 сигнал поступает на транзисторный переключатель, выполненный на транзисторе VT1. К цепи коллектора этого транзистора подключается катушка от реле "прием/передача", установленного на трансивере. При подаче на базу транзистора VT1 положительного напряжения реле срабатывает и включает трансивер на передачу.

Блок U1 представляет собой тональный генератор. Сигналы для манипуляции тонального генератора берутся с контакта 20 (4) разъема. Назначение этого контакта — готовность выходных данных — DTR (Data Terminal Ready). С контакта 20 (4) сигнал через диод VD2 и резистор R7 поступает на базу транзисторного ключа на VT2, к коллектору которого подключается вход электрической цепочки, через которую выполняется манипулирование частотой тонального генератора U1. Далее сигналы манипулируемого тонального генератора подаются на микрофонный вход передатчика, работающего в режиме SSB. Генератор при включенном терминале генерирует тон высокой частоты.

На контакт 5 (8) разъема поступает сигнал от демодулятора. Название контакта — сброс для передачи — CTS (Clear to Send). Сигнал на демодулятор поступает от приемника сначала на устройство U2 — полосовой фильтр, основное назначение которого состоит в создании достаточно узкой полосы пропускания для поступившего от приемника полезного сигнала и отфильтровывании сигнала помех от соседних работающих радиостанций. Затем сигнал поступает на устройство U3 — специальный частотный детектор, где тональные посылки разных частот превращаются в электрические сигналы разной полярности, которые перед поступлением на компьютер проходят еще одно устройство — U4. Цель этого устройства — отфильтровать выходящий сигнал от продуктов преобразования и выдать на разъем СОМ-порта электрический сигнал в точном соответствии с требованиями стандарта RS-232-C.

Контакт 7 (5) разъема COM-порта является общим для всех остальных контактов заземляющим. Он носит название — сигнальное заземление GND или SG (Signal Ground). Разъем DB25 имеет еще один заземляющий контакт 1 (в DB9 такого контакта нет). Этот контакт называется — защитное заземление — FG (Frame Ground) и служит только для соединения экранирующего корпуса модема с экранирующим корпусом компьютера. Контакты 7 (5) и 1 соединять между собой ни в коем случае нельзя!

Кроме перечисленных выше вводов и выводов, на модем следует подать от источника питания напряжение +5 В, +12 и −12 В, не забудьте общий провод (заземление) от источника питания для напряжения +5 В и двуполярного выпрямителя.

Отдельные узлы модема

Тональный генератор

Тональный генератор U1 выполнен по одному из широко известных вариантов. Принципиальная электрическая схема тонального генератора представлена на рис. 3.8.

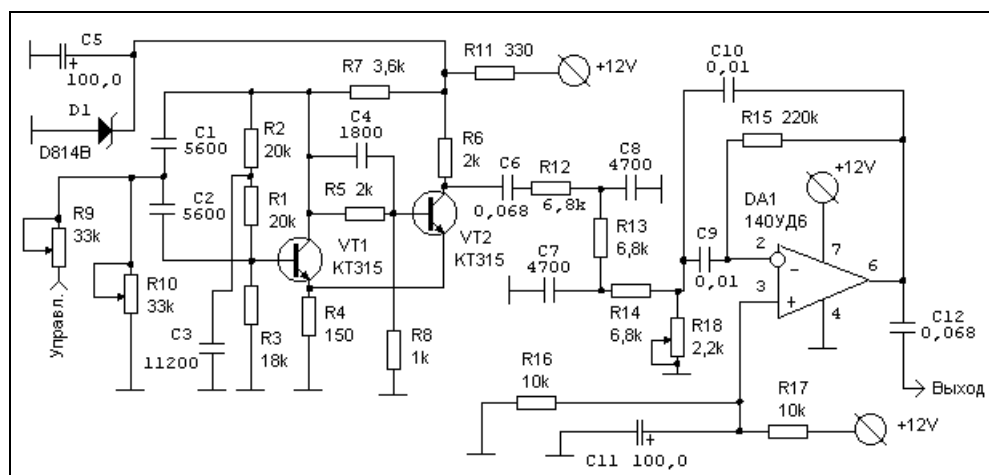


Рис. 3.8. Схема тонального генератора

Генератор выполнен на транзисторах VT1 и VT2 типа КТ315Б по схеме с обратной связью через двойной Т-мост, обладает высокой стабильностью и достаточно хорошим качеством сигнала при питании от стабилизированного источника. Конденсаторы C1, C2 и C3 должны иметь допуск не хуже 10%. На операционном усилителе DA1 типа К140УД6 выполнен полосовой фильтр, назначением которого является улучшение синусоидальности вы-

ходного сигнала. Для настройки частотомер подключается к точке выхода, регулировкой R10 устанавливается величина нижней частоты, а регулировкой R9 (при замкнутой на землю точке входа "управление") устанавливается величина верхней частоты. Изменением величины резистора R7 можно корректировать качество синусоиды генерируемого сигнала (только при осциллографическом контроле). Резистором R18 добиваются равной амплитуды для сигналов высокой и низкой частоты.

Полосовой фильтр

Полосовой фильтр U2 может иметь множество вариантов исполнения. В данном варианте модема применен сложный двухканальный вариант фильтра, блок-схема которого представлена на рис. 3.9. Сложный фильтр содержит в своем составе предварительный фильтр (блок U2.1), два фильтра каналов А и Б (блоки U2.2 и U2.3).

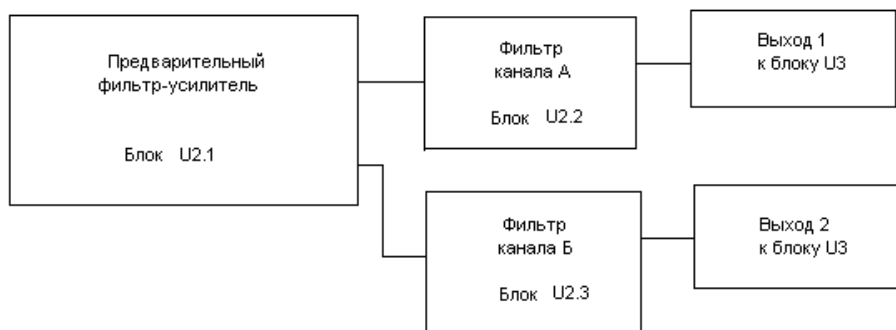


Рис. 3.9. Блок-схема фильтра

Предварительный фильтр представлен на рис. 3.10.

На операционном усилителе DA1 собран собственно сам предварительный фильтр, а на DA2 собран дополнительный усилитель сигналов НЧ. Если усиление приемника большое, то необходимости в дополнительном усилителе нет и каскад на DA2 можно не делать. Ось резистора R5 следует взять с регулировкой отверткой (под шлиц). Предварительный фильтр настраивается на среднюю частоту. После завершения настройки последующих фильтров, его следует слегка подстроить для достижения равной амплитуды выходного сигнала как на выходе канального фильтра А, так и на выходе фильтра Б.

На рис. 3.11 приводится схема одного канала фильтра А или Б. Каждый из них выполнен на трех операционных усилителях типа К140УД7(УД6) (или подобных). Для настройки фильтра сигнал от тонального генератора U1 подается на вход фильтра, при этом фильтр А настраивается на верхнюю аудиочастоту, а фильтр Б — на нижнюю. Для этого каскады на DA1, DA2

и DA3 резисторами R5 и R6 и R10 настраиваются поочередно на одну и ту же частоту.

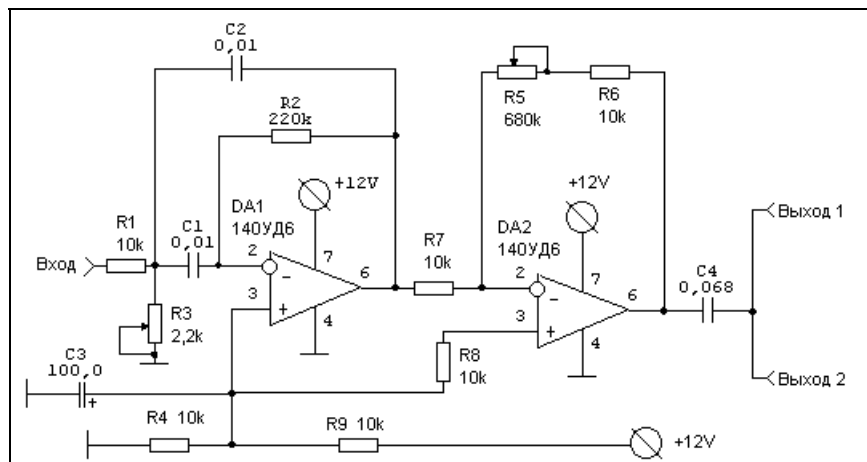


Рис. 3.10. Схема предварительного фильтра

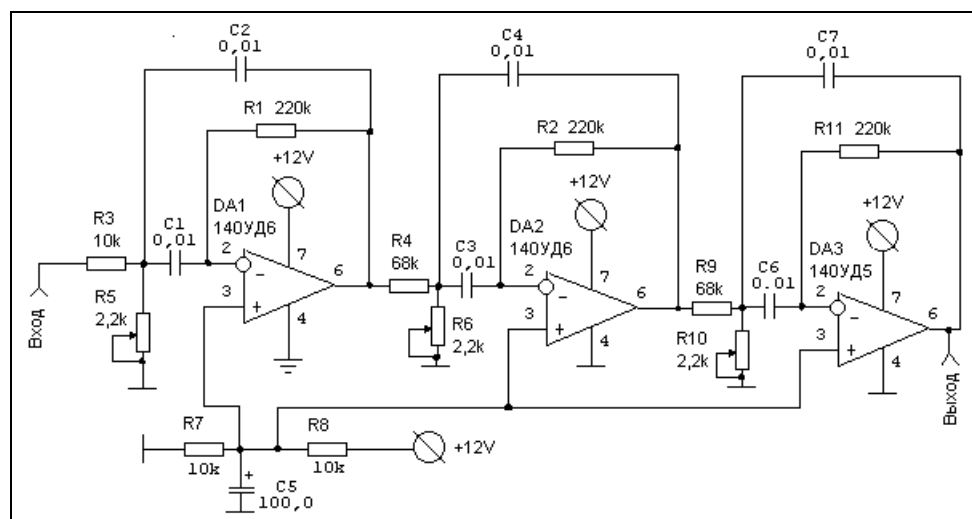


Рис. 3.11. Схема фильтра одного из каналов

Схема специального диодного детектора U3 представлена на рис. 3.12. В качестве детекторов применены мостовые двухполупериодные выпрямители. Это позволило значительно улучшить качество приема сигналов и отказаться от переключаемых конденсаторов в выходном каскаде. Схему управления

светодиодами можно упростить, используя в каждом случае по одному транзистору КТ315.

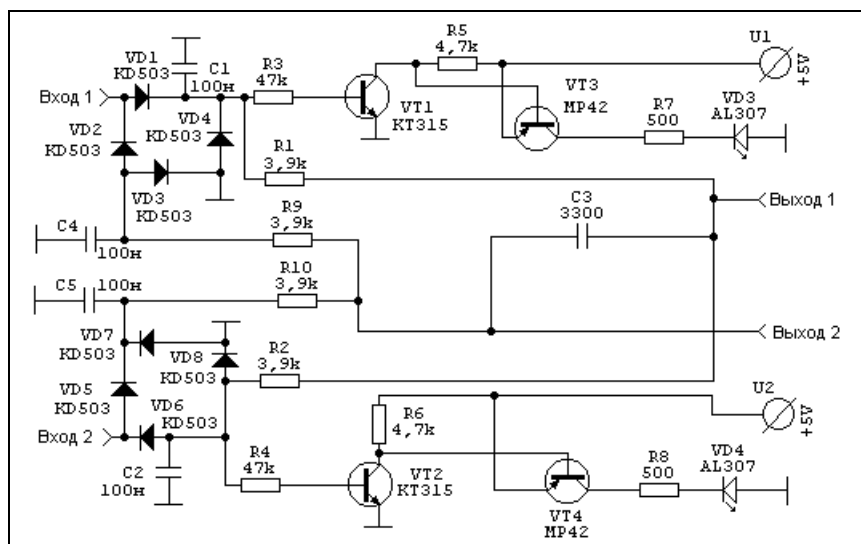


Рис. 3.12. Схема детектора

Выходное согласующее устройство

На рис. 3.13 показана принципиальная электрическая схема согласующего выходного устройства U4.

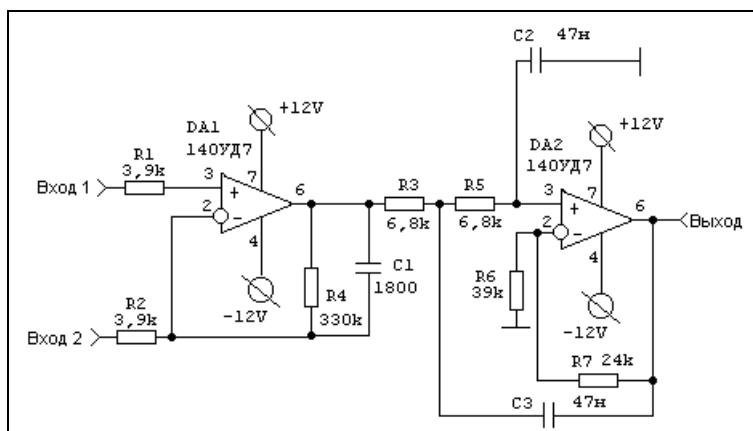


Рис. 3.13. Схема выходного устройства

На операционных усилителях DA1 и DA2 собран фильтр, который очищает полезный сигнал от продуктов преобразования и подает на вход последовательного порта компьютера сигнал со стандартными параметрами.

Настройка модема

Настройка модема чрезвычайно проста. Привожу необходимые этапы настройки.

Начать настройку модема следует с выбора величин рабочих частот. Поскольку модем в некоторых видах цифровой связи (например, RTTY) может работать с узкополосным сигналом, следует за основу при расчетах взять характеристики узкополосного (телеграфного) фильтра, установленного на вашем радиоприемнике. Работать предполагается всеми видами цифровой связи. Предположим, что радиоприемник на вашей станции имеет узкополосный фильтр с шириной полосы 300 Гц и пропускает частоты от 1000 до 1300 Гц. В этом случае величина средней частоты будет равна 1150 Гц. Учитывая величину сдвига частот, равную 200 Гц, определяем, что нижняя частота должна быть 1050 Гц, а верхняя — 1250 Гц.

Настройка тонального генератора (см. рис. 3.8) выполняется с использованием частотомера. Частотомер подключается к точке выхода генерируемого звукового сигнала. Регулировкой величины сопротивления резистора R10 устанавливается величина нижней частоты (1050 Гц), а регулировкой R9 (при замкнутой на землю точке входа "управление") устанавливается величина верхней частоты (1250 Гц). Изменением величины резистора R7 можно корректировать качество синусоиды генерируемого сигнала (только при осциллографическом контроле). Резистором R18 добиваются равной амплитуды для сигналов высокой и низкой частоты. Помните, что для RTTY используется разнос частот, равный 170 Гц, а AMTOR, PACTOR и Packet Radio (300 Бод) работают при разносе частот на 200 Гц. Если вы предполагаете работать только в режиме RTTY, то устанавливайте величину разнеса 170 Гц, если вы будете также работать и другими видами цифровой связи, то устанавливайте величину разнеса 200 Гц. Кстати, сигналы RTTY хорошо принимаются и при разносе частот в 200 Гц.

Настройку фильтров следует начать с настройки предварительного фильтра (см. рис. 3.10) на среднюю частоту (1150 Гц). Затем настроенным ранее тональным генератором настраиваются каналные фильтры (см. рис. 3.11). Сначала следует настроить фильтры каналов А и Б. Каждый из них выполнен на трех операционных усилителях типа К140УД7(УД6) (или подобных). Для настройки фильтра сигнал от тонального генератора U1 подается на вход фильтра, при этом фильтр А настраивается на верхнюю аудиочастоту (например, 1200 Гц), а фильтр Б — на нижнюю частоту (например, 1000 Гц). Для этого каскады на DA1, DA2 и DA3 резисторами R5 и R6 и R10 настраиваются поочередно на одну и ту же частоту, соответствующую данному ка-

налу. При настройке канальных фильтров следует непременно учитывать условие равенства между собой коэффициентов передачи канальных фильтров А и Б. Для этого на выход канальных фильтров поочередно подключается вольтметр переменного напряжения и, изменением величин сопротивлений межкаскадных резисторов (R4 или R9), следует добиться равенства напряжений на выходе при одинаковых напряжениях на входе.

После настройки фильтров каналов А и Б следует вернуться чуть назад и выполнить подстройку предварительного фильтра (см. рис. 3.10). На операционном усилителе DA1 собран собственно сам предварительный фильтр, а на DA2 собран дополнительный усилитель сигналов НЧ. Если усиление приемника большое, то необходимости в дополнительном усилителе нет, и каскад на DA2 можно не делать. Ось резистора R5 следует взять с регулировкой отверткой (под шлиц). Предварительный фильтр настраивается на среднюю частоту. Например, если верхняя частота составляет 1250 Гц, а нижняя — 1050 Гц, то предварительный фильтр должен быть настроен на частоту 1150 Гц. После завершения настройки последующих фильтров, его следует слегка подстроить для достижения равной амплитуды выходного сигнала как на выходе канального фильтра А, так и на выходе фильтра Б.

Специальный частотный детектор и выходной согласующий каскад настройки не требуют. Можно снять частотную характеристику детектора с подсоединенными к нему фильтрами и убедиться, что характеристика этого устройства полностью соответствует нормальной характеристике хорошего частотного детектора, например, детектора на микросхеме 564ГГ1.

Выходное согласующее устройство (см. рис. 3.13) также никакой наладки не требует. Нужно только для этого блока взять самые "свежие" операционные усилители, желательно не бывшие в употреблении. Дело в том, что мною наблюдались случаи, когда некоторые экземпляры операционных усилителей, прекрасно работающие в других устройствах, вдруг начинали при работе в этом устройстве "выбрасывать" какие-то импульсы, мешающие нормальной работе модема. Причину таких явлений я так и не нашел. Выходом из создавшегося положения всегда была замена операционного усилителя на заведомо исправный или новый.

Настраиваться на работающую в эфире пакетную станцию следует при уменьшенной до минимума громкости приемника. Настройку можно считать завершенной при попеременном мигании светодиодов в такт с принимаемыми сигналами. Затем громкость следует значительно увеличить, контролируя прием по тексту на экране компьютера. При этом не следует поддаваться панике, если светодиоды будут гореть постоянно. Эмулятор TNC TFPCX требует сильного сигнала, в то же время эмулятор TNC L2 (BayCom) сильных сигналов не любит.

Опыт показал, что пакетная станция с подобной самодельной техникой НА РАВНЫХ работает со станциями, оснащенными покупной аппаратурой производства известных фирм.

Другие варианты универсального модема

В разделе любого справочника по электропитанию радиоустройств можно найти схемы выпрямительных устройств с удвоением выпрямленного напряжения. Одна из таких схем изображена на рис. 3.14. Схемы удвоения применяются довольно часто, они четко работают в различных выпрямителях, например, для получения двухполярного блока питания слаботочных радиоустройств.

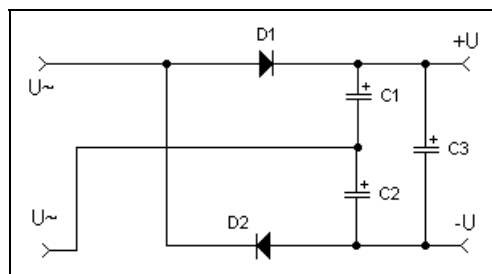


Рис. 3.14. Схема выпрямителя с удвоением напряжения

Кроме того, иногда появляется необходимость в еще большем увеличении выпрямленного напряжения, тогда на помощь приходят схемы с утроением напряжения, с учетверением напряжения и увеличением в большее количество раз. На рис. 3.15 приведена одна из схем с учетверением напряжения.

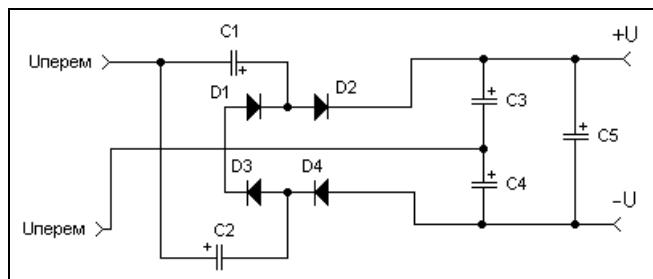


Рис. 3.15. Схема выпрямителя с учетверением напряжения

Однажды у меня появилась мысль использовать схему умножителя напряжения в качестве детектирующего устройства модема. Результат получился очень хорошим. С применением приведенных выше схем умножения выпрямленного напряжения мною разработаны еще два универсальных модема, предназначенных для работы с большинством из известных видов цифровой радиосвязи — MODEM21 и MODEM23.

Модем MODEM21

Схема модема MODEM21 практически не отличается от схемы модема MODEM22.

На рис. 3.7 представлена блок-схема модема MODEM22, но она также соответствует и MODEM21, все детали различных узлов модема рассчитаны на его работу со средней частотой около 1000 Гц. Модем работает на звуковых (аудио) частотах и совмещает в себе две основных составных части — передающую часть (модулятор) и приемную часть (демодулятор). Модулятор, в свою очередь, включает в себя устройство для включения и выключения передатчика и собственно модулятор — устройство для подачи на микрофонный вход SSB передатчика посылок от тонального генератора (U1). Демодулятор включает в себя полосовой фильтр на операционных усилителях (U2), специальный частотный детектор (U3) и выходной узел (U4). Предполагается изготовление каждого из узлов модема на отдельной плате, что позволит в дальнейшем безболезненно заменять неудачно выполненные узлы.

На блок-схеме слева указаны номера контактов разъема COM-порта для вариантов применения DB25 и DB9, справа указаны гнезда приемопередатчика, к которым подводится или от которых берется сигнал.

С контакта 4 (7) разъема COM-порта (здесь и далее первая цифра относится к разъему с 25 контактами, а цифра в скобках — к разъему с 9 контактами) берется сигнал для управления переключением передатчика прием/передача. Назначение этого контакта в системе RS-232-C — запрос для передачи, наименование — RTS (Request to send). Далее через диод VD1 и резистор R1 сигнал поступает на транзисторный переключатель, выполненный на транзисторе VT1. К цепи коллектора этого транзистора подключается катушка от реле "прием/передача", установленного на трансивере. При подаче на базу транзистора VT1 положительного напряжения реле срабатывает и включает трансивер на передачу.

Блок U1 представляет собой тональный генератор. Сигналы для манипуляции тонального генератора берутся с контакта 20 (4) разъема. Назначение этого контакта — готовность выходных данных — DTR (Data Terminal Ready). С контакта 20 (4) сигнал через диод VD2 и резистор R7 поступает на базу транзисторного ключа на VT2, к коллектору которого подключается вход электрической цепочки, через которую выполняется манипулирование частотой тонального генератора U1. Далее сигналы манипулируемого тонального генератора подаются на микрофонный вход передатчика, работающего в режиме SSB. Генератор при включенном терминале генерирует тон высокой частоты.

На контакт 5 (8) разъема поступает сигнал от демодулятора. Название контакта — сброс для передачи — CTS (Clear to Send). Сигнал на демодулятор поступает от приемника сначала на устройство U2 — полосовой фильтр, основное назначение которого состоит в создании достаточно узкой полосы

пропускания для поступившего от приемника полезного сигнала и отфильтровывании сигнала помех от соседних работающих радиостанций. Затем сигнал поступает на устройство U3 — специальный частотный детектор, где тональные послышки разных частот превращаются в электрические сигналы разной полярности, которые перед поступлением на компьютер проходят еще одно устройство U4, цель которого — отфильтровать выходящий сигнал от продуктов преобразования и выдать на разъем COM-порта электрический сигнал в точном соответствии с требованиями стандарта RS-232-C.

Контакт 7 (5) разъема COM-порта является общим для всех остальных контактов заземляющим. Он носит название — сигнальное заземление GND или SG (Signal Ground). Разъем DB25 имеет еще один заземляющий контакт 1 (в DB9 такого контакта нет). Этот контакт называется — защитное заземление — FG (Frame Ground) и служит только для соединения экранирующего корпуса модема с экранирующим корпусом компьютера. Контакты 7 (5) и 1 соединять между собой ни в коем случае нельзя!

Кроме перечисленных выше вводов и выводов, на модем следует подать от источника питания напряжение +5В, +12 и -12 В, не забудьте общий провод (заземление) от источника питания для напряжения +5 В и двуполярного выпрямителя.

Отдельные узлы модема

Тональный генератор U1 выполнен по одному из широко известных вариантов. Принципиальная электрическая схема тонального генератора представлена на рис. 3.8.

Генератор выполнен на транзисторах VT1 и VT2 типа КТ315Б по схеме с обратной связью через двойной Т-мост, обладает высокой стабильностью и достаточно хорошим качеством сигнала при питании от стабилизированного источника. Конденсаторы C1, C2 и C3 должны иметь допуск не хуже 10%. На операционном усилителе DA1 типа К140УД6 выполнен полосовой фильтр, назначением которого является улучшение синусоидальности выходного сигнала. Для настройки частотомер подключается к точке выхода, регулировкой R10 устанавливается величина нижней частоты, а регулировкой R9 (при замкнутой на землю точки входа "управление") устанавливается величина верхней частоты. Изменением величины резистора R7 можно корректировать качество синусоиды генерируемого сигнала (только при осциллографическом контроле). Резистором R18 добиваются равной амплитуды для сигналов высокой и низкой частоты.

Полосовой фильтр U2 может иметь множество вариантов исполнения. В данном варианте модема применен сложный двухканальный вариант фильтра, блок-схема которого представлена на рис. 3.9. Сложный фильтр содержит в своем составе предварительный фильтр (блок U2.1), два фильтра каналов А и Б (блоки U2.2 и U2.3).

Предварительный фильтр представлен на рис. 3.10.

На операционном усилителе DA1 собран собственно сам предварительный фильтр, а на DA2 собран дополнительный усилитель сигналов НЧ. Если усиление приемника большое, то необходимости в дополнительном усилителе нет и каскад на DA2 можно не делать. Ось резистора R5 следует взять с регулировкой отверткой (под шлиц). Предварительный фильтр настраивается на среднюю частоту. После завершения настройки последующих фильтров, его следует слегка подстроить для достижения равной амплитуды выходного сигнала как на выходе канального фильтра А, так и на выходе фильтра Б.

На рис. 3.11 показана схема одного канала фильтра А или Б. Каждый из них выполнен на трех операционных усилителях типа К140УД7 (УД6) (или подобных) по одноканальному принципу. Для настройки фильтра сигнал от тонального генератора U1 подается на вход фильтра, при этом фильтр А настраивается на верхнюю аудиочастоту, а фильтр Б — на нижнюю. Для этого каскады на DA1, DA2 и DA3 резисторами R5 и R6 и R10 настраиваются поочередно на одну и ту же частоту.

Схема специального диодного детектора U3 представлена на рис. 3.16. В качестве детекторов применены мостовые двухполупериодные выпрямители. Это позволило значительно улучшить качество приема сигналов и отказаться от переключаемых конденсаторов в выходном каскаде. Схему управления светодиодами можно упростить, используя в каждом случае по одному транзистору КТ315.

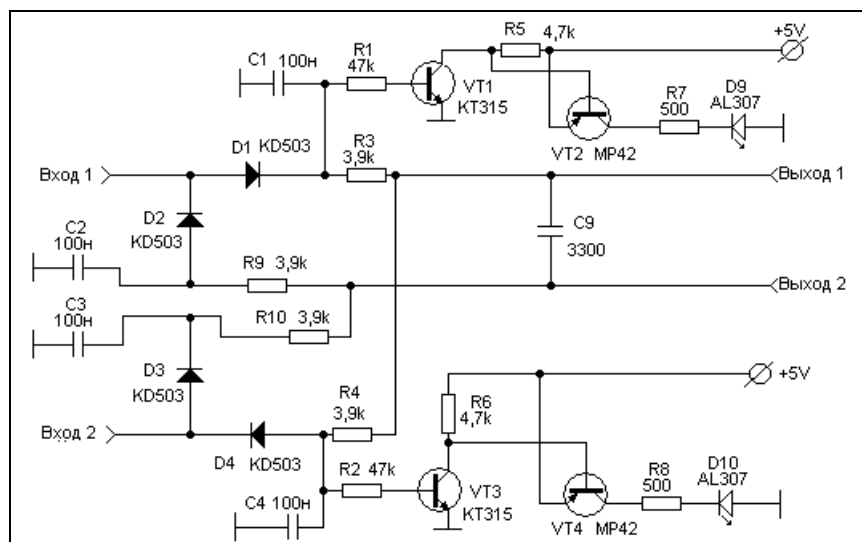


Рис. 3.16. Схема детектора

На рис. 3.13 представлена принципиальная электрическая схема согласующего выходного устройства U4.

На операционных усилителях DA1 и DA2 собран фильтр, который очищает полезный сигнал от продуктов преобразования и подает на вход последовательного порта компьютера сигнал со стандартными параметрами.

Настройка модема

Настройка модема чрезвычайно простая. Практически она ничем не отличается от настройки модема MODEM22. Привожу необходимые этапы настройки.

Начать настройку модема следует с выбора величин рабочих частот. Поскольку модем в некоторых видах цифровой связи (например, RTTY) может работать с узкополосным сигналом, следует за основу при расчетах взять характеристики узкополосного (телеграфного) фильтра. Работать предполагается всеми видами цифровой связи. Предположим, что радиоприемник на вашей станции имеет узкополосный фильтр с шириной полосы 300 Гц и пропускает частоты от 1000 до 1300 Гц. В этом случае величина средней частоты будет равна 1150 Гц. Учитывая величину сдвига частот, равную 200 Гц, определяем, что нижняя частота должна быть 1050 Гц, а верхняя — 1250 Гц.

Настройка тонального генератора (см. рис. 3.8) выполняется с использованием частотомера. Частотомер подключается к точке выхода генерируемого звукового сигнала. Регулировкой величины сопротивления резистора R10 устанавливается величина нижней частоты (1050 Гц), а регулировкой R9 (при замкнутой на землю точке входа "управление") устанавливается величина верхней частоты (1250 Гц). Изменением величины резистора R7 можно корректировать качество синусоиды генерируемого сигнала (только при осциллографическом контроле). Резистором R18 добиваются равной амплитуды для сигналов высокой и низкой частоты. Помните, что для RTTY используется разнос частот, равный 170 Гц, а AMTOR, PACTOR и Packet Radio (300 Бод) работают при разносе частот 200 Гц. Если вы предполагаете работать только в режиме RTTY, то устанавливайте величину разноса 170 Гц, если вы будете также работать и другими видами цифровой связи, то устанавливайте величину разноса 200 Гц. Кстати, сигналы RTTY хорошо принимаются и при разносе частот в 200 Гц.

Настройку фильтров следует начать с настройки предварительного фильтра (см. рис. 3.10) на среднюю частоту (1150 Гц). Затем настроенным ранее тональным генератором настраиваются каналные фильтры (см. рис. 3.11). Сначала следует настроить фильтры каналов А и Б. Каждый из них выполнен на трех операционных усилителях типа K140УД7 (УД6) (или подобных). Для настройки фильтра сигнал от тонального генератора U1 подается на

вход фильтра, при этом фильтр А настраивается на верхнюю аудиочастоту (например, 1200 Гц), а фильтр Б — на нижнюю частоту (например, 1000 Гц). Для этого каскады на DA1, DA2 и DA3 резисторами R5 и R6 и R10 настраиваются поочередно на одну и ту же частоту, соответствующую данному каналу. При настройке канальных фильтров следует непременно учитывать условие равенства между собой коэффициентов передачи канальных фильтров А и Б. Для этого на выход канальных фильтров поочередно подключается вольтметр переменного напряжения и, изменением величин сопротивлений межкаскадных резисторов (R4 или R9), следует добиться равенства напряжений на выходе при одинаковых напряжениях на входе.

После настройки фильтров каналов А и Б следует вернуться чуть назад и выполнить подстройку предварительного фильтра (см. рис. 3.10). На операционном усилителе DA1 собран собственно сам предварительный фильтр, а на DA2 собран дополнительный усилитель сигналов НЧ. Если усиление приемника большое, то необходимости в дополнительном усилителе нет и каскад на DA2 можно не делать. Ось резистора R5 следует взять с регулировкой отверткой (под шлиц). Предварительный фильтр настраивается на среднюю частоту. Например, если верхняя частота составляет 1250 Гц, а нижняя — 1050 Гц, то предварительный фильтр должен быть настроен на частоту 1150 Гц. После завершения настройки последующих фильтров, его следует слегка подстроить для достижения равной амплитуды выходного сигнала как на выходе канального фильтра А, так и на выходе фильтра Б.

Специальный частотный детектор и выходной согласующий каскад настройки не требуют. Можно снять частотную характеристику детектора с подсоединенными к нему фильтрами и убедиться, что характеристика этого устройства полностью соответствует нормальной характеристике хорошего частотного детектора, например, детектора на микросхеме 564ГГ1.

Выходное согласующее устройство (см. рис. 3.13) также никакой наладки не требует. Нужно только для этого блока взять самые "свежие" операционные усилители, желательно не бывшие в употреблении. Дело в том, что мною наблюдались случаи, когда некоторые экземпляры операционных усилителей, прекрасно работающие в других устройствах, вдруг начинали при работе "выбрасывать" какие-то импульсы, мешающие нормальной работе модема. Причину таких явлений я так и не нашел. Выходом из создавшегося положения всегда была замена операционного усилителя на заведомо исправный, или новый.

Применение в качестве детектора схемы выпрямителя с удвоением напряжения позволило увеличить чувствительность модема. Модем стал гораздо лучше принимать слабые сигналы, меньше реагировать на периодические затухания сигнала, часто встречающиеся при работе с дальними радиостанциями.

Модем MODEM23

Конструкция этого модема отличается от модема MODEM21 только другой схемой детектора. В основе новой схемы детектора, представленной на рис. 3.17, заложена схема выпрямителя с учетверением выпрямленного напряжения (см. рис. 3.15). Модем MODEM23 показал отличные результаты в работе и сейчас постоянно работает на радиостанции RA3XB.

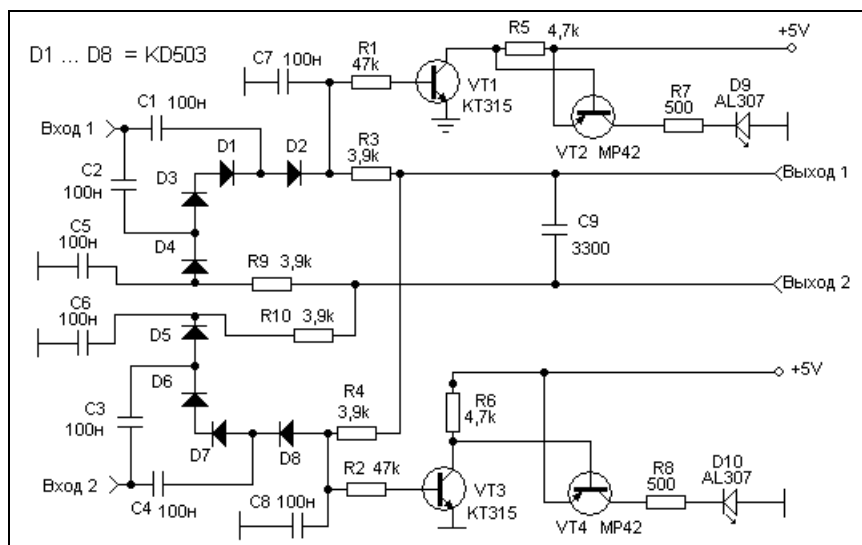


Рис. 3.17. Схема детектора для MODEM23

Настройка модема выполняется аналогично настройке MODEM21. Описание всех остальных узлов модема смотрите в описаниях модемов MODEM22 и MODEM21.

При изготовлении всех описанных выше модемов я не ставил целью организацию серийного производства этих аппаратов, поэтому разработкой монтажных плат для различных узлов модемов не занимался. Все свои радиотехнические конструкции я выполняю на одностороннем стеклотекстолите. При этом радиодетали располагаются со стороны медной обкладки, без сверления каких либо отверстий для выводов радиодеталей. Канавки на медной фольге прорезаю специальным ножом. Для экспериментатора эта технология очень даже удобна, обеспечивает быструю реализацию задуманного.

Прочие аппараты

Модем для частоты 215 кГц

Модем предназначен для работы всеми описанными в этой книге видами цифровой радиосвязи. Схема модема является экспериментальной, до конца

не отработана, но, по моему мнению, является исключительно перспективной из-за использования частот ПЧ (промежуточных частот). Модем по такой схеме можно создать для любых из применяющихся на практике величин первой промежуточной частоты — 110 кГц, 215 кГц, 465 кГц и 500 кГц. При этом каждой из величин ПЧ должны соответствовать параметры полосового фильтра и настроечных элементов детектора — частото задающих конденсаторов и резисторов. Предлагается всем желающим доработать модем по своему вкусу и возможностям, провести тестирование и сравнить работу этого модема с модемом, выполненным для работы на звуковых частотах.

Главные преимущества этой конструкции модема:

- ❑ возможность работы с различными скоростями без каких бы то ни было переключений;
- ❑ при приеме сигналов не имеет значения сдвиг частот (разница между частотами MARK и SPACE);
- ❑ упрощается настройка на принимаемую станцию;
- ❑ меньшие неприятности от эффекта Допплера при работе со спутниками.

Схема модема

На рис. 3.18 представлена блок-схема модема, все детали различных узлов модема рассчитаны на его работу со средней частотой примерно 215 кГц. Узлы модема проверялись на работоспособность в составе радиостанции, основу которой составлял радиоприемник Р-250М. Модем совмещает в себе две основных составных части — передающую часть (модулятор) и приемную часть (демодулятор).

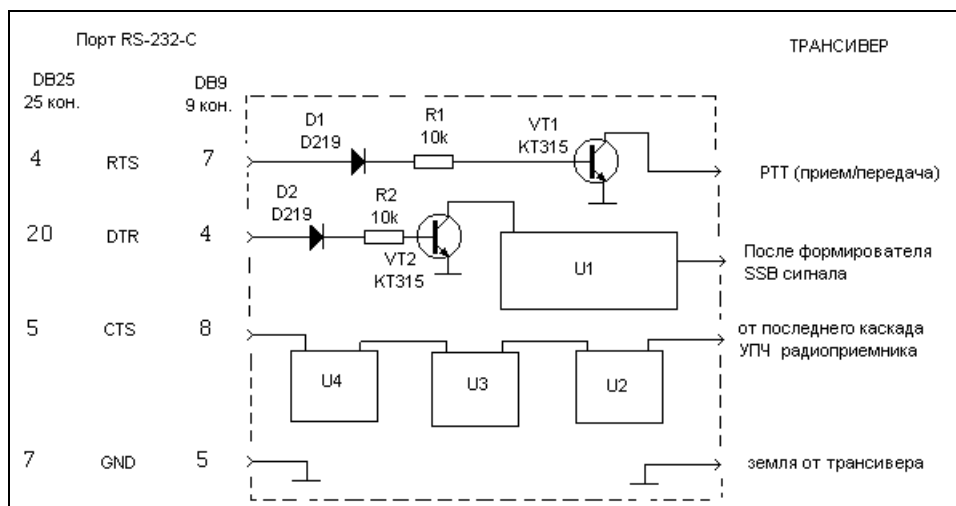


Рис. 3.18. Блок-схема модема

Модулятор включает в себя устройство для включения и выключения передатчика и собственно модулятор — устройство для подачи на вход передатчика посылок от генератора (U1). Демодулятор включает в себя полосовой фильтр (U2), специальный частотный детектор (U3) и выходной узел (U4).

Подключение модема к компьютеру должно выполняться через стандартный СОМ-порт с интерфейсом RS-232-С. Официальное ограничение по длине для соединения экранированным кабелем по стандарту RS-232-С составляет 15,2 м. На практике это расстояние должно быть как можно короче. Уровни напряжений на линиях разъема для логического нуля следует считать $-12...-3$ В, для логической единицы $+3...+12$ В. Промежуток от -3 до $+3$ В соответствует неопределенному значению. Каждый СОМ-порт имеет свой собственный разъем, который может иметь либо 25 контактов (DB25), либо 9 контактов (DB9).

На блок-схеме слева указаны номера контактов разъема СОМ-порта для вариантов применения DB25 и DB9, справа указаны гнезда приемопередатчика, к которым подводится или от которых берется сигнал. DB25 и DB9 — это разъемы СОМ-порта с 25 выводами и 9 выводами.

Компьютер через контакты 4 (7) разъема СОМ-порта (здесь и далее первая цифра относится к разъему с 25 контактами, а цифра в скобках — к разъему с 9 контактами) подает сигнал для управления переключением передатчика прием/передача. Назначение этого контакта в системе RS-232-С — запрос для передачи, наименование — RTS (Request to send). Если компьютер устанавливает на этом выводе положительное напряжение, то передатчик радиостанции включится для работы на передачу и пока положительная величина напряжения не будет отключена, передатчик будет работать на передачу. Чтобы выключить передатчик, компьютер должен подать на этот вывод отрицательное напряжение. При команде на передачу сигнал через диод VD1 и резистор R1 поступает на транзисторный переключатель, выполненный на транзисторе VT1. К цепи коллектора этого транзистора подключается устройство, предназначенное для переключения "прием/передача", установленного на передатчике. При подаче на базу транзистора VT1 положительного напряжения устройство переключения срабатывает и включает радиостанцию на передачу.

Блок U1 представляет собой специальный генератор частоты 215 кГц (см. рис. 3.19). Сигналы для манипуляции генератора берутся с контакта 20 (4) разъема.

Чтобы наложить сигналы используемого вида цифровой связи на несущую частоту, компьютер использует контакт 20 (4) разъема СОМ-порта. Назначение этого контакта — готовность выходных данных — DTR (Data Terminal Ready). Если компьютер устанавливает на этом контакте (выводе) положительное напряжение, то сигнал от контакта 20 (4) через диод VD2 и резистор R7 поступает на базу транзисторного ключа на VT2, который в тот же

момент включает в работу на определенной частоте устройство U1, являющееся генератором ПЧ. Генерируемая этим генератором частота складывается с несущей частотой передатчика и в эфир идет полезный сигнал. Получается так, что транзистор VT2 выполняет роль ключа и выполняет процесс модуляции путем манипуляции генератора ПЧ. Далее сигналы манипулируемого генератора подаются на вход передатчика, специально предусмотренный для сигналов промежуточной частоты (подается на смесительный каскад вместо сигналов от формирователя SSB).

С контакта 5 (8) разъема COM-порта компьютер принимает сигнал от демодулятора. Название контакта — сброс для передачи — CTS (Clear to Send). Чтобы стать понятным компьютеру, сигнал от последнего каскада усиления ПЧ-приемника должен пройти несколько обработок. Сначала сигнал от приемника поступает на устройство U2 (см. рис. 3.21) — полосовой фильтр, основное назначение которого состоит в создании достаточно узкой полосы пропускания для поступившего от приемника сигнала ПЧ и отфильтровывании сигнала помех от соседних работающих радиостанций. Затем сигнал поступает на устройство U3 (см. рис. 3.22) — специальный частотный детектор. Если на детектор поступает полезный сигнал, то на выходе детектора устанавливается положительное напряжение, если полезного сигнала в данный момент нет, то на выходе детектора напряжение равно нулю. Далее сигналы от детектора перед поступлением на компьютер проходят еще одно устройство — U4 (см. рис. 3.24), которое отфильтровывает выходящий сигнал от продуктов преобразования и выдает на разъем COM-порта электрический сигнал в точном соответствии с требованиями стандарта RS-232-C.

Контакт 7 (5) разъема COM-порта является общим для всех остальных контактов заземляющим. Он носит название — сигнальное заземление GND или SG (Signal Ground). Разъем DB25 имеет еще один заземляющий контакт 1 (в DB9 такого контакта нет). Этот контакт называется — защитное заземление — FG (Frame Ground) и служит только для соединения экранирующего корпуса модема с экранирующим корпусом компьютера. Контакты 7 (5) и 1 соединять между собой ни в коем случае нельзя!

Кроме перечисленных выше вводов и выводов, на модем следует подать от источника питания напряжение +5 В, +12 и -12 В, не забудьте общий провод (заземление) от источника питания для напряжения +5 В и двуполярного выпрямителя.

Отдельные узлы модема

Генератор ПЧ 215 кГц

Схема возможного варианта генератора ПЧ 215 кГц представлена на рис. 3.19. Эта схема экспериментальная, еще не прошла достаточной проверки и предназначена для применения достаточно опытными радиолюбителями-экспериментаторами.

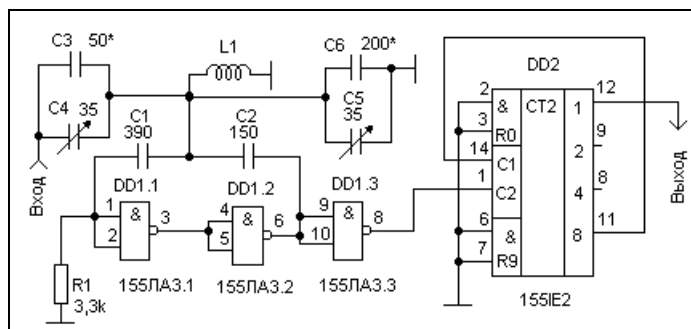


Рис. 3.19. Схема генератора

Собственно генератор работает на микросхемах DD1A и DD1B. Каскад на DD1C служит для улучшения характеристики выходного сигнала. Величина частоты на выходе DD1C должна быть в 10 раз больше рабочей частоты. Например, если рабочая частота составляет 215 кГц, то на выходе 8 микросхемы DD1C должна быть частота 2150 кГц. Следующий каскад на микросхеме DD2 работает в режиме делителя на 10.

Обратите внимание, что соединение выводов микросхемы DD2 выполнено так, чтобы последней была операция деления на два. Известно, что микросхема 155IE2 имеет в своем составе делитель на два и делитель на пять. В зависимости от порядка соединения выводов этой микросхемы можно назначать различные порядки выполнения делительных операций. Но для нашего варианта очень важно, чтобы в любом случае и при любых коэффициентах деления промежуточных звеньев последней операцией было бы деление на два.

С вывода 12 микросхемы DD2 сигнал в виде прямоугольных импульсов с частотой 215 кГц должен поступать на смесительный каскад передатчика вместо SSB-сигнала. Казалось бы, зачем усложнять схему и вводить делитель частоты, когда можно было бы сразу в генераторе получить частоту 215 кГц и подать ее в передатчик. Но в таком случае резко ухудшается стабильность частоты передатчика. Даже идеально выполненный генератор имеет какую-то нестабильность частоты, зависящую как от используемых радиодеталей, так и от температурных факторов. Если обозначить коэффициент деления буквой K , то в процессе деления стабильность частоты на выходе делителя будет в K раз лучше стабильности частоты самого генератора. В нашем случае стабильность частоты на выходе делителя будет в десять раз выше стабильности частоты на выходе генератора.

Можно сделать делитель на 40. При этом генератор должен генерировать частоту 8600 кГц. Это тоже реально возможный вариант, который может быть реализован по приведенной на рис. 3.19 схеме с добавлением еще двух делителей на 2 (можно на одной микросхеме 155TM2). Но получить еще более высокие частоты (выше 8600 кГц) на этом генераторе будет сложно.

Катушку L1 можно намотать на 4-секционном каркасе от ДВ-гетеродинной катушки малогабаритного радиоприемника. При этом катушка должна содержать 4×20 витков провода ПЭЛ 0,1.

Число витков катушки также можно выбрать по данным из табл. 3.1.

Настройку следует выполнять подбором резистора R1 и конденсаторов C3 и C6. Подбором величины сопротивления резистора добиваются работоспособности генератора, подбором величины емкости конденсатора C6 и подстройкой емкости C5 следует установить на выходе делителя (контролировать частотомером) высокую рабочую частоту (MARK), а подбором емкости конденсатора C3 и подстройкой емкости C4 (при замкнутой на землю точке "Вход") следует установить нижнюю частоту (SPACE). Установку частот следует контролировать радиоприемником с подключенным к нему демодулятором или достаточно точным частотомером.

Принципиальная электрическая схема еще одного из возможных вариантов генератора представлена на рис. 3.20. Задающий генератор выполнен по аналогичной с предыдущим вариантом схеме, но на МОП-микросхеме. Данные катушки и способы настройки смотрите в описании предыдущей схемы. Отличием является то, что следующий за генератором каскад на микросхеме DD2 561IP2 является одновременно и делителем на десять и формирователем синусоидального сигнала частотой 215 кГц.

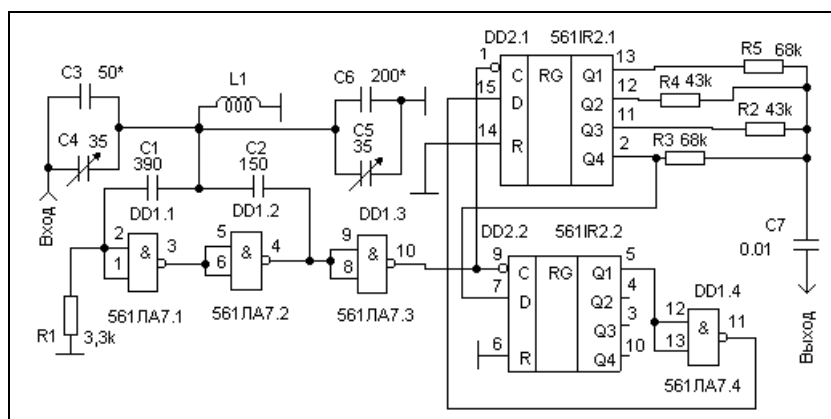


Рис. 3.20. Схема генератора

И в первом, и во втором вариантах генераторов излучается двухчастотный сигнал с непрерывающейся фазой. Это очень важно в смысле излучения помех. Тот факт, что в первом варианте используется несинусоидальный сигнал, большой роли, на мой взгляд, не имеет. Мне приходилось работать со станциями, использующими сигнал прямоугольной формы, и никаких неприятных явлений при этом не обнаруживалось.

Полосовой фильтр

Схема полосового фильтра проста. Сигнал на вход демодулятора берется с последнего каскада УПЧ-приемника, усиливается каскадом на транзисторе VT1, фильтруется ЭМФ215-1 кГц и снова усиливается каскадом на VT2. Принципиальная схема полосового фильтра представлена на рис. 3.21.

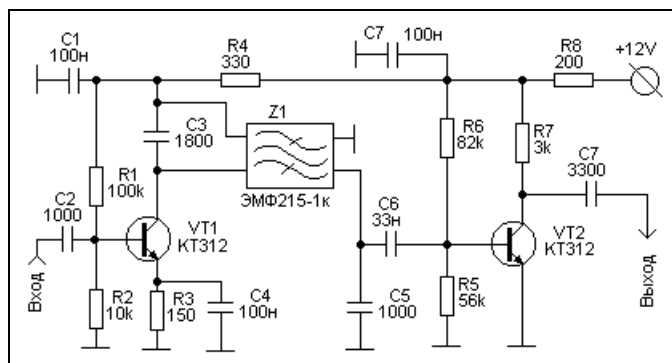


Рис. 3.21. Схема фильтра

Если у вас нет электромеханического фильтра, то можно поставить (с несколько худшими результатами) обычный двухконтурный фильтр со слабой связью между контурами или просто одиночный колебательный контур, настроенный на частоту 215 кГц.

Никакой специальной настройки фильтр, выполненный на электромеханическом фильтре, не требует. Следует проверить только исправность радиодеталей и отсутствие ошибок при монтаже.

Специальный частотный детектор

Описываемая конструкция модема использует экспериментальный вариант демодулятора сигналов, предназначенный для работы в режиме Packet Radio, но с успехом может работать в любых других цифровых видах радиосвязи. В демодуляторе применен разработанный автором оригинальный частотный детектор, отлично выполняющий свои функции в большом диапазоне частот от сотен герц до единиц мегагерц. Данный вариант детектора работает на частоте ПЧ 215 кГц и испытывался в работе совместно с приемником Р-250М и эмулятором TNC типа TFPCX в программном обеспечении.

Принципиальная электрическая схема специального частотного детектора приведена на рис. 3.22.

Каскад на транзисторе VT1 является согласующим элементом между транзисторными каскадами и ТТЛ-микросхемами, не допускает попадания сигналов с высокой амплитудой на входы микросхем с ТТЛ-уровнями сигналов.

Конденсатор C2 и резистор R2 являются дифференцирующей цепочкой для создания из сигнальной посылки укороченного импульса, DD1A и DD1B формируют фронты импульсов.

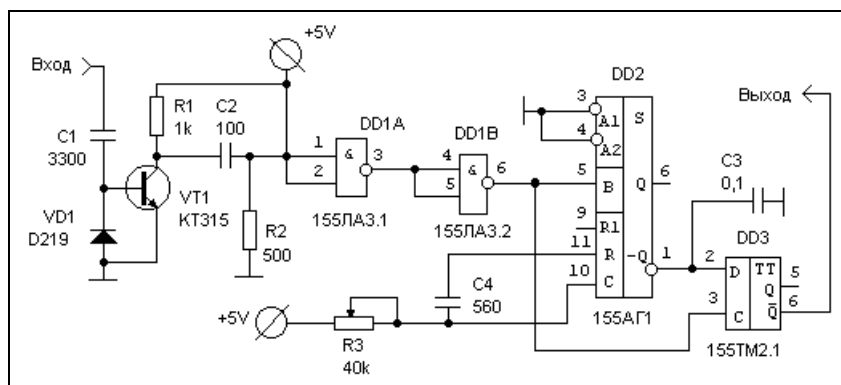


Рис. 3.22. Схема специального частотного детектора

Микросхемы DD2 и DD3 составляют собственно частотный детектор. Сформированный на элементах микросхемы DD1 укороченный импульс поступает одновременно на вывод 5 ждущего мультивибратора DD2 и на тактовый вход D-триггера DD3. Длительность импульса DD2 должна быть предварительно настроена элементами C4 и R3 таким образом, чтобы эта величина длительности была немного меньше длительности сигнала SPACE (низкая частота). Это легко выполняется с помощью простейшего сигнал-генератора ПЧ-частот и вольтметра. При этом, если на вход детектора будут поступать импульсы с частотой SPACE рабочего сигнала, или с более низкой частотой, то на выводе 6 триггера DD3 напряжение будет равно нулю.

Если на вход детектора начнут поступать импульсы с частотой выше SPACE (низкочастотного) рабочего сигнала, то на выходе триггера DD3 появится меандр с периодом, равным удвоенной частоте входных импульсов.

Выходное согласующее устройство

На рис. 3.23 представлена принципиальная электрическая схема согласующего выходного устройства U4.

На операционных усилителях DA1 и DA2 собран фильтр, который очищает полезный сигнал от продуктов преобразования и, одновременно, выдает для СОМ-порта полезные сигналы необходимой амплитуды и полярности. Практика показала достаточность такого согласователя практически для любых видов детекторов. Поскольку радиолюбитель зачастую вынужден использовать некондиционные детали, в схему добавлен потенциометр R4 (с дополнительными резисторами R2 и R5), которым необходимо устано-

тора действительно соответствует средней частоте ПЧ-канала. Изменением частоты в ту или иную сторону от средней частоты наблюдать скачкообразное увеличение показаний вольтметра и резкое падение напряжения до нуля. При этом должен загораться или гаснуть индикатор VD1 на выходном устройстве.

Подключить вольтметр постоянного напряжения на выход согласующего устройства. От звукового генератора через конденсатор емкостью 0,01 мкФ подать сигнал с частотой 300—600 Гц в точку "Вход" выходного устройства (на вход DA1). Перемещая движок потенциометра R4, добиться нулевых показаний вольтметра. Установленный режим работы выходного устройства при этом будет соответствовать режиму меандра на выходе.

Заключительную проверку можно сделать, изменяя частоту ПЧ в пределах полосы пропускания фильтра, при этом наблюдать скачкообразное изменение показаний вольтметра от нуля до максимума (и наоборот) *только* при переходе через установленную ранее частоту настройки детектора (среднюю частоту полосы пропускания фильтра ПЧ).

Все примененные в демодуляторе детали не имеют каких-либо особенностей и могут быть заменены на аналогичные.

Применение в демодуляторе оригинального частотного детектора позволяет добиться качественного приема сигналов без использования дефицитных специальных микросхем, используя только детали широкого применения. Кроме того, имеются следующие преимущества по сравнению с традиционными схемами демодуляторов:

- ❑ исключительная простота настройки детектора и демодулятора в целом;
- ❑ возможность применения более узкополосного активного фильтра принимаемых радиосигналов, что особенно важно при работе на коротких волнах;
- ❑ простая настройка на принимаемую радиостанцию, допускается уход частоты принимаемой радиостанции примерно на 80 Гц в коротковолновом диапазоне и на 400 Гц в УКВ, что значительно уменьшает неприятности, связанные с эффектом Доплера при работе со спутниками. Указанные величины весьма ориентировочны и зависят от полосы пропускания входного фильтра и настройки частоты переключения детектора;
- ❑ демодулятор работает не непосредственно с поступающими из эфира сигналами, а с аналогами этих сигналов, выработанными ждущими мультивибраторами, при этом полученные аналоги сигналов имеют более стабильные параметры;
- ❑ частотный детектор в процессе работы не производит никаких побочных явлений, которые могли бы создавать помехи приему.

К недостаткам следует отнести чувствительность к импульсным помехам.

Модем на 564ГГ1

Схема модема

Блок-схема модема, в котором используются микросхемы с ФАПЧ (Фазовая АвтоПодстройка Частоты) 564ГГ1 (561ГГ1), полностью совпадает с блок-схемой модема MODEM3. Поэтому далее я только коротко опишу различающиеся узлы модема. Прочие подробности смотрите в описании модема MODEM3.

На рис. 3.24 представлена блок-схема модема, все детали различных узлов модема рассчитаны на его работу в режиме Packet Radio со скоростью 300 Бод и со средней звуковой частотой примерно 1000 Гц. Модем работает на звуковых частотах и совмещает в себе две основных составных части — передающую часть (модулятор) и приемную часть (демодулятор).

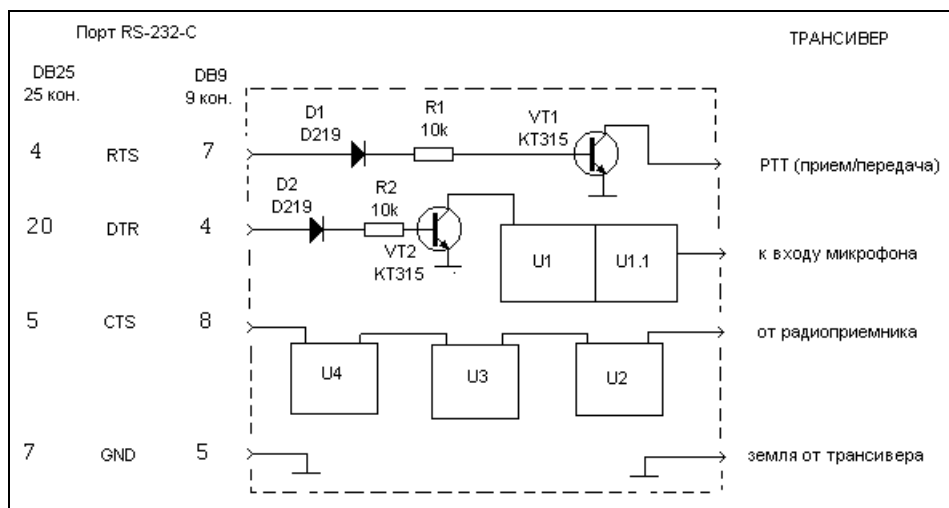


Рис. 3.24. Блок-схема модема

Модулятор включает в себя устройство для включения и выключения передатчика и собственно модулятор — устройство для подачи на вход передатчика посылок от тонального генератора. Собственно генератор обозначен как U1, выходной каскад генератора — U1.1. Демодулятор включает в себя полосовой фильтр на операционных усилителях (U2), специальный частотный детектор (U3) и выходной узел (U4).

Отдельные узлы модема

Тональный генератор

Тональный генератор U1 представлен на рис. 3.25. Генератор выполнен на микросхеме 564ГГ1 с фазовой автоподстройкой частоты.

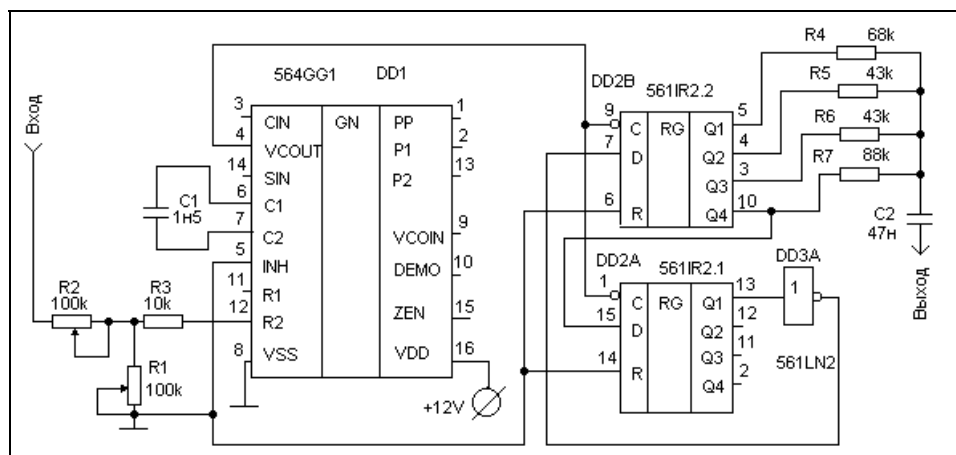


Рис. 3.25. Принципиальная электрическая схема тонального генератора

Для улучшения частотных характеристик генерируемого сигнала к генератору добавлен каскад U1.1 на транзисторе KT315, который служит фильтром нижних частот и позволяет регулировать величину амплитуды выходного сигнала. Схема этого каскада представлена на рис. 3.26.

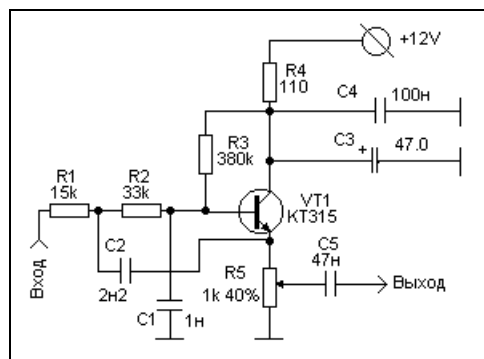


Рис. 3.26. Фильтр нижних частот

Полосовой фильтр

Полосовой фильтр U2 используется тот же, что и в модеме MODEM3.

Представленный на рис. 3.3 фильтр выполнен на двух операционных усилителях типа К140УД7(УД6) (или подобных) по одноканальному принципу.

Специальный частотный детектор

Специальный частотный детектор U3 выполнен на микросхеме 564ГГ1.

На рис. 3.27 приведена схема специального частотного детектора U3.

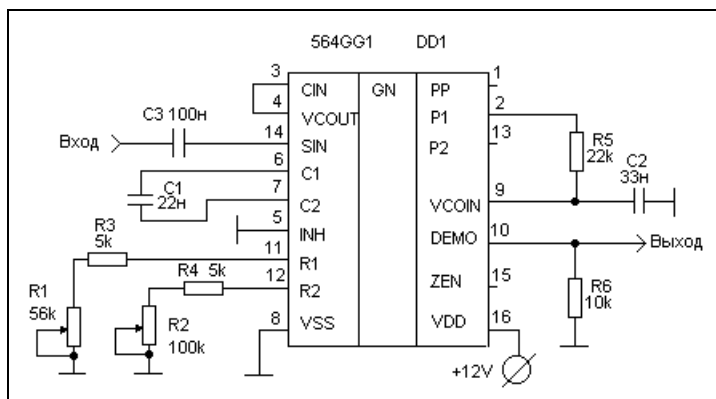


Рис. 3.27. Частотный детектор

Выходное согласующее устройство

Выходное согласующее устройство применено от модема MODEM3.

На рис. 3.5 представлена принципиальная электрическая схема согласующего выходного устройства U4.

Настройка модема

Изготовленный модем следует тщательно настроить. Особенно внимательно нужно выполнить настройку детектора, от нормальной работы которого зависит работа всего устройства. Настройка выполняется за несколько этапов.

Начать настройку модема следует с выбора величин рабочих частот. За основу при расчетах можно взять характеристики узкополосного (телеграфного) фильтра. Работать предполагается всеми видами цифровой связи. Предположим, что радиоприемник на вашей станции имеет узкополосный фильтр с шириной полосы 300 Гц и пропускает частоты от 1000 до 1300 Гц. В этом случае величина средней частоты будет равна 1150 Гц. Учитывая величину сдвига частот, равную 200 Гц, определяем, что нижняя частота должна быть 1050 Гц, а верхняя — 1250 Гц.

Далее проводим настройку тонального генератора (см. рис. 3.25). Для настройки частотомер подключается к точке выхода, регулировкой R1 устанавливается величина нижней частоты (например, 1050 Гц), а регулировкой R2 (при замкнутой на землю точке "Вход") устанавливается величина верх-

ней частоты (например, 1250 Гц). Величина амплитуды выходного сигнала должна измеряться вольтметром, подключенным к выходу тонального генератора.

Затем следует от звукового генератора подать на вход фильтра (см. рис. 3.3) сигнал SPACE с частотой 1050 Гц. Каскад на DA1 резистором R5 настраивается на эту частоту. Затем на фильтр от звукового генератора подается сигнал MARK с частотой 1250 Гц, при этом каскад на DA2 резистором R6 настраивается на эту частоту. Контролировать настройку каскадов следует вольтметром переменного напряжения, подключенным на выход фильтра. Если измерить частотную характеристику фильтра после настройки каскадов, то она должна выглядеть примерно так, как это показано на рис. 3.28 (кривая линия А).

Далее следует подключить вольтметр постоянного напряжения на выход частотного детектора (см. рис. 3.27). Подавая на вход модема поочередно сигналы от звукового генератора то 1250 Гц то 1050 Гц, нужно, вращая поочередно оси переменных резисторов R1 и R2, пытаться совместить пик положительного напряжения с частотой MARK, а минимум отрицательного напряжения с частотой SPACE. Следует иметь в виду, что резистором R2 устанавливается определенная величина частоты, а резистором R1 устанавливается величина разницы (сдвига) между частотами. Характеристика частотного детектора должна выглядеть примерно так же, как кривая В на рис. 3.28.

Настройку выходного каскада следует выполнить следующим образом.

1. Подключить вольтметр постоянного напряжения на выход согласующего устройства.
2. От звукового генератора через конденсатор емкостью 0,01 мкФ подать сигнал с частотой 300—600 Гц в точку "Вход" выходного устройства (на вход DA1).
3. Перемещая движок потенциометра R4, добиться нулевых показаний вольтметра. Установленный режим работы выходного устройства при этом будет соответствовать режиму меандра на выходе.

Настройку можно считать законченной, если характеристика полосового фильтра и характеристика частотного детектора соответствуют кривым на рис. 3.28.

Если сравнивать модемы MODEM22 и модем на 564ГГ1, то сразу бросается в глаза большое различие между входными полосовыми фильтрами. В двухканальном фильтре с острой настройкой MODEM22 суммарная ширина полосы пропускания двух каналов составляет не более 200 Гц, в то время как для нормальной работы частотного детектора на 564ГГ1 нужна ширина полосы не менее 400 Гц. Это говорит о том, что MODEM22 по этому параметру лучше приспособлен к работе на низкочастотных коротковолновых диа-

пазонах. В то же время сложность изготовления и настройки MODEM22 несравненно меньше и проще.

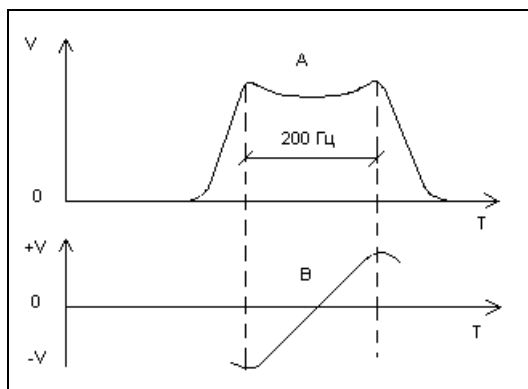


Рис. 3.28. Характеристики фильтра и детектора

Преселектор

В заключение данного раздела хочу поделиться с вами еще одной, очень нужной практически для любого коротковолновика, конструкцией.

Устройство называется "преселектор" и является предварительным усилителем высокой частоты с регулируемым сужением полосы пропускания за счет положительной обратной связи. Эффект Q-умножения.

Каждый, кто прослушал работу приемника с подключенным преселектором, непременно решает сделать себе такое устройство. Особенно рекомендую сделать такое устройство владельцам ламповых радиоприемников — Р-250, трансиверов UW3DI и всех прочих.

Проделайте такой опыт: отключите от приемника антенну и сравните уровень шумов на выходе приемника с подключенной и отключенной антенной. Если уровень шумов при подключении возрастает примерно в 10 раз, то вашему приемнику преселектор может и не понадобиться, а вот если уровень шумов возрастает в меньшее количество раз, то срочно начинайте делать для своего приемника такой преселектор.

Эффект значительного улучшения чувствительности и избирательности радиоприемника получается за счет применения мал шумящих транзисторов и сужения полосы пропускания входного контура преселектора за счет регулируемой положительной обратной связи. При изменении величины положительной обратной связи изменяется добротность имеющегося в схеме колебательного контура. Самая узкая полоса пропускания соответствует

наиболее возможной добротности контура и находится рядом с порогом генерации. Порог генерации — это такая величина положительной обратной связи, при превышении которой усилитель превращается в генератор.

Принципиальная электрическая схема преселектора представлена на рис. 3.29.

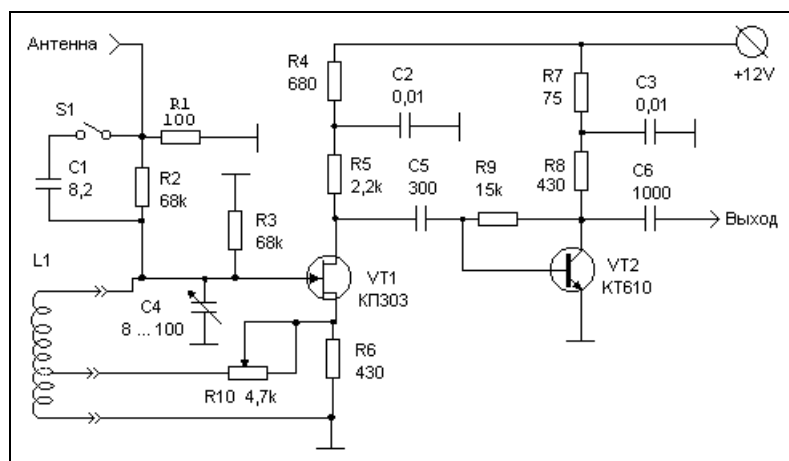


Рис. 3.29. Схема преселектора

Вместо указанной на схеме катушки L1 можно установить переключатель с набором катушек на все используемые диапазоны. Число витков катушки можно выбрать из табл. 3.1 или табл. 3.2. Отвод делается от 1/3 общего числа витков, считая снизу.

Таблица 3.1. Данные деталей колебательных контуров

Диапазон частот контура, МГц	Емкость конденсатора		Катушка индуктивности				
	Смакс, пФ	Смин, пФ	Инд, мкГн	Диаметр провода, мм	Диаметр витка, мм	Число витков	Длина намотки, мм
3,5 ... 7,8	100	20	20,3	0,64	25	38	32
3,5 ... 7,8	100	20	20,3	0,41	12	80	50
3,5 ... 7,8	100	20	20,3	0,64	38	23	19
3,5 ... 7,8	100	20	20,3	0,80	25	42	45
3,5 ... 9,6	150	20	14,0	0,80	38	17	15
3,5 ... 9,6	150	20	14,0	1,00	32	28	38

Таблица 3.1 (окончание)

Диапазон частот контура, МГц	Емкость конденсатора		Катушка индуктивности				
	Смакс, пФ	Смин, пФ	Инд, мкГн	Диаметр провода, мм	Диаметр витка, мм	Число витков	Длина намотки, мм
3,5 ... 9,6	150	20	14,0	0,64	25	25	22
3,5 ... 11,0	200	20	10,3	1,00	25	32	50
3,5 ... 11,0	200	20	10,3	0,64	12	45	25
7,0 ... 11,0	50	20	10,2	1,60	50	20	63
7,0 ... 15,6	100	20	5,2	0,64	25	16	19
7,0 ... 15,6	100	20	5,2	0,51	12	31	25
7,0 ... 15,6	100	20	5,2	1,00	38	11	19
7,0 ... 15,6	100	20	5,2	1,20	38	12	25
7,0 ... 19,0	150	20	3,4	1,60	32	13	32
7,0 ... 19,0	150	20	3,4	1,00	19	18	25
7,0 ... 19,0	150	20	3,4	1,00	25	14	25
7,0 ... 22,0	200	20	2,6	0,80	38	28	38
7,0 ... 22,0	200	20	2,6	0,64	12	20	19
7,0 ... 22,0	200	20	2,6	1,00	25	11	19
14 ... 22	50	20	2,6	1,6	38	9	25
14 ... 22	50	20	2,6	1,6	50	9	50
14 ... 31	100	20	1,3	1,6	25	8	19
14 ... 38	150	20	0,86	1,5	32	7	38
14 ... 44	200	20	0,65	1,0	38	5	25
28 ... 45	50	20	0,65	2,1	50	4	45
28 ... 45	50	20	0,65	2,0	12	13	38
28 ... 63	100	20	0,32	1,0	25	4	12
28 ... 63	100	20	0,32	2,0	19	5	19
28 ... 77	150	20	0,22	1,6	25	4	19

Табл. 3.2 также предназначена для подбора различных элементов колебательного контура. В начале каждой строки указана определенная величина индук-

тивности, а верхней цифрой каждого столбца является величина емкости конденсатора. Если у вас имеется катушка с индуктивностью 2 мкГн, то, используя совместно с ней конденсаторы емкостью от 100 до 1000 пФ, можно получить колебательный контур с частотой настройки от 11 000 до 3500 Гц.

Таблица 3.2. Частота в кГц для различных L и C

L, мкГн	Емкость, пФ									
	100	200	300	400	500	600	700	800	900	1000
1	15 800	11 100	9100	7900	7150	6503	6000	5670	5270	5000
2	11 100	9100	6503	5670	5000	4620	4230	4000	3760	3580
3	9100	6503	5270	4620	4110	3760	3490	3260	3060	2920
4	7900	5670	4620	4000	3580	3260	3000	2800	2660	2510
5	7150	5000	4110	3580	3200	2920	2680	2510	2380	2260
6	6503	4620	3760	3260	2920	2680	2460	2290	2160	2060
7	6000	4230	3490	3000	2680	2460	2280	2120	2000	1900
8	5670	4000	3260	2800	2510	2290	2120	1985	1875	1775
9	5270	3760	3060	2660	2380	2160	2000	1875	1761	1655
10	5000	3580	2920	2510	2260	2060	1900	1775	1655	1595
12	4620	3260	2660	2290	2060	1875	1730	1620	1530	1455
14	4230	3000	2460	2120	1900	1730	1610	1510	1410	1342
16	4000	2800	2290	1985	1775	1620	1510	1430	1327	1260
18	3760	2660	2160	1875	1655	1530	1410	1327	1250	1185
20	3580	2510	2060	1775	1595	1435	1342	1260	1185	1122
25	3200	2260	1840	1595	1420	1298	1205	1122	1060	1003
30	2920	2060	1655	1455	1298	1185	1080	1028	968	921
40	2510	1775	1455	1260	1122	1082	954	893	840	796
50	2260	1595	1298	1122	1003	921	850	796	750	712
60	2060	1455	1185	1028	921	840	778	728	685	650
70	1948	1342	1100	954	850	778	720	674	634	602
80	1775	1260	1028	893	841	728	689	630	593	563
90	1655	1185	968	840	750	685	635	593	560	532
100	1595	1122	921	896	712	650	613	563	532	502
120	1455	1028	840	728	650	593	550	514	485	460

Таблица 3.2 (окончание)

L, мкГн	Емкость, пФ									
	100	200	300	400	500	600	700	800	900	1000
140	1342	954	778	674	602	582	519	476	450	426
160	1260	893	745	630	563	514	476	446	430	398
180	1185	840	685	593	532	485	449	430	396	375
200	1122	796	650	563	504	460	426	398	375	356
250	1003	712	582	504	451	411	382	356	336	318
300	921	650	530	460	411	375	348	324	306	291
400	796	563	460	398	356	324	301	271	265	252
500	712	504	411	345	316	290	269	252	243	228
600	650	460	375	324	290	265	246	230	223	209
700	602	426	348	307	269	246	229	218	201	190

Модем и звуковая карта компьютера

Три технологии в цифровой радиосвязи

Если ввести понятие "технология выполнения цифровой любительской радиосвязи", то можно будет выделить три четко ограниченные технологии, которые отличаются друг от друга основными параметрами.

- Первая технология — классическая. Она характеризуется следующими параметрами.
 - Заложенная в компьютер программа только выполняет функции по обслуживанию экрана и файлов с передаваемыми и принимаемыми текстами.
 - Роль согласующего устройства между компьютером и радиостанцией выполняет специальный контроллер — TNC (Terminal Node Controller). TNC представляет собой сложный аппарат, который при передаче принимает от компьютера через COM-порт текстовые строки, производит кодирование каждого символа (буквы) из этой строки в токовые послышки и передает эти послышки в передатчик. При приеме он принимает от радиоприемника аналоговые сигналы, декодирует эти сигналы в буквы, складывает из полученных букв строки и через COM-порт передает готовые строки компьютеру. Переключает радиостанцию для работы на передачу или на прием.

- COM-порт компьютера с подключенным к нему TNC работает в нормальном режиме, соответствующем стандарту RS-232.
- Вторая технология — комбинированная. В этой технологии часть функций, которые в первой технологии выполняет TNC, передаются компьютеру.
- Загруженная в компьютер программа выполняет функции по обслуживанию экрана и файлов с передаваемыми и принимаемыми текстами. Кроме того, она при передаче производит кодирование каждого символа из передаваемой текстовой строки в токовые посылки и передает их модему. При приеме программа принимает от модема токовые посылки, декодирует эти посылки в буквы, складывает из этих букв слова и передает их для дальнейшей обработки.
 - Роль согласующего устройства между компьютером и радиостанцией играет специальный аппарат — модем. Модем значительно проще и дешевле, чем TNC. При передаче он принимает от компьютера через COM-порт токовые посылки, преобразует их в посылки с определенной частотой и передает на вход радиопередатчика. При приеме модем принимает от радиоприемника аналоговые сигналы, преобразует их в токовые посылки и передает компьютеру. Через модем компьютер переключает радиостанцию с приема на передачу и наоборот.
 - COM-порт работает в необычном для него режиме.
- Третья технология — работа через звуковую карту компьютера. Эта технология имеет много общего с технологией второй, только роль модема выполняет звуковая карта компьютера.
- Загруженная в компьютер программа выполняет функции по обслуживанию экрана и файлов с передаваемыми и принимаемыми текстами. Кроме того, при передаче она производит кодирование каждого символа из передаваемой текстовой строки в токовые посылки и передает их на звуковую карту компьютера. При приеме программа принимает от звуковой карты токовые посылки, декодирует эти посылки в буквы, складывает из этих букв слова и передает их для дальнейшей обработки.
 - Роль согласующего устройства между компьютером и радиостанцией играет звуковая карта компьютера. Такой вариант значительно проще и дешевле, чем TNC или модем. При передаче звуковая карта принимает от компьютера токовые посылки, преобразует их в посылки с определенной частотой и передает на вход радиопередатчика. При приеме звуковая карта принимает от радиоприемника аналоговые сигналы, преобразует их в токовые посылки и передает компьютеру.
 - COM-порт служит только для переключения радиостанции с приема на передачу и наоборот.

Каждая из этих технологий имеет свои плюсы и свои минусы. Рассмотрим некоторые из особенностей каждой технологии.

- ❑ Единственный, на мой взгляд, недостаток первой технологии заключается в значительной стоимости TNC. Преимуществом является отличное качество как при приеме, так и при передаче.
- ❑ Вторая технология позволяет иметь достаточно хорошее качество приема и передачи, которое в большой степени зависит от конструкции и настройки модема. Для радиолюбителей, умеющих держать в руке паяльник, это очень хороший вариант. Большинство из имеющихся в настоящее время программ по различным цифровым видам связи предназначены именно для работы с различными модемами, хотя могут отлично работать и с TNC.
- ❑ Третья технология имеет специфическую особенность. Дело в том, что в компьютере могут использоваться и очень дешевые звуковые карты, и очень дорогие. При этом дорогие карты, имеющие в своем составе специальный процессор, могут обеспечить хорошее качество сигнала. Но эти карты очень дорогие и не каждый из наших радиолюбителей сегодня в состоянии их приобрести. Дешевые звуковые карты тоже могут во многих случаях удовлетворить потребности некоторых радиолюбителей, но, как показал опыт, не всем этот вариант нравится. С аппаратными модемами получается лучше, это мое мнение.

Существует и еще одна особенность при работе со звуковыми картами. Под управлением Windows 95/98 работают практически все имеющиеся на отечественном рынке звуковые карты, а вот под управлением MS-DOS работают только те карты, драйверы которых имеются в данной программе.

Что такое TNC

Зарубежными фирмами разработано много разных моделей TNC. В нашей стране также делались попытки создать свои конструкции на базе зарубежных компонентов. В некоторых случаях это удавалось, так что у радиолюбителей можно встретить и такие конструкции. Основная масса изготавливаемых в нашей стране аппаратов TNC делается по схеме контроллера TNC-2.

Пакетный контроллер TNC-2 предназначен для проведения радиосвязей видом цифровой связи Packet Radio в следующих режимах:

- ❑ в режиме терминальной станции;
- ❑ в режиме ретранслятора (digi);
- ❑ в режиме узла (node);
- ❑ в режиме почтового ящика (PMS или MBBS);
- ❑ совместно с программой компьютера позволяет организовать работу пакетного MBBS и радио TCP/IP-систем для работы в сети Интернет.

Технические характеристики

Ниже привожу технические характеристики TNC-2, которыми обладают практически все изготавливаемые в нашей стране подобные аппараты.

- ☐ Протокол обмена AX.25.
- ☐ Скорость передачи по радиоканалу с внутренним модемом (имеется в виду модем на импортной микросхеме TCM3105N, встроенный в плату контроллера и предназначенный для работы на УКВ) — 1200/2400 бит/с.
- ☐ Скорость передачи по радиоканалу с внешним модемом (имеется в виду любой внешний модем, подключенный к специальному разъему контроллера и предназначенный для работы на определенной скорости) — 300, 4800, 9600 бит/с.
- ☐ Скорость обмена между контроллером и компьютером по COM-порту (режим асинхронный RS232) — 1200, 2400, 4800, 9600 бит/с.
- ☐ Программное обеспечение компьютера, с которым может нормально работать контроллер — TAPR-совместимое, TheNet, DED, KISS, ROSE.
- ☐ Русифицированный персональный почтовый ящик (PMS) размером 15 600 байт.
- ☐ Энергонезависимое ОЗУ (в некоторых аппаратах) — 32 Кбайт.
- ☐ Напряжение питания — 9 В.
- ☐ Ток потребления — 0,25 А.
- ☐ Светодиодные индикаторы — PWR (индикатор включения), CON (индикатор соединения с радиостанцией корреспондента), STA (статус — состояние контроллера), PTT (режим передачи), DCD (наличие несущей частоты на входе модема).

Состав контроллера

Контроллер пакетной связи состоит из двух основных узлов:

- ☐ пакетного адаптера, который представляет собой микропроцессорный комплект (мини-ЭВМ);
- ☐ модема.

Пакетный адаптер, представляющий из себя микроконтроллер, выполнен, как правило, на процессоре (CPU) Z80. В состав процессорного комплекта входят микросхемы ОЗУ и ПЗУ, двухканальный последовательный порт Z80A SIO-0. Первый канал порта программируется для работы с модемом, второй канал предназначается для работы непосредственно с компьютером. В этот же комплект обязательно входит генератор опорной частоты, выполненный на кварце с частотой 4,9152 или 9,830 МГц. Частота кварцевого генератора проходит несколько каскадов деления частоты, что позволяет работать с различными скоростями.

Программа, хранящаяся в ПЗУ, управляет работой микропроцессора. ОЗУ служит для хранения информации, как вводимой с клавиатуры, так и принятой из эфира. Формирователь пакетов производит упаковку информации в пакеты для передачи ее корреспонденту. Таймер поддерживает временные интервалы в момент установления соединения, во время обмена информацией и подтверждения, во время рассоединения.

Кроме того, в комплект непременно входят микросхемы, выполняющие роли:

- ☐ диспетчера памяти;
- ☐ преобразователя NRZ/NRZI;
- ☐ драйверов индикации;
- ☐ коммутатора прием/передача;
- ☐ преобразователя уровней TTL/RS232;
- ☐ стабилизатора напряжения.

Конструктивно контроллер выполняется на двух или на одной печатной плате. Если в качестве модема работает микросхема TCM3105, то контроллер выполняется на одной плате размером (примерно) 160×115 мм. Если для модема используются отечественные радиодетали, то модем выполняется на второй плате, примерно с такими же размерами.

Работоспособность цифровой части контроллера определяется при включении — если нет никаких ошибок или появившихся дефектов, то при включении сначала загораются, а потом гаснут два светодиода — CON и STA, а на экране появляется заставка контроллера.

Команды TNC-2

Для того чтобы вы смогли представить возможности этого аппарата, привожу перечень команд, которые пользователь может посредством клавиатуры компьютера передать на контроллер для выполнения определенных заданий.

- ☐ 3dpart ON/OFF
 - ON — разрешено оставлять сообщения в PMS (персональный почтовый ящик) для любого адресата;
 - OFF — разрешено оставлять сообщения в PMS только для/от адресата MYPCall.
- ☐ 8bitconv ON/OFF
 - ON — в пакете передается 8 битов данных в разговорном режиме (расширенный набор ASCII);
 - OFF — в пакете передается 7 битов данных в разговорном режиме (стандартный набор ASCII).

☐ Addrdisp ON/OFF

- ON — на экран выводятся позывные из пакетов, отображаемых по команде MONITOR;
- OFF — на экран не выводятся позывные из пакетов, отображаемых по команде MONITOR.

☐ AMonth ON/OFF

- ON — месяц в дате выводится в буквенном формате;
- OFF — месяц в дате выводится в цифровом формате.

☐ AUTOFwd ON/OFF

- ON — все сообщения в PMS будут автоматически маркированы для форвардинга;
- OFF — сообщения в PMS не будут маркированы для форвардинга.

☐ Autolf ON/OFF

- ON — код перевода строки <LF> вставляется в вывод на терминал после каждого кода перевода каретки <CR>;
- OFF — код <LF> не вставляется после <CR>.

☐ AWlen 7/8

7/8 — установка длины слова данных для параметров связи контроллера и терминала.

☐ Ax25l2v2 ON/OFF

- ON — контроллер использует вторую версию протокола AX25;
- OFF — контроллер использует первую версию протокола AX25.

☐ AXDelay n

- $n = 0-180$, число, определяющее задержку при использовании аналогового ретранслятора в 10 мс интервалах;
- общее время ожидания между включением передатчика и посылкой данных в эфир определяется как сумма TXDelay + AXDelay. Данная команда работает в сочетании с AXHang.

☐ AXHang n

- $n = 0-20$, число, определяющее компенсацию задержки отключения аналогового ретранслятора в 100 мс интервалах. Параметр используется для повышения пропускной способности канала аналогового ретранслятора, когда аналоговый ретранслятор имеет время задержки отключения более 100 мс;
- если контроллер определяет, что за время задержки отключения ретранслятора сигнал проходит полностью, то задержка AXDelay не добавляется.

☐ BBSmsgs ON/OFF

Команда используется при включении контроллера в BBS-систему и определяет характер вывода на экран сообщений в командном и разговорном режимах.

☐ Beacon Every/After n

- Every — маяк включается периодически через интервал времени n;
- After — маяк включается однократно через интервал времени n после каждого отсутствия сигнала в радиоканале;
- N = 0-250, число, определяющее время в 10 с интервалах. Значение n = 0 запрещает включение маяка. Маяк посылает текст, определенный командой BText.

☐ BKondel ON/OFF

- ON — при удалении символа на терминал посылается последовательность <Backspace>, <Пробел>, <Backspace>;
- OFF — при удалении символа на терминал посылается <Backslash>, (\).

☐ BBreak ON/OFF

- ON — контроллер реагирует на команду BBreak;
- OFF — контроллер не реагирует на команду BBreak.

☐ BText текст

текст — любая комбинация из 120 символов и пробелов, посылаемая в эфир при включении маяка.

Для удаления текста используйте в первой позиции текста символы % или &.

☐ BUdlist ON/OFF

- ON — игнорировать сигналы от корреспондентов, позывные которых не включены в команду LCalls;
- OFF — игнорировать сигналы от корреспондентов, позывные которых включены в команду LCalls.

Эта команда работает совместно с командой LCalls и определяет, что будет выводиться на экран при Monitor ON.

☐ Bye

Команда разъединения для удаленных пользователей PMS.

☐ CALibrate

Команда настройки радиомодема. Режимы:

- <SPACE> — переключение тона с его передачей в эфир;
- <K> — включение/выключение несущей частоты передатчика;

- <D> — одновременное излучение обоих тонов;
- <Q> — выход из режима настройки.

☐ CALSet n

- n = 0-65535, число, определяющее частоты калибровки:
 - ◇ для частоты модуляции $f_n = (525,000 / f) + 1$;
 - ◇ для частоты демодуляции $f_n = (262,500 / f) + 1$.

☐ CANline n

n = 0-7fH, число, определяющее ASCII-код символа. Команда определяет код символа отмены текущей строки.

☐ CANPac n

n = 0-7fH, число, определяющее ASCII-код символа. Команда определяет код символа отмены текущего пакета.

☐ CBell ON/OFF

- ON — в момент соединения подается звуковой сигнал;
- OFF — в момент соединения не подается звуковой сигнал.

☐ CHeck n

n = 0-250, число, определяющее тайм-аут соединения в 10 с интервалах.

☐ CLKADJ n

n = 0-65535, число, корректирующее ход внутренних часов контроллера при повышенной активности радиопорта. Коэффициент коррекции в % равен $100 + (9,16667 \times 1 / n)$.

☐ CLKSet ON/OFF

- ON — дата и время внутренних часов контроллера устанавливаются по параметрам даты и времени "home-BBS";
- OFF — контроллер не использует внешнюю загрузку даты и времени во внутренние часы.

☐ CMdtime n

n = 0-250, число, определяющее тайм-аут соединения в режиме Transparent в 1 с интервалах.

☐ CMSg ON/OFF

- ON — в момент соединения будет послан текст, установленный командой CText;
- OFF — текст посылаться не будет.

☐ MSGDisc ON/OFF

- ON — контроллер инициирует разъединение после получения подтверждения на CText, посылаемый командой CMSg;

- OFF — контроллер не будет инициировать разъединение после послы-ки текста CText командой CMSg.

☐ COMmand n

n = 0-7fH, число, определяющее ASCII-код символа. Команда определяет код символа перехода в командный режим.

☐ CONMode Conv/Tran

- Conv — контроллер переходит в разговорный режим в момент соединения;
- Tran — контроллер переходит в прозрачный режим в момент соединения.

☐ Connect call#1 [Via call#2 [,call#3..., call#9]]

- call#1 — позывной корреспондента, с которым нужно соединиться;
- call#2 — позывной ретранслятора, через который будет происходить соединение.

Если NEwmode ON, контроллер в момент соединения перейдет в разговорный или прозрачный режим, как указано в команде CONMode.

Если NEwmode OFF, контроллер после разъединения перейдет в разговорный или прозрачный режим.

☐ CONOk ON/OFF

- ON — контроллер разрешает соединение с ним корреспондентов;
- OFF — контроллер не разрешает соединение с ним корреспондентов.

☐ CONPerm ON/OFF

- ON — контроллер не будет переходить в режим разъединения даже в случае, когда все попытки возобновления соединения Retry будут безрезультатны;
- OFF — соединение и разъединение контроллера будут происходить как обычно.

☐ CONSTamp ON/OFF

- ON — в момент соединения на экран будут выводиться дата и время;
- OFF — в момент соединения дата и время не будут выводиться на экран.

☐ CONVerse или K

Команда перевода контроллера из командного в разговорный режим.

☐ CPactime ON/OFF

- ON — в разговорном режиме тайм-аут передачи пакета инициирован;
- OFF — в разговорном режиме тайм-аут не инициирован.

Если в разговорном режиме `CPactime ON`, вводимый текст будет передаваться автоматически через интервал времени, определяемый командой `PACTime`.

☐ `CR ON/OFF`

- `ON` — в разговорном режиме к передаваемым пакетам будет добавляться код инициации передачи, устанавливаемый командой `Sendpac` (обычно `<CR>`);
- `OFF` — код инициации передачи не будет добавляться к передаваемым пакетам в разговорном режиме.

☐ `CRAfter ON/OFF`

- `ON` — символ `<CR>` посылается на терминал после каждого отображаемого командой `MONITOR` кадра;
- `OFF` — символ `<CR>` не посылается на терминал после каждого кадра.

☐ `CStatus`

Команда отображения состояния всех каналов.

☐ `CText текст`

текст — любая комбинация из 120 символов и пробелов. Команда обеспечивает посылку текста другому контроллеру в момент установления соединения, если `CMSg ON`. Текст может состоять из нескольких строк, для чего необходимо использовать символ, определяемый командой `PASS` (обычно `<Ctrl>+<V>`). Для стирания текста можно воспользоваться командой `RESET` или ввести символы `%` или `&`.

☐ `CWid Every/After n`

- `EVERY` — передавать позывной телеграфом через регулярные интервалы времени;
- `AFTER` — передавать позывной телеграфом однократно через определенный интервал времени `n` после отсутствия в канале сигнала. `n = 0-250`, число, определяющее время в 100 с интервалах. Значение `n = 0` запрещает передачу позывного.

☐ `CWidtext текст`

текст — любая комбинация из 32 символов, цифр и пробелов, посылаемая в эфир командой `CWid`.

☐ `CWLen n`

`n = 1-7`, число, определяющее относительную длину точки при передаче позывного кодом Морзе по команде `CWid`.

☐ `Daytime ггммддччмм`

`ггммддччмм` — установка текущей даты и времени в следующей последовательности: `гг` — две последние цифры года, `мм` — месяц (01-12), `дд` — день (01-31), `чч` — час (00-23), `мм` — минуты (00-59).

☐ DAYUsa ON/OFF

- ON — дата отображается в формате: месяц/день/год;
- OFF — дата отображается в формате: день-месяц-год.

☐ DELeTe ON/OFF

- ON — стирание символа клавишей <Delete> (7fH);
- OFF — стирание символа клавишей <Backspace> (08H).

☐ DIGipeat ON/OFF

- ON — контроллер может работать как цифровой ретранслятор;
- OFF — контроллер не может работать как цифровой ретранслятор.

☐ Disconnect

Команда контроллеру произвести разъединение.

☐ DISPlay [тип]

Команда вывода на экран параметров контроллера. Возможен вывод параметров отдельно по типам:

- Async — вывод параметров асинхронного порта;
- Bbs — вывод параметров PMS;
- Character — вывод специальных кодов управления;
- Health — вывод счетчиков состояния и статуса;
- Id — вывод идентификационных параметров;
- Link — вывод параметров канала связи;
- Monitor — вывод параметров монитора;
- Printer — вывод параметров принтера;
- Timing — вывод временных параметров.

Если класс не указан, выводятся все параметры контроллера.

☐ DWait n

n = 0-250, число, определяющее интервал в 10 мс приращениях между окончанием сигнала в канале и началом передачи контроллером. Все станции региона должны устанавливать одинаковый интервал для повышения приоритета ретранслируемых пакетов.

☐ Echo ON/OFF

- ON — символы, принятые контроллером с компьютера, будут посылаются обратно с контроллера на компьютер;
- OFF — символы, принятые контроллером с компьютера, не будут посылаться обратно с контроллера на компьютер.

Если вы не видите на экране символов, которые набираете на клавиатуре, введите команду `Echo ON`.

Если символы, которые вы набираете на клавиатуре, дублируются, введите команду `Echo OFF`.

☐ `EDithdr n call#1 @ call#2`

- `n` — номер редактируемого сообщения;
- `call#1` — редактируемый позывной корреспондента;
- `call#2` — редактируемый позывной почтового ящика корреспондента.

Команда позволяет редактировать заголовки и производить переадресацию сообщений, находящихся в PMS.

☐ `EScape ON/OFF`

- `ON` — `<Escape>` (1bH) будет передаваться на компьютер как код (24H);
- `OFF` — `<Escape>` (1bH) будет передаваться на компьютер как код 1bH.

☐ `Firmnr ON/OFF`

Команда используется при работе контроллера в составе узла.

☐ `Flow ON/OFF`

При `Flow ON` вводимые с терминала или компьютера символы будут останавливать вывод на терминал, пока не:

- начнет передаваться вводимый пакет (разговорный режим);
- заполнится строка (командный режим);
- превысится длина пакета;
- переполнится выходной буфер терминала.

☐ `FOrward n`

`n` — номер сообщения, маркируемого для форвардинга.

Команда маркирует сообщение, находящееся в PMS, для форвардинга в местный BBS.

☐ `FNpms call#1 [Via call#2]`

- `call#1` — позывной узла, с которым нужно установить соединение;
- `call#2` — позывной цифрового ретранслятора.

Команда инициирует ручной форвардинг PMS с узлом с возможностью использования цифрового ретранслятора.

☐ `FPms call#1 [Via call#2]`

- `call#1` — позывной BBS или PMS, с которым нужно установить соединение;
- `call#2` — позывной цифрового ретранслятора.

Команда инициирует ручной форвардинг PMS с BBS или PMS с возможностью использования цифрового ретранслятора.

❑ FRack *n*

n — число, определяющее время ожидания в секциях подтверждения на посланный пакет перед следующей повторной попыткой.

В случае, если канал включает ретрансляторы, интервал определяется как $n \times (2 \times m + 1)$, где *m* — количество ретрансляторов.

❑ FULLdup ON/OFF

- ON — дуплексный режим разрешен;
- OFF — дуплексный режим запрещен.

❑ HEADERln ON/OFF

- ON — заголовок и текст отображаемых кадров выводятся отдельно на двух строках;
- OFF — заголовок и текст отображаются на одной строке.

❑ HEAlled ON/OFF

- ON — включение режима индикации работы процессора контроллера. Светодиоды STA и CON начинают мигать;
- OFF — режим нормального функционирования.

❑ (H)elp или ?

Команда вызова подсказки для пользователей, соединенных с PMS по радиоканалу.

❑ HId ON/OFF

- ON — контроллер каждые 9,5 минут будет посылать идентификационный пакет, если станция работает в режиме ретрансляции;
- OFF — идентификационные пакеты посылаться не будут.

❑ HOMEbbs call

Команда устанавливает позывной BBS, с которого в PMS будет осуществляться пересылка почты.

❑ Id

Немедленно и однократно исполняемая команда посылке идентификационного пакета при работе станции в режиме ретрансляции, аналогична команде HID.

❑ (J)log

Команда для пользователей, соединенных с PMS по радиоканалу и показывающая список позывных станций, услышанных в канале. Знак * рядом с позывным указывает на то, что станция работала через ретранслятор.

☐ (K)ill n

Команда для пользователей, соединенных с PMS по радиоканалу и удаляющая сообщение в PMS под номером n.

☐ (KM)

Команда для пользователей, соединенных с PMS по радиоканалу и удаляющая все сообщения для данного пользователя.

☐ Kill n

Команда удаляет сообщение номер n в PMS.

☐ KILONFWD ON/OFF

- ON — сообщения, помеченные для форвардинга, после форвардинга будут удалены;
- OFF — сообщения, помеченные для форвардинга, после форвардинга будут сохранены.

☐ KISS ON/OFF

- ON — по команде RESTART иницируется режим KISS;
- OFF — по команде RESTART иницируется нормальный пакетный режим.

Режим KISS используется при поддержке протокола TCP/IP. Выход из режима KISS возможен по команде PARAM AX0 255.

☐ LCAlls call#1 [,call#2,...,call#8]

Команда определяет список до восьми позывных, пакеты с которыми в зависимости от команды BUdlist будут отображаться командой MONITOR.

☐ LCok ON/OFF

- ON — контроллер будет посылать на компьютер символы нижнего регистра;
- OFF — символы нижнего регистра будут конвертироваться в символы верхнего регистра.

☐ LCStream ON/OFF

- ON — символ (A—J), следующий после символа, определяемого командой STReamsw, может быть как в верхнем, так и в нижнем регистре;
- OFF — символ (A—J), следующий после символа, определяемого командой STReamsw, должен быть только в верхнем регистре.

☐ LFadd ON/OFF

- ON — символ <LF> добавляется после каждого <CR> при передаче пакета;
- OFF — символ <LF> не добавляется.

☐ LFIgnore ON/OFF

- ON — символы <LF> игнорируются;
- OFF — символы <LF> воспроизводятся нормально.

☐ (L)ist

Команда пользователя, соединенного с PMS по радио, выводящая список сообщений, хранящихся в PMS.

☐ List

Команда выводит список сообщений, хранящихся в PMS.

☐ LOGonmsg ON/OFF

- ON — при соединении с PMS корреспонденту посылается подтверждающий текст;
- OFF — при соединении с PMS подтверждающий текст не посылается.

☐ MAll ON/OFF

- ON — в режиме разъединения на экран выводятся пакеты всех станций в канале;
- OFF — в режиме разъединения на экран выводятся только пакеты от станций в соответствии с командами BUDlist и LCAlls.

☐ MAXframe n

n = 1-7, максимальное количество кадров в посылаемом пакете.

☐ MCOM ON/OFF

- ON — выводятся на экран все кадры;
- OFF — выводятся на экран только информационные кадры.

☐ MCon ON/OFF

- ON — в режиме соединения на экран выводятся пакеты от станций в канале;
- OFF — в режиме соединения на экран пакеты не выводятся, даже если MONITOR ON.

☐ MFilter n1 [,n2,...,n4]

n = 0-7FH — ASCII-код символа. Команда определяет ASCII-коды до четырех символов, которые будут отфильтрованы. Для того чтобы очистить MF-лист, наберите MF 00 (нельзя 0 или 00H!!!).

☐ MHClear

Очистка списка станций, услышанных в канале.

☐ Mheard

Вывод списка станций, услышанных в канале.

☐ (M)ine

Команда пользователя, соединенного с PMS по радио, выводящая список сообщений *для* и *от* него.

☐ Mine

Команда выводит список сообщений *для* и *от* вашего PMS-позывного.

☐ Monitor ON/OFF

- ON — если контроллер работает не в прозрачном (TRANSPARENT) режиме, то на экране отображается информация от всех услышанных станций. Команды MAl, BUdlist, LCAlls определяют, какие пакеты будут выводиться на экран;
- OFF — вывод не адресованной вам информации запрещен. В прозрачном режиме вывод не адресованной вам информации запрещен всегда.

☐ MRpt ON/OFF

- ON — позывной ретранслятора будет отображаться на экране;
- OFF — позывной ретранслятора отображаться не будет.

☐ MSGHdr ON/OFF

- ON — сообщения, помеченные для форвардинга, будут иметь заголовок: Originated From: <позывной> mm/dd/yy hh:mm Msg #n. Формат даты зависит от команды DAYUsa;
- OFF — сообщения, помеченные для форвардинга, не будут иметь заголовка.

☐ MStamp ON/OFF

- ON — при приеме пакета будут проставляться дата и время;
- OFF — при приеме пакета не будут проставляться дата и время.

☐ MYAlias call

Команда определяет позывной, используемый, в отличие от MYcall, только для ретрансляции.

☐ MYcall call

Команда определяет позывной контроллера.

☐ MYPcall call

Команда определяет позывной PMS. Этот позывной должен быть отличен от MYcall.

☐ NEwmode ON/OFF

- ON — при подаче команды на соединение контроллер, не дожидаясь соединения, переходит в разговорный или прозрачный режим, в зависимости от команды CONMode. При разъединении контроллер возвращается в командный режим;

- OFF — контроллер до установления соединения находится в командном режиме, а после установления соединения переходит, в зависимости от команды CONMode, в разговорный или прозрачный режим. При разъединении контроллер остается в том же режиме.

□ NODetext текст

Команда определяет порядок соединения через узел для форвардинга сообщений при использовании команды FNPms. Текст должен содержать команду соединения (Connect), позывной и, при необходимости, позывной ретранслятора.

□ NOmode ON/OFF

- ON — переключение контроллера в командный, разговорный и прозрачный режимы будет происходить только при подаче соответствующей команды (^C, CONV, Trans);
- OFF — переключение контроллера будет происходить автоматически в соответствии с командой NEwmode.

□ NUcr ON/OFF

- ON — после символа <CR> на компьютер будут посылаться символы <NULL> (ASCII-код 00), требующиеся в некоторых типах компьютеров для осуществления временной задержки. Применение команды целесообразно в случае пропадания символов после <CR>. Количество посылаемых символов определяется командой NULLs;
- OFF — символы <NULL> после символа <CR> посылаться не будут.

□ NULf ON/OFF

- ON — после символа <LF> на компьютер будут посылаться символы <NULL> (ASCII-код 00), необходимые для осуществления временной задержки в некоторых типах компьютеров и терминалов. Применение команды целесообразно в случае пропадания символов в начале строк;
- OFF — символы <NULL> после символа <LF> посылаться не будут.

□ NULLs n

n = 0–30, количество нулей, вставляемых после <CR> или <LF>.

□ Paclen n

n = 0–255, количество байтов данных в пакете. Контроллер автоматически начнет передавать данные, когда количество введенных байтов превысит число n. Это значение используется в разговорном и прозрачном режимах. Значение 0 эквивалентно 256.

□ PACTime Every/After n

- Every — передача пакета происходит каждые $n \times 100$ мс;

- After — передача пакета происходит по истечении $n \times 100$ мс со времени ввода с компьютера последнего символа;
- $n = 0-250$, число выражено в интервалах по 100 мс. Этот параметр всегда используется в прозрачном режиме и может употребляться также в разговорном режиме, если CPactime ON. Если $n = 0$, то пакеты передаются без пауз.

☐ PARity n

$n = 0-3$; выбор способа проверки на четность. 0 — none; 1 — odd; 2 — none; 3 — even. Команда определяет параметр связи между компьютером и контроллером.

☐ PASSs n

$n = 0-7fH$, число, определяющее ASCII-код символа. Символ PASS позволяет передавать в пакете символы, вызывающие выполнение команд контроллера.

☐ PASSAll ON/OFF

- ON — контроллер будет воспринимать искаженные пакеты;
- OFF — контроллер не будет воспринимать искаженные пакеты.

☐ PErsist n

$n = 0-255$. Команда устанавливает порог, с которым сравнивается случайное число, вырабатываемое в случае, когда пакет должен быть передан заново. Если случайное число больше PErsist, пакет не передается. Если оно меньше или равно PErsist, то пакет передается. Значение PErsist=127 задает 50%-ную вероятность передачи неподтвержденного пакета. PErsist=255 задает 100%-ную вероятность захвата канала.

☐ PIdcheck ON/OFF

- ON — все пакеты с PID, отличным от f0H, будут игнорироваться;
- OFF — все пакеты будут восприниматься.

☐ PMS ON/OFF

- ON — PMS доступен для удаленных пользователей через эфир;
- OFF — PMS недоступен для удаленных пользователей через эфир.

☐ PPersist ON/OFF

- ON — выбор метода PERSISTANCE повторной передачи пакета;
- OFF — выбор обычного метода FRACK повторной передачи пакета.

Команда, совместно с PErsist и SLOttime, позволяет осуществить метод PERSISTANCE (настойчивый) повторной передачи пакета. Однако эффективность этого метода возможна только в случае его применения всеми станциями в канале.

□ (R)ead *n*

Команда пользователя, соединенного с PMS по эфиру, позволяющая прочитать сообщение *n* в PMS.

□ Read *n*

- *n* — номер сообщения в PMS;
- *n-n* — номера первого и последнего сообщений, которые нужно прочитать в PMS.

□ REConnect *call#1* [VIA *call#2* [... , *call#9*]

- *call#1* — позывной корреспондента, с которым нужно соединиться;
- *call#2* — позывной ретранслятора, через который будет происходить соединение.

Команда немедленно изменяет маршрут соединения.

□ REDisplay *n*

n = 0-7fH, ASCII-код символа. Команда определяет код символа, иницилирующего повторный вывод на экран компьютера исправленной строки после ее редактирования.

□ REMSysop ON/OFF

- ON — корреспондент с позывным MYPCall может корректировать содержимое PMS по эфиру;
- OFF — корреспондент с позывным MYPCall не может корректировать содержимое PMS по эфиру.

□ RESET

Немедленно исполняемая команда по реинициализации контроллера. Введенные параметры, список услышанных станций и сообщения в PMS будут стерты. Чтобы реинициализировать контроллер без потери содержимого, следует использовать команду RESTART или выключение и включение контроллера.

□ RESptime *n*

n = 0-250, интервалы времени по 100 мс. Команда устанавливает минимальную задержку подтверждения пакетов. Эта задержка может применяться совместно с ожиданием DWAIT и другими задержками. Используется для увеличения пропускной способности канала во время передачи файлов, когда посылается максимальное количество пакетов полной длины. Возможна ситуация, при которой у передающего контроллера не окажется пакета, готового к передаче, и принимающий контроллер начнет подтверждение переданных ранее пакетов, что затруднит передачу оставшихся пакетов. Установка RESPTIME 10 принимающего контроллера устранит подобную ситуацию.

❑ RESTART

Команда немедленного действия. Производит переинициализацию пакетного контроллера, используя параметры, записанные в энергонезависимом ОЗУ. Действие команды RESTART аналогично выключению и последующему включению контроллера. См. также RESET.

❑ RETry n

$n = 0-15$, количество повторных попыток передачи неподтвержденных пакетов. Неподтвержденные передаются n раз, после чего операция прекращается. Значение $n = 0$ задает неограниченное число попыток. В случае превышения n контроллер переходит в разъединенное состояние с выдачей соответствующего информационного сообщения, если не был задан прозрачный режим. См. также FRack и Ppersist.

❑ Rxblock ON/OFF

- ON задает вывод данных на терминал в формате RXBLOCK, предназначенном для прикладного программного обеспечения систем, аналогичных BBS;
- для корректной работы такого формата требуется установка AWLEN 8, AUTOLF, MFILTER и подобных в OFF;
- формат RXBLOCK данных от других станций имеет следующую структуру:
 - ◇ ffH — байт начала;
 - ◇ L0 — старший байт поля длины после логического OR с f0H;
 - ◇ L0 — младший байт поля длины;
 - ◇ PID — байт идентификатора протокола;
 - ◇ DATA — поле данных.

❑ Sale

Команда непосредственного действия, выводящая на экран информацию дилера. В случае отсутствия такой информации в EPROM выводится сообщение: not implemented.

❑ Screenln n

$n = 0-255$, ширина строки терминала в символах. Команда используется для форматирования данных, посылаемых на терминал. В командном и разговорном режимах после передачи n символов добавляется последовательность из символов возврата каретки и перевода строки.

Если компьютер сам форматирует данные, следует установить SCREENLN 0.

❑ (S)end

Команда удаленного пользователя PMS. См. SEND ниже.

□ `SEND to_call-n @ call-n $_BID`

- `to_call` — позывной адресата;
- `call` — позывной BBS;
- `n` — идентификатор подстанции;
- `$_BID` — идентификатор бюллетеня.

Параметры `n` и `$_BID` необязательны.

Команда `SEND` начинает процедуру подготовки сообщения, адресованного `TO_CALL`. Возможна адресация всем (`ALL`). Если `TO_CALL` не указан, используется значение `MYPCALL`.

Позывной BBS указывается после введения символа `@` через пробел. Идентификатор бюллетеня `BID` начинается со знака доллара `$` и составляется по местным правилам.

Команда `SEND` запрашивает заголовок и текст сообщения.

`Subject:` — заголовок максимальной длины в 28 байт, завершающийся символом возврата каретки.

`Text:` — строки текста завершаются символами возврата каретки. Конец сообщения задается вводом в начале строки символа точки и возврата каретки либо символов `/EX`, либо `<Ctrl>+<Z>`.

Длина сообщения ограничена только объемом свободной памяти `PMS`. В случае нехватки памяти выводится `Buffer overflow`.

После того как сообщение успешно записывается, `PMS` информирует: `Message saved as Msg # n`.

□ `SEndpac n`

`n = 0-7fH`, число, определяющее ASCII-код символа. Команда определяет символ, при вводе которого в разговорном режиме контроллер будет принудительно формировать пакет и ставить его в очередь на передачу. Символ можно определять как в шестнадцатеричной, так и в десятичной форме. Для разговорного режима задавайте `SENDPAC 0dH` и `CR ON`. Пакеты будут передаваться естественным образом, завершаясь символом возврата каретки. Если параметр `SPACETIME` установлен в `ON`, задайте в качестве `SENDPAC` редко используемый символ, например `<Ctrl>+<A>`, и `CR OFF`. Это позволит посылать пакеты по времени без включения в них лишних символов возврата каретки.

□ `SLottime n`

`n = 0-255`, время в 10 мс интервалах. Команда определяет промежуток времени между реализациями случайных чисел в процедуре `PERSIST`. Большее значение `SLOTTIME` уменьшает вероятность захвата контроллером радиоканала.

□ Softdcd ON/OFF

- ON — сигнал DCD вырабатывается программным обеспечением;
- OFF — сигнал DCD вырабатывается оборудованием модема.

□ SP

Расширение команды SEND PMS. Команда SP устанавливает признак сообщения в P, т. е. личное, или частное.

□ SR n

n — номер сообщения, на которое пишется ответ, указывается обязательно. Команда SR автоматически берет позывные TO и FROM из сообщения n и меняет их местами, оставляя поле SUBJECT тем же самым. Поле @BBS не заполняется, для этого можно использовать команду EDITHDR.

□ ST

Расширение команды SEND PMS. Команда SP устанавливает признак сообщения в T, т. е. для отправки в систему NTS, что накладывает ограничения на его формат.

□ STArt n

n = 0-7FH, число, определяющее ASCII-код символа. Можно указывать как в шестнадцатеричной, так и в десятичной форме. Команда определяет старт-символ терминала для возобновления передачи данных от контроллера к терминалу. Стоп-символ определяется командой STOP. Если оба этих символа установлены нулевыми, то программное управление передачей прекращается и контроллер будет реагировать только на сигнал CTS. В случае совпадения этих символов контроллер будет попеременно возобновлять и останавливать передачу при приеме такого байта.

□ STATus

Команда непосредственного действия, показывающая в шестнадцатеричной форме состояние соединения в текущем канале:

- 00 — разъединено;
- 01 — соединяется;
- 03 — разъединяется;
- 04 — соединено и все пакеты не подтверждены;
- 06 — соединено и есть неподтвержденные пакеты.

□ STExt text

text — строка длиной до 80 символов, оканчивающаяся возвратом каретки. Команда STExt определяет необязательный текст, посылаемый при подключении пользователя к PMS. Если текст не определен, то он не посылается. Данная команда не связана с LOGONMSG.

☐ STOp n

n = 0-7fH, число, определяющее ASCII-код символа. Можно указывать как в шестнадцатеричной, так и в десятичной форме. Команда устанавливает стоп-символ для приостановки терминалом передачи данных от контроллера к терминалу. Передача возобновляется старт-символом, определенным командой START.

Если оба этих символа установлены нулевыми, то программное управление передачей прекращается и контроллер будет реагировать только на сигнал CTS. В случае совпадения этих символов контроллер будет попеременно возобновлять и останавливать передачу при приеме такого байта.

☐ STREAMCa ON/Off

При задании ON после идентификатора канала выводится позывной станции, с которой установлено соединение. См. также MRPT.

☐ STREAMDb1 ON/Off

При установке ON удваиваются принимаемые символы STREAMSW (обычно "|") с тем, чтобы они лучше отличались от таких же символов, вводимых в контроллер для переключения каналов.

☐ STReamsw n

n = 0-ffH, число, определяющее ASCII-код символа. Команда определяет символ, используемый пользователем и контроллером для переключения каналов. Этот символ можно передавать в разговорном режиме с помощью команды PASS. В прозрачном режиме данный символ не переключает каналы, поэтому следует переходить в командный режим.

☐ TRACe ON/Off

Установка ON включает режим трассировки, при котором все принимаемые пакеты полностью воспроизводятся на экране в шестнадцатеричном виде.

☐ Transparent

Команда, непосредственно переводящая контроллер из командного в прозрачный режим. Состояние соединения канала не изменяется. Прозрачный режим главным образом используется для межмашинного диалога, поэтому многие параметры форматирования и управления передачей в нем не действуют.

☐ TRFlow ON/Off

- при установке ON управление передачей данных в прозрачном режиме производится символами START и STOP;
- при установке OFF используется аппаратное управление передачей (CTS/RTS), и все символы рассматриваются контроллером как дан-

ные. Если символы START и STOP установлены нулевыми, то компьютер должен всегда использовать аппаратное управление;

- если START и STOP ненулевые и TRFLOW ON, используется программное управление. Контроллер будет реагировать на эти символы оставаясь прозрачным для других данных, поступающих с терминала.

□ TRIes n

n = 0–15, количество повторных попыток передачи на текущем канале. Данная команда используется для получения информации и принудительной установки количества попыток. Если команда вводится без аргумента и есть неподтвержденный пакет, то возвращается текущее количество сделанных попыток. В случае, когда все пакеты подтверждены, возвращается количество попыток, предпринятое для получения подтверждения предыдущего пакета. TRIES всегда 0, когда RETRY 0. Эта команда полезна при получении статистики по каналам. Будучи введенной с аргументом, команда принудительно устанавливает счетчик попыток. Такое использование не рекомендуется.

□ TXDelay n

n = 0–120, время в интервалах по 10 мс. Команда устанавливает время, которое должно пройти с момента включения передатчика до момента передачи данных. Нужное значение подбирается опытным путем.

□ TXFlow ON/OFF

- если TXFLOW ON, то для определения типа управления передачей данных в прозрачном режиме используется параметр XFLOW;
- TXFLOW OFF устанавливает аппаратное управление, и все данные, передаваемые на терминал, становятся прозрачными;
- если TXFLOW ON и XFLOW ON, контроллер использует символы XON и XOFF для управления вводом с терминала. Если при этом TRFLOW OFF, то терминалу доступно только аппаратное управление выводом контроллера;
- если символы XON и XOFF установлены нулевыми, аппаратное управление включается независимо от параметра TXFLOW.

□ TXUifram ON/OFF

- команда TXUIFRAM OFF предотвращает передачу непосланных пакетов в виде UI-кадров в случае разрыва соединения;
- TXUIFRAM ON отменяет передачу кадров UI с клавиатуры в разговорном режиме, но разрешает маяки.

□ Unproto call#1 [Via call#2[,call#3...,call#9]]

- call#1 — позывной адресата;
- call#2–9 — позывные ретрансляторов.

Данная команда определяет позывные адресата и ретрансляторов (необязательно) для пакетов, посылаемых вне протокола, т. е. когда нет соединения. Пакеты посылаются в виде I-кадров. По умолчанию в качестве позывного адресата используется CQ.

Такие пакеты, посылаемые другими контроллерами, могут отображаться на экране установкой MONITOR ON, BUDLIST и LCALLS соответственно. Список ретрансляторов применяется и к пакетам маяков (посылаемых с адресом BEACON).

□ USers n

n = 1-10, количество разрешенных для соединения каналов. Данная команда управляет порядком использования каналов при очередных соединениях.

- USER 0 — соединение допускается в любом свободном канале;
- USER 1 — соединение допускается только в канале А;
- USER 2 — соединение допускается в каналах А и В и т. д.

□ (V)ersion

Команда удаленного пользователя PMS. См. команду Version ниже.

□ Version

По данной команде на экран выводится наименование программного обеспечения контроллера, записанного в ПЗУ, включая перечень возможностей, дату выпуска и контрольную сумму.

□ Xflow ON/OFF

- по команде Xflow ON контроллер считает, что компьютер или терминал реагирует на символы XON и XOFF, т. е. используется программное управление передачей данных;
- если Xflow OFF, то будет производиться аппаратное управление сигналом RTS.

□ XMitok ON/OFF

- ON — включение передатчика разрешено;
- OFF — включение передатчика запрещено.

Если XMitok OFF, то передатчик не будет включаться сигналом РТТ, хотя контроллер может пакетировать, принимать и посылать данные. Данную команду можно использовать для сбора статистики пакетной радиосвязи в отсутствие оператора с гарантией, что контроллер не будет посылать данные в эфир, а также при тестировании контроллеров.

□ XOff n

n = 0-7fH, число, определяющее ASCII-код символа. Можно указывать как в шестнадцатеричной, так и в десятичной форме. Устанавливается

символ `Xoff`, посылаемый контроллером на компьютер с тем, чтобы последний приостановил передачу данных. Обычно `Xoff` устанавливается равным `<Ctrl>+<S>`, но для того, чтобы контроллер в разговорном режиме сообщал о переполнении своих буферов звуковым сигналом, следует установить значение `<Ctrl>+<G>` (07H).

□ `XON n`

`n = 0-7fH`, число, определяющее ASCII-код символа. Можно указывать как в шестнадцатеричной, так и в десятичной форме. Устанавливается символ `XON`, посылаемый контроллером на компьютер с тем, чтобы последний возобновил передачу данных. Обычно `XON` устанавливается равным `<Ctrl>+<Q>`, но для того, чтобы контроллер в разговорном режиме сообщал об освобождении своих буферов звуковым сигналом, следует установить значение `<Ctrl>+<G>` (07H).

Следует помнить, что основная масса модемов в конструкциях TNC предназначена для работы на скоростях 1200 Бод и выше. Для тех, кто имеет возможность работать только на скорости 300 Бод, следует вместо установленного в корпусе TNC модема на микросхеме TCM3105 подключить внешний модем — MODEM22 (или 21). Уверен, что результаты будут превосходные.

Звуковая карта вместо модема

Во время обмена мнениями с другими радиолюбителями по поводу роли компьютерных звуковых карт в любительских цифровых видах связи, многие упрекают меня в том, что я умаляю роль этих устройств.

Но это совершенно не так. Мне очень даже нравятся эти устройства, и я частенько с удовольствием слушаю музыку, которая выдается компьютером посредством звуковых карт.

Цифровая связь основана на том, что сигналы от компьютера перед поступлением в передатчик должны обязательно пройти через устройство, называемое модулятором, а от приемника к компьютеру обязательно должны пройти через устройство, называемое демодулятором. Два этих устройства, объединенные в одну конструкцию, называются модемом. Хочешь или не хочешь, но без модема нет и не может быть цифровой связи. В радиолюбительской практике часто ВМЕСТО специального аппаратного модема используют ИМИТАЦИЮ модема посредством звуковой карты. То есть звуковая карта не является обязательным атрибутом цифровой связи, а является лишь СУРРОГАТНЫМ ЗАМЕНИТЕЛЕМ МОДЕМА. Это объективная реальность и от нее никуда не денешься.

При этом я не имею в виду те виды связи и программы, которые изначально предназначены для работы только со звуковой картой компьютера, такие

как PSK31, MT63 и др. Так что, по моему мнению, следует различать два способа применения звуковых карт:

- применение звуковых карт совместно с программами для цифровых видов связи, которые изначально были рассчитаны на применение аппаратных модемов;
- применение компьютерных звуковых карт совместно с программами, для работы с которыми они изначально предназначены.

В предыдущей главе этой книги я постарался довольно подробно описать программу PSK31, работа которой возможна только при наличии в компьютере цифровой карты. В этой главе я приведу основные характеристики другой популярной программы, работа которой основана только на применении звуковой карты. Эта программа называется RITTY.

Если сравнивать работу двух программ, из которых одна работает с аппаратным модемом, а другая использует звуковую карту, то сказать однозначно, что какая-то из этих программ является лучшей, просто невозможно. Многое зависит от субъективных факторов, т. е. от того, какой параметр считается данным радиолубителем самым важным. Например, для одного самым важным является безошибочный прием данных, для другого самым важным является удобство в работе, для третьего самым важным является красиво оформленная экранная заставка.

Коротко рассмотрим некоторые особенности применения дешевых звуковых карт, не имеющих в своем составе специального процессора.

В качестве модулятора звуковая карта работает довольно хорошо, т. е. она способна создавать хороший синусоидальный сигнал достаточной силы, и все было бы хорошо, если бы не некоторые особенности компьютерных операционных систем. Каждая операционная система работает с большим числом прерываний, т. е. если подошел момент для выполнения какой-то системной операции, то процессор прекращает выполнение текущего процесса и выполняет необходимую операцию. Если текущей операцией была генерация звука картой, то во время прерывания вместо звука от карты НИКАКОГО другого сигнала не будет. Наше ухо воспримет этот момент как небольшой щелчок, а вот программа вашего корреспондента воспримет это как помеху и выдаст сбой. Если MS-DOS делает не так уж много прерываний, то Windows с ее ну уж очень большим числом прерываний число сбоев увеличивает во много раз. Конечно, по сравнению с "мини-hamcom" или "мини-baycom" модемами на одном операционном усилителе это покажется выходом в рай, но по сравнению с нормальным модемом этот процесс будет выглядеть не очень здорово.

В качестве демодулятора звуковая карта выглядит еще хуже. К тем сбоям, которые создают прерывания, добавляется тот факт, что звуковая карта плохо выполняет роль детектора. Для лучшего выполнения роли детектора следует до звуковой карты ставить специальные преобразователи сигнала, а это

сводит на нет все остальные кажущиеся преимущества работы через эту карту в качестве модема. Еще одной функцией демодулятора является фильтрация поступающего от приемника сигнала. Совместно со звуковой картой должен использоваться программный фильтр. Это очень громоздкая и малоэффективная часть программы. Аппаратные фильтры работают намного лучше и не искажают сигнал. Этого мнения придерживаюсь не только я. Например, Peter Martinez, создатель AMTOR и PSK31, в своей большой статье о философии PSK31 пишет, что выполненные программно полосовые фильтры зачастую вносят искажения сигналов. Tom Sailer в документации к программе TERMAN93 предостерегает от использования узкополосных DSP-фильтров из-за вносимых ими в сигналы искажений.

При ответе на вопрос о том, почему использование звуковых карт в качестве модемов цифровой связи стало популярным, следует учесть следующие факторы. Радиолюбители, возраст которых сейчас перевалил за 60 или около того, образно говоря "родились с паяльниками в руках". Они знали, что если сами не сделают эту очень нужную на данный момент конструкцию, то попросту никогда ее не будут иметь. А радиолюбители младшего возраста родились "с папиным кошельком в руках" и самостоятельное изготовление какой-то конструкции превращается для них в проблему. Пусть не для всех, но для многих это так и есть. Поэтому изготовление даже самой простой конструкции повергает этого человека в ужас.

Когда радиолюбителям нашей страны было разрешено работать RTTY и когда появились мини-компьютеры "Радио-86РК", в журнале "Радио" был опубликован материал, сделанный москвичом Демиденко, о конструкции хорошего фильтрового модема. В это же время подобные конструкции, но на несколько ином уровне, были опубликованы в журнале "Funkamateur". Эти конструкции были исключительно удачными и использовались радиолюбителями много лет, у некоторых даже они есть и до сего времени.

Потом началась эра IBM PC и появились программы HAMCOM и BAYCOM, которые допускали возможность работать с очень простыми мини-модемами на одном операционном усилителе. Опытные радиолюбители сразу же отказались от этих "чудо-устройств" из-за очень плохого качества приема сигналов, а вот юным, которые еще не работали с нормальными модемами, они показались привлекательными, конечно же, из-за простоты реализации. Многие из них просто не имели представления о том, что можно принимать сигналы с гораздо лучшим качеством и без ошибок, которые с избытком допускались "мини-модемами". О том, что на самом деле эта конструкция представляла собой пародию на нормальный модем, никто не хотел слышать. Потом вдруг появилась возможность использовать вместо "мини-модема" звуковую карту, которая давала возможность вести прием с несколько лучшим качеством, и это для многих стало чудесным избавлением от мук с "мини-модемом", звуковым картам стали приписывать сверхъестественные возможности, короче появилось новое божество.

Особенно усердствуют в приписывании различных чудес при работе через звуковые карты те, кто не знал ничего кроме "мини-модема". В настоящее время любое упоминание о нормальном модеме у большинства ассоциируется с "мини-модемом" или с баснословными по цене модемами западных фирм. И совсем мало кто задумывается о том, что сделать модем, посредством которого можно на равных работать с любыми корреспондентами, совсем не сложно и совсем не дорого. Нужно только захотеть. Иметь хороший модем — нет ничего проще!

Программа RITTY и звуковая карта

Автором и разработчиком программы является радиолюбитель из США Brian Beezley (K6STI). Разработана во второй половине 90-х годов прошлого столетия.

Программа RITTY существует в нескольких вариантах. Некоторые из них работают только в режиме телетайпа (RTTY), другие кроме режима телетайпа имеют и режим работы PACTOR. Программа рассчитана на работу в среде MS-DOS, но может работать и под управлением Windows 95/98.

В режиме телетайпа программа может работать с большинством имеющихся звуковых карт, а вот в режиме PACTOR работа возможна только при наличии в компьютере звуковой карты фирмы Creative Labs 16 бит, имеющей два отдельных канала — канал приема и канал передачи.

Далее приведены основные характеристики программы для работы в режиме телетайпа (RTTY). Практически все эти характеристики действительны и для режима PACTOR.

RITTY сочетает эффективный модем, в качестве которого используется звуковая карта компьютера, с терминальной программой и позволяет передавать и принимать радиотелетайпные сигналы с использованием вашего компьютера. Программа использует улучшенный алгоритм цифровой обработки сигналов и не требует наличия аппаратного полосового фильтра. RITTY использует звуковую карту для аналогового ввода/вывода и имеет оптимизированный на ассемблере код для увеличения скорости работы. RITTY требует 486 или лучший процессор, математический сопроцессор, VGA, звуковую карту и DOS версии 3.3 или более поздней.

Чтобы установить RITTY под Windows 95, создайте папку \RITTY, запустите Проводник и откройте эту папку (\RITTY). При помощи правой кнопки мыши тяните значок **ritty.exe** на рабочий стол. Выберите команду **Создать ярлык**. В свойствах ярлыка измените значок на **ritty.ico**.

При работе RITTY под Windows убедитесь, что программное обеспечение при фоновой работе не потребляет слишком много времени центрального процессора и не прерывает выбор времени сигнала RITTY.

Звуковая карта

RITTY работает с большинством звуковых карт, но 16-разрядная Creative Labs работает лучше всего. Другие карты могут обеспечивать неадекватные уровни входа/выхода, предложить только грубую регулировку усиления и ограничить динамический диапазон.

Следует использовать медленную АРУ (автоматическая регулировка усиления приемника) и максимальное усиление по высокой частоте приемника. Соедините НЧ-выход приемника с линейным входом звуковой карты (или правый, или левый канал). Если ваш приемник не может полностью раскачать линейный вход, используйте вместо линейного микрофонный вход звуковой карты.

Микрофонный вход звуковой карты с высоким усилением очень легко перегрузить. Уменьшение усиления входа не будет помогать, потому что предварительный усилитель находится перед каскадом регулировки уровня. Следовательно, важно проверять искажения всякий раз, когда вы используете микрофонный вход. Настройте ваш приемник на сильный, чисто синусоидальный сигнал. Ищите гармоники, сгенерированные перевозбужденным предусилителем на спектральном экране. Вы должны видеть только один пик сигнала, поскольку вы настраиваетесь через полосу пропускания SSB. Если уровень вашего приемника перегружает микрофонный вход, добавьте дополнительный резистор (попробуйте 3,3 кОм) чтобы сформировать делитель напряжения со входным сопротивлением 540 Ом микрофонного усилителя.

Вы можете использовать микрофонный, линейный левый и линейный правый входы, чтобы разделять источники сигнала. Каждый вход сохраняет собственную установку уровня входа, так что вы можете переключаться между ними вместо того, чтобы регулировать уровень входов. (Некоторые звуковые карты не могут обеспечивать такой выбор.)

Тщательно контролируйте ваш передаваемый сигнал на наличие фона и шумов. Если сигнал грязный, добавьте 10 кОм резистор последовательно с линейным выходом и 330 Ом параллельно звуковому входу вашего трансивера. Затем повторно настройте уровень звукового сигнала для передачи.

Если вы установили выходной уровень в соответствии с этой процедурой, то теперь можно использовать регулировку уровня усиления микрофона, чтобы незначительно изменять выходную мощность.

RITTY генерирует полноразмерный, фазонепрерывный, синусоидальный сигнал. Звуковой сигнал более чист и более точен, чем FSK, встроенный во многие трансиверы.

FSK-, PTT-, и TTY-выходы

Некоторые трансиверы не позволяют использовать CW-фильтр в режиме LSB, но допускают это в режиме FSK. Для того чтобы использовать узкополосные ПЧ-фильтры, RITTY может генерировать FSK-сигнал на последовательном порте (контакт 2 для DB25, контакт 3 для DB9). Выберите порт в меню **Setup** (Установки). Выходное напряжение при передаче 1 (MARK) имеет отрицательное, 0 (SPACE) — положительное значение. Возможно, вы можете подать сигнал с COM-порта RS-232 непосредственно на вход FSK вашего трансивера. Если нет, то используйте интерфейсную схему, аналогично описанной для сигнала PTT. Корпус — контакт 7 для DB25 и 5 для DB9.

RITTY подает и на RTS-, и на DTR-линию (контакт 4 и 20 для DB25, контакт 7 и 4 для DB9) выбранного последовательного COM-порта сигнал PTT. Передача — положительное напряжение, прием — отрицательное. Вы можете получить сигнал PTT и с параллельного порта. В качестве адреса порта X введите 378 для LPT1, 278 для LPT2 или 3BC для LPT3. Соедините через 1 кОм резистор выходной контакт RS232 или TTL с базой n-p-n-транзистора, эмиттер на корпус, коллектор соедините с вашей линией PTT.

RITTY обеспечивает выходной телетайпный (таблица кодирования Baudot) сигнал на последовательном COM-порте при работе на прием для получения твердой копии на телетайпном аппарате. Сигнал RITTY в большинстве случаев содержит 72 знака в строке, заполняет поток данных так, чтобы обеспечить достаточное время для механического возврата каретки.

Рабочий экран

RITTY использует графический режим 640×480 точек. Это позволяет выводить 30 линий текста с определенными пользователем шрифтами и специальными символами, а также отображать форму сигнала в реальном режиме времени.

График в левом нижнем углу экрана — спектральный индикатор настройки. Желтый след — быстрое преобразование звукового сигнала. Синий след — средний спектр. Фиолетовые линии показывают положение фильтра канала; высота линии пропорциональна сигналу на выходе фильтра. Настройте ваш приемник так, чтобы сосредоточить пики сигнала на фиолетовых линиях. Красные стрелки указывают направо или налево всякий раз, когда самый сильный сигнал в звуковой полосе пропускания 3 кГц находится вне диапазона индикатора настройки. Красные полосы ниже базовой линии показывают звуковые частоты передачи. Используйте меню **Setup** (Установки), чтобы установить центральную частоту FSK, клавиши курсора <↑>/<↓> — чтобы изменить сдвиг частоты (также клавиши <+>/<->), клавиши курсора <влево>/<вправо> для точной настройки сдвига частоты.

Ниже графиков RITTY отображает декодированный текст, используя специальный шрифт с пятью пикселями по ширине. Каждый символ точно выровнен по пяти выборкам сигнала, из которых он декодирован. Это позволит вам сопоставить ошибки декодирования с искажениями сигнала. Нажатие любой клавиши "заморозит" вывод для более пристального осмотра.

Использование RITTY

Для запуска программы наберите слово "RITTY" в командной строке DOS. Нажатием на клавишу <Esc> на экран выводится меню со списком команд и функций. Выход из меню происходит также при нажатии клавиши <Esc>. Это удобно для быстрого контроля параметров программы. Вы можете переходить непосредственно от одного меню к другому, нажимая клавишу <Esc>.

Чтобы изменить пункт меню, используйте клавиши курсора <вверх> и <вниз> для перемещения красного указателя. Если пункт предлагает выбор, используйте клавиши <влево> и <вправо>, чтобы переместить подсветку на другой параметр. Если пункт числовой, наберите новое значение. После ввода всех цифр нажмите клавишу <Esc>, чтобы внести изменения и убрать меню. Когда вы в следующий раз вызовете меню, курсор будет там, где вы его оставили. Завершение числового ввода клавишей <Esc> удобно для неоднократного изменения отдельного параметра, чтобы сразу наблюдать эффект. Нажмите клавишу <Enter> вместо клавиши <Esc>, чтобы сохранить меню на экране и передвигать курсор к следующему пункту. Изменение параметров посредством клавиши <Enter> удобно для изменения нескольких параметров сразу.

Отдельные пункты меню имеют "горячие" клавиши (они подчеркнуты). Вы можете изменять один из этих параметров без обращения к меню, одновременно нажимая клавишу <Alt> и "горячую" клавишу.

Нажмите клавишу <Pause>, чтобы включить режим передачи. RITTY начнет передавать все, что вы введете с клавиатуры. Нажмите снова клавишу <Pause> или <Esc>, чтобы возвратиться в режим приема. Так как клавиша <Pause> неудобно расположена на некоторых портативных компьютерах (ноутбуках), RITTY воспринимает также нажатие сочетания клавиш <Ctrl>+<Enter>.

Вы можете передавать фиксированные сообщения, нажимая клавиши <F1> по <F12>. Подготовьте эти сообщения заранее в режиме приема. (Вы можете редактировать эти сообщения для последующей передачи, нажимая клавиши <Alt>+<F1>—<Alt>+<F12>.) Каждое сообщение может содержать до 27 строк. Для редактирования можно использовать клавиши <Backspace> и <Delete>. Нажатием на клавиши <Alt>+<W> удаляется слово, <Alt>+<D> удаляется строка и <Alt>+<C> очищает полное сообщение. Нажмите на клавиши <Ctrl>+<вправо> или <Ctrl>+<влево>, чтобы перейти к следующему или предыдущему слову. Нажмите <Ctrl>+<R>, чтобы ввести символ RX. Ко-

гда RITTY сталкивается с символом RX во время передачи, то немедленно переходит на прием. Вы можете также использовать <Ctrl>+<R> в режиме передачи при опережающем наборе с клавиатуры.

Одно сообщение может вызывать другое при помощи символа %. Например:

```
F1 %9 DE RA3XB
F2 HELLO %10
F9 R1CT
F10 NICK
```

Нажатие на клавишу <F1> выдает на передачу сообщение "R1CT de RA3XB", <F2> — "HELLO NICK" (ПРИВЕТ НИК). Символ % заставляет программу выдать на передачу другое сообщение, номер которого стоит в строке непосредственно после этого символа. Символ ^ используется для обновления состояния регистра. Эта функция полезна в тестовых сообщениях, которые содержат длинные строки без изменения регистра.

Используйте любой из буферов сообщений, чтобы начать отвечать корреспонденту, в то время, когда его станция еще ведет передачу. Если он переключится на прием в то время, когда вы все еще набираете текст для передачи на клавиатуре, не спешите выходить из редактора сообщений. Просто нажмите клавишу <Pause>. Сообщение будет скопировано в буфер передачи и начнется передача. Вы можете закончить ввод сообщения с клавиатуры, пока передается начало вашей фразы.

Чтобы передать ASCII-файл, введите имя файла (и полный путь к нему, если необходимо) в одном из буферов сообщений. RITTY передаст содержание файла вместо его имени. RITTY ничего не передает и отображает на экране в виде точки (·) символы ASCII, не имеющие эквивалентов в таблице телетайпных символов. Символ % не имеет никакого специального значения в файлах, но ^ имеет.

Нажмите клавишу <Esc>, чтобы очистить буфер передачи. Это удобно, если вы нажали эту функциональную клавишу по ошибке. Нажмите клавиши <Alt>+<M> или <Alt>+<S> для передачи непрерывного сигнала, соответствующего 1 или 0. Это полезно для измерения частоты передачи или для проверки неравномерности амплитудной характеристики вашего фильтра ПЧ при работе на передачу.

Единственная клавиша для редактирования в режиме передачи — <Backspace> (возврат на один символ). Вы можете стереть любой символ в текущей строке, пока он еще не передан (не изменил цвет с голубого на белый).

RITTY может передавать символы (цифры/знаки) немедленно или может помещать их в буфер и передавать полное слово только после того, как вы нажмете клавиши <пробел>, <Enter> или <Ctrl>+<R>. Режим передачи "по словам" дает возможность исправить опечатку, пока вы не ушли в наборе

далеко вперед. Режим "Word output" переключается в меню **Transmit** (Передача).

При работе в обычном режиме RITTY автоматически записывает принятый и переданный текст в файл с именем text в текущем каталоге. После осуществления выхода из RITTY вы можете его переименовать и редактировать для сохранения необходимой информации. При каждом запуске RITTY переписывает этот файл, старая информация теряется. Размер файла ограничен 64 Кбайт (приблизительно 3 часа приема на скорости в 45 Бод).

RITTY сохраняет свои параметры (только в лицензированной версии). Когда вы выходите из программы, RITTY сохраняет все настройки меню и сообщения в файле ritty.ini в каталоге, содержащем файл ritty.exe. Если файл ritty.ini существует, когда вы запускаете программу, RITTY читает этот файл и повторно инициализирует себя. Или, если такого файла нет, использует установки по умолчанию. Удалите файл ritty.ini, чтобы возвратиться к значениям по умолчанию.

Чтобы запустить RITTY с различным набором параметров, вы можете определять различные файлы инициализации. Поместите имя файла (через пробел) в командной строке. Чтобы создать файл инициализации, запустите RITTY с новым именем файла, отрегулируйте все параметры и затем выйдите из программы. Например:

C:\RITTY\rtty.exe proba.ini

Обработка сигнала

При приеме, декодировании и обработке RTTY-сигналов в программе работают выполненные программно следующие устройства.

□ Узкополосный детектор, который сохраняет линейность сигнала максимально долго. В детекторе нет ограничителя амплитуды входа. Это позволяет разбирать сигналы, пропадающие в шумах и QRM. Узкополосный детектор использует пару согласованных фильтров, для 1 (MARK) и 0 (SPACE). Полоса пропускания фильтров и центральная частота регулируются в зависимости от скорости приема. Сигнал с пары фильтров подается на детектор огибающей.

- Согласованные фильтры имеют узкие пики, но широкий пологий спад. Сильные сигналы, попадающие в полосу фильтра, могут прерывать прием. Чтобы уменьшить QRM (помехи от других станций), вы можете включить острый полосовой фильтр последовательно с каждым согласованным фильтром. Сложные фильтры имеют крутые склоны и более 100 дБ окончательного подавления внеполосных сигналов.
- BPF-фильтры включают компенсацию межсимвольной интерференции. Если ваш компьютер не очень быстрый для нормальной работы BPF, используйте 500 Гц телеграфный (CW) фильтр радиоприемника

или внешний аппаратный фильтр. Даже без дополнительного фильтрования узкополосный детектор хорошо вырезает умеренные QRM, потому что не использует ограничитель входа.

- RITTY обеспечивает сложный алгоритм АТС (Автоматическая Коррекция Порога). АТС изменяет порог, который отличает 1 (MARK) от 0 (SPACE). Для тонов равной амплитуды порог нулевой. АТС изменяет порог, чтобы компенсировать разбаланс амплитуд, возникающий из-за затуханий сигнала, неравномерности амплитудно-частотных характеристик (АЧХ) фильтра промежуточной частоты (ПЧ) при приеме и передаче, завалов АЧХ в звуковом тракте. АТС помогает восстановить текст, который иначе был бы искажен. Это позволяет вырезать помехи от соседних станций в одном канале при помощи фильтра вашего приемника и сохранять прием с оставшимся каналом. Каналы сигналов MARK и SPACE имеют для программы одинаковое значение. Но во время замираний сигнала более слабый канал может вносить большее количество шума, чем сигнал другого канала.
- Широкополосный детектор играет важную роль при приеме сигналов. Хотя узкополосный детектор увеличивает чувствительность, он не очень хорошо выделяет сигналы, разрушенные сильными ионосферными помехами. Например, полярный флаттер (дрожание сигнала) может модулировать сигнал по амплитуде и фазе. Оба компонента могут создавать незначительные боковые полосы, которые расширяют спектр сигнала вне полос пропускания согласуемых фильтров. Фазомодулированная компонента также может ухудшать когерентность фазы сигнала.
 - Широкополосный детектор RITTY обеспечивает лучший прием искаженных сигналов. Детектор (счетчик экстремумов/прохождения сигнала через ноль, со скользящим окном длиной в один бит) имеет очень короткое время когерентности. Фаза сигнала может сильно изменяться без значительного воздействия на выходной сигнал детектора.
 - Широкополосный детектор неявно содержит ограничитель и следовательно извлекает выгоду из предварительной фильтрации. RITTY обычно включает двугорбый фильтр наряду с широкополосным детектором. Этот фильтр автоматически настраивается на входной сигнал и подходит для сигналов с разносом тонов меньше чем 250 Гц. Медленные компьютеры не включают фильтр и используют вместо него телеграфный (CW) фильтр вашего радиоприемника. Эти фильтры улучшают чувствительность и понижает восприимчивость к QRM. Широкополосный детектор не требует АТС, автоподстройка (Autotune) эффективно обеспечивает эту функцию.
 - Когда включен широкополосный детектор, RITTY рисует пунктирную линию в спектральном дисплее на частоте, соответствующей событию, которое определено как MARK или SPACE. Высота линии отражает

амплитуду сигнала на входе детектора. Отключите автоподстройку и используйте контур рисунка сигнала, чтобы настроиться на двухгорбую реакцию фильтра (APU приемника должна быть включена) или поместите центральную частоту RITTY на середину полосы пропускания CW-фильтра.

- ❑ Автоподстройка (Autotune) обеспечивает автоматическую настройку двугорбого фильтра и фильтра канала. Когда автоподстройка включена, фильтры быстро схватывают самые сильные сигналы внутри полосы автоподстройки и затем медленно отслеживают их. Автоподстройка воспринимает смещения между 135 и 250 Гц при использовании узкого смещения. Нажмите клавишу <Пробел>, чтобы "заморозить" частоты фильтра (маркеры фильтра кажутся серыми). Нажмите клавишу <Пробел> снова, чтобы "разморозить". Фильтры могут захватывать помехи, но вы можете циклически переключать три последних значения настроек фильтра, нажимая клавишу <Tab>. <Tab> также замораживает автоподстройку. Используйте меню **Receive Options** (Опции приема), чтобы управлять скоростью автоподстройки и полосой пропускания.
 - Вам может потребоваться заморозить автоподстройку при работе, чтобы предотвратить перепрыгивание фильтров среди сигналов различных тонов. Фильтры с узкополосным детектором способны к изоляции одного сигнала среди многих, особенно с включенным двугорбым фильтром (BPF).
 - При вызове станций включите автоподстройку тонального генератора (AFSK) в меню **Transmit Options** (Опции передачи). RITTY затем автоматически выровняет центральную частоту тона ваших сигналов, чтобы она совпала с принятой. Смещение при передаче будет 170, 182, 200, 425, или 850 Гц в зависимости от смещения принятого сигнала.
 - Автоподстройка изменяет порог шумоподавителя широкополосного детектора. Эффективность этого обеспечивает АТС. Кроме того, автоподстройка облегчает работу шумоподавителя широкополосного детектора.
- ❑ Интеграция повторений — RITTY может обнаружить повторение (копию) текста даже когда он неправильно декодирован. Интеграция повторений (RI) выравнивает и интегрирует демодулированные формы сигнала, лежащие в основе повторенного текста так, чтобы сигналы добавились когерентно ($S + S = 2S$) но шум добавляется некогерентно ($N + N = 1,414N$). Это улучшает отношение сигнал/шум S/N фактором n для текста, повторенного n раз. RITTY показывает декодированный интегрированный текст коричневым цветом. Используйте RI, чтобы выделять позывные и другой повторенный текст из шума.
- ❑ Шумоподаватель. В RITTY есть чувствительный шумоподаватель, чтобы запрещать вывод "мусора" на экран при отсутствии принимаемого полез-

ного сигнала. Вы можете регулировать порог шумоподавления или отключать его полностью используя меню **Receive Options** (Опции приема). Когда шумоподаватель подавляет один или большее количество символов, RITTY показывает красную точку. Когда шумоподаватель открывается, новый символ записывается поверх точки. Если красные точки на мгновение появляются в течение передачи, это означает, что вы пропустили текст. Попробуйте уменьшить порог шумоподавателя. Абсолютный уровень входного сигнала не влияет на чувствительность шумоподавателя.

Меню **SETUP**

- ☐ **Source** (Источник). Выберите микрофонный вход звуковой карты только в том случае, если ваш приемник действительно не может выдавать сигнал, достаточный для линейного входа. Убедитесь, что звуковая карта не вносит искажений при использовании микрофонного входа. Некоторые звуковые карты не могут обеспечивать выбор источника входа.
- ☐ **Text Font** (Шрифт текста). Этот параметр определяет, какой экранный шрифт использовать для вывода на экран при приеме, передаче и редактировании текста.
- ☐ **Zero Style** (Стиль нуля). RITTY обеспечивает для вывода на экран цифры ноль три варианта — обычный, с точкой и перечеркнутый. Выберите тип, который вы предпочитаете. Заметьте, что обычный ноль и буква О имеют различные формы.
- ☐ **Audible Bell** (Слышимый звонок). Выберите **Yes** (Да), чтобы разрешить RITTY испускать короткий звуковой сигнал, когда принят символ "колокольчик". Выберите **No** (Нет), чтобы показать вместо этого небольшой симпатичный символ.
- ☐ **Input Level** (Уровень входов). Усиление регулируется с шагом 2 дБ для 16-разрядных звуковых карт от Creative Labs. RITTY округляет в меньшую сторону нечетные значения. (Другие звуковые карты имеют более грубое разрешение и ограниченный диапазон регулирования усиления. Некоторые старые карты не могут обеспечивать никакое регулирование усиления входа вообще.) Каждый источник сохраняет собственную установку входного уровня.
- ☐ **Output Level** (Уровень выхода). Усиление регулируется с шагом 2 дБ для 16-разрядных звуковых карт от Creative Labs. RITTY округляет в меньшую сторону нечетные значения (другие звуковые карты имеют более грубое разрешение и ограниченный диапазон регулирования усиления).
- ☐ **Output** (Выходной сигнал). Когда вы выбираете FSK, вы должны выбрать последовательный порт для выходных данных. Выбранный порт также обеспечивает сигнал PTT и для FSK и для AFSK. Биты сигнала FSK не могут быть инвертированы программно и тоны FSK не могут меняться в зависимости от принятого сигнала.

- ☐ **COM Port** (COM-порт). Выберите последовательный порт для выходных сигналов PTT и FSK. Поставьте значение **Unused** (Не использовать) для AFSK/VOX или **X** для нестандартного порта.
- ☐ **Port X Address** (Адрес порта X). Определите нестандартный шестнадцатеричный адрес для COM-порта. Для получения сигнала PTT на параллельном порте (только для AFSK) укажите 378 для LPT1, 278 для LPT2 или 3BC для LPT3.
- ☐ **Center Freq** (Центральная частота). Центральная частота находится посередине между тонами MARK и SPACE. При использовании CW-фильтра отрегулируйте этот параметр так, чтобы сосредоточить тоны принимаемого сигнала в полосе пропускания фильтра. Вы можете использовать низкочастотные тоны (800—1000 Гц), которые более приятны для слуха.
- ☐ **Clock Offset** (Подстройка часов). RTTY нечувствителен к выбору времени, но этот параметр позволит вам ввести компенсацию для незначительных отклонений параметров от типовых для различных звуковых карт. Чтобы измерять смещение ваших часов (датчика меток времени), введите значение 0, частоту 1000 Гц для сигнала тонального генератора (AFSK) (**Center Freq 1000 HZ, Fixed Shift 0 Hz, AFSK Autotune Off**) и измерьте частоту при помощи частотомера с высоким разрешением. Измеренное значение, например **1000.145 Hz**, означает, что ваши часы идут на 145 ppm быстрее. Введите значение смещения **−145 ppm** для компенсации.
- ☐ **Baud** (Бод). Вы можете выбирать одну из следующих скоростей передачи (табл. 3.3).

Таблица 3.3. Скорость передачи.

Скорость, Бод	Длина элемента, мс
45	22
50	20
57	17,6
75	13,3
100	10

Меню *RECEIVE*

- ☐ **Flywheel** (Маховик). Выберите **Auto** (Авто) в большинстве случаев. Выключайте его (значение **Off**) для редкого сигнала с временными характеристиками, слишком нерегулярными для автоматического декодирования. Включите (значение **On**) для лучшего приема нормальных сигналов при глубоких замираниях или пропадании сигнала в шуме.

- ☐ **Detector** (Детектор). Выберите **Narrow** (Узкополосный) для обычного использования. Включайте **Wide** (Широкополосный) для сигналов, разрушенных полярным флаттером. Узкополосный детектор максимизирует чувствительность.
- ☐ **Narrow Det BPF** (BPF-фильтр узкополосного детектора). Выбрав **ON** (Включено), вы включите каскадно-острый полосовой фильтр с каждым согласованным фильтром узкополосного детектора. BPF-фильтр автоматически перестраивается вместе с согласованными фильтрами. BPF-фильтры не могут работать на медленных компьютерах. Оптимизированный BPF-фильтр используется на скорости 45 Бод. BPF, оптимизированный для 75 Бод, используется на скоростях 50, 57 и 75 Бод. Никакой BPF не используется на скорости 100 Бод.
- ☐ **Wide Det BPF** (BPF-фильтр широкополосного детектора). Двугорбый BPF-фильтр обычно включается вместе с широкополосным детектором. Если ваш компьютер медленный и не может запустить BPF, отключите его (выберите **Off**) и используйте CW-фильтр. BPF-фильтр широкополосного детектора отключен при использовании смещения 425 или 850 Гц.
- ☐ **ATC**. Для нормального приема включите его (выбрав **On**), чтобы задействовать Автоматическое Исправление Порога и Адаптивную Комбинацию Канала. Выберите **Off** (Выключить), чтобы наблюдать D1 и D2 формы сигнала, не искаженные АСС.
- ☐ **Autotune** (Автоподстройка). Выберите **S**, **M** или **F** для медленной, средней или быстрой скорости срабатывания автоподстройки. Автоподстройка использует среднее спектральное значение, так что этот параметр изменяет скорость затухания следа среднего спектра. Нажмите клавишу <Пробел>, чтобы "заморозить" частоты фильтра. Нажатие клавиши <Tab> позволит циклически перебрать последние три набора частот.
- ☐ **Autotune Width** (Ширина автоподстройки). Используйте этот параметр, чтобы сузить ширину окна автоподстройки для уменьшения влияния QRM на фиксацию фильтров.
- ☐ **Squelch Threshold** (Порог шумоподавителя). Порог шумоподавителя регулируется с шагом в 1 дБ. Значение **0** отключает шумоподаватель. Используйте эту установку для очень слабых сигналов с глубокими замираниями (QSB) или всякий раз, когда вы хотите убедиться, что ничего не пропустили при приеме.
- ☐ **Unshift on Space** (Отключить регистр после пробела). Переключает декодер в режим приема символов (LTRS) всякий раз, когда получен пробел. Это помогает предотвращать вывод длинных строк текста с неправильным регистром. Однако некоторые модемы не ожидают использования этого параметра и будут принимать неправильно. Отключите **Unshift on Space** (Отключить регистр после пробела) для этих сигналов.

- ☐ **Ignore Stray LF** (Игнорировать паразитный перевод строки). RITTY может игнорировать коды LF, которые не встречаются в первой колонке. Это помогает выводить большее количество текста на вашем экране при копировании зашумленных сигналов. LF редко (если когда-либо) передается, за исключением использования в первой колонке. (LF ставится для перевода строки.)
- ☐ **Decode Both Shifts** (Декодировать оба регистра). RITTY может одновременно выводить дополнительный поток символов, который использует другое значение регистра LTRS/FIGS. Эта возможность удобна, когда сигналы DX слабые и замирающие, и могут выводиться в неправильном регистре. Вы найдете правильный вариант на одной из двух показанных линий. Часто это даст вам важную, пропущенную часть неполного вызывного.
- ☐ **Show Control Codes** (Показывать управляющие коды). RITTY может выводить на экран символы для управляющих кодов CR, LF, LTRS, FIGS и BLANK. Это может показать интересные причуды и свойства различных модемов. Это также может показать избыточность, использованную некоторыми станциями, чтобы сразиться с шумом. (CR используется для возврата каретки. BLANK — неиспользуемый код.)
- ☐ **Display Case** (Вид текста). Вы можете показывать полученный текст заглавными или строчными буквами. Верхний регистр всегда используется, когда программа запущена резидентно (TSR).
- ☐ **Rep Integ** (Интегрировать повторы). Выберите **All** (Все), чтобы позволить интегрировать повторения для всего текста. Выберите **Digit** (Цифра), чтобы включать эту процедуру только для повторенного текста, который содержит по крайней мере одну цифру. Это подавляет большинство ложных повторений, которые часто обнаруживает режим **All** (Все).
- ☐ **Display RI** (Показывать RI). Выберите **Differ** (Различные), чтобы показать интеграцию повторений только тогда, когда индивидуальные повторения отличаются. Выберите **Always** (Всегда), чтобы показывать RI всякий раз, когда обнаружено повторение. Используйте **Differ** (Различные), чтобы уменьшить помехи на экране и **Always** (Всегда) — для оценки действия RI.
- ☐ **Waveform** (Форма сигнала). Выберите **Character** (Символ) для показа с высоким разрешением "mark-minus-space" сигнала длиной в один символ. Выберите **D1** или **D2**, чтобы наблюдать форму огибающей сигнала и порог декодирования.

Меню **TRANSMIT**

- ☐ **Word Output** (Вывод по словам). Выберите **Yes** (Да), чтобы передавать только целое слово. Эта опция выключена, когда программа запущена резидентно (TSR).

- ❑ **Duplicate LTRS FIGS** (Дублировать LTRS FIGS). RITTY может добавлять избыточность в передаваемый сигнал, посылая каждый LTRS и FIGS дважды. Это полезно, когда ваш сигнал слабый. Однако это замедляет передачу.
- ❑ **Refresh LTRS FIGS** (Обновлять LTRS FIGS). RITTY может автоматически снова посылать LTRS/FIGS, если ни один не был послан некоторое время. Введите число прошедших символов перед вторичной посылкой или 0, чтобы отключить эту опцию. Эта возможность обеспечивает альтернативу дублированию LTRS/FIGS и использованию ^.
- ❑ **Add Begin/End CR** (Добавить в начале/конце CR). RITTY может автоматически добавлять возврат каретки CR, когда вы переходите в режим передачи и еще один CR, когда вы переходите на прием. Это гарантирует, что ваш переданный текст не будет напечатан за шумовыми символами. (RITTY не прибавляет CR, когда вы прерываете передачу нажатием клавиш <Esc> или <Pause>, или что-нибудь только что передали).
- ❑ **AFSK Autotune** (Автоподстройка AFSK). Выберите **On** (Включить) для автоматической установки центральной частоты AFSK-сигнала в соответствии с принятым сигналом. При передаче сдвиг частот будет 170, 182, 200, 425 или 850 Гц, в соответствии с полученным смещением. Автоподстройка AFSK полезна при работе на поиск. Отключите ее при работе на CQ, чтобы ваш передаваемый сигнал не менялся. Автоподстройка AFSK работает только если включена автоподстройка приема.
- ❑ **Fixed Shift** (Фиксированный сдвиг частот). Этот параметр определяет сдвиг частот при передаче, когда автоподстройка AFSK отключена. Обычно нет необходимости изменять этот параметр, потому что он изменяется в соответствии со сдвигом частот на приеме при помощи клавиш стрелок <вверх> или <вниз> (но сдвиг частот при передаче не меняется при изменениях, сделанных при помощи клавиш курсора <влево> или <вправо>. Это позволяет вам делать точную настройку на принимаемый сигнал и при этом отвечать с правильным разносом частот). Если вы вручную установите сдвиг частот при передаче, не забудьте, что значение изменится всякий раз, когда вы измените сдвиг частот при приеме клавишами курсора <вверх> или <вниз>.

В то время как узкополосный сдвиг частот сегодня является стандартом для любительского RTTY, широкий сдвиг плюс АТС обеспечивают меньшую восприимчивость к затуханиям сигнала.

- ❑ **Stop Bits** (Стоповые биты). Вы можете выбирать 1 / 1,4 / 1,5 или 2 AFSK-стоповых бита. 1,4 в действительности — стоповые биты 31/22 (31 мс на скорости 45 Бод), стандарт для телетайпных машин на скоростях 45 и 57 Бод. Большинство любительских RTTY сигналов сегодня все еще использует эту длину. На других скоростях обычно используются 1,5 стоповые биты. Некоторые TNC передают 2 стоповых бита. Это замедляет пе-

редачу и расширяет окно приема, в течение которого шум может генерировать ложный стартовый импульс. Однако RITTY обеспечивает и эту длину для тестовых или других специальных целей. В заключение, вы можете выбрать 1 стоповый бит. Эта длина иногда используется на более высоких скоростях и может заметно ускорять передачу на скорости 45 Бод. Однако некоторые дешифраторы не будут надежно синхронизировать сигналы с такими короткими стоповыми битами.

- Всякий раз, когда вы изменяете скорость, RITTY устанавливает при передаче длину стопового бита 1,4 на 45 и 57 Бод и 1,5 на 50, 75 и 100 Бод. Если вы вручную регулируете число передаваемых стоповых битов, не забудьте, что этот параметр может изменяться, когда вы изменяете скорость в бодах.
- Последовательный порт UART определяет длину стопового бита FSK. Это всегда 1,5 (или очень близко к этому).

□ **New Line** (Новая строка). RITTY обеспечивает четыре выбора для формирования последовательности передаваемых символов, когда вы нажимаете клавишу <Enter>. CR — самая короткая возможная последовательность; используйте это в течение соревнований, когда вы уверены, что набор текста с клавиатуры достаточно редок. Значение по умолчанию — "CR LTRS" последовательность обновляет значение регистра в начале каждой строки (RITTY посылает FIGS вместо LTRS, когда первый символ новой строки требует этого). Оба варианта последовательностей прекрасно работают, когда декодируются терминальными программами, которые не допускают повторного вывода текста на экран по предыдущей строке. CR LF LTRS работает даже для примитивных программ, которые не замедляют вывод текста. В заключение, используйте CR CR LF LTRS LTRS при передаче на телетайпный механизм. Эта последовательность обеспечивает время для возврата каретки к первой колонке печати (однако это может заставить некоторые терминальные программы ошибочно выводить линии двойного интервала). RITTY ограничивает длину строки при передаче 72 символами, когда вы выбираете эту последовательность.

RITTY добавляет LF к последовательности CR, когда параметр **New Line** (Новая строка) имеет значение CR или CR LTRS, чтобы позволить правильно воспроизводить пустые строки дешифраторами, которые должным образом игнорируют последовательности из CR.

□ **Tones** (Тоны). Этот параметр настраивает инверсию для тонов сигнала на передаче и приеме (совместное изменение параметра предотвращает неумышленную обратную передачу). Любительский стандарт FSK — частота радиочастотного сигнала при передаче 1 (MARK) выше чем при передаче 0 (SPACE). Почти все коммерческие RTTY сигналы используют противоположное соглашение. Включить значение **Reverse** (Реверс) более удобно, чем сделать переключение на USB и перенастройку.

Цвета экрана

Нажмите клавишу <C>, чтобы изменить цвета экрана. (Чтобы настроить все, включите дисплей D1 или D2 и нажмите клавиши <Alt>+<O>, чтобы показать оба регистра. Наберите какой-нибудь текст для передачи и нажмите клавишу <Esc> прежде, чем текст будет передан.) Выберите пункт на экране клавишами курсора <влево> или <вправо> (пункт мигает, когда выбран). Затем используйте клавиши <Home>, <End>, клавиши со стрелками <вверх> или <вниз>, <PgUp> и <PgDn>, чтобы циклически перебрать 64 градации, доступные для каждого из красных, зеленых и синих цветных компонентов. Нажмите клавишу <Esc> для выхода из меню цветов.

Примечания

- ❑ RITTY игнорирует символ Бодо BLANK (ПРОБЕЛ) на приеме (за исключением режима показа управляющих кодов) и никогда не передает его.
- ❑ RITTY никогда не печатает повторно текст по предыдущей строке, когда получает CR без LF, но всегда передвигает вывод на следующую строку. Когда программа получает последовательность CR, она не выводит ошибочно строки из множества пробелов. RITTY показывает все принятые символы; когда принято более 80 в одной строке, символы выводятся в следующей строке.
- ❑ Всякий раз, когда вы вводите более 80 символов в строке при передаче, RITTY показывает их на следующей строке, но не передает никакие управляющие коды. Исключение: когда последовательность новой строки — CR CR LF LTRS LTRS, RITTY обрезает текст после 72 символов и затем начинает передачу новой строки.
- ❑ RITTY вводит три пробела, когда вы нажимаете клавишу <Tab>. Это не истинное табулирование, но обеспечивает удобный способ сделать отступ абзаца.
- ❑ Когда вы переключаетесь на передачу, RITTY сначала посылает LTRS для синхронизации и затем CR (если задано), и, наконец, посылает LTRS или FIGS перед первым символом текста. Когда вы переходите на прием, RITTY добавляет только CR (если задано).
- ❑ RITTY устанавливает частоту выборки звуковой карты приблизительно 6000 Гц. Это позволяет преобразовать максимальную частоту входного сигнала 3000 Гц. Звуковые карты от Creative Labs имеют встроенные фильтры. Имеется некоторый завал АЧХ входного сигнала на частотах выше 2500 Гц, но с АТС это не оказывает никакого эффекта на принимаемые сигналы, даже для смещения 850 Гц при центральной частоте 2210 Гц.

Использование совместно с RTTY от WF1B

RITTY может работать как модем для телетайпной контест-программы WF1B, которая специально предназначена для работы в соревнованиях по радиосвязи на коротких волнах в режиме RTTY.

Для этого запустите программу так:

```
RITTY contest.ini
```

При таком запуске создается новый файл инициализации. В этом файле будут сохранены параметры для работы в соревнованиях, они немного отличаются от обычных. Проверьте все настройки меню. Настройки, рекомендованные для использования в соревнованиях, приведены ниже.

❑ Выйдите из RITTY и запустите программу следующим образом:

```
RITTY I contest.ini
```

чтобы запустить программу резидентно с настройками для соревнований.

❑ Затем запустите программу WF1B. При настройке параметров теста выберите **K6STI TNC** в **Work sheet**.

❑ Далее в правом нижнем углу появится индикатор настройки — упрощенная версия дисплея быстрого преобразования Фурье RITTY. Пики сигнала находятся внутри полосок маркера для должным образом настроенного сигнала.

❑ После выхода из WF1B наберите RITTY U, чтобы выгрузить RITTY из памяти.

Вы можете автоматизировать эту процедуру при помощи небольшого командного файла.

И еще о RITTY

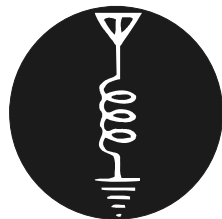
Программа RITTY может обеспечить очень хорошие результаты работы при тщательном выполнении всех заложенных автором требований. Одним из самых важных требований является использование двухканальной звуковой карты производства фирмы Creative Labs. Следует исключительно точно разобраться во всех тонкостях работы заложенных в программе детекторов и фильтров. Очень многие радиолюбители так и не смогли освоить все тонкости настройки этой программы.

Если эта программа вас очень заинтересовала, то следует поискать в Интернете полный русский перевод документации к этой программе, выполненный RU3RQ.

Если вас интересует использование этой программы в режиме PACTOR, то много интересного можно найти в Интернете на странице Кубанских радиолюбителей. Адрес этой страницы, а также русский перевод документации по использованию программы RITTY в режиме PACTOR можно найти в Интернете по адресу <http://users.kaluga.ru/ra3xb/>.

Мой личный опыт работы с этой программой показывает, что прием телеграфных сигналов программой CW_QSO совместно с MODEM21 дает гораздо меньше ошибок. Предлагаю вам самим в этом убедиться.

Глава 4



Компьютер и ИСЗ

Предисловие

Искусственные спутники Земли (ИСЗ) прочно вошли в жизнь человеческого сообщества. ИСЗ широко используются и в мирных и в военных целях. Каждый знает о спутниковом телевидении, спутниковой радиосвязи, об использовании спутников в разведке природных богатств Земли, об использовании спутников в метеорологии — это далеко не полный перечень использования ИСЗ на благо человека. Знаем мы также и о размещении на спутниках военного оборудования, о спутниках-шпионах. Но мало кто знает о том, что существуют спутники, которые используются в радиолюбительских целях.

О том, что представляют из себя радиолюбительские спутники, для каких целей они используются, как рассчитывается время появления спутника в зоне радиовидимости и какие спутники сейчас летают в космосе — вот об этом я хочу рассказать вам в этой главе.

Первые спутники

Первый в мире ИСЗ

Когда появилась идея о создании и запуске первого искусственного спутника Земли, все участники эксперимента надеялись на его успешный запуск, но никто не знал, сколько времени спутник просуществует: один день, неделю, месяц? Очень важно было принимать со спутника какие-то сигналы — только они могли доказать, что спутник находится на орбите. Было совершенно неизвестно, в каком пункте удастся принять сигналы, какой должна быть чувствительность приемной аппаратуры. Ясно было только одно: нужно иметь как можно больше пунктов наблюдения. И тогда решили привлечь к эксперименту радиолюбителей. Институту радиотехники и электроники АН СССР поручили держать связь с радиоклубами ДОСААФ и редакцией журнала "Радио". Были выделены необходимые средства на оснащение аппаратурой 28 радиоклубов, расположенных в различных городах страны. Аппаратуру доставляли туда самолетами.

Начальников клубов пригласили в Москву, здесь им читали лекции о том, как наблюдать за радиосигналами спутника, давали рекомендации относи-

тельно конструирования специальной аппаратуры. Предварительно аппаратуру испытывали на военном аэродроме. Привезли туда маленький передатчик на 20 МГц — копию того, который был установлен на первом ИСЗ, и летали с ним на самолете, а на земле вели прием его сигналов.

Судя по воспоминаниям участников эксперимента, спутник представлял собой небольшой контейнер, в котором размещался малогабаритный радиопередатчик на 20 МГц. Передатчик излучал несущую частоту, которая через небольшие промежутки времени прерывалась специальным таймером. В эфир шли короткие послышки, напоминающие непрерывную передачу телеграфных "тире". Из контейнера выходила определенной длины металлическая упругая лента, которая использовалась в качестве антенны. Питание осуществлялось от специальных электрических аккумуляторов. Срок работы передатчика зависел от времени работоспособности аккумуляторов.

Центральным радиоклубом СССР и другими радиолюбительскими организациями были организованы наблюдательные пункты, оснащенные очень приличной по тем временам аппаратурой в 28 населенных пунктах страны, в том числе и в Магадане, и на Камчатке, и на Сахалине.

Запуск спутника был осуществлен 4 октября 1957 года. Радиолюбители "схватили" спутник на самом первом его витке. Уже через четыре минуты после начала приема первых сигналов от спутника, диктор Всесоюзного радио Ю. Левитан объявил о запуске в Советском Союзе искусственного спутника Земли.

Это событие было очень важным и интересным в жизни каждого человека. Ночами в ясную погоду можно было видеть, как по небу медленно перемещается маленькая светящаяся звездочка. Мы знали, что это летел спутник. Впоследствии выяснилось, что этим светящимся в небе предметом был не спутник, а одна из огромных ступеней ракеты, которая тоже летала вокруг Земли. Спутник был настолько мал по размерам, что отражение от него нельзя было увидеть даже в мощный телескоп.

Так началась эра использования радиоаппаратуры на ИСЗ. Русское слово "спутник" с тех пор прочно вошло в обиход по всему миру.

До тех пор еще никогда на Землю не приходили радиосигналы связи из-за пределов ионосферы, с движущегося космического объекта! В субботу, 5 октября, и особенно в воскресенье, 6-го, на любительских диапазонах творилось что-то невообразимое: все оживленно обсуждали происшедшее, делились впечатлениями и выясняли какие-то детали, спрашивали, на каких частотах работает передатчик, когда его лучше слышно, каков характер сигналов. Зарубежные радиолюбители поздравляли советских радиолюбителей с успешным запуском.

Радиолюбители активно вели наблюдения за радиосигналами спутника. Помимо советских радиолюбителей в наблюдениях участвовали коротковолновики ГДР, Чехословакии, США, Англии, Голландии, Японии, Австралии и многих других стран.

Что же дали эти наблюдения? Прежде всего систематический прием сигналов с борта спутника позволял судить, правильно ли работает радиостанция на спутнике, установить продолжительность ее действия, оценить работу источников питания. Уже это немало, т. к. станция находилась в необычных условиях и подвергалась воздействию малоизученных факторов. Некоторые радиолюбители, кроме того, что делали привязку сигналов по времени, измеряли и их мощность. Такие наблюдения позволили определить величину напряженности электрического поля. По изменению тональности сигналов, так называемому эффекту Допплера, можно судить о параметрах орбиты спутника. Все эти эксперименты были исключительно важны для разработки и создания последующих конструкций аппаратуры для спутников.

Наблюдения радиолюбителей много дали и для изучения распространения радиоволн. Ведь до запуска спутника специалисты, изучающие прохождение радиоволн, зондировали ионосферу только снизу, а теперь это было сделано сверху. Наблюдения за характером прохождения сигналов позволили уточнить некоторые характеристики линий дальней коротковолновой связи.

Работаем через спутники РС

Начало радиолюбительской космической связи положил запуск в октябре 1978 года искусственных спутников Земли (ИСЗ) "Радио-1" и "Радио-2".

Инициатива их создания принадлежит журналу "Радио", при редакции которого работал координационный совет, объединивший усилия творческих коллективов радиолюбителей ДОСААФ, студенческих конструкторских бюро и инженерной общественности.

Главным вдохновителем, организатором и теоретиком первых радиолюбительских спутников в нашей стране был лауреат Государственной премии, полковник в отставке, бывший ранее разработчиком отечественной космической техники, радиолюбитель-коротковолновик (UA1AB) Владимир Леонидович Доброжанский. Только благодаря его знанию дела, связям в высоких кругах, исключительному упорству и трудолюбию удалось пройти все бюрократические препоны того времени и осуществить мечту многих радиолюбителей — запустить в космос первые два и последующие радиолюбительские спутники.

Владимир Леонидович начал с того, что написал и опубликовал в журнале "Радио" массу статей практически по всем вопросам создания, обнаружения и эксплуатации любительских ИСЗ. Прочитав эти статьи, многие радиолюбители стали писать в журнал "Радио" письма с просьбой подключить их к работе над новым проектом. Среди таких радиолюбителей был и автор этой книги. До сих пор хранятся в моем архиве письма, полученные от этого замечательного человека и радиолюбителя — Владимира Леонидовича Доброжанского. Присланный им планшет постоянно располагается над моей домашней радиостанцией. К сожалению, масса обстоятельств не позволила мне непосредственно участвовать в создании и разработке этого проекта, но

увлеченность идеей радиолюбительского использования ИСЗ осталось навсегда.

Идея В. Л. Доброжанского заключалась в разработке специального бортового комплекса радиоаппаратуры, который представлял бы собой космический ретранслятор с многостанционным доступом, запускаемый попутно (за счет резерва веса) с другим космическим объектом. Как бывший разработчик космической техники, он знал, что для точного вывода космического объекта на определенную орбиту и траекторию требуется космический аппарат строго определенного веса. Зачастую те аппараты, которые нужно было выводить в космос, имели заниженный вес, восполнять который приходилось балластом. В качестве такого "балласта" как раз и предлагалось использовать радиолюбительский бортовой комплекс.

Владимир Леонидович ежедневно, как на работу, приходил в редакцию журнала "Радио". Он нашел и привел энтузиастов из Московского энергетического института, которые впоследствии активно проявляли себя в создании и запуске ИСЗ, призвал к сотрудничеству студенческое КБ Московского авиационного института, поддерживал постоянные контакты с группами энтузиастов-радиолюбителей в городах Калуге и Молодечно (Белоруссия).

Однако ядром, вокруг которого, как на орбите, вращались идеи и разработки, мощным магнитом, который притягивал коллективы и энтузиастов, стали Координационный комитет при редакции журнала "Радио" и родившаяся вскоре Общественная лаборатория космической техники ДОСААФ, созданная на базе районного СТК Москвы.

Среди длинного списка сотрудников лаборатории, которые бескорыстно, безвозмездно, не считаясь ни с силами, ни со временем, отдавали свой творческий порыв созданию спутников, необходимо назвать, прежде всего, Владимира Борисовича Рыбкина. У него были "золотые руки и голова". Фактически через его рабочий стол прошел весь "бортовой комплекс" — аппаратура, которая была установлена на всех трех ИСЗ "Радио", построенных Общественной лабораторией ДОСААФ, СКБ МАИ "Искра" и студенческим КБ Московского энергетического института.

Затем пришло второе поколение любительских ИСЗ — эскадра из шести космических аппаратов, выведенная на орбиту одной ракетой-носителем 17 декабря 1981 года. Это были спутники: "Радио-3", "Радио-4", "Радио-5", "Радио-6", "Радио-7", "Радио-8".

В те дни я периодически, и довольно часто, включал радиоприемник на диапазоне 29 МГц, где наблюдал за работой аппаратуры на ИСЗ. Уже в те времена меня очень интересовали методы определения времени появления спутника в зоне радиовидимости и расчеты других элементов спутниковых орбит. Прослушивая эфир 18 декабря, вдруг услышал появляющиеся слабые сигналы спутника. В это время никаких спутников в зоне радиовидимости быть не должно. Стал слушать дальше. Когда сигналы стали разбираться,

оказалось, что работал спутник с неизвестным мне позывным. Не помню, по каким причинам я сдвинул рукоятку настройки, где наткнулся на работу еще одного спутника, потом следующего. Как потом выяснилось, накануне были запущены одной ракетой сразу шесть спутников, которые первые несколько дней летали плотной кучкой, затем спутники стали отставать друг от друга. В результате таких отставаний образовалась своеобразная цепочка — сначала слушаешь работу одного спутника в течение 12—15 минут, затем на такое же время появляется другой, затем — третий, четвертый, пятый, шестой, затем снова первый и т. д.

Круг общественных групп, занимающихся спутниками, к тому времени начал сужаться. "Отошли" студенческие КБ. Главную ношу — создание систем телеметрии, командной радиолинии, бортовой "логики", "доски объявлений", "роботов" — взяли на себя калужане, организовавшие под руководством А. П. Папкова свое общественное КБ при Калужском музее истории космонавтики имени К. Э. Циолковского.

23 июня 1987 года в Советском Союзе был осуществлен новый запуск искусственного спутника Земли "Космос-1861", на котором была установлена аппаратура, полностью разработанная группой Александра Павловича Папкова.

Затем были запущены спутники RS-10 и RS-11, RS-12 и RS-13. Потом были RS-15 и другие, менее известные. Кстати, RS-10/RS-11 совсем недавно прекратили свою работу, RS-12/RS-13 работают и сейчас. Работает и RS-15. Все эти спутники очень похожи друг на друга, потому что их работа осуществляется на одном и том же комплексе бортовой аппаратуры.

Несмотря на то, что в настоящее время в эфире находится множество спутников с различными самыми современными видами цифровой связи, спутники типа RS пользуются очень большой популярностью среди радиолюбителей всех стран мира. Дело в том, на мой взгляд, что эти спутники используют очень удобные для радиолюбителей диапазоны и виды связи. В качестве альтернативы приведу пример со спутником P3D (АО-40). Сначала были разрекламированы исключительные возможности этого спутника, так что радиолюбители никак не могли дождаться его запуска. Ждать пришлось несколько лет. Осенью прошлого года, наконец, он был запущен, сейчас летает, но практически никто не может через него работать. Дело в том, что принимать информацию с этого спутника можно только на диапазонах выше 2400 МГц. А аппаратуры на такой диапазон почти ни у кого нет.

Если заглянуть в Интернете по адресу <http://www.amsat.ru/>, то можно увидеть страничку отечественной организации AMSAT-RUS. Страничка заполнена материалами о прошлых славных делах и людях, а вот нового ничего нет. Похоже, что вся работа этой организации заключается в разработке и опубликовании в Интернете своей странички.

В последние годы о разработке новых российских спутников широкой общественности ничего не известно. Что-то делается при Центре подготовки космонавтов, об этом говорит тот факт, что на международной космической

станции работает аппаратура отечественного изготовления, что-то делается в группе А. П. Папкова при Калужском музее истории космонавтики. Говорят, что в самое ближайшее время на орбите появится RS-23. Но это пока только на уровне слухов. У радиолюбителей нашей страны огромный интеллектуальный потенциал, который все же должен вырваться на свободу и начать действовать. Ждем-с...

Радиолюбительские ИСЗ

На сегодняшний день на различных орбитах вокруг Земли вращаются 27 спутников с радиолюбительской аппаратурой. Некоторые из них работают очень хорошо уже долгое время, другие капризничают — то работают, то не работают, третьи работают в полсилы. Есть и такие спутники, которые исправно вращаются, а работать для радиолюбителей не хотят.

Большинство спутников с радиолюбительской аппаратурой летают вокруг Земли по круговым орбитам, имеют удаление от Земли от 500 до 1500 км. На рис. 4.1 изображена схема движения такого спутника. Один оборот спутника вокруг Земли продолжается около двух часов, время нахождения в зоне радиовидимости составляет от 5 до 18 минут. Зоной "радиовидимости" я называю пространство, в котором спутник достаточно хорошо слышен и нормально принимает передаваемые ему команды.

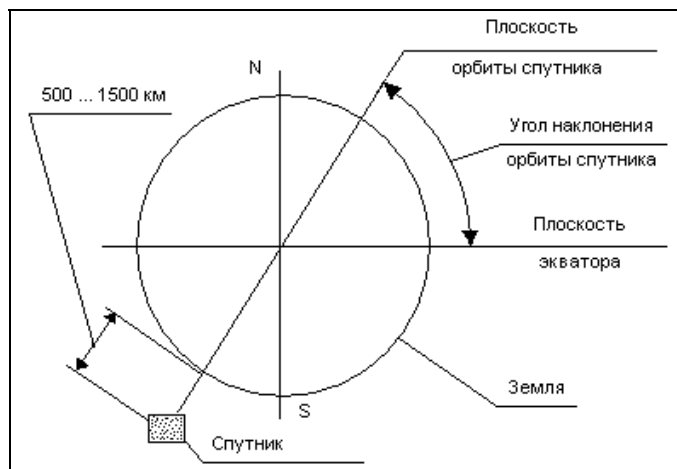


Рис. 4.1. Схема движения низкоорбитального спутника

Движение каждого спутника характеризуется периодом обращения, углом наклона плоскости орбиты спутника к плоскости экватора Земли, удалением орбиты от поверхности Земли.

Типичным представителем такого типа спутников является RS-12/13, который далее буду упоминать довольно часто. Преимущество подобных спутников в достаточно хорошем приеме поступающих от них сигналов, недостатком является слишком малая продолжительность пребывания спутника в зоне радиовидимости.

Имеются также спутники, движение которых осуществляется по эллиптической орбите. Схема движения по эллиптической орбите приведена на рис. 4.2. Обозначенная на схеме величина R_3 обозначает радиус Земли. Радиус R_3 примерно равен 3180 км.

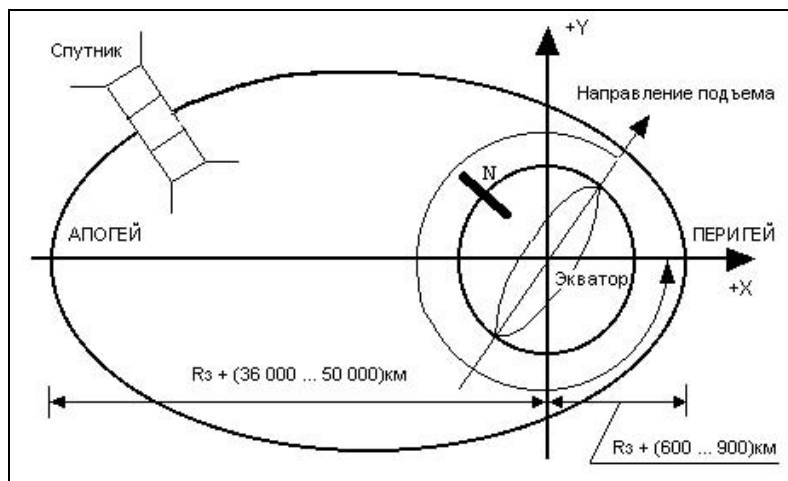


Рис. 4.2. Схема движения по эллиптической орбите

Преимуществом таких спутников является длительное время нахождения спутника в зоне радиовидимости, более простое управление антенными системами для их направления на спутник. Таких спутников сейчас два. Первым из них является ветеран с семнадцатилетним стажем — АО-10. Спутник работает в режиме ретранслятора уже очень давно и всегда хорошо. В удачные дни бывает так, что одновременно можно сработать со всеми континентами. Вторым из спутников, движущихся по эллиптическим орбитам, является АО-40. Практически с самого начала этот аппарат преследуют неудачи. В настоящее время прием информации со спутника можно вести только на диапазонах частот выше 2400 МГц. Аппаратура на такие частоты, мягко говоря, имеется не у каждого радиолюбителя, поэтому информации о работе этого спутника исключительно мало.

На рис. 4.3 приведена карта нашей планеты, выполненная в прямоугольных координатах. Траектории движения низкоорбитальных спутников выглядят на этой карте в виде одного периода синусоидальной волны. Чем больше

угол наклоения плоскости орбиты к плоскости экватора, тем "амплитуда" подобной синусоиды становится больше.

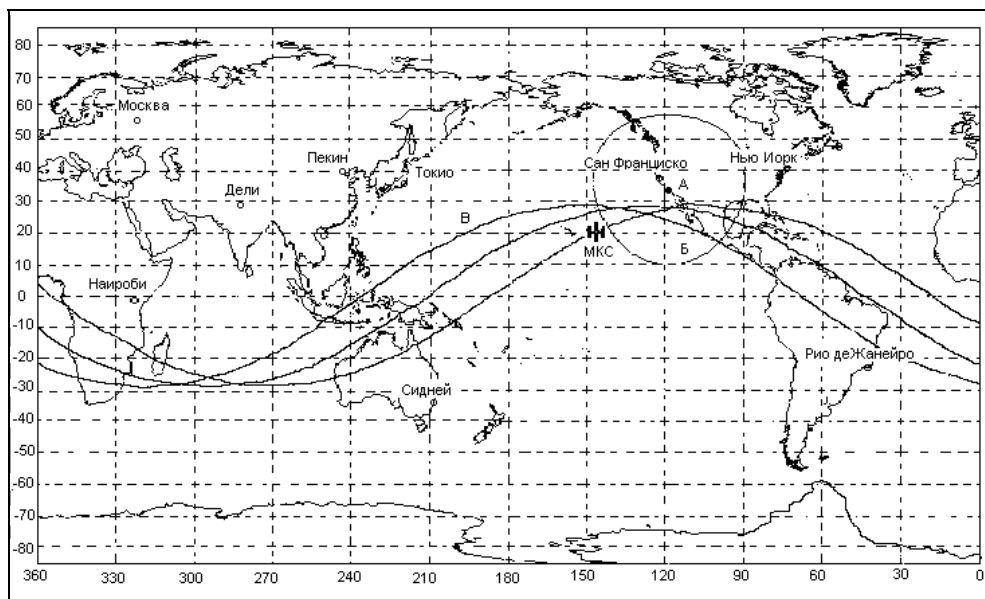


Рис. 4.3. Карта Земли с траекториями движения спутников

Точкой А на карте обозначен один из спутников. Кривая линия в виде замкнутого неправильной формы круга (кривая В) показывает границу зоны радиовидимости этого спутника. Кривая линия в виде синусоиды (линия В) показывает траекторию движения какого-то спутника, предположим, что этим спутником является МКС — Международная Космическая станция. Таких траекторий на карте показано три. Можно предположить, что это траектория трех последовательных витков одного и того же спутника.

Ниже привожу перечень всех вращающихся вокруг Земли спутников с их краткими характеристиками. Для тех читателей, которые хотели бы узнать о том или ином спутнике чуть больше, привожу адреса страниц в Интернете. Эти страницы, как правило, принадлежат радиолюбителям, готовым поделиться с друзьями своими знаниями и достижениями в спутниковой радиосвязи.

Далее приведена расшифровка некоторых специальных терминов, использованных в перечнях спутников.

□ Download — обозначает радиочастоту, на которой можно получать информацию со спутника (на которой спутник ведет передачу на Землю).

- Uplink — обозначает радиочастоту, на которой следует передавать информацию на спутник (на которой спутник принимает информацию с Земли).
- L1-band — обозначает работу на определенном частотном канале и при определенных условиях (смотреть на соответствующей странице в Интернете).
- Veacon — обозначает радиочастоту, на которой работает МАЯК — специальный радиопередатчик, передающий на Землю позывные спутника и так называемую "телеметрию" — зашифрованное сообщение о состоянии технических систем данного спутника.
- USB, LSB, SSB — работа однополосным сигналом (на верхней боковой, нижней боковой, на любой боковой полосе частот).
- CW — работа телеграфом.
- ANS — AMSAT NEWS — служба новостей организации AMSAT.

Для получения большего количества информации по всем вопросам радиолюбительских спутников обращайтесь в Интернете по адресу <http://www.amsat.org/>.

Существует в Интернете также и страничка, организованная отечественной организацией AMSAT-RUS. Эту страничку можно посмотреть по адресу <http://www.amsat.ru/>. Но эта страничка даст вам, в лучшем случае, какую-то информацию о прошлых делах, потому что новых дел у этой организации нет.

Спутники с аналоговой аппаратурой

- Phase 3D / AMSAT OSCAR 40 / AO-40
 - Uplink U-полоса 435,550–435,800 МГц CW/SSB.
 - L1-band 1 269,250–1 269,500 МГц CW/SSB.
 - L2-band 1 268,325–1 268,575 МГц CW/SSB.
 - Downlink 2 401,225–2 401,475 МГц CW/SSB.
 - Функционирует в наладочном режиме. Работает U/L-1 через S-2.
 - Выведен на промежуточную орбиту 16 ноября 2000 года ракетой Ariane 5 с космодрома во Французской Гвиане. Постепенно переведен на постоянную эллиптическую орбиту.
 - Познакомиться с графиком работы можно в Интернете по адресу <http://www.amsat-dl.org/journal/adlj-p3d.htm>.
 - Pieter Tjerk, PA3FWM, разработал новую программу под Linux для расшифровки телеметрии AO-40, которую можно взять в Интернете по адресу <http://www.cs.utwente.nl/~ptdeboer/ham/ao40/>.

□ МЕЖДУНАРОДНАЯ КОСМИЧЕСКАЯ СТАНЦИЯ ARISS

- Всемирный пакет Uplink — 145,990 МГц.
- Region 1 voice Uplink — 145,200 МГц.
- Region 2/3 voice Uplink — 144,490 МГц.
- Всемирный Downlink для voice и пакета — 145,800 МГц.
- Позывной TNC — RZ3DZR.
- Функционирует.
- Построенный и запущенный в России в сентябре 2000 года жилой модуль станции позволяет длительное нахождение экипажа на борту станции, является основой станции.
- График свободного времени команды можно найти в Интернете по адресу <http://spaceflight.nasa.gov/station/timelines/2001/may/index.html>.
- Позывные российских космонавтов: RS0ISS, RZ3DZR.
- Позывной американских астронавтов: NA1SS.

□ РАДИО СПОРТ RS-12

- Uplink от 145,910 до 145,950 МГц CW/SSB.
- Downlink от 29,410 до 29,450 МГц CW/SSB.
- Маяк 29,408 МГц.
- Включен снова в работу 1 января 2001 года. Работает нормально.
- Выведен на орбиту 5 февраля 1991 года с космодрома Байконур.
- О работе через спутник RS-12/13 есть материалы в Интернете по адресу <http://www.qsl.net/ac5dk/rs1213/rs1213.html>.

□ РАДИО СПОРТ RS-15

- Uplink от 145,858 до 145,898 МГц CW/SSB.
- Downlink от 29,354 до 29,394 МГц CW/SSB.
- Маяк 29,352 МГц (неустойчивая) частота.
- SSB средняя частота 29,380 МГц (неофициальная).
- Функционирует частично в режиме А, используется 2-метровый uplink и 10-метровый downlink.
- Выведен на орбиту 26 декабря 1994 года с космодрома Байконур.
- Dave, WB6LLO, имеет действующую информацию для RS-15 и RS-13 на его Web-странице. В дополнение к спутниковым данным там имеется информация по антенне для работы в mode-A. Страница WB6LLO в Интернете с этой информацией находится по адресу <http://home.san.rr.com/doguimont/uploads>.

□ OSCAR 10 AO-10

- Uplink от 435,030 до 435,180 МГц CW/LSB.
- Downlink от 145,975 до 145,825 МГц CW/USB.
- Маяк 145,810 МГц (немодулированная частота).
- Функционирует частично Режим — В.
- Выведен на орбиту 16 июня 1983 года ракетой Ariane с космодрома во Французской Гвиане. АО-10 был блокирован в 70-см uplink и 2-метровый downlink в течение нескольких лет.
- W4SM имеет большое количество информации относительно спутника в Интернете по адресу <http://www.cstone.net/~w4sm/AO-10.html>.

□ AMRAD AO-27

- Uplink 145,850 МГц FM.
- Downlink 436,795 МГц FM.
- Функционирует в режиме J.
- Выведен на орбиту 26 сентября 1993 года ракетой Ariane с космодрома во Французской Гвиане. АО-27 использует метод, называемый Установленным Регулированием Мощности Затемнения (TEPR), чтобы регулировать работу бортовых батарей. TEPR учитывает время, как долго спутник был в затемнении (или на солнце) и решает, какие подсистемы включить или отключить.
- АО-27 Web-страница на AMSAT-NA дает объяснения по работе со спутником АО-27: <http://www.amsat.org/amsat/sats/n7hpr/ao27.html>.

□ UO-14

- Uplink 145,975 МГц FM.
- Downlink 435,070 МГц FM.
- Функционирует в режиме J.
- UO-14 был выведен на орбиту в 22 января 1990 года ракетой Ariane с космодрома во Французской Гвиане.
- Tim, KG8OC, модифицировал информационную Web-страничку с информацией по UO-14, расположенную в Интернете по следующему адресу <http://www.qsl.net/kg8oc>.

□ JAS-1b FO-20

- Uplink от 145,900 до 146,000 МГц CW/LSB.
- Downlink от 435,800 до 435,900 МГц CW/USB.
- Функционирует. FO-20 находится в режиме JA непрерывно.
- JAS-1b (FO-20) был выведен на орбиту в феврале 1990 года и продолжает функционировать очень хорошо.

□ JAS-2 FO-29

- Voice/CW Режим JA.
Uplink от 145,900 до 146,000 МГц CW/LSB.
Downlink от 435,800 до 435,900 МГц CW/USB.
- Digital Mode JD.
Uplink 145,850, 145,870 и 145,910 МГц FM.
Downlink 435,910 МГц 1 200 Бод BPSK или 9 600 Бод FSK.
Позывной 8J1JCS.
- Digi-talker Mode.
Downlink 435,910 МГц.
- Функционирует попеременно в цифровом режиме и как Digi-источник сообщений.
- JAS-2 был успешно выведен на орбиту 17 августа 1996 года из Космического центра Tanegashima.
- Mike, KF4FDJ, собрал очень информативный документ относительно FO-29, касающийся аналогового, цифрового и режима Digi-источника сообщений. Можно получить по электронной почте копию документа Mike: kf4fdj@amsat.org.
- Mineo, JE9PEL, модифицировал свою программу приема и анализа телеметрии спутника FO-29. Программное обеспечение автоматически анализирует всю цифровую телеметрию от спутника типа текущего напряжения и температуры. JE9PEL модификация FO-29/software доступна в Интернете по адресу <http://www.ne.jp/asahi/hamradio/je9pel/>.

Спутники с аппаратурой для цифровой связи

□ PCsat

- Uplink/Downlink 145,827 МГц 1 200 Бод AX-25 AFSK через PCSAT-1.
- Aux/Uplink 435,250 МГц 9 600 Бод через PCSAT (off).
- APRS Downlink 144,390 МГц (Region 1).
- Функционирует нормально.
- Выведен на орбиту 30 сентября 2001 года ракетой Athena-1 с космодрома Kodiak на Аляске.
- Информацию по спутнику можно получить в Интернете по адресу <http://pcsat.aprs.org/>.

□ TIUNGSAT-1

- Uplink 145,850 или 145,925 МГц 9 600 Бод FSK.
- Downlink 437,325 МГц.
- Broadcast-позывной — MYSAT3 — 11.
- BBS-позывной — MYSAT3 — 12.
- Функционирует в цифровом режиме 38 400 Бод FSK.
- Выведен на орбиту 26 сентября 2000 года с космодрома Байконур.
- TiungSat-1 является первым микроспутником Малайзии и в дополнение к коммерческой информации о состоянии Земли и погоды, будет работать FM и FSK любительская радиосвязь. Превосходная передача со спутника телеметрии говорит о нормальном состоянии радиоаппаратуры. О превосходной работе на скорости 38 400 при мощности 8 Вт сообщает Chris G7UPN. Он также сообщает, что для работы со спутником сначала нужно послать ему запрос на связь. После этого спутник проверяет напряжение на батареях и, если все нормально, начинает отвечать.

□ UOSAT UO-22

- Uplink 145,900 или 145,975 МГц FM 9 600 Бод FSK.
- Downlink 435,120 МГц FM.
- Broadcast-позывной — UOSAT5 — 11.
- BBS — UOSAT5 — 12.
- Функционирует.
- Выведен на орбиту 17 июля 1991 года ракетой Ariane с космодрома во Французской Гвиане.
- Chris Jackson, G7UPN, сообщает ANS, что на UO-22 ранее наблюдалось длительное нахождение спутника под воздействием прямого солнечного излучения, в результате чего значительно возросла температура внутри спутника. Контроллеры перевели спутник в инвертированный режим, чтобы уменьшить температуру. Это действительно уменьшило температуру на различных системах (типа батарей) на 5—10 единиц. Неудачный побочный продукт этого, что downlink является теперь совершенно слабым. Только 145,900 МГц приемник пригоден для использования для связи в настоящее время.
- Большое количество информации относительно спутника доступно в Интернете по адресу <http://www.sstl.co.uk/>.

□ OSCAR-11

- Downlink 145,825 МГц FM, 1 200 Бод PSK.
- Маяк 2401,500 МГц.

- Функционирует.
- Выведен на орбиту 1 марта 1984 года ракетой Delta-Thor с военного космодрома в Калифорнии.
- OSCAR-11 праздновал семнадцатый день рождения 1 марта 2001 года. В течение периода от 15 декабря до 14 января 2002 года хорошие сигналы были получены от маяка на частоте 145 МГц. Напряжение батареи в течение суток было неизменным. Среднее наблюдаемое значение DC 13,8 В, в диапазоне от 13,4 до 14,1 В.
- Действующий план работы следующий:
 - ◊ ASCII-состояние (210 секунд);
 - ◊ ASCII-бюллетень (60 секунд);
 - ◊ BINARY SEU (30 секунд);
 - ◊ ASCII TLM (90 секунд);
 - ◊ ASCII WOD (120 секунд);
 - ◊ ASCII бюллетень (60 секунд);
 - ◊ BINARY ENG (30 секунд).
- ASCII-бюллетень — в настоящее время статическое сообщение, в котором детализируются режимы и частоты всех активных любительских спутников радио. Большое количество информации относительно OSCAR-11 доступно в Интернете по адресу <http://www.users.zetnet.co.uk/clivew/>.

□ PACSAT AO-16

- Uplink 145,90, 145,92, 145,94, 145,86 МГц FM используется специальный вид модуляции Манчестер 1200 Бод FSK.
- Downlink 437,025 МГц SSB (RC-BPSK 1200 Бод PSK).
- Маяк 2401,1428 МГц.
- Broadcast-позывной — PACSAT — 11.
- BBS — PACSAT — 12.
- Функционирует частично.
- Выведен на орбиту 22 января 1990 года ракетой Ariane с космодрома во Французской Гвиане.
- Новую WOD-совокупность текущей графики и текущие данные по телеметрии можно найти в Интернете по адресу <http://www.telecable.es/personales/ea1bcu>.

□ UoSAT-12 UO-36

- Uplink 145,960 МГц 9600 Бод FSK.
- Downlink 437,025 МГц 437,400 МГц.

- Broadcast-позывной — UOSAT12 — 11.
- BBS — UOSAT12 — 12.
- UoSAT-12 был успешно выведен на орбиту 21 апреля 1999 года с российского космодрома Байконур.
- По VK5HI информация общего пользования доступна на AMSAT-NA странице в Интернете по следующему адресу: <ftp://ftp.amsat.org/amsat/software/win32/display/ccddsp97-119.zip>.
- Дальнейшая информация относительно UO-36 доступна в Интернете по адресу <http://www.sstl.co.uk/>.

□ ITAMSAT IO-26

- Uplink 145,875, 145,900, 145,925, 145,950 МГц FM 1200 Бод.
- Downlink 435,822 МГц SSB.
- Broadcast-позывной — ITMSAT1 — 11.
- BBS — ITMSAT1 — 12.
- Функционирует частично, функция digipeater включена.
- IO-26 был выведен на орбиту 26 сентября 1993 года ракетой Ariane с космодрома во Французской Гвиане.

Спутники, которые уже не слышны

Перечисленные ниже спутники находятся на орбитах, но не функционируют в настоящее время.

□ TMSAT-1 K — 31

- Uplink 145,925 МГц (FSK — 9600 Бод).
- Downlink 436,925 МГц (FSK — 9600 Бод).
- Callsign: TMSAT1 — 11, BBS — TMSAT1-12.
- Выведен на орбиту 10 июля 1998 года с космодрома Байконур.
- С 18 декабря 2000 года никакой информации от спутника не получено. G7UPN, менеджер UoSAT, сообщил, что такое положение является временным.
- Программу ProcMail 2.00G, разработанную G7UPN для обработки файлов с изображениями от K-31, можно взять в Интернете по адресу <http://www.amsat.org/amsat/software/win32/wisp>.

□ LUSAT LO-19

- Uplink 145,84 145,86 145,88 145,90 МГц FM используется Манчестер 1200 Бод FSK.

- CW downlink 437,125 МГц.
- Цифровой downlink 437,150 МГц SSB RC-BPSK 1 200 Бод PSK.
- Broadcast-позывной — LUSAT — 11.
- BBS — LUSAT — 12.
- В настоящее время функционирует частично. CW-маяк посылает восемь каналов телеметрии и один канал состояния. В настоящее время BBS-обслуживание не доступно. Digipeater неактивен.
- Выведен на орбиту 22 января 1990 года ракетой Ariane с космодрома во Французской Гвиане.
- Общую информацию и выборки телеметрии можно найти в Интернете по адресу: <http://www.telecable.es/personales/ea1bcu/lo19.htm>.

□ SAUDISAT-1A

- Uplink пока не объявлен.
- Downlink 437,075 МГц.
- Broadcast-позывной — SAUSAT1-11.
- BBS — SAUSAT1-12.
- Функционирует, вероятно, в режиме наладки.
- Выведен на орбиту 26 сентября 2000 года с космодрома Байконур.
- Должен был работать в режиме 9600 Бод в системе store-and-forward, также должен был работать аналоговый FM-репитер. Спутник разработан и построен в Научно-исследовательском институте Королевства Саудовская Аравия. За восемь месяцев пребывания этого спутника на орбите ANS не получил о нем никакой информации.

□ SAUDISAT-1B

- Uplink еще не объявлен.
- Downlink 436,775 МГц.
- Broadcast-позывной — SAUSAT2-11.
- BBS — SAUSAT2-12.
- Функционирует, вероятно, в режиме наладки.
- Выведен на орбиту 26 сентября 2000 года с космодрома Байконур.
- Все остальные сведения аналогичны предыдущему спутнику.

□ SUNSAT SO-35

- Mode J Uplink 145,825 МГц FM.
- Mode J Downlink 436,250 МГц FM.
- Mode B Uplink 436,291 МГц FM.

- Mode B Downlink 145,825 МГц FM.
- Не функционирует.
- SUNSAT был выведен на орбиту 23 февраля 1999 года на борту ракеты Дельта II из Военно-воздушной базы Vandenberg в Калифорнии.
- SUNSAT допускает работу в режиме Packet Radio на скоростях 1200 и 9600 Бод с промежуточным накоплением информации и голосовой репитер системы "попугай", которая будет использоваться прежде всего для образовательных показов. Спутник имеет две VHF, и две UHF приемно-передающих системы.
- 1 февраля 2001 года от команды SUNSAT было получено объявление, что на спутнике произошел непредвиденный процесс, дальнейший контакт со спутником маловероятен. Более полную информацию можно получить в Интернете по адресу <http://sunsat.ee.sun.ac.za/ham1.htm>.

□ РАДИО СПОРТ RS-13

- Uplink от 21,260 до 21,300 МГц CW/SSB.
- Downlink от 29,460 до 29,500 МГц CW/SSB.
- Downlink от 145,860 до 145,900 МГц CW/SSB.
- Маяк 145,860 МГц.
- Робот Uplink 145,840 МГц.
- Не функционирует. Последнее время работал в режиме T с 2-метровым и 10-метровым downlink и 15-метровым uplink.
- Выведен на орбиту 5 февраля 1991 года с космодрома Байконур. О работе через спутник RS-12/13 есть материалы в Интернете по адресу <http://www.qsl.net/ac5dk/rs1213/rs1213.html>.

□ KITSAT KO-23

- Uplink 145,900 МГц FM (9 600 Бод FSK).
- Downlink 435,175 МГц FM.
- Не функционирует.
- Выведен на орбиту 10 августа 1992 года ракетой Ariane с космодрома во Французской Гвиане.

□ KITSAT KO-25

- Uplink 145,980 МГц FM 9 600 Бод FSK.
- Downlink 436,500 МГц FM.
- Broadcast-позывной — HL02-11.
- BBS — HL02-12.
- Не функционирует.

- Выведен на орбиту 26 сентября 1993 года ракетой Ariane с космодрома во Французской Гвиане.

□ TECHSAT-1B GO-32

- Downlink 435,225 МГц используется HDLC-телеметрией.
- TechSat-1B был успешно выведен на орбиту с космодрома Байконур 10 июля 1998 года.
- Из последних сообщений ясно, что спутник не имеет постоянно работающего продолжительного маяка, а каждые 30 секунд передает пакеты информации длительностью по 3 с на скорости 9600 Бод. TECHSAT группа создала местную страницу спутника TECHSAT, расположенную в Интернете по адресу <http://www.technion.ac.il/pub/projects/techsat/docb.html>.

□ PANSAT PO-34

- Uplink/downlink частоты никогда не сообщались.
- Спутник недоступен в настоящее время для общих uplink-передач.
- PANSAT, разработанный Военно-морской Школой, был выведен на орбиту из космического корабля Шаттл (в течение STS-95) 30 октября 1998 года. В начальный период обещали, что данные по PANSAT-спектру распространения, по цифровому transponders — будут доступными для операторов любительского радио наряду с программным обеспечением, чтобы использовать эту технологию. До настоящего времени этого не случилось.
- Для получения более подробной информации, посетите официальную страницу PANSAT в Интернете по адресу <http://www.sp.nps.navy.mil/pansat/>.
- О PANSAT была опубликована статья в журнале AMSAT-NA в номерах за июль-август 1999 года. Статья написана KD6DRA и N7HPR.

□ DOVE-17

- Downlink 145,825 МГц FM 1200 Бода AFSK.
- Beacon 2401,220 МГц.
- Не функционирует.
- Выведен на орбиту 22 января 1990 года ракетой Ariane с космодрома во Французской Гвиане.
- DOVE-17 прекратил передачи в марте 1998 года. 145,825 МГц и 2401,220 МГц downlinks — не ответил на запросы и команды станции управления. Дополнительной информации нет.

☐ WEBERSAT WO-18

- Downlink 437,104 МГц SSB 1 200 Бода PSK AX.25.
- Не функционирует.
- Выведен на орбиту 22 января 1990 года ракетой Ariane с космодрома во Французской Гвиане.

WO-18, как сообщается, находится в MBL-режиме после аварийного отказа программного обеспечения. Дополнительной информации нет.

☐ SEDSAT-1 SO-33

- Downlink 437,910 МГц FM 9600 Бод FSK.
- Спутник в настоящее время недоступен для uplink, передачи изображения и transponder. Усилия по восстановлению были неудачны.
- Выведен на орбиту 24 октября 1998 года ракетой Delta-2 с космодрома на мысе Канаверал в штате Флорида (США).
- SEDSAT-1 предназначался для практики студентов при исследовании и разработках спутников.
- Дополнительной информации нет.

Расчет элементов орбит ИСЗ

Чтобы начать работу со спутником, нужно следующее:

- ☐ уметь произвести расчет появления спутника в зоне радиовидимости вашей станции;
- ☐ иметь антенны и аппаратуру, способные нормальным образом получать сигналы от спутника и передавать сигналы на спутник.

В этом разделе будет описан один из возможных вариантов расчета времени появления спутника в зоне радиовидимости вашей станции, времени нахождения спутника в зоне радиовидимости и определение азимута точки входа спутника в зону радиовидимости. Некоторые BBS в любительской пакетной сети предлагают для удаленных пользователей специальную "услугу" — расчет этих же самых параметров. Но я не советую пользоваться этой "услугой", если вы хотите иметь точные данные. Подобные расчеты нужно делать самому, на своем компьютере. По какой программе эти расчеты делать, должен решать сам радиолюбитель. Программ для подобных целей много, все они красочные и привлекательные, поэтому выбирайте сами.

Далее приведу описание моего метода, которым пользуюсь много лет.

Чтобы начать проведение расчетов указанных выше элементов орбит, следует заглянуть в Интернет и взять там очень необходимые для проведения расчетов "кеплеровские данные".

Кеплеровские данные

В начале XVII века немецкий ученый Иоганн Кеплер установил законы, по которым движутся планеты. Движения спутников также подчиняются законам Кеплера, поэтому каждому из спутников соответствует определенный набор данных, которые рассчитываются по законам Кеплера и характерны только для этого конкретного спутника в данный период времени. Известно, что спутник с каждым витком на какое-то расстояние приближается к Земле, следовательно, его координаты в небесном пространстве изменяются. Эти изменения вынуждают периодически производить перерасчеты кеплеровских данных. Расчеты по устаревшим данным приводят к ошибкам.

Радиолюбители, как правило, для своих расчетов применяют кеплеровские данные, которые периодически обновляет и публикует NASA — Национальное агентство по авионавигации США. В Интернете файл с кеплеровскими данными можно взять в нескольких местах.

- На странице AMSAT-NA, расположенной по адресу <http://www.amsat.org/>.
- На странице по адресу <http://logsat.com/>. Эта страница рекламирует программу расчета спутниковых орбит LOGSAT, на ней можно посмотреть красочную карту с расположением облаков над Европой и над Северной Америкой. В правой колонке, примерно посередине страницы, находится маленькая статья, озаглавленная "Keplerian Elements". Внизу выделенный текст "Download latest keps.zip(~50Kb)". Если щелкнуть кнопкой мыши на этом тексте, то вы скачаете в свой компьютер текстовый файл с кеплеровскими данными по всем спутникам, находящимся в космосе. В этом файле приведены данные по очень большому количеству спутников и вам из этого файла следует взять только первые 100—150 строк. Листинг 4.1 как раз и представляет собой нужный для нас кусок из файла keps.zip.

Кроме того, этот файл бывает на пакетных BBS в рубрике KEPLER.

О том, как использовать файл с кеплеровскими данными в расчетах, я расскажу чуть позже.

Листинг 4.1. Файл keps.tle

AO-10

```
1 14129U 83058B 02033.18694793 .00000054 00000-0 10000-3 0 8617
2 14129 25.9498 229.6755 6066219 221.4578 71.5899 2.05868503112214
```

UO-11

```
1 14781U 84021B 02040.54396780 .00005950 00000-0 87244-3 0 7449
2 14781 98.0668 8.3056 0009438 341.0240 19.0615 14.75911187961024
```

RS-10/11

1	18129U	87054A	02039.55627138	.00000180	00000-0	17967-3	0	449
2	18129	82.9225	63.8908 0010859	181.6307	178.4815	13.72631583733066		
UO-14								
1	20437U	90005B	02039.68383657	.00000783	00000-0	31322-3	0	8878
2	20437	98.3284	95.1181 0009993	235.3588	124.6650	14.30982998628879		
UO-15								
1	20438U	90005C	02039.64671686	.00000603	00000-0	24969-3	0	6516
2	20438	98.2976	85.9790 0009101	247.8229	112.1983	14.29998105628574		
AO-16								
1	20439U	90005D	02040.24398332	.00000913	00000-0	36151-3	0	6867
2	20439	98.3712	105.1485 0010298	236.9368	123.0820	14.31165244628981		
DO-17								
1	20440U	90005E	02039.74312835	.00001081	00000-0	42319-3	0	6428
2	20440	98.3903	107.7232 0010348	240.1970	119.8182	14.31412403628970		
WO-18								
1	20441U	90005F	02039.54778349	.00000965	00000-0	38063-3	0	7188
2	20441	98.3858	106.9689 0010920	239.5314	120.4792	14.31251866628932		
IO-19								
1	20442U	90005G	02039.84753760	.00000955	00000-0	37583-3	0	6883
2	20442	98.4000	109.6430 0011221	238.5358	121.4730	14.31406582629020		
FO-20								
1	20480U	90013C	02038.61818672	-.00000017	00000-0	25580-4	0	3884
2	20480	99.0293	67.7003 0540686	135.8764	228.6810	12.83308609562295		
AO-21								
1	21087U	91006A	02040.50118917	.00000193	00000-0	18617-3	0	3311
2	21087	82.9357	234.3961 0033997	212.6530	147.2524	13.74851582553486		
RS-12/13								
1	21089U	91007A	02039.54108081	.00000227	00000-0	22435-3	0	4005
2	21089	82.9211	99.6653 0027639	252.2863	107.5276	13.74336559552273		
UO-22								
1	21575U	91050B	02039.91570570	.00001543	00000-0	51657-3	0	4257
2	21575	98.1235	46.5018 0007524	177.1622	182.9612	14.38638185554402		
KO-23								
1	22077U	92052B	02039.21869037	-.00000037	00000-0	10000-3	0	655
2	22077	66.0847	191.7688 0001344	174.2219	185.8819	12.86385691446024		
AO-27								
1	22825U	93061C	02040.24588147	.00000769	00000-0	32224-3	0	1937
2	22825	98.3202	83.1584 0007765	290.1487	69.8859	14.28684391436420		

IO-26

1	22826U	93061D	02040.19164727	.00000853	00000-0	35421-3	0	1459
2	22826	98.3227	84.0329	0008516	291.6588	68.3683	14.28881249436451	

KO-25

1	22828U	93061F	02040.12575562	.00000721	00000-0	29966-3	0	1425
2	22828	98.3187	84.2191	0009287	269.1829	90.8276	14.29262140404638	

POSAT

1	22829U	93061G	02040.19110928	.00001002	00000-0	40882-3	0	1708
2	22829	98.3192	84.7033	0009067	269.6696	90.3439	14.29419586436569	

RS-15

1	23439U	94085A	02039.12552304	-.00000028	00000-0	40255-3	0	6002
2	23439	64.8166	288.6167	0158145	185.4635	174.4568	11.27543901293268	

FO-29

1	24278U	96046B	02040.17028823	.00000109	00000-0	14388-3	0	4762
2	24278	98.5140	237.8759	0351444	107.3165	256.6735	13.52822842270681	

MO-30

1	24305U	96052B	02038.93559053	.00000205	00000-0	20364-3	0	5708
2	24305	82.9397	180.5351	0030016	165.4985	194.7039	13.73371305271922	

TO-31

1	25396U	98043C	02040.22296924	-.00000044	00000-0	00000+0	0	6626
2	25396	98.6635	118.9062	0003103	132.0723	228.0736	14.23267120186276	

GO-32

1	25397U	98043D	02038.91067672	.00000572	00000-0	27432-3	0	6083
2	25397	98.6607	116.9342	0001526	128.6207	231.5103	14.22781410186077	

SO-33

1	25509U	98061B	02039.54446907	.00004144	00000-0	82682-3	0	5053
2	25509	31.4332	109.7040	0361391	270.2081	85.7113	14.26551942171703	

PO-34

1	25520U	98064B	02038.57944908	.00007550	00000-0	43161-3	0	5261
2	25520	28.4614	202.8956	0007069	248.7521	111.2367	15.10819004180609	

ISS (ZARYA)

1	25544U	98067A	02040.45740741	.00071579	00000-0	84566-3	0	529
2	25544	51.6401	335.0506	0005966	56.8127	88.6217	15.60632592184101	

SO-35

1	25636U	99008C	02038.95090417	.00001556	00000-0	41691-3	0	5791
2	25636	96.4781	109.4652	0151645	83.7434	278.1010	14.42458035155654	

UO-36

1	25693U	99021A	02040.44519305	-.00017865	00000-0	-26474-2	0	6242
2	25693	64.5619	73.9545	0024395	220.6992	139.2288	14.74560309151069	

WO-39

```
1 26061U 00004A 02038.14711982 .00003984 00000-0 13588-2 0 4687
2 26061 100.2193 19.8512 0035154 261.3696 98.3498 14.36807209106457
```

OO-38

```
1 26063U 00004C 02038.96885264 .00001395 00000-0 50536-3 0 4380
2 26063 100.2187 19.7402 0036862 262.3279 97.3702 14.35108789106514
```

AO-37

```
1 26065U 00004E 02038.23225951 .00001587 00000-0 57234-3 0 4388
2 26065 100.2207 18.9399 0037079 265.7255 93.9686 14.35099703106391
```

SO-41

```
1 26545U 00057A 02038.72547566 .00004805 00000-0 68032-3 0 2441
2 26545 64.5603 105.6171 0054978 298.2923 61.2695 14.77129983 73669
```

SO-42

```
1 26549U 00057E 02038.63471968 .00004754 00000-0 68712-3 0 3873
2 26549 64.5553 108.1471 0058044 302.6600 56.8907 14.76139317 73610
```

AO-40

```
1 26609U 00072B 02040.30164575 -.00000471 00000-0 10000-3 0 1976
2 26609 7.0167 128.7673 7944406 19.6315 358.4314 1.25596762 5881
```

PCSAT

```
1 26931U 01043C 02038.73325451 .00001437 00000-0 60321-3 0 1073
2 26931 67.0461 142.8722 0005586 283.1424 76.9051 14.28691496 18659
```

SAUDISAT 1A

```
1 26545U 00057A 02038.72547566 .00004805 00000-0 68032-3 0 2441
2 26545 64.5603 105.6171 0054978 298.2923 61.2695 14.77129983 73669
```

UNISAT

```
1 26547U 00057C 02038.71519205 .00007378 00000-0 10366-2 0 25792
2 26547 64.5604 108.4539 0059905 305.4459 54.1119 14.76800924 73629
```

SAUDISAT 1B

```
1 26549U 00057E 02038.63471968 .00004754 00000-0 68712-3 0 3873
2 26549 64.5553 108.1471 0058044 302.6600 56.8907 14.76139317 73610
```

Как расшифровать кеплеровские данные

О том, каким образом можно расшифровать кеплеровские данные из двухстрочной таблицы, поясню на следующем примере. Ниже (листинг 4.2) размещена таблица в виде двух строк с кеплеровскими данными, относящимися к спутникам RS-12 и RS-13. Дело в том, что в корпусе одного спутника находится и аппаратура, относящаяся к RS-12, и аппаратура RS-13. Эта аппаратура работает поочередно. Совсем недавно этот спутник работал под позывным RS-13, в настоящее время — это RS-12.

Листинг 4.2. Кеплеровские данные спутника RS-12/13

```
RS-12/13
1 21089U 91007A 02039.54108081 .00000227 00000-0 22435-3 0 4005
2 21089 82.9211 99.6653 0027639 252.2863 107.5276 13.74336559552273
```

Листинг 4.3 содержит таблицу, в которой каждый элемент кеплеровских данных зашифрован как набор одних и тех же символов (букв). В табл. 4.1 находится описание всех кеплеровских данных, задействованных в последующих расчетах.

Листинг 4.3. Вспомогательная таблица

```
Название спутника
1 CCCCCU YYNNN Q TTTTT.TTTTTTTT .DDDDDDDD VVVVV-V GGGGG-G 0 SSSZ
2 CCCCC III.IIII RRR.RRRR EEEEEEE PPP.PPPP AAA.AAAA MM.MMMMMMMMOOOOOZ
```

Таблица 4.1. Расшифровка данных

Символ	Что обозначает сочетание символов
C	Порядковый номер спутника, присвоенный ему в каталоге NASA
Y	Две последних цифры из года запуска спутника
N	Порядковый номер запуска данного спутника среди всех других запусков в течение года
Q	Номер данного спутника среди всех других объектов, выведенных на орбиту данной ракетой-носителем
T	Эпохальное время — время прохождения спутником через точку восходящего узла, в этот момент времени должны быть измерены и зафиксированы все основные параметры орбиты. Далее эти параметры будут называться эпохальными. Восходящий узел — момент пересечения спутником плоскости экватора при движении с юга на север
D	Цифра коррекции движения — положительная или отрицательная величина, учитывающая воздействие гравитационных сил Солнца и Луны на скорость движения спутника
V, G, O, Z	В расчетах не применяются
Z	Контрольная сумма всех цифр, расположенных в строке
I	Угол наклона плоскости орбиты спутника к плоскости экватора. Может изменяться от 0 до 180. При I = 0 плоскость орбиты спутника совпадает с плоскостью экватора, при этом спутник движется с запада на восток.

Таблица 4.1 (окончание)

Символ	Что обозначает сочетание символов
	При $I = 90$ спутник всегда пролетает точно над северным и южным полюсами Земли.
	При $I = 180$ плоскость орбиты спутника совпадает с плоскостью экватора, при этом спутник движется с востока на запад
R	Долгота точки экватора, над которой проходит спутник в момент фиксации эпохального времени (долгота эпохального восходящего узла)
E	Эксцентриситет — все орбиты эксцентричные, т. е. отличны от круга. $E = 0$ — абсолютно круговая орбита, с возрастанием этой величины эллиптичность орбиты увеличивается, орбита все более вытягивается
P	Параметр перигея — измеряется как угол из центра Земли между направлением на точку восходящего узла и направлением на точку перигея орбиты. При $P = 0$ точка перигея совпадает с восходящим узлом
A	Средняя аномалия (МА) показывает положение спутника на орбите относительно перигея
M	Средняя скорость — число орбит за сутки (24 часа, 1440 мин, 86 400 сек)
O	Номер эпохальной орбиты, расчетная величина, не всегда совпадающая с действительной

Программа ORBITA

Еще в середине 90-х годов прошлого столетия мною была разработана, на базе материалов из журнала "Радио", программа для расчета элементов орбит. Программа должна была служить специальным инструментом, позволяющим самому, основываясь на экспериментальных данных, методом прогнозирования, создавать эпохальные величины, по которым затем выполнялись расчеты. В те годы о существовании файлов с кеплеровскими данными мы ничего не знали и вынуждены были идти своим путем. Иногда газета "Советский патриот" публиковала кое-какие данные, необходимые для расчетов, но потом эта информация прекратилась. Так что каждый вынужден был создавать свои методы, чтобы делать необходимые расчеты.

С тех пор программа ORBITA несколько раз дорабатывалась. На сегодняшний день в ней вижу еще многие возможности для доработки, но заниматься нет времени. В главе 5 этой книги я помещаю исходные коды этой программы, чтобы каждый желающий доработал ее на свой вкус.

В то время, когда я пишу эти строки, моя радиостанция включена на частоте работы спутника RS-12. Предварительные расчеты времени подхода спут-

ника выполнены на программе ORBITA. Погрешность совпадения фактического времени с расчетным составляет не более плюс/минус 2 минуты. Это исключительная точность. Программы, заложенные в BBS F6FBB, в лучших случаях дают погрешность в 2—3 раза худшую.

Основные положения программы ORBITA

Программа предназначена для радиолюбителей, увлекающихся радиосвязью через радиолюбительские искусственные спутники Земли. Может выполняться на любых совместимых с IBM PC компьютерах под управлением MS-DOS и Windows 95/98.

Программа проводит полные расчеты для спутников, расположенных на круговых орбитах и частичные расчеты для спутников на эллиптических орбитах.

Предназначение программы — рассчитать орбиты спутника, которые проходят через зону радиовидимости назначенной станции слежения и выдать данные о времени вхождения спутника в зону радиовидимости и выхода спутника из этой зоны в любые заданные дни и часы. Необходимость разработки этой программы была вызвана тем, что ни одна из известных автору зарубежных программ не выдавала таких данных. Все известные программы работают в режиме реального времени и выдают данные только о том, где в данный момент находится заданный спутник или группа спутников.

Программа имеет следующие варианты (версии):

- ☐ orbita 1.01 — вариант без какой бы то ни было коррекции данных с выдачей расчетных данных на экран;
- ☐ orbita 1.02 — аналог предыдущего варианта, но может выдавать данные как на экран, так и на принтер;
- ☐ orbita 1.11 — вариант с одним уровнем коррекции данных и выдачей расчетных данных на экран;
- ☐ orbita 1.12 — аналог предыдущего варианта, но может выдавать данные как на экран, так и на принтер.

Исполняемый файл имеет в обозначении слово orbit и номер варианта, например: orbit101.exe, orbit111.exe и т. д.

В состав комплекта входят следующие файлы (пример для v.1.01):

- ☐ orbit101.exe — исполняемый файл;
- ☐ orbit101.cfg — конфигурационный файл;
- ☐ orbita.doc — файл описания работы программы;
- ☐ tle1.dat — файл с кеплеровскими данными NASA;
- ☐ test_kep.exe — вспомогательная программа (утилита).

Расчеты выполняются на основании данных из файла keps.zip, публикуемого американским Национальным агентством по аэронавтике (NASA). Этот файл можно скачать из Интернета, например, со страницы по адресу <http://www.logsat.com>. Практически нужны 100—200 строк этого файла с информацией о радиолюбительских спутниках. Также аналогичные файлы можно встретить и в BBS любительской сети Packet Radio. Информация по спутнику представлена в виде двухстрочной информации с предшествующим названием спутника. Ниже дан пример информации в файле по спутнику RS-12/13 и RS-16.

RS-12/13

```
1 21089U 91007A 99083.22197456 .00000091 00000-0 80714-4 0 1482
2 21089 82.9253 160.2735 0028556 304.5201 55.3249 13.74126041407745
```

RS-16

```
1 24744U 97010A 99084.70201530 .00042969 00000-0 74392-3 0 4517
2 24744 97.2218 351.0797 0002924 272.8652 87.2892 15.50908330115392
```

Программа отыскивает необходимый спутник, расшифровывает прочитанную в двух строках информацию, преобразует эту информацию в удобный для работы вид, записывает в память и выводит расширенный список параметров орбит спутника на экран. Далее программа переводится в режим расчета, где каждая строка данных появляется после нажатия клавиши <Enter> или вводе цифры 2.

Ввод цифры 1 возвращает программу в промежуточное меню. Никаких сложностей при работе с программой не выявлено.

Конфигурационный файл служит для ввода в программу индивидуальных данных по станции слежения — название станции, ее широта и долгота. Также в этом файле располагаются числовые данные для коррекции.

Программа проводит все расчеты во времени GMT (UTC), а выдавать на экран все данные может в режиме местного времени, например, в MSK. Для этого служит четвертая строка конфигурационного файла, в которой должно находиться число часов разницы между местным временем и GMT.

Если возникнет желание просмотреть результаты расчетов в GMT, то нужно в четвертой строке конфигурационного файла поставить число 0.

Практическая работа с программой

1. Приобрести наиболее свежий файл с кеплеровскими данными NASA и поместить его в каталог программы под именем tle1.dat (или иным, но тогда при запуске программы каждый раз придется вводить имя файла).
2. Откорректировать конфигурационный файл. Текст конфигурационного файла приведен в листинге 4.4.

Листинг 4.4. Текст конфигурационного файла

```
г.Людиново      — название города — центра слежения, не более 15 символов
53.85           — сев. широта города, не более 7 знаков
324.55          — зап. долгота города, не более 7 знаков
4               — 3-зимой, 4-летом, (разница MSK и GMT в часах ( 2 знака))
```

Этот файл должен иметь шесть значащих строк, каждая из которых начинается с первой позиции строки. Никаких промежутков между этими строками быть не должно. Никаких пустых строк перед первой!!!

<RA3XB>

Заполнение этого файла не должно вызывать сложностей. Необходимо только следить за переходом с зимнего времени на летнее и наоборот. Если вы привыкли работать с временем в GMT, то поставьте в четвертой строке ноль. Постарайтесь поточнее определить широту и долготу вашего местонахождения. От этого будет зависеть точность расчетов.

3. Запустите программу ORBITA. На экране появляется первое меню, состоящее из трех пунктов.

- <I> — если нажать на эту клавишу, на экране должна появиться информация о программе.
- <N> — после нажатия на эту клавишу на экране появляются строки с информацией по всем кеплеровским данным выбранного спутника.
- <T> — после нажатия на эту клавишу на экране возникает запрос на ввод имени файла с кеплеровскими данными. Если вы предварительно дали этому файлу название tle1.dat, то следует просто нажать клавишу <Enter>. Если у вас этот файл имеет другое название, то следует это название ввести и нажать клавишу <Enter>.

Сразу же после этой процедуры программа просит ввести название спутника. Название следует вводить точно так, как оно стоит в файле с кеплеровскими данными. Должно четко соблюдаться написание букв в верхнем или нижнем регистре, наличие тире и т. д. Заканчивается ввод нажатием на клавишу <Enter>. На экран выводится информация по выбранному спутнику. Для продолжения следует снова нажать клавишу <Enter>.

Программа выдает на экран данные из конфигурационного файла. Если все данные соответствуют действительности — нажмите клавишу <Enter>. Появляется запрос ввести дату. Сначала вводится число, затем вводится запятая, затем две цифры, обозначающие месяц, и снова вводится запятая, затем две последние цифры года. Наличие запятых между цифрами обязательно. Заканчивается процедура ввода даты нажатием на клавишу <Enter>.

зоны радиовидимости. В десятом столбце распечатаны примерные значения направления на спутник в момент входа его в зону радиовидимости. В следующем столбце печатается "?" и останавливается курсор. Чтобы распечатать следующую строку, следует нажать клавишу <Enter>, а чтобы выйти в меню, нужно ввести цифру 1 и нажать клавишу <Enter>.

Вся выводимая на экран информация копируется в текстовый файл text.txt. Этот файл располагается в каталоге программы ORBITA. Его всегда можно взять и распечатать на принтере, а также сделать копию и сохранить эту копию для дальнейшего использования. Нужно помнить, что при просмотре информации о другом спутнике прежняя информация стирается. В файле text.txt находится информация только о последнем просмотренном спутнике.

Как принимать информацию с RS

Если у вас имеется радиоаппаратура, предназначенная непосредственно для работы с интересующим вас спутником, и если вы освоили приемы расчета орбит, то проблемой может стать изготовление или приобретение соответствующих антенных систем. Дело в том, что сигналы от спутников приходят с сильными замираниями и порой бывают очень слабыми. Поэтому для работы со спутниками применяются направленные антенны с большим коэффициентом усиления и позволяющие вести прием сигналов с различной поляризацией. При этом антенны постоянно должны поворачиваться вслед за перемещающимся по небу спутником. На некоторых станциях установлены специальные следящие системы, которые в автоматическом режиме постоянно следят за перемещениями спутника и поворачивают антенны, ориентируя их на спутник.

Большинство спутников работают на частотах ультракоротковолнового диапазона (УКВ), начиная от 144 МГц до 24 000 МГц. Отечественные производители радиоаппаратуры пока никаких аппаратов на частоты этих диапазонов для нас не изготавливают, поэтому нужно или приобретать импортную аппаратуру, или строить самому.

Пока вы будете решать, какую аппаратуру вам следует использовать для работы со спутниками, — делать самому или покупать импортную, предлагаю понаблюдать за сигналами маяка спутника RS-12 и за работой других радиолюбителей через ретранслятор этого спутника, который передает свои сигналы на Землю на частотах 29,410—29,450 МГц.

Если у вас имеется связной радиоприемник с диапазоном 29 МГц, то подключайте к радиоприемнику антенну из длинного провода, определяйте времена подхода спутника и слушайте. Зачастую бывает, что обычные связные приемники не имеют достаточной чувствительности и сигналы спутника слышны на уровне шумов или даже хуже. В этом случае огромную услугу вам окажет аппарат, описание и схема которого находится в предыдущей главе этой книги. Аппарат называется "преселектор", он является предвари-

тельным усилителем высокой частоты, обеспечивающим приемнику очень высокую чувствительность при низком уровне собственных шумов. Чувствительность этого аппарата достаточна для нормального приема информации со спутника RS-12 и аналогичных.

Если у вашего связного приемника нет диапазона 29 МГц, то можно изготовить смесительный каскад с кварцевым гетеродином, изображенный на рис. 4.5.

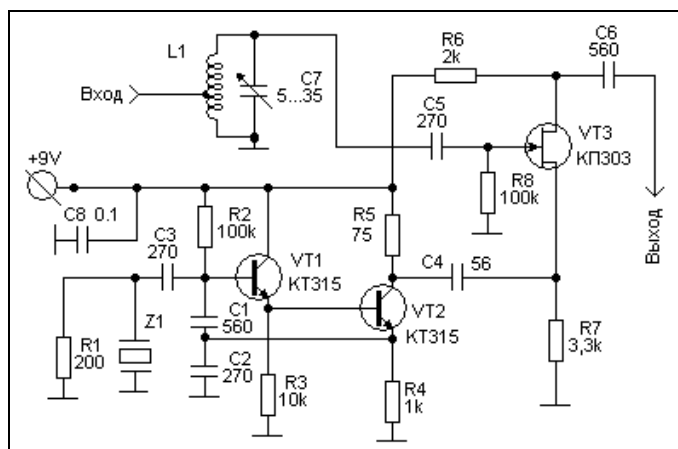


Рис. 4.5. Принципиальная схема смесительного каскада

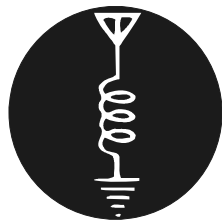
Смесительный каскад выполнен на транзисторе VT3. На транзисторах VT1 и VT2 собран кварцевый генератор. Рабочая частота кварца Z1 должна подбираться исходя из имеющейся у вас приемной аппаратуры. Например, мною для приема сигналов от спутника RS-12 используется радиоприемник P-250M, у которого нет диапазона 29,4 МГц. В аналогичный генератор я поставил кварц с рабочей частотой 10 МГц, прием сигналов со спутника получается при установленной на радиоприемнике частоте 19,4 МГц. При приеме происходит сложение частоты настройки радиоприемника с частотой кварцевого генератора и в итоге идет прием нужной частоты 29,4 МГц.

Число витков катушки L1 следует выбрать из табл. 3.1. Отвод от катушки для подключения входа должен быть сделан от 1/4 витков катушки L1, считая от заземленного конца.

Резистор R1 служит для уменьшения тока, протекающего через кварц. Если у вас используется малоактивный кварц и генератор не начинает работать, то этот резистор следует убрать.

На вход смесительного каскада следует подать сигнал от преселектора, а выход смесительного каскада соединить с антенным входом вашего радиоприемника.

Глава 5



Как сделать программу

Предисловие

Начиная примерно с 1994 года, в нашей стране начали становиться доступными компьютеры IBM PC и появилась возможность приобретать по различным каналам программы для любительской радиосвязи, разработанные иностранными программистами.

В те годы я старался освоить все попадавшие мне в руки программы по радиосвязи. Восхищался их красочностью, большим разнообразием команд и функций. Переводил файлы документации многих программ на русский язык, делал к этим программам свои вспомогательные файлы, проводил и другую работу, которую можно назвать словом "русификация".

По мере того, как мною осваивались разные программы, я стал замечать, что очень многие из них по качеству приема сигналов работают в несколько раз хуже созданных мною ранее аналогичных программ для "Радио-86РК", хотя по красочности были бесподобными. Потом стал замечать другую особенность — многие из задействованных в программе команд и функций оставались невостребованными. И еще. Программы, содержащие в своем составе несколько видов цифровой связи, как правило, имеют более плохие параметры, содержат наиболее примитивные подпрограммы приема и передачи.

Так постепенно я пришел к выводу о том, что нужно снова начать создавать свои программы, которые должны по основным параметрам — приему и передаче без ошибок нужной информации — превосходить многие зарубежные. В то же время на красочное оформление заставок и рабочих экранов обычно не обращаю большого внимания. Не считаю этот параметр главным в программе для радиосвязи.

В этой главе я привожу исходные тексты некоторых своих программ с описаниями их основных особенностей. Это делается мною в надежде на то, что тексты исходных кодов разработанных мною программ помогут начинающим программистам создавать свои программы на эту же тему, по этому же виду радиосвязи. Пускай тексты моих программ не всегда совершенны с точки зрения профессиональных программистов, но все программы работоспособны и опробованы работой в эфире.

Эта глава и размещенные в ней файлы с исходными кодами программ ни в коем случае не являются учебными пособиями по программированию. Эти файлы просто рассказывают вам о том, как та или иная задача была мною решена в прошлые годы при создании определенной программы.

Буду считать свои цели достигнутыми, если кому-то удастся на базе разработанных мною исходных текстов создать новые превосходные программы, гораздо лучшие по всем параметрам.

Как сделана программа ORBITA

Программа ORBITA предназначена для расчета различных параметров орбит ИСЗ. Создана в конце 90-х годов прошлого столетия, несколько раз дорабатывалась. Разрабатывалась в среде программирования QBasic, хорошо компилируется в Turbo Basic. Программа хорошо работает как под управлением MS-DOS, так и под управлением Windows 95/98/ME/2000/XP.

Определенную ценность представляют именно те функции программы, в которых располагается весь математический аппарат, необходимый для проведения расчетов и производится выполнение этих расчетов.

Любители программировать в Visual Basic могут с успехом использовать эти функции при создании своих приложений для Windows. Но в таком случае программа будет работать только под управлением Windows.

Ниже приводится листинг исходного файла orbit101.bas. Листинг прерывается различными пояснениями.

Листинг 5.1. Файл orbit101.bas

```
'Программа для расчетов параметров орбит спутников,
' находящихся на круговых орбитах.
ON ERROR GOTO 10020
OPEN "orbit101.cfg" FOR INPUT AS #1: CLOSE 'Проверка
OPEN "tle1.dat" FOR INPUT AS #1: CLOSE ' файлов
10 CLS
   SCREEN 0, 1: WIDTH 80: COLOR 14, 1, 0: CLS
   ON ERROR GOTO 10000
   GOTO 18
15 T$ = "orbit101.cfg"
   OPEN T$ FOR INPUT AS #2
   LINE INPUT #2, A$
   Num = Num + 1
   LINE INPUT #2, B$
   LINE INPUT #2, C$
   LINE INPUT #2, D$
   PRINT A$: AA$ = LEFT$(A$, 15)
   PRINT B$: BB$ = LEFT$(B$, 7)
```

```

PRINT C$: CC$ = LEFT$(C$, 7)
PRINT D$: DD$ = LEFT$(D$, 2)
CLOSE #2
STAN$ = AA$           '- название станции слежения
LAT# = VAL(BB$)       '- сев. широта станции слежения
LON# = VAL(CC$)       '- зап. долгота станции слежения
TZ% = VAL(DD$)        '- разница между местн. временем и GMT
T1 = LAT#
L1 = LON#
PRINT
PRINT STAN$, LAT#, LON#, TZ%
PRINT : PRINT : PRINT
PRINT "  Удостоверьтесь в правильности данных в конфигурационном
файле."
PRINT : PRINT
INPUT "      Нажмите <Enter>"; C
CLS
RETURN

```

В предыдущем листинге сначала проводится проверка присутствия в каталоге нужных дополнительных файлов, затем, начиная с метки 15, проводится чтение информации из конфигурационного файла. Чтение проводится построчно, из каждой строки берется только определенное число символов, которые затем декодируются.

Листинг 5.2 представляет собой главную экранную заставку, на которой размещено первое меню.

Листинг 5.2. Продолжение 1

```

18  DEFSNG A-Z
20  COLOR 10, 4
    LOCATE 3, 20: PRINT "      П Р О Г Р А М М А   ORBITA v.1.01   ";
    LOCATE 4, 20: PRINT "  ДЛЯ РАСЧЕТА ДАННЫХ ПО ОРБИТАМ СПУТНИКОВ  ";
    LOCATE 5, 20: PRINT "    на IBM PC. Автор RA3XB,  апрель 1999г   ";
    COLOR 14, 1, 1: GOSUB 9400
    LOCATE 9, 20: PRINT "  <I> — информация о программе"
    LOCATE 10, 20: PRINT "  <N> — прочитать данные в режиме NASA  "
    LOCATE 11, 20: PRINT "  <T> — расчет параметров орбит по ORBIT_RS"
    COLOR 12, 1, 0: LOCATE 14, 20: PRINT "  <Esc> — выход в DOS": COLOR
14, 1, 1

```

```

100  GOSUB 9500
      LOCATE 16, 20, 0, 0, 7: COLOR 10, 4: PRINT "  ВЫБЕРИТЕ
<I,N,T,Esc>--> ";
      COLOR 30: PRINT STRING$(2, 176): Z$ = "IiNnTt" + CHR$(27)
110  K$ = INKEY$: IF K$ = "" THEN 110
      COLOR 14, 1, 1: LOCATE 16, 46: PRINT SPACE$(2): K = INSTR(Z$, K$)
      ON K GOTO 700, 700, 200, 200, 10100, 10100, 120, 120
      GOSUB 9510: COLOR 28, 1, 0: GOTO 100
120  CLOSE : CLS : END
200  GOSUB 300: GOTO 20

```

В листинге 5.3 приводится текст подпрограммы вывода на экран основных данных по выбранному спутнику. Сначала программа запрашивает ввод названия файла, содержащего кеплеровские данные, затем просит ввести название спутника. При этом написание названия спутника должно в точности соответствовать написанию его названия в файле с кеплеровскими данными. Не путать верхний и нижний регистры при написании букв.

Листинг 5.3. Продолжение 2

```

300  '***** Прочитать данные в режиме NASA *****
      CLS : COLOR 14, 1, 0: LOCATE 2, 10
      INPUT "Введите имя файла с данными (по умолчанию — tle1.dat): "; F$
      IF F$ = "" THEN F$ = "TLE1.DAT"
      LOCATE 3, 10: INPUT "Введите слово (название спутника) для поиска:
"; P$
      SP$ = P$ ' — название спутника
      OPEN F$ FOR INPUT AS #1
350  LINE INPUT #1, Test$
      Num = Num + 1
      IF EOF(1) THEN 360
      IF INSTR(Test$, P$) <= 0 THEN 350 LINE INPUT #1, A$
      LINE INPUT #1, B$
      PRINT "Line #"; Num; ": "; Test$
      PRINT A$
      PRINT B$
      CLOSE #1
      GL$ = A$: GK$ = B$
      NK$ = MID$(GL$, 3, 5): YZ$ = MID$(GL$, 10, 2): YY$ = MID$(GL$, 19,
2): TE$ = MID$(GL$, 21, 12)

```

```

DD$ = MID$(GL$, 34, 10): VV$ = MID$(GL$, 46, 7): GG$ = MID$(GL$,
55, 7)
II$ = MID$(GK$, 10, 7): RA$ = MID$(GK$, 18, 8): EE$ = MID$(GK$, 27,
7)
PP$ = MID$(GK$, 35, 8): AA$ = MID$(GK$, 44, 8): MM$ = MID$(GK$, 53,
11)
OO$ = MID$(GK$, 64, 5)

M# = VAL(MM$)           ' — число орбит в сутки
NR = VAL(OO$)           ' — номер эпохальной орбиты
PD# = VAL(DD$)          ' — коррекция на движение
PS# = 1 / VAL(MM$)      ' — период обращения — орбит/сутки
L0 = VAL(RA$)           ' — долгота восходящего узла
R0 = ((PS# * 1440 * 7.365 * 60 / 3.14) - 2 * 6371) / 2
                        ' — высота спутника над Землей
GS# = 360 / VAL(MM$)    ' — количество градусов долготы на оборот
YR = VAL(YY$)          ' — эпохальный год
IN = VAL(II$)

ET# = VAL(TE$)          ' — эпохальное время по NASA

CLS
LOCATE 2, 10: PRINT " Данные по спутнику "; P$
LOCATE 5, 2: PRINT "Номер спутн. по каталогу NASA          "; NK$;
LOCATE 6, 2: PRINT "Год запуска                             "; YZ$;
LOCATE 7, 2: PRINT "Время эпохальное                       "; ET#;
LOCATE 8, 2: PRINT "Коррекция на движение                 "; PD#;
LOCATE 9, 2: PRINT "Коррекция перемещения                 "; GG$;
LOCATE 10, 2: PRINT "Наклонение орбиты                      "; II$;
LOCATE 11, 2: PRINT "Долгота восходящего узла (R.A.A.N)    "; L0;
LOCATE 12, 2: PRINT "Эксцентриситет                        "; EE$;
LOCATE 13, 2: PRINT "Аргумент перигея                       "; PP$;
LOCATE 14, 2: PRINT "Средняя аномалия                       "; AA$;
LOCATE 15, 2: PRINT "Количество орбит за сутки             "; MM$;
LOCATE 16, 2: PRINT "Эпохальный номер орбиты               "; OO$;
LOCATE 17, 2: PRINT "Эпохальный год                         "; YR;
LOCATE 18, 2: PRINT "Период обращения (минут)              "; PS# * 1440;
LOCATE 19, 2: PRINT "Высота спутника                       "; R0;
LOCATE 20, 2: PRINT "Количество градусов долготы на оборот "; GS#;
LOCATE 23, 10: INPUT "Для выхода нажать <Enter>          "; C: CLS

```

```

        CLOSE #1
        RETURN
360    PRINT
        PRINT "    Проверьте, не перепутаны ли в названии спутника
        PRINT "          строчные и заглавные буквы!"
        PRINT
        PLAY "L3 O2 G L5 A L4 GFED L2 C" 'FEHLERCONDITION WO SIND...
        INPUT "          НЕТ такого спутника! "; C:
        RETURN

```

В листинге 5.4 должна располагаться информация, которую вы хотите выдать пользователю. Пока здесь ничего нет.

Начиная с метки 9400 продолжается оформление главной экранной заставки.

Начиная с метки 1000 идут строки обработки ошибок. Задается текст строк, которые будут выдаваться на экран в случае обнаружения ошибок.

Листинг 5.4. Продолжение 3

```

700  CLS : COLOR 14, 1, 1: PRINT "ЗДЕСЬ пока ничего нет!"
        INPUT "Нажмите <Enter> :"; C: GOTO 10
9400  COLOR 1, 2
        LOCATE 24, 6: PRINT "          ТОЛЬКО ДЛЯ НЕКОММЕРЧЕСКОГО
РАСПРОСТРАНЕНИЯ И ИСПОЛЬЗОВАНИЯ          ";
        LOCATE 25, 6: PRINT "          Разработана Тяпичевым Г.А. (с) — RA3XB —
в апреле 1999 года          ";
        COLOR 14, 1, 1: RETURN
9500  SOUND 990, 3: SOUND 32000, 1: SOUND 990, 1: SOUND 32000, 1: SOUND
990, 4
        RETURN
9510  FOR ISI = 1 TO 8
        SOUND 990, 1: SOUND 32000, 1
        NEXT ISI: RETURN
10000 PRINT ERL, ERR
        STOP
10020 IF ERR = 53 THEN CLS : GOTO 10030 ELSE GOTO 10000
10030 IF ERL = 80 THEN PRINT "orbit101.cfg не найден": GOTO 10060
        IF ERL = 90 THEN PRINT "tle1.dat не найден"
10060 PRINT "Файлы orbit101.cfg, и tle1.dat должны быть в текущей
директории!"
        PLAY "L3 O2 G L5 A L4 GFED L2 C" 'FEHLERCONDITION WO SIND...
        STOP

```

В листинге 5.5 размещаются строки программы начала расчета элементов орбит ИСЗ. В двух строках с операторами DATA приведен массив цифр, соответствующих числу дней от начала года до начала расчетного месяца. Одна строка служит для високосного года, другая — для обычного.

Далее программа запрашивает ввод даты и времени, начиная с которого должен выполняться расчет элементов орбит.

Листинг 5.5. Продолжение 4

```

10100    COLOR 14, 1, 0: CLS
        CLEAR
        COLOR 10, 4:
        LOCATE 4, 20: PRINT "
        LOCATE 5, 20: PRINT "          Приближенный расчет
        LOCATE 6, 20: PRINT "
        LOCATE 7, 20: PRINT "          орбит спутников RS.
        LOCATE 8, 20: PRINT "
        LOCATE 9, 20: PRINT "          Время — местное
        LOCATE 10, 20: PRINT "
        COLOR 14, 1, 0
        RD = 57.295779#
        RM = 6371.1
        TC = .0000308
        UD = 1.8E+07
        DATA 0,31,60,91,121,152,182,213,244,274,305,335
        DATA 0,31,59,90,120,151,181,212,243,273,304,334
        DIM MD(2, 12), Z0(2), ZB(2)
        FOR I = 0 TO 1: FOR J = 1 TO 12
        READ MD(I, J): NEXT J: NEXT I
        DEF FN YR (Y) = SGN(Y / 4 - INT(Y / 4))
        DEF FN RL (L) = L - 360 * INT(L / 360)
        LOCATE 15, 10
        INPUT "Нажмите клавишу <Enter>"; C

        GOSUB 300
        GOSUB 15
        L1 = FN RL(L1)
        'Перевести эпох. время ET# в календарное
        Z(1) = YR: Z(2) = ET#
        'корректировка данных NASA — эп. времени и RAAN

```



```

GOSUB 11600
HR = HR + TZ%      ' приведение к местному времени
MI = MI - 2        ' корректировка (уменьшение) эп. времени на 2 минуты
GOSUB 11200: EP = YR: GOSUB 11400
ET# = Z(2)
L0 = L0 - 6        ' корректировка (уменьшение) RAAN на 6 градусов

```

В двух из шести последних строк предыдущего листинга показана возможность корректировки времени (минуты) и RAAN (градусы). Это можно ввести в случае обнаружения постоянных больших несовпадений расчетного времени появления спутника в зоне радиовидимости и времени фактического.

Начиная с метки 10600 выводится второе меню, которое служит для начала расчетов, выбора нового спутника и выхода из программы.

Листинг 5.6. Продолжение 5

```

10110   CLS : PRINT : PRINT
        PRINT "      Введите текущее время сеанса связи через спутник ";
SP$
INPUT "      Календарная дата — день, месяц, год..."; DY, MN, YR
INPUT "      Время — час, мин, сек....."; HR, MI, SC
        PRINT : PRINT
        PRINT "      Проверьте введенные данные"
INPUT "      1 — повторный ввод "; C: ON C GOTO 10110
Z0(1) = Z(1): PD = PD#: R0 = (R0 + RM) / RM
NF = 120: Z0(1) = YR: Z0(2) = Z(2)
GOSUB 11200: EP = Z0(1): GOSUB 11400: ZB(1) = Z(1): ZB(2) = Z(2)
ND = INT((ZB(2) - ET#) / PS#)

        DT = (PS# + PD# * ND) / NF
        UT = (DT - TC) * UD

10600   CLS
        PRINT
        PRINT TAB(10); "      Спутник "; SP$; " орбита N"; NR + ND
        PRINT TAB(10); "      Наземная станция: "; STAN$
        PRINT : PRINT
        PRINT TAB(15); "      Примите решение: "
        PRINT TAB(15); "      ====="
        PRINT "      1 = выбрать новый спутник"

```

```

PRINT "          11 = выдать расписание сеансов связи"
PRINT "          21 = выход"
PRINT : PRINT : PRINT
INPUT CM: ON INT(CM / 10) + 1 GOSUB 15000, 16000, 19000
GOTO 10600

```

Начиная с метки 11200 начинаются строки программы с непосредственными математическими расчетами элементов орбит.

Каждая из подпрограмм имеет свою метку и краткое название, в котором фактически содержится все необходимое разъяснение.

Листинг 5.7. Продолжение 6

```

11200 REM*Перевод календ. времени в эпохальное
      Z(1) = YR
      Z(2) = MD(FNYR(YR), MN) + DY + ((SC / 60 + MI) / 60 + HR) / 24
      RETURN
11400 REM*Перевод Z в EP
      S = SGN(EP - Z(1)): IF S = 0 THEN GOTO 11450
      FOR YR = Z(1) + (S - 1) / 2 TO EP - (S + 1) / 2 STEP S
        Z(2) = Z(2) - S * (366 - FNYR(YR))
      NEXT YR: Z(1) = EP
11450 RETURN
11600 REM*Перевод Z в календарное время
11610 EP = Z(1) + INT((Z(2) - 1) / (366 - FNYR(Z(1))))
      GOSUB 11400
      IF Z(2) < 1 OR Z(2) >= 367 - FNYR(Z(1)) THEN GOTO 11610
      YR = Z(1)
      FOR IM = 1 TO 12
        IF Z(2) >= (MD(FNYR(YR), IM) + 1) THEN MN = IM
      NEXT IM
      DY = INT(Z(2) - MD(FNYR(YR), MN)): Z(2) = Z(2) - INT(Z(2))
      HR = INT(Z(2) * 24): Z(2) = Z(2) * 24 - HR
      MI = INT(Z(2) * 60): Z(2) = Z(2) * 60 - MI
      SC = INT(Z(2) * 60): RETURN
11800 REM*Вычисление ARCSIN(X)
      IF X = 1 THEN Y = 90: GOTO 11840
      IF X = -1 THEN Y = -90: GOTO 11840
      Y = RD * ATN(X / SQR(ABS(1 - X * X)))

```

```

11840 RETURN
11900 REM*Вычисление ARCCOS(X)
      IF X = 0 THEN Y = 90: GOTO 11940
      Y = RD * ATN(SQR(ABS(1 - X * X)) / X)
      IF X < 0 THEN Y = Y + 180
11940 RETURN
12000 REM*Вычисление ARCCTG(X)
      IF X = 0 THEN Y = 90: GOTO 12040
      Y = RD * ATN(1 / X)
      IF X < 0 THEN Y = Y + 180
12040 RETURN
12200 REM*Вычисление T2,L2,GM
      X = SIN(EA / RD) * SIN(IN / RD): GOSUB 11800: T2 = Y
      IF IN > 90 THEN S = 1: GOTO 12240
      S = -1
12240 IF FNRL(EA) > 180 THEN S = -S
      IF T2 = 90 OR T2 = -90 THEN L2 = 0: GOTO 12270
      X = COS(EA / RD) / COS(T2 / RD): GOSUB 11900
      L2 = FNRL(LA + S * Y)
12270 X = COS(T1 / RD) * COS(T2 / RD) * COS((L1 - L2) / RD)
      X = X + SIN(T1 / RD) * SIN(T2 / RD): GOSUB 11900: GM = Y
      RETURN
12400 REM*Вычисление AZ
      L2 = LS
      IF L2 < L1 / 2 THEN T2 = L2 / 1.8
      IF L2 > L1 / 2 AND L2 < L1 THEN T2 = 90 - (L2 - L1 / 2) / 2.1
      IF L2 > L1 AND L2 < 360 THEN T2 = (L2 - L1) / 2
      IF L1 <> L2 THEN GOTO 12440
      IF T2 < T1 THEN AZ = 180: GOTO 12520
      AZ = 0: GOTO 12520
12440 IF FNRL(L1 - L2) <> 180 THEN 12470
      IF T2 < -T1 THEN AZ = 180: GOTO 12520
      AZ = 0: GOTO 12520
12470 IF T2 = 90 THEN AZ = 0: GOTO 12520
      IF T2 = -90 THEN AZ = 180: GOTO 12520
X = COS(T1 / RD) * TAN(T2 / RD) - SIN(T1 / RD) * COS((L1 - L2) / RD)
      X = X / SIN((L1 - L2) / RD): GOSUB 12000: AZ = INT(Y)
      IF L2 < L1 / 2 AND AZ < 2 THEN AZ = AZ + 180
      IF L2 < L1 / 2 THEN AZ = AZ + 180

```

```

      IF L2 > L1 THEN AZ = AZ + 180
      IF AZ > 360 THEN AZ = AZ - 360
12520 RETURN
15000 REM*K новому спутнику
      RETURN 10100

```

В листинге 5.8 находятся строки программы, которые служат для вычисления времени входа и выхода спутников в зону и из зоны радиовидимости, а также для вывода всех необходимых данных на экран.

Листинг 5.8. Продолжение 7

```

16000 REM*Вычисление восходящих узлов орбит
      ON CM - 20 GOTO 16020
      GOTO 16360
16020 CLS
      PRINT "   Проход зоны видимости станции "; STAN$;
      PRINT "           спутником "; SP$
      PRINT "   ОРБ."; TAB(9); "Время узла"; TAB(25); "Узел";
      PRINT TAB(36); "Вход"; TAB(48); "Выход"; TAB(60); "Азим.";
      PRINT TAB(65); "<Enter>=продолж."
      PRINT "   N"; TAB(10); "чч мм cc"; TAB(25); "град"; TAB(36);
      PRINT "чч мм"; TAB(48); "чч мм"; TAB(60); "град"; TAB(68);
      PRINT "1 = выход";
      DD = 0
      Z0(2) = ET#
      Ниже строка задействована без дополнительной корректировки
16040 TD# = PS# * ND + PD# * ND * PS#
      Z(1) = Z0(1): Z(2) = Z0(2) + TD#: GOSUB 11600
      IF DD = DY THEN 16160
      IF YR < 70 THEN YR2 = 2000 + YR
      DD = DY
      PRINT TAB(20); "Дата: "; DY; "."; MN; "."; YR2; "г"
      Ниже строка задействована без дополнительной корректировки
16160 LS = FNRL(GS# * ND + L0 + GS# * ND * PD#)
      PRINT NR + ND; TAB(8); HR; TAB(12); MI; TAB(16); SC;
      PRINT TAB(24); INT(LS + .5);
      ZI = -1: Z0 = NF - 1: CZ = 0: X = 1 / R0: GOSUB 11900: G0 = Y
      GOSUB 12400

```

```

FOR I = 0 TO NF - 1
  LA = LS + (GS# + GS# * PD#) / NF * I: EA = 360 * I / NF: GOSUB 12200
  IF GM > G0 THEN GOTO 16250
  IF CZ = 0 THEN CZ = 1: ZI = I
  GOTO 16260
16250 IF CZ = 1 THEN CZ = 0: Z0 = I - 1
16260 NEXT I
  IF ZI = -1 THEN GOTO 16319
  Z(1) = Z0(1): Z(2) = Z0(2) + TD# + DT * ZI: GOSUB 11600
  PRINT TAB(34); HR; TAB(39); MI;
  Z(1) = Z0(1): Z(2) = Z0(2) + TD# + DT * Z0: GOSUB 11600
  PRINT TAB(48); HR; TAB(53); MI;
  PRINT TAB(60); AZ;
  GOTO 16320
16319 PRINT TAB(35); "Орбита за горизонтом";
16320 PRINT TAB(73);
  INPUT C: ON C GOTO 16360, 16350
16350 ND = ND + 1: GOTO 16040
16360 RETURN
  GOTO 16320
  REM* DY — день, MN — месяц, YR — год
  REM* HR — часы, MI — минуты, SC — секунды
  REM* L0 — долгота восх.узла, PN — коррекция долготы
  REM* PS — период, PD — коррекция, R0 — высота
19000 REM*Выход в DOS
  END

```

Приведенный выше листинг 5.8 можно дополнить. Например, можно очень просто сделать одновременный вывод информации и на экран и на принтер. Для этого в последнем продолжении следует после каждой строки с оператором PRINT дописать точно такую же строку с оператором LPRINT.

Советую вам посредством среды программирования Visual Basic и использования вышеприведенного листинга написать свою программу типа ORBITA для Windows.

Вспомогательная программа

Очень часто для выполнения какой-либо задачи требуется разработать специальную программу. Исходные коды основного файла одной из таких вспомогательных программ WSPOM_RS привожу в этом разделе книги. Де-

ло в том, что задействованный в этой программе математический аппарат полностью соответствует математическому аппарату программы ORBITA. Программа разработана в среде программирования Turbo C++ 3.

Вполне возможно, что кому-то из читателей захочется сделать свою программу для расчета элементов круговых орбит на языке C/C++, тогда он сможет воспользоваться кодами моей вспомогательной программы. Функции математического аппарата этой программы в своем наименовании содержат число, которое соответствует числовой метке точно такой же математической подпрограммы в ORBITA.

Например, фунуция `mat1800()` выполняет точно такие же действия, как и подпрограмма, начинающаяся с цифровой метки 11800 в программе ORBITA.

Начало файла `wspom_rs.cpp` располагается в листинге 5.9.

В начале листинга 5.9 проводится объявление подключаемых файлов, массива переменных с количеством дней до начала последующего месяца, объявление прочих переменных величин и задействованных в программе функций.

Листинг 5.9. Файл `wspom_rs.cpp`

```
/* Файл "wspom_RS" для отработки команд на Turbo C++ v.3 */
#include <iostream.h>
#include <time.h>
#include <conio.h>
#include <stdio.h>
#include <math.h>
#include <dos.h>
#define RD 57.295779
#define RM 6371.1
#define TC 0.0000308
#define UD 1.8E+07
int md[2][12] = { 0, 31, 60, 91, 121, 152, 182, 213, 244, 274, 305, 335,
                  0, 31, 59, 90, 120, 151, 181, 212, 243, 273, 304, 334;
int *mas;
double z02,zb2;
char *name[]={ "RS-10" };
char *cl_st[]={ "Ludinovo" };
int dy,mn,yr,ep;
double hr,mi,sc;
int zl,s,z01,zb1,nf,az;
```

```
long n0,nd;
double z2, ut, dt;
double in, l0, pn, ps, pd;
double r0, f1, td;
double t1, t2, l1, l2;
int fn_yr,zi;
double la,gm,ls,ea,dd,xx,yy,z0,g0,fn_rl;
void print(void);
void sp_dat(void);
void ep1200(void);
void ep1400(void);
void ep1600(void);
void mat1800(void);
void mat1900(void);
void mat2000(void);
void mat2200(void);
void mat2400(void);
void mat4180(void);
void mat6000(void);
void mat6040(void);
int fnyr(double);
double fnrl(double);
int c_break(void);
```

В листинге 5.10 описана главная функция `main()` и функция, организующая прекращения работы программы после нажатия сочетания клавиш `<Ctrl>+<Break>`.

Листинг 5.10. Продолжение 1

```
void main(void)
{
    int repeat;
        textbackground(1);
        textcolor(14);
        clrscr();
    ctrlbrk(c_break);
    sp_dat(); // считываются в память данные по спутнику
    ep1200(); // перевод календарного времени в эпохальное
```

```
z01 = z1;
z02 = z2;
do
{
    mat4180();
    mat6000();
    repeat = getchar();
    if(repeat == '\x1b') break;
}
while (repeat == 'Y' || repeat == 'y');
return;
}

int c_break(void)
{
printf("\n Нажато Ctrl-Break. Программа завершается.");
return(0);
}
```

В листинге 5.11 описываются функции математического аппарата и перевода реального времени в эпохальное и наоборот.

Листинг 5.11. Продолжение 2

```
// Перевод календарного времени в эпохальное
void ep1200(void)
{
    z1 = yr;
    fnyr(z1);
    z2 = md[fn_yr][mn-1] + dy + ((sc/60 + mi)/60 + hr)/24;
    return;
}

// перевод Z в EP
void ep1400(void)
{
    int i,yy,yz;
        if(ep > z1) s = 1;
        if(ep < z1) s = -1;
```



```

        if(ep == z1) s = 0;
        if(s == 0) return;
yz = (z1 + (s-1)/2);
yy = (ep - (s + 1)/2);
        if(yz < yy) goto wyh1;
for (i = yz; i >= yy; i-- )
{
    fnyr(i);
    z2 = z2 - (366 - fn_yr)*s;
}
z1 = ep;
return;
wyh1:
for(i = yz; i <= yy; i++)
{
    fnyr(i);
    z2 = z2 - (366 - fn_yr)*s;
}
z1 = ep;
return;
}

```

// перевод Z в календарное время

```
void ep1600(void)
```

```

{
    int im;
    double nn;
        fnyr(z1);
nn = ((z2 - 1)/(366 - fn_yr));
ep = z1 + floor(nn);
    ep1400();
    yr = z1;
    fnyr(yr);
    for (im = 1; im <= 12; im++)
    {
        if(z2 >= md[fn_yr][im-1] + 1) mn = im;
    }
    dy = floor(z2 - md[fn_yr][mn-1]);
    if(dy > 31 || dy < 0) dy = 31;
}

```

```
z2    =    z2 - floor(z2);
hr    =    floor(z2 * 24);
z2    =    z2 * 24 - hr;
mi    =    floor(z2 * 60);
z2    =    z2 * 60 - mi;
sc    =    floor(z2 * 60);
return;
}

// вычисление ARCSIN(X)
void mat1800(void)
{
    if(xx == 1) yy = 90; return;;
    if(xx == -1) yy = - 90; return;;
    yy = RD * atan(xx/sqrt(fabs(1-xx * xx)));
    return;
}

// вычисление ARCCOS(X)
void mat1900(void)
{
    if(xx == 0) yy = 90; return;;
    yy = RD * atan(sqrt(fabs(1 - xx * xx))/xx);
    if(xx < 0) yy = yy +180;
    return;
}

// вычисление ARCCTG(X)
void mat2000(void)
{
    if(xx == 0) yy = 90; return;;
    yy = RD * atan(1/xx);
    if (xx < 0) yy = yy +180;
    return;
}

// вычисление T2, L2, GM
void mat2200(void)
{
    static int ss;
```

```

xx = sin(ea/RD) * sin(in/RD);
    mat1800();
t2 = yy;
    if(in > 90)  ss =1; goto wyh4;;
ss = -1;
wyh4:
    fnrl(ea);
    if(fn_rl > 180) ss = ss * (-1);
    if(t2 == 90 || t2 == -90)  l2 = 0;  goto wyh5;;
xx = cos(ea/RD)/cos(t2/RD);
    mat1900();
l2 = fnrl(la + ss * yy);
l2 = fn_rl;
wyh5:
xx = cos(t1/RD) * cos(t2/RD) * cos((l1 - l2)/RD);
xx = xx + sin(t1/RD) * sin(t2/RD);
    mat1900();
gm = yy;
return;
    }

// вычисление AZ
void mat2400(void)
{
l2 = ls;
    if(l2 < l1/2) t2 = l2/1.8;
    if(l2 > l1/2 && l2 < l1) t2 = 90 - (l2 -l1/2)/2.1;
    if(l2 > l1 && l2 < 360) t2 = (l2-l1)/2;
    if(l1 != l2)  goto wyh6;
    if(t2 < t1) az = 180;return;;
az = 0; return;
wyh6:
    if(fnrl(l1 - l2) != 180) goto wyh7;
    if(t2 < -t1) az = 180; return;;
az = 0; return;
wyh7:
    if(t2 == 90) az = 0; return;;
    if(t2 == -90) az = 180;return;;
xx = cos(t1/RD) * tan(t2/RD) - sin(t1/RD) * cos((l1 - l2)/RD);

```

```

xx = xx/sin((l1 - l2)/RD);
    mat2000();
az = floor(yy);
    if(l2 < l1/2 && az < 2) az = az + 180;
    if(l2 < l1/2) az = az + 180;
    if(l2 > l1) az = az + 180;
    if(az > 360) az = az - 360;
return;
}

// задаются определенные данные, необходимые для расчетов
void mat4180(void)
{
    ep1200();
z01 = z1; z02 = z2;
ps    = ps/1440;
pd    = pd/1440;
r0    = (r0 + RM)/RM;
t1    = 53.85; l1 = 325.35;
l1    = fnrl(l1); l1 = fn_rl;
dy    = 1; mn = 10; yr = 95;
hr    = 00; mi = 00; sc = 00;
    ep1200();
ep = z01;
    ep1400();
zb1 = z1; zb2 = z2;
/* printf("Число zb1 = %d, число zb2 = %f, число z1 = %d, число
z02 = %f",zb1,zb2,z1,z02);
*/
nd = floor((zb2 - z02)/ps);
dt = (ps + pd * nd)/nf;
ut = (dt - TC) * UD;
// printf("Величина nd = %ld, Значение dt = %3.6f, Величина
ut = %3.6f",nd,dt,ut);
return;
}

```

В листинге 5.12 описана функция, выполняющая вывод на экран всей необходимой информации. Сначала выводятся строки заголовка, затем строки с информацией.

Листинг 5.12. Продолжение 3

```
//вычисление восходящих узлов орбит
void mat6000(void)
{
int i, letter;
static int cz;
printf("\nПроход спутником %s зоны видимости станции %s",name[0],
cl_st[0]);
printf("\nОпб.   Время узла   Узел   Вход   Выход   Азим ");
printf("\n No     чч мм cc     град.   чч мм   чч мм   град.");
whod30:   dd = 0;
whod40:   td=ps * nd + pd * nd * (nd -1)/2;
z1 = z01;
z2 = z02 + td;
ep1600();
if(dd == dy) goto whod160;
dd = dy;
cout <<"\n    Дата: " << dy << " - " << mn << " - " << yr;
whod160:   ls = fnrl(pn * td +l0);  ls = fn_rl;
if(ls < 0) ls = 360 + ls;
printf("\n%6ld%4.0f%3.0f%3.0f%7.0f", (n0 + nd),hr,mi,sc,ls);
ls = fnrl(pn * td + l0);  ls = fn_rl;
zi = -1; z0 = nf -1; cz = 0; xx = 1/r0;
mat1900();
g0 = yy;
mat2400();
for(i = 0; i <= (nf-1); i++)
{
la = ls + pn * dt * i;
ea = 360 * i/nf;
mat2200();
//cout<<"\n+++ cz = "<< cz;
if(gm > g0) goto wyh250;
if(cz == 0) {cz = 1;  zi = i;};
goto wyh260;
//cout<<"... zi = " <<zi <<"  z0 = "<<z0 << "  cz = " << cz;
wyh250:   if(cz == 1)  cz = 0; z0 = i - 1;;
```

```

wyh260:
    }
    if(zi == -1) goto wyh319;
z1 = z01;
z2 = z02 + td + dt * zi;
// printf("\n,,,,, z2 = %f .... td = %f ... dt = %f .... zi =
%d,,,,, ", z2, td, dt, zi);
    ep1600();
printf("%6.0f %3.0f", hr, mi);
z1 = z01;
z2 = z02 + td + dt * z0;
// printf("\n**** z2 = %f *** td = %f *** dt = %f *** z0 = %d
**", z2, td, dt, z0);
    ep1600();
printf("%6.0f %3.0f %6d      ", hr, mi, az);
letter = getchar();
if(letter == 'x' || letter == 'X') return;
nd = nd + 1;
goto whod40;
wyh319: printf("      Орбита за горизонтом      ");
letter = getchar();
if(letter == 'x' || letter == 'X') return;
nd = nd + 1;
goto whod40;
wyh360: return;
    }

```

В листинге 5.13 описаны вспомогательные функции, необходимые для выполнения расчетов.

Листинг 5.13. Продолжение 4

```

// вспомогательная функция для определения високосного года
int fnyr(double x)
{
double y = 4.0;
double result;
result = fmod(x, y);
if(result == 0) goto wyh1;
fn_yr = 1;

```

```
return(fn_yr);
wyhl: fn_yr = 0; return(fn_yr);
    }

// вспомогательная функция
double fnrl(double l)
    {
double x;
//    if(l > 360) x = l -360;
//    x = l;
x = (l - 360 * floor(l/360));
/* printf("^^^^^ fn_rl = %d",x);    */
fn_rl = x;
return(fn_rl);
    }

// задает определенные кеплеровские данные
void sp_dat(void)
    {
n0    =    38700;
dy    =    27;
mn    =    9;
yr    =    95;
hr    =    0; mi = 46; sc = 13;
in    =    83.0; lo = 330.0;
pn    =    362.0498;
ps    =    104.98413;
pd    =    -0.0000001;
r0    =    1008.4;
nf    =    120;
return;
    }
```

Как сделана программа CW_QSO

Ниже приведены листинги файла `cw_qso.cpp`, созданного в среде программирования Turbo C++ 3. Этот файл с исходными кодами программы CW_QSO предлагается вам для рассмотрения и создания, на базе разработанных мною исходных кодов, своих собственных программ для радиосвязи телеграфом.

В начальной части листинга 5.14 проводится объявление подключаемых файлов, задействованных в программе функций и переменных величин. Кроме того, здесь же размещены и специальные таблицы, необходимые для кодирования символов при передаче и декодирования символов при приеме, а также некоторые из текстов, которые часто используются в программе.

Листинг 5.14. Файл cw_qso.cpp

```

/*****
/*  PROGRAMM_NAME                CW_QSO v.1.10   (c) RA3XB                */
/*****
/*  Программа для проведения CW (телеграфной) любительской */
/*      радиосвязи посредством компьютера IBM PC.                */
/*      Автор разработки — Тяпичев Геннадий А.  < RA3XB >          */
/*      Начало: май 1998г, продолжено в мае 2000 года              */
/*****
/*  Программа создавалась на Turbo C++ v.3                        */
/*****

#include <iostream.h>
#include <stdio.h>
#include <dos.h>
#include <stdlib.h>
#include <process.h>
#include <ctype.h>
#include <string.h>
#include <bios.h> #include <conio.h>
#include <sys/timeb.h>
#include <time.h>
#include "cw_qsol.h"
#include <alloc.h>
#define BORDER 1
#define ESC 27
#define REV_VID 0x70
#define NORM_VID 0x4a
#define MAX 250
#define FILENAME "qso_log.dat"
#define FILESPEC "cw_qso.spc"
#define FILEHLP "cw_qso.doc"
FILE *fp;
FILE *log_file;

```



```
FILE *init_file;
FILE *send_file;
FILE *save_file;
FILE *printer;
void bord_col(int num);
void draw_border(int startx, int starty, int endx, int endy);
void write_video(int x, int y, char *p, int attrib);
void save_video(int startx, int endx, int starty, int endy, unsigned int
*buffer);
void restore_video(int startx, int endx, int starty, int endy, unsigned int
*buffer);
void goto_xy(int x, int y), cls(int zwet), screen_col(int rez);
void wind_zw(int attr, int starx, int starty, int x, int y);
void zastawka(void), zastawka2(void);
void err_wind(char *txt, int x, int y, int endx, int endy);
void write_string(int x, int y, int attrib, char *p);
void write_char(int x, int y, char ch, int attrib);
void display_help(int x, int y, int count);
void zabitie(int x, int y, int attrib, int len);
void pered_simw(int ch);
void pauza(int count);
void zwuk(int count);
void perek_zw(void);
void peredacha(void);
void getxy(void);
void get_time(void); //int date_flag);
void read_string(int x, int y, int len, char *p);
void read_word(int x, int y, int len, char *p);
void scr_scrup(int attr, int x, int y, int xl, int yl);
void init_sys(), send_str(char *str);
void cls_rx(void), cls_tx();
void copy_buf(int count);
void promt(char *text, int x, int y, int endx, int endy);
void send_buf0(int x, int y, int len, int attrib);
void send_buf1(int x, int y, int len, int attrib);
void co_buf(int x, int y, int len, int count, int attrib);
void write_simw(int ch, int attrib);
void wyzow(void), wyzow1(void), wyzow2(void);
void wyhod(void), wyhod2(void), wyhod3(void);
```

```
void alt_f2(), alt_f3(), alt_f4(), alt_f5();
void alt_f6(),alt_f7(),alt_f8(),alt_f9(),alt_f10();
void copy_buf_fl(int x);
void cls_buf1(int len),cls_buf0(int len);void mouse(void);
int init_str(char *str, int len),exec_cmd(int ch);
void izm_skor_p(void),izm_ton(void),izm_tonl(void);
void uwel_skor(void), umen_skor(void);
void addqso(void),nomer_qso(void),getslots(void),wlist(void);
void ctrl_msg(void);
void ct_f5(void),ct_f6(void),ct_f7(void),ct_f8(void),ct_f9(void);
void ust_dtr(void),ust_rts(void),ust_dtr_rts(void),sbros_dtr_rts(void);
void ProwSBl(void);
void NullRegistr(void);
void SetKanal(void);
void SetRegistr(void);
void SBlRegSet(int,int);
int inport(void);
void priem(void);
void perek1(void);
void rx_read(void);
void izm_pr(void),izm_prl(void),izm_skor_priema(void);
void rus_lat(void);
void djoist(void);
void dis_help(void);
void cursor_ubrat(void);
void t_wysok(void), t_nizk(void);
locate();

struct {
    char frequency[5];
    char mode[6];
    char tim[8];
    char date[12];
    char call_sign[9];
    char rst_sent[5];
    char rst_rcvd[5];
    char name_k[10];
    char qth[13];
    char comments[65];
```

```
    char contest[17];
    int number;
} qso;

struct CD
{
    char name[9];
    char diap[5];
    char date[12];
    char tim[8];
    char rs1[5];
    char rs2[5];
    char namkor[10];
    char qth[13];
    char mod[6];
    char category[17];
    char infor[65];
    int number;
} disc;

unsigned char simwol;
short int skorost;
short int skorost2;
int znak;
int flag_zw = 1;
unsigned int xx1,yy1;
unsigned int xx0=3,yy0=0;
unsigned int ton = 1000;
unsigned char far *vid_mem;
int slots[MAX];
int prt,prtl,i;
int flag_rus = 0;
unsigned int simw_nom;           // номер символа в таблице
char simwol2;
unsigned int sch_pomeh = 5;      // данные счетчика помех
unsigned int sch_posylok = 8;    // данные счетчика посылок
char port[7];
char my_call[15];
```

```

char my_name[20];
char my_qth[25];
char my_log[80];
char save_fn[80];
char buf1[400];
char buf2[4000];
char buf0[80];
char temper[10];

int tabl[] =
{0x55,0x31,0x40,0x32,0x3f,0x2f,0x27,0x23,0x21,0x20,0x30,0x38,0x3c,0x3e,
 0x78,0x36,0x2a,0x45,0x28,0x4c,0xc5,0x05,0x18,0x1a,0x0c,0x02,0x12,0x0e,
 0x10,0x04,0x17,0x0d,0x14,0x07,0x06,0x0f,0x16,0x1d,0x0a,0x08,0x03,0x09,
 0x11,0x0b,0x19,0x1b,0x1c,0x6d,0x73,0x6d,0x7d,0x80,0x13,0x05,0x18,0x1a,
 0x0c,0x02,0x12,0x0e,0x10,0x04,0x17,0x0d,0x14,0x07,0x06,0x0f,0x16,0x1d,
 0x0a,0x08,0x03,0x09,0x11,0x0b,0x19,0x1b,0x1c,0x1f,0x24,0x15,0x1e,0x80};
int abc[] = {
238,160,161,230,164,165,228,163,229,168,169,170,171,172,173,174,
175,233,224,225,226,227,166,162,236,235,167,232,237,239,231,234,
158,128,129,150,132,133,148,131,149,136,137,138,139,140,141,142,
143,153,144,145,146,147,134,130,156,155,135,152,157,159,151,154,
96,65,66,67,68,69,70,71,72,73,74,75,76,77,78,79,80,
81,82,83,84,85,86,87,88,89,90,123,124,125,126,88,
96,97,98,99,100,101,102,103,104,105,106,107,108,109,110,111,
112,113,114,115,116,117,118,119,120,121,122,123,124,125,126,88};

char tabl[] ={
00,'E','T','I','A','N','M','S','U','R','W','D','K','G','O','H',
'V','F','ю','L','я','P','J','B','X','C','Y','Z','Q','ч','ш','5',
'4', 00,'3', 00, 00, 00,'2', 00, 00, 00, 00, 00, 00, 00,'1','6',
00,'/', 00, 00, 00, 00, 00,'7', 00, 00, 00,'8', 00,'9','0','>',
00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00,'?', 00, 00, 00, 00,
00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00,'-'};
char tab2[] ={
00,'e','т','и','а','н','м','с','у','р','в','д','к','г','о','х',
'ж','ф','ю','л','я','п','й','б','ь','ц','ы','з','щ','ч','ш','5',
'4', 00,'3', 00, 00, 00,'2', 00, 00, 00, 00, 00, 00, 00,'1','6',
00,'/', 00, 00, 00, 00, 00,'7', 00, 00, 00,'8', 00,'9','0','>',
00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00,'?', 00, 00, 00, 00,

```

```

00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00,
'-';
char *txt1[] = { "Нажмите <Enter>  " };
char *txt2[] = { "CW" };
char *txt3[] = { "* * * ПЕРЕДАЧА * * * " };
char *txt4[] = { "* * * ПРИЕМ * * * " };
char *txt5[] = {
" <Alt-X>-выход; <Alt-Fn>-тексты; <Home>-передача; <End>-прием;
<F1>-HELP";
char *txt7[] = { "Выходим в DOS ? (Y/N)  " };
char *txt8[] = {"CALL: "};
char *txt9[] = {"NAME: "};
char *txt10[] = {"QTH: "};
char *txt11[] = {"RST: "};
char *txt12[] = {"темпер.: "};

char *txt20[] = {"Введите ПОЗЫВНОЙ корреспондента"};
char *txt21[] = {"Введите ИМЯ корреспондента и RST"};
char *txt22[] = {"      Введите ИМЯ файла"};
char *txt23[] = {"НЕ ЗАБУДЬТЕ ввести данные по QSO!!!"}; int ch_k;
int ch_n;
int yy=0;
int xx=3;
int col;
int row;
int flag1 = 0;
int date_flag = 0;
int save = 0;
int print =0;
int border = 0;
int v_attribute;
int count,ch;
int old_x,old_y;
unsigned int nomer = 0;
char help,str;
unsigned int com_x;
int com_adr1 = 0x2fc;
int com_adr2 = 0x2fe;
int nn,pp;

```

```
int del1 = 2;  
int del2 = 10;
```

В листинге 5.15 размещена основная функция программы — `main()`. Начальная часть функции `main()` содержит вызовы функций для инициализации программы, вывод на экран рабочих заставок и задание определенных значений некоторым из переменных.

Листинг 5.15. Продолжение 1

```
main()                // Начало подпрограммы main()  
{  
    init_sys(); // инициализация программы  
    zastawka(); // выдает на экран первую заставку  
    zastawka2(); // выдает на экран рабочую заставку  
    skorost = 24; // назначается начальная величина скорости  
    skorost2 = skorost;  
    ton = 1000; // назначается начальный тон 1000 Гц  
    outportb(0x201,0x10); // открывает порт джойстика  
    izm_skor_p(); // устанавливает начальную скорость  
    izm_skor_priema();  
  
    zabitie(2,31,0x0,14);  
    write_string(2,31,0xf,*txt4);  
    if(xx1 < 3) xx1 = 4;  
    goto_xy(xx1-1,yy1);  
    cursor_ubrat();  
  
whod1:  
    get_time(); // выдает на экран текущее время  
    if(kbhit()) perek1(); // задействует функцию переключения  
    prt = inport(); // получить данные из порта для  
                    // проверки наличия сигнала  
    if(prt == 1) priem(); // если сигнал есть, перейти на прием  
    prt1 = inportb(0x201); // проверяем нажатие джойстика  
    if(prt1 == 239 || prt1 == 223) djoist(); // работа джойстика  
    goto whod1;  
}  
// конец подпрограммы main();
```

Описание функции `main()` окончилось. Далее идет описание всех остальных функций, задействованных в программе. В листинге 5.16 описываются следующие функции:

- ❑ работа порта джойстика в режиме полуавтоматического телеграфного ключа;
- ❑ большая функция всех переключений, задействованных в программе;
- ❑ функция инициализации программы, в которой проводится проверка наличия в каталоге всех необходимых файлов, устанавливается режим работы СОМ-порта и звуковой карты компьютера.

Листинг 5.16. Продолжение 2

```
// убрать курсор (погасить)
void cursor_ubrat(void)
{
    union REGS r;
    r.h.ah=1;
    r.h.ch=0x20;
    int86(0x10, &r, &r);
    return;
}

// джойстик — полуавтоматический телеграфный ключ
void djoist(void)
{
    whdl:
        if(prtl == 239) {zwuk(2); pauza(2); goto wyhl;}
        if(prtl == 223) {zwuk(7); pauza(2);}
    wyhl:
        prtl = inportb(0x201);
        if(prtl == 239 || prtl == 223) goto whdl;
        sbros_dtr_rts();
        return;
}

// Переключения, задействованные в программе
void perek1()
{
    ch_k = getch();
    if(ch_k == 0) { ch_k = getch() + 128;}
```

```

if(ch_k == 27) {sbros_dtr_rts();exit(1);} /* Esc – быстрый выход */
    if(ch_k == 192) goto wyh1; /* F6 – ввод CALL */
    if(ch_k == 193) goto wyh2; /* F7 – ввод NAME */
    if(ch_k == 194) goto wyh3; /* F8 – ввод QTH */
    if(ch_k == 195) goto wyh4; /* F9 – ввод RST */
    if(ch_k == 196) goto wyh5; /* F10 – ввод температуры */
    if(ch_k == 223) perek_zw(); /* Ctrl-F2 – переключение звука */
    if(ch_k == 223) cls_rx(); /* Ctrl-F3 – очистка окна приема*/
    if(ch_k == 224) cls_tx(); /* Ctrl-F4 – очистка окна передачи*/
    if(ch_k == 261) addqso(); /* F11 – ввести QSO в журнал */
    if(ch_k == 262) zastawka2(); /* F12 – обновление экрана */
    if(ch_k == 215) {wlist();zastawka2();}/* Shift-F4 – чит.журнал*/
    goto whod2;

wyh1: write_string(0,3,0x07,txt8[0]);
    zabitie(0,9,0x5b,16); goto whod2;
wyh2: write_string(1,3,0x07,txt9[0]);
    zabitie(1,9,0x5b,16); goto whod2;
wyh3: write_string(2,4,0x07,txt10[0]);
    zabitie(2,9,0x5b,16); goto whod2;
wyh4: write_string(0,60,0x07,txt11[0]);
    zabitie(0,65,0x5b,14); goto whod2;
wyh5: write_string(1,56,0x07,txt12[0]);
    zabitie(1,65,0x5b,8); goto whod2;

whod2:

    if(ch_k == 192) goto wyh10; //F6 – ввод CALL
    if(ch_k == 193) goto wyh11; //F7 – ввод NAME
    if(ch_k == 194) goto wyh12; //F8 – ввод QTH
    if(ch_k == 195) goto wyh13; //F9 – ввод RST
    if(ch_k == 196) goto wyh14; //F10 – ввод температуры
    exec_cmd(ch_k);
    return;

wyh10:
    if(flag1 == 0) {xx = 0; yy =9; flag1 = 1;}
    ch_n = getch(); write_char(xx,yy,ch_n,0x5b);
    if(ch_n == 13 || ch_n == 3) {
        write_string(0,3,0x4e,txt8[0]); flag1 = 0;

```



```
    read_word(0,9,6,qso.call_sign);
    locate();
    return;}
    yy++;
    goto whod2;
wyh11:
    if(flag1 == 0) {xx = 1; yy =9; flag1 = 1;}
    ch_n = getch(); write_char(xx,yy,ch_n,0x5b);
    if(ch_n == 13 || ch_n == 3) {
        write_string(1,3,0x4e,txt9[0]); flag1 = 0;
        read_word(1,9,8,qso.name_k); return;}
    yy++;    goto whod2;

wyh12:
    if(flag1 == 0) {xx = 2; yy =9; flag1 = 1;}
    ch_n = getch(); write_char(xx,yy,ch_n,0x5b);
    if(ch_n == 13 || ch_n == 3) {
        write_string(2,4,0x4e,txt10[0]); flag1 = 0;
        read_word(2,9,8,qso.qth); return;}
    yy++;    goto whod2;

wyh13:
    if(flag1 == 0) {xx = 0; yy =65; flag1 = 1;}
    ch_n = getch(); write_char(xx,yy,ch_n,0x5b);
    if(ch_n == 13 || ch_n == 3) {
        write_string(0,60,0x4e,txt11[0]); flag1 = 0;
        read_word(0,65,5,qso.rst_sent); return;}
    yy++;    goto whod2;

wyh14:
    if(flag1 == 0) {xx = 1; yy =65; flag1 = 1;}
    ch_n = getch(); write_char(xx,yy,ch_n,0x5b);
    if(ch_n == 13 || ch_n == 3) {
        write_string(1,56,0x4e,txt12[0]); flag1 = 0;
        read_word(1,65,6,temper); return;}

    yy++;    goto whod2;

wyh6:
    err_wind(txt7[0],19,10,23,69);
```

```
        if(ch_k == 121 || ch_k == 89) {
            sbros_dtr_rts(); exit(1);        }
        else
            goto_xy(18,0);    return;
wyh7:
    return;
}

// Инициализация программы
void init_sys()
{
    char ch, *err;
    init_file = fopen("CW_QSO1.CFG", "r");
    if (init_file == 0) {
        printf("НЕ открывается файл 'CW_QSO1.CFG'\n");
        printf("\nСоздайте текстовым редактором следующее:\n");
        printf("    COM – порт\n");
        printf("    позывной своей р/станции\n");
        printf("    свое имя\n");
        printf("    свой город (QTH)\n");
        printf("    имя Log файла и путь к нему\n");
        printf("    имя Save файла и путь к нему\n");
        getchar();
        exit(1);
    }
    else {
// чтение строк конфигурационного файла
        init_str(port, 7);
        init_str(my_call, 15);
        init_str(my_name, 20);
        init_str(my_qth, 25);
        init_str(my_log, 80);
        init_str(save_fn, 80);
        init_str(temper,10);
        init_str(qso.call_sign,10);
        init_str(qso.name_k,15);
        init_str(qso.qth,20);
        init_str(qso.rst_sent,6);
    }
    fclose(init_file);
```

```

    if(strcmp(port,"COM1")==0) {com_x = 0; com_adr1=0x3fc;
com_adr2=0x3fe;}// goto wyh;}
    if(strcmp(port,"COM2")==0) {com_x = 1; com_adr1=0x2fc;
com_adr2=0x2fe;}//goto wyh;}
    if(strcmp(port,"COM3")==0) {com_x = 2; com_adr1=0x3ec;
com_adr2=0x3ee;}// goto wyh;}
    if(strcmp(port,"COM4")==0) {com_x = 3; com_adr1=0x2ec;
com_adr2=0x2ee;}//goto wyh;}
    _bios_serialcom(_COM_INIT,com_x,_COM_1200|_COM_CHR8|_COM_STOP1);
// Инициализация компьютерной звуковой карты
ProwSBl();
NullRegistr();
SetKanal();
SetRegistr();
SBlRegSet(0xb0, 0x11);

    nomer_qso();
}
// Проверка наличия конфигурационного файла
int init_str(char *str, int len)
{
    char *err;
    err = fgets(str, len, init_file);
    if (err == NULL) {
        printf("Ошибка чтения файла CW_QSO1.CFG\n");
        exit(1);
    }
    str[strlen(str) - 1] = 0;
return(0);
}

```

В листинге 5.17 располагается функция приема символов. Эта функция использует при декодировании принятых сигналов определенные константы, которые оператор назначает сам при изменении скорости приема. Каждая такая константа действует в некотором диапазоне частот, поэтому оператор должен сам примерно определять, с какой скоростью поступают сигналы от корреспондента. В программах зарубежных разработчиков используется другой метод — метод автоматической настройки программы на скорость поступающих сигналов. Этот метод мне не нравится тем, что программа, работающая по методу автоматической настройки, при этой самой автоматической перестройке теряет как минимум один символ. Поскольку такая перестройка происходит после каждой паузы, то символов теряется очень много.

В используемом мною варианте никаких потерь при перестройке не происходит, т. к. никакой перестройки не проводится. Программа сразу начинает прием любого символа после любой паузы.

Листинг 5.17. Продолжение 3

```
// подпрограмма приема символа
void priem(void)
{
    unsigned int ee, hh;
    unsigned int simw_nom;
    n1:  simw_nom = 0;
        ee=sch_pomeh;
    s1:  ee--;
        if(ee == 0) goto v1; // сигнал прошел проверку на имп. помеху
    delay(del1);
        prt = inport();          // получить данные из порта и
                                // проверить на наличие сигнала
        if(prt == 1) goto s1; // если сигнал есть, продолжить проверку
                                // сигнала, не является ли он помехой
        goto wyh0;  // на выход из подпрограммы, т. к. сигнал
                    // был помехой
        // если сигнал не помеха, он проходит далее для анализа
    v1:  hh = sch_posylok;  // включить счетчик для проверки сигнала
    val: hh--;
    if(hh == 0 ) goto l1; // сигнал длительнее точки, анализировать далее
    delay(del2);
        prt = inport();          // получить данные из порта
                                // проверить на наличие сигнала
        if(prt == 1) goto val; // продолжить проверку
        simw_nom++;              // сигнал кончился досрочно, это точка
        goto k1;                 // проверить длительность паузы после точки

    l1:  hh = sch_posylok;  // провести доп.анализ длительного сигнала
    lal: hh--;
        if(hh == 0) goto z1; // сигнал длиннее типе
    delay(del2);
        prt = inport();
        if(prt == 1) goto lal; // сигнал присутствует
```

```

z1:
zal: prt = inport();
    if(prt == 1) goto zal; // дождаться окончания длинного сигнала
    simw_nom= simw_nom+2;   // сигнал кончился, похоже, что это
                           // типе
k1:  hh = sch_posylok;      // начало проверки паузы
    ee = sch_pomeh;
q1:  ee--;
    if(ee == 0) goto r1; // пауза прошла проверку на помеху
delay(del1);
    prt = inport();
    if(prt != 1) goto q1; // пауза продолжается
    goto tt;              // пауза есть интервал между посылками
r1:  hh--;
    if(hh == 0) goto pal; // пауза пошла на дальнейшую проверку
delay(del2);
    prt = inport();
    if(prt != 1) goto r1;
    goto tt;
pal: hh = sch_posylok;      // проверка паузы, не является ли она
                           // интервалом между буквами
p1:  hh--;
    if(hh == 0) goto m0; // пауза есть интервал между буквами, выход
delay(del2);
    prt = inport();
    if(prt != 1) goto p1; // пауза продолжается
    goto m2;              // пауза кончилась, на выход

tt:  simw_nom = simw_nom * 2; // продолжаем дальше исследовать символ,
    goto v1;

m2:  if(flag_rus == 1) simwol2 = tab2[simw_nom];
    else simwol2 = tab1[simw_nom];
    rx_read();
    hh = sch_posylok;      // провести доп.анализ паузы
la8: hh--;
    if(hh == 0) goto m7; // пауза длинная – выдать пробел
delay(del2);
    prt = inport();

```

```

        if(prt != 1) goto la8; // сигнала нет
        goto n1;
m0:
    if(flag_rus == 1) simwol2 = tab2[simw_nom];
        else simwol2 = tab1[simw_nom];
        rx_read();
        hh = sch_posylok;      // провести доп.анализ паузы
la7:   hh--;
        if(hh == 0) goto m7; // пауза длинная – выдать пробел
        delay(del2);
        prt = inport();
        if(prt != 1) goto la7; // сигнала нет
        goto n1;              // сигнал появился

t1:   simwol2 = 0x2e;
        rx_read();
        goto n1;

m7:   simwol2 = 0x20;
        rx_read();
wyh0: return;
    }

// подпрограмма проверки наличия сигнала от приемника
int inport(void)
{
    int ch; asm{ mov dx, com_adr2
        in al, dx
        and al,00010000b
        jz wyh1
        jnz wyh2
    }
wyh1: prt = 0; t_nizk(); return prt;
wyh2: prt = 1; t_wysok();
    return prt;
}

// ТОН ВЫСОКИЙ
void t_wysok(void)

```

```
    {
goto_xy(0,32); putchar('o');
goto_xy(0,49); putchar(' ');
return;
    }

// тон низкий
void t_nizk(void)
{
goto_xy(0,32); putchar(' ');
goto_xy(0,49); putchar('o');
return;
}
```

В листинге 5.18 размещены функции заставок. Эти функции не представляют интереса, но я не стал их убирать, чтобы дать один из возможных вариантов оформления. На мой взгляд, не самый лучший. У вас должно получиться лучше.

Листинг 5.18. Продолжение 4

```
// заставка начальная
void zastawka(void)
{
    int i;
    scren_col(3);
    cls(0x0e);
    draw_border(0,1,23,78);
    wind_zw(0x70,2,5,7,74);
    wind_zw(0x1e,8,5,14,74);
    wind_zw(0xc0,15,5,21,74);
    wind_zw(0x5e,5,13,17,66);
    draw_border(5,14,17,65);
    goto_xy(7,30);
    printf(" П Р О Г Р А М М А ");
    goto_xy(9,32);
    printf(" CW_QSO v.1.10 ");
    goto_xy(11,20);
    printf("для проведения любительской радиосвязи");
```

```

goto_xy(12,25);
printf("посредством компьютера IBM PC");
goto_xy(14,18);
printf("Разработана Тяпичевым Геннадием А. (с) RA3XB");
goto_xy(16,32);
printf("ноябрь 2000 года");
goto_xy(24,32);
printf("Нажмите <Enter> ");
getch();
}

// заставка рабочая – рабочий экран
void zastawka2(void)
{
    cls(0x07);
    wind_zw(0x6e,0,0,2,7);
    goto_xy(0,3); puts("CALL:");
    goto_xy(1,3); puts("NAME:");
    goto_xy(2,4); puts("QTH:");
    wind_zw(0x6e,0,56,2,63);
    goto_xy(0,60); puts("RST:");
    goto_xy(1,56); puts("темпер.:");
    goto_xy(2,56); puts("скорость");
    wind_zw(0x5a,0,8,2,25);
    wind_zw(0x5a,0,64,2,79);
    wind_zw(0x5a,17,0,15,79); goto_xy(17,2);
    puts("Принтер: ОТКЛ      *** Автор – RA3XB, 2000г ***
    Файл сохранения: ОТКЛ");
    write_video(0,39,*txt2,0xc);          /* надпись CW */
    write_video(2,31,*txt4,0xf);          /* надпись ПРИЕМ */
    wind_zw(0x5b,24,1,24,78); goto_xy(24,1);
    write_video(24,1,*txt5,0x5b);
    wind_zw(0x1e,18,0,23,79);
    wind_zw(0x1d,3,0,16,79); goto_xy(16,0);
    goto_xy(1,74);
    printf("%dГц",ton);
    izm_skor_p();
    izm_skor_priema();
    goto_xy(0,9); printf("%s",qso.call_sign);

```



```
goto_xy(1,9); printf("%s",qso.name_k);  
goto_xy(2,9); printf("%s",qso.qth);  
goto_xy(0,65); printf("%s",qso.rst_sent);  
goto_xy(1,65); printf("%s",temper);  
return;  
}
```

В листинге 5.19 приведены различные вспомогательные функции для работы с выводом текста на экран, для работы с цветом. Здесь также размещены функции, организующие вывод на экран вспомогательных информационных окон.

Листинг 5.19. Продолжение 5

```
/* ввести цвет бордюра */  
void bord_col(int num)  
{  
    union REGS r;  
    r.h.ah=0x0b;  
    r.h.bh=0;  
    r.h.bl=num;  
    int86(0x10,&r,&r);  
    return;  
}  
  
/* установка курсора */  
void goto_xy(int x,int y)  
{  
    union REGS r;  
    r.h.ah=2; /* функция установки курсора */  
    r.h.dl=y; /* координата колонки */  
    r.h.dh=x; /* координата строки */  
    r.h.bh=0; /* видеостраница */  
    int86(0x10,&r,&r);  
    return;  
}  
  
/* нарисовать обрамление окна — цветную рамку */  
void draw_border(int startx,int starty,int endx,int endy)  
{  
    register int i;
```

```

        for(i=startx;i<endx;i++) {
goto_xy(i,starty);
putchar(186);
goto_xy(i,endy);
putchar(186);
}
        for(i=starty;i<endy;i++) {
goto_xy(startx,i);
putchar(205);
goto_xy(endx,i);
putchar(205);
}
goto_xy(startx,starty); putchar(201);
goto_xy(startx,endy ); putchar(187);
goto_xy(endx ,starty); putchar(200);
goto_xy(endx ,endy ); putchar(188);
return;
}

/* вывод строки с определенным атрибутом */
void write_video(int x,int y,char *p,int attrib)
{
union REGS r;
register int i,j;
for(i=y; *p; i++) {
goto_xy(x,i);
        r.h.ah=9;           /* функция записи символа */
        r.h.bh=0;           /* видеостраница */
        r.x.cx=1;           /* число повторений символа */
        r.h.al=*p++;        /* символ */
        r.h.bl=attrib;      /* атрибут */
        int86(0x10,&r,&r);
}
return;
}

/* вывод одного символа на экран в цвете */
void write_simw(int ch,int attrib)
{

```

```

union REGS r;
    r.h.ah=9;           /* функция вывода символа */
    r.h.bh=0;           /* видеостраница */
    r.x.cx=1;           /* число повторений символа */
    r.h.al=ch;          /* символ */
    r.h.bl=attrib;      /* атрибут */
    int86(0x10,&r,&r);
}

/* сохранение части экрана */
void save_video(int startx,int endx,int starty,int endy,unsigned int
*buffer)
{
    union REGS r;
    register int i,j;
    for(i=starty;i<endy;i++)
        for(j=startx;j<endx;j++) {
            goto_xy(j,i);
            r.h.ah=8; /* функция чтения символа */
            r.h.bh=0; /* видеостраница */
            *buffer++ = int86(0x10,&r,&r);
            putchar(' '); /* очистка экрана */
        }
    return;
}

/* восстановление части экрана */
void restore_video(int startx,int endx,int starty,int endy,unsigned int
*buffer)
{
    union REGS r;
    register int i,j;
    for(i=starty;i<endy;i++)
        for(j=startx;j<endx;j++)
        {
            goto_xy(j,i);
            r.h.ah=9; /* функция записи символа */
            r.h.bh=0; /* видеостраница */
            r.x.cx=1; /* число повторений символа */

```

```
        r.h.al=*buffer++;          /* СИМВОЛ      */
        r.h.bl=0x1d;
        int86(0x10,&r,&r);
    }
    return;
}

/* очистка экрана */
void cls(int zwet)
{
    union REGS r;
    r.h.ah=6; /* код прокрутки экрана */
    r.h.al=0; /* код очистки экрана */
    r.h.ch=0; /* начальная строка */
    r.h.cl=0; /* начальная колонка */
    r.h.dh=24; /* конечная строка */
    r.h.dl=79; /* конечная колонка */
    r.h.bh=zwet; /* фон и цвет букв */
    int86(0x10,&r,&r);
    return;
}

/* Установка цветового режима экрана */
void scren_col(int rez)
{
    union REGS r;
    r.h.ah=0; /*функция установки режима*/
    r.h.al=rez; /*1 или 3 или другой режим экрана*/
    int86(0x10,&r,&r);
    return;
}

/* Установка активной видеостраницы */
void stran(int num)
{
    union REGS r;
    r.h.ah=5; /*функция выбора страницы*/
    r.h.al=num; /*номер выбираемой страницы*/
}
```

```

        int86(0x10,&r,&r);
        return;
    }

    /*          Задать окно (цветное)          */
void wind_zw(int attr,int startx,int starty,int x,int y)
    {
        union REGS r;          r.h.ah=6;
        r.h.al=0;
        r.h.bh=attr; /*цвет фона и букв*/
        r.h.ch=startx;
        r.h.cl=starty;
        r.h.dh=x;
        r.h.dl=y;
        int86(0x10,&r,&r);
        return;
    }

    /* Вывести на экран строку с дополнительными атрибутами */
void write_string(int x, int y, int attrib,char *p)
    {
        register int i;
        unsigned char far *v;
        vid_mem =(unsigned char far *)0xb8000000;
        v = vid_mem;
        v += (x*160) + y*2; /* вычислить адрес */
        for(i=y;i<=(y+strlen(p) + 2);i++) {
            *v++ = *p++; /* вывести символ */
            *v++ = attrib;
        }
        return;
    }

    /* прочитать строку из видеопамати в буфер */
void read_string(int x,int y,int len, char *p)
    {
        register int i;
        unsigned char far *v;
        vid_mem =(unsigned char far *)0xb8000000;

```

```

        v = vid_mem;
        v += (x*160) + y*2; /* вычислить адрес */
        for(i=y;i<=(y+strlen(p) + len);i++) {
            *p++ = *v++; /* вывести символ из памяти в строку*/
            *v++;
        }
    return;
}

/* прочитав слово из видеопамати в буфер */
void read_word(int x,int y,int len,char *p)
{
    register int i;
    unsigned char far *v;
    vid_mem=(unsigned char far *)0xb8000000;
    v = vid_mem;
    v += (x*160) + y*2; /* вычислить адрес */
    for(i=y;i<=(y+strlen(p) + len);i++) {
        *p++ = *v++; /* вывести символ из памяти в строку*/
        *v++;
    }
    return;
}

// Заполняет строку нулями
void zabitie(int x, int y, int attrib,int len)
{
    register int i;          unsigned char far *v;
    vid_mem=(unsigned char far *)0xb8000000;
    v = vid_mem;
    v += (x*160) + y*2; /* вычислить адрес */
    for(i=y;i<=(y+len);i++) {
        *v++ = 0; /* вывести символ */
        *v++ = attrib;
    }
    return;
}

/* Вывести символы с определенными атрибутами */
void write_char(int xx, int yy, char ch, int attrib)

```

```

    {
        register int i;
        unsigned char far *v;
        vid_mem=(unsigned char far *)0xb8000000;
        ch = ch_n;
        v =vid_mem;
        v += (xx*160) + yy*2;
        if(ch == 13 || ch == 3) return;
        *v++ = ch; /* вывести символ */
        *v++ = attrib; /* вывести атрибуты */
        yy++;
        return;
    }

// Создать окно для подтверждения выхода в DOS
void err_wind(char *txt,int x,int y,int endx,int endy)
{
    unsigned int *p;
    /* размещение памяти для видеобуфера */
    p=(unsigned int*)malloc((endx-x+1)*(endy-y+1));      if(!p) exit(1);
/* Вы можете здесь сами обработать ошибку */
    /* сохранение части экрана */
    save_video(x-1,endx,y-1,endy+1,p);
    wind_zw(0x4a,x-1,y-1,endx-1,endy);
    draw_border(x-1,y-1,endx-1,endy);
    /* высвечивание меню на своем месте */
    goto_xy(x+1,y+20);
    printf("%s",txt7[0]);
    ch_k = getch();
    /* восстановление части экрана */
    restore_video(x-1,endx,y-1,endy+1,p);
    free(p);
    return;
}

```

В листинге 5.20 располагаются функции, организующие передачу символа кодом Морзе.

Функция передачи берет очередной символ из ячейки `simwol` и дает ему название `ch_i`. Далее выполняется исследование символа `ch_i`. Если код сим-

вола меньше 127 и больше 44, то символ поступает для последующей обработки на часть функции, начинающуюся с метки `whod1`, где выполняется передача символа с использованием таблиц для кодирования.

Листинг 5.20. Продолжение 6

```
// подпрограмма передачи кода Морзе
void pered_simw(int simwol)
{
    int count;
    register int dd,bb;
    register int ch_z;
    int i;
    int j;
    int ch_i = simwol;
    const int AA = 44;
    if(ch_i == 0x20) pauza(8);
    if(ch_i < 44 ) return;
    if(ch_i < 127 ) goto whod1;
    for(j=0;j<=64;j++)

    { bb = abc[0+j];
      if(bb == ch_i) {
        ch_i = abc[j+64]; break;}
      }
whod1:
    i = ch_i - AA;
    ch_z = tabl[i];
    dd = 8;
kt:   ch_z = ch_z<<1;
    dd--;
    bb = ch_z & 0x100;
    if(dd == 0) goto wyh;
    if(bb == 0) goto kt;
g:    ch_z = ch_z<<1;
    bb = ch_z & 0x100;
    if(bb == 0) {count = 2; zwuk(count); goto k2;}
    count = 7;
    zwuk(count);
```



```
k2:      count = 2;
        pauza(count);
        dd--;
        if(dd == 0) goto wyh1;
        goto g;
wyh1: count = 4;
        pauza(count);
        wyh:
        return;
    }

// пауза между посылками
void pauza(int count)
{
    int j;
    for(j=0; j<=count;j++) {
        delay(skorost);
    };
    return;
}

// звуковая посылка
void zwuk(int count)
{
    int j;
    if(flag_zw == 1) sound(ton);
    else nosound();
    SBlRegSet(0xb0, 0x39); // 1200 Гц
    ust_dtr_rts();
    for(j=0;j<=count;j++) {
        delay(skorost);
    };
    nosound();    sbros_dtr_rts();        // ust_rts();
    SBlRegSet(0xb0, 0x11);
    return;
}

// переключ звука
void perek_zw(void)
```

```
{
if(flag_zw == 1) {
flag_zw = 0; return;
}
flag_zw = 1;
}
```

В листинге 5.21 располагаются описания различных вспомогательных функций, которые используются при работе на передачу, при выдаче на экран текущего времени и др. Много разных функций, но все они очень простые.

Листинг 5.21. Продолжение 7

```
// п/программа передачи
void peredacha(void)
{
write_video(2,31,*txt3,0xf);
bord_col(0x3);
goto_xy(18,1);
wyzow();
        sbros_dtr_rts();        write_video(2,31,*txt4,0xf);
bord_col(0);
}

// подпрограмма выхода из передачи (концовка)
void wyhod3(void)
{
write_video(2,31,*txt3,0xf);
bord_col(0x3);
goto_xy(18,1);
wyhod();
        sbros_dtr_rts();
write_video(2,31,*txt4,0xf);
bord_col(0);
}

// чтение позиции курсора
void getxy()
{
int i;
```

```

union REGS regs;
regs.h.ah = 3; // читает поз. курсора
regs.h.bh = 0;
int86(0x10, &regs, &regs);
i = regs.x.dx;
yy1 = 1 + (i & 0xff);
xx1 = 1 + (i >> 8);
return;
}

```

// выводит текущее время на экран в определенное место

```

void get_time(void)
{
    struct tm *time_now;
    time_t secs_now;
    char str[80];
    tzset();
    time(&secs_now);
    time_now = localtime(&secs_now);
    strftime(str, 80, "%H:%M %d-%m-%Y", time_now);
    goto_xy(1, 27);
    puts(str); goto_xy(3, 0);
}

```

/* Скроллинг в окне */

```

void scr_scrup(int attr, int frow, int fcol, int trow, int tcol)
{
    union REGS regs;
    regs.h.ah = 6;
    regs.h.al = 1;
    regs.h.bh = attr; /* обычные видеоатрибуты */
    regs.h.ch = frow;
    regs.h.cl = fcol;
    regs.h.dh = trow;
    regs.h.dl = tcol;
    int86(0x10, &regs, &regs);
}

```

// Очистить окно приема

```

void cls_rx()

```

```
{
    wind_zw(0x1d,3,0,16,79);
    goto_xy(16,0);
}

// Очистить окно передачи
void cls_tx()
{
    wind_zw(0x1e,18,0,23,79);
    goto_xy(23,0);
}

/* Копирование данных из файла в буфер buf1[] */
void copy_buf(int count)
{
    int i,letter;
    if(count ==1) goto wyh1;
    if(count ==2) goto wyh2;
    if(count ==3) goto wyh3;
    if(count ==4) goto wyh4;
    if(count ==5) goto wyh5;
    if(count ==6) goto wyh6;
    if(count ==7) goto wyh7;
    if(count ==8) goto wyh8;
    if(count ==9) goto wyh9;
    if(count ==10) goto wyh10;
    if(count ==11) goto wyh11;
wyh1:    if((fp = fopen("eqp.msg","r")) == NULL)
        {
            printf("Невозможно открыть файл eqp.msg");
            getchar();
            exit(0);
        }
        goto wyh20;
wyh2:    if((fp = fopen("rpt.msg","r")) == NULL)
        {
            printf("Невозможно открыть файл rpt.msg");
            getchar();
```

```
        exit(0);
    }
    goto wyh20;
wyh3:   if((fp = fopen("my_name.msg", "r")) == NULL)
    {
        printf("Невозможно открыть файл my_name.msg");
        getchar();
        exit(0);
    }
    goto wyh20;
wyh4:   if((fp = fopen("konec.msg", "r")) == NULL)
    {
        printf("Невозможно открыть файл konec.msg");
        getchar();
        exit(0);
    }
    goto wyh20;
wyh5:   if((fp = fopen("qrz.msg", "r")) == NULL)
    {
        printf("Невозможно открыть файл qrz.msg");
        getchar();
        exit(0);
    }
    goto wyh20;
wyh6:   if((fp = fopen("cq.msg", "r")) == NULL)
    {
        printf("Невозможно открыть файл cq.msg");
        getchar();
        exit(0);
    }
    goto wyh20;
wyh7:   if((fp = fopen("f5.msg", "r")) == NULL)
    {
        printf("Невозможно открыть файл f5.msg");
        getchar();
        exit(0);
    }
    goto wyh20;
wyh8:   if((fp = fopen("f6.msg", "r")) == NULL)
```

```
    {
        printf("Невозможно открыть файл f6.msg");
        getchar();
        exit(0);
    }
    goto wyh20;
wyh9:  if((fp = fopen("f7.msg","r")) == NULL)
    {
        printf("Невозможно открыть файл f7.msg");
        getchar();
        exit(0);
    }
    goto wyh20;
wyh10: if((fp = fopen("f8.msg","r")) == NULL)
    {
        printf("Невозможно открыть файл f8.msg");
        getchar();
        exit(0);
    }
    goto wyh20;
wyh11: if((fp = fopen("f9.msg","r")) == NULL)
    {
        printf("Невозможно открыть файл f9.msg");
        getchar();
        exit(0);
    }
wyh20: for(i=0;i<400;i++)
    {
        if((letter = fgetc(fp)) != EOF)
        {
            buf1[i] = letter;
        }
    }
fclose(fp);
return;
}
```

```
// выдать текст из буфера buf0 на передачу
void send_buf0(int x,int y,int len,int attrib)
```

```

{
    int i;
    unsigned int ch;
    xx = x;
    yy = y;
    for(i=0;i<len;i++)
    {
        ch = buf0[i];
        if(ch == NULL) break;
        if(ch < 32)
        {
            if(ch == 0x0A | ch == 0x0D) {xx++; yy = 0;
            if(xx>=22) scr_scrup(0x1e,18,0,23,79);
            }
        }
        else
        {
            simwol = buf0[i];    goto_xy(xx,yy);
write_simw(ch,attrib);
            if(kbhit())    {
                if((ch_k = getch()) == 0x1b)
                    break;}
            pered_simw(simwol);
            yy++;
            if(yy > 79)
                {yy = 1; xx++;}
            if(xx == 23 & yy>= 79)
                {
                    scr_scrup(0x1e,18,0,23,79);
                    xx = 23; yy = 1;
                }
        }
    }
    cls_tx();
}

// выдать текст из буфера buf2 на передачу
void send_buf2(int x,int y,int len,int attrib)
{
    int i;

```

```

    unsigned int ch;
xx = x;
yy = y;
    for(i=0;i<len;i++)
    {
        ch = buf2[i];
        if(ch == NULL) break;
        if(ch < 32)
        {
            if(ch == 0x0A | ch == 0x0D) {xx++; yy = 0;
                                         if(xx>=23) scr_scrup(0x1e,18,0,23,79);
                                         xx=23; yy=1;
            }
        }
    else
    {
        simwol = buf2[i];
goto_xy(xx,yy);
write_simw(ch,attrib);
        if(kbhit()) {
            if((ch_k = getch()) == 0x1b)
                break;}
        pered_simw(simwol);
        yy++;
        if(yy > 79)
            {yy = 1; xx++;}
        if(xx == 23 & yy >= 79)
        {
            scr_scrup(0x1e,18,0,23,79);
            xx = 23; yy = 1;
        }
    }
    }
    cls_tx();
}

// выдать текст из буфера buf0 на передачу
void send_buf1(int x,int y,int len,int attrib)
{
    int i;

```



```

    unsigned int ch;
xx = x; yy = y;
    for(i=0;i<len;i++)
    {
        ch = buf1[i];
        if(ch == NULL) break;
        if(ch < 32)
        {
            if(ch == 0x0A | ch == 0x0D) {xx++; yy = 0;
                                         scr_scrup(0x1e,18,0,23,79);
                                         }
        }
    }
    else
    {
        simwol = buf1[i];
goto_xy(xx,yy);
        write_simw(ch,attrib);
        if(kbhit()) {
            if((ch_k = getch()) == 0x1b)
                break;}
            pered_simw(simwol);
            yy++;
            if(yy > 79)
                {yy = 1; xx++;}
            if(xx == 23 & yy >= 79)
            {
                scr_scrup(0x1e,18,0,23,79);
                xx = 23; yy = 1;
            }
        }
    }
    cls_tx();
}

// вывести текст из буфера на экран
void co_buf(int x,int y,int len,int count,int attrib)
{
    int i;
    unsigned int ch;

```

```

xx = x;
yy = y;
    for(i=0;i<len;i++)
        {
if(count == 1) ch = buf1[i];
if(count == 0) ch = buf0[i];
        if(ch == NULL) break;
        if(ch < 32)
            {
                if(ch ==0x0A | ch == 0x0D) {xx++; yy = 0;
                                                if(xx>=23) scr_scrup(0x1e,18,0,23,79);
                                                }
            }
        else
            {
                simwol = buf1[i];
goto_xy(xx,yy);
write_simw(ch,attrib);
                yy++;
                if(yy > 79)
                    {yy = 1; xx++;}
                if(xx == 24 & yy >= 79)
                    {
                        scr_scrup(0x1e,18,0,23,79);
                        xx = 23; yy = 1;
                    }
            }
        }
    }

/* начальная фраза QSO */
void wyzow(void)
{
    goto_xy(18,1);
    if(qso.call_sign[0] == NULL) {promt(txt20[0],19,10,23,69);
        goto_xy(18,1); return;
    }
    else
        printf("%s de %s",qso.call_sign,my_call);
}

```

```
    read_string(18,1,17,buf0);
    goto_xy(18,1);
    send_buf0(18,1,17,0x1d);
    cls_buf0(80);
}

/* начальная фраза QSO вариант для текста по <Alt>—<F5> */
void wyzowl(void)
{
    goto_xy(18,1);
    if(qso.call_sign[0] == NULL) {prompt(txt20[0],19,10,23,69);
    goto_xy(18,1); return;
    }
    else
    printf("%s de %s — HELLO DR OM — UR RST IS %s\n",qso.call_sign,my_call,qso.rst_sent,qso.rst_sent);
    read_string(18,1,52,buf0);
    goto_xy(18,1);
    send_buf0(18,1,52,0x1d);
    cls_buf0(80);
}

/* начальная фраза QSO второй вариант*/
void wyzow2(void)
{
    goto_xy(18,1);
    if(qso.call_sign[0] == NULL) prompt(txt20[0],19,10,23,69);
    if(qso.name_k[0] == NULL) { prompt(txt21[0],19,10,23,69);
    goto_xy(18,1); return;
    }
    else
    printf("%s de %s — R OK DR %s",qso.call_sign,my_call,qso.name_k);
    read_string(18,1,35,buf0);
    goto_xy(18,1);
    send_buf0(18,1,35,0x1d);
    cls_buf0(80);
}

/* конечная фраза QSO */
void wyhod(void)
```

```

{
goto_xy(18,1);
if(qso.call_sign[0] == NULL) {promt(txt20[0],19,10,23,69);
goto_xy(18,1); return;}
    else
        printf("%s de %s pse k",qso.call_sign,my_call);
        read_string(18,1,25,buf0);
        goto_xy(18,1);
        send_buf0(18,1,25,0x1d);
        cls_tx();
        cls_buf0(80);
}

/* конечная фраза QSO вариант 2*/
void wyhod2(void)
{
goto_xy(18,1);
if(qso.call_sign[0] == NULL) {promt(txt20[0],19,10,23,69); goto_xy(18,1);}
    else
        printf("%s de %s k",qso.call_sign,my_call);
        read_string(18,1,20,buf0);
        goto_xy(18,1);
        send_buf0(18,1,20,0x1d);
        cls_tx();
        cls_buf0(80);
}

// Создать окно подсказки
void promt(char *text,int x,int y, int endx,int endy) {
    unsigned int *p;
    /* размещение памяти для видеобуфера */
    p=(unsigned int*)malloc((endx-x+1)*(endy-y+1));
    if(!p) exit(1); /* Вы можете здесь сами обработать ошибку */
    /* сохранение части экрана */
    save_video(x-1,endx,y-1,endy+1,p);
    wind_zw(0x4a,x-1,y-1,endx-1,endy);
    draw_border(x-1,y-1,endx-1,endy);
    /* высвечивание меню на своем месте */

```

```
goto_xy(x+1,y+13);
printf("%s",text);
sleep(1);
/* восстановление части экрана */
restore_video(x-1,endx,y-1,endy+1,p);
free(p);
return;
}

// очистка буфера buf1
void cls_buf1(int len)
{
    int i;
    for(i=0;i<len;i++)
        buf1[i]=0x0;
}

// очистка буфера buf0
void cls_buf0(int len)
{
    int i;
    for(i=0;i<len;i++)
        buf0[i]=0x0;
}

// очистка буфера buf2
void cls_buf2(int len)
{
    int i;
    for(i=0;i<len;i++)
        buf2[i]=0x0;
}
```

В листинге 5.22 находится функция переключений при работе на передачу, а также различные функции, задействованные в этих переключениях.

Здесь также много функций передачи различных файлов, некоторые функции по работе с буферами. Все описанные в этом листинге функции очень простые и какого-либо пояснения не требуют.

Листинг 5.22. Продолжение 8

```
/* переключения при передаче */
int exec_cmd(int ch_k)
{
    int i, j;
get_time();
    switch (ch_k) {
        case F4      : izm_pr1();
                       break;
        case F5      : izm_pr();
                       break;
        case ALT_F2   : alt_f2();
                       break;
        case ALT_F3   : alt_f3();
                       break;
        case ALT_F4   : alt_f4();
                       break;
        case ALT_F5   : alt_f5();
                       break;
        case ALT_F6   : alt_f6();
                       break;
        case ALT_F7   : alt_f7();
                       break;
        case ALT_F8   : alt_f8();
                       break;
        case ALT_F9   : alt_f9();
                       break;
        case ALT_F10  : alt_f10();
                       break;
        case SHF_F2   : izm_ton1();
                       break;
        case SHF_F3   : izm_ton();
                       break;
        case CTL_F5   : ct_f5();
                       break;
        case CTL_F6   : ct_f6();
                       break;
        case CTL_F7   : ct_f7();
                       break;
```

```
        case CTL_F8 : ct_f8();
                      break;
        case CTL_F9 : ct_f9();
                      break;
        case F2      : uwel_skor();
                      break;
        case F3      : umen_skor();
                      break;
        case 199     : peredacha();      //<Home>
                      break;           //<End>
        case 207     : wyhod3();
                      break;
        case PGUP    : rus_lat();
                      break;
        default      : break;
    }
    return(0);
}
```

// ОБЩИЙ ВЫЗОВ

```
void alt_f2(void)
{
    ust_rts();
    write_video(2,31,*txt3,0xf);
    bord_col(0x3);
    goto_xy(18,1);
    copy_buf(6);
    co_buf(18,1,400,1,0x1e);
    send_buf1(18,1,400,0x1d);
    sbros_dtr_rts();
    write_video(2,31,*txt4,0xf);
    bord_col(0);
    cls_buf1(400);
}
```

// ВЫДАТЬ QRZ

```
void alt_f3(void)
{
    write_video(2,31,*txt3,0xf);
```

```
bord_col(0x3);
goto_xy(18,1);
    ust_rts();
copy_buf(5);
    co_buf(18,1,400,1,0x1e);
send_buf1(18,1,400,0x1d);
    sbros_dtr_rts();
write_video(2,31,*txt4,0xf);
bord_col(0);
cls_buf1(400);
}
```

```
void copy_buf_fl(int count)
{
    int i,letter;
    for(i=0;i<count;i++)
    {
        if((letter = fgetc(fp)) != EOF)
        {
            buf2[i] = letter;
        }
    }
    fclose(fp);
    return;
}
```

// вызов определенного корреспондента

```
void alt_f4(void)
{
    goto_xy(18,1);
    if(qso.call_sign[0] == NULL) {promt(txt20[0],19,10,23,69);
                                goto_xy(18,1); return;
                                }
    else
        bord_col(0x3);
write_video(2,31,*txt3,0xf); goto_xy(18,1);
printf("%s %s de %s %s %s PSE K",qso.call_sign,qso.call_sign,
                                my_call,my_call,my_call);
read_string(18,1,45,buf0);
```



```
goto_xy(18,1);
    ust_rts();
send_buf0(18,1,45,0x1d);
    sbros_dtr_rts();
    write_video(2,31,*txt4,0xf);
bord_col(0);
cls_tx();
cls_buf0(80);
}

// начало qso
void alt_f5(void)
{
    write_video(2,31,*txt3,0xf);
    bord_col(0x3);          ust_rts();
    goto_xy(18,1);
    wyzow1();
    if(qso.call_sign[0] == NULL) { bord_col(0);
    write_video(2,31,*txt4,0xf);
        sbros_dtr_rts(); return;}
    copy_buf(3);
    co_buf(18,1,400,1,0x1e);
    send_buf1(18,1,400,0x1d);
    wyhod();
    write_video(2,31,*txt4,0xf);
    bord_col(0);
        sbros_dtr_rts();
    cls_buf1(400);
}

// об annaparatype
void alt_f6(void)
{
    write_video(2,31,*txt3,0xf);
    bord_col(0x3);
        ust_rts();
    goto_xy(18,1);
    wyzow2();
    if(qso.call_sign[0] == NULL) { bord_col(0);
```

```
    write_video(2,31,*txt4,0xf);
        sbros_dtr_rts(); return;}
    if(qso.name_k[0] == NULL) { bord_col(0);
    write_video(2,32,*txt4,0xf);
        sbros_dtr_rts(); return;}
    copy_buf(1);
    co_buf(18,1,400,1,0x1e);
    send_buf1(18,1,400,0x1d);
wyhod();
        sbros_dtr_rts();
    write_video(2,31,*txt4,0xf);
    bord_col(0);
    cls_buf1(400);
}
// концовка qso
void alt_f7(void)
{
    write_video(2,31,*txt3,0xf);
    bord_col(0x3);
        ust_rts();
    goto_xy(18,1);
wyzow2();
    if(qso.call_sign[0] == NULL) { bord_col(0);
    write_video(2,31,*txt4,0xf);
        sbros_dtr_rts(); return;}
    copy_buf(4);
    co_buf(18,1,400,1,0x1e);
    send_buf1(18,1,400,0x1d);
wyhod2();
        sbros_dtr_rts();
    write_video(2,32,*txt4,0xf);
    bord_col(0);
    cls_buf1(400);
}

// работать с клавиатуры
void alt_f8(void)
{
    write_video(2,31,*txt3,0xf);
```

```

    bord_col(0x3);
        ust_rts();
    goto_xy(18,1);
wyzow2();
    goto_xy(18,1);
    xx=18; yy=1;
    do{
        do{
    ch_n = getch();
        if(ch_n == 13) goto wyh1;
        if(ch_n == 8) {goto_xy(xx,--yy); putchar(0x0);goto wyh0;}
        write_char(xx,yy,ch_n,0x1e);
    yy++;
    if(ch_n == 32)
        {
            read_word(18,1,20,buf0);
            xx=18; yy=1;
        }
wyh0:
    if(ch_n == 27) exit(0);
        }
        while(ch_n != 32);
    send_buf0(18,1,17,0x1d);
    xx=18; yy=1; goto_xy(18,1);
        }
        while(ch_n != 13 || ch_n != 9);
wyh1:
        wyhod2();
            sbros_dtr_rts();
        write_video(2,31,*txt4,0xf);
        bord_col(0);
        cls_buf1(400);
    }

// просьба повторить имя и qth
void alt_f9(void)
{
    write_video(2,31,*txt3,0xf);
    bord_col(0x3);

```

```
        ust_rts();
    goto_xy(18,1);
wyzow();
    if(qso.call_sign[0] == NULL) { bord_col(0);
    write_video(2,32,*txt4,0xf);
        sbros_dtr_rts();    return;}
    copy_buf(2);
    co_buf(18,1,400,1,0x1e);
send_buf1(18,1,400,0x1d);
    bord_col(0);
wyhod();
    write_video(2,31,*txt4,0xf);
        sbros_dtr_rts();
    bord_col(0);
    cls_buf1(400);
}
```

// передать ASCII-файл

```
void alt_f10(void)
```

```
{
char filename[15];
promt(txt22[0],19,10,23,69);
goto_xy(18,1);
gets(filename);
if((fp = fopen(filename, "r")) == NULL)
{
    printf("Невозможно открыть файл %s\n",filename);
    getchar();
    exit(0);
}
write_video(2,31,*txt3,0xf);
bord_col(0x3);
    ust_rts();
copy_buf_f1(4000);
send_buf2(23,1,4000,0x1e);
write_video(2,31,*txt4,0xf);
    sbros_dtr_rts();
```

```
        bord_col(0);
        cls_buf2(4000);
    }

// общая подпрограмма для Ctrl_Fn qso
void ctrl_msg(void)
{
    write_video(2,31,*txt3,0xf);
    bord_col(0x3);
        ust_rts();
    goto_xy(18,1);
    wyzow();
    if(qso.call_sign[0] == NULL) { bord_col(0);
    write_video(2,31,*txt4,0xf);
    return;}
    copy_buf(count);
    co_buf(18,1,400,1,0x1e);
    send_buf1(18,1,400,0x1d);
    wyhod2();
    write_video(2,31,*txt4,0xf);
        sbros_dtr_rts();
    bord_col(0);
    cls_buf1(400);
}

// выдать на передачу текст f5.msg
void ct_f5(void)
{
    count = 7;
    ctrl_msg();
}

// выдать на передачу текст f6.msg
void ct_f6(void)
{
    count = 8;
    ctrl_msg();
}
```

```
// выдать на передачу текст f7.msg
void ct_f7(void)
{
    count = 9;
    ctrl_msg();
}

// выдать на передачу текст f8.msg
void ct_f8(void)
{
    count = 10;
    ctrl_msg();
}

// выдать на передачу текст f9.msg
void ct_f9(void)
{
    count = 11;
    ctrl_msg();
}

// увеличение скорости передачи
void uwel_skor(void)
{
    static int wsp;
    getxy();      // запомнить координаты курсора в xx1 и yy1
    wsp = skorost - 3;
    skorost = wsp;
    if(skorost < 8) skorost = 48;
    izm_skor_p();
    skorost2 = skorost;
    izm_skor_priema();
    goto_xy(xx1-1,yy1);
    return;
}

// уменьшение скорости передачи
void umen_skor(void)
{
    static int wsp;
```

```
getxy();      // запомнить координаты курсора в xx1 и yy1
    wsp = skorost +3;
    skorost = wsp;
if(skorost > 48) skorost = 9;
    izm_skor_p();
skorost2 = skorost;
    izm_skor_priema();
goto_xy(xx1-1,yy1);
    return ;
}

//изменение скорости передачи
void izm_skor_p(void)
{
    goto_xy(2,65);
    printf("      ");
    goto_xy(2,65);
if(skorost ==48) {puts("42"); del1=2; del2 = 10;sch_posylok=23;}
if(skorost ==45) {puts("45"); del1=2; del2 = 10;sch_posylok=22;}
if(skorost ==42) {puts("48"); del1=2; del2 = 10;sch_posylok=21;}
if(skorost ==39) {puts("50"); del1=2; del2 = 10;sch_posylok=20;}
if(skorost ==36) {puts("55"); del1=2; del2 = 10;sch_posylok=19;}
if(skorost ==33) {puts("60"); del1=2; del2 = 10;sch_posylok=18;}
if(skorost ==30) {puts("65"); del1=2; del2 = 10;sch_posylok=17;}
if(skorost ==27) {puts("70"); del1=2; del2 = 10;sch_posylok=16;}
if(skorost ==24) {puts("75"); del1=1; del2 = 10;sch_posylok=15;}
if(skorost ==21) {puts("85"); del1=1; del2 = 10;sch_posylok=14;}
if(skorost ==18) {puts("95"); del1=1; del2 = 10;sch_posylok=13;}
if(skorost ==15) {puts("110");del1=1; del2 = 10;sch_posylok=12;}
if(skorost ==12) {puts("125");del1=1; del2 = 10;sch_posylok=11;}
if(skorost == 9) {puts("150");del1=1; del2 = 10;sch_posylok=10;}
    return;
}

// изменение тона сигнала в динамике — понижение
void izm_ton(void)
{
static int wsp;
    getxy();      // запомнить координаты курсора в xx1 и yy1
```

```
wsp = ton - 100;
    ton = wsp;
    if(ton < 400) ton =1600;
        izm_skor_p();
        goto_xy(xx1-1,yy1);
goto_xy(1,74);
    puts("      ");
goto_xy(1,74);
printf("%dГц",ton);
    return;
}
```

// изменение тона сигнала в динамике — повышение

```
void izm_tonl(void)
{
static int wsp;
    getxy(); // запомнить координаты курсора в xx1 и yy1
    wsp = ton + 100;
    ton = wsp;
    if(ton > 1600) ton =400;
        izm_skor_p();
        goto_xy(xx1-1,yy1);
goto_xy(1,74);
    puts("      ");
goto_xy(1,74);
printf("%dГц",ton);
    return;
}
```

// уменьшение скорости приема

```
void izm_pr(void)
{
    skorost2 = skorost2+3;
    if(skorost2 > 48) skorost2 = 9;
        izm_skor_priema();
        return ;
}
```

// уменьшение скорости передачи

```
void izm_prl(void)
```



```
{
    skorost2 = skorost2 -3;
if(skorost2 < 9) skorost2 = 48;
    izm_skor_priema();
    return ;
}

//изменение скорости приема
void izm_skor_priema(void)
{
    goto_xy(2,74);
    printf("  ");
    goto_xy(2,74);

    if(skorost2 ==48) {puts("42"); del1=2; del2 = 10;sch_posylok=23;}
    if(skorost2 ==45) {puts("45"); del1=2; del2 = 10;sch_posylok=22;}
    if(skorost2 ==42) {puts("48"); del1=2; del2 = 10;sch_posylok=21;}
    if(skorost2 ==39) {puts("50"); del1=2; del2 = 10;sch_posylok=20;}
    if(skorost2 ==36) {puts("55"); del1=2; del2 = 10;sch_posylok=19;}
    if(skorost2 ==33) {puts("60"); del1=2; del2 = 10;sch_posylok=18;}
    if(skorost2 ==30) {puts("65"); del1=2; del2 = 10;sch_posylok=17;}
    if(skorost2 ==27) {puts("70"); del1=2; del2 = 10;sch_posylok=16;}
    if(skorost2 ==24) {puts("75"); del1=1; del2 = 10;sch_posylok=15;}
    if(skorost2 ==21) {puts("85"); del1=1; del2 = 10;sch_posylok=14;}
    if(skorost2 ==18) {puts("95"); del1=1; del2 = 10;sch_posylok=13;}
    if(skorost2 ==15) {puts("110");del1=1; del2 = 10;sch_posylok=12;}
    if(skorost2 ==12) {puts("125");del1=1; del2 = 10;sch_posylok=11;}
    if(skorost2 == 9) {puts("150");del1=1; del2 = 10;sch_posylok=10;}

    return;
}
```

В листинге 5.23 размещены функции программы, предназначенные для работы с аппаратным журналом. Аппаратный журнал, по моему мнению, должен быть в каждой программе, предназначенной для цифрового вида связи. Встроенный в эту программу аппаратный журнал является простейшим представителем своего вида, но обладает всеми необходимыми функциями для хранения и поиска записей о проведенных радиосвязях.

Листинг 5.23. Продолжение 9

```
// добавить запись о QSO в аппаратный журнал
void addqso(void)
{
    int i;
    if(nomer == MAX)
    {
        puts("В этом файле журнала больше места нет. Сделайте новый.\n");
        getchar();
        return;
    }
    if((fp = fopen( FILENAME, "a")) == NULL)
    {
        printf("Невозможно открыть файл %s\n", FILENAME);
        exit(0);
    }
    for(i=0; i<20; i++)
    {
        disc.name[i] = qso.call_sign[i];
        disc.namkor[i] = qso.name_k[i];
        disc.qth[i] = qso.qth[i];
        disc.rs1[i] = qso.rst_sent[i];
        read_word(1, 26, 5, disc.tim);
        read_word(1, 44, 10, disc.date);
        read_word(0, 38, 2, disc.mod);
    }
    promt(txt23[0], 19, 10, 23, 69);
    goto_xy(19, 5);
    puts("    Так Вы УВЕРЕНЫ, что все правильно в экранной таблице?");
    puts("\n        Тогда нажимайте <Enter>, иначе для выхода жмите
        <Esc>!!!");

    i = getch();
    if(i == 27) {cls_tx(); return;}
    else
        cls_tx();
    goto_xy(18, 0);
    printf("Введите диапазон (до 3 цифр): ");
    gets(disc.diap);
```

```

printf("Введите RST к Вам (до 3 цифр): ");
gets(disc.rs2);
printf("Введите информацию по диплому (до 13 букв): ");
gets(disc.category);
printf("Введите дополнительную информацию (до 50 букв) : ");
gets(disc.infor);
disc.number = nomer;
if(nomer == 12 || nomer == 25) {nomer = nomer+2; goto wyh;}
else
    nomer = nomer + 1;
wyh:  fwrite(&disc, sizeof(disc), 1, fp);
      fclose(fp);
      cls_tx();
      return;
    }

/* вывести записи на экран */
void wlist(void)
{
    int ch;
    if((fp = fopen(FILENAME,"r")) == NULL)
    {
        printf("Невозможно открыть файл %s\n",FILENAME);
        exit(0);
    }
    cls(0x1e);
    goto_xy(0,0);
    puts(
"  N   Диап.      Дата   Время   Позывной      Город   Мод.
Диплом");
    puts("
=====
");
    while(fread(&disc,sizeof(disc), 1, fp)==1)
    {
        if(disc.number >= 24) {ch_k = getch();
            if(ch_k == 27) break;
            else
                goto wyh;}
    }
    else

```

```

wyh:
    printf(
"%4d |%4s|%12s|%6s|%9s |%15s|%4s|%15s|\n", disc.number, disc.diap,
    disc.date, disc.tim, disc.name, disc.qth, disc.mod, disc.category);
    }

        printf("    Для продолжения нажмите <Enter>");
                getch();

        fclose(fp);
        return;
    }

//ввести номер последнего QSO из файла
void nomer_qso(void)
{
if((fp = fopen(FILENAME, "r")) == NULL)
{ printf("Невозможно открыть файл %s\n", FILENAME);
    exit(0);
}
while(fread(&disc, sizeof(disc), 1, fp) != NULL)
{    nomer = disc.number; }
nomer = nomer + 1;
    fclose(fp);
}

```

В листинге 5.24 располагаются функции, которые управляют величинами электрических напряжений на различных выводах СОМ-порта, а также функции для работы со звуковой картой компьютера в режиме передачи (генерация звука нужной частоты).

Листинг 5.24. Продолжение 10

```

// установка сигнала DTR
void ust_dtr(void)
{    asm mov dx, com_adr1
    asm mov al, 1
    asm out dx, al
    }

// установка сигнала RTS
void ust_rts(void)

```

```
{
asm mov dx, com_adr1
asm mov al, 2
asm out dx,al
}

// установка сигнала DTR и RTS
void ust_dtr_rts(void)
{
asm mov dx, com_adr1
asm mov al,3
asm out dx,al
}

// сброс сигналов DTR и RTS
void sbros_dtr_rts(void)
{
asm mov dx,com_adr1
asm mov al,0
asm out dx,al
}

// далее расположены несколько функций, работающих со звуком
void SB1RegSet(int regis, int value)
{
outportb(0x220,regis);
delay(6);
outportb(0x221,value);
delay(24);
}

void NullRegistr(void)
{
SB1RegSet(0x01, 00);
SB1RegSet(0x08, 00);
SB1RegSet(0x20, 00);
SB1RegSet(0x40, 00);
SB1RegSet(0x60, 00);
SB1RegSet(0x80, 00);
}
```

```
SBlRegSet (0xa0, 00);  
SBlRegSet (0xb0, 00);  
SBlRegSet (0xbd, 00);  
SBlRegSet (0xc0, 00);  
SBlRegSet (0xe0, 00);  
}
```

```
void SetKanal(void)  
{  
    SBlRegSet (0x4, 0x21);  
    delay(90);  
}
```

```
void SetRegistr(void)  
{  
    SBlRegSet (0x20, 0x1);  
    SBlRegSet (0x40, 0x10);  
    SBlRegSet (0x60, 0xf0);  
    SBlRegSet (0x80, 0x77);  
    SBlRegSet (0xa0, 127);  
    SBlRegSet (0x23, 0x1);  
    SBlRegSet (0x43, 00);  
    SBlRegSet (0x63, 0xf0);  
    SBlRegSet (0x83, 0x77);  
    SBlRegSet (0xb0, 0x39);  
    SBlRegSet (0xc0, 0x7);  
}
```

```
void ProwSBl(void)  
{  
    SBlRegSet (0x04, 0x60);  
    SBlRegSet (0x04, 0x80);  
    nn=inportb(0x220);  
    SBlRegSet (0x02, 0xff);  
    SBlRegSet (0x04, 0x21);  
    delay(90);  
    pp=inportb(0x220);  
    SBlRegSet (0x04, 0x60);  
    SBlRegSet (0x04, 0x80);  
}
```

```
if((pp - 0xc0) == 0) {goto_xy(18,1);
    printf("Все OK, карта SB установлена"); }
else
{goto_xy(4,5);printf(" Необходимая звуковая карта не обнаружена!!!");
    getch();}
}
```

В листинге 5.25 располагается одна из важных функций — функция вывода принятого символа на экран. Также здесь находятся некоторые вспомогательные функции, в том числе и функция поиска в файле аппаратного журнала нужной информации и выдачи этой информации на экран в информационную строку.

Листинг 5.25. Продолжение 11

```
// вывести принимаемый символ на экран
void rx_read(void)
{
    goto_xy(xx0,yy0);
    putchar(simwol2);
    yy0++;
    if(yy0 > 78) {yy0 = 1; xx0++;}
    if(xx0 == 16 & yy0 >= 78)
    {
        {scr_scrup(0x1d,3,0,16,79);
        xx0 = 16; yy0 = 1; }
    }
}

// переключение рус/лат
void rus_lat(void)
{
    if(flag_rus == 1) {flag_rus = 0; goto_xy
    (0,77); puts("LAT");}
    else
    {flag_rus = 1;goto_xy(0,77); puts("РУС");}
}

/* вывести help на экран */
void dis_help(void)
```

```
{
char *string;
char msg[80];
int len,dl=1;
int ch;
    if((fp = fopen(FILEHLP,"r")) == NULL)
    {
        printf("Невозможно открыть файл %s\n",FILEHLP);
        exit(0);
    }
    cls(0x1e);
    goto_xy(0,0);
    while(fgets(msg,len,fp) != 0)
    {
        if(dl >= 23) { dl = 1; ch_k = getch();
            if(ch_k == 27) break;
            else
                goto wyh;}
        else
            wyh:
            printf("      %s",msg); dl++;}
        printf("      Для продолжения нажмите <Enter>");
            getch();

        fclose(fp);
        return;
    }

/* Поиск определенной записи и вывод данных на экран */
locate()
{
    char name[20];
    char fflag,pause;
    int zif;
    fflag='n';
    if((fp= fopen(FILENAME,"r")) == NULL)
    {
        printf("Невозможно открыть файл %s\n", FILENAME);
        exit(0);
    }
}
```



```

    stpcpy(name,qso.call_sign);
    strlwr(name);
    zif = strlen(name);
    while(fread(&disc,sizeof(disc), 1, fp)==1)
    {
        if(strnicmp(disc.name, name,zif)==0)
        {
            fflag='y';

goto_xy();
goto_xy(17,2);
printf("    Последняя связь: %d %s %s %s %s %s\n",disc.number,disc.date,disc.tim,disc.name,disc.rs1,disc.rs2);
goto_xy(1,9); printf("                ");
goto_xy(1,9); printf("                ");

goto_xy(2,9); printf("%s",disc.qth);
goto_xy(1,9); printf("%s",disc.namkor);

goto_xy(xx1,yy1);

        }
    }
    fclose(fp);
    if(fflag=='n')
{ goto_xy(); goto_xy(17,2);
printf("                Записей о связи с этим позывным в журнале нет.                ");
goto_xy(1,9); printf("                ");
goto_xy(2,9); printf("                ");
goto_xy(xx1,yy1);
}
    return 0;
}

```

На этом закончились листинги исходных кодов программы CW_QSO. Надеюсь, что этот материал поможет вам сделать свои собственные программы для телеграфной радиосвязи. Пускай сначала все будет не так уж и хорошо, но последующие варианты с каждым разом будут становиться все лучше и лучше.

Как сделана программа RTTY_QSO

В этом разделе книги мною размещаются исходные коды программы RTTY_QSO, которые находятся в файле `rtty_qso.cpp`. Программа разработана в среде программирования Turbo C++ 3. Текст файла `rtty_qso.cpp` несколько напоминает предыдущий файл с исходными кодами для программы телеграфной связи. Но это только внешнее сходство, выражающееся в одинаковом оформлении заставки и рабочего экрана.

В программе написано, что она разработана в 2000 году. Это не совсем так. В 2000 году были разработаны основные части программы — функции приема и передачи, а все внешнее оформление взято из намного ранее разработанных программ в надежде на последующее переоформление внешнего вида. Но время проходило, появлялись все новые идеи, которые требовали немедленной реализации, а оформление программы так и осталось непеределанным. Надеюсь, что вы выполните эту работу в своих программах более аккуратно.

В листинге 5.26 размещено начало файла с исходными кодами, где выполняются объявления подключаемых файлов, размещаются таблицы для кодирования и декодирования символов, объявляются задействованные в программе функции и переменные величины.

Листинг 5.26. Файл `rtty_qso.cpp`

```

/*****
/*  PROGRAMM_NAME          RTTY_QSO v.1.00   (c) RA3XB          */
/*****
/*  Программа для проведения RTTY (телетайпной) любительской */
/*      радиосвязи посредством компьютера IBM PC.          */
/*      Автор разработки — Тяпичев Геннадий А.   < RA3XB >    */
/*      Разработана в сентябре 2000 года          */
/*****
/*  Программа создавалась на Turbo C++ v.3          */
/*****

#include <iostream.h>
#include <stdio.h>
#include <dos.h>
#include <stdlib.h>
#include <process.h>
#include <ctype.h>
#include <string.h>

```

```
#include <bios.h>
#include <conio.h>
#include <sys/timeb.h>
#include <time.h>
#include "rtty_qso.h"
#define BORDER 1
#define ESC 27
#define REV_VID 0x70
#define NORM_VID 0x4a
#define MAX 250
#define FILENAME "qso_log.dat"
#define FILESPEC "qso_log.spc"
FILE *fp;
FILE *log_file;
FILE *init_file;
FILE *send_file;
FILE *save_file;
FILE *printer;
void hlp4_wind(int border);
void bord_col(int num);
void draw_border(int startx, int starty, int endx, int endy);
void write_video(int x, int y, char *p, int attrib);
void save_video(int startx, int endx, int starty, int endy, unsigned int
*buffer);
void restore_video(int startx, int endx, int starty, int endy, unsigned int
*buffer);
void goto_xy(int x, int y), cls(int zwet), scren_col(int rez);
void wind_zw(int attr, int starx, int starty, int x, int y);
void zastawka(void), zastawka2(void);
void err_wind(int x, int y, int endx, int endy);
void write_string(int x, int y, int attrib, char *p);
void write_char(int x, int y, char ch, int attrib);
void hlp_wind(int count, int x, int y, int border);
void display_help(int x, int y, int count);
void zabitie(int x, int y, int attrib, int len);
void hlp2_wind(int border);
void hlp3_wind(int border);
void pered_simw(int ch);
void pauza(void);
```

```
void zwuk(void);
void perek_zw(void);
void peredacha(void);
void getxy(void);
void get_time(/*int date_flag*/void);
void read_string(int x,int y,int len,char *p);
void read_word(int x, int y,int len,char *p);
void scr_scrup(int attr,int x,int y,int xl,int yl);
void init_sys(),send_str(char *str);
void cls_rx(void),cls_tx();
void copy_buf(int count);
void promt(char *text,int x,int y, int endx,int endy);
void send_buf0(int x,int y,int len,int attrib);
void send_buf1(int x,int y,int len,int attrib);
void co_buf(int x,int y,int len,int count,int attrib);
void write_simw(int ch, int attrib);
void wyzow(void),wyzow1(void),wyzow2(void);void wy-
hod(void),wyhod2(void),wyhod3(void);
void alt_f2(), alt_f3(), alt_f4(), alt_f5();
void alt_f6(),alt_f7(),alt_f8(),alt_f9(),alt_f10();
void copy_buf_fl(int x);
void cls_buf1(int len),cls_buf0(int len);
void mouse(void);
int init_str(char *str, int len),exec_cmd(int ch);
void izm_skor_p(void),izm_ton(void),izm_ton1(void);
void uwel_skor(void), umen_skor(void);
void addqso(void),nomer_qso(void),getslots(void),wlist(void);
void
ctrl_msg(void),ct_f5(void),ct_f6(void),ct_f7(void),ct_f8(void),ct_f9(void
);
void ust_dtr(void),ust_rts(void),ust_dtr_rts(void),sbros_dtr_rts(void);
int inport(void);
void priem(void);
void perek1(void);
void rx_read(void);
//void izm_pr(void),izm_pr1(void),izm_skor_priema(void);
void rus_lat(void);
void pered_ru_reg(void), pered_la_reg(void), pered_cif_reg(void);
void t_wysok(void), t_nizk(void);
```

```
void cursor_ubrat(void);  
void rewers_pr(void), rewers_pered(void);  
locate();
```

```
struct {  
    char frequency[5];  
    char mode[6];  
    char tim[8];  
    char date[12];  
    char call_sign[9];  
    char rst_sent[5];  
    char rst_rcvd[5];  
    char name_k[10];  
    char qth[13];  
    char comments[65];  
    char contest[17];  
    int number;  
} qso;
```

```
struct CD  
{  
    char name[9];  
    char diap[5];  
    char date[12];  
    char tim[8];  
    char rs1[5];  
    char rs2[5];  
    char namkor[10];  
    char qth[13];  
    char mod[6];  
    char category[17];  
    char infor[65];  
    int number;  
} disc;
```

```
unsigned char simwol;  
short int skorost;  
int znak;  
int flag_zw = 0;
```

```
unsigned int xx1,yy1;
unsigned int ton = 1000;
unsigned char far *vid_mem;
int slots[MAX];
int prt,i;
int flag_rus = 1;
int flag_lat = 0;
int reg_ru = 0, reg_la = 1, reg_cif = 0;
unsigned int simw_nom;           // номер символа в таблице
unsigned int len;
char simwol2;
unsigned int sch_pomeh = 5;      // данные счетчика помех
unsigned int sch_posylok = 8;    // данные счетчика посылок
char port[7];
char my_call[15];
char my_name[20];
char my_qth[25];
char my_log[80];
char save_fn[80];
char buf1[400];
char buf2[4000];
char buf0[80];
char temper[10];
char buf_kor[10];
char buf_prom[10];

char abc_l1[] = {
93,'T', 10,'O', 32,'H','N','M', 13,'L','R','G','I','P','C','V',
'E','Z','D','B','S','Y','F','X','A','W','J',124,'U','Q','K',91};
char abc_l2[] = {
93,'t', 10,'o', 32,'h','n','m', 13,'l','r','g','i','p','c','v',
'e','z','d','b','s','y','f','x','a','w','j',124,'u','q','k',91};
char abc_r1[] = {
93,'T', 10,'O', 32,'X','H','M', 13,'Л','Р','Г','И','П','Ц','Ж',
'E','З','Д','Б','С','Ы','Ф','Ь','А','В','Й','|','У','Щ','К',91};
char abc_r2[] = {
93,'т', 10,'о', 32,'х','н','м', 13,'л','р','г','и','п','ц','ж',
'е','з','д','б','с','ы','ф','ь','а','в','й','|','у','щ','к',91};
```

```

char abc_cif[] = {
93, '5', 10, '9', 32, 'я', ',', '.', 13, ')', '4', 'ш', '8', '0', ':', '=',
'3', '+', '#', '?', 39, '6', 'э', '/', '-', '2', 'ю', '|', '7', '1', '(', 91};

char *txt1[] = { "Нажмите <Enter>  " };
char *txt2[] = { "RTTY" };
char *txt3[] = { "* * * ПЕРЕДАЧА * * * " };
char *txt4[] = { "* * * ПРИЕМ * * * " };
char *txt5[] = {
" <Alt-X>-выход; <Alt-Fn>-тексты; <Home>-передача; <End>-прием;
<F1>-HELP";
char *txt7[] = { "Выходим в DOS ? (Y/N)  " };
char *txt8[] = { "CALL: " };
char *txt9[] = { "NAME: " };
char *txt10[] = { "QTH: " };
char *txt11[] = { "RST: " };
char *txt12[] = { "темпер.: " };
char *txt20[] = { "Введите ПОЗЫВНОЙ корреспондента";
char *txt21[] = { "Введите ИМЯ корреспондента и RST";
char *txt22[] = { "      Введите ИМЯ файла";
char *txt23[] = { "НЕ ЗАБУДЬТЕ ввести данные по QSO!!!";
char *help4[] = {
"              Дополнительное      меню              ",
"      =====",
"Ctrl-F1 - инфо о программе      Shift-F1 - инфо о програм. ",
"Ctrl-F2 - переключ. звука      Shift-F2 - повысить тон  ",
"Ctrl-F3 - очистка окна TX      Shift-F3 - понизить тон  ",
"Ctrl-F4 - очистка окна RX      Shift-F4 - просмотр. журнал ",
"Ctrl-F5 - передать t5.msg      Alt-X      - выход в DOS      ",
"Ctrl-F6 - передать t6.msg      <End> - выход на прием    ",
"Ctrl-F7 - передать t7.msg      <Home> - выход на передачу  ",
"Ctrl-F8 - передать t8.msg      <Esc> - срочный выход    ",
"Ctrl-F9 - передать t9.msg      <PgUP> - переключен. РУС/ЛАТ"
};

char *help1[] = {
"F1 - данная таблица (HELP)      Alt-F1 - доп. таблица HELP ",
"F2 - увеличить скорость пер.      Alt-F2 - общий вызов (CQ)  ",
"F3 - уменьшить скорость пер.      Alt-F3 - передать      QRZ   ",

```

```

"F4 — прием норм/обратный      Alt-F4 — вызов корреспонд.",
"F5 — передача норм/обратн.    Alt-F5 — текст начала QSO ",
"F6 — ввести позывной — CALL   Alt-F6 — о своей аппарат.",
"F7 — ввести имя корресп.      Alt-F7 — концовка      QSO ",
"F8 — ввести город — QTH       Alt-F8 — передача с клав. ",
"F9 — ввести передав. RST       Alt-F9 — просьба повторить",
"F10 — ввести темп. воздуха     Alt-F10 — передать файл  ",
"F12 — обновление экрана       F11 — ввести QSO в журнал "
};

```

```

char *help2[] = {
    "                ИНФОРМАЦИЯ О ПРОГРАММЕ                ",
    "        Коды программы разработаны Тяпичевым Г.А. — RA3XB ",
    "в 2000 году в среде программирования Turbo C++ v.3. ",
    "Программа предназначена только для радиолюбительской ",
    "связи и только по принципу 'как есть', т. е. никаких га-",
    "рантий и никаких претензий. Распространяется бесплатно. ",
    };

```

```

char *help3[] = {
    "        Программа рассчитана на работу в составе любительской ",
    "радиостанции совместно с простейшим модемом, преобразую-",
    "щим аудиосигналы приемника в токовые и бестоковые посыл-",
    "ки, удовлетворяющие условиям работы последовательного ",
    "порта COM1 или COM2 компьютера. Более подробные сведения ",
    "находятся в файле rtty_qso.doc, входящем в состав комп-",
    "лекта RTTY_QSO. ",
    };

```

```

int ch_k;
int ch_n;
int yy0=0;
int xx0=3;
int xx,yy;
int col;
int row;
int flag1 = 0;
int date_flag = 0;
int save = 0;
int print =0;

```



```

int border = 0;
int v_attribute;
int count,ch;
int old_x,old_y;
unsigned int nomer = 0;
char help,str;
unsigned int com_x;
int com_adr1 = 0x2fc;
int com_adr2 = 0x2fe;
int nn,pp;
char nota = 0x0e;
int rew_pr=1, rew_per=1;
int rr=1, rr1 = 27, rr2 = 0, rr3 = 31; int fl_d=0, fl_e=0, fl_pus=0,
iz=0;

```

В листинге 5.27 описывается главная функция программы — функция `main()`. Первой в работу включается функция `init_sys()`, которая проводит инициализацию программы. Затем на экран выдается первая информационная заставка, в которой сообщаются название программы и имя автора. После нажатия на клавишу <Enter> на экране появляется заставка рабочего экрана.

Листинг 5.27. Продолжение 1

```

main()                // Начало подпрограммы main()
{
    init_sys();
    zastawka();
    zastawka2();
    skorost = 11;
    ton = 1000;
    izm_skor_p();
    if(flag_zw == 1) {goto_xy(2,78); putchar(nota);}
    cursor_ubrat();
    rus_lat();
    zabitie(2,31,0x0,14);
    write_string(2,31,0xf,*txt4);
    if(xx1 < 3) xx1 = 4;
    goto_xy(xx1-1,yy1);
}

```

```

whod1:
    get_time();
    if(kbhit()) perek1();
    prt = inport();          // получить данные из порта
                              // проверить на наличие сигнала
    if(prt == 1) priem();    // если сигнал есть, перейти на прием
    goto whod1;
}
// конец подпрограммы main();

```

В листинге 5.28 описывается функция переключений, связанных с текстами рабочего экрана программы. Через несколько лет после создания эта часть программы мне кажется очень неудобной. Старайтесь придумать что-либо свое или позаимствуйте на стороне.

Листинг 5.28. Продолжение 2

```

// большая подпрограмма разных переключений
void perek1()
{
    ch_k = getch();
    if(ch_k == 0) { ch_k = getch() + 128;}
    if(ch_k == 27) {sbros_dtr_rts();exit(1);} /* Esc – быстрый выход */
    if(ch_k == 173) goto wyh6;    /* Alt-X – выход через y/n */
    if(ch_k == 187) goto wyh8;    /* F1 – help1 */
    if(ch_k == 222) goto wyh9;    /* Ctrl-F1 – help2 */
    if(ch_k == 212) goto wyh20;   /* Shift-F1 – help3 */
    if(ch_k == 232) goto wyh21;   /* Alt-F1 – help4 */
    if(ch_k == 192) goto wyh1;    /* F6 – ввод CALL */
    if(ch_k == 193) goto wyh2;    /* F7 – ввод NAME */
    if(ch_k == 194) goto wyh3;    /* F8 – ввод QTH */
    if(ch_k == 195) goto wyh4;    /* F9 – ввод RST */
    if(ch_k == 196) goto wyh5;    /* F10 – ввод температуры */
    if(ch_k == 223) perek_zw();    /* Ctrl-F2 – переключение звука */
    if(ch_k == 223) cls_rx();      /* Ctrl-F3 – очистка окна приема*/
    if(ch_k == 224) cls_tx();      /* Ctrl-F4 – очистка окна передачи*/
    if(ch_k == 261) addqso();      /* F11 – ввести QSO в журнал */
    if(ch_k == 262) zastawka2();   /* F12 – обновление экрана */
    if(ch_k == 215) {wlist();zastawka2();} /* Shift-F4 – чит.журнал*/
    goto whod2;
}

```

```

wyh1: write_string(0,3,0x07,txt8[0]);
      zabitie(0,9,0x5b,16); goto whod2;
wyh2: write_string(1,3,0x07,txt9[0]);
      zabitie(1,9,0x5b,16); goto whod2;
wyh3: write_string(2,4,0x07,txt10[0]);
      zabitie(2,9,0x5b,16); goto whod2;
wyh4: write_string(0,60,0x07,txt11[0]);
      zabitie(0,65,0x5b,14); goto whod2;
wyh5: write_string(1,56,0x07,txt12[0]);
      zabitie(1,65,0x5b,8); goto whod2;
wyh8: hlp_wind(11,4,10,BORDER); goto whod2;
wyh9: hlp2_wind(BORDER);      goto whod2;
wyh20: hlp3_wind(BORDER);      goto whod2;
wyh21:hlp4_wind(BORDER);
whod2:
      if(ch_k == 192) goto wyh10; //F6 - ввод CALL
      if(ch_k == 193) goto wyh11; //F7 - ввод NAME
      if(ch_k == 194) goto wyh12; //F8 - ввод QTH
      if(ch_k == 195) goto wyh13; //F9 - ввод RST
      if(ch_k == 196) goto wyh14; //F10 - ввод температуры
exec_cmd(ch_k);
bord_col(0);
return;
wyh10:
      if(flag1 == 0) {xx = 0; yy =9; flag1 = 1;}
      ch_n = getch(); write_char(xx,yy,ch_n,0x5b);
      if(ch_n == 13 || ch_n == 3) {
        write_string(0,3,0x4e,txt8[0]); flag1 = 0;
        read_word(0,9,6,qso.call_sign);
locate();
return;}
      yy++;      goto whod2;
wyh11:
      if(flag1 == 0) {xx = 1; yy =9; flag1 = 1;}
      ch_n = getch(); write_char(xx,yy,ch_n,0x5b);
      if(ch_n == 13 || ch_n == 3) {
        write_string(1,3,0x4e,txt9[0]); flag1 = 0;
        read_word(1,9,8,qso.name_k);

```

```
return;}
    yy++;    goto whod2;
wyh12:
    if(flag1 == 0) {xx = 2; yy =9; flag1 = 1;}
    ch_n = getch(); write_char(xx,yy,ch_n,0x5b);
    if(ch_n == 13 || ch_n == 3) {
        write_string(2,4,0x4e,txt10[0]); flag1 = 0;
        read_word(2,9,8,qso.qth);
    return;}
    yy++;    goto whod2;
wyh13:
    if(flag1 == 0) {xx = 0; yy =65; flag1 = 1;}
    ch_n = getch(); write_char(xx,yy,ch_n,0x5b);
    if(ch_n == 13 || ch_n == 3) {
        write_string(0,60,0x4e,txt11[0]); flag1 = 0;
        read_word(0,65,5,qso.rst_sent);
    return; }
    yy++;    goto whod2;
wyh14:
    if(flag1 == 0) {xx = 1; yy =65; flag1 = 1;}
    ch_n = getch(); write_char(xx,yy,ch_n,0x5b);
    if(ch_n == 13 || ch_n == 3) {
        write_string(1,56,0x4e,txt12[0]); flag1 = 0;
        read_word(1,65,6,temper);
    return; }
    yy++;    goto whod2;
wyh6:
    err_wind(/*txt7[0],*/19,10,23,69);
    if(ch_k == 121 || ch_k == 89) {
        sbros_dtr_rts(); exit(1);    }
    else
        goto_xy(18,0);
return;
wyh7:
    return;
}
/* Конец подпрограммы переключений */
```

В листинге 5.29 выполнено описание функции инициализации программы и функции чтения данных из конфигурационного файла. Функции очень простые, никаких особенностей.

Листинг 5.29. Продолжение 3

```
// инициализация программы
void init_sys()
{
    char ch, *err;
    init_file = fopen("RTTY_QSO.CFG", "r");
    if (init_file == 0) {
        printf("НЕ открывается файл 'RTTY_QSO.CFG'\n");
        printf("\nСоздайте текстовым редактором следующее:\n");
        printf("    COM-порт\n");
        printf("    позывной своей р/станции\n");
        printf("    свое имя\n");
        printf("    свой город (QTH)\n");
        printf("    имя Log файла и путь к нему\n");
        printf("    имя Save файла и путь к нему\n");
        getchar();
        exit(1);
    }
    else {
        init_str(port, 7);
        init_str(my_call, 15);
        init_str(my_name, 20);
        init_str(my_qth, 25);
        init_str(my_log, 80);
        init_str(save_fn, 80);
        init_str(temper, 10);
        init_str(qso.call_sign, 10);
        init_str(qso.name_k, 15);
        init_str(qso.qth, 20);
        init_str(qso.rst_sent, 6);
    }
    fclose(init_file);
    if(strcmp(port, "COM1")==0) {com_x = 0; com_adr1=0x3fc;
com_adr2=0x3fe;}
```

```
    if(strcmp(port,"COM2")==0) {com_x = 1; com_adr1=0x2fc;
com_adr2=0x2fe;}
    if(strcmp(port,"COM3")==0) {com_x = 2; com_adr1=0x3ec;
com_adr2=0x3ee;}
    if(strcmp(port,"COM4")==0) {com_x = 3; com_adr1=0x2ec;
com_adr2=0x2ee;}
    _bios_serialcom(_COM_INIT,com_x,_COM_1200|_COM_CHR8|_COM_STOP1);
    nomer_qso();
}

// чтение конфигурационного файла
int init_str(char *str, int len)
{
    char *err;
    err = fgets(str, len, init_file);
    if (err == NULL) {
        printf("Ошибка чтения файла RTTY_QSO.CFG\n");
        exit(1);
    }
    str[strlen(str) - 1] = 0;
    return(0);
}
```

В листинге 5.30 выполнено описание функций, обеспечивающих прием сигналов от модема, декодирование этих сигналов и превращение их в понятные всем нам буквенные символы.

Как только в функции `main()` обнаруживается сигнал стартового бита, в работу вступает функция `priem()`. Каждая информационная посылка просматривается функцией три раза и по этим трем пробам делается вывод о том, представляет полученная посылка сигнал 1 или сигнал 0. Точно так же просматриваются все пять информационных посылок. Сочетание из пяти информационных посылок представляет телетайпный код символа. Функция просматривает кодировочные таблицы, в которых номер места расположения символа в таблице является телетайпным кодом этого символа. Итак, после того, как функция определила телетайпный код символа, она идет к соответствующей таблице и берет символ, находящийся на указанном месте. Этот символ и есть та необходимая нам буква, которая затем отображается на экране.

В любительской радиосвязи есть правило, по которому прежде чем передать в эфир позывной своей радиостанции, нужно передать сочетание из двух букв DE. Это сочетание как бы соответствует словам "здесь находится". Со-

четание DE мною используется в функции приема, чтобы автоматически выводить в информационную строку позывной вызывающей вас радиостанции. Как только через функцию приема проходит это сочетание, в работу вступает часть функции, начинающаяся с метки `perl`, все следующие после этого сочетания символы функция считает буквами от позывного и выводит эти буквы в информационную строку.

Автоматический вывод позывного вызывающей станции применяется только в моих программах. Ничего подобного в программах других авторов я не встречал.

Листинг 5.30. Продолжение 4

```
// подпрограмма приема символа
void priem(void)
{
    register int dd, kk;
    register int z1,z2,z3,z4;
k1:  if(kbhit()) perek1();
    prt = inport();           // проверяем наличие сигнала
    if(prt == rr) goto k1;    // ждем прихода стартового бита
        delay(skorost);      // выходим на середину стартового бита
    delay(skorost/2);        // сдвигаемся еще на четверть
        dd = 1; kk = 5;      // начальные значения вспомогат. величин
    delay(skorost);          // выходим на 1/4-ю часть 1-го бита
    z1 = inport();           // проверяем наличие сигнала
    delay(skorost/2);        // выходим на 1/2-ю часть 1-го бита
    z2 = inport();           // проверяем наличие сигнала
    if(z2 == z1){goto_xy(0,46); putchar(' ');} // индикация ошибки
    else {goto_xy(0,46); putchar('o');}
        delay(skorost/2);    // выходим на 3/4-ю часть 1-го бита
    z3 = inport();           // проверяем наличие сигнала
    if(z2 == z1) { if(z2 == rr) dd++; } // если z1=z2, принимаем это как
                                   // истину
    else {if(z3 == rr) dd++;} // иначе истинное значение в z3
    dd = dd << 1;           // проводим сдвиг на 1 знак влево
    kk--;                   // уменьшаем значение счетчика на 1
n1:  delay(skorost);        // выходим на 1/4-ю часть 2-го, а затем и
                                   // последующих информационных битов
        z1 = inport();      // проверяем наличие сигнала в этом месте
    delay(skorost/2);        // выходим на 1/2-ю часть 2-го и след. битов
    z2 = inport();           // проверяем наличие сигнала в этом месте
```

```
if(z2 == z1){goto_xy(0,35); putchar(' ');}
else {goto_xy(0,35); putchar('o');}      // индикация ошибки
    delay(skorost/2);                    // выходим на 3/4-ю часть бита
    z3 = inport();                       // проверяем наличие сигнала
if(z2 == z1) { if(z2 == rr) dd++;} // если z1=z2 считаем это истиной
else {if(z3 == rr) dd++;} // иначе истину дает значение z3
    dd = dd << 1;                        // проводим сдвиг на 1 знак влево
    kk--;                                // уменьшаем значение счетчика на 1
    if(kk > 0) goto n1;                  // продолжаем анализ символа
    simw_nom = (dd/2 - 32/2); // получаем номер символа в таблице
    if(simw_nom == 0) {                 // получен символ русского регистра
        reg_ru = 1; reg_la = 0;
        reg_cif = 0; goto vyh;}
    if(simw_nom == 31) {                // получен символ латинского регистра
        reg_la = 1; reg_ru = 0;
        reg_cif = 0; goto vyh;}
    if(simw_nom == 27) {               // получен символ регистра цифр
        reg_ru = 0; reg_la = 0;
        reg_cif = 1; goto vyh;}
if(flag_rus == 1)
    { if(reg_ru == 1) simwol2 = abc_r2[simw_nom];
      if(reg_la == 1) simwol2 = abc_r1[simw_nom];
      if(reg_cif ==1) simwol2 = abc_cif[simw_nom];
    }
else
    { if(reg_ru == 1) simwol2 = abc_l2[simw_nom];
      if(reg_la == 1) simwol2 = abc_l1[simw_nom];
      if(reg_cif == 1) simwol2 = abc_cif[simw_nom];
    }
if(fl_d == 1) goto per1;
if(simwol2 == 'd' || simwol2 == 'D') fl_d = 1;
    rx_read();
vyh: return;
per1: if(fl_e == 1) goto per2;
    if(simwol2 == 'e' || simwol2 == 'E') fl_e = 1;
    else fl_d = 0;
    rx_read();
    return;
per2: if(fl_pus == 1) goto per3;
```



```

        if(simwol2 == ' ') fl_pus = 1;
        else {fl_d = 0; fl_e = 0;}
        rx_read();
        return;
per3: if(simwol2 == ' ') goto per4;
        buf_kor[iz] = simwol2; iz++;
        rx_read();
        return;
per4: buf_kor[iz] = 0; iz++;
        buf_kor[iz] = 0;
        stpcpy(buf_prom, buf_kor);
        goto_xy(17,2); printf("%s      ",buf_prom);
        fl_d = 0; fl_e = 0; fl_pus = 0; iz = 0;
        rx_read();
        return;
    }

// подпрограмма проверки наличия сигнала от приемника
int inport(void)
    {
        int ch;
asm{  mov dx, com_adr2
        in al, dx
        and al,00010000b
        jz wyh1
        jnz wyh2
        }
        wyh1: prt = 0; t_nizk(); return prt;
        wyh2: prt = 1; t_wysok();
        return prt;
    }

// тон высокий
void t_wysok(void)
    {
        goto_xy(0,32); putchar('o');
        goto_xy(0,49); putchar(' ');
        return;
    }

```

```
// тон низкий
void t_nizk(void)
{
goto_xy(0,32); putchar(' ');
goto_xy(0,49); putchar('o');
return;
}
```

В листинге 5.31 описаны функции оформления экрана. Функций много, но все они простые и специального пояснения не требуют. Тем более, что я советую делать оформление экрана своими методами, а не использовать мои, которые трудно назвать наилучшими.

Листинг 5.31. Продолжение 5

```
// заставка начальная
void zastawka(void)
{
int i;
scren_col(3);
cls(0x0e);
draw_border(0,1,23,78);
wind_zw(0x70,2,5,7,74);
wind_zw(0x1e,8,5,14,74);
wind_zw(0xc0,15,5,21,74);
wind_zw(0x5e,5,13,17,66);
draw_border(5,14,17,65);
goto_xy(7,30);
printf(" П Р О Г Р А М М А ");
goto_xy(9,30);
printf(" RTTY_QSO v.1.20 ");
goto_xy(11,20);
printf("для проведения любительской радиосвязи");
goto_xy(12,25);
printf("посредством компьютера IBM PC");
goto_xy(14,18);
printf("Разработана Тяпичевым Геннадием А. (с) RA3XB");
goto_xy(16,32);
printf("апрель 2001 года");
```

```

goto_xy(24,32);
printf("Нажмите <Enter> ");
getch();
}

```

// заставка рабочая – рабочий экран

```

void zastawka2(void)
{
    cls(0x07);
    wind_zw(0x6e,0,0,2,7);
    goto_xy(0,3); puts("CALL:");
    goto_xy(1,3); puts("NAME:");
    goto_xy(2,4); puts("QTH:");
    wind_zw(0x6e,0,56,2,63);
    goto_xy(0,60); puts("RST:");
    goto_xy(1,56); puts("темпер.:");
    goto_xy(2,56); puts("скорость");
    wind_zw(0x5a,0,8,2,25);
    wind_zw(0x5a,0,64,2,79);
    wind_zw(0x5a,17,0,15,79); goto_xy(17,2);
    puts("Принтер:  ОТКЛ      * RTTY_QSQ12, RA3XB, 2001г *
Файл сохранения:  ОТКЛ");
    write_video(0,39,*txt2,0xc);          /* надпись RTTY */
    write_video(2,31,*txt4,0xf);          /* надпись ПРИЕМ */
    wind_zw(0x5b,24,1,24,78); goto_xy(24,1);
    write_video(24,1,*txt5,0x5b);
    wind_zw(0x1e,18,0,23,79);
    wind_zw(0x1d,3,0,16,79); goto_xy(16,0);
    goto_xy(1,74);
    printf("%dГц",ton);
        izm_skor_p();
    goto_xy(0,9); printf("%s",qso.call_sign);
    goto_xy(1,9); printf("%s",qso.name_k);
    goto_xy(2,9); printf("%s",qso.qth);
    goto_xy(0,65); printf("%s",qso.rst_sent);
    goto_xy(1,65); printf("%s",temper);
    return;
}

```

```
// убрать курсор (погасить)
```

```
void cursor_ubrat(void)
{
    union REGS r;
    r.h.ah=1;
    r.h.ch=0x20;
    int86(0x10,&r,&r);
    return;
}
```

```
/* ввести цвет бордюра */
```

```
void bord_col(int num)
{
    union REGS r;
    r.h.ah=0x0b;
    r.h.bh=0;
    r.h.bl=num;
    int86(0x10,&r,&r);
    return;
}
```

```
/* установка курсора */
```

```
void goto_xy(int x,int y)
{
    union REGS r;

    r.h.ah=2; /* функция установки курсора */
    r.h.dl=y; /* координата колонки */
    r.h.dh=x; /* координата строки */
    r.h.bh=0; /* видеостраница */
    int86(0x10,&r,&r);
    return;
}
```

```
/* нарисовать обрамление окна — цветную рамку */
```

```
void draw_border(int startx,int starty,int endx,int endy)
{
    register int i;
```

```

        for(i=startx;i<endx;i++) {
goto_xy(i,starty);
putchar(186);
goto_xy(i,endy);
putchar(186);
}
        for(i=starty;i<endy;i++) {
goto_xy(startx,i);
putchar(205);
goto_xy(endx,i);
putchar(205);
}
goto_xy(startx,starty); putchar(201);
goto_xy(startx,endy ); putchar(187);
goto_xy(endx ,starty); putchar(200);
goto_xy(endx ,endy ); putchar(188);
return;
}

/* вывод строки с определенным атрибутом */
void write_video(int x,int y,char *p,int attrib)
{
    union REGS r;
    register int i,j;
    for(i=y; *p; i++) {
        goto_xy(x,i);
        r.h.ah=9;           /* функция записи символа */
        r.h.bh=0;           /* видеостраница */
        r.x.cx=1;           /* число повторений символа */
        r.h.al=*p++;        /* символ */
        r.h.bl=attrib;      /* атрибут */
        int86(0x10,&r,&r);
    }
    return;
}

/* вывод одного символа на экран в цвете */
void write_simw(int ch,int attrib)
{
    union REGS r;

```

```
        r.h.ah=9;                /* функция вывода символа */
        r.h.bh=0;                /* видеостраница          */
        r.x.cx=1;                /* число повторений символа */
        r.h.al=ch;               /* символ              */
        r.h.bl=attrib;           /* атрибут              */

        int86(0x10,&r,&r);

    }

/* сохранение части экрана */
void save_video(int startx,int endx,int starty,int endy,unsigned int
*buffer)
{
    union REGS r;
    register int i,j;

    for(i=starty;i<endy;i++)
        for(j=startx;j<endx;j++) {
            goto_xy(j,i);
            r.h.ah=8; /* функция чтения символа */
            r.h.bh=0; /* видеостраница          */
            *buffer++ = int86(0x10,&r,&r);
            putchar(' '); /* очистка экрана */
        }
    return;
}

/* восстановление части экрана */
void restore_video(int startx,int endx,int starty,int endy,unsigned int
*buffer)
{
    union REGS r;
    register int i,j;
    for(i=starty;i<endy;i++)
        for(j=startx;j<endx;j++)
        {
            goto_xy(j,i);
            r.h.ah=9; /* функция записи символа */
            r.h.bh=0; /* видеостраница          */
        }
```

```
        r.x.cx=1; /* число повторений символа */
        r.h.al=*buffer++; /* символ */

        r.h.bl=0x1d;
        int86(0x10,&r,&r);
    }
    return;
}

/* очистка экрана */
void cls(int zwet)
{
    union REGS r;
    r.h.ah=6; /* код прокрутки экрана */
    r.h.al=0; /* код очистки экрана */
    r.h.ch=0; /* начальная строка */
    r.h.cl=0; /* начальная колонка */
    r.h.dh=24; /* конечная строка */
    r.h.dl=79; /* конечная колонка */
    r.h.bh=zwet; /* фон и цвет букв */
    int86(0x10,&r,&r);
    return;
}

/* Установка цветового режима экрана */
void scren_col(int rez)
{
    union REGS r;
    r.h.ah=0; /*функция установки режима*/
    r.h.al=rez; /*1 или 3 или другой режим экрана*/
    int86(0x10,&r,&r);
    return;
}

/* Установка активной видеостраницы */
void stran(int num)
{
    union REGS r;
    r.h.ah=5; /*функция выбора страницы*/
```

```
        r.h.al=num; /*номер выбираемой страницы*/
        int86(0x10,&r,&r);
        return;
    }

    /*          Задать окно (цветное)          */
void wind_zw(int attr,int startx,int starty,int x,int y)
    {
        union REGS r;
        r.h.ah=6;
        r.h.al=0;
        r.h.bh=attr; /*цвет фона и букв*/
        r.h.ch=startx;
        r.h.cl=starty;
        r.h.dh=x;
        r.h.dl=y;
        int86(0x10,&r,&r);
        return;
    }

/* Вывести на экран строку с дополнительными атрибутами */
void write_string(int x, int y, int attrib,char *p)

    {
        register int i;
        unsigned char far *v;
        vid_mem =(unsigned char far *)0xb8000000;
        v = vid_mem;
        v += (x*160) + y*2; /* вычислить адрес */
        for(i=y;i<=(y+strlen(p) + 2);i++) {
            *v++ = *p++; /* вывести символ */
            *v++ = attrib;
        }
        return;
    }

/* прочитать строку из видеопамати в буфер */
void read_string(int x,int y,int len, char *p)
```



```

{
    register int i;
    unsigned char far *v;
    vid_mem=(unsigned char far *)0xb8000000;
    v = vid_mem;
    v += (x*160) + y*2; /* вычислить адрес */
    for(i=y;i<=(y+strlen(p) + len);i++) {
        *p++ = *v++; /* вывести символ из памяти в строку*/
        *v++;
    }
    return;
}

```

/* прочитать слово из видеопамати в буфер */

void read_word(int x,int y,int len,char *p)

```

{
    register int i;
    unsigned char far *v;
    vid_mem=(unsigned char far *)0xb8000000;
    v = vid_mem;
    v += (x*160) + y*2; /* вычислить адрес */
    for(i=y;i<=(y+strlen(p) + len);i++) {
        *p++ = *v++; /* вывести символ из памяти в строку*/
        *v++;
    }
    return;
}

```

// Заполняет строку нулями

void zabitie(int x, int y, int attrib,int len)

```

{
    register int i;
    unsigned char far *v;
    vid_mem=(unsigned char far *)0xb8000000;
    v = vid_mem;
    v += (x*160) + y*2; /* вычислить адрес */

```

```
        for(i=y;i<=(y+len);i++) {
            *v++ = 0; /* вывести символ */
            *v++ = attrib;
        }
    return;
}

/* Вывести символы с определенными атрибутами */
void write_char(int xx, int yy, char ch, int attrib)
{
    register int i;
    unsigned char far *v;
    vid_mem=(unsigned char far *)0xb8000000;
    ch = ch_n;
    v =vid_mem;
    v += (xx*160) + yy*2;
    if(ch == 13 || ch == 3) return;
    *v++ = ch; /* вывести символ */
    *v++ = attrib; /* вывести атрибуты */
    yy++;
    return;
}

// Создать окно для подтверждения выхода в DOS
void err_wind(int x,int y,int endx,int endy)
{
    unsigned int *p;
    /* размещение памяти для видеобуфера */
    p=(unsigned int*)malloc((endx-x+1)*(endy-y+1));
    if(!p) exit(1); /* Вы можете здесь сами обработать ошибку */
    /* сохранение части экрана */
    save_video(x-1,endx,y-1,endy+1,p);
    wind_zw(0x4a,x-1,y-1,endx-1,endy);
    draw_border(x-1,y-1,endx-1,endy);
    /* высвечивание меню на своем месте */
    goto_xy(x+1,y+20);
    printf("%s",txt7[0]);
    ch_k = getch();
    /* восстановление части экрана */
}
```

```

    restore_video(x-1, endx, y-1, endy+1, p);
    free(p);
    return;
}

/* Вывести на экран таблицу help */
void hlp_wind(int count, int x, int y, int border)
{
    register int i, j;
    int endx, endy, choice, len;
    unsigned int *p;
    if((x>24) || (x<0) || (y>79) || (y<0)) {
        printf(" выход за пределы экрана");
        return;
    }
    /* вычисление размеров */
    len=0;
    for(i=0; i<count; i++)
        if(strlen(help1[i]) > len) len=strlen(help1[i]);
    endy=len+2+y;
    endx=count+1+x;

    if((endx+1>24) || (endy+1>79)) {
        printf(" выход за пределы экрана");
        return;
    }
    /* размещение памяти для видеобуфера */
    p=(unsigned int*)malloc((endx-x+1)*(endy-y+1));
    if(!p) exit(1); /* Вы можете здесь сами обработать ошибку */
    /* сохранение части экрана */
    save_video(x-1, endx, y-1, endy+1, p);
    wind_zw(0x4a, x-1, y-1, endx-1, endy);
    if(border) draw_border(x-1, y-1, endx-1, endy);
    /* высвечивание меню на своем месте */
    display_help(x, y+1, count);
    sleep(10);
    /* восстановление части экрана */
    restore_video(x-1, endx, y-1, endy+1, p);

```

```
    free(p);
    return;
}

/* высвечивание help на своем месте */
void display_help(int x,int y,int count)
{
    register int i;
    for(i=0;i<count;i++,x++) {
        goto_xy(x,y);
        cout<<help1[i];
    }
    return;
}

// выдать на экран help2
void hlp2_wind(int border)
{
    register int i;
    int x,y,endx, endy,x1,y1;
    unsigned int *p;
    x = 4; y = 10; endx =16 ; endy = 69;
    x1 = x; y1 = y;
    /* размещение памяти для видеобуфера */
    p=(unsigned int*)malloc((endx-x+1)*(endy-y+1));
    if(!p) exit(1); /* Вы можете здесь сами обработать ошибку */
    /* сохранение части экрана */
    save_video(x-1,endx,y-1,endy+1,p);
    wind_zw(0x4a,x-1,y-1,endx-1,endy);
    if(border) draw_border(x-1,y-1,endx-1,endy);
    /* высвечивание меню на своем месте */
    for(i=0; i<=10; i++,x1++) {
        goto_xy(x1,y1+1);
        cout<<help2[i];
    }
    sleep(10);
    /* восстановление части экрана */
    restore_video(x-1,endx,y-1,endy+1,p);
```

```
    free(p);
    return;
}

// выдать на экран help3
void hlp3_wind(int border)
{
    register int i;
    int x, y, endx, endy, x1, y1;
    unsigned int *p;
    x = 4; y = 10; endx = 12; endy = 69;
    x1 = x; y1 = y;
    /* размещение памяти для видеобуфера */
    p=(unsigned int*)malloc((endx-x+1)*(endy-y+1));
    if(!p) exit(1); /* Вы можете здесь сами обработать ошибку */
    /* сохранение части экрана */
    save_video(x-1, endx, y-1, endy+1, p);
    wind_zw(0x4a, x-1, y-1, endx-1, endy);
    if(border) draw_border(x-1, y-1, endx-1, endy);
    /* высвечивание меню на своем месте */
    for(i=0; i<=6; i++, x1++) {
        goto_xy(x1, y1+1);
        cout<<help3[i];
    }
    sleep(10);
    /* восстановление части экрана */
    restore_video(x-1, endx, y-1, endy+1, p);
    free(p);
    return;
}

// выдать на экран help4
void hlp4_wind(int border)
{
    register int i;
    int x, y, endx, endy, x1, y1;
    unsigned int *p;
    x = 4; y = 10; endx = 16; endy = 69;
    x1 = x; y1 = y;
```

```
/* размещение памяти для видеобуфера */
p=(unsigned int*)malloc((endx-x+1)*(endy-y+1));
if(!p) exit(1); /* Вы можете здесь сами обработать ошибку */
/* сохранение части экрана */
save_video(x-1,endx,y-1,endy+1,p);
wind_zw(0x4a,x-1,y-1,endx-1,endy);
if(border) draw_border(x-1,y-1,endx-1,endy);
/* высвечивание меню на своем месте */
for(i=0; i<=10; i++,x1++) {
    goto_xy(x1,y1+1);
    cout<<help4[i];
}
sleep(10);
/* восстановление части экрана */
restore_video(x-1,endx,y-1,endy+1,p);
free(p);
return;
}
```

В листинге 5.32 описаны функции, используемые при передаче символа телетайпом. Основная функция берет из ячейки под названием `simwol` код символа, который должен быть передан телетайпом, и присваивает ему имя `ch_i`. После этого начинается исследование взятого для передачи символа. Если код символа больше или равен 65, но в то же время он меньше или равен 93, то начинается просмотр кодировочной таблицы `abc_11` (если код символа другой, то используются другие таблицы). В этой таблице ведется поиск символа с точно таким же кодом. Когда такой символ в таблице найден, функцией запоминается номер места, которое занимает найденный символ и присваивается этому номеру имя `ch_z`. Поскольку таблица `abc_11` содержит только латинские символы, на передачу идет код латинского регистра, если перед этим шли символы только другого регистра (например, русские). Затем `ch_z` поступает к метке `gg`, начиная с которой проводится его исследование на наличие в коде этого символа токовых и бестоковых посылок. Следует пояснить, что кодовые таблицы составлены таким образом, что номер места, занимаемого в таблице каким-либо символом, является телетайпным кодом этого символа. Так что каждому символу в таблице выделяется только одно место, номер которого соответствует телетайпному коду этого символа. Начиная с метки `gg` функция пять раз выполняет сдвиг символа `ch_z` и каждый раз обрабатывает соответствующим образом каждую токовую или бестоковую посылку. При этом каждому виду посылок соответствуют свои сигналы для управления модемом.

После того как код одного символа прошел на передачу, функция берет для передачи следующий символ и так до тех пор, пока все символы из буфера не будут переданы через модем в эфир.

В зависимости от кода символа в работе может находиться либо таблица `abc_l1`, либо таблица `abc_l2`, либо `abc_r1`, либо `abc_r2`, либо `abc_cif`. Таблицы `abc_l1` и `abc_l2` содержат символы латинского алфавита, таблицы `abc_r1` и `abc_r2` содержат символы русского алфавита, а таблица `abc_cif` содержит цифры, символы пунктуации и некоторые символы русского алфавита. Перед передачей группы символов какого-либо регистра должен быть передан в эфир код этого регистра.

Подробнее о кодировании телетайпного сигнала следует читать соответствующий материал в *главе 2*.

Листинг 5.32. Продолжение 6

```
// подпрограмма передачи символа телетайпом
void pered_simw(int simwol)
{
    int count1;
    register int dd, kk;
    int i;
    unsigned int j;
    unsigned int ch_i;
    unsigned int ch_z;
    unsigned char bb;
kal:
    ch_i = simwol;
    if(ch_i <= 9 || ch_i == 27) return;
    if(ch_i == 10) ch_z = 2;
    if(ch_i == 13) ch_z = 8;
    if(ch_i >= 65 && ch_i <= 93) {
        for(j=0; j<=32; j++)
        {
            bb = abc_l1[0+j];
            if(bb == simwol) {ch_z = j; break;}
        }
    }
    if(reg_la == 0) {pered_la_reg(); pered_la_reg();}
    }
    if(ch_i >= 97 && ch_i <=125) {
        for(j=0; j<=32; j++)
        {
            bb = abc_l2[0+j];
```

```
        if(bb == simwol) {ch_z = j; break;}
    }
    if(reg_ru == 0) {pered_ru_reg(); pered_ru_reg();}
    }
    if(ch_i >=128 && ch_i <= 159) {
        for(j=0;j<=32;j++)
            { bb = abc_r1[0+j];
              if(bb == simwol) {ch_z = j; break;}
            }
        if(reg_la == 0) {pered_la_reg(); pered_la_reg();}
    }
    if(ch_i >=160 && ch_i <= 175) {
        for(j=0;j<=32;j++)
            { bb = abc_r2[0+j];
              if(bb == simwol) {ch_z = j; break;}
            }
        if(reg_ru == 0) {pered_ru_reg(); pered_ru_reg();}
    }
    if(ch_i >= 224 && ch_i <= 230 || ch_i == 235 || ch_i == 236) {
        for(j=0;j<=32;j++)
            {bb = abc_r2[0+j];
              if(bb == simwol) {ch_z = j; break;}
            }
        if(reg_ru == 0) {pered_ru_reg(); pered_ru_reg();}
    }
    if(ch_i >= 32 && ch_i <= 63) {
        for(j=0;j<=32;j++)
            { bb = abc_cif[0+j];
              if(bb == simwol) {ch_z = j; break;}
            }
        if(reg_cif == 0) {pered_cif_reg(); pered_cif_reg();}
    }
    if(ch_i == 231 || ch_i== 232 || ch_i == 237 || ch_i == 238 || ch_i ==
239) {
        for(j=0;j<=32;j++)
            { bb = abc_cif[0+j];
              if(bb == simwol) {ch_z = j; break;}
            }
        if(reg_cif == 0) {pered_cif_reg(); pered_cif_reg();}
    }
}
```



```
gg:
    zwuk();
    dd = 6;
kt: dd--;
    if(dd == 0) {pauza(); delay(skorost); return;}
    ch_z = ch_z << 1;
    kk = ch_z & 0x20;
    if(kk != 0) {pauza(); goto kt;}
        else zwuk();
    goto kt;
}
```

// звуковая посылка высокого тона

```
void pauza(void)
{
    int j;
    if(flag_zw == 1) sound(ton);
        else nosound();
    delay(skorost);
    delay(skorost);
    return;
}
```

// звуковая посылка низкого тона

```
void zwuk(void)
{
    int j;
    if(flag_zw == 1) sound(ton+200);
        else nosound();
    if(rew_per == 1) ust_dtr_rts();
    else ust_rts();
        delay(skorost);
        delay(skorost);
        if(flag_zw == 1) sound(ton);
        else nosound();
    if(rew_per == 1) ust_rts();
    else ust_dtr_rts();
    return;
}
```

```
// передать (русский) символ регистра строчных букв
void pered_ru_reg(void)
{
    unsigned int ch_z = 0x0;
    int dd, kk;
        zwuk();
        dd = 6;
    kt: dd--;
        if(dd == 0) {pauza(); delay(skorost);
        reg_ru = 1; reg_la =0; reg_cif = 0; return;}
        ch_z = ch_z << 1;
        kk = ch_z & 0x20;
        if(kk != 0) {pauza(); goto kt;}
            else    zwuk();
            goto kt;
    }

// передать символ (латинского) регистра заглавных букв
void pered_la_reg(void)
{
    unsigned int ch_z = 0x1f;
    int dd, kk;
        zwuk();
        dd = 6;
    kt: dd--;
        if(dd == 0) {pauza(); delay(skorost);
        reg_la = 1; reg_cif =0; reg_ru =0; return;}
        ch_z = ch_z << 1;
        kk = ch_z & 0x20;
        if(kk != 0) {pauza(); goto kt;}
            else    zwuk();
            goto kt;
    }

// передать символ регистра цифр
void pered_cif_reg(void)
{
    unsigned int ch_z = 0x1b;
    int dd, kk;
```

```

    zwuk();
    dd = 6;
kt: dd--;
    if(dd == 0) {pauza(); delay(skorost);
    reg_cif = 1; reg_la = 0; reg_ru = 0; return;}
    ch_z = ch_z << 1;
    kk = ch_z & 0x20;
    if(kk != 0) {pauza(); goto kt;}
        else    zwuk();
    goto kt;
}

```

В листинге 5.33 выполнено описание большого числа различных вспомогательных функций. Это функции по управлению звуком компьютерного динамика, функции для работы с буферами, функции для передачи отдельных небольших фраз. Все эти функции достаточно простые и специального пояснения не требуют.

Листинг 5.33. Продолжение 7

```

// переключ. звука
void perek_zw(void)
{
    if(flag_zw == 1) {
        flag_zw = 0;
        goto_xy(2,78); putchar(' ');
        return;
    }
    flag_zw = 1; goto_xy(2,78); putchar(nota);
}

// п/программа передачи (начало)
void peredacha(void)
{
    write_video(2,31,*txt3,0xf);
    bord_col(0x3);
    ust_dtr_rts();
    goto_xy(18,1);
    wyzow();
}

```

```
        sbros_dtr_rts(); nosound();
write_video(2,31,*txt4,0xf);
bord_col(0);
}
```

// подпрограмма выхода из передачи (концовка)

```
void wyhod3(void)
{
    write_video(2,31,*txt3,0xf);
    bord_col(0x3);
        ust_dtr_rts();
    goto_xy(18,1);
wyhod();
        sbros_dtr_rts(); nosound();
    write_video(2,31,*txt4,0xf);
    bord_col(0);
}
```

// чтение позиции курсора

```
void getxy()
{
    int i;
    union REGS regs;
    regs.h.ah = 3; // читает поз. курсора
    regs.h.bh = 0;
    int86(0x10,&regs,&regs);
    i = regs.x.dx;
    yyl = 1 + (i & 0xff);
    xxl = 1 + (i >> 8);
    return;
}
```

// выдает на экран (в нужное место) текущее время

```
void get_time(/*int date_flag*/void)
{
    struct tm *time_now;
        time_t secs_now;
        char str[80];
        tzset();
```

```

        time(&secs_now);
        time_now = localtime(&secs_now);
        strftime(str,80,
//      "    ****   Сегодня %d-%b-%Yг    ****   Местное время %H:%M
****",time_now);
        "%H:%M           %d-%m-%Y",time_now);
        goto_xy(1,27);
        puts(str); goto_xy(3,0);
    }

```

/* Скроллинг в окне */

```

void scr_scrup(int attr, int frow, int fcol, int trow, int tcol)
{
    union REGS regs;
    regs.h.ah = 6;
    regs.h.al = 1;
    regs.h.bh = attr;          /* обычные видеоатрибуты */
    regs.h.ch = frow;
    regs.h.cl = fcol;
    regs.h.dh = trow;
    regs.h.dl = tcol;
    int86(0x10, &regs, &regs);
}

```

// Очистить окно приема

```

void cls_rx()
{
    wind_zw(0x1d,3,0,16,79);
    goto_xy(16,0);
}

```

// Очистить окно передачи

```

void cls_tx()
{
    wind_zw(0x1e,18,0,23,79);
    goto_xy(23,0);
}

```

/* Копирование данных из файла в буфер buf1[] */

```

void copy_buf(int count)

```

```
{
    int i, letter;
    if(count ==1) goto wyh1;
    if(count ==2) goto wyh2;
    if(count ==3) goto wyh3;
    if(count ==4) goto wyh4;
    if(count ==5) goto wyh5;
    if(count ==6) goto wyh6;
    if(count ==7) goto wyh7;
    if(count ==8) goto wyh8;
    if(count ==9) goto wyh9;
    if(count ==10) goto wyh10;
    if(count ==11) goto wyh11;
wyh1:    if((fp = fopen("eqp.msg", "r")) == NULL)
    {
        printf("Невозможно открыть файл eqp.msg");
        getchar();
        exit(0);
    }
    goto wyh20;
wyh2:    if((fp = fopen("rpt.msg", "r")) == NULL)
    {
        printf("Невозможно открыть файл rpt.msg");
        getchar();
        exit(0);
    }
    goto wyh20;
wyh3:    if((fp = fopen("my_name.msg", "r")) == NULL)
    {
        printf("Невозможно открыть файл my_name.msg");
        getchar();
        exit(0);
    }
    goto wyh20;
wyh4:    if((fp = fopen("konec.msg", "r")) == NULL)
    {
        printf("Невозможно открыть файл konec.msg");
        getchar();
        exit(0);
    }
}
```

```
        goto wyh20;
wyh5:    if((fp = fopen("qrz.msg","r")) == NULL)
        {
            printf("Невозможно открыть файл qrz.msg");
            getchar();
            exit(0);
        }
        goto wyh20;
wyh6:    if((fp = fopen("cq.msg","r")) == NULL)
        {
            printf("Невозможно открыть файл cq.msg");
            getchar();
            exit(0);
        }
        goto wyh20;
wyh7:    if((fp = fopen("f5.msg","r")) == NULL)
        {
            printf("Невозможно открыть файл f5.msg");
            getchar();
            exit(0);
        }
        goto wyh20;
wyh8:    if((fp = fopen("f6.msg","r")) == NULL)
        {
            printf("Невозможно открыть файл f6.msg");
            getchar();
            exit(0);
        }
        goto wyh20;
wyh9:    if((fp = fopen("f7.msg","r")) == NULL)
        {
            printf("Невозможно открыть файл f7.msg");
            getchar();
            exit(0);
        }
        goto wyh20;
wyh10:   if((fp = fopen("f8.msg","r")) == NULL)
        {
            printf("Невозможно открыть файл f8.msg");
```

```

        getchar();
        exit(0);
    }
    goto wyh20;
wyh11:    if((fp = fopen("f9.msg","r")) == NULL)
    {
        printf("Невозможно открыть файл f9.msg");
        getchar();
        exit(0);
    }
wyh20:    for(i=0;i<400;i++)
    {
        if((letter = fgetc(fp)) != EOF)
        {
            buf1[i] = letter;
        }
    }
    fclose(fp);
    return;
}

// выдать из буфера buf0 на передачу
void send_buf0(int x,int y,int len,int attrib)
{
    int i;
    unsigned int ch;
    xx = x;
    yy = y;
    for(i=0;i<len;i++)
    {
        ch = buf0[i];
        if(ch == NULL) break;
        if(ch == 0x0A | ch == 0x0D) {xx++; yy = 0;
            if(xx>=24) scr_scrup(0x1e,18,0,23,79);
        }

        simwol = buf0[i];
        goto_xy(xx,yy);
        write_simw(ch,attrib);
        if(kbhit()) {

```



```

        if((ch_k = getch()) == 0x1b)
            break;}
        pered_simw(simwol);
        yy++;
        if(yy > 79)
            {yy = 1; xx++;}
        if(xx == 23 & yy>= 79)
            {
                scr_scrup(0x1e,18,0,23,79);
                xx = 23; yy = 1;
            }
    }
    cls_tx();
}

// выдать из буфера buf2 на передачу
void send_buf2(int x,int y,int len,int attrib)
{
    int i;
    unsigned int ch;
    xx = x;
    yy = y;
    for(i=0;i<len;i++)
    {
        ch = buf2[i];
        if(ch == NULL) break;
        if(ch ==0x0A | ch == 0x0D) {xx++; yy = 0;
                                if(xx>=24) scr_scrup(0x1e,18,0,23,79);
                                xx=23; yy=1;
                                }

        simwol = buf2[i];
        goto_xy(xx,yy);
        write_simw(ch,attrib);
        if(kbhit()) {
            if((ch_k = getch()) == 0x1b)
                break;}
            pered_simw(simwol);
            yy++;
            if(yy > 79)

```

```
        {yy = 1; xx++;}
    if(xx == 23 & yy >= 79)
    {
        scr_scrup(0x1e,18,0,23,79);
        xx = 23; yy = 1;
    }
}

cls_tx();
}

// выдать из буфера buf0 на передачу
void send_buf1(int x,int y,int len,int attrib)
{
    int i;
    unsigned int ch;
    xx = x;
    yy = y;
    for(i=0;i<len;i++)
    {
        ch = buf1[i];
        if(ch == NULL) break;
        if(ch == 0x0A | ch == 0x0D) {xx++; yy = 0;
                                   if(xx>=24) scr_scrup(0x1e,18,0,23,79);
                                   }

        simwol = buf1[i];
        goto_xy(xx,yy);
        write_simw(ch,attrib);
        if(kbhit()) {
            if((ch_k = getch()) == 0x1b)
                break;}
        pered_simw(simwol);
        yy++;
        if(yy > 79)
            {yy = 1; xx++;}
        if(xx == 23 & yy >= 79)
        {
            scr_scrup(0x1e,18,0,23,79);
            xx = 23; yy = 1;
        }
    }
}
```

```

        }
        cls_tx();
    }

// вывести текст из буфера на экран
void co_buf(int x,int y,int len,int count,int attrib)
{
    int i;
    unsigned int ch;
    xx = x;
    yy = y;
    for(i=0;i<len;i++)
    {
        if(count == 1) ch = buf1[i];
        if(count == 0) ch = buf0[i];
        if(ch == NULL) break;
        if(ch == 0x0A | ch == 0x0D) {xx++; yy = 0;
                                   if(xx>=24) scr_scrup(0x1e,18,0,23,79);
                                   }

        simwol = buf1[i];
        goto_xy(xx,yy);
        write_simw(ch,attrib);
        yy++;
        if(yy > 79)
            {yy = 1; xx++;}
        if(xx == 24 & yy >= 79)
            {
                scr_scrup(0x1e,18,0,23,79);
                xx = 23; yy = 1;
            }
    }
}

/* начальная фраза QSO */
void wyzow(void)
{
    goto_xy(18,1);
    if(qso.call_sign[0] == NULL) {promt(txt20[0],19,10,23,69);
    goto_xy(18,1); return;

```

```
    }
else
printf(" RYRYRYRY %s de %s ",qso.call_sign,my_call);
read_string(18,1,33,buf0);
goto_xy(18,1);
send_buf0(18,1,33,0x1d);
cls_buf0(80);
}

/* начальная фраза QSO вариант для текста по <Alt>—<F5> */
void wyzowl(void)
{
goto_xy(18,1);
if(qso.call_sign[0] == NULL) {prompt(txt20[0],19,10,23,69);
goto_xy(18,1); return;
}
else
printf(" RYRYRYRY %s de %s — HELLO DR OM — UR RST IS %s %s ",
qso.call_sign, my_call,qso.rst_sent,qso.rst_sent);
read_string(18,1,64,buf0);
goto_xy(18,1);
send_buf0(18,1,64,0x1d);
cls_buf0(80);
}

/* начальная фраза QSO второй вариант*/
void wyzow2(void)
{
goto_xy(18,1);
if(qso.call_sign[0] == NULL) prompt(txt20[0],19,10,23,69);
if(qso.name_k[0] == NULL) { prompt(txt21[0],19,10,23,69);
goto_xy(18,1); return;
}
else
printf(" RYRYRYRY %s de %s — R OK DR
%s",qso.call_sign,my_call,qso.name_k);
read_string(18,1,48,buf0);
goto_xy(18,1);
send_buf0(18,1,48,0x1d);
```

```

        cls_buf0(80);
    }

/* конечная фраза QSO */
void wyhod(void)
{
    goto_xy(18,1);
    if(qso.call_sign[0] == NULL) {promt(txt20[0],19,10,23,69);
        goto_xy(18,1); return;}
    else
        printf(" %s de %s pse k ",qso.call_sign,my_call);
        read_string(18,1,25,buf0);
        goto_xy(18,1);
        send_buf0(18,1,25,0x1d);
        cls_tx();
        cls_buf0(80);
}

/* конечная фраза QSO вариант 2*/
void wyhod2(void)
{
    goto_xy(18,1);
    if(qso.call_sign[0] == NULL) {promt(txt20[0],19,10,23,69); go-
to_xy(18,1);}
    else
        printf(" %s de %s k",qso.call_sign,my_call);
        read_string(18,1,20,buf0);
        goto_xy(18,1);
        send_buf0(18,1,20,0x1d);
        cls_tx();
        cls_buf0(80);
}

// Создать окно подсказки
void promt(char *text,int x,int y, int endx,int endy)
{
    unsigned int *p;
    /* размещение памяти для видеобуфера */
    p=(unsigned int*)malloc((endx-x+1)*(endy-y+1));

```

```
    if(!p) exit(1); /* Вы можете здесь сами обработать ошибку */
/* сохранение части экрана */
    save_video(x-1, endx, y-1, endy+1, p);
wind_zw(0x4a, x-1, y-1, endx-1, endy);
    draw_border(x-1, y-1, endx-1, endy);
/* высвечивание меню на своем месте */
    goto_xy(x+1, y+13);
    printf("%s", text);
    sleep(1);
/* восстановление части экрана */
    restore_video(x-1, endx, y-1, endy+1, p);
    free(p);
    return;
}

// очистка буфера buf1
void cls_buf1(int len)
{
    int i;
    for(i=0; i<len; i++)
        buf1[i]=0;
}

// очистка буфера buf0
void cls_buf0(int len)
{
    int i;
    for(i=0; i<len; i++)
        buf0[i]=0;
}

// очистка буфера buf2
void cls_buf2(int len)
{
    int i;
    for(i=0; i<len; i++)
        buf2[i]=0;
}
```

В листинге 5.34 приведено описание функции для организации многих различных переключений при работе на передачу. Далее описываются функции, задействованные при этих переключениях. Функций довольно много, но все они очень простые и специальных пояснений, по моему мнению, не требуют.

Листинг 5.34. Продолжение 8

```
/* переключения при передаче */
int exec_cmd(int ch_k)
{
    int i, j;
    get_time();
    switch (ch_k) {
        case F4      : rewers_pr();
                       break;
        case F5      : rewers_pered();
                       break;
        case ALT_F2   : alt_f2();
                       break;
        case ALT_F3   : alt_f3();
                       break;
        case ALT_F4   : alt_f4();
                       break;
        case ALT_F5   : alt_f5();
                       break;
        case ALT_F6   : alt_f6();
                       break;
        case ALT_F7   : alt_f7();
                       break;
        case ALT_F8   : alt_f8();
                       break;
        case ALT_F9   : alt_f9();
                       break;
        case ALT_F10  : alt_f10();
                       break;
        case SHF_F2   : izm_ton1();
                       break;
        case SHF_F3   : izm_ton();
                       break;
    }
```

```
        case CTL_F5 : ct_f5();
                    break;
        case CTL_F6 : ct_f6();
                    break;
        case CTL_F7 : ct_f7();
                    break;
        case CTL_F8 : ct_f8();
                    break;
        case CTL_F9 : ct_f9();
                    break;
        case F2      : uwel_skor();
                    break;
        case F3      : umen_skor();
                    break;
        case 199     : peredacha();           //<Home>
                    break;                   //<End>
        case 207     : wyhod3();
                    break;
        case PGUP    : rus_lat();
                    break;
        case PGDN    : stpcpy(qso.call_sign, buf_prom);
                    locate();
                    break;
        default     : break;
    }
    return(0);
}
```

// ОБЩИЙ ВЫЗОВ

```
void alt_f2(void)
{
    if(rew_per == 1)  ust_rts();
    else ust_dtr_rts();
        write_video(2,31,*txt3,0xf);
        bord_col(0x3);
        goto_xy(18,1);
        copy_buf(6);
        co_buf(18,1,400,1,0x1e);
}
```



```
send_buf1(18,1,400,0x1d);
    sbros_dtr_rts(); nosound();
    write_video(2,31,*txt4,0xf);
bord_col(0);
cls_buf1(400);
}

// выдать QRZ
void alt_f3(void)
{
    write_video(2,31,*txt3,0xf);
    bord_col(0x3);
    goto_xy(18,1);
    ust_rts();
copy_buf(5);
    co_buf(18,1,400,1,0x1e);
send_buf1(18,1,400,0x1d);
    sbros_dtr_rts(); nosound();
    write_video(2,31,*txt4,0xf);
    bord_col(0);
    cls_buf1(400);
}

// копировать буфер
void copy_buf_fl(int count)
{
    int i,letter;
    for(i=0;i<count;i++)
    {
        if((letter = fgetc(fp)) != EOF)
        {
            buf2[i] = letter;
        }
    }
    fclose(fp);
    return;
}

// вызов определенного корреспондента
void alt_f4(void)
```

```

{
goto_xy(18,1);
if(qso.call_sign[0] == NULL) {prompt(txt20[0],19,10,23,69);
                             goto_xy(18,1); return;
                             }

else
bord_col(0x3);
write_video(2,31,*txt3,0xf); goto_xy(18,1);
printf(" %s %s de %s %s %s PSE K",qso.call_sign,qso.call_sign,
        my_call,my_call,my_call);

read_string(18,1,45,buf0);
goto_xy(18,1);
    ust_rts();
send_buf0(18,1,45,0x1d);
    sbros_dtr_rts(); nosound();
    write_video(2,31,*txt4,0xf);
bord_col(0);
cls_tx();
cls_buf0(80);
}

// начало qso
void alt_f5(void)
{
write_video(2,31,*txt3,0xf);
bord_col(0x3);
    ust_rts();
goto_xy(18,1);
wyzowl();
if(qso.call_sign[0] == NULL) { bord_col(0);
write_video(2,31,*txt4,0xf);
    sbros_dtr_rts();nosound(); return;}
copy_buf(3);
co_buf(18,1,400,1,0x1e);
send_buf1(18,1,400,0x1d);
wyhod();
write_video(2,31,*txt4,0xf);
bord_col(0);
    sbros_dtr_rts();          nosound();

```

```
    cls_buf1(400);
}

// об аппаратуре
void alt_f6(void)
{
    write_video(2,31,*txt3,0xf);
    bord_col(0x3);
    ust_rts();
    goto_xy(18,1);
    wyzow2();
    if(qso.call_sign[0] == NULL) { bord_col(0);
    write_video(2,31,*txt4,0xf);
        sbros_dtr_rts();nosound(); return;}
    if(qso.name_k[0] == NULL) { bord_col(0);
    write_video(2,32,*txt4,0xf);
        sbros_dtr_rts(); nosound(); return;}
    copy_buf(1);
    co_buf(18,1,400,1,0x1e);
    send_buf1(18,1,400,0x1d);
    wyhod();
        sbros_dtr_rts();          nosound();
    write_video(2,31,*txt4,0xf);
    bord_col(0);
    cls_buf1(400);
}

// концовка qso
void alt_f7(void)
{
    write_video(2,31,*txt3,0xf);
    bord_col(0x3);
    ust_dtr_rts();
    goto_xy(18,1);
    wyzow2();
    if(qso.call_sign[0] == NULL) { bord_col(0);
    write_video(2,31,*txt4,0xf);
        sbros_dtr_rts(); nosound(); return;}
    copy_buf(4);
```

```
co_buf(18,1,400,1,0x1e);
send_buf1(18,1,400,0x1d);
wyhod2();

    sbros_dtr_rts();          nosound();
    write_video(2,32,*txt4,0xf);
    bord_col(0);
    cls_buf1(400);
}

// работать с клавиатуры
void alt_f8(void)
{
    write_video(2,31,*txt3,0xf);
    bord_col(0x3);
    ust_rts();
    goto_xy(18,1);
    wyczow2();
    goto_xy(18,1);
    xx=18; yy=1;
    do{
        do{
            ch_n = getch();
            if(ch_n == 13) goto wyh1;
            write_char(xx,yy,ch_n,0x1e);
            yy++; len++;
            if(ch_n == 32)
            {
                read_word(18,1,len,buf0);
                xx=18; yy=1;
            }
            if(ch_n == 27) exit(0);
        }
        while(ch_n != 32);
    }
    send_buf0(18,1,len,0x1d);
    xx=18; yy=1; goto_xy(18,1); len = 1;
}
while(ch_n != 13 || ch_n != 9);
wyh1:
    wyhod2();
    sbros_dtr_rts(); nosound();
```

```
write_video(2,31,*txt4,0xf);
bord_col(0);
cls_buf1(400);
}
```

// просьба повторить имя и qth

```
void alt_f9(void)
{
    write_video(2,31,*txt3,0xf);
    bord_col(0x3);
        ust_dtr_rts();
    goto_xy(18,1);
    wyzow();
    if(qso.call_sign[0] == NULL) { bord_col(0);
    write_video(2,32,*txt4,0xf);
        sbros_dtr_rts(); nosound(); return;}
    copy_buf(2);
    co_buf(18,1,400,1,0x1e);
    send_buf1(18,1,400,0x1d);
    bord_col(0);
    wyhod();
    write_video(2,31,*txt4,0xf);
        sbros_dtr_rts();          nosound();
    bord_col(0);
    cls_buf1(400);
}
```

// передать ASCII файл

```
void alt_f10(void)
{
    char filename[15];
    promt(txt22[0],19,10,23,69);
    goto_xy(18,1);
    gets(filename);
    if((fp = fopen(filename, "r")) == NULL)
    {
        printf("Невозможно открыть файл %s\n",filename);
        getchar();
        exit(0);
    }
}
```

```
write_video(2,31,*txt3,0xf);
bord_col(0x3);
    ust_dtr_rts();
copy_buf_fl(4000);
send_buf2(23,1,4000,0x1e);
write_video(2,31,*txt4,0xf);
    sbros_dtr_rts();    nosound();
    bord_col(0);
    cls_buf2(4000);
}
```

// общая подпрограмма для Ctrl_Fn qso

```
void ctrl_msg(void)
{
    write_video(2,31,*txt3,0xf);
    bord_col(0x3);
        ust_rts();
    goto_xy(18,1);
    wyzow();
    if(qso.call_sign[0] == NULL) { bord_col(0);
    write_video(2,31,*txt4,0xf);
        return;}
    copy_buf(count);
    co_buf(18,1,400,1,0x1e);
    send_buf1(18,1,400,0x1d);
    wyhod2();
    write_video(2,31,*txt4,0xf);
        sbros_dtr_rts();
    bord_col(0);
    cls_buf1(400);
}
```

// выдать на передачу текст f5.msg

```
void ct_f5(void)
{
    count = 7;
    ctrl_msg();
}
```

```
// выдать на передачу текст f6.msg
```

```
void ct_f6(void)
{
    count = 8;
    ctrl_msg();
}
```

```
// выдать на передачу текст f7.msg
```

```
void ct_f7(void)
{
    count = 9;
    ctrl_msg();
}
```

```
// выдать на передачу текст f8.msg
```

```
void ct_f8(void)
{
    count = 10;
    ctrl_msg();
}
```

```
// выдать на передачу текст f9.msg
```

```
void ct_f9(void)
{
    count = 11;
    ctrl_msg();
}
```

```
// увеличение скорости передачи
```

```
void uwel_skor(void)
{
    static int wsp;
    getxy();      // запомнить координаты курсора в xx1 и yy1
    wsp = skorost -1;
    skorost = wsp;
    if(skorost < 5) skorost =11;
    izm_skor_p();
    goto_xy(xx1-1,yy1);
    return;
}
```

// уменьшение скорости передачи

```
void umen_skor(void)
{
    static int wsp;
    getxy();      // запомнить координаты курсора в xx1 и yy1
    wsp = skorost + 1;
    skorost = wsp;
    if(skorost > 11) skorost = 5;
    izm_skor_p();
    goto_xy(xx1-1,yy1);
    return ;
}
```

// изменение скорости передачи

```
void izm_skor_p(void)
{
    goto_xy(2,65);
    printf("      ");
    goto_xy(2,65);
    if(skorost ==11) puts("45");
    if(skorost ==10) puts("50");
    if(skorost ==9) puts("55");
    if(skorost ==8) puts("62");
    if(skorost ==7) puts("71");
    if(skorost ==6) puts("83");
    if(skorost ==5) puts("100");
    return;
}
```

// изменение тона сигнала в динамике — понижение

```
void izm_ton(void)
{
    static int wsp;
    getxy();      // запомнить координаты курсора в xx1 и yy1
    wsp = ton - 100;
    ton = wsp;
    if(ton < 400) ton =1600;
    izm_skor_p();
    goto_xy(xx1-1,yy1);
}
```



```
goto_xy(1,74);
    puts("      ");
goto_xy(1,74);
printf("%dГц",ton);
    return;
}

// изменение тона сигнала в динамике — повышение
void izm_ton1(void)
{
static int wsp;
    getxy(); // запомнить координаты курсора в xx1 и yy1
    wsp = ton + 100;
    ton = wsp;
    if(ton > 1600) ton =400;
    izm_skor_p();
    goto_xy(xx1-1,yy1);
goto_xy(1,74);
    puts("      ");
goto_xy(1,74);
printf("%dГц",ton);
    return;
}
```

В листинге 5.35 описаны функции, предназначенные для работы с встроенным в программу простейшим аппаратным журналом, в котором должны храниться записи о всех внесенных в этот журнал радиосвязях. Описанные функции позволяют вводить данные в аппаратный журнал и просматривать все имеющиеся в журнале записи.

Листинг 5.35. Продолжение 9

```
// добавить запись о QSO в аппаратный журнал
void addqso(void)
{
    int i;
    if(nomer == MAX)
    {
        puts("В этом файле журнала больше места нет. Сделайте новый.\n");
        getchar();
    }
}
```

```
        return;
    }
    if((fp = fopen( FILENAME, "a")) == NULL)
    {
        printf("Невозможно открыть файл %s\n", FILENAME);
        exit(0);
    }
for(i=0;i<20;i++)
{
    disc.name[i] = qso.call_sign[i];
    disc.namkor[i] = qso.name_k[i];
    disc.qth[i] = qso.qth[i];
    disc.rs1[i] = qso.rst_sent[i];
read_word(1,26,5,disc.tim);
read_word(1,44,10,disc.date);
read_word(0,38,2,disc.mod);
}
    promt(txt23[0],19,10,23,69);
goto_xy(19,5);
puts("    Так Вы УВЕРЕНЫ, что все правильно в экранной таблице?");
puts("\n        Тогда нажимайте <Enter>, иначе для выхода жмите
        <Esc>!!!");

    i = getch();
    if(i == 27) {cls_tx(); return;}
    else
    cls_tx();
    goto_xy(18,0);
    printf("Введите диапазон (до 3 цифр): ");
    gets(disc.diap);
    printf("Введите RST к Вам (до 3 цифр): ");
    gets(disc.rs2);
    printf("Введите информацию по диплому (до 13 букв): ");
    gets(disc.category);
    printf("Введите дополнительную информацию (до 50 букв) : ");
    gets(disc.infor);
    disc.number = nomer;
    if(nomer == 12 || nomer == 25) {nomer = nomer+2; goto wyh;}
    else
    nomer = nomer + 1;
```

```

wyh:    fwrite(&disc, sizeof(disc), 1, fp);
        fclose(fp);
        cls_tx();
        return;
    }

/* вывести записи на экран */
void wlist(void)
{
    int ch;
    if((fp = fopen(FILENAME, "r")) == NULL)
    {
        printf("Невозможно открыть файл %s\n", FILENAME);
        exit(0);
    }
    cls(0x1e);
    goto_xy(0,0);
    puts("  N      Диап.      Дата   Время      Позывной      Город      Мод.
Диплом");
    puts("
=====
");
    while(fread(&disc, sizeof(disc), 1, fp) == 1)
    {
        if(disc.number >= 24) {ch_k = getch();
        if(ch_k == 27) break;
        else
            goto wyh;}
    }
    else
wyh:
    printf("%4d |%4s|%12s|%6s|%9s |%15s|%4s|%15s|\n", disc.number, disc.diap,
disc.date, disc.tim, disc.name, disc.qth, disc.mod, disc.category);
    }
    printf("      Для продолжения нажмите <Enter>");
        getch();

    fclose(fp);
    return;
}

```

```
//ввести номер последнего QSO из файла
void nomer_qso(void)
{
if((fp = fopen(FILENAME, "r")) == NULL)
{ printf("Невозможно открыть файл %s\n", FILENAME);
  exit(0);
}
while(fread(&disc,sizeof(disc),1,fp) != NULL)
{   nomer = disc.number; }
nomer = nomer + 1;
    fclose(fp);
}
```

В листинге 5.36 приведено описание функций по управлению сигналами DTR, RTS и другими, необходимыми для управления через COM-порт работой модема. Функции выполнены на ассемблере с целью повышения быстродействия.

Листинг 5.36. Продолжение 10

```
// установка сигнала DTR
void ust_dtr(void)
{
    asm mov dx, com_adr1
    asm mov al, 1
    asm out dx,al
}

// установка сигнала RTS
void ust_rts(void)
{
    asm mov dx, com_adr1
    asm mov al, 2
    asm out dx,al
}

// установка сигналов DTR и RTS
void ust_dtr_rts(void)
{
    asm mov dx, com_adr1
```

```

asm mov al,3
asm out dx,al
}

// сброс сигналов DTR и RTS
void sbros_dtr_rts(void)
{
asm mov dx,com_adr1
asm mov al,0
asm out dx,al
}

```

В листинге 5.37 располагается описание одной из важнейших функций программы — функции вывода на экран принятого символа. Далее в этом листинге располагаются различные вспомогательные файлы и функция поиска необходимой записи в аппаратном журнале. Все эти функции достаточно простые и, по моему мнению, специального пояснения не требуют.

Листинг 5.37. Продолжение 11

```

// вывести принимаемый символ на экран
void rx_read(void)
{
//int x = xx;
//int y = yy;
goto_xy(xx0,yy0);
putchar(simwol2);
yy0++;
if(yy0 > 78) {yy0 = 1; xx0++; goto_xy(xx0,yy0);}
if(simwol2 == 13) {xx0 = xx0+1; yy0 = 1; goto_xy(xx0,yy0);}
if(simwol2 == 10) {xx0 = xx0+1; yy0 = 1; goto_xy(xx0, yy0);}
if(xx0 >= 17)
{
scr_scrup(0x1d,3,0,16,79);
xx0 = 16; yy0 = 1; goto_xy(xx0,yy0); }
}

// переключение рус/лат
void rus_lat(void)

```

```
{
goto_xy(0,77);
if(flag_rus == 1) {flag_rus = 0; puts("LAT");}
else {flag_rus = 1; puts("РУС");}
}

// реверс норм/обрат сигналов приема
void rewers_pr(void)
{
goto_xy(2,26);
if(rew_pr == 0){rew_pr = 1; rr = 1; puts("нор");}
else {rew_pr = 0; rr = 0; puts("обр");}
}

// реверс сигналов передачи
void rewers_pered(void)
{
goto_xy(2,53);
if(rew_per == 1) {rew_per = 0; puts("обр"); }
else {rew_per = 1; puts("нор");}
}

/* Поиск определенной записи и вывод данных на экран */
locate()
{
char name[20];
char fflag,pause;
int zif;
fflag='n';
if((fp= fopen(FILENAME,"r")) == NULL)
{
printf("Невозможно открыть файл %s\n", FILENAME);
exit(0);
}
strcpy(name,qso.call_sign);
strlwr(name);
zif = strlen(name);
```

```

        while(fread(&disc,sizeof(disc), 1, fp)==1)
        {
            if(strncmp(disc.name, name,zif)==0)
            {
                fflag='y';

getxy();
goto_xy(17,2);
printf("%s          Последняя связь: N%d  %s  %s  %s  %s  %s ",
buf_prom,disc.number,disc.date,disc.tim,disc.name,disc.rs1,disc.rs2);
goto_xy(0,9); printf("%s  ",qso.call_sign);
goto_xy(1,9); printf("          ");
goto_xy(1,9); printf("%s",disc.namkor);
goto_xy(2,9); printf("          ");
goto_xy(2,9); printf("%s",disc.qth);
goto_xy(xx1,yy1);

            }

        }

        fclose(fp);
        if(fflag=='n')
{
getxy();
goto_xy(0,9); printf("%s  ",qso.call_sign);
goto_xy(1,9); printf("          ");
goto_xy(2,9); printf("          ");
goto_xy(17,2);
printf("%s          Записей о связи с этим позывным
в журнале нет.          ",buf_prom);

goto_xy(xx1,yy1);
}

        return 0;
    }

```

Программа CW_QSO имеет встроенный аппаратный журнал, простейший по форме, но имеющий достаточно функций для обычного хранения записей о проведенных радиосвязях. Одним из недостатков встроенного аппаратного журнала является отсутствие возможности распечатать файл с данными о проведенных радиосвязях на принтере. Для работы с принтером более подходит программа QSO_LOG, файл с исходными кодами которой представлен в следующем разделе.

Как сделана программа QSO_LOG

В листинге 5.38 и последующих продолжениях размещен текст файла `qso_log.cpp`, который представляет собой основной файл с исходными кодами программы QSO_LOG. Программа разработана в среде программирования Turbo C++ 3. Эта программа представляет собой самый простой вариант аппаратного журнала для любительской радиостанции, но в ней имеется практически все необходимое для нормального учета радиосвязей.

Почти такая же программа встроена мною во все разработанные программы для цифровой любительской радиосвязи. Отличие программы QSO_LOG от встроенных вариантов в том, что эта программа может выполнять распечатку записей на принтере, а встроенные варианты работать с принтером не могут. Поэтому программу QSO_LOG рекомендую всегда держать в одном каталоге с программой для цифровой радиосвязи.

В приведенном варианте программы запись информации в файл данных выполняется в символьном (текстовом) виде. Советую сделать в своей аналогичной программе запись в файл данных в двоичном формате. Это избавит вас от некоторых проблем.

Листинг 5.38. Файл `qso_log.cpp`

```
/* QSO_log.V1.96 — программа для учета радиосвязей */
#include <iostream.h>
#include <stdio.h>
#include <conio.h>
#include <string.h>
#include <stdlib.h>
#include <process.h>
#define FILENAME "qso_log.dat"
#define TEMPFILE "tempqs"
#define MAX 250

int adduser(), deluser(), chloc(), locate();
int wlist(), wlist2();
int sort(), plist(), plist2(), info();
int openrw(), openwr(), nofind(), showuser();
int getslots(), getslot();
void nomer_qso(void);
FILE *fp, *tp, *printer;
```



```
struct CD
{
    char name[9];
    char diap[5];
    char date[12];
    char tim[8];
    char rs1[5];
    char rs2[5];
    char namkor[10];
    char qth[13];
    char mod[6];
    char category[17];
    char infor[65];
    int number;
    } user;

int slots[MAX];
int count;
int nomer = 1;
```

В листинге 5.39 описана функция `main()`. Функция выводит на экран главное меню и организует работу всех остальных функций программы.

Листинг 5.39. Продолжение 1

```
/* Главная подпрограмма main() */
main()
{
    char select;
    nomer_qso();
    textbackground(1);
    textcolor(14);
    getslots();
    do
    {
        clrscr();
        puts("\n\n    ПРОГРАММА:      QSO_log.V1-2001 ");

        puts("    Автор – Тяпичев Г.А. <RA3XB>, октябрь 2001 года.");
```

```
puts("\n\n\t Учет важных QSO ");  
puts("\t *****\n");  
puts("\t0 - информация о программе");  
puts("\t1 - добавить запись");  
puts("\t2 - выводить запись на экран");  
puts("\t3 - выводить записи как для отчета");  
puts("\t4 - изменить номер записи");  
puts("\t5 - сортировка записей по номерам");  
puts("\t6 - найти запись р/связи");  
puts("\t7 - вывести на печать");  
puts("\t8 - печатать как отчет");  
puts("\t9 - выйти из программы\n\n");  
printf(" \t\t Пожалуйста, введите Ваш выбор: ");  
select = getch();  
  
putchar('\n');  
  
switch(select)  
{  
    case '1' : adduser(); break;  
    case '2' : wlist(); break;  
    case '3' : wlist2(); break;  
    case '4' : chloc(); break;  
    case '5' : sort(); break;  
    case '6' : locate(); break;  
    case '7' : plist(); break;  
    case '8' : plist2(); break;  
    case '0' : info(); break;  
    case '9' : break;  
    default:  
        puts("Ошибка, повторите ввод\n\n");  
}  
  
while(select != '9');  
  
return(0);
```

```
/* Конец подпрограммы MAIN() */
```

В листинге 5.40 описаны функция добавления очередной записи в журнал и вспомогательные функции для работы с номерами записей. Никаких специфических особенностей эти функции не имеют.

Листинг 5.40. Продолжение 2

```
/* Добавить очередную запись в журнал */
adduser()
{
    if(count >= MAX)
    {
        puts("К сожалению, свободных ячеек нет\n");
        getchar();
        return 0;
    }
    if((fp = fopen( FILENAME, "a")) == NULL)
    {
        printf("Невозможно открыть файл %s\n", FILENAME);
        exit(0);
    }
    puts("\n");
    gets(user.name);
    printf("Введите позывной: ");
    gets(user.name);
    printf("Введите диапазон (3 цифры): ");
    gets(user.diap);
    printf("Введите дату (дд-мм-гггг): ");
    gets(user.date);
    printf("Введите время р/связи (чч:мм): ");
    gets(user.tim);
    printf("Введите RST от Вас: ");
    gets(user.rs1);
    printf("Введите RST к Вам: ");
    gets(user.rs2);
    printf("Введите имя корреспондента: ");
    gets(user.namkor);
    printf("Введите QTH корреспондента: ");
    gets(user.qth);
    printf("Вид модуляции: ");
    gets(user.mod);
    printf("Введите информацию по диплому: ");
    gets(user.category);
    printf("Введите дополнительную информацию : ");
```

```
    gets(user.infor);
    user.number = nomer;
    if(nomer == 12 || nomer == 26) { nomer = nomer + 2;}
    else
        nomer = nomer + 1;
    fwrite(&user, sizeof(user), 1, fp);
    fclose(fp);
    getslots();
    return 0;
}

/* Удалить запись из журнала */
deluser()
{
    char delname[10];
    char fflag;
    fflag='n';
    openrw();
    puts("Удаление записи из журнала\n");
    gets(delname);
    printf("Введите позывной: ");
    gets(delname);
    strlwr(delname);
    while(fread(&user, sizeof(user), 1, fp)==1)
    {
        if(strcmp(user.name, delname) != 0)
            fwrite(&user, sizeof(user), 1, tp);
        else
            fflag='y';
    }
    fclose(fp);
    fclose(tp);
    if(fflag=='n')
        nofind();
    else
    {
        opennr();
        while(fread(&user, sizeof(user), 1, tp)==1)
            fwrite(&user, sizeof(user), 1, fp);
    }
}
```

```
        fclose(fp);
        fclose(tp);
    }
    getslots();
    return 0;
}

/* Изменение номера записи */
chloc()
{
    char cname[20];
    char fflag;
    fflag='n';
    if(count>=MAX)
    {
        puts("К сожалению, свободных номеров нет\n");
        getchar();
        return 0;
    }
    openrw();
    puts("Изменение номера записи\n");
    gets(cname);
    printf("Введите позывной: ");
    gets(cname);
    strlwr(cname);
    while(fread(&user, sizeof(user), 1, fp)==1)
    {
        if(strcmp(user.name, cname)!=0)
            fwrite(&user, sizeof(user), 1, tp);
        else
        {
            fflag='y';
            puts("Текущая информация\n");
            showuser();
            puts("\nНовый номер записи\n");
            getslot();
            fwrite(&user, sizeof(user), 1, tp);
        }
    }
}
```

```
fclose(fp);
fclose(tp);
    if(fflag=='n')
        nofind();
    else
        {
            openwr();
            while(fread(&user, sizeof(user), 1, tp)==1)
                fwrite(&user, sizeof(user), 1, fp);
            fclose(fp);
            fclose(tp);
        }
getslots();
return 0;
}

/* Поиск определенной записи и вывод данных на экран */
locate()
{
    char name[20];
    char fflag,pause;
    int zif;
    fflag='n';
    if((fp= fopen(FILENAME,"r")) == NULL)
        {
            printf("Невозможно открыть файл %s\n", FILENAME);
            exit(0);
        }
    puts("Поиск записи\n");
    printf("Введите позывной: ");
    gets(name);
    strlwr(name);
    zif = strlen(name);
    while(fread(&user,sizeof(user), 1, fp)==1)
        {
            if(strnicmp(user.name, name,zif)==0)
                {
                    fflag='y';
                    showuser();
                }
        }
}
```

```

printf("Для продолжения нажмите <Enter>");
    getchar();
    putchar('\n');
}

}

fclose(fp);
fclose(tp);
if(fflag=='n')
    nofind();
return 0;
}

```

В листинге 5.41 описаны функции для работы с принтером и функции форматированного вывода текста на экран. Никаких особенностей в этих функциях нет.

Листинг 5.41. Продолжение 3

```

/* распечатать все записи на принтере */
plist()
{
    if((fp = fopen(FILENAME, "r")) == NULL)
    {
        printf("Невозможно открыть файл %s\n", FILENAME);
        exit(0);
    }

    if((printer = fopen("prn", "w")) == NULL)
    {
        printf("Печатающее устройство не готово к работе\n");
        fclose(fp);
        exit(0);
    }

    while(fread(&user, sizeof(user), 1, fp)==1)
    {
        fprintf(printer, "%4d,%4s,%12s,%6s,%9s,%15s,%4s,%15s\n", user.number,
user.diap, user.date, user.tim, user.name, user.qth, user.mod, user.category);
    }

    fclose(printer);
    fclose(fp);
}

```

```

        return 0;
    }

/* распечатать все записи на принтере как отчет о соревнованиях */
plist2()
{
    if((fp = fopen(FILENAME,"r")) == NULL)
    {
        printf("Невозможно открыть файл %s\n",FILENAME);
        exit(0);
    }

    if((printer = fopen("prn","w")) == NULL)
    {
        printf("Печатающее устройство не готово к работе\n");
        fclose(fp);
        exit(0);
    }

// puts(" N Диап.   Дата   Время       Позывной   Принят   Передан       Инфо");
// puts("
=====");

    while(fread(&user,sizeof(user), 1, fp)==1)
    {
        fprintf(printer,"%5d|%5s|%12s|%7s|%9s|%7s|%7s|%12s|\n",user.number,
user.diap,user.date,user.tim,user.name,user.rs2,user.rs1,user.category);
    }

    fclose(printer);
    fclose(fp);
    return 0;
}

/* вывести записи на экран */
wlist()
{
    int ch_k;
    if((fp = fopen(FILENAME,"r")) == NULL)
    {
        printf("Невозможно открыть файл %s\n",FILENAME);
        exit(0);
    }

```



```

        gotoxy(0,0);
        puts("  N   Диап.      Дата   Время   Позывной      Город   Мод.
Диплом");
        puts("
=====
==");
        while(fread(&user,sizeof(user), 1, fp)==1)
        {
            if(user.number >= 23) {ch_k = getch();
                                if(ch_k == 27) break;
                                else
                                    goto wyh;
                                }

            else
        wyh:
        printf("%4d %4s %12s %6s %9s %15s %4s %15s\n",user.number,user.diap,
user.date,user.tim,user.name,user.qth,user.mod,user.category);
            }
            printf("      Для продолжения нажмите <Enter>");
            getchar(); getchar();
            fclose(fp);
            return 0;
        }

/* вывести записи на экран как для соревнований */
wlist2()
{
    int ch_k;
    if((fp = fopen(FILENAME,"r")) == NULL)
    {
        printf("Невозможно открыть файл %s\n",FILENAME);
        exit(0);
    }
    gotoxy(0,0);
    puts("  N   Диап.      Дата   Время   Позывной   Принято Передано
Инфо");
    puts("
=====
==");
    while(fread(&user,sizeof(user), 1, fp)==1)

```

```

        {
            if(user.number >= 23) {ch_k = getch();
                                if(ch_k == 27) break;
                                else
                                    goto wyh;
                                }
        }
    else
        wyh:
        printf("%5d|%5s|%12s|%7s|%9s   |%7s|%7s|%12s|\n",user.number,user.diap,
            user.date,user.tim,user.name,user.rs2,user.rs1,user.category);
        }
        printf("    Для продолжения нажмите <Enter>");
        getchar(); getchar();
        fclose(fp);
        return 0;
    }

```

В листинге 5.42 описаны вспомогательные функции для сортировки и поиска нужных записей, а также функции для вывода на экран найденной информации.

Листинг 5.42. Продолжение 4

```

/* Сортировать записи по порядку номеров */
sort()
{
    struct CD temp[MAX];
    int index, loop1, loop2, endloop;
    loop1=0;
    loop2=0;
    endloop=0;
    index = 0;
    if((fp = fopen(FILENAME,"r")) == NULL)
    {
        printf("Невозможно открыть файл %s\n",FILENAME);
        exit(0);
    }
    while(fread(&user, sizeof(user), 1, fp)==1)
    {
        temp[index]=user;

```

```
        index++;
    }
    fclose(fp);
    if((fp = fopen(FILENAME, "w")) == NULL)
    {
        printf("Невозможно открыть файл %s\n", FILENAME);
        exit(0);
    }
    for(loop1=1; loop1<MAX+1; loop1++)
    {
        for(loop2=0; loop2<count; loop2++)
            if(temp[loop2].number==loop1)
            {
                fwrite(&temp[loop2], sizeof(temp[loop2]), 1, fp);
                endloop++;
            }
        if(endloop==count)
            break;
    }
    fclose(fp);
    return 0;
}
```

/* Вывести данные из записи на экран */

```
showuser()
{
    printf("Номер п/п:      %d\n", user.number);
    printf("Диапазон :      %s\n", user.diap);
    printf("Дата      :      %s\n", user.date);
    printf("Время      :      %s\n", user.tim);
    printf("Позывной :      %s\n", user.name);
    printf("RST от Вас:      %s\n", user.rs1);
    printf("RST к Вам :      %s\n", user.rs2);
    printf("Имя корп.:      %s\n", user.namkor);
    printf("QTH корп.:      %s\n", user.qth);
    printf("Модуляция:      %s\n", user.mod);
    printf("Диплом      :      %s\n", user.category);
    printf("Доп. инфо:      %s\n", user.infor);
    puts("\n\n");
}
```

```
        return 0;
    }

/* Запись с таким именем не найдена */
nofind()
{
    char pause;
    puts("Запись с таким именем не найдена");
    getchar();
    return 0;
}

/* Невозможно открыть файлы (данных и вспомогательный) */
openrw()
{
    if((fp = fopen(FILENAME, "r")) == NULL)
    {
        printf("Невозможно открыть файл %s\n", FILENAME);
        exit(0);
    }

    if((tp = fopen(TEMPFILE, "w")) == NULL)
    {
        printf("Невозможно открыть файл %s\n", TEMPFILE);
        fclose(fp);
        exit(0);
    }

    return 0;
}

/* Невозможно открыть файлы (данных и вспомогательный) */
openwr()
{
    if((fp = fopen(FILENAME, "w")) == NULL)
    {
        printf("Невозможно открыть файл %s\n", FILENAME);
        exit(0);
    }

    if((tp = fopen(TEMPFILE, "r")) == NULL)
    {
        printf("Невозможно открыть файл %s\n", TEMPFILE);
```

```
        fclose(fp);
        exit(0);
    }
    return 0;
}
```

/* Проверка ячейки на занятость назначение номера записи */

```
getslot()
{
    int index, flag, pause;
    do
    {
        flag=0;
        printf("Введите номер записи: ");
        scanf("%d",&user.number);
        for(index=0;index<count;index++)
        {
            if(slots[index]==user.number)
            {
                printf("К сожалению, ячейка номера занята. Попробуйте другую\n");
                flag=1;
            }
        }
    }
    while(user.number<1 || flag==1);
    count++;
    slots[count]=user.number;
    return 0;
}
```

/* Чтение из файла */

```
getslots()
{
    int index;
    index=0;
    count=0;
    if((fp = fopen(FILENAME,"r")) != NULL)
    {
        while(fread(&user,sizeof(user), 1, fp)==1)
```

```

        {
            slots[index]=user.number;
            index++;
            count++;
        }
    fclose(fp);
}

return 0;
}

// ввести номер последнего QSO из файла
void nomer_qso(void)
{
if((fp = fopen(FILENAME, "r")) == NULL)
{ printf("Невозможно открыть файл %s\n",FILENAME);
  exit(0);
}
while(fread(&user,sizeof(user),1,fp) != NULL)
{   nomer = user.number; }
nomer = nomer + 1;
    fclose(fp);
}

/* Информация */
info()
{
    int pause;

puts("\n");
puts("\t\t Программа разработана в октябре 1996 года для применения");
puts("\tw любительской радиосвязи при учете самых важных и значимых
QSO");
puts("\tnа радиостанции UA3XBI. Переработана RA3XB в октябре 2001г.");
puts("\n\t\t      Некоторые пояснения ");
puts("\t      =====\n");
puts("\t\t 1. Заполнение базы данных выполняется через пункт 1 меню.");
puts("\tМожно вводить не только латинскими буквами. Каких-либо других");
puts("\тособенностей в этом пункте нет. Программа позволяет сделать 250");
puts("\т\записей.");
puts("\t\t 2. Пункты 2, 3, 4 и 8 особенностей не имеют. Будьте");

```

```

puts("\tвнимательны при выполнении пункта 2!!!");
puts("\t\t 3. Сортировка записей осуществляется автоматически. При");
puts("\tсортровке все записи выстраиваются строго по присвоенным");
puts("\tномерам.");
puts("\t\t 4. В пункте 6 в качестве позывного можно задать префикс");
puts("\tтого или одну первую букву. При этом на экран последовательно");
puts("\tбудут выведены все записи, имеющие аналогичную префиксную");
puts("\tчасть позывного.");
puts("\t\t 5. В пункте 7 на печать выводятся все записи в сокращении.");
    printf("\n\n\t    Для продолжения нажмите <Enter>");
//    getchar();
        getchar();
        clrscr();
puts("\n\t6. Максимальное количество вводимых букв не должно превы-
шать:");
puts("\n\t    для позывного — 9 букв, для диапазона — 4 буквы,");
puts("\t    время — 6, RST — 4, имя корп. — 9, QTH — 12,");
puts("\t    модуляция — 5, инфo1 — 17, инфo2 — 65 букв.");
puts("\n\t7. НИКАКИХ ИЗМЕНЕНИЙ В ФАЙЛЕ qso_log.dat НЕ
ДЕЛАТЬ!!!");
puts("\n\n\t\t Ж Е Л А Ю      У С П Е Х О В  !");
    printf("\n\t    Для продолжения нажмите <Enter>");
    getchar();
    return 0;
}

```

Программа QSO_LOG выполнена в простейшем варианте. Порой ничего большего и не требуется, но выглядит такая простейшая программа не очень красиво. Поэтому советую усложнить эту программу, сделать ее более красочной. Уверен, что вам это удастся сделать.

Как сделать программу для PSK31

Мною не разрабатывались программы для PSK31 потому, что считаю этот вид связи не перспективным. Вполне возможно, что некоторые из радиолюбителей еще длительное время будут пользоваться этим видом связи, но широкая популярность ему не грозит.

В то же время я советую вам подробнее разобраться с принципами создания программ для PSK31, чтобы в дальнейшем применить эти принципы при создании прочих своих программ.

Тем, кто решает серьезно заняться созданием программ для PSK31 или других подобных программ, предназначенных для работы со звуковой картой компьютера, советую чаще заглядывать в Интернет на странички известных разработчиков программ для любительской радиосвязи. Многие из этих адресов вы найдете в моей книге.

Много очень интересного материала можно найти в Интернете на страничке американского радиолюбителя Мое Wheatley (AE4JY), которая располагается по адресу <http://www.qsl.net/ae4jy/>.

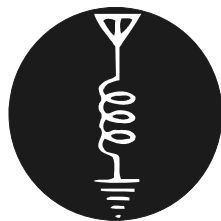
Этот радиолюбитель-программист на своей страничке в Интернете разместил несколько вариантов исходных кодов различных программ для PSK31.

- ❑ Один из вариантов представляет собой программу WinPSK 2.11, которая включает в себя подключаемую библиотеку PSKCore.dll и управляющую часть программы. Выполняется на Visual C++ 6. При этом в файле PSKCore.dll находится драйвер, выполняющий формирование символов и выдачу информационных посылок на микрофонный вход передатчика, а также и декодирование поступающих от приемника сигналов.
- ❑ Второй вариант представляет собой программу WinPSK 1.2, в которой подключаемой библиотеки нет, драйвер находится вместе с другими файлами в одном пакете и компилируется посредством Visual C++ 6.

Кстати, сейчас многие радиолюбители-программисты используют созданный AE4JY файл PSKCore.dll при написании своих программ. Например, радиолюбитель из Великобритании Julian V. Moss (G4ILO) создал свою программу для PSK31 под названием PSK31TEST. Эта программа также использует библиотеку PSKCore.dll и создана она в среде программирования Borland Delphi. Найти программу можно в Интернете по адресу <http://www.tech-pro.co.uk/> или на указанной выше странице AE4JY.

Надеюсь, что эти мои советы помогут вам создать свои программы для любительской цифровой радиосвязи PSK31, предназначенные для работы только под управлением Windows.

Заключение



Эта книга написана, чтобы познакомить радиолюбителей, имеющих собственные любительские радиостанции, а также и широкий круг читателей с особенностями применения компьютеров для целей любительской радиосвязи. Я попытался сделать обзор всех основных видов цифровой любительской радиосвязи, дать описание простых в настройке и эффективных в работе любительских аппаратов, необходимых для проведения этих видов связи.

Среди читателей обязательно найдутся недовольные тем, что в книге много внимания уделено "старому" виду связи — телетайпу. Как довод в пользу нестарееющего телетайпа привожу таблицу, в которой размещены все пункты по цифровым видам связи из календаря международных соревнований по радиосвязи на март и апрель 2002 года, опубликованного в журнале "Радиомир КВ и УКВ" № 2, 2002.

Дата	Время, UTC	Mode	Contest
2–3.03.2002	21:00/1:59	RTTY	Открытый чемпионат Украины
10.03.2002	00:00/04:00	RTTY	North American Sprint
16–18.03.2002	02:00/02:00	RTTY	BARTG Spring RTTY CONTEST
16–17.03.2002	12:00/12:00	SSTV	DARC International HF SSTV Contest
27–28.04.2002	12:00/12:00	RTTY	SP DX RTTY Contest

Из пяти запланированных в марте и апреле соревнований по цифровым видам радиосвязи только одно проводится видом связи SSTV (любительское телевидение с медленной разверткой), а все остальные — только телетайпом (RTTY). Такое же положение с преимущественным применением телетайпа наблюдается и в целом по любому году.

Так что вывод напрашивается один — телетайп был и пока остается одним из основных видов любительской цифровой радиосвязи. И это будет продолжаться еще достаточно долго, пока не найдется этому виду связи достойная замена.

Открытый сатириками принцип "Спасение утопающих — дело рук самих утопающих" очень здорово подходит к нашей радиолюбительской действи-

тельности. Никто, кроме нас самих, не позаботится об обеспечении нас необходимыми для радиосвязи аппаратами или компьютерными программами. Так что большинство из нас, радиолюбителей, будет иметь на своих радиостанциях только то, что сделано своими руками. Поэтому, не задумываясь, беритесь за паяльник для создания нужной конструкции, или за клавиатуру компьютера, чтобы создать нужную программу.

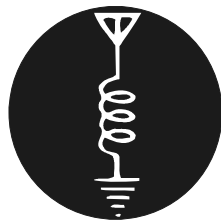
По всем, возникшим в процессе работы над книгой, вопросам можно обращаться к автору по электронной почте **<mailto:ra3xb@kaluga.ru>**.

Ответы на отдельные вопросы можете найти на моей странице в Интернете, которая располагается по адресу **<http://users.kaluga.ru/ra3xb/>**.

Желаю успехов!

Геннадий А. Тяпичев — RA3XB.

Приложение



Описание дискеты

Дискета содержит четыре каталога:

- ☐ 02 ☐ 03 ☐ 04 ☐ 05

Каталог 02 содержит следующие подкаталоги:

- ☐ BAYCOM — вспомогательные файлы к программе BAYCOM;
- ☐ BCM — вспомогательные файлы к программе BCM;
- ☐ BCT — вспомогательные файлы к программе BCT;
- ☐ CW_QSO — исполняемый и другие файлы программы CW_QSO;
- ☐ F6FBB — вспомогательные файлы к программе F6FBB;
- ☐ QSO_LOG — исполняемый и другие файлы программы QSO_LOG;
- ☐ RTTY_BBS — исполняемый и другие файлы программы RTTY_BBS;
- ☐ RTTY_GT1 — исполняемый и другие файлы программы RTTY_GT1;
- ☐ RTTY_QSO — исполняемый и другие файлы программы RTTY_QSO;
- ☐ Terman93 — вспомогательные файлы к программе Terman93;
- ☐ TFPCX — вспомогательные файлы к программе TFPCX.

Каталог 03 содержит следующие файлы:

- ☐ modem21.zip — описание конструкции модема MODEM21;
- ☐ modem22.zip — описание конструкции модема MODEM22;
- ☐ modem3.zip — описание конструкции модема MODEM3.

Каталог 04 содержит подкаталог ORBITA, в котором находится исполняемый и вспомогательные файлы программы ORBITA, предназначенной для расчета элементов орбит искусственных спутников Земли (ИСЗ).

Каталог 05 содержит следующие подкаталоги:

- ☐ CW_QSO — здесь находятся файлы с исходными кодами, относящиеся к программе CW_QSO;
- ☐ ORBITA — здесь находятся файлы с исходными кодами программы ORBITA;
- ☐ QSO_LOG — здесь находятся файлы с исходными кодами программы QSO_LOG;
- ☐ RTTY_QSO — здесь находятся файлы с исходными кодами программы RTTY_QSO.

Список литературы

1. Бензарь В. К., Лебедев В. И. Вокруг Земли на радиоволне. — Минск: Польша, 1986.
2. Тяпичев Г. Простой демодулятор // Радиоловитель. КВ и УКВ. — 1999. — № 6.
3. Тяпичев Г. Пакет через Sound Blaster // Радиоловитель. КВ и УКВ. — 1999. — № 8.
4. Тяпичев Г. Модем // Радиоловитель. КВ и УКВ. — 2000. — № 12.
5. Тяпичев Г. Знакомьтесь: режим РАСТОР // Радиоловитель. КВ и УКВ. — 2001. — № 6.
6. Тяпичев Г. Как выбрать программу для работы телетайпом // Радиомир КВ и УКВ. — 2001. — № 9.
7. Тяпичев Г. Универсальный любительский модем // Радиомир КВ и УКВ. — 2001. — № 11.
8. Тяпичев Г. RTTY-BBS — электронная доска объявлений // Радиомир КВ и УКВ. — 2002. — № 2.
9. Тяпичев Г. О расчете орбит для спутников RS // Радиоловитель. КВ и УКВ. — 1996. — № 7.