

## Глава 4

# Компьютер как тренажер телеграфиста

---

### Что такое телеграф

Телеграф — это такой вид радиосвязи, при котором радиостанции (передающая сигналы и приемная) обмениваются между собой прерывистыми тональными сигналами, которые принято называть *тональными посылками*. При этом сами сигналы имеют строго определенную (для данной скорости передачи) длительность звучания и размеры пауз между тональными посылками.

История создания телеграфа повествует, что 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 обоими концами к электрической батарее. За опытом внимательно наблюдал пассажир по имени Самюэль Морзе. Это натолкнуло его на мысль попытаться создать систему передачи сигналов по проводам. После пяти лет экспериментов в 1837 году ему удалось 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 (два типа тональных посылок) — точка и тире.

---

После этого многие изобретатели пытались придумывать иные различные коды, из которых один был хитроумнее другого, но время выбрало лучший из них, тот, что 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 из 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, Ь, 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 потребуется немного больше времени и терпения.

## Принципы построения телеграфного сигнала

Для желающих заняться изучением телеграфа в табл. 4.1 приведен код Морзе.

**Таблица 4.1.** 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	_ _ _ ·	Вопросит. знак	.. _ _ _
R	Р	· _ ·	Двоеточие	_ _ _ _
S	С	· · ·	Восклицат. знак	_ _ _ _
T	Т	_	Точка с запятой	_ · _ _ ·
U	У	.. _	Кавычки	· _ _ _ ·
V	Ж	· · _ _	Начало передачи	_ _ _ _
W	В	· _ _	Знак раздела	_ _ _
X	Ь	_ _ _	Конец передачи	· _ _ _

Таблица 4.1. Окончание

Латинский алфавит	Русский алфавит	Код Морзе
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 — коротковолнников 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. 4.2 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 она является ответом.

Таблица 4.2. Q-код

Q — код	Зашифрованная фраза
QRA?	Какой ваш адрес?
QRA	Мой адрес...
QRB?	Какое 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 радиолюбительский код, в котором употребляются сокращенные до двух-трех букв известные слова радиолюбительского 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, которую 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. Такой телеграф 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 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 радиолучитель на слух контролирует правильность приема компьютером телеграфных сигналов.

## Windows-приложение как тренажер

Разработанное мною Windows-приложение WinCW1 представляет собой простой тренажер для самостоятельного изучения приема телеграфных сигналов кода Морзе на слух. Название этой программы WinCW1, по моему мнению, должно обозначать «Телеграф (CW1) для Windows».

Для любителей программирования в среде Borland C++ Builder далее приведены исходные коды этой программы.

---

Достаточно быстро изучить процесс программирования в среде C++ Builder можно с помощью книги «Быстрое программирование на C++» [3]. Эта книга фактически является самоучителем работы в среде Borland C++ Builder и предназначена для людей, не владеющих языком программирования C/C++. Кроме того, в приложении 2 этой книги вы можете найти краткие сведения о среде быстрого создания компьютерных программ Borland C++ Builder.

---

Если вы решили начать процесс создания своей программы по описанному ниже образцу программы WinCW1, то сначала необходимо создать директорию для хранения этой программы, например, D:\WinCW1\. Затем запускаем в работу Borland C++ Builder любой версии. В директории D:\WinCW1\ создаем проект приложения WinCW1. Все эти процедуры очень подробно описаны в книге «Быстрое программирование на C++» [3] и в других учебниках по программированию в Borland C++ Builder.

Программа WinCW1 создавалась для учебных целей, поэтому очень упрощена. Для нормальной работы этой программы необходимо использовать дополнительно файлы подключаемых библиотек CWcore1.dll и CWcore1.lib, а также файлы с записанными в них звуковыми сигналами определенной частоты.

Поэтому не забудьте в созданную вами директорию D:\WinCW1\ скопировать файлы CWcore1.dll и CWcore1.lib, а также файлы с записанными звуковыми сигналами — файлы zw800.wav, zw1000.wav, zw1200.wav и другие аналогичные. Вполне возможно, что вы захотите обойтись только одним или двумя звуковыми файлами. Перечисленные файлы можно взять в Internet на сайте [r3xb.nm.ru](http://r3xb.nm.ru).

После создания проекта нового приложения в заданной директории, необходимо подключить к этому проекту файл CWcore1.lib, который связывает приложение с библиотекой. Подключение выполняется посредством выбора пунктов меню **Add to Project**.

Начнем разработку проекта программы с создания формы и ее заполнения компонентами. На рис. 4.1 представлен предлагаемый вариант формы.

Первыми на форму устанавливаются три компонента для выбора строки из списка — компоненты `ComboBox`. Один из этих компонентов служит для выбора частоты генерируемого звукового сигнала, второй служит для выбора скорости передачи, третий — для выбора определенного текста. Краткое описание этого компонента можно найти в книге «Быстрое программирование на C++» [3].

После установки компонента `ComboBox` на форму необходимо в свойствах этого компонента выбрать строку `Items` и щелкнуть на расположенном справа квадратике. Появляется диалоговое окно **String List Editor** — редактор перечня строк (рис. 4.2).

Вначале текстовое окно пустое, курсор находится в левом верхнем углу. Печатаем текст первой строки и нажимаем клавишу <Enter> для перевода строки. Затем таким же образом печатаем последующие строки. В заключении щелкаем на кнопке **OK**. Свойство `ItemIndex` оставляем равным -1.

Точно таким же образом оформляем перечни выбираемых строк и в оставшихся двух компонентах `ComboBox`.

Давайте сначала посмотрим файл WinCW.h, чтобы познакомиться с названиями примененных компонентов и использованных в программе функций.

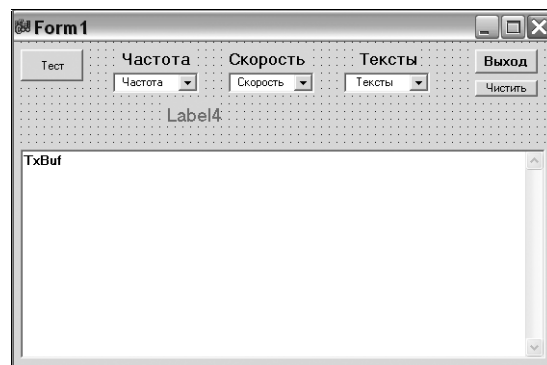


Рис. 4.1. Форма нового проекта



Рис. 4.2. Окно для редактирования выбираемых строк

**Листинг 4.1.** Файл *WinCW.h*


---

```

//-----
#ifndef WinCWH
#define WinCWH
//-----
#include <Classes.hpp>
#include <Controls.hpp>
#include <StdCtrls.hpp>
#include <Forms.hpp>
#include <Buttons.hpp>
//-----
class TForm1 : public TForm
{
__published: // IDE-managed Components
    TMemo *TxBuf;
    TButton *Button1;
    TComboBox *ComboBox1;
    TSpeedButton *SpeedButton1;
    TComboBox *ComboBox2;
    TComboBox *ComboBox3;
    TLabel *Label1;
    TLabel *Label2;
    TLabel *Label3;
    TButton *Button2;
    TLabel *Label4;
    void __fastcall SpeedButton1Click(TObject *Sender);
    void __fastcall TxBufKeyPress(TObject *Sender, char &Key);
    void __fastcall Button1Click(TObject *Sender);
    void __fastcall ComboBox2Change(TObject *Sender);
    void __fastcall ComboBox3Change(TObject *Sender);
    void __fastcall ComboBox1Change(TObject *Sender);
    void __fastcall Button2Click(TObject *Sender);
private: // User declarations
public: // User declarations
    __fastcall TForm1(TComponent* Owner);
};
//-----
extern PACKAGE TForm1 *Form1;
//-----
#endif
//-----

```

---

Я думаю, что не стоит комментировать приведенный листинг. Если вы приняли за создание программы, значит уже имеете опыт рассмотрения подобных файлов. Следует отметить только тот факт, что файл полностью сгенерирован программой и добавлять в него что-либо от себя пока не стоит.

Далее, в листинге 4.2, приведен текст файла `WinCW.cpp`.

#### Листинг 4.2. Файл `WinCW.cpp`

---

```
//-----
#include <vcl.h>
#pragma hdrstop
#include <mmsystem.h>
#include "WinCW.h"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm1 *Form1;

extern "C" __declspec(dllimport) void Say(String);
extern "C" __declspec(dllimport) String GetVariant(void);
extern "C" __declspec(dllimport) void SetSkorost(int);
extern "C" __declspec(dllimport) void SetTXFrequency(int);
extern "C" __declspec(dllimport) void SetFlagZw(int);
extern "C" __declspec(dllimport) void TXsimwol(char);

char chz;
int skorost, ton;
int flag_zw;
void Sound(int);           // функция добавлена автором
void SimTX(char);          // функция добавлена автором
void ClearBuf1(void);      // функция добавлена автором

String FileName;
String TextName;
char buff1[4000];
//-----
```

---

В начале листинга 4.2 выполнено, как обычно, подключение необходимых заголовочных файлов. Строка `TForm1 *Form1;` представляет собой объявление формы `Form1`. Далее идут строки, в которых выполняется объявление функций, импортируемых из подключаемой библиотеки. Строка объявления выглядит так:

```
extern "C" __declspec(dllimport) String GetVariant(void);
```

В этой строке объявляется функция `GetVariant()`, предназначенная для получения от библиотеки сведений об варианте исполнения данной версии библиотеки. Если вы помните, то в самой подключаемой библиотеке эта функция объявлялась как экспортируемая и имела вид:

```
extern "C" __declspec(dllexport) String GetVariant(void);
```

Подобные объявления дают возможность четкого выполнения функции.

Далее, до конца листинга, идут обычные объявления глобальных переменных величин, а также функций, разработанных автором и вставленных в этот файл.

Листинг 4.3 начинается с функции описания формы. В тело этой функции добавлена строка, устанавливающая заголовок для главного окна программы и ниже две строчки, которые убирают текст с метки `Label4` и окна редактирования `TxBuf`.

#### Листинг 4.3. Продолжение 1 файла *WinCW.cpp*

---

```
//-----
__fastcall TForm1::TForm1(TComponent* Owner)           // 1
    : TForm(Owner)
{
    Caption = "Телеграфный тренажер";
    Label4->Caption = "";
    TxBuf->Text = "";
}
//-----
// функция, вызываемая кнопкой «Тест»
void __fastcall TForm1::SpeedButton1Click(TObject *Sender)    // 2
{
    String s;
    Say("Привет от DLL!");
    s = GetVariant();
    Label4->Caption = "Вариант - CWcore1 var."+s;
    SetSkorost(20);
    Sound(3);
}
//-----
// Функция выдает звуковой сигнал заданной частоты
void Sound(int sec)                                           // 3
{
    PlaySound(FileName.c_str(), NULL, SND_ASYNC);
    for(int i=0; i<sec; i++)
        Sleep(1000);
    PlaySound(NULL, NULL, NULL);
}
//-----
// Функция выдачи символа на передачу с клавиатуры
```



---

```

void __fastcall TForm1::TxBufKeyPress(TObject *Sender, char &Key)
// 4
{
    TXsimwol(Key);
}
//-----
// функция для кнопки «Выход»
void __fastcall TForm1::Button1Click(TObject *Sender) // 5
{
    Close();
}
//-----

```

---

Функция 2 является реакцией на нажатие кнопки **Тест**. Функция, как и сама эта кнопка, не являются обязательными и служат только для того, чтобы начинающий пользователь мог удостовериться в работоспособности подключаемой библиотеки. Подобным проверкам был посвящен полностью предыдущий проект специального тестирующего приложения.

Функция 3 разработана автором и предназначена только для выполнения в составе предыдущей функции 2.

Функция 4 служит для озвучивания буквенного символа, введенного с клавиатуры. Для того чтобы задействовать эту функцию, достаточно в запущенной на выполнение программе выполнить следующие действия:

- ввести необходимую звуковую частоту;
- ввести необходимую величину скорости передачи сигнала;
- щелкнуть мышкой на любом участке окна **TxBuf**, т.е. активизировать это окно.

Далее, при нажатии на любую буквенную клавишу клавиатуры вашего компьютера, в динамике будет звучать код Морзе заданной буквы.

Функция 5 является реакцией на нажатие кнопки **Выход**.

Листинг 4.4 начинается с функции (6), которая служит для выбора тона звукового сигнала и, одновременно, определяет название звукового файла, из которого этот сигнал должен выполняться.

---

#### Листинг 4.4. Продолжение 2 файла *WinCW.cpp*

---

```

//-----
// функция переключения тона
void __fastcall TForm1::ComboBox2Change(TObject *Sender) // 6
{
    int a = ComboBox2->ItemIndex;
    switch(a)
    {
        case 0: FileName = "zw800.wav", ton = 800; break;
        case 1: FileName = "zw1000.wav", ton = 1000; break;
    }
}

```

```

    case 2: FileName = "zw1200.wav", ton = 1200; break;
    case 3: FileName = "zw1600.wav", ton = 1600; break;
    default: FileName = "zw1000.wav", ton = 1000;
}
SetTXFrequency(ton);
}
//-----
// Функция переключения скорости передачи кода Морзе
void __fastcall TForm1::ComboBox3Change(TObject *Sender)    // 7
{
    int b = ComboBox3->ItemIndex;
    switch(b)
    {
        case 0: skorost = 50; break;
        case 1: skorost = 40; break;
        case 2: skorost = 30; break;
        case 3: skorost = 20; break;
        case 4: skorost = 10; break;
        default: skorost = 30;
    }
    SetSkorost(skorost);
}
//-----
// Функция выбора задействованного текстового файла
void __fastcall TForm1::ComboBox1Change(TObject *Sender)    // 8
{
    int b1 = ComboBox1->ItemIndex;
    switch(b1)
    {
        case 0: TextName = "text1.txt"; break;
        case 1: TextName = "text2.txt"; break;
        case 2: TextName = "text3.txt"; break;
        case 3: TextName = "text4.txt"; break;
        case 4: TextName = "text5.txt"; break;
        case 5: TextName = "text0.txt"; break;
        default: TextName = "text0.txt";
    }
    TxBuf->Lines->LoadFromFile(TextName); // Записывает текст файла в
TxBuf
    int Size = TxBuf->GetTextLen(); // Определяет число символов в тексте
    Size++; // Учитывает символ нуля
    Label4->Caption = "Передано символов: "+IntToStr(Size); // На экран
    strcpy(buff1, (TxBuf->Text).c_str()); // выводит текст из буфера buff1

```

```

int j =0; // Установка нулевого значения вспомогательной переменной j
do      // Начинается цикл do ... while()
{
    // В переменной chz сохраняется значение символа,
    chz = buff1[j]; // находящегося в buff1[j]
    SimTX(chz);     // Символ chz подается на передачу
    j = j++;        // Переменная j увеличивается на единицу
}
while(j != Size); // Заканчивается цикл do ... while()
TxBuf->Text = "";  // Очищается от текста окно TxBuf
}
//-----

```

Здесь используется «переключатель» типа `switch()`. Этот переключатель представляет собой конструкцию `switch/case/default`. В данном случае, задействованный нами компонент `ComboBox2` имеет свойство `ItemIndex`, которое представляет собой номер выбранной строки. Поясню. Компонент содержит несколько строк, подобных обычному меню выбора. Выбор каждой из этих строк вызывает задействование различных параметров. Например, в данном варианте при выборе первой строки (она имеет `ItemIndex = 0`) устанавливается частота тона 800 Гц и при этом задействуется файл `zw800.wav`.

При выборе следующей строки в работу включается другой файл, и т.д. Каждая строка завершается оператором `break`, после выполнения которого все последующие строки не обрабатываются. Если выбрана такая величина `ItemIndex`, которой нет в перечне, тогда управление передается последней строке, начинающейся со слова `default`. При этом, в нашем примере, начинает работать файл `zw1000.wav`, который вызывает звук частотой 1000 Гц. Последней строкой вызывается выполнение библиотечной функции, которая устанавливает и в приложении и в библиотеке единую величину рабочей частоты, которая соответствует сделанному нами выбору.

Подробнее о переключателе `switch()` можно прочитать в книге «Быстрое программирование на C++» [3] или другой литературе по C/C++.

В функции 7 задействован точно такой же переключатель меню, но предметом выбора в данном случае являются величины скоростей передачи сигналов Морзе.

В функции 8 задействован тоже такой переключатель, как и в двух предыдущих функциях, но переключает он выбор текстового файла, текст которого должен передаваться звуковым кодом Морзе.

Дальше разворачиваются действия, которые сначала подготавливают текст выбранного файла для передачи, для чего переносят его в специальный буфер. В то же время текст из файла размещается на экране (для наглядности) в окне **TxBuf**.

Текст, находящийся в буфере, начинает обрабатываться циклом `do ... while(<условие>)`. Это еще одна новинка для вас, хотя в C/C++ применяется довольно часто. В этом цикле все основные действия и операторы заключены в фигурных скобках, расположенных сразу же после оператора `do` и перед оператором `while(<условие>)`.

Смысл цикла можно сформулировать так: цикл выполняется до тех пор, пока является истиной условие, заключенное в скобках после оператора `while()`. В нашем примере цикл будет продолжаться до тех пор, пока переменная величина `j` будет меньше или больше числа символов в тексте.

Как только цикл заканчивается, окно **TxBuf** очищается от текста. Это выполняется последней строкой в функции 8.

О цикле `do ... while(<условие>)` можно прочитать в книге «Быстрое программирование на C++» [3] или другой литературе по C/C++.

Листинг 4.5 начинается с функции 9, которая создана автором специально для передачи звуковым сигналом символа, взятого из буфера `buff1`. Об этих буферах можно прочитать в книге «Быстрое программирование на C++» [3].

---

**Листинг 4.5.** Продолжение 3 файла *WinCW.cpp*

---

```
//-----
// функция передачи звуковым сигналом буквенного символа из буфера
void SimTX(char) // 9
{
    TXsimwol(chz);
    return;
}
//-----
// Функция очистки буфера (заполняет нулевым значением)
void ClearBuf1(void) // 10
{
    for(int i=0; i<4000;i++)
    {
        buff1[i] = 0;
    }
}
//-----
// Функция очистки от текста всех компонентов
void __fastcall TForm1::Button2Click(TObject *Sender) // 11
{
    TxBuf->Text = "";
    ClearBuf1();
    Label4->Caption = "";
}
//-----
```

---

Функция 10 служит для очистки буфера от текста. Эта процедура выполняется путем заполнения всего пространства буфера нулевыми символами. В этой функции задействован еще один цикл — цикл `for(;;)`. Подробное описание этого цикла можно найти в книге «Быстрое программирование на C++» [3].

Функция 11 также предназначена для очистки и реагирует на нажатие кнопки **Чистить**. Но она очищает от текста все задействованные компоненты.

В заключении следует выбрать из меню команду создания выполняемого файла разрабатываемого приложения, а затем выбрать **Run⇒Run**.

На рис. 4.3 вы видите главное окно разработанного приложения.

## Как работает программа тренажера

Программа представляет собой простейший тренажер, предназначенный для изучения приема на слух сигналов кода Морзе. Нужно знать, что это скорее просто учебная программа, которая может (в крайнем случае) послужить и тренажером.

Смысл работы программы заключается в том, что заранее нужно подготовить текстовые файлы с наборами изучаемых буквенных символов, а затем прослушивать звучание этих текстов при одновременной записи принимаемых символов карандашом на бумагу. Начинать нужно с наименьшей скорости, повышать скорости нужно постепенно и очень осторожно. В организациях РОСТО можно найти литературу с рекомендациями по изучению приема телеграфных сигналов, там же могут быть и готовые тексты для обучающихся.

Порядок работы должен быть следующий.

- После запуска программы устанавливается необходимая величина скорости приема сигналов и наиболее удобный для вас тон сигнала.
- Затем выбираете нужный текст и включаете его на прослушивание. Одновременно с прослушиванием звучания буквенных символов, их нужно записывать на бумагу. Прослушивание с одновременной записью во много раз улучшает запоминание.

Программа позволяет вычислить скорость приема телеграфа по методу PARIS. Так называется очень древний метод определения скорости. Для выполнения этого метода нужно подготовить в текстовом редакторе NOTEPAD файл с текстом, который записан в листинге 4.6. Файл следует назвать `Text0.txt`.

### Листинг 4.6. Файл `Text0.txt`

```
//-----
PARIS PARIS PARIS PARIS PARIS PARIS PARIS PARIS PARIS PARIS
PARIS PARIS PARIS PARIS PARIS PARIS PARIS PARIS PARIS PARIS
//-----
```

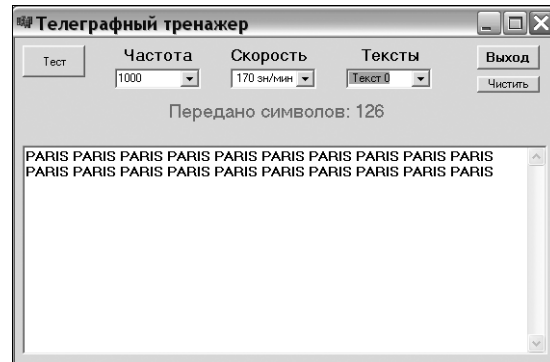


Рис. 4.3. Телеграфный тренажер

Файл `Text0.txt` содержит две строки, которых 20 раз повторяется слово PARIS (с одним пустым символом между каждым словом). Этот текст содержит ровно 100 полезных буквенных символов. Запускать на звучание этот файл нужно с одновременным включением секундомера. Выключается секундомер сразу же после окончания звучания текстовых символов.

Измерение скорости приема выполняется в единицах измерения «Символов за минуту» следующим образом.

- Количество полезных символов (100) делится на число секунд, этим самым определяется число прозвучавших символов за одну секунду.
- Полученное частное от деления умножаем на 60 (число секунд в минуте) и получаем величину скорости в симв/мин (или знаков/минуту).

Предлагаю вам текстовый файл `text5.txt`, который может быть очень полезен для начинающих коротковолновиков. Текст этого файла представлен в листинге 4.7.

#### Листинг 4.7. Файл `Text5.txt`

---

```
//-----
CQ CQ CQ de UA3XBI UA3XBI pse k
- -
Ge dr OM - tks fer call - ur RST is 589 589 589 -
- MY NAME IS GENADY GENADY GENADY -
MY QTH IS KALUGA KALUGA KALUGA - PSE HW ? -
- -
R ok dr OM -
hr QRU - Vy pse ur QSL - my QSL sure -
Vy tks fer QSO - best 73 es fb dx -
gb - dsw -
sk
//-----
```

---

В файле `text5.txt` записано большинство кодовых сокращений, которыми пользуются коротковолновики при проведении телеграфных радиосвязей. Рекомендую прослушивать текст этого файла (с одновременной записью) очень часто и до тех пор, пока эти типовые кодовые фразы не станут для вас узнаваемыми с первого раза и на разных скоростях приема.

Кроме перечисленных выше возможностей, программа позволяет прослушивать звучание кодов отдельных букв (как русского, так и латинского алфавитов), полученных непосредственно с помощью клавиатуры компьютера.

Для работы в этом режиме следует:

- ♦ выбрать необходимую частоту звучания сигнала;
- ♦ выбрать необходимую величину скорости передачи кода Морзе;
- ♦ установить курсор в любое место окна **TxBuf** и щелкнуть левой кнопкой мышки. Такой процесс называется: «активизировать окно TxBuf».

После выполнения этих условий нажатие на любую буквенную клавишу компьютерной клавиатуры вызовет запись в окно **TxBuf** введенной с клавиатуры буквы и звуковое воспроизведение телеграфного кода этой буквы.

## Программа CW\_QSO как тренажер

В качестве хорошего тренажера для изучения приема и передачи телеграфных сигналов можно использовать программу **CW\_QSO**. Программа располагается на приложенном к книге компакт-диске.

Эта программа **CW\_QSO** была написана довольно давно. В те времена работающие на языке Си программисты использовали, как правило, различные версии среды программирования Borland Turbo C/C++ которая работала под управлением операционной системы MS DOS.

Не следует думать, что созданные для MS DOS программы не будут работать под управлением операционных систем Microsoft Windows 95/98/ME/XP!

Как показал опыт, программы, созданные для операционной системы MS DOS, как правило, нормально работают под управлением операционных систем Microsoft Windows 95/98. Но под управлением операционных систем Microsoft Windows ME/XP могут работать далеко не все из этих программ.

Однако существует достаточно простой способ, чтобы «заставить» эти программы работать на компьютерах, на которых установлены операционные системы Microsoft Windows ME/XP. Этот способ предусматривает использовать вариант MS DOS, задействованный в конкретной версии одной из операционных систем Microsoft Windows ME/XP.

## Как запустить программу под управлением MS DOS

Запустить в работу на самом современном компьютере (например, с ЦПУ Pentium 4) программу, предназначенную для работы под управлением MS DOS, можно с помощью обычной дискеты емкостью 1,44 Мб, отформатированной, как системная дискета. Подробно процесс создания загрузочной дискеты описан в приложении 1 этой книги.

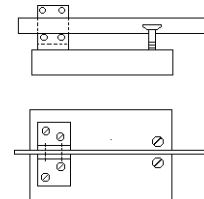
## Что может программа CW\_QSO

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 не-

скольких шурупов 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. Свободный 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 пальцами, 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, описано ниже, в описании программы `cw_qso`. Это только одна из возможных простейших конструкций манипулятора для полуавтоматического 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. 4.4. схематично изображен один из возможных вариантов простого телеграфного ключа. Собственно говоря, телеграф никогда не был цифровым видом связи. Но с появлением компьютеров стали создаваться программы и для приема и для передачи телеграфных сигналов.

Новую жизнь любительскому телеграфу дало в свое время развитие космической техники. Дело в том, что наряду со специальными (военными или метеорологическими) спутниками стали создаваться спутники, на которых была установлена аппаратура для проведения любительских радиосвязей, в том числе и телеграфом. Было очень интересно провести через установленный на спутнике ретранслятор телеграфную радиосвязь с корреспондентом из другого континента!



**Рис. 4.4.** Схема телеграфного ключа

Кроме того, каждый из создаваемых в ту пору отечественных ИСЗ имел на борту специальную измерительную систему, которая постоянно следит за состоянием всех технических систем, находящихся на борту спутника, и передает на Землю информацию именно телеграфными сигналами. Эта передача информации передавалась непрерывно определенными «порциями». В начале и конце каждого такого пакета информации передается специальный позывной, принадлежащий данному спутнику. Информация о состоянии технических систем спутника, передаваемая непрерывно (пакетами) называется «телеметрическая информация», или просто «телеметрия». По появлению или исчезновению на определенной частоте передачи этой информации можно судить о появлении спутника в зоне радиовидимости, или его ухода из этой зоны. Передатчик, излучающий эту непрерывную информацию, называется «маяком» (Beacon).



Телеграф пока занимает видное место в любительской радиосвязи. Этим видом связи проводятся самые интересные и самые редкие из радиосвязей. Поэтому я постараюсь дать как можно больше информации о телеграфе, дам описание компьютерных программ, которые помогут начинающим освоить этот интересный вид радиосвязи.

## Программа CW\_QSO работает с кодом Морзе

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

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. Было создано особенно много различных тренажеров для изучения приема на слух телеграфных сигналов, для наращивания спортсменами-скоростниками скоростей 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, задействованной в данный момент на станции.
5. Текст с подтверждением приема всех переданных корреспондентом сообщений.
6. Текст прощания с корреспондентом и передача наилучших пожеланий.

В п. 2 указано, что корреспонденты обмениваются кодовыми сообщениями, описывающих качество 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». Эта кодовая фраза характерна для станций работающих с 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 связь со станцией, если у нее 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 разных сайтах в Internet можно найти самые разнообразные из 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. В названии программы заложено ее предназначение — 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 операционных систем MS DOS или Microsoft Windows 95/98. Для работы под управлением операционных систем Microsoft Windows 95/98 все файлы, входящие в пакет программы следует разместить в одном подкаталоге, затем с помощью файлового менеджера **Мой компьютер** выбрать используемый файл `cw_qso.exe` и щелкнуть на нем правой кнопкой мыши. В появившемся меню выбрать строку **Создать ярлык**. В результате будет создан файл с расширением `PIF`, который можно перетащить на **Рабочий стол** операционной системы Microsoft Windows, а затем запускать программу непосредственно с рабочего стола с помощью созданной пиктограммы, например, дважды щелкая на ней мышью. Лучший вариант работы этой программы на современном компьютере описан в приложении 1.

## Общие положения

Для нормальной работы компьютера с этой программой `cw_qso` в радиоэфире необходимо дополнительное устройство, называемое «модем». Модем нужен для согласования компьютера с передатчиком радиостанции. Если использовать компьютер и программу в качестве тренажера или просто игрушки, то модем иметь не обязательно. Для работы в эфире рекомендуется использовать модем MODEM3, описание конструкции которого можно скачать на сайте `r3xb.nm.ru`.

Программа CW\_QSO не имеет зарубежных аналогов и рассчитана на пользователей, предпочитающих при проведении радиосвязей использовать *русский* язык. Хорошо работает и с текстами, использующими латинский алфавит.

### Перечень команд

Все команды программы CW\_QSO выполняются с помощью сочетаний клавиш, установленных им в соответствие. Перечень всех команд, задействованных в программе, приведен в табл. 4.3.

**Таблица 4.3.** 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+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+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+F2>	Повысить тон звучания сигнала в громкоговорителе компьютера
<Shift+F3>	Понизить 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 РУС/ЛАТ символов

Для ввода команд в программе задействованы функциональные клавиши <F1>–<F12>, а также сочетания нажатия функциональной клавиши с клавишами управ-

ляющими — <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 килобайт. 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 компьютера, что особенно важно, если Вы решили заняться изучением приема телеграфных сигналов, или вам просто нужно контролировать работу 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. 4.4.

**Таблица 4.4.** 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

Рабочий экран разбит на три части. Его 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 сим-

вол 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 передаче

В программе задействованы большие возможности по выбору различных, заранее подготовленных текстов, возможность работы непосредственно с клавиатурой.

Для работы с клавиатуры следует нажать сочетание клавиш <Alt+F8>, дождаться пока программа выдаст в эфир позывные, затем как можно быстрее набирать 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>.

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 о том, 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 описанного выше телеграфного ключа, подключенному к игровому порту (порту джойстика). Имеется в виду не USB порт, применяющийся для подключения новых джойстиков, а тот старый игровой порт, располагавшийся на звуковых картах.

---

Простой двухсторонний телеграфный ключ позволяет работать телеграфом 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 учета проведенных радиосвязей

С помощью клавиши <F11> можно 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. В результате нажатия клавиши <F11> на экране открывается окно подсказки. Если вы действительно что-то упустили, то следует нажать клавишу <Esc> и команда будет отменена, если все нормально, то нажимайте клавишу <Enter> и вводите недостающие данные. Все введенные в журнал QSO можно просмотреть с помощью 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`. Эта программа простого аппаратного журнала и должна находиться в одной директории с программой `cw_qso.exe`. Все введенное в файл базы данных одной программой можно просмотреть другой программой и наоборот.

### Проверка скоростей передачи

Проверка указанных на экране скоростей передачи выполняется с помощью сочетания 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:

$$(100 / t) 60 = A \text{ знаков/минуту},$$

где 100 — число символов (знаков) в файле «g»;

$t$  — число секунд (по секундомеру);

60 — число секунд в 1-ой минуте;

$A$  — полученная величина скорости (знаков/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.