

УДК 621.391

М.В. Маленьков, М.А. Никитин

### ВОЗМОЖНОСТИ И МЕТОДЫ СОЗДАНИЯ ПРОГРАММНОЙ БИБЛИОТЕКИ ДЛЯ РАБОТЫ С МОБИЛЬНЫМИ ТЕЛЕФОНАМИ СТАНДАРТА GSM

164

*Рассматриваются методы и механизмы реализации программной библиотеки для работы с мобильными телефонами стандарта GSM, подключенными к серийному ПК. Предлагается спецификация объекта, реализующего базовые свойства и функции по работе с GSM-терминалом.*

*This article describes basic methods and strategy of software realization of GSM-terminal object library. It's proposed a sample design of possible object, utilized basic methods and functions of involved objects.*

#### Коммуникационная составляющая стандарта GSM и связанные с ним стандарты

Большинство современных мобильных телефонов стандарта GSM имеют возможность так или иначе подключаться к персональному компьютеру. Необходимость в таком подключении возникает в случае, если пользователь хотел бы подключаться к сети Интернет, используя встроенный во многие современные GSM-терминалы модуль GPRS, отсылать или принимать факсы, редактировать телефонную книгу и т.д. Изначально при разработке стандарта GSM проектировщики определили, что любой GSM-терминал по сути своей является модемом и как следствие может быть подключен к персональному компьютеру через стандартный асинхронный (коммуникационный) порт. Более того, набор команд для управления GSM-терминалом базируется на ставшем стандартом de-facto наборе команд Hayes, что значительно упрощает взаимодействие с терминалом, т. к. в данном случае его можно рассматривать как стандартный Hayes-совместимый модем. Таким образом, к примеру, любая операционная система может использовать подключенный к ПК GSM-терминал без каких-либо дополнительных настроек. Однако для реализации некоторых специфичных для сотового телефона функций, таких как, к примеру, редактирование телефонной книги или отсылка коротких сообщений (SMS), необходимо использовать расширенные команды, специфичные для стандарта GSM и описанные в соответствующих документах консорциума GSM.



Разработкой и поддержкой стандарта GSM занимается European Telecommunications Standards Institute (ETSI), расположенный в городе Sophia-Antipolis (Франция). На интернет-сайте организации (<http://www.etsi.org>) можно получить доступ к документации, содержащей, в частности, информацию о спецификации расширенных команд GSM-терминала, упоминавшихся ранее. Программные библиотеки, приведенные в данной статье, базируются на спецификации ETS 300 916 от июля 1998 г. (четвертая редакция) — «*Digital cellular telecommunications system (Phase 2+); AT command set for GSM Mobile Equipment (ME) (GSM 07.07 version 5.5.1)*» [1]. В свою очередь, некоторые производители мобильных телефонов используют собственные AT-расширения для реализации некоторых «фирменных» особенностей. В данной работе использовался документ A30880-A10-A001–3-D376 (*AT command set for S45 Siemens mobile phones and modems. Release/Version: 1.8*) компании Siemens [2].

### Логика построения библиотеки

В настоящей работе не ставилась задача рассмотреть полную реализацию библиотеки взаимодействия компьютера с GSM-терминалом. В ней описываются лишь основные методы и принципы построения подобной кодовой библиотеки.

Любому взаимодействию с GSM-терминалом должны предшествовать некие процедуры инициализации порта, сбор данных о терминале и т.д. По завершении работы необходимо выполнить некоторые процедуры, как-то, закрытие коммуникационного порта и т.д. Было бы логично выполнить данные процедуры автоматически, скрыв от прикладного программиста рутинные операции взаимодействия с терминалом на аппаратном уровне.

На рис. 1 показан упрощенный алгоритм подсистемы-конверта, обрамляющего собственно код библиотеки работы с GSM-терминалом. Основной задачей библиотеки на данном этапе является сбор данных о наличии свободных коммуникационных портов в системе, открытии и закрытии указанного порта согласно заданным характеристикам и сбор основных данных о подключенном к указанному коммуникационному порту устройстве.

### Методы работы с коммуникационным портом

Для работы с коммуникационными портами в среде операционных систем Microsoft Windows 9X/NT используются стандартные вызовы API Windows, а именно: ReadFile, WriteFile, CreateFile, GetCommTimeouts, SetCommTimeouts, CloseHandle, FlushFileBuffers, GetCommState, SetCommState, BuildCommDCB, WaitCommEvent, SetCommMask, GetCommMask, GetLastError и ClearCommError.

Для того что бы начать работу с выбранным коммуникационным портом, прежде всего необходимо открыть данный порт и получить *заголовок (handle)* для открытого порта. В дальнейшем все взаимодействие программы с открытым коммуникационным портом будет идти с ис-



пользованием именно этого заголовка. Вот пример функции на Visual Basic, реализующей процедуру открытия и инициализации коммуникационного порта.

166

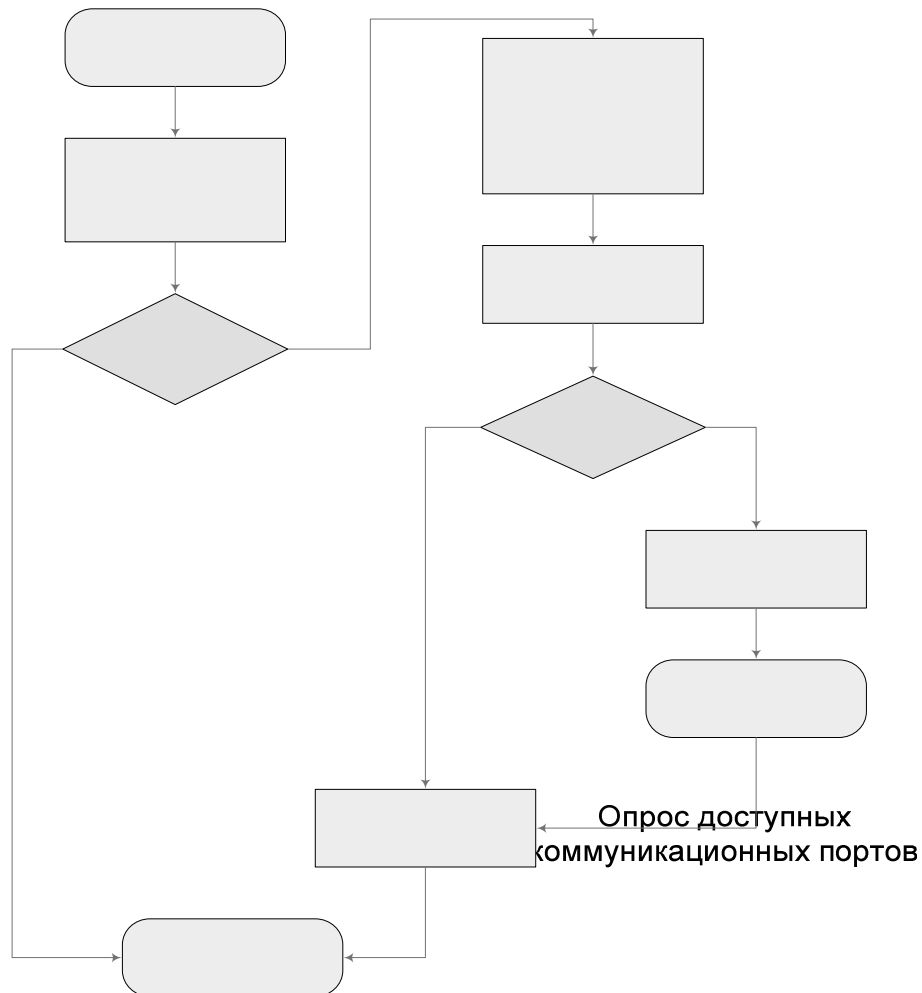


Рис. 1

Листинг 1

Процедура открытия коммуникационного порта

```
Public Function hOpenPort(strPortName As String, _
    Optional strDCB As String = "300,n,8,1", _
    Optional lTimeOut As Long = 1000) As Long

    Dim hPort As Long
    Dim DCBConf As DCB
    Dim CTConf As COMMTIMEOUTS

    hPort = CreateFile(strPortName, &HC0000000, 0, 0, 3, 0, 0)
    If hPort = -1 Then GoTo Trap
```

ДА

НЕТ

Если есть свободные  
комм. порты?



```

If GetCommState(hPort, DCBConf) = 0 Then GoTo Trap
If BuildCommDCB(strDCB, DCBConf) = 0 Then GoTo Trap
If SetCommState(hPort, DCBConf) = 0 Then GoTo Trap

'set all timeout values
CTConf. ReadIntervalTimeout = 0
CTConf. ReadTotalTimeoutConstant = lTimeOut
CTConf. ReadTotalTimeoutMultiplier = 0
CTConf. WriteTotalTimeoutConstant = lTimeOut
CTConf. WriteTotalTimeoutMultiplier = 0
If SetCommTimeouts(hPort, CTConf) = 0 Then GoTo Trap

hOpenPort = hPort
Exit Function

```

167

```

Trap:
    hOpenPort = -1
End Function

```

Особенностью данного метода является то, что для открытия коммуникационного порта используется стандартная функция для открытия файла *CreateFile*, однако при работе с коммуникационным портом помимо собственно открытия порта необходимо так же выполнить его инициализацию. Для описания параметров, с которыми открывается коммуникационный порт, используется структура *DCB*. Для удобства программирования в списке функций API Windows существует стандартная функция, которая заполняет данную структуру значениями «по умолчанию».

В дальнейшем для записи данных в открытый коммуникационный порт используется стандартная функция API — *WriteFile*. Вот пример функции, записывающей в открытый коммуникационный порт строку текста.

Листинг 2

#### Пример функции, реализующей запись в коммуникационный порт

```

Public Function bWritePortStr(hPort As Long, _
    strCommand As String) As Boolean

    Dim iInc As Integer
    Dim byCommand(65536) As Byte
    Dim lRetBytes As Long
    Dim lResult As Long
    Dim lRet As Long

    For iInc = 0 To Len(strCommand) - 1
        byCommand(iInc) = Asc(Mid(strCommand, iInc + 1, 1))
    Next iInc

    Call FlushFileBuffers(hPort)

    lResult = WriteFile(hPort, byCommand(0), _
        Len(strCommand), lRetBytes, 0)

```



```
Call FlushFileBuffers(hPort)
Call ClearCommError(hPort, lRet, vbNull)
bWritePortStr = CBool(lResult)
End Function
```

Соответственно для чтения из порта используется функция *ReadFile*. Вот пример функции на Visual Basic, модифицированной для чтения строки из коммуникационного порта.

Листинг 3

168

#### Процедура чтения строки из коммуникационного порта

```
Public Function strReadPortStr(hPort As Long) As String
    Dim strTMP As String
    Dim byCommand(65536) As Byte
    Dim lRetBytes As Long
    Dim iInc As Integer

    For iInc = 1 To 255
        If ReadFile(hPort, byCommand(0), 1, _
            lRetBytes, 0) = 0 Then Exit For
        If lRetBytes <> 0 Then 'character received
            If byCommand(0) <> 13 Then 'not CR
                strTMP = strTMP & Chr(byCommand(0))
            End If
        Else
            Exit For
        End If
    Next iInc

    Call FlushFileBuffers(hPort)

    strReadPortStr = strTMP
End Function
```

По завершении работы с коммуникационным портом его необходимо корректно закрыть. Для этого используется функция API *CloseHandle*. Вот пример ее использования в коде программы.

Листинг 4

#### Закрытие коммуникационного порта

```
Public Sub ClosePort(hPort As Long)
    Call FlushFileBuffers(hPort)
    Call CloseHandle(hPort)
End Sub
```

Приведенные выше фрагменты кода иллюстрируют базовые методы работы программы с коммуникационными портами в среде Windows. Однако для большего удобства представляется целесообразным объединить данные функции в рамках общей программной библиотеки. Наиболее естественным способом создания такого рода библиотеки является



реализация подсистемы в виде группы контролов (*controls*). Объектная структура таких контролов позволит программисту легко взаимодействовать с GSM-терминалом и скроет процедуру взаимодействия компьютер-терминал. Схема, приведенная на рис. 2, иллюстрирует возможные методы и свойства базового объекта COMTerm.

Базовая спецификация коммуникационного объекта

Рассмотрим примерную спецификацию искомого объекта. На рис. 2 представлен COMTerm объект в виде «черного ящика», у которого имеется ряд свойств и методов доступных «снаружи».

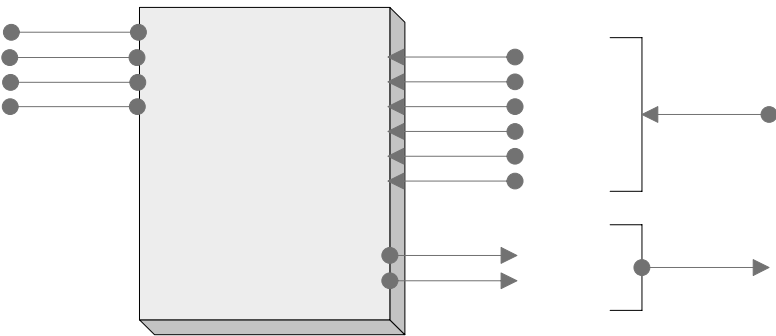


Рис. 2

Свойства данного объекта можно разделить на три основные части, а именно:

- 1. **Свойства GET** предназначены исключительно для получения программой неких данных от COMTerm объекта.
- 2. **Свойства LET** предназначены исключительно для передачи программой неких параметров, определяющих функционирование COMTerm объекта.
- 3. **Общие функции** реализуют некоторые общие методы взаимодействия прикладной программы с COMTerm объектом. Они могут как получать, так и принимать данные.

Назначение основных LET-свойств объекта COMTerm:

<b>EnumerateCOM()</b>	Установить номер подключаемого коммуникационного порта.
<b>Connect()</b>	Установить скорость обмена данными.
<b>Disconnect()</b>	Управление потоком данных.
<b>Execute()</b>	Выставлять сигнал DTR.
<b>bDTR</b>	Выставлять сигнал RTS.
<b>bRTS</b>	Величина тайм-аута по ошибке соединения.
<b>TimeOut</b>	

Назначение основных GET-свойств объекта COMTerm:

<b>strResult</b>	Строка-ответ GSM-терминала.
<b>iCOMList</b>	Массив, содержащий перечисление доступных коммуникационных портов.

COMTerm  
object



Назначение основных методов (функций) объекта COMTerm:

<b>iEnumirateCOM()</b>	Вернуть количество свободных COM-портов в системе.
<b>bConnect()</b>	Подключиться к выбранному коммуникационному порту.
<b>Disconnect()</b>	Закрыть соединение с коммуникационным портом.
<b>bExecute()</b>	Выполнить AT-команду.

Таким образом, типичный сеанс работы с GSM-терминалом может выглядеть примерно следующим образом. (Предполагается, что все действия по открытию, инициализации и работе с коммуникационным портом выполняются объектом COMTerm и приведенный пример является соответственно расшифровкой секции «Обмен данными с устройством», рис. 3).

170

Подключаемся к "COM4: " на скорости 19200 bps  
Коррекция ошибок: None (DTR=ON, RTS=OFF)

```
→ ATZ
← OK
→ ATE0
← OK
→ AT+CGMI
← SIEMENS OK
→ AT+CGMM
← S45 OK
→ AT+CGMR
← 23 OK
→ AT+CGSN
← 123456789123456 OK
→ AT+CSCS?
← +CSCS: "GSM" OK
```

Информация о телефоне, подключенном к порту "COM4: "  
Производитель: SIEMENS  
Модель телефона: S45  
Серийный номер телефона: 123456789123456  
Версия прошивки: 23  
Кодовая страница: GSM

Рис. 3

### Дальнейшее развитие древа объектов

В предыдущем параграфе рассматривались общие принципы построения объекта, реализующего базовые функции взаимодействия компьютер-терминал. На основе этого класса предполагается создать объект GSMTerm, реализующий в себе общеупотребительные свойства мобильного телефона, такие как записная книжка, работа с SMS и т.д. Очевидно, что такой функционал также удобно организовать в виде отдельных объектов. Так, к примеру, объект «записная книжка» будет



инкапсулировать в себя все свойства работы с телефонной книжкой GSM-телефона и обладать такими свойствами как, редактирование, удаление, изменение и т.д.

### **Возможности применения предложенной библиотеки объектов**

Согласно вышеописанной спецификации была создана структура классов, реализующих базовые функции протокола работы с GSM-терминалами. На данный момент ряд описанных классов используется в авторском проекте SMS Wizard 2.0 (<http://www.sms-wizard.com>). В настоящее время рассматривается возможность распространения данных классов в отдельной библиотеке объектов, доступной для сторонних программистов.

171

### **Список литературы**

1. *European Telecommunications Standards Institute (ETSI) ETS 300 916 «Digital cellular telecommunications system (Phase 2+); AT command set for GSM Mobile Equipment (ME) (GSM 07.07 version 5.5.1)».*
2. *Siemens Mobile A30880-A10-A001 – 3-D376 (AT command set for S45 Siemens mobile phones and modems. Release/Version: 1.8).*

### **Об авторах**

М.В. Маленьков — аспирант, КГУ.

М.А. Никитин — д-р физ.-мат. наук, проф., КГУ.

УДК 004

*Д.А. Савкин*

### **МЕТОДИКА ПРОЕКТИРОВАНИЯ РЕГИОНАЛЬНОГО «ЭЛЕКТРОННОГО ПРАВИТЕЛЬСТВА» (на примере калининградской области)**

*Раскрывается понятие «электронное правительство» на региональном и федеральном уровнях. Описаны требования к интернет-порталу регионального «электронного правительства» и основные подходы к методике его проектирования.*

*A notion «e-government» on regional and federal level is presented. Requirements are described of internet-portal regional «e-government» and basic approaches of a creation method.*

### **1. Понятие «электронное правительство»**

В последние годы как на государственном, так и на региональном и муниципальном уровнях интенсивно применяются информационно-коммуникационные технологии для повышения эффективности и ка-