

# Программирование порта LPT в Visual Basic

Д. ЗАХАРОВ, г. Прокопьевск Кемеровской обл.

*Овладев управлением интерфейсными портами компьютера, радиолюбитель может подключить к ним различные сигнальные и исполнительные устройства и датчики, превращая компьютер в центр управления бытовой электроникой, системой охраны квартиры или в измерительный прибор. Наиболее привлекательен для начинающего параллельный порт LPT, исходя из значения для подключения к компьютеру принтера. Отсюда происходит и аббревиатура LPT — Line Printer Terminal (первые принтеры печатали информацию "line by line" — построчно). Позже область применения этого порта значительно расширилась, к нему стали подключать самые разные периферийные устройства. К сожалению, сегодня его (как, впрочем, и другие порты компьютера) постепенно вытесняет быстродействующая универсальная последовательная шина USB.*

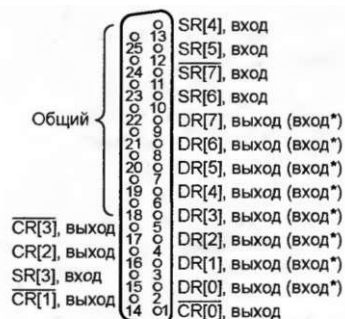
Разъем порта LPT на системном блоке компьютера — 25-контактная розетка DB-25F. На ее контакты можно подавать и снимать с них логические сигналы уровней, характерных для микросхем структуры ТТЛ. Логически низким считается напряжение 0...0,8 В, высоким — 2,4...5 В. Соединять выходные контакты разъема с общим проводом или с источником напряжения, не превышающего +5 В, рекомендуется только через резисторы сопротивлением не менее 300 Ом. Не допускается подавать как на входы, так и на выходы порта отрицательное напряжение или положительное более 5 В. Подключать к порту и отключать что-либо от него можно только при полностью отключенном от сети 220 В компьютере (сетевая вилка вынута из розетки). Если подключаемое устройство имеет сетевое питание, оно тоже должно быть физически отсоединено от сети.

Несоблюдение этих требований может иметь тяжелые последствия. Если расположенная внутри компьютера микросхема контроллера параллельного порта выйдет из строя, потребуются ремонт или замена материнской платы.

При включении компьютера его параллельный порт работает в режиме Centronics — простейшем и традиционном для этого порта с момента его появления в компьютерах. Иногда этот режим называют Simple Parallel Port (SPP). Более сложные режимы EPP и ECP используются, как правило, для скоростного обмена информацией с лазерными принтерами и сканерами. Мы их рассматривать не будем, потому что программирование работы с портом в таких режимах доступно лишь опытным программистам.

С точки зрения программы порт LPT в режиме Centronics представляет собой три восьмизрядных регистра в пространстве ввода-вывода микропроцессора: регистр данных DR по

адресу &H378, регистр состояния принтера SR по адресу &H379 и регистр управления принтером CR по адресу &H37A. Указанные адреса относятся к порту LPT1, обычно единственному в компьютере. Если в нем имеются другие параллельные порты, им также



Примечание. Контакты 2-9 становятся входами при CR[5]=1.

Рис. 1

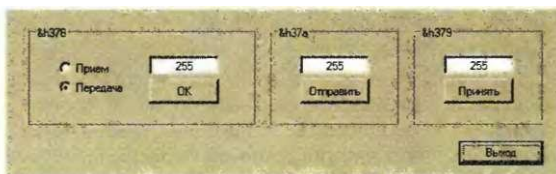


Рис. 2

отводят по три регистра с последовательными адресами. Например, регистры порта LPT2 обычно имеют адреса &H278—&H27A.

Входы и выходы регистров порта (правда, не все) соединены с контактами интерфейсного разъема, как показано на рис. 1. Поэтому, записывая в эти регистры определенные коды, можно устанавливать соответствующие логические уровни напряжения на выходных контактах разъема, а читая коды из регистров, определять уровни поданных на входы внешних сигналов.

Работать с портом LPT можно практически в любой среде программирования и операционной системе. Наиболее доступными считаются среды Visual Basic и Delphi, причем во всем, что требуется для программирования порта, они весьма схожи. Нужно сказать, что современные многозадачные операционные системы (в том числе семейства Windows) не допускают прямых обращений из прикладных программ к портам компьютера. Это сделано для того, чтобы избежать конфликтов между одновременно выполняемыми программами, если они случайно обратятся к одному и тому же порту в один и тот же момент времени. Связь с портами возможна лишь через специальные программы-драйверы, автоматически выполняющие все, что необходимо для разрешения конфликтов. Программисту остается лишь написать несколько управляющих команд.

Мы будем использовать одну из самых популярных библиотек таких программ — Inpout32.dll второй версии, которую легко найти в Интернете. Она применима в различных средах программирования и операционных системах. Работая в Windows 98, файл Inpout32.dll необходимо скопировать в папку C:\Windows\system\, а в Windows XP — в папку C:\Windows\system32\ . Во многих случаях достаточно просто поместить этот файл в папку исполняемой программы. Для программирования в DOS дополнительные драйверы не нужны, достаточно предусмотренных в используемом языке программирования обычных команд ввода-вывода в порт.

Дальнейшее изложение относится к работе с параллельным портом в системе программирования Visual Basic 6.0 под управлением Windows XP. Для ее освоения разработана простая программа. Ее проект, в том числе исполняемый файл test.exe и файл главной (и единственной) формы Form1.frm приложены к статье. При запуске

этой программы на экране монитора появится окно, показанное на рис. 2. Нажимая в нем на экранные кнопки и вводя числа в соответствующие поля, можно устанавливать уровни напряжения на выходах порта и считывать состояние его входов (оно будет отображено числом в соответствующем поле). Библиотеку для работы с портом LPT "подключает" к программе фрагмент файла Form1.frm, показанный в таблице.

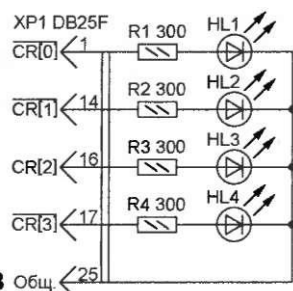
```
Option Explicit
Private Declare Function Inp Lib "inpout32.dll"
Alias "Inp32" (ByVal PortAddress As Integer) As Integer
Private Declare Sub Out Lib "inpout32.dll" _
Alias "Out32" (ByVal PortAddress As Integer, ByVal Value As Integer)
```

Прежде всего, разберем работу с регистром управления CR (напомним, его адрес — &H37A). В рассматриваемом случае ее выполняет подпрограмма

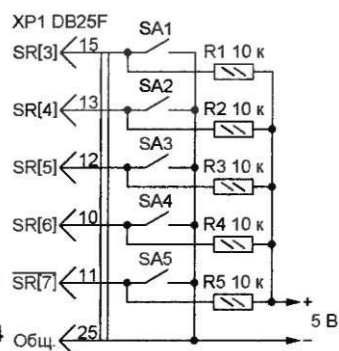
```
Private Sub Command4_Click()
    Out &H37A, Text2.Text
End Sub
```

При нажатии на экранную кнопку **Command4** ("Отправить") она записывает в регистр по адресу &H37A двоичный код, соответствующий десятичному числу, введенному в поле над этой кнопкой.

Для наглядности соберем и подключим к разъему LPT светодиодный узел по схеме, изображенной на **рис. 3**. Введем



**Рис. 3**



**Рис. 4**

в нужное поле число 4 (двоичное 00000100) и нажмем на кнопку "Отправить". После этого все четыре светодиода окажутся включенными. Дело в том, что разряды CR[0], CR[1] и CR[3] соединены с контактами разъема через инверторы, поэтому при записи 0 в эти разряды уровни на соответствующих им контактах стали высокими. Чтобы включить только светодиод HL3, нужно ввести число 15 (двоичное 00001111), а при вводе числа 11 (двоичное 0001011) все светодиоды будут выключены. Старшие разряды регистра управления (CR[4]—CR[7]) с контактами разъема не соединены, поэтому их состояние в данном случае никакого значения не имеет.

Чтобы изучить работу с регистром состояния SR, подключим к разъему порта узел, схема которого изображена на **рис. 4**. При разомкнутых выключателях SA1—SA5 через резисторы R1—R5 на контакты разъема поступает напряжение высокого логического уровня. Его источником могут быть любой сетевой адаптер с выходным напряжением 5 В, батарея из трех гальванических

элементов и даже один из выходов порта LPT, на котором описанным ранее способом установлен нужный уровень напряжения. Во многих компьютерах резисторы, по назначению аналогичные R1—R5, уже имеются, в установке внешних резисторов в таких случаях нет необходимости.

При нажатии на экранную кнопку "Принять" будет выполнена подпрограмма

```
Private Sub Command5_Click()
    Text3.Text = Inp(&H379)
End Sub
```

Она выведет в поле над кнопкой число, отображающее содержимое регистра SR. Если все выключатели (рис. 4) разомкнуты, это будет 126 (двоичное 01111110), а если они замкнуты — 134 (10000110). Значения разрядов SR[3]—SR[6] соответствуют уровням, поданным на соответствующие контакты разъема, а значение разряда SR[7] инверсно уровню на контакте 11. Так как младшие разряды SR[0]—SR[2] на разъеме не выведены, их значения не зависят от поданных на его контакты сигналов.

Главный регистр порта — регистр данных DR по адресу &H378. Именно через него печатаемая информация побайтно передается на принтер. Все восемь разрядов регистра соединены с контактами разъема, причем без инверторов. Эти восемь цепей часто объединяют названием "шина данных". В исходном состоянии она работает только на вывод. Однако почти во всех современных компьютерах имеется возможность переключить ее на параллельный ввод восьмиразрядных двоичных кодов. Для этого достаточно записать единицу в разряд CR[5] регистра управления.

К сожалению, в режиме Centronics никакие сигналы о том, в каком направлении работает шина данных порта LPT, на его разъеме не выводятся. Поэтому необходимо соблюдать особую осторожность и подавать на эту шину внешние сигналы, только удостоверившись, что ее программное переключение "на прием" выполнено. Иначе могут быть повреждены интерфейсные микросхемы как самого компьютера, так и подключенного к порту источника сигналов. Этот недостаток устранен в режимах EPP и ECP, где предусмотрен полный набор сигналов управления направлением передачи информации по шине данных параллельного порта.

В рассматриваемой тестовой программе с регистром данных работает подпрограмма

```
Private Sub Command3_Click()
    Out &H378, Text1.Text
    Text1.Text = Inp(&H378)
End Sub
```

При нажатии на экранную кнопку "OK" она записывает в регистр данных число из поля, находящегося над кноп-

кой, а затем читает содержимое регистра и отображает его в том же поле. Естественно, если регистр работает как выходной (на экране отмечен пункт "Передача"), число в поле остается режимом. Чтобы убедиться, что логические уровни на контактах 2—9 разъема порта в этом случае соответствуют введенному в поле вручную и записанному в регистр данным числу, подключите к разъему узел, аналогичный тому, схема которого показана на рис. 3, но с увеличенным до восьми числом светодиодов и резисторов.

Операцию переключения шины данных на ввод выполняет подпрограмма

```
Private Sub option1_Click()
    Out &H37A, 32
End Sub
```

Ее вызов происходит при нажатии на экранную кнопку с зависимой фиксацией "Прием". Кнопкой "Передача" вызывают аналогичную подпрограмму, отличающуюся лишь тем, что она записывает в регистр управления не 32 (двоичное 00100000), а ноль, возвращая таким образом шину данных в режим вывода.

Когда шина данных переведена в режим ввода, процедура Out в рассмотренной ранее подпрограмме, вызываемой при нажатии на кнопку "OK", фактически не работает. Однако функция Inp возвращает значение, соответствующее уровням на выводах 2—9, установленных подключенными к ним внешними цепями. В виде десятичного числа оно появляется в поле над кнопкой "OK". Задавать логические уровни на линиях шины данных можно с помощью узла, подобного использовавшемуся для работы с регистром состояния (рис. 4).

Чтобы не усложнять программу, отображение в поле ввода над кнопкой "Отправить" изменений состояния регистра управления с помощью кнопок "Прием" и "Передача" не предусмотрено.

Освоив приведенные в статье примеры, мы научились выводить через порт из компьютера 12 и выводить в него 5 логических сигналов либо (в другом режиме) выводить 4 и вводить 13 таких сигналов. Теперь можно разрабатывать гораздо более сложные программы и устройства, подключаемые через порт LPT к компьютеру

**От редакции.** Упомянутые в статье и другие необходимые для работы с тестовой программой файлы находятся на нашем FTP-сервере по адресу <ftp://ftp.radio.ru/pub/2007/09/testlpt.zip>.