

Parallel Port Interface Box



Introduction

The Parallel Port Interface Box is a simple device that connects to the IBM PC's parallel (aka printer) port. Plugging this box into the printer port immediately interfaces your PC to the outside world. For instance, you can hook up motors to the box, and write computer programs to control them. You could interface sensors and turn your computer into a home control unit. For example, you can have motion sensors which can detect when to turn on lights. You can interface a smoke alarm to your computer. The computer can call and play a pre-recorded panic message to 911 when the alarm goes off. You can design a burglar alarm system as well. Almost all PCs have a parallel port. The advantages of using this port are (1) unlike the serial RS-232 port, you can control up to 8 devices *simultaneously*. (2) unlike the AT expansion bus slot, you don't have to open up your computer. This Parallel Interface Box can be built for less than \$20 in parts, and within an afternoon. The rest of this tutorial article is broken down as follows:

- [Parts List with Potential Vendor Source](#)
- [Theory of Operation](#)
- [Construction](#)
- [QBasic Programming](#)
- [References](#) For this card, Microsoft's **QuickBasic** will be used to do some simple programs.

Parts List and Potential Vendor Source

Below is a parts list I used for my construction. Additionally, I list the source from which I bought it from, along with the vendor part number and cost. I did some shopping around, and found Digikey a possible single-source for parts. However, I don't find Digikey to be the cheapest place for parts. The 34-pin header and connector, and header, housing and crimps are a bit pricey. Jameco, a cheap source, does not have such parts. These are not critical parts, and you can substitute them for whatever part you might have in your junk box.

TABLE 1A: PARALLEL INTERFACE BOX PARTS LIST

PART DESCRIPTION	VENDOR PART	PRICE (1995)	QUANTITY
74367 HEX BUFFER	DIGIKEY DM74LS367-AN-ND	0.95	3
RED T1 3/4 LED	DIGIKEY #MR3050QT-ND	0.45	8
YELLOW T1 3/4 LED	DIGIKEY #MR3350QT-ND	0.45	8
DB25 MALE CONNECTOR	DIGIKEY #225M-ND	0.92	1
DB25 MALE HOUSING	DIGIKEY #925GP-ND	0.55	1
220 OHM RESISTORS			9
0.100 CENTER HEADERS	E.G. DIGIKEY #WM4000-ND		
0.100 CENTER HOUSINGS	E.G. DIGIKEY #WM2601-ND		
0.100 CRIMPS	E.G. DIGIKEY #WM2200-ND		
RED BINDING POST JACKS	DIGIKEY #J117-ND	0.59	9
BLACK BINDING POST JACKS	DIGIKEY #J118-ND	0.59	2
34 PIN STRAIGHT LEAD WIREWRAP HEADER	DIGIKEY #CHW34G-ND	4.20	1
34 PIN CONNECTOR	DIGIKEY #MSC34G-ND	4.76	1
RIBBON CABLE 34 PIN	DIGIKEY #MC34M-X-ND	10.39	5 FT
WIRE WRAP			
ENCLOSURE BOX 6x3 3/16x1 7/8	RADIO SHACK 270-223	3.19	1

Theory of Operation

The parallel port on the IBM PC is a 25-pin female port. The figure and table below describes the pin functions:

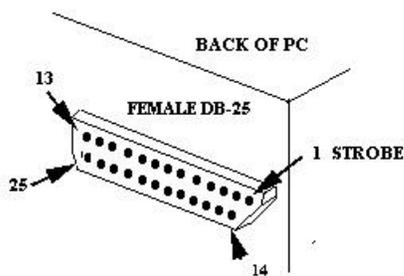


TABLE 2: PARALLEL PORT PINOUT

PIN NO.	FUNCTION	TYPE
1	STROBE	CONTROL
2	DATA BIT 0	OUTPUT
3	DATA BIT 1	OUTPUT
4	DATA BIT 2	OUTPUT
5	DATA BIT 3	OUTPUT
6	DATA BIT 4	OUTPUT
7	DATA BIT 5	OUTPUT
8	DATA BIT 6	OUTPUT
9	DATA BIT 7	OUTPUT
10	ACKNOWLEDGE	STATUS
11	BUSY	STATUS
12	PE: PAPER TRAY EMPTY	STATUS
13	PRINTER ON-LINE	STATUS
14	AUTO LINEFEED AFTER (CR) CARRIAGE RETURN	CONTROL
15	PRINTER ERROR	STATUS
16	INITIALIZE PRINTER	CONTROL
17	SELECT/DESELECT PRINTER	CONTROL
18-25	UNUSED/GROUND	

The parallel port has 4 function types for a total of 25 pins: data (8 pins), control (4), status (5) and ground (8). To understand the function of the data, control and status types, consider what happens when you print something on your printer. The printer prints out alphanumeric characters onto paper (thus using the data lines). Sometimes it does a carriage return and linefeed (hence using the control lines). Sometimes, the printer doesn't print because you ran out of paper, or you forgot to have the printer on-line (status lines). Thus the printer has a number of input and output related function types. The 8 data lines are used for 8 digital OUTPUT lines. For example, you can turn on 8 different motors. The 5 status lines are used for 5 digital INPUT lines. Thus you can interface 5 different sensors, like pushbuttons. The 4 control lines can be used for 4 additional digital output lines (thus 4 more motors!).

Parallel Port Addresses

Each device in a computer has an assigned memory address. For example, your CD-ROM occupies an address. So does your hard and floppy drives. The parallel port is no exception. The IBM Technical Reference Manual describes 2 possible addresses for the parallel port. If you have an old PC with a monochrome display adapter (the old Hercules Green Monitor), the address is 3BCH (956 decimal). If you have a computer with a CGA, EGA, VGA or Super VGA display card (most common for computers after 1990) the parallel port address is 378H (888 decimal). This address is important because we can then write appropriate software. This will become clearer in the QBasic Programming Section. If you are not sure about the address of your parallel port consider the following: If

your computer is a 286, 386, 486 or Pentium and you have a color monitor (VGA or SVGA), your computer is probably using 378H as the parallel port address.

TABLE 3: PARALLEL PORT ADDRESS

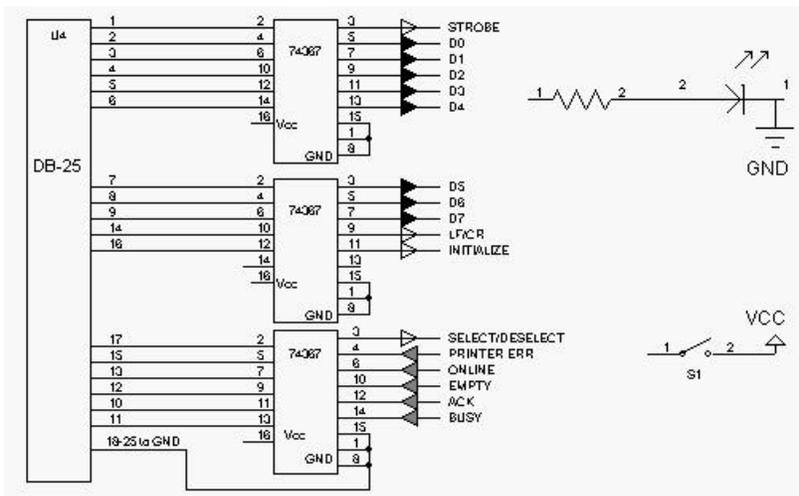
ADAPTOR	DATA	STATUS	CONTROL
NON-MONO E.G. VGA AND SVGA	378h (888d)	379h (889d)	37Ah (890d)
MONO DISPLAY CARD	3BCh (956d)	3BDh (957d)	3BEh (958d)

NOTE: The suffix "h" will be used to denote hexadecimal numbers. The suffix "d" will be used for decimal numbers. This table information will be useful later in the QBasic Programming section.

Circuit Construction

Schematic

The figure below describes the basic circuit.

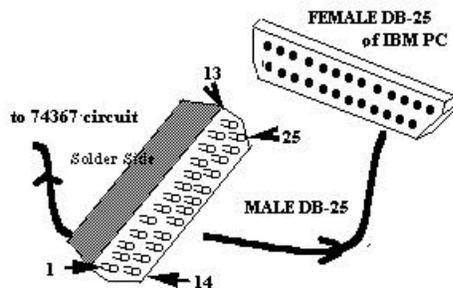


You can click below to download the Adobe Acrobat format of the schematic. I've tried to get a more generic format e.g. GIF or PS but with no success (as seen to above fuzzy picture). I tried to provide a Windows Meta File format but have found bad results e.g. doesn't load into Word well. If you don't know what Acrobat is, visit [Adobe](#) to download a copy of its FREE PDF-file viewer.

Download the [Acrobat \(PDF\) file of the schematic. Recommended! 6695 Bytes](#)

As one can see there are 8 digital output lines D0 to D7. These are symbolized by a black arrow head. There are also 4 additional output lines: Strobe, LF/CR, Initialize, Select/Deselect. These are symbolized by a hollow arrow head. One should use these digital output lines with some caution. These lines may require some additional programming steps to use. If 8 digital output lines are enough for your application, then try to avoid using these lines. For example, the Strobe line is an important line in itself. It

assures that the parallel data on lines D0 to D7 are ready to be sent out *simultaneously*. One can also see 5 digital input lines: Printer Error, Online, Empty, Acknowledge and Busy. These are symbolized by grey arrow heads. The circuit requires an external 5 V source. This is connected to Vcc (pin 16) and GND (pin 8) of the 74367. The DB-25 connector pin numbers are those of the male connector part. This connector will plug into the PC parallel port. This is emphasized in the figure below



The 74367 and Current

The circuit makes use of the 74367 Hex Driver Buffer chip. This protects the motherboard from sinking or sourcing too much current. The parallel port can sink or source about 5 mA of current. Thus one cannot directly hook up a motor, for example, across D0 and GND. A motor can easily demand an amp of current! If one would attach a motor directly across D0 and GND, you could possibly **blow** and damage your motherboard! With the 74367 as a protective buffer, if your application draws or sinks too much current, these chips would blow before surging your motherboard. You can think of these chips like electrical "fuses". These buffer chips are commonly used to interface real world devices to a PC. An additional note: There are many types of 74367 chips. There is the plain 74367. Also there are the 74LS367, 74HCT367 and other TTL variants. The difference between them is voltage that defines a HI and LO signal. I recommend using the 74HCT367. For output, a HI is defined by a minimum voltage of 4.9 V. A LO is defined by a maximum of 0.1 V. For input, a HI is defined by a minimum of 2.0 V and a LO is defined by a maximum of 0.8 V. The HCT series also allows immediate interfacing to CMOS and TTL type chips.

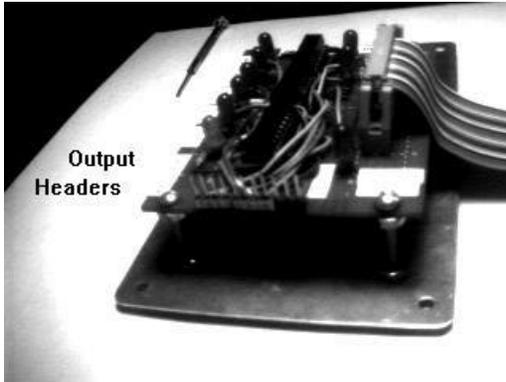
Interface Box

The interface box can be seen in the following photo.



I used 8 red colored binding posts connected to D0 to D7. These posts allow me to hook up an external real-world device such as a transistor very easily. I can use alligator clips or banana plugs, clipping them to these binding posts. I drilled out 8 holes on the Radio Shack cabinet box and inserted these binding posts. I then soldered the binding posts' ends to crimped wires (Digikey part WM2200). I inserted the crimped wires to a housing (Digikey part WM2601-ND). The housing them fits snugly across a header (Digikey part WM4000-ND). Lastly the header is soldered across the D0 to D7 lines of the 74367. One

can see the header pins in the following photo.



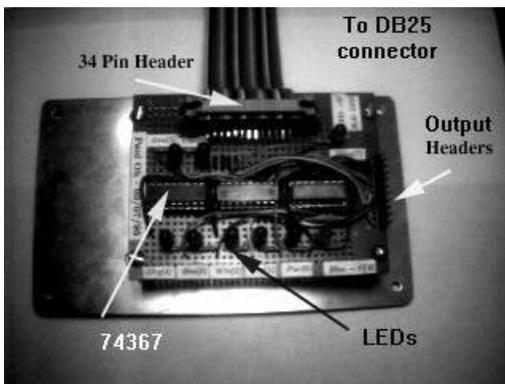
In my prototype, I did not design for digital inputs, but one can easily use the same procedure of binding posts-crimped wire-housing-header to the 5 status lines.

Interfacing Real World Devices

In the schematic, I show another real world device, an LED and a Switch. One can use a 220 Ohm resistor in series with the LED. I soldered 8 red LEDs to the 8 digital output lines D0 to D7. In the QBasic Programming Section, I will show a QBasic program which will sequence through all 8 LEDs, lighting them up. Also in the schematic is a single-pole-single-throw (SPST) switch. This demonstrates how the circuit can be used for digital input detection. As mentioned previously, you cannot hook up a motor directly across a digital output line and ground. A real world device such as a motor draws too much current. Instead, one would hook up a transistor, like a 2N2222. You can connect the transistor base to a digital line. The Source would be hooked up to the motor's power supply. The transistor's emitter is then hooked to the ground of the 74367 circuit, and the ground of the Motor's power supply. The decision on using a transistor like the 2N2222 would depend on the current draw of the motor. The 2N2222 can handle upto an amp or so. This is fine for, say a Radio Shack 99 cents toy motor.

Additional Construction Tips

As mentioned in the Part List section, there are many optional parts. I used the 34 pin straight lead wirewrap header and connector. I wanted to have a "neat" bus of wires to solder to the DB-25. I also wanted to wirewrap for quick circuit troubleshooting. Of course, there is no real need to use these parts. I listed them as a matter of convenience to the reader. Also, I used a yellow LED for the Strobe output. As one will read in the QBasic programming section, the Strobe is an important signal. I wanted to use a yellow LED to distinguish it from the other 8 red LEDs. The LEDs, and 34 pin connector are seen more clearly in the photo of the constructed circuit board:



QBasic Programming

The section describes how to light the 8 LEDs connected across the 8 digital output lines D0 to D7. Recall, that the 74367 chips require an external 5 V power supply. After plugging the DB-25 to PC's parallel port, turn on the 5 V power supply. You can now launch QBasic to test the board. Try the following program:

```

100 REM TESTING THE 8 DIGITAL OUTPUT LINES
110 PORTBASEADDR = 888: REM FOR NON-MONO GRAPHICS CARDS
120 FOR X = 0 TO 255
130 OUT PORTBASEADDR, X
140 FOR DELAY = 1 TO 500: NEXT DELAY
150 NEXT X

```

If everything has been assembled correctly and the software properly compiled, then one should see the 8 LEDs should "count" from 0 to 255 in a binary fashion. Recall from Table 3, the parallel port address is assigned 888d (378h). We thus can access the 8 digital lines D0 to D7 by writing a number between 0 and 255 decimal. In QBasic, we do this using the OUT statement. In Turbo C, I believe the equivalent statement is OUTPORT. One thing to also notice is how the Strobe line flashes each time data is sent along D0 to D7. It is a kind of syncing signal. Strobe ensures that all data along D0 to D7 are sent at the *same* time, hence the name *parallel* port. This is important, especially in timed applications. For example suppose you made an 8 motor driven robot arm. If you want a synergistic, coordinated motion, you should make sure that all data along D0 to D7 hits all 8 motors at the same time.

Using the Control lines as Additional Digital Outputs

Table 3 gives the address of the control lines as 958d. You can use OUT as well, with the following table:

TABLE 4: CONTROL LINE BIT ASSIGNMENTS

CONTROL BIT	FUNCTION
BIT 0 - STROBE	LO=NORMAL, HI=D0-D7 STROBED
BIT 1 - LF/CR	LO=NORMAL, HI=DIGITAL OUTPUT
BIT 2 - INIT	LO=NORMAL, HI=DIGITAL OUTPUT
BIT 3 - SELECT/DESELECT	LO=NORMAL, HI=DIGITAL OUTPUT

Thus, you use the statement OUT 958, 2 to turn on Bit 1, the LF/CR line for digital output. OUT 958, 6 would turn on Bits 1 and 2.

Using the Status lines for Digital Input

Table 3 notes that the address of the status line is 957d. We can use QBasic's INP statement to read signals into the computer. INP is used in conjunction with the following table:

TABLE 5: STATUS LINE BIT ASSIGNMENTS

CONTROL BIT	FUNCTION
BIT 0-2 NOT USED	
BIT 3 - PRINTER ERROR	LO=PRINTER ERROR, HI=NO ERROR
BIT 4 - ON-LINE	LO=NOT ONLINE, HI=PRINTER ONLINE
BIT 5 - PAPER	LO=PAPER, HI=NO PAPER
BIT 6 - ACKNOWLEDGE	LO=DATA SENT, HI=NORMAL
BIT 7 - BUSY	LO=BUSY, HI=NOT BUSY

Suppose one uses the switch in the schematic, soldered to ONLINE. The QBasic statement: SWITCHSTATUS = INP(957) would return the status of the switch. That is, if the SPST switch is pressed, then SWITCHSTATUS would return a number 16d. If the switch is not pressed, then SWITCHSTATUS should be 0.

Where to go from here?

Congratulations if you got this far! So what can and cannot you do with this Parallel Port Interface Box? Well, 8 digital output lines (plus possibly 4 others), and 5 digital input lines affords you a many possibilities. First can you do? Well, this box is a **cheap** and quickly constructable unit. It immediately plugs into the back of your PC. For an example real world application, suppose you wanted to make a home security device. You can use 5 input sensors: strain gages (for smashed windows), sound sensors (for loud sounds), heat sensors (for fires), another strain gage (for detecting if someone is standing on your front door's "Welcome" mat, and an infrared emitter/detector sensor (to detect if someone has walked into hallway). You can use the 8 digital lines to do many event driven things. For example, if the heat sensor is triggered, you can use 1 digital output to turn on another circuit to dial the fire department. If the Welcome mat detects someone standing on it, you can use yet another digital output to turn on the porch light. The possibilities are limitless... almost. Lastly, you could hook up the 8 digital outputs to a digital-to-analog (DAC) converter. You can then have analog output, which can give you a voltage range, to say, do motor control! What can't you do? Well, 5 digital inputs is 3 shy of a full byte. This is unfortunate. Most digital-to-analog converters require a full 8 bits. Thus, you cannot use a DAC for analog input. Oh well... The alternative is to have a 4-bit DAC. You will just have less resolution. Best of luck, and enjoy!

Stay Tuned!

I plan to create some PCB (Printed Circuit Board) artwork for this circuit. I will make the artwork freely available, in the spirit of the World Wide Web. I am shocked at some

companies that sell a parallel port interface box for \$100s, considering you can make one yourself for less \$20. Who knows, I might contract out the PCB artwork to a manufacturer and sell this circuit on the web for about \$15 or so. Maybe I will make some companies cry... let me know if you think this is a good idea :-) Email me!

References

- The Robot Builder's Bonanza by Gordon McComb *Tab Books* ISBN 0-8306-0800-1

NOTE: This book's schematics have errors! Some are clearly noticeable, but many are not. I am quite confident of my schematic though.

Click [here](#) to go Paul's Main Page

Click [here](#) to go our Robotics Lab

Click [here](#) to email me your comments and suggestions