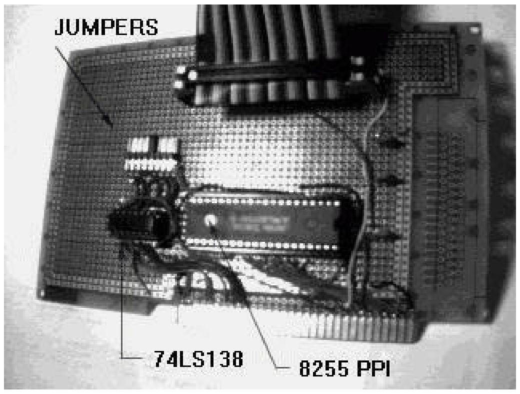
8255 PPI IBM PC Interface Card



Introduction

The 8255 Programmable Peripheral Interface (PPI) is a versatile and easy to construct circuit card the plugs into an available slot in your IBM PC. Such a card allows you to do both digital input and output (DIO) to your PC. For example, you may want to have your PC turn on a switch, or have a switch electronically activate your PC to execute a program. The 8255 has 3 8-bit TTL-compatiable I/O ports. Thus technically you could control up to 24 individual devices. For example, with the circuit description that follows, you could build a home security device, with a digital input detecting if someone rang your doorbell, and having a digital output turning on a light switch, the other remaining 22 DIOs could then be used to detect for sound detection (broken glass say), doors being opened, digitally dial 911, detect smoke etc. Hopefully this preamble has excited your curiosity and imagination. This article is broken down as follows:

For this card, Microsoft's **QuickBasic** will be used to do some simple programs.

Parts List and Potential Vendor Source

Below is a parts I used for my construction. Additionally, I list the source from which I bought it from, along with the vendor part number and cost (in 1998). I did some shopping around, and found these places to offer the best tradeoff between price and single-source parts availability - hopefully this saves you time! The first table is what is necessary to make a card. The second table is an additionally prototyping area circuit that interfaces with the card via a 40-pin ribbon cable. The parts in this table also include a simple testing circuit which is described later. A note about the Digikey parts: I did not find such a part in Jameco. I had these lying around in my parts box. Other similar sockets could be used. But I recommend that at least one be a wirewrap type, since I use both soldering and wirewrapping for the 8255 Card circuit.

TABLE 1A: 8255 CARD PARTS LIST

PART DESCRIPTION	VENDOR PART	PRICE (1998)	QUANTITY
8255 PPI 40 PIN IC	JAMECO #52417	3.95	1
40 PIN WIREWRAP SOCKET	JAMECO #41179	1.65	1
74HCT138 3-TO-8 DECODER	JAMECO #44927	0.29	1
16-PIN SOLDER SOCKET	JAMECO #37372	0.12	1
0.1 uF CAPACITORS	JAMECO #15270	1.00 FOR 10	3
40 PIN HEADER CONNECTOR	DIGIKEY #CHW40G-ND	5.11	2
40 PIN SOCKET CONNECTOR	DIGIKEY #CSC40T-ND	2.45	2
2 ROW 8-POSITION JUMPER	JAMECO #109516	0.39	1
SHORTING BLOCKS	JAMECO #22023	0.20	2
PC BUS PROTOTYPING CARD	JAMECO #21531	17.95	1
RIBBON CABLE 40 PIN	JAMECO #105726	20.00 FOR 25 FT	3 FT
WIRE WRAP			

TABLE 1B: BREADBOARD INTERFACE PARTS LIST

PART DESCRIPTION	VENDOR PART	PRICE (1995)	QUANTITY
6 IN. SOLDERLESS BREADBOARD	RADIO SHACK #276-174	12.49	2
74LS04 HEX INVERTOR	JAMECO #46316	0.29	1
LEDS T1-3/4	JAMECO #34745	0.19	8
220 OHM RESISTORS			
TERMINAL BLOCKS	JAMECO #99426	0.59	14

Theory of Operation

This section is organized as follows:

- The Base Address
- The Control Modes
- The Jumper Settings

The Base Address

This PC Interface Card plugs into any available 8 or 16-bit slot (also known as an AT-slot) on your PC's motherboard, just like a sound card or disk drive controller card does. Your CPU (Central Processing Unit) communicates with cards by knowing the card's address. From IBM's Technical Reference Manuals, one can learn that the motherboard allows for several available addresses for any cards you wish to plug into a slot. By physically using jumpers on the card, we can assign an address to the card. In software, we can tell the CPU what this address is (more about this in the Programming section).

TABLE 2: EXPANSION SLOT ADDRESSES

ADDRESS (HEX/DEC)	DESCRIPTION	ADDRESS (HEX/DEC)	DESCRIPTION
218-21F (536-543)	AVAILABLE	390-39F (906-927)	AVAILABLE
250-277 (592-631)	AVAILABLE	3AA-3AF (938-943)	AVAILABLE
280-2EF (640-751)	AVAILABLE	3B0-3BF (944-959)	AVAILABLE
300-31F (768-799)	AVAILABLE	3F8-3FF	COM 1

For notation purposes, a number with an H next to it denotes hexadecimal notation and plain numbers will denote denote plain decimal. For this card, I chose to use 280H (640 Decimal) for the address of the card. This is because many other cards tend to use the traditional 300H. By choosing 280H the probability of address conflict is a bit less. Thus 280H will known as the base address of the card. Of course, you can change this, by using the table above as a guide, and physically changing the jumper configuration on the board, as well as in your software program.

Control Modes

The 8255 allows for three distinct operating modes (Modes 0, 1 and 2) as follows:

- Mode 0: Ports A and B operate as either inputs or outputs and Port C is divided into two 4-bit groups either of which can be operated as inputs or outputs
- Mode 1: Same as Mode 0 but Port C is used for handshaking and control
- Mode 2: Port A is bidirectional (both input and output) and Port C is used for handshaking. Port B is not used.

Each port is 8-bit TTL-compatiable. As such, the 8255 can conceivable be configured to control 24 devices (1 bit/device). The various modes can be set by writing a special value to the control port. The control port is Base Address + 3 (hence 640+3 = 643 Decimal). The table below shows the special values needed to configure Ports A, B and C for either input or output:

TABLE 3: 8255 (CONTROL) MODE CONFIGURATION

CONTROL WORD (HEX)	CONTROL WORD (DECIMAL)	PORT A	PORT B	PORT C
80	128	OUT	OUT	OUT
82	130	OUT	IN	OUT
85	133	OUT	OUT	IN
87	135	OUT	IN	IN
88	136	IN	OUT	OUT
8A	138	IN	IN	OUT
8C	140	IN	OUT	IN
8F	143	IN	IN	IN

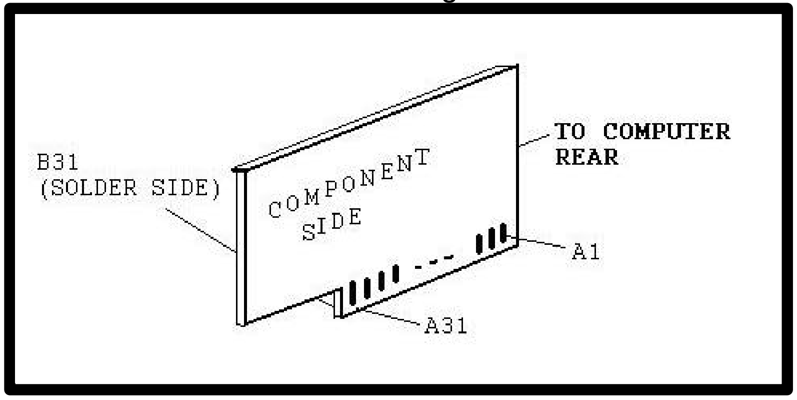
As mentioned the Control Port is Base Address + 3. Port A is always at Base Address; Port B is Base Address + 1; Port C is Base Address + 2. Thus in our example Ports A, B and C are at 640, 641 and 642 (Decimal) respectively. By writing say, 128 to the Control port will then configure the 8255 to have all three Ports set for output. This can be done using QuickBasic's OUT statement, for example:

```
110 BaseAddress = 640
120 PortA = BaseAddress
130 ControlPort = BaseAddress + 3
140 OUT ControlPort, 128
150 OUT PortA, 1
```

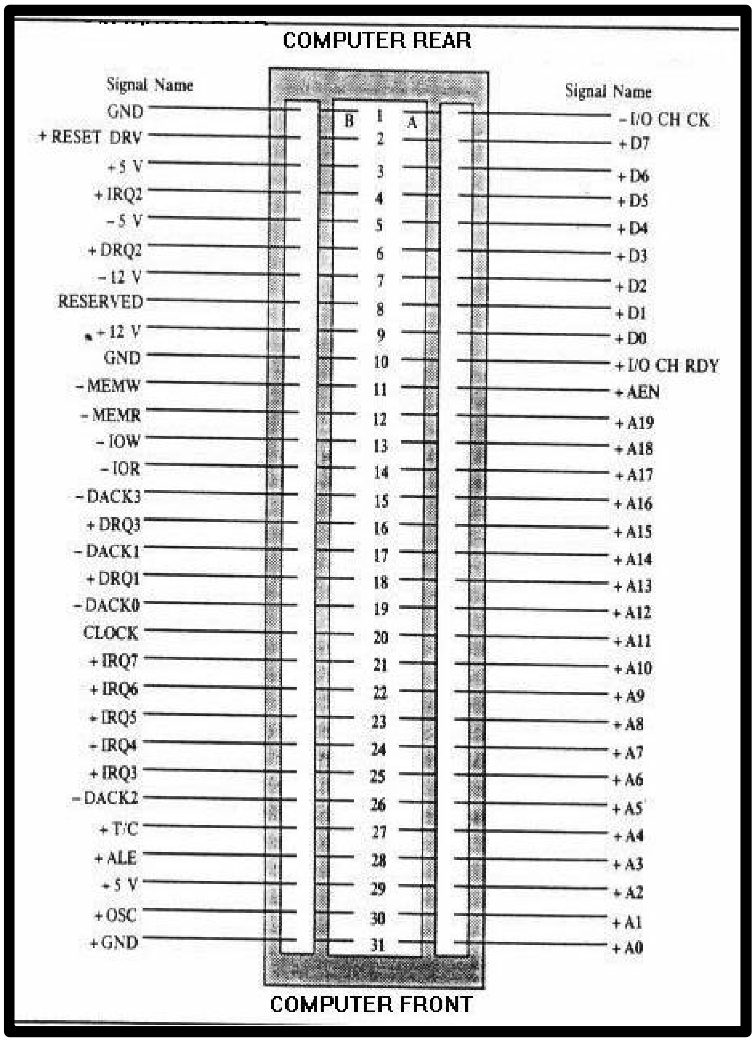
More software coding will be described later in the Programming Section.

Jumper Settings, the 74LS138 and IBM Expansion Bus

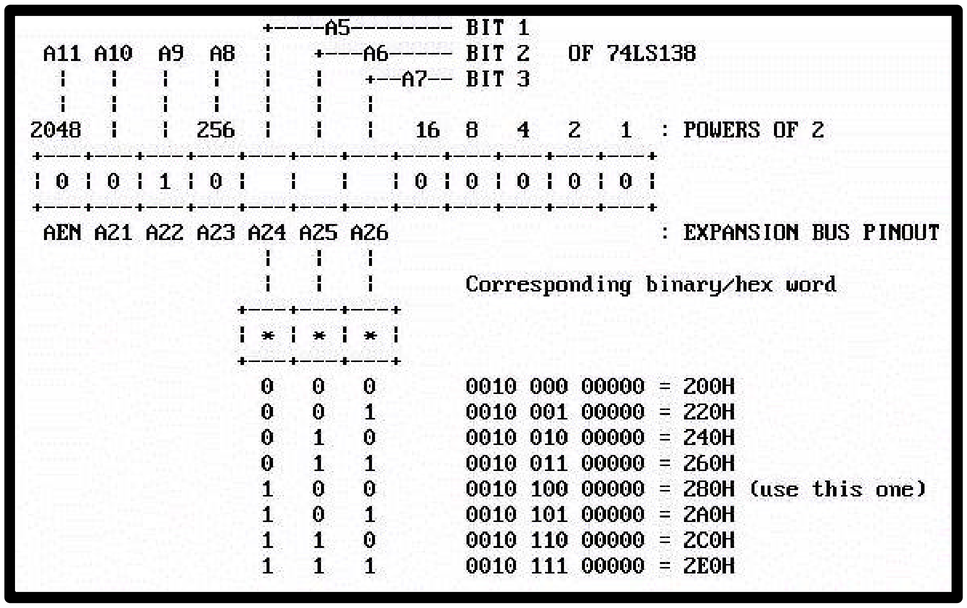
The 74LS138 is a 3-to-8 line decoder. It is a 16-pin IC, taking a 3-bit word on pins 1, 2 and 3. This leads to 8 possible combinations, and explains how the jumper can be physically set to provide for the 8 possible card address settings mentioned above. Examining the schematic in the Construction Section along with the explanation that follows will provide more insight on how this physically works. If you examine any PC card that plugs into the expansion slot on your motherboard, you can count 31 pads (typically gold) on each side of the card. See figure below:



Altogether there are 62 pads on this card. Such cards are also known as 8-bit ISA Bus (XT-style) cards. Due to the upward compatiability of PC systems, such cards can plug into expansion slots on XTs, ATs, 386s, 486s and Pentiums. The following diagram shows the pinouts of such expansion slots:



Getting back to the 74LS138... address lines A8 and A9 on the bus drive the control inputs to the 74LS138. Additionally, AEN (Address Enable), located at pin A11 is low when the card is recognized by the microprocessor. When A8 and A11 is low and A9 is high, the 74LS138 will decode bus address lines A5, A6 and A7 which are the 3-bit inputs to the 74LS138. The following picture may provide more insight:



```
                    +----A5-------- BIT 1
 A11 A10  A9  A8  ¦   +---A6----- BIT 2   OF 74LS138
  ¦   ¦   ¦   ¦   ¦   ¦   +--A7-- BIT 3
  ¦   ¦   ¦   ¦   ¦   ¦   ¦
2048  ¦   ¦  256  ¦   ¦   ¦   16  8   4   2   1  : POWERS OF 2
+---+---+---+---+---+---+---+---+---+---+---+---+
¦ 0 ¦ 0 ¦ 1 ¦ 0 ¦   ¦   ¦   ¦ 0 ¦ 0 ¦ 0 ¦ 0 ¦ 0 ¦
+---+---+---+---+---+---+---+---+---+---+---+---+
 AEN A21 A22 A23 A24 A25 A26                    : EXPANSION BUS PINOUT
                  ¦   ¦   ¦
                  ¦   ¦   ¦        Corresponding binary/hex word
                +---+---+---+
                ¦ * ¦ * ¦ * ¦
                +---+---+---+
                  0   0   0        0010 000 00000 = 200H
                  0   0   1        0010 001 00000 = 220H
                  0   1   0        0010 010 00000 = 240H
                  0   1   1        0010 011 00000 = 260H
                  1   0   0        0010 100 00000 = 280H (use this one)
                  1   0   1        0010 101 00000 = 2A0H
                  1   1   0        0010 110 00000 = 2C0H
                  1   1   1        0010 111 00000 = 2E0H
```

As a result, we can see that the jumper provides for 8 possible card address lines listed in the above figure. This in comparision with TABLE 2 shows that the only available only jumper position 5 is fully available, corresponding to 280H ((640).

Card Construction

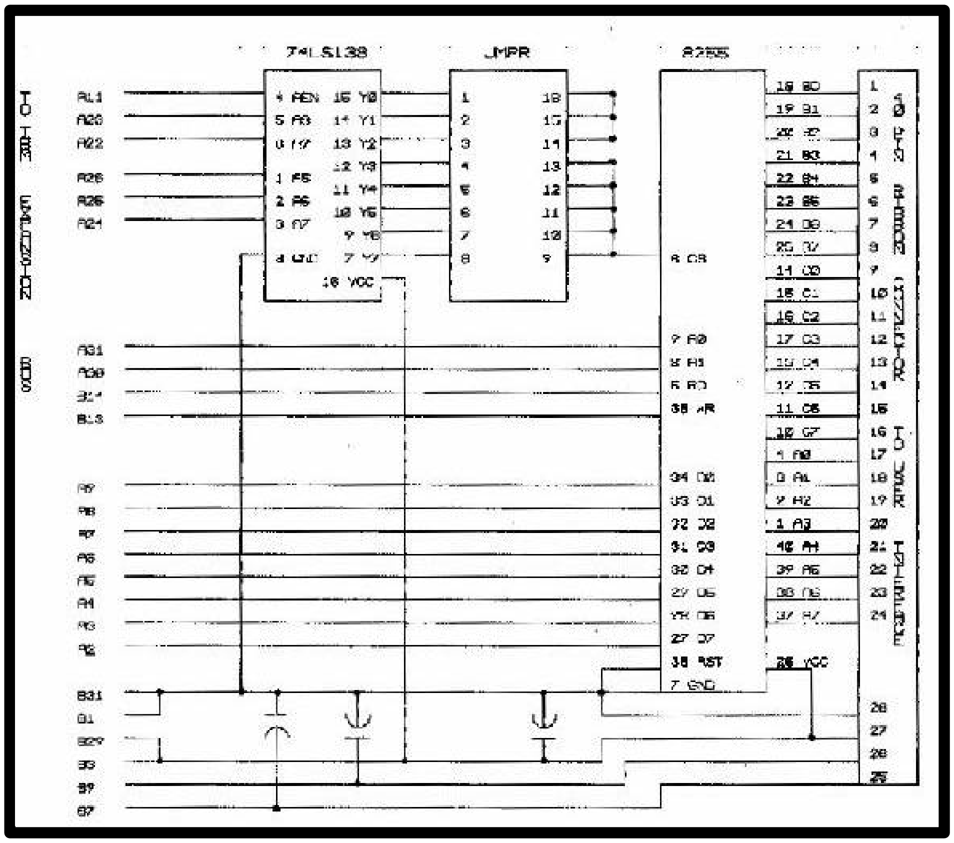
This section is organized as follows:

- General Intro and Schematic
- Card Construction Hightlights
- Breadboard Interface Construction

General Intro and Schematic

The schematic the figure below is relatively straightforward. I recommend however that you use a combination of soldering and wirewrapping using sockets for all IC component placement. A wirewrapping socket was used for both the 8255 and the 40-pin header. This provides for the easiest and most straightforward way of connecting these two components. Be careful to note the headers pinouts and how they lineup with the ribbon cable. That is, one row of the header will represent all even-numbered pinouts, while the other row represents all the odd-numbered pinouts. I used soldering (with 24 AWG) wire to connect the prototyping card's expansion bus pads. Note again that the component side represents pads A1 to A31 and the solder side's pads are B1 to B31. The Construction Highlights section that follows may give additional construction pointers.

Schematic



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! 10884 Bytes

Card Construction Highlights

The following photos (forgive the slight distortion) hopefully will give the builder a clearer idea of what is involved in the card's construction. I feel that such photos protray important information that a schematic alone cannot provide. Component placement is not critical.

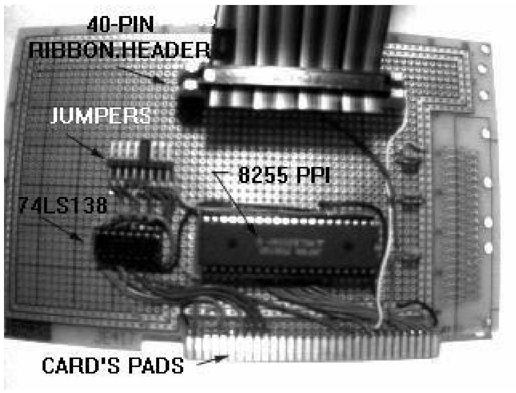


Figure A

This figure shows the overall view of the card and the component placement. If one looks closely, one can see the label 1 on both the 40-pin header and its ribbon cable (see also Figure C). Also the jumpers clearly show how jumper 5 has been shorted with a shorting block. One can also see the "tedious" soldering required along the card's pads (see Figure D as well). Again I emphasize that these pads represent pins A1 to A31 of the card (component side).

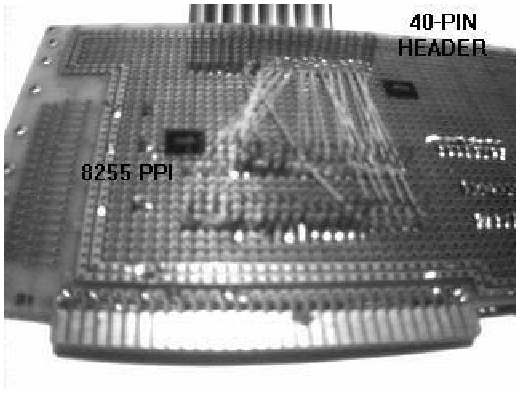


Figure B

This is the solder side. This view was provided to show the wirewrapping involved. For clarity during construction, I placed labels for pin 1 on both the 8255 wirewrapping socket and the 40-pin header. Recall that with this 40-pin header, one row represented all odd numbers i.e. the ribbon's 1st, 3rd, 5th, 7th etc wire, while the other row represents all the even numbers, i.e. the ribbon's 2nd, 4th, 6th etc wire.

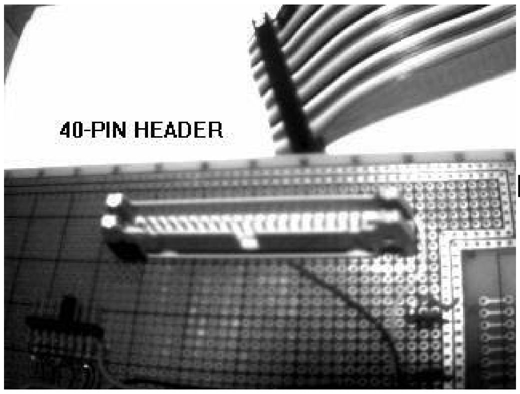


Figure C

Another view of the 40-pin header, but with the ribbon cable detached. Note that this is a polarized type of header socket.

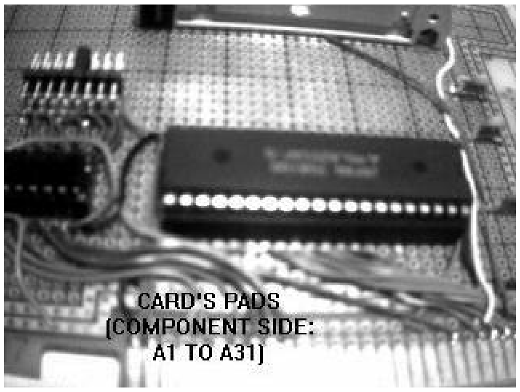


Figure D

This is a clearer view of the prototyping card's 31 pins (A1-A31) and the soldering/wiring involved.

Breadboard Interface Construction

Breadboard

This section is necessary for two reasons. First, this breadboarding inteface provides a way to test your 8255 card. Second, it facilitates the accessability of the 24 I/O lines by using the breadboard and screw terminal blocks. As the Figure (top) shows, the 24 I/Os are brought out from a 40-pin header to the terminal blocks, all on a single breadboard. Recall that in the card's construction in the sections above a 40-pin ribbon cable and header connector were used to connect to the 8255's Ports A, B and C. It is this opposite end of the ribbon cable that is brought to the breadboard and terminal blocks. It is simply a matter of matching up the correct ribbon wire with a terminal block. The Figure (botton) also shows a simple test circuit, the schematic of which is provided below. It uses all 8-bits of Port A configured for output. The Programming section below will show how to light up the LEDs, which would indicate that Port A is working properly.

QBasic Programming

Taking care to check all your wiring, now it is possible to check if your circuit indeed works. Insert the 8255 card into an empty slot in your PC's motherboard (assuming you turned off the power first!). Make sure the card is seated properly and the wirewrapping socket pins do not come into contact with any other cards. Launch QBasic (I have not tried testing under Windows) under a DOS environment. Try executing the following code:

```
100 REM 8255 PPI SET PORTS A,B,C TO OUTPUT
110 BASEADDR = 640: REM 280H/JUMPER AT POSITION 5
120 PORTA = BASEADDR
130 CNTRL = BASEADDR + 3
140 OUT CNTRL, 128
150 FOR SGNAL = 0 TO 255
160 OUT PORTA, SGNAL:PRINT "DECIMAL="; SGNAL
170 FOR DELAY = 1 TO 500: NEXT DELAY
180 NEXT SGNAL
999 END
```

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.

Congradulations if you have gotten this far! In my experience troubleshooting should begin in checking if the card is seated nicely in its slot on the motherboard. Also check the jumper. Check your wiring.

Of course the above is a simple circuit, but I leave it to your imagination to come up with other possibilities given that you have 24 digital inputs and outputs. Things like motor speed control, switching on relays, reading thermistors, strain gages etc.

References

- 8255A/8255A-5 Programmable Peripheral Interface, *Intel* Order No. 231308-002
- Build This Experimenter's I/O Card, *Radio-Electronics*, June 1990 pp. 73-78

Click here to go Paul's Main Page

Click here to go to Columbia University's Robotics Lab

Click here to email me