

OT: EPP problems

Source: <http://sci.tech-archive.net/Archive/sci.electronics.design/2006-02/msg00756.html>

- *From:* Robert Baer <robertbaer@xxxxxxxxxxxxxx>
 - *Date:* Sat, 04 Feb 2006 07:14:34 GMT
-

Seems my computer motherboard is OK; other boards, including an ISA board that supports SPP/EPP/ECP all have the same problem.

No matter how one does the programming of the registers, during the time when a pin "wants" to be an input (or more accurately, does not act like an output), the equivalent resistance on that pin is about 2.4K (tested with 2.2K and 1.0K load to ground).

What follows is a different test program with comments:

```
' READLPT2.BAS
' Attempt to read data from parallel port using EPP protocol
DEFINT A-Z
PRNT = &H378 'set as needed
DATAs = PRNT + 0: STATUS = PRNT + 1: CONTROL = PRNT + 2 'SPP
' ^--r/w ^--read only ^--r/w (from IBM)
ADDRESSrw = PRNT + 3: DATArw = PRNT + 4 'EPP
' ^--pin 17 pulse* ^--pin 14 pulse* *=if only one used
BIT5 = &H20: PIN17 = &H8: NBIT5 = &HFF XOR BIT5
' The following pins (as output) must be high:
' Address Strobe=pin 17, Data Strobe=pin 14, Write=pin 1, Reset=pin 16.
' NOTE: After system boot, all pins default high except data pins and pin 17.
' CONTROL bit 5 defaults low.

CLS
LOCATE 1, 1
PRINT " First read STATUS bits (cannot write them) init"
SEN = INP(STATUS)
PRINT "D0:"; (SEN AND &H1) / &H1, "ghost Pin 01 /STROBE (reads inverted) 0"
PRINT "D1:"; (SEN AND &H2) / &H2, "ghost Pin 14 /AUTO FD (reads inverted) 0"
PRINT "D2:"; (SEN AND &H4) / &H4, "ghost Pin 16 /INIT 0"
PRINT "D3:"; (SEN AND &H8) / &H8, "Pin 15 /ERROR 1"
PRINT "D4:"; (SEN AND &H10) / &H10, "Pin 13 SLCT (from printer) 1"
PRINT "D5:"; (SEN AND &H20) / &H20, "Pin 12 PE (reads inverted) 1"
PRINT "D6:"; (SEN AND &H40) / &H40, "Pin 10 /ACK (reads inverted) 1"
PRINT "D7:"; (SEN AND &H80) / &H80, "Pin 11 BUSY (reads inverted) 0"
PRINT " Then read CONTROL bits (EPP names given)"
CTL = INP(CONTROL)
PRINT "D0:"; (CTL AND &H1) / &H1, "Pin 01 /WRITE (reads inverted) 0"
PRINT "D1:"; (CTL AND &H2) / &H2, "Pin 14 /DataStrobe (reads inverted) 0"
PRINT "D2:"; (CTL AND &H4) / &H4, "Pin 16 /RESET 0"
PRINT "D3:"; (CTL AND &H8) / &H8, "Pin 17 /AddressStrobe 0"
```

OT: EPP problems

```
PRINT "D4: "; (CTL AND &H10) / &H10, "no pin IRQ EN 0"  
PRINT "D5: "; (CTL AND &H20) / &H20, "no pin (added bidirectional for EPP) 0"  
PRINT "D6: "; (CTL AND &H40) / &H40, "ghost Pin 10 Interrupt 0"  
PRINT "D7: "; (CTL AND &H80) / &H80, "ghost Pin 11 /Wait 0"  
PRINT " Read SPP data:: HEX$(INP(DATAs)); ":";  
PRINT " Read EPP address:: HEX$(INP(ADDRESSrw)); ":";  
PRINT " Read EPP data:: HEX$(INP(DATArw))  
PRINT "According to BeyondLogic.org, one must initialize the nAddressStrobe."  
PRINT " nDataStrobe, nWrite and nReset to inactive high by setting SPP control"  
PRINT " port to XXXX0100."  
LCTL = CTL AND NBIT5 'force bit 5 low  
HCTL = CTL OR BIT5 'force bit 5 high  
HICTL = CTL AND &HF0 'mask hi nybble  
INIT = HICTL OR &H4 ' and force as noted
```

```
WHILE INKEY$ = ""  
SOUND 2000, 1  
FOR I = -32767 TO 32766  
OUT CONTROL, INIT 'PIN 14 goes low for 25nSec  
T = INP(DATArw) 'about 400nSec later, DATA lines go low for 3uSec  
NEXT I  
WEND  
PRINT "CONTROL port init: "; HEX$(INIT), "Reads: "; HEX$(INP(CONTROL))  
OUT CONTROL, &H0 'attempt to reset  
PRINT "PIN 14 goes low for about 25nSec, about 400nSec later, DATA lines go low"  
PRINT " for 3uSec; when high, can be pulled down to 2.4V by 2.2K to ground."  
SYSTEM
```

Granted, nothing is connected to any of the pins, but still, during "input" time, one should see the equivalent of an LSTTL (or LTTL) input, and not a heavy pullup (2.4K).
It would seem that the original designers of this scheme did not know how to do a true or complete tri-state of the output stage – and all other designers slavishly followed like sheep.

Has anyone connected an outside parallel data source to an EPP port (with the correct logic) and gotten it to work without interrupt programming?

It would be nice if there was full disclosure on this...

±