
 PARALLEL DATA TRANSFER - SORCERER/PIO/MICROBEE.
 =====
Frank Herne.
=====

[Ed: This is the only article I have seen interfacing a Sorcerer to a PIO bearing computer. Hence its value as at the very least as a starting point for other tinkerers. It is transcribed from hard copy, so beware any mistakes. Please write in with any ERRATA. Soon to come, an article about reading Bee tapes.]

This first bit is by way of a preamble and a sort of summary for those members into parallel data transfer. Yearbook 3 indicates that some users are not getting their parallel ports to work in 8-bit mode. There are two methods that I know work:

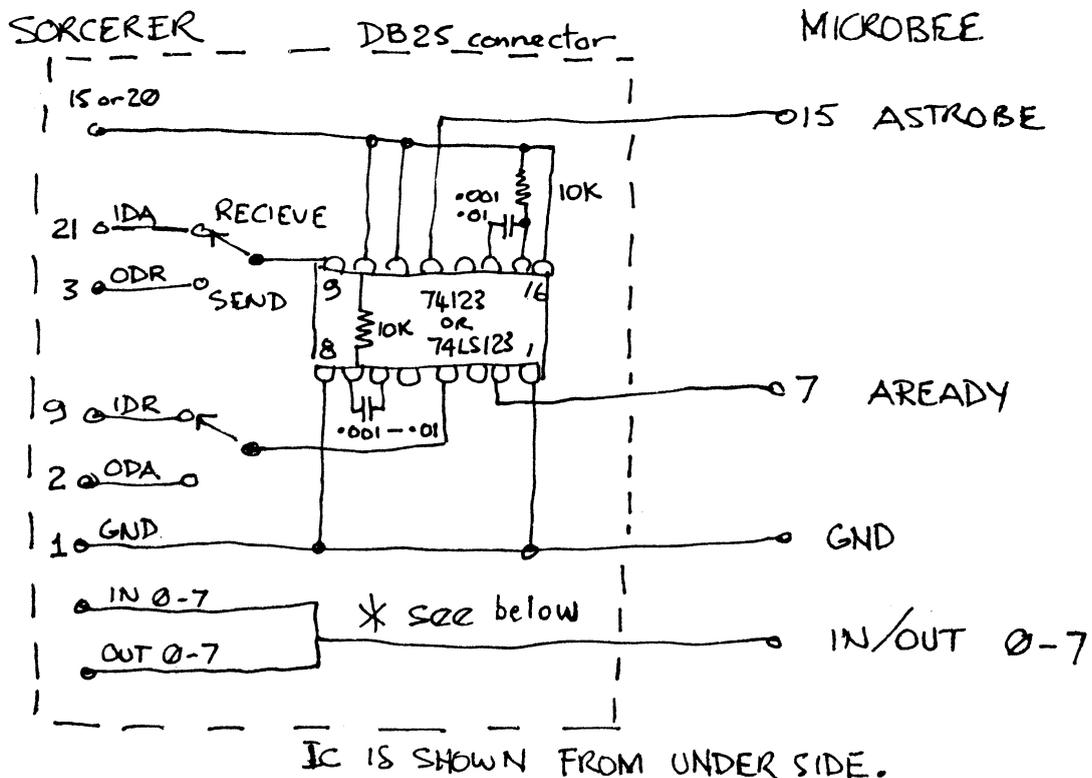
1. That by Ken Grimes (N/L No.41 Oct 82) for those who fear the removal of screws from the Sorcerer and,

2. My own internal mod to link pin 11 of the 74LS374 to pin 14 of the parallel port connector. (N/L No.54 Dec 83).

[Ed: 3. That by Ivan Reid also in N/L No.54]

4. Sorcerer to Sorcerer data transfer using the drivers also in this issue.]

The following is a method for data transfer between a Sorcerer and a Microbee. The problem to be solved is that of bidirectional data transfer along a common buss. It does not need either methods 1 or 2 above.



* Link input lines to output lines inside the Sorcerer DB25 connector.

The mod: Disconnect pin 1 of 74LS374 (8H/1, Mk1; 9H/1, Mk2) from ground and join it to pin 8 of 74LS74 (8F/8, Mk1; 10E/8, Mk2). Now the output buffer (Octal D-type Flip Flops) will only be enabled when data is sent (OUTPUT DATA AVAILABLE goes low) and is reset when Acknowledge

 (OUTPUT DATA ACCEPTED) goes high. The reason is so that the input and output bits can be sent along a common buss. And this is needed because a Z80 PIO (inside the MicroBee) uses bidirectional data lines. There are no drawbacks as far as I can see.

The above circuit diagram includes a switch and 74LS123 which can be mounted inside the DB25 connector. The switch selects the handshake lines for the Sorcerer depending on whether it is sending or receiving. The MicroBee does software switching.

Sending Sorcerer Basic Programs to MicroBee.

 Sorcerer: Add this line to your program - 0 NULL 10 :LIST :NULL0 :END
 Set the switch to 'send'.
 BYE to Monitor.
 >SE S=5<CR> <-- Delay is to allow the Bee to process the
 >SE O=P<CR> the lines. Isn't the Sorcerer wonderful!
 Bee: IN#1 ON<CR>
 Sorcerer: PP<CR> <-- Will not show on screen now.
 RUN<CR>

Watch the Bee Screen for errors which will occur in lines longer than 64 characters.

Sending MicroBee Basic Programs to the Sorcerer.

 Bee: Add this line to your program.
 50 OUT#1 ON :PRINT"PP";CHR\$(13) :LIST :OUT#0 :END
 Set the switch to 'receive'.
 Note the Bee sends a linefeed at the end of each line which must be filtered out as per the MicroBee Engineers Notebook or in the Sorcerer as below.

Sorcerer: BYE<CR>
 >EN FE00<CR>
 FE00: CD 1E FE 0A 28 F9 C9/<CR>
 >SE I=FE00<CR>
 Bee: RUN<CR>

Sending the MicroBee Pak to the Sorcerer.

 Sorcerer: Enter this code in a suitable location, e.g., 100H. The following routine loads the Bee Pak into a Sorcerer RAM Pac at C000.

```

21 00 C0          LD      HL,C000 ;Point to buffer
CD 1E E0          LOOP1  CALL   E01E ;Parallel in
FE C3            CP      C3      ;Wait for first byte
20 F9            JR      NZ,LOOP1
77              LOOP2  LD      (HL),A ;Store it
23              INC     HL      ;Point next
7C              LD      A,H
FE E0            CP      E0      ;Is it end?
C8              RET     Z      ;Yes!
CD 1E E0          CALL   E01E ;No! Get a byte
18 F5            JR      LOOP2 ;Repeat
    
```

Bee: (Written in Basic 'cause I dunno how a PIO works. It still takes less than 30 seconds.)

```

100 OUT#1 :REM SET PIO
110 FOR J=0 TO 10 :REM SEND SOME NULLS
    
```

] 10.

S.C.U.A. No. 55 JANUARY 1984.

```
120 OUT 0,0 :NEXT J
130 FOR A = 49152 TO 57343
140 OUT 0,PEEK(A) :REM SEND BYTE
150 NEXT A :OUT0 :END
```

Sorcerer: >GO 100<CR>

Bee: >RUN<CR>

When the prompt returns the job should be done.
