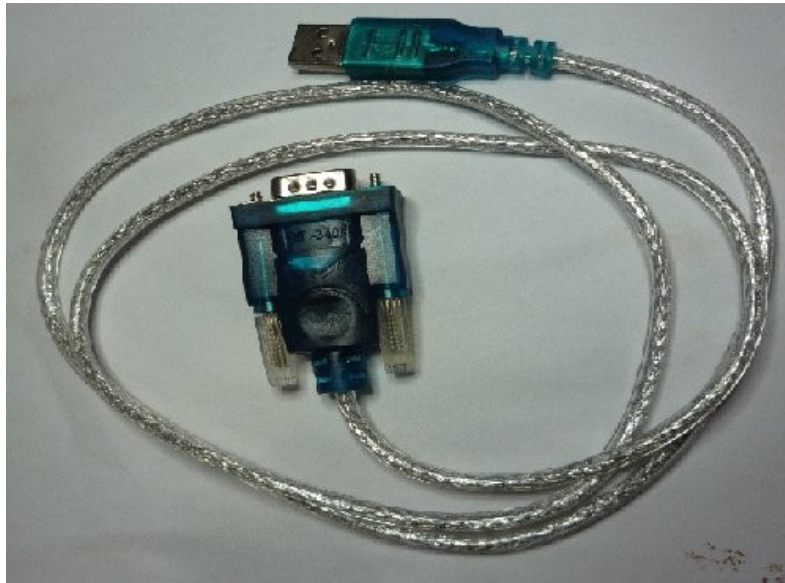


## PC – Microtan Serial I/O Connection

This captures the setting-up and testing of a PC to Microtan Serial I/O connection and provides an alternative to using the audio tape facility for program and data storage and retrieval.

### Hardware Set Up

My lap-top only has USB I/O ports so I needed a USB-Serial COM interface Cable fitted with a USB connector at one end and a DB-9 female connector at the other. These available at Amazon for a couple of pounds.



Installation in the windows environment is straight forward – plug it into a USB port and windows (in my case XP) installs the necessary drivers. It appears to my system as COM Port 8 (other USB ports appear as different COM ports).

A cable then needs to be made up with a DB-9 male connector at one end and a 14 way DIL header that will plug into the TANEX E1 socket as shown in Figure 1 below.

ACIA Pin		E1 Socket	Cable		D9 Pin
10	TX	3	Red	RX	2
12	RX	12	Blue	TX	3
2	GND	7	Black	GND	5
9	CTS	8		CTS	8
16	DCD	10		DCD	1
17	DSR	11		DSR	6

*Figure 1 - MT65-PC Cable connections*

Note: pins 7, 8, 10 & 11 on the E1 socket are all connected to ground.

## **PC Set-UP**

For testing purposes I used the Windows Hyper-terminal Application (found in Accessories – Communications).

After naming a new connection ‘Microtan’, the connection settings only gave me one option ‘COM8’ but if other serials ports like COM 1 had been available on the Laptop these can be selected if they are being used for this interface.

For testing purposes, I set the Port settings as 19200 Baud Rate, 8 Data Bits, Parity None, 1 Stop Bit, 'None' Flow Control.

To set the screen options, go into File – Properties and tab Settings. I used the ANSI Terminal Emulation and ticked the following ASCII Settings options:

- Send line ends with line feeds
- Echo typed characters locally
- Append line feeds to incoming line ends

## **Microtan testing**

There are 5 registers on the UART that control and execute serial I/O data transfer.

\$BFD3 – sets the port settings

\$BFD2 – controls data transfers

\$BFD1 – monitors status of registers /data transfers

\$BFD0(Read) - holds received data

\$BFD0(Write) – data to be transmitted

### MT65 Initialise Process

Using Monitor ‘M’ command, load \$BFD3 with #1F [19200 Baud, 8 N 1]

Press ‘ESC’ and modify \$BFD2 with #0B [DTR/RTS low]

### MT65 Transmit Test Process

Continuing from above

Using Monitor ‘M’ command, confirm \$BFD1 is \$10 ie bit 4 is set [\$BFD0(Write) is clear]

Press ‘ESC’ and modify \$BFD0 with #5A [character ‘Z’]

Press 'CR' to send character

Confirm on PC the character ‘Z’ has been received

Using Monitor ‘M’ command, confirm \$BFD1 is \$10 [ie character has been sent and register is clear]

### MT65 Receive Test Process

On the PC key a character on the Hyper-terminal screen

Using Monitor ‘M’ command, confirm \$BFD1 is \$18 [ie character has been received]

Press 'ESC' and confirm the \$BFD0 register contains ascii value of the character keyed on the PC

Press ‘LF’ to confirm \$BFD1 is \$10 [ie character has been read and register is clear]

Test complete