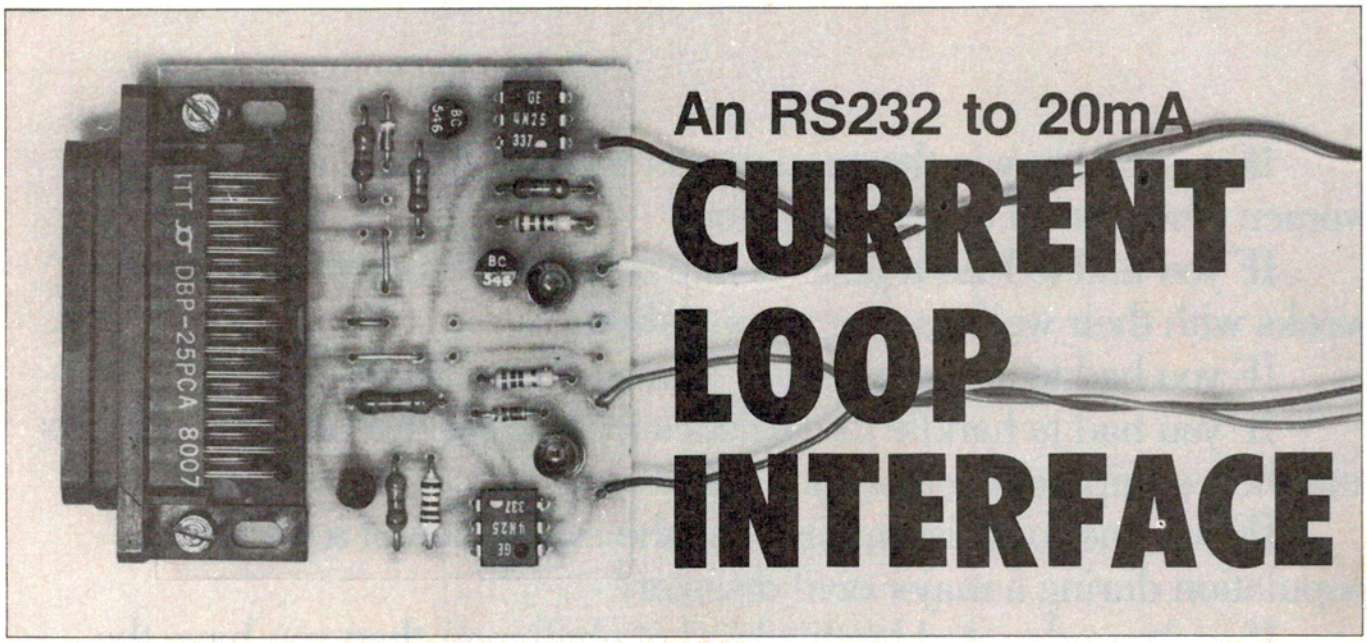


An RS232 to 20mA CURRENT LOOP INTERFACE

This project enables your computer to communicate with 'current loop' devices from its serial port.

Geoff Nicholls

MANY COMPUTER HACKERS built the early computer projects such as the ETI-632 VDU or (dare I say it) the EA 2650 computer series. These early projects used 20 mA current loops for their serial interfaces, and most are probably stuck away somewhere becoming dust encrusted. Why don't you dig the old beast out and see if it still works? You can run a Microbee remotely via the input/output redirection commands, or you can turn it into a terminal with the Net ROM or Telcom programs.

The interface uses opto-couplers to isolate the two transmission systems to the tune of 500 V. This allows complete flexibility in interconnecting the devices. Power is only required for the voltage-oriented RS-232 side of the interface.

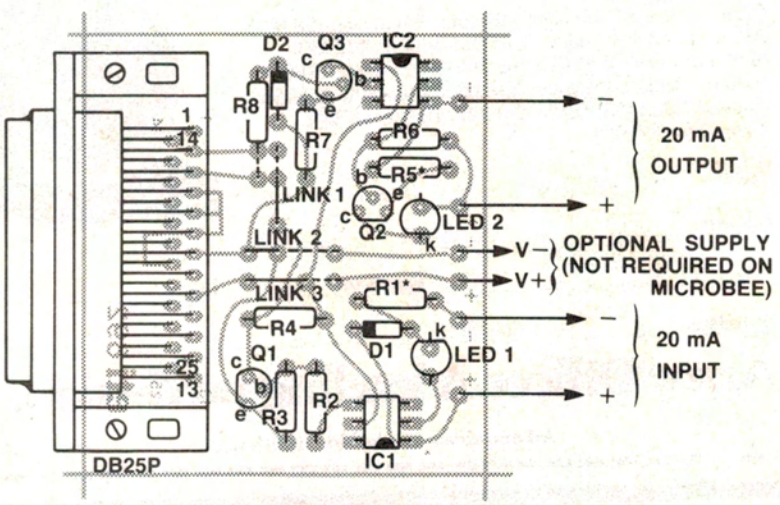
On the Microbee the power is taken from pin 9 of the RS-232 connector as 12 V is available there (on Series II and later 'Bees). Other computers probably won't have this facility so the pc board has a link to select pin 9 or an external supply.

A 'proper' RS-232 receiver requires a negative voltage for logical one signals. Many modern computers will work with a 'one' that only goes to zero volts (the Microbee for example). To allow either option to be used, another link selects between the RS-232 signal ground and an external negative supply for the RS-232 driver.

The current loop side is purely passive and requires the device on the other end of the current loop to provide the loop power supply.

There is one more link on the pc board to select the sex of the RS-232 interface. When RS-232 was first proposed it was intended to connect terminal devices to communications equipment such as modems. To enable a straight through cable to be used, two different pin standards were defined (hence the 'sex').

This was fine until people started using RS-232 connectors with equipment that was not either a terminal or a modem. These days a computer might be wired as either sex, so a pair of links has been provided to configure the RS-232 part of the interface as either sex. Note that R8 is actually the RS-



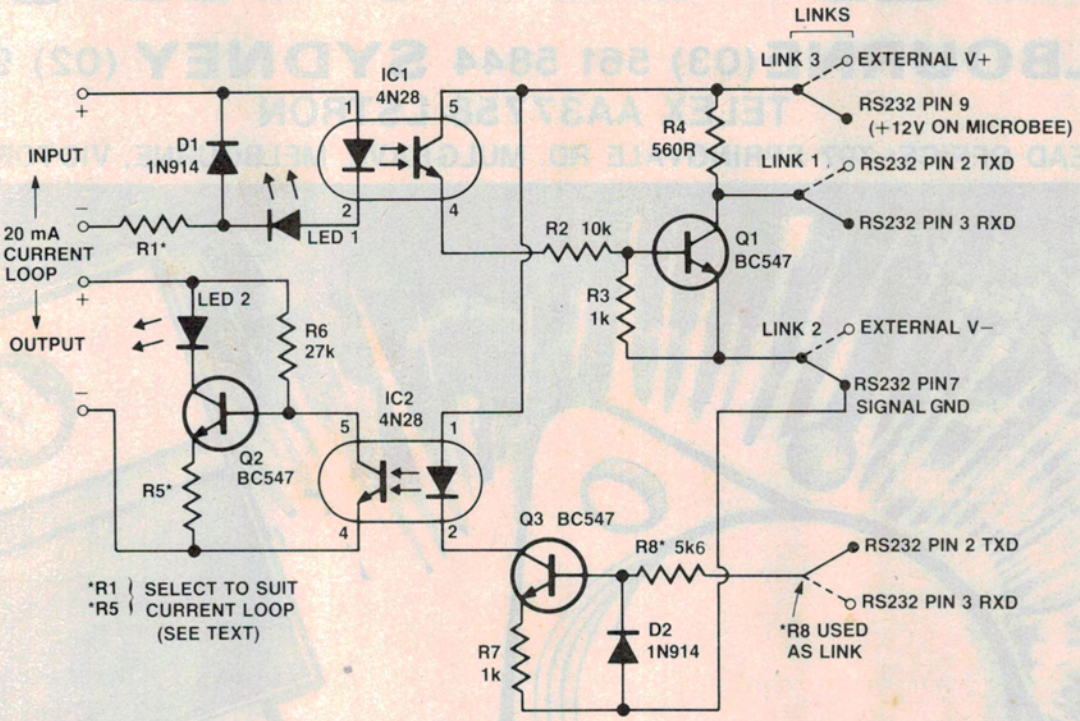
PARTS LIST — ETI-692

Resistors ..... all ¼ W, 5%
R1, R5 ..... see text
R2 ..... 10k
R3, R7 ..... 1k
R4 ..... 560R
R6 ..... 27k
R8 ..... 5k6

Semiconductors
D1, D2 ..... 1N914
LED1, LED2 ..... Red LED
Q1, Q2, Q3 ..... BC547
IC1, IC2 ..... 4N28

Miscellaneous
ETI-692 pc board; optional DB25P right angle pc mount plug.

Estimated price: $14-$16



232 receiver link. [The logical solution to the 'sex' problem is to make all devices as Data Terminal Equipment (DTE), i.e: pin 2 is always an output, and use cross-coupled cables exclusively].

Setting up the links

Firstly, decide how to configure the links on the pc board. If your computer (or other RS-232 device) is set up as DTE then pin 2 of the connector will be an output and pin 3 an input. This is the case for the Microbee or a terminal. For a DTE device, R8 goes to the closest hole and link 1 also goes to the closest hole. The overlay shows the DTE connection as the solid lines.

To configure as DCE (Data Communications Equipment, such as a modem) then R8 and link 1 should go to the far holes.

If your RS-232 receiver can do without a negative voltage for logical 'one' then connect link 2 to pin 7 of the connector. This is shown on the overlay as the solid link. If an external negative voltage is required then put link 2 in the dotted position and connect the negative supply to the pad near the edge of the board. (The supply should be negative with respect to pin 7 of the RS-232. Any device needing the negative-going input will have a suitable supply rail with a common ground.)

The positive supply is already provided on Microbees at pin 9 (Series II and later), and in this case link 3 should be as shown on the overlay. Other computers or terminals may have a positive supply on some other pin that can be connected with a short wire. Once again, the supply should have a common ground in pin 7.

Construction

Once the links have been sorted out the remaining components can be soldered in. If you are using a right-angle DB25P connector you should drill its mounting holes and bolt it down before soldering, to reduce any stress on the pins.

The only components to watch are the two ICs (the notches face away, toward the board edges), the 1N914 diodes and the LEDs, which should have the flats toward one another. The transistors' leads should go in without much bending.

The values of R1 and R5 have not been specified since they depend on the way the rest of the loop is wired. If you don't have data on the loop wiring then try around 180R for both to start with.

Testing

Make up a temporary loop supply by connecting a 1k resistor in series with a 20 V dc power supply.

Connect the supply to the 20 mA output, observing the polarity. Plug into the computer and try setting and resetting the RS-232 output. I have provided a short program for the Microbee to test both input and output sections. LED2 should be on when the ▶

HOW IT WORKS — ETI-692

The equivalence of the various signal standards is shown in Table 1. When the RS-232 output line is positive Q3 will be on and about 10 mA will flow through the LED in optocoupler IC2. The transistor in IC2 will be on, thus pulling the base of Q2 down, turning it off. Under these conditions the loop current will be set by R6, and will be under 1 mA, or essentially zero.

When the RS-232 line is at 0 V or negative the optocoupler transistor will be off, and R6 will bias Q2 on, passing the 20 mA or so loop current through LED2.

The input loop current flows through the LED in IC1 and also LED1. Diode D1 prevents reverse breakdown of either LED from reversed loop polarity. If the loop current is flowing then the transistor in IC2 will pass base current for Q1 thus dropping the RS-232 driver output down to the negative rail, as selected by link 2.

When no loop current is flowing R3 will keep Q1 turned off and the RS-232 output will be pulled positive by R4.

The handshaking lines RTS, CTS, DSR and DTR on pins 4, 5, 6 and 20 respectively are tied together to ensure the RS-232 port is always active. It was beyond the scope of the project to convert these to X-ON, X-OFF protocol!

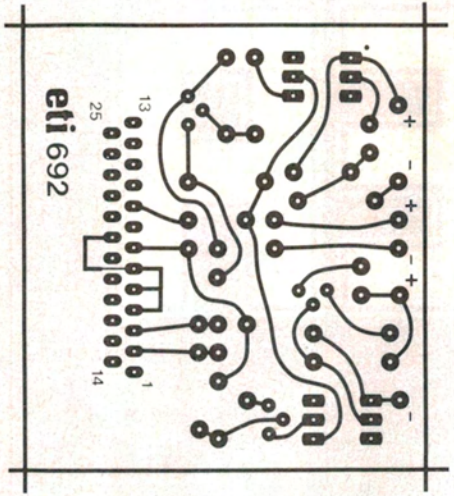
LOGIC LEVEL	TTL	RS-232C	20 mA
'ZERO'	0 to 0.8 V	5 to 15 V	0 mA
'ONE'	2.4 to 5 V	−5 to −15 V	20 mA

TABLE 1. Transmission standards

RS-232 output is 0 V (or negative) and off when the output is positive.

Connect the temporary loop supply to the 20 mA input through a switch and read the RS-232 port (or run the Input Test Microbee program). Toggling the switch should flash LED1 and change the RS-232 input's state.

Once the unit passes these tests it's ready to use!



PROGRAM LISTING FOR MICROBEE

Connect a 1k resistor in series with a 20 Vdc power supply to power the input and output 20 mA loops. Type in and run this test program. Note that the input test is slow to respond to changes due to the code required to isolate the RS-232 bit from the other bits in the Z-80 PIO port. This is done by the code from line 240 to line 270.

```
00100 REM **** Test Program ETI-692 ***
00110 REM  Written in Microworld Basic
00120 REM  By Geoff Nicholls, ETI
00130 PRINT "Output Test"
00140 PRINT "Press ESC then RETURN to run Input Test"
00150 PRINT "  Just press RETURN to toggle output."
00160 CURS 960:PRINT"LED 2";
00170 CURS 966:OUT 2,32:INPUT" on";A0$;
00180 IF A0$=CHR$(27) THEN 210
00190 CURS 966:OUT 2,0:INPUT"off";A0$;
00200 IF A0$<>CHR$(27) THEN 170
00210 PRINT\\"Input Test"
00220 CURS 960:PRINT"LED1";
00230 A=IN(2)
00240 FOR N=7 TO 5 STEP-1
00250 B=INT(2^FLT(N)):A=A-B
00260 IF A<0 THEN LET A=A+B
00270 NEXT N
00280 IF A>15 THEN LET L0$="off" ELSE LET L0$=" on"
00290 CURS 966:PRINT L0$;CHR(13);
00300 GOTO 230
00310 END
```