

Vic20- RS232 Interface

- ★ Allows the VIC to connect to Modems, Printers, VDUs, or any other RS232 compatible device
- ★ Converts TTL levels to true RS232
- ★ Provides full buffering for protection of computer
- ★ Full 'X line' interface possible as well as simple '3 line' interface

by Mark Brighton

RS232 is the name given to an industry standard form of serial data communication which is used on many peripheral devices to interface them with a computer.

A byte of serial data is represented by a series of transitions between +12V and -12V on a serial data line. The marks and spaces created by these transitions contain the information for the byte of data as well as some other signals, the purpose of which is to synchronise the receiving device to the serial data stream.

The format of a 'word' of data, including these synchronising signals, is as follows:

1. The start bit. This signal alerts the receiving device that a byte of data follows, and synchronises the receiver circuitry to the incoming data.

2. Data bits. This is the ASCII encoded data, and may consist of seven or eight bits as selected by the user. It is sent least significant bit first.

3. The parity bit. This is an optional error checking bit selected by the user to conform with the requirements of the receiving set. It may be set for odd or even parity, or disabled.

4. Stop bit(s). These are one or two bits of data which are transmitted at the end of a word to separate it from the next word.

The polarity of these signals may be selected by sending normal or inverted data, either of which may be required by different devices. Apart from the serial data lines (Sin and Sout), several other status and handshake signals are provided.

Those available on the VIC 20 are:

1. Data terminal ready (DTR). This signal is sent to indicate that the data

terminal is ready to send or receive data.

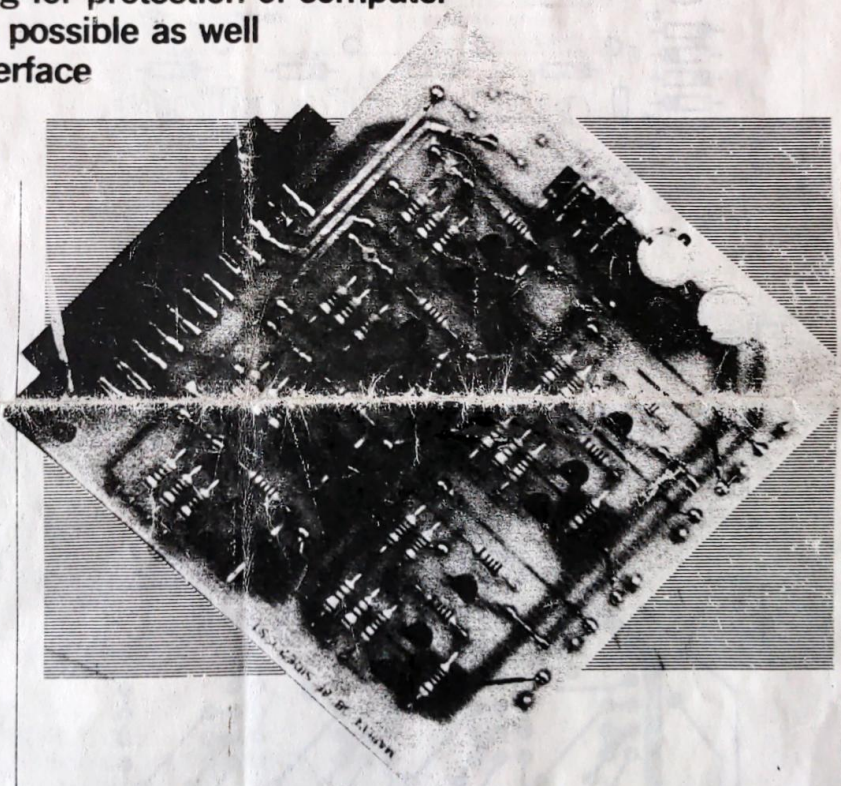
2. Data set ready (DSR). This indicates that the data set is ready to send or receive data.

3. Request to send (RTS). This signal tells the receiving device (usually called the 'data set') that the VIC (data terminal) wishes to send data.

4. Clear to send (CTS). This allows the data set to signal that it is ready to pass data from the data terminal.

5. Carrier detect (DCD). This lets the data set tell the data terminal that the communication link is established.

In addition to those lines already mentioned, there are two ground lines, protective ground and signal ground. Signal ground must always be connected, since RS232 requires that both devices have equal ground potential.



Any equipment which cannot meet this requirement is not RS232 compatible.

Circuit Description

The circuit consists of two transmit channels and four receive channels, with a power supply which provides approximately +12 and -12V from the 9V AC output on pins 10 and 11 on the user port.

Transmit Mode

Sout from the VIC (pin M, user port) is connected to the base of TR9 via a 10k Ω resistor, R17. As TR9 turns on, bringing its collector down to 0V, TR10 turns on, raising its collector voltage to +12V. The normal, or non-inverting, output is taken via R23, a 330 Ω resistor which limits the current that may be drawn from this output to about 30mA.

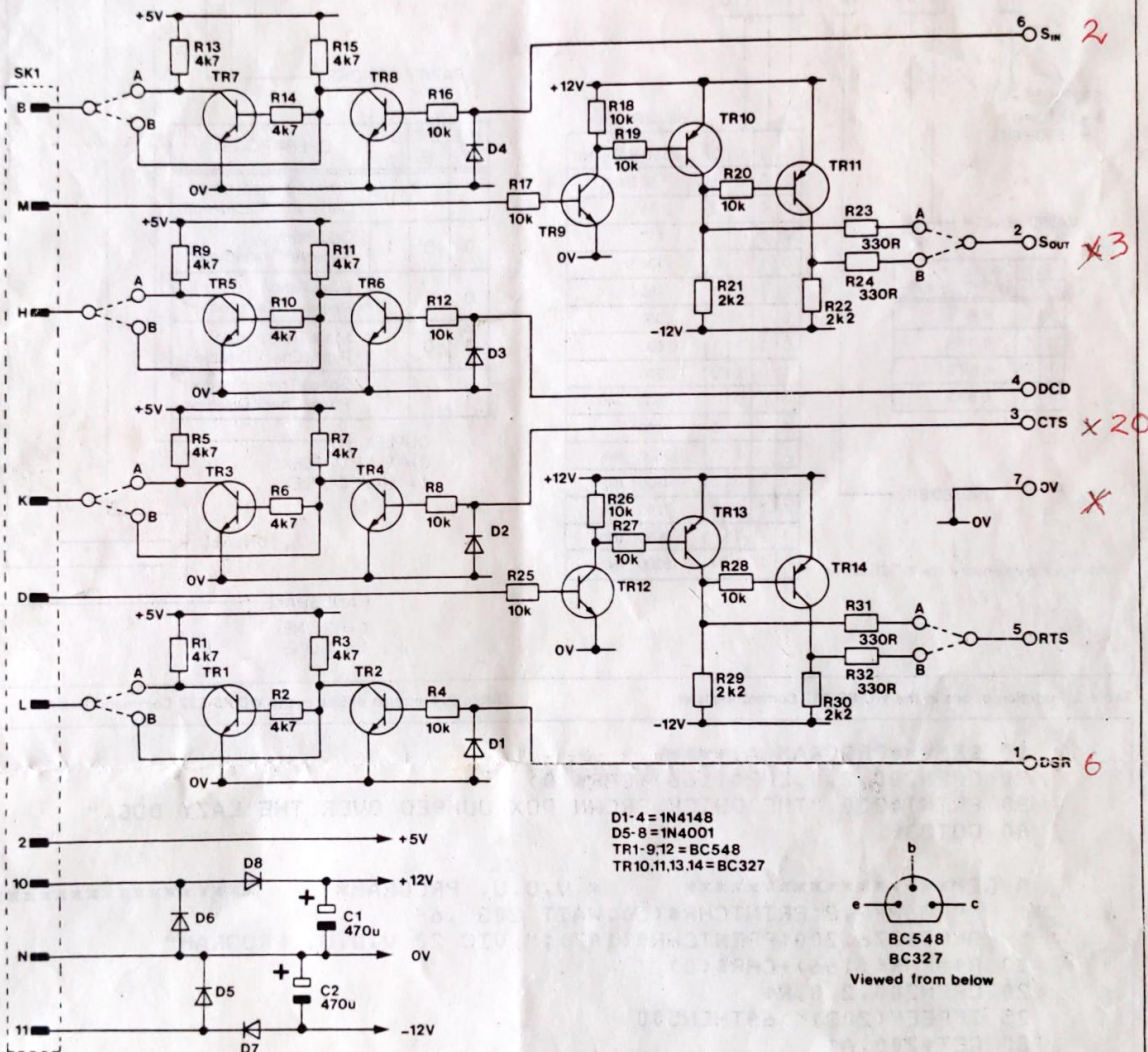


Figure 1. Circuit diagram

A second output stage is also driven from the collector of TR10, and this is identical in operation except that the output is inverted with respect to the input. The output that is used depends on the requirements of the data set.

Receive Mode

The incoming signal first has any negative content removed by D1. It then turns TR2 on and off via R4. The inverting output of the receive circuit is taken from the collector of TR2 which also drives TR1 via R2 to obtain a non-inverting output at the collector of TR1. The choice of which output to wire to the VIC is determined by the polarity of the incoming data. The VIC requires a signal which sits high between 'words' and drops low for data. SIN is on pins B and C on the port, and these are connected together.

June 1983 Maplin Magazine

Construction

Referring to the circuit diagram and parts list, first insert all through pins (see figure 1 for signal polarity pins) and Veropins. Solder them in, not forgetting to solder both sides of the through pins. Insert all other components and solder them in. Attach the edge connector to the board, bending its pins flush with the pads on the board, and solder. Clean the board thoroughly and inspect for dry joints, shorts etc.

Testing

Plug the board into the user port, component side upwards, and switch the computer on. If the computer fails to initialise, switch off and re-check carefully for incorrectly placed components, etc. After the computer initialises, measure test points 1 to 4 with a multimeter. These readings should be approximately as follows:

TP1 — 0V
TP2 — +5V
TP3 — +12V
TP4 — -12V

If all is well, switch off the VIC and remove the interface card. Wire the board to the data set. Reconnect to the VIC, switch both the VIC and data set on, and type in program A. Run the program, and the receiving party should receive the message 'the quick brown fox jumps over the lazy dog' continuously. Also included is a program to make the VIC act like a 'dumb terminal', for use with a modem, to call information and ordering services such as the Maplin on-line computer, Southend (0702) 552941.

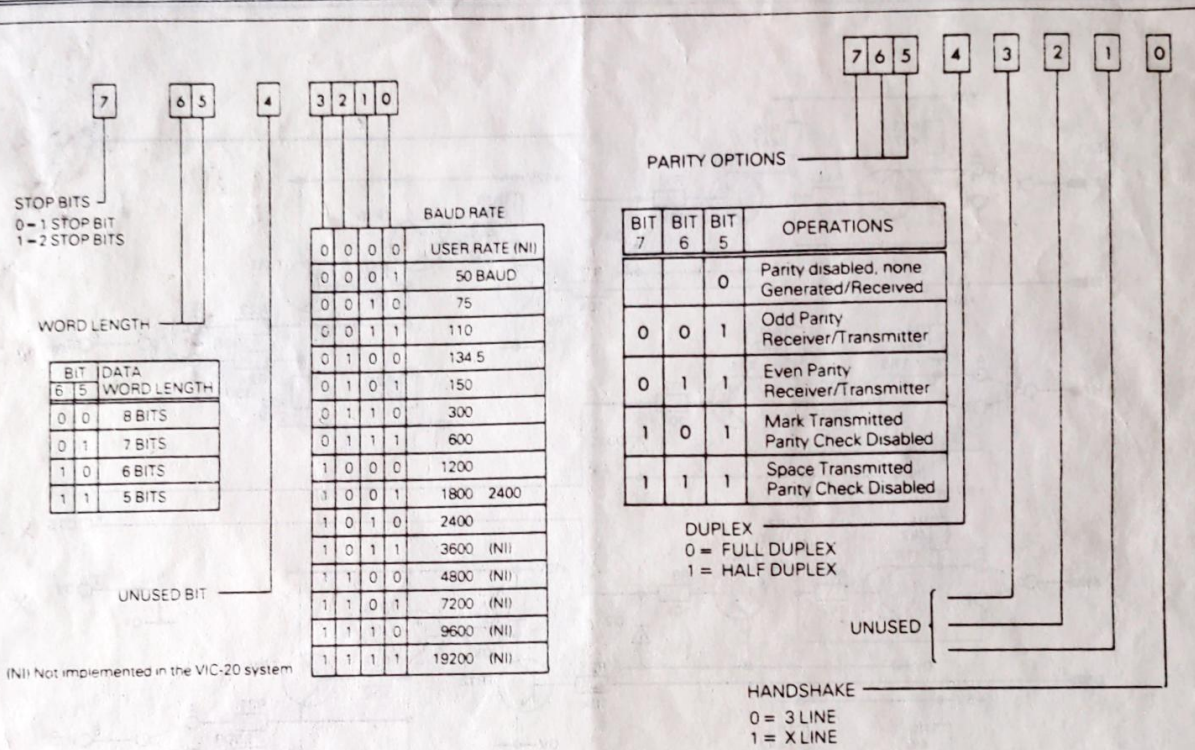


Table 1. Function of bits in the VIC RS-232 Control Register

Table 2. Function of bits in the VIC RS-232 Command Register

```

10 REM***PROGRAM A ***
20 OPEN200,2,0,CHR$(166)+CHR$(0)
30 PRINT#200,"THE QUICK BROWN FOX JUMPED OVER THE LAZY DOG."
40 GOTO30

```

```

0 REM***** * V.D.U. PROGRAM* *****
1 POKE36879,8:PRINTCHR$(5):WAIT 203,64
5 POKE36876,200:PRINTCHR$(147);" VIC 20 V.D.U. PROGRAM"
10 R$=CHR$(166)+CHR$(0)
20 OPEN200,2,0,R$
25 IFPEEK(203)<>64THEN500
30 GET#200,A$
35 IFA$=""THENGOTO25
40 IF ASC(A$)>95THENGOTO25
50 IF A$=CHR$(13)THENPRINTA$;;GOTO25
60 IF ASC(A$)<32THENGOTO25
70 PRINTA$;;GOTO25
500 POKE203,64:GETS$
510 IFS$=CHR$(17)THENS$=CHR$(10)
520 IFS$=CHR$(19)THENS$=CHR$(140)
530 PRINT#200,S$;;POKE203,64:S$="":GOTO30

```

PARTS LIST FOR VIC 20 RS232 INTERFACE

Resistors - All 0.4W 1% Metal Film

R1,2,3,4,5,6,7,9,
10,11,13,14,15 4K7
R4,8,12,16,17,18,
19,20,25,26,27,
28 10K
R21,22,29,30 2k2
R23,24,31,32 330R

Capacitors

C1,2 470uF 16V P.C. Electrolytic

12 off (M47K)

12 off (M10K)

4 off (M2K2)

4 off (M330R)

2 off (FF15R)

Semiconductors

D1-4 inc.

D5-8 inc.

TR1-9 inc., 12

TR10,11,13,14

1N4148

1N40001

BC548

BC327

4 off

4 off

10 off

4 off

(QL808)

(QL730)

(Q873Q)

(Q866W)

Miscellaneous

SK1

P.C. Edgecon 2 x 12 way

Veropin 2141

Track Pin

P.C.B.

(B374R)

(FL21X)

(FL82D)

(GB28F)

1 Pkt

1 Pkt

A complete kit of all parts is available

Order As LK11M (VIC 20/RS232 Interface kit). Price £9.45.

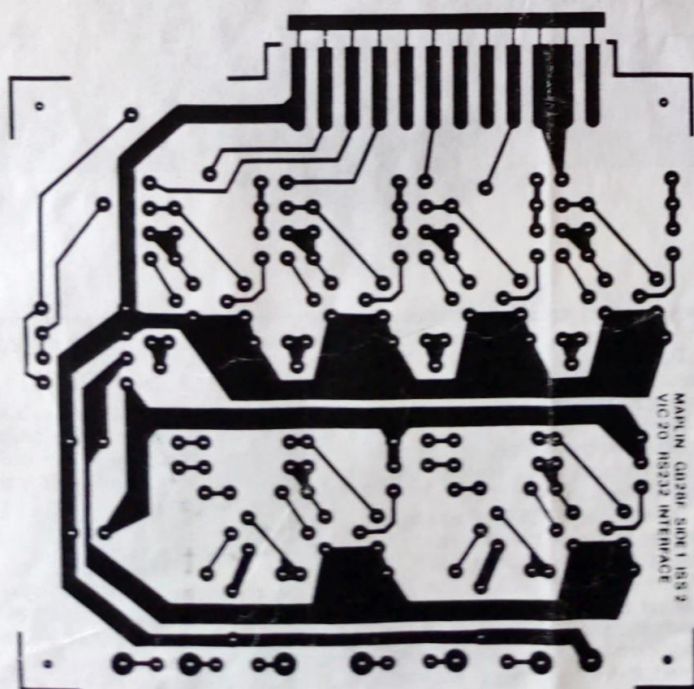
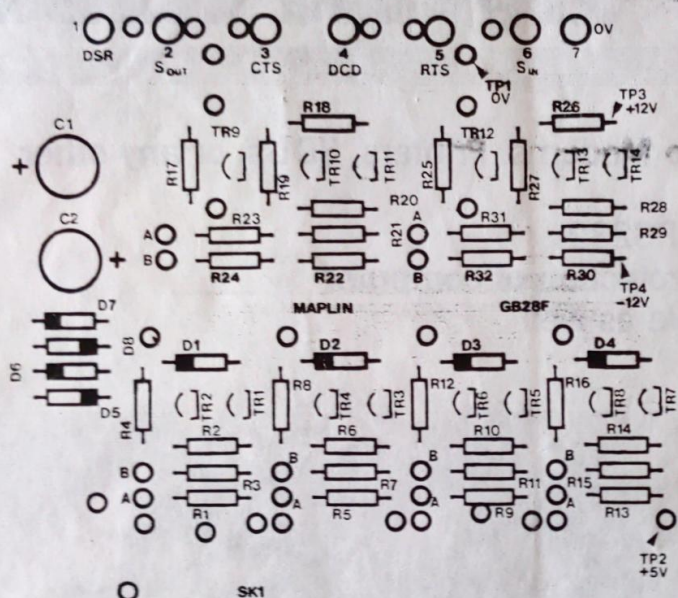


Figure 2: PCB layout

Usage

To use the RS232 serial data port on the VIC the channel must first be opened as a file, specifying Baud rate (speed), number of bits per character, number of stop bits, and odd/even or disabled parity bit. This information is given by two characters after the 'OPEN' command in the form:

OPEN LF,2,0,A\$. Where LF is the logical file number, i.e., any number between 1 and 255 (if LF is greater than 127, then linefeed follows carriage return), and A\$ is two characters sent to control register and command register, the functions of which are explained in tables 1 and 2. So, for example, we can see that to set Baud rate to 300, 7 bits per character, 2 stop bits, and no parity, the OPEN command would be:

OPEN 200,2,0,CHR\$(166)+CHR\$(0). Having opened the RS232 channel, data is sent and received using 'PRINT LF,DATA \$' and 'GET LF,DATA \$'.

Note: To type 'PRINT' do not use the abbreviation 'P'. Instead, use 'P shift, R' followed by logical file number etc. It is possible to list through the RS232 port, to send a program to a friend for instance, by typing 'CMD LF : LIST', where LF=logical file number.

Remember when programming that the VIC allocates two 256 byte buffers (for transmit and receive) in the 506 bytes below RAMTOP, so there is less memory available to BASIC. Also 'DIM' statements or variables should be left until after the 'OPEN' command, as the computer performs an automatic 'CLR' before allocating the buffers.

Bibliography:

VIC Revealed by Nick Hampshire
VIC Programmers Reference Guide
Commodore

Connecting to the Maplin Modem

With reference to figure 5, page 5, issue 5 of Electronics, the following connections should be made:

VIC 20 interface	Maplin Modem
pin 2 (Sout)	to pin 17 (RS232 in)
pin 6 (Sin)	to pin 10 (RS232 out)
pin 7 (Sground)	to pin 21 or 22 (Sground)

Ensure that the link on the Modem is in the RS232 position.