23rd January 1995

Support Group Application Note
Number: 234
Issue: 3.02
Author: DW

Acorn

Peripheral Interfacing via the Serial Port

This document describes the hardware configurations of the serial port over the history of Acorn 32 bit computers, and details some of the software protocols needed to drive them.

Applicable Hardware :	All RISC OS and RISC *i*X computers	Related Application Notes:	None

Support Group
Acorn Computers Limited
Acorn House
Vision Park
Histon, Cambridge
CB4 4AE

Table of Contents

Throughout this Application Note, terms are used which may be unfamiliar to the reader. Signals are defined in Appendix C, and a Glossary has been provided which aims to supply sufficient background information.

Introduction

A serial port allows data to be transferred between two computers, or a computer and peripheral device, as a stream of data bits sent one after the other down a single wire. This is fundamentally different from parallel communications (commonly used with printers), which transfers data many bits at a time over a number of wires. Serial communication can be synchronous or asynchronous.

As the practice of serial communication between devices dates from the very early history of computing, many different standards have sprung up over the years; today, as a result of this, there is no de-facto software communication protocol and no de-facto hardware connector. However, there is a set of standards which most manufacturers tend to agree on; these are the RS232 physical connector specification and the RS423 signal level electrical specification.

The Hardware Layer

The full RS232 standard connector has 25 pins, usually built into a D-type connector. Most modern computers and peripherals do not implement the full RS232 standard, but employ the signal specification from the RS423 standard, again with a D connector. The Acorn serial port is conformant to the RS423 serial port electrical standard.

There are two main classifications of RS232 cable; straight-through and crossover. Broadly speaking, if the cable is linking the computer to a communications device, such as a modem, then a straight-through cable is used - in other words, one where the transmit line is connected to the same-numbered pins at either end of the cable, as is the receive line, etc. A modem is a piece of Data Communication Equipment (DCE); hence the straight-through cable is sometimes referred to as "DTE to DCE."

If the cable is connecting the computer to a peripheral such as a printer, then the transmit line from the computer needs to be connected to the peripheral's receive line and vice versa; the two lines have to be crossed over. A printer, like the computer itself, is a piece of Data Terminal Equipment (the terminology goes back to the concept of a large computer which had a number of user-interface terminals connected to it by serial lines), and so the cross-over cable is sometimes referred to as "DTE to DTE" or a "null modem" cable.

Consult the manufacturer of your peripheral to determine which cable type is required.

Communication via the Serial Port

Data is transferred down a serial line in packets, which comprise a data "word" wrapped up with a "parity" bit and one or more "stop" bits. The waveform looks rather like this:

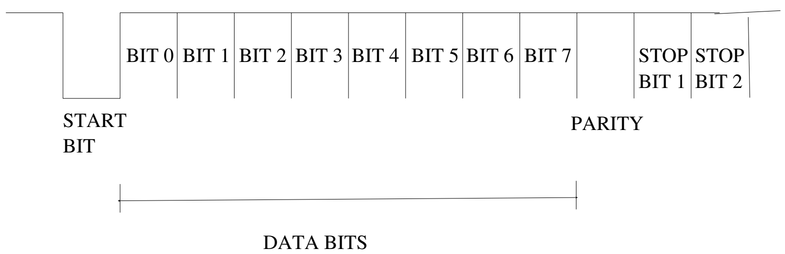


Figure 1: A Typical Serial Data Stream

where the "on" and "off" voltages fall into this range:

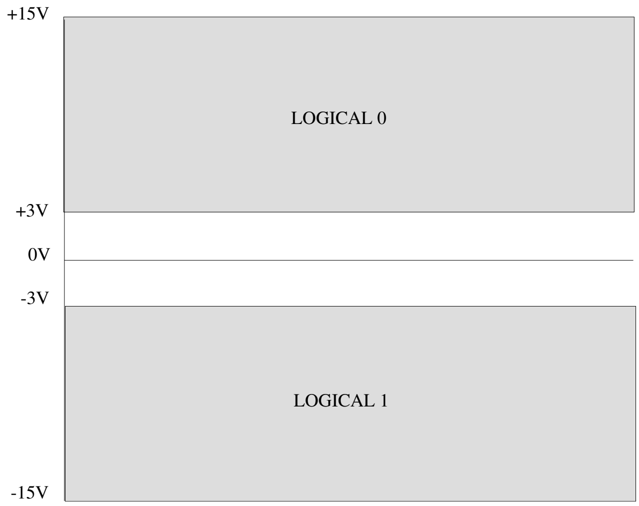


Figure 2: Serial Voltage Ranges

The format of the packet (number of data bits, type of parity etc) and the serial baud rate can be set up using the `*CONFIGURE BAUD n` and `*CONFIGURE DATA n` commands from the command line; the following table details the settings. Consult the documentation supplied with your peripheral to determine which settings are appropriate.

n	Baud rate
0	9600
1	75
2	150
3	300
4	1200
5	2400
6	4800
7	9600
8	19200
9	50
10	110
11	134.5
12	600
13	1800
14	3600
15	7200
16	38400
17	57600
18	115200

n	Word length	Parity	Stop bits
0	7	even	2
1	7	odd	2
2	7	even	1
3	7	odd	1
4	8	none	2
5	8	none	1
6	8	even	1
7	8	odd	1

Figure 3: Baud Rate and Word Format Configuration Settings

eg `*CONFIGURE BAUD 7` would make the system default to 9600 baud communications.

Note that baud rates above 19200 are only available on the Risc PC; all other baud rates listed above are available on all systems fitted with RISC OS 3.1 or later.
For high baud rates, the transmit and receive rates must be set to the same value; at low baud rates (1200 and below), a system timer can be used for the "receive" rate, hence a different receive rate can be set using `*FX7`, eg `*FX7,3` would set a receive rate of 300 baud.

A Note on Baud Rates

A common misconception in serial communications concerns the definition of "baud" as the number of bits transmitted per second (bps), particularly when used in reference to modems.

The baud rate is the number of changes in signal state per second. The Public Switched Telephone Network (PSTN) bandwidth will not support a "true" baud rate greater than 600 under any circumstances; however, each change of signal state may represent a 1, 2, 4, 8 or 16 bit sequence, depending on the coding method in use. Thus the term "baud" is incorrectly used if the transmission speed is classed as greater than 600 baud; however, at transmission rates of 300 and 600 bits per second including one start, one parity and two stop bits, a 1:1 mapping of bits to baud is achieved.

The data rate, however, is (bits per second)*(word length)/(total packet size in bits), which usually amounts to 300*8/11 or 600*8/11 "useful" bps.

Fax transmission rates of 9600 bps require Quadrature Amplitude Modulation, in which each combination of phase angle and amplitude represents one of 16 4-bit patterns; the baud rate is therefore 9600/16=600 baud, which is still just within the PSTN bandwidth. At higher transmission speeds, other coding methods allow each line state to represent an 8, 16 or 32 bit pattern.

In short, the "old" definition of the baud (ie as the reciprocal of the duration of the shortest signalling element) was meaningful for Morse code, in which a dot was the shortest-duration element and a dash had a duration three times as long. It no longer has any real meaning in digital transmission, as all line states last for an equal time.

Communication Protocols and Flow Control

Owing to the nature of serial communications, it is necessary to have a signalling system by which the peripheral can inform the host that it is present, switched on, and ready to receive data or has data to transmit, and vice versa. The last two items in this list comprise flow control.

Unlike IBM PC compatibles, Acorn machines default to using the DSR line to flag readiness to transmit, and require the presence of the DCD signal. IBM PC compatibles use CTS instead of DSR, and do not necessarily use DCD. On machines fitted with the 82710 or 82711 serial controllers (these machines being the A3010, A3020, A4000, A5000, A4 and Risc PC), it is possible to change which signals are used for flagging in software, using the `SWI"OS_SerialOp",0` command. Thus an IBM compatible cable may be used; the option to reprogram the serial port in this manner is currently provided by some serial communications software. If you have a suitable machine and prefer to use an over-the-counter IBM standard cable rather than resort to making your own, first check with the supplier of your communications software that there is an option to reprogram the serial controller.

If you have an older Acorn machine (Archimedes 300 series, 440, 400/1 series, 540, R140 or R200 series), you will need to have a cable wired to a specification based on the figures in Appendix A of this document.

The need for flow control and "handshaking," in which the receiving device verifies the parity bits included in words and can request the re-sending of a corrupted word, has led to the development of a number of serial communications protocol standards. These vary in their ease of implementation, speed and robustness of transfer, and a number of them are described below.

XON / XOFF

This is the earliest software flow control system which is still in use, and makes use of the flow control characters &11 (XON) and &13 (XOFF). There is no error checking implemented as standard. XON /

XOFF is incompatible with SLIP and PPP, as they require the full 8 bits of the data word to send Internet frame data; ie under Internet Protocol, &11 and &13 cannot be reserved for flow control.

XModem
This is probably the most widely available communications protocol, and is supported by a large number of communications packages on many types of host. XModem transfers files in blocks of 128 bytes. Each block has a checksum, or CRC (Cyclic Redundancy Checksum) added to the end, which is calculated by summing the values of the bytes in the packet and taking a pair of bits from the sum. From this checksum, the receiving system can determine whether the packet was corrupted during transmission, and if so, it can ask the sending system to retransmit that block. Although fairly slow in transferring data, XModem will produce reliable transfers.

YModem
Based on the same protocol set as XModem, YModem uses 1024 byte packets wherever possible, and a different system for packet integrity checking. Although YModem can fall back on smaller packets where applicable, there is no backward compatibility with XModem's checksum system.

Kermit
This is another packet-oriented serial communication protocol. Developed at the University of Columbia, the protocol standard is Public Domain, and hence Kermit ranks with XModem for widespread use. Although fairly intensive on encoding and checksumming, and hence fairly slow, Kermit connections are very robust.

SLIP
SLIP provides a point-to-point connection between two devices for the transmission of Internet datagrams; the devices can be either two computers, or a computer and an Internet router. SLIP modifies a standard Internet datagram by appending a special SLIP END character to it, which allows datagrams to be distinguished as separate. Normal parameters for an asynchronous lines apply to SLIP; SLIP requires a port configuration of 8 data bits, no parity, and EIA or hardware flow control. SLIP does not provide any protection against line errors and data corruption, being reliant on other high-layer protocols for this. Over a particularly error-prone dialup link, therefore, SLIP on its own would not be satisfactory. A SLIP system also needs to have its IP address configuration set every time the SLIP is loaded and configured, as it cannot determine network addresses dynamically.

PPP
The Point-to-Point Protocol has a number of advantages over SLIP; it is designed to operate both over asynchronous connections and bit-oriented synchronous systems, can configure connections to a remote network dynamically, and test that the link is usable. Full information on PPP can be obtained from Internet RFC 1171 (see Glossary), and RFC 1220 describes how PPP can be used with remote bridging.

Communication with BBC Model B and Master Series Computers

It is possible to transfer data between a 32-bit Acorn computer and an 8 bit BBC Model B / Master series computer via the Serial Port; the wiring diagram for the appropriate cable is presented as Figure 9 in Appendix A.

If no suitable communications packages are available, it is possible to transfer data without using one of the recognised communications protocols; this is not recommended except in exceptional circumstances and with a very short cable.

On the transmitting system (assumed to be the Model B / Master), issue:

```
*FX8,4
*FX3,1
```

The first call sets the transmission rate to 1200 baud, and the second selects the serial port as the output device.

On the receiving machine, (assumed to be a 32 bit Acorn machine), issue:

```
*FX156,20
*FX7,4
*FX2,1
```

The first command configures the parity and word size appropriately: note that the Model B defaults to one stop bit per word, whereas the 32 bit range defaults to two stop bits. The second sets the receive rate to 1200 baud, and the third causes standard input to be via the serial port.

Performing a **`LIST`** operation on a BASIC program stored in the BBC's RAM, or a **`*TYPE`** on a plain text file stored on its disc, will cause the program or text to be loaded into whichever RISC OS application has the caret on the receiving machine.

Communication is terminated by issuing

```
*FX3,0
*FX2,0
```

on both machines; it is suggested that the receiving machine terminates communication first.

Telecommunication Standards and Data Compression

The CCITT (European Telecommunication Standards body) has defined a number of standards for modem-to-modem communications over the PSTN network, and translation from these standards codes to communication capabilities is often a point of confusion. The following list covers most of the standards currently available.

V.21 and **Bell 103**
Specifies 300 bps 2-wire full duplex communications using FSK (Frequency Shift Keying) modulation schemes. AT&T created the Bell 103 specification during the days of telephone system monopoly in the USA. The two standards, V.21 and Bell 103, differ slightly, but all of today's modems support both V.21 and Bell 103.

V.22 and **Bell 212A**
CCITT standard for 1200 bps full duplex modems. Specifies 1200 bps 2-wire full duplex communications using QPSK (Quadrature Phase Shift Keying) modulation at 600 baud. Again, these two standards differ slightly.

V.22 bis
CCITT standard for 2400 bps full duplex modems. Specifies 2400 bps 2-wire full duplex communications using a QAM (Quadrature Amplitude Modulation) scheme at 600 baud.

V.23
Specifies an asymmetrical communication scheme which implements 1200 bps data transmission in one direction, and 75 baud data transmission in a back channel. This FKS-based standard is popularly used in Europe for applications which require high data rates in only one direction, eg CET 1 / 2 / 3 Teletext.

V.24
Specifies a serial interface; analogous to RS232.

V.25 and **V.25 bis**
CCITT standard for auto-dial commands for modems and other auto-dial devices. Not commonly used, because of the popularity of the Hayes command set.

V.26, **V.26 bis** and **V.26 ter**
Specifies half and full duplex leased-line communications at 1200 and 2400 bps. The specifications employ QPSK modulation at 1200 baud. V.26 ter was the first modem standard to specify echo cancellation.

V.27, **V.27 bis** and **V.27 ter**
Specifies 4800 bps communications requiring 2 wires for half duplex and 4 wires for full duplex operation. The standards specify QAM modulation at 1600 baud. The Group 3 Fax standard references V.27 ter as the base requirement for 2-wire half duplex fax communications.

V.29
CCITT standard for 9600 bps half duplex modems. Specifies 9600 bps communications requiring 2 wires for half duplex operation, and 4 wires for full duplex. The standard specifies QAM modulation at 2400 baud. Group 3 Fax standard T.4 references V.29 as an option for fax transmissions faster than 4800 bps V.29 ter. A high percentage of fax transmissions rely on V.29, but virtually all fall back on V.29 ter.

V.32
CCITT standard for 9600 bps full duplex modems. Specifies 2-wire full duplex 9600 bps communications using QAM modulation at 2400 baud and echo cancellation. V.32 modems offer an upgrade path from V.22 bis for asynchronous dial-up modem applications.
V.32 AUTOMODE was recently published as an annex to V.32, and defines an automatic fall-back capability which does not support 9600 bps communications.

V.33
Specifies 4-wire full duplex and 2-wire half duplex communications at 14400 bps. V.33 employs TCM (Trellis-Coded Modulation). The Group 3 Fax study group has modified T.4 to include this.

V.34
The current state of the art in serial communications over PSTN, V.34 specifies full duplex 28800 bps.

V.42
Specifies error correction techniques which can be implemented in modems independently of transmission speed and modulation system. The recommendation includes LAPM (Link Access Procedures for Modems) and MNP (Microcom Networking Protocol) 2 to 4 error correction. The CCITT standard (which is the V.42 specification) utilises MNP4 and LAPM. When a modem makes a connection, it tries to use LAPM; if the receiving modem does not support LAPM the connecting modem tries MNP4, and if the receiving modem does not support MNP4 the system falls back on asynchronous non-error-correcting communication. LAPM and MNP4 protocol querying and detection is entirely transparent to the user.

V.42 bis and **MNP5**
This specifies compression algorithms which can be implemented in modems independently of transmission speed and modulation system. V.42 bis provides a 4:1 compression ratio, using the Lempel-Ziv algorithm (as used in !Squash under RISC OS 3). MNP5 uses a combination of dynamic Huffman and run-length encoding. MNP5 is not a standard as defined by a specific organisation, but has become a standard in its own right for 2:1 data compression.

The Programmer's Interface to the Serial Port

There is a "raw" device which corresponds to the serial port (this is `serial:`); however, writing to the serial port by directly using this device is deprecated. The approved programmer interface is via the `SWI"OS_SerialOp"` call (SWI &57), which is described starting on page 2-459 of the RISC OS 3 Programmer's Reference Manual. Of particular interest is the provision for implementation of flow control; the serial status word, accessible via `SWI"OS_SerialOp",0` allows XON/XOFF with CTS handshaking, use of DCD or use of DSR to implement signalling of intent to transmit / receive data.

Unlike IBM compatibles fitted with the 82710 / 82711, Acorn computers default to the behaviour associated with the 6551 serial controller fitted in the original Archimedes range. In order to allow an IBM type lead to be used, bit 1 (ignore DCD) of the serial port status word must be set using the appropriate `SYS"OS_SerialOp",0` mask. However, to produce a robust program capable of coping with a noisy connection, it is necessary to periodically reset the status word so that DCD can be checked. The type of cable which is connected may either be selected by a user menu option, or it is acceptable to check the state of DCD on program startup.

Full details of this status word are listed on Page 2-462 of the RISC OS 3 Programmer's Reference Manual.

Connecting Printers to the Serial Port

!Printers is capable of sending data to a suitable printer via a serial link; the printer usually has to be specially configured to receive data in this manner, and it is suggested that you refer to your printer manual for information on how the printer's DIP switches must be set and which formats of serial word it will accept. !Printers must itself be configured according to the information supplied in the User Guide, and some suitable cable wiring diagrams are shown in Appendix A. Printers tend to be configured to the DTE standard, however you should check this with your printer supplier. It is also worth noting that a cable for this purpose usually has fewer wires to be soldered, as data is generally only sent in one direction; from computer to printer.

Troubleshooting

First of all, make sure that your serial lead and your communications package are compatible; if you are using an A5000 or later with an IBM type lead, make sure that your communications package supports reprogramming of the serial controller.

On some modems, the modem takes control of the RI line, and this may cause an Acorn computer to hang up. The solution to this is to leave Pin 9 (RI) of the serial port disconnected if such a modem is in use.

Sometimes, when handshaking signals become corrupted (this only occasionally happens over long, unshielded cables in areas where a lot of electrical equipment is operating), communications will "hang up" in a deadlocked state. Rather than reset the computer or the peripheral, it is often possible to "wake up" the peripheral by sending a Break Level. Not to be confused with an operation involving the Break key on the keyboard, a Break Level sends a 0V pulse to the RxD pin on the peripheral. A Break Level can be sent, if your communications package does not already implement it, by entering BASIC and issuing

`SYS"OS_SerialOp",2,`<*duration in centiseconds*>

eg `SYS"OS_SerialOp",2,20` would send a "short break."

A "short break" is generally about 0.2 seconds long, and a "long break" can be as much as 1.5 seconds. It is suggested that a "long break" only be tried as a last resort before a device reset. Any characters being sent when the break is issued may be garbled; however, if the break succeeds in waking the peripheral, a robust communication protocol would simply ask for the last data packet to be re-transmitted.

On computers prior to the Risc PC running RISC OS 3.10, communications may be unreliable above 9600 baud, depending on the length and impedance of cable connected. The unreliability appears in the form of occasional "missed words" when receiving data.

A pair of soft-loadable patch modules, SerialDev and SerialUtil, are available via Acorn dealers, Acorn Education Centres and bulletin boards, and may be downloaded via ftp over Internet from ftp.acorn.co.uk. SerialUtil is applicable for use under RISC OS 3.1 in conjunction with communications packages written under RISC OS 2, and provides enhanced arbitration of the claiming of the serial interrupt vector. SerialDev is a modified version of the serial port device driver which forms part of RISC OS 3.10, and improves the interrupt latency of the serial port to give improved communications reliability at high baud rates (eg 9600 baud). This improved driver was included as part of RISC OS 3.11.

If you think that you may have a hardware fault with your serial port, you may find it useful to make up a loopback testing plug; using this plug, in conjunction with serial loopback test software, should pass data straight from the "transmit" pins to the "receive" pins.

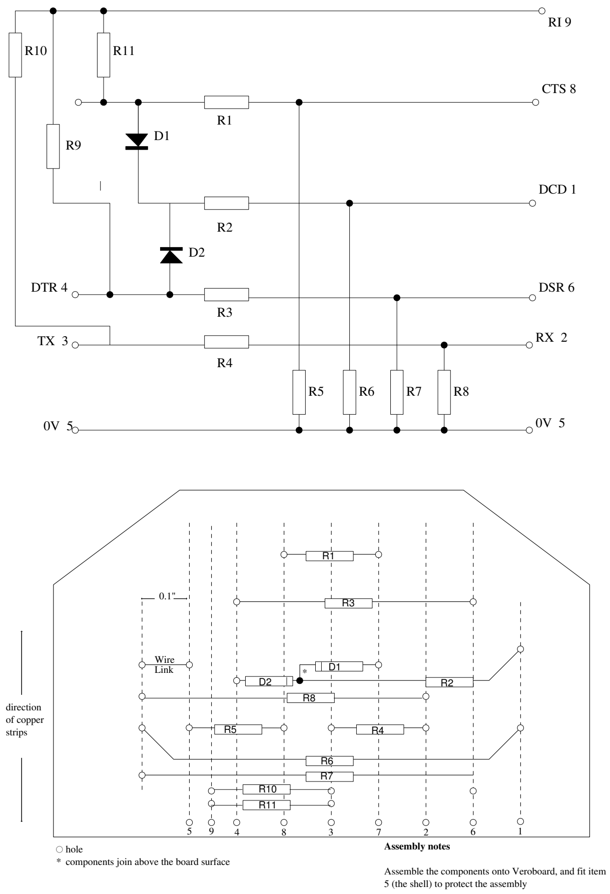


Figure 4: Generic Loopback Test Plug

Please note that there are different versions of this Loopback connector, modified to work correctly with different machines; hence when building a plug from the parts lists below, use the generic board layout above and simply omit any components marked as N/F.

Item	Part no.	Description	Qty
1	0276,081	CIRCUIT & ASSEMBLY DRAWING	1*
3	0800,288	CONR 9W SCKT 'D' ST MS SB	1
5	0800,991	CONR 9W SHELL 'D' + SCREWS	1
R1	0502,331	RES 330R C/MF 5% 0W25	1
R2	0502,332	RES 3K3 C/MF 5% 0W25	1
R3	0502,331	RES 330R C/MF 5% 0W25	1
R4	0502,331	RES 330R C/MF 5% 0W25	1
R5	0502,181	RES 180R C/MF 5% 0W25	1
R6	0502,562	RES 5K6 C/MF 5% 0W25	1
R7	0502,181	RES 180R C/MF 5% 0W25	1
R8	0502,181	RES 180R C/MF 5% 0W25	1
R9			N/F
R10			N/F
R11			N/F
D1	0790,048	DIODE BAT85 SBL	1
D1	0790,048	DIODE BAT85 SBL	1

* Per batch

Figure 5: Archimedes 300 series and 440 serial loopback parts list

Item	Part no.	Description	Qty
1	0276,081	CIRCUIT & ASSEMBLY DRAWING	1*
3	0800,288	CONR 9W SCKT 'D' ST MS SB	1
5	0800,991	CONR 9W SHELL 'D' + SCREWS	1
R1	0502,271	RES 270R C/MF 5% 0W25	1
R2	0502,272	RES 2K7 C/MF 5% 0W25	1
R3	0502,271	RES 270R C/MF 5% 0W25	1
R4	0502,181	RES 180R C/MF 5% 0W25	1
R5	0502,181	RES 180R C/MF 5% 0W25	1
R6	0502,562	RES 5K6 C/MF 5% 0W25	1
R7	0502,181	RES 180R C/MF 5% 0W25	1
R8	0502,181	RES 180R C/MF 5% 0W25	1
R9			N/F
R10			N/F
R11			N/F
D1	0790,048	DIODE BAT85 SBL	1
D1	0790,048	DIODE BAT85 SBL	1

* Per batch

Figure 6: Archimedes 400/1 serial loopback parts list

Item	Part no.	Description	Qty
1	0276,081	CIRCUIT & ASSEMBLY DRAWING	1*
3	0800,288	CONR 9W SCKT 'D' ST MS SB	1
5	0800,991	CONR 9W SHELL 'D' + SCREWS	1
R1	0502,122	RES 1K2 C/MF 5% 0W25	1
R2	0502,122	RES 1K2 C/MF 5% 0W25	1
R3	0502,122	RES 1K2 C/MF 5% 0W25	1
R4	0502,122	RES 1K2 C/MF 5% 0W25	1
R5	0502,472	RES 4K7 C/MF 5% 0W25	1
R6			N/F
R7	0502,472	RES 4K7 C/MF 5% 0W25	1
R8	0502,472	RES 4K7 C/MF 5% 0W25	1
R9			N/F
R10			N/F
R11			N/F
D1	0790,048	DIODE BAT85 SBL	1
D1	0790,048	DIODE BAT85 SBL	1

* Per batch

Figure 7: A3000, A30n0, A4000, A5000 serial loopback parts list

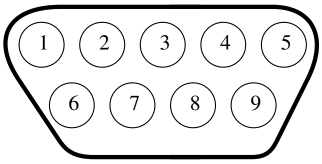
Item	Part no.	Description	Qty
1	0276,081	CIRCUIT & ASSEMBLY DRAWING	1*
3	0800,288	CONR 9W SCKT 'D' ST MS SB	1
5	0800,991	CONR 9W SHELL 'D' + SCREWS	1
R1	0502,271	RES 270R C/MF 5% 0W25	1
R2	0502,272	RES 2K7 C/MF 5% 0W25	1
R3	0502,271	RES 270R C/MF 5% 0W25	1
R4	0502,222	RES 2K2 C/MF 5% 0W25	1
R5			N/F
R6			N/F
R7			N/F
R8	0502,222	RES 2K2 C/MF 5% 0W25	1
R9	0502,222	RES 2K2 C/MF 5% 0W25	1
R10	0502,183	RES 18K C/MF 5% 0W25	1
R11	0502,222	RES 2K2 C/MF 5% 0W25	1
D1	0790,048	DIODE BAT85 SBL	1
D1	0790,048	DIODE BAT85 SBL	1

* Per batch

Figure 8: Risc PC 600 serial loopback parts list

Appendix A: Cable Configurations

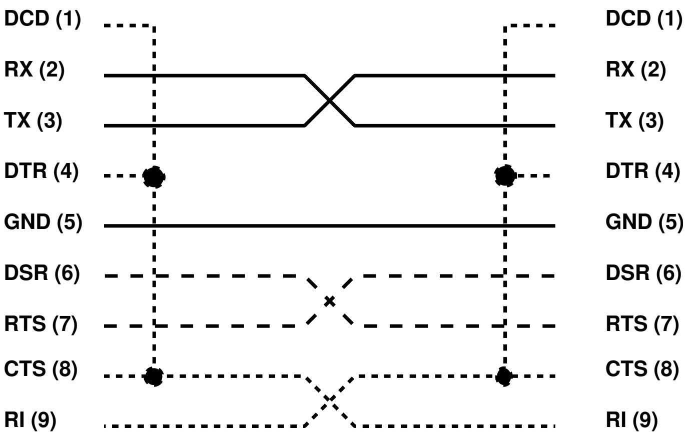
The pinout of the Serial Port is as follows:



Pin	Signal
1	DCD
2	RxD
3	TxD
4	DTR
5	0V
6	DSR
7	RTS
8	CTS
9	RI

Figure 9: Acorn Serial Port Pinout

9-way serial connectors on other systems generally have the same pinout; however, it is suggested that you consult the appropriate documentation for the other system. Some suitable cable configurations for Acorn machines are shown below; if you make your own cables, ensure that the cable you use is adequately screened.



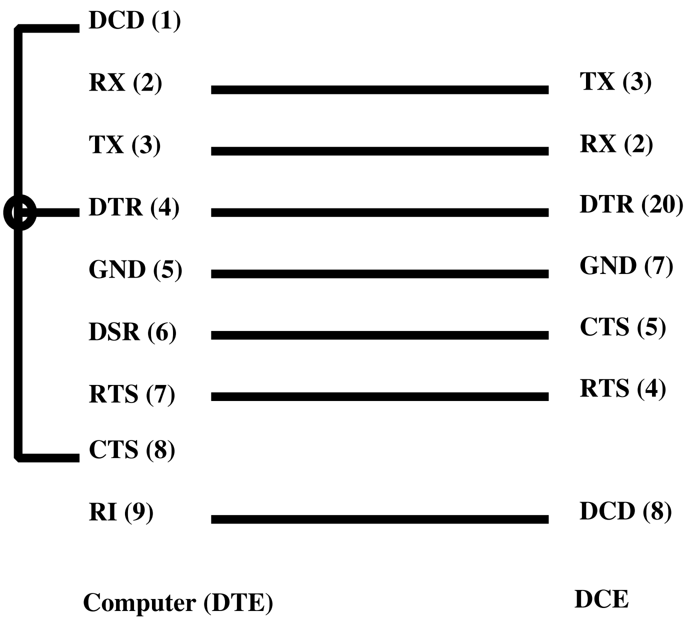


Figure 10:
DTE to DTE (Acorn computer to Acorn computer)

Figure 11:
DTE to DCE (Acorn computer to 25-pin Modem)

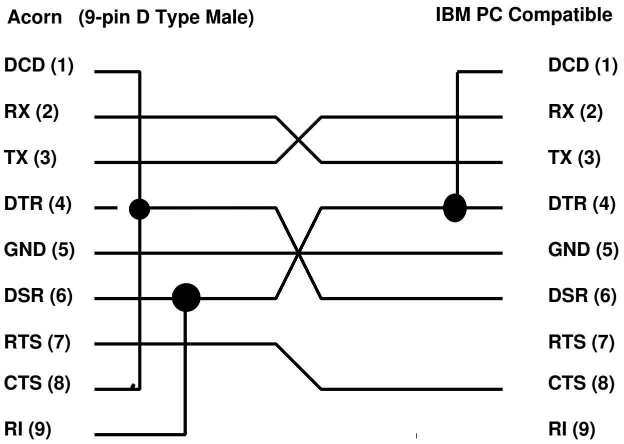


Figure 12:
Acorn Computer to IBM PC compatible

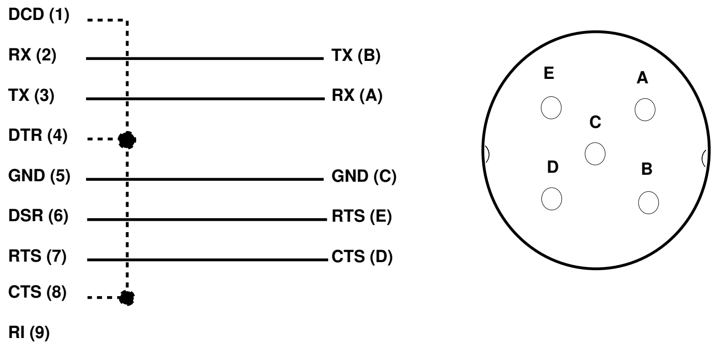


Figure 13:
Acorn 32 bit computer to BBC Model B / Master Series

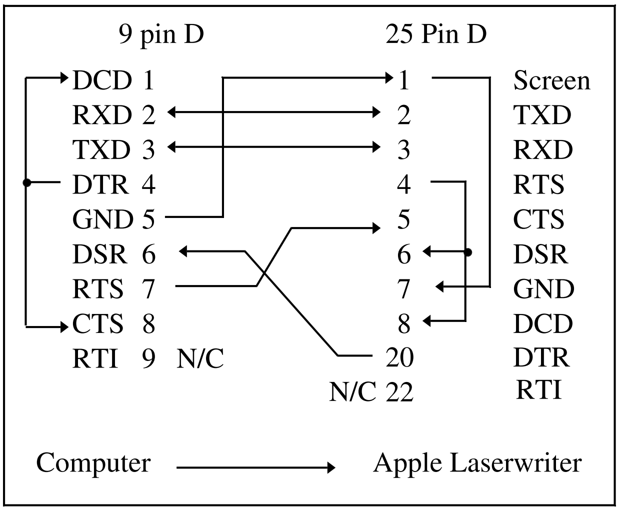


Figure 14:
Acorn Computer to Apple LaserWriter

Risc PCs and A5000s use an 82710 or 82711 peripheral controller, which provides the driver hardware for the serial port; as this IC is used in many IBM PC compatibles, it is possible to software-configure the computer to use the serial port in exactly the same way as an IBM PC compatible does. However, the system is pre-configured to behave as an Acorn machine, so Archimedes cables which require DCD and have flow control via DSR will work correctly without any reconfiguration.

Some software communications packages have an option to reconfigure the serial port on A5000s and Risc PC so that IBM-type cables can be used.

Appendix B:Useful Contacts

For SLIP modules, which can be used in conjunction with the Acorn TCP / IP suite on all Acorn 32 bit machines, including the Risc PC:

Gnome Computers Limited
25a Huntingdon Street
St Neots
Cambridgeshire
PE19 1BG

Tel: 0480 406164
email: chris@gnome.co.uk

For copies of the RS232, RS423 and CCITT standards documents:

BSI Standards
Linford Wood
Milton Keynes
MK14 6LE

Tel: 0908 221166

For Internet RFCs:
Anonymous FTP to

doc.ic.ac.uk
or
ota.ox.ac.uk

These two sites are given as example only, and are convenient owing to their UK location; RFCs are available on many more Internet sites.

Appendix C: RS232 Signals

For completeness, and for use when interfacing Acorn systems to hardware which uses 25 way serial connectors, there follows a list of the signals present in the full RS232 serial port specification.

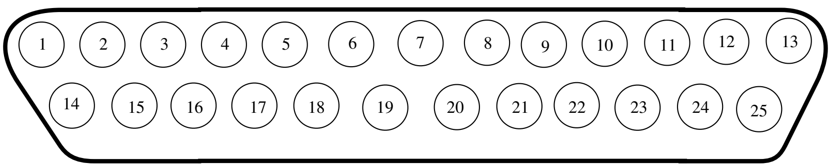


Figure 15: 25 Way Serial Pinout

Pin 1. **Prot** (Protective Ground)
This will usually form a connection between any metal screening on the cable and the metal chassis of the computer and peripheral.

Pin 2. **TXD** (Transmitted Data)
This is the line all data is transmitted on. Transmission will only occur if line 5, CTS, is active.

Pin 3. **RXD** (Received Data)
This is the line all data is received on.

Pin 4. **RTS** (Request To Send)
This is the line which indicates that an RS232 device is ready to transmit data. In order to find out whether data is expected, a receiving device tests this line.

Pin 5. **CTS** (Clear To Send)
The state of this line is used to indicate that a device is ready to receive data transmitted to it by another RS232 device.It is used to inhibit data transfer until the receiving device is ready to accept it.

Pin 6. **DSR** (Data Set Ready)
This line is used to indicate that a connected RS232 receiving device is switched on.

Pin 7. **GND** (Signal Ground)
This provides a common reference for both input and output signals on both systems.

Pin 8. **DCD** (Data Carrier Detect)
This line is used for hardware flow control on Acorn systems, instead of CTS.

Pin 9. Not connected

Pin 10. Not connected

Pin 11. **STF** (Select XMT Frequency)

Pin 12. **dcd** (Secondary DCD)

Pin 13. **cts** (Secondary CTS)

Pin 14. **xmt** (Secondary XMT)

Pin 15. **Xclk** (Transmit Clock)

Pin 16. **rcv** (Secondary RCV)

Pin 17. **Rclk** (Receive Clock)

Pin 18. Not connected

Pin 19. **rts** (Secondary RTS)

Pin 20. **DTR** (Data Terminal Ready)
This line is used to indicate whether the RS232 transmitting device is switched on.

Pin 21. **SQ1** (Signal Quality)

Pin 22. **RI** (Ring Indicator)

Pin 23. **DRS** (Data Rate Select)

Pin 24. (External Transmit Clock)

Pin 25. **BY** (Busy; Standby)

Glossary

Asynchronous communication: Communication over a serial line where one device has to "sit and listen" while the other sends data; the two devices are not able to send data simultaneously.

Baud: See "A Note on Baud Rates"

DCE: Data Communications Equipment. A piece of equipment which obeys the DCE specification as laid down in the RS232 standard. Usually a modem, or other device which passes information through it to other devices.

DTE: Data Terminal Equipment. A piece of hardware such as a computer or a printer, which obeys the DTE specification as laid down in the RS232 standard, and which generally does not relay data on electronically.

FTP: Internet File Transfer Protocol. Used to download data from an FTP server (usually a UNIX or VMS host), via an FTP client such as the one supplied as part of the Acorn TCP/IP suite.

Hayes Command Set: This is a command protocol used for software control of many of today's high-specification modems. A Hayes command sequence starts with the letters `AT` (short for ATtention), and can be followed by a number of parameters. A list of Hayes commands is usually supplied with a modem supporting the command set.

Modem: Short for MODulator / DEModulator. A piece of equipment which converts serial data into audio-ftrequency tones suitable for sending over the Public Switched Telephone Network (PSTN), and which can take audio data back in from a remote system and convert it to serial data.

Packet: A standard "chunk" of data sent over a serial link. A basic packet comprises a start bit, a single data word, a parity bit and either one or two stop bits. Acorn systems default to two stop bits. Communication protocols generally use larger packets, containing many words per packet.

Parity: A bit added onto the end of a word in a serial data packet, which allows a simple measure of detecting whether a word was received without being corrupted. Parity comes in two flavours, even and odd; this determines the type of checksum calculation carried out.

RFC: A "Request For Comment" document. These are usually available over the Internet, and are the standards documents which define Internet protocols. Two sites which carry many of the RFCs as plain text documents are listed in the Useful Contacts section.

Synchronous Communication: Communication over a serial line where both parties in the transaction can be sending and receiving data simultaneously.

Word: A sequence of bits in a basic serial packet which contains the "useful" data being transferred. In serial communication, a word is normally seven or eight bits long.