aculab

DATA BUFFER USERS MANUAL

==========

A C U L A B L T D

DB9 Data Buffer User Manual

==========

Date : 17-08-87

Issue : 1.01

Note :-

The product to which this manual refers may be subject to revision without notice.

TABLE OF CONTENTS

1. Introduction.....1

1.1. The DB9 data buffer.....1
1.2. Hardware description.....1
1.3. Software description.....1
1.4. Input/Output tradeoff.....2

2. The Front Panel Controls.....3

2.1. The CLEAR switch.....3
2.2. The HOLD switch.....3
2.3. The COPY switch.....3

3. The Front Panel Indicators.....4

3.1. The POWER indicator.....4
3.2. The DATA indicator.....4

4. Connecting the data buffer.....5

4.1. Basic arrangement.....5
4.2. Input and Output connections.....5

5. The Parallel Input.....6

5.1. Parallel input connector.....6
5.2. Parallel input connector listing.....6
5.3. Parallel input protocol.....7

6. The Parallel Output.....8

6.1. Parallel output connector.....8
6.2. Parallel output connector listing.....8
6.3. Parallel output protocol.....9

7. The Setup Switches.....10

7.1. Setup switch settings.....10
7.2. Setup switch listing.....10

8. The Serial Ports.....11

8.1. Serial port operation.....11
8.2. Serial port handshaking.....11
8.3. The backward channel.....12
8.4. Notes regarding backward channel operation.....12

9. Serial Input.....14

9.1. The serial input connector.....14
9.2. Serial input connector listing.....14
9.3. Serial input with DTR.....15
9.4. DTR Protocol listing.....16
9.5. Serial input with XON/XOFF.....17
9.6. XON/XOFF protocol listing.....18

10. Serial Output.....19

10.1. The serial output connector.....19
10.2. Serial output connector listing.....19
10.3. Serial output with DTR.....20
10.4. DTR protocol listing.....21
10.5. Serial output with XON/XOFF.....22
10.6. XON/XOFF protocol listing.....23

1. Introduction.

1.1. The DB9 data buffer.

The Aculab DB9 data buffers are a range of intelligent microprocessor controlled devices reflecting a third-generation approach to design with respect to features, facilities and ease of installation and use.

Although designed principally for printer buffering, data buffers find many other uses including protocol conversions, data aquisition and machine monitoring.

As a printer buffer it accepts input data as it becomes available (up to the point that the buffer becomes full), and simultaneously provides data at the rate that the printer demands (usually much more slowly).

This frees the host computer from the requirement to wait for the printing to be completed before it continues processing and thus usually results in a substantial increase in system throughput.

1.2. Hardware description.

The buffer consists of a mains-powered unit containing a single-chip microcomputer.

Buffer space is provided by eight 256k dynamic RAM chips providing 256k bytes of storage.

The processor reads two separate 8-way switches to determine the power-on configuration.

Operator control and information is provided by three front-panel switches and two LEDs.

1.3. Software description.

Upon initialisation due to a power-up or front-panel reset (the CLEAR button), the system software clears the buffer, reads the setup switches and determines the input and output ports to be used, the baud rates and word lengths etc..

The processor then puts the inputs and outputs into a 'ready' condition and implements a rotating FIFO (first-in first-out) buffer within its available memory space.

In the event that the buffer comes within a certain number of characters of being full (according to the protocol used on the input to allow for the inevitable 'overrun' that will occur on some serial protocols), it sets its input to a 'busy' state to prevent further data arriving.

This busy condition persists until the buffer has emptied by a certain number of characters (again depending upon protocol) when the input is once again set to 'ready' in order to receive data once again.

1.4. Input/Output tradeoff.

It should be noted that although the buffer has several notionally separate activities, namely input and output and backward channel activities, it regards its main priority to be to serve the forward channel input.

Whereas serial data input at 9600 Baud fixes the maximum character rate at about 1000 characters/second, the rate of parallel data input is the lesser of

1) the maximum rate at which it can be transmitted and

2) the maximum rate at which it can be accepted.

The data buffer therefore tries to dedicate as much time as possible to the input task at the expense of the other tasks in order to maximise the transfer rate, it does however maintain a minimum throughput level for both output and backward channel that is normally sufficient to keep most printers busy.

In any event the condition will not persist because at very high rates of input the buffer will quite quickly become full and no longer able to accept input.

2. The Front Panel Controls.

2.1. The CLEAR switch.

Operation of the front panel CLEAR switch will put the data buffer into the initialised (empty) state, the current selection on the setup switches will be read and implemented and the inputs will be set to 'ready'.

2.2. The HOLD switch.

This switch causes output from the buffer to be temporarily halted. Input to the buffer still continues provided the buffer does not become full.

2.3. The COPY switch.

Operation of the COPY switch causes all data currently stored in the buffer to be repeated.

More specifically, this consists of all data that has passed through the buffer since it was last CLEARed, unless that amounts to more than the total capacity of the buffer (256k bytes) in which case only the last 256k bytes will be repeated.

If the COPY button is pressed whilst the buffer still contains data, then the copy operation itself will not be initiated until the entire current contents of the buffer has been emptied.

If the COPY button is operated repeatedly, then the number of times it has been pressed will be counted, and that number of complete copies will be produced.

3. The Front Panel Indicators.

3.1. The POWER indicator.

There are two indicators on the buffer, the POWER indicator shows that the unit has power applied to it.

The DB9 has no separate power on/off switch and so it should be isolated from the mains supply at a switched supply socket when not in use.

The data buffer may be connected to a common connector block along with other computer equipment that is to be in use at the same time, and the whole system switched on and off using the same mains switch.

3.2. The DATA indicator.

The DATA indicator provides information about the general level of activity in the buffer.

If there is no data in the buffer (and by implication no input or output activity), then the DATA indicator will not be lit.

If there is data in the buffer but no input or output activity, then the DATA indicator will flash at a very slow rate.

If there is output activity then the DATA indicator will flash at a medium rate.

If there is input activity then the DATA indicator will flash at a very fast rate.

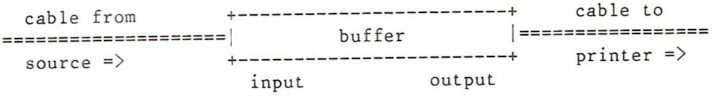
If the buffer is full then the DATA indicator will be continually lit.

This may seem at first sight to be rather complicated, but in use is fairly intuitive and serves to indicate the general level of activity in the buffer.

4. Connecting the data buffer.

4.1. Basic arrangement.

Arrange the data buffer between the source of data (often referred to as the 'host computer') and the destination as shown below.



The cables for input and output must be wired with due reference to the tables of input and output connections shown below, along with the information supplied with both the source machine and the printer (or whatever the destination device is).

In general it will be found that for relatively short cables (less than 4 metres), the parallel interfaces are simple to use and data transfer is very fast.

Short cable runs may be implemented using simple multi-way cable, but for longer runs use either ribbon cable or twisted-pair multi-way cable with each signal twisted with a ground to help prevent cross-talk in the cable.

The serial protocols are very suitable for data transfer over greater distances as fewer connections are required and hence the cabling is less expensive.

Such difficulties that may be encountered in establishing a data link can usually be overcome by careful reference to the information below for the buffer and the information supplied for the source and destination devices.

4.2. Input and Output connections.

When looking directly at the back panel of the data buffer, connectors to the left of the panel are input connectors, and connectors to the right are output connectors.

On the underside of the buffer there is two banks of setup switches, one for the input connectors and one for the output connectors.

5. The Parallel Input.

5.1. Parallel input connector.

The connector used is a 36-way connector compatible with the Amphenol 57-series commonly used for parallel printer cables.

This connector is found on the left of the back panel when looking directly at it.

To select the use of the parallel connector for input, input setup switches 1-4 must be switched OFF (for further information see the section dedicated to the setup switches).

5.2. Parallel input connector listing.

1)	-STROBE input	19)	0V
2)	D0 input	20)	0V
3)	D1 input	21)	0V
4)	D2 input	22)	0V
5)	D3 input	23)	0V
6)	D4 input	24)	0V
7)	D5 input	25)	0V
8)	D6 input	26)	0V
9)	D7 input	27)	0V
10)	-ACK output	28)	0V
11)	BUSY output	29)	0V
12)	PE output	30)	0V
13)	SELECT output	31)	-INIT input
14)	n/c	32)	-FAULT output
15)	n/c	33)	n/c
16)	0V	34)	n/c
17)	n/c	35)	n/c
18)	+5 Volt supply	36)	n/c

This pin configuration uses a sub-set that has been determined to operate successfully with the majority of the numerous different 'Centronics compatible' implementations found on a wide range of equipment.

Note that the PE (paper empty) and -FAULT signals are always false and the SELECT signal always shows true when the buffer is powered-on.

The +5 Volt supply on pin 18 of the connector can supply a maximum of 50mA to external interfaces etc..

5.3. Parallel input protocol.

1) If pin 11 (BUSY) is low then data may be set up on the input lines (pins 2-9) and left for at least 1us to settle.

2) Pin 1 (-STROBE) may be brought low for a minimum of 1us then set high again. The falling edge of the strobe latches the data on the input. The data on pins 2-9 must be held for at least 1 us hold time.

3) Less than 1us after the rising edge of -STROBE, pin 11 (BUSY) will be put high.

4) When the buffer is ready to accept another character it will put a low level on pin 11 (BUSY), then put pin 10 (-ACK) low for at least 1 us, then high again.

It should be noted that the signals on pins 10 and 11, BUSY and -ACK are alternative handshake signals, either or both of which may be used by the device sending data to the buffer.

6. The Parallel Output.

6.1. Parallel output connector.

The connector used is a 36-way connector compatible with the Amphenol 57-series commonly used for parallel printer cables.

This connector is found on the right of the back panel when looking directly at it.

To select the use of the parallel connector for output, output setup switches 1-4 must be switched OFF (for further information see the section dedicated to the setup switches).

6.2. Parallel output connector listing.

1)	-STROBE output	19)	0V
2)	D0 output	20)	0V
3)	D1 output	21)	0V
4)	D2 output	22)	0V
5)	D3 output	23)	0V
6)	D4 output	24)	0V
7)	D5 output	25)	0V
8)	D6 output	26)	0V
9)	D7 output	27)	0V
10)	n/c	28)	0V
11)	BUSY input	29)	0V
12)	n/c	30)	0V
13)	n/c	31)	n/c
14)	n/c	32)	n/c
15)	n/c	33)	n/c
16)	Ground	34)	n/c
17)	n/c	35)	n/c
18)	n/c	36)	n/c

6.3. Parallel output protocol.

1) The level at pin 11 (BUSY) is checked, and when it is low data is placed on pins 2-9 (D0-D7) and left for at least 2us to settle.

2) A low going pulse is produced on pin 1 (-STROBE) that is at least 2us in length.

3) If pin 11 (BUSY) is going to go high then it is expected to do so within 5 us (the data on pins 2-9 will remain steady for at least 5 us or until pin 11 (BUSY) is returned low, whichever is the longest).

It should be noted that the data buffer uses the handshake signal on pin 11 (BUSY) from the destination device, a connection to pin 10 (normally -ACK, acknowledge) is not required by the buffer.

In fact, normally only the active signals on pins 1-9 and pin 11 and one of the grounds are required for successful operation.

Caution should be exercised to determine which pins on the printer itself are connected to ground, usually all of the pins mentioned above are grounded but examples have been found where either the grounds on pins 14 and 16 or the grounds on pins 19-30 have been entirely omitted.

7. The Setup Switches.

7.1. Setup switch settings.

On the underside of the buffer one 8-way setup switch is provided for input port configuration, and another is provided for output port configuration.

It is necessary for these switches to be set correctly for the buffer to be able to communicate properly with other devices.

7.2. Setup switch listing.

Note:- 1 indicates that the switch is ON, 0 indicates that the switch is OFF. a '-' indicates that this combination is not used.

Switch	Output	Input
8	xon/xoff enabled	xon/xoff enabled
7	parity even	parity even
6	parity enabled	parity enabled
5	8 data bits	8 data bits
4		
3		
2		
1		
0 0 0 0	Parallel	Parallel
0 0 0 1	-	-
0 0 1 0	-	-
0 0 1 1	1200 out, 75 in	-
0 1 0 0	75 out, 1200 in	-
0 1 0 1	50-	~~50 baud~~
0 1 1 0	75 baud	75 baud
0 1 1 1	100 baud	100 baud
1 0 0 0	110 baud	110 baud
1 0 0 1	150 baud	150 baud
1 0 1 0	300 baud	300 baud
1 0 1 1	600 baud	600 baud
1 1 0 0	1200 baud	1200 baud
1 1 0 1	2400 baud	2400 baud
1 1 1 0	4800 baud	4800 baud
1 1 1 1	9600 baud	9600 baud

8. The Serial Ports.

8.1. Serial port operation.

The following general points should be noted with respect to serial port operation.

The baud rate, parity and word length set on a serial port applies to the both the send data and receive data functions on the same connector (with the exception of the split baud rate settings in the table above.)

On serial protocols the input only checks for one stop bit and the output always generates two stop bits. In situations where only one stop bit is required on output this may reduce the average character rate but the effect is very slight while ensuring compatibility and reducing the number of parameters to set up.

8.2. Serial port handshaking.

Handshaking is the general term for the orderly management of data transfer to and from a data port. It is sometimes known as 'Flow control'.

Hardware handshaking using DTR is probably the most common protocol encountered on microcomputers and printers although some types of equipment may utilise the XON/XOFF handshake protocol.

The data buffer always operates DTR handshaking even when XON/XOFF is selected, thus when using XON/XOFF on the serial output connector a suitable handshake signal is not available for connection to pin 20 (DTR), it may be 'tied high' by connecting it to a guaranteed high level (pin 6 (DSR) of the serial output connector is always high).

Similarly, if it is desired that low levels should not be seen on one or more of the input handshake signals from the buffer (for example, many a modem will hang up the line if a low level is encountered on its DTR), then the appropriate pins may be disconnected and if necessary tied high by connecting to a pin that is a guaranteed high level (pin 4 (RTS) on the serial input connector).

8.3. The backward channel.

The backward channel is a data path, present only when serial input and serial output is selected, from the output port to the input port, ie. data travels in the direction opposite to the main data path through the buffer.

This data path is secondary to the main function of the buffer, (which is of course to buffer in the forward direction,) and has itself only a minimum of buffering.

The function of the backward channel (apart from its use in software handshaking) is provided for use by such devices as KSR printers that are used as terminals, and similar devices. The main buffer (forward channel) providing the buffering action for the printer, the backward channel allowing the keyboard to be used to transmit to the host.

The use of the backward channel will normally be trouble-free when using hardware handshaking on both input and output, however there are some possible pitfalls when XON/XOFF is selected, and the following section should be noted.

8.4. Notes regarding backward channel operation.

1) The backward channel uses hardware handshakes only (DTR/CTS) regardless of the protocol selected on the main (forward) channel.

2) XON and XOFF codes are inserted and removed from the backward channel data stream by the protocol handlers for the forward channel and thus could be inadvertently removed from/inserted into 'escape sequences' or other context-dependent control strings. Thus backward channel data virtually must be pure ASCII text with no control codes apart from text-related codes like CR, LF, FF etc.

The backward channel input handshakes (pin 5 (CTS) on the output connector) could operate for either of the following reasons.

1) The input data rate is faster than the output data rate, and this may be true even though the input and output baud rates may be set the same if the received data consists of fewer data bits than the send data (due to parity, word length, or just because the output data always has two stop bits).

2) Because the backward channel send data handshake (pin 5 (CTS) of the input connector) has been put low by the host.

Note that backward channel activity is considered a task of secondary importance and, although characters arriving at the backward channel input whilst pin 5 (DTR) is high will always be accepted, the rate of backward channel output is dependent on the demands placed upon the data buffer by its higher-priority tasks, also, the amount of buffering it provides is quite small.

Therefore if the backward channel input is operated without handshakes it is by no means guaranteed that there will be no data loss. Reliable operation without handshakes is usually possible if slower Baud rates or slow average data rates (such as would be produced by a manually-operated keyboard for example) are used.

9. Serial Input.

9.1. The serial input connector.

The connectors used for serial input and output are 25-way 'D' female connectors compatible with the type normally used for RS-232 serial data cables.

The connector for serial input may be found on the left of the back-panel when looking directly at it.

Associated with the serial input connector is the 8-way input setup switch on the underside of the buffer, this switch must be configured correctly for the serial input to function properly.

9.2. Serial input connector listing.

Pin		Name	Function	Direction
1)		FG	frame ground	0V
2)		TXD	send (tx) data	(output) -14V
3)	*	RXD	receive (rx) data	(input)
4)		RTS	ready to send	(output) +5V
5)		CTS	send data handshake	(input)
6)	*	DSR	(not connected)	(input)
7)		SG	signal ground	0V
20)	*	DTR	receive data handshake	(output) +13V

In the listing given above, the signals marked with an asterisk (along with pin 7, signal ground) are those primarily concerned with the receive function on the input, the other signals are concerned with transmitting software handshakes (XON/XOFF) and data to be transmitted over the backward channel.

This connector is wired to look like a terminal or printer to a host computer (a DTE in communications terms). Many host computers have outputs that are designed to directly drive printers and terminals (they are wired as DCEs) and thus may be connected to this connector with a standard RS232C cable (wired pin 1-1, 2-2 ... etc.)

A notable exception are the many 'PC's and compatibles now available, these are usually themselves wired as terminals (DTEs) and will thus require a 'reversing' cable (sometimes called a NULL MODEM). This same situation exists when connecting a PC to a printer, and so a cable provided for that purpose is normally correct.

9.3. Serial input with DTR.

This protocol is implemented on the serial input port using the connector on the left of the back-panel when looking directly at it.

For a unidirectional application with hardware handshakes probably as few as three wires need connecting (depending upon the host machine), these are to pin 3 (RXD), pin 7 (0V) and pin 20 (DTR).

It should be noted that although pin 6 (DSR) is usually part of the receive function it is not required in this implementation.

If the input is to be run without handshakes then pin 20 (DTR) may be left unconnected, although care must be taken to ensure that the buffer never becomes full or else data loss will occur.

For bidirectional operation also connect pin 2 (TXD), pin 5 (CTS) and possibly pin 4 (RTS) will be required as well. If it is desired that the 'backward channel' output be operated without handshakes then pin 5 (CTS) may be connected back to pin 4 (RTS) within the connector to ensure transmission.

9.4. DTR Protocol listing.

1) Upon initialisation with this protocol selected, a high level is placed upon pin 20 (DTR) and pin 4 (RTS).

2) Serial data is accepted on pin 3 (RXD).

3) If the buffer becomes filled to within a certain number of characters of its total capacity, or if it can temporarily not receive data, it asserts a 'busy' status by putting a low level on pin 20 (DTR), however it continues to accept data as it arrives up to the point of overflow

   This margin of about 256 characters is provided to accomodate the 'buffered' or 'pipelined' serial output devices used on some machines, further data should NOT be deliberately transmitted after pin 20 (DTR) has gone low.

4) When the buffer is again ready to receive data, it asserts a 'ready' status on pin 20 (DTR) by putting it high again.

also :-

5) 'Backward channel' serial data is transmitted from pin 2 (DATA OUT) as and when it becomes available provided always that pin 5 (CTS) is at a high (ready) level.

9.5. Serial input with XON/XOFF.

This protocol is implemented on the serial input port using the connector on the left of the back-panel when looking directly at it.

The essential connections to implement this protocol are pin 3 (RXD), pin 7 (0V) and pin 2 (TXD) and pin 5 (CTS), the backward channel handshake.

It should be noted that, as before, although pin 6 (DSR) is usually part of the receive function it is not required in this implementation.

Pin 20 (DTR), the input handshake signal, may be either connected or left not connected as required. It will still be active during XON/XOFF activity and will indicate the state of readiness of the input.

XON and XOFF handshakes are always transmitted from pin 2 (TXD) regardless of the state of pin 5 (CTS) the backward channel handshake.

If there is no hardware handshake control line from the host to connect to pin 5 (CTS) then pin 5 may be connected in the input connector either to pin 7 (0V) in which case it will inhibit any backward channel data, or to pin 4 (RTS) and in this event the buffer will always send its backward channel data immediately it becomes available.

9.6. XON/XOFF protocol listing.

1) Upon initialisation with this protocol selected, a high level is placed upon pin 20 (DTR) and 4 (RTS).

2) Serial data is accepted on pin 3 (RXD).

3) If the buffer becomes filled to within a certain number of characters of its total capacity, or if it can temporarily not receive data, it asserts a 'busy' status by putting a low level on pin 20 (DTR) and transmits an XOFF character (ASCII code 13H, DC3, control 'S') from pin 2 (TXD) (provided that pin 5 (CTS) is high).

   Data is still accepted as it arrives on pin 3 (DATA IN) up to the point of overflow (if this occurs the 'buffer-overflow' LED is lit). This margin of about 256 characters is provided to accomodate the 'buffered' or 'pipelined' serial output devices used on some machines, further data should NOT be deliberately transmitted after pin 20 (DTR) has gone low or XOFF has been transmitted.

4) When the buffer is again ready to receive data, it asserts a 'ready' status on pin 20 (DTR) by putting it high again and transmits an XON character (ASCII code 11H, DC1, control 'Q') from pin 2 (DATA OUT).

also :-

5) 'Backward channel' serial data is transmitted from pin 2 (TXD) whenever it is available always provided that pin 5 (CTS) is high (ready).

10. Serial Output.

10.1. The serial output connector.

The connector for serial output may be found on the right of the back-panel when looking directly at it.

Associated with this connector is an 8-way setup switch on the underside of the buffer, this switch must be configured correctly for the serial output to function properly.

10.2. Serial output connector listing.

Pin		Name	Function	Direction
1)		FG	frame ground	0V
2)		TXD	receive data	(input)
3)	*	RXD	send data	(output)
4)		RTS	(not connected)	(input)
5)		CTS	receive data handshake	(output)
6)	*	DSR	ready to receive	(output)
7)		SG	signal ground	0V
20)	*	DTR	send data handshake	(input)

The signals marked with an asterisk (along with pin 7, signal ground) are those primarily concerned with the transmit function on the output, the other signals are concerned with receiving software handshakes (XON/XOFF) and data associated with the backward channel.

For a uni-directional buffer application with hardware handshakes (DTR), probably as few as three wires need connecting, these are to pin 3 (TXD), pin 7 (0V - signal ground) and 20 (DTR) although sometimes pin 6 (RTS) is required as well.

There is an apparent contradiction in the listing above between the conventional names (TXD, RXD etc) and the functions of the pins. This may perhaps be understood by considering that these names are the names of the connections rather than of the pins themselves.

This connector is wired as a DCE in communications terminology, and is suitable for connection to most printers and terminals (DTEs) using a standard RS232C cable (wired pin 1-1, 2-2 ... etc.).

10.3. Serial output with DTR.

This protocol is implemented on the serial output port using the connector on the right of the back-panel when looking directly at it.

For a unidirectional buffer application with hardware handshakes probably as few as three wires need connecting, these are to pin 3 (RXD), pin 7 (0V - ground) and 20 (DTR) although sometimes pin 6 (DSR) is required as well.

If transmission without regard to handshakes is desired then pin 20 (DTR) may be connected to the high level on pin 6 (DSR) within the connector to satisfy the handshakes and ensure transmission, needless to say the unit will then transmit as and when it has data to transmit with total disregard for the state of readiness of the destination device.

For a bidirectional implementation also connect pin 2 (TXD) and pin 5 (CTS) to provide handshakes, (pin 4 is not required).

10.4. DTR protocol listing.

1) Upon initialisation a high level is placed upon pin 6 (DSR) and 5 (CTS).

2) Pin 20 (DTR) is checked until a high level is detected upon it.

3) Serial data is transmitted from pin 3 (RXD).

4) Pin 20 (DTR) is continuously checked, and if it should return to a low level then serial transmission from pin 3 (RXD) will be stopped. There may be characters that are in the process of being transmitted at this time, these cannot be stopped and so to avoid data loss they must be accepted by the destination device, such characters will however be kept to an absolute minimum.

also :-

5) If the backward channel is able to accept characters, pin 5 (CTS) will be at a high level.

6) Data will be accepted at pin 2 (TXD).

7) If the backward channel becomes full or no longer able to accept data, pin 5 (CTS) will be put to a low level, data will continue to be accepted at pin 2 (RXD) to allow for overrun to the extent of about 8 characters before data will be lost, further data should NOT be deliberately transmitted to the buffer after CTS has gone low.

8) When the backward channel is once again ready to accept data it will put a high level on pin 5 (CTS).

10.5. Serial output with XON/XOFF.

This protocol is implemented on the serial output port using the connector on the right of the back-panel when looking directly at it.

Pins that need to be connected are PIN 2 (TXD), PIN 3 (RXD) and pin 7 (0V) although sometimes pin 6 (DSR) is required as well.

With XON/XOFF selected the hardware handshake on pin 20 is still active so in order to achieve transmission without regard to hardware handshakes pin 20 (DTR) may if necessary be 'tied high' by connecting it to the high level on pin 6 (DSR) of the output connector.

For a bidirectional data implementation you will normally need to connect pin 5 (CTS) to handshake the backward channel data, this is not otherwise required because XON/XOFF's are always acted upon regardless of the state of the backward channel input handshakes.

When XON/XOFF protocol is selected, XON and XOFF characters are removed from the backward channel data stream on reception.

10.6. XON/XOFF protocol listing.

1) Upon initialisation a high level is placed upon pin 6 (DSR) and 5 (CTS).

2) Pin 20 (DTR) is checked until a high level is detected upon it.

3) Serial data is transmitted from pin 3 (RXD).

4) Pin 20 (DTR) is continuously checked, and if it should return to a low level then serial transmission from pin 3 (RXD) will be stopped.

   In addition any data arriving at pin 2 (TXD) is checked and if an XOFF character (ASCII code 13H, DC3, control 'S') is received then it is removed from the backward channel data stream and transmission from pin 3 (RXD) will be stopped.

   There may be one or two characters that are in the process of being transmitted at this time, these cannot be stopped and so to avoid data loss they must be accepted by the destination device, such characters will however be kept to an absolute minimum.

5) On receipt of an XON character on pin 2 (TXD) (or a high level on pin 20 (DTR), if that is what stopped transmission) transmission is recommenced.

also :-

6) If the backward channel is able to accept data pin 5 (CTS) will be at a high level.

7) Data will be accepted at pin 2 (TXD).

8) If the backward channel becomes full or no longer able to accept data, pin 5 (CTS) will be put to a low level, data will continue to be accepted at pin 2 (TXD) to allow for overrun to the extent of about 8 characters before data will be lost, further data should NOT be deliberately transmitted to the buffer after CTS has gone low.

9) When the backward channel is once again ready to accept data it will put a high level on pin 5 (CTS).