A DRAGON-32 SERIAL PORT

Designed by Jim Fuller, G4WPI

Fully hardware compatible with the Dragon 64's RS232 port.
Leaves the expansion port free for a DOS cartridge etc.
Runs Dragon 64 communication programs.
Programmable baud rates from 50-19,200 baud.

A NOTE ABOUT COMPATABILITY

This serial port has been designed to emulate the serial port that is standard issue on the Dragon 64. The addition of this interface will allow your Dragon 32 to run most Dragon 64 communication programs provided that the program does not require 64K of RAM and that no calls to Dragon 64 ROM routines are used by the program.

The Dragon 64 ROM contains additional routines to support its serial port. These extra routines are naturally not present in the original Dragon 32 ROM but this is of no great importance since most communication programs use their own routines to drive the serial port. It is mentioned here simply for completeness.

THE R6551 ACIA

To ensure software compatibility with the Dragon 64, an R6551 ACIA is used at the heart of this interface. The ACIA is memory mapped at addresses $FF04 to $FF07. These four addresses have the following functions and uses.

ADDRESS	WRITE	READ
$FF04	TX Data Register	RX Data Register
$FF05	Programmed Reset	Status Register
$FF06	Command Register	Command Register
$FF07	Control Register	Control Register

The command register controls parity, receiver echo, transmitter interrupt control, the state of the RTS line, receiver interrupt control, and the state of the DTR line.

The control register controls the number of stop bits, word length, receiver clock source, and baud rate.

The status register indicates the states of the IRQ, DSR, and DCD lines, the transmit and receive data registers, overrun, framing and parity error conditions.

At first glance it appears that the state of the CTS line is not detectable by software, but by using logical interrogation of the status register this too can be detected, although most Dragon communication programmers appear to have overlooked this capability in the past.

PROGRAMMING EXAMPLES IN BASIC

This short program can be used to give your serial port a self test. Make sure that the connections out of your serial port are wired is described in the REM statements. Run the program. Anything that you type is sent out of the serial port, received back in again and then echoed to your screen.

Baud rate = 300, word length = 7 bits, 1 stop bit, even parity. Baud rates higher than 300 are too fast for Dragon BASIC. In such cases it is better to use a machine code routine.

```
 5 REM RS232 SELF TEST
10 REM CONNECTIONS FOR TX WITH TALKBACK:
20 REM STRAP CTS (5) TO 12V (3)
30 REM STRAP RXD (2) TO TXD (6)
40 REM
50 POKE&HFF06,&H6B:REM SET COMMAND REGISTER
60 POKE&HFF07,&H36:REM 1STOP BIT,7DATA BITS,300 BAUD
70 REM ANYTHING TO SEND
```

```
 80 CH$=INKEY$:IF CH$="" THEN 130
 90 REM AWAIT TX REG EMPTY
100 IF (PEEK(&HFF05)AND16)=0 THEN 90
110 REM SEND KEYBOARD CHAR
120 POKE&HFF04,ASC(CH$)
130 REM ANYTHING RECEIVED?
140 IF (PEEK(&HFF05)AND8)=0 THEN 70
150 REM ACCEPT INPUT
160 PRINTCHR$(PEEK(&HFF04))
170 GOTO 70
```

PROGRAMMING IN ASSEMBLY LANGUAGE

```
*A SIMPLE TERMINAL PROGRAM
*USING IRQ INTERRUPTS
*TO RECEIVE DATA.
*WRITTEN USING DREAM.

         ORG      $2000

START    LDA      #$38
         STA      $FF07
         LDA      #$6B
         STA      $FF06

*SET UP NEW IRQ VECTORS
         ORCC     #$10
         LDX      $10D
         STX      <IRQVTR,PC
         LEAX     <SERIN,PC
         STX      $10D
         LDA      $FF06          :ENABLE IRQ
         ANDA     #$FD           :FROM
         STA      $FF06          :ACIA CHIP.
         ANDCC    #$EF

KEYIN    JSR      $BBB5          :BLINK CURSOR
         JSR      $BBE5          :KEY INPUT?
         BEQ      KEYIN          :NO.
         CMPA     #3             :TEST FOR
         BEQ      END            :BREAK KEY.
KEYI1    LDB      $FF05          :READ STATUS REG
         BITB     #$10           :TDR EMPTY?
         BEQ      KEYI1          :BRANCH IF NOT.
         STA      $FF04          :STORE TO TDR
         JSR      $B54A          :PRINT CHAR TO SCREEN.
         BRA      KEYIN          :GET NEXT KEY.

*SERIAL RX ROUTINE
SERIN    LDA      $FF02          :SET PIA0
         LDA      $FF05          :RD STATUS REG
         BPL      SERI1          :BRANCH IF INTERRUPTED NOT BY ACIA
         BITA     #$8            :TEST IF RDR FULL?
         BEQ      SERI1          :BRANCH IF RDR EMPTY.
         LDA      $FF04          :READ CHAR FROM RDR.
         ANDA     #$7F           :CLEAR BIT 7.
         JSR      $B54A          :PRINT CHAR TO SCREEN.
SERI1    RTI                     :RETURN FROM INTERRUPT.
```

```
*RESET OLD IRQ VECTORS AND END PROGRAM
END     ORCC    #$10
        LDX     <IRQVTR,PC
        STX     $10D
        LDA     $FF06           :DISABLE IRQ
        ANDA    #2              :INTERRUPT
        STA     $FF06           :BY ACIA.
        ANDCC   #$EF
        RTS                     :RETURN TO BASIC.

IRQVTR  RMB     2               :TEMP STORE FOR ORIGINAL IRQ VECTOR.
```

A MACHINE CODE ROUTINE TO REDIRECT PRINTER OUTPUT TO THE SERIAL PORT

This routine uses the Dragon's RAM hook at address $167 to re-direct printer output to the serial port. Assemble the following code and save it to tape or disk as a machine code file. When you wish to use a serial printer, load the program into a free RAM location (eg. CLOADM"SERPRINT.BIN",&H7800) and then EXEC it. Subsequently, all printer output will then be routed to your serial port.

```
*ROUTINE TO REDIRECT CENTRONICS
*OUTPUT TO THE SERIAL PORT
*
*RELOCATABLE CODE
*
*EOL DELAY APPROX 1/2 SECS.

SERPNT  LDA     #$7E
        STA     $167
        LEAX    <POUT,PC
        STX     $168
        LDA     #$A
        STA     $FF06
        LDA     #$98
        STA     $FF07
        RTS
POUT    PSHS    A
        LDA     <$6F
        CMPA    #-2
        BEQ     OUTC
        PULS    A,PC
OUTC    LEAS    2,S
        PSHS    A
POLL    LDA     $FF05
        BITA    #$10
        BEQ     POLL
        PULS    A
        STA     $FF04
EOL     CMPA    #$0D
        BNE     END
EOLDLY  PSHS    X
        LDX     #54000          :500MS APPROX.
TLOOP   LEAX    -1,X
        BNE     TLOOP
        PULS    X
END     RTS
```

LLISTING BASIC PROGRAMS TO A SERIAL PRINTER

The machine code routine as listed on Page 3 works very well when used with the PRINT#-2 command to send a text string to the printer but when the LLIST command is used to obtain a listing of a BASIC program, the result is usually that the listing is printed all on one continuous line. This problem is due to the way the command is interpreted in the DRAGON's ROM. If your printer LLISTs programs in such a way then try using the following program instead.

Type in the following BASIC program taking care that the DATA statements are all typed in correctly. After typing in the program, save a copy of it to tape or disk before doing anything else.

You may now RUN the program. This will set-up a small machine-code routine in memory from $7700 to $7797. When the program stops and the "OK?" prompt is displayed, you can now save this small block of machine-code for use another time.

If you want to save the program to disk, use: SAVE"LLIST.BIN",&H7700,&H7797,&H7700

If you want to save the program to tape, use: CSAVEM"LLIST",&H7700,&H7797,&H7700

Now that you safely have a copy of the machine-code, there is no longer a need to keep a copy of the BASIC loader program so you may now delete it.

```
10 'HEX COMPILER FOR A ROUTINE
20 'TO LLIST A BASIC PROGRAM TO
30 'THE SERIAL-PORT.
40 '(C) J.T.FULLER 1988
50 DATA BD,BA,77,86,0A,B7,FF,06,86,98,B7,FF,07,0F,9C,8D,03,8D,27,39,BE,01,77,AF,8C,79,BE,01,68,AF,8C,71
60 DATA 30,8C,0A,BF,01,77,30,8C,29,BF,01,68,39,AE,8C,63,BF,01,77,AE,8C,5B,BF,01,68,39,86,10,97,89,0F,2B
70 DATA 0F,2C,BD,83,FF,CE,FF,FF,DF,2B,BD,8E,D6,86,0D,8D,01,39,32,62,34,02,B6,FF,05,85,10,27,F9,35,02,B7
80 DATA FF,04,81,0A,27,1F,81,0D,27,0C,0C,9C,96,9C,91,9B,25,13,86,0D,20,DE,8D,0E,0F,9C,B6,01,4A,81,01,27
90 DATA 04,86,0A,20,CF,39,34,10,8E,D6,D8,30,1F,26,FC,35,90,00,00,00,00,00,00,00
100 CLS
110 FOR B=&H7700 TO &H7797
120 READA$
130 PRINT@237,HEX$(B)
140 POKEB,VAL("&H"+A$)
150 NEXTB
```

USING THE NEW LLIST PROGRAM

Switch the Dragon OFF and then back ON again. Type in the following POKE statements:

POKE 155,X (where X=the maximum line width of your printer. The default value is 132)
POKE 330,Y (where Y=2 if your printer needs a carriage-return and line-feed sequence at the end of each printed line, or Y=1 if only a single carriage return is adequate. The default value is 1)

Now load "LLIST.BIN" into memory. This machine-code is written in position-independent code so that you may load the program to any free RAM location, however the safest place is probably at its default location at $7700.

Next load in the the BASIC program that you want to make a listing of and then type EXEC&H7700 to list the program to your printer.

Status Register – $FF05

The Status register is used to indicate to the processor the status of various functions.

	Status	Set by	Cleared by
0	Parity Error*	0 = No error 1 = Error	Self Clearing**
1	Framing Error	0 = No error 1 = Error	Self Clearing**
2	Overrun*	0 = No Error 1 = Error	Self Clearing**
3	Receive Data Register Full	0 = Not full 1 = Full	Read Receive Data Register
4	Transmit Data Register Empty	0 = Not Empty 1 = Empty	Write Transmit Data Register
5	DCD	0 = DCD Low 1 = DCD High	Not resettable Reflects DCD State
6	DSR	0 = DSR Low 1 = DSR High	Not resettable Reflects DSR State
7	IRQ	0 = No Interrupt 1 = Interrupt	Read Status Register

	7	6	5	4	3	2	1	0
HARDWARE RESET	0	0	0	0	0	0	0	0
PROGRAM RESET	-	-	-	0	0	0	0	0

Command Register – $FF06

7	6	5	4	3	2	1	0

PARITY CHECK CONTROLS

BIT 7	BIT 6	BIT 5	OPERATION
-	-	0	Parity Disabled - No parity Bit Generated - No Parity Bit Received
0	0	1	Odd Parity Receiver and Transmitter
0	1	1	Even Parity Receiver and Transmitter
1	0	1	Mark Parity Bit Transmitted, Parity Check Disabled
1	1	1	Space Parity Bit Transmitted, Parity Check Disabled

DATA TERMINAL READY

0 = Disable Receiver and all Interrupts (DTR high)

1 = Enable Receiver and all Interrupts (DTR low)

RECEIVER INTERRUPT ENABLE

0 = IRQ Interrupt Enabled from Bit 0 of Status Reg.

1 = IRQ Interrupt Disabled

TRANSMIT CONTROLS

BIT 3	BIT 2	TRANSMIT INTERRUPT	RTS LEVEL	TRANSMITTER
0	0	Disabled	High	Off
0	1	Enabled	Low	On
1	0	Disabled	Low	On
1	1	Disabled	Low	Transmit BRK

NORMAL/ECHO MODE FOR RECEIVER

0 = Normal

1 = Echo (Bits 2 and 3 must be 0)

	7	6	5	4	3	2	1	0
HARDWARE RESET	0	0	0	0	0	0	0	0
PROGRAM RESET	-	-	-	0	0	0	0	0

DRAGON 32 SERIAL-PORT (C) J.T. FULLER OCTOBER 1987

Control Register – $FF07

7	6	5	4	3	2	1	0

STOP BITS

0 = 1 Stop Bit
1 = 2 Stop Bits
1 Stop Bit if Word Length=8 Bits & prty.
1.5 Stop Bits if Word Length=5 Bits &no prty.

WORD LENGTH

BIT 6	BIT 5	DATA WORD LENGTH
0	0	8
0	1	7
1	0	6
1	1	5

RECEIVER CLOCK SOURCE

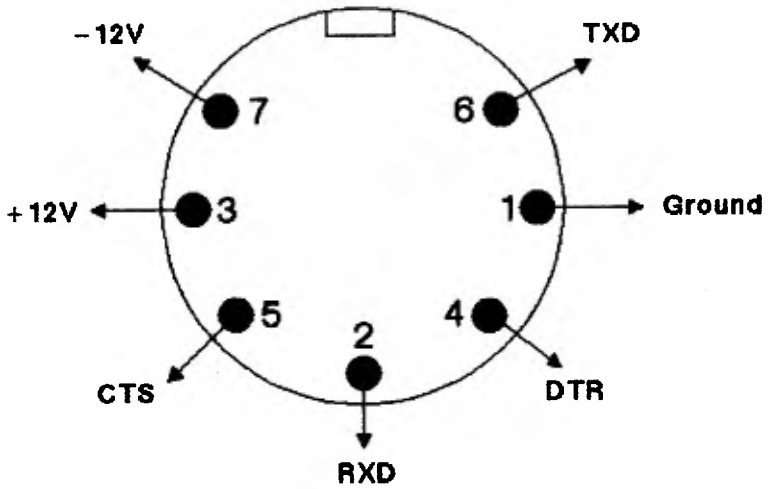
0 = External Receiver Clock
1 = Baud Rate Generator

7	6	5	4	3	2	1	0	
0	0	0	0	0	0	0	0	HARDWARE RESET
–	–	–	–	–	–	–	–	PROGRAM RESET

BAUD RATE GENERATOR

0	0	0	0	16x EXTERNAL CLOCK
0	0	0	1	50 BAUD
0	0	1	0	75
0	0	1	1	109.92
0	1	0	0	134.58
0	1	0	1	150
0	1	1	0	300
0	1	1	1	600
1	0	0	0	1200
1	0	0	1	1800
1	0	1	0	2400
1	0	1	1	3600
1	1	0	0	4800
1	1	0	1	7200
1	1	1	0	9600
1	1	1	1	19,200

DRAGON RS-232 PIN-OUTS



(Viewed from the side of the case)

DRAGON 32 SERIAL PORT

FITTING INSTRUCTIONS

All reasonable precautions have been made to ensure the reliability of the following instructions. Incorrect wiring to the Dragon's expansion port can damage expensive components so please take care and check all connections carefully.

Apologies to all who already have sufficient knowledge of the Dragon's internals and to whom some of the directions given here may seem over simplified.

TOOLS REQUIRED

A narrow-tipped soldering iron, a screw-driver, a drill, drill bits 15.9mm (5/8ths) and 3.2mm, a small pair of narrow wire-clippers.

REMOVAL OF THE DRAGON CIRCUIT BOARD

1. Switch of the power supply and disconnect all external leads to the Dragon.
2. Turn the Dragon upside-down and remove the four screws that hold the case together. Turn the computer upright again and lift off the lid exposing the inside of your Dragon.
3. Unscrew the four screws securing the keyboard and remove the two bits of black plastic at each side of the keyboard. The keyboard is linked to the main circuit-board by a short length of ribbon cable. This cable meets the main board at a white plastic connector which can now be disconnected to free the keyboard. After you have removed the keyboard, avoid flexing the ribbon cable about too much because any rough handling or bending of the cable is likely to cause a break in one or more of the cables wires. This will result in disabling a row of keys on the keyboard after you re-assembled the computer. It is wise to lay the keyboard down on its face so that the weight is not pressing down on the ribbon cable.
4. Inside the computer you will now see two separate boards. The board at the rear of the computer, and with a large metal heat sink attached, is the power supply board. The two boards are connected to each other by another short length of ribbon cable which you can now disconnect.
5. On the main circuit board, near the cartridge port, you will find the microprocessor chip (IC19) which will bear the number 6809E. For safety, it is best to remove this chip before doing any soldering to the computer. Removal of this chip must be done very carefully so as no to bend any of the legs too much. The chip is not soldered to the board but is held in an IC holder. The best way to remove the chip in the absence of a proper IC remover is to use a screwdriver to prise out the chip from its holder. Do this gradually and try not to touch any of the legs with the screwdriver if possible. Make a note of which way round the chip sits so that it is not refitted the wrong way later.
6. Above the cartridge port, at the right-hand side of the computer, you will see a black plastic cover held in place by two screws. Remove these screws and lift off the black cover. There should now be just four remaining screws holding in the main circuit board which can now bw removed. With careful manoeuvering you should now be able to take out the main circuit board from the case.

SOLDERING TO THE CIRCUIT BOARD

7. Lay down the Dragon's circuit board, component side down, on a level table top. Position it so that the cassette port is at the left-hand side, the cartridge port is at the right-hand side, and the RESET switch is at the bottom left (see diagram 4).
8. On the underside of the cartridge port, find the two rows of 20 pins where the cartridge port connector has been soldered to the main circuit board (diagram 3). This is the place where you will solder most of the connections to the new board. Before doing any soldering to these pins, it helps if you put a touch more solder to each one so as to give the new leads something more substantial to hold them.
9. Position the new board on top of the Dragon's circuit board with the double ribbon cable on the right-hand side (diagram 4). Find the first ribbon cable which is the one with colours running from BLACK through to YELLOW. All the wires on this first ribbon cable are soldered to odd-numbered pins on the cartridge port. Lay the cable flat against the main circuit board and slide the cable up to the odd numbered row of pins. Follow the colour coding to ensure that the leads of the cable are soldered to the right places (diagram

3). Each wire from the ribbon cable should rest directly on top of the pins and parallel to the surface of the Dragon board so that the cable can lie flat against it. Be very careful that no solder bridges across to any of the other pins.

10. Now find the second ribbon cable which has colours running from WHITE through to ORANGE. All the wires on this cable are to be soldered to even numbered pins on the cartridge port, these are the pins making up the row nearest to the edge of the board. When all of the wires have been connected properly, both ribbons must be able to lie flat against the Dragon circuit board. There is little room available under the circuit board when it is fitted back into the case so any cable that sticks up in the air too much will hinder the replacement of the board.
11. Connect leads from pillars 1-4 of the interface to their appropriate points on the underside of the Dragon circuit board (diagram 1). There are many places where you can connect the lead to GROUND, try anywhere connected to the wide track that encircles the edge of the board or perhaps pins 33 & 34 of the expansion port. The +5 Volts can be found at Pin 2 of the connector to the power supply board. Pin 1 of this connector is the one furthest from the cassette port.
12. Use a hole punch or a drill to make a hole of about 5/8 inch at the left hand side of the Dragon's case, underneath where the keyboard sits. This will be used to hold the DIN connector. Drill two smaller holes at both sides of this larger hole to allow for the insertion of self-tapping screws to hold in the socket. Four more holes (approx 3.2mm diam.) can now be drilled through the base of the Dragon's case, at the right-hand side, under the keyboard. These four holes will be used to fix the new RS232 board so you could lay down the board and use it as a template to judge where to drill the holes.
13. Turn the interface board over so that it is up the same way as the Dragon's circuit board, with the component side facing downwards. Fold the ribbon cable as shown in diagram 4. Now turn the whole lot back up the right way, component side facing upwards. You may now fit the main board back into the case, but before you do, it is important to put a piece of card (not too thick) between the ribbon cable and the underside of the main board at the point where the cable folds over. This card will help in protecting against the possibility of the insulation on the cable being pierced by sharp projections on the underside of the main circuit board.
14. Refit the board back into the base of the Dragon's case. Refit the black plastic cover that goes over the cartridge port and then replace all screws to secure the main circuit board. Fix the interface board using small bolts pushed through the holes which you have already drilled into the base of the case under where the keyboard will sit. Small spacers will be needed between the interface and the base of the case but do not tighten them down too much. Finger tight should do.
15. Locate pins 22 and 24 of IC26 (6821 PIA chip) and using a small pair of narrow wire clippers, carefully cut the legs of each of these two pins getting as close to the surface of the circuit board as possible. Bend each of the two pins out at right angles and join then together by soldering a very short piece of wire between them. To these two pins, you can now solder the lead running from pillar 5 of the interface.

WIRING THE 7 PIN DIN SOCKET

16. There should now be just one length of ribbon cable left unattached. This is used to connect to the 7 pin DIN socket which you can now fix into the 5/8 inch hole that you previously made in the side of the case.
17. All the wires to the DIN socket are again colour coded so follow the directions given in diagram 2 to ensure the correct connections.

FINISHING OFF

18. Replace the 6809 processor (IC19) back into its socket on the main circuit board.
19. Reconnect the short ribbon cable between the main circuit board and the power supply board.
20. Reconnect the short ribbon cable coming from the keyboard and refit the keyboard back into its proper place in the case.
21. Put the cover back on the case and secure with the four corner screws.
22. This concludes the fitting of the Dragon 32 serial port upgrade.

TESTING

23. Try using the BASIC "RS232 SELF TEST" program with this documentation. Make sure that the the correct straps are made between pins 5 & 3 and pins 2 & 6.

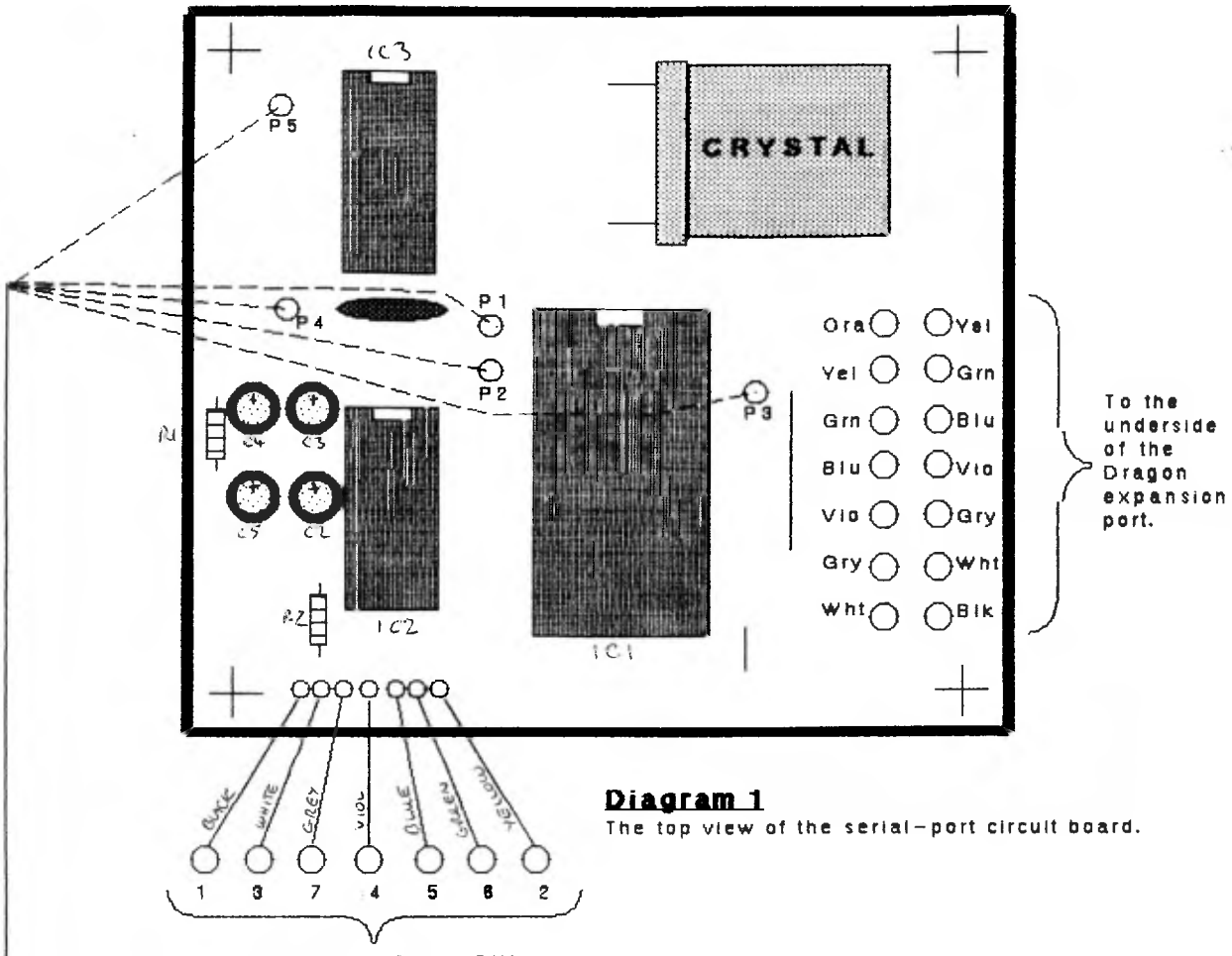


Diagram 1
The top view of the serial-port circuit board.

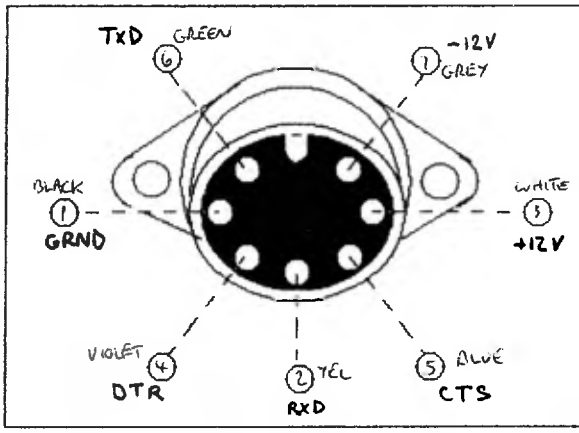


Diagram 2
The rear view of the 7-pin DIN connector, as seen from the inside of the case, showing colour coded connections to the serial-port circuit board.

Pillar 1 - Ground.
Pillar 2 - Underside of IC33 (74LS138) pin 11
Pillar 3 - Underside of IC26 (6821 PIA) pins 37&38
Pillar 4 - +5 volts, (Pin 2 of the ribbon cable from the PSU board.)
Pillar 5 - Topside of IC26 (6821 PIA) to pins 22&24 after they have been snipped.

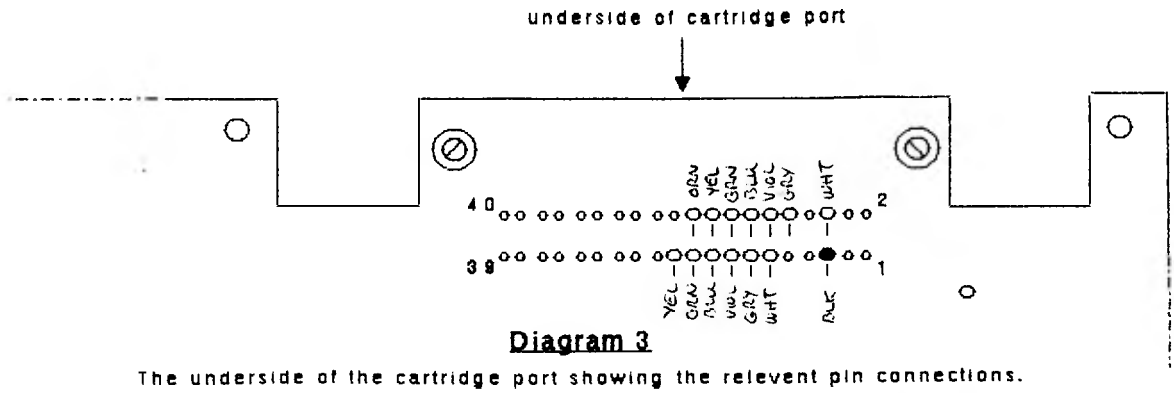


Diagram 3

The underside of the cartridge port showing the relevent pin connections.

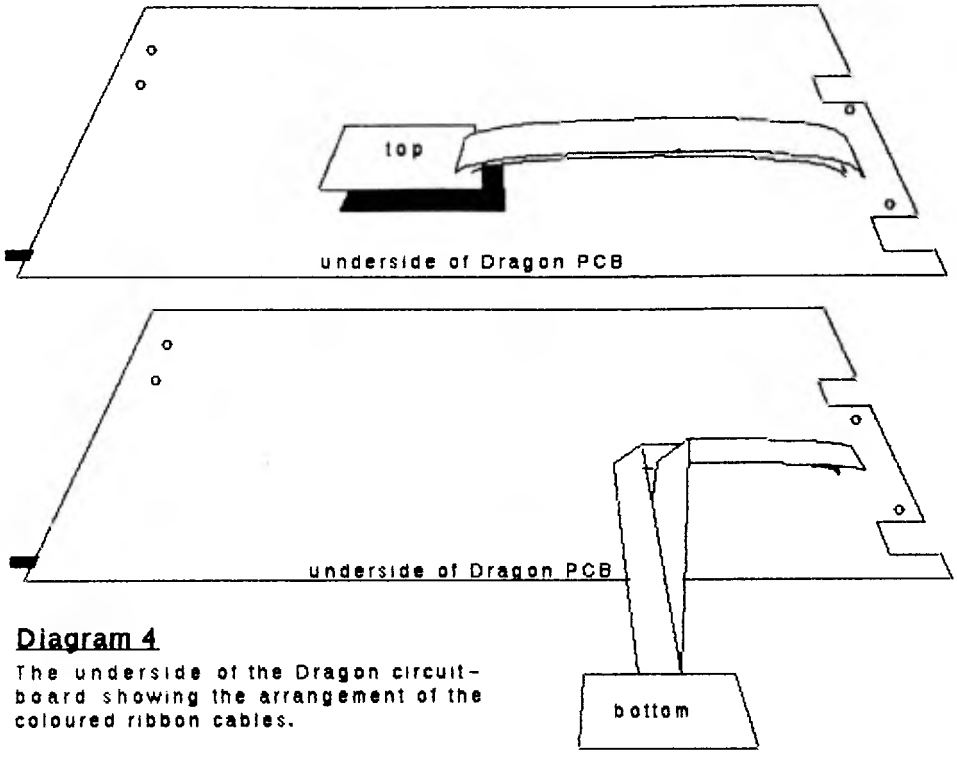


Diagram 4

The underside of the Dragon circuit-board showing the arrangement of the coloured ribbon cables.

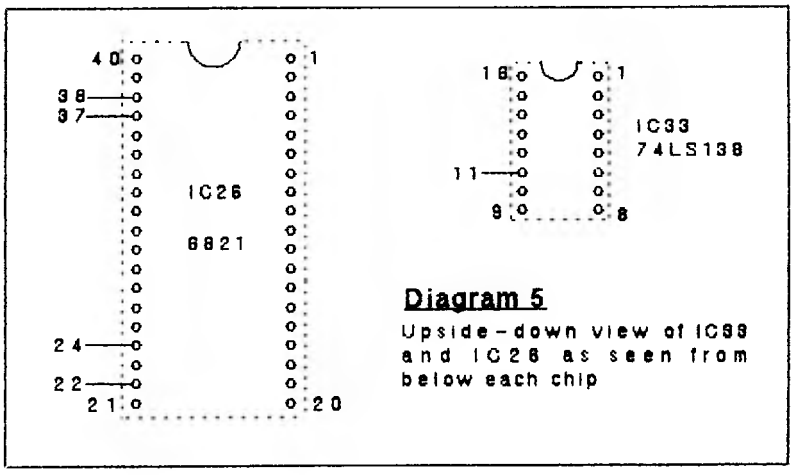
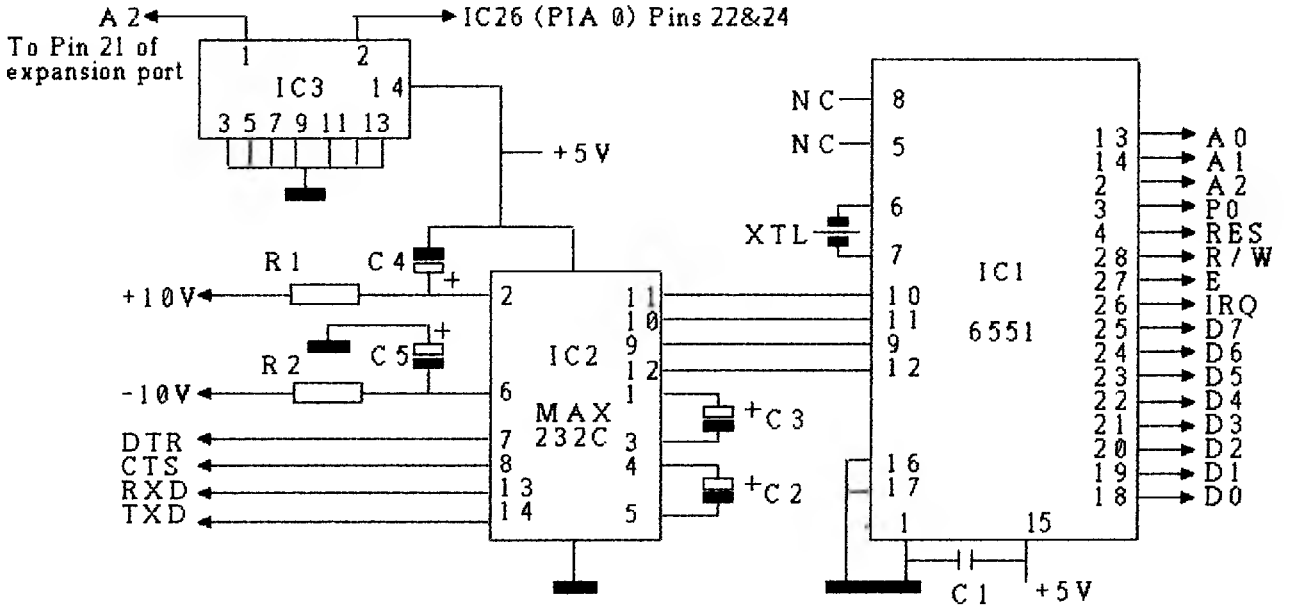


Diagram 5

Upside-down view of IC33 and IC26 as seen from below each chip

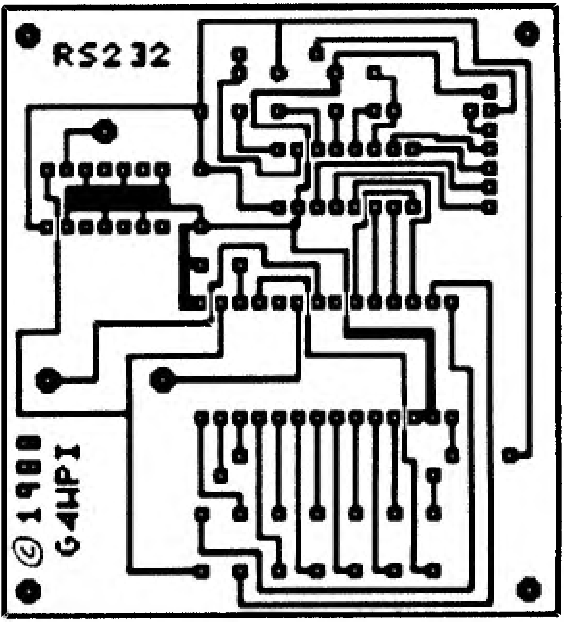
Dragon 32 Serial Port

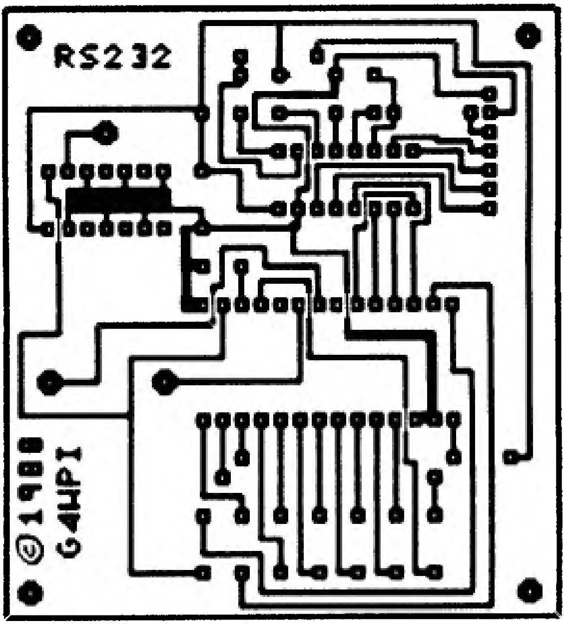
J.T. Fuller, G4WPI



Component List

IC1	6551A	Maplin	UR03D
IC2	MAX232C	"	FD92A
IC3	4069UBE	"	QX25C
XTL	1.8432 Mhz crystal	"	UM96E
C1	0.1uf ceramic cap	"	YR75S
C2,C3,C4,C5	22uf sub-min elect, or tantalum caps	"	YY36P
R1,R2	2K2 0.25W resistors	"	E2K2
	7 pin DIN chassis socket	"	HH37S
	7 pin DIN plug	"	HH30S
	16 way IDC ribbon cable	"	XR80B


RS232
©1988
G4WPI


RS232
©1988
G4WPI

V E R S I O N 4.0

D R A G O N P A C K E T - R A D I O T E R M I N A L P R O G R A M

(C) J.T. Fuller G4WPI

42 kitchener Road, Amesbury, Wilts, SP4 7AD. Tel. (0980) 622158

GENERAL

* Runs on a Dragon 64, or a Dragon 32 with a D-64 type serial-port.
* If you are using a 64K Dragon, the program will relocate itself to make use of the extra memory available.
* Written in assembly language (machine code) occupying nearly 7K of memory.
* Works with all popular makes of TNC. KPC-2, KPC-4, KAM, TNC-200, TNC-220, TNC-2, TINY-2, MICROPOWER-2, PK-80, PK-87, PK88, PK-232, MFJ, G0BSX Mk2, and others.

* Supports 300, 600, 1200 baud RS232 computer baud rates at all legal (10 bit) protocols.
* Supports both hardware or software flow control using DTR/CTS, RTS/CTS, or XON/XOFF handshaking.
* Instructions given on how to customise the program's default baud rate to suit your own preference.
* Details provided for making up your own Dragon/TNC RS232 lead.

SCREEN DISPLAY

* Uses the Dragon's high resolution graphics screen.
* 24 lines with 51 characters per line.
* Optional black text on green background or green text on black background.
* Screen protection facility - inverts the display if left unattended. Helps protect against "screen burn" and prolongs the life of your TV/monitor.
* Split-screen display. Uses two windows. The received text is displayed in the upper window and transmitted text displayed in the bottom window.
* A status line separates the two windows and indicates the status (ON or OFF) of various buffers and facilities used by the software.

TEXT HANDLING FACILITIES

* A Capture buffer facility stores both received and transmitted text. Holds room for 22,139 characters on a Dragon 64 or 14,203 on a Dragon 32.
* A Message buffer can be used to prepare your own text files. Holds up to 22,139 characters on a Dragon 64 (14,203 on a D-32).
* Text files may be listed to the screen, printer or transmitted to another station or BBS via your TNC.
* The program incorporates facilities to load and save files to tape or Dragondos disk.
* The Autosave facility can automatically save data to disk when the capture buffer fills up.
* Received text can be simultaneously sent to the screen and a printer for a hard-copy.

KEYBOARD

* Unbuffered keyboard input, characters typed into the keyboard are immediately sent to the TNC.
* Repeat key facility incorporated. Even works on the Dragon 32!
* Enhanced upper and lower case ASCII character sets incorporating the "_\^{}|" characters which are not available on the standard Dragon display.
* Right arrow key sends the ")>" string followed by a carriage-return, for use at the end of your overs.
* Keys provided to send "RYRYRY" or "+?" for use with multimode TNC's.
* A Handy keyboard overlay is provided to familiarise the user with the use of the various function keys.

COST (P&P included)

Please make any cheques payable to J.T.Fuller, stating callsign or BBS number, and whether tape or disk format is required.

V4.0 Tape or Dragondos disk	..... £10-00.
EPROM (2764 type)	..... £13-00 *

* Due to a shortage of Dragon EPROM cartridges I regret that one cannot be supplied with your EPROM. Most non-games type cartridges are suitable provided they take a 2764 EPROM. Cartridges previously used for BMK packet. Ex-RTTY, AMTOR, & CW, type cartridges should be fine.

10/92