INSTRUCTION MANUAL
DELUXE RS232 INTERFACE

V2.a

Omnitronix, Inc.
PO Box 43
Mercer Is., WA 98040
(206) 236-2983

WHAT IS BAUD RATE, PARITY, ETC???

Baud Rate refers to the speed at which data is sent across a wire. 1200 Baud is four times as fast as 300 Baud. Other than that, the best simple answer to that is that you don't need to know in order to make your printer work. The most important thing you DO need to know is that, when hooking two RS232 machines together (e.g. your computer and your printer), you have to have both machines set with the same settings. If your printer is set at 1200 baud, you need to set your computer at 1200 baud, and so on. How do you do that? First, you look up in your printer manual and find out how it is set. If you must choose your printer settings, use 1200 Baud, 8 Bit word, 1 Stop Bit, No Parity. Now you have established how your printer is set. You now should attempt to set the computer the same way. If you are using a very friendly program, it will ask you for what Baud you want to use, and what Parity, etc. In more simple programs, somewhere you will be asked to enter two numbers into the computer program, the Control Register and the Command register. These two numbers indicate to the computer what Baud rate, etc., it should operate at. More on these Control and Command Register numbers later. You get your program set correctly with the same settings as the printers settings. When you use your program now, it should print perfectly. If it does print, but it prints garbage, you got the two numbers wrong.

BASIC TERMINAL PROGRAM ROUTINE

There are several ways to send and receive data thru the USER I/O port. An example of one way to do so is the BASIC Terminal Program below.

```
10 REM SET UP VARIABLES AND GOSUB STANDARD/COMMODORE ASCII CONVERSION
20 OPEN2,2,3,CHR$(38)+CHR$(96):PRINTCHR$(144)
30 C$=CHR$(147):K$=CHR$(20):J$=CHR$(187):CR$=CHR$(13)
40 PRINTC$;"LOADING DATA...":GOSUB300
100 PRINTC$;"TERMINAL MODE":PRINTJ$;
105 REM MAIN PROGRAM LOOP
110 GET#2,A$:IFA$=""THEN130
120 A=A%(ASC(A$)):PRINTK$+CHR$(A)+J$;:GOTO110
130 GETA$:IFA$=""THEN110
200 PRINT#2,CHR$(T%(ASC(A$)));:GOTO110
295 REM THIS SETS UP STANDARD/COMMODORE ASCII CONVERSION
300 DIMR%(255),T%(255)
310 FORZ=32TO64:T%(Z)=Z:NEXT:T%(13)=13:T%(20)=8:T%(160)=32
320 FORZ=65TO90:Y=Z+32:T%(Z)=Y:NEXT:FORZ=91TO95:T%(Z)=Z:NEXT
330 FORZ=193TO218:Y=Z-128:T%(Z)=Y:NEXT
340 T%(133)=03:T%(134)=19:T%(135)=17:T%(136)=16
350 FORZ=0TO255:Y=T%(Z):IFY<>0THENR%(Y)=Z
360 NEXTZ
390 RETURN
```

An OPEN statement in line 20 opens a channel to the USER I/O port, and the PRINT# and GET# BASIC commands found in lines 110 - 200 send and receive the modem data. The above program allows you to use the VIC-20 or C64 with a 300 Baud modem. It has nothing fancy in it, but it works. What we will do here is examine the parts that specifically involve the RS232 channel, so you will understand it and can modify it if you want to. For a complete description of these features, see the PROGRAMMERS REFERENCE GUIDE for the Commodore.

THE OPEN COMMAND

The OPEN command is the key to using the USER I/O port for serial communications. It determines the baud rate, word length, parity, stop bits, and handshaking. The OPEN command for RS232 communication is fully described in the PROGRAMMERS REFERENCE GUIDE for your computer. You have to make sure you get this OPEN command right, because if you get the settings wrong, your modem or printer won't be able to make any sense out of the data you send it.

You see the OPEN command in line 20 of the program above. The exact command is:

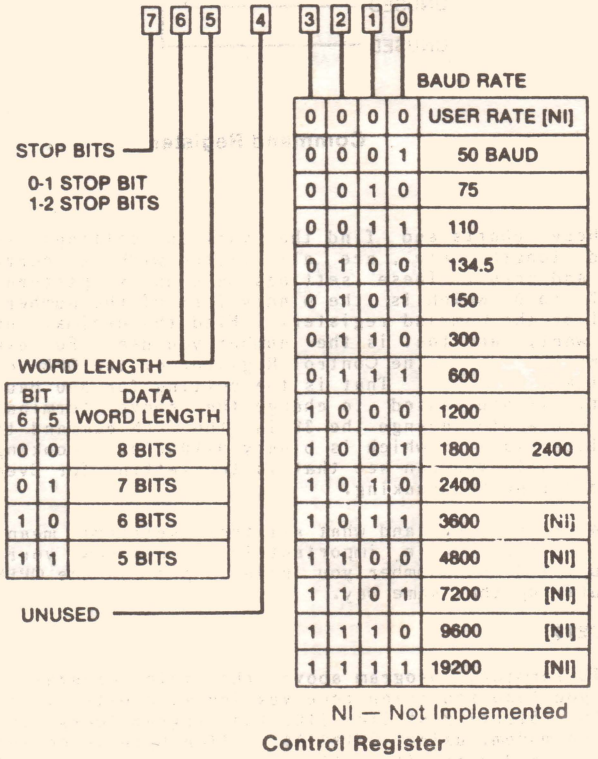
OPEN2,2,3,CHR$(38)+CHR$(96)

The first 2 in the command is the LOGICAL FILE NUMBER. This can be any number from 1 to 255 that you wish to use. Once the program has executed your OPEN command, the Logical File number is then used in your PRINT# and GET#s to specifiy where PRINT and GET from (example: PRINT#2 or GET#2). That is pretty much all you need to know about the first 2.

The second 2 is the device number for the USER I/O port. If it was an 8, the data would go to a disk drive, and a 1 would send the data to the cassette recorder. As we are dealing with the USER I/O port, the number is always a 2.

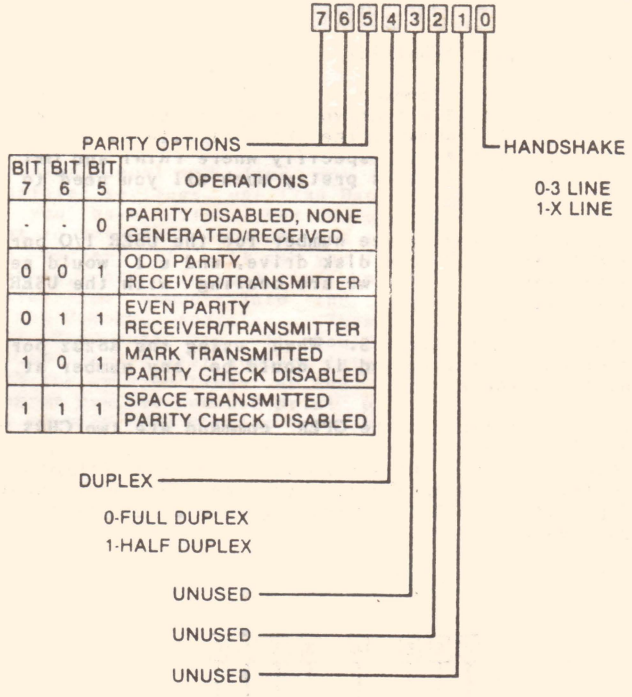
The third number is a 3. When using the RS232 port, this three serves no purpose at all, and it could be any number at all. But some number has to be there.

The last two items in the OPEN command are two CHR$ bytes (CHR$ is pronounced "character string"). The first of these is called the CONTROL REGISTER, and the second is called the COMMAND register. By using the two following charts, you can determine what numbers the Control Register and the Command Register should be for the particular setting you want.



NI — Not Implemented

Control Register



BIT 7	BIT 6	BIT 5	OPERATIONS
-	-	0	PARITY DISABLED, NONE GENERATED/RECEIVED
0	0	1	ODD PARITY RECEIVER/TRANSMITTER
0	1	1	EVEN PARITY RECEIVER/TRANSMITTER
1	0	1	MARK TRANSMITTED PARITY CHECK DISABLED
1	1	1	SPACE TRANSMITTED PARITY CHECK DISABLED

Command Register

Look over these charts and find the various settings you want. Baud rate, word length, etc., are all listed with a corresponding pattern of ones and zeros. These settings make up a pattern of on or off bits, from 7 to 0, which is the binary form of the number you use for the Control or the Command register. Find the decimal number for the setting you want, and that is the number you use. For example, in the terminal program above the Control Register of the OPEN command is 38 which is a binary 00100110. That is the setting for 300 Baud, 8 Bit word, 1 stop bit. If you wanted to change the above terminal routine for 1200 baud, you would change the 38 in the OPEN command to a 40. The Command Register is a 96 which is binary 01100000. Looking at the Command Register chart you can see that is the setting for Even Parity, Full Duplex, and 3 line handshaking.

You don't have to understand what all these settings mean in order to be able to use them. More important is to know how your printer should be set up, and what number you have to put in the OPEN command to set the computer up that same way.

PRINT# AND GET#

In the BASIC Terminal program above, the main program loop which sends the data you type and which receives incoming data is found with lines 110 - 200. Starting at line 110, the program looks for a byte coming in from the modem, using GET#2,A$. If a byte is coming in, it gets the character and puts it in A$. If it sees one, line 120 prints it on the screen and then goes back to line 110 to look for another. If there are no incoming bytes, the program goes to line 130 to look for keyboard input. If you are typing something, it puts a character in A$ and goes to line 200 to print it out to the modem using PRINT#2,A$. The additional data in line 200 converts the character you typed from Commodore ASCII to Standard ASCII. The program keeps on sending the data you type until it doesn't see any more. It then goes back to line 110 and starts all over.

These are the simple basics for a terminal program. There can be many fancy modifications you can do to it, but the basic routine is right there. Additionally, two other program modules have been included in this manual, to let you see more ways to program using the PRINT# command. These modules have line numbers specifically planned to allow them to be merged directly into the above program. If you wish to add a module, be sure you type in the BASIC Terminal program with exactly the same line numbers as above, and the routine with exactly the same line numbers as given below.

```
     AUTODIAL PHONE # ROUTINE FOR A HAYES SMARTMODEM
     ACCESSED BY PRESSING FUNCTION 1 KEY
     FUNCTION 5 KEY REPEATS LAST COMMAND SENT OUT

140 IFA$=CHR$(133)THEN400
160 IFA$=CHR$(135)THENGOSUB450
370 READNP:FORZ=1TONP:READPH$(Z):NEXTZ
400 PRINTC$;"PHONE LOG":PRINT
410 FORZ=1TONP:PRINTZ;PH$(Z):NEXTZ:PRINT
420 INPUT"SELECTION";PH
430 IFPH=0ORPH>NPTHENGOTO110
440 PRINTC$:TP$=CR$+"ATDT"+PH$(PH)+CR$
450 PRINT#2,TP$:GOTO100
999  REM THE FIRST NUMBER IN THE DATA STATEMENT IS  THE NUMBER  OF PHONE #
1000 DATA 5,5551234,6669876,3337654,4444321,8882345

     ROUTINE TO SEND YOUR OWN LOGON COMMANDS USING FUNCTION 3 KEY
     FUNCTION 5 KEY REPEATS LAST COMMAND SENT OUT

150 IFA$=CHR$(134)THENGOSUB500
160 IFA$=CHR$(135)THENGOSUB510
380 READNC:FORZ=1TONC:READCM$(Z):NEXT
500 CN=CN+1:TP$=CM$(CN)+CR$
510 PRINT#2,TP$
520 IFCM$(CN+1)="END"THENCN=0
530 RETURN
1997 REM THE 3 IN THE DATA STATEMENT IS THE NUMBER OF COMMANDS
1998 REM INCLUDING THE "END", WHICH MUST BE INCLUDED
1999 REM PUT THE COMMANDS USED IN QUOTES, AS DONE BELOW
2000 DATA 3,"SAMPLE STRING","SAMPLE STRING","SAMPLE STRING","END"
```

Given the data above, you could build yourself a suitable terminal program for your needs, and adjust it for any modem you might wish to use.

USING A PRINTER

The first thing needed to communicate with a printer is to use the OPEN command, as was done in the terminal program above. A serial printer should have the baud rate, word length, parity, and all other items set correctly, or it may not funtion correctly. Usually most newer printers have a series of switches which let you set the printer for all of these items, just as you can set the computer for them. What you need to do is set both the printer and computer the same way. Although you should check your printer manual first, most newer printers often don't use parity signals, and have parity disabled as shown in the Command Register chart.

PRINTER HANDSHAKING

In most cases the computer can send data to the printer much faster than the printer can print it. The printer lets the computer know when it is BUSY and when it is READY by a signal usually found on your printers pin 20. When the printer is READY this pin has a plus voltage

on it. When the printer is BUSY this pin has a minus voltage on it. This is called HANDSHAKING. These signals are supposed to go into the Deluxe RS232 Interface pins 5 & 6. Since it would be a hassle for you to switch the pins around, we have designed the switches to do this job for you. Just switch the two switches into the PRINTER positions and all the right connections are made.

Occasionally you will find serial printers which use BUSY and READY signals which are just the opposite as described above. When it is a minus voltage it was READY and when it was a plus voltage it was BUSY. Normally this would keep your printer from working at all. If this is the case with your printer, just switch the NORMAL switch on the Deluxe RS232 Interface to INVERTED. This inverts the control lines 5, 6, 8, and 20, allowing your printer to work correctly.

Some printers put the handshaking signal on the printers pin 4. If this is the case with your printer, then you need to make a modification to the RS232 Interface. On the bottom of the Deluxe RS232 Interface printed circuit board there are two sets of two squares right next to each other, each connected to the one right across from it. If you cut these connections and resolder them as below, it switches the signals on pins 4 & 20. This will handle printers which only handshake on pin 20.

PIN 20 HANDSHAKE

PIN 4 HANDSHAKE

ACTUALLY PRINTING SOMETHING

For BASIC programs, the way to print something out is by using the PRINT# command. First, open a channel to the printer using the OPEN statement described in the OPEN COMMAND section. The command:

PRINT#2,"THIS IS A LINE OF TEXT FOR THE PRINTER TO PRINT"

will print the data in quotes out on the printer. If you have a BASIC program which is designed to print out to the Commodore type printer (OPEN?,4,?), it can be converted to work with your serial printer by changing the OPEN statement which opens the printer line to one which suits your printer, leaving the first number (the Logical File number) of the OPEN statement the same. That leaves all the PRINT# commands in your program with the right number. Then your program will work fine.

Another way to print to the printer is by using the CMD command. Once you have opened a channel to Device number 2 (The USER I/O port), a CMD2 command routes all data that was destined to print on your screen to the device you have specified in the CMD command. For example, if you wanted to list a program to the printer, you would load the program into memory, and from READY, type in:

OPEN2,2,3,CHR$(3)+CHR$(1) (Used as an example. You use yours.)
CMD2
LIST

At that point, the program whould start listing out to the printer, just as it would have previously gone to the screen. When the printing is done, you need to type:

PRINT#2
CLOSE2

That closes off the channel. Until that point, everything will still go out to the printer. If, instead of typing LIST, you had typed PRINT"THIS IS A LINE", the printer would have printed THIS IS A LINE. This method could be used within a program for printing to the printer, but then you would not see the data being printed on the screen.

HOW IS A MODEM DIFFERENT THAN A SERIAL PRINTER?

Beside the obvious differences between a modem and a serial printer, the difference as far as hooking up the RS232 Interface is concerned is that some of the signals that come out of the RS232 cable have to go different places on a printer than they do on a modem. The switches on the Deluxe RS232 Interface are easily set for your printer or modem. For a Modem, set both switches on MODEM. For a printer, set both switches on PRINTER. In most all cases set the third switch on NORMAL. With these settings, most every modem and serial printer will work correctly. This section is designed to give you additional understanding of the RS232 connection, in case you are having difficulties. If you are not having any trouble with it, you can skip this section.

The main difference between a printer and a modem is that the signals on pins 2 and 3 are reversed on a printer. A modem requires that the transmit data line go to pin 2 and the receive data line go to pin 3. On a printer, it is just the reverse. Transmit data goes to pin 3 and receive data goes to pin 2. This will be explained further, but first, look at the following chart:

COMPUTER (DTE)		MODEM (DCE)	CHART#1
2 Transmit data to modem	---------->	2 Receive data from computer
3 Receive data from modem	<----------	3 Transmit data to computer
4 Request To Send to modem	---------->	4 Request to send from comp
5 Clear To Send from modem	<----------	5 Clear To Send to computer
6 Data Set Ready from modem	<----------	6 Data Set Ready to computer
7 Signal Ground	----------	7 Signal Ground
8 Carrier Detect from modem	<----------	8 Carrier Detect to computer
12 HI Speed Indicator from modem	<----------	12 HI Speed Indicator to cor
20 Data Terminal Ready to modem	---------->	20 Data Term Ready from comp
22 Ring Indicator from modem	<----------	22 Ring Indicator to computi

Some modems may not have all the above connections. The COMPUTER pin listing on the left would be called DTE which stands for "Data Terminal Equipment". The term DTE would generally apply to the device that you use, the part you type on. This is the standard signal arrangement for devices such as your computer. The Computer DTE pin chart is the pin configuration of the Deluxe RS232 Interface. The MODEM pin listing on the right would be DCE (Data Communications Equipment). You can see that the DTE computer plugs right into the Data Communication Equipment (DCE) modem, and all the right pins connect together. A computer is almost always set up to be the DTE. A modem is almost always set up to be DCE. They will plug right into each other and be correctly connected up. The reason a printer is different than a modem is that printers have evolved from teletypes. You can still find printers around that have keyboards on them. Since it is a "Data Terminal", a teletype would be Data Terminal Equipment also, or DTE. Since printers have evolved from teletypes, a printer is usually DTE. Looking at the Computer pin listing, imagine if you tried to connect two DTE computers together. It wouldn't work. For one thing, the Transmit Data line of one would be going right into the Transmit data line of the other. The switch setting on the Deluxe RS232 are designed to allow you to handle this kind of pin incompatibility. Look at the following chart, which shows the pin set up you should use when you are using a printer.

The main and most important difference bewteen DTE and DCE setups is that pins 2 and 3 (transmit data and receive data) are the opposite on a printer from the way they are on a modem. The switch labeled MODEM1/PRINTER1 handles this. Since you can't very easily rewire the connections on our cable, changing the switches can change what pins

certain signals can go in or out of. When switch 1 is in the MODEM position, data is transmitted from the computer on pin 2 and received into the computer on pin 3. When you switch it to PRINTER1, data is transmitted from the computer on pin 3 and received on pin 2. Look at the following chart, which shows the other main differences between DCE and DTE.

COMPUTER (DTE)		PRINTER (DTE) CHART #2
2 Receive data from printer	<----------	2 Transmit data to computer
3 Transmit data to printer	---------->	3 Receive data from computer
4 Carrier Detect from printer	<----------	4 Request to send to computer
5 Data Terminal Ready to printer	---------->	5 Clear to Send from computer
6 Data Terminal Ready to printer	---------->	6 Data Set Ready from computer
7 Signal Ground	----------	7 Signal Ground
8 Request to send to printer	---------->	8 Carrier Detect to computer
12 HI Speed not used	----------	12 HI Speed not used
20 Clear To Send/Data Set Ready to printer	<-	20 Data Term Ready to computer
22 RI not used	----------	22 RI not used

The printer pins have the same functions in the same pin positions as the computer does. They are both DTE. You don't have to know everything about the functions of these different pins, but a brief knowledge of their purpose will help you understand how to hook up your printer. Transmit Data (pin 2) sends outgoing data. Receive data (pin 3) receives incoming data. Data Terminal Ready (pin 20) tells the computer that the printer is ready to receive data. Clear to Send and Data Set Ready on the computer (pins 5&6) are the pins that the printer Data Terminal Ready goes to. On the Deluxe RS232 Interface, Clear to Send and Data Set Ready are hooked together. They both do exactly the same function, and so this causes no difficulty. If you look over the chart above, you can see how these connections need to be swapped around a bit when hooking up a printer. Since you can not easily pull the pins out of our RS232 cable and swap them around, the switches in the RS232 cover allow you to change what pins on the RS232 connector these particular signals come out of. This allows you to set the Deluxe RS232 Interface up so that it is compatible with either a printer of a modem.

MODEM1/PRINTER1 - When this switch is in the MODEM1 position, pin 2 is Transmit data, pin 3 is Receive data, pin 4 is Request to Send, and pin 8 is Carrier Detect. When the switch is in the PRINTER1 position, it swaps pins 2 & 3, and pins 4 & 8.

MODEM2/PRINTER2 - When this switch is in the MODEM1 position, pin 5 is Clear to Send, Pin 6 is Data Set Ready, and pin 20 is Data Terminal Ready. When this switch is in the PRINTER2 position, pins 5 & 6 both are swapped to pin 20, and vise versa.

NORMAL/INVERTED - This switch affects control pinsd 4, 5, 6, 8, 12, 20, and 22. When this switch is in the NORMAL position, all these pins are set to act the same as a Commodore 1011A. When the signal on the USER I/O port connection is HI, the signal on the corresponding RS232 pin will be HI. When this switch is in the INVERTED position all of these control pins are inverted. When the signal on the USER I/O port pin is HI, the corresponding RS232 pin will be LOW.

TRANSFERING BASIC PROGRAMS

Several persons had asked us about how to transfer BASIC programs onto their Commodore from another computer which was not a Commodore. This can be done fairly easily. The following method assumes that you have a terminal program on the other computer which can load up a BASIC program as an ASCII file (not tokens) in it's buffer, and send it out the RS232 port. If you can do this with the other computer, then this is what you have to do on the Commodore end.

The following BASIC program should be typed into your Commodore and run just before sending over the BASIC program from the other computer. This program below gets each character of the BASIC program as it is sent over the line, prints it on the screen, then the pokes put carriage returns in the keyboard buffer, causing the commands printed on the screen in line 4 to be executed, and the transfered BASIC program line to be entered into the computer as if you were typing in a basic program at 300 baud. The BASIC program being transfered must be in ASCII, not tokens, and the line numbers of the BASIC program must not start below 7.

```
0 OPEN2,2,0,CHR$(6):PRINTCHR$(147)
1 GET#2,A$:IFVAL(A$)=0THEN1
2 PRINTA$;
3 GET#2,A$:PRINTA$;:IFA$<>CHR$(13)THEN3
4 PRINT:PRINT"POKE152,1:GOTO6"
5 POKE631,19:POKE632,13:POKE633,13,POKE634,13:POKE635,13:POKE198,5:END
6 PRINTCHR$(147);:GOTO3
```

OTHER ACCESSORIES

Omnitronix offers a Complete Serial Printer Driver for use with the VIC-20 and C64. This is a program which loads into the computer before using other software or printing to the printer. It intercepts the data being sent out to the printer and handles it in various ways. A compact driver is supplied which allows you to use most application software, even if that software does not allow you to select using a serial printer. A Complete Graphics driver is supplied which allows the your printer to work like a Commodore type printer. The Graphics Driver allows you to print PET graphics and control code data to the printer. You can select to have graphics and control codes printed in brackets i.e. (BLK), (CLR), (198). If you have a serial printer which can do bit map graphics, the software can be configured to print actual graphics characters. A CBM Emulation mode is available which allows you to used Commodore printer type commands such tabbing or switching character sets. The driver corrects a difficulty with the Commodore ROM software which sometimes leaves the last bit of data you send out unprinted. The Complete Serial Printer Driver contains both VIC-20 and C64 drivers, and is available on disk only. The price is $29.95.

Omnitronix has also introduced a Serial Bus RS232 Printer Interface called the Printmaster/S. This interface will allow you to use all your programs which use a printer with your serial printer. Full graphics capability is supported on popular graphics printers such as Epson, Okidata, Nec, etc. Full RS232 voltage levels are provided. You can select 300, 600, 1200, or 2400 baud, 7 or 8 bit word, and parity. Catalog # 10-117 $119.95

NAME

ADDRESS

CTY/ST/ZIP

Place
Stamp
Here

OMNITRONIX, INC.
PO BOX 43
MERCER IS WA 98040

INSTRUCTION MANUAL
DELUXE RS232 INTERFACE

V2.a

Omnitronix, Inc.
PO Box 43
Mercer Is., WA 98040
(206) 236-2983