Checking Comm Ports

The CPP has a built in test feature for testing the RS-232 ports of the CPP. This is an off line test. Once entered it is exited by either entering an Escape or cycling power.

Comm Port #1
The operating Baud rate and the site ID setting can be determined from the L keyin over comm port #4. Immediately below the comm port #1 Baud rate setting, the CPP prints the binary setting of dip switch #1which controls the Baud rates for comm port #1 and #4. Printed below are the settings of switch 1. These switch settings are defined below.

Table 2.1

Switch Position										Function	
1	2	3	4	5	6	7	8	9	10		
0	0	0	x	x	x	x	x	x	x	300 Baud	Comm Port #4 Baud rates
1	0	0	x	x	x	x	x	x	x	1200 Baud	
0	1	0	x	x	x	x	x	x	x	2400 Baud	
1	1	0	x	x	x	x	x	x	x	4800 Baud	
0	0	1	x	x	x*	x	x	x	x	9600 Baud	
1	0	1	x	x	x	x	x	x	x	19.2K Baud	
0	1	1	x	x	x	x	x	x	x	38.4K Baud	
1	1	1	x	x	x	x	x	x	x	38.4K Baud	
1	2	3	4	5	6	7	8	9	10		
x	x	x	0	0	0	x	x	x	x	300 Baud	Comm Port #1 Baud rates
x	x	x	1	0	0	x	x	x	x	1200 Baud	
x	x	x	0	1	0	x	x	x	x	2400 Baud	
x	x	x	1	1	0	x	x	x	x	4800 Baud	
x	x	x	0	0	1*	x	x	x	x	9600 Baud	
x	x	x	1	0	1	x	x	x	x	19.2K Baud	
x	x	x	0	1	1	x	x	x	x	38.4K Baud	
x	x	x	1	1	1	x	x	x	x	38.4K Baud	

Positions number 1, 2, and 3 of switch #1 are decoded to select the baud rate for Comm Port #4 or on site port. Factory default is 9600 baud. Switch positions 4, 5, and 6, are used to set the Baud rate of comm port #1. The factory default for both Baud rates is 9600 Baud.

Comm Port Test
The built in test feature uses comm port #2 to send and receive a string to a selected comm port. A string is sent out of comm port #2 at 9600 Baud, and the receiving comm port returns the string at 4800 Baud. If all operates properly, the CPP prints an OK. Otherwise, and error message is printed.

Comm port 3-0 is on the CPP processor board and comm ports 3-1 through 3-8 are on the expanded comm port board. Proper operation of comm port #2 can be determined by testing comm port number 3-0, and a comm port on the expansion board. It is highly unlikely that both boards will experience a failure. The integrity of the cable used for testing can be verified by using it to communicate with comm port #4.

To test the comm ports, over comm port #4 enter JC<cr>. This takes the CPP off line and the CPP prints the message;

```
Connect 3-x to C-2, Enter x
```

Connect a standard telephone cable between comm port #2 and the comm port to be tested, *e.g.* comm port 3-0. Enter a 0<cr> to start the test. A proper response is presented below.

```
AaBbCcDdEeFfGgHhJiKk
AaBbCcDdEeFfGgHhJjKk
AaBbCcDdEeFfGgHhJjKk     OK
Connect 3-x to C-2, Enter x
```

An improper response indicating that the selected comm port did not receive the string is presented below.

```
AaBbCcDdEeFfGgHhJjKk
No response 3-0
Connect 3-x to C-2, Enter x
```

This could be because the receiving comm port hardware did not get the string or because comm port #2 transmitting hardware did not send the string. The proper operation of comm port #2 and the test cable can be verified as discussed above.

An improper response indicating that the selected comm port received the string and returned a response is presented below.

```
AaBbCcDdEeFfGgHhJjKk
AaBbCcDdEeFfGgHhJjKk
No response C-2
Connect 3-x to C-2, Enter x
```

As before, until the proper operation of comm port #2 is verified, this could be a hardware fault in either comm port #2 receiving circuitry or the comm port under test transmitting circuitry.

An improper response is presented below. This is just an example. Comm port #2 checks the returned string against the transmitted string, and prints and error if the two are different.

```
AaBbCcDdEeFfGgHhJjKk
AaBbCcDdEeFfGgHhJjKk
AABBCCDDEEFFGGHHJJKK  Not OK
Connect 3-x to C-2, Enetr x
```

The x can be 0 through 8 to test the 3-x comm ports, and C1 to test comm port #1.

When testing is complete, enter an Escape and the CPP will restart.

Comm Port #1 Additional Testing

One problem sometimes encountered when connecting to comm port #1 is that the computer is set up to send an automatic carriage return/line feed. This will not work properly. If problems are encountered with comm port #1 this additional testing can provide some information.

Comm port #1 is set up to use a dollar sign ID to bring the comm port on line, e.g., $1. Until this is received properly, comm port #1 will not come on line. Over comm port #4 enter an M<cr>. The CPP will print a number of memory addresses. One of these will be RX1B = XXXX, where XXXX is the memory address of the receive buffer for comm port #1. To look at this memory, subtract two from the XXXX and enter MXXXX<cr>. The CPP will print out two lines of hexadecimal data. The first two characters are the receive buffer address pointer, which points to where in the receive buffer the next received byte will be written.

As an example, XXXX = 8BDF. Enter M8BDD<cr>. The CPP responds with two lines, a partial example of which is presented below.

```
M8BDD
8B DF 24 00 00 …
```

The first two hexadecimal characters (RX1B buffer pointer) point to address 8BDF.

Over comm port #1, enter a $. On the front panel display, a 1r should come up on the top line of the display. If it does not, then the CPP did not receive the $ character. If it did flash 1r, then over comm port #4 enter MR<cr>. The CPP prints out the above memory locations. It should be 8BDF 24. The 24H is the dollar sign. If it is not a 24H, then the CPP did receive the $ properly. Either the Baud rate is wrong, the wiring is wrong, or more characters are being received, such as $crlf.

If the 24H is correct, then over comm port #1 enter the ID, 1. The front panel display should flash 1rt, indicating that it received the $ followed by the ID properly and echoed the ID.