

## Special Protocols

### General

An option available allows the CPP to provide polled data transmissions corresponding to other protocols. This is set up from the Main Menu. From the Main Menu select Special Protocols. The CPP responds with the following questions.

- 1.) None(0), Polled (1), Broadcast (2) = N **1**<cr>
- 2.) Address (00-99) =00 <cr>
- 3.) Comm Port = 2 <cr>
- 4.) Baud Rate  
1200(2) 2400(3) 4800(4) 9600(5) 19.2K(6) 38.4K(7) = 0 **5**<cr>
- 5.) # Data Bits, 8-7 = 0 **7**<cr>
- 6.) Parity; 0=None, 1=Even, 2=odd = **1** <cr>

Ret to Main Menu

**Line 1** asks if the data transmission is to be automatically broadcast or if it is to respond to a command poll. Only the polled feature is presently enabled.

**Line 2** asks the address this CPP is to respond to.

**Line 3** asks which comm port on the CPP is to be used as the broadcast port. This question is only provided for future applications. At present the CPP accepts only comm port number 2 for the broadcast port.

**Line 4** asks the Baud rate for the broadcast transmission.

**Line 5** asks if the data bytes consist of 7 or 8 data bits.

**Line 6** asks for the type of parity to be transmitted

### Format

The CPP saves the data value and the data value status in memory arrays. The special protocol mode only outputs the data value. If the data value has a bad status indication, the CPP sends -99.99 as the data value. Data is transmitted as percent of full scale entered into the CPP.

## **Mexico City Protocol**

The Protocol for Mexico City consists of sending a sign byte, five data digits and a decimal point for each data value. The data values are combined into the existing string protocol. An example is given below for the first eight channels. Data is returned in engineering units. A bad data point is sent as -9999.0.

```
*00:+50.000/+500.10/+50.020/+500.30/+50.040/+500.50/+50.060/-9999.0:CC<crLf>
```

### **Commands**

The CPP will respond to three commands presently available in the existing network. These are defined below. Either a carriage return, or a carriage return/line feed pair can sent with the commands. If check characters are sent they are checked for corresponding to the received command string. If check characters are not sent, the CPP evaluated the received command based on the string received.

### **Return Data**

The return data command is as follows;

```
*II:SCA/SS/EE:CC<cr>
```

Where:

- II = Station ID
- SCA = Command
- SS = Channel starting number – 00 = chan #1
- EE = Channel ending number
- CC = Checksum

The format of the returned data is presented below. The command requests that the CPP return data from channels 1 – 8 inclusive.

Command from central

```
*00:SCA/0/8:9B<cr>
```

CPP response

```
*00:+50.000/+500.10/+50.020/+500.30/+50.040/+500.50/+50.060/-9999.0:CC<crLf>
```

### **Set Time**

The central can download the time to the remote CPP units. The command string is as follows;

```
*II:TMPO/HH/MM/SS:CC<crLf>
```

Where:

- II = Station ID
- TMPO = Command
- HH = Hours
- MM = Minutes

SS = Seconds

Command from central  
\*01:TMPO/23/30/00:<cr>

CPP response  
\*01:TIEMPO OK :

If the detects an improper time value, or cannot interpret the command, no response is returned.

### **Return Status**

The CPP computes a status byte for each received command. The status byte is the status associated with the last command received. After responding to the STA command, the CPP clears the status byte. The response for this command deviates from the other responses in that the response does not start with an asterisk and then the address. Instead it starts with an asterisk and then sends the status byte. The command to retrieve the status byte is as follows;

Command from central  
\*10:STA:<cr>

Remote response  
\*HH::CC<cr>

The HH is a hexadecimal representation of the status byte. The byte is defined as follows.

- HH = 00 = No activity
- 01 = No starting asterisk
- 02 = No Colon
- 03 = Not our address
- 04 =
- 05 = Not a command we use
- 06 = Error in command context
- 07 = Check character error
- 08 = Starting channel > ending channel
- 09 = Return data OK
- 10 = Set time accepted
- 11 =