External I/O Of CPP

1.0 General

The CPP has provisions for inputting and outputting analog and digital signals to an external unit, such as a DigiMux, DigIO, Dutec, OptoMux, or TSX Momentum. The CPP is set up to send and receive external signals over the serial ports. The CPP can be configured to use external analog and digital I/O over the same comm port, or it can be configured to use separate comm ports to use the external I/O devices.

This information can also be simultaneously output in a Modbus protocol over another comm port. The Modbus capability is presented in Tech Note TN33. The CPP external I/O is set up from the Main Menu. From the Main Menu select External I/O. The CPP responds with the following question/answer narrative.

2.0 Analog Outputs

This section describes setting up the CPP to output analog values to an external device.

1. External I/O
2. Analog Out(AO), In (AI) - Digital Out(DO), In(DI) = **AO<cr>**

3. Analog Output Setup
4. DAC(D), or Modbus(M) = **D<cr>** (For Modbus, see Tech Note TN33)
5. DigiMux (D), Dutec(U), Opto(O), TSX(T) = D **<cr>**
6. DAC Address = **080<cr>**
7. DAC Comm Port # 3-08**<cr>**
8. DAC Baud Rate 1=300, 2=1200, 3=2400, 4=4800, 5=9600, 6=19.2K(6) = 2 **5<cr>**
9. Analog Output # = **01<cr>**
10. CPP Chan # = **3<cr>**
11. Volt(V) or Current(I) = V = **<cr>**
12. Hold On Cals Y/N Y = **<cr>**
13. Hold On Bad Y/N Y = **N<cr>**
14. Enable Y/N N = **Y<cr>**

15. Analog Output # = **<cr>**

    Return To line 2

Line 1 is a heading.

Line 2 is asking for the type, either analog I/O or digital I/O.

Line 3 indicates that the user selected the analog output set up.

Line 4 is asked only if the Modbus protocol driver is resident in the firmware.

Line 5 is asking for the Manufacturer of the DAC.

Line 6 is asking for the module address of the DAC.

Line 7 is asking which comm port on the CPP is connected to the analog output module. Inputting a comm port of N indicates to the CPP, that actually there is not a physical DAC connected. This is used to configure the CPP to store data that can be transmitted in a Modbus format. The actual analog outputs are not needed.

Line 8 asks at what baud rate the comm port is to operate.

Line 9 starts the setup of the analog outputs, and asks which analog output is to be configured. Analog output #1 in the CPP corresponds to analog output #0 in Dutec and Optomux nomenclature and analog output #16 in the CPP corresponds to analog output #15 in Dutec and Optomux nomenclature.

Line 10 asks which channel on the CPP is to be output to this analog conversion output.

Line 11 asks for the type of DAC output, either voltage or current. A selection of current assumes that it is a 4-20mA signal.

Line 12 asks if the CPP is to hold the last value it has before a calibration flag is detected on this CPP channel.

Line 13 asks a similar question regarding the detection of bad data status flags.

Line 14 enables this analog conversion output.

The analog outputs are updated every fifteen seconds. Even though data is being sent to the analog output module every fifteen seconds, some values, such as computed channels, and data being received over serial ports may be updated at a slower rate. The analog output number corresponds one to one with the channel designations in the analog output module. (Both Dutec and OptoMux number their outputs as 0 - 15. The CPP designates these numbers as 1 - 16). In the above example, data from CPP channel # 03 would be output over analog output #01. The serial strings will be output over comm port #7 in the CPP.

If either the Hold On Cals or Hold On Bad feature is selected, the CPP will hold the last value output prior to detecting either a calibration flag or bad data flags.

3.0 Analog Inputs

This feature allows the CPP to receive and store digital data from an external analog to digital converter.

3) Analog Input Setup
4) DigiMux (D), Dutec(U), Opto(O), TSX(T) = D **<cr>**
5) ADC Address = **080<cr>**
6) ADC Comm Port # 3-08**<cr>**
7) ADC Baud Rate 1=300, 2=1200, 3=2400, 4=4800, 5=9600, 6=19.2K(6) = 2 **5<cr>**
8) Analog Input # = **01<cr>**
9) CPP Chan # = **3<cr>**
10) Volt(V) or Current(I) = V = **<cr>**

11) Analog Input # = **<cr>**

Return to line 2

Line 4 is asking for the Manufacturer of the external ADC.

Line 5 is asking for the module address of the external ADC.

Line 6 asks for the CPP comm port #.

Line 7 asks at what baud rate the comm port is to operate.

Line 8 starts the setup of the analog inputs, and asks which analog input is to be retrieved. Analog input #1 in the external device corresponds to analog input #0 in Dutec and Optomux nomenclature and analog input #16 in the CPP corresponds to analog input #15 in Dutec and Optomux nomenclature.

Line 9 asks which channel to store the retrieved data.

Line 10 asks for the type of ADC input, either voltage or current. A selection of current assumes that it is a 4-20mA signal.

4.0 Digital Outputs

In line two above inputting DO selects setting up external digital output bits. The internal output bits are mapped into the external output bit register.

3) Digital Output Setup
4) DigiMux(D), Dutec(U), Opto(O), TSX(T) = D **<cr>**
5) DO Address = **040 <cr>**
6) DO Comm Port # 3-08 **<cr>**
7) DO Baud Rate 1=300, 2=1200, 3=2400, 4=4800, 5=9600, 6=19.2K(6) = 5 **<cr>**
8) CPP Starting Bit = **31 <cr>**
9) Number bits = **10 <cr>**
10) Enable Y/N = **Y<cr>**

Return to line 2

Lines 4-7 are the same as above.

Line 8 asks for the starting bit in the CPP output register.

Line 9 asks for the number of bits in the CPP output register to send to the external device. Bits 25 through 34 of the CPP output register will be sent to the external device as bits 1 through 10. In Dutec and Opto this is bits 0 through 9.

5.0 Digital Inputs

In line two above inputting DI selects setting up external digital input bits. The CPP internal input bits are mapped into the external input bit register.

3) Digital Input Setup
4) DigiMux(D), Dutec(U), Opto(O), TSX(T) = D **<cr>**
5) DI Address = **040 <cr>**
6) DI Comm Port # 3-08 **<cr>**
7) DI Baud Rate 1=300, 2=1200, 3=2400, 4=4800, 5=9600, 6=19.2K(6) = 5 **<cr>**
8) CPP Starting Bit = **31 <cr>**
9) Number bits = **10 <cr>**
10) Enable Y/N = **Y<cr>**

Return to line 2

Lines 4-7 are the same as above.

Line 8 asks for the starting bit in the CPP input register.

Line 9 asks for the number of bits to accept to the external device. Bits 1 through 10 will be accepted from the external device, and stored in the CPP input register as bits 25 through 35. In Dutec and Opto this is bits 0 through 9.

6.0 Setting DAC output levels

A user can override the channel values and force the DAC outputs to certain levels. This is useful for testing or calibrating the analog output levels. This is done by entering the following.

SAO01<cr>
Set Value (%) = **10<cr>**
>

The first input SAO01 selects analog output number one. The CPP then asks for the level to set the output in percent of full scale. In the example the user entered 10%. This could have been input in one string as SAO 01 10<cr>, with or without the spaces. The CPP will output this level for 3 minutes and then revert back to outputting the channel values. The CPP can be instructed to return to outputting the channel values by inputting a value of 111%, e.g. SAO01 111<cr>.

7.0 Copy Outputs To Comm Port #4

The CPP can also be commanded to echo the I/O strings to comm port #4. This is a toggle on, toggle off feature.

7.1 Toggle analog outputs

```
SAOP<cr>
AO On

SAOP<cr>
AO Off
```

7.2 Toggle analog inputs

```
SAIP<cr>
AI On

SAIP<cr>
AI Off
```

7.3 Toggle digital outputs

```
SDOP<cr>
DO On

SDOP<cr>
DO Off
```

7.4 Toggle digital inputs

```
SDIP<cr>
DI On

SDIP<cr>
DI Off
```