T3 Modules

User's Manual
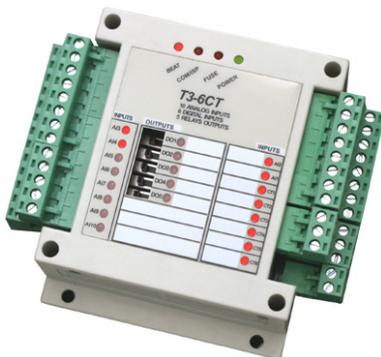
T3-8AI8AO: 8 Universal Input
8 analog output

T3-4AO: 10 Universal Inputs
4 Analog Outputs
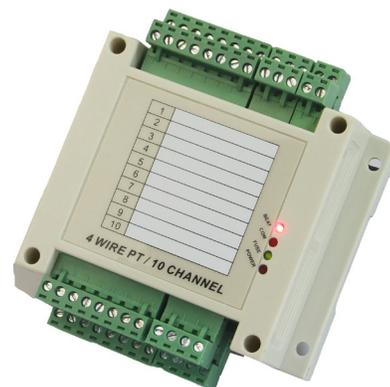8 dry-contact relay outputs

T3-32I: 32 Universal Inputs

T3-8I13O: 8 Universal Inputs
13 relay outputs

T3-6CT: 10 Universal Inputs
6 AC Current Inputs (6 CTs)
5 Relay Outputs

T3-PT10: 10 Universal PT100 Inputs

1 Introduction

1.1 About this manual

   The purpose of this manual is to provide instructions to simply and quickly install and operate the T3 Module equipment. The manual begins with a general description of the product followed by the instructions for correct hardware installation. Its configuration and operation of the device are described in detail later in the manual.

1.2 General Description

   The T3 Series are general purpose input / ouput modules for building integrators. Available in several input/output configurations, the T3 Series modules provide convenient termination for field devices and interfacing to your HVAC, lighting, temperature sensors, and other typical building automation applications. Each of the analog inputs can be jumper configured for signals of either 0-5V, 0-20mA, or dry contact. The outputs are available in dry contacts 1amp/output, 0-10V analog, and PNP sinking. The modules are slave devices that can be easily controlled via the RS485 serial interface using the industry standard Modbus Protocol.

   Highlights:
   • Surge-Protected Analog Inputs with 10-bit Resolution
   • Outputs can Individually be switched to ON, OFF, AUTO
   • High Impact Plastic Enclosure provides durability in commercial environments
   • Standard Modbus Protocol allows for up to 254 unique devices on one RS485 Network

1.3 Technical Data

T3-8AIAO	8 Analog Outputs @ 0-10VDC 200mA total, 8 Analog Inputs @ 0-5V, 0-20mA, Dry
T3-32I	32 Analog Inputs @0-5V, 0-10V, 0-20mA, Dry
T3-8I13O	13 Dry-contact relay outputs x 2amps @120V, 8 analog inputs @0-5V, 0-20mA, dry
T3-4AO	8 dry-contact relay outputs x2amps @120V, 4 analog outputs @0-10V, 10 analog input @ 0-5V, 0-10V, 0-20mA, dry
T3-6CT	T3 series, general purpose 10 Analog Inputs, 6 Digital Inputs, 5 Relay Outputs, RS485 network
T3-PT10	T3 series, 10 Universal PT100 Input module, with 2-3 or 4 wire connections, RS485 network
Operating Temperature	-30~70°C (-22~158°F)
Supply Voltage	12~24VAC/DC ±10%, 50-60Hz
Power Consumption	100mA at 12VDC
Relay Contacts Rating	Max 2A
Ambient Humidity	10-90 %Rh
Plastic Housing	Flammability Rating UL 94HB
Enclosure Rating	IP31
Temperature Sensor	10K Thermistor ±0.5°C
Color	White/Off-White

T3 Model	Universal Inputs		AO	DO
--	0-5V 0-20mA	0-5V 0-10V 0-20mA	--	--
T3-8AI8AO	8	--	8	--
T3-32I	--	32	--	--
T3-8I13O	8	--	--	13 Relay 120VAC, 2A
T3-4AO	--	10	4	13 Relay 120VAC, 2A
T3-6CT	10 Universal Inputs, 6 AC Current Inputs (6CTs), 5 Relay Outputs			
T3-PT10	10 x PT100 inputs, 2-3-4 wire			

Dimensions: 115(L) x 90(W) x 40mm(H)
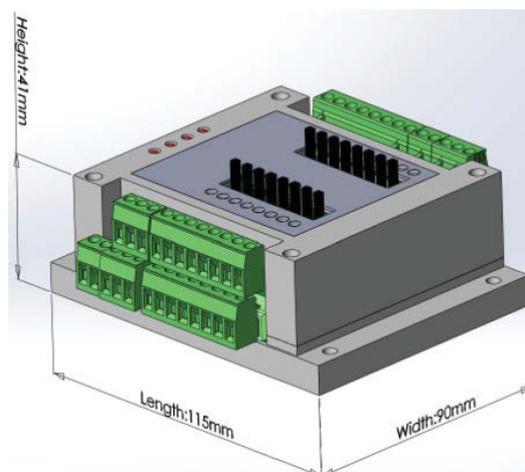
T3-4AO

Description

The T3 Series are general purpose input / output modules for building integrators. Available in several input/output configurations, the T3 Series modules provide convenient termination for field devices and interfacing to your HVAC, lighting, temperature sensors, and other typical building automation applications. Each of the analog inputs can be jumper configured for signals of either 0-5V, 0-20mA, or dry contact.The outputs are available in dry contacts 2amp/output, 0-10V analog, and PNP sinking. The modules are slave devices that can be easily controlled via the RS485 serial inter face using the industry standard Modbus Protocol.

Highlights:
- Surge-Protected Analog Inputs with 10-bit resolution.
- Outputs can individually be switched to ON, OFF or AUTO.
- High Impact Plastic Enclosure Provides durability in commercial environments.
- Standard modbus protocol allows for up to 254 unique devices on one RS485 network.

Special Features

The T3-4AO has a few additional features which the other T3 series do not have due to a more advanced CPU. For example the faster scan rates for the inputs In on/off mode, inputs 3 through 10 can count pulses up to 1 khz on each channel.
In analog mode, inputs 3thru 10 are 12 bits compared to the previous 12 bits, inputs 1 and 2 remain as 8 bits and slower at pulse counting.
For developers there is a significant improvement with the larger rom and ram space: 128k versus 64k for the flash space and 3k ram versus 1k of ram space compared to the earlier models. This gives more room for developers to add features such as Bacnet, PLC type logic, logging, etc. Secondly there's an additional serial port, currently the port is unused but developers will be able to use the second port to manage a subnet of local sensors, keypads and displays for example, or use it in repeater mode to extend and isolate the RS485 main network.

Technical Data

| T3-4AO	8 Dry-Contact Relay Outputs x 2amps @120V, 4 Analog Outputs @0-10V, 10 Analog Input @ 0-5V, 0-10V, 0-20mA, Dry
Operating Temperature	-30~70°C (-22~158°F)
Supply Voltage	12~24VAC/DC ±10%, 50-60Hz
Power Consumption	100mA at 12~24VAC/DC
Relay Contacts Rating	Max 2A
Ambient Humidity	10-90 %Rh
Plastic Housing	Flammability Rating UL 94HB
Enclosure Rating	IP31
Temperature Sensor	10K Thermistor ±0.5°C
Color	White/Off-white
Dimensions	115 x 90 x 40mm

PT-10

Description

The PT-10 is a low-cost, high-accuracy digital thermometer read out designed with 10 Univeral PT100 Inputs. Its combination of features makes this instrument well suited for a wide variety of industrial applications.

Highlights:

• 10 Universal PT 100 Inputs

• Serial RS 485 Standard

• Light-weight and Compact

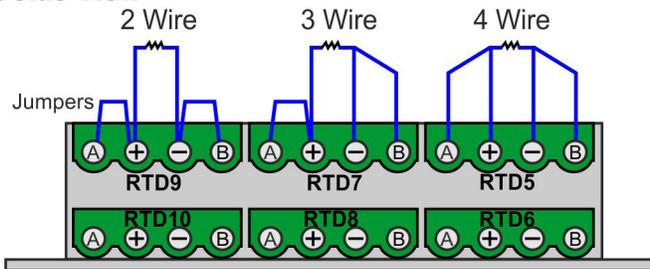• Usable with 2, 3 or 4 Wire connections
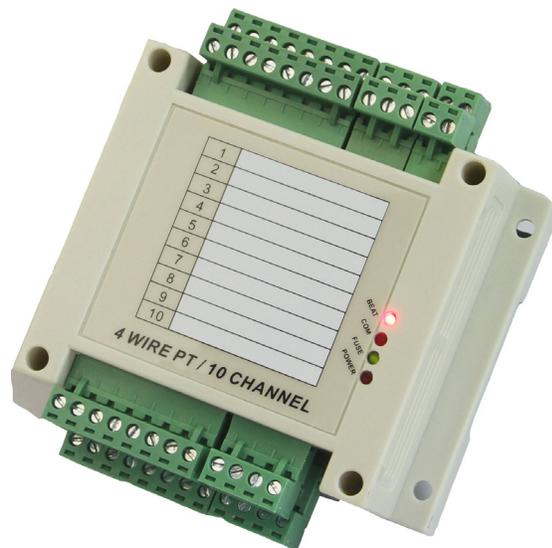
Technical Data

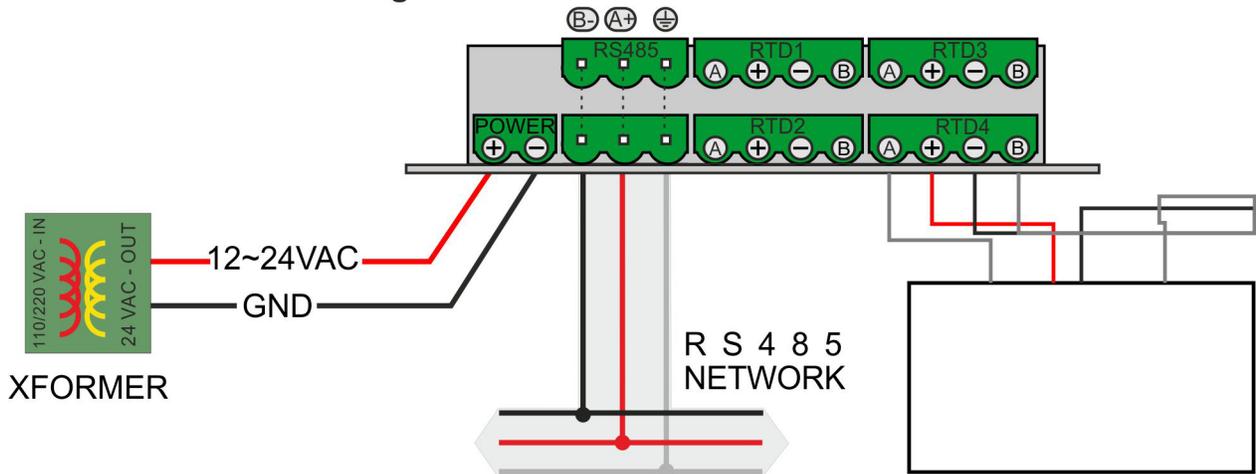|	
Resistance Range	0Ω to 400Ω, auto-ranging
Resistance Accuracy, One year	0Ω to 20Ω: 0.01Ω 20Ω to 400Ω: 0.05% (15 ppm) of reading
Resistance Accuracy, Short-Term	0Ω to 30Ω: 0.005Ω 30Ω to 400Ω: 0.03% (15 ppm) of reading
Resistance Resolution	Adjustable to .001Ω (See Register List)
Temperature Range	-200°C to 962°C (-328°F to 1764°F)
Temperature Coefficient of Resistance	10 ppm/°C
Temperature Accuracy	0.01°C
Temperature Resolution	Adjustable to .001°C (See Register List)
Probe Connection	4-wire with shield, 5-pin DIN connector (2 & 3 - Wire Connection also possible)
Maximum Acceptable Lead Resistance	100Ω
Probe Excitation Current	PRT 1 mA
Communications	RS-485 Serial Standard
AC Power	12~24 VAC/DC ±10, 50/60 Hz, 10 W, Nominal, 1A Maximum

Environmental Conditions

Ambient Temperature Range	0-55°C (32-131°F); [Full Accuracy 16-30°C (61-86°F)]
Ambient Relative Humidity	Maximum 80% for Temperature < 31°C, Decreasing Linearly to 50% at 40°C
Pressure	75kPa - 106kPa
Main Voltage within ±10% of Nominal	
Vibrations Should be Minimized	
Altitude Less than 2,000 Meters	
Indoor Use Only	

Wiring Diagram

Left Side View

2 Wire   3 Wire   4 Wire

Jumpers

A (+) (−) B   A (+) (−) B   A (+) (−) B
RTD9   **RTD7**   **RTD5**

RTD10   **RTD8**   **RTD6**
A (+) (−) B   A (+) (−) B   A (+) (−) B

Right Side View

B- A+ ⏚
RS485   RTD1   RTD3
A (+) (−) B   A (+) (−) B

POWER   RTD2   RTD4
(+) (−)   A (+) (−) B   A (+) (−) B

110/220 VAC - IN   24 VAC - OUT

12~24VAC

GND

XFORMER

R S 4 8 5
NETWORK

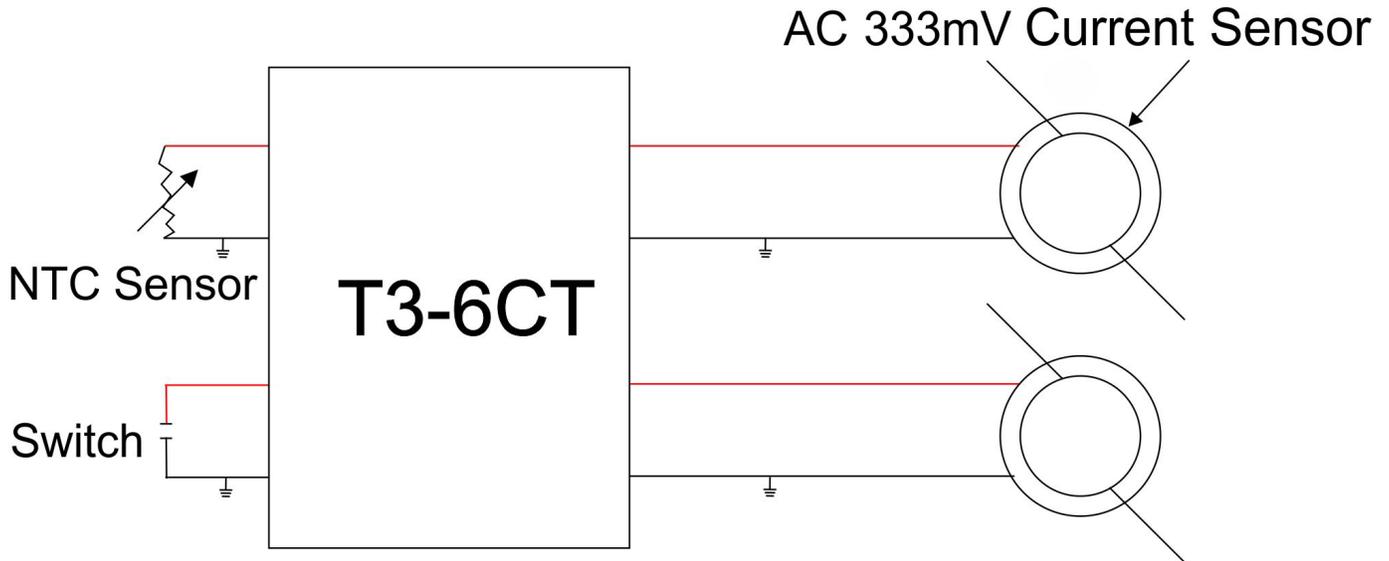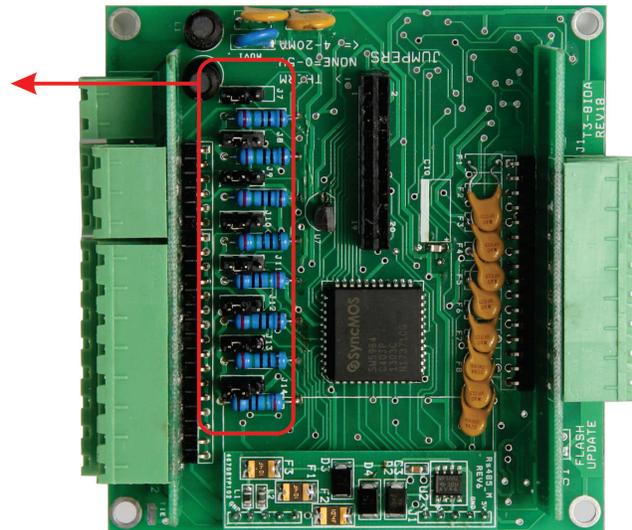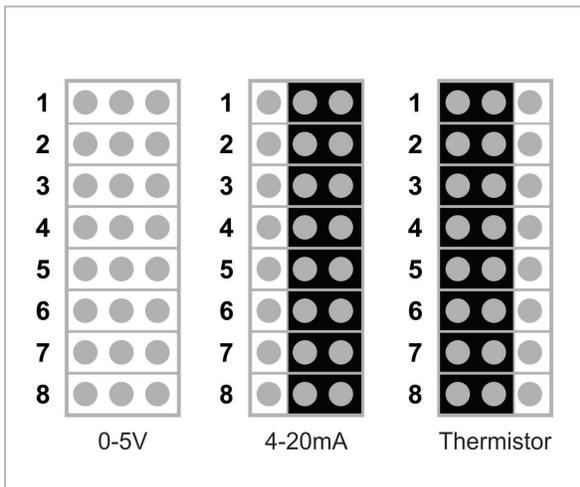**T3-PT10 Terminal Block Diagram**

T3-CT connect with current sensor

T3-8IO Jumper Settings

There are 8 inputs, each input has a jumper, all you need to do is to set the single jumper to the appropriate signal type: 0-5V, 4-20mA, thermistor as below the picture displayed.

1.5 Standard Operation

1.5.1 Inputs

Each input of a T3 Module can be jumper-configured in 1 of 3 ways:

- •0-5V Signal
- •0-20mA Signal
- •Dry Contact, Thermistor

The value of each input is stored as a 10-bit number in the respective modbus register. The registers addresses are as follows:

T3 Model	Number of Inputs	Register Addresses
T3-8AIAO	8	108-115
T3-32I	32	100-131
T3-8I13O	8	118-133
T3-4AO	10	190-199

Table1: Input Register Addresses

A 5V, or 20mA, would give a reading of 1024. Each input has a corresponding LED which will light up if the value of the input is greater than 512. For more info on reading the input registers, see the section on Serial Communications.

T3-4AO is 5V, or 20mA, and would give a reading of 4095. Each input has a corresponding LED which will light up if the value of the input is greater than 2047. For more info on reading the input registers, see the section on Serial Communications.

1.5.2 Outputs

The state of each output is determined by its corresponding switch position. The switches have 3 states – 'hand', 'off', and 'auto'. When switched to 'hand', the corresponding output will be switched on - 10V for analog, contacts closed off from the relay, or 0V for sinking outputs. When switched to 'off', the output will be set to 0V for analog, open contact for relay, or open circuit for sinking outputs. When switched to 'auto', Analog outputs will be set to the level stored in the corresponding MODBUS output registers. For Digital or Sinking outputs, a register value 0 is deactivate and register value 1000 is activated.

The output registers are as follows:

T3 Model	Number of Outputs	Register Addresses
T3-8AIAO	8	100-107
T3-32I	0	--
T3-8I13O	13	100-112
T3-4AO	12	100-111

Table 2: Output Register Addresses

These registers can be changed using the RS485 serial interface. For analog outputs, a 0 corresponds to 0V. Likewise, a 1024 corresponds to 10V. For relay or sinking outputs, the output will be activated by any number greater than 512. The output registers are stored in RAM, thus the contents of each register will be lost upon power-off. Each output has a corresponding LED which will light up if the value of the output is greater than 512 (5V). For more info on writing the output registers, see the section on Serial Communications.

1.5.3 Analog Output Calibration

The T3-8IO-A has an output calibration feature that allows for an adjustment of +/- 1.28V. Calibration is controlled via the calibration register located at registery address 13. By default, this is 128, which corresponds to 0V calibration. A value of 0 would give a –1.28V offset. A value of 255 would give a +1.28V offset. It is recommended that the calibration be determined while the output is set to 5V. The calibration value is located in flash memory and will be restored upon power-up.

1.5.4 Baudrate

All T3-Modules have adjustable Baudrates set by MODBUS registery 15. By default baud is set to 19.2kbps.
Value 1 will set the baud to 19200 bps.
Value 0 will set the baud to   9600 bps.

1.6 Accessing T3 Series Registers Via Serial Communications

The T3-Modules have a built-in serial interface for communication over an RS485 Network. Communication is currently implemented using Modbus Protocol. However, future versions of the T3-Modules will work with both BACnet and TCP/IP Protocols. For detailed information on Modbus Protocol, see the chapter entitled Modbus Serial Communications.

1.6.1 Connecting the T3 module to a computer

The T3-Modules connect to a computer serial port via the RS485 interface. Figure 14 shows how the T3-Module should be connected to the serial port of a PC.

T3- COMPUTER CONNECTION

USB RS485

R S 4 8 5
NETWORK
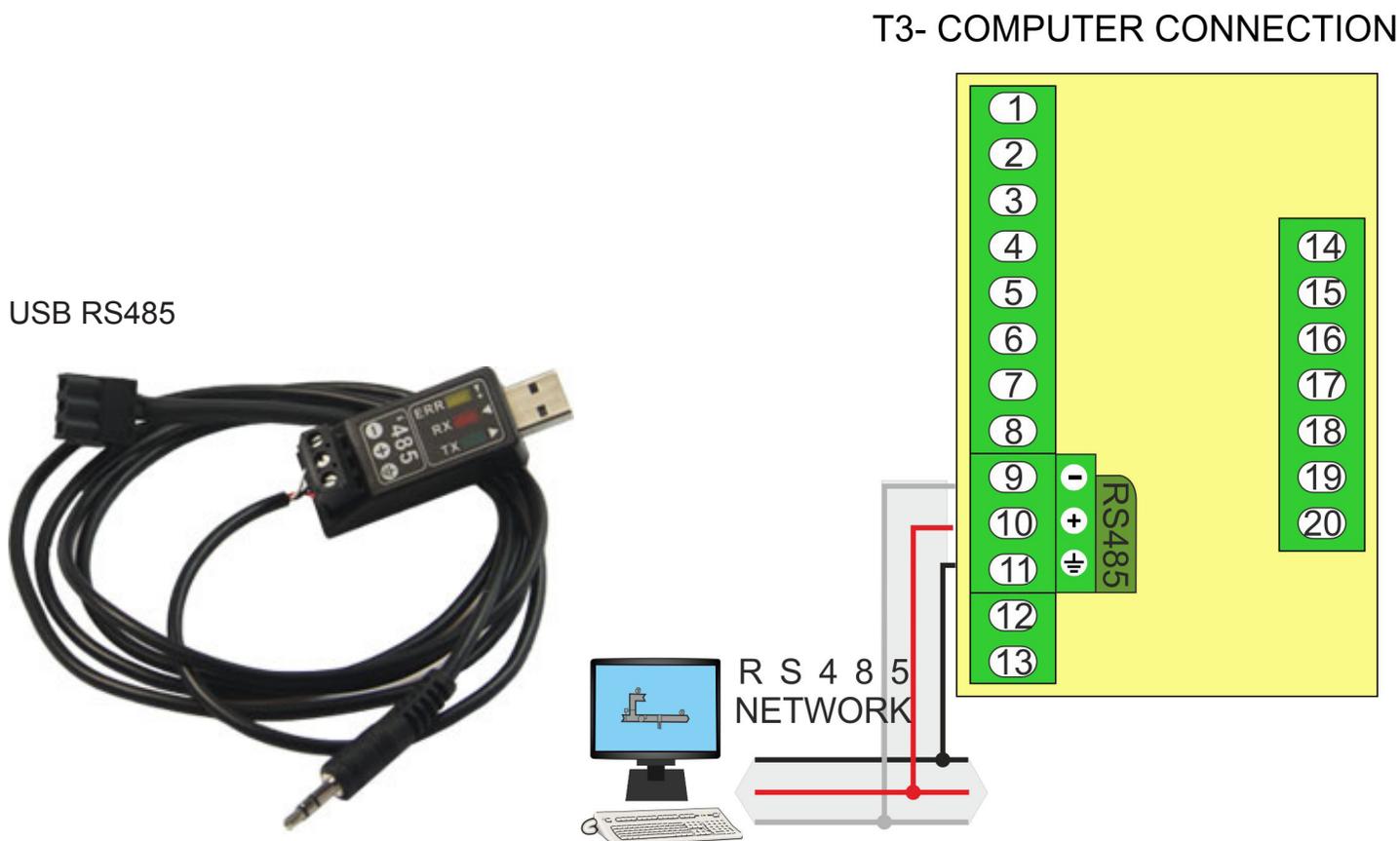
How to set up pulse counting features

1.T3-4AO

You can change the number to choose which chanel as pulse. For example, when you need the 3 chanel as pulse chanel,you can change 6=Pulse Input to 3= Pulse Input.

Address	T3-4AO	
	Bytes	Details
200	1	Range Setting for each input.
201	1	200 correspond to input1,
202	1	201 correspond to input2, etc.
203	1	0 = raw data,
204	1	1 = 10K Celsius,
205	1	2 = 10K Fahrenheit,
206	1	3 = 0-100%
207	1	4 = ON/OFF,
208	1	5 = OFF/ON
209	1	6 = Pulse Input
		7 = Lighting Control 8 = type3 10K Celsius, 9=TYPE 3 10K Fahrenheit,10 = No use, 11 =V0_5,12= V0_10,13 = I0_20ma.

2.T3-8I13O

You can change the number to choose which chanel as pulse. For example, when you need the 3 chanel as pulse chanel,you can change 6=Pulse Input to 3= Pulse Input.

Address	T3-8I13O	
	Bytes	Details
183	1	Range Setting for each input.
184	1	183 correspond to input1,
185	1	184 correspond to input2, etc.
186	1	0 = raw data,
187	1	1 = 10K Celsius,
188	1	2 = 10K Fahrenheit,
189	1	3 = 0-100%
190	1	4 = ON/OFF,
		5 = OFF/ON
		6 = Pulse Input
		7 = Lighting Control 8 = type3 10K Celsius, 9=TYPE 3 10K Fahrenheit,10 = No use, 11 =V0_5,12= V0_10,13 = I0_20ma.

3.T3-28IN

You can change it by jumper.

Address	T3-8I13O		
	Bytes	Details
130	2	Pulse input1 high byte
131	2	Pulse input1 low byte
132	2	Pulse input2 high byte
133	2	Pulse input2 low byte
134	2	Reset Pulse1 number
135	2	Reset Pulse2 number

4.Pulse speed detail for all T3 Module

Module	Frequency
T3-4AO	<1KHZ
T3-8I13O	<10HZ
T3-28IN	≤10KHZ

1.6.2 List of registers in the T3-8AIAO

Note: When using the Modbus Poll software, addressing should be set to "Protocol Addresses (Base 0)" under the "Display" menu.

Address	Bytes	Register and Description
0 to 3	4	Serial Number, 4 byte value
4	1	EEPROM Hardware Version Number
5	1	Firmware Version Number
6	1	ADDRESS. Modbus Device Address
7	1	Product Model
8	1	Hardware Revision
9	1	PIC Version Number
13	1	Calibration Registery – used to calibrate the outputs.
15	1	Baudrate Setting: 0 will set 9600bps, 1 will set 19200bps
22	1	Heartbeat register:It will increase every min from 0 to 255 and cycle the roution
100	2	Output 1 Register
101	2	Output 2 Register
102	2	Output 3 Register
103	2	Output 4 Register
104	2	Output 5 Register
105	2	Output 6 Register
106	2	Output 7 Register
107	2	Output 8 Register
108	2	Input 1 Register
109	2	Input 2 Register
110	2	Input 3 Register
111	2	Input 4 Register
112	2	Input 5 Register
113	2	Input 6 Register
114	2	Input 7 Register
115	2	Input 8 Register
116	2	Register 116 , 117 and 118 hold the position information on each of the hand-off-auto switches on the T3 modules
117	2	Each switch has three positions and therefore each switch requires 2 bits to hold the state. Modbus registers are 16 bits wide so we can hold the status of 8 switches in register 116, the next 8 are held in register 117
118	2	and so on, up to the number of switches on the particular T3 module. The switch states are as follows: 00= Off , the switch is in the center position 10= Auto, the switch is positioned towards the terminal block 01= Hand, manually on. The switch is positioned towards the center of the module (away from the terminal block).
119 - 125	1	Range for each Input, 118 Correspond to ch1.0 = raw data,1 = 10K Celsius, 2 = 10K Fahrenheit ,3 = 0 - 100%,4 = ON/OFF, 5 = OFF/ON
126-133	1	Filter Coefficient for Input 1 to 8, value is 0 through 100,default is 20.

Example: Registery 118 reads 5 (hex 05)

Registery 119 reads 138 (hex 8A)

The Pulse Count for Channel 1 is then 1418 pulse (hex 058A)

Writing to registery 134 will clear registeries 118 and 119. Subsequent registeries 135 to 138 are optional memory to

1.6.3 Lits of Registers in the T3-32I

Note: Addressing should be set to "Protocol Addresses (Base 0)" under the "Display" menu.

Address	Bytes	Register and Description
0 to 3	4	Serial Number, 4 byte value
4	1	EEPROM Hardware Version Number
5	1	Firmware Version Number
6	1	ADDRESS. Modbus Device Address
7	1	Product Model
8	1	Hardware Revision
9	1	PIC Version Number
13	1	Calibration Register – used to calibrate the outputs.
15	1	Baudrate Setting: 0 will set 9600bps, 1 will set 19200bps
22	1	Heartbeat register: It will increase every min from 0 to 255 and cycle the roution
100	2	Input 1 Register
101	2	Input 2 Register
102	2	Input 3 Register
103	2	Input 4 Register
104	2	Input 5 Register
105	2	Input 6 Register
106	2	Input 7 Register
107	2	Input 8 Register
108	2	Input 9 Register
109	2	Input 10 Register
110	2	Input 11 Register
111	2	Input 12 Register
112	2	Input 13 Register
113	2	Input 14 Register
114	2	Input 15 Register
115	2	Input 16 Register
116	2	Register 116 , 117 and 118 hold the position information on each of the hand-off-auto switches on the T3 modules
117	2	Each switch has three positions and therefore each switch requires 2 bits to hold the state. Modbus registers are 16 bits wide so we can hold the status of 8 switches in register 116, the next 8 are held in register 117 and so on, up to the number of switches on the particular T3 module.
118	2	The switch states are as follows: 00= off , the switch is in the center position 10=auto, the switch is positioned towards the terminal block 01= hand, manually on. The switch is positioned towards the center of the module (away from the terminal block).
119	2	Input 20 Register
120	2	Input 21 Register
121	2	Input 22 Register
122	2	Input 23 Register
123	2	Input 24 Register
124	2	Input 25 Register
125	2	Input 26 Register
126	2	Input 27 Register
127	2	Input 28 Register
128	2	Input 29 Register
129	2	Input 30 Register
130	2	Input 31 Register
131	2	Input 32 Register
228-259	1	Range for each input, 228 correspond to ch1.0 = raw data, 1 = 10K Celsius, 2 = 10K Fahrenheit ,3 = 0 - 100%,4 = ON/OFF, 5 = OFF/ON
260-291	1	Filter coefficient for input 1 to 32,value is 0 through 100,default is 20.
336	1	0= waiting for a 1 command;1=reset on cmd, will reset itself to 0 after one reset;2=reset every hour;3=disabled (default)

1.6.4 List of Registers in the T3-8I13O

Address	Bytes	Register and Description
0 to 3	4	Serial Number, 4 byte value
4	1	EEPROM Hardware Version Number
5	1	Firmware Version Number
6	1	ADDRESS. Modbus Device Address
7	1	Product Model
8	1	Hardware Revision
9	1	PIC Version Number
13	1	Calibration Register – used to calibrate the outputs.
15	1	Baudrate Setting: 0 will set 9600bps, 1 will set 19200bps
16-21	1	Reserved
22	1	Heartbeat register:It will increase every min from 0 to 255 and cycle the roution
23-99	1	Reserved
100	2	Output 1 Register
101	2	Output 2 Register
102	2	Output 3 Register
103	2	Output 4 Register
104	2	Output 5 Register
105	2	Output 6 Register
106	2	Output 7 Register
107	2	Output 8 Register
108	2	Output 9 Register
109	2	Output 10 Register
110	2	Output 11 Register
111	2	Output 12 Register
112	2	Output 13 Register
113 - 115	1	Reserved
116	2	Register 116 and 117 hold the position information on each of the hand-off-auto switches on the T3 modules
117	2	Each switch has three positions and therefore each switch requires 2 bits to hold the state. Modbus registers are 16 bits wide so we can hold the status of 8 switches in register 116, the next 8 are held in register 117 and so on, up to the number of switches on the particular T3 module. The switch states are as follows: 00= off , the switch is in the center position 10=auto, the switch is positioned towards the terminal block 01= hand, manually on. The switch is positioned towards the center of the module (away from the terminal block).
118	2	IN1 high word
119	2	IN1 low word
120	2	IN2 high word
121	2	IN2 low word
122	2	N3 high word
123	2	IN3 low word
124	2	IN4 high word
125	2	IN4 low word
126	2	IN5 high word
127	2	IN5 low word
128	2	IN6 high word
129	2	IN6 low word
130	2	IN7 high word
131	2	N7 low word
132	2	IN8 high word
133	2	IN8 low word

1.6.4 List of Registers in the T3-8I13O (continued)

Address	Bytes	Register and Description
134-138	5	Date stamp of Channel 1: Year, Month, Day, Hour, Minute respectively.
139-143	5	Date stamp of Channel 2: Year, Month, Day, Hour, Minute respectively
144-148	5	Date stamp of Channel 3: Year, Month, Day, Hour, Minute respectively.
149-153	5	Date stamp of Channel 4: Year, Month, Day, Hour, Minute respectively.
154-158	5	Date stamp of Channel 5: Year, Month, Day, Hour, Minute respectively.
159-163	5	Date stamp of Channel 6: Year, Month, Day, Hour, Minute respectively.
164-168	5	Date stamp of Channel 7: Year, Month, Day, Hour, Minute respectively.
169-173	5	Date stamp of Channel 8: Year, Month, Day, Hour, Minute respectively.
174	1	Spare
175-182	2	Analog reading from each channel, whatever the channel be set as analog or pulse mode. 175 correspond to ch1
183-190	1	Range for each input, 183 correspond to ch1.0 = raw data, 1 = 10K Celsius, 2 = 10K Fahrenheit ,3 = 0 - 100%,4 = ON/OFF, 5 = OFF/ON
191	1	Filter coefficient for input 1, 0 through 100, default is 20.
192	1	Filter coefficient for input 2, 0 through 100, default is 20.
193	1	Filter coefficient for input 3, 0 through 100, default is 20.
194	1	Filter coefficient for input 4, 0 through 100, default is 20.
195	1	Filter coefficient for input 5, 0 through 100, default is 20.
196	1	Filter coefficient for input 6, 0 through 100, default is 20.
197	1	Filter coefficient for input 7, 0 through 100, default is 20.
198	1	Filter coefficient for input 8, 0 through 100, default is 20.
199	1	Timer for input 1, how long time the lighting control take over the outputs
200	1	Timer for input 2, how long time the lighting control take over the outputs
201	1	Timer for input 3, how long time the lighting control take over the outputs
202	1	Timer for input 4, how long time the lighting control take over the outputs
203	1	Timer for input 5, how long time the lighting control take over the outputs
204	1	Timer for input 6, how long time the lighting control take over the outputs
205	1	Timer for input 7, how long time the lighting control take over the outputs
206	1	Timer for input 8, how long time the lighting control take over the outputs
207	1	Input1 timer Left, how much time left for the lighting control
208	1	Input 2 timer Left, how much time left for the lighting control
209	1	Input 3 timer Left, how much time left for the lighting control
210	1	Input 4 timer Left, how much time left for the lighting control
211	1	Input 5 timer Left, how much time left for the lighting control
212	1	Input 6 timer Left, how much time left for the lighting control
213	1	Input 7 timer Left, how much time left for the lighting control
214	1	Input 8 timer Left, how much time left for the lighting control
215	1	light control disable/enable, each bit correspond to one output,output1 correspond to least significant bit, 0 = disable,1 = enable
216	1	Select which input as lighting control trigger, 0 = disable lighting control,1= input1,2=input2
217	1	Select which input as lighting control trigger, 0 = disable lighting control,1= input1,2=input2
218	1	Select which input as lighting control trigger, 0 = disable lighting control,1= input1,2=input2
219	1	Select which input as lighting control trigger, 0 = disable lighting control,1= input1,2=input2
220	1	Select which input as lighting control trigger, 0 = disable lighting control,1= input1,2=input2
221	1	Select which input as lighting control trigger, 0 = disable lighting control,1= input1,2=input2
222	1	Select which input as lighting control trigger, 0 = disable lighting control,1= input1,2=input2
223	1	Select which input as lighting control trigger, 0 = disable lighting control,1= input1,2=input2

1.6.4 List of Registers in the T3-8I13O (continued)

Address	Bytes	Register and Description
224	1	Select which input as lighting control trigger,0 = disable lighting control,1= input1,2=input2
225	1	Select which input as lighting control trigger,0 = disable lighting control,1= input1,2=input2
226	1	Select which input as lighting control trigger,0 = disable lighting control,1= input1,2=input2
227	1	Select which input as lighting control trigger,0 = disable lighting control,1= input1,2=input2
228	1	Select which input as lighting control trigger,0 = disable lighting control,1= input1,2=input2

1.6.6 List of Registers in the T3-4AO

Address	Bytes	Register and Description
0 - 3	4	Serial Number 4 Bytes value
4	1	Firmware Version , low byte
5	1	Firmware Version, hi byte
6	1	Modbus Device Address
7	1	Product Model
8	1	Hardware Revision
9	1	PIC Version Number
10-12		Reserved
13	1	Calibration Register-Used to calibrate the outputs
14		Reserved
15	1	Baudrate Setting: 0 = 9600bps, 1 = 19200bps
16	1	Firmware Update Register, used to show the status of firmware updates
17-21		Reserved
22	1	Heartbeat register: It will increase every min from 0 to 255 and cycle the roution
23-99		Reserved
100	2	Output 1 Register
101	2	Output 2 Register
102	2	Output 3 Register
103	2	Output 4 Register
104	2	Output 5 Register
105	2	Output 6 Register
106	2	Output 7 Register
107	2	Output 8 Register
108	2	Output 9 Register
109	2	Output 10 Register
110	2	Output 11 Register
111	2	Output 12 Register
112	0	Reserved
113	0	Reserved
114	0	Reserved
115	0	Reserved
116	2	Register 116 , 117 and 118 hold the position information on each of the hand-off-auto switches on the T3 modules
117	2	Each switch has three positions and therefore each switch requires 2 bits to hold the state. Modbus registers are 16 bits wide so it can hold the status of 8 switches in register 116, the next 8 are held in register 117 and so on, up to the number of switches on the particular T3-Module.
118	2	The switch states are as follows: • 00= OFF, the switch is in the center position • 10= AUTO, the switch is positioned towards the terminal block • 01= HAND, manually on. The switch is positioned towards the center of the module (away from the terminal block).
119	2	Input 1 Register, high word
120	2	Input 1 Register, low word

1.6.5 List of Registers in the T3-4AO (Continued)

Address	Bytes	Register and Description
121	2	Input 2 Register, high word
122	2	Input 2 Register, low word
123	2	Input 3 Register, high word
124	2	Input 3 Register, low word
125	2	Input 4 Register, high word
126	2	Input 4 Register, low word
127	2	Input 5 Register, high word
128	2	Input 5 Register, low word
129	2	Input 6 Register, high word
130	2	Input 6 Register, low word
131	2	Input 7 Register, high word
132	2	Input 7 Register, low word
133	2	Input 8 Register, high word
134	2	Input 8 Register, low word
135	2	Input 9 Register, high word
136	2	Input 9 Register, low word
137	2	Input 10 Register, high word
138	2	Input 10 Register, low word
139-143	5	Date Stamp of Input 1, YEAR, MONTH, DAY, HOUR, MINUTE respectively
144-148	5	Date Stamp of Input 2, YEAR, MONTH, DAY, HOUR, MINUTE respectively
149-153	5	Date Stamp of Input 3, YEAR, MONTH, DAY, HOUR, MINUTE respectively
154-158	5	Date Stamp of Input 4, YEAR, MONTH, DAY, HOUR, MINUTE respectively
159-163	5	Date Stamp of Input 5, YEAR, MONTH, DAY, HOUR, MINUTE respectively
164-168	5	Date Stamp of Input 6, YEAR, MONTH, DAY, HOUR, MINUTE respectively
169-173	5	Date Stamp of Input 7, YEAR, MONTH, DAY, HOUR, MINUTE respectively
174-178	5	Date Stamp of Input 8, YEAR, MONTH, DAY, HOUR, MINUTE respectively
179-183	5	Date Stamp of Input 9, YEAR, MONTH, DAY, HOUR, MINUTE respectively
184-188	5	Date Stamp of Input 10, YEAR, MONTH, DAY, HOUR, MINUTE respectively
189	1	Assign each input sample type. 0 = analog; 1 = pulse. Input 1;correspond to Bit0, input 2;correspond to bit1 and so on.
190	2	Analog Input1 original data
191	2	Analog Input2 original data
192	2	Analog Input3 original data
193	2	Analog Input4 original data
194	2	Analog Input5 original data
195	2	Analog Input6 original data
196	2	Analog Input7 original data
197	2	Analog Input8 original data
198	2	Analog Input9 original data
199	2	Analog Input10 original data
200	1	Range Setting for each input.
201	1	200 Correspond to input1
202	1	201 Correspond to input2, etc.
203	1	0 = Raw Data
204	1	1 = 10K Celsius
205	1	2 = 10K Fahrenheit

1.6.5 List of Registers in the T3-4AO (Continued)

Address	Bytes	Register and Description
206	1	3 = 0 - 100%
207	1	4 = ON/OFF
208	1	5 = OFF/ON
209	1	6 = Pulse Input, 7 = Lighting Control
210	1	Filter Coefficient for input 1, 0 through 100, default is 20.
211	1	Filter Coefficient for input 2, 0 through 100, default is 20.
212	1	Filter Coefficient for input 3, 0 through 100, default is 20.
213	1	Filter Coefficient for input 4, 0 through 100, default is 20.
214	1	Filter Coefficient for input 5, 0 through 100, default is 20.
215	1	Filter Coefficient for input 6, 0 through 100, default is 20.
216	1	Filter Coefficient for input 7, 0 through 100, default is 20.
217	1	Filter Coefficient for input 8, 0 through 100, default is 20.
218	1	Filter Coefficient for input 9, 0 through 100, default is 20.
219	1	Filter Coefficient for input 10, 0 through 100, default is 20.
220	1	Timer for Input 1, how long time the lighting control take over the outputs
221	1	Timer for Input 2, how long time the lighting control take over the outputs
222	1	Timer for Input 3, how long time the lighting control take over the outputs
223	1	Timer for Input 4, how long time the lighting control take over the outputs
224	1	Timer for Input 5, how long time the lighting control take over the outputs
225	1	Timer for Input 6, how long time the lighting control take over the outputs
226	1	Timer for Input 7, how long time the lighting control take over the outputs
227	1	Timer for Input 8, how long time the lighting control take over the outputs
228	1	Timer for Input 9, how long time the lighting control take over the outputs
229	1	Timer for Input 10, how long time the lighting control take over the outputs
230	1	Input 1 Timer Left, how much time left for the lighting control
231	1	Input 2 Timer Left, how much time left for the lighting control
232	1	Input 3 Timer Left, how much time left for the lighting control
233	1	Input 4 Timer Left, how much time left for the lighting control
234	1	Input 5 Timer Left, how much time left for the lighting control
235	1	Input 6 Timer Left, how much time left for the lighting control
236	1	Input 7 Timer Left, how much time left for the lighting control
237	1	Input 8 Timer Left, how much time left for the lighting control
238	1	Input 9 Timer Left, how much time left for the lighting control
239	1	Input 10 Timer Left, how much time left for the lighting control
240	2	Light Control Disable/Enable, each bit correspond to one output, output1 correspond to least significant bit, 0 = disable, 1 = enable
241	1	Select which input as lighting control trigger, 0 = disable lighting control, 1= input1, 2=input2
242	1	Select which input as lighting control trigger, 0 = disable lighting control, 1= input1, 2=input2
243	1	Select which input as lighting control trigger, 0 = disable lighting control, 1= input1, 2=input2
244	1	Select which input as lighting control trigger, 0 = disable lighting control, 1= input1, 2=input2
245	1	Select which input as lighting control trigger, 0 = disable lighting control, 1= input1, 2=input2
246	1	Select which input as lighting control trigger, 0 = disable lighting control, 1= input1, 2=input2
247	1	Select which input as lighting control trigger, 0 = disable lighting control, 1= input1, 2=input2
248	1	Select which input as lighting control trigger, 0 = disable lighting control, 1= input1, 2=input2

List of Registers for the T3-PT10

Address	Bytes	Register and Description
0~3		Serial number
4		Firmware version number
5		software version number
6		modbus service address
7		product model
8		hardware model
9		baudrate
10		spare
11		spare
12		hardware Calibration flag write 2 to this register to hardware calibration.
13		set modbus float int 0= float 1 = int. when set INT, 250 = 25.0C. When set float, show temperature directly 25.0000C
18		Resolution (Resol:XXX.XXX): - Set the period for serial transmission
18~21		spare
22		Heartbeat register:It will increase every min from 0 to 255 and cycle the roution
23~99		spare
100~101	4	When set INT, 100 will show 0 and 101 show the temperature for channel 1, 250= 25.0C when set flaot,100 and 101 show 25.0000C
102~103	4	When set INT, 102 will show 0 and 103 show the temperature for channel 2, 250= 25.0C when set float, 102 and 103 together show 25.0000C
104~105	4	When set INT, 104 will show 0 and 105 show the temperature for channel 3, 250= 25.0C when set float, 104 and 105 together show 25.0000C
106~107	4	When set INT, 106 will show 0 and 107 show the temperature for channel 4, 250= 25.0C when set float, 106 and 107 together show 25.0000C
108~109	4	When set INT, 108 will show 0 and 109 show the temperature for channel 5, 250= 25.0C when set float, 108 and 109 together show 25.0000C
110~111	4	When set INT, 110 will show 0 and 111 show the temperature for channel 6, 250= 25.0C when set float, 110 and 111 together show 25.0000C
112~113	4	When set INT, 112 will show 0 and 113 show the temperature for channel 7, 250= 25.0C when set float, 112 and 113 together show 25.0000C
114~115	4	When set INT, 114 will show 0 and 115 show the temperature for channel 8, 250= 25.0C when set float, 114 and 115 together show 25.0000C
116~117	4	When set INT, 116 will show 0 and 117 show the temperature for channel 9, 250= 25.0C when set float, 116 and 117 together show 25.0000C
118~119	4	When set INT, 118 will show 0 and 119 show the temperature for channel 10, 250= 25.0C when set float, 118 and 119 together show 25.0000C
120~129		spare
130~136	6	real time .second, minute, hour, day, week, month,year
140~159	20	Temperature offset. 140~141 for channel 1 offset 142~143 for channel2 offset ect when set float , 140 and 141 default is 100.000. when set int,140 and 141 default is 100 .
160~167	8	four point calibration raw AD data. (long type)
168~173	6	channel 1 RTD parameter.
174~179	6	channel 2 RTD parameter.
180~185	6	channel 3 RTD parameter.
186~191	6	channel 4 RTD parameter.
192~197	6	channel 5 RTD parameter.
198~203	6	channel 6 RTD parameter.
204~209	6	channel 7 RTD parameter.
210~215	6	channel 8 RTD parameter.

List of Registers for the T3-PT10 Continued

Address	Bytes	Register and Description
216~221	6	channel 9 RTD parameter.
222~227	6	channel 10 RTD parameter.
228	1	range for channel1 0 =C 1=F 2 = ohm.
229	1	range for channel2 0 =C 1=F 2 = ohm.
230	1	range for channel3 0 =C 1=F 2 = ohm.
231	1	range for channel4 0 =C 1=F 2 = ohm.
232	1	range for channel5 0 =C 1=F 2 = ohm.
233	1	range for channel6 0 =C 1=F 2 = ohm.
234	1	range for channel7 0 =C 1=F 2 = ohm.
235	1	range for channel8 0 =C 1=F 2 = ohm.
236	1	range for channel9 0 =C 1=F 2 = ohm.
237	1	range for channel10 0 =C 1=F 2 = ohm.
238	1	filter for channel 1 default = 20
239	1	filter for channel 2 default = 20
240	1	filter for channel 3 default = 20
241	1	filter for channel 4 default = 20
242	1	filter for channel 5 default = 20
243	1	filter for channel 6 default = 20
244	1	filter for channel 7 default = 20
245	1	filter for channel 8 default = 20
246	1	filter for channel 9 default = 20
247	1	filter for channel 10 default = 20

List of Registers for the T3-6CT

Address	Bytes	Register and Description
0 to 3	4	Serial Number, 4 byte value
4	1	EEPROM Hardware Version Number
5	1	Firmware Version Number
6	1	ADDRESS. Modbus Device Address
7	1	Product Model
8	1	Hardware Revision
9	1	PIC Version Number
10-12		Reserved
13	1	Calibration Register-Used to calibrate the outputs
14		Reserved
15	1	Baudrate Setting: 0 = 9600bps, 1 = 19200bps
16		Reserved
17	1	Deadmaster time unit:minute defult:10
18	1	Set the realy status when deadmaster time is finish.0:nochange. 1:on .2:off.
22	1	Heartbeat register:It will increase every min from 0 to 255 and cycle the routin
100	2	Output 1 Register
101	2	Output 2 Register
102	2	Output 3 Register
103	2	Output 4 Register
104	2	Output 5 Register
105	2	Reserved
106	2	Reserved
107	2	Reserved
108	2	CT Input 1 Register
109	2	CT Input 2 Register
110	2	CT Input 3 Register
111	2	CT Input 4 Register
112	2	CT Input 5 Register
113	2	CT Input 6 Register
114	2	Analog Input 1 Register
115	2	Analog Input 2 Register
116	2	Analog Input 3 Register
117	2	Analog Input 4 Register
118	2	Analog Input 5 Register
119	2	Analog Input 6 Register
120	2	Analog Input 7 Register
121	2	Analog Input 8 Register
122	2	Analog Input 9 Register
123	2	Analog Input 10 Register
124	1	Switch Bank 1 Register

List of Registers for the T3-6CT Continued

Address	Bytes	Register and Description
125	1	Switch Bank 2 Register
126	1	Range Setting for each input
127	1	126 correspond to input1
128	1	127 correspond to input2, etc.
129	1	0 = raw data,
130	1	
131	1	1 = 10K Celsius,
132	1	
133	1	
134	1	2 = 10K Fahrenheit
135	1	
136	1	
137	1	3 = 0-100%
138	1	
139	1	4 = ON/OFF
140	1	
141	1	5 = OFF/ON
142	1	Filter coefficient for input 1,0 through 100,default is 20.
143	1	Filter coefficient for input 2,0 through 100,default is 20.
144	1	Filter coefficient for input 3,0 through 100,default is 20.
145	1	Filter coefficient for input 4,0 through 100,default is 20.
146	1	Filter coefficient for input 5,0 through 100,default is 20.
147	1	Filter coefficient for input 6,0 through 100,default is 20.
148	1	Filter coefficient for input 7,0 through 100,default is 20.
149	1	Filter coefficient for input 8,0 through 100,default is 20.
150	1	Filter coefficient for input 9,0 through 100,default is 20.
151	1	Filter coefficient for input 10,0 through 100,default is 20.
152	1	Filter coefficient for input 11,0 through 100,default is 20.
153	1	Filter coefficient for input 12,0 through 100,default is 20.
154	1	Filter coefficient for input 13,0 through 100,default is 20.
155	1	Filter coefficient for input 14,0 through 100,default is 20.
156	1	Filter coefficient for input 15,0 through 100,default is 20.
157	1	Filter coefficient for input 16,0 through 100,default is 20.

For example, if we would like to read the input 2 register at module node address #1,

Slave Address	Function	Starting Address Hi	Starting Address Lo	No. of Points Hi	No. of Points Lo	CRC Hi Byte	CRC Hi Byte
1	3	0	109	0	1	xx	xx

Or we read 8 values after input 2 in module 1,

Slave Address	Function	Starting Address Hi	Starting Address Lo	No. of Points Hi	No. of Points Lo	CRC Hi Byte	CRC Hi Byte
1	3	0	109	0	8	xx	xx

Or we write 600 to output 4 in module 1,

Slave Address	Function	Starting Address Hi	Starting Address Lo	No. of Points Hi	No. of Points Lo	CRC Hi Byte	CRC Hi Byte
1	6	0	103	0	600	xx	xx

More details can be found in Modbus Secial Communication Section below.

1.6.7 Note About Registers When Updating the Firmware

There are two registers that will tell the CPU information about the model and hardware of the T3-Module.

NOTE: after updating the firmware you MUST setup these registers first or the module may not function properly.

Product Model is Register Address 7. The model number is already configured and it's a fixed number when it leaves the factory. There is no need for the customer to do anything.
The corresponding values are as follows:
- T3-4AO = 28
- T3-8AIAO = 21
- T3-8I13O = 20
- T3-32AI = 22
- T3-6CT = 29
- T3-PT10 = 26

Hardware revision is register address 8. The hardware revision can be found by removing the front cover of the module. It is written in white silkscreen on the edge of the board.

1.7 Lighting Control with the T3-8I-13O Module

It is possible to use the T3-Module as a lighting control module. The logic is embedded directly in the T3-Module to make light switching response faster and reduce the polling required for lighting applications. The idea is that each input can be configured as a lighting switch input. Any or all of the inputs can be configured this way. Each input corresponds to one lighting zone.

Next, any or all of the outputs can be assigned to any of the zones. Finally, each zone has a timer which allows the operator to set how long the zone will go to "occupied mode".

An additional set of timers show how much time is left for a particular override event, these are read/write registers so that the Master can initiate events and override events already under way. Finally, there is an auto/manual bit for each output to override the local T3 logic, for example during commissioning or special events.

1. • The "Range" for each input is configured in registers 183 thru 190,
     set any input to act as a lighting switch by setting the range to 7.
   • Each input corresponds to one zone,
   • Short circuit the input to GND and this will trigger an event
   • If the zone was previously on, it will flip to off and vice versa.

2. • Each Zone has an 'override time' setting which sets how long the lights will be triggered on for a particular hit of the switch.
   • The values are in minutes, they are read/write and are stored in registers 199 thru 206 ,
   • 199 is the time for zone 1 which is controlled by input1
   • 200 for zone2 and so on..

3 • Each zone has a 'time left' register which shows the remaining time left after a particular hit on the switch.
   • These are read/write values in 'minutes' and are stored in registers 207 thru 214
   • Each time there is a hit on a particular hit on a light switch, the 'time left' register will be filled in with this 'over ride time' setting.
   • For example, a hit on switch 1 will trigger a copy of register 199 to register 207.
   • Then Register 207 will start counting down.

4 • Each output has an auto/manual bit so that the lighting control logic (and any other future logic embedded in the module) can be disabled.
   • Register 215 , auto/manual register, 2 byte length.
   • 0 = manual (lighting control disabled),
   • 1 = auto (lighting control enabled).
   • Each bit corresponds to one output with output 1 starting at the least significant bit
   • Output 13 corresponds to the 13th bit.

5 • Assign Outputs to Zones in registers 216 thru 228
   • 13 outputs, one register for each output which assigns that particular output to a particular zone.
   • Since there are 8 zones, the these registers will accept a value from 1 to 8,
       • 0 = n/a,
       • 1 = means this output will be linked to zone1 (and controlled by input1).
       • 2 = means this output will be linked to zone2 (and controlled by input2).
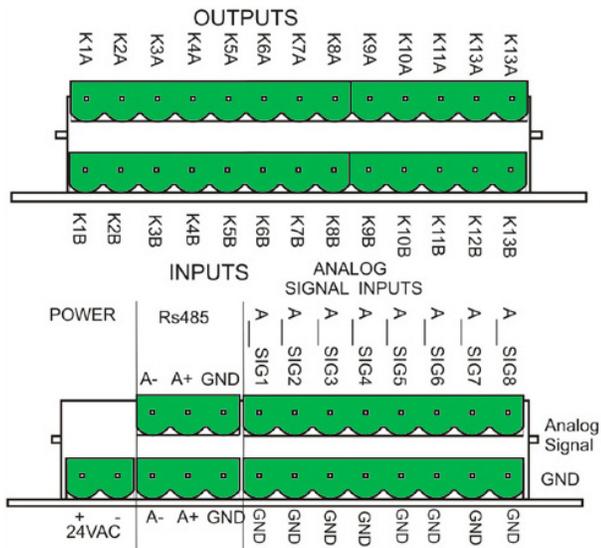   • ETC

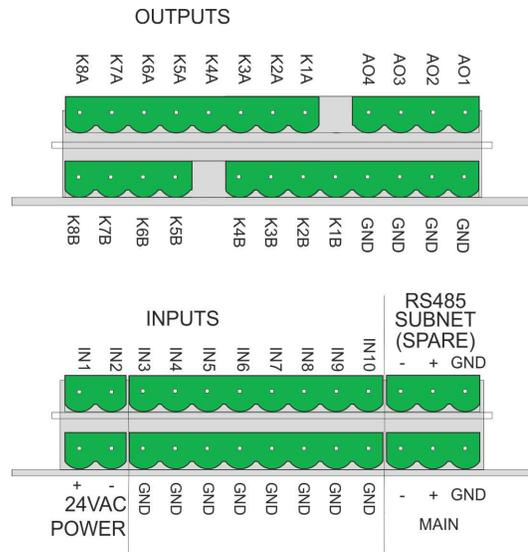## 1.8 Installation

1.8.1 Terminal Block Connections (T3-8IO)
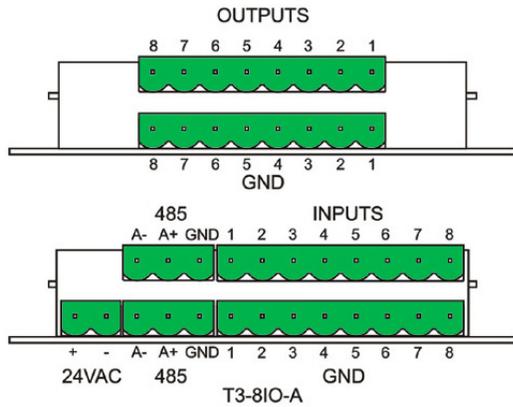
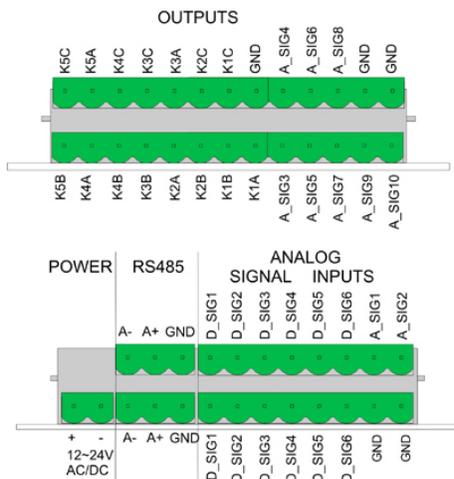### 1.8.2 Terminal Block Connections (T3-32I)
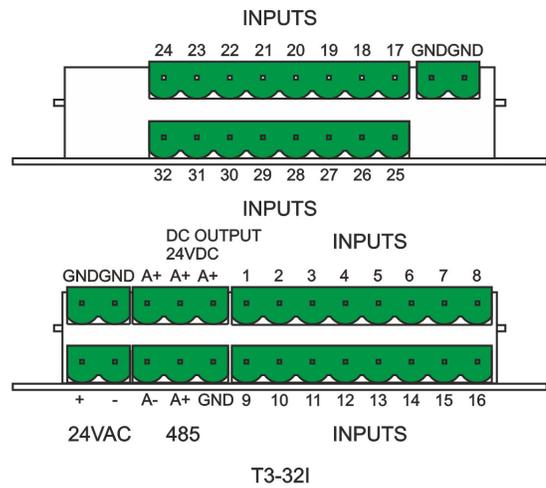
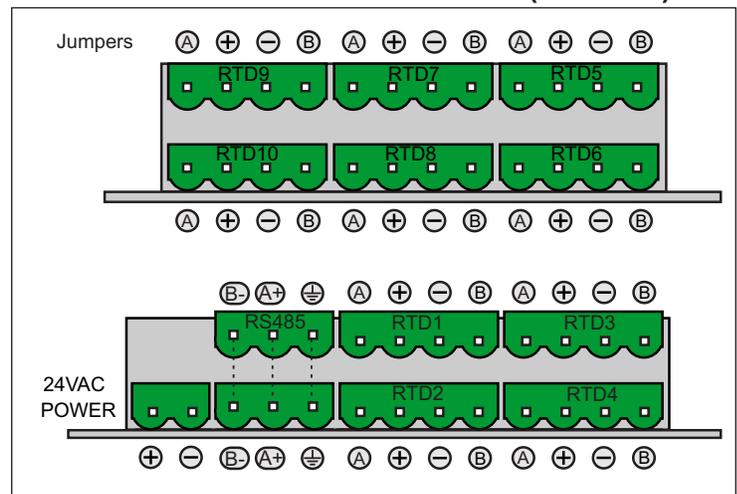### 1.8.3 Terminal Block Connections (T3-8-13

1.8.4 Terminal Block Connections (T3-4AO)


1.8.5 Terminal Block Connections (T3-6CT)

1.8.6 Terminal Block Connections (T3-PT10)

1.9 Typical Sensor Inputs

Title	Typical Wiring Diagram	Jumper
Thermistors		1 2 3 4 5
Dry Contacts		1 2 3 4 5
Transducer Signal 0-10V 0-5V	POWER GND ⚠ T3 module and transducer power share GND. Transducer shall be half wave rectified.	1 2 3 4 5
Transducer 4-20mA	4~20mA POWER GND ⚠ T3 module and transducer power share GND. Transducer shall be half wave rectified.	1 2 3 4 5

2.0 Network Diagram
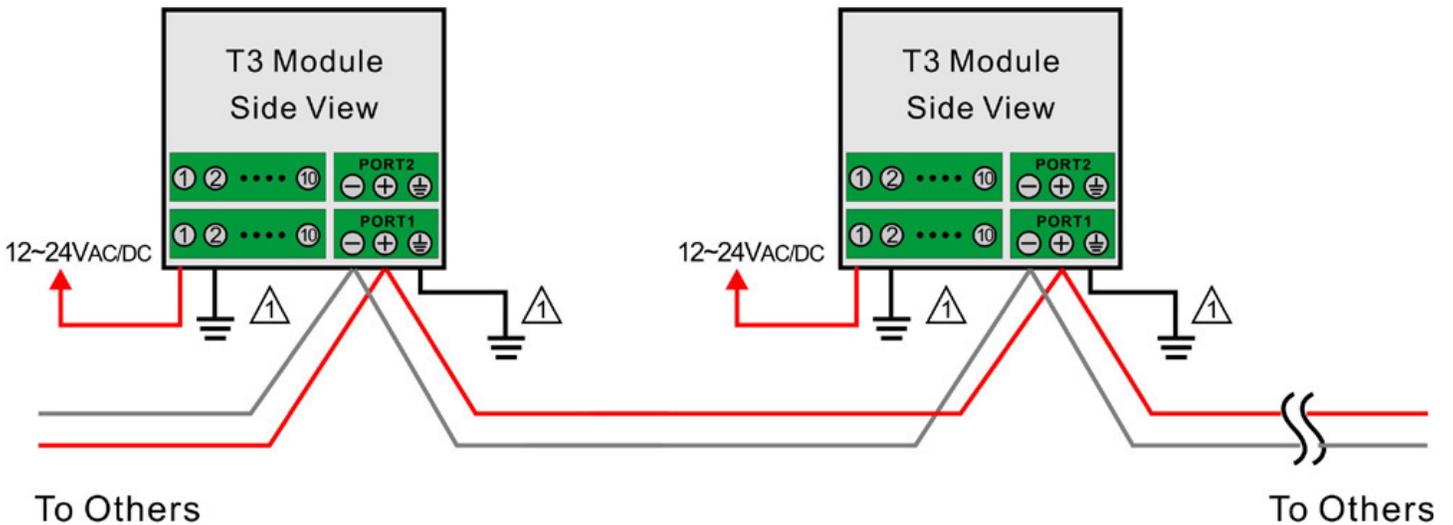
⚠ Tie GNDs to earth ground at each cabinet.

1.9.6 Mounting

- • External wiring is connected to a terminal block on the circuit board
- • The enclosure is comprised of a base section and a cover
- • The base section can be mounted directly on a wall or on a wall box
- • Length of cables : No practical limitation on cable length
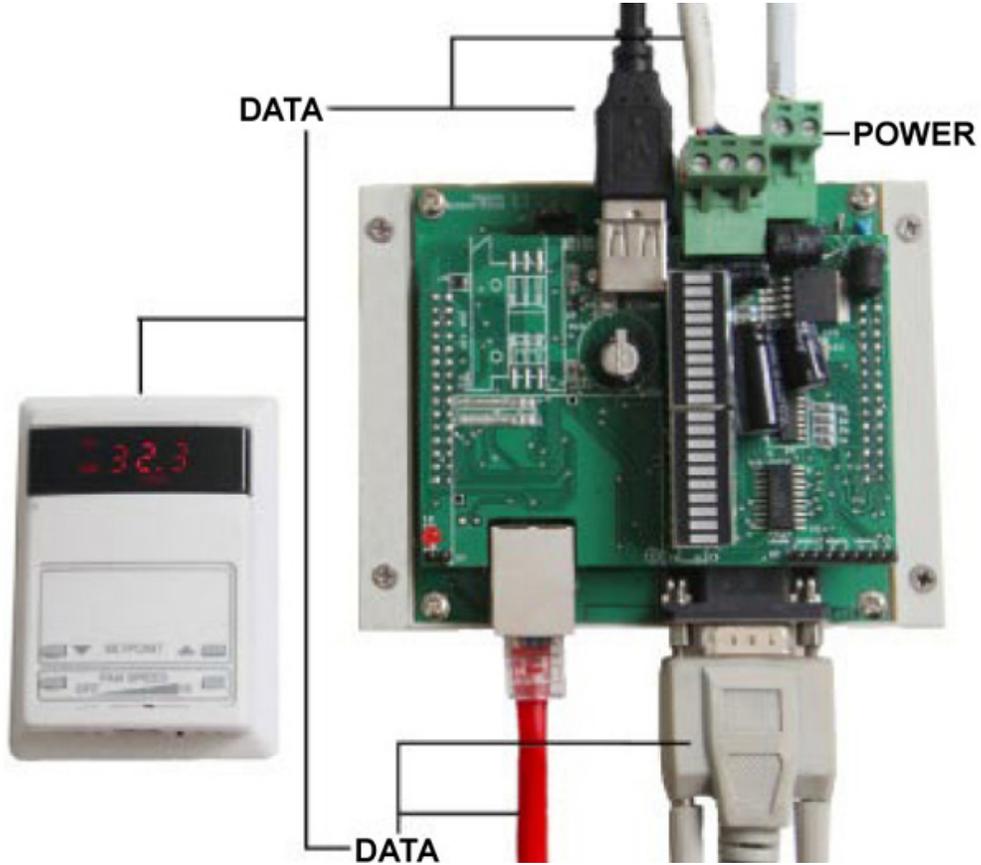
1.9.7 T3-4AO Jumper Settings
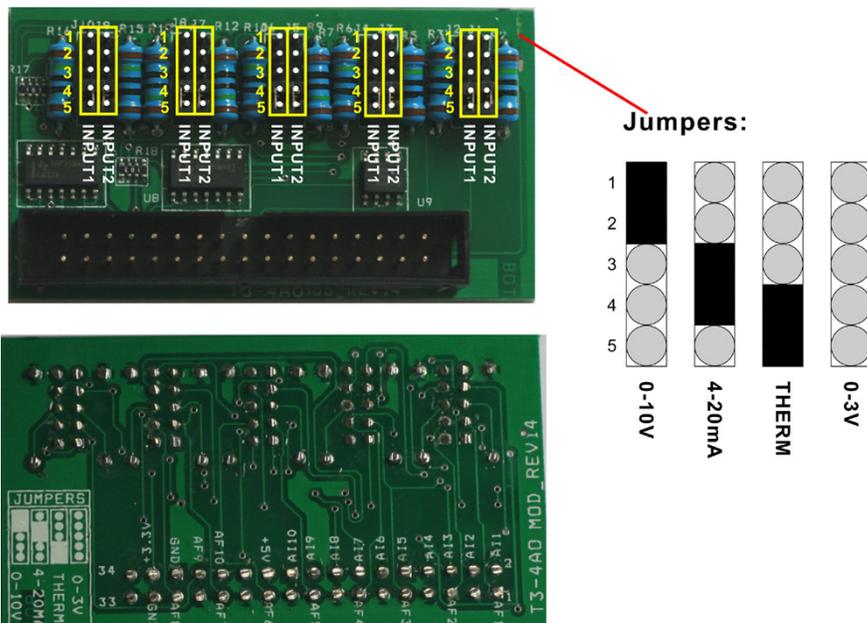
2 Modbus Serial Communications

2.1 Overview

Modbus protocol is a widely used and well-documented communication method. It provides simple and effective means of programming our various products.

A typical Modbus packet looks like this:

Byte1   Modbus ID, the destination address for a particular message
Byte2   Function
Byte3   Starting address of the particular storage registers) to be read or written, hi byte,
Byte4   Starting address low byte
Byte5   No. of registers to read/write (hi byte)
Byte7   CRC hi byte
Byte8   CRC low byte

During normal operation, the slave will immediately send a response to the master request.

MASTER	SLAVE
Send Request	
Command	
Data Request	→ **Request Received**
	Perform Command
	Interpret Data
Response Received ←	**Send Response**
Interpret Data	
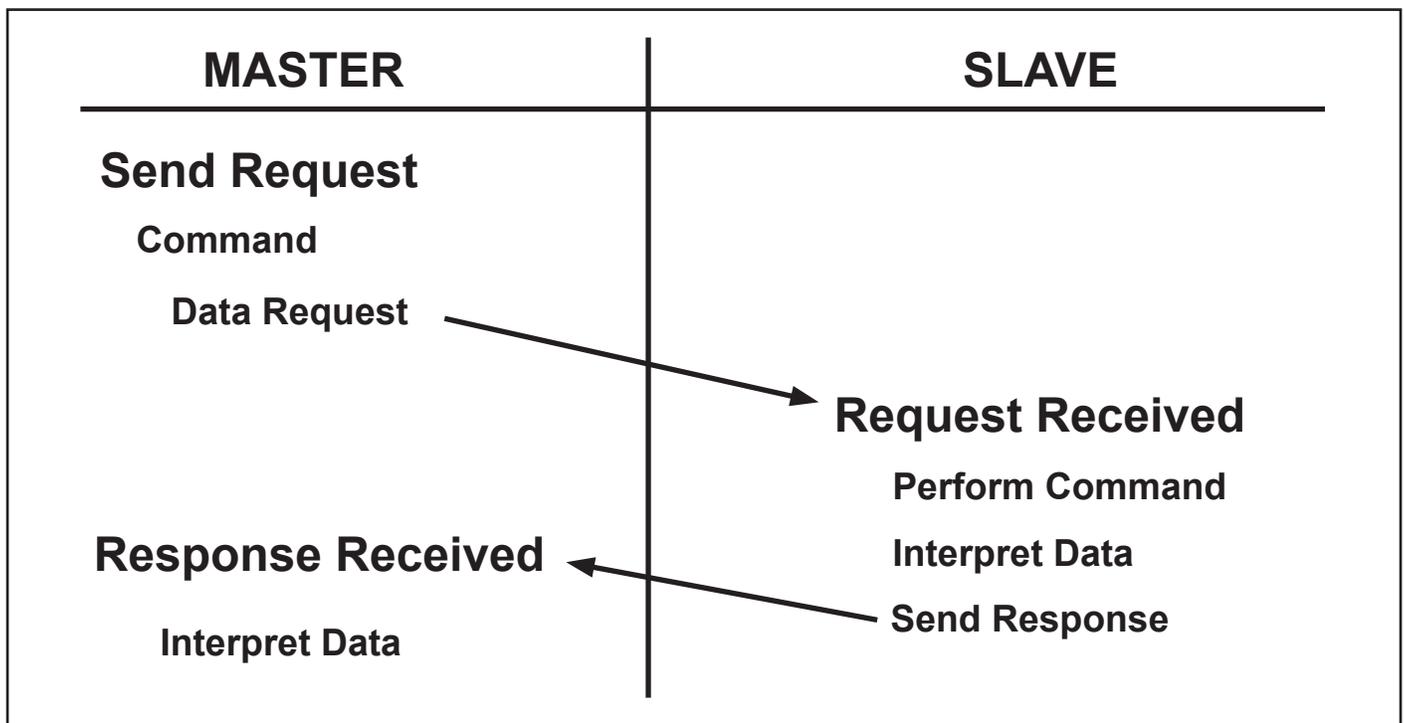
[Notice]: Most errors during message transfer are timeout errors. This is because bytes being distorted or missing will not trigger a response, resulting in a timeout error.

Software tools can be found at: http://www.modbustools.com/modbus_poll.asp If your application can read & write bytes to a separate PC running the 'Modbus Slave' application, you will be able to read & write bytes to theTstat5. Note: When using the Modbus Poll software, addressing should be set to "Protocol Addresses (Base 0)" under the "Display" menu.

2.2 Modbus Examples

2.2.1 READ Command (0x03)

This function is used to read the contents of multiple memory registers. The master to the Modbus must specify, the Modbus ID, it's starting register and quantity of register desired. By convention if a data were to contain 2 byte, we would first send the Hi byte and then the Lo byte. You can change the Modbus ID whatever you want by writing register 6.

The master to the Modbus network will issue a read command:
- Modbus ID=11
- Read 6 bytes of data
- Starting at register number 107 (6Bh)

Byte #	Field Name (Hex)	Data	Description
Byte1	Slave Address	11	Tstat with ID11 will be read
Byte2	Function	03	Read operation
Byte3	Starting Address Hi	00	
Byte4	Starting Address Lo	6B	Reading starting from register #6B
Byte5	No. of Register to read Hi	00	
Byte6	No. of Register to read Lo	03	Read a total of 3 registers
Byte7	Error Check (CRC) HI byte	XX	The CRC is calculated using the CRC
Byte8	Error Check (CRC) LO byte	XX	Routine described below

The slave device with ID=11 will answer the master within a few milliseconds with the following response.

Byte #	Field Name (Hex)	Data	Description
Byte1	Slave Address	11	Slave with ID11 is responding
Byte2	Function	03	We're responding to a read command
Byte3	Byte Count	06	6 bytes are coming
Byte4	Data1 Hi	02	Byte1 of the data
Byte5	Data1 Lo	2B	Byte2 of the data
Byte6	Data2 Hi	00	Byte3 of the data
Byte7	Data2 Lo	00	Byte4 of the data
Byte8	Data3 Hi	00	Byte5 of the data
Byte9	Data3 Lo	64	Byte6 of the data
Byte10	Error Check (CRC) HI byte	XX	The CRC is calculated using the CRC
Byte11	Error Check (CRC) LO byte	XX	Routine described below

Example of the Read Command:
The Master sends the Read querie:

Slave Address	Function	Starting Address Hi	Starting Address Lo	No. of Points Hi	No. of Points Lo	CRC Hi Byte	CRC Lo Byte
11	3	0	(6Bh) 107	0	3	xx	xx

The device node sends back the following response:

Slave Address	Function	Byte Count	Data1 Hi	Data1 Lo	Data2 Hi	Data2 Lo
11	3	6	(02h) 2	(2Bh) 43	(00h) 0	(00h) 0

Data3 Hi	Data3 Lo	CRC Hi Byte	CRC Lo Byte
(00h) 0	(64h) 100	xx	xx

2.2.2 WRITE command (0x06)

This function is used to write to a single memory register. The master of the Modbus must specify the Modbus ID, its register address to be written and the data desired.

The master to the Modbus network will issue a write command:

- Modbus ID=11
- Write to address 11
- Enter data 3 (03h)

Byte #	Field Name (Hex)	Data	Description
Byte1	Slave Address	11	Destination address
Byte2	Function	06	This is a write command
Byte3	Register Address Hi	00	Address which will be written to, hi byte
Byte4	Register Address Lo	01	Address which will be written to, low byte
Byte5	Data Hi	00	Data that we are writing, hi byte
Byte6	Data Lo	03	Data we are writing, low byte
Byte7	Error Check (CRC) HI byte	XX	The CRC is calculated using the CRC
Byte8	Error Check (CRC) LO byte	XX	Routine described below

The slave device with ID=11 will answer the master within a few milliseconds with the following response.

Byte #	Field Name (Hex)	Data	Description
Byte1	Slave Address	11	Destination address
Byte2	Function	06	This is a write command
Byte3	Register Address Hi	00	Address which will be written to, hi byte
Byte4	Register Address Lo	01	Address which will be written to, low byte
Byte5	Data Hi	00	Data that we are writing, hi byte
Byte6	Data Lo	03	Data we are writing, low byte
Byte7	Error Check (CRC) HI byte	XX	The CRC is calculated using the CRC
Byte8	Error Check (CRC) LO byte	XX	Routine described below

[Notice]: In this case the Slave device just sends back the message to let the Master know the query has been properly received.

Example of the Write Command

The Master sends the Write Query:

Slave Address	Function	Starting Address Hi	Starting Address Lo	Data Hi	Data Lo	CRC Hi Byte	CRC Lo Byte
11	6	0	(01h) 1	0	3	xx	xx

The device node sends back the following response:

Slave Address	Function	Starting Address Hi	Starting Address Lo	Data Hi	Data Lo	CRC Hi Byte	CRC Lo Byte
11	6	0	(01h) 1	0	3	xx	xx

2.2.3 MULTIPLE-WRITE Command (0x10)

This function is used to write to multiple memory registers. The master of the Modbus must specify the Modbus ID, its starting address register, the amount of register desired and the data. NOTE: This is used for firmware update only. It is not used to write device registers.

The master to the Modbus network will issue a multiple-write command:
- Modbus ID=11
- Write to address 291 (123h)
- Number of Registers 3
- Data 1 = 10 (000Ah)
- Data 2 = 11(000Bh)
- Data 3 = 12 (000Ch)

Byte #	Field Name (Hex)	Data	Description
Byte1	Slave Address	11	Destination address ID 11
Byte2	Function	10	This is a multiple write command
Byte3	Register Start Address Hi	01	This is the address we are currently writing to in the code space of the device
Byte4	Register Start Address Lo	23	In this case we want to write to register address 0x0123
Byte5	Quantity of Registers to write Hi	00	We will be writing a variable amount of bytes at a time
Byte6	Quantity of Registers to write LOW	10	in this case we want to write to 10H or 16 registers
Byte7	Byte Count	20	If byte count is the same as number of Registers, dealing with 8 bits. If byte count is the same as number of Registers, dealing with 16 bits.

Byte #	8 bits	Byte #	16 bits
Byte8	Data 1	Byte8	Data1 Hi
Byte9	Data 2	Byte9	Data1 Lo
Byte10	Data 3	Byte10	Data2 Hi
Byte11	Data 4	Byte11	Data2 Lo
[…]		[…]	
Byte22	Data 15	Byte38	Data16 Hi
Byte23	Data 16	Byte39	Data16 Lo
Byte 24	Error Check HI	Byte40	Error Check HI
Byte 25	Error Check LO	Byte41	Error Check LO

[Notice]: Byte 7 is used as a byte count. Thus if the byte count is the same as the number of registers to write then we know we are dealing with 1 byte registers. Similarly, if the byte count is double the number of registers, we are dealing with 2 byte registers.

The slave device with ID=11 will answer the master within a few milliseconds with the following response.

Byte #	Field Name (Hex)	Data	Description
Byte1	Slave Address	11	Destination node ID
Byte2	Function	10	This is a multiple write command
Byte3	Register Start Address Hi	00	Starting address we are writing to, hi byte
Byte4	Register Start Address Lo	01	Start address low byte
Byte5	Quantity of Registers Hi	00	Number of registers to be written to, hi byte
Byte6	Quantity of Registers Lo	0A	Number of registers, low byte
Byte7	Error Check (CRC) HI byte	XX	The CRC is calculated using the CRC
Byte8	Error Check (CRC) LO byte	XX	Routine described previously

Example of the Multiple-Write Command
The Master sends the Multiple-Write querie:

Slave Address	Function	Starting Address Hi	Starting Address Lo	Quantity. of Regs Hi	Quantity. of Regs Lo	Byte Count
11	(10h) 16	(01h) 1	(23h) 35	0	3	6

Data 1 Hi	Data 1 Lo	Data 2 Hi	Data 2 Lo	Data 3 Hi	Data 3 Lo	CRC Hi Byte	CRC Lo Byte
(00h) 00	(0Ah) 10	(00h) 00	(0Bh) 12	(00h) 00	(0Ch) 13	xx	xx

Slave Address	Function	Starting Address Hi	Starting Address Lo	Quantity. of Regs Hi	Quantity. of Regs Lo	CRC Hi Byte	CRC Lo Byte
11	10	(01h) 1	(23h) 35	0	3	xx	xx

2.3 CRC Error Correcting Details

The following is a collection of code snippets to get your application started:

static unsigned char auchCRCHi[ ] = {

```
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81,
0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0,
0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01,
0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81,
0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0,
0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01,
0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81,
0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0,
0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01,
0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81,
0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0,
0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01,
0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81,
0x40
} ;
```

/* Table of CRC values for low–order byte */

static unsigned char auchCRCLo[ ] = {

```
0x00, 0xC0, 0xC1, 0x01, 0xC3, 0x03, 0x02, 0xC2, 0xC6, 0x06, 0x07, 0xC7, 0x05, 0xC5, 0xC4,
0x04, 0xCC, 0x0C, 0x0D, 0xCD, 0x0F, 0xCF, 0xCE, 0x0E, 0x0A, 0xCA, 0xCB, 0x0B, 0xC9, 0x09,
0x08, 0xC8, 0xD8, 0x18, 0x19, 0xD9, 0x1B, 0xDB, 0xDA, 0x1A, 0x1E, 0xDE, 0xDF, 0x1F, 0xDD,
0x1D, 0x1C, 0xDC, 0x14, 0xD4, 0xD5, 0x15, 0xD7, 0x17, 0x16, 0xD6, 0xD2, 0x12, 0x13, 0xD3,
0x11, 0xD1, 0xD0, 0x10, 0xF0, 0x30, 0x31, 0xF1, 0x33, 0xF3, 0xF2, 0x32, 0x36, 0xF6, 0xF7,
0x37, 0xF5, 0x35, 0x34, 0xF4, 0x3C, 0xFC, 0xFD, 0x3D, 0xFF, 0x3F, 0x3E, 0xFE, 0xFA, 0x3A,
0x3B, 0xFB, 0x39, 0xF9, 0xF8, 0x38, 0x28, 0xE8, 0xE9, 0x29, 0xEB, 0x2B, 0x2A, 0xEA, 0xEE,
0x2E, 0x2F, 0xEF, 0x2D, 0xED, 0xEC, 0x2C, 0xE4, 0x24, 0x25, 0xE5, 0x27, 0xE7, 0xE6, 0x26,
0x22, 0xE2, 0xE3, 0x23, 0xE1, 0x21, 0x20, 0xE0, 0xA0, 0x60, 0x61, 0xA1, 0x63, 0xA3, 0xA2,
0x62, 0x66, 0xA6, 0xA7, 0x67, 0xA5, 0x65, 0x64, 0xA4, 0x6C, 0xAC, 0xAD, 0x6D, 0xAF, 0x6F,
0x6E, 0xAE, 0xAA, 0x6A, 0x6B, 0xAB, 0x69, 0xA9, 0xA8, 0x68, 0x78, 0xB8, 0xB9, 0x79, 0xBB,
0x7B, 0x7A, 0xBA, 0xBE, 0x7E, 0x7F, 0xBF, 0x7D, 0xBD, 0xBC, 0x7C, 0xB4, 0x74, 0x75, 0xB5,
0x77, 0xB7, 0xB6, 0x76, 0x72, 0xB2, 0xB3, 0x73, 0xB1, 0x71, 0x70, 0xB0, 0x50, 0x90, 0x91,
0x51, 0x93, 0x53, 0x52, 0x92, 0x96, 0x56, 0x57, 0x97, 0x55, 0x95, 0x94, 0x54, 0x9C, 0x5C,
0x5D, 0x9D, 0x5F, 0x9F, 0x9E, 0x5E, 0x5A, 0x9A, 0x9B, 0x5B, 0x99, 0x59, 0x58, 0x98, 0x88,
0x48, 0x49, 0x89, 0x4B, 0x8B, 0x8A, 0x4A, 0x4E, 0x8E, 0x8F, 0x4F, 0x8D, 0x4D, 0x4C, 0x8C,
0x44, 0x84, 0x85, 0x45, 0x87, 0x47, 0x46, 0x86, 0x82, 0x42, 0x43, 0x83, 0x41, 0x81, 0x80,
0x40
} ;
```

For example: to calculate the crc of the data in the message stored in memory location *puchMsgunsigned short CRC16 (unsigned char *puchMsg, unsigned char usDataLen

```
{
     unsigned char uchCRCHi = 0xFF ; /* high byte of CRC initialized */
     unsigned char uchCRCLo = 0xFF ; /* low byte of CRC initialized */
     unsigned uIndex ; /* will index into CRC lookup table */
     while (usDataLen--) /* pass through message buffer */
{
          uIndex = uchCRCHi ^ *puchMsg++ ; /* calculate the CRC */
          uchCRCHi = uchCRCLo ^ auchCRCHi[uIndex] ;
          uchCRCLo = auchCRCLo[uIndex] ;
     }
     return (uchCRCHi << 8 | uchCRCLo) ;
}
```

2.4 Modbus Poll Software

Modbus Poll is a simple modbus communications tool developed by Witte Communications http://www.modbustools.com/modbus_poll.asp that can be used to read and write registers of modbus devices. The following is a brief set of instructions for communicating with a device.
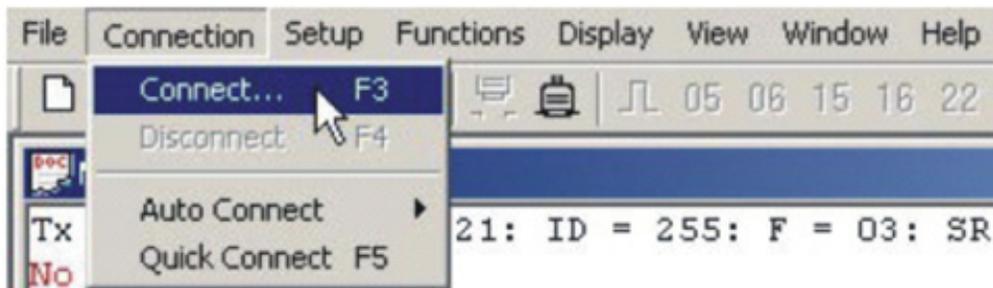
The first time Modbus Poll is used, it should be set to base 0 addressing. This is done by selecting "Protocol Addressing (Base 0) " from the Display menu:
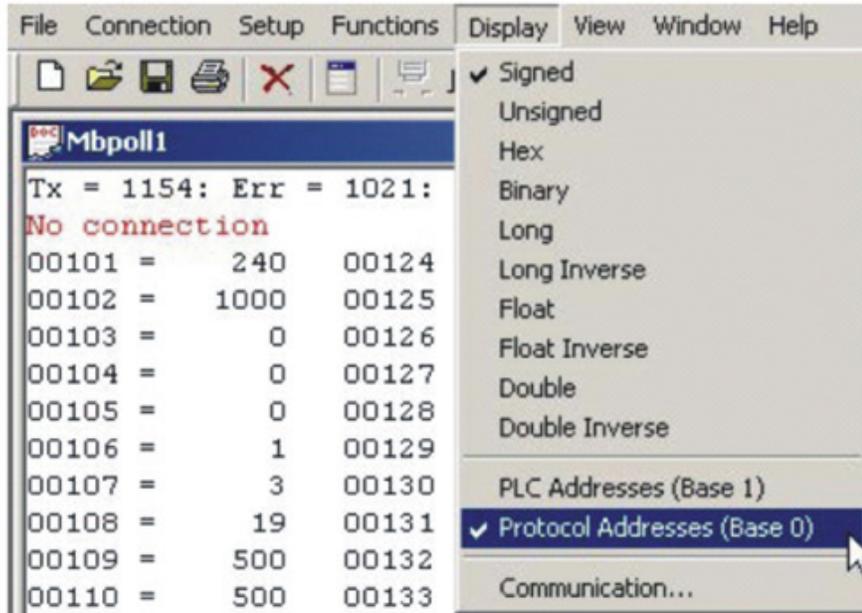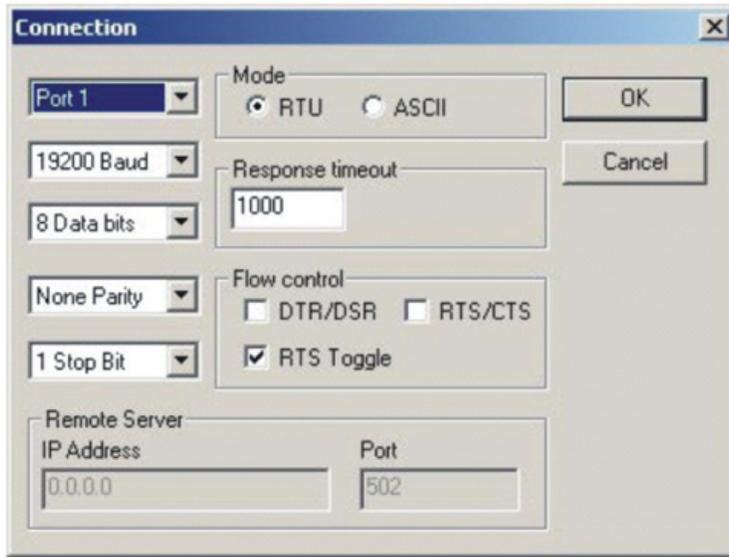

The Setup menu:
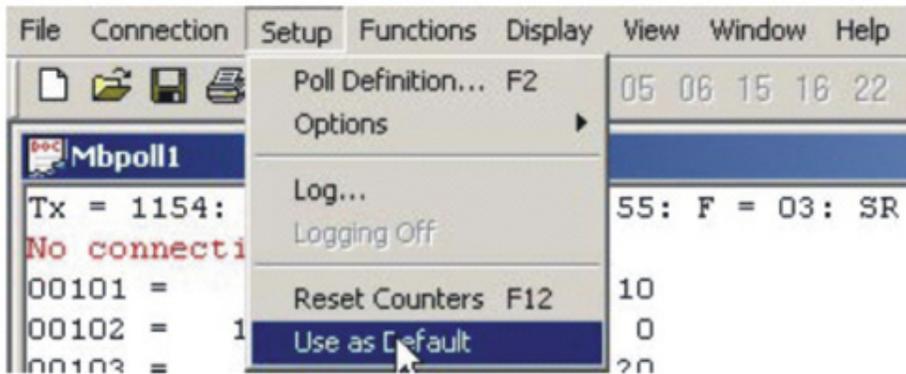
At this point, the connection to the device needs to be established. Select "Connect..." from the Connection menu:

Unless the device has specifically been setup for 9600 baud, the default connections settings should be as follows:

After the connection is established, it is necessary to setup the poll definitions. This is done by selecting "Poll Definition..." from the Setup menu:
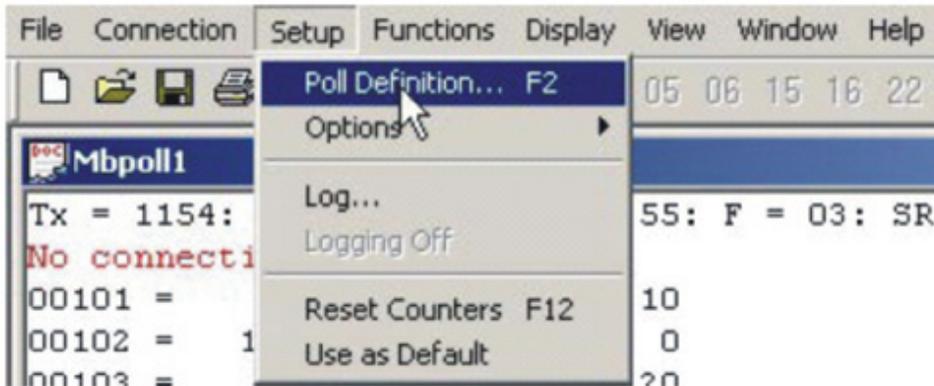
Within the Poll Definitions dialog window, there are several parameters that need to be set.
- Slave ID is the modbus address of the device being read or written. (255 is the generic address to which all devices will respond.)
- Function should be set as 03 HOLDING REGISTER.
- Address is the starting address of the registers to be read.
- Length is the number of registers to be read.
- Scan Rate is the frequency with which the device will be polled.
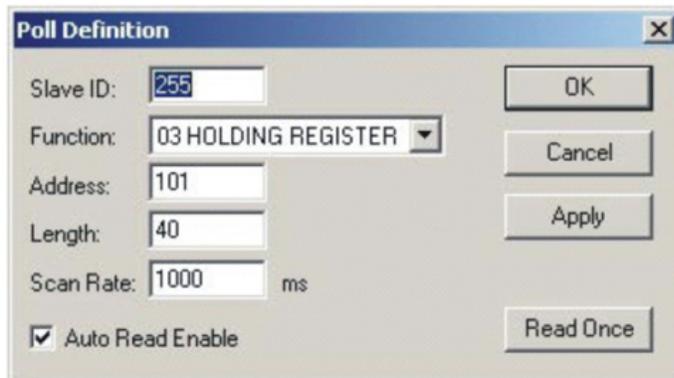
Once the Poll Definitions have been setup and applied, the main window will show a list of each register address and its corresponding value.
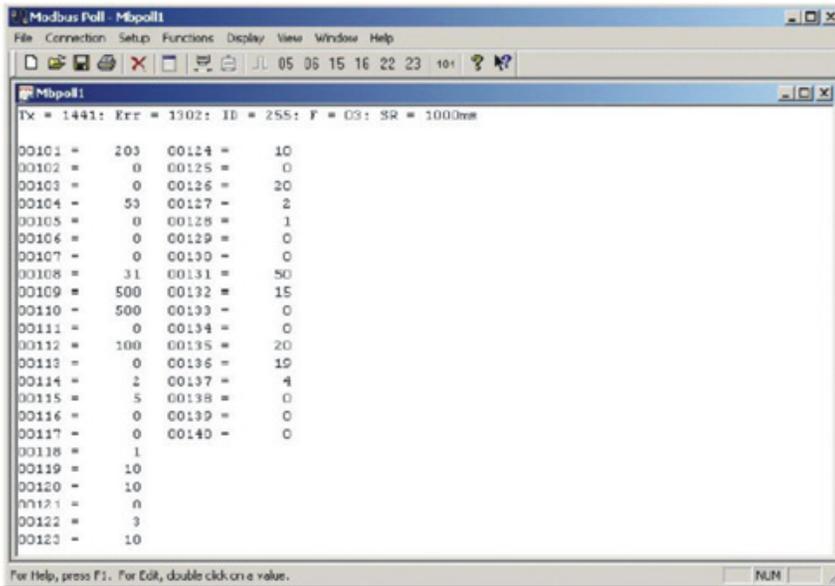
In order to write a value to a specific register, select "06 Write Single Register..." from the Functions menu:
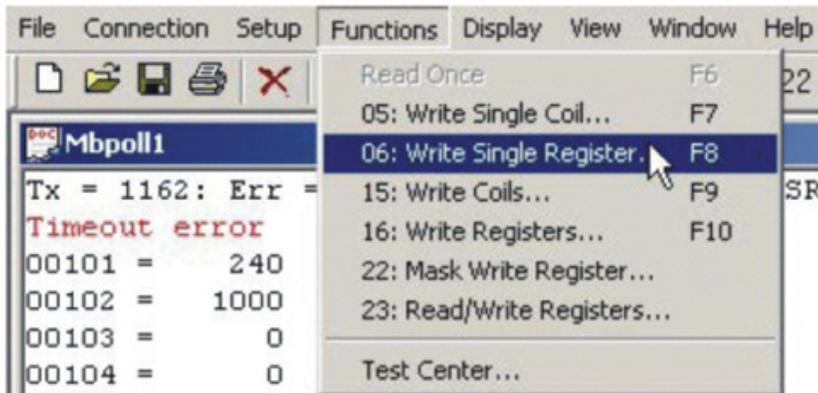
Slave ID is the modbus address of the device. Address is the address of the register that will be written. Value is the value being written.
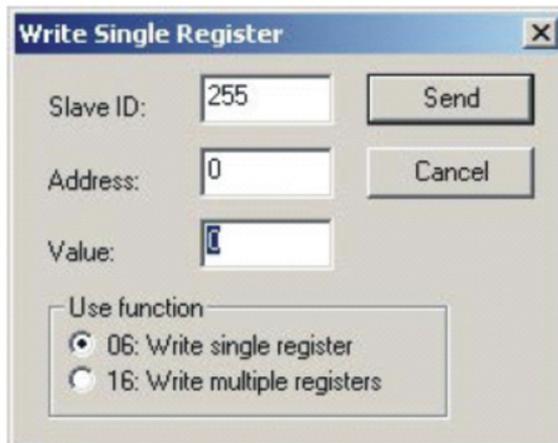
3 Instructions for Updating Devices with Temco ISP

For TEMCO devices that utilize the Temco ISP, the flash update must be done using the provided NWT3000. To perform a firmware update, follow these instructions:

1.) Download and install the NWT3000 software: http://www.temcocontrols.com/ftp/software/9TstatSoftware.zip
2.) Connect the device to the serial port of your computer using the RS232-485 converter included in the package.
3.) Power up the device.
4.) Open the NWT3000 software and select Update Firmware from the Tool menu:
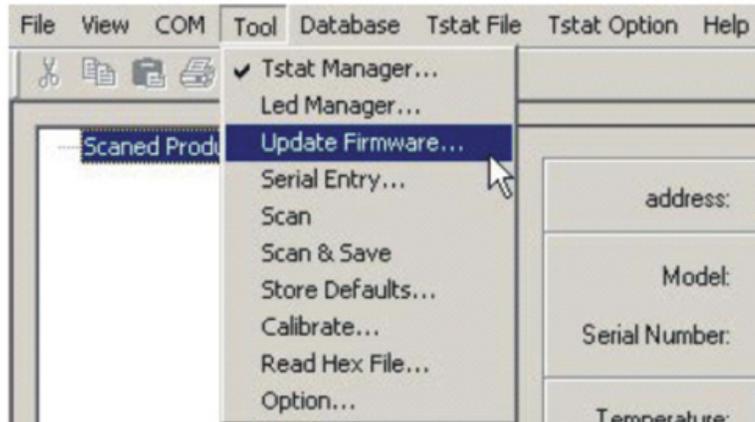
5.) The software will now open the Update Firmware window and will scan for available devices.
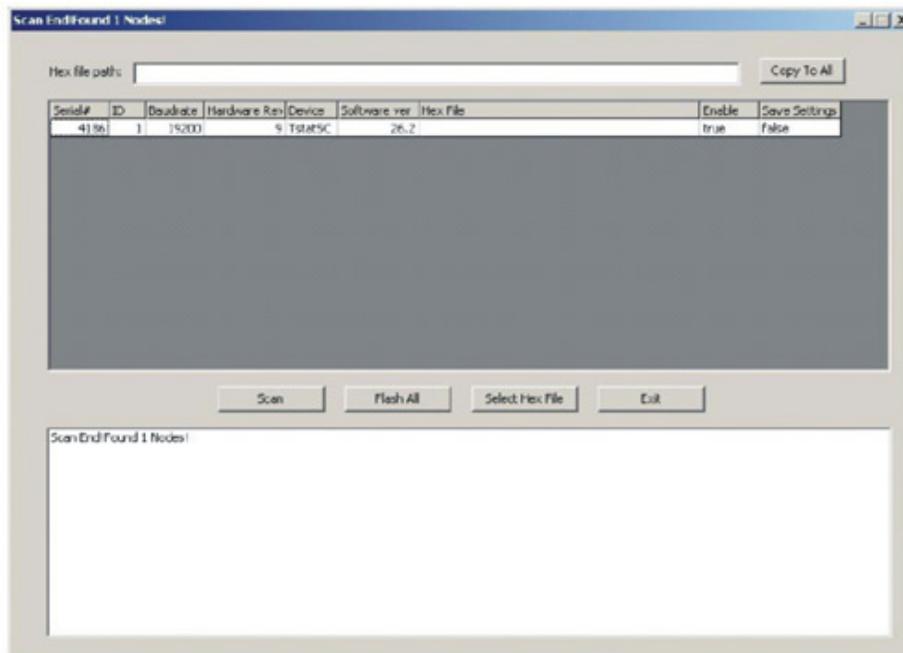
6.) For each device that is found, you can specify the hex file to be used for the update. Do this by clicking in the Hex File column of the row you wish to specify. Alternatively you can click Select Hex File and then "Copy to All" if all devices are to receive the same file. You can also choose to save the current settings or to load the default settings by selecting True or False from the Save Settings column.
7.) At this point simply click Flash All and the software will update each device one by one.

3.1 Protocol for Developers Wanting to Update Devices with Temco ISP

All devices programmed with Temco ISP are capable of being updated over the RS485 network. The master on the network sends a command to a particular device, which forces it to go into a 'flash update mode'. The device first resets itself and then jumps to the 'In System Programming' (ISP) code section. Note that all non-volatile parameters should be read and saved prior to this for safe keeping.

NOTE: Multiple-Write Command of the Modbus protocol is used.

3.1.1 Protocol

In order for the front end to communicate with the ISP flash, a series of registers have been defined, which are used as control registers for the Update functions. Reading and writing to these registers will allow the Front end to monitor the status of the update process. They are stored in the non-volatile memory space to keep track of the steps attempted and completed. Below is a description of these control status registers.

Register	Register address	Description
EEPROM_VERSION_NUMBER	4	Software Version
EEP_ADDRESS	6	ID number of the device
EEP_UPDATE_STATUS	16	Update Register state

Table 1. Flash Update Function Registers

It is important to note 'EEP_UPDATE_STATUS' which is located at register address 16. Writing to this register will cause the device to either reset itself, erase its flash or start programming depending on the action being taken. Below is a description of the values and explanation of the EEP_UPDATE_STATUS register.

Function	Value	Description of EEP_Update_Status
Update initialize	7Fh	Tell the Tstat to reset and jump into the ISP to be in update mode
Update ready		Tstat is in the ISP and ready to update
Erase flash	3Fh	Tell the Tstat to erase Flash Memory
Erase done		Erase Flash Memory done
Start Programming	1Fh	Start Programming - In upload state
Normal State	01h	Update is complete, tstat reboots with new flash image

Table 2. EEP_UPDATE_STATUS register value description

- For the device to jump into update mode, a write command of value 7Fh must be sent to the EEP_UPDATE_STA TUS. The device will then reset itself and run in ISP mode.
  Note: The device will not send any response in this step. To verify the T3-Module is in ISP mode, the same write command must be sent again (write 7Fh to register #16), at which point the T3-Module will respond with a regular modbus response. This is necessary for clearing the Interrupt vectors and making sure all RAM memory is cleared.
- All Modbus communication commands are followed by a response. This Flash Update Protocol makes use of that criteria and thus only sends a response once the action has been completed. Therefore the 'update initial ize' and 'erase flash' step require a longer timeout period than the 'programming' step. (250ms and 500ms respec tively)
- Sending a write command of value 3Fh to EEP_UPDATE_STATUS will force the device to erase its entire flash memory. Once the response is received, the device is ready to download the data of the new firmware.
- Sending a write command of value 1Fh to EEP_UPDATE_STATUS will let the device know it is about to receive new firmware. The device is now ready to accept the new hex file and will maintain a running tally of the current programming location in the EEP_UPDATE_PTR.
- At this point, the data must be sent using the multiple-write command. Packets can be of size 1 data byte to a maximum of 128 data bytes.
- In the event of an interrupted flash update, the Master can poll the EEP_UPDATE_PTR and begin programming from this location.

3.1.2 Example of a Programming Routine

The ISP has been designed using polling vectors rather than Interrupt vectors in order to free up as many interrupts for the program itself. Given that polling is now used, communications are more susceptible to timing and response delay problems. Therefore, when sending a write function or multiple write function to the ISP device, a short timeout delay is required before receiving a response (≈20ms). If a response was not received during that period of time the FRONT END would need to resend the data once again. Below is a diagram representation of the Flash-Update Protocol.
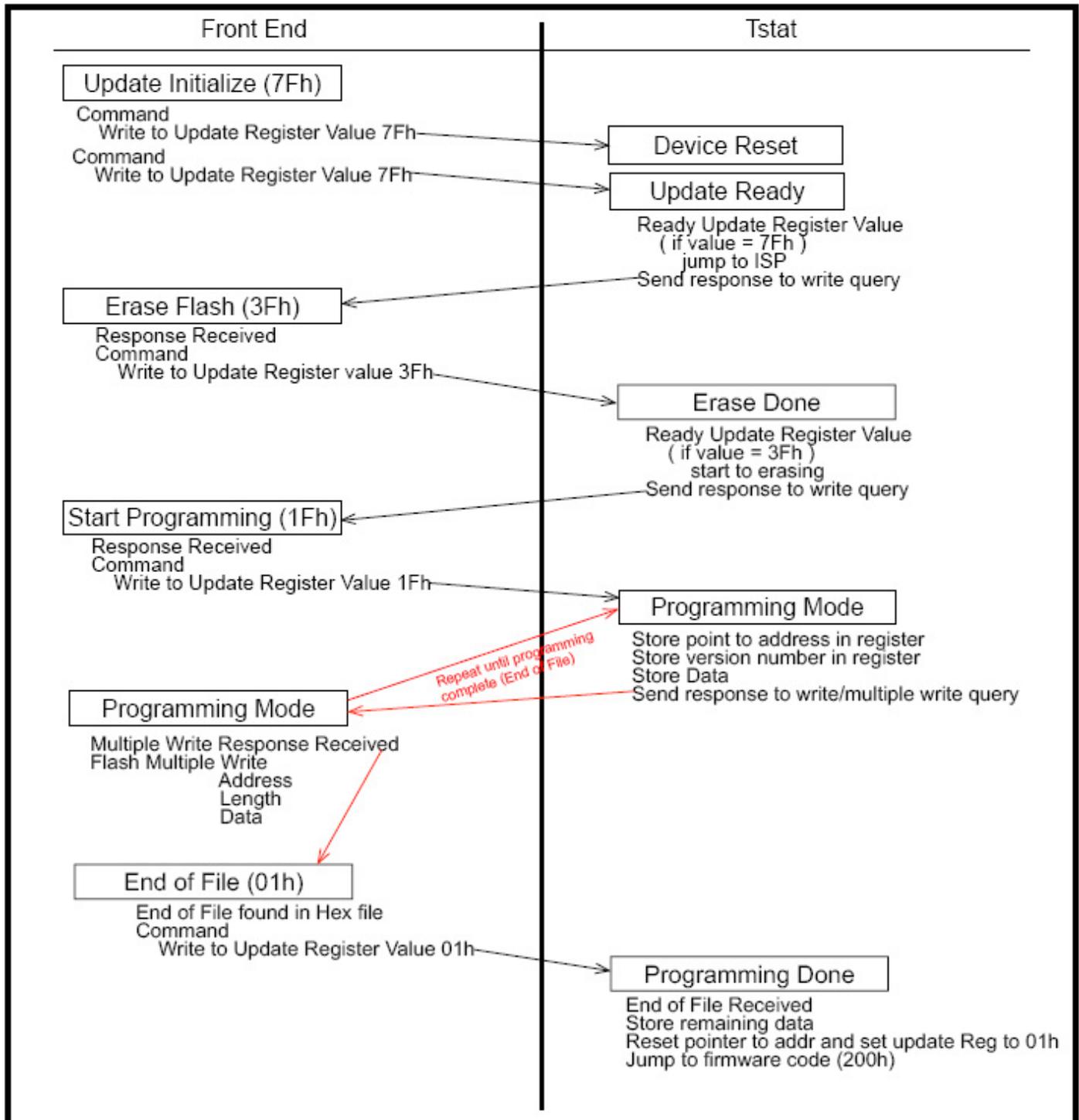
3.1.3 Example of a Programming Routine (Front End Side)

UPDATE INITIALIZE
1. • Send Modbus Write Command to address Update_Register value 7Fh
   • The device will reset itself. Make sure all volatile infomation be saved prior to this step
   • Device will not send a respond

2. • Send Modbus Write Command to address Update_Register value 7Fh again
   • A response will be received if the Device has properly reset itself and booted under ISP mode

ERASE FLASH
3. • Send Modbus Write Command to address Update_Register value 3Fh
   • A response will be received once the Device has properly Erase all Flash Memory
   • This will step require a longer response timeout period (approx 500ms)

TART PROGRAMMING
4. • Send Modbus Write Command to address Update_Register value 1Fh
   • A response will be received once the Device has properly set itself for programming mode

PROGRAMMING MODE
5. • Extracting data from Intel Hex file. A typical line would look like the following:
   :10 0080 00 AF5F67F0 602703E0 322CFA92 007780C3 FD
6. • Verify checksum
   10 + 00 +08 + 00 + AF + ... + C3 + FD = 900
   • If two last digits of the sum is zero, Hex file is correct
7. • Send data using Modbus Multiple-Write Command
    Address 0080h
    Data length of 10h
    Data AF5F67F0 602703E0 322CFA92 007780C3
8. • Repeat step 5 through 7 until end of Hex file is reached

   IMPORTANT NOTE to ensure proper reset of the device, the value at address register 0000h of the Goal chip must remain as FF.

   • Most (but not all) of Temco's Hex file will contain this line:
        :03 0000 00 020200 F9
   • Data written to the Goal Flash register MUST be modified from 020200 to FF0200

END OF FILE
9. • End of file found in Hex file
   :00 0000 01 FF
   Bit 7 and 8 are 01
10. • Send Modbus Write Command to address Update_Register value 01h
   • This will cause the device to reset itself and boot in normal operation mode

3.1.4 To Resume a Previously Interrupted Programming Routine

The EEP_UPDATE_STATUS register keeps track of which step is being performed during the update protocol and the EEP_UPDATE_PTR keeps track of which register is currently being written to.

• If the device was in the Erase Flash mode, the EEP_UPDATE_STATUS register will read 3Fh. The Front End is then required to repeat this step and follow up from there.
• If the device was in the Programming mode, the EEP_UPDATE_STATUS register will read 1Fh. The Front Ends then needs to read the EEP_ UPDATE_PTR. In order to resume this step the Front End needs to re-write to this register again and then follow up from there.

The following diagram represents the update resume procedure.
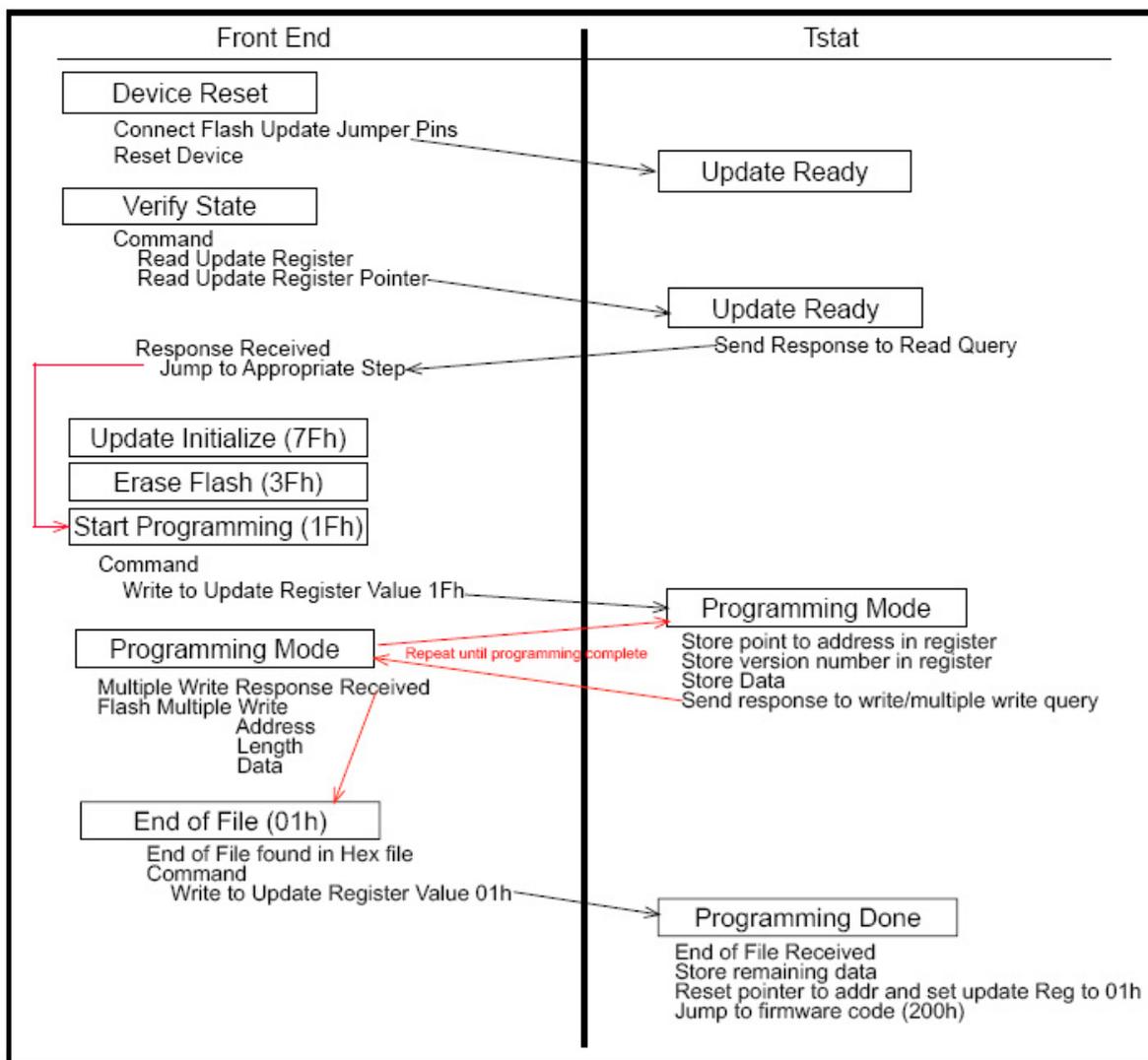
IMPORTANT:
In order for the device to jump into the ISP mode, it has to reset itself. Upon reset, if the value at address register 0000h is FF, the device will jump to the ISP code section. This is a hardware criteria of the Goal Chip and an efficient way to jump to In System Programming mode while clearing all buffers. The front end must ensure that only value FF is to be written to address register 0000h. When reading the hex file, there will be a line like this:

Data of the new firmare   Modified data to be uploaded
;03 0000 00 020200 F9   --------   Needs to be changed to   --------   ;03 0000 00 FF0200 FC
(Intel Hex format described below):

3.1.5 Intel Hex File

All firmware files produced by our compilers are saved under the Intel Hex file format. This format of record can be broken down in its different fields as described below.

3.1.5.1 Example of an Intel Hex file

Take for instance a typical message such as the following:
                    :ll   aaaa tt   D1D2D3D4 D5D6D7D8 D9D0D1D2 D3D4D5D6 ee
                    :10 0080 00 AF5F67F0   602703E0   322CFA92   007780C3   61
• The first character (:) indicates the start of a record.
• The next two characters indicate the record length (10h).
• The next four characters give the load address (0080h).
• The next two characters indicate the record type. (00)
• Then we have our data
• The last two characters are a checksum (sum of all bytes + checksum = 00).
Record types:
• 00 - Data record
• 01 - End of file record
• 02 - Extended segment address record
• 03 - Start segment address record
• 04 - Extended linear address record
• 05 - Start linear address record

3.1.6 Intel Hex File

In the case where the device is locked, there is still a possibility to reboot the device and upload a new firmware. This requires to physically link the jumpers of the Flash Update Jumper pins during restart:

    • Power down the device
    • Link the jumpers of the Flash Update Jumpers
   • Power up the device

Doing the above steps will force the device to be in ISP mode so that new firmware can be loaded. In order to return to normal operation once the upload has been done the Jumper needs to be removed and power need to be recycled.

4 Getting Started and Installing T3 - Module

4.1 Getting Started:

When setting up your T3-Module please check and set the following settings, and read the following instructions in order to have the module function according to your specific needs.

1. Connect 2-Holed power adapter into the 2-Pin port marked "Power" on the T3-Module.

2. Plug the 24 V power cable into the wall.

3. Connect 3-Hole RS485 cableinto the 3-Pin port marked "RS485" on the T3-Module.

4. Connect the other end of the RS485 Cable into an empty Com Port in your computer.

5. If you have not installed Modbus Poll on your system, please do so now

   i.   Double click the file "ModbusPollSetup.exe"

   ii. Accept the terms of the licence agreement and press "Next"

   iii. Select your desired location to install the product and press "Next"

   iv. Select the components you wish to install and press "Install"

   v.   Once the installation has completed press "Next"

   vi. Press "Finish" to complete

6. On your computer open ModBus Poll

7. Before you start collecting data you must decide if you want to run your module in Base 1 or Base 0

   • Scroll over the "Display" tab and click to see the drop down menu

   • Select "Base 1" or "Base 0" from the drop down menu
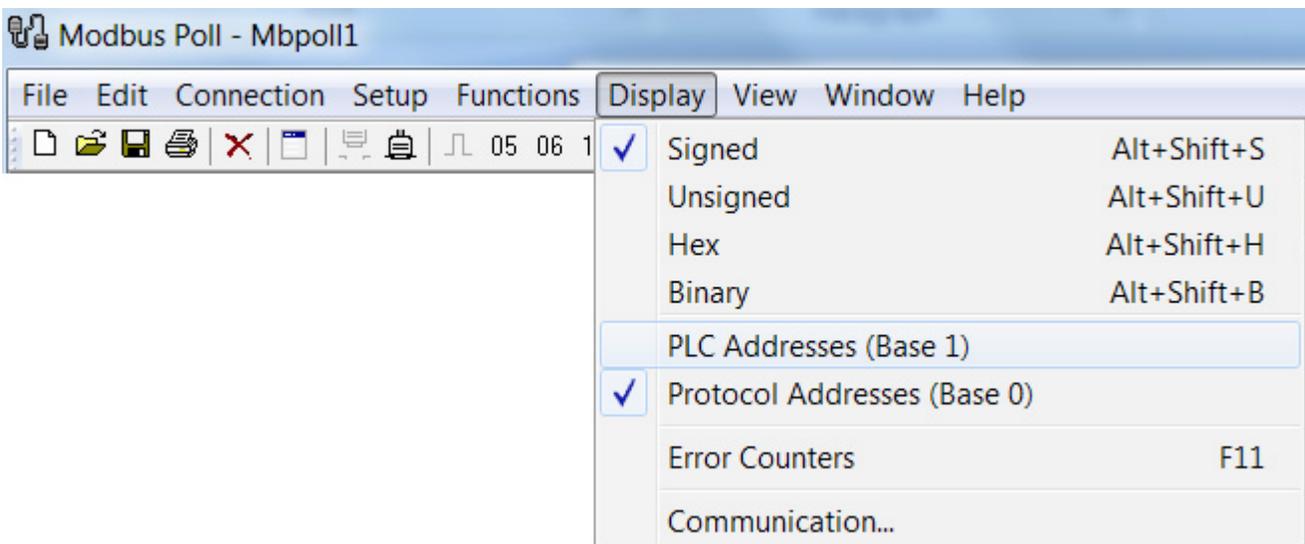
• If you use "Base 0" the Registers will be set to those stated below
• If you use "Base 1" the Registers below will be modified by adding 40001
- Ex. Base 0 Register 00186 = Base 1 Register 40187
ValueBase1 = ValueBase0 + 40001

8. Baud Rate controls the speed at which information is retrieved from the T3-Module.
    All T3-Modules can be set to either 19200 or 9600, 19200 being the faster speed.

    • To define the Baud Rate go to "Connections -> Connect"
   and define your Baud Rate.
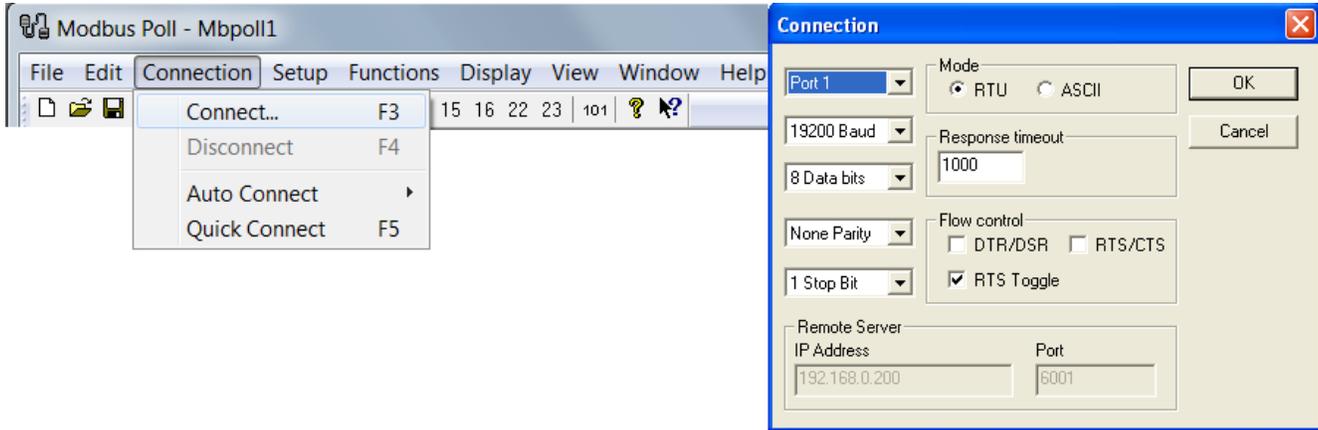
    • If the Baud Rate is set to 19200 then "Double-Click" Register 15
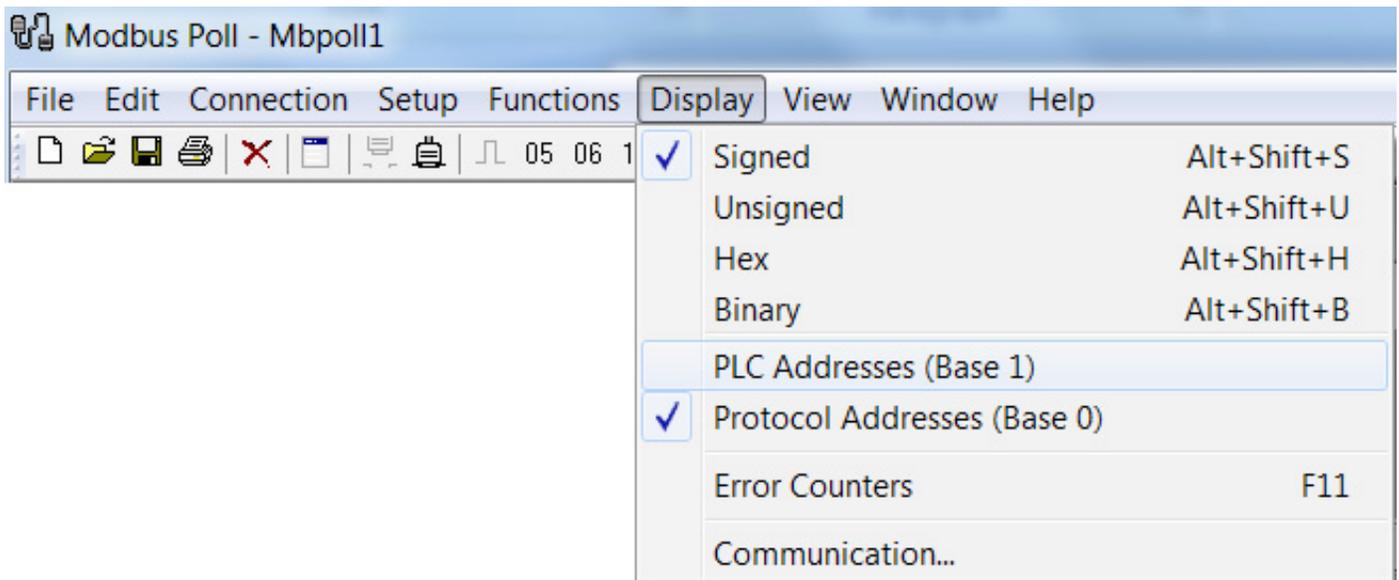   and set the "Value" to "1"

9. Your Com Port must be set before your device can retrieve the information
from the T3-Module.

    • Different computers use different Com Ports, so you must first see which
       Com Port your device is using.

    1. To do this Open Device manager and see which Com Port your T3-Module uses
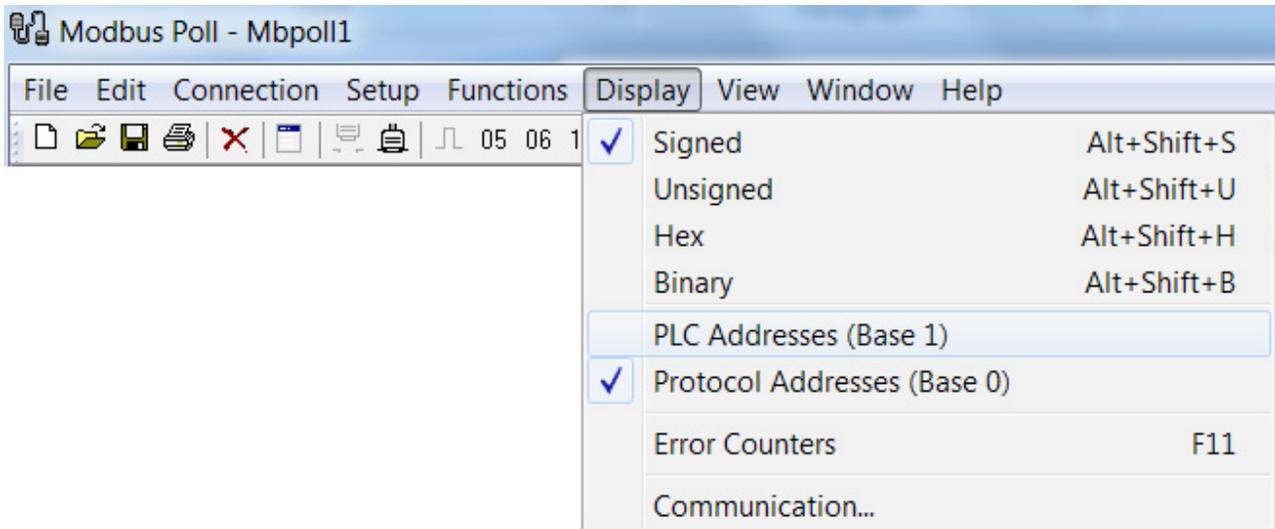
2. Then set your Com Port within Mbpoll1

10. Press "OK" to connect to your T3-Module

•If you do not need to change these settings before beginning you may simply

press the (quick connect) icon.

3.2 Instructions

Input Registers Reference Table

Set Range Register	Pulse Counting Register	Date Stamp Register
00183 = Input 1	00118 – 00119 = Input 1	00134 - 00138 = Input 1
00184 = Input 2	00120 – 00121 = Input 2	00139 - 00143 = Input 2
00185 = Input 3	00122 – 00123 = Input 3	00144 - 00148 = Input 3
00186 = Input 4	00124 – 00125 = Input 4	00149 - 00153 = Input 4
00187 = Input 5	00126 – 00127 = Input 5	00154 - 00158 = Input 5
00188 = Input 6	00128 – 00129 = Input 6	00159 - 00163 = Input 6
00189 = Input 7	00130 – 00131 = Input 7	00164 - 00168 = Input 7
00190 = Input 8	00132 – 00133 = Input 8	00169 - 00173 = Input 8

To change Input Method please read and follow these steps:

We will use Input 1 for our example

1. Connect T3 Device to Serial Port
2. Open the program
3. Double-Click register "00183"
4. This will open a pop-up window titled "Write Single Register"
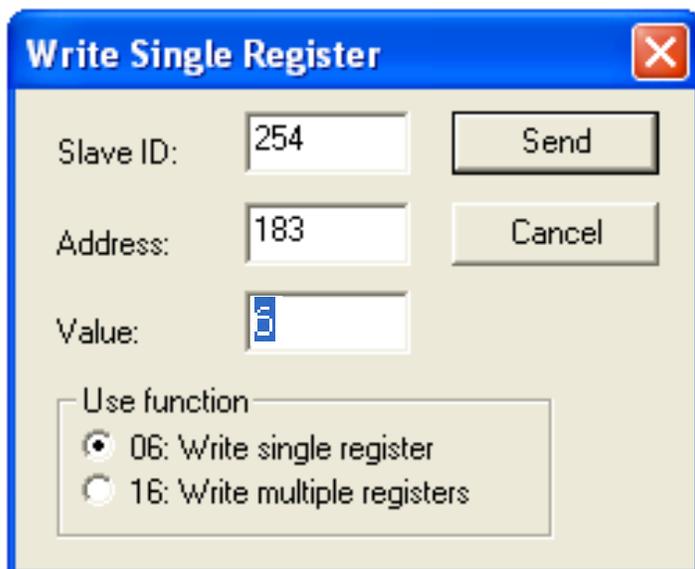
5. Set "Value" to "6" for Pulse Counting

6. Click "Send" to Confirm

7. When asked "Request OK?" - Click OK

8. Pulses will now be counted on Register 00118 and 00119

- Register 00118 counts "1" for every "65535" pulses
- Register 00119 counts single pulse

Ex. 70,000 pulses will be appear as
00118 = 1
00119 = 4465
Total Pulses = Value00118 x 65535 + Value00119

9. In order to save the data collected in Mbpoll1, the first step in using the T3-Module is to set the Date Stamp Register

CAUTION: Range Counter will reset immediately after writing date

• Before you begin counting pulses follow these instructions

1. Set Value of Register "00134" to the last 2 digits of the current year
   - Ex. 2012 = Register Value of "12"

2. Set Value of Register "00135" to the current month using 2 places
   - Ex. March = Register Value of "03"
   November = Register Value of "11"

3. Set Value of Register "00136" to the current day using 2 places
   - Ex. 5th = Register Value of "05"
   28th = Register Value of "28"

4. Set Value of Register "00137" to the current Hour using 2 places
   - Ex. 8:00 am = Register Value of "08"
   8:00 pm = Register Value of "20"

5. Set Value of Register "00138" to the current Minute using 2 places
   - Ex. 0:07 = Register Value of "07"
   0:24 = Register Value of "24"

10. To save your information simply click the   Button (or select Save As from the File Menu) and select the directory in which to save your file
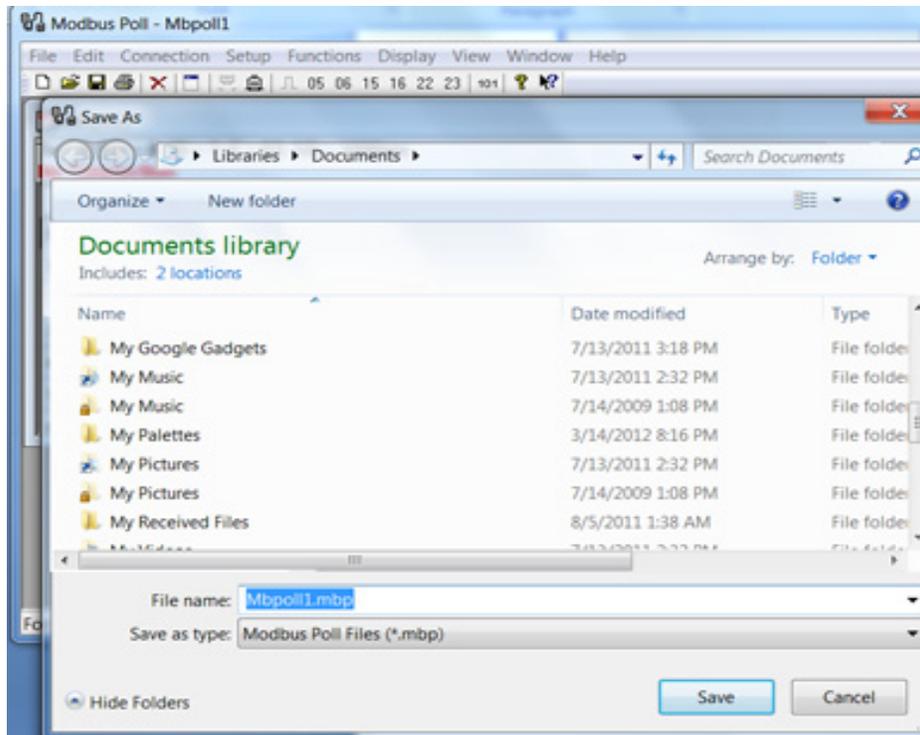
11. To clear the Range Registers (00118-00119), Right-Click Register "00134"

• Set the "Value" to any single Integer (1-999)

• This will subsequently, also act as the creation of a new testing file