Tstat5 Series

Tstat5A,B,C,D,F,G

Tstat5E

Microprocessor Based Thermostat

Datasheet Rev. 5

Table of Contents

TSTAT5

Microprocessor Based Thermostat Datasheet

Tstat5 Series

This full-featured CPU based thermostat is designed for small cooling and heating air handling systems in residential and commercial facilities. The unit provides features which eclipse standard mechanical thermostats at a price that fits conventional HVAC projects.

The unit comes in several models to suit any mechanical equipment or application.

Highlights:

-Tight control of 0.5°C provides comfortable indoor environment.

-High impact plastic enclosure provides durability in commercial environments.

-Customizable sequence of operation table (FCU with modulating or on/off valve, single or 3-speed fan, pressure independent VAV, stage sequencer…)

Technical Data

TSTAT5A.....................3 relays x 10amps @220V, 2 analog outputs 10V @ 100ma, 1 analog/digital input

TSTAT5B…................3 relays x 10amps @220V, 2 relays @ 1amp 24V or 220V, 1 analog/digital input

TSTAT5C...5 relays x 1amps @24V, 2 analog inputs, 1 digital input

TSTAT5D…....same as 5C plus two analog outputs (10V @100ma)

TSTAT5E….....................5 relays x 1amps @24V, 8 analog inputs, 2 analog outputs (10V @100ma)

Operating temperature…....... -30-70°C(-22~158°F)

Supply voltage……..12~24VAC/DC ±20%, 50-60Hz

Power consumption….......100mA at 12VDC

Relay contacts rating…................…max 6A

Ambient humidity…….............…10-90 %Rh

Operating Environment...0 ~ 99% humidity non condensing

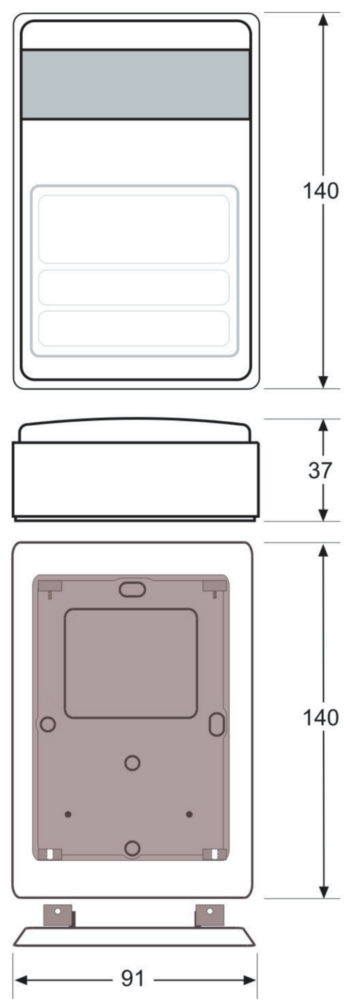
Plastic housing…... Flammability rating UL 94V0 file E194560

Enclosure rating….................................IP31

Temperature sensor...10K thermistor ±0.5°C

Colour……....…....................White/Off-white

Weight …………................….……. 200g



Standard Operation

During normal operation the display will show the current room temperature. The first hit on either of the upper pair of keys (Fig.1) will switch the display to showing the current setpoint. Subsequent hits will adjust the room setpoint up or down by 1 degree. The Tstat5 can be set to operate in degrees (Celsius or Fahrenheit) at setup time. After 10 seconds the keypad times out and the display switches back to showing the room temperature (Fig.2).

The lower pair of keys (Fig.3) allows adjustment of the off-on-auto mode and fan speed if applicable. There is flexibility in the modes that can be used. The modes are usually configured at time of installation. The current mode is shown with the first hit on the lower pair of keys and subsequent hits will adjust the mode of operation accordingly.

Fig. 1

Fig. 2

Fig. 3

Control Functionality

The controller uses PI (proportional and integral) action, in order to achieve high control accuracy.

The P action takes care of coarse offset corrections. However, when only P control is used, there will be a permanent proportional offset in the room temperature, i.e. the temperature will be kept constant – but at a higher or lower value than the setpoint. This is corrected by the built-in integral action.

The I action senses both the magnitude and the duration of any offset and can, therefore, modulate the control signal, so that any permanent offset is completely eliminated (Figure 4).

The PI parameters (proportional gain and integration time) can be set in advanced menu.

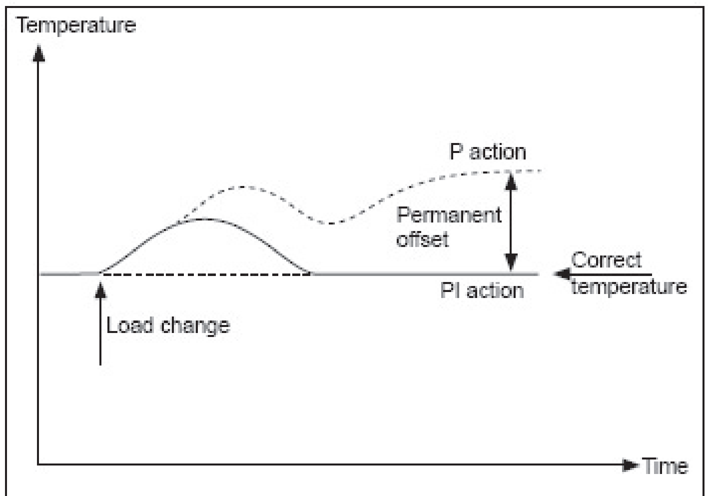


Fig. 4: PI action

Temperature Sensor

The controller monitors the temperature conditions in the room with its built-in sensor, which is located in the controller so as to not be affected by the temperature of the wall on which it is mounted.

It is possible to connect an external sensor for monitoring the temperature of different locations.

Changes in temperature are monitored continuously at the shortest time interval possible.

Installation

Terminal Block Connections (Tstat5A/B)

1 .... 24VAC live
2 .... neutral
3,4 .... 5A=0-10V, 5B=relay
5 .... External sensor input
6,7,8 .... Network communication
9 .... Common
10,11,12 .... Outputs 220VAC
(Note 3,4 Terminals:The A type has two analog outputs. On the B type, the terminals are on/off type outputs)

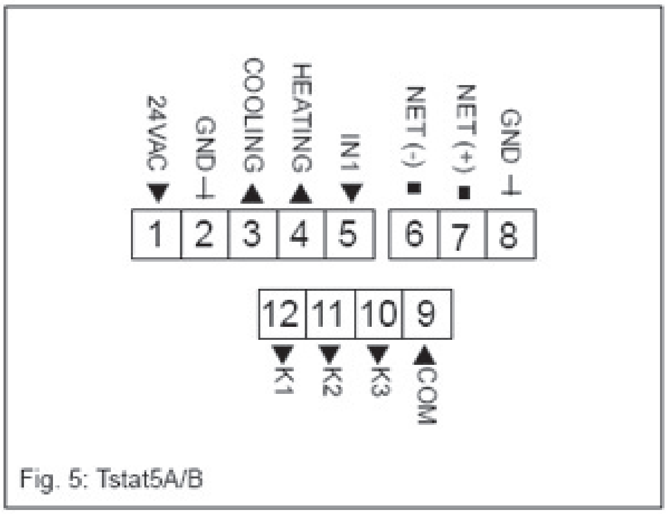


Fig. 5: Tstat5A/B

Terminal Block Connections (Tstat5C/D)

1 .... 24VAC live
2 .... neutral
3 .... Analog input 2
4 .... Analog input1
5 .... Digital input 1
6,7,8 .... Network communication
9, 10 .... 5D=0-10V, 5C=spare
11,12,13,14,15 .... Outputs 24VAC
16 .... Common

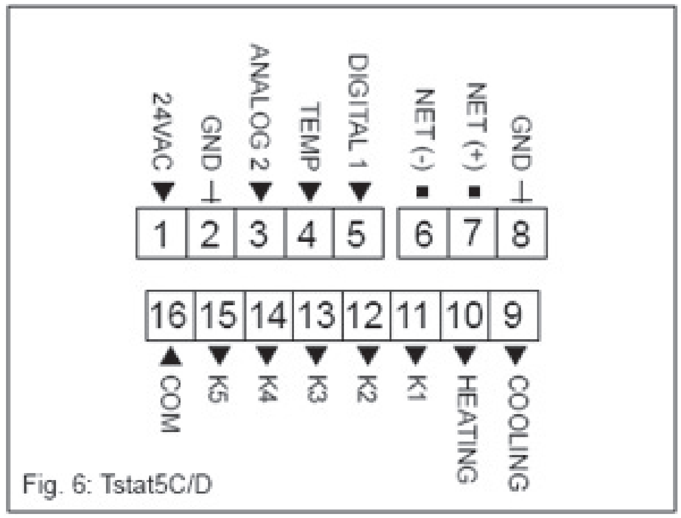


Fig. 6: Tstat5C/D

Terminal Block Connections (Tstat5E)

1 .... 24VAC live
2 .... neutral
1,2,3,4,5,6,7,8,9 .... Analog input
10,11,12 .... Network communication
13 .... Analogout2
14 .... Analogout1
15,16,17,18,19 .... Outputs 24VAC
20 .... Common

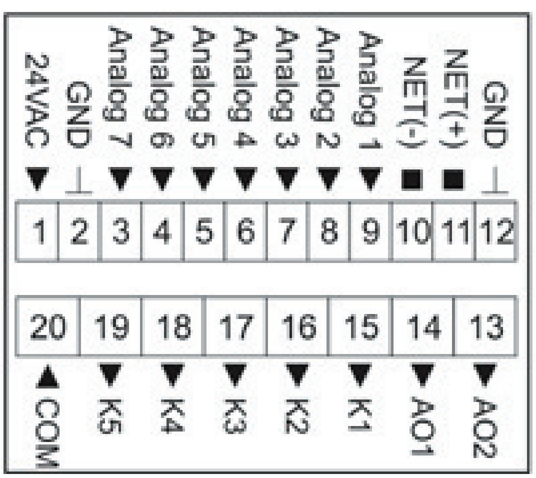


Fig.7 Tstat5E

Terminal Block Connections (Tstat5G)

1 .... 24VAC live
2 .... neutral
3 .... Analog Input1
4 .... Digital Input1
5 .... Analog Input2
6, 7, 8 .... Network Communication
9 .... Analog Output1
10 .... Analog Output2
11, 12, 13, 14, 15 .... Outputs 24VAC
16 .... Common

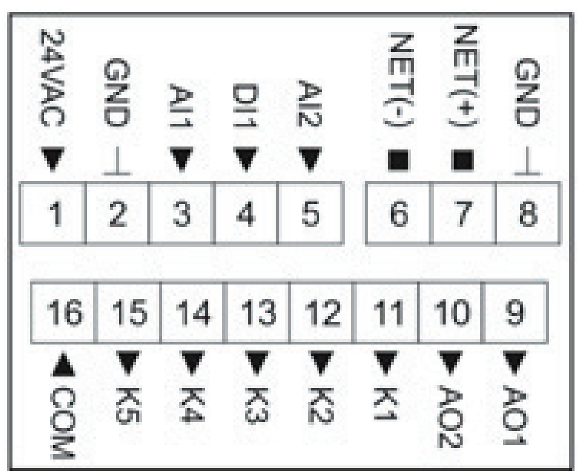


Fig.8 Tstat5G

Mounting

External wiring is connected to a terminal block on the circuit board (figures 2 and 3).

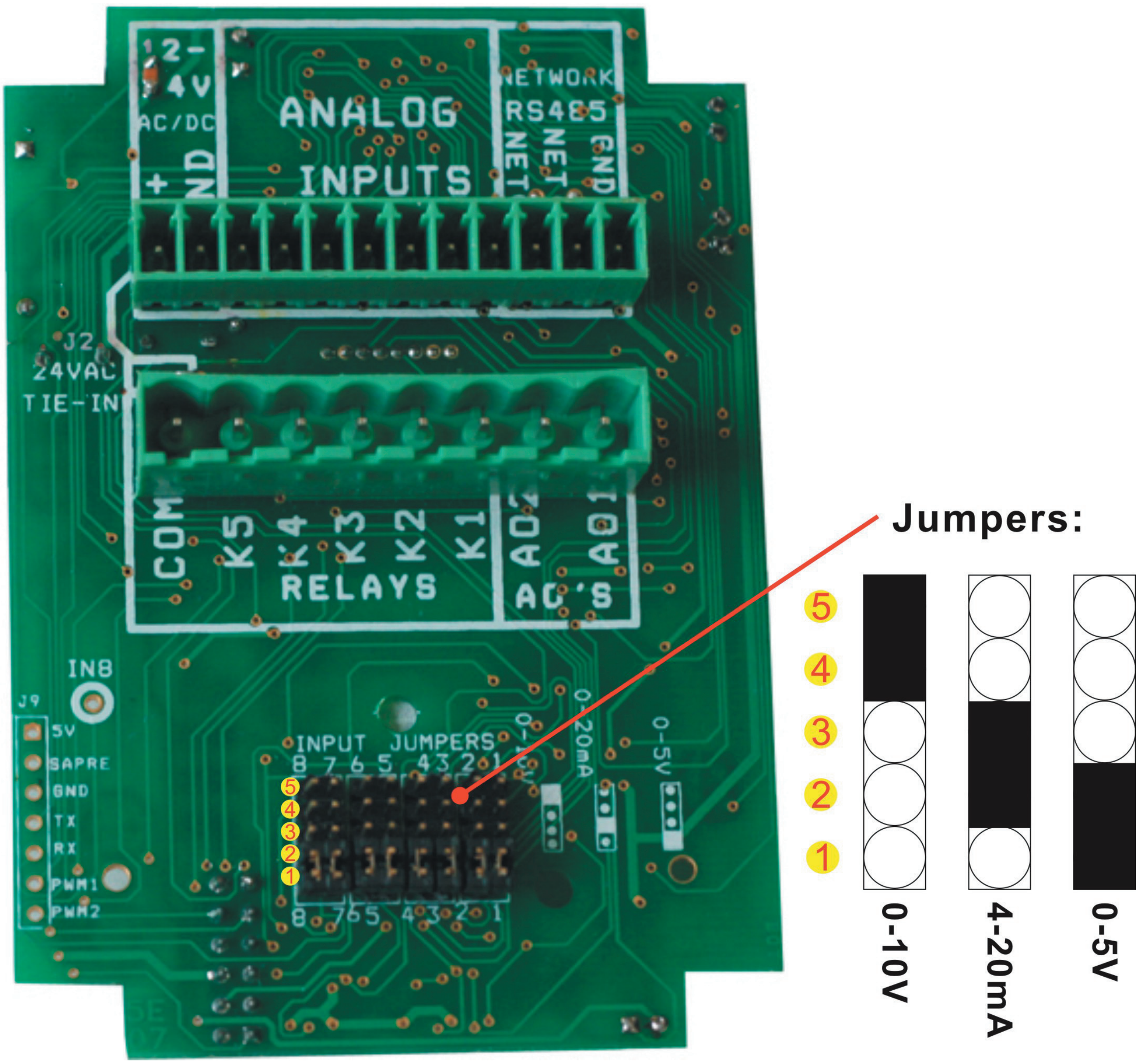
The enclosure comprises a base section and a cover. The base section can be mounted directly on a wall or on a wall box. If mounted on a wall box, the cables should enter the enclosure via the hole in the base section.

If mounted directly on a wall, the cables should enter from above.

Length of cables
There is no practical limit on the sensor cable length however accuracy of thermistors will be slightly affected on longer runs. Doing some quick calculations show that the cable would have to be tens of thousands of feet long to have an error of one DegC (2F) so there is not really any real limit to how long sensor cables can be run.

-Typical resistance values for 20 to 22 guage wire is from 10 to 20 ohms per 300m (1000ft).
-10k thermistor, at 20DegC one degree represents a change of resistance of 500 ohms
-(1000ft / 20 ohms) x (500 ohms / 1 DegC) =~ 25,000 ft to make one degree C of error.

Jumper Settings



Wiring Examples

Tstat5A - 24VAC Modulating Valves, 1-Speed Fan

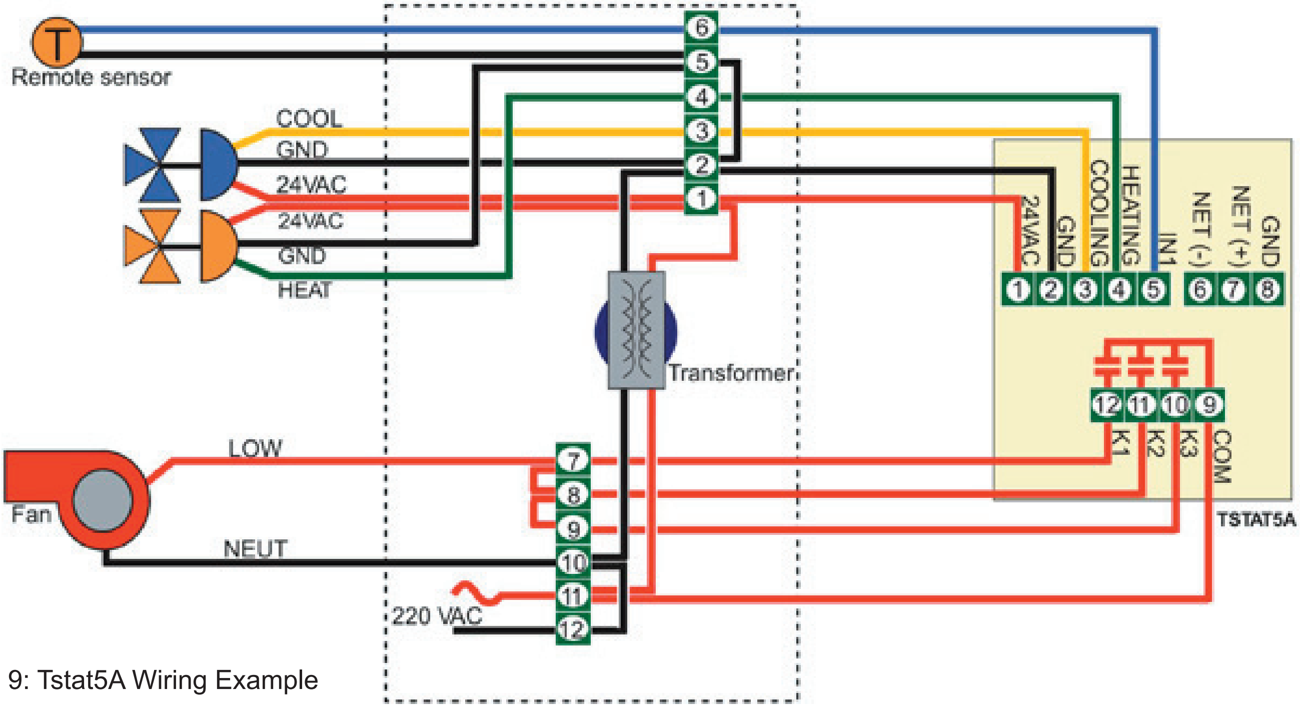


Fig. 9: Tstat5A Wiring Example

Tstat5B - 220VAC ON/OFF Valves, 3-Speed Fan

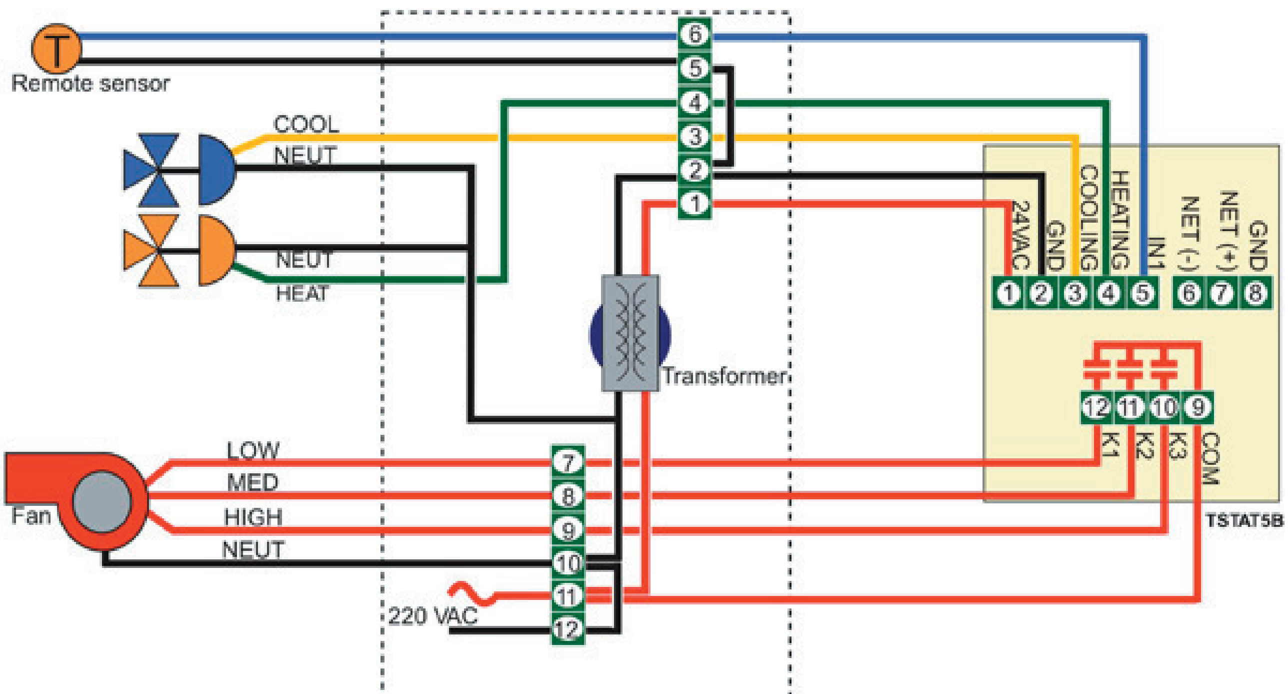


Fig. 10: Tstat5B Wiring Example

Wiring Examples (continued)

Tstat5C - 24VAC ON/OFF Valves, 1-Speed Fan

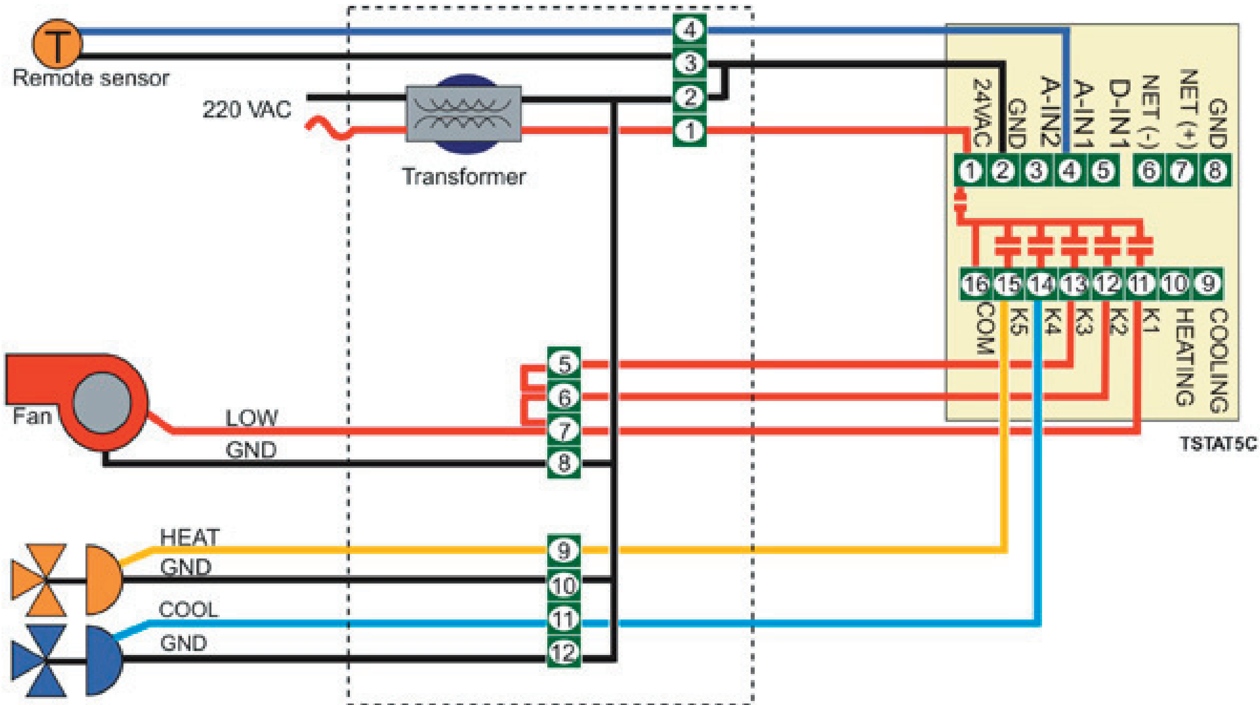


Fig. 11: Tstat5C Wiring Example

Tstat5D - 24VAC Modulating Valves, 3-Speed Fan

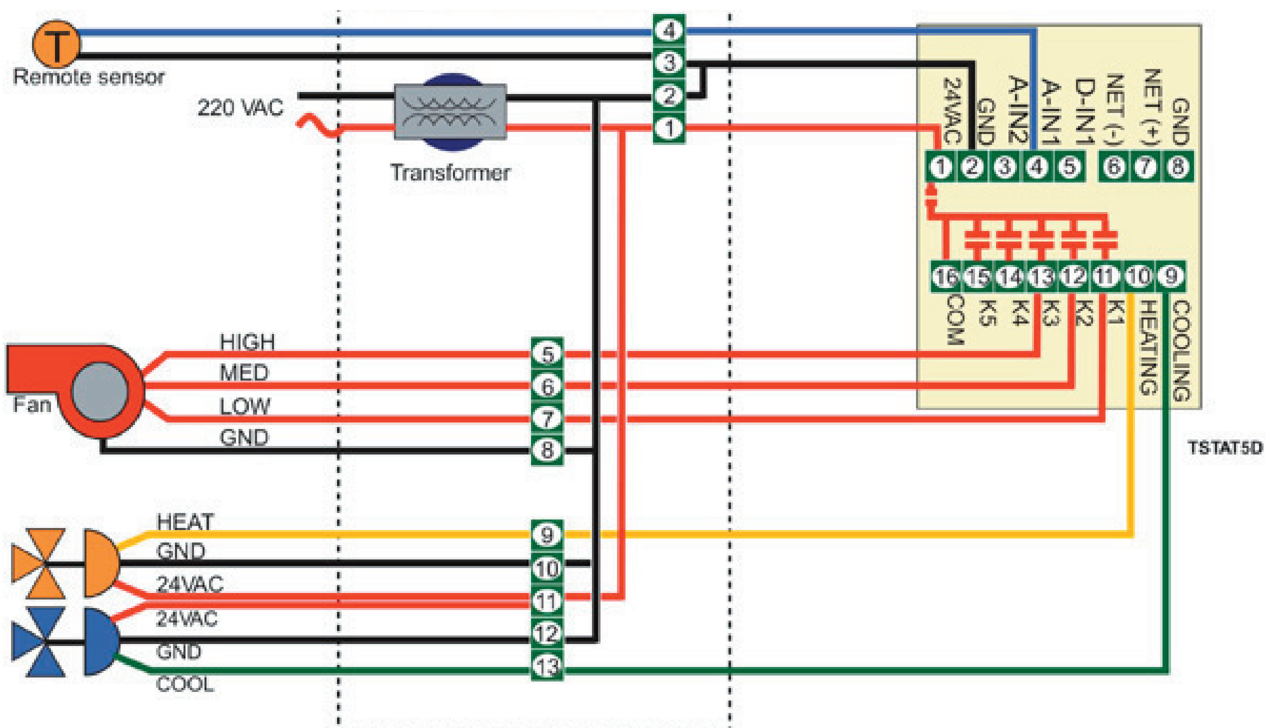


Fig. 12: Tstat5D Wiring Example

Wiring Examples (continued)

Tstat5E - 24VAC Modulating Valves, 3-Speed Fan, 8AI

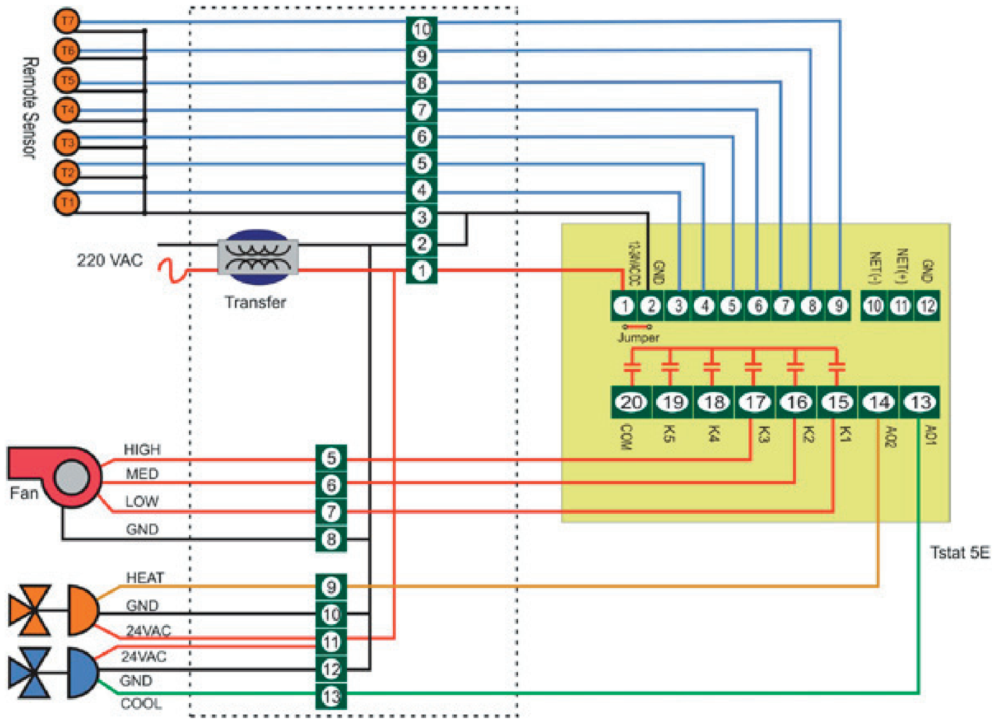


Fig. 13: Tstat5E Wiring Example

Tstat5G - 24VAC Modulating Valves, 3-Speed Fan

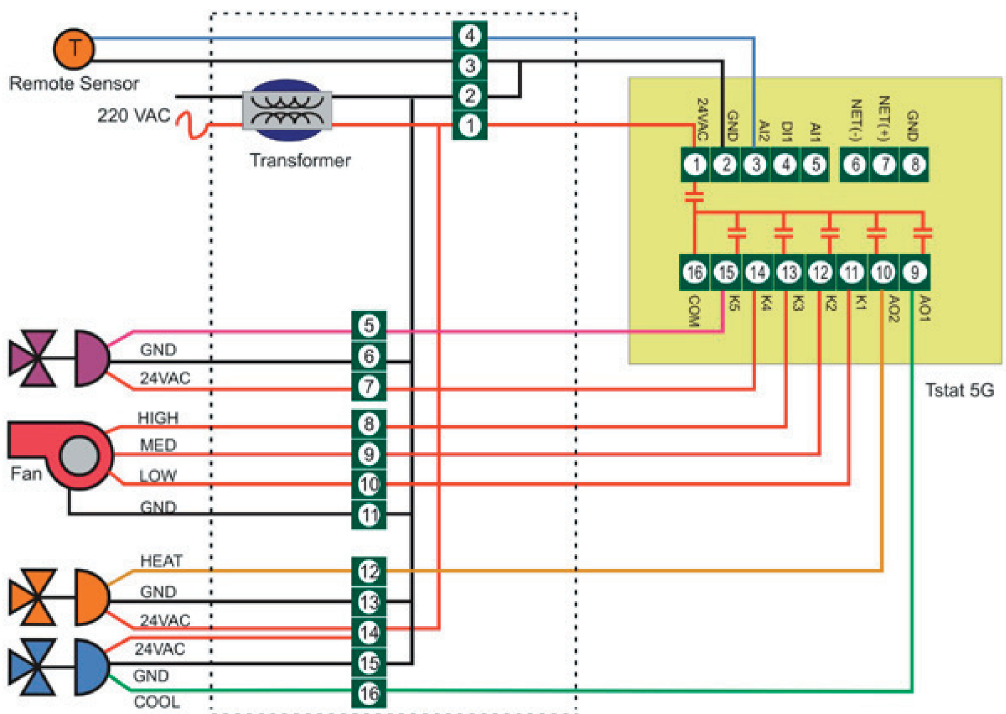


Fig. 14: Tstat5G Wiring Example

Tstat5 Quickstart Guide

Welcome to the Tstat5 Quickstart Guide. The following steps should help you quickly get connected to your new tstat. Before you begin, please verify that you have the following items: Tstat5, power adapter with network cable assembly, RS485-232 converter, Windows-based PC.

1.) Remove the Tstat, power/network cable assembly, and RS485-232 converter from the box.
2.) Attach the network cable to the RS485 side of the RS485-232 converter.
3.) Attach the power/network cable to the upper socket of the tstat by inserting the 8-pin connector.

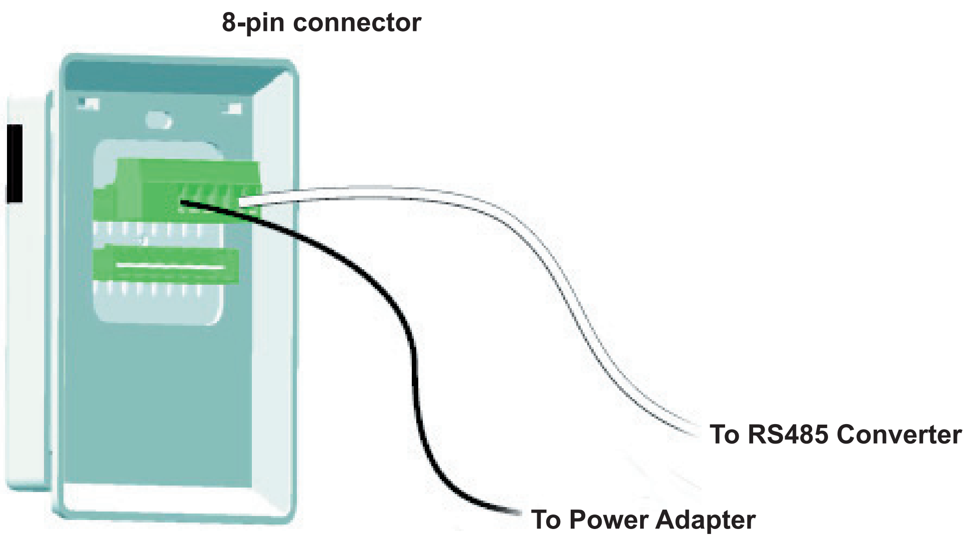


Fig. 15: Tstat5 Rear Panel

4.) Connect the other side of the adapter to the serial port of your computer. This can be done either directly, or using a serial cable.

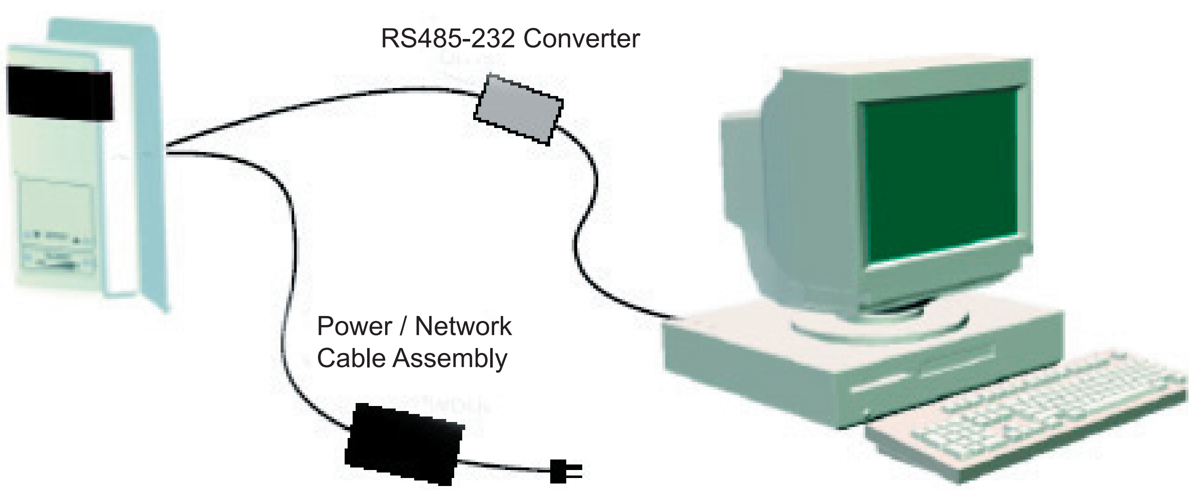


Fig. 16: Tstat5 Connections

5.) Plug the power adapter into the wall.

6.) At this point, the tstat should be powered up. It will quickly show the firmware version at startup before entering standard operating mode.

7.) To access the tstat from you computer, you must first install the Tstat Manger software which can be downloaded here: http://www.temcocontrols.com/ftp/tstat5software.zip

8.) Open the Tstat Manager software. It will automatically try to connect to the tstat using device address 255.

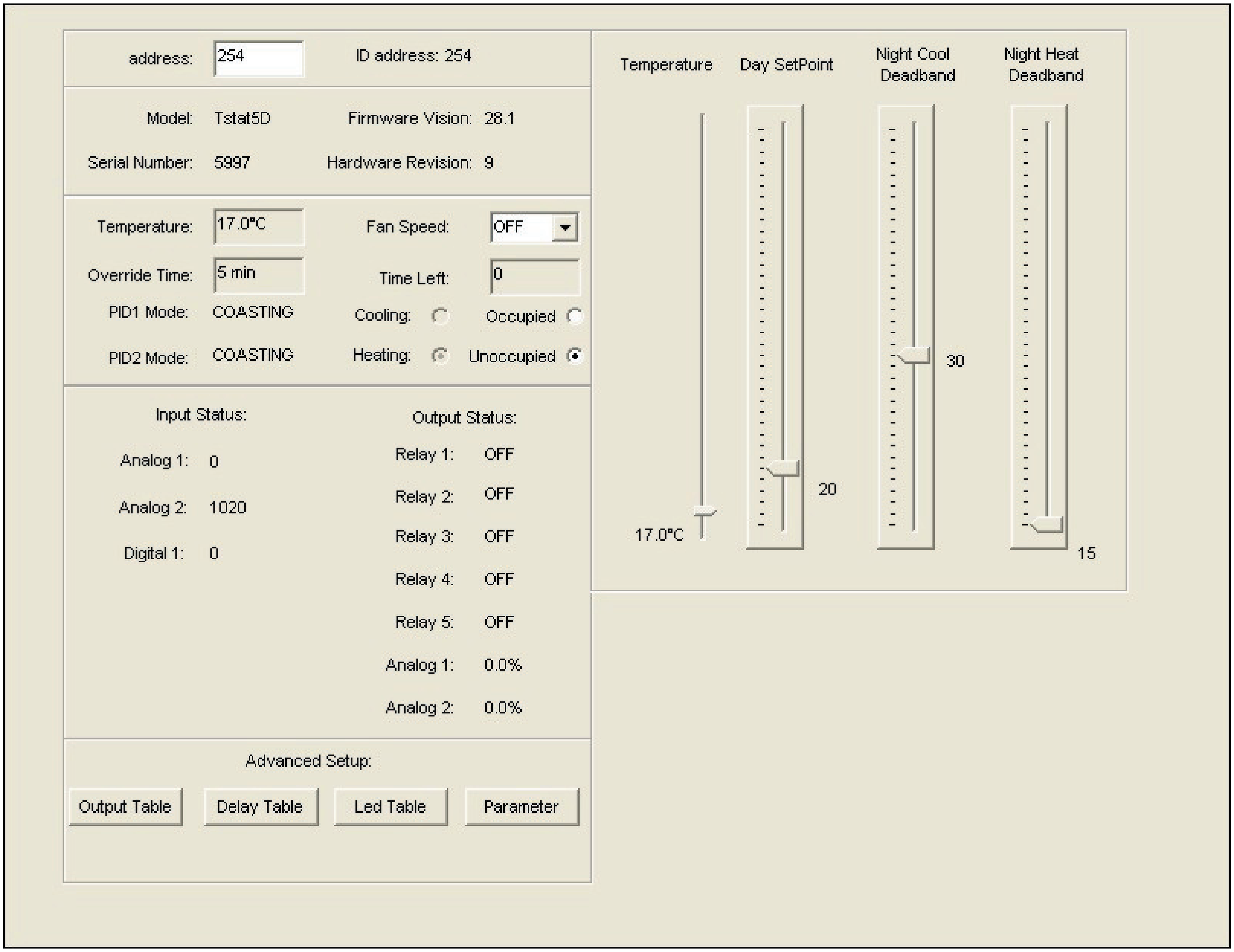


Fig. 17: Tstat Manager Interface

9.) If no tstat is found, you may need to adjust the COM port using the “COM” menu, or you may need to check that the cables and adapter are properly connected.

10.) Using the Tstat Manager software you can now communicate with your tstat and adjust the settings.

Advanced Menu and Setup Operation

Setting Up Advanced Menu Items

The Tstat5 series thermostat is a microprocessor based thermostat that can be programmed to operate according to many built in sequences of operation. The parameters can be adjusted through a menu system on the front keypad that is accessed by pressing a special combination of keystrokes. This allows a technician to make quick adjustments to the tstat without special tools, but prevents casual user from accessing the menu.

To get into the menu system, press the lower left key and the lower right key at the same time and hold them for about ten seconds until the display shows “CAL”. There are several menu items, each of which has a mnemonic or ‘shorthand name’ to help in remembering the parameters. For example, the first menu item is “Calibration” and therefore CAL is shown (Fig.14).

Once you are into the menu system, the upper pair of keys can be used to adjust the parameters up and down as required. The first hit on either of the upper pair of keys will flip the display from showing “CAL”, for example, to showing the current value. Subsequent hits on the up & down arrow keys will cause the display to flash, providing you feedback that you are in the process of adjusting the parameter (Fig.15).

There are many different keypad labeling arrangements, but the special menu functions will remain the same regardless of the labeling on the tstat5. The lower pair of keys will get you into the menu system and once you are in, you will use these keys to scroll through the various menu parameters. Similarly, the next higher pair of keys is used to adjust the particular parameter up and down (Fig.16).

When you are finished adjusting parameters, simply leave the keypad alone for approximately 10 seconds and the display will time out and switch back to normal operation showing the current room temperature. If you are in the middle of adjusting a parameter and need more time to complete the operations you can just hit the up and down key periodically to refresh the timeout function. If the display does timeout before you have completed your adjustments, you can quickly re-enter the menu system and scroll back to your particular parameter of interest. The values you set through the keypad menu system are saved in the non-volatile memory of the thermostat and are not affected by power outages.

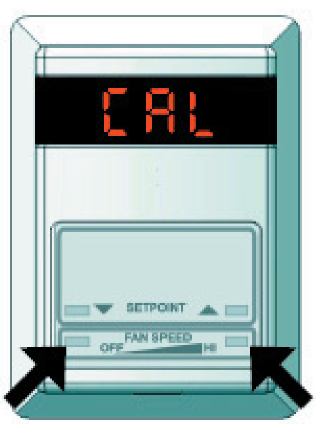


Fig. 18

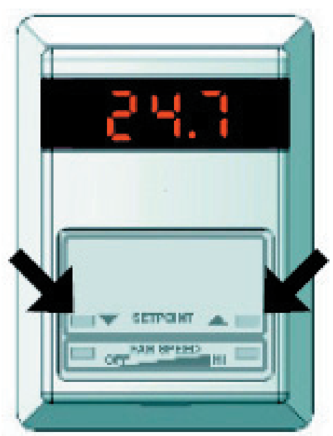


Fig. 19

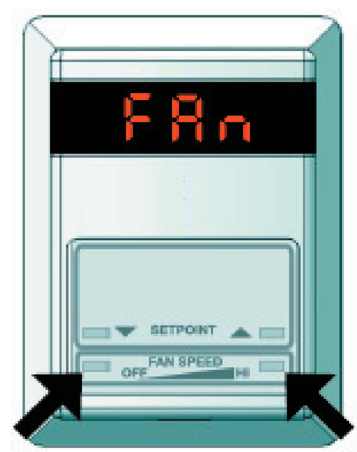


Fig. 20

Advanced Menu Item Details

The Tstat5 series thermostats have several advanced menu items which can be adjusted in the field to suit the application and tune the operation of the thermostat. Generally speaking, all the parameters are set up at the factory on an order-by-order basis and will give satisfactory results out of the box. If problems arise, see the following (Table 1) for a description of each of the menu items.

Table 1: Advanced Menu Items

Code	Description (Range, Default)
Add	**Modbus Device Address (1-254, 254)** This is the modbus address of the tstat. It is the address to which the stat will respond when receiving serial communication.
CAL	**Calibration of the Selected Temperature Sensor (0-1000, 500)** To calibrate the temperature shown on the tstat display you will need a handheld mercury thermometer or digital thermometer. Hold the meter close to the thermostat and allow it to come to equilibrium. Use the keypad to get into the menu mode until CAL is shown on the display. Now you can adjust the display using the up and down buttons till the temperature shown matches the handheld meter. When you are done, just let the display time out to normal operation, the display will stop flashing and will show the current room temperature. You can repeat this sequence if necessary till the readings on the thermostat and meter agree. The thermostat will store the calibration figures even through extended power outages and should not need to be adjusted for many years. The main point to keep in mind when calibrating is to let everything come to equilibrium. The thermostat should be powered up for 5 minutes prior to any calibration and the thermometer should be left near the thermostat for about the same amount of time. The calibration value is centered around 500 (50.0°) This means that anything above 500 will be added on to the raw temperature and anything below 500 will be subtracted from the raw temperature. Calibration units are in increments of 0.1° (i.e. 500 means 50.0°) and are in the same units (C or F) as the tstat. Some calibration tips: • The main error in calibration comes from not waiting long enough for the handheld thermometer to come to equilibrium. • Calibrate using the customer's thermometer, even if it is not an accurate one so that all subsequent measurements are compared to the same benchmark. • The sensor inside the thermostat is a digital chip capable of resolving down to 0.06°C so the weak link in calibrating is usually the procedure used rather than the tstat accuracy. • Make sure the tstat is mounted in a location free of drafts. .
tSS	**Temperature Sensor Select (0-3, 0)** The tstat has an extra input for use with an external temp sensor. tSS = 0: The tstat will use the internal temperature sensor IC for the display and PID calculations tSS = 1: The tstat will use an external thermistor which is shown on the display and used for PID calculations. tSS = 2: The tstat will use an internal thermistor which is shown on the display and used for PID calculations. tSS = 3: The tstat will use an average of internal thermistor and external thermistor which is shown on the display and used for PID calculations.
FIL	**Temperature Sensor Filter (0-10, 5)** Filter used for the raw temperature being read by the sensor. This configures the weighted average used when filtering the raw temperature. 0 corresponds to no filter. 10 corresponds to a high level of filtering. Set this to a low value if you want the input to respond quickly, a high value will smooth the readings more but make them respond more slowly.
AI1	**Analog Input 1 Range (0-4, 1)** This register controls the range of analog input 1. Default setting is 1: 10K thermistor. Actual value of analog data stored in register 180. AI1 = 0: The analog input is configured as raw 10-bit data AI1 = 1: The analog input is configured as a temperature governed by the 10K thermistor curve AI1 = 2: The analog input is configured as a percentage figure, varying from 0 to 100 over the 0-5VDC range. AI1 = 3: The analog input is configured as an on/off 1 or 0 value. AI1 = 4: The analog input is configured as a custom sensor with values calculated using the custom-built lookup table.

Code	Description (Range, Default)
AI2	**Analog Input 2 Range (0-4, 0)** This register controls the range of analog input 2. Default setting is 0: raw data. Actual value of analog data is stored in register 181. AI2 = 0: The analog input is configured as raw 10-bit data AI2 = 1: The analog input is configured as a temperature governed by the 10K thermistor curve AI2 = 2: The analog input is configured as a percentage figure, varying from 0 to 100 over the 0-5VDC range. AI2 = 3: The analog input is configured as an on/off 1 or 0 value. AI2 = 4: The analog input is configured as a custom sensor with values calculated using the custom-built lookup table.
dI1	**Digital Input 1 Function(0-5, 0)** The tstat has an extra digital input that can be used to trigger occupancy mode. dI1 = 0: The tstat will not respond to any signals on the digital input dI1 = 1:Freeze protect mode.The tstat will heat room when the ambient temperature less than freeze setpoint. dI1 = 2:Occupency sensor input.Any event from this input will reset the override timer. dI1 = 3: Use dI1 as sweep off, the tstat will go into unoccupied mode whenever this contact is closed. This is an edge triggered event so that the user can still override the time clock input at the keypad anytime after the clock contact closes. The clock input can only initiate unoccupied mode, it will not put the room into occupied mode. Only the user can put the room into occupied mode from the keypad. dI1 = 4: Use dI1 as a clock input, the tstat will go into unoccupied mode whenever this contact is closed. The tstat will return to occupied mode whenever this contact is opened. Pressing a button on the keypad will override unoccupied mode for the duration of the ORT. However, the occupancy mode itself cannot be overriden. dI1 = 5: Use dI1 as a heating/cooling mode controller.First should config the register 214 to the corresponding mode.
Ort	**Unoccupied Override Timer (0-255, 0)** This register controls the amount of time for which the unoccupied state will be overridden if a user presses a button on the tstat. It is in units of minutes. If Ort is set to 0, this means the user cannot override the unoccupied state. This function is ignored if DI1 is set to 1.
dAC	**'Digital to analog converter' analog output calibration (0-255, 100)** This feature allows the on board 'digital to analog converter' or DAC to be calibrated. It is calibrated at the factory and is not normally adjusted in the field. To calibrate the DAC, connect a meter to analog output#1, normally associated with the cooling output signal of the thermostat or relay card. Get into the 'menu mode' and hit the up down arrow keys until "DAC" is displayed. After the first hit of the up or down keys, the analog output signal will switch from normal operation to the full scale value of 10VDC. Use the up/down keys until the meter on the output signal reads 10VDC. When the thermostat times out to normal mode, the output signal wil switch back to normal operation.
bAu	**Baud Rate (0-1, 1)** This is the rate of serial communication. It can be set to either 9.6 kb/s or 19.2 kb/s.
dSC	**Short Cycle Delay (0-20, 0)** This parameter adjusts the delay between cycling of the mode of operation. It is the number of minutes after entering coasting mode until the tstat can re-enter the mode it came from. For example, if the tstat is in Cooling1 mode, and then enters Coasting mode, it will take a delay, dSC minutes, until it can re-enter into Cooling1 mode. This value is in increments of 1 min.
dCH	**Changover Delay (0-200, 0)** This parameter adjusts the delay between switching from a heating mode of operation to a cooling mode of operation or vice versa. It is the number of minutes after leaving cooling or heating mode before the tstat can enter the opposite mode. This value is in increments of 1 min.
PPr	**Proportional Term (10-255, 20)** The proportional term is the 'P' term of the familiar PID control strategy and determines how fast a valve will react to a deviation from setpoint at a particular instant in time. The default value of 2.0° (C or F) is fine for most applications, where a 2.0° deviation is required to make the valve respond 100%. For example, with the PPr term set to 2.0 (°C) and the cooling setpoint is set to 20°C, the valve will be open 100% by the time the room hits 22°C. A larger PPr term will make the valve lazy since the deviation from setpoint will have to be greater before it opens 100%. A smaller value makes the valve respond more quickly. The factory setting of 2.0° (C or F) is fine where the thermostat is located out of the direct airflow in an office size room. For a smaller room or if the thermostat is located directly under the air vent, a slower acting valve is required to avoid short cycling, so set the value of PPr to 3.0° or 4.0°. The PPr term acts in cooperation with the PIn term which is described next. The P value is in increments of 0.1° (i.e. 20 means 2.0°) and is in the same units (C or F) as the tstat.

Code	Description (Range, Default)
PIn	**Integral Term (0-255, 50)** The integral term is the ‘I’ term of the familiar PID control strategy and determines how fast a valve will react to a deviation from setpoint over time. For example with the room slightly above setpoint, the ‘P’ term may be basically satisfied, but a small deviation still exists. This deviation is summed up or ‘Integrated’ over time and the I term will gradually open the valve to make up the final small deviation from setpoint. The default value of 5.0 (%/Deg minute) is fine for most applications and will cause the valve to open 5% for one degree (C or F) of error per minute. For example, when the PIn term set to the default of 5.0 (%/Deg minute), the cooling setpoint is set to 20°C, and the room temperature is 21°C, the valve will be open partially due to the “P” term described earlier but the condition continues and we would like the valve to be opening up slowly to make up the final temperature error. If this situation of 1.0°C error continues for one minute, the error accumulates and the I term nudges the valve open an additional 5%. If the previous explanation is not clear, a couple of helpful reminders are as follows: -think of the I term as the opposite of the P term, -”a bigger I means faster valve, smaller I means lazier valve”. -The default value of 5% will work fine for most applications. -If the valve is short cycling, make the I term lazier (smaller). The I value is in increments of 0.1 %/°min (i.e. 50 means 5.0%/°min) and is in the same units (C or F) as the tstat.
SOP	**Sequence of Operations (0-2, 1)** The Sequence of operation is normally set at the factory and does not need to be adjusted. The thermostat supports field adjustment of the operation to suit different variations of mechanical equipment. Setting this value to a different value will cause the thermostat to stop working properly, so be careful not to adjust this value unless you are familiar with the various sequences. Standard Operation: When SOP is set to 1, the sequence of operations is stored in a table that allows for basically any arbitrary sequence of operation, for example the tstat could be set up to control 5 stages of cooling, 5 stages of heating, or anything in between. Each output is individually assigned to be active in any particular section of the cooling or heating cycle. There are 7 discreet steps, Heat3, Heat2, Heat1, Coasting, Cool1, Cool2 and Cool3. So the table is a 5 outputs x 7 steps spreadsheet arrangement and you fill in the blanks to suit the application. The settings can be stored in an external text file that is easily read and modified in a text editor. The “Tstat Factory” software utility on our website (http://www.temcocontrols.com/ftp/tstat5software.zip) allows you to send your favorite sequence of operations table to a new tstat speeding up the configuration process. Transducer Mode: Setting SOP to 2, puts the Tstat into transducer mode. In this mode, the cooling analog output corresponds directly to the room temperature in degrees C (i.e. at 25°C, the output would be 2.5V). The heating analog output corresponds directly to the setpoint in degrees C. And relay1 corresponds to the occupied/unoccupied mode (occupied = relay1 ON, unoccupied = relay1 OFF). Test Mode: A special sequence of operations is embedded in the tstat that assists in commissioning of the installation and testing of the tstats. When SOP is set to ‘0’ this is the testing sequence and the unit will cycle the relay outputs on and off in a slow rotation. The analog outputs are also cycled in a slow ramp, the cooling goes from 0-10V while the heating goes in reverse from 10 to 0V. The duty cycle of this rotation is approximately 20 seconds, be sure the mechanical system is able to handle this sort of cycling before using this feature.
HC	**Heating Cooling Mode Configuration (0-5, 0)** This item configures the method by which the tstat determines the heating or cooling mode. HC = 0: mode is controlled automatically by the PID. PID > 52 is heating mode, PID < 48 is cooling mode. HC = 1: mode is controlled by the keypad or serial communication. This is for keypad configurations in which the user or serial com can manually set heating or cooling. HC = 2: mode is controlled by the active *high* digital input. High is heating, low is cooling. HC = 3: mode is controlled by the active *low* digital input. High is cooling, low is heating. HC = 4: mode is controlled by difference in temperature of setpoint and analog in1 sensor. If the temperature of the sensor is greater than the setpoint, the tstat will be in cooling mode, and if the temperature of the sensor is less than the setpoint, the tstat will be in heating mode. This is primarily used for 2-pipe systems. HC = 5: same as mode 4, but using the analog in2 sensor instead of analog in1.
Hdb Cdb	**Heating & Cooling Deadbands (1-200, 10)** If there is one setpoint, the heating setpoint follows the cooling setpoint and is calculated by: Heating Setpoint = Setpoint - Heating Deadband. Cooling Setpoint = Setpoint + Cooling Deadband If there are two setpoints, heating and cooling are separately adjusted. The setpoints are calculated as follows: Heating Setpoint = Max(Cooling Setpoint + Cooling Deadband , Heating Setpoint) Cooling Setpoint = Min(Cooling Setpoint, Heating Setpoint - Cooling Deadband) The min value for Cdb is 1.0° (C or F) to ensure that simultaneous heating and cooling is never allowed. The maximum value is arbitrarily set to 20.0°. The deadband values are in increments of 0.1° (i.e. 20 means 2.0°) and are in the same units (C or F) as the tstat.

Code	Description (Range, Default)
C_F	**Degrees C/Degrees F (0-1, -)** The display can be switched to show Degrees C or Degrees F. 0 = C, 1 = F.
FAn	**Number of Fan Speeds to show on the display (0-3, 3)** The number of fan speeds allowed. Fan = 3, user will see “Off, -1-, -2-, -3-, Aut” Fan = 2, user will see “Off, -1-, -2-, Aut” Fan = 1, user will see “Off, -1- , Aut” , Fan = 0, user will see “Off, On”
nHd nCd	**Night Heating Deadband (0-35, 10) for deg C, (0-95, 10) for deg F** **Night Cooling Deadband (0-99, 10) for deg C and F** When the tstat is in unoccupied mode, and APP is set to 0, the heating setpoint is adjusted downwards by the amount of the nHd. The cooling setpoint is adjusted upwareds by the amount of nCd. The night deadband values are in increments of 1° (i.e. 10 means 10°) and are in the same units (C or F) as the tstat. Note: The night heating setpoint is prevented through an internal software interlock from being set below 5°C, regardless of the user heating setpoint and the value stored in NHS.
nHS nCS	**Night Heating Setpoint (0-255, 15) for deg C, (0-255, 65) for deg F** **Night Cooling Setpoint (0-255, 30) for deg C, (0-255, 80) for deg F** When the tstat is in unoccupied mode, and APP is set to 0, the heating and cooling setpoints are changed to these values. The night setpoint values are in increments of 1° (i.e. 20 means 20°) and are in the same units (C or F) as the tstat. Tstat is in the unoccupied mode, then 0 = unoccupied, 1 = occupied.
APP	**Application (0-1, 0)** 0 - OFFICE applications mode The night time setpoints are specified value Night Heating Setpoint = nHS value. Night Cooling Setpoint = nCS value. 1 - HOTEL or RESIDENTIAL applications mode The night time setpoints are a specified deadband in relation with the day time setpoints Night Heating Setpoint = Cooling Setpoint - nHd value. Night Cooling Setpoint = Cooling Setpoint + nCd value.
POS	**Power on setpoint (0-255, 20) for deg C, (0-255, 68) for deg F** Certain applications require the thermostat to power up with a known setpoint that is stored through a power outage. This feature is useful in some of the transducer modes where the central DDC controller can cycle the power to the thermostats to reset the room setpoints to a known value every day. The power on setpoint value is in increments of 1° (i.e. 20 means 20°) and is in the same units (C or F) as the tstat.
POn	**Power on Mode (0-3, 3)** This setting allows the thermostat to power up in one of three modes: 0 = power off, 1 = power up in on mode, 2 = last value (default), 3 = auto mode. The on and off settings are self explanatory and are useful in certain DDC applications where the central controller can cycle the power to each thermostat to sweep them off each evening for example. The default value is “last value” and will cause the thermostat to power up in whatever state it was in before the power outage.
PAd	**Number of buttons on the keypad (0-5, 1)** The keypad has up to six buttons. The setting is not normally adjusted in the field, but it does offer some flexibility to adjust the configuration to accommodate changes in the project specifications or possibly to use a tstat from one project on another in an emergency. BUT=2a(0) sets the keypad to only one pair of buttons which are used for adjustng the setpoint. BUT=2b(5) similiar to 2, except that in the 2b configuration, cooling and heating mode can only be changed via the serial interface. BUT=4a(1) sets two buttons for fan speed, and 2 for the setpoint. BUT=4b(4) sets the lower left button for cool/heat mode, the lower right button for fan speed, and the 2 middle buttons for setpoint. BUT=6A(2) sets 2 buttons for fan speed, 2 for setpoint and 2 for heating/cooling mode. BUT=6b(3) sets 2 buttons for fan speed, 2 for cooling setpoint and 2 for heating setpoint.
Aut	**Auto mode only (0-2, 0)** This setting allows the manual modes to be locked out to duplicate the operation of an auto only thermostat. Set this to “0” and the fan will be allowed to go into manual speeds. The user will see “Off, -1-, -2-, -3-, Aut”. Set this to ‘1’ to force the tstat into auto mode when switched on. The user will see “Off, On” The default setting is “0” to allow manual modes for the fan. Set this to ‘2’ to force the tstat into DDC mode,The user can not adjust setpoint and fan speed from keypad,but can adjust them from serial port.

Code	Description (Range, Default)
Ou1 Ou2	**Output settings (0-4, 0)** Sets the full-scale voltage of the analog outputs. Ou1 sets analog out 1 (Cooling). Ou2 sets analog out 2 (Heating). This setting is used to match the analog outputs to various types of actuators, transducers or other controllers. For example, by setting the output range to act over a 5VDC scale you can set the tstat up as a transducer to interface into a master DDC controller. Or perhaps you have a valve that operates over the 2-10VDC range, this 'output' type setting lets you tailor the tstat to the particular application. OuX = 0, the output will act in on/off mode. • There are 4 types of tstats. Only the Tstat5A and Tstat5CM have analog output capability. • For Tstat5B and Tstat5C, the firmware recognizes the relays and this will be permanently set to 0 and is not adjustable. • For Tstat5A and Tstat5CM with analog outputs, the output will be 0V when OFF and 10V when ON. This is useful only if you happen to have a Tstat5A or 5CM and need a couple of extra on/off outputs OuX = 1, the outputs will modulate from 0V to 10V over the 0-100% range of any particular stage of heating or cooling. OuX = 2, same as the '1' setting but the output modulates over the 0-5V scale OuX = 3, same as the '1' setting but the output modulates over the 2-10V full scale OuX = 4, same as the '1' setting but the output modulates in reverse i.e. 10V-0V Note: For a 4-20ma actuator it is simple to convert the 2-10VDC signal to a 4-20ma signal by tying in a 250 ohm resistor in series with the output and making sure the grounds of the actuator and tstat are common.
SLO SHI	**Setpoint Minimum (0-255, 15) for deg C, (0-255, 55) for deg F** **Setpoint Maximum (0-255, 50) for deg C, (0-255, 99) for deg F** Rev24: The maximum and minimum allowable user setpoint settings. The occupants cannot adjust the setpoint above or below these settings. The min and max setpoint values are in increments of 1° (i.e. 20 means 20°) and are in the same units (C or F) as the tstat. Note: the heating and cooling deadbands act in a way that reduces these settings by the amount of the deadband. For example, if the highest setpoint allowed is 'SHI' = 30°C and the heating deadband 'Hdb' = 2°C, heating will actually only be active up to 28°C. Similarly, if the 'Cdb' cooling deadband parameter is at 2°C and the minimum setpoint is at 20°C, then cooling takes place only as low as 22°C.
LOC	**Keypad lockout (0-3, 0)** Rev25 only: This setting is useful to keep the building occupants from experimenting in the menu system. When the LOC parameter is set to '1' the keypad will be locked out from all menu operations. The normal operation of the keypad is not affected; the fan and setpoint buttons work as usual. When the LOC parameter is set to '2' the keypad will be locked out from partial menu operations allowing maintenance personnel to access some of the less critical menu parameters while maintaining a LOC on functions reserved for the primary administrator. This option allows access to calibration of the internal and external temperature sensor (CAL and CAE) and the override time parameter (ORT). LOC = 3,The user can not do anything from keypad except enter menu mode.In menu mode,the user can set setpoint ,fan speed,calibration and override timer. When the menu system is locked out, the only way to adjust the tstat parameters is through the network port or through the communications jack at the bottom of the tstat. The parameter can be set back to '0' only though the communications ports as well
dIS	**Display Setting (0-6, 0)** This allows the display to be configured in a few different modes. 0 = display the room temperature. 1 = display the setpoint. 2 = blank display. 3 = display pid2 value. 4 = display pid2 setpoint. 5 = display by manually. 6 = dispaly nothing except keypad pressed.
VTT	**Valve Transient Time (10-255, 0)** This setting allows the user to adjust the valve transient time from fully open to fully closed. Value ranges from 10 (10s) to 255 (255s)
DEF	**Accept Default Setting (0-1, 0)** This allows the user to define the current Tstat configuration as the new FACtory default. Look below in Factory Default for greater details.
FAC	**Factory Defaults (0-1, 0)** This is a special variable that allows the commissioning agent to set all the settings back to the factory default settings. The factory defaults are a good starting point if the thermostat is not behaving as you'd expect, or if you are just not sure about some of the settings. When this 'FAC' setting is displayed, the up and down keys will toggle the display between "no" and "yes". Simply set the value to "yes" and leave the thermostat alone till the display stops flashing. At this point, the settings are all re-set to the factory default values and the thermostat will reset itself. Note: There are a few exceptions to this Factory Defaults restoration, the "SOP" or sequence of operations setting is not restored and also the CAL and CAE calibration values for internal and external sensors are not affected. Note: Firmware Rev24 version tstats store the sequence of operations in a separate table and this table is also not affected.

Accessing Tstat Registers Via Serial Communications

The tstat has a built-in serial interface for communication over an RS-485 network. Communication is currently implemented using Modbus Protocol. However, future versions of the tstat will work with both Bacnet and TCP/IP Protocols. For detailed information on Modbus Protocol, see the chapter entitled Modbus Serial Communications.

Connecting the Tstat to a computer

The Tstat connects to a computer serially via the RS485 interface. An RS232 to RS485 converter is required in order to communicate with a standard PC. Figure 17 shows how the Tstat should be connected to the serial port of a PC.

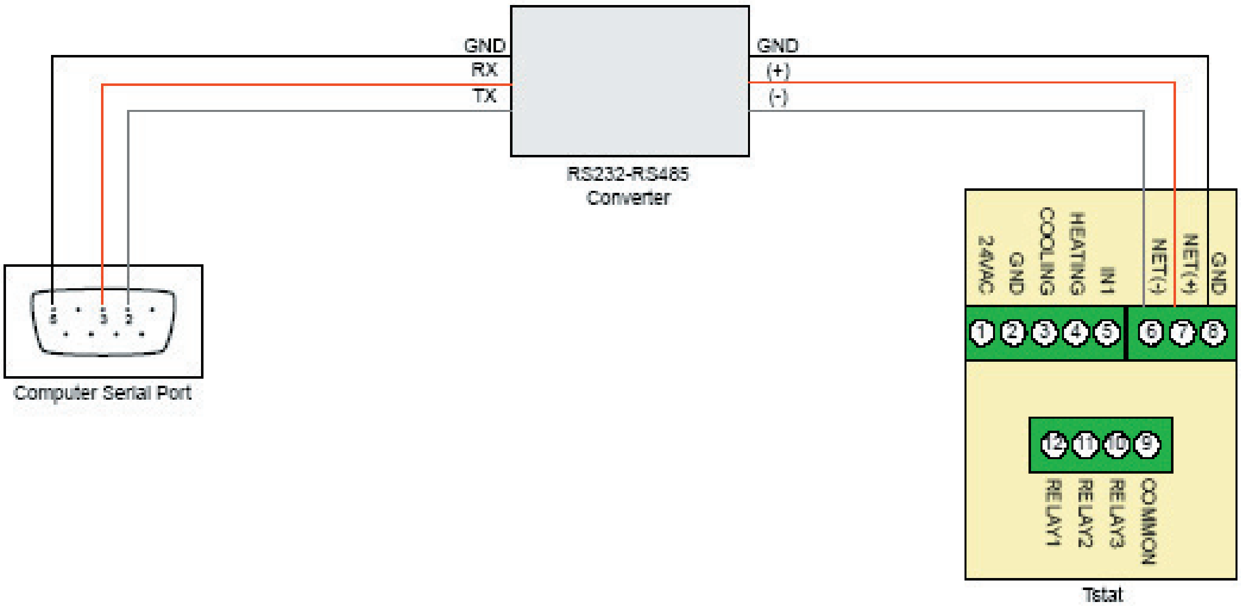


Table 2: List of registers in the Tstat Version 25

All of the registers of the tstat are accessible using the serial interface via the read and/or write commands. The following is a detailed list of all of the tstat registers.

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

Address	Bytes	Range	Defaults		Register and Description
			°C	°F	
0 to 3	4	-	-	-	Serial Number - 4 byte value. Read-only
4 to 5	2	-	-	-	Software Version – 2 byte value. Read-only
6	1	0-255	254	254	ADDRESS. Modbus device address
7	1	0-255	-	-	Product Model. This is a read-only register that is used by the microcontroller to determine the product
8	1	0-255	-	-	Hardware Revision. This is a read-only register that is used by the microcontroller to determine the hardware rev
9	1	0-255	-	-	PIC firmware version
10	1	0-255	-	-	PLUG_N_PLAY_ADDRESS, ‘plug n play’ address, used by the network master to resolve address conflicts. See VC code for algorithms
15	1	0-1	0	0	Base address selection. 0 = Protocol address, 1 = PLC address.
16	1	0-255	-	-	Update Register. Look below in ISP section
11 to 100					Blank, for future use
101	2	0-3000	-	-	ROOM TEMPERATURE reading in Deg from the sensor selected by TSS. Writing a temperature value to this register will calibrate the tstat by automatically adjusting the calibration register
102	2	0-1000	-	-	COOLING_VALVE, a number from 0-1000 representing 0% (closed) to 100% (open)
103	2	0-1000	-	-	HEATING_VALVE, a number from 0-1000 representing 0% (closed) to 100% (open)
104	2	0-100	-	-	PID, current PI calculation for cooling term
105					NOT USED FOR REV 25
106	1	0-3	-	-	COOL_HEAT_MODE, heating or cooling mode. 0=none, 1=cooling, 2=heating.
107	1	0-7	-	-	MODE_OPERATION, heating or cooling state: 0-7 = coasting, cooling 1,2,3, heating 1,2,3
108	1	0-255	-	-	DIGITAL_OUTPUT_STATE, bit 0 thru 4 = relay 1 thru 5.

109	2	0-1000	500	500	CALIBRATION, this is the calibration factor for the internal sensor, normally maintained by the tstat,
110	2	0-1000	500	500	CALIBRATION_EXTERNAL , this is the calibration factor for the external sensor, normally maintained by the tstat,
111	1	0-3	0	0	TEMP_SELECT, Sensor to be used for the PID calculations, 0 = internal sensor IC, 1= external sensor, 2 = internal thermistor, 3 = average the internal thermistor and external sensor
112	1	0-255	100	100	DAC_OFFSET , Calibration data for the 0-10VDC signal, internal variable maintained by tstat
113	1				NOT USED FOR REV 25
114	1	10-255	60	60	PTERM , proportional term for PI calculation
115	1	0-255	50	50	ITERM , integral term for PI calculation
116	1				NOT USED FOR REV 25
117	1				NOT USED FOR REV 25
118	1	0-2	1	1	SEQUENCE , control sequence ie fancoil, heatpump etc.
119	1	1-200	10	10	COOLING_DEADBAND , offset from setpoint for cooling to begin. Units of 0.1 deg.
120	1	1-200	10	10	HEATING_DEADBAND , offset from setpoint for heating to begin. Units of 0.1 deg.
121	1	0-1	0	1	DEGC_OR_F , engineering units, Deg C = 0, DegF = 1
122	1	0-3	3	3	FAN , number of fan speeds. Single speed = 1 up to three speed fan = 3
123	1	0-35 (C) 0-95 (F)	10	10	NIGHT_HEATING_DEADBAND , heating deadband in the night time or OFF mode. Units of 1 deg.
124	1	0-99	10	10	NIGHT_COOLING_DEADBAND , cooling deadband for the night (OFF) mode. Units of 1 deg.
125	1	0-1	0	0	APPLICATION , application: 0 = office, 1 = Hotel or Residential
126	1				Not used
127	1	0-3	2	2	POWERUP_MODE, mode of operation on power up. 0 = power off, 1 = power up in on mode, 2 = last value (default), 3 = auto mode.
128	1	0-5	1	1	KEYPAD_SELECT , variable to select various keypad arrangements. Refer to PAd description in Table 1: Advanced Menu Items Number of buttons on the keypad The keypad can have up to six buttons. The setting is not normally adjusted in the field. Care should be taken to coordinate with the settings in register 106, the Heat / Cool changeover parameter 128=0 , two buttons, for adjusting the setpoint. 128=1 , 4 buttons, lower pair for the mode and upper pair for the setpoint. 128=2 , 6 button keypad, with heat cool manual selection. Lower pair for the mode, next pair for the setpoint and upper pair for the heat or cool mode. 128=3 , 6 button keypad, with separate heating and cooling setpoints. Lower pair for the mode, next pair for the cooling setpoint and uppermost pair for the heating setpoint.
129	1	0-2	0	0	AUTO_ONLY , enables or disables manual mode. 0 = Manual Fan Modes 1-x Allowed (depending on R122 value, 1 = Auto Mode Only, 2 = DDC mode, the user can not change setpoint and fan speed from keypad.
130	1				NOT USED FOR REV 25
131	1	0-255	50	99	MAX_SETPOINT, Setpoint high, the highest setpoint a user will be able to set from the keypad.
132	1	0-255	15	55	MIN_SETPOINT, Setpoint Low, the lowest setpoint a user will be able to set from the keypad.
133	1	0-3	0	0	SPECIAL_MENU_LOCK, Special menu lockout via keypad, serial port only, 0 = normal, setpoint, fanspeed and full menu available. 1 = disable menu, setpoint and fan speed is available, menu not works; 2 = part menu, set point ,fan speed available, in menu mode,only can change calibration and overrid time; 3 = part menu, setpoint and fanspeed can not change outside menu, in menu mode, can change calibration , override timer, setpoint, fan speed.
134	1	0-1	0	0	FACTORY_DEFAULTS, Reset all parameters to the factory settings
135	1	MIN-MAX	20	68	COOLING_SETPOINT , current cooling setpoint - limits are set by the max and min setpoints
136					NOT USED FOR REV26
137	1	0-4	0	0	FAN_SPEED, current operating fan speed
Relay Output Tables (bit0 = relay1, bit1 = relay2, bit2 = relay3, bit3 = relay4, bit4 = relay5) Fan0 table is for the off state. Fan1, Fan2, and Fan3 are for the manual states. Fan4 is for the Auto state. These states are controlled by the user. The mode of operation (coasting, cooling, heating) is determined by the PID parameter.					
138	1	0-255	0	0	FAN0_OPERATION_TABLE_COAST
139	1	0-255	65	65	FAN0_OPERATION_TABLE_COOL1
140	1	0-255	202	202	FAN0_OPERATION_TABLE_COOL2
141	1	0-255	204	204	FAN0_OPERATION_TABLE_COOL3
142	1	0-255	65	65	FAN0_OPERATION_TABLE_HEAT1
143	1	0-255	210	210	FAN0_OPERATION_TABLE_HEAT2
144	1	0-255	212	212	FAN0_OPERATION_TABLE_HEAT3
145	1	0-255	1	1	FAN1_OPERATION_TABLE_COAST
146	1	0-255	65	65	FAN1_OPERATION_TABLE_COOL1
147	1	0-255	201	201	FAN1_OPERATION_TABLE_COOL2
148	1	0-255	201	201	FAN1_OPERATION_TABLE_COOL3
149	1	0-255	65	65	FAN1_OPERATION_TABLE_HEAT1

Address	Bytes	Range	Defaults		Register and Description
150	1	0-255	209	209	FAN1_OPERATION_TABLE_HEAT2
151	1	0-255	209	209	FAN1_OPERATION_TABLE_HEAT3
152	1	0-255	2	2	FAN2_OPERATION_TABLE_COAST
153	1	0-255	66	66	FAN2_OPERATION_TABLE_COOL1
154	1	0-255	202	202	FAN2_OPERATION_TABLE_COOL2
155	1	0-255	202	202	FAN2_OPERATION_TABLE_COOL3
156	1	0-255	66	66	FAN2_OPERATION_TABLE_HEAT1
157	1	0-255	210	210	FAN2_OPERATION_TABLE_HEAT2
158	1	0-255	210	210	FAN2_OPERATION_TABLE_HEAT3
159	1	0-255	4	4	FAN3_OPERATION_TABLE_COAST
160	1	0-255	68	68	FAN3_OPERATION_TABLE_COOL1
161	1	0-255	204	204	FAN3_OPERATION_TABLE_COOL2
162	1	0-255	204	204	FAN3_OPERATION_TABLE_COOL3
163	1	0-255	68	68	FAN3_OPERATION_TABLE_HEAT1
164	1	0-255	212	212	FAN3_OPERATION_TABLE_HEAT2
165	1	0-255	212	212	FAN3_OPERATION_TABLE_HEAT3
Address	**Bytes**	**Range**	**Defaults**		**Register and Description**
166	1	0-255	0	0	FANAUT_OPERATION_TABLE_COAST
167	1	0-255	65	65	FANAUT_OPERATION_TABLE_COOL1
168	1	0-255	202	202	FANAUT_OPERATION_TABLE_COOL2
169	1	0-255	204	204	FANAUT_OPERATION_TABLE_COOL3
170	1	0-255	65	65	FANAUT_OPERATION_TABLE_HEAT1
171	1	0-255	210	210	FANAUT_OPERATION_TABLE_HEAT2
172	1	0-255	212	212	FANAUT_OPERATION_TABLE_HEAT3
	Analog Output Tables (bit0,1 = analog out1, bit2, 3 = analog out2, 00 = 0%, 01 = 0-100%, 11 = 100%)				
173	1	0-255	0	0	VALVE_OPER_TABLE_COAST, Analog output state for each of the 8 modes of operation
174	1	0-255	1	1	VALVE_OPER_TABLE_COOLING1
175	1	0-255	3	3	VALVE_OPER_TABLE_COOLING2
176	1	0-255	3	3	VALVE_OPER_TABLE_COOLING3
177	1	0-255	4	4	VALVE_OPER_TABLE_HEATING1
178	1	0-255	4	4	VALVE_OPER_TABLE_HEATING2
179	1	0-255	12	12	VALVE_OPER_TABLE_HEATING3
180	2	0-65535	-	-	External Sensor 1 - Filtered, calibrated value for analog in 1
181	2	0-65535	-	-	External Sensor 2 - Filtered, calibrated value for analog in 2
182	1	0-255	15	65	Night heating setpoint
183	1	0-255	30	80	Night cooling setpoint
184	1	0-255	1	1	Bit 0 is read/write and shows the occupancy mode. Bit 0 = 0 means unoccupied. Bit 0 = 1 means occupied. Bit 1 is read only and shows the reset state. Bit 1 = 0 means hardware restart. Bit 1 = 1 means software restart. Bit 2 is read/write and is the reset prevention bit. Bit 2 = 0 means the tstat will automatically reset after certain registers are changed. Bit 2 = 1 prevents this reset. Changing this bit from 1 to 0 will trigger a reset. Bit 3 is the state of the digital input. Bit 3 = 1 means logic high. Bit 3 = 0 means logic low. Bit 4,5: Reserved, used for some non standard occupancy sensor logic. Bit6 0=no delay on modbus reply, 1= 10ms delay before send for slower PLC's to switch from TX to RX. Bit7, RS485/wireless communications mode: The normal communications method is a bus topology using RS485 which uses a 'transmit enable' or TX_EN line on the RS485 hardware whenever transmission from the thermostat to the bus takes place. For wireless devices this is typically taken care of by the radio module itself so it is not needed. Default = 0, When bit7 is 0, the RS485 chip, TX_EN line is used for normal RS485 bus communications. When bit7 is 1, the TX_EN line is not used, allowing the radio module to communicate one-to-one with the Tstat
185	1	0-1	1	1	BAu - Baudrate
186	1	0-4	0	0	Ou1 - Output1 Scale - 0=On/Off, 1=0-10V, 2=0-5V, 3=2-10V, 4= 10-0V
187	1	0-4	0	0	Ou2 - Output2 Scale - 0=On/Off, 1=0-10V, 2=0-5V, 3=2-10V, 4= 10-0V
188	1	0-4	1	1	AI1 – Analog input 1 range 0=10-bit raw data, 1=10K thermistor, 2=0-100%, 3 = on/off, 4= custom
189	1	0-4	0	0	AI2 – Analog input 2 range 0=10-bit raw data, 1=10K thermistor, 2=0-100%, 3 = on/off, 4= custom
190	1	0-0	0	0	dI1 – Digital input 1 range 0 = ON/OFF.
191	1	0-255	0	0	OUTPUT1_DELAY_OFF_TO_ON – delay time for output1 going from OFF to ON (sec)
192	1	0-255	0	0	OUTPUT2_DELAY_OFF_TO_ON – delay time for output2 going from OFF to ON (sec)
193	1	0-255	0	0	OUTPUT3_DELAY_OFF_TO_ON – delay time for output3 going from OFF to ON (sec)
194	1	0-255	0	0	OUTPUT4_DELAY_OFF_TO_ON – delay time for output4 going from OFF to ON (sec)
195	1	0-255	0	0	OUTPUT5_DELAY_OFF_TO_ON – delay time for output5 going from OFF to ON (sec)
196	1	0-255	0	0	OUTPUT1_DELAY_ON_TO_OFF – delay time for output1 going from OFF to ON (sec)
197	1	0-255	0	0	OUTPUT2_DELAY_ON_TO_OFF – delay time for output2 going from OFF to ON (sec)
198	1	0-255	0	0	OUTPUT3_DELAY_ON_TO_OFF – delay time for output3 going from OFF to ON (sec)

199	1	0-255	0	0	OUTPUT4_DELAY_ON_TO_OFF – delay time for output4 going from OFF to ON (sec)
200	1	0-255	0	0	OUTPUT5_DELAY_ON_TO_OFF – delay time for output5 going from OFF to ON (sec)
201	1	0-20	0	0	MODBUS_CYCLING_DELAY – delay time (in minutes) for switching out of heating or cooling and then back in.
202	1	0-200	0	0	MODBUS_CHANGOVER_DELAY – delay time (in minutes) for switching from cooling into heating or vice versa.
203	1	0-6	0	0	dIS – Display. This sets the display to either room temperature or setpoint. 0 = room temp, 1 = setpoint, 2 = Blank Display, 3 = PID2 value, 4 = PID2 setpoint, 5 = set segment code by manually, 6 = Display sleep
					LED TABLE
					Determines what activates the LEDs
204	1	0-24	3	3	LED1 (top left to bottom right)
205	1	0-24	2	2	LED2
206	1	0-24	1	1	LED3
207	1	0-24	15	15	LED4
208	1	0-24	13	13	LED5
209	1	0-24	14	14	LED6
210	1	0-24	19	19	LED7

0 = NONE	14 = HEATING MODE
1 = RELAY1	15 = FAN OFF
2 = RELAY2	16 = FAN LOW
3 = RELAY3	17 = FAN MED
4 = RELAY4	18 = FAN HI
5 = RELAY5	19 = FAN AUTO
6 = COASTING STAGE	20 = HEAT1 OR COOL1
7 = COOLING1 STAGE	21 = HEAT2 OR COOL2
8 = COOLING2 STAGE	22 = HEAT3 OR COOL3
9 = COOLING3 STAGE	23 = COOL1,2, OR 3
10 = HEATING1 STAGE	24 = HEAT1, 2, OR 3
11 = HEATING2 STAGE	25 = OCCUPANCY
12 = HEATING3 STAGE	26 = STAGE 1,2 OR 3
13 = COOLING MODE	27 = STAGE 2 OR 3

211	1	0-255	0	0	Unoccupied Override Timer, Ort. 0=disabled, not 0=number of minutes manual override is allowed
212	1	0-255	-	-	OVERRIDE_TIMER_DOWN_COUNT - Number of minutes remaining on the timer when unoccupied override timer is in effect.
213	1	0-100	20	20	Temperature sensor filter, FIL, Set this to a low value if you want the input to respond quickly, a high value will smooth the readings more but make them respond more slowly.
214	1	0-5	0	0	Heating cooling mode configuration, HC, 0=PID, 1=Keypad, 2=Digital_in1, 3=!Digital_in1, 4=Analog_in1, 5=Analog_in2
215	2	0-3000	-	-	Internal Temperature Sensor IC - Shows the filtered, calibrated value of the internal temperature sensor IC
216	2	0-3000	-	-	Internal Thermistor Sensor - Shows the filtered, calibrated value of the internal thermistor sensor
217	2	0-1000	500	500	Calibration Internal Thermistor - Calibration value used on the internal thermistor
218	2	0-1000	500	500	Calibration Analog Input2 - Calibration value used on the analog input 2
219	2	0-65535	-	-	Lookup Table 1 - 0.0V value Sensor value that corresponds to 0.0V
220	2	0-65535	-	-	Lookup Table 1 - 0.5V value Sensor value that corresponds to 0.5V
221	2	0-65535	-	-	Lookup Table 1 - 1.0V value Sensor value that corresponds to 1.0V
222	2	0-65535	-	-	Lookup Table 1 - 1.5V value Sensor value that corresponds to 1.5V
223	2	0-65535	-	-	Lookup Table 1 - 2.0V value Sensor value that corresponds to 2.0V
224	2	0-65535	-	-	Lookup Table 1 - 2.5V value Sensor value that corresponds to 2.5V
225	2	0-65535	-	-	Lookup Table 1 - 3.0V value Sensor value that corresponds to 3.0V
226	2	0-65535	-	-	Lookup Table 1 - 3.5V value Sensor value that corresponds to 3.5V
227	2	0-65535	-	-	Lookup Table 1 - 4.0V value Sensor value that corresponds to 4.0V
228	2	0-65535	-	-	Lookup Table 1 - 4.5V value Sensor value that corresponds to 4.5V
229	2	0-65535	-	-	Lookup Table 1 - 5.0V value Sensor value that corresponds to 5.0V
230	2	0-65535	-	-	Lookup Table 2 - 0.0V value Sensor value that corresponds to 0.0V
231	2	0-65535	-	-	Lookup Table 2 - 0.5V value Sensor value that corresponds to 0.5V
232	2	0-65535	-	-	Lookup Table 2 - 1.0V value Sensor value that corresponds to 1.0V
233	2	0-65535	-	-	Lookup Table 2 - 1.5V value Sensor value that corresponds to 1.5V
234	2	0-65535	-	-	Lookup Table 2 - 2.0V value Sensor value that corresponds to 2.0V
235	2	0-65535	-	-	Lookup Table 2 - 2.5V value Sensor value that corresponds to 2.5V
236	2	0-65535	-	-	Lookup Table 2 - 3.0V value Sensor value that corresponds to 3.0V
237	2	0-65535	-	-	Lookup Table 2 - 3.5V value Sensor value that corresponds to 3.5V
238	2	0-65535	-	-	Lookup Table 2 - 4.0V value Sensor value that corresponds to 4.0V
239	2	0-65535	-	-	Lookup Table 2 - 4.5V value Sensor value that corresponds to 4.5V
240	2	0-65535	-	-	Lookup Table 2 - 5.0V value Sensor value that corresponds to 5.0V
241	2	0-2	-	-	Universal PID input select, 0=none, 1=analog_in1, 2=analog_in2
242	2	0-65535	-	-	Universal PID upper deadband
243	2	0-65535	-	-	Universal PID lower deadband
244	2	0-65535	-	-	Universal PID pterm
245	2	0-65535	-	-	Universal PID iterm
246	2	0-65535	-	-	Universal PID setpoint
247	1	0-3	-	-	Output 1 PID Control 0 = PID1

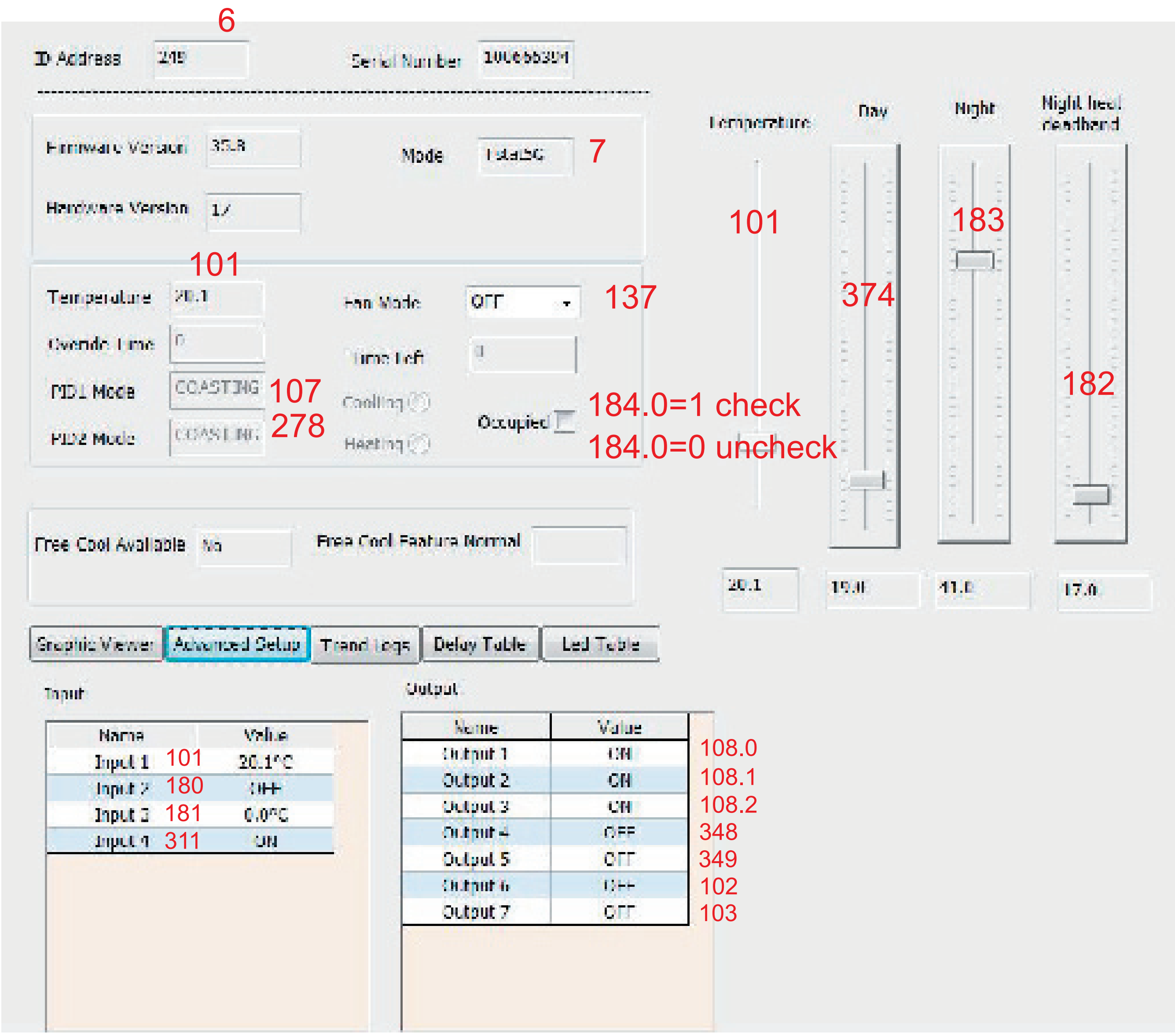
248	1	0-3	-	-	Output 2 PID Control	1 = PID2
249	1	0-3	-	-	Output 3 PID Control	2 = Maximum of PID1 and PID2
250	1	0-3	-	-	Output 4 PID Control	3 = Minimum of PID1 and PID2
251	1	0-3	-	-	Output 5 PID Control	
252	1	0-3	-	-	Output 6 PID Control	
253	1	0-3	-	-	Output 7 PID Control	
254	1	0-255	-	-	Universal PID Output - Coasting	
255	1	0-255	-	-	Universal PID Output - Cooling1	
256	1	0-255	-	-	Universal PID Output - Cooling2	
257	1	0-255	-	-	Universal PID Output - Cooling3	
258	1	0-255	-	-	Universal PID Output - Heating1	
259	1	0-255	-	-	Universal PID Output - Heating2	
260	1	0-255	-	-	Universal PID Output - Heating3	
Analog Output Tables (bit0, 1 = analog out1, bit2, 3 = analog out2, 00 = 0%, 01 = 0-100%, 11 = 100%)						
261	1	0-255	-	-	Universal PID Valve Output - Coasting	
262	1	0-255	-	-	Universal PID Valve Output - Cooling1	
263	1	0-255	-	-	Universal PID Valve Output - Cooling2	
264	1	0-255	-	-	Universal PID Valve Output - Cooling3	
265	1	0-255	-	-	Universal PID Valve Output - Heating1	
266	1	0-255	-	-	Universal PID Valve Output - Heating2	
267	1	0-255	-	-	Universal PID Valve Output - Heating3	
268	1	0-6	3	3	Number of Heating Stages in Universal Table-(Maximum # of total heating and cooling states is 6)	
269	1	0-6	3	3	Number of Cooling Stages in Universal Table-(Maximum) # of total heating and cooling states is 6)	
270	1	0-100	-	-	Universal PID	
271	2	0-65535	-	-	PID1 Units High byte - Upper 2 bytes of the PID1 units in ASCII	
272	2	0-65535	-	-	PID1 Units Low byte - Lower 2 bytes of the PID1 units in ASCII	
273	2	0-65535	-	-	PID2 Units High byte - Upper 2 bytes of the PID2 units in ASCII	
274	2	0-65535	-	-	PID2 Units Low byte - Lower 2 bytes of the PID2 units in ASCII	
275	2	0-65535	-	-	Universal Night Setpoint	
276	1	0-6	3	3	Number of Heating Stages in Original Table - (Maximum # of total heating and cooling states is 6)	
277	1	0-6	3	3	Number of Cooling Stages in Original Table - (Maximum # of total heating and cooling states is 6)	
278	1	0-19	-	-	PID2 heating or cooling state. 0=coasting, 1=cooling1, 2=cooling2, 3=cooling3, 4=heating1, 5=heating2, 6=heating3, 14=cooling4, 15=cooling5, 16=cooling6, 17=heating4, 18=heating5, 19=heating6.	
279	1	10-255	-	-	Valve travel time. The time of the valve travel from one end to another end. The units is sencond.	
280	1	0	0	0	Determine the output1 mode. Output1 always is ON/OFF mode	
281	1	0	0	0	Determine the output2 mode. Output2 always is ON/OFF mode	
282	1	0	0	0	Determine the output3 mode. Output3 always is ON/OFF mode	
283	1	0-1	0	0	Determine the output4 mode. 0, ON/OFF mode; 1, floating valve for cooling; 2, lighting control; 3, PWM	
284	1	0-1	0	0	Determine the output5 mode. 0, ON/OFF mode; 1, floating valve for heating; 2, lighting control; 3, PWM	
285	1	0-100	-	-	Valve percent. Show the valve opened how much percent. READ ONLY	
Interlock for each output, analog and digital output. 0, interlock always ON; 1, DI1 determine the interlock status; 2, AI1 determine the interlock status, the range of AI1 must be ON/OFF; 3, AI2 determine the interlock status, the range of AI2 must be ON/OFF; 4, TIMER OR, the output OR with the period timer; 5, TIMER AND, the output AND with the period timer.						
286	1	0-5	0	0	Interlock for output1	
287	1	0-5	0	0	Interlock for output2	
288	1	0-5	0	0	Interlock for output3	
289	1	0-5	0	0	Interlock for output4	
290	1	0-5	0	0	Interlock for output5	
291	1	0-5	0	0	Interlock for output6	
292	1	0-5	0	0	Interlock for output7	
293	1	1-10	10	10	Setpoint increment. The value is expanded 10 times, the increment is from 0.1 to 1.	
294	2	0-65535	0	0	Last key pressed counter. Hong long time past since the last key pressed. Reset if any key is pressed. The units is minute.	
295	1	0-255	5	40	Freeze protect setpoint. If the ambient temperature less than the setpoint, the heating valve will open some time the Delay to off register set .	
296	1	5-255	10	10	Delay to open. The heating valve will open if the ambient temp less than the Freeze temp setpoint last the time this register set .The units is second.	
297	1	5-255	30	30	Delay to close. The duration the heating valve open. The units is minute.	

298	1	0-5	0	0	Analog input1 function selection. 0, normal; 1, freeze protect sensor input; 2, occupancy sensor input; 3, sweep off mode; 4, clock mode; 5, change over mode. Refer to dI1 on page13.
299	1	0-5	0	0	Analog input2 function selection. 0, normal; 1, freeze protect sensor input; 2, occupancy sensor input; 3, sweep off mode; 4, clock mode; 5, change over mode. Refer to dI1 on page13.
300	1	0-5	0	0	dI1 – Digital input 1 function. Refer to dI1 description in Table 1: Advanced Menu Items
301	2	0-65535	0	0	Period timer ON time.
302	2	0-65535	0	0	Period timer OFF time.
303	1	0-2	1	1	Period timer units. 0, second; 1, minute; 2, hour.
304	1	0-255	-	-	Keypad encode value. The reverse value read from P0 port when some key is pressed. READ ONLY
305	1	0-255	-	-	LED hundred's segment code. Drive the LEDs by manually, the register 203 (display) must be set 5.
306	1	0-255	-	-	LED ten's segment code. Drive the LEDs by manually, the register 203 (display) must be set 5.
307	1	0-255	-	-	LED digital's segment code. Drive the LEDs by manually, the register 203 (display) must be set 5.
308	1	0-255	-	-	LED status's segment code. Drive the LEDs by manually, the register 203 (display) must be set 5.
309	1	0-255	0	0	Input auto/ manual enable. Bit0 correspond to analog input1 (register 180); bit1 to analog input2 (register 181); bit2 to digital input1 (register 311). 0, auto mode, the corresponding input value from sensor; 1, manual mode, the corresponding value from serial port.
310	1	0-255	0	0	Output auto/manual enable. Bit 0 to 4 correspond to output1 to output5, bit 5 correspond to output6 (register 102), bit 6 correspond to output7 (register 103). 0, auto mode; 1, manual mode.
311	1	0-1	1	1	Digital manual input. Write the manual value for digital input when digital input in manual mode.
312	1	0-1	0	0	Output1 manual input.
313	1	0-1	0	0	Output2 manual input.
314	1	0-1	0	0	Output3 manual input.
315	1	0-1	0	0	Output4 manual input.
316	1	0-1	0	0	Output5 manual input.
317	1	0-255	0	0	Dead master. The Tstat will go to occupied mode automatically after the time set in the register no serial communication since power on. 0, disable the function. the units is minute.
318	1	0-5	1	1	Rouding display. 0, round the display to digit; 1, round the display to the nearest 1/10 unit; 5, round the display to the nearest 1/2 unit. 2, 3, 4 reserved.
319	1	0-1	0	0	Timer selection. 0. Period timer; 1, Rotation timer.
320	1	0-255	0	0	Rotation flag group. Put the corresponding output into rotation group by setting the corresponding bit. bit 0 correspond to output2; bit1 correspond to output3; bit2 correspond to output4; bit3 correspond to output5. 0, not put into rotation group; 1, put into rotation group. The rotation group at least has two items and they must be consecutive.
321	1	1-4	-	-	The output 2 is controlled by which output table in the rotation group. READ ONLY.
322	1	1-4	-	-	The output 3 is controlled by which output table in the rotation group. READ ONLY.
323	1	1-4	-	-	The output 4 is controlled by which output table in the rotation group. READ ONLY.
324	1	1-4	-	-	The output 5 is controlled by which output table in the rotation group. READ ONLY.
325	1	0-65535	-	-	Rotation time left. Hong long time left the rotation will happen. READ ONLY.
326	1	0-1	-	-	Show the size of E2 chip. 0 = 24c02, 1 = 24c08/24c16.
327	1	0-3	0	0	Assign the timer be used for which feature. 0 = period timer, 1 = rotation timer, 2 = interlock, 3 = PWM timer.
328	1	0-2	0	0	The output1 function, there are three functions for the output1. 0 = normal ON/OFF output, 1 = rotation , 2 = lighting control.
329	1	2-5	-	-	Show which output table is using for this output when this output function be set rotation
330	1	2-5	-	-	Show which output table is using for this output when this output function be set rotation
331	1	2-5	-	-	Show which output table is using for this output when this output function be set rotation
332	1	2-5	-	-	Show which output table is using for this output when this output function be set rotation
333	2	0-65535	-	-	How much time left before rotation action.
334	1	0-2	0	0	The output2 function, there are three functions for the output2 .0 = normal ON/OFF output, 1 = rotation , 2 = lighting control.
335	1	0-2	0	0	The output3 function, there are three functions for the output3. 0 = normal ON/OFF output, 1 = rotation , 2 = lighting control.
336	1	0-2	0	0	The output4 function, there are three functions for the output4. 0 = normal ON/OFF output, 1 = rotation , 2 = lighting control.
337	1	0-2	0	0	The output5 function, there are three functions for the output5. 0 = normal ON/OFF output, 1 = rotation , 2 = lighting control.
338	1	0-255	20	68	Default occupied setpoint. Works in concert with the "occupied setpoint control register", register 339

340	1	0-1	0	0	Enable/disable PIR correspond 1/0 respectively.
341	1	0-255	0	0	PWM output range in COAST mode. 0 = CLOSE, 1 = OPEN, 2 = 0-100%,3 = 50-100%,4 = 0-50%. MSb 4 bits correspond to output4 and LSB 4 bits correspond to output5
342	1	0-255	64	64	PWM output range in COOLING2 mode. 0 = CLOSE, 1 = OPEN, 2 = 0-100%,3 = 50-100%,4 = 0-50%. MSb 4 bits correspond to output4 and LSB 4 bits correspond to output5
343	1	0-255	38	38	PWM output range in COOLING3 mode. 0 = CLOSE, 1 = OPEN, 2 = 0-100%,3 = 50-100%,4 = 0-50%. MSb 4 bits correspond to output4 and LSB 4 bits correspond to output5
344	1	0-255	16	16	PWM output range in COOLING1 mode. 0 = CLOSE, 1 = OPEN, 2 = 0-100%,3 = 50-100%,4 = 0-50%. MSb 4 bits correspond to output4 and LSB 4 bits correspond to output5
345	1	0-255	4	4	PWM output range in HEATING1 mode. 0 = CLOSE, 1 = OPEN, 2 = 0-100%,3 = 50-100%,4 = 0-50%. MSb 4 bits correspond to output4 and LSB 4 bits correspond to output5
346	1	0-255	3	3	PWM output range in HEATING2 mode. 0 = CLOSE, 1 = OPEN, 2 = 0-100%,3 = 50-100%,4 = 0-50%. MSb 4 bits correspond to output4 and LSB 4 bits correspond to output5
347	1	0-255	1	1	PWM output range in HEATING3 mode. 0 = CLOSE, 1 = OPEN, 2 = 0-100%,3 = 50-100%,4 = 0-50%. MSb 4 bits correspond to output4 and LSB 4 bits correspond to output5
348	1	0-100	-	-	The ON period take how many percentage for output4
349	1	0-100	-	-	The ON period take how many percentage for output5
350	1	0-255	0	0	Free cooling configuration. bit0, free cool enable/disable, 0 = disable, 1= enable. bit1, free cool available decided by local or external master. 0 = local, 1 = NC. bit2, free cool available status, 0 = npt available, 1= available. bit4, show the status if NC is OK when the free cool decided by NC.
					Analog Output Tables (bit0, 1 =analog out1, bit2, 3=analog out2, 00=0%, 01=0-100%, 11=100%)
351	1	0-255	0	0	Analog output OFF table, coasting mode
352	1	0-255	0	0	Analog output OFF table, cooling1 mode
353	1	0-255	0	0	Analog output OFF table, cooling2 mode
354	1	0-255	0	0	Analog output OFF table, cooling3 mode
355	1	0-255	0	0	Analog output OFF table, heating1 mode
356	1	0-255	0	0	Analog output OFF table, heating2 mode
357	1	0-255	0	0	Analog output OFF table, heating3 mode
358	1	0-255	0	0	Register lock.All registers except fan speed and manual inputs/outputs register are not writable. 0 = lock,1 = no lock.
359	1	0-255	0	0	Outside temperature for free cooling,from external sensor or NC.
360	2	0-255	0	0	If outside temp be set from NC.The communication with NC must be set in this time,otherwise will set error status and use external sensor.
361	1	0-255	0	0	If the outside air temp is lower than the room temeprature by this amount, then the free cooling is worthwhile, 350 bit2 = 1.If the OAT is greater than the room temp, then free cooling mode is not worhtwhile. , 350 bit2 = 0
					Output table in free cooling mode,0 = 0%, 1 = 100%,2 = MIN->100%,3 = MIN|100%,4 = MIN.Bit7 through 4 correspond to OFF table,bit 3 through 0 correspond to ON table.
362	1	0-255	0	0	Free cooling output configuration.Coasting mode
363	1	0-255	0	0	Free cooling output configuration.Cooling1 mode

Address	Bytes	Range	Default		Register and Description
364	1	0-255	0	0	Free cooling output configuration.Cooling2 mode
365	1	0-255	0	0	Free cooling output configuration.Cooling3 mode
366	1	0-255	0	0	Free cooling output configuration.Heating1 mode
367	1	0-255	0	0	Free cooling output configuration.Heating2 mode
368	1	0-255	0	0	Free cooling output configuration.heating3 mode
369	1	0-255	0	0	Min Air,the units is percent.Set the minimum output for free cooling,the default is 15%
370	1	0-255	0	0	Outside air temperature in hottest day
371	1	0-255	0	0	Outside air temperature in coldest day
372	1	0-255	0	0	Offset in hottest day
373	1	0-255	0	0	Offset in coldest day
374	2	0-65535	0	0	Store setpoint in two bytes ,the resolution is 0.1
375	1	0-255	0	0	Crrent setpoint = user setpoint + offset setpoint
376	1	0-255	0	0	Setpoint offset
377	1	0-255	0	0	Change over sensor mode,1 = cooling mode,0 = heating mode.
The following register is for Tstat 5E.					
Address	**Bytes**	**Range**	**Default**		**Register and Description**
359	1	0-255	0	0	ANALOG INPUT RANGE. 0 = raw data, 1 = thermistor, 2 = %, 3 = ON/OFF, 4 = CUSTOMER, 5 = OFF/ON
360	1	0-255	0	0	ANALOG INPUT RANGE. 0 = raw data, 1 = thermistor, 2 = %, 3 = ON/OFF, 4 = CUSTOMER, 5 = OFF/ON
361	1	0-255	0	0	ANALOG INPUT RANGE. 0 = raw data, 1 = thermistor, 2 = %, 3 = ON/OFF, 4 = N/A, 5 = OFF/ON
362	1	0-255	0	0	ANALOG INPUT RANGE. 0 = raw data, 1 = thermistor, 2 = %, 3 = ON/OFF, 4 = N/A, 5 = OFF/ON
363	1	0-255	0	0	ANALOG INPUT RANGE. 0 = raw data, 1 = thermistor, 2 = %, 3 = ON/OFF, 4 = N/A, 5 = OFF/ON
364	1	0-255	0	0	ANALOG INPUT RANGE. 0 = raw data, 1 = thermistor, 2 = %, 3 = ON/OFF, 4 = N/A, 5 = OFF/ON
365	1	0-255	0	0	ANALOG INPUT RANGE. 0 = raw data, 1 = thermistor, 2 = %, 3 = ON/OFF, 4 = N/A, 5 = OFF/ON
366	1	0-255	0	0	ANALOG INPUT RANGE. 0 = raw data, 1 = thermistor, 2 = %, 3 = ON/OFF, 4 = N/A, 5 = OFF/ON
367	2	0-255	0	0	Analog inptu1 value
368	2	0-255	0	0	Analog inptu2 value
369	2	0-255	0	0	Analog inptu3 value
370	2	0-255	0	0	Analog inptu4 value
371	2	0-255	0	0	Analog inptu5 value
372	2	0-255	0	0	Analog inptu6 value
373	2	0-255	0	0	Analog inptu7 value
374	2	0-255	0	0	Analog inptu8 value
375	2	0-255	0	0	Calibration for analog input1
376	2	0-255	0	0	Calibration for analog input2
377	2	0-255	0	0	Calibration for analog input3
378	2	0-255	0	0	Calibration for analog input4
379	2	0-255	0	0	Calibration for analog input5
380	2	0-255	0	0	Calibration for analog input6
381	2	0-255	0	0	Calibration for analog input7
382	2	0-255	0	0	Calibration for analog input8
383	1	0-255	0	0	The first character on LCD line 1, ASCII code
384	1	0-255	0	0	The second character on LCD line 1, ASCII code
385	1	0-255	0	0	The third character on LCD line 1, ASCII code
386	1	0-255	0	0	The fourth character on LCD line 1, ASCII code
387	1	0-255	0	0	The fifth character on LCD line 1, ASCII code
388	1	0-255	0	0	The sixth character on LCD line 1, ASCII code
389	1	0-255	0	0	The seventh character on LCD line 1, ASCII code
390	1	0-255	0	0	The eighth character on LCD line 1, ASCII code
391	1	0-255	0	0	The first character on LCD line 2, ASCII code
392	1	0-255	0	0	The second character on LCD line 2, ASCII code
393	1	0-255	0	0	The third character on LCD line 2, ASCII code
394	1	0-255	0	0	The fourth character on LCD line 2, ASCII code
395	1	0-255	0	0	The fifth character on LCD line 2, ASCII code
396	1	0-255	0	0	The sixth character on LCD line 2, ASCII code
397	1	0-255	0	0	The seventh character on LCD line 2, ASCII code
398	1	0-255	0	0	The eighth character on LCD line 2, ASCII code
399	1	0-255	0	0	LCD turn off delay, 0 - 255 minutes
400	1	0-255	0	0	Select what parameter will display, 0 = nothing, 1 =temperature, 2 = setpoint, 3 through 10 corre-spond to input 1 through 8, 11 = mode
401	1	0-255	0	0	Select what parameter will display, 0 = nothing, 1 =temperature, 2 = setpoint, 3 through 10 corre-spond to input 1 through 8, 11 = mode
402	1	0-255	0	0	DAY icon control register. 0 = AUTO, 1 = turn OFF, 2 = turn ON.
403	1	0-255	0	0	NIGHT icon control register. 0 = AUTO, 1 = turn OFF, 2 = turn ON.

Common Registers

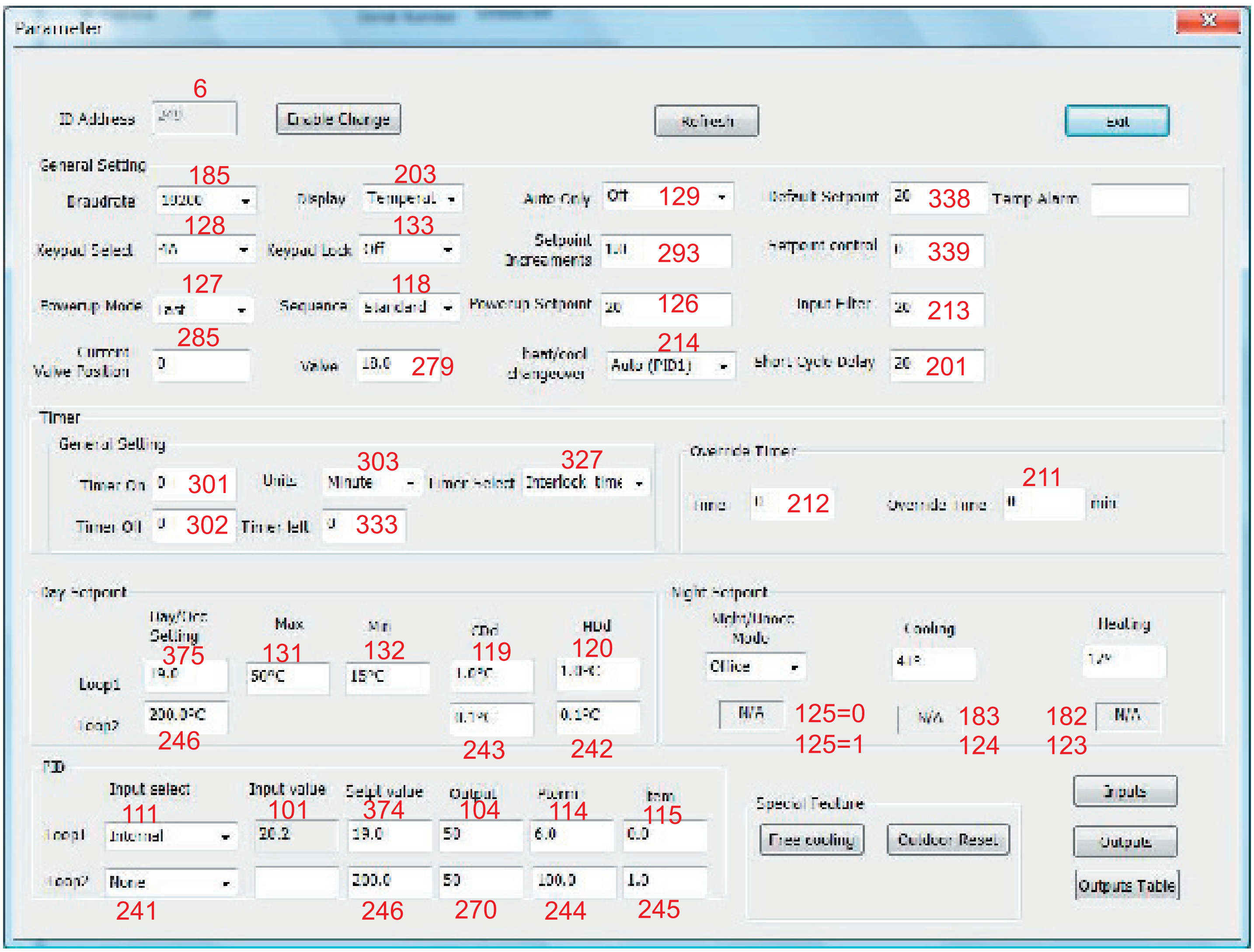


Main Screen Pic1

Inputs/Outputs Register for Tstat5 Series Thermostat								
	5A	5B	5C	5D	5E	5F	5G	5H
Out1	108.0	108.0	108.0	108.0	108.0	108.0	108.0	102
Out2	108.1	108.1	108.1	108.1	108.1	108.1	108.1	103
Out3	108.2	108.2	108.2	108.2	108.2	108.2	108.2	387
Out4		108.3	108.3	108.3	108.3	348	348	388
Out5		108.4	108.4	108.4	108.4	349	349	389
Out6	102			102	102		102	390
Out7	103			103	103		103	391

Inputs/Outputs Register for Tstat5 Series Thermostat (Continued)								
	5A	5B	5C	5D	5E	5F	5G	5H
IN1	101	101	101	101	367	101	101	349
IN2	180	180	180	180	368	180	180	350
IN3			181	181	369	181	181	351
IN4			311	311	370	311	311	352
IN5					371			353
IN6					372			354
IN7					373			355
IN8					374			356

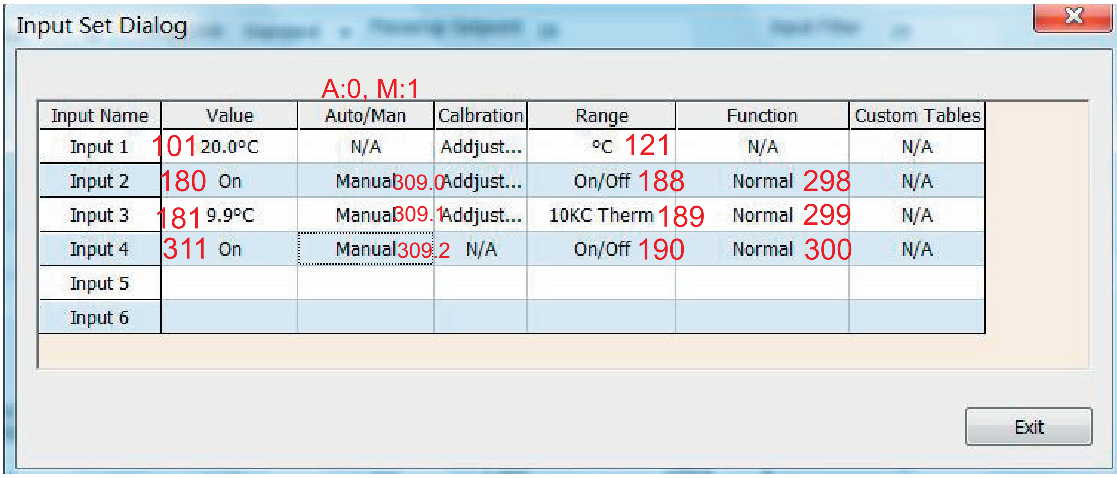
(Inputs & Outputs Table)



(Parameters Pic2)

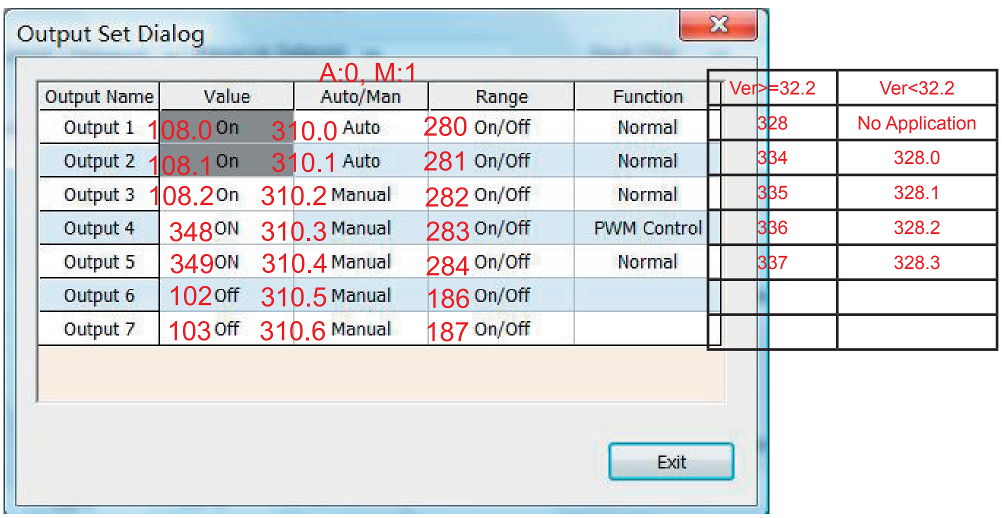
Input Dialog:

For 5G mode, registers maybe different depend on mode type.

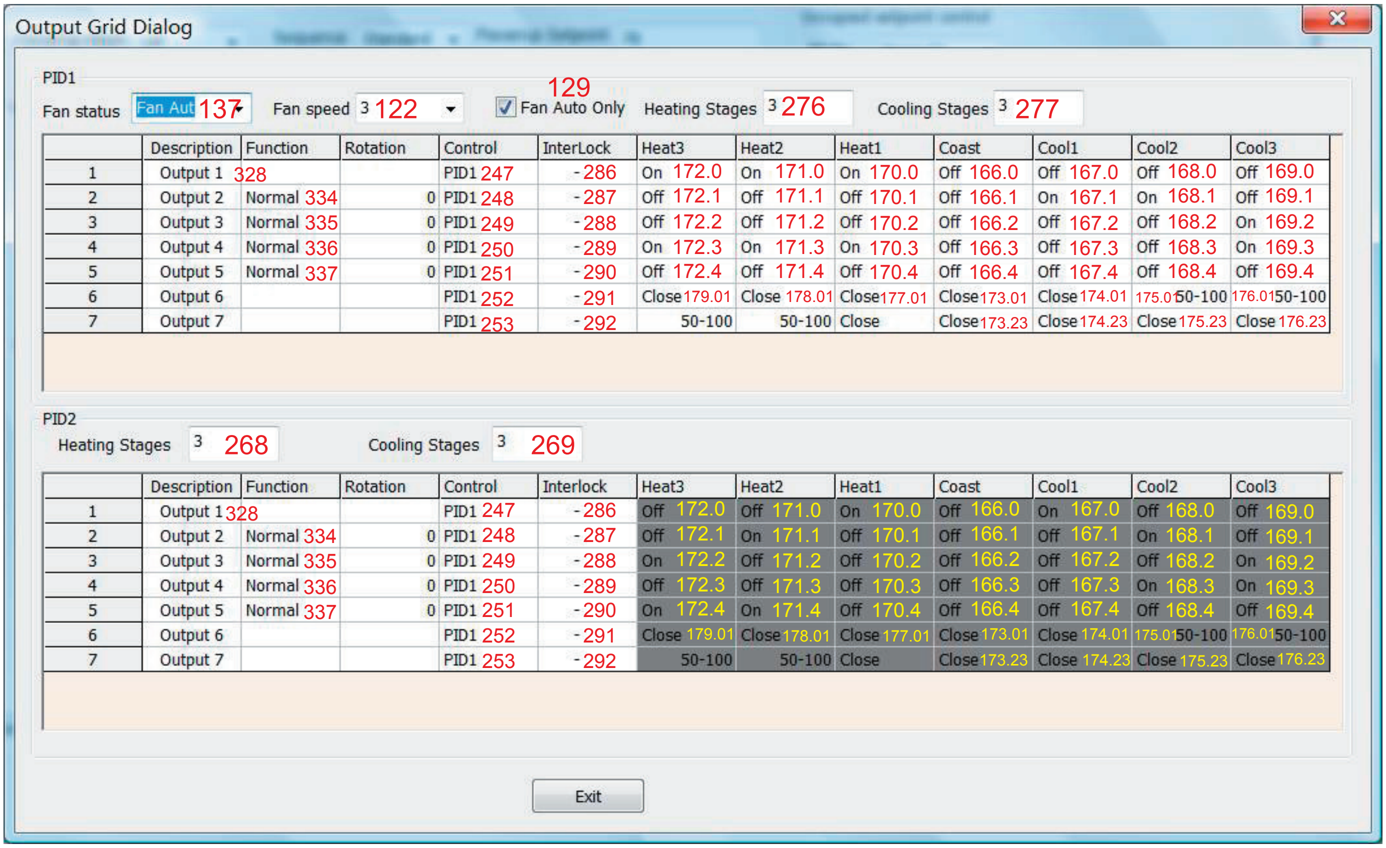


Output Dialog:

For 5G mode, registers maybe different depend on mode type.



	5A	5B	5C	5D	5E	5F	5G	5H
Out1	108.0	108.0	108.0	108.0	108.0	108.0	108.0	102
Out2	108.1	108.1	108.1	108.1	108.1	108.1	108.1	103
Out3	108.2	108.2	108.2	108.2	108.2	108.2	108.2	387
Out4		108.3	108.3	108.3	108.3	348	348	388
Out5		108.4	108.4	108.4	108.4	349	349	389
Out6	102			102	102		102	390
Out7	103			103	103		103	391
IN1	101	101	101	101	367	101	101	349
IN2	180	180	180	180	368	180	180	350
IN3			181	181	369	181	181	351
IN4			311	311	370	311	311	352
IN5					371			353
IN6					372			354
IN7					373			355
IN8					374			356



Inputs/Outputs Range Setting Registers								
	5A	5B	5C	5D	5E	5F	5G	5H
Out1	280	280	280	280	280	280	280	
Out2	281	281	281	281	281	281	281	
Out3	282	282	282	282	282	282	282	
Out4		283	283	283	283	283	283	
Out5		284	284	284	284	284	284	
Out6	186			186	186		186	
Out7	187			187	187		187	
IN1	121	121	121	121	359	121	121	341
IN2	188	188	188	188	360	188	188	342
IN3			189	189	361	189	189	343
IN4			190	190	362	190	190	344
IN5					363			345
IN6					364			346
IN7					365			347
IN8					366			348

Inputs/Outputs Auto/Manual Registers								
	5A	5B	5C	5D	5E	5F	5G	5H
Out1	310.0	310.0	310.0	310.0	310.0	310.0	310.0	
Out2	310.1	310.1	310.1	310.1	310.1	310.1	310.1	
Out3	310.2	310.2	310.2	310.2	310.2	310.2	310.2	
Out4		310.3	310.3	310.3	310.3	310.3	310.3	
Out5		310.4	310.4	310.4	310.4	310.4	310.4	
Out6	310.5			310.5	310.5		310.5	
Out7	310.6			310.6	310.6		310.6	
IN1	309.0	309.0	309.0	309.0	309.0	309.0	309.0	
IN2	309.1	309.1	309.1	309.1	309.1	309.1	309.1	
IN3			309.2	309.2	309.2	309.2	309.2	
IN4								
IN5								
IN6								
IN7								
IN8								

For example, if we would like to read the temperature at tstat node address #1,

Slave Address	Function	Starting Address Hi	Starting Address Lo	No. of Points Hi	No. of Points Lo	CRC Hi Byte	CRC Lo Byte
1	3	0	101	0	1	xx	xx

and the tstat sends back the following data:

Slave Address	Function	Byte Count	Data Hi	Data Lo	CRC Hi Byte	CRC Lo Byte
1	3	2	0	99	xx	xx

Slave Address	Function	Starting Address Hi	Starting Address Lo	No. of Points Hi	No. of Points Lo	CRC Hi Byte	CRC Lo Byte
1	3	0	110	0	8	xx	xx

Slave Address	Function	Starting Address Hi	Starting Address Lo	No. of Points Hi	No. of Points Lo	CRC Hi Byte	CRC Lo Byte
1	6	0	110	0	100	xx	xx

Slave Address	Function	Starting Address Hi	Starting Address Lo	No. of Points Hi	No. of Points Lo	CRC Hi Byte	CRC Lo Byte
1	3	0	101	0	1	xx	xx

Note when updating to Version 25

There are two new registers that will tell the CPU information about the model and hardware of the tstat. NOTE: after updating to version 25 you MUST setup these registers first or the tstat will not function properly.

Product Model is register address 7. The corresponding values are as follows:

Tstat5A	=	2
Tstat5B	=	1
Tstat5B2	=	3
Tstat5C	=	4
Tstat5D	=	12

Hardware revision is register address 8. The hardware revision can be found by removing the back plate of the tstat. It is written in the lower center of the board. If the revision is a letter (i.e. REVM, REVO, REVP), the hardware register should be set to 0. If the revision is a number (i.e. 01, 02, 03), the hardware register should be set to match the board.

339	1	0-2	0	0	Occupied Setpoint Control Register: 0 = normal, setpoint is managed by the serial port and keypad, the stat will remember the last occupied setpoint and use that during the next occupied period. 1 = Default mode, the last occupied setpoint if forgotten and the occupied setpoint gets reset to the default. 2 = trigger an event, when a master controller writes 2 to this register, the default setpoint will be copied to the occupied setpoint after which the Tstat will set the value back to 1 to show the event has been serviced.

Frequently Asked Questions

How can I change the address of the TSTAT?

The best way of configuring the TSTAT address is to use MBPoll. You can download it here:
http://www.modbustools.com/modbus_poll.asp
When you use MBPoll, it's important to remember to set the addressing to 'Protocol Addresses (Base 0)' under the 'Display' menu. This will insure that the addresses in MBPoll match the addresses in the tstat. Once you have connected to the Tstat with MBPoll, you can select 'Write Single Register' from the Function menu or by hitting 'F8'. Then enter 255 for the 'Slave ID', 6 for 'Address', and the new address of the Tstat for 'Value'. Then hit enter or Send. Next time you read/write to the Tstat you should be able to use this new address.

How do the stages relate to the setpoint, deadbands, and PID?

The tstat has the ability to function in 7 distinct stages - coasting + 6 stages that can be custom configured as heating or cooling stages. The stages are determined by the value of the PID as shown in figure 18. Note: there is a built-in hysteresis of 2% at each transition point between the stages.

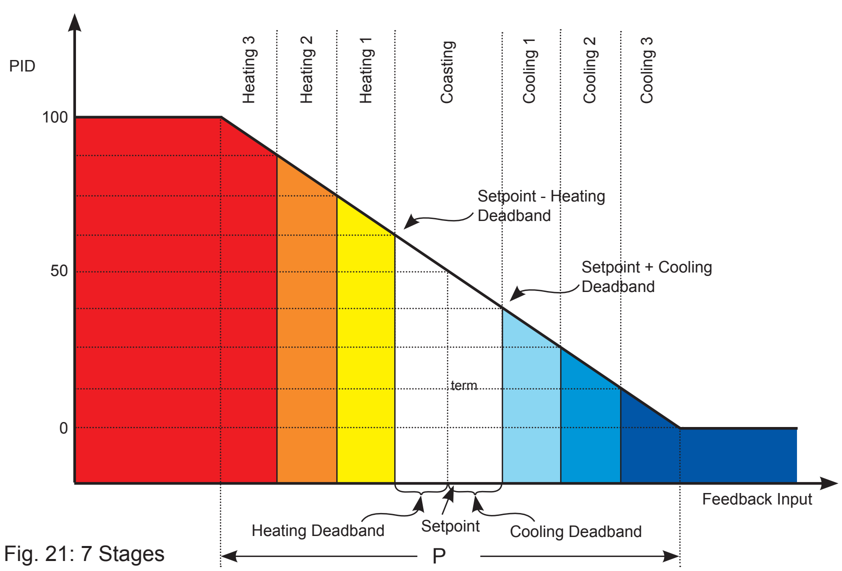


Fig. 21: 7 Stages

The PID is calculated using the Pterm, Iterm, and ΔT by the following formula:

$$PID = 50 + \frac{a\Delta T}{P_{term}} + \int b\Delta T \cdot I_{term} dt$$

Where a and b are constants and ΔT is the difference between room temperature and, either cooling or heating setpoint. The cooling and heating setpoints are calculated by adding or subtracting the respective deadband from the setpoint as is shown in figure 18.

What values should I use for the P and I terms?

Making the P term big makes the thermostat lazy.
Making the I term big does the opposite.
If the sensor is in the room, you need a lazy acting PID due to a lag between action and response.
If the sensor is in the supply duct (or pipe) then you need a quick acting PID due to a lag between action and response.

-Use something like 6.0° or 7.0° for the P term when the sensor is in the room. This means you need 6 or 7 degrees of change to get a full response on the mechanical equipment.
-You can use the I term to fine tune the response over time, use a value of 5.0 or so, to get a 5% response over one minute to make up the small error left over from the P term control.
-Too much P or I means your equipment will cycle, loosen up both if you see cycling.
-This also works for on/off and modulating type equipment.

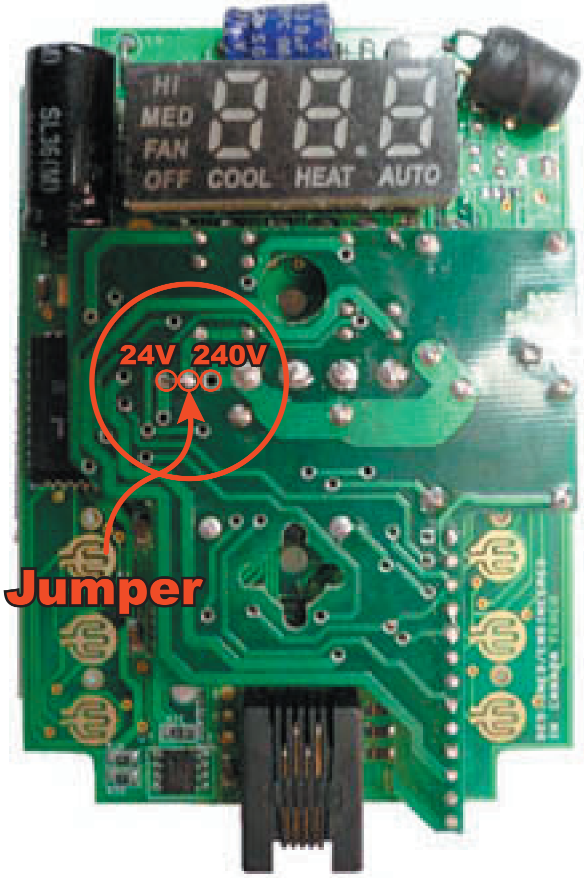
What is the relationship between the setpoint and the output signals?

The Tstat uses a PI calculation to decide on heating or cooling mode. We will consider only the cooling mode in this example. Let us say the CP term is set to 10.0°, the temperature is 31°C, the setpoint is 20°C and the Cooling deadband is 1°C. The tstat will be at rest from 20°C to 21°C, this is the deadband range where the room temperature is satisfied. Over the next 10°C of deviation (from 21°C to 31°C) the P term will go from 0 to 100%. Since there are three stages of cooling in the tstat, and if the valve is active from 0-10V on the first stage of cooling, the cooling output will go from 0 to 10V over the range of 21°C to 24.3°C.

How can I change the voltage level of the valve outputs on a Tstat5B?

The Tstat5B and Tstat5B2 have valve outputs that can be configured via a jumper as either 240VAC or 24VAC, depending on which type of valve is being used with the system. In general the jumper is preset at the factory for each order of Tstats and does not need to be changed. However, from time to time, a Tstat will need to be recommissioned to work with a different type of valve. In this case, it is necessary to change the jumper to set the correct output. The following procedure describes this process:

1.) Remove Tstat from enclosure by first unscrewing 3 screws.
2.) Find the jumper located on the front side of the Tstat (see photos below).
3.) Adjust the jumper according the required output - 24V or 240V. Note: for older Tstat5Bs (see left side photo) the jumper must be adjusted using a soldering iron.
4.) Replace Tstat into enclosure using the 3 screws.



Tstat5B

Tstat5B2

<u>What is the correlation between fan speed and occupancy mode?</u>

It is important to remember that Fan Speed OFF will activate the nighttime deadband/setpoints.

The Fan speed is not directly tied to the occupancy. The occupancy is meant to signify whether or not there are/should be people in the building. Usually a building is occupied during the day and unoccupied at night. The fan speed is independent of this. It can be set to any speed regardless of the occupancy. However, when switching into occupied mode, the fan speed should not remain OFF – because of the reason mentioned above. Therefore the tstat automatically switches the fan to AUTO, at which point the user has full control to manually adjust the tstat.

When changing to unoccupied mode, the fan speed will automatically switch to OFF. If the tstat is unoccupied, the user still has the ability to manually adjust the tstat. This activates the ORT. When the ORT times out, the tstat automatically returns to OFF mode.

<u>When changing modules on the Tstat, what do I need to watch out for?</u>

Changing modules on the Tstat can change their functionality from Digital outputs to Analog outputs. Tstats 5A, 5B, 5B2 are grouped in the same family whereas Tstats 5C, 5D are another.

Product Name	5A	5B	5B2	5C	5D
Model Number	1	2	3	4	12

The best way of configuring the TSTAT Product Model Number is to use MBPoll. You can download it here:
http://www.modbustools.com/modbus_poll.asp

HOW TO DO

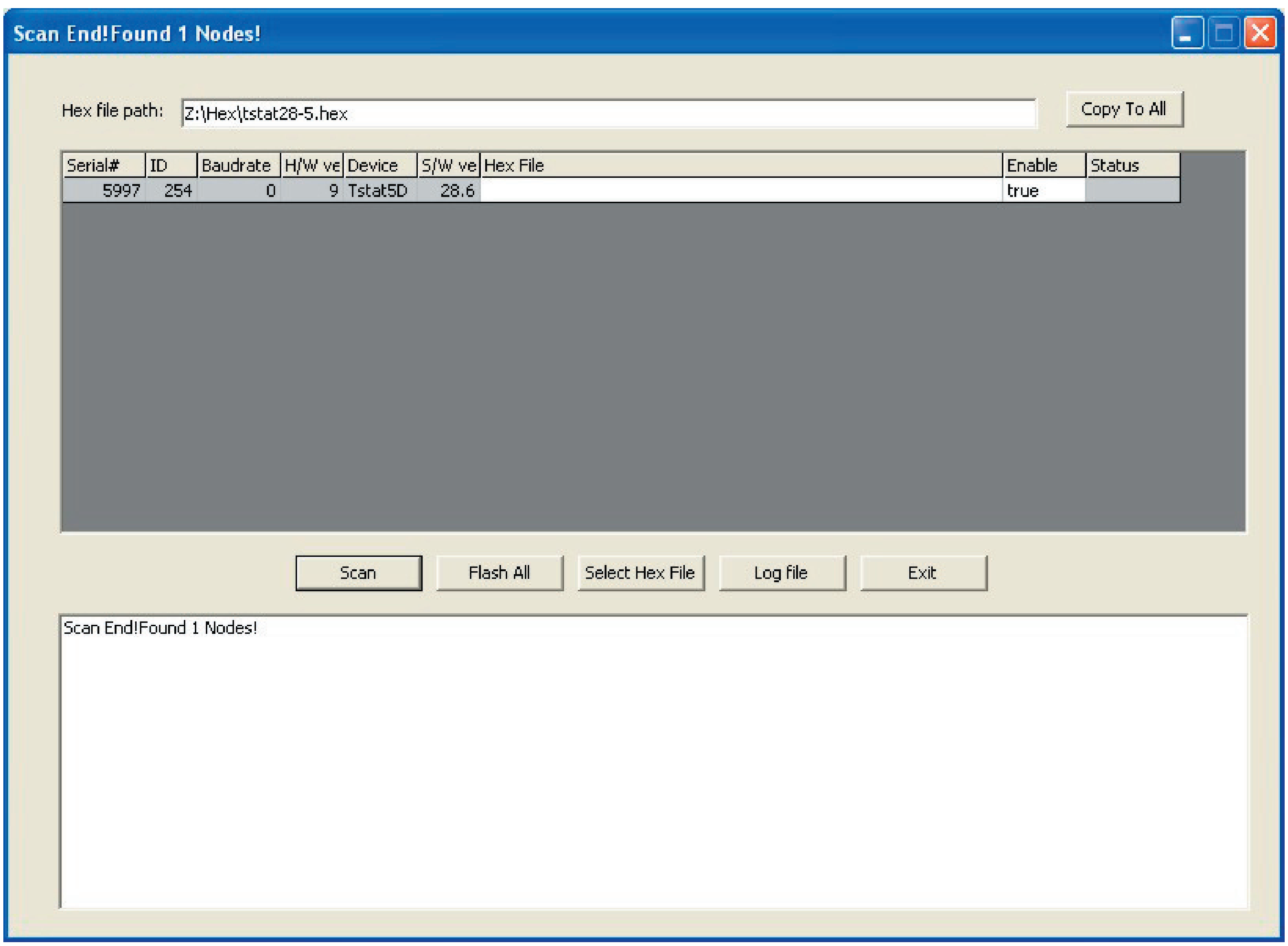
HOW TO ERASE E2 CHIP

Click the tstat which you want to erase in the left side of T3000.
Press Ctrl+Shift+F1 at the same time.
The following screen will pop up.

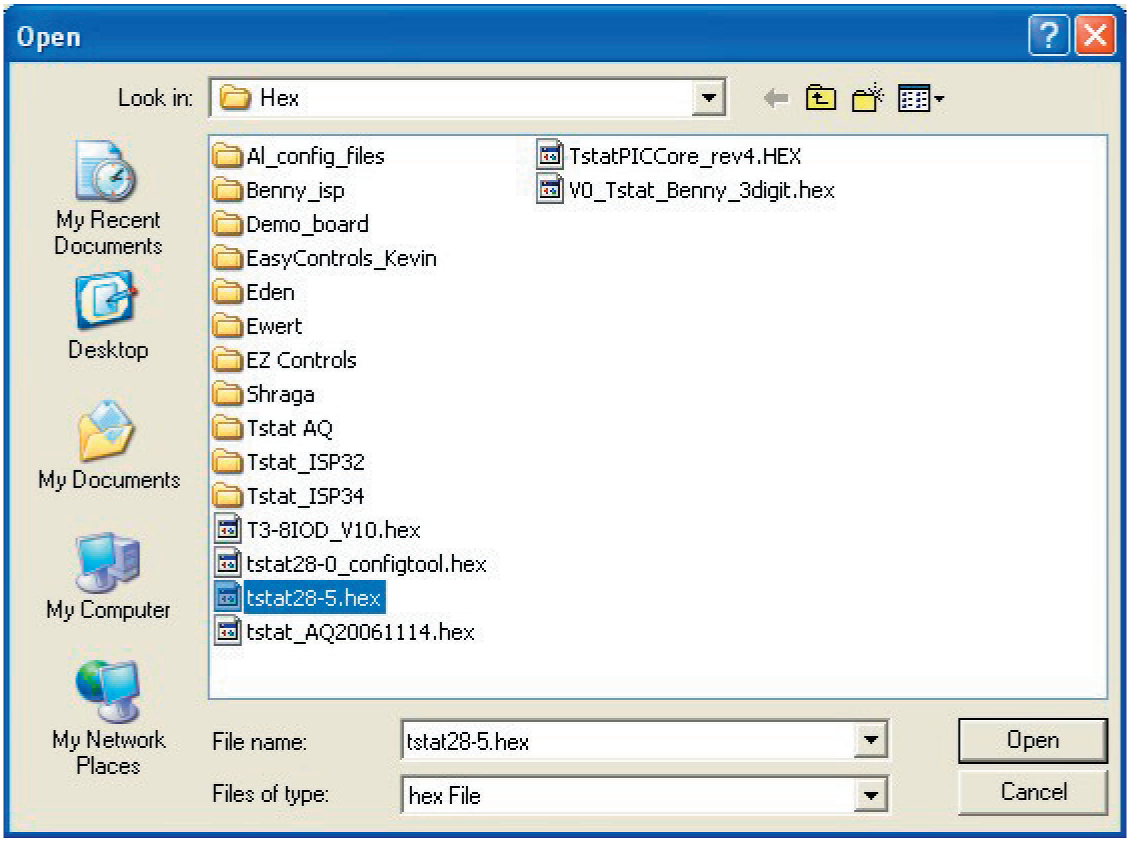
Click OK.

How to update firmware

Choose Tool/Update Firmeare... in the menu of T3000
Click Scan button in the middle of the following screen



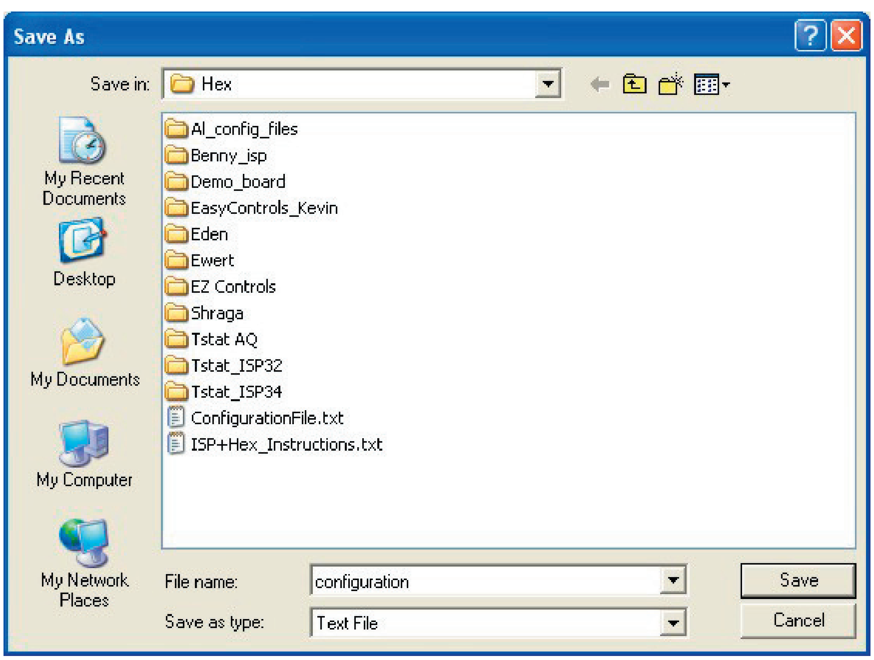
Click the Select hex file button to select the hex file to be loaded.



Click Flash All to load the hex file to Tstat.

How to upload configuration file

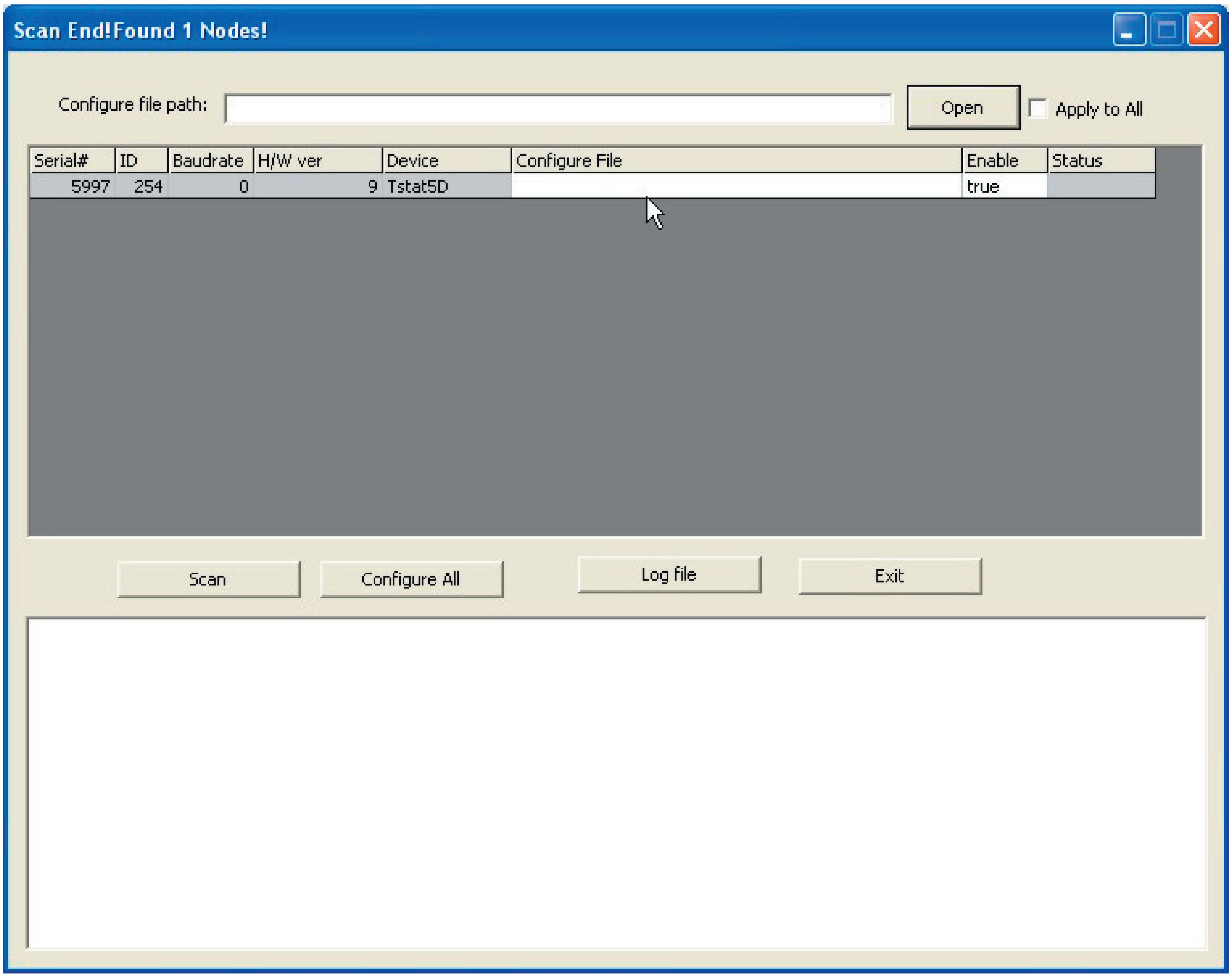
Choose File/save as... in the menu of T3000.
Select a directory you want the configuration file to be stored in the listbox following Save in.



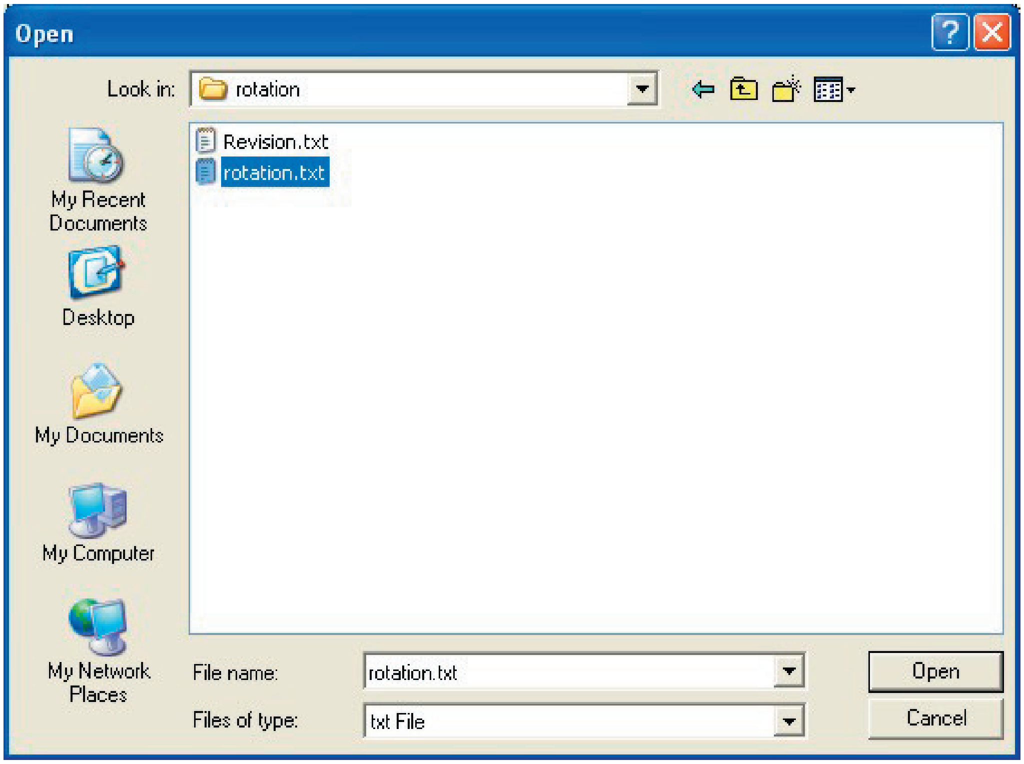
Input a file name for the configuration file in the listbox following File name.
Click Save button.

How to load configuration file

Choose File /Load in the menu of T3000.
The load configuratin file screen will pop up.



Click Open to select the configuration file to be loaded.



Selected Apply tp All and Enable should be true.
Click Configure All.

How to find the tstat automatically

Click Tool/Scan in the menu of T3000
Just wating,The T3000 will scan the tstat automatically.

How set up serial port and baudrate

Choose Tool/Optition... in the menu of T3000.
Select the serial port you are using and the baudrate you need.

Modbus Serial Communications

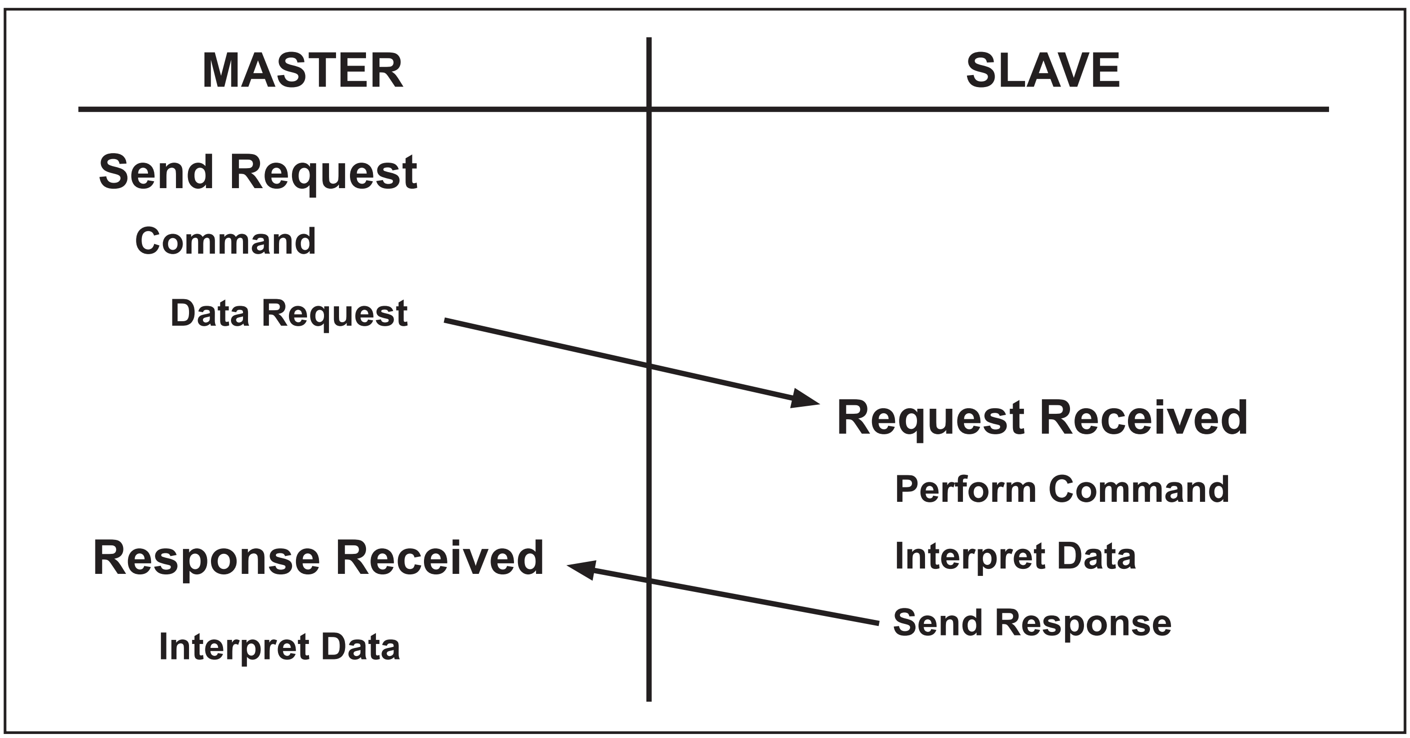
Overview

Modbus protocol is a widely used and well-documented communications method. It provides a simple and effective means of programming our various products.
The device communications follow a routing of queries issued by the master (your product) and our devices (the slaves). Modbus has various command types. we use only the READ , WRITE and MULTIPLE WRITE commands, type 3 and type 5 commands in the Modbus parlance. The master will prepare a packet comprised of the target address, command type (read or write), the starting address and number of bytes to be accessed, and finally a CRC for error detection.
All devices are set to communicate over RS485 network using 19.2 kbaud. N81 byte structure. Normal RS485 distance, termination and cabling rules apply. Physical layer is standard twisted pair + ground cabling. A shield is optional.

A typical Modbus packet looks like this:

Byte1	Device ID, the destination address for a particular message
Byte2	Function
Byte3	Starting address of the particular storage register(s) to be read or written, hi byte,
Byte4	Starting address low byte
Byte5	No. of registers to read/write (hi byte)
Byte6	No. of registers to read/write (low byte)
Byte7	CRC hi byte
Byte8	CRC low byte

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



[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.

Modbus Examples

READ Command (0x03):

This function is used to read the contents of multiple memory registers. The master to the Modbus must specify, the device 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.

The master to the Modbus network will issue a read command:
- Device 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 Regs Hi	No. of Regs 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			

WRITE command (0x06):

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

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

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

MULTIPLE-WRITE Command (0x10):

This function is used to write to multiple memory registers. The master of the Modbus must specify the device 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:

- Device ID=11
- Write to address 291 (123h)
- Number of Registers 3
- Data1 = 10 (000Ah)
- Data2 = 11 (000Bh)
- Data3 = 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 variables 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 double the 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

The device node sends back the following response:

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

CRC Error Correcting Details

For programmers familiar with error detection schemes, CRC calculations will be familiar turf. For programmers new to Modbus we will offer sample source code in Visual C and Cbuilder to get your application started. For embedded platforms we can provide some sample Modbus code on request. 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
} ;
```

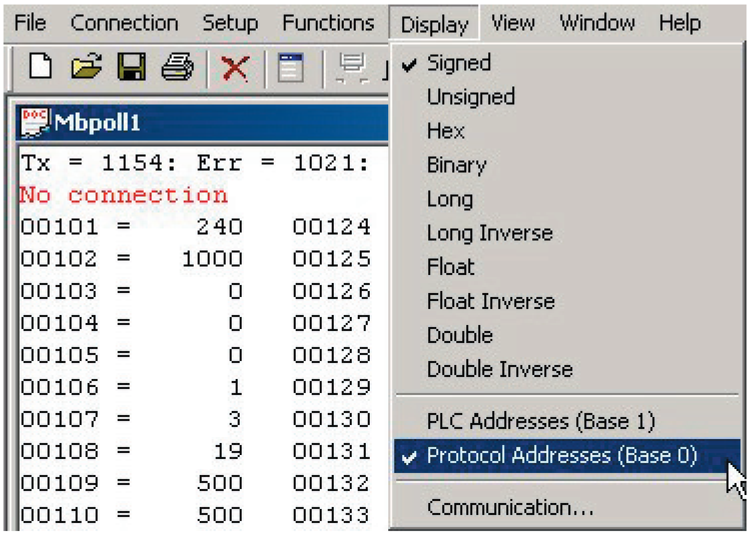
For example: to calculate the crc of the data in the message stored in memory location *puchMsg

```
unsigned 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) ;
}
```

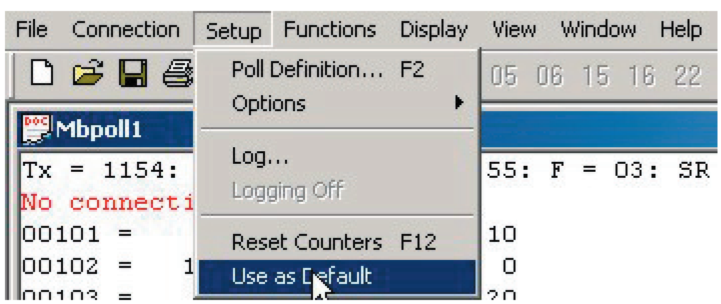
Modbus Poll Software

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.

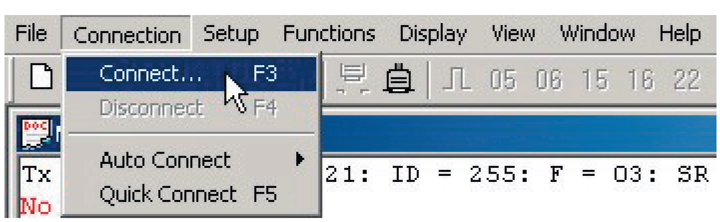
The first time Modbus Poll is used, it should be set to base 0 addressing. This is done by selecting "Protocol Addressing (Base 0) " from the Display menu:



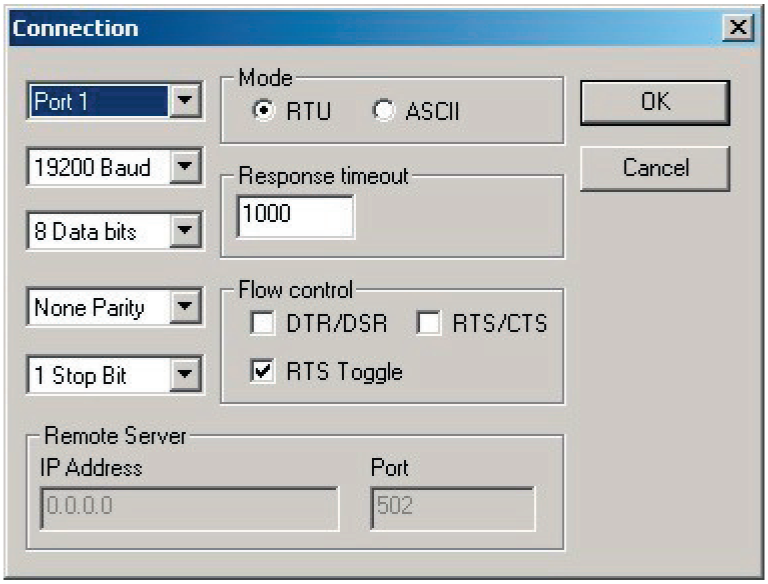
It is a good idea to then save this as a default so that addressing protocol does not need to be selected each time the program is running. Saving the default setting is done from the setup menu:



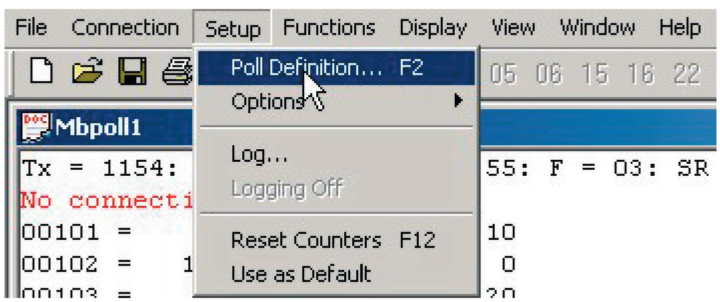
At this point, the connection to the device needs to be established. Select "Connect..." from the Connection menu:



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

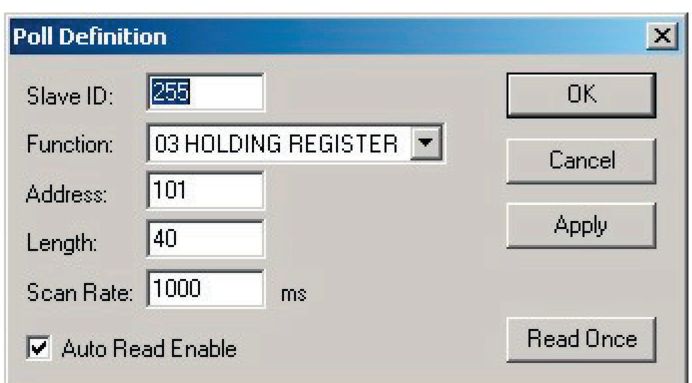


After the connection is established, it is necessary to setup the poll definitions. This is done by selecting “Poll Definition...” from the Setup menu:

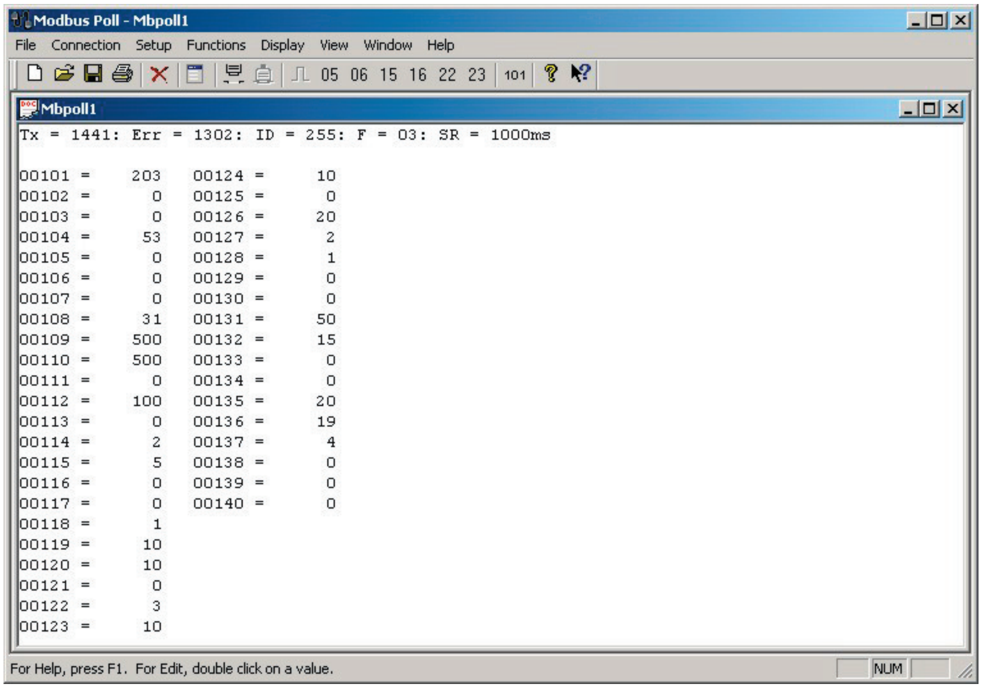


Within the Poll Definitions dialog window, there are several parameters that need to be set.

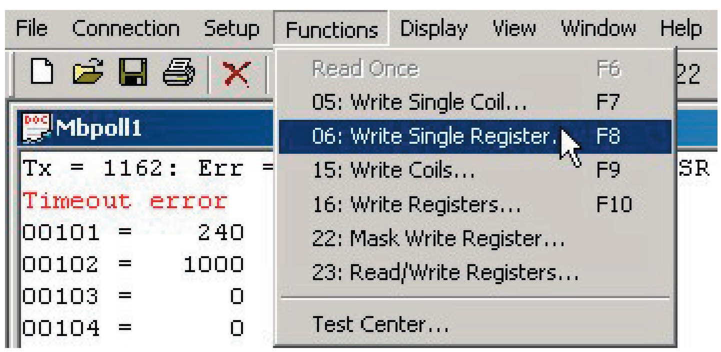
Slave ID is the modbus address of the device being read or written. (255 is the generic address to which all devices will respond.)
Function should be set as 03 HOLDING REGISTER.
Address is the starting address of the registers to be read.
Length is the number of registers to be read.
Scan Rate is the frequency with which the device will be polled.



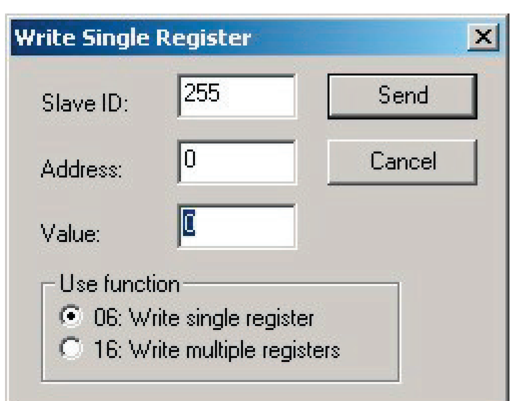
Once the Poll Definitions have been setup and applied, the main window will show a list of each register address and its corresponding value.



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.

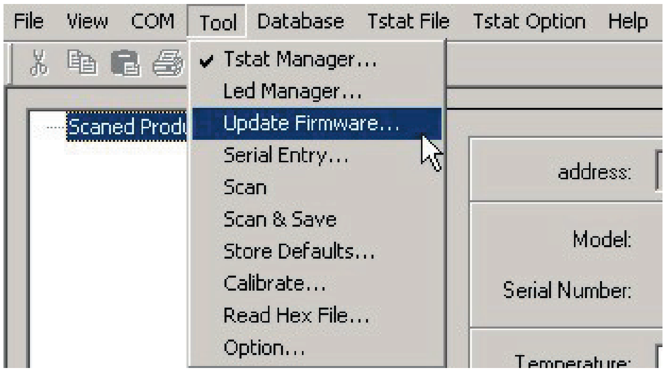


Flash Update Protocol

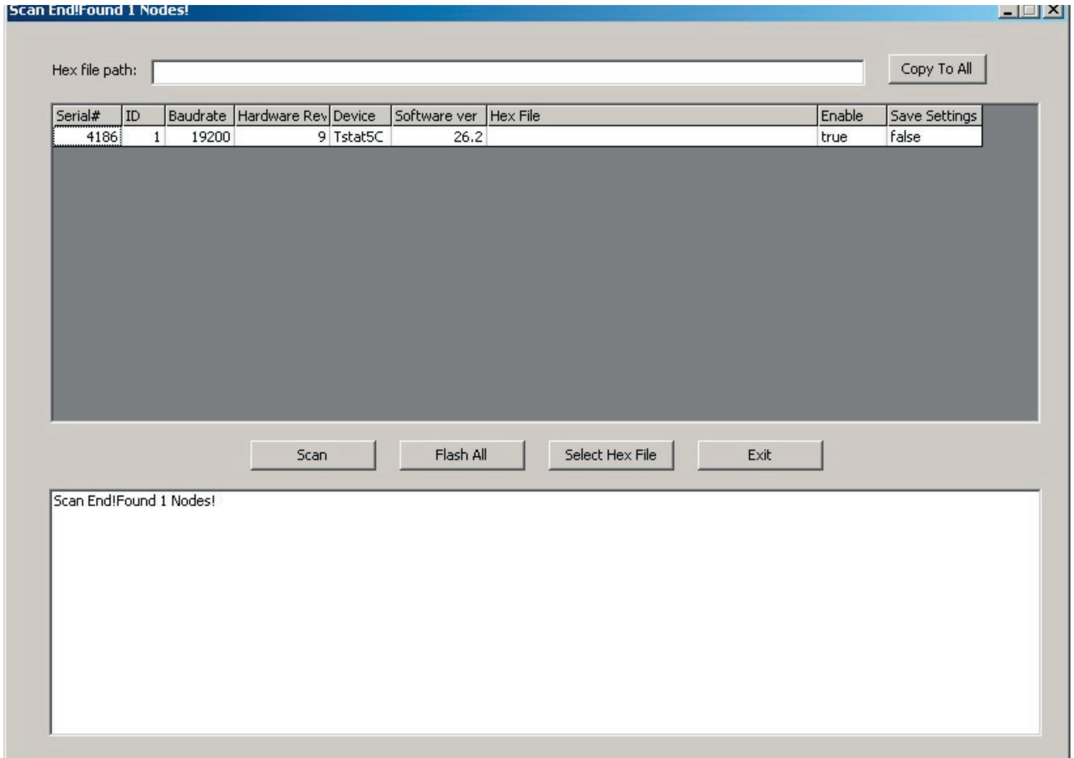
Instructions for Updating Devices with Temco ISP

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.

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.

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_UPDARTE_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_STATUS. 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 T3module is in ISP mode, the same write command must be sent again (write 7Fh to register #16), at which point the T3module will respond with a regular modbus response. This is necessary for clearing the Interrupt vectors and making sure all RAM memory is cleared.

[Notice]: All devices are installed with a 'Flash Update' jumper. Linking the jumper upon resetting the device will force the firmware to start in ISP mode and then wait for further instructions. Note that at this stage the device may not have a proper ID and by default will be set to 254.

- All Modbus communication commands are always 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 initialize' and 'erase flash' step require a longer timeout period than the 'programming' step. (250ms and 500ms respectively)

- 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.

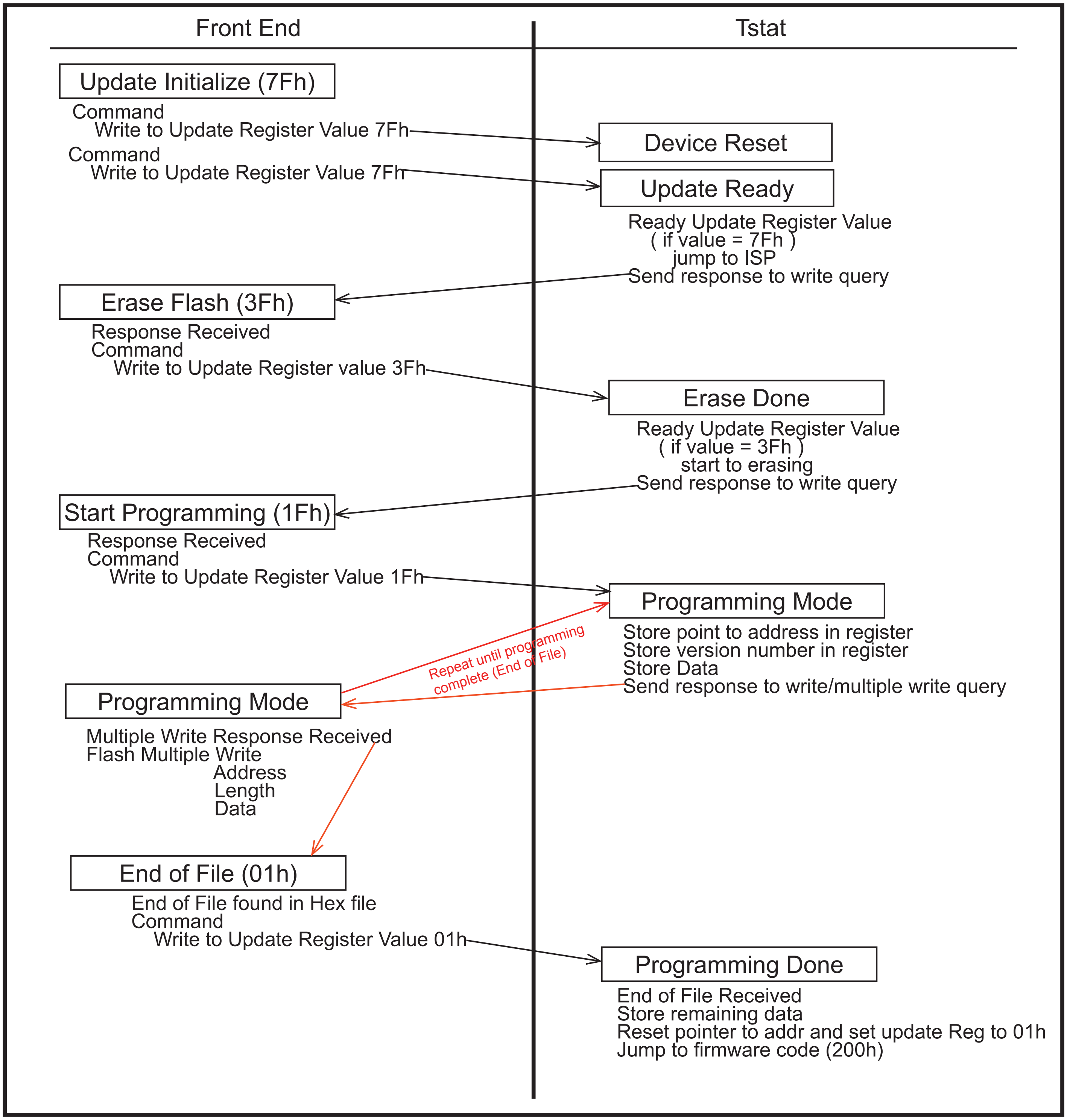
- At this point, the data must be sent using the multiple-write command. Packets can be of size 1 data byte to a **maximum of 128 data bytes.**

- In the event of an interrupted flash update, the master can poll the EEP_UPDATE_PTR and begin programming from this location.

Below is a graphical representation of the protocol.

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 programm itself. Given that polling is now used, communications is 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.



Example of a Programming Routine (Front End Side)

UPDATE INITIALISE

1 - Send Modbus **Write** Command to address **Update_Register** value **7F**h
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 **7F**h 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 **3F**h
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)

START PROGRAMMING

4 - Send Modbus **Write** Command to address **Update_Register** value **1F**h

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 **0080**h
Data length of **10**h
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 **02**0200 to **FF**0200

END OF FILE

9 - End of file found in Hex file
:00 0000 **01** FF
Bit 7 and 8 are 01

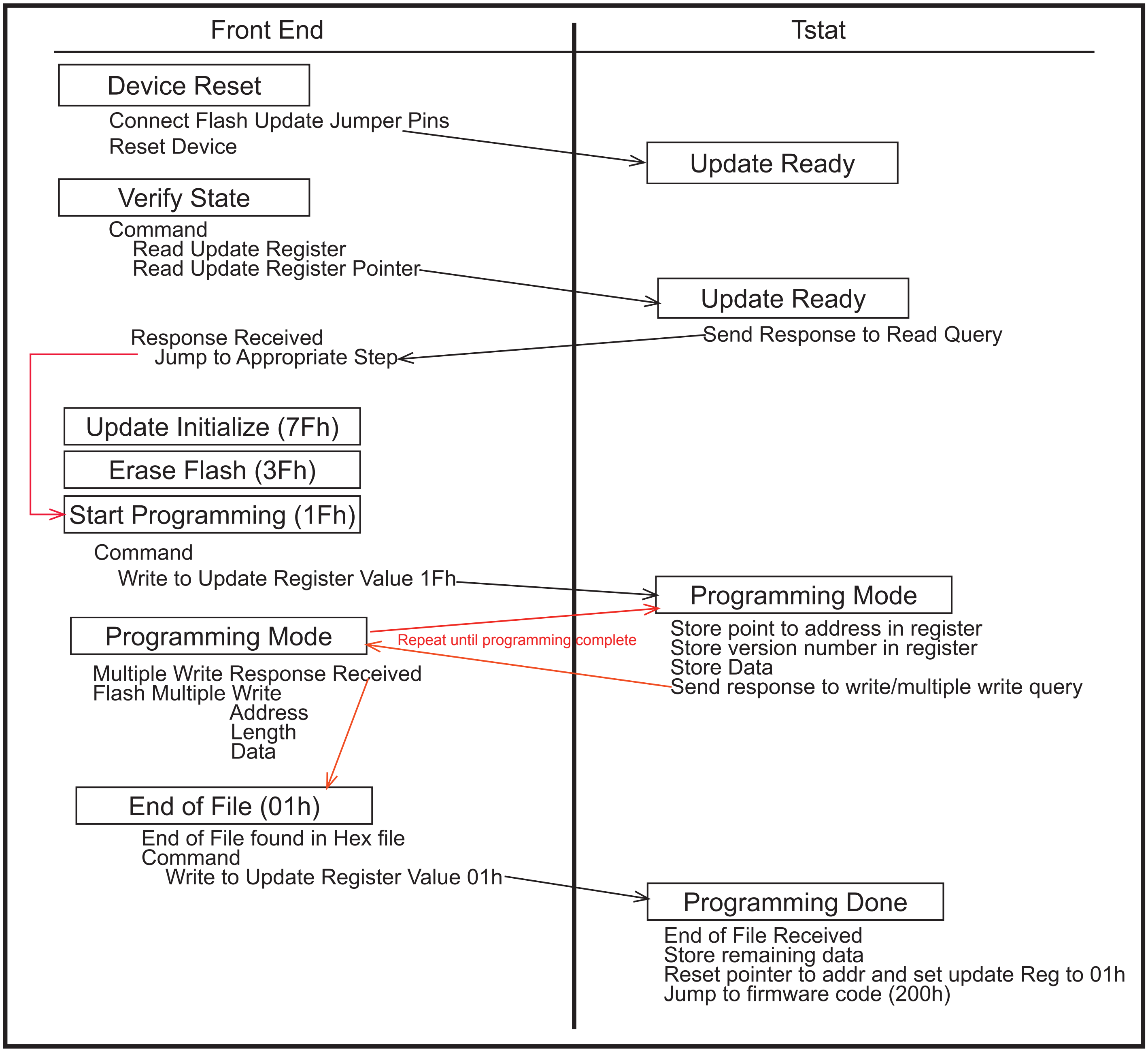
10 - Send Modbus **Write** Command to address **Update_Register** value **01**h
This will cause the device to reset itself and boot in normal operation mode

To Resume a Previously Interrupted Programming Routine

If during the programming sequences the upload was interrupted, there is still possibility for the front end to resume its 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 register in order to know where the last upload was being performed. Not that this register is written to before the data has been uploaded. Thus, in order to resume this step the Front End needs to re-write to this register again and then follow up from there.

Assuming the new software has only been partially uploaded, the Flash Update Jumper pins need to be connected upon resetting of the board. 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 such as this:

Data of the new Firmware				Modified data to be uploaded
;03 0000 00 020200 F9	→	need to change to this to	→	;03 0000 00 FF0200 FC
(Intel Hex format described below):				

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.

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

Flash Update Jumper

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.

Revision History

Ver28.8

- Implemented a safety mechanism to prevent users from setting the Pterm and deadbands in such a way that the tstat would permanently be in coasting mode.

Ver28.7

- Changed the resolution of valve travel time from 0.1 second to 1 second.
- Added the averaging sensor feature. A new function setting for a temperature sensor input, it can be assigned to average with the internal sensor and external sensor then feed back to the PID.

Ver28.6

- Delay the timeout for serila port receive and reset the watch dog when response mutiple write
- Added three functions for analog1,2 input,move the function in register 190 to register 300.

Ver28.5

- Added a 10 second delay when the mode of operation is decreased

Ver28.4

- Initilize the pidoutput array to determine the output controlled by which output table. Ver28.3
- Fixed PID2 bug.Can not get the right pid to control the output when the user select min and max funtion in pid control
- Adjust the involatile data structure because the data has extended the size of E2 chip

Ver28.2

- Fixed the pid bug .The Pid will get wrong value as temperature nearby setpoint if set the cool statge equals to zero

Ver28.1

- Fixed the code of PID2 part.the PID2 also can work if the range of corresponding input is 10K thermistor.
- Added sendZero() routine to reduce code space, used by several routines.

Ver28.0

- Added register 317 to store the value which determine how long time the Tstat will go occupied mode after power on and see no serial communication.
- Added register 318 for rounding display.
- Fixed keypad press counter feature.

Ver27.9

- Added config tool feature.The tstat will transfer the configuration itself to another tstat by pressing the lower right button.we realized this feature by a define predefine comamnd.So if the user need this feature,they have to tell us before we programm tstat.
- Fixed the keypad in 2A mode.
- Added five register to operate relays when the corresonding relay is set in manual mode.
- Added a register 15 to select the base address.0,Protocol Address.1,PLC Address
- Fixed a bug that the Tstat can not go to Cooling stage3 in some cases.

Ver27.8

- Fixed the occupancy sensor feature for Shraga.There are two modes about using the feature.one mode is in office mode and another mode is in hotel mode.Now we have only one way to change the tstat to go to occupied mode,that is write 1 to register 184.Before the tstat goes to occupied mode if the user change the fan speed from 0 to a value bigger than 0.Now if we change the fan speed whatever from keypad or serial port,tstat will reset the override timer.

Ver27.7
- Added a function item for each input.the function include "normal","freeze protect","occupancy sensor"
- Added auto/manu control bit for each input and output.

Ver27.6
- Added freeze protect feature.
- Added Period timer feature.

Ver27.5
- Changed the value range of filter.The filter can be set from 0 to 100.
- Update documents for hotel/office deadband. document information were flipped. Office is setpoint. Hotel is Deadband

Ver27.4
- Added LED table option Stage_1-2-3 and Stage_2-3
- Added fan speed change for Kevin in keypad 2B settings

Ver27.3
- Fixed trigger override timer mode,any key be pressed the Tstat enter override timer mode.

Ver27.2
- Fixed the tempetature display mode on LEDs

Ver27.1
- Clear item zero when the temperature in deadband area

Ver27-0
- Removed EA=0 lines from the pic read function.Reduced flicker to the display.Note that pic was left unchanged.
- Placed back the code which now only enables writing to hardware-rev and Product-Model only once

Ver26.9
- Fixed pid algorithm.change the pid_error arithmetic when at coating mode to make iterm decrease and increase fast
- Fixed the protocol of scan.Now read and write register 10 not Modbus protocol.Because added serialnumber data in the data packet
- Fixed occupied_trigger_handler() and related to make bit3 ofregister 184 can show Digital input1 status at any time and any case

Ver26.8
- Fixed pid bug .expanded pterm greater than night_cooling_db + night_heating_db.
- Added flash_info function at code 0x100.
- Added a register for last key pressed timer.
- Added serialnumber data when reply modbus function 25 to speed up scan.

Ver26.7
- Fixed the value range of MODBUS_MODE_OPERATION register(107) forcompatible backward.Now use the following value:
  coasting 0 cooling1 1 cooling2 2 cooling3 3 cooling4 14   cooling5 15 cooling6 16
  coasting 0 heating1 4 heating2 5 heating3 6 heating4 17 heating5 18 heating6 19
- Fixed pid bug ,the PID was winding up past 100, taking too long to wind down. Also in the deadband region, made sure the I term can decay but will not incease.
- Added 0---50% item for output table to determin how much the valve open.now the EEP_VALVE_OPER_TABLE_BEGIN register every bit is defined like this:

bit7 : '0' heating valve(output7)not select 0-50% ; '1' heating valve(output7)select 0-50% bit6 : '0' cooling valve(output6)not select 0-50% ; '1' cooling valve(output6)select 0-50% bit5 4 : floating valve ,the range of valve open.0,close;1,0-100%;2,50-100%;3,open. bit3 2 : heating valve(output7), the range of valve open.0,close;1,0-100%;2,50-100%;3,open. bit1 0 : cooling valve(output6),the range of valve open.0,close;1,0-100%;2,50-100%;3,open.

- After loaded the configuration file or an new hex file of the same rev number, the occupied state is left unchanged.
- Fixed some code when POM == 1, in this mode,only do something when fan speed is zero.
- When tstat at unoccupied mode ,recaculate cooling_db and heating_db to make sure PID work correctly.
- Fixed calibration bug which can not calibrate at TSS = 0.
- Fixed LED status,turn all leds off except OFF led when the tstat working in the OFF mode.
- Fixed occupied status in REG184.
- Limit TSS less than 2 if the tstat is 5A or 5B.

Ver26.6

- Added DDC mode in AUTO_ONLY regiser to make the user can not change setpoint and ON/OFF mode from keypad.
- Added interlock for each output.Add seven registers 285 to 293 to store the interlock selection item.
- Added a new register to adjust the setpoint increment as an user variable.The range of setpoint increment from 0.1 to 1.
- Tss register is left unchanged when the user changes C to F or F to C.
- Added a new item DEF in the menu to store current configuration into flash as new factory default.
- Tstat displays "---" when the tstat can not detect the pic chip though there is a pic chip If the tstat is working on unoccupied mode,turn off all LEDs except OFF LED.

Ver26.5

- Shorten the delay between chatacters when tstat response.
- The baudrate does not change after changed C to F or revease.
- Fixed bug in TSS register.

Ver26.4

- Added "floating valve" feature.Added five registers from 280 to 284 to determine relay1 to relay5 output mode. Relay4 and relay5 be used to control floating valve by pulse mode if MODE_OUTPUT4 AND MODE_OUTPUT5 be set '1'.

Ver26.3

- Fixed the original PID's Pterm = cooling_db + heating_db + 10 to avoid always staying coasting mode when the user set the Pterm less than cooling_db + heating_db.
- Setup writes to registers 180, 181, 215, 216 - this will automatically adjust calibration values of these registers.
- Implemented an output strategy to PWM the relay coils in order to reduce the current and heat production.

Ver26.2

- Created custom lookup tables to allow for setup of almost any sensor on the analog inputs.
- Created staging control to allow for a custom number of heat and cool stages.
- Created a secondary PID that can be used to control the ouputs based on custom settings.
- Created system to store current settings as defaults by writing a value of 159 (0x9F) to UPDATE_STATUS register.
- Added partial menu lock.

Ver26.1

- Fixed bug in EEP location of UPDATE_STATUS register.
- Implemented display blank option by setting MODBUS_DISPLAY = 2.
- Added MODBUS_OVERRIDE_TIMER_DOWN_COUNT to hold the remaining value on the override timer.
- Created calibration terms for both the Analog In2 and for the Internal Thermistor.
- Created registers to show the internal temperature sensor IC and the internal thermistor value at all times.

- Combined CAL and CAE into one menu item, making it automatically adjust the correct calibration term based on the TSS.

Ver25.7

- Fixed factory defaults bug when setting defaults from keypad.
- Reset factory defaults value to 0 at startup.
- Added a prevention measure so that 0 or 255 cannot be written to the address.
- Gave serial interrupt priority to reduce communication errors
- Rearraged location of many RAM variables to free up space in the data memory.
- Adjusted HC setting so that when the external sensor is hotter, we are in heating mode, not cooling mode.

Ver25.6

- Adjusted filter so that the temperature does not jump when the new reading is very different.
- Fixed bug that caused the iterm to rollover from a positive to a negative number.
- Change plug_n_play address to register 10.
- Added support for the Tstat5B2.
- Made TEMPERATURE FILTER a variable rather than a constant. Set up new menu item, FIL, to control this variable.

Ver25.5

- Made change to auto_plug_and_play check_data() routine. Now tstat will not respond if randval != 1.
- Fixed bug in check hardware and model regarding 5A with rev0 hardware
- update_status missing in the ISP update routine in Ver25.3 update_status variable was missing when writting through modbus, therefore no update was possible given Tstat did not reset

Ver25.4

- Changed PID so that iterm will continue to influence the tstat even when the temperature is between the deadbands
- Fixed bug that prevented tstat from jumping into ISP mode.
- Fixed bug in PID.

Ver25.3

- Added 2 more button configurations - 2b and 4b. 2b is the same as 2, except cooling and heating mode can only be controlled via the serial interface. In 4b, the lower left button switches between cooling and heating mode, while the lower right button scrolls through the fan speeds - off, 1, 2, 3, auto.

Ver25.2

- Implemented blinking display whenever setpoint or fan speed is changed via the serial interface.
- If there is an E2 error, the tstat no longer freezes. Instead it simply does not start up any of the timers. The serial communications will still work. This is to allow you to adjust the product model and hardware rev registers.
- Changed the function of occupied/unoccupied.

  Changing the fan speed no longer changes the occupied or unoccupied state of the tstat. However, setting the fan to 0 will adjust the setpoints to the nighttime settings in the same manor as unoccupied.

  Bit 0 of register 184 – INFO_BYTE – is now both read and write. You can use this bit to put the tstat into occupied or unoccupied mode. (1 = occupied, 0 = unoccupied). Note: when DI1=2, you cannot change this bit via the serial line because the occupied state is solely controlled by the state of the Digital input.

  If the tstat is in unoccupied mode, pressing any button will activate the ORT. At this point, the tstat is still in unoccupied mode, but now, the setpoints are returned to normal daytime setpoints for the duration of the timer. Note: when DI1=1, pressing a button on the tstat will not activate the ORT, but will simply switch the tstat back into occupied mode.

Changing the occupied mode of the tstat automatically switches the Fan to AUTO

- Added 2 more states to the LED control register - COOL1_2_OR_3 and HEAT1_2_OR_3.

Ver25.1

- Combined heating and cooling PIDs into one PID.
- Separated Output scale, Out, into 2 output scales, Ou1 and Ou2. This is to allow distinct control of both analog outputs.
- Added read-only register items, Product Model and Hardware Revision, to give information about the device.
- Added menu item, dIS, to allow a choice of displaying either temperature or setpoint.
- Added baudrate selector to allow a choice of either 19.2 or 9.6 kb/s
- Added system of transition delays so that each output can be controlled separately.
- Added menu items, dSC and dCH, for short cycling and changover delays preventing the mode of operation from changing too quickly
- Allow higher temperatures and setpoints to allow for more flexible use with boilers.
- Analog inputs storage format is controlled by the 2 menu items, AI1 and AI2.
- IN1 has been separated into two menu items: TSS temperature sensor select, dI1 digital input select.
- Added flag during serial communication that makes the decimal point blink once for each packet received.
- Added system of registers to control the functionality of the LEDs.
- Changeover delay: dCH - this is a delay between when the tstat can change from cooling mode into heating mode or vice versa. The units are minutes 0-200.
- Cycling delay: dSC - this is a delay between when the tstat leaves cooling and then returns to cooling. Or when it leaves heating and when it can return to heating. The units are minutes. 0-20.
- Address in the menu. This is the first item, ADD. It can be set from 1-254.
- Setup bit1 of info_byte to reflect a watchdog reset. SRS will be displayed on power-up.
- Setup bit2 of info_byte to prevent reset during serial writes.
- Added code to control zero-voltage crossing based on External Interrupt 1.
- Added Low voltage test to prevent writing to the E2 chip if the voltage gets too low.
- Added memory storage when doing ISP.
- Added a 2nd query of PIC on startup incase first one was missed..
- Changed routines of PIC slightly to fix small bugs in I2C communication.
- Added feature to tSS and DI1 to allow for 2-pipe operation.
- Added Ort - OVERRIDE_TIMER to allow user to override the unoccupied mode of the tstat by pressing any button.
- Fixed bug with changing the address from the keypad menu.
- Changed units of output delays from 0.1 seconds to 1.0 seconds.
- Modified the update_status register to allow for a complete erase of the eeprom by writing 0x8F.
- Users now allowed to write to the COOL_HEAT_MODE register if tstat is in 6-button mode.
- Fixed eeprom erase function triggered by writing to register 16.
- Fixed serial write to AUTO_ONLY.

Ver25.0

- Fixed a bug concerning the SOP table when changing the setting of FAN
- Implemented control for tstat5D

Ver24.5

- Added code to switch between occupied and unoccupied mode upon trigger of digital or analog input
- Added menu items, night heating and cooling setpoint. nHS, nCS. Night heating and cooling deadbands changed to nHd, nCd
- Added response_data() before reset on certain serial com items so that tstat replies before resetting
- Changed code so that AUT=0 allows manual fan speeds. AUT=1 only allows off and auto

Ver24.4

- TXEN pin set low immediately. This will prevent interference on the 485 bus in case of E2 failure.
- Now make sure that the fan does not increase above the allowed level set by FAN_MODE.

- Changed the timing of responses to valid and invalid packets. 4ms needed to recover from invalid packet. 10ms delay when responding to a valid packet.

Ver24.3

- Added code so that an Analog Tstat can fuction as ON/OFF
- Increased the EVENT_QUEUE_SIZE from 3 to 6
- Changed display of nHS and nCS parameter
- Added EXTERNAL_SENSOR_1

Ver24.2

- Re-wrote modbus routines to comply with modbus standards and to reduce error rate
- Changed extern_operation_default table so that OUT4 and OUT5 correspond to table.
- Changed LED_r to look better when displayed.
- Adjusted temperature filtering. Now we read temperature every 500ms, rather than 1000ms. Also changed the TEMPERATURE FILTER from 10 to 5.
- Added register 180 - EXTERNAL_SENSOR_0 register. Now it is possible to read the external sensor regardless of IN1.
- PID was not getting updated when changes were made over the serial com.
- Changed default night heating and cooling setbacks from 3 to 10

Ver24.1

- Added sequence of operations in a separate table, can assign any output to a particular stage of heating or cooling
- Added Loc setting, to lock out the menu system from the keypad
- Added code to initialize_eeprom() to set the address to 1 as a default
- Increased Heating and Cooling Pterm Max from 100 to 255
- When an on/off module is detected, MUX pins will also be used to drive relay4 and relay5
- Changed the placement of the output scaling. Previously the scaling was taking place with each refresh_ouputs and accumulating. Now it only takes place once
- Valve should not return to 0 upon entering stage 2. This was corrected

Ver24.0:

Ver13: Added RS485 communications
Ver12 : PAD setting for various keypad arrangements
SHI, SLO settings, field adjustable maximum and minimum user setpoints.
Ver11: Feb10: SOP=14, 4 pipe changeover now works properly.
Ver10 Dec25: Add new self test (SOP=0) features to test E2 chip and external sensor
Ver10 Dec25: Add new range for outputs under FSO “Full scale output” setting.
FSO=0 0-10v FSO=1 0-5v FSO=2 2-10VDC Ver10
Dec25: Add new sequence SOP=16, one relay is energized whenever fan is on for interlock to DDC.
Oct: Add 2 pipe and external sensor as changeover sensor functions
Oct: Added 4 pipe fancoil sequences
Oct: °F is working properly now throughout the system
Oct: On power up, the software rev number scrolls over the display.
Oct: Added BUT menu item to allow for 6 button, 4 button and 2 button keypads
Sept: added new menu item FSO for FULL SCALE OUTPUT especially helps for use in transducer modes
FSO= 0, output goes from 0-10V
FSO = 1 , outputs go from 0-5V
Sept: Enhance calibration features of DAC, useful for transducer modes
Sept: New model of tstat TSTAT6 is line voltage, 220VAC 10amps, no external relay pack required.

Aug: Add VAV functions , not complete.
April2002: enhanced setpoint deadband calculations single and dual setpoint and heating/cooling setpoint features.
Mar 2002: added external temp sensor capability

Jan 2002: original version
Feb 2005: added LOC = 2; a new menu lock function for partial locking of the keypad menu system.
Feb 2005: AI1, AI2 = 2: Added ability to configure analog inputs to be reported as a number from 0 to 100% over the 0-5VDC input range. This is for future work with CO2 sensors.