T3000 Building Automation System

Table of Contents

Introduction

The T3000 Building automation system is a complete software and hardware solution to allow a building operator to manage and control large buildings. Equipment such as chillers, air handlers, boilers, fans, lights, pumps, dampers, valves and other mechanical equipment can be scheduled and controlled according to the user defined sequences. All the system status, temperatures and settings are displayed on custom graphics screens where the operator can proactively manage the facility. The system has powerful communications capabilities making it possible to integrate with other systems and scale to large campass style networks encompassing many buildings.
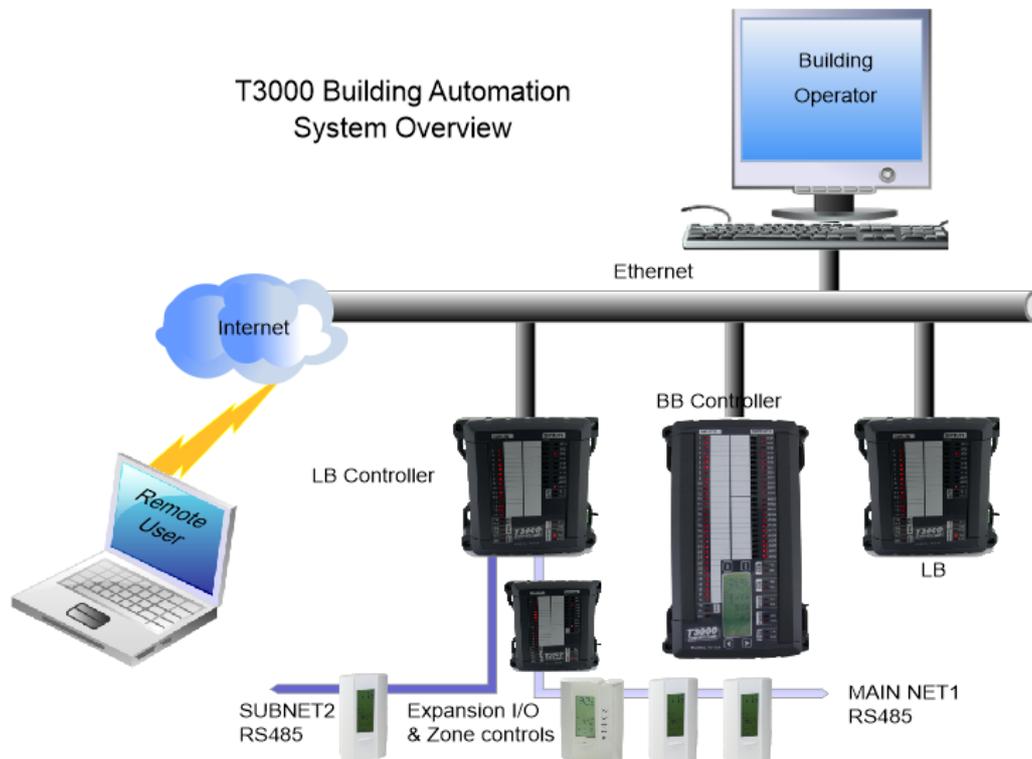
Once the operator is logged in they may access the entire system over the internet or locally from a PC inside the building. End users and tenants can also access particular zones which empowers them and reduces service calls.

Simultaneous users may access the system locally and remotely over the network. Alarms, system graphics and trend data are stored directly on the controller so there is no central server or single point of failure.

Interactive graphic displays, floor plans and custom menus make operation simple for any user. The T3000 hardware and software make it easy to configure and reconfigure the system throughout the life of the building. The system is primarily targeted at HVAC applications but is well suited to lighting, green houses and other light industrial projects. The system supports bacnet and modbus protocols which means its possible to integrate with the other systems in the building such as access control, lighting, power meters to name a few.

GENERAL OVERVIEW

The T3 series controllers are programmable controllers which contain all the logic for the application, hold trend log data, schedules and so on. The controller senses the building conditions using 'inputs' such as temperature sensors, switches and other sensors which in turn causes the 'outputs' on the controller to go on and off, modulate open or closed and so on. Each of these outputs is wired to pumps, fans, valves, dampers and other actuators which effect the commands from the programmable logic and keep the building in the target conditions. The controller operates stand alone, that is it doesn't need a computer or other server connected to it in order to perform the control tasks.

All the software required to set up a system and operate it on a daily basis is license free, the operators can copy the software to as many PC's as they like without paying any license or recurring cloud fees. The source code is also available on github for developers to use in their own projects.

All hardware communicates using industry standard Bacnet and Modbus protocols over Ethernet and RS485 networks.
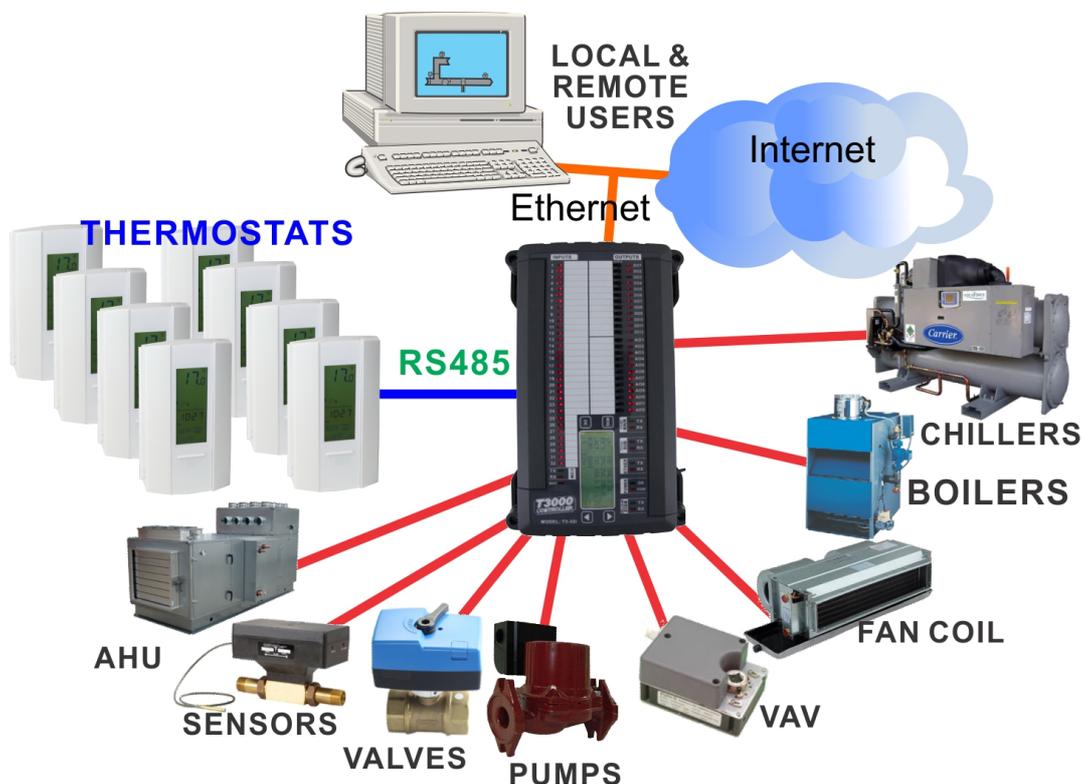
System Highlights

SUITED FOR BOTH LARGE & SMALL SYSTEMS

The system is equally at home in a single panel controlling a home as it is on a large building with several controllers connected over a network. Programming and managing a large system is the same as a smaller site, the operator can maneuver through the networked system in one seamless interface built from many controllers and devices which is presented as one continuous 'system'.
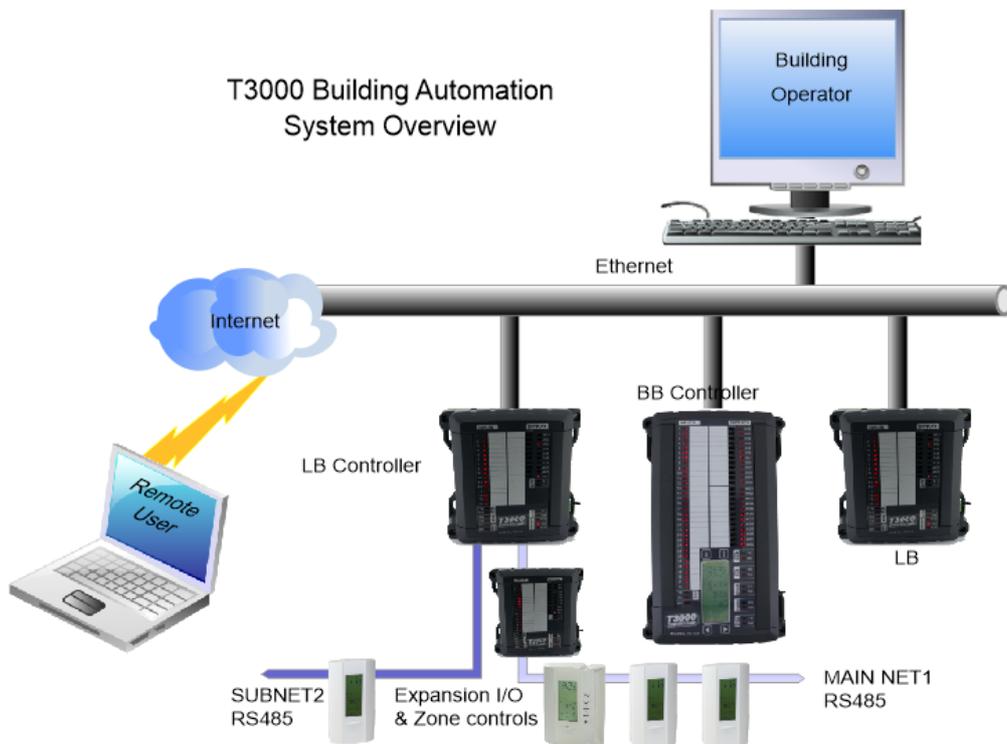
EASY TO USE

Once a system is installed the engineering staff can manage and evolve the system, testing out new control logic and adding new control points. The staff has all the tools on site with no license schemes to hold them back. The operator can add variables to displays, adjust programs and tweak the system in an iterative manner.

NETWORKING IS BUILT IN

Controllers are tied together over the backbone network which is standard Ethernet infrastructure. There are also two subnets on each controller for tying in thermostats and external i/o over RS485 twisted pair cabling over long distances, up to several hundred meters long. When the project is programmed and set up, the operator can refer to any point in the system by its network ID or custom user name without creating special network points, the system automatically takes care of all the lower level networking details and updates the network points automatically.

ACCESS FROM ANYWHERE

An optional GPRS radio module can plug into the controller for remote sites where there is no internet connection available. Alarms can be routed to a central dispatch office or to phones. Remote users can log in and assist the building operator with both users signed in at the same time. Anywhere you have mobile network coverage you can have remote access, use it to monitor boats or control bathroom lighting in parks for example.

GPRS
MOBILE
NETWORK

RADIO &
SIM CARD

FAULT TOLERANT

If a panel locks up due to a surge there is an on board watchdog which will reset the board automatically. Other panels in the system are unaffected by the outage, the latest values will be assumed for any network variables or temperatures and when the panel comes back online the data will automatically begin refreshing again. No user intervention is required for the panel to resume normal operation. All system programming is stored in non volatile flash which is automatically loaded on power up. The real time clock is backed up with a super cap that will keep the clock alive during a power outage of several days so there are no batteries to replace, ever.

FLEXIBLE WARRANTY

If a panel has a problem, send it to us and we'll do our best to fix it free of charge for the life of the system. There's no time limit to this offer. If the board cannot be repaired due to water damage or obvious wiring mistakes, or perhaps it is decades old and due for an update, just return your unit and you'll receive a healthy discount on a new one.

Control Concepts

A simplified control application is shown below where the controller maintains the temperature in a hot water tank. There is a temperature sensor to sense the temperature of the tank, a valve to add heat to the tank, and the programmable controller. The PC is used initially to program the controller and can be removed once the setup is done, the system will operate stand alone. In most buildings there will be a PC in the operator's office dedicated to the system though it is not required, everything will run just as well without the PC.
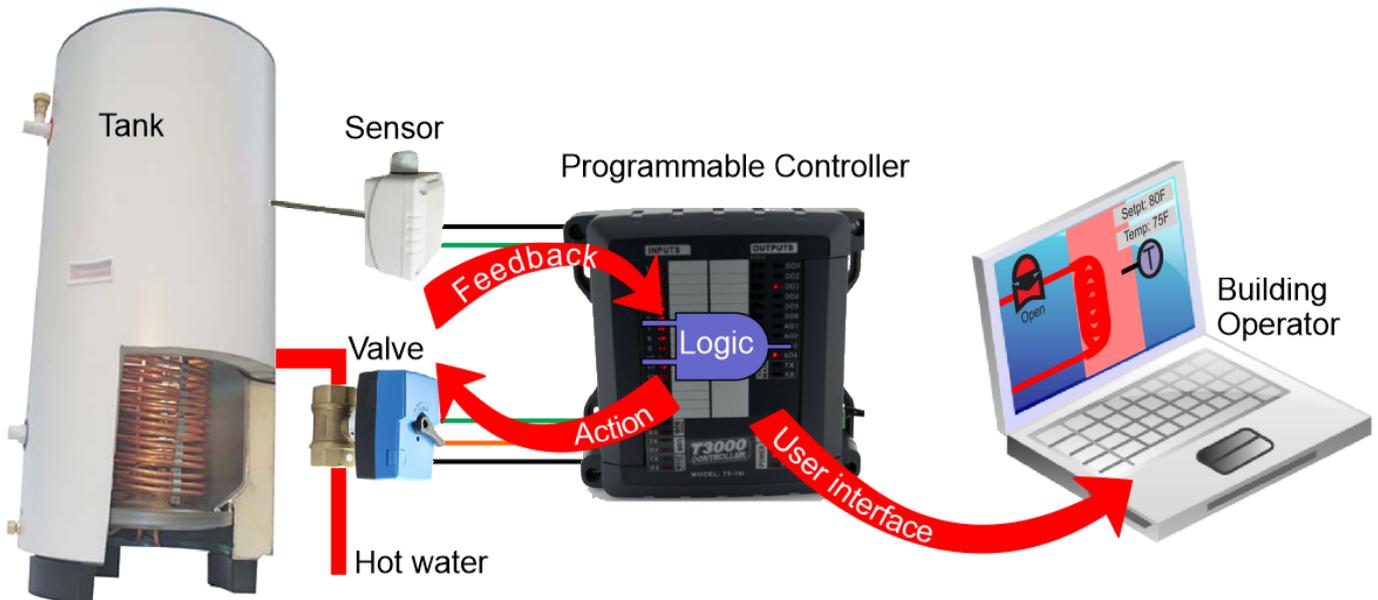
The controls technician will define the inputs and outputs, create some variables for setpoints, write a program and then finally do some graphics to tie it all together for the building operators. The T3000 software has both the engineering tools for the controls techs and the day to day operator interface combined into one program.

First define the input for the temperature sensor, give it a name and define the engineering units.

INPUT

Input	Panel	Full Label	Auto/Manual	Value	Units	Range	Calibration	Sign	Filter	Status	Jumper	Label
IN1	10	TANK TEMP	Auto	-40.00	Deg.F	10K -40 to 250	0.0	-	0	Open	Thermistor Dry...	TANKTMP

Then do the same for the outputs, in this case the hot water valve which brings in hot water from the boiler.

OUTPUT

Output	Panel	Full Label	Auto/Manual	HOA Switch	Value	Units	Range	PWM Period	Status	Label
OUT1	10	BOILER SUPPLY VALVE	Auto	AUTO	Off		Off/On	0	OK	DHWVALV

Next create a setpoint which is the target temperature for the domestic hot water

VARIABLE

Variable	Full Label	Auto/Manual	Value	Units	Label
VAR1	HOT WATER SETPOINT	Manual	125.000	Deg.F	HWSETPT

Add some logic to open and close the valve to maintain the temperature near the setpoint.

```
10 REM *****DOMESTIC HOT WATER **********
20 REM VALVE OPENS WHEN SUPPLY TEMP IS BELOW SETPOINT
30 REM VALVE CLOSES WHEN SUPPLY TEMP IS ABOVE SETPOINT
40 IF TANKTMP < HWSETPT THEN START DHWVALV
50 IF TANKTMP > HWSETPT + 1 THEN STOP DHWVALV
60 REM ******END **********
```

Finally tie all the inputs, outputs and setpoints together into a graphical user interface where the building operator can manage the system. All the tools which the controls technician uses during setup, and later the building operator to manage the system are included in the T3000 software.
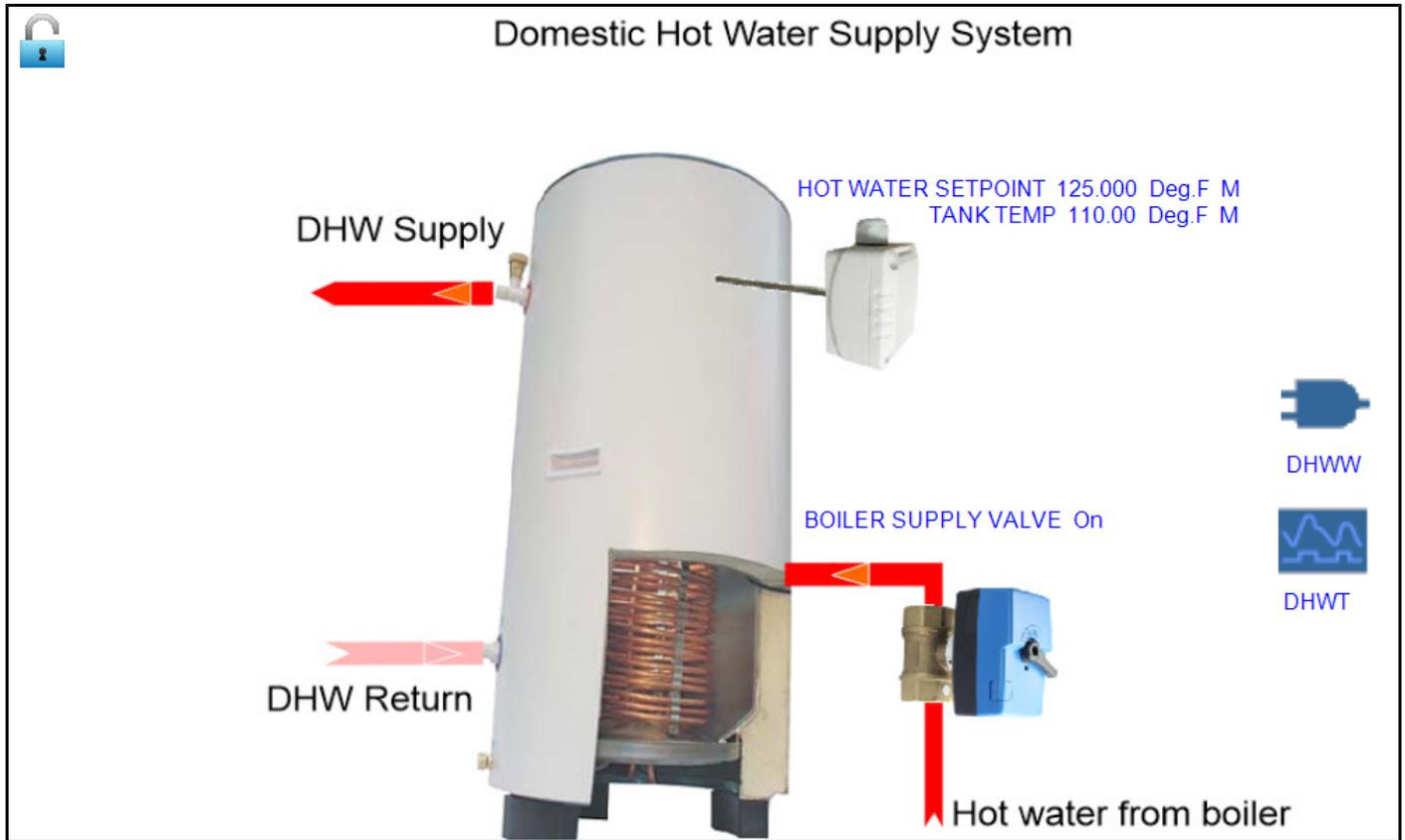
TYPICAL NETWORK DIAGRAM

A typical network diagram is shown here with 3 controllers connected together on an Ethernet lan. The PC connects into the lan for local operation while remote operators connect in over the internet. Zone controllers and expansion points connect to the T3-LB controller on the left over the RS485 network which is suited for longer distances and lower bandwidth connections.

T3000 Building Automation
System Overview

Building
Operator

Ethernet

Internet

LB Controller1

BB Controller2

2

Remote
User

1

LB controller3

SUBNET2
RS485

Expansion I/O
& Zone controls

MAIN NET1
RS485
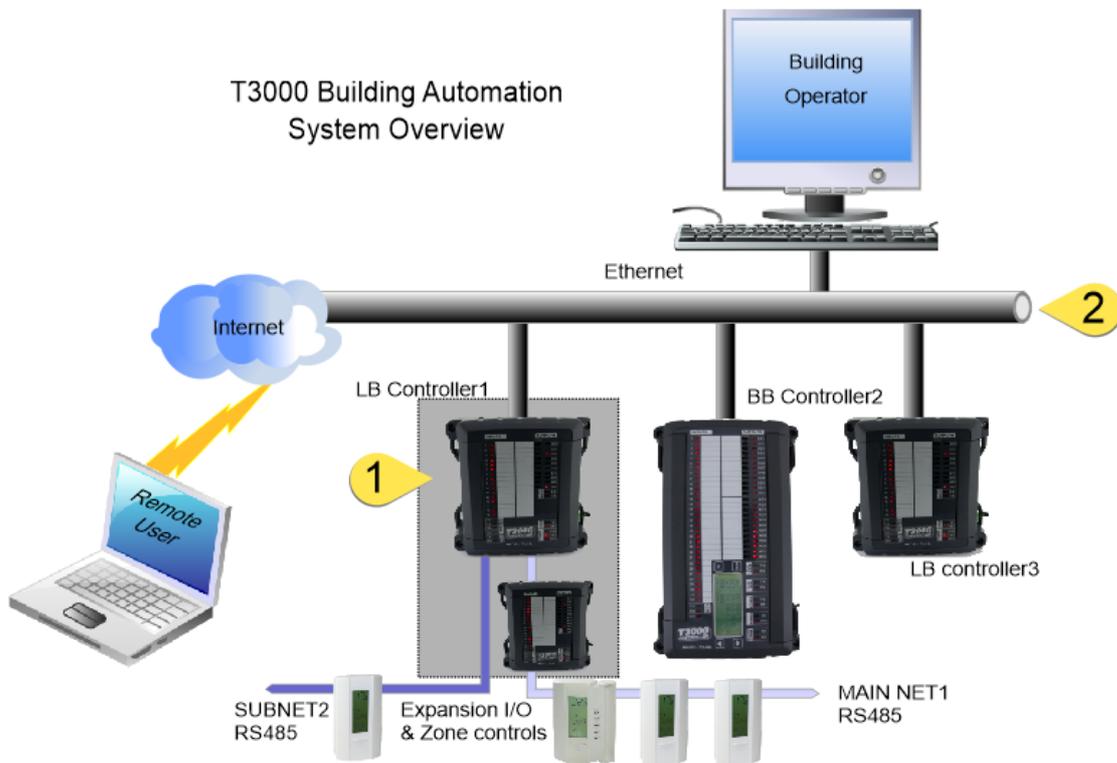
The LB controller at tab1 in the diagram above is an example of a controller with expansion points. The expansion modules are normally located in the same control cabinet as the controller so that a failure in the network cable will not cause problems in the rest of the building. However the expansion bus uses regular RS485 cabling so the expansion modules can be mounted long distances, hundreds of meters away if necessary.

The main Ethernet backbone is shown at tab 2 and is made up of the usual Ethernet infrastructure of cat5 cabling, hubs and routers. The control system will normally be on a separate lan but in smaller systems its safe to combine the building automation network with the office or home lan, the communications load of a typical control system is low compared to the typical PC user.

Micro Brother 'MB' model

4 Relay Outputs

8 Universal
Inputs

2 Analog Outs

2 RS485
Ports

Ethernet

Little Brother 'LB' model

6 Relay Outputs

16 Universal
Inputs

4 Analog Outs

2 RS485
Ports

Ethernet

USB Port

Optional
Zigbee/GSM/CDMA

Big Brother 'BB' model

INPUTS   OUTPUTS

12 Relay Outputs

32 Universal
Inputs

12 Analog Outs

2 RS485
Ports

Ethernet

USB Port

RS232 Port

Optional
Zigbee/GSM/CDMA

Each of the controllers has two RS485 ports which support a subnetwork used for expansion i/o, thermostats, vav controllers and sensors. A program in the controller can refer to the subnet objects by using a PANEL + "DOT" + SUBPANEL notation. For example there is a T3 controller with the panel ID=1, and a thermostat connected on the subnetwork with ID=2. We can refer to INPUT number 3 on this thermostat by typing 1.2IN3 in our programs and so on. The controller will set up the polling of the thermostat and update the object 1.2IN3 automatically.

The controllers support logging large amounts of trend log data and system graphics directly on an microSD card. The slot can accept up to 32 gigs of storage. Additionally, units shipped after May, 2017 have a 4gb chip soldered directly onto the CPU module so these models support microSD operations without the separate microSD card in the slot.

When the PC is connected, trend log data will sync in the background from the controller to the PC. Rather than sending large amount of data over the network each time the user wants to see a trend log, the cached data from the PC is used which speeds up the viewing times. The system will also replicate the graphics and floor plans in a similar way, once an operator sets up a graphical template on a controller, a copy is sent out to the controller microSD card. Any operators logging in from other PC's new to the system will be able to see the same graphics without having to manually copy files to each of the PC's. There is no requirement to have a dedicated server running for any of the data replicating services, its all handled by the T3000 software and hardware automatically.

Overview

SOFTWARE

The T3 controllers have inputs, outputs, schedules and other features baked into the firmware which allow you to set up a typical control system quicklys. Here is a run down of the software feature list for the T3 controllers:
.

16 Control Basic programs of up to 10k total size

Each controller can be expanded with external T3 modules, up to a maximum of 64 inputs and 64 outputs per controller

128 Variables , these are used as virtual points in your calculations, displays and trend logs.

16 PID Controllers, used to control a process to a setpoint.

32 Trend logs, use these to see the temperatures and state of the building over extended periods.

Runtime totalizers and event logs

16 User screens [TBD: Expand to 128], used for floorplans and system diagrams.

8 Weekly schedules [TBD: expand to 16], used to schedule occupied periods.

4 Annual schedules [TBD: expand to 8], used to override occupied periods on holidays.

16 Variable arrays [TBD: add arrays]

8 User name/ password combinations [TBD: expand to 50]

5 Custom tables , used for custom sensors

8 Custom digital engineering units , on/off is standard but you can add custom states like purge/fill for example.

Digital/Analog alarms , used to alert the operator about failed equipment.

Programming editor

Graphical user screen editor for custom displays

JPG, BMP, PNG file format for floor plans, mechanical system diagrams and icons

T3000 OPERATING SYSTEM SOFTWARE

The T3000 software is used by both casual users for day to day interaction with the building as well as experienced operators and contractors when they are setting up a system. The software runs on any PC running Windows 7,8 &10 and there are no licensing restrictions allowing you to copy the software to as many machines as you like. There are no linux or Mac versions. Third party software can always be used with T3 controller hardware by integrating over Modbus and Bacnet protocols, but there will always be a few procedures that are best accomplished using the T3000 software.

GETTING AROUND THE SYSTEM

Most of the system configuration is done in a series of tables which the user can navigate and edit like working in a spreadsheet. The tables are updated dynamically with real-time information even as items are being edited. Navigate in the program using the menu system and mouse. There are hotkeys for many operations so experts can get around the system quickly, see the next section on the Menu and work areas for more info.

[TO BE DONE] There is a command line which is brought up by pressing the space bar and allows you to jump to 1INS for example, to quickly view the inputs on panel 1.

ALARMS

Alarms are generated by the T3 controllers based on readings and status of the field sensors. When a certain temperature is too hot for a certain period for example, the alarm will be generated. These alarms are programmed in the Control Basic language using the ALARM and DALARM statements. These statements allow the user to define their own logic and timing of the alarm statements which can be as complex or simple as you like. A typical simple alarm for the room temperature is shown here, if the room is over or under setpoint by 10 Degrees for more than 100 seconds the alarm will trigger.

200 DALARM , TEMP101 - SETPT101 > 10 , 100, ROOM101 IS VERY HOT
210 DALARM , TEMP101 - SETPT101 < -10 , 100, ROOM101 IS VERY COLD

When an alarm occurs, an alarm symbol is displayed at the bottom right of the screen. Clicking on this symbol brings up the alarm log where the user can choose to acknowledge or clear the alarm. The alarm message is logged to the microSD disk locally as well as on the hard disk of any PC connected to the system. Each alarm shows the panel number from which the alarm originated, the date, time and alarm priority.

More details on the alarm statements and alarm handling are described here.

GRAPHICAL INTERFACE

The T3000 operating system can display floor plans and mechanical diagrams of the building using bitmaps which can be generated using any popular graphics package. Use a program such as Autocad, Corel Draw or Visio to prepare the graphic templates of the building and systems. These jpegs are stored on the hard disk of the PC and are used in the user interface of the particular building. Real time data is overlaid on the picture to show the temperatures, setpoints and so on of the system. You can move elements around on the display, add items and reconfigure the system interactively as the building operator works in the system. The screens can be nested to several levels deep enabling an operator to navigate through the entire building via the linked pages. Icons pointing to the trend logs, programming, and any associated item can be placed on the screen, there are a few canned icons and the user can create their own custom icons.

1.8.4   CONTROL BASIC

Control Basic is the programming language used for the logic running in the T3 controllers. There are many new modern languages which can be considered superior for their particular tasks but when it comes to programming a building, control basic gets the job done and its easy to learn and maintain later on. A sample of the programming goes like this:

```
10 IF SCHEDULE THEN START FAN101

20 IF TEMP1 < SETPOINT1 THEN START HEAT1

30 IF TEMP1 > SETPOINT1 + 1 THEN STOP HEAT1
```

Looking at these three lines you can see the programming is easy to read, the usual 'sequence of operations' from the mechanical system designer can be translated into basic programming quickly using the many built in control basic functions. The programming is stored in the local flash memory of the controller and starts running as soon as power is applied to the panel. One special feature of the T3 system is the ability to tweak the program while the building is running, there's no need to jump between software tools or take the panel offline. Just edit the program and hit the send key, the program is sent to the controller and runs immediately.

Menus and Work Area

The T3000 software has the menu system and quick access icons along the top bar, the tree on the left, status line at the bottom. The following section describes these zones in more detail.
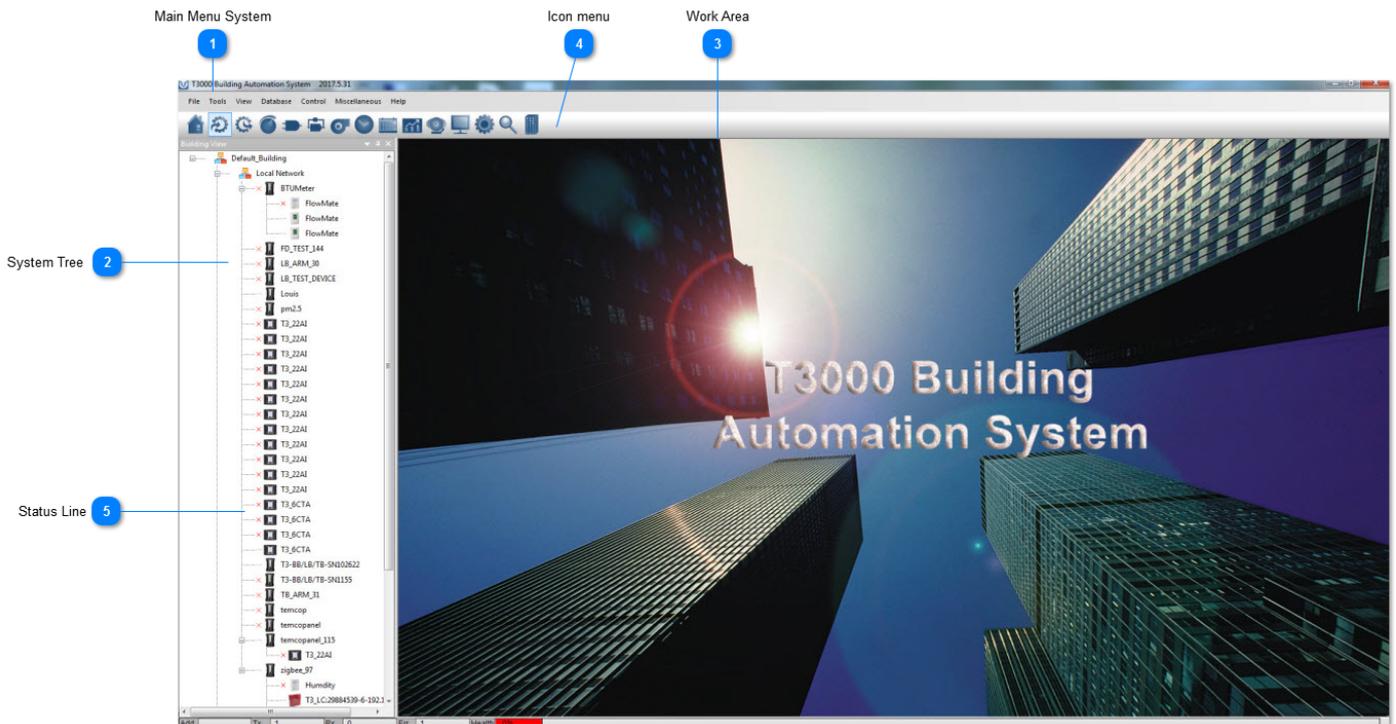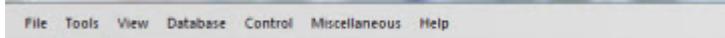
Main Menu System

The main T3000 Menu system shows along the top of the screen, logically grouped by function.

> [TO BE DONE] Keyboard shortcuts for these items.
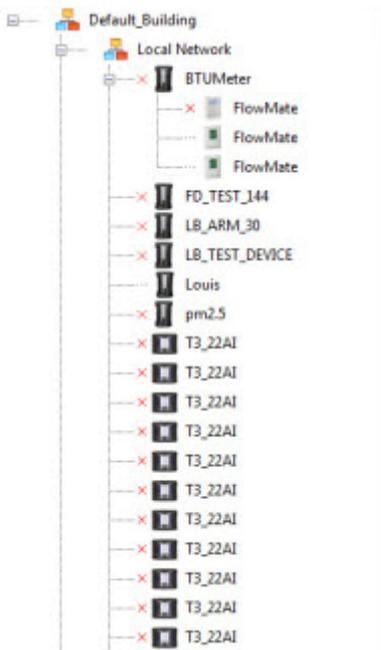
System Tree

The System Tree section shows along the left hand panel. The tree can be turned off and sorted & edited in different ways. Items which are offline will have a small red 'X' near the node. The tree is described in more detail here in the System Tree section.

Work Area

Variable	Full Label	Auto/Manual	Value	Units	Label
VAR1	HP HEAT FROM STORAGE	Auto	Off	Off/On	HT1STR_M
VAR2	HP HEAT FR SOLARCOL	Auto	Off	Off/On	HT1COL_M
VAR3	GAS HEAT	Auto	Off	Off/On	HEAT2_M
VAR4	SUMMERTIME	Auto	On	Off/On	SUMMER_T
VAR5	CLCT - STORE *DELTAT	Auto	0.000	Deg.F	CLSTR_DT
VAR6	CLCT TO STORE *DIR	Auto	Off	Off/On	CL2STR_M
VAR7	CLCT TO STORE *HP	Auto	Off	Off/On	CLSTHP_M
VAR8	WASTE HT TO STORAGE	Auto	Off	Off/On	WHT2ST_M
VAR9	WASTE HT TO COLECTR	Auto	Off	Off/On	WHT2CL_M

This is the main work area which can be text based for working with grids as shown on the left, or it can be graphical for working with the operator displays and floorplans as shown on the right. Setting up graphical displays is described in the Graphical Screens section

Icon menu

Shortcuts to the most commonly used work areas of the system. Inputs, Outputs, User screens, trend logs and so on all all presented here in icon format. There are also these same items available from the text menu system.

Hover text will show the function of each of these icons when you place the mouse over the icons without pressing any buttons.
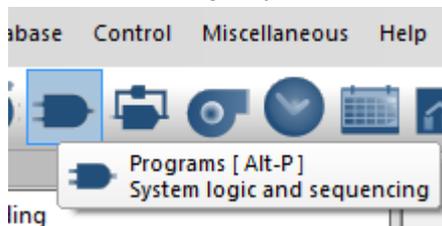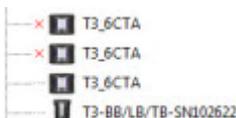
Status Line

The communications status line shows the address of the device which is currently communicating in the "Add"ress box. The RX and TX numbers show how many packets have been sent to the current device in the current session, each time you click on a different device the numbers will reset to zero. The ERR box shows the number of errors in the current session with the current device. The Health box shows the percentage of successful packets for the current session for up

to 100 packets. When the session hits more than 100 packets the earlier packets are pushed out of the tally, only the latest 100 are used in the math. Finally the status line shows a log of the current communications activity.

System Requirements

Minimal System Requirements

The T3000 application is not particularly demanding on the CPU, a bare bones Win10 system will run the application just fine. There is no linux or Mac version of the software. Windows XP is no longer supported.

• Microsoft Windows 8 thru Win10
• SVGA or better, color display
• 1Ghz Hz processor
• 1G RAM
• 2 GB free hard drive space
• Ethernet
• USB to RS485 converter is recommended for testing RS485 communications.

Keyboard and Mouse

This chapter explains the use of the keyboard and mouse, and describes how to move around within the T3000 menus. Some basic commands are discussed, as well as how to load T3000 software from the and how to get the system running.

Throughout this manual the following notation formats are used:

1.   Words in capital letters are keywords and must be entered exactly as shown, for example, they keyword: OUT1 cannot be typed in lower case or any mixture of upper and lower case.

2.   Items in square brackets [ ] are optional. (i.e. [LET] A = 10)

3.   Program listings appear in a line printer font as shown:

        10 IF SCHEDULE1 THEN START FAN1 ELSE STOP FAN1

2.1   USING THE KEYBOARD

The system is driven mainly by the keyboard to allow working in the field without a mouse as much as possible. Keyboard shortcuts also speed up many operations for experienced users. The following is a list of the important keys and their functions:

The Arrow keys, ¬, ↑ , #, #.
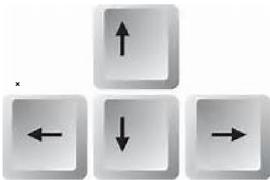
Located on the keypad to the right of the main keyboard. The cursor on the display will show which item has the focus, the **Up**, **Down**, **Left**, and **Right** arrow keys move the focus around within the various grids and menu and the programming editor.

The ENTER key, # **.**

The enter key is an important key to become familiar with, changes are easy to make but sometimes the user can end up in edit mode inadvertently by hitting the enter key. The Enter key uses the normal convention to accept a choice in the menu system or dialog boxes. In the programming editor, the Enter key will start a new line. The behavior which will take a little getting used to is for editing items in the grid. When the user is in one of the many grid views and the focus is on the description field of an item like 'AIR HANDLING UNIT4' as shown below, hitting Enter begins the editing mode. The color of the item changes from purple to a blue indicating edit mode is now in effect. Make the changes and then hit enter again. The item will flash red briefly to show the item has been changed and data is being sent out to the controller. Finally, the cursor changes to purple again to show that the change was written successfully.

   The cursor is in a text field > hit enter > blue shows we're in edit mode >enter >red shows changes are being sent to the controller > purple again means the changes are sync'd with the controller.

The Escape key, Esc.

During editing you may decide to back out of the changes, the ESC key will let you revert back before the changes are sent to the controller. In most cases this will be only a few characters of text which are changed but during editing of a basic program for example there may be several lines of code already changed in which case the system will ask you to confirm.

The TAB key.

This key will be used to move the cursor from one field to the next in many of the grids and various dialog screens within the program.

The Insert key, Ins.

This key is handled a little unconventionally like the 'Enter' key is but saves mouse clicks and helps get around the system quickly. In many of the grids such as Programs, Weekly Routines, Graphics Screens, and Trend logs there is a detail page for each of these items which pops up with the INSERT key. The example below shows the Graphics screens grid, hitting INS drills down into the air handling unit graphics detail display. Note that double clicking with the mouse also has the same action.
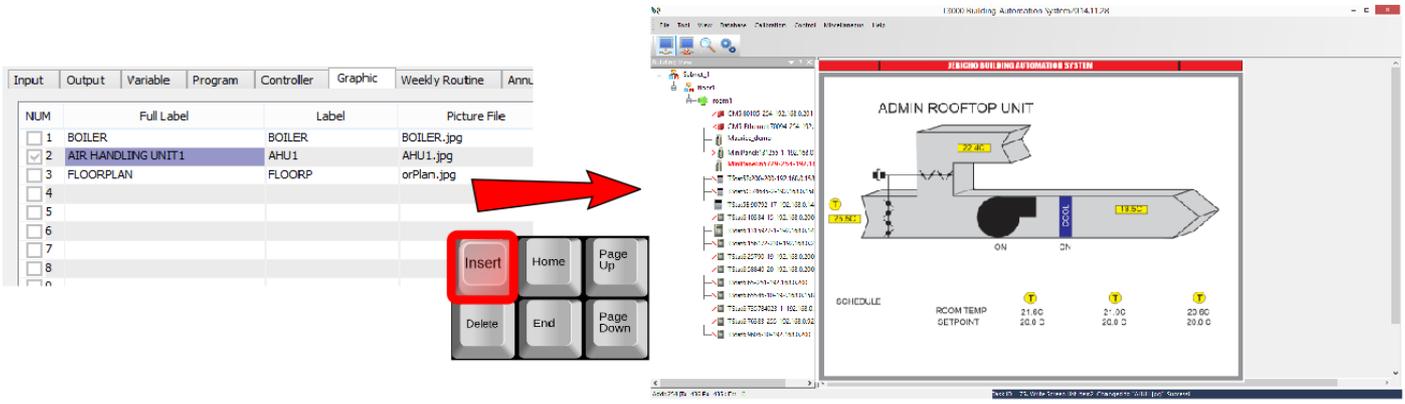
HOT KEYS

While working in the T3000 menu system you will notice a number of "Hot Keys" which can be used directly from the keyboard as well. These let the experienced operator move around the system quickly and minimize mouse clicks. The hover tool tips show up under the item also shows the hot key combo.
The following is a brief description of each of the hot keys:

Control	Miscellaneous	Help

Graphics — 1
- Graphics (Alt - G)
- Programs (Alt - P) — 2 Programs
Inputs — 3
- Inputs (Alt - I)
- Outputs (Alt - O) — 4 Outputs
Variables — 5
- Variables (Alt - V)
- Loops (Alt - L) — 6 Loops
schedules — 7
- Schedules (Alt - S)
- Holidays (Alt - H) — 8 Holidays
Trend Logs — 9
- Trend Logs (Alt - T)
- Alarms (Alt - A) — 10 Alarms
Remote Pointers — 11
- Remote Points (Alt - T)
- Customer Units (Alt - U) — 12 Customer Units
Configuration — 13
- Configuration (Alt - E)

1 Graphics

Graphics (Alt - G)

Graphics show Floor plan,system diagrams,graphical diaplays,and the hot key is Alt-G.

2 Programs
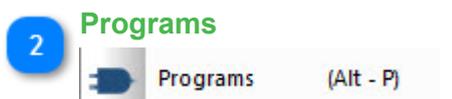
Programs (Alt - P)

Programs show System logic and sequencing,and the hot key is Alt-P.

3 Inputs
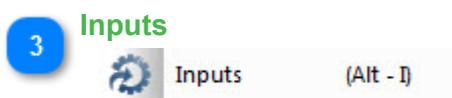
Inputs (Alt - I)

Inputs show sensors and feedback points wired to the controller,and the hot key is Alt-I.

4 Outputs
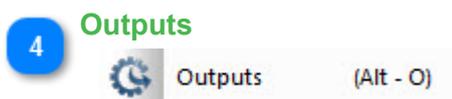
Outputs (Alt - O)

Outputs show Valves,relays and actuators wired to the controller,and the hot key is Alt-O.

5 Variables
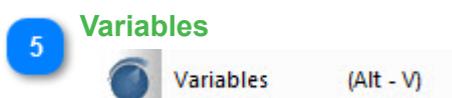
Variables (Alt - V)

Variables show setpoints,counters,timers and other system defined items,and the hot key is Alt-V.

6 Loops

Loops (Alt - L)

PID Loops show Proportional-Integral-Derivative feedback control loops,and the hot key is Alt-L.

7 **schedules**
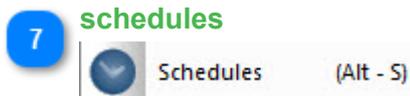
Schedules (Alt - S)

Schedules show Daily on and off events for the week,and the hot key is Alt-S.

8 **Holidays**

Holidays (Alt - H)

Holidays show Holidays and days with special schedules during the year,and the hot key is Alt-H.

9 **Trend Logs**

Trend Logs (Alt - T)

Trend Logs show Graphs and historical data,and the hot key is Alt-T.

10 **Alarms**

Alarms (Alt - A)

Alarms show Current and past alarm events,and the hot key is Alt-A.

11 **Remote Pointers**
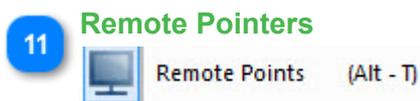
Remote Points (Alt - T)

Remote Points shows Network points from other nodes used by the controller,and the hot key is Alt-T.

12 **Customer Units**

Customer Units (Alt - U)

The hot key of Customer Units is Alt-U.

13 **Configuration**

Configuration (Alt - E)
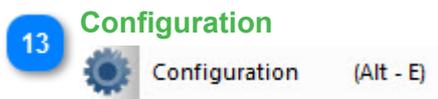
Configuration show setup and advanced parameters,and the hot key is Alt-E.

Open Source Project

The T3000 front end is one of the few building automation systems on the market which is open source. Primarily the software is designed to work with Temco Controls building automation controllers and sensors but the source code is freely available and anyone can integrate devices from other manufacturers. Building owners, controls contractors and other manufacturers in the industry are encouraged to join the project and contribute to making the software more universal and more useful. You may use the project with no restrictions for both private and commercial use.

The latest compiled version of the software is here: T3000Software.zip
There is no signup required, the software is license free and fully functional.

The source code is hosted on the Github open source repository site: T3000 Open Source Project
This is a C++ project using Microsoft Foundation Classes.

There is another version of the T3000 project under way targeting Linux environments, it is written in C# and is also hosted on github however the project is just in the beginning stages: T3000 Cross Platform Project

Parties interested in cooperating on the projects are encouraged to contact us and we'll set you up as a contributor.

T3 Controller Hardware

T3 Controller

The T3 controllers are stand alone programmable controllers which can operate on their own independent of the PC and don't require servers or cloud services to perform their job. Each of the panels has an Ethernet port and two RS485 subnets. Communications protocols include Bacnet and Modbus both of which operate over ethernet IP and the RS485 ports.

Expansion i/o can be added using expansion modules which come in a variety of i/o types and counts to fit the application. The expansion modules are usually located in the same control cabinet as the controller but the designer is free to locate them remotely, they connect to the main controller over twisted pair RS485 so they can be located several hundred meters if necessary. There is an Ethernet port on most models of the T3 expansion modules, this port can be used by integrators for managing the i/o blocks over IP but when integrating to the T3 controller, these should be connected over RS485.

The T3000 Controller hardware is shown below. The hardware has many hardware features which make the controller easy to commission and manage over the life cycle of the building. Each output has a hand-off-auto switch which will override the controller logic during troubleshooting. There are plenty of LEDs to see the status at a glance, all the terminal blocks are pluggable. Each input and output has its own pair of terminals to make wiring simple, most devices are connected to the controller with a single wire going to a single terminal. There is no need to join up grounds on an external ground lug. Each RS485 network connection has a separate upstream and a downstream connector to make RS485 debugging easy. In summary, this controller hardware is made with the building operator and controls technicians in mind, making both installation and maintenance easier.
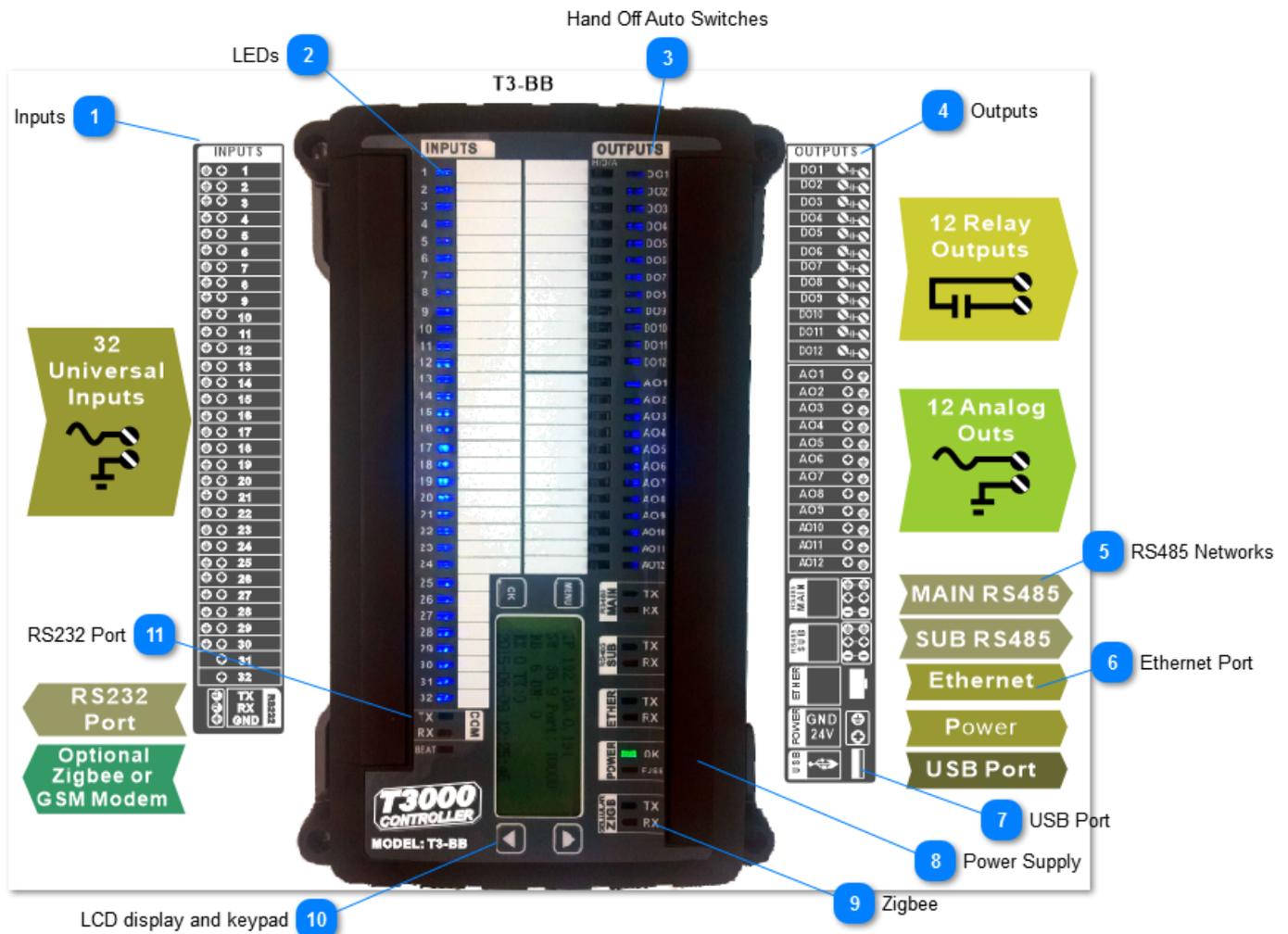
The controller is powered by an Arm M3 core processor operating at 120Mips which is plenty of power for the typical controls application. Communications are reliable over all the ports simultaneously while still running the user prgramming at least once per second, no matter how much logic the user has worked up in the application.

The following section describes the main features of the controller. More information on the T3 hardware is described in the next section entitled Hardware Architecture.

.
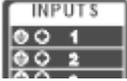
Inputs

The Inputs are called 'universal inputs' which means by selecting the appropriate range you can deal with control signals from most any typical HVAC field device you are likely to encounter: dry contacts, thermistors and transducers can all be wired to the intput terminals with no interfacing required. The front end software lets you set up the range and the engineering units for each of the inputs. There are no hardware jumpers to set as is often the case with other controllers, sensor type and range are all selected in software from the GUI.

Custom ranges can be set up for transducers, for example a 100psi transducer with a 4-20ma signal can be set up easily in the front end software including the custom range and custom engineering units.

Pulse counting is supported at 1hz on any input of the T3 controllers and many of the T3 i/o expansion modules as well. There are some 1khz high speed pulse counters some inputs as well, see the pulse counting section for more details.

Each input has an LED for visual indication of the status of each point. The inputs are protected against surges from static and lightning, as well as ground loops due to wiring mistakes.

LEDs

There are many LED's which show the controller status at a glance, all of the inputs and outputs have their own LED. All of the communications ports have a TX and RX led. The power supply, heartbeat and fuse also have leds. The analog inputs and output LEDS will dim proportionally to the voltage on the terminal. Inputs which are set to an unused range will be off.

The LEDS can be also configured in the reverse sense to accomodate normally closed relays. Normally the LED will be off when the contacts are open and on when the contacts are closed. When the input is configured for the reverse sense the LED works in reverse from that, ie. the LED is on when the contacts are open and off when the contacts are open.

Hand Off Auto Switches

Each of the outputs has a manual override switch. Should the panel go down or the operator need to do some testing, the outputs can be managed with switches. If the operator forgets to put the switch back into the auto position an alarm will sound at the operator terminal as a reminder. The current state of all the hand off auto switches is shown in the GUI and also reflected in the relevant modbus registers / bacnet variables.

4 Outputs
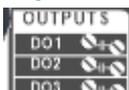
There are two types of outputs, the relay outputs and analog 0 to 10V outputs. The relays are dry contacts with a UL listed rating of 120vac at 5 amps. The analog outputs are 0-10V signal with 100ma drive capability. The analog signal can be wired directly to an external 12VDC relay and configured as relay outputs if the application calls for more binary outputs.

The analog outputs are also capable of sourcing 20ma per channel at the zero volt output setting, this enables the use with dimming ballasts which require an actual sinking current to turn down to the fully dimmed or 'off' state.

5 RS485 Networks

There are two RS485 networks on each controller, one is labeled 'main' and the other 'sub' though they are functionally equivalent. Typically the subnet is used for expansion I/O and the main is used for controller-to-controller level connections. The ports can be set up for Modbus RTU protocol and Bacnet MSTP protocol at baud rates up to 112k. Each of the networks can run a different protocol, MSTP on the main and Modbus on the sub for example.

To make debugging in the field easier there are two connectors for each of the two RS485 ports which are connected internally on board the controller. Conceptually they can be thought of as one connector going to the 'upstream' devices and the other goes to the 'downstream' devices though they are connected inside the controller to each other.

The ports are not galvanically isolated but there is plenty of hardware protection against wiring mistakes and lightning surges. When there are ground loops in the system the two fuses will open up to protect the controller, once the loop is sorted out the fuse closes and communications are restored. There is more information on setting up and the use of the RS485 ports here:
Communications Ports

6 Ethernet Port

The Ethernet port is 100Mbaud/10Mbaud auto switchover. The controller can talk both Bacnet and Modbus protocols over IP, simultaneously if the application calls for it.

7 USB Port

The USB port is available for ad-hoc connections and debugging, it supports Modbus RTU slave only at the moment. When there is an optional GSM modem installed on the controller the USB port will be occupied by the GSM functions.

8 Power Supply

The controller can operate on AC or DC from 12V on up to 24V. Power consumption is 10 watts maximum with all relays on and all analog outputs loaded to 100%. The OK led will light up when power is applied and the FUSE led will light up when there is a ground loop or hardware problems with the controller. The fuse is self resetting and operation will resume normally once the wiring issues are straightened out.

There hardware is well protected from surge and brownout conditions. When power goes outside the normal bands the controller will gracefully power down without the relay outputs chattering.

Zigbee

The T3-BB and T3-LB models support an optional zigbee module which can connect wirelessly to zigbee thermostats and other devices over a mesh network. Communications activity will show on the two LEDs shown.

LCD display and keypad

There is a display on the T3-BB model which shows the current IP address and communications activity. Currently the keypad has no menu system built in, all settings are accessed through the communications ports.
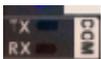
RS232 Port

The T3-BB has an RS232 port which supports Modbus and built in drivers for integrating with a few devices such as power meters, weigh scale and so on. The port is currently not available to the user application other than these built in modbus and fixed purpose functions. If you have an application which calls for communicating to a particular device just let us know and we'll be glad to integrate it.

Hardware Architecture

Here is an overview of the hardware architecture of the T3 controllers which is comprised of three main boards: the CPU core module, a lower PCB which has the outputs and the top PCB which deals with the inputs. There are two auxilliary CPUs communicate to the main CPU over I2C connections and handle the peripherals, one on the top board primairly for the inputs and one on the lower board mainly for the outputs.

.
.

.
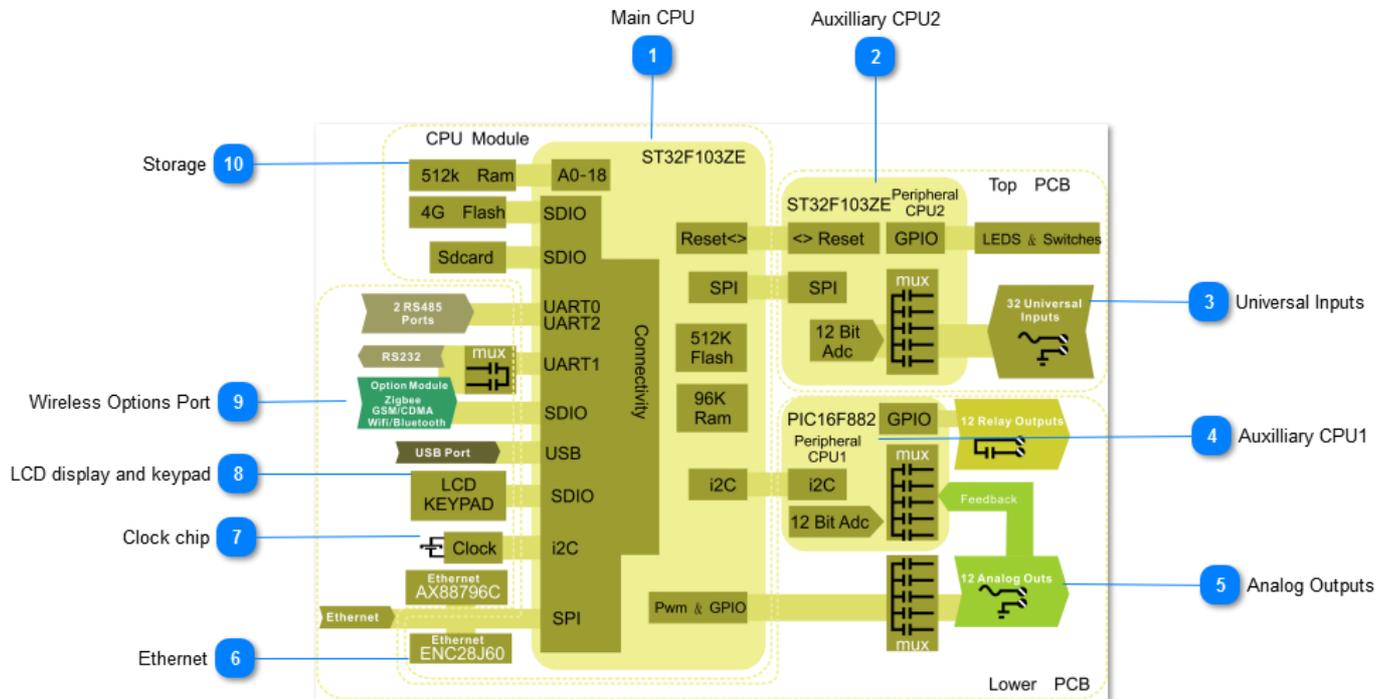
① Main CPU

ST32F103ZE

The heart of the T3 controllers is a 32 bit, ST32F103ZE Arm processor from the M3 core family. This is a high performance CPU operating at up to 120 Mips which is plenty of processing power for controlling typical HVAC applications while maintaining fast and responsive communications. Earlier versions of the product shipped before Q1 2017 use the Asix AX11015 which is a general purpose CPU running at 100 Mips with an on board Ethernet controller. Due to some issues handling multiple connections over TCP we migrated to the Arm based chip which has solved that as well as opened up other capabilities thanks to the extra serial ports on the Arm chip.

The 200 pin SODIMM socket which hosts the main CPU can also be populated with a 'Raspberry Pi compute 3' module which runs the Rasbian operating system, and now also the Win10 IoT environment. There is currently not much of software support for the Pi version but the hardware is in production and the open source project behind it is slowly gaining steam: T3000 Cross platform project

2 Auxilliary CPU2

ST32F103ZE Peripheral CPU2

This CPU resides on the upper level PCB and is in charge of the analog inputs as well as monitoring the LEDs and hand-off-auto switches. It communicates to the main CPU over a dedicated I2C link. The auxiliary CPU also plays the role of the watchdog, should the main CPU lock up due to an electrical surge the auxilliary CPU, after a period of approximately ten seconds, will reset the main CPU by toggling the reset line. When normal operation resumes the on board logic and programming are automatically loaded and operation resumes without user intervention.

3 Universal Inputs

32 Universal Inputs

The universal inputs are managed by the auxiliary CPU#2 on the top PCB. All inputs are digitized by this CPU using 12bit a/d conversion which are then read by the main CPU over the I2C bus. The inputs can be practically any type of input such as thermistor, dry contact, 0-3V, 0-10V and 4-20ma signals. The CPU manages the routing of these signals through the appropriate hardware by routing the signal from the field devices through various on board resistors and voltage dividers. Because of this hardware there is no need to physically adjust jumpers to set the type of signal, the setting is performed in software and the CPU will manage the muxing of the signals in firmware. More can be read about the input signal handling here: Inputs

4 Auxilliary CPU1

PIC16F882 Peripheral CPU1

The Auxilliary CPU on the lower PCB is in charge of enabling the relays and monitoring the analog output feedback signals. This chip communicates to the main CPU over a dedicated I2C connection.

5 Analog Outputs

12 Analog Outs

The analog outputs of the T3 controllers is derived from a pair of PWM signals on the main CPU module multiplexed to several channels using a multiplexor chip, U11 in the partial schematic below. Each signals is buffered and smoothed out by a charge storage capacitor, C46 in this example for chann is occurring very quickly the single PWM channel can produce several analog output signals which

The analog outputs are buffered by an opamp and two transistors to allow sourcing of up to 100ma The sinking transistor Q22 provides extra drive to zero volts when connecting to adjustable speed ballasts, these devices may require up to 20ma in actual sinking current to get the signal down to f

The D5 double acting transorb provides fast protection against static electricity and wiring mistakes begins to conduct the fuse F11 will warm up, eventually going to open circuit and protecting the D5 mistake is corrected the current goes down and F1 cools off and begins conducting again. Norma

Finally, there is an analog feedback signal which is routed to the auxilliary CPU#1, this improves th
outputs to less than 0.1V even with varying output loads. This unique feature allows the T3 able to
hardware failures on the analog outputs greatly assisting the commisssioning crew and later, the m
troubleshooting.
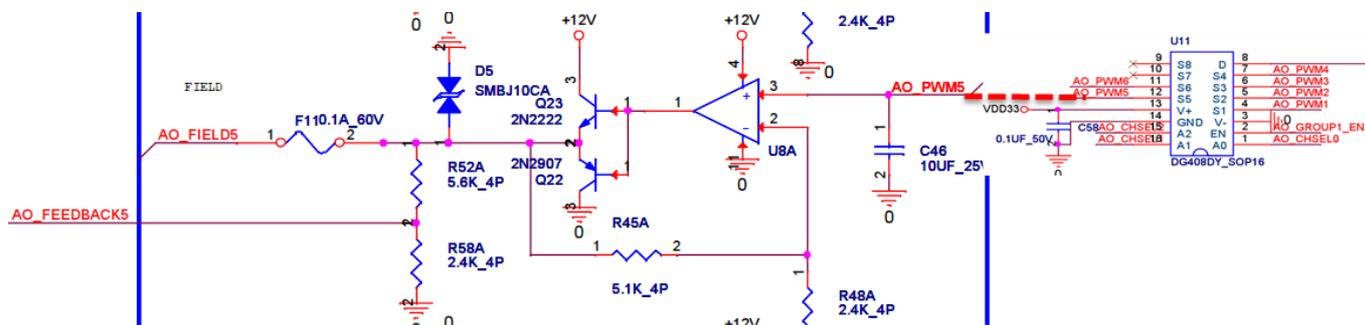
6 Ethernet

The Ethernet coprocessor is a separate chip from the main CPU, greatly reducing the computing load on the main CPU. The ENC28J60 chip is made especially for embedded devices providing 10/100 Mbps thoughput with packet buffering and checksum off loading. The chip supports auto MDIX changeover which means you can connect a PC directly to the controller with either a regular Ethernet patch cable or a crossover cable, the controller will work with either setup. The Ethernet chip communicates to the main Arm CPU over a high performance SPI slave interface.
There is a second Ethernet chip on board the main motherboard which is the ax88796C and is similar to the ENC28J60 and is not normally populated. If you;'re working with the Pi version of this controller this chip is required.

7 Clock chip

The real time clock for the Arm based controllers is on board the CPU directly,it can store time even during prolonged power outages. In the case of the Raspberry Pi version of the product, since the Pi compute does not have an on board clock this IC will handle the time keeping chores. If you're working with the Pi version this chip should be populated.

8 LCD display and keypad

The main Arm CPU communicates to the LCD over a total of six GPIO lines and uses an additional five GPIO lines for the keypad. The ribbon cable can be extended up to a meter or so in order to mount the keypad outside the enclosure. The display is a 130x 80 dot matrix display with backlight. There are plans to support larger displays and touch screens on the roadmap. Here is a partial schematic of the keypad and display header connections in case you care to make your own cables.
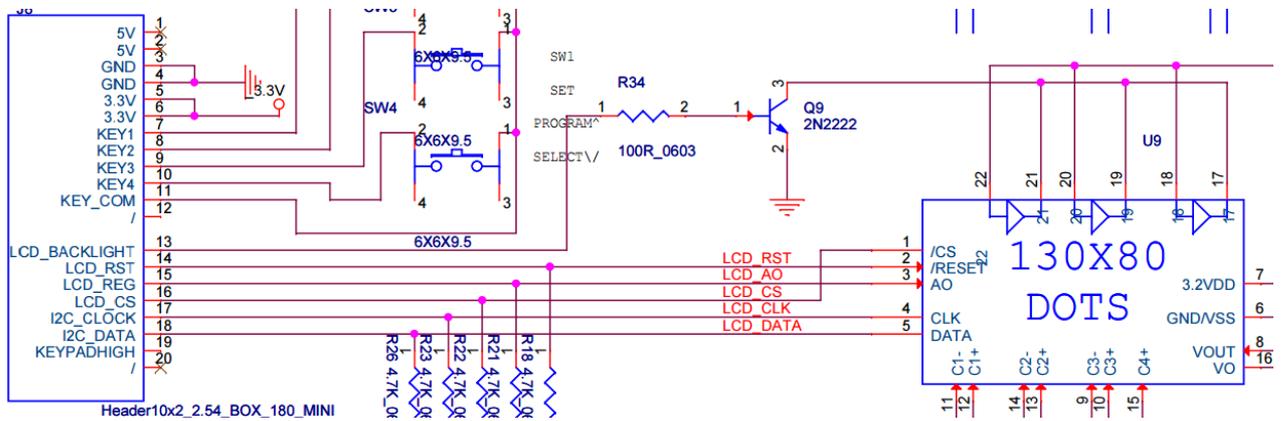
Wireless Options Port

Option Module
Zigbee
GSM/CDMA
Wifi/Bluetooth

The wireless options port is a header accommodates various wireless option modules. The socket can be fitted with a GSM/CDMA cellular modem allowing connections where there is no internet available. There is also a zigbee 900hz option module which allows connecting to the growing collection of zigbee equipment on the market.

The connection from the main CPU to this socket is accomplished with several GPIO, USB, UART and SDIO which means this socket will support present and upcoming standards as they come onto the market. Work is under way on a bluetooth module supporting the new BT5 spec which doubles the speed or regular bluetooth, and increases the range by four.
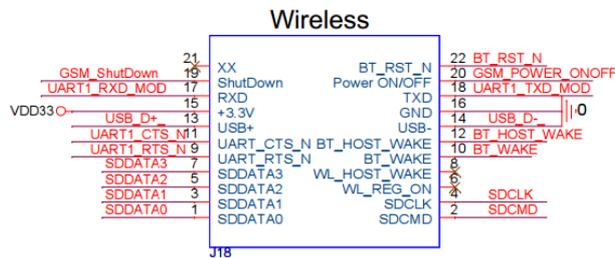
Storage

512k   Ram

4G   Flash

Sdcard

Ram: The CPU module has an external 512k ram chip on board which supports the main application.

SD Card: This slot accommodates an SD card up to 32G in size, its used for storing trend logs and graphics templates for the building. The slot is accessible without taking the lid off the controller, the slot is located along the edge of the lower PCB, opposite the ethernet connector on early models and near the ethernet connector on later models.

*Flash Chip: This chip performs the same function as the SD card slot, only its on board the CPU module directly so the SD card is not required. This flash chip was added on production starting in

Q1 or 2017 so your model may not have the chip in which case you'll need to insert an flash card in the SD slot. You can tell if have this chip by looking at the device info details in the T3000 front end.

Communications Ports

The list of communications ports for the various models of controllers is shown below.

Ports	BB	LB	TB
RS485	2	2	2
RS232	1	0	0
Ethernet	1	1	1
USB Port	1	1	0
*Option Port	1	1	0

*The BB and LB models have an option port which can be populated with either a zigbee or GSM 3G modem.

·   RS485 Communication Speed is adjustable for Modbus master or slave and Bacnet MSTP at a baud rates from 9600 to 115k baud. For longer cable runs the practical speed limitation is 38.4kbaud, you can experiment with faster baud rates if the cabling is well laid out with shielding and earthing at all the nodes.
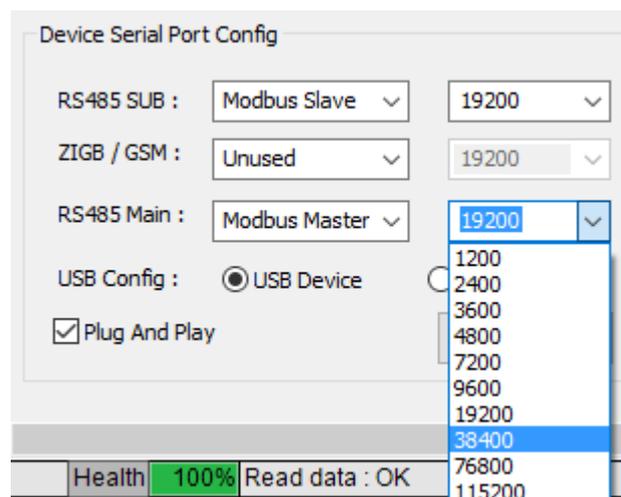
Ethernet speed is 100/10 MBaud

RS232 Com port on the BB model can be used as a debugging terminal and for integrating. Baud rate us adjustable to 9600, 19.2k, 38.4k

.

The USB Port is normally available as a COM port for debugging purposes. If a GSM modem is installed, the modem will occupy this port and it will not be available for your application or debugging.

Micro Brother 'MB' model

8 Universal Inputs

4 Relay Outputs

2 Analog Outs

2 RS485 Ports

Ethernet

Little Brother 'LB' model

16 Universal Inputs

6 Relay Outputs

4 Analog Outs

2 RS485 Ports

Ethernet

USB Port

Optional Zigbee/GSM/CDMA

Big Brother 'BB' model

32 Universal Inputs

RS232 Port

12 Relay Outputs

12 Analog Outs

2 RS485 Ports

Ethernet

USB Port
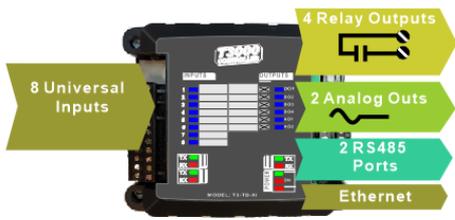
Optional Zigbee/GSM/CDMA

Wireless

Wireless Connectivity:

The system supports an optional zigbee add-on module which enables connecting to wireless thermostats and other zigbee devices. The zigbee devices form a mesh which makes the network more robust than a regular point to point type network. If a particular node experiences interference or goes offline the network will heal itself and work around the problem node.

The system also supports a GSM module for remote connectivity where there is no internet connection available. Most mobile carriers can be accommodated with either the GSM modems over 3G networks. There is no CDMA option as of this writing, Feb 2017. You will need to locally source a prepaid sim card which is then inserted into the wireless module and allows connecting to remote sites such as boats, tractor trailers, remote cabins and so on. The only charges are the local carrier fees which are available for a flat rate per gig of data with no monthly fees, as arranged with the local carriers yourself.

Wifi and bluetooth connections are in the product pipeline, expected to be in production in mid 2017.

TBD: Show zigbee, wifi, bluetooth and GSM modules here.

Tstat6-Zigbee

Expansion Modules

The T3 controllers can be expanded by adding input & output modules on the RS485 expansion bus. Typically the expansion I/O are located within the same cabinet as the controller but can be located remotely up to several hundreds of meters away if the application calls for it.

The maximum number of I/O points for each T3 controller is currently limited to 64 inputs and 64 outputs. You can add as many T3 controllers as you like to the system so very large systems can be built up. A typical connection arrangement is shown here with the PC connected to the T3 controller over ethernet and the expansion I/O connected to the T3 over RS485 on the RS485 subnet. In this arrangement the T3 controller treats the expansion points transparently like internal points directly on the controller itself.

Below shows the expansion inputs as seen from the T3000 front end starting at input 17 as shown by the word 'External' at Tab1. Expansion input 17 resides on a t3-22i expansion module as shown at tab2. From a programming perspective, these expansion i/o appear just like inputs and outputs of the T3 controller hardware itself. When working with these expansion points the operator only need refer to 10IN17 for example. There is no need to deal with the fact that this input is physically connected to a separate expansion module which happens to be at address ID=70 in this case, you can see the subnet ID at Tab3.

Currently, only Modbus protocol is available for the expansion points subnetwork, however all info on the Ethernet network is available in both Modbus and Bacnet protocols by defaut. The fact these are expansion points is taken care of automatically, you can work with 'bacnet object INPUT17' when using the ethernet communications.

Input	Panel	Full Label	Auto/Manual	Value	Units	Range	Calibration	Sign	Filter	Status	Jumper	Label	External	Product N
IN1	10	TANK TEMP	Manual	110.00	Deg.F	10K -40 to 250	0.0	-	0	Normal	Thermistor Dry...	TANKTMP		
IN2	10		Auto	-40.00	Deg.F	10K -40 to 250	0.0	-	0	Open	Thermistor Dry...	APT2		
IN3	10		Auto	-40.00	Deg.F	10K -40 to 250	0.0	-	0	Open	Thermistor Dry...	APT3		
IN4	10		Manual	20.00	Deg.F	10K -40 to 250	0.0	-	0	Normal	Thermistor Dry...	APT4		
IN5	10		Manual	20.00	Deg.F	10K -40 to 250	0.0	-	0	Normal	Thermistor Dry...	APT5		
IN6	10		Manual	20.00	Deg.F	10K -40 to 250	0.0	-	0	Normal	Thermistor Dry...	APT6		
IN7	10		Manual	20.00	Deg.F	10K -40 to 250	0.0	-	0	Normal				
IN9	10		Manual	75.00	Deg.F	10K -40 to 250	0.0	-	0	Normal	Thermistor Dry...	APT9		
IN10	10		Manual	80.00	Deg.F	10K -40 to 250	0.0	-	0	Normal	Thermistor Dry...	APT10		
IN11	10		Manual	20.00	Deg.F	10K -40 to 250	0.0	-	0	Normal	Thermistor Dry...	APT11		
IN12	10		Manual	20.00	Deg.F	10K -40 to 250	0.0	-	0	Normal	Thermistor Dry...	APT12		
IN13	10		Manual	20.00	Deg.F	10K -40 to 250	0.0	-	0	Normal	Thermistor Dry...	APT13		
IN14	10		Manual	20.00	Deg.F	10K -40 to 250	0.0	-	0	Normal	Thermistor Dry...	APT14		
IN15	10		Manual	20.00	Deg.F	10K -40 to 250	0.0	-	0	Normal	Thermistor Dry...	APT15		
IN16	10		Manual	20.00	Deg.F	10K -40 to 250	0.0	-	0	Normal	Thermistor Dry...	APT16		
IN17	10-70		Auto	2.96		Unused		-	1	Normal	Thermistor Dry...	AI0	External	T3-22I
IN18	10-70		Auto	2.96		Unused		-	1	Normal	Thermistor Dry...	AI1	External	T3-22I
IN19	10-70		Auto	2.96		Unused		-	1	Normal	Thermistor Dry...	AI2	External	T3-22I
IN20	10-70		Auto	2.97		Unused		-	1	Normal	Thermistor Dry...	AI3	External	T3-22I
IN21	10-70		Auto	2.96		Unused		-	1	Normal	Thermistor Dry...	AI4	External	T3-22I
IN22	10-70		Auto	2.96		Unused		-	1	Normal	Thermistor Dry...	AI5	External	T3-22I
IN23	10-70		Auto	2.96		Unused		-	1	Normal	Thermistor Dry...	AI6	External	T3-22I
IN24	10-70		Auto	2.97		Unused		-	1	Normal	Thermistor Dry...	AI7	External	T3-22I
IN25	10-70		Auto	2.84		Unused		-	1	Normal	Thermistor Dry...	AI8	External	T3-22I

The following sections show some of the expansion i/o modules which are integrated into the T3000 system. Its not difficult to integrate other devices to be mapped as external I/O and show up within the T3000 i/o map, so if you have a module or product that you'd like to see here let us know.

1 <TODO : Control Name>

External

External shows starting at input 17.

2 <TODO : Control Name>

T3-22I

T3-22i expansion module.

3 <TODO : Control Name>

| IN16 | 10 |
| IN17 | 10-70 |
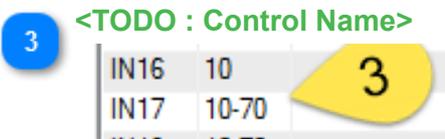
The subnet ID.

T3-8o MODULE

The T3-8o Output Module has eight universal inputs which can be set up for the common sensor signals and pulse counting. The hardware and software setup is the same as for the T3 controllers which are explained in detail here: inputs

There are 6 relay dry contacts with a rating of 5A/24VAC on each of the DO's. Then there are eight analog output signals, they operate form 0-10V and have a drive current of 100ma each. Thanks to this relatively high drive current it is possible to energize a 12VDC relay with the output directly, this allows the system designer to convert any of the analog outputs to a relay output if necessary. The reverse is not true though, a relay output cannot operate in 10V analog mode on this module, though it can be set up to operate in PWM mode which is useful for electric heating applications.

Each output has an LED for a quick visual indication of the output state, in the case of an analog output the LED dims according to the percentage on. Each output also has an LED and a manual override switch for local operation.

All connections including the inputs, outputs, RS485 ports are all well protected from short circuits and accidental connections to 24vac.

The input signals have the following characteristics:

All inputs can be software configured for sensing either analog or digital signals. The analog inputs can have one of four ranges: 0-3V, 0-10V, thermistor/ dry contact, and 4-20ma. Once the jumper is set up you then set up the engineering units and sensor signal type in software from the many built in software ranges. You can also configure your own custom sensors using your own table of voltages versus signal.

Analog Outputs are used for modulating type actuators, variable speed drives, valves and damper motors. They are fixed in hardware to operate over 0-10V and can source a maximum of 100 mA per channel. If your application calls for more binary outputs, any spare analog outputs can be set up as relay outputs by wiring to an external relay, the 100ma drive is enough to energize the smaller 12VDC coil relays. Then in software, the output range is configured as on/off binary mode.

Digital Inputs are wired to dry contacts to detect the open and closed states, there is an internal pull up resistor to provide the excitation voltage for the dry contacts. When the contact is closed,

the controller will read 0V or 'closed' and when the contact is open the controller will see the 3V pullup voltage and show 'open'. You can also select from many other ranges such as ON-OFF , the reverse of that: OFF-ON, or OPEN/CLOSED and so on. You can also set up your own wording such as UP-DOWN.

The Relay Outputs are dry contact relays rated at 2A, 120VAC. Both sides of the relay are wired to the terminal block so you can assign any relay to its own independent circuit. You can also common up the relays to share a common ground or supply using jumpers or a strip wired into the terminal block.

There are

8 Universal Inputs
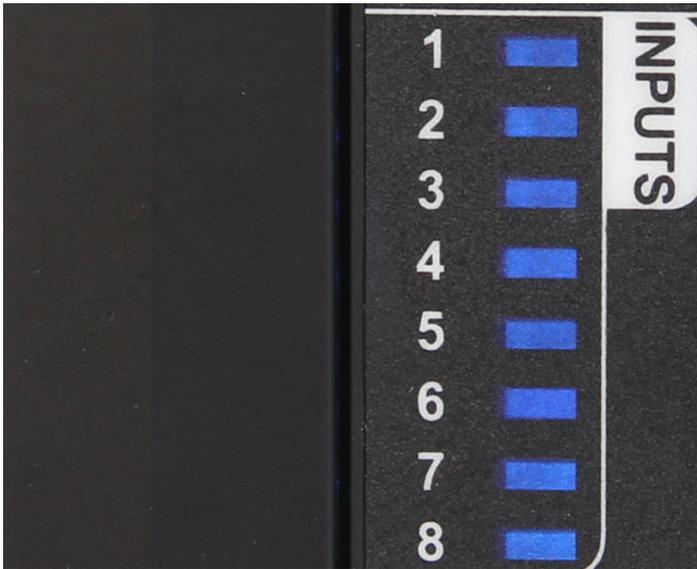
There are eight inputs that can be software configured for sensing either analog or digital signals.Each input has an LED for a quick visual indication of the input state.

6 Relay Outputs
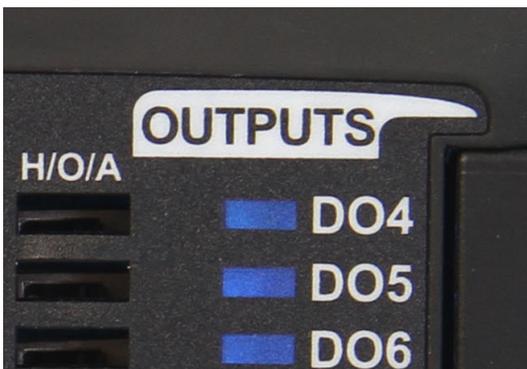
There are six digital outputs,wired to dry contacts to detect the open and closed states.Each output has an LED for a quick visual indication of the output state and a manual override switch for local operation.
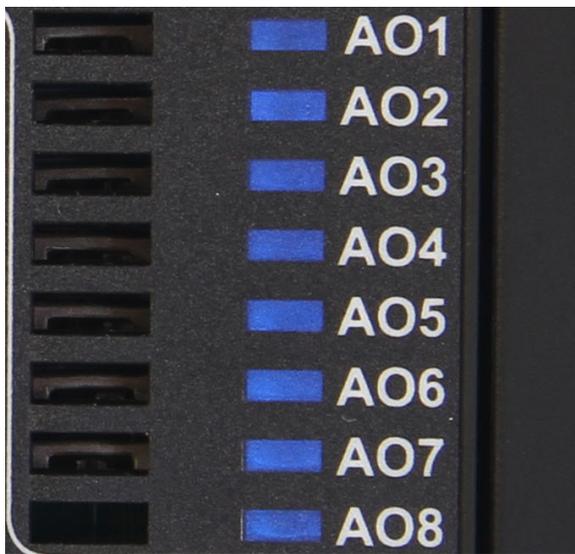
8 Analog Outputs

There are six analog outputs,used for modulating type actuators, variable speed drives, valves and damper motors.Each output has an LED for a quick visual indication of the output state and a manual override switch for local operation.

Power

The fuse LED lights up when there is a ground loop or other wiring mistake in the system. When the problem is fixed the fuse will cool down and normal operation should resume. If this LED is stuck in the on condition with no other sensors and network connected, then there is a hardware problem.

Ethernet

The device supports both Bacnet and Modbus over the 100MB ethernet port. Both protocols can be used simultaneously. The device is not a router however, communications from the RS485 port will not be routed over to the ethernet port. For this type of routing, please refer to the T3-controller series.

RS485 Subnet

RS485 port supports either Bacnet MSTP protocol and Modbus RTU.

Heart Beat LED

The beat led will blink once every second or so during normal operation. In the first 10 seconds after power up, the beat LED will be flashing two times quickly, a pause, then two more blips in a repeating manner. This mode is called the 'isp' mode or in system programming mode, during this time it is possible to load firmware using the ISPTool. Even if there is a bad firmware update you can always get into this ISP mode by cycling power to save a bricked module. If the unit does not start with the two blip isp mode and then switch to the one blip for normal operation then there's a problem with the hardware and/or firmware. Flashing the unit with a fresh copy of the firmware is the best bet, in the first few seconds after power up if necessary.

Normally firmware is updated during the 'normal' mode with the unit in operation and beat led flashing once every second. From the T3000 application you can go to help -> check for updates and follow the firmware update prompts.

T3-22i MODULE

The T3-22i input Module has twenty-two universal inputs which can be set up for the common sensor signals and pulse counting. The hardware and software setup is the same as for the T3 controllers which are explained in detail here: inputs .

All connections including the inputs, RS485 ports are all well protected from short circuits and accidental connections to 24vac.

The input signals have the following characteristics:

All inputs can be software configured for sensing either analog or digital signals. The analog inputs can have one of four ranges: 0-3V, 0-10V, thermistor/ dry contact, and 4-20ma. Once the jumper is set up you then set up the engineering units and sensor signal type in software from the many built in software ranges. You can also configure your own custom sensors using your own table of voltages versus signal.

Digital Inputs are wired to dry contacts to detect the open and closed states, there is an internal pullup resistor to provide the excitation voltage for the dry contacts. When the contact is closed, the controller will read 0V or 'closed' and when the contact is open the controller will see the 3V pullup voltage and show 'open'. You can also select from many other ranges such as ON-OFF , the reverse of that: OFF-ON, or OPEN/CLOSED and so on. You can also set up your own wording such as UP-DOWN.

1 **22 Universal Inputs**
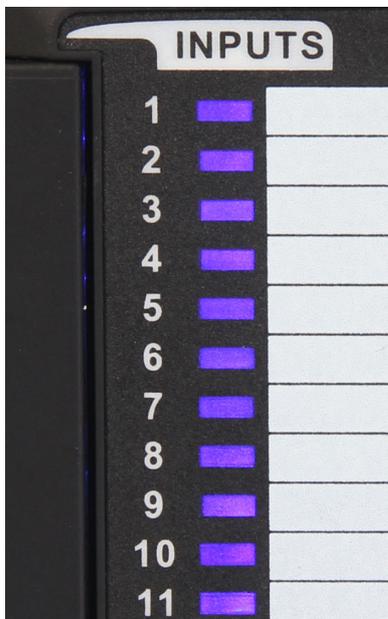
There are 22 inputs that can be software configured for sensing either analog or digital signals.Each input has an LED for a quick visual indication of the input state and writeable area for labelling. For analog ranges the leds will dim according to the voltage on the input terminal. For binary ranges teh LED can be set to be on when a contact is closed or the reverse for on when the contacts are open.

RS485 Port

RS485 port supports either Bacnet MSTP protocol and Modbus RTU at baud rates up to 112k baud.

Power

The fuse LED lights up when there is a ground loop or other wiring mistake in the system. When the problem is fixed the fuse will cool down and normal operation should resume. If this LED is stuck in the on condition with no other sensors and network connected, then there is a hardware problem.

Ethernet Port

The device supports both Bacnet and Modbus over the 100MB ethernet port. Both protocols can be used simultaneously. The device is not a router however, communications from the RS485 port will not be routed over to the ethernet port. For this type of routing, please refer to the T3-controller series.

Heart Beat LED

The beat led will blink once every second or so during normal operation. In the first 10 seconds after power up, the beat LED will be flashing two times quickly, a pause, then two more blips in a repeating manner. This mode is called the 'isp' mode or in system programming mode, during this time it is possible to load firmware using the ISPTool. Even if there is a bad firmware update you can always get into this ISP mode by cycling power to save a bricked module. If the unit does not start with the two blip isp mode and then switch to the one blip for normal operation then there's a

problem with the hardware and/or firmware. Flashing the unit with a fresh copy of the firmware is the best bet, in the first few seconds after power up if necessary.

Normally firmware is updated during the 'normal' mode with the unit in operation and beat led flashing once every second. From the T3000 application you can go to help -> check for updates and follow the firmware update prompts.

T3-PT12

The T3-PT12 is a precision temperature measurement module which can accept up to twelve instrument grade sensors.

The unit accepts platinum Pt elements, either 100 ohm or 1k ohm variety. It also can accept 10k thermistors, Type II and Type III. Cabling can be accomplished using two , three or four wire connections. Communications is supported over Bacnet and Modbus over both RS485 and the Ethernet ports. All readings and the various settings are available as Bacnet objects and Modbus registers.

The unit has12 Analog PT100 or PT1000 Inputs,supports both MSTP and IP connections.

The RS485 port has separate upstream and a downstream connectors to make troubleshooting easier.

Each input as well as the RS485 connections have a separate screw terminal, there's no need to gang two wires under one terminal for any of the terminations.

1 12 Sensor inputs
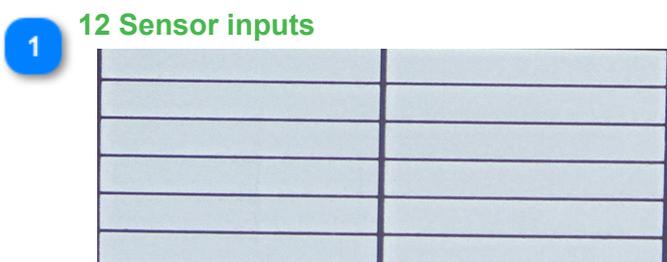
Writeable area for labelling the sensors

2 RS485 Port

RS485 port supports either Bacnet MSTP protocol and Modbus RTU.

3 Power

The fuse LED lights up when there is a ground loop or other wiring mistake in the system. When the problem is fixed the fuse will cool down and normal operation should resume. If this LED is stuck in the on condition with no other sensors and network connected, then there is a hardware problem.

4 Ethernet port

The device supports both Bacnet and Modbus over the 100MB ethernet port. Both protocols can be used simultaneously. The device is not a router however, communications from the RS485 port will not be routed over to the ethernet port. For this type of routing, please refer to the T3-controller series.

5 Heart Beat LED

The beat led will blink once every second or so during normal operation. In the first 10 seconds after power up, the beat LED will be flashing two times quickly, a pause, then two more blips in a repeating manner. This mode is called the 'isp' mode or in system programming mode, during this time it is possible to load firmware using the ISPTool. Even if there is a bad firmware update you can always get into this ISP mode by cycling power to save a bricked module. If the unit does not start with the two blip isp mode and then switch to the one blip for normal operation then there's a problem with the hardware and/or firmware. Flashing the unit with a fresh copy of the firmware is the best bet, in the first few seconds after power up if necessary.

Normally firmware is updated during the 'normal' mode with the unit in operation and beat led flashing once every second. From the T3000 application you can go to help -> check for updates and follow the firmware update prompts.

PWM

PWM

The PWM is a low cost device that is normally used to convert a pulse into a voltage signal. An RTU or other simple controller with limited resources can pulse one of its outputs to generate modulating analog voltage outputs using this module. To generate a 5V signal, the RTU will turn generate a pulse of varying duty cycle, 5 seconds on followed by up to a minute off will generate a 5V analog outut. Precision of the output voltage is up to 0.1 volts.

The inputs can also be set up as general purpose 'universal inputs' for sensing of various sensors. The inputs can be configured as dry contacts for use with switches, occupancy sensors, contactors, current sensors and popular thermistor curves, 10k type 2 and 3 and most of the typical ranges of the rest of the T3 modules.

The main difference of this low cost model to the other T3 series expansion units are:

   -For measuring of 4-20ma signals, these require the addition of an external 250 ohm resistor whereas the rest of the T3 series has this on board

   -This unit has physical jumpers for selecting the signal type whereas the other T3 modules select the sensor type in software.

   -No enclosure, just a cabinet mount type backplate for mounting.
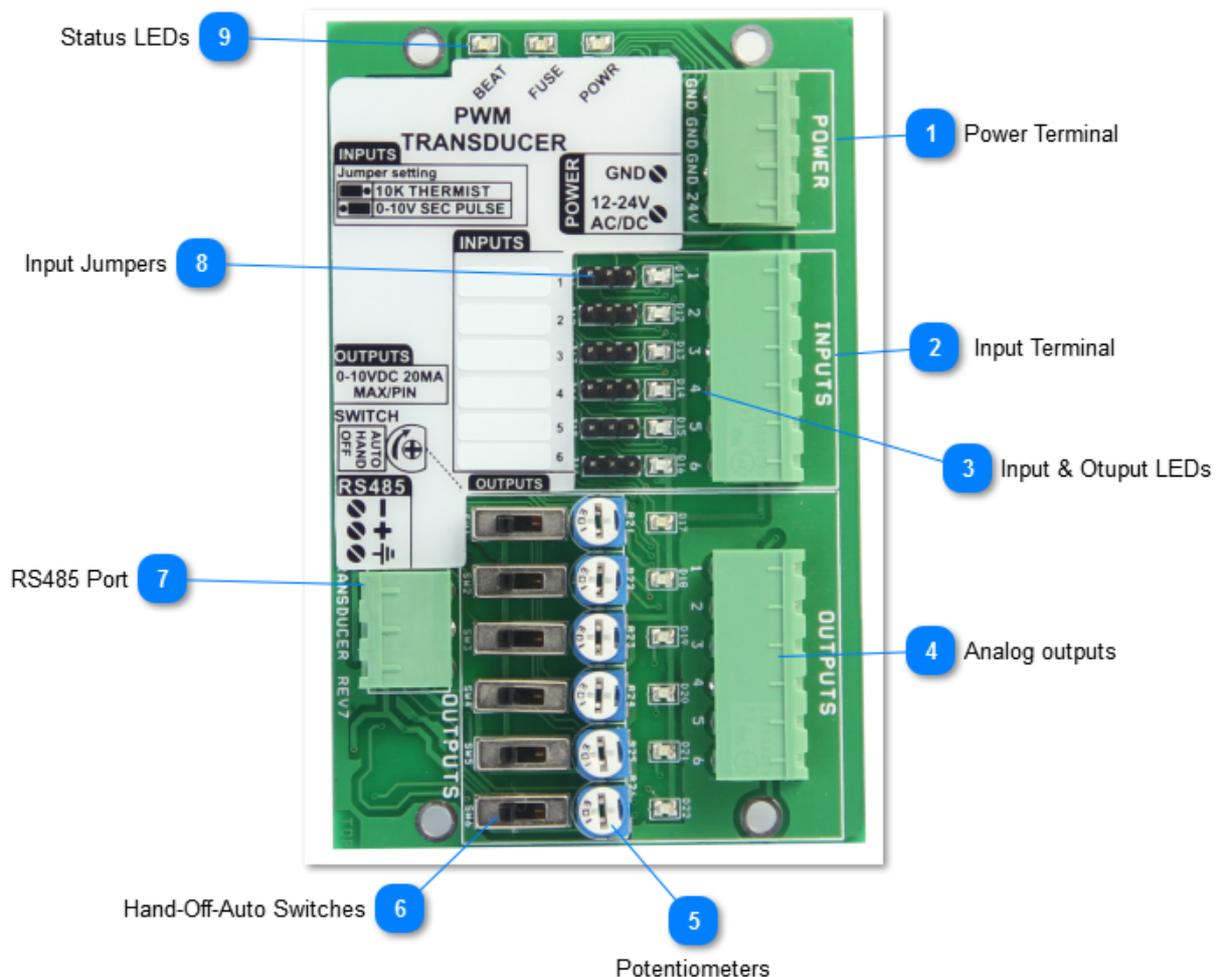
This device can communicate over RS485 using either bacnet or modbus protocols at baud rates up to 112k.

It has a H-O-A (hand-off-auto) override switch for each of the six outputs, as well as a potentiometer for adjusting the out voltage when the output is in manual mode. When the switch is in the hand position the corresponding output can be adjusted by potentiometer from zero to the full scale 10V.

When the switch is in the off positon the output voltage is 0V. When the switch is in auto the output is set by the duty cycle of the corresponding input.

For each input there is one output, six of these input-output pairs on the board. The input signals can be 5VDC, 10VDC, and 24VAC. No jumper is required, the unit will auto detect the signal type. When using AC signals, the frequency can be anywhere in the 50 to 60Hz range. The output voltage is proportional to the duration of the pulse, with the relationship being one volt for each second duration on the pulse. FOr example, a 4 second pulse will produce an output of 4 volts, five seconds for a five volt output and so on up to the maximum of 10 seconds for 10V. The input pulse must be refreshed every 60 seconds otherwise output will fail safe to 0V and input LED for the channel will flash 4 times per second as a fault indicator. If the PWM signal is less than 1 second, output is set to 0V. It is therefore not possible to set an output voltage between zero and 1V.

Status Indicators Red LED - variable intensity to indicate output voltage, dim light indicates 1V and bright light indicates10V. Each input also has an LED which shows the status of the PWM signal or in the case of using the PWM as a general purpose i/o module, the state of the input on or off, and so on.
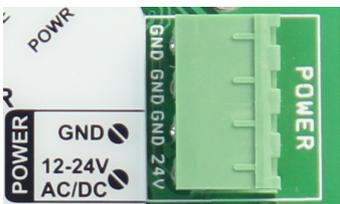
Power Terminal

The power connection has 24C ac or dc supply. There are three ground terminals to provide space for grounding from various devices and power supplies, they are connected in parallel internally.

Input Terminal
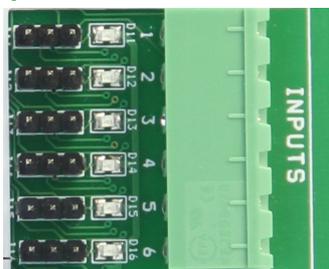
There are six inputs,each of which has a corresponding LED and jumper. No jumper is actually required for most PWM type applications, the unit will detect the signal type automatically. The input pulse signal must be refreshed every 60 seconds otherwise output will fail safe to 0V. When this happens the corresponding LED will flash 4 times per second as a fault indicator. If the PWM signal on-time falls to less than 1 second the output will go to 0V.
When the unit is used as general purpose I/O for the T3 controller, you can select the sensor type with this jumper.

Input & Otuput LEDs

The input LEDS will show the incoming pulse signal on each of the inputs. If the input signal does not see a pulse for 60 seconds it will flash four times in a repeating cycle till a pulse comes in as a warning of the lost signal.

If the module is used as general purpose i/o, the input will reflect the voltage on the input terminal. For example if the input is configured as a binary on-off input the LED will be ON when the contacts are closed and OFF when the contacts are closed. The sense of the LED can be reversed as well, configured to be OFF when the contacts are open and vice versa for closed. For analog inputs such as thermistors and transmitters the LED will be bright or dim depending on the signal. The dimming is done in only about ten steps of brightness so serves mainly to help spot open or short circuit sensors.

The Output LEDS will modulate from dim to bright according to the voltage on the output terminal.

Analog outputs
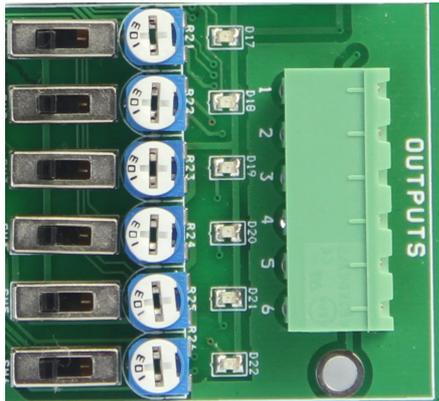
There are six output channels,each of the outputs has a potentiometer and hand-off-auto swtich. When the switch is in the off position the output is at zero volts. In hand the output is at full scale, normally 10V but it can be adjusted down with this pot to any custom setting. In the auto position the output modulates from zero to 10V as per the timing of the pulse on the associated input.

Potentiometers

These pots set the maximum voltage for each of the outputs when in manual mode.

Hand-Off-Auto Switches
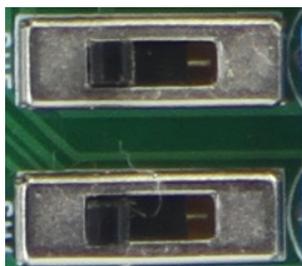
These switches allow the operator to turn the outputs on manually, overriding the software settings for testing and debugging in the field. When the switch is set to the manual on position, the output will go to full scale, as determined by the potentiometer.

RS485 Port

RS485 serial communications port, Modbus and bacnet protocols at up to 112k baud.

Input Jumpers
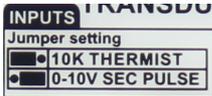
When the PWM is used in normal PWM to voltage applications there is no need to use jumpers here, the default unit is shipped with no jumpers installed. The PWM can auto detect the the type of signal whether its a 0-5V pulse, 10V pulse, or 24VAC pulse.

When using the PWM as general purpose expansion I/O for a T3 (or other) controller you can select the signal type by setting the jumper here and the corresponding 'range' setting from the T3000 software. For 10k thermisters and binary dry contact signals the jumper will be set to the left. For 0-10V and 0-5V signals, the jumper will be set to the right. Once the jumper is set you can refer the the Inputs section for selecting the appropriate range.

Note that most T3 modules do not have physical jumpers these days, the 'jumper' is set in firmware, the PWM is an exception and requires manually moving a jumper.

Status LEDs

The beat LED will blink once every second during normal operation. In the first 10 seconds after power up the beat LED will be flashing two times quickly, a pause, then two more blips in a repeating manner. This mode is called the 'isp' mode or in system programming mode, during this time it is possible to load firmware using the ISPTool. Even if there is a bad firmware update you can always get into this ISP mode by cycling power in order to revive a bricked module. If the unit does not start with the two blip isp mode and then switch to the one blip for normal operation after approximately ten seconds this indicates there's a problem with the hardware and/or firmware. Re-flashing the unit with a fresh copy of the firmware is the best bet, in the first few seconds after power up if necessary.

The fuse LED lights up when there is a ground loop or other wiring mistake in the system. When the problem is fixed the fuse will cool down and normal operation should resume. If this LED is stuck in the on condition with no other sensors and network connected this indicates a hardware problem and hte unit should be returned for repairs which are usually free unless the board has been seriously damaged by water or high power.

The power LED lights up when the power is connected.

Installing and connecting

Installing T3000 Software

The following chapter discusses how to run the T3000 software and connect to the controller for the first time.

Download and install the latest T3000 software from the web site: [T3000 Software](#)
A zip file will be downloaded to your local download directory where you can unzip it and store it for future installs. You can ignore the security warnings while unzipping and running the application.

Two programs will be installed, T3000 is the main application used for all setup and also later on for the graphical front end of the system. The other program is called ISPTool which is short for 'In system Programming" tool. It is a simple stand alone utility for updating firmware in the controllers and can be called from withint the T3000 main software or it can be run directly as the stand alone application.

Connect the PC and the controller through a hub as shown here, or if you dont have a hub you can connect your PC directly to the controller, you can use either a straight thru cable or crossover cable. The controllers and T3000 application can coexist smoothly on the typical small office or home lan so you are free to plug the controller into the office/home lan hub. The default setting is a static IP address at 192.168.0.3. The T3000 application can discover the device no matter what your local network setup is and will give you a dialog with a suggested IP setting for the controller.

Power is connected on the white cable as shown, any supply from 12V to 24V and either AC or DC will be fine such as a spare wall transformer or normally in the HVAC industry a 24vac transformer is used. The controller is protected from reverse polarity and over voltages up to 40V. Power consumption depends on the model but is typically less than 2 watts. During the inital setup there there should be no other sensors or controllers connected yet, just the network cable and the PC.

After power up, the T3 controller will go through a bootup sequence for approximately fifteen seconds where the heartbeat LED is flashing with two quick pulses every second or so which is the inital bootup mode, and after that it will go to 'run mode' where the heartbeat LED is flashing once every second. Start the T3000 application and click on the magnifying glass to scan the network and discover any connected devices.
  See the section on troubleshooting if the device doesn't power up and the LEDs flash as described here.
Troubleshooting Ethernet Connections

As devices are discovered they are added to a tree on the left. All items are saved in a database on the hard disk so you don't need to re-scan each session. The status of the connection to all the devices is updated from time to time in the background, any devices which are in the database but offline will show with an 'x' in the tree.

Clicking on a device in the left hand tree will show the details in the main work area, some devices have only one detail screen while others have several screens. Here we're looking at the details for the controller called 'Chamber' which is a T3-BB controller. Under the menu system are the icons which link to the detail screens for the "Chamber' controller such as inputs, outputs and so on. Other nodes such as the thermostats will have only one or two detail screens associated with them.

Input	Full Label	Auto/Manual	Value	Units	Range	Calibration	Sign	Filter	Status	Jumper	Label
IN1	CHAMBER TEMP1	Auto	16.40	Deg.C	10K -40 to 120	0.0	+	20	Normal	Thermistor Dry...	TEMP1
IN2	CHAMBER TEMP2	Auto	16.30	Deg.C	10K -40 to 120	0.0	-	20	Normal	Thermistor Dry...	TEMP2
IN3	ROOM TEMP	Auto	16.60	Deg.C	10K -40 to 120	0.0	+	20	Normal	Thermistor Dry...	ROOMTE
IN4	FAN STATUS	Auto	On		Off/On		-	0	Normal	Thermistor Dry...	FANSTAT
IN5		Auto	-40.00	Deg.C	10K -40 to 120	0.0	+	0	Open		Unused
IN6		Auto	-40.00	Deg.C	10K -40 to 120	0.0	-	0	Open		Unused
IN7		Auto	-40.00	Deg.C	10K -40 to 120	0.0	-	0	Open	Thermistor Dry...	
IN8		Auto	120.00	Deg.C	10K -40 to 120	0.0	-	0	Short		Unused
IN9		Auto	-40.00	Deg.C	10K -40 to 120	0.0	-	0	Open		Unused
IN10		Auto	-40.00	Deg.C	10K -40 to 120	0.0	-	0	Open		Unused
IN11		Auto	-40.00	Deg.C	10K -40 to 120	0.0	-	0	Open		Unused
IN12		Auto	-40.00	Deg.C	10K -40 to 120	0.0	-	0	Open		Unused
IN13		Auto	-40.00	Deg.C	10K -40 to 120	0.0	-	0	Open		Unused
IN14		Auto	-40.00	Deg.C	10K -40 to 120	0.0	-	0	Open	4-20 ma	
IN15		Auto	-40.00	Deg.C	10K -40 to 120	0.0	-	0	Open		Unused
IN16		Auto	-40.00	Deg.C	10K -40 to 120	0.0	-	0	Open		Unused
IN17		Auto	296.00		Unused	0.0	-	0	Normal		Unused
IN18		Auto	759.00		Unused	0.0	-	0	Normal		Unused
IN19		Auto	294.00		Unused	0.0	-	0	Normal		Unused
IN20		Auto	1023.00		Unused	0.0	-	0	Normal		Unused
IN21		Auto	295.00		Unused	0.0	-	0	Normal		Unused
IN22		Auto	1021.00		Unused	0.0	-	0	Normal		Unused
IN23		Auto	296.00		Unused	0.0	-	0	Normal		Unused

Tree: Building1 — Local Network — CHAMBER — 2883884 — FANCE_TEST_BOARD — Panel_113 — Panel14 — Panel254 — t8123 — t8123 — Power Meter — PowerMeter — T3-22I:100-254-192.168.0.163 — TINY — T3-8AI8AO6DO:90867-10 — t8 — TSTAT6-3

Add 20 | Tx 211 | Rx 210 | Err 1 | Health 99% | Write Input List Item4 .Changed to "Off/On" Success!

A new controller will have no user names and passwords so there is no sign on screen at this point. See the section on users and passwords for setting up passwords. If a user name and password has already been entered on the controller you will need to log in with that.

1 **PC or Laptop**

<TODO>: Insert description text here...

2 **Hub**

<TODO>: Insert description text here...

3 **T3 Controller**

<TODO>: Insert description text here...

Hardware Details

1. LCD Display

The IP address will show on the display but cannot be changed from the keypad for the time being. The data and time show here along with the number of packets sent over the ethernet connection. If devices are connected to either of the RS485 ports you will see the device count and how many are currently online.

2. Heartbeat LED

Normally this LED is beating every second to show normal operation. If the LED is off or on all the time, the unit is locked up or had a bad firmware update. In this case you can cycle power to get back into firmware update mode. For approximately ten seconds after power up the unit will be in flash update mode looking for new firmware, the heartbeat LED will be flashing with two short pulses and then a pause with the LED off, repeating every second to show during this time. See the section on updating firmware to revive the controller.

3. Power to Controller

The power supply can be any 12V to 24V supply, AC or DC with capacity of approximately 2 watts. Surge and reverse polarity protected with slow blow self resetting fuse on board.

Caution: Voltages over 40V can cause damage to the controller which requires a trip to the factory for repairs.

4. Ethernet Connection to the T3 controller

Connect the Ethernet port to the hub as shown. The RX and TX LEDS will blink to show communications activity during normal operation. If the T3000 application is operating and there is no activity on these LEDS that would indicate the IP address is not set correctly or there is a bad cable.

5 **Ethernet Hub**

Ethernet hub. You can connect through the home / office lan or through a separate hub as shown. T3000 communications will not interfere with normal lan traffic.

6 **Ethernet connection to the PC.**

You can connect directly to the T3 controller without a hub using either a crossover cable or the normal straight through ethernet cable. Auto MDIX whcih is available on most PCs and laptops these days will sort out which type of cable is in use and communications will flow normally.

3.2   SETTING- UP THE T3000 SYSTEM

This section explains the general setup of a system once you are connected and logged in.

3.2.1   SOFTWARE SET-UP

There are several parameters which the user will enter throughout use of the system. Some of the information such as the file path to various items, the last connection used, the last IP address entered among other parameters are saved to the hard disk in order to reduce keystrokes for the operator. Some of these settings are saved in a filed named CONFIG.INI which is saved to the T3000 installation directory.
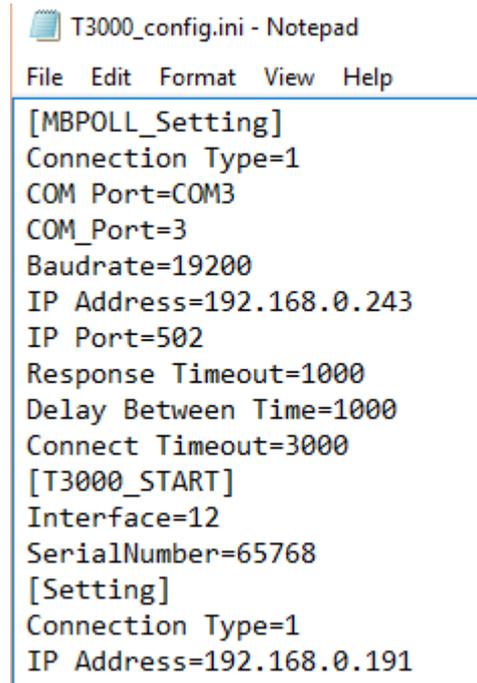
System List

The list of buildings you will connect to is stored in the 'Building Database' which is found under the 'Database' menu item.

This option is used to make calls to remote systems via the modem. It can be set up with the names and numbers of 24 remote systems. Enter the name, and modem phone number. Press **Ins** any where along the line to dial the remote site (the COM ports must be configured first: Chapter 8).
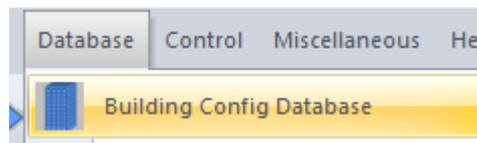
	Building	Protocol	IP / Domain / Tel#	IP Port	COM Port	Baud Rate		Build
Selected	Building1	Auto	N/A	N/A	N/A	N/A	Database\Buildings\Building1\Building1.mdb	
	Building2	Remote Device	Building2.com	10000	N/A	N/A	Database\Buildings\Building2\Building2.mdb	
	Chamber2	Auto	N/A	N/A	N/A	N/A	Database\Buildings\Chamber2\Chamber2.mdb	

FILE PATHS

The default installation folder for T3000 is \Program Files (86)\T3000. Under the main T3000 installation folder is a \Database\ subfolder and below that is a directory for each project. The graphical floorplans and system graphics are stored in their own separate subfolder for each project called 'image'.
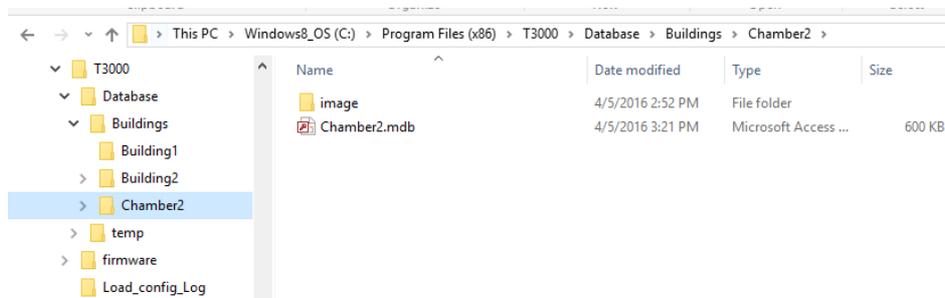
This database is a standard Microsoft Database file format which can be opened and edited in T3000 as well as other mainstream programs such as Excell, Acess, SQL, and Openoffice. The MDB format is a legacy file format originally designed for access from a single PC so work is under way to migrate this file format to an SQL format for easier multiuser sharing.

TIME AND DATE
TBD:
Time and Date is found in the Configuration work menu.

There is the option to let the time sync from an internet time server using the NTP or network time protocol, in this way the controller will never be out of sync even with daylight savings time or after prolonged power outages. Alternatively, if the controller is blocked from communicating out onto the internet it can sync locally with the PC every time someone is online with T3000. And finally, you can let the onboard clock manage the time locally with no sync to any PC or server, the local clock is accurate to a few seconds per day and has a battery backup for power outages which will keep the clock alive for two or three days.

USER LIST

User List is found in the Configuration work menu.

User List provides the means for system security and individual access to the system. Each user has to Sign- On with his name and password to gain access to the system. There is space to allow for 32 different users and pass words plus their access levels. Access to different parts of the system is determined by system supervisor.

3.2.2   HARDWARE SETUP

The options needed to configure the T3000 hardware are all found in the CONFIGURATION pull-down work menu. Once you are done altering the hardware setup the system will ask if you wish to save the changes that you have made.

For more information on these hard ware options see the relevant sections in Chapter 8 - Configuration.

COMMUNICATION

This option allows you to set up the serial port, modem, mouse and printer in your system so all of its components can talk to each other.

MODEM CONFIGURATION

If the modem does not initialize properly when you first Sign- On then you will have to alter the default initialization string. The "Initializing String" is the command string that is sent to the modem in order to bring it to a proper working order. A default string is loaded that the system will first try to initialize the modem with. If this fails the operator will have to edit this string with the proper commands for the modem. Consult the Owner's Manual for the modem you are installing for further information.

BOARDCONFIGURATION

Board Configuration is used to set up the T3000 Controller, Input and Output cards connected to a panel.

1 **Building LIst**

Building
Building1
Building2
Chamber2

<TODO>: Insert description text here...

2 Protocol

Protocol
Auto
Remote Device
Auto

3 Ip

IP / Domain / Tel#
N/A
Building2.com
N/A

<TODO>: Insert description text here...

4 IP port

IP Port
N/A
10000
N/A

<TODO>: Insert description text here...

5 COM Port

COM Port
N/A
N/A
N/A

<TODO>: Insert description text here...

6 Baud Rate

Baud Rate
N/A
N/A
N/A

<TODO>: Insert description text here...

Troubleshooting Ethernet Connections

If you're having troubles connecting you can isolate the PC and device from the main network with a small separate hub. Scan the network to discover the devices on your network by clicking the magnifying glass 🔍 on the menu bar. The default IP address of the controller is set to a static IP of 192.168.0.3 with DHCP disabled. T3000 can discover the device no matter what subnet your PC and device happen to be on by scanning the network with the magnifying glass tool.

If your network happens to be on a different subnet than 192.168.0.X a message will pop up asking if you want to change the IP address of the device to the same subnet as your PC.

Different Subnets Detected! ✕

The subnet of your PC and the device must match.
Would you like to change the IP address of the device to match the IP address of your PC?

	Current		Proposed
Device IP address:	192.168.0.97	➡	192.168.3.2
PC's IP address:	192.168.3.4		

Done		Cancel
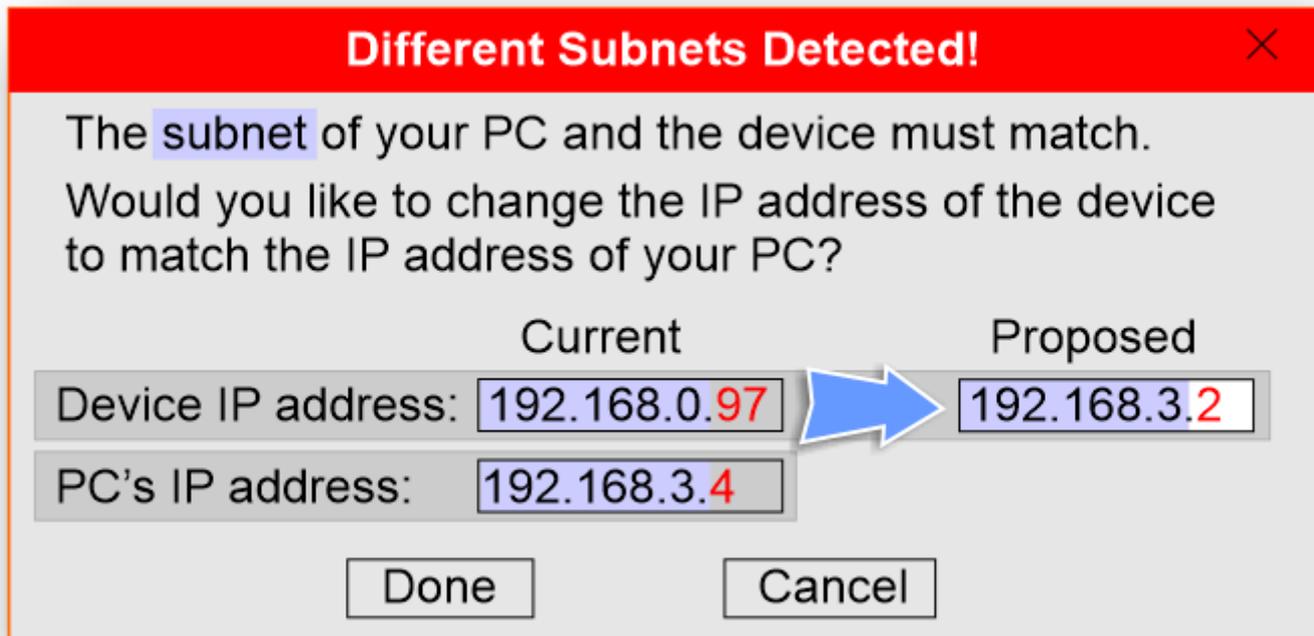
If you're still having problems connecting and don't happen to have a spare hub for isolating from the office/home lan you can try a direct connection as shown here, direct from the PC to the controller with no hub in between. The PC will normally have what is called auto MDI-X capability which means it can detect the direct connection to another host and will automatically flip the connections, no crossover cable is required.
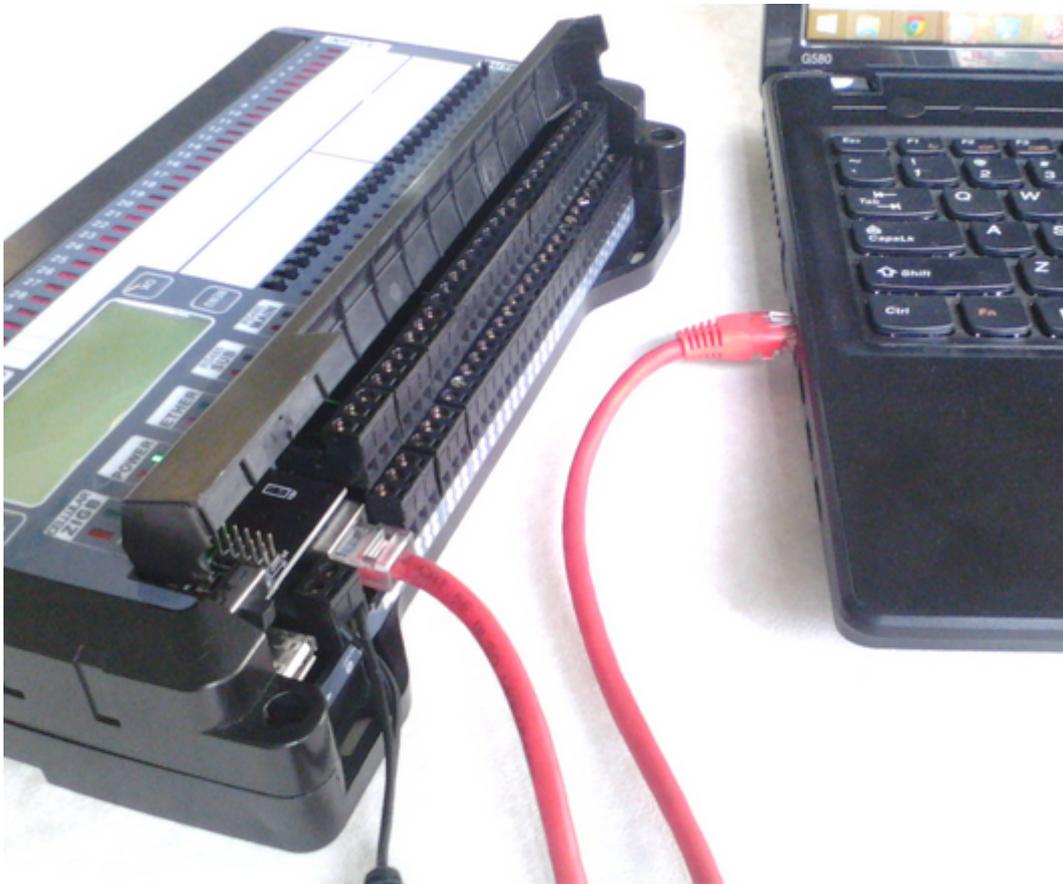
If there are sensors and actuators attached to the controller, try disconnecting them. In any system with many devices and multiple power supplies, it is easy to get 24vac crossed around. Disconnect all the field devices and also the RS485 networks and repeat the scan operation.

Finally, be sure you have the latest T3000 application, it is updated every two weeks, approximately. You can update the software by clicking the check update button.

Installing T3000 Software

The following chapter discusses how to run the T3000 software and connect to the controller for the first time.

Download and install the latest T3000 software from the web site: T3000 Software
A zip file will be downloaded to your local download directory where you can unzip it and store it for future installs. You can ignore the security warnings while unzipping and running the application.

Two programs will be installed, T3000 is the main application used for all setup and also later on for the graphical front end of the system. The other program is called ISPTool which is short for 'In system Programming" tool. It is a simple stand alone utility for updating firmware in the controllers and can be called from withint the T3000 main software or it can be run directly as the stand alone application.

Connect the PC and the controller through a hub as shown here, or if you dont have a hub you can connect your PC directly to the controller, you can use either a straight thru cable or crossover cable. The controllers and T3000 application can coexist smoothly on the typical small office or home lan so you are free to plug the controller into the office/home lan hub. The default setting is a static IP address at 192.168.0.3. The T3000 application can discover the device no matter what your local network setup is and will give you a dialog with a suggested IP setting for the controller.

Power is connected on the white cable as shown, any supply from 12V to 24V and either AC or DC will be fine such as a spare wall transformer or normally in the HVAC industry a 24vac transformer is used. The controller is protected from reverse polarity and over voltages up to 40V. Power consumption depends on the model but is typically less than 2 watts. During the inital setup there there should be no other sensors or controllers connected yet, just the network cable and the PC.

After power up, the T3 controller will go through a bootup sequence for approximately fifteen seconds where the heartbeat LED is flashing with two quick pulses every second or so which is the inital bootup mode, and after that it will go to 'run mode' where the heartbeat LED is flashing once every second. Start the T3000 application and click on the magnifying glass to scan the network and discover any connected devices.
  See the section on troubleshooting if the device doesn't power up and the LEDs flash as described here.
Troubleshooting Ethernet Connections

As devices are discovered they are added to a tree on the left. All items are saved in a database on the hard disk so you don't need to re-scan each session. The status of the connection to all the devices is updated from time to time in the background, any devices which are in the database but offline will show with an 'x' in the tree.

Clicking on a device in the left hand tree will show the details in the main work area, some devices have only one detail screen while others have several screens. Here we're looking at the details for the controller called 'Chamber' which is a T3-BB controller. Under the menu system are the icons which link to the detail screens for the "Chamber' controller such as inputs, outputs and so on. Other nodes such as the thermostats will have only one or two detail screens associated with them.

Input	Full Label	Auto/Manual	Value	Units	Range	Calibration	Sign	Filter	Status	Jumper	Label
IN1	CHAMBER TEMP1	Auto	16.40	Deg.C	10K -40 to 120	0.0	+	20	Normal	Thermistor Dry...	TEMP1
IN2	CHAMBER TEMP2	Auto	16.30	Deg.C	10K -40 to 120	0.0	-	20	Normal	Thermistor Dry...	TEMP2
IN3	ROOM TEMP	Auto	16.60	Deg.C	10K -40 to 120	0.0	+	20	Normal	Thermistor Dry...	ROOMTE
IN4	FAN STATUS	Auto	On		Off/On		-	0	Normal	Thermistor Dry...	FANSTAT
IN5		Auto	-40.00	Deg.C	10K -40 to 120	0.0	+	0	Open		Unused
IN6		Auto	-40.00	Deg.C	10K -40 to 120	0.0	-	0	Open		Unused
IN7		Auto	-40.00	Deg.C	10K -40 to 120	0.0	-	0	Open	Thermistor Dry...	
IN8		Auto	120.00	Deg.C	10K -40 to 120	0.0	-	0	Short		Unused
IN9		Auto	-40.00	Deg.C	10K -40 to 120	0.0	-	0	Open		Unused
IN10		Auto	-40.00	Deg.C	10K -40 to 120	0.0	-	0	Open		Unused
IN11		Auto	-40.00	Deg.C	10K -40 to 120	0.0	-	0	Open		Unused
IN12		Auto	-40.00	Deg.C	10K -40 to 120	0.0	-	0	Open		Unused
IN13		Auto	-40.00	Deg.C	10K -40 to 120	0.0	-	0	Open		Unused
IN14		Auto	-40.00	Deg.C	10K -40 to 120	0.0	-	0	Open	4-20 ma	
IN15		Auto	-40.00	Deg.C	10K -40 to 120	0.0	-	0	Open		Unused
IN16		Auto	-40.00	Deg.C	10K -40 to 120	0.0	-	0	Open		Unused
IN17		Auto	296.00		Unused	0.0	-	0	Normal		Unused
IN18		Auto	759.00		Unused	0.0	-	0	Normal		Unused
IN19		Auto	294.00		Unused	0.0	-	0	Normal		Unused
IN20		Auto	1023.00		Unused	0.0	-	0	Normal		Unused
IN21		Auto	295.00		Unused	0.0	-	0	Normal		Unused
IN22		Auto	1021.00		Unused	0.0	-	0	Normal		Unused
IN23		Auto	296.00		Unused	0.0	-	0	Normal		Unused

A new controller will have no user names and passwords so there is no sign on screen at this point. See the section on users and passwords for setting up passwords. If a user name and password has already been entered on the controller you will need to log in with that.

1 **PC or Laptop**

<TODO>: Insert description text here...

2 **Hub**

<TODO>: Insert description text here...

3 **T3 Controller**

<TODO>: Insert description text here...

Menu Items and Tree

2.Configure your MiniPanel

2.1:Save the date of your Minipanel to a project file.

File	Tool	View

Save As...

Load File

Import...

Exit

.prg   *input the name of the project*
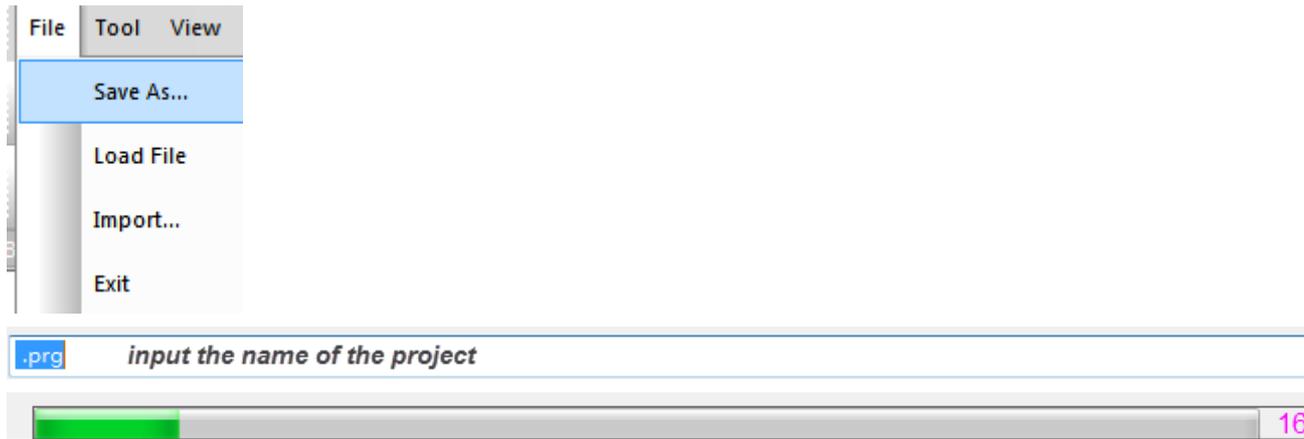
2.2:Load the date of a project file to your Minipanel

Save As...

Load File

Import...

Exit

Prg files (*.prg)
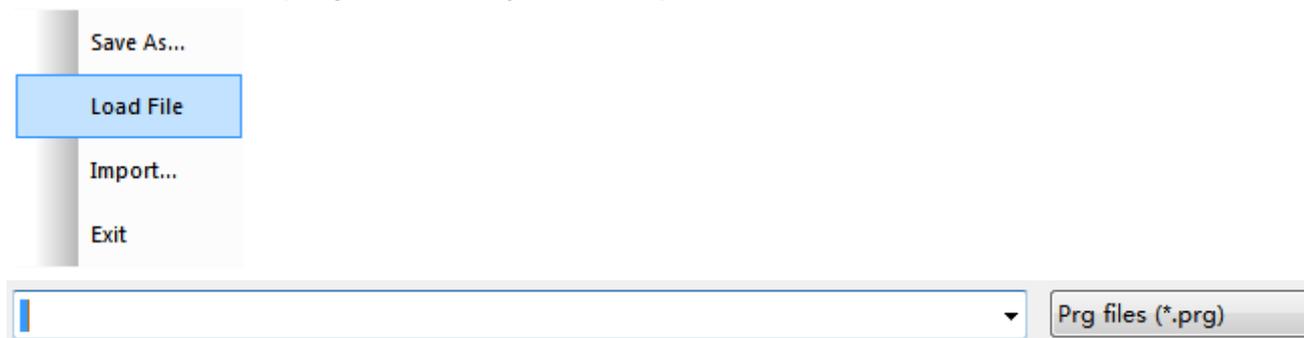
Note: There will be a demo project to show how to do these steps.

System Tree
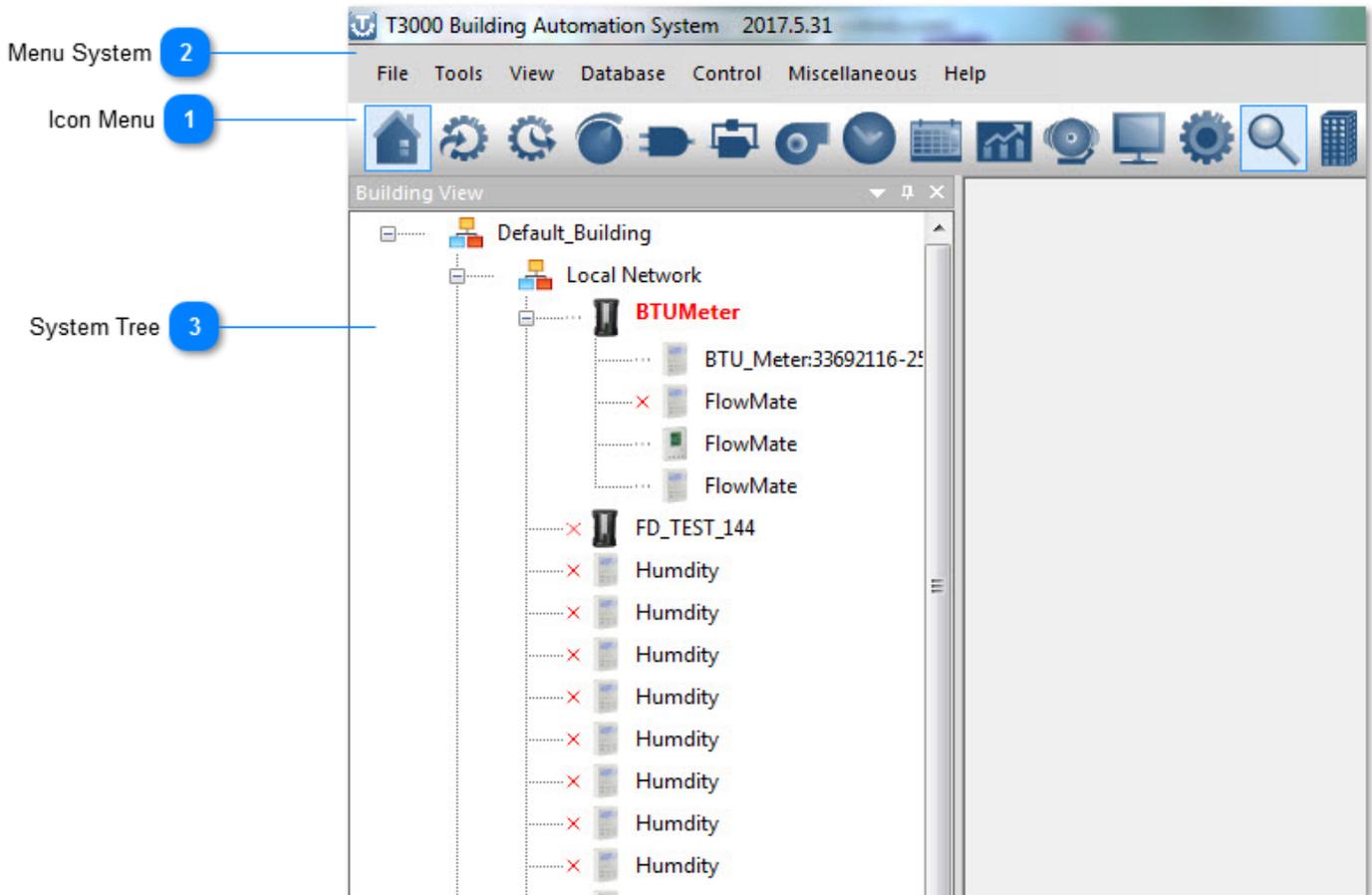
A controller with an RS485 subnetwork will show the subnet devices on a branch below the controller.

Icon Menu

<TODO>: Insert description text here...

Menu System

<TODO>: Insert description text here...

System Tree

- Default_Building
  - Local Network
    - **BTUMeter**
      - BTU_Meter:33692116-2:
      - ✕ FlowMate
      - FlowMate
      - FlowMate
    - ✕ FD_TEST_144
    - ✕ Humdity
    - ✕ Humdity
    - ✕ Humdity
    - ✕ Humdity
    - ✕ Humdity
    - ✕ Humdity
    - ✕ Humdity
    - ✕ Humdity

<TODO>: Insert description text here...

Building Database

‹Building Configuration

notes: When protocol is auto ,IP ,Port ,Com Port ,Baud Rate disable.
          When protocol is Modbus Tcp or Romote Device ,Ip,Port enable ,Comport Baud Rate disable.
          When protocol is Modbus 485 or Bacnet MSTP ,IP,Port disable ,Comport Baud Rate enable.

1 **Building List**

Building
Building 1
Building2
Chamber 2

<TODO>: Insert description text here...

2 Protocol

Protocol

Modbus 485
Remote Device
Auto

<TODO>: Insert description text here...

3 IP/Domain/Tel#

IP / Domain / Tel#

N/A
Building2.com
N/A

<TODO>: Insert description text here...

4 IP Port

IP Port

N/A
10000
N/A

<TODO>: Insert description text here...

5 COM Port

COM Port

N/A
N/A

<TODO>: Insert description text here...

6 Baud Rate

Baud Rate

19200
N/A
N/A

<TODO>: Insert description text here...

T3000 Remote Connection

In order to connect to a T3 controller from a remote site you'll need to follow a few steps. The T3 controllers have a dyndns updater feature so that you can use a dynamic IP address rather than paying for a permanent IP address. There are three free services and also a popular paid service at Dyndns. Access these by connecting to the T3 controller and then clicking on the

advanced gear icon which brings up the dyanamic IP address settings. You'll need to set up an account at one of the listed services and enter the user/password credentials here along with the domain name you chose when signing up for the account. The T3 controller will then connect to the domain service and update the IP address of the controller periodically. Since the IP doesn't change often a setting of ten minutes will be fine.

In your router you'll need to set port forwarding to the IP address of the controller for the following ports:

    UDP Port 47898 is the bacnet port.

    UDP 1234 is the scan port, the T3000 software uses this port to discover new devices on the network.

    TCP 502 is the Modbus port, even if you're using bacnet on the project certain blocks of data get passed around in Modbus format.

You can use the tools at portforward.com to check that port forwarding is working correctly.

[TBD]: Get rid of the scan on port 1234, scan on port 47898 instead.

Here are the settings for a typical setup showing the various port forwards for a wingate router in this example. Your router will have a different appearance but similar general settings to configure.

To set up T3000 to log into a remote site click on the building icon at tab1, fill in an entry for a new building at line 2 and select this building by right clicking on a given line, then hit 'Select' to set that building as the current connection. Tab2 shows that the 'Remote Building' has been selected. The protocol is set at tab3, in this case its set to 'Remote' building as opposed to 'Auto' where the PC and the T3 controllers are on the same subnet. At tab4 we can set up an IP if your remote site has a fixed IP, or you can use a dynamic IP address by setting up an account at dyndns or any of the other supported services. Tab5 shows the Modbus TCP port, this is adjustable but it is suggested to set it as 502 unless there's a good reason to adjust it. The bacnet and scan ports are not adjustable.
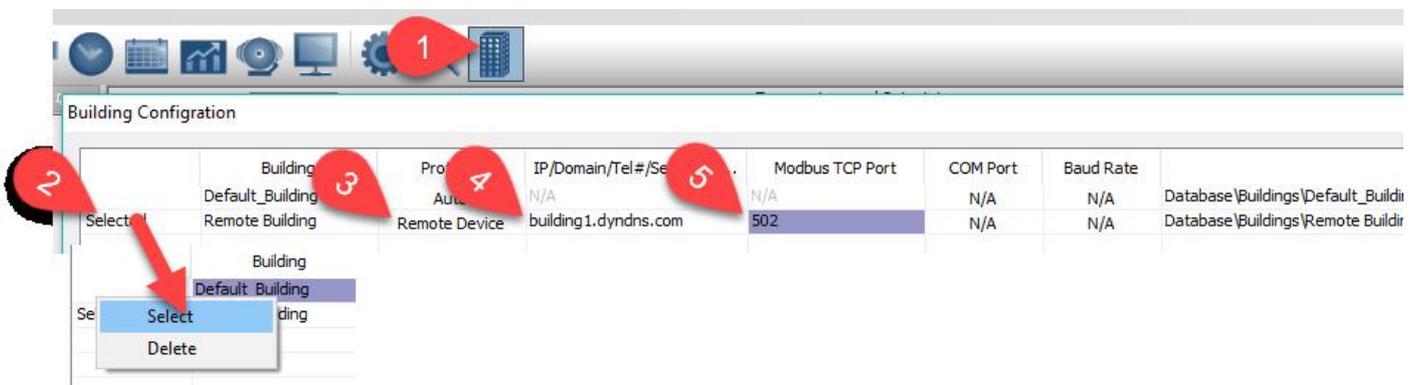
Select DDNS Server

<TODO>: Insert description text here...

User Name

<TODO>: Insert description text here...

Password

<TODO>: Insert description text here...

4 **Domai**

Domain : www.demo@dyndns.com

<TODO>: Insert description text here...

Domain : www.demo@dyndns.com

<TODO>: Insert description text here...

CONTROL

The T3000 software presents the operator with a number of work screens for configuring the points needed to interact with the T3000 control panels. These work areas include screens for "INPUTS", "OUPUTS", "SCHEDULES" and "VARIABLES" to name a few. The behavior of the controller is defined in these work screens and the information is stored in the 'program' file for the controller. This chapter explains each of the various tables.

The Control Menu as displayed:
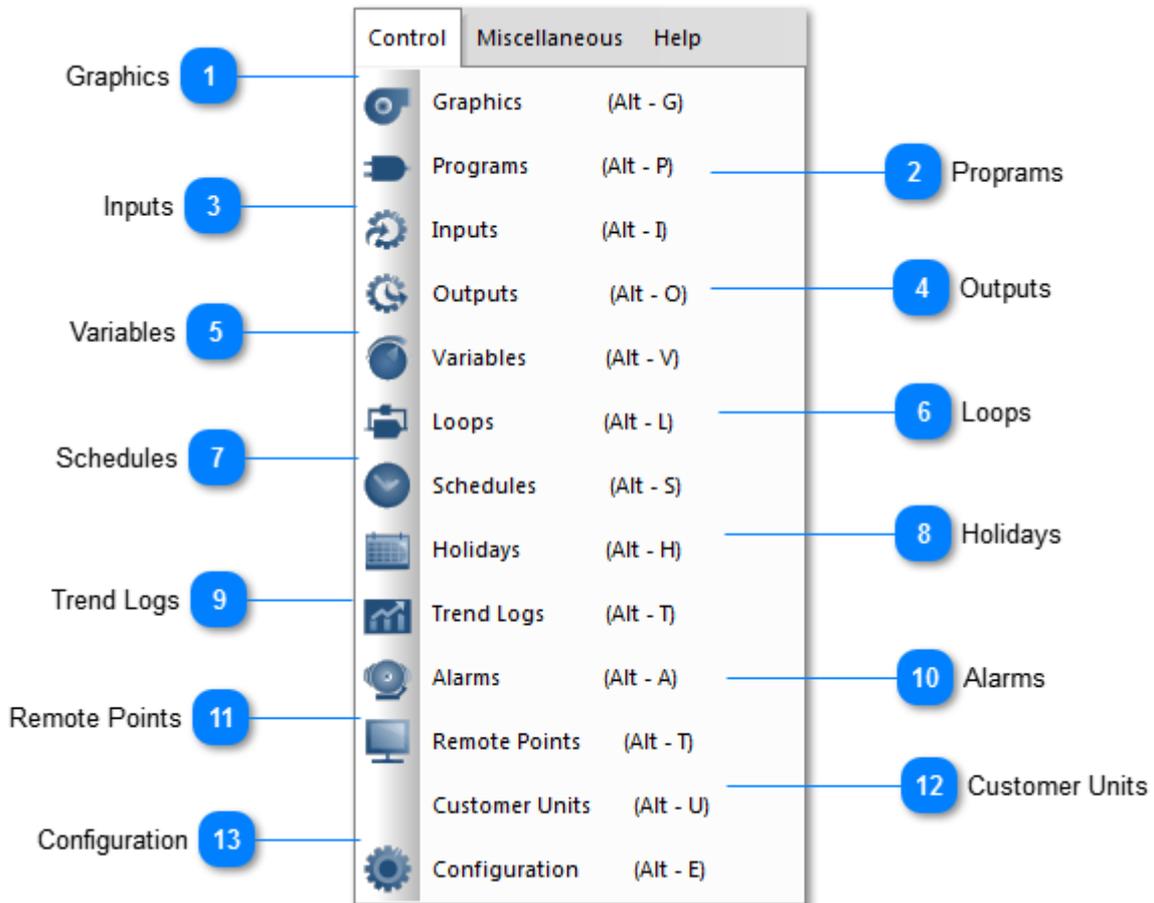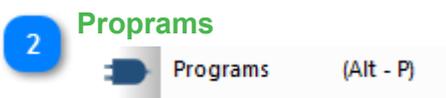
Graphics

Graphics   (Alt - G)

<TODO>: Insert description text here...

Proprams

Programs   (Alt - P)

<TODO>: Insert description text here...

Inputs

Inputs   (Alt - I)

<TODO>: Insert description text here...

Outputs

Outputs   (Alt - O)
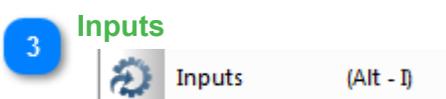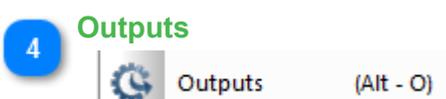
<TODO>: Insert description text here...

5 Variables

Variables   (Alt - V)

<TODO>: Insert description text here...

6 Loops

Loops   (Alt - L)

<TODO>: Insert description text here...

7 Schedules

Schedules   (Alt - S)

<TODO>: Insert description text here...

8 Holidays

Holidays   (Alt - H)

<TODO>: Insert description text here...

9 Trend Logs

Trend Logs   (Alt - T)

<TODO>: Insert description text here...

10 Alarms

Alarms   (Alt - A)

<TODO>: Insert description text here...

11 Remote Points

Remote Points   (Alt - T)

<TODO>: Insert description text here...

12 Customer Units

Customer Units   (Alt - U)
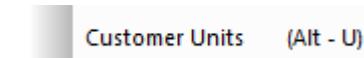
<TODO>: Insert description text here...

13 Configuration

Configuration   (Alt - E)

<TODO>: Insert description text here...

6.1 Graphic Screens

Hot Key: Alt-G

KEYWORD: GRP
Usage：GRP1,GRP2,GRP3...
How to show: when you use one of these items , the label of the item will be shown in the place where you use

Screens are used to group points of all types into one logical unit for display on the screen. Thus the operator is able to see how a change in one parameter is affecting other related parameters. This enables the operator to see and change the value and auto/manual status of points. The operator can also call up the history information of a point or the program associated with that group. The elements of a screen can be displayed on top of a graphic picture to augment the display of information.

A screen made up of other screens can be used as a menu. Screens may be nested several levels. [TO BE DONE] The TAB key is used as a quick exit key to the Main Menu. This is useful when many screens are nested and the operator does not want to escape through them all. The "Page-Up" and "Page-Down" keys will take you through consecutive screens.

[TO BE DONE] A Screen can be programmed to load up automatically after signing on by entering the Default Screen number in the "USER LIST" selection from Configuration (see Chapter 8 for more info). This screen then becomes the Main Menu for the user, letting them access only their particular room for example.

Sample Graphics screen grind. Hitting the [INSERT] or INS will drill down into a particular screen:

Graphic Screens provide the building operator with a view of the system and associated real time information. Graphic Screens are made up of two parts, the underlying graphical image and on top of that are various icons, inputs, outputs, schedules, and links to other screens. They are easy to customize on the fly so the both the designer and later, the building the operator can easily make changes to suit their working routine.
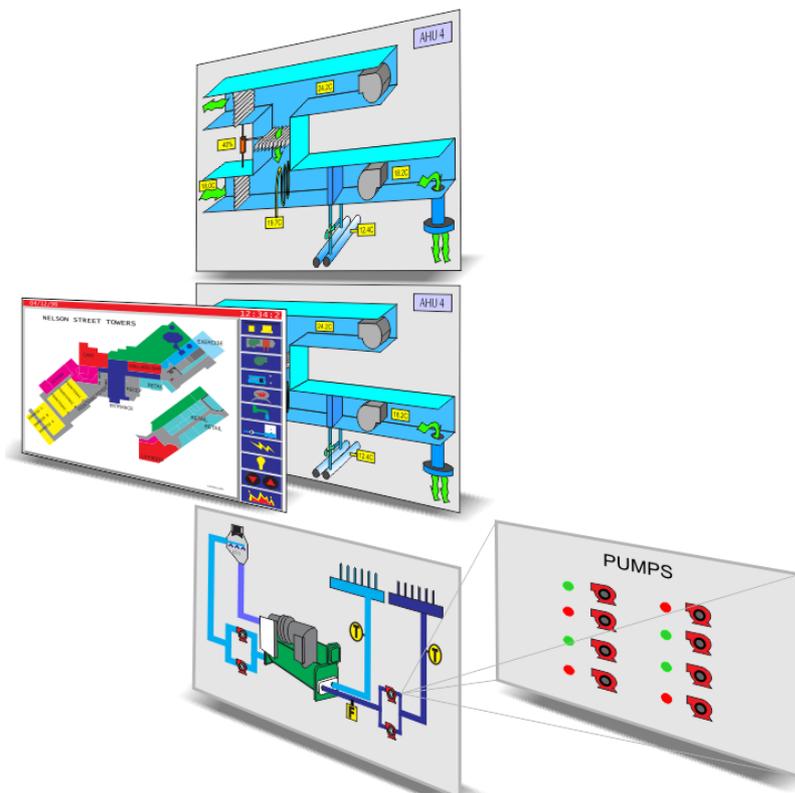
Graphical Screens can be nested so that the operator can drill down to more detailed screens. A large building will typically have an overall floorplan with hot spots leading to each of the zones.

Then for major systems there will be icons leading to the chiller room, boiler room and air handlers. The screens can contain links to screens from other controllers which allows you to build up large nested systems for campus size projects. The screens are saved on each of the controllers locally, a change made to one display will immediately be available to other users logging in from elsewhere so there are no syncing delays or conflicts to deal with.

The TAB key is used as a fast exit key to the Main Menu, this is a handy trick to get back to the home position quickly. The "Page-Up" and "Page-Down" keys bypass the nested menu system and bring up the next higher or lower numbered screen respectively. Hitting the ESC key pulls up the previous screen, regardless of whether the user arrived there by clicking on icons in the nested menu system or by using the "Page-Up" and "Page-Down" keys.

[TBD] A graphical screen can be associated with a particular user when they log on, this screen then becomes that users Main Menu. See the USER LIST section of this manual.

A sample 'Graphical Screens' work area is shown below. Each of the rows in this grid refer to one screen and each controller can host up to sixteen screens [TB: increase to 32]. The first row in this example contains the details for the first screen 'Air handling unit 1', the second line is for screen number two and so on.

Screens setup fields:

a) Full_label. . . . .   A 20 character descriptor.

b) Label . . . . . .   An 8 character label.
            [TO BE DONE] Typing this label into the command line let's you jump to this screen quickly, faster than using the mouse.

c) Picture file . . . . .The name of a graphic file to be splashed on the screen first, the underlay for this particular display.

Note: the name of picture must be less than 10 chars, otherwise ,T3000 only get 10 chars as the name of the file

To program or view an individual Screen press "Ins" while highlighting that row. The graphic display pops up, you can add or remove items from the display by hitting the insert key again. and then set how the item will display, such as showing the the short or long label text along with the value, the value only, color and icons.
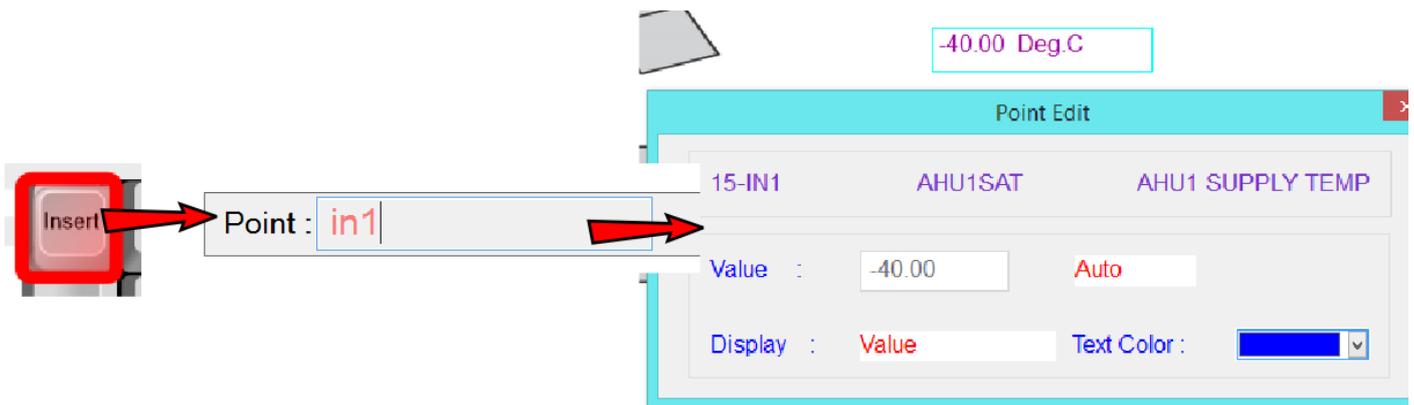
 [TO BE DONE] The work menu reappears if the cursor is moved to the top line of the display. If a graphic is loaded and you are using a mouse, the work menu appears when the mouse is moved to the top of the screen.

The Graphic display can contain up to 46 individual elements on a single screen. These elements can be any combination of the following:

a)   Points (GRPS, PRGS, INS, OUTS, VARS, CONS, WRS, ARS, AYS).

b)   Custom menus (i.e. MENU#2).

c)   [TO BE DONE] Command to run the System Program (i.e. MACRO#1).

d)   [TO BE DONE] Any of 50 keywords in Chapter 4.

e)   [TO BE DONE] TEXT File - use Text keyword.

f)   [TO BE DONE] Command to operating system, same as running the command from keyword.
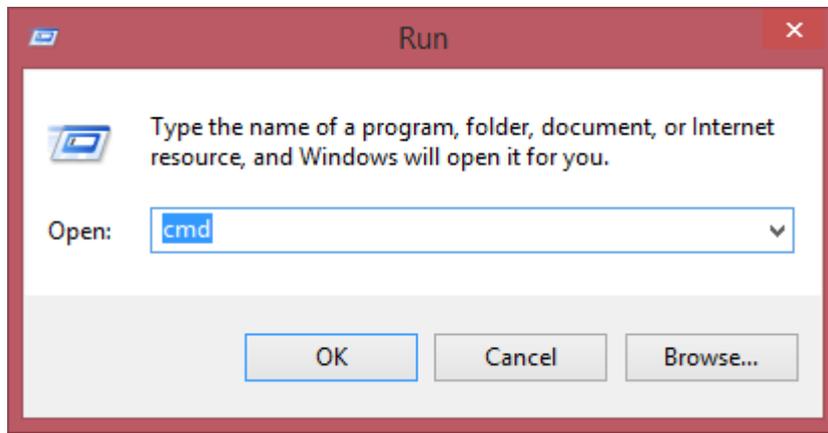
Graphic	Full Label	Label	Picture File	Refresh Rate	
☑ 1			8-7-20.png	15	
☐ 2			1	15	
☐ 3				15	
☐ 4				15	
☐ 5				15	
☐ 6				15	
☐ 7				15	
☐ 8				15	
☐ 9				15	
☐ 10				15	
☐ 11				15	
☐ 12				15	
☐ 13				15	
☐ 14				15	
☐ 15				15	
☐ 16				15	

Show Graphic

1 **Full Label**

Full Label

A unique twenty character descriptor used to identify the graphic display throughout the system.

2 **Label**

Label

A unique eight character descriptor to be used anytime this screen is referred to in the system. Use this name to identify it when placing elements on another display. In the image below we're currently adding items for screen 1 which is for AHU1 and we decide we'd like to add a link from here to jump to AHU2 whcih is screen number2. Just hit the insert key and type AHU2, the short hand name for screen number 2. We could also accomplish the same thing by typing the direct reference GRP2 as well. Once an item has a short hand name you can use either the direct reference or the shorthand name interchangeably.
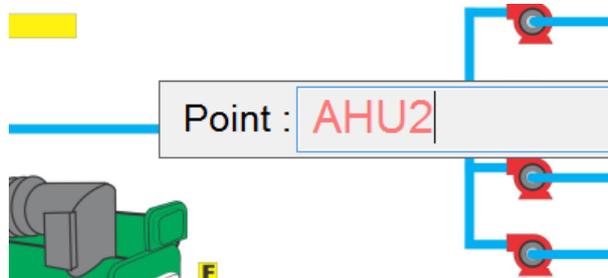
[TBD] Jump to this graphic display from the command line.

3 Picture File

Picture File

This is the file name for the image file which will be pulled up from the local hard disk as the underlay for this graphic display. It can be a jpeg, bmp or png file. Click on this column and you are presented with a browser box to select the image file from your hard disk.
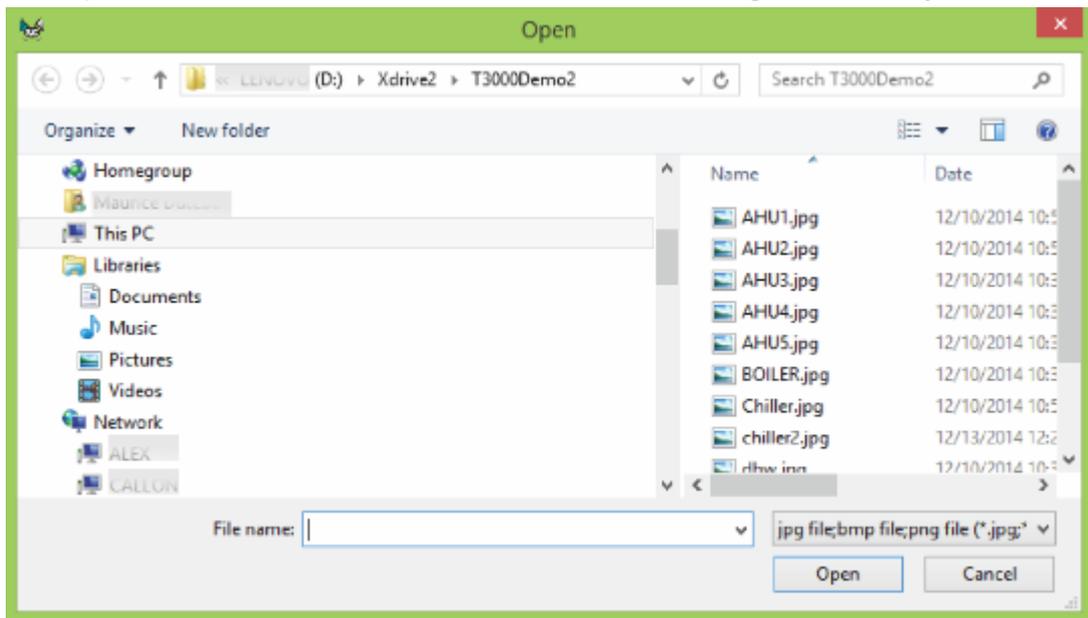
If the file is moved or renamed the software may not be able to locate the image file so a basic white display is presented with the data overlaid in the usual positions.

[TBD] Store the image file on the SD disk of the controller so that users logging in from other locations can download the image automatically.

4 Refresh Rate

Refresh Rate

Refresh rate is the interval which T3000 will update the screen with new data from the controller. The default setting is 15 seconds which is fine for most situations. Whenever a parameter is changed the

screen will also be updated immediately in the background. There are cases, such as with a slow GSM connection when you would want to set this value to a longer period. Setting a fast refresh rate can generate a lot of network traffic.

[TBD] The operator can also hit a hot key [F7] to manually refresh the data on command.

6.1.1 Insert Point

Point : [   ]

DEALING WITH ELEMENTS ON A SCREEN

When an element in a screen is selected with the ENTER keystroke or the equivalent on the mouse buttons, the system response is as follows for each type:

SCREENS .	**(GRP)**	**. . . . . Calls that group.**
PROGRAM	*(PRG)*	*. . . . . Enters control bas*
INPUT.	**(IN)**	**Pop-up Menu.**
OUTPUT	**(OUT) .**	**Pop-up Menu.**
VARIABLE	**(VAR)**	**Pop-up Menu.**
WEEKLY ROUTINE. . .	*(WR)*	*Pop-up Menu.*
ANNUAL ROUTINE . .	*(AR)*	*Pop-up Menu.*
ANALOG MONITOR . .	*(AMON).*	*Displays monitor data menu*
DIGITAL MONITOR	*(DMON) .*	*Displays monitor data menu*
ARRAYS	*(AY) .*	*Displays array data.*
ARRAY ELEMENT . . .	*(AY1[?])..*	*Edit value.*
MACRO		*. Call Macro.*
MENU		*. Call Menu.*
DOS		*. Run DOS command.*
TEXT		*Text window popup.*
KEYWORD		*. Do function of Keyword*

Additionally, hitting the INS key or the equivalent mouse button on Elements will bring up the Pop-up work screen.

SCREENS NOTES:

1. If a Screen is intended to be displayed with a graphic file, but the user does not have the graphic, the points are displayed in two columns regardless of the position information.
2. No value is displayed if the point is a screen, analog monitor, array, or program.

NOTES:
GRP,IN,OUT,VAR :**SUPPORTED**
OTHERS **:TBD**

6.1.2 Edit Point
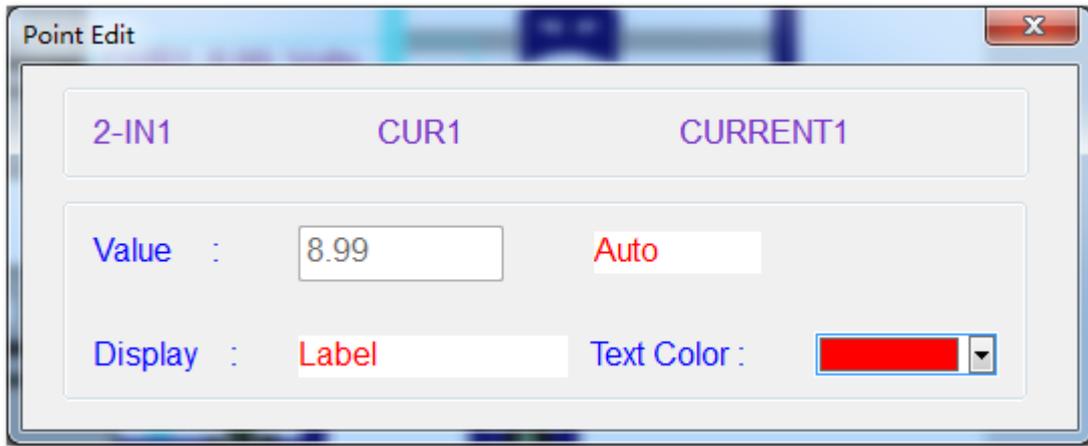
POP- UP WINDOW FUNCTIONS

A description of each of the items appearing in a pop-up window follows.

FIELD TITLE	FUNCTION OF FIELD
VALUE	Value of point.
HIGH LIMIT	Floating point number, above which, an analog point will change to the high color.
LOW LIMIT.	Floating point number, above which, an analog point will change to the low color.

DISPLAY. . . . . . .

Determines how the point in question will be displayed on the graphic screen. A number of options are avail able for displaying the points info:

20 char: Displays 20 character Full label.

Value: Displays value of point only.

Icon: Displays representative Icon only.
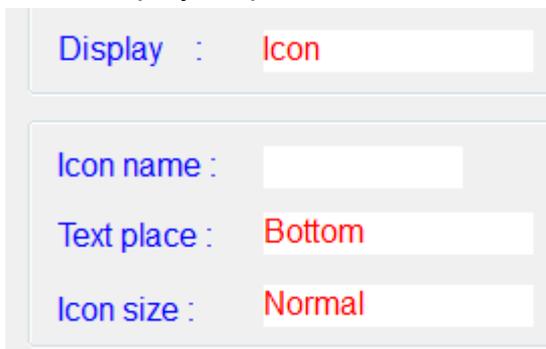
Icon/8ch: Displays representative and 8 character Label.

Display : Icon Label

Icon name :

Text place : Bottom

Icon size : Normal

Icon/20c: Displays representative and 8 character Full label.

Display : Icon Description

Icon name :

Text place : Bottom

Icon size : Normal

Icon/Val: Displays representative icon and point value.

Display : Label

8 char: Displays 20 character Label.

Display : Full Description

AUTO/MAN   . . . .
When in auto the point is operational and will read the value as determined by the programming. When in manual the point is
not operational and will remain at a fixed value as determined by the operator. To change from auto to manual or vice versa
simply highlight the field and press ENTER.

6.2 Programs

Hot Key: Alt-P

KEYWORD: PRG
Usage：**PRG1,PRG2,PRG3...**
How to show: when you use one of these items , the label of the item will be shown in the place where you use

Control Basic is the programming language of the T3000. To access an individual program press the "Ins" key while high lighting that program. The programming language is discussed in Chapter 10.
Sample Control Basic work screen:

Num	Full Label	Status	Auto/Manual	Size	Run Status	Label
☑ 1	AHU1 PROGRAM	ON	Auto	15	Normal	AHU1P
☐ 2	PRG2	OFF	Auto	50	Normal	AHU2P
☐ 3		OFF	Auto	0	Normal	AHU3P
☐ 4		OFF	Auto	0	Normal	BP
☐ 5		OFF	Auto	0	Normal	CHP
☐ 6		OFF	Auto	0	Normal	
☐ 7		OFF	Auto	0	Normal	
☐ 8		OFF	Auto	0	Normal	
☐ 9		ON	Auto	0	Normal	
☐ 10		ON	Auto	0	Normal	
☐ 11		ON	Auto	0	Normal	
☐ 12		ON	Auto	0	Normal	
☐ 13		ON	Auto	0	Normal	
☐ 14		ON	Auto	0	Normal	
☐ 15		ON	Auto	0	Normal	
☐ 16	COUNT	ON	Auto	36	Normal	

Control Basic set-up fields:

1 **Full Label**

Full Label

A 20 character descriptor of the point.

2 **Status**

Status

Indicates whether the program is running or not (ON/OFF).

3 **Auto/Manual**

Auto/Manual

In "Auto" the running of the program can be controlled by either the program timer or another program. In "Manual" the program can be stopped and started by the operator by toggling
the status field.

Size

Size

The length in bytes of the program, maxi mum size is 2500 bytes.

Run Status

Run Status

The time between each running of the program (mins: secs).

Label

Label

An 8 character descriptor of the point.

See Chapter 10 for more information on how to program Control Basic and use the Control Basic Editor

Num	Full Label	Status	Auto/Manual	Size	Run Status	Label
1	AHU1 PROGRAM	ON	Auto	24	Normal	AHU1P
2	PRG2	OFF	Auto	50	Normal	AHU2P
3						
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						
16	COUNT					

Bacnet Program IDE

Send (F2)   Clear (F3)   Load File (F7)   Save File (F6)   Refresh (F8)

```
10 IF TIME-ON( AHU1FAN ) > 20 THEN STOP AHU1FAN
20 IF TIME-OFF( AHU1FAN ) > 5 THEN START AHU1FAN
30  VAR1 = AHU1FAN
```

Programming   Insert

Programs pool size :	2000	Bytes
Programs size :	50	Bytes
Free memory :	1950	Bytes

Inputs

Hot Key: Alt-I

KEYWORD: IN

Usage：IN1,IN2,IN3...

The T3 controllers use 'Inputs' to sense the real world conditions using temperature sensors, switches and various transducers. The signal comes into the controller in the form of a control signal known as either digital or analog. A digital input has two states, the on and off states while an analog signal varies gradually from 0 to 100% of full scale. The T3 controllers inputs are 'universal' in that any input can be software defined to operate as analog or digital.

The sensor signal conditioning hardware is also universal where many sensor types can be wired directly to the controller. Thermistors, dry contacts, 0-10V signals and 4-20ma signals can all be wired directly to the controller with no extra transducer or resistor dividers to add externally. With most controllers, selecting a signal type will require moving a jumper but on the T3 controllers this is set up in the software.

Each T3 controller has a fixed number of inputs and outputs on board as shown in the table below. The software can handle more I/O than is on board, to take advantage of this you can expand the controller with expansion modules to a maximum of 64 inputs and 64 outputs. When the application calls for more than this you'll need to add another controller. Data can be shared easily between controllers allowing the system to grow very large but its best to keep a particular system on a given panel to keep the system compartmentalized.

For adding expansion points to the controller I/O count please refer to the section on Expansion I/O

T3 Family I/O List

	Name	IN	DO	AO
Controllers	T3-BB	32	12	12
	T3-LB	16	6	4
	T3-TB	8	6	2
Expansion	T3-8Ao	8	6	8
	T3-22i	22	0	0
	T3-Pt12	12 x Pt sensors		
	T3-6ct	6 x Ct sensors		

To get to the Inputs screen, hit Alt-I or click on the inputs icon in the menu system:
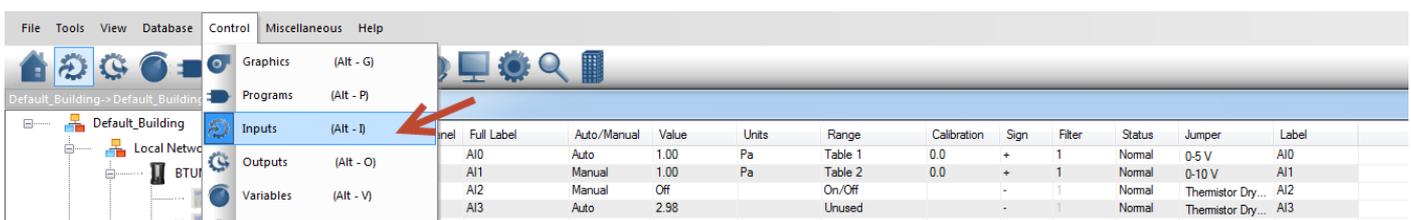
Sample Input work screen:

Input	Panel	Full Label	Auto/Manual	Value	Units	Range	Calibration	Sign	Filter	Status	Jumper	Label
IN1	1	SUPPLY AIR TEMP	Auto	-40.00	Deg.F	10K -40 to 250	0.0	-	5	Open	Thermistor Dry...	AHU1SAT
IN2	1	RETURN AIR TEMF	Auto	248.00	Deg.F	10K -40 to 250	0.0	-	5	Shorted	Thermistor Dry...	AHU1RAT
IN3	1	MIXED AIR TEMP	Auto	-40.00	Deg.F	10K -40 to 250	0.0	-	5	Open	Thermistor Dry...	AHUMAT
IN4	1	ROOM TEMP	Auto	-40.00	Deg.F	10K -40 to 250	0.0	-	5	Open	Thermistor Dry...	AHU1RT
IN5	1	FAN STATUS	Auto	On		Off/On		-	5	Normal	Thermistor Dry...	AHU1FS
IN6	1	OUTSIDE AIR TEMI	Auto	-40.00	Deg.F	10K -40 to 250	0.0	-	5	Open	Thermistor Dry...	OAT
IN7	1	IN 7	Auto	2.99		Unused	0.0	-	5	Normal	Thermistor Dry...	IN7
IN8	1	IN 8	Auto	2.99		Unused	0.0	-	5	Normal	Thermistor Dry...	IN8
IN9	1	IN 9	Auto	2.99		Unused	0.0	-	5	Normal	Thermistor Dry...	IN9
IN10	1	IN 10	Auto	2.99		Unused	0.0	-	5	Normal	Thermistor Dry...	IN10
IN11	1	IN 11	Auto	2.99		Unused	0.0	-	5	Normal	Thermistor Dry...	IN11
IN12	1	IN 12	Auto	2.99		Unused	0.0	-	5	Normal	Thermistor Dry...	IN12
IN13	1	IN 13	Auto	2.99		Unused	0.0	-	5	Normal	Thermistor Dry...	IN13

• **FULL LABEL**

A unique twenty character descriptor used to identify the item throughout the system. Used for programming, trend logging, reports, and showing on the user displays.

Auto/Manual

The auto/manual field allows the operator override the real world conditions during commissioning and testing. Hitting the enter key with the cursor in this field will cycle the mode from auto to manual and back. Normally the input is in auto and the value will update according to the real world conditions. In manual mode, the value is fixed and the operator is in charge of the input and can type a new simulated value in the 'value' cell.

If you want to see how the rest of the system will respond when the outdoor air temperature hits 0C, 32F it is possible to simulate this condition by setting the value to 0C. First toggle the auto/manual cell to manual mode and then maneuver to the 'Value' field to enter the value you want to simulate, 0C in this example. Now the rest of the system will behave and operate as if the outdoor temperature is really at 0 Deg C. An alarm will be created after five minutes if the operator forgets to return the input to auto mode.

Value

This cell shows the current value of the input. If the auto/manual field is in auto this will show the true real time data from the inputs. When the auto/manual setting is set to manual the value will stay at a fixed value. Assuming the mode is in manual mode, if the input has a binary range you can toggle between the two values by simply hitting enter on the value cell. When the input is configured as an analog item you can edit it by typing in a new value, again only when the item is in manual mode. Most of the time the input will be in auto mode and the value you see here will reflect the true conditions in the field.

Units

Units

The engineering units for the inputs, this field is fixed when you set the Range field. You can add custom engineering units in the 'custom tables' section.

Range

The range setting is used to select the type of sensor connected to each of the inputs. There are many pre-defined ranges to select from, both analog and binary. The user can also define their own ranges using the custom range feature described further below.

To set the range highlight the field, press ENTER, and select either Analog, Digital or Custom Digital. Then enter the desired range number.

6 Calibration

Calibration

e) Cal.(Calibration)..   The value in this field is added into the measured value to ad just for sensor in accuracy. T mum correction is +/- 30. If the correction amount is between -3.0 and +3.0 there is 1 decimal point in the co otherwise there is no decimal point.

Calibration (CAL)   Adds or subtracts an offset to the analog input readings for calibration purposes. In th example below, the input shows 24.1C but our reference sensor shows 25.1C, we enter a 1 in the calibra and now the controller and meter values agree at 25.1C. The resolution of the calibration field is 1 decima Use the 'Sign' field to change the offset to either a positive or negative value.

NUM	Full Label	Auto/Manual	Value	Units	Range	Calibration	Sign	Filter	Status	Label
1	ROOM TEMP	Auto	24.20	Deg.C	10K -40 to 120	1.0	+	0	Normal	ROOMTEMP

7 Filter

Filter

Filter is the number of samples that are averaged to give the value. A larger filter will slow the response of the system to a change at the input, but will filter out fluctuations that are not wanted.

8 Label

Eight character descriptor, a unique ID used to refer to this input throughout the system: programming, trend logs and so on.

9 Status

Status

Normal means the sensor is operating in the usual range. Open or Short means the input is short circuited, or open circuit. The cabling and sensor should be checked.

10 <TODO : Control Name>

Jumper

<TODO>: Insert description text here...

Pulse Counting

<TODO>:Insert pulse counting details here

Custom Digital Ranges

Ranges 23 thru 30 are Custom Digital Ranges, the user can edit the names of these ranges to suit the application. In the example below we have set up a range as Tank1 and Tank2 to be used to indicate which of the two tanks are being filled. The custom ranges you define here can be also used by the outputs and variables, they all share the same custom digital ranges.

To edit the custom digital ranges, click on the "Edit" button which brings up the dialog below where you can type in the name for the OFF and ON state for up to eight custom digital ranges.

In the example below we have a custom range for an item which will show TANK1 when the item is off and TANK2 when the item is on. For an input, the system will show "TANK1" when the contacts are open and 'TANK2 'when the input is shorted. In the case of the outputs, when the output is off, the value will show as TANK1 and when the output is on 'TANK2' is shown. In the case of variables, when the value is zero the word 'TANK1' will be showing in the value column and "TANK2" when the value is 1.

The direct-inverse column lets you reverse the normal logic which is useful if you want to reverse the state of the LED and control the value of the item (0 or 1) when used in programming and logging. If we set the range to inverse, everything written in the above paragraph is reversed. The LED will be on for an input when the terminal is shorted instead of open and vice versa. The value of TANK1 becomes 1 in all programming and logging, instead of zero in the direct mode.

Custom Digital Range

NUM	Digital Off	Digital On	Direct_Invers
1	TANK1	TANK2	Invers
2			Direct

Num

NUM
1
2
3
4
5
6
7
8

<TODO>: Insert description text here...

Digital Off

Digital Off
TANK1

<TODO>: Insert description text here...

Digital On

Digital On
TANK2

<TODO>: Insert description text here...

Direct_Invers

Direct_Invers
Direct
Direct
Direct
Direct
Direct
Direct
Direct
Direct

<TODO>: Insert description text here...

Range Setting

<

The following analog ranges are available:

1.Y3K -40 to 150 Deg.C

2.Y3K -40 to 300 Deg.F, same as range 1 but using Degrees F.

3.10K -40 to 120 Deg.C , Type 2 thermistor.

4.10K -40 to 150 Deg.F, same as range 3 using Deg F.

5.G3K -40 to 120 Deg.C

6.G3K -40 to 250 Deg.F

7.KM10K -40 to120 Deg.C   (used for Temco LCD sensor)

8.KM10K -40 to250 Deg. F

 9.   A10K -50 to 110 Deg. C

10.   A10K -60 to 200 Deg. F

11.   0 to 5 Volts

12.   0 to 100 Amps (over 0-5 Volts)

13.   0 to 20 mA

14.   0 to 20 psi (over 0-5 Volts)

15.   0 to 2^{32} counts (4,294,967,296 pulse counter)

16.   0 to 3000 FPM

17.   0 to 100 % (0-5 Volts)

18.   0 to 100 % (4 - 20 mA)

19.   0 to 255 pulse

20-24. Table1 through Table 5

25.   Unused

The following digital input ranges are available. The word to the left will show when the input is zero volts or short circuit. The word on the right side of the slash will show when the input is open circuit. If the contact is reverse then you can select the number to the right of the slash.

For example: Select '1' for the Off/On range. When the input is short circuit the word Off will show on the controller screens and logic and when the input is 3V or open circuit, the word On will show everywhere in the system. If the contacts are the inverse of this then select 12, now the word On will show when the input is short circuit and Off will show when the input is open circuit.

1/12 Off/On	2/13 Close/Open	3/14 Stop/Start	4/1
5/16 Norm/Alarm	6/17 Norm/High	7/18 Norm/Low	8/1
9/20 Cool/Heat	10/21 Un/Occupied	11/22 Low/High	

TIP: The status LED's for each input are illuminated when there is an open circuit and off when the input is shorted. It will be most intuitive to technicians and personnel in the field if the LED is on when the equipment is on and OFF when it is actually off. You can connect to a normally closed set of contacts, and select range 1. When the relay is closed when the equipment is off and the LED will be off, when the equipment is on the relay is open and the LED goes on. You can also connect to a normally open set of contacts and select the inverted logic, ie range 12 instead of range 1. Then the inverted logic applies, the LED will be the reverse of when range 1 is selected and match the state of the equipment with a normally open set of contacts.

Select Range Number   ☒

Enter Units Number :   [34]   [OK]   [Cancel]   [_____]

Digital Units

○ 0. Unused		○ 10. Unoccupy/Occupy	○ 21. Occupy/Unoccupy
○ 1. Off/On	○ 12. On/Off	○ 11. Low/High	○ 22. High/Low
○ 2. Close/Open	○ 13. Open/Close	○ 23. /	[Edit]
○ 3. Stop/Start	○ 14. Start/Stop	○ 24. /	
○ 4. Disable/Enable	○ 15. Enable/Disable	○ 25. /	
○ 5. Normal/Alarm	○ 16. Alarm/Normal	○ 26. /	
○ 6. Normal/High	○ 17. High/Normal	○ 27. /	
○ 7. Normal/Low	○ 18. Low/Normal	○ 28. /	
○ 8. No/Yes	○ 19. Yes/No	○ 29. /(#	
○ 9. Cool/Heat	○ 20. Heat/Cool	○ 30. /	

Input Analog Units

○ 31. Y3K -40 to 150 Deg.C	○ 40. A10K -60 to 200 Deg.F	○ 50. Table 1
○ 32. Y3K -40 to 300 Deg.F	○ 41. 0.0 to 5.0 Volts	○ 51. Table 2
○ 33. 10K-40 to 120 Deg.C(Type2)	○ 42. 0.0 to 100 Amps	○ 52. Table 3
⊙ 34. 10K-40 to 250 Deg.F(Type2)	○ 43. 0.0 to 20 ma	○ 53. Table 4
○ 35. G3K -40 to 120 Deg.C	○ 44. 0.0 to 20 psi	○ 54. Table 5
○ 36. G3K -40 to 250 Deg.F	○ 45. Low Speed Count	○ 55. High Speed Coun
○ 37. 10K-40 to 120 Deg.C(Type3)	○ 46. 0.0 to 3000 FPM	○ 56. HZ
○ 38. 10K-40 to 250 Deg.F(Type3)	○ 47. 0 to 100 %(0-5V)	
○ 39. A10K -50 to 110 Deg.C	○ 48. 0 to 100 %(4-20ma)	
	○ 49. 0.0 to 10.0 Volts	

Note: the input card has jumpers for three jumper settings for different types of sensors. The default setting from the factory is the 'thermistors & dry contacts' setting since these are the most common signal type in the HVAC world. In this setting a 10k pullup resistor to 3V creates a weak pullup voltage for excitation for the field device . This excitation voltage is pulled down by the contacts or thermistors to generate the signal voltage to the controller.

Next is a 10V setting, this is used for 0-10V transducers such as humidity, current and C02 sensors. Then there is a 4-20 mA setting for this type of transducer. And finally, by removing the jumper altogether the field device signal goes direct to the a/d converter of the controller. The cpu handling the a/d conversions is a three volt chip therefore the maximum signal you should apply to the input is three volts. If you have a five volt signal you can select the 10V range.

Select Range Number window

1   **4**   **5**

Select Range Number

Enter Units Number : [50] [OK] [Cancel] [_____]

Digital Units

Unused, default — **2**

Direct logic, digital ranges — **3**

○ 0. Unused		○ 10. Unoccupy/Occupy ○ 21. Occupy/Unoccupy
○ 1. Off/On	○ 12. On/Off	○ 11. Low/High ○ 22. High/Low
○ 2. Close/Open	○ 13. Open/Close	○ 23. TANK1/TANK2 [Edit]
○ 3. Stop/Start	○ 14. Start/Stop	○ 24. /
○ 4. Disable/Enable	○ 15. Enable/Disable	○ 25. /
○ 5. Normal/Alarm	○ 16. Alarm/Normal	○ 26. /
○ 6. Normal/High	○ 17. High/Normal	○ 27. /
○ 7. Normal/Low	○ 18. Low/Normal	○ 28. /
○ 8. No/Yes	○ 19. Yes/No	○ 29. /
○ 9. Cool/Heat	○ 20. Heat/Cool	○ 30. /

6 — Additional Inverted Ranges

7 — Custom Digital ranges

Input Analog Units

Analog Ranges — **8**

○ 31. Y3K -40 to 150 Deg.C	○ 40. A10K -60 to 200 Deg.F	◉ 50. Table 1 (psi)
○ 32. Y3K -40 to 300 Deg.F	○ 41. 0.0 to 5.0 Volts	○ 51. Table 2
○ 33. 10K-40 to 120 Deg.C(Type2)	○ 42. 0.0 to 100 Amps	○ 52. Table 3
○ 34. 10K-40 to 250 Deg.F(Type2)	○ 43. 0.0 to 20 ma	○ 53. Table 4
○ 35. G3K -40 to 120 Deg.C	○ 44. 0.0 to 20 psi	○ 54. Table 5
○ 36. G3K -40 to 250 Deg.F	○ 45. Pulse Count (Slow 1Hz)	○ 55. Pulse Count (Fast 100Hz)
○ 37. 10K-40 to 120 Deg.C(Type3)	○ 46. 0.0 to 3000 FPM	○ 56. HZ
○ 38. 10K-40 to 250 Deg.F(Type3)	○ 47. 0 to 100 %(0-5V)	
○ 39. A10K -50 to 110 Deg.C	○ 48. 0 to 100 %(4-20ma)	

9 — Custom Analog Ranges

10 — Pulse Counters

---

1 Edit box for Range setting

Enter Units Number : [50]

Use this box to directly enter the range number from the table below, or select any of the radio buttons below to select the particular range.

2 Unused, default

○ 0. Unused

3 Direct logic, digital ranges

○ 1. Off/On

○ 2. Close/Open

○ 3. Stop/Start

○ 4. Disable/Enable

○ 5. Normal/Alarm

○ 6. Normal/High

○ 7. Normal/Low

○ 8. No/Yes

○ 9. Cool/Heat

4 **Inverted digital ranges**
- ○ 12. On/Off
- ○ 13. Open/Close
- ○ 14. Start/Stop
- ○ 15. Enable/Disable
- ○ 16. Alarm/Normal
- ○ 17. High/Normal
- ○ 18. Low/Normal
- ○ 19. Yes/No
- ○ 20. Heat/Cool

5 **Continuation of direct logic ranges**
- ○ 10. Unoccupy/Occupy
- ○ 11. Low/High

6 **Additional Inverted Ranges**
- ○ 21. Occupy/Unoccupy
- ○ 22. High/Low

7 **Custom Digital ranges**
- ○ 23.   TANK1/TANK2
- ○ 24.   /
- ○ 25.   /
- ○ 26.   /
- ○ 27.   /
- ○ 28.   /
- ○ 29.   /
- ○ 30.   /

8 **Analog Ranges**

Input Analog Units

- ○ 31. Y3K -40 to 150 Deg.C
- ○ 32. Y3K -40 to 300 Deg.F
- ○ 33. 10K-40 to 120 Deg.C(Type2)
- ○ 34. 10K-40 to 250 Deg.F(Type2)
- ○ 35. G3K -40 to 120 Deg.C
- ○ 36. G3K -40 to 250 Deg.F
- ○ 37. 10K-40 to 120 Deg.C(Type3)
- ○ 38. 10K-40 to 250 Deg.F(Type3)
- ○ 39. A10K -50 to 110 Deg.C
- ○ 40. A10K -60 to 200 Deg.F
- ○ 41. 0.0 to 5.0 Volts
- ○ 42. 0.0 to 100 Amps
- ○ 43. 0.0 to 20 ma
- ○ 44. 0.0 to 20 psi
- ○ 45. Pulse Count (Slow 1Hz)
- ○ 46. 0.0 to 3000 FPM
- ○ 47. 0 to 100 %(0-5V)
- ○ 48. 0 to 100 %(4-20ma)

Custom Analog Ranges

9

- ◉ 50. Table 1 (psi)
- ○ 51. Table 2
- ○ 52. Table 3
- ○ 53. Table 4
- ○ 54. Table 5

<TODO>: Insert description text here...

Pulse Counters

10

- ○ 45. Pulse Count (Slow 1Hz)   ○ 55. Pulse Count (Fast 100Hz)

<TODO>: Insert description text here...

6.4 Outputs

Hot Key: Alt-O

KEYWORD: OUT
Usage: 10 OCCUPIED THEN START OUT1 ELSE STOP OUT1

Outputs are the control points which send a signal from the T3000 out to the field devices like fans, pumps, dampers and valves. Outputs maybe either digital with two states such as with a fan or pump which can be on or off. Or they can be analog with continuously variable state such as with a damper motor that modulates from 0 to 100%. Each T3 controller can has a certain number of on board outputs, in the case of the BB there are 24 outputs which can be expanded up to a maximum of 64 outputs by adding expansion modules. Generally the system designer will try to keep all the control points for a particular machine or equipment room on the same controller. If there are more than 64 outputs in a project such as a large mechanical room with several chillers, pumps and valves working together, it is a simple matter to add another T3 controller and share a few variables over the network to create a seamless solution.

See the section titled Expansion Points for how to add expansion i/o to the controller.

TYPE OF MINIPANEL	INNER INPUTS	INNER DIGITE OUTPUTS	INNER ANALOG OUTPUTS	OUTER II
MP24	32	12	12	32
MP10	16	6	4	48
MP8	8	6	2	56

Sample Output work screen:

FULL LABEL

A unique twenty character descriptor used to identify the item throughout the system.
Used for programming, trend logging, reports and so on.
A 20 character descriptor which is usually used to associate a given point with the physical location.

Auto/Manual

Use the auto/manual feature to temporarily take control of the input in manual mode, this is useful for commissioning and debugging. For example, if you want to see how the rest of the system will respond when the outdoor air temperature hits 0C, 32F, it is possible to simulate this condition. First move the cursor to the input which is set up to measure the outside air temperature for the building, then set the A/M field to manual and finally move over to the Value field to enter in the simulated temperature of 0C. When the testing

is complete you must set the auto/manual field back to auto and the input will return to showing the actual outdoor temperature. The auto/manual field has only two states: Auto and Manual. Hitting the enter key with the cursor in this field will cycle the mode from auto to manual and back.

When in auto the out put will accept commands from Control Basic. When in manual the output will remain at a fixed value. To change from auto to manual or vice versa simply highlight the

field and press ENTER.

3 Value

Value

The current value of the inputs. If the auto/manual field is in auto this will show the true real time data from the inputs. When the auto/manual setting is set to manual the value will stay fixed at whatever was last measured or written by the user.

This field contains the value of the output point which may be digital or analog depending on the range selected. To toggle a digital value highlight the field and press ENTER. To change an analog value highlight the field and press ENTER followed by the new value and ENTER. Note that it is only practical to change an input value when the point is in manual.

4 Units

Units

The engineering units for the inputs, this field is fixed when you set the Range field.

5 Label

Eight character descriptor, a unique ID used to refer to this particular item throughout the system: programming, trend logs and so on.

6 Range

The following analog ranges are available:
1.0 to 10 Volts
2.0 to 100 % Open
3.0 to 20 psi(actual output is 0-10 Volts)
4. 0 to 100 %
5.0 to 20 mA(actual output is 0-10 Volts)
6.Unused
The following digital output ranges are available (select 1 for Off/On, or 12 for On/Off):
1/12Off/On2/13Close/Open3/14Stop/Start4/15Dis/Enable
5/16Norm/Alarm6/17Norm/High7/18Norm/Low8/19No/Yes
9/20 Cool/Heat10/21 Un/Occupied 11/22 Low/High

Select the range according to the following convention: the state on the right side of

the slash (/) is has a value of 0 Vdc, and on the right side of the slash the value is

13 V dc (0 Vdc/13 V dc). Only when the output is 13 V dc the LED will turn on

As a recommendation: use normally closed relays if the equipment need to be operational in case of a power failure. Then select digital output 12 (On/Off). When the equipment is commanded to turn off, the volt age will be 13 V dc , opening the electrical

circuit, thus turning the equipment off.

Ranges 23 thru 30 are Custom Digital Ranges

Select Range Number

Enter Units Number : `1` [OK] [Cancel] [_____]

Digital Units

- ⊙ 0. Unused
- ⊙ 1. Off/On
- ⊙ 2. Close/Open
- ⊙ 3. Stop/Start
- ⊙ 4. Disable/Enable
- ⊙ 5. Normal/Alarm
- ⊙ 6. Normal/High
- ⊙ 7. Normal/Low
- ⊙ 8. No/Yes
- ⊙ 9. Cool/Heat

- ⊙ 12. On/Off
- ⊙ 13. Open/Close
- ⊙ 14. Start/Stop
- ⊙ 15. Enable/Disable
- ⊙ 16. Alarm/Normal
- ⊙ 17. High/Normal
- ⊙ 18. Low/Normal
- ⊙ 19. Yes/No
- ⊙ 20. Heat/Cool

- ⊙ 10. Unoccupy/Occupy
- ⊙ 11. Low/High
- ⊙ 23. /
- ⊙ 24. /
- ⊙ 25. /
- ⊙ 26. /
- ⊙ 27. /
- ⊙ 28. /
- ⊙ 29. /(#
- ⊙ 30. /

- ⊙ 21. Occupy/Unoccupy
- ⊙ 22. High/Low
- [Edit]

Output Analog Units

- ⊙ 31. 0.0 -> 10 Volts
- ⊙ 32. 0.0 -> 100 %Open
- ⊙ 33. 0.0 -> 20 psi

- ⊙ 34. 0.0 -> 100 %
- ⊙ 35. 0.0 -> 100 %Cls
- ⊙ 36. 0.0 -> 20 ma
- ⊙ 37. 0.0 -> 100 PWM

7 **Status**

Status

8 **HOA Switch**

HOA Switch

6.5 Variables

Hot Key: Alt-V

KEYWORD: VAR
Usage：10 IF VAR1 > 100 THEN STOP OUT1

Variables are virtual points used to store the many parameters in a typical system such as setpoints, modes and so forth. Once the variable is created it can be referred to by the user defined name in the user programs, displays, and logs. The user can define alarms for the variables as well. The range and the engineering units can be set up from the predefined ranges and also custom ranges & units can be created. Each T3 controller can store up to 128 variables.

Sample Variables work screen:

	Full Label	Auto/Manual	Value	Units	Label
	1	**2**	**3**	**4**	**5**
Variable	Full Label	Auto/Manual	Value	Units	Label
VAR1	TESTO METER HUM	Auto	47.900	%RH	TESTOHUM
VAR2	TESTO METER TEMP	Auto	29.400	Deg.C	TESTOTMP
VAR3	HEATMODE	Auto	On	Off/On	HEATMODE
VAR4	COOLMODE	Auto	Off	Off/On	COOLMODE
VAR5	AVERAGE CHAMBER TEMP	Auto	29.400	Deg.C	CHAMTEMP
VAR6	CHAMBER TEMP SETPT	Manual	50.000	Deg.C	TEMPSETP
VAR7	CHAMBER HUM SETPT	Manual	80.000	%	HUMSETPT
VAR8	SYSTEM ON	Manual	On	Off/On	SYSTEMON
VAR9	CHANGEOVER FLAG	Auto	Off	Off/On	CHANGFLG
VAR10	CHANGEOVER TIMER	Auto	00:00	Time	CHANGTMR
VAR11	CHANGEOVER DELAY	Manual	01:00	Time	CHANGDEL

Page 46

1 **Full Label**

A unique twenty character descriptor used to identify the item throughout the system. Used for programming, trend logging, reports and so on.

2 **Auto/Manual**

The auto/manual field has only two states: Auto and Manual. Hitting the enter key with the cursor in this field will cycle the mode from auto to manual and back.
When in "Auto" the the true readings from the sensors and field devices will be shown here and update as conditions change. When in "Manual" the operator is in charge of the item, the real world sensor inputs are overridden. This feature allows the operator to temporarily take control of the input during commissioning and debugging. For example, if you want to see how the rest of the system will respond when the outdoor air temperature hits 0C / 32F you can manually command this condition.

| IN2 | CHAMBER TEMP2 | Auto | 120.00 | Deg.C | |
| IN3 | OUTSIDE AIR TEMP | Manual | 0.00 | Deg.C | |

First move the cursor to the row for the outside temperature sensor, IN3 in this case. Toggle the Auto/Manual field to Manual as shown in the green cell. Then maneuver right to the red tinted 'Value' column and type in the number you wish to set for the simulation, in this case 0 Deg C. The cell will flash solid red briefly while the data is written to the controller, then back to white to show that the value was written successfully. Now the rest of the system will operate based on an outdoor temperature of 0C. When the testing is complete you must set the auto/manual field back to auto and the input will return to showing the actual outdoor temperature.

3 **Value**

Value

The current value of the variable

4 **Units**

Units

Select Range Number

Enter Units Number : `55`   [OK]   [Cancel]

Digital Units

- ○ 0. Unused
- ○ 1/12. Off/On
- ○ 2/13. Close/Open
- ○ 3/14. Stop/Start
- ○ 4/15. Disable/Enable
- ○ 5/16. Normal/Alarm
- ○ 6/17. Normal/High
- ○ 7/18. Normal/Low
- ○ 8/19. No/Yes
- ○ 9/20. Cool/Heat

- ○ 10/21. Unoccupy/Occupy
- ○ 11/22. Low/High
- ○ 23. /
- ○ 24. /
- ○ 25. /
- ○ 26. /
- ○ 27. /
- ○ 28. /
- ○ 29. /
- ○ 30. /

Variable Analog Units

- ○ 31. deg.Celsius
- ○ 32. deg.Fahrenheit
- ○ 33. Feet per Min
- ○ 34. Pascals
- ○ 35. KPascals
- ○ 36. lbs/sqr.inch
- ○ 37. inches of WC
- ○ 38. Watts
- ○ 39. KWatts
- ○ 40. KWH

- ○ 41. Volts
- ○ 42. KV
- ○ 43. Amps
- ○ 44. ma
- ○ 45. CFM
- ○ 46. Seconds
- ○ 47. Minutes
- ○ 48. Hours
- ○ 49. Days
- ○ 50. Time

- ○ 51. Ohms
- ○ 52. %
- ○ 53. %RH
- ○ 54. p/min
- ● 55. Counts
- ○ 56. %Open
- ○ 57. CFH
- ○ 58. GPM
- ○ 59. GPH
- ○ 60. GAL

- ○ 61. CF
- ○ 62. BTU
- ○ 63. CMH

The following **direct** digital output ranges are available:

1. Off/On	2. Close/Open	3top/S
		Norm/
5. Norm/Alarm	6. Norm/High	7.ow
9. Cool/Heat	10. Un/ Occupied	11. Low/High

The following **inverted** digital output ranges are available:

12. On/Off	13. Open/Close	14. Start/Stop	15. Enable/Dis
16. Alarm/Norm	17. High/Norm	18. Low/Norm	19. Yes/No
20. Heat/Cool	21. Occupied/Un	22. High/Low	

Units 23 thru 30 are Custom Digital Ranges.

⑤ Label

Eight character descriptor, a unique ID used to refer to this particular item throughout the system during programming, creating trend logs and so on.

PID Loops Introduction

PID control stands for 'proportional, integral derivative' control which is the basis for many control strategies. A PID loop is calculated based on a setpoint, the feedback sensor reading and the difference between the two. Temperature control is the most common application of PID control in the HVAC world so we'll refer to temperatures and room setpoints in this discussion but PID loops can be used for any process like air air flow, building static pressure and water level to name a few.

Below is a graphical view of a typical fan coil application in a room. The room temperature is along the X axis while the Y axis is the PID loop response from zero to 100%. The heating and cooling setpoints are shown along the bottom on the X axis. As the room temperature goes cold towards the left of the graph the heat comes on in two stages as shown by the red and yellow blocks. As the room heats up towards the right of the graph, the cooling PID ramps up and causes the fan to come on in the purple area and the cooling valve to modulate open as shown in the blue area.
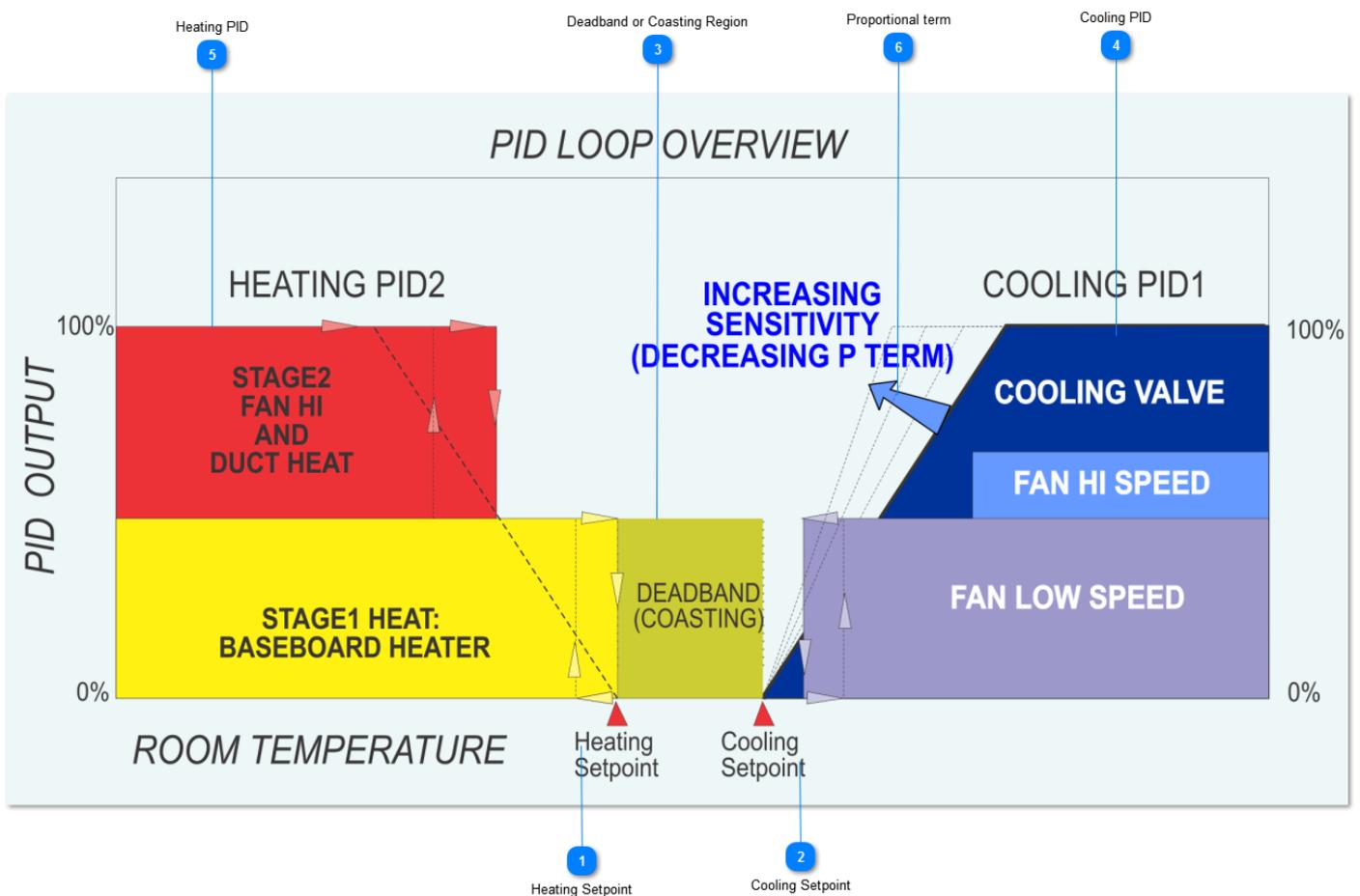
Below are some explanations for the various sections of this graphic.
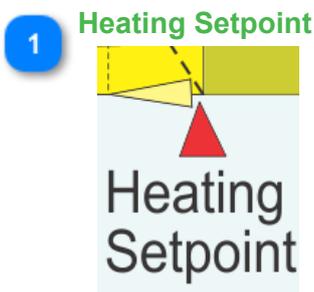
1 Heating Setpoint

The heating setpoint is shown here at pointer #1. In this example we have a separate heating and cooling setpoint since there is both heating and cooling in this application. The heating setpoint will be a variable which is normally adjusted warmer for the occupied period and cooler for the unoccupied period.

2 **Cooling Setpoint**

The cooling setpoint is shown here, it is also typically a variable in the T3 system which will be adjusted and managed by programs and schedules elsewhere in the T3 system according to the occupied and unoccupied period among other things. If the building takes advantage of load shedding the user program may nudge the setpoint warmer during periods of high demand.

3 **Deadband or Coasting Region**

The region between the heating and the cooling setpoints is called the deadband region. Make this deadband large for more energy savings, or it can be a small area where you want to control the room temperature to a very narrow band. A small deadband region will use more energy than a wide deadband because the system must jump between heating and cooling every time there is a small disturbance so keep the deadband large for more energy savings and less cycling on the equipment.

4 **Cooling PID**
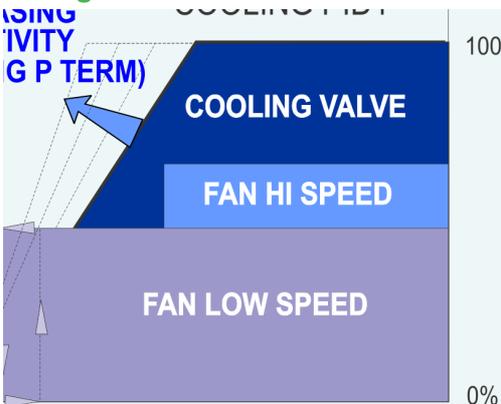
The cooling PID is shown graphically as the blue region. As the room temperature goes along the X axis towards the right of the graph, the cooling PID output rises from 0 till it 100% after which it can get no larger, hence the flat section towards the far right of the graph.

The low speed fan comes on when the PID hits a certain trigger level of about 10% and high speed kicks in when the PID term hits approximately 80%. There is hysterisis between each stage which is simplified somewhat in this diagram, the low speed shows the dotted line for the low speed fan coming on and another for the fan going off.

5 Heating PID

The heating PID is shown graphically as the dotted line in the red heating region. As the room temperature goes colder & colder along the X axis towards the left side, the heating PID output will rise along the dotted line till it hits 100%. After hitting the max it continues as a horizontal line further towards the left of the graphic. No matter how much colder the room gets, the PID loop response can get no larger than 100%. As the room temperature falls below the heating setpoint the PID output rises which in turn causes the first stage of heating to come on. As the room gets colder the PID output hits 50% and stage2 heat comes on.

The programming for the staging is done in the control basic programs and the PID loop is set up in the PID work screens described in the next section.

6 Proportional term

The blue arrow on the graphic shows how the PID term looks for different values of the 'P' setting which refers to the 'proportional' term in the PID terminology. When the temperature gets above setpoint the PID output will rise which starts the fan and opens the cooling valve to get the room temperature back to the setpoint.
A small P term translates into a STEEP line here which means the system will be more responsive, turning on the fan and opening the valve at the slightest bump in temperature over the sepoint.
A large P term decreases the slope of this line which means the room temperature will have to be hotter, further to the right on this graph, in order to get a 100% response from the PID loop.

Small P term = hyperactive PID loop, think of a chipmunk opening the valve at the slightest disturbance.
Large P term = lazy PID loop. Think of a lazy bear, he'll eventually get around to opening the valve.

Adjusting the P term is done in the PID work screens below.

PID Loop Work Screen

Hot Key: Alt-L

KEYWORD: PID

Usage: 10 DAMPER1 = **PID1**
20 **IF COOLING THEN VALVE1 = PID2**
30 **IF HEATING THEN VALVE1 = PID3**

The PID Loop work screen is where you configure the PID loops and manage the pid terms. Once the PID loop is created in the grid it can then be referred to in the user programs to modulate dampers & valves. There are 64 PID loops in the T3 controller which leaves plenty of spares in the typical application.

A short programming example shows how the PID loops can be used in the user programs:

OUTPUT1 is a modulating damper called 'DAMPER1' and OUTPUT2 is the 'FAN1'.
Normally the damper will follow PID loop 1 as programmed in the first line.
The second line is evaluated next and overrides the earlier line, when the fan is off the damper will close.

10   DAMPER1 = PID1
20   IF NOT FAN THEN DAMPER1 = 0

The following items describe the terms in the PID Loop work screen.

① Input or Feedback for the PID loop

These are the 'Inputs' or feedback for the PID loop, the sensors which are compared to the setpoint to calculate the response of the PID loop. A typical example is a temperature sensor in the chamber as shown above. PID loop 1 is a cooling PID loop and PID2 is a heating loop which both use the chamber temperature for the feedback. Type in the name of the sensor in the 'Input' column or you can refer to them directly by the direct reference such as IN1, IN2 and so on. Once you have entered a valid name for an input the row will update and show the current value and units. A variable can also be used in this column.

② PID Loop Current Output

These two columns show the current value of the PID loop output, it is the response of the PID loop to a given deviation from setpoint. When the loop is satisfied the output will be small, tending towards 0% when the input signal is satisfied or over setpoint. When the PID loop is under the setpoint the output will increase and eventually hit 100%, it can get no larger than a 100% response. The behavior of the PID loop, whether it is fast acting or slow acting or positive or reverse acting, is determined by the settings further down in this list.

When in "Auto" the PID loop is running and updating according to the real time readings from the sensors and setpoint settings. The user can set the PID loop to manual mode for commissioning and debugging, in this case the value of the output can be edited from the keyboard, overriding the true settings from the sensors and setpoints. Care should be taken to return the PID loop to auto when done or the loop will not update.

3 PID Loop Setpoint

This section allows you to configure the setpoint or target for the PID loop, hit enter on the cell in the 'Setpoint' column and type in the variable name or the direct address, VAR1 for example to define where teh PID loop will read its setpoint. Setpoints are normally set up as a [variable](#) which is managed with [schedules](#) and [control basic programs](#) but they can also be simple variables with no logic attached. In this case they are fixed and never change, the user can edit them and manage them from the [user graphic screens](#) with the keyboard.

The engineering units for the setpoint and the 'Input' should always be the same, in the example above both are showing as Degrees C. If there is a mismatch the PID loop will still function but it is good practice to have them set up properly. You cannot edit the Units column here however, changing the engineering units is done in the inputs or variables screens.

4 Proportional, Integral and Derivative Terms

The PID loop gets its name from these three parameters, the Proportional, Integral and Derivative terms showing here. The proportional term can be thought of as the "rigidity" of the PID loop and its response to a deviation from setpoint. A large value will make the valve or damper lazy while a small term will be more hyper active.

The units of the Prop term are in the same units as the feedback sensor, the input range, so if you are in DegC like the example above and have this set to 1, the response of the PID loop will operate from 0 to 100% given a one degree C deviation from setpoint. On the other hand if we have the P term set to 10, that means the temperature will have to deviate by a full 10 degrees C from setpoint to get a 100% response from the PID loop.

The Integral term accumulates error over time and adds it to the PID loop calculation. This can be thought of as a nudge from the PID loop if the temperature is coasting along a small amount above

or below setpoint the integral term will accumulate and eventually nudge the damper or valve open a little to achieve setpoint. This size of the I term works the opposite of the P term in that a large value will make the PID more active and a small I term will make the PID loop lazier.

The derivative term allows the PID loop to compensate for a rapidly changing process, if the temperature is quickly approaching setpoint then the D term will increase to slow down the PID response. For common HVAC processes this is not really a concern as the system changes slowly and the D term can generally be left at zero.

Bias is a one time offset added to the PID loop, this is not normally useful and left at zero. It is a feature dating back to the old pneumatic control days to hold a PID loop at a certain value when the setpoint is achieved.

When initially setting up PID loops you will leave the I, D and bias terms at zero because these terms operate behind the scenes and over time there will be windup which can hold the PID at a certain (hard to understand) value for a long while. The room temperature can be right at the setpoint but the valve is staying open at 50% for example, because in fact the valve may need to be open by this amount to maintain the setpoint. Once commissioning is done you can add a small amount of I term. D term is actually left at zero for the most part, as is Bias.

5 Action

Action

The PID loop can be either direct acting or reverse acting. When this term is set to positive, the slope of the PID action is positive and a deviation over the setpoint will cause an increase in the PID response. Conversely, a negative acting PID loop will be the opposite, as the feedback sensor shows the process is under the setpoint the PID will increase.

It is convenient to remember that the slope of the cooling PID graph in the previous section slopes up to the right which is a positive slope. The heating PID loop slopes up to the left which is a negative slope. Cool = '+' and Heat = '-'

In the case of air flow control for a VAV damper, the damper should open up when the flow is under setpoint so that would be a negative action.

6 Bias

Bias

Out put bias in %. This is the value which the controller will reach at equilibrium if no Integral term is used. If you use the integral term this bias will not be used. This bias feature was useful on the old pneumatic systems where you need a fixed output on the PID loop at setpoint.

PID LOOP MATH DISCUSSION

The output value of a controller is calculated according to the following formulas:

OUTPUT   =   Op + Oi + Od + Ob

Op   =   Proportional Output = A * 100 / Pb * (I - S)

Oi   =   Integral Output=A * R * t/60 * (I - S) + Oi

Od   =   Derivative Output=A * D*((I - S) - (It-1 - St-1)) / t

Ob   =   Bias

A   =   Action (+1 for direct acting, -1 for reverse acting)

Pb   =   Proportional Band

I   =   Input

S   =Setpoint

R   =Reset Constant = Repeats per hour Definition: One repeat per hour will cause an output change of one over one hour for an error of one.

D   =Rate Constant = Minutes   Definition: One minute will cause an output of one for an error changing by one per minute.

T   =Time in Minutes since last scan.

It-1   =Input on last scan.

St-1   =Setpoint on last scan.

Note: Controllers are executed once per second.

Output Limiting Procedure:

1.   The sum of Op + Od + Ob is limited to 0-100.

2.   Oi is added to the previous sum.

3.   The output is again limited to 0-100.

4.   Oi is adjusted by the amount that the output was limited in step 3.

Note: Oi is reset to zero when R is set to 0.

Oi is not reset when panel is shut off and on.

PID Loops Introduction

PID LOOP OVERVIEW

HEATING PID2

COOLING PID1

INCREASING
SENSITIVITY
(DECREASING P TERM)

PID OUTPUT

100%

STAGE2
FAN HI
AND
DUCT HEAT

COOLING VALVE

STAGE1 HEAT:
BASEBOARD HEATER

DEADBAND
(COASTING)

FAN LOW SPEED

0%

ROOM TEMPERATURE

Heating
Setpoint

Cooling
Setpoint

100%

0%

<TODO>: Insert description text here... And don't forget to add keyword for this topic

PID Loop Math Discussion

For those that are keen to understand the math behind the PID loops the following discussion explains the The output value of a PID Loop is calculated according to the following formulas:

OUTPUT = Op + Oi + Od + Ob

Op = Proportional Output = A * 100 / Pb * (I - S)

Oi = Integral Output=A * R * t/60 * (I - S) + Oi

Od = Derivative Output=A * D*((I - S) - (It-1 - St-1)) / t

Ob = Bias

A = Action (+1 for direct acting, -1 for reverse acting)

Pb = Proportional Band

I = Input

S =Setpoint

R =Reset Constant = Repeats per hour Definition: One repeat per hour will cause an output change of one over one hour for an error of one.

D =Rate Constant = Minutes   Definition: One minute will cause an output of one for an error changing by one per minute.

T =Time in Minutes since last scan.

It-1 =Input on last scan.

St-1 =Setpoint on last scan.

Note: The PID loop math is executed once per second.

Output Limiting Procedure:

1.   The sum of Op + Od + Ob is clipped at 100%, it cannot go to any number larger than that to avoid windup.

2.   Oi is added to the previous sum.

3.   The output is again limited to 0-100.

4.   Oi is adjusted by the amount that the output was limited in step 3.

Note: Oi is reset to zero when the I term is set to 0.

Oi is not saved when panel is shut off and on.

6.7 Schedules

Hot Key: Alt-S

KEYWORD: WR
Usage：WR1,WR2,WR3...
How to show: when you use one of these items , the label of the item will be shown in the place where you use

Weekly Routines are essentially time clocks with 4 ON/OFF times per day of the week. The week has 9 days to allow for two over rides in addition to the weekly schedule. These are used for holiday schedules, and are are specified by the Annual Routines Menu.

Sample Input work screen:

NUM	Full Label	Auto/Manual	Output	Holiday1	State1	Holiday2	State2	Label
☑ 1		Manual	ON		OFF		OFF	
☐ 2		Manual	ON		OFF		OFF	
☐ 3		Manual	ON		OFF		OFF	
☐ 4		Manual	ON		OFF		OFF	
☐ 5		Manual	ON		OFF		OFF	
☐ 6		Manual	ON		OFF		OFF	
☐ 7		Manual	ON		OFF		OFF	
☐ 8		Manual	ON		OFF		OFF	

Edit Schedule

NUM	Full Label	Auto/Manual	Output	Holiday 1	State1	Holiday 2	State2	Label
☑ 1	CHILLER SCHEDULE	Auto	OFF		OFF		OFF	
☐ 2		Auto	OFF		OFF		OFF	
☐ 3		Auto	OFF		OFF		OFF	
☐ 4		Auto	OFF		OFF		OFF	
☐ 5		Auto	OFF		OFF		OFF	
☐ 6		Auto	OFF		OFF		OFF	
☐ 7		Auto	OFF		OFF		OFF	
☐ 8		Auto	OFF		OFF		OFF	
☐ 9		Auto	OFF		OFF		OFF	
☐ 10		Auto	OFF		OFF		OFF	
☐ 11								
☐ 12								
☐ 13								
☐ 14								
☐ 15								
☐ 16								

(Keyboard keys overlay: Insert, Home, Page Up, Delete, End, Page Down)

Edit Schedule

Schedule Time

	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday	Holiday1	Holiday2
ON							01:00		06:03
OFF								06:03	
ON							06:03	00:25	00:01
OFF								00:01	03:20
ON					00:19		00:01	03:25	15:06
OFF						00:01			01:00
ON						03:19		01:00	15:00
OFF							01:00		06:03

Example for program

IF WR1 THEN START OUT1 ELSE STOP OUT1

1 Full Label

A unique twenty character descriptor used to identify the item throughout the system. Used for programming, trend logging, reports and so on.

2 Auto/Manual

Controls whether the routine is operational or not

3 Label

Eight character descriptor, a unique ID used to refer to this particular item throughout the system: programming, trend logs and so on.

4 Output

The state of the routine (ON/OFF)

5 Holiday 1

When point entered here is "On", the day will be the eighth day of the week.

6 State1

The state of Holiday1 (ON/OFF).

7 Holiday2

When point entered here is "On", the day will be the ninth day of the week. Note that in the case of both Holidays being on, Holiday1 is regarded.

8 State2

The state of Holiday2 (ON/OFF).

 To program the times for an individual routine press the "Ins" key while highlighting the Weekly Routine to be set up. The 9 day week will be displayed with the standard work menu operations. The desired start and stop times are then simply entered in the appropriate fields.

Note: **The times in Weekly Routines can be accessed from Control Basic using the WR- ON and WR- OFF functions. This is useful for programs that need to look ahead and see when an event is to start.**

The times specified must be in increasing order throughout each day. If no time is in a day column the Weekly Routine will maintain its current value.

6.8 Holidays

Hot Key: Alt-H

KEYWORD: AR
Usage：AR1,AR2,AR3...
How to show: when you use one of these items , the label of the item will be shown in the place where you use

Holidays are days for special action throughout the calendar year.

Sample Holidays work screen:

FULL LABEL

A unique twenty character descriptor used to identify the item throughout the system. Used for programming, trend logging, reports and so on.

Auto/Manual

Auto/Manual

Controls whether the routine is operational or not.

Value

Value

The state of the routine (ON/OFF).

Label

Label

Eight character descriptor, a unique ID used to refer to this particular item throughout the system: programming, trend logs and so on.

To program the times for an individual routine press the "INS" key while highlighting that point. The year is then displayed in 4 calendar format pages (use PgUp and PgDn keys to change page) with months available on each page. Work menu operations are again available. To set a day as a holiday move the cursor to the desired day and press "ENTER".

The Annual Routines defined here are the special event over rides which appeared in the previous section which described set ting up the ON and OFF times for Weekly Routines (repeated below).

When the annual over rides are defined, they can be entered into the "OR1" and "OR2" fields of the Weekly Routine setup area shown above, in order to let the Annual Routines override the weekly routines. In the ex ample above, we have entered "AR-1" or "Annual Routine 1" to override the Weekly Routine defined by "WR-1" or Weekly Routine 1

6.9 Trend Logs

Monitors are used to collect historical data for any point defined in the system regardless of their nature (either Analog or Digital). A base T3000 system can handle up to 16 separate monitors. The number of monitors avail able is dependent on the amount of memory that your system has. With enough memory the system can easily handle up to 128 monitors. Each of these separate monitors can trend up to 14 points.

1 **Label**

Label

An 8 character descriptor.

2 **Interval**

Interval

The time between samples in hours : minutes : seconds

3 **Status**

Status

"ON" indicates the monitor is running, "OFF" indicates that it is stopped.

4 **Input**

Input

The name of the points to be included on the monitor. The TAB key is used to move the cursor to this section. The TAB or ESC

key is used to return to the main monitor work screen. Up to 14 points can be monitored here per monitor. Of these 14 points

only 5 may be digital, i.e. On/Off or Start/Stop. If more than 5 Digital points are entered onto the monitor than a warning

flashes on the screen to indicate that you have entered too many Digital points. If this hap pens the Digital points over 5 will be deleted from the monitor.

An Ana log Monitor can also be commanded to sample from Control Basic by set ting it to ON or OFF (i.e. amon1 = ON). When this occurs the monitor samples all its in puts and restarts the interval time. In using this feature it is best to stop the monitor (s = 0), and control all the sampling from Control Basic.

To access monitor data and view the graph, press the "INS" key while highlighting the desired monitor.
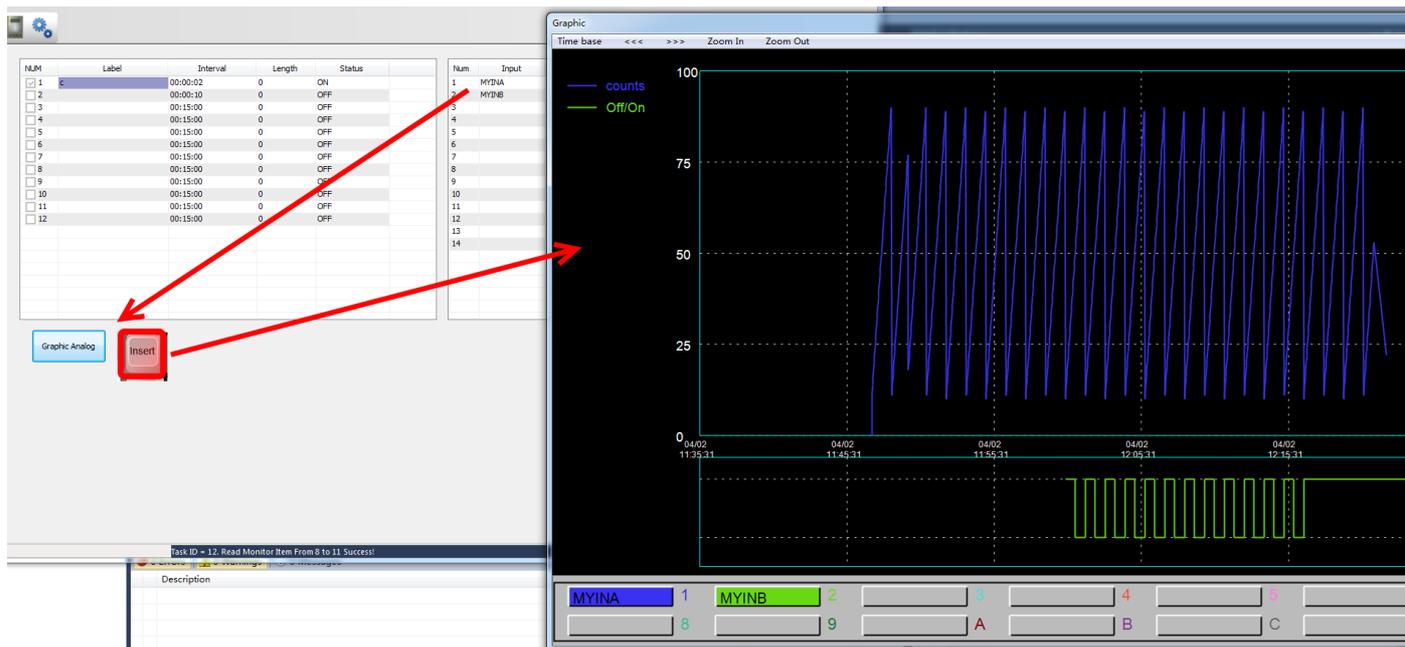
Graph Display work options:

Time base   The period of time displayed on a full screen. This time base can be altered to change the graphs perspective. For example the Time Base can be changed from hours to minutes

< < <   The graph can be shifted 1/2 screen left at the cur rent zoom level by clicking on the "<-" symbol at the top of the screen, or use the corresponding keyboard key.

> > >   The graph can be shifted 1/2 screen right at the current zoom level by clicking on the "->" symbol at the top of the screen, or use the corresponding keyboard key

Zoom In   While in the graphing mode, you can "Zoom In" on the data to view the data from a closer perspective. The Zoom range must first be set below.

Zoom Out   "Zoom Out" resets the monitor back one zoom level. This can only be used af ter Zoom In has been used to re store the graph to its original display mode.

6.10 ALARM LOG

NUM	Panel	Message	Time	Acknowlege	Res	Delete	
1							
2							
3							
4							
5							
6							
7							
8							
9							
10							
11							
12							
13							
14							
15							
16							

Alarms announce problem conditions in the system. They are generated by the ALARM and DALARM statements in Programs. The ALARM statement generates analog alarms and the DA LARM statement generates an alarm only if the condition persists for more than a defined period of time. These statements allow for a custom alarm message which can contain a maximum of 69 characters. When an alarm occurs, the alarm message can be sent to the printer along with the panel number from which the alarm originated, as well as the date and time through a control basic program. A red alarm box will blink at the top right of the screen, and will be removed only until all the alarms are deleted.

The event of an alarm restoring (going out of alarm condition) or being acknowledged is printed with the time of their occurrence. When an alarm is both acknowledged and restored it disappears from the system. The "Alarm Log" selection from the Data Menu shows all the alarms currently in the system with the corresponding panel (Pan). It is from this selection that alarms are acknowledged (ACK), restored (RES), or deleted (DEL). The time at which an alarm occurred can be seen right next to every alarm message.

To acknowledge an alarm, go to the ALARM LOG and move to the "Ack" column and press ENTER. Restoring. And deleting an alarm is done in the same manner. Alarms can be cleared from the system at the "DEL" column.

If there is any alarm in the system that is neither acknowledged or restored, the beeper in the PC is activated. The beeping can be stopped by acknowledging the alarm or by pressing Ctrl-A. Alarms can be directed to specific panels with the ALARM- AT statement. The default is to alarm all panels.

For information on the programming of ALARM and DALARM statements see Chapter 10.

7.1.1 Set your device IP address
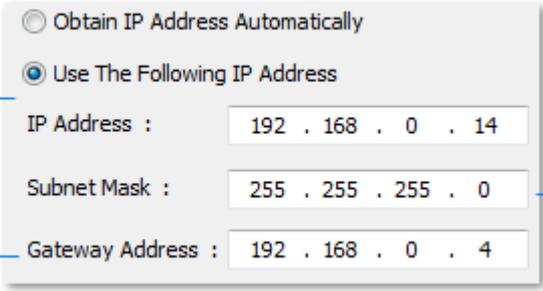
7.1.3 TIME & DATE

The time and date is set through this selection. This is the only time in the entire system. It is sent to all panels and also sets the internal clock in the personal computer.

You are first prompted to reset the time in hours: minutes: seconds - i.e. 14:32:34.

Next, you are prompted to reset the date by month-day-year - i.e. 10-29-95.

7.1.4 Device information
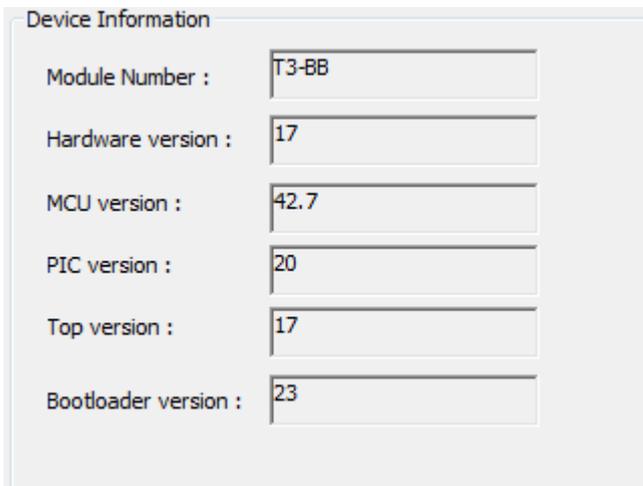
Device Information

Module Number :	T3-BB
Hardware version :	17
MCU version :	42.7
PIC version :	20
Top version :	17
Bootloader version :	23

7.1.5 PANEL ID
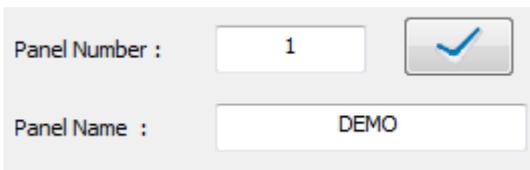
Panel Number : 1

Panel Name : DEMO
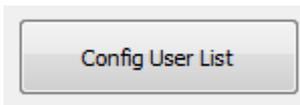
7.2 USER LIST

Config User List
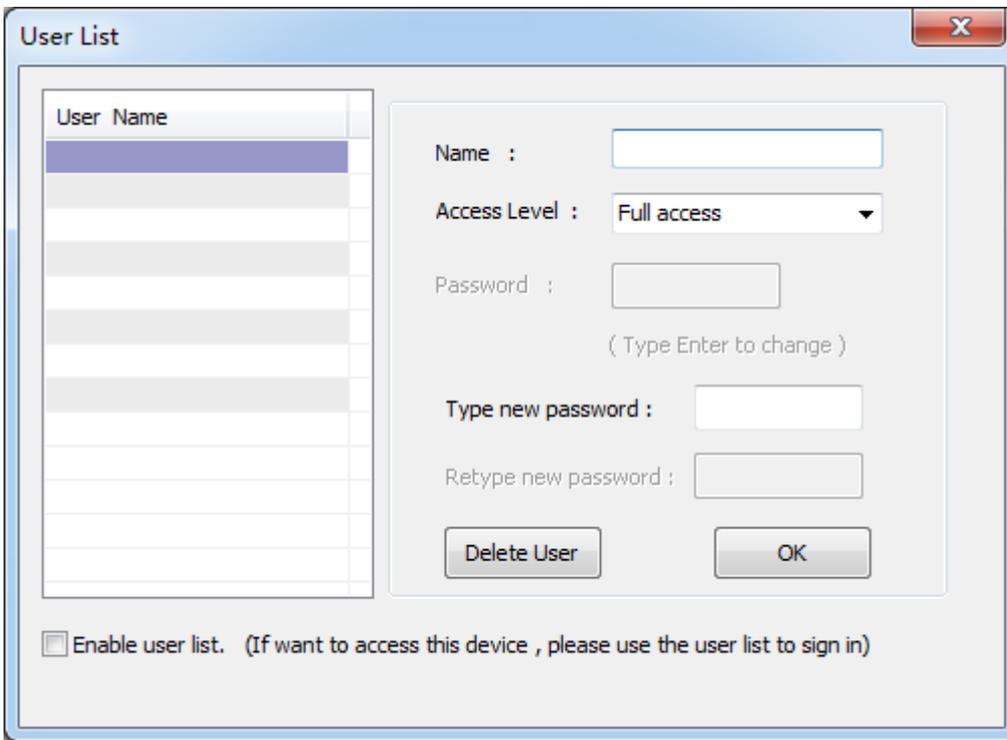
User List provides the means for system security and individual access to the system. Each user has to Sign- On with his name and give his pass word to gain access to the system. There is space to allow for 10 different users and passwords (8 for the Mini-Panel).

User List setup fields
a) Name . . . . . .   An eight character name of the user. First step for gaining system access when Signing-On.

b) Access level . . . . The system has 4 different access levels as follows:
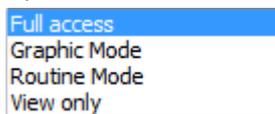
c) Default panel . . .   The default panel the user will see once log-in the system.

d) Default screen . .   The screen, the above user, first sees when signing on to the system. If the "Default screen" field is 0, then the default Work Menu is displayed after signing on. If the "Default screen" field is nonzero, that screen becomes the Main Menu for that particular operator. This can be either a text or graphic based display.

e) Password . . . .   An eight character name of the user. Second step for gaining system access when Signing- On. If all is correct, then that user is logged on to the system according to the above parameters.

When entering your "Password" the characters entered are not echoed to the screen. The system first asks you to "Type new Pass word" and then to "Retype new Pass word". If both passwords are the same then you will be allowed to exit the User List. The password will then have entered and retyped again correctly before you will be granted access to leave the User List. **Ensure that the letters entered are CAPITAL letters.**

8.3   Dyndns config

8.4 Device Serial Port Configuration

the description of these comports :
COM0:For the common UART ,Baud rate is 9600 or 19200
COM1:For the communication of zigbee
COM2:For the high speed UART ,Baud rate is less than 961200. If you want to use
MSTP, you must select the COM port .

IP Address

<TODO>: Insert description text here...

2 Subnet Mask

Subnet Mask : `255 . 255 . 255 . 0`

<TODO>: Insert description text here...

3 Gateway Address

Gateway Address : `192 . 168 . 0 . 4`

<TODO>: Insert description text here...

8 Control Basic Programming

Control Basic is the language used in the T3000 Building Automation System to implement the user defined logic of the controllers. These Control Basic programs live and run in the controllers to manage the environment in the building. The basic language has been around a long time so it may sound a bit 'dated' to be using this but its easy for newcomers to learn and there are many extensions on the root language to make the programming of controls applications easy. The programs read like a sequence of operations, a couple simple examples follow:

Here's a quick program to turn a fan on and off based on a schedule:
IF SCHEDULE1 THEN START FAN101 ELSE STOP FAN101

And here's a simple program to open an on/off type valve to heat a room for example.
IF TEMP101 < SETPOINT THEN OPEN VALVE101
IF TEMP101 > SETPOINT + 0.5 THEN CLOSE VALVE101

If you have a modulating valve things are also easy using the built in PID objects:
VALVE101 = PID1

The English like structure makes it easy for non programmers to follow along and debug. The following chapter describes the Control Basic language along with some examples.

8.1   THE CONTROL BASIC EDITOR

To get to the program editor, hit the Hot-Key combination **ALT-P,** or maneuver in the menu system to CONTROL, PROGRAMS and the table of programs appears as shown below:

This table lists all the programs in a particular T3000 panel, the size of each program, and a 20 and 8 character description of each program, similar to the naming conventions of INPUTS and OUTPUTS. To see a particular program, use the arrow keys to maneuver the cursor onto one of the records, and press the "INS" key. A

full screen editor should pop- up and depending on what is in the particular program, you should see a listing of that program, or a blank screen if there is no example file loaded.

To enter a line, type a line number followed by a simple "Statement" as shown:

10 START OUT1

Program lines are checked for errors when the "F2" command is given to send the current program. The editor screen has a Help Bar along the top and bottom row of the screen to remind you of the special function keys. When you hit F2, the current program is checked for errors. If the check was successful, the T3000 will begin executing the current program. If errors are found, messages will appear in the Error Window at the bottom of the screen, and the T3000 continues executing whatever program was last successfully sent.

Cursor movement within the editor is accomplished using the arrow keys on the key pad. If the program is longer than a single screen, the program will scroll in the editor window.

The special function keys which are active during the editing of basic programs are as follows:

"Ins"	The Insert key which causes a pop- up window to appear showing detailed information on any INPUT, OUTPUT or VARABLE in the program. This is very useful for debugging programs. The pop-up window can be used to toggle the item On or Off, or to manually set the outdoor air temperature to 35 Deg C for example
"Del"	The Delete key deletes the character at the current cursor position.
"End"	The End key moves the cursor to the end of the line in which it is in.
"Esc"	Exits the current editing session, prompting you to send the program if changes have been made.
"PgUp"	The PageUp keystroke scrolls the editor window one screen upwards.
"PgDn"	The PageDown keystroke scrolls the editor window one screen downwards.
CTRL-L	Highlights the current line for copying or deleting.
CTRL-U	Un-selects the currently selected block.
CTRL-D	Deletes the currently selected block.
CTRL-Y	Deletes a line at the current cursor position, regardless of whether the line is highlighted or not.

CTRL-O	Undoes the last block or line delete.
CTRL-C	Copies the currently highlighted block.
CTRL-F	Searches the current program for a specified string. The default search string is taken from the current cursor position, or a new one can be typed in by the user.
CTRL-N	Searches for the next occurrence of the same string previously searched for the FIND command (CTRL-F).
CTRL-R	Searches for all occurrences of a specified string and re places them with a second string. The replace command does not prompt on each substitution. The default search string is taken from the current cursor position, but a new one can be typed in by the user.
CTRL-E	Re-numbers the entire program in increments of 10. Use this command to make space between two lines for a new line.
F2	Checks the current program for errors and sends it to be executed if there are no errors. If there are errors, the old program continues executing (if there was one) and the user is able to continue editing.
F3	Clears the current editor buffer. If there was a program executing in the T3000, that program is cleared as well. The user is prompted before al lowing the current program to be cleared.
F4	Prints the current program on the printer.
F5	Loads a BASIC file from the hard disk. Works in conjunction with the F6 command to allow the user to transfer Basic programs from old projects, other panels , and different versions by using the Save and Load commands. Basic files are saved in text format with the .BAS ex tension. The directory used by the file Operations is determined in the CONFIGURATION, SETUP, FILEPATH screen.
F6	The Save command allows the user to save Basic files in a text format on the hard disk. Operates in conjunction with the LOAD command above.

10.2 CONTROL BASIC EXAMPLE

Control Basic program as used by the T3000 system is essentially the same as any other Basic program, a sample of which is shown below:

```
10 REM *** START / STOP ***

20  DA LARM AHU7- FRZ , 60 , AHU7 FREEZE STAT ALARM 30 DA LARM AHU7- FS , 300 , AHU7 FIL TER PRESS HIGH

40 REM **** DE CIDE IF HEAT ING MODE IN FORCE ****

50 IF AHU7- HV 1 THEN AHU7- HM = 1 ELSE AHU7- HM = 0

60 IF DM-7 OR PC-7 OR WU-7 THEN START AHU7-S/S ELSE STOP AHU7-S/S 70 REM *** MIXED AIR DAMPER ***

80 REM ** FREE COOL MODE **

90 IF OAT AHU7- RTS - 2.5 THEN START AHU7- FCM

100 IF OAT AHU7- RTS - 1 THEN STOP AHU7- FCM 110 AHU7- MAD = MAX( AHU7- AIR , 1- CON1 )

120 IF NOT AHU7- FCM THEN AHU7- MAD = AHU7- AIR

130 IF AHU7- HV 1 THEN AHU7- MAD = AHU7- AIR

140 IF OAT 10 THEN AHU7- MAD = MIN( AHU7- MAD , TIME- ON( AHU7-S ) )

150 IF NOT AHU7-S THEN AHU7- MAD = 0

160 AHU7- DSP = MAX( 12 , MIN( OCCT-7 , 24 ) )

170 IF NOT AHU7-S/S THEN AHU7- DSP = AHU7- MAT

180 REM *** HEAT ING CON TROL ***

190 AHU7- HV = 1- CON2

200 IF AHU7- SAT 30 THEN AHU7- HV = 0

210 IF AHU7- RTS OCCT-7 THEN AHU7- HV = 0

220 IF NOT AHU7-S THEN AHU7- HV = 0

230 IF AHU7- FRZ THEN AHU7- HV = 100

240 AHU7- HSP = AHU7- DSP - 0.5

250 IF NOT AHU7-S/S THEN AHU7- HSP = AHU7- SAT

260 REM *** COOL ING CON TROL ***

270 AHU7- CSP = AHU7- DSP + 0.5

280 IF NOT AHU7-S/S THEN AHU7- CSP = AHU7- SAT

290 AHU7- CV = 1- CON3

295 IF AHU7- RTS OCCT-7 + 1 THEN GOTO 330
```

300 IF AHU7- FCM AND AHU7- MAD 90 THEN AHU7- CV = 0

330 IF AHU7- SAT 10 THEN AHU7- CV = 0

340 IF NOT AHU7-S THEN AHU7- CV = 0

Control Basic programs have line numbers for each new program statement. Each statement, command, and operator is separated from the next item by a space. Program execution starts at the top-most line and barring any GOTO

statements in the program, follows on through each line in the program.

10.3   PROGRAM FLOW

There are some special mechanisms employed by the T3000 operating system to protect the pumps and fans, and control them efficiently. Understanding the program flow in a T3000 system will help understand how to better program a real time system and will also make your programs easier to follow.

For the most part, program logic in the T3000 panel follows through from the top of the program to the last line in sequential order. Since Basic supports the GOTO command, but program execution can actually take any path that you desire, but GOTO statements can cause a program to be difficult to follow.

Although an OUTPUT can appear many times in a program, with some lines commanding the output to go on, and other lines telling the output to go off, the output is not actually switched until all the lines in a particular panel are evaluated. This mechanism gives lines appearing at the end of the program veto power over earlier lines. Safety program statements such as *emergency shutdown* appear at the end of a program to be sure the shut down logic over rides all other possible logic. A useful consequence of this mechanism is that a particular output can appear many times in a program, breaking up the logic amongst several simple program statements.

When the last line is processed, the T3000 system automatically begins execution from the start of the program again. There is no need for a RETURN statement as in other Basic dialects as the return is implied in a T3000 program. Each program can be thought of as a treadmill on which the T3000 keeps on running continuously churning through the code and adjusting the outputs based on the program logic. The example below shows the logic flow for the T3000 system:

Good programming practice will dictate that a particular output is programmed in only one program. If the program for a particular system is so large that it must be broken up, the program would be broken along logical boundaries so that each output only appears in one program. Future trouble shooting sessions will be simplified if each output is programmed in only one program.

10.4   CONTROL BASIC COMPONENTS

```
PRO GRAM #1
    10 REM   : ********** SF1 **********
    20 IF OAT > 20 THEN START COOL MODE
    30 IF OAT < 19 THEN STOP  COOL
MODE PRO GRAM #2
PRO GRAM #3
..
PRO GRAM #32
SET OU PUTS TO LAST COM MANDED STATE
```
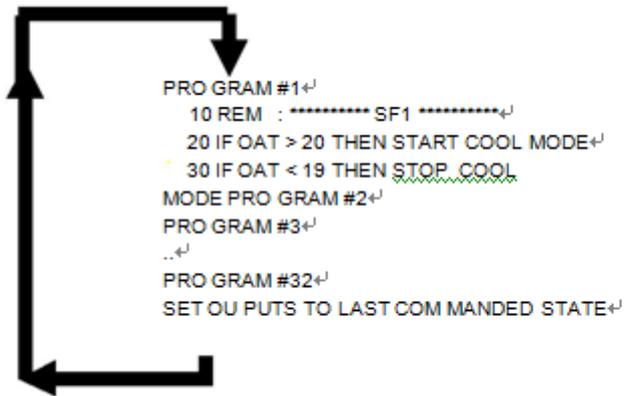
This section lists the components used in Basic program. Following this list is a detailed list of all the program commands and functions. Most of the items are explained with a short ex ample on how to use the command.

The following is a list of the fundamental components of a Basic program:

REAL NUMBERS . .   Ordinary numbers.

DAYS . . . .   The 7 days of the week.

DATES . . . . . .   The calendar date.

TIMES . . . .   Time in HH:MM:SS format.

VARIABLES . . . .   The T3000 soft ware sup ports a large table of Variables or memory locations which can be used for setpoints, accumulators, modes and other useful bits of information pertinent to the particular application.

OPERATORS   . . .   Mathematical operators such as the greater than(**>**), less than (**<**), and equals (**=**) symbols.

COMMANDS . . . .   Specifies an action word in programming language such as the START command in the following example:

IF IN1 > 20 THEN START OUT1

FUNCTIONS . . . .   Functions are built in routines which return a value. For instance MAX(T1, T2, T3...) in the example shown will return the larger of the items enclosed in brackets.

VAR1 = MAX(IN1 , IN2 , IN3)

EXPRESSIONS. . . . Notation that has a value. Usually a combination of above elements.

The complete list of Control Basic components:

STATEMENT	TYPE	DESCRIPTONSUMMARY
NUMBERS	DATA	Real numbers are handled by the T3000 in full floating point for mat
DAYS	DATA	Days are evaluated as an integer from 0 for sun day to 6 for saturday
DATES	DATA	Dates are handled internally by the T3000 as a number from 1 to 365
TIMES	DATA	Times are handled internally as hours, but can be typed HH:MM:SS in a program

VARIABLES	DATA	Variables are handled internally as real numbers with full floating point notation
NOT	OPERATOR	Evaluates to TRUE if the opposite is true. IF NOT HOT evaluates to true when COLD
^	OPERATOR	Exponent, raises a number to a power as in 2^2 = 4
* / \ MOD	OPERATOR	Multiplication, division, integer portion of a division, remained of a division
+ -	OPERATOR	Addition and subtraction
< > ... etc	OPERATOR	Less than, greater than, and other combinations.
AND	LOGIC	Logical AND, both conditions must be TRUE for the expression to evaluate to TRUE
OR	LOGIC	Logical OR, if either expression is TRUE the result is TRUE
XOR	LOGIC	Logical Exclusive OR, evalutates to TRUE if the two expressions are different
ABS	FUNCTION	Absolute Value, returns the positive equivalent of a negative or positive number
ALARM	COMMAND	Causes an alarm, has built in hysterisis to avoid multiple alarms from the same event
ALARM-AT	COMMAND	Selects which panel will annunciate alarms, default is ALL panels
AVG	FUNCTION	Average of all numbers enclosed in the brackets following this command
CALL	COMMAND	Calls an other program which functions as a sub routine, takes arguments
CLEAR	COMMAND	Clears all the local variables in a given program
CLOSE	COMMAND	Sets a variable or out put to 0, same as STOP but is more intuitive in some cases
CONPROP	FUNCTION	Allows a Basic program to up date the proportional term in a particular PID controller
CONRATE	FUNCTION	Allows a Basic program to up date the derivative term in a particular PID controller
CONRESET	FUNCTION	Allows a Basic program to up date the integral term in a particular PID controller
DALARM	COMMAND	Delayed alarm, allows generation of alarms if a condition is true for a certain time
DECLARE	COMMAND	First line of a sub routine to accept the list of arguments from a CALL statement
DISABLE	COMMAND	Same as STOP or CLOSE, sets an item to 0
DOM	FUNCTION	Day of month, returns the day of the month, starting with 1 on the first of the month
DOW	FUNCTION	Day of week, returns a number from 0 to 6 starting with Sun day as zero
DOY	FUNCTION	Day of the year, starting with Jan1 as 1
ENABLE	COMMAND	Same as START or OPEN, sets the value of an item to 1.
END	COMMAND	Ends execution of a program, T3000 assumes and end at the end of each program
FOR	COMMAND	Loop control statement, FOR A =1 TO 10 ....PRO GRAM LINES.....NEXT A
GOSUB	COMMAND	Branches to sub routine, executes program till next RE TURN statement, and loops back
GOTO	COMMAND	Unconditionally branches program execution to an other line
HANGUP	COMMAND	Ends a phone call in progress
IF	COMMAND	Logical decision, IF HEAT MODE THEN START OUT1
IF+	COMMAND	Executes the program line only once when the condition goes from FALSE to TRUE
IF-	COMMAND	Opposite of IF+, executes the line only when a condition be comes FALSE
INT	FUNCTION	Returns the integer part of an expression
INTERVAL	FUNCTION	IF INTERVAL(0:01:00) executes the following programming once every minute
LET	COMMAND	Assignment operator, LET B = 20, has the same effect as B = 20
LN	FUNCTION	Natural logarithm of the argument
LN-1	FUNCTION	In verse logarithm of the argument
MAX	FUNCTION	Maximum of a list of arguments enclosed in brackets following the statement
MIN	FUNCTION	Minimum of a list of arguments enclosed in brackets following the statement
NEXT	COMMAND	Used in conjunction with the FOR statement to increment the loop counter
ON	COMMAND	ON VAR1 GOTO 100 , 200 , 300 , controls program path depending on VAR1

ON-ALARM	COMMAND	Evaluates to TRUE when ever there is an alarm in the system.
ON-ERROR	COMMAND	Used with PHONE, REMOTE- GET, REMOTE- SET, and RUN- SYSTEM
OPEN	COMMAND	Sets an item to 1, OPEN OUT1 is the same as START OUT1
PHONE	COMMAND	Phones a number, allows T3000 to dial with a modem for numeric paging
POWER -LOSS	FUNCTION	Looks for loss of power, 1 the first time the function is evaluated, o afterwards
PRINT	COMMAND	Print strings at a printer connected to the panel
PRINT- AT	COMMAND	Print strings at a printer connected to an other panel on the net work
REM	COMMAND	Remark, used to add comments to a program
REMOTE- GET	COMMAND	10 REMOTE- GET VAR1 = 2IN1, gets a reading from an other site via modem
REMOTE-SET	COMMAND	Sets a variable at a remote site via modem.
RE TURN	COMMAND	Causes program execution to return from a sub routine
RUN-MACRO	COMMAND	Causes a macro command or series of key strokes to be run
SCANS	FUNCTION	The number of scans a panel is per forming in one second
SET-PRINTER	COMMAND	Tells a T3000 system there is a printer at a particular panel
SQR	FUNCTION	Returns the square root of the argument
START	COMMAND	Sets the value of an items to 1, same as OUT1 = 1 or OPEN OUT1
STATUS	FUNCTION	Status of a panel on the net work, 0 = No Comm, 1 =OK, 2 = no program
STOP	COMMAND	Sets the value of an items to 0, same as OUT1 = 0 or CLOSE OUT1
TBL	FUNCTION	Looks up a value in a table
TIME	FUNCTION	Finds the cur rent time
TIME-OFF	FUNCTION	Finds the time variable has been false
TIME-ON	FUNCTION	Finds the time variable has been true
WAIT	COMMAND	Pauses program execution
WR-ON	FUNCTION	Finds when a weekly routine is set to come on
WR-OFF	FUNCTION	Finds when a weekly routine is set to go off
UNACK	FUNCTION	Determines if there are unacknowledged alarms in system
USER-A	FUNCTION	Finds the user logged on to the local port
USER-B	FUNCTION	Finds the user logged on to the remote port

The following sections list the components of a Basic program, along with some examples and descriptions on how they operate.

In this section the various components are listed alphabetically. The explanation of each term consists of 3 parts:

1. Purpose . . . . . Explains when the function is used.

2. Format . . . . . Shows how the function is used.

3. Example . . . . Gives a practical application.

9 Programming a controller

This section explains the programming steps to setup a panel for a controls application. The steps are presented here in a somewhat logical order but the system is flexible, you can complete the sections you have information on as the data comes in and finish off the other parts later. It is assumed that at this point, the T3000 has been setup as per instructions in Chapters 3 and 8.

1. The Input and Output List

List all the inputs and outputs for the system to be controlled. A spreadsheet is generally available from the original project description. If you are starting from scratch, list the equipment and for each of these work out the list of inputs and outputs. Be sure to check the sequence of operations for the system, there's often wording in the text which doesn't match up well with the drawings, those few words can be expensive words which require extra sensors and i/o for your project.

Assuming you are doing your own list, its OK to proceed as information comes in, you can always revise the list as the project proceeds. The points don't have to be in any particular sequential order but its definitely best to keep related i/o near others on the same equipment. And certainly its best to keep i/o from one air handler for example, on one controller so that if the network goes down the equipment can still operate in stand alone mode. Software can connect any i/o to other i/o in the system but its best to keep the list in logical order.

-Once the list is together its time to assign some names to the i/o for entering into the system. In the T3000 Building Automation System we can we can assign each input or output with two names, a long one which is 20 characters for showing in reports and user screens, and a shorter one which is 8 characters to be used in the programming.

- Each input and output needs to be assigned to a physical input or output point on the controller. For inputs this is easy, any input can be jumper configured to match up with most any sensor, switch, transducer or meter. For the outputs, you need to assign the analog items to one of the analog outputs. On the binary outputs to enable fans and pumps the job is a little easier, assign each of these to an output relay on the controller.

TIP: If you need more binary outputs, a spare analog output can be converted to a binary output with an external 12VC coil relay, the analog output will drive the relay to gain one extra really from an analog output. The reverse is not true though, relays are fixed as binary outputs.

- The input list should show the sensor or transducer type and jumper settings.

- For valves and dampers especially, there are many arrangements for the i/o depending on the equipment and project specs so its best to have a data sheet for the exact item. The valves will be specified as normally open or closed, gather that information from the mechanical drawings and sequence description. The output signal can be floating control, modulating or on/off, some have feedback signal as well. All this info is in the data sheets for the actuator and is important to have on hand when laying out the system.

2. Configure "System" Components

Run the T3000 software and Sign-On with your Name and Password. Setup the T3000 as per instructions in Chapters 3 and 8.

3. Program Inputs

"Inputs" are the sensors and devices which are connected into the T3000 panel so that the T3000 can be programmed to act upon. For more information on In puts see section 6.3.

Select "Inputs" from the CONTROL menu, press ALT-I, or type INS from the command line.

Enter in a description that will identify each point. Each point must be unique or the system will not accept the description. When entering the 20 character "Description" as it is known, try to be as descriptive as possible. You should also develop a consistent nomenclature for the various types of control points such as "RTS" in front of all room temperature sensors. Using this nomenclature, the room sensor in office 102 would be named "RTS OFFICE102".

Also enter the "Label" which is the eight character short hand name for the input point which will be used in the Control Basic programs. Again, each LABEL must be unique or the system will not accept the name. By using the above nomenclature, the label for the same office temperature sensor would be "RTS102 ".

Other Examples:

Input Point		Description	Label
Air Handling Unit 1	Supply Air Temperature	AHU1 SUPPLY AIR TEMP	AHU1.SAT
Air Handling Unit 1	Supply Fan Status	AHU1 SUPPLY FAN STAT	AHU1.S
Pump # 1 Radiation Loop Start/ Stop	P1 RAD LOOP S/S	P1.S/S	

Select the range:

To set the range high light the field, press ENTER, and enter the desired range number. If no standard range will work with the sensor, see the section of this manual describing the construction of a "TABLE".

The "D" field is a simple check mark field to verify the correct operation of each sensor. Toggle the field back to normal, check for correct readings, and enter calibration values if required.

Your changes are sent to the panel automatically if you leave the current screen, but to see the effect of current changes, you must Send and then Load the Input changes.

Input	Panel	Full Label	Auto/Manual	Value	Units	Range	Calibration	Sign	Filter	Status	Jumper	Label
IN1	1	SUPPLY AIR TEMP	Auto	-40.00	Deg.F	10K -40 to 250	0.0	-	5	Open	Thermistor Dry...	AHU1SAT
IN2	1	RETURN AIR TEMP	Auto	248.00	Deg.F	10K -40 to 250	0.0	-	5	Shorted	Thermistor Dry...	AHU1RAT
IN3	1	MIXED AIR TEMP	Auto	-40.00	Deg.F	10K -40 to 250	0.0	-	5	Open	Thermistor Dry...	AHUMAT
IN4	1	ROOM TEMP	Auto	-40.00	Deg.F	10K -40 to 250	0.0	-	5	Open	Thermistor Dry...	AHU1RT
IN5	1	FAN STATUS	Auto	On		Off/On			5	Normal	Thermistor Dry...	AHU1FS
IN6	1	OUTSIDE AIR TEMP	Auto	-40.00	Deg.F	10K -40 to 250	0.0	-	5	Open	Thermistor Dry...	OAT
IN7	1	IN 7	Auto	2.99		Unused	0.0	-	5	Normal	Thermistor Dry...	IN7
IN8	1	IN 8	Auto	2.99		Unused	0.0	-	5	Normal	Thermistor Dry...	IN8
IN9	1	IN 9	Auto	2.99		Unused	0.0	-	5	Normal	Thermistor Dry...	IN9
IN10	1	IN 10	Auto	2.99		Unused	0.0	-	5	Normal	Thermistor Dry...	IN10
IN11	1	IN 11	Auto	2.99		Unused	0.0	-	5	Normal	Thermistor Dry...	IN11
IN12	1	IN 12	Auto	2.99		Unused	0.0	-	5	Normal	Thermistor Dry...	IN12
IN13	1	IN 13	Auto	2.99		Unused	0.0	-	5	Normal	Thermistor Dry...	IN13
IN14	1	IN 14	Auto	2.99		Unused	0.0	-	5	Normal	Thermistor Dry...	IN14
IN15	1	IN 15	Auto	2.99		Unused	0.0	-	5	Normal	Thermistor Dry...	IN15
IN16	1	IN 16	Auto	2.99		Unused	0.0	-	5	Normal	Thermistor Dry...	IN16
IN17	1	IN 17	Auto	2.99		Unused	0.0	-	5	Normal	Thermistor Dry...	IN17
IN18	1	IN 18	Auto	0.00		Unused	0.0	-	5	Normal	Thermistor Dry...	IN18
IN19	1	IN 19	Auto	2.99		Unused	0.0	-	5	Normal	Thermistor Dry...	IN19
IN20	1	IN 20	Auto	2.99		Unused	0.0	-	5	Normal	Thermistor Dry...	IN20
IN21	1	IN 21	Auto	3.00		Unused	0.0	-	5	Normal	Thermistor Dry...	IN21
IN22	1	IN 22	Auto	3.00		Unused	0.0	-	5	Normal	Thermistor Dry...	IN22
IN23	1	IN 23	Auto	3.00		Unused	0.0	-	5	Normal	Thermistor Dry...	IN23
IN24	1	IN 24	Auto	3.00		Unused	0.0	-	5	Normal	Thermistor Dry...	IN24
IN25	1	IN 25	Auto	3.00		Unused	0.0	-	5	Normal	Thermistor Dry...	IN25
IN26	1	IN 26	Auto	3.00		Unused	0.0	-	5	Normal	Thermistor Dry...	IN26
IN27	1	IN 27	Auto	3.00		Unused	0.0	-	5	Normal	Thermistor Dry...	IN27
IN28	1	IN 28	Auto	3.00		Unused	0.0	-	5	Normal	Thermistor Dry...	IN28

4. Program Outputs

"Outputs" are the output signals from the T3000 sent to the fans, pumps, boilers, etc. which are connected to the T3000. Outputs can be either analog or digital. The software setting determines which mode the output will work with. The engineering units for each output can be configured to match the actual units normally associated with the device being controlled (i.e. Stop/Start , Close/Open, or % Open, % Closed, CFM). For more information on "Out-puts" see section 5.2.

To get to the outputs screen, type ALT-O or Select "Outputs" from the CONTROL menu. Alternatively, you can type OUTS from the command line.

Just as was described for the inputs, each OUT PUT re quires a 20 character description and an 8 character label. The names should follow a naming nomenclature similar to that used for the inputs, and each name must be unique.

To set the range highlight the field, press ENTER, and enter the desired range.

Enter in the Low and High volt ages for any analog devices such as valves or dampers which operate over 2-10 VDC range rather than the normal 0-10V range.

Enter starts delays as required. This is used for digital out puts only and de lays the start of a device in minutes: seconds up to a maximum delay of 99 minutes 50 seconds. A value entered in this field superimposes a lag on all other control actions such as weekly routines or basic programs which govern this point.

If you want to increase the security level on the output, enter that level in the "S" field.

Output	Panel	Full Label	Auto/Manual	HOA Switch	Value	Units	Range	PWM Period	Status	Label
OUT1	1	AHU1 FAN	Auto	AUTO	Off		Off/On	0	OK	AHU1FAN
OUT2	1	AHU1 COOLING	Auto	AUTO	Off		Off/On	0	-	AHU1COOL
OUT3	1	AHU1 HEATING	Auto	AUTO	Off		Off/On	0	OK	AHU1HEAT
OUT4	1	AHU1 MIX ᴸ?	Auto	AUTO	Off		Off/On	1	OK	
OUT5	1	┼	Manual	AUTO	0.00		Unused	32	OK	
OUT6	1		Auto	AUTO	0.00		Unused	0	OK	OUT6
OUT7	1	OUT7	Auto	AUTO	0.00		Unused	0	OK	OUT7
OUT8	1	OUT8	Auto	AUTO	0.00		Unused	0	OK	OUT8
OUT9	1	OUT9	Auto	AUTO	0.00		Unused	28	OK	OUT9
OUT10	1	E	Auto	AUTO	0.00		Unused	65	OK	?
OUT11	1		Auto	AUTO	0.00		Unused	0	OK	ACACAAA
OUT12	1	OUT12	Auto	AUTO	0.00		Unused	0	OK	OUT12
OUT13	1	OUT13	Auto	AUTO	0.00		Unused	0	OK	OUT13
OUT14	1	OUT14	Auto	AUTO	0.00		Unused	0	OK	OUT14
OUT15	1	OUT15	Manual	AUTO	0.00		Unused	168	OK	OUT1 ᴸ?
OUT16	1		Manual	AUTO	0.00		Unused	65	OK	
OUT17	1	CACACACACACAAA	Auto	AUTO	0.00		Unused	0	OK	OUT17
OUT18	1	OUT18	Auto	AUTO	0.00		Unused	0	OK	OUT18
OUT19	1	OUT19	Auto	AUTO	0.00		Unused	0	OK	OUT19
OUT20	1	OUT20	Auto	AUTO	0.00		Unused	0	OK	OUT20
OUT21	1	OUT21	Manual	AUTO	0.00		Unused	11	OK	g@
OUT22	1	?	Auto	AUTO	0.00		Unused	0	OK	
OUT23	1	OUT23	Auto	AUTO	0.00		Unused	0	OK	OUT23
OUT24	1	OUT24	Auto	AUTO	0.00		Unused	0	OK	OUT24
OUT25	1	OUT25	Auto	AUTO	0.00		Unused	0	-	OUT25
OUT26	1	OUT26	Auto	AUTO	0.00		Unused	0	-	OUT26
OUT27	1	6 €	Manual	AUTO	688.38	Unused	Out of range	25	-	
OUT28	1	.e••	Auto	AUTO	0.00		Unused	0	-	OUT28

Toggle all outputs to be decommissioned. Later, as you verify their correct operation toggle the field back to normal.

When satisfied with their operation, Send and then Load the Outputs.

5.   Define programming jobs based on out puts that you have to control

For example, a panel may have the following control jobs.

- Boiler Control

- Chiller Control

- Fan System 1 Control

Select "Programs" from the Control menu, or by typing PRGS from "Direct Access". Enter in the program description and label. Each point must be unique or the system will not accept the description. When entering the description try to be as descriptive as possible.

For example: Air Handling Unit 1 Control = AHU1 PROGRAM

Program	Full Label	Status	Auto/Manual	Size	Run Status	Label	
☑ 1	AHU1 PROGRAM	OFF	Auto	406	Normal		
☐ 2		OFF	Auto	64	Normal		
☐ 3		ON	Auto	29	Normal		
☐ 4		ON	Auto	0	Normal		
☐ 5		ON	Auto	0	Normal		
☐ 6		ON	Auto	0	Normal		
☐ 7		ON	Auto	0	Normal		
☐ 8		ON	Auto	0	Normal		
☐ 9		ON	Auto	0	Normal		
☐ 10		ON	Auto	0	Normal		
☐ 11		ON	Auto	0	Normal		
☐ 12		ON	Auto	0	Normal		
☐ 13		ON	Auto	0	Normal		
☐ 14		ON	Auto	0	Normal		
☐ 15		ON	Auto	0	Normal		
☐ 16		ON	Auto	0	Normal		

6. Write Individual Programs

Press the "Ins" key while highlighting the line of the de sired program, or using "Direct Access" type the program's label. As you write a program you will want to make use of 4 other point types:

- Variables - VARS

- Controllers - CONS

- Weekly Routines - WRS

- Annual Routines - ARS

Set up these points as the need for them arises. When finished writing your program, send the program by pressing "F2". This will first check the program for errors. If correct, it will send the program to the panel. If not, it will supply an error message complete with line numbers. For more information on Control Basic Programming see chapter 10.

7. Define Graphic Screens:

Graphic Screens are used to group points of all types into one logical unit for displaying on the screen. The elements of a group can be displayed over a graphic picture to augment the display of information.

Select "Graphic" from the Control menu. Enter in the name and label. Each label must be unique or the system will not accept the description. When entering the label try to be as descriptive as possible.

Graphic	Full Label	Label	Picture File	Refresh Rate	
☑ 1	CACACACACABM	AHU1	radke	15	
☐ 2				15	
☐ 3				15	
☐ 4				15	
☐ 5		slO?		1	
☐ 6			CACACAAA	0	
☐ 7				15	
☐ 8				15	
☐ 9				15	
☐ 10				101	
☐ 11	◄¶├?			15	
☐ 12				15	
☐ 13				15	
☐ 14				15	
☐ 15				15	
☐ 16		"u	?	1	

8. Design screens:

Press the "Ins" key while highlighting the line of the desired screen.

To display a point on the screen, move the cursor to the desired location and press the "Ins" key. A pop- up box will appear on the screen to let you specify the element to be added to the display. A second pop- up menu will appear which controls the colour and size of the element text, and several other functions associated with this element. Press the "Esc" key to exit pop- up menu. Continue to add points to display until completion. When satisfied with the display go to File and Save.

For a more detailed description on building Screens see section 6.1.

9. Define Monitors

There are two types of monitors: Ana log which trends analog points, and Digital which logs start and stop times of digital points. Both are displayed together on a single monitor from the Data pull-down menu.

Set up monitors by entering the monitor name and entering the points to be monitored. Enter the interval time between samples, and then Send. Sampling of these points will now take place. If for some reason you decide to change some aspect of a monitor, it will erase all previous data and start a new one once you send the changes to the panel. For more information on Monitors see section 7.1.

10. Save the Program

Using the "File" selection on the Program menu, "Save" the panel program. Steps 1 through 9 are sent to the panel each time that you do a "Send". By "Saving" the program, the program is saved in your PC and is available to be down loaded back into the panel if the panel crashes or loses its program.

Control Basic Programming

Numbers, Time and Date Handling

The following sections list the control basic programming language, it is suited for real time control applications with many advanced features built into the language directly which allows the programmer to create powerful programs with a small number of statements. Along with the description of the statements are some short examples for each.

The list of program statements and commands is listed alphabetically. Each item consists of 3 parts:

1.   Purpose . . . . . Explains when the function is used.

2.   Format . . . . . Shows how the function is used.

3.   Example . . . .   Gives a short example program

10.4.1 REAL NUMBERS

Real numbers between +/- 3.4*10^38 are recognized. The number can be entered in a number of different formats.

Examples: 0.00000123 , 7.879E-12 , 100

10.4.2 DAYS

Each of the 7 weekdays evaluates to a real number as shown below:

SUN = 0 MON = 1 TUE = 2 WED = 3 THU = 4 FRI = 5 SAT = 6

Example:10 IF DOW = 1 THEN OPEN SPRINKLER1
          20 IF TIME-ON (SPRINKLER1) > 0:05:00 THEN CLOSE SPRINKLER1

This example will open a sprinkler valve on Monday morning at 12:00:00 and turn it off at 12:05:00

10.4.3 DATES (JAN 1 to DEC 31)

[TBD] Dates can be referred to in programming statements like a real number which allow logic operations on calendar events for certain days of the year, AUG12 and NOV10 can be used in the program statements.

The month abbreviations are:
JAN, FEB, MAR, APR, MAY, JUN, JUL, AUG, SEP, OCT, NOV, DEC.

Note that February is allowed to have 29 days as happens during a leap year rather than the usual 28 days.

Example:10 IF DOY = AUG12 THEN STOP FAN

10.4.4 TIMES (0:0:0 to 23:59:59)

Times are evaluated to real numbers using the following formula:

time = hour*100 + minutes + seconds / 100

The time of 14:30:30 is returned as a real number of value = 1430.30

If the seconds are not specified, the time is interpreted as hours and minutes only, ie 12:30 returns a value of 1230

EXAMPLE: IF TIME = 13:00:05 THEN START PUMP1

See the TIME command for more examples of math using time

10.4.7 **COMMANDS AND FUNCTIONS**

As sown in the previous pages, a Basic program is comprised of Basic commands and functions. The commands and functions operate in conjunction with a mixture of mathematical operators, variables, days, dates, times and real numbers to accomplish a given control sequence.

COMMANDS

Every control basic line must have at least one command. If more than one command is on one line, they must be separated by semicolons.

FUNCTIONS

functions represent values that are dependent on 1 or more other values or some system constant such as time. They can be used in any statement where it calls for an expression.

In this section the Commands and Functions are listed alphabetically. The explanation of each term consists of 3 parts:

1.   Purpose . . . . . Explains when the function is used.

2.   Format . . . . . Shows how the function is used.

3. Example . . . . Gives a practical application.

10.4.5 MATHEMATICAL OPERATORS

Mathematical Operators are discussed below in their order of priority. Operators of higher priority are executed before operators of lower priority. Within the same group, operators have the same priority and are evaluated left to right. The order of execution can be modified by the use of parenthesis. The general form for the use of operators is:

expression operator expression

Note that NOT has a unique form as shown in the next section.

NOT

NOT is a unique operator in that it only acts on one value which must be a variable. (All other operators act on 2 expressions). NOT per forms a logical not operation on an expression. If the expression is 0 the result is 1. If the expression is non-zero the result is 0.

Example: 10 IF NOT OUT1 THEN STOP OUT2

20 IF NOT (OUT1 OR OUT2) THEN START OUT3

"^ " (T3000 only)

The ^ symbol is used to raise a value to a power. Press shift and 6 on the PC keyboard.

Example: 10 A = IN1 ^ VAR1

* , / , \ , MOD

These operators perform multiplication and division.

*   Multiplication.

/   Division.

| \ | The integer portion of a division | (ie: 13 \ 5 = 2). |
| MOD | The remainder of a division | (ie: 13 MOD 5 = 3). |

+ , -

These operators perform addition and subtraction.

< , > , <= , >= , <>

These operators per form comparisons. The result is 1 if the comparison is true, otherwise the result is 0.

 Result is true if
left expression is
LESS THAN right
expression.

Result is true if left expression is GREATER THAN right expression.

Result is true if left expression is LESS THAN OR EQUAL to right expression.

Result is true if left expression is GREATER THAN OR EQUAL to right expression.

Result is true if both expressions are EQUAL.

Result is true if expressions are NOT EQUAL.

10.4.6
LOGIC

_____

AND

AND per forms the logical "and" of the two expressions. The result is true (1) if both expressions are non-zero, otherwise the result is false (0).

OR

OR per forms the logical "or" of the two expressions. The result is true (1) if either or both expressions are non-zero, otherwise the result is false (0).

XOR

XOR performs the logical "exclusive or" of the two expressions. The result is true (1) if the two expressions are different, otherwise the result is false (0).

10.4.9 <u>**EXPRESSIONS**</u>

Expressions evaluate real number. They can be a combination of the following:

Variables		Functions	Operators
Real Numbers	Days	Dates	Times

Examples:	10
	IN3
	10 + IN3
	A + IN3 - SQR(ABS(VAR12)) * 12.3 / TUE

Variables

<u>**VARIABLES**</u>

Variables as used in Control Basic consist of conventional variables, and points.

LOCAL VARIABLES

There are 26 local variables available to each program. These variables can assume any floating point value. The current value of a local variable can be determined by using the PRINT command (i.e.: PRINT A). Local variables are unique to each program and cannot be used anywhere else in the system. The symbols used for local variables are the characters of the alphabet (i.e.: A,B,C...Z).

POINTS

The 10 point types are variables because they have a value. This value may be accessed, and some points are set through Control Basic. The table below summarizes the fields of each point that can be read or written to:

POINT TYPE	MNEMONIC	READ FIELD	WRITE FIELD
Outputs	OUT1 thru OUT128 Output	Output	
Inputs	IN1 thru IN128	Input	Input
Variables	VAR1 thru VAR128 Value		Value
Controllers	CON1 thru CON64	Output	Setpoint Value
Weekly Routines	WR1 thru WR32	Output	n/a
Annual Routines	AR1 thur AR8	Output	n/a
Programs	PRG1 thru PRG128	Status	Status
Control Groups	GRP1 thru GRP32	n/a	n/a
Digital Monitors	DMON1 thru DMON128	On Time	n/a
Analog Monitors	AMON1 thru AMON96	S(Status)	S
Array Elements	AY1[x] thru AY48[x]	Value	Value

All point type variables can be referred to by their label. Some statements and functions would be illogical using certain points. If an illogical operation is attempted, nothing will happen.

Examples: START WR1 (cannot start Weekly Routines in BASIC)

DMON1 = 23 * A. (DMONS cannot be set by BASIC)

List of Statements

The complete list of Control Basic statements and operators:

STATEMENT	TYPE	DESCRIPTONSUMMARY
NUMBERS	DATA	Real numbers are handled by the T3000 in full floating point format
DAYS	DATA	Days are evaluated as an integer from SUN=0, MON=1, TUE = 2, WED=3... to SAT = 6
DATES	DATA	Dates are handled internally by the T3000 as a number from JAN1=1 up to DEC31=365
TIMES	DATA	Times are handled internally as hours, but can be typed HH:MM:SS in a program
VARIABLES	DATA	Variables are handled internally as real numbers with full floating point notation
NOT	OPERATOR	Evaluates to TRUE if the opposite is true. IF NOT HOT evaluates to true when COLD
^	OPERATOR	Exponent, raises a number to a power as in 2^2 = 4
* / \ MOD	OPERATOR	Multiplication, division, integer portion of a division, remained of a division
+ -	OPERATOR	Addition and subtraction
< > ... etc	OPERATOR	Less than, greater than, and other combinations.

AND	LOGIC	Logical AND, both conditions must be TRUE for the expression to evaluate to TRUE
OR	LOGIC	Logical OR, if either expression is TRUE the result is TRUE
XOR	LOGIC	Logical Exclusive OR, evalutates to TRUE if the two expressions are different
ABS	FUNCTION	Absolute Value, returns the positive equivalent of a negative or positive number
ALARM	COMMAND	Causes an alarm, has built in hysterisis to avoid multiple alarms from the same event
ALARM-AT	COMMAND	Selects which panel will annunciate alarms, default is ALL panels
AVG	FUNCTION	Average of all numbers enclosed in the brackets following this command
CALL	COMMAND	Calls an other program which functions as a sub routine, takes arguments
CLEAR	COMMAND	Clears all the local variables in a given program
CLOSE	COMMAND	Sets a variable or out put to 0, same as STOP but is more intuitive in some cases
CONPROP	FUNCTION	Allows a Basic program to up date the proportional term in a particular PID controller
CONRATE	FUNCTION	Allows a Basic program to up date the derivative term in a particular PID controller
CONRESET	FUNCTION	Allows a Basic program to up date the integral term in a particular PID controller
DALARM	COMMAND	Delayed alarm, allows generation of alarms if a condition is true for a certain time
DECLARE	COMMAND	First line of a sub routine to accept the list of arguments from a CALL statement
DISABLE	COMMAND	Same as STOP or CLOSE, sets an item to 0
DOM	FUNCTION	Day of month, returns the day of the month, starting with 1 on the first of the month
DOW	FUNCTION	Day of week, returns a number from 0 to 6 starting with Sun day as zero
DOY	FUNCTION	Day of the year, starting with Jan1 as 1
ENABLE	COMMAND	Same as START or OPEN, sets the value of an item to 1.
END	COMMAND	Ends execution of a program, T3000 assumes and end at the end of each program
FOR	COMMAND	Loop control statement, FOR A =1 TO 10 ....PRO GRAM LINES.....NEXT A
GOSUB	COMMAND	Branches to sub routine, executes program till next RE TURN statement, and loops back
GOTO	COMMAND	Unconditionally branches program execution to an other line
HANGUP	COMMAND	Ends a phone call in progress

IF	COMMAND	Logical decision, IF HEAT MODE THEN START OUT1
IF+	COMMAND	Executes the program line only once when the condition goes from FALSE to TRUE
IF-	COMMAND	Opposite of IF+, executes the line only when a condition be comes FALSE
INT	FUNCTION	Returns the integer part of an expression
INTERVAL	FUNCTION	IF IN TER VAL(0:01:00) executes the following programming once every minute
LET	COMMAND	Assignment operator, LET B = 20, has the same effect as B = 20
LN	FUNCTION	Natural logarithm of the argument
LN-1	FUNCTION	In verse logarithm of the argument
MAX	FUNCTION	Maximum of a list of arguments enclosed in brackets following the statement
MIN	FUNCTION	Minimum of a list of arguments enclosed in brackets following the statement
NEXT	COMMAND	Used in conjunction with the FOR statement to increment the loop counter
ON	COMMAND	ON VAR1 GOTO 100 , 200 , 300 , controls program path depending on VAR1
ON-ALARM	COMMAND	Evaluates to TRUE when ever there is an alarm in the system.
ON-ERROR	COMMAND	Used with PHONE, REMOTE- GET, REMOTE- SET, and RUN- SYSTEM
OPEN	COMMAND	Sets an item to 1, OPEN OUT1 is the same as START OUT1
PHONE	COMMAND	Phones a number, allows T3000 to dial with a modem for numeric paging
POWER -LOSS	FUNCTION	Looks for loss of power, 1 the first time the function is evaluated, o afterwards
PRINT	COMMAND	Print strings at a printer connected to the panel
PRINT- AT	COMMAND	Print strings at a printer connected to an other panel on the net work
REM	COMMAND	Remark, used to add comments to a program
REMOTE- GET	COMMAND	10 REMOTE- GET VAR1 = 2IN1, gets a reading from an other site via modem
REMOTE-SET	COMMAND	Sets a variable at a remote site via modem.
RE TURN	COMMAND	Causes program execution to return from a sub routine
RUN-MACRO	COMMAND	Causes a macro command or series of key strokes to be run
SCANS	FUNCTION	The number of scans a panel is per forming in one second
SET-PRINTER	COMMAND	Tells a T3000 system there is a printer at a particular panel
SQR	FUNCTION	Returns the square root of the argument
START	COMMAND	Sets the value of an items to 1, same as OUT1 = 1 or OPEN OUT1
STATUS	FUNCTION	Status of a panel on the net work, 0 = No Comm, 1 =OK, 2 = no program
STOP	COMMAND	Sets the value of an items to 0, same as OUT1 = 0 or CLOSE OUT1
TBL	FUNCTION	Looks up a value in a table
TIME	FUNCTION	Finds the current time
TIME-OFF	FUNCTION	Finds the time variable has been false
TIME-ON	FUNCTION	Finds the time variable has been true
WAIT	COMMAND	Pauses program execution
WR-ON	FUNCTION	Finds when a weekly routine is set to come on
WR-OFF	FUNCTION	Finds when a weekly routine is set to go off
UNACK	FUNCTION	Determines if there are unacknowledged alarms in system
USER-A	FUNCTION	Finds the user logged on to the local port
USER-B	FUNCTION	Finds the user logged on to the remote port

ABS

Purpose: Returns the absolute value of an expression.

Format: ABS(expression)

Example: 10 REM THE NUMBER OF DAYS FROM THE MIDDLE OF THE YEAR
           20 A = ABS(DOY - 183)

ALARM, ALARM-AT

ALARM

Purpose: To generate alarms based on user defined conditions and includes automatic hysterisis to reduce multiple triggers from the same event.

Format: ALARM x > y , z , string
ALARM x < y , z , string

Description: x > y and x < y are the alarm conditions

z is the alarm hysterisis, once the alarm is triggered the condition must swing back by this amount before another event is triggered.

string is the alarm message with a length of up to 69 characters.

Example: 10 ALARM SAT > SETPOINT + 5 , 2 , SUPPLY AIR TEMP IS TOO HIGH

In this example, the alarm is triggered when the supply air is 5 degrees above setpoint. Another alarm will not be triggered till the alarm resets itself by swinging thru 3 Degrees again (5 - 2 = 3 degrees) over setpoint.

ALARM- AT

Purpose: Selects which panel(s) will receive the alarms.

Format: ALARM-AT panel1 panel2 ...] or ALARM-AT ALL

Example: 10 ALARM-AT ALL

20REM ALARMS WILL GO TO ALL PANELS.

30ALARM-AT 2 8

40REM ALARMS WILL GO TO PANELS 2 AND 8 ONLY

_____

AVG

Purpose: Returns the average of a series of numbers

Format: AVG(expression , expression , ...)

Note: A space is needed before and after each comma.

Example:
  10 AVGTEMP = AVG(RMTEMP1 , RMTEMP2 , RMTEMP3)

CALL

[TBD] This statement not implemented yet

CALL
Purpose: To call a program with a list of arguments to be used in that program. Used in conjunction with DECLARE

Format: CALL program = [arg1 , arg2 ...]

Description: program is the particular program to call, ie PRG1 or PRG2 for example.

         = arg1, arg2 are arguments to be passed to the program

Note: Arguments may be any point, local variable, or expression and the '=' sign is necessary before the list of args.

      Up to 40 arguments may be listed.

Programs can be nested up to 10 levels using CALL and DECLARE.

Example: 10 CALL PRG1 = SATZ1 , 21 , SF1 , CON1 , 18.5 , 22.8 , VALVE1

The values of each of the arguments get passed to PRG1. PRG1 uses them and may change the values if the arguments are not constants.

CLEAR

Purpose: Clears all local variables (i.e. A-Z) to zero.

Format: CLEAR

Example: 10 CLEAR

_____

CLOSE

Purpose: To set a value of a variable to 0 (i.e. off).

Format: CLOSE variable

Example: 10 CLOSE VENT

20 CLOSE A

30 CLOSE VAR1

DALARM

Purpose: To program delayed alarms, a condition must exist for a certain amount of time before the alarm is generated. A new alarm will not be created till the condition returns to normal and then returns to alarm at which time the delay timer starts again.

Format: DALARM expression , num , string

Description: expression is the alarm condition

num is the number of seconds to delay before causing an alarm

string is the alarm message with a length of up to 69 characters.

Example: 10 DALARM FAN1 AND NOT FAN1_ST , 60 ,FAN1 NOT COMING ON
            20 DALARM NOT FAN1 AND FAN1_ST , 60 ,FAN1 TURNED ON MANUALLY

Declare

[TBD] This statement not implemented yet

DECLARE

Purpose: To receive a list of arguments into a program.

Used in conjunction with CALL.

Format: DECLARE [var1 , var2 ...]

Description: var1, var2, ... are variables (i.e. A, B, VAR1)

Example: 10 DECLARE SAT SAS FAN CTRL MIN-TEMP MAX-TEMP VALVE 100 END

Notes:

1.   DECLARE must be used on the first line in the program. Using DECLARE makes the program a "subroutine" and should only be run from a CALL statement.

2.   The variables are set to the value of the corresponding arguments in the CALL statement.

3.   Use END statement to end the program.

4.   When END statement is executed, the values of the variables are copied back to the original arguments in the CALL statement.

5.   [TBD] Disable the "Timer" field of a program using DECLARE so that it will not run unless called by CALL.

Program	Full Label	Status	Auto/Manual	Size
☐ 1	MODE PROGRAMMING	ON	Auto	1483
☑ 2	VALVE SUBROUTINE	OFF	Auto	1081

6.   CALL and DECLARE are useful for keeping a program modular. The logic only has to be written once and can be used by many systems by calling the routine with different arguments for each system.

DISABLE

Purpose: To set the value of a variable to 0 (i.e. off).

Format: DISABLE variable

Example:

10	DISABLE OUT1
20	DISABLE A
30	DISABLE VAR1
40	DISABLE PRG10

DOM, DOW, DOY

Purpose: To find the current day of the month.

Format: DOM

Example: 10   IF+ DOM = 1 THEN 100

          20 END

          30 REM NEW MONTH

DOW

Purpose: To find the current day of week (Sun=0).

Format: DOW

Example: 10 A = DOW

                    IF DOW = MON THEN
          20   START BLR2

DOY

Purpose: To find the day of year as a number from JAN1 which evaluates to 1, on up to DEC31 which is 365.

Format: DOY

Example: 10 IF DOY = JUL1 THEN GOSUB 100 20 END

30 REM HOLIDAY PROCEDURE

40 RETURN

ENABLE

Purpose: Sets the value of a variable or other item to 1.
           Also can be used to turn a binary variable or output ON

Format: ENABLE variable

Example: 10 ENABLE FAN101
           20 ENABLE A
           30 ENABLE HEATMODE

END

Purpose: Ends execution of a program. Used to control flow of logic through a given program.

Format: END

Example: 100 REM CHECK IF THE BOILERS ARE RUNNING
           110 IF NOT BOILER-S THEN END
           120 REM ONLY EXECUTE THE NEXT LINES IF THE BOILERS ARE RUNNING
           130 START PUMP101

Programming tip:
          GOTO statements, END statements and similar jumps in the basic programming
language can lead to code which is hard to debug.
           Logic with a lot of GOTOS and jumps is referred to as spaghetti code, use these
statements sparingly.
           The above example can be done with the same logic below without using the END
command
          BOILER   is an output, shows us the boiler is asked to come on.
          BOILER-S is an input which shows that the boiler is actually running.

          100 IF BOILER THEN START PUMP101 ELSE STOP PUMP101
          110 IF NOT BOILER-S THEN STOP PUMP101

FOR

Purpose: Looping control. Used in conjunction with NEXT.

Format: FOR a = x TO y [STEP z]

Description: a is a variable to be used as a counter

x is an expression that is the initial value of the counter y is an expression that is the maximum of the counter z is an expression to be used as an increment

If x is greater than y, when z is positive, or if x is less than y, when z is negative, execution jumps to the statement following the NEXT. Otherwise the lines following the FOR statement are executed until the NEXT statement is en countered. Then the counter is incremented by z. If you do not specify a value for z it is assumed to be 1. If the counter is now greater than y execution continues with the statement following the NEXT. If the counter is equal or less than y,

execution branches back to the statement following the FOR statement. If the value of z is negative, the test is re versed. The counter is decremented, and the loop is executed until the counter is less than y. FOR NEXT loops can be nested up to 10 levels.

Example 1: OUT1 will slowly ramp up to CON1

10 FOR A = 0 TO CON1 STEP .1 20 OUT1 = A

30 WAIT .1

40 NEXT A

Example 2: Check the status of all panels 10 FOR A = 1 TO 12

20 IF STATUS(A) < > 1 THEN GOSUB 100 30 NEXT A

40 END

50 REM ACTION TO BE TAKEN WHEN BAD PANEL

60 RETURN

GOSUB

Purpose: To call a subroutine. Used in conjunction with RETURN.

Format: GOSUB line

Description: line is the line number

EXAMPLE
10 GOSUB 100
20 REM THIS LINE GETS EXECUTED AFTER LINE 120
30 END

100 REM SUBROUTINE GOES HERE
110 A = A + 1
120 RETURN

GOTO

GOTO

Purpose: To redirect the program execution.

Format: GOTO line

Purpose: Redirects program flow, jumps from the current line to another line number.

Format: GOTO linenumber

Description: GOTO statement causes the program to jump to the line referred to by linenumber

Example:

```
10 GOTO 100

20 REM THIS LINE DOES NOT GET EXECUTED

30 REM LINE 120 JUMPS BACK HERE

40 B = B + 1

50 END

100 REM LINE 10 JUMPS THE PROGRAM TO THIS LINE

110 A = A + 1

120 GOTO 30
```

HANGUP [TBD]

Purpose: To end a phone call over the dialup modem.

Format: HANGUP

Example: 10 PHONE 555-1234

20 PRINT "MESSAGE"

30 WAIT 0:02 : WAIT FOR PRINT LINE TO BE DONE

40 HANGUP

IF

Purpose: Logic statement for decision making.

Format: IF x THEN clause [ELSE clause]

Description: x is any expression

clause is a BASIC statement or sequence of statements (separated by commas) or a line number to branch to

If x is true (non zero), the clause following the THEN is executed. If x is false (zero), the clause after the ELSE is executed.

Example: 10   IF OUT1 THEN START OUT2

          IF AVG(A , B , C) > 22.4 THEN 30 ELSE 50

          20

          30 PRINT "AVERAGE IS ABOVE 22.4"

          40 END

          50 PRINT "AVERAGE IS BELOW 22.4"

          60 END

IF+

Purpose: To detect occurrence of event.

Format: IF+ x THEN clause [ELSE clause]
Description: x is any expression

clause is a BASIC statement or sequence of statements (separated by commas), or a line number to branch to

If x is true, and on the previous check it was false, the clause following the THEN is executed. Otherwise the clause after the ELSE is executed.

Example: 10   IF+ CHILLER THEN GOSUB 40

            20   IF+ TIME > 12:00 THEN GOSUB 60

            30 END

            40   PRINT "THE CHILLER HAS JUST COME ON"

            50 RETURN

            60   REM RING LUNCH BELL

            70   START BELL

            80   WAIT 00:00:05
            90   STOP BELL

IF-

Purpose: To detect occurrence of event.

Format: IF- x THEN clause [ELSE clause]

Description: x is any expression

clause is a BASIC statement or sequence of statements (separated by commas), or a line number to branch to

If x is False, and on the previous check it was true the clause following the THEN is executed. Otherwise the clause after the ELSE is executed.

Example: 10 IF- CHILLER THEN GOSUB 30 20 END

30 PRINT "THE CHILLER HAS JUST GONE OFF"

40 RETURN

INT

Purpose: To find the integer portion of an expression.

Format: INT(expression)

Example: 10 REM TO MAKE VAR1 = THE HOUR OF THE DAY (0-23). 20 VAR1 = INT(TIME / 100)

INTERVAL

Purpose: To perform an operation at a regular interval. The functions returns 1 every expression time, otherwise it will return 0.

Format: INTERVAL(expression)

Example: 10 IF INTERVAL(0:10:00) THEN GOSUB 50

20   REM VAR1 WILL INCREMEN BY 2 EVER HOUR

30   VAR1 = VAR1 + 2 * INTERVAL

40 END

50   REM THIS ROUTINE EXECUTE EVERY 10 MINUTES

60 RETURN

LN, LN-1

LN

Purpose: To find the natural logarithm of an expression.

Format: LN(expression)

Example: 10   VAR1 = LN(IN1)

LN-1 (T3000 only)

Purpose: To find the inverse natural logarithm of an expression.

Format: LN-1(expression)

Example: 10 VAR2 = LN-1(VAR1)

LET

Purpose: Assigns the value of an expression to a variable.

Format: LET a = x
Description: a is a variable

x is an expression

Example: 10   OUT1 = 5.5

20   LET VAR3 = ABS(6 * VAR1 - 100)

MAX, MIN

MAX

Purpose: To find the maximum value from a list of expressions.

Format: MAX(expression , expression ...)

Example: 10 BLDG-HIGH = MAX(RM1 , RM2 , RM3 , RM4)

Note: A space is needed before and after each comma.

MIN

Purpose: To find the minimum value from a list of expressions.

Format: MIN(expression , expression ...)

Example: 10 VAR1 = MIN(ABS(DOY - 183) , VAR2)

NEXT

Purpose: Looping control. Used in conjunction with FOR.

Format: NEXT [a[,b,...c]]

Description: a,b, and c are variables used as counters in FOR statements

Example: also see FOR 100 NEXT 120 NEXT A

130 NEXT B , C , D

ON

Purpose: Selectable GOTO or GOSUB.

Format: ON n GOTO line [, line...]

ON n GOSUB line [, line...]

Description: n is an expression rounded to an integer line is a line number in the program

The value of n determines which line number in the list the program uses for branching. For ex am ple, if n is 3, the third line number in the list is the point to which the program branches.
If the value of n is illegal for the given list, no branch occurs.

Example: VAR1 = AVG (IN1 , IN2 , IN3 , IN4) - 20

20	ON VAR1 GOSUB 100 , 200 , 300 , 400 , 500
30 END	
100	REM THE AVERAGE IS 21 : RETURN
200	REM THE AVERAGE IS 22 : RETURN
300	REM THE AVERAGE IS 23 : RETURN
400	REM THE AVERAGE IS 24 : RETURN
500	REM THE AVERAGE IS 25 : RETURN

ON-ALARM

Purpose. To redirect program execution if there was an alarm.

Format: ON-ALARM line

Description line is the line number

Example

```
10 DALARM FIRE = 1 , 10 , THERE IS A FIRE
20 ON-ALARM 100
30 REM CONTINUES HERE IF NO ALARM
40 END
100        REM SPECIAL FIRE PROCEDURES
110 STOP BOILER1 : STOP BOILER2
```

Note This branch happens only once when the alarm is generated.

ON- ERROR

Purpose: To redirect program execution if there was an error.

Format: ON-ERROR line

Description: line is the line number

Example:

10 PHONE 555-1234

20 ON-ERROR 100

30 REM CONTINUES HERE IF NO ERROR

40 REM PROCEDURE IF CALL NOT SUCCESSFUL

Note: ON-ERROR can be used after the following statements:

PHONE REMOTE GET REMOTE   SET RUN-SYSTEM.

OPEN

Purpose: To set the value of a variable to 1.
            Also can be used to turn a binary variable or output to ON

Format: OPEN variable

Examples: 10 OPEN VENT 20 OPEN A

30 OPEN VAR1

PIDProp, PIDInt, PIDDeriv

The PIDPROP, PIDRESET and PIDRATE statements allow you to change the PID terms in a control basic program. These commands will be useful for a program which adjusts the PID parameters on the fly such as for a self tuning PID.

A quick summary of the PID terms from the [PID controller discussion](#) are shown here for reference:

The proportional term can be thought of as the "rigidity" of the PID loop and its response to a deviation from setpoint. A large value will make the valve or damper lazy while a small term will be more hyper active.

The Integral term accumulates error over time and adds it to the PID loop calculation. A large value for the I term will make the PID more active and a small I term will make the PID loop lazier.

The derivative term allows the PID loop to compensate for a rapidly changing process, if the temperature is quickly approaching setpoint then the D term will increase and slow down the PID response. For common HVAC processes this is left at zero.

NUM	Input	Value	Units	A/M	Output	Setpoint	Set Value	Units	Action	Prop	Int	Der	Bias
1	CHAMTEMP	62.4	Deg.C	Auto	100.0%	TEMPSETP	50.0	Deg.C	+	1	0	0	0
2	CHAMTEMP	62.4	Deg.C	Auto	0.0%	TEMPSETP	50.0	Deg.C	-	1	0	0	0
3	HUMIDITY	65.0	%RH	Auto	0.0%	HUMSETPT	80.0	%	+	1	0	0	0
4	HUMIDITY	65.0	%RH	Auto	100.0%	HUMSETPT	80.0	%	-	1	0	0	0
5				Auto	0.0%				-	0	0	0	0
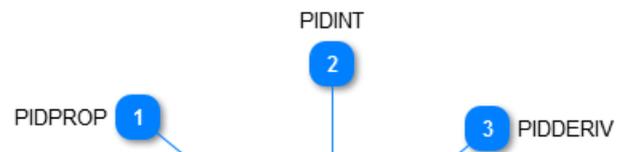
PIDPROP

Purpose: A user defined program can adjust the proportional term of the PID control loops.

Format: PIDPROP(PID# , value)

Range of the value: 0 TO 4000

Description: PID# is the PID loop you wish to change and value is the new proportional term to be set.

PIDPROP returns the value that is currently entered into the PID loop proportional term.

Example: 10 A = PIDPROP(1 , 2.3 * SAT * FACTOR)

PIDINT

Purpose: Enable user programs to change the PID loop integral setting.

Format: PIDINT(pid# , value)

Range of the Value: 0 to 255

Description: pid# is the PID loop you wish to change and value is the new integral value you want to set for this loop.

PIDINT returns the current value that is entered into the PID loop Integral term.

Example: 10 A = PIDINT(1 , 0.85)

PIDDERIV

Purpose: Allows programs to change the PID loop derivative setting.

Description: PID# is the loop you wish to change the derivative value on, value is the new setting.

Format: PIDDERIV(pid# , value)

Range for the value: 0 to 2.00

PIDDERIV returns the value that is entered into the PID loop derivative setting.

Example: 10 IF ABS(SAT - SAS) < 0.5 THEN A = PIDDERIV(5 , 50) ELSE A = PIDDERIV(5 , 0)

PID LOOP MATH DISCUSSION

The output value of a controller is calculated according to the following formulas:

OUTPUT =   Op + Oi + Od + Ob

Op   =   Proportional Output = A * 100 / Pb * (I - S)

Oi   =   Integral Output=A * R * t/60 * (I - S) + Oi

Od   =   Derivative Output=A * D*((I - S) - (It-1 - St-1)) / t

Ob   =   Bias

A   =   Action (+1 for direct acting, -1 for reverse acting)

Pb   =   Proportional Band

I   =   Input

S   =Setpoint

R   =Reset Constant = Repeats per hour Definition: One repeat per hour will cause an output change of one over one hour for an error of one.

D   =Rate Constant = Minutes   Definition: One minute will cause an output of one for an error changing by one per minute.

T   =Time in Minutes since last scan.

It-1   =Input on last scan.

St-1   =Setpoint on last scan.

Note: Controllers are executed once per second.

Output Limiting Procedure:

1.   The sum of Op + Od + Ob is limited to 0-100.

2.   Oi is added to the previous sum.

3.   The output is again limited to 0-100.

4.   Oi is adjusted by the amount that the output was limited in step 3.

Note: Oi is reset to zero when R is set to 0.

Oi is not reset when panel is shut off and on.

PID Loops Introduction

<TODO>: Insert description text here... And don't forget to add keyword for this topic

PHONE

PHONE   (T3000 only)

Purpose: To make connection with a remote computer using a modem.

Format: PHONE telephone_number_string

Description: string is a string of characters representing the phone number

Example: 10 IF+ TIME >4:00 THEN 15 ELSE 70 15 PHONE 9-555-1234

20 ON-ERROR 100

30 RUN-MACRO 1

40 WAIT 0:10

50 HANGUP

60 END

70 REM CONNECTION NOT MADE

PHONE   (Mini-Panels only)

Purpose: To allow the Mini-Panel to call a PC running the T3000 software.

Format: PHONE string

Description: string is a string of characters representing the phone number

In order for the PHONE statement to work all Mini- Panels on the network must have Version 1.91 or higher. The PHONE statement utilizes output 8 of the Mini- Panel which is connected to the modified modem interface box. The modem must be set to not echo characters (ATEO).

Example: 10 REM VAR31 = 1 WHEN CONNECTED

20 REM E = 1 WHEN ERROR TRYING TO CONNECT 30 REM R = RETRY COUNTER

40 REM VAR32 = TIME TO REMAIN CONNECTED , 0 WHEN NO ALARMS 50 IF E THEN GOTO 130

60 IF VAR31 THEN GOTO 200

70 IF VAR32 = 0:00:00 THEN GOTO 220

80 REM CALLOUT ALARMS

90 PHONE ATDT 555-3075

100 ON-ERROR 120

110 VAR31 = 1 : E = 0 : END 120 E = 1 : END

130 REM ERROR SO TRY AGAIN LATER

140 IF R > 20 THEN R = 0 , E = 0 , VAR32 = 0 , END

150 IF TIME-ON(E) MOD 0:03:00 < 0:00:05 THEN R = R + 1 , GOTO 80 160 GOTO 220

170 REM ALARM SO INCREASE CONNECT TIME

180 IF VAR32 = 0 THEN VAR32 = VAR32 + 0:01:00 ELSE VAR32 = VAR32 + 0:00:15

190 END

200 REM IS IT TIME TO END CALL
210 IF TIME-ON(VAR31) > VAR32 THEN VAR31 = 0 , R = 0 , VAR32 = 0 ,
HANGUP

220 REM ALARM STATEMENTS HERE

230 DALARM VAR31 , 5 , ALARMS FROM BOSTON BUILDING 240 ALARM TEMP > SETP , 2 ,
TEST ALARM

250 ON-ALARM 180

POWER-LOSS

Purpose: To allow a program to perform some logic determine if there was a power loss to the panel. The function returns 1 the

first time after a power loss, otherwise it returns 0.

Format: POWER-LOSS

Example: 10 IF POWER-LOSS THEN GOSUB 30 20 END

30 REM INITIALIZATION ROUTINE

40 RETURN

PRINT, PRINT-AT

PRINT (T3000 only)

Purpose: To print strings on a printer

Format: PRINT string [;]

Description: string is the string to be printed

If a semi colon is included at the end of the PRINT statement no CR/LF (Carriage return / Line feed) will be done, otherwise a CR/LF is done.

Example:

10 IF+ FIRE THEN GOSUB 100

20 END

100 REM FIRE INSTRUCTIONS

110 PRINT "THE FOLLOWING PROCEDURE SHOULD NOW BE

CARRIED OUT:"

120 PRINT "1. .... etc."

130 RETURN

Note: The PRINT-AT statement must be used before using PRINT.

Keywords recognized in PRINT statement:

- DATE   prints the date and time

- TIME   prints the date and time

- USER-A   prints the name of user logged on at port A (local)

- USER-B   prints the name of user logged on at port A (remote)

- BEEP   causes the printer to beep

- POINT/LOCAL VARIABLE VALUES

prints values from -999,999.9 to 999,999.9

Remember to use the print statement in conjunction with the PRINT- AT statement to specify which panel to print at. If you do not use the PRINT-AT statement, it will not print.

Be careful when the IF+, IF-, and INTER VAL statements are used with PRINT. Make sure that the same item is not printed repeatedly.

Examples: 10 PRINT-AT 2 , 5 10 PRINT-AT 2 , 5
20 IF+ USER-A THEN PRINT USER-A ; "HAS SIGNED ON AT PORT A ON" ; DATE
30 IF+ VALUE > MAXIMUM THEN GOSUB 100
40 IF INTERVAL(01:00) THEN PRINT "BUILDING LOW IS: ; L : " TIME:" ; TIME

```
50 END
100 PRINT "VALUE IS" ; VALUE ; " AT " ; DATE ; BEEP ; BEEP ; BEEP ; BEEP
110 PRINT "MESSAGE"
120 PRINT "MESSAGE"
130 RETURN
```

PRINT-AT [TBD]

Purpose: To select which panel(s) will receive the print message(s).

Format: PRINT-AT panel [panel panel] or PRINT-AT ALL

Examples: 10 PRINT-AT ALL

20	PRINT "THIS IS PRINTED AT ALL PANELS IN SYSTEM."
30	PRINT-AT 2
40	PRINT "THIS IS PRINTED AT PANEL 2 ONLY."
50	PRINT-AT 1 3
60	PRINT "THIS IS PRINTED AT PANELS 1 AND 3."
70	PRINT-AT 1 5 8
80	PRINT "THIS IS PRINTED AT PANELS 1, 5 AND 8."

Note: Commas are not needed to separate panel numbers

REM

Purpose: Add text comments in program to help understand the programming statements.

Format: REM string

Description: string is a string of characters.

Example: 10 REM *** HEATING AND COOLING SETPOINTS ***

Note: REM commands are ignored by the program, they are only to provide comments in the code.

REMOTE-GET, REMOTE-SET

Purpose: To get a value from another system.

Format: REMOTE-GET a = remote point

Description: a is the variable that will assume the value of the remote point remote point is the point in the remote system

Example: 100   PHONE 555-1234 REMOTE-GET A = 3-IN5

110

120 HANGUP

130 ALARM A >24 , 2 , REMOTE SITE TOO HOT

REMOTE-SET	(T3000 only)

Purpose: To set a value in another system.

Format: REMOTE-SET remote point = x

Description x is the variable that the remote point is to be set to

remote point is the point in the remote system

REM START A REMOTE OUTPUT

Example: 10

20   PHONE 555-

30   A = 1

40   REMOTE-SE

RETURN

Purpose: To return from a subroutine. Used in conjunction with GOSUB.

Format: RETURN

Example: 10   GOSUB
                                                                                            150

                                                                                            REM CONTIN
           20   SUBROUTINI

           30 END

           40   REM SPECIA

           50   FACTOR = A

           60
           RETURN

RUN-MACRO

[TBD]

Purpose: To start a system program running in a connected PC.

Format: RUN-SYSTEM num

Description: num is a number from 1 through 16 representing the desired program

Example: 10 IF+ TIME >4:00 THEN PHONE 9-555-1234 ELSE 100 20 ON-ERROR 100

30 RUN-SYSTEM 1

35 REM WAIT WHILE SYSTEM PROGRAM IS RUNNING

40 WAIT 0:10

50 HANGUP

60 END

70 REM CONNECTION NOT MADE

SCANS

Purpose: Returns the number of scans per second the panel is performing for all logic in all programs.

Format: SCANS

Example: 10 A = SCANS

SET-PRINTER

[TBD]

 T3-BB model only

Purpose: Allows a user program to send custom data out the serial port. Normally this will be a serial printer but can also be custom devices communicating over a serial link. .

Format: SET-PRINTER A
             SET-PRINTER B
             SET-PRINTER 0

Example:
  10 SET-PRINTER B : REM PRINTER CONNECTED TO 1200 BAUD MODEM PORT
  20 SET-PRINTER A : REM PRINTER CONNECTED TO 9600 BAUD LOCAL PORT
  30 SET-PRINTER 0 : REM NO PRINTER CONNECTED NOW

Using lower-case letter in the SET- PRINTER statement will allow only messages created with the PRINT statement to be printed out to the serial printer. In other words, messages generated by either the system or ALARM statements will not be printed.

Example: 10 Set-Printer b

SQR

Purpose: To find the square root of an expression.

Format: SQR(expression)

Example: 10 A = SQR(IN1)

Purpose: To set a value of a variable to 1. Normally used to turn an output on. Also, normally only used with items which are configured as binary, ON-OFF for the range.

Format: START variable

Examples: 10 START OUT1
            20 START A
            30 START VAR1

STOP

Purpose: To set a value of a variable to 0(i.e. OFF).

Format: STOP variable

Examples: 10   STOP OUT1

            20 STOP A

            30 STOP VAR1

            40   STOP PRG10

STATUS	
Purpose:	To find the status of a panel. The function returns 0 if there's loss of communication, 2 if the panel has no program, and 1 if everything is fine.

Format: STATUS(expression)

Example: 10   REM TAKE SPECIAL ACTION IF PANEL 2 NOT OK

20   IF STATUS(2) THEN GOSUB 40

30
END

40   START OUT1 : STOP OUT2 : RETURN

TBL

Purpose: To look up a value in a table.

Format: TBL(expression , table#)

Example: 10 FLOW = TBL(VAR1 , 1)

TIME-ON, TIME-OFF

Purpose: Used in programs to trigger events based on the duration of time an item has been on or off.

Use only with items which are binary such as ON-OFF and START-STOP for the range.

Can be used with outputs, inputs, variables, schedules

TIME-ON

Purpose: Returns the length of time that an item has been on

Format: TIME-ON(item)

Example: 10 REM START OUT2 10 MINUTES AFTER OUT1 IS ON
            20 IF TIME-ON(OUT1) > 0:10 THEN START OUT2

TIME-OFF

Purpose: Returns the length of time that an item has been off

Format: TIME-OFF(item)

Example: 10 START OUT1 FOR 20 MINUTES, THEN STOP FOR 20 MINUTES, REPEATING
            20 IF TIME-OFF(OUT1) > 0:20 THEN START OUT1
            30 IF TIME-ON(OUT1) > 0:20 THEN STOP OUT1

TIME

Purpose: The TIME statement enables programs to use time in the calculations in an easy to read format. TIME returns the time as a number with hours as the thousands and hundreds digits, minutes are the ones and tens digits. Seconds are stored after the decimal place in the first and second decimals.

Note: If the seconds are not specified, the time will be interpreted as hours:minutes and the seconds are assumed as zero.

    12:30:05=1250.05
    12:30 = 1230

Times are evaluated to real numbers using the following formula:

time = hour*100 + minutes + seconds / 100

The time of 14:30:30 is returned as a real number of value = 1430.30

EXAMPLE: 10   IF TIME = 13:00:05 THEN START PUMP1
           20 IF+ PUMP1 THEN VAR6 = TIME
           30 REM VAR6 CONTAINS THE TIME THAT PUMP1 STARTED

UNACK

Purpose: To determine if there are any unacknowledged alarms in the system. The function returns 1 if there are alarms that are still on and that have not been acknowledged, otherwise it will return 0.

Format: UNACK

Example: 10 IF UNACK THEN START ALRM-OUT ELSE STOP ALRM-OUT

USER[X]

Purpose: Provides information about which user is logged on and through which port.
    The function will checks all ports and the various communications ports for logged in users and return the user ID for any matches.

Format: USER[X]
    Where X is the port number through which the user has logged in.
        0 = nobody logged in
        1 = checks users from all ports
        2 = Ethernet port
        3 = Main RS485 port
        4 = Subnet RS485 port
        5 = RS232 port (available on the T3-BB model only) [TBD]
        6 = GSM port , only applicable if the unit is equipped with a GSM module.

    The function returns the user ID from the list of users defined in the user list.

10 REM SEND A PERSONAL MESSAGE TO USER NO. 5
20 IF+ USER[6] = 5 THEN GOSUB 40
30 END
40 PRINT "HELLO TO USER#5 SIGNED IN OVER THE GSM LINK"
50 RETURN

WAIT   (T3000 only)

Purpose: To pause program execution.

Format: WAIT expression

Examples: 10 START OUT2

              20 WAIT 0:10

              30REM WAIT 10 MINUTES AT LINE 20

              40WAIT 00:00:10

              50REM WAIT 10 SECONDS AT LINE 30

              60WAIT 150

              70REM WAIT 1 HOUR AND 30 MINS AT LINE 40

              80WAIT A + B * AVG (C , D)

              90REM WAIT WHATEVER THE EXPRESSION EVALUATES TO

Note:
This statement is handy for controlling timing sequences. Other programs in the panel will not be affected, only the program where the statement appears will be suspended.

WR-ON / WR-OFF

[TBD] change this to SCH-ON and SCH-OFF

WR-ON

Purpose: To find when a schedule (weekly routine) is set to come on.

Format: WR-ON(expression1 , expression2)

Description: expression1 = schedule number (1-16)
                expression2 = which on event in the day (1-4)

Example: 10 VAR1 = WR-ON(1 , 1)

WR-OFF

Purpose: To find when a weekly routine is set to go off.

Format: WR-OFF(expression1 , expression2)

Description: expression1 = routine number (1-16) expression2 = which time off (1-4)

Example: 10 VAR1 = WR-OFF(1 , 1)

10.5   ERROR MESSAGES

When the programming changes are done the user hits the F2 key or "SEND" from the menu system to send the changes to the controller. The software checks the program for errors and if all is clear sends the program to the controller. If errors are found the program is not sent and the errors are displayed in a popup and also at the bottom of the editor.
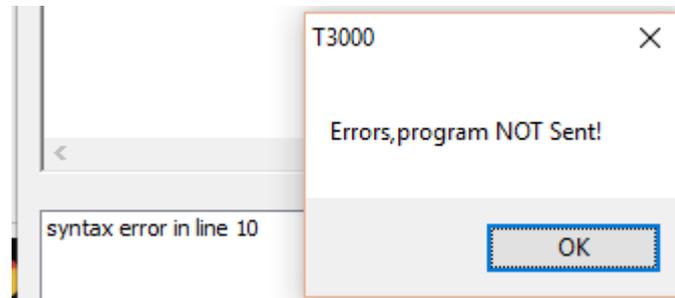
10.5.1 SYNTAX ERRORS

Syntax errors usually are caused by spelling mistakes of syntax errors.

Here is a list of the error messages:

1. Expecting variable

2. Expecting =

3. Expecting operator

4. Bracket error

5. Not variable on left of equals

6. Function argument not expression

7.   Illegal extra characters on line

8.   Expecting another expression

9.   Illegal line number in goto

10.   String is missing end quote (")

11.   Attempt to change the value of a point in another panel

10.5.2 STRUCTURAL ERRORS

1. Program too large —- text buffer

2. Too many GOTOs —- allowed 50

3. Program too large at line ###

4. Too many lines —- allowed 100

5.   FOR without NEXT

6.   GOTO or GOSUB to undefined line ###

7.NEXT without FOR in line ###

8.Too many FOR's —- allowed 10

9.Expecting line number after line ###

10. Line after ### is too long

Port Forwarding

Port Forwarding

Following are the steps for port forwarding:

Step1. Set port forward for router:TCP 502(modbus TCP);UDP 47808(BIP port); UDP 1234(Temco private scan).

Step2.Click to show the tab as below,select "Remote Device"protocol.
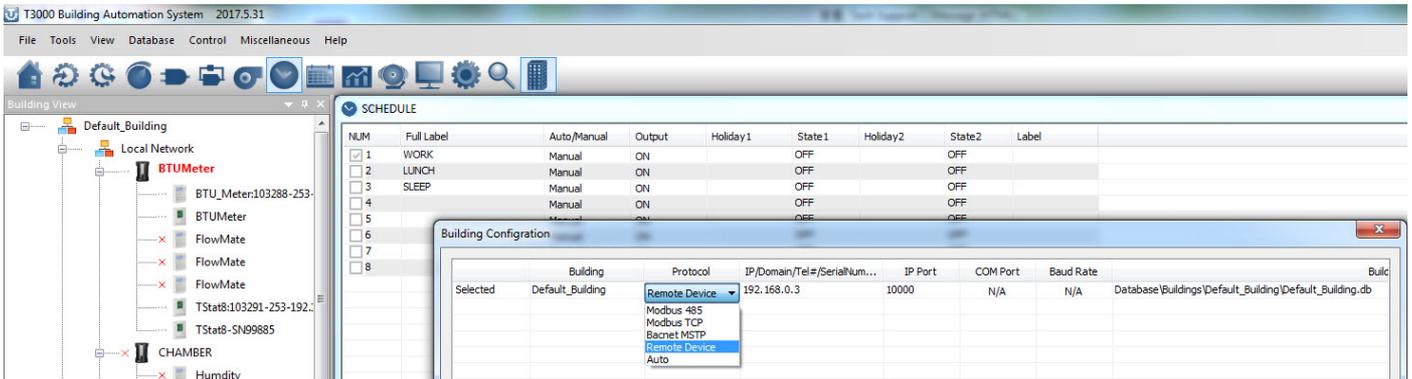
Step3.It will show the tab as below, click to fil outer net IP or domain name.
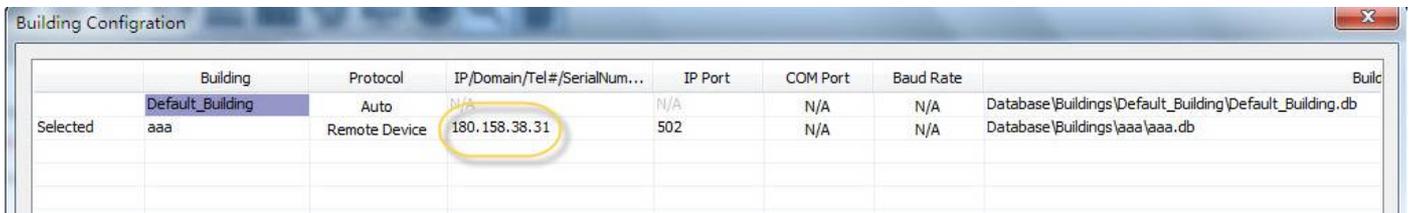
Step4.Scan via T3000 software,the remote device can be connected.
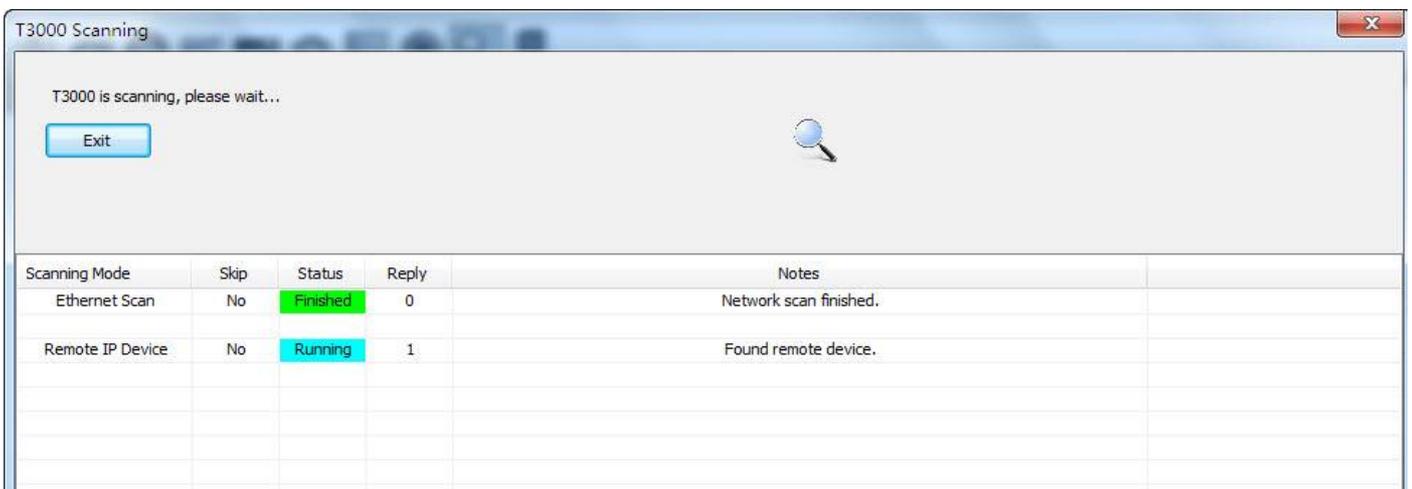
Network&Remote points support

T3-serial controller (only new ARM) can support network points and remote points.

1.*Network points function*

For example,there are panel 1 and 2 in network. We can use 2.2.VARx 2.2.OUTx 2.2.Inx to read or write the network ponts.Range of x is 1 to 255.

2.*Remote Bacnet points*

For example,panel 1 is T3000 controller, device 3 is MSTP slave device,which is integrated in only master RS485 port, and this port is set to master MSTP function.We can use 1.3.AVx 1.3.Aix 1.3.DOx 1.3.AOx to read and write remote bacnet points.Range of x is 1 to 255.

3.*Remote Modbus points*

For example,panel 2 is controller, device 4 is Modbus slave device, which is integrated in master or slave RS485 port, and this port is set to master modbus function.We can use 2.4.MB_REGx or 2.4.REGx to read and write remote modbus points. MB_REG is base 0, and REG is base1.Range of x is 0 to 2047.

RS485 PORT

T3000 Controller
Panel #1

set to master
MSTP function

MSTP Slave
ID #3
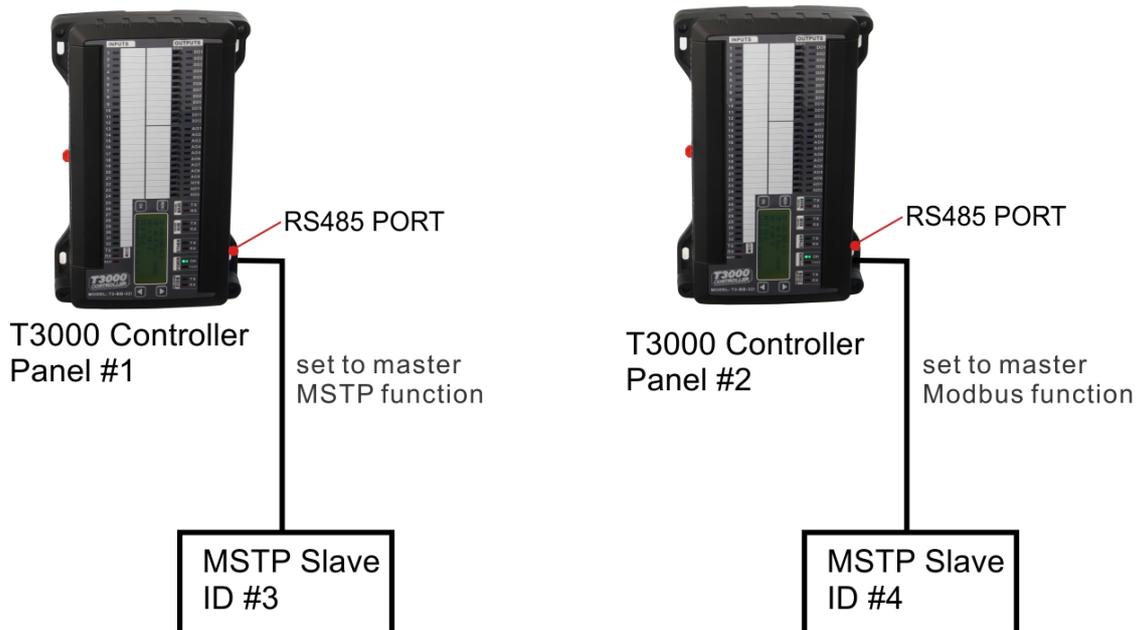
In Panel1:

2.2.VARx
2.2.OUTx   Network points
2.2.Inx   x:-255

1.3.AVx
1.3.AOx   Remote Bacnet point
1.3.DOx   x:1-255
1.3.AIx

RS485 PORT

T3000 Controller
Panel #2

set to master
Modbus function

MSTP Slave
ID #4

In Panel 2:

2.4.MB_REGx   Remote Modbus point
x:0 - 2047

T3000 Release History

Ver 2016.4.14
1 All settings use the same initial file(T3000_config.ini) [TBD clarify]
2 Update all T3 expansion modules to use common UI

Ver T3000 Code Rev 2016.3.26
1 Firmware update service, can manage the firmware and updates of all devices in the building.
2 Can Connect to a firmware update server at www.newfirmware.com to download firmware
3 Check all devices in the building for new firmware and allow user to update any or all

2 Auto detect the braudrate during network discovery
3 Misc fixes for the high baud rate firmware flashing feature.

Ver Rev2016.3.18
1 Firmware updates with ISPTool checks for high baud rate, updates with high baud rate, then sets it back
to the original setting.
2 Modbus Tool can now do multi register writes.
3 Check newfrmware.com for new firmware files.

Ver Rev2016.3.9
1 Move open source T3000 Project From VS2010 to VS2015

Ver Rev2016.2.20
1 Update the latest help file of T3000 to github

Ver 2016.1.4
1 Building Database, as building name changes in the database, the project folder also changes.
2 Fixed AQ readings for Air Quality device.
3 Firmware updates: check the firmware file for a match with the device.
    prevent user from loading bad files to the device.
4 Auto Discover nodes, T3000 will show the progress of the scan: which port and how many devices, etc.

Ver 2015.12.25
1 T3 controllers, allow changing the baud rate to   "9600", "19200", "38400","57600","115200"

Ver 2015.12.23
1 When scanning and discovering new devices,
    If one device is connected by both ethernet and com port, auto select the ethernet connection
    and update the node in the system tree.
2 Misc Bacnet fixes.

Ver 2015.12.18
1 Changed description of HUMTEMPSENSOR to HUM , Model number is 14.
2 Refresh the data in the background for sensors: HUM, AQ, HUM_R
3 Updated the help system
4 Updated the Alarm log .
5 Allow sensors to store a user name: HUM,AQ,HUM_R all have 20 character description which is stored in
the device, non volatile.
6 Updated T3Modules UI to show inputs and outputs
7 ISPTool firmware update tool can auto detect the baud rate of devices stuck in update mode.
8 Misc UI updates.

Ver 2015.11.20
1 Different subnets detected, when T3 controller is stuck in update mode and the PC and device are on a
different IP subnet
    The isptool will shows a message to let user change the ip address of the T3 controller.

Ver 2015.11.3
1 Fixed a bug when loading the config file of Tstat5F, T3000 will crash
2 Added register 453 for the HUM sensors, filter setting for hum and temp sensors.
3 Added UI support for the T3-22AI
4 Added UI support for the T3-8AI8AO6DO
5 Fixed bug where a Tstat5 which is stuck showing "WARNING E2_error", could not update firmware, now can update.
6 Added the dead master for T3 to the GUI
7 Added the output ranges "0-5V" "0-10V" for Tstat6
8 While configuring Tstat7 outputs, was showing "Encountered an improper argument", fixed.
9 Added the pwm range to the output ranges of the tstat7
10 Fixed a bug where PM5E can't communicate with T3000

Ver 2015.10.14
1 Building configuration is easier to use.
2 Custom Input range UI is updated.

Ver 2015.8.18
new features:
1 Can change the device name in the System Tree
2.For custom devices connected to a T3 controller subnet, device information can be shown on the UI
3.User can assign names to the various modes of operation on the Tstat6
4. Write the user assigned name to the device if it supports this feature.
5. #Modes/Speeds, change the names from "1,2,3" to "3,4,5"

Ver 2015.8.3
1 New feature, can change the name of the Tstat6 in the System tree and the Tstat6 UI.
2 When Scanning , the name of Tstat6 will be shown in the tree view.

Ver 2015.7.30
1 ISPTool can support "Arm" devices
2 Fixed the bug when minimize the window ,can't restore the window.
3 No Image information only show in the first 5 seconds.
4 Trend logs if data interval larger than the value we sampling don't connect the two point.
5 T3 controller keywords changed
     PID CON ->   PID
     SCHEDULE WR ->   SCH
     HOLIDAY AR ->   HOL
     GROUP GRP ->   GRP

ver 2015.7.28
1 Customer can define the name of Tstat6 modes such as "Heat Only" "Cool Only", 'Summer' and so on.
2 The user names are stored in the Tstat6.

ver 2015.7.27
1 ISPTool: Ethernet can update firmware of ethernet devices.
2 Bug fixed: Changing the Building name ,T3000 shows incorrect messages.

ver 2015.7.16
1 Default splash screen supports private branding, user can change the bitmap file and store this in the zip file.

ver 2015.7.10
1 Updated icons in icon menu
2 Code from sub branch in GIT repo is merged with main branch. [TBD confirm]

3 T3 controller inputs & outputs, "UNUSED" is replaced by "-"
4 New feature, T3000 will go back to the last screen automatically
5 Use cache to speed up user interface, read new data in the background .
6 Updated Sliders
　　　- Auto scale for sliders
　　　- The min and max values will get the rounding down instead of up.
　　　- in dual setpoint mode ,hide the "setpoint"
　　　- in single setpoint mode ,show the heating and cooling setpoints.
　　　- user can't edit heat or cool in single setpoint mode, only the 'setpoint' can be changed.
　　　- deadband will snap to new size in increments
7 updated batch firmware update feature.
　　　- Show the status of all devices :online or offline
　　　-Show the version of the device and the firmware ,if the software version is the latest version ,don't update it again.
　　　-when selecting a hex or bin file ,click the button of "Apply", the file path will match with the correct device .
　　　-when opening the dialog ,show the stauts of all devices including: "Online" "Software Version" "Firmware Version"
　　　- Check the configuration file ,automatically.[TBD confirm]
8 Custom output ranges, can edit custom ranges in the output dialog directly instead of jumping to a new dialog.
9 Tstat GUI: Changed the word "Internal Sensor" to "Internal Temp Sensor" in the input grid.
10 Tstat GUI: Swap the cooling & heating setpoints .
11 When a device is offline, don't need to click ok, just show a notification message which disappears after next click.

2015.7.9
1 Improve connection speed to bacnet products.
2 Improve speed of graphic screens.
3 Update the status bar with communications details, health and # of packets.
4 Bacnet program IDE. Add operate '&' and "|" feature. eg:
　　　10 VAR1 = 17 & 14 to allow programming of modbus registers at the bit level.
5 Update Trend log view, can auto scroll now.
6 System tree, when sorting nodes the status showed incorrectly, fixed.
7 Updated T3 controller GUI ethernet connections, added "Network number" "MAC address" and so on.
8 In grid view, press "UP" or key "DOWN" to jump to the next controller, providing the next panel is online.
9 Sort product view list by right click and select by name or by connection
10 Icons updated, hotkeys updated
11 Add feature BIP to MSTP. User can access devices on a subnet of the T3 controllers
12 All bacnet grids, column headers aligned left side.

ver 2015.7.7
1 Support new baud rate on T3 controller
2 Updated the defaults for AirQuality sensors
3 Fixed PID Tables for the Tstat6 Auto/Off register.
4 Monitor the register ID, and then show the graphic [TBD clarify]
5 Fixes in the building database configuration
6 Fixes on the inputs and outputs grid of the T3 controllers [TBD confirm]
****
ver 2015.6.26
1 Updated CO2's Output signal range .1:40-20mA 2:0-5v 3:0-10v
2 Deleted Temco Log, user can customize the splash screen.
****
ver 2015.6.15
1.Batch flash bug: T3000 can restore the config file after flashing [TBD confirm].

****

ver 2015.6.12
1.New Feature: "Batch flash" if T3000 is interrupted, can now resume from where it left off.
2. Added Table to store the batch flash progress info.
3.Added Tables and fields to store the register information from custom devices.
4 Updated the function of Delete All Nodes, don't delete customer defined items in the node database
5 Updated the building database path to use the relative path rather than full path.
6 Updated Humidity Chamber reporting, can generate reports directly from T3000.
7 Fix two hot keys, Shift+D ,Shift+F
8 Updated the input grid for Tstats, show '-' rather than 'UNUSED'
9.Fixed installation issues for Win8, "Unable to save file:C:\windows\Downloaded Instations\..."
10 Added custom modbus devices features, user can add any modbus device to T3000 environment

****
ver 2015.5.29
1. Bacnet IP gateway operations to MSTP added, still some work to be done.
2. Network discovery can find MSTP devices which are connected to the Mini Panel MSTP subnet.
3. Bacnet Input and Output add a user label for all the expansion I/O on the external T3 modules on the subnet.
4. Connection to remote buildings, now user can properly configure the domain name for each building
        works using dynamic IP services such as dyndns.org and port forwarding at the remote building.
6. Batch Flash Humidity Sensors in tstat6 during calibration.
7. Add some read/write features for the various settings related to the barometric sensor of the C02 sensor.

8. Add some read/write features for the the T3 expansion modules, more parameters & settings can be edited.

****
ver 2015.5.25
1. CO2 sensors have a barometric pressure sensor, added this sensor to the UI
2. Changed T3 inputs and outputs, use a common grid similar for all products
3. ISP Mode bug, if a device is in firmware update mode, while T3000 is starting, T300 can crash. Fixed.
4. Added Products Registers List Maintenance System, all registers are in a database. T3000 user,
   with appropriate privileges, can add/edit the register database and add new devices.

****
ver 2015.4.15
1 update the register list of tstat6
2 update t3000 help files
3 update the tstat5 LCD for loop1.

****
ver 2015.4.10

1 Fixed: Trend log inputs did not show correctly.
2 Change the point types in the graphical screens window. [TBD clarify]
3 Double click to enter the graphical screens from the grid directly.
4 Program 20 character label disappears after writing, fixed
5 T3000 can't run with Chinese version of Microsoft Access Database, fixed.
6 Fixed a bug where the cursor switches to a circle when clicking a device on the tree many times.

****
ver 2015.4.7
1 Add the "10k Type2" and "10k Type3" for Tstat6 ,Tstat7 ,Tstat5I inputs
2 Add the factory reset button for T3 modules.

3 Add the prompt for T3 modules when clicking the factory reset.
****

****
ver 2015.4.2
1 Fixed the graphical screens, labels didn't show normally in the first few seconds.
2 When editing graphical screens, write data into T3 controllers immediately.
3 Fixed trend log windows, when user switches focus from the inputs window to another application caused crash.
4 Fix label interpreter, before "VAR-3" and "3--VAR" will crash t3000.

****
ver 2015.3.31
1 Add F1 key help system to T3000.
2 Add a new device, the CS3000 to T3000 .
3 Fixed a bug with ISPTool crashing when the device didn't respond
4 New feature added to export the register list for all devices to a spreadsheet.

****
ver 2015.3.26
1. Modify the building list database, now user can add remote ip or domain for remote sites.
2. Add ability to discover new devices on remote sites.
3. Add trend log feature for graphing historical data
4. User programs can be larger than 400 bytes for each program. Now can be 10k total, 2k max per program.
5. Add Dyndns feature , T3 controllers can update the external IP address so users can access the system using a free dyndns domain name.
6. Add time zone feature to sync time from NTP servers.
****
ver 2015.2.7
1. Add feature remote point.
2. Scan function changed , if user select Modubus 485 and COM1,it will only scan this port;
3. Other misc fixes

****
ver 2014.12.24
1.Added Building database to manage connections to many remote sites.

****

ver 2014.12.24
1.Added Building database to manage connections to many remote sites.
****
ver 2014.11.28
1. Add Graphic user screens, can add items, edit point ,delete point. Add Icon .
2. Add graphical window for login screen.
3. Optimize the scan feature.
****
ver 2014.11.11
1 ISPTool can be called automatically from T3000 without filling in COM port and other parameters.
2 T3000 will display the zigbee information for a Tsat6 with Zigbee module.
****
ver 2014.10.17
1 Fix crashing of T3000 when updating database.
2 Updated the AQ,HUM-R graphical user interface
3 HumChamber Test rig, added testo temp and humidity sensor information.
****
ver 2014.10.11

1 Fix a bug when updating database which caused T3000 to crash.
2 Updated the AQ,HUM-R graphical user interface
3 Modbus Pressure Sensor added to GUI.
4 Updated custom modbus devices in the database feature.
****

Ver 2014.9.26,date Released:2014-9-26
****

1 Fixed connection to a remote building, can see subnets now
2 Misc T3000 crashing fixes.

Ver 2014.9.10,date Released:2014-9-10
****

1 Updated custom modbus device database feature.
2. Added modbus tool command to 'Read/Write Once'
3 Updated the Sequence register for Tstat6
4 Updated scan results, show the model number now.
6 Tstats EMS [Norway] supported in the GUI.
7 Updated the GUI for Tstat6,7,5i
8 Fixed the background scanning, fixed displaying of nodes after making changes.

****
Ver 2014.9.4,date Released:2014-9-4
****

1 Can add 3rd party devices to the T3000 environment, user can add their own modbus devices to the database.
2 Updated the Isp firmware updating tool to better detect model number in hex or bin
   and help users to select the proper firmware for a given device
3 Updated Hum Series sensors GUI.
4 Modbus tool, added one column to show the Register Name in the grid.
5 Updated the monitoring of a register value in isptool. Users can monitor the registers' value and write into the text or Excel. [TBD confirm]

Ver 2014.6.19,date Released:2014-6-19
****

1Fix the display of the connection to the Tstats when connected to a T3 controller subnet. Show as net.subnet.controller notation. [TBD confirm]

****

Ver 2014.6.18,date Released:2014-6-18
****

1 When the T3 controller in on a different subnet,T3000 can change the ip of device
2 All inputs can be toggled to auto/manual mode.
3 When t3000 cannot find the database engine, T3000 will go to the temco's website to download. [TBD confirm]
****

Ver 2014.6.17,date Released:2014-6-17
****

1.Control Basic Editor: When finshed editing and the users' program has been sent successfully, there is no need to show the exit Y/N confirmation dialog.
    Just exit right out with the 'Esc' key.
2.Control Basic Editor: Update the error messages after each send of the program.
3.Device Discovery: Fixed misc bugs related to auto discovery of bacnet devices and changes in the IP address.
    Sync the changes properly with the database.
4 Online status of the nodes and the lower connection status bar now synchronized better.

5 When user changes the bacnent "Auto/Manual" field the "value" column will refresh instantly, before it was slow.
6 Match number of inputs of the controller to the hardware model, ie if it has 16 inputs T3000 will show only 16 inputs for the LB model,
        32inputs for the LB model and so on.
7. Change the multi flash update interface, Still to be done: Auto flash and support pause and resume.
8 Batch flash tool can now correctly handle 64k and 128k thermostats.
9 Add feature that user can access the remote bacnet device using a remote IP address.
10. Add the bacnet interface to show the device software version , hardware version ,bootloader version,and other information.
****

Ver 2014.6.13,date Released:2014-6-13
****

1 Modbus tools: Fill the register name from db,automatically
2 When temco's device is connected , modbus tool can show full text of the register description from the database
3 When Scanning ,Tx and Rx can show the data
4 Delete "Deep Scan",now t3000 scans all devices automatically, even with ID conflicts.
****

Ver 2014.6.10,date Released:2014-6-10
****

1 Modbus tool can connect temco's device automatically on all com ports and USB.
2 Updated the config file for Tstat6
3 Fixed some bugs with Tstat6 VAV functions
****

Ver 2014.6.6,date Released:2014-6-6
****

1 Fixed VAV function (previous version doesn't fit in Tstat6 firmware)
2 Changed output range text to show "Close" rather than "Closed"
3 All ranges have added "Open/closed" "closed/open" "on/off" "off/on"
4 Fixed some user interface issues with the Modbus tool
****

Ver 2014.2.5,date Released:2014-4-30
****

1 Fixed the some issues with the setpoint calculations when the number of cooling stages are more than 0.

****
Ver 2014.2.4,date Released:2014-4-26
****

1 Fixed Output table on Tstat 5A, analog outputs
2 Fixed firmware update tool, handles hex files with lines more than 256 chars properly now.
3 Fixed If device is connected by both com port and ethernet ,T3000 lets user select which connection to add to the database
4 Fixed differ the states of connection and disconnection.

Ver 2014.2.3,date Released:2014-4-4
****

1 Fixed some bugs about LightingController (Input-mapping,group-mapping ,write address error)
2 Fixed Tstat's schedule (Occupied ,Unoccupied)
3 Added new modbus device, "Tstat 5i"
4 Added "T3000 Help System, hit F1 for help.

Ver 2014.2.2,date Released:2014-3-21
****

1 Fixed In Tstat6 Main Page,sliders are easier to use

2 Fixed Connected and Disconnected ,the two states show different icons
3 Added Ethernet-Modbus C02 device to the GUI.

Ver 2014.2.1,date Released:2014-3-18
****
1 Fixed In Advanced page for Tstat6, Cool and Heat Deadband's showing incorrectly
2 Fixed In Tstat6 Inputs Dialog ,Internal Sensor's Calibration can not update correctly.

Ver 2014.2.0,date Released:2014-3-10
****
1 Tstat5 Advanced setup page crashing, fixed.
2 Fixed MPC5G can't show the Airflow setting [TBD confirm]
3 Fixed the issue of T3's inputs range

Ver 2014.1.4,date Released:
****
1 Tstat6, Combined LED Table and 'Display Name' into one button called "Display".
2 Fixed some errors in the configuration of Tstat5E LCD display
3 Added Co2 alarming system ,added master CO2 sensor and a subnet of CO2 sensor nodes
4 Updated T3-PT10

Ver 2014.1.3,date Released:2014-2-17
****
1 Bacnet-CM5 boiler controller updated
2 Added T3-6CT current sensor module
3 Add auto discover method for MiniPannel
4 Added T3-PT10 to the GUI, modbus high accuracy Pt sensor module with 10 channels.
5 Delete unused files "Mbpoll" "RegisterMonitor" in project

Ver 2014.1.2,date Released:2014-1-23
****
1 Add "Check Update" to look for a new version of T3000
2 Add "Add net device manually"
3 Fix "Tstat6's Range"
4 Adding parts list and project equipment database management to T3000
5 Add 64 bit and 32 bit support for the T3000 database.
6 Change "Output table" [off]

Ver 2014.1.1,date Released:2014-1-14
****
1 Fix C02 GUI
2 Add the selection of CO2,Humidity in the display name config

Ver 2014.1.0,date Released:2014-1-9
****
1 Add support for Lighting Controller in T3000
2 Add "Register Viewer" to Modbus tool, can see all modbus registers, values and descriptions

Ver 2013.27,date released:2013-12-18
****
1 Tstat6, add "lighting sensor" to GUI
2 Add multi-language "English ,Simple Chinese ,Your language"
3 Add support for win7 64 bits
4 Add support to change modbus ID from T3000
****

Ver 2013.26,date released:2013-12-2
****
1 T3000 GUI updated for "T3-8I13O"

2 Fix "Change ID to 255" in the ISPTool
3 Lighting Controller, weekly and calendar schedules.
****

Ver 2013.25,date released:2013-11-28
****

1 Add "Occupied/Unoccupied" into Input Range for [TBD]
2>to export these items "Input_Name,Range ,Function"
3>to add "Led Table"
4>to add "LC"
5>to fix "CM5" Multi-Read error
6>to fix "subscript out of range"
7>Can edit the name of T3-Module
****

Ver 2013.24,date released:2013-11-8
****

1>to support "Save as..." and "Load"the config file of T3-Modules
2>to support "Save as..." and "Load"the config file of LC
3>to fix the problem of NC's IP that T3000 can not change ip by comport.
****

Ver 2013.23,date released:2013-10-30
****

1 Tstat6: Update the "VAV" dialog
2 Tstat6: to add the "Transducer" to output ranges
3 Tstat6: add "VAV" registers to config file
****

Ver 2013.22,date released:2013-10-15
****

1 Tstat6, add "VAV" Functions
****

Ver 2013.21,date released:2013-9-25
****

1 Fixes and updates for Tstat5G
2 Fixes to UI for T3 controller
3 Misc fixes to the Modbus tool.
4 Tstat6: fixes with the slider control
****

Ver 2013.20,date released:2013-9-22
****

1>to show NC's MAC Address
2>Change ISPTool some tips
"Hex file matches the chip. "
"Hex file doesn't match the chip. "
"Begin Time" -> Change to "Start time"
****

Ver 2013.19,date released:2013-9-13
****

1>Change Tstat5G the same as Tstat5E
2>Fix "Setpoint Value can not write"
3>Fix "Heating Set Point "in night sepoint for loop1 is 1/10 value
4>Fix "Set Heating DB for Day setpoint for Loop1"
5>Fix "Cooling ,Heating DB for Loop2 are not saved to disk"

```
********************************************************************************

Ver 2013.18,date released:2013-9-6
********************************************************************************

1>Updata PID2
2>Add MiniPanel
3>Update Humchamber
4>Rebuild Cumstom Sensor Table
********************************************************************************


Ver 2013.17,date released:2013-8-30
********************************************************************************

1>Add Tstat subnet into NC .
2>Update T3-Module (T3-32AI,T3-8I13O)
3>Support T3 serials products.
For Bacnet:
1.Add programming context,when user use bacnet to program code,if codes contains error,it will prompt the
user which line occurs error.
2.Add static text to show how many free memory the program has,and how many it has used.
3.Add range choose dialog.
4.Add "load descriptors" function like old T3000 do.(Read all date program needed,like "Programs" "Inputs"
"Outputs" and so on)
5.Add wait show dialog,to show the "load descriptors" progress step!
********************************************************************************


Ver2.6.0.16,date released:2013-8-7
********************************************************************************

1>Add VC++ installation package to installation file.
2>Update T3-Module (T3-4AO).
3>Support T3 serials products.
4>Fix  "lcd rotate enable" Add "None" Reg 564
5>Fix "Custom table" Read Error.
********************************************************************************


Ver2.6.0.15,date released:2013-8-5
********************************************************************************

1>Add the Custom input names fucntion to The main page
2>fix the input2,3 "Customer sensor Table"
********************************************************************************


Ver2.6.0.14,date released:2013-8-1
********************************************************************************

<1>fix Output PID2 OFF VALVE TABLE
<2>Add T3 To T3000
<3>Add the function of flashing Serial no
********************************************************************************


Ver2.6.0.13,date released:2013-7-26
********************************************************************************

<1>fix Output PID2 Auto /OFF  FAN/VALVE
<2>Add BACnet to T3000
<3>Add MBPoll to T3000
<4>Fix Fan Speed in output table
********************************************************************************


Ver2.6.0.12,date released:2013-7-26
********************************************************************************

<1>fix Output PID2 Auto /OFF  FAN/VALVE
<2>Add Schedule ON/OFF
```

```
****************************************************************************


Ver2.6.0.11,date released:2013-7-15
****************************************************************************
<1>fix the Schedule of Tstat6/7
<2>Add "RegisterMonitor" [Tool->RegisterMonitor] to T3000
<3>fix the config file
<4>Update CM5
****************************************************************************



Ver2.6.0.10,date released:2013-6-25
****************************************************************************
<1>Use Multi-write to config NC
****************************************************************************


Ver2.6.0.9,date released:2013-6-15
****************************************************************************
<1>add scan method One ID by One ID
****************************************************************************


Ver2.6.0.8,date released:2013-6-7
****************************************************************************
<1>Add log file ,when user scan the device
<2>fix some bugs tstat slide
****************************************************************************


Ver2.6.0.7,date released:2013-5-27
****************************************************************************
<1>Add the feature that user can change the tree node name by Pressing F2
****************************************************************************


Ver2.6.0.6,date released:2013-5-27
****************************************************************************
<1>Add PID2 Fan OFF For Tstat5
<2>Fix Tstat 6 slide
****************************************************************************


Ver2.6.0.5,date released:
****************************************************************************
<1>fix the bug that user can't import his db file or create default t3000.mdb
****************************************************************************


Ver2.6.0.4,date released:2013-5-17
****************************************************************************
<1>Add the "Filter" in the Input
<2>Add the "Hum" and "Co2" to the Input
****************************************************************************


Ver2.6.0.3,date released: 2013-5-16
****************************************************************************
<1>Support Baud 9600 and 19200
****************************************************************************

Ver2.6.0.2,date released:  2013-5-10
****************************************************************************
<1>Add DisplayConfig For Tstat5E,Tstat6,Tstat7
<2>the choice of Loop1,Loop2 is changed ,the Value of them will be changed.
<3>let the data keep sync and fresh in the backgrand
```

<4>fix some crashes when freshing the UI data.
****
Ver2.6.0.1,date released: 2013-5-6
****
<1>Fix the error of missing some DLLs. MFC100U.dll and MSCR100.dll

Ver2.6.0.0,date released: 2013/4/19
****
<1>fix a bug which the boud rate changed to 9600 accidentally.
<2>Add T3000RegAddress. Changed the architecture of the register table.Create two register table in t3000.mdb which include all TSTAT product register list.The T3000.exe will load this value by runtime, instead of compiling.
<3>Modify sevral error in ParameterDlg ,the changed can also support tstat6 and tstat7
<4>Make the program run more smoothly when user click refresh in OutPutDlg and InputSetDlg ,it will no longer appear noresponse.
<5>Change InputTable set dialog .It can change the Input name and then save to DB.
<6>Add HumChamber dialog.
<7>Merge function "Refresh" and "Refresh6" in ParameterDlg.
<8>Change the global variable read tstat buffer size from 512 to 1024.

Ver2.5.0.109,date released: 2013/2/27
****
<1>User can be allowed to resize the col ,row of every Table
<2>Output 4-5 can work in the model of ON/OFF PWM ,can't change in the model of Float
<3>Output 6-7 can work in the model of 0-10/0-5
****
Ver2.5.0.108,date released: 2013/2/27
****
<1>the size of hex file match with the chip memory
<2>Extend the input table size

****
Ver2.5.0.107,date released: 2013/2/25
****
<1>Fix the output1-3
<2>The attibute of "Fan Mode" is readonly in the first page,which keep pace with"Mode" (Advanced->Outputs Table)
<3>Fix Tstat5E&&Tstat5G config File.
<4>Fix Heating Stages and Cooling Stages in the the Outputs Table
<5>Add The MAC Address In the Net Controler and Change the defalt Device name
   from "Tstat5A" to "UNUSED"
<6>Fix some bugs in config file and output table
<7>check the config file data ,and the function of loading the value of the registers
****

ver2.5.0.106, date released: 2013/1/18
****
<1>Fix the "Loop1"->output Value and "Loop2"->output ,make them Read-only
<2>Fix the "Tstat5E" input/output table.
<3>Change The Function of "Scan"
<4>Add the "Freeze Function" in the Advanced Setup
<5>remove the column of Rotation in output table.
<6>Fix "Save as.." the register list of The Tstat5E and 5C
<7>"Database", the Exit botton and Delete botton
<8>Fix the out table
<9>Change "NC Config"->"Apply" botton.
<10> Change the setting of "Fan Model"
****

ver2.5.0.105, date released:2013/1/8

```
******************************************************************************
<1>Fix the "Loop2"->Input Value Read-only
******************************************************************************
ver2.5.0.104, date released: 2013/1/5
******************************************************************************
<1>Fix the error that input1 can input value, in the Advanced Model line "Input value"
<2>Add the reminder of PID2 Off Setpoint "Input value"
   if >=100 "out of range" ,if can't fail to write to the 275 register ,
   "please try again."If write sucessfully,"Write ok"
<3>Keep the steps both Heating Set Point and the set Bars in the first page.


******************************************************************************

ver2.5.0.103, date released:2012/12/11
******************************************************************************
<1>Merge ISPTool into T3000.
<2>Add "Tool->ISPToolForOne"
<3>Add "Tool->ISPToolForMany"
<4>fix the errro: Can't get a correct the E2 chip size from Tstat.
<5>Add .net4.0 component in install package.
******************************************************************************

ver2.5.0.102, date released:2012/12/1
******************************************************************************
<1>Fix Tstat6/7 config file read /write error .
<2>Be able to Load the correct config file
<3>write to the value of register ,according to the config file.
<4>support windows xp/vista/win7
******************************************************************************

ver2.5.0.101, date released:2012/12/1
******************************************************************************
<1>Fix the Error about 102,101 and so on .
<2>Don't limit the T3000 client time of users.
<3>Fix the About Dialog Time.can link to the website of www.temcocontrols.com
<4>Fix the Error about "Contact Software Developer,Please!"
******************************************************************************

ver2.5.0.100, date released:2012-11-20
******************************************************************************
updated T3000 issue


******************************************************************************

ver2.5.0.99, date released:2012-11-05
******************************************************************************
(1)updated NC scan function
(2)fix the Dialog 'Parameter'
(3)changed output6&7 function
******************************************************************************

ver2.5.0.98, date released:2012-10-30
******************************************************************************
(1)updated NC scan function
(2)fixt Dialog 'Parameter'
(3)changed output4&5 for PWM function
******************************************************************************

ver2.5.0.97, date released:2012-10-29
```

```
********************************************************************************
(1)fixt schedule function
(2)fixt temperature slider for tstat5G
********************************************************************************


ver2.5.0.96, date released:2012-10-25
********************************************************************************
(1)updated  outputs function
(2)fixt temperature slider
********************************************************************************


ver2.5.0.95, date released:2012-10-19
********************************************************************************
(1)updated all for PID2 in AUTO mode in off mode
(2)fixt temperature slider


********************************************************************************


ver2.5.0.94, date released:2012-10-17
********************************************************************************
(1)updated all for PID2 in AUTO mode
(2)fixt output4&5 functions in Output Table
(3)update tstat6 for Humidity
********************************************************************************



ver2.5.0.93, date released:2012-10-15
********************************************************************************
(1)fixt Output Table-Out6&7 for PID2
(2)fixt slider control
********************************************************************************



ver2.5.0.92, date released:2012-10-8
********************************************************************************
(1)fixed "value for "PID2 off set point" shows wrong and not auto updated"for TSTAT5G
(2)fixed Output Table-Out6 and Out7
********************************************************************************



ver2.5.0.91, date released:2012-9-28

********************************************************************************
(1)fixed "Output 4 and 5"
(2)fixed PID.
********************************************************************************



ver2.5.0.90, date released:2012-9-19

********************************************************************************
(1)Add ISPTOOL(TOOl->ISPTOOL)
(2)fixed Version History.
********************************************************************************



ver2.5.0.89, date released:2012-9-14

********************************************************************************
```

Update TSTAT5G
Update item:
(1)Input value" for Loop2.
(2)fixed deadband and night setpoint.
****

ver2.5.0.88, date released:2012-9-13

****
Update TSTAT5E and Tstat6
Update item:
(1)tstat6&7,Slider control,cooling and heating exchange like TSTAT5E.
(2)Output Grid Dialog box,enable the column 'Control'.
(3)show value for Input value.
****

ver2.5.0.87, date released:2012-9-12

****
Parameter Dialog box add Maximization.
****

ver2.5.0.86, date released:2012-9-10

****
Fixed Input "value" for Loop1.
****

ver2.5.0.85, date released:2012-9-5

****
Update TSTAT5E
Slider control and others etc.
****

ver2.5.0.84, date released:2012-8-20

****
when user operation 'Building config Database' pop 'running time error!'
****

ver2.5.0.83, date released:2012-8-15

****
the file is just for CM5 and LC merged update
****

ver2.5.0.82, date released:2012-8-10

****
Update TSTAT5E

enable for input3 to input8

****

DEMO PROGRAM

1 .Download the configuration file called [TBD]

2. Read the readme file .

    1>you should put the image fold to the installation path to replace the image file fold.

      ..\Database\Buildings\Default_Building\image
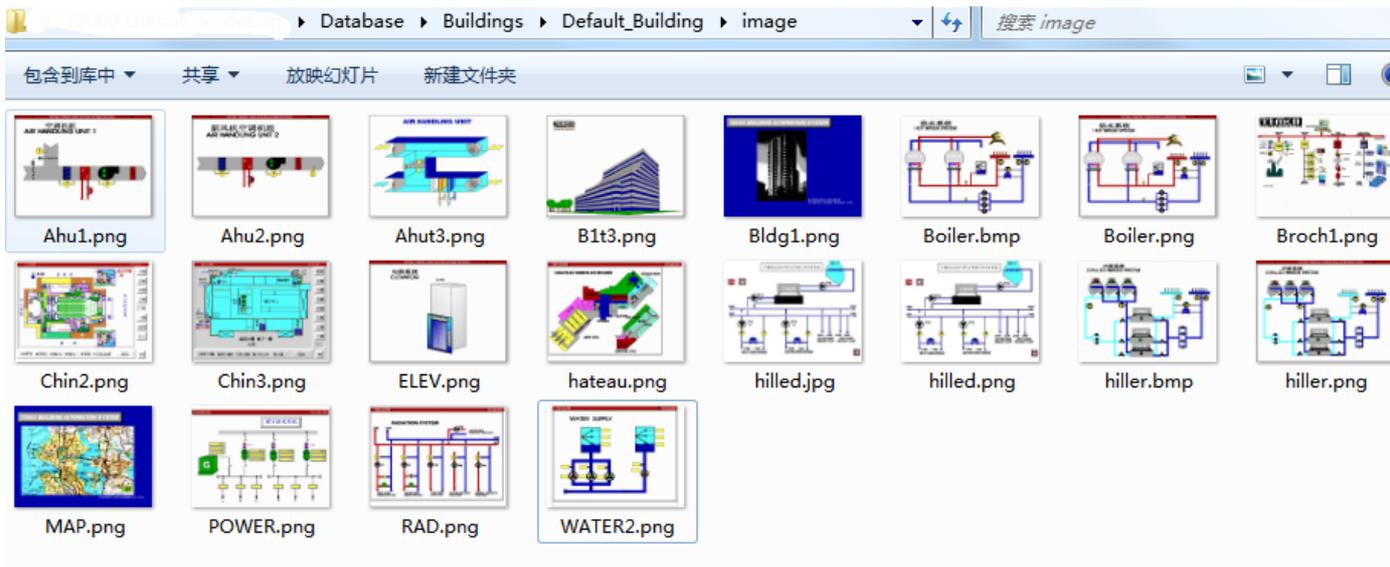
    2>you should open T3000 to load the T3000DEMOSHOW.prg to your T3 controllers.

      2.1.select your T3 controllers
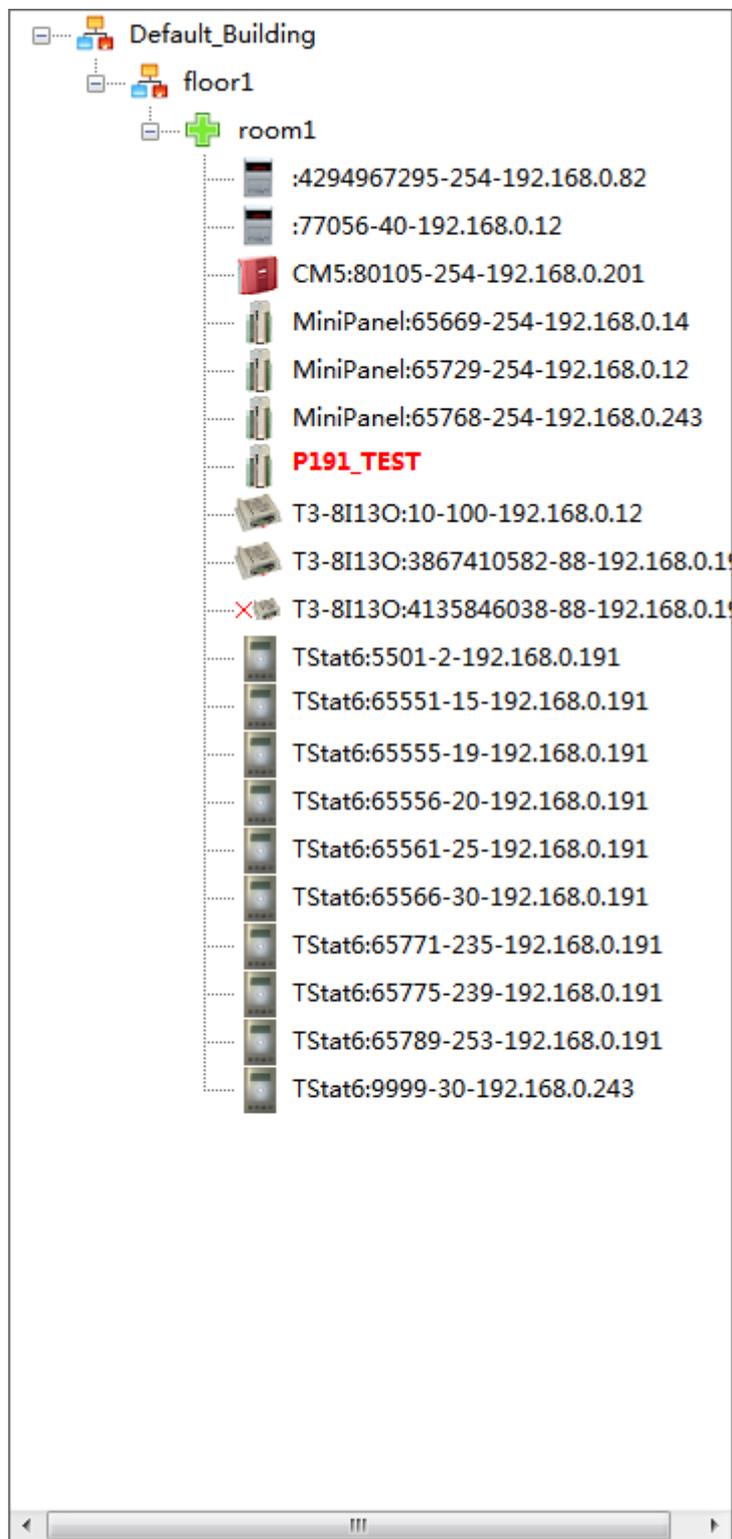
2.2.select the menu
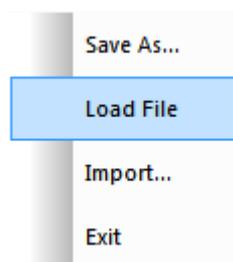
2.3.select the demo project file
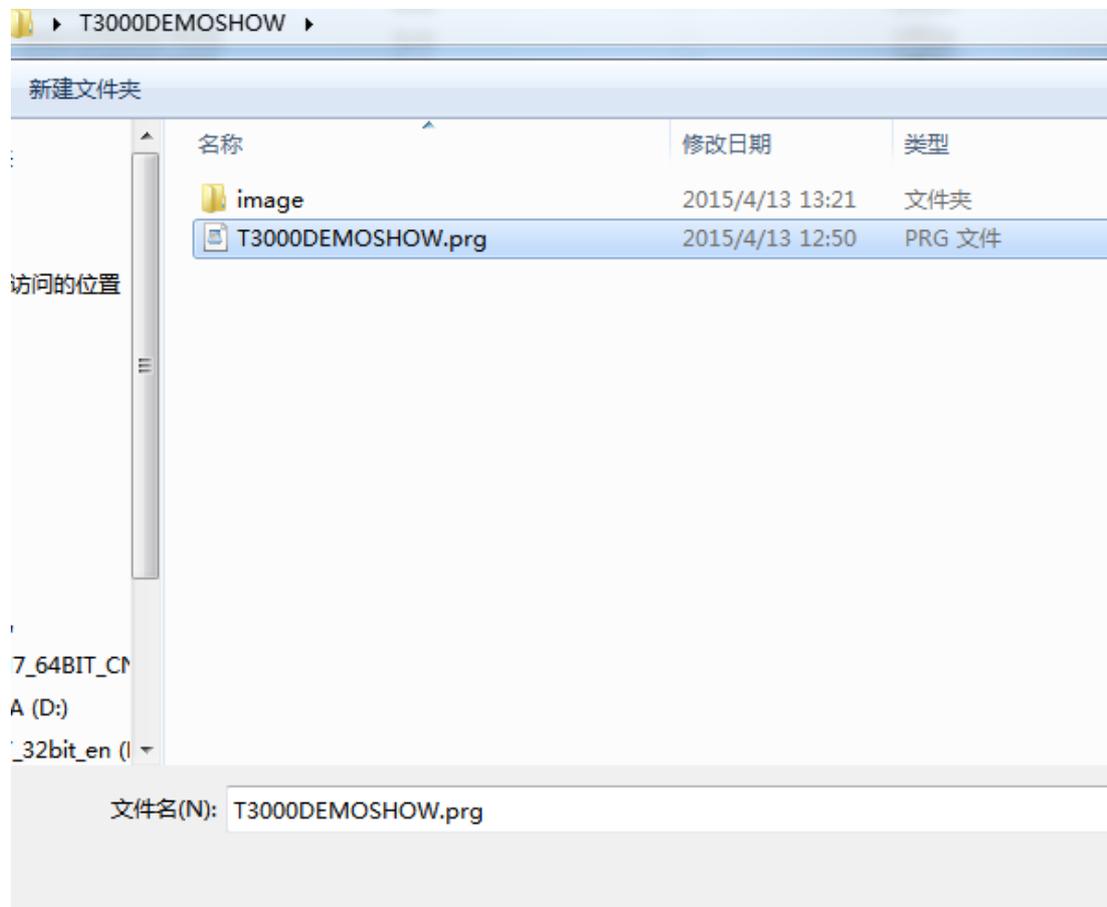
2.4.If finished ,please check these data .

        Check these data [Inputs ,Outputs ,Variables, Programs, PID Loops,
Graphics ,Schedules, Holidays, Trend Logs]

DEMO SHOW

>>>INPUTS

| Input	Panel	Full Label	Auto/Manual	Value	Units	Range	Calibration	Sign	Filter	Status	Jumper	Label
IN1	1	SUPPLY AIR TEMP	Auto	-40.00	Deg.F	10K -40 to 250	0.0	-	5	Open	Thermistor Dry...	AHU1SAT
IN2	1	RETURN AIR TEMP	Auto	248.00	Deg.F	10K -40 to 250	0.0	-	5	Shorted	Thermistor Dry...	AHU1RAT
IN3	1	MIXED AIR TEMP	Auto	-40.00	Deg.F	10K -40 to 250	0.0	-	5	Open	Thermistor Dry...	AHUMAT
IN4	1	ROOM TEMP	Auto	-40.00	Deg.F	10K -40 to 250	0.0	-	5	Open	Thermistor Dry...	AHU1RT
IN5	1	FAN STATUS	Auto	On		Off/On		-	5	Normal	Thermistor Dry...	AHU1FS
IN6	1	OUTSIDE AIR TEMP	Auto	-40.00	Deg.F	10K -40 to 250	0.0	-	5	Open	Thermistor Dry...	OAT
IN7	1	IN 7	Auto	2.99		Unused	0.0	-	5	Normal	Thermistor Dry...	IN7
IN8	1	IN 8	Auto	3.00		Unused	0.0	-	5	Normal	Thermistor Dry...	IN8
IN9	1	IN 9	Auto	3.00		Unused	0.0	-	5	Normal	Thermistor Dry...	IN9
IN10	1	IN 10	Auto	3.00		Unused	0.0	-	5	Normal	Thermistor Dry...	IN10
IN11	1	IN 11	Auto	3.00		Unused	0.0	-	5	Normal	Thermistor Dry...	IN11
IN12	1	IN 12	Auto	3.00		Unused	0.0	-	5	Normal	Thermistor Dry...	IN12
IN13	1	IN 13	Auto	3.00		Unused	0.0	-	5	Normal	Thermistor Dry...	IN13
IN14	1	IN 14	Auto	3.00		Unused	0.0	-	5	Normal	Thermistor Dry...	IN14
IN15	1	IN 15	Auto	3.00		Unused	0.0	-	5	Normal	Thermistor Dry...	IN15
IN16	1	IN 16	Auto	3.00		Unused	0.0	-	5	Normal	Thermistor Dry...	IN16
IN17	1	IN 17	Auto	3.00		Unused	0.0	-	5	Normal	Thermistor Dry...	IN17
IN18	1	IN 18	Auto	0.00		Unused	0.0	-	5	Normal	Thermistor Dry...	IN18
IN19	1	IN 19	Auto	3.00		Unused	0.0	-	5	Normal	Thermistor Dry...	IN19
IN20	1	IN 20	Auto	3.00		Unused	0.0	-	5	Normal	Thermistor Dry...	IN20
IN21	1	IN 21	Auto	3.00		Unused	0.0	-	5	Normal	Thermistor Dry...	IN21
IN22	1	IN 22	Auto	3.00		Unused	0.0	-	5	Normal	Thermistor Dry...	IN22
IN23	1	IN 23	Auto	3.00		Unused	0.0	-	5	Normal	Thermistor Dry...	IN23
IN24	1	IN 24	Auto	3.00		Unused	0.0	-	5	Normal	Thermistor Dry...	IN24
IN25	1	IN 25	Auto	3.00		Unused	0.0	-	5	Normal	Thermistor Dry...	IN25
IN26	1	IN 26	Auto	3.00		Unused	0.0	-	5	Normal	Thermistor Dry...	IN26
IN27	1	IN 27	Auto	3.00		Unused	0.0	-	5	Normal	Thermistor Dry...	IN27
IN28	1	IN 28	Auto	3.00		Unused	0.0	-	5	Normal	Thermistor Dry...	IN28
IN29	1	IN 29	Auto	3.00		Unused	0.0	-	5	Normal	Thermistor Dry...	IN29
IN30	1	IN 30	Auto	3.00		Unused	0.0	-	5	Normal	Thermistor Dry...	IN30
IN31	1	IN 31	Auto	3.00		Unused	0.0	-	5	Normal	Thermistor Dry...	IN31
IN32	1	IN 32	Auto	3.00		Unused	0.0	-	5	Normal	Thermistor Dry...	IN32
IN33	1	IN 33	Auto	0.00		Unused	0.0	-	5	Normal	Thermistor Dry...	IN33
IN34	1	IN 34	Auto	0.00		Unused	0.0	-	5	Normal	Thermistor Dry...	IN34
IN35	1	IN 35	Auto	0.00		Unused	0.0	-	5	Normal	Thermistor Dry...	IN35
IN36	1	IN 36	Auto	0.00		Unused	0.0	-	5	Normal	Thermistor Dry...	IN36
IN37	1	IN 37	Auto	0.00		Unused	0.0	-	5	Normal	Thermistor Dry...	IN37
IN38	1	IN 38	Auto	0.00		Unused	0.0	-	5	Normal	Thermistor Dry...	IN38
IN39	1	IN 39	Auto	0.00		Unused	0.0	-	5	Normal	Thermistor Dry...	IN39
IN40	1	IN 40	Auto	0.00		Unused	0.0	-	5	Normal	Thermistor Dry...	IN40
IN41	1	IN 41	Auto	0.00		Unused	0.0	-	5	Normal	Thermistor Dry...	IN41
IN42	1	IN 42	Auto	0.00		Unused	0.0	-	5	Normal	Thermistor Dry...	IN42
IN43	1	IN 43	Auto	0.00		Unused	0.0	-	5	Normal	Thermistor Dry...	IN43
IN44	1	IN 44	Auto	0.00		Unused	0.0	-	5	Normal	Thermistor Dry...	IN44
IN45	1	IN 45	Auto	0.00		Unused	0.0	-	5	Normal	Thermistor Dry...	IN45
IN46	1	IN 46	Auto	0.00		Unused	0.0	-	5	Normal	Thermistor Dry...	IN46
IN47	1	IN 47	Auto	0.00		Unused	0.0	-	5	Normal	Thermistor Dry...	IN47
IN48	1	IN 48	Auto	0.00		Unused	0.0	-	5	Shorted	Thermistor Dry...	IN48
IN49	1	IN 49	Auto	0.00		Unused	0.0	-	5	Normal	Thermistor Dry...	IN49
IN50	1	IN 50	Auto	0.00		Unused	0.0	-	5	Normal	Thermistor Dry...	IN50
IN51	1	IN 51	Auto	0.00		Unused	0.0	-	5	Normal	Thermistor Dry...	IN51
IN52	1	IN 52	Auto	0.00		Unused	0.0	-	5	Normal	Thermistor Dry...	IN52
IN53	1	IN 53	Auto	0.00		Unused	0.0	-	5	Normal	Thermistor Dry...	IN53
IN54	1	IN 54	Auto	0.00		Unused	0.0	-	5	Normal	Thermistor Dry...	IN54
IN55	1	IN 55	Auto	0.00		Unused	0.0	-	5	Normal	Thermistor Dry...	IN55
IN56	1	IN 56	Auto	0.00		Unused	0.0	-	5	Normal	Thermistor Dry...	IN56
IN57	1	IN 57	Auto	0.00		Unused	0.0	-	5	Normal	Thermistor Dry...	IN57
IN58	1	IN 58	Auto	0.00		Unused	0.0	-	5	Normal	Thermistor Dry...	IN58
IN59	1	IN 59	Auto	0.00		Unused	0.0	-	5	Normal	Thermistor Dry...	IN59
IN60	1	IN 60	Auto	0.00		Unused	0.0	-	5	Normal	Thermistor Dry...	IN60
IN61	1	IN 61	Auto	0.00		Unused	0.0	-	5	Normal	Thermistor Dry...	IN61
IN62	1	IN 62	Auto	0.00		Unused	0.0	-	5	Normal	Thermistor Dry...	IN62
IN63	1	IN 63	Auto	0.00		Unused	0.0	-	5	Normal	Thermistor Dry...	IN63
IN64	1	IN 64	Auto	0.00		Unused	0.0	-	5	Normal	Thermistor Dry...	IN64

>>>OUTPUTS

Output	Panel	Full Label	Auto/Manual	HOA Switch	Value	Units	Range	PWM Period	Status	Label
OUT1	1	AHU1 FAN	Auto	AUTO	Off		Off/On	0	OK	AHU1FAN
OUT2	1	AHU1 COOLING	Auto	AUTO	Off		Off/On	0	-	AHU1COOL
OUT3	1	AHU1 HEATING	Auto	AUTO	Off		Off/On	0	OK	AHU1HEAT
OUT4	1	AHU1 MIX ᴸ?	Auto	AUTO	Off		Off/On	1	OK	
OUT5	1	┼	Manual	AUTO	Off		Off/On	32	OK	
OUT6	1		Auto	AUTO	0.00		Unused	0	OK	OUT6
OUT7	1	OUT7	Auto	AUTO	0.00		Unused	0	OK	OUT7
OUT8	1	OUT8	Auto	AUTO	0.00		Unused	0	OK	OUT8
OUT9	1	OUT9	Auto	AUTO	Off		Unused	28	OK	OUT9
OUT10	1	E	Auto	AUTO	Off		Unused	65	OK	?
OUT11	1		Auto	AUTO	0.00		Unused	0	OK	ACACAAA
OUT12	1	OUT12	Auto	AUTO	0.00		Unused	0	OK	OUT12
OUT13	1	OUT13	Auto	AUTO	0.00		Unused	0	OK	OUT13
OUT14	1	OUT14	Auto	AUTO	0.00		Unused	0	OK	OUT14
OUT15	1	OUT15	Manual	AUTO	Off		Off/On	168	OK	OUT1 ᴸ?
OUT16	1		Manual	AUTO	Off		Off/On	65	OK	
OUT17	1	CACACACACACAAA	Auto	AUTO	0.00		Unused	0	OK	OUT17
OUT18	1	OUT18	Auto	AUTO	0.00		Unused	0	OK	OUT18
OUT19	1	OUT19	Auto	AUTO	0.00		Unused	0	OK	OUT19
OUT20	1	OUT20	Auto	AUTO	0.00		Unused	0	OK	OUT20
OUT21	1	OUT21	Manual	AUTO	0.00		Unused	11	OK	g@
OUT22	1	?	Auto	AUTO	0.00		Unused	0	OK	
OUT23	1	OUT23	Auto	AUTO	0.00		Unused	0	OK	OUT23
OUT24	1	OUT24	Auto	AUTO	0.00		Unused	0	OK	OUT24
OUT25	1	OUT25	Auto	AUTO	0.00		Unused	0	-	OUT25
OUT26	1	OUT26	Auto	AUTO	0.00		Unused	0	-	OUT26
OUT27	1	6€	Manual	AUTO	688.38	Unused	Out of range	25	-	
OUT28	1	.e••	Auto	AUTO	0.00		Unused	0	-	OUT28
OUT29	1	OUT29	Auto	AUTO	0.00		Unused	0	-	OUT29
OUT30	1	OUT30	Auto	AUTO	0.00		Unused	0	-	OUT30
OUT31	1	OUT31	Auto	AUTO	0.00		Unused	0	-	OUT31
OUT32	1	OUT32	Manual	AUTO	0.00		Unused	17	-	OUT32
OUT33	1	g括	Manual	AUTO	0.00		Unused	0	-	⌟
OUT34	1	OUT34	Auto	AUTO	0.00		Unused	0	-	OUT34
OUT35	1	OUT35	Auto	AUTO	0.00		Unused	0	-	OUT35
OUT36	1	OUT36	Auto	AUTO	0.00		Unused	0	-	OUT36
OUT37	1	OUT37	Auto	AUTO	0.00		Unused	0	-	OUT37
OUT38	1	OUT38	Auto	AUTO	0.00		Unused	0	-	
OUT39	1	"孈?	Auto	AUTO	0.00		Unused	0	-	•
OUT40	1	OUT40	Auto	AUTO	0.00		Unused	0	-	OUT40
OUT41	1	OUT41	Auto	AUTO	0.00		Unused	0	-	OUT41
OUT42	1	OUT42	Auto	AUTO	0.00		Unused	0	-	OUT42
OUT43	1	OUT43	Auto	AUTO	0.00		Unused	0	-	OUT43
OUT44	1	ᴸ?	Manual	AUTO	0.00		Unused	10	-	t括
OUT45	1	⌟┼ .e••	Auto	AUTO	0.00		Unused	0	-	OUT45
OUT46	1	OUT46	Auto	AUTO	0.00		Unused	0	-	OUT46
OUT47	1	OUT47	Auto	AUTO	0.00		Unused	0	-	OUT47
OUT48	1	OUT48	Auto	AUTO	0.00		Unused	0	-	OUT48
OUT49	1	OUT49	Manual	AUTO	0.00		Unused	0	-	OUT49
OUT50	1	€↵ ?	Auto	AUTO	917.67		Unused	0	-	括
OUT51	1	OUT51	Auto	AUTO	0.00		Unused	0	-	OUT51
OUT52	1	OUT52	Auto	AUTO	0.00		Unused	0	-	OUT52
OUT53	1	OUT53	Auto	AUTO	0.00		Unused	0	-	OUT53
OUT54	1	OUT54	Auto	AUTO	0.00		Unused	0	-	OUT54
OUT55	1	OUT55	Manual	AUTO	0.00		Unused	192	-	€
OUT56	1	豪	Auto	AUTO	12.03		Unused	0	-	_e••
OUT57	1	OUT57	Auto	AUTO	0.00		Unused	0	-	OUT57
OUT58	1	OUT58	Auto	AUTO	0.00		Unused	0	-	OUT58
OUT59	1	OUT59	Auto	AUTO	0.00		Unused	0	-	OUT59
OUT60	1	OUT60	Auto	AUTO	0.00		Unused	0	-	OUT60
OUT61	1	OUT ᴸ?	Manual	AUTO	0.00		Unused	0	-	括
OUT62	1		Manual	AUTO	0.00		Unused	0	-	
OUT63	1	OUT63	Auto	AUTO	0.00		Unused	0	-	OUT63
OUT64	1	OUT64	Auto	AUTO	0.00		Unused	0	-	OUT64

>>>VARIABLES

Variable	Full Label	Auto/Manual	Value	Units	Label
VAR1	AHU1 HEATING SETPT	Auto	0.000	Deg.C	AHU1HSP
VAR2	AHU1 COOLING SETPT	Auto	0.000	Deg.C	AHU1CSP
VAR3	AHU1 HEATING MODE	Auto	Off	Off/On	HEATMODE
VAR4	AHU1 COOLING MODE	Auto	On	Off/On	COOLMODE
VAR5	AHU1 OVERIDE MODE	Auto	Off	Off/On	AHU1OM
VAR6	AHU1 DAYTIME SETPT	Manual	0.000	Deg.C	AHU1DSP
VAR7	AHUf FREECOOL AVAIL	Auto	Off	Off/On	AHU1COOL
VAR8	AHU1 SETPOINT ADJUST	Manual	0.000	Deg.C	AHU1SPA
VAR9	AHU1 SETPOINT	Manual	0.000	Deg.C	AHU1SP
VAR10	AHU2 HEATING SETPT	Auto	2.000	Deg.C	AHU2HSP
VAR11	AHU2 COOLING SETPT	Auto	0.000	Deg.C	AHU2CSP
VAR12	AHU2 HAEATING MODE	Auto	Off	Off/On	HEATMOD2
VAR13	AHU2 COOLING MODE	Auto	Off	Off/On	COOLMOD2
VAR14	AHU2 OVERIDE MODE	Auto	Off	Off/On	AHU2OM
VAR15	AHU2 DAYTIME SETPT	Auto	0.000	Deg.C	AHU2DSP
VAR16	AHU2 SETPOINT	Manual	0.000	Deg.C	AHU2SP
VAR17	DAYTIME	Auto	0.000	Unused	DAYTIMER
VAR18	OCCUPIED MODE	Manual	Off	Off/On	OCCUPIED
VAR19	NIGHTTIME SETPOINT	Manual	0.000	Deg.C	NITESP
VAR20	TIMEBASE	Manual	0.000	Counts	TIMEBASE
VAR21	BOILER SUPPLY SETPT	Auto	0.000	Deg.C	BADSETP
VAR22	BOILER SETPT @ 0DEGC	Manual	0.000	Deg.C	BLRSP0
VAR23	AHU2 SETPT ADJUST	Auto	0.000	Deg.C	
VAR24	RTS1 TEMP CHANGE	Auto	0.000	Deg.C	RTSC1
VAR25	LOSS1 THERMAL	Auto	0.000	Unused	LOSS1
VAR26	LOSS2 AIR FLOW	Auto	0.000	Unused	LOSS2
VAR27	ROOM1 AIRFLOW	Manual	0.000	Unused	AIRFLOW1
VAR28	GAIN1 SOLAR HEAT	Auto	0.000	Unused	GAIN1
VAR29	GAIN2 AIR FLOW	Auto	0.000	Unused	GAIN2
VAR30	GAIN3 INTERNAL HEAT	Auto	0.000	Unused	GAIN3
VAR31	AHU1 WARMUP MODE	Manual	Off	Off/On	AHU1WUM
VAR32	TOTAL HEAT GAIN	Auto	0.000	Unused	GAIN
VAR33	TOTAL HEAT LOSS	Auto	0.000	Unused	LOSS
VAR34	NET ENERGY 1	Auto	0.000	Unused	ENERGY1
VAR35	SUPPLY TEMP RISE 1	Auto	0.000	Unused	RISE1
VAR36	RTS2 TEMP CHANGE	Auto	0.000	Unused	RTSC2
VAR37	NET ENERGY 2	Auto	0.000	Unused	ENERGY2
VAR38	SUPPLY TEMP RISE2	Auto	0.000	Deg.C	RISE2
VAR39	DAY OF YEAR	Auto	0.000	Days	DOY
VAR40	OAT ADJUST	Auto	0.000	Unused	OATADJ1
VAR41	AHU1 WARMUP MODE	Auto	Off	Off/On	WARMUP1
VAR42	OAT ADJUST 2	Auto	0.000	Deg.C	OATADJ2
VAR43	AHU2 AIRFLOW	Auto	0.000	Unused	AIRFLOW2
VAR44	LOSS1 THERMAL	Auto	0.000	Unused	LOSS1_2
VAR45	LOSS2 AIR FLOWL	Auto	0.000	Unused	LOSS2_2
VAR46	DOM HOTWATER SETPT	Manual	0.000	Deg.C	DHWSP
VAR47	GAIN1 SOLAR	Auto	0.000	Unused	GAIN1_2
VAR48	GAIN2 AIRFLOW	Auto	0.000	Unused	GAIN2_2
VAR49	GAIN3 INTERNAL	Auto	0.000	Unused	GAIN3_I
VAR50	TOTAL HEAT LOSS2	Auto	0.000	Unused	T_LOSS2
VAR51	TOTAL HEAT GAIN2	Auto	0.000	Unused	T_GAIN2
VAR52	HEAT ENERGY ADDED1	Auto	0.000	Unused	HEATENR1
VAR53	HEAT ENERGY ADDED2	Auto	0.000	Unused	HEATENR2
VAR54	FAN ENERGY ADDED	Auto	0.000	Unused	FANENERG
VAR55	COOL ENERGY ADDED1	Auto	0.000	Unused	COOLENR1
VAR56	COOL ENERGY ADDED2	Auto	0.000	Unused	COOLENR2
VAR57	ENERGY BILL SCAN1	Auto	0.000	Unused	EBILL1
VAR58	RATE OF ENERGY	Auto	0.000	Counts	ERATE
VAR59	ENERGY BILL SCAN2	Auto	0.000	Unused	EBILL2
VAR60	DOM HOTWAT SETPOINT2	Auto	0.000	Deg.C	DHWSP2
VAR61		Auto			
VAR62		Auto			
VAR63		Auto			
VAR64		Auto			
VAR65		Auto			
VAR66		Auto			
VAR67		Auto			
VAR68		Auto			
VAR69		Auto			
VAR70		Auto			
VAR71		Auto			
VAR72		Auto			
VAR73		Auto			
VAR74		Auto			
VAR75		Auto			
VAR76		Auto			
VAR77		Auto			
VAR78		Auto			
VAR79		Auto			
VAR80		Auto			
VAR81		Auto			
VAR82		Auto			
VAR83		Auto			
VAR84		Auto			
VAR85		Auto			
VAR86		Auto			
VAR87		Auto			
VAR88		Auto			
VAR89		Auto			
VAR90		Auto			
VAR91		Auto			
VAR92		Auto			
VAR93		Auto			
VAR94		Auto			
VAR95		Auto			
VAR96		Auto			
VAR97		Auto			
VAR98		Auto			
VAR99		Auto			
VAR100		Auto			
VAR101		Auto			
VAR102		Auto			
VAR103		Auto			
VAR104		Auto			
VAR105		Auto			
VAR106		Auto			
VAR107		Auto			
VAR108		Auto			
VAR109		Auto			
VAR110		Auto			
VAR111		Auto			
VAR112		Auto			
VAR113		Auto			
VAR114		Auto			
VAR115		Auto			
VAR116		Auto			
VAR117		Auto			
VAR118		Auto			
VAR119		Auto			
VAR120		Auto			
VAR121		Auto			
VAR122		Auto			
VAR123		Auto			
VAR124		Auto			
VAR125		Auto			
VAR126		Auto			
VAR127		Auto			
VAR128		Auto			

>>>Programs

Num	Full Label	Status	Auto/Manual	Size	Run Status	Label	
☑ 1	OCCUPIED PROG	ON	Auto	82	Normal	OPRG	
☐ 2	AHU1 PROGRAM	ON	Auto	5	Normal	APRG	
☐	AHU1 SIMULATOR	ON	Auto	0	Normal	ASMLT	
☐ 4	CALENDAR SIMULATOR	ON	Auto	0	Normal	CSMLT	
☐ 5	PERFORMANCE CALCS	ON	Auto	0	Normal	PER	
☐ 6	BOILER PROGRAM	ON	Auto	0	Normal	BPGG	
☐ 7	BOILER ROTATE RPOG	ON	Auto	0	Normal	BRPRG	
☐ 8	CHILLER PROGRAM	ON	Auto	0	Normal	CP	
☐ 9	COOLING TOWER PROGRM	ON	Auto	0	Normal	CTPRG	
☐ 10	COOLING TOWER SIMULT	ON	Auto	0	Normal	FSPRG	
☐ 11	FIRE SYSTEM PROGRAM	ON	Auto	0	Normal	FSSMLT	
☐ 12	FIRE SYSTEM SIMULATR	ON	Auto	0	Normal	BSMLTPRG	
☐ 13	BILER SIMULATOR PRG	ON	Auto	0	Normal	WSPRG	
☐ 14	WATER SUPPLY PROGRAM	ON	Auto	0	Normal	WSSMLT	
☐ 15	WATER SUPPLY SIMULAT	ON	Auto	0	Normal	WSP	
☐ 16	bid test	ON	Auto	84	Normal	SWS	

Insert   Home   Page Up

Delete   End   Page Down

Programming

Bacnet Program IDE

Send (F2)   Clear (F3)   Load File (F7)   Save File (F6)   Refresh (F8)

```
10 REM ***** FAN ****
20 IF WR1 THEN START SPARE ELSE STOP SPARE
30 REM *********** VALVE ******
40  SPARE = CON1
```

>>>PID Loops

NUM	Input	Value	Units	A/M	Output	Setpoint	Set Value	Units	Action	Prop	Int	Der	Bias
1				Auto	0.0%				-	0	0	0.00	0
2				Auto	0.0%				-	0	0	0.00	0
3				Auto	0.0%				-	0	0	0.00	0
4				Auto	0.0%				-	0	0	0.00	0
5				Auto	0.0%				-	0	0	0.00	0
6				Auto	0.0%				-	0	0	0.00	0
7				Auto	0.0%				-	0	0	0.00	0
8				Auto	0.0%				-	0	0	0.00	0
9				Auto	0.0%				-	0	0	0.00	0
10				Auto	0.0%				-	0	0	0.00	0
11				Auto	0.0%				-	0	0	0.00	0
12				Auto	0.0%				-	0	0	0.00	0
13				Auto	0.0%				-	0	0	0.00	0
14				Auto	0.0%				-	0	0	0.00	0
15				Auto	0.0%				-	0	0	0.00	0
16				Auto	0.0%				-	0	0	0.00	0

>>>Graphics

NUM	Full Label	Label	Picture File	Mode	Refresh Rate
1	MAIN MENU	M	B1t3.png	Graphic	15
2	AHU2 DISPLAY	AHU2	Ahut3.png	Graphic	
3	BOILER		Boiler.png		
4	WATER MONITORING	W	WATER.p		
5	AIR HANDLING UNIT 1	1	Ahu1.png		
6	CHATEAU WHISTLER	CHAT	hateau.png		
7	TAIWAN NATNL THEATRE	T	Chin2.png		
8	POWER MONITORING	P	POWER.png		
9	AIR HANDLING UNIT2	2	Ahu2.png		
10	FIRE ALARM SYSTEM	F	Chin2.png	Graphic	15
11	NETWORK LAYOUT	NET	Broch1.png	Graphic	15
12	MAP TO OTHER SITES	MAP	MAP.png	Graphic	15
13	KIWANIS MANOR	BLD2	Bldg1.png	Graphic	15
14	LIGHTING DISPLAY	L	Chin3.png	Graphic	15
15	RADIATION HEATING	RAD	RAD.png	Graphic	15
16	CHILLED WATER SYSTEM	CH	hiller.png	Graphic	15

Insert   Home   Page Up

Delete   End   Page Down

enter

MAP TO OTHER SITES

enter

CH

B

AHU2

T3000 BUILDING AUTOMATION SYSTEM

TEMCO CONTROLS LTD

TRICOM CONTROLS T3000 BUILDING AUTOMATION SYSTEM

空调机组
AIR HANDLING UNIT 1
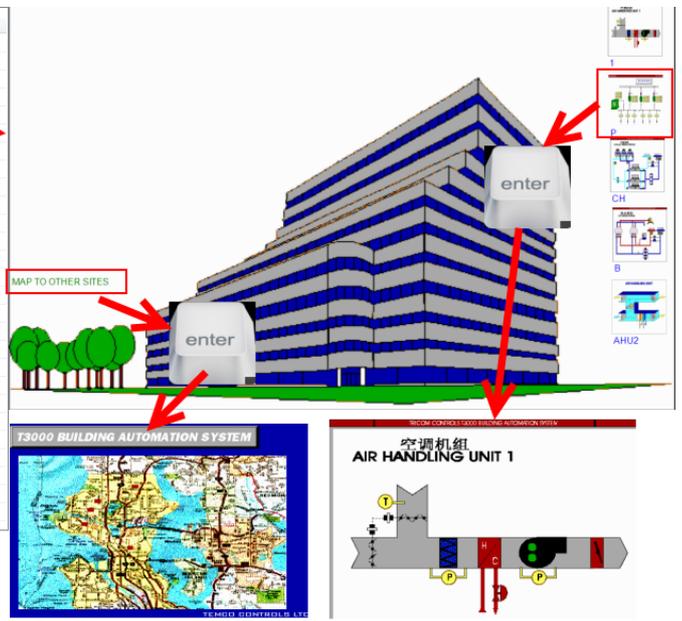
FIRMWARE UPDATE, SINGLE DEVICE

Using T3 controllers as an example.

1> Download the latest firmware from temcocontrols.com ,or paste "http://www.temcocontrols.com/ftp/firmware/T3_BB_LB_TB/" to your browser .

Select the version that you want to update to download .

If you get the latest firmware from temco's engineer, don't need the step 1.

2> If your computer didn't install the T3000 ,please download   T3000Software.zip .

If you have installed the lastest T3000,don't need the step2.

3> Click "Load firmware for a single device" in Tools menu.
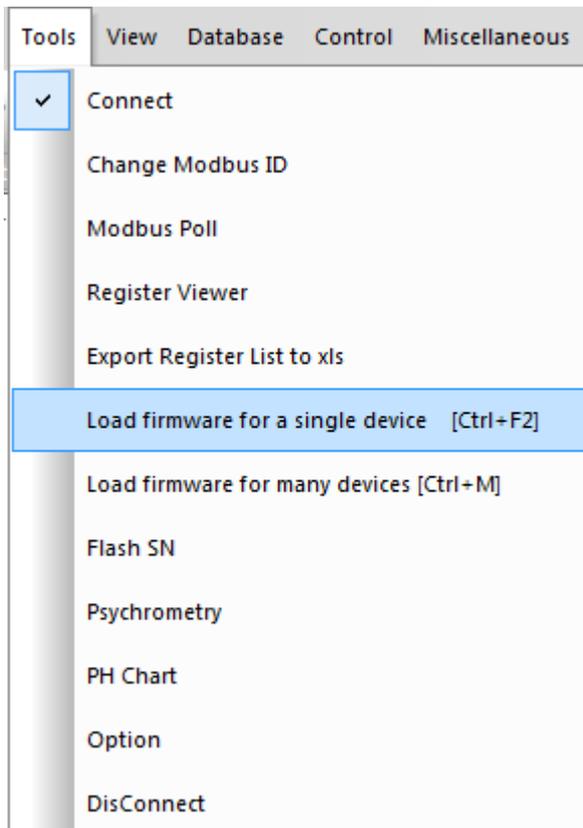
and then it shows a window.
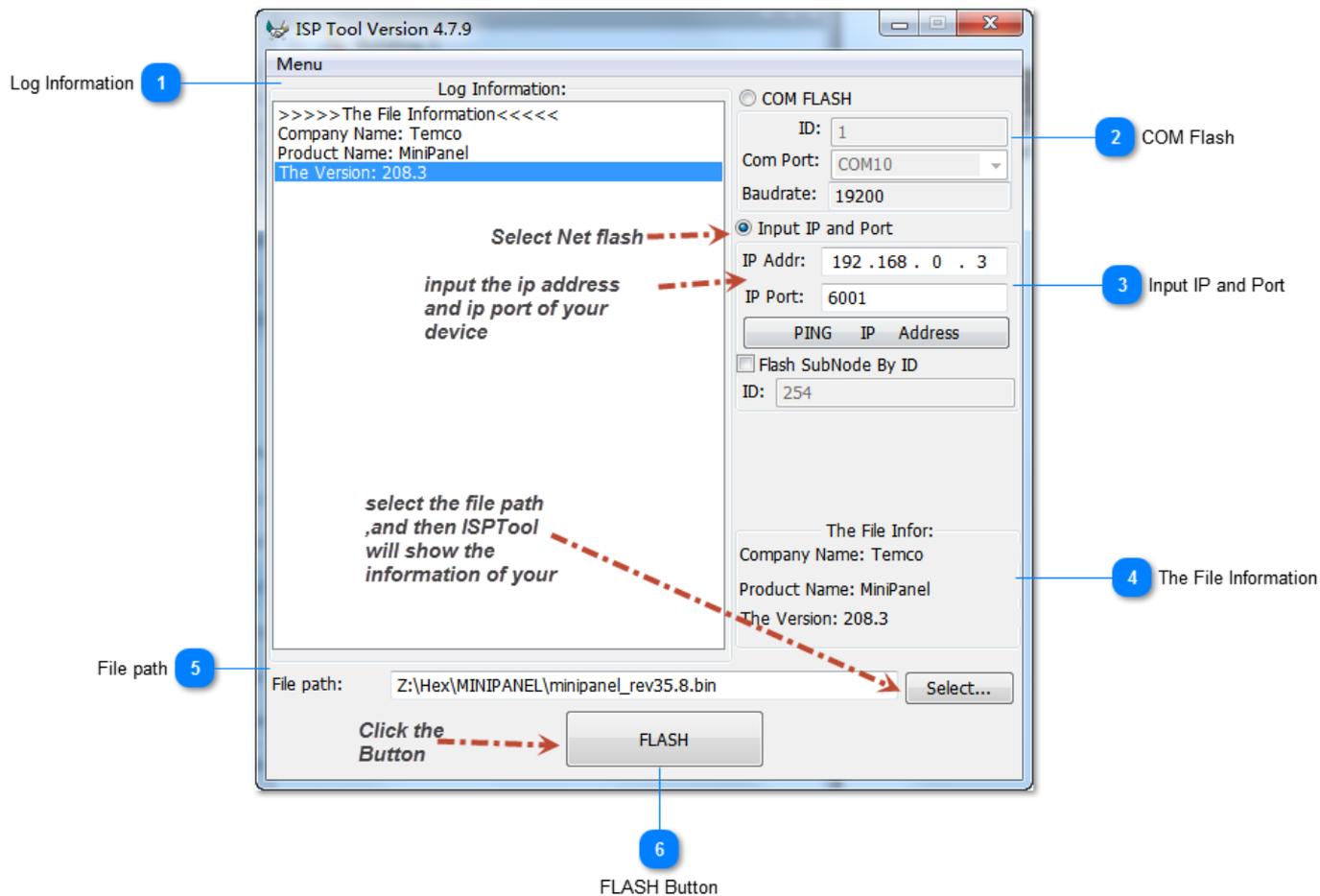
If you want to update other device ,using Com port ,please select "COM FLASH"
And then input the Modbus ID of your device .If you don't know your Modbus ID ,you
can try flash the device ,using the 255.
**select your Com Port .**
and then just click the button of "FLASH"

notes: If you don't know the ip address and ip port of the device, don't care about
**the two parameters .**
ISPTool will show the right message in the blank area ,including the ip and ip
port.
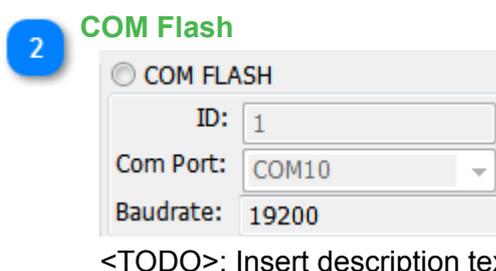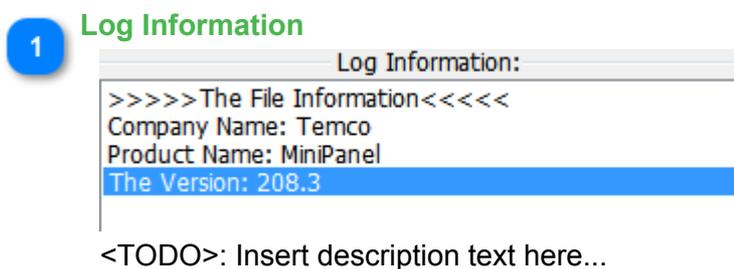
Log Information

<TODO>: Insert description text here...

COM Flash

<TODO>: Insert description text here...

Input IP and Port

<TODO>: Insert description text here...

The File Information

The File Infor:

Company Name: Temco

Product Name: MiniPanel

The Version: 208.3

<TODO>: Insert description text here...

File path

File path: Z:\Hex\MINIPANEL\minipanel_rev35.8.bin   Select...

<TODO>: Insert description text here...

FLASH Button

FLASH

<TODO>: Insert description text here...

FIRMWARE UPDATES: BATCH LOADING

1.You must find your devices by T3000 , using "Scan"

2 "Load firmware for many devices"

√	Connect
	Change Modbus ID
	Modubs Poll
	Register Viewer
	Registers Maintenance System
	Refresh Current Tstat(F8)
	Refresh Left Tree View(F3)
	Export Registers List
	Load firmware for a single device [Ctrl+F2]
	Load firmware for many devices [Ctrl+M]
	DisConnect
	S.N Repair Tool
	Flash SN

Load firmware for many devices

Hex or Bin File : 1 2 Apply To All Select 3 Start
The configure File : 4 5 ... Apply To All Select 6
7 □ Select/UnSelect 8 Reset 9 ☑ Flash the different version

NUM	SN	Panel ID	Name	Com	IP	Port	Is Sub	Is Online	Software Version	Firmware Version	Hex Bin File 11	Configuration file 12	Firmware
☑ 3	65555	19	TStat6	10			NO	Offline	0.0				
☑ 4	65776	240	TStat6	10			NO	Offline	0.0				
☑ 5	81474	252	TStat6	10			NO	Offline	0.0				
☑ 6	65540	253	TStat6	10			NO	Offline	0.0				
☑ 7	80105	254	CM5		192.168.0.201	10000	NO	Online	37.0	2.2	Z:\Hex\CM5\bootldr_cm5_rev22_HWREV11.bin		
☑ 8	80491	254	TStat6	10			NO	Offline	0.0				
☑ 9	65669	254	MiniPanel		192.168.0.14	10000	NO	Online	37.1	12.6	Z:\Hex\MINIPANEL\minipanel_rev36.9.bin		
☑ 10	65768	254	MiniPanel		192.168.0.243	10000	NO	Online	37.1	12.6	Z:\Hex\MINIPANEL\minipanel_rev36.9.bin		
☑ 11	-1	254	MiniPanel		192.168.0.212	10000	NO	Online	37.1				
☑ 12	77992	254	CM5		192.168.0.48	10000	NO	Offline	0.0				
☑ 13	1000	254	MiniPanel		192.168.0.113	10000	NO	Offline	0.0				
☑ 14	11345	254	MiniPanel		192.168.0.251	10000	NO	Online	37.1	12.6	Z:\Hex\MINIPANEL\minipanel_rev36.9.bin		
☑ 15	73625	3	TStat5A	10			NO	Online	37.8				
☑ 16	65574	42	TStat6	10			NO	Offline	0.0	10			
☑ 17	51898	7	TStat5G	10			NO	Offline	0.0			Z:\Temp\T3000\Konferansrum PMC.txt	

1   show the path of a hex or bin file

2   click to select the path of a hex or bin file

3   "Apply To All Select" ,the path of the hex or bin(1) will automatically fill in the column of "Hex Bin File

4	show the path of a configuration file
5	click to select the path of a configuration file
6	"Apply To All Select" ,the path of the configuration file (4) will automatically fill in the column of "Con
7	Select/Unselect
8	Reset ,to clear the column of "Hex Bin File" and the column of "Configuration File"
9	if selected ,T3000 will compare the "Software Version" with "Firmware Version", Only if the Firmwar
10	When you show the window, the tree column will show the values ,automatically.
11	show the path of a hex or bin matching with the device. clicking the blank to select the file ,the file w
12	show the path of a configuration file matching with the device. clicking the blank to select the file ,th

14 How to use subnet

Description:

All subnets connected to MP can be used in the MP ,directly .

1-48-VAR 101

Description : This is a register value of the Modbus ID 48 connected to pannel ID 1
Panel ID =1
  Subnet Modbus ID =48
VAR101 : Register Number 101 Modbus

Format :

1-48-VAR101

Description : This is a register value of the Modbus ID 48 connected to pannel ID 1

Panel ID =1   (What's Panel ID ,please refer to Chapter 7 7.1.5)

Subnet Modbus ID =48

VAR101 : Register Number is 101

Notes : Register Number is PLC Address (Base 1). If you use Protocol Address (Base 0),

you must plus one to your current register number .

How to use in program:
EXAMPLE 1:GET THE VALUE OF THE REGISTER VALUE OF A SUBNET

10 REM *** THE VALUE OF 1-48-VAR101 ASSIGN TO VAR1 ***
20 VAR1 = 1-48-VAR101

EXAMPLE 1:SET THE VALUE OF THE REGISTER VALUE OF A SUBNET

10 REM *** VAR1 ASSIGN TO THE VALUE OF 1-48-VAR101 ***
20 1-48-VAR101 = VAR1

NOTE: When you use the register number ,you must know whether the register number can have the property of R/W.

Example show :

10 VAR1 = 1-48-VAR101
20 VAR2 = 1-48-VAR102
30 VAR3 = 1-47-VAR101
40 VAR4 = 1-47-VAR102
50 VAR5 = 1-20-VAR101
60 VAR6 = 1-20-VAR102
70 VAR7 = 1-19-VAR101
80 VAR8 = 1-19-VAR102
90 VAR9 = 1-44-VAR101
100 VAR10 = 1-44-VAR102
110 VAR11 = 1-46-VAR101
120 VAR12 = 1-46-VAR102
130 VAR13 = 1-49-VAR101
140 VAR14 = 1-49-VAR102

150 REM *** START COOLING PROCESS ***
160 IF VAR3 OR VAR5 OR VAR 7 OR VAR9 > 150 THEN START OUT1 , START OUT32
170 IF IN2 <= 160 THEN START OUT4, OUT5, OUT6, OUT7, OUT8, OUT9, OUT10, OUT11, OUT12, OUT13, OUT14
180 IF IN3 >= 700 THEN START OUT2
190 IF IN3 >= 850 THEN STOP OUT32 , STOP OUT2 , STOP OUT1
200 IF IN2 <= 20 THE STOP OUT32 , STOP OUT1
220 IF VAR3 = 100 THEN STOP OUT4 , STOP OUT11

230 IF VAR5 = 100 THEN STOP OUT5 , STOP OUT12
240 IF VAR7 = 100 THEN STOP OUT6 , STOP OUT13
250 IF VAR9 = 100 THEN STOP OUT7 , STOP OUT14

260 REM * START COOLING PROCESS ROOM1 & ROOM6 *****
270 IF REM VAR1 OR VAR11 OR VAR13 > 240 THEN START OUT1 , START OUT32
280 IF IN2 < 200 THEN START OUT3, OUT8, OUT9, OUT10, OUT15, OUT16
290 IF IN3 >= 700 THEN START OUT2
300 IF IN3 >= 900 THEN STOP OUT32 , THEN STOP OUT2 , STOP OUT1
310 IF IN2 <= 20 THE STOP OUT32 , STOP OUT1
320 IF VAR11 > 240 THEN START OUT3 , START OUT10
330 IF VAR11 = 200 THEN STOP OUT3 , STOP OUT10
320 IF VAR11 > 240 THEN START OUT8 , START OUT15
330 IF VAR11 = 200 THEN STOP OUT8 , STOP OUT15
340 IF VAR13 > 240 THEN START OUT9 , START OUT16
350 IF VAR13 = 200 THEN STOP OUT9 , STOP OUT16

360 REM * START HUMIDIFIER PROCESS *****
380 IF VAR2 OR VAR4 OR VAR6 OR VAR 8 OR VAR10 OR VAR12 < 850 THEN START OUT2 ,START (OUT17 TO OUT23) , START (OUT10 TO OUT16)
390 IF IN3 < 450 THEN START OUT1 , START OUT32
400 WHEN IN3 = 700 THEN STOP OUT1 , STOP OUT32
410 IF VAR2 OR VAR4 OR VAR6 OR VAR 8 OR VAR10 OR VAR12 > 950 THEN STOP OUT32 , OUT2

Expansion Inputs & Outputs

Description:

The T3 controllers can be expanded with external input / output modules.

The expansion points can be addressed the same as the on board i/o by referring to the input or output ID such as IN25 or OUT25. As of this writing, early 2017 the T3 controllers can be expanded to a maximum of 64 inputs and 64 outputs, this will be expanded to 128 inputs and 128 outputs in the very near term [TBD: update this as soon as this firmware is available].

TYPE OF T3	INNER INPUTS	INNER DIGITE OUTPUTS	INNER ANALOG OUTPUTS	OU
MP24	32	12	12	32
MP10	16	6	4	48
MP8	8	6	2	56

From the above table ,we can see the different MP support the different number of inner IO .I will show an example of **MP24** connecting a **T3-8I13O**

You can know other types MP, in the same way .

Please see the following screen shot :

License Agreement

All T3000 software is free and can be downloaded here: T3000Software.zip

The T3000 Building Automation System is free and open source. This means:
You may use this software for personal and commercial use free of charge.

You can install the software on as many computers as you like.
You can redistribute the T3000 Building Automation System in any manner you choose.

You can download and modify the software for your own commercial purposes without any royalties or fees, the license is very liberal MIT model, you do not need to share your derived products with anyone but of course any contributions are most welcome. The source code is here on github: T3000 Open Source Project

The T3000 Building Automation System IS DISTRIBUTED "AS IS". NO WARRANTY OF ANY KIND IS EXPRESSED OR IMPLIED. THE AUTHORS WILL NOT BE LIABLE FOR DATA LOSS, DAMAGES, LOSS OF PROFITS OR ANY OTHER KIND OF LOSS WHILE USING THIS SOFTWARE.
Installing and using T3000 Building Automation System signifies acceptance of these terms and conditions of the license.
Web: http://www.temcocontrols.com
Email: sales3@temcocontrols.com

Bacnet Integration

Software	Hardware	Bacnet		Modbus/Bacnet
Auto-Man	HOA	Out of	Output	Commands
AUTO	AUTO	FALSE	Follows local logic on the T3 controller, if any. Otherwise goes as commanded	Partial Control.
MAN- ON	AUTO	FALSE	Output is ON, managed by user at the PC or network commands	Writeable
MAN- OFF	AUTO	FALSE	Output is OFF, managed by user at the PC or network commands	Writeable
ANY	OFF	TRUE	Output is OFF, managed by user at the switch, network commands ignored	Ignored
ANY	ON	TRUE	Output is ON, managed by user at the switch , network commands ignored	Ignored

IP SETTINGS

1 — IP address Configuration   <TODO : Control Name>

5

IP Address
- ○ Obtain IP Address Automatically
- ● Use The Following IP Address
- IP Address : 192 . 168 . 0 . 109
- Subnet Mask : 255 . 255 . 255 . 0
- Gateway Address : 192 . 168 . 0 . 4
- Panel Number : 1 ✓
- Panel Name : DEMO
- Port : 10000

Time
- 2016/ 9/ 6 ▾   SYNC Local PC
- 14:52 ▴▾   Refresh Time
- ☑ Synchronized with the time server
- Time Server : time.nist.gov ▾
- Last Update : N/A
- Time Zone :
- (UTC + 08:00) , China Standard Time , Hol ▾
- SD Card Status: No SD Card

Load Factory Default — 2 — Load Factory Default button

Clean Device Database — 3 — Clean Device Database button

Config User List — 4 — Config User List button

7 — SD Card Status

8 — Dyndns Configuration

Device Basic Information 6

Device Serial Port Config
- RS485 SUB : Modbus Slave ▾   19200 ▾
- Zigbee : Unused ▾   19200 ▾
- RS485 Main : Modbus Master ▾   19200 ▾
- USB Type : ● Com port   ○ GSM
- Network Health

Dyndns Config
- ☐ Enable Dyndns Service
- Select DDNS Server : newfirmware.com ▾
- User Name :
- Password :
- Domain :
- Check the external IP address automatically 10 min

Device Information 9

Device Information
- Module Number : T3-BB
- Hardware version : 17
- MCU version : 42.7
- PIC version : 20
- Top version : 17
- Bootloader version : 23

Panel Information
- Object Instance : 40628
- Serial Number : 65729
- MAC Address : 00-0C-0C-01-01-00
- MS/TP Network : 10
- BIP Network : 1
- Modbus ID : 1

[OK]   [Cancel]

---

1 **IP address Configuration**

2 **Load Factory Default button**

3 **Clean Device Database button**

4 **Config User List button**

Config User List

5 **<TODO : Control Name>**

Time

2016/ 9/ 6

SYNC Local PC

14:52

Refresh Time

☑ Synchronized with the time server

Time Server : time.nist.gov

Last Update : N/A

Time Zone :

<TODO>: Insert description text here...

6 **Device Basic Information**

Panel Number : 1

Panel Name : DEMO

Port : 10000

<TODO>: Insert description text here...

7 **SD Card Status**

SD Card Status: No SD Card

<TODO>: Insert description text here...

8 **Dyndns Configuration**

Dyndns Config

☐ Enable Dyndns Service

Select DDNS Server : newfirmware.com

User Name :

Password :

Domain :

Check the external IP address automatically 10 min

<TODO>: Insert description text here...

9 **Device Information**

Device Information

Module Number :	T3-BB
Hardware version :	17
MCU version :	42.7
PIC version :	20
Top version :	17
Bootloader version :	23

<TODO>: Insert description text here...

Offline Mode

T3000 has a feature where you can operate on a T3 controller in offline mode. This is useful when you don't have a controller on hand for demo purposes or working in the office for later deployment to a remote job site.Currently only T3 controllers are supported.Below is an explanation on how to operate on a virtual controller.

First, load into the T3 controller in offline mode as the link *http://tinyurl.com/ybaj9d3u/ ExamplePRGFiles.zip*,and set T3000 to offline mode from the menu system.

The currently selected device showing at Tab 1 will start flashing red-blue-green to signify that it is no longer online. Right click on the 'Virtual Device' icon at Tab 2.
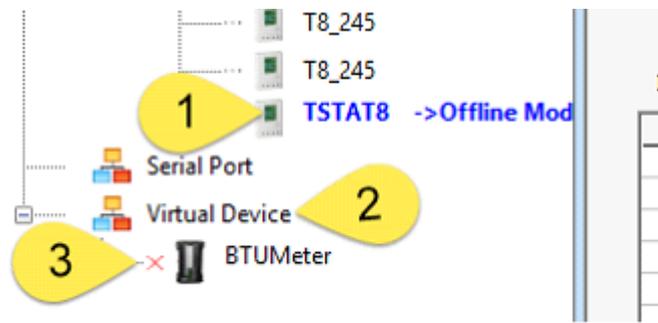
Which brings up two choices for sorting the tree and the 'Add virtual device' option which is what we're interested in.
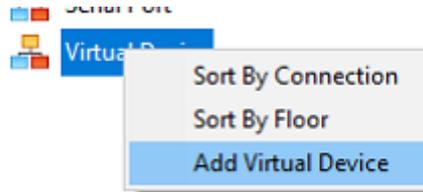
Many devices show up in the next dialog which pops up. The only working one at present is the T3 controller series, BB, LB and TB programmable controllers.
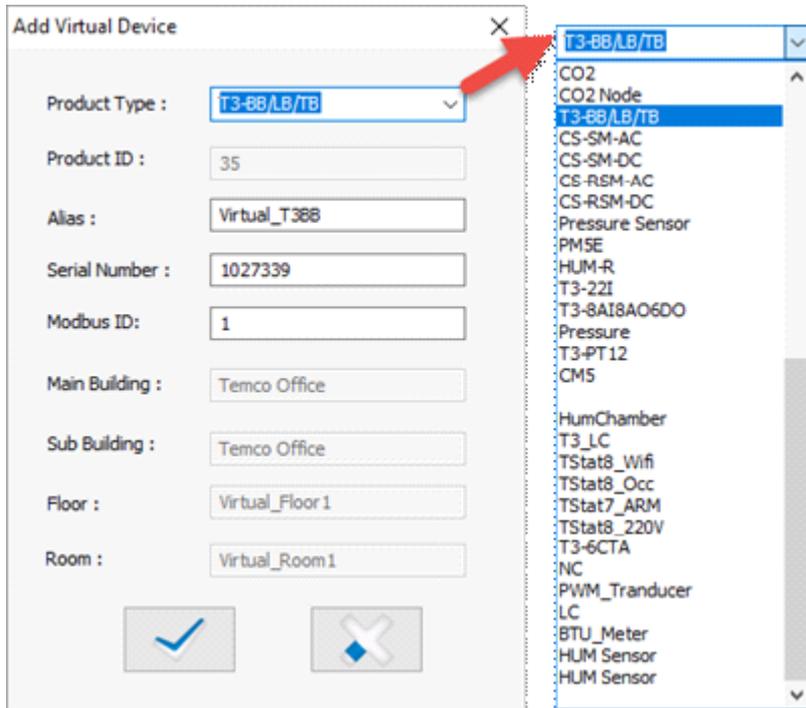

Next load an example file into the virtual controller.You will see the status of the file loading down at the bottom of the screen.
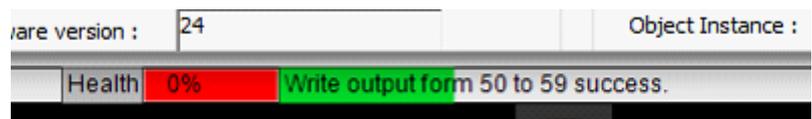

After loading the 'prg' file you can view the inputs, outputs, programming and so on of the virtual controller as if you were online with actual hardware.
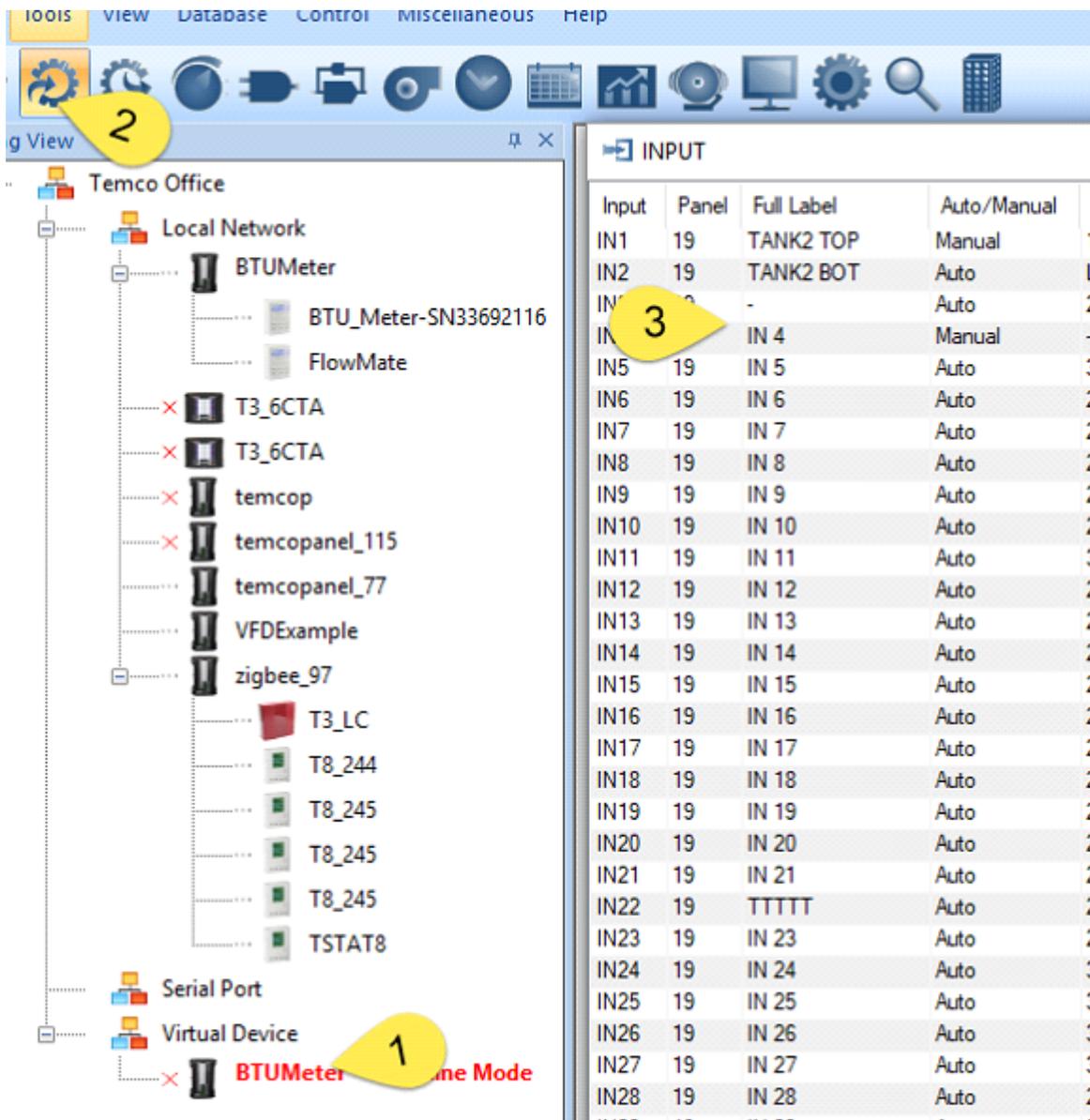
After editing you should save your changes using the File -> Save menu item and select a folder & name for the updated program. The file is stored as a 'prg' or program file for backup purposes, for copying to another controller or be emailed to someone. The PRG file contains all the i/o point definitions, programming, user screens and so on which a controller uses to perform its job.
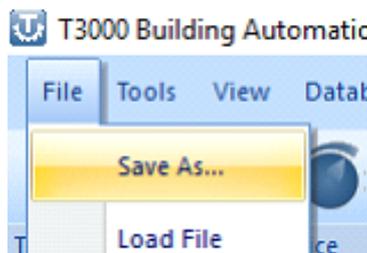

A good naming convention helps keep track of the most recent version of the file while still keeping the earlier versions on hand in case of problems.
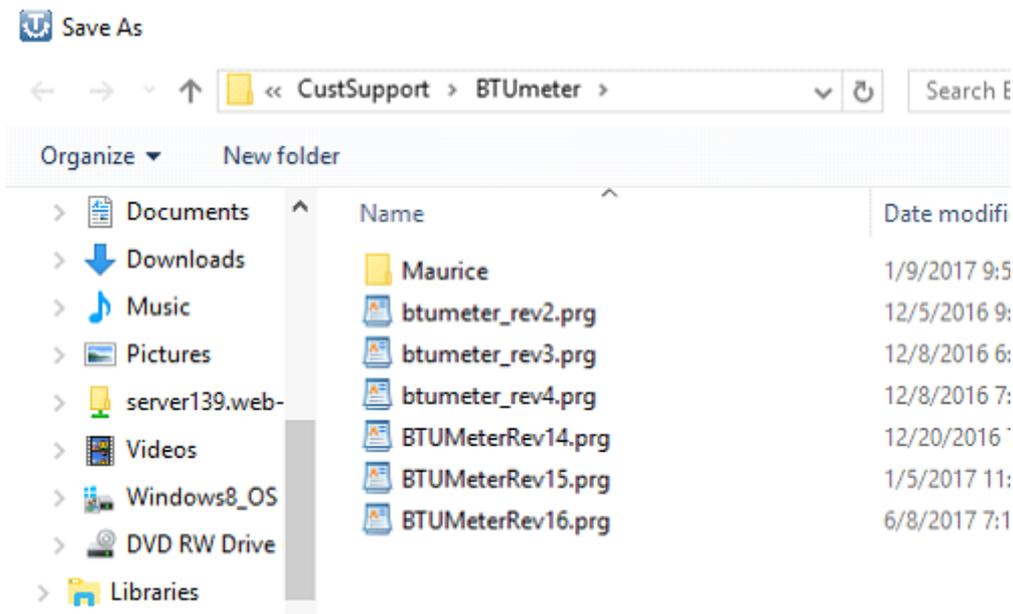
The only files which the PRG file does not contain are the graphics images which need to be copied over manually to any other PC.'s which will be accessing the system. In this example, DemoSite1 is currently the selected building. The image files are stored in the folder shown at Tab 2, in this case the folder is called C:\Program Files (x86)\T3000\Database\Buildings\DemoSite1\image. All image files for this particular site are stored in this folder. Tab 3 shows the pictures are in the 'image' subfolder.
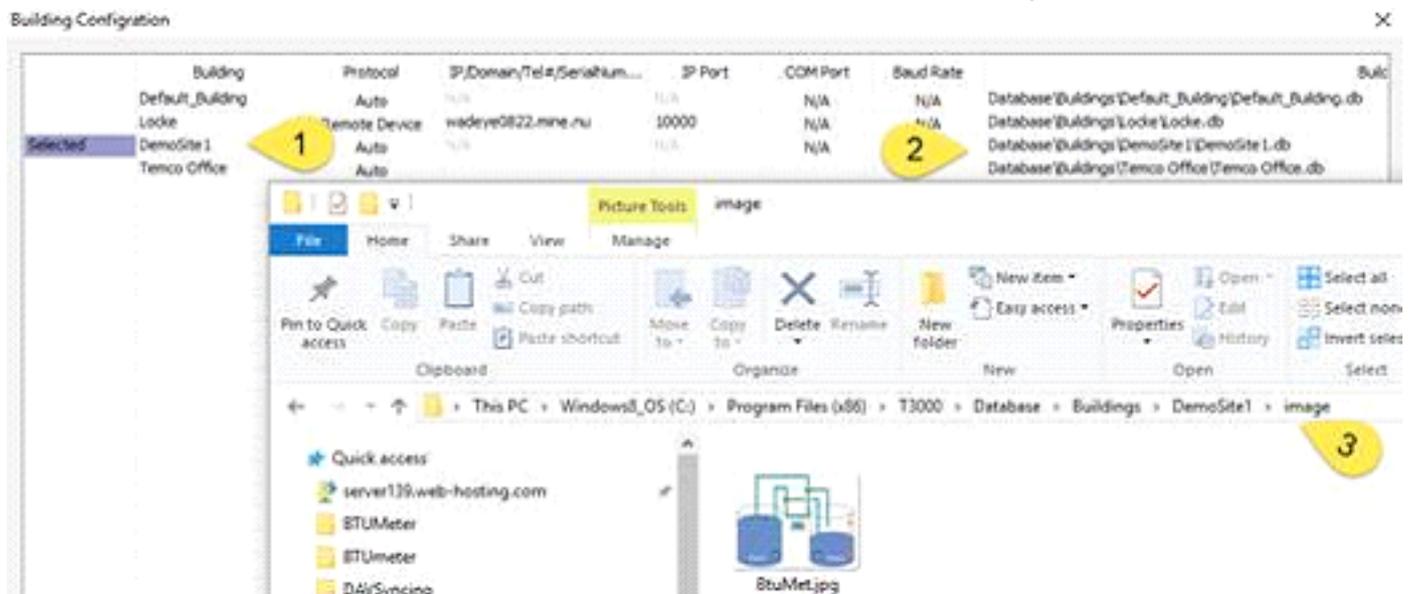
While it is possible to add a second virtual device, the system only allows you to operate on one at a time. Changes made to one virtual controller will affect the other virtual controllers. TBD: allow multiple virtual devices.

<TODO : Topic name>

<TODO : Topic name>