

## T3000 Building Automation System



---

User guide Rev7.3  
April 2016

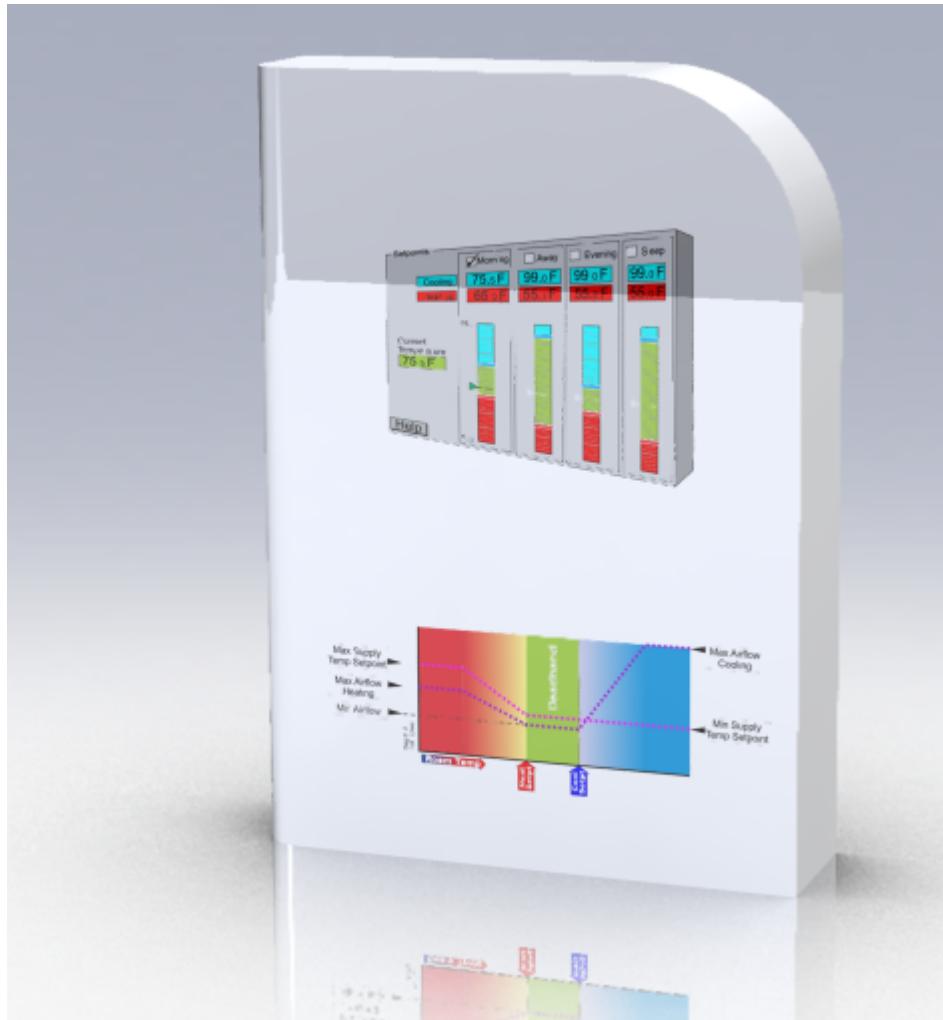
## Table of Contents

System Description .....	7
General Overview .....	8
System Highlights .....	9
System Description .....	12
Software .....	15
SoftwareOverview .....	16
System Requirements.....	19
Keyboard and Mouse .....	20
Open Source Project .....	23
T3 Controller Hardware .....	24
T3 Controller .....	25
Expansion Modules.....	29
T3-8AO .....	30
Communications Ports.....	32
Wireless.....	33
Installing and connecting .....	34
Installing T3000.....	35
Hardware Details .....	38
Misc .....	40
Troubleshooting Ethernet Connections.....	43
5 System Setting .....	45
6 Control .....	47
6.1 Graphic Screens.....	48
6.1.1 Insert Point .....	54
6.1.2 Edit Point .....	55
6.2 Programs .....	57
Inputs.....	59
Menu .....	65
Pulse Counting.....	66
6.4 Outputs .....	67
6.5 Variables .....	71
6.6 PID Loops.....	75
PID Loops Introduction .....	76
PID Loop Work Screen.....	79
PID LOOP MATH DISCUSSION .....	82
PID Loops Introduction .....	83
PID Loop Math Discussion .....	84
6.7 Schedules .....	85
6.8 Holidays .....	88
6.9 Trend Logs.....	90
6.10 ALARM LOG.....	93
Panel Configuration .....	94
8 Control Basic Programming .....	98
9 Programming a controller .....	106
10 Programming Statements .....	113
11 T3000 Release History .....	144
12 A DEMO SHOW .....	156
13 How to update firmware and load configuration file one by one .....	164
How to batch to update firmware and load configuration file .....	166
14 How to use subnet .....	168
Expansion Inputs & Outputs .....	170
License Agreement .....	173



## Introduction

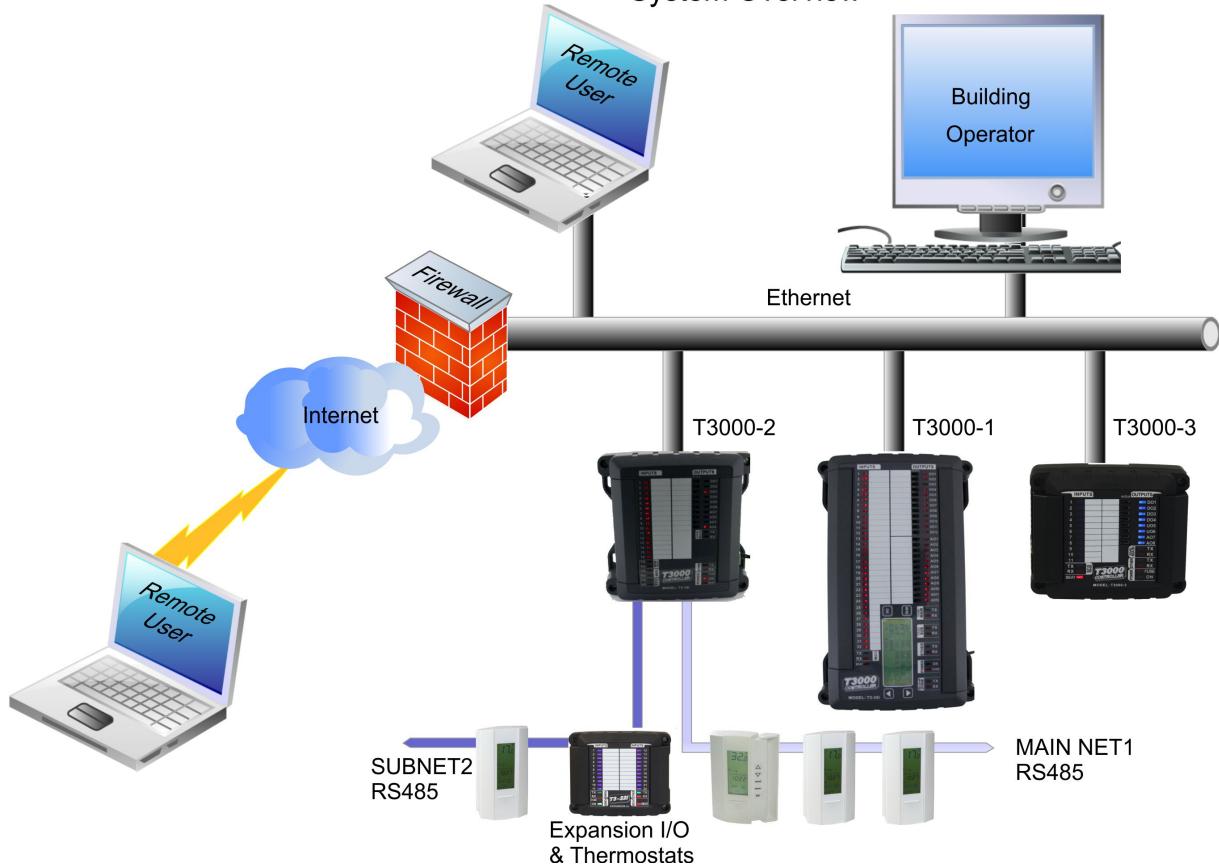
The T3000 Building Automation System is a complete solution which allows a building operator to manage the indoor environment of commercial, institutional and residential buildings. Equipment such as chillers, air handlers, boilers, fans, lights, pumps, dampers, valves and other mechanical equipment can be scheduled and controlled according to the sequence defined by the control system designers. Extensive monitoring and logging features allow the operator to see what is going on in their building to proactively improve comfort and energy efficiency. The system has powerful communications capabilities making it possible to integrate with other systems and can scale to virtually any size from a single home thermostat on up to an entire campus with thousands of controllers.



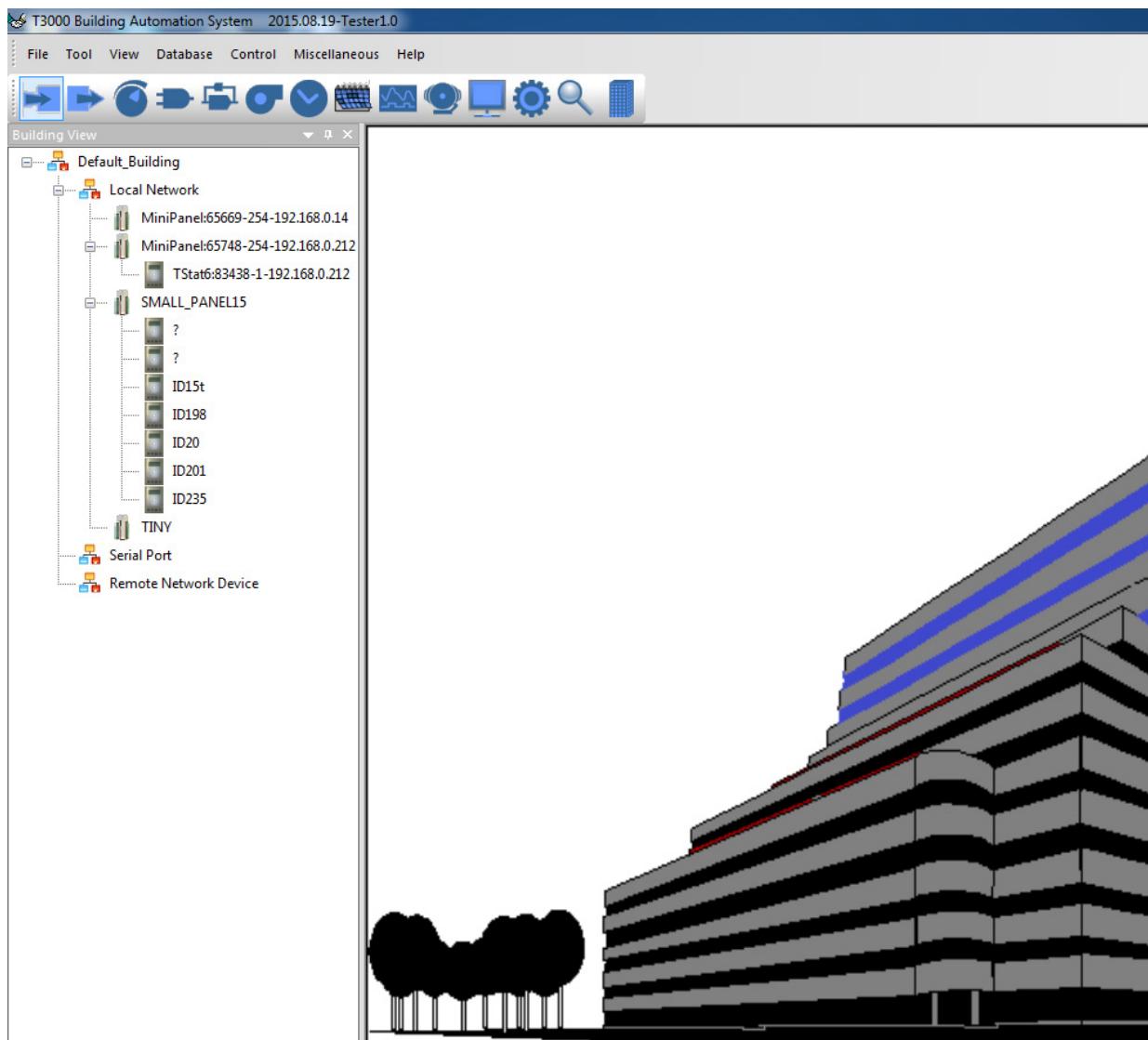
Monitoring of a number of buildings can be done from one central location. An operator with the appropriate privileges may access the entire system regardless of whether they are logged on over the internet or locally from a PC inside the building. End users and tenants can be set up with access to only their own areas which gives them the power to control their own area as they like.

Simultaneous users may access the system over one of the many communications channels. Alarms, system graphics and trend data are stored directly on the controller so there is no central server required and no single point of failure.

## T3000 Building Automation System Overview



Interactive graphic displays, floor plans and custom menus make operation simple for any user. The flexible T3000 hardware and software makes it easy to configure and reconfigure the system throughout the life of the building.



The T3000 control panels can be connected together to form a complete building automation system which normally includes the mechanical systems in the facility and may also include lighting, access control and the alarm system.

## System Description

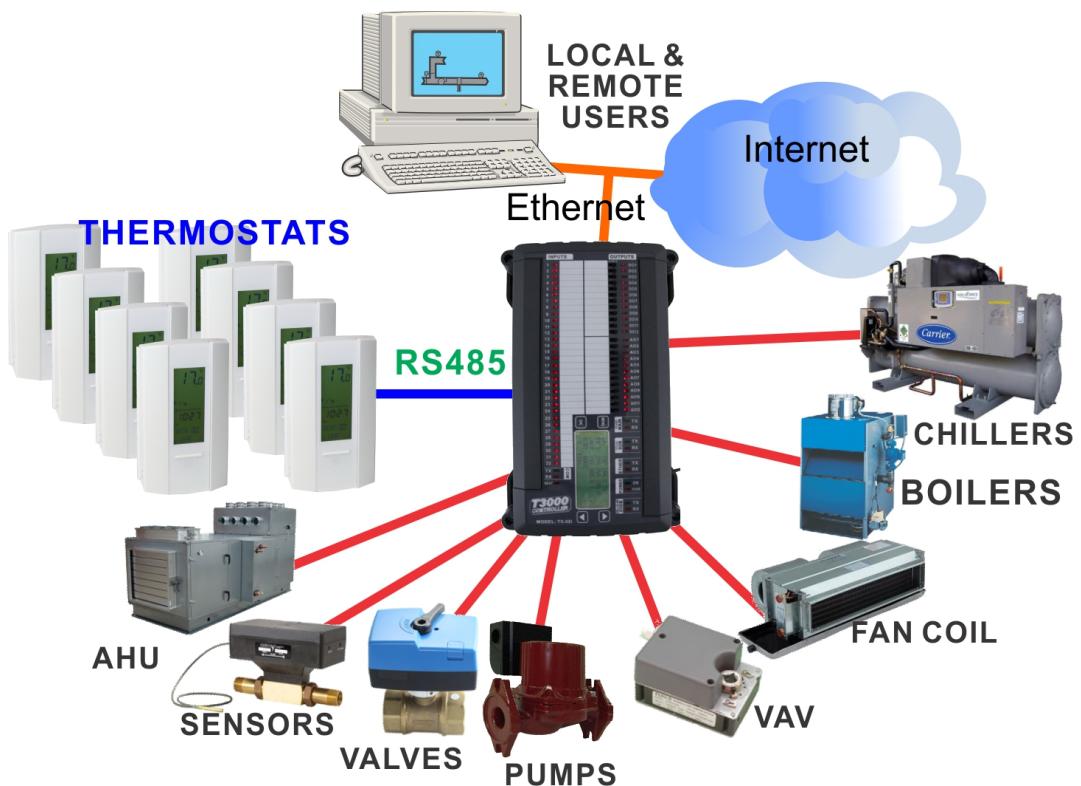
The introductory topics will give you the brief overview of the T3000 Building Automation System concept. A person with some controls experience will be up & running with the product in a short time. A new user will be able to iteratively configure, test and experiment till they are familiar and will soon be able to set up a professional grade controls system on their own.

## General Overview

### GENERAL OVERVIEW

The T3000 Building Automation System is a total solution which the building operator can use to manage the mechanical and lighting systems of the building. It is simple enough to use that beginners will be able to perform basic tasks such as adjusting setpoints & schedules, tend to alarms and perform day to day troubleshooting of the system without special training. Advanced users will be able to dig into advanced features such as tuning the logic, adding new screens & reports and optimizing the building. All the software is included for each site, advanced features like data archiving and creating new screens can be done by the building owner without paying any license or recurring cloud fees.

All software and hardware communicate with industry standard Bacnet and Modbus over Ethernet and RS485 networks. The front end is free open source software so the building operator can be confident they have full control of their system and are not locked into expensive options during the life cycle management of the building.



## System Highlights

### BUILT TO LAST, BUILT TO GROW

The T3000 Building Automation System is easy enough for the engineering departments to take over management of the system if they choose and expand the system with their own staff as budgets permit. Once the control system is installed, commissioned and handed over to the building owner that is just the beginning point of a control system built with the T3 controllers. The staff has all the tools on site with no license schemes to hold them back. They can make improvements at will to make their daily jobs easier.



### LARGE OR SMALL PROJECTS

There is no practical limit to the size of the system which can be built up with a network of controllers and expansion modules. On the other hand, because there are no software licenses or monthly 'cloud' fees it is very cost effective for small sites.

**Simple, low cost, full fledged  
building automation system**



### NETWORKING CENTRIC SYSTEM

Networking is accomplished with regular ethernet cabling and hubs so administrators can expand the system over existing infrastructure or build a separate subnet. The system maintains a database of the names of all points in the system, if the item happens to be located on another controller all the details of creating the network point and keeping the info up to date is handled transparently. You can refer to inputs, outputs and other system variables freely in your programming, alarming and logging regardless of which panel the item resides. For a simple example the outside air temperature sensor in the building which can be given the name "OUTSIDE\_TEMP" and is connected to panel

1, input1. Any controller in the building can use this item in the program by referring to the sensor by the name "OUTSIDE\_TEMP".

## ACCESS FROM ANYWHERE

An optional GPRS or CDMA radio module can plug into the controller for remote sites where there is no internet connection available. This allows the operator to access the system with the full functionality of an onsite connection, it doesn't matter if you are on site or remotely, the look & feel is the same. 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.



## 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 nonvolatile flash and will be automatically loaded on power up. The real time clock has an infinite life super cap that will keep the clock alive for several days, there are no batteries to replace ever.

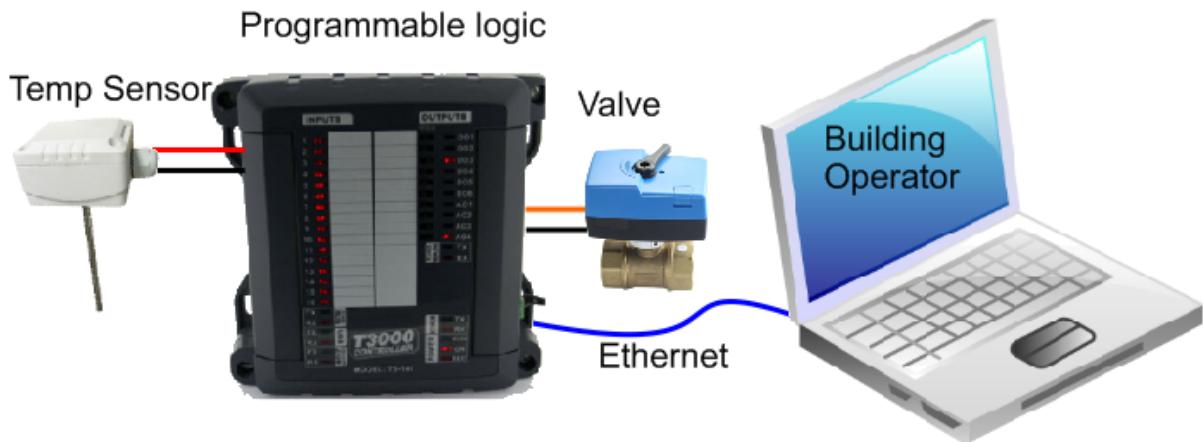
## WORLD CLASS WARRANTY

If a panel has a problem, send it to us and we'll do our best to fix it. There's no charge for this service and there's no time limit on the offer. We do ask you to pay shipping both ways. If the board cannot be repaired due to water damage or obvious wiring mistakes we'll ask you to buy a new one, just return your damaged unit and we'll offer a healthy discount on a new one.



## System Description

The T3000 system suited for automating the mechanical systems in larger buildings but is also a good match for light industrial applications and larger residential projects. A simplified example is shown below, a control system will maintain the temperature in a tank. There will be a temperature sensor to show the temperature of the tank, a valve to add heat to the tank, and the controller. The PC is used initially to program the controller. Normally there will be a PC located on site to manage the system, adjust setpoints & schedules but it can be removed and the system will operate stand alone. The operator can then log in remotely or locally and alarms can be sent to the operator.



There are several models of controllers and external expansion modules which allow the system designer to create large systems. All control decisions are performed directly by the control panels independent of other computers on the network so there is no single point of failure. The front end GUI allows the person setting up the system to create the logic, user screens and so on. This same software is also used by the building operator in the day to day running of the system. There are no separate tool suites for contractors, all the software is included for free and it can be copied to an unlimited number of PCs.

### T3000 HARDWARE ARRANGEMENT

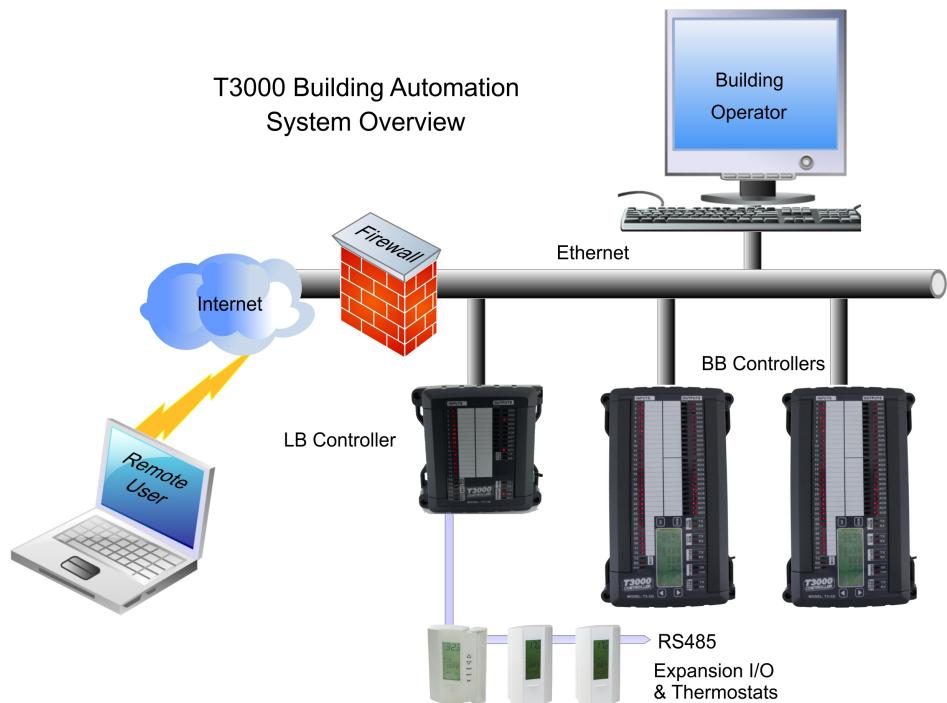
A typical T3000 panel installation is shown below. The T3000 panel is connected to an Ethernet cable which could be run to other T3000's distributed throughout a large building or office complex for example. The wiring from the sensors and other field devices can be connected directly to the controller. Expansion is provided by a number of different T3 expansion modules.

A PC is normally connected to the controller through a hub or router. You can connect direct to the controller with a normal ethernet cable, normally this would require a 'crossover cable' but thanks to the 'Auto MDI-X' feature on most PC's these days it is not necessary.

TYPICAL T3000 SYSTEM CONFIGURATION Diagram goes here.

### 1.5 T3000 MINI-PANEL HARDWARE ARRANGEMENT

The following diagram shows a typical mechanical room system overview. The BB controller has 32 inputs and 24 outputs and can be expanded using the T3 series of I/O modules, up to a maximum of 64 inputs and 64 outputs on each of the controllers. The i/o of the expansion modules can be programmed, logged, alarmed and so on as if they are local control points of the controller panel itself.



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 if the designer chooses, the expansion bus uses regular RS485 cabling so the expansion modules can be mounted long distances, hundreds of meters away if necessary.

The main backbone of the larger systems where there are multiple controllers in the building will be Ethernet cat5 cabling together with standard ethernet hubs and routers. The control system will normally be on a separate sub network but in a small office for example, it is possible to combine the building automation network with the office or home lan. Care must be taken to avoid IP conflicts since a duplicate IP address will affect communications.



Each of the controllers has two RS485 ports which support a subnetwork of smart devices such as thermostats, vav controllers and sensors. Access to the I/O of these subnets is built into the system by referring to the master controller and subnet address. Routing of information is handled transparently by the routing features of the controllers.

The controllers all have an SD card slot which allows storage for the configuration and logic file, but also for long term logging and a local copy of floorplans and system graphics. There is no requirement to have a dedicated server running for archiving. Data from the lower level controllers will replicate its way through to other computers in the system when a connection is available.

This diagram shows a typical small scale system with three controllers joined together over Ethernet. There is no practical limit to the number of panels that can be on this network, vast systems can be constructed using industry standard network gear. The system also supports legacy RS485 backbone cabling though the communications will be much slower than with ethernet.

## **Software**

The following section describes the T3000 software features

### SOFTWARE

TBD: Confirm these.

- 128 Control Basic programs
- Up to 128 inputs, 128 outputs per controller
- 128 Variables
- 48 PI Controllers
- 32 Trend logs
- Runtime totalizers and event logs
- 128 User screens
- 16 Weekly schedules
- 8 Annual schedules
- 16 Variable arrays
- 50 User name/ password combinations
- 5 Custom tables
- 8 Custom digital engineering units
- Digital/Analog alarms
- Programming editor
- Graphical user screen editor
- Icon representation of items on displays
- JPG and BMP file format for floor plans, mechanical system diagrams and icons

### 1.8 THE 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. All software is included with each project so there are no special licenses or addons or recurring fees. The software runs on any PC running Windows XP, Vista, Win7 and 8. There are no versions for linux or Mac, sorry. The front end software for the T3000 system is heavily keyboard driven since many tasks are performed by technicians troubleshooting in the field. Experienced users will also appreciate the many keyboard shortcuts to speed up the day to day work. Casual users will be able to find their way around the system without special training.

#### 1.8.1 GETTING AROUND THE SYSTEM

Most of the system configuration is done in a series of tables which the user can navigate and edit like an ordinary spreadsheet. The tables are updated dynamically with real-time information even as the tables are being edited. Navigating the program can be done through the drop down menus, with the mouse, with Hot Key combinations. [TO BE DONE] For experienced users there is a command line which is brought up by pressing the space bar, from here you can type in commands to quickly maneuver around the system.

### **1.8.2 ALARMS**

Alarms can be generated using custom logic in combination with the ALARM and DALARM statements in Control Basic. These statements allow for fine grained alarm generation of 70 characters, each tailored to a very specific state of real time conditions and timing.

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 SD 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.

### **1.8.3 GRAPHICAL INTERFACE**

The T3000 operating system can display floor plans and mechanical diagrams of the building using jpegs 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 placed in a special folder for each building and pop up when the user logs into the particular building. Real time data is overlaid on the static picture to show the temperatures, setpoints and so on of the system. The system is easy to configure so you can move elements around on the display, add items and reconfigure the system interactively though out the life cycle of the building. The screens can be nested to several levels deep enabling an operator to navigate through the entire building via the linked graphic images and real time data. Icons pointing to the trend logs, programming, and any associated item can also be placed and moved around on the screen while online.

### **1.8.4 CONTROL BASIC**

Control Basic is the programming language used for the user programs which control the building. This language is very easy to read and maintain, novice users will get up to speed in no time. A sample of the programming is as follows:

```
10 IF SCHEDULE THEN START FAN101
```

```
20 IF TEMP1 < SETPOINT1 THEN START HEAT1
```

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

The programming is stored in the local flash memory of the controller. There is also a backup stored on the SD disk if there's one inserted in the SD card slot. A power outage does not affect the operation of the controller, it will resume operation without user intervention as soon as power is restored. The programming can be changed while the building is operating so it is simple to update the logic in an interative manner.



## System Requirements

### Minimal System Requirements

The T3000 application is not particularly demanding on the CPU, a modest Win10 system will run the application with good response. There is no linux or Mac version of the software. Windows XP is no longer supported because the Microsoft Visual Studio tools used to compile the project have recently recently removed support for XP.

- Microsoft Windows 8 thru Win10
- SVGA or better, color display
- 500 MHz processor
- 256 MB 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:

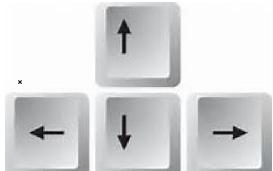
1. Words in capital letters are keywords and must be entered exactly as shown, for example, they keyword: OUT1 cannot be typed in lower case or any mixture of upper and lower case.
2. Items in square brackets [ ] are optional. (i.e. [LET] A = 10)
3. Program listings appear in a line printer font as shown:

```
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, $\leftarrow$ , $\uparrow$ , $\downarrow$ , $\rightarrow$ .



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 > changes are done.

#### The Escape key, Esc.

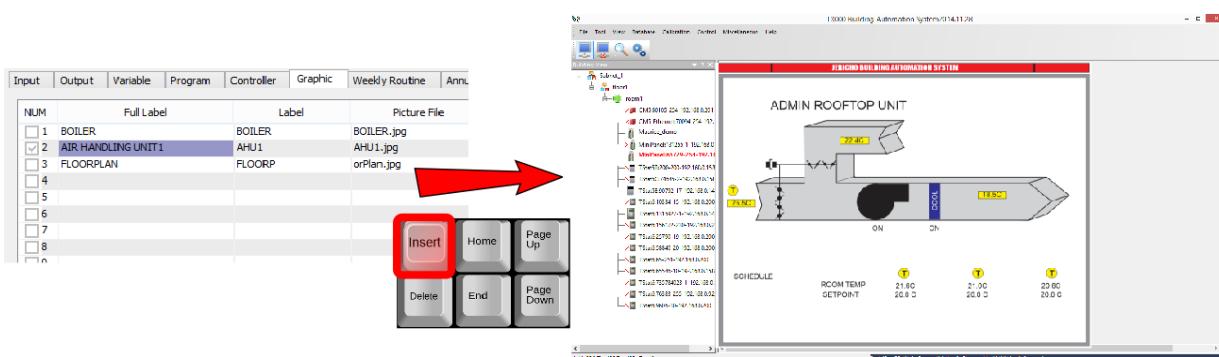
During editing you may decide to back out of the changes, the ESC key will let you revert back before the changes are sent to the controller. In most cases this will be only a few characters of text which are changed but during editing of a basic program for example there may be several lines of code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

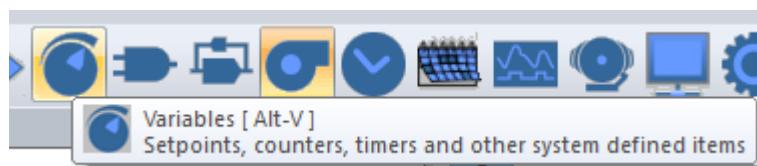
This key is handled a little unconventionally like the 'Enter' key is but saves mouse clicks and helps get around the system quickly. In many of the grids such as Programs, Weekly Routines, Graphics Screens, and Trend logs there is a detail page for each of these items which pops up with the INSERT key. 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.

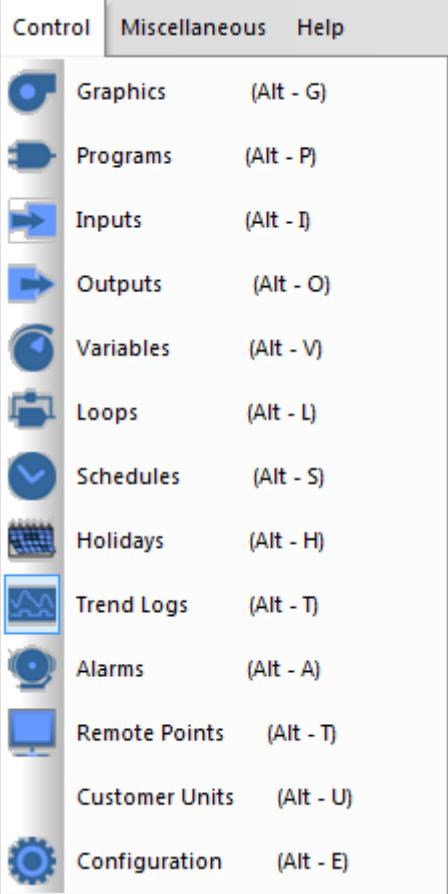


## HOT KEYS

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





## Open Source Project

The T3000 front end is one of the few building automation systems on the market which is open source. This is done in order to give customers and partners confidence in our solution. We've never charged for software in our many years in this business. The open source code is maturing and there is a growing community supporting the work. We encourage other manufacturers to join in the project and contribute to making the software more & more powerful. 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.

Here is the source code hosted on the open source repository site Github: [T3000 Open Source Project](#)

The screenshot shows a GitHub wiki page for the 'temcocontrols / T3000\_Building\_Automation\_System' repository. The page title is 'Home'. It features a sidebar on the right with navigation links like 'Pages 19', 'Cloning the fork', 'Commit changes', etc. The main content area contains several sections of text and links, including a welcome message, instructions for contributing, and links to README files and development guides. At the bottom left, there's a 'Quick Links' section with links to 'Introduction to T3000 project' and other resources.

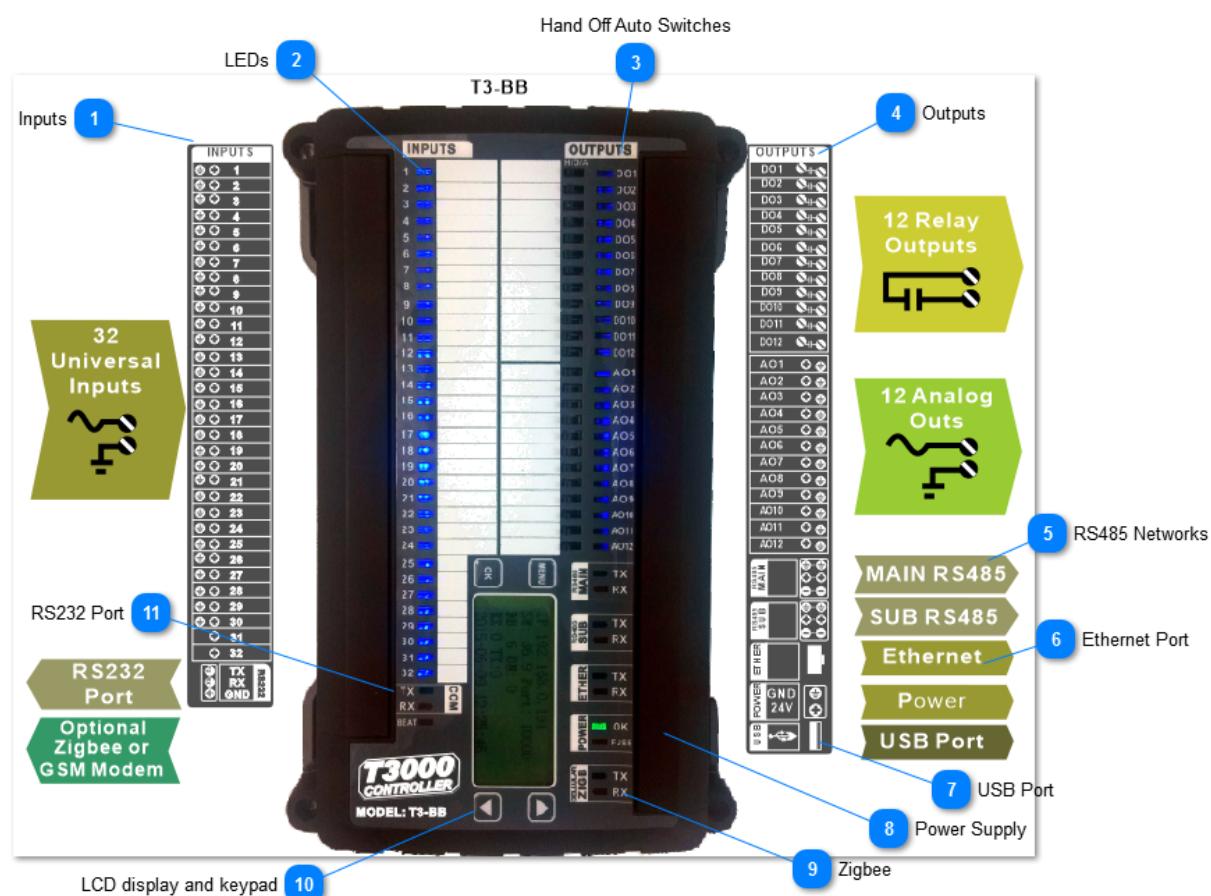
## T3 Controller Hardware

## 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 normal duties. 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.

Extra 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 using RS485 so they can be located several hundred meters if necessary.

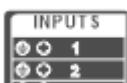
The T3000 Controller hardware is shown below. The hardware has many hardware features which make the controller easier to commission and work with over the life cycle of the building. Each output has a hand-off-auto switch to make it easy to override the controller logic during troubleshooting. There 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 an upstream and a downstream connector to make RS485 debugging easy. In summary, this controller hardware is made with the building operator in mind with many features to help keep the system running as intended.



The heart of the T3 controllers is an Asix AX11015 CPU which chip provides 100 MIPS of cpu processing power. There is an on board ethernet coprocessor to handle the IP traffic which reduces the load on the main CPU. There is 512k flash and an additional 512k external ram chip. This is plenty of power for the typical controls application, no matter how complex the user programming all of the user code runs one time each second.

- The Real Time Clock provides 1 second accuracy per day.  
When there's an internet connection available the unit can sync with internet NTP time servers  
such as time.windows.com and others.
- Watchdog Timer provides an automatic restart in the event of a computer crash. When power resumes the unit starts up with no user intervention.
- SD disk slot allows storage of local trend logs, programming setup and graphics templates.  
Large disks such as 4G and larger are supported.
- A new version of the controller is under way with a more powerful 64 bit CPU which will begin shipping in Q2 2016.

## 1 Inputs



The Input cards have universal inputs which can accept signals from common control system devices such as dry contacts, thermistors and transducers. The front end software lets you set up the range and the engineering units for each of the inputs.

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 a 1Mhz high speed pulse counters on a few of the 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.

## 2 LEDs



There are many LED's on the controllers to show the status at a glance, all of the inputs and outputs have an 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.

## 3 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.

## 4 Outputs



There are two types of outputs, the relay outputs and analog 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.

## 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 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.

## 6 Ethernet Port



The Ethernet port is 100Mbaud/10Mbaud auto switchover. The controller can talk over Bacnet and Modbus 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 problem with the controller.

There is hardware on board for surge and brownout protection. When power goes outside the normal bands the controller will gracefully power down and resume when it returns to normal.

## 9 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.

#### 10 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 function.

#### 11 RS232 Port

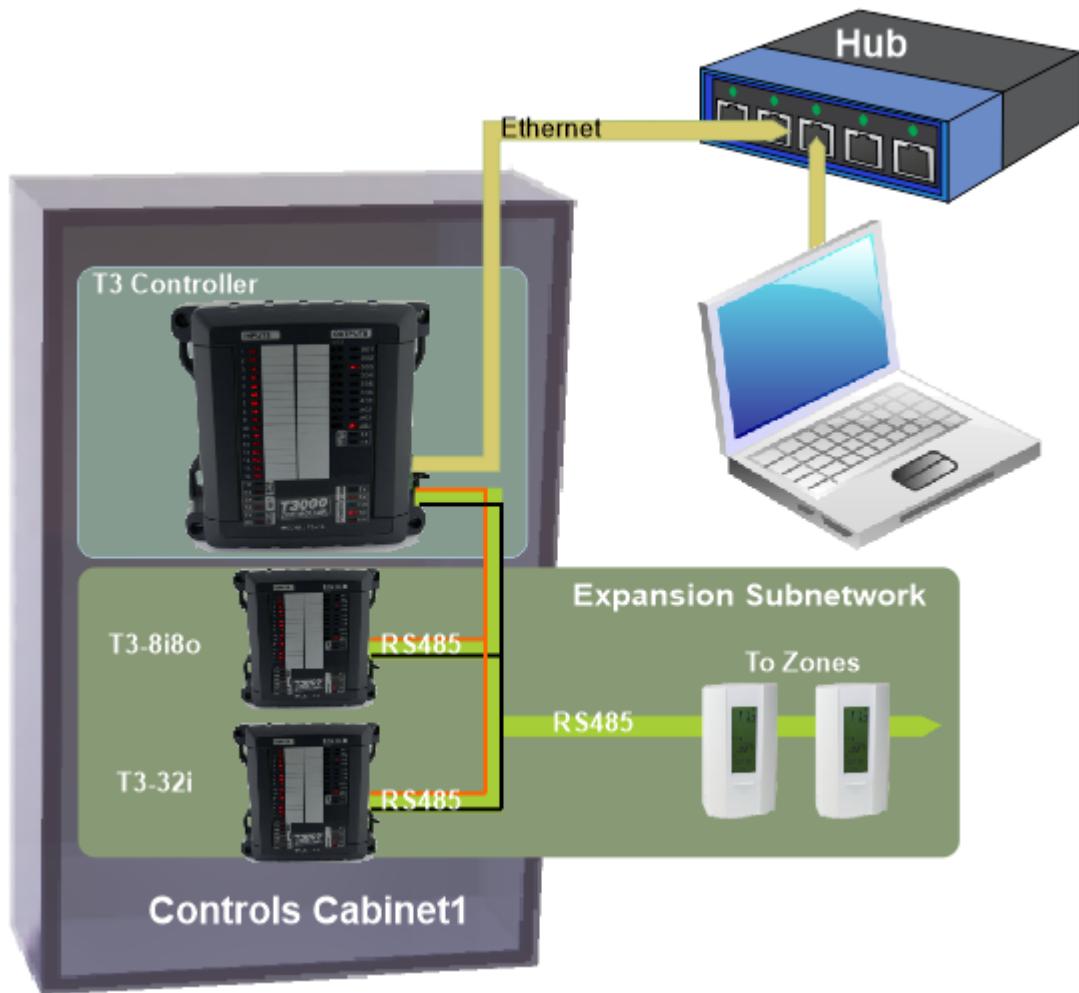


The T3-BB has an RS232 port which supports Modbus and built in drivers for a few devices such as power meters, weigh scale and so on. The port is currently not available to the user application other than these built in modbus and fixed purpose functions.

## 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 could be located remotely up to several hundreds of meters away if the application calls for this.

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 controlled treats the expansion points transparently like internal points directly on the controller itself.



## T3-8AO

### T3-8AO MODULE

The T3-8AO Output Module has eleven 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 from 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.

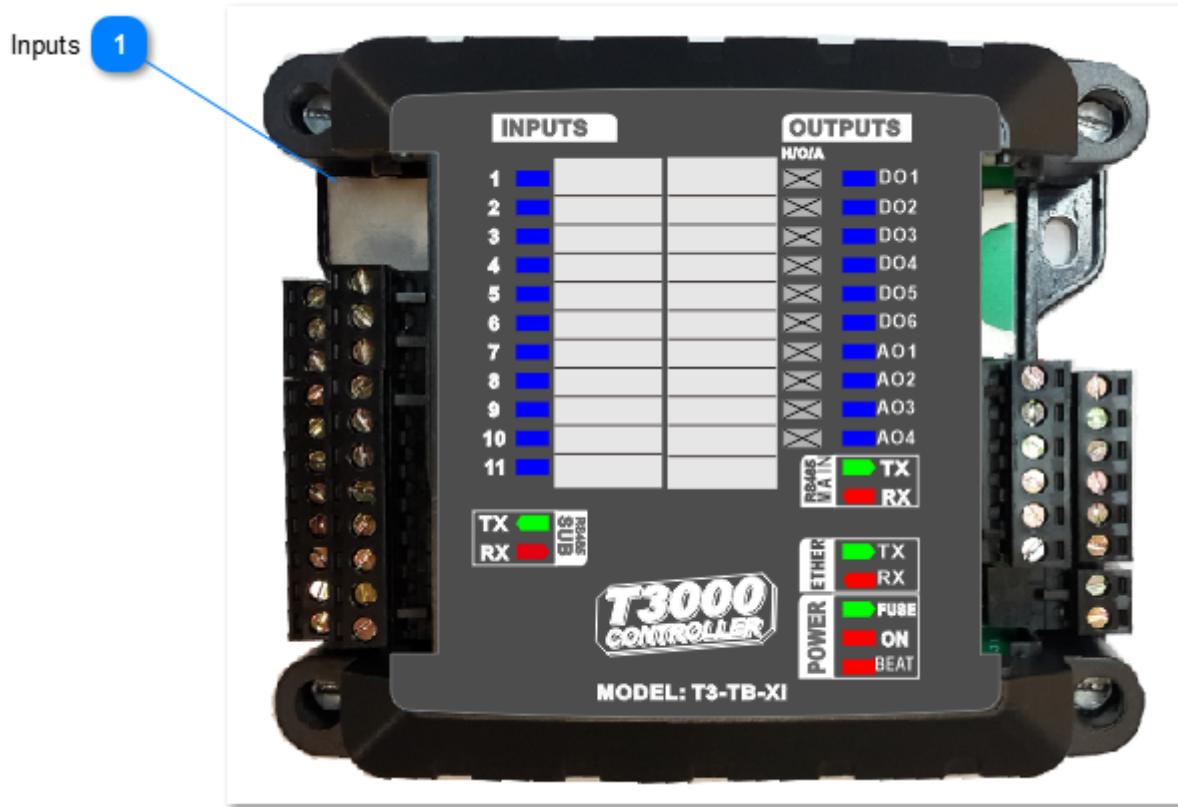
The input signals have the following characteristics:

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.

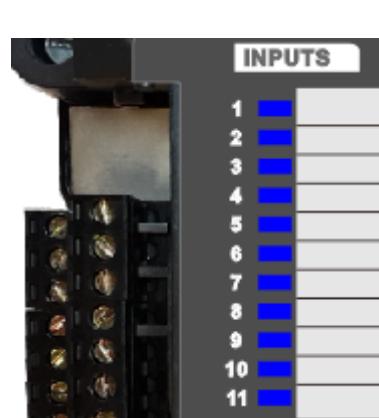
Analog Outputs are used for modulating type actuators, variable speed drives, valves and damper motors. They are fixed in hardware to operate over 0-10V and can source a maximum of 100 mA per channel. If your application calls for more binary outputs, any spare analog outputs can be set up as relay outputs 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 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.

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



Inputs

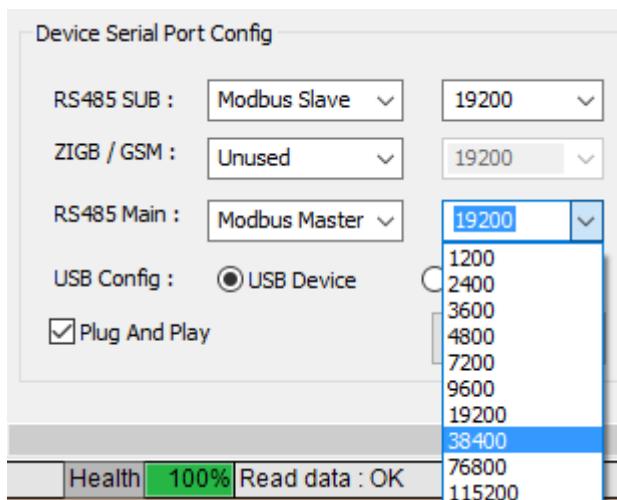
<TODO>: Insert description text here...

## Communications Ports

	BB	LB	TB
Communications ports:			
RS485 ports:	2	2	2
RS232 Port	1	0	0
Ethernet Port	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.



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.



## 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, March 2016. You will need to locally source a prepaid sim card which is then inserted into the modules 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.

TBD: Show zigbee and GSM modules here.



Tstat6-Zigbee

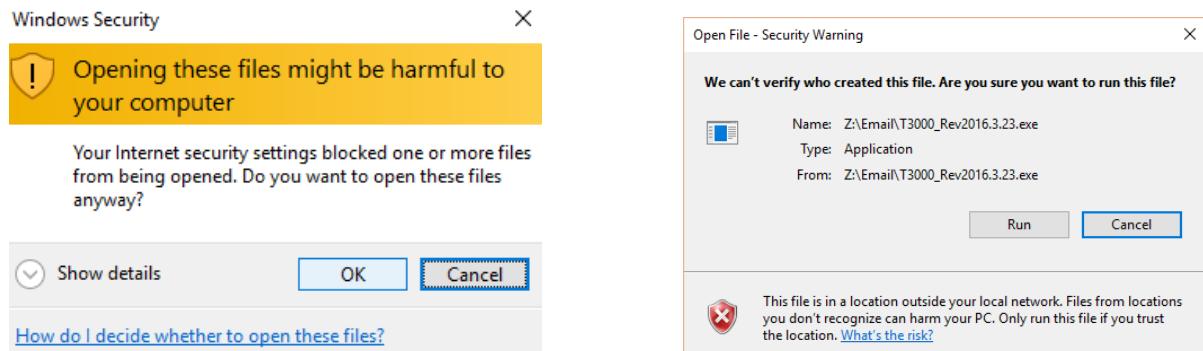
## Installing and connecting

## Installing T3000

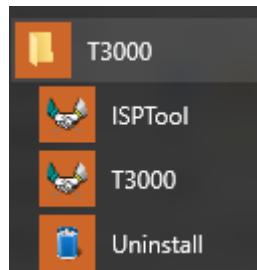
The following chapter discusses how to run the T3000 software and connect to the controller for the first time.

Download and install the latest T3000 software from the web site: [T3000 Software](#)

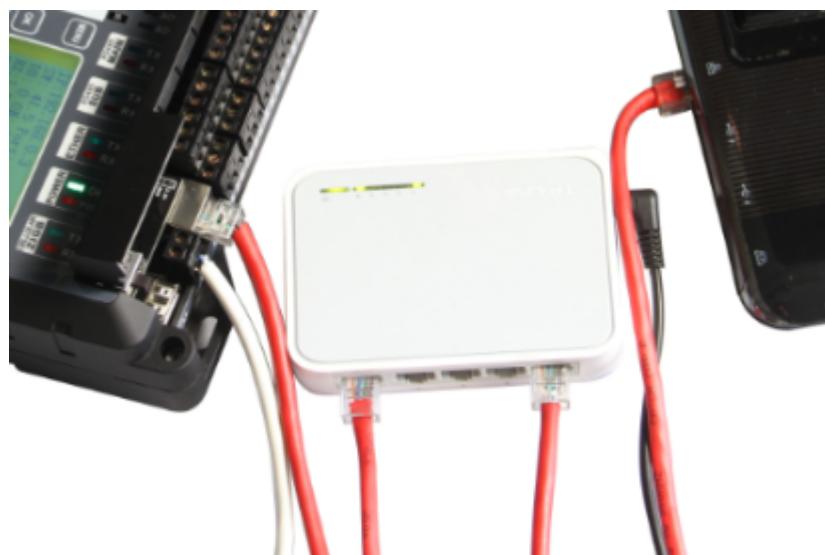
A zip file will be downloaded to your local download directory where you can unzip it and store it for future installs. You can ignore the security warnings while unzipping and running the application.



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 ISPTool is a simple and lean stand alone utility for updating firmware in the controllers which will come in handy for power users.



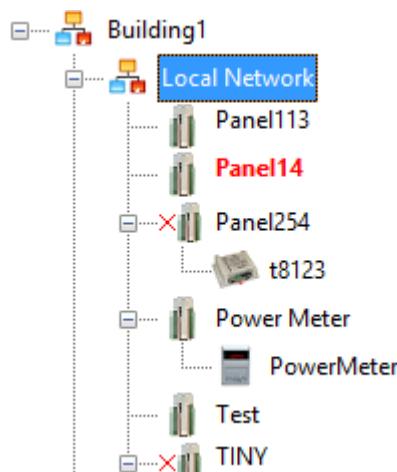
Connect the PC and the controller through a hub as shown here. The controllers and T3000 application can coexist smoothly on the typical small office or home lan but in a commercial building the control system would normally be on a separate network. Power is connected as show, any supply from 12V to 24V and either AC or DC will be fine, the capacity should be approximately 2 watts.



After power up, the T3 controller will go through a bootup sequence for approximately fifteen seconds where the heartbeat is flashing with two quick pulses every second. 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 to discover any connected devices.



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.

## 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.

### 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.**

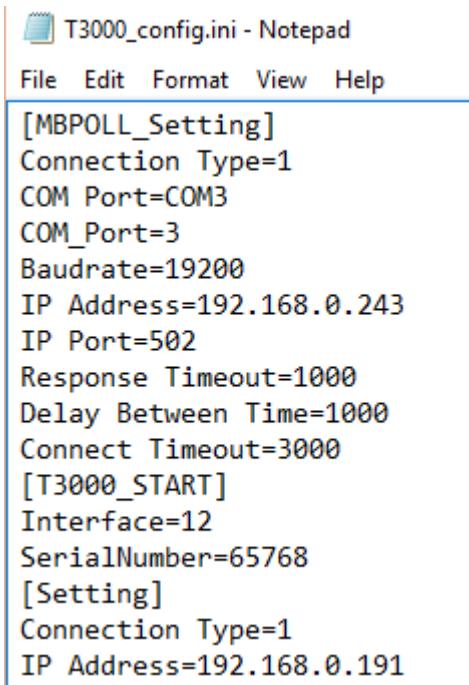


## 3.2 SETTING- UP THE T3000 SYSTEM

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 file named CONFIG.INI which is saved to the T3000 installation directory.

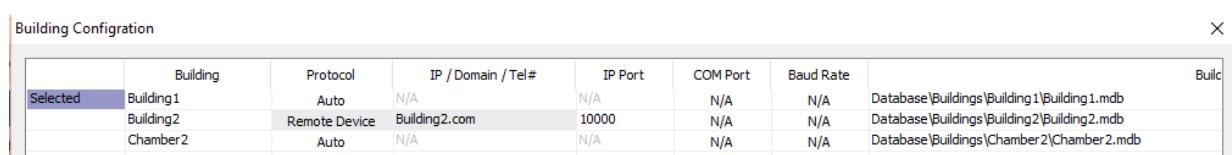
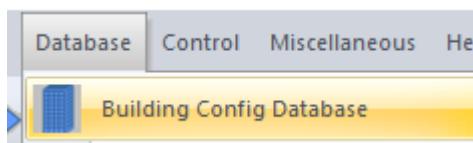


```
T3000_config.ini - Notepad
File Edit Format View Help
[MBPOLL_Setting]
Connection Type=1
COM Port=COM3
COM_Port=3
Baudrate=19200
IP Address=192.168.0.243
IP Port=502
Response Timeout=1000
Delay Between Time=1000
Connect Timeout=3000
[T3000_START]
Interface=12
SerialNumber=65768
[Setting]
Connection Type=1
IP Address=192.168.0.191
```

### System List

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

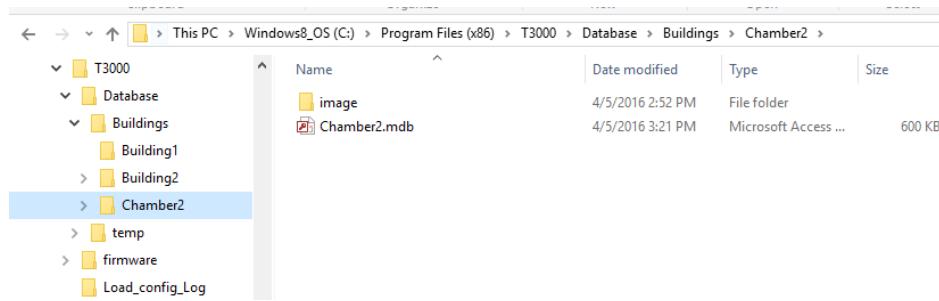
This option is used to make calls to remote systems via the modem. It can be set up with the names and numbers of 24 remote systems. Enter the name, and modem phone number. Press **Ins** anywhere along the line to dial the remote site (the COM ports must be configured first: Chapter 8).



Selected	Building	Protocol	IP / Domain / Tel#	IP Port	COM Port	Baud Rate	Build
	Building1	Auto	N/A	N/A	N/A	N/A	Database\Buildings\Building 1\Building 1.mdb
	Building2	Remote Device	Building2.com	10000	N/A	N/A	Database\Buildings\Building 2\Building 2.mdb
	Chamber2	Auto	N/A	N/A	N/A	N/A	Database\Buildings\Chamber 2\Chamber 2.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'.



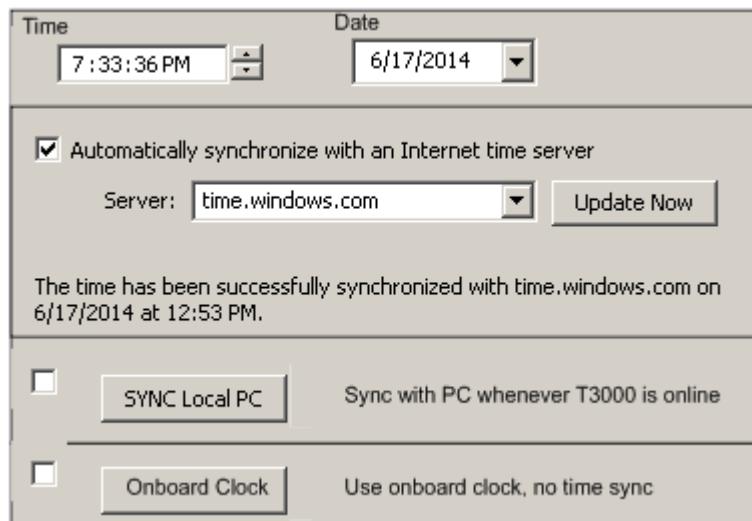
This database is a standard Microsoft Database file format which can be opened and edited in T3000 as well as other mainstream programs such as Excell, Access, SQL, and Openoffice. The MDB format is a legacy file format originally designed for access from a single PC so work is under way to migrate this 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.

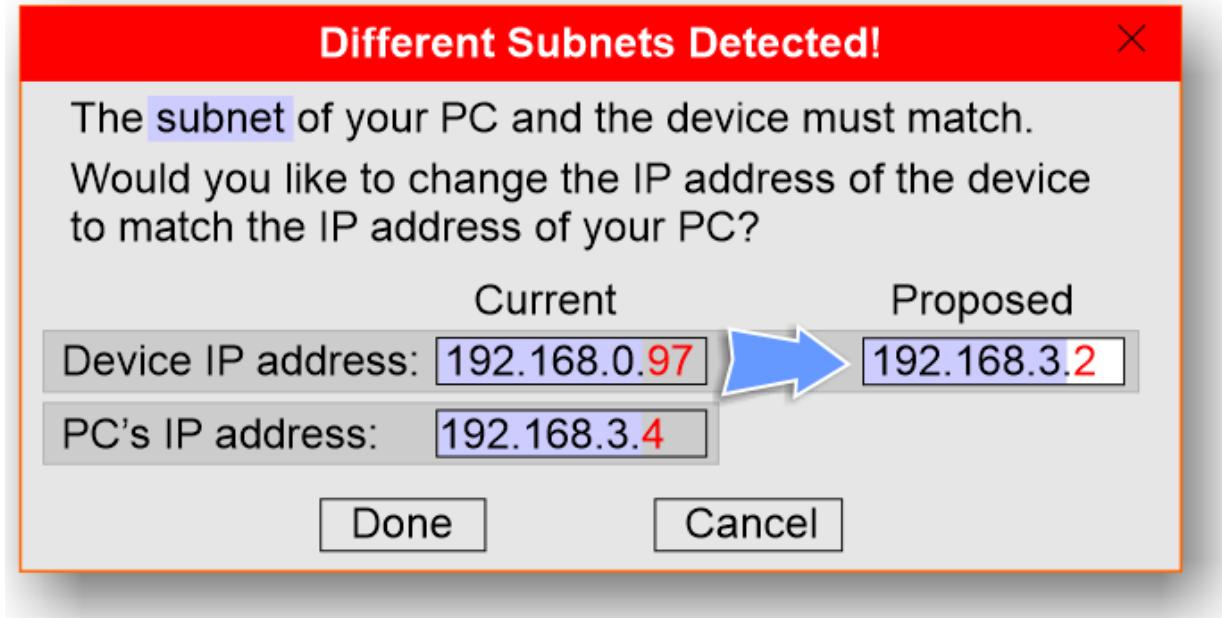
### **BOARDCONFIGURATION**

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

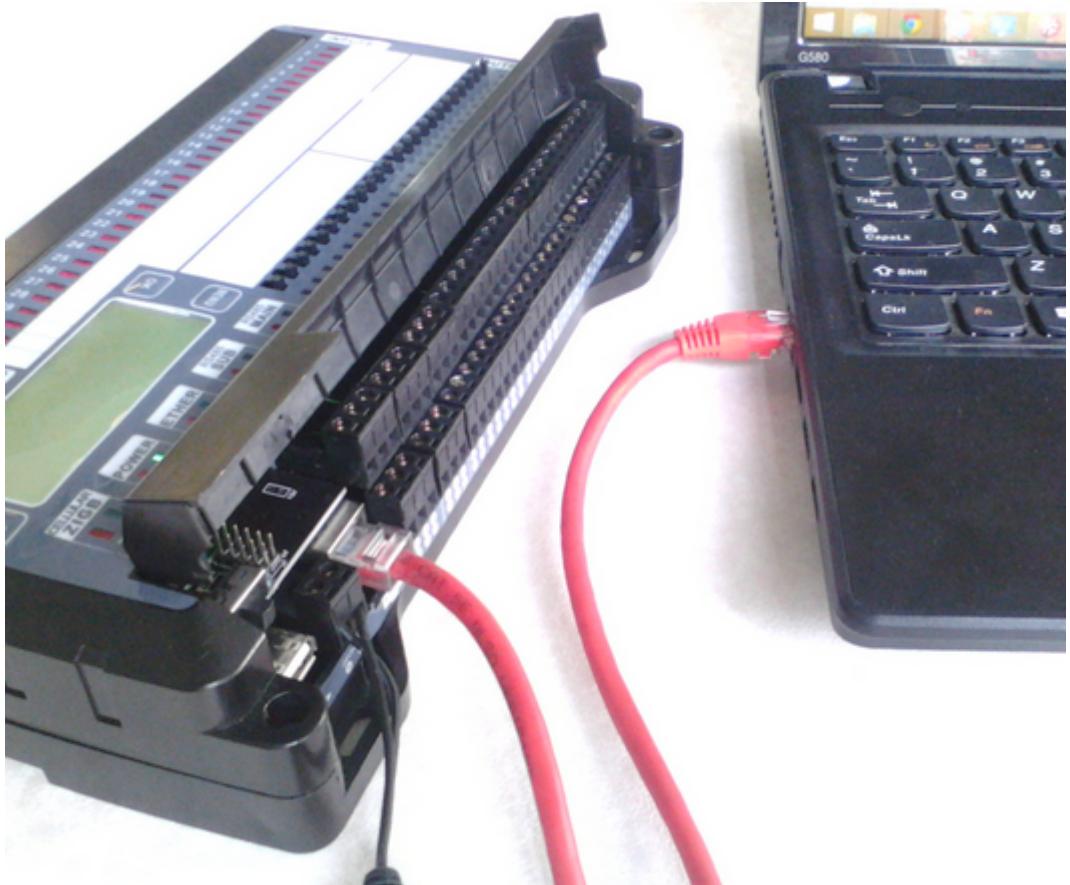
## Troubleshooting Ethernet Connections

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.

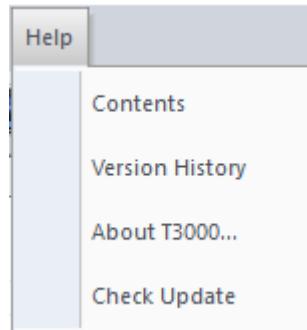


If you're still having problems connecting and don't happen to have a spare hub for isolating from the office/home lan you can try a direct connection as shown here, direct from the PC to the controller with no hub in between. The PC will normally have what is called auto MDI-X capability which means it can detect the direct connection to another host and will automatically flip the connections, no crossover cable is required.



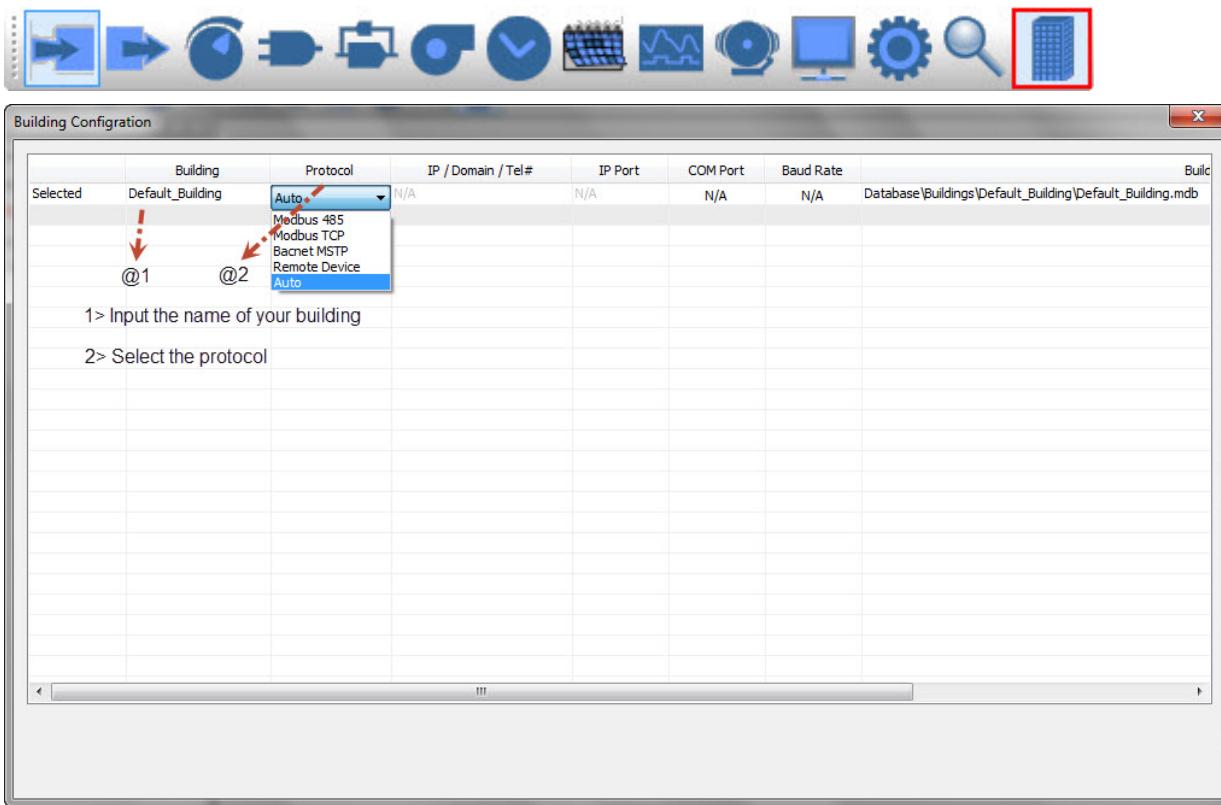
If there are sensors and actuators attached to the controller, try disconnecting them. In any system with many devices and multiple power supplies, it is easy to get 24vac crossed around. Disconnect all the field devices and also the RS485 networks and repeat the scan operation.

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.



## 5 System Setting

### 1. Building Configuration



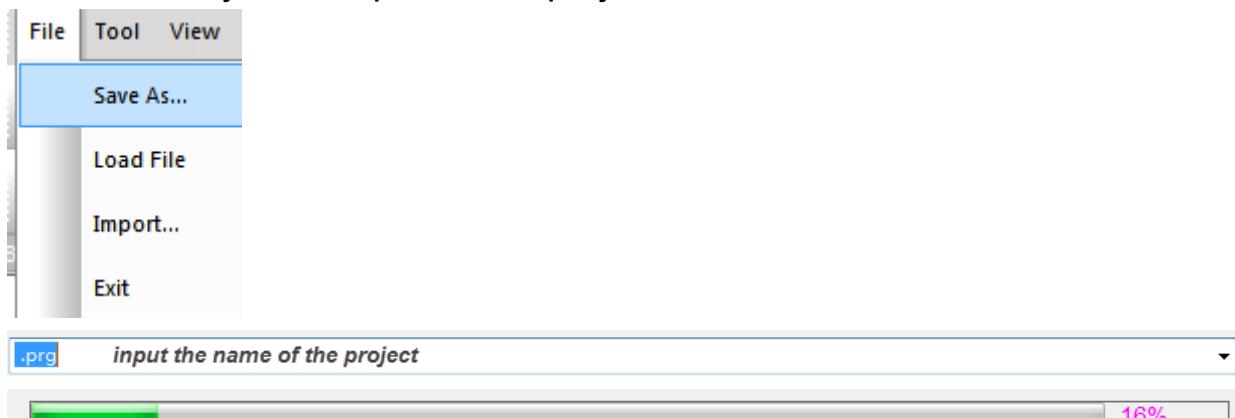
notes: When protocol is auto ,IP ,Port ,Com Port ,Baud Rate disable.

When protocol is Modbus Tcp or Romote Device ,Ip,Port enable ,Comport Baud Rate disable.

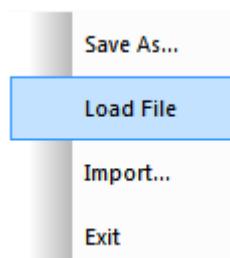
When protocol is Modbus 485 or Bacnet MSTP ,IP,Port disable ,Comport Baud Rate enable.

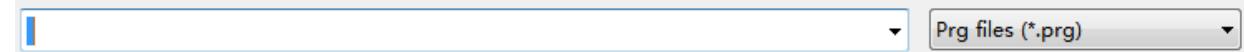
### 2. Configure your MiniPanel

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



2.2: Load the date of a project file to your Minipanel





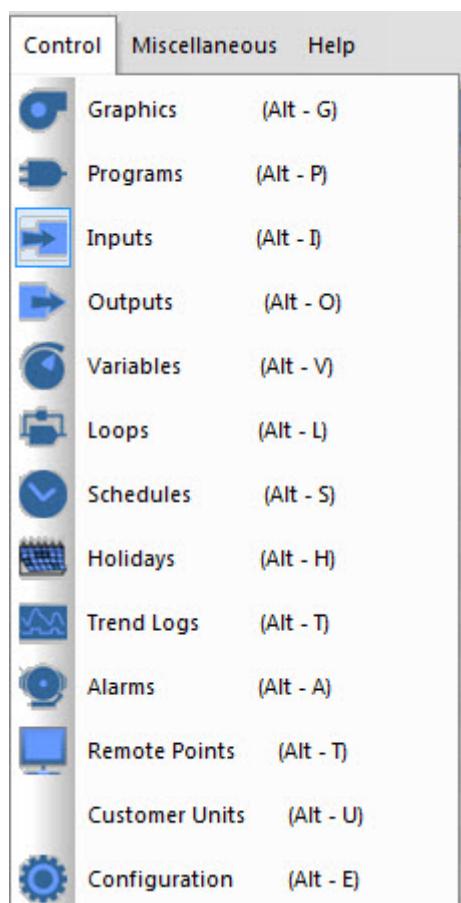
Note: There will be a demo project to show how to do these steps.

## 6 Control

# 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:



## 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

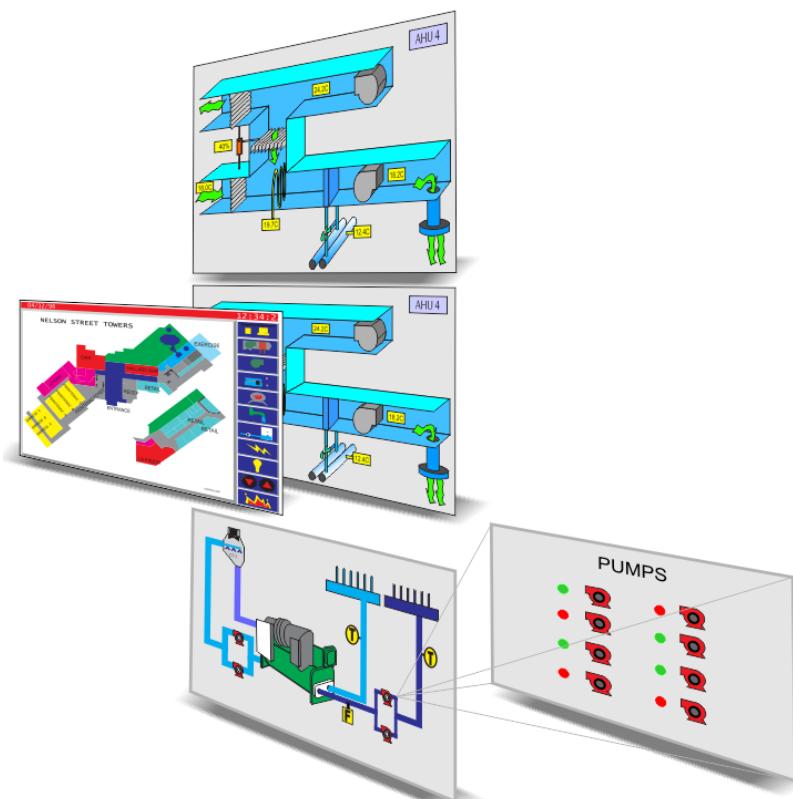
Screens are used to group points of all types into one logical unit for display on the screen. Thus the operator is able to see how a change in one parameter is affecting other related parameters. This enables the operator to see and change the value and auto/manual status of points. The operator can 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.

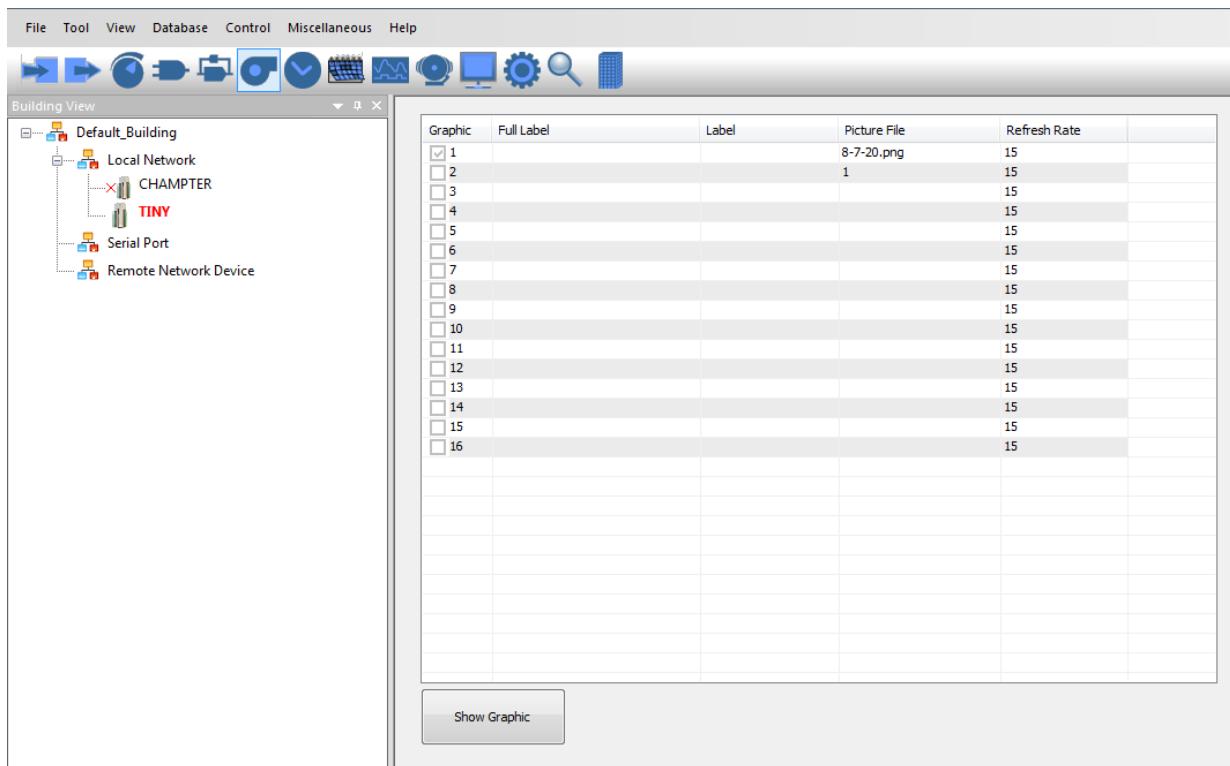


Graphical Screens can be nested so that the operator can drill down to more detailed screens. A large building will typically have an overall floorplan with hot spots leading to each of the zones. Then for major systems there will be icons leading to the chiller room, boiler room and air handlers. The 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.

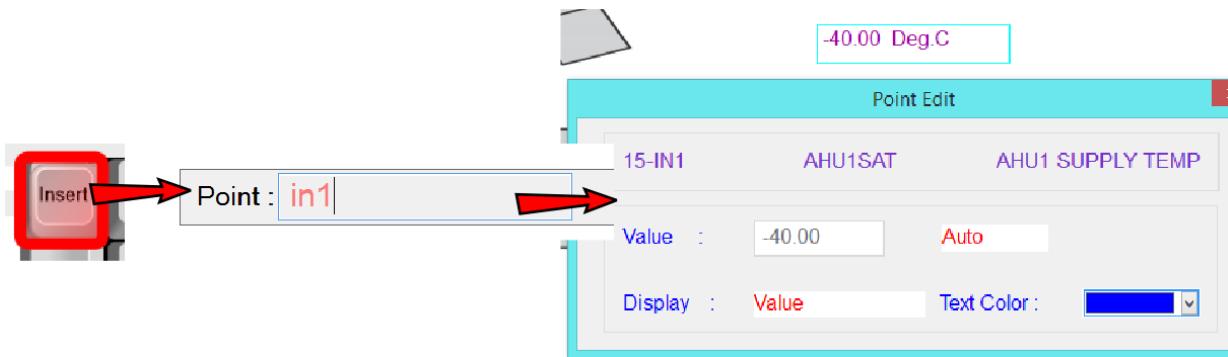
b) Label . . . . . An 8 character label.

[TO BE DONE] Typing this label into the command line lets 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

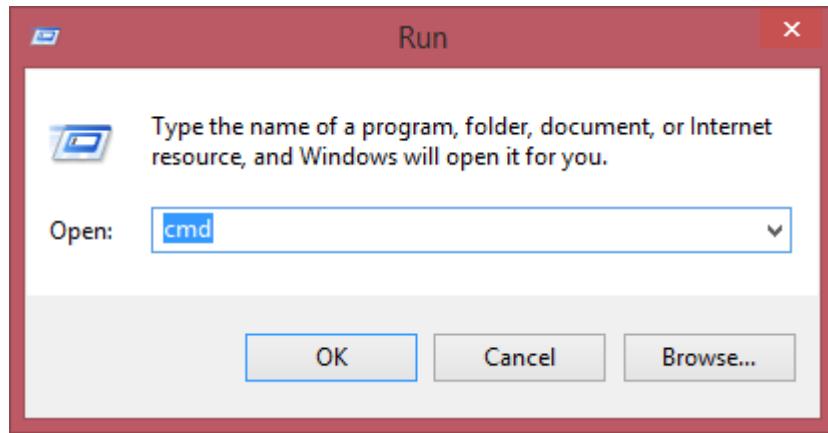
To program or view an individual Screen press "Ins" while highlighting that row. The graphic display pops up, you can add or remove items from the display by hitting the insert key again. and then set how the item will display, such as showing the short or long label text along with the value, the value 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
<input checked="" type="checkbox"/> 1			8-7-20.png	15
<input type="checkbox"/> 2			1	15
<input type="checkbox"/> 3				15
<input type="checkbox"/> 4				15
<input type="checkbox"/> 5				15
<input type="checkbox"/> 6				15
<input type="checkbox"/> 7				15
<input type="checkbox"/> 8				15
<input type="checkbox"/> 9				15
<input type="checkbox"/> 10				15
<input type="checkbox"/> 11				15
<input type="checkbox"/> 12				15
<input type="checkbox"/> 13				15
<input type="checkbox"/> 14				15
<input type="checkbox"/> 15				15
<input type="checkbox"/> 16				15

**Show Graphic**

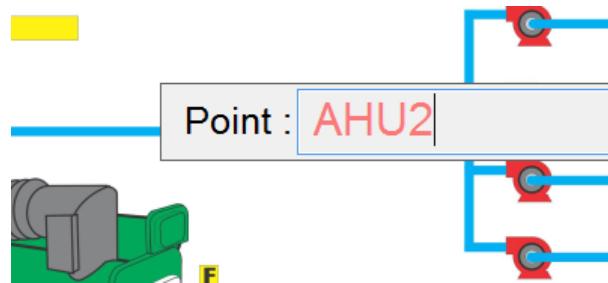
1 **Full Label**  
Full Label

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 which is screen number 2. 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.



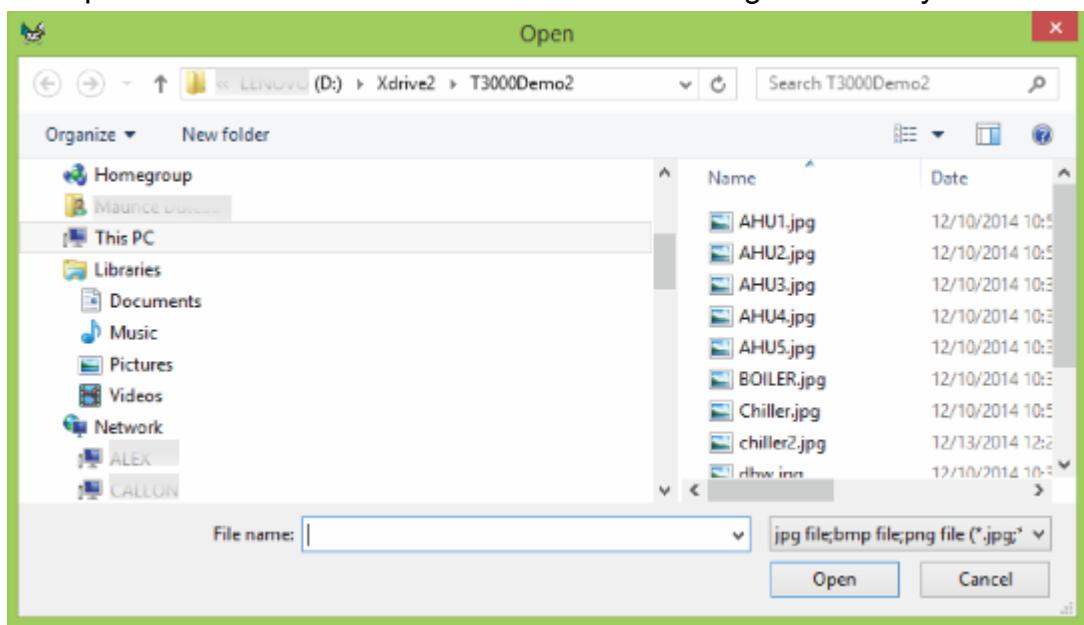
[TBD] Jump to this graphic display from the command line.

### Picture File

3

#### 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.

### Refresh Rate

4

#### 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:

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

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

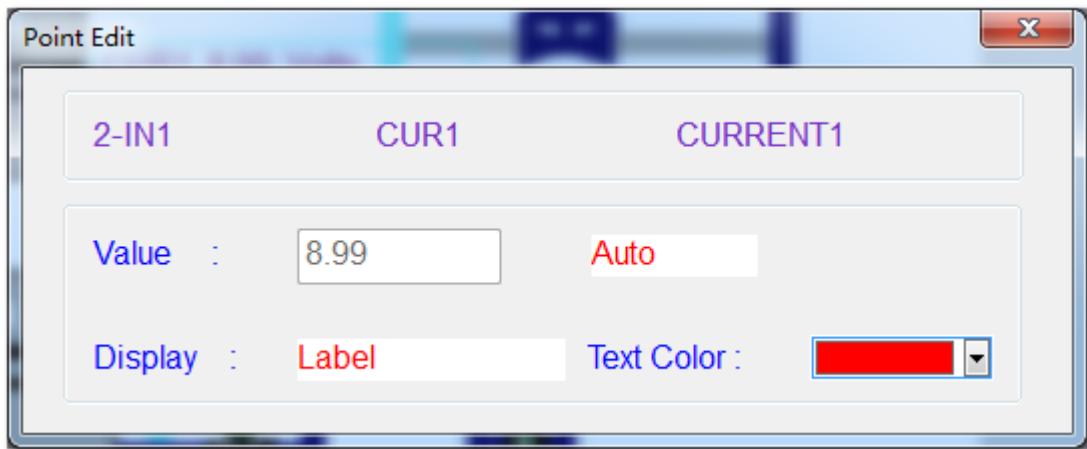
#### **SCREENS NOTES:**

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.

<b>FIELD TITLE</b>	<b>FUNCTION OF FIELD</b>
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 available for displaying the points info:

20 char: Displays 20 character Full label.

**Value:** Displays value of point only.

Display : Value

**Icon:** Displays representative Icon only.

A screenshot of a dialog box titled "Display Icon". It contains four input fields: "Display" (set to "Icon"), "Icon name" (empty), "Text place" (set to "Bottom"), and "Icon size" (set to "Normal").

**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 highlighting 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.

#### 4 Size

Size

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

#### 5 Run Status

Run Status

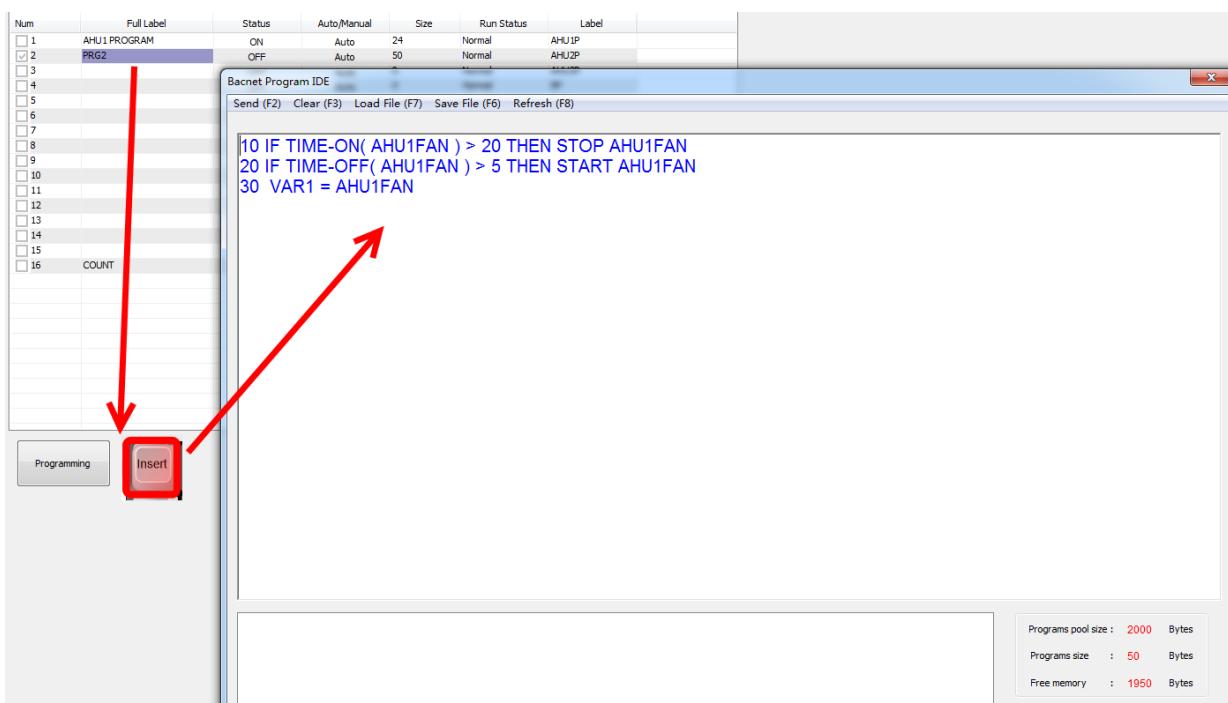
The time between each running of the program (mins: secs).

#### 6 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



## 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:

1.

	FULL LABEL	Auto/Manual	Value	Units	Range	Calibration	Filter		Label	
NUM	Full Label	Auto/Manual	Value	Units	Range	Calibration	Sign	Filter	Status	Label
1	ELEVATOR FLOOR3	Manual	On		On/Off			64	Open	FLOOR3
2	AHU1 FAIL MODE	Auto	Normal		Normal/Alarm			64	Normal	AHU1FAIL
3	CHILLER FAIL	Auto	Alarm		Normal/Alarm			64	Normal	CHFAIL
4	AHU1 OCCUPIED SWITCH	Manual	On		Off/On			64	Open	AHU10CC
5	WATER LEVEL	Auto	On		On/Off			64	Normal	WATERHI
6	ELEVATOR FLOOR1	Auto	On		Off/On			64	Normal	FLOOR1
7	AHU2 SUPPLY TEMP1	Auto	22.30	Deg.C	10K -40 to 120	0.0	-	64	Normal	AHU2SAT
8	ELEVATOR FLOOR4	Auto	On		Off/On			64	Normal	FLOOR4
9	RM301 OCCUPIED	Auto	No		No/Yes			64	Normal	RM301OCC
10	OUTSIDE AIR TEMP	Auto	-32.20	Deg C	10K -40 to 120	0.0	-	64	Open	OAT
11	OUTSIDE LIGHT SENSOR	Auto	2.81	Volts	0.0 to 5.0	0.0	-	64	Open	OL
12	RM401 KEYSWITCH	Auto	On		On/Off			64	Normal	RM401KEY
13	ROOM 401 OCCUPIED	Manual	On		Off/On			64	Open	RM401OCC
14	EF1 SMOKE DAMPER	Auto	On		Off/On			64	Normal	EF1SMOKE
15	AHU2 SMOKE DAMPER	Auto	On		Off/On			64	Normal	AHU2SMOK
16	AHU1 SMOKE DAMPER	Auto	On		Off/On			64	Normal	AHU1SMOK
17	CONDENSOR SUPP TEMF	Auto	-40.00	Deg.C	10K -40 to 120	0.0	+	0	Open	CWST
18	ELEVATOR FLOOR2	Auto	On		Off/On			0	Normal	ELEV2
19	ELEVATOR FLOOR3	Auto	On		Off/On			0	Normal	ELEV3
20	ELEVATOR FLOOR4	Auto	On		Off/On			0	Normal	ELEV4
21	AHU1 FIRE ALARM	Auto	On		Off/On			0	Normal	AHU1FIRE
22	AHU2 FIRE ALARM	Auto	On		Off/On			0	Normal	AHU2FIRE
23	DAYTIME LIGHT1 STAT	Auto	On		Off/On			0	Normal	LIGHT1
24	DAYTIME LIGHT2 STAT	Auto	On		Off/On			0	Normal	LIGHT2
25	NITE LIGHT STATUS	Auto	On		Off/On			0	Normal	LIGHT3
26	EMERGENCY LITE 1	Auto	On		Off/On			0	Normal	ELIGHT1
27	BOILER1 STATUS	Auto	On		Off/On			0	Normal	B1S
28	CHILLER STATUS	Auto	On		Off/On			0	Normal	CH1S

## 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.

A 20 character descriptor which is usually used to associate a given point with the physical location.

## 2 Auto/Manual

The auto/manual field is a feature to let 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.

The manual mode is useful 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. 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.

## 3 Value

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. Hitting enter of the value field, when the input is in manual mode, will toggle the value from on to off and back. 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.

#### 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 input throughout the system: programming, trend logs and so on.

#### 6 Range

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

The following analog ranges are available:

1. Y3K -40 to 150 Deg.C
2. Y3K -40 to 300 Deg.F
3. 10K -40 to 120 Deg.C
4. 10K -40 to 150 Deg.F
5. G3K -40 to 120 Deg.C
6. G3K -40 to 250 Deg.F
7. KM10K -40 to 120 Deg.C (used for Temco LCD sensor)
8. KM10K -40 to 250 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

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

The following digital input ranges are available. The word on the left will show when the input is zero circuit. The word on the right side of the slash will show when the input is open circuit. If the contact 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 logic and when the input is 3V or open circuit, the word On will show everywhere in the system. If the contact this then select 12, now the word On will show when the input is short circuit and Off will show when the

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

Ranges 23 thru 30 are Custom Digital Ranges.

TIP: The status LED's for each input are illuminated when there is an open circuit and off when the input is shorted. This can be most intuitive to technicians and personnel in the field if the LED is on when the equipment is on and off when the equipment is off. To achieve this you can wire the status to a normally closed set of contacts. When the equipment is off the contacts short circuits the input and the LED shows off. When the equipment is on, the relay is open and the LED is on.

### Select Range Number



Enter Units Number:

33

OK

Cancel

10K -40 to 120

#### 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. /

#### Input Analog Units

- 31. Y3K -40 to 150 Deg.C
- 32. Y3K -40 to 300 Deg.F
- 33. 10K -40 to 120 Deg.C
- 34. 10K -40 to 250 Deg.F
- 35. G3K -40 to 120 Deg.C
- 36. G3K -40 to 250 Deg.F
- 37. KM10K -40 to 120 Deg.C
- 38. KM10K -40 to 250 Deg.F
- 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. 0.0 to  $2^{22}$  counts
- 46. 0.0 to 3000 FPM
- 47. 0 to 100 %(0-5V)
- 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 factory is the 'thermistors & dry contacts' setting since these are the most common signal type in this setting a 10k pullup resistor to 3V creates a weak pullup voltage for excitation for the field device. The 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 CO<sub>2</sub> sensors. The mA setting for this type of transducer. And finally, by removing the jumper altogether the field device goes to the a/d converter of the controller. The CPU handling the a/d conversions is a three volt chip therefore the signal you should apply to the input is three volts. If you have a five volt signal you can select the 10V setting.

7

## Calibration

### Calibration

e) Cal.(Calibration).. The value in this field is added into the measured value to adjust for sensor inaccuracy. The maximum correction is +/- 30. If the correction amount is between -3.0 and +3.0 there is 1 decimal point in the correction, otherwise there is no decimal point.

Calibration (CAL) Adds or subtracts an offset to the analog input readings for calibration purposes. In the example below, the input shows 24.1C but our reference sensor shows 25.1C, we enter a 1 in the calibration field and now the controller and meter values agree at 25.1C.

The resolution of the calibration field is 1 decimal place. 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

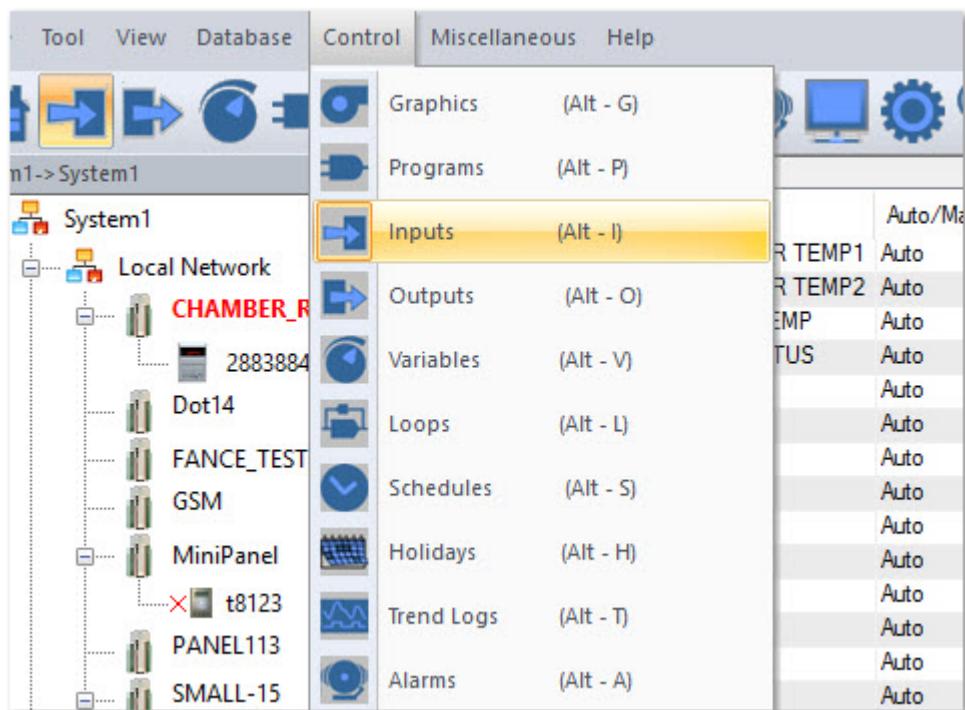
8

## 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.

## Menu



## Pulse Counting

<TODO>:Insert pulse counting details 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 I/O
MP24	32	12	12	32
MP10	16	6	4	48
MP8	8	6	2	56

Sample Output work screen:

NUM	Full Label	Auto/Manual	Value	Units	Range	0%	100%	Status	Label	HOA Switch
1	AHU1 SUPPLY FAN	On			Off/On			OK	AHU1FAN	MAN-OFF
2	AHU1 RETURN FAN	Off			Off/On			OK	AHU1RET	MAN-OFF
3	AHU1 MIXED AIR DAMP	On			Off/On			OK	AHU1MAD	MAN-OFF
4	AHU1 HEATING VALVE	0.00	%Open		0.0 -> 100	0	0	OK	AHU1HEAT	MAN-OFF
5	AHU1 COOLING VALVE	0.00	%Open		0.0 -> 100	0	0	OK	AHU1COOL	MAN-OFF
6								OK		MAN-OFF
7								OK		MAN-OFF
8								OK		MAN-OFF
9								OK		MAN-OFF
10								OK		MAN-OFF
11			0.00			0	0	OK		
12			0.00			0	0	OK		
13			0.00			0	0	OK		
14			0.00			0	0	OK		
15			0.00			0	0	OK		
16								OK		
17								OK		
18								OK		
19								OK		
20								OK		
21								OK		
22								OK		
23								OK		
24								OK		
25								OK		
26								OK		
27								OK		

### 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.

A 20 character descriptor which is usually used to associate a given point with the physical location.

## 2 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 output 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):

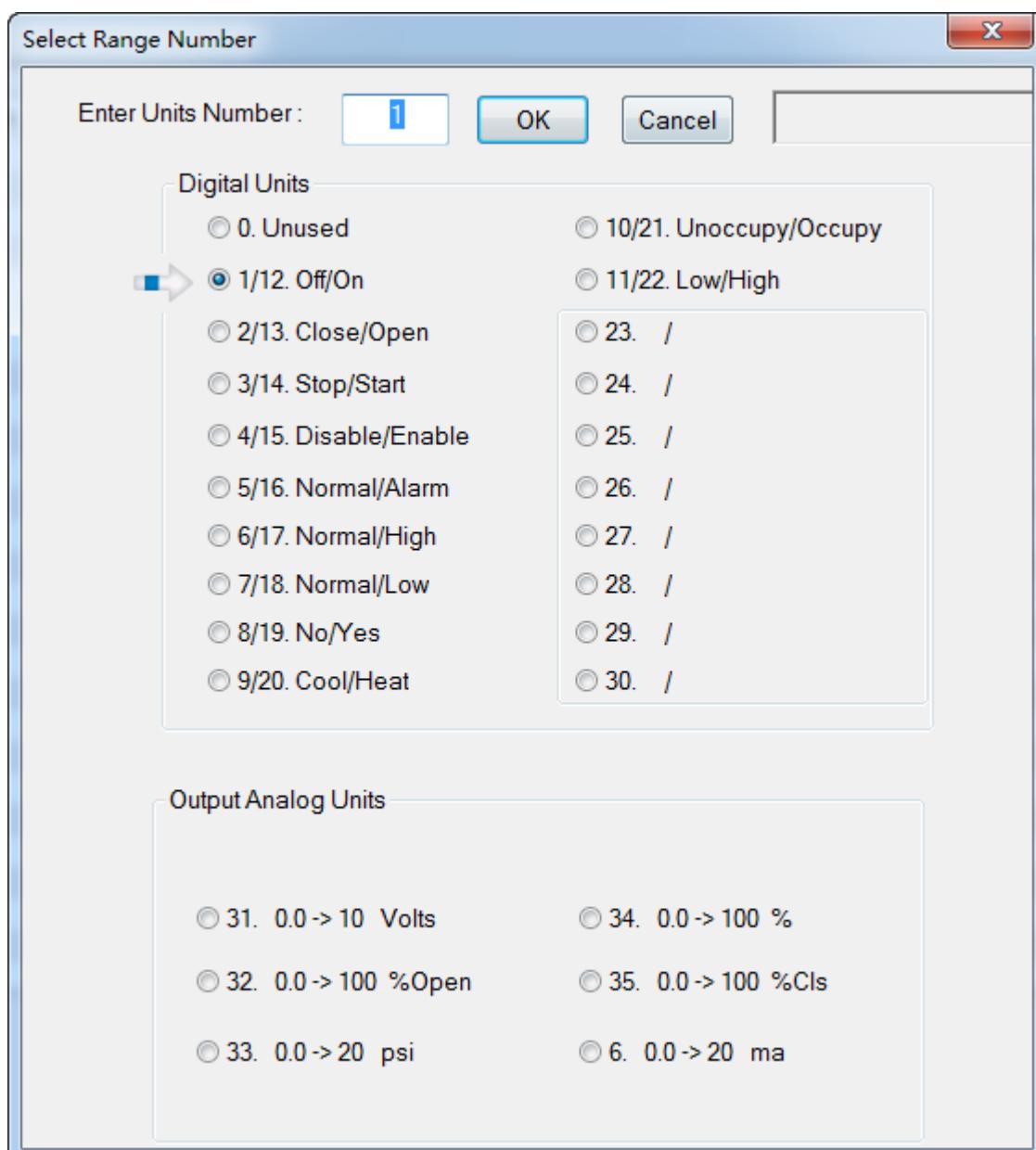
1/12Off/On2/13Close/Open3/14Stop/Start4/15Dis/Enable  
5/16Norm/Alarm6/17Norm/High7/18Norm/Low8/19No/Yes  
9/20 Cool/Heat10/21 Un/Occupied 11/22 Low/High

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



When the range selected is 0-100 % or 0-100 % open, “Low” sets the output voltage when the value is 0 %. Note that when the value is 1 % the voltage output will be either 0 V (if Low is less than High) or 12 V (if Low is greater than High). This is to ensure that the output transducer is driven fully to the 0 % position.

8 100%

100%

When the range selected is 0- 100 % or 0- 100 % open, “High” sets the output voltage when the value is 100 %. Note that when the value is 101 % the voltage output will be either 12V (if “High” is greater than “Low”) or 0 V (if “High” is less than “Low”). This is to ensure that the output transducer is driven fully to the 100% position.

9 Status

Status

10 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
Variable	Full Label	Auto/Manual	Value	Units	Label
VAR 1	TESTO METER HUM	Auto	47.900	%RH	TESTOHUM
VAR 2	TESTO METER TEMP	Auto	29.400	Deg.C	TESTOTMP
VAR 3	HEATMODE	Auto	On	Off/On	HEATMODE
VAR 4	COOLMODE	Auto	Off	Off/On	COOLMODE
VAR 5	AVERAGE CHAMBER TEMP	Auto	29.400	Deg.C	CHAMTEMP
VAR 6	CHAMBER TEMP SETPT	Manual	50.000	Deg.C	TEMPSETP
VAR 7	CHAMBER HUM SETPT	Manual	80.000	%	HUMSETP
VAR 8	SYSTEM ON	Manual	On	Off/On	SYSTEMON
VAR 9	CHANGEOVER FLAG	Auto	Off	Off/On	CHANGFLG
VAR 10	CHANGEOVER TIMER	Auto	00:00	Time	CHANGTMR
VAR 11	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 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
...	...	...	...	...

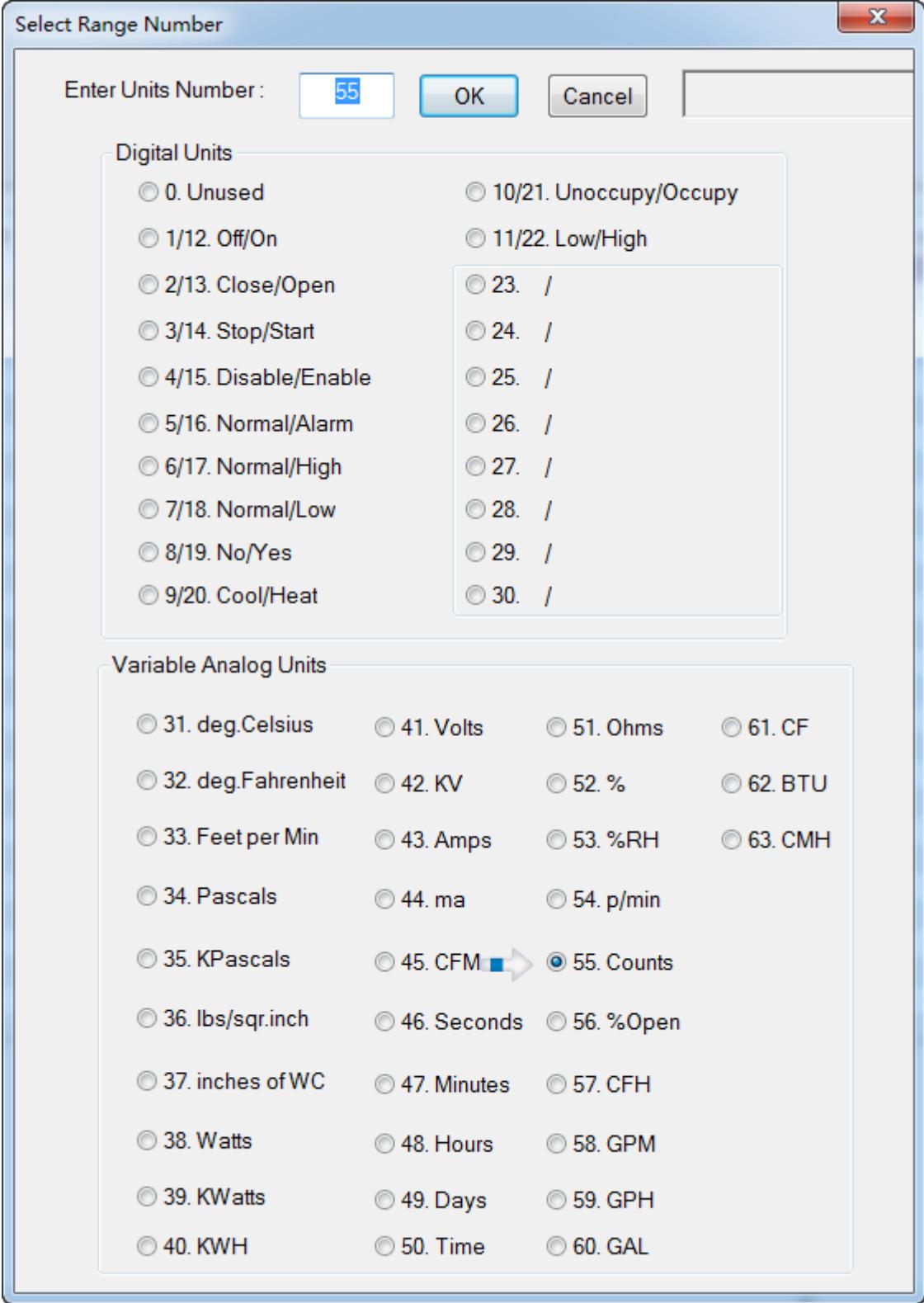
First move the cursor to the row for the outside temperature sensor, IN3 in this case. Toggle the Auto/Manual field to Manual as shown in the green cell. Then maneuver right to the red tinted

'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



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

- |                   |                     |                  |           |                       |
|-------------------|---------------------|------------------|-----------|-----------------------|
| 1. Off/On         | 2.                  | Close/Open       | 3.        | 4. Dis/<br>Stop/Start |
| 5. Norm/<br>Alarm | 6.                  | Norm/High        | 7.        | Enable<br>Norm/Low    |
| 9. Cool/Heat      | 10. Un/<br>Occupied | 11. Low/<br>High | 8. No/Yes |                       |

The following  
**inverted**

digital output  
ranges are  
available:

12. On/Off	17. High/ Norm	13. Open/ Close	14. Start/ Stop	15. Enable/Dis
16. Alarm/ Norm	18. Low/ Norm	19. Yes/No		
20. Heat/Cool Units 23 thru 30 are Custom Digital Ranges.	21.	22. High/ Occupied/Un Low		

5

### Label

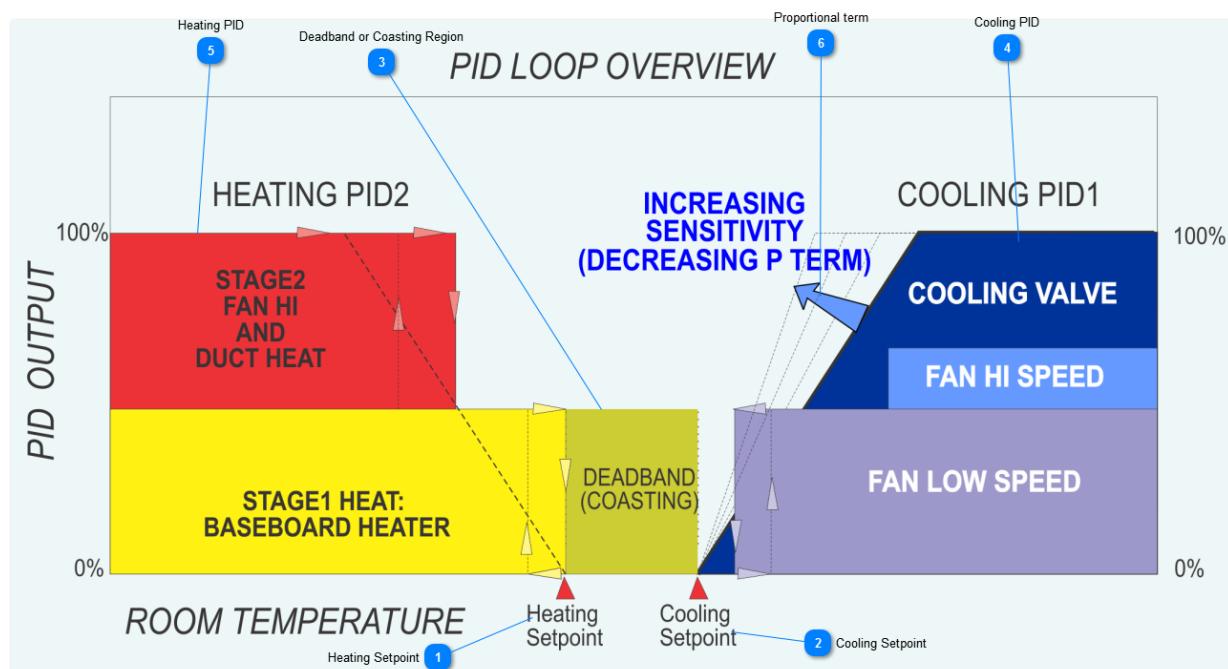
Eight character descriptor, a unique ID used to refer to this particular item throughout the system during programming, creating trend logs and so on.



## 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



**Heating Setpoint**

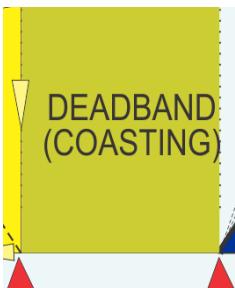
The heating setpoint is shown here at pointer #1. In this example we have a separate heating and cooling setpoint since there is both heating and cooling in this application. The heating setpoint will be a variable which is normally adjusted warmer for the occupied period and cooler for the unoccupied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

### 2 Cooling Setpoint

# Cooling Setpoint

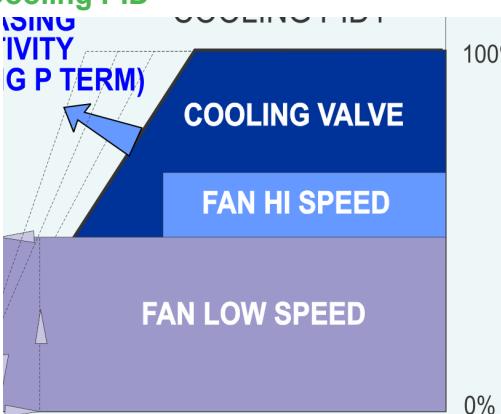
The cooling setpoint is shown here, it is also typically a variable in the T3 system which will be adjusted and managed by programs and schedules elsewhere in the T3 system according to the occupied and unoccupied period among other things. If the building takes advantage of load shedding the user program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

Input or Feedback for the PID loop			PID Loop Current Output			Proportional, Integral and Derivative Terms							
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

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

Input	Value	Units
CHAMTEMP	62.4	Deg.C
CHAMTEMP	62.4	Deg.C

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

A/M	Output
Auto	100.0%
Auto	0.0%

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.

A/M	Output
Manual	55%
Auto	0.0%

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

Setpoint	Set Value	Units
TEMPSETP	50.0	Deg.C
TEMPSETP	50.0	Deg.C
HUMSETP	80.0	%

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 the 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

Prop	Int	Der
1	0	0
1	0	0

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.

## Action

5

### 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.

## Bias

6

### 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:

$$\text{OUTPUT} = \text{Op} + \text{Oi} + \text{Od} + \text{Ob}$$

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.

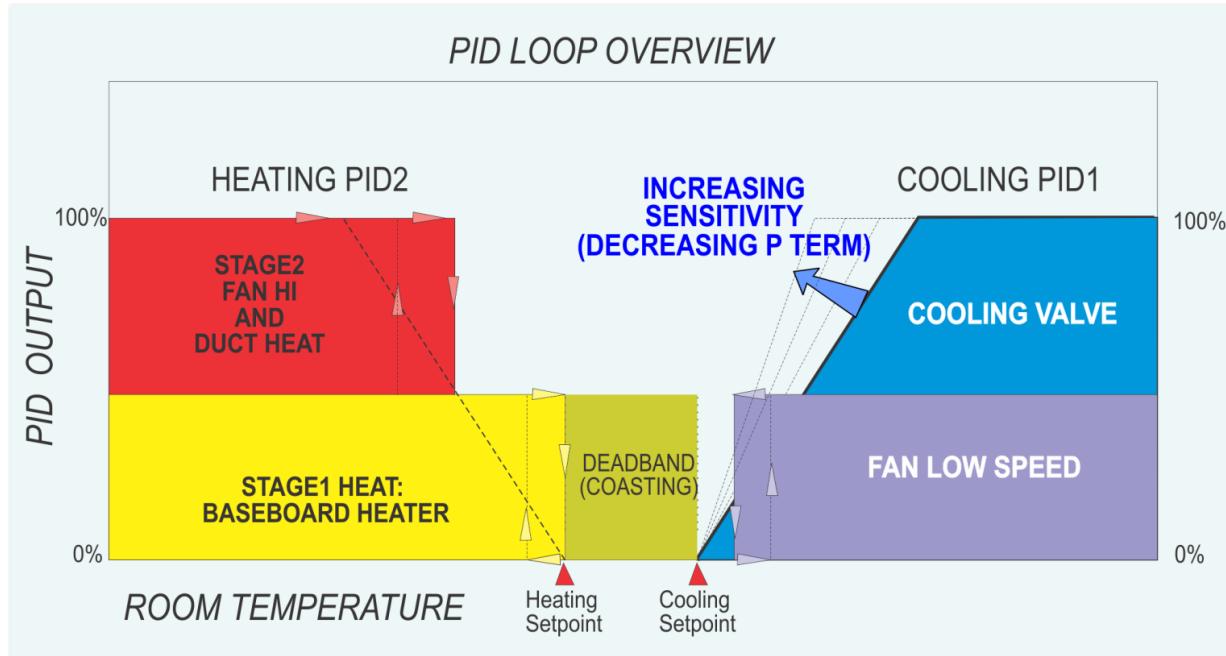
Output Limiting Procedure:

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

## 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:

$$\text{OUTPUT} = \text{Op} + \text{Oi} + \text{Od} + \text{Ob}$$

$$\text{Op} = \text{Proportional Output} = A * 100 / \text{Pb} * (I - S)$$

$$\text{Oi} = \text{Integral Output} = A * R * t / 60 * (I - S) + \text{Oi}$$

$$\text{Od} = \text{Derivative Output} = A * D * ((I - S) - (I_{t-1} - S_{t-1})) / t$$

$$\text{Ob} = \text{Bias}$$

$$A = \text{Action (+1 for direct acting, -1 for reverse acting)}$$

$$\text{Pb} = \text{Proportional Band}$$

$$I = \text{Input}$$

$$S = \text{Setpoint}$$

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.

I<sub>t-1</sub> = Input on last scan.

S<sub>t-1</sub> = 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 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	Holiday1	State1	Holiday2	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								

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

Holiday1

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:

NUM	Full Label	Auto/Manual		Value	Label
		Auto/Manual	Value		
1		Auto	OFF		
2		Auto	OFF		
3		Manual	OFF		
4		Auto	OFF		
5		Auto	OFF		
6		Auto	OFF		
7		Auto	OFF		
8		Auto	OFF		

The screenshot displays the HMI's holiday configuration screen. At the top, there's a table with columns: NUM, Full Label, Auto/Manual, Value, and Label. The 'Label' column is currently empty. Row 3 is highlighted with a blue background, and its 'Full Label' value is 'CHILY'. A context menu is open at the bottom-left of this row, containing options like 'Edit Calendar Day', 'Insert', 'Home', 'Page Up', 'Delete', 'End', and 'Page Down'. A large red arrow points from the 'Label' column in the table to a large calendar window titled 'Dialog' that spans across the bottom of the screen. This calendar shows the months from January 2015 to December 2015, with specific dates highlighted in yellow to indicate where the 'CHILY' label is applied.

### 1 FULL LABEL

1

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

2

### Auto/Manual

Controls whether the routine is operational or not.

### 3 Value

#### Value

The state of the routine (ON/OFF).

### 4 Label

#### Label

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

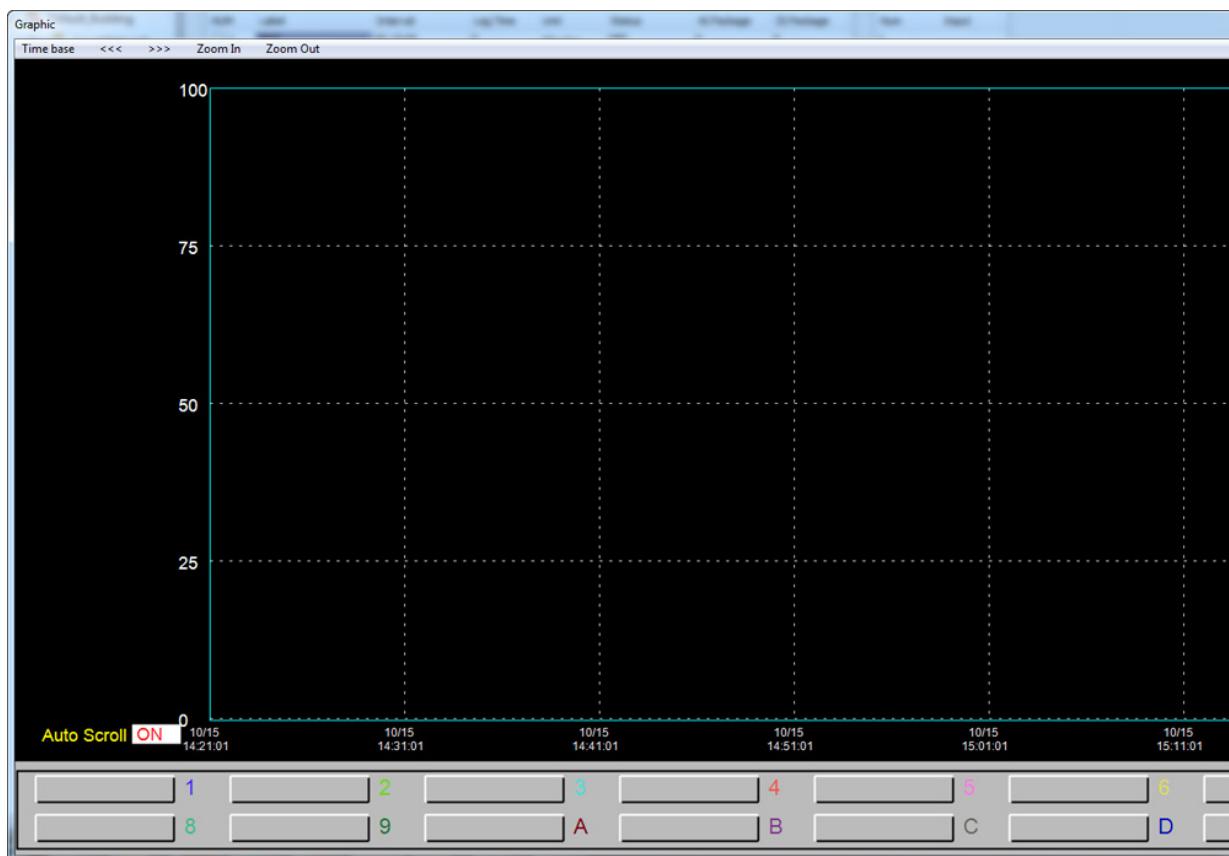
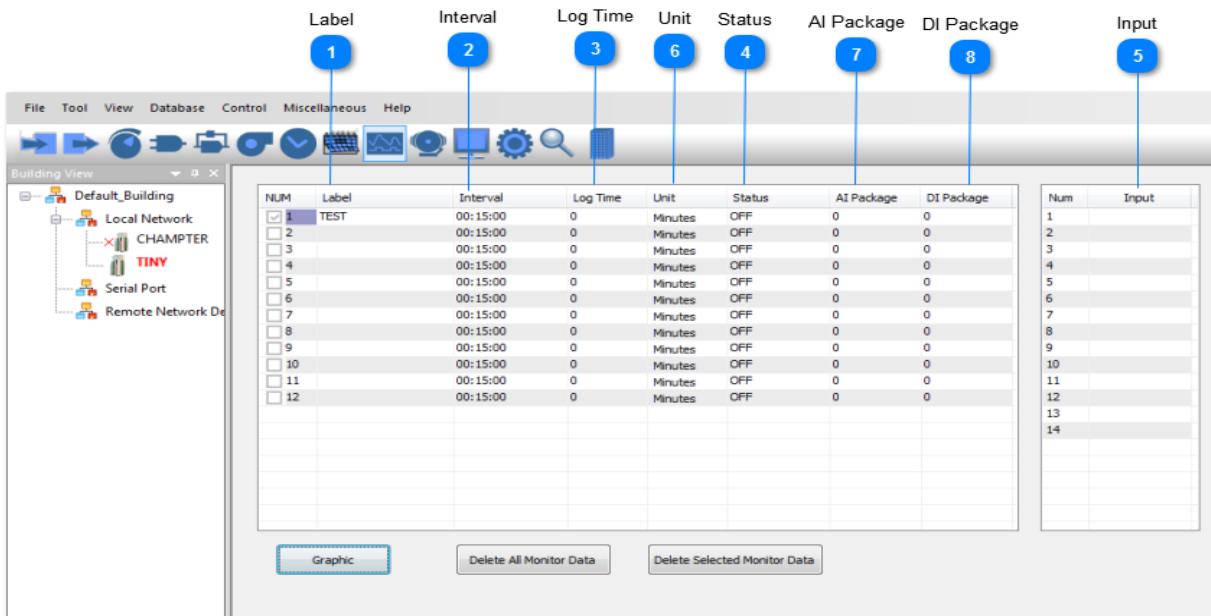
To program the times for an individual routine press the "INS" key while highlighting that point. The year is then displayed in 4 calendar format pages (use PgUp and PgDn keys to change page) with months available on each page. Work menu operations are again available. To set a day as a holiday move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

[Log Time](#)

Time interval for which dates are maintained by the monitor. Expressed in the units of time defined by the “Units” field

## 4 Status

[Status](#)

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

## 5 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

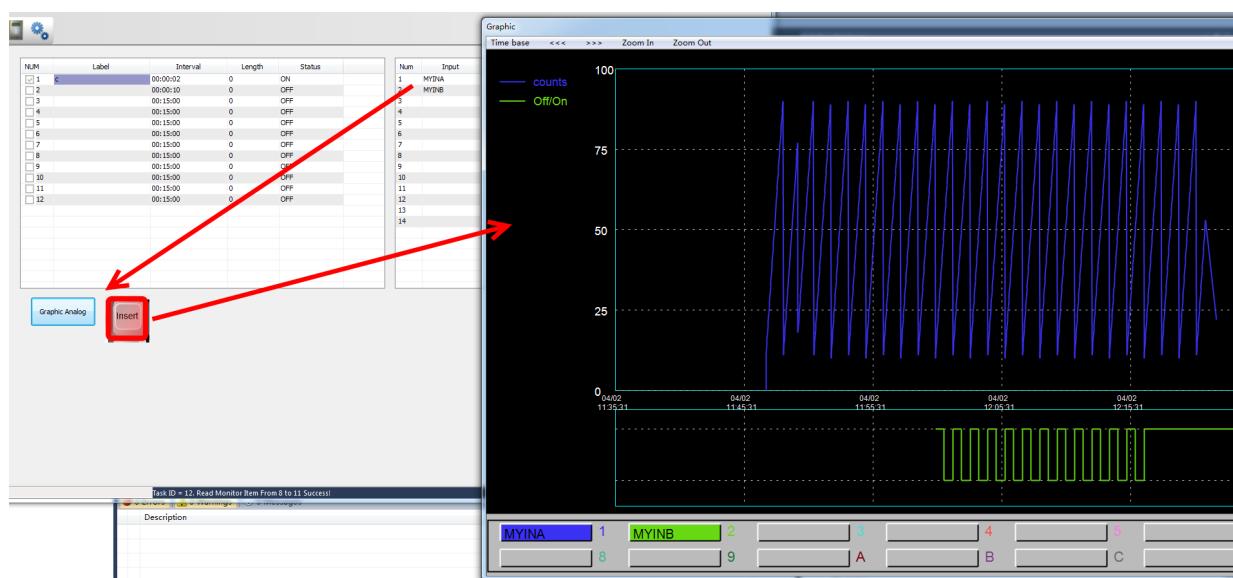
key is used to return to the main monitor work screen. Up to 14 points can be monitored here per monitor. Of these 14 points

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 happens the Digital points over 5 will be deleted from the monitor.

An Analog Monitor can also be commanded to sample from Control Basic by setting it to ON or OFF ( i.e. amon1 = ON ). When this occurs the monitor samples all its inputs 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 current 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 after Zoom In has been used to restore the graph to its original display mode.

6

### **Unit**

Unit

Minutes,Hours#Days

7

### **AI Package**

AI Package

The Total of Analog Input Packages in the state of monitor

8

### **DI Package**

DI Package

The Total of Digital Input Packages in the state of monitor

## 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](#).

## Panel Configuration

### 7.1.1 Set your device IP address

Obtain IP Address Automatically

Use The Following IP Address

IP Address : 192 . 168 . 0 . 14

Subnet Mask : 255 . 255 . 255 . 0

Gateway Address : 192 . 168 . 0 . 4

### 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

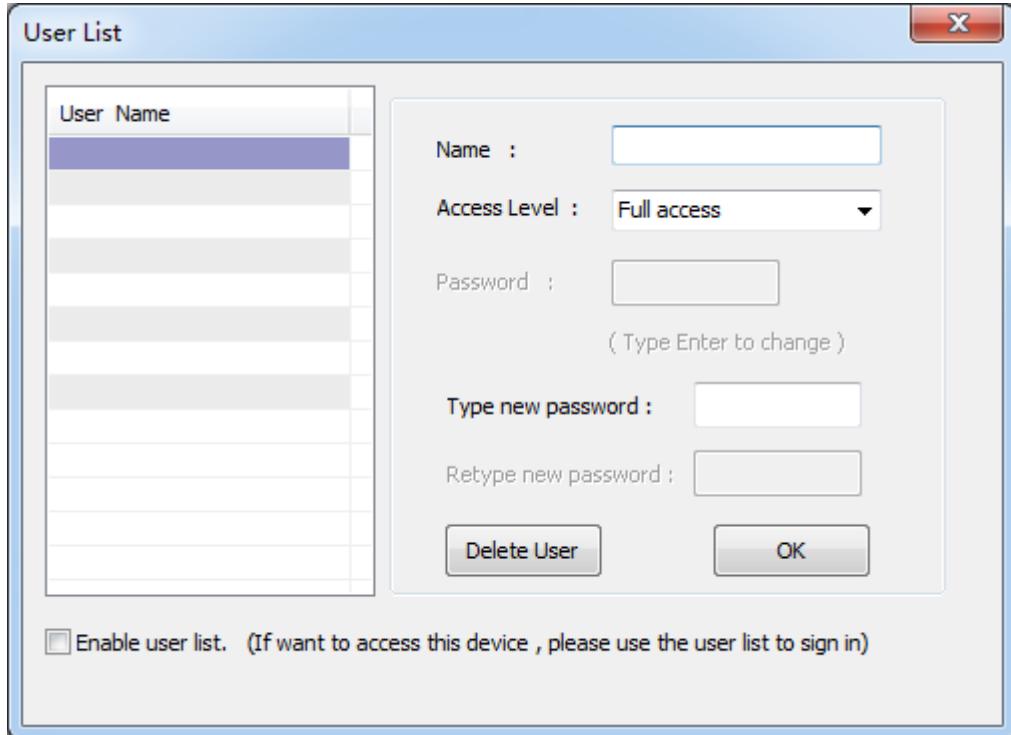
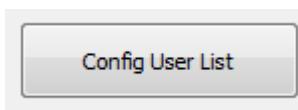
Module Number :	MP24
Hardware version :	21
MCU version :	37.3
PIC version :	0
C8051 version :	0
SM5964 version :	0
Bootloader version :	23
Serial Number :	65598

### 7.1.5 PANEL ID

Panel Number : 2

Panel Name : PANEL14

## 7.2 USER LIST

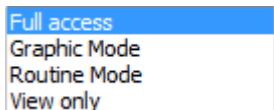


User List provides the means for system security and individual access to the system. Each user has to Sign-On with his name and give his password to gain access to the system. There is space to allow for 10 different users and passwords (8 for the Mini-Panel).

#### 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

Dyndns Config

Enable Dyndns Service

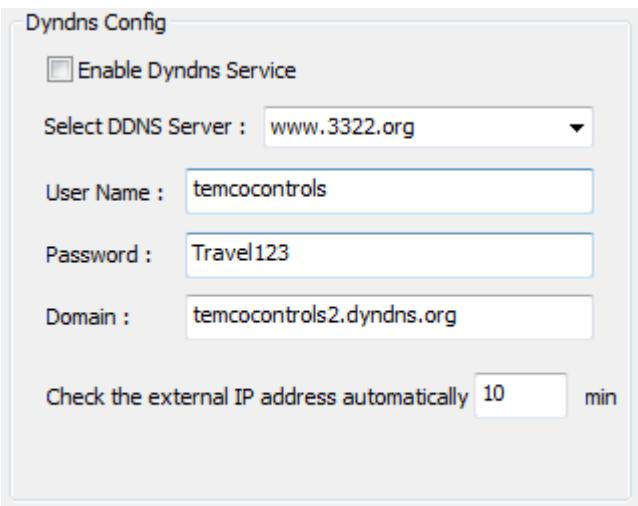
Select DDNS Server : [www.3322.org](http://www.3322.org)

User Name : temcocontrols

Password : Travel123

Domain : temcocontrols2.dyndns.org

Check the external IP address automatically  min



## 8.4 Device Serial Port Configuration

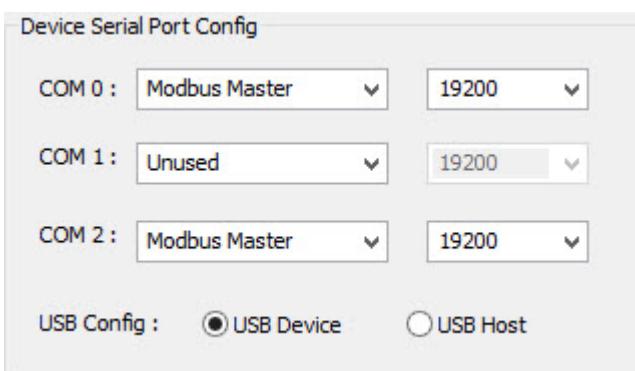
Device Serial Port Config

COM 0 : Modbus Master  19200

COM 1 : Unused  19200

COM 2 : Modbus Master  19200

USB Config :  USB Device  USB Host



**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 .**



## 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

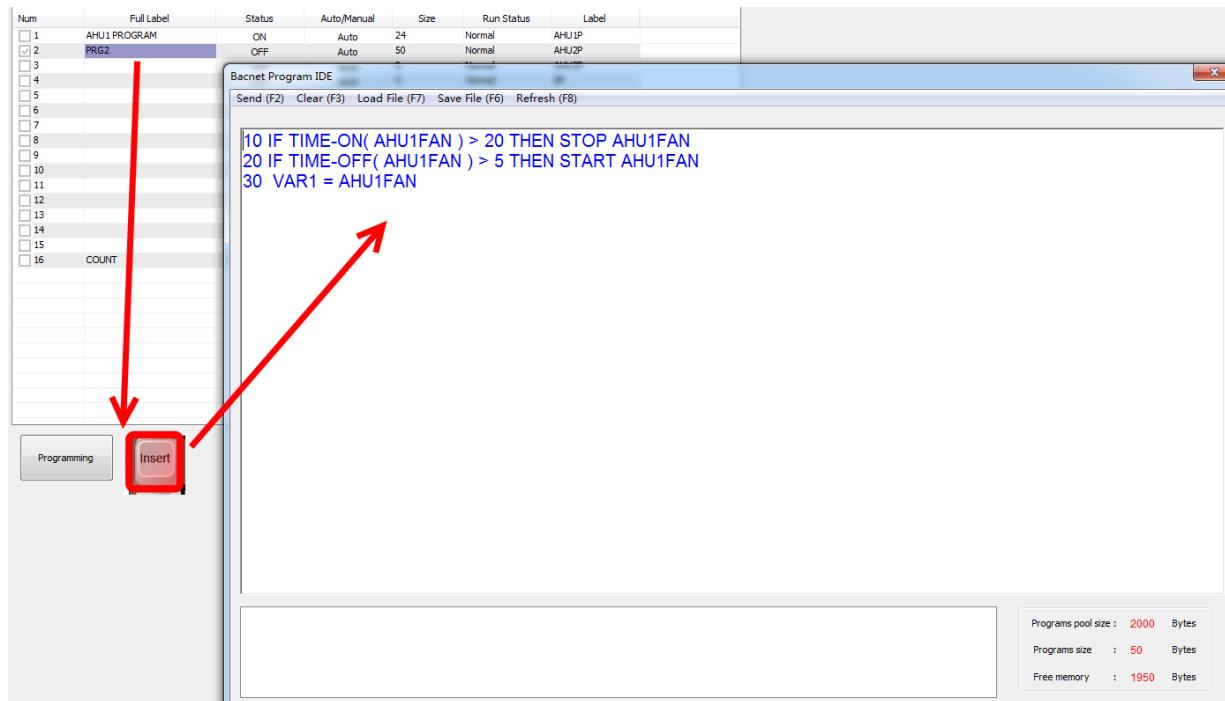
And here's a simple program to open an on/off type valve to heat a room for example.  
IF TEMP101 < SETPOINT THEN OPEN VALVE101  
IF TEMP101 > SETPOINT + 0.5 THEN CLOSE VALVE101

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 replaces 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 allowing 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 extension. 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.

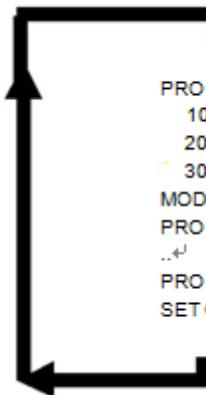
For the most part, program logic in the T3000 panel follows through from the top of the program to the last line in sequential order. Since Basic supports the GOTO command, but program execution can actually take any path that you desire, but GOTO statements can cause a program to be difficult to follow.

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



```

PROGRAM #1↓
10 REM :***** SF1 *****
20 IF OAT > 20 THEN START COOL MODE↓
30 IF OAT < 19 THEN STOP COOL
MODE PROGRAM #2↓
PROGRAM #3↓
..↓
PROGRAM #32↓
SET OUT PUTS TO LAST COMMANDED 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 example 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 supports 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	DESCRIPTION SUMMARY
NUMBERS	DATA	Real numbers are handled by the T3000 in full floating point format
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, evaluates 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 hysteresis 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 output to 0, same as STOP but is more intuitive in some cases
CONPROP	FUNCTION	Allows a Basic program to update the proportional term in a particular PID controller
CONRATE	FUNCTION	Allows a Basic program to update the derivative term in a particular PID controller
CONRESET	FUNCTION	Allows a Basic program to update 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 RETURN 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 becomes 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	Inverse 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, 0 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 Inputs 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
Air Handling Unit 1	Supply Fan Status	AHU1 SUPPLY FAN STAT
Pump # 1 Radiation Loop Start/ Stop	P1 RAD LOOP S/S	P1.S/S

#### Select the range:

To set the range highlight 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	Full Label	Auto/Manual	Value	Units	Range	Calibration	Sign	Filter	Status	Jumper	Label
IN1		Auto	0.00	Volts	0.0 to 5.0	0.0	-	0	Normal	Unused	
IN2		Auto	0.00		Unused	0.0	-	0	Normal	Unused	
IN3		Auto	0.00		Unused	0.0	-	0	Normal	Unused	
IN4		Auto	0.00		Unused	0.0	-	0	Normal	Unused	
IN5		Auto	0.00		Unused	0.0	-	0	Normal	Unused	
IN6		Auto	0.00		Unused	0.0	-	0	Normal	Unused	
IN7		Auto	0.00		Unused	0.0	-	0	Normal	Unused	
IN8		Auto	0.00		Unused	0.0	-	0	Normal	Unused	
IN9		Auto	0.00		Unused	0.0	+	0	Normal	Unused	
IN10		Auto	0.00		Unused	0.0	+	20	Normal	Unused	
IN11		Auto	0.00		Unused	0.0	+	0	Normal	Unused	
IN12		Auto	0.00		Unused	0.0	+	0	Normal	Unused	
IN13		Auto	0.00		Unused	0.0	+	0	Normal	Unused	
IN14		Auto	0.00		Unused	0.0	+	0	Normal	Unused	
IN15		Auto	0.00		Unused	0.0	+	0	Normal	Unused	
IN16		Auto	0.00		Unused	0.0	+	0	Normal	Unused	
IN17		Auto	0.00		Unused	0.0	+	0	Normal	Unused	
IN18		Auto	0.00		Unused	0.0	+	0	Normal	Unused	
IN19		Auto	0.00		Unused	0.0	+	0	Normal	Unused	
IN20		Auto	0.00		Unused	0.0	+	0	Normal	Unused	
IN21		Auto	0.00		Unused	0.0	+	0	Normal	Unused	
IN22		Auto	0.00		Unused	0.0	+	0	Normal	Unused	
IN23		Auto	0.00		Unused	0.0	+	0	Normal	Unused	
IN24		Auto	0.00		Unused	0.0	+	0	Normal	Unused	
IN25		Auto	0.00		Unused	0.0	+	0	Normal	Unused	
IN26		Auto	0.00		Unused	0.0	+	0	Normal	Unused	
IN27		Auto	0.00		Unused	0.0	+	0	Normal	Unused	
IN28		Auto	0.00		Unused	0.0	+	0	Normal	Unused	

## 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 requires 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 voltages 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 outputs only and delays the start of a device in minutes: seconds up to a maximum delay of 99 minutes 59 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	Full Label	Auto/Manual	Value	Units	Range	PWM Period	Status	Label	HOA Switch
OUT1			On		Off/On	0	OK		MAN-ON
OUT2			0.00		Unused	0	OK		MAN-ON
OUT3			0.00		Unused	0	OK		MAN-ON
OUT4			0.00		Unused	0	OK		MAN-ON
OUT5			0.00		Unused	0	OK		MAN-ON
OUT6			0.00		Unused	0	OK		MAN-ON
OUT7		Auto	0.00		Unused	0	OK		AUTO
OUT8		Manual	3.50	Volts	0.0 > 10	10	OK		AUTO
OUT9		Auto	0.00		Unused	0	-		AUTO
OUT10		Auto	0.00		Unused	0	-		AUTO
OUT11		Auto	0.00		Unused	0	-		AUTO
OUT12		Auto	0.00		Unused	0	-		AUTO
OUT13		Auto	0.00		Unused	0	-		AUTO
OUT14		Auto	0.00		Unused	0	-		AUTO
OUT15		Auto	0.00		Unused	0	-		AUTO
OUT16		Auto	0.00		Unused	0	-		AUTO
OUT17		Auto	0.00		Unused	0	-		AUTO
OUT18		Auto	0.00		Unused	0	-		AUTO
OUT19		Auto	0.00		Unused	0	-		AUTO
OUT20		Auto	0.00		Unused	0	-		AUTO
OUT21		Auto	0.00		Unused	0	-		AUTO
OUT22		Auto	0.00		Unused	0	-		AUTO
OUT23		Auto	0.00		Unused	0	-		AUTO
OUT24		Auto	0.00		Unused	0	-		AUTO
OUT25		Auto	0.00		Unused	0	-		AUTO
OUT26		Auto	0.00		Unused	0	-		AUTO
OUT27		Auto	0.00		Unused	0	-		AUTO

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

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		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	42	Normal		

## 6. Write Individual Programs

Press the “Ins” key while highlighting the line of the desired 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 Screens:

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 “Screens” 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.

NUM	Full Label	Label	Picture File	Mode	Refresh Rate
<input checked="" type="checkbox"/> 1	Hiller	TEST	hiller.bmp	Graphic	0
<input type="checkbox"/> 2	Boiler		Boiler.bmp	Graphic	0
<input type="checkbox"/> 3	Text			Graphic	0
<input type="checkbox"/> 4				Graphic	0
<input type="checkbox"/> 5				Graphic	0
<input type="checkbox"/> 6				Graphic	0
<input type="checkbox"/> 7				Graphic	0
<input type="checkbox"/> 8				Graphic	0
<input type="checkbox"/> 9				Graphic	0
<input type="checkbox"/> 10				Graphic	0
<input type="checkbox"/> 11				Graphic	0
<input type="checkbox"/> 12				Graphic	0
<input type="checkbox"/> 13				Graphic	0
<input type="checkbox"/> 14				Graphic	0
<input type="checkbox"/> 15				Graphic	0
<input type="checkbox"/> 16				Graphic	0

## 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.



## 10 Programming Statements

# CHAPTER 10

The complete list of Control Basic components:

<Todo> work on this table.....

ag	1	afrg
	2	
	3	aerg
	4	
ag	5	agb

STATEMENT	TYPE	DESCRIPTION
NUMBERS	DATA	Real numbers are handled by the T3000 in full floating point format
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, evaluates 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 hysteresis 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 output to 0, same as STOP but is more intuitive in some cases
CONPROP	FUNCTION	Allows a Basic program to update the proportional term in a particular PID controller
CONRATE	FUNCTION	Allows a Basic program to update the derivative term in a particular PID controller
CONRESET	FUNCTION	Allows a Basic program to update 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
--------	----------	---

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.

#### 10.4.1 **REAL NUMBERS**

Any real number between  $-3.4 \times 10^{-38}$  to  $3.4 \times 10^{38}$  is recognized. The number can be entered in a number of different formats.

Examples:.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 START PUMP7

#### 10.4.3 **DATES ( JAN 1 to DEC 31 )**

Each date evaluates to a real number. Note that February is always considered to have 29 days. Thus Mar 1 is always 61. The month abbreviations are:

JAN	MAY	SEP	
FEB	JUN		OCT
MAR	JUL		NOV
APR	AUG		DEC

10 IF DOY = AUG 12 THEN STOP

Example: FAN

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

Times are evaluated to real numbers using the following formula:

time = hour\*100 + (60\*min + sec)/36.

Examples: 14:30:30 = 1450.833

#### 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 performs 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 perform 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
<=	Result is true if left expression is GREATER THAN OR EQUAL
=	Result is true if both expressions are EQUAL.
<>	Result is true if expressions are NOT EQUAL.

#### 10.4.6 **LOGIC**

---

##### **AND**

AND performs 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 performs 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).

#### **10.4.7 COM MANDS AND FUNC TIONS**

As shown 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.

#### **ABS**

Purpose: To find 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**

Purpose: To cause analog alarms.

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

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

z is an expression representing the alarm differential

string is the alarm message. Up to 69 characters may be entered.

Example: 10 ALARM SAT > SETPOINT + 2 , 2 , SAT IS TOO HIGH

### **ALARM- AT (T3000 only)**

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

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

Example: 10

20

30

40

ALARM-AT  
ALL  
REM  
ALARMS  
WILL GO  
TO ALL  
PANELS.  
ALARM-AT 2  
8  
REM  
ALARMS  
WILL GO TO  
PANELS 2  
AND 8 ONLY

### **AVG**

Purpose: To find the average of a number of values.

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

Example: 10

AVGTEMP =  
AVG( RMTEMP1 ,  
RMTEMP2 ,  
RMTEMP3 )

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

### **CALL (T3000 only)**

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 a program point (i.e. PRG1)

arg1, arg2 are arguments to be passed to the program

Note: Arguments may be any point, local variable, or expression.

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 (T3000 only)**

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

Format: CLEAR

Example: 10 CLEAR

### **CLOSE**

To set a value of a variable

Purpose: to 0 (i.e. off).

Format: CLOSE variable

Example: 10 CLOSE VENT

20 CLOSE A

30 CLOSE VAR1

### **CONPROP**

(T3000 only)

Enable programs to  
change the controller

Purpose: constants.

Format: CONPROP(con# , value ) RANGE: 0 TO 4000

Description: con# is the controller you wish to change value is the new value you want to set

CONPROP returns the actual value that is entered into the controller constants. This is the same as value unless the value is outside the range allowed.

10 A = CONPROP( 1 , 2.3

Example: \* SAT \* FACTOR )

---

**CONRATE**

(T3000 only)

Enable programs to  
change the controller  
constants.

Purpose:

Format: CONRATE(con# , value ) RANGE: 0 TO 255

Description: con# is the controller you wish to change value is the new value you want to set

CONRATE returns the actual value that is entered into the controller constants. This is the same as value unless the value is outside the range allowed.

Example: 10 A = CONRATE( 1 , 0.85 )

**CONRESET** (T3000 only)

Enable programs to change the

Purpose: controller constants.

Format: CONRATE(con# , value ) RANGE: 0 TO 2.00

Description: con# is the controller you wish to change value is the new value you want to set

CONRESET returns the actual value that is entered into the controller constants. This is the same as value unless the value is outside the range allowed.

Example: 10 IF ABS( SAT - SAS ) < 0.5 THEN A = CONRESET( 5 , 50 ) ELSE A =  
CONRESET( 5 , 0 )**DALARM**

Purpose: To cause digital alarms.

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. Up to 69 characters may be entered

Example: 10 DALARM SF-C <>SF-S , 60 ,FAN STATUS ALARM 20 DALARM IN1 = IN2 , 10 ,  
IN1 AND IN2 ARE EQUAL**DECLARE** (T3000 only)

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 to the original arguments in the CALL statement.
5. Disable the “Timer” field of a program using DECLARE so that it will not run un-less called by CALL.
6. CALL and DECLARE are useful when a similar control routine has to be executed repeatedly.

## **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**

Purpose: To find the current day of the month.

Format: DOM

Example:                   IF+ DOM = 1 THEN  
              10                   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 (Jan 1 = 1).

Format: DOY

Example: 10 IF DOY = JUL 1 THEN GOSUB 100 20 END

```
30 REM HOLIDAY PROCEDURE
```

```
40 RETURN
```

## **ENABLE**

Purpose: To set a value of a variable to 1(i.e. ON).

Format: ENABLE variable

Example: 10 ENABLE OUT1 20 ENABLE A

```
30 ENABLE VAR1
```

## **END**

Purpose: To end execution of the program.

Format: END

Example: 100 IF NOT BOILER-S THEN END 110 START SF : START RF

```
120 END
```

## **FOR (T3000 only)**

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 encountered. 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 reversed. 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 110
30 END
40 REM THIS LINE GETS
EXECUTED AFTER LINE 10
50 RETURN
```

## **GOTO**

To redirect the program

Purpose: execution.

Format: GOTO line

Description: line is the line number

Example: 10 GOTO 100

20 REM THIS LINE DOES NOT GET EXECUTED

30 REM THIS LINE GETS EXECUTED AFTER LINE 10

## **HANGUP**

Purpose: To end phone call.

Format: HANGUP

Example: 10 PHONE 555-1234

20 PRINT "MESSAGE"

30 WAIT 0:02 : WAIT FOR PRINT TO OCCUR

40 HANGUP

## IF

Purpose: Decision making.

Format: IF x THEN clause [ELSE clause]

Description: x is any expression

clause is a BA SIC 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 (if it exists).

Example: 10

```
IF  
OUT1  
THEN  
START  
OUT2  
IF  
AVG( A ,  
B , C )  
> 22.4  
THEN  
30  
ELSE  
20      50  
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.

IF+ x THEN clause [ELSE

Format: 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.

```
IF+
CHILLER
THEN
GOSUB
40
        IF+ TIME >
        12:00 THEN
20      GOSUB 60
30 END
        PRINT "THE
        CHILLER
        HAS JUST
40      COME ON"
50
RETURN
        REM RING
        LUNCH
60      BELL
70      START BELL
        WAIT
80      00:00:05
90      STOP BELL
100
RETURN
```

### IF-

To detect occurrence

Purpose: of event.

IF- x THEN clause

Format: [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

To perform an operation at a regular

Purpose: interval. The function returns 1 every expression time, otherwise it will return 0.

Format: INTERVAL( expression )

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

Example: 50

REM VAR1  
WILL  
INCREMENT  
BY 2 EVERY  
HOUR

20

VAR1 =  
VAR1 + 2 \*  
INTERVAL( 100 )

30

40 END

REM THIS  
ROUTINE IS  
EXECUTED  
EVERY 10  
MINUTES

50

60 RETURN

## LET

Assigns the value of an expression to a

Purpose: 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 )

**LN**  
(T3000  
only)

To find the  
natural  
logarithm  
of an

Purpose: 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 )

**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** (T3000 only)

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** (T3000 only)

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 example, 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: 10                    VAR1 = AVG ( IN1 ,  
                                  IN2 , IN3 , IN4 ) - 20  
                                  ON VAR1  
                                  GOSUB 100 ,  
                                  200 , 300 , 400 ,  
20                            500  
                                  30 END  
                                  REM THE  
                                  AVERAGE IS 21 :  
100                          RETURN  
                                  REM THE  
                                  AVERAGE IS 22 :  
200                          RETURN  
                                  REM THE  
                                  AVERAGE IS 23 :  
300                          RETURN  
                                  REM THE  
                                  AVERAGE IS 24 :  
400                          RETURN  
                                  REM THE  
                                  AVERAGE IS 25 :  
500                          RETURN

**ON-**  
**ALARM**

To redirect  
program

Purpose: execution if

there was  
an alarm.

ON-ALARM

Format: line

line is the

Description: line number

10  
DALARM  
FIRE = 1 ,  
10 , THERE

Example: IS A FIRE

20 ON-ALARM 100  
30 REM  
CONTINUES HERE  
IF NO ALARM  
40 END

REM SPECIAL  
FIRE  
100 PROCEDURES  
110 STOP BOILER1 :  
STOP BOILER2

This branch  
happens  
only once  
when the  
alarm is  
generated.

Note:

### **ON- ERROR**

To redirect  
program  
execution if  
there was

Purpose: an error.

ON-  
ERROR

Format: line

line is the

Description: line number

10 PHONE

Example: 555-1234

20 ON-ERROR 100  
30 REM  
CONTINUES HERE  
IF NO ERROR

40 REM  
PROCEDURE  
IF CALL NOT  
SUCCESSFUL

ON-  
ERROR  
can be  
used  
after the  
following

Note: statements:

PHONE REMOTE      SET RUN-  
GET REMOTE      SYSTEM.

## **OPEN**

Purpose: To set the value of a variable to 1 (i.e. ON).

Format: OPEN variable

Examples: 10 OPEN VENT 20 OPEN A

30 OPEN VAR1

## **POWER-LOSS**

Purpose: To 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

## **PHONE** (T3000 only)

Purpose: To make connection with a remote computer using a modem.

Format: PHONE 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

**PRINT** (T3000 only)

Purpose: To print strings at the 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 IN TER 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 (T3000 only)**

Purpose: To select which panel(s) will receive the print message(s).  
PRINT-AT panel [panel panel] or

Format: PRINT-AT ALL

Examples: 10 PRINT-AT ALL

	PRINT
	"THIS IS
	PRINTED
	AT ALL
	PANELS
	IN
20	SYSTEM."
	PRINT-AT
30	2
	PRINT
	"THIS IS
	PRINTED
	AT PANEL
40	2 ONLY."
	PRINT-AT
50	1 3
	PRINT
	"THIS IS
	PRINTED
	AT
	PANELS 1
60	AND 3."
	PRINT-AT
70	1 5 8

PRINT  
"THIS IS  
PRINTED  
AT  
PANELS  
1, 5 AND  
8."  
80

Commas are not needed to separate

Note: panel numbers

## REM

Purpose: To make remarks in program.

Format: REM string

Description: string is a string of characters.

Example: 10 REM \*\*\* HEATING AND COOLING SETPOINTS \*\*\*

Note: REM commands don't affect the program. They provide information only.

## REMOTE- GET (T3000 only)

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.

REMOTE-SET remote point =

Format: x

x is the variable that the  
Description 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 555-1234  
30 A = 1  
40 REMOTE-SET 2-OUT4  
= A  
50 HANGUP

### **RETURN**

To return from a subroutine.

Used in conjunction with

Purpose: GOSUB.

Format: RETURN

Example: 10 GOSUB  
150  
20 REM CONTINUES  
HERE AFTER  
SUBROUTINE AT LINE  
150  
30 END  
40 REM SPECIAL  
CALCULATIONS  
50 FACTOR = A \* AVG( B ,  
C , D ) + 12  
60 RETURN

### **RUN-MACRO**

(T3000  
only)

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: To find the number of scans per second the panel is performing.

Format: SCANS

Example: 10 A = SCANS

### **SET-PRINTER**

Purpose: To let the panel know there is a serial printer connected to one of its ports.

Format: SET-PRINTER A

SET-PRINTER B

SET-PRINTER 0

Example: 10 SET-PRINTER B : REM PRINTER CONNECTED TO 1200 BAUD MO-DEM PORT

20 SET-PRINTER A : REM PRINTER CONNECTED TO 9600 BAUD LO-CAL PORT

FAULT

30 SET-PRINTER 0 : REM NO PRINTER CONNECTED - THIS IS DE

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 )

START

Purpose: To set a value of a variable to 1(i.e. on).

Format: START variable

Examples: 10 START OUT1 20 START A

30 START VAR1

**STATUS** (T3000 only)

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.

Purpose: 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

**STOP**

To set a value of a variable to

Purpose: 0(i.e. OFF).

Format: STOP variable

Examples: 10 STOP OUT1  
20 STOP A  
30 STOP VAR1  
40 STOP PRG10

**TBL**

To look up a

Purpose: value in a table.

Format: TBL( expression ,  
table# )

Example: 10 FLOW = TBL( VAR1 , 1 )

## TIME

To find the current time. It returns the time as a number where 1 hour =

Purpose: 100  
(i.e.:  
12:30=1250)

Format: TIME

Example: 10 REM CAUSE KEYBOARD MACRO 1 TO RUN AT 6 AM  
20 IF+ TIME =6:00 THEN RUN-MACRO 1

## TIME-ON

Purpose: To find the time that a variable has been true.

Format: TIME-ON( variable )

Example: 10 REM START OUT2 10 MINUTES AFTER OUT1 IS ON 20 IF TIME-ON( OUT1 ) >0:10 THEN START OUT2

## TIME-OFF

Purpose: To find the time that a variable has been false.

Format: TIME-OFF( variable )

Example: 10 REM DONT START OUT1 IF NOT OFF FOR 20 MINS 20 IF TIME-OFF( OUT1 ) >0:20 THEN GOTO 40

30 START OUT1

40 END

## WAIT (T3000 only)

Purpose: To pause program execution.

Format: WAIT expression

Examples: 10 START OUT2

20 WAIT 0:10

30

40

REM WAIT 10  
MINUTES AT  
LINE 20  
WAIT  
00:00:10

50	REM WAIT 10 SECONDS AT LINE 30
60	WAIT 150
70	REM WAIT 1 HOUR AND 30 MINS AT LINE 40
80	WAIT A + B * AVG ( C , D )
90	REM WAIT WHATEVER EXPRESSION EVALUATES TO

This statement is very useful for controlling sequential events.

Note:  
grams

in the panel will not be affected. Only the program where the statement appears in will be suspended.

## **WR-ON**

Purpose: To find when a weekly routine is set to come on.

Format: WR-ON( expression1 , expression2 )

Description: expression1 = routine number (1-16) expression2 = which time on (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 )

## **UNACK (T3000 only)**

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-A (T3000 only)**

Purpose: To find the user logged on at the local port (Port A). The function returns the user number, or 0 if no one logged on.

Format: USER-A

Example: 10 REM SEND A PERSONAL MESSAGE TO USER NO. 5

20 IF+ USER-A = 5 THEN GOSUB 40

30 END

40 PRINT "MESSAGE"

50 RETURN

### **USER-B (T3000 only)**

To find  
the user  
logged  
on at the  
remote  
port  
(Port B  
through  
modem).

Purpose: The  
function  
returns  
the user  
number,  
or 0 if  
no one  
logged  
on.

Format: USER-B

Example: 10       REM NOTIFY LOCAL USER IF REMOTE USER COMES ON

20       IF USER-B THEN START OUT1

## **10.4.8 VARIABLES**

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 thru 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)

### 10.4.9 EXPRESSIONS

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

## 10.5 ERROR MESSAGES

When the command SEND is given, the program is first compiled. If errors occur the program is not sent, and the errors (syntax and/or structural) are displayed.

### 10.5.1 SYNTAX ERRORS

Syntax errors usually are caused by misspelling of words or unfamiliarity with statement format.

The errors are as follows:

- |                                     |  |
|-------------------------------------|--|
| 1. Expecting variable               | 7. Illegal extra characters on line                |
| 2. Expecting =                      | 8. Expecting another expression                    |
| 3. Expecting operator               | 9. Illegal line number in goto                     |
|                                     | 10. String<br>is<br>missing<br>end<br>quote<br>(") |
| 4. Bracket error                    | 11. Attempt<br>to<br>change<br>the<br>value of     |
| 5. Not variable on left of equals   |  |
| 6. Function argument not expression | a point in another panel                           |

### 10.5.2

## STRUCTURAL ERRORS

- 
- |  |                       |   |
|--|-----------------------|---|
| Program too large —                    | 7.                    | NEXT<br>without<br>FOR in<br>line ###     |
| 1. text buffer                         |                       | Too many<br>FOR's —<br>allowed<br>10      |
| Too many GOTOS —                       | 8.                    | Expecting<br>line<br>number<br>after line |
| 2. allowed 50                          |                       |   |
| Program too large at                   | 9.                    |   |
| 3. line ###                            |                       |   |
| ###                                    |                       |   |
| Too many lines —                       | 10. Line after ### is |   |
| 4. allowed 100                         | too long              |   |
| 5. FOR without NEXT                    |                       |   |
| 6. GOTO or GOSUB to undefined line ### |                       |   |

## 11 T3000 Release History

\*\*\*\*\*  
ver 2015.4.10

- 1>> Fix Monitor inputs can't show normally.
  - 2>> Change the point type of ScreenEdit window.
  - 3>> Screen window now can use left double click to full screen or cancel it.
  - 4>> Program full label disappear when write then last item's full label.
  - 5>> Fixed the bug that T3000 can't run ,due to the Chinese Access .
  - 6>> Fixed the bug that The cursor will circle,when clicking the devices ,many times.
- \*\*\*\*\*

ver 2015.4.7

- 1>> Add the "10k Type2" and "10k Type3" for Tstat6 ,Tstat7 ,Tstat5!
  - 2>> Add the button of factory reset in the T3 module .
- \*\*\*\*\*

ver 2015.4.3

- 1>> Add the prompt in T3 ,when clicking the factory reset.
- \*\*\*\*\*

ver 2015.4.2

- 1>> Fixed the label in screen window can't show normally in the first few seconds.
  - 2>> When add or edit label in screen window ,it will write data into minipanel right now.Before we write into device when exit the window.
  - 3>> Fixed the monitor input list crash when user change focus to other application.
  - 4>> Error handle with the label point . Before "VAR-3" "3--VAR" will make the t3000 crash.
- \*\*\*\*\*

ver 2015.3.31

- 1>> Add the help document of Minipanel ,if the user don't know how to do ,just pressing "F1"
  - 2>> Add the CS3000 to T3000 .
  - 3>> Fixed the bug of isptool that isptool can't return ,when the device don't response
  - 4>> Add the feature that can export the registers list of all devices from T3000
- \*\*\*\*\*

ver 2015.3.26

- 1. Modify the building configuration user interface , user can add remote ip or domain.
  - 2. Add feature scan remote ip address.
  - 3. Add feature monitor analog and digital signal.
  - 4. Program code now it can bigger than 400 bytes.
  - 5. Add Dyndns feature , user can fill in user name and password to management.
  - 6. Add time zone feature , for sync time.
- \*\*\*\*\*

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 changed,....
- \*\*\*\*\*

ver 2014.12.24

- 1.>support many buildings
- \*\*\*\*\*

ver 2014.12.24

1.>support many buildings

\*\*\*\*\*

ver 2014.11.28

1. Add feature Graphic.Add point ,edit point ,delete point. Add Icon .
2. Add feature User Login.bmp.
3. Optimize the scan feature.

\*\*\*\*\*

ver 2014.11.11

1>ISPTool start from T3000,it don't need to fill the parameters ,t3000 will fill them automatically.

2>T3000 will zigbee information in the skin of Tstat6

\*\*\*\*\*

ver 2014.10.17

- 1>Fix the bug,when updating database ,t3000 will crash
- 2>updated the AQ,HUM-R
- 3>HumChamber Updated ,using testo

\*\*\*\*\*

ver 2014.10.11

- 1>Fix the bug,when updating database ,t3000 will crash
- 2>updated the AQ,HUM-R
- 3>Pressure Sensor can be supported in t3000 and modbus poll.
- 4>A new device can be added to T3000's DB by Temco Modbus Poll.

\*\*\*\*\*

Ver 2014.9.26,date Released:2014-9-26

\*\*\*\*\*

1>Fixed many buildings in T3000,if user select other buildings or subnets ,t3000 can't show the information of the building or the subnet

2>Fixed some bugs ,T3000 crash.

Ver 2014.9.10,date Released:2014-9-10

\*\*\*\*\*

1>Updated the CustomProductTable compatible with all the units that support the Modbus protocol

2>Added the functionality of 'Read/Write Once'

3>Updated the Sequency

4>The model no must show in the scan result.

6>Tstat-runar can be supported.

7>Fixed the part of graphic ,which can support tstat6,7,5i

8>Fixed the automatical scan in background,if user change the node of the network,the net device can't show correctly.

\*\*\*\*\*

Ver 2014.9.4,date Released:2014-9-4

\*\*\*\*\*

1>Added the CustomProductTable for customer's device in modbus poll.

2>Updated the Isp firmware updating tool to better detect model nmuber in hex or bin  
and help users to select the proper firmware for a given device

3>Updated Hum Series sensors.

4>Added one column for the register address in modbus poll.That's to say,there are three  
column "Register Name","Rgister Address","Register Value".

5>Updated the monitor of the register value in isptool.users can monitor the registers' value and write into  
the text or Excel.

Ver 2014.6.19,date Released:2014-6-19

\*\*\*\*\*

1>Fix the ipport of tstat (when the tstat is connected to the minipanel,tstat can't show the same ipport with  
minipanel)

\*\*\*\*\*

Ver 2014.6.18,date Released:2014-6-18

\*\*\*\*\*

1>when the net device is in the different subnet ,T3000 can change the ip of device

2>A/M can be fit in the all inputs

3>when t3000 can find the db engine ,t3000 will go to the temco's website to download.

\*\*\*\*\*

Ver 2014.6.17,date Released:2014-6-17

\*\*\*\*\*

1.Control Basic Editor: When finished 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 bacnet "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, TBD : Auto flash and support pause and resume.

8. Multi-flash update tool can now correctly handle 64k and 128k thermostats.

9. Add feature that user can access the remote bacnet device by internet at 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>Fill the register name from db,automatically

2>when temco's device is connected ,temco's mbpoll can connect to the device and to show the register list in the temco's modbus poll

3>When Scanning ,Tx and Rx can show the data

4>delete "Deep Scan",now Click scan ,t3000 scan all the temco's devices

\*\*\*\*\*

Ver 2014.6.10,date Released:2014-6-10

\*\*\*\*\*

1 Modbus Poll can connect temco's device automatically,if user have been connected the device.

2 Updated the config file about Tstat6

3 Fixed some bugs about VAV functions

\*\*\*\*\*

Ver 2014.6.6,date Released:2014-6-6

\*\*\*\*\*

1 Fixed VAV function(previous verion can't fit tstat6)

2 Changed "Close" to "Closed"

3 All ranges have added "Open/close" "close/open" "on/off" "off/on"(value is the same ,show is different)

4 Fixed some features about user experience in the temco modbus poll

\*\*\*\*\*

Ver 2014.2.5,date Released:2014-4-30

\*\*\*\*\*

1 Fixed the cooling db ,sp will show ,when cooling stages are more than 0. the heating db,sp will be the same.

\*\*\*\*\*

Ver 2014.2.4,date Released:2014-4-26

\*\*\*\*\*

1 Fixed Output table on Tstat 5A, analog outputs

- 2 Fixed in system programming routine in T3000 and isptool, handles hex files with lines more than 256 chars properly now.
- 3 Fixed If NC or other Devices connect by com port and ethernet ,T3000 will find two the same device ,and then let users select which one the user want to add to the database of T3000
- 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>Supported "Tstat 5i"
- 4>Added "T3000 Help System"(from now ,users can press "F1",and then get help)

Ver 2014.2.2,date Released:2014-3-21

- \*\*\*\*\*  
1>Fixed In Main Page,The day Slider and the night Slide ,they are easy to use.
- 2>Fixed Connected and Disconnected ,the two states show different icons
- 3>Updated CO2 Sensors-Ethernet

Ver 2014.2.1,date Released:2014-3-18

- \*\*\*\*\*  
1>Fixed In Advanced page,Cool and Heat Deadband's Value,unnormal
- 2>Fixed In Inputset Dialog ,Internal Sensor's Calibration can not update ,immediately

Ver 2014.2.0,date Released:2014-3-10

- \*\*\*\*\*  
1>Fixed the bug that when input some parameters in the advanced page ,program crash.
- 2>Fixed MPC5G can't use Airflow setting
- 3>Fixed the issue of T3's Range

Ver 2014.1.4,date Released:

- \*\*\*\*\*  
1>Combined LED Table and 'Display Name' into one button called "Dsiplay".
- 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>Added T3-PT10 to the user interface ,allows configuarig of calibration ,filter and other parameters of the T3-Pt10

Ver 2014.1.3,date Released:2014-2-17

- \*\*\*\*\*  
1>Bacnet-CM5 Updated
- 2>T3-6CT Finished
- 3>Add the scan method for MiniPannel
- 4>Add T3-PT10
- 5>delete useless file "Mbpoll" "RegisterMonitor"

Ver 2014.1.2,date Released:2014-1-23

- \*\*\*\*\*  
1>Add "Check Update"  
Note:the menu [Help-Check Update]  
the function can help users find the new version T3000
- 2>Add "Add net device manually"
- 3>fix "Tstat6's Range"
- 4>Add "Parts&&Vendors ECO 6.0" to T3000[Item Master,Parts List,Assy Tree]
- 5>The driver of T3000's DB is supported in the 64bits and 32bits
- 6>Change "Output table"[off]

Ver 2014.1.1,date Released:2014-1-14

\*\*\*\*\*

- 1>Fix the error that the co2 can't show in the first page
- 2>Add the selection of CO2,Humidity in the display name config

Ver 2014.1.0,date Released:2014-1-9

\*\*\*\*\*

- 1>Support LC to config in T3000
- 2>Add "Register Viewer" Tool

Ver 2013.27,date released:2013-12-18

\*\*\*\*\*

- 1>to add "lighting sensor"
- 2>to multi-language "English ,Simple Chinese ,Your language"
- 3>to suit for win7 64 bits
- 4>to support "chang ID fro T3"

\*\*\*\*\*

Ver 2013.26,date released:2013-12-2

\*\*\*\*\*

- 1>to finish "T3-8I13O"
- 2>to fix "Change ID to 255" in the ISPTool
- 3>LC weeely annual rountines

\*\*\*\*\*

Ver 2013.25,date released:2013-11-28

\*\*\*\*\*

- 1>to add "Occupied/Unoccupied" into Input Range
- 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>to update "VAV" Function
- 2>to add the "Transducer" to outputset
- 3>to add "VAV" registers to config file

\*\*\*\*\*

Ver 2013.22,date released:2013-10-15

\*\*\*\*\*

- 1>to add "VAV" Function

\*\*\*\*\*

Ver 2013.21,date released:2013-9-25

\*\*\*\*\*

- 1>to fix and update Tstat5G
- 2>to update MiniPanel

3>to fix MBPoll

4>to rebuild 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 sepont 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 "Icd 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 attribute 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 Controller 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 button and Delete button

<8>Fix the out table

<9>Change "NC Config"->"Apply" button.

<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 successfully,"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 error: 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

\*\*\*\*\*

## 12 A DEMO SHOW

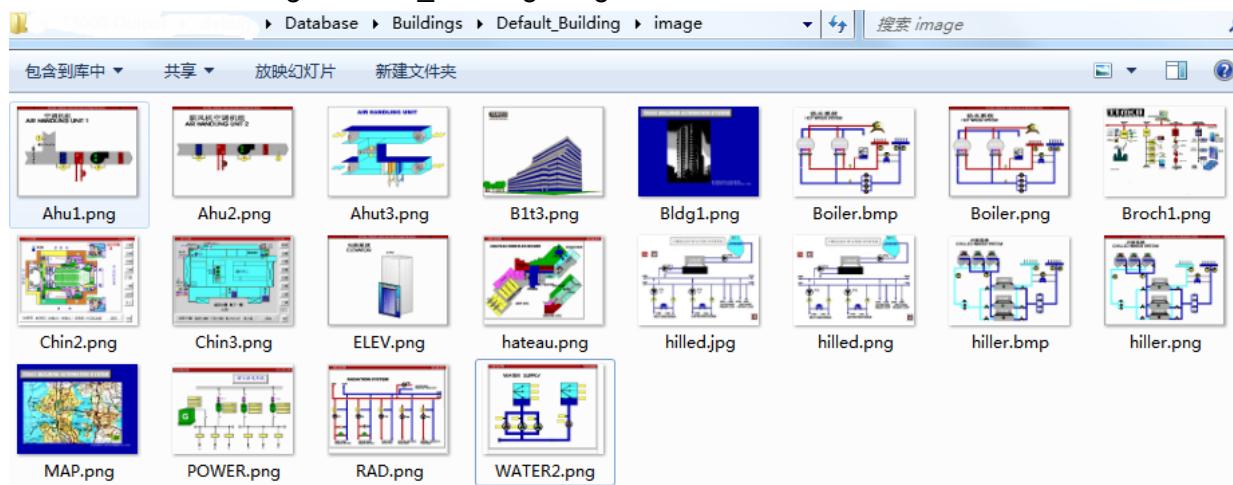
1. Download the configuration file called [T3000DEMOSHOW.zip](#)

<http://www.temcocontrols.com/ftp/software/T3000DEMOSHOW.zip>

2. Read the readme file .

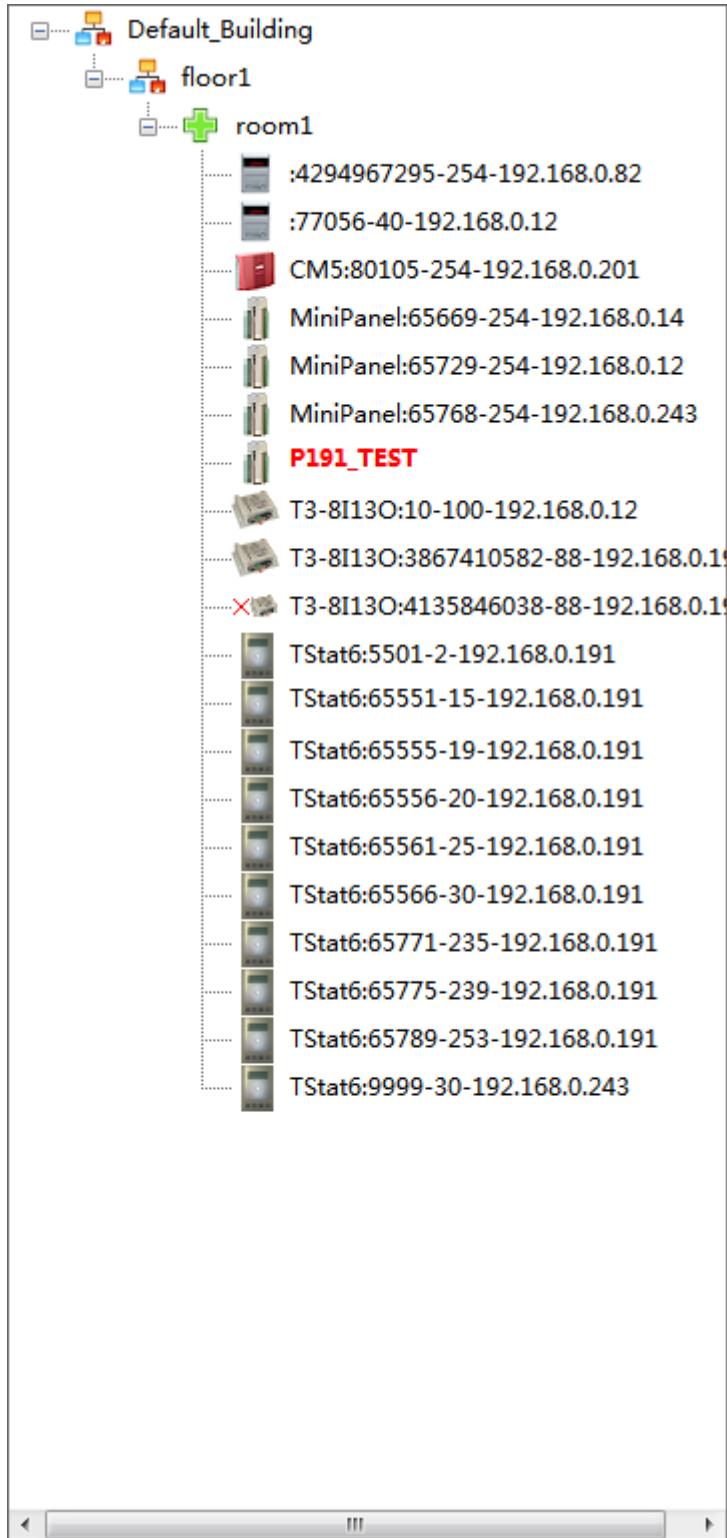
1>you should put the image fold to the installation path to replace the image file fold.

..\\Database\\Buildings\\Default\_Building\\image

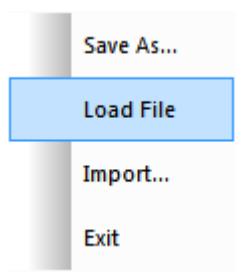


2>you should open T3000 to load the T3000DEMOSHOW.prg to your minipanel.

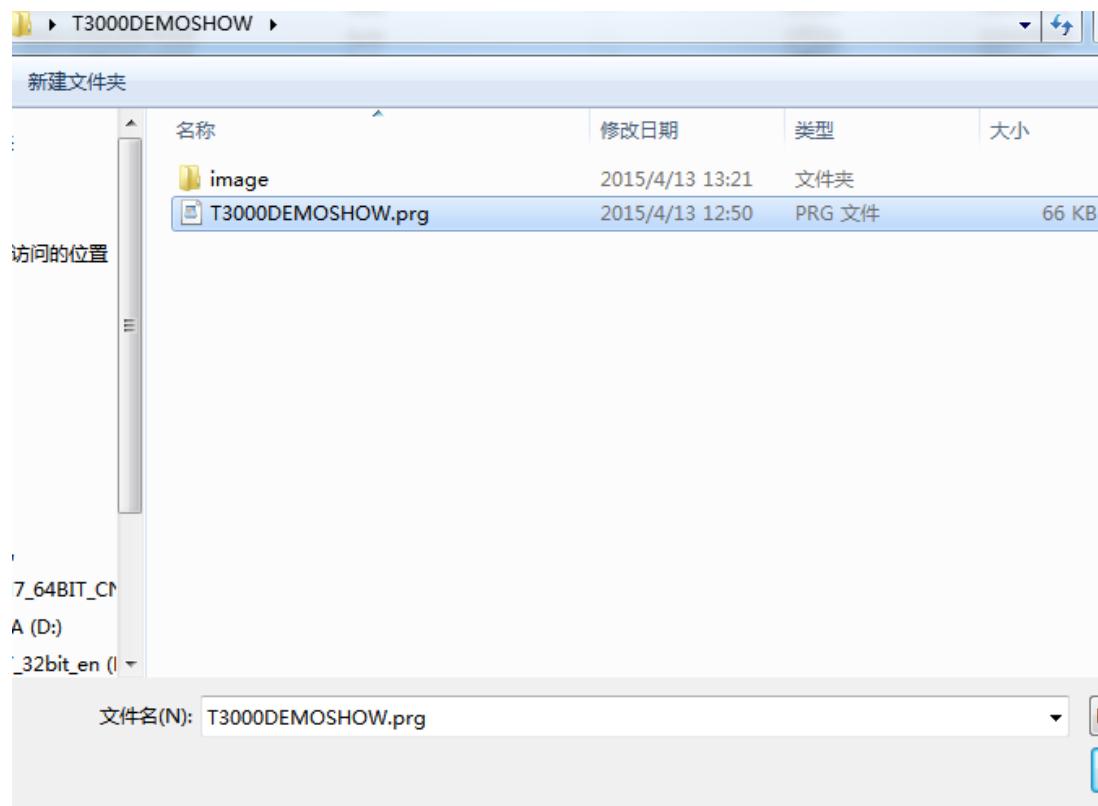
2.1.select your minipanel



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	Full Label	Auto/Manual	Value	Units	Range	Calibr...	Sign	Filter	Status	Label
IN1	ELEVATOR FLOOR3	Manual	On		On/Off		-	64	Open	FLOOR3
IN2	AHU1 FAIL MODE	Auto	Normal		Normal/Alarm		-	64	Normal	AHU1FAIL
IN3	CHILLER FAIL	Auto	Normal		Normal/Alarm		-	64	Normal	CHFAIL
IN4	AHU1 OCCUPIED SWITCH	Manual	On		Off/On		-	64	Open	AHU10CC
IN5	WATER LEVEL	Auto	On		On/Off		-	64	Normal	WATERHI
IN6	ELEVATOR FLOOR1	Auto	Off		Off/On		-	64	Normal	FLOOR1
IN7	AHU2 SUPPLY TEMP1	Auto	120.00	Deg.C	10K-40 to 120	0.0	-	64	Short	AHU2SAT
IN8	ELEVATOR FLOOR4	Auto	Off		Off/On		-	64	Normal	FLOOR4
IN9	RM301 OCCUPIED	Auto	No		No/Yes		-	64	Normal	RM301OCC
IN10	OUTSIDE AIR TEMP	Auto	120.00	Deg.C	10K-40 to 120	0.0	-	64	Short	OAT
IN11	OUTSIDE LIGHT SENSOR	Auto	0.00	Volts	0.0 to 5.0	0.0	-	64	Short	OL
IN12	RM401 KEYSWITCH	Auto	On		On/Off		-	64	Normal	RM401KEY
IN13	ROOM 401 OCCUPIED	Manual	On		Off/On		-	64	Open	RM401OCC
IN14	EF1 SMOKE DAMPER	Auto	Off		Off/On		-	64	Normal	EF1SMOKE
IN15	AHU2 SMOKE DAMPER	Auto	Off		Off/On		-	64	Normal	AHU2SMOK
IN16	AHU1 SMOKE DAMPER	Auto	Off		Off/On		-	64	Normal	AHU1SMOK
IN17	CONDENSOR SUPP TEMP	Auto	120.00	Deg.C	10K-40 to 120	0.0	+	0	Short	CWST
IN18	ELEVATOR FLOOR2	Auto	Off		Off/On		-	0	Normal	ELEV2
IN19	ELEVATOR FLOOR3	Auto	Off		Off/On		-	0	Normal	ELEV3
IN20	ELEVATOR FLOOR4	Auto	Off		Off/On		-	0	Normal	ELEV4
IN21	AHU1 FIRE ALARM	Auto	Off		Off/On		-	0	Normal	AHU1FIRE
IN22	AHU2 FIRE ALARM	Auto	Off		Off/On		-	0	Normal	AHU2FIRE
IN23	DAYTIME LIGHT1 STAT	Auto	Off		Off/On		-	0	Normal	LIGHT1
IN24	DAYTIME LIGHT2 STAT	Auto	Off		Off/On		-	0	Normal	LIGHT2
IN25	NITE LIGHT STATUS	Auto	Off		Off/On		-	0	Normal	LIGHT3
IN26	EMERGENCY LITE 1	Auto	Off		Off/On		-	0	Normal	ELIGHT1
IN27	BOILER1 STATUS	Auto	Off		Off/On		-	0	Normal	B1S
IN28	CHILLER STATUS	Auto	Off		Off/On		-	0	Normal	CH1S
IN29	COLING TOWER STAT	Auto	Off		Off/On		-	0	Normal	CT1S
IN30	EF1 STATUS	Auto	Off		Off/On		-	0	Normal	EF1S
IN31	EXIT LIGHTING STAT	Auto	Off		Off/On		-	0	Normal	LIGHT4
IN32	EMERGENCY LITE 2	Auto	Off		Off/On		-	0	Normal	ELIGHT2
IN33	CONDENSOR RETRM TEMP	Auto	120.00	Deg.C	10K-40 to 120	0.0	+	0	Open	CWRT
IN34	SMOKE DETECTOR 22 FL	Auto	Alarm		Normal/Alarm		-	0	Open	SMOKE22
IN35	SMOKE DETECTOR 21 FL	Auto	Alarm		Normal/Alarm		-	0	Open	SMOKE21
IN36	SMOKE DETECTOR 20 FL	Auto	Alarm		Normal/Alarm		-	0	Open	SMOKE20
IN37	SMOKE DETECTOR 19 FL	Auto	Alarm		Normal/Alarm		-	0	Open	SMOKE19
IN38	SMOKE DETECTOR 2	Auto	Alarm		Normal/Alarm		-	0	Open	SMOKE2
IN39	SMOKE DETECTOR 3	Auto	Alarm		Normal/Alarm		-	0	Open	SMOKE3
IN40	SMOKE DETECTOR 4	Auto	Alarm		Normal/Alarm		-	0	Open	SMOKE4
IN41	SMOKE DETECTOR 5	Auto	Normal		Normal/Alarm		-	0	Open	SMOKE5
IN42	SMOKE DETECTOR 6	Auto	Normal		Normal/Alarm		-	0	Open	SMOKE6
IN43	SMOKE DETECTOR 7	Auto	Normal		Normal/Alarm		-	0	Open	SMOKE7
IN44	SMOKE DETECTOR 8	Auto	Normal		Normal/Alarm		-	0	Open	SMOKE8
IN45	SMOKE DETECTOR 9	Auto	Normal		Normal/Alarm		-	0	Open	SMOKE9
IN46	TRANSFORMER2 VOLTS	Auto	5559.20	%(4-20ma)	0 to 100	0.0	+	0	Open	T2VOLTS
IN47	TRANSFORMER2 KVA	Auto	5559.20	%(4-20ma)	0 to 100	0.0	+	0	Open	T2KVA
IN48	BUILDING SUPPLY TEMP	Auto	120.00	Deg.C	10K-40 to 120	0.0	+	0	Open	BLDSUPT
IN49		Auto	0.00		Unused	0.0	-	0	Open	
IN50		Auto	0.00		Unused	0.0	-	0	Open	
IN51		Auto	0.00		Unused	0.0	-	0	Open	
IN52		Auto	0.00		Unused	0.0	-	0	Open	
IN53		Auto	0.00		Unused	0.0	-	0	Open	
IN54		Auto	0.00		Unused	0.0	-	0	Open	
IN55		Auto	0.00		Unused	0.0	-	0	Open	
IN56		Auto	0.00		Unused	0.0	-	0	Open	
IN57		Auto	0.00		Unused	0.0	-	0	Open	
IN58		Auto	0.00		Unused	0.0	-	0	Open	
IN59		Auto	0.00		Unused	0.0	-	0	Open	
IN60		Auto	0.00		Unused	0.0	-	0	Open	
IN61		Auto	0.00		Unused	0.0	-	0	Open	
IN62		Auto	0.00		Unused	0.0	-	0	Open	
IN63		Auto	0.00		Unused	0.0	-	0	Open	
IN64		Auto	0.00		Unused	0.0	-	0	Open	

>>>OUTPUTS

Output	Full Label	Auto/Manual	Value	Units	Range	PWM Period	Status	Label	HOA Switch
OUT1	SPARE		0.00	%Open	0.0 -> 100	0	OK	SPARE	MAN-OFF
OUT2	BUILDING CIRC PMP7		On	Off/On	0	OK	P7	MAN-OFF	
OUT3	BOILER		Off	Off/On	0	OK	BOILER	MAN-OFF	
OUT4	BOOLING TOWER		Off	Off/On	0	OK	CT1	MAN-OFF	
OUT5	BUILDING CIRC PUMP8		Off	Off/On	0	OK	P8	MAN-OFF	
OUT6	SUMP PUMP		Off	Off/On	0	OK	P4	MAN-OFF	
OUT7	RM401 FANCOIL		Off	Off/On	0	OK	RM401FAN	MAN-OFF	
OUT8	EXHAUST FAN		Off	Off/On	0	OK	EF1	MAN-OFF	
OUT9	RM301 FANCOIL		On	On/Off	0	OK	RM301FAN	MAN-OFF	
OUT10	AHU1 SUPPLY FAN		Off	Off/On	0	OK	AHU1_FAN	MAN-OFF	
OUT11	AHU2 SUPPLY FAN		Off	Off/On	0	OK	AHU2_FAN	MAN-OFF	
OUT12	WATER SUPPLY PUMP		Off	Off/On	0	OK	P3	MAN-OFF	
OUT13	ELEVATOR UP		Off	Off/On	0	OK	ELEV_DN	MAN-OFF	
OUT14	SPARE		Off	Off/On	0	OK	SPARE	MAN-OFF	
OUT15	SPARE		Off	Off/On	0	OK	SPARE	MAN-OFF	
OUT16	RM401 LIGHTS		Off	Off/On	0	OK	LITE7	MAN-OFF	
OUT17	DAYLIGHT3 RM201		off	Off/On	0	OK	RM201	MAN-OFF	
OUT18	NORMAL LIGHT		Off	Off/On	0	OK	LITE1	MAN-OFF	
OUT19	DAYTIME LIGHTS2		off	Off/On	0	OK	LITE2	MAN-OFF	
OUT20	DAYTIM1		off	Off/On	0	OK	LITE3	MAN-OFF	
OUT21	NIGHTTIME LIGHTS		off	Off/On	0	OK	LITE4	MAN-OFF	
OUT22	DAYLIGHT1		off	Off/On	0	OK	LITE5	MAN-OFF	
OUT23	COOLING TOWER PUMP		Off	Off/On	0	OK	P2	MAN-OFF	
OUT24	ELEVATOR UP		Off	Off/On	0	OK	ELEV_UP	MAN-OFF	
OUT25	CHILLER ENABLE	Manual	Off	Off/On	0	-	CH1	AUTO	
OUT26	P16	Auto	Off	Off/On	0	-	P16	AUTO	
OUT27		Auto	0.00	Unused	0	-		AUTO	
OUT28	CENT BYPASS VALUE	Auto	0.00	%	0.0 -> 100	0	-	CBYPASS	AUTO
OUT29	CHILLER 2	Auto	Off	Off/On	0	-	CH2	AUTO	
OUT30	CHILLER 3	Auto	Off	Off/On	0	-	CH3	AUTO	
OUT31		Auto	Off	Off/On	0	-		AUTO	
OUT32	COOLING TOWER FAN2	Auto	Off	Off/On	0	-	CT2	AUTO	
OUT33	COOLING TOWER FAN3	Auto	Off	Off/On	0	-	CT3	AUTO	
OUT34	WATER SUPPLY PUMP3	Auto	Off	Off/On	0	-	P3	AUTO	
OUT35	WATER SUPPLY PUMP4	Auto	off	Off/On	0	-	P4	AUTO	
OUT36	WATER SUPPLY PUMP5	Auto	Off	Off/On	0	-	P5	AUTO	
OUT37	WATER SUPPLY PUMP6	Auto	Off	Off/On	0	-	P6	AUTO	
OUT38	AHU1 DAMPER	Auto	0.00	%Open	0.0 -> 100	0	-	AHU1MAD	AUTO
OUT39	AHU1 VALVE	Auto	0.00	%Open	0.0 -> 100	0	-	AHU2VLV	AUTO
OUT40	LIGHTING ZONE2	Auto	Off	Off/On	0	-	LIGHT2	AUTO	
OUT41	LIGHTING ZONE3	Auto	Off	Off/On	0	-	LIGHT3	AUTO	
OUT42	LIGHTING ZONE4	Manual	Off	Off/On	0	-	LIGHT4	AUTO	
OUT43	BLDING CIRC PUMP P9	Auto	Off	Off/On	0	-	P9	AUTO	
OUT44	BLDING CIRC PUMP P10	Auto	Off	Off/On	0	-	P10	AUTO	
OUT45	STEAM VALVE 1	Auto	0.00	%	0.0 -> 100	0	-	V1	AUTO
OUT46	STEAM VALVE 2	Auto	0.00	%	0.0 -> 100	0	-	V2	AUTO
OUT47	AHU2 VALVE	Auto	0.00	%	0.0 -> 100	0	-	AHU2_VLV	AUTO
OUT48	ROOM 301 VALVE	Auto	0.00	%	0.0 -> 100	0	-	401VLV	AUTO
OUT49	ROOM 401 VALVE	Auto	0.00	%	0.0 -> 100	0	-	301VLV	AUTO
OUT50		Auto		Unused	0	-		AUTO	
OUT51		Auto		Unused	0	-		AUTO	
OUT52		Auto		Unused	0	-		AUTO	
OUT53		Auto		Unused	0	-		AUTO	
OUT54		Auto		Unused	0	-		AUTO	
OUT55		Auto		Unused	0	-		AUTO	
OUT56		Auto		Unused	0	-		AUTO	
OUT57		Auto		Unused	0	-		AUTO	
OUT58		Auto		Unused	0	-		AUTO	
OUT59		Auto		Unused	0	-		AUTO	
OUT60		Auto		Unused	0	-		AUTO	
OUT61		Auto		Unused	0	-		AUTO	
OUT62		Auto		Unused	0	-		AUTO	
OUT63		Auto		Unused	0	-		AUTO	
OUT64		Auto		Unused	0	-		AUTO	

## >>>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 OVERRIDE MODE	Auto	Off	Off/On	AHU1OM
VAR6	AHU1 DAYTIME SETPT	Manual	0.000	Deg.C	AHU1DSP
VAR7	AHU1 FREECOOL AVAIL	Auto	Off	Off/On	AHU1COOL
VAR8	AHU1 SETPOINT ADJUST	Manual	0.000	Deg.C	AHUSPA
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 HEATING MODE	Auto	Off	Off/On	HEATMOD2
VAR13	AHU2 COOLING MODE	Auto	Off	Off/On	COOLMOD2
VAR14	AHU2 OVERRIDE 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 @ ODEGC	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_2
VAR50	TOTAL HEATLOSS2	Auto	0.000	Unused	TLOSS2
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	
3	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 PROGRAM	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	pid test	ON	Auto	84	Normal	SWS	

Red arrow points from the checked row in the table to the 'Insert' button on the toolbar.

Red arrow points from the 'Insert' button to the Bacnet Program IDE window.

Red arrow points from the 'Insert' button to the 'Delete' button on the toolbar.

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

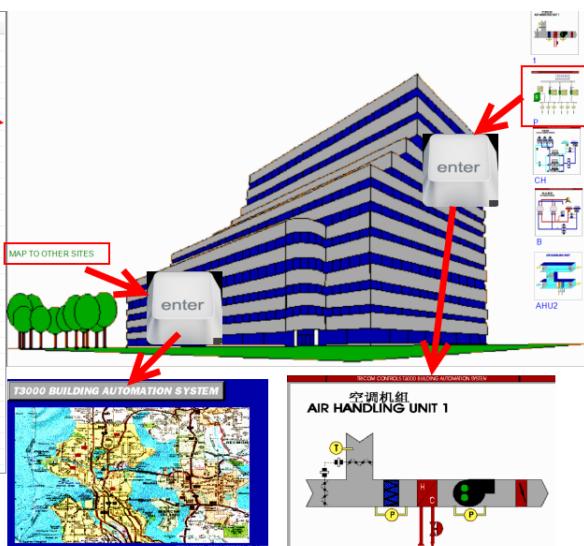
```

### >>>PID Loops

NUM	Input	Value	Units	A/M	Output	Setpoint	Set Value	Units	Action	Proportional	Reset	Rate	Bias	Sample Time
1	BLDSUPT	120.0	Deg.C	Auto	1297043.3%	1-VAR1	0.0	Deg.C	-	22	2	2	0	0
2	BLDSUPT	120.0	Deg.C	Auto	0.0%	1-VAR60	0.0	Deg.C	-	0	0	0	0	0
3	AHU2SAT	120.0	Deg.C	Auto	0.0%	1-VAR10	2.0	Deg.C	-	0	0	0	0	0
4	AHU2SAT	120.0	Deg.C	Auto	0.0%	1-VAR11	0.0	Deg.C	-	0	0	0	0	0
5				Auto	0.0%				-	0	0	0	0	0
6				Auto	0.0%				-	0	0	0	0	0
7				Auto	0.0%				-	0	0	0	0	0
8				Manual	0.0%				-	0	0	0	0	0
9				Auto	0.0%				-	0	0	0	0	0
10				Auto	0.0%				-	0	0	0	0	0
11				Auto	0.0%				-	0	0	0	0	0
12				Auto	0.0%				-	0	0	0	0	0
13				Auto	0.0%				-	0	0	0	0	0
14				Auto	0.0%				-	0	0	0	0	0
15				Auto	0.0%				-	0	0	0	0	0
16				Auto	0.0%				-	0	0	0	0	0

### >>>Graphics

NUM	Full Label	Label	Picture File	Mode	Refresh Rate
1	MAIN MENU	M	B1t3.png	Graphic	15
2	AHU2 DLS	AHU2	Ahu3.png	Graphic	15
3	BOILER		Boiler.png	Graphic	15
4	WATER MONITORING	W	water.png	Graphic	15
5	AIR HANDLING UNIT1	1	Ahu1.png	Graphic	15
6	CHATEAU WHISTLER	CHAT	hateau.png	Graphic	15
7	TAIWAN NATNL THEATRE	T	Chn2.png	Graphic	15
8	POWER MONITORING	P	POWER.png	Graphic	15
9	AIR HANDLING UNIT2	2	Ahu2.png	Graphic	15
10	FIRE ALARM SYSTEM	F	Chn2.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	Chn3.png	Graphic	15
15	RADIATION HEATING	RAD	RAD.png	Graphic	15
16	CHILLED WATER SYSTEM	CH	hiller.png	Graphic	15



## 13 How to update firmware and load configuration file one by one

Using MiniPanel as an example.

1> Download the latest firmware from [temcocontrols.com](http://www.temcocontrols.com) ,or paste "http://www.temcocontrols.com/ftp/firmware/MINIPANEL/" to your browser .

Select the version that you want to update to download .



## Index of /ftp/firmware/MINIPANEL

- [Parent Directory](#)
- [MINI\\_revision.TXT](#)
- [minipanel\\_rev36.9.bin](#)
- [minipanel\\_rev37.9.bin](#)
- [minipanel\\_rev38.2.bin](#)
- [minipanel\\_rev38.4.bin](#)
- [minipanel\\_rev38.5.bin](#)

If you get the latest firmware from temco's engineer, don't need the step 1.

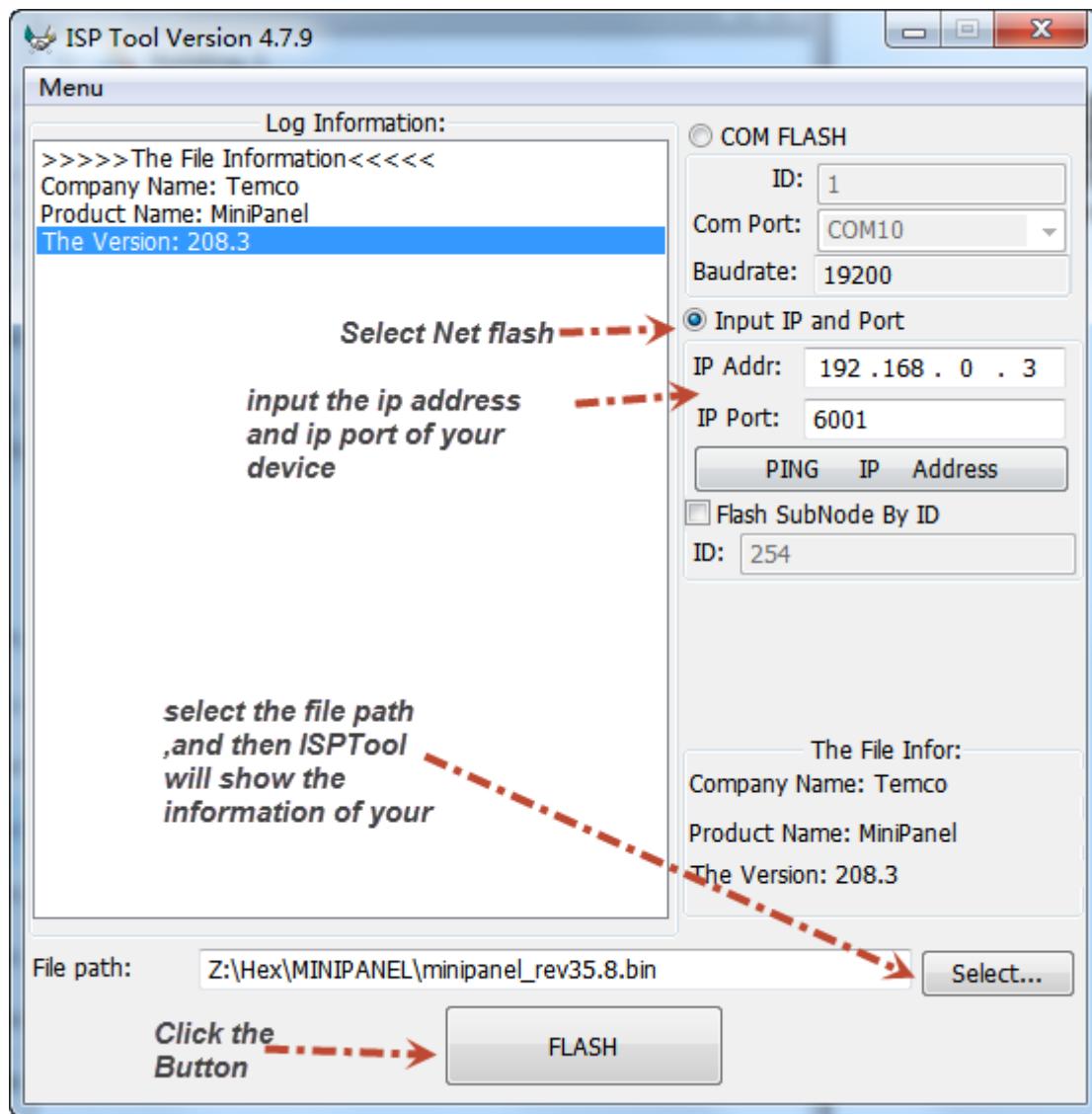
2> If your computer didn't install the T3000 ,please download [T3000Software.zip](#) .

If you have installed the T3000,don't need the step2.

3> Click the icon of ISPTool



and then show 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 .

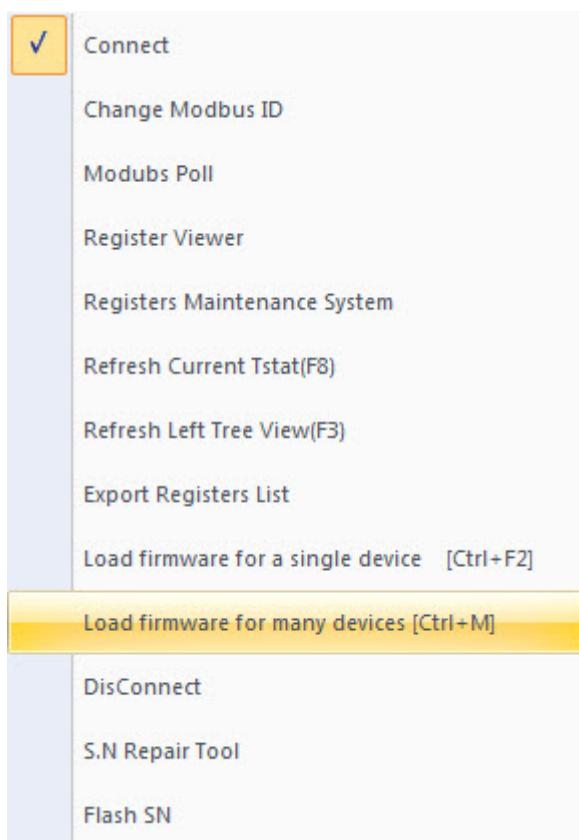
ISPTool will show the right message in the blank area ,including the ip and ip port.

## How to batch to update firmware and load configuration file

1. You must find your devices by T3000 , using "Scan"



2 "Load firmware for many devices"



Load firmware for many devices															
Hex or Bin File :			1	2	Apply To All Select	3	The config file :			4	5	... Apply To All Select	6		
NUM	SN	Panel ID	Name	Com	IP	Port	Is Sub	Is Online	Software Version	Firmware Version	Hex Bin File	11	Configuration file	12	Firmware
<input checked="" type="checkbox"/>	3 65555	19	TStat6	10			NO	Offline	0.0						
<input checked="" type="checkbox"/>	4 65776	240	TStat6	10			NO	Offline	0.0						
<input checked="" type="checkbox"/>	5 81474	252	TStat6	10			NO	Offline	0.0						
<input checked="" type="checkbox"/>	6 65540	253	TStat6	10			NO	Offline	0.0						
<input checked="" type="checkbox"/>	7 80105	254	CMS	192.168.0.201	10000		NO	Online	37.0	2.2	Z:\Hex\CMS\bootldr_cm5_rev22_HWREV11.bin				
<input checked="" type="checkbox"/>	8 80491	254	TStat6	10			NO	Offline	0.0						
<input checked="" type="checkbox"/>	9 65669	254	MinPanel	192.168.0.14	10000		NO	Online	37.1	12.6	Z:\Hex\MINIPANEL\minipanel_rev36.9.bin				
<input checked="" type="checkbox"/>	10 65768	254	MinPanel	192.168.0.243	10000		NO	Online	37.1	12.6	Z:\Hex\MINIPANEL\minipanel_rev36.9.bin				
<input checked="" type="checkbox"/>	11 -1	254	MinPanel	192.168.0.212	10000		NO	Online	37.1						
<input checked="" type="checkbox"/>	12 77992	254	CMS	192.168.0.48	10000		NO	Offline	0.0						
<input checked="" type="checkbox"/>	13 1000	254	MinPanel	192.168.0.113	10000		NO	Offline	0.0						
<input checked="" type="checkbox"/>	14 11345	254	MinPanel	192.168.0.251	10000		NO	Online	37.1	12.6	Z:\Hex\MINIPANEL\minipanel_rev36.9.bin				
<input checked="" type="checkbox"/>	15 73625	3	TStatSA	10			NO	Online	37.8						
<input checked="" type="checkbox"/>	16 65574	42	TStat6	10			NO	Offline	0.0						
<input checked="" type="checkbox"/>	17 51898	7	TStatSG	10			NO	Offline	0.0						

- 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 "Com  
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 Firmware  
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 .



Description : This is a register value of the Modbus ID 48 connected to panel 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 panel 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:

There are some descriptions in the chapter 6.3 and 6.4 about the IO of MP . you can use outer inputs and outer output ,just like the inner inputs and inner outputs.

That's to say ,they are the same.

IN2:it is inner inputs

IN33:it is outer inputs

You can use them in programs or screens in the same way ,so you can see ,whatever inner or outer ,for users ,they are the same .

There have a table to descript the IO of MP

TYPE OF MINIPANEL	INNER INPUTS	INNER DIGITE OUTPUTS	INNER ANALOG OUTPUTS	OUT
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 :

**1>INPUTS**

NUM	Full Label	Auto/Manual	Value	Units	Range	Calibration	Sign	Filter	Status	Label	
1	ELEVATOR FLOOR	Manual	On		On/Off		64	Open	FLOOR3		
2	AHU1 FAIL MODE	Auto	Normal		Normal/Alarm		64	Normal	AHU1FAIL		
3	CHILLER FAIL	Auto	Alarm		Normal/Alarm		64	Normal	CHFAIL		
4	AHU1 OCCUPIED S	Manual	On		Off/On		64	Open	AHU10CC		
5	WATER LEVEL	Auto	On		On/Off		64	Normal	WATERHI		
6	ELEVATOR FLOOR	Auto	On		Off/On		64	Normal	FLOOR1		
7	AHU2 SUPPLY TEM	Auto	70.00	Deg.C	10K -40 to 120	0.0	-	64	Normal	AHU2SAT	
8	ELEVATOR FLOOR	Auto	On		Off/On		64	Normal	FLOOR4		
9	RM301 OCCUPIED	Auto	No		No/Yes		64	Normal	RM3010CC		
10	OUTSIDE AIR TEM	Auto	20.60	Deg.C	10K -40 to 120	0.0	-	64	Open	OAT	
11	OUTSIDE LIGHT SE	Auto	1.62	Volts	0.0 to 5.0	0.0	-	64	Normal	OL	
12	RM401 KEYSWITCH	Auto	On		On/Off		64	Normal	RM401KEY		
13	ROOM 401 OCCUPI	Manual	On		Off/On		64	Open	RM401OCC		
14	EF1 SMOKE DAMP	Auto	On		Off/On		64	Normal	EF1SMOKE		
15	AHU2 SMOKE DAM	Auto	On		Off/On		64	Normal	AHU2SMOK		
16	AHU1 SMOKE DAM	Auto	On		Off/On		64	Normal	AHU1SMOK		
17	CONDENSOR SUP	Auto	-40.0	Deg.C	10K -40 to 120	0.0	+	0	Open	CWST	
18	ELEVATOR FLOOR	Auto	On		Off/On		0	Normal	ELEV2		
19	ELEVATOR FLOOR	Auto	On		Off/On		0	Normal	ELEV3		
20	ELEVATOR FLOOR	Auto	On		Off/On		0	Normal	ELEV4		
21	AHU1 FIRE ALARM	Auto	On		Off/On		0	Normal	AHU1FIRE		
22	AHU2 FIRE ALARM	Auto	On		Off/On		0	Normal	AHU2FIRE		
23	DAYTIME LIGHT1 S	Auto	On		Off/On		0	Normal	LIGHT1		
24	DAYTIME LIGHT2 S	Auto	On		Off/On		0	Normal	LIGHT2		
25	NITE LIGHT STATU	Auto	On		Off/On		0	Normal	LIGHT3		
26	EMERGENCY LITE	Auto	On		Off/On		0	Normal	ELIGHT1		
27	BOILER1 STATUS	Auto	On		Off/On		0	Normal	B1S		
28	CHILLER STATUS	Auto	On		Off/On		0	Normal	CH1S		
29	COOLING TOWER ST	Auto	On		Off/On		0	Normal	CT1S		
30	EF1 STATUS	Auto	On		Off/On		0	Normal	EF1S		
31	EXIT LIGHTING ST	Auto	On		Off/On		0	Normal	LIGHT4		
32	EMERGENCY LITE	Auto	On		Off/On		0	Normal	ELIGHT2		
33	CONDENSOR RET	Auto	120.00	Deg.C	10K -40 to 120	0.0	+	0	Open	CWRT	
34	SMOKE DETECTOF	Auto	Alarm		Normal/Alarm		0	Open	SMOKE22		
35	SMOKE DETECTOF	Auto	Alarm		Normal/Alarm		0	Open	SMOKE21		
36	SMOKE DETECTOF	Auto	Alarm		Normal/Alarm		0	Open	SMOKE20		
37	SMOKE DETECTOF	Auto	Alarm		Normal/Alarm		0	Open	SMOKE19		
38	SMOKE DETECTOF	Auto	Alarm		Normal/Alarm		0	Open	SMOKE2		
39	SMOKE DETECTOF	Auto	Alarm		Normal/Alarm		0	Open	SMOKE3		
40	SMOKE DETECTOF	Auto	Alarm		Normal/Alarm		0	Open	SMOKE4		
41	SMOKE DETECTOF	Auto	Normal		Normal/Alarm		0	Open	SMOKE5		
42	SMOKE DETECTOF	Auto	Normal		Normal/Alarm		0	Open	SMOKE6		
43	SMOKE DETECTOF	Auto	Normal		Normal/Alarm		0	Open	SMOKE7		
44	SMOKE DETECTOF	Auto	Normal		Normal/Alarm		0	Open	SMOKE8		
45	SMOKE DETECTOF	Auto	Normal		Normal/Alarm		0	Open	SMOKE9		
46	TRANSFORMER2 V	Auto	5559.20	%(4-20ma)	0 to 100	0.0	+	0	Open	T2VOLTS	
47	TRANSFORMER2 K	Auto	5559.20	%(4-20ma)	0 to 100	0.0	+	0	Open	T2KVA	
48	BUILDING SUPPLY	Auto	120.00	Deg.C	10K -40 to 120	0.0	+	0	Open	BLDSUPT	
49		Auto	0.00			0.0	-	0	Open		
50		Auto	0.00			0.0	-	0	Open		
51		Auto	0.00			0.0	-	0	Open		
52		Auto	0.00			0.0	-	0	Open		
53		Auto	0.00			0.0	-	0	Open		
54		Auto	0.00			0.0	-	0	Open	Other inputs are unuseful	
55		Auto	0.00			0.0	-	0	Open		
56		Auto	0.00			0.0	-	0	Open		
57		Auto	0.00			0.0	-	0	Open		
58		Auto	0.00			0.0	-	0	Open		
59		Auto	0.00			0.0	-	0	Open		
60		Auto	0.00			0.0	-	0	Open		
61		Auto	0.00			0.0	-	0	Open		
62		Auto	0.00			0.0	-	0	Open		
63		Auto	0.00			0.0	-	0	Open		
64		Auto	0.00			0.0	-	0	Open		

2>OUTPUTS

NUM	Full Label	Auto/Manual	Value	Units	Range	0%	100%	Status	Label	HOA Switch
1	SPARE	Auto	0.00	%Open	0.0 > 100	0	0	-	SPARE	AUTO
2	BUILDING CIRC PMP7	Auto	On		Off/On			-	P7	AUTO
3	BOILER	Auto	Off		Off/On			-	BOILER	AUTO
4	BOILING TOWER	Auto	Off		Off/On			-	CT1	AUTO
5	BUILDING CIRC PUMP8	Auto	Off		Off/On			-	P8	AUTO
6	SUMP PUMP	Auto	Off		Off/On			-	P4	AUTO
7	RM401 FANCOIL	Auto	Off		Off/On			-	RM401FAN	AUTO
8	EXHAUST FAN	Auto	Off		Off/On			-	EF1	AUTO
9	RM301 FANCOIL	Auto	10.00	Volts	0.0 > 10	0	0	-	RM301FAN	AUTO
10	AHU1 SUPPLY FAN	Auto	0.00	Volts	0.0 -> 10	0	0	-	AHU1_FAN	AUTO
11	AHU2 SUPPLY FAN	Auto	Off		Off/On			OK	AHU2_FAN	AUTO
12	WATER SUPPLY PUMP	Auto	Off		Off/On			OK	P3	AUTO
13	ELEVATOR UP	Auto	Off		Off/On			OK	ELEV_DN	AUTO
14	SPARE	Auto	Off		Off/On			OK	SPARE	AUTO
15	SPARE	Auto	Off		Off/On			OK	SPARE	AUTO
16	RM401 LIGHTS	Auto	Off		Off/On			OK	LITE7	AUTO
17	DAYLIGHT3 RM201	Auto	Off		Off/On			OK	RM201	AUTO
18	NORMAL LIGHT	Auto	Off		Off/On			OK	LITE1	AUTO
19	DAYTIME LIGHTS2	Manual	Off		Off/On			OK	LITE2	AUTO
20	DAYTIM1	Auto	Off		Off/On			OK	LITE3	AUTO
21	NIGHTTIME LIGHTS	Auto	Off		Off/On			OK	LITE4	AUTO
22	DAYLIGHT1	Auto	Off		Off/On			OK	LITE5	AUTO
23	COOLING TOWER PUMP	Auto	Off		Off/On			OK	P2	AUTO
24	ELEVATOR UP	Auto	Off		Off/On			-	ELEV_UP	AUTO
25	CHILLER ENABLE	Manual	On		Off/On			-	CH1	AUTO
26	P16	Manual	Off		Off/On			-	P16	AUTO
27		Auto	0.00			0	0	-		AUTO
28	CENT BYPASS VALUE	Auto	0.00	%	0.0 -> 100	0	0	-	CBYPASS	AUTO
29	CHILLER 2	Auto	Off		Off/On			-	CH2	AUTO
30	CHILLER 3	Auto	Off		Off/On			-	CH3	AUTO
31		Auto	Off		Off/On			-		AUTO
32	COOLING TOWER FAN2	Auto	Off		Off/On			-	CT2	AUTO
33	COOLING TOWER FAN3	Auto	Off		Off/On			-	CT3	AUTO
34	WATER SUPPLY PUMP3	Auto	Off		Off/On			-	P3	AUTO
35	WATER SUPPLY PUMP4	Auto	Off		Off/On			-	P4	AUTO
36	WATER SUPPLY PUMP5	Auto	Off		Off/On			-	P5	AUTO
37	WATER SUPPLY PUMP6	Auto	Off		Off/On			-	P6	AUTO
38	AHU1 DAMPER	Auto	0.00	%Open	0.0 -> 100	0	0	-	AHU1MAD	AUTO
39	AHU1 VALVE	Auto	0.00	%Open	0.0 -> 100	0	0	-	AHU2VLV	AUTO
40	LIGHTING ZONE2	Auto	Off		Off/On			-	LIGHT2	AUTO
41	LIGHTING ZONE3	Auto	Off		Off/On			-	LIGHT3	AUTO
42	LIGHTING ZONE4	Manual	Off		Off/On			-	LIGHT4	AUTO
43	BLDING CIRC PUMP P9	Auto	Off		Off/On			-	P9	AUTO
44	BLDING CIRC PUMP P10	Auto	Off		Off/On			-	P10	AUTO
45	STEAM VALVE 1	Auto	0.00	%	0.0 -> 100	0	0	-	V1	AUTO
46	STEAM VALVE 2	Auto	0.00	%	0.0 -> 100	0	0	-	V2	AUTO
47	AHU2 VALVE	Auto	0.00	%	0.0 -> 100	0	0	-	AHU2_VLV	AUTO
48	ROOM 301 VALVE	Auto	0.00	%	0.0 -> 100	0	0	-	401VLV	AUTO
49	ROOM 401 VALVE	Auto	0.00	%	0.0 -> 100	0	0	-	301VLV	AUTO
50		Auto						-		AUTO
51		Auto						-		AUTO
52		Auto						-		AUTO
53		Auto						-		AUTO
54		Auto						-		AUTO
55		Auto						-		AUTO
56		Auto						-		AUTO
57		Auto						-		AUTO
58		Auto						-		AUTO
59		Auto						-		AUTO
60		Auto						-		AUTO
61		Auto						-		AUTO
62		Auto						-		AUTO
63		Auto						-		AUTO
64		Auto						-		AUTO

These are 24 inner outputs

These are 13 outer outputs extended by T3-8I13O

Other outputs are unuseful

## **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 source code is here on github:

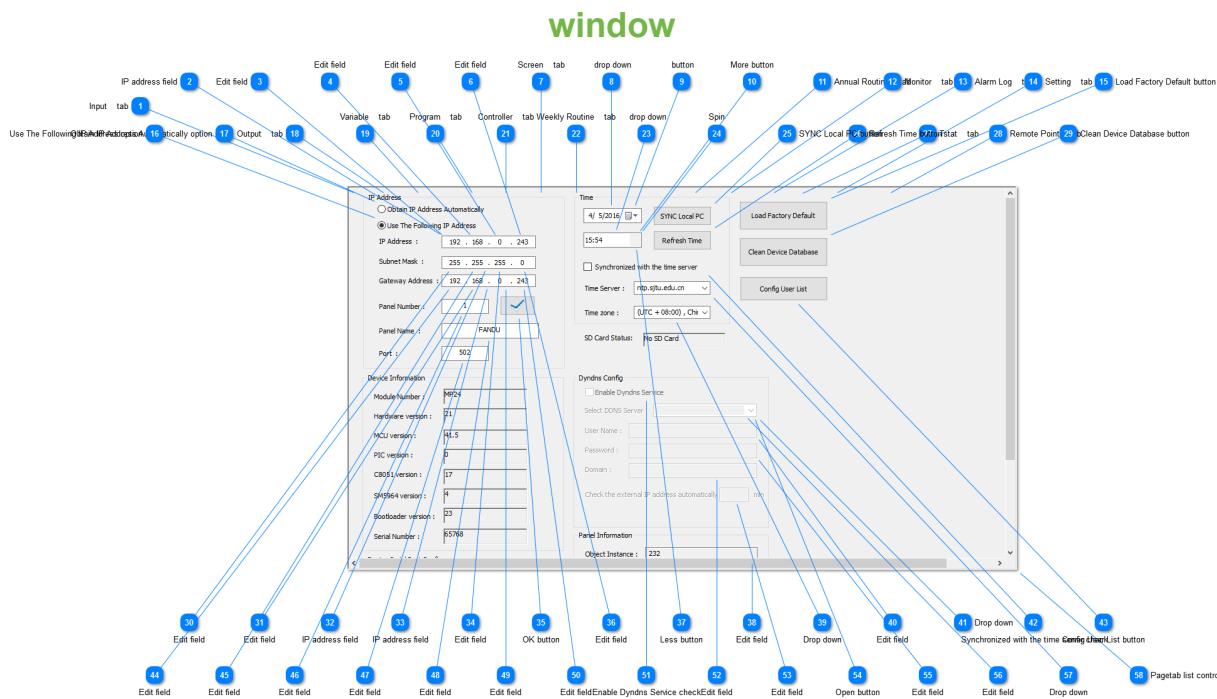
[T3000 Open Source Project](#)

The T3000 Building Automation System IS DISTRIBUTED "AS IS". NO WARRANTY OF ANY KIND IS EXPRESSED OR IMPLIED. YOU USE AT YOUR OWN RISK. THE AUTHOR 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: [sales2@temcocontrols.com](mailto:sales2@temcocontrols.com)



## 1 Input tab

## 2 IP address field

## 3 Edit field

part 1

## 4 Edit field

part 2

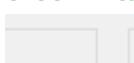
## 5 Edit field

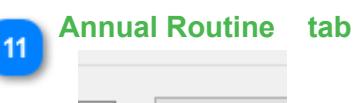
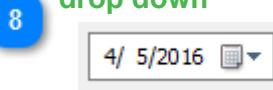
part 3

## 6 Edit field

part 4

## 7 Screen tab





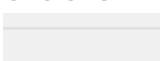
**19** Variable tab



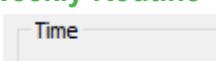
**20** Program tab



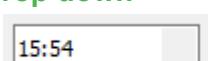
**21** Controller tab



**22** Weekly Routine tab



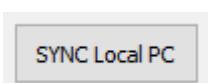
**23** drop down



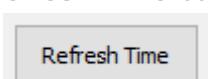
**24** Spin



**25** SYNC Local PC button



**26** Refresh Time button



**27** Tstat tab

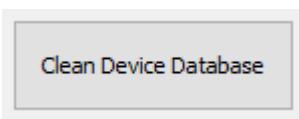


**28** Remote Point tab



**29** Clean Device Database button





30 Edit field

part 1

31 Edit field

part 2

32 IP address field

33 IP address field

34 Edit field

part 3

35 OK button



36 Edit field

part 4

37 Less button



38 Edit field

39 Drop down

**40** Edit field

**41** Drop down

**42** Synchronized with the time server check

 Synchronized with the time server

**43** Config User List button

**44** Edit field

part 1

**45** Edit field

part 2

**46** Edit field

**47** Edit field

**48** Edit field

**49** Edit field

part 3

**50** Edit field

part 4

**51** Enable Dyndns Service check

Enable Dyndns Service

**52** Edit field

**53** Edit field

**54** Open button



**55** Edit field

**56** Edit field

**57** Drop down

A dropdown menu showing the text "ntp.sjtu.edu.cn" followed by a downward-pointing arrow.

**58** Pagetab list control

<p>IP Address</p> <p><input type="radio"/> Obtain IP Address Automatically</p> <p><input checked="" type="radio"/> Use The Following IP Address</p> <p>IP Address : 192 . 168 . 0 . 243</p> <p>Subnet Mask : 255 . 255 . 255 . 0</p> <p>Gateway Address : 192 . 168 . 0 . 243</p> <p>Panel Number : 1 <input checked="" type="checkbox"/></p> <p>Panel Name : FANDU</p> <p>Port : 502</p>	<p>Time</p> <p>4/ 5/2016 <input type="button" value="▼"/> SYNC Local PC</p> <p>15:54 <input type="button" value="Refresh Time"/></p> <p><input type="checkbox"/> Synchronized with the time server</p> <p>Time Server : ntp.sjtu.edu.cn</p> <p>Time zone : (UTC + 08:00) , China</p> <p>SD Card Status: No SD Card</p>	<p>Load Factory Default</p> <p>Clean Device Database</p> <p>Config User List</p>
<p>Device Information</p> <p>Module Number : MP24</p> <p>Hardware version : 21</p> <p>MCU version : 41.5</p> <p>PIC version : 0</p> <p>C8051 version : 17</p> <p>SM5964 version : 4</p> <p>Bootloader version : 23</p> <p>Serial Number : 65768</p>	<p>Dyndns Config</p> <p><input type="checkbox"/> Enable Dyndns Service</p> <p>Select DDNS Server : <input type="button" value="▼"/></p> <p>User Name : <input type="text"/></p> <p>Password : <input type="text"/></p> <p>Domain : <input type="text"/></p> <p>Check the external IP address automatically <input type="text"/> min</p>	<p>Panel Information</p> <p>Object Instance : 232</p>