iStat6

Description:

This full-featured programmable thermostat is specifically designed by our engineers for full compatibility with ControlCore software products.
In conjunction with the ControlCore suite, it provides unique features, full programmability, easy Modbus integration, all at one of the best performance/price ratio available on the market.

Highlights:
- Licensed for ControlCore programming
- High impact plastic enclosure, provides durability in commercial environments
- 8 AUI, 4 AO, 4-20mA option, 5 relays
- Real Time Clock
- Photo cell (measures ambient light)

Technical data:

- 5 relays x 2amps @24V
- 8 AUI, jumper selectable (Therm, 0-10V, 4-20ma), dry contact, 12 bit resolution
- AUIs are software configurable as Analog/Digital/Counter/Frequency inputs
- 4 AO, jumper selectable: 0-10V (30mA max) or 4-20mA mode, 10 bit resolution.
- Default Modbus address: 255.

Operating temperature…............................... -30-70°C (-22~158°F)
LCD screen temperature limitation.……….... 0-35°C (32~95°F)

Supply voltage…......................………..16~24V AC/DC, 50-60Hz

Power consumption…...100mA at 12VDC

Relay contacts rating…...2A @ 24VDC
            UL File No.: E43149CSA File No.: LR26550

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

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

Plastic Housing…............…Flammability rating UL 94V0 file E194560
Enclosure rating…...IP31

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

Programmable momentary push-button

Supports ControlCore™ and standard Modbus/RTU protocol.
RS485 interface 19200 / 8-N-1
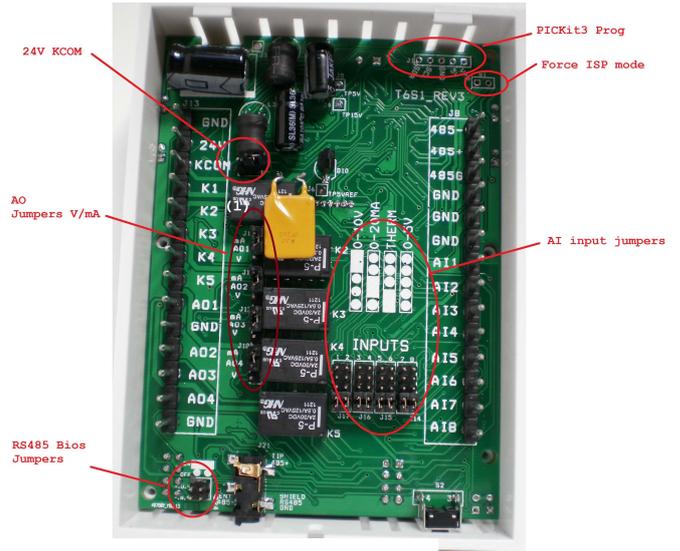User selectable baud rates (9600, 19200, 38400, 57600, 115200)

Colour…..White/Off-white
Weight ………...………. 200g
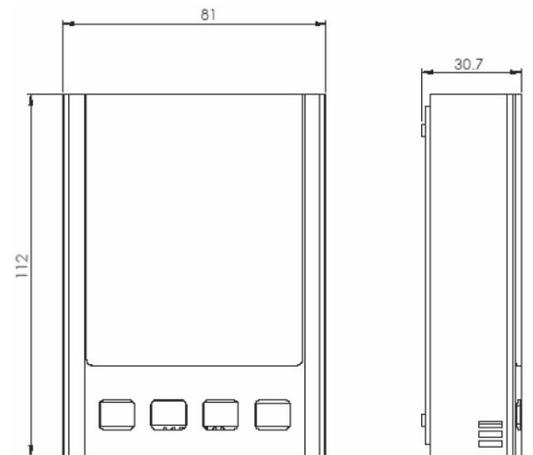
iStat6 board:
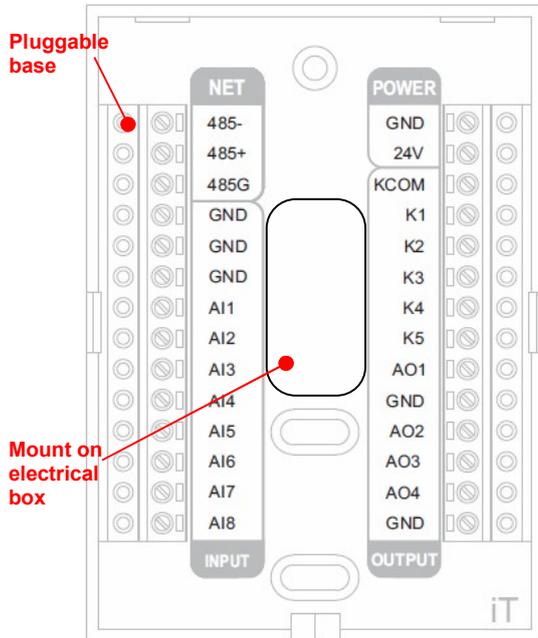
Jumper settings:

Wiring Diagram:

① Optional Network
② Relay Rating: Dry contact, 2A any one relay, 4A combined
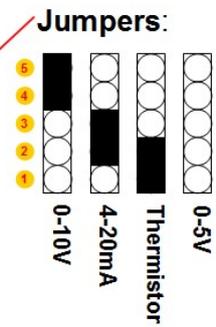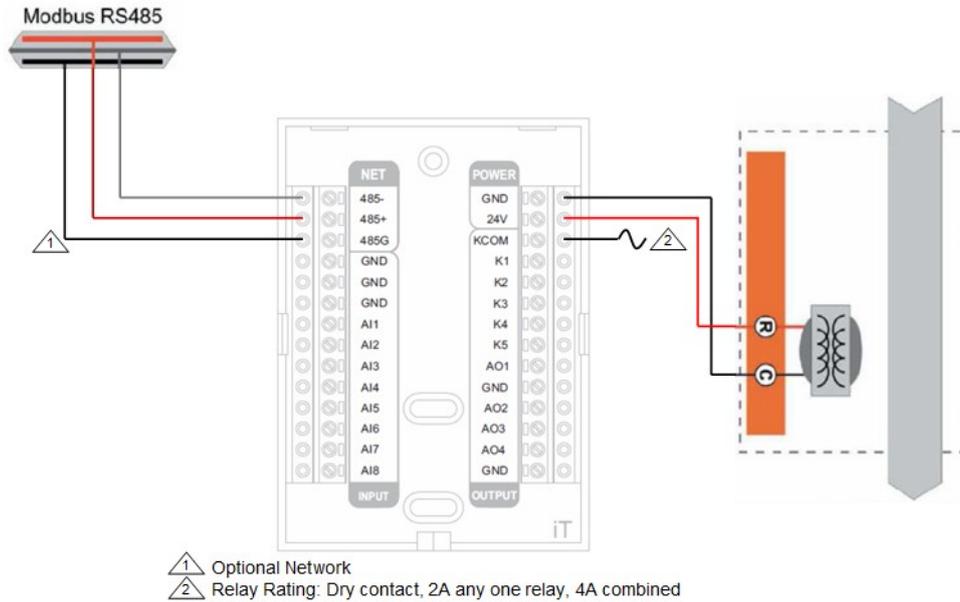
One Year Guarantee and Limitation of Liability:

iSquared software inc. guarantees to the original purchaser that this product is free from defects in both material and workmanship under normal use and service within one (1) year from the date of shipment from the factory or from its official sales channel to the original purchaser.

The liability of iSquared software inc. under this guarantee shall not in any case exceed the cost of correcting defects in the product of supplying replacement product as herein provided and upon the expiration of the guarantee period all such liability shall terminate.

Programming your device: ControlCore quick tutorial

The following tutorial will walk you through the creation of a simple ControlCore project. No actual embedded hardware is required, we will use simulation mode. Upon completion you will have created a typical control algorithm for a series of embedded devices and you will be able to actually test it out using ControlCore hardware simulation facility. The final project will look something like the display shown below.
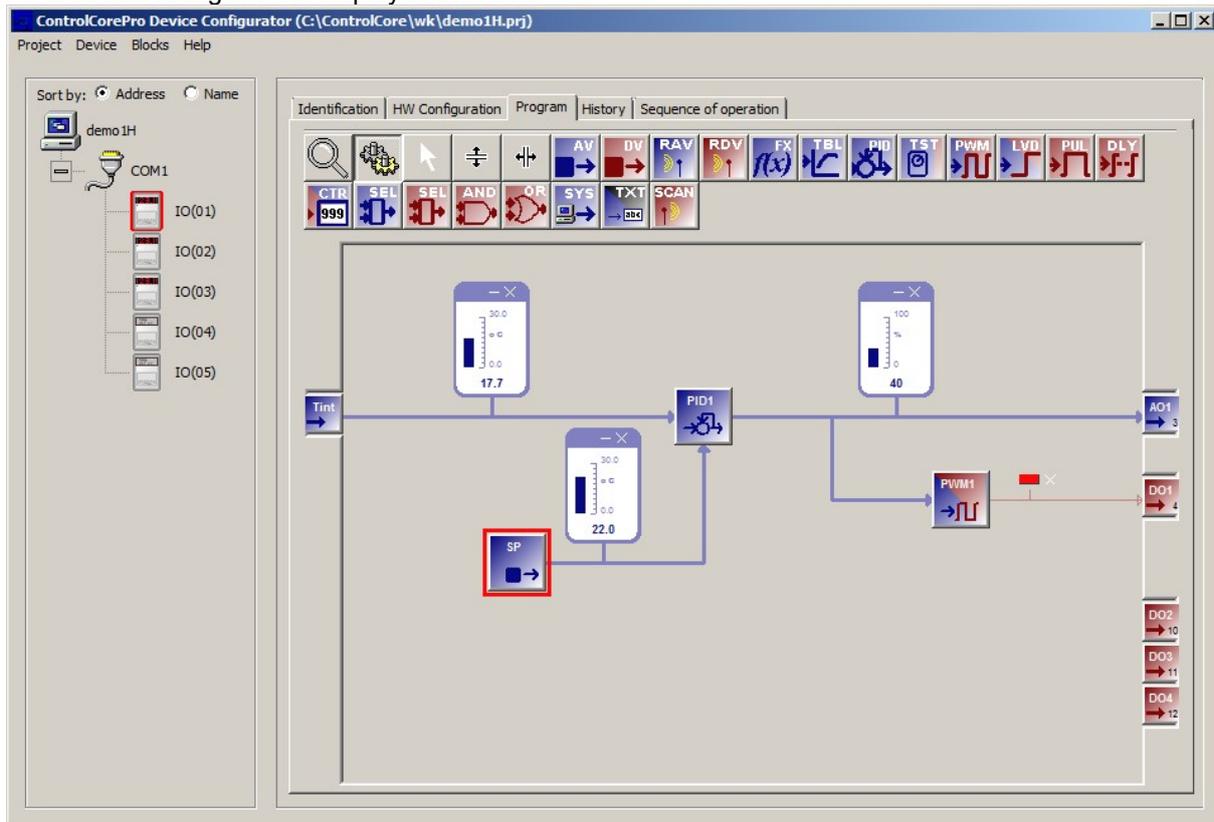

Programming your device: Starting a new ControlCore project

When you start ControlCore you will be presented with a blank workspace. The first step will be to create a new project, select **Project -> New Project…** from the main menu at the top of the window.
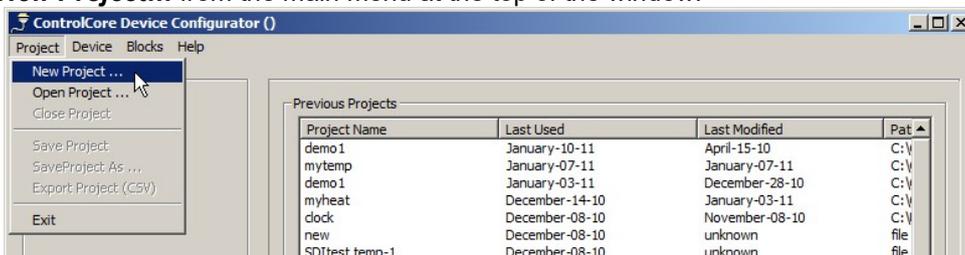

The new project's icon will appear on the left side of the window. By clicking on the icon, you will see the project's settings on the right. The default connection type is a local serial port. Other available options are "Remote Ethernet Gateway" and "CatNet". For the purpose of this tutorial, we will use the "Local Comm Port" option, assuming that your devise is connected to the local port of your computer.
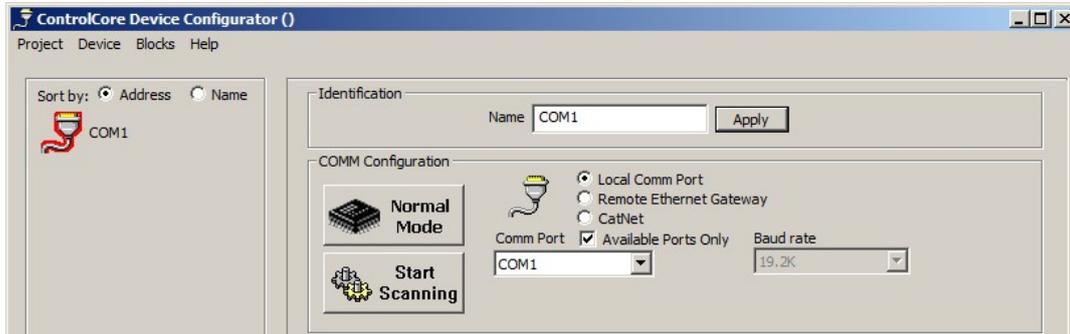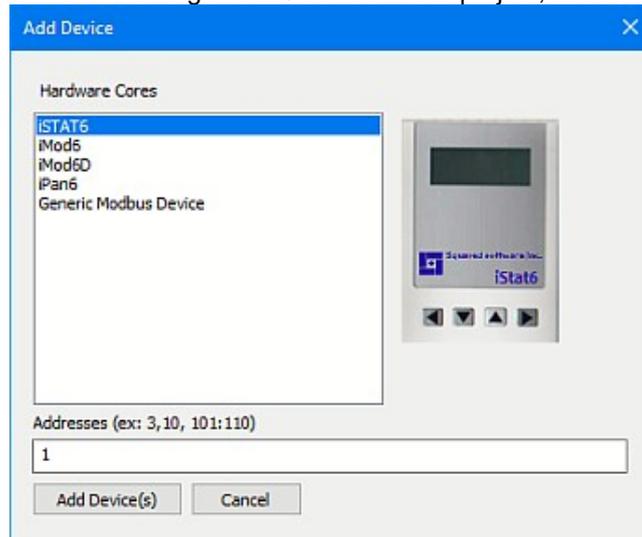
Programming your device: Add devices to your project

You can add devices to your new project by selecting *Device -> Add Device(s)* from the top menu bar.
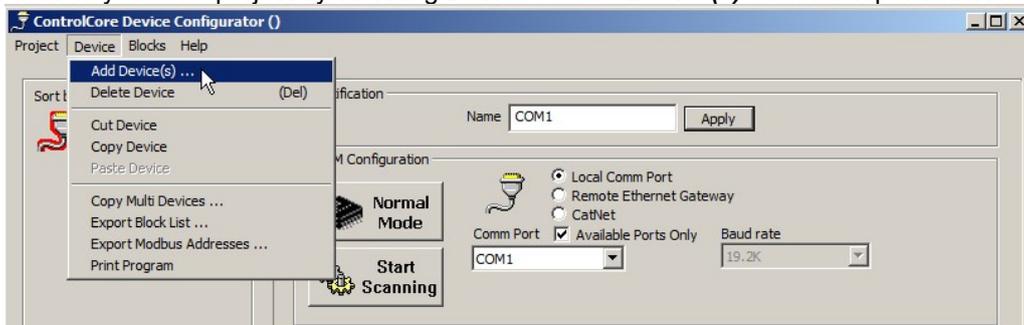

This will display the dialog box shown below. Select the type of device to add from the left selection list and then enter the device's address in the text box at the bottom. You can add several devices in one step by entering several device addresses in the text box. Note that valid addresses are 1..253. Addresses 0 and 255 are reserved and 254 is a special universal address. When a device sees the address 254 it always responds, hence it is VERY important that you never use 254 on an RS485 network with more than 1 device connected. Generally 254 is used to access a device where the actual address is not known.

For the purpose of this tutorial, we will add a single iStat6 device to our project, at address 1.
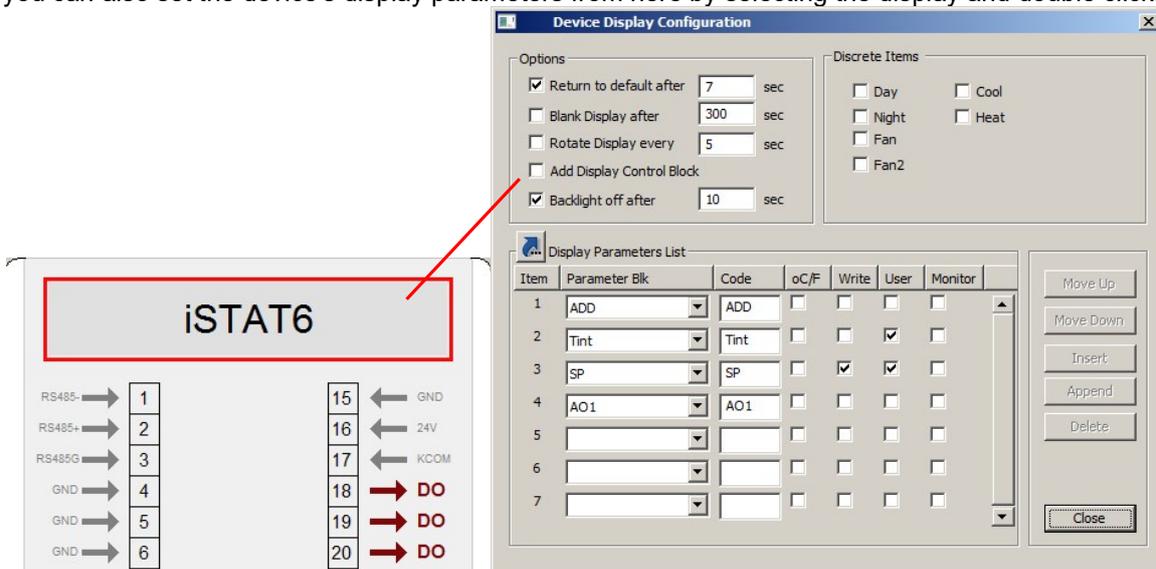
Programming your device: Device Hardware Configuration

The devices created in the previous step are now shown on the left hand side tree view. Selecting one of the devices displays its configuration tabs on the right. The HW Configuration tab allows access to the device's hardware configuration, (i.e. IO pins and display). When you hover the mouse over a configurable I/O pin, the pin's options are displayed at the bottom, clicking on the selected pin will sequentially toggle through the available options for that pin. Select each pin and setup the pin configuration shown below.
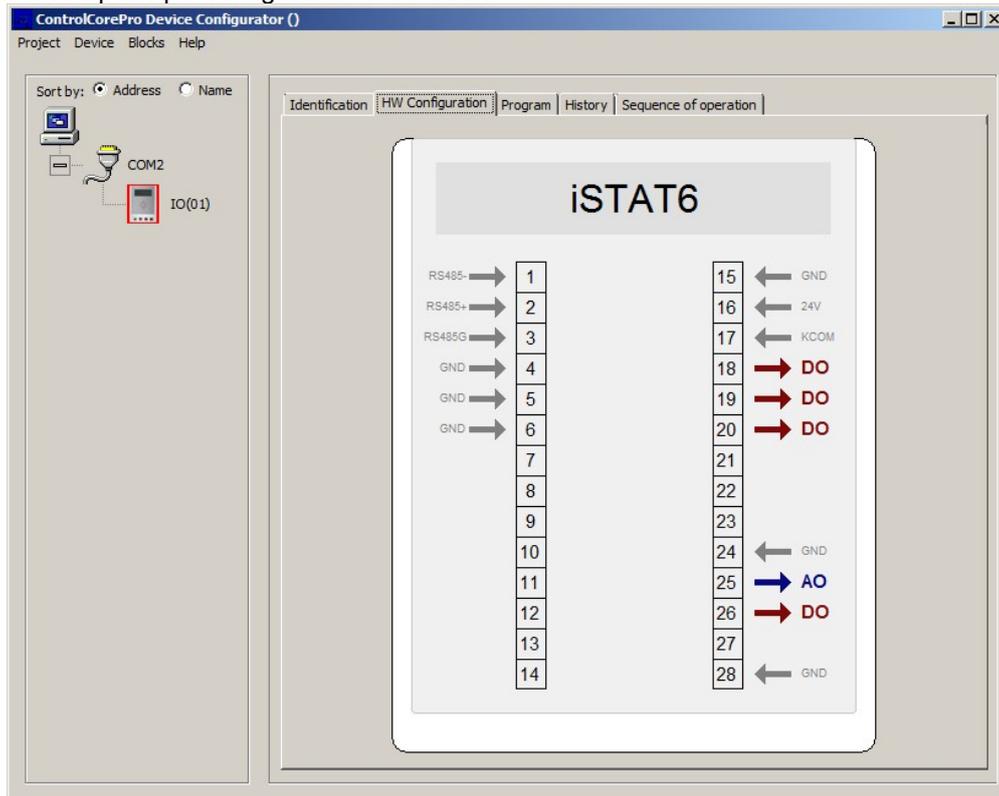

Note that you can also set the device's display parameters from here by selecting the display and double clicking it.

Programming your device: Device Programming

The Program tab allows access to the device's embedded program, from this tab you can create the actual control algorithm that will be downloaded into the device. Initially, the programming tab is set to the edit mode which is indicated by the selection of the arrow cursor icon at the top.

To create a program you will select function blocks from the top row of icons; drop them in the workspace; and then double-click them to access each block's parameter definition dialog. Connections among the blocks are done by setting the input parameter selections on each block's parameter definition dialog. Note that the I/O pins previously defined on the Hardware Configuration tab appear as blocks on the left or right hand side of the workspace, left hand for inputs, right hand side for outputs.
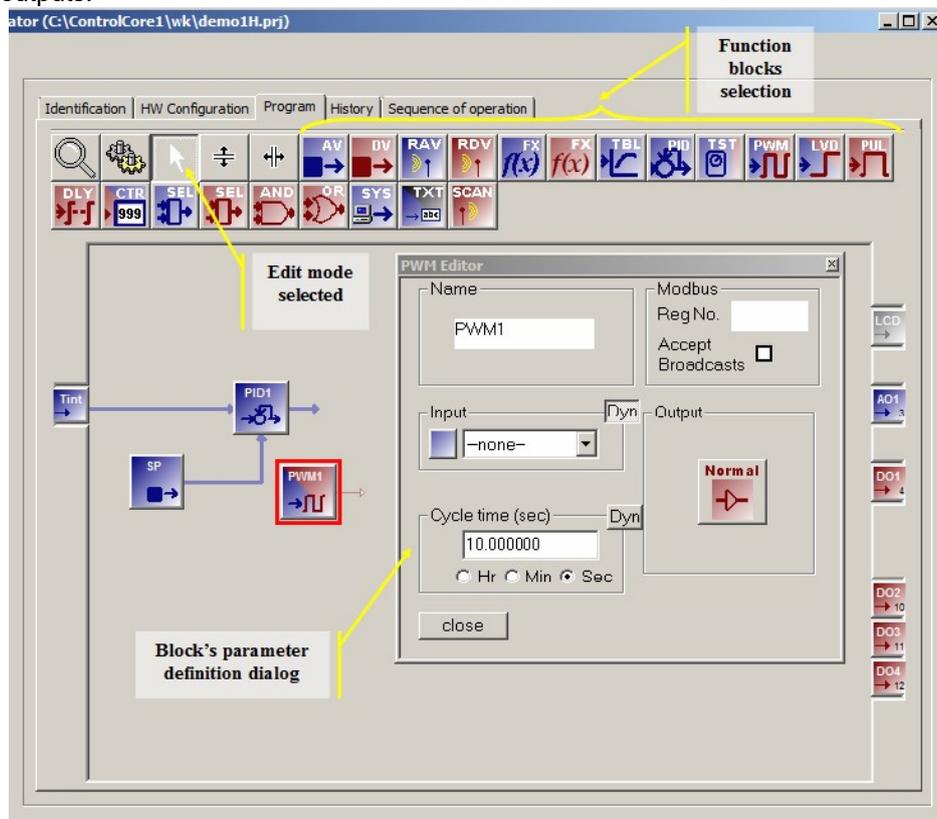
For the purpose of this tutorial we will create a simple temperature controller.

Step 1	Insert a PID function block. This block implements a standard PID control algorithm.
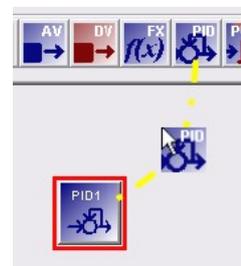
Step 2

Insert an AV function block. This type of block simply holds an user settable analog value. It will be the temperature set-point. Its name can be changed to "SP" by double-clicking the block and modifying the "Name" field.

Step 3

Now we will do the input connections to the PID1 function block. Double-click the PID1 function block to invoke its parameter definition dialog and set the "Input" parameter to "Tint" the temperature input block.
Similarly, set the PID1's "Set Point" parameter to "SP", the block we created in the previous step.

Step 4

We will use the analog output block AO1 to drive the heater element. We assume that we have a power amplifier compatible with a 0 to 5V signal. To connect it to the PID control output, double-click the AO1 block to invoke its parameter definition dialog and set the "Input" parameter to "PID1".

Step 5

Alternatively, if we do not have a linear power amplifier to drive the heater element we can drive it via a PWM (Pulse Width Modulation) block. To add it, simply select and drop a PWM type function block and connect its "Input" parameter to "PID1". Then connect its output to one of the digital output blocks, DO1, for example.
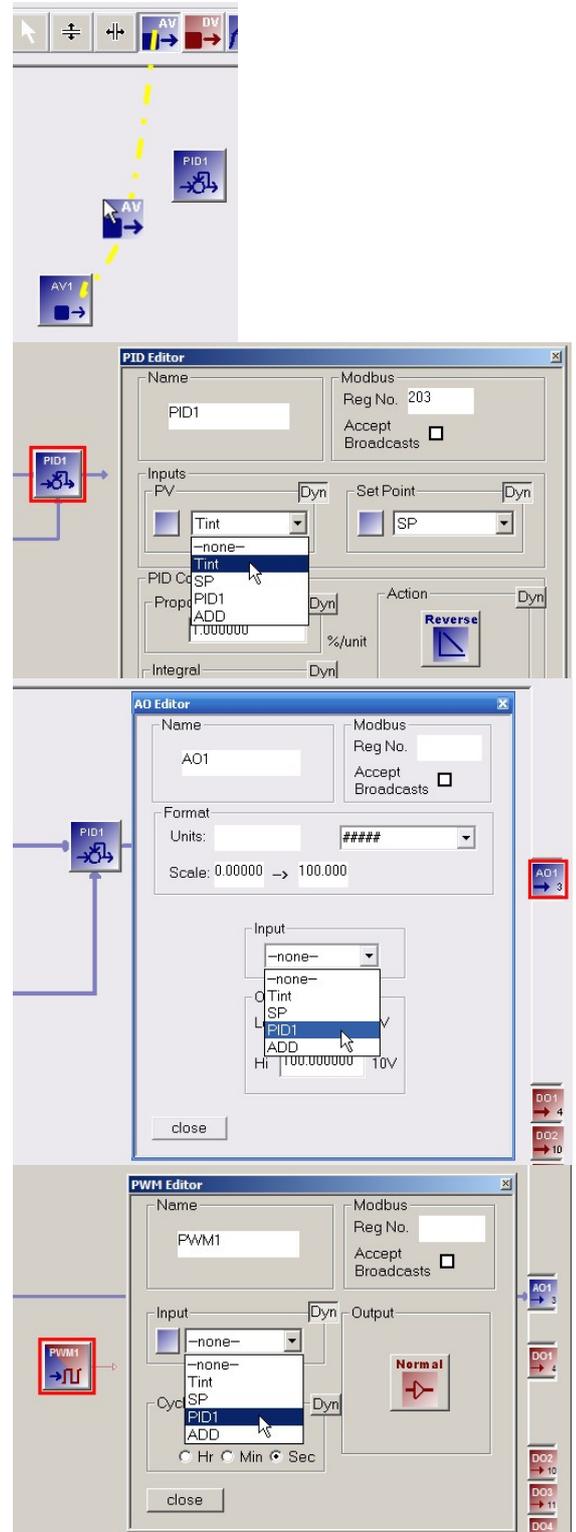
Congratulations you have successfully created a temperature control program for your device. This program can now be downloaded and run on the target embedded device. Your final program should look like the one shown below.
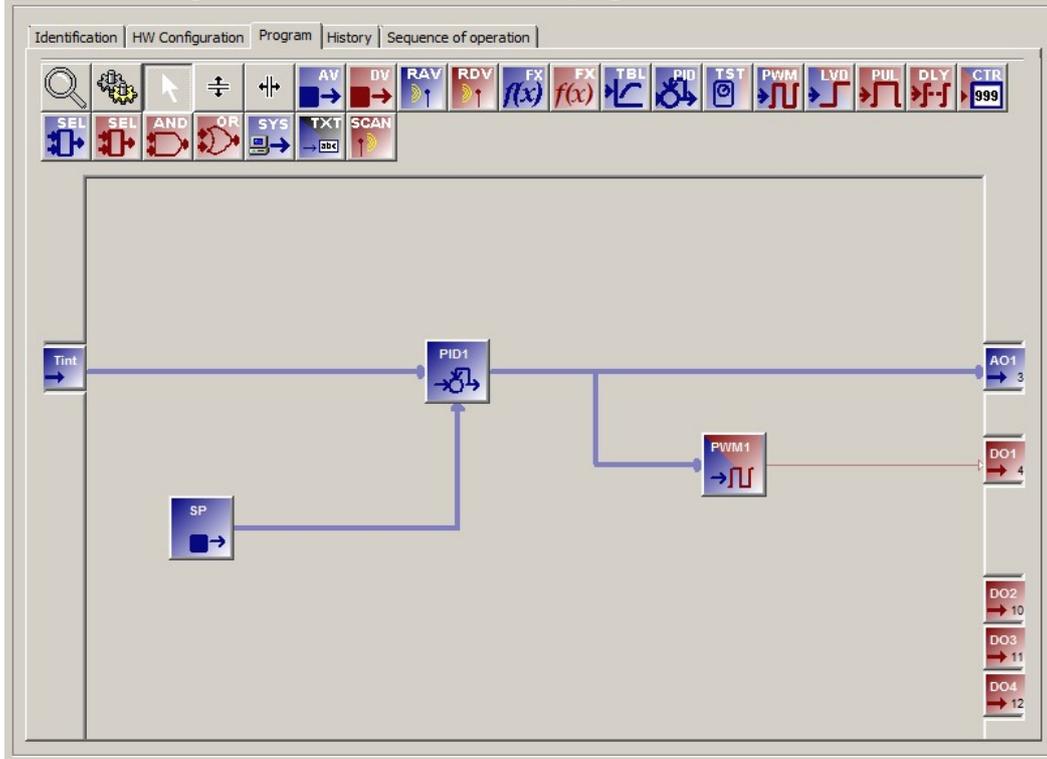
**Programming your device: Run Mode & Live Monitoring**

Once you have completed a program you can download it into the embedded device and monitor it in real-time. To do this, simply change the Program tab mode from edit mode to Run mode by selecting the run mode icon.

When you select the monitor mode the system will automatically download the current program into the selected device. It will pop up a dialog to request that you confirm, select OK to continue.

While in RUN mode you can add monitors to any one of the connection lines to view or set its current value in real-time. To add a monitor hover over a connection with the mouse, a monitor will appear automatically. If you move away the monitor disappears. To make a monitor remain permanently click on the connection line before moving away. There are two types of monitors, analog monitors that show the current value and digital monitors that show the state (ON or OFF). To remove a monitor, click the X on the upper right hand side.
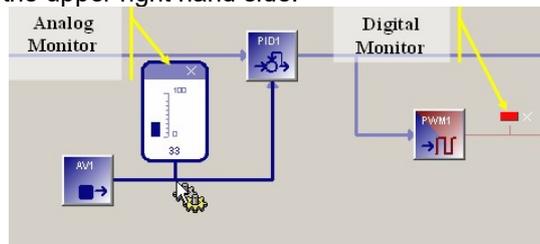
To set a value on an analog monitor, click and drag the mouse along the bar graph or type the desired value while the monitor is highlighted. For digital monitors the state can be toggled by clicking on the LED or by pressing the space bar while the LED is highlighted.

When hovering over a block in run mode, a box will indicate the block's Modbus register, if defined. Also, the function block's dialog box information can be displayed in run mode, but all fields are read only. Blocks can be moved around for a better view.