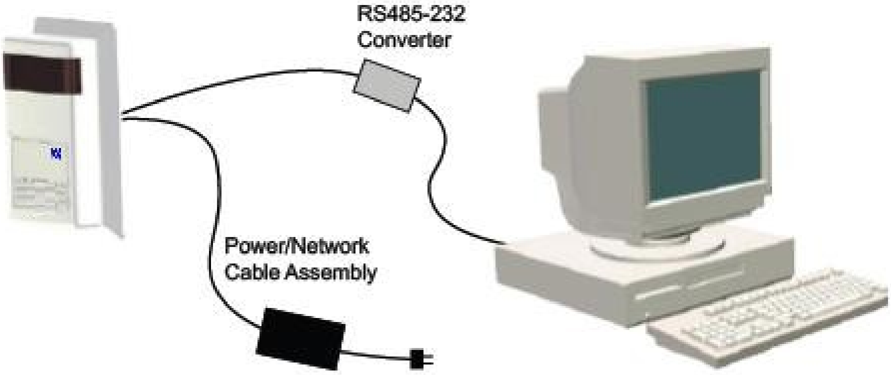
INTRODUCTION

This manual covers technical detail information relating to the ION Core Graphical Programming Application. The ION Micron range is used as the controller hardware for illustration.

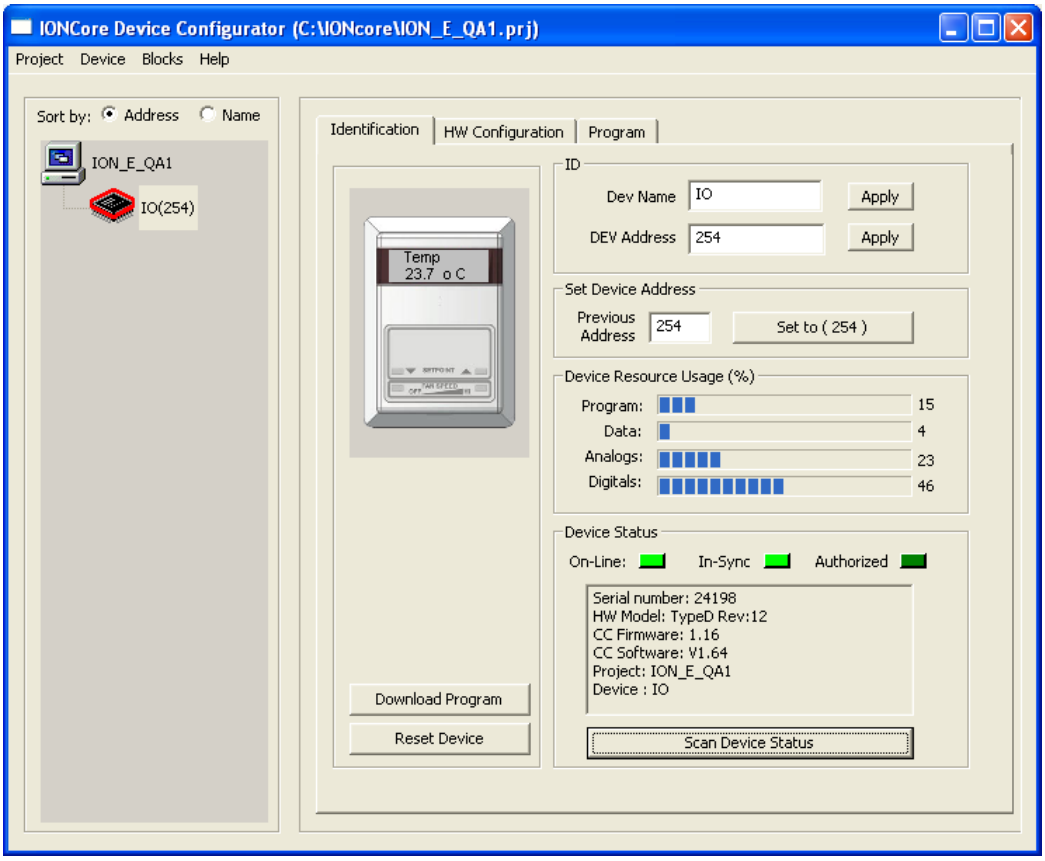
Getting Started

ION Micron to PC communication connection default values;
Address = 254; Baud Rate = 19K2; Data Bits = 8 Bits; Stop Bits = One; Parity = None



Identification Page

When communication is established the following ION Core Identification page shows various ION firmware and software version information ..



Sort by: ● Address ○ Name

ION_E_QA1
IO(254)

ID
Dev Name: IO
DEV Address: 254

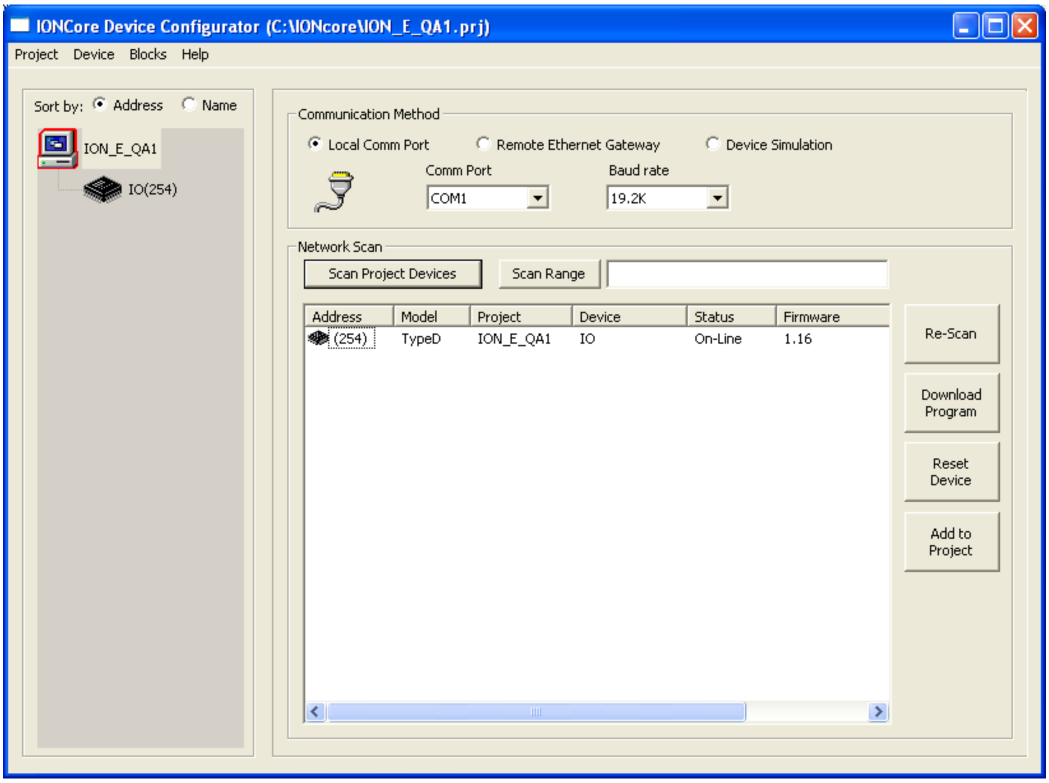
Set Device Address
Previous Address: 254
Set to (254)

Device Resource Usage (%)
Program: 15
Data: 4
Analogs: 23
Digitals: 46

Device Status
On-Line: 🟩 In-Sync 🟩 Authorized 🟩

Serial number: 24198
HW Model: TypeD Rev:12
CC Firmware: 1.16
CC Software: V1.64
Project: ION_E_QA1
Device : IO

Network Scan and Device Simulation Mode Page



Sort by: ● Address ○ Name

Communication Method

● Local Comm Port ○ Remote Ethernet Gateway ○ Device Simulation

Comm Port: COM1

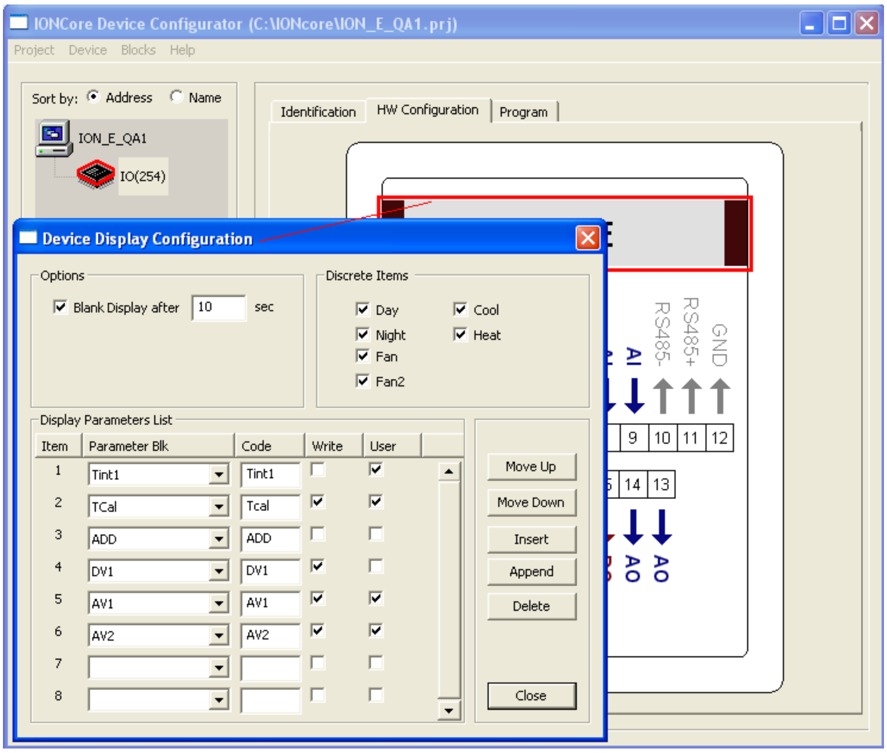
Baud rate: 19.2K

Network Scan

Address	Model	Project	Device	Status	Firmware
(254)	TypeD	ION_E_QA1	IO	On-Line	1.16

Hardware Configuration Page

Display Configuration



Sort by: ● Address ○ Name

Options

☑ Blank Display after 10 sec

Discrete Items

☑ Day ☑ Cool

☑ Night ☑ Heat

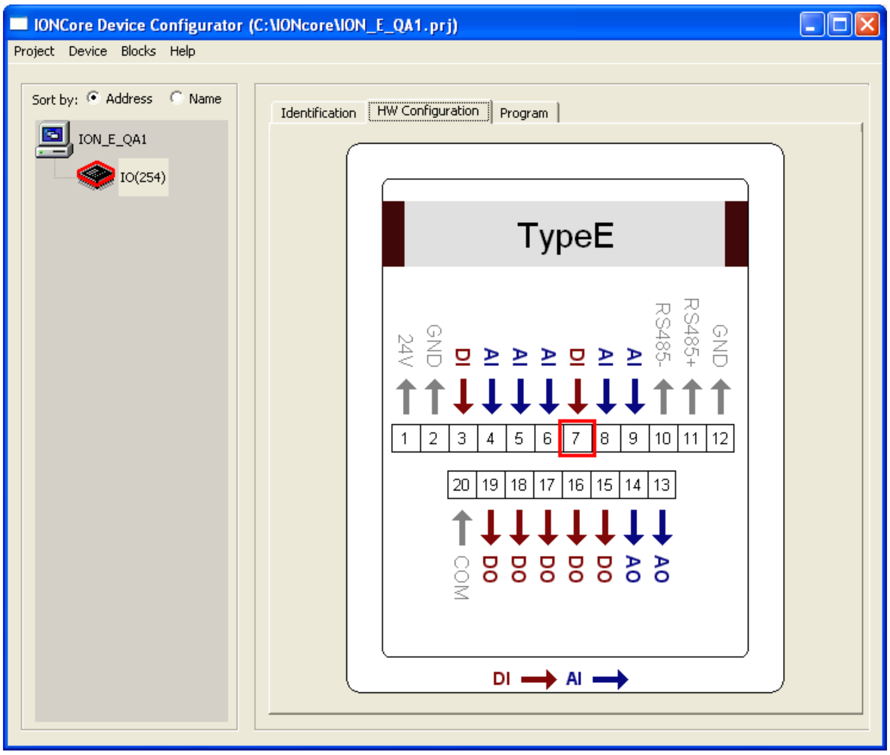
☑ Fan

☑ Fan2

Display Parameters List

Item	Parameter Blk	Code	Write	User
1	Tint1	Tint1	☐	☑
2	TCal	Tcal	☑	☑
3	ADD	ADD	☐	☐
4	DV1	DV1	☑	☐
5	AV1	AV1	☑	☑
6	AV2	AV2	☑	☑
7			☐	☐
8			☐	☐

Input/Output Configuration Page



IONCore Device Configurator (C:\IONcore\ION_E_QA1.prj)

Project Device Blocks Help

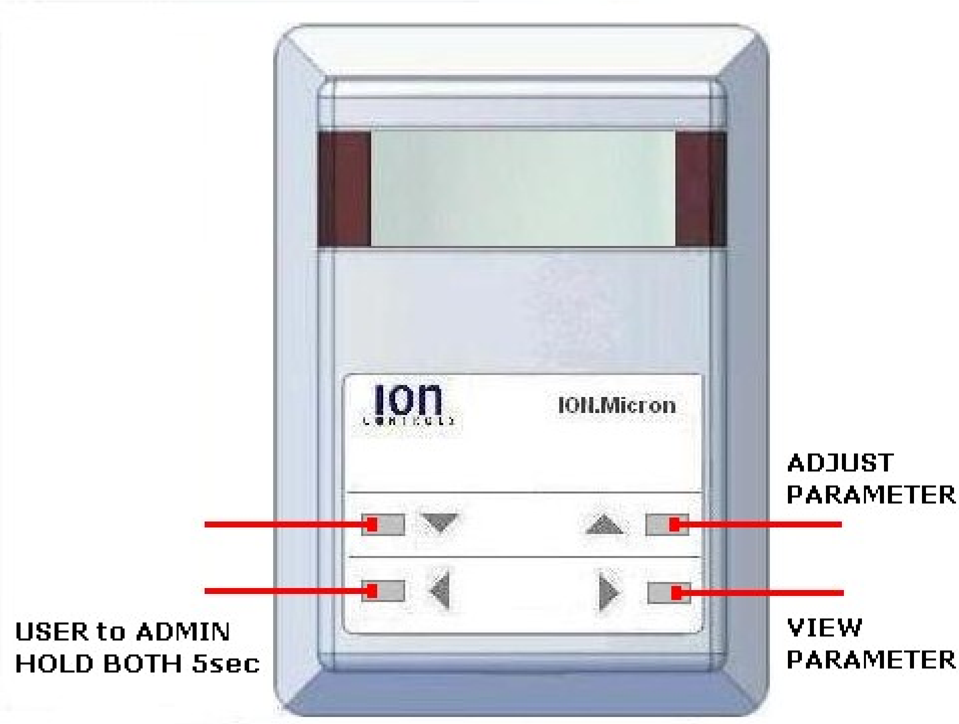
Sort by: ● Address ○ Name

ION_E_QA1

IO(254)

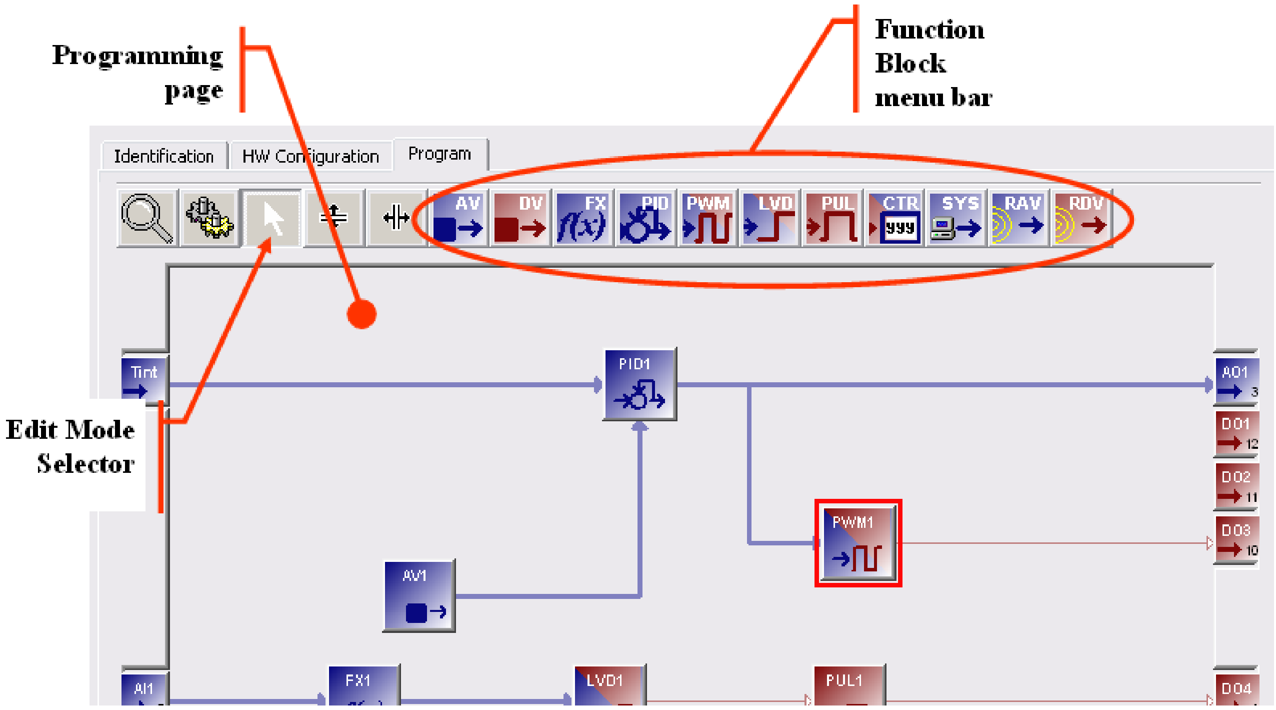
Identification | HW Configuration | Program

Keypad Functionality



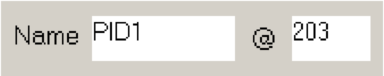
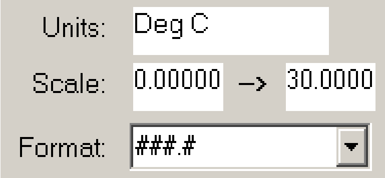
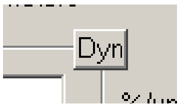
Device Programming

Device programming is done via the ***Program Tab*** while in edit mode. To create an embedded control algorithm for a device you must select and deposited a series of interconnected function blocks onto the programming page. Click on the desired function block menu bar and then click on the program page in the position where you want it to be place. Once a function block is on the programming page you can double-click it to expose it's definition dialog box.



Each function block definition dialog box contains parameters that are specific to the type of function block selected. The following describes parameters that are common to all function block types.

Function Block Common Parameters

Parameter	Description
 Name PID1 @ 203	All function blocks allow you to define a name for the block. A name can be any combination of alphanumeric characters up to a maximum of 8 that is unique from other blocks on the device's program page. The trailing field after the '**@**' allows you to define a Modbus register number to access this block's value from a remote Modbus compatible controller.
 Units: Deg C Scale: 0.00000 -> 30.0000 Format: ###.#	Most analog blocks definition dialog boxes will contain these three parameters. Together they define the analog value's range and how it should be displayed.
 Dyn	Many of the function block parameters have the (Dyn) option. This indicates that the parameter can optionally be dynamic. Selecting the DYN button will allow you to assign an other block to provide the value of the parameter instead of entering it statically.

Analog input (AI) blocks represent a physical analog input on the device.

AI definition parameters

Parameter	Description
Name	Function block name, see common parameters
Units	Parameter's engineering units, see common parameters
Scale	Parameter's scale, see common parameters
Format	Parameter's format, see common parameters
Type	Type of input.
Lo	The engineering unit value that corresponds to the minimum input value.
Hi	The engineering unit value that corresponds to the maximum input value.
Filter	Selects the filter to be applied to the input. Moving the slider to the right increases the amount of filtering. Placing it at the far left turns filtering off entirely.

Analog output (AO) blocks represent a physical analog output on the device.

AO definition parameters

Parameter	Description
Name	Function block name, see common parameters
Units	Parameter's engineering units, see common parameters
Scale	Parameter's scale, see common parameters
Format	Parameter's format, see common parameters
Input	The function block that will drive the value of the analog output.
Output Scale	The engineering value that will correspond to the 0Volt and 10Volt output values.

Digital input (DI) blocks represent a physical digital input on the device.

DI definition parameters

Parameter	Description
Name	Function block name, see common parameters

Digital Output (DO) blocks represent a physical digital output on the device.

DO definition parameters

Parameter	Description
Name	Function block name, see common parameters
Input	The function block that will drive the value of the digital output.

The **LCD** function block allows a control strategy to adjust the currently displayed parameter on the Micron. It only appears on the programming page when the **Add Display Control Block** option is checked in the **Device Display Configuration** window.

LCD definition parameters

Parameter	Description
Name	Function block name, cannot be edited.
Input	Analog input value whose value controls which parameter is displayed on the Micron. The value (1..n) corresponds to the display item as it appears in the "Device Display Configuration" parameter list. If the value is zero or the input was not defined, the Micron display behaves normally.

Analog value (AV) function blocks simply hold a user settable analog value. It can be used as an input for another function block, a typically use would be to hold the set-point value for a control block. AV values are saved to nonvolatile memory.

AV definition parameters

Parameter	Description
Name	Function block name, see common parameters
Units	Parameter's engineering units, see common parameters
Scale	Parameter's scale, see common parameters
Format	Parameter's format, see common parameters
Initial Value	The initial value, if enabled, will be the default value for the block when the program is initially downloaded.

Digital value (DV) function blocks simply hold an user settable state value. DV state values are saved to nonvolatile memory.

DV definition parameters

Parameter	Description
Name	Function block name, see common parameters
Initial Value	he initial state, if enabled, will be the default value for the block when the program is initially downloaded.

F(x) Function Block - The F(x) function block will allow you to derive a value from an arbitrary mathematical formula involving any output from the device's existing function blocks. You can mix analog and digital values liberally. A digital value used in an analog context has the value 1.0 (On) or 0.0 (Off), and an analog value used in a digital context is On (value != 0.0) or Off (value = 0.0).

F(x) definition parameters

Parameter	Description
Name	Function block name, see common parameters
Units	Parameter's engineering units, see common parameters
Scale	Parameter's scale, see common parameters
Format	Parameter's format, see common parameters
Formula	The mathematical formula to compute the block's output value.

Mathematical Operators - The mathematical formula can contain the following operators:

Operator	Description
+	Add
-	Subtract
*	Multiply
/	Divide

Comparison Operators

Operator	Description
>	Greater Than
<	Less Than
>=	Greater Than or Equal
<=	Less Than or Equal
=	Equal

Boolean Operators

Operator	Description
! not	NOT
& and	AND
| or	OR

Operator Precedence

Operator precedence is divided into three groups as shown below. Multiple operators from the same group are evaluated left to right. You can use parenthesizes liberally to force a particular evaluation order as required.

Precedence	Operators
Highest	**! ()**
	*** / &**
Lowest	**+ - > < >= <= = |**

Supported functions

Operator	Description
dt()	The elapsed time interval, in seconds, since last iteration
sin(x)	Trigonometric SIN function x is in radians
cos(x)	Trigonometric COS function x is in radians
sqrt(x)	Square root
min(x,y)	lesser of x or y
max(x,y)	greater of x or y
mod(x,y)	Floating point modulus, remainder of x / y

The **table function block** implements a custom configurable lookup table. It allows the user to enter series of lookup table values that transform an input value into an output value.

TBL definition parameters

Parameter	Description
Name	Function block name, see common parameters
Units	Parameter's engineering units, see common parameters
Scale	Parameter's scale, see common parameters
Format	Parameter's format, see common parameters
Input	The initial value, if enabled, will be the default value for the block when the program is initially downloaded.
Table Points List	List of input values and respective output values separated by a coma.

Proportional-Integral-Derivative Control (PID) function blocks implement a standard PID control algorithm.

PID definition parameters

Parameter		Description
Name		Function block name, see common parameters
Input		The measured value or (***process value***) that is to be controlled.
Set Point		The target value (***set point***) that the control block is striving to attain.
Control parameters	Proportional	Control's ***Proportional*** setting.
	Integral	Control's ***Integral*** setting.
	Derivative	Control's ***Derivative*** setting.
	Deadband	If the error value is less than this value the control will not change its output. This can help prevent unnecessary cycling.

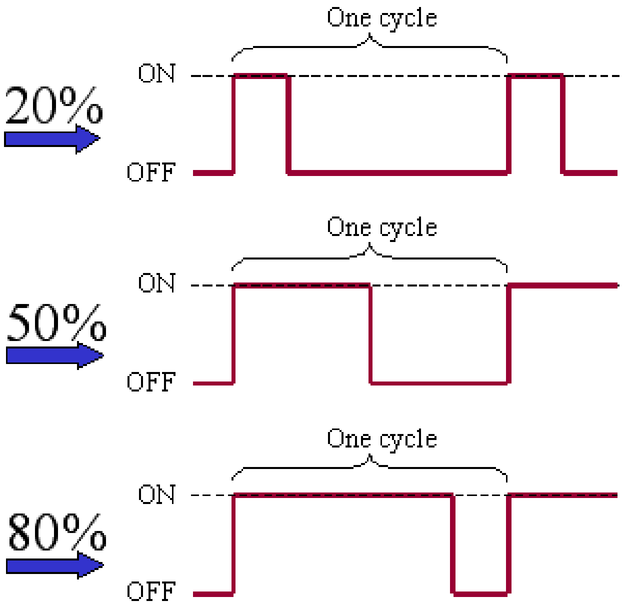
A **proportional–integral–derivative controller (PID controller)** attempts to correct the ***error*** between a measured ***process variable*** and a desired ***set point*** by calculating and then outputting a corrective action that can adjust the process accordingly.
The PID controller algorithm involves three separate parameters; Proportional, Integral and Derivative values. The ***Proportional*** value determines the reaction to the current error, the ***Integral*** determines the reaction based on the sum of recent errors and the ***Derivative*** determines the reaction to the rate at which the error has been changing. The weighted sum of these three actions is used to adjust the process via a control element such as the position of a control valve or the power supply of a heating element. By "tuning" the three constants in the PID controller algorithm the PID can provide control action designed for specific process requirements. The response of the controller can be described in terms of the responsiveness of the controller to an error, the degree to which the controller overshoots the setpoint and the degree of system oscillation. Note that the use of the PID algorithm for control does not guarantee optimal control of the system. Some applications may require using only one or two modes to provide the appropriate system control. This is achieved by setting the gain of undesired control outputs to zero. A PID controller will be called a PI, PD, P or I controller in the absence of the respective control actions. PI controllers are particularly common, since derivative action is very sensitive to measurement noise, and the absence of an integral value prevents the system from reaching its target value due to the control action.

The **Thermostat Control Algorithm** (TST) function block implements a simple ON/OFF type thermostat control algorithm.

TST definition parameters

Parameter	Description
Name	Function block name, see common parameters
Input	The measured value or (***process value***) that is to be controlled.
Set Point	The target value (***set point***) that the control block is striving to attain.
Deadband	If the error value is less than this value the control will not change its output. This can help prevent unnecessary cycling.
Hysteresis	Specifies the accumulate time require before the output changes from off to on or on to off. This would eliminate relay chatter for noisy inputs for example.
Reverse / Direct	Toggle button between **Reverse** and **Direct** mode (push to switch between modes): - In **Reverse** mode, the TST responds when the input is below the set-point , heating application for example. - In **Direct** mode, the TST responds when the input is above the set-point, cooling application for example.

Pulse-width modulation (PWM) function blocks translate an analog value to a modulated digital output. The digital output will be a square wave whose duty cycle is proportional to the analog input level. The diagrams below show typical output waveforms for 20% input, 50% input and 80%.



PWM definition parameters

Parameter	Description
Name	Function block name, see common parameters
Input	Analog input value that will drive the waveform's duty cycle. This input should vary from 0 to 100.
Cycle time	Total time of one cycle of the output waveform. The time is expresses in seconds. Valid values are 1 .. 600 seconds.

Level Detector (LVD) function blocks monitor an analog value and will set its output high whenever the monitored value goes over a specified threshold.

LVD definition parameters

Parameter	Description
Name	Function block name, see common parameters
Input	The analog value to monitor
Threshold value	The threshold value, if the input is above this value the output will be high.

Pulse (PUL) function blocks will produce a specific fixed length pulse when triggered. The pulse width is adjustable.

PUL definition parameters

Parameter	Description
Name	Function block name, see common parameters
Slope	Determines the trigger for the pulse, low-to-high or high-to-lo.
Input	The digital input to monitor.
Pulse Width	The duration of the pulse when triggered.
Shape	The shape of the output pulse, (normally low and high for the duration of the pulse) or (normally high and low for the duration of the pulse)

Delay (DLY) function blocks will introduce a specific fixed delay when triggered. The delay time is adjustable.

DLY definition parameters

Parameter	Description
Name	Function block name, see common parameters
Input	The digital input to monitor.
On Delay	The delay, expressed in seconds, for the output to toggle to an ON state. If the input returns to the OFF state the delay is reset.
Off Delay	The delay, expressed in seconds, for the output to toggle to an OFF state. If the input returns to the ON state the delay is reset.

Counter (CRT) function blocks count the number of transitions occurring on the specified digital input. The output is the accumulated count represented as an analog value. The count value can be set by the user or it can be cleared by a transition on a designated digital input.

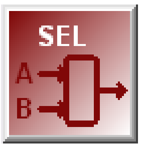
CTR definition parameters

Parameter	Description
Name	Function block name, see common parameters
Units	Parameter's engineering units, see common parameters
Scale	Parameter's scale, see common parameters
Format	Parameter's format, see common parameters
Input	Input digital value to monitor and count transitions. Transitions can be defined as low-to-high or high-to-lo.
Reset	Input digital value to monitor to clear the count value. Count will be cleared on transitions of the digital state. Transitions can be defined as low-to-high or high-to-lo.

The **Analog selector block** (SEL) allows an digital value to select one of two analog input values.

SEL definition parameters

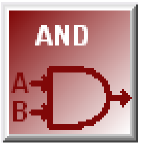
Parameter	Description
Name	Function block name, see common parameters
Units	Parameter's engineering units, see common parameters
Scale	Parameter's scale, see common parameters
Input (A)	One of the two analog values from which the selection is done.
Input (B)	The other analog value from which the selection is done.
Selector Input	When (Low) the A input is used to drive the output value. When (High) the B input is used to drive the output value. If the input is not define the output is not driven and it is possible to set the output value directly, via ION Core or Modbus.



The **Digital selector block** (SEL) allows an digital value to select one of two digital input values.

SEL definition parameters

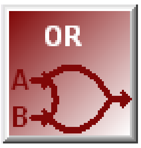
Parameter	Description
Name	Function block name, see common parameters
Input (A)	One of the two digital values from which the selection is done.
Input (B)	The other digital value from which the selection is done.
Selector Input	When (Low) the A input is used to drive the output value. When (High) the B input is used to drive the output value. If the input is not define the output is not driven and it is possible to set the output value directly, via ION Core or Modbus.



The **AND** function block (AND) implements the digital AND function (Boolean operator AND)

AND definition parameters

Parameter	Description
Name	Function block name, see common parameters
Input (A)	First digital input.
Input (B)	Second digital input.
Output Normal / Invert	Toggle button between **Normal** and **Invert** mode (push to switch between modes): - In **Normal** mode, the result is the logical AND of the two inputs. - In **Invert** mode, the result is the inverse of the logical AND of the two inputs., i.e. NAND function.



The **OR** function block (OR) implements the digital OR function (Boolean operator OR)

OR definition parameters

Parameter	Description
Name	Function block name, see common parameters
Input (A)	First digital input.
Input (B)	Second digital input.
Output Normal / Invert	Toggle button between **Normal** and **Invert** mode (push to switch between modes): - In **Normal** mode, the result is the logical OR of the two inputs. - In **Invert** mode, the result is the inverse of the logical OR of the two inputs., i.e. NOR function.

SYS

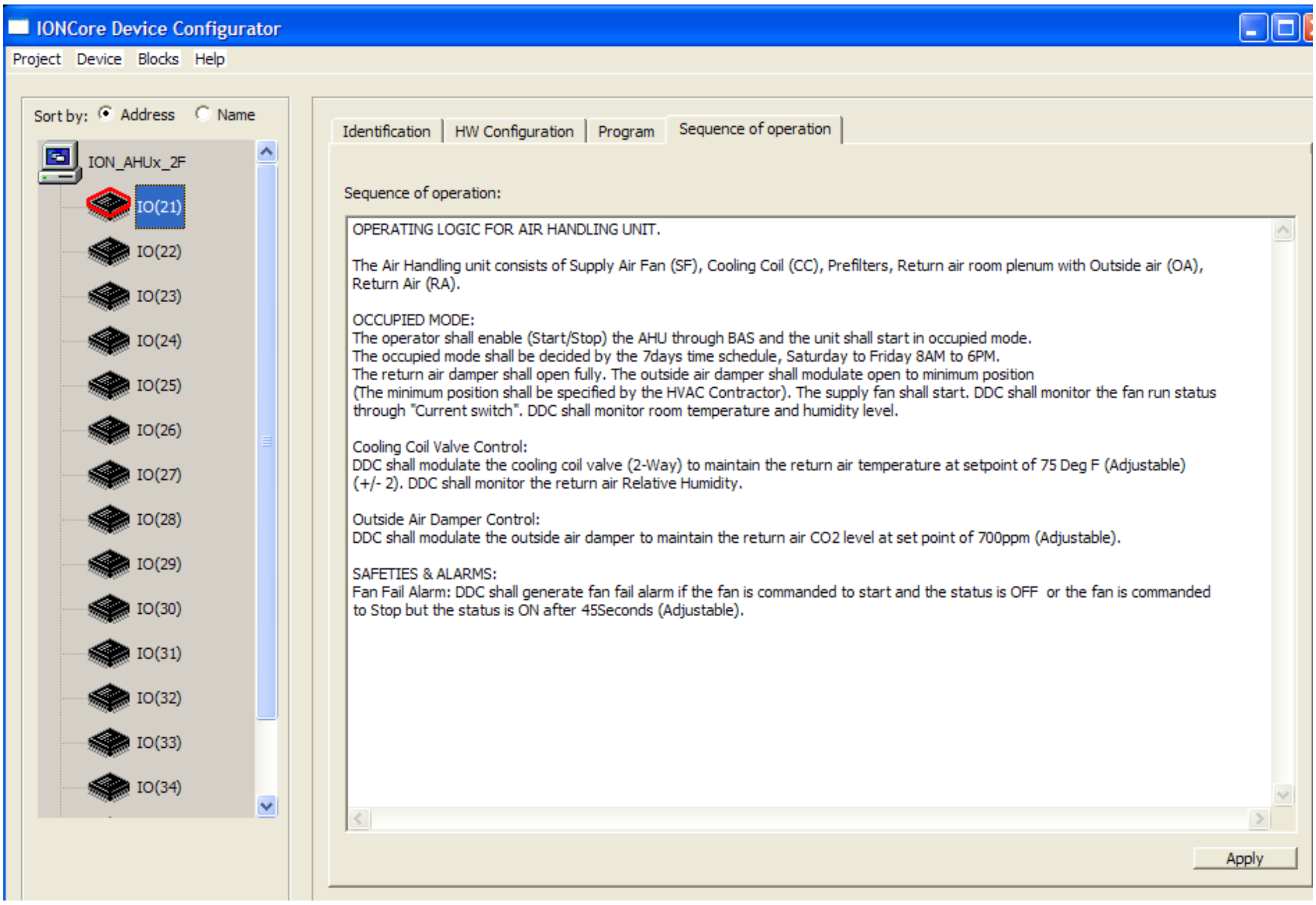
System (SYS) function blocks provide access to information inside the device.

SYS definition parameters

Parameter	Description	
Name	Function block name, see common parameters	
Units	Parameter's engineering units, see common parameters	
Scale	Parameter's scale, see common parameters	
Format	Parameter's format, see common parameters	
System Parameter	The system parameter required. The following options are available:	
	Device Address	The device's current RS485 communication address
	Avg Loop Time	The average loop time required to compute all program function blocks.
	Max Loop Time	The maximum loop time required to compute all program function blocks and service network requests.
	Last Comm Time	Number of seconds since the last network communication from a supervisory type controller.

Sequence of Operation Tab

Input text description of program for reference aid.



Sequence of operation:

OPERATING LOGIC FOR AIR HANDLING UNIT.

The Air Handling unit consists of Supply Air Fan (SF), Cooling Coil (CC), Prefilters, Return air room plenum with Outside air (OA), Return Air (RA).

OCCUPIED MODE:
The operator shall enable (Start/Stop) the AHU through BAS and the unit shall start in occupied mode.
The occupied mode shall be decided by the 7days time schedule, Saturday to Friday 8AM to 6PM.
The return air damper shall open fully. The outside air damper shall modulate open to minimum position
(The minimum position shall be specified by the HVAC Contractor). The supply fan shall start. DDC shall monitor the fan run status through "Current switch". DDC shall monitor room temperature and humidity level.

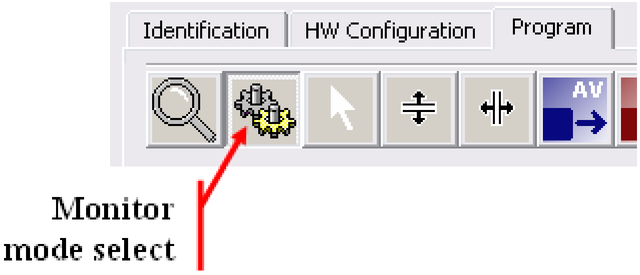
Cooling Coil Valve Control:
DDC shall modulate the cooling coil valve (2-Way) to maintain the return air temperature at setpoint of 75 Deg F (Adjustable) (+/- 2). DDC shall monitor the return air Relative Humidity.

Outside Air Damper Control:
DDC shall modulate the outside air damper to maintain the return air CO2 level at set point of 700ppm (Adjustable).

SAFETIES & ALARMS:
Fan Fail Alarm: DDC shall generate fan fail alarm if the fan is commanded to start and the status is OFF or the fan is commanded to Stop but the status is ON after 45Seconds (Adjustable).

Device Live Monitor Mode

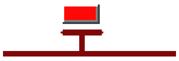
The live monitor mode is available via the ***Program Tab***. To enter the live mode, select the monitor mode icon from the device's program tab.



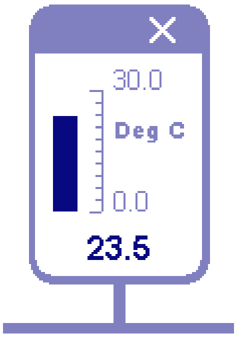
Monitor mode allows you to interact with the actual operating device. You can view all of the device's internal values live and/or set values interactively.

When you first select the monitor mode button the system will verify that the corresponding physical device contains the correct program corresponding to your current configuration. If not it will request confirmation to upload the current program to the device. Most of the time the device's program will be synchronized so the upload will not be necessary.

You can add monitors (digital or analog) to any function block output by hovering over it's output connection with the mouse. To leave a monitor on the screen simply click on the connection, a monitor will be left at this position even if you move away. To remove a monitor click the (X) on its top right side.



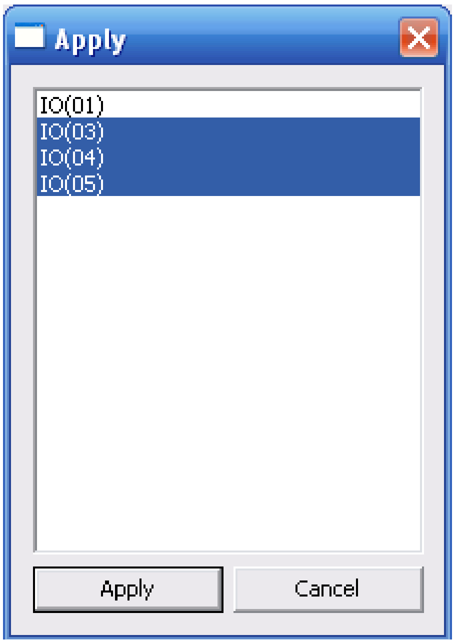
Digital monitors show the state of a digital signal. You can change the state interactively by clicking on the LED with the mouse. If the monitor is highlighted you can also toggle the state by pressing the keyboard's space bar or return key. You can also set or clear the state via the keyboard Up-Arrow and Down-Arrow keys.



Analog monitors show the block's current value as a bar graph and also in numerical form at the bottom . The scale and units are retrieved from the block's definition parameters. You can change the value interactively by clicking or dragging on the bar graph with the mouse. If the monitor is highlighted you can also set the value through the keyboard, simply type the desired value and then hit the return key. You can also increment or decrement the value via the keyboard Up-Arrow and Down-Arrow keys.

Duplicating Device Configurations
After you have completed the configuration of a single device you can copy that configuration on to other devices of the same type. This can avoid the tedious job of configuring hundreds of devices that all have the same hardware configuration and program.

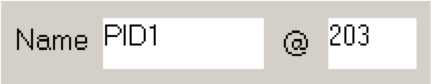
To copy a device configuration select the configured device on the main tree view and then select **Device -> Copy Multi Device(s)** from the top menu bar. This will display the dialog box shown below. Select all the devices in the list that are to receive the configuration and press "Apply". Note that only devices of the same model are shown in the list.



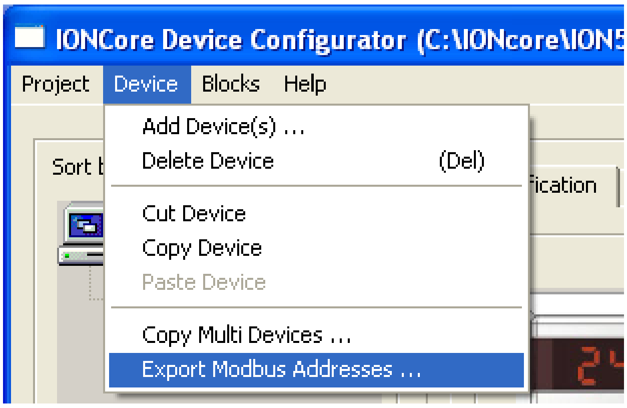
Note that the list shown is a standard Microsoft Windows multi-select list. You can drag the mouse over several items or use combinations of shift and control key with the mouse to make your selection.

Modbus Access

Any device equipped with the ION Core firmware automatically supports the Modbus protocol and can provide and accept data from a Modbus master controller. Any one of the defined function blocks can be assigned a Modbus register address, this address will be used by a Modbus master controller to access the block's value. To make a block's value accessible via Modbus simply enter a register number following the blocks name as shown below:


Name PID1 @ 203

You can also create a text file describing all Modbus addresses for a device. Select the "**Export Modbus Address ...**" item from the "**Device**" menu as shown



This will create a text file containing all the defined Modbus register addresses for the selected device, a sample is shown below:

```
Modbus Addresses for Device: IO(01)
Address, Blk Name, Format
00101, DO1, (0)Off/(1)On
00102, DO2, (0)Off/(1)On
00103, DO3, (0)Off/(1)On
00104, DO4, (0)Off/(1)On
00105, PWM1, (0)Off/(1)On
00201, Tint, o C X10
00202, AI1, o C X10
00203, PID1, % X10
00204, SP, o C
00205, OUT,
```

end