

# P864 HELP FILE

---



P864

# Contents

1. Programming Tool	2
1.1. Programming Tool Configuration	2
1.2. Programming Tool Introduction	3
1.3. Input & Output Configuration (Program File)	4
1.4. Programming Tool Environment	5
1.5. Page Menu and Canvas	10
1.6. Logic Blocks	18
1.7. Points	19
1.8. Null Values Interpretations	20
1.9. Subroutines	21
2. Logic Blocks	23
2.1. And Gate	23
2.2. Anti Short Cycle	24
2.3. Change of State Detector	25
2.4. Comment	26
2.5. Comparitor Hysteresis	27
2.6. Delay on Break	28
2.7. Delay on Make	29
2.8. Division	30
2.9. Enthalpy Calculator	31
2.10. Exclusive OR Gate	32
2.11. Floating Motor Controller	33
2.12. Gated Transfer	34
2.13. High / Low Limiter	35
2.14. High Selector	36
2.15. Increment Decrement	37
2.16. Input	38
2.17. Latch Gate	39
2.18. Linear Converter	40
2.19. Low Selector	41
2.20. Multiplication	42
2.21. OR Gate	43
2.22. Output	44
2.23. PID Controller	45
2.24. Power Of	47
2.25. Pulse Width Modulation	48
2.26. Restrictor	49
2.27. Runtime	50
2.28. Single Shot	51
2.29. Subtraction	52
2.30. Switch	53
2.31. Transfer Data	54
2.32. Two Input Linear Converter	55
2.33. Velocity Pressure to FPM Converter	56
2.34. Within	57
3. Device Configuration	58
4. System Settings	61
5. Inputs/Outputs Live Data	64
6. Airflow Sensing	66
7. Optergy Air Wireless Sensor	70
Appendix	71
P864 Diagnostic Points	71

## Version History

### 1.0 (P864 v 5.6.8)

+ Programming Tool

### 2.0 (P864 v 5.7.9)

+ 1.9 Subroutines and Subcaller Topic

+ 3.0 Device Configuration

+ 4.0 System Settings

+ 5.0 Inputs & Outputs

### 3.0 (P864 v 5.7.9)

+ Airflow Sensor Topic

### 4.0 (P864 v 5.7.9)

+ P864 Diagnostic Points

### 5.0 (P864 v 5.7.9)

+ Optergy Air Wireless Sensor

# 1. Programming Tool

## 1.1 Programming Tool Configuration

*This setting applies to Proton or Optergy Enterprise Programming Tool Settings.*

The Programming Tool Configuration page allows users to edit the value that points retain in the event that they timeout.

There are currently two choices:

**Zero/Inactive:** When a point times out the value will default to zero or inactive.

**Last Known Value:** When a point times out the value will default to the most recent value before the timeout occurred.

### Edit Programming Tool Settings

To update the default timeout value, click  (Edit). A popup will appear that displays a dropdown box which the user can select their preferred default timeout value. Once the user has selected their desired value click save to apply.

## 1.2 Programming Tool Introduction

*Required Privilege: Programming Tool*

The creation of programmable logic for the control of BACnet devices is called Programming Tool otherwise known as Direct Digital Control (DDC). Programming Tool sequences are stored and carried out in Optergy. All this can be done within the Google Chrome Browser (Chrome is the only Supported Browser) and no other add-ons or downloads are required for this functionality.

The Programming Tool offers a number of predefined logic blocks that have specific functions, those blocks are executed based on fixed values or values coming from other BACnet points such as AV's and BV's either hosted locally in Optergy or other BACnet devices on the BACnet network.

Each block has a unique sequence number and that number determines the order in which the Programming is executed. Once the Programming passes through all the sequence numbers, it starts over and executes the Programming sequentially once again.

The speed at which the Programming takes to cycle through is dependent on a number of factors such as number of blocks, load of the overall system and speed of the network.

*This feature is available in Proton & Optergy Enterprise.*

To view what speed the Programming is executing at go to: Activity and Monitoring > Log Viewer

Select the Programming Tool log to view the speed.

Here is an example of Programming execution time from the log file:

Tue Mar 29 10:09:52 AEDT 2016 Sequence: 597.0ms, 5 Writes: 400ms, Total: 997.0ms

This shows the time it has taken to execute the sequence and the write time. These are added together to show the total execution time.

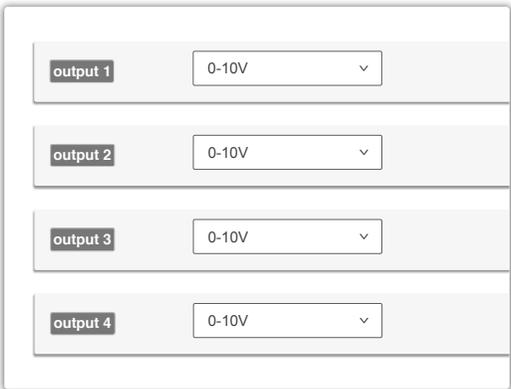
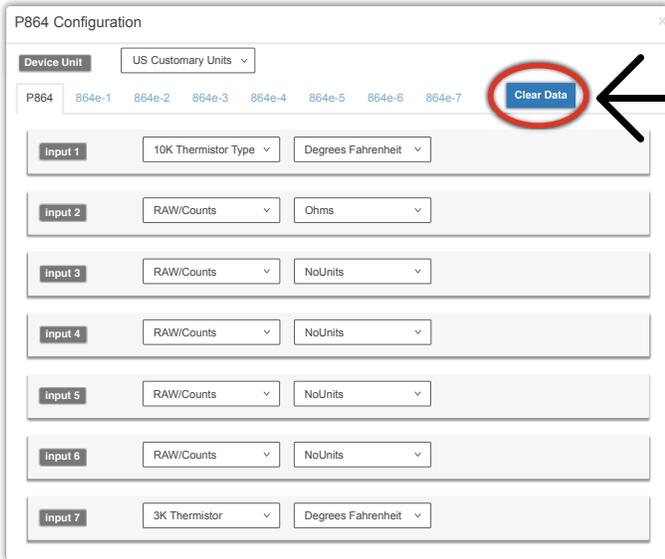
## 1.3 Input & Output Configuration (Program File)

Each program can have the following input configuration preset in the program. Doing this assures that each recipient controller has precisely the same attributes.

For P864 and 864e the settings are:

- Default Units (Metric or US Customary) (default is Metric)
- Input 1-8 (input configuration type) (default is Raw Counts)
- Input 1-8 (unit choice) (default is No Units)
- Output 1-4 (voltage or current) (default is Voltage)

Clear Data simply clears all the user settings and returns to default.



### For the 864e

There is another button to configure Output Relinquish Default, this is used to preset the outputs to a known value if the 864e loses contact with the P864.

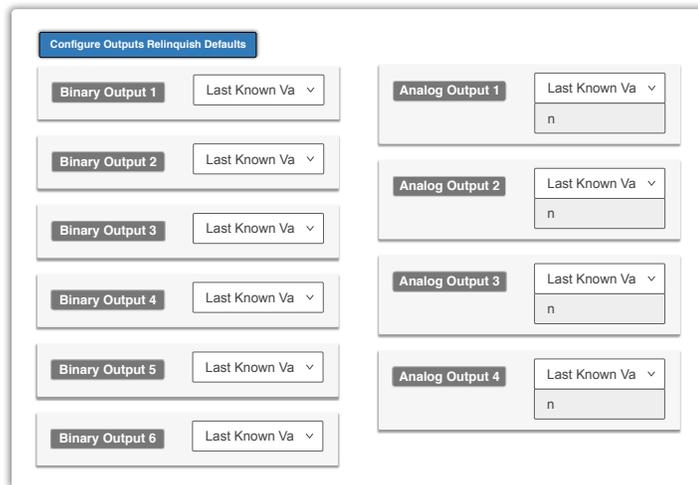
Options include:

#### Binary Output 1-8:

- Last Known Value (default)
- Active (On)
- Inactive (Off)

#### Analog Value 1-4:

- Last Known Value (default)
- User Specified (0-100%)

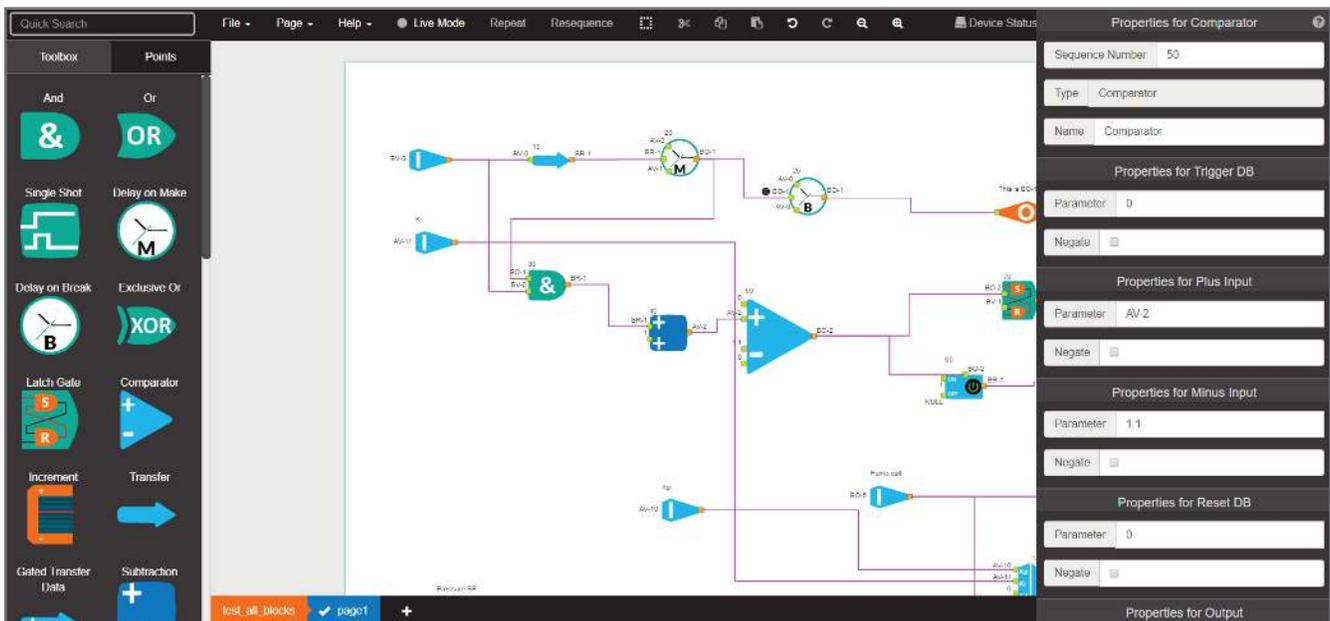


## 1.4 Programming Tool Environment

### The Optergy Programming Tool Environment

There are four distinct environments in the Programming Tool web interface:

1. Toolbox (holds all the logic blocks)
2. The Page
3. Logic Block Properties (appears when you right click on a logic block)
4. Tool Bar



#### Tool Box

The Tool Box holds all the logic blocks that can be dragged onto the page and used in the Programming Tool.

#### Page

The Page is the canvas that the Programming Tool logical blocks are arranged and connected.

#### Logic Block Properties

The Logic Block Properties can be edited, this is where the BACnet device and object details can be entered along with fixed values or negating (reversing the property).

#### Toolbar

The Toolbar is the place where you can edit the program, pages enable live mode and provides tools for editing the page.

## Programs and Pages

Optergy Programming Tool module uses the concept of Programs and Pages to create programming logic.

The easiest way to describe this would be to think of folders and files.

Think of the program as a folder and think of pages as the files.

All the pages in a program make up the entire logic and are compiled together.

The reason we have the concept of pages is some Programming Tool programs can be very large and span multiple things, for example you can have all logic related to an AHU on one page, and all logic related to boilers on another page. When combined they all form part of the same program.

## Programs

When you enter the Programming Tool you will be prompted with 3 options; New Program, Open Program and Import Program.

## New Programs

If you select new program you will be prompted to enter the name of the program and the name of the first page.

## Open Program

If you select open program you will be prompted to select an existing program and the page you want to open within that program.

The first screenshot shows the 'Optergy DDC Tool' main menu with three options: 'New Program', 'Open Program', and 'Import Program'. The second screenshot shows the 'Create New' dialog box with the 'Program' dropdown set to 'Custom Block'. It contains fields for 'Program Title', 'Page Title', and 'Password Protection' (checked). Below these are 'Password' and 'Confirm Password' fields, and a 'New Program' button. The third screenshot shows the 'Open' dialog box with the 'Program' dropdown set to 'Custom Block'. It displays a list of programs and pages to choose from: '3\_0\_2\_rc14\_blank', '864\_backup\_test', 'blank', 'blank2', and 'descriptors'. Each item has a folder icon and a colored dot (green for the first, blue for the others).

## File Menu

The program menu is the where you will find additional menus related to the entire program.

### New

If you select new program you will be prompted to enter the name of the program and the name of the first page.

### Open

If you select open you will prompted to select an existing program and the page you want to open.

### Save

If you select save this will save the current page to the current program.

### Save As

If you select save this will allow you to save the current page with a different name in the current program *OR* save as another existing program with the same or different name.

The save as window also allows you to create new program to save the page to with the same or a different name.

### Import

If you have a program that you exported from another Optergy installation, you can then import that program.

If you select import you will get a browse window to navigate for the program you want to import.

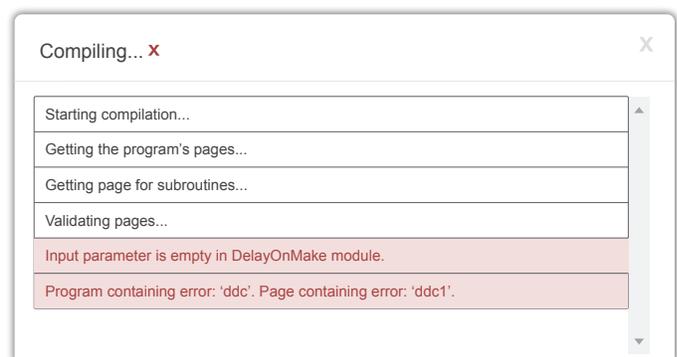
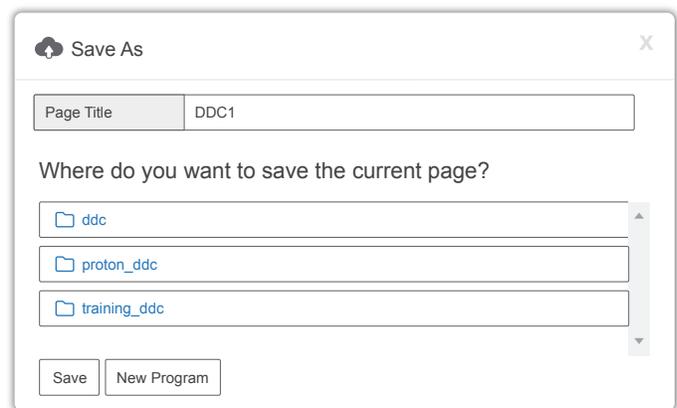
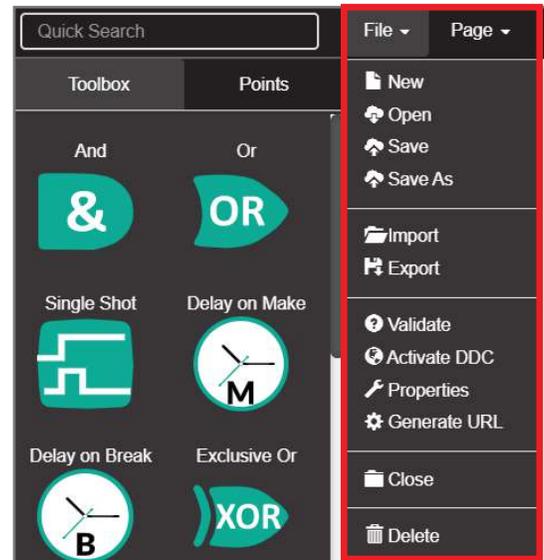
**Note:** Programs are exported as zip files and should be imported as the same zip files.

### Export

Selecting export will export the current program as a zip file to Chromes default download directory. **Note: This is usually the downloads directory for windows users.**

### Validate

Prior to publishing the programming logic you can validate it, this will check for errors while compiling and advise of any errors that will need to be fixed prior to publishing.



## Activate DDC

Publishing will actually compile the programming logic and then publish. If there are errors in the logic, the logic will not publish and list all the errors found. Once the logic is published successfully it will start executing the logic in Optergy.

## Programming Logic Points

*(Located on left side bar)*

If there are no errors in the logic and the logic has been saved, you can select logic points to get a summary of the points used throughout all the pages.

DDC Points

Successfully compiled program.

Analog Values

Binary Values

BV-0

BV-1

BV-3

BV-4

Branches

Device Instances

Manual Inputs

## Properties

Properties provides a quick summary and editing functions of the current program, current page canvas size, first and last sequence number and by what number you want to increment the sequence numbers.

Program Properties

Current Program

Title	DDC Demo	File Name	ddc_demo
-------	----------	-----------	----------

Current Page

Title	DDC1	File Name	ddc1
Canvas Width	4000	Size	Custom
Canvas Height	4000	Orientation	Portrait

Increment Sequence Numbers by 10

Pages in the Current Program

- ddc1

Save Changes Cancel

## Current Program

Provides the details of the current program title and file name, you can also edit the program title here and the file name will automatically change as well.

## Current Page

Provides the details of the current page title and file name, you can also edit the page title here and the file name will automatically change as well.

## Canvas Width and Height

The page canvas size you are currently working on can be modified in width and height.

The minimum size is 300 x 300 pixels. The default is 4000 x 4000 pixels. The maximum is 10,000 x 10,000 pixels.

## Size

You can set the size of the canvas to standard page sizes. The canvas width and height will automatically adjust.

## Orientation

If you set the size to a standard page size then you can also select the orientation as either portrait or landscape.

## Increment Sequence Number By

You can define the amount sequence numbers increase by every time you add function block.

The maximum number of sequence numbers that can be used are, 1,999,999,999.

## Pages in the Current Program

Provides a summary list of the pages within the current program.

## Save Changes

If you make any changes in program properties window, ensure you select save changes for the changes to take effect.

## Close

To close the current page on the current tab you are in select close.

**Note:** This does not close the browser tab, the tab will remain open and you will need to close it manually.

If you only have one tab open then both the page and the program will be closed.

## Delete

Select delete to delete an entire program and all its pages.

You will get a warning message prior to deleting a program to confirm the deletion.

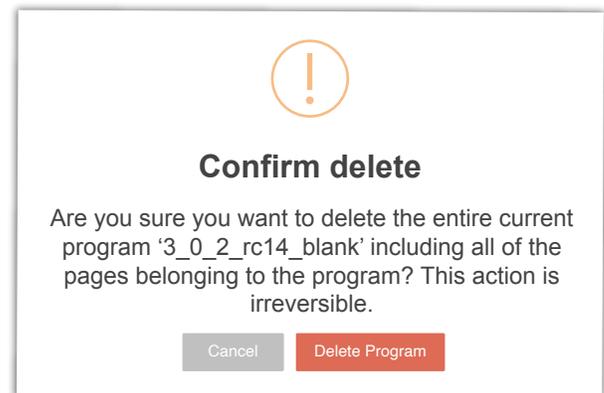
## Encrypted

False means the program can be decoded by others.

True means the contents cannot be read by anyone except this program.

## Generate URL

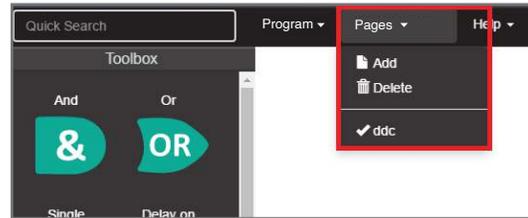
This is used in the Display Tool to allow the program and live data to be inserted as a view only object using the Proton/Optergy Enterprise DDC display object.



## 1.5 Page Menus and Canvas

### Page Menu

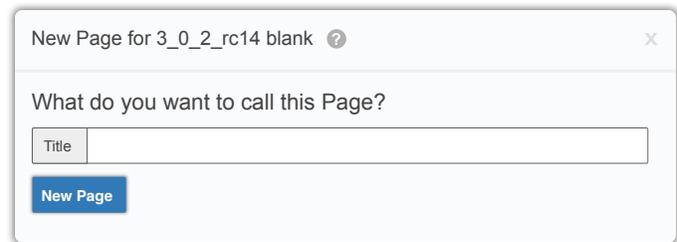
The page menu is where you will find additional menus related to pages.



### New

Select add to add a page to the current program.

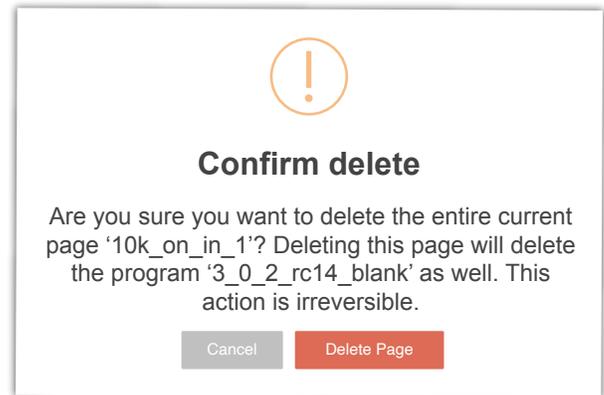
This will open a new browser tab and ask you for a page title for the new page.



### Delete

Select delete to delete the current page.

You will get a warning message prior to deleting a page to confirm the deletion.



### Print

Allows the user to print using local PC printer options.

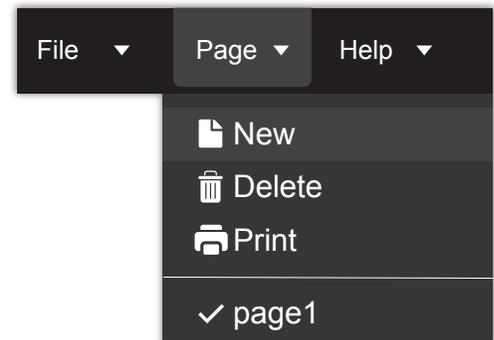
### Page

**Note:** If you are deleting a page and the page is the only page in the program, then the program will also be deleted, as there must be a minimum of one page for a program to exist.

If you have multiple pages in your program you will have all the pages listed under the page menu.

The page with the tick next to it will be the current page you are working on in the current tab you are in.

When selecting one of the other pages this will open in another tab.



**Note:** As we are working in a browser window and referencing a URL, you may actually have the same page open in different tabs or it may be opened by different people. Try and avoid opening the same page in different tabs. Also avoid multiple people working on the same pages. Last save wins.

## Page (Canvas)

The page canvas is the place you can add logic blocks to from the tool box in order to create the program.

## Adding Logic Blocks to the Canvas

To add a logic block, hover over the block you want to add, the pointer will change to a hand, you can then left click and hold on the block and drag it to the canvas.

## Moving Around the Page Canvas

To move the page canvas, left click and hold on a free area on the canvas and you can move the canvas around.

## Logic Block Selection

There are two methods for selecting logic blocks. (Selected blocks will have a blue outline around the block to visually show that it has been selected.)

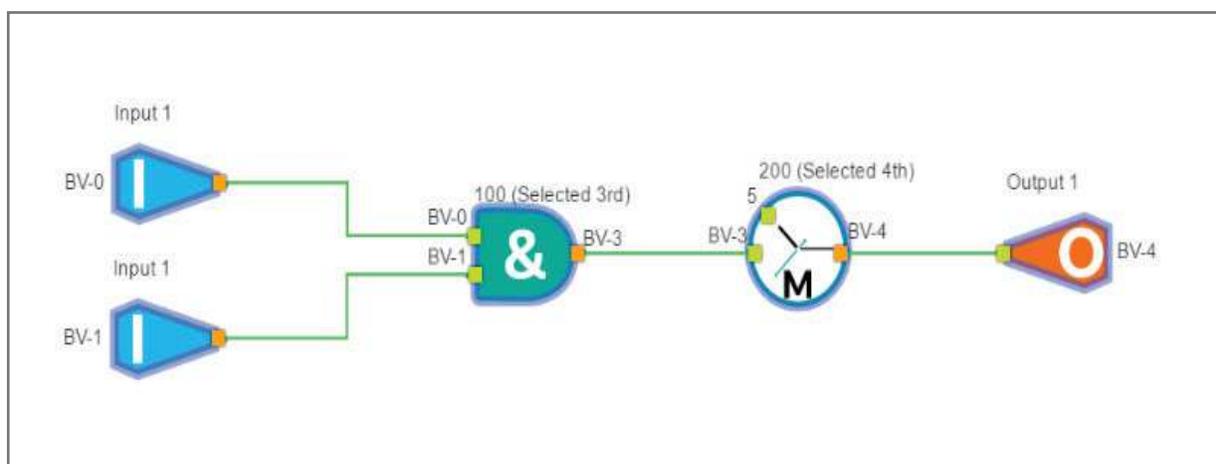
### Method 1:

CTRL and Click - if you hold the CTRL key on the keyboard while left clicking on the block you will select it, if the block is already selected the same action will de-select the block.

### Method 2:

Select the marquee tool  on the toolbar, using the marquee tool you can select multiple blocks on the canvas at the same time. A rectangular box will appear showing the blocks that have been selected within that area. Once selected you can use the CTRL and Click to de-select any blocks not required to be selected. Vice versa you can select some blocks using CTRL and Click and then use the Marquee Tool.

**Note:** As making selections you will notice next to each block the order in which the blocks were selected. The reason for this will become evident when re-sequencing the blocks sequence numbers.



## Deleting Logic Blocks and Connection Lines

To delete logic blocks and connection lines, hold SHIFT and left click on the block or line you want to delete.

## Moving Blocks on the Page Canvas

To move logic blocks around the page canvas, left click and hold on logic block and you can move the logic block to a different part of the page canvas.

## Moving Multiple Blocks on the Page Canvas

To move multiple logic blocks around the page canvas, select multiple blocks as described in the logic block selection, left click and hold on any of the selected logic blocks and you can move multiple logic blocks to a different part of the page canvas.

## Aligning Blocks on the Page Canvas

There are a number of alignment tools for aligning the logic blocks. Once one or more blocks are selected the alignment tool bar will appear just below the main toolbar.



Aligns the blocks to the block that is the topmost of all selected.



Aligns the blocks to the block that is the bottom most of all selected.



Aligns the blocks to the block that is the rightmost of all selected.



Aligns the blocks to the block that is the leftmost of all selected.



Aligns the blocks horizontally in the middle of all the selected blocks.



Aligns the blocks vertically in the middle of all the selected blocks.

## Connection Lines between Logic Blocks

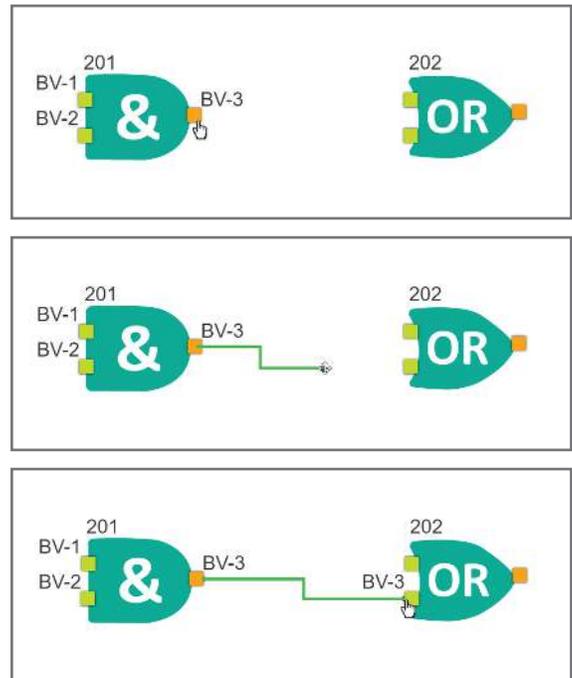
Logic blocks can be connected together using connection lines. Connection lines assist when reading the program and shows the relationship between logic blocks, as well as making it easy to follow.

To join, two blocks connect an output of a block (outputs are in orange) to input of another block (inputs are green).

To ensure you are on the orange output of a block the cursor will change to a hand. When the cursor is a hand you can click and drag the connection line to the input of another logic block.

**Note:** Once logic blocks are connected, moving the logic block will also move the connection line.

Also making a connection with the output defined, will propagate the defined output to the input. As shown in the example above, BV-3 was automatically propagated to the input.



## Repeat Logic Points

You can select logic block if you want to repeat the blocks multiple times.

Select the block you want to repeat and click on **Repeat** in the toolbar.

A repeat settings window will pop up where you can enter how you want the repeat to execute.

## Repeat Settings

### Repeat Modules

How many times do you want the block to repeat?

### Direction of Repeat

Use the drop down box to choose if you want to repeat horizontally or vertically.

### Repeat Parameter's Increment Value

You can select how the parameter values of the block are to increment by. For example if you want the sequence number to increment by 10 then each block will increment

by 10 from the original block selected. So if you repeat 2 times and the original block was 10, then the first repeated block will be 20, then second 30. This behavior is the same for all the other repeat parameters.

Repeat Settings
✕

Repeat Module	2	Times	Horizontally ▾
Repeat Parameters		Increment By	
<input checked="" type="checkbox"/> Sequence Number			10
<input checked="" type="checkbox"/> Branch Numbers			1
<input checked="" type="checkbox"/> Device Instances			1
<input checked="" type="checkbox"/> Object Instances			1
<input checked="" type="checkbox"/> Array Indices			1
<input checked="" type="checkbox"/> Substitution Numbers			10
<input checked="" type="checkbox"/> Manual Input			1

Repeat Module

## Resequencing Logic Blocks

You can resequence logic blocks based on the order you select them or using the marquee tool (see logic block selection) or the existing sequence order.

After you make your selections select **Resequence** and this will open up the Resequence Settings window.

## Resequence Starting From

This is the number the first block or existing first block sequence number will be.

## In Increments Of

This is the number of sequences between each block.

**Note:** It is best practice to leave gap of sequences between each block. This way if you are making edits and need to add a few blocks then you do not need to resequence all the block and you can manually assign the block sequence numbers.

## New Sequence Numbers Should Be Ordered By

The new sequence numbers can be ordered by the existing sequence number or the order of selection.

## Cut, Copy and Paste

You can cut, copy and paste blocks individually or multiple blocks and connection lines. This can be done by using the toolbar buttons.



You can also use key board shortcuts:

Copy = CTRL + C

Cut = CTRL + X

Paste = CTRL + V

**Note:** You can copy between pages, in the same browser session.

## Undo Redo

You can undo and redo actions on the page canvas. This can be done by using the toolbar buttons.



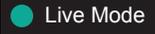
You can also use key board shortcuts:

Undo = CTRL + Z

Redo = CTRL + Y

## Live Mode

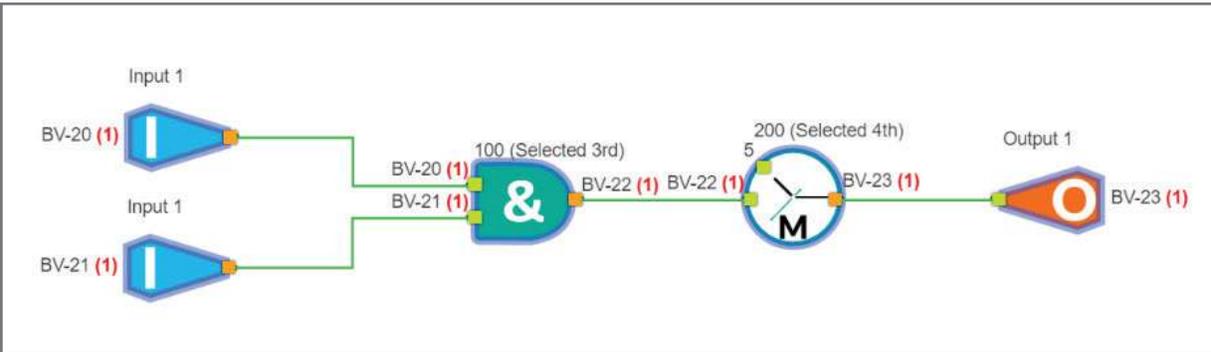
When a program is compiled and running it can be viewed in the page canvas in real time.

To view the program in real time select Live Mode on  from the toolbar.

Then either select a block or a number of blocks in the visible page canvas.

The selected block/s will then display the live value next to the inputs and outputs.

To stop viewing the live selected blocks click anywhere in an unused part of the page canvas.



Note: You are still in live view mode at this point so selecting blocks again using the marquee tool or individually will display the live values. To exit live view mode select Live Mode off  from the toolbar.

## Logic Block Properties

To view and edit the logic block properties you must right click on the block. Once you right click on the block, this will open up the properties to the right of the page canvas.

## Block Properties

### Sequence Number

The sequence number of the block. This can be edited.

### Type

The block type. This cannot be edited.

### Name

The name of the block. This can be edited, but is not visible on the page canvas.

### Number of Inputs

The number of inputs can be set for the following blocks:

AND OR ADDITION HIGH SELECTOR LOW SELECTOR

The selection can be made between 2 and 16 inputs. The block size will automatically change on the page canvas.

Properties for And	
Sequence Number	201
Type	AndGate
Name	And
Number of Inputs	2 <input type="button" value="Change"/>
Properties for Input 1	
Name	Input 1
Parameter	BV-1
Negate	<input type="checkbox"/>
Properties for Input 2	
Name	Input 2
Parameter	BV-2
Negate	<input type="checkbox"/>
Properties for Output	
Name	Output
Parameter	BV-3
Negate	<input type="checkbox"/>
<input type="button" value="Delete Module"/>	

## Properties for Inputs and Outputs

### Name

The name of the Input. This can be edited, but is not visible on the page canvas.

### Parameter

The parameter the block will reference to execute a function.

If a non-compliant parameter has been entered the Parameter will be shaded red to indicate an incorrect parameter entry.

For binary blocks this can be a:

- BR (Branch Point)
- BV (Binary Value)
- BO (Binary Output)
- BI (Binary Input)
- MV (Multi-State Value)
- MO (Multi-State Output)
- MI (Multi-State Input)
- 1 (True) or 0 (False)

For analog blocks this can be a:

- BR (Branch Point)
- AV (Analog Value)
- AO (Analog Output)
- AI (Analog Input)
- MV (Multi-State Value)
- MO (Multi-State Output)
- MI (Multi-State Input)
- or a number.

To reference a point on the Programming Block input or output the following syntax need to be used:

'point type'-'point instance'

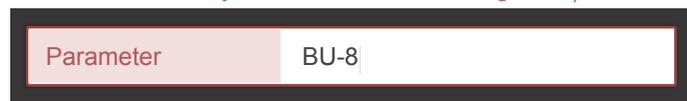
eg: BV-0 is referring to Binary Value 0, AI-50 is referring to Analog Input 50.

By default, Programming Tool writes at Priority Array 14 if the point has a Priority Array, or Present Value otherwise. To write at another priority, the following syntax should be used:

'point type'-'point instance'-'priority array'

eg: BV-0-8 refers to Binary Value 0 at Priority 8.

*Incorrect value entry shows red color/shading over parameter*



**Note:** BR (Branch Points) are software points that are only used in the Program. Please note that live values of BR points will not be visible in Live Mode. There are 1000 Branch points available that can be used in the program, BR-0 to BR-999. The same branch points can be used multiple times in the same program.

## Negate

Ticking the Negate check box will reverse the action of the incoming parameter.

For binary blocks, True (1) inputs will be read as False (0) and False (0) inputs will be read as True (1).

For analog blocks, it will invert values between 0 and 100, but not limited to 0 and 100 for example;

-50 = 150

0 = 100

50 = 50

100 = 0

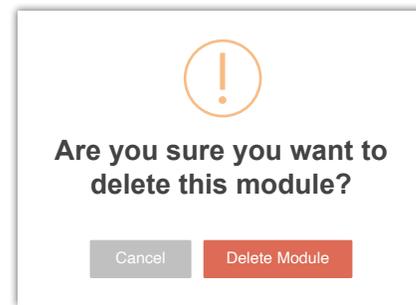
125 = -25



**Note:** Values that are negated will appear with a solid dot above the input or output for binary values and a solid triangle above the input or output for Analog Values.

## Delete

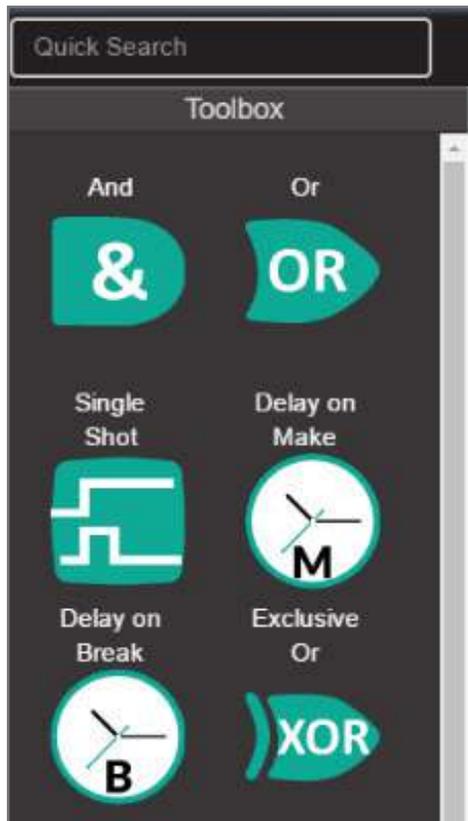
The block can be deleted from the properties, if you choose this you will get a warning message to confirm deletion.



## 1.6 Logic Blocks

### Toolbox

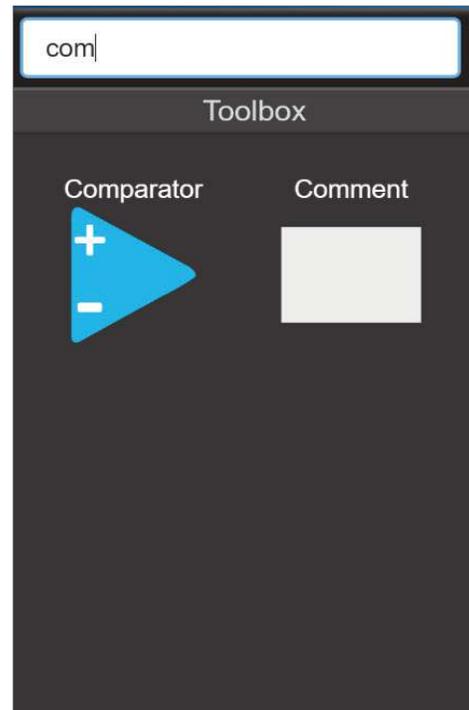
The toolbar holds all the Logic Blocks that can be used to create the program using the the Programming Tool.



### Quick Search

The Quick Search allows to find a logic block quickly.

For example if you are looking for comment box, then you can start typing in the word and this will reduce the numbers of blocks in the tool box to only those blocks whose description matches the search query.



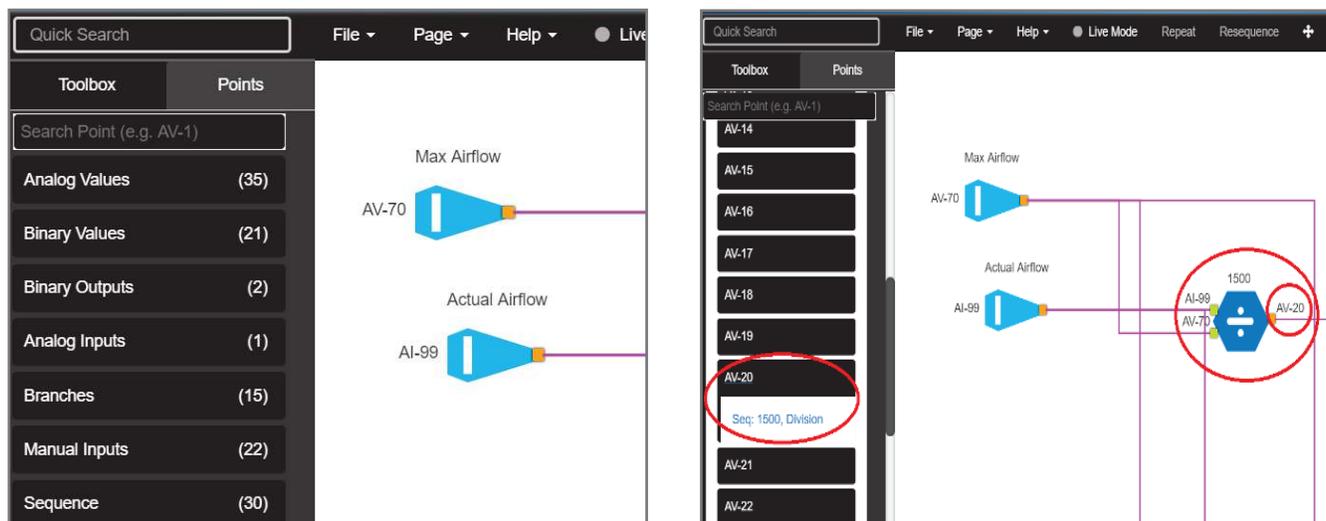
### Logic Blocks Available to Use

- Addition
- And Gate
- Comment
- Comparator Hysteresis
- Delay on Break
- Delay on Make
- Division
- Exclusive OR Gate
- Gated Transfer
- High Low Limiter
- High Selector
- Increment Decrement
- Input
- Latch Gate
- Linear Converter
- Low Selector
- Multiplication
- Or Gate
- Output
- PID Controller
- Runtime
- Single Shot
- Subtraction
- Switch
- Transfer Data
- Two Input Linear Converter
- Within
- Power Of
- Pulse Width Modulation
- Anti-Short Cycle
- Enthalpy
- Floating Motor Controller
- Restrictor
- Change of State Detector
- Velocity to FPM Converter

## 1.7 Points

### Points Tab

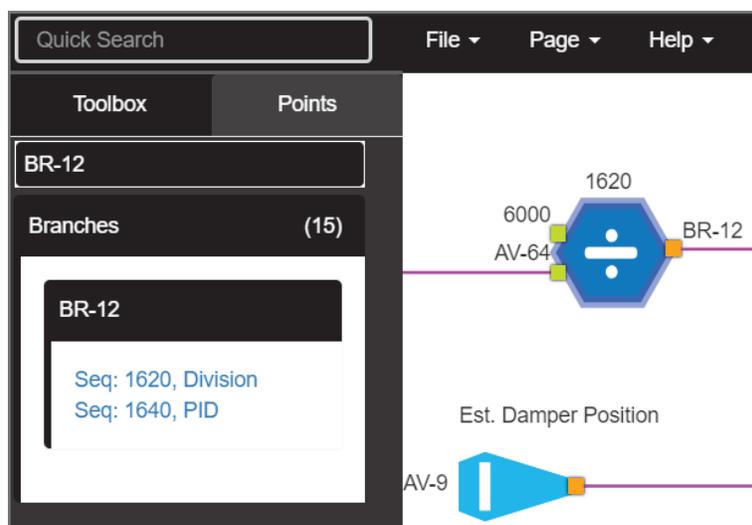
The Points Tab holds all the points that are entered as parameters on any of Logical Blocks.



### Search Point

The Search Point allows to find a point quickly.

For example if you are looking for a Branch Point, then you can start typing in the word in this format.



### Point Type Available to Use

- Analog Inputs (AI)
- Analog Outputs (AO)
- Analog Values (AV)
- Binary Inputs (BI)
- Binary Outputs (BO)
- Binary Values (BV)
- Multistate Input (MSI)
- Multistate Output (MSO)
- Multistate Value (MSV)
- Branches (BR)
- Device Instance (DE)
- Manual Inputs (MAN)
- Sequences (SEQ)

## 1.8 Null Values Interpretations

Block	Input	Null Effect
Addition	Any Input	Null is interpreted as zero value
Division	Any Input	Null is interpreted as zero value
Multiplication	Any Input	Null is interpreted as zero value
Subtraction	Any Input	Null is interpreted as zero value
And Gate	Any Input	Null is interpreted as false (0)
Exclusive Or	Any Input	Null is interpreted as false (0)
Or Gate	Any Input	Null is interpreted as false (0)
Comparator	Plus Input	Null is interpreted as zero value
	Minus Input	Null is interpreted as zero value
	Trigger	Null is interpreted as zero value
	Reset	Null is interpreted as zero value
Delay On Break	Input	Null is interpreted as false (0)
	Delay	Null is interpreted as zero value
	Show Timer	Null is interpreted as zero value
Delay On Make	Input	Null is interpreted as false (0)
	Delay	Null is interpreted as zero value
	Show Timer	Null is interpreted as zero value
Gated Transfer	Gate	Null is interpreted as false (0)
	Input	<b>Null is passed through</b>
High Low Limiter	Input	Null is interpreted as zero value
	High	Null is interpreted as zero value
	Low	Null is interpreted as zero value
Low Selector	Any Input	Null is interpreted as zero value
High Selector	Any Input	Null is interpreted as zero value
Increment / Decrement	Upper Limit	Null is interpreted as zero value
	Plus Input Amount	Null is interpreted as zero value
	Plus Input	Null is interpreted as false (0)
	Minus Input Amount	Null is interpreted as false (0)
	Minus Input	Null is interpreted as zero value
	Lower Limit	Null is interpreted as zero value
Latch Gate	Set	Null is interpreted as false (0)
	Reset	Null is interpreted as false (0)
Linear Converter	Input	Null is interpreted as zero value
	Slope	Null is interpreted as zero value
	Zero	Null is interpreted as zero value
PID	Kp	Null is interpreted as zero value
	Ki	Null is interpreted as zero value
	Kd	Null is interpreted as zero value
	SP	Null is interpreted as zero value
	FB	Null is interpreted as zero value
	iMAX	Null is interpreted as zero value
	iLIM	Null is interpreted as zero value
	STUP	Null is interpreted as zero value
	Reset	Null is interpreted as false (0)
	Runtime	Input
Single Shot	Input	Null is interpreted as false (0)
Switch	Control	Null is interpreted as false (0)
	On	<b>Null is passed through</b>
	Off	<b>Null is passed through</b>
Transfer Data	Input	<b>Null is passed through</b>
Two Point Linear Converter	Input	Null is interpreted as zero value
	Out 1	Null is interpreted as zero value
	Out 2	Null is interpreted as zero value
	In 1	Null is interpreted as zero value
	In 2	Null is interpreted as zero value
Within	High Limit	Null is interpreted as zero value
	Input	Null is interpreted as zero value
	Low Limit	Null is interpreted as zero value

## 1.9 Subroutines

### Subroutines

Subroutines are used to run the same program across multiple device instances. This saves the need for having to repeat the same program over and over. This is achieved by using a function block called a subroutine caller which then references a page with the program you would like to repeat.

### Subroutines Caller

The Subroutine Caller function appears like any other function. It has a sequence number that is referenced for the order in which the program will be read.

When the program reaches the subroutine caller it will then cycle through the program page it is referencing. It will cycle through the subroutine pages program sequence and then return to the program and continue cycling through the rest of the program sequence.

If the next sequence number is another subroutine caller function block, then it will again cycle through the subroutine pages program sequence and then return to the program to continue again.

### Subroutine Caller Function Block Properties

The Subroutines Program need to be placed in a separate program than the subroutine callers. When referencing the 'Program Name' in the subroutine caller the subroutines program name should be referenced.

### Sequence Number

The subroutine function references a sequence number and will be cycled through the program like any other function block. This number must be unique to the program.

### Type

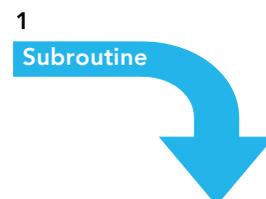
This is function block type. This cannot be modified.

### Name

The name of the subroutine function block can be set. This can be modified. The default name is subroutine. This value is not visible in the page canvas.

### Program Name

This is the program name of the page you want to reference. This must be set.



Properties for Subroutine	
Sequence Number	1
Type	Subroutine
Name	Subroutine
Program Name	Sub-Rtn-1
Page Name	SR-1
SUBST0	3001001:BV-1
SUBST1	3001001:BV-2
SUBST2	3001001:BV-3
SUBST3	3001001:BV-4
SUBST4	3001001:BV-5
SUBST5	3001001:AV-1
SUBST6	3001001:AV-2
SUBST7	3001001:AV-3
SUBST8	3001001:AV-4
SUBST9	3001001:AV-5

## Page Name

This is the page that holds the program that you want the subroutine function block to reference. This must be set.

## Substitution Values

Each subroutine function block can have 32 Substitution values. SUBST0 to SUBST31. Substitution values are values you want to reference in your subroutine program.

## Example

Your subroutine program will reference airflow in multiple devices to perform a calculation.

To do this for all the device numbers that will be read in the subroutine, a substitution value can be used. So for all the subroutine callers that will reference the subroutine program, they will all reference SUBST0 for the airflow, this entered as device number:object, e.g. 3001001:AI-10.

In the subroutine program this will be referenced as SU-0.

Unused substitution values can be left blank.

## Subroutine Program

Subroutine program is created in the same way as a normal program and page and also follows the same rules. Points in the host Optergy or other BACnet devices and objects can be referenced, as can be done with normal program. The only notable difference is the substitution values will also be referenced to take references from devices.

The substitution values are entered into function blocks just like any other objects.

The subroutine caller block can reference 32 substitution values SUBST0 to SUBST31.

Which will be read in the subroutine program as SU-0 to SU-31 respectively.

## 2. Logic Blocks

### 2.1 And Gate

Description: The AND function of two or more binary-type inputs, sets the binary-type output

The output will be set to True (1) only if all inputs are True (1).

If any input is False (0), the output is False (0).



Input 1	Input 2 to 16	Output
False (0)	False (0)	False (0)
True (1)	False (0)	False (0)
False (0)	True (1)	False (0)
True (1)	True (1)	True (1)

## 2.2 Anti Short Cycle

Description: Filters out signals that change states too quickly.

There are 3 inputs: the input signal, minimum ON, and minimum OFF.

**The input signal:** This is the signal that is checked, to ensure that it does not change states too quickly.

**The minimum ON:** This value represents the minimum amount of time that the signal should be ON for.

**The minimum OFF:** This value represents the minimum amount of time that the signal should be OFF for.

There is 1 output.

This output is the signal, after the input signal was held to the minimum ON and OFF times.



## 2.3 Change of State Detector

Description: Detects a change of state.

This block detects if the input value changes from the internal value by more than the deadband. If so, then the output will pulse momentarily.



There are 2 inputs: the input value, and the deadband.

**The input value:** This value represents the value desired to be checked. This value will be stored as an internal value when the output pulses ON.

**The deadband:** This value represents the value that the internal variable has to change by in order to generate the output pulse.

There is 1 output which is the binary output.

The output represents the status of the internal variable in relation to the input value. If the input value is more than the deadband value away from the internal variable, the output will output a short pulse. The value of the internal variable will then be replaced by the value of the input value.

## 2.4 Comment

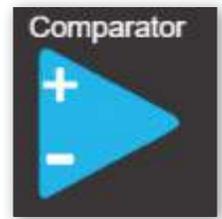
Description: A box that allows you to type text into it describing the program or making general comment.



## 2.5 Comparator Hysteresis

Description: Compare two analog inputs and produce a binary output as a result of the comparison.

The output will be ON when the plus input is greater than or equal to the minus input plus the trigger dead band (TDB).



The output will go False (0) when the plus input is less than or equal to the minus input minus the restore deadband (RDB).

Condition	Output
Plus Input GREATER THAN (Minus Input + TDB)	True (1)
Plus Input LESS THAN (Minus Input - RDB)	False (0)
(Minus Input - RDB) LESS THAN Plus Input LESS THAN (Minus Input + TDB)	No Change

## 2.6 Delay on Break

Description: A binary transition from True (1) to False (0) after the delay period (in seconds) has elapsed.

The "Properties For Count" is simply a feedback from the DOB function block. If an AV is entered in you can display the remaining time as it counts down. This property can be left blank if is not a requirement.



## 2.7 Delay on Make

Description: A binary transition from False (0) to True (1) after the delay period (in seconds) has elapsed.

The "Properties For Count" is simply a feedback from the DOB function block. If an AV is entered in you can display the remaining time as it counts down. This property can be left blank if is not a requirement.



## 2.8 Division

Description: Divide one value by another.



## 2.9 Enthalpy Calculator

Description: Calculates the enthalpy of the system, depending on various inputs, as well as the mode of the block.



There are 2 modes: Basic Mode and Advance Mode.

### Basic Mode

The basic mode will calculate the enthalpy using 2 inputs.

There are 2 inputs: the temperature and the relative humidity.

**The temperature:** The temperature of the area. The units can be chosen to be in Celsius or Fahrenheit.

**The relative humidity:** This represents the relative humidity, as a percentage.

There is 1 output, which is the calculated enthalpy.

### Advanced Mode

There are 3 inputs: the temperature, the altitude, and the relative humidity.

**The temperature:** The temperature of the area. The units can be chosen to be in Celsius or Fahrenheit.

**The altitude:** This represents the altitude, in meters.

**The relative humidity:** This represents the relative humidity, as a percentage.

There are 7 outputs: the enthalpy, the humidity ratio, the atmospheric pressure, the wet bulb temperature, the dew point, the partial pressure at dewpoint, and the partial pressure at saturation.

**The enthalpy:** This is the calculated enthalpy.

**The atmospheric pressure:** This is the calculated atmospheric pressure.

**The wet bulb temperature:** This is the calculated wet bulb temperature.

**The dew point:** This is the calculated dew point.

**The partial pressure at dewpoint:** This is the partial pressure at dewpoint.

**The partial pressure at saturation:** This is the partial pressure at saturation.

The main differences between the basic block and the advanced block are the different numbers of inputs and outputs, and the basic block is assumed to be at altitude of 0.

## 2.10 Exclusive OR Gate

Description: The Exclusive OR function of two binary-type inputs, sets the binary-type output.

The output will be set to True (1) if either input is True (1) and the other input False (0).

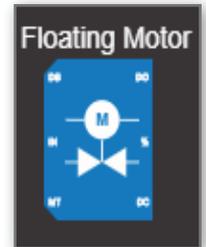
If all inputs are False (0) or all inputs are True (1), the output is False (0).



Input 1	Input 2	Output
False (0)	False (0)	False (0)
True (1)	False (0)	True (1)
False (0)	True (1)	True (1)
True (1)	True (1)	False (0)

## 2.11 Floating Motor Controller

Description: Calculates an output percentage, an open state and a close state based on a deadband, a desired position and a fully open time.



There are 3 inputs: the deadband, the desired position, and the time to fully open.

**The deadband:** This is the deadband within which the open position shall be considered to be finished. It is taken as a percentage.

**The desired input:** This is the desired position, as a percentage.

**The drive time:** This is the time taken to drive the desired position from 0% to 100%. This is read in seconds(s).

There are 3 outputs: the current open position, the opening status, and the closing status.

**The current open position:** This is the position that is currently being outputted. It is written as a percentage.

**The opening status:** This is the opening status, it is true if the output is increasing, and false otherwise.

**The closing status:** This is the closing status, it is true if the output is decreasing, and false otherwise.

**Note:** When the floating motor controller desired input is 0% or 100%, it will over-drive the point by 10%.

## 2.12 Gated Transfer

Description: Write the input value to the output value only when the Gate input is True (1). This function will not write to the output if the Gate input is False (1).

When the gate is closed the last value to be transferred through will be held.



## 2.13 High / Low Limiter

Description: Restrict an analog value to a range defined by two limits. It has three analog inputs and one analog output.



The output will equal the analog input as long as it is within the range defined by the high limit and low limit.

If the analog input exceeds the high limit, the output will be set to the value of the high limit. If the analog input is less than the low limit, the output will be set to the low limit.

If the high limit value is less than the low limit value, the high limit has priority (that is, the output will be set to the high limit, regardless of the analog input value).

## 2.14 High Selector

Description: Select the highest values from up to sixteen inputs. The output will equal the value of the highest input. All inputs must be assigned.



## 2.15 Increment Decrement

Description: Increase the Output value by the Up value or decrease it by a Down value only if a True (1) value is passed to the Up Input or Down Input, respectively, for each cycle of the program.

For each cycle of the program, the value of the Up input value is added to Output whenever the Up Input is True (1).

Similarly, the value of the Down input is subtracted from the Output whenever the Down Input is True (1).

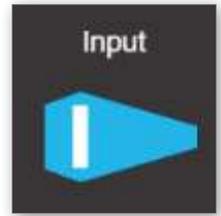
The Output value is limited to the range defined by the Upper Limit and Lower Limit inputs.

**Note:** This function only adds to the Output or subtracts from the Output it when either the Up Input or Down Input is True (1).



## 2.16 Input

Description: A tag that labels an input parameter.



## 2.17 Latch Gate

Description: Two binary inputs, Set (S) and Reset (R), are used to switch the binary output between True (1) and False (0) respectively.



A minimum of 1 program cycle is required on the set (S) input is required to be True (1) and the Reset (R) input is False (0) to set the output to True (1).

The output will stay True (1) once it is set, even if the set (S) input transitions False (0). If the reset (R) input is set to True (1) the output will be set to False (0).

The Reset (R) input has priority over the Set (S) input, so the output will be False (0) if both inputs are True (1).

Set (S)	Reset (R)	Output
False (0)	False (0)	Retains Previous Output
True (1)	False (0)	True (1)
False (0)	True (1)	False (0)
True (1)	True (1)	False (0)

## 2.18 Linear Converter

Description: Perform a linear conversion on the input and produce an output using zero and range values as follows.  $\text{Output} = \text{Zero} + \text{Input} \times \text{Slope}$



## 2.19 Low Selector

Description: Select the lowest values from up to sixteen inputs. The output will equal the value of the lowest input. All inputs must be assigned.



## 2.20 Multiplication

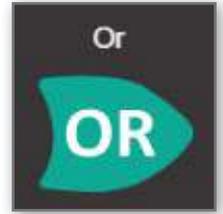
Description: Multiply one value by another.



## 2.21 OR Gate

Description: The OR function of two or more binary-type inputs, sets the binary-type output.

The output will be set to True (1) if any inputs are True (1). If all input are False (0), the output is False (0).



<b>Input 1</b>	<b>Input 2 to 16</b>	<b>Output</b>
False (0)	False (0)	False (0)
True (1)	False (0)	True (1)
False (0)	True (1)	True (1)
True (1)	True (1)	True (1)

## 2.22 Output

Description: A tag that labels an output parameter.





## 2.23 PID Controller

Description: The PID provides a proportional integral and derivative (PID) control. The output is gradually iterated in an attempt to get the feedback input (FB) to match the setpoint (SP).

The output of the PID controller is the sum of the following components: proportional (P), integral (I), derivative (D), and a constant of 50.

The output of the PID Block is limited between 0 and 100.

$$\text{Output} = 50 + P + I + D$$

The primary inputs are the setpoint (SP) and feedback input (FB).

The other inputs are used as tuning inputs, and they are normally set to fixed values as needed to achieve responsive and stable control.

### Error

This difference is called the error (E), and it is an intermediate variable representing how far away from setpoint (SP) the feedback input (FB) is.

$$E = \text{FB} - \text{SP}$$

### Proportional Constant (Kp)

The proportional component (P) responds immediately and is proportional to the value of the error.

The (Kp) input will directly affect the proportional output. The proportional component is equal to the error times the proportional constant.

$$P = E \times K_p$$

### Integral Constant (Ki)

The integral component (I) is calculated as a running total over time. It generates incremental adjustments over time in order to have the feedback input match the setpoint.

The initial output of the integral component is set to the value of integral start up (STUP). From then on, the integral component is a running total, with an integral adjustment value added every second.

The integral adjustment value is the error times the integral constant (Ki) divided by 60.

The integral constant is divided by 60 since it represents the change per minute, and the calculation is performed every second.

$$I = I_{\text{previous}} + (E \times K_i / 60)$$

The rate of change of the integral component is limited by the maximum integral change (Imax), which is the maximum amount the integral component is permitted to change per minute.

The maximum permitted integral adjustment (which is added once per second) is then  $I_{\text{max}}/60$ .

Finally, the value of the integral component is limited by the integral limit (limit). I is not permitted to be greater than limit or less than -limit.  $I \leq \text{limit}$  and  $I \geq -\text{limit}$ .

### Derivative Constant (Kd)

The derivative component (D) is used mainly in integrating processes for faster response but may be prone to overshoot.

$D = K_d * (E - E_{prev})$ , where  $E_{prev}$  represents the value of E from the previous logic cycle.

For PI control without the D component,  $K_d$  may be left as 0.

### Integral start up (STUP)

This constant is used to initialize the value of the integral component upon startup. The integral component is also set to this value whenever  $K_i$  is set to 0.

### Reset

Reset In order to reset the output at start up, a True (1) value can be written to the Reset input, this will reset the output to the start up (STUP) value.

The Reset input must transition back to False (0) before it can be triggered again.

## 2.24 Power Of

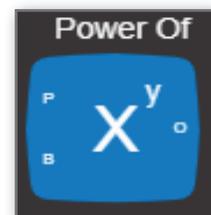
Description: Outputs the value of the base input to the power of the power input.

There are 2 inputs: the power and the base.

**The power:** This is the power that the base is to be raised by.

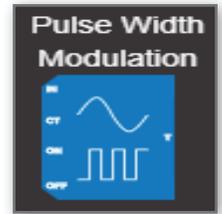
**The base:** This is the base that will be raised.

There is 1 output, which is the calculated output. This is the calculated output of all the inputs.



## 2.25 Pulse Width Modulation

Description: Converts an analog signal as a percentage into a digital signal, based on the cycle time.



There are 4 inputs: the input signal, the cycle time, the minimum ON time and the minimum OFF time.

**The input signal:** This value (between 0-100) represents the percentage of the cycle time that the signal is ON.

**The cycle time:** This represents the period of the digital output, in milliseconds.

**The minimum ON time:** This represents the minimum time that the signal must be ON. If this value is larger than the calculated ON time, it will override the calculated value.

**The minimum OFF time:** This represents the minimum time that the signal must be OFF. If this value is larger than the calculated OFF time, it will override the calculated value.

There is 1 output, which is the converted digital output. This binary output pulses ON for a calculated amount of time, and OFF for a calculated amount of time.

## 2.26 Restrictor

Description: Restrict the rate of change of an analog value.

There are 4 inputs: the input, the reset, the maximum up and the maximum down.

**The input:** This is the signal to be slowed down.

**The reset:** This signal controls whether the output is reset.

**The maximum up:** This is the maximum amount per second that the output signal will increase by until it reaches the input value.

**The maximum down:** This is the maximum amount per second that the output signal will decrease by until it reaches the input value.

The output will be immediately set to zero when the reset input is FALSE (0).

There is 1 output, the output signal. This represents the output of the block, as it increases or decreases towards the input value.



## 2.27 Runtime

Description: Outputs a cumulative analog run time (in hours) for the period the binary input is True (1).



## 2.28 Single Shot

Description: When the input is True (1) the output will be true for one program cycle and then revert to False (0).

The input must become False (0) in order to repeat the action once it becomes True (1).



## 2.29 Subtraction

Description: Subtract one input value from another.



## 2.30 Switch

Description: Select an output value from two input values, depending on the value of a binary input.

It has two analog inputs, one binary input and an analog output.

The output equals the ON analog input if the binary control input is True (1), and the output equals the OFF analog input if the binary control input is False (0).



## 2.31 Transfer Data

Description: Copy a value from one property to another. This is used to transfer analog or binary data from one property to another without changing the data content.



## 2.32 Two Input Linear Converter

Description: Perform a linear conversion on the input to produce an output. Two points (IN1, OUT1) and (IN2, OUT2) are used to define the line relating input to output.



A straight-line relationship (of the form  $y = a + bx$ ) determines the output as a function of the input.

By inputting two points on this line, the line is then defined. When input = IN1, output = OUT1, and when input = IN2, output = OUT2.

*Example:*

If you wanted to use this function to convert F° to C°, you could use the following values for IN1, OUT1, IN2 and OUT2:

IN1=32

OUT1=0

IN2=212

OUT2=100

## 2.33 Velocity Pressure To FPM Converter

Description: Performs an operation to convert an input that represents velocity pressure into an output that represents velocity.



There are 3 inputs: the input signal, the k factor, and the zero.

**The input signal:** This is the signal to be converted.

**The k factor:** This is the represents the multiplier.

**The zero:** This represents an error correction value.

There is 1 output, which is the velocity. This is the calculated output of all the inputs.

## 2.34 Within

Description: Turn the output from False (0) to True (1) whenever the Input Value is within the range defined by the Limit 1 and Limit 2 values.



This has three analog inputs and one binary output.

The output is True (1) whenever the input value is greater than or equal to the lower limit value and the input value is less than or equal to the higher limit value.

Otherwise, the output is False (0).

**Note:** The output is True (1) if the Input Value is equal to either of the limits.

# 3. Device Configuration

The screenshot shows the Optergy web interface for configuring device instance 864001. The page is titled "Device Configuration" and includes a navigation bar with "VAV Operations", "Inputs/Outputs Live Data", and "Programming". The main content area is divided into five numbered sections:

- Section 1: Device Configuration** includes fields for Software Version (1.00), Physical MAC Address (02:12:c2:15:15:56), Device Location, Device Description, and Units (US Customary Units).
- Section 2: IP Configuration** includes Ethernet Modes (192.168.1.MAC), IP Address (192.168.1.1), Subnet Mask (255.255.255.0), Default Gateway (192.168.1.1), BACnet/IP Network Number (1), and BACnet IP Port (UDP) (47808).
- Section 3: BACnet Configuration** includes Device Instance (864001) and BACnet Mode (IP Only).
- Section 4: Expansion RS-485 Port (routing network)** includes Port Protocol (864).
- Section 5: Foreign Device Registration** includes a dropdown menu (Disable), BACnet Connection Status (Disconnected), BACnet IP Address (192.168.1.10), BACnet Port Number (47808), and Registration Lifetime (seconds) (300). A "Test Registration" button is also present.

At the bottom of the page, there are "Save Changes" and "Enable Editing" buttons.

## Section 1: Device Configuration

Details are BACnet properties and can be used to describe the device and its location as well as setting the units Metric or US Customary ([Metric is default state](#)), This mode can also be set via BACnet object: BV-20010 (Inactive=metric, Active=US Customary Units)

## Section 2: IP Configuration

Used for web user interface connectivity and BACnet/Modbus connectivity. The default state is 1.MAC (e.g. 192.168.1.10 assume switch position is set to 1-0), other choices include DHCP, where the device gets an IP address from the DHCP server, or Manual where the user chooses the IP address, subnet mask and gateway. IP devices on the same BACnet internetwork must use the same BACnet IP network number. UDP port uses the BACnet [Default UDP:47808](#) (BAC0), this can be changed if needed.

## Section 3: BACnet Configuration

This is where the user can setup the BACnet device instance. This value can be set to:

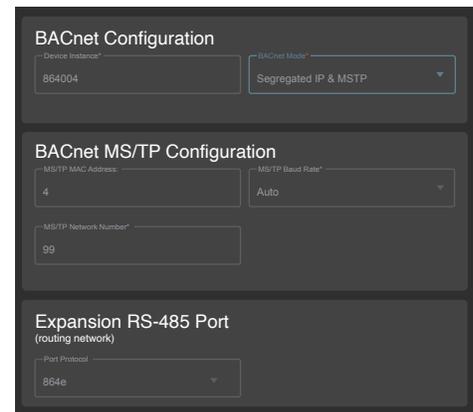
- 1.MAC - This means the MAC address will preset the Device instance 864010 (assumes MAC switch is set to 1-0) ([default is 1.MAC](#)).
- Web User Interface - Value is written from the user interface.
- Using Proton or Optergy Enterprise BACnet Device Table which can write one or many P864 devices.

**BACnet Mode** - Sets the communication method. Choices include:

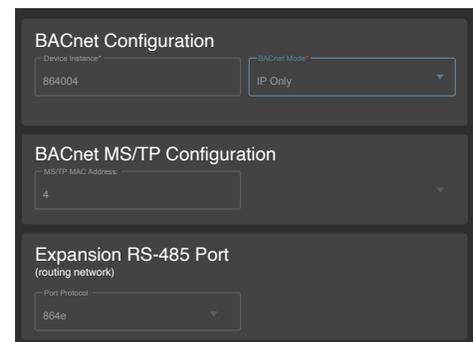
- IP only (web user interface connection via IP, BACnet IP)
- MS/TP only (web user interface connection via IP, BACnet MS/TP)
- Segregated mode (web user interface connection via IP, BACnet IP enabled and BACnet MS/TP enabled). This is intended to ease startup so the device will scan in regardless of how the user connects to it (default is segregated mode).

Note: Choosing the mode of operation changes the menu choices.

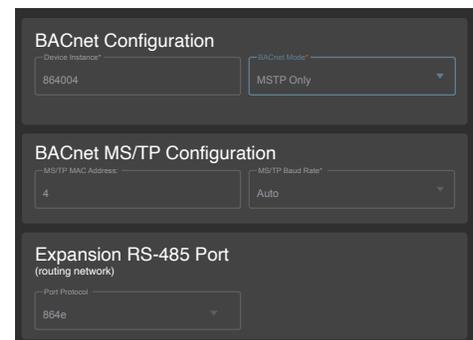
**Segregated Mode:** (BACnet IP and BACnet MS/TP are both enabled) the P864 is set to Autobaud on the MS/TP) note the MS/TP network number must be the same as the router that the P864 is connected to. In this scenario you can use IP or use BACnet MS/TP (not intended for both at the same time. EXP2 Router/Gateway port is disabled.



**IP Only:** In this mode the device uses only TCP/IP mode and now the router port (Data 2 [R] previously called EXP 2) is enabled to use 864e, BACnet MS/TP or the Modbus Gateway.



**BACnet MS/TP Only:** In this mode the P864 is communicating to a router on BACnet MS/TP, and the router port is open to bring BACnet MS/TP devices on the EXP2 (Expansion RS-485 setup for BACnet MS/TP). Autobaud is only available in MS/TP only mode. This syncs up the baud rate automatically (within a few minutes).



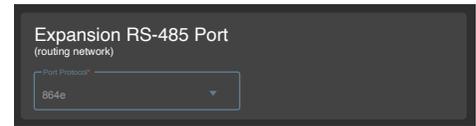
### BACnet MS/TP Configuration

Sets the MAC address on the MS/TP it cannot be overwritten. Physical switch on the P864 sets this.

## Section 4: Expansion RS-485 Port (Data 2 [R] previously called EXP 2)

Options include:

- 864e (Expansion IO module communication) (default is 864e mode)

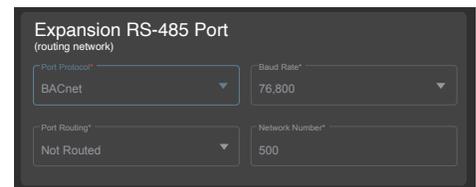


- BACnet MS/TP (Router Mode)

Baud Rate: 9.6k, 19.2k, 38.4k, 76.8k (default), 115.2k

Port Routing: This option lets the user route the MS/TP traffic in Routable mode on EXP2 to any client or Non-Routable (default) limit to the device for use in the local P864 (eg. Sensor network)

Network Number: Is the unique BACnet network number that needs to be setup (500 is default)



- Modbus Gateway

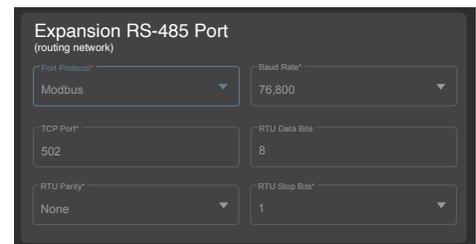
Baud Rate: 9.6k, 19.2k, 38.4k, 76.8k, 115.2k

TCP IP port: 502 (default) is the Modbus standard port

Data bits: 8 (default)

Parity: None (default)

Stop bits: 1 (default)



## Section 5: Foreign Device Registration

On occasion the user may find it necessary to cross subnets to connect to a BACnet network. If there is not BACnet BBMD on the subnet, the device will be unable to join the remote network. BACnet has a mechanism called Foreign Device Registration, this allows a device to register with a BBMD and re-subscribe on a regular interval. This will connect the device to the BACNET IP network. The BBMD must support incoming Foreign device registrations. (Default mode is disabled).

**Enable/Disable:** Turns on and off the subscription to the device.

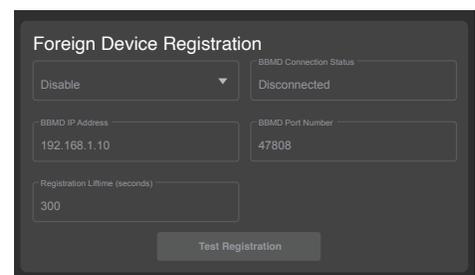
**BBMD IP address:** Is the IP Address of the BBMD that accepts incoming foreign device registration requests.

**BBMD Connection Status:** Shows connected or disconnected after the user tests the registration.

**BBMD Port number:** BACnet reserves UDP 47808 (BAC0).

**Registration interval:** How often does the device request to join (default is every 300 seconds).

**Test Registration:** Performs immediate connection test and validates.



To use this feature, fill out all the fields and test the connection assuming it is successful then press Save Changes.

# 4. System Settings

Users can create a separate user profile for each user type. At a minimum there must be 1 admin user, this is setup as soon as you login.

To create users, navigate to Date: 6/30/20, 1:15PM admin

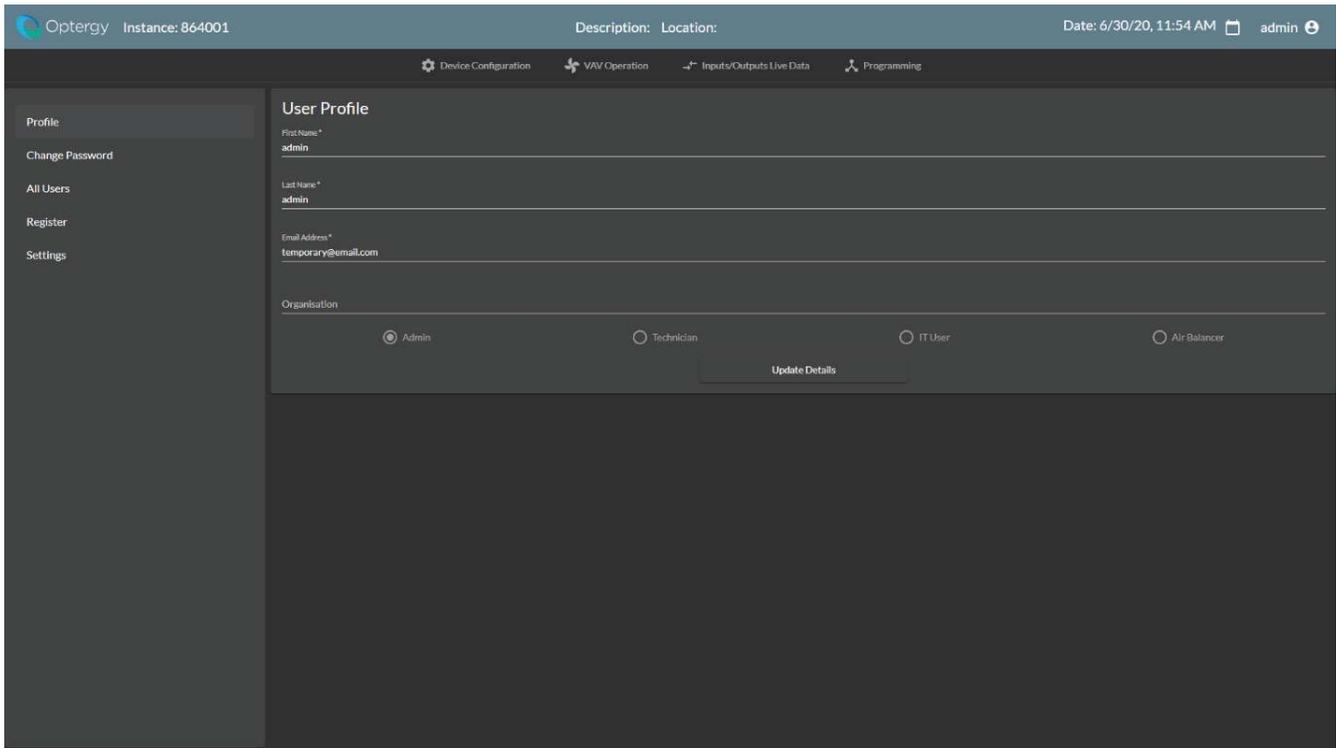
Click  to reveal the settings menu.

There are 4 user types:

- **admin:** Can see and do all features.
- **Technician:** Can see and do all features (adding users).
- **IT User:** Can only see device configuration/networking.
- **Air Balancer:** Can only see the VAV page.

Note: The first admin user is always called 'admin' (case sensitive) and uses the password that the user creates.

Users can self-administer their login.



Optergy Instance: 864001 Description: Location: Date: 6/30/20, 11:54 AM admin

Device Configuration VAV Operation Inputs/Outputs Live Data Programming

Profile  
Change Password  
All Users  
Register  
Settings

**User Profile**

First Name\*  
admin

Last Name\*  
admin

Email Address\*  
temporary@email.com

Organisation

Admin  Technician  IT User  Air Balancer

Update Details

Change a password.

The screenshot shows the Optergy web interface for instance 864001. The top navigation bar includes the instance name, description, location, date (6/30/20, 11:56 AM), and the user 'admin'. Below the navigation bar, there are tabs for 'Device Configuration', 'VWV Operation', 'Inputs/Outputs Live Data', and 'Programming'. On the left sidebar, the 'Change Password' option is selected. The main content area is titled 'Update Password' and contains three input fields: 'Previous Password \*', 'New Password \*', and 'Password Check'. A 'Reset Password' button is located at the bottom of the form.

View all users on the device.

The screenshot shows the Optergy web interface for instance 864001. The top navigation bar includes the instance name, description, location, date (6/30/20, 11:56 AM), and the user 'admin'. Below the navigation bar, there are tabs for 'Device Configuration', 'VWV Operation', 'Inputs/Outputs Live Data', and 'Programming'. On the left sidebar, the 'All Users' option is selected. The main content area is titled 'Select a user' and contains a form with the following fields: 'First Name', 'Last Name', 'Email', 'Organisation', and 'Role'. There are 'Save' and 'Cancel' buttons at the bottom of the form. On the right side of the main content area, there is a preview of the selected user: 'User: admin', 'Name: admin/admin', and 'Role: Admin'.

Register new users.

The screenshot shows the 'User Registration' form in the Optergy web interface. The interface includes a top navigation bar with 'Instance: 864001', 'Description: Location:', and 'Date: 6/30/20, 11:55 AM' along with a user profile icon for 'admin'. Below the navigation bar are tabs for 'Device Configuration', 'VAV Operation', 'Inputs/Outputs Live Data', and 'Programming'. A left sidebar contains menu items: 'Profile', 'Change Password', 'All Users', 'Register', and 'Settings'. The main content area is titled 'User Registration' and contains the following fields: 'Username \*', 'First Name \*', 'Last Name \*', 'Email Address \*', 'Organisation', and 'Password \*'. Below the password field is a 'Confirm Password' field. At the bottom of the form, there are four radio button options: 'Admin', 'Technician', 'IT User', and 'Air Balancer'. A 'Submit' button is located at the bottom center of the form.

System Settings allow for:

**Firmware Update** Press the paperclip to locate filename.upd and press illuminated upload button

Note: Firmware can be upgraded or downgraded.

**Factory Reset** changes all settings to out of the box defaults. Pressing this deletes all programs and settings including network settings and resets to 192.168.1.MAC and user: admin/password: admin

**Reboot Controller** simply restarts the controller.

The screenshot shows the 'System Configuration' page in the Optergy web interface. The top navigation bar and sidebar are identical to the previous screenshot. The main content area is titled 'System Configuration' and contains three sections: 'Firmware Update', 'Factory Reset', and 'Reboot Controller'. The 'Firmware Update' section includes a text prompt 'Please select new UPD file to update from [1, 2, 9]', a file selection icon, and two radio buttons: 'Software File Selected' (selected) and 'No update selected'. An 'Update' button is located to the right of the radio buttons. The 'Factory Reset' section includes a warning text 'All Database and Network Settings will revert to out of the box default' and a red 'Reset' button. The 'Reboot Controller' section includes a warning text 'This will reboot the controller. The back-end and webserver will become unresponsive.' and a red 'Reboot Controller' button. At the bottom of the page, there is a copyright notice: '© 2019 Optergy Pty Ltd.'

# 5. Inputs/Outputs Live Data

Setting the values here will set the BACnet attributes like description, present value, priority array and so on. The user can set input types using the drop-down list.

## Universal Inputs supported include:

- 10 K-Ohm Thermistor NTC type 2 {P864 and 864e}
- 3K-Ohm Thermistor NTC {P864 and 864e}
- 0-10 VDC {P864 and 864e}
- 4-20 mA {P864 and 864e}
- Dry contact {P864 and 864e}
- Counts {P864 and 864e}
- Pulse (0-100 Hz) {P864 and 864e}

## Binary Outputs supported include:

- 24 VDC (MOSFET) {P864}
- Normally Open Contact/Normally Closed Contact {864e}
- Relay common can be dry, ground or 24VDC/VAC (note all relay common tied together)

## Analog Outputs supported include:

- 0-10 VDC {P864 and 864e}
- 4-20 mA {P864 and 864e}

## Notes:

1. Universal Input resolution on P864 is 24 bits, and 864e is 12 bits.
2. Binary Outputs can be overridden from the web user interface, doing so will override them at Priority 8.
  - a. Choices include ON/OFF/AUTO (where AUTO sets the Priority 8 to NULL).
3. Analog Outputs can be overridden by typing in a percentage 0-100%, and then clicking away from the field.
4. Units can be picked for items that have them using the drop-down list.
5. Descriptors can be added and write to the BACnet description property.
6. Scaling is used for 0-10V or 4-20 mA to say equal 20 to 100%.
7. Pulse value sets the pulse amount to say 1 pulse = 10 liters.

### Inputs and Output page (P864 shown)

When a P864 is connected to an 864e (2 connected in this example), the active device setting is green colored and the can tab over the next device.

### Inputs and Output page (P864, 864e-1 and 864e-2 shown)

# 6. Airflow Sensing

The P864 can be equipped with an airflow sensor that measure pressure (0-500 Pascal or 0 to 2 Inches of water column), used in conjunction with a flow sensor this can be used to derive Airflow (in Liters per Second or Cubic Feet per Minute). The AFS-864 is a calibrated sensor module that has 4 pins that connect to a connector on the P864. The device is factory calibrated and outputs a communication stream to the P864 which then populates BACnet points that can be used in programming and in the user interface. Below you will find relevant data points that you can design your Program logic around.

To use the airflow sensor, it must be connected to the lower board of a P864 (with the power disconnected). On power up the P864 will automatically detect the presence of this AFS-864 and the relevant BACnet points shall be available to use.

To use the airflow sensor, it must be connected to the lower board of a P864 (with the power disconnected). On power up the P864 will automatically detect the presence of this AFS-864 and the relevant BACnet points shall be available to use.

## Detect the sensor (done automatically by P864)

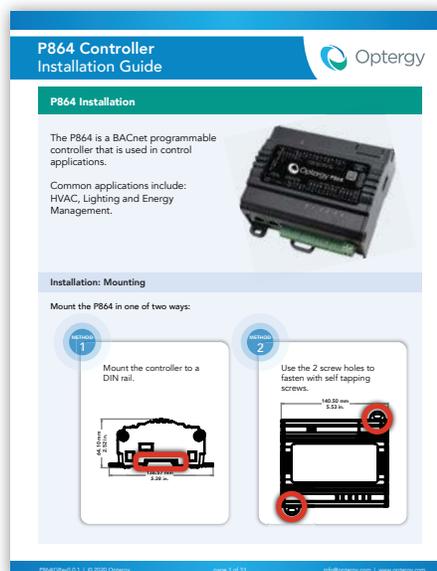
- BI-20000
- **Description:** Airflow Sensor Detected
- **Present Value:** Active (Sensor is detected, Inactive is sensor not detected)

## Read pressure and airflow values:

BACnet data points for velocity/airflow/pressure include:

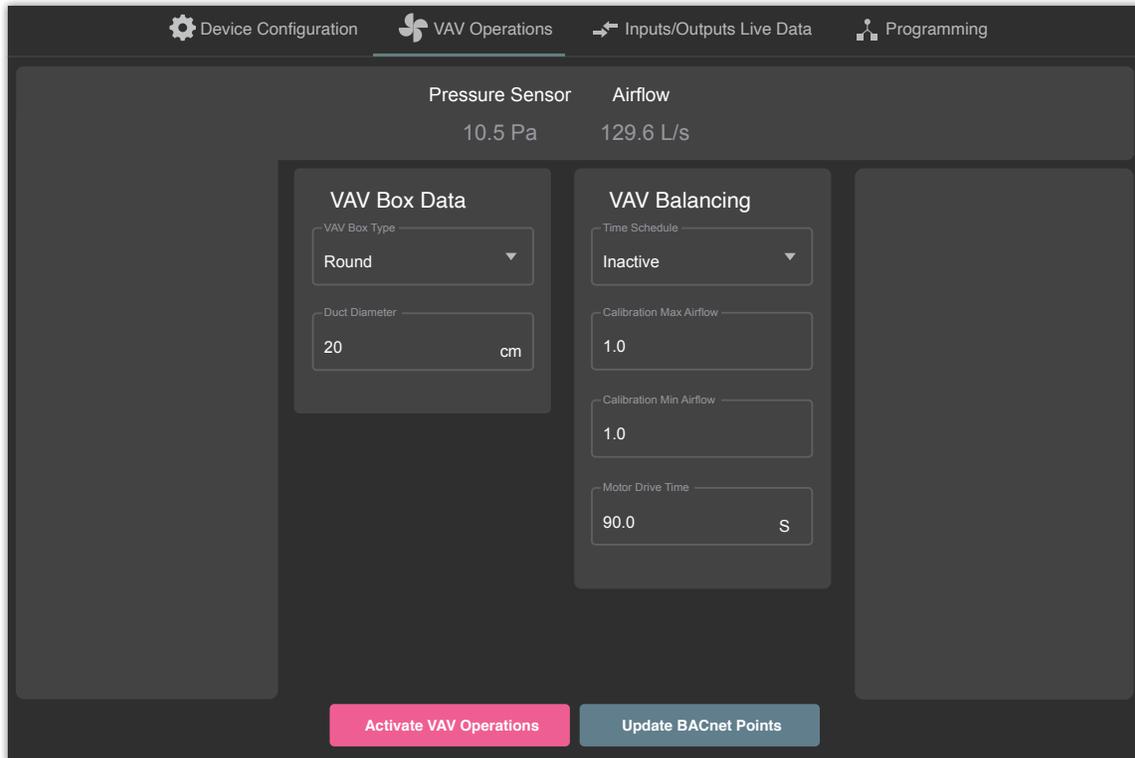
- AI-97 Velocity
  - o Present value <number>
  - o units: ft/min or m/s
- AI-98 Sensor DP
  - o Present value <number>
  - o units: in. H2O or Pa
- AI-99 Airflow
  - o Present value <number>
  - o units: ft<sup>3</sup>/min or L/s

The Optergy VAV program that is built in uses these points and can be seen in more detail here:

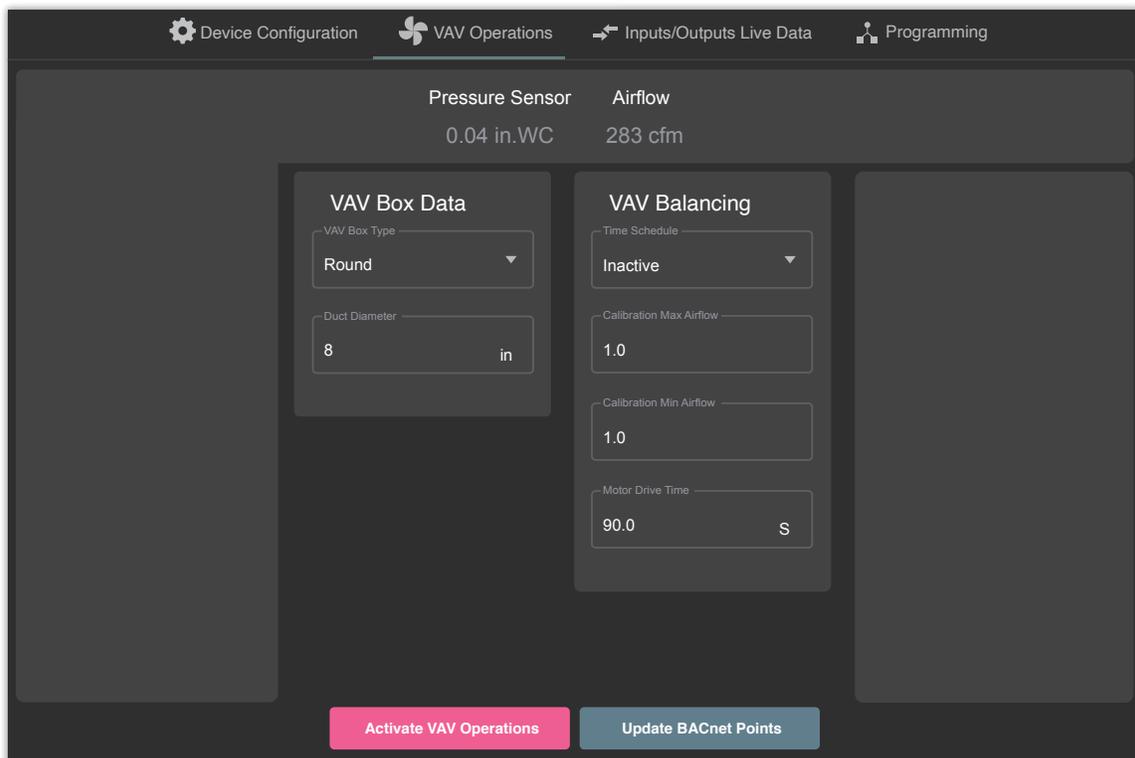


[View the P864 Installation Guide >](#)

**Metric Units (Pressure: Pascals or Pa, Airflow: Liters per second of L/s)**



**US Customary Units (Pressure: Inches of water column or " w.c. , Airflow: Cubic feet per minute or CFM)**



**Velocity and airflow calculation:**

Velocity is feet per minute or {fpm}, or meters per second {m/s}

Pressure is in inches of water column {in. H<sub>2</sub>O or "w.c.}, or Pascals {Pa}

**Basic formulae used in airflow calculation:***Converting Velocity Pressure to fpm*

For standard air density (.075 lb/ft<sup>3</sup> or 1.20 kgs/m<sup>3</sup>), use formula:

$$\text{fpm} = 4005 \times \sqrt{VP} \qquad \text{m/s} = 1.29 \times \sqrt{VP}$$

For non-standard air density use formula:

$$\text{fpm} = 1096 \times \sqrt{VP/\text{Density(in.Hg)}} \qquad \text{m/s} = 1.412 \times \sqrt{VP/\text{Density(kPa)}}$$

Where:

VP = in. H<sub>2</sub>O or Pa

*Air Flow Formulas*

To find the volume of air flowing through ductwork, inlets, outlets, hoods, etc., use the following formula:

Basic formula:  $Q = AV$

Velocity unknown:  $V = Q/A$

Area unknown:  $A = Q/V$

**Area calculation:**

Where the area of a circle is:  $A = \pi r^2$  or  $3.1416 (\text{Pi}) * r^2$  (radius: inch or cm)

- US units:  $3.1416 (\text{Pi}) * r^2 (\text{inch}) / 144 = \text{Square feet}$
- Metric units:  $3.1416 (\text{Pi}) * r^2 (\text{cm}) / 10,000 = \text{Square meters}$

And

Area of rectangle is:  $A = \text{Length} * \text{Width}$

- US units:  $A = L (\text{inch}) * W (\text{inch}) / 144 = \text{Square feet}$
- Metric Units =  $A = L (\text{cm}) * W (\text{cm}) / 10000 = \text{Square meters}$

**Duct type:**

- BV-5001 Duct type
- Present Value: Active = Round, Inactive = Rectangle
  - o Description: VAV Box Type (0 = Round, 1 = Rectangle)

**Round duct:**

- AV-5027 Duct Diameter
  - o Present Value: <number>
  - o Description: VAV Duct Diameter
  - o Units: inch or cm

**Rectangular ducts:**

- AV-5025 Duct Height
  - Present Value: <number>
  - Description: VAV Duct Height (Inch or cm)
- AV-5026 Duct Width
  - Present Value: <number>
  - Description: VAV Duct Width (Inch or cm)

**Calibration Factor**

When airflow is measured in the field its not uncommon to have some losses due to instrument accuracy, leaks or other factors that affect the actual airflow that is measured by the test and balance contractor. They get corrected using a K factor or calibration factor as it may be called. This value is a constant and is simply a multiplier against the current P864 calculated value.

Optergy has a calibration minimum factor which adjusts the minimum airflow and offsets it. There is a maximum calibration factor this adjusts the airflow reading.

Let's use an example.

P864 reads 100 to 300 L/s from its calculation

Test and Balance (TAB) measures 120 and 320 L/s. the TAB contractor then adjusts the K factor to:

- Calibration minimum airflow= $1.20$  (320 L/s divided by 300 L/s)
- Calibration maximum airflow= $1.07$  (120 L/s divided by 100 L/s)

# 7. Optergy Air Wireless Sensor

When used together with the Optergy P864 the Optergy Air wireless sensor can measure and communicate an environments metrics.

For further installation information refer to the Optergy Air wireless sensor & transceiver installation guides.



## P864 BACnet Points for the Optergy Air Wireless Sensor & Transceiver

Optergy Air Sensor 1		
Sensor Description	BACnet Object	Units (auto assigned based on the Optergy P864 configuration)
Temperature	AI-101	°C or °F
Humidity Level	AI-102	%rH
Sensor Online Status	BI-20031	Shows whether this sensor is detected (This BI's description will show the version number and the sensor's serial number)
Low Battery Level Warning	BI-102	No units
Sensor Locating LED	BO-101	No units

Optergy Air Transceiver		
Description	BACnet Object	Units (auto assigned based on the Optergy P864 configuration)
Transceiver Detected	BI-20008	Shows whether the transceiver is detected (This BI's description will show the version number)
Sensor Slot Paired Status	BI-20021	Shows whether this sensor slot is paired to a sensor
Enable Pairing	BO-102	No units

# Appendix

## P864 Diagnostic Points

Object Type	Object Instance	Object Property	Object Name	Description	Present Value	Notes
Analog Input	97	Present Value	Airflow Sensor Air Velocity Reading	Airflow Sensor Air Velocity Reading	0	Airflow in m/s or ft/min
Analog Input	98	Present Value	Airflow Sensor DP Reading	Airflow Sensor DP Reading	0	Pressure in Pa or inch w.c.
Analog Input	99	Present Value	Airflow Sensor Reading	Airflow Sensor Reading	0	Airflow in l/s or ft/min
Analog Input	20000	Present Value	Software uptime	Software uptime	170.18	Hours
Analog Input	20001	Present Value	Last boot timestamp	Last boot timestamp	1,606,135,050,000	Epoch time stamp
Analog Input	20002	Present Value	Reboot count	Reboot count	1	Hardboot
Analog Input	20003	Present Value	Software restart count	Software restart count	0	Softboot
Analog Input	20004	Description	MSTP Active MAC Addresses	MSTP Active MAC Addresses	0	Description will show any connected ms/tp to data 2 port
Analog Input	20005	Present Value	System CPU Usage	System CPU Usage	3.9	Percent
Analog Input	20006	Present Value	System Free Memory (MB)	System Free Memory (MB)	213	Megabytes
Analog Input	20007	Description	Physical MAC Address	02:66:89:aa:9d:f1	0	Ethernet MAC address
Analog Input	20010	Present Value	AI Objects count	AI Objects count	51	unitless
Analog Input	20011	Present Value	AV Objects count	AV Objects count	185	unitless
Analog Input	20012	Present Value	BI Objects count	BI Objects count	35	unitless
Analog Input	20013	Present Value	BV Objects count	BV Objects count	111	unitless
Analog Input	20014	Present Value	Notification Class Objects count	Notification Class Objects count	0	unitless
Analog Input	20015	Present Value	Trend Log Objects count	Trend Log Objects count	0	unitless
Analog Input	20016	Present Value	Schedule Objects count	Schedule Objects count	0	unitless
Analog Input	20017	Present Value	Calendar Objects count	Calendar Objects count	0	unitless
Analog Input	20018	Present Value	Alarm Objects count	Alarm Objects count	0	unitless
Analog Input	20020	Present Value	Hardware Version	Hardware Version	(Supported in later hardware version)	unitless

# P864 Diagnostic Points continued.

Object Type	Object Instance	Object Property	Object Name	Description	Present Value	Notes
Analog Input	20021	Present Value	24V Output Voltage Level	24V Output Voltage Level	(Supported in later hardware version)	VDC
Analog Input	20022	Present Value	Power Supply Voltage Level	Power Supply Voltage Level	(Supported in later hardware version)	VDC
Analog Input	20023	Present Value	RTC Battery Voltage Level	RTC Battery Voltage Level	(Supported in later hardware version)	VDC
Analog Input	20024	Present Value	MAC Address Switches Current Value	MAC Address Switches Current Value	77	Set via rotary switches on P864 circuit board
Analog Input	20030	Present Value	DDC Cycle time	DDC Cycle time	0	ms
Analog Input	20031	Description	DDC Problems	<Problems shown here>	0 (default)	description denotes error, if blank no problems
Analog Input	26001	Present Value	AI-1 Calibration Offset	AI-1 Calibration Offset	0 (default)	unitless, can be used to fix an offset to a value read
Analog Input	26002	Present Value	AI-2 Calibration Offset	AI-2 Calibration Offset	0 (default)	unitless, can be used to fix an offset to a value read
Analog Input	26003	Present Value	AI-3 Calibration Offset	AI-3 Calibration Offset	0 (default)	unitless, can be used to fix an offset to a value read
Analog Input	26004	Present Value	AI-4 Calibration Offset	AI-4 Calibration Offset	0 (default)	unitless, can be used to fix an offset to a value read
Analog Input	26005	Present Value	AI-5 Calibration Offset	AI-5 Calibration Offset	0 (default)	unitless, can be used to fix an offset to a value read
Analog Input	26006	Present Value	AI-6 Calibration Offset	AI-6 Calibration Offset	0 (default)	unitless, can be used to fix an offset to a value read
Analog Input	26007	Present Value	AI-7 Calibration Offset	AI-7 Calibration Offset	0 (default)	unitless, can be used to fix an offset to a value read
Analog Input	26008	Present Value	AI-8 Calibration Offset	AI-8 Calibration Offset	0 (default)	unitless, can be used to fix an offset to a value read
Analog Input	26011	Present Value	AI-11 Calibration Offset	AI-11 Calibration Offset	0 (default)	unitless, can be used to fix an offset to a value read
Analog Input	26012	Present Value	AI-12 Calibration Offset	AI-12 Calibration Offset	0 (default)	unitless, can be used to fix an offset to a value read
Analog Input	26013	Present Value	AI-13 Calibration Offset	AI-13 Calibration Offset	0 (default)	unitless, can be used to fix an offset to a value read
Analog Input	26014	Present Value	AI-14 Calibration Offset	AI-14 Calibration Offset	0 (default)	unitless, can be used to fix an offset to a value read
Analog Input	26015	Present Value	AI-15 Calibration Offset	AI-15 Calibration Offset	0 (default)	unitless, can be used to fix an offset to a value read
Analog Input	26016	Present Value	AI-16 Calibration Offset	AI-16 Calibration Offset	0 (default)	unitless, can be used to fix an offset to a value read
Analog Input	26017	Present Value	AI-17 Calibration Offset	AI-17 Calibration Offset	0 (default)	unitless, can be used to fix an offset to a value read
Analog Input	26018	Present Value	AI-18 Calibration Offset	AI-18 Calibration Offset	0 (default)	unitless, can be used to fix an offset to a value read
Analog Input	26021	Present Value	AI-21 Calibration Offset	AI-21 Calibration Offset	0 (default)	unitless, can be used to fix an offset to a value read
Analog Input	26022	Present Value	AI-22 Calibration Offset	AI-22 Calibration Offset	0 (default)	unitless, can be used to fix an offset to a value read

# P864 Diagnostic Points continued.

Object Type	Object Instance	Object Property	Object Name	Description	Present Value	Notes
Analog Input	26023	Present Value	AI-23 Calibration Offset	AI-23 Calibration Offset	0 (default)	unitless, can be used to fix an offset to a value read
Analog Input	26024	Present Value	AI-24 Calibration Offset	AI-24 Calibration Offset	0 (default)	unitless, can be used to fix an offset to a value read
Analog Input	26025	Present Value	AI-25 Calibration Offset	AI-25 Calibration Offset	0 (default)	unitless, can be used to fix an offset to a value read
Analog Input	26026	Present Value	AI-26 Calibration Offset	AI-26 Calibration Offset	0 (default)	unitless, can be used to fix an offset to a value read
Analog Input	26027	Present Value	AI-27 Calibration Offset	AI-27 Calibration Offset	0 (default)	unitless, can be used to fix an offset to a value read
Analog Input	26028	Present Value	AI-28 Calibration Offset	AI-28 Calibration Offset	0 (default)	unitless, can be used to fix an offset to a value read
Analog Input	26031	Present Value	AI-31 Calibration Offset	AI-31 Calibration Offset	0 (default)	unitless, can be used to fix an offset to a value read
Analog Input	26032	Present Value	AI-32 Calibration Offset	AI-32 Calibration Offset	0 (default)	unitless, can be used to fix an offset to a value read
Analog Input	26033	Present Value	AI-33 Calibration Offset	AI-33 Calibration Offset	0 (default)	unitless, can be used to fix an offset to a value read
Analog Input	26034	Present Value	AI-34 Calibration Offset	AI-34 Calibration Offset	0 (default)	unitless, can be used to fix an offset to a value read
Analog Input	26035	Present Value	AI-35 Calibration Offset	AI-35 Calibration Offset	0 (default)	unitless, can be used to fix an offset to a value read
Analog Input	26036	Present Value	AI-36 Calibration Offset	AI-36 Calibration Offset	0 (default)	unitless, can be used to fix an offset to a value read
Analog Input	26037	Present Value	AI-37 Calibration Offset	AI-37 Calibration Offset	0 (default)	unitless, can be used to fix an offset to a value read
Analog Input	26038	Present Value	AI-38 Calibration Offset	AI-38 Calibration Offset	0 (default)	unitless, can be used to fix an offset to a value read
Analog Input	26041	Present Value	AI-41 Calibration Offset	AI-41 Calibration Offset	0 (default)	unitless, can be used to fix an offset to a value read
Analog Input	26042	Present Value	AI-42 Calibration Offset	AI-42 Calibration Offset	0 (default)	unitless, can be used to fix an offset to a value read
Analog Input	26043	Present Value	AI-43 Calibration Offset	AI-43 Calibration Offset	0 (default)	unitless, can be used to fix an offset to a value read
Analog Input	26044	Present Value	AI-44 Calibration Offset	AI-44 Calibration Offset	0 (default)	unitless, can be used to fix an offset to a value read
Analog Input	26045	Present Value	AI-45 Calibration Offset	AI-45 Calibration Offset	0 (default)	unitless, can be used to fix an offset to a value read
Analog Input	26046	Present Value	AI-46 Calibration Offset	AI-46 Calibration Offset	0 (default)	unitless, can be used to fix an offset to a value read
Analog Input	26047	Present Value	AI-47 Calibration Offset	AI-47 Calibration Offset	0 (default)	unitless, can be used to fix an offset to a value read
Analog Input	26048	Present Value	AI-48 Calibration Offset	AI-48 Calibration Offset	0 (default)	unitless, can be used to fix an offset to a value read
Analog Input	26051	Present Value	AI-51 Calibration Offset	AI-51 Calibration Offset	0 (default)	unitless, can be used to fix an offset to a value read
Analog Input	26052	Present Value	AI-52 Calibration Offset	AI-52 Calibration Offset	0 (default)	unitless, can be used to fix an offset to a value read

# P864 Diagnostic Points continued.

Object Type	Object Instance	Object Property	Object Name	Description	Present Value	Notes
Analog Input	26053	Present Value	AI-53 Calibration Offset	AI-53 Calibration Offset	0 (default)	unitless, can be used to fix an offset to a value read
Analog Input	26054	Present Value	AI-54 Calibration Offset	AI-54 Calibration Offset	0 (default)	unitless, can be used to fix an offset to a value read
Analog Input	26055	Present Value	AI-55 Calibration Offset	AI-55 Calibration Offset	0 (default)	unitless, can be used to fix an offset to a value read
Analog Input	26056	Present Value	AI-56 Calibration Offset	AI-56 Calibration Offset	0 (default)	unitless, can be used to fix an offset to a value read
Analog Input	26057	Present Value	AI-57 Calibration Offset	AI-57 Calibration Offset	0 (default)	unitless, can be used to fix an offset to a value read
Analog Input	26058	Present Value	AI-58 Calibration Offset	AI-58 Calibration Offset	0 (default)	unitless, can be used to fix an offset to a value read
Analog Input	26061	Present Value	AI-61 Calibration Offset	AI-61 Calibration Offset	0 (default)	unitless, can be used to fix an offset to a value read
Analog Input	26062	Present Value	AI-62 Calibration Offset	AI-62 Calibration Offset	0 (default)	unitless, can be used to fix an offset to a value read
Analog Input	26063	Present Value	AI-63 Calibration Offset	AI-63 Calibration Offset	0 (default)	unitless, can be used to fix an offset to a value read
Analog Input	26064	Present Value	AI-64 Calibration Offset	AI-64 Calibration Offset	0 (default)	unitless, can be used to fix an offset to a value read
Analog Input	26065	Present Value	AI-65 Calibration Offset	AI-65 Calibration Offset	0 (default)	unitless, can be used to fix an offset to a value read
Analog Input	26066	Present Value	AI-66 Calibration Offset	AI-66 Calibration Offset	0 (default)	unitless, can be used to fix an offset to a value read
Analog Input	26067	Present Value	AI-67 Calibration Offset	AI-67 Calibration Offset	0 (default)	unitless, can be used to fix an offset to a value read
Analog Input	26068	Present Value	AI-68 Calibration Offset	AI-68 Calibration Offset	0 (default)	unitless, can be used to fix an offset to a value read
Analog Input	26071	Present Value	AI-71 Calibration Offset	AI-71 Calibration Offset	0 (default)	unitless, can be used to fix an offset to a value read
Analog Input	26072	Present Value	AI-72 Calibration Offset	AI-72 Calibration Offset	0 (default)	unitless, can be used to fix an offset to a value read
Analog Input	26073	Present Value	AI-73 Calibration Offset	AI-73 Calibration Offset	0 (default)	unitless, can be used to fix an offset to a value read
Analog Input	26074	Present Value	AI-74 Calibration Offset	AI-74 Calibration Offset	0 (default)	unitless, can be used to fix an offset to a value read
Analog Input	26075	Present Value	AI-75 Calibration Offset	AI-75 Calibration Offset	0 (default)	unitless, can be used to fix an offset to a value read
Analog Input	26076	Present Value	AI-76 Calibration Offset	AI-76 Calibration Offset	0 (default)	unitless, can be used to fix an offset to a value read
Analog Input	26077	Present Value	AI-77 Calibration Offset	AI-77 Calibration Offset	0 (default)	unitless, can be used to fix an offset to a value read
Analog Input	26078	Present Value	AI-78 Calibration Offset	AI-78 Calibration Offset	0 (default)	unitless, can be used to fix an offset to a value read
Binary Input	20000	Present Value	Airflow Sensor detected	Airflow Sensor detected	Inactive	Detected=active (1), not detected=inactive (0)
Binary Input	20001	Present Value	864e Module 1 detected	864e v5.1 Module 1 detected	Active	Detected=active (1), not detected=inactive (0)

## P864 Diagnostic Points continued.

Object Type	Object Instance	Object Property	Object Name	Description	Present Value	Notes
Binary Input	20002	Present Value	864e Module 2 detected	864e v5.1 Module 1 detected	Active	Detected=active (1), not detected=inactive (0)
Binary Input	20003	Present Value	864e Module 3 detected	864e Module 3 detected	Inactive	Detected=active (1), not detected=inactive (0)
Binary Input	20004	Present Value	864e Module 4 detected	864e Module 4 detected	Inactive	Detected=active (1), not detected=inactive (0)
Binary Input	20005	Present Value	864e Module 5 detected	864e Module 5 detected	Inactive	Detected=active (1), not detected=inactive (0)
Binary Input	20006	Present Value	864e Module 6 detected	864e Module 6 detected	Inactive	Detected=active (1), not detected=inactive (0)
Binary Input	20007	Present Value	864e Module 7 detected	864e Module 7 detected	Inactive	Detected=active (1), not detected=inactive (0)
Binary Input	20010	Present Value	Core Update Status	Core Update Status	Inactive	Internal use
Binary Input	20011	Present Value	BACnet Update Status	BACnet Update Status	Inactive	Internal use
Binary Input	20012	Present Value	API-UI Update Status	API-UI Update Status	Inactive	Internal use
Binary Input	20010	Present Value	Metric Mode	<blank>	Active	US Customary=active (1) Metric=inactive (0)



# Complete Building Optimisation

