

# MATTER INTEGRATION FOR CRESTRON

Revision: 1.0

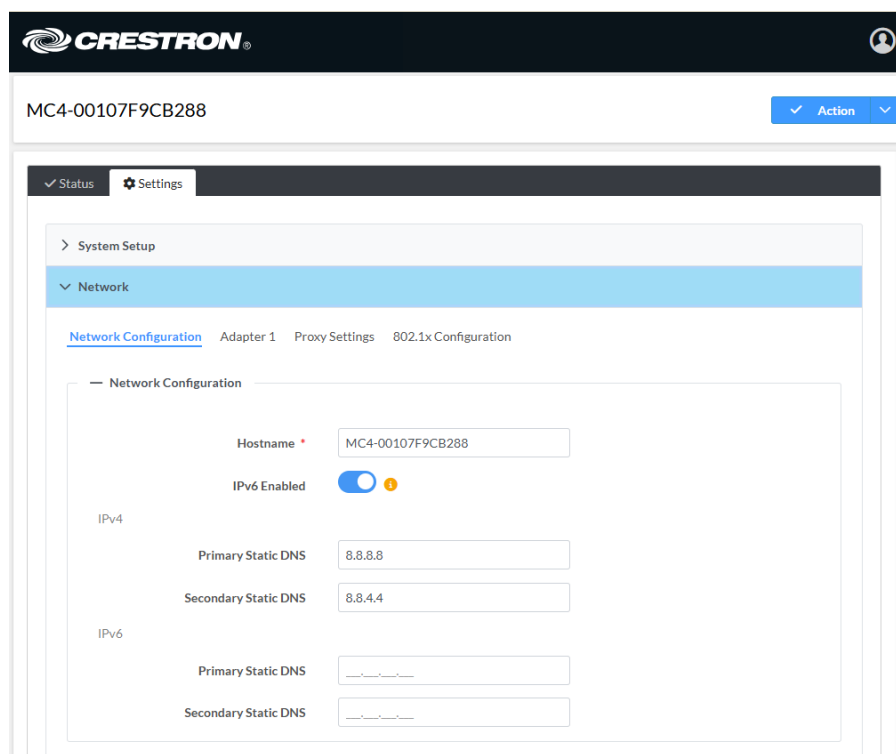
Date: 30 March 2026

## MODULE DESCRIPTION

This help file details the setup guide for integrating Matter devices into Crestron Custom. One recommended integration is combining these Matter modules with our SIMPL Home module to get Matter devices into Crestron Home.

This module is only compatible with 4-series Crestron processors and requires a Windows computer to setup the Matter devices to be controlled by the processor.

**NOTE:** To use this module, you must first enable IPv6 on the processor. To do this go to the processor's web interface in a browser and enter your login details. Then go to "Settings" -> "Network" and make sure "IPv6 Enabled" is on.



## Table of Contents

|                                              |    |
|----------------------------------------------|----|
| Module Description.....                      | 1  |
| Commissioning Matter Devices.....            | 4  |
| Commissioning Tool.....                      | 4  |
| Program Features .....                       | 4  |
| Step 1: Select Interface .....               | 5  |
| Step 2: Select Processor .....               | 5  |
| Step 3: Commissioning Devices.....           | 7  |
| Step 3 - Device Information Screen .....     | 11 |
| Step 4 - Uploading and Saving.....           | 13 |
| SIMPL Programming.....                       | 14 |
| General Programming Notes.....               | 14 |
| Parameters .....                             | 14 |
| Outputs.....                                 | 14 |
| Identify.....                                | 16 |
| Inputs .....                                 | 16 |
| Outputs.....                                 | 17 |
| OnOff.....                                   | 18 |
| Inputs .....                                 | 18 |
| Outputs.....                                 | 19 |
| Level Control.....                           | 20 |
| Inputs .....                                 | 20 |
| Outputs.....                                 | 20 |
| Colour Control.....                          | 21 |
| Inputs .....                                 | 21 |
| Outputs.....                                 | 23 |
| Power Source .....                           | 24 |
| Outputs.....                                 | 24 |
| Robot Vacuum Cleaner Run Mode .....          | 26 |
| Inputs .....                                 | 26 |
| Outputs.....                                 | 26 |
| Robot Vacuum Cleaner Clean Mode.....         | 27 |
| Robot Vacuum Cleaner Operational State ..... | 28 |
| Inputs .....                                 | 28 |
| Outputs.....                                 | 28 |

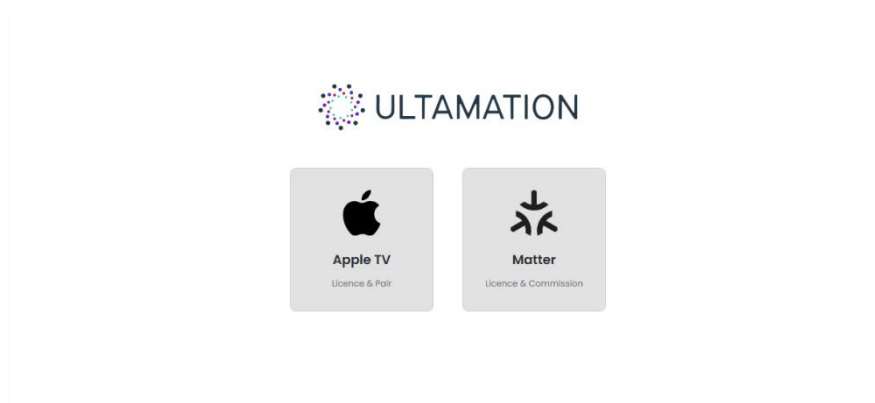
|                                         |                                     |
|-----------------------------------------|-------------------------------------|
| Service Area.....                       | 30                                  |
| Inputs .....                            | 30                                  |
| Outputs.....                            | 30                                  |
| Door Lock.....                          | 31                                  |
| Inputs .....                            | 31                                  |
| Outputs.....                            | 32                                  |
| SIMPL Matter Device Types.....          | 34                                  |
| SIMPL Matter Colour Light .....         | 34                                  |
| Inputs .....                            | 35                                  |
| Outputs.....                            | 37                                  |
| SIMPL Matter Dimmable Light.....        | <b>Error! Bookmark not defined.</b> |
| SIMPL Matter Robot Vacuum Cleaner ..... | 39                                  |
| Inputs .....                            | 39                                  |
| Outputs.....                            | 40                                  |
| Support.....                            | 43                                  |
| Licencing.....                          | 43                                  |
| Cloud Licence .....                     | 43                                  |

## COMMISSIONING MATTER DEVICES

For the Crestron system to be able to communicate with a Matter device it must be "paired" with our controller. You can do this using our new Windows application, the Ultamation Commissioning Tool. This is also the tool that allows you to pair Apple TVs to a Crestron processor.

### Commissioning Tool

Please download and install the commissioning tool. This can be obtained from the Matter product page on our shop website.



### Program Features

The commissioning tool provides the following features:

- Create a new config file
- Download a config file from a processor or load it from a file
- Commission Matter devices
- View Matter devices configuration
- View Implemented/Unimplemented clusters
- Save or upload the config file
- See available licences

## Step 1: Select Interface

The screenshot shows the 'COMMISSIONING TOOL v5.0.0.0' interface. On the left is a dark sidebar with a vertical menu containing five items: '1. Select Interface' (highlighted), '2. Select Processor', '3. Commissioning', '4. Device Information', and '5. Upload'. At the bottom of the sidebar is a 'Home' button and a circular logo. The main content area is titled 'Select Interface' and contains the following text: 'Select your PC's correct network interface. If the wrong one is selected, you may not be able to connect to your matter devices. If you experience any issues commissioning, try changing the interface selected here and try again.' Below this text is a dropdown menu currently showing 'Ethernet 3'.

This step is required to ensure that you can properly communicate with the matter devices throughout the commissioning process. Please ensure you have the correct interface selected before proceeding. If you have trouble commissioning devices, one troubleshooting step would be to change the selected interface here and try again.

## Step 2: Select Processor

**IMPORTANT:** Ensure that your machine's firewall includes exceptions for the Ultamation Commissioning Manager tool to ensure you receive discovery packets for both Crestron and Matter devices.

The screenshot shows the 'COMMISSIONING TOOL v5.0.0.0' interface. The sidebar menu is the same as in Step 1, but '2. Select Processor' is now highlighted. The main content area is titled 'Processor' and contains the following elements: a section labeled 'Discovered Devices:' with a table showing one device '[MC4] MC4-' with a MAC address field containing '@19:'; two input fields for 'Processor MAC:' (containing '00:10:7f') and 'Processor IP:' (containing '192.168'); and three buttons at the bottom: 'New Config File', 'Load Config File', and 'Extract Config'.

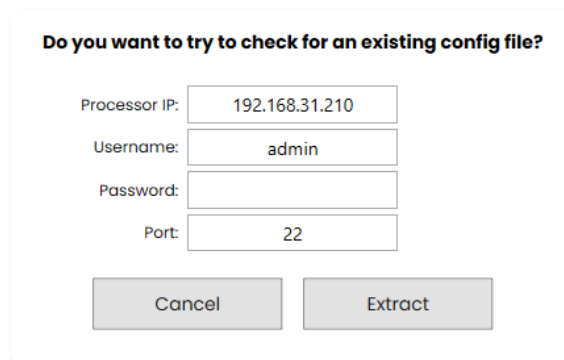
Here you can select a processor that you can download a config file from or select the processor that you want to upload the config file to. If you do not have access to the Crestron processor, please ensure you type in the MAC address of the processor before proceeding.

**NOTE:** When adding new devices to the config file after your initial setup, make sure to stop the program and download the config file from the processor first. If this doesn't happen the commissioning details can become out of sync and will require re-commissioning of devices.

You can load the file in two ways, by clicking the "Extract Config" button which will load the config file from the processor or by clicking "Load Config File" which will let you select a config file from your computer.

When starting from scratch, select the "New Config File" button.

To connect to a processor, you must know the SSH username and password.



**Do you want to try to check for an existing config file?**

Processor IP:

Username:

Password:

Port:

**NOTE:** Too many incorrect attempts will lock you out of the Crestron process for a time (default 24 hours). This can be cleared using a different machine from Crestron Toolbox.

## Step 3: Commissioning Devices

**COMMISSIONING TOOL**  
v5.0.0.0

1. Select Interface  
2. Select Processor  
3. Commissioning  
4. Device Information  
5. Upload

Processor MAC: 00:10:F0: [ ] Processor IP: 192.168. [ ]

**Commissioning**

Licences Used: 1/4

**Commissionable Devices**

| Device Name      | IPv6 Address             | Action     |
|------------------|--------------------------|------------|
| 1527FD3AAB932984 | fe80::217:88ff:fe6f:f6e0 | Commission |

**Commissioned Devices**

| Device ID | Device Name                | IPv6 Address                        | Licence | Action |
|-----------|----------------------------|-------------------------------------|---------|--------|
| 1         | Aqara Smart lock U200 Lite | fd94:e865:956a:186aa:6e65:e074:9330 |         | Remove |

Home

The top table displays the commissionable devices that have been found over mDNS on the network. Matter devices will only appear as commissionable once you have enabled a window through following the instructions available on the device or the device's native app.

The bottom table shows the currently commissioned devices found on the config file.

### *Commissioned Devices Table Features*

- Changing Device ID
  - Must be unique, conflicting device IDs will turn red and prevent saving to the file
- Changing Device Name
- Removing Device from the config file
- Commissioned devices will turn orange if they cannot be found through mDNS packets, this may require re-commissioning devices
  - To re-commission a device, please first remove it from the config file by clicking on the remove button for that device.

On this screen you will also be able to see the number of licences used against the number of licences available.

Conflicting Device Id's

COMMISSIONING TOOL  
v5.0.0.0

1. Select Interface

2. Select Processor

3. Commissioning

4. Device Information

5. Upload

Home

Processor MAC: 00:10:7f Processor IP: 192.16

Commissioning

Licences Used: 2/4

Commissionable Devices

| Device Name    | IPv6 Address | Action |
|----------------|--------------|--------|
| Discovering... |              |        |

Commissioned Devices

| Device ID | Device Name                                                                    | IPv6 Address                        | Licence | Action |
|-----------|--------------------------------------------------------------------------------|-------------------------------------|---------|--------|
| 2         | Aqara Smart lock U200 Lite                                                     | fd94:e865:956a:186aa:6e65:e074:9330 |         | Remove |
| 2         | This ID conflicts with another in the table. Please ensure all IDs are unique. |                                     |         | Remove |

Removing Device Confirmation

Are you sure you would like to delete Aqara Smart lock U200 Lite from the config file?

Cancel

Confirm

Device Not Found by mDNS

COMMISSIONING TOOL  
v5.0.0.0

1. Select Interface

2. Select Processor

3. Commissioning

4. Device Information

5. Upload

Home

Processor MAC: 00:10:7f Processor IP: 192.16

Commissioning

Licences Used: 1/4

Commissionable Devices

| Device Name    | IPv6 Address | Action |
|----------------|--------------|--------|
| Discovering... |              |        |

Commissioned Devices

| Device ID | Device Name                | IPv6 Address                        | Licence | Action |
|-----------|----------------------------|-------------------------------------|---------|--------|
| 2         | Aqara Smart lock U200 Lite | fd94:e865:956a:186aa:6e65:e074:9330 |         | Remove |

### *Commission a Device*

It is recommended that you commission devices one at a time, making sure to only open one pairing window at a time. Devices do not advertise with a very readable name and cannot be identified until the commissioning has been done so to avoid confusion do this one at a time.

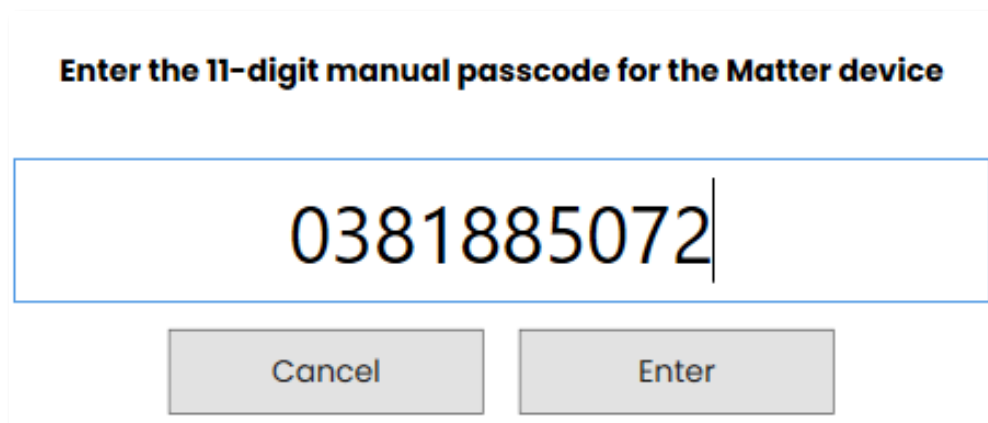
1. Open a commissioning window through the app used to set up the Matter device initially. If you get an option, make sure to choose the one that involves a "pairing code"
2. Click commission on the appropriate row in the "Commissionable Devices" table.
3. Enter the pairing code displayed on the app into the Commissioning Tool's prompt
4. Click "Enter"
5. You will eventually receive a message saying whether the commissioning process was successful or failed.

**NOTE:** If the commissioning process for a device fails, most devices will require a one-minute break before commissioning can be attempted again.

### *Common reasons for failure*

- Failing on Step 1:
  - The device has closed its pairing window, and a new one will need to be opened.
  - The pairing code has been entered incorrectly
- Failing on Step 4:
  - If the device has been commissioned once by the commissioning tool, it needs to be removed from the device - this will be done in a similar place to where you opened the pairing window

#### Step 4: Enter the "pairing code"



The screenshot shows a dialog box with the title "Enter the 11-digit manual passcode for the Matter device". Inside the dialog, there is a large text input field containing the 11-digit code "0381885072". Below the input field, there are two buttons: "Cancel" on the left and "Enter" on the right.

Commissioning Progress Screen

COMMISSIONING TOOL  
v5.0.0.0

1. Select Interface

2. Select Processor

3. Commissioning

4. Device Information

5. Upload

Home

Processor MAC: 00:10:7F Processor IP: 192.16

Commissioning

Licences Used: 1/4

| Device Name      | IPv6 Address             | Action     |
|------------------|--------------------------|------------|
| F0060375AFB8130A | fe80::217:88ff:fe6f:f6e0 | Commission |

| Device ID | Device Name                | IPv6 Address                        | Licence | Action |
|-----------|----------------------------|-------------------------------------|---------|--------|
| 2         | Aqara Smart lock U200 Lite | fd94:e865:956a:186aa:6e65:e074:9330 |         | Remove |

Commissioning Device...

Attempting to finalise commissioning

Successful Commissioning Screen

COMMISSIONING TOOL  
v5.0.0.0

1. Select Interface

2. Select Processor

3. Commissioning

4. Device Information

5. Upload

Home

Processor MAC: 00:10:7F Processor IP: 192.16

Commissioning

Licences Used: 2/4

| Device Name      | IPv6 Address             | Action     |
|------------------|--------------------------|------------|
| 041E28B31A403077 | fe80::217:88ff:fe6f:f6e0 | Commission |

| Device ID | Device Name                | IPv6 Address                        | Licence | Action |
|-----------|----------------------------|-------------------------------------|---------|--------|
| 2         | Aqara Smart lock U200 Lite | fd94:e865:956a:186aa:6e65:e074:9330 |         | Remove |
| 3         | BSB002                     | fe80::217:88ff:fe6f:f6e0%18         |         | Remove |

Commissioned Successfully

## Failed Commissioning Screen

**COMMISSIONING TOOL**  
v5.0.0.0

1. Select Interface

2. Select Processor

3. Commissioning

4. Device Information

5. Upload

Home

Processor MAC: 00:10:7F: Processor IP: 192.168

### Commissioning

Licences Used: 1/4

Commissionable Devices

| Device Name      | IPv6 Address             | Action     |
|------------------|--------------------------|------------|
| 544400C35C47CA0F | fe80::217:88ff:fe6f:f6e0 | Commission |

Commissioned Devices

| Device ID | Device Name                | IPv6 Address                        | Licence | Action |
|-----------|----------------------------|-------------------------------------|---------|--------|
| 1         | Aqara Smart lock U200 Lite | fd94:e865:956a:186aa:6e65:e074:9330 |         | Remove |

**Failed to Commission Device**

Attempting Step 1 of 5

### Step 3 - Device Information Screen

This is the third step on the commissioning manager. This step will display all the available Cluster's as well as the configuration required to get this working on Crestron.

There are two available display modes – "Complete Device Information" and "Implemented Clusters only".

#### *Implemented Clusters Only*

**COMMISSIONING TOOL**  
v5.0.0.0

1. Select Interface

2. Select Processor

3. Commissioning

4. Device Information

5. Upload

Home

Processor MAC: 00:10:7F:9 Processor IP: 192.168

### Device Information

Below is the Endpoint and Cluster information. Use this to configure your SIMPL program

Show All

**Aqara Smart lock U200 Lite – Device ID : 1**

| Endpoint ID | Endpoint Name | Device Type  | Clusters           |
|-------------|---------------|--------------|--------------------|
| 1           | Endpoint 1    | Door Lock    | Identify Door Lock |
| 2           | Endpoint 2    | Power Source | Power Source       |

**Aqara Smart lock U200 Lite – Device ID : 2**

| Endpoint ID | Endpoint Name | Device Type  | Clusters           |
|-------------|---------------|--------------|--------------------|
| 1           | Endpoint 1    | Door Lock    | Identify Door Lock |
| 2           | Endpoint 2    | Power Source | Power Source       |

This screen displays the available clusters that have been implemented as well as the configuration you will need to place into Crestron to get the devices working.

The "Endpoint Name" field will contain the name of individual devices for Matter devices such as the "Philips Hue Bridge" that can control multiple child devices.

### Crestron Configuration

- Device ID
  - This needs to be placed into every module to link them together
- Endpoint ID
  - This needs to match the "Endpoint ID" parameter field for the Cluster highlighted in green.
- Cluster
  - This will be the cluster needed to be added for this part of the device's functionality
  - Will be prefixed by "Simpl" on the actual module name

### Complete Device Information

The "Complete Device Information" will show you everything that the device can do so that in the future as more updates are made to our modules you can just add these to your program without re-commissioning devices.

**COMMISSIONING TOOL**  
v5.0.0.0

1. Select Interface

2. Select Processor

3. Commissioning

4. Device Information

5. Upload

Home

Processor MAC: 00:10:7F:9 Processor IP: 192.16

### Device Information

Below is the Endpoint and Cluster information. Use this to configure your SIMPL program

Show Supported Clusters

**Aqara Smart lock U200 Lite - Device ID : 1**

| Endpoint ID | Endpoint Name | Device Type  | Clusters   |              |           |
|-------------|---------------|--------------|------------|--------------|-----------|
| 1           | Endpoint 1    | Door Lock    | Identify   | Descriptor   | Door Lock |
| 2           | Endpoint 2    | Power Source | Descriptor | Power Source |           |

**Aqara Smart lock U200 Lite - Device ID : 2**

| Endpoint ID | Endpoint Name | Device Type  | Clusters   |              |           |
|-------------|---------------|--------------|------------|--------------|-----------|
| 1           | Endpoint 1    | Door Lock    | Identify   | Descriptor   | Door Lock |
| 2           | Endpoint 2    | Power Source | Descriptor | Power Source |           |

Key:

- Grey – not yet implemented
- Green - implemented

## Step 4 – Uploading and Saving

COMMISSIONING TOOL  
v0.0.0.0

1. Select Interface  
2. Select Processor  
3. Commissioning  
4. Device Information  
5. Upload

Home

**Finish**

Upload to selected processor or save config file

Your config includes devices that communicate over Thread.  
To support communication with these devices,  
we must include your Crestron processor username/password in the config file.

Upload Config File Save Config File

Processor MAC: 00:10: Processor IP: 192.168.

This is the final step and can upload the config file to the processor chosen or save the file.

If the processor displayed underneath the two buttons is not the processor you were wanting to upload the config file to, go back to step 1 and change to the correct processor.

Clicking "Upload Config File" will bring up a screen that requires entering the SSH details for the processor as before with "Extract config file". The config file will be saved on the processor at the file path: \user\Ultamation\CommissionedDevices.json. In the event of a processor failure, you may need to recommission devices as the latest config file is needed to regain control on starting the program. The config file will change some values on program startup as well.

If your configuration includes Matter Over Thread devices, then in order for the processor to communicate with them we need to set some IP routing on the processor. In order to facilitate this, we must store the password used for SSH in the config file when uploading to the processor. This means your credentials will be stored in plaintext on the processor. If this is unacceptable to you, please remove any Matter Over Thread devices and upload.

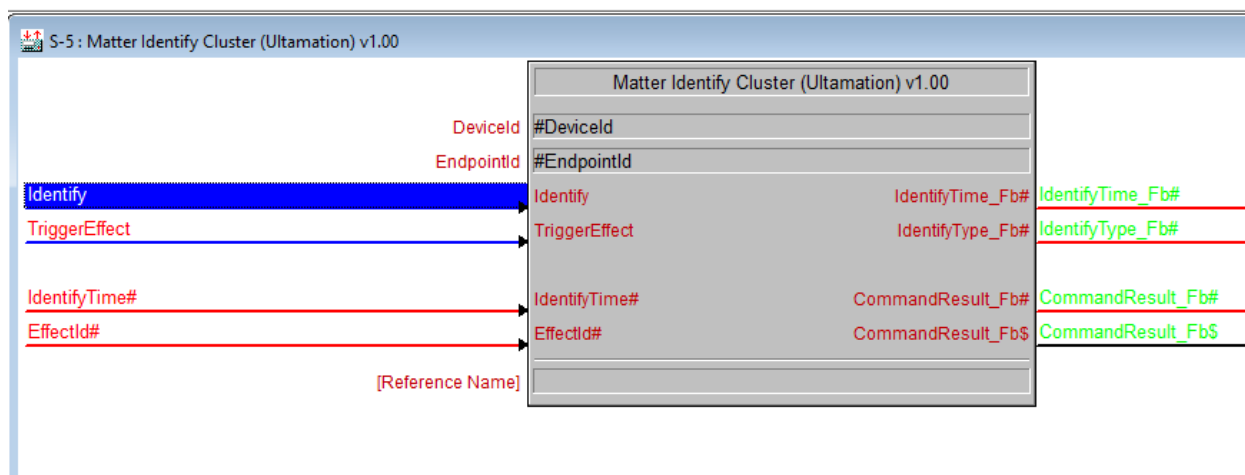
Clicking "Save Config File" will bring up the file explorer and allow you to save the config file where you wish to.

## SIMPL PROGRAMMING

### General Programming Notes

There are two ways to create the program required for the SIMPL integration of Matter. You can use the pre-built Matter device modules or individual modules that contain specific functionality..

The matter device modules combine the required functionality modules for a specific device type; some devices may have more functionality that doesn't fit into this so by using the specific device type version some functionality that the device has may not be usable.



Every Matter module has the following formatting so they will not be explained on each individual module description:

#### Parameters

**DeviceId** – This is the Device ID listed on the commissioning tool

**EndpointId** – This is the endpoint ID shown in the table that matches the cluster you are adding to the program.

#### Outputs

**CommandResult\_Fb\$** – This will display the result of the most recent command; this is needed as not every device will implement all the commands on a given Cluster. It is advised to try each command and see if the command feedback is "Success" or "Unsupported command".

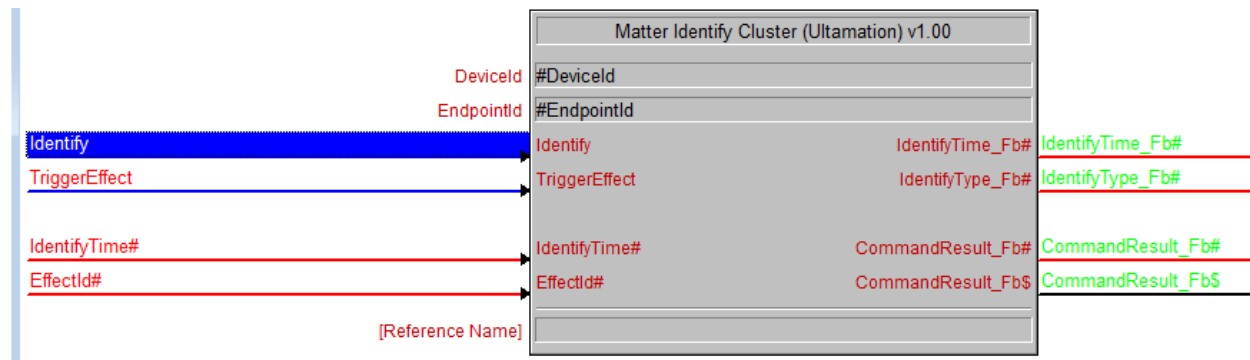
**CommandResult\_Fb#** – This will display a number representing the result of the command, should you wish to do any programming with the result of each command.

| Code | Description                  |
|------|------------------------------|
| 0    | Success                      |
| 1    | Failure                      |
| 125  | Invalid Subscription         |
| 126  | Unsupported Access           |
| 127  | Unsupported Endpoint         |
| 128  | Invalid Action               |
| 129  | Unsupported Command          |
| 133  | Invalid Command              |
| 134  | Unsupported Attribute        |
| 135  | Constraint Error             |
| 136  | Unsupported Write            |
| 137  | Resource Exhausted           |
| 138  | Not Found                    |
| 139  | Unreportable Attribute       |
| 140  | Invalid Data Type            |
| 141  | Unsupported Read             |
| 146  | Data Version Mismatch        |
| 148  | Timeout                      |
| 155  | Unsupported Node             |
| 195  | Unsupported Cluster          |
| 197  | No Upstream Subscription     |
| 198  | Needs timed interaction      |
| 199  | Unsupported Event            |
| 201  | Paths Exhausted              |
| 202  | Failsafe Required            |
| 203  | Invalid in State             |
| 204  | No Command Response          |
| 205  | Terms and Conditions Changed |
| 206  | Maintenance Required         |

Each Cluster implements a group of similar functionalities such as the OnOff cluster providing the ability to turn on/off a device and monitor it's "On" state.

Use the previously mentioned "Device Information Page" to build the SIMPL program according to the table. This can then be linked with our SIMPL Home product to get the devices into Crestron Home, either natively or through a custom UI tile.

## Identify



### Inputs

**Identify** – Executes the Identify Command for the length of time defined in **IdentifyTime#**.

**TriggerEffect** – Executes the trigger effect command with the effect defined in **EffectId#**.

**IdentifyTime#** – The time in seconds the device will identify itself for.

**EffectId#** – One of the following values:

| Value | Name           | Example                                                                                                                                                                                     |
|-------|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0     | Blink          | e.g., Light is turned on/off once.                                                                                                                                                          |
| 1     | Breathe        | e.g., Light is turned on/off over 1 second and repeated 15 times.                                                                                                                           |
| 2     | Okay           | e.g., Colored light turns green for 1 second; non-colored light flashes twice.                                                                                                              |
| 11    | Channel Change | e.g., Colored light turns orange for 8 seconds; non-colored light switches to the maximum brightness for 0.5s and then minimum brightness for 7.5s.                                         |
| 254   | Finish Effect  | Complete the current effect sequence before terminating. e.g., if in the middle of a breathe effect (as above), first complete the current 1s breathe effect and then terminate the effect. |
| 255   | Stop Effect    | Terminate the effect as soon as possible.                                                                                                                                                   |

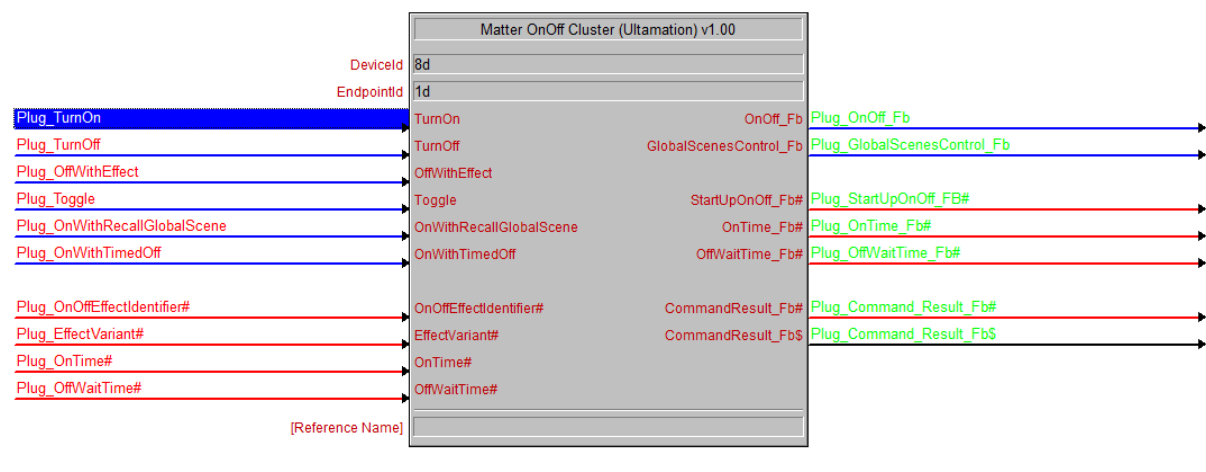
## Outputs

**IdentifyTime\_Fb#** - The remaining length of time the device will continue to identify itself for

**IdentifyType\_Fb#** - How the device identifies itself from one of the following values:

| Value | Name              | Summary                                                                                                           |
|-------|-------------------|-------------------------------------------------------------------------------------------------------------------|
| 0     | None              | No presentation.                                                                                                  |
| 1     | Light Output      | Light output of a lighting product.                                                                               |
| 2     | Visible Indicator | Typically a small LED.                                                                                            |
| 3     | Audible Beep      | Some sound value.                                                                                                 |
| 4     | Display           | Presentation will be visible on display screen.                                                                   |
| 5     | Actuator          | Presentation will be conveyed by actuator functionality such as through a window blind operation or inwall relay. |

## OnOff



### Inputs

**TurnOn** – Executes the Turn On command

**TurnOff** – Executes the Turn Off command

**OffWithEffect** – Turns the device off using the **OnOffEffectIdentifier#** and **EffectVariant#** inputs.

**Toggle** – Swaps the device's on/off state.

**OnWithRecallGlobalScene** – Turns on the device and sets it to the state it was in before it turned off

**OnWithTimedOff** – Turns the device on with a guarded off duration, that will prevent this command turning the device on again if it is turned off in that duration. If this command is issued whilst the device is online the period that the device is turned on will increase. Uses the **OnTime#** input to determine how long to turn the device on for. Uses the **OffWaitTime#** for the guarded duration.

**OnOffEffectIdentifier#** - Can be one of two values 0 – Delayed All Off or 1 – Dying Light. These values will change the meaning of the **EffectVariant#** input.

**EffectVariant#** -

When **OnOffEffectIdentifier#** is Delayed All Off

| Value | Name                       | Summary                                                        |
|-------|----------------------------|----------------------------------------------------------------|
| 0     | Delayed Off with Fast Fade | Fades to off in 0.8 seconds                                    |
| 1     | No Fade                    | Turns straight off                                             |
| 2     | Delayed Off with Slow Fade | Dims by 50% in 0.8 seconds and then fades to off in 12 seconds |

When **OnOffEffectIdentifier#** is Dying Light

| Value | Name                 | Summary                                                               |
|-------|----------------------|-----------------------------------------------------------------------|
| 0     | Dying Light Fade Off | Increases the dim by 20% in 0.5 seconds then fades to off in 1 second |

**OnTime#** - Changes the on duration when used with **OnWithTimedOff**

**OffWaitTime#** - Changes the off-guard time when used with **OnWithTimedOff**

### Outputs

**OnOff\_Fb** – The current on state of the associated device

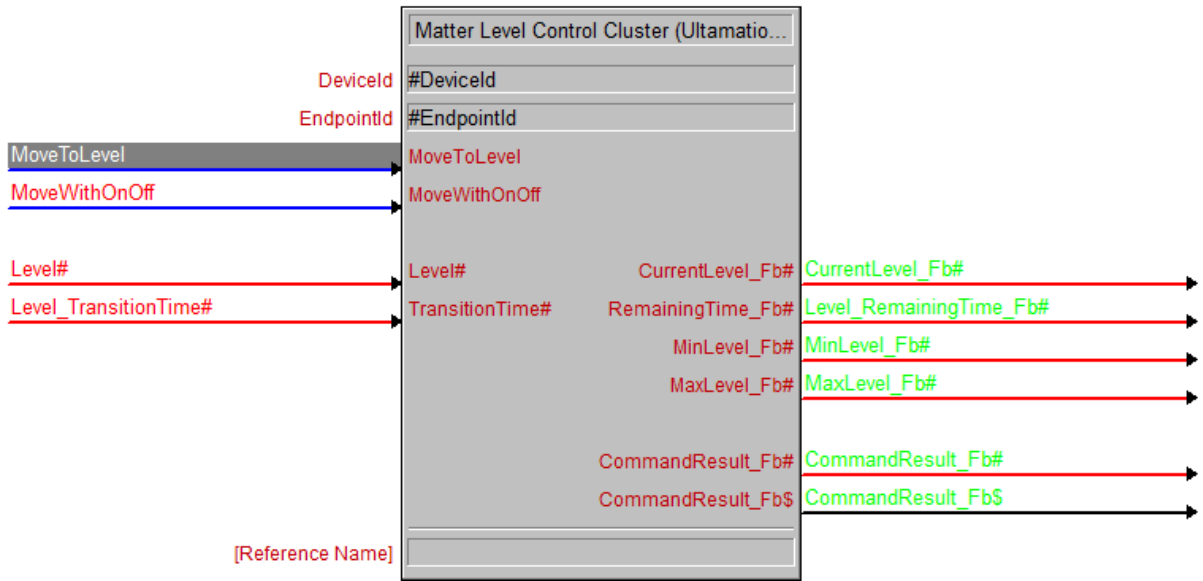
**GlobalScenesControl\_Fb** – Is true when a command changes the state of the associated device to be on.

**StartUpOnOff\_Fb#** - Indicates what the device will do when starting up: 0 means off, 1 means on, 2 means toggle (whatever the previous attribute was when it lost power)

**OnTime\_Fb#** - The length of time remaining in 1/10ths of a second before the device will turn off, will only have a value after an **OnWithTimedOff** is received.

**OffWaitTime\_Fb#** - The length of time in 1/10ths of a second that the Off state shall be guarded for to prevent an **OnWithTimedOff** turning the device back on.

Level Control



Inputs

**MoveToLevel** – Changes the level of the associated device using the **Level#** input and will take the length specified in **TransitionTime#**

**MoveWithOnOff** – When raised the **MoveToLevel** command will also turn the light on/off if the level is set to a level that would cause the state to change.

**Level#** - A value between 0 and 65535 representing 0 to 100% values.

**TransitionTime#** - The time in 1/10<sup>th</sup> of a second for the transition to complete.

Outputs

**CurrentLevel\_Fb#** - The current level of the associated device, represented as a percentage between 0 and 65535.

**RemainingTime\_Fb#** - The time remaining in 1/10<sup>th</sup> of a second for the transition to finish.

**MinLevel\_Fb#** - The minimum level the device can be set to. Within the range of (1-254)

**MaxLevel\_Fb#** - The maximum level the device can be set to. Within the range of (1-254)

## Colour Control



### Inputs

**MoveToHue** – Will transition the hue, using the **Hue#**, **TransitionTime#** and **Direction#** signals.

**CycleHueUp** – Upon raising this will move the hue upwards using the **Rate#** as the number of steps per second. If this would reach the maximum hue value it will wrap around to the lowest hue value. When the signal is lowered the command to stop will be sent.

**CycleHueDown** – This will move the hue downwards using the **Rate#** as the number of steps per second. If this would reach the minimum hue value it will wrap around to the lowest hue value. When the signal is lowered the command to stop will be sent.

**IncrementHue** – This will increase the hue value by the amount specified in **StepSize#** over the period specified by **TransitionTime#**.

**DecrementHue** – This will decrease the hue value by the amount specified in **StepSize#** over the period specified by **TransitionTime#**.

**MoveToSaturation** – Will change the saturation value using a transition. Uses the **Saturation#** and **TransitionTime#**.

**MoveSaturationUp** – Upon raising this will move the saturation upwards using the **Rate#** as the number of steps per second. This will stop when it reaches the maximum saturation value or the signal is lowered.

**MoveSaturationDown** – Upon raising this will move the saturation downwards using the **Rate#** as the number of steps per second. This will stop when it reaches the minimum saturation value or the signal is lowered.

**IncrementSaturation** – This will increase the saturation value by the amount specified in **StepSize#** over the period specified by **TransitionTime#**.

**DecrementSaturation** – This will decrease the saturation value by the amount specified in **StepSize#** over the period specified by **TransitionTime#**.

**MoveToHueAndSaturation** – Will change the hue and saturation to the requested values using a transition. Will use the **TransitionTime#**.

**MoveToColourTemperature** – Will move the colour temperature to the value specified in **ColourTemperatureMireds#** using a transition specified in **TransitionTime#**.

**MoveColourTemperatureUp** – Upon raising this will increase the colour temperature using the **Rate#** as the number of steps per second until it reaches the maximum value or the signal is lowered. This is bounded by **MinimumColourTemperatureMireds#** and **MaximumColourTemperatureMireds#**.

**MoveColourTemperatureDown** – Upon raising this will decrease the colour temperature using the **Rate#** as the number of steps per second until it reaches the minimum value or the signal is lowered. This is bounded by **MinimumColourTemperatureMireds#** and **MaximumColourTemperatureMireds#**.

**IncrementColourTemperature** – This will increase the colour temperature by the amount specified in **StepSize#** over the period specified by **TransitionTime#** within the boundaries of **MinimumColourTemperatureMireds#** and **MaximumColourTemperatureMireds#**.

**DecrementColourTemperature** – This will decrease the colour temperature by the amount specified in **StepSize#** over the period specified by **TransitionTime#** within the boundaries of **MinimumColourTemperatureMireds#** and **MaximumColourTemperatureMireds#**.

**Hue#** – A value between 0 and 3600.

**Saturation#** – A value between 0 and 65535.

**ColourTemperatureMireds#** – A value between 0 and 65535.

**Direction#** – One of the following:

| Value | Name              |
|-------|-------------------|
| 0     | Shortest Distance |
| 1     | Longest Distance  |
| 2     | Up                |
| 3     | Down              |

**TransitionTime#** - The time in 1/10ths of a second you wish for the change in value to take place.

**Rate#** - A value between 0 and 255 representing the number of movements per second. The values for Hue and Saturation are between 0 and 254.

**StepSize#** - A value between 0 and 255 representing the change to be added/subtracted from the current value.

**MinimumColourTemperatureMireds#** - A value between 0 and 65535.

**MaximumColourTemperatureMireds#** - A value between 0 and 65535.

### Outputs

**SupportsHueAndSaturation\_Fb** - This will be high if the light supports hue and saturation.

**SupportsEnhancedHue\_Fb** - This will be high if the light supports enhanced hue.

**SupportsColourLoop\_Fb** - This will be high if the light supports Colour Loop.

**SupportsColourTemperature\_Fb** - This will be high if the light supports Colour Temperature.

**CurrentHue\_Fb#** - A value between 0 and 3600 representing the current Hue of the device.

**CurrentSaturation\_Fb#** - A value between 0 and 65535 representing the current Saturation

**RemainingTime\_Fb#** - The time remaining in 1/10<sup>th</sup>s of a second until the active transitions will complete.

**ColourTemperatureMireds\_Fb#** - A value between 0 and 65535 representing the current colour temperature.

**ColourMode\_Fb#** - One of the following representing what determines the device's colour:

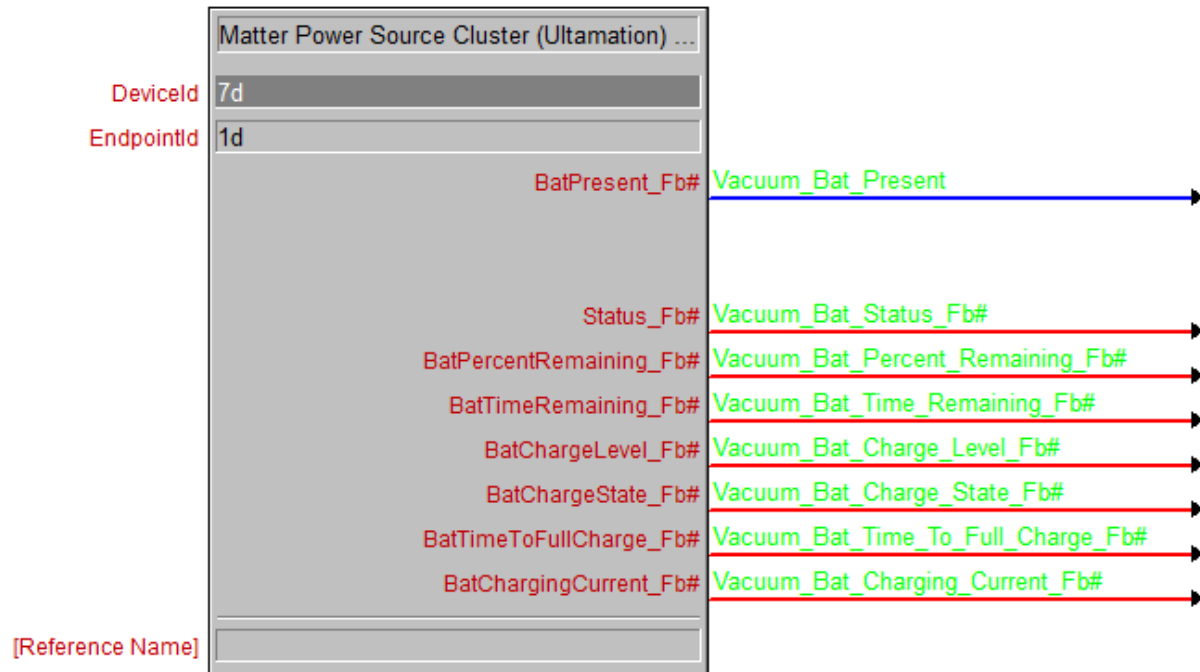
| Value | Name                               |
|-------|------------------------------------|
| 0     | Current Hue and Current Saturation |
| 1     | Current X and Current Y            |
| 2     | Colour Temperature Mireds          |

**MinColourTemperatureMireds\_Fb#** - A value between 0 and 65535.

**MaxColourTemperatureMireds\_Fb#** - A value between 0 and 65535.

## Power Source

Not all the feedback will be available for every device that contains this. Unsupported attributes will all remain 0.



## Outputs

**BatPresent\_Fb** - This signal will be raised if the associated device detects a battery

**Status\_Fb#** - A value representing the status of the power source:

| Value | Name        | Details                                                   |
|-------|-------------|-----------------------------------------------------------|
| 0     | Unspecified | The source is not specified                               |
| 1     | Active      | The source is available and currently supplying power     |
| 2     | Standby     | The source is available but not currently supplying power |
| 3     | Unavailable | Source is not currently available to supply power         |

**BatPercentRemaining\_Fb#** - The battery percentage between 0 and 1000. This will be the battery percentage as a value x10.

**BatTimeRemaining\_Fb#** - The estimated time in seconds before the battery will no longer be able to provide power to the device, a value of 0 may mean the device is unable to assess the value.

**BatChargeLevel\_Fb#** - A value representing the need to charge the battery

| Value | Name | Details                  |
|-------|------|--------------------------|
| 0     | OK   | The charge level is okay |

|   |          |                                                      |
|---|----------|------------------------------------------------------|
| 1 | Warning  | The charge level is low; you may need to charge soon |
| 2 | Critical | Charge level is critical. You need to charge now     |

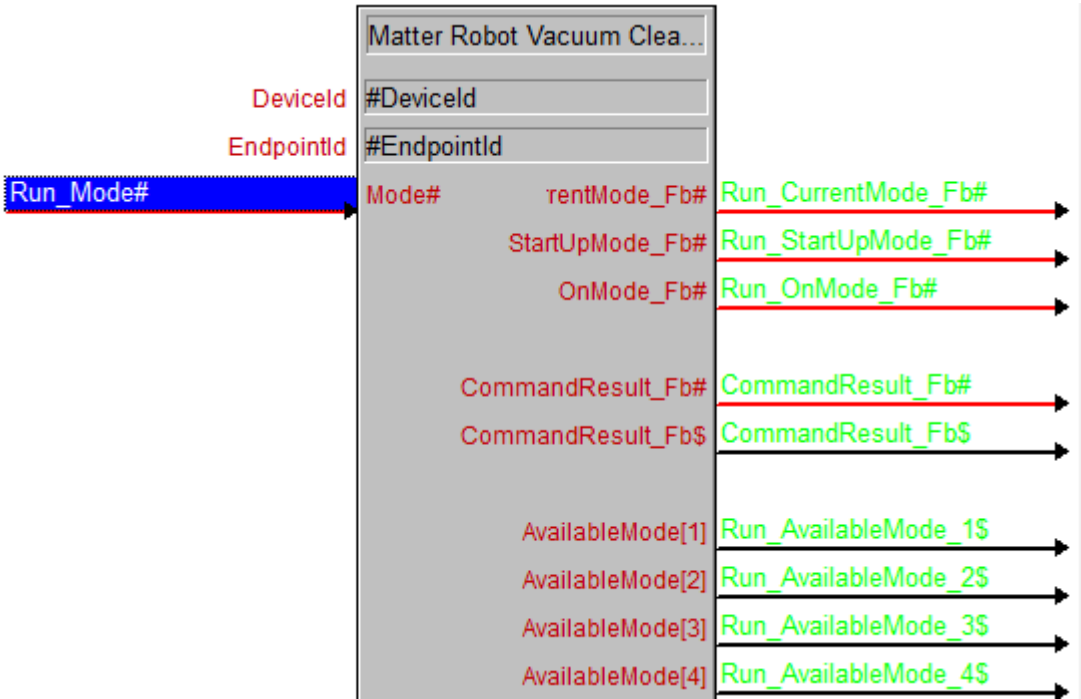
**BatChargeState\_Fb#** - A value representing the charging status of the battery

| Value | Name              |
|-------|-------------------|
| 0     | Unknown           |
| 1     | Is Charging       |
| 2     | Is At Full Charge |
| 3     | Is Not Charging   |

**BatTimeToFullCharge\_Fb#** - The estimated time in seconds before the battery source will be at full charge. A value of 0 may mean that it is unable to get an estimated time.

**BatChargingCurrent\_Fb#** - The assessed current in mA (milliamps) presently supplied to charge the battery source.

Robot Vacuum Cleaner Run Mode



Inputs

**Mode#** - On changing this value a command will be sent to change to one of the modes shown in "AvailableMode[x]"

Outputs

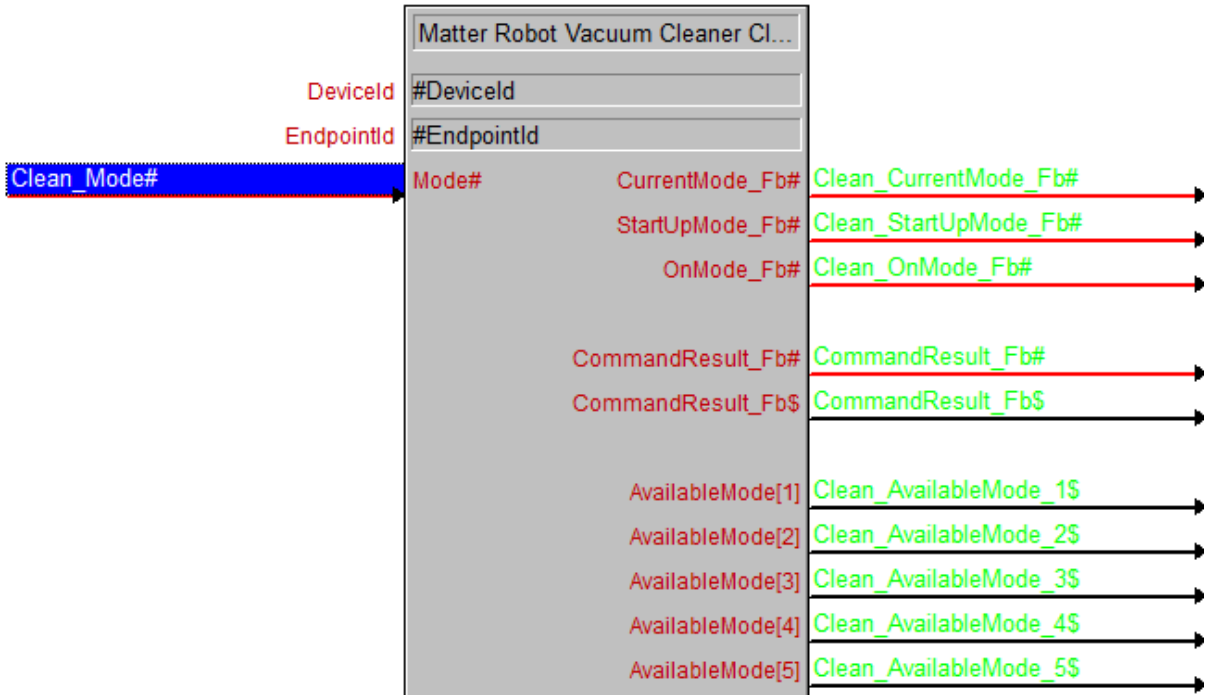
**CurrentMode\_Fb#** - The currently selected running mode as described in "AvailableMode[x]"

**StartUpMode\_Fb#** - This will show what mode the device will start up in when it gains power.

**OnMode\_Fb#** - If this value is not 0 then the **OnMode** is dependent on the On Off Cluster, if this is the case when turning from off to on the **CurrentMode** will change to the mode specified here.

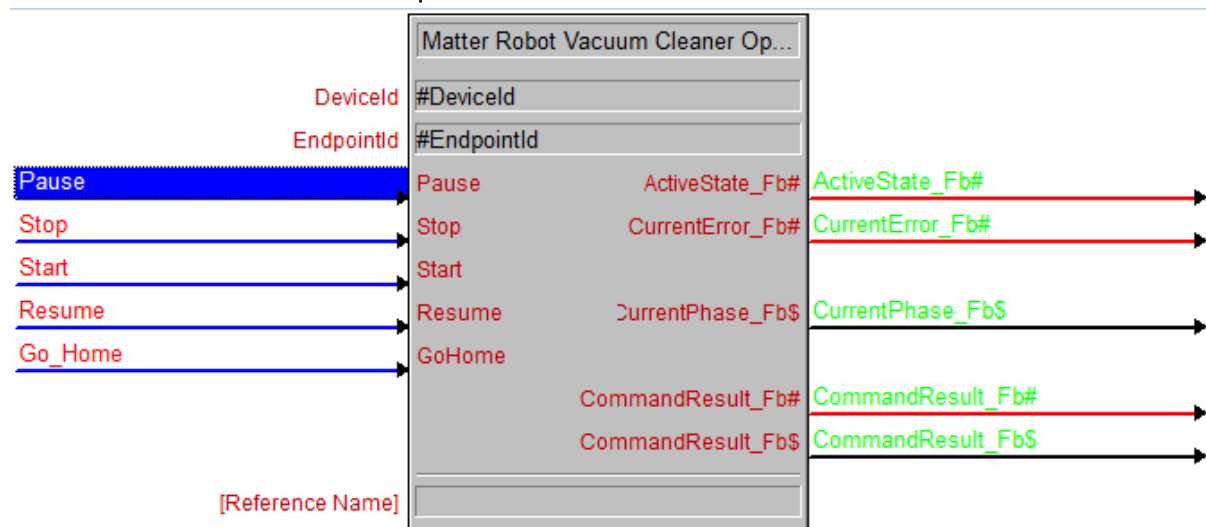
**AvailableMode[x]** - A string describing the available modes for the running state of the associated device. The number here is used to change to the mode using the Mode# signal.

Robot Vacuum Cleaner Clean Mode



This module has the same inputs and outputs as the RVC Run mode cluster but will represent the cleaning mode of the associated device.

## Robot Vacuum Cleaner Operational State



### Inputs

**Pause** – Upon raising, will pause the current mode the associated device is operating in.

**Stop** – This will end the current mode the associated device is operating in, preventing resuming from working.

**Start** – This will start the associated device based on the mode selected in the RVC run mode Cluster.

**Resume** – This will resume the mode the associated device was operating in based on the mode the device was in after a Pause command.

**GoHome** – This will send the device back to seek the charging dock, providing the current state of the associated device allows for this action.

### Outputs

**ActiveState\_Fb#** - Will display the current state of the device as shown in the following table.

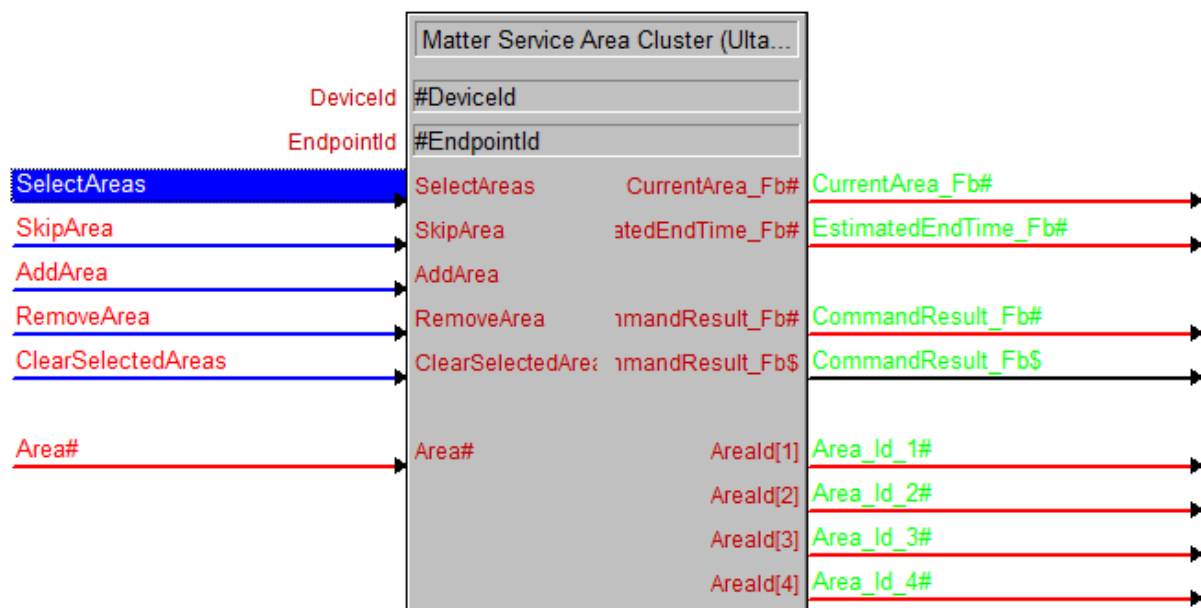
| Value | Name               |
|-------|--------------------|
| 0     | Stopped            |
| 1     | Running            |
| 2     | Paused             |
| 3     | Error              |
| 64    | Seeking Charger    |
| 65    | Charging           |
| 66    | Docked             |
| 67    | Emptying Dust Bin  |
| 68    | Cleaning Mop       |
| 69    | Filling Water Tank |
| 70    | Updating Maps      |

**CurrentError\_Fb#** - Will display the current state of the device as shown in the following table.

| Value | Name                         |
|-------|------------------------------|
| 0     | No Error                     |
| 1     | Unable to Start or Resume    |
| 2     | Unable to Complete Operation |
| 3     | Command Invalid State        |
| 64    | Failed To Find Charging Dock |
| 65    | Stuck                        |
| 66    | Dust Bin Missing             |
| 67    | Dust Bin Full                |
| 68    | Water Tank Empty             |
| 69    | Water Tank Missing           |
| 70    | Water Tank Lid Open          |
| 71    | Mop Cleaning Pad Missing     |
| 72    | Low Battery                  |
| 73    | Cannot Reach Target Area     |
| 74    | Dirty Water Tank Full        |
| 75    | Dirty Water Tank Missing     |
| 76    | Wheels Jammed                |
| 77    | Brush Jammed                 |
| 78    | Navigation Sensor Obscured   |

**CurrentPhase\_Fb\$** - This will display the device specific, current phase such as "Vacuuming" or "Charging"

## Service Area



The available areas for this by default is 50. A device may have up to 255 areas, if you need more than 50 you can edit the MAX\_AREAS constant in the "ServiceAreaCluster vx.xx usf" file.

### Inputs

**SelectAreas** – Upon rising this will send the list of areas that has been created using the **AddArea** signal.

**SkipArea** – This will send a command telling the device to skip the area currently in **Area#**.

**AddArea** – This will add the **Area#** to a list of areas internally until it is removed or cleared using the other inputs.

**RemoveArea** – This will remove the **Area#** from the list of areas internally.

**ClearSelectedAreas** – This will remove all areas from the list of areas internally.

**Area#** – This should be one of the id's listed in the **AreaId[x]** outputs. To be used to control which areas the device operates over.

### Outputs

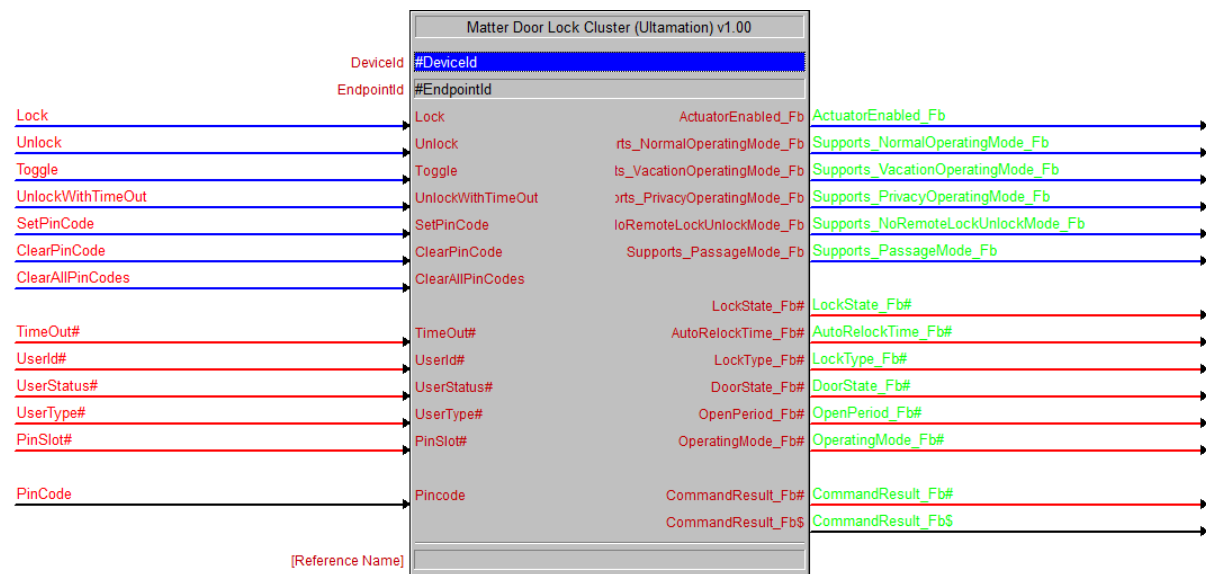
**CurrentArea\_Fb#** – The current area the device is operating in from the id's listed in the **AreaId[x]** outputs.

**EstimatedEndTime\_Fb#** – The estimated Epoch time in seconds for the completing the task in the **CurrentArea\_Fb#** if available.

**AreaId[x]** – The id for the area with the name shown in the **AreaName[x]** output.

**AreaName[x]** – The name of the area with the id shown in the **AreaId[x]** output

## Door Lock



### Inputs

**Lock** – This will lock the door, taking in the **Pincode** if the device supports pin codes.

**Unlock** – This will unlock the door, taking in the **Pincode** if the device supports pin codes.

**Toggle** – This will toggle the locked state of the door.

**UnlockWithTimeOut** – This will unlock the door with a timeout, that will lock the door again after the time specified by **TimeOut#** in seconds. Taking in the value in **Pincode** again if the device supports pin codes. This may require configuring the device separately to allow the locking to happen. If the device does not support this, the device will unlock and stay unlocked.

**SetPinCode** – This will allow you to set a pin code for a specific user using the **UserId#**, **UserStatus#**, **UserType#** and **PinCode#** inputs.

**ClearPinCode** – This will clear the pin code for a given pin slot using the **PinSlot#** input.

**ClearAllPinCodes** – This will clear the all the pin codes on the device.

**Timeout#** - The time in seconds to be used with **UnlockWithTimeOut**

**UserId#** - The user id for a given pin code command

**UserStatus#** - One of the following values:

| Value | Name              | Details                              |
|-------|-------------------|--------------------------------------|
| 0     | Available         | The user ID is available             |
| 1     | Occupied Enabled  | The user ID is occupied and enabled  |
| 2     | Occupied Disabled | The user ID is occupied and disabled |

**UserType#** - One of the following values:

| Value | Name                     | Details                                                                                                              |
|-------|--------------------------|----------------------------------------------------------------------------------------------------------------------|
| 0     | Unrestricted User        | The user ID type is unrestricted                                                                                     |
| 1     | Year Day Schedule User   | The user can open a lock within a specific period                                                                    |
| 2     | Weekday Schedule User    | The user can open the lock based on a specific period within a recurring weekly schedule                             |
| 3     | Programming User         | The user can program and operate the door lock                                                                       |
| 4     | Non-Access User          | The user is recognised but will not be able to operate the lock                                                      |
| 5     | Forced User              | The user can open the lock, but the device will alert that this user has opened the lock                             |
| 6     | Disposable User          | The user can open the lock once, afterwards they will not be able to                                                 |
| 7     | Expiring User            | The user can open a lock for a limited amount of time                                                                |
| 8     | Schedule Restricted User | The user is restricted to a weekday schedule or year day schedule. If neither are set the user will be denied access |
| 9     | Remote Only User         | The user is restricted to remote lock/unlock only.                                                                   |

**PinSlot#** - The id for the slot of the pin code to remove

**Pincode** - The pin code for use with unlock/lock or setting

### Outputs

**ActuatorEnabled\_Fb** - This will be high if the lock is currently able to process Lock, Unlock and Unlock with Timeout commands

**Supports\_NormalOperatingMode\_Fb** - Will be high if the lock supports the normal operating mode

**Supports\_VacationOperatingMode\_Fb** - Will be high if the lock supports the vacation operating mode

**Supports\_PrivacyOperatingMode\_Fb** - Will be high if the lock supports the privacy operating mode

**Supports\_NoRemoteLockUnlockMode\_Fb** - Will be high if the lock supports the no remote lock unlock mode

**Supports\_PassageMode\_Fb** - Will be high if the lock supports the passage mode

**LockState\_Fb#** - One of the following values:

| Value | Name             |
|-------|------------------|
| 0     | Not Fully Locked |
| 1     | Locked           |
| 2     | Unlocked         |
| 3     | Unlatched        |

**AutoRelockTime\_Fb#** - The time in seconds the lock will relock automatically after an unlock command. If this is 0 auto-relocking is disabled.

**LockType\_Fb#** - One of the following values:

| Value | Name                |
|-------|---------------------|
| 0     | Dead Bolt           |
| 1     | Magnetic            |
| 2     | Other               |
| 3     | Mortise             |
| 4     | Rim                 |
| 5     | Latch Bolt          |
| 6     | Cylindrical Lock    |
| 7     | Tubular Lock        |
| 8     | Interconnected Lock |
| 9     | Dead Latch          |
| 10    | Door Furniture      |
| 11    | Euro Cylinder       |

**DoorState\_Fb#** - One of the following values:

| Value | Name                   |
|-------|------------------------|
| 0     | Door Open              |
| 1     | Door Closed            |
| 2     | Door Jammed            |
| 3     | Door Forced Open       |
| 4     | Door Unspecified Error |
| 5     | Door Ajar              |

**OpenPeriod\_Fb#** - The number of minutes the door has been open since the last time it was opened

**OperatingMode\_Fb#** - The current operating mode from the following:

| Value | Name     |
|-------|----------|
| 0     | Normal   |
| 1     | Vacation |

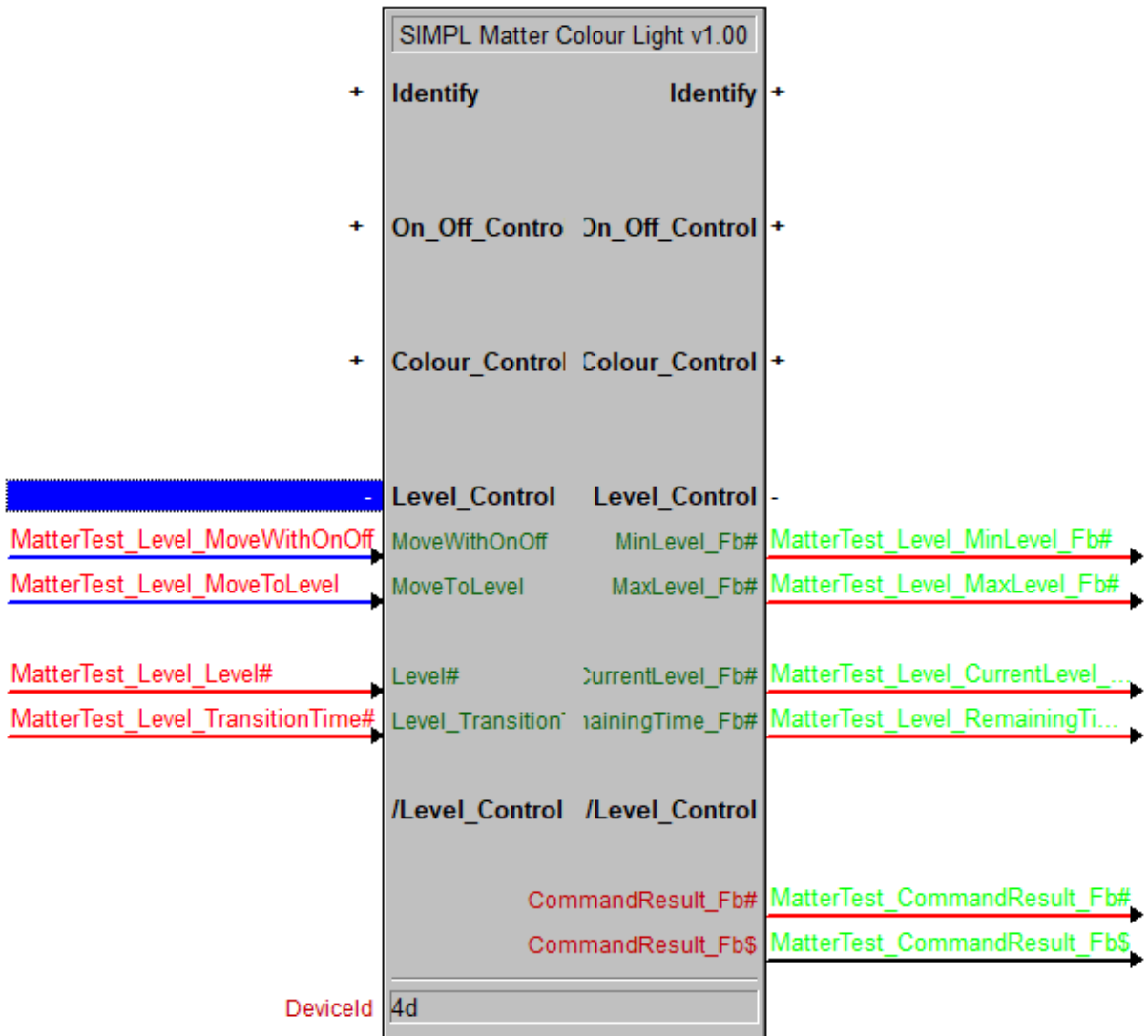
|   |                       |
|---|-----------------------|
| 2 | Privacy               |
| 3 | No Remote Lock Unlock |
| 4 | Passage               |

SIMPL Matter Device Types

The following are the available device type modules that can be used in the module rather than adding in individual modules shown above. These can be used for each device endpoint if the device type matches what is shown in the "Device Type" column on the commissioning tool.

SIMPL Matter Colour Light

This device type contains the functionality needed to control a colour light, supporting hue and saturation as well as colour temperature depending on the light's capabilities.



## Inputs

**TurnOn** – Executes the Turn On command

**TurnOff** – Executes the Turn Off command

**OffWithEffect** – Turns the device off using the **OnOffEffectIdentifier#** and **EffectVariant#** inputs.

**Toggle** – Swaps the device's on/off state.

**OnWithRecallGlobalScene** – Turns on the device and sets it to the state it was in before it turned off

**OnWithTimedOff** – Turns the device on with a guarded off duration, that will prevent this command turning the device on again if it is turned off in that duration. If this command is issued whilst the device is online the period that the device is turned on will increase. Uses the **OnTime#** input to determine how long to turn the device on for. Uses the **OffWaitTime#** for the guarded duration.

**OnOffEffectIdentifier#** - Can be one of two values 0 – Delayed All Off or 1 – Dying Light. These values will change the meaning of the **EffectVariant#** input.

**EffectVariant#** -

When **OnOffEffectIdentifier#** is Delayed All Off

| Value | Name                       | Summary                                                        |
|-------|----------------------------|----------------------------------------------------------------|
| 0     | Delayed Off with Fast Fade | Fades to off in 0.8 seconds                                    |
| 1     | No Fade                    | Turns straight off                                             |
| 2     | Delayed Off with Slow Fade | Dims by 50% in 0.8 seconds and then fades to off in 12 seconds |

When **OnOffEffectIdentifier#** is Dying Light

| Value | Name                 | Summary                                                               |
|-------|----------------------|-----------------------------------------------------------------------|
| 0     | Dying Light Fade Off | Increases the dim by 20% in 0.5 seconds then fades to off in 1 second |

**OnTime#** - Changes the on duration when used with **OnWithTimedOff**

**OffWaitTime#** - Changes the off-guard time when used with **OnWithTimedOff**

**Colour\_TransitionTime#** - The time in 1/10ths of a second you wish for the change in value to take place for colour related commands.

**Direction#** - One of the following:

| Value | Name              |
|-------|-------------------|
| 0     | Shortest Distance |
| 1     | Longest Distance  |
| 2     | Up                |
| 3     | Down              |

**MoveToHue** – Will transition the hue, using the **Hue#**, **Colour\_TransitionTime#** and **Direction#** signals.

**Hue#** - A value between 0 and 3600.

**MoveToSaturation** – Will change the saturation value using a transition. Uses the **Saturation#** and **Colour\_TransitionTime#**.

**Saturation#** - A value between 0 and 65535.

**MoveToHueAndSaturation** – Will change the hue and saturation to the requested values using a transition. Will use the **Colour\_TransitionTime #**.

**MoveToColourTemperature** – Will move the colour temperature to the value specified in **ColourTemperatureMireds#** using a transition specified in **Colour\_TransitionTime #**.

**ColourTemperatureMireds#** - A value between 0 and 65535.

**Rate#** - A value between 0 and 255 representing the number of movements per second. The values for Hue and Saturation are between 0 and 254.

**CycleHueUp** – Upon raising this will move the hue upwards using the **Rate#** as the number of steps per second. If this would reach the maximum hue value it will wrap around to the lowest hue value. When the signal is lowered the command to stop will be sent.

**CycleHueDown** – This will move the hue downwards using the **Rate#** as the number of steps per second. If this would reach the minimum hue value it will wrap around to the lowest hue value. When the signal is lowered the command to stop will be sent.

**MoveSaturationUp** – Upon raising this will move the saturation upwards using the **Rate#** as the number of steps per second. This will stop when it reaches the maximum saturation value or the signal is lowered.

**MoveSaturationDown** – Upon raising this will move the saturation downwards using the **Rate#** as the number of steps per second. This will stop when it reaches the minimum saturation value or the signal is lowered.

**MoveColourTemperatureUp** – Upon raising this will increase the colour temperature using the **Rate#** as the number of steps per second until it reaches the maximum value or the signal is lowered. This is bounded by **MinimumColourTemperatureMireds#** and **MaximumColourTemperatureMireds#**.

**MoveColourTemperatureDown** – Upon raising this will decrease the colour temperature using the **Rate#** as the number of steps per second until it reaches the minimum value or

the signal is lowered. This is bounded by **MinimumColourTemperatureMireds#** and **MaximumColourTemperatureMireds#**.

**MinimumColourTemperatureMireds#** - A value between 0 and 65535.

**MaximumColourTemperatureMireds#** - A value between 0 and 65535.

**StepSize#** - A value between 0 and 255 representing the change to be added/subtracted from the current value.

**IncrementColourTemperature** - This will increase the colour temperature by the amount specified in **StepSize#** over the period specified by **Colour\_TransitionTime #** within the boundaries of **MinimumColourTemperatureMireds#** and **MaximumColourTemperatureMireds#**.

**DecrementColourTemperature** - This will decrease the colour temperature by the amount specified in **StepSize#** over the period specified by **Colour\_TransitionTime #** within the boundaries of **MinimumColourTemperatureMireds#** and **MaximumColourTemperatureMireds#**.

**IncrementHue** - This will increase the hue value by the amount specified in **StepSize#** over the period specified by **Colour\_TransitionTime #**.

**DecrementHue** - This will decrease the hue value by the amount specified in **StepSize#** over the period specified by **Colour\_TransitionTime #**.

**IncrementSaturation** - This will increase the saturation value by the amount specified in **StepSize#** over the period specified by **Colour\_TransitionTime #**.

**DecrementSaturation** - This will decrease the saturation value by the amount specified in **StepSize#** over the period specified by **Colour\_TransitionTime #**.

**MoveToLevel** - Changes the level of the associated device using the **Level#** input and will take the length specified in **TransitionTime#**

**MoveWithOnOff** - When raised the **MoveToLevel** command will also turn the light on/off if the level is set to a level that would cause the state to change.

**Level#** - A value between 0 and 65535 representing 0 to 100% values.

**Level\_TransitionTime#** - The time in  $1/10^{\text{th}}$  of a second for the transition to complete.

## Outputs

**OnOff\_Fb** - The current on state of the associated device

**GlobalScenesControl\_Fb** - Is true when a command changes the state of the associated device to be on.

**StartUpOnOff\_Fb#** - Indicates what the device will do when starting up: 0 means off, 1 means on, 2 means toggle (whatever the previous attribute was when it lost power)

**OnTime\_Fb#** - The length of time remaining in 1/10ths of a second before the device will turn off, will only have a value after an **OnWithTimedOff** is received.

**OffWaitTime\_Fb#** - The length of time in 1/10ths of a second that the Off state shall be guarded for to prevent an **OnWithTimedOff** turning the device back on.

**SupportsHueAndSaturation\_Fb** - This will be high if the light supports hue and saturation.

**SupportsEnhancedHue\_Fb** - This will be high if the light supports enhanced hue.

**SupportsColourLoop\_Fb** - This will be high if the light supports Colour Loop.

**SupportsColourTemperature\_Fb** - This will be high if the light supports Colour Temperature.

**ColourMode\_Fb#** - One of the following representing what determines the device's colour:

| Value | Name                               |
|-------|------------------------------------|
| 0     | Current Hue and Current Saturation |
| 1     | Current X and Current Y            |
| 2     | Colour Temperature Mireds          |

**Colour\_RemainingTime\_Fb#** - The time remaining in 1/10<sup>th</sup>s of a second until the active transitions will complete.

**CurrentHue\_Fb#** - A value between 0 and 3600 representing the current Hue of the device.

**CurrentSaturation\_Fb#** - A value between 0 and 65535 representing the current Saturation

**ColourTemperatureMireds\_Fb#** - A value between 0 and 65535 representing the current colour temperature.

**MinColourTemperatureMireds\_Fb#** - A value between 0 and 65535.

**MaxColourTemperatureMireds\_Fb#** - A value between 0 and 65535.

**MinLevel\_Fb#** - The minimum level the device can be set to. Within the range of (1-254)

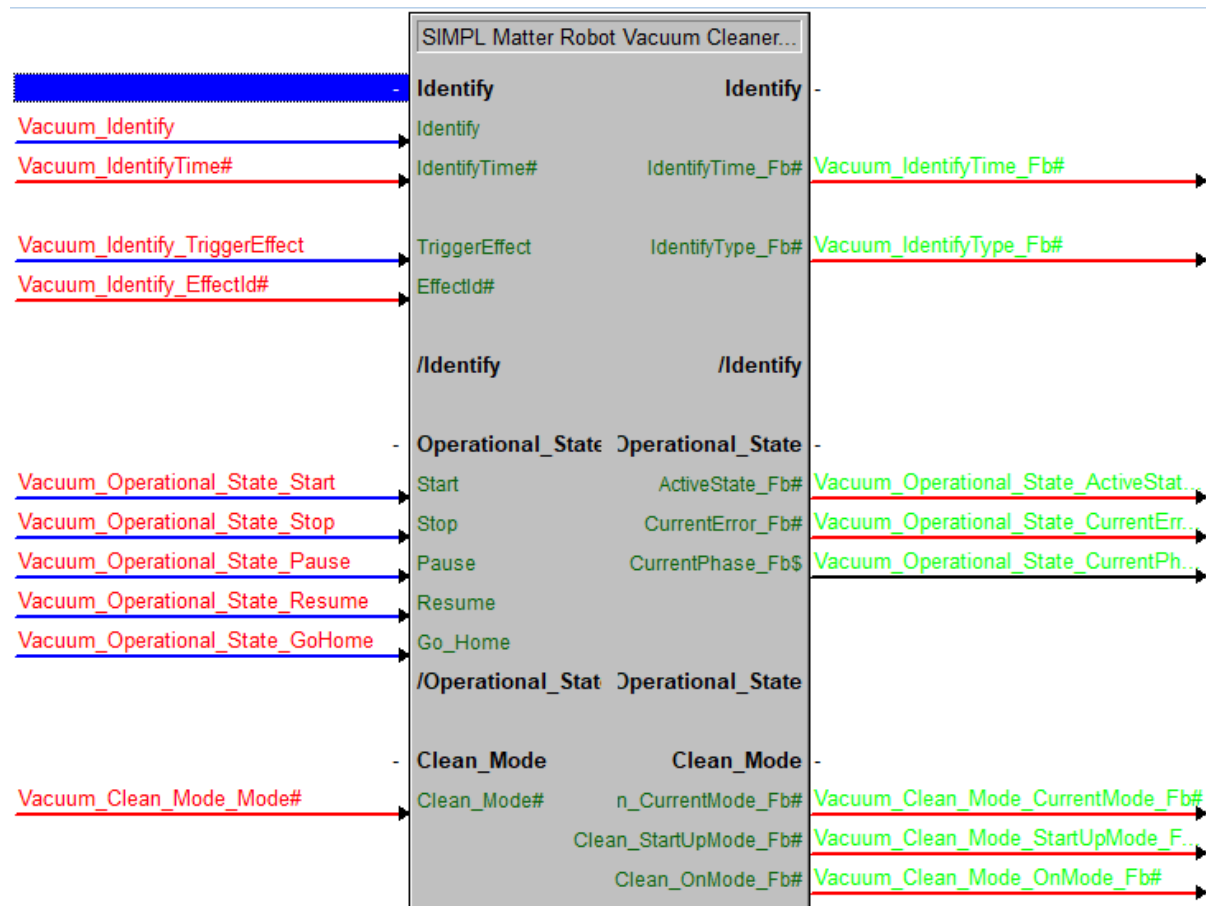
**MaxLevel\_Fb#** - The maximum level the device can be set to. Within the range of (1-254)

**CurrentLevel\_Fb#** - The current level of the associated device, represented as a percentage between 0 and 65535.

**Level\_RemainingTime\_Fb#** - The time remaining in 1/10<sup>th</sup> of a second for the transition to finish.

## SIMPL Matter Robot Vacuum Cleaner

This device type contains the capabilities for altering the cleaning mode and running mode of the device from the lists of the available modes. The state of the device such as running, paused, go home to charger can all be controlled too as well as selecting certain areas from the list of areas on the device to operate over.



### Inputs

**Start** – This will start the associated device based on the mode selected in the RVC run mode Cluster.

**Stop** – This will end the current mode the associated device is operating in, preventing resuming from working.

**Pause** – Upon raising, will pause the current mode the associated device is operating in.

**Resume** – This will resume the mode the associated device was operating in based on the mode the device was in after a Pause command.

**Go\_Home** – This will send the device back to seek the charging dock, providing the current state of the associated device allows for this action.

**Clean\_Mode#** - On changing this value a command will be sent to change to one of the modes shown in "Clean\_AvailableMode\_IxI\$"

**Run\_Mode#** - On changing this value a command will be sent to change to one of the modes shown in "Run\_AvailableMode\_[]\$"

**Area#** - This should be one of the id's listed in the **Areaid[]** outputs. To be used to control which areas the device operates over.

**AddArea** - This will add the **Area#** to a list of areas internally until it is removed or cleared using the other inputs.

**RemoveArea** - This will remove the **Area#** from the list of areas internally.

**ClearSelectedAreas** - This will remove all areas from the list of areas internally.

**SelectAreas** - Upon rising this will send the list of areas that has been created using the **AddArea** signal.

**SkipArea** - This will send a command telling the device to skip the area currently in **Area#**.

### Outputs

**ActiveState\_Fb#** - Will display the current state of the device as shown in the following table.

| Value | Name               |
|-------|--------------------|
| 0     | Stopped            |
| 1     | Running            |
| 2     | Paused             |
| 3     | Error              |
| 64    | Seeking Charger    |
| 65    | Charging           |
| 66    | Docked             |
| 67    | Emptying Dust Bin  |
| 68    | Cleaning Mop       |
| 69    | Filling Water Tank |
| 70    | Updating Maps      |

**CurrentError\_Fb#** - Will display the current state of the device as shown in the following table.

| Value | Name                         |
|-------|------------------------------|
| 0     | No Error                     |
| 1     | Unable to Start or Resume    |
| 2     | Unable to Complete Operation |
| 3     | Command Invalid State        |
| 64    | Failed To Find Charging Dock |

|    |                            |
|----|----------------------------|
| 65 | Stuck                      |
| 66 | Dust Bin Missing           |
| 67 | Dust Bin Full              |
| 68 | Water Tank Empty           |
| 69 | Water Tank Missing         |
| 70 | Water Tank Lid Open        |
| 71 | Mop Cleaning Pad Missing   |
| 72 | Low Battery                |
| 73 | Cannot Reach Target Area   |
| 74 | Dirty Water Tank Full      |
| 75 | Dirty Water Tank Missing   |
| 76 | Wheels Jammed              |
| 77 | Brush Jammed               |
| 78 | Navigation Sensor Obscured |

**CurrentPhase\_Fb\$** - This will display the device specific, current phase such as "Vacuuming" or "Charging"

**Clean\_CurrentMode\_Fb#** - The currently selected clean mode as described in "Clean\_AvailableMode\_x\$"

**Clean\_StartUpMode\_Fb#** - This will show what clean mode the device will start up in when it gains power.

**Clean\_OnMode\_Fb#** - If this value is not 0 then the **OnMode** is dependent on the On Off functionality, if this is the case when turning from off to on the **CurrentMode** will change to the mode specified here.

**Clean\_AvailableMode\_x\$** - A string describing the available modes for the clean state of the associated device. The number here is used to change to the mode using the Mode# signal.

**Run\_CurrentMode\_Fb#** - The currently selected running mode as described in "Run\_AvailableMode\_x\$"

**Run\_StartUpMode\_Fb#** - This will show what mode the device will start up in when it gains power.

**Run\_OnMode\_Fb#** - If this value is not 0 then the **OnMode** is dependent on the On Off functionality, if this is the case when turning from off to on the **CurrentMode** will change to the mode specified here.

**Run\_AvailableMode\_x\$** - A string describing the available modes for the running state of the associated device. The number here is used to change to the mode using the Mode# signal.

**CurrentArea\_Fb#** - The current area the device is operating in from the id's listed in the Areald[x] outputs.

**EstimatedEndTime\_Fb#** - The estimated Epoch time in seconds for the completing the task in the **CurrentArea\_Fb#** if available.

**Areaid[x]** – The id for the area with the name shown in the **AreaName[x]** output.

**AreaName[x]** – The name of the area with the id shown in the **Areaid[x]** output

## SUPPORT

If you have any issues with an integration solution please let us know by contacting Ultamation support on [support@ultamation.com](mailto:support@ultamation.com) and please include as much detail about your issue as possible, such as a recent processor error log.

Licence verification messages are posted to the error log, so please ensure you have checked this.

## LICENCING

This integration solution (including software, images and all other associated assets distributed as part of the purchased download package) is licenced on a PER PROCESSOR basis.

A purchase should not be completed without correct information as refunds cannot be issued for errors or changes made to details following purchase.

This is an electronic product and there is no physical delivery.

The integration solution is provided without any warranty with respect to the reliability of the controlled device or changes to device protocol. We will endeavour, through best efforts, to maintain the integration solution's functionality and any bug fixes will be provided free-of-charge. Additional functionality may be released as a variation of this integration solution and this will be a separate, purchasable, product.

## CLOUD LICENCE

This integration solution contacts Ultamation's licencing server at startup. If the server finds a matching licence for the integration solution and processor then the integration solution will be licenced. Otherwise, the integration solution will check the offline licence key. If you purchase a licence **after** you have loaded the integration solution, please reboot the system to see changes take effect.

If you purchased a licence **before** it was migrated to the cloud service, i.e. you have a licence key already, you must enter this into the **Offline Key** user attribute. If you purchased the integration solution **after** it was migrated, and you don't have a licence key, no further action is required.

If no licence exists for the product/processor the integration solution will enter a short trial period (ONE HOUR) to allow for verification of correct control or evaluation.

To request an OFFLINE key, please contact [support@ultamation.com](mailto:support@ultamation.com) with your order details and a brief explanation why you REQUIRE offline activation. Ultamation reserve the right to refuse offline activation.

**NOTE:** Once an offline key has been issued no further licence changes will be granted. Moving the integration solution to a new processor will require an additional licence purchase.