MUSE
AUTOMATOR

Instruction Manual
v1.0

HARMAN
A SAMSUNG COMPANY

Installation & Setup

MUSE Automator is a no-code/low-code software application designed for use with AMX MUSE Controllers. It is built on Node-RED, a widely used flow-based programming tool.

Prerequisites

Before installing MUSE Automator, you **must** install several dependencies outlined below. If these dependencies are not installed first, Automator will not run correctly.

1) Install NodeJS (v20.11.1+) & Node Package Manager (NPM) (v10.2.4+)

Automator is a custom version of the Node-RED software, so it requires NodeJS to run on your system. It also requires Node Package Manager (NPM) to be able to install third-party nodes. To install NodeJS and NPM, go to the following link and follow the installation instructions: https://docs.npmjs.com/downloading-and-installing-node-js-and-npm

2) Install Git (v2.43.0+)

Git is a version control system. For Automator, it enables the Project feature so that you can organize your flows into discrete projects. It also enables the Push/Pull functionality required to deploy your flows to a physical MUSE Controller. To install Git, go to the following link and follow the instructions: https://git-scm.com/book/en/v2/Getting-Started-Installing-Git

Note: The Git installer will take you through a series of installation options. It is recommended to use the default and installer-recommended options. Please refer to the Git documentation for more information.

Install MUSE Automator

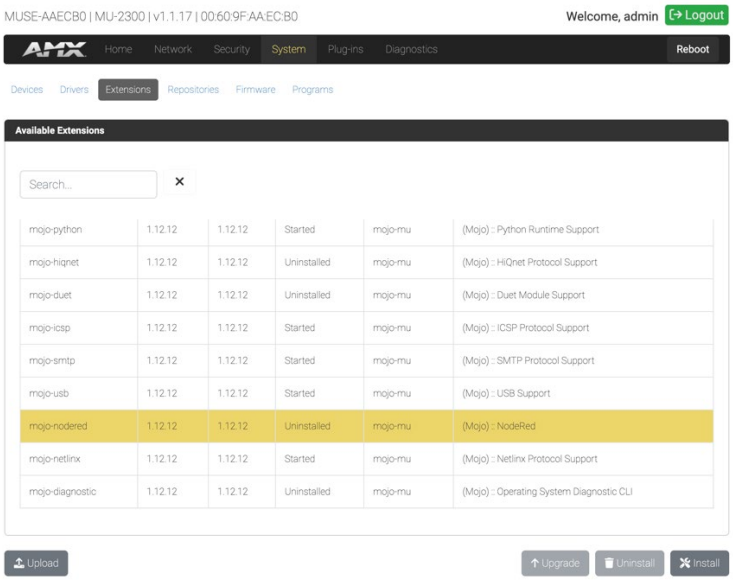
Once Git, NodeJS, and NPM have been installed, you can install MUSE Automator. Install MUSE Automator on your Windows or MacOS PC and follow the respective installer instructions.

Install MUSE Controller Firmware

To use MUSE Automator with an AMX MUSE controller, you will need to update the MUSE controller firmware available on amx.com.

Enable Node-RED Support in MUSE Controller

Node-RED is disabled on the MUSE controller by default. It must be manually enabled. To do this, log into your MUSE controller and navigate to System > Extensions. In the Available Extensions list, scroll down to *mojo-nodred* and click it to select it. Press the *Install* button to install the Node-RED extension and allow the controller to update. See screenshot below for reference:



Other Information

If you have a firewall enabled on your PC, you will need to make sure you have Port 49152 open for Automator to communicate through this port properly.

Getting Started with MUSE Automator

Get to know Node-RED

Since Automator is essentially a customized version of Node-RED, you should first become familiar with the Node-RED application. The software has a relatively shallow learning curve. There are hundreds of articles and instructional videos available to learn Node-RED, but a good place to start is in the Node-RED documentation: https://nodered.org/docs. In particular, read through the Tutorials, Cookbook, and Developing Flows to familiarize yourself with the application's features and user interface.

This guide will not cover the basics of Node-RED or flow-based programming, so it is imperative that you review the official Node-RED documentation prior to getting started.

Automator Interface Overview

The Automator editor interface is essentially the same as the Node-RED default editor with some tweaks to themes and some custom functionality that enables interaction between the editor and a MUSE controller.

1. **MUSE Automator Palette** – custom nodes for working with HARMAN devices
2. **Flow Tab** – For switching between views of multiple flows
3. **Workspace** – Where you build your flows. Drag nodes from the left and drop onto workspace
4. **Push/Pull Tray** – For managing projects locally or on a controller. Push, pull, start, stop, delete a project.
5. **Deploy Button/Tray** – For deploying flows from the editor to the local Node-RED server
6. **Hamburger Menu** – Main menu of application. Create projects, open projects, manage flows, etc.

Automator Modes of Working

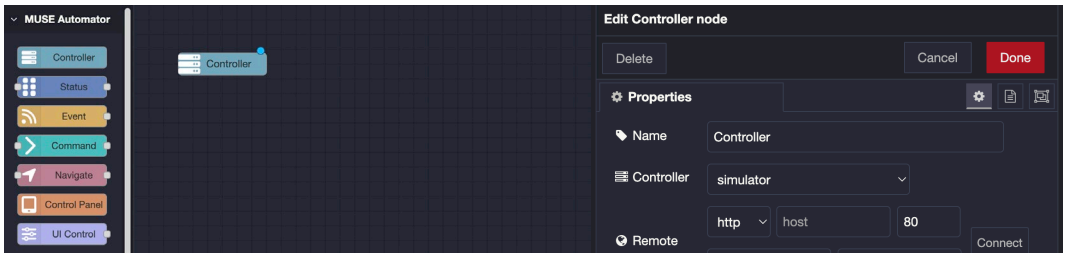
There are three distinct ways of working with Automator. These are not constrictive “modes” per se, but just methods of using Automator. We use the term mode here for simplicity.

1. **Simulation** – Flows are deployed locally and run on a MUSE simulator so you can test without a physical controller.
2. **Connected** – You are connected to a physical MUSE controller and flows are deployed and then run locally on a PC. If you shutdown Automator, the flows will cease to operate.
3. **Standalone** – You have pushed your deployed flows to a MUSE controller to run independently on the controller.

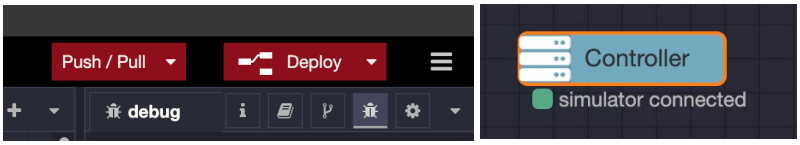
Regardless of which mode you are running, you should know which devices you are intending to control or automate, and then load their respective drivers to either the simulator or a physical controller. The method for loading drivers to either target is very different. Loading drivers to the simulator occurs in the Automator *Controller* node edit dialog (see Adding Drivers & Devices). Loading drivers to a MUSE controller is done in the controller’s web interface. To learn more about loading drivers to your MUSE controller, refer to documentation at https://www.amx.com/products/mu-3300#downloads.

Simulation Mode

To use Automator in Simulation Mode, drag a *Controller* node to the workspace and open its edit dialog. Select *simulator* from the dropdown box and click the Done button. You can now use nodes which can access the endpoints of the simulator device.



Click the Deploy button and you should see the simulator status indicated as connected with a solid green indicator box:



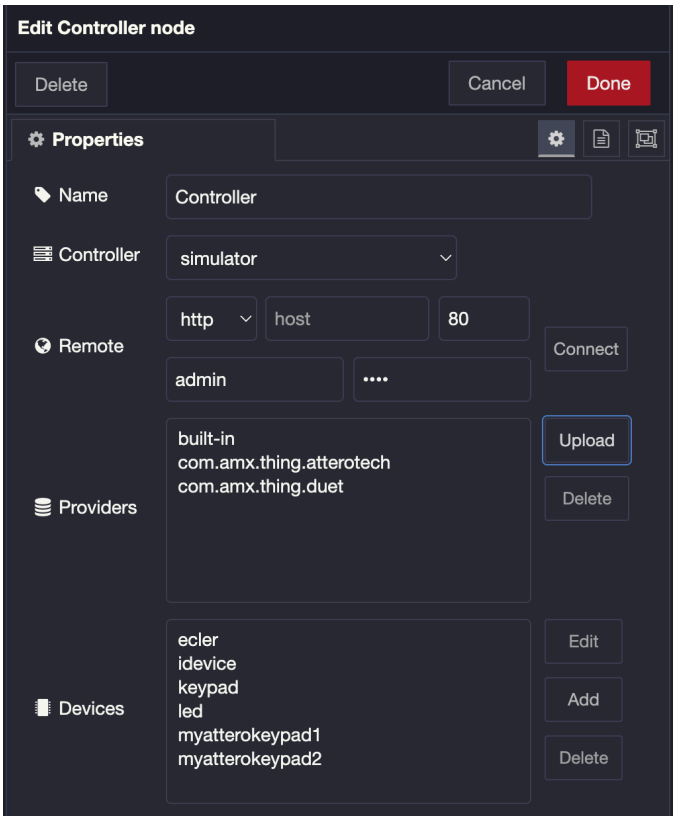
Add Drivers & Devices

There are several simulators already built into the Automator Controller Node:

- CE Series IO Extenders: CE-IO4, CE-IRS4, CE-REL8, CE-COM2
- MU Series Controller I/O ports: MU-1300, MU-2300, MU-3300
- MU Series Controller front panel LED: MU-2300, MU-3300
- A generic NetLinx ICSP device

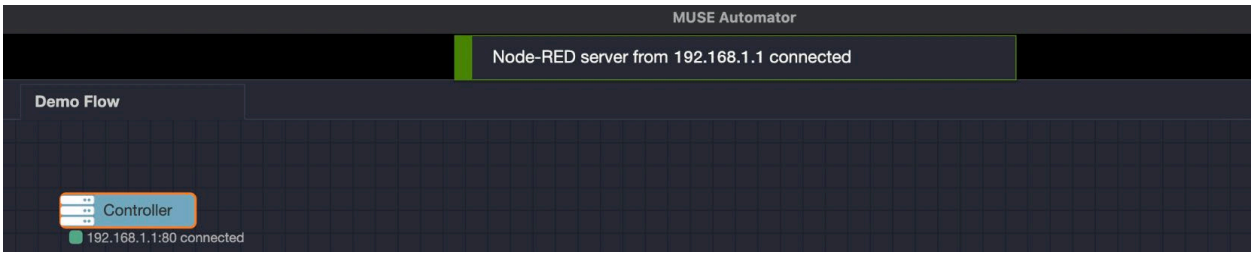
To add devices to your simulator:

1. Click the *Upload* button next to the list of Providers. This will open your file system dialog. Select the corresponding driver for the intended device. Note: the following driver types can be uploaded:
    a. DUET modules (Retrieve from developer.amx.com)
    b. Native MUSE drivers
    c. Simulator files
2. Once the driver has been uploaded, you can add the respective device by clicking the *Add* button next to the Devices list.



Connected Mode

Connected mode requires that you have a physical MUSE controller on your network to which you can connect. Open your *Controller* node and enter the address of your MUSE controller. Port is 80 and set by default. Enter the username and password for your controller and then press the *Connect* button. You should observe a notification that Automator has connected to the Node-RED server on the MUSE Controller. See screenshot below.



Standalone Mode

This mode of working with Automator simply involves pushing your flows from your local PC to the Node-RED server running on a MUSE controller. This requires Projects to be enabled (which requires the installation of git). Read below to learn more about Projects and Push/Pull.

Deploying

Anytime you make a change to a node you will need to deploy those changes from the editor to the Node-RED server to make the flows run. There are some options for what and how to deploy your flows in the Deploy dropdown. To learn more about deploying in Node-RED, please see the Node-RED documentation.

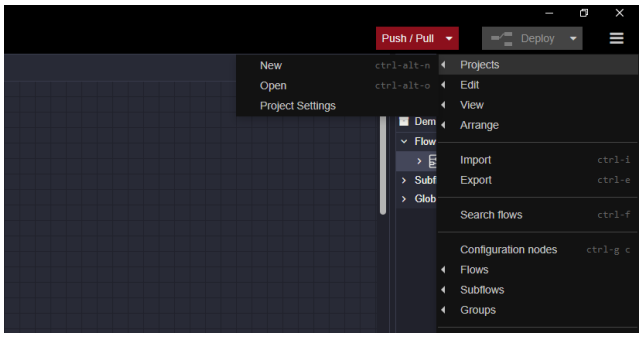
When deploying in Automator, flows are deployed to the local Node-RED server running on your PC. Then, the deployed flows must be "pushed" from your local PC to the Node-RED server running on the MUSE Controller.

A good way to determine if you have any undeployed changes to your flows/nodes is in the *Deploy* button in the upper right corner of the application. If it is grayed out and non-interactive, then you have no undeployed changes in your flows. If it is red and interactive, then you have undeployed changes in your flows. See screenshots below.

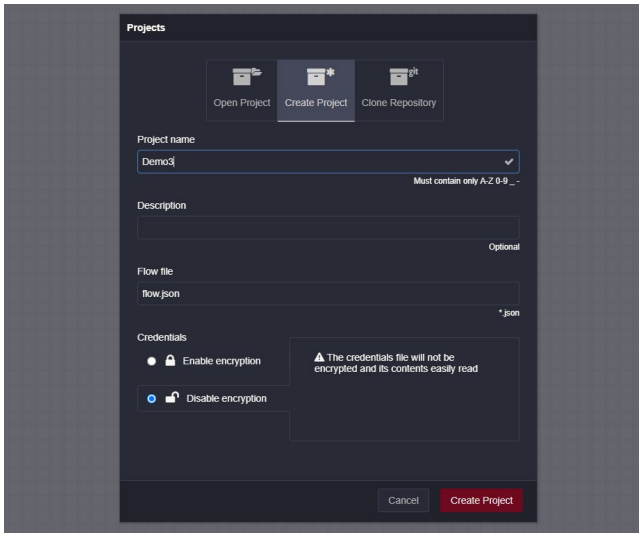
Projects

To Push/Pull from your local Node-RED server to the server running on your controller, the Projects feature needs to be enabled in Automator. The Projects feature is automatically enabled if *git* is installed on your PC. To learn how to install git, see the Install Git section of this guide.

Assuming, you've installed git and restarted MUSE Automator, you can create a new project by clicking the hamburger menu in the upper-right corner of the application.



Enter a project name (no spaces or special characters allowed), and for now, select the *Disable encryption* option under *Credentials.* Press the *Create Project* button to complete project creation.



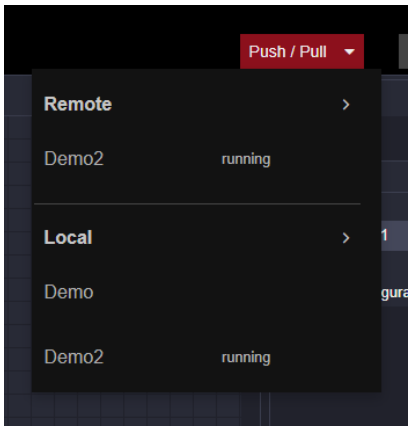
Now that you have created a project, you can Push/Pull to a physical MUSE controller.

Pushing/Pulling Projects

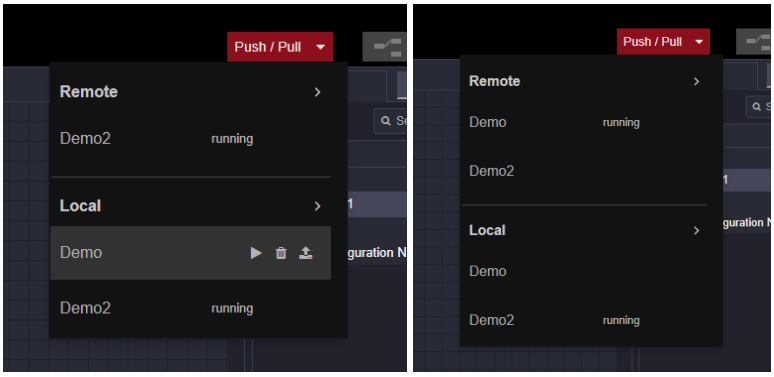
Pushing and pulling your flows from your PC to the Node-RED server on a MUSE controller is a unique feature in Automator. A couple of steps need to be performed before you can Push/Pull

1. Make sure you are connected to your MUSE controller via the *Controller node*
2. Make sure you have deployed any changes in your flows (the *Deploy* button should be grayed out)

To push your deployed flows from your PC, click the *Push/Pull* down arrow.



Hover over the Local project and click the upload icon to push the project from your local Node-RED server to the Node-RED server on your MUSE controller.



After pushing your local project to the controller, press the *Push/Pull* (not the arrow) button and the project should appear to be running on the controller.

In the same way, a project that's been pushed to a controller, can be pulled from the controller to your PC. Hover over the Remote project click the download icon to pull the project.

Run a Project

Projects that are running on the controller or running on your local Node-RED server will be indicated by a label of *running*. To run a different project on either the Remote server or Local server, hover over the project and click on the play icon. Note: only one project can run at a time on Local or Remote.

Delete a Project

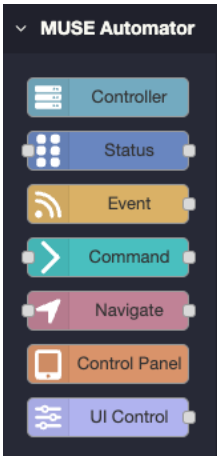
To delete a project, hover over the project name under Local or Remote and click the trash can icon. Warning: be cautious about what you are deleting, or you may lose work.

Stopping a Project

There may be scenarios where you want stop or start an Automator project locally or remotely on the controller. Automator provides the ability to start or stop any project as a needed. To stop a project, click to expand the Push/Pull tray. Hover over any running project in either the Remote or Local list and then click on the stop icon.

MUSE Automator Node Palette

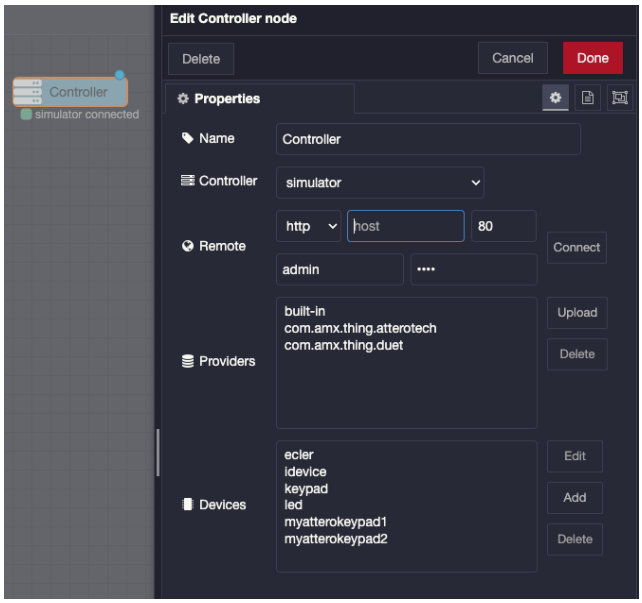
Automator ships with our own custom node palette also titled *MUSE Automator*. There are currently seven nodes provided which enable functionality and interaction with the simulator and MUSE controllers.



Controller

The Controller node is what provides your flows simulator or MUSE controller context and programmatic access to the devices that have been added to the controller. It has the following fields that can be configured:

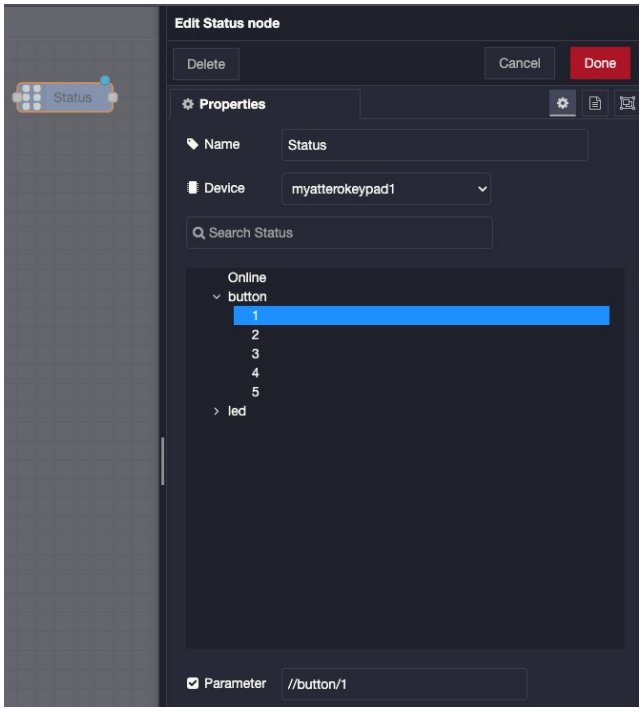
- **Name** – universal name property for all nodes.
- **Controller** – the controller or simulator to which you want to connect. Select *simulator* to connect to the simulated MUSE controller. To connect to a physical controller, make sure it is connected to your network and enter its IP address in the *host* field. Press the *Connect* button to connect to the controller.
- **Providers** – the list of drivers that have been uploaded to your simulator or controller. Press the *Upload* button to add a driver. Select a driver and press *Delete* to delete a driver from the list.
- **Devices** – the list of devices that have been added to the simulator or controller.
    - *Edit* – Select a device from the list and click *Edit* to edit its properties
    - *Add* – Click to add a new device (based on the drivers in the Providers list).
        - *Instance* – When adding a new device a unique instance name is required.
        - *Name* – Optional. Name for the device
        - Description – Optional. Description for the device.
        - *Driver* – Select the appropriate driver (based on the drivers in the Providers list).
    - *Delete* – Select a device from the list and click *Delete* to delete the device.



Status

Use the Status node to get the status or state of a specific device parameter.

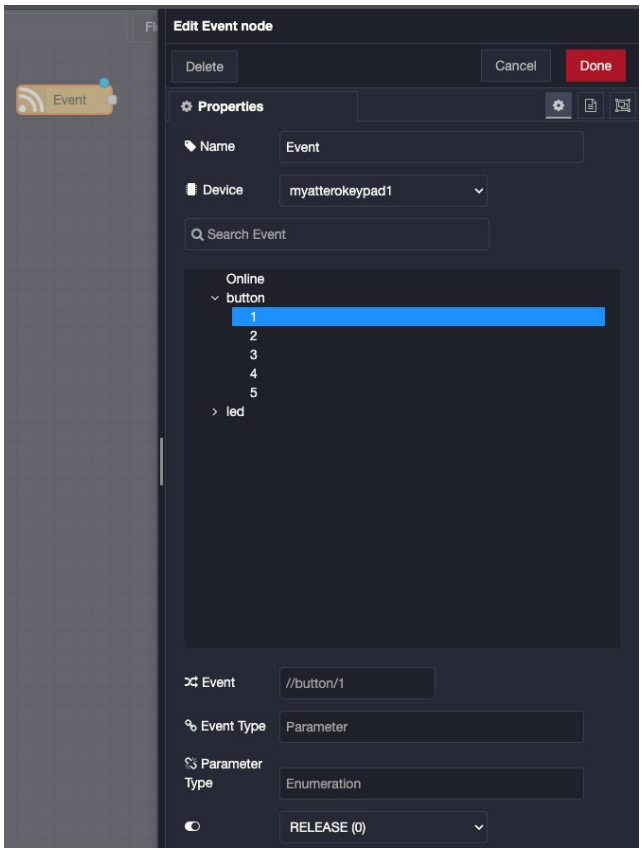
- **Name** – universal name property for all nodes.
- **Device** – select the device (based on the Devices list in the Controller node). This will generate a parameters tree in the list below. Select the parameter for status retrieval.
- **Parameter** – Read-only field which shows the parameter path of the selected parameter.



Event

Use the Event node to listen for device events such as changes in state to trigger an action (such as a command)

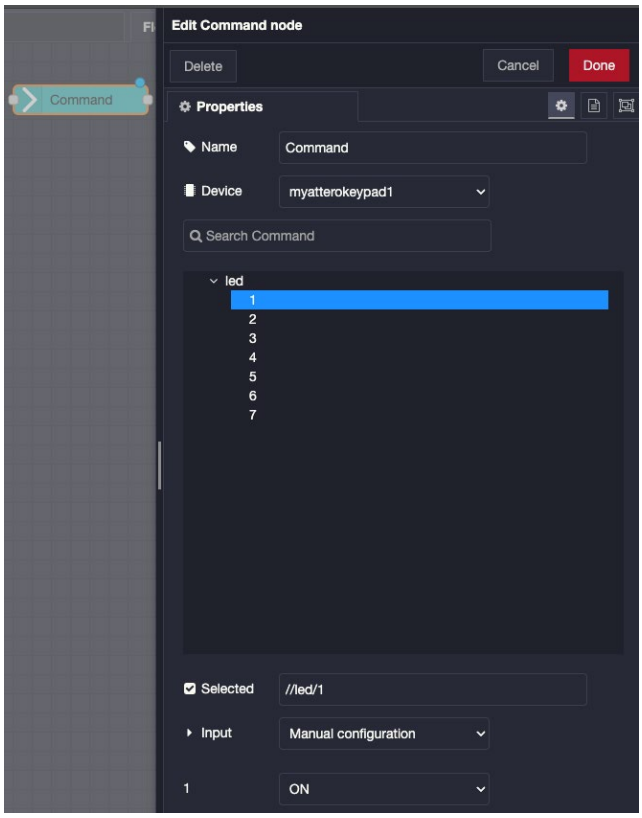
- **Name** – universal name property for all nodes.
- **Device** – select the device (based on the Devices list in the Controller node). This will generate a parameters tree in the list below. Select a parameter from the list.
- **Event** – Read-only field which shows the parameter path
- **Event Type** – Read-only type of the selected parameter event.
- **Parameter Type** – Read-only data type of the selected parameter.
- **Event (unlabeled)** – Dropdown box with the list of events that can be listened for



Command

Use the Command node to send a command to a device.

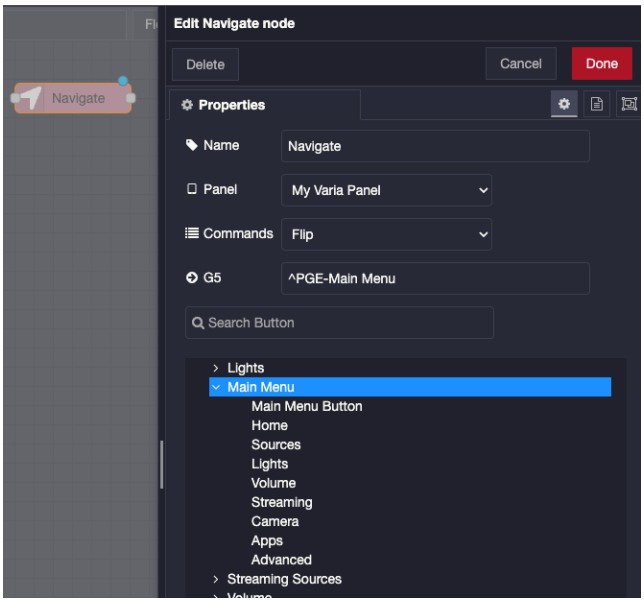
- **Name** – universal name property for all nodes.
- **Device** – select the device (based on the Devices list in the Controller node). This will generate a parameters tree in the list below. Only parameters that can be set will be shown.
- **Selected** - Read-only field which shows the parameter path.
- **Input** – Choose Manual configuration to see the available commands in the dropdown box which can be executed.



Navigate

Use the Navigate node to perform a page flip to a TP5 touch panel

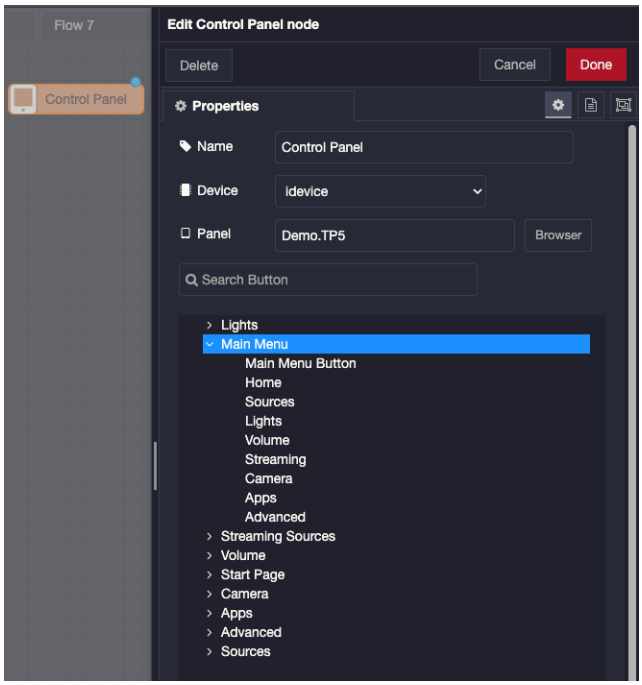
- **Name** – universal name property for all nodes.
- **Panel** – Select the touch panel (added via the Control Panel node)
- **Commands** – Choose the Flip command
- **G5** – An editable string of the command to send. Select the page from the generated list of panel pages to populate this field.



Control Panel

Use the Control Panel node to add touch panel context to the flow.

- **Name** – universal name property for all nodes.
- **Device** – Select the touch panel device
- **Panel** – Click Browse to upload a .TP5 file. This will generate a read-only tree of the touch panel file pages and buttons. Reference this list as verification of the file.

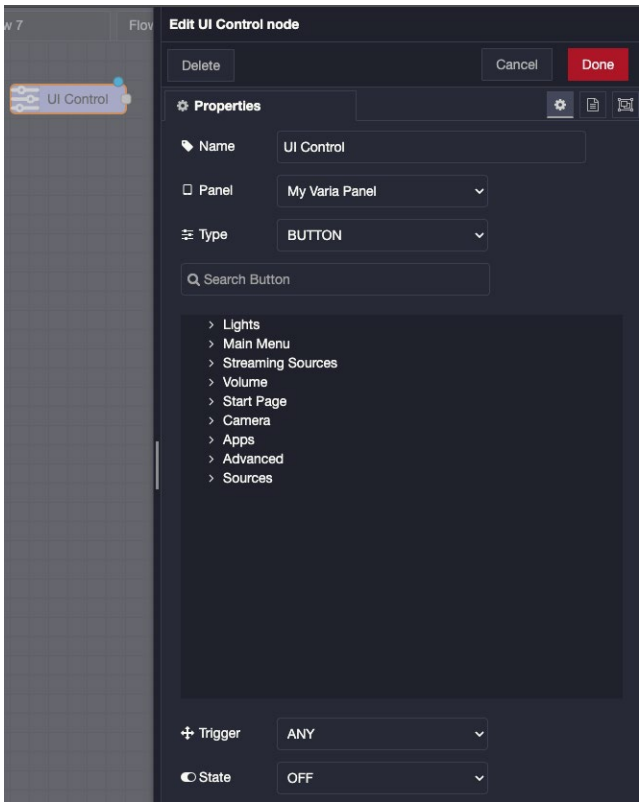


UI Control

Use the UI Control node to program buttons or other controls from the touch panel file.

- **Name** – universal name property for all nodes.

- **Device** – Select the touch panel device
- **Type** – Select the UI control type. Select the UI control from the page/button tree below
- **Trigger** – Choose the trigger for the UI control (for example, PUSH or RELEASE)
- **State** – Set the state of the UI control when it is triggered (for example, ON or OFF)



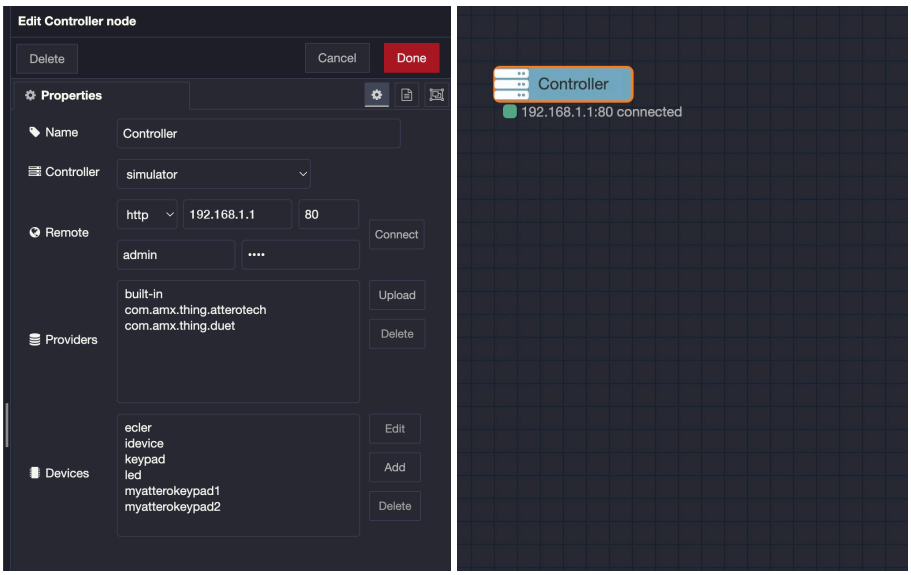
Example Workflow

In this example workflow, we will:

- Connect to a MUSE controller
- Build out a flow that allows us to toggle the state of a relay on a MU-2300
- Deploy the flow to our local Node-RED server

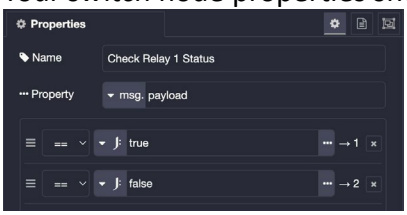
Connect to MUSE Controller

1. Setup your MUSE controller. Refer to documentation at
2. Drag a *Controller* node from the MUSE Automator node palette to the canvas and double click it to open its edit dialog.
3. Input the IP address of your MUSE controller and press the *Connect* button and then the *Done* button. Then press the *Deploy* button. Your dialog and Controller node should look like:



Build & Deploy a Flow

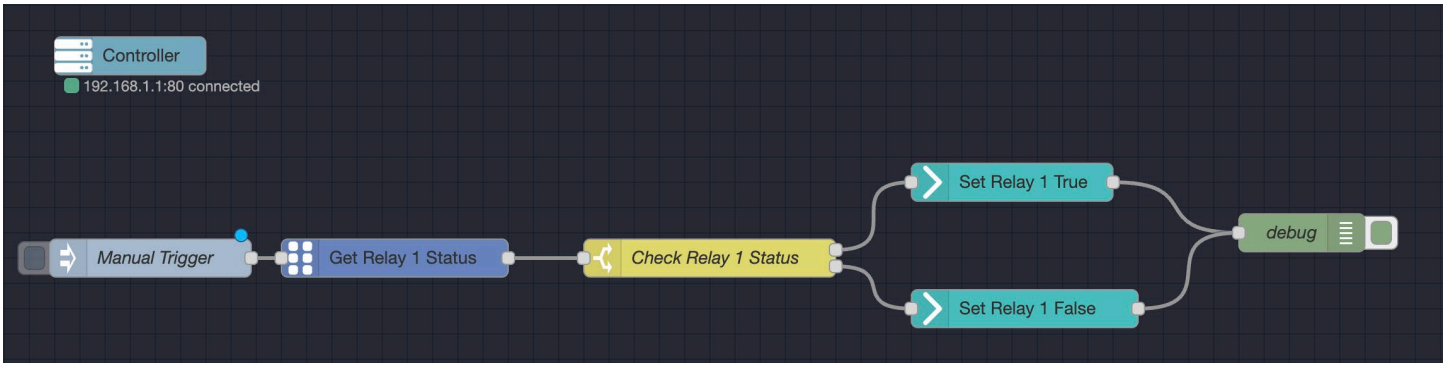
1. Next, let's start building a flow by dragging several nodes to the canvas. Drag the following nodes and place in left to right order:
    a. Inject
    b. Status
    c. Switch (under the *function* palette)
    d. Command (drag two)
    e. Debug
2. Double-click the *Inject* node and change its name to "Manual Trigger" and press *Done*
3. Double-click the *Status* node and modify the following properties:
    a. Change its name to "Get Relay 1 Status"
    b. From the *Device* dropdown, select *idevice*
    c. Expand the *relay* leaf node in the tree and select *1* and then *state*
    d. Press *Done*
4. Double-click the *Switch* node and modify the following properties:
    a. Change the name to "Check Relay 1 Status"
    b. Click the *+add* button at the bottom of the dialog. You should now have two rules in the list. One points to *1* port and two points to *2* port
    c. Type true into the first field and set the type to *expression*
    d. Type false into the second field and set the type to *expression*
    e. Your switch node properties should look like so:



5. Double-click the first *Command* node and modify the following properties:
    a. Change the name to "Set Relay 1 False"
    b. From the *Device* dropdown, select *idevice*
    c. Expand the *relay* leaf node in the tree and select *1* and then *state* then press *Done*

6. Double-click the second *Command* node and modify the following properties:
    a. Change the name to “Set Relay 1 True”
    b. From the *Device* dropdown, select *idevice*
    c. Expand the relay leaf node in the tree and select *1* and then *state* then press *Done*
7. Wire the all the nodes together like so:
    a. *Inject* node to the *Status* node
    b. *Status* node to the *Switch* node
    c. *Switch* node port 1 to the *Command* node named “Set Relay 1 False”
    d. *Switch* node port 2 to the *Command* node named “Set Relay 1 True”
    e. Wire both *Command* nodes to the *debug* node

Once you’ve completed configuring and wiring your node, your flow canvas should look something like so:



You are now ready to deploy your flow. In the upper right-hand corner, of the application click the *Deploy* button to deploy your flow to the local Node-RED server. If you are connected to a MUSE controller, you should now be able to continually press the button on the *inject* node and see the relay state changing from *true* to *false* in the debug pane (and see/hear the relay switching on the controller itself!).

Additional Resources

- AMX YouTube Channel - https://www.youtube.com/@AMXbyHARMAN
- AMX Developer Resources - https://developer.amx.com/#!/main
- Node-RED YouTube Channel - https://www.youtube.com/@Node-RED
- Node-RED Documentation - https://nodered.org/docs/

3000 RESEARCH DRIVE, RICHARDSON, TX 75082 AMX.com | 800.222.0193 | 469.624.8000 | +1.469.624.7400 | fax 469.624.7153
Last Revised: 2024-03-01