



This practical session should be a bit of fun for you.

The objective is to merge together two previous projects to produce a brand new application to control a Neopixel Strip using verbal commands from Alexa.

You might find it useful to do some background reading on the following two projects before tackling this application. Just click on the hypertext links.

- [Controlling a Neopixel Strip](#)
- [Alexa - Turn ON Christmas](#)

Nodes required

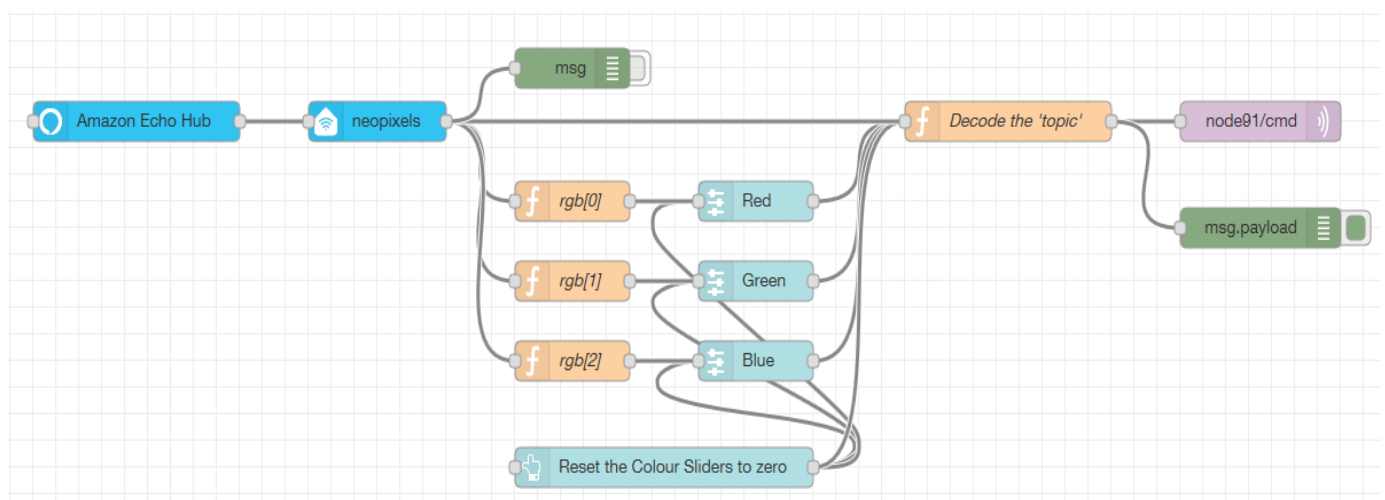
- Function and Debug nodes
- node-red-dashboard: ui_button and ui_slider
- MQTT-Out node
- node-red-contrib-amazon-echo: amazon-echo-hub
- node-red-contrib-amazon-echo: amazon-echo-device

You will probably have to install the Amazon node as well as the dashboard.

Please contact Mr D if you have problems finding or installing these nodes.

Node-RED flow

Using the following screen-shot as a guide create this Node-RED flow.

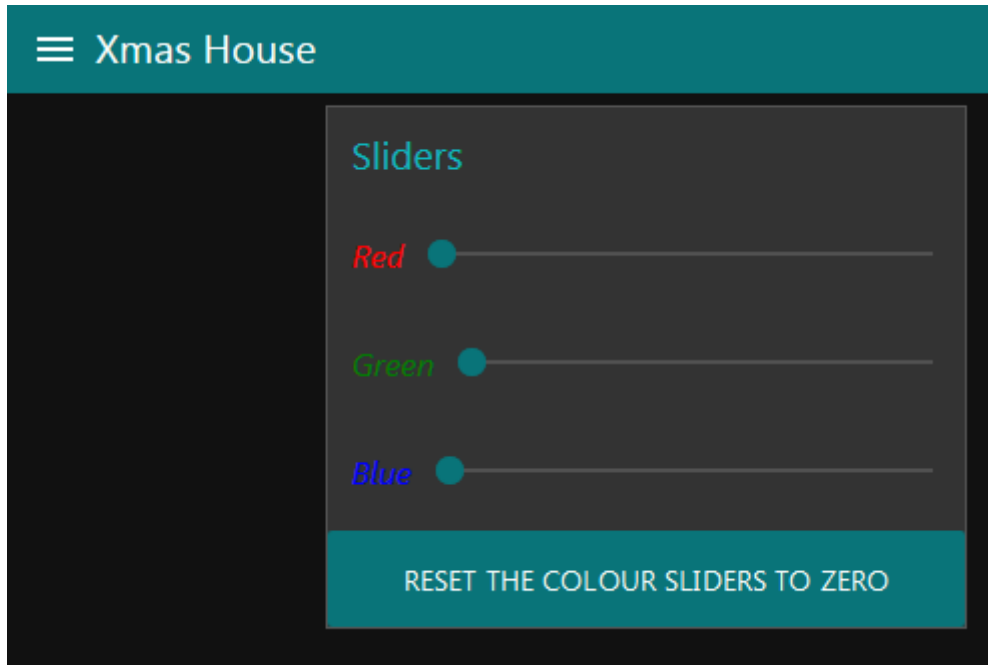


There are two parts to the flow - the Amazon Echo section that receives your verbal commands and the section that reacts to the dashboard buttons/sliders.



Dashboard

Here's a screen-shot of the simple dashboard layout.



If you drag any of the three sliders from the left to the right, the Neopixels with the corresponding colour will gradually increase their brilliance.

I'm sure you can guess what clicking the button labelled 'Reset the Colour Sliders to Zero' will do.

Here's a screen-shot of the properties for the 'red' slider - the other two sliders are the same except for the topic name.

As you can see the payload will vary from 0 to 250 in steps of 10.

Also the topic name is set to 'red'.

If you look at the 'Label' property you will see a piece of HTML code to color and italicise the text string.

Have a quick look at the properties for the other two sliders.

Group	[Xmas House] Sliders
Size	auto
Label	<code><i>Red</i></code>
Tooltip	optional tooltip
Range	min 0 max 250 step 10
Output	continuously while sliding
If <code>msg</code> arrives on input, pass through to output: <input type="checkbox"/>	
When changed, send:	
Payload	Current value
Topic	red
Name	Red



Controlling a Neopixel strip - REVISITED

So to summarise, every time you move the position of a slider a message is sent from the slider to the function node labelled 'Decode the topic'.

The format of these messages is:

msg		Type
.payload	0 ... 255	integer
.topic	"Red" or "green" or "blue"	String

If you look inside the function node you should see the following JavaScript.

```
Name Decode the 'topic'
```

```
Function
```

```
1 var red,green,blue;
2
3 red = flow.get("red")||0;
4 green = flow.get("green")||0;
5 blue = flow.get ("blue")||0;
6
7 if (msg.topic == "red") {
8     red = msg.payload;
9 }
10 else if (msg.topic == "green") {
11     green = msg.payload;
12 }
13 else if (msg.topic == "blue") {
14     blue = msg.payload;
15 }
16 else if (msg.topic == "reset_neopixels") {
17     red = 0;
18     green = 0;
19     blue = 0;
20 }
21 else if (msg.topic == "neopixel_values") {
22     red = msg.rgb[0];
23     green = msg.rgb[1];
24     blue = msg.rgb[2];
25 }
26
27 flow.set("red",red);
28 flow.set("green",green);
29 flow.set("blue",blue);
30
31 msg.payload = "NeopixelAll," + red + "," + green + "," + blue;
32 return msg;
```



If you locate line-7 you will see the JavaScript that tests the topic against the text string “red”. If the test is satisfied, it assigns the payload to the variable named red. Two similar tests are performed on lines 10 and 13 for the topics green and blue.

Lines 27-29 store the values for red, green and blue to a set of flow variables while line 31 formats the payload which is sent to the Neopixel strip via an MQTT-Out node. It should be noted that this project makes use of a Wemos D1 Mini (microcontroller) to drive the Neopixel strip. How to do this is covered in the tutorial guide [Controlling a Neopixel Strip](#).

Here’s a screen-shot of the properties for the ‘rest button’ on the dashboard.

Group [Xmas House] Sliders

Size auto

Icon optional icon

Label Reset the Colour Sliders to zero

Tooltip optional tooltip

Colour optional text/icon color

Background optional background color

☒ When clicked, send:

Payload

Topic reset_neopixels

→ If arrives on input, pass through to output: ☐

Make a note of the value for payload and the topic name. Then go and look at the JavaScript lines 16-19 in the ‘Decode the topic’ function node.

You should be able to work out what the JavaScript does.



Amazon Echo

Here's a screen-shot of the wiring for the Amazon Echo Hub and Device.



Note:

You may need to talk to Mr D about the version of the *Amazon Echo Dot* you plan to use with this flow. Some Echo Dots will only connect if they are on Port 80 and Node-RED is started as the 'root' administrator. Not recommended.

The following commands can be entered in the Raspberry Pi server to forward port 80 to another port (e.g. port 8980) and thus avoid using 'root' on port 80.

Run these commands on your Pi if you're using a WiFi connection.

```
sudo apt-get install iptables-persistent

sudo iptables -I INPUT 1 -p tcp --dport 80 -j ACCEPT

sudo iptables -A PREROUTING -t nat -i wlan0 -p tcp --dport 80 -j
REDIRECT --to-port 8980

sudo netfilter-persistent save

sudo netfilter-persistent reload
```

Run these commands if you're using Ethernet CAT cable

```
sudo apt-get install iptables-persistent

sudo iptables -I INPUT 1 -p tcp --dport 80 -j ACCEPT

sudo iptables -A PREROUTING -t nat -i eth0 -p tcp --dport 80 -j REDIRECT
--to-port 8980

sudo netfilter-persistent save

sudo netfilter-persistent reload
```



Here are two screen-shots showing the properties you need to enter in the Amazon Echo Hub and Amazon Echo Device.

Properties

Port 8980

Enable Input ☒

Properties

Name neopixels

Alexa uses this **Name** to identify y unique for each device. If you have Alexa to discover again.

Topic neopixel_values

Make a note of the topic name: *'neopixel_values'*

You now need to say... "Alexa, discover devices".

After about a minute Alexa should say that 'Neopixels' has been discovered.

Make sure the 'debug' node in the flow is switched-on and then say...

"Alexa, turn on neopixels."

The 'debug' node should show an output that looks like this screen-shot.

Hopefully the strip of Neopixels should be illuminated as a single colour.

The various elements in the JavaScript object indicate the state of the Neopixels.

For example, 'on: true' and 'bri:254' says the Neopixels are on with maximum brightness.

The 'rgb' values indicate the colours that have been switched on.

In this example RED(0) is on maximum (255) while GREEN(1) and BLUE(2) are off (i.e. 0).

Make a note of the topic name.

It should match the property you entered in the Amazon Echo Device node.

```
neopixel_values : msg : Object
  ▼ object
    on: true
    bri: 254
    hue: 0
    sat: 254
    ct: 199
    colormode: "hs"
  ▼ rgb: array[3]
    0: 255
    1: 0
    2: 0
  payload: "on"
  deviceid: "aaffa6a5e5f6b"
  topic: "neopixel_values"
  _msgid: "7b4326b3.e6b768"
```



If you skip back to page-1 and look at the Node-RED flow you should see the output from the Amazon Echo Device node goes to two places.

1. The input pin of the 'Decode the topic' function node.

Line 21 tests for the specific topic name and if it is satisfied, the message payload is sent to the Neopixel strip to illuminate a mixed colour scheme.

2. The input pin of three function nodes (rgb[0], rgb[1] and rgb[2].

Lines 22 to 24 extract the 'red', 'green' and 'blue' elements from the JavaScript object. In this particular case the elements form part of an integer array - this is why the bracket notation is used to isolate each item. The output from each function node becomes the input to the dashboard sliders and forces the sliders to move to a position that matches the colour value.

Dimming the Neopixels

Try giving Alexa the following verbal command and see what happens.

"Alex, set Neopixels to 50"

Although you thought Alexa would set the brilliance of the Neopixels to 50% you probably found out that nothing actually happened - what a shame!!

However, if you look at the 'debug' node you should see something like this:

```
neopixel_values : msg : Object
  ▼ object
    on: true
    bri: 128
    hue: 0
    sat: 254
    ct: 199
    colormode: "hs"
    ▶ rgb: array[3]
    payload: "on"
```

Have a look at the 'bri' entry.

It has a value of 128, which is about half of 255.

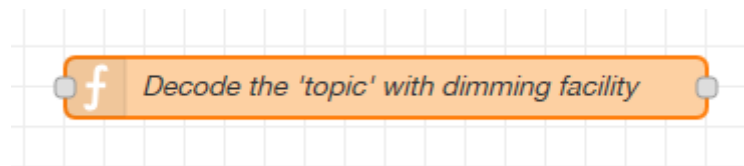
This is indicating the Neopixels should be set to 128/255 roughly 50%.

What we need to do is isolate the 'bri' value and apply some simple maths.



Controlling a Neopixel strip - REVISITED

First of all let's get the Neopixels to dim correctly with this new function node.



The JavaScript is listed below, most of it is the same as the old function node.

```
Name Decode the 'topic' with dimming facility

Function
1 var red,green,blue;
2
3 red = flow.get("red")||0;
4 green = flow.get("green")||0;
5 blue = flow.get ("blue")||0;
6
7 if (msg.topic == "red") {
8     red = msg.payload;
9 }
10 else if (msg.topic == "green") {
11     green = msg.payload;
12 }
13 else if (msg.topic == "blue") {
14     blue = msg.payload;
15 }
16 else if (msg.topic == "reset_neopixels") {
17     red = 0;
18     green = 0;
19     blue = 0;
20 }
21 else if (msg.topic == "neopixel_values") {
22     red = (msg.rgb[0] * msg.bri)/255;
23     green = (msg.rgb[1] * msg.bri)/255 ;
24     blue = (msg.rgb[2] * msg.bri)/255;
25 }
26
27 flow.set("red",red);
28 flow.set("green",green);
29 flow.set("blue",blue);
30
31 msg.payload = "NeopixelAll," + red + "," + green + "," + blue;
32 return msg;
```

Lines 22 to 24 are the ones you need to change.

The maths is fairly simple. It takes the value of msg.bri divides this by 255 and then multiplies the result by each of the rgb values.

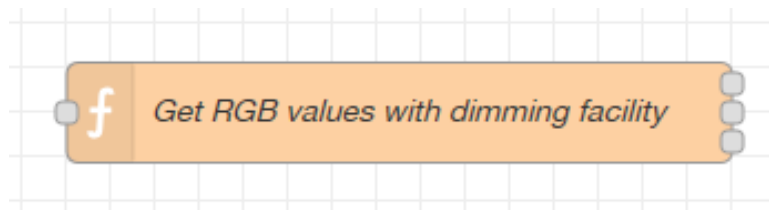


Now let's sort the dashboard sliders.

What we want to happen is when you tell Alexa to set a percentage brilliance, the sliders should move accordingly.

For example if you just had the red Neopixels on full brilliance, the red slider would be at the extreme right of its travels. If you then asked Alexa to set a percentage brilliance of 50%, the slider should move to the centre of its travel.

Here's a new function node to achieve this task.



Notice the node has a single input, but three outputs.

Here's what the JavaScript looks like inside the node.

```
Name Get RGB values with dimming facility

Function
1 var msg1 = (msg.rgb[0] * msg.bri)/255;
2 var msg2 = (msg.rgb[1] * msg.bri)/255;
3 var msg3 = (msg.rgb[2] * msg.bri)/255;
4
5 return [{ payload: msg1},{ payload: msg2}, {payload: msg3}];
```

You should immediately recognise the maths is the same as the previous node.

Line 5 is a bit clever as it outputs each message to a unique output pin.

Just be very careful when you type this in to make sure all the various types of brackets are paired correctly. For example the *return* statement starts with an opening square bracket, therefore it must end with a square bracket.

Have a word with Mr D if you need further explanation.

Node-RED flow in json format

If you get stuck you can visit this [hyperlink](#) and import the json formatted file.

If you intend to build this project and use a Wemos D1 Mini to drive the Neopixels, then you will need to read the other tutorials mentioned in this document to see how to handle the logic interfacing requirements.



Here are some I illuminated earlier

Three examples of solid colours.



Power supply requirements

***** BE VERY CAREFUL ***** the NeoPixel strips run off of +5V, so you need to ensure you have a power supply capable of supplying sufficient current to power the WeMos D1 Mini AND the NeoPixel strip.

Here's a photo of the eight position NeoPixel strip I've been using.



According to the specification sheet, each LED can consume 20mA at maximum brightness. That means 60mA per NeoPixel position (as there are three LEDs at each position). On my NeoPixel strip there are eight positions which will give a possible total current consumption of 480mA (assuming all NeoPixels are switched on at maximum brightness) - nearly half an Amp !!!!

You may find it easier and SAFER to power the NeoPixel strip from a completely separate and independent power supply. If you do this, then just make sure you connect the grounds together. See Mr D for more details.

Please get Mr D to check your wiring BEFORE you apply power.

Congratulations you now know how to control NeoPixels.