Internet of Things at Bohunt School (Wokingham)

Controlling a Neopixel strip

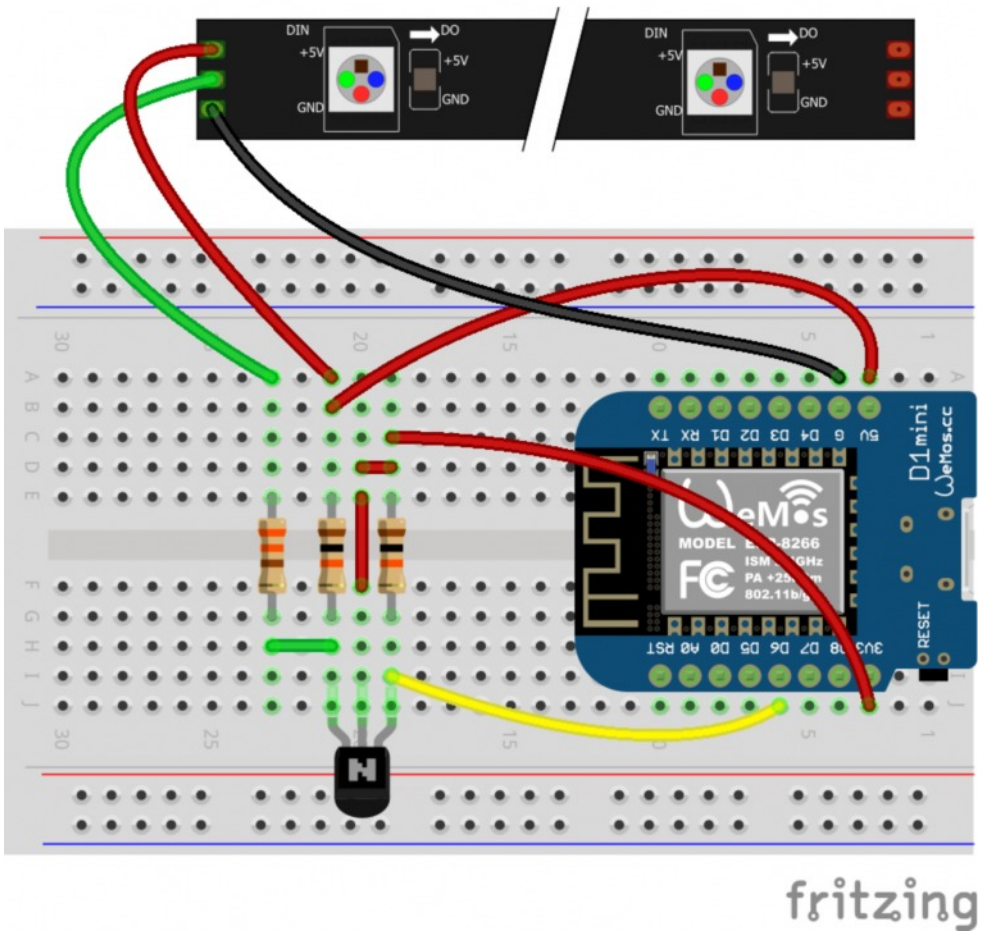
This practical session should be a bit of fun for you.

The objective is to connect a NeoPixel strip to a WeMos D1 Mini and write a Node-RED flow to create attractive and interesting light-sequences.

Building your node

Collect a breadboard fitted with a Wemos D1 Mini, a 2N7000 Field Effect Transistor (FET), a couple of 10K ohm (Brown, Black, Orange) and one 330 ohm (Orange, Orange, Brown) resistors, and some connecting wire from Mr D.

Using this image as a guide wire up your node.



It should be noted that the NeoPixel strip needs +5V and ground. The Data-In connection uses 5V logic-levels - hence the use of the level-shifter circuit to convert +3.3V signal from the WeMos to +5V for the strip.

You may find you can connect the D6 pin (GPIO-12 data output) directly to the Data-In pin of the NeoPixel strip (and dispense with the level shifter).

More information about power supply requirements appear on page-10.

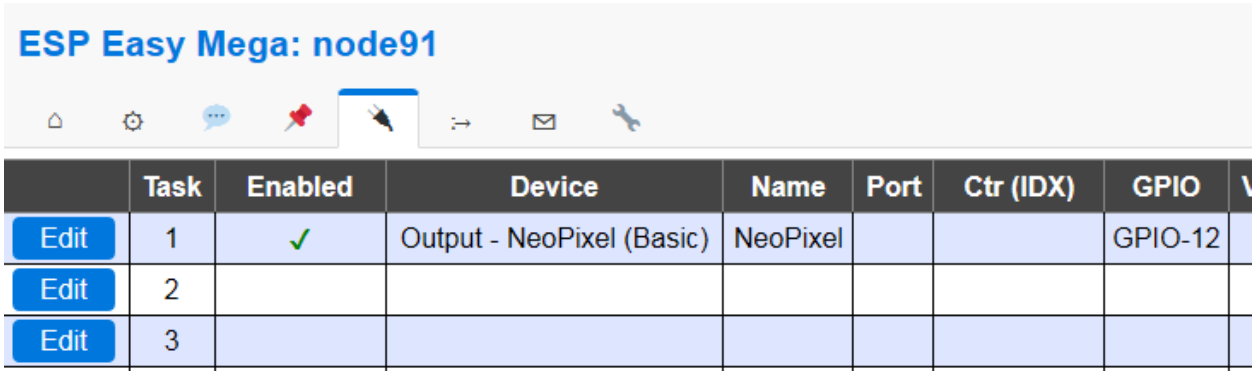
Controlling a Neopixel strip

Configuring the WeMos D1 Mini

The first thing you need to do is set-up the device drivers for your node.

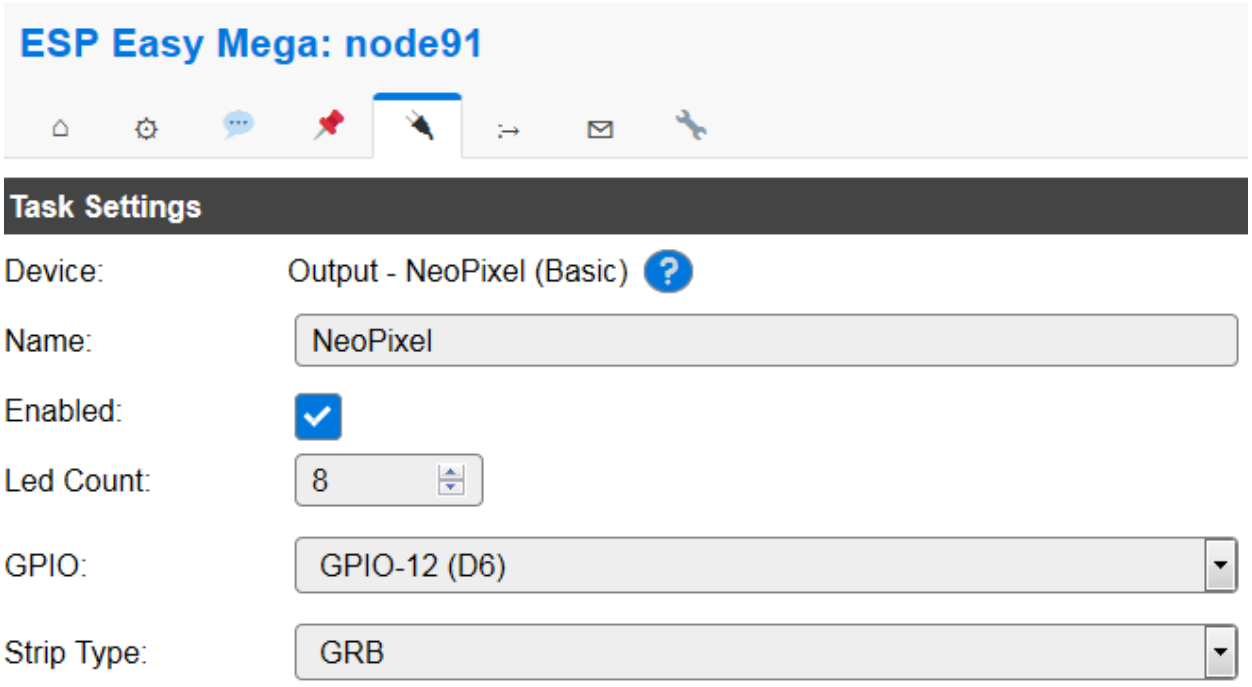
Enter the IP address for your node into a web browser. The node I used was 'node 91', so the IP address I entered was: 192.168.1.91

This action will bring up the web management screen for the WeMos D1 Mini.



Click the 'Devices' tab and select NeoPixels from the drop-down menu.

Then using this screen-shot as a guide, fill-in the 'Task Settings'



Make sure the 'Led Count' matches the number of NeoPixels in the strip.

Also make sure the 'GPIO' value matches the wiring you performed on page-1.

The NeoPixel strip you will be using has Red-Green-Blue (RGB) leds, so make sure 'Strip Type' is set to GRB. GRBW is for NeoPixel strips that have a fourth white coloured led. If you make the wrong selection the NeoPixel won't work!

Internet of Things at Bohunt School (Wokingham)

Controlling a Neopixel strip

NeoPixel command

The command to operate the NeoPixel strip is:

NeoPixel, <position>, <red value>, <green value>, <blue value>

The value for <position> can vary from 1 to 8 (for a NeoPixel strip of 8 leds)

The <colour value> can vary from 0 to 255 for any of the three colours.

Notice where the commas occur in the command - as that's important.

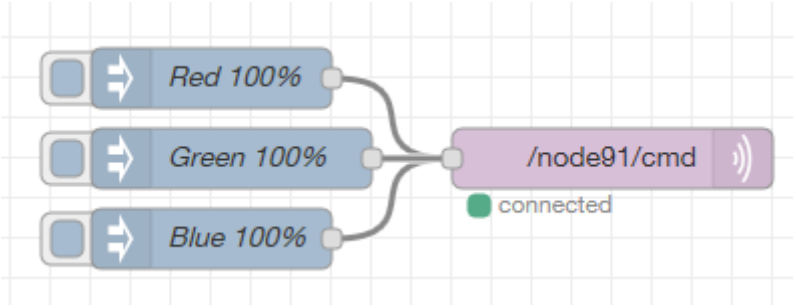
Here's an example to set the first NeoPixel to maximum red colour.

NeoPixel, 1, 255, 0, 0

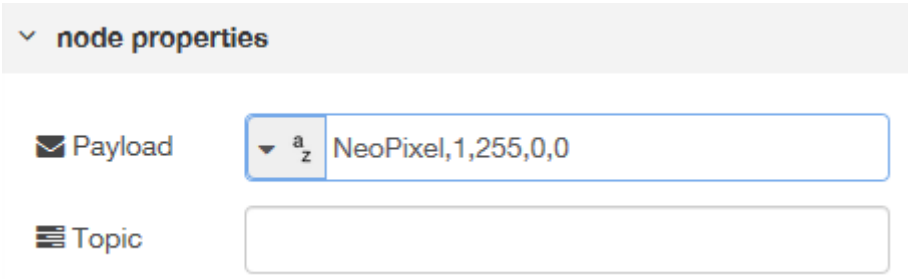
The best way to send a command to the NeoPixel strip is to set-up a 'topic' in a 'MQTT Output' node and then send a message-payload to it.

Turning a single NeoPixel on and off

Using this image as a guide, construct this basic Node-RED flow.



The flow uses a set of 'inject' nodes to send a message payload to the 'MQTT' node. This is what the settings for the first 'inject' node look like.

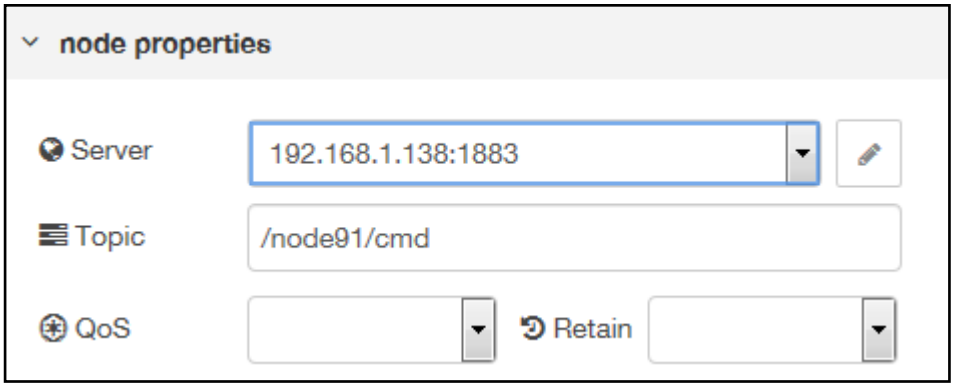


You can see the format is exactly the same as the command at the top of this page.

You should be able to set-up the other two 'inject' nodes that define the settings to inject 100% green colour and 100% blue.

If you get stuck please ask Mr D for assistance.

Next, double-click the 'MQTT Output' node and check/enter these values.



Check the 'Server' setting - making sure it matches your Raspberry Pi.

Enter /node91/cmd as the 'Topic' for the node.

Change the /node91 part to match the WeMos D1 Mini you are using.

You should be able to click the various 'inject' nodes and see what happens on the NeoPixel strip.

You will probably find the lights are very BRIGHT.

Go back to the 'inject' nodes and alter the brightness values.

For example, set the red colour to a lower brightness value of 20.

NeoPixel, 1, 20, 0, 0

Try setting brightness values for the other NeoPixels.

You could also try 'mixing' the brightness values for the red, green and blue leds together to obtain different colours.

For example, tryout *NeoPixel, 1, 40, 0, 40 and see what colour you get*.

Scaling a percentage to a brightness value

As mentioned at the top of page-3, the brightness value can range from 0 (which means no light at all) to 255 (which means maximum brightness).

Although these values are easy to understand, they are not very 'user friendly' unless you are very good and very quick at maths.

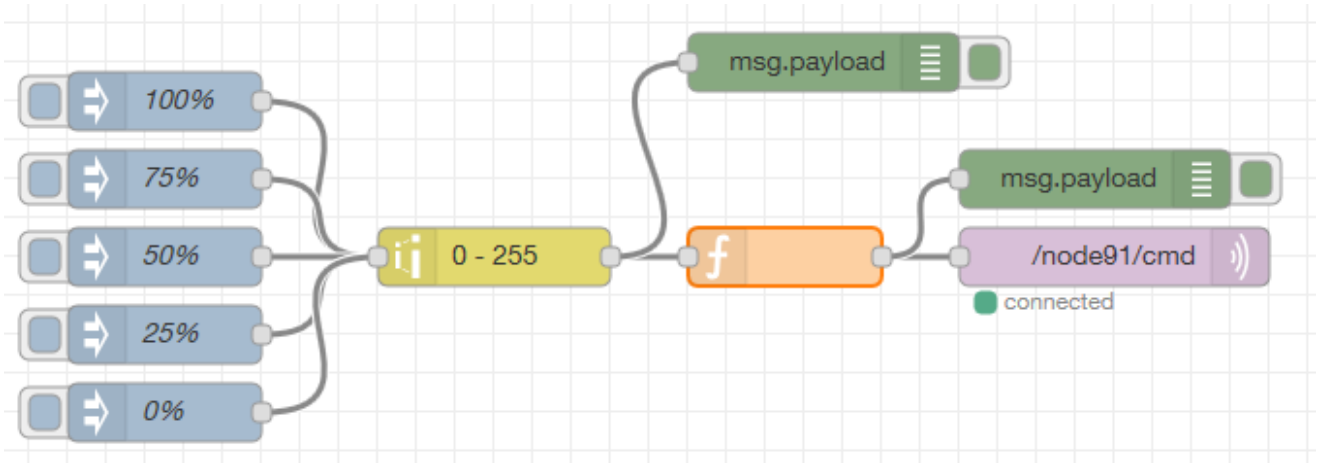
It would be much nicer and easier if the brightness could be expressed as a percentage. It would be very obvious if the brightness was 0% or 100%. I think you could relate to 25%, 50% or even 75% brightness levels.

Internet of Things at Bohunt School (Wokingham)

Controlling a Neopixel strip

There is a very useful node called 'range' (that can be found in the 'function' category of the Palette) that will map one range onto another. This will be ideal for what we want to do in converting a percentage to a brightness value.

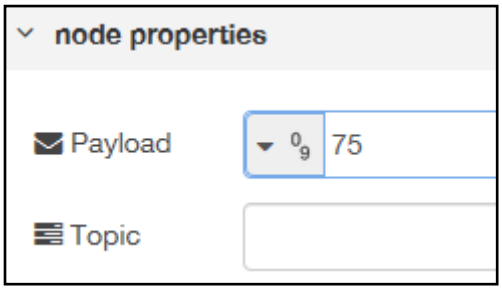
Using this screen-shot as a guide, construct this Node-RED flow.



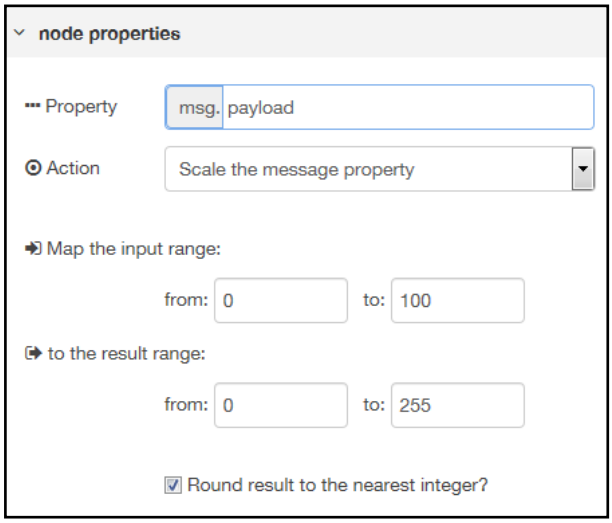
The 'inject' nodes have been changed to inject a numerical value.

Here's an example for the 75% node.

You can work out the others.



These are the settings for the 'range' node.



As you can see, the input range is 0 to 100 (i.e. our percentage values) and the result range is 0 to 255 (which matches the RGB brightness values).

Let's have a look at the contents of the 'function' node as shown below.

```
Function
1  msg.payload = "NeoPixel,1,"+msg.payload+",0,0";
2  return msg;
```

On Line-1 the JavaScript just takes the incoming message (msg.payload) (which represents the brightness value for 'red') and inserts it into the command string. Then sends the message on to the 'MQTT Output' node.

A couple of 'debug' nodes have been added so you can see the brightness value and the format of the command sent to the 'MQTT output' node.

Try clicking the various 'inject' nodes and see what happens on the NeoPixels.

Summary of what you have achieved so far

OK, so you tried out on page-3 setting a certain NeoPixel to maximum brightness for the red, green and blue colours. Then on page-4 you altered the brightness levels. Finally on page-5 you discovered how to use the 'range' node to convert a percentage value to a brightness value (i.e. 0 to 255).

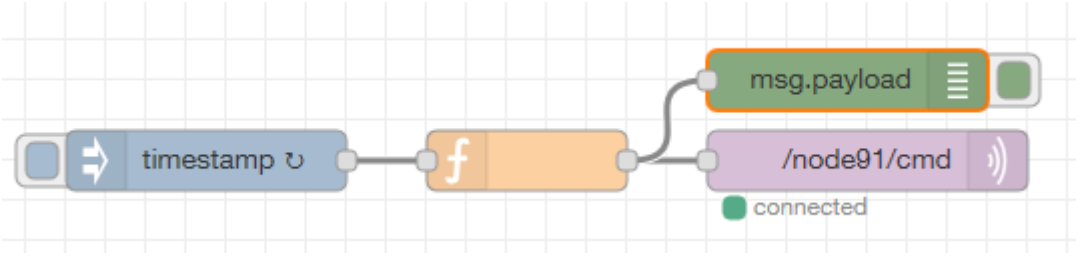
The next thing to do is put some animation into the NeoPixels.

Animating the NeoPixel strip

If you sit back and think what you have achieved so far, you will soon realise that there are numerous visual patterns you can make.

Let's start off by getting the NeoPixels to ripple along the strip with a single colour (say blue). So it doesn't hurt your eyes let's set the brightness at 20%.

Using this image as a guide, build this simple Node-RED flow.



The 'inject' node simply triggers the 'function' node every second.

All of the "magic" is in the 'function' node - shown on the next page.

Controlling a Neopixel strip

Here's a screen-shot of the JavaScript code inside the 'function' node.

Function

```
1  var pos = flow.get("pos") || 0;
2
3  if (pos >=8)
4  {
5      pos = 1;
6  }
7
8  else
9  {
10     pos = pos + 1;
11 }
12
13 flow.set("pos", pos);
14
15 msg.payload = "NeoPixelAll,0,0,0";
16 node.send(msg);
17
18 msg.payload = "NeoPixel," + pos + ",0,0,51";
19 node.send(msg);
20
```

Line-1 reads the value of the flow-variable "pos" and assigns it to a local variable also called 'pos'. (Really hope that doesn't confuse you too much!!)

Lines-3 to 11 check the value of 'pos' and either reset it to 1 if it is 8 or just increment it by unity.

Line-13 writes the new value of 'pos' back to the flow-variable.

Lines-15 and 16 sends a 'clear' command to turn all the NeoPixels off.

Lines-18 and 19 sends a command to turn-on the NeoPixel at position 'pos'.

Try out this JavaScript code and watch the blue NeoPixels flash in turn.

You could try out other values for brightness and/or colours.

Changing colours and position

The next task is to create an animation where the position and colour of the NeoPixels change.

All you need to do is substitute the code in the 'function' node for this:

```
var pos    = flow.get("pos");
var colour = flow.get("colour");

if (pos >= 8)
{
    msg.payload = "NeoPixelAll,0,0,0";
    node.send(msg);
    pos=1;
    if (colour == "red")
    {
        colour = "green";
    }
    else if (colour == "green")
    {
        colour = "blue";
    }
    else if (colour == "blue")
    {
        colour = "red";
    }
}
else
{
    if (colour == "red")
    {
       msg.payload = "NeoPixelAll,0,0,0";
       node.send(msg);
       msg.payload = "NeoPixel," + pos + ",20,0,0";
       node.send(msg);
    }
    else if (colour == "blue")
    {
        msg.payload = "NeoPixelAll,0,0,0";
        node.send(msg);
        msg.payload = "NeoPixel," + pos + ",0,0,20";
        node.send(msg);
    }
    else if (colour == "green")
    {
        msg.payload = "NeoPixelAll,0,0,0";
        node.send(msg);
        msg.payload = "NeoPixel," + pos + ",0,20,0";
        node.send(msg);
    }
    pos = pos + 1;
}
flow.set("pos",pos);
flow.set("colour",colour);
```

OK, I know that piece of JavaScript is rather long and looks very complicated.

Basically all it does is increment the variable called 'pos' from 1 to 8.

When it reaches 8 it is reset to 1 and at the same time the variable called 'colour' is changed from 'red' to 'green' to 'blue'.

Please have a chat with Mr D who can explain the JavaScript to you.

Going off-road

If you would like a bit of fun why not put a strip of NeoPixels in a model and create some spectacular illumination sequences?

Here's model of a house that has a Christmas theme - it is crying out for some really nice illuminations. Can you help?

I'm sure you can have a great deal of fun creating all sorts of light sequences.

Here are some I made earlier

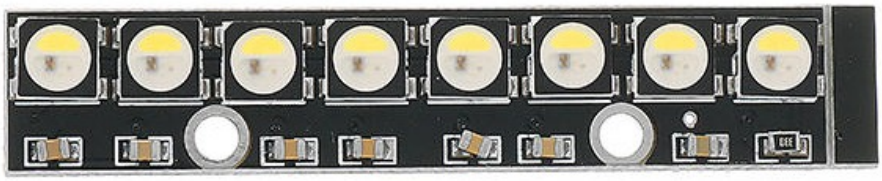
Three examples of solid colours.



Power supply requirements

As mentioned on page-1, 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.