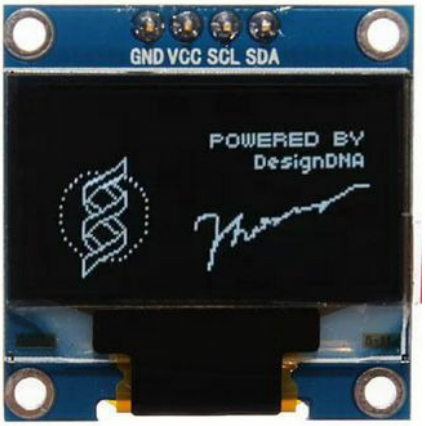
Bohunt School *(Wokingham)*

Internet of Things (IoT) and Node-RED

This practical session should be a bit of fun for you. It involves adding an OLED display panel to the SRF05 ultrasonic distance sensing device.

Organic Light-Emitting Diode (OLED) display panel

Here's a photo of a typical miniature OLED panel.
The one shown here uses the SSD1306 driver chip.
It has a screen resolution of 128 x 64 pixels.



The main advantages of an OLED are:

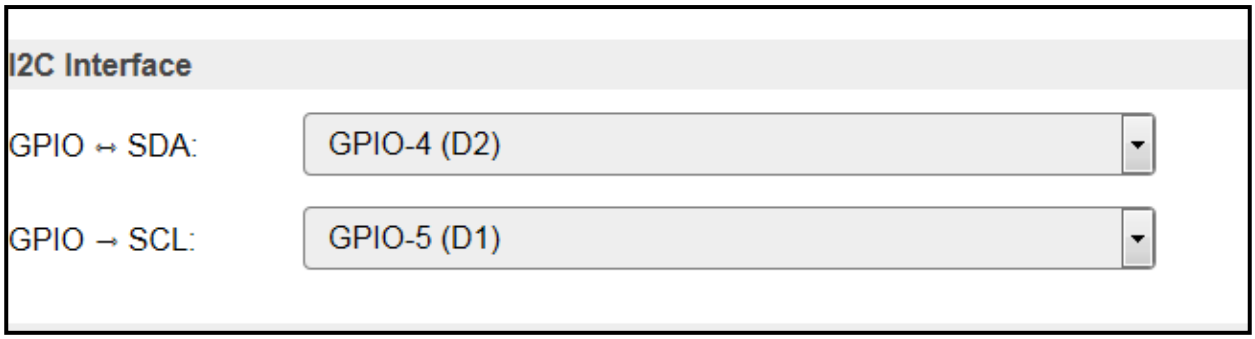
- **works without a backlight**
- **can display deeper blacks**
- **thinner and lighter than an LCD panel**

The one shown here has a 4-wire, I2C interface.

The connections are: Ground and VCC (+3.3v to +5v), Serial Clock (SCL) and Serial Data (SDA). The SCL signal is unidirectional, the SDA is bidirectional.

Setting-up I2C on your WeMos D1 Mini

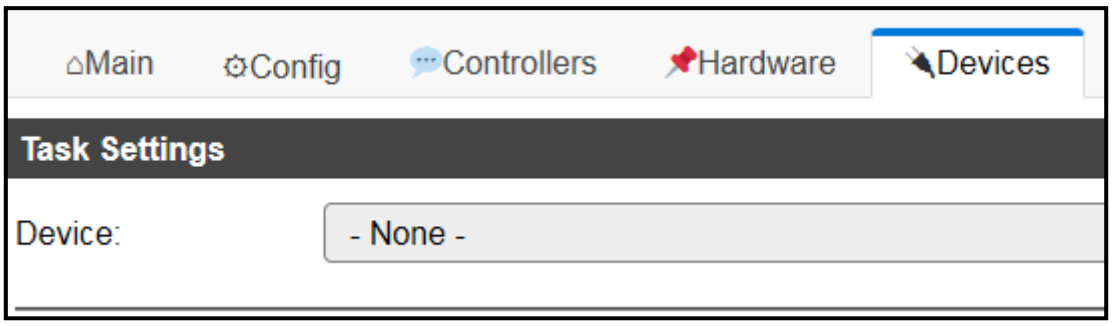
Login to your WeMos D1 Mini and select the 'Hardware' tab to reveal this page.



Set GPIO-SDA to GPIO-4 (D2) and GPIO-SCL to GPIO-5 (D1) then click Submit.

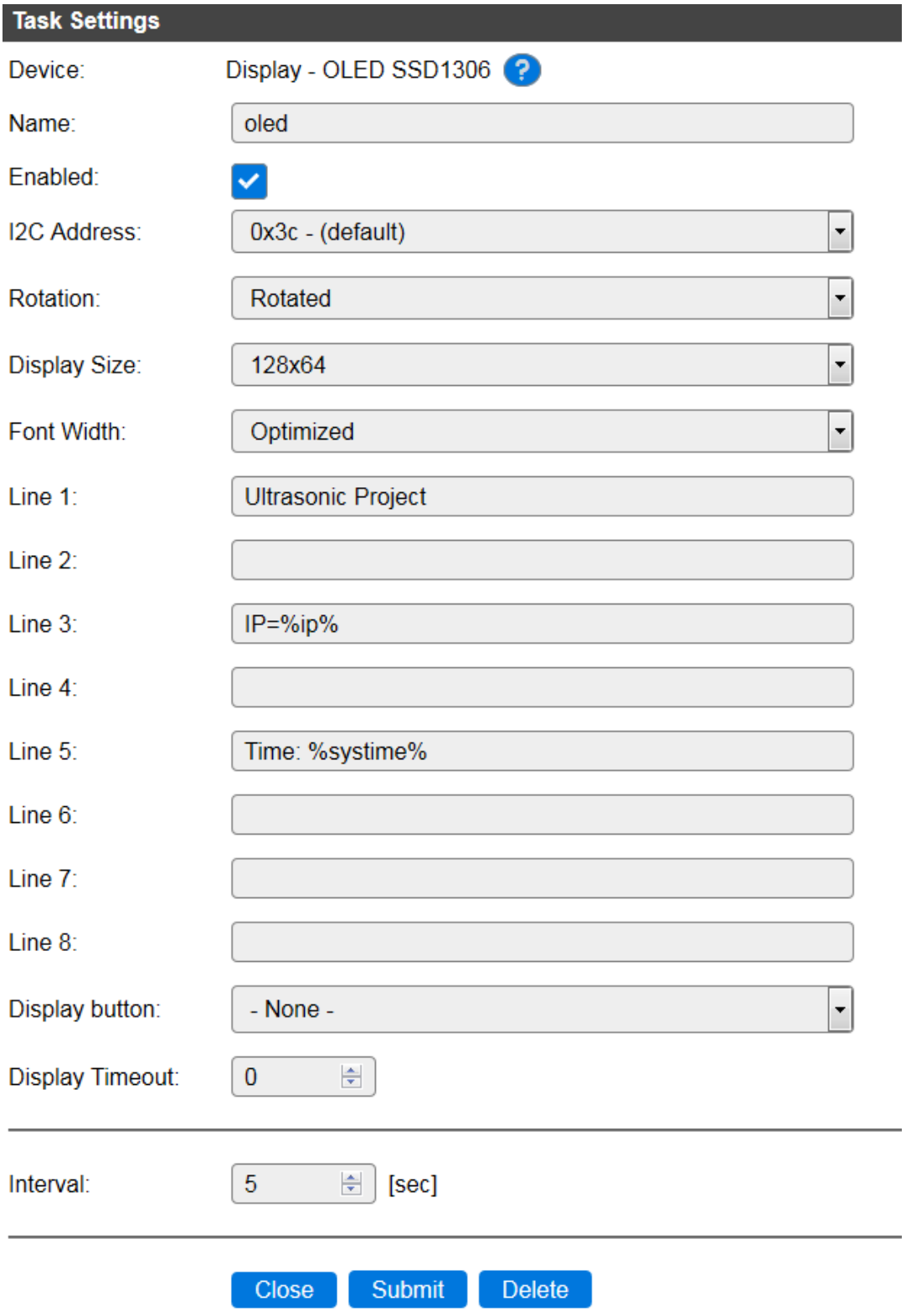
Setting-up the OLED device drivers on your WeMos D1 Mini

Login to your WeMos D1 Mini and select the 'Devices' tab to reveal this page.



Scroll down until you find 'Display - OLED SSD1306'. Click it to select it.

Fill-in the details as shown below, then click 'Submit'.

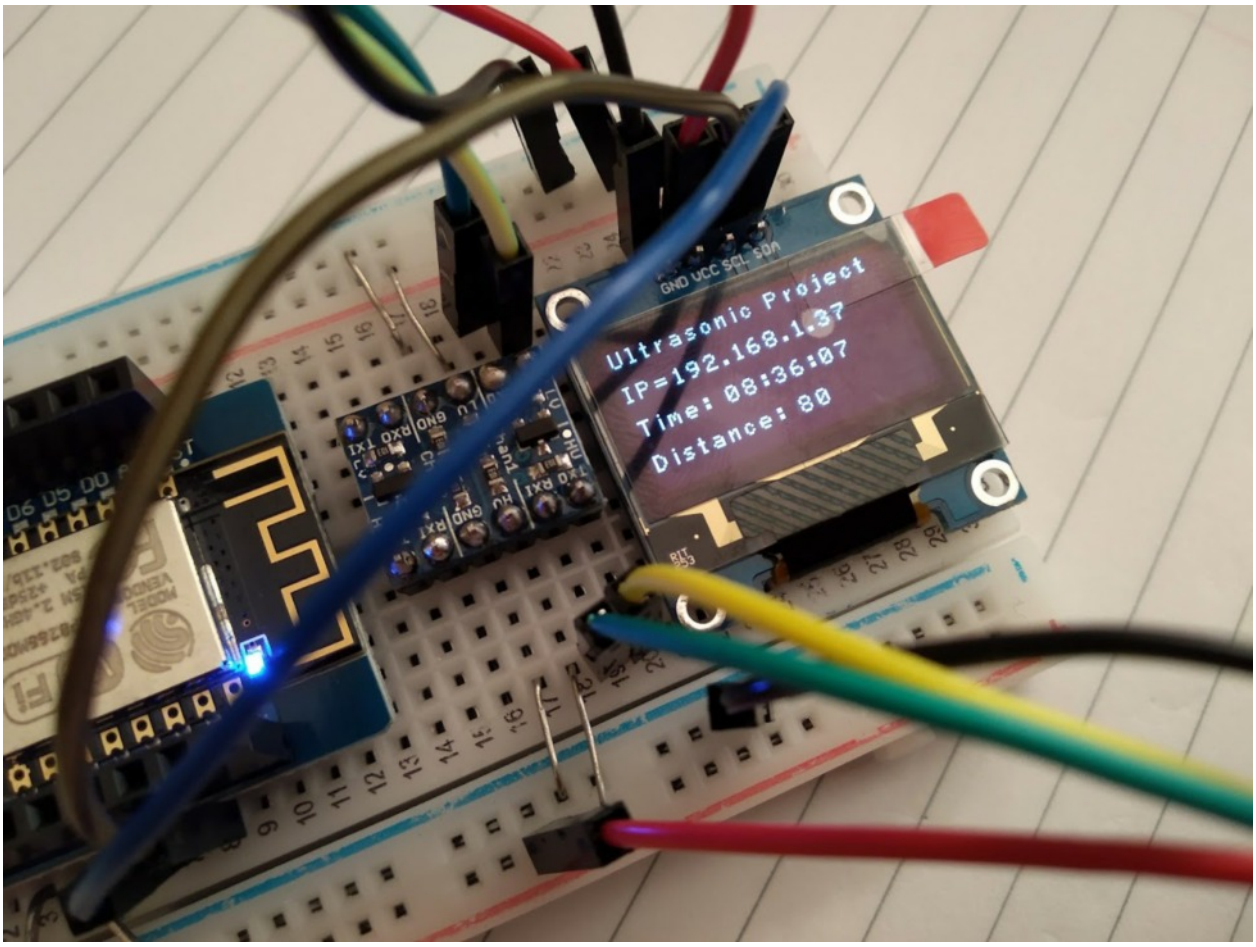


The SSD1306 usually has a default address of 0x3c for the I2C interface.

Your device might have a different address - check with Mr D if you need help in sorting this out.

Wiring your WeMos D1 Mini to the OLED panel

If you look back to Page 1 you will see you defined the hardware connections for the I2C device, so that SCL connects to D1 and SDA to D2.



The above photo shows the SCL and SDA connections you need to implement.

Static and Dynamic Text

If you look at Line 1, on the previous page, you can see the entry is a piece of static text - which means it is never going to change.

Line 3 is a mixture of static and dynamic text. The text that says “IP=” is static while the entry %ip% means lookup the IP address for this node and display it here. So if your node was node37 then 192.168.1.37 would be displayed.

Line 5 displays a dynamic piece of text (%systime%) which is the current time.

Some of the lines are currently not used - lines 2, 4, 6, 7 and 8.

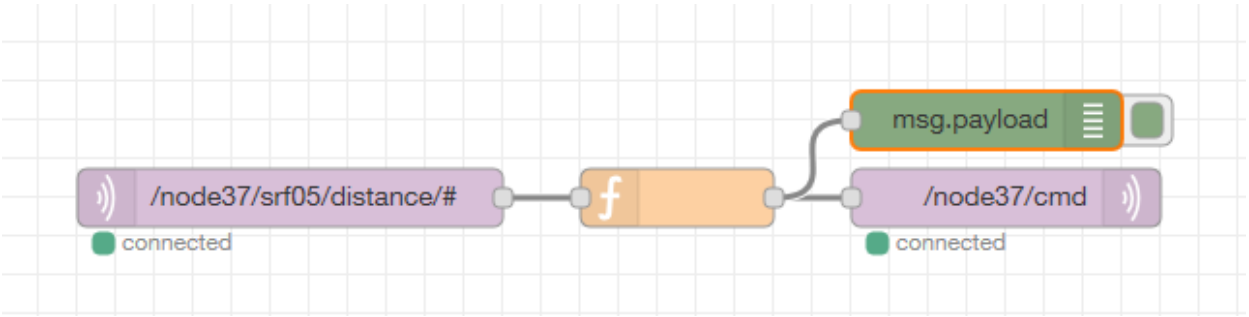
Getting messages/data on the OLED display panel

The resolution of the OLED is 128 by 64 pixels which means you can display 8 lines or rows of characters. The description above explained how you could put static information and/or system variables on the display.

If you want to display a true variable then this can be achieved using the MQTT Output node (that you have probably used many times already).

Let's assume you want to display a distance reading to an object on the OLED.

Here's a piece of Node-RED flow that would achieve this requirement.



The MQTT Input node filters out the 'distance' value which is then passed as an input to a function node as the 'msg.payload' message.

Function

```
1  var distance = msg.payload;
2
3  msg.payload = "oled,7,1,Distance: "+distance;
4  return msg;
5
```

This piece of JavaScript transforms the message by adding the text "oled, 7,1,Distance:" and then appending the 'distance' variable to the string. This is then sent to the 'MQTT Output' node as /node37/cmd.

The final result is a command that tells the OLED panel to display the 'distance' value on row 7.

Notes:
(1) The OLED SSD1306 driver chip references the rows as row1 to row8.

(2) In early versions of ESP Easy, the first column had a reference of '1'.
In recent versions of ESP Easy, the first column has a reference of '0'.

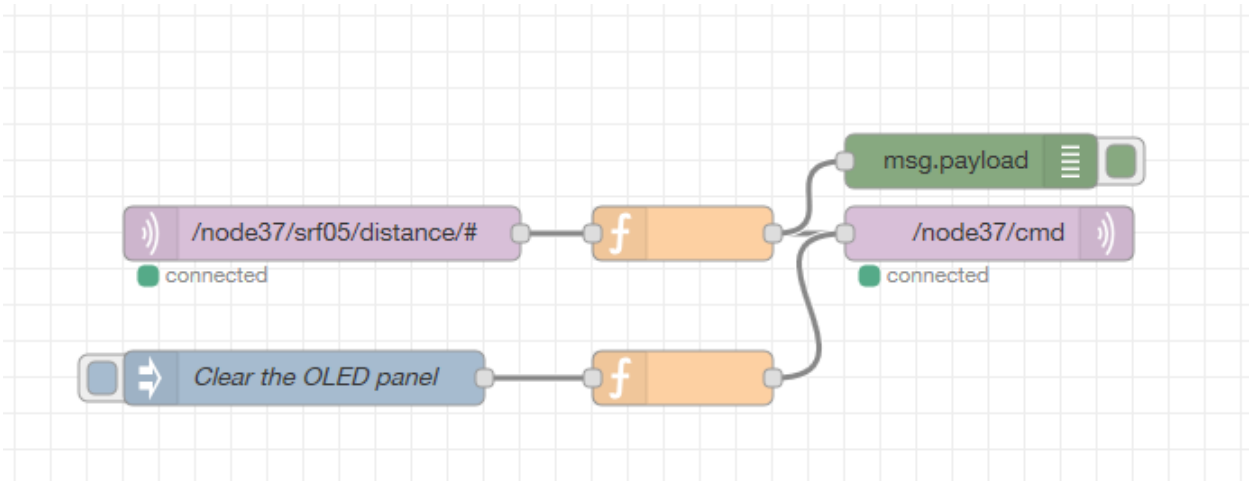
This means you may have to adjust the values in the above 'function' node.

E.g. line 3 msg.payload = "oled, <row>, <column>, <text>;

Sending a command to the OLED panel

As well as sending text to the OLED panel, commands can be sent as well.

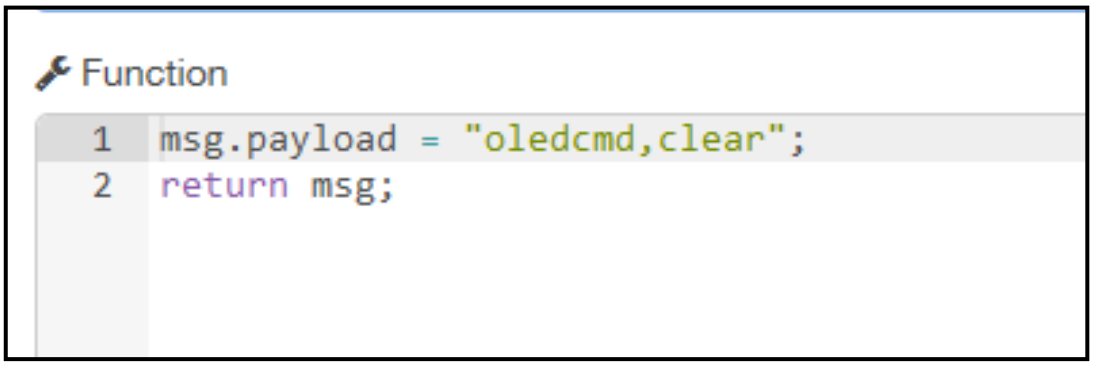
Here's a piece of Node-RED flow that takes input from an 'Inject' node and sends it to a 'function' node that transforms the message into a command.



This is a very simple 'function' node that sets the msg.payload to a text string:

"oledcmd,clear"

which will clear the contents currently displayed on the OLED panel.



Break-Out time

I'm sure you can think of loads of additional things (i.e. text messages) you could display on the OLED panel.

Congratulations you now know how to drive an OLED display panel.