

**Practical – 8****Node RED: Connect LED to Internet of Things**

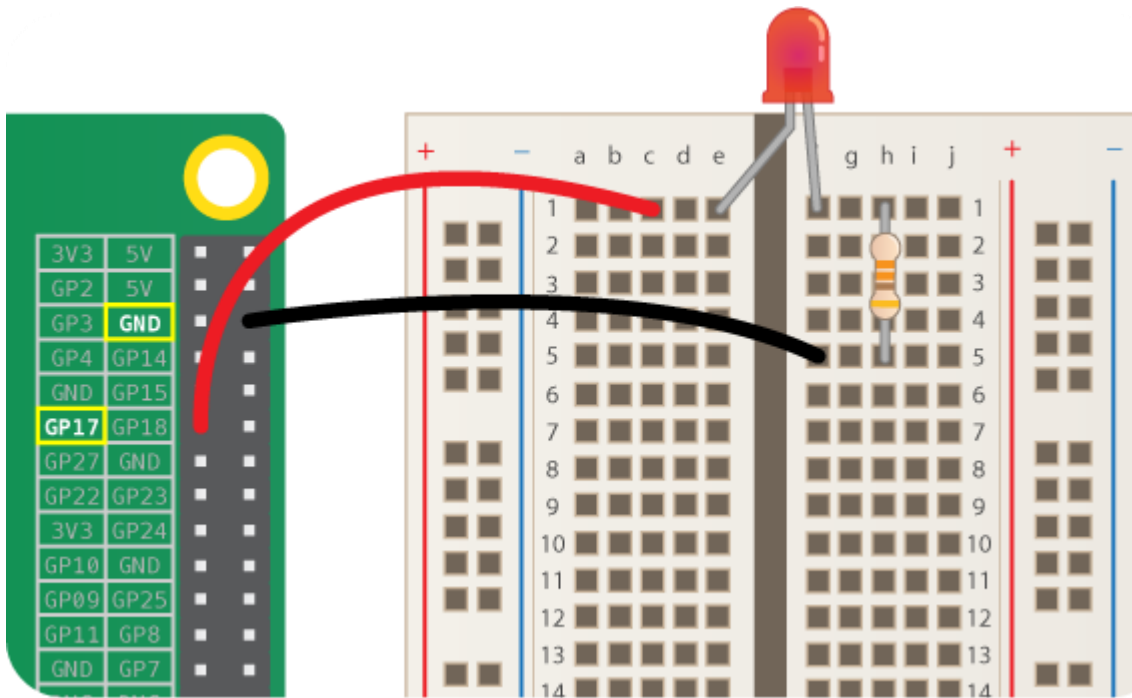
Node-RED is a programming tool for wiring together hardware devices, APIs and online services. It provides a browser-based editor that makes it easy to wire together flows using the wide range of nodes in the palette that can be deployed to its runtime in a single-click.

**GPIO Pins:**

Wedge Silk	Python (BCM)	WiringPi GPIO	Name	P1 Pin Number		Name	WiringPi GPIO	Python (BCM)	Wedge Silk
			3.3v DC Power	1	2	5v DC Power			
SDA		8	GPIO02 (SDA1, I2C)	3	4	5v DC Power			
SCL		9	GPIO03 (SCL1, I2C)	5	6	Ground			
G4	4	7	GPIO04 (GPIO_GCLK)	7	8	GPIO14 (TXD0)	15		TXO
			Ground	9	10	GPIO15 (RXD0)	16		RXI
G17	17	0	GPIO17 (GPIO_GEN0)	11	12	GPIO18 (GPIO_GEN1)	1	18	G18
G27	27	2	GPIO27 (GPIO_GEN2)	13	14	Ground			
G22	22	3	GPIO22 (GPIO_GEN3)	15	16	GPIO23 (GPIO_GEN4)	4	23	G23
			3.3v DC Power	17	18	GPIO24 (GPIO_GEN5)	5	24	G24
MOSI		12	GPIO10 (SPI_MOSI)	19	20	Ground			
MISO		13	GPIO09 (SPI_MISO)	21	22	GPIO25 (GPIO_GEN6)	6	25	G25
		(no worky 14)	GPIO11 (SPI_CLK)	23	24	GPIO08 (SPI_CE0_N)	10		CD0
			Ground	25	26	GPIO07 (SPI_CE1_N)	11		CE1
IDSD		30	ID_SD (I2C ID EEPROM)	27	28	ID_SC (I2C ID EEPROM)	31		IDSC
G05	5	21	GPIO05	29	30	Ground			
G6	6	22	GPIO06	31	32	GPIO12	26	12	G12
G13	13	23	GPIO13	33	34	Ground			
G19	19	24	GPIO19	35	36	GPIO16	27	16	G16
G26	26	25	GPIO26	37	38	GPIO20	28	20	G20
			Ground	39	40	GPIO21	29	21	G21

### Wiring up the LED:

Wire up an LED to GPIO pin 17 on your Raspberry Pi by following this diagram



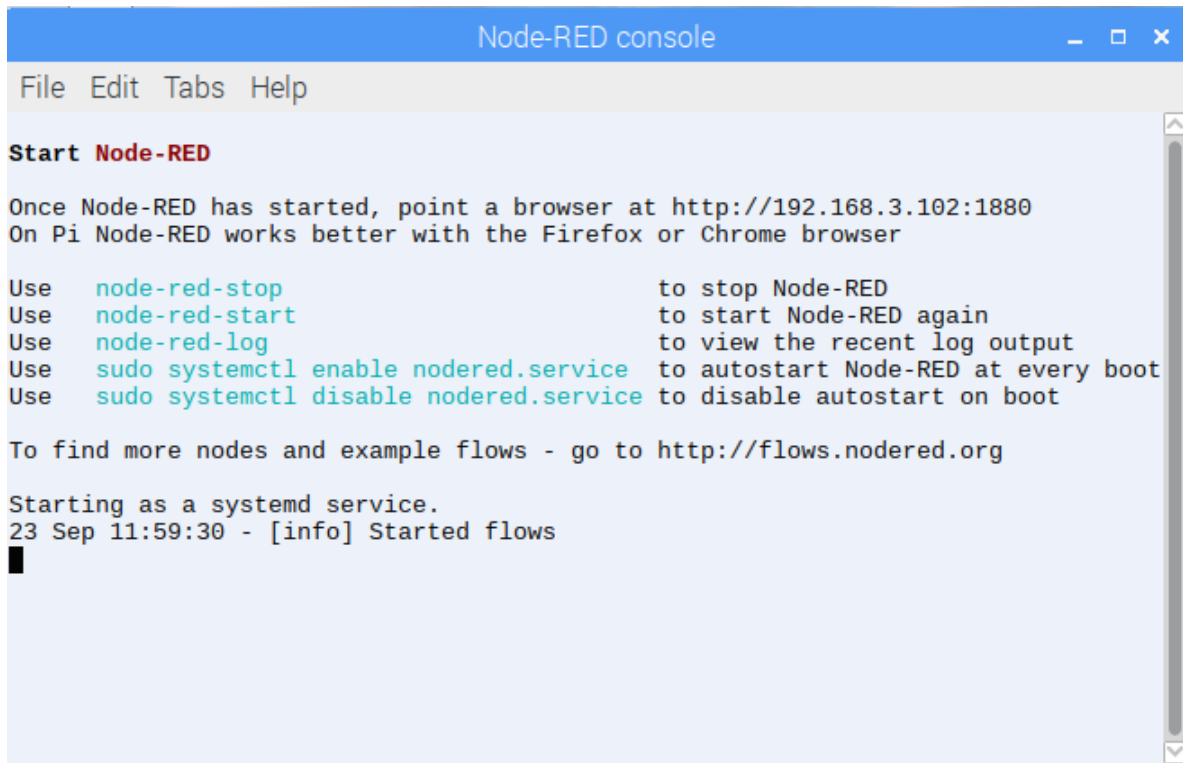
The positive leg of the LED is usually longer, and it is this leg which should be inserted into the left side of the breadboard (e1 on the diagram).

### Start Node – RED :

Start up your Raspberry Pi. Click on the Raspberry icon, then the **Programming** menu to open Node-RED.



You should see a window displaying information about Node-RED starting up.



A screenshot of a terminal window titled "Node-RED console". The window has a menu bar with "File", "Edit", "Tabs", and "Help". The main content area is light blue and contains the following text:

```
Start Node-RED

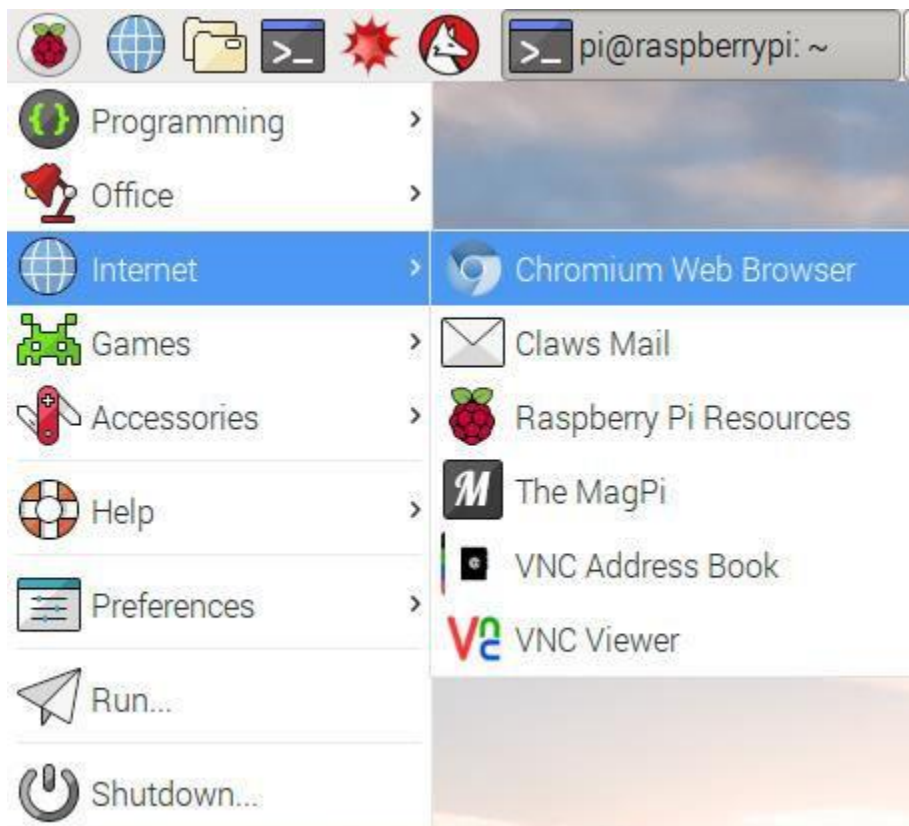
Once Node-RED has started, point a browser at http://192.168.3.102:1880
On Pi Node-RED works better with the Firefox or Chrome browser

Use  node-red-stop           to stop Node-RED
Use  node-red-start         to start Node-RED again
Use  node-red-log           to view the recent log output
Use  sudo systemctl enable nodered.service to autostart Node-RED at every boot
Use  sudo systemctl disable nodered.service to disable autostart on boot

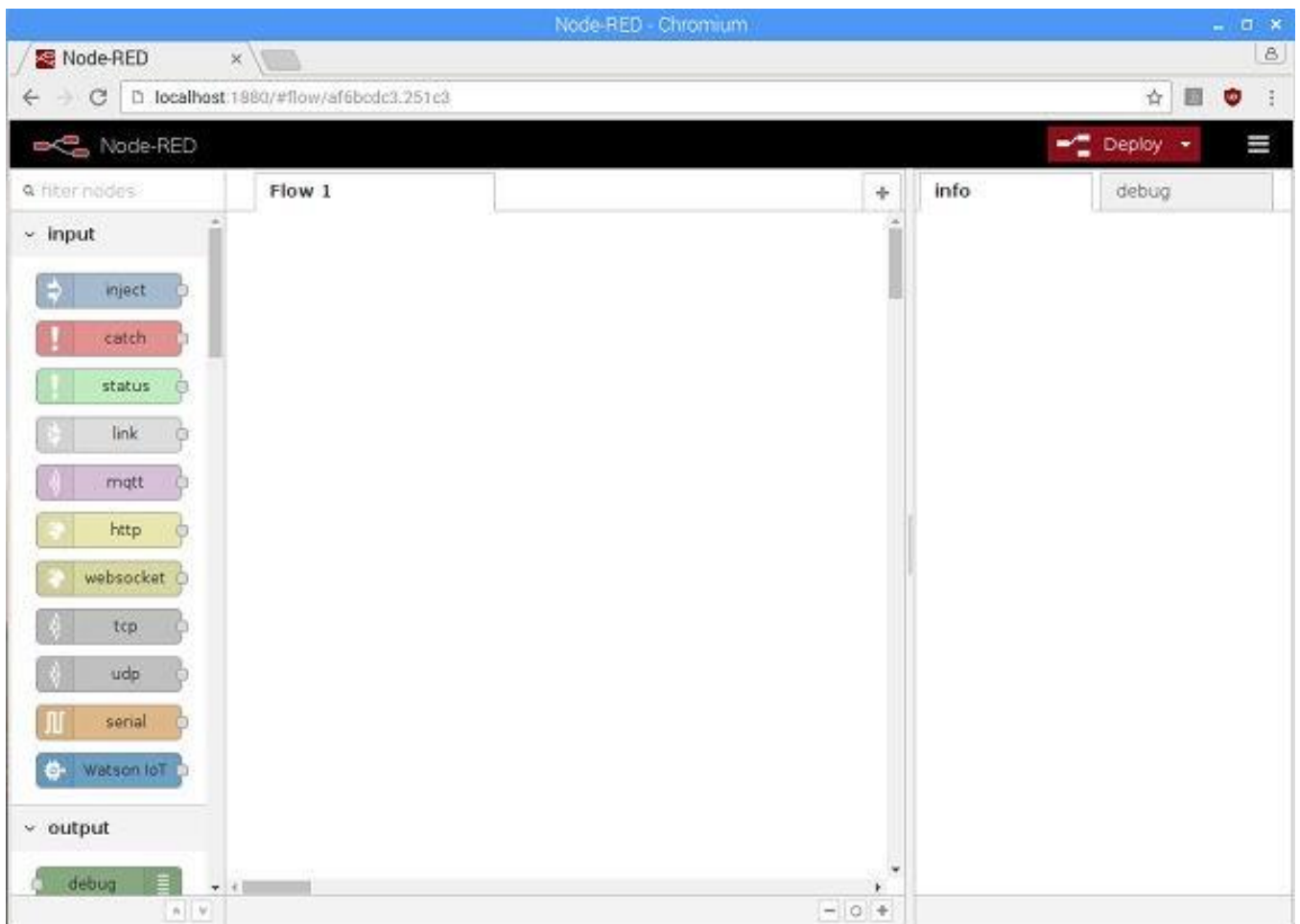
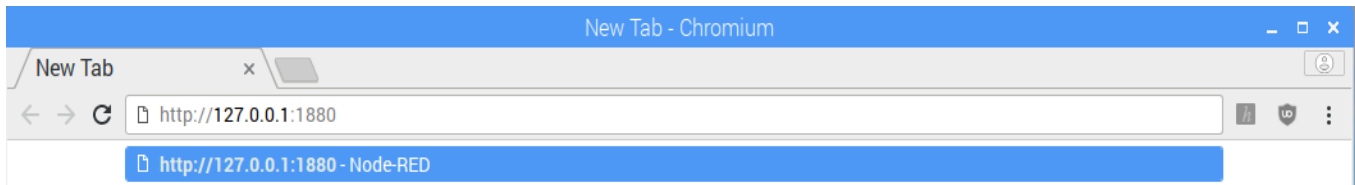
To find more nodes and example flows - go to http://flows.nodered.org

Starting as a systemd service.
23 Sep 11:59:30 - [info] Started flows
█
```

Now go to the **Internet** menu and open **Chromium Web Browser**.



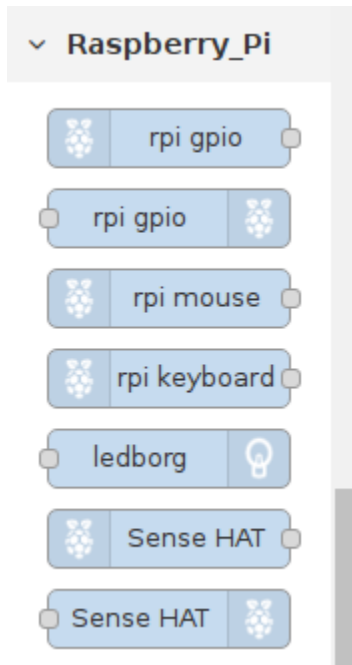
In Chromium, locate the address bar at the top and type in `localhost:1880`, then press Enter. This will display the Node-RED interface. (Your Raspberry Pi does not need to be connected to the internet to use Node-RED: `localhost` is the address the Raspberry Pi uses to refer to itself and `:1880` means that it is looking at port 1880.)



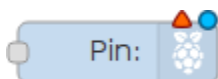
### Connecting to a GPIO pin :

Programs in Node-RED are called **flows**. You can see that your blank page is labelled as **Flow 1** in the tab at the top. You can create as many flows as you want and they can all run at the same time. For this guide, we will only need one flow.

The coloured blocks on the left side of the interface are the **nodes**. Scroll right down to the bottom of the list and you will see some nodes labelled **Raspberry Pi**.



You will see two nodes with the label **rpi gpio**: these are the ones we will use to talk to the GPIO pins on the Raspberry Pi. The first one in the list, with the raspberry icon on the left, is for inputs. Using a button push to control something would be an example of an input. The second node, with the raspberry icon on the right, is for outputs. Switching on an LED would be an example of an output. Drag an output node onto the blank page in the middle.



Double-click on the node and a box will appear to let you configure the node. Change the GPIO pin to be **GPIO17** and tick **Initialise pin state?**. Leave the setting for **Initial level of pin** on **low**. Give the node a name - we called it Green LED because the LED we used was green, but if yours is a different colour feel free to change the name. When you are finished, click **Done**.

**Edit rpi-gpio out node**

Cancel

Done

● GPIO

Pin 11 - GPIO17 ▼

Pi 3 Model B

Type

Digital output ▼

☒ Initialise pin state?

initial level of pin - low (0) ▼

Name

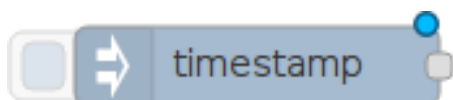
Green LED

Pins in Use: 11

Tip: For digital output - input must be 0 or 1.

### Injecting message :

Now scroll back up to the list of nodes. To turn the LED on and off, we need an input. In Node-RED we can inject messages into the flow and cause things to happen as a result. Drag an **inject** node onto the flow.




Double-click on the inject node. Use the drop down next to **Payload** to change the data type to **string** and type **1** in the Payload box - this will be the message. Type **On** in the **Name** box. Press Done.


**Edit inject node**


Cancel

Done

 Payload


▼ a\_z 1

 Topic

 Repeat

none ▼

☐ Inject once at start?

 Name

On|

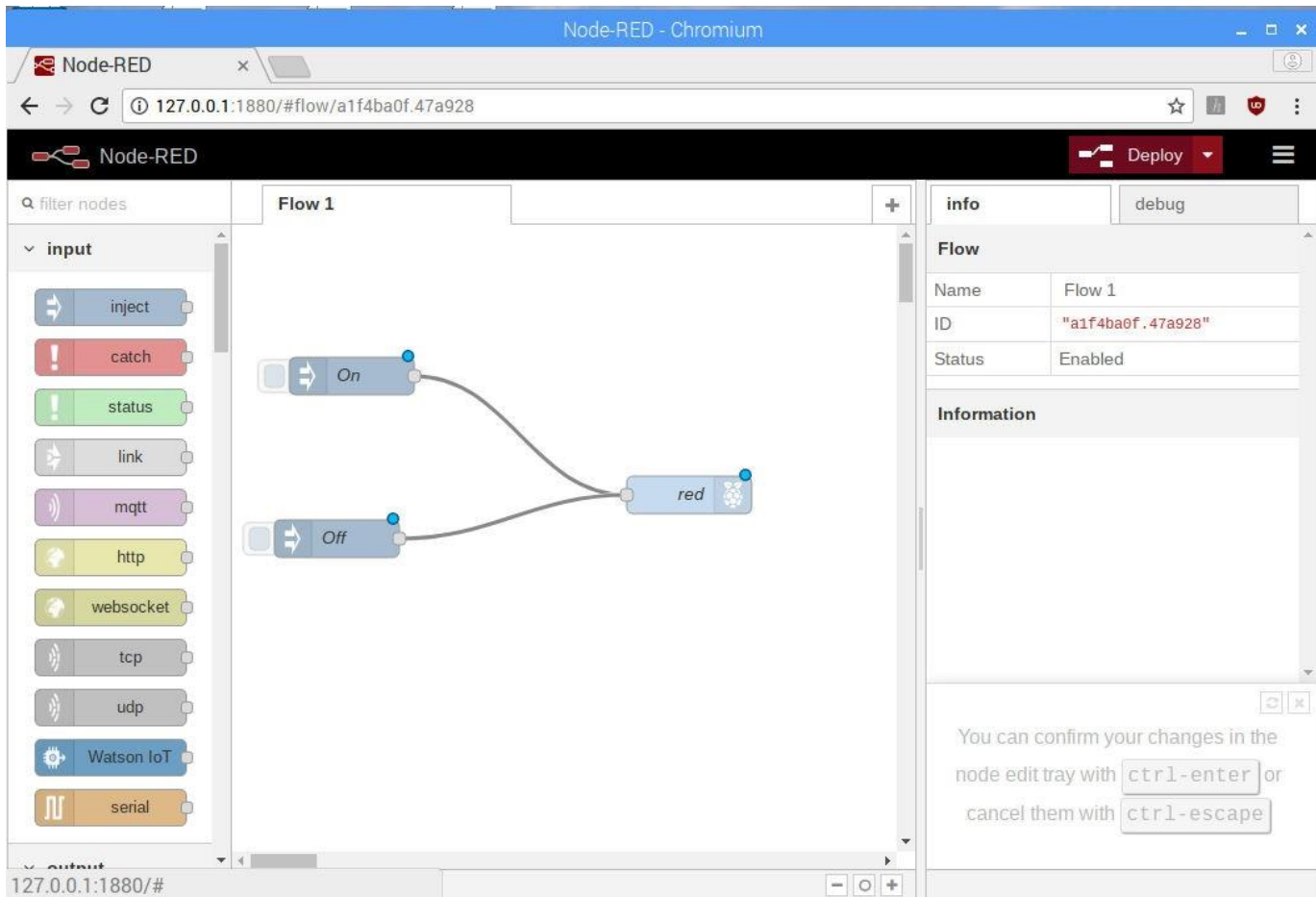
**Note:** "interval between times" and "at a specific time" will use cron.  
See info box for details.

Repeat the previous steps to create another inject node, except this time add **0** as the payload message, and call this node **Off**.





Now look for the grey dot on the right side of the inject nodes. Click and drag from the grey dot on the **On** node to the grey dot on your LED node to join them up. Repeat for the **Off** node, also joining it to the LED node.



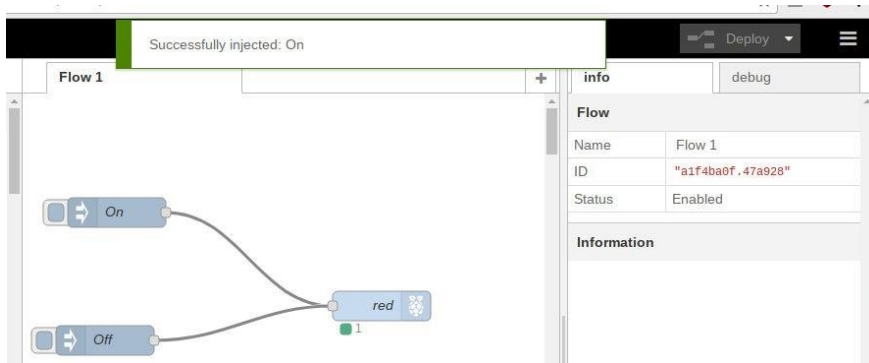
### Deploying the flow :

Our flow is finished, so we can deploy it. Click on the big red **Deploy** button on the top right of the screen. A message should pop up at the top saying "Successfully deployed". This is similar to pressing the green flag in Scratch or F5 to run your code in Python





Now click on the blue square on the left of the **On** node to inject the message **1**. The **Green LED** node receives the message and your LED should light up. You should be able to turn the LED off by clicking the blue square on the **Off** node, which injects the message **0**.



**Now for blinking of LED A simple 3 node flow that uses a Twitter tag to trigger a LED on a Raspberry PI.**

