RPi Node-Red: Advanced Multi-Button Board + RGB LED

Overview

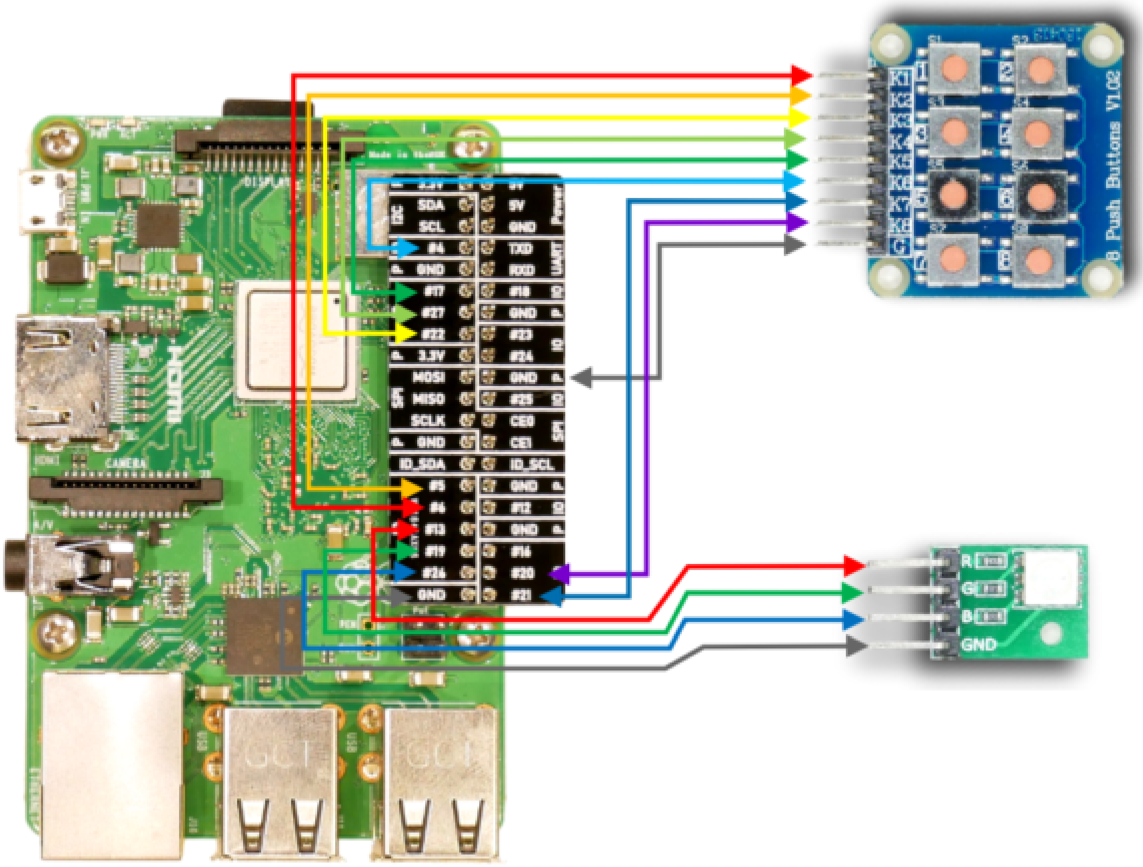
This tutorial will utilize a multi-button board input device to control an RGB LED module. Students will learn about and utilize binary counting to program the multi-button board. This tutorial adds to prior knowledge on the Button + LED tutorial.

Requirements

- Raspberry Pi 3 Model B
- Multi-Button Board
- RGB LED Module
- F-F Jumper Cables

Getting Started

Setting up the Hardware



Multi-Button Board to Raspberry Pi

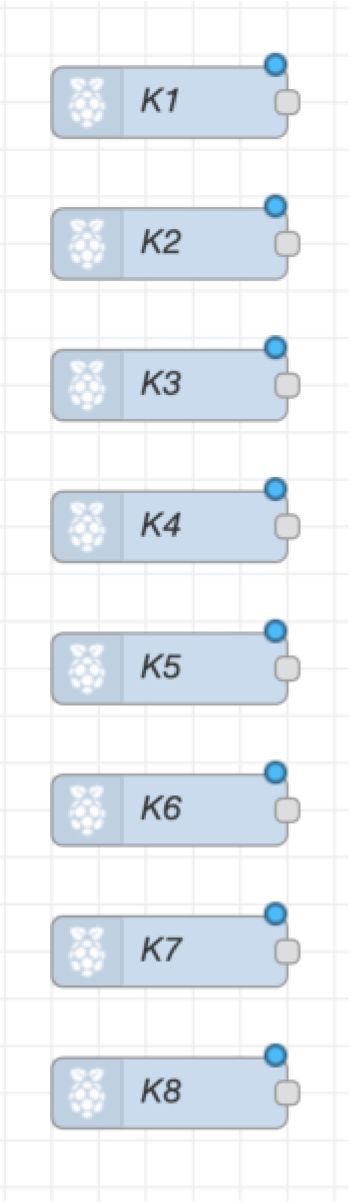
- K1 to #6
- K2 to #5
- K3 to #22
- K4 to #27
- K5 to #17
- K6 to #4
- K7 to #21
- K8 to #20
- G to GND

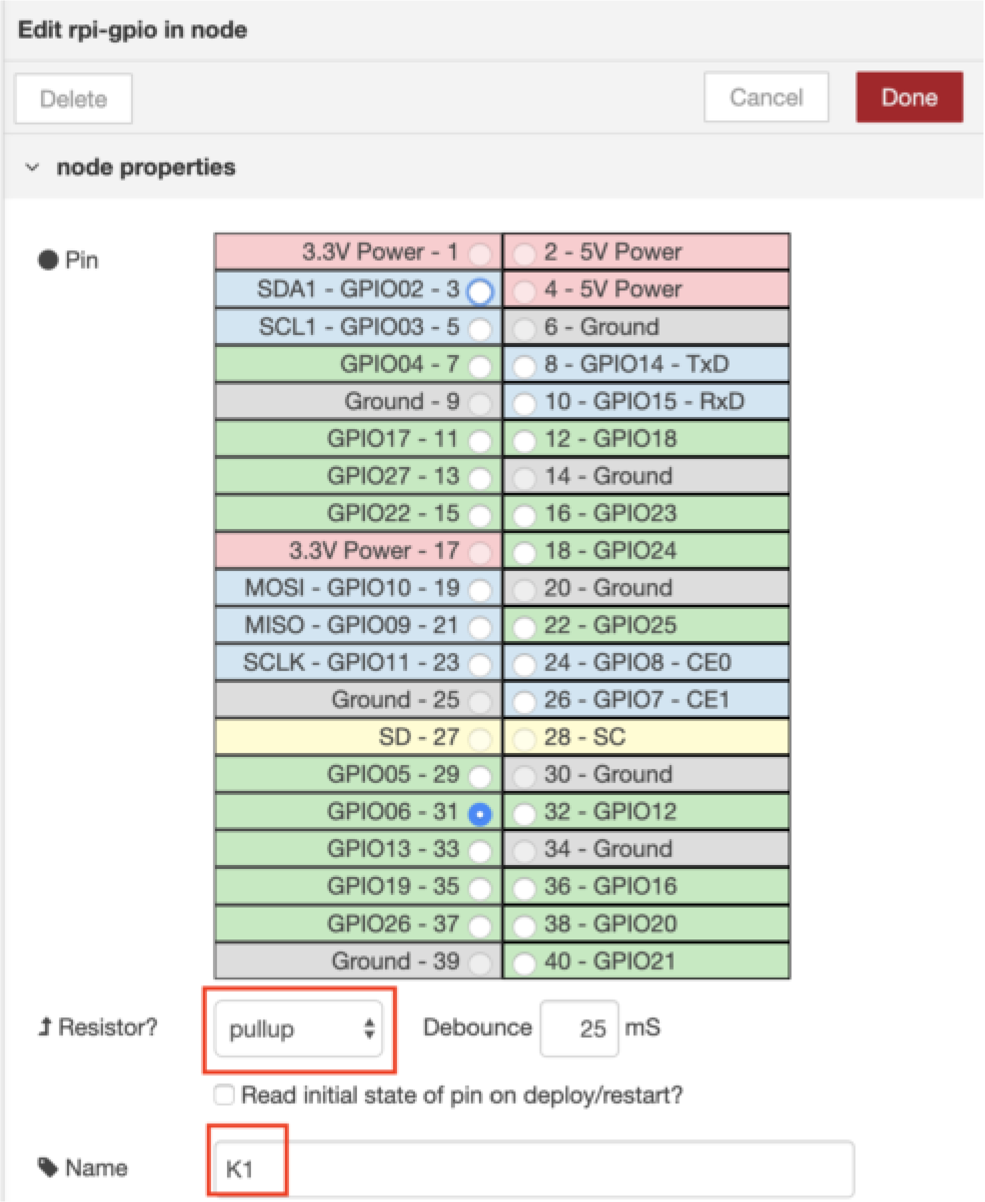
RGB LED to Raspberry Pi

- R to #13
- G to #19
- B to #26
- GND to GND

Setting up Node-Red

1. Insert RPi-GPIO-In Nodes
    1. These nodes act as inputs from the multi-button board.
    2. Place 8 nodes, representing the 8 button inputs.
    3. Assign each button (K1, K2…, K8) to their respective GPIO pin on the Raspberry Pi.
    4. Set the Resistor setting to *pullup.*
    5.





Edit rpi-gpio in node

node properties

● Pin

3.3V Power - 1 ○	○ 2 - 5V Power
SDA1 - GPIO02 - 3 ○	○ 4 - 5V Power
SCL1 - GPIO03 - 5 ○	○ 6 - Ground
GPIO04 - 7 ○	○ 8 - GPIO14 - TxD
Ground - 9 ○	○ 10 - GPIO15 - RxD
GPIO17 - 11 ○	○ 12 - GPIO18
GPIO27 - 13 ○	○ 14 - Ground
GPIO22 - 15 ○	○ 16 - GPIO23
3.3V Power - 17 ○	○ 18 - GPIO24
MOSI - GPIO10 - 19 ○	○ 20 - Ground
MISO - GPIO09 - 21 ○	○ 22 - GPIO25
SCLK - GPIO11 - 23 ○	○ 24 - GPIO8 - CE0
Ground - 25 ○	○ 26 - GPIO7 - CE1
SD - 27 ○	○ 28 - SC
GPIO05 - 29 ○	○ 30 - Ground
GPIO06 - 31 ●	○ 32 - GPIO12
GPIO13 - 33 ○	○ 34 - Ground
GPIO19 - 35 ○	○ 36 - GPIO16
GPIO26 - 37 ○	○ 38 - GPIO20
Ground - 39 ○	○ 40 - GPIO21

Resistor? pullup

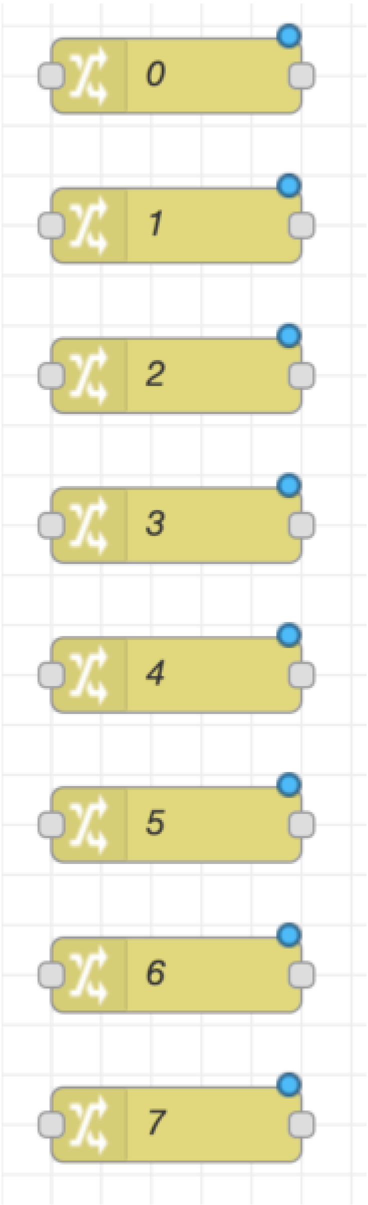
Debounce 25 mS

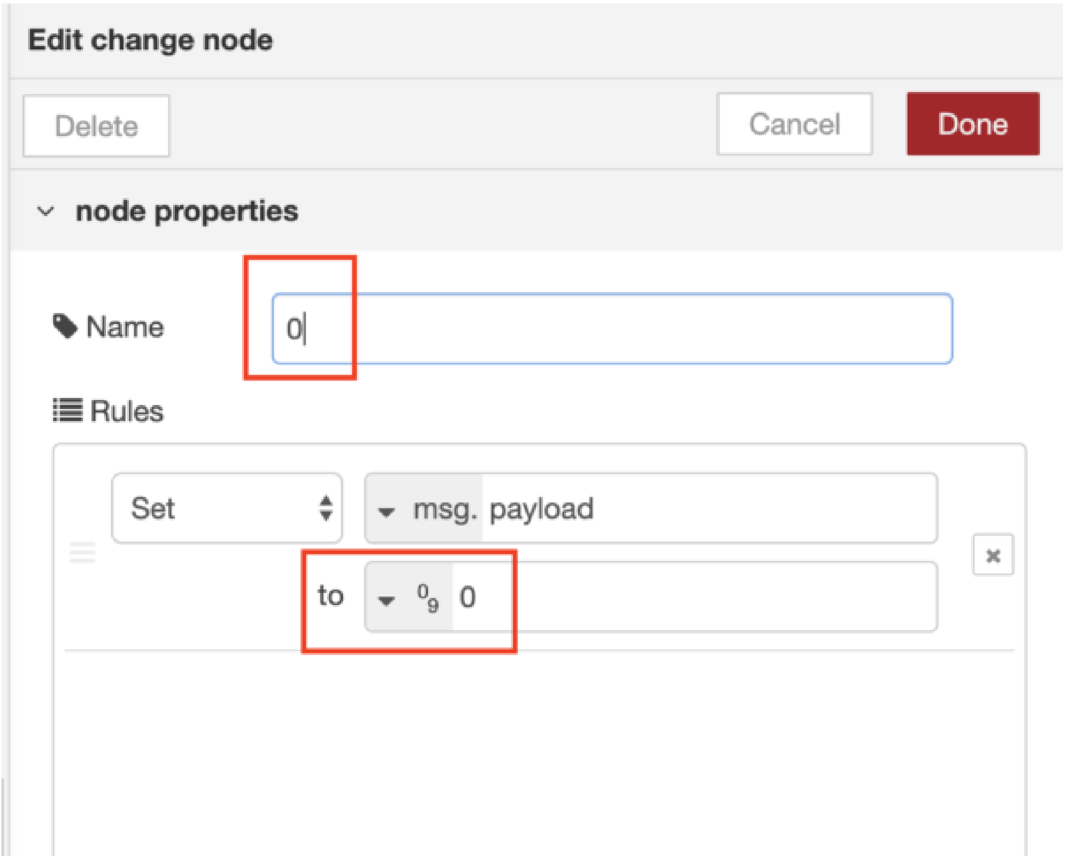
□ Read initial state of pin on deploy/restart?

Name K1

2. Insert Change Nodes
    1. These nodes translate the input nodes from the multi-button board into number values from 0 to 7 (numbering starts at 0 due to how computers start counting at 0 instead of 1).
    2. Place 8 nodes, labeling them from 0 to 7.
    3. For each node, change the data type to *number* and set the values to 0-7 starting with the first node.

4.





Edit change node

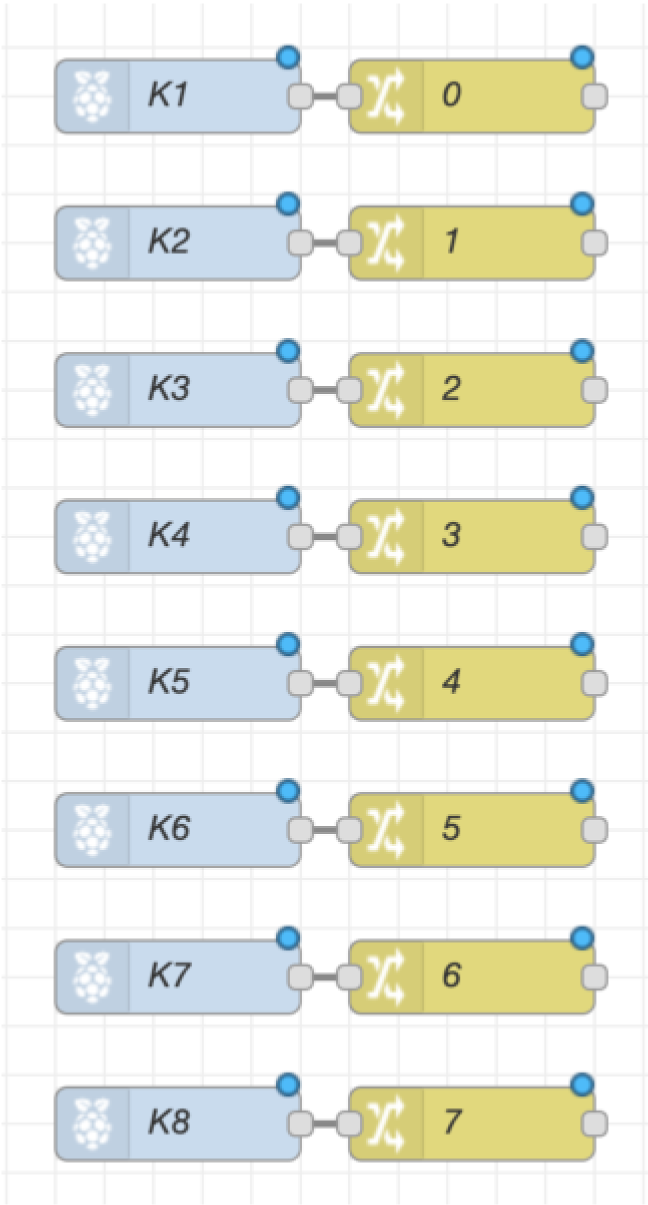
node properties

Name: 0

Rules

Set msg. payload to 0

3. Connect an RPi-GPIO-In node to a Change node
    1. Connecting the two node types translates each binary 0 or 1 button input into a different decimal value.

2. 

4. Insert Function Nodes
    1. These nodes act as “decoders” that translates the numbered value into a bit-string.
    2. The RGB values represent three individual switches. By having a bit-string representation, each bit can determine the color’s bit state.
    3. Function for [4] node

        1.
           ```
           // Value retrieved from initial input
           var value = msg.payload;

           // subtract_bit determined by bit location
           var subtract_bit = 4;
           ```

```
// If value is greater than the
subtract_bit, indicates that the bit state
for [4] is 1
if (value >= subtract_bit) {
     var new_value = {payload: value -
subtract_bit};
    var bit_state = {payload: 1};
    return [new_value, bit_state];
}

// Otherwise, bit state for [4] is 0
else {
    value = {payload: value};
    var bit_state = {payload: 0};
    return [value, bit_state];
}
```

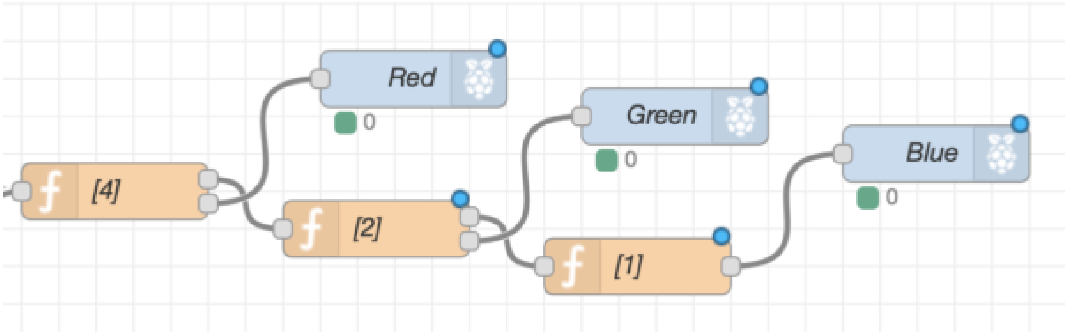
4. Function for [2] node

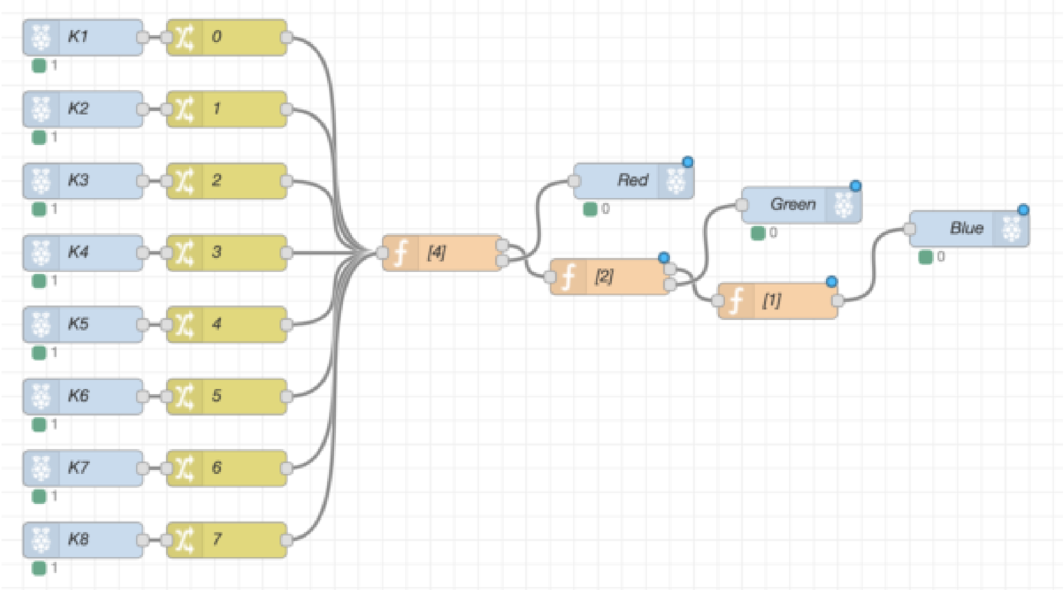
    1.
       ```
       // Value retrieved from initial input
       var value = msg.payload;

       // subtract_bit determined by bit location
       var subtract_bit = 2;

       // If value is greater than the
       subtract_bit, indicates that the bit state
       for [2] is 1
       if (value >= subtract_bit) {
            var new_value = {payload: value -
       subtract_bit};
           var bit_state = {payload: 1};
           return [new_value, bit_state];
       }

       // Otherwise, bit state for [2] is 0
       else {
       ```

```
                value = {payload: value};
                var bit_state = {payload: 0};
                return [value, bit_state];
            }
```

    5. Function for [1] node

        1.
           ```
           var bit_state = {payload: msg.payload};
           return bit_state;
           ```

    6. 

5. Insert RPi-GPIO-Out Nodes
    1. These nodes act as output nodes for the RGB LED module.
    2. Place three nodes, each representing one of the RGB LEDs and assign to their respective GPIO pin.
    3. 

Conclusion

If you have successfully followed this tutorial module, you should be able to change the RGB LED module’s color from the multi-button board.

Node-Red Solution

[{"id":"3fc0ad9f.487e12","type":"change","z":"b8b26cb8.1baef","name":"2","rules":[{"t":"set","p":"payload","pt":"msg","to":"2","tot":"num"}],"action":"","property":"","from":"","to":"","reg":false,"x":410,"y":280,"wires":[["5f4cfd28.f4a124"]]},{"id":"cbd11229.36a9","type":"rpi-gpio in","z":"b8b26cb8.1baef","name":"K2","pin":"29","intype":"up","debounce":"25","read":false,"x":290,"y":220,"wires":[["c4e5685f.e3fa88"]]},{"id":"96dd66c6.026928","type":"rpi-gpio in","z":"b8b26cb8.1baef","name":"K4","pin":"13","intype":"up","debounce":"25","read":false,"x":290,"y":340,"wires":[["75069901.3cc9b8"]]},{"id":"d710bb14.62b1f8","type":"rpi-gpio in","z":"b8b26cb8.1baef","name":"K3","pin":"15","intype":"up","debounce":"25","read":false,"x":290,"y":280,"wires":[["3fc0ad9f.487e12"]]},{"id":"2b18a8d6.7f92f8","type":"rpi-gpio in","z":"b8b26cb8.1baef","name":"K6","pin":"7","intype":"up","debounce":"25","read":false,"x":290,"y":460,"wires":[["ac5a8574.e332f8"]]},{"id":"3c93bf81.fb10d","type":"rpi-gpio in","z":"b8b26cb8.1baef","name":"K8","pin":"38","intype":"up","debounce":"25","read":false,"x":290,"y":580,"wires":[["d0ff8cfe.8f2c5"]]},{"id":"aac40b7a.f36bb8","type":"rpi-gpio in","z":"b8b26cb8.1baef","name":"K7","pin":"40","intype":"up","debounce":"25","read":false,"x":290,"y":520,"wires":[["3954b9a.a55c346"]]},{"id":"551dc587.3f572c","type":"rpi-gpio in","z":"b8b26cb8.1baef","name":"K5","pin":"11","intype":"up","debounce":"25","read":false,"x":290,"y":400,"wires":[["631bd1a6.8dd45"]]},{"id":"31bb63ee.bf082c","type":"change","z":"b8b26cb8.1baef","name":"0","rules":[{"t":"set","p":"payload","pt":"msg","to":"0","tot":"num"}],"action":"","property":"","from":"","to":"","reg":false,"x":410,"y":160,"wires":[["5f4cfd28.f4a124"]]},{"id":"c4e5685f.e3fa88","type":"change","z":"b8b26cb8.1baef","name":"1","rules":[{"t":"set","p":"payload","pt":"msg","to":"1","tot":"num"}],"action":"","property":"","from":"","to":"","reg":false,"x":410,"y":220,"wires":[["5f4cfd28.f4a124"]]},{"id":"75069901.3cc9b8","type":"change","z":"b8b26cb8.1baef","name":"3","rules":[{"t":"set","p":"payload","pt":"msg","to":"3","tot":"num"}],"action":"","property":"","from":"","to":"","reg":false,"x":410,"y":340,"wires":[["5f4cfd28.f4a124"]]},{"

id":"631bd1a6.8dd45","type":"change","z":"b8b26cb8.1baef","nam
e":"4","rules":[{"t":"set","p":"payload","pt":"msg","to":"4","
tot":"num"}],"action":"","property":"","from":"","to":"","reg"
:false,"x":410,"y":400,"wires":[["5f4cfd28.f4a124"]]},{"id":"a
c5a8574.e332f8","type":"change","z":"b8b26cb8.1baef","name":"5
","rules":[{"t":"set","p":"payload","pt":"msg","to":"5","tot":
"num"}],"action":"","property":"","from":"","to":"","reg":fals
e,"x":410,"y":460,"wires":[["5f4cfd28.f4a124"]]},{"id":"3954b9
a.a55c346","type":"change","z":"b8b26cb8.1baef","name":"6","ru
les":[{"t":"set","p":"payload","pt":"msg","to":"6","tot":"num"
}],"action":"","property":"","from":"","to":"","reg":false,"x"
:410,"y":520,"wires":[["5f4cfd28.f4a124"]]},{"id":"d0ff8cfe.8f
2c5","type":"change","z":"b8b26cb8.1baef","name":"7","rules":[
{"t":"set","p":"payload","pt":"msg","to":"7","tot":"num"}],"ac
tion":"","property":"","from":"","to":"","reg":false,"x":410,"
y":580,"wires":[["5f4cfd28.f4a124"]]},{"id":"23c6a01e.0e013","
type":"rpi-gpio
in","z":"b8b26cb8.1baef","name":"K1","pin":"31","intype":"up",
"debounce":"25","read":false,"x":290,"y":160,"wires":[["31bb63
ee.bf082c"]]},{"id":"5f4cfd28.f4a124","type":"function","z":"b
8b26cb8.1baef","name":"[4]","func":"// Value retrieved from
initial input\nvar value = msg.payload;\n\n// subtract_bit
determined by bit location\nvar subtract_bit = 4;\n\n// If
value is greater than the subtract_bit, indicates that the bit
state for [4] is 1\nif (value >= subtract_bit) {\n   var
new_value = {payload: value - subtract_bit};\n   var
bit_state = {payload: 1};\n   return [new_value,
bit_state];\n}\n\n// Otherwise, bit state for [4] is 0\nelse
{\n   value = {payload: value};\n   var bit_state =
{payload: 0};\n   return [value,
bit_state];\n}","outputs":2,"noerr":0,"x":590,"y":340,"wires":
[["fb4d798f.1edc38"],["a17a86af.bfb648"]]},{"id":"fb4d798f.1ed
c38","type":"function","z":"b8b26cb8.1baef","name":"[2]","func
":"// Value retrieved from initial input\nvar value =
msg.payload;\n\n// subtract_bit determined by bit
location\nvar subtract_bit = 2;\n\n// If value is greater than
the subtract_bit, indicates that the bit state for [2] is
1\nif (value >= subtract_bit) {\n   var new_value = {payload:
value - subtract_bit};\n   var bit_state = {payload: 1};\n
return [new_value, bit_state];\n}\n\n// Otherwise, bit state
for [2] is 0\nelse {\n   value = {payload: value};\n   var

bit_state = {payload: 0};\n return [value,
bit_state];\n}","outputs":2,"noerr":0,"x":730,"y":360,"wires":
[["bb73fc3f.28e14"],["74e9a7d8.f4a0e8"]]},{"id":"bb73fc3f.28e1
4","type":"function","z":"b8b26cb8.1baef","name":"[1]","func":
"var bit_state = {payload: msg.payload};\nreturn
bit_state;","outputs":1,"noerr":0,"x":870,"y":380,"wires":[["9
a25ca9f.d2dca8"]]},{"id":"a17a86af.bfb648","type":"rpi-gpio
out","z":"b8b26cb8.1baef","name":"Red","pin":"33","set":"","le
vel":"0","freq":"","out":"out","x":750,"y":280,"wires":[]},{"i
d":"74e9a7d8.f4a0e8","type":"rpi-gpio
out","z":"b8b26cb8.1baef","name":"Green","pin":"35","set":"","
level":"0","freq":"","out":"out","x":890,"y":300,"wires":[]},{
"id":"9a25ca9f.d2dca8","type":"rpi-gpio
out","z":"b8b26cb8.1baef","name":"Blue","pin":"37","set":"","l
evel":"0","freq":"","out":"out","x":1030,"y":320,"wires":[]}]