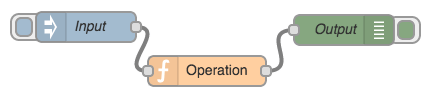
# **Node-red cơ bản**

# *Node-red nó là gì ? Node-RED được dựa trên Node.js, nó có thể được xem như một web server mà bạn có thể cấu hình tùy chỉnh các chức năng gọi là “flow” từ bất kỳ trình duyệt nào trên máy tính. Mỗi ứng dụng Node-RED bao gồm các node có thể liên kết được với nhau với các dạng là input, output và operation.*

Một ví dụ đơn giản để chúng ta có thể hình dung được các node khác nhau sẽ tương tác như thế nào



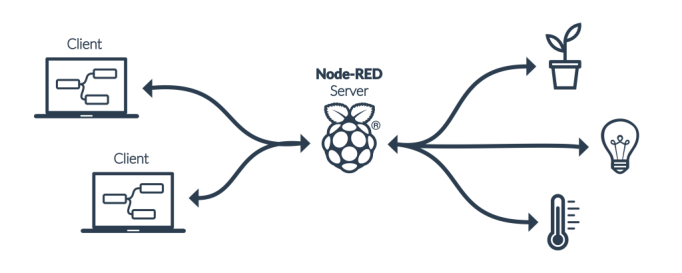
Ví du: để điều khiển động cơ thì:

Input: là nút bấm bật tắt hoặc là tốc độ mong muốn của động cơ

Operation: hàm thực thi.

Output: động cơ.

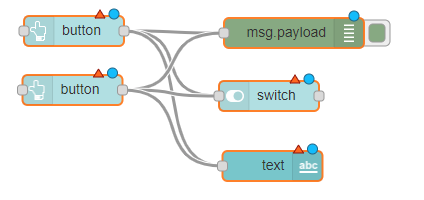
Với Node-RED ta có thể hình dung cách tương tác và giao tiếp với các thiết bị một cách tổng quan như hình dưới đây.  Raspberry sẽ đóng vai trò là Server, còn lại sẽ là client:



# **Ví dụ với Node-RED dashboard**

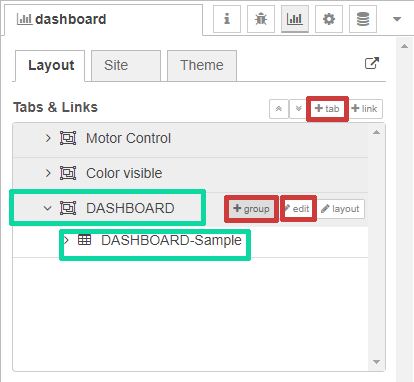
Chúng ta sẽ bắt đầu với ví dụ node-red cơ bản đầu tiên. Sử dụng 2 nút nhấn và kiểm tra trạng thái hiển thị 2 nút nhấn trên giao diện xem thế nào. Để lập trình với node-red trước tiên ta cần khởi động PI3. Truy cập wifi trên máy tính cá nhân do PI3 phát ra. Sau khi kết nối thành công thì trên máy tính cá nhân truy cập vào địa chỉ <http://10.3.141.1:1880/> trên giao diện web .

Bước 1: Thực hiện kéo thả 3 khối gồm 2 button ON/ OFF, 1 switch để thay đổi trạng thái, 1 text để hiện giá trị hiện tại nút nhấn và 1 msq.payload để debug



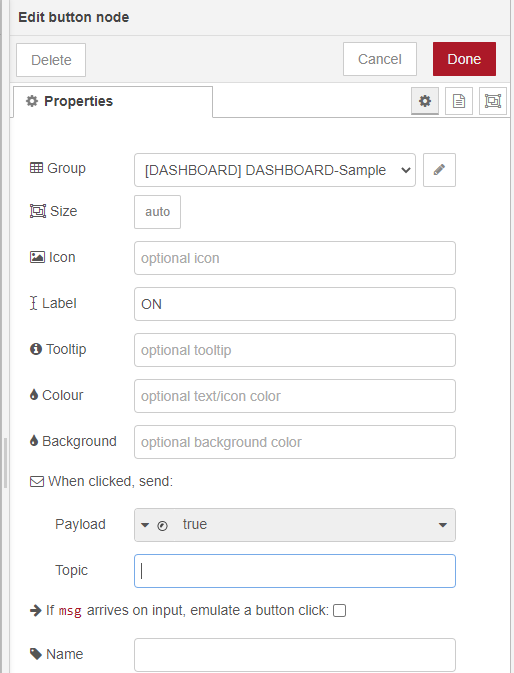
Bước 2: Tạo layout giao diện dashboard

Chọn New tab -> new group. Sau đó chọn edit để đổi lại tên như hình dưới đây:



Bước 3:Cấu hình input, output:

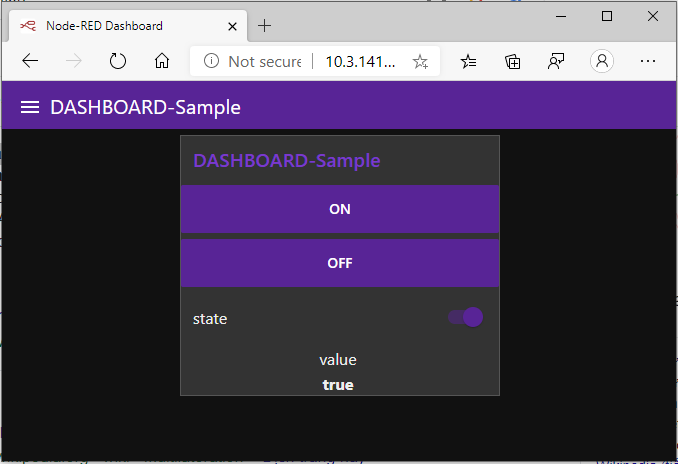
* Button1: Kích đúp vào button và sửa lại các mục như sau:



* Group: Chính là table đã thêm ở bước trước. Chọn DASHBOARD
* Label: Đặt tên hiện thị cho button. Chúng ta đặt trước 1 button là ON.
* Payload: kích vào hình tam giác bên phía tay trai chọn kiểu dữ liệu là Boolean sau đó đặt giá trị là True. Tương đương với button ON.
* Button2: Tương tự cài đặt cho button còn lại với label là “OFF” và giá trị là FALSE.
* Switch: Cũng đặt Group là DASHBOARD. Label là “State”
* Text: Cũng đặt Group là DASHBOARD. Label là “Value”.

Bước 4: Chọn Deploy, quan sát debug và giao diện,

chọn đường dẫn <http://10.3.141.1:1880/ui/>



1. **Ví dụ sử dụng GPIO**

Vi điều khiển trong các board mạch tương tác với thế giới bên ngoài thông qua các chân GPIO (General Purpose Input Output Pins).

Các chân GPIO được nối với các thiết bị cảm biến để theo dõi, đo đạc môi trường bên ngoài hoặc thiết bị điều khiển đóng ngắt như bật tắt đèn, loa, động cơ…Như vậy, khi nói đến các nền tảng phần cứng, bên cạnh sức mạnh của vi điều khiển (tốc độ và bộ nhớ) thì số chân GPIO là hết sức quan trọng vì nó đem lại khả năng mở rộng của thiết bị.

* 1. **Bật tắt led (OUTPUT)**

Các chân tín hiệu digital sẽ có 2 chế độ hoạt động, là Output dùng để điều khiển các module gắn vào pin này hoặc Input để thu thập tín hiệu. Trong bài này chúng ta thử sử dụng digital pin ở chế độ output để bật tắt đèn LED trên mạch.

Bước 1: Thực hiện kéo thả 2 khối gồm 2 button ON/ OFF, và 9 rpi gpio out.

A screenshot of a cell phone

Description automatically generated

*A screenshot of a cell phone

Description automatically generated*

*Hình 3.1.1: Sơ đồ mạch nguyên lý khối LED*  *Hình 3.1.2. Sơ đồ mạch nguyên lý*

A circuit board

Description automatically generated

*Hình 3.1.3:* PCB 3D

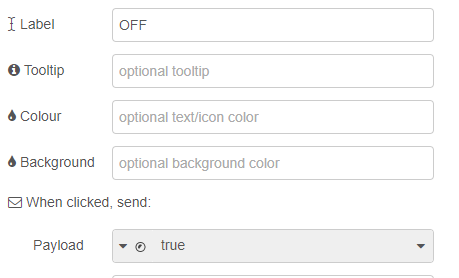
*Để bật tắt LED D12 trên mạch(hình 3.1.3) sẽ tương đương với cấp nguồn cho SIGNAL-LED9 (hình 3.1.1) và đối chiếu sang hình 3.1.2 sẽ là cấp nguồn cho GPIO 15 trên PI3. Ta sẽ nhập PIN 15 và tương tự các PIN khác như hình dưới đây:*

A close up of a device

Description automatically generated

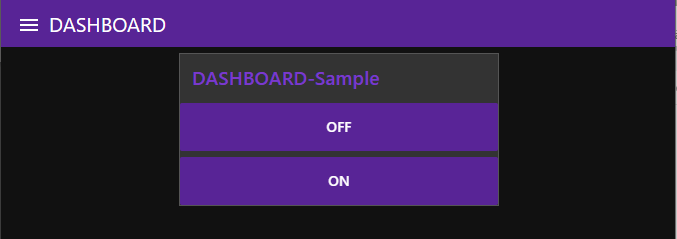
Bước 2: Tạo layout giao diện dashboard: tương tự nhưu các ví dụ trước.

Bước 3: Nháy kép vào Button OFF chọn Payload là “true” và Button ON chọn payload là “false”. Do chân IO này đã được kéo trở lên mức cao.



Bước 4: Chọn Deploy, quan sát debug và giao diện,

chọn đường dẫn <http://10.3.141.1:1880/ui/>



* 1. **Ví dụ sử dụng GPIO Đọc dữ liệu từ button**

Ngược lại với chế độ digital output là digital input. Bạn sẽ dùng digital input khi pin digital nối với các module trả về trạng thái là 0 hoặc 1. Trong bài này chúng ta tìm hiểu các sử dụng nút nhấn (button) và đọc tín hiệu nút nhấn để phát hiện mỗi khi nút được nhấn và bật đèn led. Ngược lại, chúng ta sẽ tắt đèn led khi nút nhấn được thả ra.

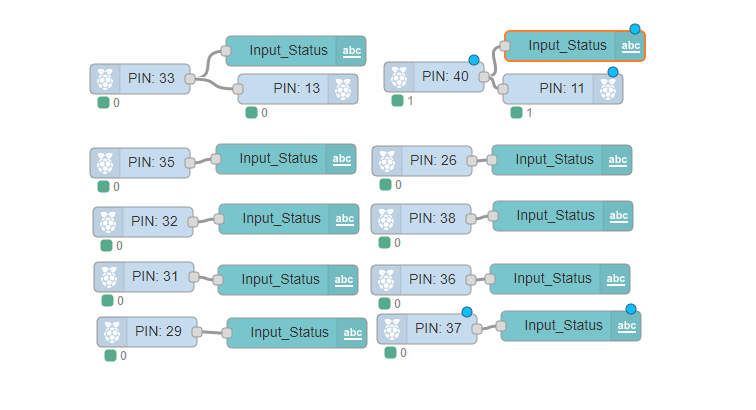
Bước 1: Thực hiện kéo thả 2 khối gồm 10 rpi gpio in, 10 text, và 2 rpi gpio out.

A picture containing screenshot

Description automatically generated

*Hình 3.2.1: Sơ đồ mạch nguyên lý khối LED*

*Để đọc dữ liệu từ chân J10 trên mạch (hình 3.1.3) sẽ tương đương với đọc dữ liệu từ DIN 03. Đối chiếu với sơ đồ mạch trên hình 3.1.2 sẽ là đọc dữ liệu từ PIN 33 trên PI3. Ta sẽ nhập PIN 33 và tương tự các PIN khác như hình dưới đây:*



Bước 2: Tạo layout giao diện dashboard: tương tự nhưu các ví dụ trước.

Bước 3: Chọn Deploy, quan sát debug và giao diện,

chọn đường dẫn <http://10.3.141.1:1880/ui/>

1. **Điều khiển động cơ:**

Pulse Width Modulation (PWM) là một khái niệm được sử dụng rộng rãi trong vật lý, điện tử và viễn thông. Nó được định nghĩa là một dạng điều chế tín hiệu (signal modulation) để lấy dạng sóng analog từ các đầu vào kỹ thuật số. PWM rất hữu ích trong việc thay đổi cường độ của tín hiệu như độ sáng của diode LED, hoặc cung cấp năng lượng cho động cơ servo.

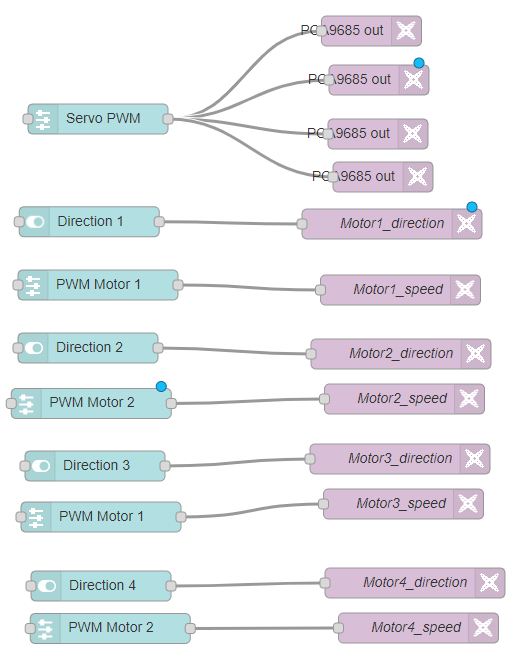
Đặt trường hợp ta muốn đèn LED sáng với nhiều mức độ khác nhau, giải pháp đầu tiên có thể kể đến là thay vì sử dụng tín hiệu số (digital) để điều khiển LED, ta có thể chuyển sang tín hiệu tương tự (analog). Khi đó, hiệu điện thế sẽ được tinh chỉnh theo thời gian, và đèn LED của chúng ta sẽ sáng theo yêu cầu.

Tuy nhiên, với PI việc tăng / hạ áp tại một pin bất kì không phải là một việc đơn giản. Đơn cử như muốn đưa điện áp từ một hiệu điện thế bất kì về 5V, ta phải sử dụng IC giảm áp, ví dụ như LM2596. Vì vậy, kĩ thuật điều chế độ rộng xung ra đời để hỗ trợ chúng ta khắc phục vấn đề này. Với kĩ thuật này, nhờ vào tần số dao động của tín hiệu điện trong mạch mà ta có được kết quả của việc sử dụng tín hiệu tương tự bằng tín hiệu số.

Tín hiệu số sẽ tạo ra các xung vuông, một dạng tín hiệu bật / tắt, tín hiệu bật / tắt này tương ứng với mức 5V và 0V. Bằng việc thay đổi khoảng thời gian bật và tắt này trong một chu kì rất ngắn, ta có thể tạo ra được "ảo giác" đèn LED sáng mờ. Khoảng thời gian bật / tắt trong một chu kì được gọi là biên độ xung - pulse width.

Để hạn chế phức tạp trong lập trình tạo xung PWM, trên mạch đã thiết kế thêm 1 IC PCA9685 có nhiệm vụ tạo ra 16 kênh PWM, việc giao tiếp với PI sử dụng phương thức I2C.

Việc lập trình giao tiếp với IC PCA9685 tham khảo Tab Test DC motor trên địa chỉ <http://10.3.141.1:1880/>



Chú ý: khi nháy kép vào node PCA9685 out ta thấy tab  PCA9685 Device. Tab này có nhiệm vụ cài đặt địa chỉ I2C trên PI. PCA9685 trên mạch mặc định sử dụng địa chỉ 0x64 theo hệ thập phân.

1. **Đọc dữ liệu từ cảm biến màu**

Dữ liệu từ cảm biến màu cũng được xử lý trước qua 1 con IC, nhiệm vụ của IC này sẽ đọc các giá trị màu đo được và trả lại giá trị đó cho PI qua giao thức I2C. Địa chỉ I2C của cảm biến màu sử dụng là 0x41 theo hệ thập phân.Việc lập trình giao tiếp với IC cảm biến màu tham khảo Tab Color Sensor trên địa chỉ <http://10.3.141.1:1880/>

1. **Kết nối tay cầm và điều khiển**

Các đội thi đước cung cấp 1 node đọc dữ liệu từ tay

điều khiển Bluetooth như hình sau:

A close up of a logo

Description automatically generated

Ở node này ta có 11 đầu ra chính là tín hiệu trả về tương ứng từ tay điều khiển. Rê chuột vào từng ô vuông nhỏ sẽ hiển thị thông tin nó là đầu ra của nút bấm nào. 19 đầu ra và giá trị trả về được mô tả như sau:

*H3: trả giá trị về là -100 và 100*

*V3: trả giá trị về là -100 và 100*

*H2: trả giá trị thay đổi từ -100 đến 100*

*V2: trả giá trị thay đổi từ -100 đến 100*

*H1: trả giá trị thay đổi từ -100 đến 100*

*V1: trả giá trị thay đổi từ -100 đến 100*

*LJ: trả giá trị 0 hoặc 1 tương ứng với không ấn và có ấn*

*RJ: trả giá trị 0 hoặc 1 tương ứng với không ấn và có ấn*

*HO: trả giá trị 0 hoặc 1 tương ứng với không ấn và có ấn*

*RB: trả giá trị 0 hoặc 1 tương ứng với không ấn và có ấn*

*LB: trả giá trị 0 hoặc 1 tương ứng với không ấn và có ấn RT: trả giá trị 0 hoặc 1 tương ứng với không ấn và có ấn*

*LT: trả giá trị 0 hoặc 1 tương ứng với không ấn và có ấn*

*SL: trả giá trị 0 hoặc 1 tương ứng với không ấn và có ấn*

*ST: trả giá trị 0 hoặc 1 tương ứng với không ấn và có ấn*

*Các đầu ra trên tương ứng với vị trí các nút được mô tả trên*

*hình vẽ sau:*

A picture containing black, red

Description automatically generated

Các bước để lập trình với tay cầm như sau:

* Đưa tay cầm về trạng thái phát bluetooth:
  + ấn nút home + nút X trong khoảng 5 giây đến khi thấy đèn led tắt.
  + nhả nút home và ấn lại nút home 1 lần nữa. Nếu thấy Led nhấp nháy nhanh hơn thì thành công.
* Kết nối Bluetooth:
  + Nhập đỉa chỉ Mac cung cấp sẵn được dán trên tay cầm vào node Gamepad Controller bằng cách nháy kép chuột vào nó.

A screenshot of a cell phone

Description automatically generated

Sử dụng các node tùy theo mục đích. Có 1 ví dụ cung cấp

sẵn để điều khiển động cơ như sau:

A close up of a map

Description automatically generated