

# LAB THỰC HÀNH SỐ 4

## Gửi Dữ Liệu Đến Node-Red Thông Qua MQTT

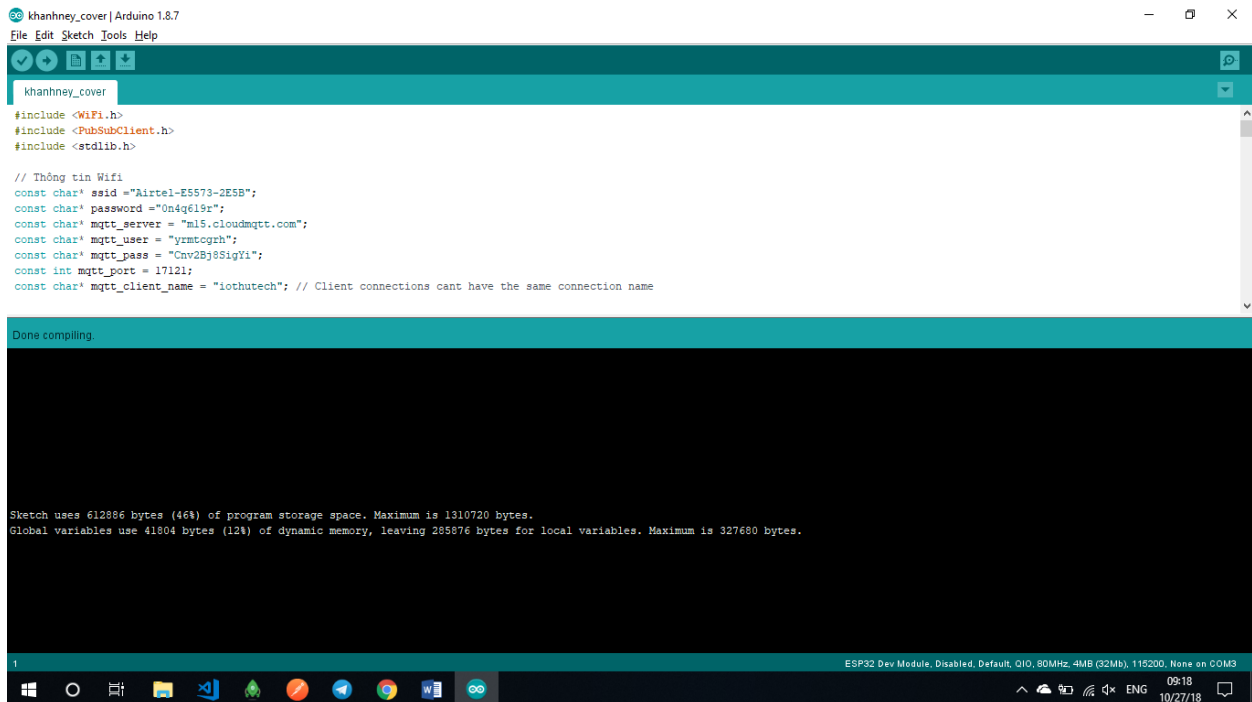
### Đề Tài: Bật Tắt Đèn Led

#### I. GIỚI THIỆU MẠCH ESP32

- Trong ESP 32 có tất cả 34 chân GPIO
  - GPIO 00 – GPIO 19
  - GPIO 21 – GPIO 23
  - GPIO 25 – GPIO 27
  - GPIO 32 – GPIO 39
- Lưu ý:
  - Không bao gồm các chân **20, 24, 28, 29, 30 và 31**
  - Các chân GPIO 32 – GPIO 39 chỉ thiết lập ở chế độ Input và không có chức năng Pullup hoặc Pulldown cho phần mềm.
  - Các chân GPIO 06 – GPIO 11 thường được dùng để giao tiếp với thẻ nhớ ngoài thông qua giao thức SPI nên hạn chế sử dụng để thiết lập IO.
- Sơ đồ chân ESP 32



- Các bạn lấy code tại link Github dưới đây:  
[https://github.com/iothutech/iot/blob/master/Lab4\\_ESP32\\_mqtt\\_pubsub/ControlLedArduino.txt](https://github.com/iothutech/iot/blob/master/Lab4_ESP32_mqtt_pubsub/ControlLedArduino.txt)
- Sau đó chúng ta kiểm tra code xem còn lỗi hay không rồi mới cập nhật code cho ESP
- Dưới đây là màn hình kiểm tra code thành công và không phát sinh ra lỗi.



The screenshot shows the Arduino IDE interface. The top menu bar includes 'File', 'Edit', 'Sketch', 'Tools', and 'Help'. The toolbar contains icons for opening, saving, and running the sketch. The main text area displays the code for 'khanhney\_cover' with the following content:

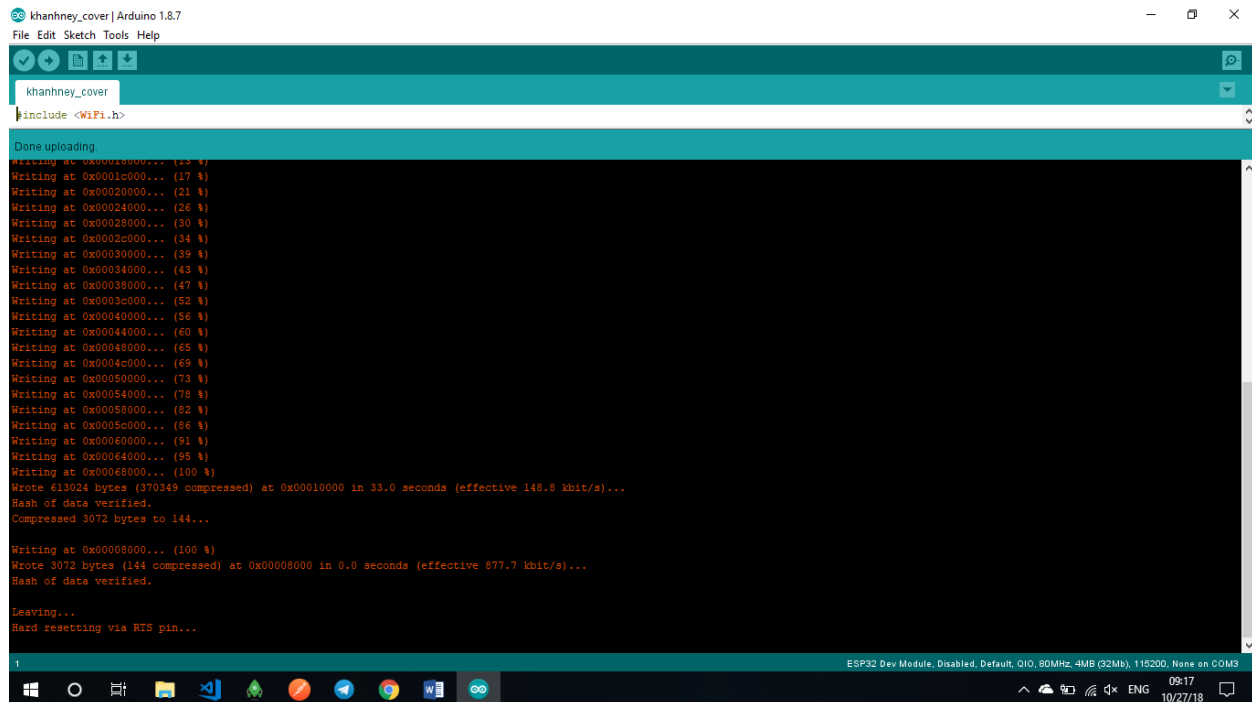
```
#include <WiFi.h>
#include <PubSubClient.h>
#include <stdlib.h>

// Thông tin Wifi
const char* ssid = "Airtel-E5573-2E5B";
const char* password = "0n4q6l9r";
const char* mqtt_server = "m15.cloudmqtt.com";
const char* mqtt_user = "yrmtogrh";
const char* mqtt_pass = "Cnv2Bj85igYi";
const int mqtt_port = 17121;
const char* mqtt_client_name = "iothutech"; // Client connections cant have the same connection name
```

Below the code, a status bar indicates 'Done compiling.' The bottom status bar shows 'ESP32 Dev Module, Disabled, Default, QIO, 80MHz, 4MB (32Mb), 115200, None on COM3'. The system tray at the bottom right shows the time as 09:18 on 10/27/18.

Sketch uses 612886 bytes (46%) of program storage space. Maximum is 1310720 bytes.  
Global variables use 41804 bytes (12%) of dynamic memory, leaving 285876 bytes for local variables. Maximum is 327680 bytes.

- Sau đó chúng ta thực hiện công đoạn cập nhật code cho mạch
- Dưới đây là màn hình hiển thị cập nhật code thành công.



```
khanhney_cover | Arduino 1.8.7
File Edit Sketch Tools Help

khanhney_cover
#include <WiFi.h>

Done uploading.
Writing at 0x00000000... (13 %)
Writing at 0x0001c000... (17 %)
Writing at 0x00020000... (21 %)
Writing at 0x00024000... (26 %)
Writing at 0x00028000... (30 %)
Writing at 0x0002c000... (34 %)
Writing at 0x00030000... (39 %)
Writing at 0x00034000... (43 %)
Writing at 0x00038000... (47 %)
Writing at 0x0003c000... (52 %)
Writing at 0x00040000... (56 %)
Writing at 0x00044000... (60 %)
Writing at 0x00048000... (65 %)
Writing at 0x0004c000... (69 %)
Writing at 0x00050000... (73 %)
Writing at 0x00054000... (78 %)
Writing at 0x00058000... (82 %)
Writing at 0x0005c000... (86 %)
Writing at 0x00060000... (91 %)
Writing at 0x00064000... (95 %)
Writing at 0x00068000... (100 %)
Wrote 613024 bytes (370349 compressed) at 0x00010000 in 33.0 seconds (effective 148.8 kbit/s)...
Hash of data verified.
Compressed 3072 bytes to 144...

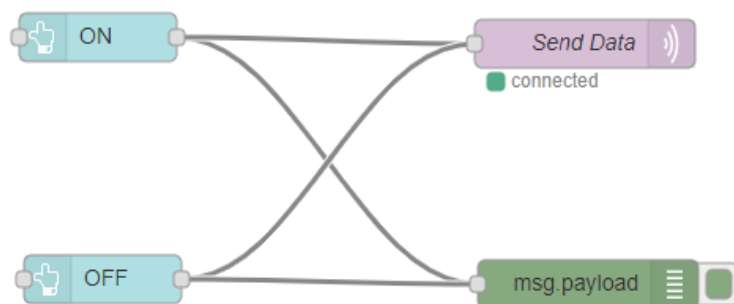
Writing at 0x00000000... (100 %)
Wrote 3072 bytes (144 compressed) at 0x00000000 in 0.0 seconds (effective 877.7 kbit/s)...
Hash of data verified.

Leaving...
Hard resetting via RTS pin...

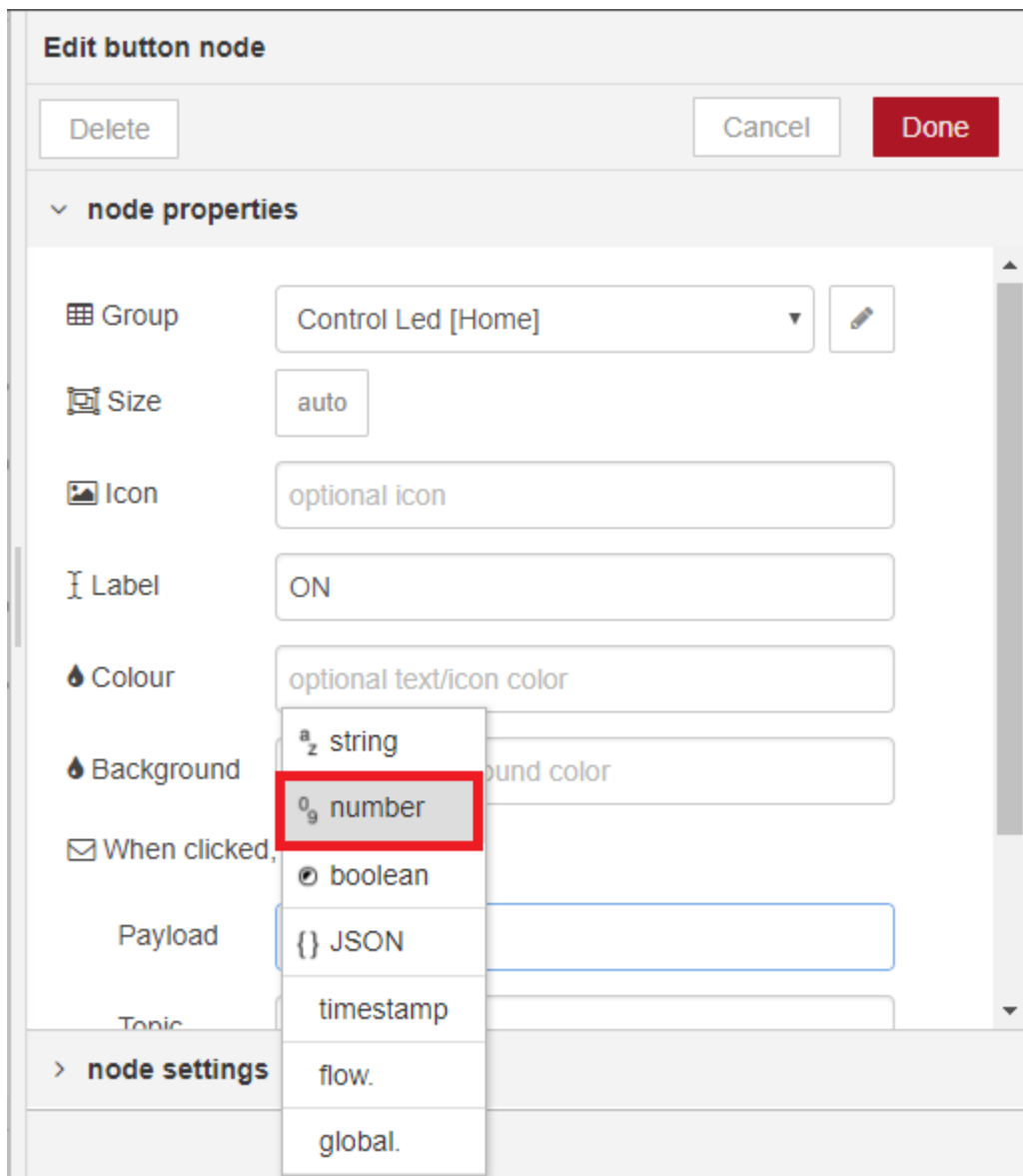
ESP32 Dev Module, Disabled, Default, QIO, 80MHz, 4MB (32Mb), 115200, None on COM3
09:17
10/27/18
```

## 2. Chạy Trên Node-Red

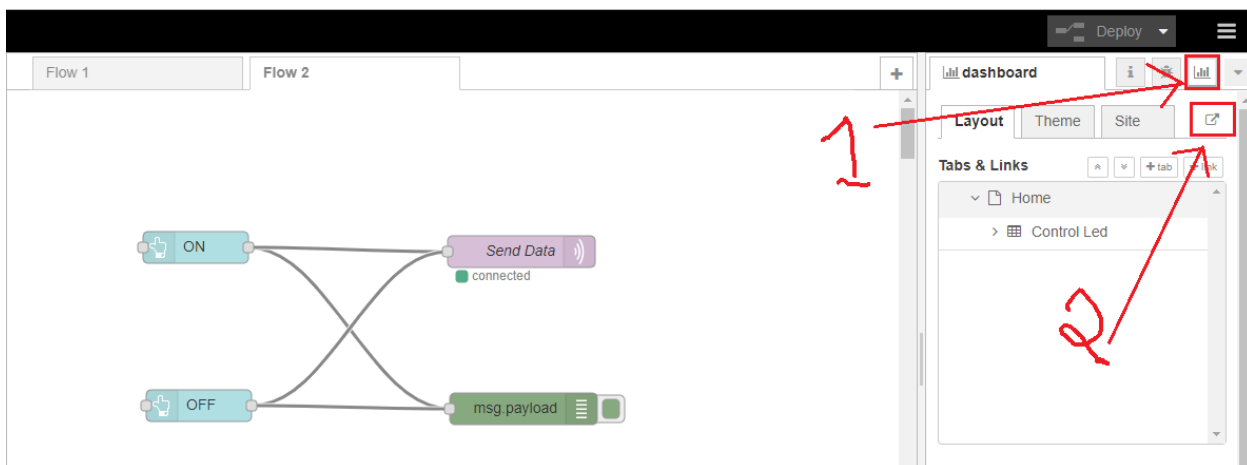
- Chúng ta sẽ cấu hình Node-Red như màn hình dưới.
- Các node sử dụng trong chương trình này bao gồm: MQTT in, MQTT out, Debug, Button.
- Sau đó chúng ta cấu hình cho các node MQTT bình thường như các Lab thực hành trước.



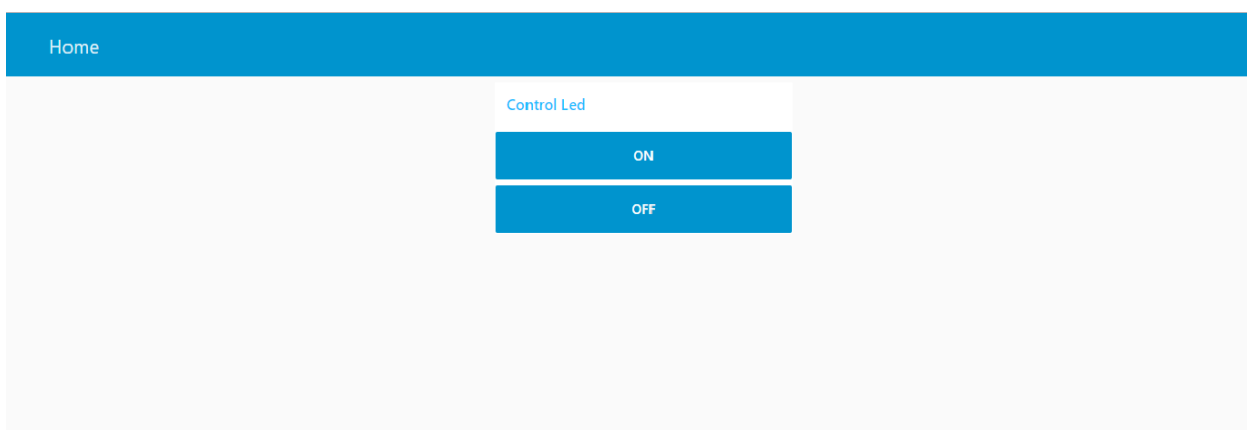
- Bước tiếp theo chúng ta bấm vào biểu tượng button ON để cấu hình cho Button đó. Đầu tiên chúng ta đặt tên cho Button đó là ON, sau đó lựa chọn kết quả của payload là kiểu Number và đặt một con số phù hợp với điều kiện tắt đèn như ở code. Hiện tại code của mình đang để mức trên 600 thì thực hiện bật đèn. Nên mình thiết lập là 7000.



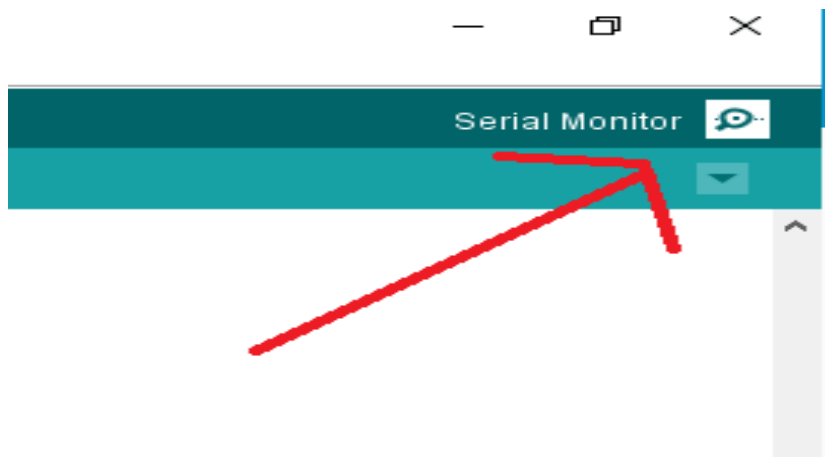
- Tiếp đó bấm vào biểu tượng dưới đây để mở Dashboard điều khiển bóng đèn.



- Dưới đây sẽ là giao diện Dashboard điều khiển và chúng ta bắt đầu test thôi.



- Trong khi test thì các bạn mở Serial Monitor bên tay phải của ứng dụng Arduino IDE để xem dữ liệu đã được nhận chưa nhé.



- Và đây là kết quả: Chúc các bạn thành công.

The screenshot shows the Arduino IDE interface. The top-left pane displays the 'ControlLed' sketch. The bottom-left pane shows the upload progress: 'Done uploading.', 'Hash of data verified.', 'Writing at 0x00000000... (100 %)', 'Wrote 3072 bytes (144 compressed) at 0x00000000 in 0.0 seconds (effective 910.2 kbit/s)...', 'Hash of data verified.', 'Leaving...', 'Hard resetting via RTS pin...'. The right pane shows the Serial Monitor output for COM3 at 115200 baud. The output text is as follows:

```
3fff001c, len: 952
load: 0x40078000, len: 6084
load: 0x40080000, len: 7936
entry 0x40080310

Connecting to kof-com
.....
WiFi connected
IP address: 192.168.0.102
Connecting to MQTT...
connected
HELLO KHANHNEY_TESTY/
7000
HIGH
7000y
HELLO KHANHNEY_TESTY/
7000
HIGH
7000y
HELLO KHANHNEY_TESTY/
7000
HIGH
7000y
HELLO KHANHNEY_TESTY/
7000
HIGH
7000y
HELLO KHANHNEY_TESTY/
7000
HIGH
7000y
```

The Serial Monitor window has checkboxes for 'Autoscroll' and 'Show timestamp', and a dropdown menu for 'Newline' set to '\n'. The bottom status bar shows the time as 16:40 on 11/02/18.