Nút nhấn

Nút nhấn sẽ giúp việc ESP8266 khởi động một hành động nào đó khi cần thiết. Trong nhiều ứng dụng chúng ta hầu như đều cần những kích hoạt từ bên ngoài. Xuyên suốt cuốn sách này, sẽ dùng nút nhấn để kích hoạt chạy các ứng dụng mẫu cũng như đèn LED để thông báo các trạng thái. Trong phần này, nhấn nút đèn LED sẽ chuyển trạng thái (từ sáng -> tắt và ngược lại).

Đây là ví dụ đơn giản, trong thực tế việc xử lý nút nhấn khá phiền phức. Bởi vì nút nhấn vật lý khi được nhấn sẽ tạo ra hàng loạt các xung lên xuống (nhiễu, bouncing...). Thường thì chỉ cần đảm bảo mức Logic của chân đo được đã được giữ ổn định trong khoảng 100 mili giây là được xem đã ổn định.

Ngoài cách dùng ngắt để xác định nút nhấn có được nhấn hay không - cách này sẽ tiết kiệm tài nguyên tính toán của CPU, nó chỉ được gọi khi có sự kiện sảy ra, thì còn một cách nữa là hỏi vòng: Cách này đòi hỏi CPU liên tục kiểm tra xem mức Logic của nút nhấn. Đồng thời việc đáp ứng cũng không nhanh bằng sử dụng ngắt.

> **Ghi chú**
>
> Yêu cầu: Nhấn nút (GPIO0)thì chớp tắt đèn LED (GPIO6) và in ra cổng Serial

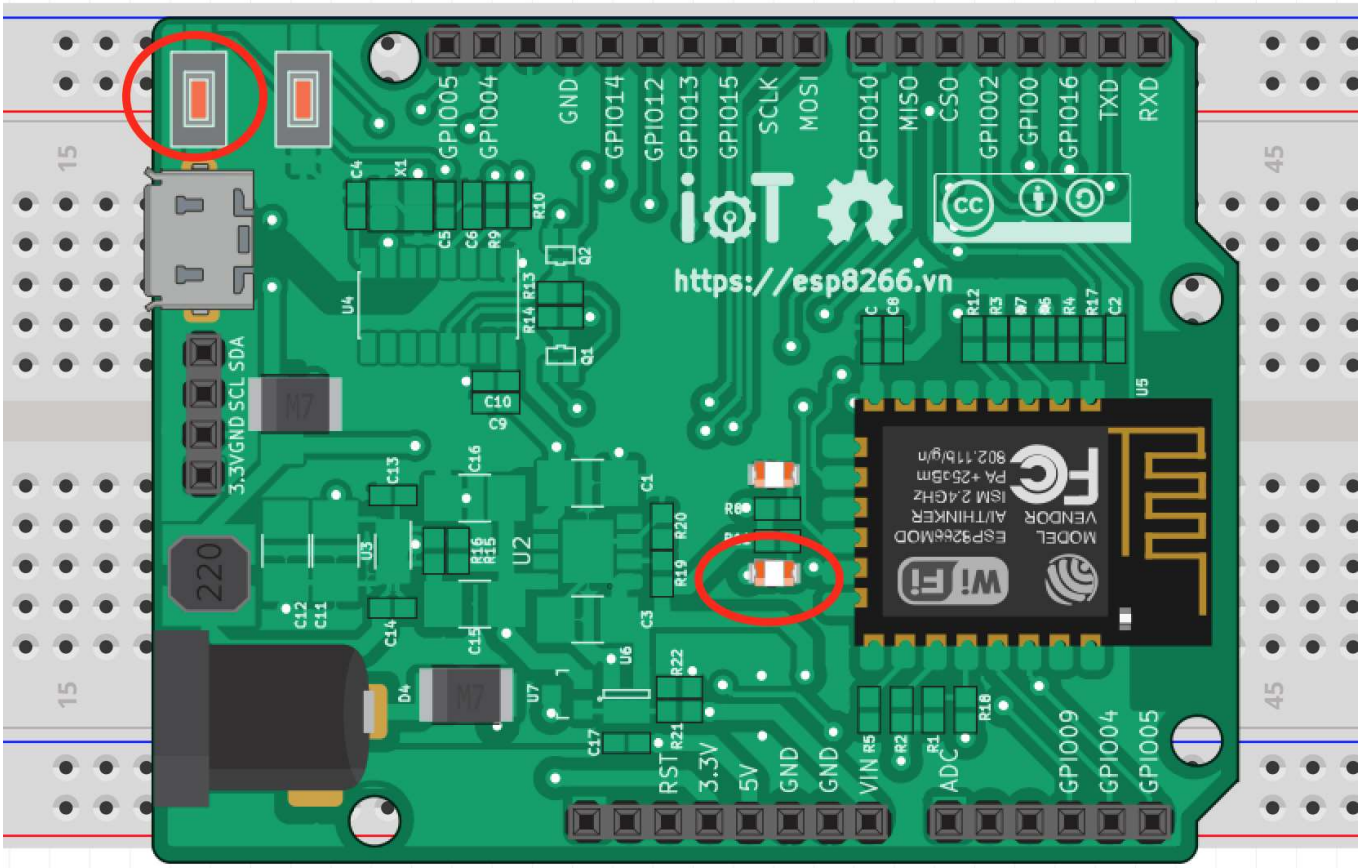
Demo

Chuẩn bị

Tên board mạch	Link
Board IoT Wifi Uno	https://iotmaker.vn/esp8266-iot-wifi-uno.html

Đấu nối

Nút nhấn và LED board `IoT WiFi Uno` được khoanh tròn như trong hình


https://esp8266.vn
GPIO5
GPIO4
GND
GPIO14
GPIO12
GPIO13
GPIO15
SCLK
MOSI
GPIO10
MISO
CS0
GPIO2
GPIO0
GPIO16
TXD
RXD
3.3V
GND
SCL
SDA
220
U2
RST
3.3V
5V
GND
GND
VIN
ADC
GPIO9
GPIO4
GPIO5

Code

```
int ledPin = 16;                // LED connected to digital pin 16
int btnPin = 0;                 // BUTTON connected to digital pin 0
int ledState = LOW;

void blink()
{
  if (ledState == LOW) {
    ledState = HIGH;
  } else {
    ledState = LOW;
  }
  digitalWrite(ledPin, ledState);
  Serial.println("Pressed, value=" + String(ledState));
}
void setup()
{
  pinMode(ledPin, OUTPUT);      // sets the digital pin as output
  pinMode(btnPin, INPUT);       // sets the digital pin as input
  pinMode(btnPin, INPUT_PULLUP); //pull-up button
  attachInterrupt(btnPin, blink, FALLING);
  Serial.begin(115200);
}

void loop()
{

}
```

Digital IO

Tên chân trong Arduino (Pin number) giống với thứ tự chân của ESP8266. `pinMode`, `digitalRead`, và `digitalWrite` đều sử dụng Pin Number như nhau, ví dụ như đọc GPIO2, gọi hàm `digitalRead(2)`.

Chân GPIO0..15 có thể là `INPUT`, `OUTPUT`, hay `INPUT_PULLUP`. Chân GPIO16 có thể là `INPUT`, `OUTPUT` hay `INPUT_PULLDOWN_16`. Khi khởi động, tất cả các chân sẽ được cấu hình là `INPUT`.

Mỗi chân có thể phục vụ cho một tính năng nào đó, ví dụ `Serial`, `I2C`, `SPI`. Và tính năng đó sẽ được cấu hình đúng khi sử dụng thư viện. Hình bên dưới thể hiện sơ đồ chân đối với module ESP-12 phổ biến.

GPIO6 và GPIO11 không được thể hiện bởi vì nó được sử dụng cho việc kết nối với Flash. Việc sử dụng 2 chân này có thể gây lỗi chương trình.

> **Ghi chú**
>
> Một số board và module khác (ví dụ ESP-12ED, NodeMCU 1.0) không có GPIO9 và GPIO11, họ sử dụng với chế độ `DIO` cho Flash, trong khi ESP12 chúng ta nói bên trên sử dụng chế độ `QIO`

Ngắt GPIO hỗ trợ thông qua các hàm `attachInterrupt`, `detachInterrupt` Ngắt GPIO có thể gán cho bất kỳ GPIO nào, ngoại trừ `GPIO16`. Đều hỗ trợ các ngắt tiêu chuẩn của Arduino như: `CHANGE`, `RISING`, `FALLING`.