



Search

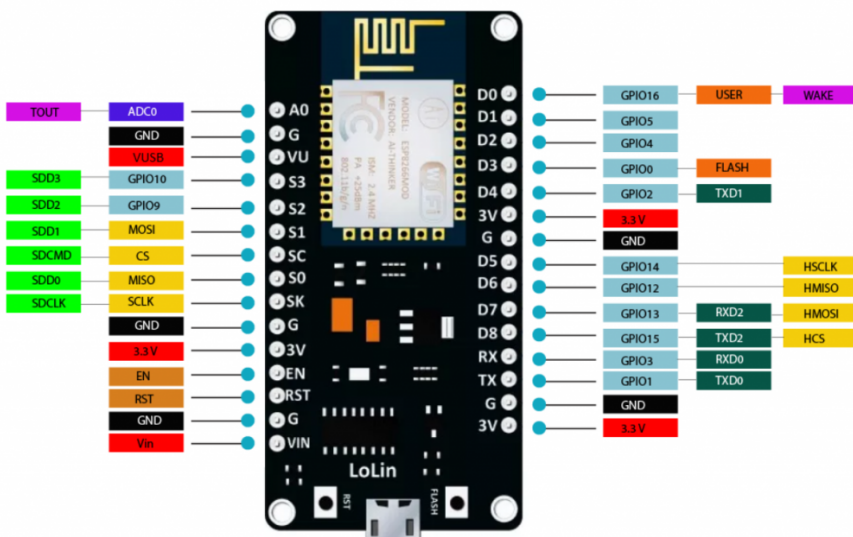

[GIỚI THIỆU](#) [DỰ ÁN](#) [ĐÀO TẠO](#) [HỢP TÁC](#) [CỘNG ĐỒNG](#) [LIÊN HỆ](#)

Chức năng nhập xuất tín hiệu số (GPIO) trên NodeMCU, ESP32 Dev Kit và những lưu ý khi sử dụng

GPIO (General Purpose Input/Output) là một chức năng cơ bản và phổ biến của vi điều khiển, nó bao gồm các chân đầu vào và đầu ra tín hiệu số được lập trình bởi người dùng. Bài viết này sẽ giúp các bạn hiểu rõ hơn về chức năng GPIO trên board NodeMCU và ESP32 Dev Kit và những lưu ý quan trọng khi sử dụng.

1. Sơ đồ chân GPIO và những lưu ý khi sử dụng:

NodeMCU:



Chúng ta biết rằng ở mỗi chân trên vi điều khiển có thể thực hiện nhiều chức năng khác nhau, NodeMCU có tổng cộng 13 chân GPIO tuy nhiên một số chân được dùng cho những mục đích quan trọng khác vì vậy chúng ta phải lưu ý khi sử dụng như sau:

- Tất cả các GPIO đều có trở kéo lên nguồn bên trong (ngoại trừ GPIO16 có trở kéo xuống GND). Người dùng có thể cấu hình kích hoạt hoặc không kích hoạt trở kéo này.
- GPIO1 và GPIO3: hai GPIO này được nối với TX và RX của bộ UART0, NodeMCU nạp code thông qua bộ UART này nên tránh sử dụng 2 chân GPIO này.
- GPIO0, GPIO2, GPIO15: đây là các chân có nhiệm vụ cấu hình mode cho ESP8266 điều khiển quá trình nạp code nên bên trong NodeMCU (có tên gọi là strapping pins) có các trở kéo để định sẵn mức logic cho chúng như sau: GPIO0: HIGH, GPIO2: HIGH, GPIO15: LOW. Vì vậy khi muốn sử dụng các chân này ở vai trò GPIO cần phải thiết kế một nguyên lý riêng để tránh xung đột đến

Like Fanpage để cập nhật thông tin!

TAPIT - Learning, Research and St Community
Đã thích 5.9K lượt thích

Bạn và 102 người bạn khác thích nội dung này

Device → AWS IoT Core → SNS Rule → Amazon Simple Notification Service

TAPIT - Learning, Research and St Community
on Sunday

Trong các ứng dụng IoTs, chức năng gửi cảnh báo (Push Notification) khi có sự kiện là rất quan trọng. Bài viết này hướng dẫn các bạn cách thực hiện chức năng cảnh báo qua Email kết hợp giữa AWS IoT Core và dịch vụ Amazon SNS.

Chi tiết bài viết tại: <https://tapit.vn/huong-dan-bao-qua-email-su-dung-.../>

56 Bình luận 16

TAPIT - Learning, Research and St Community
last Saturday

Xem thêm các bài viết từ TAPIT

[Hướng dẫn gửi cảnh báo qua Email sử dụng AWS IoT Core và Amazon SNS](#)

[Preprocessors và Macros trong chương trình C](#)

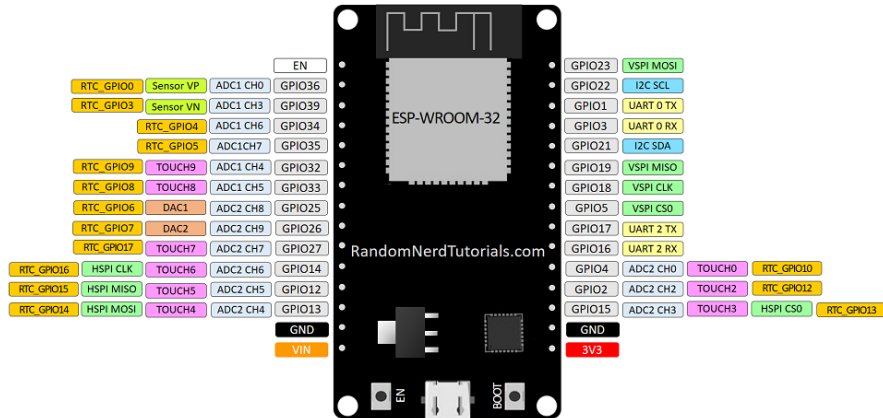
[Cập nhật chương trình từ xa trên vi điều khiển STM32 - Firmware Over The Air](#)

quá trình nạp code. Các bạn có thể tham khảo nguyên lý thiết kế mạch nạp ở bài viết [này](#)

- GPIO9, GPIO10: hai chân này được dùng để giao tiếp với External Flash của ESP8266 vì vậy cũng không thể dùng được (đã test thực nghiệm).

Như vậy, các GPIO còn lại: GPIO 4, 5, 12, 13, 14, 16 có thể sử dụng bình thường.

ESP32 DevKit V1 – DOIT (version 30 pins):



ESP32 có rất nhiều kit phát triển khác nhau nên các bạn lưu ý bài hướng dẫn này chỉ viết cho ESP32 DEVKIT V1 – DOIT loại 30 chân, các kit khác tương tự. Trên kit có tổng cộng 25 chân GPIO tuy nhiên cũng giống với NodeMCU, khi dùng GPIO trên ESP32 DEVKIT cần lưu ý:

- GPIO34, 35, 36, 39 chỉ dùng được chức năng INPUT, không dùng được chức năng OUTPUT.
- GPIO1, GPIO3: Giống như NodeMCU 2 chân này cũng được nối đến bộ UART0 và ESP32 Dev Kit cũng nạp code thông qua bộ UART này nên tránh sử dụng 2 chân này cho chức năng I/O.
- GPIO0, 2, 4, 5, 12, 15 đã được định sẵn mức logic bên trong module ESP32 để phục vụ quá trình nạp code, nên tránh sử dụng các GPIO này.

Ngoài ra đối với những bạn thiết kế mạch sử dụng module ESP32 rời (module tích hợp trên ESP32 Dev Kit) thì ngoài lưu ý ở trên còn lưu ý thêm các chân từ GPIO6 đến GPIO11. Các chân này dùng để giao tiếp SPI với External Flash nên không thể dùng được chức năng I/O và trên ESP32 Dev Kit đã ẩn đi các chân này.

Một lưu ý quan trọng khác cho cả NodeMCU và ESP32 Dev Kit là mức điện áp INPUT tối đa mà các chân GPIO có thể hoạt động được là 3.6V (khác với các board Arduino là 5V) vì vậy nếu tác động mức điện áp lớn hơn 3.6V vào chân INPUT sẽ làm hỏng chân GPIO.

2. Khai báo và sử dụng chức năng GPIO

Để lập trình sử dụng chức năng GPIO trên 2 board NodeMCU và ESP32 Dev Kit, có thể sử dụng các hàm sau:

`pinMode(PIN_NUMBER, MODE);`

(FOTA) – P5

[Cập nhật chương trình từ xa trên vi điều khiển STM32 – Firmware Over The Air \(FOTA\) – P4](#)

[Tìm hiểu về hệ sinh thái STM32Cube](#)

Xem thêm các chủ đề khác

GIỚI THIỆU

DỰ ÁN

ĐÀO TẠO

[Khóa VDK MSP430](#)

[Khóa Internet of Thing](#)

[Khóa lập trình Arduino chuyên sâu](#)

[Khóa vi điều khiển STM32](#)

[Khóa Xử lý ảnh](#)

[Tài liệu](#)

[Học viên](#)

[Kiểm tra](#)

HỢP TÁC

CỘNG ĐỒNG

[Các cuộc thi khoa học](#)

[Hội nghị – hội thảo](#)

[Hoạt động ngoại khóa](#)

HỆ THỐNG NHÚNG

[Vi điều khiển](#)

[Vi điều khiển MSP430](#)

[Vi điều khiển lõi ARM](#)

[Arduino Board](#)

[Máy tính nhúng](#)

INTERNET OF THING

[Bluetooth](#)

PHẦN CỨNG

[Thiết kế PCB](#)

[Thiết kế 3D](#)

DSP-AI

LIÊN HỆ

Chức năng: Đề khai báo sử dụng một chân là Input hoặc Output.

PIN_NUMBER: số thứ tự chân GPIO

MODE: INPUT, OUTPUT, INPUT_PULLUP.

digitalRead(PIN_NUMBER);

Chức năng: đọc giá trị digital tại chân *PIN_NUMBER*

PIN_NUMBER: số thứ tự chân GPIO

Giá trị trả về: 0 (tương ứng LOW) hoặc 1 (tương ứng HIGH)

digitalWrite(PIN_NUMBER, VALUE);

Chức năng: ghi mức điện áp *VALUE* ra chân *PIN_NUMBER*

PIN_NUMBER: số thứ tự chân GPIO

VALUE: LOW hoặc HIGH

attachInterrupt(INTERUPT_PIN, ISR, MODE);

Chức năng: khai báo sử dụng ngắt ngoài trên chân *INTERUPT_PIN*

INTERUPT_PIN:

Đối với NodeMCU: tất cả các GPIO ngoại trừ GPIO16 đều có hỗ trợ interrupt

Đối với ESP32 DEVKIT: tất cả các GPIO đều hỗ trợ interrupt

ISR: chương trình ngắt, là một hàm do người dùng tự định nghĩa, hàm này sẽ được gọi khi có ngắt xảy ra. Lưu ý: hàm ngắt là hàm không có giá trị truyền vào và không có giá trị trả về.

MODE: là thông số định nghĩa khi nào ngắt được kích hoạt. Có 4 giá trị sau:

LOW: kích hoạt ngắt khi giá trị logic tại chân là mức thấp

HIGH: kích hoạt ngắt khi giá trị logic tại chân là mức cao

CHANGE: kích hoạt ngắt bất cứ khi nào giá trị logic tại chân thay đổi

RISING: kích hoạt ngắt khi mức logic của chân thay đổi từ thấp sang cao

FALLING: kích hoạt ngắt khi mức logic của chân thay đổi từ cao sang thấp

detachInterrupt(INTERUPT_PIN);

Chức năng: tắt chức năng ngắt tại *INTERUPT_PIN*

Ví dụ: chương trình sử dụng chức năng interrupt để đảo trạng thái led

```
1 #define INTERRUPT_PIN 0
2 #define LED_PIN 16
3 void setup()
4 {
```

```
5 pinMode(INTERRUPT_PIN, INPUT_PULLUP);
6 attachInterrupt(INTERRUPT_PIN, blinkLed, FALLING);
7 pinMode(LED_PIN, OUTPUT);
8 }
9 void loop()
10 {
11
12 }
13 void blinkLed()
14 {
15   digitalWrite(LED_PIN, !digitalRead(LED_PIN));
16 }
```

Tài liệu tham khảo thêm:

- [Datasheet ESP8266 SoC](#)
- [Datasheet ESP32 SoC](#)
-

Xem thêm: [Tổng hợp hướng dẫn Internet of Things với NodeMCU ESP8266 và ESP32](#)

Nhóm TAPIT IoTs

Tagged on: ESP32 ESP8266 GPIO NODEMCU

 Khai Le

 13/06/2019

 INTERNET OF THING

 No Comments


[← Hướng dẫn nạp firmware ESP32 và ESP8266 bằng ESPTOOL, không cần IDE Arduino](#)

[STM32F411 - Tìm hiểu cấu trúc và lập trình nhập xuất GPIO cơ bản →](#)

Facebook Comments:

0 bình luận

Sắp xếp theo Mới nhất



Thêm bình luận...

Plugin bình luận trên Facebook

CÔNG TY TNHH TAPIT

Công ty TNHH Kỹ thuật TAPIT chuyên kinh doanh các thiết bị điện tử; Cung cấp dịch vụ tư vấn, nghiên cứu, thiết kế và thi công các giải pháp công nghệ; Đồng thời có các chương trình đào tạo kỹ thuật ngắn hạn trực tiếp cho sinh viên và kỹ sư.

CỘNG ĐỒNG TAPIT

Cộng đồng TAPIT được thành lập vào 01/2016, là một môi trường quy tụ các sinh viên và kỹ sư đến từ nhiều lĩnh vực khoa học kỹ thuật khác nhau. Đến với cộng đồng, các thành viên cùng học tập, nghiên cứu, chia sẻ các kiến thức, kỹ năng và các cơ hội phát triển bản thân.

CÁC LIÊN KẾT

[Các bộ thiết bị thực hành giáo dục](#)

[Các khóa đào tạo thường xuyên](#)

[Kênh Video hướng dẫn kỹ thuật](#)

[Chợ Linh kiện điện tử Đà Nẵng](#)

LIÊN HỆ

Mr. Nguyễn Huỳnh Nhật Thương
SĐT: 0981001119
Email: nhatthuongqn@gmail.com
Facebook: [Thuong Nguyen](#)

TAPIT
Build the future with us!