

LẬP TRÌNH ESP32

Bài 5: Lập trình ESP32 ngắt ngoài EXTI

POSTED ON 29/06/2021 BY KHUÊ NGUYỄN

**29
Th6**

ESP32 và Platform IO

Bài 5: Lập trình ESP32 ngắt ngoài EXTI

Trong bài này chúng ta sẽ cùng học lập trình ESP32 ngắt ngoài hay EXTI (External Interrupts). Hiểu cách hoạt động của một ngắt trong chương trình nhúng, cách lập trình để tránh sự xung đột khi dùng ngắt.

Bài 5 trong Serie **Học lập trình ESP32 từ A tới Z**



Mục Lục

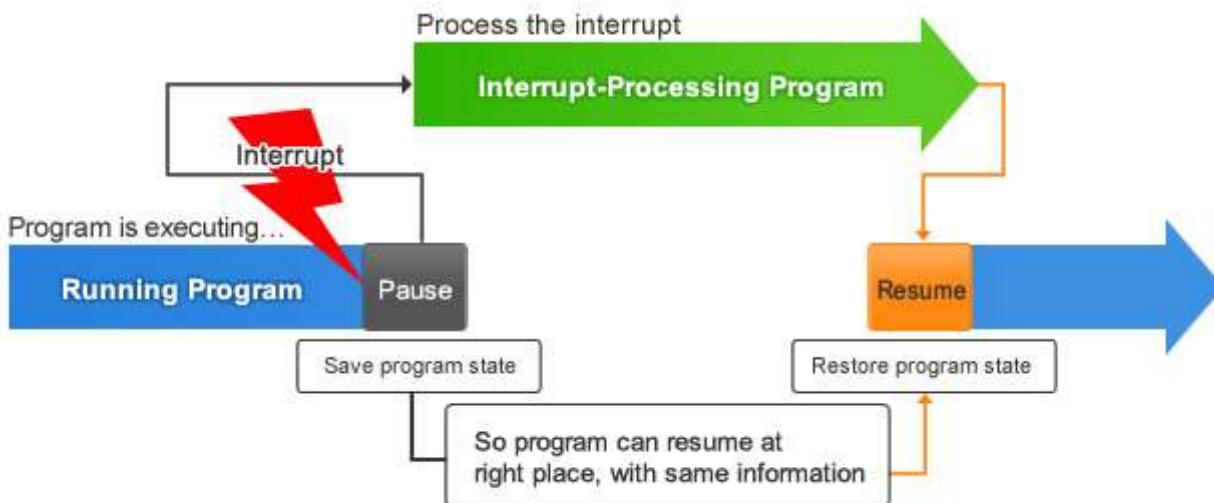
1. Ngắt ngoài là gì?
2. Ngắt ngoài trong ESP32
3. Lập trình ESP32 ngắt ngoài điều khiển led bằng nút nhấn
 - 3.1. Sơ đồ nguyên lý
 - 3.2. Code và giải thích code
 - 3.3. Nạp code và kết quả
4. Kết
 - 4.1. Related posts:

Ngắt ngoài là gì?

Đầu tiên chúng ta phải hiểu về khái niệm ngắt và tuần tự trước đã.

- Tuần tự: Một chương trình C,Cpp sẽ được xử lý tuần tự theo chiều từ trên xuống dưới và sẽ bị ảnh hưởng bởi các câu lệnh rẽ nhánh và loop
- Ngắt: Chương trình sẽ bị thoát ra khỏi quá trình tuần tự, xử lý xong các lệnh trong ngắt, sau đó mới quay lại xử lý tiếp

Nói một cách đơn giản, Ngắt là một công việc được ưu tiên làm trước, khi xử lý xong mới được làm các việc khác. Các sự kiện ngắt có thể đến từ nhiều nguồn khác nhau như: ADC, IO, UART, TIMER...



Ngắt là một sự kiện được ưu tiên xử lý trước

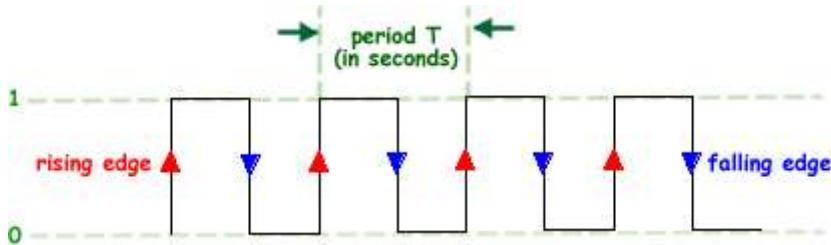
Ngắt ngoài EXTI chính là một sự kiện ngắt được sinh ra với nguồn từ các chân IO của ESP32.

Ngắt ngoài trong ESP32

Khác với các chip Arduino truyền thống, trên ESP32 mọi chân Input đều có thể được cài đặt để sử dụng với ngắt ngoài. Nó không bị giới hạn như Arduino Uno hoặc các loại tương đương là chỉ có 2 chân có thể được sử dụng làm ngắt ngoài

Ngắt ngoài có các kiểu ngắt khác nhau tương ứng cho tín hiệu xung trên chân GPIO đó, cụ thể:

LOW	Interrupt được kích hoạt khi chân ở mức LOW
HIGH	Interrupt được kích hoạt khi chân ở mức HIGH
CHANGE	Interrupt được kích hoạt khi chân thay đổi mức, từ HIGH sang LOW, hoặc LOW sang HIGH
FALLING	Interrupt được kích hoạt khi chân thay đổi mức, từ HIGH sang LOW
RISING	Interrupt được kích hoạt khi chân thay đổi mức, từ LOW sang HIGH



Để lập trình ngắt ngoài trên ESP32 chúng ta sử dụng 2 hàm chính đó là:

```
attachInterrupt(GPIOPin, ISR, Mode);
```

Đây là hàm chức năng được sử dụng để cài đặt ngắt ngoài cho 1 chân cụ thể.

Hàm này có ba đối số đầu vào:

GPIOPin – Là chân GPIO được chỉ định interrupt.

ISR – Là tên của hàm sẽ được gọi khi có interrupt

Mode – Khai báo chế độ interrupt được sử dụng, các chế độ tương ứng với các kiểu ngắt bên trên đã nêu.

```
detachInterrupt(GPIOPin);
```

Đây là hàm chức năng để xóa các cấu hình ngắt trên chân đó đi. Đối số truyền vào là chân được chỉ định ngắt.

```
void IRAM_ATTR ISR() {
    Statements;
}
//example
void IRAM_ATTR NgatNgoai(){
    //do some things
}
```

Đây là hàm được gọi khi có một sự kiện ngắt xảy ra.

ISR : Interrupts Service Routine hay định tuyến dịch vụ, là tên hàm sẽ được gọi khi ngắt xảy ra

IRAM_ATTR khai báo cho trình biên dịch biết rằng hàm sẽ được biên dịch và đặt trong RAM (IRAM) của ESP32. Ngược lại hàm sẽ được đặt trong FLASH.

Một hàm khi được đặt trên FLASH sẽ phải chờ **Load** xong mới có thể thực hiện gây tốn thời gian, do đó hàm ISR không nên được đặt trên FLASH.

Lập trình ESP32 ngắt ngoài điều khiển led bằng nút nhấn

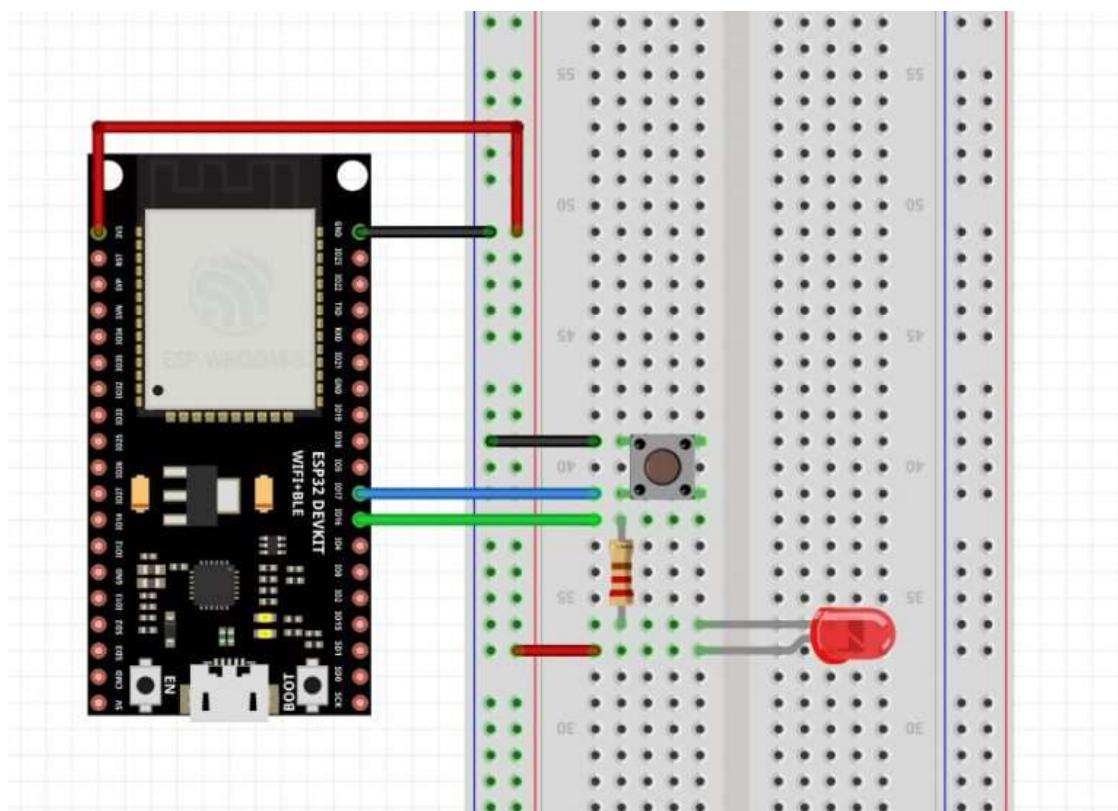
Với ví dụ này chúng ta sẽ điều khiển led bằng nút nhấn, nhưng không cần đọc giá trị nút nhấn như bài 1.

Chuẩn bị

- Breadboard
 - Kit ESP32 dev
 - Trở 200R
 - Led cảm
 - Nút nhấn

Sơ đồ nguyên lý

Chúng ta sử dụng sơ đồ nguyên lý như bài 1.



Sơ đồ nguyên lý ngắt ngoài esp32

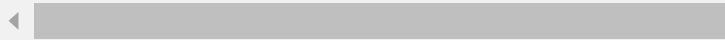
Code và giải thích code

Full code

Full Code

```

02 #include <Arduino.h>
03
04 const int led = 17;
05 const int button = 16;
06 bool ledstates = 0;
07 portMUX_TYPE mux = portMUX_INITIALIZER_UNLOCKED;
08
09 void IRAM_ATTR buttonPush() {
10     portENTER_CRITICAL_ISR(&mux); //Phần quan trọng cần được bảo
11     Serial.println("Button Pushed!!!");
12     digitalWrite(led, ~ledstates);
13     portEXIT_CRITICAL_ISR(&mux); //Cho phép tiếp tục chạy các tâ
14 }
15 void setup() {
16     pinMode(button, INPUT_PULLUP);
17     // cài đặt ngắt vào chân button, kiểu ngắt là falling (xung
18     attachInterrupt(digitalPinToInterrupt(button), buttonPush,
19     // chọn led là đầu ra
20     pinMode(led, OUTPUT);
21     //ghi giá trị ban đầu
22     digitalWrite(led, ledstates);
23 }
24 void loop() {
25     // do nothing
26 }
```



Giải thích code:

Chúng ta khởi tạo các chân led và button, một biến để lưu trữ giá trị của Led

```

const int led = 17;
const int button = 16;
bool ledstates = 0;
```

```
portMUX_TYPE mux = portMUX_INITIALIZER_UNLOCKED;
```

Khai báo một biến mux với kiểu portMUX_TYPE biến này có tác dụng chống sự xung đột giữa các CPU và chương trình chính khi hoạt động.

Tiếp tới khởi tạo 1 hàm được gọi khi ngắt xảy ra:

```
void IRAM_ATTR buttonPush() {
    portENTER_CRITICAL_ISR(&mux); //Phần quan trọng cần được bảo vệ khỏi mọi truy c
    Serial.println("Button Pushed!!!");
    digitalWrite(led, ~ledstates);
    portEXIT_CRITICAL_ISR(&mux); //Cho phép tiếp tục chạy các task khác
}
```

Trong hàm này, chúng ta sẽ đảo trạng thái led, và in ra màn hình dòng chữ nút đã được bấm

Trong setup chúng ta khởi tạo Ngắt, chân Led và ghi giá trị ban đầu cho Led

```
Serial.begin(9600);
pinMode(button, INPUT_PULLUP);
// cài đặt ngắt vào chân button, kiểu ngắt là falling (xung xuống), hàm gọi khi
attachInterrupt(digitalPinToInterrupt(button), buttonPush, FALLING);
// chọn led là đầu ra
pinMode(led, OUTPUT);
//ghi giá trị ban đầu
digitalWrite(led, ledstates);
```

Trong loop không cần làm gì cả.

Nạp code và kết quả

Nhấn nút upload trên thanh công cụ nạp, sau đó nhấn nút để thay đổi trạng thái led nhé

Kết

Lập trình ESP32 ngắt ngoài rất quan trọng trong quá trình code một dự án nhúng, ngắt rất tiện lợi cho việc bắt các sự kiện tức thời, không cần CPU phải gọi lệnh kiểm tra, thế nên rất được ưa chuộng

Nếu thấy hay hãy chia sẻ bài viết này tới bạn bè nhé. Đừng quên tham gia nhóm **Nghiên lập trình** để kết nối với những người cùng đam mê.

5/5 - (1 bình chọn)

Related Posts:

1. [Bài 1: Lập trình ESP32 Webserver chế độ Wifi Station bật tắt Led](#)
2. [Bài 8: Lập trình ESP32 Sleep mode chế độ ngủ tiết kiệm năng lượng](#)
3. [Bài 7: Lập trình ESP32 Touch Pin bật tắt led với một cái chạm tay](#)
4. [Bài 6: Lập trình ESP32 Timer Millis và ngắt Timer](#)
5. [Hướng dẫn cài đặt Platform IO lập trình ESP32](#)
6. [Tổng quan về sơ đồ chân ESP32 và ngoại vi](#)



KHUÊ NGUYỄN

Chỉ là người đam mê điện tử và lập trình. Làm được gì thì viết cho anh em xem thôi. :D

3 THOUGHTS ON “BÀI 5: LẬP TRÌNH ESP32 NGẮT NGOÀI EXTI”



Minh Phạm says:

Vì sao khi nhấn button thì điện áp ở trên chân GPIO lại chuyển từ HIGH sang LOW?

(Mình thấy trong code đang setup interrupt kiểu là FAILING)

06/07/2021 AT 9:49 CHIỀU

TRẢ LỜI



Khuê Nguyễn says:

Vì lúc set nút nhấn là INPUT_PULLUP, nghĩa là lúc rảnh rỗi sẽ được kéo lên, có điện áp là 3.3V – HIGH

06/07/2021 AT 11:43 CHIỀU

TRẢ LỜI



hoàng thắng says:

bạn có tài liệu hướng dẫn ngắt uart trên esp32 ko ạ

11/09/2021 AT 12:08 CHIỀU

TRẢ LỜI

Trả lời

Email của bạn sẽ không được hiển thị công khai. Các trường bắt buộc được đánh dấu *

Bình luận *

Tên *

Email *



Trang web



PHẢN HỒI

Fanpage

 Khuê Nguyễn Creator - Học...
2.754 lượt thích

Đã thích **Chia sẻ**

 **Khuê Nguyễn Creator - Học
Lập Trình Vi Điều Khiển**
khoảng một tháng trước

Lý do thời gian gần đây mình không viết bài
và làm thêm gì cả là đây 😊)
Chính thức ra mắt sản phẩm định vị thông
minh vTag.

Đây là một sản phẩm định vị đa năng với
3 công nghệ định vị WIFI, GPS, LBS kết
hợp với sóng NB-IOT dành riêng cho các
sản phẩm IOT.

Chỉ với 990.000đ chúng ta đã có thể có
sản phẩm để:

- Định vị trẻ em, con cái... Xem thêm



Bài viết khác

Lập trình 8051 - AT89S52

 **Khuê Nguyễn Creator**

[Lập trình 8051](#)**PROTEUS**

Bài 1: Tổng quan về 8051 và chip AT89S51 - 52

Tổng quan về 8051

8051 là một dòng chip nhập môn cho lập trình viên nhúng, chúng được sử...

[ĐỌC THÊM](#)

Lập trình STM32 và CubeMX

**Khuê Nguyễn Creator**

Lập trình STM32 HID Host giao tiếp với chuột và bàn phím

Lập trình STM32 USB HID Host giao tiếp với chuột và bàn phím máy tính

Trong bài này chúng ta sẽ cùng học STM32 HID Host, biến STM32 giống như...

[ĐỌC THÊM](#)



Lộ trình học lập trình nhúng từ A tới Z

Lộ trình học lập trình nhúng từ A tới Z

Lập trình nhúng là một ngành có cơ hội nhưng cũng đòi hỏi nhiều kiến...

3 COMMENTS

[ĐỌC THÊM](#)

Lập trình STM32 và CubeMX

Khuê Nguyễn Creator

ST

STM32 Cube

Lập trình STM32F407 SDIO đọc dữ liệu thẻ nhớ

Lập trình STM32 SDIO đọc ghi dữ liệu vào thẻ nhớ SD card

Trong bài này chúng ta cùng học cách lập trình STM32 SDIO, một chuẩn giao...

[ĐỌC THÊM](#)

Lập trình STM32 và CubeMX



Khuê Nguyễn Creator



Lập trình STM32F407 DAC chuyển đổi số sang tương tự

Lập trình STM32 DAC tạo sóng hình Sin trên KIT STM32F407 Discovery

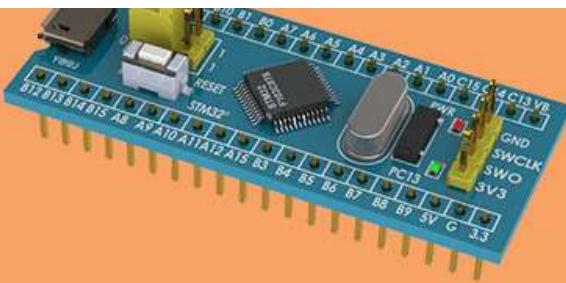
Trong bài này chúng ta sẽ cùng nhau tìm hiểu STM32 DAC với KIT STM32F407VE...

[ĐỌC THÊM](#)

Lập trình STM32 và CubeMX



Khuê Nguyễn Creator



Sử dụng hàm printf để in Log khi Debug trên STM32

Hướng dẫn sử dụng printf với STM32 Uart để in Log trên Keil C

Trong bài này chúng ta sẽ học cách retarget hàm printf của thư viện stdio...

3 COMMENTS

[ĐỌC THÊM](#)

ESP32 và Platform IO



Khuê Nguyễn Creator



Bài 9 WIFI: Lập trình ESP32 OTA nạp firmware trên Internet

Lập trình ESP32 FOTA nạp firmware qua mạng Internet với OTA Drive

Trong bài này chúng ta sẽ học cách sử dụng ESP32 FOTA (Firmware Over The...

4 COMMENTS

[ĐỌC THÊM](#)

Lập trình Nuvoton



Khuê Nguyễn Creator

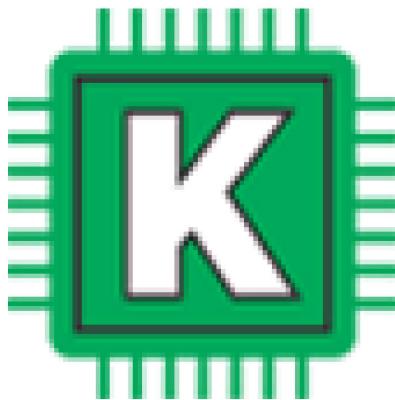


Cài đặt SDC Complier và Code:Blocks IDE

Hướng dẫn cài đặt SDCC và Code:Blocks lập trình Nuvoton

Ở bài này chúng ta sẽ cài đặt các công cụ cần thiết cho việc...

[ĐỌC THÊM](#)



KHUÊ NGUYỄN CREATOR

Chia sẻ đam mê

Blog này làm ra để lưu trữ tất cả những kiến thức, những câu chuyện của mình. Đôi khi là những ý tưởng nhất thời, đôi khi là các dự án tự mình làm. Chia sẻ cho người khác cũng là niềm vui của mình, kiến thức mỗi người là khác nhau, không hẳn quá cao siêu nhưng sẽ có lúc hữu dụng.

DMCA PROTECTED

Liên Kết

Nhóm: Nghiện Lập Trình

Fanpage: Khuê Nguyên Creator

My Shop

Thông Tin

Tác Giả

Chính Sách Bảo Mật



Copyright 2022 © Khuê Nguyễn