

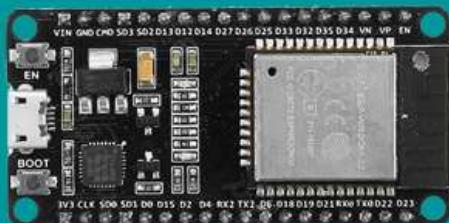
**LẬP TRÌNH ESP32**

Bài 6: Lập trình ESP32 Timer Millis và ngắt Timer

POSTED ON 30/06/2021 BY KHUÊ NGUYỄN

30
Th6

ESP32 và Platform IO

**Khuê Nguyễn Creator**

Bài 6: Lập trình ESP32 Timer Millis và ngắt Timer

Chào các bạn, trong bài này mình sẽ hướng dẫn các bạn lập trình ESP32 Timer với 2 chủ đề chính là sử dụng Millis và tạo ngắt Timer. Bộ định thời hay Timer là ngoại vi rất cơ bản và rất hay được sử dụng trong lập trình vi điều khiển

Bài 6 trong Serie **Lập trình ESP32 từ A tới Z**



Mục Lục



1. Timer là gì?
2. Hàm Delay, Millis và ngắt Timer trong ESP32
 - 2.1. Hàm delay()
 - 2.2. Hàm mills()
 - 2.3. Ngắt Timer
3. Lập trình Esp32 sử dụng Timer hẹn giờ bật tắt Led
 - 3.1. Sơ đồ nguyên lý
 - 3.2. Bật tắt led sử dụng Millis()
 - 3.3. Bật tắt Led sử dụng Ngắt Timer
4. Kết luận
 - 4.1. Related posts:

Timer là gì?

Timer hay bộ định thời là một ngoại vi của vi điều khiển có chức năng đếm xung, từ đó có thể cài đặt được thời gian nhất định. Theo công thức

$$\text{Time} = T(\text{clock}) \times N \text{ (số lần đếm)}$$

Với

- T là chu kỳ của 1 xung cấp vào Timer

- N là số lần đếm của Timer

Trong vi điều khiển Timer thường được sử dụng như bộ hẹn giờ, lặp lại một hành động nhất định khi tới đúng thời điểm.

Hàm Delay, Millis và ngắt Timer trong ESP32

Trong STM32 có 2 hàm phổ biến để hẹn giờ đó là `delay()` và `millis()`

Hàm delay()

Hàm delay cực kì phổ biến và rất hay được sử dụng trong các chương trình nhúng. Bằng cách truyền vào 1 tham số kiểu int, tương ứng với số mili giây (ms) mà MCU sẽ chờ trong đó. Ví dụ:

```
loop()
{
    digitalWrite(LED, HIGH);
    delay(1000);
    digitalWrite(LED, LOW);
    delay(1000);
}
```

Giải thích:

Trong hàm lặp vô hạn loop() chúng ta bật Led bằng digitalWrite(), sau đó chờ trong vòng 1s (1000 ms) thì tắt Led, rồi lại chờ 1s thì bật led. Lặp lại liên tục như vậy.

Thực tế hàm delay chính là một vòng lặp có điều kiện. Khi MCU xử lý tới hàm này, chúng sẽ đi vào vòng lặp tới khi điều kiện được thỏa mãn thì sẽ thoát ra

- **Ưu điểm:** Dễ dàng thao tác
- **Nhược điểm:** Chương trình sẽ không thể xử lý các tác vụ khác khi đang delay, dẫn tới chậm, mất dữ liệu

Hàm millis()

Hàm `millis()` có nhiệm vụ trả về một số `unsigned long` – là thời gian (tính theo mili giây) kể từ lúc mạch Arduino bắt đầu chương trình của bạn. Nó sẽ tràn số và quay số 0 (sau đó tiếp tục tăng).

Khi dùng hàm `millis()` chúng ta sẽ lấy giá trị đo được hiện tại trừ đi giá trị đo được tại thời điểm trước đó => khoảng chênh lệch giữa 2 thời điểm. Sau đó so sánh với 1 số là thời gian delay mong muốn. Nếu giá trị đó bằng hoặc vượt quá sẽ xử lý các tác vụ bên trong.

Hàm `millis()` còn thường được gọi là ngắt mềm.

- **Ưu điểm:** Chương trình vẫn có thể xử lý được các tác vụ khác khi đang chờ
- **Nhược điểm:** Thiếu tính chính xác, nếu vi điều khiển phải xử lý một tác vụ dài hơn thời gian delay, chúng ta sẽ không thể chạy các tác vụ đúng thời điểm, mà phải chờ vi điều khiển xử lý hết rồi mới chạy tới

Ngắt Timer

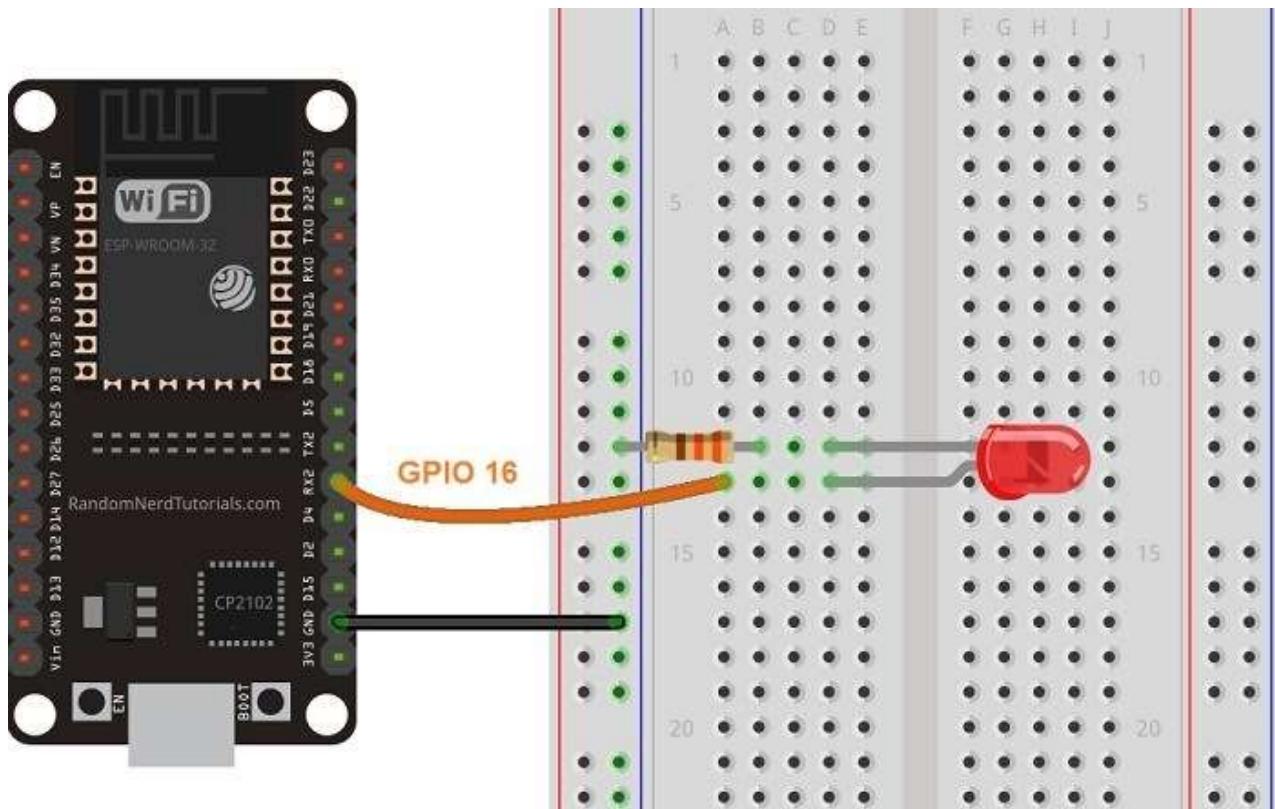
Vậy để khắc phục 2 nhược điểm của cách trên chúng ta có Ngắt Timer.

- **Ưu điểm:** Với ngắt Timer chúng ta có thể xử lý được các tác vụ trong chương trình chính mà vẫn có thể xử lý các tác vụ khác một cách lập tức khi tới thời gian.
- **Nhược điểm:** Nó có thể bị xung đột với các loại ngắt khác nếu không biết cách xử lý tốt, gây nên treo chương trình...

Lập trình Esp32 sử dụng Timer hẹn giờ bật tắt Led

Sơ đồ nguyên lý

Chúng ta sẽ sử dụng chân 16 để điều khiển Led. Sơ đồ nguyên lý như sau



Điều khiển Led bằng Timer

Bật tắt led sử dụng millis()

Cách xử dụng millis() như sau:

Full Code

```

02 #include <Arduino.h>
03
04 const int ledPin = 16;           // chân led
05
06 // biến trạng thái Led
07 int ledState = LOW;
08
09 // Giá trị lần cuối cùng được cập nhật
10 unsigned long previousMillis = 0;
11
12 const long interval = 1000; // giá trị delay (milliseconds)
13
14 void setup() {
15     // set led là output
16     pinMode(ledPin, OUTPUT);
17 }
18
19 void loop() {

```

```

20
21 // khởi tạo một biến lưu giá trị hiện tại của Timer
22 unsigned long currentMillis = millis();
23
24 //lấy giá trị hiện tại trừ đi giá trị trước đó và so sánh
25 if (currentMillis - previousMillis >= interval) {
26     // nếu giá trị đó bằng hoặc vượt qua
27     // lưu giá trị hiện tại thành giá trị lần cuối cập nhật
28     previousMillis = currentMillis;
29
30     // đảo giá trị led States
31     if (ledState == LOW) {
32         ledState = HIGH;
33     } else {
34         ledState = LOW;
35     }
36
37     // ghi giá trị đó lên Led
38     digitalWrite(ledPin, ledState);
39 }
40 //nếu giá trị chưa vượt qua thì không làm gì cả
41 }
```



Với kiểu làm này, nếu trong loop() mình để một lệnh delay(5000) nghĩa là chờ 5s thì bài toán nháy Led mỗi 1s sẽ bị sai, do vì điều khiển không thể quay lại kiểm tra millis() được

```

18 void loop() {
19
20     // khởi tạo một biến lưu giá trị hiện tại của Timer
21     unsigned long currentMillis = millis();
22
23     //lấy giá trị hiện tại trừ đi giá trị trước đó và so sánh với khoảng thời gian delay
24     if (currentMillis - previousMillis >= interval) {
25         // nếu giá trị đó bằng hoặc vượt qua
26         // lưu giá trị hiện tại thành giá trị lần cuối cập nhật
27         previousMillis = currentMillis;
28
29         // đảo giá trị led States
30         if (ledState == LOW) {
31             ledState = HIGH;
32         } else {
33             ledState = LOW;
34         }
35
36         // ghi giá trị đó lên Led
37         digitalWrite(ledPin, ledState);
38     }
39     //nếu giá trị chưa vượt qua thì không làm gì cả
40     delay(5000);
41 }
```

Vậy nên hãy lưu ý khi dùng hàm millis() nhé!

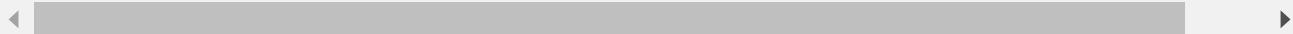
Bật tắt Led sử dụng Ngắt Timer

Cách xử lý ngắt Timer trong ESP32 như sau:

Full Code

```

02 #include <Arduino.h>
03
04 const int ledPin = 16;           // chân led
05 // biến trạng thái Led
06 int ledState = LOW;
07
08 hw_timer_t* timer = NULL; //khởi tạo timer
09 portMUX_TYPE timerMux = portMUX_INITIALIZER_UNLOCKED;
10
11 // hàm xử lý ngắt
12 void IRAM_ATTR onTimer() {
13     portENTER_CRITICAL_ISR(&timerMux); //vào chế độ tránh xung
14     digitalWrite(ledPin, ~ledState); // đảo giá trị Led
15     portEXIT_CRITICAL_ISR(&timerMux); // thoát
16 }
17
18 void setup() {
19     Serial.begin(115200);
20     //khởi tạo chân led là output
21     pinMode(ledPin,OUTPUT);
22
23     //khởi tạo timer với chu kỳ 1us vì thạch anh của ESP chạy
24     timer = timerBegin(0, 80, true);
25     //khởi tạo hàm xử lý ngắt ngắt cho Timer
26     timerAttachInterrupt(timer, &onTimer, true);
27     //khởi tạo thời gian ngắt cho timer là 1s (1000000 us)
28     timerAlarmWrite(timer, 1000000, true);
29     //bắt đầu chạy timer
30     timerAlarmEnable(timer);
31 }
32
33 void loop() {
34     //do nothing
35 }
```



Với chương trình này, chúng ta có thể xử lý bất kỳ tác vụ nào trong Loop() mà không ảnh hưởng tới việc nhấp nháy của Led.

Kết luận

Học cách sử dụng ESP32 Timer rất có ích khi xử lý nhiều tác vụ khác nhau, tuy rằng ESP32 chỉ có 1 Timer, vì vậy khả năng xử lý không được tốt cho lắm. Thế nhưng bù lại ESP32 có thể sử dụng RTOS (hệ điều hành thời gian thực mình sẽ có bài học về sau), điều này làm giảm đi đáng kể công việc của Timer.

Nếu thấy bài viết này có ích, hãy chia sẻ cho bạn bè. Và cùng gia nhập những người nghiên cứu lập trình tại đây nhé: [Hội anh em nghiên cứu lập trình](#)

5/5 - (3 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 5: Lập trình ESP32 ngắt ngoài EXTI](#)
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

2 THOUGHTS ON “BÀI 6: LẬP TRÌNH ESP32 TIMER MILLIS VÀ NGẮT TIMER”



Nguyễn Thị Huyền Thương says:

Em chào anh ạ.

Em hiện đang học lập trình esp32 để làm một cái định vị gps, dữ liệu sẽ được gửi qua sim800L đến điện thoại. Em muốn hỏi là với màn hình oled và esp32 mình có thể hiển thị thời gian thực: giờ, phút giây trên oled mà không sử dụng ngoại vi nào được không ạ. Nếu được thì anh có thể nói chi tiết một chút cho em với ạ. Em mới bắt đầu học nên hiểu biết về esp32 đang còn hạn chế ạ

Em cảm ơn anh ạ.

17/07/2022 AT 7:40 SÁNG

TRẢ LỜI



Khuê Nguyễn says:

Em có thể dùng timer để sinh ra ngắt 1s, còn thời gian có thể kết nối với wifi để get thời gian thực về thông qua API. khoảng 1p hoặc 10p cập nhật 1 lần

28/07/2022 AT 11:40 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 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



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



Khuê Nguyễn Creator



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



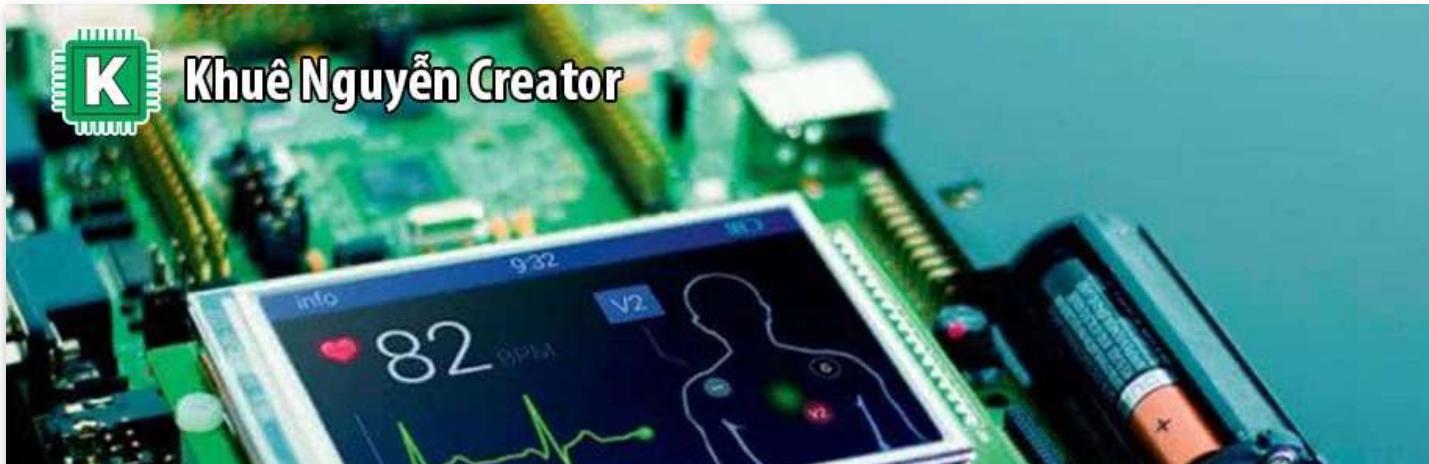
Lập trình STM32 HID Host giao tiếp với chuột và bàn phím

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](#)

Khuê Nguyễn Creator





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



Lập trình STM32F407 SDIO đọc dữ liệu thẻ nhớ

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

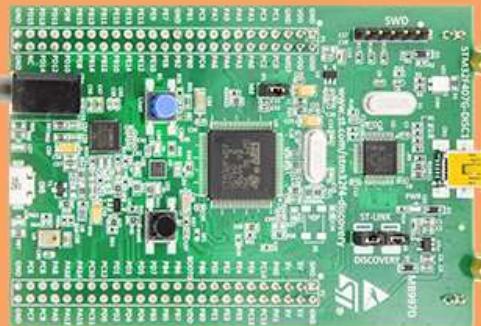
Trong bài này chúng ta cùng học cách lập trình STM32 SDIO, một chuẩn giao...

[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

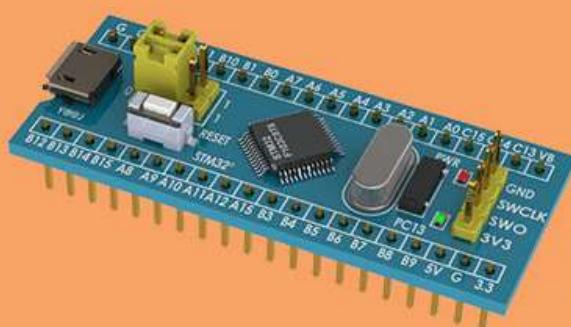
Trong bài này chúng ta sẽ cùng nhau tìm hiểu STM32 DAC với KIT STM32F407VE...

[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

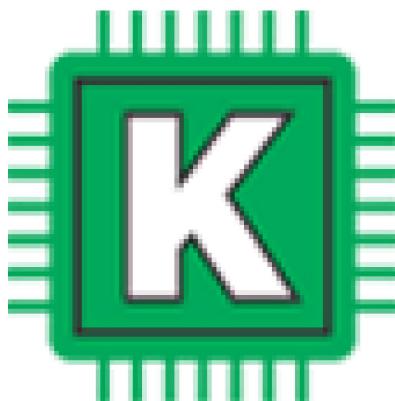


Cài đặt SDC Complier và Code:Blocks IDE

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: Nghiên Lập Trình

<https://khuenguyencreator.com/lap-trinh-esp32-timer-millis-va-ngat-timer/>

16/17

Fanpage: Khuê Nguyên Creator

My Shop

Thông Tin

Tác Giả

Chính Sách Bảo Mật



Copyright 2022 © Khuê Nguyễn