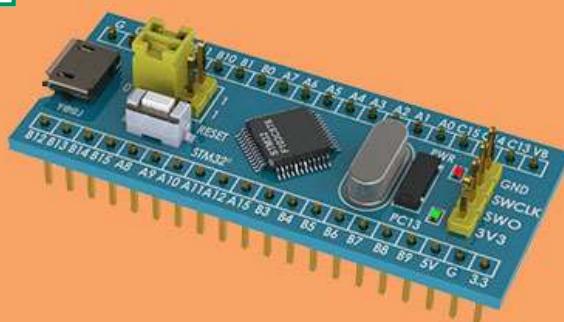
**LẬP TRÌNH STM32**

# Bài 5: Lập trình Timer STM32 chớp tắt Led dùng ngắn

POSTED ON 09/07/2020 BY KHUÊ NGUYỄN

09  
Th7**Khuê Nguyễn Creator**

## Bài 5: Timer là gì, điều khiển Timer chế độ Time base tạo trễ

Timer là một ngoại vi được ứng dụng rất nhiều trong lập trình nhúng, trong STM32 có khá nhiều Timer đa số là 16bit và chúng có thể làm việc một cách độc lập với nhau

Trong bài này ta sẽ tìm hiểu Timer STM32 được sử dụng như thế nào, lập trình Timer STM32 điều khiển chớp tắt led không dùng delay

## Bài 5 trong Serie **Lập trình STM32 từ A tới Z**

### Mục Lục

- 1. Timer là gì?
- 2. Các chức năng chính của Timer STM32
- 3. Cấu hình Timer STM32 chế độ Time Base
- 4. Lập trình Timer STM32 chớp tắt led
  - 4.1. Related posts:



## Timer là gì?

Timer dịch đơn giản là người ghi giờ, bộ hẹn giờ.

Ví dụ khi bạn cần thức dậy sau 8 tiếng ngủ bạn sử dụng một bộ hẹn giờ, đếm ngược hoặc đếm xuôi đủ 8 tiếng, chuông sẽ kêu và bạn thức dậy.

Hoặc khi chạy 100m người trọng tài chính là Timer để đo thời gian bạn chạy hết 100m đó.

Trong lập trình Timer là một khối độc lập, có tác dụng tạo ra các sự kiện hoặc ngắt để kích hoạt các ngoại vi khác hoạt động, hoặc đo thời gian hoạt động của 1 giá trị đầu vào nào đó.

Trong STM32F103C8 có 4 bộ Timer trong đó:

- Timer 1: Là bộ Advanced – control Timer hay là bộ Timer điều khiển nâng cao, có nhiều chức năng nhất
- 3 bộ Timer chung là Timer 2, 3, 4 Có chức năng tương tự nhau và độc lập với nhau

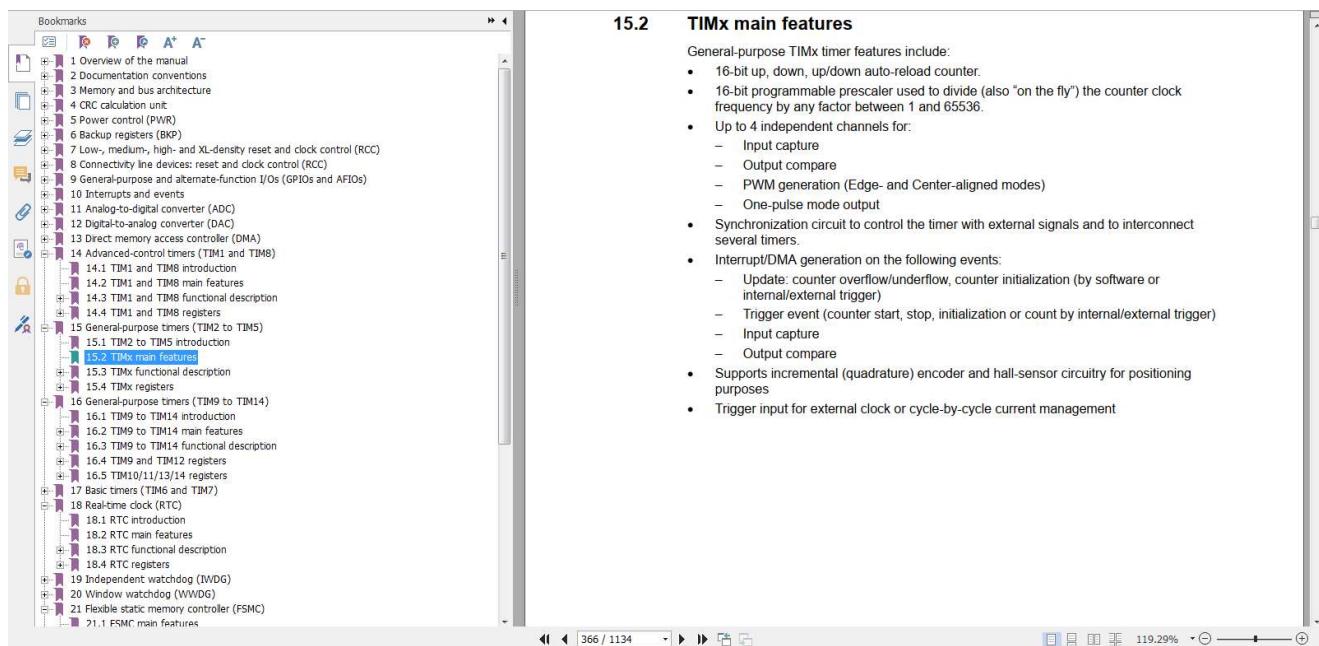
Trong bài này chúng ta sẽ làm việc với Timer 2, các chức năng đều tương tự với các Timer khác

## Các chức năng chính của Timer STM32

Các chức năng chính của Timer STM32 bao gồm:

- Thanh ghi 16bit đếm lên, xuống, lên/xuống tự nạp lại
- 16 bit bộ chia tần số để chia tần số từ APB (giá trị dao động từ 1 – 65536)
- 4 Kênh độc lập mỗi Timer cho các chức năng:
  - + Input Capture
  - + Output Compare
  - + One Pulse
- Đồng bộ hóa với các mạch tạo tín hiệu bên ngoài để kết hợp nhiều bộ Timer với nhau
- Ngắt/DMA được sinh ra khi có các sự kiện:
  - + Cập nhật: tràn Counter, khởi tạo Counter (bởi phần mềm hoặc kích hoạt internal/external trigger)
  - + Sự kiện kích hoạt (Bắt đầu đếm, dừng đếm, khởi tạo bộ đếm hoặc đếm bởi internal/external trigger)
  - + Input Capture: Bắt xung đầu vào
  - + Output Compare: So sánh xung đầu ra
- Hỗ trợ điều khiển Encoder và Hall-sensor
- Đầu vào kích hoạt cho đồng hồ bên ngoài hoặc quản lý theo chu kỳ

Chi tiết các bạn tham khảo mục 15.2 trong Reference Manual



## Bài 5: Lập trình Timer STM32 chớp tắt Led dùng ngắt 50

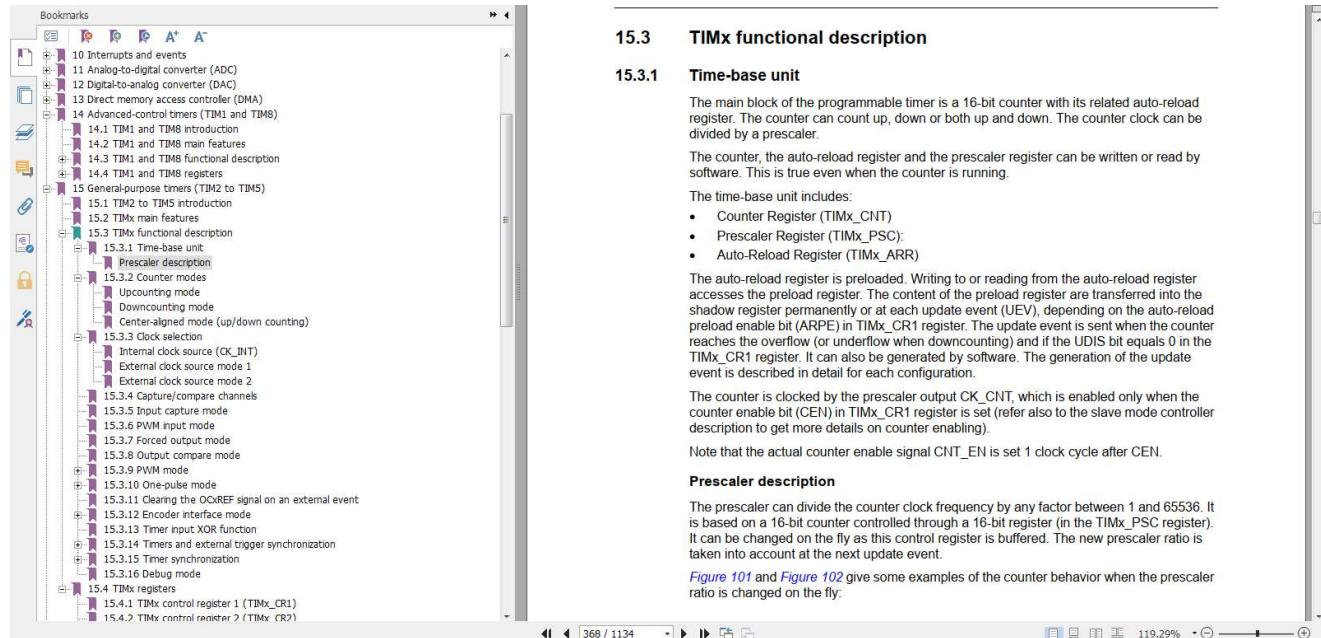
Trong bài này chúng ta sẽ cấu hình Timer 2 chế độ Time-base unit.

Mô tả chức năng này như sau:

- Chế độ Timer-Base unit là chế độ chính của Timer bao gồm các chế độ đếm: lên, xuống hoặc cả lên lên và xuống
- Xung Clock được chia bởi bộ chia tần Prescaler để lấy thời gian thích hợp đếm 1 lần
- Các thanh ghi quản lý bao gồm:
  - + Counter Register (TIMx\_CNT): lưu giá trị đếm
  - + Prescaler Register (TIMx\_PSC): lưu giá trị chia tần số cơ sở cấp cho Timer để tạo ra tần số thích hợp
  - + Auto-Reload Register(TIMx\_ARR): lưu giá trị đích đếm lên hoặc đếm xuống
- Thanh ghi Auto-Reload sẽ được nạp trước khi Timer hoạt động, và có thể nạp trong khi Timer hoạt động, trước khi một sự kiện cập nhật xảy ra (UEV).

– Chế độ đếm sẽ được hoạt động khi Bit CEN của thanh ghi TIMx\_CR1 được bật

### Chi tiết các bạn tham khảo mục 15.3.1



## Bài 5: Lập trình Timer STM32 chớp tắt Led dùng ngắn 51

Nhìn vào mô tả này chúng ta có thể viết ra các bước để Timer hoạt động như sau:

1. Khởi tạo bộ Timer 1 với xung Clock thích hợp(Tất cả các ngoại vi đều phải có bước này)
2. Ghi giá trị cho bộ chia tần Prescaler
3. Ghi giá trị cho thanh ghi Auto-reload
4. Bật Timer cho hoạt động
5. Xử lý các sự kiện xảy ra như tràn ....

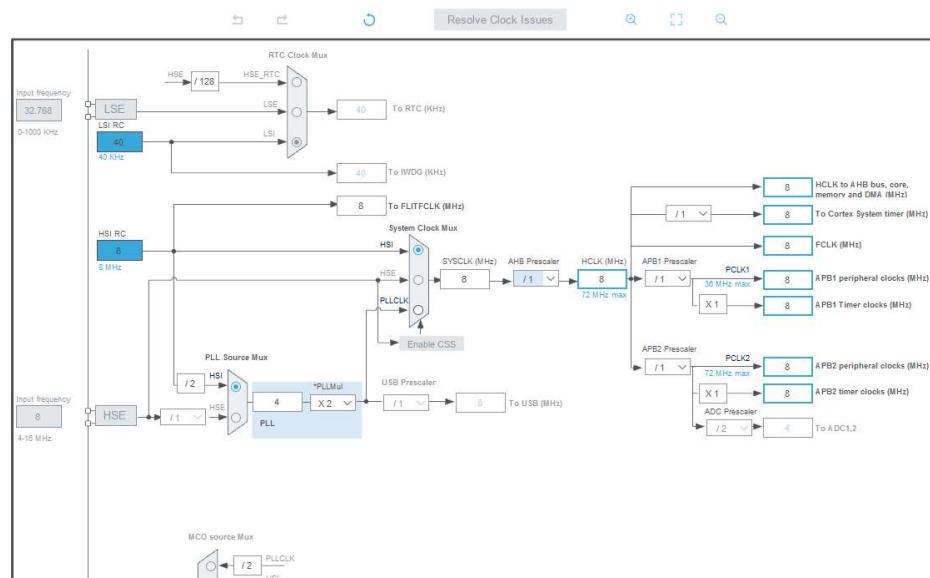
Các bước cơ bản là vậy, sau đây mình sẽ hướng dẫn các bạn cấu hình trên CubeMX

## Cấu hình Timer STM32 chế độ Time Base

Đề bài của chúng ta như sau: Sử dụng TimerSTM32 cụ thể là Timer 2 nhấp nháy led mỗi 500ms một lần.

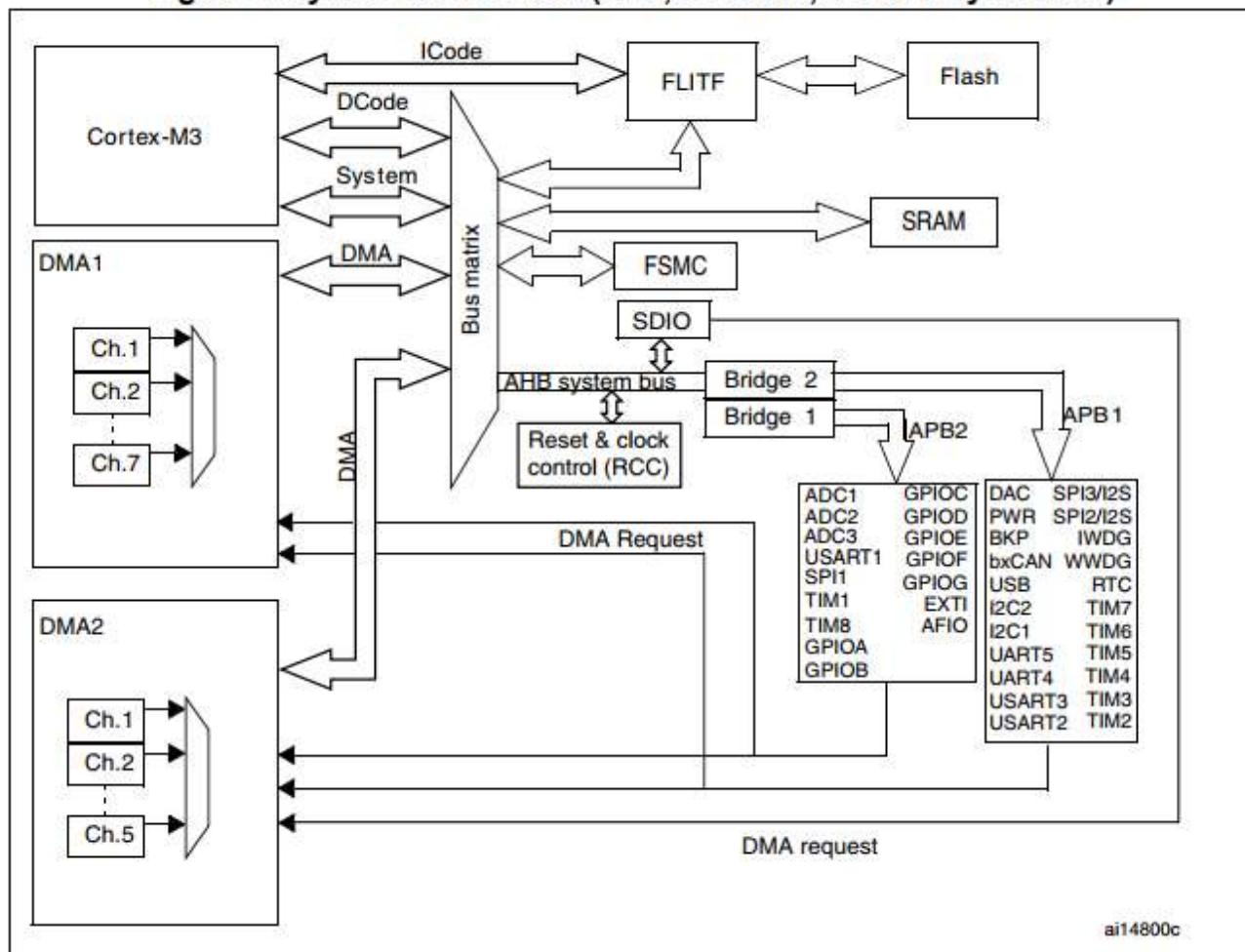
Các bạn mở phần mềm, chọn chip STM32F103C8, cấu hình SYS debug: Serial Wire.

Chuyển qua Tab Clock Configuration , chúng ta chọn Thạch Anh nội 8Mhz. Vì xung Clock của Timer 2 sẽ được cấp bởi bộ APB1 thế nên xung Interval Clock có tần số là 8Mhz



Bài 5: Lập trình Timer STM32 chớp tắt Led dùng ngắn 52

Chi tiết về cách cấu hình Clock các bạn xem Bảng 1 Mục 3.1

**Figure 1. System architecture (low-, medium-, XL-density devices)**

## Bài 5: Lập trình Timer STM32 chớp tắt Led dùng ngắt 53

Trong Tab Timer các bạn chọn Timer 2. Clock Source chọn Internal Clock.

Trong bảng Parameter Settings:

Chọn Prescale là 8000 sẽ đếm mỗi 1ms vì  $8\text{MHz}/8000 = 1\text{kHz} \Rightarrow T = 1\text{ms}$

Counter Mode : Up down đều được

Counter Period: 499 , đếm từ 0 đến 499 là 500 lần 1ms ta sẽ được 500ms

Auto-preload: Enable

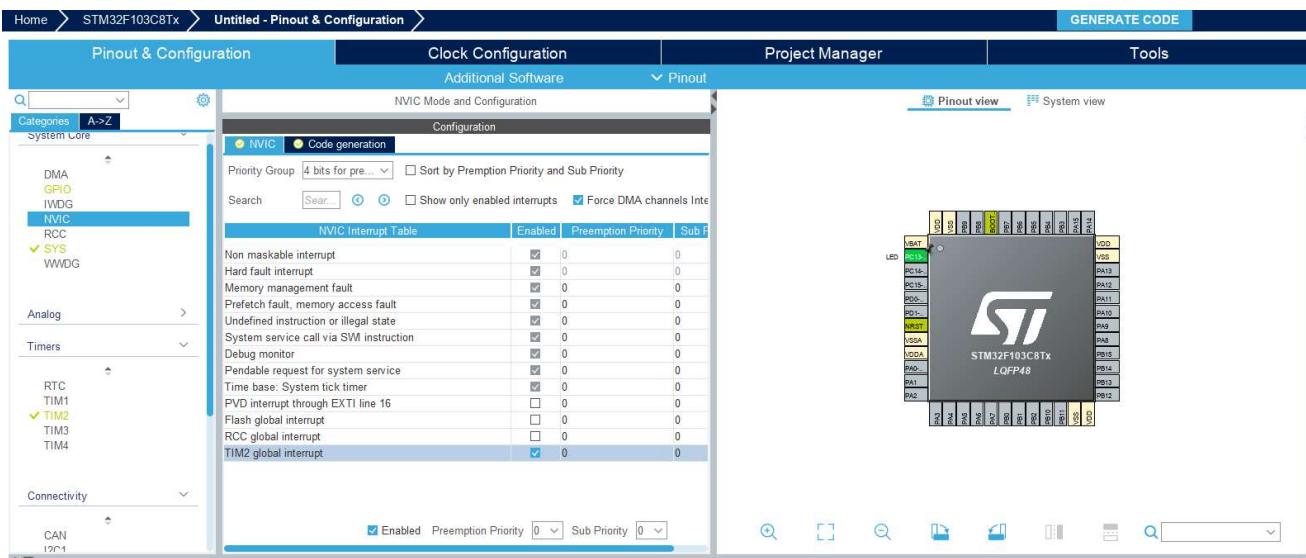
Các thông số còn lại giữ nguyên

## Bài 5: Lập trình Timer STM32 chớp tắt Led dùng ngắt 54

Chọn PC13 là LED output

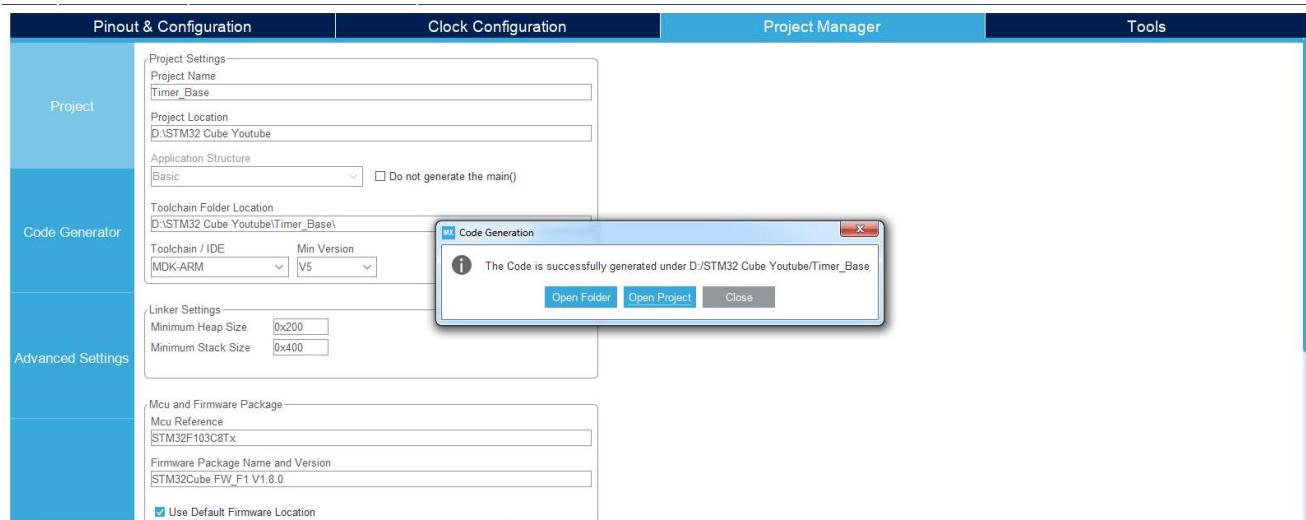
## Bài 5: Lập trình Timer STM32 chớp tắt Led dùng ngắt 55

Trong NVIC tick vào Tim2 global Interupts để bật ngắt cho Timer 2



## Bài 5: Lập trình Timer STM32 chớp tắt Led dùng ngắt 56

Sau đó đặt tên Project – chọn tool MDK-ARM V5 và ấn Gen Code



## Bài 5: Lập trình Timer STM32 chớp tắt Led dùng ngắt 57

# Lập trình Timer STM32 chớp tắt led

Mở Project – Nhấn Build (F7)

Trong phần stm32f1xx\_it.c đã có hàm thực thi ngắt Timer2, các bạn nhấn chuột phải chọn Go to Define hàm HAL\_TIM\_IRQHandler(&htim2);

```

201  * @brief This function handles TIM2 global interrupt.
202  */
203 void TIM2_IRQHandler(void)
204 {
205     /* USER CODE BEGIN TIM2_IRQHandler_0 */
206
207     /* USER CODE END TIM2_IRQHandler_0 */
208     HAL_TIM_IRQHandler();
209     /* USER CODE */
210
211     /* USER CODE */
212 }
213
214 /* USER CODE BEGIN */
215
216 /* USER CODE END */
217 /*****END OF FILE*****/
218

```

0 Error(s), 0 Warning(s)

## Bài 5: Lập trình Timer STM32 chớp tắt Led dùng ngắt 58

Tại đây các bạn nhấn CtrF Tìm từ update nhấn Find Next sẽ tìm thấy, phần xử lý ngắt Update Event.

Chọn File HAL\_TIM\_PeriodElapsedCallback(htim); nhận chuột phải Go to Define để tìm thấy nơi viết hàm đó.

Đây là hàm gọi về khi có sự kiện ngắt Update xảy ra

```

3283 #else
3284     HAL_TIM_OC_DelayElapsed();
3285     HAL_TIM_PWM_PulseFinished();
3286 #endif /* USE_HAL_TIM_REGISTER_CALLBACKS */
3287 }
3288 htim->Channel = HAL_TIM_CHANNEL_TIM2;
3289 }
3290 /* TIM Update event */
3291 if (__HAL_TIM_GET_FLAG(htim))
3292 {
3293     if (__HAL_TIM_GET_IT_SOURCE(htim))
3294     {
3295         __HAL_TIM_CLEAR_IT(htim);
3296 #if (USE_HAL_TIM_REGISTER_CALLBACKS)
3297         if (htim->PeriodElapsedCall)
3298             htim->PeriodElapsedCall(htim);
3299 #else
3300         HAL_TIM_PeriodElapsedCallback(htim);
3301 #endif /* USE_HAL_TIM_REGISTER_CALLBACKS */
3302     }
3303 /* TIM Break input event */
3304 if (__HAL_TIM_GET_FLAG(htim, TIM_FLAG_BREAK) != RESET)
3305 {
3306     if (__HAL_TIM_GET_IT_SOURCE(htim, TIM_IT_BREAK) != RESET)
3307     {

```

## Bài 5: Lập trình Timer STM32 chớp tắt Led dùng ngắt 59

Coppy hàm đó dán vào khu tiền xử lý trên hàm main() trong main.c nhớ bỏ \_\_weak nhé.

Sau đó trong hàm này chúng ta viết thêm hàm đảo trạng thái led như sau:

```

58  /* Private user code ----- */
59  /* USER CODE BEGIN 0 */
60  void HAL_TIM_PeriodElapsedCallback(TIM_HandleTypeDef *htim)
61  {
62      /* Prevent unused argument(s) compilation warning */
63      UNUSED(htim);
64      HAL_GPIO_TogglePin(LED_GPIO_Port,LED_Pin);
65      /* NOTE : This function should not be modified, when the callback is needed,
66           the HAL_TIM_PeriodElapsedCallback could be implemented in the user file
67      */
68  }
69  /* USER CODE END 0 */
70
71 /**
72  * @brief  The application entry point.
73  * @retval int
74  */
75 int main(void)
76 {
77     /* USER CODE BEGIN 1 */

```

Bài 5: Lập trình Timer STM32 chớp tắt Led dùng ngắt 60

Trước while (1) ta khởi chạy Timer với ngắt bằng câu lệnh sau

```

/* USER CODE BEGIN SysInit */
/* USER CODE END SysInit */

/* Initialize all configured peripherals */
MX_GPIO_Init();
MX_TIM2_Init();
/* USER CODE BEGIN 2 */
HAL_TIM_Base_Start_IT(&htim2);
/* USER CODE END 2 */

/* Infinite loop */
/* USER CODE BEGIN WHILE */
while (1)
{
    /* USER CODE END WHILE */
    /* USER CODE BEGIN 3 */
}

```

Bài 5: Lập trình Timer STM32 chớp tắt Led dùng ngắt 61

Sau đó Build và Nạp chương trình và xem kết quả

Kết

Timer là một ngoại vi không thể thiếu trong việc lập trình nhúng. Timer STM32 ngoài được sử dụng trong việc hẹn giờ, chúng còn được sử dụng đo đếm thời gian của các xung, tạo ra các xung có tần số khác nhau mà bài sau chúng ta sẽ đề cập.

5/5 - (4 bình chọn)

## Related Posts:

1. [Bài 12: Lập trình STM32 với giao thức SPI](#)
2. [Bài 11: Lập trình STM32 với Giao thức UART](#)
3. [Bài 6: STM32 Timer chế độ Input Capture và Output Compare](#)
4. [Bài 4: Lập trình Ngắt Ngoài STM32 EXTI](#)
5. [Bài 3: Lập trình STM32 GPIO điều khiển Led và nút nhấn](#)
6. [Bài 2: Tổng quan về KIT STM32F103C8T6 Blue Pill](#)



**KHUÊ NGUYỄN**

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

## 9 THOUGHTS ON “BÀI 5: LẬP TRÌNH TIMER STM32 CHỚP TẮT LED DÙNG NGẮT”



*Anh Tuấn says:*

Sao em làm đến bước go to define HAL\_TIM\_IRQHandler(&htim2); nó lại báo lỗi cái này chưa đc define vậy à 😞 em ktra các bước trên làm đúng hết r mà 😞



21/04/2021 AT 9:08 CHIỀU

TRẢ LỜI



*Anh Tuấn says:*

À thôi anh ơi :V em biết sao bị vậy e 😊

21/04/2021 AT 9:30 CHIỀU

TRẢ LỜI

**Khuê Nguyễn** says:

Nhấn F7 để Build trc nhé

01/05/2021 AT 9:47 CHIỀU

TRẢ LỜI

**Nguyễn Hùng Thịnh** says:

anh ơi, như vậy là bị sao, em cx bị

21/06/2021 AT 4:00 CHIỀU

TRẢ LỜI

**Khuê Nguyễn** says:

bị sao em?

21/06/2021 AT 4:31 CHIỀU

TRẢ LỜI

**Lê Văn Hiếu** says:

Hay, cảm ơn ad.

03/07/2021 AT 10:01 SÁNG

TRẢ LỜI

**nguyễn Hoàng Quý** says:A ơi sao e làm chạy được đèn led PC13 sáng nhưng nó lại không nhấp nháy  
q

24/11/2021 AT 3:54 CHIỀU

TRẢ LỜI

**Trần Văn Sỹ** says:Cho em hỏi lỗi sau khi sinh code thì k hiện open project mà chỉ có open folder. Sau đó thì nó hiện lỗi code đã được tạo nhưng mdk-arm có vấn đề q.  
Ai biết lỗi này k q! Em cảm ơn

11/06/2022 AT 3:57 CHIỀU

TRẢ LỜI

**Khuê Nguyễn** says:

Xem lại sinh code đúng version chưa nhé

## 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 Nguyen 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](#)



## 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



### 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



STM32  
Cube

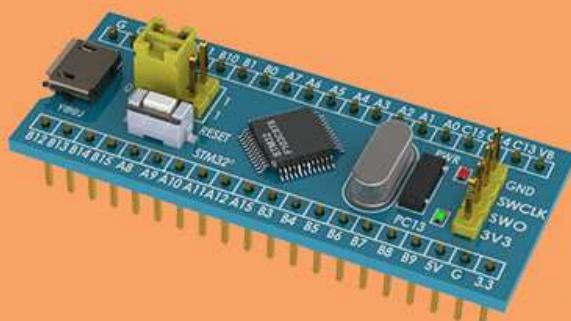
# Lập trình STM32F407 DAC chuyển đổi số sang tương tự

**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 STUDIO và Code::BLOCKS lập trình Nuvoton

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

**ĐỌC THÊM**



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.

## Liên Kết

Nhóm: Nghiên Lập Trình

My Shop

## Thông Tin

Tác Giả

Chính Sách Bảo Mật



Copyright 2022 © Khuê Nguyễn