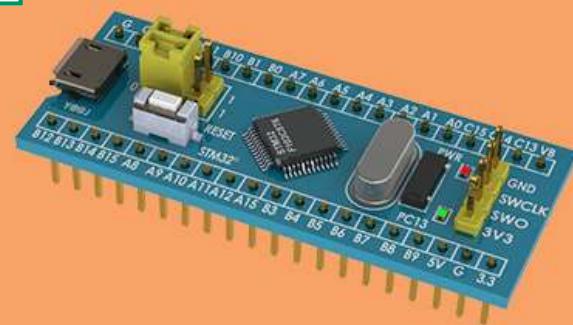


**LẬP TRÌNH STM32**

# Bản đồ bộ nhớ (Memory map) vi điều khiển STM32F103

POSTED ON 23/07/2021 BY KHUÊ NGUYỄN

23  
Th7**Khuê Nguyễn Creator**

## Cấu trúc và bộ nhớ vi điều khiển STM32f103

Khi học bất kì con vi điều khiển nào đầu tiên chúng ta cần tìm hiểu cấu trúc và bản đồ bộ nhớ (memory maps) của vi điều khiển đó. Điều này sẽ giúp chúng ta nắm rõ được cách thức nó hoạt động, chạy chương trình, chạy code như thế nào.

Bài viết nằm trong Serie Học STM32 từ A tới Z

## Mục Lục

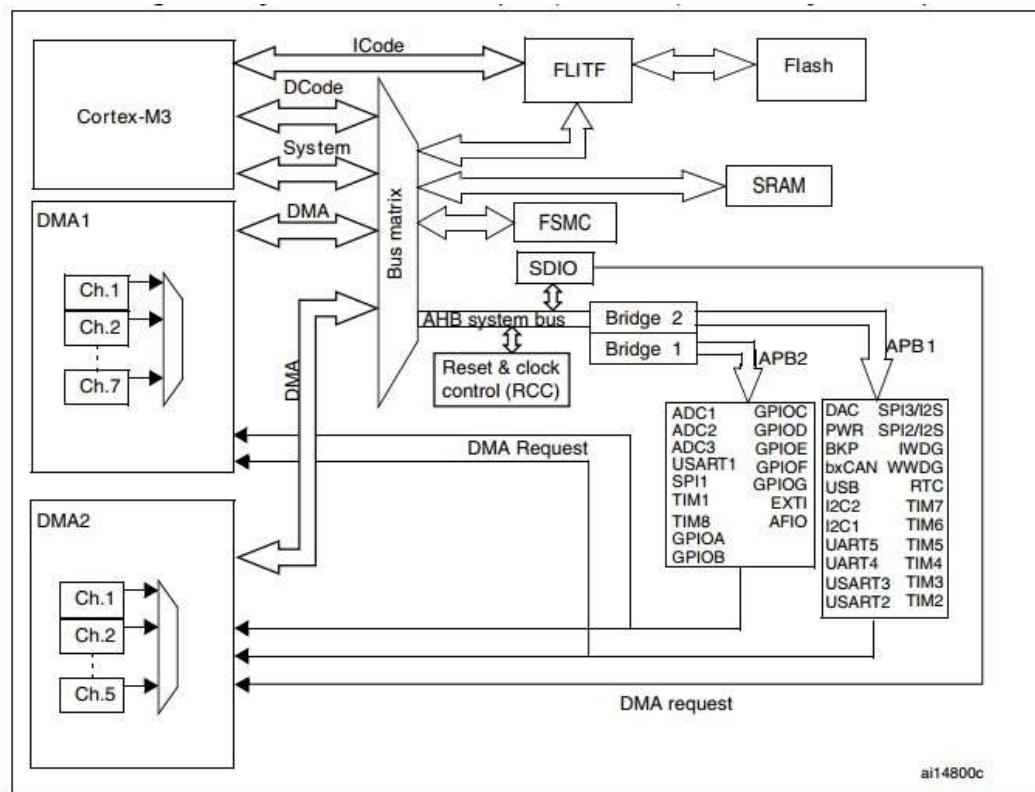
1. Kiến trúc vi điều khiển STM32F103
  - 1.1. Sơ đồ tổng quan
  - 1.2. Dcode Bus
  - 1.3. System Bus
  - 1.4. DMA bus
  - 1.5. BusMatrix
  - 1.6. AHB/APB bridges (APB)
2. Core registers ( Thanh ghi lỗi)
3. Bản đồ bộ nhớ STM32f103 (Memory Maps)
  - 3.1. Code Area
  - 3.2. SRAM Area
    - 3.2.1. Sram là gì?
    - 3.2.2. Sram trong STM32 tổ chức ra sao?
  - 3.3. Peripheral Area
  - 3.4. External RAM Area
4. Kết
  - 4.1. Related posts:



# Kiến trúc vi điều khiển STM32F103

## Sơ đồ tổng quan

- 4 masters:
  - Cortex®-M3 core DCode bus (D-bus) và System bus (S-bus)
  - GP-DMA1 & 2 (general-purpose DMA)
- 4 slaves:
  - Internal SRAM: Ram nội
  - Internal Flash memory: Bộ nhớ Flash nội
  - FSMC: Giao thức kết nối bộ nhớ ngoài
  - AHB to APBx (APB1 or APB2), kết nối tới các ngoại vi APB



Kiến trúc vi điều khiển STM32f103

## Dcode Bus

Kết nối lõi Cortex M3 với bộ nhớ Flash và các chuẩn kết nối dữ liệu bộ nhớ.

## System Bus

Bus này kết nối bus hệ thống của lõi Cortex®-M3 (bus thiết bị ngoại vi) với BusMatrix quản lý data giữ Core và DMA

## DMA bus

Bus này kết nối giao diện chủ AHB của DMA với BusMatrix quản lý quyền truy cập của CPU DCode và DMA tới SRAM, bộ nhớ Flash và thiết bị ngoại vi.

## BusMatrix

BusMatrix quản lý việc phân xử truy cập giữa bus hệ thống lõi và bus chủ DMA. Nó sử dụng thuật toán Round Robin. Trong các thiết bị kết nối, BusMatrix bao

gồm năm bus master (CPU DCode, bus hệ thống, Ethernet DMA, DMA1 và bus DMA2) và ba slave (FLITF, SRAM và AHB2APB).

Trong các thiết bị khác, BusMatrix bao gồm master (CPU DCode, bus hệ thống, bus DMA1 và bus DMA2) và bốn slave (\FLITF, SRAM, FSMC và AHB2APB). Các thiết bị ngoại vi AHB được kết nối trên bus hệ thống thông qua BusMatrix để cho phép truy cập DMA.

## AHB/APB bridges (APB)

Hai cầu AHB / APB cung cấp kết nối đồng bộ đầy đủ giữa AHB và 2 bus APB. APB1 được giới hạn ở 36 MHz, APB2 hoạt động ở tốc độ tối đa (lên đến 72 MHz tùy thuộc vào thiết bị).

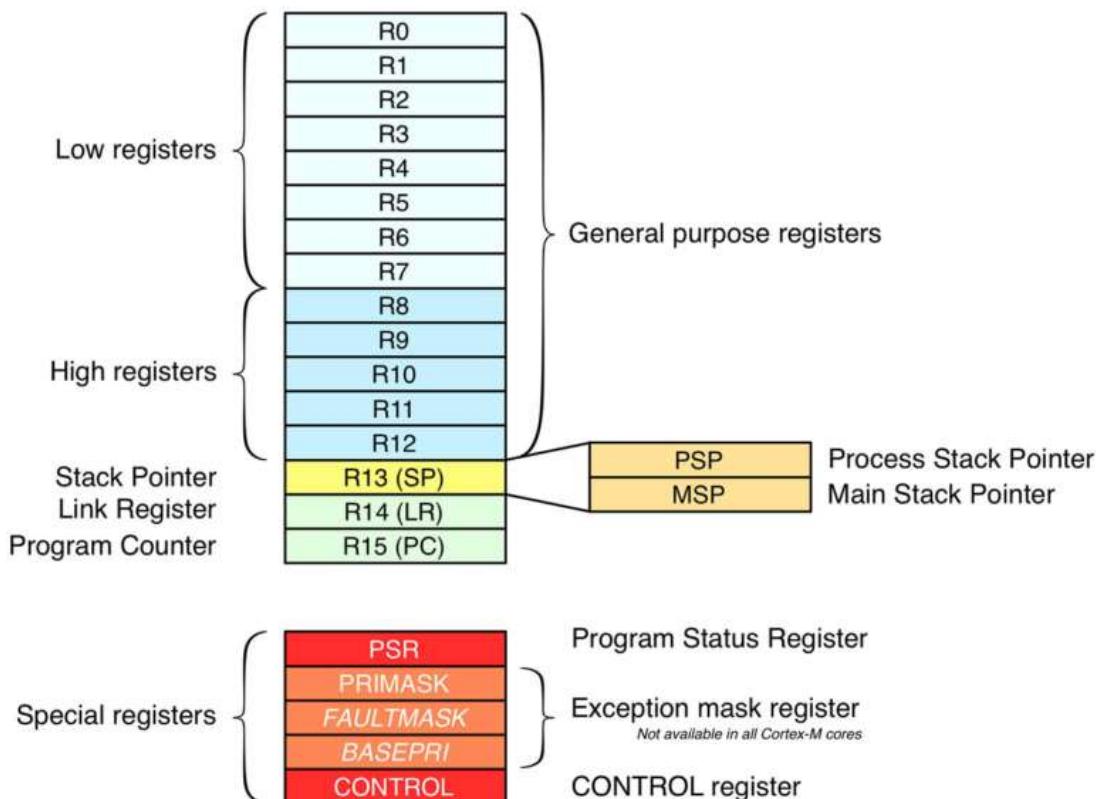
Sau mỗi lần reset thiết bị, tất cả các đồng hồ ngoại vi đều bị tắt (ngoại trừ SRAM và FLITF). Trước khi sử dụng thiết bị ngoại vi, bạn phải bật clock của nó trong thanh ghi RCC\_AHBENR, RCC\_APB2ENR hoặc RCC\_APB1ENR

## Core registers ( Thanh ghi lõi)

R0-R12 là các thanh ghi có mục đích chung và có thể được sử dụng làm toán hạng cho các lệnh ARM.

Một số thanh ghi có mục đích chung, có thể được trình biên dịch sử dụng như các thanh ghi có chức năng đặc biệt.

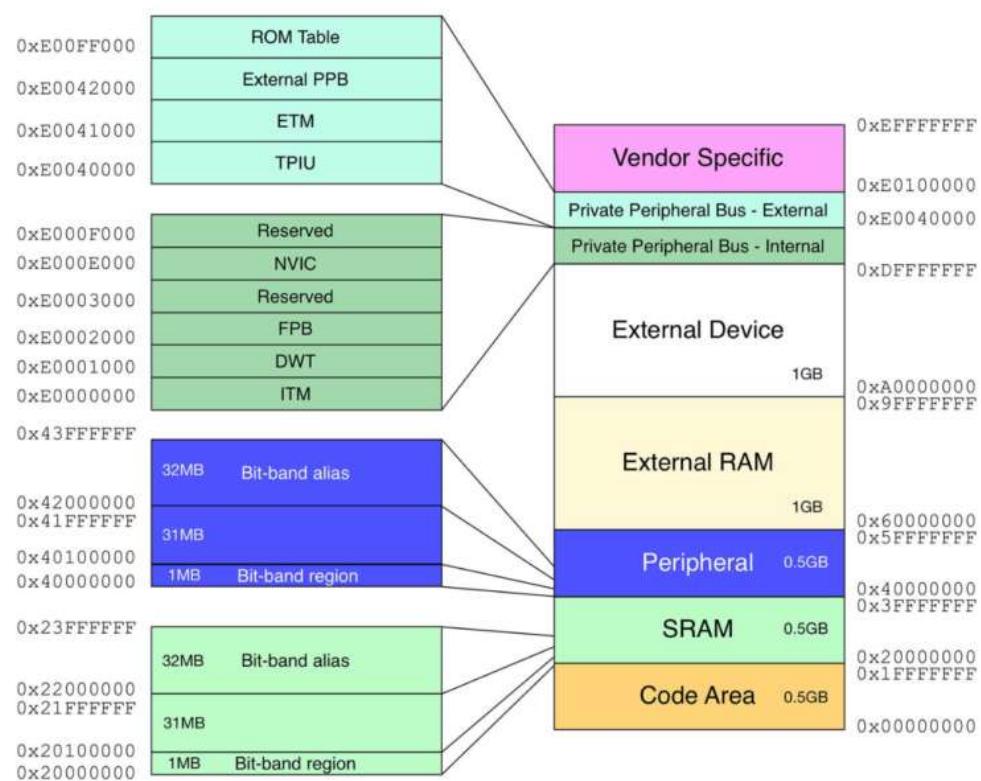
Thanh ghi R13 là thanh ghi Stack Pointer (SP), có cấu trúc dạng ngăn xếp, nghĩa là vào đầu tiên thì ra sau cùng First in Last Out

*Cấu trúc thanh ghi STM32*

## Bản đồ bộ nhớ STM32f103 (Memory Maps)

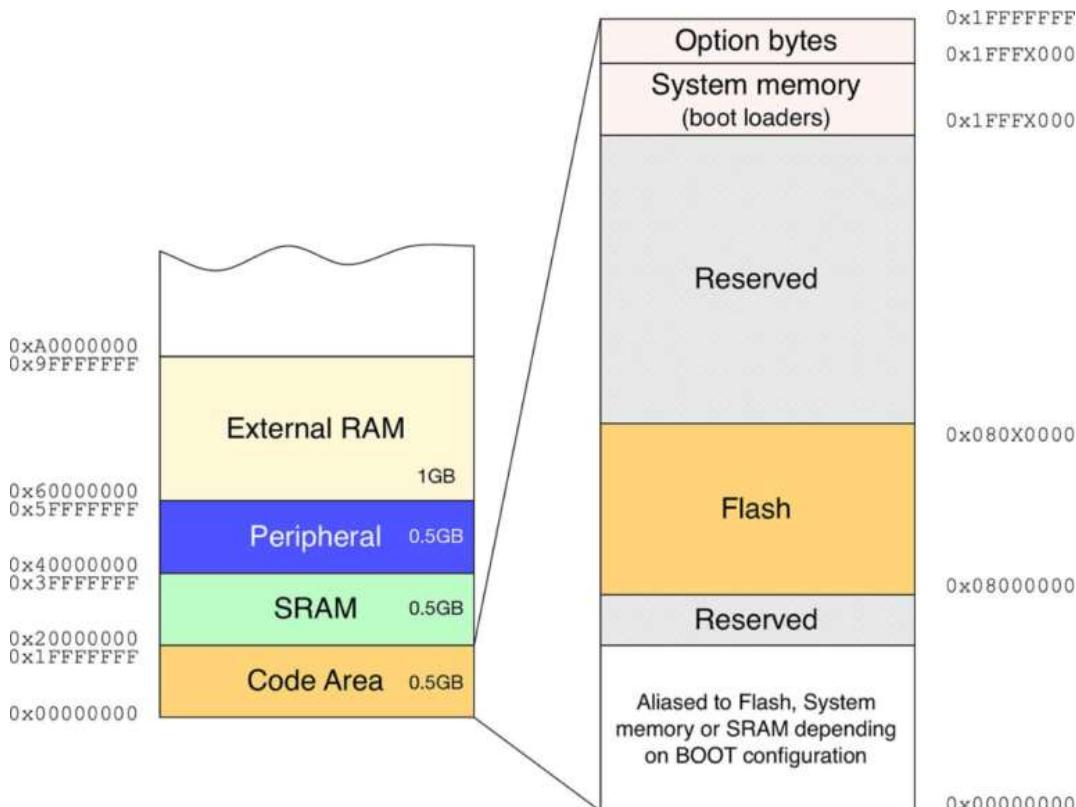
ARM xác định một không gian địa chỉ bộ nhớ được tiêu chuẩn hóa chung cho tất cả các lõi Cortex-M, điều này thực thi khả năng di động của code giữa các nhà sản xuất khác nhau.

Không gian bộ nhớ (memory) được thiết kế chia thành một số vùng khác nhau. Code, SRAM, Peripherals, External RAM, External Device, Pravite peripheral bus internal and external, Vendor Specific như hình sau:



Bản đồ bộ nhớ (Memory map stm32)

## Code Area



Bản đồ vùng nhớ chương trình STM32 (Code Area)

512MB đầu tiên được dành riêng cho vùng code. Tất cả các bộ xử lý Cortex-M ánh xạ vùng mã bắt đầu từ địa chỉ 0x0000 0000. Khu vực này cũng bao gồm con trỏ đến đầu ngăn xếp (thường được đặt trong SRAM) và bảng vectơ.

Trên thực tế, đối với tất cả các thiết bị STM32, một vùng bắt đầu từ địa chỉ 0x0800 0000 được liên kết với bộ nhớ FLASH MCU bên trong và nó là vùng chứa mã chương trình. Nhờ cấu hình khởi động cụ thể, khu vực này cũng được đặt bí danh từ địa chỉ 0x0000 0000.

Điều này có nghĩa là hoàn toàn có thể tham chiếu đến nội dung của bộ nhớ FLASH bắt đầu từ địa chỉ 0x0800 0000 và địa chỉ 0x0000 0000

Hai phần cuối dành riêng cho bộ nhớ Hệ thống (System Memory) và Byte tùy chọn ( Options Byte)

## **System Memory – Bộ nhớ hệ thống**

Bộ nhớ hệ thống là vùng ROM dành riêng cho bộ nạp khởi động (Boot Loader). Mỗi họ STM32 cung cấp một số Boot Loader khác nhau được lập trình sẵn vào chip trong quá trình sản xuất. Những Boot Loader này có thể được sử dụng để tải code từ một số thiết bị ngoại vi, bao gồm USART, USB và CAN Bus.

## **Option Bytes – Vùng tùy chọn byte**

Vùng byte tùy chọn chứa một loạt cờ bit có thể được sử dụng để định cấu hình một số khía cạnh của MCU (như bảo vệ đọc ghi FLASH, v.v.)

## **SRAM Area**

### **Sram là gì?**

SRAM là viết tắt của Static random-access memory (Bộ nhớ truy cập ngẫu nhiên tĩnh hay RAM tĩnh). SRAM lưu giữ các bit dữ liệu trong bộ nhớ miễn là nguồn điện được cung cấp đầy đủ. Không giống như DRAM, lưu bit dữ liệu trong các pin chứa tụ điện và bóng bán dẫn, SRAM không cần phải làm tươi theo định kỳ.

### **Sram trong STM32 tổ chức ra sao?**

SRAM bắt đầu từ địa chỉ 0x2000 0000 và có thể kéo dài đến 0x3FFF FFFF.

SRAM được sử dụng để truy xuất data và bộ nhớ stack.

Tuy nhiên, địa chỉ kết thúc thực tế phụ thuộc vào số lượng SRAM nội bộ khả dụng. Ví dụ: giả sử MCU STM32F103RB với 20KB SRAM, chúng ta có địa chỉ cuối cùng là 0x200013FF. Cố gắng truy cập vào một vị trí bên ngoài khu vực này sẽ gây ra lỗi Bus Fault.

Trong vùng nhớ này, 1MB đầu tiên được chia thành vùng Bit-band. Mỗi bit trong vùng Bit-band được đánh địa chỉ độc lập. Bạn cũng có thể thực thi code từ vùng nhớ SRAM, Vi xử lý không giới hạn việc thực thi chương trình từ đây.

## Peripheral Area

Mọi thiết bị ngoại vi được cung cấp bởi MCU (bộ định thời, I2C và SPI, USART, v.v.) đều được đặt bí danh cho vùng này. Chi tiết về các thanh ghi của ngoại vi sẽ phụ thuộc vào từng dòng vi điều khiển khác nhau. Các bạn nên đọc tài liệu của từng loại để hiểu rõ hơn

## External RAM Area

Nếu như bộ nhớ RAM được tích hợp sẵn trên vi điều khiển không đủ và bạn muốn kết nối với 1 bộ nhớ RAM ngoài thì bạn có thể khai thác được chức năng của vùng nhớ này.

Các thiết bị Cortex-M có thể thực thi mã và tải / lưu trữ dữ liệu từ các bộ nhớ ngoài, giúp mở rộng tài nguyên bộ nhớ trong, thông qua giao diện EMI / FSMC.

## Kết

Nắm rõ cấu trúc và bộ nhớ (memory) của **stm32** sẽ giúp ích rất nhiều trong việc hiểu sâu về vi điều khiển, từ đó trở thành một lập trình viên ưu tú hơn. Nên nhớ để nâng cao trình độ của mình, chúng ta cần phải đào sâu nghiên cứu những thứ tưởng chừng rất đơn

Nếu bài viết này có ích hãy chia sẻ nó và tham gia hội anh em [Nghiên Lập Trình](#) để giao lưu nhé!

4.2/5 - (4 bình chọn)

## Related Posts:

1. [Lập trình STM32 điều khiển LCD1602 chế độ 8bit và 4bit](#)
2. [Bài 16: Lập trình STM32 USB CDC truyền nhận dữ liệu qua cổng COM ảo](#)
3. [Bài 15: Lập trình STM32 Window Watchdog Timer kiểm lỗi phần mềm](#)
4. [Lập trình STM32 từ A tới Z](#)
5. [Bài 12: Lập trình STM32 với giao thức SPI](#)
6. [Bài 11: Lập trình STM32 với Giao thức UART](#)



**KHUÊ NGUYỄN**

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

## 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**

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 đẽ:

- Định vị trẻ em, con cái... Xem thêm



## Bài viết khác

### Lập trình 8051 - AT89S52



Khuê Nguyễn Creator



### 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

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



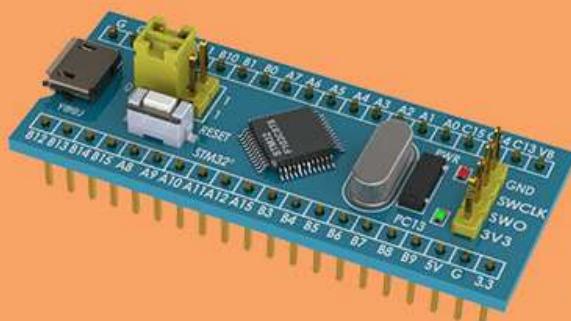
## Lập trình STM32F407 DAC chuyển đổi số sang tương tự

## 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

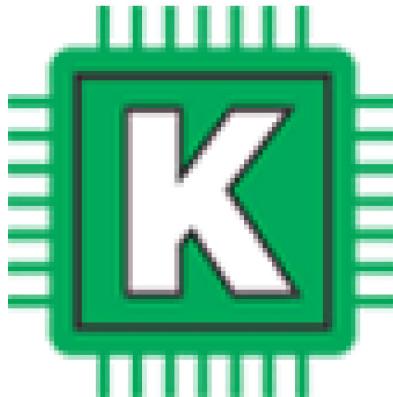


## Cài đặt SDC Complier và Code:Blocks IDE

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

Fanpage: Khuê Nguyên Creator

My Shop

### Thông Tin

Tác Giả

Chính Sách Bảo Mật



Copyright 2022 © Khuê Nguyễn