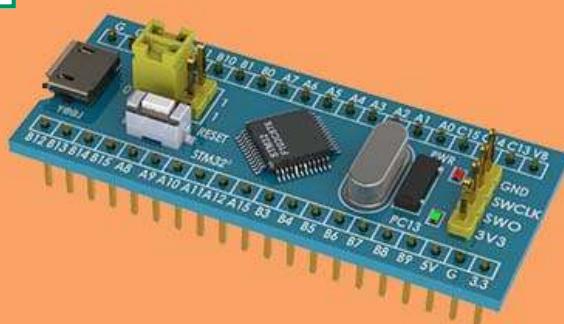
**LẬP TRÌNH STM32**

# Bài 20: Lập trình STM32 Flash đọc, ghi và xóa dữ liệu

POSTED ON 14/08/2021 BY KHUÊ NGUYỄN

14  
Th8**Khuê Nguyễn Creator**

## Bài 20: Lập trình STM32 đọc, ghi, xóa dữ liệu trên Flash

Flash là gì và cách lập trình flash trên STM32, tổ chức bộ nhớ STM32 flash, cách đọc, ghi, xóa bộ nhớ Flash và tại sao nó rất quan trọng trong lập trình. Đó là những kiến thức chúng ta sẽ học trong bài hôm nay.

Bài 20 trong Serie **Học STM32 từ A tới Z**

## Mục Lục



1. Flash là gì?
  - 1.1. Tại sao chúng ta phải sử dụng Flash, EEPROM
  - 1.2. Khác biệt giữa flash và EEPROM
2. Tổ chức bộ nhớ flash trên STM32f103c8t6
3. Đọc, ghi, xóa dữ liệu trên flash
4. Lập trình STM32 Flash đọc ghi dữ liệu trên STM32
  - 4.1. Ghi dữ liệu vào Flash
  - 4.2. Đọc dữ liệu từ STM32 Flash
  - 4.3. Kết quả
5. Kết
  - 5.1. Related posts:

## Flash là gì?

**Bộ nhớ flash** là một loại bộ nhớ máy tính kiểu bộ nhớ điện tĩnh (non-volatile memory), có thể bị xóa và lập trình lại (reprogrammed).

Về mặt kỹ thuật thì bộ nhớ flash có thể được dùng như một loại **EEPROM** mà ở đó nó có thể được đọc/ghi bằng điện và không mất dữ liệu khi ngừng cung cấp điện. Có 2 kiểu bộ nhớ flash chính đã được tạo ra là NAND và NOR được cấu thành từ các cổng logic.

Bộ nhớ flash được cấu thành từ các phần tử (cell) nhớ riêng rẽ với các đặc tính bên trong giống như những cổng logic tương ứng đã tạo ra nó; do đó, ta có thể thực hiện thao tác đọc/ ghi, lưu trữ dữ liệu theo từng phần tử (cell) nhớ một.

## Tại sao chúng ta phải sử dụng Flash, EEPROM

Bộ nhớ flash ngoài việc để lưu chương trình mà **lập trình** viên nạp vào, còn có một công dụng khác đó là lưu trữ dữ liệu quan trọng, không bị mất đi mỗi khi reset hoặc tắt điện.

Lấy ví dụ: Chiếc điều hòa nhà bạn, khi bạn set 27 độ và tắt điện điều hòa đi. Thì khi mở lên chúng vẫn sẽ chạy ở nhiệt độ 27 độ. Điều đó nghĩa là setting của bạn

đã được lưu lại, và chúng sẽ được lưu ở Flash hoặc EEPROM. ( ROM sẽ không thể đọc, ghi nhiều lần)

## Khác biệt giữa flash và EEPROM

Bộ nhớ flash kiểu **cổng NAND** có thể được ghi và đọc theo từng khối (block) hoặc trang (page) nhớ, còn bộ nhớ flash kiểu **cổng NOR** thì có thể được đọc hoặc ghi một cách độc lập theo từng từ (word) hoặc byte nhớ của máy.

Thông thường bộ nhớ Flash trên vi điều khiển sẽ được thiết kế kiểu NAND vậy nên chúng ta phải đọc ghi theo từng khối và page.

EEPROM sẽ được đọc ghi theo từ byte và word giống NOR flash.

Vậy nên hãy lưu ý phần này khi lập trình với Flash trên vi điều khiển nhé.

## Tổ chức bộ nhớ flash trên STM32f103c8t6

Bộ nhớ STM32 Flash được tổ chức như là 1 khối chính (main block) lên đến 64 Kb (hoặc 128 Kb) chia thành 128 pages, mỗi page 1 Kbytes (medium-density devices) và 1 khối thông tin (information block). Bắt đầu từ địa chỉ 0x0800 0000, bộ nhớ chương trình sẽ được lưu ở đây.

**Table 3. Flash module organization (medium-density devices)**

Block	Name	Base addresses	Size (bytes)
Main memory	Page 0	0x0800 0000 - 0x0800 03FF	1 Kbyte
	Page 1	0x0800 0400 - 0x0800 07FF	1 Kbyte
	Page 2	0x0800 0800 - 0x0800 0BFF	1 Kbyte
	Page 3	0x0800 0C00 - 0x0800 0FFF	1 Kbyte
	Page 4	0x0800 1000 - 0x0800 13FF	1 Kbyte
	⋮	⋮	⋮
	Page 127	0x0801 FC00 - 0x0801 FFFF	1 Kbyte
	System memory	0x1FFF F000 - 0x1FFF F7FF	2 Kbytes
Information block	Option Bytes	0x1FFF F800 - 0x1FFF F80F	16

# Đọc, ghi, xóa dữ liệu trên flash

STM32 có một bộ nhớ NAND Flash được nhúng sẵn trong vi điều khiển. Tất cả mô tả về việc lập trình sẽ được mô tả trong mục 3.3.3 Embedded flash memory của [reference manual stm32](#)

Để đọc chính xác dữ liệu từ bộ nhớ Flash, số lượng các trạng thái chờ/độ trễ (**LATENCY**) phải được lập trình trong thanh ghi **FLASH\_ACR** (Flash access control register) đúng với khoảng điện áp cung cấp của thiết bị và tần số clock của CPU (HCLK).

**Lưu ý:** Nếu bạn không setting đúng sẽ dẫn tới trường hợp, khi nâng xung nhịp của hệ thống lên cao sẽ không đọc được flash.

Trong thư viện STM32\_HAL\_Flash. Phần setting này đã được lập trình đúng nên các bạn có thể yên tâm sử dụng nhé

Chi tiết sẽ được mô tả trong bảng sau:

**Table 5. Number of wait states according to CPU clock (HCLK) frequency**

Wait states (WS) (LATENCY)	HCLK (MHz)			
	Voltage range 2.7 V - 3.6 V	Voltage range 2.4 V - 2.7 V	Voltage range 2.1 V - 2.4 V	Voltage range 1.71 V - 2.1 V
0 WS (1 CPU cycle)	0 < HCLK ≤ 30	0 < HCLK ≤ 24	0 < HCLK ≤ 18	0 < HCLK ≤ 16
1 WS (2 CPU cycles)	30 < HCLK ≤ 64	24 < HCLK ≤ 48	18 < HCLK ≤ 36	16 < HCLK ≤ 32
2 WS (3 CPU cycles)	64 < HCLK ≤ 90	48 < HCLK ≤ 72	36 < HCLK ≤ 54	32 < HCLK ≤ 48
3 WS (4 CPU cycles)	90 < HCLK ≤ 100	72 < HCLK ≤ 96	54 < HCLK ≤ 72	48 < HCLK ≤ 64
4 WS (5 CPU cycles)	-	96 < HCLK ≤ 100	72 < HCLK ≤ 90	64 < HCLK ≤ 80
5 WS (6 CPU cycles)	-	-	90 < HCLK ≤ 100	80 < HCLK ≤ 96
6 WS (7 CPU cycles)	-	-	-	96 < HCLK ≤ 100

Thao tác xóa và ghi trên STM32 Flash phải thực hiện theo page (trang) hoặc xóa toàn bộ chip (mass erase).

Ngoài ra chúng ta cần khóa và mở khóa vùng nhớ để bảo vệ dữ liệu chứa trong đó.

# Lập trình STM32 Flash đọc ghi dữ liệu trên STM32

Đầu tiên chúng ta sẽ định nghĩa page bắt đầu và kết thúc

```

24  /* LAYOUT Includes */
25  /* USER CODE BEGIN Includes */
26  #define FLASH_ADDR_PAGE_126 ((uint32_t)0x0801F810) //Page 126
27  #define FLASH_ADDR_PAGE_127 ((uint32_t)0x0801FC00) //Page 127
28
29  #define FLASH_USER_START_ADDR    FLASH_ADDR_PAGE_126
30  #define FLASH_USER_END_ADDR      FLASH_ADDR_PAGE_127 + FLASH_PAGE_SIZE
31
32  uint32_t startpage = FLASH_USER_START_ADDR;
33  uint32_t dataread;
34  //uint32_t *datain = "Hello World";
35  /* USER CODE END Includes */
36
37  /* Private typedef ----- */
38  /* USER CODE BEGIN PTD */
39  /* */
40  /* Private Data for reading from memory address */

```

## Ghi dữ liệu vào Flash

1. Mở khóa STM32 Flash
2. Tạo biến struct để xóa
3. Chọn kiểu xóa theo Page
4. Địa chỉ bắt đầu xóa
5. Số lượng page cần xóa
6. Sau đó gọi câu lệnh xóa và truyền vào struct xóa
7. Tiếp tới chúng ta có thể ghi dữ liệu vào bằng lệnh program, vì là ghi kiểu word (32 bit = 4byte). Vậy nên để ghi ô nhớ ngay sau đó, chúng ta sẽ cộng thêm 4 byte cho start page nhé.

```

72 void FLASH_WritePage(uint32_t startPage, uint32_t endPage, uint32_t data32)
73 {
74     HAL_FLASH_Unlock();
75     FLASH_EraseInitTypeDef EraseInit;
76     EraseInit.TypeErase = FLASH_TYPEERASE_PAGES;
77     EraseInit.PageAddress = startPage;
78     EraseInit.NbPages = (endPage - startPage)/FLASH_PAGE_SIZE;
79     uint32_t PageError = 0;
80     HAL_FLASHEx_Erase(&EraseInit, &PageError);
81     HAL_FLASH_Program(FLASH_TYPEPROGRAM_WORD, startPage, data32); //4 byte đầu tiên
82     //HAL_FLASH_Program(FLASH_TYPEPROGRAM_WORD, startPage + 4, 0x34567890); // 4byte tiếp theo
83     HAL_FLASH_Lock();
84 }

```

## Đọc dữ liệu từ STM32 Flash

Đọc dữ liệu ra chúng ta chỉ cần truy xuất theo địa chỉ kiểu 32bit.

```

84  }
85  uint32_t FLASH_ReadData32(uint32_t addr)
86  {
87      uint32_t data = *(__IO uint32_t *) (addr);
88      return data;
89  }

```

Cuối cùng, trước while (1) hãy thử ghi dữ liệu vào và đọc ra nhé.

**Lưu ý: Flash sẽ cần 1 khoảng thời gian xóa và ghi vậy nên chúng ta không nên xóa và ghi liên tục như biến, hãy lưu flash khi có 1 sự kiện gì đó mà thôi.**

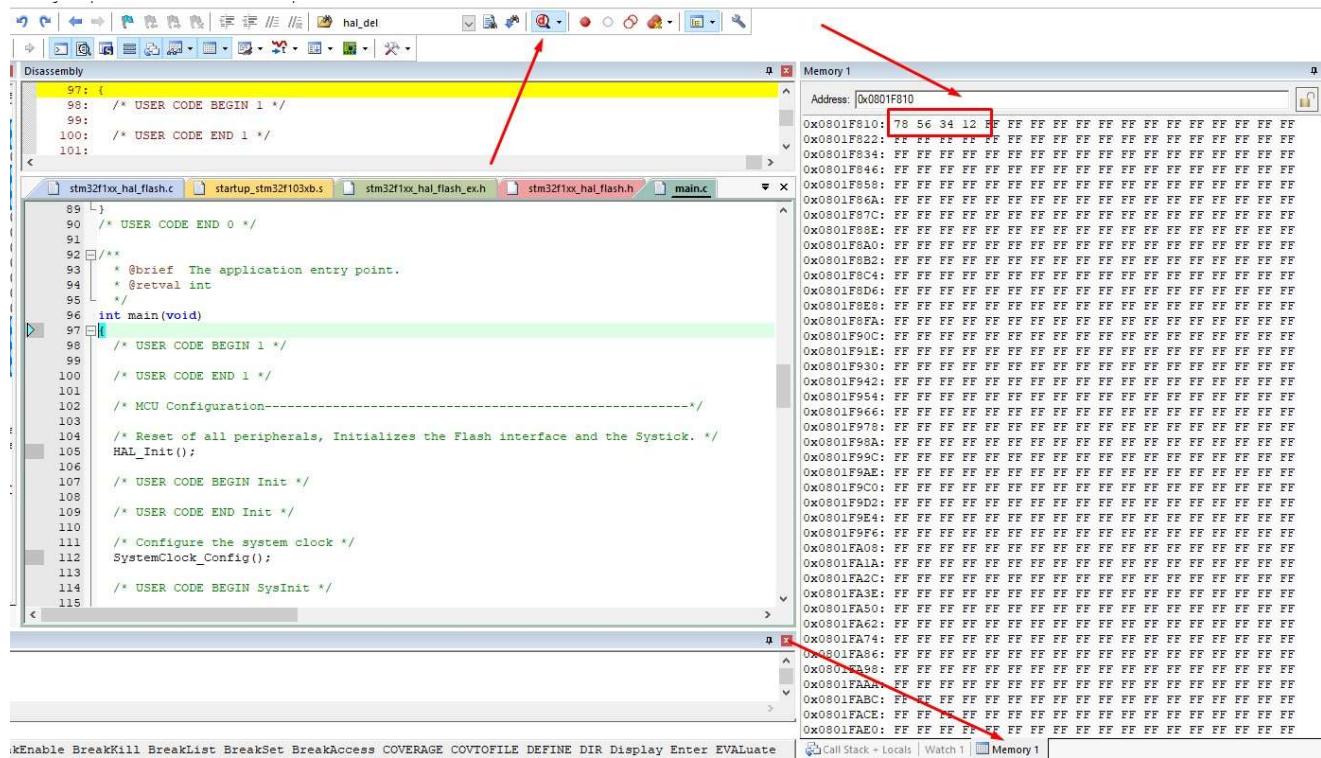
```

117
118     /* Initialize all configured peripherals */
119     MX_GPIO_Init();
120
121     /* USER CODE BEGIN 2 */
122     FLASH_WritePage(FLASH_USER_START_ADDR, FLASH_USER_END_ADDR, 0x12345678);
123     dataread = FLASH_ReadData32(FLASH_USER_START_ADDR);
124
125     /* USER CODE END 2 */
126
127     /* Infinite loop */
128     /* USER CODE BEGIN WHILE */
129     while (1)
130     {
131         /* USER CODE END WHILE */

```

## Kết quả

Nạp và nhấn Debug. Dưới tab Memory chúng ta copy địa chỉ start page. Nhấn Run và xem kết quả.



## Kết

Lập trình STM32 Flash rất quan trọng, các ứng dụng đa phần đều cần sử dụng flash để lưu trữ dữ liệu. Lập trình với Flash cũng tương đối đơn giản và dễ dàng. Tuy nhiên các bạn không nên xóa và ghi flash quá nhiều lần trong 1 thời gian ngắn, có thể làm sai dữ liệu nhận được.

Nếu cảm thấy bài viết này có ích hãy chia sẻ với bạn bè nhé. Đừng quên ra nhập hội những anh em **Nghịen lập trình** nhé!!

5/5 - (1 bình chọn)

## Related Posts:

1. [Lập trình STM32 đọc nhiệt độ với DS18b20 giao tiếp OneWire](#)
2. [Hướng dẫn download và sử dụng tài liệu Lập trình STM32](#)
3. [Bài 14: Sử dụng STM32 IWDG Independent Watchdog Timer chống treo vi điều khiển](#)
4. [Bài 10: Giao thức I2C, lập trình STM32 với module RTC DS3231](#)
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

## 2 THOUGHTS ON “BÀI 20: LẬP TRÌNH STM32 FLASH ĐỌC, GHI VÀ XÓA DỮ LIỆU”



### Tú Huỳnh says:

Tạo biến struct để xóa

Chọn kiểu xóa theo Page

Địa chỉ bắt đầu xóa

Số lượng page cần xóa

chỗ này ban đầu em đọc 3 dòng cuối hơi lúng túng vì không biết nó ở đâu ra. may là xuống dưới tí có cái ảnh ví dụ em mới : à 3 dòng đó là các biến định nghĩa trong struct xóa của cái dòng ở trên ^^.

10/12/2021 AT 1:07 CHIỀU

TRẢ LỜI



### Khuê Nguyễn says:

^^!

25/12/2021 AT 4:36 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 - Họ...  
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 😊

Chỉnh sửa và xác nhận nhằm định rõ thông tin

<https://khuenguyencreator.com/lap-trinh-stm32-flash-doc-ghi-va-xoa-du-lieu/>

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



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



Khuê Nguyễn Creator



### Tổng quan về 8051

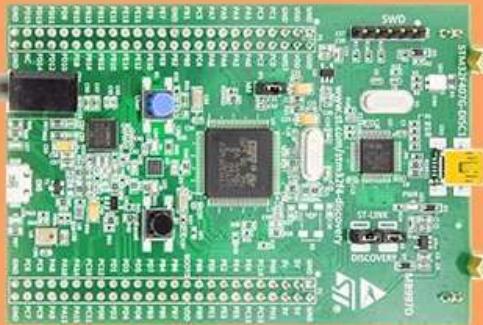
8051 là một dòng chip nhập môn cho lập trình viên nhúng, chúng được sử...

[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

<https://khuenguyencreator.com/lap-trinh-stm32-flash-doc-ghi-va-xoa-du-lieu/>

11/16

[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

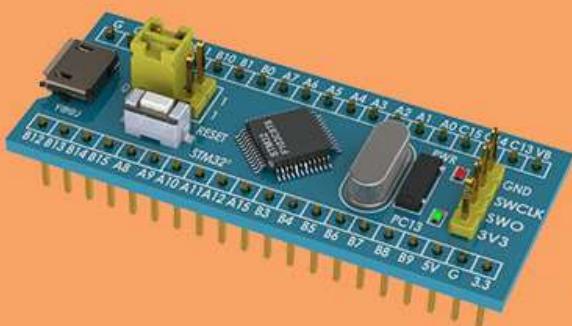
# chuyển đổi số sang tương tự

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

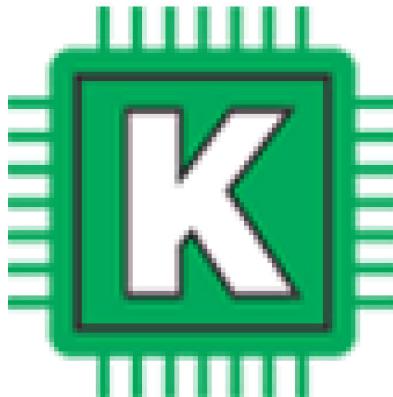


## Cài đặt SDC Complier và Code:Blocks IDE

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: Nghiện Lập Trình

Fanpage: Khuê Nguyên Creator

My Shop

### Thông Tin

Tác Giả

## Chính Sách Bảo Mật



Copyright 2022 © Khuê Nguyễn