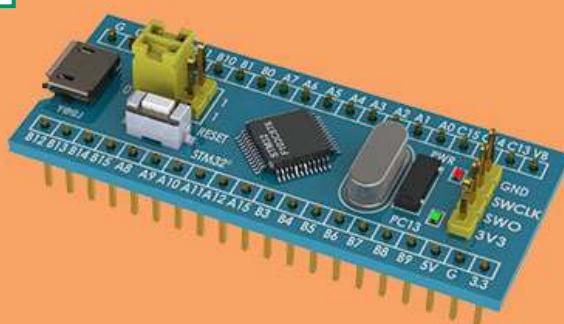


**LẬP TRÌNH STM32**

Bài 14: Sử dụng STM32 IWDG Independent Watchdog Timer chống treo vi điều khiển

POSTED ON 28/02/2021 BY KHUÊ NGUYỄN

28
Th2**Khuê Nguyễn Creator**

Bài 14: Sử dụng STM32 IWDG Chống treo vi điều khiển

STM32 IWDG Independent Watchdog Timer (IWDG) là một loại Timer rất hay được sử dụng trong thực tế, chức năng chủ yếu của Timer này là kiểm tra vi điều khiển có bị treo hay hoạt động sai không. Từ đó đưa ra lệnh Reset, khiến chương trình chạy lại từ đầu.

Bài 14 trong serie **Học lập trình STM32 từ A tới Z**

Mục Lục

- 
1. Watchdog Timer là gì?
 - 1.1. STM32 IWDG trong vi điều khiển STM32
 2. Sự khác nhau giữa IWDG và WWDG
 - 2.1. Independent Watchdog Timer
 - 2.2. Window Watchdog Timer
 3. Independent Watchdog Timer trong STM32
 - 3.1. Sơ đồ khối của bộ STM32 IWDG
 - 3.2. Nguyên lý hoạt động của STM32 IWDG
 - 3.3. Các tham số cần quan tâm
 4. Lập trình STM32 IWDG
 - 4.1. Cấu hình STM32 IWDG trên CubeMX
 - 4.2. Lập trình STM32 IWDG trên KeilC
 5. KẾT
 - 5.1. Related posts:

Watchdog Timer là gì?

Watchdog dịch ra nghĩa là Nhìn Chó Đùa đấy =)), Theo Google dịch
Watchdog là người canh gác, bảo vệ việc các tác nhân bên trong hoặc bên
ngoài không mong muốn làm ảnh hưởng tới sự hoạt động của vi điều khiển.



Bảo vệ của MCU

STM32 IWDG trong vi điều khiển STM32

Watchdog Timer là bộ ngoại vi tích hợp trên vi điều khiển STM32 có khả năng giúp cho người dùng phát hiện ra hệ thống bị treo, chạy sai và tạo ra một ngắt hoặc một tín hiệu reset chip.

Đối với các sản phẩm điện tử được phát hành ra ngoài thị trường, thì do nhiều yếu tố bên trong (code chưa chặt chẽ, code có bug...) và yếu tố bên ngoài (nhiều, điều kiện nhiệt độ, độ ẩm) dẫn đến việc hoạt động sai của VDK dẫn đến các hiện tượng chạy chức năng bị sai, hoặc treo chip. Trong trường hợp như vậy chúng ta cần có các biện pháp khắc phục tạm thời ví dụ như reset chip để chạy lại chương trình. Thì bộ Watchdog Timer được sinh ra với mục đích tạo ra một tín hiệu reset bằng Software (điều khiển ngắt bằng phần mềm).

Trong STM32 Watchdog Timer có 2 loại đó là:

- Independent Watchdog Timer hay STM32 IWDG
- Window Watchdog Timer hay STM32 WWDG

Sự khác nhau giữa IWDG và WWDG

IWDG và WWDG được tạo ra chung một mục đích thế nhưng chúng có sự khác nhau một chút về cách hoạt động và mục đích hoạt động.

Independent Watchdog Timer

Đúng như cái tên Independent chính là độc lập, nghĩa là bộ Timer này sử dụng nguồn xung LSI (Low-speed clock) vì vậy chúng có thể hoạt động ngay cả khi nguồn clock của chương trình chính không hoạt động. Điều này phù hợp với chức năng kiểm soát lỗi treo chip kể cả do phần cứng hoặc phần mềm. Khi ngắt IWDG xảy ra, MCU sẽ lập tức bị reset mà không cần làm gì cả.

Có thể tưởng tượng IWDG là một người bảo vệ của tòa nhà, trong tòa nhà đó có 1 công ty là MCU của các bạn. Vì bảo vệ đó không phải nhân viên của Công Ty MCU vậy nên cho dù MCU có hoạt động hay không hoạt động, gã bảo vệ đó vẫn sẽ thực hiện công việc của mình.

Window Watchdog Timer

Với WWDG chúng có nguồn xung từ bộ **APB1 Clock** nghĩa là bộ Timer này sẽ hoạt động khi chương trình hoạt động. Một khi xảy ra treo hệ thống liên quan tới Clock chính. Bộ Timer này sẽ không hoạt động.

Vậy nên chức năng chính của WWDG là kiểm xoát lỗi phần mềm (các Bug), nếu các tác vụ (Task) hoạt động bất thường như kết thúc sớm hơn hoặc muộn hơn dự kiến, ngắt WWDG sẽ xảy ra, ngắt WWDG cho phép chương trình thực thi một số lệnh trong ngắt trước khi Reset hoàn toàn MCU.

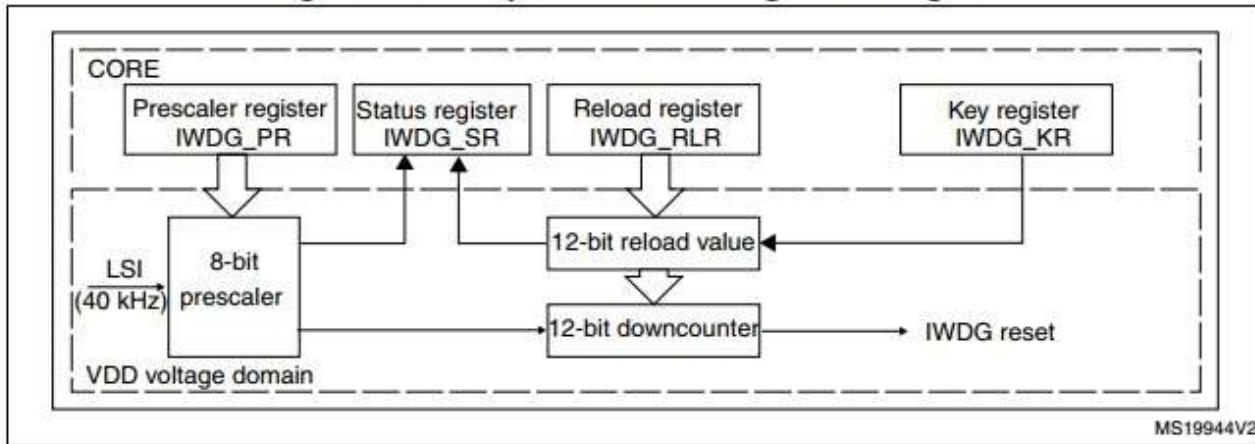
Có thể tưởng tượng WWDG chính là một người quản lý của công ty. Người này có chức năng kiểm xoát sự hoạt động của nhân viên, ghi lại các lỗi nếu thấy nhân viên hoạt động sai. Từ đó tìm ra lỗi, các phần tử không đạt yêu cầu. Tất

nhiên, nếu toàn bộ công ty đó không hoạt động, bộ WWDG này cũng không hoạt động.

Independent Watchdog Timer trong STM32

Sơ đồ khối của bộ STM32 IWDG

Figure 182. Independent watchdog block diagram



Sơ đồ khối của bộ IWDG

Bao gồm:

1. IWDG_PR: Thanh ghi hệ số chia, thanh ghi này sẽ xác định thời gian mỗi 1 lần giảm của counter
2. IWDG_SR: Thanh ghi trạng thái lưu trạng thái prescaler update và reload update khi có sự kiện ghi vào 2 thanh ghi này
3. IWDG_RLR: Thanh ghi lưu giá trị nạp lại để nạp vào bộ 12-bit reaload counter
4. IWDG_KR: Thanh ghi điều khiển việc chạy và nạp lại cho bộ IWDG

Nguyên lý hoạt động của STM32 IWDG

- IWDG_PR và IWDG_RLR được nạp vào xác định chu kì tràn của downcounter
- Khi nạp vào thanh ghi IWDG_KR giá trị 0xCCCC timer sẽ bắt đầu hoạt động

- Trước khi downcounter đếm từ giá trị nạp lại tới 0 nếu ghi vào IWDG_KR giá trị 0xAAAA thì bộ counter sẽ đếm lại từ đầu (giá trị reload)
- Nếu downcounter đếm đến 0, một sự kiện reset sẽ được sinh ra và reset chip

Vậy nên để chip không bị reset chúng ta cần nạp 0xAAAA vào IWDG_KR trước khi counter đếm đến 0.

Các tham số cần quan tâm

Khi khởi tạo Timer trước khi ghi vào 2 thanh ghi PR và RLR cần ghi vào KEY giá trị 0x5555 để bỏ chế độ bảo vệ của timer.

Chu kỳ đếm của Timer xác định bởi hệ số chia và giá trị thanh ghi Reload, theo công thức.

$$\text{Timeout} \approx \frac{(4 \times RLR \times 2^{PR})}{\text{LSI clock frequency}}$$

Trong chế độ Debug để chip không reset liên tục ta cần enable bit DBG_IWDG_STOP để tắt timer này đi.

Lập trình STM32 IWDG

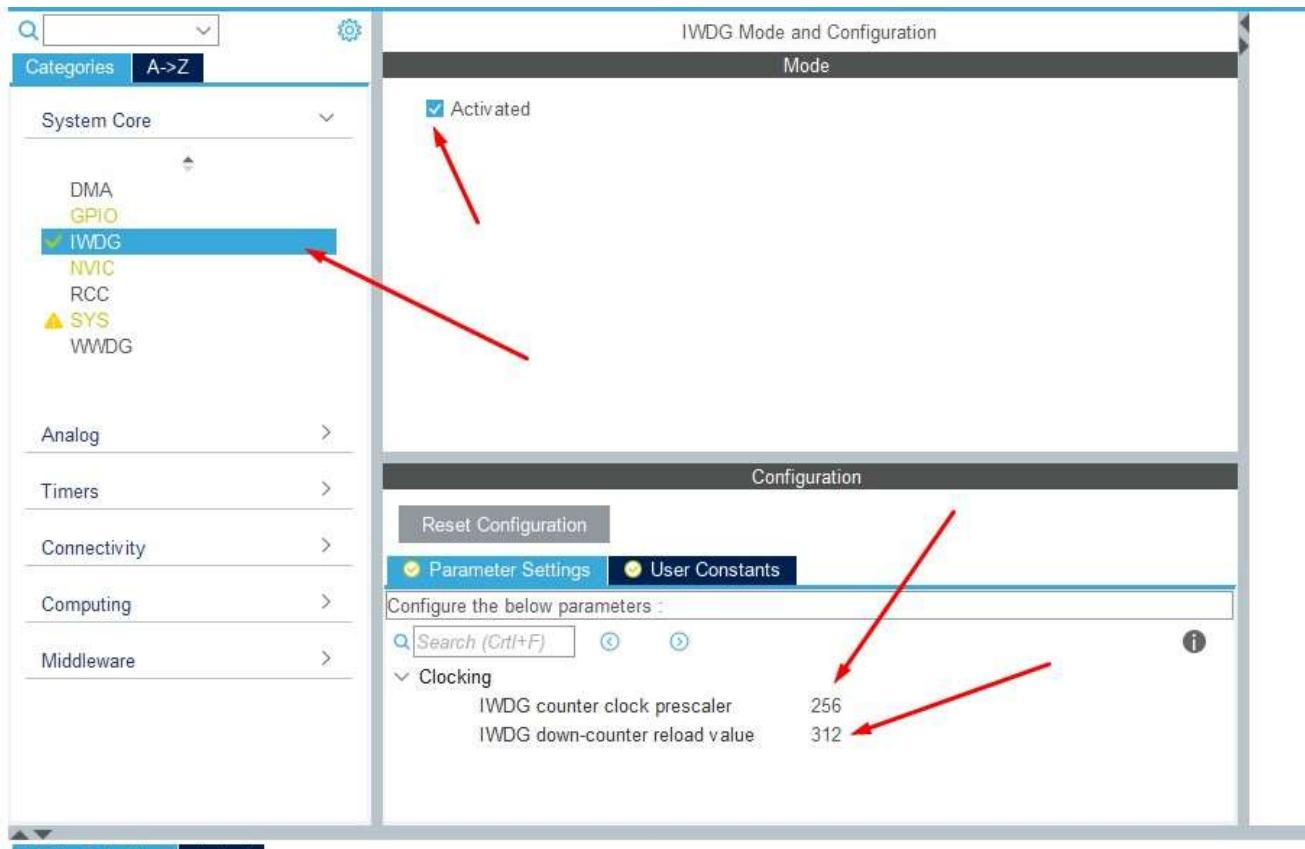
Kịch bản đặt ra: Trong chương trình chính ta sẽ đảo trạng thái led PCB13 liên tục, sau mỗi lần đảo trạng thái chúng ta sẽ làm mới timer, vậy nên MCU sẽ không reset.

Khi nhấn nút tại PA0, chương trình chính sẽ đi vào while(1) và không làm mới timer, vậy nên led sẽ đứng yên tại thời điểm nhấn nút sau mấy giây rồi bắt đầu lại nháy như thường

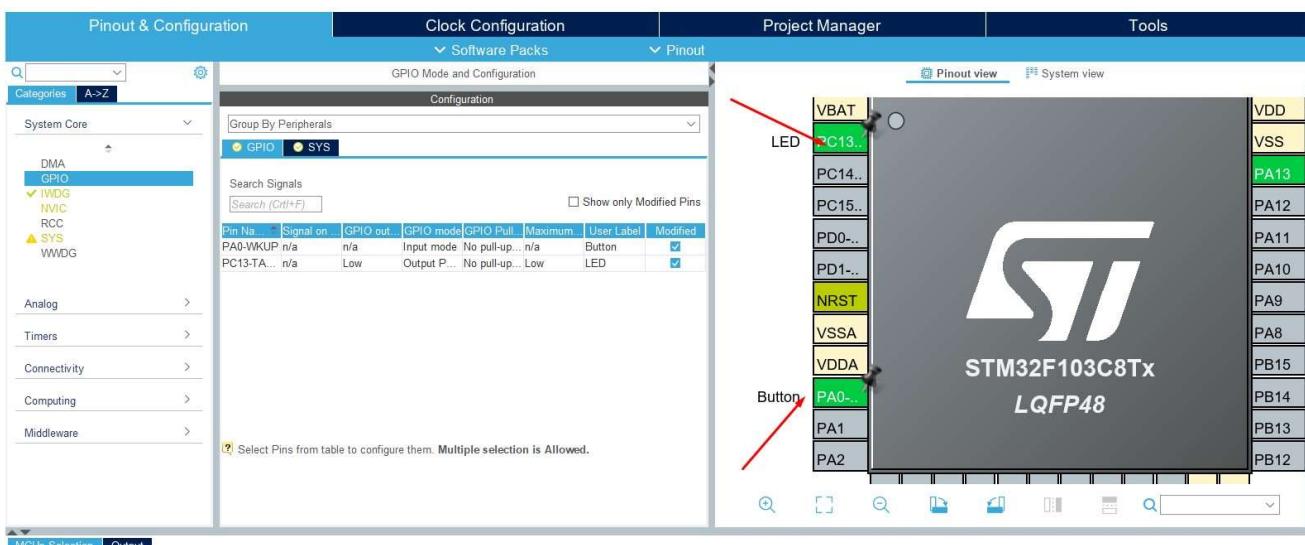
Cấu hình STM32 IWDG trên CubeMX

Mở CubeMX chọn chip, trong SYS chọn debug wire

Trong tab IWDG chọn Active và set thông số như hình. Ta sẽ có chu kỳ tràn của Timer là 2s



Trong tab GPIO chọn PC13 là GPIO_OUT đặt tên là Led, PA0 là GPIO_IN đặt là Button



Gen code và mở trong keilc

Lập trình STM32 IWDG trên KeilC

Trước while(1) chúng ta khởi tạo bộ IWDG, trong while đảo trạng thái led mỗi 0,5s, sau đó làm mới timer.

```

88  /* Initialize all configured peripherals */
89  MX_GPIO_Init();
90  MX_IWDG_Init();
91  /* USER CODE BEGIN 2 */
92  HAL_IWDG_Init(&hiwdg); // khai tao IWDG
93
94  /* USER CODE END 2 */
95
96  /* Infinite loop */
97  /* USER CODE BEGIN WHILE */
98  while (1)
99  {
100     /* USER CODE END WHILE */
101
102     /* USER CODE BEGIN 3 */
103     HAL_GPIO_TogglePin(LED_GPIO_Port, LED_Pin);
104     HAL_Delay(500);
105     HAL_IWDG_Refresh(&hiwdg); // Lam moi Timer
106     if(HAL_GPIO_ReadPin(Button_GPIO_Port, Button_Pin) == 0)
107     {
108         while(1); // vong lap vo han
109     }
110 }
```

Khi nút nhấn PA0 được bấm, chương trình sẽ rơi vào vòng lặp vô hạn. Khi vào while 1 đủ 2s, MCU sẽ tự reset và bắt đầu nháy lại từ đầu

KẾT

STM32 IWDG là 1 Timer được sử dụng khá phổ biến, chúng sẽ rất hữu dụng khi đối phó với các tác nhân không thể kiểm soát được từ bên ngoài làm hệ thống hoạt động sai.

Chúc các bạn học tập thật tốt, và nhớ đánh giá bài viết này nếu thấy hay nhé !!!

5/5 - (4 bình chọn)

Related Posts:

1. [Bản đồ bộ nhớ \(Memory map\) vi điều khiển STM32F103](#)
2. [Lập trình STM32 quét ma trận phím Keypad 3x4](#)
3. [Lập trình STM32 từ A tới Z](#)
4. [Bài 12: Lập trình STM32 với giao thức SPI](#)
5. [Bài 5: Lập trình Timer STM32 chớp tắt Led dùng ngắt](#)

6. Bài 4: Lập trình Ngắt Ngoài STM32 EXTI



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 14: SỬ DỤNG STM32 IWDG INDEPENDENT WATCHDOG TIMER CHỐNG TREO VI ĐIỀU KHIỂN”



minh anh says:

set thông số 256 và 312 → chu kì tràn của Timer là 2s: Ad giải thích giúp mình tính toán ntn được k ah. Thanks ad nhiều

05/03/2021 AT 11:55 CHIỀU

TRẢ LỜI



Đậu says:

Các bài viết của anh rất hay. Cảm ơn anh rất nhiều. Nếu có thời gian anh làm thêm kênh Youtube quay lại màn hình quá trình làm việc và giải thích thì sẽ tuyệt vời hơn nhiều. Chúc anh luôn thành công

20/05/2021 AT 2:01 CHIỀU

TRẢ LỜI



Khuê Nguyễn says:

Cám ơn em, anh cũng có kênh youtube nhưng chưa đầy mạnh dc thôi.

20/05/2021 AT 11:02 CHIỀU

TRẢ LỜI

Trả lời

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

Bình luận *

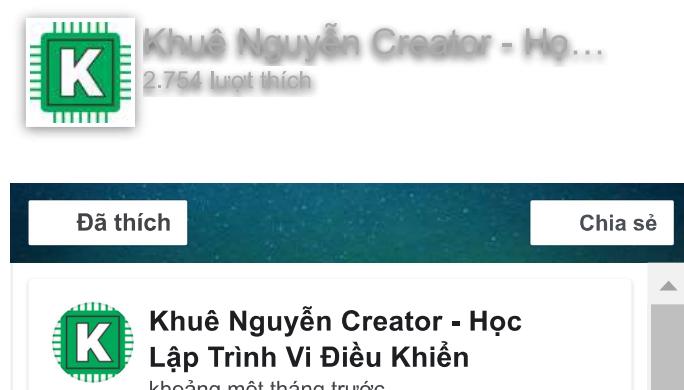
Tên *

Email *

Trang web

PHẢN HỒI

Fanpage



Khuê Nguyễn Creator - Học Lập Trình Vi Điều Khiển

2.754 lượt thích

Đã thích Chia sẻ

Khuê Nguyễn Creator - Học Lập Trình Vi Điều Khiển

khoảng một tháng trước

<https://khuenguyencreator.com/su-dung-stm32-iwdg/>

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

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



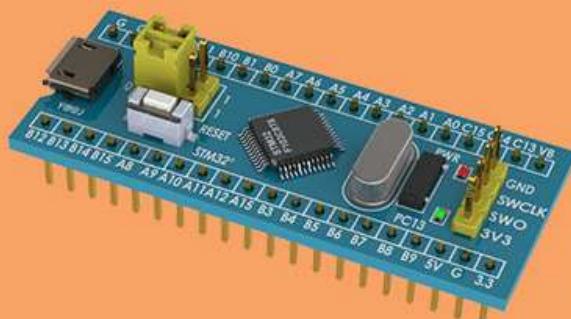
Lập trình STM32F407 DAC chuyển đổi số sang tương tự

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...

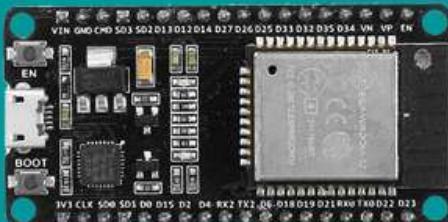
3 COMMENTS

[ĐỌC THÊM](#)

ESP32 và Platform IO



Khuê Nguyễn Creator



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

Lập trình ESP32 FOTA ngp 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



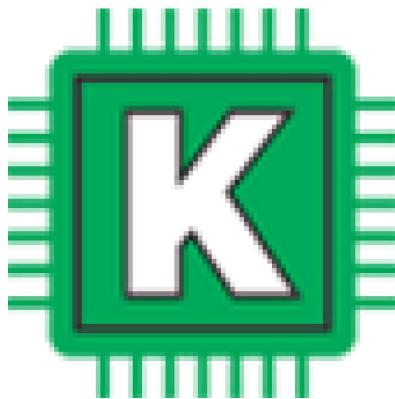
Khuê Nguyễn Creator



Cài đặt SDC Complier và Code:Blocks IDE



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

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: Nghịên Lập Trình

Fanpage: Khuê Nguyên Creator

My Shop

Thông Tin

Tác Giả

Chính Sách Bảo Mật



Copyright 2022 © Khuê Nguyễn