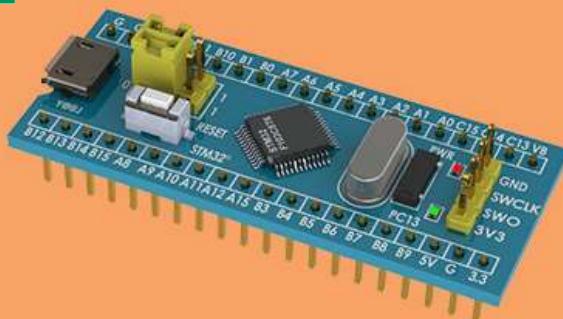


**LẬP TRÌNH STM32**

Bài 16: Lập trình STM32 USB CDC truyền nhận dữ liệu qua cổng COM ảo

POSTED ON 09/03/2021 BY KHUÊ NGUYỄN

09
Th3**Khuê Nguyễn Creator**

Bài 15: STM32 USB CDC truyền nhận dữ liệu

Giao thức USB nói chung và STM32 USB CDC nói riêng là một giao thức khá khó lập trình. Nhưng ST đã hỗ trợ lập trình viên rất nhiều bằng cách tạo ra các gói source Midware USB hỗ trợ rất nhiều cho lập trình. Việc chúng ta cần là học cách sử dụng API của các middleware đó.

Bài 16 trong Serie **Học lập trình STM32 từ A tới Z**



Mục Lục

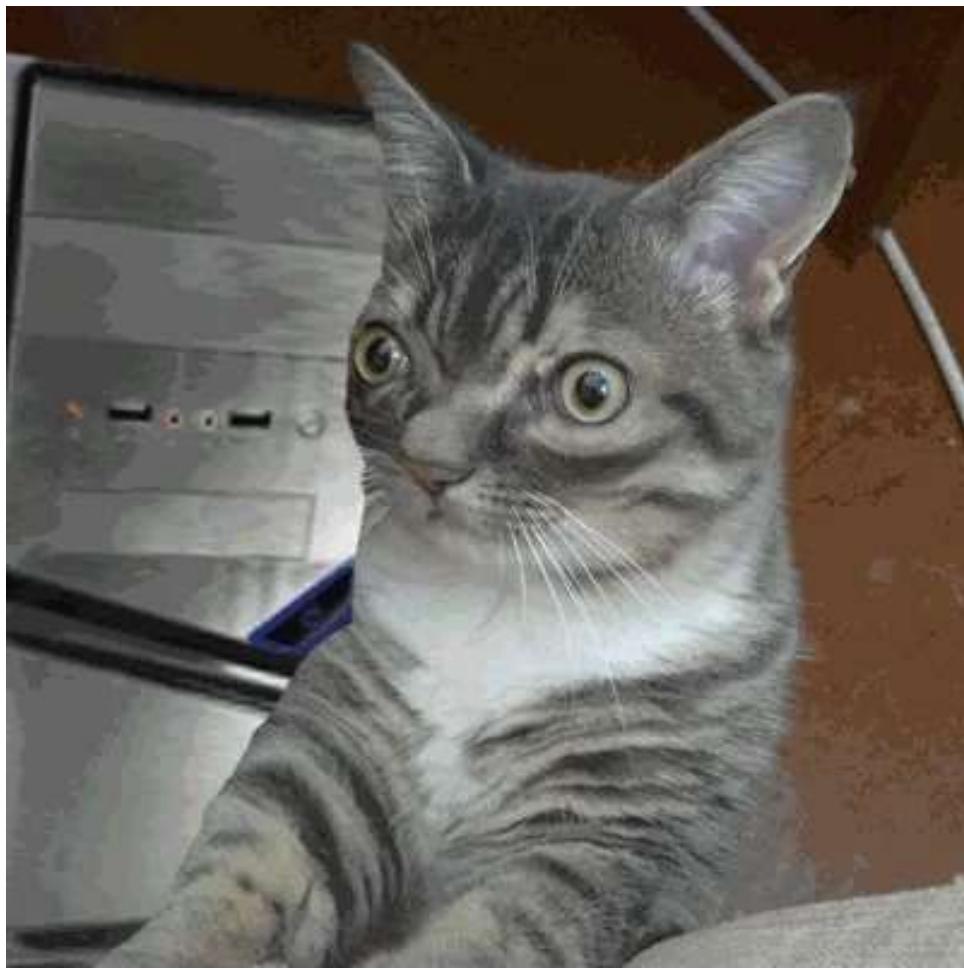
1. Giao thức usb là gì?
2. Một số kiến thức cơ bản về giao thức usb
 - 2.1. Kiến trúc hệ thống USB
 - 2.2. Cấu trúc điều khiển truyền nhận
 - 2.3. Các kiểu truyền nhận trong USB
 - 2.4. Tài liệu về giao thức USB
3. Giao thức Usb cdc trên stm32
 - 3.1. USB CDC là gì?
 - 3.2. STM32 USB CDC
4. Lập trình stm32 usb cdc
 - 4.1. Thiết lập STM32 Usb CDC trên Cube MX
 - 4.2. Lập trình STM32 Usb CDC với Keil C
 - 4.3. Cài đặt Driver STM32 Usb CDC
 - 4.4. Gửi dữ liệu lên STM32 USB CDC với Hercules Terminal
5. Kết
 - 5.1. Related posts:

Giao thức usb là gì?

USB (Universal Serial Bus) là một chuẩn kết nối tuần tự đa dụng trong máy tính. Dùng để kết nối các ngoại vi với máy tính theo quy chuẩn Plug and Play (Cắm và chạy), với tính năng cắm nóng thiết bị (không phải khởi động lại hệ thống) giúp việc kết nối trở nên đơn giản hơn rất nhiều.

Giao thức USB được phát hành vào năm 1996, hiện đang được duy trì bởi tổ chức USB-IF. Theo wikipedia

Đã có bốn thế hệ USB, gồm: **USB 1.x, USB 2.0, USB 3.x** và **USB4**



Giao thức USB rất thông dụng

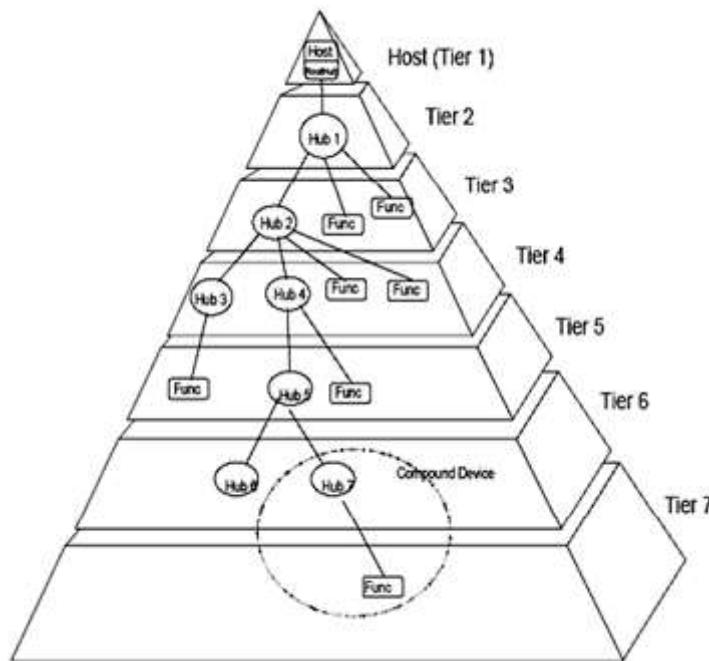
Một số kiến thức cơ bản về giao thức USB

Kiến trúc hệ thống USB

Một hệ thống USB được mô tả bởi 3 định nghĩa:

- USB Host: Chỉ có 1 Host duy nhất trong hệ thống (thông thường sẽ là máy tính cá nhân hoặc máy tính nhúng). Chúng điều khiển mọi luồng dữ liệu và cách thức giao tiếp trong hệ thống.
- USB Device: Trong một hệ thống có thể có nhiều device chúng có thể đóng nhiều vai trò như: Thiết bị đầu cuối tương tác với người dùng (HID), lớp giao tiếp (CDC), lưu trữ dữ liệu (MSP), Mở rộng cổng kết nối (HUB)
- USB Connect: Được hiểu là kiểu kết nối để các thiết bị USB có thể giao tiếp được với Host.

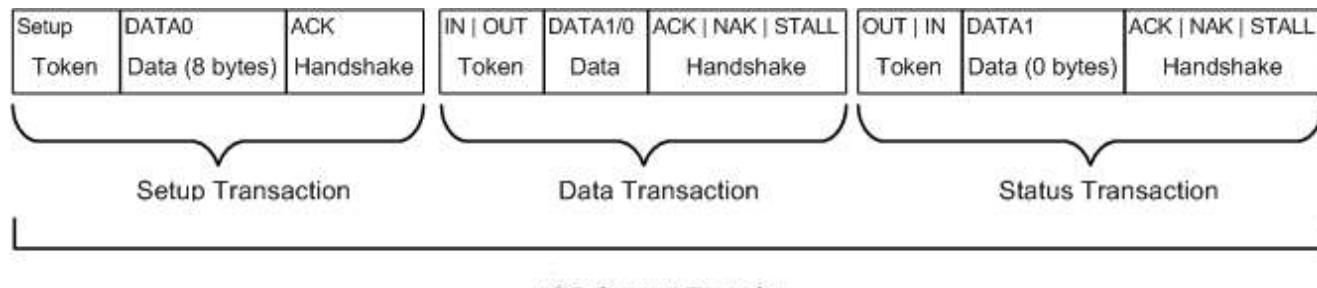
Kiến trúc bus: Về mặt vật lý, kiến trúc Bus USB là một tầng sao, với Host là trung tâm. Mỗi tia là 1 kết nối giữa Host với Hub, Host với Device hoặc Device với Hub. Với 7bit địa chỉ, một Host có thể quản lý 127 thiết bị trong mạng lưới của nó.



hệ thống USB

Cấu trúc điều khiển truyền nhận

Để tất cả các thiết bị trong cùng hệ thống hiểu nhau chúng phải tuân thủ theo quy tắc điều khiển truyền nhận (Control Transfer).



Một gói Control Transfer được chia thành 3 Transaction:

- Setup Transaction: Thiết lập
- Data Transaction: Dữ liệu

- Status Transaction: Trạng thái

Mỗi Transaction lại chia thành các gói tin:

- Setup hoặc IN/OU: Gói thông báo, cho biết rằng các data phía sau dùng để làm gì, được xác định rõ bởi các Mã Token tương ứng với chức năng đó
- DATA: Gói dữ liệu xác định dữ liệu truyền của transaction
- ACK: Gói bắt tay, thông báo dữ liệu truyền thành công hay thất bại

Mỗi gói tin (Packet) lại bao gồm các trường riêng, trong đó có 3 loại trường (fields) bắt buộc trong mỗi gói tin đó là:

- Synch: Trường đồng bộ, thông báo thời điểm bắt đầu gói
- PID: Packet identifier Trường định danh gói tin. Cho biết gói tin này dùng làm gì, hướng dữ liệu của gói. Nếu là gói bắt tay, chúng sẽ cho biết đã truyền nhận thành công hay chưa
- EOP: End of Packet Trường kết thúc, cho biết gói tin đã kết thúc
- Còn lại các trường khác sẽ tùy thuộc vào kiểu gói tin truyền nhận là gì quyết định

Các kiểu truyền nhận trong USB

Chuẩn USB định nghĩa 4 kiểu transfer (cũng có thể gọi là 4 kiểu endpoints), chúng bao gồm:

1. Control transfer
2. Interrupt transfer
3. Bulk transfer
4. Isochronous transfer

Tài liệu về giao thức USB

Nói về USB thì cần rất nhiều tài liệu bạn mới có thể hiểu hết được, thế nhưng để hiểu mức cơ bản thì bạn có thể đọc cuốn USB in a Nutshell. Nó khá cô đọng và không quá dài như các tài liệu khác

Link: <https://drive.google.com/open?id=1wEQKyJVI3d0I6MbVtHehQ1Ciqf878DOC>

Hoặc tham khảo các tài liệu khác tại vidieukhien.org: <http://vidieukhien.org/mot-so-tai-lieu-co-ban-ve-usb.html>

Giao thức Usb cdc trên stm32

USB CDC là gì?

USB communications device class hay được hiểu là Lớp thiết bị giao tiếp thông qua USB. Nghe có vẻ hơi tối nghĩa nhưng bạn có thể hiểu rằng nó được sử dụng để các thiết bị giao tiếp với nhau chứ không tương tác với con người như HID class.

Các thiết bị thuộc lớp này cũng được triển khai trong các hệ thống nhúng như điện thoại di động để điện thoại có thể được sử dụng làm modem, fax hoặc cổng mạng. Các giao diện dữ liệu thường được sử dụng để thực hiện truyền dữ liệu hàng loạt.

STM32 USB CDC

Trong STM32f103c8t6 chỉ hỗ trợ giao thức USB kiểu Device, thế nên ta sẽ sử dụng kit Bluepill như một thiết bị để truyền nhận dữ liệu giữa nó và máy tính. Cách thức giao tiếp đã được chuẩn hóa trong thư viện USB của CubeMx, vậy nên trong bài này chúng ta sẽ không xét tới những phần đó nữa, mà tập trung vào việc sử dụng dữ liệu truyền nhận được.

Lập trình stm32 usb cdc

Thiết lập STM32 Usb CDC trên Cube MX

Mở Cubemx lên, trong tab SYS chọn debug Serial Wire

Clock Configuration

SYS Mode and Configuration

Mode

Debug: Serial Wire

System Wake-Up

Timebase Source: SysTick

Configuration

Tab RCC chúng ta chọn nguồn dao động là thạch anh ngoại HSE.

RCC Mode and Configuration

Mode

High Speed Clock (HSE): Crystal/Ceramic Resonator

Low Speed Clock (LSE): Disable

Master Clock Output

Configuration

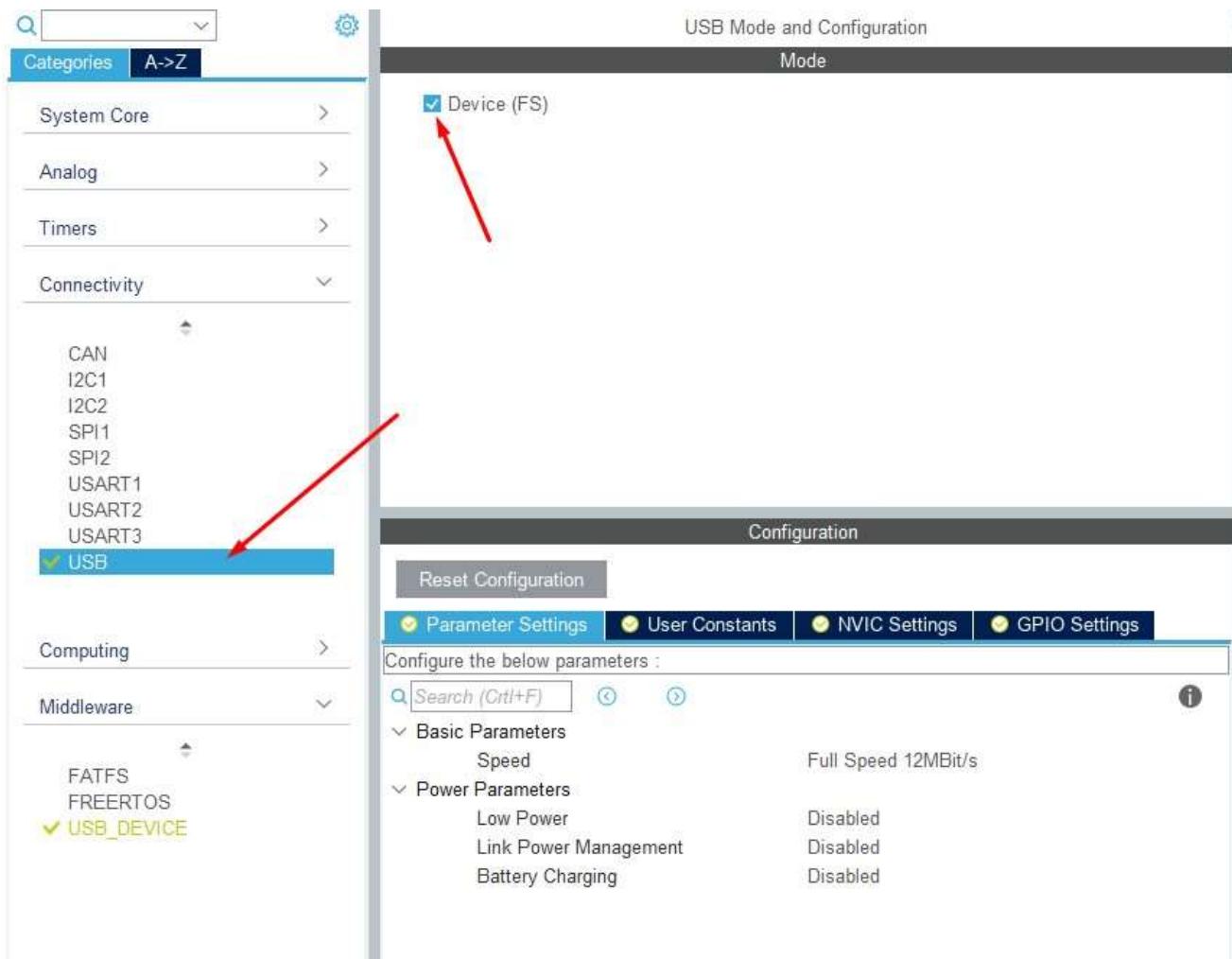
Reset Configuration

Parameter Settings User Constants NVIC Settings GPIO Settings

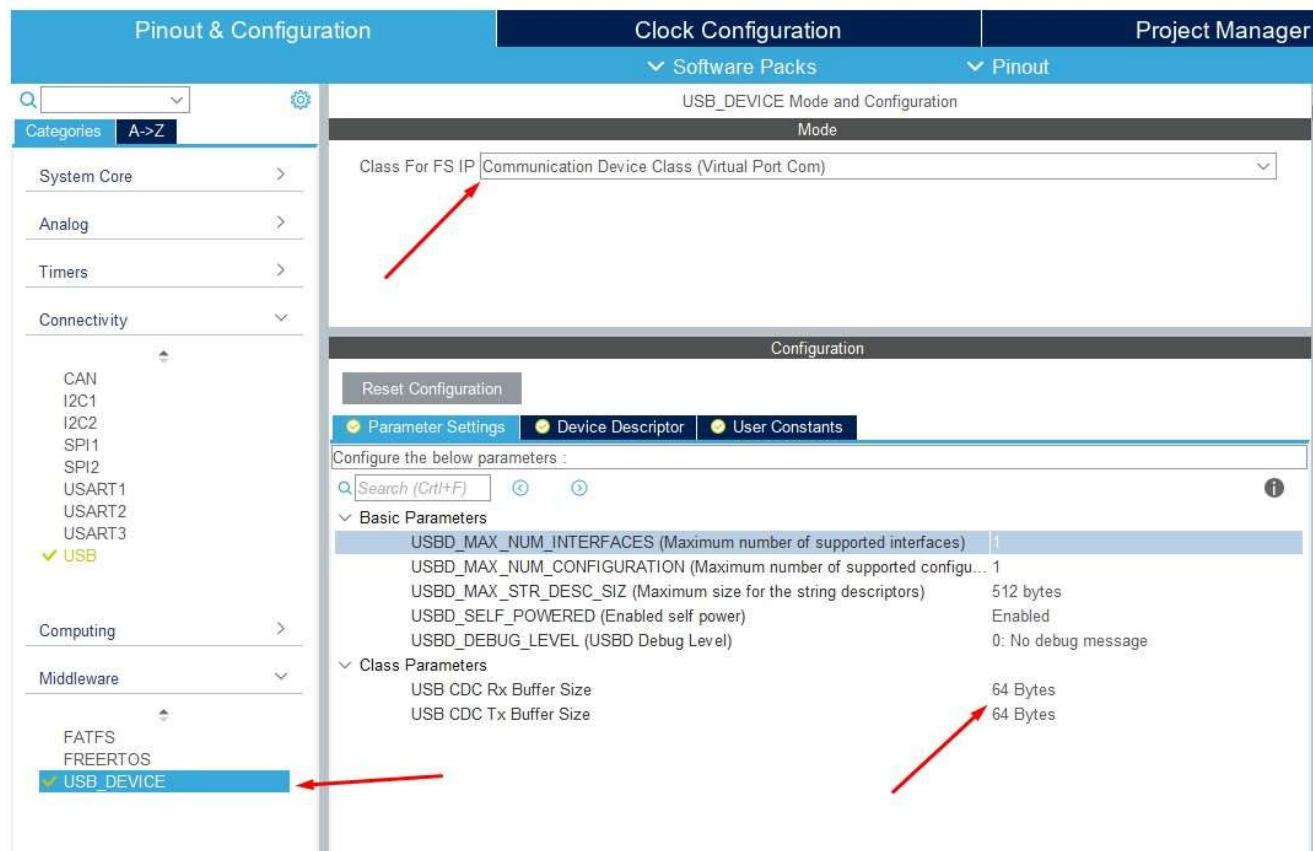
Configure the below parameters :

Search (Ctrl+F)

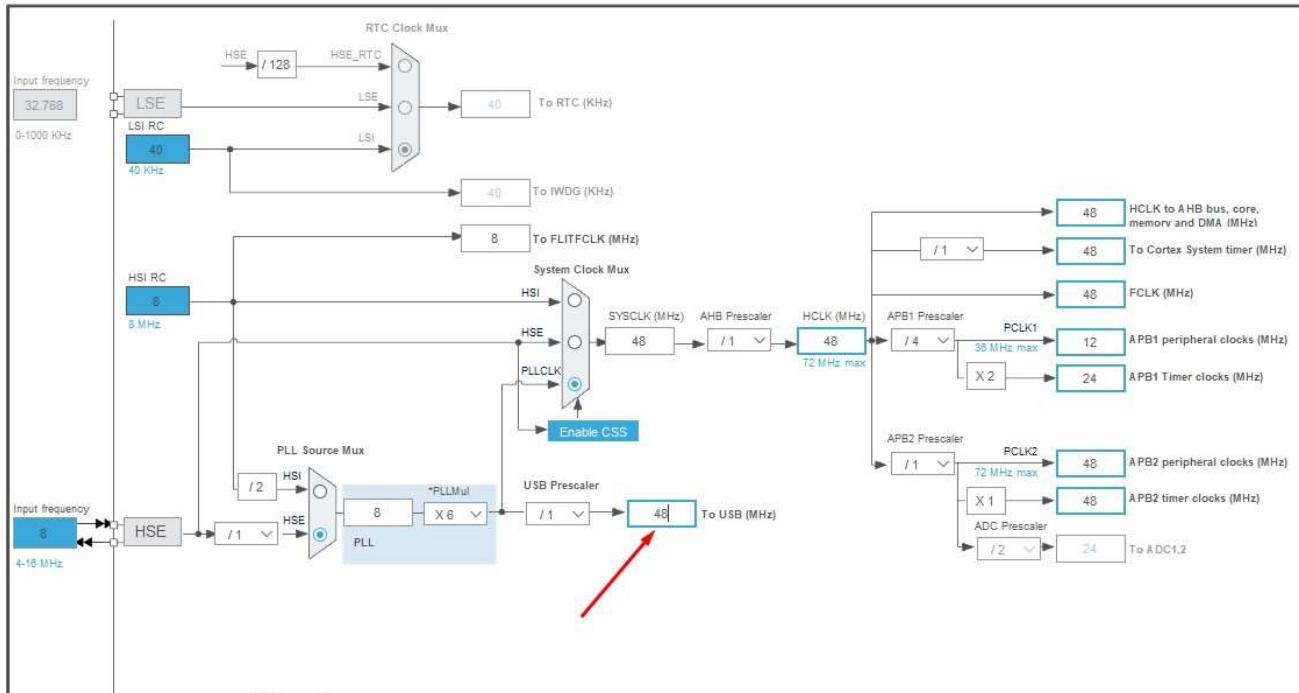
Trong Connectivity chọn USB và tick và Device (FS). Giao thức USB Device Full Speed.



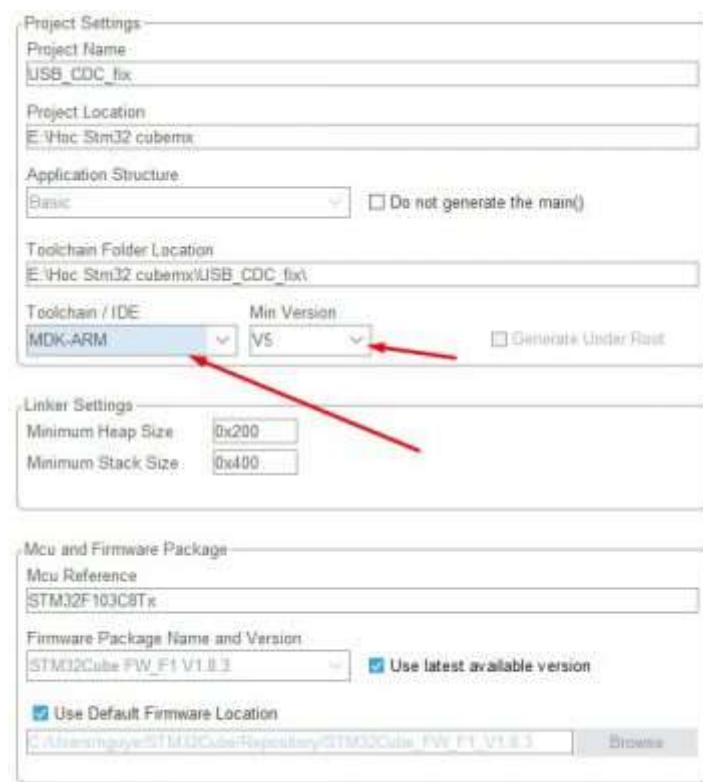
Phần Middleware chọn USB_DEVICE chọn Clas CDC hay cổng com ảo. Chính sửa kích thước Buffer truyền và nhận. Ở đây mình sử dụng 64bytes.



Chuyển qua Tab Clock Config -> Chọn tần số cho bộ USB là 48Mhz (Bắt buộc), tần số của mạch có thể chọn cao hơn, ở đây mình chọn 48Mhz.



Đặt tên, chọn IDE sau đó Gen code.



Lập trình STM32 Usb CDC với Keil C

Mở Project vừa tạo, chọn File usbd_cdc_if.c. Code thêm vào phần CDC_Receive_FS như hình.

```

Project: USB_CDC_fix
  -> USB_CDC_fix
    -> Application/MDK-ARM
      -> startup_stm32f103xb.s
    -> Application/User
      -> main.c
      -> usbd_device.c
      -> usbd_conf.c
      -> usbd_desc.c
      -> usbd_cdc_if.c
      -> stm32f1xx_it.c
      -> stm32f1xx_hal_msp.c
    -> Drivers/STM32F1xx_HAL_Driver
      -> stm32f1xx_hal_gpio_ex.c
      -> stm32f1xx_hal_i2c.c
      -> stm32f1xx_hal_lcd.c
      -> stm32f1xx_hal_pcd.c
      -> stm32f1xx_hal_rcc.c
      -> stm32f1xx_hal_rcc_ex.c
      -> stm32f1xx_hal_gpioc.c
      -> stm32f1xx_hal_dmac.c
      -> stm32f1xx_hal_cortex.c
      -> stm32f1xx_hal_pwr.c
      -> stm32f1xx_hal_flash.c
      -> stm32f1xx_hal_flash_ex.c

main.c usbd_cdc_if.c usbd_conf.c usb_device.c usbd_desc.c usbd_def.h startup_stm32f103xb.s stm32f1xx_hal.c

285     *           in receiving more data while previous ones are still not sent.
286
287     * @param   Buf: Buffer of data to be received
288     * @param   Len: Number of data received (in bytes)
289     * @retval Result of the operation: USBD_OK if all operations are OK else USBD_FAIL
290 */
291 static int8_t CDC_Receive_FS(uint8_t* Buf, uint32_t *Len)
292 {
293     /* USER CODE BEGIN 6 */
294     USBD_CDC_SetRxBuffer(&hUsbDeviceFS, &Buf[0]);
295     USBD_CDC_ReceivePacket(&hUsbDeviceFS);
296     CDC_Transmit_FS(UserRxBufferFS,strlen((const char*)UserRxBufferFS)-1); // truyen du lieu vua nhan duoc
297     for(int i = 0; i<APP_RX_DATA_SIZE; i++)
298     {
299         UserRxBufferFS[i] = 0; // xoa buffer nhan
300     }
301     return (USBD_OK);
302     /* USER CODE END 6 */
303 }
304
305 /**
306     * @brief   CDC_Transmit_FS
307     *          Data to send over USB IN endpoint are sent over CDC interface
308     *          through this function.
309     * @note
310 */

```

Giải thích code:

Khi có một gói tin truyền đến, MCU sẽ gọi hàm CDC_Receive_FS để đọc dữ liệu nhận được bằng lệnh USBD_CDC_ReceivePacket(&hUsbDeviceFS);

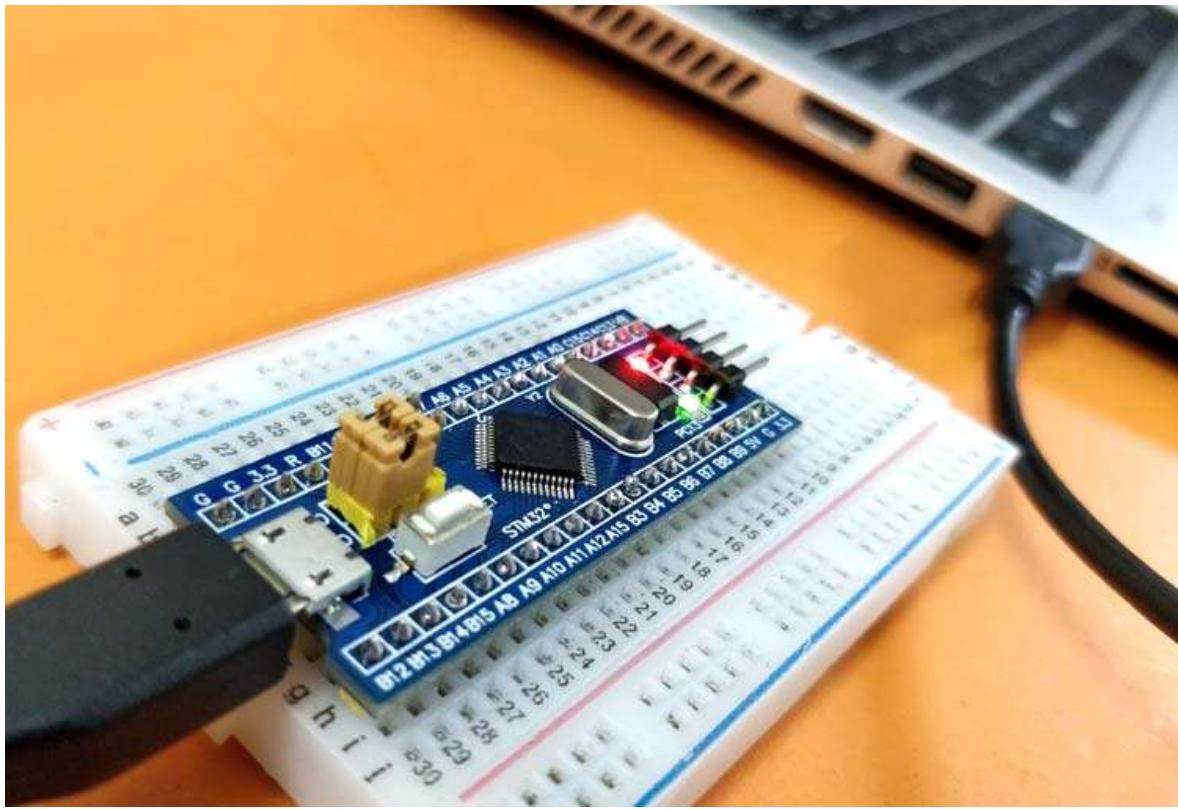
Sau khi đọc xong gói tin, dữ liệu sẽ được lưu tại biến: UserRxBufferFS

Sau đó chúng ta thực hiện lệnh gửi dữ liệu với độ dài lấy từ hàm **strlen((const char*)UserRxBufferFS)** – 1. Tại sao lại – 1 vì khi nhận được 1 chuỗi bất kì, cuối gói tin nhận được sẽ có 1 byte kiểm tra dữ liệu, vậy nên nếu gửi tất cả sẽ thừa 1 byte.

Sau khi gửi xong, chúng ta xóa UserRxBufferFS; để chuẩn bị nhận giá trị tiếp theo.

Cài đặt Driver STM32 Usb CDC

Khi nạp xong cho Bluepill, chúng ta cắm dây micro usb vào kit và kết nối với máy tính. Khi đó máy tính sẽ báo rằng chưa có driver cho thiết bị

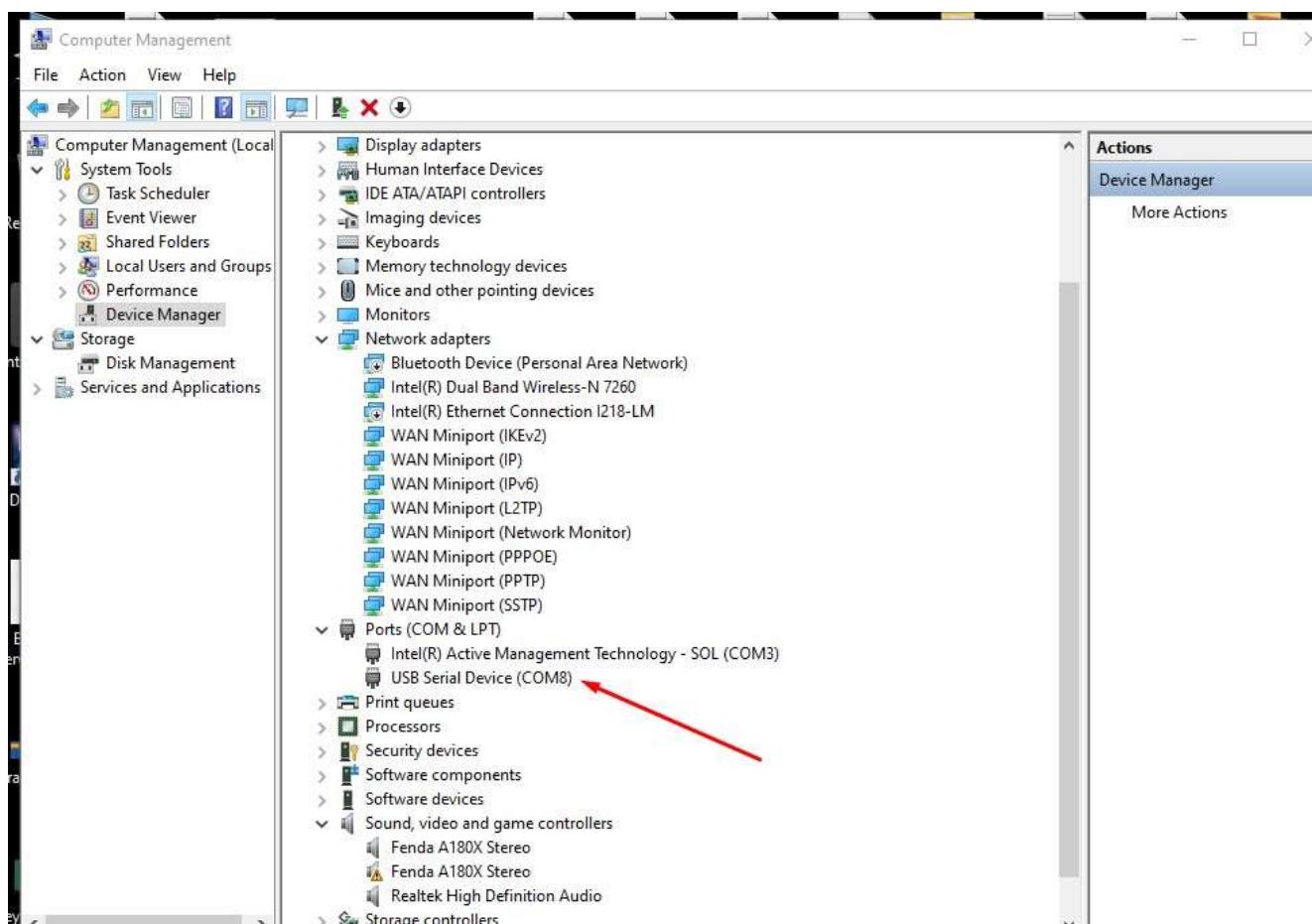


Để sử dụng được Virtual Com Port chúng ta cần cài Driver để máy tính biết cần giao tiếp với cái gì. Cũng giống như khi bạn cài driver cho USB to COM sử dụng PL2303 hoặc CH340 vậy.

Nhấn vào link để down về: <https://www.st.com/en/development-tools/stsw-stm32102.html>

Product overview

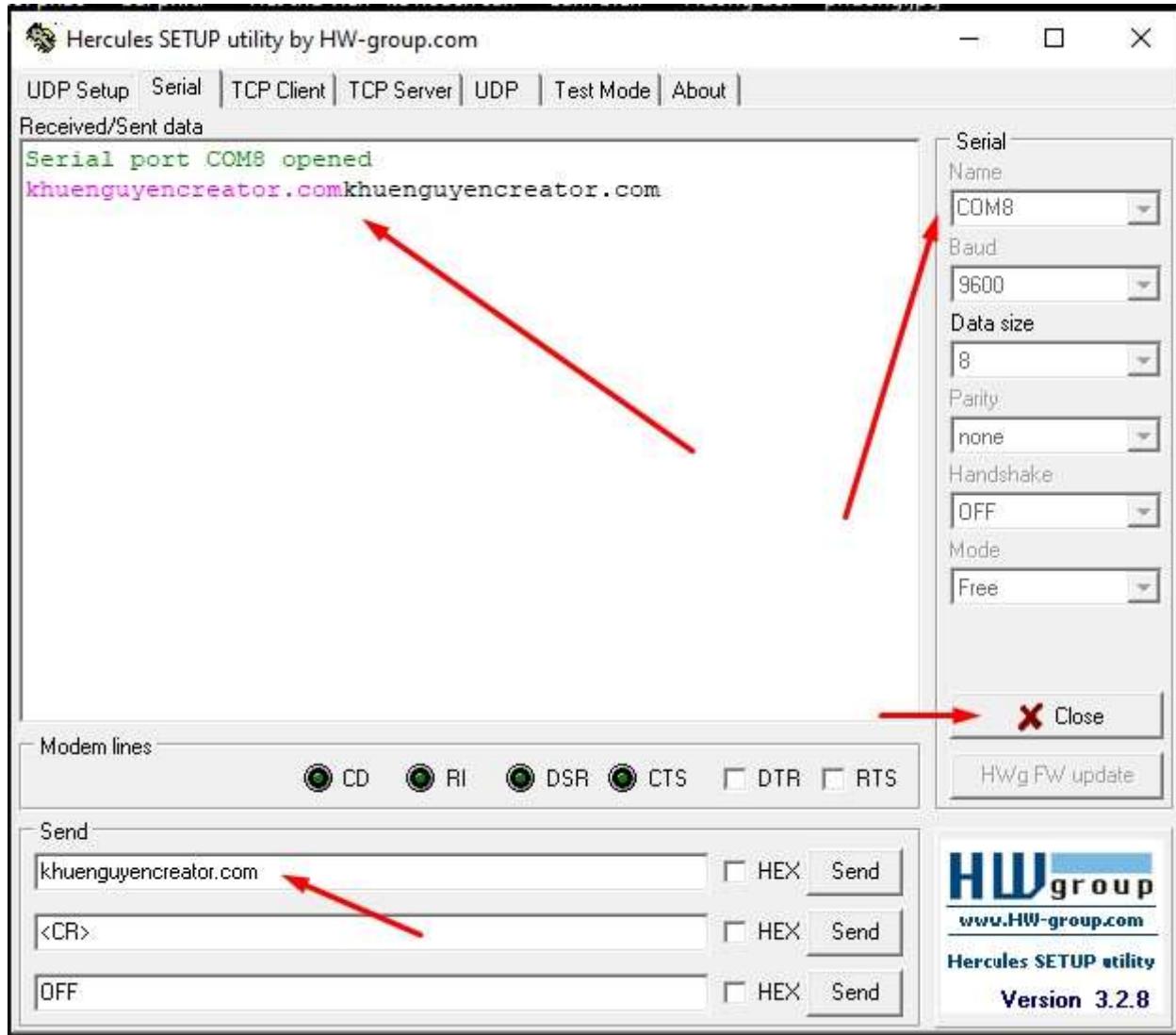
Download bản phù hợp với máy tính của bạn, rồi cài đặt, sau khi cài xong, nhấn vào manager xem đã ok chưa. Hiển thị giống hình là đã cài đúng



Gửi dữ liệu lên STM32 USB CDC với Hercules Terminal

Sau khi đã cài đặt xong, chúng ta mở phần mềm Hercules Terminal, chọn cổng COM tương ứng. Nếu bạn chưa có phần mềm có thể tham khảo bài viết: [Hướng](#)

dẫn sử dụng Hercules Terminal



Gửi dữ liệu bất kì lên, mạch sẽ phản hồi lại đúng dữ liệu đó.

Kết

STM32 USB CDC được sử dụng rất nhiều trong việc truyền nhận giữa các thiết bị nhúng với nhau hoặc giữa chúng với máy tính, vì tính tiện dụng và hiệu quả và tốc độ nó mang lại.

Nếu bạn thấy bài viết này có ích hãy chia sẻ với bạn bè mình và cùng vào nhóm những Anh Em **Nghiện Lập trình** nhé !!!

5/5 - (1 bình chọn)

Related Posts:

1. [Lập trình STM32 điều khiển LCD1602 chế độ 8bit và 4bit](#)
2. [Bản đồ bộ nhớ \(Memory map\) vi điều khiển STM32F103](#)
3. [Bài 18: Lập trình STM32 USB HID Keyboard bàn phím máy tính](#)
4. [Bài 17: Lập trình STM32 USB HID chuột máy tính](#)
5. [Bài 13: Lập trình STM32 RTC – Real Time Clock](#)
6. [Lập trình STM32 từ A tới Z](#)



KHUÊ NGUYỄN

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

ONE THOUGHT ON “BÀI 16: LẬP TRÌNH STM32 USB CDC TRUYỀN NHẬN DỮ LIỆU QUA CỔNG COM ẢO”

Vu Duy says:

anh cho em hỏi với. Giờ em đang dùng con camera OV 7670 thu hình ảnh về rồi qua STM32 rồi từ STM32F103c8 qua cổng USB cho về máy tính liệu có được không ạ

04/04/2021 AT 4:18 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

Lý do thời gian gần đây mình không viết bài và làm thêm gì cả là đây 😊)

Chỉn chu thực ra mai san pham uinn vi truong
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

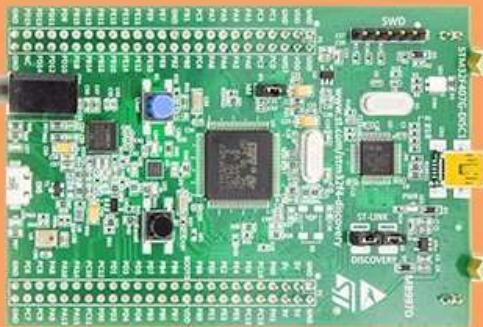
8051 là một dòng chip nhập môn cho lập trình viên nhúng, chúng được sử...

[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

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

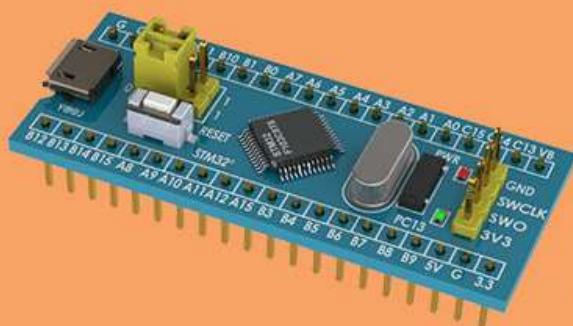
Trong bài này chúng ta sẽ cùng nhau tìm hiểu STM32 DAC với KIT STM32F407VE...

[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](#)



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: Nghịên Lập Trình

Fanpage: Khuê Nguyên Creator

My Shop

Thông Tin

Tác Giả

Chính Sách Bảo Mật



Copyright 2022 © Khuê Nguyễn