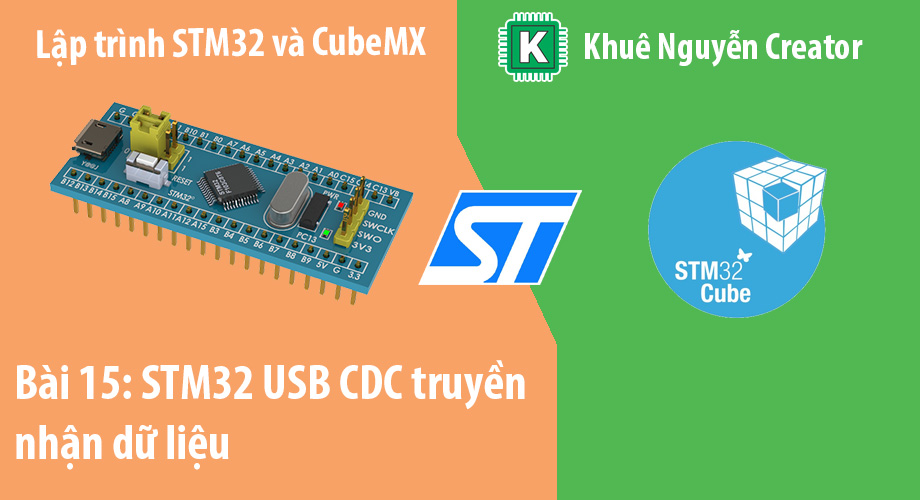
**Bài 16: Lập trình STM32 USB CDC truyền nhận dữ liệu qua cổng COM ảo**

POSTED ON [09/03/2021](https://khuenguyencreator.com/lap-trinh-stm32-usb-cdc-truyen-nhan-du-lieu-qua-cong-com-ao/) BY [KHUÊ NGUYỄN](https://khuenguyencreator.com/author/nguyenkhue2608/)

[](https://khuenguyencreator.com/lap-trinh-stm32-usb-cdc-truyen-nhan-du-lieu-qua-cong-com-ao/)

**09  
Th3**

Giao thức USB nói chung và STM32 USB CDC nói riêng là một giao thức khá khó lập trình. Nhưng ST đã hộ 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 midware đó.

Bài 16 trong Serie [Học lập trình STM32 từ A tới Z](https://khuenguyencreator.com/lap-trinh-stm32-tu-a-toi-z/)

Mục Lục

[](https://khuenguyencreator.com/lap-trinh-stm32-usb-cdc-truyen-nhan-du-lieu-qua-cong-com-ao/)

* [Giao thức usb là gì?](https://khuenguyencreator.com/lap-trinh-stm32-usb-cdc-truyen-nhan-du-lieu-qua-cong-com-ao/#Giao_thuc_usb_la_gi)
* [Một số kiến thức cơ bản về giao thức usb](https://khuenguyencreator.com/lap-trinh-stm32-usb-cdc-truyen-nhan-du-lieu-qua-cong-com-ao/#Mot_so_kien_thuc_co_ban_ve_giao_thuc_usb)
  + [Kiến trúc hệ thống USB](https://khuenguyencreator.com/lap-trinh-stm32-usb-cdc-truyen-nhan-du-lieu-qua-cong-com-ao/#Kien_truc_he_thong_USB)
  + [Cấu trúc điều khiển truyền nhận](https://khuenguyencreator.com/lap-trinh-stm32-usb-cdc-truyen-nhan-du-lieu-qua-cong-com-ao/#Cau_truc_dieu_khien_truyen_nhan)
  + [Các kiểu truyền nhận trong USB](https://khuenguyencreator.com/lap-trinh-stm32-usb-cdc-truyen-nhan-du-lieu-qua-cong-com-ao/#Cac_kieu_truyen_nhan_trong_USB)
  + [Tài liệu về giao thức USB](https://khuenguyencreator.com/lap-trinh-stm32-usb-cdc-truyen-nhan-du-lieu-qua-cong-com-ao/#Tai_lieu_ve_giao_thuc_USB)
* [Giao thức Usb cdc trên stm32](https://khuenguyencreator.com/lap-trinh-stm32-usb-cdc-truyen-nhan-du-lieu-qua-cong-com-ao/#Giao_thuc_Usb_cdc_tren_stm32)
  + [USB CDC là gì?](https://khuenguyencreator.com/lap-trinh-stm32-usb-cdc-truyen-nhan-du-lieu-qua-cong-com-ao/#USB_CDC_la_gi)
  + [STM32 USB CDC](https://khuenguyencreator.com/lap-trinh-stm32-usb-cdc-truyen-nhan-du-lieu-qua-cong-com-ao/#STM32_USB_CDC)
* [Lập trình stm32 usb cdc](https://khuenguyencreator.com/lap-trinh-stm32-usb-cdc-truyen-nhan-du-lieu-qua-cong-com-ao/#Lap_trinh_stm32_usb_cdc)
  + [Thiết lập STM32 Usb CDC trên Cube MX](https://khuenguyencreator.com/lap-trinh-stm32-usb-cdc-truyen-nhan-du-lieu-qua-cong-com-ao/#Thiet_lap_STM32_Usb_CDC_tren_Cube_MX)
  + [Lập trình STM32 Usb CDC với Keil C](https://khuenguyencreator.com/lap-trinh-stm32-usb-cdc-truyen-nhan-du-lieu-qua-cong-com-ao/#Lap_trinh_STM32_Usb_CDC_voi_Keil_C)
  + [Cài đặt Driver STM32 Usb CDC](https://khuenguyencreator.com/lap-trinh-stm32-usb-cdc-truyen-nhan-du-lieu-qua-cong-com-ao/#Cai_dat_Driver_STM32_Usb_CDC)
  + [Gửi dữ liệu lên STM32 USB CDC với Hercules Terminal](https://khuenguyencreator.com/lap-trinh-stm32-usb-cdc-truyen-nhan-du-lieu-qua-cong-com-ao/#Gui_du_lieu_len_STM32_USB_CDC_voi_Hercules_Terminal)
* [Kết](https://khuenguyencreator.com/lap-trinh-stm32-usb-cdc-truyen-nhan-du-lieu-qua-cong-com-ao/#Ket)
  + [Related posts:](https://khuenguyencreator.com/lap-trinh-stm32-usb-cdc-truyen-nhan-du-lieu-qua-cong-com-ao/#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 tô chức USB-IF. Theo wikipedia

Đã có bốn thế hệ USB, gồm: [USB 1.x](https://vi.wikipedia.org/w/index.php?title=USB_1.x&action=edit&redlink=1), [USB 2.0](https://vi.wikipedia.org/wiki/USB_2.0), [USB 3.x](https://vi.wikipedia.org/w/index.php?title=USB_3.x&action=edit&redlink=1) và [USB4](https://vi.wikipedia.org/w/index.php?title=USB4&action=edit&redlink=1)

[Giao thức USB rất thông dụng](https://giphy.com/gifs/hackernoon-hacker-noon-W632WXHvRRSj8UoFit)

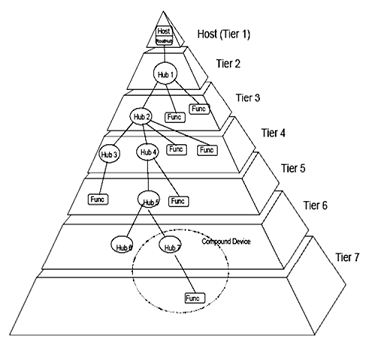
**Một số kiến thức cơ bản về giao thức usb**

**Kiến trúc hệ thống USB**

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

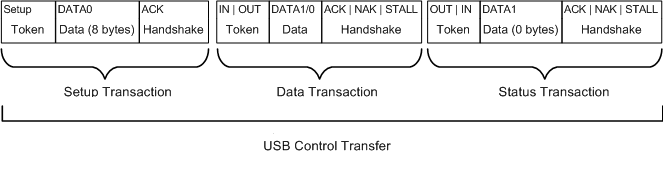
* USB Host: Chỉ có 1 Host duy nhất trong hệ thống ( thông thường sẽ là máy tính cá nhân hoặc máy tính nhúng). Chúng điều khiển mọi luồng dữ liệu và cách thức giao tiếp trong hệ thống.
* USB Device: Trong một hệ thống có thể có nhiều device chúng có thể đóng nhiều vai trò như: Thiết bị đầu cuối tương 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=1wEQKyJVl3d0I6MbVtHehQ1Ciqf878DOC>

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.

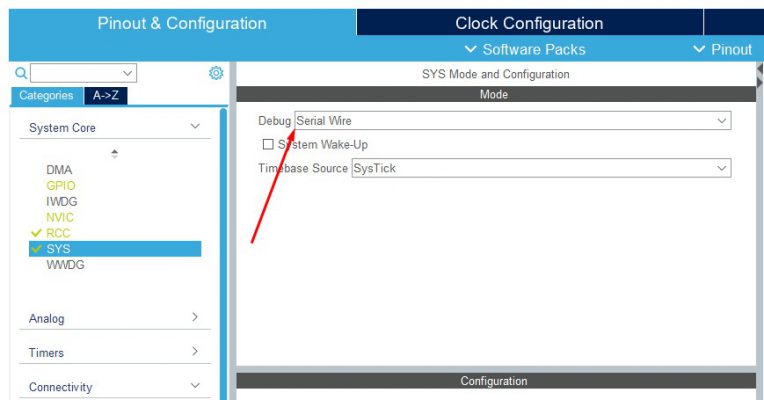
**STM32 USB CDC**

Trong STM32f103c8t6 chỉ hỗ trợ giao thưc 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.

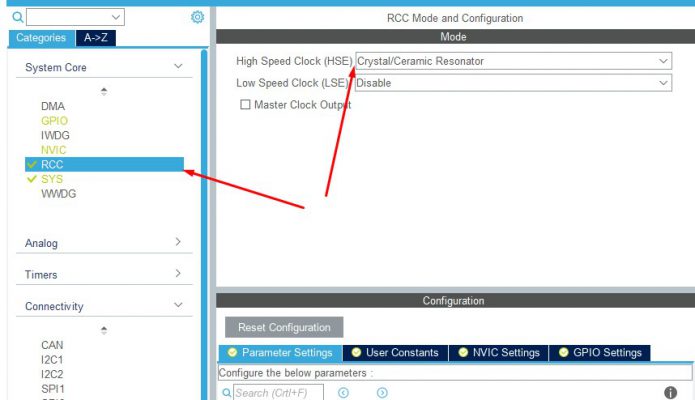
**Lập trình stm32 usb cdc**

**Thiết lập STM32 Usb CDC trên Cube MX**

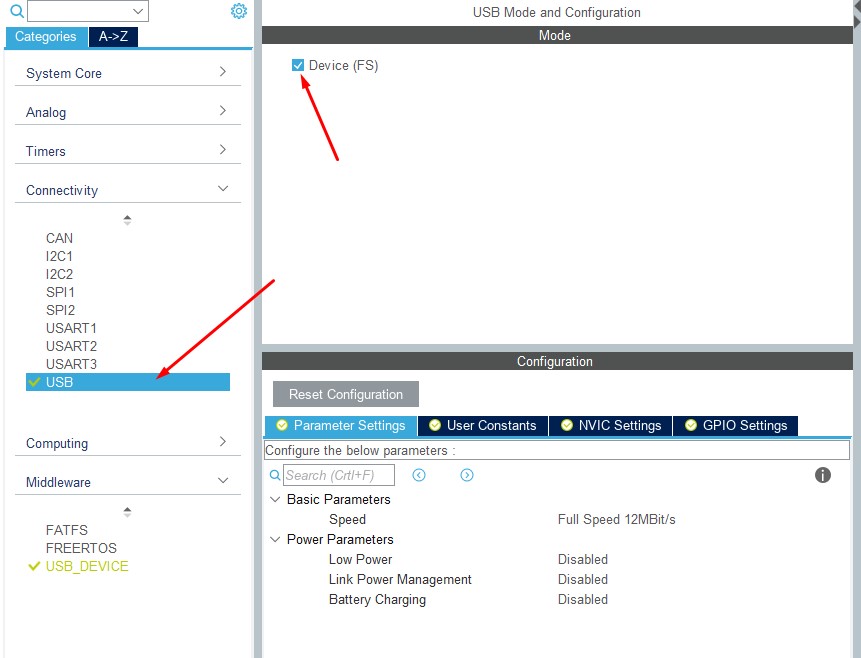
Mở Cubemx lên, trong tab SYS chọn debug Serial Wire



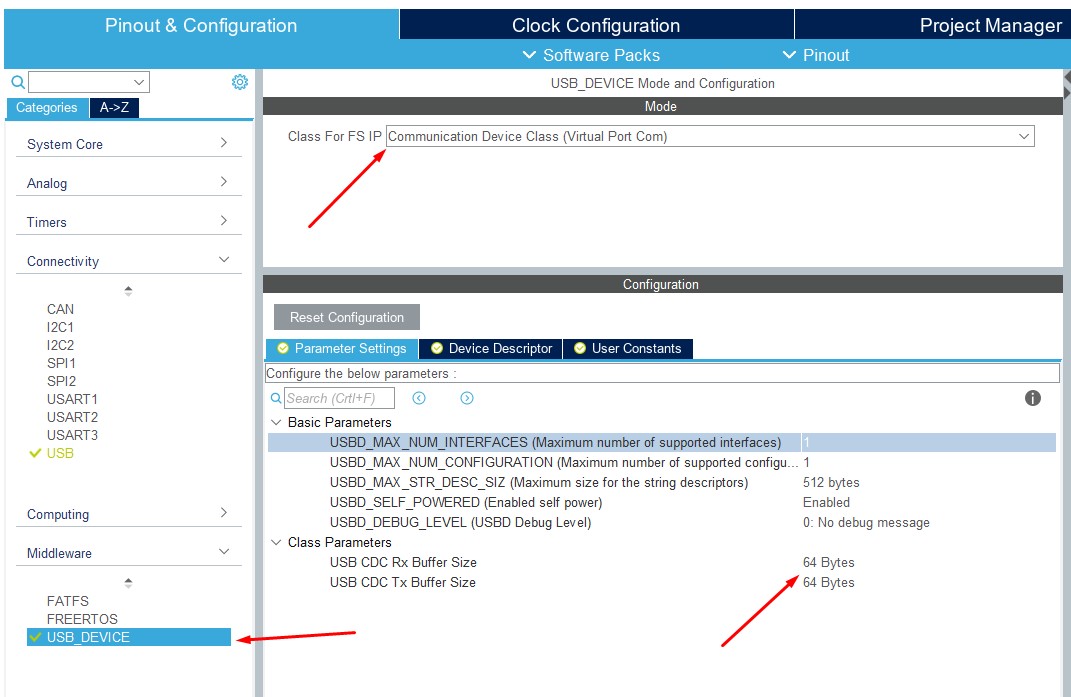
Tab RCC chúng ta chọn nguồn dao động là thạch anh ngoại HSE.



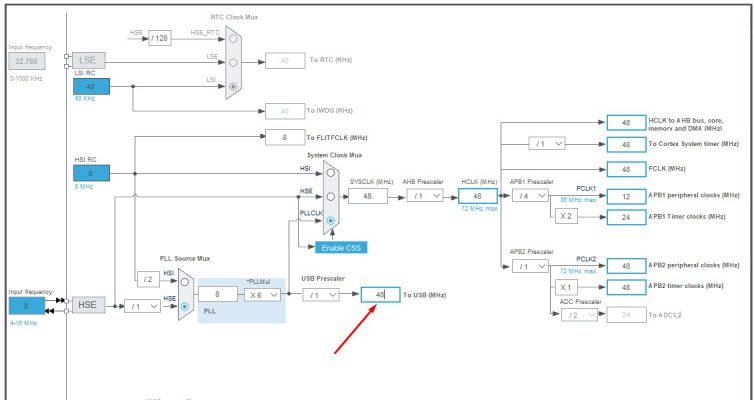
Trong Conectivity chọn USB và tick và Device (FS). Giao thức USB Device Full Speed.



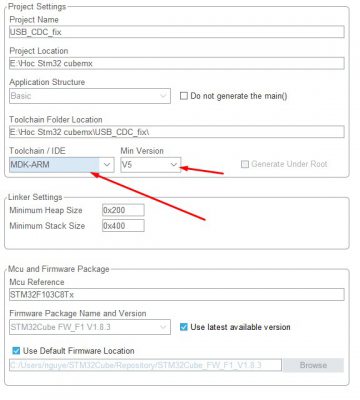
Phần Middleware chọn USB\_DEVICE chọn Clas CDC hay cổng com ảo. Chỉnh sửa kích thước Buffer truyền và nhận. Ở đây mình sử dụng 64bytes.



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.

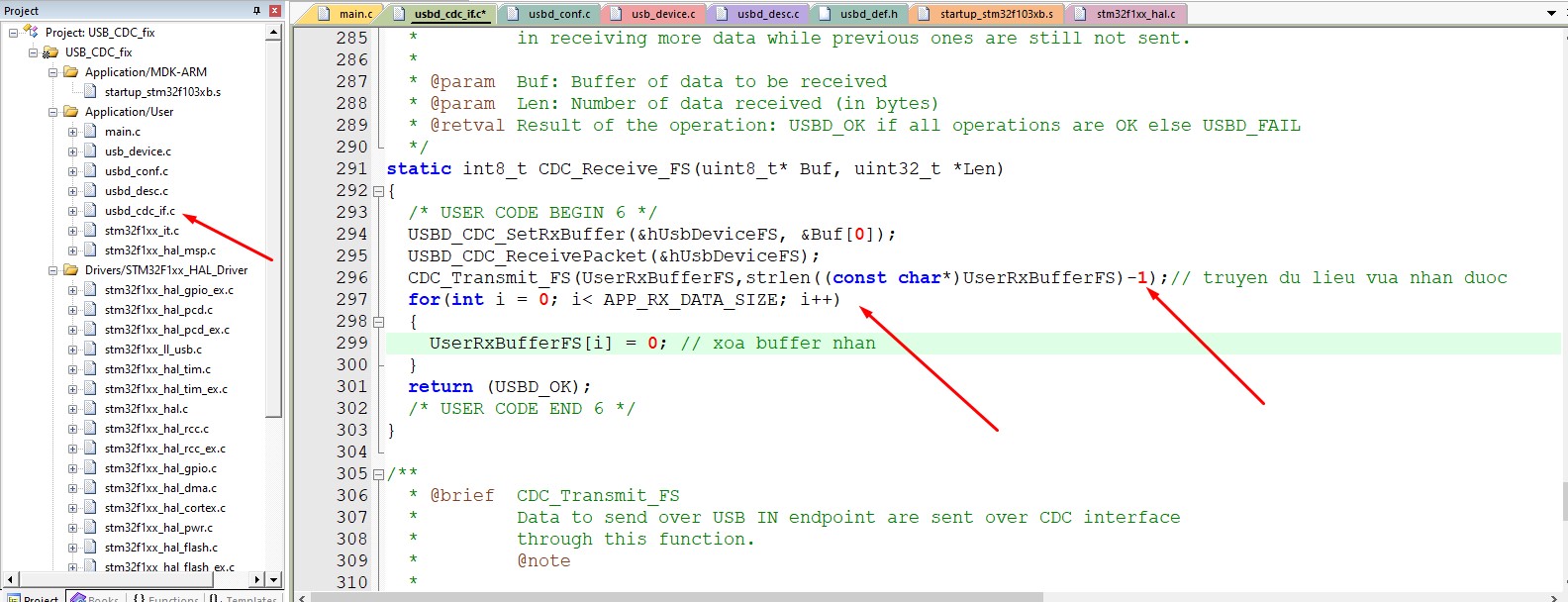


Đặt tên, chọn IDE sau đó Gen code.



**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.



Giải thích code:

Khi có một gói tin truyền đến, MCU sẽ gọi hàm CDC\_Recive\_FS để đọc dữ liệu nhận được bằng lệnh USBD\_CDC\_RecivePacket(&hUsbDeviceFS);

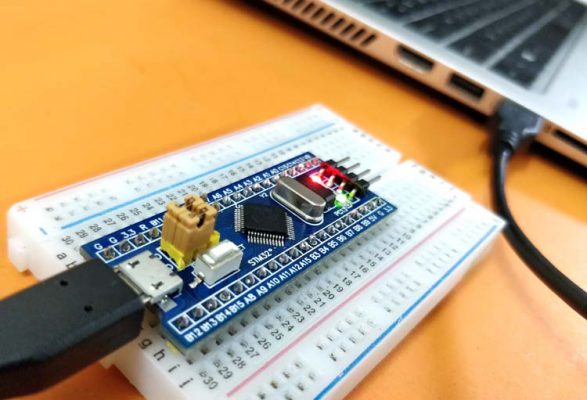
Sau khi đọc xong gói tin, dữ liệu sẽ được lưu tại biến: UserRxBufferFS

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 đữ 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.

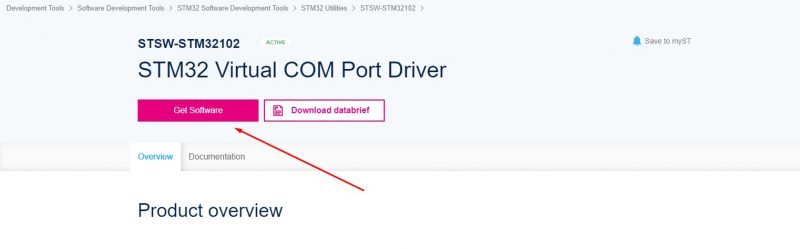
**Cài đặt Driver STM32 Usb CDC**

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

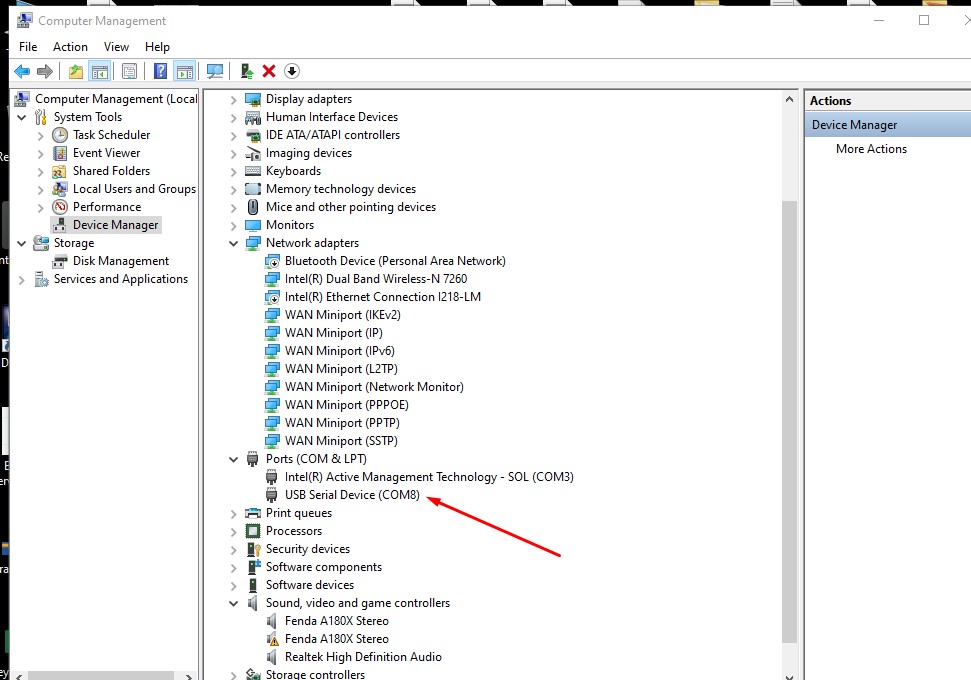


Để sử dụng được Vitual Com Port chúng ta cần cài Driver để máy tính biết cần giao tiếp với cái gì. Cũng giống như khi bạn cài driver cho USB to COM sử dụng PL2303 hoặc CH340 vậy.

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

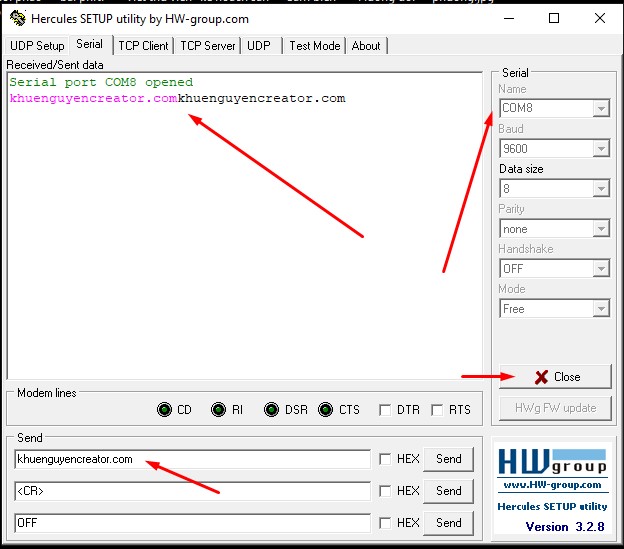


Download bản phù hợp với máy tính của bạn, rồi cài đặt, sau khi cài xong, nhấn vào manager xem đã ok chưa. Hiển thị giống hình là đã cài đúng



**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](https://khuenguyencreator.com/huong-dan-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.