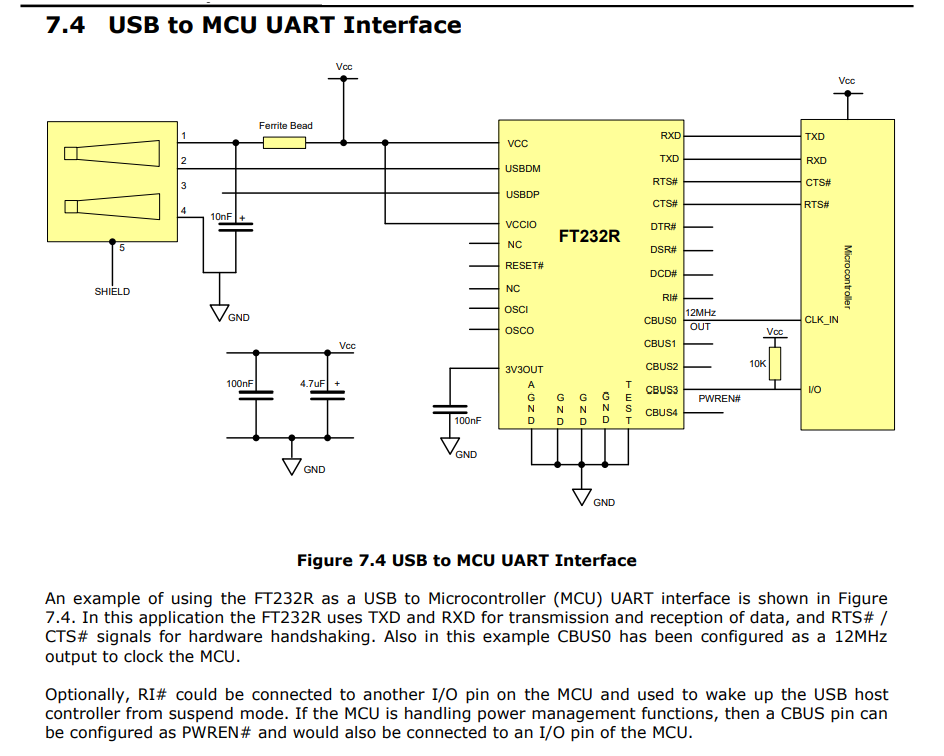
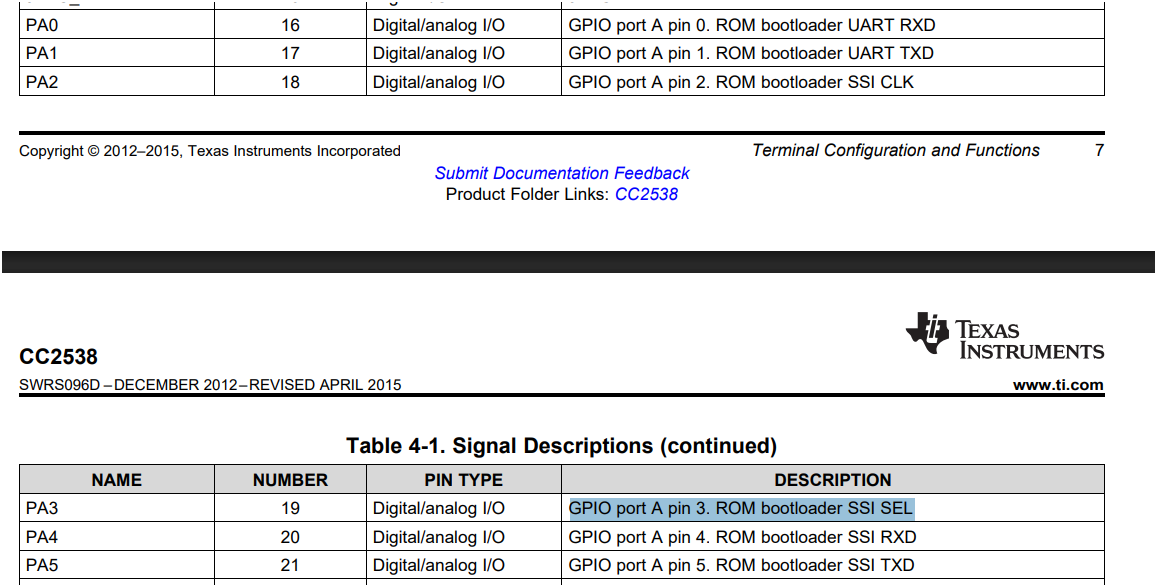
**Tại sao lại kết nối CTS và RTS để nạp code?**

Bởi vì: (mạch mẫu, CTS và RTS) Đây chỉ là giao tiếp USB qua UART chứ ko phải là interface để nạp code, bởi vì interface để nạp code cần có thêm chân gpio để đưa vào chế độ Auto programer. mà chân select lại là chân PA3 không phải chân mặc định để vào bootloader (PA7).

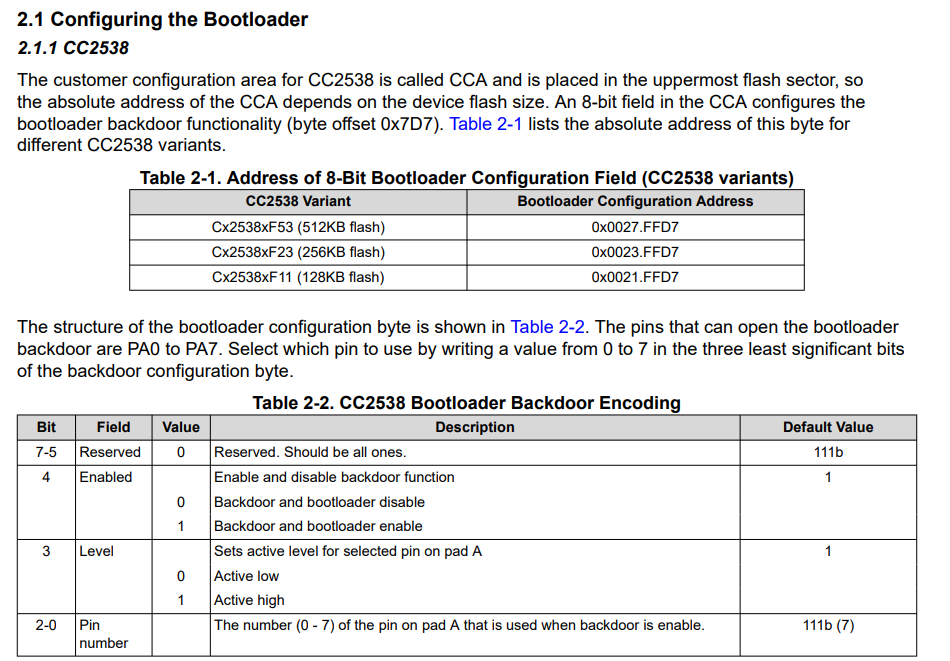
Trong trường hợp này chân PA3 kéo ra header làm header SPI bình thường.

Với lại chân PA3 lại là chân chip select trong chế độ SPI (tại sao lại làm công tắc kéo xuống đất).

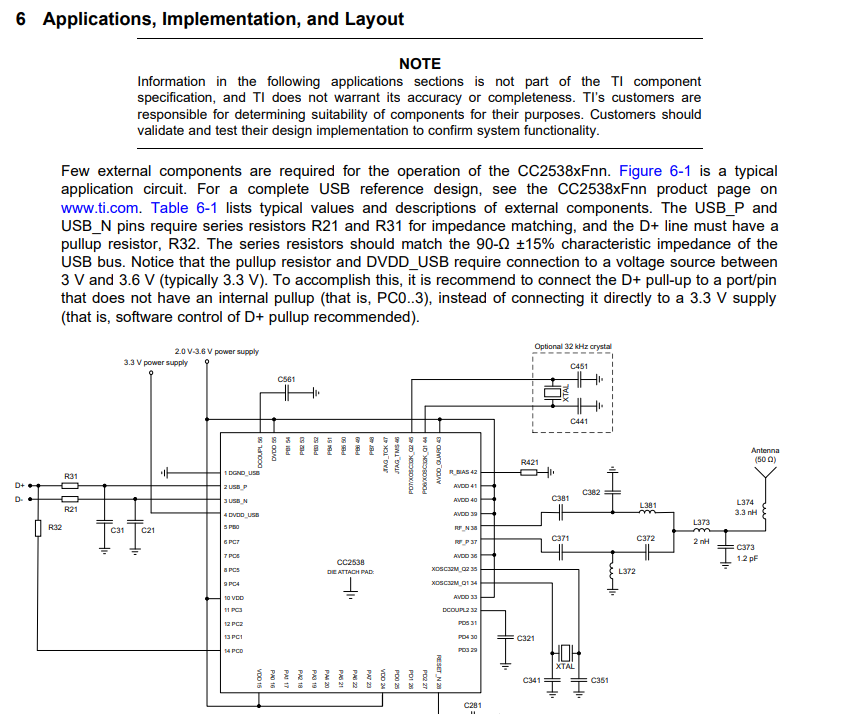




**USB D+ và USB D- ko có Pull up nối vào PC0, Host sẽ ko nhận biết dc tín hiệu usb highspeed 2.0 ?? – Trả lời: ko cần thiết..**

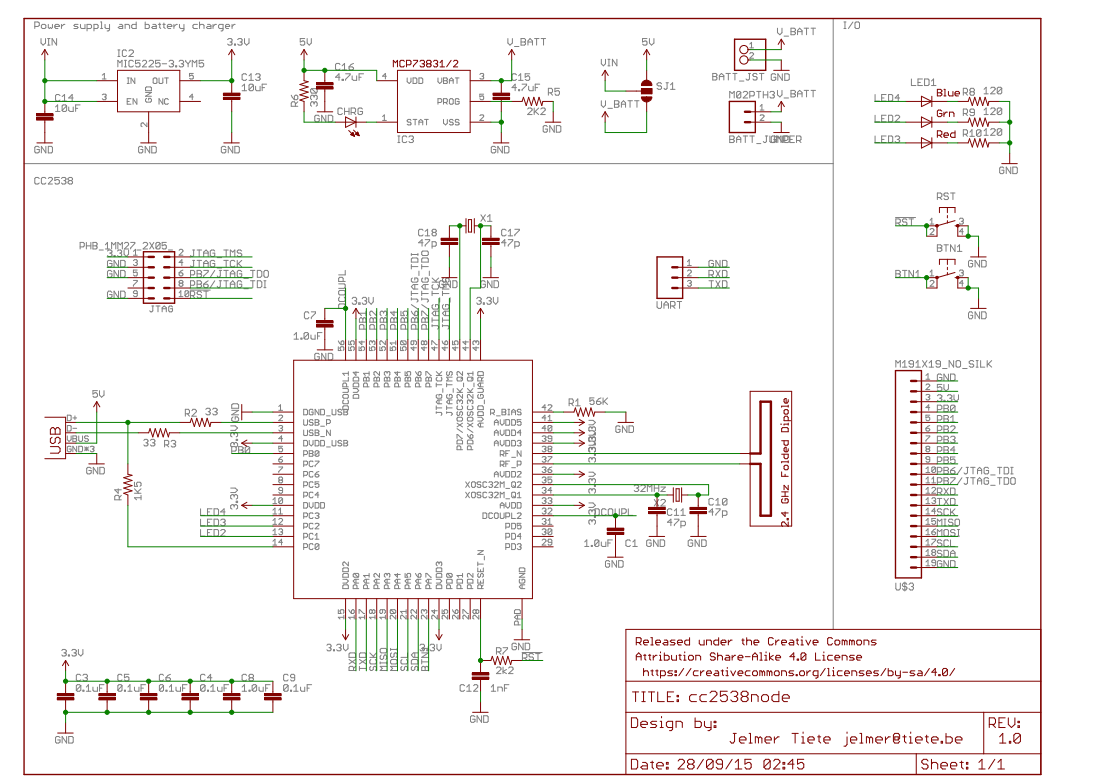




Nhìn trên hình thì hai chân usb ko có pull up D+ và ko có điện trở phối hợp trở kháng  


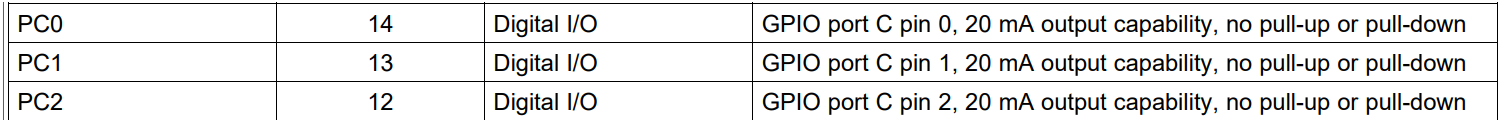
**Câu hỏi 3: Tại sao chân PA7 trong datasheet mặc định là active high =1 để vào bootloader mà tại sao trong file tham khảo họ lại làm nút nhấn kéo xuống đất.**

**Trả lời: Tùy chip CC2538 khi xuất xưởng đc sản xuất sẽ có mặc định chân boot riêng**

****

**Câu hỏi 4: Tham khảo trên mạng thấy nta chỉ cách nạp firmware sử dụng raspberry pi để xóa CCA flash, set flag xong rồi mới nạp qua uart sử dụng tool CC2538 BSL.Python để nạp. Liệu lúc mới mua thì CC2538 có nạp qua dc usb sử dụng ft232Rl đc ko. vì có khả năng con chip mới xuất xưởng chưa đc mở bootloader.**

Giải đáp thắc mắc chính bản thân:



Tại sao 3 chân nta ko sử dụng hoặc có sẽ dụng sẽ nối vào LED?

Trong module CC2538 có thêm CC2592 thì 3 chân này kết nối với CC2592

Giải thích: Trong ứng dụng IOT thì 3 chân này ko bị chiếm dụng bởi các ngoại vi và do ko có pull up hay pull down, có dòng ngõ ra cao nên có thể điều khiển led relay và ngoại vi… Mục đích của các chân này.

+ Mục đích chính: Dành riêng cho điều khiển thiết bị ngoài

+ Tránh can thiệp và mâu thuẫn với các ngoại vi khác

+ Test led ngay trên board

+ Ko có pull up/ pull down :

**❌ Vấn đề:**

* Nếu chân PC0 có **pull-up nội mặc định**:
  + Ngay khi MCU vừa cấp nguồn, **chưa kịp cấu hình GPIO**, pull-up nội đã kéo chân PC0 lên mức HIGH.
  + → **Relay lập tức bật**, bơm nước chạy ngay **trước khi MCU xử lý gì cả**.

**❌ Vấn đề:**

* Nếu chân PC0 có **pull-up nội mặc định**:
  + Ngay khi MCU vừa cấp nguồn, **chưa kịp cấu hình GPIO**, pull-up nội đã kéo chân PC0 lên mức HIGH.
  + → **Relay lập tức bật**, bơm nước chạy ngay **trước khi MCU xử lý gì cả**.

Một số module chỉ có DTR nma ko có RTS hoặc hoặc có CTS nma ko có RTS.

Giải thích: chỉ có DTR mà ko có RTS ( bán tự động, phải reset bằng tay); chỉ có CTS: đơn giản là chỉ kiểm tra trạng thái thiết bị một chiều.

**PHẦN NẠP CODE CHO CC2538**

Có hai cách nạp:

C1: **Firmware flashing on TI CC2538 using OpenOCD and UART**

Link hướng dẫn: <https://duncankn.medium.com/firmware-flashing-on-ti-cc2538-using-openocd-and-uart-cbceb1e2a82e>

Tóm tắt: Cách này là dùng raspberry Pi để flashing

1. Sử dụng JTAG để xóa vùng nhớ đặc biệt CCA, vùng nhớ chưa byte cấu hình bootloader backdoor để đưa vào chế độ BSL

2. Tiếp theo sử dụng UART để nạp firmware kết hợp 2 button reset và boot để nạp.

Bài viết này cần một số giải đáp: Chung quy là đều mục đích là nạp code qua UART dùng BSL, nhưng có thêm bước Open OCD để xóa flash và set flag trong vùng nhớ CCA để đưa vào chế độ bootloader sau khi reset.

Nghi vấn 1: Có thể họ đang muốn truy cập trực tiếp CCA để xóa flash CCA và set flag theo mục đích của họ.

Nghi vấn 2: Họ ko nạp code qua ft232Rl mà nạp code qua raspberry pi, ám chỉ là họ muốn đơn giản ko gắn thêm ft232rl cho cồng kềnh, hai là khi mcu mới xuất xưởng thì bootloader chưa dc bật nên phải cấu hình lại qua JTAG.

