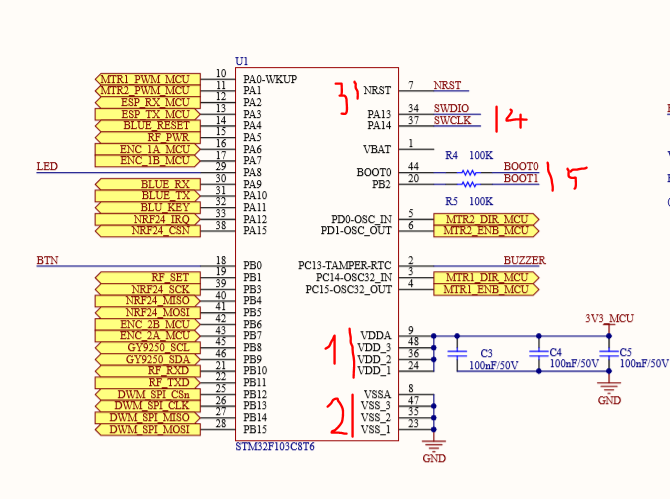
# Project xe cân bằng

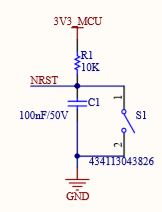
1. **Hardware của xe cân bằng**
2. Chip LQFP-48 của STM32F103C8T6: nó là chip, điều khiển mọi hành động của xe.



1.1: Các chân nguồn cấp cho chip hoạt động (từ 2v đến 3v6 nếu ko sử dụng ADC,còn sử dụng ADC thì VDD cần trong khoảng từ 2v4 đến 3v6).

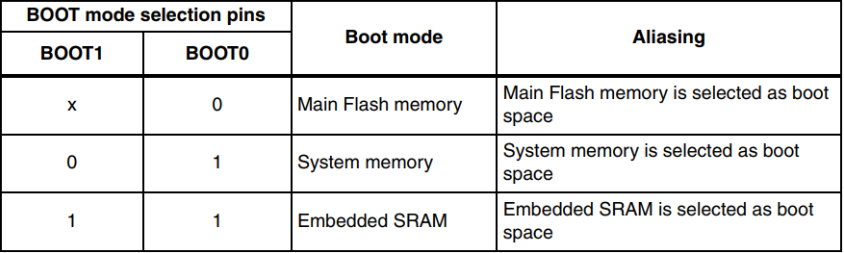
1.2: Các chân nối đất của chip

1.3: Chân reset của chip: Chân này được thiết kế như hình dưới đây. Khi ta nhấn nút S1, chân NRST sẽ được kéo xuống ở mức thấp, qua đó reset chip. Đây là 1 circuit rất phổ biến để reset 1 vdk.



1.4: Chuẩn SW của vi điều khiển

1.5: Các chân cấu hình chế độ BOOT của vdk.



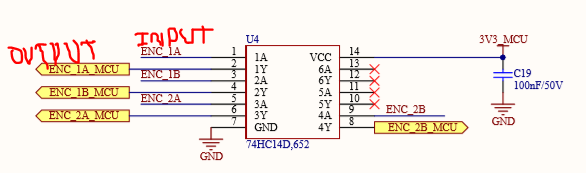
* **BOOT0 =0: boot từ Flash memory, hay thực thi chương trình do người dùng viết code nạp vào**
* BOOT0 = 1\_\_ BOOT1= 0: chạy ct từ bộ nhớ của hệ thống: ct bootloader của hãng ST
* BOOT0 = 1\_\_ BOOT1 = 1: chạy ct từ RAM

Với thiết kế thông thường, BOOT0 sẽ được nối với 1 trở kéo xuống GND để chạy code của user đưa vào chip.

Read more: <http://laptrinharmst.blogspot.com/2018/04/nap-code-cho-stm32-bang-boot-loader.html>

1. Sensor MPU6050 (hoặc GY9250) và 74HC14D\_652

2.1: 74HC14D\_652: là 1 loại IC logic chuyển mạch trigger, tức là chỉ chuyển trạng thái khi điện áp vượt ngưỡng điện áp cho phép. Bên trong 74HC14D có 6 con opamp mắc sẵn theo kiểu Schmitt trigger để lọc nhiễu tín hiệu đầu vào. Ta chỉ cần cấp 1 tín hiệu đầu vào, và sẽ có được tín hiệu đầu ra đảo trạng thái được lọc nhiễu rất tốt. Ví dụ, input là 0V thì output là 5V và ngược lại. (nếu cấp 3.5V đầu vào để giả lập cho tín hiệu nhiễu thì đầu ra vẫn 5V tùy vào bạn tính toán hoặc theo hysteric mặc định của nó.)

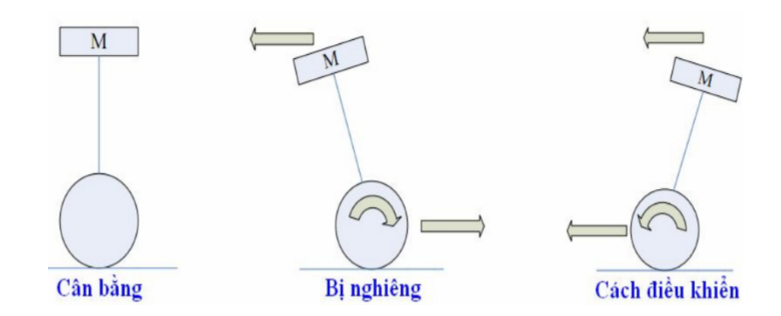


Ngõ ra của 74HC14D là tín hiệu của encoder, sẽ được đưa về vdk xử lí.

2.2: MPU6050

1. Driver điều khiển động cơ bước DRV8825 (2 cái), động cơ bước (2 cái), bánh xe (2 cái).
2. Module Bluetooth: dùng để giao tiếp UART giữa máy tính, điện thoại với robot.
3. **Lý thuyết điều khiển robot**
4. **Ý tưởng cân bằng robot**

Robot cân bằng thì góc nghiêng **θ** của thân robot và trục thẳng đứng bằng 0. Nếu như góc nghiêng **θ** khác 0, ví dụ như robot đang có xu hướng ngã về phía trước thì ta phải điều khiển cho 2 bánh xe cũng di chuyển về phía trước.



Vì vậy, ta cần phải thu thập được những thông tin sau từ trạng thái của robot: Setpoint (điểm làm cho robot cân bằng, ứng với góc **θ** bằng 0), hướng nghiêng, góc nghiêng và tốc độ nghiêng của robot. Tất cả thông tin này được lấy từ MPU6050 và được đưa vào PID để tạo ra tín hiệu điều khiển động cơ giữ robot thăng bằng.