VIỆN CNTT ITPLUS

---------o0o--------

ĐỒ ÁN

KẾT THÚC CUỐI KHÓA

NGÀNH LẬP TRÌNH

THIẾT KẾ XE 2 BÁNH TỰ CÂN BẰNG DÙNG STM32

LỚP EP0823EHCM -- Nhóm 2

Học Viên Lê Ngọc Hiếu

Ngô Anh Tuấn

Nguyễn Nam Hào

Nông Văn Toàn

GV Hướng dẫn Ngô Sinh Quyền

Tp.HCM,12/2023

LỜI NÓI ĐẦU

Đối với xe ba hay bốn bánh, việc cân bằng và ổn định của chúng là nhờ trọng tâm của chúng nằm trong bề mặt chân đế do các bánh xe tạo ra. Đối với các xe hai bánh có cấu trúc như xe đạp, việc thăng bằng khi không di chuyển là hoàn toàn không thể, vì việc thăng bằng của bánh xe dựa trên tính chất con quay hồi chuyển ở hai bánh xe khi đang quay. Còn đối với xe hai bánh tự cân bằng, là loại xe chỉ có hai bánh với trục của hai bánh xe trùng nhau, để cho xe cân bằng, trọng tâm của xe bao gồm cả người sử dụng chúng cần được giữ nằm ngay giữa các bánh xe. Điều này giống như ta giữ một cây gậy thẳng đứng cân bằng trong lòng bàn tay.

Đề tài gồm 5 chương:

- Chương 1: Mở đầu
- Chương 2: Tổng quan
- Chương 3: Nội dung và phương pháp nghiên cứu
- Chương 4: Kết quả và thảo luận
- Chương 5: Kết luận và đề nghị

Với kết quả đạt được của đề tài này chúng em hy vọng nó sẽ là một tài liệu tham khảo bổ ích cho các bạn muốn tìm hiểu. Trong thời gian ngắn thực hiện đề tài cộng với kiến thức còn nhiều hạn chế, nên không thể tránh được những thiếu sót, chúng em rất mong được sự góp ý của quý thầy (cô) và các bạn để đề tài hoàn thiện hơn.

LỜI CẢM ƠN

Trong suốt khóa học lập trình nhúng tại lớp EP0823EHCM, với sự giúp đỡ của quý Thầy Cô và giáo viên hướng dẫn về mọi mặt từ nhiều phía và nhất là trong thời gian thực hiện đề tài.

Nhóm xin chân thành cảm ơn đến quý Thầy trong bộ môn lập trình nhúng đã giảng dạy những kiến thức chuyên môn làm cơ sở để thực hiện tốt đồ án tốt nghiệp và đã tạo điều kiện thuận lợi cho chúng em thực hiện hoàn tất khóa học.

Đặc biệt, chúng em xin chân thành cảm ơn thầy Ngô Sinh Quyền đã tận tình hướng dẫn và cho chúng em những lời chỉ dạy quý báu, giúp chúng em định hướng tốt trong khi thực hiện đồ án.

Chúng tôi bày tỏ lòng biết ơn sâu sắc đến bạn bè, gia đình đã khích lệ động viên tôi trong quá trình học tập và nhiên cứu.

TP. HCM, ngày 20 tháng 12 năm 2023

Sinh viên thực hiện

Lê Ngọc Hiếu – Ngô Anh Tuấn- Nguyễn Nam Hào – Nông Văn Toàn

MỤC LỤC

DANH SÁCH CÁC CHỮ VIẾT TẮT

MATLAB: .....Matrix Laboratory

COM: .....Component Object Model

RS 232: .....Recommended Standard 232

DC: ..... Direct Current

MPU: .....Motion Processing Unit

CAN: .....Controller Area Network

Segway PT: ..... Segway Personal Transporter

LQR: ..... Linear Quadratic Regulator

DOF: ..... Degree of Freedom

CPU: .....Central Processing Unit

PID: .....Proportional Integral Derivative

PWM: .....Pulse Width Modulation

GA: ..... Genetic Algorithm

USB: ..... Universal Serial Bus

ROM: ..... Read Only Memory

UART: ..... Universal Asynchronous Receiver-Transmitter

RAM: .....Random Access Memory

SPI: ..... Serial Peripheral Interface

I2C: ..... Inter Integrated Circuit

I/O: .....Input/Output

EEPROM: ..... Electrically Erasable Programmable Read Only Memory

SRAM: .....Static Random Access Memory

PC: .....Personal Computer

LCD: .....Liquid Crystal Display

SD: ..... Secure Digital

MMC: ..... Multi Media Card

OLED: .....Organic Light Emitting Diode

IDE: .....Integrated Development Environment

DMP: .....Digital Motion Processor

ADC: ..... Analog to Digital Converter

TX: .....Transmitter

RX: ..... Receiver

CSR: .....Cambridge Silicon Radio

ENA: ..... Enable A

ENB: .....Enable B

GND: ..... Ground

VIN: ..... Voltage Input

SS: .....Slave Select

MISO: .....Master Input Slave Output

MOSI: .....Master Output Slave Input

SCK: ..... Serial Clock

LED: .....Light Emitting Diode

SDA: ..... Serial Data Line

SCL: ..... Serial Clock Line

DLPF: .... Digital Low Pass Filter

DANH SÁCH CÁC HÌNH

DANH SÁCH CÁC BẢNG

Chương 1:
MỞ ĐẦU

1.1 Lý do chọn đề tài

Hiện nay, cuộc sống con người đang ngày càng phát triển kèm theo đó là tăng lên về số lượng của các siêu thị, sân bay, khu du lịch. Xe hai bánh tự cân bằng được tạo ra để giúp con người có thể thoải mái hơn trong việc lựa chọn hàng hoá, thưởng thức cảnh đẹp ở các khu du lịch do không cần phải bận tâm quá nhiều đền việc di chuyển. Với ưu điểm: xe nhỏ gọn có thể di chuyển ở khu vực đông người và đặc biệt là xe hoạt động dựa trên cảm biến cân bằng và chạy bằng điện nên không tạo ra khí thải rất thích hợp để sử dụng ở những khu vực trên.

Ngoài ra, xe hai bánh tự cân bằng còn có thể sử dụng làm robot phục vụ trong nhà hàng, robot sắp xếp sách trong thư viện hoặc xe lăn điều khiển tự động dành cho người khuyết tật… Trong giáo dục, xe được dùng làm mô hình học tập cho ngành Cơ Điện Tử phục vụ cho nghiên cứu về ứng dụng cảm biến và kỹ thuật điều khiển.

Xuất phát từ nhu cầu thực tiễn ấy, chúng em đã chọn đề tài: "THIẾT KẾ, CHẾ TẠO MÔ HÌNH XE HAI BÁNH TỰ CÂN BẰNG" nhằm đáp ứng nhu cầu ham muốn học hỏi của bản thân và giúp cho các bạn sinh viên có thể tiếp cận, tìm hiểu sâu hơn về board mạch STM32, cảm biến và các linh kiện điện tử chuyên dùng.

1.2 Mục tiêu đề tài

Mục tiêu của đề tài là xây dựng mô hình xe hai bánh tự cân bằng di chuyển trên địa hình phẳng, dựa trên nền tảng lý thuyết mô hình con lắc ngược. Khả năng di chuyển và cân bằng trên hai bánh làm mô hình có thể di chuyển hiệu quả, linh động hơn, dễ dàng xoay trở trong điều kiện không gian chật hẹp. Trong thời gian thực hiện luận văn tốt nghiệp những mục tiêu của đề tài được đề ra như sau:

- Tìm hiểu các loại xe scooter, nguyên lý cơ bản về cân bằng.
- Thiết kế mô hình cho xe.
- Tìm hiểu mạch điều khiển trung tâm và mạch lái động cơ.
- Tìm hiểu cảm biến MPU-6050 và đọc các giá trị thô đo được từ cảm biến.

- Xây dựng thuật toán điều khiển cho động cơ, giữ thăng bằng.
- Lập trình điều khiển.

1.3 Tình hình nghiên cứu trong và ngoài nước

1.3.1 Xe hai bánh tự cân bằng

Th.S Mai Tuấn Đạt cùng các cộng sự đã nghiên cứu và chế tạo thành công xe hai bánh tự cân bằng. Những kết quả mới, nổi bật của đề tài này là đã nghiên cứu 3 phương pháp điều khiển: đặt cực, cuốn chiếu, trượt. Họ đã sử dụng lần lượt các phương pháp này để thiết kế các bộ điều khiển giữ thăng bằng cho xe. Nhóm nghiên cứu cũng đã thực hiện mô phỏng điều khiển bằng phần mềm Matlab - Simulink, cải tiến phương thức giao tiếp truyền thông trong toàn bộ sản phẩm bằng việc đồng bộ hoá tất cả các luồng dữ liệu, thông tin, điều khiển từ board điều khiển trung tâm sang các board chức năng thông qua một chuẩn giao tiếp CAN duy nhất. Xe có thể đạt tốc độ tối đa 25 km/h, năng lượng được cung cấp từ bình điện, sử dụng được 26 km, nặng 54 kg.

Hình 1.1 Xe hai bánh tự cân bằng

1.3.2 Xe scooter của Segway

Segway PT viết tắt của Segway Personal Transporter là một phương tiện giao thông cá nhân có hai bánh, hoạt động trên cơ chế tự cân bằng do Dean Kamen phát minh. Loại xe này được sản xuất bởi công ty Segway Inc. ở Hoa Kỳ. Đặc điểm nổi bật của Segway là cơ chế tự cân bằng nhờ hệ thống máy tính, động cơ và con quay hồi chuyển đặt bên trong xe, nó giúp cho xe dù chỉ có một trục chuyển động với hai bánh nhưng luôn ở trạng thái cân bằng, người sử dụng chỉ việc ngả về đằng trước hoặc đằng sau để điều khiển xe đi tiến hoặc đi lùi. Với các điều khiển sang phải hoặc sang trái, Segway có một cần lái gọi là "Lean Steer" - muốn điều khiển sang phải hoặc sang trái chỉ cần nghiêng cần lái về phía đó. Động cơ của Segway PT có thể đạt tốc độ hay 20 km/h.

Hình 1.2 Xe scooter Segway

Cơ chế tự cân bằng của Segway dựa trên hoạt động của hệ thống máy tính, hai cảm biến độ nghiêng và năm con quay hồi chuyển đặt trong xe. Dựa trên các số liệu của cảm biến, máy tính sẽ tính toán để truyền lệnh cho các động cơ phụ di chuyển bánh xe về phía trước hoặc phía sau để tái lập cân bằng cho xe. Với các mẫu Segway PT mới, quá

trình này lặp đi lặp lại khoảng 100 lần một giây, đủ để cân bằng xe cho dù người lái ở trạng thái nào. Khi xe đạt tới vận tốc tối đa, các phần mềm trong Segway sẽ tự động điều khiển xe hơi nghiêng về sau giúp xe di chuyển chậm lại, cơ chế này giúp hạn chế khả năng người điều khiển tiếp tục nghiêng về trước ngay cả khi Segway đã ở vận tốc tối đa. Các Segway cũng sẽ tự động giảm tốc và dừng lại khi gặp chướng ngại vật.

1.3.3 nBot

nBot do David P.Anderson sáng chế được lấy ý tưởng để cân bằng như sau: các bánh xe sẽ phải chạy theo hướng mà phần trên robot sắp ngã. Trong thực tế, điều này đòi hỏi hai cảm biến thông tin phản hồi: Cảm biến góc nghiêng để đo góc nghiêng của robot với trọng lực, và encoder trên bánh xe để đo vị trí cơ bản của robot. Bốn thông số ngõ vào để xác định hoạt động và vị trí của xe cân bằng là:

- Góc nghiêng.
- Đạo hàm của góc nghiêng vận tốc góc.
- Vị trí bánh xe.
- Đạo hàm vị trí bánh, vận tốc bánh xe.

Bốn giá trị đo lường được cộng lại và phản hồi tới điện áp động cơ, tương ứng với moment quay, cân bằng và bộ phận lái robot.

Hình 1.3 nBot

1.3.4 JOE

Phòng thí nghiệm điện tử công nghiệp của Viện Công nghệ Federal, Lausanne, Thụy Sĩ, đã tạo ra cuộc cách mạng đầu tiên khi xây dựng mô hình xe hai bánh. Robot JOE cao 65cm, nặng 12kg, tốc độ tối đa khoảng 1,5m/s, có khả năng leo dốc nghiêng đến 30^o. Nguồn điện cấp là nguồn pin 32V, 1.8Ah. Hình dạng của nó gồm hai bánh xe, trục mỗi bánh gắn một động cơ DC, chiếc xe này có thể chuyển động xoay theo hình U. Hệ thống điều khiển được lắp từ hai bộ điều khiển state-space tách rời nhau, điều khiển động cơ để giữ cân bằng cho hệ thống. Những thông tin về trạng thái của JOE được cung cấp bởi hai encoder quang và con quay hồi chuyển. JOE được điều khiển bởi một bộ điều khiển từ xa R/C. Bộ điều khiển trung tâm và xử lý tín hiệu là một board xử lý tín hiệu số (DSP) có khả năng xử lý dấu chấm động (SHARC floating point), FPGA XILINC, 12 bộ biến đổi A/D 12bit và 4 bộ biến đổi A/D 10bit.

Hình 1.4 Robot JOE

1.3.5 Balance bot I

Balance bot I do Sanghyuk thực hiện là một robot hai bánh tự cân bằng sử dụng giải thuật LQR. Nó cao 50cm với khung chính được làm bằng nhôm. Hai trục bánh xe

gắn động cơ DC có hộp giảm tốc. Robot sử dụng ba bộ vi xử lý Atmel. Vi điều khiển chính thi hành những nguyên lý kiểm soát và thực hiện thuật toán ước lượng. Vi điều khiển thứ 2 kiểm soát toàn bộ cảm biến analog. Cái còn lại điều khiển động cơ DC. Bốn thông số đầu vào của robot là góc nghiêng, vận tốc nghiêng, góc quay bánh xe, vận tốc góc quay.

Hình 1.5 Balance Bot I

1.3.6 Balancing robot

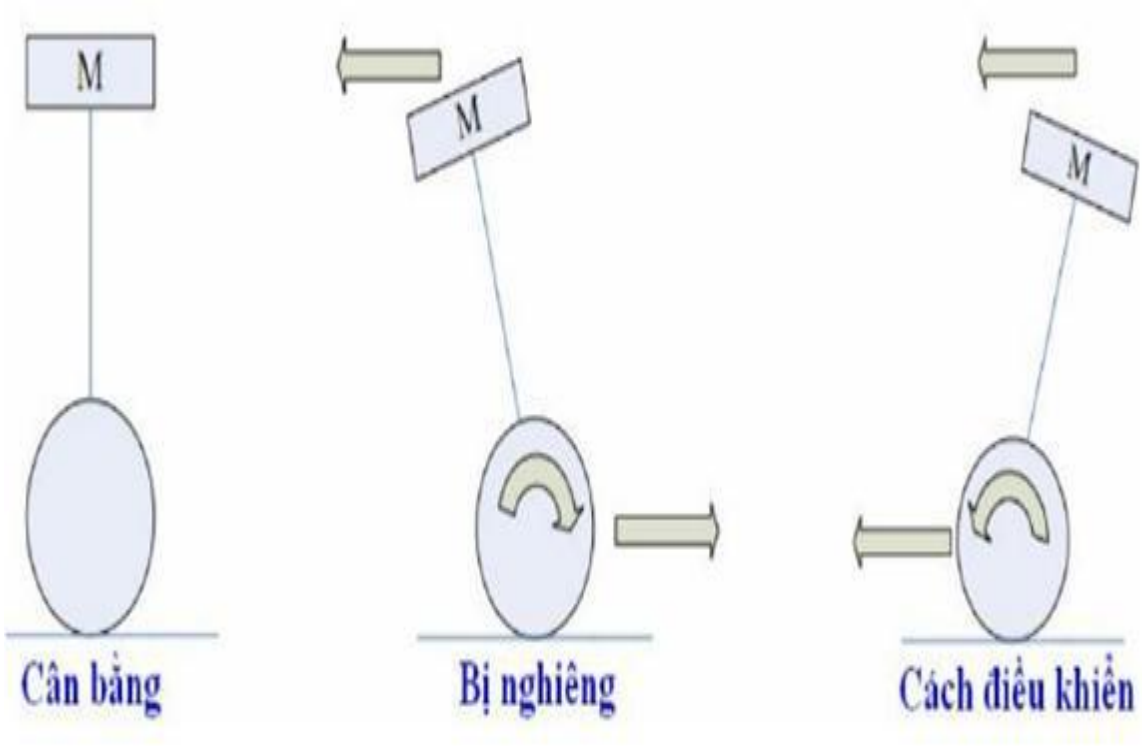
Vào năm 2003, Jack Wu và Jim Bai là những sinh viên trường Đại học Carnegie Mellol dưới sự giúp đỡ của GS.Chris Atkeson đã thực hiện đề tài robot hai bánh tự cân bằng như luận văn tốt nghiệp. Robot này có thể xác định vị trí của nó so với môi trường và lái động cơ theo hướng này. Để đo góc nghiêng của robot, các sinh viên này sử dụng hệ thống đo lường góc 2DOF được tích hợp sẵn của hãng Rotomotion. Hệ thống này gồm gia tốc kế ADXL202 và mạch con quay hồi chuyển. Vi mạch điều khiển dùng trên robot là BasicX24, có nhiều tính năng khác nhau. Nó được dùng như bộ điều khiển động cơ COM1 được nối với Pocket PC và COM3 thì nối với bộ điều khiển servo Mini SSC 12. Nó còn được sử dụng như CPU chính cho việc điều khiển thăng bằng cho robot.

Hình 1.6 Balancing Robot

Chương 2:
TỔNG QUAN

2.1 Nguyên lý cân bằng

Xe hai bánh tự cân bằng là loại xe chỉ có 2 bánh với trục của hai bánh xe trùng nhau để cho xe cân bằng, trọng tâm của xe (bao gồm cả người sử dụng) cần được giữ nằm ngay giữa các bánh xe.



Hình 2.1 Nguyên lý hoạt động của xe hai bánh tự cân bằng

Hoạt động của xe dựa vào nguyên lý con lắc ngược kết hợp điều khiển độc lập hai bánh xe.

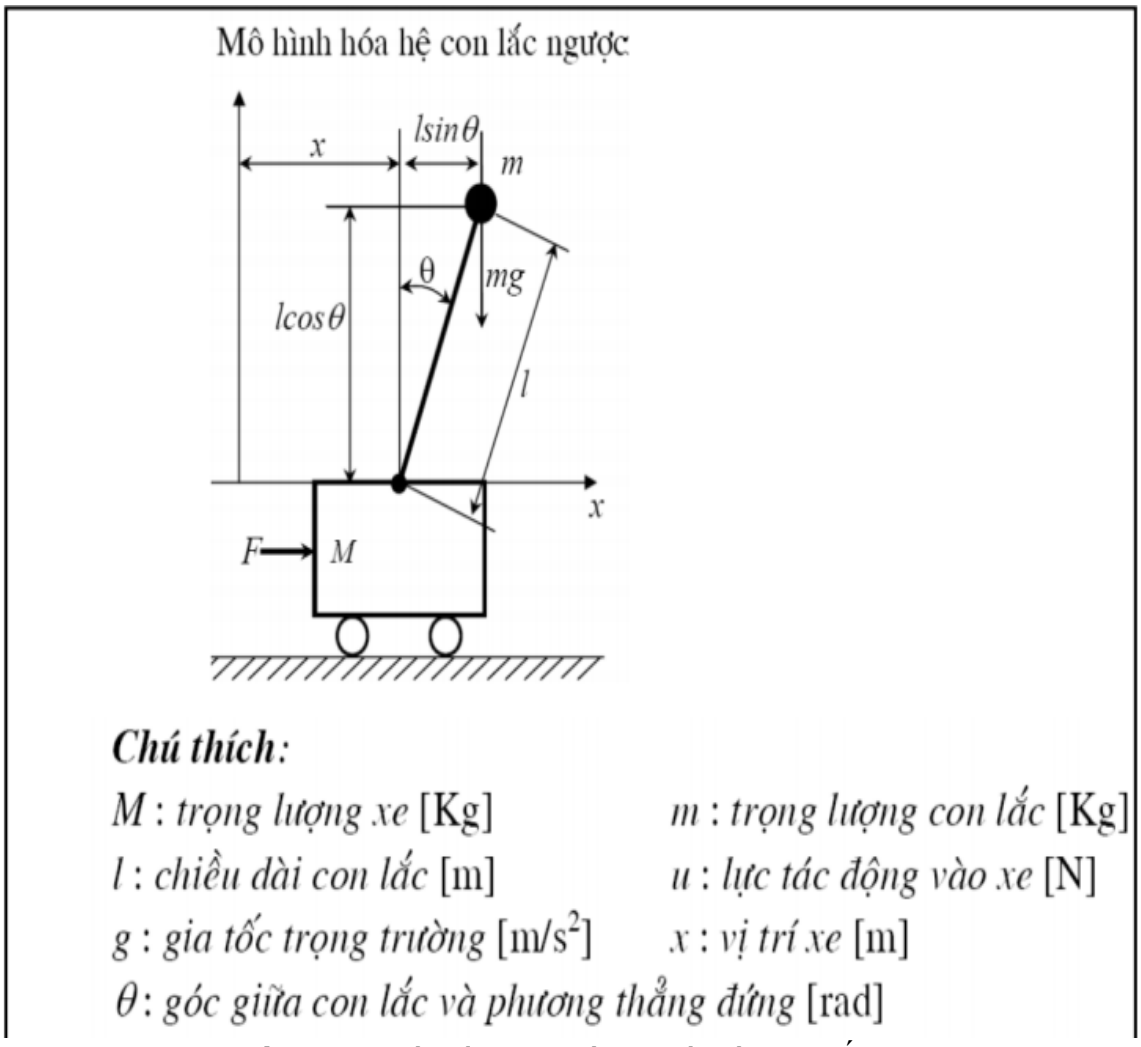
- Khi mô hình đứng cân bằng: góc nghiêng giữa thân xe và trục thẳng đứng bằng 0.
- Khi mô hình nghiêng về phía trước: góc nghiêng giữa thân xe và trục thẳng đứng lớn hơn 0, nếu không có điều khiển thì theo quán tính hai bánh sẽ chạy về phía sau dẫn đến xe bị ngã. Nên trong trường hợp này chúng ta sẽ điều khiển cho hai bánh xe chạy về phía trước. Dẫn đến góc nghiêng tiến về 0, xe sẽ trở về trạng thái cân bằng.

- Ngược lại khi mô hình nghiêng về phía sau.

2.2 Nền tảng nghiên cứu từ con lắc ngược:

Có nhiều phương pháp để tính động lực học, chẳng hạn: phương pháp Newton, phương pháp Lagrange, phương pháp theo năng lượng… Nhưng trong đề tài này phương pháp Newton được sử dụng với các ưu điểm của nó. Thứ nhất nó sử dụng các phương pháp tính cơ học thông thường. Thứ hai, các công thức và hệ phương trình trong quá trình tính không quá phức tạp. Thứ ba, kết quả tính động lực học của mô hình con lắc ngược được phổ biến hiện nay ở các tài liệu tham khảo được sử dụng để kiểm tra sự sai sót trong quá trình tính toán động lực học của mô hình xe hai bánh tự cân bằng.

Ta xét mô hình toán học của con lắc với các thông số sau:



Hình 2.2 Phân tích lực trên xe và trên con lắc

Phân tích lực của xe, ta có:

- Gọi x_p, y_p là tọa độ vật nặng m ở đầu con lắc:

$$x_p = x + l\sin\theta \rightarrow \ddot{x}_p = \ddot{x} - l(sin\theta)\dot{\theta}^2 + l(cos\theta)\ddot{\theta} \quad [2.1]$$

$$y_p = l\cos\theta \rightarrow \ddot{y}_p = -(l(cos\theta)\dot{\theta}^2 + l(sin\theta)\ddot{\theta}) \quad [2.2]$$

- Áp dụng định luật 2 Newton cho chuyển động theo phương x:

$$M.\ddot{x} + m.\ddot{x}_p = F \quad [2.3]$$

- Khai triển các đạo hàm ở công thức [2.3] ta có:

$$(M+m)\ddot{x} - ml(\sin\theta)\dot{\theta}^2 + ml(cos\theta)\ddot{\theta} = F \quad [2.4]$$

- Áp dụng định luật 2 Newton cho chuyển động quay quanh trục của con lắc:

$$m\ddot{x}_p l(cos\theta) - m\ddot{y}_p l(sin\theta) = mgl(sin\theta) \quad [2.5]$$

- Thay [2.1], [2.2] vào [2.5] ta có:

$$ml\ddot{x}(cos\theta) - ml^2(cos\theta)(sin\theta)\dot{\theta}^2 + ml^2(cos\theta)^2\ddot{\theta} + ml^2(cos\theta)(sin\theta)\dot{\theta}^2 + ml^2(sin\theta)^2\ddot{\theta} = mgl(sin\theta) \quad [2.6]$$

- Rút gọn ta được:

$$m\ddot{x}(cos\theta) + ml\ddot{\theta} = mg(sin\theta) \quad [2.7]$$

- Từ [2.4] và [2.7] ta thu được:

$$\ddot{x} = \frac{F+ml(sin\theta)\dot{\theta}^2 - mg(cos\theta)(sin\theta)}{M+m-m(cos\theta)^2} \quad [2.8]$$

$$\ddot{\theta} = \frac{F(cos\theta) - (M+m)g(sin\theta) + ml(cos\theta)(sin\theta)\dot{\theta}^2}{ml(cos\theta)^2 - (M+m)l} \quad [2.9]$$

2.3 Ưu nhược điểm của xe hai bánh tự cân bằng

- **Ưu điểm**

- Không ô nhiễm môi trường, sử dụng bình điện có thể sạc.
- Giá thành thấp, cuốn hút người sử dụng vì hình dáng kỳ lạ.
- Nhỏ gọn, ít chiếm diện tích, dễ sử dụng.

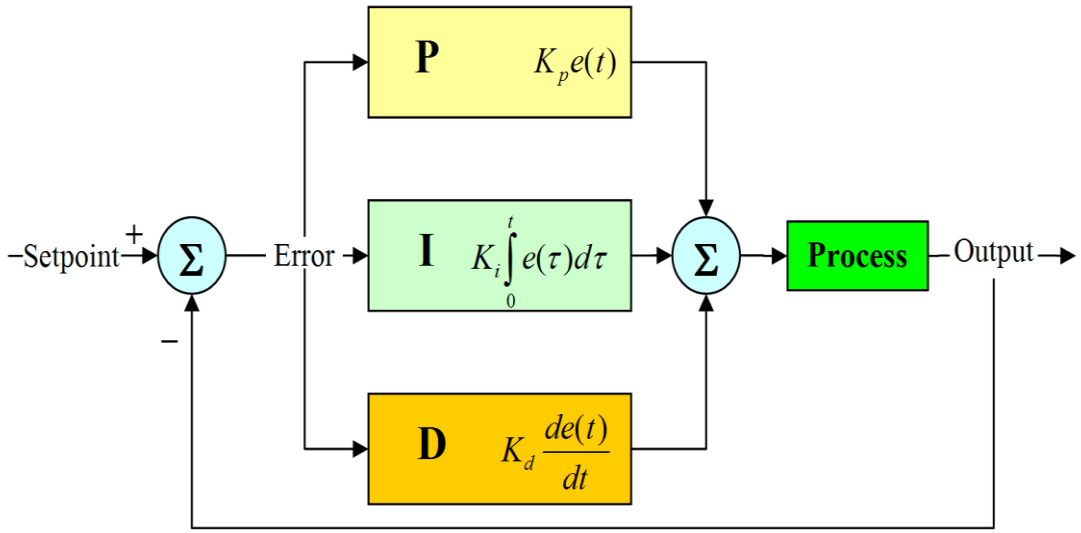
- **Nhược điểm**

- Không thể thư giãn và khá mệt do phải đứng trong khi điều khiển.
- Không đủ nhanh để đi đường trường và không đủ an toàn để lên xuống lề đường.
- Không thể leo bậc thang có chiều cao quá nửa bán kính xe.

2.4 Thuật toán điều khiển – Bộ điều khiển PID

2.4.1 Cơ bản về bộ điều khiển PID

Có nhiều thuật toán để giải quyết bài toán cân bằng như LQR, Fuzzy và PID. Trong đề tài này bộ điều khiển được sử dụng là PID.



Hình 2.3 Bộ điều khiển PID

Một bộ điều khiển vi tích phân tỉ lệ (bộ điều khiển PID - Proportional Integral Derivative) là một cơ chế phản hồi vòng điều khiển (bộ điều khiển) tổng quát được sử dụng rộng rãi trong các hệ thống điều khiển công nghiệp - bộ điều khiển PID được sử dụng phổ biến nhất trong số các bộ điều khiển phản hồi. Một bộ điều khiển PID tính toán một giá trị "sai số" là hiệu số giữa giá trị đo thông số biến đổi và giá trị đặt mong muốn. Bộ điều khiển sẽ thực hiện giảm tối đa sai số bằng cách điều chỉnh giá trị điều khiển đầu vào. Trong trường hợp không có kiến thức cơ bản về quá trình, bộ điều khiển PID là bộ điều khiển tốt nhất. Tuy nhiên, để đạt được kết quả tốt nhất, các thông số PID sử dụng trong tính toán phải điều chỉnh theo tính chất của hệ thống - trong khi kiểu điều khiển là giống nhau, các thông số phải phụ thuộc vào đặc thù của hệ thống.

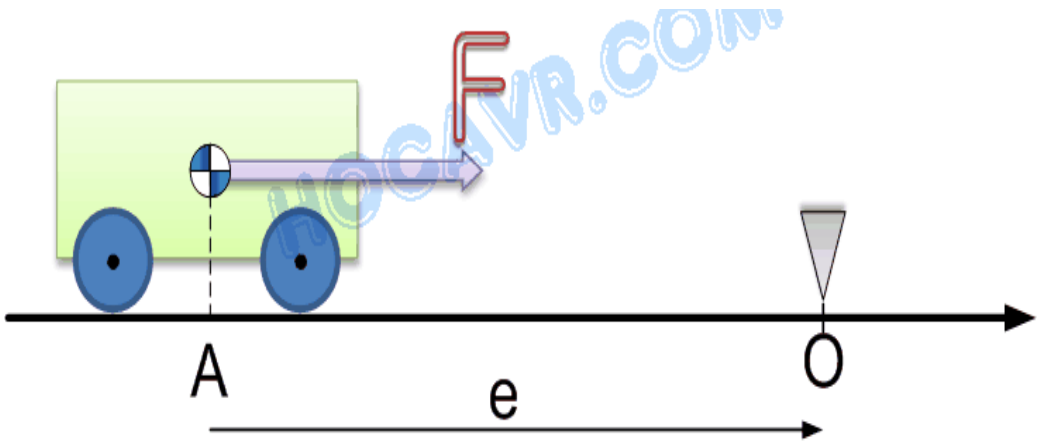
Giải thuật tính toán bộ điều khiển PID bao gồm 3 thông số riêng biệt, do đó đôi khi nó còn được gọi là điều khiển ba khâu: các giá trị tỉ lệ, tích phân và đạo hàm, viết

tắt là P, I, và D. Giá trị tỉ lệ xác định tác động của sai số hiện tại, giá trị tích phân xác định tác động của tổng các sai số quá khứ, và giá trị vi phân xác định tác động của tốc độ biến đổi sai số. Tổng chập của ba tác động này dùng để điều chỉnh quá trình thông qua một phần tử điều khiển như vị trí của van điều khiển hay bộ nguồn của phần tử gia nhiệt. Nhờ vậy, những giá trị này có thể làm sáng tỏ về quan hệ thời gian: P phụ thuộc vào sai số hiện tại, I phụ thuộc vào tích lũy các sai số quá khứ và D dự đoán các sai số tương lai, dựa vào tốc độ thay đổi hiện tại.

Bằng cách điều chỉnh 3 hằng số trong giải thuật của bộ điều khiển PID, bộ điều khiển có thể dùng trong những thiết kế có yêu cầu đặc biệt. Đáp ứng của bộ điều khiển có thể được mô tả dưới dạng độ nhạy sai số của bộ điều khiển, giá trị mà bộ điều khiển vọt lố điểm đặt và giá trị dao động của hệ thống. Lưu ý là công dụng của giải thuật PID trong điều khiển không đảm bảo tính tối ưu hoặc ổn định cho hệ thống.

2.4.2 Ứng dụng PID cho điều khiển vị trí xe

Giả sử ta có 1 xe đồ chơi có gắn động cơ. Động cơ chạy sinh ra lực đẩy xe chạy tới lui như trong hình 2.4.



Hình 2.4 Ví dụ điều khiển vị trí xe trên đường thẳng

Gọi F là lực do động cơ tạo ra để điều khiển xe. Ban đầu xe ở vị trí A, nhiệm vụ đặt ra là tạo 1 lực F để đẩy xe đến đúng vị trí O với các yêu cầu chính xác, nhanh và ổn định.

Nếu vị trí hiện tại của xe cách xa O hay nói cách khác là sai số lớn thì ta cần tác động 1 lực F để xe tiến nhanh về O. Một cách đơn giản để làm điều này là dùng quan hệ tuyến tính: $F = Kp * e$.

Trong đó Kp là một hằng số dương nào đó mà ta gọi là hệ số P (Propotional gain), e là sai số cần điều khiển tức là khoảng cách từ vị trí hiện tại của xe cho đến điểm đích là O. Mục tiêu điều khiển là làm sao đưa xe tiến về O càng nhanh càng tốt đồng nghĩa với việc sai số e tiến dần về 0. Nếu Kp lớn thì F sẽ lớn và xe nhanh chóng tiến về O. Tuy nhiên, lực F quá lớn làm cho gia tốc của xe nhanh. Khi xe chạy đến O (tức e = 0) thì do lực quán tính xe tiếp tục tiến về bên phải và lệch điểm O về bên phải, sai số e lúc đó lại trở nên khác 0, giá trị sai số lúc này được gọi là vọt lố (overshoot). Lúc này sai số e là số âm, lực F xuất hiện theo chiều ngược với F ban đầu để kéo xe về điểm O. Nhưng do Kp lớn nên lực F cũng lớn và có thể kéo xe lệch về bên trái điểm O. Quá trình cứ tiếp diễn xe cứ mãi dao động quanh điểm O. Có trường hợp xe dao động càng ngày càng xa điểm O. Lúc này bộ điều khiển được nói là không ổn định. Nhằm giảm vọt lố người ta đề xuất dùng một thành phần "thắng" trong bộ điều khiển. Khi xe ở xa điểm O thì bộ điều khiển sinh ra lực F lớn nhưng khi xe đến gần điểm O thì phần "thắng" sẽ giảm tốc độ xe lại. Khi một vật dao động quanh 1 điểm thì vật đó có vận tốc cao nhất ở tâm dao động. Nói cách khác là ở gần điểm O sai số e của xe thay đổi nhanh nhất. Tốc độ thay đổi của e có thể tính bằng đạo hàm của e theo thời gian. Như vậy khi xe từ A tiến về O, đạo hàm của sai số e tăng giá trị nhưng ngược chiều của lực F (do e đang giảm nhanh dần). Nếu dùng đạo hàm làm thành phần "thắng" thì có thể giảm được vọt lố của xe. Thành phần "thắng" này chính là thành phần D (Derivative) trong bộ điều khiển PID. Thêm thành phần D vào bộ điều khiển P hiện tại ta thu được bộ điều khiển PD như sau:

$F = Kp * e + Kd(de/dt)$.

Trong đó (de/dt) là vận tốc thay đổi của sai số e và Kd là một hằng số không âm gọi là hệ số D (Derivative gain).

Sự có mặt của thành phần D làm giảm vọt lố của xe, khi xe tiến gần về lực F gồm 2 thành phần $Kp * e \geq 0$ (thành phần P) và $Kd * (de/dt) \leq 0$ (thành phần D). Trong một số trường hợp thành phần D có giá trị lớn hơn P và lực F đổi chiều, " thắng" xe lại, vận tốc của xe vì vậy giảm mạnh gần điểm O. Một vấn đề nảy sinh là nếu thành phần D quá lớn so với thành phần P hoặc bản thân thành phần P nhỏ thì khi xe đến gần điểm O, xe có thể dừng hẳn, thành phần D bằng 0 (do sai số e không thay đổi nữa), lúc này lực $F = Kp * e$. Trong khi Kp và e lúc này đều nhỏ nên lực F cũng nhỏ và có thể không

thắng được lực ma sát tĩnh. Như vậy xe sẽ đứng yên mặc dù sai số e vẫn chưa bằng 0. Sai số e trong tính huống này gọi là steady state error (sai số trạng thái tĩnh). Để tránh steady state error người ta thêm vào bộ điều khiển một thành phần có chức năng "cộng dồn" sai số. Khi steady state error xảy ra, 2 thành phần P và D mất tác dụng, thành phần điều khiển mới sẽ "cộng dồn" sai số theo thời gian và làm tăng lực F theo thời gian. Đến một lúc nào đó, lực F đủ lớn để thắng ma sát tĩnh và đẩy xe tiến tiếp về điểm O. Thành phần "cộng dồn" này chính là thành phần I (Integral - tích phân) trong bộ điều khiển PID. Vì chúng ta đều biết, tích phân một đại lượng theo thời gian chính là tổng của đại lượng đó theo thời gian. Bộ điều khiển đến thời điểm này đã đầy đủ PID:

$F = Kp * e + Kd * (de/dt) + Ki * \int e\,dt$ với ($\int edt$ là tích phân của e theo t)

Như vậy chức năng của từng thành phần trong bộ điều khiển PID đã rõ. Tùy mục đích sử dụng và đối tượng điều khiển mà bộ PID có thể được lượt bớt để trở thành bộ điều khiển P, PI hoặc PD. Công việc chính của người thiết kế bộ điều khiển PID là chọn các hệ số Kp, Kd, Ki sao cho bộ điều khiển hoạt động tốt và ổn định (quá trình này gọi là PID gain tuning).

2.4.3 Cách biến đổi sang bộ PID số

Thành phần P là quan hệ tuyến tính vì vậy chỉ cần áp dụng trực tiếp công thức $Kp * e$ mà không cần bất kỳ xấp xỉ nào. Kế đến là xấp xỉ cho đạo hàm của biến e. Vì thời gian lấy mẫu cho các bộ điều khiển thường rất bé nên có thể xấp xỉ đạo hàm bằng sự thay đổi của e trong 2 lần lấy mẫu liên tiếp: $De/dt = (e(k) - e(k-1))/h$

Trong đó $e(k)$ là giá trị hiện tại của $e, e\ (k-1)$ là giá trị của e trong lần lấy mẫu trước đó và h là khoảng thời gian lấy mẫu.

Tích phân của biến e được tính bằng tổng diện tích các hình chữ nhật tại mỗi thời điểm đang xét. Mỗi hình chữ nhật có chiều rộng bằng thời gian lấy mẫu h và chiều cao là giá trị sai số e tại thời điểm đang xét.

$\int_0^t edt = \sum_0^n e(k) * h$

Tổng hợp các xấp xỉ công thức của bộ điều khiển PID số:

$F = Kp * e + \frac{e(k)-e(k-1)}{h} + \sum_0^n e(k) * h$

Trong đó F là output từ bộ điều khiển

- Từ những phân tích trên chúng em thực hiện bộ điều khiển PID dùng trong đề tài bằng cách sau:

Đầu tiên ta lấy tín hiệu đầu vào là góc đặt cho xe cân bằng (trong đề tài là 0). Tín hiệu từ cảm biến MPU6050 sẽ được so sánh với góc đặt để đưa vào bộ PID. Ngõ ra bộ PID là giá trị xung PWM để điều khiển động cơ. Nếu góc nghiêng càng lớn thì ngõ ra bộ PID càng lớn và như vậy PWM sẽ càng lớn đồng nghĩa moment quay của động cơ càng mạnh để đưa xe về vị trí cân bằng.

2.4.4 Phương pháp tìm thông số PID

- Chỉnh định bằng tay:

Chọn Kp trước thử bộ điều khiển P với đối tượng thật (hoặc mô phỏng), điều chỉnh Kp sao cho thời gian đáp ứng đủ nhanh, chấp nhận overshoot nhỏ. Thêm thành phần D để loại overshoot, tăng Kd từ từ và chọn giá trị thích hợp. Lúc này steady state error có thể sẽ xuất hiện. Thêm thành phần I để giảm steady state error. Nên tăng I từ bé đến lớn để giảm steady state error đồng thời không để cho overshoot xuất hiện trở lại.

- Chỉnh định bằng phần mềm:

Hầu hết các ứng dụng công nghiệp hiện đại không còn điều chỉnh PID bằng các phương pháp tính toán thủ công như trên nữa. Thay vào đó người ta dùng phần mềm tự động để chỉnh định thông số PID (thực hiện trên mô hình toán, kiểm nghiệm trên mô hình thực). Ví dụ như là dùng giải thuật di truyền (GA) để tìm thông số sao cho sai số đo được nhỏ hơn giá trị yêu cầu.

- Chỉnh định bằng phương pháp Ziegler - Nichols:

Một phương pháp điều chỉnh PID khác là phương pháp Ziegler - Nichols được John G.Ziegler và Nathaniel B.Nichols đưa ra vào những năm 1940. Giống như chỉnh định bằng tay, độ lợi Ki và Kd lúc đầu được gán bằng không. Độ lợi P được tăng cho tới khi hệ thống dao động tuần hoàn. Đặt giá trị Kp = Ku.

Đo chu kì dao động Tu. Sau đó tính từng thông số theo bảng 2.1

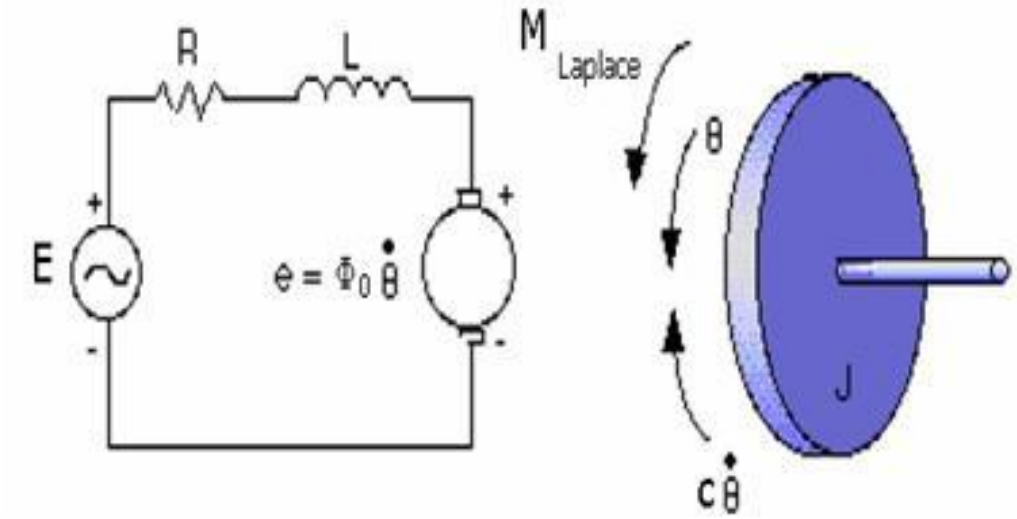
Bảng 2.1 Phương pháp Ziegler - Nichols

Phương pháp Ziegler - Nichols			
Loại điều khiển	K_p	K_i	K_d
P	$K_u/2$	-	-
PI	$K_u/2.2$	$1.2\,K_p/T_u$	-
Classic PID	$0.6K_u$	$2\,K_p/T_u$	$K_pT_u/8$
Pessen Integral Rule	$0.7K_u$	$2.5\,K_p/T_u$	$0.15K_pT_u$
Some overshoot	$0.33K_u$	$2\,K_p/T_u$	$K_pT_u/3$
No overshoot	$0.2K_u$	$2\,K_p/T_u$	$K_pT_u/3$

Trong phạm vi đề tài phương pháp chỉnh định bằng tay được sử dụng.

2.5 Mô hình lý thuyết động cơ DC

Động cơ DC có thể được mô hình hóa một cách đơn giản như hình vẽ:

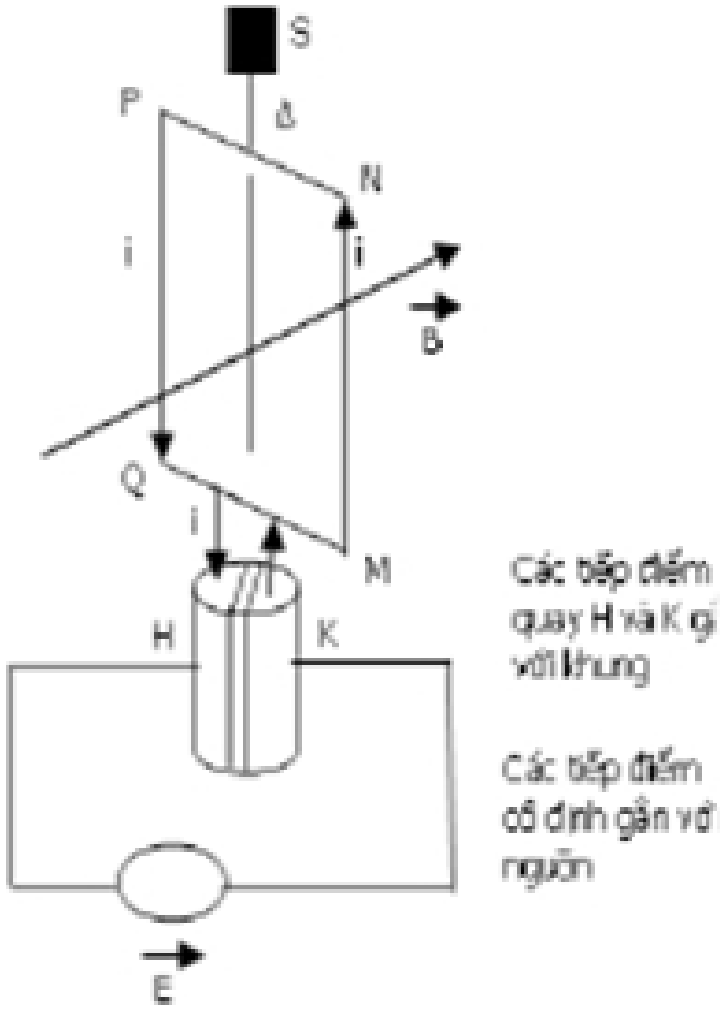


Hình 2.5 Mô hình động cơ

Động cơ gồm một cuộn dây có N vòng dây quấn xung quanh một khung chữ nhật cạnh a, b có thể quay quanh một trục Δ. Vị trí của nó được xác định bởi góc quay θ.

Cuộn dây có điện trở tổng cộng R và độ tự cảm L. Hệ chuyển động có momen quán tính J với trục Δ. Một nam châm vĩnh cửu tạo ra từ một từ trường B xuyên qua khung dây. Hệ cơ S tác dụng lên trục một ngẫu lực cản, kí hiệu (Γ) được tính theo công thức: $\Gamma = c * \dot{\theta}$ (với c là hằng số giảm chấn của kết cấu cơ khí).

Khi ta đặt điện áp một chiều E vào 2 đầu chổi điện H và K thì torng cuộn dây phần ứng có dòng điện Iu. Theo hiện tượng cảm ứng điện từ khung dây sẽ chịu tác dụng của lực điện trường (lực Laplace) và quay quanh trục Δ. Chú ý rằng cứ mỗi nửa vòng lại chuyển mạch, ở đây H và K là 2 vành bán khuyên. Điều này bảo đảm cho khung luôn quay theo 1 chiều nhất định với vận tốc góc $\omega = \dot{\theta}$.



Hình 2.6 Mô hình khung dây động cơ đặt trong từ trường

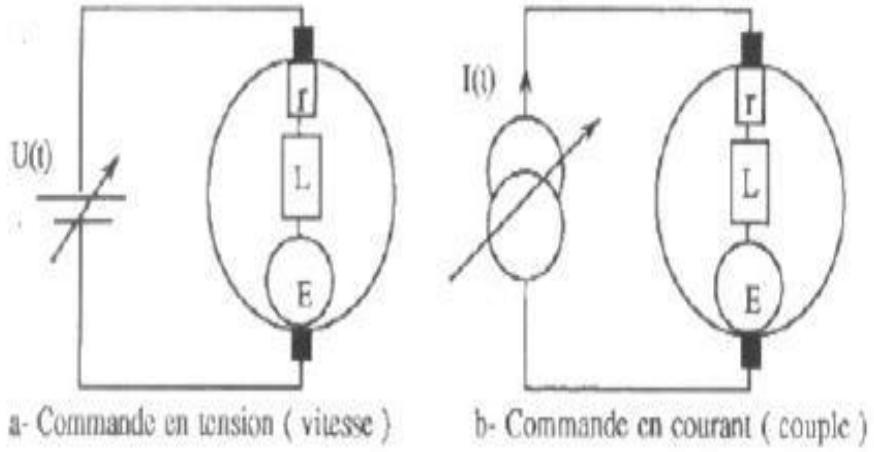
$E(t), I(t)$ là tổng điện áp cảm ứng và dòng dẫn trong động cơ. $C(t), \theta'(t)$ là ngẫu lực điện từ và vận tốc xoay của rotor. Ta có công thức:

$$E(t) = k * \theta'(t)$$

$$C(t) = k * I(t)$$

Với k là hằng số phụ thuộc vào đặc tính của động cơ, loại cuộn dây, giá trị từ trường, thường là hằng số với nam châm vĩnh cữu. Cuộn dây dẫn được mô hình hóa bởi điện trở R và độ dẫn L, được áp vào điện áp U(t), có phương trình dưới đây:

$$U(t) = r * I(t) + L\frac{d}{dt}I(t) + E(t)$$



Hình 2.7 Sơ đồ động cơ điện

Phương trình cơ học: $J\frac{d}{dt}\theta(t) = C(t) - Fs(t) - fv\theta v\theta(t)$

Trong mối quan hệ ở trên, sự phụ thuộc vận tốc góc $\theta'(t)$ theo $C(t)$ là đặc tính cơ học của quán tính vật rắn và ma sát xoắn. Với J là moment quán tính là hằng số, $Fs(t)$ là ngẫu lực xoay và fv là hệ số ma sát quay.

2.6 Board mạch điều khiển

STM32F103C8T6 là một vi điều khiển (MCU) 32-bit dựa trên kiến trúc ARM Cortex-M3 của hãng STMicroelectronics. MCU này có 64 KB bộ nhớ flash, 20 KB bộ nhớ RAM, và 32 chân I/O. STM32F103C8T6 được sử dụng trong nhiều ứng dụng khác nhau, bao gồm:

- Thiết bị điện tử tiêu dùng
- Thiết bị công nghiệp

- Thiết bị tự động hóa
- Thiết bị IoT

Các tính năng nổi bật của STM32F103C8T6:

- Kiến trúc ARM Cortex-M3 mang lại hiệu năng cao, khả năng xử lý đa luồng và khả năng tiết kiệm năng lượng.
- Bộ nhớ flash 64 KB cho phép lưu trữ chương trình và dữ liệu.
- Bộ nhớ RAM 20 KB cho phép lưu trữ dữ liệu tạm thời.
- 32 chân I/O cho phép kết nối với các thiết bị ngoại vi khác nhau.
- Hỗ trợ nhiều giao thức ngoại vi, bao gồm USB, UART, SPI, I2C, CAN, ...

Ứng dụng của STM32F103C8T6:

STM32F103C8T6 là một MCU linh hoạt và có thể được sử dụng trong nhiều ứng dụng khác nhau. Dưới đây là một số ví dụ:

- Thiết bị điện tử tiêu dùng: STM32F103C8T6 có thể được sử dụng trong các thiết bị điện tử tiêu dùng như TV, máy tính xách tay, điện thoại thông minh, ...
- Thiết bị công nghiệp: STM32F103C8T6 có thể được sử dụng trong các thiết bị công nghiệp như máy móc, thiết bị tự động hóa, ...
- Thiết bị tự động hóa: STM32F103C8T6 có thể được sử dụng trong các thiết bị tự động hóa như robot, máy móc, ...
- Thiết bị IoT: STM32F103C8T6 có thể được sử dụng trong các thiết bị IoT như cảm biến, máy đo, ...

Lợi ích của việc sử dụng STM32F103C8T6

STM32F103C8T6 mang lại một số lợi ích cho người dùng, bao gồm:

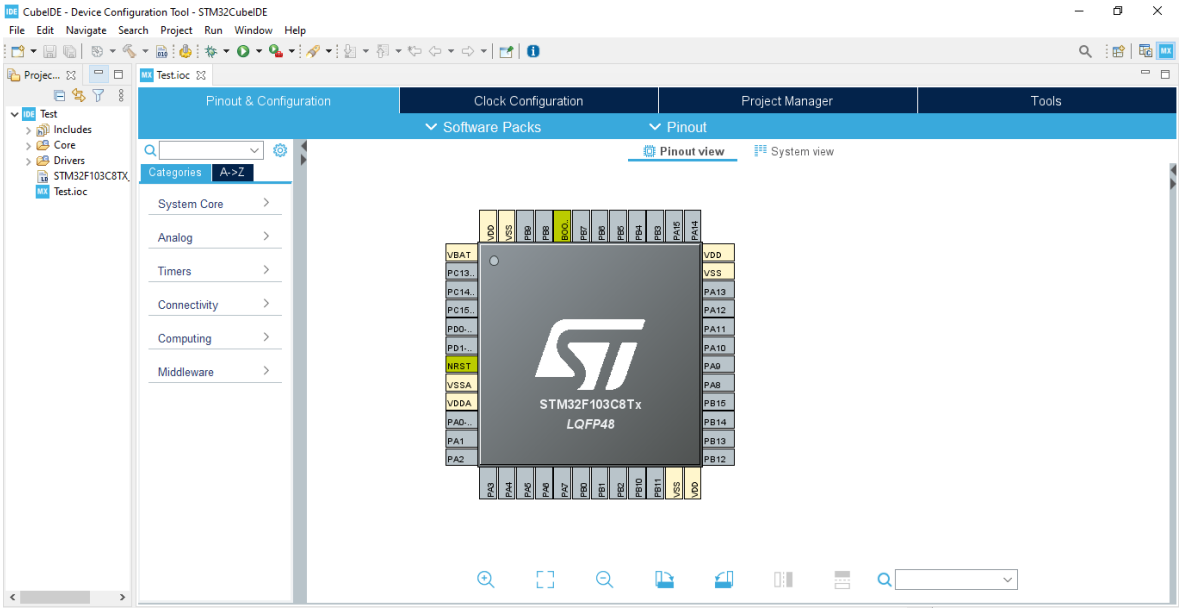
- Hiệu năng cao: ARM Cortex-M3 mang lại hiệu năng cao cho STM32F103C8T6, cho phép nó chạy các ứng dụng phức tạp một cách mượt mà.
- Khả năng mở rộng: STM32F103C8T6 có thể được mở rộng bằng cách sử dụng các module ngoại vi khác nhau, giúp nó đáp ứng được nhiều yêu cầu khác nhau của ứng dụng.
- Chi phí hợp lý: STM32F103C8T6 có giá thành hợp lý, giúp nó trở thành một lựa chọn tốt cho các dự án có ngân sách hạn chế.

- Môi trường lập trình bo mạch STM32

Thiết kế bo mạch nhỏ gọn, trang bị nhiều tính năng thông dụng mang lại nhiều lợi thế cho STM32, tuy nhiên sức mạnh thực sự của STM32 nằm ở phần mềm. Môi trường lập trình đơn giản dễ sử dụng, ngôn ngữ lập trình Wiring dễ hiểu và dựa trên nền tảng C/C++ rất quen thuộc với người làm kỹ thuật.Và quan trọng là số lượng thư viện code được viết sẵn và chia sẻ bởi cộng đồng nguồn mở là cực kỳ lớn.

Môi trường lập trình STM32CubeIDE có thể chạy trên ba nền tảng phổ biến nhất hiện nay là Windows, Macintosh OSX và Linux. Do có tính chất nguồn mở nên môi trường lập trình này hoàn toàn miễn phí và có thể mở rộng thêm bởi người dùng có kinh nghiệm.

Ngôn ngữ lập trình có thể được mở rộng thông qua các thư viện C++. Và do ngôn ngữ lập trình này dựa trên nền tảng ngôn ngữ C của AVR nên người dùng hoàn toàn có thể nhúng thêm code viết bằng AVR C vào chương trình nếu muốn.



Hình 2.8 Giao diện STM32CubeIDE

Chương 3:

NỘI DUNG VÀ PHƯƠNG PHÁP NGHIÊN CỨU

3.1 Phương pháp nghiên cứu đề tài

3.1.1 Xây dựng mô hình lý thuyết gồm có:

- Tiếp cận từ mô hình tương đương là con lắc ngược đến mô hình của đề tài.
- Mô phỏng mô hình bằng Matlab.

3.1.2 Tiếp cận mô hình thực gồm có:

- Thiết kế khung sườn của mô hình.
- Công suất điện và điện tử (điều khiển hai động cơ).
- Mạch cảm biến.
- Calib cảm biến.
- Bộ điều khiển trung tâm.
- Lập trình vi điều khiển.

3.2 Vật liệu thực hiện

3.2.1 Phần cứng

- Vi điều khiển STM32F103C8T6
- Module điều khiển động cơ L298
- Module Encoder GA25-370.
- Module Bluetooth HC-05
- Cảm biến MPU-6050
- Các đồ dùng cần thiết trong quá trình làm mạch: mũi hàn, khoan, nhựa thông, đồng hồ VOM và một số linh kiện điện tử khác…

3.2.2 Phần mềm

Để có được một mạch điện tử dùng vi điều khiển cần phải có sự kết hợp của hai thành phần là phần cứng và phần mềm. Phần mềm là các chương trình hỗ trợ cho chúng ta trong quá trình thiết kế và lập trình cho vi điều khiển. Trong đồ án, chúng em sử dụng các phần mềm và chương trình sau:

- STM32CubeMX: phần mềm cấu hình chân cho STM32.
- STM32CubeIDE: phần mềm biên dịch và lập trình cho STM32.

3.3 Nội dung thực hiện

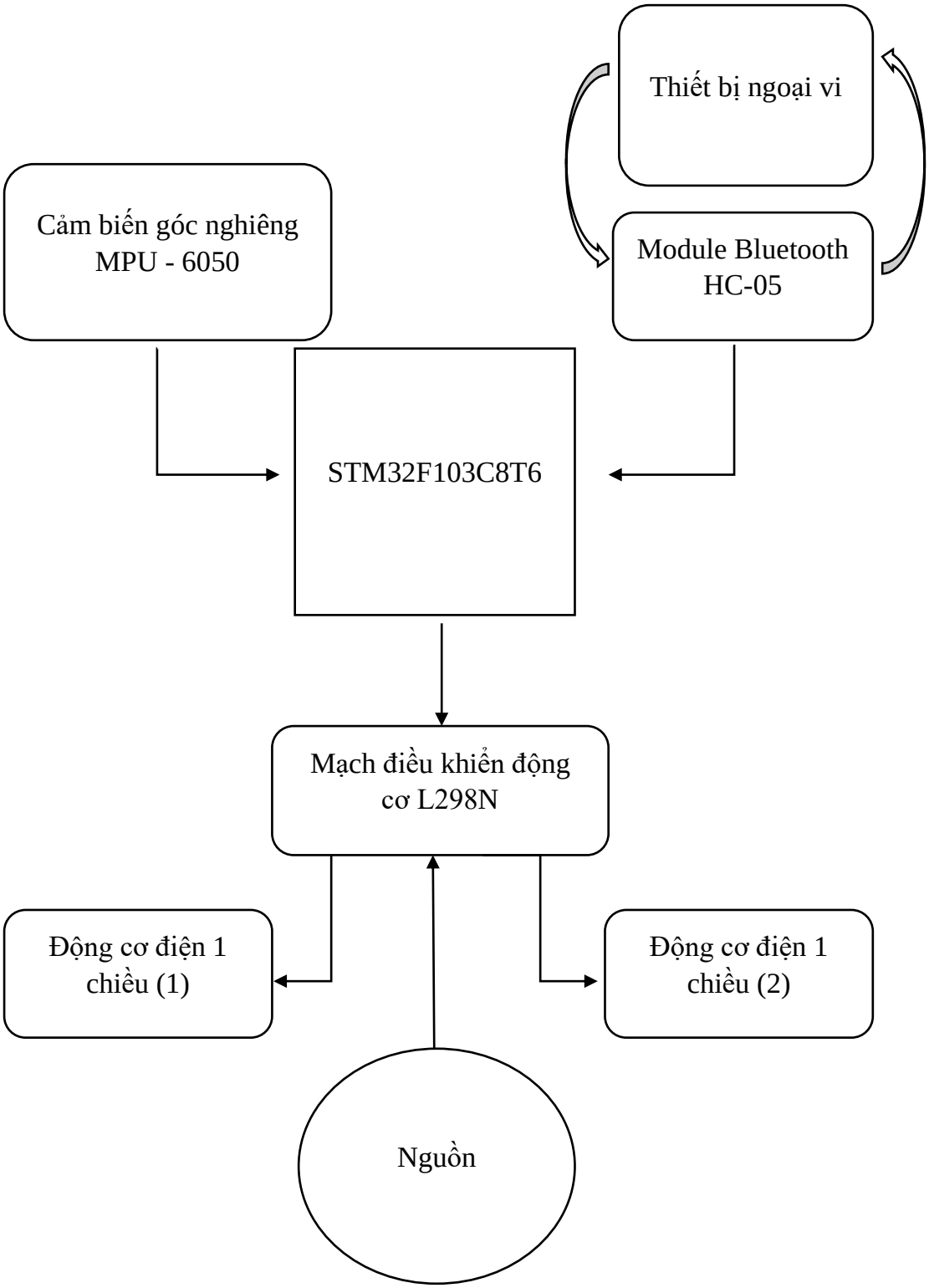
Cấu trúc của mô hình gồm hai phần: phần cứng và phần mềm. Phần cứng là thuật ngữ chỉ các mạch điện tử trong mô hình, nó quyết định các chức năng của mô hình. Nhưng để khai thác hết các chức năng của mô hình này, thì cần phải có phần mềm là các chương trình điều khiển giao tiếp với bộ vi xử lý thông qua các phần tử ngoại vi như cảm biến. Phần cứng và phần mềm có quan hệ chặt chẽ với nhau để tạo nên một mô hình xe hai bánh tự cân bằng có chất lượng và hiệu quả.

Mô hình được xây dựng từ các tấm mica mỏng nhẹ làm thân xe, trên khung gắn hai động cơ DC, hai động cơ gắn đồng trục với nhau. Sử dụng vi điều khiển STM32F103C8T6 làm bộ điều khiển trung tâm, điều khiển mạch lái động cơ và đọc các giá trị thô từ cảm biến MPU6050. Nguồn cung cấp cho xe hoạt động gồm: nguồn 12V cung cấp cho mạch lái động cơ, nguồn 9V cung cấp cho vi điều khiển. Xe có thể điều khiển tới, lui, quay trái, quay phải bằng smart phone truyền nhận qua module Bluetooth HC-06.

Chương 4:

KẾT QUẢ VÀ THẢO LUẬN

4.1 Sơ đồ khối tổng quát



Hình 4.1 Sơ đồ khối của toàn bộ mô hình

4.2 Các khối chức năng của mô hình

4.2.1 Khối cảm biến

MPU-6050 là cảm biến của hãng InvenSense. MPU-6050 là một trong những giải pháp cảm biến chuyển động đầu tiên trên thế giới có tới 6 (mở rộng tới 9) trục cảm biến tích hợp trong 1 chip duy nhất. MPU-6050 sử dụng công nghệ độc quyền MotionFusion của InvenSense có thể chạy trên các thiết bị di động, tay điều khiển... MPU-6050 tích hợp 6 trục cảm biến bao gồm: 3 trục con quay hồi chuyển và cảm biến gia tốc 3 chiều.

Hình 4.2 Module cảm biến MPU-6050

Ngoài ra, MPU-6050 còn có 1 đơn vị tăng tốc phần cứng chuyên xử lý tín hiệu (Digital Motion Processor - DMP) do cảm biến thu thập và thực hiện các tính toán cần thiết. Điều này giúp giảm bớt đáng kể phần xử lý tính toán của vi điều khiển, cải thiện tốc độ xử lý và cho ra phản hồi nhanh hơn. Đây chính là 1 điểm khác biệt đáng kể của MPU-6050 so với các cảm biến gia tốc và gyro khác. MPU-6050 có thể kết hợp với cảm biến từ trường (bên ngoài) để tạo thành bộ cảm biến 9 góc đầy đủ thông qua giao tiếp I2C.

Các cảm biến bên trong MPU-6050 sử dụng bộ chuyển đổi tương tự - số (Anolog to Digital Converter - ADC) 16-bit cho ra kết quả chi tiết về góc quay, tọa độ... Với 16-bit sẽ có $2^{16} = 65536$ giá trị cho 1 cảm biến. Hơn nữa, MPU-6050 có sẵn bộ đệm dữ liệu 1024 byte cho phép vi điều khiển phát lệnh cho cảm biến, và nhận về dữ liệu sau khi MPU-6050 tính toán xong.

Cảm biến MPU-6050 có thể hoạt động ở chế độ tốc độ xử lý cao hoặc chế độ đo góc quay chính xác (chậm hơn). MPU-6050 có khả năng đo ở phạm vi:

- con quay hồi chuyển: ± 250 500 1000 2000 dps.
- gia tốc: ± 2 ± 4 ± 8 ± 16g.

4.2.2 Module Bluetooth HC-05

Đây là module bluetooth SLAVE nghĩa là không thể chủ động kết nối bằng vi điều khiển, mà cần sử dụng smartphone, laptop, bluetooth usb... để dò tín hiệu và kết nối từ smartphone, laptop, bluetooth usb... Sau khi kết nối thành công, có thể gửi và nhận tín hiệu từ vi điều khiển đến các thiết bị này. Với board mạch HC-05 việc giao tiếp giữa STM32 với smartphone trở nên hết sức dễ dàng.

Module bluetooth được tích hợp trên board cho phép sử dụng nguồn từ 3.3V đến 5V cung cấp cho board mà không cần lo lắng về chênh lệch điện áp 3V - 5V gây hỏng board. Tần số làm việc của board là: 2,4GHz. Module Bluetooth gồm 6 chân theo thứ tự: KEY, VCC, GND, TX, RX, STATE.

Hình 4.3 Molude Bluetooth HC-05

Board HC-05 có hai chân TX và RX dùng để giao tiếp với vi xử lý thông qua UART. Sau đó, dữ liệu này sẽ được truyền thông qua sóng Bluetooth đến smartphone hoặc máy tính kết nối với board HC-06. Mặc định, khi chưa kết nối HC-05 nhận lệnh ở chế độ AT, còn khi đã kết nối nó hoạt động ở chế độ truyền nhận dữ liệu.

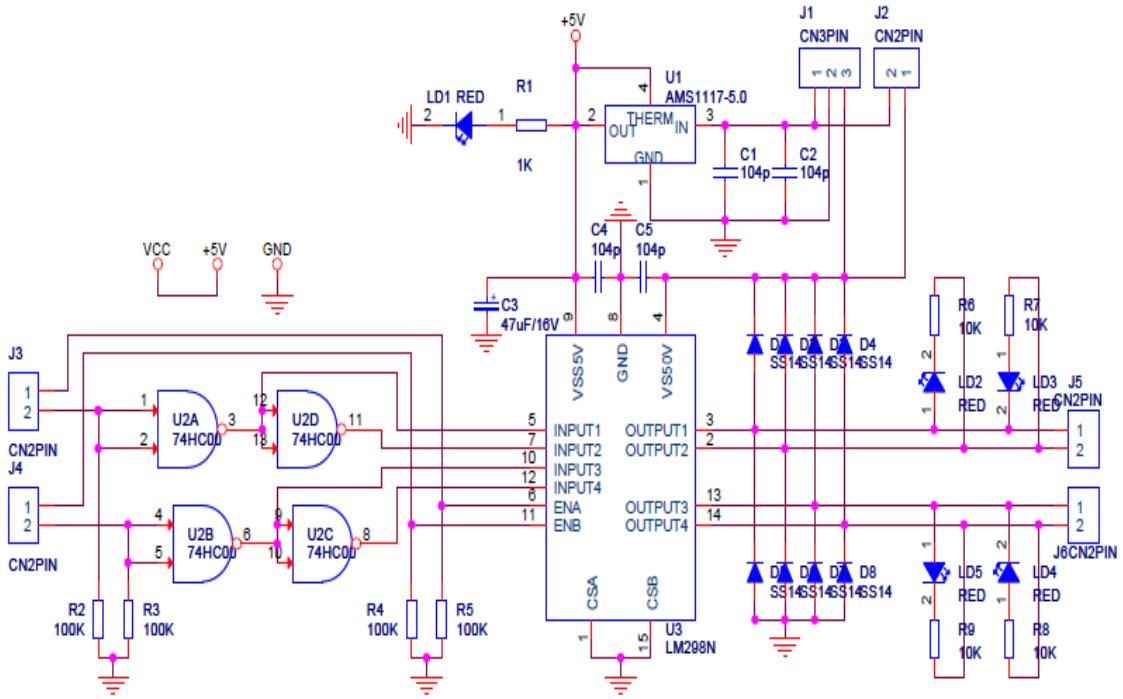
- Chế độ AT: Cho phép thay đổi cấu hình tính năng cho board HC-05.

- Chế độ truyền nhận dữ liệu: Ở chế độ này vi điều khiển có thể giao tiếp không dây với smartphone hoặc máy tính thông qua Bluetooth nhờ HC-06. Để vi điều khiển STM32 có thể truyền nhận với smartphone thông qua sóng Bluetooth chỉ việc kết nối chân TX của vi điều khiển vào chân RX của HC-06 và RX của vi điều khiển vào TX của HC-06.

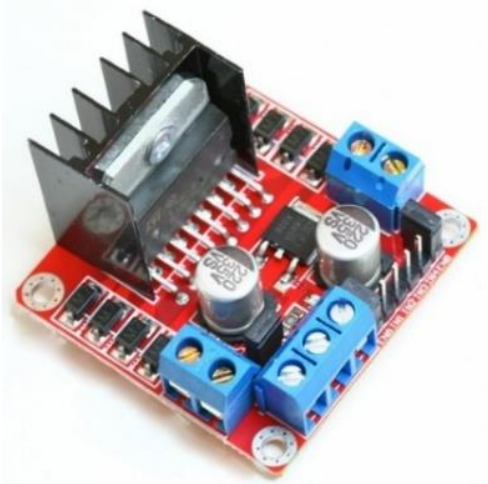
Một số thông số của module bluetooth HC-06:

- Sử dụng CSR mainstream bluetooth chip, bluetooth V2.0 protocol standards.
- Điện thế hoạt động 3.3 - 5V.
- Dòng điện khi hoạt động: khi Pairing 30 mA, sau khi pairing hoạt động truyền nhận bình thường 8 mA.
- Baudrate có thể chọn được: 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200.
- Kích thước của module chính: 28 mm x 15 mm x 2.35 mm.
- Dải tần sóng hoạt động: 2.4GHz.
- Nhiệt độ lưu trữ: -40 to 85° C.

4.2.3 Khối công suất



Hình 4.4 Sơ đồ nguyên lý module L298



Hình 4.5 Module điều khiển động cơ L298

Đối với mạch công suất có khá nhiều loại mạch sử dụng các IC khác nhau .Trong đề tài này cần một mạch công suất bao gồm mạch cầu H kết hợp PWM để đảo chiều và điều chỉnh tốc độ động cơ. Vì vậy chúng tôi chọn Module L298. Module L298 có thể điều khiển 2 động cơ DC hoặc 1 động cơ bước cùng lúc với dòng tối đa cho mỗi cầu H lên đến 2A.

- Các chân của module L298

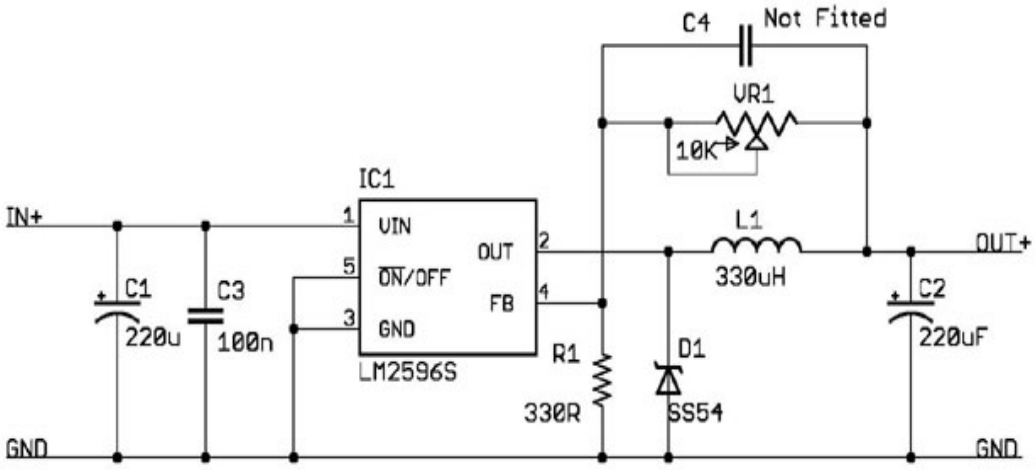
Chân VCC và GND: hai chân cấp nguồn để điều khiển động cơ. Module này đã có tích hợp sẵn IC ổn áp LM7805 nên không cần cấp thêm nguồn 5V nữa.

Ngõ ra OUTA, OUTB và OUTC, OUTD: hai cặp ngõ ra để kết nối với động cơ. Chân ENA và ENB: hai chân enable cho hai động cơ. Nó dùng để kết nối với ngõ ra PWM từ vi xử lý. Các chân IN1, IN2 và IN3, IN4: hai cặp chân dùng để điều chỉnh chiều quay của động cơ.

Bảng 4.1 Bảng chân trị L298

IN1/IN3	IN2/IN4	Trạng thái của động cơ
0	0	Ngừng
0	1	Quay thuận
1	0	Quay ngược
1	1	Ngừng

4.2.4 Khối nguồn



Hình 4.6 Sơ đồ nguyên lý LM2596

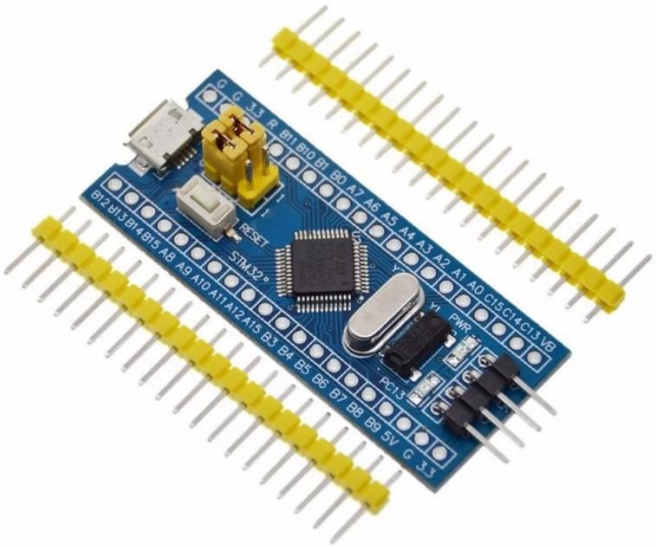
Thông số kỹ thuật:

- Điện áp đầu vào: Từ 3V đến 40V.
- Điện áp đầu ra: Điều chỉnh được trong khoảng 1.5V đến 35V.
- Dòng đáp ứng tối đa là 3A.
- Kích thước: 45 x 20 x 14mm.

Hình 4.7 Module LM2596

Trong đề tài chúng tôi sử dụng nguồn từ 3 pin Panasonic. Cấp vào 2 chân In của LM2596 để lấy ra 12V cho động cơ và 7v cho Board STM32F103C8T6.

4.2.5 Khối xử lý



Hình 4.8 Mạch STM32F103C8T6

- Nguồn cấp: điện áp cấp 5VDC qua cổng Micro USB sẽ được chuyển đổi thành 3.3VDC qua IC nguồn và cấp cho Vi điều khiển chính.
- Các chân nguồn:

  - VDD là chân nguồn chính của vi điều khiển. Chân này phải được cấp điện áp từ 2.2 V đến 3.6 V.
  - VBAT là chân nguồn phụ của vi điều khiển. Chân này có thể được sử dụng để cung cấp điện áp cho các thành phần ngoại vi, chẳng hạn như bộ nhớ flash. Điện áp của chân VBAT có thể dao động từ 1.8 V đến 3.6 V.
  - Các chân Input Output

STM32F103C8T6 có tổng cộng 32 chân, trong đó có 24 chân có thể được cấu hình làm đầu vào hoặc đầu ra.

Chân đầu vào được sử dụng để nhận tín hiệu từ các thiết bị ngoại vi. Chân đầu vào có thể được sử dụng cho nhiều chức năng khác nhau, chẳng hạn như:

- Đọc trạng thái của một nút nhấn.
- Nhận dữ liệu từ một cảm biến.
- Kích hoạt ngắt.

Chân đầu ra được sử dụng để xuất tín hiệu đến các thiết bị ngoại vi. Chân đầu ra có thể được sử dụng cho nhiều chức năng khác nhau, chẳng hạn như:

- Điều khiển một đèn LED.
- Truyền dữ liệu đến một thiết bị ngoại vi.
- Kích hoạt một thiết bị ngoại vi.

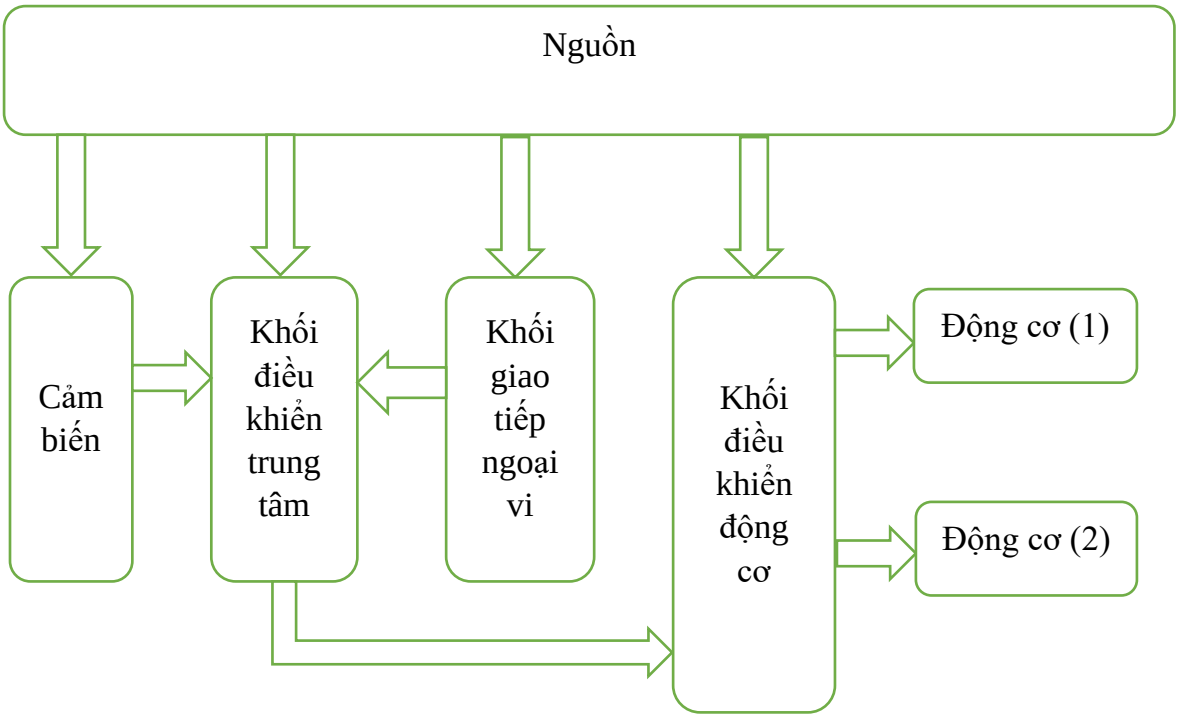
Có một số chân output của STM32F103C8T6 có chức năng đặc biệt. Các chân này thường được sử dụng cho các chức năng như:

- Kích hoạt ngắt: Một số chân output có thể được sử dụng để kích hoạt ngắt. Khi chân này được đặt thành trạng thái HIGH, nó sẽ kích hoạt ngắt.
- Kết nối với đồng hồ bên ngoài: Một số chân output có thể được sử dụng để kết nối với đồng hồ bên ngoài. Đồng hồ bên ngoài có thể được sử dụng để tăng tốc độ của vi điều khiển.
- Điều khiển các chức năng khác của vi điều khiển: Một số chân output có thể được sử dụng để điều khiển các chức năng khác của vi điều khiển. Ví dụ, chân PC13 có thể được sử dụng để điều khiển đèn LED tích hợp của vi điều khiển.

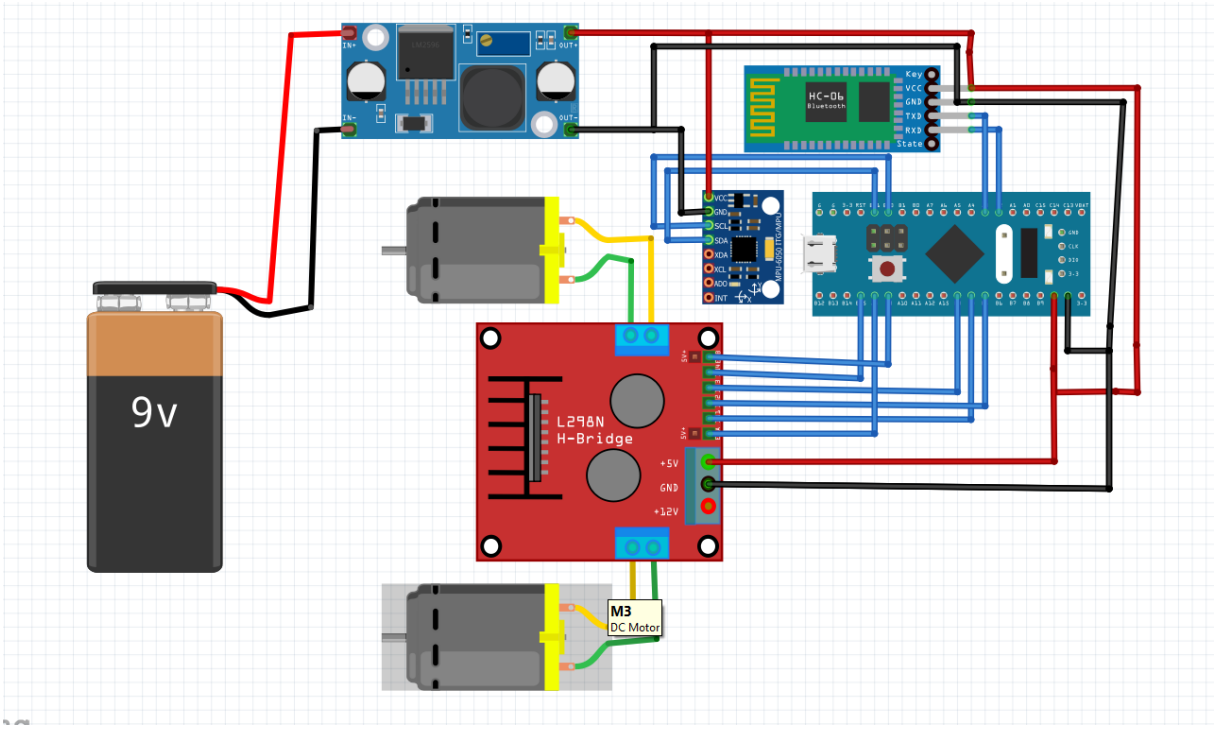
Bảng 4.2 Các thông số của STM32F103C8T6

Vi điều khiển	STM32F103C8T6
Điện áp hoạt động	5V – DC (chỉ được cấp qua cổng USB)
Tần số hoạt động	72 MHz
Dòng tiêu thụ	30mA
Điện áp vào khuyên dùng	2.2V – 3.6V
Số chân Digital I/O	24
Số chân Analog	10
Dòng tối đa trên mỗi chân I/O	30 mA
Dòng ra tối đa (5V)	500 mA
Dòng ra tối đa (3.3V)	50 mA
Bộ nhớ flash	64 KB
SRAM	20 KB
EEPROM	1 KB

4.3 Kết nối các thiết bị



Hình 4.9 Sơ đồ khối của mô hình



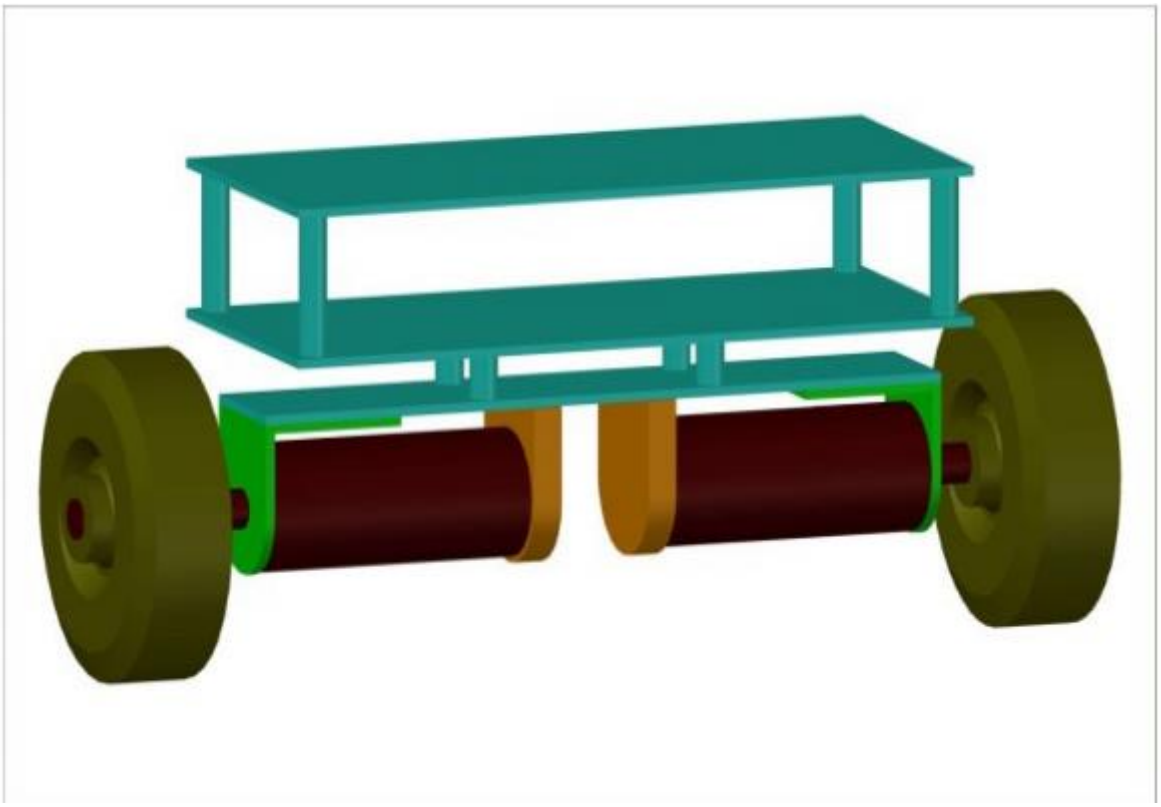
Hình 4.10 Sơ đồ nối dây của mô hình

Bảng 4.3 Nối dây cho mô hình

Chân STM32F103C8T6 Blue Pill ARM Cortex-M3.	Kết nối với các thiết bị khác
PA2 –UART TX	RX HC-05
PA3 – UART RX	TX HC-05
PB12	L298 In1
PB13	L298 In2
PB14	L298 In3
PB15	L298 In4
PA8 -PWM	L298 ENA
PA9 - PWM	L298 ENB
PB7 -I2C_SDA	SDA-MPU 6050
PB6-I2C_SCL	SCL- MPU 6050

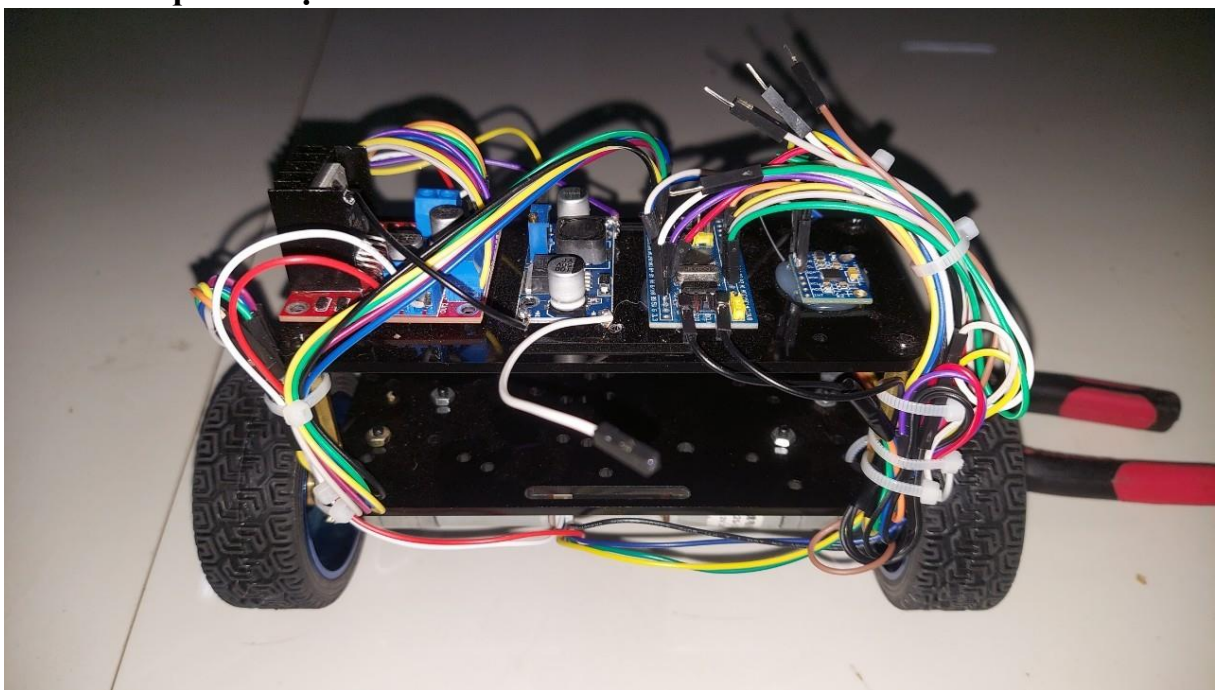
4.4 Thiết kế mô hình

Mô hình gồm các tấm mica hình chữ nhật

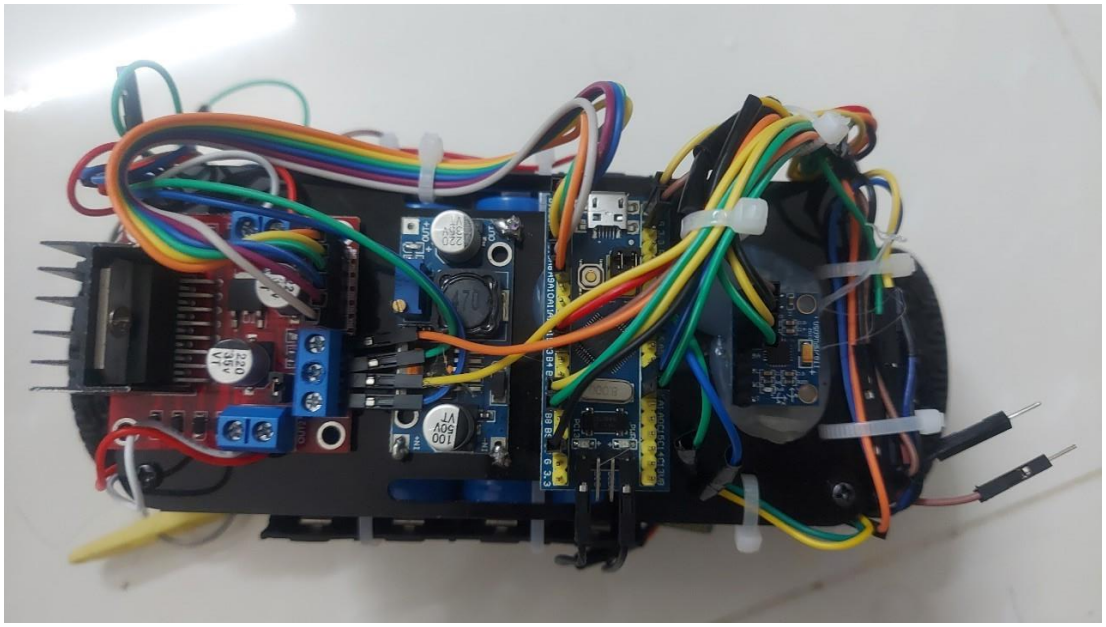


Hình 4.11 Khung xe mô hình.

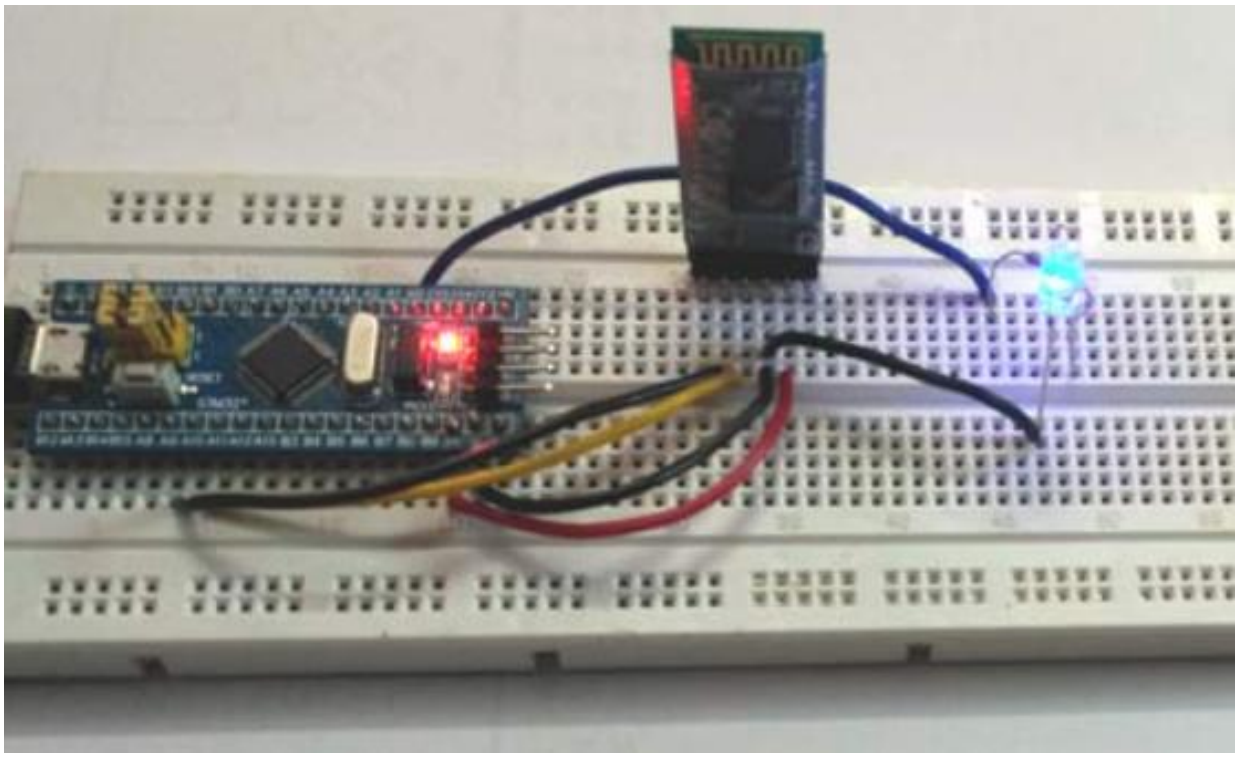
4.5 Kết quả chế tạo



Hình 4.12 Mặt trước của mô hình xe hai bánh tự cân bằng

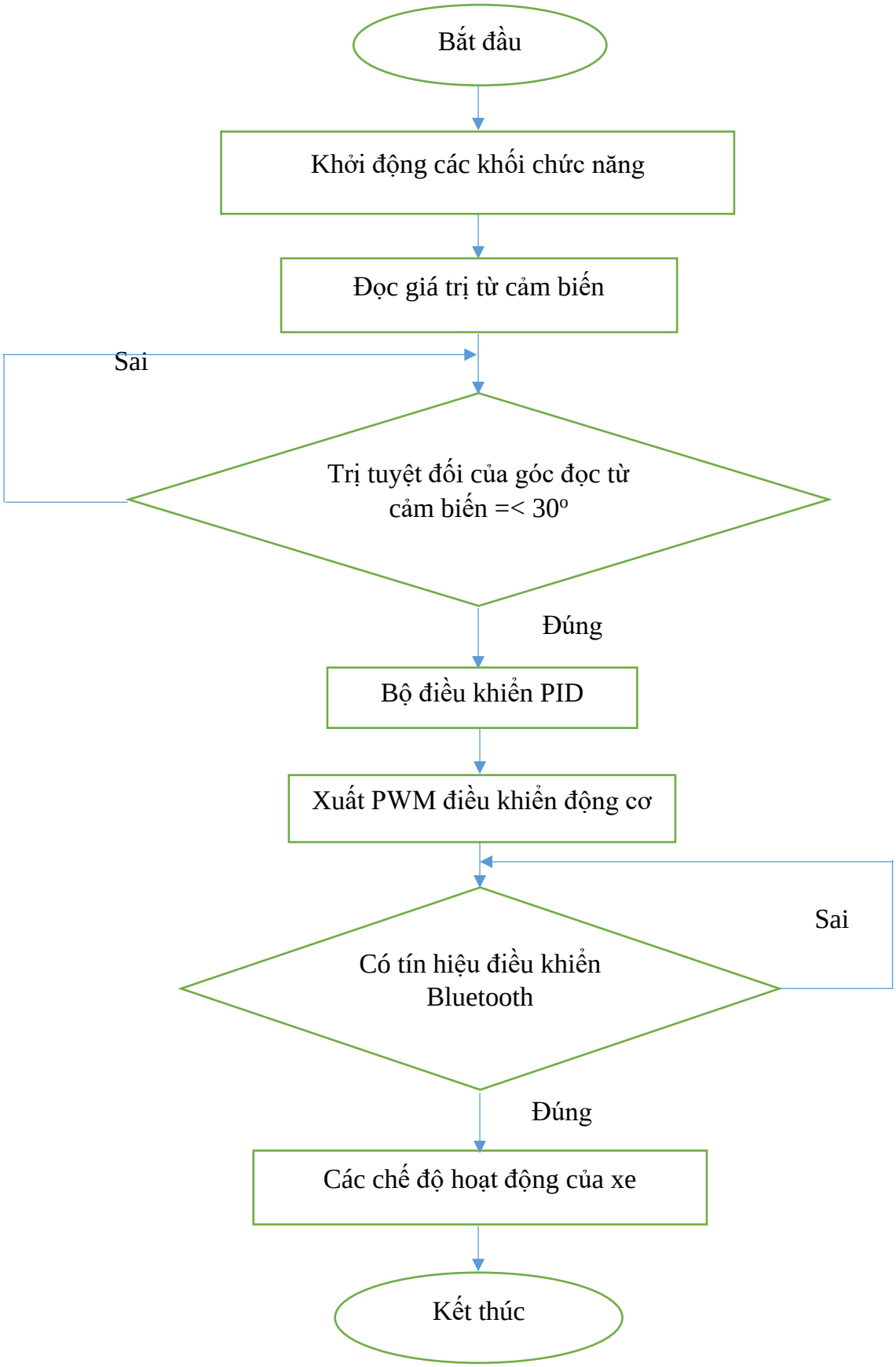


Hình 4.13 Mặt trên của mô hình



Hình 4.14 STM32F103C8T6 và Module Bluetooth HC-05

4.6 Lưu đồ giải thuật



Hình 4.15 Lưu đồ giải thuật

4.7 Đọc cảm biến

Trong quá trình đọc cảm biến cần phải cài đặt giá trị cho các thanh ghi của cảm biến và sau đây là một số thanh ghi thường dùng.

4.7.1 Các thanh ghi thường dùng

- Thanh ghi Who_Am_I:

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
75	117	-	WHO_AM_I[6:1]						-

Hình 4.16 Thanh ghi Who_Am_I

Bảy bit của thanh ghi chứa địa chỉ I2C của MPU. Có hai địa chỉ: 0x68 (nếu AD0 = 0) và 0x69 (nếu AD0 = 1) tùy thuộc vào chân AD0. Mặc định đối với board GY521 thì address = 0x68.

- Thanh ghi sample rate devider:

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
19	25	SMPLRT_DIV[7:0]							

Hình 4.17 Thanh ghi sample rate devider

Tốc độ lấy mẫu $= \frac{Gyroscope\ Output\ Rate}{1+SMPLRT\_DIV}$

- Thanh ghi Configuration:

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
1A	26	-	-	EXT_SYNC_SET[2:0]			DLPF_CFG[2:0]		

Hình 4.18 Thanh ghi configuration

EXT_SYNC_SET: cài đặt chế độ FSYNC.

DLPF_CFG: cài đặt cho DLPF (digital low pass filter).

Ở đây ta cài giá trị 0x00: disable SYNC và set filter 260Hz cho Acc; set filter 256Hz và tốc độ lấy mẫu 8kHz cho Gyro.

- Thanh ghi Gyroscope Configuration

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
1B	27	XG_ST	YG_ST	ZG_ST	FS_SEL[1:0]		-	-	-

Hình 4.19 Thanh ghi Gyroscope Configuration

FS_SEL: chọn full scale range cho Gyro

FS_SEL	Full Scale Range
0	± 250 °/s
1	± 500 °/s
2	± 1000 °/s
3	± 2000 °/s

Hình 4.20 Full scale của Gyro

Ở đây ta cài giá trị 0x00: set full scale range cho Gyro là ±250 deg/s.

- Thanh ghi Accelerometer Configulation:

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
1C	28	XA_ST	YA_ST	ZA_ST	AFS_SEL[1:0]		-		

Hình 4.21 Thanh ghi Accelerometer Configulation

AFS_SEL: chọn full scale range cho Acc

AFS_SEL	Full Scale Range
0	± 2*g*
1	± 4*g*
2	± 8*g*
3	± 16*g*

Hình 4.22 Full scale của Acc

Ở đây ta cài giá trị 0x00: set Accelerometer full scale range là ±2g.

- Thanh ghi Power Manegement

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
6B	107	DEVICE_RESET	SLEEP	CYCLE	-	TEMP_DIS	CLKSEL[2:0]		

Hình 4.23 Thanh ghi Power Manegement

CLKSEL: chọn nguồn clock cho MPU. Mặc định nguồn clock là dao động thạch anh nội 8MHz. Tuy nhiên để cải thiện tính ổn định, ta có thể dùng một trong các gyro x, y, z để làm clock chuẩn.

- Các thanh ghi Acc và Gyro dữ liệu:

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
3B	59	ACCEL_XOUT[15:8]							
3C	60	ACCEL_XOUT[7:0]							
3D	61	ACCEL_YOUT[15:8]							
3E	62	ACCEL_YOUT[7:0]							
3F	63	ACCEL_ZOUT[15:8]							
40	64	ACCEL_ZOUT[7:0]							

Hình 4.24 Thanh ghi Acc dữ liệu

ACCEL_XOUT: thanh ghi 16bit chứa giá trị là số bù 2, lưu giữ giá trị gia tốc trên trục X mới nhất.

ACCEL_YOUT: thanh ghi 16bit chứa giá trị là số bù 2, lưu giữ giá trị gia tốc trên trục Y mới nhất.

ACCEL_ZOUT: thanh ghi 16bit chứa giá trị là số bù 2, lưu giữ giá trị gia tốc trên trục Z mới nhất.

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
43	67	GYRO_XOUT[15:8]							
44	68	GYRO_XOUT[7:0]							
45	69	GYRO_YOUT[15:8]							
46	70	GYRO_YOUT[7:0]							
47	71	GYRO_ZOUT[15:8]							
48	72	GYRO_ZOUT[7:0]							

Hình 4.25 Thanh ghi Gyro dữ liệu

GYRO_XOUT: thanh ghi 16bit chứa giá trị là số bù 2, lưu trữ giá trị tốc độ góc xoay quanh trục X mới nhất.

GYRO_YOUT: thanh ghi 16bit chứa giá trị là số bù 2, lưu trữ giá trị tốc độ góc xoay quanh trục Y mới nhất.

GYRO_ZOUT: thanh ghi 16bit chứa giá trị là số bù 2, lưu trữ giá trị tốc độ góc xoay quanh trục Z mới nhất.

4.7.2 Quy trình đọc dữ liệu từ MPU6050

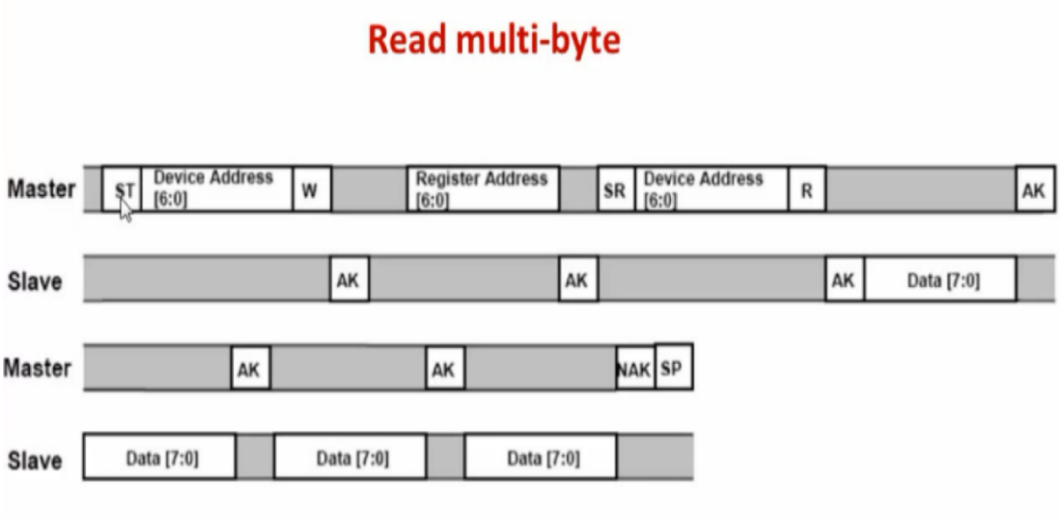
- Kiểm tra kết nối giữa Vi điều khiển và MPU6050

Đọc dữ liệu thanh ghi Who_Am_I. Kiểm tra có đúng địa chỉ MPU không, nếu đúng thì có nghĩa là kết nối I2C giữa MPU và STM32 đã được thiết lập.

- Cấu hình cho MPU6050:
  - Tốc độ lấy mẫu là 400Hz, thông số bộ lọc thông thấp cho Acc là 260Hz và Gyro là 256Hz, tầm full scale của Acc là ±2g và của Gyro là 250deg/s.
  - Cấu hình GyroX như một nguồn phát xung clock để tăng tính ổn định, disable tính năng đo nhiệt độ và chế độ sleep.
- Đọc giá trị thô giá trị Acc và Gyro
- Gộp 2 thanh ghi dữ liệu Acc/Gyro 8bit thành giá trị Acc/Gyro 16bit

4.7.3 Đọc giá trị thô từ cảm biến

Quá trình đọc dữ liệu trên bus i2c:



Hình 4.26 Khung truyền

Giải thích khung truyền (ở đây Master là STM32 và Slave là MPU-6050) :

- Master gửi tín hiệu start.
- Master gửi 7 bit địa chỉ thiết bị slave và bit write (bit 0) cuối cùng.
- Slave nhận được 8 bit ở trên thi gửi trả tin hiệu AK (bit 0).
- Master nhận được tín hiệu AK thì master gửi di 8 bit địa chỉ của thanh ghi có trong con slave.
- Sau khi nhận 8 bit địa chỉ thi slave gửi trả bit AK.
- Sau khi nhận bit AK master gửi tin hiệu restart rồi gửi lại địa chỉ thiết bị một lần nữa cùng bit read (bit 1).
- Sau khi nhận 8 bit trên thi slave gửi trả bit AK và ngay sau đó gửi 8 bit data.
- Nếu master muốn đọc tiếp dữ liệu thì sẽ gửi tiếp bit AK và slave se gửi tiếp 8 bit luôn mà ko cần tín hiệu AK ở trước nữa.
- Nêu ko muốn đọc dữ liệu nữa thì master sẽ gửi bit NAK (bit 1)sau đó gửi tín hiệu stop.

Giá trị vận tốc góc và gia tốc là số nguyên có dấu.Giá trị này sau khi thu được sẽ lưu vào 2 thanh ghi: thanh ghi Acc dữ liệu và thanh ghi Gyro dữ liệu.

4.7.4 Xử lý giá trị thu được

Quan sát các giá trị thu được ở hình 4.21 ta thấy đây là giá trị chưa hiệu chỉnh có nghĩa là mức zero ở đây chưa đúng là giá trị 0 mà là một giá trị khác 0. Do đó ta cần phải hiệu chỉnh cảm biến.

Trước tiên calib gyro do gyro có đặc tính hay bị trôi. Đặt cảm biến nằm ngang trên một mặt phẳng. Ta tạo 1 mảng 100 giá trị và bắt đầu lấy giá trị gyro 16bit. Khi có đủ 100 giá trị gyro 16bit, ta tính trung bình cộng giá trị ấy và thu được một giá trị tạm gọi là gyro_zero.

Do Accelerometer có đặc tính khá nhạy nên khi lắc mạnh cảm biến accelerometer cho ra kết quả sai lệch rất lớn nhưng bù lại góc tính từ accelerometer sẽ không bị trôi. Vì thế ta cũng calib accelerometer như đã làm với gyro. Sau khi đã có giá trị zero của gyro và acc ta xử lý tiếp như sau:

Acc: Lấy giá trị Acc thu về ta trừ đi mức zero của Acc để thu được giá trị thực chất. Trong đề tài cảm biến MPU6050 được đặt song song mặt đất và trục Y nằm song song trục nối 2 động cơ. Góc nghiêng của xe so với phương thẳng đứng được tính dựa vào công thức:

*accYAngle=(arctan2(accX,accZ) * RAD_TO_DEG.*

Ở đây accX và accZ là những giá trị có sau khi lấy AccX và AccZ thu về trừ cho mức zero. Hàm arctan2() trả về góc với đơn vị là radian trong khoảng từ – pi đến + pi. Do đó ta cộng thêm Pi rồi nhân hệ số RAD_TO_DEG để chuyển đổi góc đo được từ 0 đến 360 độ.

Gyro: Tương tự Acc giá trị thu về từ gyro ta trừ cho mức zero để thu được giá trị đúng. Sau đó ta chia cho độ nhạy của gyro là 131LBS(deg/s) để thu được tốc độ góc với đơn vị là deg/s.

*GyroYangle = GyroYangle+ GyroYrate * dt*

GyroYrate ở trên chính là giá trị thu được sau khi ta chia gyro cho 131.

4.8 Thiết kế bộ PID số

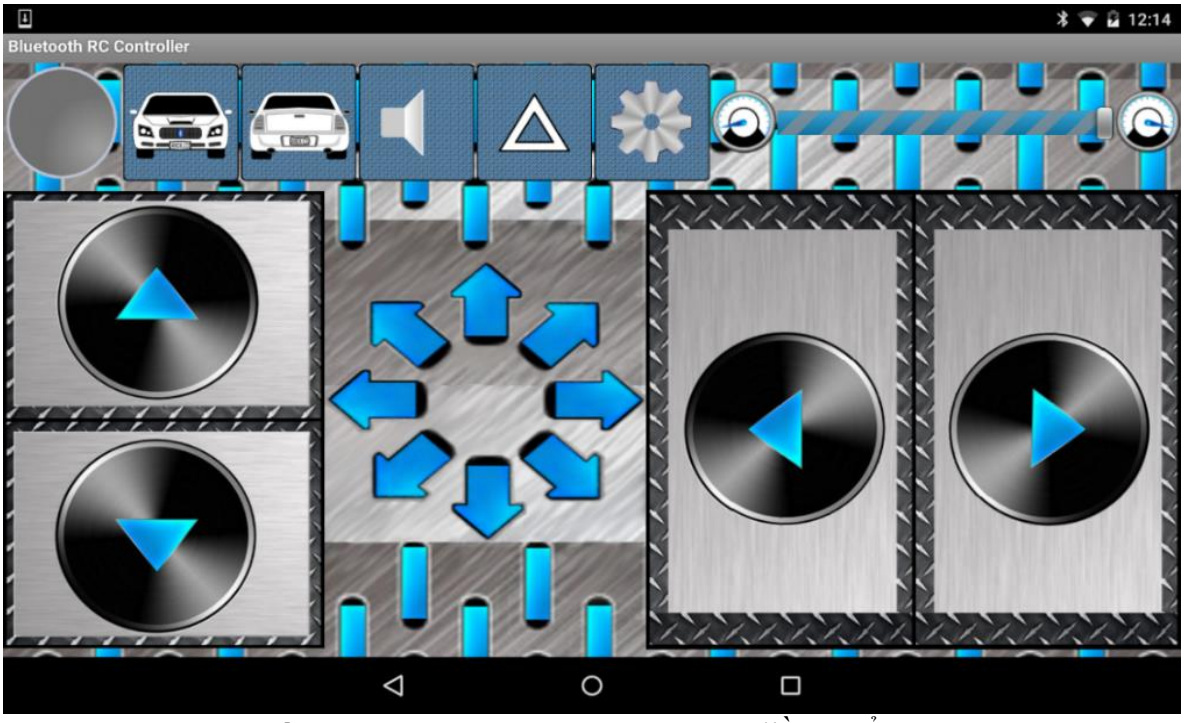
Ta thiết kế bộ PID số nhằm đưa giá trị của cảm biến về giá trị cân bằng. Đầu vào của bộ PID số là giá trị góc đã được xử lý sau khi thu được từ cảm biến. Kết hợp đầu vào và giá trị đặt là góc ở vị trí cân bằng ta lần lượt tạo ra các giá trị Eht, Edelta, Esum. Với khoảng thời gian lấy mẫu là 1s thì ta có: Eht là sai số thời điểm hiện tại. Eht bằng giá trị đặt trừ đầu vào. Edelta là giá trị có được sau khi lấy Eht trừ cho Eht của 1 giây trước đó. Cuối cùng Esum là giá trị cộng dồn sai số hiện tại.

Lần lượt nhân Kp, Kd, Ki cho Eht, Edelta, Esum. Tổng ba giá trị vừa nhân chính là output của bộ PID. Do PWM của động cơ là giá trị output của PID nên ta cần giới hạn output này trong khoảng giá trị từ -255 đến 255. Về cách tìm thông số PID có nhiều cách như đã được giới thiệu ở chương 2. Trong đề tài chúng tôi chọn phương pháp chỉnh định bằng tay để tìm thông số PID.

Sau đây là các bước để tìm thông số PID:

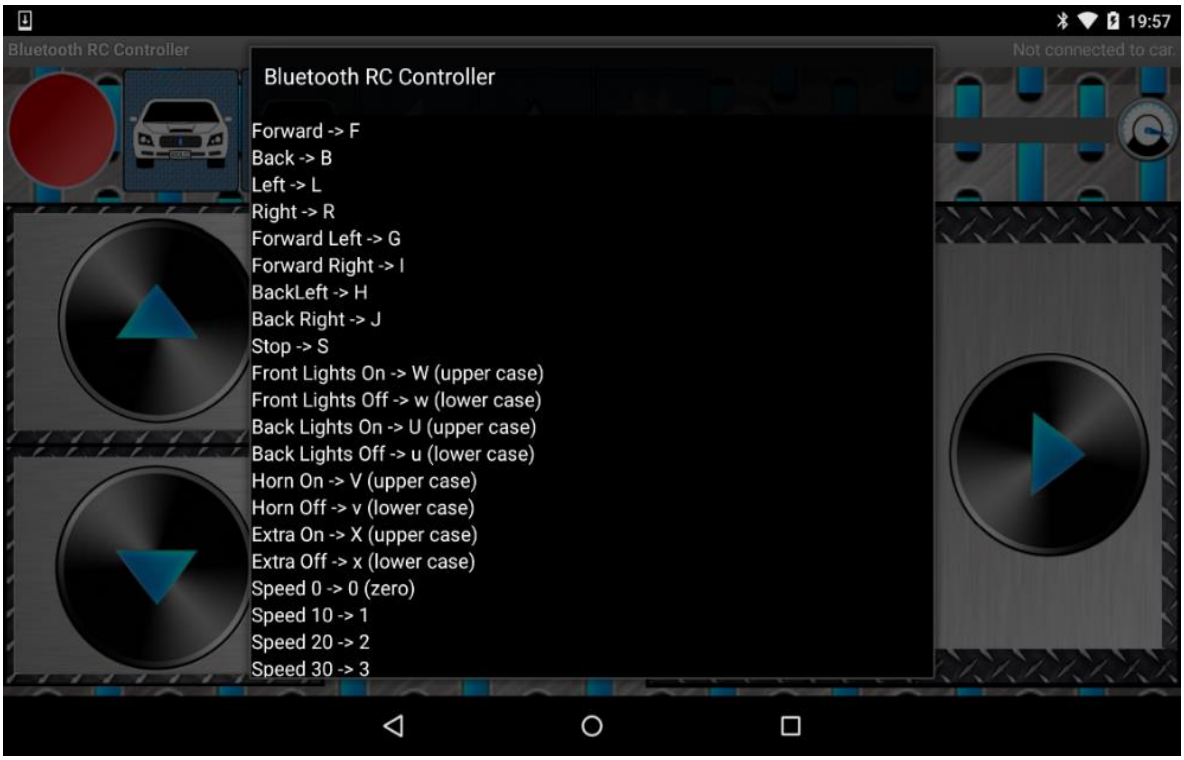
- Chọn Kp trước: thử bộ điều khiển P với mô hình, điều chỉnh Kp sao cho xe cân bằng, thời gian đáp ứng đủ nhanh, chấp nhận vọt lố nhỏ.
- Thêm thành phần D để loại vọt lố, tăng Kd từ từ cho đến khi xe cân bằng tốt hơn. Lúc này trạng thái sai số tĩnh sẽ xuất hiện.
- Thêm thành phần I để giảm trạng thái sai số tĩnh. Tăng I từ bé đến lớn để giảm trạng thái sai số tĩnh đồng thời không để cho vọt lố xuất hiện trở lại.

4.9 Các chế độ hoạt động của xe khi có tín hiệu bluetooth.



Hình 4.27 Giao diện của ứng dụng điều khiển xe

Mô hình được điều khiển chạy tới lui, xoay trái, xoay phải bởi ứng dụng chạy trên nền Android tên là Bluetooth RC controller.



Hình 4.28 Phần setting của ứng dụng Bluetooth RC controller

4.10 Điều khiển xe cân bằng

Trong đề tài này để điều khiển xe chạy tới ta không thể chỉ chú ý đến chiều quay và vận tốc của động cơ mà còn phải lưu ý tới việc giữ thăng bằng cho xe. Dựa trên nguyên lý hoạt động của xe là khi góc nghiêng của xe so với phương thẳng đứng lớn hơn 0 thì hai động cơ sẽ chạy tới trước nhằm giúp xe trở lại vị trí cân bằng. Vì thế muốn xe chạy tới thì chúng tôi sẽ phát tín hiệu điều khiển làm thay đổi góc nghiêng của xe mà cụ thể là cho nó lớn hơn 0.

Dựa trên chức năng thay đổi tốc độ của ứng dụng Bluetooth RC Controller ta tạo một biến tốc độ có giá trị từ 1 đến 9 tương ứng với 9 cấp tốc độ của ứng dụng Bluettooth RC Controller. Sau đó ta lấy biến tốc độ này nhân với một hằng số để tạo ra giá trị góc nghiêng so với phương thẳng đứng giúp xe di chuyển. Kế đến là điều khiển cho xe lui. Cũng tương tự như cho xe chạy tới ta tạo biến tốc độ và nhân nó với một hằng số âm. Trong đề tài này chúng tôi lấy biến tốc độ nhân với 0,2 hoặc -0,2 cho trường hợp xe tới hoặc lùi.

Bây giờ sau khi xe đã chạy tới lui được thì vấn đề đặt ra là làm sao cho xe quẹo trái cũng như quẹo phải. Như ta đã biết muốn xe quẹo phải thì tất nhiên là vận tốc của bánh trái phải lớn hơn vận tốc bánh phải. Ngược là cho trường hợp muốn xe quẹo trái. Vì vậy muốn điều khiển cho xe chuyển hướng ta sẽ tạo ra một giá trị PWM cộng thêm bằng cách dùng biến tốc độ ở phần điều khiển xe tới lui nhân với một hằng số dương. Muốn xe quẹo bên nào ta cho giá trị PWM cộng thêm của bên đó bằng 0 và bên còn lại sẽ là giá trị mà ta vừa tính ở trên.

4.11 Khảo nghiệm tính ổn định của mô hình

Bảng 4.4 Bảng khảo nghiệm

STT	**P**	**I**	**D**	Khoảng dịch chuyển của mô hình (cm)	Góc điều khiển	Trạng thái của mô hình
1	26	390	1.18	2	1.5^o	Di chuyển được
2	28	230	1.14	2	1.5^o	Di chuyển được
3	30	350	1.18	1	1.5^o	Di chuyển được
4	34	440	1.182	Dao động qua lại quanh vị trí cân bằng.	1.5^o	Di chuyển được

Các số liệu ở cột khoảng cách dịch chuyển của mô hình được đo trong khoảng thời gian 1 phút, giá trị này bằng khoảng cách từ trục nối hai bánh xe tới vị trí ban đầu. Góc điều khiển là giá trị người điều khiển đặt ra đề mô hình có thể di chuyển mà vẫn cân bằng.

Qua các kết quả trong bảng khảo nghiệm chúng em thấy với các thông số PID lần lượt là 34 – 440 – 1.182 thì mô hình dao động qua lại quanh vị trí cân bằng và có thể điều khiển di chuyển tới lui được.

Chương 5:

KẾT LUẬN VÀ ĐỀ NGHỊ

5.1 Những kết quả đạt được

- Thiết kế được mô hình xe tự cân bằng trên hai bánh.
- Xác định được mô hình toán học và hàm trạng thái của mô hình.
- Giao tiếp máy tính để hiệu chỉnh cảm biến và các thông số PID.
- Lập trình phần mềm cho vi điều khiển giúp mô hình cân bằng.
- Điều khiển mô hình di chuyển tới lui trái phải được trên địa hình phẳng.
    - Nhược điểm, những vấn đề chưa giải quyết được:
- Mô hình chưa di chuyển được trên những địa hình có độ dốc cao.
- Mô hình còn rung khi di chuyển.
- Chưa xử lý được khi góc lệch lớn.

5.2 Đề nghị

- Do chuẩn bị chưa được tốt nên mô hình còn nhiều khuyết điểm. Vì vậy chúng em có những đề nghị sau để mô hình hoạt động tốt hơn:
    - Cải tiến và thiết kế mô hình xe scooter cỡ lớn có thể di chuyển và vận tải như một phương tiện giao thông.
    - Nghiên cứu sử dụng giải thuật LQR thay cho PID.

TÀI LIỆU THAM KHẢO

http://codientu.org/threads/6894/

http://www.hocavr.com/index.php/app/dcservo#5-PID

http://tienanh.info/hoc-tap/co-dien-tu/dieu-khien-xe-2-banh-tu-can-bang-01-05-2014.html

https://bayesianadventures.wordpress.com/2013/10/27/balance-good-everything-good/

http://ctms.engin.umich.edu/CTMS/index.php?example=InvertedPendulum&section=ControlPID

http://brettbeauregard.com/blog/2011/04/improving-the-beginners-pid-introduction/

http://blog.tkjelectronics.dk/2012/09/a-practical-approach-to-kalman-filter-and-how-to-implement-it

http://www.kickstarter.com/projects/tkjelectronics/balanduino-balancing-robot-kit

http://www.khoahocphothong.com.vn/news/detail/11059/che-tao-hoan-chinh-xe-hai-banh-tu-can-bang.html