**LỜI MỞ ĐẦU**

Đất nước ta đang trong tiến trình phát triển công nghệ ngày một nhanh chóng cùng với sự phát triển của nền công nghiệp 4.0 lan tỏa trên toàn thế giới. Điều này đòi hỏi một lượng lớn kỹ sư nói chung và kỹ sư cơ điện tử nói riêng cống hiến sức mình cho sự phát triển của đất nước trong thời đại mới. Trải qua bốn năm ngồi trên ghế nhà trường với trăn trở làm sao để thực hiện được những điều Bác dặn em đã không ngừng học hỏi, tìm tòi góp nhặt những kiến thức quý báu mà các thầy các cô đã giảng dạy để hôm nay đây em có thể hoàn thành đồ án này. Như là một lời chứng minh sắt đá rằng em đã sẵn sàng bước vào công cuộc xây phát triển và xây dựng đất nước, góp một phần nhỏ của mình vào công cuộc đưa đất nước sánh vai với các cường quốc năm châu.

**LỜI CẢM ƠN**

“Không thầy đố mày làm lên” – Đúng như câu tục ngữ, em đã không thể hoàn thành đề tài của luận văn này nều không có sự giúp đỡ của các thầy các cô cũng như người thân và bạn bè xung quanh. Do vậy với sự trân trọng và lòng cảm kích em xin gửi lời cảm ơn chân thành đến ThS. Hoàng Văn Mạnh, thầy không chỉ là người đã hướng dẫn em trong suốt quá trình hoàn thành đồ án tốt nghiệp mà còn là người cho em những động lực rất lớn để em không ngừng sáng tạo và phát triển sản phẩm, nâng cao chất lượng nghiên cứu.

Xin cảm ơn thầy cô trong khoa Cơ học kỹ thuật và cũng như thầy cô trong toàn trường Đại học Công nghệ - ĐHQGHN đã truyền đạt nhiều kiến thức quý báu trong suốt 4 năm học, những kiến thức đó là chìa khóa giúp em thực hiện được những hoài bão trong cuộc sống và là hành trang cho em bước vào thời kỳ phát triển mới của đất nước.

Cảm ơn Nguyễn Xuân Tiến, người bạn đã sát cánh cùng em qua nhiều khó khăn trong quá trình học tập và tìm hiểu kiến thức khoa học. Tiến cũng là người nhiệt tình giúp đỡ em trong nhiều phần khó khăn của đề tài.

Ngoài ra, em xin cảm ơn bố, mẹ và em trai của em những người đã, đang và luôn là chỗ dựa tinh thần cho em mọi lúc mọi nơi.

Trong quá trình làm luận văn và thực hiện đề tài, dù đã cố gắng tuy nhiên không thể tránh được sai sót không đáng có. Em rất mong thầy cô chỉ bao thêm giúp em hoàn thành và đạt kết quả tốt hơn nữa. Em xin chân thành cảm ơn

|  |  |  |
| --- | --- | --- |
|  |  | *Hà Nội, ngày 23 tháng 4 năm 2018*  Sinh viên |

**TÓM TẮT**

**Tóm tắt:** Nội dung nghiên cứu của đề tài bao gồm 2 phần chính:

Thứ nhất, xe tự cân bằng có khả năng tự đứng thăng bằng trên hai bánh tại mặt phẳng và mặt nghiêng. Xe tự cân bằng này hoàn toàn có thể được coi là một hệ thống cơ điện tử hoàn chỉnh bao gồm cảm biến tiếp nhận thay đổi môi trường, vi điều khiển tiếp nhận thông tin từ cảm biến và gửi tín hiệu điều khiển một hệ cơ học. Vi điều khiển 32 bit, một dòng vi điều khiển mạnh mẽ so với vi điều khiển 8 bit truyền thống, được sử dụng làm phần vi điều khiển xử lý cho xe hai bánh tự cân bằng.

Thứ hai, phần mềm điều khiển, theo dõi, cài đặt cho xe sử dụng được trên thiết bị di động. Có khả năng điều khiển xe di chuyển và ghi lại giao động của xe tự cân bằng. Việc xây dựng phần mềm di động để quản lý thiết bị là nhu cầu tất yếu trong thời đại công nghiệp 4.0 cùng với sự phát triển của IoT. Phần mềm đi kèm với thiết bị sẽ giúp tăng giá trị thiết bị lên rất nhiều.

***Từ khóa:*** *Xe tự cân bằng, cảm biến gia tốc 3 trục, vi điều khiển 32 bit, hệ cơ điện tử*

**LỜI CAM ĐOAN**

Em xin cam đoan đây là công trình tìm hiểu, nghiên cứu và chế tạo độc lập của riêng em. Các số liệu sử dụng phân tích trong luận án đều có nguồn gốc rõ rang, đã công bố theo đúng quy định. Các kết quả, thiết kế trong luận án do em tự tìm hiểu phân tích một cách trung thực, khách quan và chưa được công bố trong các công trình khác. Nếu không đúng như đã nếu trên, em xin hoàn toàn chịu trách nhiệm về đề tài của mình.

|  |  |  |
| --- | --- | --- |
|  |  | *Hà Nội ngày 23, tháng 04, năm 2018*  Sinh viên  PHẠM TIẾN THÀNH |

MỤC LỤC

DANH MỤC HÌNH ẢNH

**CÁC KÝ HIỆU VIẾT TẮT**

DMP Digital Motion Processor

ADC Analog Digital Converter

dps degree per second

FIFO First in first out

CMOS

MEMS

APB

PWM

USB

CAN

RPM

**MỞ ĐẦU**

* ***Tính cần thiết của đề tài***

Ngày nay hệ thống cơ điện tử đã và đang xuất hiện trong mọi mặt của cuộc sống. Các công ty đang hướng đến trải nghiệm của người dùng bằng cách sử dụng các hệ thống cơ điện tử trong các ứng dụng của họ. Máy giặt có thể thay đổi nhiệt độ nước để phù hợp với từng loại vải, điều hòa có thể tự tìm kiếm người sử dụng và hướng dòng khí tươi mát đến họ,... tất cả đều là kết quả của việc ứng dụng hệ thống cơ điện tử vào sản phẩm.

Nhu cầu của xã hội và người sử dụng ngày càng cao và tinh vi, đòi hỏi người kỹ sư cũng như các nhà sản xuất ngày càng phải sử dụng nhiều công nghệ mới hơn, nhanh hơn, mạnh mẽ hơn vào quá trình phát triển sản phẩm. Vì lý do này mà hiện này các cảm biến không ngừng phát triển và có thêm nhiều loại cảm biến mới, các vi xử lý 32 bit cũng được phát triển và cải thiện đáng kể về tốc độ so với dòng vi điều khiển 8 bit truyền thống. Điều này là cơ hội đối với các kỹ sư phát triển phần mềm giúp họ có thể tạo ra được các sản phẩm đáp ứng được người sử dụng, nhưng đó cũng là thách thức đối với các kỹ sư bởi tính mới và phương pháp sử dụng.

Do đó đề tài “Nghiên cứu, thiết kế xe tự cân bằng sử dụng cảm biến gia tốc 3 trục và vi điều khiển 32 bit” bằng việc sử dụng cảm biến gia tốc 3 trục và dòng vi điều khiển mới 32 bit sẽ là một ví dụ thiết thực về việc thiết kế hệ thống cơ điện tử và có thể ứng dụng cho các hệ thống cơ điện tử khác.

* ***Đối tượng và phương pháp nghiên cứu***

Xe tự cân bằng chính là một hệ thống cơ điện tử, các đối tượng trong hệ thống cơ điện tử cơ bản gồm có: cảm biến, vi điều khiển và phần cơ học chấp hành. Trong khóa luận này em tập chung chính vào 2 đối tượng của hệ thống cơ điện tử và 1 đối tượng tổng quan đó là:

* Cảm biến gia tốc 3 trục
* Vi điều khiển 32 bit
* Lý thuyết xe tự cân bằng
* ***Nội dung nghiên cứu***

CHƯƠNG 1. TÌM HIỂU VỀ HỆ THỐNG XE HAI BÁNH TỰ CÂN BẰNG

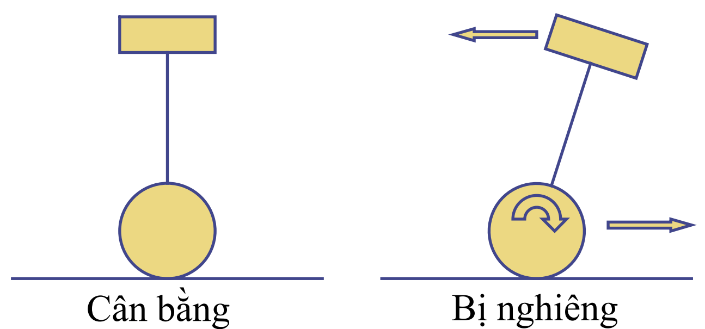
CHƯƠNG 2: NGHIÊN CỨU CẢM BIẾN GIA TỐC 3 TRỤC MPU6050

CHƯƠNG 3. NGHIÊN CỨU VI ĐIỀU KHIẾN 32 BIT STM32F103C8T6

CHƯƠNG 4: THIẾT KẾ XE TỰ CÂN BẰNG

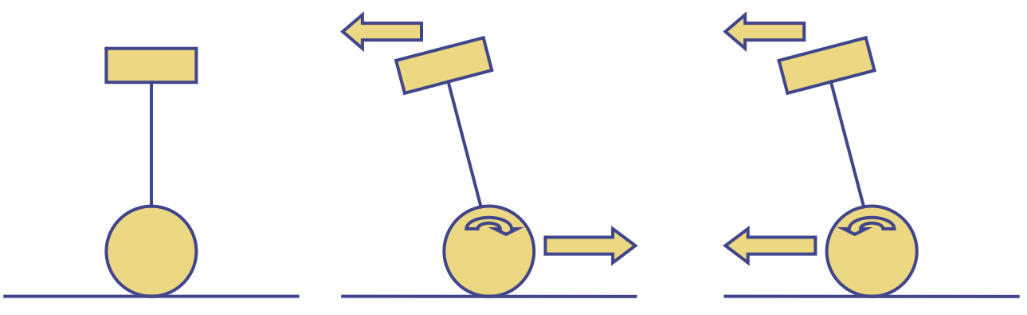
**CHƯƠNG 1. TÌM HIỂU VỀ HỆ THỐNG XE HAI BÁNH TỰ CÂN BẰNG**

**1.1. Thế nào là xe hai bánh tự cân bằng**



Mô tả nguyên lý giữ thăng bằng

Đối với các loại xe ba hay bốn bánh thì việc giữ thăng bằng và ổn định là nhờ trọng tâm chủa chúng nằm trong bề mặt chân đế do các bánh xe tạo nên. Còn đối với xe hai bánh tự cân bằng (hai bánh song song) để giữ thăng bằng cho xe trọng tâm của xe phải cần được giữ nằm cân bằng ngay giữa hai bánh xe. Giống như việc chúng ta giữ một cây gậy thẳng đúng cân bằng trong lòng bàn tay vậy.



Mô tả cách thức di chuyển

Tuy nhiên việc xác định trọng tâm của xe là tương đối khó khăn do hình dạng xe và các vật gắng trên xe có phần phức tạp. Tuy nhiên chúng ta hoàn toàn có thể xác định được góc giữa mặt phẳng sàn nhà và xe cũng như trọng lực. Do đó, thay vì đi tìm trọng tâm nằm giữa hai bánh ta cần giữ cho xe luôn thẳng đứng, vương góc với mặt sàn.

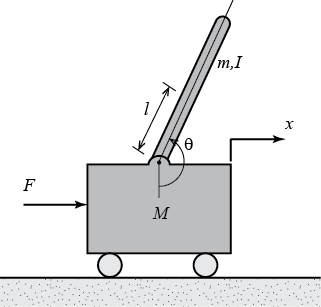
Đi kèm với đó là việc khi ta giữ một góc giữa xe và mặt sàn khác thì xe có xu hướng tiến hoặc lùi (tùy theo quy ước) về phía trước hoặc sau. Đây chính là nguyên lý để vân hành xe di chuyển. Để dừng lại thì ta chỉ cần điều chỉnh lại góc về giá trị .

**1.2. Phương trình toán học, phương pháp tính toán động lực học của xe hai bánh tự cân bằng**

Có nhiều phương pháp dùng 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.

Bên cạnh các ưu điểm này, nó vẫn có nhược điểm là phải tuyến tính hóa tính toán tại vị trí góc . Tuy nhiên việc này không trầm trọng trong mô hình của đề tài, vì mô hình chỉ hoạt động xung quanh vị trí ().

* *Nền tảng mô hình toán học, phương trình con lắc ngược*



Mô hình con lắc ngược

Ta xem xét mô hình toán học của con lắc ngược với các tham số như sau:

() Khối lượng xe (kg)

() Khối lượng con lắc (kg)

() Ma sát của xe (N)

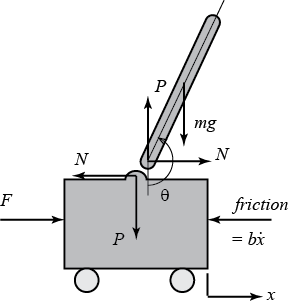
() chiều cài ½ con lắc (m)

() Momen quan tính của con lắc (Nm)

() Lực tác độn vào xe (N)

() Vị trí của xe (m)

() Góc của con lắc so với phương thẳng đứng (rad)



Phân tích lực trên xe và trên con lắc

Tổng hợp các lực trong sơ đồ tác dụng lên xe theo hướng ngang, ta sẽ có được phương trình chuyển động sau đây.

|  |  |  |
| --- | --- | --- |
|  |  | (1) |

Tổng hợp các lực trong sơ đồ tác dụng lên con lắc theo hướng ngang, ta sẽ có biểu hiện sau đây cho lực phản ứng .

|  |  |  |
| --- | --- | --- |
|  |  | (2) |

Thay thế phương trình này vào phương trình đầu tiên, ta sẽ có được một trong hai phương trình điều khiển cho hệ thống này.

|  |  |  |
| --- | --- | --- |
|  |  | (3) |

Để có được phương trình thứ hai của chuyển động cho hệ thống này, ta tổng hợp các lực vuông góc với con lắc. Giải quyết hệ thống dọc theo phương này bằng cách đơn giản hóa toán học. Ta sẽ nhận được phương trình sau đây.

|  |  |  |
| --- | --- | --- |
|  |  | (4) |

Để thoát khỏi và thuật ngữ trong phương trình ở trên, tổng hợp momen tại khối tâm của con lắc để có được phương trình sau đây.

|  |  |  |
| --- | --- | --- |
|  |  | (5) |

Kết hợp hai biểu thức cuối cùng này, bạn sẽ có được phương trình điều khiển thứ hai.

|  |  |  |
| --- | --- | --- |
|  |  | (6) |

Vì các kỹ thuật phân tích và kiểm soát mà chúng ta sẽ sử dụng trong ví dụ này chỉ áp dụng cho các hệ thống tuyến tính nên tập hợp các phương trình này cần được tuyến tính hóa. Cụ thể, chúng ta sẽ linearize phương trình về vị trí cân bằng trở lên,, và sẽ cho rằng hệ thống nằm trong một khu phố nhỏ của equillbrium này. Giả định này phải có giá trị hợp lý vì được kiểm soát mà chúng tôi mong muốn rằng con lắc không lệch hơn 20 độ so với vị trí đứng lên trên. Để cho đại diện cho độ lệch của vị trí của bàn đạp từ cân bằng, nghĩa là,. Một lần nữa giả sử một độ lệch nhỏ () từ cân bằng, chúng ta có thể sử dụng các phép tính xấp xỉ nhỏ sau đây của các hàm phi tuyến trong các phương trình hệ của chúng ta:

|  |  |  |
| --- | --- | --- |
|  |  | (7) |
|  |  | (8) |
|  |  | (9) |

Sau khi thay thế xấp xỉ trên vào phương trình điều phối phi tuyến của chúng ta, chúng ta đi đến hai phương trình tuyến tính của chuyển động. chú thích đã được thay thế cho đầu vào

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | (10) | | |
|  |  | | (11) |

**1.3. Tại sao phải thiết kế xe hai bánh tự cân bằng**

Các dòng xe nhiều hơn 3 bánh trong thực tế là chủ yếu, do việc giữ thăng bằng là đơn giản. Tuy nhiên với nhiều trường hợp như nghiên cứu các robot di chuyển trong địa hình hẹp thì việc sử dụng các robot tự cân bằng 2 bánh mang lại ưu điểm rất lớn ví dụ như xe có thể lập tức quay ngoắt 180 độ trong khoảng không hẹp.

Hơn nữa với địa hình gồ ghề phức tạp như dốc lớn thì việc trọng tâm của các xe lớn hơn 3 bánh lại gặp khó khăn. Ngược lại, các xe hai bánh tự cân bằng lại rất linh động trong việc giữ thăng bằng khi di chuyển trên địa hình phức tạp.

Không chỉ thế việc phát triển xe hai bánh tự cân bằng còn là ý tưởng cho việc thiết kế các robot giống người di chuyển và giữ cân bằng trên hai chân. Vì vậy việc phát triển và thiết kế xe hai bánh tự cân bằng có một tầm quan trọng nhất định trong sự phát triển của khoa học công nghệ hiện nay của nước ta cũng như trên thế giới.

**1.4. Một số hình ảnh về xe hai bánh tự cân bằng đã được tiến hành trong và ngoài nước**

//Viết thêm

**CHƯƠNG 2: NGHIÊN CỨU CẢM BIẾN GIA TỐC 3 TRỤC MPU6050**

**2.1. Cảm biến là gì? Tầm quan trọng của cảm biến**

Cảm biến là các phần tử nhạy cảm dùng để biến đổi các đại lượng đo lường, kiểm tra hay điều khiển từ dạng này sang dạng khác thuận tiện hơn cho việc tác động của các phần tử khác. Cảm biến là một thiết bị chịu tác động của đại lượng cần đo m không có tính chất điện và cho một đặc trưng mang bản chất điện (như điện tích, điện áp, dòng điện, trở kháng) kí hiệu là s có . Cảm biến thường dùng ở khâu đo lường và kiểm tra. Căn cứ theo dạng đại lượng đầu vào người ta phân ra các loại cảm biến như: cảm biến chuyển dịch thẳng, chuyển dịch góc quay, tốc độ, gia tốc, mô men quay, nhiệt độ, áp suất, quang, bức xạ,...

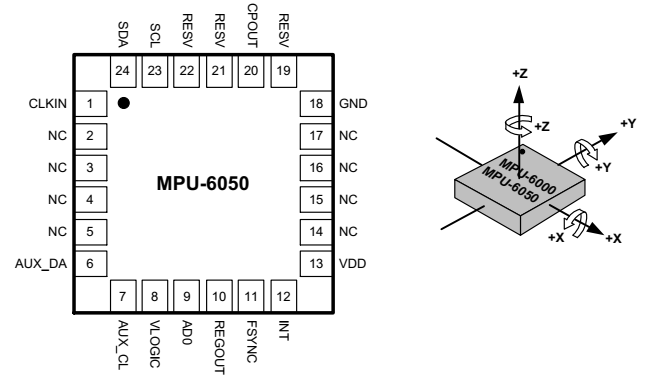
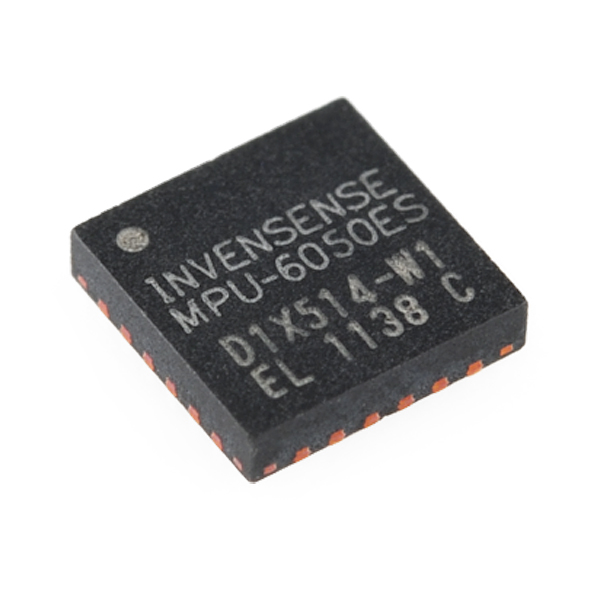
Hiện nay công nghệ về cảm biến được ứng dụng rất nhiều trong khoa học kỹ thuật, cảm biến chính là công cụ không thể thiếu để số hóa các đại lượng đo lường trong cuộc sống, thay thế cho các thiết bị đo tương tự truyền thống ví dụ như: Cảm biến nhiệt độ, độ ẩm (DHT11, DHT22) thay thế cho nhiệt kế thủy ngân, nhiệt kế lò xo; Cảm biến siêu âm (SRF04, SRF05) giúp xác định độ xa thay thế cho một số thước truyền thống; Cảm biến nhịp tim sử dụng ánh sáng thay thế cho việc đo nhịp tim thông thường,... Không những thế các cảm biến còn giúp xác định được thêm những đại lượng tương đối khó đo được trong tự nhiên như cảm biến ánh sáng (LS6b), cảm biến vân tay,... Và ưu điểm mà cảm biến mang lại là rất lớn không những giúp số hóa các đại lượng để thu thập phân tích một cách dễ dàng mà tốc độ làm việc và độ chính xác của các cảm biến hiện nay là rất cao. Điều này khiến tầm quan trọng của cảm biến trong các ngành khoa học kỹ thuật nói chung và ngành cơ điện tử nói riêng là không thể không nhắc đến. Việc nghiên cứu về cảm biến để hiểu và sử dụng thành thạo một cảm biến đóng vai trò rất lớn trong thành công của một đề tài cơ điện tử.

**2.2. Cảm biến gia tốc 3 trục MPU6050**



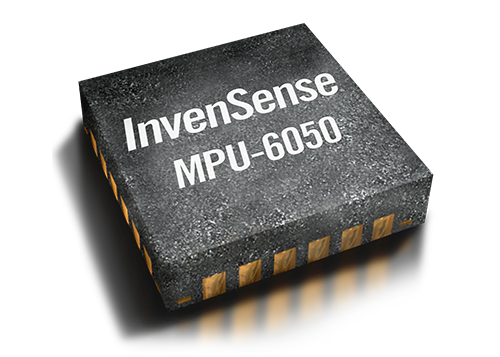
Ứng dụng cảm biến chuyện động trong trò chơi thực tế ảo

Cảm biến chuyển động từ lâu nay được các nhà sáng chế smartphone, table,... coi là phần không thể thiếu trong các thiết bị của họ bởi giá trị to lớn mà nó mang lại đến trải nghiệm của người sử dụng. Trong smartphone, cảm biến chuyển động được các ứng dụng với các chức năng như là nhận biển cử chỉ và điều khiển thiết bị, tăng cường điều khiển cho các trò chơi chuyển động thực, chụp ảnh toàn cảnh, ứng dụng đi bộ và chỉ đường phương tiện,... Với khả năng theo dõi chính xác chuyển động người dùng của nó, công nghệ theo dõi chuyển động có thể biến các thiết bị di động cầm tay hay máy tính bảng trở thành thiết bị 3D thông minh được sử dụng trong nhiều ứng dụng từ sức khỏe, thể dục đến các dịch vụ dựa trên vị trí. Các yêu cầu chính cho cảm biến chuyển động là nhỏ gọn, tiêu thụ năng lượng thấp, độ chính xác cao trong nhiều lần sử dụng, chống chịu được sự va đập và đơn giản trong việc lập trình – tất cả những điều này đều cần nằm trong một mức giá tiêu dùng thấp.



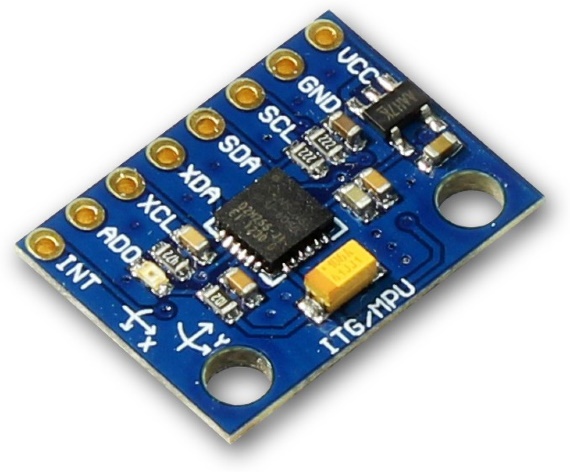
Cảm biến MPU6050 và sơ đồ chân

Cảm biến MPU6050 một cảm biến của hãng InvenSense là thiết bị tích hợp 6 trục cảm biến chuyển động đầu tiên trên thế giới nó phối hợp 3 trục con quay hồi chuyển, 3 trục gia tốc kế và bộ xử lý chuyển động số (DMP) trong một chíp nhỏ . Với đường truyền chuyên dụng của nó, nó trực tiếp chấp nhận đầu vào từ la bàn 3 trục bên ngoài để cung cấp đầu ra 6 trục hoàn chỉnh. Tích hợp phần hiệu chỉnh thời gian chạy cho phép nhà sản xuất loại bỏ được sự phức tạp, chi phí và mức độ tích hợp của các thiết bị rời rạc, đảm bảo chuyển động tối ưu hiệu suất cho người sử dụng. Cảm biến MPU6050 cũng được thiết kế để giao tiếp với nhiều cảm biến kỹ thuật số phi quán tính như cảm biến áp suất, trên cổng phụ trợ của nó.



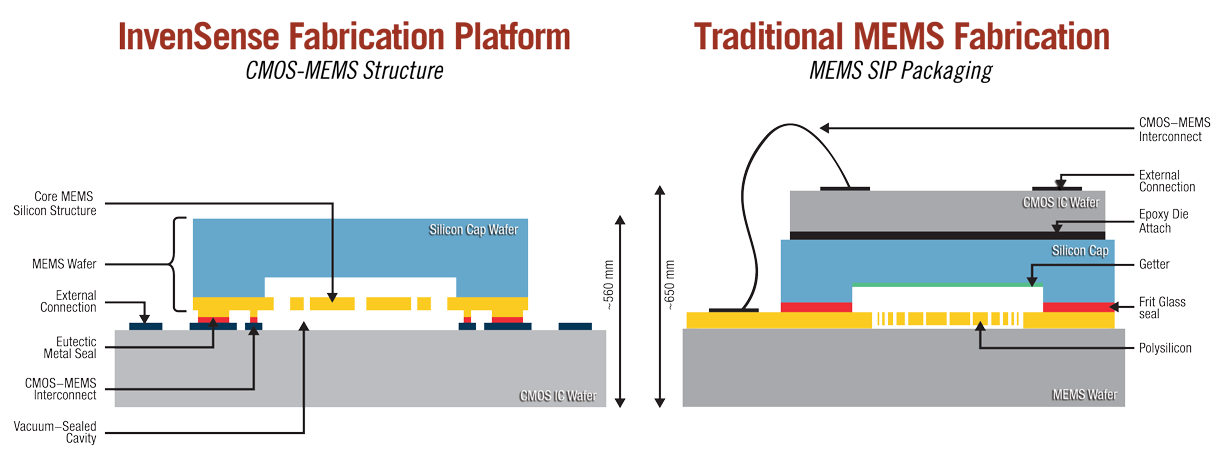
Mô hình thiết kế MPU6050

Cảm biến MPU6050 có 3 bộ chuyển đổi tương tự sang số (ADCs) dùng cho số hóa tín hiệu ra con quay và 3 bộ ADCs dùng cho số hóa tín hiệu ra gia tốc kế. Để việc theo dõi cả chuyển động nhanh và chuyển động chậm đều được chính xác, lập trình viên có thể đặt được khoảng giá trị lấy mẫu của con quay hồi chuyển trong khoảng các giá trị và (dps), tương tự lập trình viên có thể đặt được khoảng giá trị vận tốc của gia tốc kế trong khoảng giá trị và .



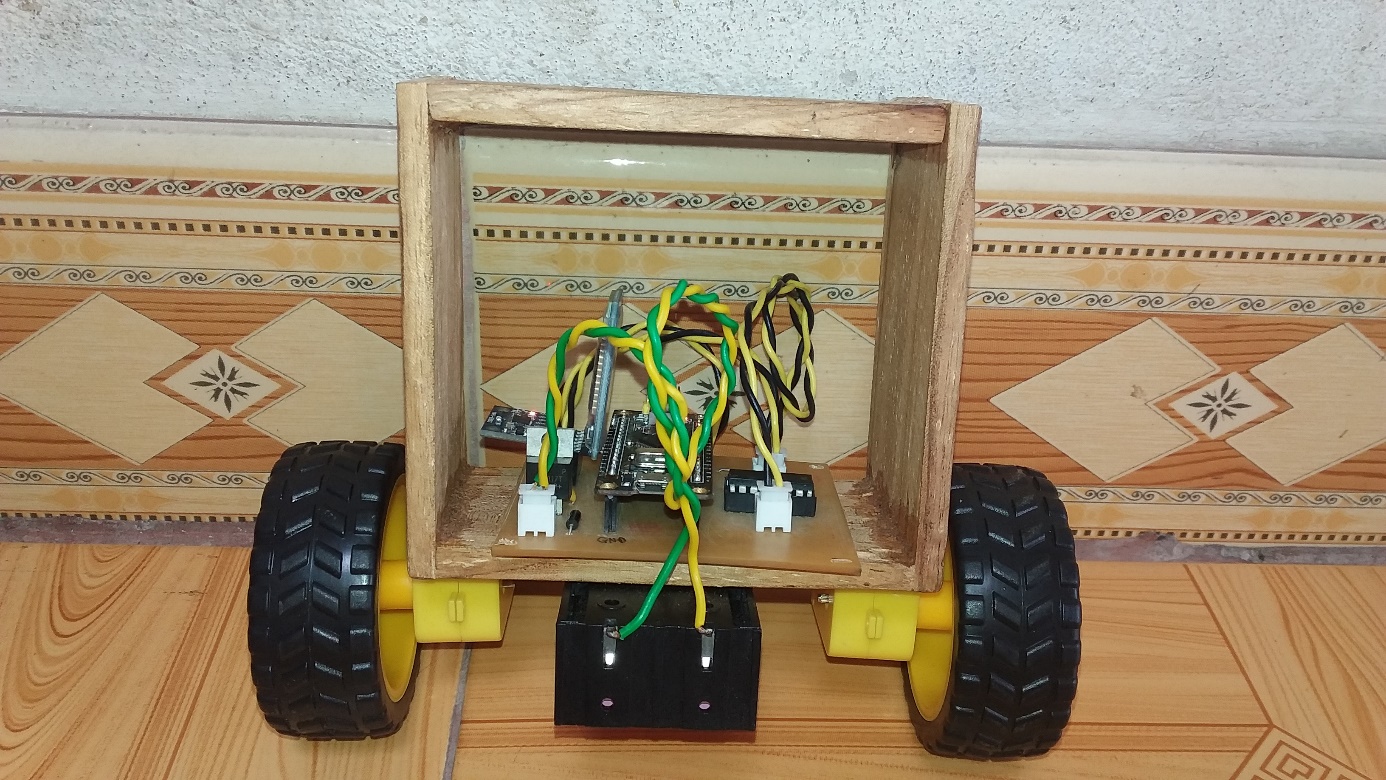
Cảm biến MPU6050 trong module đã ra chân

Một bộ đệm 1024 Byte FIFO trên cảm biến giúp mức tiêu thụ điện của hệ thống thấp hơn bằng cách cho phép bộ vi điều khiển đọc dữ liệu cảm biến trên các trong các chuyển động và sau đó vào chế độ năng lượng thấp khi đó cảm biến thu thập được nhiều dữ liệu. Với toàn bộ tiến trình và các bộ phận cần thiết trên cảm biến để hỗ trợ rất nhiều chuyển động dựa trên các trường hợp sử dụng, cảm biến MPU6050 chỉ cho phép duy nhất các ứng ứng dụng năng lượng thấp truy cập với yêu cầu xử lý đã được giảm thiểu cho bộ xử lý hệ thống. Băng việc cung cấp một ngõ ra tích hợp chuyển động dung hợp, bộ DMP trong cảm biến xuất ra yêu cầu tính toán xử lý chuyên sâu từ hệ thống vi điều khiển, giảm thiểu nhu cầu lấy dữ liệu thường xuyên của đầu ra cảm biến chuyển động. Cảm biến còn được gắn thêm cảm biến nhiệt độ với sai số nhiệt độ trên nhiệt độ hoạt động của chip.



Mô hình nền tảng InvenSense và mô hình MEMS truyền thống

Giao tiếp với toàn bộ thanh ghi của vi cảm biến bằng việc sử dụng đường truyền ở tần số . Tận dụng nền tảng Nasiri-Fabrication đã được cấp bằng sáng chế và đã được chứng minh, nó tích hợp các tấm silic NEMS với CMOS xuyên qua lớp silic, invenSense đã làm cảm biến MPU6050 giảm kích thước mang tính đột phá về công nghệ , trong khi nó mang đến hiệu quả cao nhất, mức độ ồn nhỏ nhất và chi phí nhỏ nhất về lượng chất bán dẫn cho các thiết bị điện tử. Bộ phận này có khả năng chống sốc mạnh mẽ và có bộ lọc thông thấp có thể lập trình cho các con quay hồi chuyển, gia tốc kế và cảm biến nhiệt độ trên chip.



Sản phẩm xe hai bánh tự cân bằng

Cảm biến được sử dụng trong một số ứng dụng như: công nghệ BlurFree (làm mở trong các hình ảnh/ video), công nghệ AirSign (AirSign như một khinh khí cầu sử dụng trong bảo mật và đánh dấu), công nghệ touchAnywhere, công nghệ MotionCommand (Điều khiển bằng chuyển động), cho phép chuyển động trong các game, nhận biết tư thế, dịch vụ dựa trên vị trí, thiết bị sức khỏe, thể dục, thể thao đeo trên người, đồ chơi...

**CHƯƠNG 3. NGHIÊN CỨU VI ĐIỀU KHIẾN 32 BIT STM32F103C8T6**

**3.1. Lịch sử vi điều khiển 32 bit**



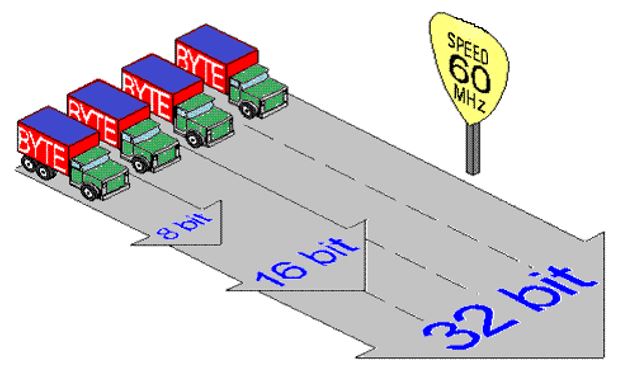
Logo hãng STMicroelectronics

Đề tài sử dụng vi điều khiển STM32F103C8T6 một trong những dòng vi điều khiển của hãng STMicroelectronics. Hãng STMicroelectronics là công ty hàng đầu trên thế giới trong việc cung cấp các giải pháp bán dẫn có đóng góp tích cực cho cuộc sống của con người, ngày nay và tương lai. Đặc biệt là các giải pháp cho Smart Driving và IoT. Các chuyên gia của công ty nhận định rằng, trong thời gian không xa 80% xe hơi sẽ được khởi động bởi hệ thống chip điện tử. Do đó việc phát triển các dòng vi điều khiển hiện đại sẽ hoàn toàn phù hợp trong thời gian tới. Dòng vi điều khiển 32 bit có nhiều ưu điểm và chức năng mới, thuận tiện cho người kỹ sư lập trình. Tốc độ xử lý nhanh (lên đến 72Mhz đối với dòng STM32F1, 168Mhz đối với STM32F4), có ưu điểm cao hơn hẳn so với dòng vi điều khiển 8 bit thông thường. Mặc dù vi điều khiển 8 bit hoàn toàn đáp ứng đề tài, nhưng em vẫn quyết định ứng dụng dòng vi điều khiển 32 bit vào đề tài nhằm tìm hiểu công nghệ cho sự phát triển sau này. Cũng như góp một phần nhỏ tài liệu cho những bạn muốn tìm hiểu về các dòng vi điều khiển 32 bit hiện nay.

**3.2. Vi điều khiển 32 bit và vi điều khiển 8 bit**

Công nghệ vi điều khiển 8 bit đã tồn tại khoảng nửa thế kỉ. Các thiết bị này cho phép triển khai nhanh chóng và dễ dàng các dự án nhúng nhỏ. Bên cạnh đó, chúng ta được biết đến công nghệ vi điều khiển 32 bit mới hơn, đã đạt được rất nhiều sự nổi tiếng. Các thiết bị này cung cấp gần như tất cả mọi thứ từ sức mạnh xử lý cao đến các thiết bị ngoại vi phong phú cùng với các công cụ lập trình phát triển dễ dàng và nhanh chóng giúp cho người kỹ sư lập trình rất nhiều.

* *Kiến trúc*



8, 16 hay 32 bit về cơ bản là để xác định kích thước của đường truyền nội bộ và kích thước thanh ghi của vi điều khiển. Vì vậy một vi điều khiển 8 bit sẽ có byte không gian địa chỉ trong khi vi điều khiển 32 bit sẽ có byte hay tính ra là 4GB không gian địa chỉ. Hầu hết các dòng vi điều khiển 8 bit có dòng địa chỉ 16 bit (8 bit thấp và 8 bit cao) cho phép chúng có nghĩa là 64KB bộ nhớ. Do đó, với các ứng dụng cần lớn hơn 64KB thì vi điều khiển 32 bit là một thiên đường cho các kỹ sư cơ điện tử.

Mỗi cổng vào ra cho một vi điều khiển 32 có 16/32 chân mỗi cổng, không giống như 8 chân mỗi cổng cho vi điều khiển 8 bit. Điều này thì có thể không ảnh hưởng nhiều. Nhưng nếu một ứng dụng mà chúng ta chần nhiều chân vào ra thì một vi điều khiển 32 bit sẽ phục vụ rất tốt. Một vi điều khiển 8 bit điển hình sẽ không có số pin trên 64 chân trong khi đó số pin 100-144 là rất phổ biến trong dòng vi điều khiển 32 bit.

* *Cách thức phát triển phần mềm*

Với các vi điều khiển 8 bit thì việc phát triển phần mềm là tương đối dễ dàng. Chúng có vài thanh ghi để tinh chỉnh với quá trình cài đặt dễ dàng. Ví dụ, một vi điều khiển 32 bit điển hình có nhiều cài đặt xung nhịp nội bộ. Các cài đặt này được yêu cầu cài đặt trước khi khỏi động vi điều khiển. Điều này thì rất đơn giản với các vi điều khiển 8 bit, chúng ta có thể thực hiện tại nhiều vị trí với việc khoier tạo con trỏ ngăn xếp, mất khoảng 4 dòng lệnh (ví dụ như tinyAVR và megaAVR của Atmel). Nhưng các nhà sản xuất vi điều khiển 32 bit đang cố gắng để phân tích khuân mẫu này và đưa ra các mẫu và thư viện mặc định, làm cho quá trình phát triển đơn giản và nhanh hơn. Chỉ cần sử dụng các thư viện này trong dự án của chúng ta là các khởi tạo sẽ được thực hiện.

* *Hiệu suất*

Vi điều khiển 32 bít cung cấp khả năng tính toán đang kinh ngạc khi chúng có thanh ghi 32 bit. Nếu ta lập trình bằng ngôn ngữ assemble, thì việc giới hạn kích thước thanh ghi là điều hiển nhiên sẽ xảy ra. Ví như khi ta muốn cộng hai số 32 bit trên một vi điều khiển 8 bit thì ta phải chia số đó ra thành nhiều phần sau đó cộng các phần lại với nhau, điều này làm cho việc lập trình tính toán rất phức tạp. Khi ta lập trình bằng ngôn ngữ C việc tính toán trở nên mờ mịt, tất nhiên là vẫn có cách tính toán nhưng sẽ mất nhiều thời gian khi sư dụng vi điều khiển 8 bit. Do đó vi điều khiển 8 bit không phải một lựa chọn hoàn hảo khi lập trình các ứng dụng liên quan đến tính toán cao.

* *Thiết bị ngoại vi*

Hiển nhiên vi điều khiển 8 bit truyền thống sẽ bị thiếu rất nhiều thiết bị ngoại vi. Do các thiết bị ngoại vi ngày càng phát triển. Không những thế nó còn bị giới hạn không gian địa chỉ bộ nhớ. Khi ta muốn sử dụng thêm các thiết bị ngoại vi, các thanh ghi dùng điều khiển chúng sẽ cần đặt vào trong cùng một địa chỉ có sẵn. Bên cạnh đó các vi điều khiển 32 bit có rất nhiều các thiết bị ngoại vi, được cung cấp ADCs, DACs, nhiều timer, phần cứng chuyên dụng cho kết nối,...

Tuy vậy vẫn có một số ít vi điều khiển 8 bit được cung cấp ngoại vi phong phú

* *Sự tiêu thụ năng lượng*

Kiến trúc của vi điều khiển 8 bit được biết đến là tiêu thụ năng lượng ít vì sử dụng ít phần bán dẫn hơn. Hơn nữa vi điều khiển 8 bit cũng đã có lịch sử phát triển lâu dài nên sự tiết kiệm về năng lượng được cải thiện đáng kế. Các vi điều khiển 32 bit thì tiêu thụ năng lượng hơn, mặc dù dòng điện là chỉ khoảng vài mA.

Nhưng với sự ra đời của dòng vi điều khiển 32 bit Cortex M0, mọi thứ đã thay đổi ất nhiều. Công nghệ này được giới thiệu vào năm 2012 và nó được xem như là bộ vi điều khiển ARM tiết kiệm năng lượng nhất kể từ năm 2016.

Tóm lại, dòng vi điều khiển 32 bit phù hợp với các ứng dụng cần sự tính toán cường độ cao. Còn vi điều khiển 8 bit thì sử dụng trong các ứng dụng đơn giản với giá thành thấp, mức độ tính toán đơn giản. Hiện nay trên thế giới các kỹ sư và các lập trình viên nhúng đang dần chuyển sang sử dụng vi điều khiển 32 bit vì những ưu điểm của chúng.

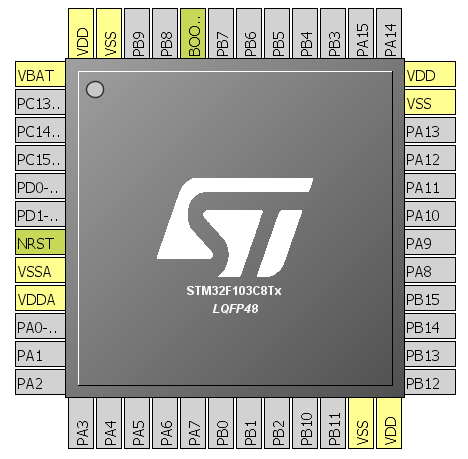
**3.3. Một số đặc tính của vi điều khiển STM32F103C8T6**



Vi điều khiển STM32F103C8T6

STM32F103C8T6 là một vi điều kiển sử dụng lõi vi điều khiển ARM Cortex-M3 32 bit (CortexM3 thế hệ sau của CortexM0) hoạt động tại xung nhịp cao 72Mhz, truy xuất bộ nhớ nhúng tốc độ cao (bộ nhớ Flash lên đến 128 Kbytes và bộ nhớ truy cập ngẫu nhiên tĩnh (SRAM) lên đến 20 Kbytes), các cổng được cải tiến hàng loạt và các thiết bị ngoại vi đã được kết nối với đường kết nối ngoại vi nâng cao (APB). Vi điều khiển có 2 bộ chuyển đổi ADC 12 bit, 3 timer thường và một timer PWM, cũng như cổng giao tiếp cơ bản và nâng cao lên đến 2 cổng và cổng SPI, ba cổng giao tiếp UART, một cổng USB và một cổng CAN.

Thiết bị hoạt động ở mức điện áp . Hoạt động được cả dải nhiệt độ đến và khoảng rộng nhất là đến . Một bộ điều chế tiết kiệm năng lượng toàn diện cho phép thiết kế các ứng dụng công suất thấp.



Sơ đồ chân của vi điều khiển STM32F103C8T6

Dòng vi điều kiển này còn cung cấp nhiều chuẩn chân VFQFPN, UFQFPN, LFBGA, LQFP, UFBGA, LQFP, TFBGA, LQFP cho phép người kỹ sư thiết kế đa dạng sử dụng chip trong nhiều ứng dụng.

CHƯƠNG 4: THIẾT KẾ XE TỰ CÂN BẰNG

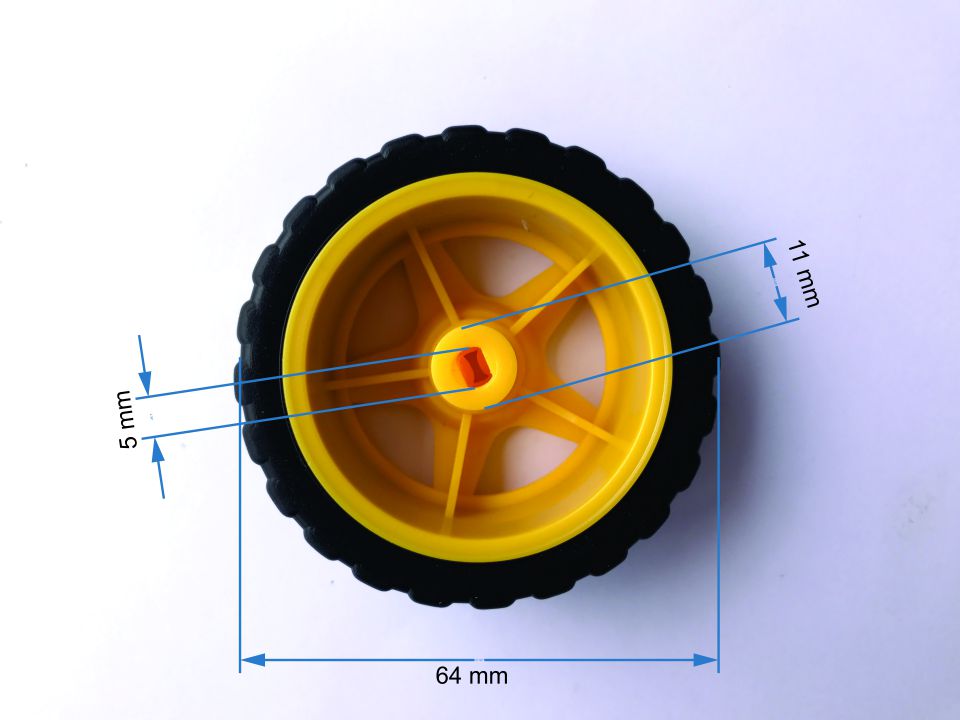
4.1. Thiết kế mạch điện cho xe tự cân bằng

4.2. Thiết kế, lắp ráp phần cứng của xe tự cân bằng



Động cơ giảm tốc

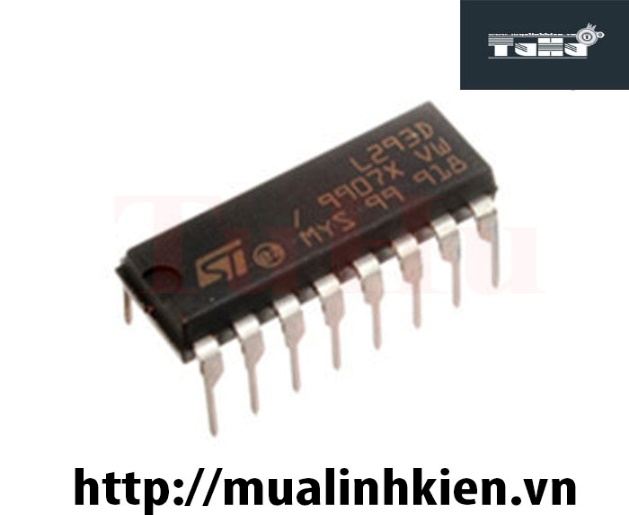
Xe hai bánh tự cân bằng sử dụng 2 động cơ giảm tốc . Tốc độ quay với ứng với điện áp và là và Mô men xoắn cực đại .



Bánh xe động cơ giảm tốc

//Hình ảnh

Khung xe



4.3. Thiết kế phần mềm điều khiển cho xe tự cân bằng

**PHẦN III. KẾT LUẬN**

* ***Kết quả đạt được***

Xe tự cân bằng có khả năng tự cân bằng trên mặt phảng

Điều khiển được xe di chuyển các hướng bằng phần mềm điều khiển trên di động

Ghi lại được các giá trị giao động của xe

* ***Hạn chế còn sót lại***

Xe chưa di chuyển được trên dịa hình gồ ghề, sỏi đá

Sản phẩm chưa được thiêt kế gọn gàng và thẩm mỹ

* ***Đề xuất cải tiến***

//

* ***Phương hướng phát triển***

TÀI LIỆU THAM KHẢO

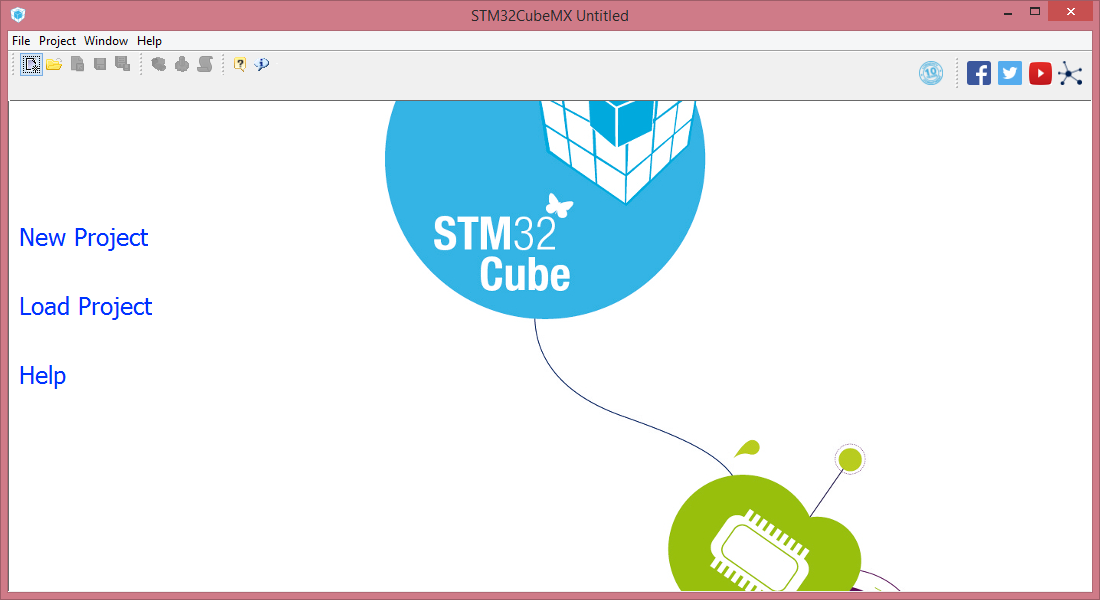
PHỤ LỤC

1. Hướng dẫn sinh mã nguồn tự động cho dòng vi điều khiển STM32 bằng phần mềm STM32CubeMx

1.1. Chuẩn bị

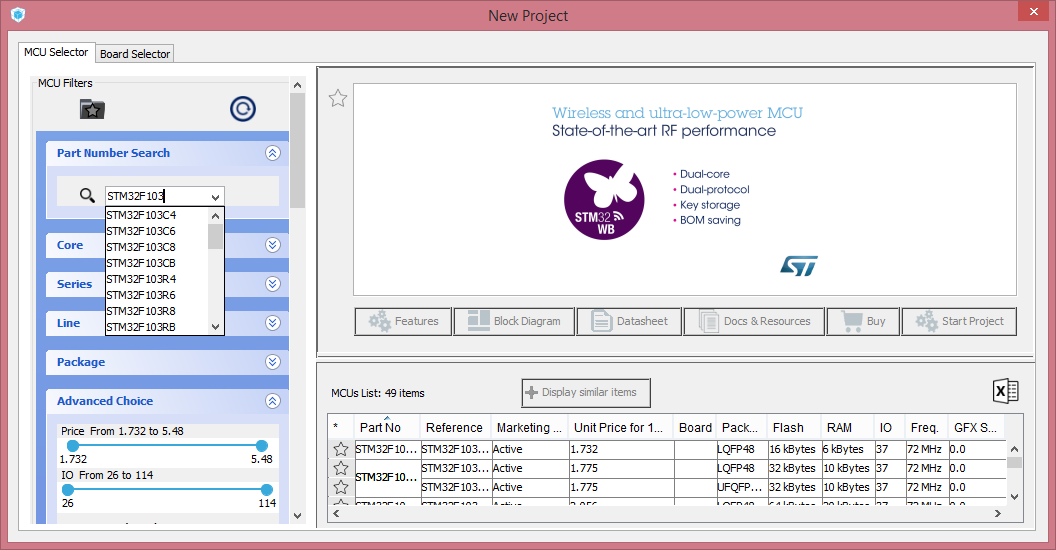
1.2. Hướng dẫn

Như đã giới thiệu từ ban đầu, các vi điều khiển 32 bit tuy mạnh mẽ nhưng việc cài đặt và lập trình bước đầu tương đối vất vả, do số lượng thanh ghi lớn cũng như cấu trúc hoạt động. Tuy nhiên, các nhà sản xuất đã có những nỗ lực rất lớn trong việc tạo ra các công cụ hỗ trợ cho người kỹ sư phần mềm. Đó là phần mềm STM32CubeMX do chính hãng STMicroelectronics phát triển.



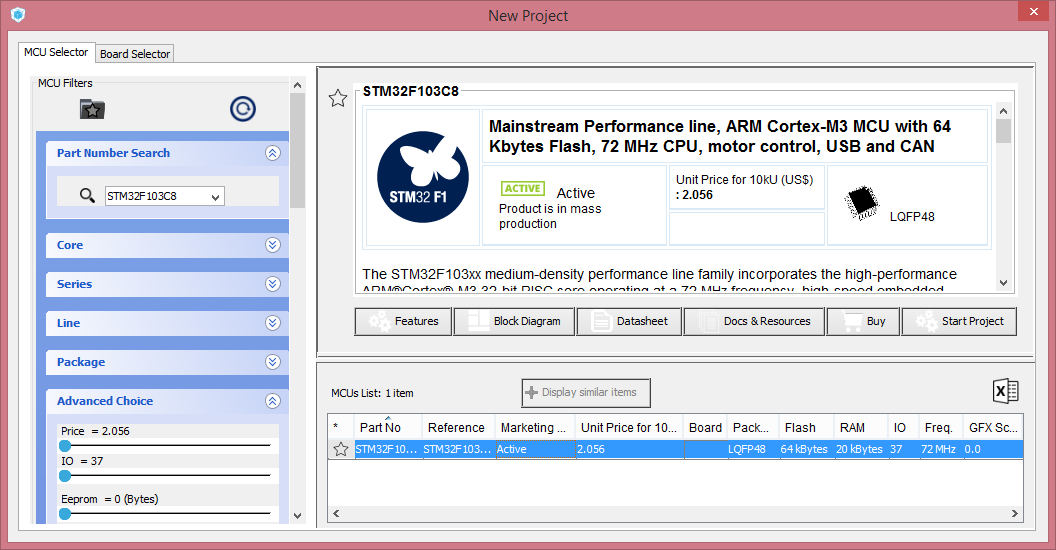
Giao diện ban đầu của phần mềm STM32CubeMX

Phần mềm STM32CubeMX là một phần mềm giao diện cho phép người kỹ sư lập trình lựa chọn, cài đặt và tùy chỉnh các thuộc tính, chức năng trên vi điều khiển một cách dễ dàng. Sau đó phần mềm sẽ tự động sinh ra phần mã nguồn (code) giúp người lập trình tiến nhanh và gần hơn đến các phần xử lý logic của dự án.



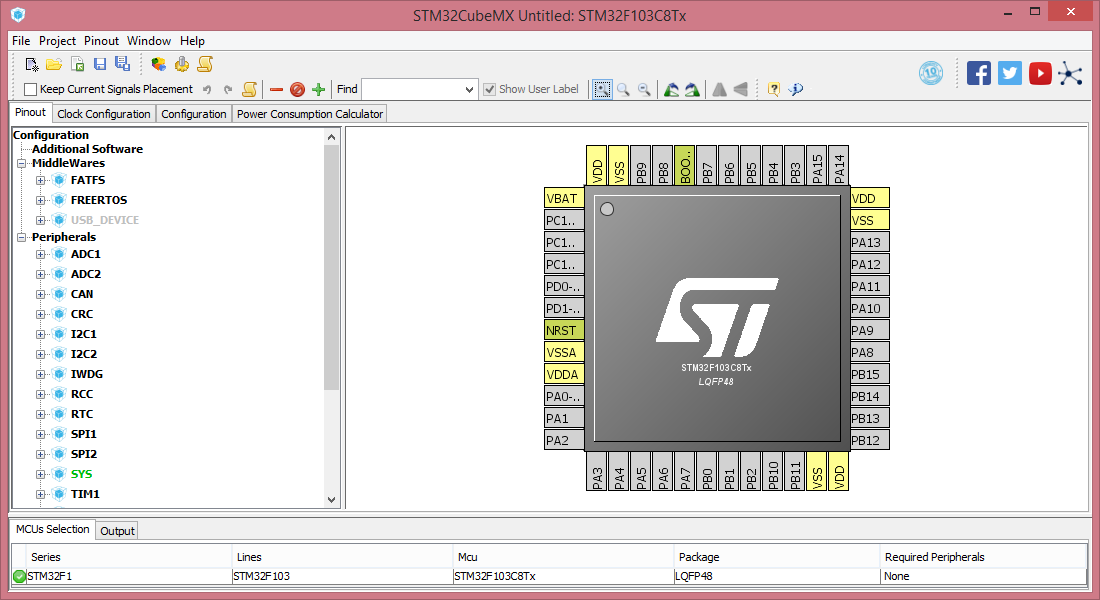
Lựa chọn vi điều khiển

Sau khi click vào “New Project” (Dự án mới), kỹ sư lập trình sẽ cần phải lựa chọn loại vi điều khiển cần phát triển bằng cách tìm kiếm tên vi điều khiển trong ô tìm kiếm (cách đơn giản nhất).



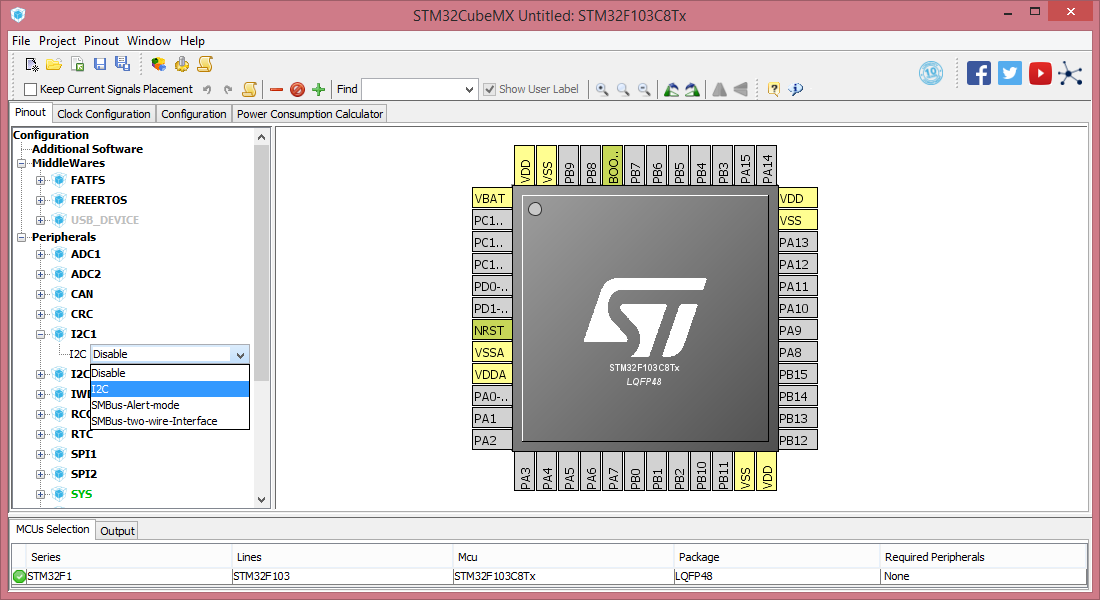
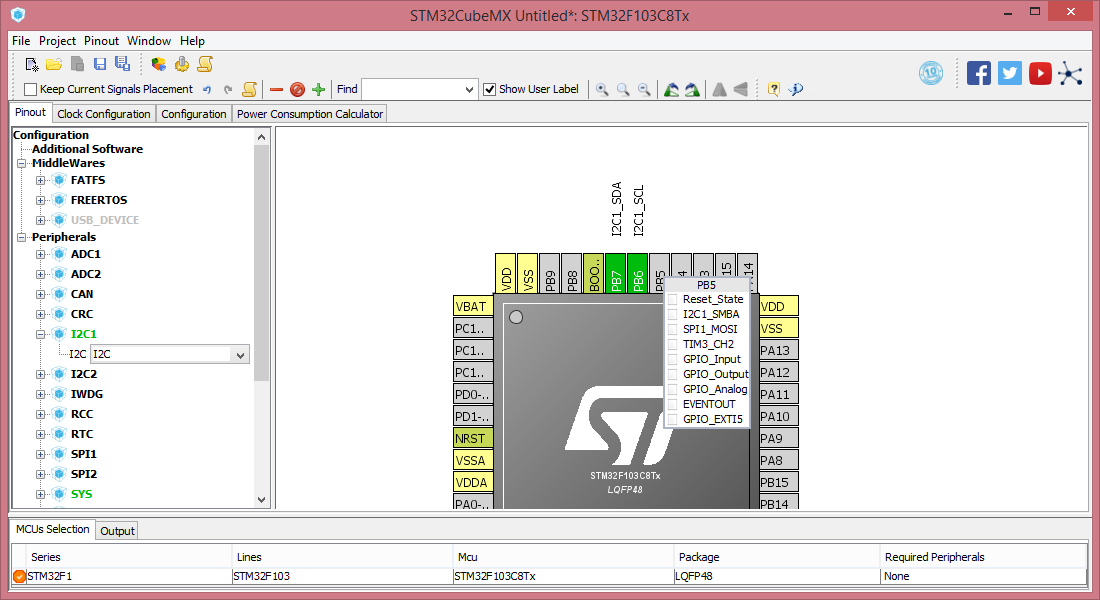
Bắt đầu một project mới

Sau khi tìm kiếm thành công, một danh sách các vi điều khiển sẽ được lọc ra trong bảng. Click chọn vi điều khiển và nhấn vào “Start Project” là chúng ta đã đi được bước đầu tiên trong việc tạo một dự án cho dòng vi điều khiển 32 bit này. Bên cạnh đó phần mềm STM32CubeMx cũng hiển thị nhiều thông tin giúp người kỹ sư lập trình hiểu hơn về vi điều khiển như phần thông tin vi điều khiển, kiểu chân, giá cả,... cùng nhiều thông tin khác.



Giao diện cấu hình cho vi điều khiển

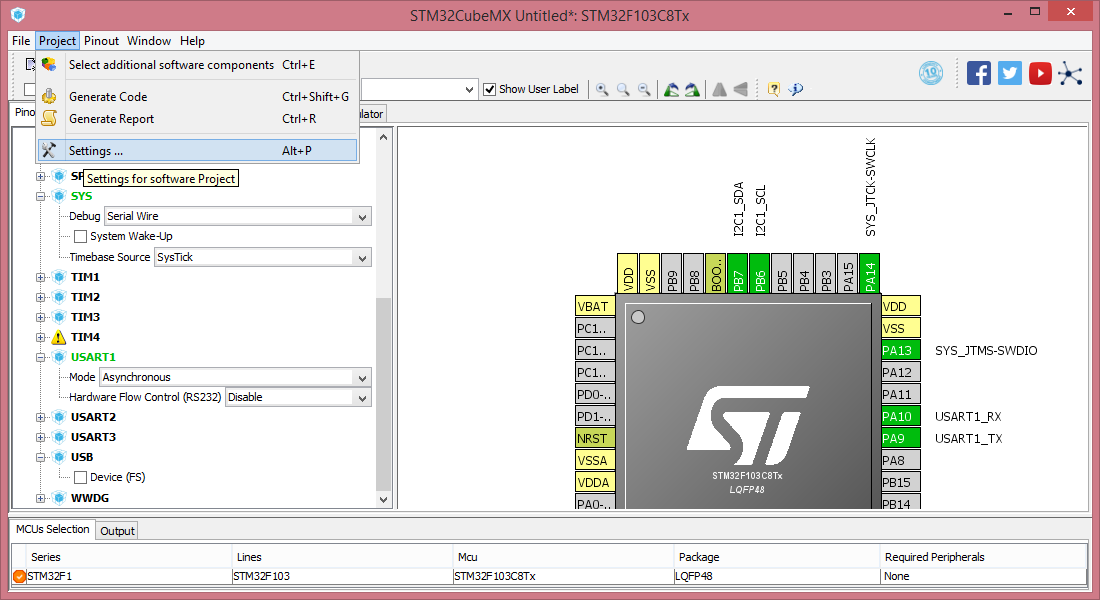
Sau khi đã tạo được dự án mới, giao diện thuận tiện của STM32CubeMx hiện lên vô cùng trực quan với 4 tab chính: cấu hình các chân I/O (pinout), cấu hình xung clock (Clock Configuration), cấu hình các chức năng (Configuration) và tính toán mức độ tiêu thụ năng lượng (Power Consumption Calulator).

Cấu hình chức năng chân cho vi điều khiển

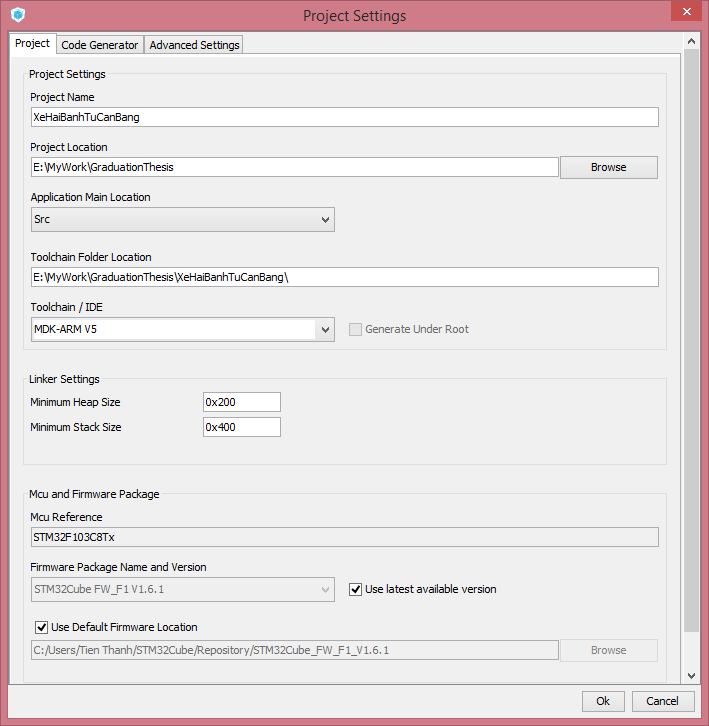
Khi cấu hình chân I/O cho vi điều khiển, phần mềm phân chia rất rõ ràng thành 2 phần, phần danh sách giao tiếp các thiết bị ngoại vi và phần trực quan vi điều khiển. Đối với phần danh sách giao tiếp các thiết bị ngoại vi, người kỹ sư lập trình chỉ cần lựa chọn loại giao tiếp mình cần sử dụng, ví dụ như , người kỹ sư lập trình sẽ chọn I2C1 hoặc I2C2 (vì STM32F103C8T6 cung cấp 2 cổng ) và chọn thuộc tính cho nó, chọn “I2C” để sử dụng cổng này, “Disable” để ngắt cổng và một số tùy chọn nâng cao khác. Đối với phần trực quan hiển thị các chân vi điều khiển, kỹ sư click chuột trái vào một chân của vi điều khiển, nó sẽ hiển thị ra các tùy chọn có thể thiết lập đối với chân đó như biến nó thành một chân reset, một chân ngắt Timer, một chân đầu ra,... Hoàn toàn trực quan và dễ dàng sử dụng.

Tương tự các thiết lập khác đều tương đối dễ dàng cho người kỹ sư lập trình sử dụng vào cài đặt cho vi điều khiển của mình.



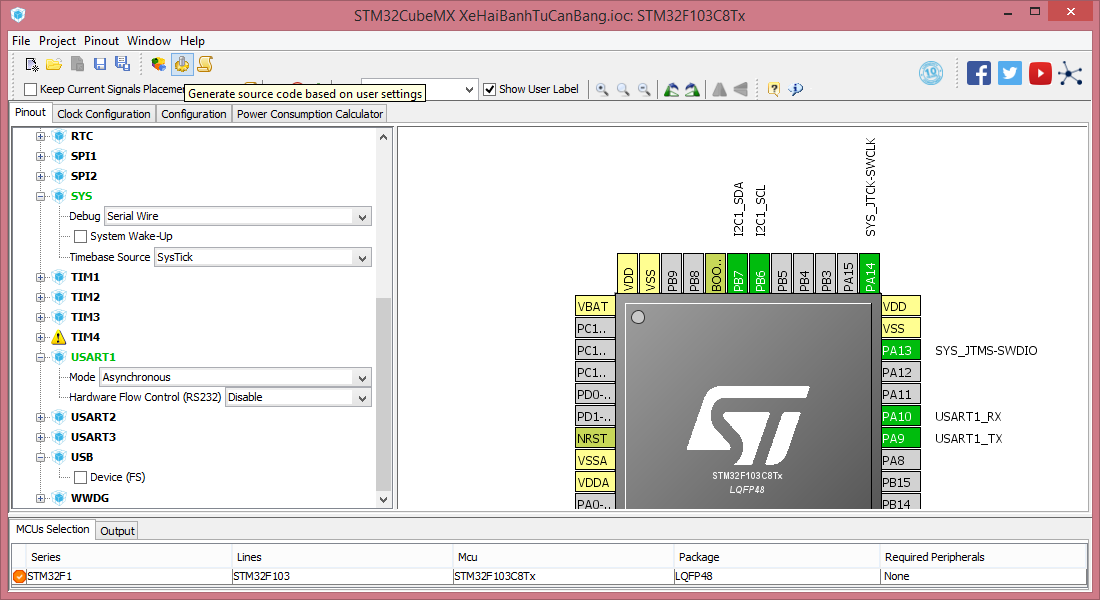
Lựa chọn cài đặt dự án

Sau khi thiết lập xong toàn bộ các thuộc tính cần dùng cho vi điều khiển, kỹ sư lập trình cần vào cài đặt dự án (Settings...) bằng cách nhấn vào “Project” trên thanh menu và lựa chọn “Setting...” hoặc sử dụng tổ hợp phím “Alt+P”.



Cài đặt dự án

Tại phần cài đặt dự án, người kỹ sư điền đầy đủ thông tin về dự án. Như tên dự án, địa chỉ thư mục lưu dự án,... Quan trọng nhất là công cụ để phát triển mã nguồn về sau này tại phần “Toolchain/ IDE”, có rất nhiều công cụ phù hợp với việc tự động sinh mã nguồn của STM32CubeMx như EWARM, MDK-ARM V5, TrueSTUDIO, SW4STM32,... Sau khi lựa chọn xong công cụ phát triển, chúng ta nhấn OK và trong lần đầu tiên sử dụng STM32CubeMx cho một dòng vi điều khiển mới, công cụ STM32CubeMx sẽ cần download các thư viện hỗ trợ, hỗ trợ ngoại vi từ trên server về máy của chúng ta. Việc này hoàn toàn tự động, tiện ích và do chính nhà sản xuất vi điều khiển cung cấp do đó sẽ vô cùng an toàn cho vi điều khiển cũng như dự án.



Sinh mã nguồn cho vi điều khiển

Cuối cùng là sinh mã nguồn cho vi điều khiển bằng cách nhấn vào biểu tượng răng cưa tại thanh menu. Mã nguồn sẽ được tự động sinh ra theo như những gì mà người kỹ sư lập trình đã cài đặt từ đầu. Sẽ không phải tốn thêm một giây nào cho việc cài đặt xung nhịp hoạt động của vi điều khiển hay thiết lập giao tiếp. Do đó người kỹ sư lập trình có thể bắt tay trực tiếp vào phần xử lý logic của dự án.