

**ĐẠI HỌC QUỐC GIA TP.HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC BÁCH KHOA**

**KHOA ĐIỆN – ĐIỆN TỬ**

**BỘ MÔN ĐIỀU KHIỂN TỰ ĐỘNG**

-----oOo-----



**BÁO CÁO THỰC TẬP NGOÀI TRƯỜNG**

**CÔNG TY TNHH CÔNG NGHỆ PIF LAB**

**GVHD: TS. NGUYỄN VĨNH HẢO**

**SVTH: NGUYỄN VÕ HỒNG MỸ HIỀN**

**MSSV: 2010260**

**TP. HỒ CHÍ MINH, THÁNG 08 NĂM 2023**

## LỜI CẢM ƠN

Lời đầu tiên, em xin gửi lời cảm ơn đến công ty PIF Lab và các anh, chị, bạn bè trong công ty vì sự giúp đỡ và chỉ bảo tận tình trong suốt thời gian thực tập vừa qua.

Thực tập tốt nghiệp tại công ty PIF Lab đã mang đến cho em những trải nghiệm vô cùng quý báu và đáng nhớ. Em xin bày tỏ lòng biết ơn đặc biệt đến cô Phương, anh Việt - người đã trực tiếp hướng dẫn, hỗ trợ và giám sát quá trình thực tập của em. Nhờ có sự chỉ bảo tận tâm của các anh chị, em đã được tiếp cận với những kiến thức, kỹ năng và kinh nghiệm thực tế trong lĩnh vực Embedded.

Em cũng xin cảm ơn thầy Nguyễn Vĩnh Hào - giảng viên hướng dẫn thực tập của em, bên cạnh những kiến thức chuyên môn mà thầy truyền đạt và hướng dẫn giúp em có nền tảng tốt khi tìm hiểu những lĩnh vực sâu hơn. Qua đó, góp phần giúp em hoàn thành tốt khóa thực tập lần này.

Trong suốt thời gian em thực tập tại công ty sẽ có những điểm thiếu sót, bài báo cáo sẽ còn những điểm chưa tốt, em mong được mọi người đóng góp ý kiến để em có thể rút kinh nghiệm sửa chữa cho hoàn chỉnh hơn và cho công việc tương lai sau này.

Em xin chân thành cảm ơn!

*Tp. Hồ Chí Minh, ngày 11 tháng 08 năm 2023.*

Nguyễn Võ Hồng Mỹ Hiền

## MỤC LỤC

<b>LỜI CẢM ƠN .....</b>	<b>2</b>
<b>MỤC LỤC .....</b>	<b>3</b>
<b>DANH MỤC HÌNH ẢNH.....</b>	<b>5</b>
<b>ĐÁNH GIÁ TỪ DOANH NGHIỆP .....</b>	<b>7</b>
<b>ĐÁNH GIÁ TỪ GIẢNG VIÊN HƯỚNG DẪN .....</b>	<b>8</b>
<b>PHẦN 1: GIỚI THIỆU CÔNG TY THỰC TẬP .....</b>	<b>9</b>
<b>PHẦN 2: NỘI DUNG CÔNG VIỆC THỰC HIỆN .....</b>	<b>10</b>
<b>2.1. Thời gian và nhiệm vụ được giao .....</b>	<b>10</b>
<b>2.2. Giới thiệu tổng quan về đề tài.....</b>	<b>10</b>
<b>2.3. Nội dung1: Thiết kế sơ đồ nguyên lí .....</b>	<b>11</b>
<b>2.3.1. Thiết kế mạch nguồn .....</b>	<b>17</b>
<b>2.3.2. Thiết kế mạch MCU .....</b>	<b>20</b>
<b>2.4. Nội dung 2: Thiết kế mạch PCB .....</b>	<b>22</b>
<b>2.5. Nội dung 3: Lập trình State Machine.....</b>	<b>26</b>
<b>2.5.1. Lý thuyết.....</b>	<b>26</b>
<b>2.5.2. Thực hành.....</b>	<b>27</b>
<b>2.6. Nội dung 4: Hàn mạch, chuẩn bị phần cơ khí và lắp xe .....</b>	<b>31</b>
<b>2.7. Nội dung 5: Các giải thuật để lập trình cân bằng cho Self Balancing Robot .....</b>	<b>34</b>
<b>PHẦN 3: TỔNG KẾT CÔNG VIỆC THỰC TẬP .....</b>	<b>37</b>
<b>3.1. Kết quả công việc thực tập.....</b>	<b>37</b>
<b>3.2. Kinh nghiệm học được sau khi thực tập .....</b>	<b>37</b>

<b>3.3. Kế hoạch nghề nghiệp trong tương lai .....</b>	<b>37</b>
<b>PHẦN 4: TÀI LIỆU THAM KHẢO .....</b>	<b>38</b>
<b>PHẦN 5: PHỤ LỤC .....</b>	<b>39</b>

## DANH MỤC HÌNH ẢNH

Hình 1: Sơ đồ khối của Self Balancing Robot .....	11
Hình 2: Lưu đồ thiết kế sơ đồ nguyên lí .....	11
Hình 3: Sơ đồ khối của mạch thiết kế.....	12
Hình 4: Sơ đồ khối nguồn cung cấp .....	12
Hình 5: Sơ đồ nguyên lí của mạch cầu H .....	13
Hình 6: Sơ đồ nguyên lí của khối nguồn .....	14
Hình 7: Sơ đồ nguyên lí khối MCU .....	15
Hình 8: Sơ đồ nguyên lí khối sensor .....	16
Hình 9: Cấu trúc cơ bản của nguồn LDO .....	17
Hình 10: Hoạt động của nguồn LDO.....	18
Hình 11: Đáp ứng quá độ của nguồn LDO khi tụ thay đổi.....	19
Hình 12: Đáp ứng của LM1117 khi đầu vào có sự thay đổi.....	19
Hình 13: Sơ đồ cấp nguồn của STM32F103C8T6.....	20
Hình 14: Mạch nút ấn reset cho SMT32F103C8T6 .....	21
Hình 15: Các chế độ BOOT của STM32F103C8T6 .....	21
Hình 16: Layout khối nguồn của mạch cầu H .....	22
Hình 17: Layout khối nguồn của mạch chính .....	22
Hình 18: Vị trí đặt tụ decoupling và tụ bypass .....	23
Hình 19: Layout khối thạch anh của mạch chính.....	24
Hình 20: Layout và đi dây của mạch chính .....	24
Hình 21: 3D View của mạch chính .....	24
Hình 22: Layout và đi dây của mạch cầu H.....	25

Hình 23: 3D View của mạch cầu H.....	25
Hình 24: Đề bài State Machine .....	27
Hình 25: Flow chart của State Machine .....	29
Hình 26: Flow chart đọc nút nhấn.....	30
Hình 27: Sản phẩm hoàn thiện của mạch chính.....	31
Hình 28: Sản phẩm hoàn thiện của mạch cầu H .....	31
Hình 29: 3D view của Self Balancing Robot trên SolidWork.....	32
Hình 30: Self Balancing Robot .....	33
Hình 31: Flow chart của Self Balancing Robot .....	34
Hình 32: Qui trình tính toán của bộ lọc Kalman.....	35
Hình 33: Mô hình và công thức bộ điều khiển PID .....	35

## ĐÁNH GIÁ TỪ DOANH NGHIỆP

Họ và tên sinh viên: Nguyễn Võ Hồng Mỹ Hiền

MSSV: 2010260

Khoa: Điện - Điện tử

Trường: Đại học Bách Khoa Thành phố Hồ Chí Minh

Trong thời gian từ ngày 12 tháng 06 năm 2023 đến ngày 11 tháng 08 năm 2023

Tại: CÔNG TY TNHH CÔNG NGHỆ PIF LAB

Địa chỉ: Số 4 Đ. số 1, Khu Phố 2, Thành phố Thủ Đức, Thành phố Hồ Chí Minh 700000

Sau quá trình thực tập tại đơn vị của sinh viên, chúng tôi có một số nhận xét đánh giá như sau:

Nội dung đánh giá	Giỏi	Khá	Trung bình	Yếu
Ý thức kỷ luật				
Thái độ làm việc				
Tác phong làm việc				
Năng lực, kiến thức chuyên môn				
Khả năng hoàn thành công việc				

***Đánh giá chung sau khi thực tập:***

.....  
.....  
.....

***Điểm đánh giá quá trình thực tập (theo thang điểm 10):*** .....

*Tp. Hồ Chí Minh, ngày 11 tháng 08 năm 2023*

**Xác nhận của đơn vị thực tập**

*(Ký và ghi rõ họ và tên)*

## ĐÁNH GIÁ TỪ GIẢNG VIÊN HƯỚNG DẪN

Họ và tên sinh viên: Nguyễn Võ Hồng Mỹ Hiền

MSSV: 2010260

Khoa: Điện - Điện tử

Trường: Đại học Bách Khoa Thành phố Hồ Chí Minh

Sau quá trình thực tập của sinh viên, giảng viên hướng dẫn nhận xét, đánh giá như sau:

**Đánh giá quá trình thực tập tại đơn vị thực tập:**

.....  
.....

**Thực hiện báo cáo thực tập:**

.....  
.....

**Đánh giá chung sau khi thực tập:**

.....  
.....

**Điểm đánh giá về thực hiện báo cáo thực tập: ...../10.**

*Bằng chữ:*.....

*Tp. Hồ Chí Minh, ngày 11 tháng 08 năm 2023*

**Giảng viên hướng dẫn**

*(Ký và ghi rõ họ và tên)*



## **PHẦN 1: GIỚI THIỆU CÔNG TY THỰC TẬP**

Được thành lập vào ngày 22 tháng 12 năm 2009, tổ chức “Pay It Forward” tạo ra một cộng đồng liên kết giữa các kỹ sư, nhà khoa học, giảng viên và sinh viên, trong đó kiến thức và niềm đam mê trong lĩnh vực điện và điện tử có thể được chia sẻ và truyền lại cho các thế hệ tiếp theo.

Pay It Forward nhằm mục đích xây dựng một cộng đồng ham học hỏi với các yếu tố chính là sáng tạo, đổi mới, sẵn sàng giúp đỡ người khác, cùng nhau phát triển và đạt được nhiều thành tựu hơn trong sự nghiệp của chính mọi người.

PIF Lab là Trung tâm Đổi mới của Cộng đồng Pay It Forward với nhiều kỹ sư và nhà khoa học giàu kinh nghiệm và đam mê, luôn tìm tòi nghiên cứu và khai thác công nghệ tiên tiến để tạo ra các sản phẩm thông minh và hữu ích.

Các sản phẩm của PIF Lab được phát triển và tạo ra dựa trên nhu cầu của nhu cầu sinh hoạt, công nghiệp và giáo dục.

PIF Lab cung cấp các giải pháp và hỗ trợ kỹ thuật cho các công ty khởi nghiệp.

PIF Lab nhận thấy vai trò quan trọng của công nghệ đối với cộng đồng và luôn nỗ lực hết mình để phục vụ cộng đồng với những thành tựu đã đạt được trong suốt những năm qua.

## **PHẦN 2: NỘI DUNG CÔNG VIỆC THỰC HIỆN**

### **2.1. Thời gian và nhiệm vụ được giao**

Thời gian thực tập 2 tháng (từ ngày 12/06/2032 đến ngày 11/08/2023)

#### **Lịch trình thực tập:**

**Tuần 1:** Làm quen với phần mềm vẽ mạch PCB Kicad. Chọn đề tài: Self-balancing Robot.

**Tuần 2:** Lựa chọn modules, IC. Vẽ sơ đồ khối nguồn điện cung cấp

**Tuần 3:** Hoàn thành vẽ schematic và bắt đầu layout.

**Tuần 4:** Hoàn thành thiết kế PCB.

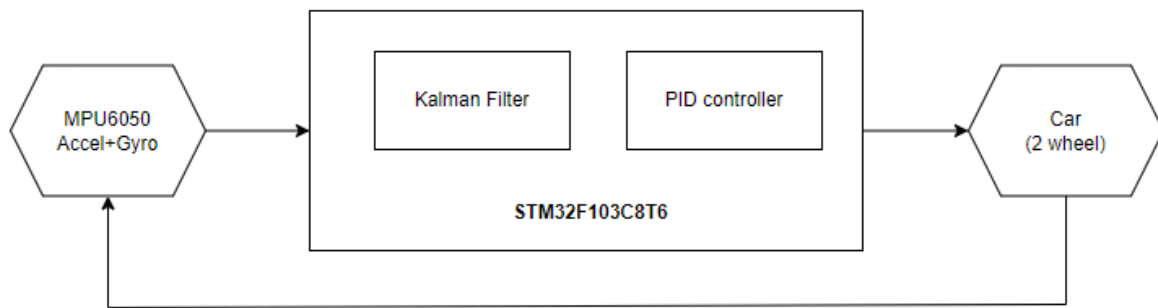
**Tuần 5-6:** Trong thời gian chờ mạch lập trình State Machine và chuẩn bị phần cơ khí để lắp xe.

**Tuần 7:** Hàn mạch, test mạch, lắp xe và bắt đầu lập trình.

**Tuần 8:** Tune PID cho xe và hoàn thành project.

### **2.2. Giới thiệu tổng quan về đề tài**

Self-balancing Robot là xe tự cân bằng hai bánh. Sử dụng vi xử lý chính là STM32F103C8T6, board mạch chính và mạch cầu tự thiết kế. Ứng dụng bộ lọc Kalman để ước lượng giá trị góc từ gia tốc kế và con quay hồi chuyển. Sử dụng bộ điều khiển PID để điều khiển góc.



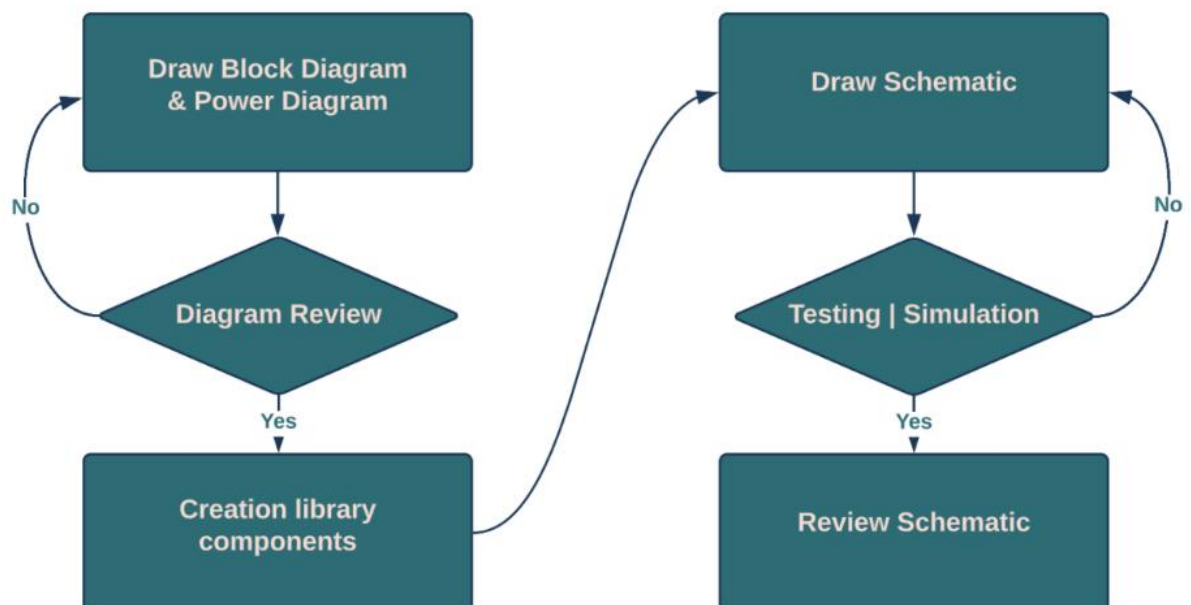
Hình 1: Sơ đồ khối của Self Balancing Robot

Xe có thể đứng vững và ổn định sau khi hiệu chỉnh có thông số. Có thể điều khiển xe bằng Bluetooth.

Hướng phát triển: sử dụng bộ điều khiển LQR (điều chỉnh toàn phương tuyến tính).

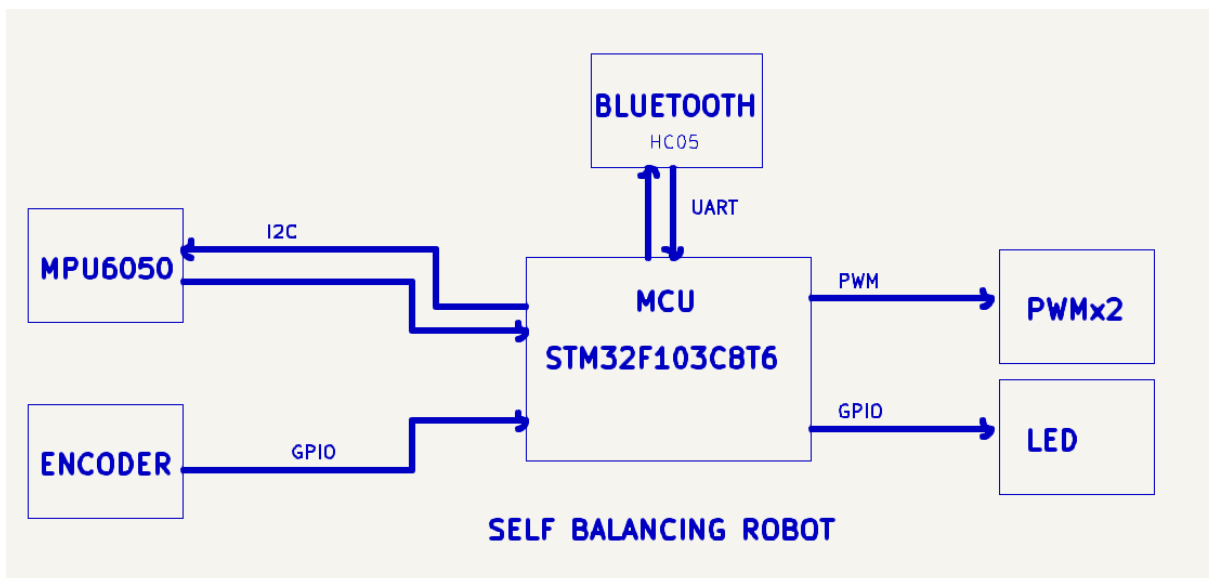
### 2.3. Nội dung1: Thiết kế sơ đồ nguyên lí

Quy trình để thiết kế sơ đồ nguyên lí:



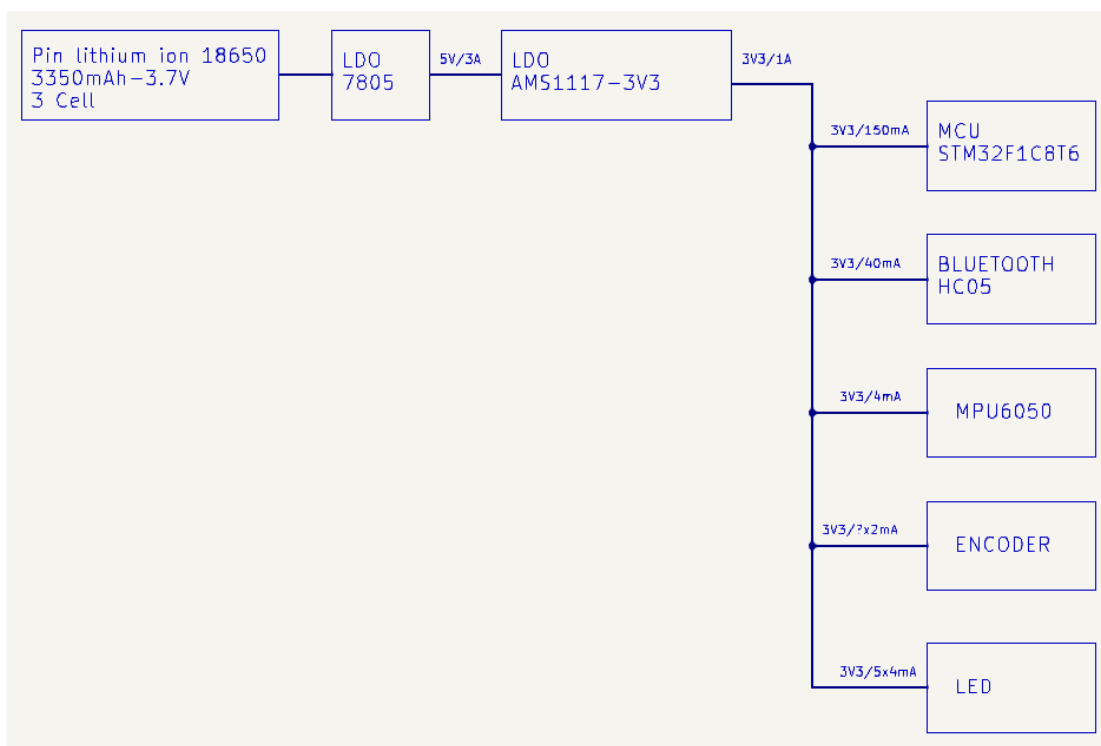
Hình 2: Lưu đồ thiết kế sơ đồ nguyên lí

### Sơ đồ khối (Block diagram)



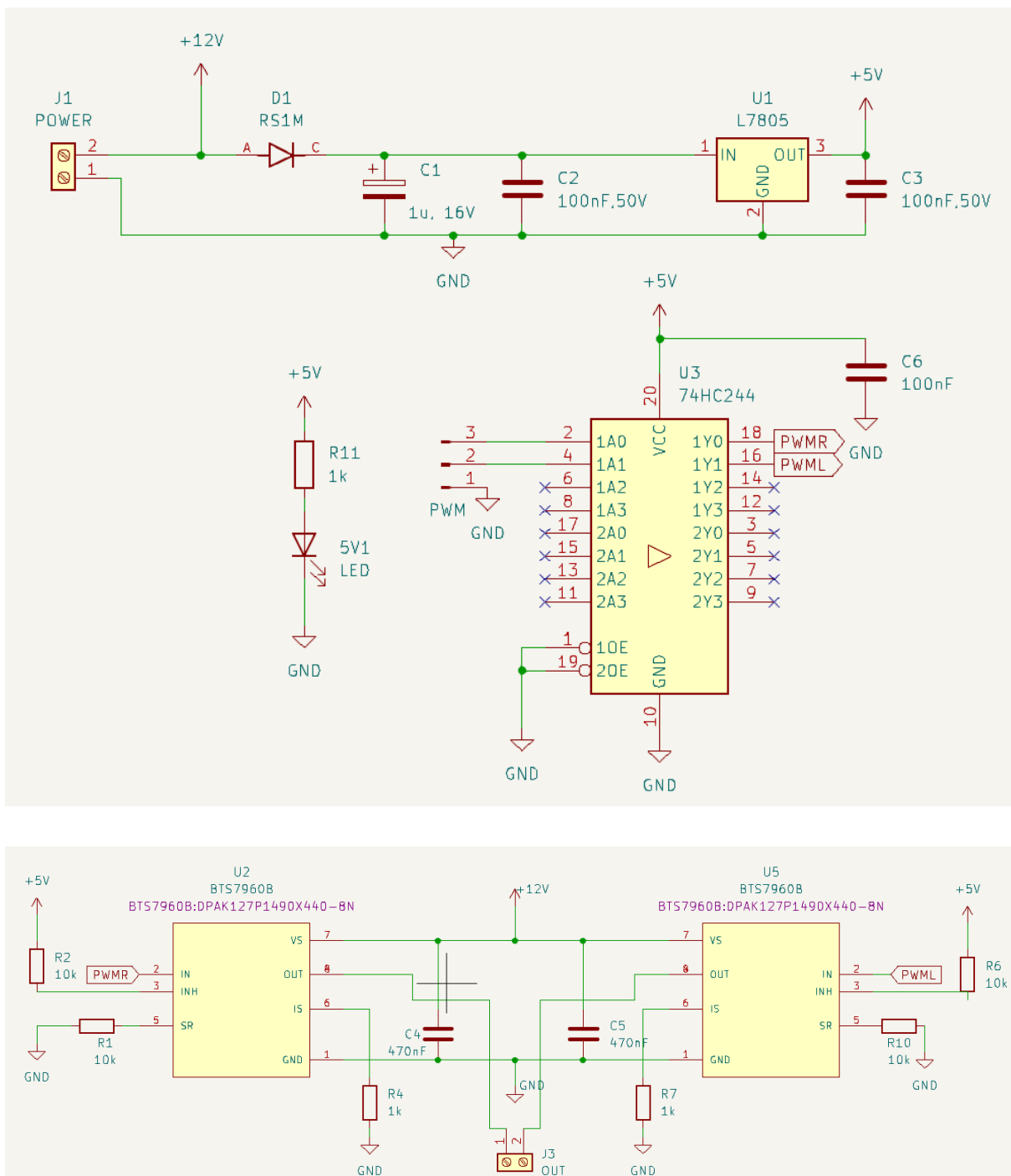
Hình 3: Sơ đồ khối của mạch thiết kế

### Sơ đồ khối nguồn cung cấp (Power diagram)



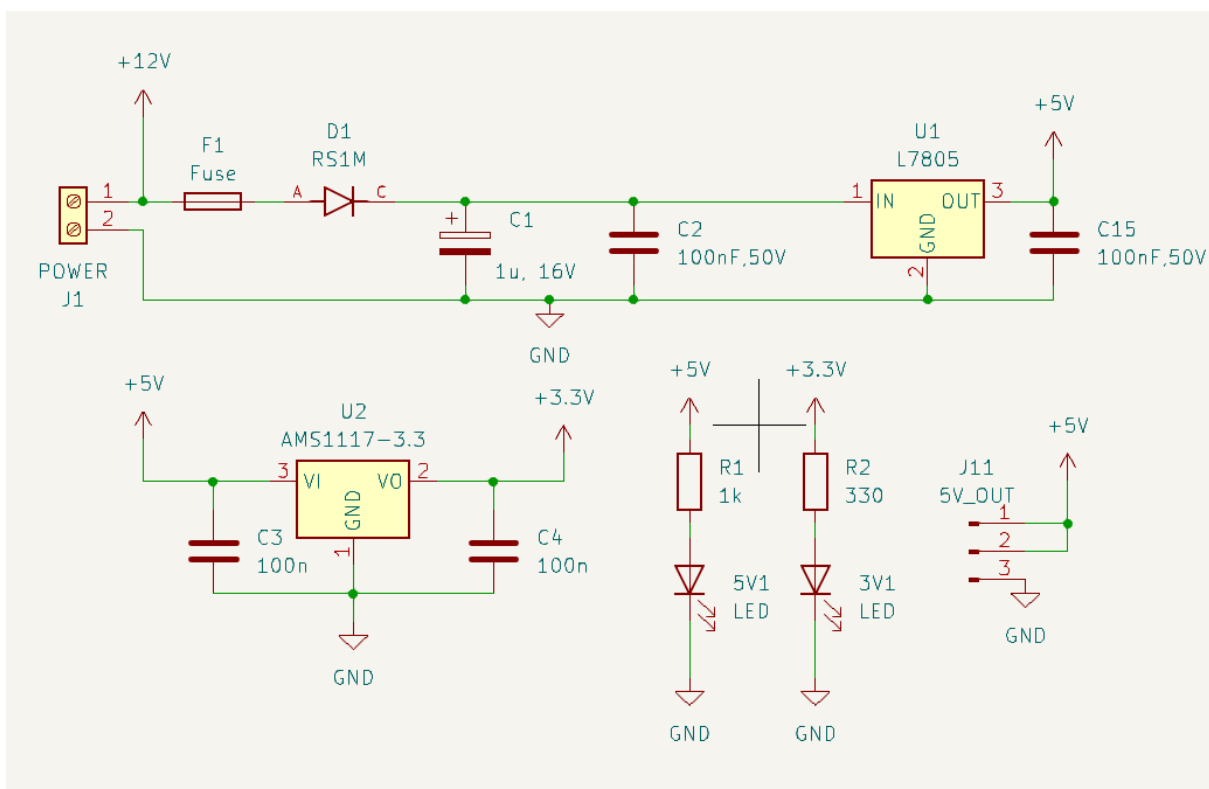
Hình 4: Sơ đồ khối nguồn cung cấp

## Sơ đồ nguyên lí của board mạch cầu H



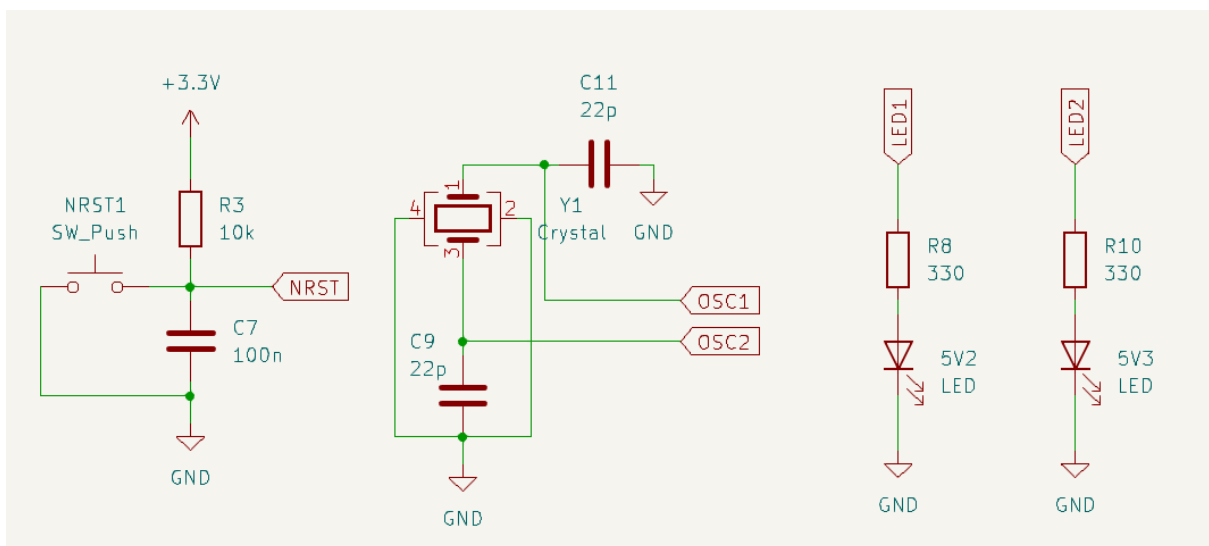
Hình 5: Sơ đồ nguyên lí của mạch cầu H

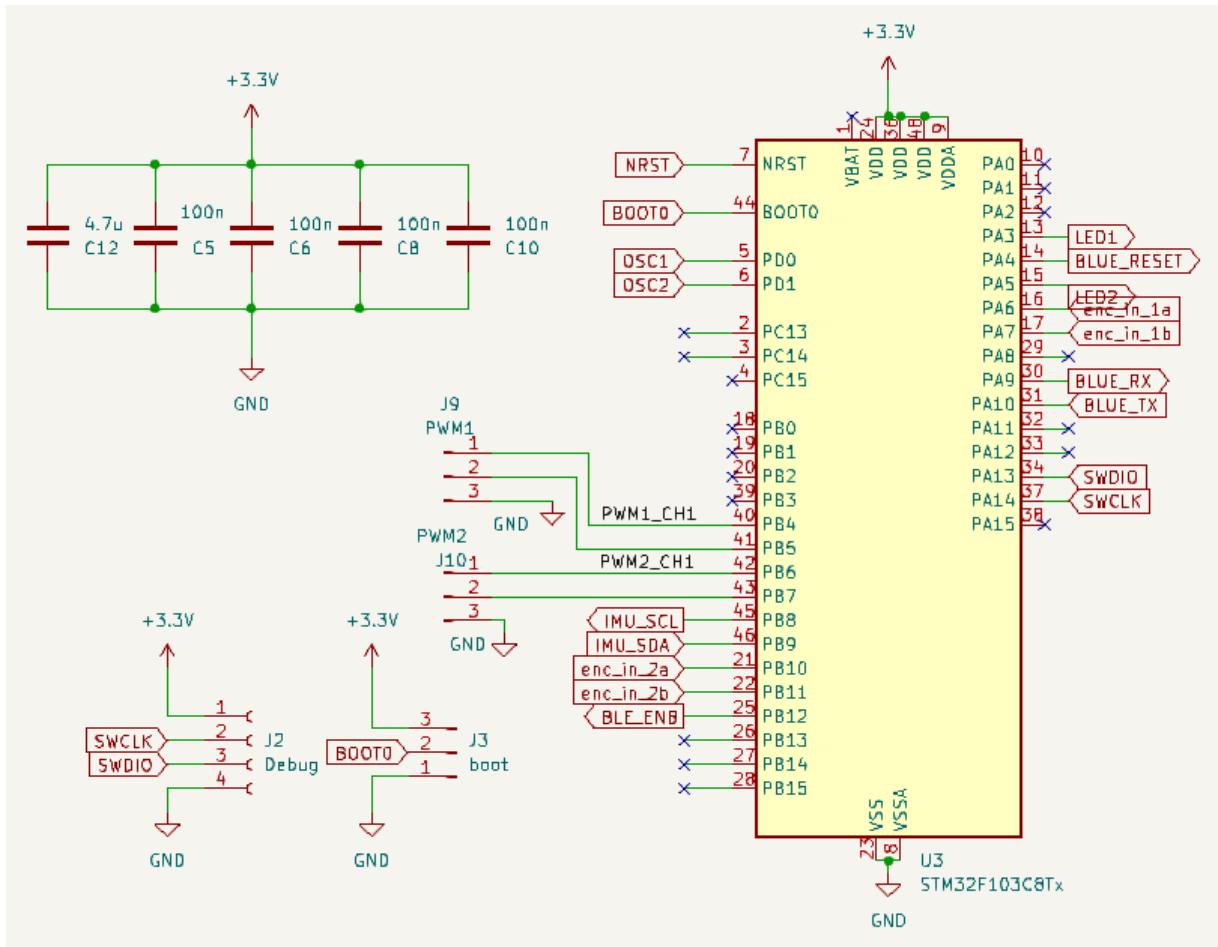
## Sơ đồ nguyên lý khối nguồn của mạch chính



Hình 6: Sơ đồ nguyên lý của khối nguồn

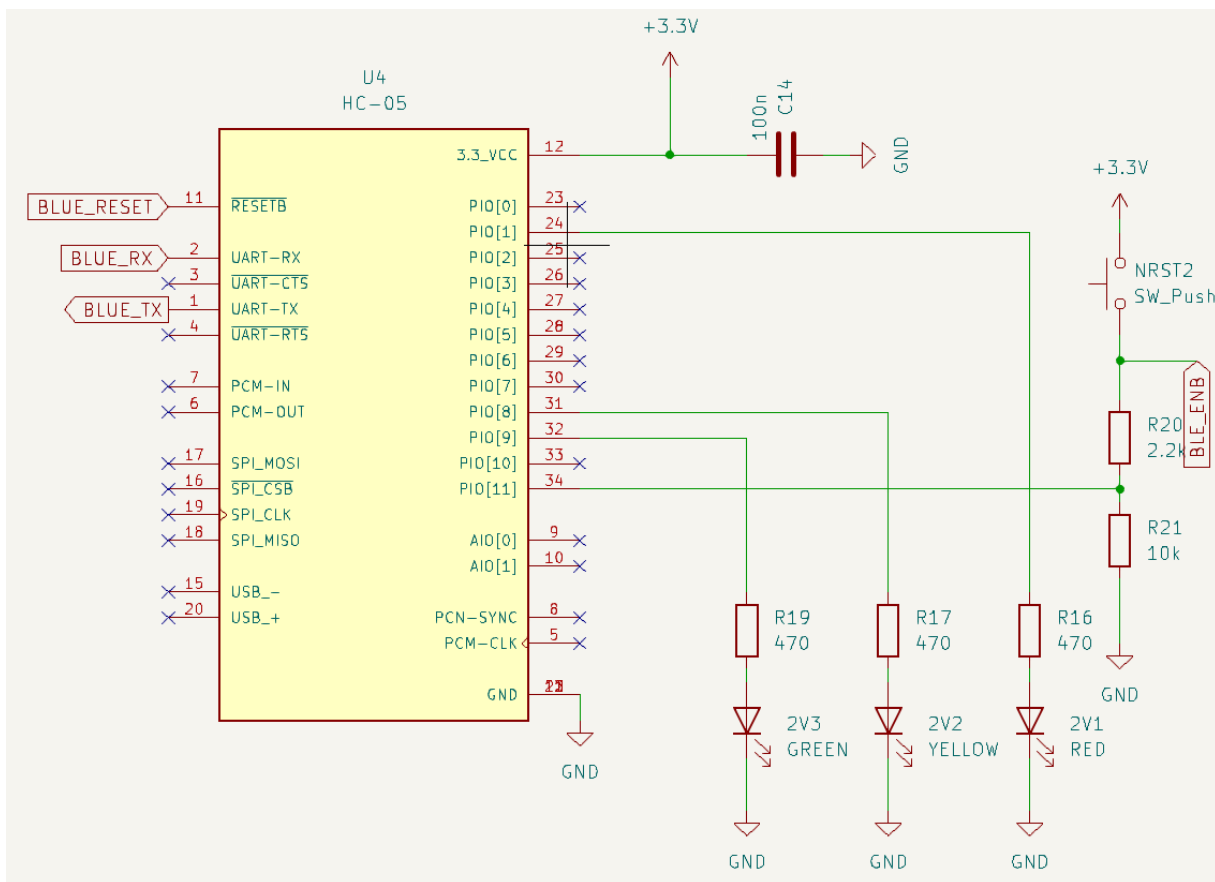
## Sơ đồ nguyên lý khối MCU của mạch chính



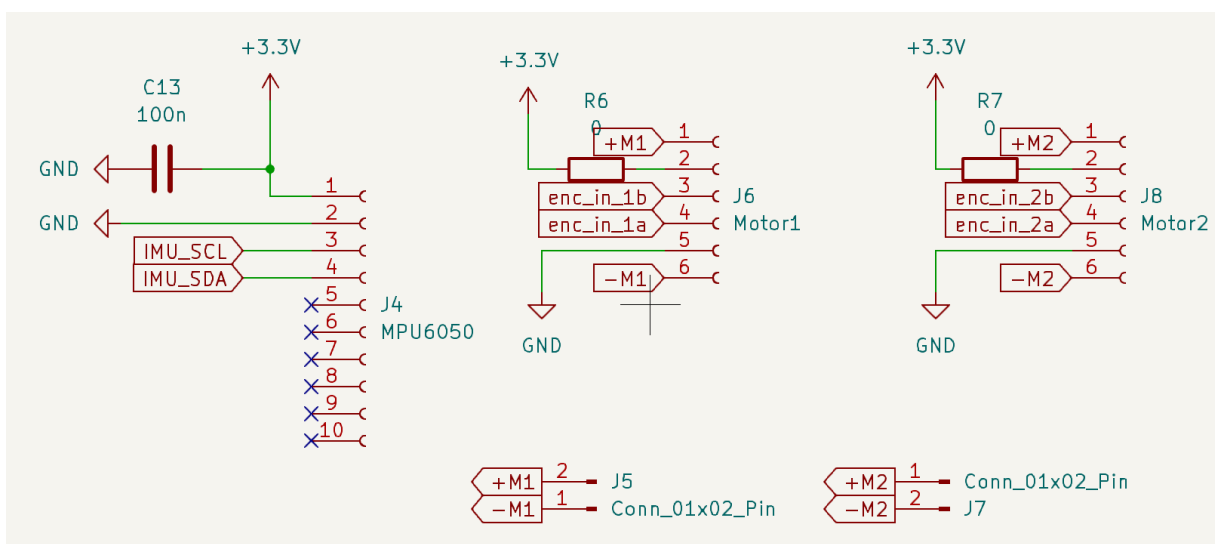


Hình 7: Sơ đồ nguyên lý khối MCU

## Sơ đồ nguyên lý khối Bluetooth của mạch chính



## Sơ đồ nguyên lý khối sensor



Hình 8: Sơ đồ nguyên lý khối sensor



### 2.3.1. Thiết kế mạch nguồn

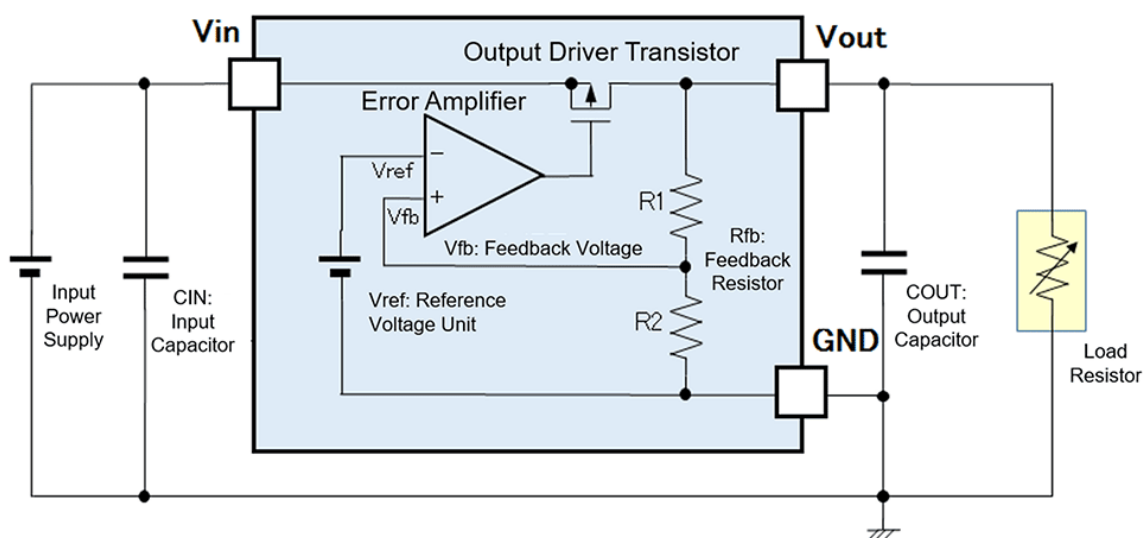
Có 2 loại nguồn ổn áp chính là nguồn tuyến tính (linear regulator) và nguồn xung (switching regulator)

Nên dùng nguồn LDO khi dòng tiêu thụ nhỏ cỡ vài trăm mA, chênh lệch áp ngõ vào và áp ngõ ra không quá lớn (hạn chế không cần thiết tiêu tán trên IC, gây nóng IC), điện áp ngõ vào lớn hơn điện áp ngõ ra (vì nó không thể tăng áp ngõ ra).

Nên dùng nguồn Switching khi không có nguồn LDO có thể đáp ứng dòng tiêu thụ trên tải, mạch yêu cầu hiệu suất nguồn cao và mạch cần tăng áp.

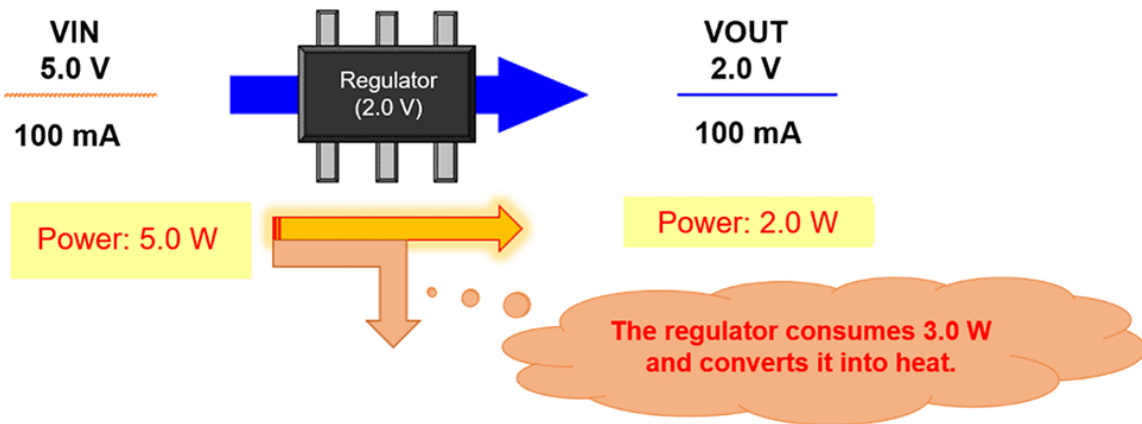
Tóm lại, nguồn switching có hiệu suất cao hơn rất nhiều, có thể kéo tải cao hơn và nó có thể hoạt động tăng áp. Tuy nhiên, nguồn LDO đơn giản hơn, chi phí rẻ hơn, nhiễu thấp hơn và thiết kế đơn giản hơn so với nguồn Switching. Do đó tùy vào mục đích sử dụng ta chọn loại nguồn phù hợp. Với đề tài này thì em chọn thiết kế nguồn LDO.

**Nguồn tuyến tính** (linear regulator) hay thường được gọi là LDO (Low DropOut), một bộ nguồn đơn giản, cho phép tạo một điện áp đầu ra thấp hơn điện áp đầu vào.



Hình 9: Cấu trúc cơ bản của nguồn LDO

## Tính toán công suất nguồn LDO



Hình 10: Hoạt động của nguồn LDO

Hiệu suất của nguồn LDO:

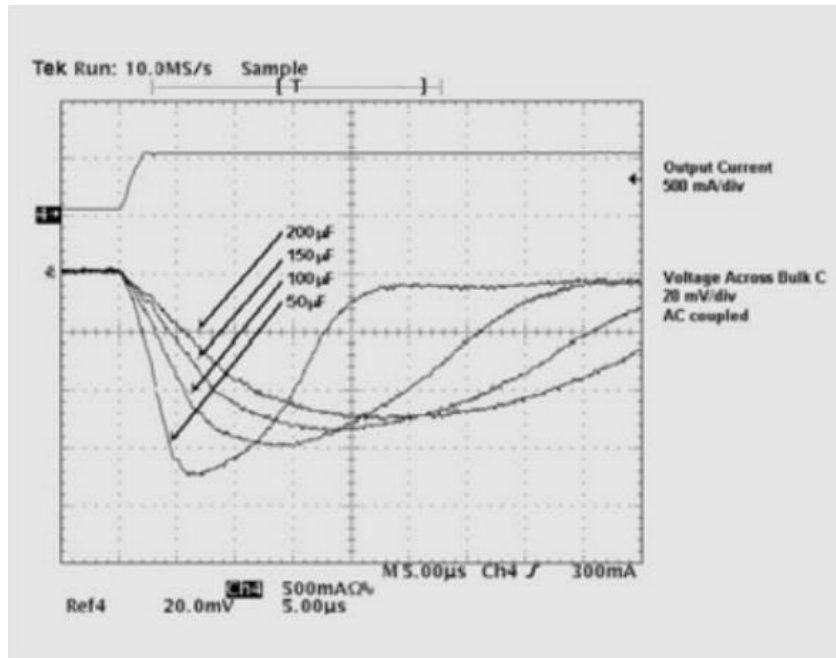
$$I_{out} = I_{in} + I_Q$$

$$H = \frac{I_{out} \cdot V_{out}}{I_{in} V_{in}}$$

Trong đó,  $I_Q$  là dòng điện tĩnh (Quiescent Current) cỡ uA

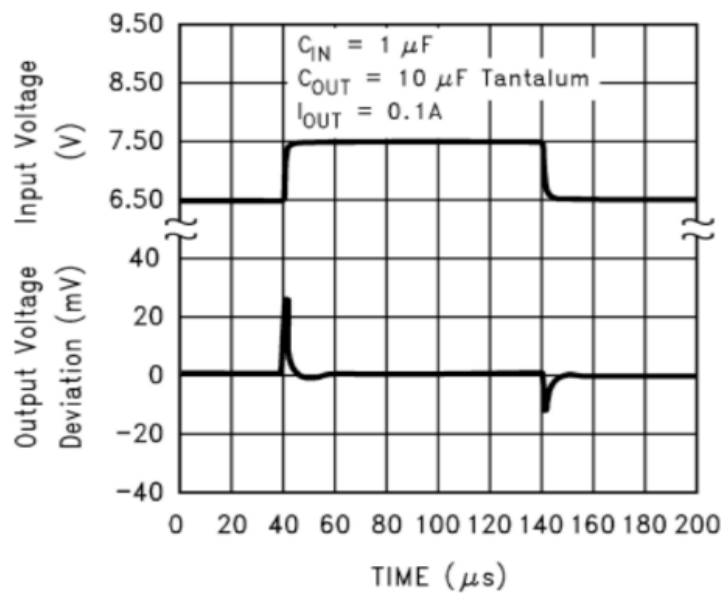
Đáp ứng quá độ của nguồn LDO:

Đáp ứng quá độ của nguồn là do có sự thay đổi của tải hoặc điện áp nguồn đầu vào, khi đó nguồn phải điều chỉnh để điện áp ngõ ra ổn định lại thì thời gian từ lúc  $V_{out}$  thay đổi cho đến khi nó ổn định lại với một sai số cho phép được gọi là thời gian đáp ứng của nguồn. Thứ ta cần quan tâm là hai thông số điện áp dao động và thời gian đáp ứng.



Hình 11: Đáp ứng quá độ của nguồn LDO khi tụ thay đổi

Dựa vào hình trên ta có thể thấy quá trình đáp ứng phụ thuộc vào giá trị của tụ lọc ở ngõ ra  $C_{out}$ . Giá trị tụ càng lớn thì điện áp dao động càng nhỏ và thời gian đáp ứng càng chậm và ngược lại. Có 3 loại tụ thường được sử dụng với nguồn LDO là tụ gốm, tụ nhôm và tụ tantalum.



Hình 12: Đáp ứng của LM1117 khi đầu vào có sự thay đổi

Có thể thấy thời gian đáp ứng khi có sự thay đổi ngõ vào của nguồn LDO cỡ chục  $\mu s$ .

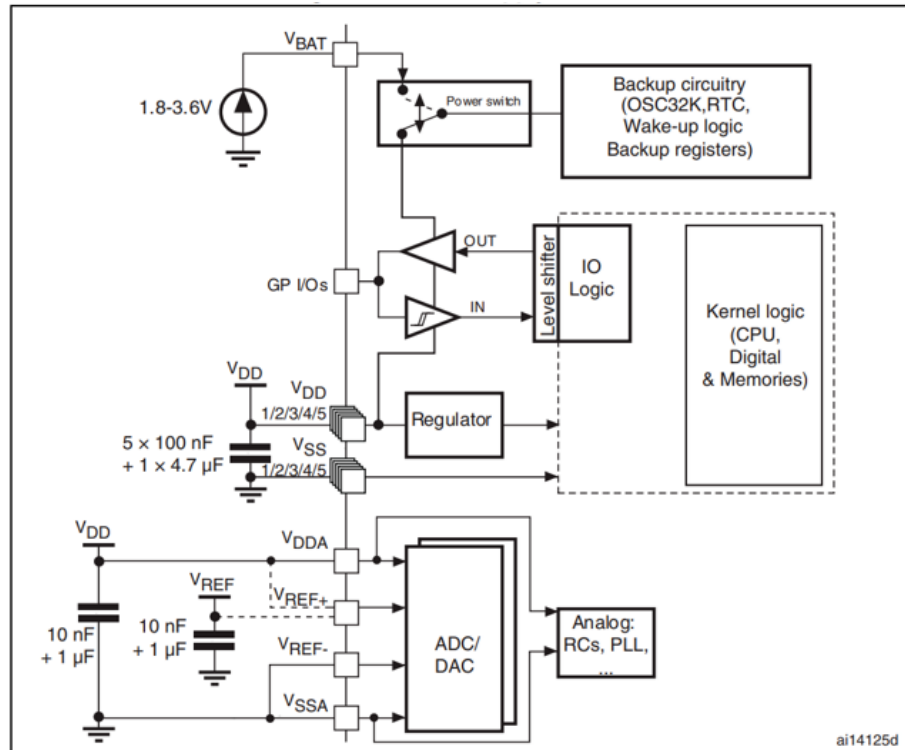
### Bảo vệ mạch nguồn:

Bảo vệ quá dòng: khi PCB bị đoản mạch hoặc tải sử dụng dòng điện quá mức cho phép có thể gây quá tải và hỏng nguồn điện. Trong trường hợp này, ta nên mắc thêm một cầu chì, với giá trị của cầu chì phải cao hơn dòng điện tối đa mà phần cứng tiêu thụ nếu không thì cầu chì sẽ thường xuyên ngắt. Để hạn chế thay cầu chì, nên em chọn sử dụng cầu chì tự phục hồi.

### 2.3.2. Thiết kế mạch MCU

#### Tụ decoupling và tụ bypass

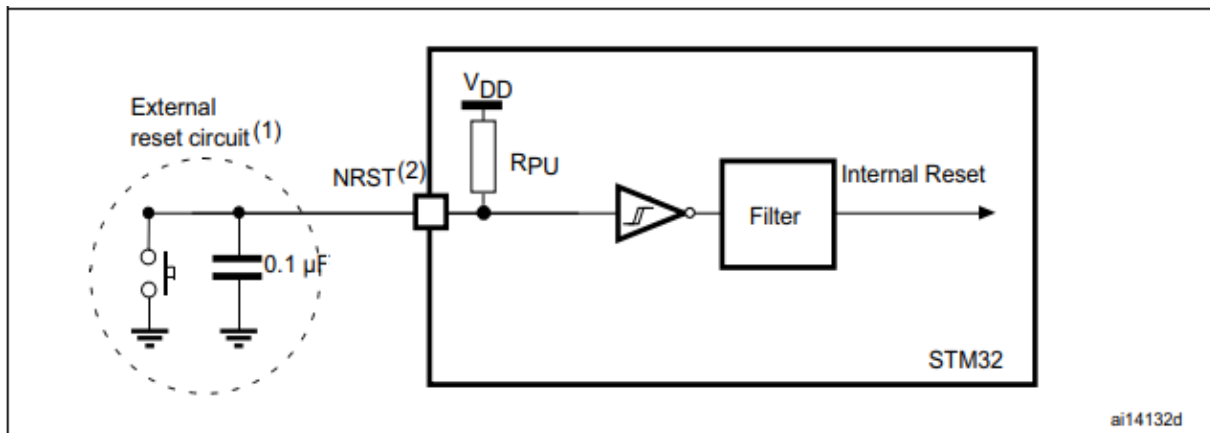
Tụ decoupling duy trì nguồn ổn định hơn, không để nguồn sụt áp quá nhanh. Tụ bypass lọc nhiễu nguồn tần số cao, tránh các gai điện áp. Giá trị tụ xem trong datasheet của MCU.



Hình 13: Sơ đồ cấp nguồn của STM32F103C8T6

**Nút nhấn Reset:** đây là một trong những bắt buộc khi thiết kế mạch MCU. Thông thường thiết kế nút bấm reset cho MCU cũng như bao nút bấm input khác sẽ bao gồm một trở kéo lên 10K và một tụ chống rung 100nF đối với RST mức thấp.

Tuy nhiên, STM32F103C8T6 có hỗ trợ trở kéo lên nội nên chỉ cần tụ là đủ.



Hình 14: Mạch nút ấn reset cho SMT32F103C8T6

### Nạp code cho MCU:

STM32F103C8T6 hỗ trợ 2 chuẩn nạp là JTAG và SWD. Em chọn sử dụng Stlink chuẩn SWD nên khi thiết kế header nạp code sẽ bao gồm tín hiệu SWDIO, SWDCLK và 2 dây nguồn.

### Chọn BOOT:

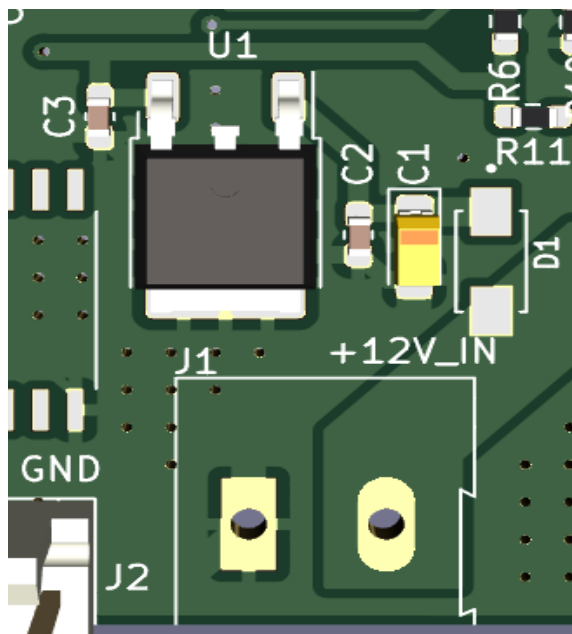
STM32F103C8T6 có các modes như sau:

Boot mode selection pins		Boot mode	Aliasing
BOOT1	BOOT0		
x	0	Main Flash memory	Main Flash memory is selected as boot space
0	1	System memory	System memory is selected as boot space
1	1	Embedded SRAM	Embedded SRAM is selected as boot space

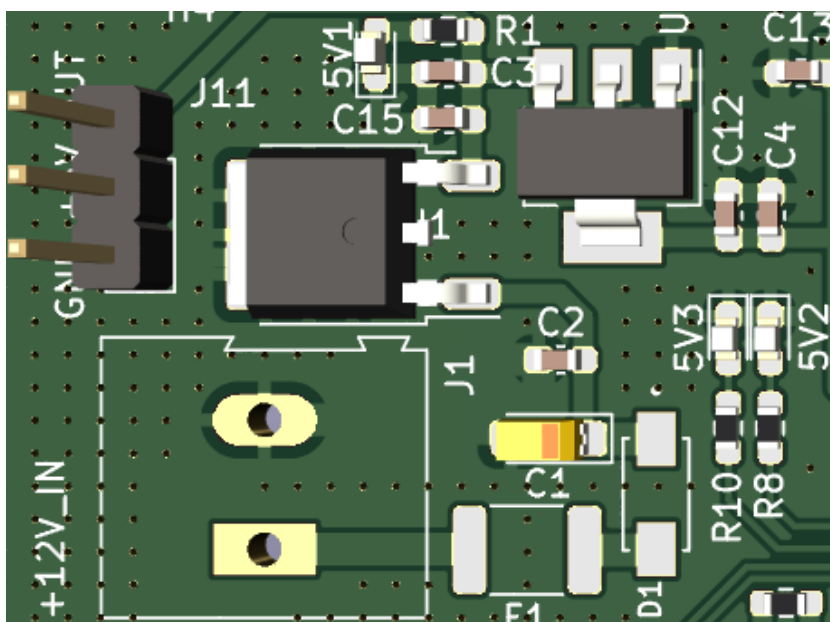
Hình 15: Các chế độ BOOT của STM32F103C8T6

## 2.4. Nội dung 2: Thiết kế mạch PCB

**Layout nguồn LDO:** đặt tụ lọc nguồn ngay trước ngõ vào và ngõ ra của IC, nguồn cung cấp cho linh kiện khác lấy ra sau tụ lọc Cout.

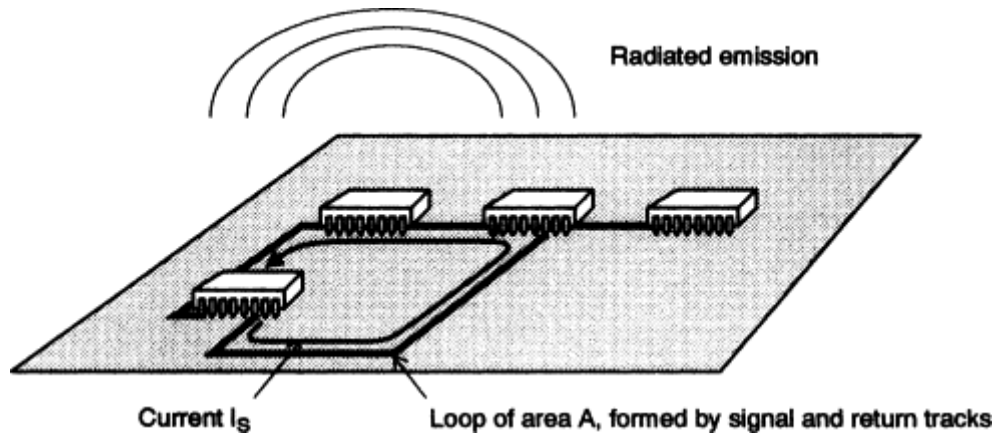


Hình 16: Layout khối nguồn của mạch cầu H

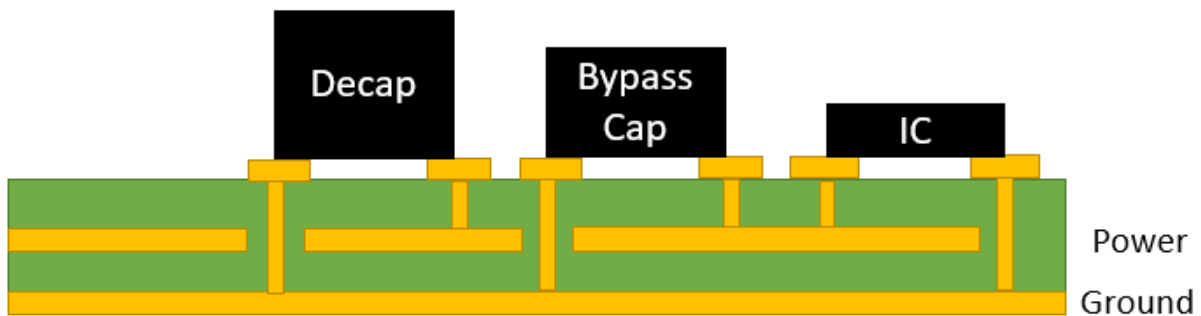


Hình 17: Layout khối nguồn của mạch chính

Đặt linh kiện sao cho vòng từ Vout về feedback là ngắn nhất, tức là đi dây để diện tích vòng từ chân nguồn về chân GND tín hiệu là nhỏ nhất, đặt biệt với các tín hiệu tần số cao và dòng lớn. Nó liên quan đến bức xạ phát xạ từ đường dây đó.

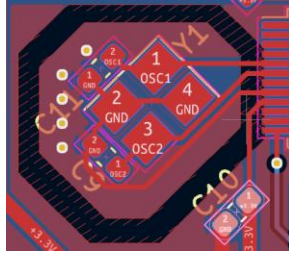


**Layout tụ decoupling và tụ bypass:** đặt tụ decoupling và tụ bypass càng gần IC càng tốt, trong đó tụ bypass đặt gần IC hơn



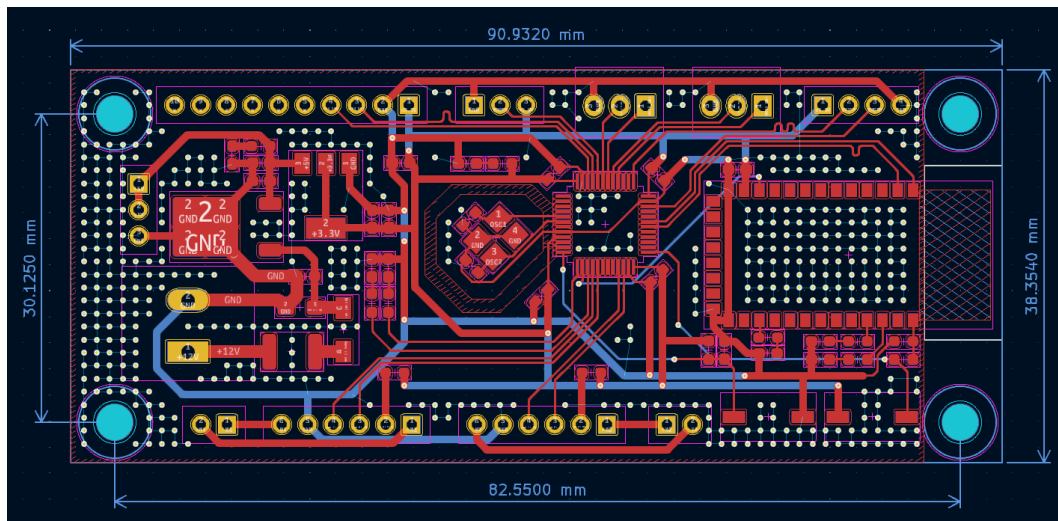
Hình 18: Vị trí đặt tụ decoupling và tụ bypass

**Layout thạch anh:** Đặt thạch anh càng gần chân XTAL của IC càng tốt, tránh đặt thạch anh gần các linh kiện, đường dây tần số cao, đặt tụ liên kết với thạch anh gần chân thạch anh, đường dây kết nối IC, thạch anh, tụ càng ngắn, càng rộng càng tốt. Nên cắt GND cho thạch anh, lí do là các tín hiệu khác có thể sẽ không đo ngang vùng có thạch anh, tuy nhiên đường return pad (GND) của nó có thể đi ngang qua vùng này, gây nhiễu cho thạch anh và ngược lại. Việc cắt GND sẽ làm đường return pad không đi vào vùng này.



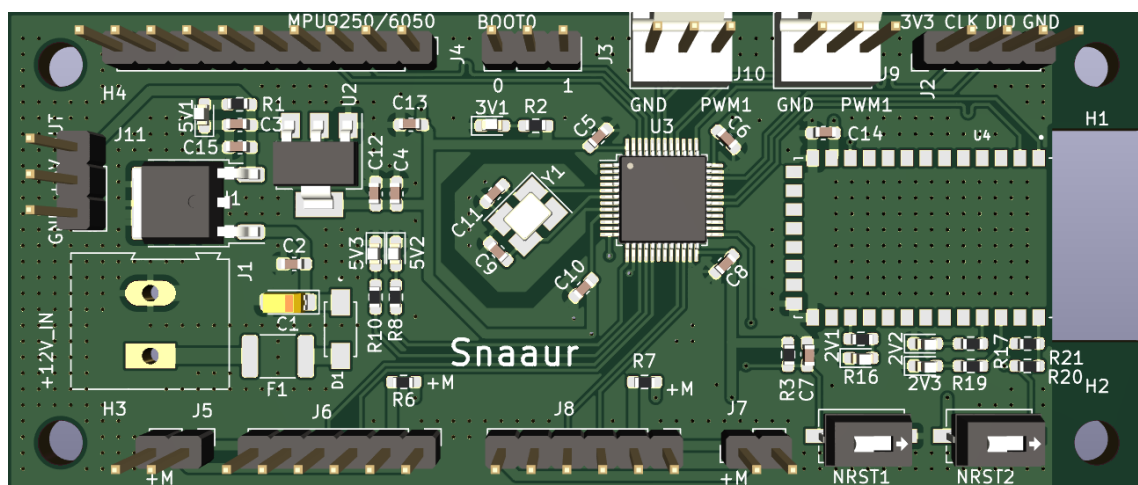
Hình 19: Layout khối thạch anh của mạch chính

Layout và đi dây của mạch chính:



Hình 20: Layout và đi dây của mạch chính

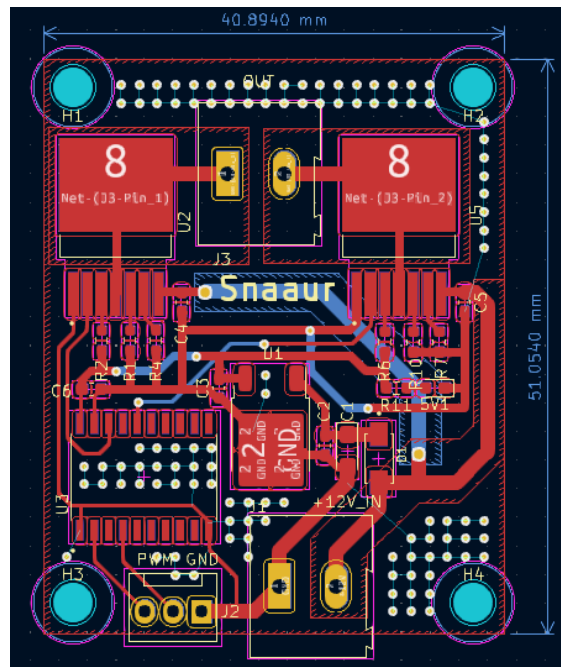
### 3D View của mạch chính:



Hình 21: 3D View của mạch chính

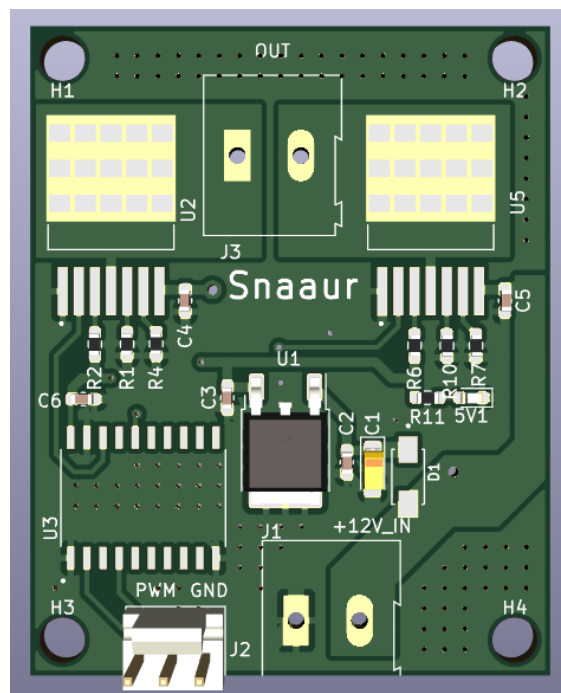


Layout và đi dây của mạch cầu H:



Hình 22: Layout và đi dây của mạch cầu H

3D View của mạch cầu H:



Hình 23: 3D View của mạch cầu H

## 2.5. Nội dung 3: Lập trình State Machine

### 2.5.1. Lý thuyết

Trong lập trình nhúng, bản chất là chúng ta lập trình cho một thiết bị hoạt động, khi thiết bị hoạt động sẽ có rất nhiều trạng thái (hoặc các mode hoạt động) cần phải chuyển đổi qua lại với nhau. State machine là việc chúng ta định nghĩa ra các trạng thái hoạt động đó (định nghĩa các mode hoạt động) và lập trình theo thiết bị hoạt động theo các trạng thái mà chúng ta định nghĩa.

Các ưu điểm khi lập trình State Machine:

Khi sử dụng State Machine chúng ta sẽ dễ dàng tiếp cận các vấn đề hơn, code được mạch lạc, logic hơn, đồng thời mọi người sẽ dễ hiểu hơn mô hình hoạt động của thiết bị của chúng ta, dễ qua đó dễ sửa đổi, bảo trì và nâng cấp thiết bị. Đặc biệt, với trạng thái máy, chúng ta có thể dễ dàng tạo ra được các tiến trình bất đồng bộ (Các chức năng chạy song song với nhau).

Việc chia ra các trạng thái còn được ứng dụng để tối ưu hóa tài nguyên (phần cứng, phần mềm) tốt hơn. Cụ thể, tại mỗi trạng thái, thiết bị chỉ cung cấp các tài nguyên đủ để thực hiện các hoạt động trong trạng thái đó, còn các tài nguyên khác sẽ được đưa vào trạng thái nghỉ.

Một State Machine có thể được mô tả bởi các thành phần sau:

Một tập các trạng thái của hệ thống.

Trạng thái khởi tạo.

Tập hợp các sự kiện Input.

Tập hợp các sự kiện Output.

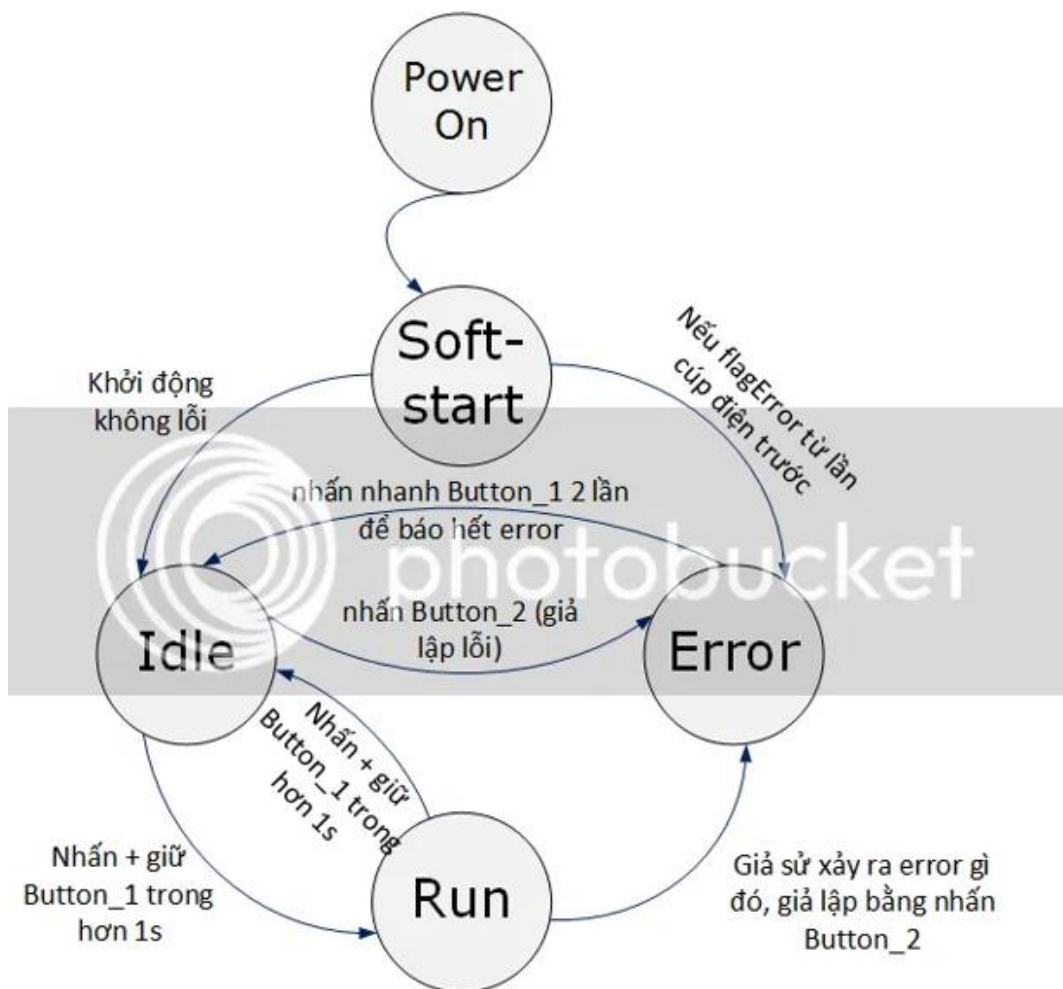
Tập hợp các hành động hay các sự kiện output ứng với mỗi trạng thái của hệ thống (state event handler).

Các sự kiện chuyển trạng thái (state transtion).

Để cài đặt State Machine thông thường sẽ định nghĩa ra các trạng thái của hệ thống. Trong mỗi vòng lặp sẽ có sử dụng switch() để check trạng thái hiện tại của hệ thống và các điều kiện chuyển tiếp. Nếu điều kiện chuyển tiếp thỏa mãn thì gọi hàm xử lý và update sang trạng thái mới.

### 2.5.2. Thực hành

Đề bài:



Hình 24: Đề bài State Machine

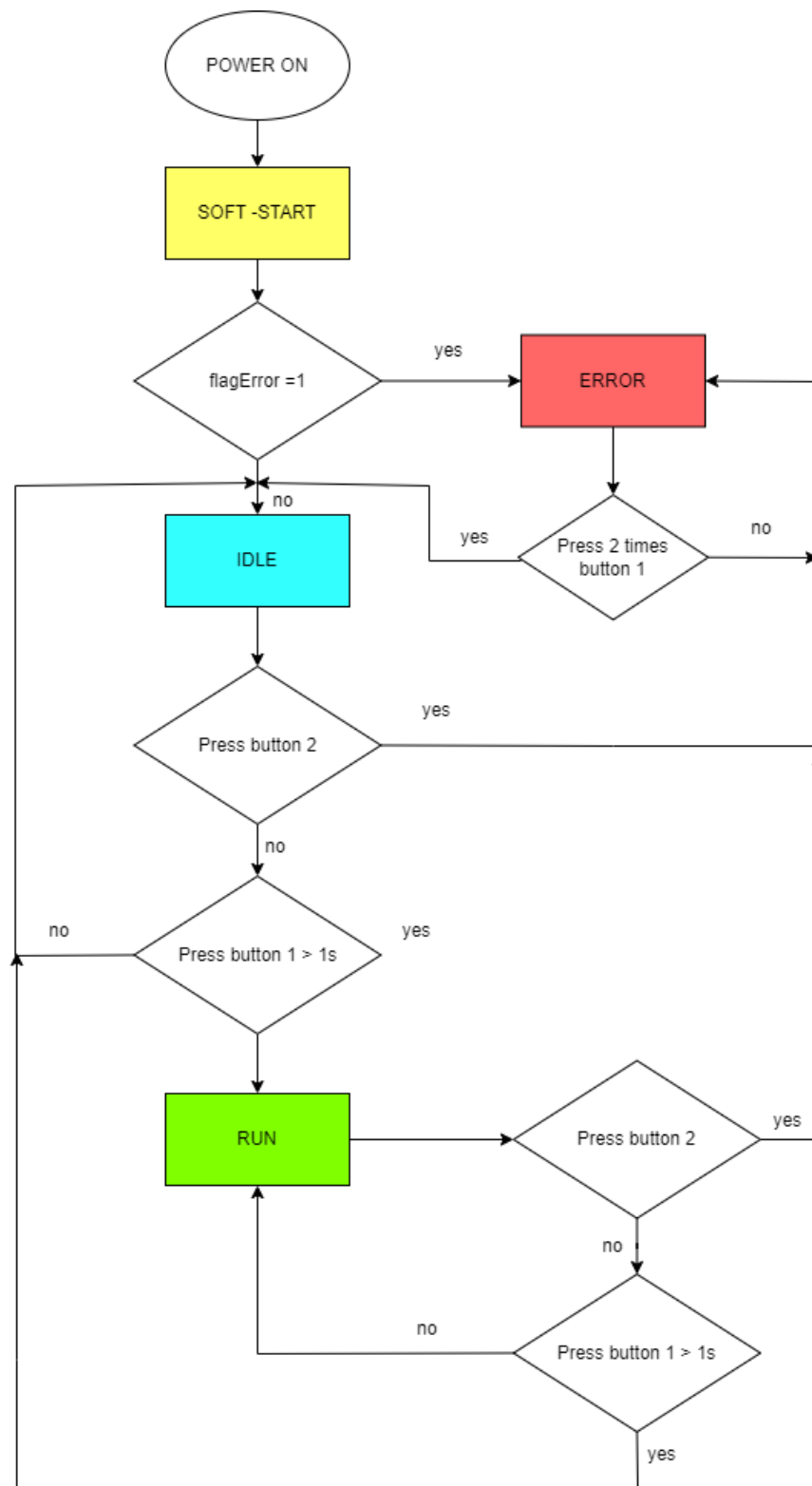
Các trạng thái hiển thị:

<b>Indicated LEDs Table</b>		
<b>State</b>	<b>LED1</b>	<b>LED2</b>
Soft-start	Fast Blinky	Off
Idle	Slow Blinky	Off
Run	Always ON	PWM Control
Error	Fast Blinky	Fast Blinky

Công việc của từng trạng thái:

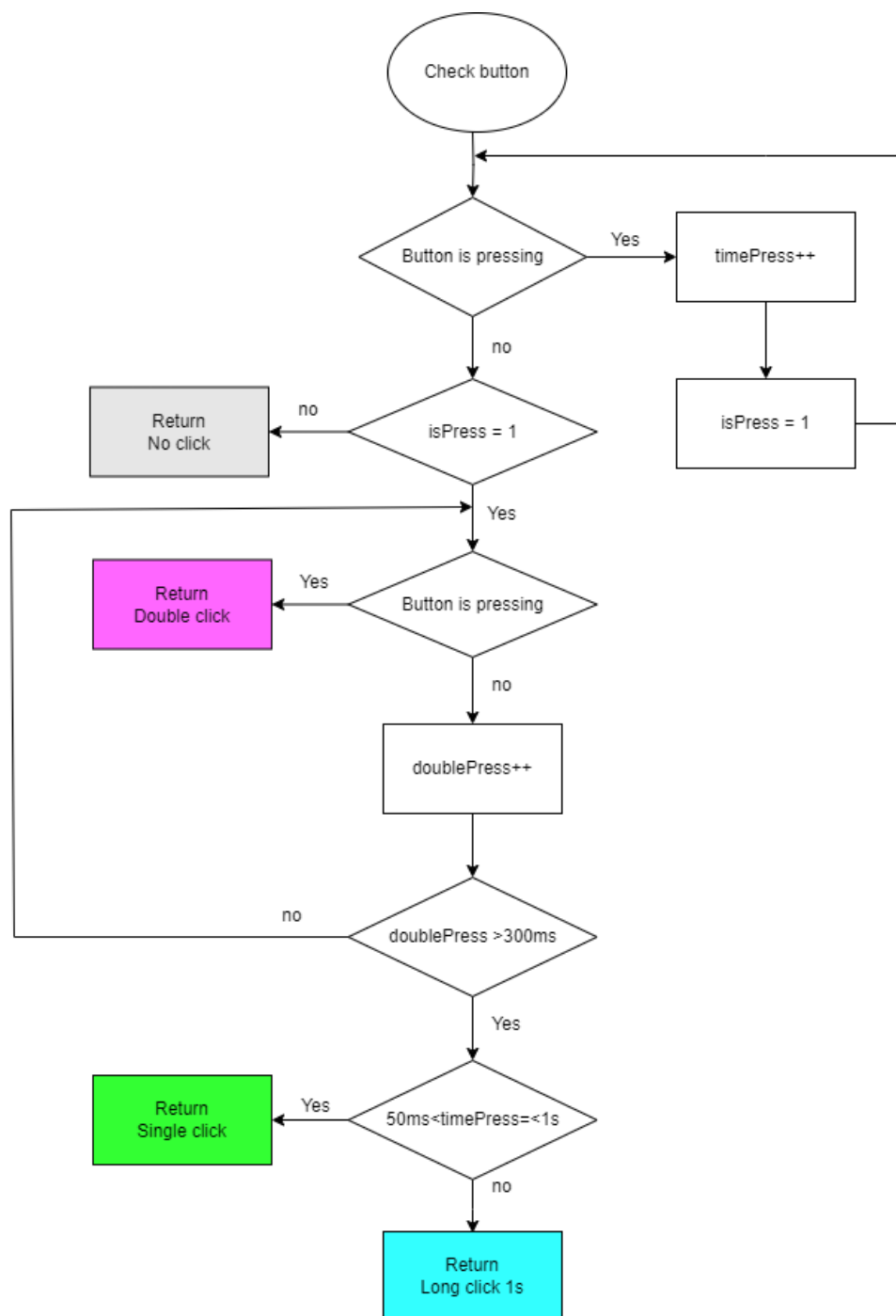
<b>Task Table</b>	
<b>State</b>	<b>Task</b>
Soft-start	Khởi động các mô-đun liên quan, check flagError trong EEPROM Nếu không có Error chuyển sang Idle State
Idle	Đợi flagRun từ UART hoặc Button_1
Run	Điều khiển PWM cho LED2 sáng dần - tắt dần
Error	Báo Error; đợi xóa Error (khi xóa error thì flagError = 0) Lưu giá trị flagError hiện thời vào EEPROM
<i>Tất cả mọi trạng thái đều được chỉ thị bằng LEDs và gửi qua UART.</i>	

**Flow chart của State Machine:**



Hình 25: Flow chart của State Machine

Flow chart đọc nút nhấn:



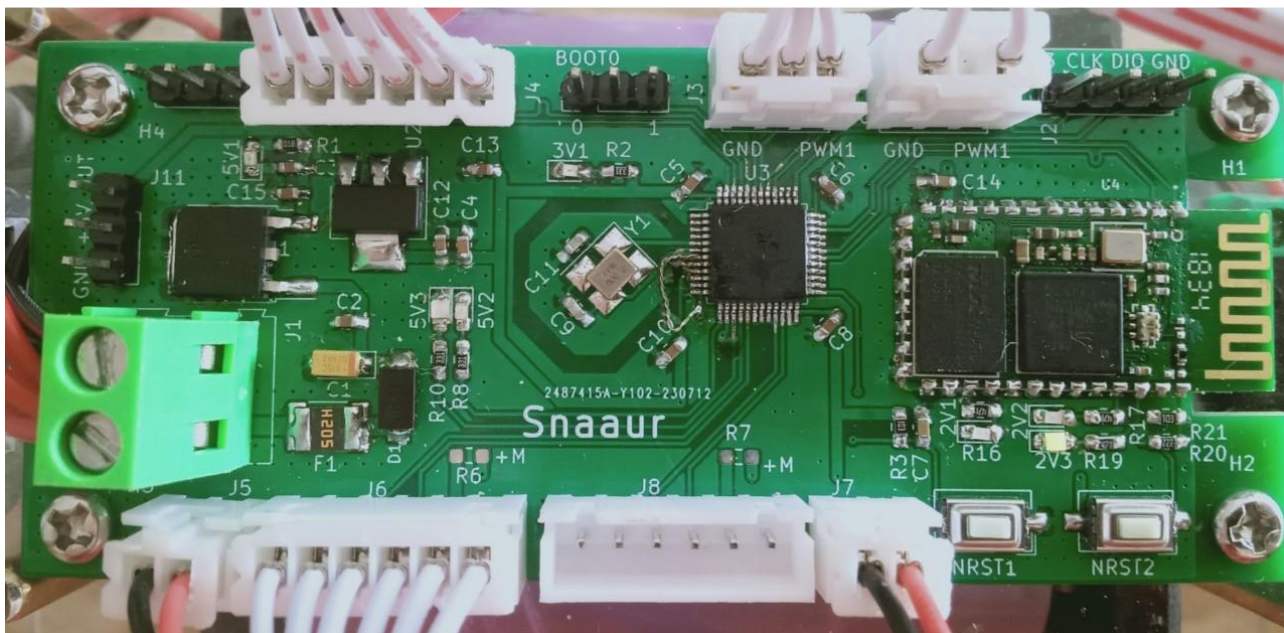
Hình 26: Flow chart đọc nút nhấn

Code được up lên Github với đường dẫn trong phần phụ lục.



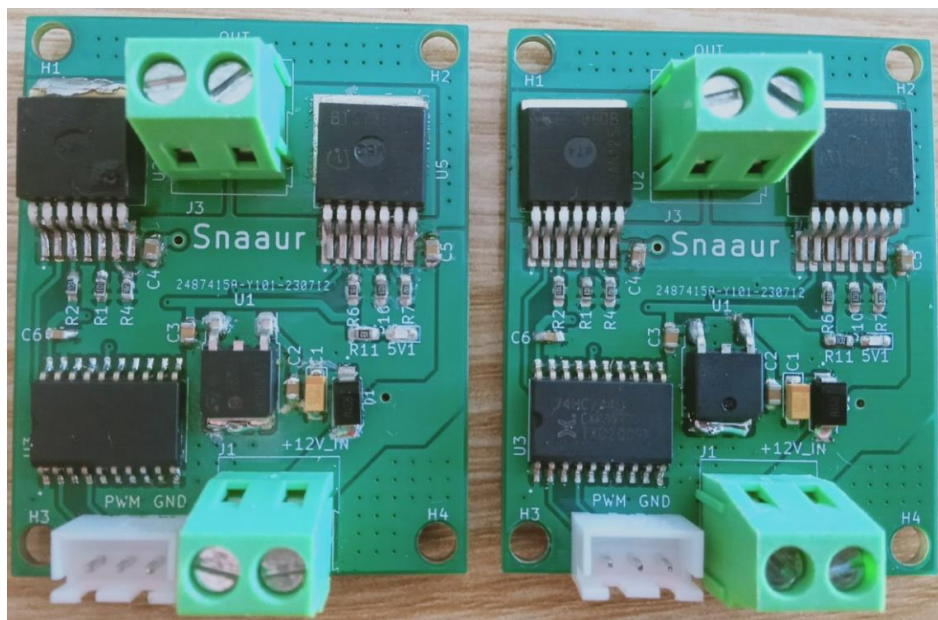
## 2.6. Nội dung 4: Hàn mạch, chuẩn bị phần cơ khí và lắp xe

Sản phẩm hoàn thiện của mạch chính:



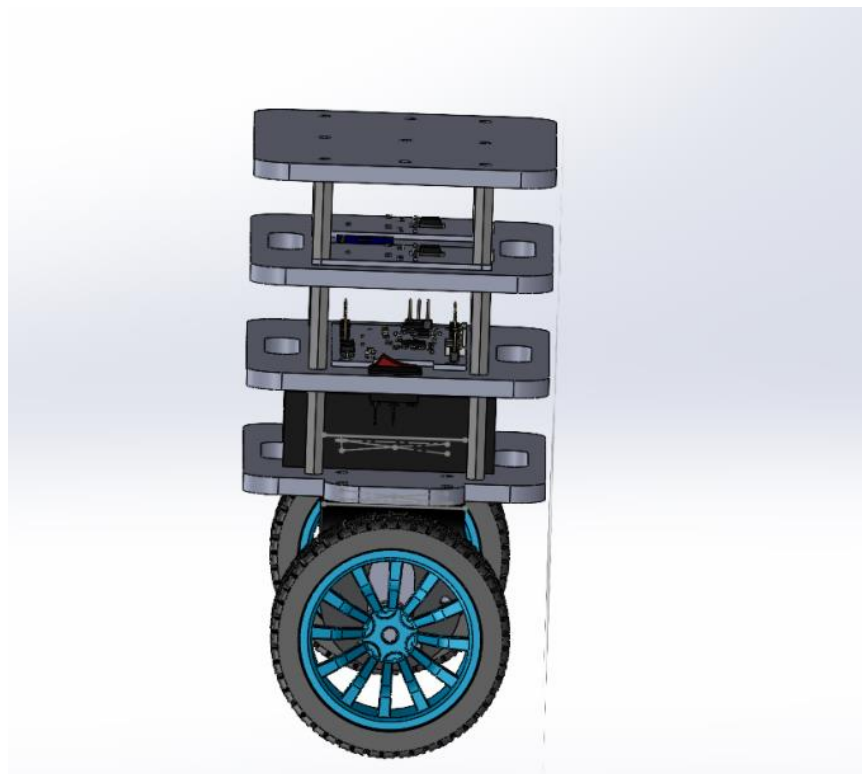
Hình 27: Sản phẩm hoàn thiện của mạch chính

Sản phẩm hoàn thiện của mạch cầu H:



Hình 28: Sản phẩm hoàn thiện của mạch cầu H

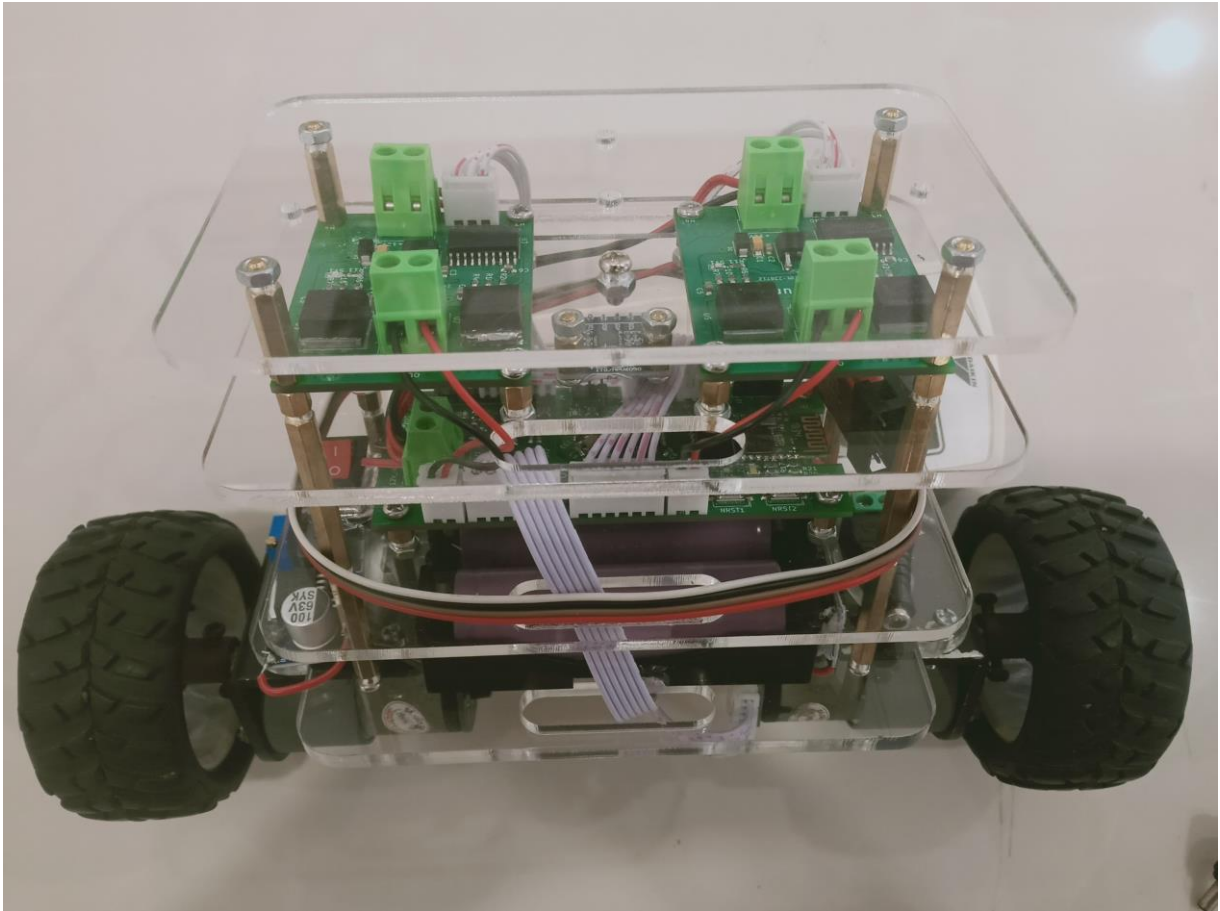
Phần cơ khí của Self Balancing Robot:



*Hình 29: 3D view của Self Balancing Robot trên SolidWork*



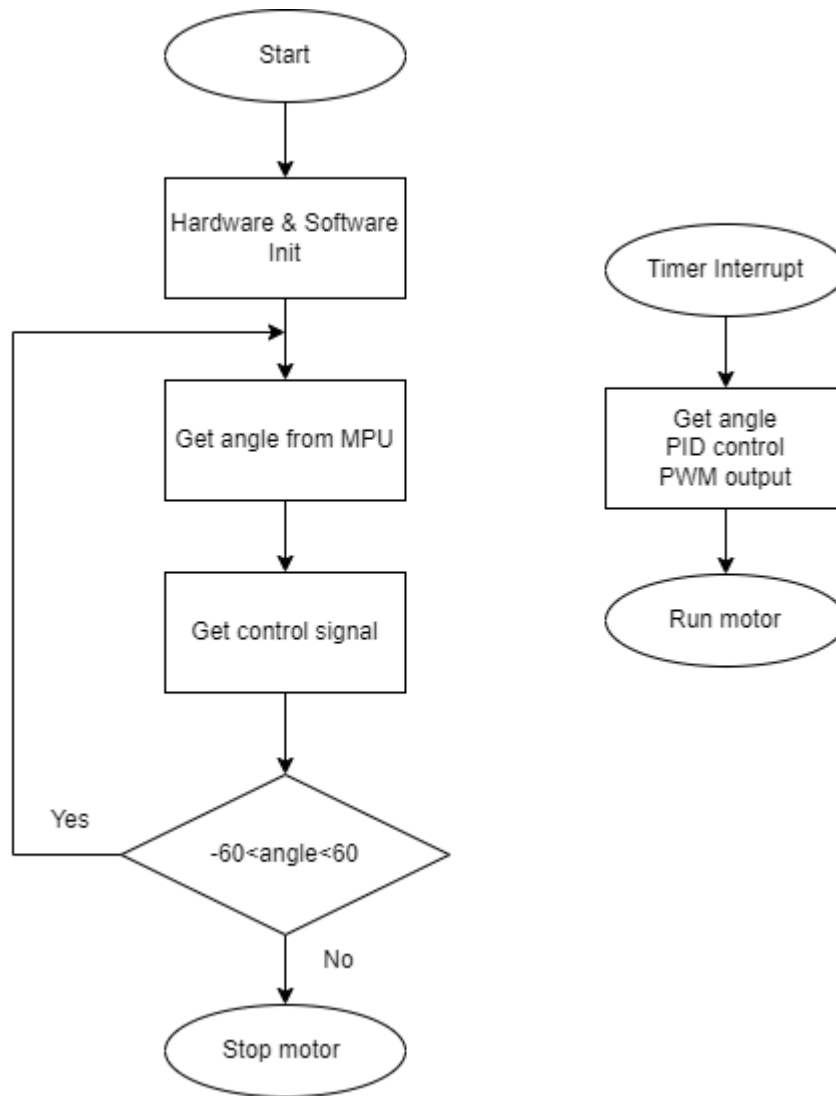
Self Balancing Robot hoàn thiện:



*Hình 30: Self Balancing Robot*

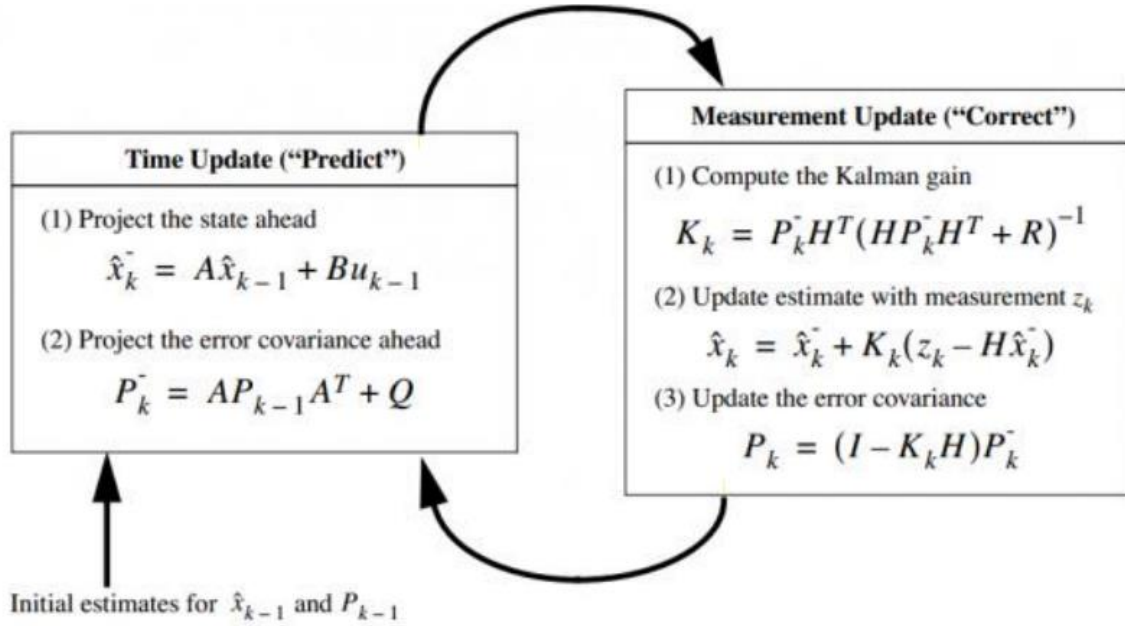
## 2.7. Nội dung 5: Các giải thuật để lập trình cân bằng cho Self Balancing Robot

Flow chart của Self Balancing Robot:



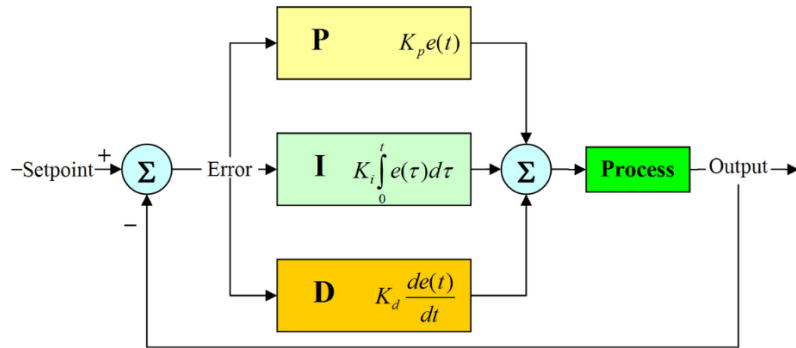
Hình 31: Flow chart của Self Balancing Robot

Sử dụng MPU6050 và áp dụng bộ lọc Kalman để xác định góc nghiêng của xe:



Hình 32: Quy trình tính toán của bộ lọc Kalman

Sử dụng bộ điều khiển PID để điều khiển góc ở vị trí cân bằng:



\* Bộ điều khiển PID liên tục:

$$G_{PID}(s) = K_P + \frac{K_I}{s} + K_D s$$

\* Bộ điều khiển PID rời rạc:

$$G_{PID}(z) = K_P + \frac{K_I T}{2} \frac{z+1}{z-1} + \frac{K_D}{T} \frac{z-1}{z}$$

hoặc

$$G_{PID}(z) = K_P + K_I T \frac{z}{z-1} + \frac{K_D}{T} \frac{z-1}{z}$$

Hình 33: Mô hình và công thức bộ điều khiển PID

Chuyển từ miền rời rạc (Z) sang miền thời gian ta có công thức sau:

$$u_k = u_{k1} + K_p.(e_k - e_{k1}) + \frac{1}{2}.K_i.T.(e_k + e_{k1}) + \frac{1}{T}K_d.(e_k - 2.e_{k1} + e_{k2})$$

Code được up lên Github với đường dẫn trong phần phụ lục.

## **PHẦN 3: TỔNG KẾT CÔNG VIỆC THỰC TẬP**

### **3.1. Kết quả công việc thực tập**

Sau 8 tuần thực tập tại công ty, mặc dù đôi khi vẫn có gặp khó khăn (trong cách tiếp cận vấn đề mới, một số công cụ sử dụng...) nhưng sự giúp đỡ rất nhiệt tình của các anh, chị và các bạn là động lực to lớn để bản thân em hoàn thành tốt nhất có thể những công việc được giao đúng hạn. Các nhiệm vụ được giao liên quan đến ngành Embedded giúp em có cái nhìn trực quan và hiểu rõ hơn về nó.

### **3.2. Kinh nghiệm học được sau khi thực tập**

Sau khoảng thời gian thực tập tại công ty bản thân đã học hỏi được rất nhiều kinh nghiệm từ các anh/chị đi trước. Những bài học, kinh nghiệm rất hữu ích cho công việc sau này.

Kỹ năng mềm: kỹ năng làm việc nhóm, kỹ năng thuyết trình, kỹ năng ứng xử trong công ty, kỹ năng giải quyết vấn đề...

Hiểu thêm về các hệ thống nhúng và cách xử lý các vấn đề chuyên môn

Rèn luyện thêm kỹ năng lập trình, kỹ năng vẽ PCB...

### **3.3. Kế hoạch nghề nghiệp trong tương lai**

Hoàn thành xong chương trình Đại Học và cố gắng học hỏi, trau dồi thêm nhiều kiến thức từ trên giảng đường.

Rèn luyện thêm kỹ năng tiếng Anh chuyên ngành.

Rèn luyện thêm các kỹ năng mềm: kỹ năng thuyết trình trước đám đông, kỹ năng giao tiếp, ứng xử trong công ty, kỹ năng làm việc nhóm, kỹ năng giải quyết vấn đề...

Trong khoảng thời gian thực tập bản thân đã được trải nghiệm môi trường làm việc thực tế ở công ty và cảm nhận được Công ty TNHH công nghệ PIF Lab chính là môi trường rất tốt để bản thân tham gia trau dồi học hỏi, làm việc và cống hiến sau này.

## PHẦN 4: TÀI LIỆU THAM KHẢO

- [1] KiCad Docs 7.0, [online]. Available: [Documentation | KiCad](#)
- [2] Nisshinbo Micro Devices Inc, Power Management IC Basic Vol. 2, “What Is a Linear Regulator (LDO Regulator)”, [online]. Available: <https://www.nisshinbo-microdevices.co.jp/en/design-support/basic/02-linear-regulator.html>
- [3] Brian M. King, Advanced Analog Products, ” Understanding the load-transient response of LDOs”, Texas Instruments Incorporated, [online]. Available: [https://www.ti.com/lit/an/slyt151/slyt151.pdf?ts=1680867564532&ref\\_url=https%253A%252F%252Fwww.google.com%252F](https://www.ti.com/lit/an/slyt151/slyt151.pdf?ts=1680867564532&ref_url=https%253A%252F%252Fwww.google.com%252F)
- [4] Zachariah Peterson, “Decoupling Capacitor and Bypass Placement Guidelines”, update September 25, 2020, [online]. Available: <https://resources.altium.com/p/bypass-and-decoupling-capacitor-placement-guidelines>
- [5] T.Williams, “Instrumentation Reference Book (Fourth Edition)”, 2010, [online]. Available: <https://www.sciencedirect.com/topics/engineering/radiated-emission>
- [6] Long Tran, “Data Fusion with 9 Degrees of Freedom Inertial Measurement Unit To Determine Object’s Orientation”, Senior Project, June 2017.
- [7] Slide môn Cơ sở điều khiển tự động và Lý thuyết điều khiển nâng cao của thầy Huỳnh Thái Hoàng.
- [8] Ebin Philip, Sharath Golluri, “Implementation of an Autonomous Self-Balancing Robot Using Cascaded PID Strategy”, 2020 6<sup>th</sup> International Conference on Control, Automation and Robotics.

## **PHẦN 5: PHỤ LỤC**

[1] Link thiết kế mạch PCB và source code của Self Balancing Robot:

<https://github.com/NVHMH/Self-Balancing-Robot.git>

[2] Link source code của bài tập State Machine:

[https://github.com/NVHMH/FW2\\_state\\_machine](https://github.com/NVHMH/FW2_state_machine)