**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP.HỒ CHÍ MINH**

**KHOA ĐIỆN – ĐIỆN TỬ**

**BỘ MÔN ĐIỀU KHIỂN VÀ TỰ ĐỘNG HÓA**



**BÁO CÁO ĐỒ ÁN 2**

**ĐIỀU KHIỂN TỰ ĐỘNG**

**ĐỀ TÀI: XÂY DỰNG PID CHO HỆ LÒ NHIỆT**

**GVHD: Nguyễn Minh Tâm**

**SVTH:**

**Nguyễn Trung Thảo 20151564**

**TP. HỒ CHÍ MINH – 5/2023**

# PHIẾU NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

Họ và tên Sinh viên: Nguyễn Trung Thảo MSSV: 20151564

Ngành: Công Nghệ Kỹ Thuật Điều Khiển Và Tự Động Hóa

Họ và tên Giáo Viên hướng dẫn: Nguyễn Minh Tâm

**NHẬN XÉT:**

1. Về nội dung đề tài & khối lượng thực hiện:

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

1. Ưu điểm của đề tài:

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………..………………..

1. Khuyết điểm:

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

1. Đề nghị cho bảo vệ hay không?

……………………………………………………………………………………………………………………………………………………………………

1. Đánh giá loại:

……………………………………………………………………………………………………………………………………………………………………

1. Điểm…………….(Bằng Chữ:……………………………………………….)

Tp. Hồ Chí Minh, ngày tháng năm 2022

Giáo viên hướng dẫn

(Ký & ghi rõ họ tên)

**LỜI CAM ĐOAN**

Em xin cam kết đề tài này là do em tự thực hiện dựa vào một số tài liệu trước đó và không sao chép từ tài liệu hay công trình nào đã có trước đó

**LỜI CẢM ƠN**

Nhóm sinh viên thực hiện đề tài xin gửi lời cảm ơn sâu sắc đến Thầy Nguyễn Minh Tâm, giảng viên thuộc khoa Điện - Điện tử đã trực tiếp hướng dẫn và tận tình giúp đỡ tạo điều kiện để em hoàn thành tốt đề tài.

Nhóm sinh viên thực hiện cũng gửi lời đồng cảm ơn đến thầy Nguyễn Văn Thái cùng các anh chị, các bạn trong 3DVisionLAB đã chia sẻ trao đổi kiến thức cũng như những kinh nghiệm quý báu trong thời gian thực hiện đề tài.

Cảm ơn đến gia đình đã tạo động lực và điều kiện để chúng con có thể hoàn thành được đồ án một cách tốt nhất.

Xin chân thành cảm ơn!

MỤC LỤC

# CHƯƠNG 1 TỔNG QUAN

## **Đặt vấn đề**

Nghiên cứu về việc xây dựng bộ điều khiển PID cho lò nhiệt là rất cấp thiết vì nó giúp cải thiện hiệu suất và tiết kiệm năng lượng của quá trình sản xuất. Hiện nay, các hệ thống lò nhiệt thường sử dụng bộ điều khiển PID để điều khiển quá trình nấu chín. Tuy nhiên, các bộ điều khiển này không phải lúc nào cũng hoạt động hiệu quả và đúng như mong muốn. Việc xây dựng bộ điều khiển PID cho lò nhiệt sẽ giúp tối ưu hóa quá trình sản xuất, tăng cường năng suất và giảm thiểu lượng khí thải và chất thải, từ đó giúp bảo vệ môi trường và phát triển bền vững. Nghiên cứu này cũng có tiềm năng ứng dụng rộng rãi trong các lĩnh vực sản xuất khác, giúp cải thiện hiệu suất và tiết kiệm năng lượng trong quá trình sản xuất. Vì vậy, nghiên cứu về việc xây dựng bộ điều khiển PID cho lò nhiệt là rất cấp thiết và có ảnh hưởng lớn đến cuộc sống và phát triển của con người.

Thế giới đang trong cuộc cách mạng thứ 4

## **Mục Tiêu**

Trong đề tài này, mục tiêu em đặt ra trong việc nghiên cứu và thiết kế giải pháp này là:

- Nghiên cứu ứng dụng bộ điều khiển PID cho lò nhiệt.

- Nghiên cứu phát triển giao diện điều khiển theo dõi từ xa.

- Thiết kế và thi công PCB

## **1.3. Nội dung nghiên cứu**

Trong đề tài này, em sẽ nghiên cứu về:

- Bộ điều khiển PID điều khiển nhiệt độ

- giao tiếp I2C và UART giữa vi xử lý và các thiết bị ngoại vi

- cái j đó liên quan đến điều khiển từ xa.

## **1.4. Giới hạn**

Phạm vi của đề tài như sau:

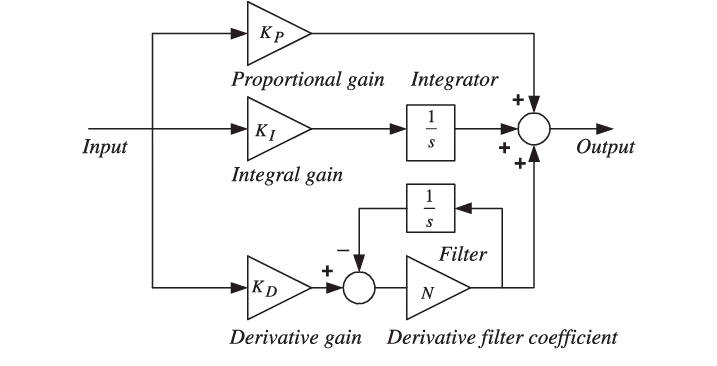
- Mô hình lò nhiệt với kích thước :…………

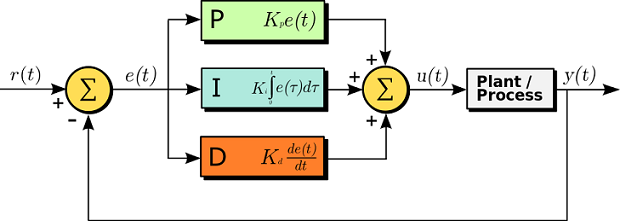
- Phạm vi theo dõi và điều khiển từ xa:

# CHƯƠNG 2 CƠ SỞ LÝ THUYẾT

## **2.1. Tổng quan về lý thuyết bộ điều khiển PID**

Bộ PID (Proportional-Integral-Derivative) là một bộ điều khiển phản hồi lặp được sử dụng trong các hệ thống điều khiển tự động. Bộ điều khiển liên tục theo dõi và điều chỉnh giá trị đầu ra để đạt được giá trị đầu vào mong muốn. Giá trị điều khiển của bộ PID được tính toán bằng cách sử dụng sai số(error) giữa giá trị đặt(setpoint) và giá trị thực tế. Để đạt được kết quả tốt nhất, các thông số của bộ PID cần được điều chỉnh theo tính chất của đối tượng mà bộ điều khiển PID đang điều khiển. Bộ PID có hiệu suất mạnh mẽ trong nhiều điều kiện hoạt động và khá đơn giản vì thế nó được sử dụng để điều khiển các quá trình trong các hệ thống công nghiệp, như quá trình sản xuất, quá trình điều khiển nhiệt độ, áp suất, độ ẩm, v.v.





*Hình 2.1. Sơ đồ bộ điều khiển bộ PID*

Thành phần của bộ PID:

*Thành phần tỷ lệ (Proportional):* Thành phần này phản hồi với sự khác biệt giữa giá trị đầu vào và giá trị đặt trước đó theo một tỷ lệ cố định được gọi là hệ số tỷ lệ (proportional gain). Thành phần tỷ lệ giúp bộ điều khiển phản hồi nhanh với sự thay đổi của giá trị đầu vào, nhưng nó không giải quyết được các vấn đề liên quan đến sai số ổn định và sai số dài hạn.

*Thành phần tích phân (Integral):* Thành phần này tích lũy các sai số dài hạn và giúp giảm độ sai lệch dài hạn. Thành phần tích phân tính toán trung bình trọng số của sai số trong một khoảng thời gian nhất định. Hệ số tích phân (integral gain) được sử dụng để điều chỉnh tốc độ tích phân. Nếu hệ số tích phân quá cao, bộ điều khiển có thể trở nên quá nhạy cảm và dễ dàng gây ra dao động.

*Thành phần vi phân (Derivative):* Thành phần này giúp giảm độ sai lệch ngắn hạn và giải quyết vấn đề dao động của thành phần tỷ lệ. Thành phần vi phân phản hồi với tốc độ thay đổi của giá trị đầu vào. Hệ số vi phân (derivative gain) được sử dụng để điều chỉnh tốc độ phản hồi của thành phần vi phân. Tuy nhiên, nếu hệ số vi phân quá cao, bộ điều khiển có thể trở nên quá nhạy cảm và dễ dàng gây ra nhiễu.

**2.2.1. Các phương pháp tìm thông số PID**

***Phương pháp thử và sai (Trial and Error Method):*** Đây là phương pháp đơn giản nhất và phổ biến nhất trong việc tìm kiếm thông số PID. Phương pháp này bao gồm việc tìm kiếm các giá trị của hệ số P, I và D bằng cách thực hiện các thử nghiệm và đánh giá kết quả. Phương pháp này có thể tốn thời gian, nhưng đối với các ứng dụng đơn giản, nó có thể là phương pháp hiệu quả.

***Phương pháp Ziegler-Nichols:*** Đây là phương pháp tìm thông số PID phổ biến nhất. Phương pháp này bao gồm việc sử dụng một loạt thử nghiệm để xác định các giá trị tối ưu cho hệ số P, I và D. Phương pháp Ziegler-Nichols được sử dụng rộng rãi trong các ứng dụng công nghiệp, nơi các thông số PID được tìm thấy nhanh chóng và chính xác. Các bước thực hiện:

*Bước 1: Tìm hệ số P*

Bắt đầu bằng cách đặt hệ số I và hệ số D bằng 0.

Tăng hệ số P cho đến khi hệ thống bắt đầu dao động với tần số f\_oscillation.

Ghi lại giá trị của hệ số P, k\_p\_critical.

*Bước 2: Tìm hệ số I và hệ số D*

Sử dụng giá trị k\_p\_critical để tính giá trị của hệ số I và hệ số D theo công thức:

k\_i = 0.45 k\_p\_critical / T\_oscillation

k\_d = 0.12 k\_p\_critical \* T\_oscillation  
Trong đó, T\_oscillation là chu kỳ dao động và được tính bằng 1/f\_oscillation.

*Bước 3: Kiểm tra và điều chỉnh*

Kiểm tra hệ thống với các giá trị hệ số PID đã tính toán.

Nếu hệ thống không đáp ứng yêu cầu, thử điều chỉnh các giá trị hệ số PID để tìm giá trị tốt nhất.

***Phương pháp Cohen-Coon:*** Phương pháp này cũng sử dụng các thử nghiệm để tìm kiếm các giá trị tối ưu cho hệ số P, I và D. Tuy nhiên, phương pháp này cho phép tìm kiếm các thông số PID với độ chính xác cao hơn so với phương pháp thử và sai. Các bước thực hiện như sau:

*Bước 1: Tìm hệ số K*

Đặt hệ số I và hệ số D bằng 0.

Tăng hệ số P cho đến khi hệ thống bắt đầu dao động với tần số f\_oscillation.

Ghi lại giá trị của hệ số P, k\_p\_critical.

*Bước 2: Tìm hệ số τ*

Thực hiện một thử nghiệm với một tín hiệu bước đầu vào và ghi lại thời gian t\_d cho đến khi hệ thống đạt đến 63,2% giá trị ổn định.

Tính giá trị của hệ số τ theo công thức: τ = 0,5t\_d.

*Bước 3: Tìm các hệ số P, I và D*

Sử dụng giá trị k\_p\_critical và τ để tính các giá trị của hệ số P, I và D theo công thức sau:

K\_p = 0.6 k\_p\_critical

T\_i = 0.5 τ

T\_d = 0.125 τ

*Bước 4: Kiểm tra và điều chỉnh*

Kiểm tra hệ thống với các giá trị hệ số PID đã tính toán.

Nếu hệ thống không đáp ứng yêu cầu, thử điều chỉnh các giá trị hệ số PID để tìm giá trị tốt nhất.

***Phương pháp đáp ứng tần số:*** Phương pháp này sử dụng phân tích tần số của hệ thống để xác định các thông số PID tối ưu. Phương pháp này thường được sử dụng trong các ứng dụng điều khiển hệ thống động cơ. Các bước thực hiện như sau:

*Bước 1: xác định hàm truyền đạt của hệ thống*

Để xác định hàm truyền đạt của hệ thống, ta có thể sử dụng phương pháp đo và tính toán dựa trên dữ liệu đầu vào và đầu ra của hệ thống.

*Bước 2: Xác định tần số cắt của hệ thống*

Tần số cắt là tần số tối đa mà hệ thống có thể truyền tín hiệu mà không bị giảm độ lớn. Tần số cắt có thể được xác định bằng cách tìm tần số tối đa mà hệ thống có thể truyền qua đường truyền đạt.

*Bước 3: Thực hiện bộ điều khiển PID*

Sử dụng các thông số Kp, Ki, Kd của PID để điều chỉnh đáp ứng của hệ thống với tín hiệu đầu vào tần số khác nhau. Các thông số này có thể được điều chỉnh bằng các phương pháp như tinh chỉnh bằng tay hoặc sử dụng các thuật toán tối ưu hóa.

*Bước 4: Kiểm tra đáp ứng tần số*

Kiểm tra đáp ứng của hệ thống đối với các tín hiệu đầu vào tần số khác nhau để đảm bảo rằng hệ thống đáp ứng đúng với yêu cầu của ứng dụng.

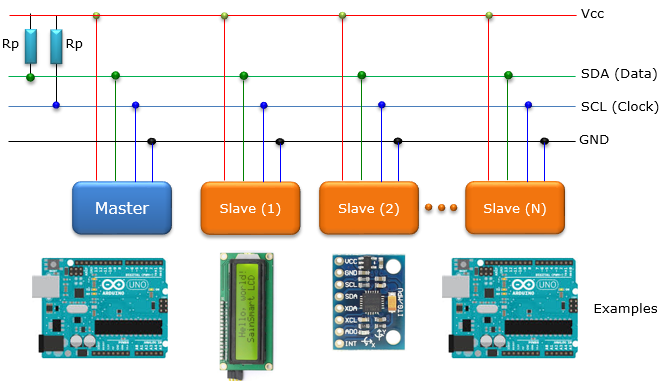
***Phương pháp tối ưu hóa:*** Phương pháp này sử dụng các thuật toán tối ưu hóa để tìm kiếm các giá trị tối ưu cho hệ số P, I và D. Phương pháp này thường được sử dụng trong các ứng dụng phức tạp và yêu cầu độ chính xác cao.

## **2.2. Tổng quan về các chuẩn giao tiếp giữa vi điều khiển và ngoại vi**

**2.2.1. Chuẩn giao tiếp I2C**

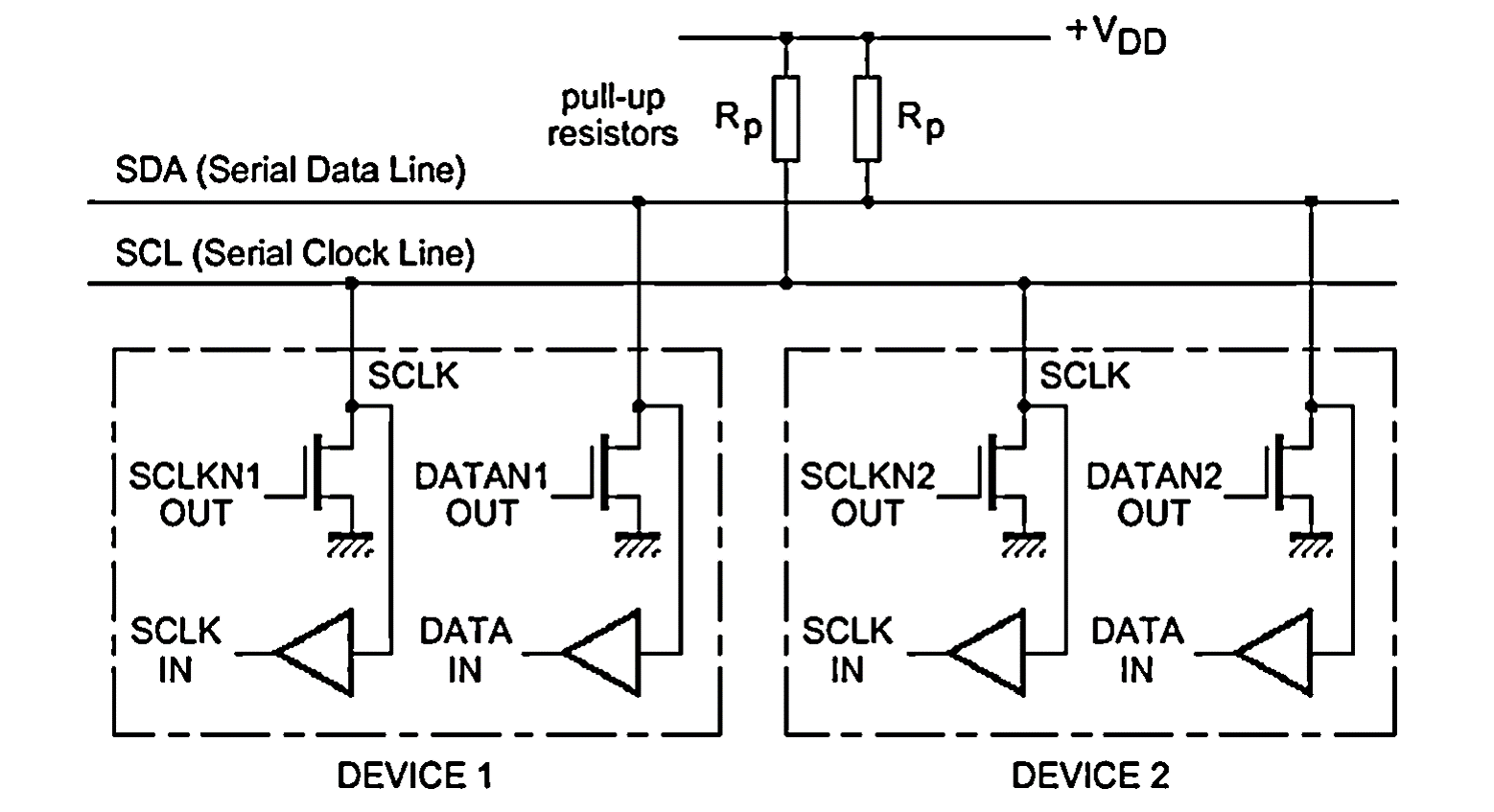
***2.2.1.1. Tổng quan chuẩn I2C***

I2C là một giao thức truyền thông đồng bộ được sử dụng để kết nối các vi mạch trong một hệ thống điện tử. "I2C" là viết tắt của "Inter-Integrated Circuit". Là chuẩn giao tiếp cấp bậc bao gồm một thiết bị chủ(master) và nhiều tớ(slave).



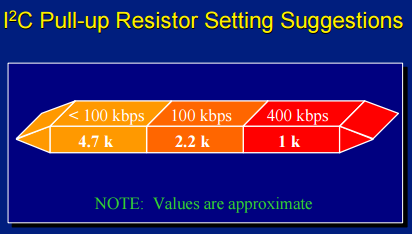
*Hình - Hệ thống các thiết bị giao tiếp theo chuẩn I2C*

Chuẩn giao tiếp I2C sử dụng hai dây để truyền thông giữa các thiết bị: dây SDA (Serial Data) và dây SCL (Serial Clock). Dây SDA được sử dụng để truyền dữ liệu giữa các thiết bị, còn dây SCL được sử dụng để đồng bộ hóa thời gian giữa các thiết bị trong hệ thống thiết bị chủ sẽ cấp xung và khống chế tần số trên đường dây này. Trong quá trình giao tiếp chỉ có một thiết bị chủ xuất xung trên đường SCL. Vì nhiều thiết bị tớ cùng dùng chung SDA và SCL nên chúng được phân biệt với nhau bằng địa chỉ(gồm 7 bit) nằm trong mỗi thiết bị tớ(địa chỉ này được nhà sản xuất quy định). Thiết bị chủ sẽ gửi địa chỉ của thiết bị tớ muốn giao tiếp trên đường SDA để chọn thiết bị cần giao tiếp. Trên lý thuyết thiết bị chủ có thể giao tiếp đến 128 thiết bị trên một cặp dây SDA và SCL.



Hình - Kết nối phần cứng của chuẩn I2C

Cặp dây I2C (SDA và SCL) cần được nối với điện áp dương thông qua điện trở kéo lên(pull-up resistor) nhằm xác định điện áp của mức logic 1. Điều này là cần thiết vì cấu tạo cặp chân I2C là cực máng để hở. Giá trị trở kéo dao động từ 1KOhm đến 10KOhm phục thuộc vào tốc độ trao đổi thông tin giữa các thiết bị.

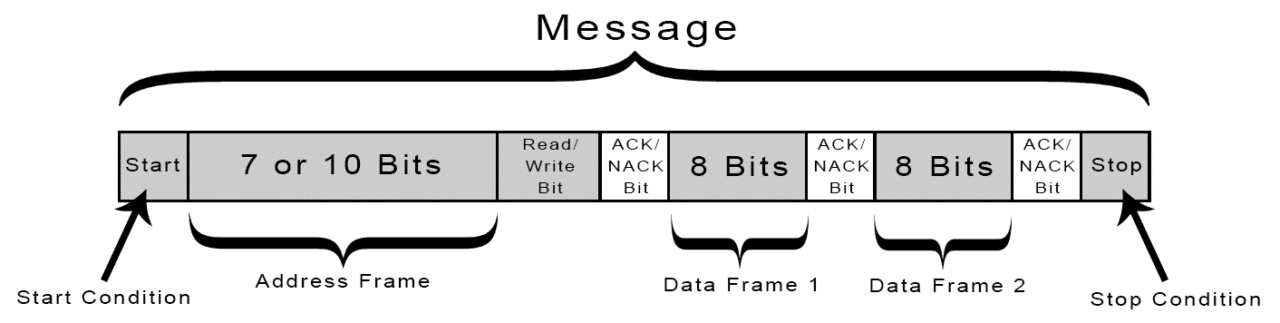


*Hình – Lựa chọn giá trị điện trở kéo lên theo tốc độ giao tiếp*

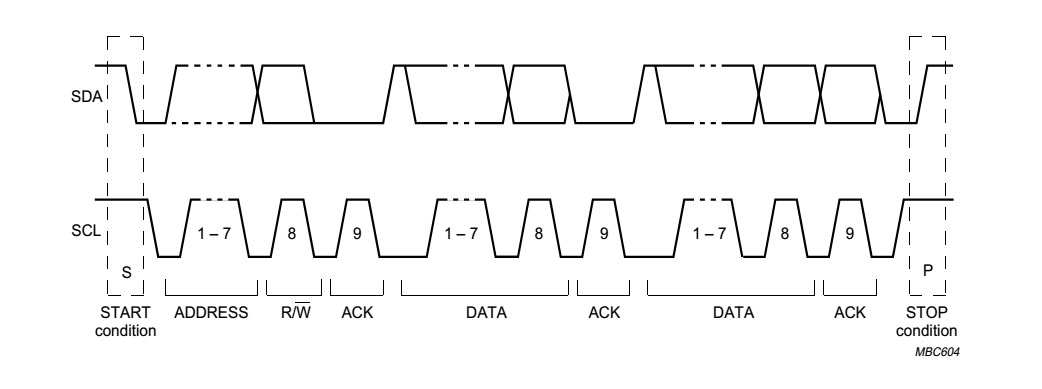
***2.2.1.2. Quy trình truyền nhận dữ liệu***

Cấu trúc khung truyền nhân dữ liệu của chuẩn I2C bao gồm:

* Tín hiệu bắt đầu bắt đầu giao tiếp
* 7 bit địa chỉ thiết bị tớ và 1 bit 0 hoặc 1 ở LSB quyết định đọc hay ghi dữ liệu.
* Tín hiệu phản hồi từ thiết bị tớ
* Dữ liệu truyền mỗi lần 8 bit và giữa mỗi lần truyền xong 8 bit là 1 bit xác nhận từ thiết bị tớ
* Tín hiệu kết thúc giao tiếp



*Hình – cấu trúc khung truyền dữ liệu I2C*



*Hình – dạng sóng truyền nhận dữ liệu chuẩn I2C*

*Bước 1:* Thiết bị chủ sẽ gửi điều kiện bắt đầu giao tiếp (kéo chân SDA xuống logic 0 sau đó đến SCL)

*Bước 2:* Thiết bị chủ sẽ gửi địa chỉ gồm 7 bit kèm theo 1 bit yêu cầu đọc hoặc ghi (tương ứng 1 hoặc 0). Các thiết bị tớ sẽ so sánh địa chỉ thiết bị chủ gửi. Nếu đúng thì kéo chân SDA xuống logic 0 để xác nhận(ACK). Nếu không thiết bị tớ chờ lệnh bắt đầu ở lần tiếp theo.

*Bước 3:* Tùy vào cách mà thiết bị tớ yêu cầu thiết bị chủ sẽ gửi lệnh hoặc địa chỉ ở 8 bit tiếp theo đồng thời chờ bit ACK để biết thiết bị tớ đã nhận thành công. Điều này nằm trong datasheet của thiết bị tớ.

*Bước 4:* Thiết bị chủ gửi dữ liệu hoặc nhận dữ liệu từ tớ cho đến khi hoàn tất.

*Bước 5:* thiết bị chủ gửi tín hiệu kết thúc.

**2.2.2. Chuẩn giao tiếp UART**

**2.2.2.1. Tổng quan chuẩn UART**

UART là viết tắt của "Universal Asynchronous Receiver/Transmitter" (Bộ nhận/gửi không đồng bộ đa năng), là một loại giao diện truyền thông số được sử dụng để kết nối các thiết bị điện tử với nhau. UART cho phép truyền dữ liệu giữa các thiết bị thông qua hai dây truyền thông đơn giản: một dây truyền dữ liệu (TX) và một dây nhận dữ liệu (RX). UART là chuẩn giao tiếp ngang hàng không phân biệt chủ tớ.

UART hoạt động theo cơ chế không đồng bộ, có nghĩa là dữ liệu được truyền đi mà không cần tín hiệu đồng bộ hóa. Thay vào đó, UART sử dụng một tần số cơ bản để đồng bộ hóa các bit dữ liệu, với tốc độ truyền được đặt trước(Baud rate) và được đảm bảo giữa các thiết bị được kết nối. Nếu tốc độ này khác nhau dữ liệu mà thiết bị nhận sẽ không đúng. Có nhiều tốc độ truyền nhưng phổ biến là 9600 và 115200 bps(bit per second).

Diagram

Description automatically generated

*Hình – sơ đồ nối dây giao tiếp UART*

**2.2.2.2. Quy trình truyền nhận dữ liệu**

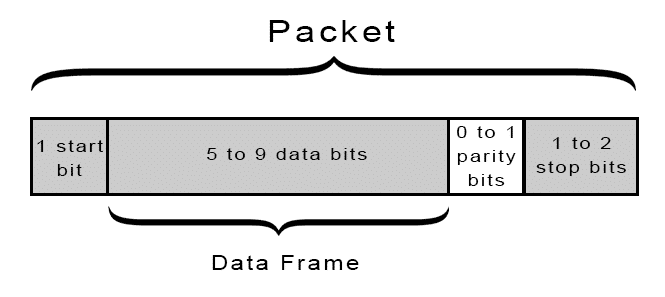
Cấu trúc khung truyền nhân dữ liệu của chuẩn UART bao gồm:

1 bit bắt đầu giao tiếp(logic 0).

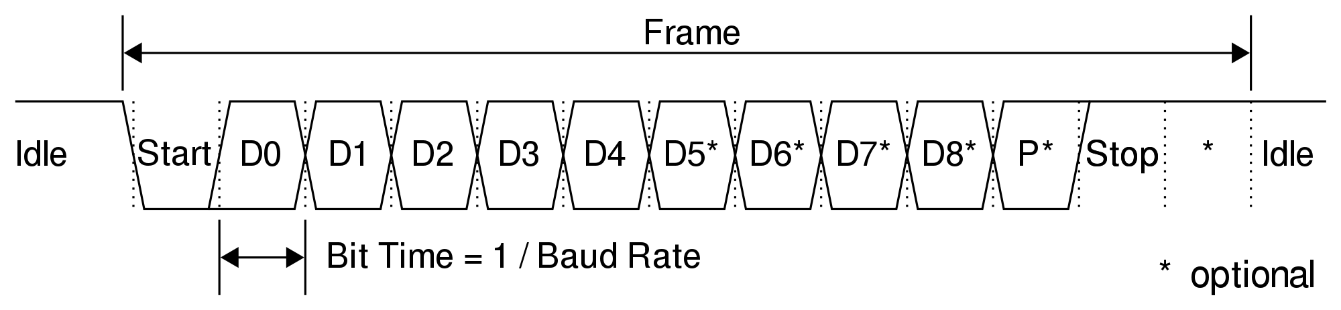
Từ 5 đến 9 bit data tùy theo cấu hình

1 bit kiểm tra lỗi dữ liệu (nếu số lượng số 1 trong dữ liệu là chặn thì bit này bằng 0 và ngược lại)

1 hoặc 2 bit dừng giao tiếp.



*Hình – cấu trúc khung truyền dữ liệu UART*



*Hình – dạng sóng truyền nhận dữ liệu chuẩn UART*

*Bước 1:* Vi điều khiển kéo chân Tx xuống 0 1 khoảng thời gian bằng 1 / Baud Rate

*Bước 2:* Vi điều khiển gửi đủ các bit dữ liệu thường là 8bit và bit parity nếu có

*Bước 3:* Vi điều khiển gửi tín hiệu kết thúc

# CHƯƠNG 3 TÍNH TOÁN VÀ THIẾT KẾ

## **3.1 Yêu cầu của hệ thống**

Xuất phát từ yêu cầu của đề tài và các vấn đế trong thực tế, ta cần phải có những yêu cầu trong thiết kế mô hình như sau:

- Hệ thống áp dụng bộ điều khiển PID để điều khiển nhiệt độ.

- Giám sát và cài đặt nhiệt độ bên trong lo nhiệt thông qua Mobile-app.

## **3.2. Thiết kế phần cứng**

### 3.2.1. Lựa chọn thiết bị

### 3.2.1.1. Module Wifi ESP8266

|  |  |
| --- | --- |
| Chuẩn không giây | 802.11 b/g/n |
| Dải tần số | 2.4GHz-2.5GHz |
| Giao tiếp | UART / HSPI / I2C / I2S / PWM |
| Điện áp làm việc | 3.0 ~ 3.6V |
| Dòng làm việc | 80mA (trung bình) |
| Analog (ADC) | 12 bit – 16 cổng |
| DACs (digital to analog) | 8 bit – 2 cổng |

*Lý do chọn:* có chứ năng kết nối wifi, rẻ hơn ESP32 và dễ dàng lập trình bằng arduinoIDE.



3.2.1.2. Vi Điều khiển STM32F103C8T6

*Lý do chọn:* hoạt động ổn định và rẻ hơn các vi điều khiển khác, hỗ trợ debug lỗi bằng ST-linkV2.

|  |  |
| --- | --- |
| Giao tiếp | UART / HSPI / I2C / I2S / PWM /DMA/CAN |
| Điện áp làm việc | 2.0 ~ 3.6V |
| Dòng làm việc | 94mA (trung bình) |
| ADC | 12 bit - 10 kênh |
| DAC | Không có |



3.2.1.3. Cảm biến nhiệt độ SHT31

*Lý do chọn:* Quy đổi nhiệt độ và độ ẩm thành dữ liệu số 16 bit giao tiếp I2C, không cần mạch khếch đại như PT100.

|  |  |
| --- | --- |
| Thang độ ẩm | 0 ~ 100% RH |
| Ngõ ra | I2C |
| Độ phân giải | 16bit |
| Sai số | ±2% RH |
| Điện áp cung cấp | 2.15 ~ 5.5V |
| Nhiệt độ hoạt động | -40°C ~ 125°C |



3.2.1.4. led 7 đoạn 3 led và 4 led cathode chung:

*Lý do chọn:* nhỏ gọn và có sẵn

|  |  |
| --- | --- |
| Điện áp làm việc | 1.8V |
| Điện áp ngược | 5V |
| Dòng liên tục mỗi thanh led | 30mA |
| Dòng đỉnh mỗi thanh led | 200mA |



3.2.1.5. Encoder rotary

|  |  |
| --- | --- |
| Số vị trí | 20 |



3.2.1.6. AMS1117 3v3

Có thể chọn bất kì linh kiện nào hạ áp tuyến tính từ 5v xuống 3v.

|  |  |
| --- | --- |
| Điện áp vào | MAX 18V |
| Điện áp đầu ra | 3.3V |
| Dòng ra | 1A |



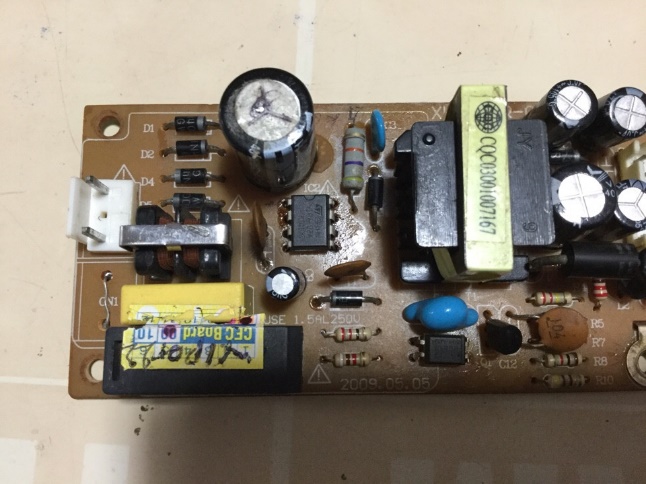
3.2.1.7. IC buck XL1509 – 5v

Lý do chọn: rẻ, sơ đồ mạch đơn giản

|  |  |
| --- | --- |
| Điện áp vào | MAX 45V |
| Điện áp đầu ra | 5V |
| Dòng ra | 2A |



3.2.1.8. Mạch nguồn xung 220 => 5V (Board mạch tháo máy)

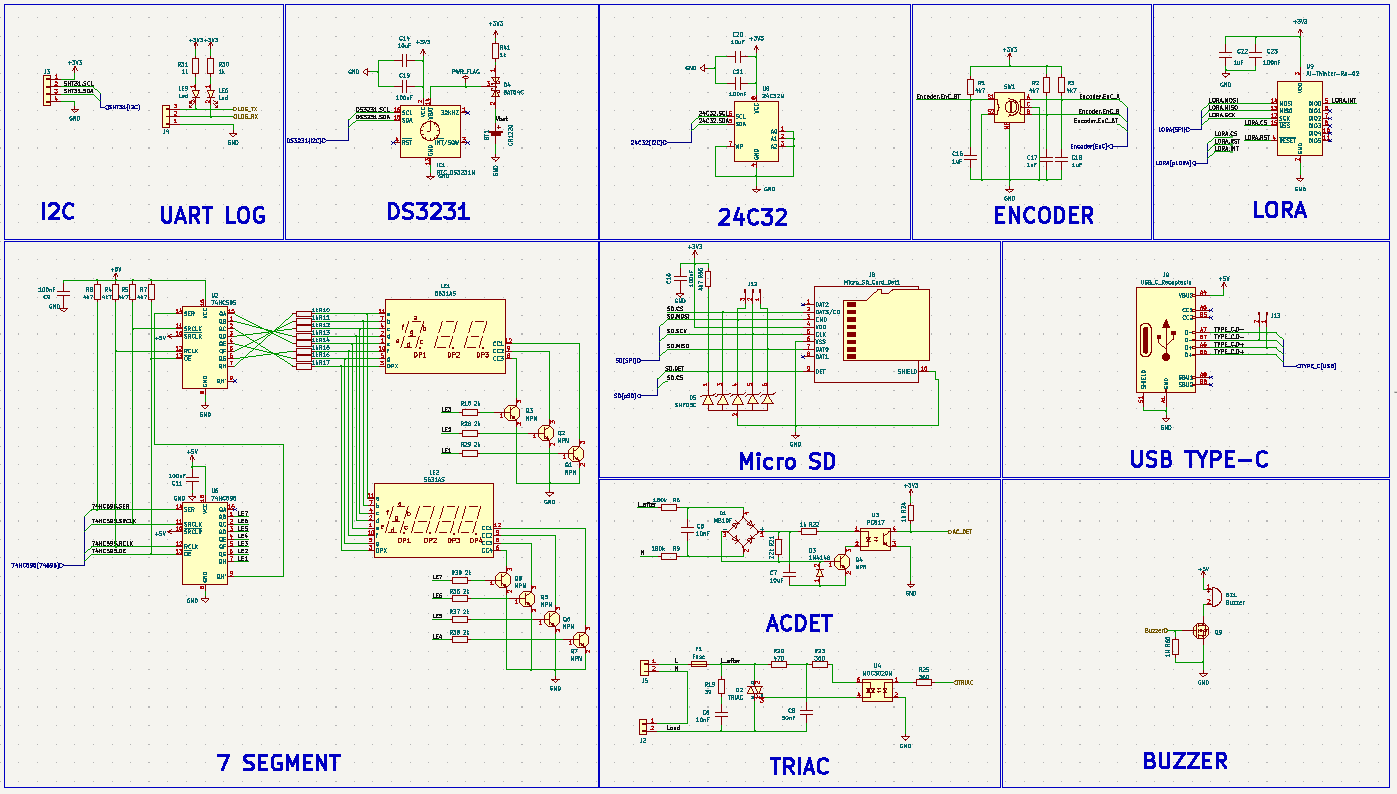
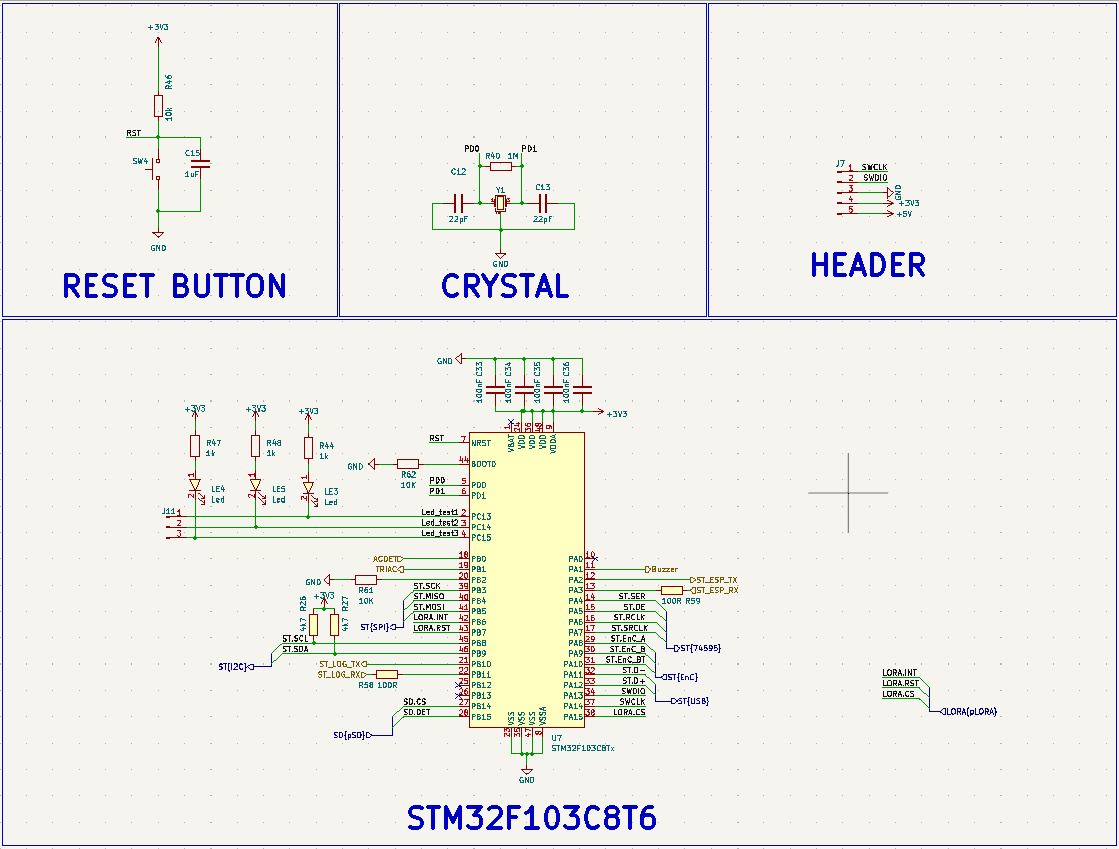
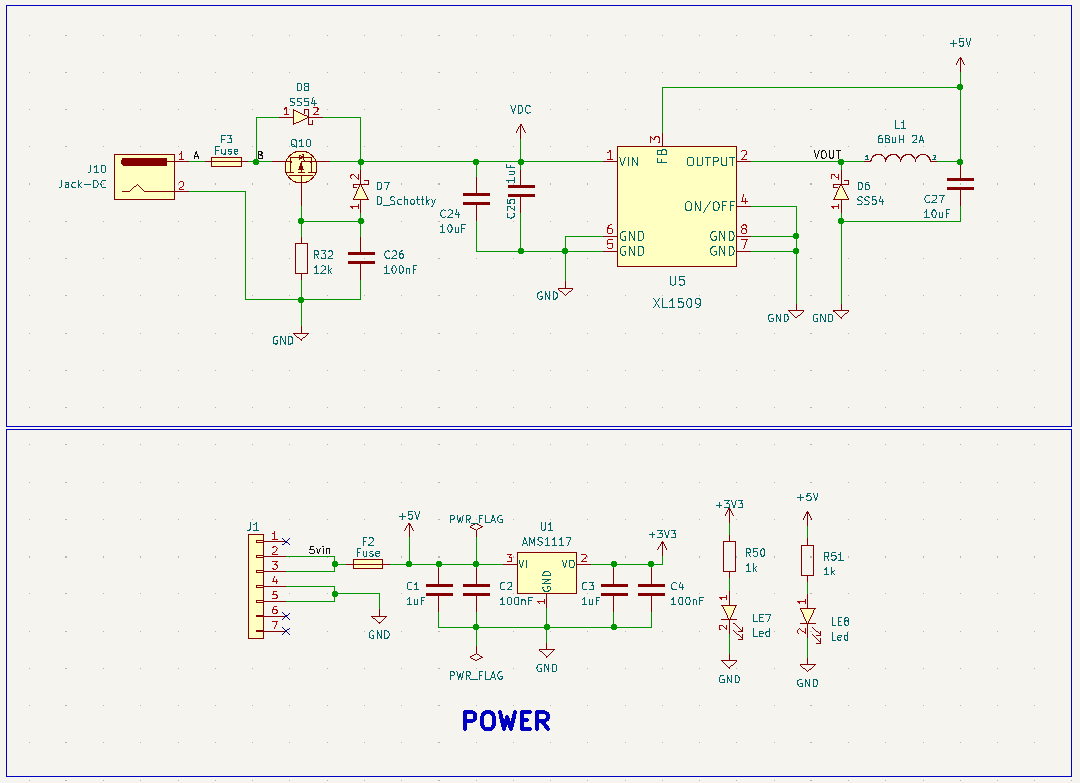
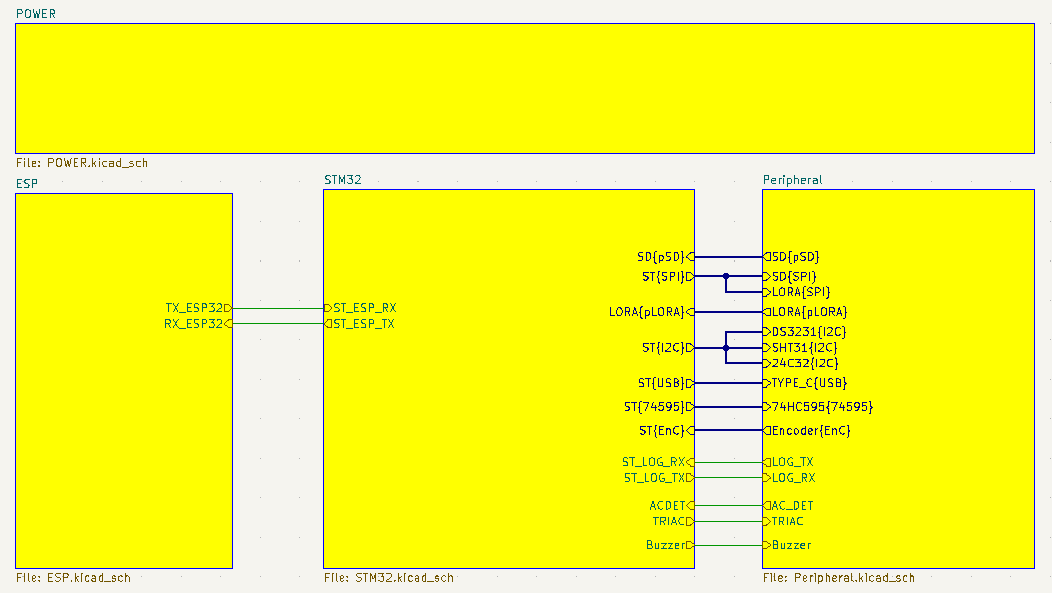


3.2.1.9. Đèn sợi đốt E27

|  |  |
| --- | --- |
| Điện áp | 220 VAC |
| Công suất | 60W |



### 3.2.2. Sơ đồ nối dây

Diagram, schematic

Description automatically generated

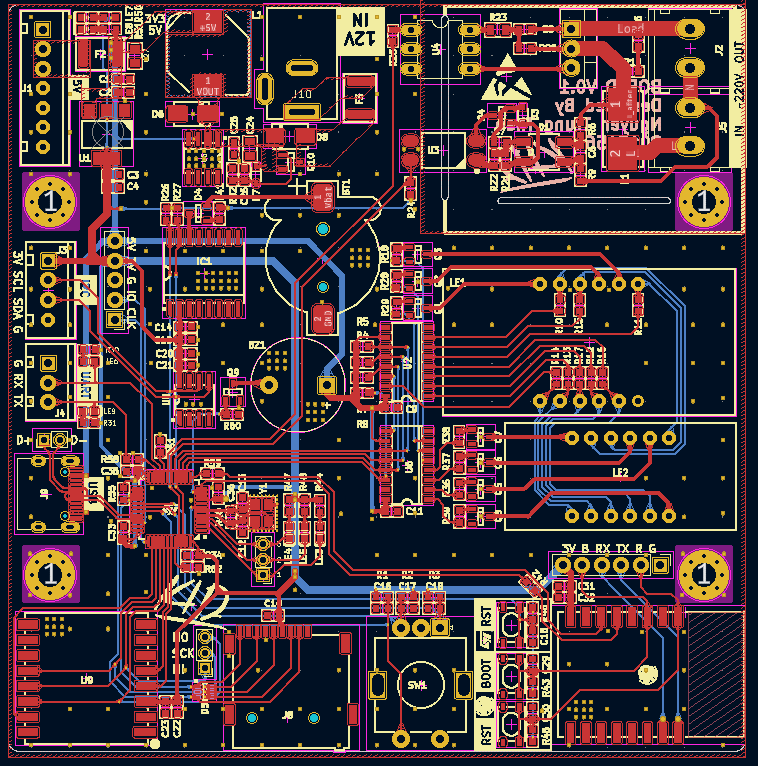
## **3.3. Thiết kế phần mềm**

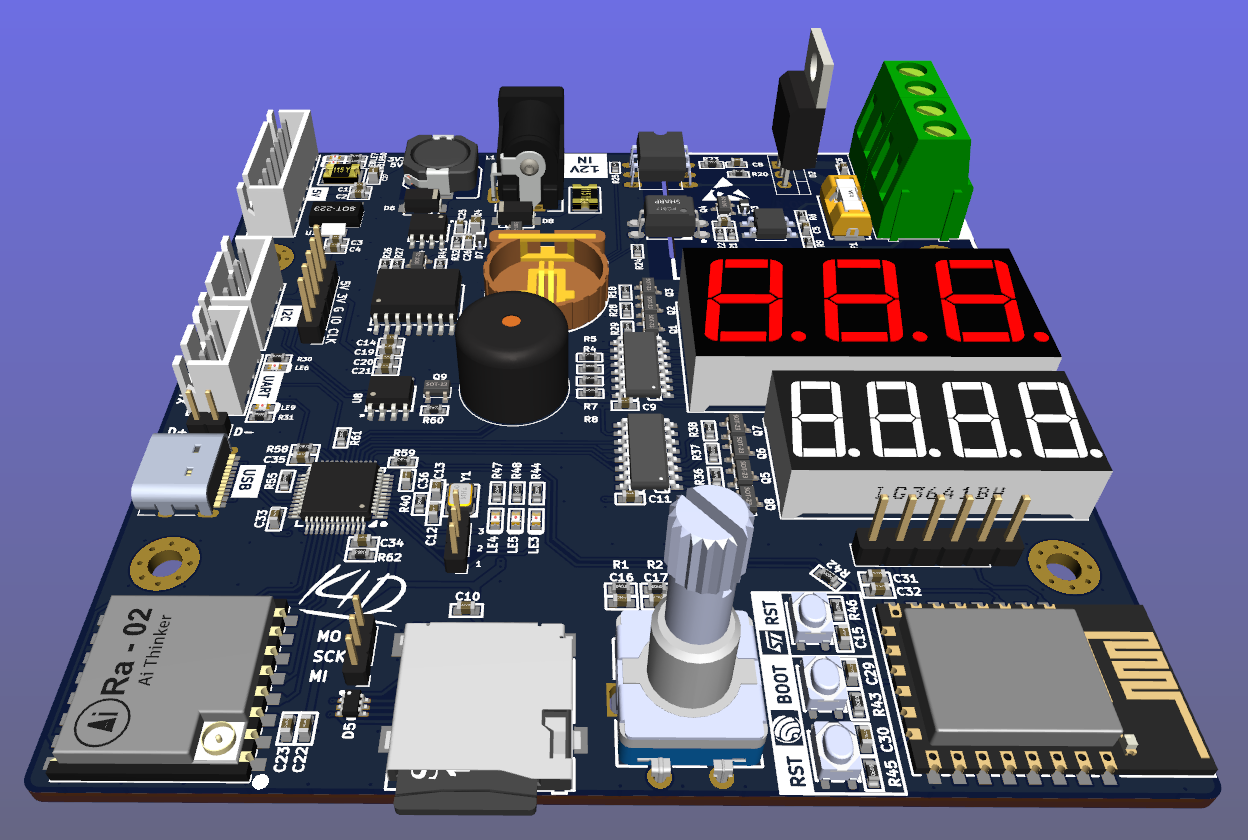
### 3.3.1 Lưu đồ giải thuật chương trình

### 3.3.1 Lưu đồ giải thuật Mobie App

# CHƯƠNG 4 KẾT QUẢ

## **4.1. Kết quả phần cứng**





## **4.2. Kết quả phần mềm**

# CHƯƠNG 5 KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## **5.1. Kết Luận**

## **5.2. Hướng Phát triển**

# TÀI LIỆU THAM KHẢO

<https://kienltb.wordpress.com/2015/04/03/chon-tro-pullup-i2c-i2c-bus-pullup-resistor-calculation/>