

## MỤC LỤC

DANH MỤC HÌNH VẼ .....	3
DANH MỤC BẢNG BIỂU.....	4
LỜI MỞ ĐẦU .....	5
CHƯƠNG 1: TỔNG QUAN ĐỀ TÀI.....	6
1.1.    Đặt vấn đề.....	6
1.2.    Ý tưởng và mục tiêu .....	7
1.3.    Các mô hình tương tự.....	7
CHƯƠNG 2: PHÂN TÍCH HỆ THỐNG.....	10
2.1.    Yêu cầu chức năng .....	10
2.2.    Yêu cầu phi chức năng .....	10
2.3.    Sơ đồ khối hệ thống.....	11
2.3.1. <i>Khối cảm biến</i> .....	11
2.3.2. <i>Khối xử lý</i> .....	12
2.3.3. <i>Khối thực thi</i> .....	14
2.3.4. <i>Khối hiển thị</i> .....	15
2.3.5. <i>Luồng hoạt động của hệ thống</i> .....	16
CHƯƠNG 3: THIẾT KẾ HỆ THỐNG .....	18
3.1.    Thuật toán PID .....	18
3.2.    Chế độ Deep Sleep .....	21
3.3.    Giao thức sử dụng .....	22
3.3.1. <i>Giao tiếp I2C</i> .....	22
3.3.2. <i>Giao tiếp UART</i> .....	24
3.4.    Thiết kế phần cứng .....	25
3.4.1. <i>Sơ đồ mạch</i> .....	25

3.4.2. Thiết kế mô hình.....	26
3.4.3. Tính toán công suất.....	27
CHƯƠNG 4: KIỂM THỬ VÀ ĐÁNH GIÁ .....	30
4.1. Kiểm thử.....	30
4.1.1. Điều chỉnh PID của chế độ làm lạnh.....	30
4.1.2. Điều chỉnh PID của chế độ làm nóng.....	31
4.1.3. Độ ổn định khi sử dụng bộ PID trong quá trình làm lạnh.....	32
4.1.4. Độ ổn định khi sử dụng bộ PID trong quá trình làm nóng.....	32
4.1.5. Độ ổn định khi thay đổi chế độ luân phiên .....	32
4.2. Đánh giá.....	33
CHƯƠNG V: KẾT LUẬN.....	35
PHỤ LỤC .....	36
PHÂN CÔNG CÔNG VIỆC.....	<b>Lỗi! Thẻ đánh dấu không được xác định.</b>
TÀI LIỆU THAM KHẢO.....	36

**DANH MỤC HÌNH VẼ**

Hình 2.1 Sơ đồ khối hệ thống.....	11
Hình 2.2 Cảm biến BME280 .....	12
Hình 2.3 Vi xử lý ESP32.....	13
Hình 2.4 TEC1-12706 .....	14
Hình 2.5 Module L298N .....	15
Hình 2.6 LCD1602 & Module I2C .....	16
Hình 3.1 Cách hoạt động của thuật toán PID .....	18
Hình 3.2 Sơ đồ thuật toán hệ thống áp dụng PID.....	20
Hình 3.3 Sơ đồ các khối trong chế độ Deep Sleep.....	22
Hình 3.4 Nguyên tắc I2C.....	23
Hình 3.5 Khung dữ liệu trong I2C .....	23
Hình 3.6 Phương thức truyền UART .....	24
Hình 3.7 Gói dữ liệu trong UART .....	24
Hình 3.8 Sơ đồ mạch .....	25
Mô hình thiết kế sản phẩm .....	26

**DANH MỤC BẢNG BIỂU**

Bảng 2.1. Một vài thông số cơ bản của ESP32 .....	13
Bảng 3.1. Bảng năng lượng tiêu thụ của ESP32 .....	27
Bảng 3.2. Năng lượng tiêu thụ của BME280 .....	28
Bảng 3.3. Năng lượng tiêu thụ của TEC1-12706.....	28
Bảng 3.4. Năng lượng tiêu thụ của LCD1602.....	28
Bảng 3.5. Năng lượng tiêu thụ của L298 .....	29
Bảng 4.1 $K_i, K_d = 0$ .....	30
Bảng 4.2 $K_p = 40, K_i = 0$ .....	30
Bảng 4.3 $K_p = 40, K_i = 5$ .....	31
Bảng 4.4 Test case làm nóng.....	31
Bảng 4.5 Độ ổn định trong chế độ làm lạnh.....	32
Bảng 4.6 Độ ổn định trong chế độ làm nóng .....	32
Bảng 4.7 Test thay đổi chế độ luân phiên .....	32

## LỜI MỞ ĐẦU

*Trong bối cảnh công nghệ phát triển, các thiết bị điện tử đóng vai trò thiết yếu trong nhiều ngành công nghiệp và nghiên cứu khoa học. Tuy nhiên, các thiết bị này rất nhạy cảm với nhiệt độ và độ ẩm, hai yếu tố ảnh hưởng đáng kể đến hiệu suất và tuổi thọ của chúng. Đề tài "Tủ kiểm soát nhiệt độ, độ ẩm cho thiết bị điện tử" ra đời nhằm giải quyết vấn đề bảo vệ thiết bị điện tử khỏi các tác động tiêu cực từ môi trường.*

*Dự án sử dụng bo mạch ESP32 với khả năng kết nối không dây và điều khiển linh hoạt, cùng với cảm biến nhiệt độ và độ ẩm BME280 để giám sát điều kiện bên trong tủ. Khi phát hiện nhiệt độ hoặc độ ẩm vượt quá ngưỡng cài đặt, hệ thống sẽ tự động điều chỉnh bằng cách kích hoạt các thiết bị phụ trợ như quạt làm mát hoặc máy hút ẩm để duy trì điều kiện ổn định.*

*Hệ thống không chỉ có khả năng ứng dụng cao trong môi trường yêu cầu điều kiện bảo quản nghiêm ngặt mà còn mang lại tiềm năng phát triển thành giải pháp IoT cho các thiết bị thông minh. Đây là một giải pháp hứa hẹn nhằm bảo vệ thiết bị điện tử, góp phần gia tăng tuổi thọ và hiệu quả hoạt động của chúng.*

*Nhờ vào sự linh hoạt và khả năng mở rộng của ESP32 cùng với công nghệ IoT, dự án hứa hẹn sẽ mang lại nhiều lợi ích cho việc quản lý và bảo vệ các thiết bị điện tử trong những điều kiện môi trường khác nhau.*

## CHƯƠNG 1: TỔNG QUAN ĐỀ TÀI

### 1.1. Đặt vấn đề

Trong thời đại công nghệ số hiện nay, lĩnh vực sản xuất linh kiện và vi mạch điện tử đang phát triển mạnh mẽ. Khi mà các công ty công nghệ và vi mạch như NVIDIA, Intel, và các nhà sản xuất chip khác ngày càng chú trọng vào việc phát triển và nâng cao hiệu suất các dòng chip tiên tiến, nhu cầu về quy trình bảo quản linh kiện cũng trở nên quan trọng hơn bao giờ hết. Do thiết bị điện tử, đặc biệt là các linh kiện và vi mạch, rất nhạy cảm với điều kiện môi trường, nhất là nhiệt độ và độ ẩm. Khi nhiệt độ quá cao hoặc độ ẩm không ổn định, các linh kiện dễ bị oxy hóa, hư hỏng, dẫn đến suy giảm hiệu suất và tuổi thọ:

- Chip vi xử lý và vi điều khiển (CPU, GPU, MCU): các vi xử lý hiện đại chứa hàng tỷ transistor siêu nhỏ. Khi nhiệt độ tăng quá cao, transistor có thể bị hỏng hoặc giảm hiệu suất, gây ra hiện tượng quá nhiệt và làm hỏng vi mạch.
- Tụ điện (gốm, và nhôm): tụ điện dễ bị ảnh hưởng bởi nhiệt độ và độ ẩm. Tụ tantalum, thường được sử dụng trong mạch nhạy cảm, có thể gặp sự cố hoặc giảm tuổi thọ khi nhiệt độ thay đổi đột ngột hoặc độ ẩm quá cao.
- Mạch in: PCB dễ bị cong vênh hoặc giãn nở khi tiếp xúc với nhiệt độ và độ ẩm không ổn định, dẫn đến gãy mạch hoặc kết nối kém, ảnh hưởng nghiêm trọng đến hiệu suất của mạch.
- Linh kiện quang học (đi-ốt quang, cảm biến quang): các linh kiện này nhạy cảm với môi trường ẩm, vì độ ẩm cao có thể làm mờ các bề mặt quang học và gây nhiễu tín hiệu.
- Pin và ắc quy: pin lithium-ion và các loại ắc quy dễ bị quá nhiệt, có thể gây cháy nổ. Độ ẩm cao cũng có thể gây gỉ sét, ảnh hưởng đến các chân tiếp xúc và làm giảm tuổi thọ. và cũng dựa trên việc học máy đòi hỏi rất nhiều kiến thức về toán học và việc học sâu.

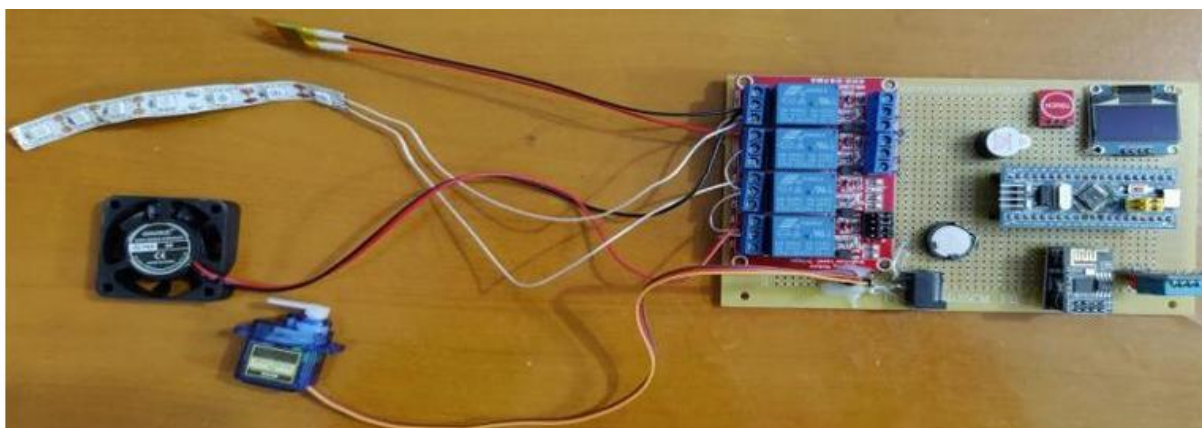
## 1.2. Ý tưởng và mục tiêu

Nhận thức được tầm quan trọng và tính ứng dụng của việc bảo quản các thiết bị điện tử và với mong muốn hiểu rõ hơn về môn học kỹ thuật vi xử lý, đồng thời nâng cao khả năng áp dụng, thực hành các kiến thức đã học như sử dụng và lập trình vi điều khiển, nhóm đã quyết định chế tạo một thiết bị bảo quản các linh kiện, tuy nhiên ở thời điểm hiện tại, định hướng sản phẩm đang có hướng theo mục đích là học tập hơn là hướng ứng dụng ra thị trường.

Sản phẩm giúp người dùng nắm được các thông số cơ bản nhất trong môi trường tại nơi đặt linh kiện. Bên cạnh đó, với mục tiêu phát triển và nâng cấp sản phẩm trong tương lai, là tiền đề để tạo nên một sản phẩm có tính ứng dụng, mạch sẽ được thiết kế để có thể đo được nhiều thông số khác của chất lượng không khí như nồng độ bụi mịn, nồng độ khí ga, nồng độ khí CO<sub>2</sub>, H<sub>2</sub>, NH<sub>3</sub>... tính toán, điều khiển các thiết bị tản nhiệt một cách chính xác mà vẫn đảm bảo tích hợp việc duy trì và tiết kiệm năng lượng.

## 1.3. Các mô hình tương tự

[1] [Zheng, Jiahao, et al. "Design of an Intelligent Storage Cabinet System based on STM32." 2023 3rd International Conference on Education, Information Management and Service Science \(EIMSS 2023\). Atlantis Press, 2023.](#)



- Nội dung:

Hệ thống tủ lưu trữ thông minh dựa trên vi điều khiển STM32. Hệ thống bao gồm các thành phần như vi điều khiển, màn hình hiển thị điện tử, nút cảm ứng, cảm biến nhiệt

độ và độ ẩm, cùng với mô-đun WIFI. Mục tiêu của hệ thống giám sát và kiểm soát điều kiện môi trường trong tủ lưu trữ theo thời gian thực.

- Chức năng:
  - Giám sát môi trường: Hệ thống sử dụng cảm biến DHT11 để theo dõi nhiệt độ và độ ẩm trong tủ. Nếu phát hiện điều kiện bất thường, hệ thống sẽ cảnh báo bằng còi.
  - Điều chỉnh tự động: Khi độ ẩm vượt quá ngưỡng nhất định, hệ thống sẽ tự động kích hoạt quạt và thiết bị sưởi để giảm độ ẩm.
  - Khử trùng: Hệ thống cung cấp chức năng khử trùng bằng đèn UV, có thể được lập lịch hoặc điều khiển thủ công.
  - Điều khiển từ xa: Người dùng có thể theo dõi và điều khiển hệ thống qua ứng dụng di động kết nối qua WiFi.
- Kết quả:

Hệ thống hoạt động hiệu quả với khả năng tự động điều chỉnh các thành phần như quạt và đèn UV dựa trên dữ liệu cảm biến. Khi độ ẩm vượt quá 60%, hệ thống sẽ tự động kích hoạt quạt; nếu vượt quá 80%, còi sẽ kêu báo động.

[2] [Budijono, Santoso. "Smart temperature monitoring system using ESP32 and DS18B20." IOP Conference Series: Earth and Environmental Science. Vol. 794. No. 1. IOP Publishing, 2021.](#)





Trình bày hệ thống giám sát nhiệt độ thông minh trong tủ đông.

- Phương pháp:
  - Cảm biến nhiệt độ DS18B20 được sử dụng để đo nhiệt độ từ  $-55^{\circ}\text{C}$  đến  $+125^{\circ}\text{C}$ .
  - Vi điều khiển ESP32 xử lý dữ liệu và kết nối Wi-Fi để gửi thông tin đến cơ sở dữ liệu đám mây.
  - Hệ thống phát cảnh báo âm thanh khi nhiệt độ vượt quá ngưỡng đã định.
- Kết quả:
  - Giám sát liên tục và ghi lại nhiệt độ trong 24 giờ.
  - Gửi dữ liệu đến cơ sở dữ liệu và hiển thị trên màn hình OLED.
  - Cảnh báo qua SMS hoặc âm thanh khi có sự cố.

[3] [Olajire, Oluwatobi Toheeb. An analysis of pressure, temperature and humidity in a room using a Raspberry pi/Arduino uno/Arduino uno and BME280. Diss. Dublin, National College of Ireland, 2020.](#)

Sử dụng Raspberry Pi, Arduino Uno và BME280 để đo và xử lý nhiệt độ và độ ẩm.

- Nội dung:
    - Thu thập dữ liệu: Sử dụng Raspberry Pi/Arduino Uno với cảm biến BME280 để ghi lại sự biến đổi của nhiệt độ, độ ẩm và áp suất.
    - Phân tích: Phân tích dữ liệu thu thập được để xác định điều kiện tối ưu nhằm ngăn chặn độ ẩm.
    - Trực quan hóa: Hiển thị dữ liệu đã phân tích trên một trang web để người dùng tương tác.
- ⇒ Các dự án đều sử dụng vi điều khiển khác nhau (STM32, ESP32, Raspberry Pi/Arduino Uno), với các cảm biến khác nhau phù hợp với mục tiêu cụ thể của từng dự án.
- ⇒ Dự án STM32 có tính năng bảo quản đặc biệt với các chế độ khử trùng và sấy tự động, thích hợp cho các không gian lưu trữ.
- ⇒ Dự án ESP32 và DS18B20 thiên về ứng dụng trong các hệ thống giám sát đơn giản, tiết kiệm chi phí. Trong khi đó, dự án sử dụng Raspberry Pi/Arduino Uno và BME280 là lựa chọn tốt cho các hệ thống cần kiểm soát môi trường chặt chẽ và yêu cầu dữ liệu chính xác về cả nhiệt độ, độ ẩm, và áp suất.

## CHƯƠNG 2: PHÂN TÍCH HỆ THỐNG

### 2.1. Yêu cầu chức năng

#### Đo lường nhiệt độ

- Sử dụng cảm biến BME280 để đo nhiệt độ bên trong tủ bảo quản.
- Đảm bảo đọc giá trị nhiệt độ chính xác và cập nhật liên tục.

#### Điều khiển nhiệt độ

Dựa trên giá trị nhiệt độ đo được, hệ thống sẽ điều khiển module làm lạnh/nóng tủ. Nếu nhiệt độ chênh lệch khỏi ngưỡng cài đặt, hệ thống sẽ kích hoạt module tăng/giảm nhiệt độ của tủ để duy trì nhiệt độ ở mức ngưỡng đảm bảo cho thiết bị điện tử.

#### Hiện thị trên màn hình

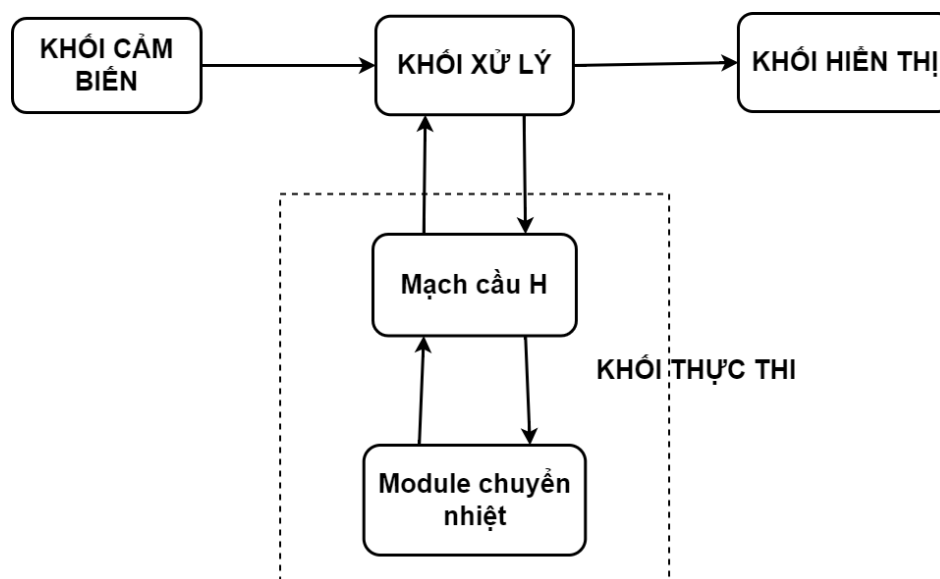
Màn hình hiển thị:

- Nhiệt độ hiện tại (đơn vị °C).
- Trạng thái hệ thống: "Heating", "Cooling", "Maintaining", hoặc "Error".
- Giá trị PWM của module nóng lạnh.

### 2.2. Yêu cầu phi chức năng

- Độ chính xác: Cung cấp kết quả đo chính xác và đáng tin cậy với sai số từ cảm biến mong muốn khoảng  $\pm 1.5\%$  đến  $\pm 3\%$ .
- Độ trễ thấp: Hoạt động một cách nhanh chóng và hiệu quả trong khoảng 1-5 ms khi có sự kích hoạt từ các sự kiện đã định trước hoặc yêu cầu từ người dùng, giảm thiểu thời gian chờ đợi trong quá trình thu thập hoặc phân tích dữ liệu. Tương tự đối với thời gian đáp ứng cảm biến.
- Kích thước màn hình: Yêu cầu màn hình có kích thước từ 2.8 inch giúp người dùng dễ dàng xem thông số trạng thái của tủ bảo quản một cách rõ ràng và thuận tiện.
- Thiết kế: Nhỏ gọn với kích thước của thiết bị mong muốn, sử dụng adapter 12V, 5A để cấp nguồn.
- Độ vượt ngưỡng nhiệt độ, độ ẩm cố định khi điều khiển dòng là  $\sim 1^{\circ}\text{C}$

## 2.3. Sơ đồ khối hệ thống



Hình 2.1 Sơ đồ khối hệ thống

### 2.3.1. Khối cảm biến

**Vai trò:** Thu thập dữ liệu đầu vào cho hệ thống, cụ thể là nhiệt độ hiện tại.

**Đầu vào:** Nhiệt độ môi trường hoặc nhiệt độ từ bề mặt làm lạnh/làm nóng.

**Đầu ra:** Tín hiệu nhiệt độ (dạng tín hiệu tương tự hoặc số) gửi đến khối xử lý.

Ở đây chúng em lựa chọn BME280 là một cảm biến môi trường đa chức năng do **Bosch Sensortec** sản xuất. Nó được thiết kế để đo **nhiệt độ, áp suất khí quyển, và độ ẩm**, với độ chính xác cao khi đo

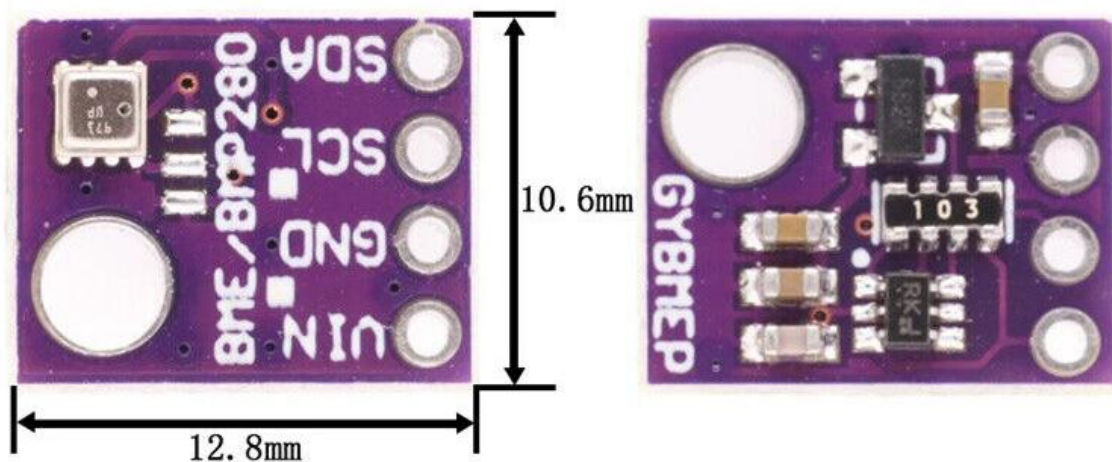
**Đo nhiệt độ:**

- + Dải đo:  $-40^{\circ}\text{C}$  đến  $+85^{\circ}\text{C}$ .
- + Độ chính xác:  $\pm 1.0^{\circ}\text{C}$ .

**Đo độ ẩm:**

- + Dải đo: 0% đến 100%.
- + Độ chính xác:  $\pm 3\%$ .

**Kích thước nhỏ gọn:** 2.5mm x 2.5mm x 0.93mm.



Hình 2.2 Cảm biến BME280

### 2.3.2. Khối xử lý

- **Vai trò:**

- Là "bộ não" của hệ thống, quyết định hoạt động điều khiển dựa trên dữ liệu từ khối cảm biến.
- Tính toán thuật toán điều khiển (PID) để đạt nhiệt độ mục tiêu.
- Gửi tín hiệu điều khiển đến khối thực thi.

- **Đầu vào:**

- Tín hiệu nhiệt độ từ khối cảm biến.
- Giá trị nhiệt độ mục tiêu (Setpoint), được lập trình trước hoặc điều chỉnh bởi người dùng.

- **Hoạt động:**

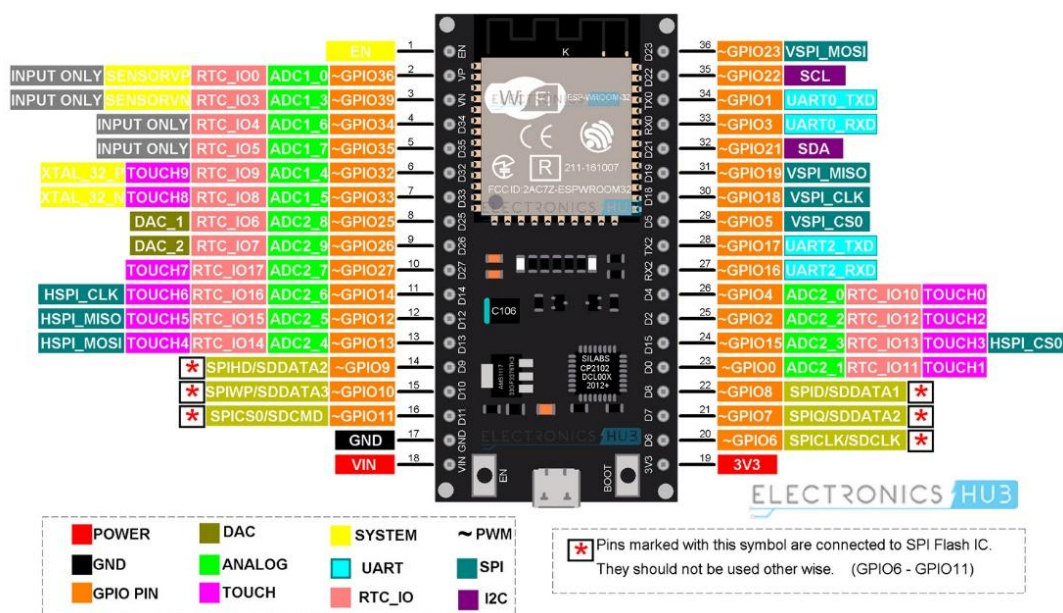
- So sánh nhiệt độ hiện tại với nhiệt độ mục tiêu.
- Quyết định chế độ hoạt động: làm nóng, làm lạnh, hoặc duy trì nhiệt độ.
- Tính toán giá trị điều khiển phù hợp (dựa trên PID hoặc các thuật toán khác).

- **Đầu ra:**

- Tín hiệu điều khiển gửi đến khối thực thi.

Chúng em sử dụng ESP32 một chip kết hợp Wi-Fi và Bluetooth tần số 2,4 GHz đơn, được thiết kế với công nghệ tiết kiệm năng lượng 40 nm của TSMC. Chip này được thiết

kể để đạt hiệu suất năng lượng và RF tốt nhất, thể hiện tính bền vững, linh hoạt và độ tin cậy trong nhiều ứng dụng.



Hình 2.3 Vi xử lý ESP32

Thông số kỹ thuật	ESP32
MCU	Xtensa® Dual-Core 32-bit LX6 600 DMIPS
802.11b/g/n Wifi	Yes, HT40
Bluetooth	Bluetooth 4.2 and below
Typical Frequency	160Mhz
SRAM	512 kbytes
Flash	SPI Flash, up to 16 MBytes
GPIO	36
Hardware/ Software PWM	1/16 Channels
SPI/I2C/I2S/UART	4/2/2/2
Working Temperature	-40oC to 125oC

Bảng 2.1. Một vài thông số cơ bản của ESP32

### 2.3.3. Khởi thực thi



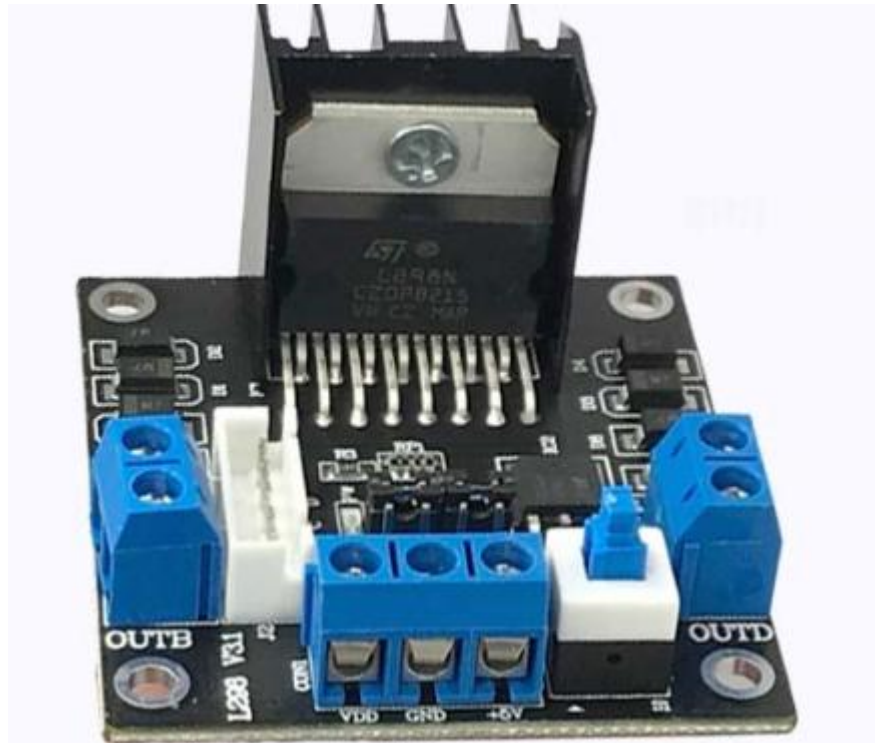
Hình 2.4 TEC1-12706

Thông số kỹ thuật	TEC1 – 12706
Điện áp đầu vào	12VDC
Dòng điện đầu vào tối đa	6A
Công suất làm lạnh tối đa	60W
Chênh lệch nhiệt độ tối đa	67 độ C
Kích thước	40 x 40 x 3.8mm

### L298N

- Module Điều Khiển Động Cơ L298 V3 là module điều khiển động cơ tích hợp 2 mạch cầu H sử dụng MOSFET kênh N và kênh P.
- Điều khiển 2 động cơ DC hoặc 1 động cơ bước 2 pha 4 dây.
- Dải điện áp hoạt động từ 5-35VDC.
- Dòng liên tục trên mỗi kênh là 2A.
- Điều khiển tốc độ động cơ bằng PWM (phương pháp điều chế độ rộng xung).
- Kết nối dễ dàng với vi điều khiển và động cơ thông qua các connector tích hợp trên mạch.



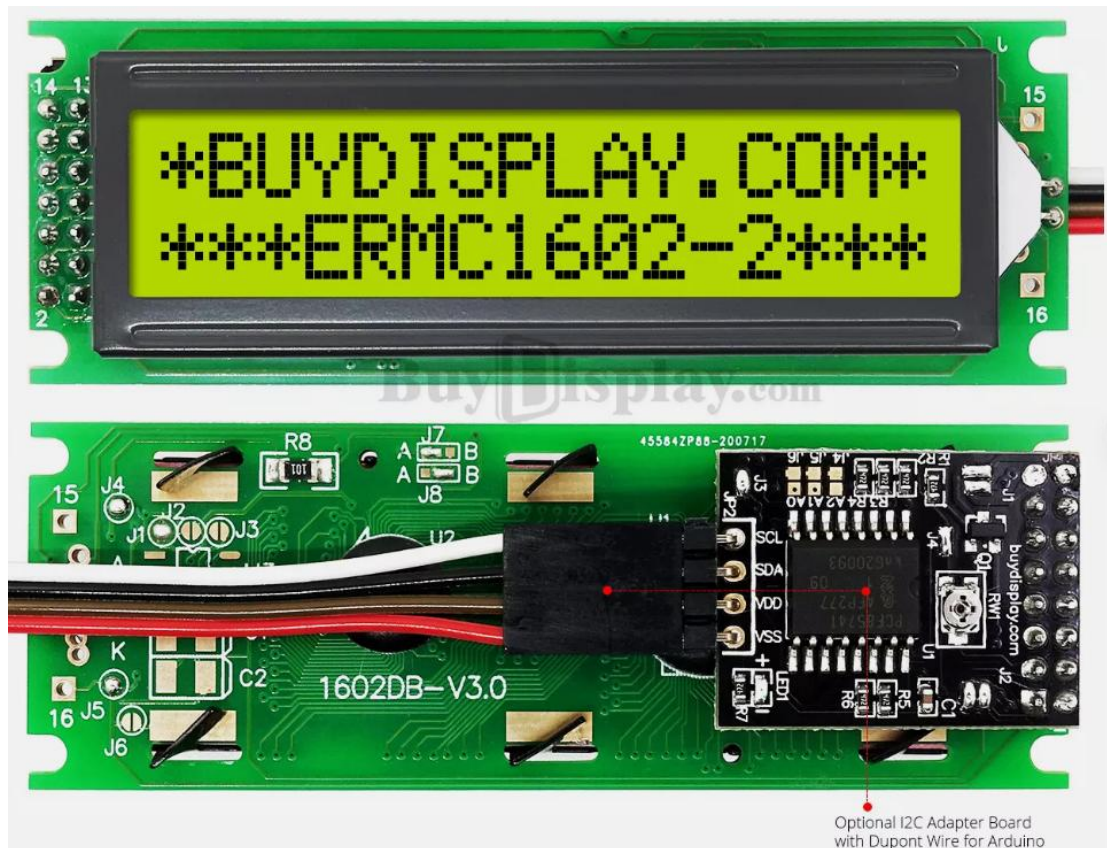


Hình 2.5 Module L298N

#### 2.3.4. Khối hiển thị

- **Vai trò:**
  - Hiển thị các thông tin cần thiết về hệ thống cho người dùng.
  - Có thể cung cấp giao diện tương tác để người dùng điều chỉnh nhiệt độ mục tiêu hoặc quan sát trạng thái hệ thống.
- **Đầu vào:**
  - Dữ liệu từ khối xử lý (nhiệt độ hiện tại, trạng thái hệ thống, giá trị PID, v.v.).
- **Đầu ra:**
  - Thông tin hiển thị trực quan như nhiệt độ, trạng thái (làm nóng/lạnh), và các thông số khác.

Màn hình LCD 16×2 là một loại màn hình sử dụng công nghệ crystal lỏng (LCD). Nó có kích thước nhỏ gọn, dễ sử dụng và có thể kết hợp với Arduino để hiển thị dữ liệu. Với 16 cột và 2 hàng, màn hình này có thể hiển thị tối đa 32 ký tự.



Hình 2.6 LCD1602 & Module I2C

### 2.3.5. Luồng hoạt động của hệ thống

1. **Khối cảm biến** thu thập dữ liệu nhiệt độ thực tế từ môi trường hoặc thiết bị cần kiểm soát.
2. Dữ liệu được gửi đến **khối xử lý** để so sánh với nhiệt độ mục tiêu.
3. **Khối xử lý** tính toán và gửi lệnh điều khiển đến **khối thực thi**:
  - Nếu nhiệt độ quá cao: Kích hoạt chế độ làm lạnh.
  - Nếu nhiệt độ quá thấp: Kích hoạt chế độ làm nóng.
4. **Khối thực thi** thực hiện lệnh để đạt được nhiệt độ mong muốn.
5. Trạng thái và thông tin nhiệt độ được gửi đến **khối hiển thị**, giúp người dùng giám sát hệ thống.

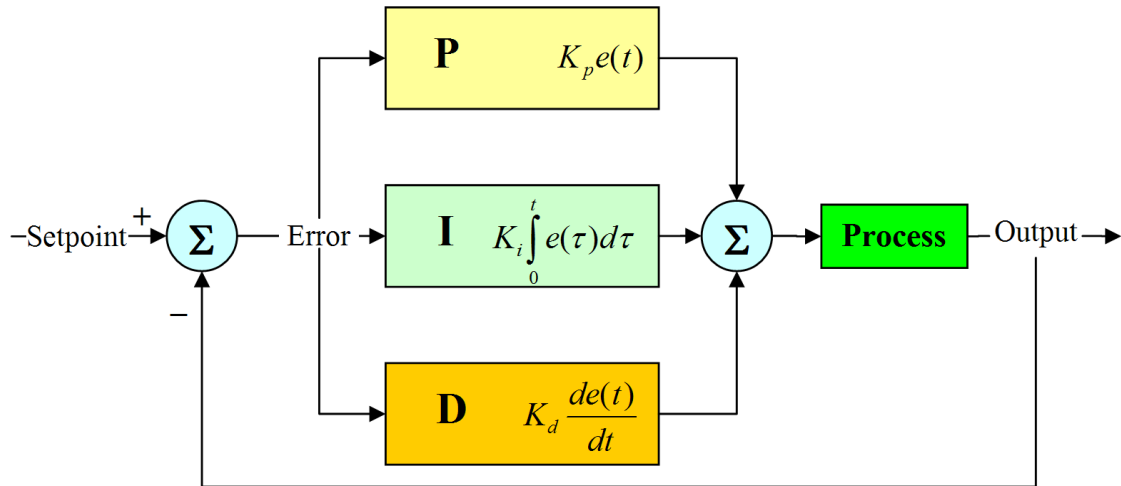


STT	Tên linh kiện	Thông số kỹ thuật chính	Ghi chú
1	Vi điều khiển ESP32	Bộ điều khiển chính	-
2	LCD 16x2 (kèm module I2C)	16 cột, 2 hàng, hỗ trợ I2C để tiết kiệm năng lượng	Hiển thị nhiệt độ, độ ẩm
3	Cảm biến BME280	Đo nhiệt độ, độ ẩm, áp suất; độ chính xác $\pm 3\%$	Nhỏ gọn, giá ~150,000 VNĐ
4	TEC1-12706 (Sò nóng lạnh)	Điện áp tối đa 15.4 V; dòng tối đa 6 A; công suất 60 W	Làm nóng/lạnh cho tủ điện tử
5	Quạt tản nhiệt	Điện áp 12VDC; công suất 3W; kích thước 6x6x1.5 cm	Tản nhiệt cho mặt nóng của TEC1
6	Module mạch cầu H L298N	Dòng tối đa 2 A/kênh; hỗ trợ PWM điều khiển TEC1	Điều khiển luân phiên nóng/lạnh

## CHƯƠNG 3: THIẾT KẾ HỆ THỐNG

### 3.1. Thuật toán PID

PID một bộ điều khiển phổ biến trong các hệ thống tự động để duy trì một giá trị cố định (setpoint) bằng cách điều chỉnh đầu ra của hệ thống. Bộ PID hoạt động dựa trên sự sai lệch (error) giữa giá trị thực tế và giá trị mong muốn.



Hình 3.1 Cách hoạt động của thuật toán PID

PID gồm ba thành phần chính:

#### **P - Proportional (Tỷ lệ):**

- Điều chỉnh đầu ra dựa trên **độ lớn của sai lệch** (error):

$$e(t) = \text{Setpoint} - \text{Output} (SP - PV)$$

- Công thức:  $P_{out} = K_p \times \text{error}$
- Nếu  $K_p$  lớn, phản ứng của hệ sẽ nhanh nhưng có thể gây dao động.

#### **I - Integral (Tích phân):**

- Tích lũy sai lệch theo thời gian, giúp loại bỏ sai lệch nhỏ kéo dài (offset).
- Công thức:  $I_{out} = K_i \times \int \text{Error} dt$
- Nếu  $K_i$  lớn, hệ có thể phản ứng chậm nhưng bù được sai số ổn định.

**D - Derivative (Đạo hàm):**

- Dự đoán xu hướng thay đổi của sai lệch, giúp giảm dao động và tăng độ ổn định.
- Công thức:  $D_{out} = K_d \times d(Error)/dt$
- Nếu  $K_d$  lớn, hệ sẽ phản ứng nhanh hơn nhưng dễ bị ảnh hưởng bởi nhiễu.

**Công thức điều khiển PID tổng quát:**

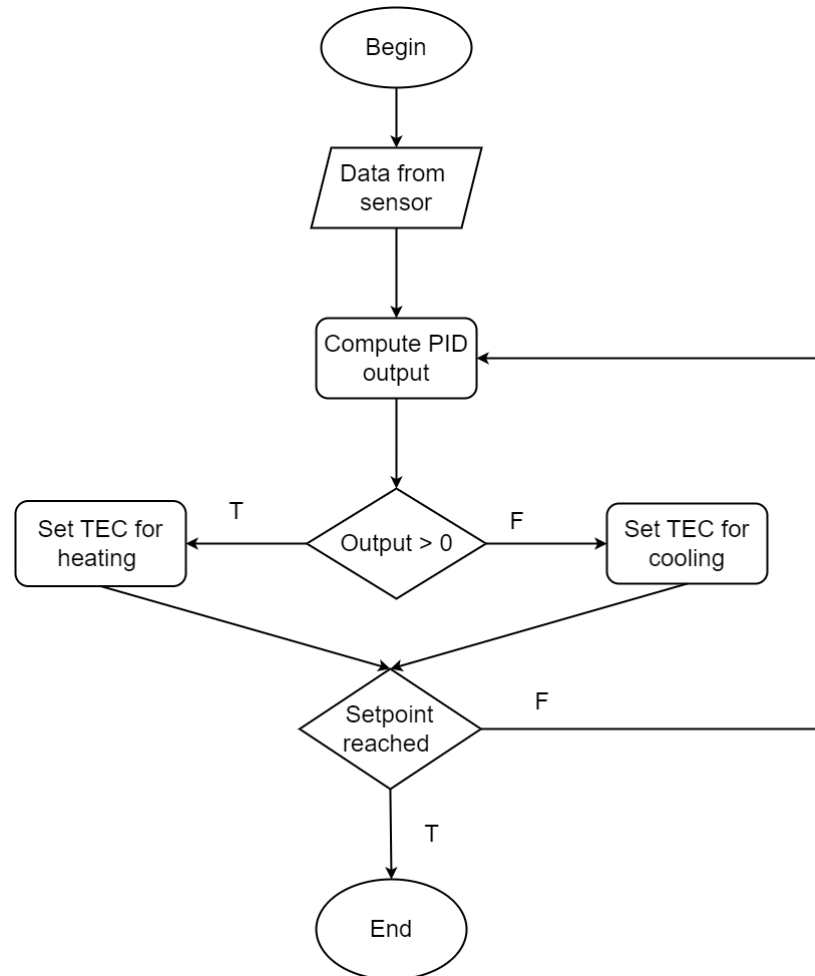
$$Output = K_p \times Error + K_i \times \int (Error)dt + K_d \times d(Error)/dt$$

**Error:** Sự sai lệch giữa giá trị đo được và giá trị mong muốn.

**Kp, Ki, Kd:** Các tham số điều chỉnh (gain) của từng thành phần.

**Ứng dụng PID cho hệ thống:**

- Đạt nhiệt độ nhanh hơn mà không vượt quá Setpoint (hạn chế overshoot).
- Giữ nhiệt độ ổn định trong thời gian dài (giảm dao động xung quanh Setpoint).
- Cần hiệu quả năng lượng (giảm bật/tắt TEC liên tục).



Hình 3.2 Sơ đồ thuật toán hệ thống áp dụng PID

### Phương pháp điều chỉnh hệ số PID

Mỗi hệ thống lại có 1 bộ PID riêng vì thế có rất nhiều phương pháp khác nhau để điều chỉnh PID.

#### 1) Điều chỉnh thủ công

B1: Đặt  $K_i = 0$ ,  $K_d = 0$ .

B2: Tăng  $K_p$  đến khi vòng điều khiển bắt đầu dao động, sau đó giảm  $K_p$  xuống khoảng một nửa để đạt đáp ứng “1/4 suy giảm biên độ”.

B3: Tăng  $K_i$  để cải thiện thời gian xử lý, nhưng tránh làm mất ổn định.

B4: Nếu cần, tăng  $K_d$  để giảm thời gian hồi phục, nhưng tránh vượt quá mức.

→ Phương pháp này dễ thực hiện và không yêu cầu công thức phức tạp. Phù hợp với hệ thống cần giữ trạng thái online. Có thể gây vượt lỗi nhẹ; giảm  $K_p$  hơn nữa nếu cần hạn chế vượt lỗi.

## 2) Phương pháp Ziegler-Nichols

B1: Đặt  $K_i = 0$ ,  $K_d = 0$ .

B2: Tăng  $K_p$  cho đến khi hệ thống dao động ổn định tại độ lợi tới hạn ( $K_u$ ).

B3: Ghi lại thời gian dao động ( $P_u$ ).

B4: Áp dụng công thức để tính thông số PID:

$$+ \text{ P: } K_p = 0.5K_u$$

$$+ \text{ PI: } K_p = 0.45K_u, K_i = \frac{1.2K_p}{P_u}$$

$$+ \text{ PID: } K_p = 0.6K_u, K_i = \frac{2K_p}{P_u}, K_d = \frac{K_p \times P_u}{8}$$

→ Ưu điểm: dễ áp dụng và cung cấp thông số khởi đầu nhanh chóng. Hiệu quả cao với hệ thống tuyến tính nhưng có thể cần tinh chỉnh thêm cho hệ thống phi tuyến. Phù hợp cho điều chỉnh PID tự động và mô phỏng.

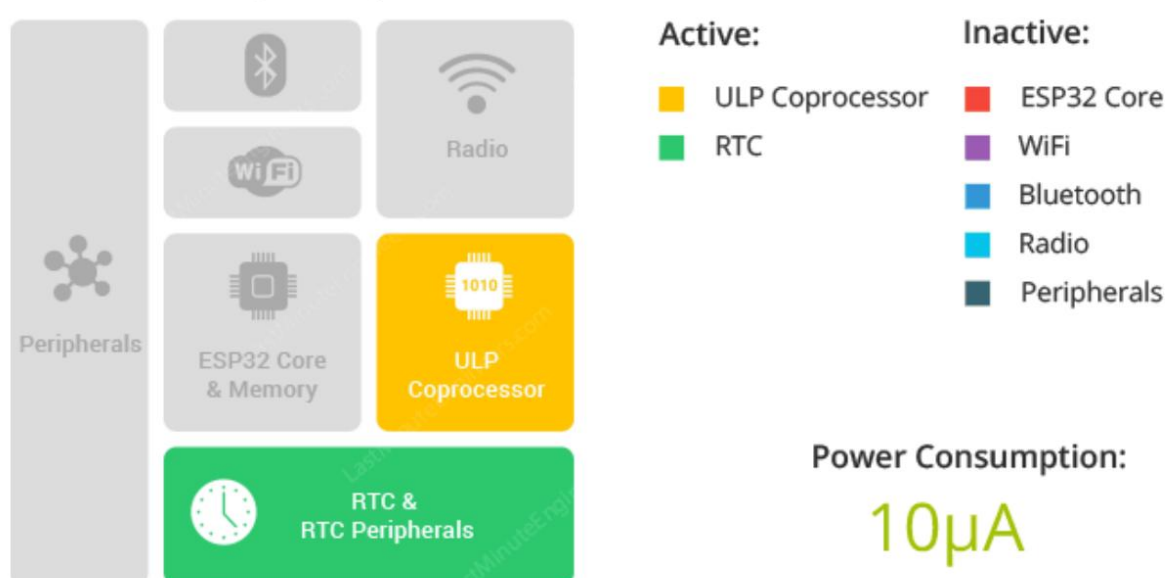
## 3.2. Chế độ Deep Sleep

Các dòng vi điều khiển khác nhau đều có chế độ tiết kiệm năng lượng (SLEEP) với hiệu suất khác nhau.

- Active Mode.
- Modem Sleep Mode.
- Light Sleep Mode.
- Deep Sleep Mode.
- Hibernation Mode.

Trong chế độ Deep Sleep, tất cả các hoạt động của ESP32 đều được tắt, chỉ còn duy nhất một số linh kiện được duy trì để giữ cho ESP32 hoạt động. Các linh kiện này bao gồm:

- RTC (Real-Time Clock): đồng hồ thời gian thực để giữ cho ESP32 có thể đánh thức sau khi thời gian Deep Sleep kết thúc.
- ULP (Ultra-Low Power Co-processor): ULP là một bộ xử lý siêu tiết kiệm năng lượng được tích hợp sẵn trong ESP32 và có thể được sử dụng để theo dõi các cảm biến, tính toán nhanh hoặc truyền nhận dữ liệu.



Hình 3.3 Sơ đồ các khối trong chế độ Deep Sleep

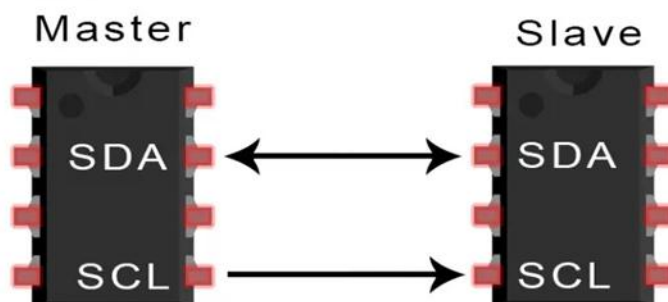
### 3.3. Giao thức sử dụng

#### 3.3.1. Giao tiếp I2C

Chuẩn giao tiếp I2C (Inter-Integrated Circuit) là một giao thức truyền thông nối tiếp được xây dựng trên nguyên tắc Master-Slave.

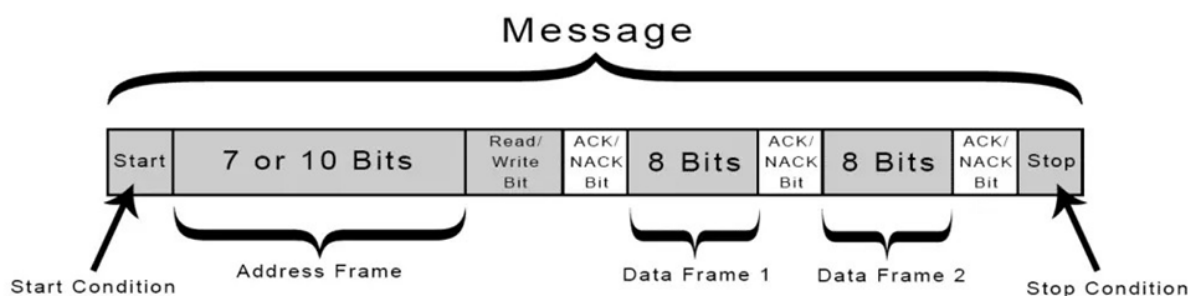
Master điều khiển quá trình truyền và nhận dữ liệu từ các Slave. Một Master có thể kết nối với nhiều Slave, và mỗi Slave được xác định bằng một địa chỉ duy nhất.

Trong đó tín hiệu (SCL) được Master điều khiển, dữ liệu (bit) được truyền từng bit một qua đường SDA.



Hình 3.4 Nguyên tắc I2C

**Nguyên lý hoạt động:** dữ liệu được truyền dưới dạng tin nhắn và được chia thành các khung dữ liệu. Mỗi tin nhắn bao gồm một khung địa chỉ chứa địa chỉ nhị phân của thiết bị Slave và một hoặc nhiều khung dữ liệu chứa dữ liệu được truyền.



Hình 3.5 Khung dữ liệu trong I2C

**Điều kiện bắt đầu:** Dây SDA chuyển từ mức điện áp cao xuống mức điện áp thấp trước khi dây SCL chuyển từ mức cao xuống mức thấp.

**Điều kiện dừng:** Dây SDA chuyển từ mức điện áp thấp lên mức điện áp cao sau khi dây SCL chuyển từ mức thấp lên mức cao.

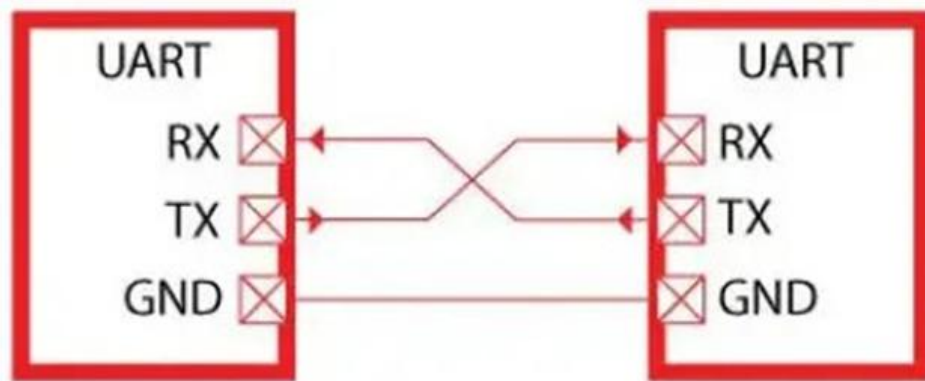
**Khung địa chỉ:** Đây là một chuỗi duy nhất gồm 7 hoặc 10 bit để xác định địa chỉ của thiết bị Slave mà Master muốn truyền thông.

**Read/Write Bit:** Đây là một bit đơn xác định liệu Master đang gửi dữ liệu đến Slave (mức điện áp thấp) hay yêu cầu dữ liệu từ Slave (mức điện áp cao).

**ACK/NACK Bit:** Mỗi khung dữ liệu trong một tin nhắn được theo sau bởi một bit ACK/NACK. Nếu khung địa chỉ hoặc khung dữ liệu được Slave nhận thành công, một bit ACK sẽ được trả lại cho Master từ Slave để xác nhận.

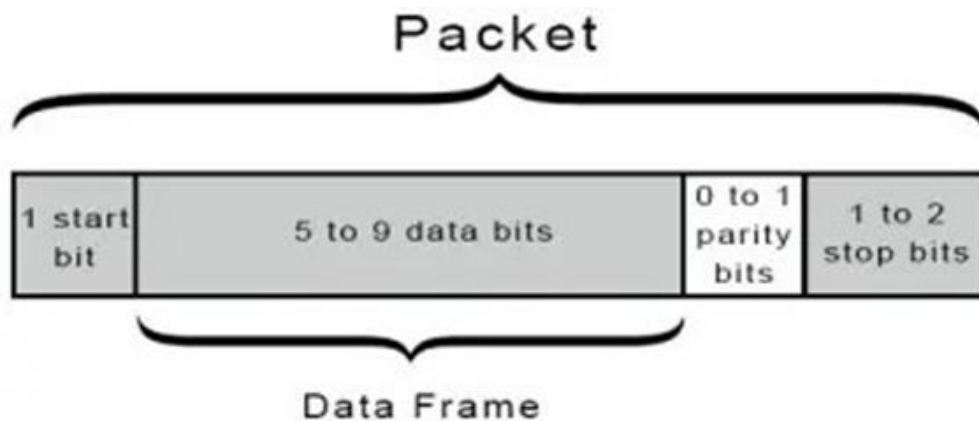
### 3.3.2. Giao tiếp UART

Bên cạnh những phương thức giao tiếp đồng bộ thì, UART (Universal Asynchronous Receiver-Transmitter), là bộ truyền nhận dữ liệu nối tiếp bất đồng bộ, có nghĩa là không có tín hiệu đồng hồ để đồng bộ hóa đầu ra của các bit từ UART truyền đến việc lấy mẫu các bit bởi UART nhận.



Hình 3.6 Phương thức truyền UART

UART truyền chuyển đổi dữ liệu song song từ một thiết bị điều khiển như CPU thành dạng nối tiếp, truyền nó nối tiếp đến UART nhận, sau đó chuyển đổi dữ liệu nối tiếp trở lại thành dữ liệu song song cho thiết bị nhận.



Hình 3.7 Gói dữ liệu trong UART

**Start bit:** Bắt đầu truyền, đường truyền được kéo từ mức cao xuống thấp.

**Data frame:** Khung dữ liệu chính, từ 5-9 bit, thường truyền LSB (bit có trọng số nhỏ nhất) trước.

**Parity bit (tùy chọn):** Kiểm tra lỗi dựa trên tính chẵn/lẻ của số bit 1.

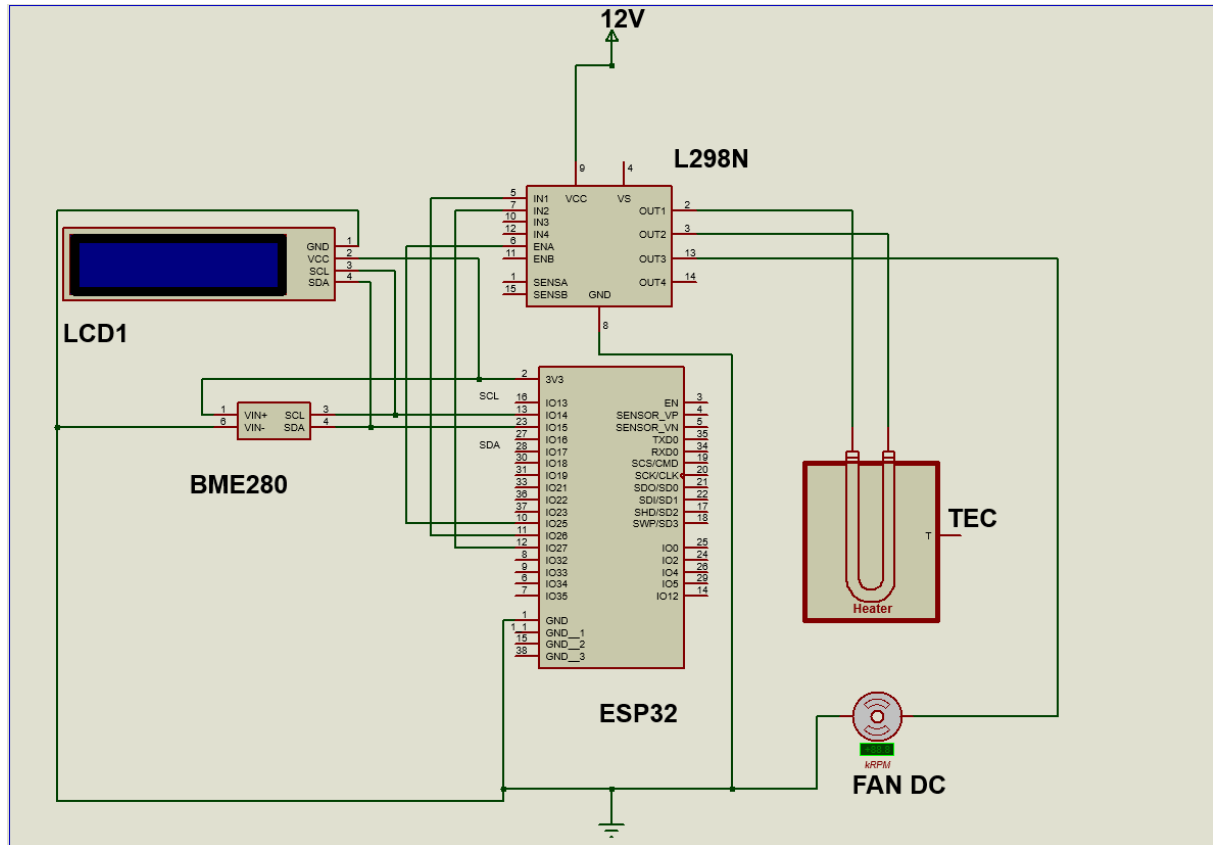


**Stop bit:** Báo hiệu kết thúc gói, mức điện áp được kéo cao trong ít nhất 1-2 chu kỳ bit.

Các gói dữ liệu này phải được truyền trên cùng 1 tốc độ (Baudrate) để đảm bảo việc lấy mẫu đúng thời điểm, giảm sự sai lệch tổng quá trình truyền nhận.

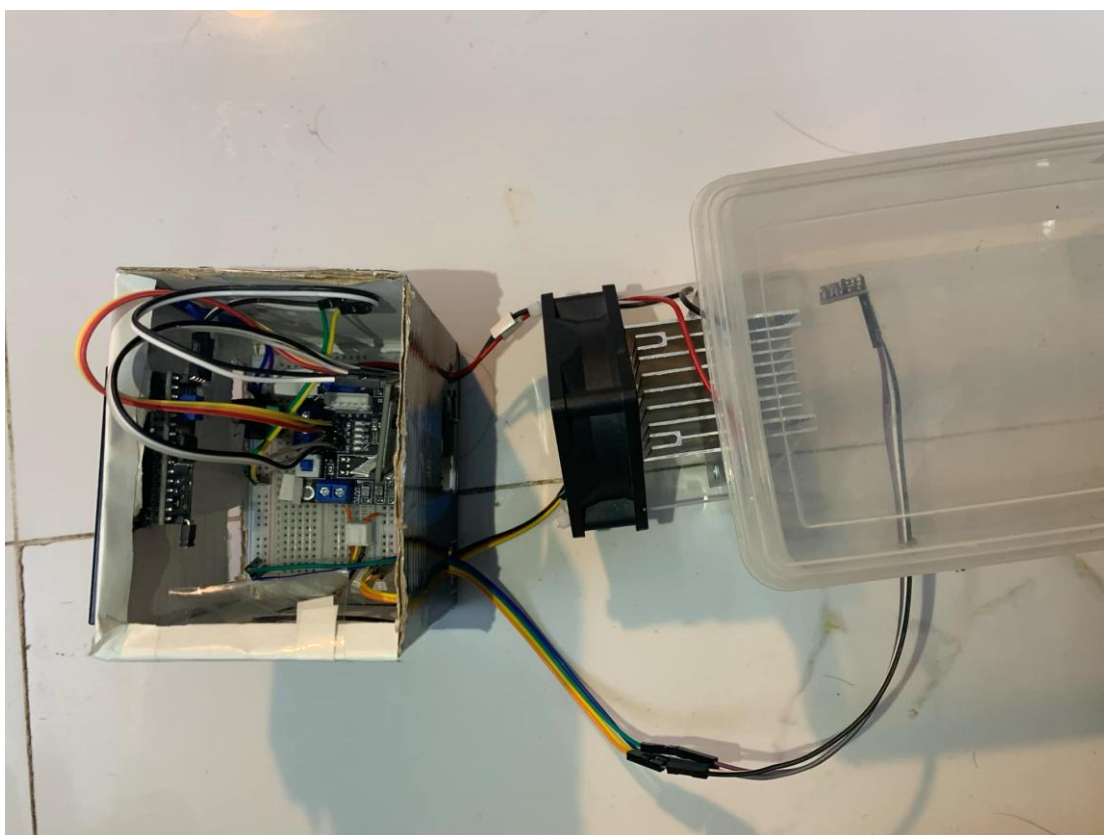
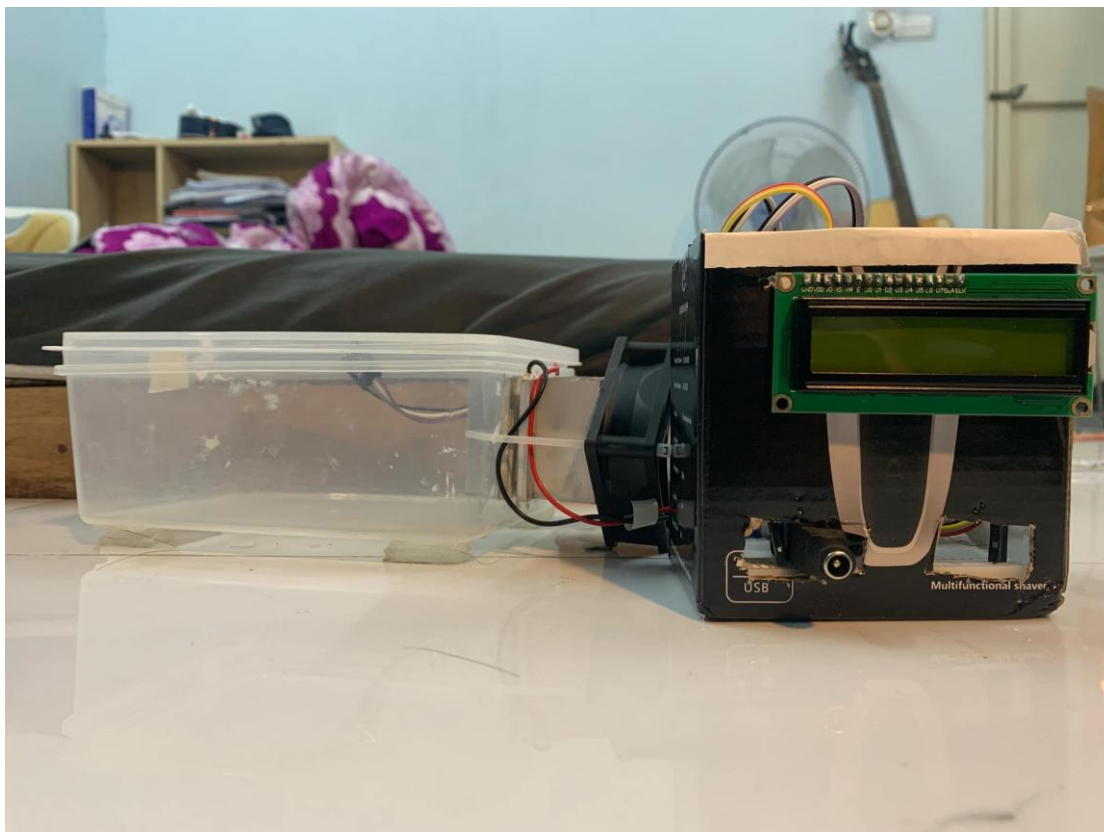
### 3.4. Thiết kế phần cứng

#### 3.4.1. Sơ đồ mạch



Hình 3.8 Sơ đồ mạch

### 3.4.2. Thiết kế mô hình



*Mô hình thiết kế sản phẩm*

### 3.4.3. Tính toán công suất

Từ nguồn 12V – 3A cung cấp, ta tính toán được công suất tiêu thụ trên từng khối và toàn mạch

#### a) Vi xử lý ESP32

Trong quá trình thiết kế hệ thống, nhằm tối ưu năng lượng tiêu thụ cho toàn mạch, nhóm quyết định sử dụng chế độ tiết kiệm năng lượng Deep Sleep của ESP32 với nguồn cấp 5V.

Tại Active mode, dòng điện tiêu thụ cho các chế độ làm việc gồm Wifi và BT/BLE đạt tối đa là  $I_{\max} = 240 \text{ mA}$ .

Với nguồn cung cấp  $U = 5\text{V}$  cấp riêng, ta tính được công suất tiêu thụ:

$$P_{\text{ESP32,active}} = U \times I = 5 \times 240 \times 10^{-3} = 1.2 \text{ (W)}$$

Power mode	Description			Power Consumption
Active (RF working)	Wi-Fi Tx packet			Please refer to Table 5-4 for details.
	Wi-Fi/BT Tx packet			
	Wi-Fi/BT Rx and listening			
Modem-sleep	The CPU is powered up.	240 MHz <sup>*</sup>	Dual-core chip(s)	30 mA ~ 68 mA
			Single-core chip(s)	N/A
		160 MHz <sup>*</sup>	Dual-core chip(s)	27 mA ~ 44 mA
			Single-core chip(s)	27 mA ~ 34 mA
		Normal speed: 80 MHz	Dual-core chip(s)	20 mA ~ 31 mA
			Single-core chip(s)	20 mA ~ 25 mA
Light-sleep	-			0.8 mA
Deep-sleep	The ULP coprocessor is powered up.			150 μA
	ULP sensor-monitored pattern			100 μA @1% duty
	RTC timer + RTC memory			10 μA
Hibernation	RTC timer only			5 μA
Power off	CHIP_PU is set to low level, the chip is powered down.			1 μA

Bảng 3.1. Bảng năng lượng tiêu thụ của ESP32

Dòng tiêu thụ của ESP32 trong chế độ Deep Sleep:

$$I = 150 + 100 + 10 = 260 \text{ (}\mu\text{A)}$$

Ta tính được công suất tiêu thụ:

$$P_{\text{ESP32,deepsleep}} = U \times I = 5 \times 260 = 1300 \text{ (}\mu\text{W)}$$

b) BME280

Current consumption	1.8 $\mu\text{A}$ @ 1 Hz humidity and temperature 2.8 $\mu\text{A}$ @ 1 Hz pressure and temperature 3.6 $\mu\text{A}$ @ 1 Hz humidity, pressure and temperature 0.1 $\mu\text{A}$ in sleep mode
---------------------	--

Bảng 3.2. Năng lượng tiêu thụ của BME280

Dòng điện tiêu thụ cho chế độ đo của cảm biến gồm độ ẩm, áp suất và nhiệt độ đạt  $I = 3.6 \mu\text{A}$ .

Với nguồn cung cấp  $U = 3.3\text{V}$ , ta tính được công suất tiêu thụ:

$$P_{\text{BME}} = U \times I = 3.3 \times 3.6 \times 10^{-6} = 11.88 (\mu\text{W})$$

c) TEC1-12706

Th( $^{\circ}\text{C}$ )	27	50	Hot side temperature at environment: dry air, N <sub>2</sub>
DT <sub>max</sub> ( $^{\circ}\text{C}$ )	70	79	Temperature Difference between cold and hot side of the module when cooling capacity is zero at cold side
U <sub>max</sub> (Voltage)	16.0	17.2	Voltage applied to the module at DT <sub>max</sub>
I <sub>max</sub> (amps)	6.1	6.1	DC current through the modules at DT <sub>max</sub>
Q <sub>Cmax</sub> (Watts)	61.4	66.7	Cooling capacity at cold side of the module under DT=0 $^{\circ}\text{C}$
AC resistance(ohms)	1.8 ~ 2.2	2.0~2.4	The module resistance is tested under AC

Bảng 3.3. Năng lượng tiêu thụ của TEC1-12706

Theo datasheet của TEC1–12706, dòng tiêu thụ đạt tối đa khi nhiệt độ mặt nóng đạt 50  $^{\circ}\text{C}$  là  $I = 3\text{A}$

Công suất tiêu thụ của TEC1–12706 với nguồn cấp  $U = 12\text{V}$  là:

$$P_{\text{TEC}} = U \times I = 12 \times 3\text{A} = 36 (\text{W})$$

d) LCD1602

Characteristics	Symbol	Limit			Unit	Test Condition
		Min.	Typ.	Max.		
Operating Current	I <sub>CC</sub>	-	0.2	0.4	mA	External clock (Note)
Input High Voltage	V <sub>IH1</sub>	0.7VDD	-	VDD	V	Pins: (E, RS, R/W, DB7 - 0)
Input Low Voltage	V <sub>IL1</sub>	-0.3	-	0.55	V	
Input High Voltage	V <sub>IH2</sub>	0.7VDD	-	VDD	V	Pin OSC1
Input Low Voltage	V <sub>IL2</sub>	-0.2	-	0.2VDD	V	
Input High Current	I <sub>IH</sub>	-1.0	-	1.0	$\mu\text{A}$	Pins: (RS, R/W, DB7 - 0) VDD = 3.0V
Input Low Current	I <sub>IL</sub>	-5.0	-15	-30	$\mu\text{A}$	
Output High Voltage (TTL)	V <sub>OH1</sub>	0.75VDD	-	-	V	I <sub>OH</sub> = - 0.1mA Pins: DB7 - 0
Output Low Voltage (TTL)	V <sub>OL1</sub>	-	-	0.2VDD	V	I <sub>OL</sub> = 0.1mA Pins: DB7 - 0

Bảng 3.4. Năng lượng tiêu thụ của LCD1602

Theo Datasheet của màn hình LCD1602, ta có  $I_{\max} = 0.4 \text{ mA}$ .

Công suất tiêu thụ của LCD1602 trong trạng thái hoạt động tối đa là:

$$P_{\text{LCD}} = U \times I = 5 \times 0.4 \times 10^{-3} = 2 \text{ (mW)}$$

e) Module L298

Symbol	Parameter	Value	Unit
$V_S$	Power supply	50	V
$V_{SS}$	Logic supply voltage	7	V
$V_I, V_{en}$	Input and enable voltage	-0.3 to 7	V
$I_O$	Peak output current (each channel):		
	• Non repetitive (t = 100 ms)	3	A
	• repetitive (80% on -20% off; $t_{on} = 10 \text{ ms}$ )	2.5	A
	• DC operation	2	A
$V_{sens}$	Sensing voltage	-1 to 2.3	V
$P_{tot}$	Total power dissipation ( $t_{case} = 75^\circ\text{C}$ )	25	W
$T_{op}$	Junction operating temperature	-25 to 130	$^\circ\text{C}$
$T_{stg}, T_j$	Storage and junction temperature	-40 to 150	$^\circ\text{C}$

Bảng 3.5. Năng lượng tiêu thụ của L298

Từ bảng có thể thấy dòng điện tiêu thụ đạt max với  $I_{\max} = 2\text{A}$  với mỗi kênh.

Với công thức  $P = U.I$  và nguồn cấp 12V, công suất tiêu thụ tối đa của module L298 là:

$$P_{\text{LCD}} = U \times I = 12 \times 2 = 24 \text{ (W)}$$

### Trường hợp 1: ESP32 hoạt động ở Active Mode

Tổng công suất:  $P_{tol} = P_{esp32} + P_{bme} + P_{tec} + P_{lcd} + P_{l298}$

$$= 1.2 + (11.88 \times 10^{-6}) + 36 + (2 \times 10^{-3}) + 24 \approx 61.2\text{W}$$

### Trường hợp 2: ESP32 hoạt động ở Deep Sleep Mode

Tổng công suất:  $P_{tol} = P_{esp32 \text{ deep}} + P_{bme} + P_{tec} + P_{lcd} + P_{l298}$

$$= (2.3 \times 10^{-3}) + (11.88 \times 10^{-6}) + 36 + (2 \times 10^{-3}) + 24 \approx 60\text{W}$$

⇒ Sử dụng thuật toán PID, tản nhiệt và các chế độ khác nhau để tối ưu hóa năng lượng tiêu thụ

## CHƯƠNG 4: KIỂM THỬ VÀ ĐÁNH GIÁ

### 4.1. Kiểm thử

Chúng em đã thực hiện các test để đánh giá sự hiệu quả của các bộ PID như sau:

#### 4.1.1. Điều chỉnh PID của chế độ làm lạnh

Setpoint = 22, current ~ 23.2.

Test Step	Nhận xét
<a href="#">kp0 ki0 kp0 (cooling).xlsx</a>	(Kp = 0) Hệ thống không phản hồi, không thay đổi nhiệt độ.
<a href="#">kp10 ki0 kp0 (cooling).xls</a>	(Kp = 10) Hệ thống có phản hồi với xung PWM dao động trong khoảng 12- 13, tuy nhiên nhiệt độ không thay đổi đáng kể.
<a href="#">kp100 ki0 kp0 (cooling).xls</a>	(Kp = 100) Hệ thống phản hồi tốt, tuy nhiên nhiệt độ thay đổi chậm, không đạt được setpoint mong muốn. (22.9)

Bảng 4.1  $K_i, K_d = 0$

Test Step	Nhận xét
<a href="#">kp40 ki5 kp0 (cooling).xls</a>	( $K_i = 5$ ) Hệ thống phản hồi nhanh, và có đạt được setpoint, tuy nhiệt độ vượt quá setpoint 1 khoảng 0.5 ~0.8 C cho tới khi ổn định.
<a href="#">kp40 ki20 kp0 (cooling).xls</a>	( $K_i = 20$ ) Hệ thống phản hồi rất nhanh đạt setpoint trong thời gian ngắn, độ vượt lớn khoảng 1 – 1.2 C cho tới khi ổn định.

Bảng 4.2  $K_p = 40, K_i = 0$

Test Step	Nhận xét
<a href="#">kp40_ki5_kd10_cooling.xls</a>	$K_p = 40, K_i = 5, K_d = 10$ . Hệ thống phản hồi nhanh, tuy nhiên xung PWM dao động nhiều.
<a href="#">kp40_ki5_kd50_cooling.xls</a>	$K_p = 40, K_i = 5, K_d = 50$ . Hệ thống phản hồi chậm, PWM dao động mạnh.

Bảng 4.3  $K_p = 40, K_i = 5$ 

⇒  $K_p=40, K_i = 5, K_d = 1$ .

#### 4.1.2. Điều chỉnh PID của chế độ làm nóng

Tương tự quá trình làm lạnh, sự hiệu quả của bộ PID làm nóng là tương đương nhau. Vì thế, chúng em sẽ chọn những test case chính để đưa ra bộ PID tối ưu.

Test Step	Nhận xét
<a href="#">kp10_ki0_kp0_heating.xls</a>	Hệ thống có phản hồi với xung PWM dao động trong khoảng 12- 13, tuy nhiên nhiệt độ không thay đổi đáng kể.
<a href="#">kp40_ki5_kd0_heating.xls</a>	Hệ thống phản hồi nhanh chóng, tuy nhiên nhiệt độ dao động quanh setpoint khoảng 1-1.2 C
<a href="#">kp40_ki20_kd0.5_heating.xls</a>	Hệ thống phản hồi tốt, tuy nhiên xung PWM dao động lớn dẫn đến công suất bị ảnh hưởng.

Bảng 4.4 Test case làm nóng

⇒  $K_p=50, k_i=3, k_d=0.5$ .

Với bộ PID đã tối ưu ở bên trên, chúng em sẽ kiểm tra độ ổn định của hệ thống qua các test:

**4.1.3. Độ ổn định khi sử dụng bộ PID trong quá trình làm lạnh**

Nhiệt độ hiện tại ~ 23, giá trị Setpoint = 22

Data	Nhận xét
<a href="#">set22_cooling_01.xls</a>	Thời gian hệ thống đạt setpoint khoảng 1', nhìn chung các lần test case đều có sự tương đồng về các thông số. ⇒ Hệ thống hoạt động tương đối ổn định khi sử dụng bộ PID
<a href="#">set22_cooling_02.xls</a>	
<a href="#">set22_cooling_03.xls</a>	

Bảng 4.5 Độ ổn định trong chế độ làm lạnh

**4.1.4. Độ ổn định khi sử dụng bộ PID trong quá trình làm nóng**

Nhiệt độ hiện tại ~ 23.6, giá trị Setpoint = 25

Data	Nhận xét
<a href="#">set25_heating_01.xls</a>	Thời gian hệ thống đạt setpoint khoảng 2', nhìn chung các lần test case đều có sự tương đồng về các thông số. ⇒ Hệ thống hoạt động tương đối ổn định khi sử dụng bộ PID, tuy nhiên hiệu quả không cao bằng quá trình làm nóng
<a href="#">set25_heating_02.xls</a>	
<a href="#">set25_heating_03.xls</a>	

Bảng 4.6 Độ ổn định trong chế độ làm nóng

**4.1.5. Độ ổn định khi thay đổi chế độ luân phiên**

Hệ thống đang trong quá trình làm nóng được tác động nhiệt độ bên ngoài sẽ chuyển sang chế độ làm nóng một cách trực tiếp.

Data	Nhận xét
<a href="#">set24_heating_to_cooling_01.xls</a>	Hệ thống phản hồi tương đối nhanh với sự thay đổi trạng thái, đạt setpoint nhanh chóng với độ ổn định tốt.
<a href="#">set24_heating_to_cooling_02.xls</a>	
<a href="#">set24_heating_to_cooling_03.xls</a>	

Bảng 4.7 Test thay đổi chế độ luân phiên



## 4.2. Đánh giá

### 4.2.1. Hiệu suất hệ thống

#### Ưu điểm:

- Hệ thống có thể đạt được nhiệt độ mục tiêu nhờ vào PID control. Dù các lần test có thể khác nhau về thời gian, hệ thống vẫn phản hồi và điều chỉnh PWM để đạt được Setpoint. Điều này cho thấy PID đã thực hiện đúng nhiệm vụ điều khiển.
- Chế độ Deep Sleep giúp giảm đáng kể công suất tiêu thụ khi hệ thống không cần điều chỉnh nhiệt độ. Đây là một điểm mạnh cho các ứng dụng yêu cầu tối ưu hóa năng lượng.
- Việc sử dụng 2 bộ PID riêng biệt cho chế độ làm nóng và làm lạnh cho phép hệ thống thích nghi tốt hơn với các điều kiện nhiệt độ khác nhau, tránh xung đột trong điều khiển.

#### Hạn chế:

Trong một số trường hợp, thời gian để đạt và giữ ổn định nhiệt độ mục tiêu có thể còn dài, đặc biệt khi Setpoint thay đổi đột ngột. Điều này có thể do thông số PID chưa được tối ưu hoàn toàn.

### 4.2.2. Mức tiêu thụ năng lượng

#### Ưu điểm:

- Thiết kế tối ưu năng lượng: Hệ thống đã tính toán kỹ các mức tiêu thụ năng lượng từ ESP32, BME280, TEC, và các thành phần khác. Điều này cho thấy bạn có cái nhìn rõ ràng về mức độ tiêu hao năng lượng và khả năng duy trì hoạt động ổn định của hệ thống với nguồn 12V, 3A.

#### Hạn chế:

- Hiệu quả năng lượng của TEC: TEC tiêu thụ một lượng lớn công suất (tối đa 36W khi dòng 3A). Nếu PID không được tối ưu, có thể dẫn đến việc tiêu thụ năng lượng không cần thiết khi duy trì nhiệt độ gần Setpoint.

## CHƯƠNG V: KẾT LUẬN

Nhóm chúng em đã cơ bản hoàn thành các công việc được đặt ra và đạt được mục tiêu ban đầu của dự án. Sản phẩm đã đáp ứng đầy đủ các yêu cầu chức năng và phi chức năng cơ bản, đảm bảo hệ thống hoạt động ổn định, chính xác, và phù hợp với các tiêu chí kỹ thuật đề ra.

Trong quá trình triển khai và hoàn thiện sản phẩm, nhóm đã gặp phải một số lỗi và sự cố không mong muốn, khiến quá trình thực hiện tốn nhiều thời gian và công sức hơn dự kiến. Tuy nhiên, những khó khăn này đã trở thành cơ hội để nhóm học hỏi và tích lũy thêm nhiều kinh nghiệm quý giá trong việc phát triển, xử lý lỗi, và tối ưu hóa hệ thống.

Nhờ sự nỗ lực và tinh thần làm việc kiên trì, nhóm đã từng bước khắc phục các vấn đề tồn đọng, hoàn thiện sản phẩm một cách tối ưu nhất trong khả năng hiện tại. Dự án không chỉ giúp chúng em nâng cao kỹ năng chuyên môn mà còn củng cố khả năng làm việc nhóm, lập kế hoạch và giải quyết vấn đề thực tế.

Trong tương lai, nhóm hy vọng sẽ tiếp tục cải thiện sản phẩm bằng cách nâng cấp các tính năng, tối ưu hiệu suất hệ thống, và mở rộng khả năng ứng dụng để đáp ứng tốt hơn các yêu cầu thực tiễn.

## PHỤ LỤC

### TÀI LIỆU THAM KHẢO

- [1] Datasheet ESP32: [esp32\\_datasheet\\_en.pdf](#)
- [2] Datasheet BME280: [Bosch Sensortec BME280 Pressure, Temperature and Humidity Sensor Digital Output 1.8V 8-Pin LGA T/R | Compare Price | Easybom](#)
- [3] Datasheet TEC1-12706: <https://tecteg.com/wp-content/uploads/2017/06/TEC1-12706-site-ready.pdf>
- [4] Datasheet LCD1602: [waveshare.com/datasheet/LCD\\_en\\_PDF/LCD1602.pdf](http://waveshare.com/datasheet/LCD_en_PDF/LCD1602.pdf)
- [5] Datasheet L298: <https://www.st.com/resource/en/datasheet/l298.pdf>
- [6] Bộ điều khiển PID. [Online]. Available: [Bộ điều khiển PID – Wikipedia tiếng Việt](#)
- [7] Giới thiệu chuẩn giao tiếp I2C. [Online]. Available: [Giới thiệu chuẩn giao tiếp I2C HTE LINH KIẾN ĐIỆN TỬ PRO](#)
- [8] UART Protocol. [Online]. Available: [UART Protocol - Prodigy Technovations](#)