

ĐẠI HỌC BÁCH KHOA HÀ NỘI
TRƯỜNG ĐIỆN – ĐIỆN TỬ
KHOA TỰ ĐỘNG HÓA



BÁO CÁO BÀI TẬP LỚN
THIẾT KẾ HỆ THỐNG NHÚNG - EE4251
NHÓM 01:LẬP TRÌNH MẠCH ĐO NHIỆT
ĐỘ VÀ ĐẾM XUNG GIAO TIẾP MÁY
TÍNH QUA CHUẨN RS485-UART

Họ và tên:

Lê Văn Vinh-20210943

Lê Văn Công-20210128

Trần Văn Bắc-20212700

Giáo viên hướng dẫn

Thầy Lê Công Cường

Hà Nội- 1/2025

Lời cảm ơn

Trước tiên chúng em xin cảm ơn thầy Lê Công Cường đã có các hỗ trợ và hướng dẫn trong suốt quá trình làm bài tập lớn. Thông qua các hướng dẫn của thầy đã giúp chúng em tiết kiệm thời gian và tìm được các phương án phù hợp để giải quyết các yêu cầu của bài tập lớn. Thông qua việc thực hiện bài tập lớn đã giúp chúng em trau dồi các kỹ năng về triển khai phần cứng và khả năng lập trình phần mềm trên nền tảng các hệ thống nhúng cũng như khả năng quản lý công việc và phân chia thời gian trong việc thực hiện bài tập lớn một cách tiết kiệm và hiệu quả. Một lần nữa chúng em xin chân thành cảm ơn thầy.

Mục lục

Tóm tắt	4
Kế hoạch thực hiện.....	5
I.Thành phần mạch phần cứng	8
1.Bo mạch vi điều khiển STM32F103C8T6.....	8
2.Cảm biến đo nhiệt độ, áp suất BMP180.....	9
a.Giới thiệu.....	9
b.Nguyên lý hoạt động và giao tiếp	9
c.Hoạt động và độ chính xác	10
3.Cảm biến điện dung MPR121	11
a.Giới thiệu.....	11
b.Nguyên lý hoạt động và giao tiếp	11
c.Hoạt động	12
4.Màn hình LCD PCF8574	12
a.Giới thiệu.....	12
b.Nguyên lý hoạt động và giao tiếp	12
c.Hoạt động truyền và nhận dữ liệu	13
5. Module UART TTL to RS485 V2	14
a.Giới thiệu.....	14
b.Hoạt động và kết nối	14
c.Điều kiện hoạt động	15
6.Module UART CH340.....	16
a.Giới thiệu.....	16
b.Hoạt động và kết nối	16
7.Sơ đồ kết nối chân	17
II.Thiết kế phần mềm	18
1. Xác định các Task và các tham số	18
2.Phân tích tính đáp ứng yếu tố thời gian thực	22
3.Thuật toán và lập trình	23
III.Kết quả đánh giá.....	28
1.Các kết quả đánh giá	28
2.Hình ảnh mạch và hình ảnh giao diện web	28

Tóm tắt

I. Yêu cầu của bài tập lớn

-Các yêu cầu chính

- +Lập trình vi điều khiển STM32 đọc được nhiệt độ và đếm số xung
- +Truyền các thông số đọc được lên máy tính qua chuẩn RS485 và UART
- +Cài đặt thông số lập lịch và nhận các lệnh điều khiển từ máy tính
- Phương pháp dự định triển khai
 - + Sử dụng cảm biến BMP180 để đo nhiệt độ và thực hiện giao tiếp với STM32 bằng I2C
 - +Sử dụng cảm biến điện dung MPR121 để thực hiện đếm số lần chạm
 - +Sử dụng màn hình LCD để hiển thị các kết quả thông số đo được
 - +Tạo Sever để truyền thông số đo được lên hiển thị trên màn hình máy tính
 - +Tạo hệ điều hành FreeRTOS để quản lý các task đồng thời cho phép hệ thống nhận lệnh điều khiển từ máy tính

Kế hoạch thực hiện

Thời gian	Nhiệm vụ	Nội dung thực hiện	Trạng thái
Tuần 1	<ul style="list-style-type: none">-Họp để xác định yêu cầu đề tài, phương thức triển khai và các đối tượng phần cứng cần sử dụng-Tạo sơ đồ kết nối các thành phần chính của mạch-Tiến hành hàn linh kiện cố định thành 1 bo mạch tránh hiện tượng dây kết nối lỏng và tăng tính mỹ quan-Test mạch để đảm bảo mạch đo được các thông số yêu cầu và hiện thị lên LCD	<ul style="list-style-type: none">-Xác định các yêu cầu của bài toán về đo nhiệt độ và đếm xung-Xác định thành phần phần cứng gồm mạch STM32, màn hình LCD, cảm biến BMP180, cảm biến MPR121, mạch giao tiếp RS485 và UART-Hàn thành công mạch phần cứng-Thực hiện đo kiểm mạch để xác định mạch đo chính xác nhiệt độ và số xung đếm	Hoàn thành
Tuần 2	<ul style="list-style-type: none">-Phân tích yêu cầu về phần mềm để đảm bảo yếu tố về thời gian thực của hệ thống-Phân tích và tính toán các tham số cho các Task được phân chia-Phân tích yêu cầu bài tập theo mô hình đơn nhiệm-Phân tích yêu cầu bài tập theo yêu cầu mô hình đa nhiệm	<ul style="list-style-type: none">-Tiến hành xác định các task cần thực hiện-Đo thời gian thực thi của các task-Phân tích và lựa chọn tham số dựa trên các số liệu đo được-Lựa chọn thuật toán cho mô hình đơn nhiệm-Lựa chọn thuật toán cho mô hình đa nhiệm	Hoàn thành
Tuần 3	<ul style="list-style-type: none">-Lập trình bài toán theo mô hình đơn nhiệm-Lập trình bài toán theo mô hình đa nhiệm	<ul style="list-style-type: none">-Lập trình mô hình đơn nhiệm theo 2 phương án-Lập trình mô hình đa nhiệm theo 2 phương án	Hoàn thành

	<ul style="list-style-type: none"> -Tạo Client để hiện thị các kết quả truyền được từ mạch -Lập trình cho phép hệ thống nhận lệnh điều khiển từ máy tính 	<ul style="list-style-type: none"> -Lập trình giao diện hiện thị kết quả và điều khiển lệnh từ máy tính truy cập hệ thống 	
Tuần 4	<ul style="list-style-type: none"> -Kiểm tra hệ thống để đảm bảo hoạt động chính xác và ổn định -Thực hiện kiểm tra các thao tác từ máy tính đến hệ thống -Viết báo cáo về bài tập lớn 	<ul style="list-style-type: none"> -Kiểm tra hoạt động của các task đảm bảo sự ổn định và chính xác -Viết báo cáo và đánh giá giao diện điều hành -Đánh giá kết quả 	Hoàn thành

Phân chia công việc

Thành viên	Nhiệm vụ	Thời gian thực hiện	Chấm điểm
Chung	Lựa chọn C,D,T Lựa chọn các mô hình lập lịch Viết báo cáo Tính toán lại kết quả các mô hình lập lịch		
Lê Văn Vinh	- Hàn mạch	Tuần 1	9.5/10
	- Tổng hợp datasheet	Tuần 2	
	- Tạo thư viện BMP	Tuần 3	
	- Viết 2 mô hình đơn nhiệm main loop model và Simple periodic time -triggered scheduler	Tuần 4	
Lê Văn Công	- Phân chia công việc	Tuần 1	9.5/10
	- Debug mạch	Tuần 2	
	- Tạo thư viện LCD	Tuần 3	
	- Viết mô hình đa nhiệm Deadline-driven scheduling	Tuần 4	
Trần Văn Bắc	- Tạo form github (folder...)	Tuần 1	9.5/10
	- Tạo Client để hiển thị các kết quả truyền được từ mạch	Tuần 2	
	- Tạo thư viện MPR121 và funtion	Tuần 3	
	- Viết mô hình đa nhiệm Fixed-priority scheduling	Tuần 4	

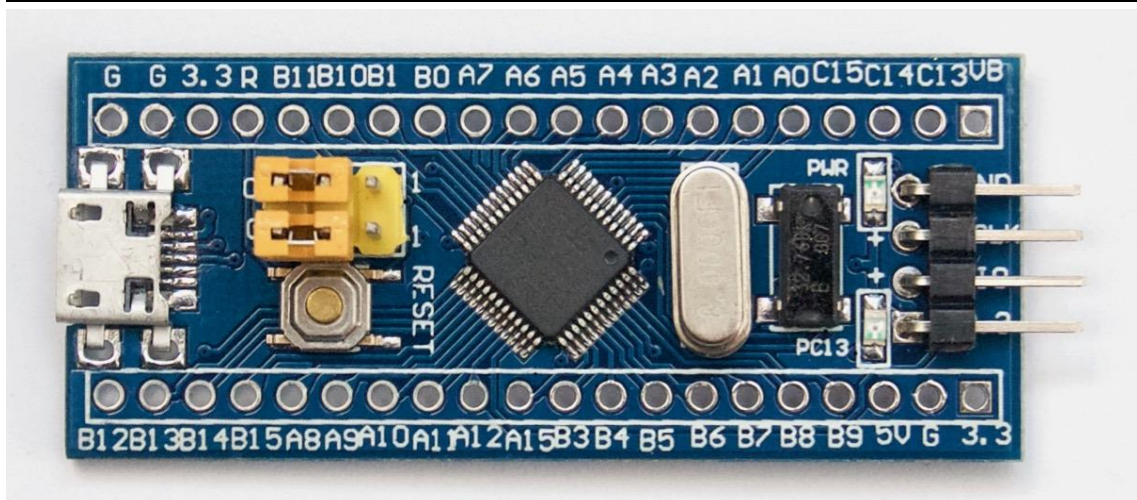
I. Thành phần mạch phân cứng

1. Bo mạch vi điều khiển STM32F103C8T6

Dòng vi điều khiển STM32F103C8T6 thuộc series STM32 của hãng STMicroelectronics, là một trong những dòng vi điều khiển phổ biến nhờ hiệu năng cao, độ tin cậy và giá thành hợp lý. Nó được sử dụng rộng rãi trong các ứng dụng nhúng, từ thiết bị IoT đến hệ thống công nghiệp. Bo mạch có thể chịu đựng trong môi trường nhiệt độ khoảng $-40 \rightarrow 105^{\circ}\text{C}$ nên được ứng dụng nhiều trong công nghiệp.

STM32F103C8T6 hỗ trợ nhiều cách thức giao tiếp như UART, I2C, SPI ... giúp người dùng dễ dàng triển khai các phương án trên các hệ thống IoT. Thông số của dòng STM32F103C8T6 được liệt kê dưới đây :

Lõi vi điều khiển	ARM 32-bit Cortex-M3
Điện áp đầu vào	2.0-3.6V
Bộ nhớ Flash	64Kb
SRAM	20Bb
Tần số Clock max	72MHz
GPIO	32
LED	PC13
ADC	2 bộ ADC 12 bit có 10 kênh
Giao tiếp	CAN, UART, SPI, USB, I2C,
Timer	3 timer 16 bit
Chế độ hỗ trợ	Input capture,output compare,PWM....



2. Cảm biến đo nhiệt độ, áp suất BMP180

a. Giới thiệu

BMP180 là phiên bản kế thừa tương thích chức năng của BMP085, thuộc thế hệ cảm biến áp suất kỹ thuật số độ chính xác cao mới dành cho các ứng dụng tiêu dùng.

Với điện tử tiêu thụ điện năng cực thấp và điện áp thấp, BMP180 được tối ưu hóa để sử dụng trong điện thoại di động, PDA, thiết bị định vị GPS và các thiết bị ngoài trời. Với độ nhiễu áp suất thấp ở độ cao chỉ 0,25m trong thời gian chuyển đổi nhanh, BMP180 mang lại hiệu suất vượt trội. Giao diện I²C cho phép dễ dàng tích hợp hệ thống với vi điều khiển. BMP180 dựa trên công nghệ áp điện (piezo-resistive), đảm bảo độ bền EMC, độ chính xác cao và tính ổn định lâu dài.

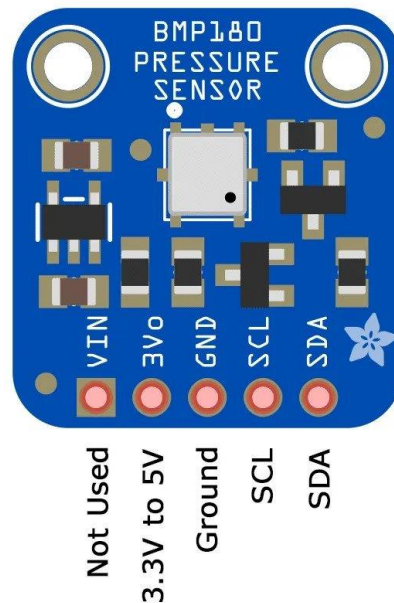
b. Nguyên lý hoạt động và giao tiếp

BMP180 tích hợp một cảm biến nhiệt bán dẫn bên trong. Cảm biến nhiệt bán dẫn dựa trên sự thay đổi điện áp hoặc điện trở khi nhiệt độ thay đổi. Giá trị này được chuyển đổi thành tín hiệu kỹ thuật số thông qua bộ chuyển đổi ADC (Analog-to-Digital Converter) bên trong BMP180. Mục đích để đo nhiệt độ môi trường bên ngoài và tính toán để bù cho phép đo áp suất bởi độ chính xác của nhiệt độ ảnh hưởng đến độ chính xác của áp suất.

BMP180 sử dụng một cảm biến áp suất dạng áp điện, phần tử áp điện được thiết kế dưới dạng màng mỏng (membrane) có tính đàn hồi. Khi áp suất thay đổi, màng này bị biến dạng, dẫn đến thay đổi điện trở của các phần tử áp điện. Sự thay đổi điện trở này được chuyển đổi thành tín hiệu điện tương ứng, sau đó được chuyển đổi thành dữ liệu kỹ thuật số bởi bộ ADC tích hợp.

Các giá trị đo được về nhiệt độ và độ ẩm là những giá trị thô, các giá trị thô này sẽ được tính toán về giá trị thực trên 1 thuật toán với các thông số hiệu chỉnh được lưu trữ bên trong E2PROM của cảm biến để tính toán chính xác.

Giao tiếp của cảm biến BMP180 với vi điều khiển sử dụng phương thức I²C.



Trên sơ đồ nhận thấy cảm biến có 2 chân cung cấp nguồn cho cảm biến hoạt động. Còn lại 2 chân gồm chân SCL và SDA trong đó SCL (Serial Clock) là đường xung nhịp nối tiếp, cung cấp xung nhịp đồng bộ cho việc truyền tải dữ liệu trên SDA. SCL điều khiển tốc độ truyền của dữ liệu giữa các thiết bị. Còn chân SDA (Serial Data) là đường dữ liệu nối tiếp, được sử dụng để truyền tải dữ liệu giữa các thiết bị trên bus I2C. Dữ liệu trên SDA được truyền theo chuỗi bit, một bit tại một thời điểm. Giao thức giao tiếp I2C có các điều kiện tín hiệu bus đặc biệt, bao gồm điều kiện bắt đầu, điều kiện dừng và các điều kiện dữ liệu nhị phân. Khi có điều kiện bắt đầu SCL ở mức cao và SDA chuyển từ mức cao xuống mức thấp. Sau đó, địa chỉ slave sẽ được gửi. Sau 7 bit địa chỉ, bit điều khiển hướng truyền dữ liệu (R/W) sẽ xác định chế độ đọc hoặc ghi. Nếu một thiết bị slave nhận ra rằng nó đang được địa chỉ hóa, nó phải gửi tín hiệu xác nhận (ACK) bằng cách kéo SDA xuống thấp trong chu kỳ SCL thứ 9 (chu kỳ ACK). Khi có điều kiện dừng SCL cũng ở mức cao, nhưng SDA chuyển từ mức thấp lên mức cao

c. Hoạt động và độ chính xác

BMP180 hỗ trợ bốn chế độ hoạt động, mỗi chế độ cân bằng giữa độ phân giải, thời gian chuyển đổi và mức tiêu thụ năng lượng:

- Ultra Low Power (Siêu tiết kiệm năng lượng).
- Standard (Tiêu chuẩn).
- High Resolution (Độ phân giải cao).
- Ultra High Resolution (Độ phân giải siêu cao).

Chế độ này được chọn bằng cách đặt biến `oversampling_setting` trong mã C với các giá trị:

- 0: Ultra Low Power.
- 1: Standard.

- 2: High Resolution.
- 3: Ultra High Resolution.

Điện áp hoạt động : 1.8-3.6V

Dải nhiệt độ đo được: 0-65°C

Dải đo áp suất:300-1000hPa

Độ chính xác đo nhiệt độ : $\pm 0.5^{\circ}\text{C}$

Độ chính xác đo áp suất : $\pm 1\text{hPa}$

3. Cảm biến điện dung MPR121

a. Giới thiệu

MPR121 là một cảm biến đầu vào cảm ứng điện dung được sản xuất bởi NXP Semiconductors. Dòng cảm biến này chủ yếu được sử dụng để phát hiện các tín hiệu cảm ứng từ các ngón tay người, nhờ vào công nghệ cảm biến điện dung. MPR121 có thể được sử dụng để phát triển các giao diện người-máy (HMI), bàn phím cảm ứng, hoặc các ứng dụng khác yêu cầu phát hiện cảm ứng.

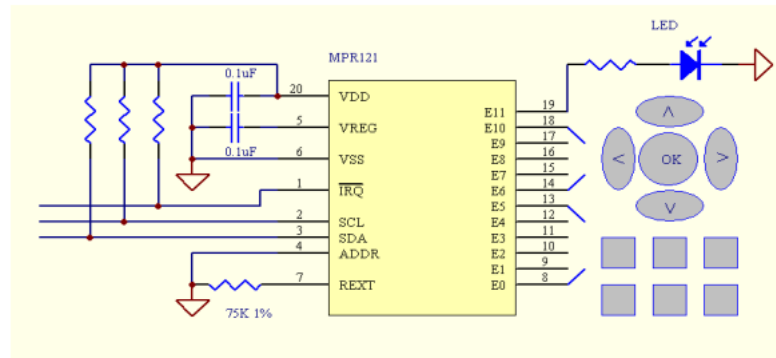
b. Nguyên lý hoạt động và giao tiếp

MPR121 sử dụng 12 điện cực cảm ứng (chạm hoặc không chạm) để phát hiện sự thay đổi của điện dung. Các điện cực này được kết nối với các ngón tay hoặc các bộ phận cơ thể người để nhận diện tín hiệu. Hoạt động của cảm biến dựa trên nguyên lý thay đổi điện dung : khi một vật thể có điện tích, chẳng hạn như ngón tay, tiếp xúc hoặc gần các điện cực, nó sẽ thay đổi điện dung của các điện cực đó. Điều này là vì cơ thể con người có thể tạo ra một dòng điện nhỏ do tính dẫn điện của cơ thể, làm thay đổi điện trường xung quanh các điện cực. Sự thay đổi điện dung này là rất nhỏ, nhưng cảm biến MPR121 có thể phát hiện sự thay đổi này với độ nhạy cao.

Tương tự với cảm biến BMP180, dòng MPR121 cũng sử dụng giao thức I2C để trao đổi dữ liệu với vi xử lý. Khi một thay đổi điện dung được phát hiện (ví dụ khi có ngón tay chạm vào một điện cực), MPR121 sẽ thông qua giao tiếp I2C gửi tín hiệu này tới vi điều khiển, cho phép hệ thống biết rằng một hành động cảm ứng đã xảy ra.

MPR121 tuân theo các tín hiệu giao tiếp chuẩn của I2C khi có các điều kiện bắt đầu và điều kiện kết thúc. Khi có điều kiện bắt đầu, bộ điều khiển kéo chân SDA xuống thấp trong khi chân SCL ở mức cao. Sau đó bộ điều khiển gửi địa chỉ 7 bit của MPR121 kèm theo bit R/W để xác định đọc hoặc ghi. Nếu MPR121 nhận ra địa chỉ của mình, nó sẽ gửi tín hiệu xác nhận (ACK) bằng cách kéo chân SDA xuống thấp trong chu kỳ đồng hồ thứ 9, khi đó bộ

điều khiển có thể gửi dữ liệu cấu hình hoặc đọc dữ liệu từ các thanh ghi của MPR121. Khi có điều kiện kết thúc, bộ điều khiển kéo chân SDA lên cao trong khi chân SCL ở mức cao.



c. Hoạt động

Dải nhiệt độ hoạt động : -40°C-85°C

Điện áp hoạt động : 1.71V -3.6V

4. Màn hình LCD PCF8574

a. Giới thiệu

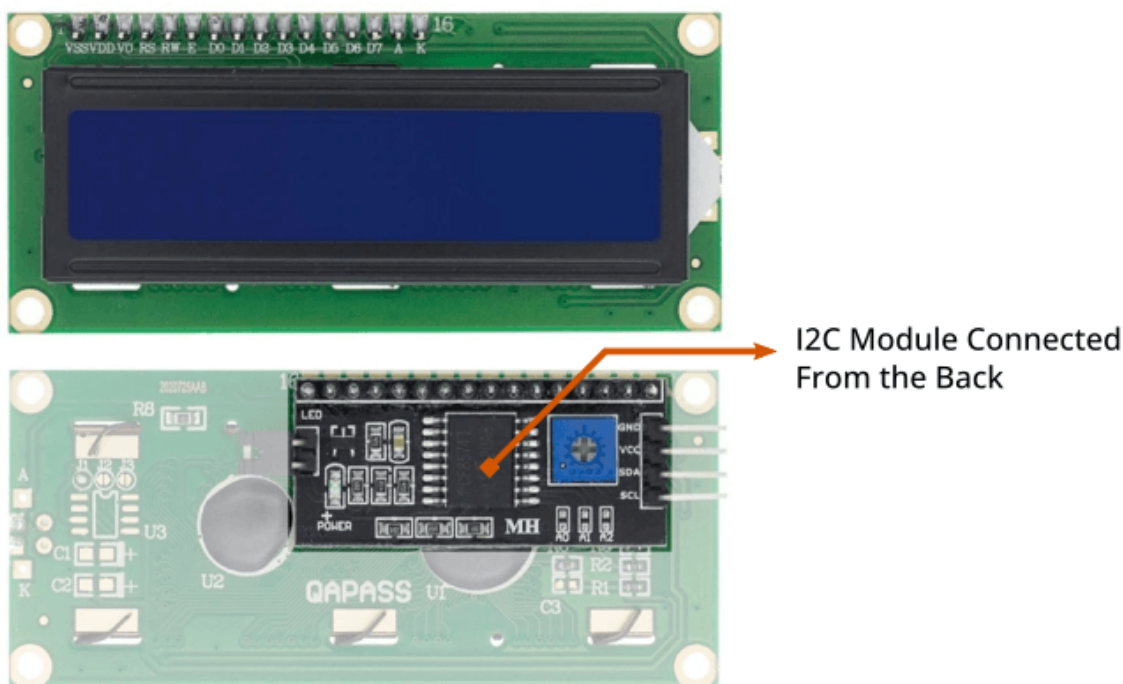
Màn hình LCD PCF8574 là một loại màn hình LCD sử dụng chip mở rộng giao tiếp I2C PCF8574 để điều khiển, giúp giảm số lượng chân điều khiển cần thiết so với việc kết nối trực tiếp với vi điều khiển. IC mở rộng PCF8574 giao tiếp qua giao thức I2C, nó cung cấp 8 chân I/O, cho phép kết nối dễ dàng với màn hình LCD 16x2.

b. Nguyên lý hoạt động và giao tiếp

Hoạt động của LCD dựa trên sự kết hợp giữa 2 thành phần chính bao gồm màn hình LCD và IC PCF8574. Màn hình LCD hoạt động dựa trên nguyên tắc hiển thị ký tự bằng cách điều khiển các ma trận điểm hoặc các phân đoạn đã định sẵn. PCF8574 là một bộ mở rộng GPIO cung cấp 8 chân I/O (P0-P7), cho phép kết nối với các chân điều khiển của màn hình LCD, bao gồm:

- RS (Register Select): Chọn chế độ hoạt động của LCD (lệnh hoặc dữ liệu).
- EN (Enable): Kích hoạt LCD để nhận dữ liệu.
- D4-D7: Dữ liệu 4-bit gửi đến LCD (hoạt động ở chế độ 4-bit để tiết kiệm chân kết nối).

PCF8574 nhận dữ liệu từ vi điều khiển qua I2C, sau đó xuất tín hiệu tương ứng qua các chân I/O để điều khiển màn hình LCD. Lệnh hoặc dữ liệu được xử lý bởi bộ điều khiển tích hợp trên LCD (thường là HD44780 hoặc tương tự) để hiển thị ký tự lên màn hình.



Với việc sử dụng giao thức I2C như các cảm biến trên thì đầu ra của LCD vẫn bao gồm 4 chân chính Vcc, GND, SDA, SCL trong đó Vcc và GND được dùng để cấp nguồn cho LCD và 2 chân SDA và SCL được dùng để truyền dữ liệu và cung cấp xung nhịp giữa LCD và vi điều khiển.

Giao tiếp I2C với thiết bị này được bắt đầu bởi bộ điều khiển bằng cách gửi một điều kiện bắt đầu, tức là chuyển mức từ cao xuống thấp trên chân SDA trong khi tín hiệu SCL đang ở mức cao.

Sau điều kiện bắt đầu, một byte địa chỉ thiết bị được gửi đi, bắt đầu từ bit có trọng số cao nhất (MSB) và bao gồm cả bit xác định hướng dữ liệu (R/W). Thiết bị này không phản hồi với địa chỉ gọi chung. Sau khi nhận được byte địa chỉ hợp lệ, thiết bị phản hồi bằng tín hiệu xác nhận, tức là kéo mức thấp trên chân SDA trong khoảng thời gian tín hiệu SCL ở mức cao liên quan đến xung đồng hồ xác nhận.

Các chân địa chỉ (A0–A2) của thiết bị không được thay đổi trong khoảng thời gian từ điều kiện bắt đầu đến điều kiện dừng.

c. Hoạt động truyền và nhận dữ liệu

Vi điều khiển gửi các lệnh hoặc dữ liệu hiển thị qua PCF8574 đến LCD bằng giao thức I2C sau đó PCF8574 nhận dữ liệu và chuyển đổi dữ liệu thành dạng song song để điều khiển LCD

Khi bộ điều khiển chính muốn gửi dữ liệu tới PCF8574, bộ điều khiển chính kéo mức thấp trên SDA trong khi SCL ở mức cao. Sau đó bộ điều khiển chính gửi 7 bit địa chỉ của PCF8574 kèm theo bit điều khiển R/W = 0 (ghi), PCF8574 phản hồi bằng tín hiệu ACK (kéo SDA xuống thấp). Sau đó bộ điều

khởi phát gửi 8 bit dữ liệu, PCF8574 xác nhận bằng tín hiệu ACK sau khi nhận đủ 1 byte. Cuối cùng, bộ điều khiển khởi phát kết thúc giao tiếp bằng cách kéo mức cao SDA trong khi SCL ở mức cao.

Khi bộ điều khiển khởi phát muốn nhận dữ liệu từ PCF8574, bộ điều khiển khởi phát kéo mức thấp trên SDA trong khi SCL ở mức cao. bộ điều khiển khởi phát gửi 7 bit địa chỉ của PCF8574 kèm theo bit điều khiển R/W = 1 (đọc). PCF8574 phản hồi bằng tín hiệu ACK sau đó PCF8574 xuất dữ liệu trạng thái các chân I/O (8 bit) lên bus SDA. Bộ điều khiển khởi phát đọc dữ liệu và phản hồi ACK nếu cần đọc thêm byte hoặc NACK nếu không cần đọc thêm. Bộ điều khiển khởi phát kết thúc giao tiếp bằng cách kéo mức cao SDA trong khi SCL ở mức cao.

Điện áp hoạt động :2.5-6V

5. Module **UART TTL to RS485 V2**

a. Giới thiệu

RS485 là một tiêu chuẩn giao tiếp nối tiếp được phát triển bởi EIA/TIA (Electronic Industries Alliance/Telecommunications Industry Association). Đây là giao thức phổ biến trong các hệ thống truyền thông công nghiệp, nhúng, và điều khiển tự động nhờ vào khả năng truyền dữ liệu ổn định trên khoảng cách xa và trong môi trường nhiễu cao

b. Hoạt động và kết nối

RS485 sử dụng tín hiệu vi sai (differential signaling) để truyền dữ liệu giữa các thiết bị. Điều này giúp giảm nhiễu và tăng độ ổn định của tín hiệu khi truyền trên khoảng cách xa. RS485 định nghĩa các đặc tính điện cho giao tiếp hai dây, bán song công, và đa điểm bằng cách sử dụng cặp xoắn đôi, trong đó dây A là dây mang tín hiệu dương và dây B là dây mang tín hiệu âm. Tín hiệu được truyền đi dưới dạng vi sai có nghĩa khi $A > B$ thì có mức logic 1 và khi $A < B$ thì có mức logic 0. Khi có tác động của nhiễu thì cả 2 dây mang điện áp dương và âm đều bị tác động nhưng do chỉ quan tâm về mức logic chênh lệch giữa A và B nên giảm tối thiểu tác động của nhiễu lên việc truyền dữ liệu. Ngoài ra RS485 cho phép hỗ trợ tối đa 32 thiết bị truyền thông trên cùng một đường truyền (bus) mà không cần phần cứng phức tạp. Tốc độ truyền tin RS485 có thể truyền dữ liệu lên đến 1000 mét với tốc độ tối đa khoảng 100 kbps.

Module TTL to RS485 là một thiết bị giúp chuyển đổi tín hiệu nối tiếp TTL (Transistor-Transistor Logic) của vi điều khiển (như Arduino, STM32, Raspberry Pi) sang tín hiệu RS485. Nó cho phép các thiết bị sử dụng giao tiếp nối tiếp thông thường giao tiếp trên chuẩn RS485, hỗ trợ truyền dữ liệu ổn định trên khoảng cách xa và trong môi trường nhiễu cao.

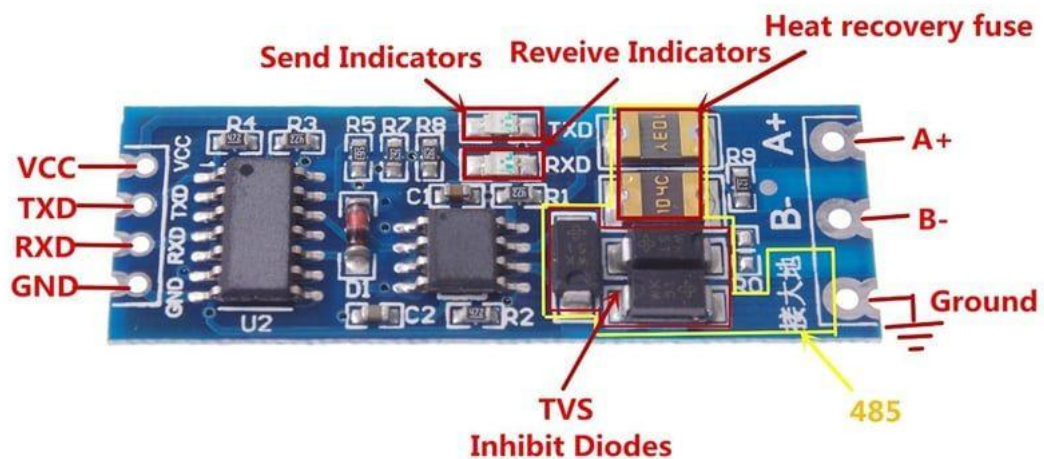
Module giao tiếp TTL to RS485 bao gồm IC chuyển đổi RS485, chân giao tiếp TTL để nhận lệnh từ vi điều khiển và 2 chân giao tiếp RS485 1 chân mang tín hiệu dương(A) và 1 chân mang tín hiệu âm(B).

Hoạt động của module dựa trên 2 chế độ:

+Chuyển đổi tín hiệu TTL sang RS485: khi đó tín hiệu từ chân TX của vi điều khiển được gửi tới IC chuyển đổi RS485 sau đó IC chuyển đổi tín hiệu TTL thành tín hiệu vi sai trên hai dây A và B của RS485.

+ Chuyển đổi tín hiệu RS485 sang TTL: tín hiệu vi sai từ dây A và B được IC chuyển đổi RS485 xử lý và xuất ra tín hiệu TTL tại chân RX để gửi về vi điều khiển.

Trong dự án RS485 được sử dụng như 1 thiết bị trung gian để tăng cường kết nối UART giữa hệ thống và máy tính. Sử dụng 2 RS485 để làm 2 đường truyền và nhận khác nhau tránh sự xung đột trên cùng 1 đường truyền. Phiên bản sử dụng trong dự án V2 được thiết kế với khả năng chống nhiễu cao, tích hợp các bộ đệm, Cầu chì tự phục hồi, Diod chống nhiễu giúp hệ thống chạy ổn định, an toàn hơn và không làm cháy board điều khiển trung tâm. Phần chân giao tiếp RS485 trên mạch có chân Mass, nếu hệ thống có đường dây mass tiếp đất thì có thể sử dụng để nối vào chân Mass này giúp tăng khả năng chống nhiễu và chống sét.



c.Điều kiện hoạt động

Điện áp hoạt động :3-5VDC

Điện áp giao tiếp:3-5VDC

Khoảng cách truyền nhận tối đa 1000m (nên ưu tiên dưới 800m để đảm bảo kết nối)

Có đèn led thông báo trạng thái truyền nhận RX và TX.

6.Module UART CH340

a.Giới thiệu

Module USB to UART CH340 là một thiết bị chuyển đổi tín hiệu USB sang tín hiệu UART, giúp các máy tính hoặc thiết bị có cổng USB giao tiếp với vi điều khiển, module nhúng hoặc các thiết bị nối tiếp sử dụng chuẩn UART.

IC chính của module là CH340, một dòng IC chuyển đổi USB-UART được thiết kế bởi hãng WCH, với ưu điểm giá thành rẻ, dễ sử dụng và hỗ trợ trên nhiều hệ điều hành.

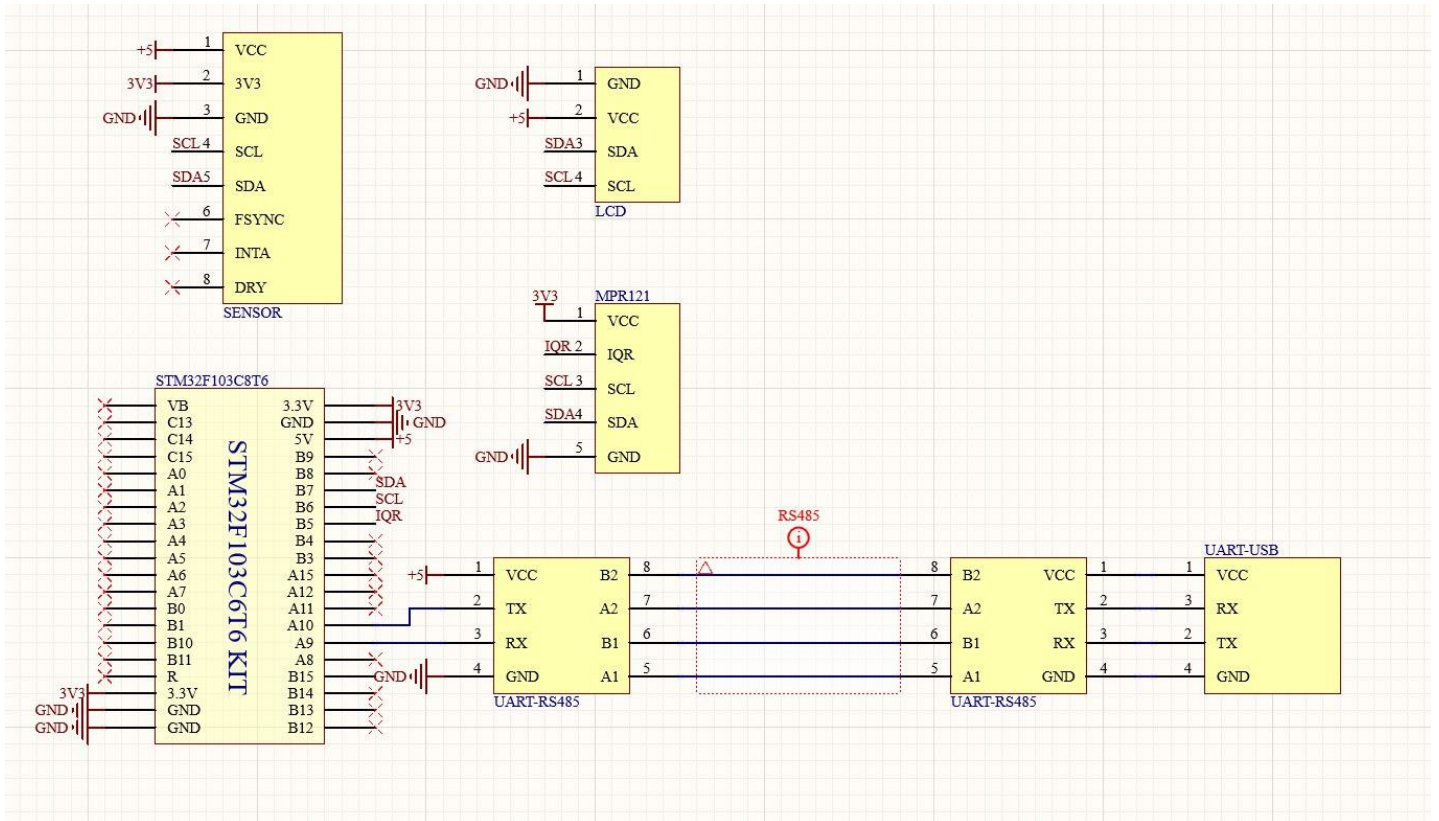
b.Hoạt động và kết nối

Module USB to UART CH340 có cấu tạo bao gồm IC CH340 đóng vai trò thực hiện việc chuyển đổi chính giữa UART và USB, cổng USB và chân giao tiếp UART. Trong đó chân giao tiếp UART gồm chân TXD đóng vai trò chân truyền tín hiệu, RXD đóng vai trò chân nhận tín hiệu, chân Vcc và GND đóng vai trò cung cấp nguồn và áp. Chức năng chính của mạch nhằm kết nối thiết bị nhúng và máy tính. Cổng USB được kết nối với cổng USB của máy tính và máy tính sẽ nhận module như 1 cổng COM ảo. Chân giao tiếp UART được kết nối với modul RS485 tăng cường khả năng truyền tin.



Điện áp hoạt động: module có thể hoạt động ở 2 mức điện áp là 3.3V và 5V

7. Sơ đồ kết nối chân



II. Thiết kế phần mềm

1. Xác định các Task và các tham số

Mạch đo có các chức năng gồm : đo nhiệt độ, đo áp suất, đếm xung, truyền UART, truyền I2C lên LCD

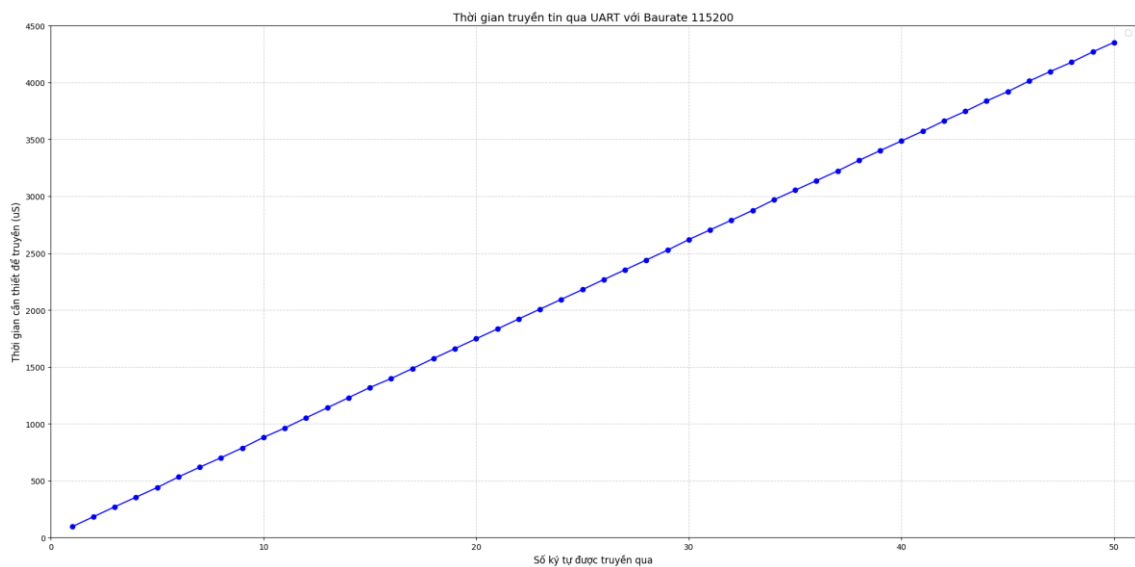
Các Task được phân chia bao gồm

- +Đo nhiệt độ
- +Đo áp suất
- +Hiện thị LCD
- +UART ở chế độ TX
- +UART ở chế độ RX
- +Đếm xung bằng ngắt

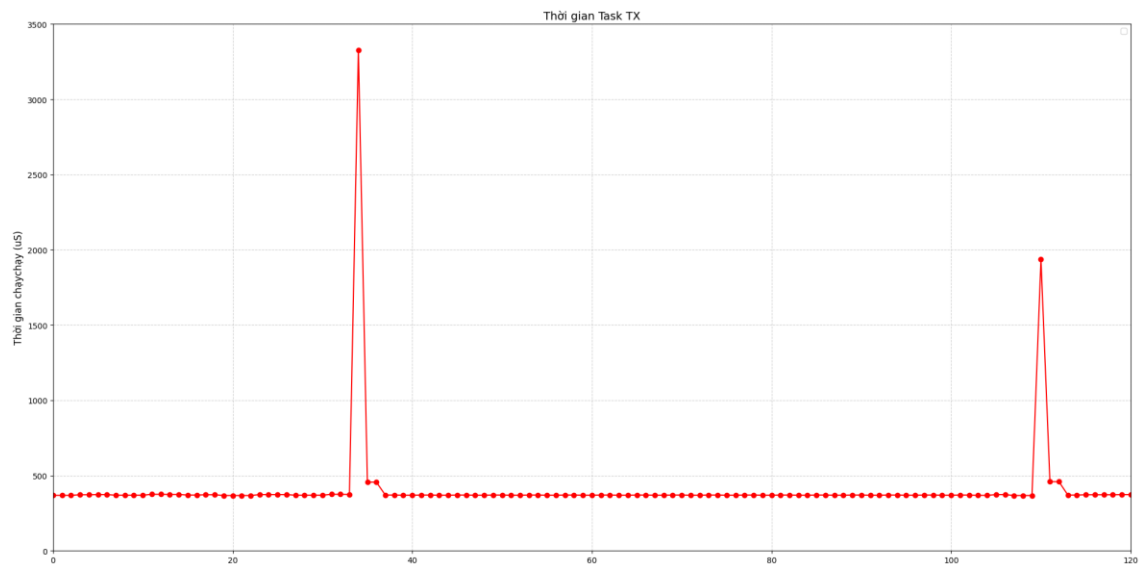
Để xác định các tham số cần thiết cho các Task cần tiến hành đo thời gian thực hiện các Task

Dưới đây là hình ảnh đo thời gian thực hiện của các Task

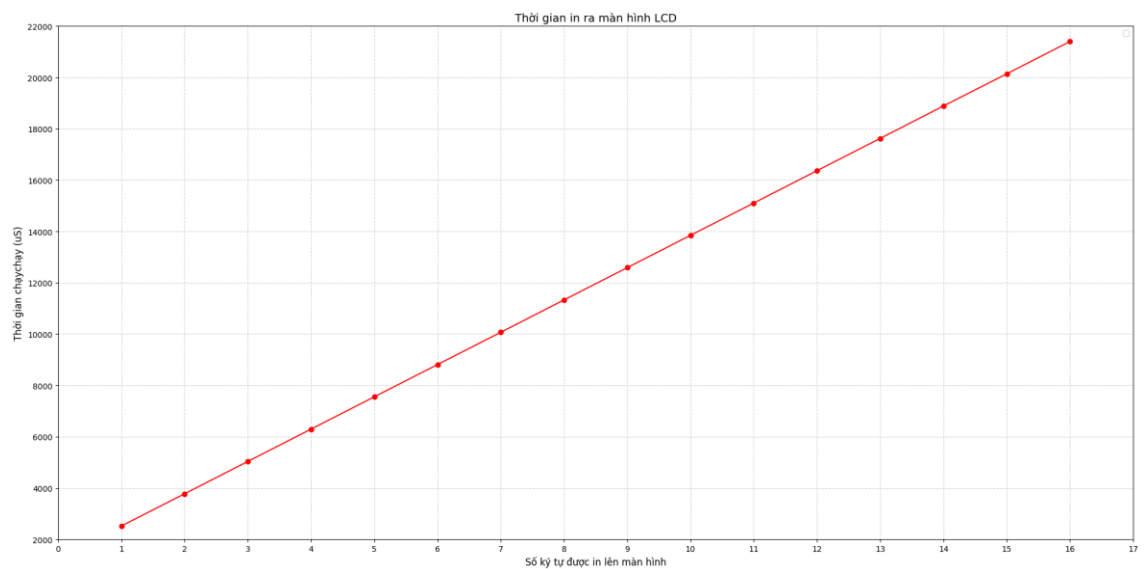
-UART ở chế độ TX với max là 50 ký tự



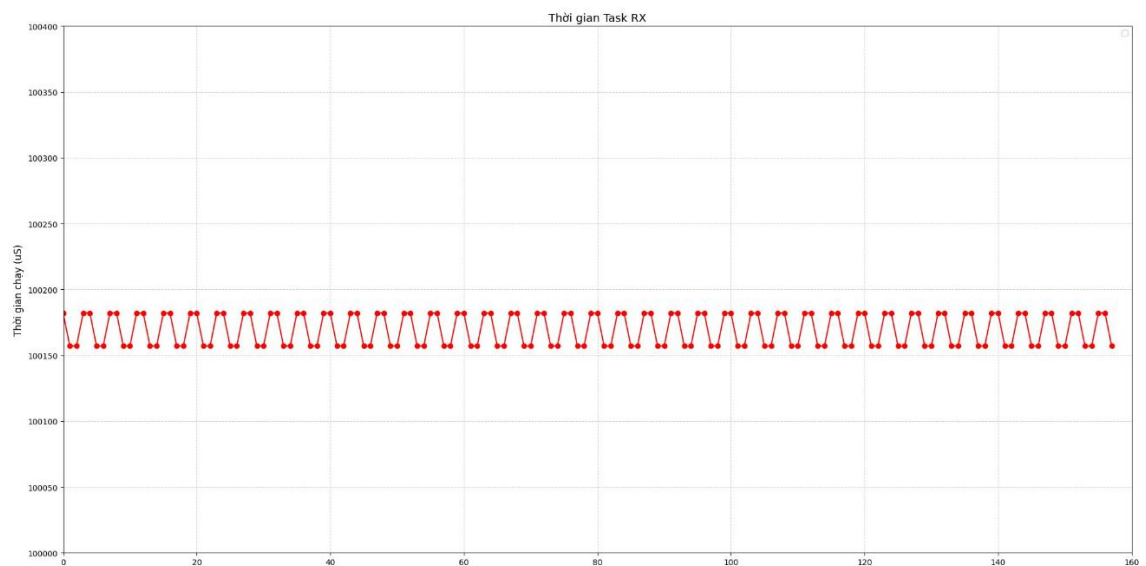
-Thời gian chạy ở chế độ TX của UART



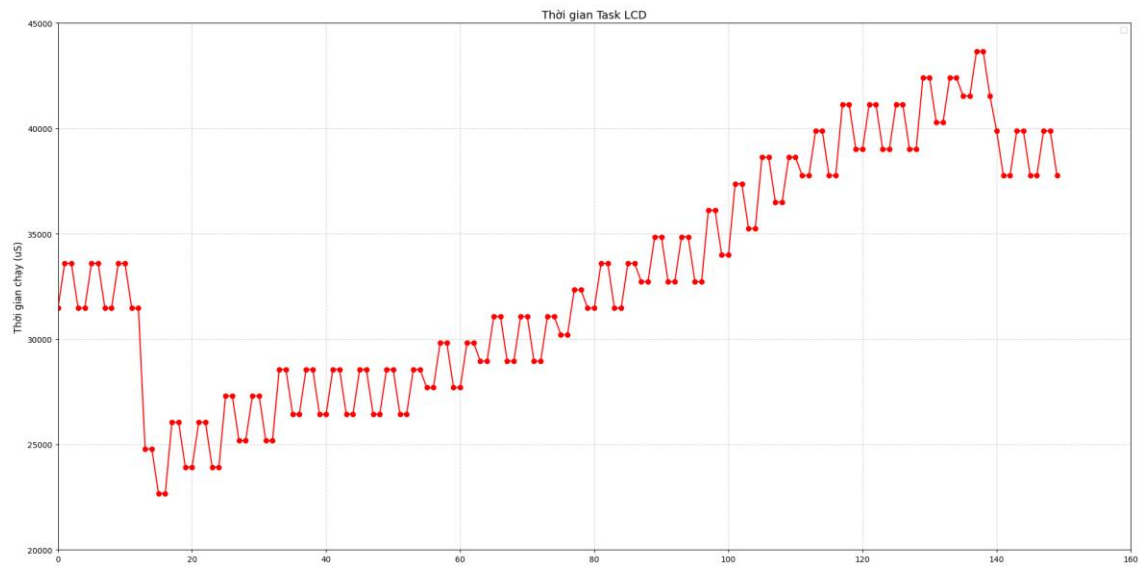
-In ra màn hình LCD các ký tự riêng lẻ



- Thời gian chạy ở chế độ RX của UART



-Chạy task LCD



Bảng số liệu tổng hợp kết quả và lựa chọn tham số

Task	Thời gian thực thi	Tham số C	Tham số D	Tham số T
Hiển thị LCD	43ms	50ms	100ms	1000ms
Đo nhiệt độ	6550 μ s	7ms	5000ms	60s(*)
Đo áp suất	35185 μ s	36ms	5000ms	60s(*)
UART TX	4353 μ s	10ms	100ms	1000ms
UART RX	500900 μ s	100ms	150ms	150ms
Key	735 μ s	1ms	100ms	200ms

-Giải thích lựa chọn tham số

+Tham số C là thời gian thực thi của task, là khoảng thời gian CPU cần để thực thi hoàn chỉnh một task mà không bị gián đoạn bởi các yếu tố bên ngoài. Các giá trị của tham số C được làm tròn nhằm dễ dàng cho việc tính toán xử lý.

+Tham số D là thời hạn hoàn thành của task, cũng chính là khoảng thời gian tối đa mà task phải hoàn thành kể từ khi nó được kích hoạt. Các giá trị về D thường xét theo điều kiện xấu nhất có thể xảy ra.

+Tham số T là chu kỳ của task, nó là khoảng thời gian kể từ kết thúc task trước đó đến khi nó sẵn sàng. Với nhiệt độ thì thời gian để nhiệt độ thay đổi cỡ 1 ° C là cỡ 60 phút(*). Tương tự với chu kỳ đo áp suất. Với chu kỳ của bàn phím thì tốc độ gõ phím cỡ nhanh của một người là 6 lần trên giây nên chu kỳ cỡ 200ms là con số hợp lý. Đối với TX, RX thì vừa phải phù hợp với tốc độ lấy mẫu của các ngoại vi vừa phải đáp ứng với tốc độ người dùng. Đối với LCD1602 thì việc để chu kỳ quá nhanh sẽ gây ra không hiển thị được, chu kỳ quá lâu thì ko phù hợp với tốc độ hiển thị cần thiết.

- Hiện thị LCD: Đây là tốc độ mà LCD có thể hiện thị ổn định
- Đo nhiệt độ : Lấy thời gian lấy mẫu trung bình của độ thay đổi nhiệt độ trong môi trường trong nhà
- Đo áp suất : Lấy thời gian lấy mẫu trung bình của độ thay đổi áp suất trong môi trường trong nhà
- UART TX ,UART RX : Đối với TX, RX thì vừa phải phù hợp với tốc độ lấy mẫu của các ngoại vi vừa phải đáp ứng với tốc độ người dùng
- Key: Tốc độ bấm phím có thể đáp ứng khoảng 5 lần /1s

(*) thời gian thay đổi nhiệt độ khoảng 60s nhưng đã trong code dự án đã được điều chỉnh về 5ms để tránh thời gian chờ quá lâu trong lúc chạy code.

2. Phân tích tính đáp ứng yếu tố thời gian thực

-Các mô hình cho lập trình đơn nhiệm

+Main loop model : mô hình lập trình dựa trên một vòng lặp chính liên tục kiểm tra sự và xử lý các sự kiện hoặc nhiệm vụ. Mỗi khi 1 task được hoàn thành thì 1 task khác sẽ được tiếp tục chạy. Các task hoạt động theo 1 thứ tự đã được cố định trước khi thỏa mãn điều kiện thì lập tức sẽ được chạy, không thể can thiệp để thay đổi thứ tự các task. Sự sắp xếp thứ tự của các task được phân chia sao cho hợp lý. Trong project, thứ tự các task được chia là

+Đếm xung

+Đọc nhiệt độ

+Đọc áp suất

+Hiển thị LCD

+Truyền UART

+Nhận UART

Cần thực hiện các task về đọc và đo để lấy dữ liệu cho các task hiển thị và truyền nhận UART thực thi.

Đánh giá yếu tố đáp ứng thời gian thực

Để đánh giá tính đáp ứng thời gian thực của các mô hình này cần đánh giá tổng CPU utilization.

$$M = \sum \frac{C_i}{T_i} = \frac{50}{1000} + \frac{7}{60 \times 10^3} + \frac{36}{60 \times 10^3} + \frac{10}{1000} + \frac{1}{200} + \frac{100}{150} = 0.7324167$$

Đối với 2 mô hình đơn nhiệm ban đầu thì chắc chắn sẽ đáp ứng được CPU utilization khi đưa ra tổng thời gian đủ lớn.

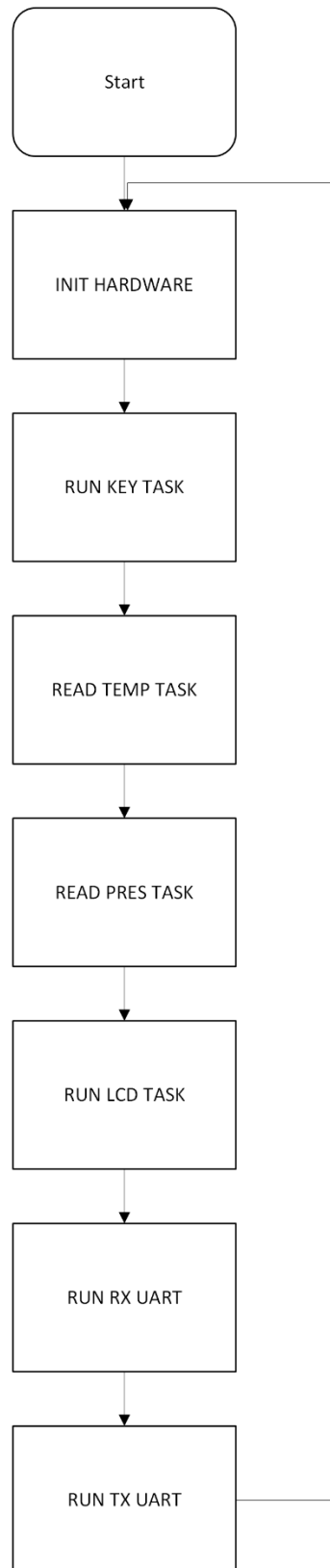
Đối với thuật toán RMS sẽ đáp ứng được vì :

$$M = 0.7324167 < 6 * (2^{\frac{1}{6}} - 1) \approx 0.73476$$

Đối với thuật toán EDF thì $M < 100\%$ là hoàn toàn thỏa mãn.

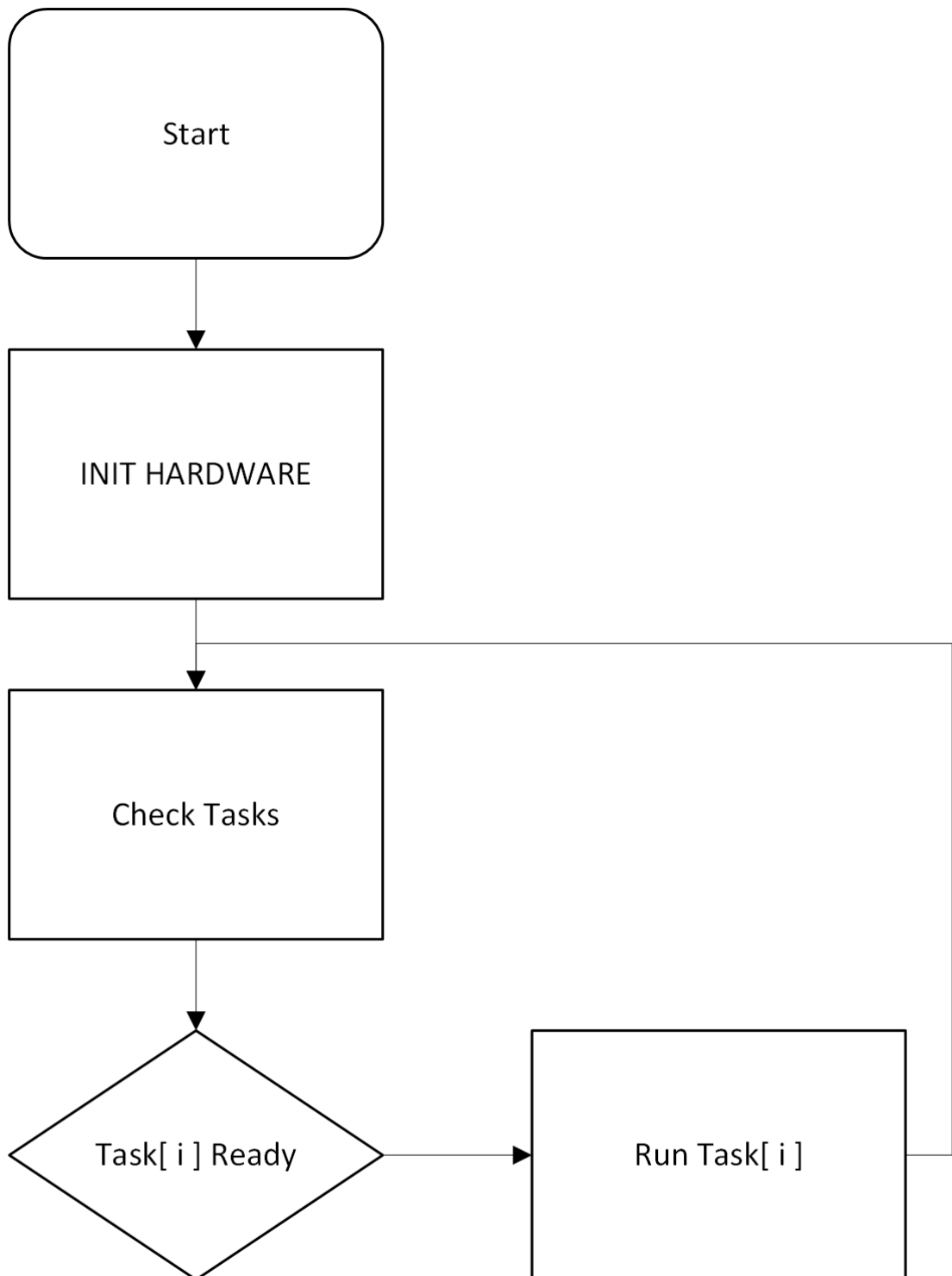
3. Thuật toán và lập trình

-Thuật toán cho mô hình main loop model: Chương trình chạy tuần tự các task, khi đến hết task cuối cùng thì quay lại ban đầu thực hiện tiếp.



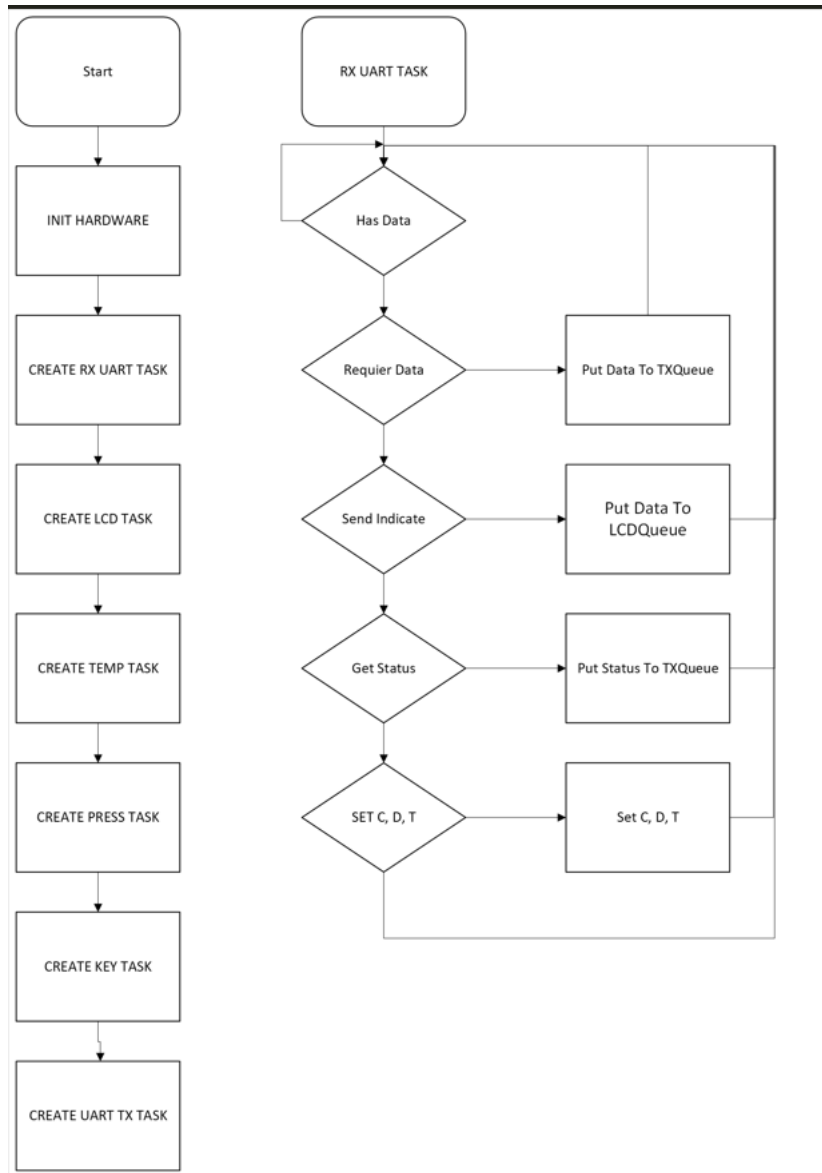
-Thuật toán cho mô hình Simple periodic time -triggered scheduler

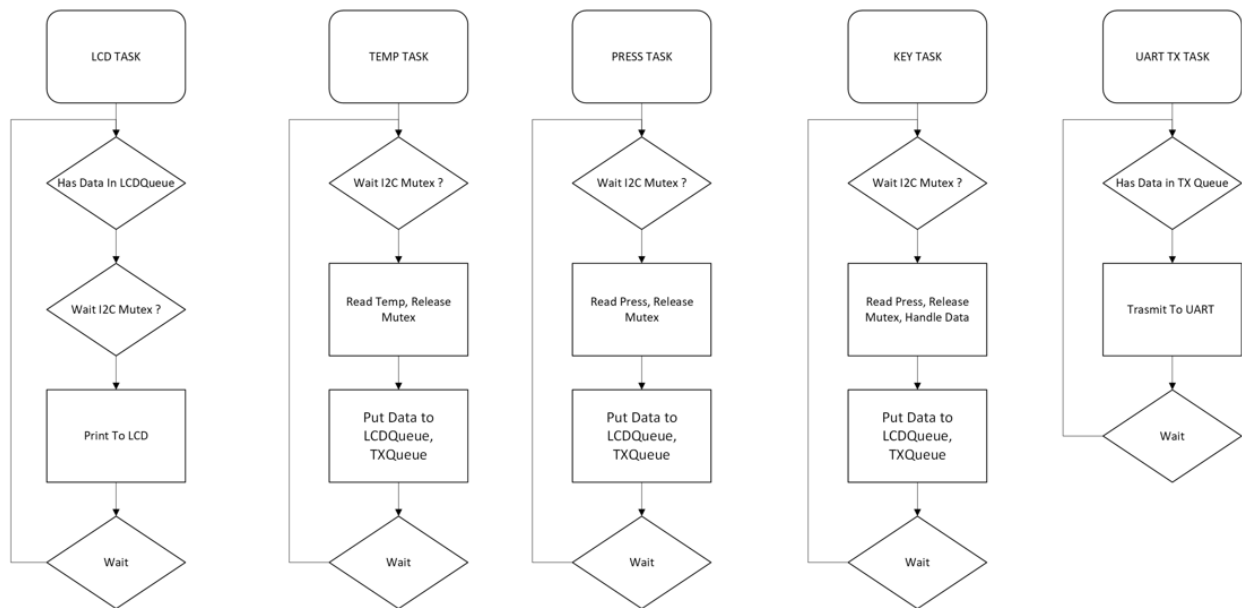
Chương trình xác định Task nào đang sẵn sàng sẽ được thực thi, Task nào sẵn sàng trước thì thực thi trước, nếu có 2 Task cùng sẵn sàng thì sẽ xét theo thứ tự cài sẵn.



-Thuật toán cho mô hình Fixed-priority scheduling:

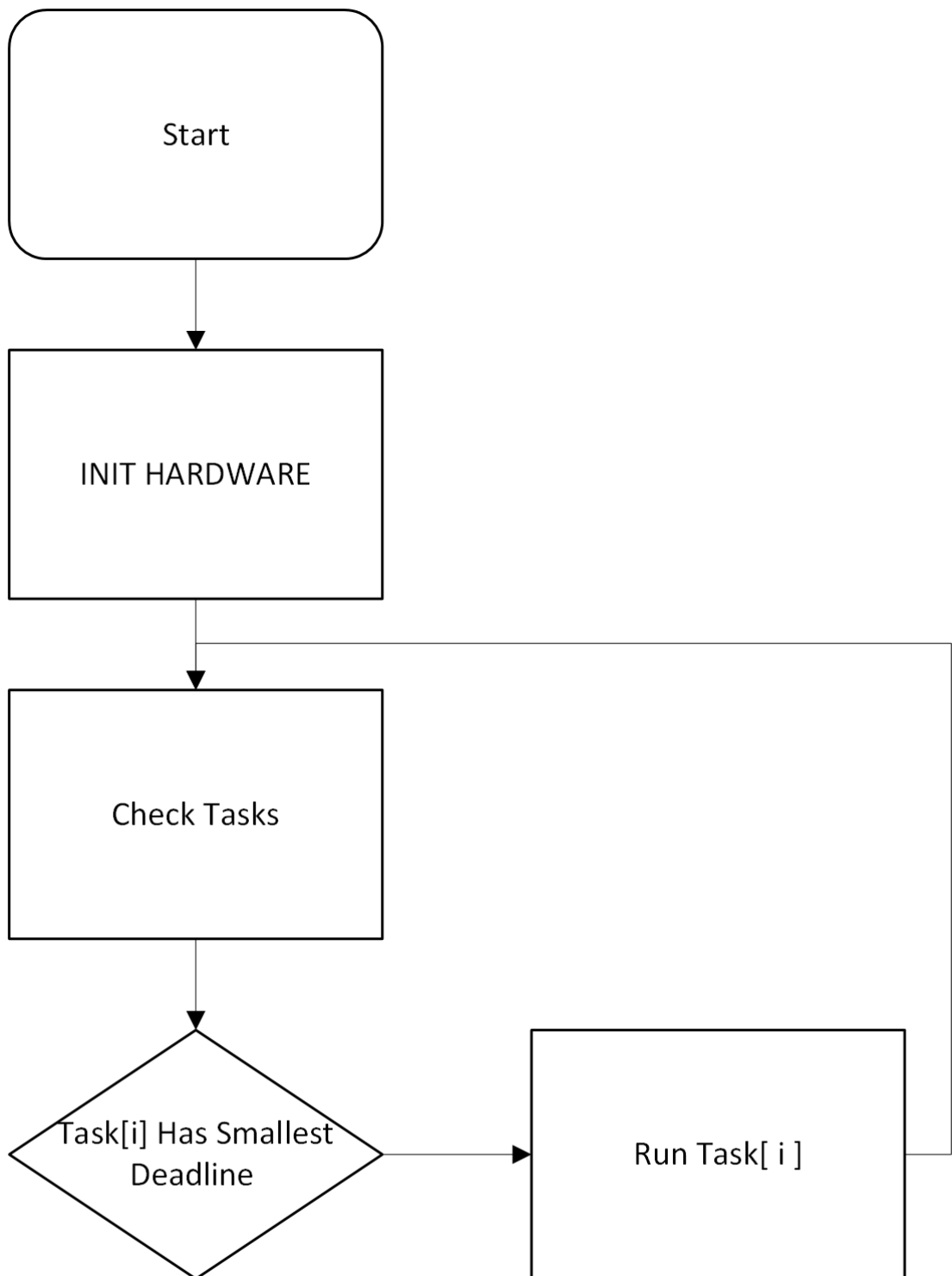
Chương trình chạy trên hệ điều hành Free-RTOS CMSIS, các Task được cài đặt mức ưu tiên cố định. Truyền nhận dữ liệu giữa các task sử dụng Queue, sử dụng Mutex để quản lý tài nguyên chung, ở đây là Bus I2C.





-Thuật toán cho mô hình Deadline-driven scheduling

Chương trình tính toán deadline cho các task sau mỗi khi có 1 task thực hiện xong. Task nào có Deadline sớm nhất sẽ ưu tiên được thực thi trước.



III. Kết quả đánh giá

1. Các kết quả đánh giá

- Các kết quả của project đạt được

+ Mạch hiển thị nhiệt độ và áp suất lên màn hình LCD cũng như hiển thị trên giao diện điều khiển

+ Giao diện điều khiển có khả năng truy cập sâu vào hệ thống cũng như thay đổi các tham số phức tạp như C, D, T của các task

+ Kết quả hiển thị chính xác của nhiệt độ và áp suất có độ chính xác khá cao với sai số của áp suất và nhiệt độ lần lượt khoảng $\pm 100Pa$ và $\pm 0.5^{\circ}C$

+ Mô hình đếm xung hoạt động ổn định có hiển thị số lần chạm, mỗi lần chạm vào ra được tính là 2 xung

+ Mạch phần cứng hoàn thiện chắc chắn, có độ bền cao, được tiết kiệm gọn gàng và dễ dàng kết nối

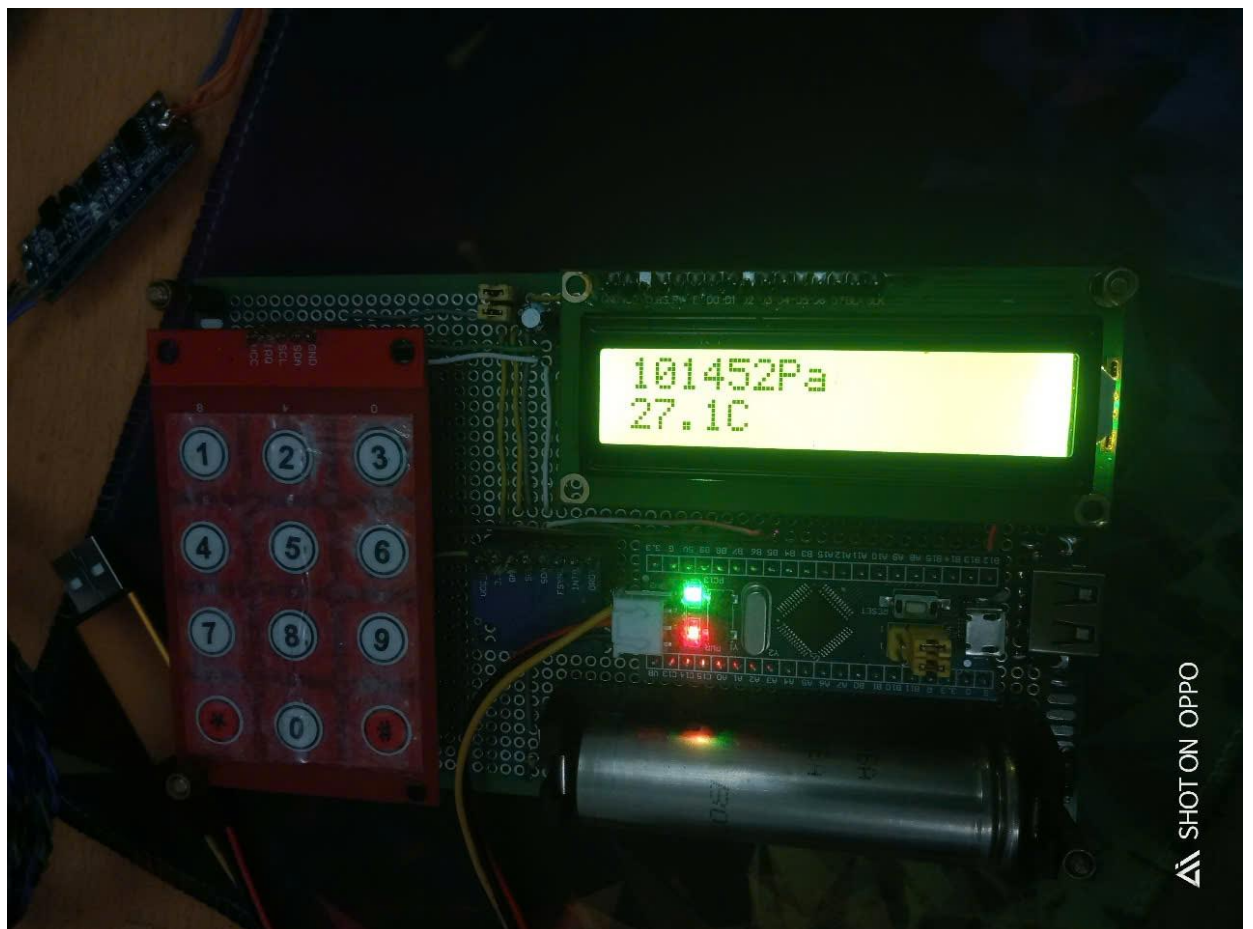
+ Thực hiện đầy đủ các yêu cầu của project về phần cứng cũng như phần mềm, đảm bảo hệ thống hoạt động ổn định và hiệu quả

+ Các mô hình đơn nhiệm và đa nhiệm được thực hiện trên những cơ sở thuật toán để tối ưu việc quản lý các task phù hợp với tài nguyên.

- Đánh giá kết quả đạt được

Với mục tiêu của môn học giúp sinh viên hiểu và thực thi triển khai hệ thống nhúng và lập trình hệ điều hành quản lý các tác vụ khác nhau, dự án đã hoàn thành các yêu cầu đặt ra về các phần về phần cứng và phần mềm cũng như phát triển và triển khai thêm các module phần cứng như đo thêm áp suất, có hệ thống bàn phím thực hiện đếm xung bằng ngắt cũng như triển khai giao diện web giúp mọi người dễ dàng tiếp cận và điều khiển giao diện 1 cách đơn giản. Giao diện web tích hợp đầy đủ các khả năng điều khiển từ máy tính giúp cho việc quản lý và điều hành hệ thống dễ dàng hơn. Từ đó có thể nhận thấy việc hoàn thiện 1 hệ thống nhúng không quá khó khăn với sinh viên cũng như có thể thêm các ý tưởng để cải thiện thêm cho hệ thống hoạt động tốt hơn

2. Hình ảnh mạch và hình ảnh giao diện web



RS485 UART Web Interface

Chọn cổng COM: COM7 Baud rate: 115200 Close
Data sent successfully!

Nhập địa chỉ: 111 Chọn mã hàm: Chính Task C D T Nhập dữ liệu: 1 Gửi dữ liệu

- Đọc dữ liệu
- Gửi thông báo
- Kiểm tra trạng thái
- Chính Task C D T
- Khởi phục C D T

Bảng Dữ Liệu Gửi và Nhận

Dữ liệu gửi						Dữ liệu nhận						
Time	Start	Address	Code	Data	CRC	Stop	Start	Address	Code	Data	CRC	Stop
22:03:31	3A	6F	04	31	FFF5	CRLF	3A	6F	05	496E76616C6964204320442054	65C5	CRLF

Bảng Dữ Liệu Cập Nhật

Dữ liệu nhận						Giải Mã		
Time	Start	Address	Code	Data	CRC	Stop	Thuộc Tính	Giá Trị
22:03:56	3A	6F	02	3130313435395061	E2F3	CRLF	Áp Suất	101459Pa
22:03:54	3A	6F	01	32362E3943	4642	CRLF	Nhiệt độ	26.9C
22:03:51	3A	6F	02	3130313434385061	B2CF	CRLF	Áp Suất	101448Pa
22:03:49	3A	6F	01	32362E3943	4642	CRLF	Nhiệt độ	26.9C
22:03:46	3A	6F	02	3130313435355061	22F0	CRLF	Áp Suất	101455Pa
22:03:44	3A	6F	01	32362E3943	4642	CRLF	Nhiệt độ	26.9C