|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI  **VIỆN ĐIỆN TỬ - VIỄN THÔNG**  logo_128  BÁO CÁO BÀI TẬP LỚN  **KỸ THUẬT VI XỬ LÝ**  **Đề tài: Máy kiểm tra thân nhiệt từ xa**  Giảng viên hướng dẫn : **TS.Hàn Huy Dũng**  Mã lớp : **124833**  Nhóm sinh viên thực hiện: **Nhóm 12**   |  |  |  | | --- | --- | --- | | **Quách Thị Dung** | **ĐT-07** | **20182440** | | **Trần Mạnh Kiên** | **ĐT-08** | **20182618** | | **Phạm Đức Thắng** | **ĐT-07** | **20182782** | | **Lê Thị Lan** | **ĐT-08** | **20182629** |   Hà Nội, 5 -2021 |

LỜI NÓI ĐẦU

Hiện tại dịch Covid-19 tiếp tục diễn biến phức tạp, nguy cơ lây lan cao.Vì vậy việc thiết kế tạo ra 1 máy kiểm tra đo thân nhiệt từ xa, cho kết quả chính xác và nhanh chóng là 1 nhu cầu thiết yếu. Sự phát triển của ngành điện tử ngày càng mạnh mẽ thì việc sử dụng vi điều khiển để thiết kế, chế tạo các thiết bị báo giờ là không khó. Các thiết bị báo giờ chỉ cần có tác động thay đổi một chút thì có thể sử dụng chúng ở nhiều nơi mong muốn. Chính vì vậy kết hợp với quá trình học tập môn “Kỹ thuật Vi xử lý” và dưới sự giảng dạy của thầy TS. Hàn Huy Dũng chúng em đã lựa chọn đề tài “Máy kiểm tra thân nhiệt từ xa” để nghiên cứu và hiểu sâu hơn về môn học, có thể áp dụng lý thuyết vào thực hành.

Nhóm chúng em xin bày tỏ lời cảm ơn sâu sắc đến TS. Hàn Huy Dũng đã hướng dẫn chỉ bảo để chúng em có thể hoàn thành tốt đề tài này. Chúng em mong có được những đóng góp, nhận xét của thầy giáo để có thể hoàn thiện hơn nữa.

Trong quá trình thực hiện đề tài này, dù rất cố gắng nhưng do vốn kiến thức cũng như các kỹ năng còn hạn hẹp nên không thể tránh khỏi những sai sót. Nhóm chúng em rất mong nhận được những đóng góp, nhận xét, chia sẻ của thầy để các sản phẩm tiếp theo của nhóm sẽ hoàn thiện hơn.

Chúng em chân thành cảm ơn!

MỤC LỤC

[LỜI NÓI ĐẦU i](#_Toc79788341)

[DANH MỤC HÌNH VẼ iv](#_Toc79788342)

[DANH MỤC BẢNG BIỂU v](#_Toc79788343)

[CHƯƠNG 1. TỔNG QUAN ĐỀ TÀI 1](#_Toc79788344)

[1.1 Đặt vấn đề 1](#_Toc79788345)

[1.2 Giới thiệu đề tài 1](#_Toc79788346)

[1.3 Phân công công việc 1](#_Toc79788347)

[CHƯƠNG 2. MÔ TẢ SẢN PHẨM 3](#_Toc79788348)

[2.1 Mục tiêu 3](#_Toc79788349)

[2.2 Yêu cầu chưc năng 3](#_Toc79788350)

[2.3 Yêu cầu phi chức năng 3](#_Toc79788351)

[CHƯƠNG 3. CƠ SỞ LÝ THUYẾT 4](#_Toc79788352)

[4.1 Vi điều khiển ESP32 4](#_Toc79788353)

[4.2 Cảm biến nhiệt độ LM35 6](#_Toc79788354)

[4.3 Còi buzzer 8](#_Toc79788355)

[4.4 LCD1602 8](#_Toc79788356)

[4.5 Mạch chuyển đổi I2C cho LCD 11](#_Toc79788357)

[CHƯƠNG 4. THIẾT KẾ MẠCH 13](#_Toc79788358)

[4.1 Xây dựng sơ đồ khối của mạch 13](#_Toc79788359)

[2.2 Nguyên lý hoạt động của mạch 13](#_Toc79788360)

[4.3 Lựa chọn linh kiện 14](#_Toc79788361)

[4.4 Xây dựng sơ đồ nguyên lý của mạch 15](#_Toc79788362)

[CHƯƠNG 5. SƠ ĐỒ MẠCH VÀ CODE HỆ THỐNG 16](#_Toc79788363)

[5.1 Sơ đồ mạch 16](#_Toc79788364)

[6.2 Công nghệ sử dụng 16](#_Toc79788365)

[CHƯƠNG 7. TIẾN HÀNH THỬ NGHIỆM, TEST LỖI VÀ KHẮC PHỤC HỆ THÔNG 18](#_Toc79788366)

[7.1 Tiến hành thử nghiệm 18](#_Toc79788367)

[7.2 Kiểm thử 18](#_Toc79788368)

[7.3 Test lỗi hệ thống 19](#_Toc79788369)

[KẾT LUẬN 20](#_Toc79788370)

[TÀI LIỆU THAM KHẢO 21](#_Toc79788371)

[PHỤ LỤC 22](#_Toc79788372)

# DANH MỤC HÌNH VẼ

[Hình 1 Vi điều khiển ESP32 4](#_Toc79788402)

[Hình 2 Cảm biến nhiệt độ LM35 6](file:///C:\Users\Bap%20Cai\Desktop\DuLieu\20202\VI%20XỬ%20LÝ\BTL\BaoCaoTest.docx#_Toc79788403)

[Hình 3 Sơ đồ chân của cảm biến LM35 7](#_Toc79788404)

[Hình 4 LCD 1602 9](#_Toc79788405)

[Hình 5 Sơ đồ chân tín hiệu LCD1602 9](#_Toc79788406)

[Hình 6 Mạch chuyển đổi I2C cho LCD 11](#_Toc79788407)

[Hình 7 Sơ đồ chân mạch chuyển đổi I2C 12](#_Toc79788408)

[Hình 8 Sơ đồ cách kết nối mạch chuyển đổi I2C 12](#_Toc79788409)

[Hình 9 Sơ đồ khối hệ thống 13](#_Toc79788410)

[Hình 10 Sơ đồ khối hệ thống với các linh kiện sử dụng 14](#_Toc79788411)

[Hình 11 Giao diện Arduino IDE 17](#_Toc79788412)

# DANH MỤC BẢNG BIỂU

[Table 1 Bảng Phân công công việc 2](#_Toc79788570)

[Table 2 Danh sách linh kiện sử dụng 14](#_Toc79788571)

[Table 3 Bảng các bài kiểm thử hệ thống 18](#_Toc79788572)

# 

# CHƯƠNG 1. TỔNG QUAN ĐỀ TÀI

## Đặt vấn đề

Hiện tại dịch Covid-19 tiếp tục diễn biến phức tạp, nguy cơ lây lan cao. Hàng ngày có hàng nghìn ca nhiễm mới. Vì vậy việc thiết kế tạo ra 1 máy kiểm tra đo thân nhiệt từ xa, cho kết quả chính xác và nhanh chóng là 1 nhu cầu thiết yếu. Máy đo thân nhiệt từ xa là một trong những giải pháp công nghệ hữu hiệu nhằm hạn chế sự lây nhiễm chéo của Covid-19 trước tình hình dịch bệnh diễn biến ngày càng phức tạp. Nếu nhân viên y tế sử dụng nhiệt kế để đo trực tiếp như cách thường làm hiện nay thì vẫn có khả năng gặp rủi ro khi tiếp xúc. Thiết bị đo thân nhiệt từ xa cho phép cách ly nhân viên y tế với người cần đo nhiệt độ nên loại trừ được khả năng lây nhiễm này.

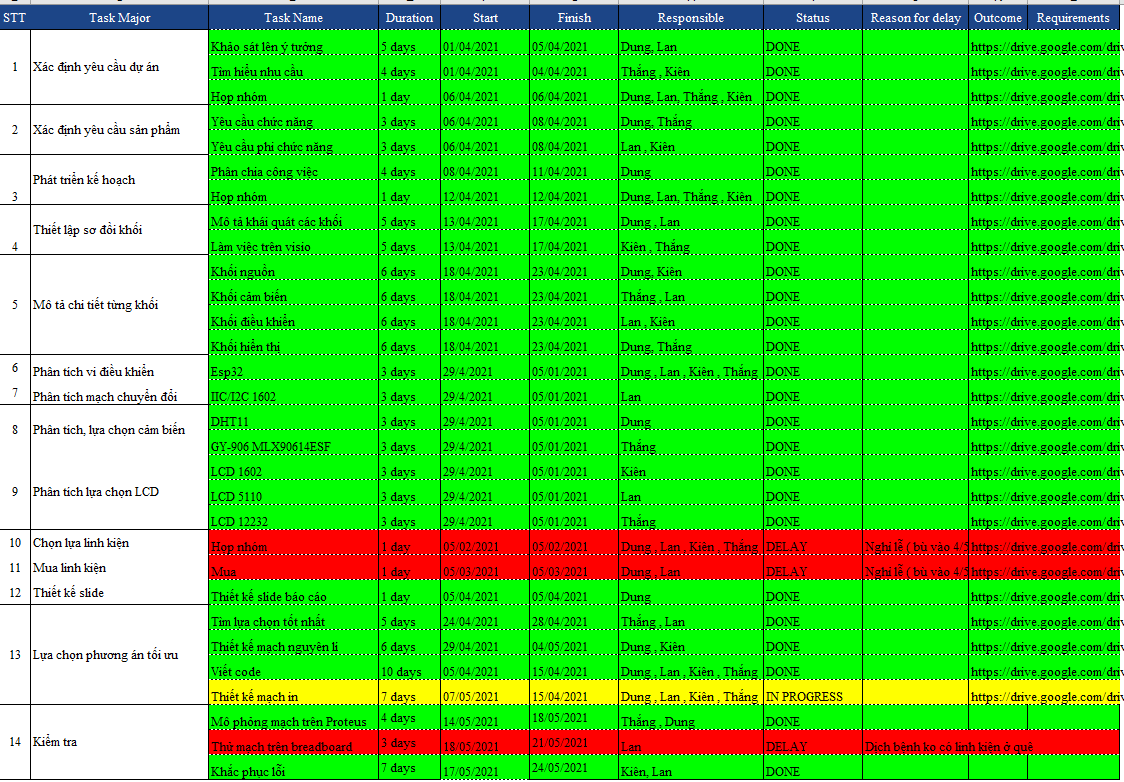
## Giới thiệu đề tài

Từ thực trạng cùng nhu cầu nêu trên, nhóm 12 thuộc lớp Kỹ thuật vi xử lý, dưới sự hướng dẫn của TS Hàn Huy Dũng, đã quyết định thực hiện tìm hiểu và triển khai một dự án với đề tài “Thiết kế máy đo thân nhiệt từ xa”. Dự án với mục tiêu cụ thể là thiết kế 1 máy đo thân nhiệt từ xa, giữ được khoảng cách khi đo thân nhiệt, hiện thị được thân nhiệt của người được đo lên màn hình, khi nhiệt độ quá ngưỡng thì báo còi có thể sàng lọc được những trường hợp có triệu chứng sốt để tránh lây lan dịch bệnh.

## Phân công công việc

Nhằm phát triển “Máy đo thân nhiệt từ xa” dựa trên những kiến thức đã học trong học phần Kỹ thuật vi xử lý cùng kiến thức tìm hiểu liên quan, để có thể làm việc nhóm thì việc xây dựng các mục công việc rồi phân công công việc cho từng thành viên là hết sức cần thiết, chính vì vậy trong chương này chúng em sẽ trình bày bảng phân công công việc đến từng thành viên.

Table Bảng Phân công công việc



Link cụ thể của bảng phân công công việc đó chúng em để ở link này ạ

<https://drive.google.com/file/d/1JFBYlDHG3WXGukTYzShGJfbGWEW4F2V-/view?usp=sharing>



# CHƯƠNG 2. MÔ TẢ SẢN PHẨM

## 2.1 Mục tiêu

Máy đo thân nhiệt được thiết kết đáp ứng các yêu cầu về độ chính xác, cho sai số thấp. Tiết kiệm nhân lực, không cần phải có người đứng cầm máy để đo như 1 số máy trên thị trường, hoàn toàn không có tiếp xúc gần giữa người với người, đảm bảo không có khả năng lây lan dịch bệnh. Máy đo thân nhiệt có còi báo giúp cảnh báo các trường hợp có dấu hiệu sốt và có tính chính xác cao

## 2.2 Yêu cầu chưc năng

Dưới đây là 1 số yêu cầu chúng em liệt kê ra:

* Lấy mẫu đo nhiệt độ cơ thể người mà không cần đo trực tiếp cơ thể
* Khoảng cách đo tối đa: 1m
* Đo được nhiệt độ trong khoảng 20 °C – 100 °C
* Gửi dữ liệu đo hiển thị lên LCD.
* Khi phát hiện nhiệt độ quá ngưỡng đã cài đặt, cảm biến sẽ truyền tín hiệu đến chuẩn giao tiếp và kết hợp với còi báo phát ra âm thanh

## 2.3 Yêu cầu phi chức năng

Từ các đặc điểm của môi trường thực tế cũng như nhu cầu sử dụng, nhóm đưa ra các yêu cầu phi chức năng. Máy hoạt động với nguồn cung cấp 5V, độ chính xác cao 95%. Khoảng cách tối đa để có thể đo được nhiệt độ là 1m. Nhiệt độ hoạt động của máy từ -10 – 600C, độ ẩm hoạt động: 0 – 100%. Máy được thiết kế nhỏ gọn, tiện lợi phù hợp với khí hậu tại Việt Nam, kích thước khoảng 9x9x2cm dễ dàng sửa chữa, lắp đặt. Đặc biệt giá thành rẻ nhất có thể (khoảng 800.000đ).

# CHƯƠNG 3. CƠ SỞ LÝ THUYẾT

## 4.1 Vi điều khiển ESP32



Hình Vi điều khiển ESP32

Đặc tính của CPU ESP32:

* Chạy hệ 32 bit
* Tốc độ xử lý 160MHZ up to 240 MHz
* Tốc độ xung nhịp đọc flash chip 40mhz --> 80mhz (tùy chỉnh khi lập trình)
* RAM: 520 KByte SRAM :520 KB SRAM liền chip –(trong đó 8 KB RAM RTC tốc độ cao – 8 KB RAM RTC tốc độ thấp (dùng ở chế độ DeepSleep).

**Nguồn điện hoạt động*:***

* Nhiệt độ hoạt động -40 + 85C
* Điện áp hoạt động: 2.2-3.6V
* Số cổng GPIOs : 34

Các loại giao tiếp mà ESP32 hỗ trợ:

* 8-bit DACs( digital to analog)  2 cổng
* Analog(ADC)  12-bit  16 cổng.
* I²C – 2 cổng
* UART – 3 cổng
* SPI – 3 cổng (1 cổng cho chip FLASH )
* I²S – 2 cổng
* SD card /SDIO/MMC host
* Slave (SDIO/SPI)
* Ethernet MAC interface with dedicated DMA and IEEE 1588 support
* CAN bus 2.0
* IR (TX/RX)
* Băm xung PWM (tất cả các chân )
* Ultra low power analog pre-amplifier

**Cấu trúc sơ đồ chân của ESP32**

Diagram, schematic

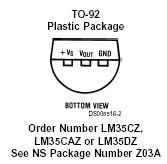
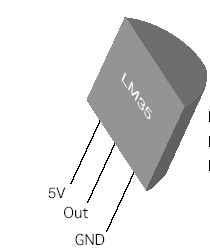
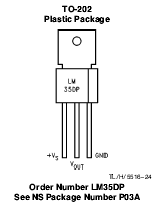
Description automatically generated

Hình 2 Cấu trúc sơ đồ chân của ESP32

* *Các thiết bị ngoại vi ESP 32 bao gồm :*
* 18 kênh chuyển đổi từ tương tự sang số (ADS)
* 3 giao diện SPI
* 3 giao diện UART
* 2 giao diện I2C
* 16 kênh dầu ra PWM
* 2 bộ chuyển đổi kỹ thuật số sang tương tự (DAC)
* 2 giao diện I2S
* 10 GPIO cảm biến điện dung
* Chân chỉ được cấu hình làm đầu vào: GPIO 34, GPIO 35, GPIO 36, GPIO 37, GPIO 38, GPIO 39
* Chân được kết nối với bộ nhớ SPI Flash nội bộ trên ESP-WROOM-32: GPIO 6 (SCK / CLK), GPIO 7 (SDO / SD0), GPIO 8 (SDI / SD1), GPIO 9 (SHD / SD2), GPIO 10 (SWP / SD3), GPIO 11 (CSC / CMD).
* Chân sử dụng đọc ADC-Bộ chuyển đổi tương tự sang số: ADC1\_CH0 (GPIO 36),ADC1\_CH1 (GPIO 37),ADC1\_CH2 (GPIO 38), ADC1\_CH3 (GPIO 39), ADC1\_CH4 (GPIO 32), ADC1\_CH5 (GPIO 33), ADC1\_CH6 (GPIO 34), ADC1\_CH7 (GPIO 35), ADC2\_CH0 (GPIO 4), ADC2\_CH1 (GPIO 0), ADC2\_CH2 (GPIO 2), ADC2\_CH3 (GPIO 15), ADC2\_CH4 (GPIO 13), ADC2\_CH5 (GPIO 12), ADC2\_CH6 (GPIO 14), ADC2\_CH7 (GPIO 27), ADC2\_CH8 (GPIO 25), ADC2\_CH9 (GPIO 26)
* Chân chuyển đổi kỹ thuật số sang tương tự (DAC): DAC1 (GPIO25), DAC2 (GPIO26)

## 4.2 Cảm biến nhiệt độ LM35

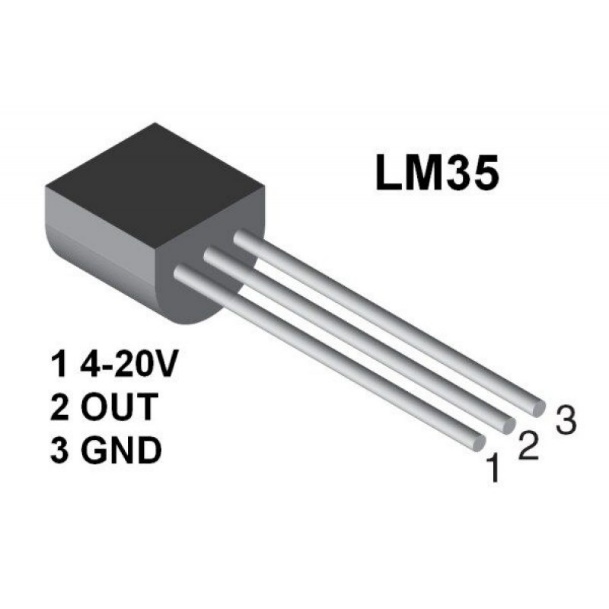
Figure Cảm biến nhiệt độ LM35



Cảm biến nhiệt độ LM35 có điện áp Analog đầu ra tuyến tính theo nhiệt độ thường được sử dụng để đo nhiệt độ của môi trường hoặc theo dõi nhiệt độ của thiết bị,..., cảm biến có kiểu chân TO-92 với chỉ 3 chân rất dễ giao tiếp và sử dụng.

Thông số kỹ thuật của cảm biến LM35

* Điện áp hoạt động: 4~20VDC
* Công suất tiêu thụ: khoảng 60uA
* Khoảng đo: -55°C đến 150°C
* Điện áp tuyến tính theo nhiệt độ: 10mV/°C
* Sai số: 0.25°C
* Kiểu chân: TO92
* Kích thước: 4.3 × 4.3mm



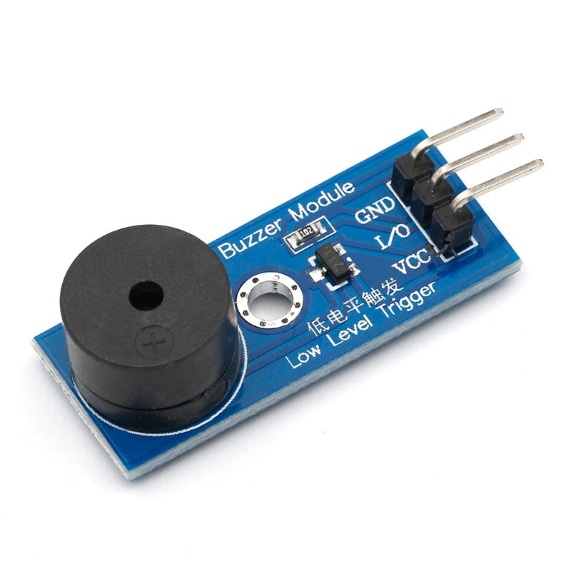
Hình Sơ đồ chân của cảm biến LM35

Cách kết nối chân của LM35:

* (1) : nối nguồn 4-20V
* OUT ↔ MCU.IO
* GND: nối GND

Từ việc phân tích các thống số cũng như sơ đồ chân tín hiệu của LM35, cảm biến LM35 đáp ứng được tất cả các yêu cầu của máy đo tín hiệu, giá thành rẻ ( khoảng 26.000đ) là 1 lựa chọn phù hợp với đề tài

## 4.3 Còi buzzer



Hình 4 Còi buzzer

Còi Buzzer có tuổi thọ cao, hiệu suất ổn định, chất lượng tốt, được sản xuất nhỏ gọn phù hợp thiết kế với các mạch còi buzzer nhỏ gọn, mạch báo động.

Thông số kỹ thuật:

* Nguồn : 3.5V - 5.5V
* Dòng điện tiêu thụ: <25mA
* Tần số cộng hưởng: 2300Hz ± 500Hz
* Biên độ âm thanh: >80 dB
* Nhiệt độ hoạt động:-20 °C đến +70 °C
* Kích thước : Đường kính 12mm, cao 9,7mm

Còi Buzzer 5VDC có 3 chân:

* GND: nối với GND
* I/O: nối với chân tín hiệu
* VCC: Nối nguồn

## 4.4 LCD1602

Ngày nay, thiết bị hiển thị [**LCD 1602**](http://www.suachualaptop24h.com/Linh-kien-laptop.html) (Liquid Crystal Display) được sử dụng trong rất nhiều các ứng dụng của vi điều khiển. LCD 1602 có rất nhiều ưu điểm so với các dạng hiển thị khác như: khả năng hiển thị kí tự đa dạng (chữ, số, kí tự đồ họa); dễ dàng đưa vào mạch ứng dụng theo nhiều giao thức giao tiếp khác nhau, tiêu tốn rất ít tài nguyên hệ thống, giá thành rẻ,…



Hình LCD 1602

Thông số kỹ thuật:

* Điện áp MAX : 7V
* Điện áp MIN : - 0,3V
* Hoạt động ổn định : 2.7-5.5V
* Điện áp ra mức cao : > 2.4
* Điện áp ra mức thấp : <0.4V
* Dòng điện cấp nguồn : 350µA - 600µA
* Nhiệt độ hoạt động : -30 – 75 độ C

Ảnh có chứa đồng hồ

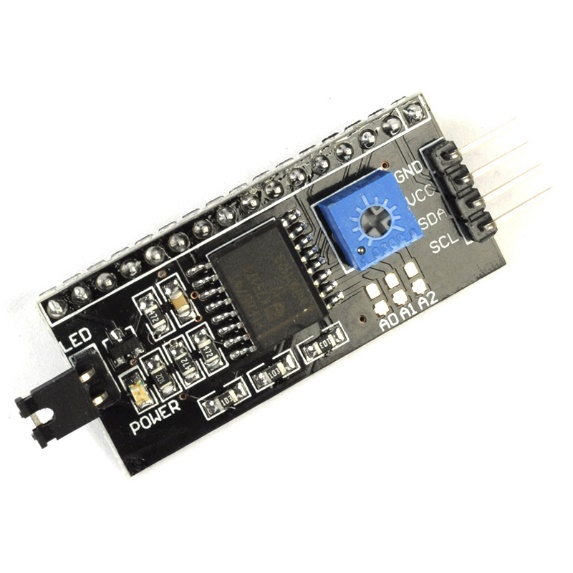
Mô tả được tạo tự động

Hình Sơ đồ chân tín hiệu LCD1602

Chức năng của từng chân LCD 1602:

* Chân số 1 - VSS : chân nối đất cho LCD được nối với GND của mạch điều khiển.
* Chân số 2 - VDD : chân cấp nguồn cho LCD, được nối với VCC=5V của mạch điều khiển.
* Chân số 3 - VE : điều chỉnh độ tương phản của LCD.
* Chân số 4 - RS : chân chọn thanh ghi, được nối với logic "0" hoặc logic "1":
* Logic “0”: Bus DB0 - DB7 sẽ nối với thanh ghi lệnh IR của LCD (ở chế độ “ghi” -   write) hoặc nối với bộ đếm địa chỉ của LCD (ở chế độ “đọc” - read).
* Logic “1”: Bus DB0 - DB7 sẽ nối với thanh ghi dữ liệu DR bên trong LCD
* Chân số 5 - R/W : chân chọn chế độ đọc/ghi (Read/Write), được nối với logic “0” để ghi hoặc nối với logic “1” đọc.
* Chân số 6 - E : chân cho phép (Enable). Sau khi các tín hiệu được đặt lên bus DB0-DB7, các lệnh chỉ được chấp nhận khi có 1 xung cho phép của chân này như sau:
* Ở chế độ ghi: Dữ liệu ở bus sẽ được LCD chuyển vào thanh ghi bên trong khi phát hiện một xung (high-to-low transition) của tín hiệu chân E.
* Ở chế độ đọc: Dữ liệu sẽ được LCD xuất ra DB0-DB7 khi phát hiện cạnh lên (low-to-high transition) ở chân E và được LCD giữ ở bus đến khi nào chân E xuống mức thấp.
* Chân số 7 đến 14 - D0 đến D7: 8 đường của bus dữ liệu dùng để trao đổi thông tin với MPU. Có 2 chế độ sử dụng 8 đường bus này là: Chế độ 8 bit (dữ liệu được truyền trên cả 8 đường, với bit MSB là bit DB7) và Chế độ 4 bit (dữ liệu được truyền trên 4 đường từ DB4 tới DB7, bit MSB là DB7).
* Chân số 15 - A: nguồn dương cho đèn nền.
* Chân số 16 - K: nguồn âm cho đèn nền.

## 4.5 Mạch chuyển đổi I2C cho LCD



Hình Mạch chuyển đổi I2C cho LCD

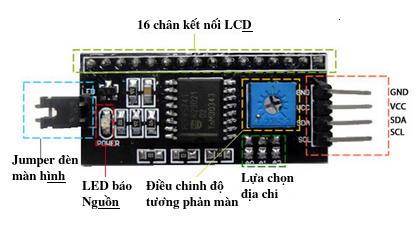
LCD có quá nhiều chân gây khó khăn trong quá trình kết nối và chiếm dụng nhiều chân của vi điều khiển? Module chuyển đổi I2C cho LCD sẽ giải quyết vấn đề này cho bạn, thay vì sử dụng tối thiểu 6 chân của vi điều khiển để kết nối với LCD (RS, EN, D7, D6, D5 và D4) thì với module chuyển đổi bạn chỉ cần sử dụng 2 chân (SCL, SDA) để kết nối. Module chuyển đổi I2C hỗ trợ các loại LCD sử dụng driver HD44780(LCD 1602, LCD 2004, … ), kết nối với vi điều khiển thông qua giao tiếp I2C, tương thích với hầu hết các vi điều khiển hiện nay.

Ưu điểm

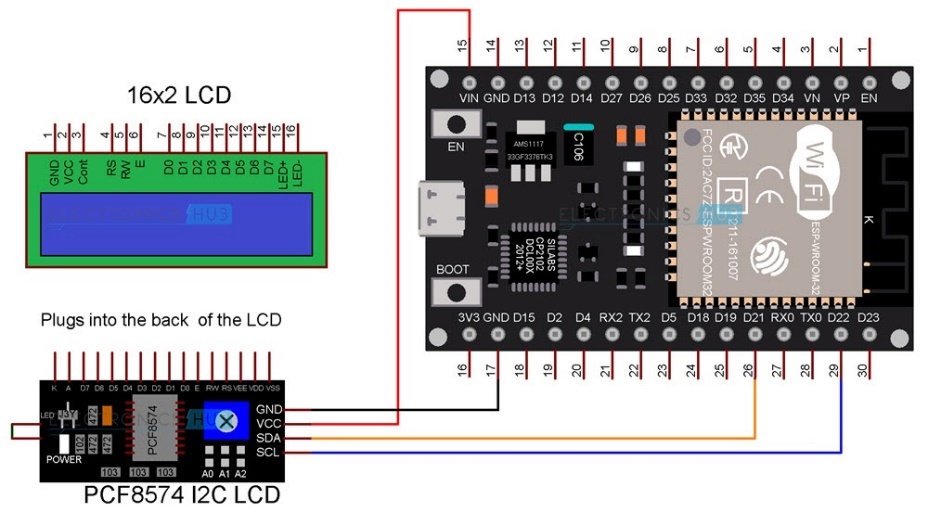
* Tiết kiệm chân cho vi điều khiển
* Dễ dàng kết nối với LCD

Thông số kĩ thuật

* Điện áp hoạt động: 2.5-6V DC
* Hỗ trợ màn hình: LCD1602,1604,2004 (driver HD44780)
* Giao tiếp: I2C
* Địa chỉ mặc định: 0X27 (có thể điều chỉnh bằng ngắn mạch chân A0/A1/A2)
* Kích thước: 41.5mm(L)x19mm(W)x15.3mm(H)
* Trọng lượng: 5g
* Tích hợp Jump chốt để cung cấp đèn cho LCD hoặc ngắt
* Tích hợp biến trở xoay điều chỉnh độ tương phản cho LCD



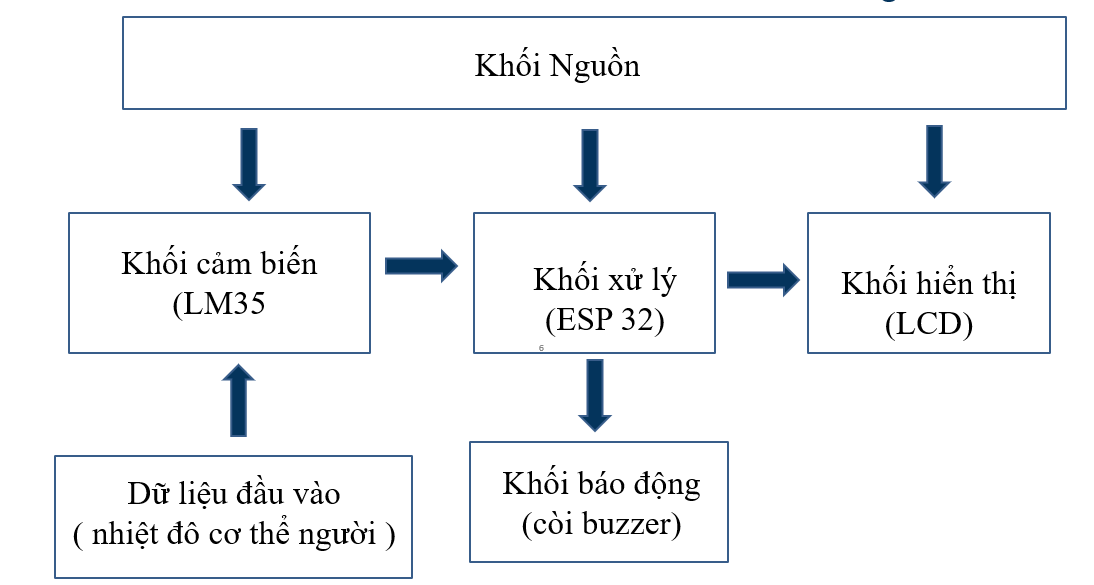
Hình Sơ đồ chân mạch chuyển đổi I2C



Hình Sơ đồ cách kết nối mạch chuyển đổi I2C

# CHƯƠNG 4. THIẾT KẾ MẠCH

## 4.1 Xây dựng sơ đồ khối của mạch



Hình Sơ đồ khối hệ thống

Thiết bị được chia thành năm khối chính bao gồm:

* Khối dữ liệu đầu vào: khối dữ liệu đầu vào này là nhiệt độ cơ thể người
* Khối nguồn: chuyển đổi điện áp từ pin và adapter sang điện áp 5v và 3.3v cung cấp cho các khối khác hoạt động
* Khối cảm biến : Khối này có cảm biến LM35. Cảm biến LM35 nhận dữ liệu từ khối dữ liệu và gửi sang cho khối xử lý. Khi nhận được tín hiệu từ khối xử lý thì phát ra còi báo động
* Khối xử lý: Vi điều khiển ESP32 là chip xử lý trung tâm, điều khiển hoạt động của thiết bị.
* Khối hiển thị lên LCD: gồm màn hình LCD
* Khối báo động: Còi báo buzzer sẽ được kích hoạt khi quá ngưỡng nhiệt độ được thiết lập.

## Nguyên lý hoạt động của mạch

Dữ liệu đầu vào được đưa vào khối cảm biến, lúc này khối cảm biến là cảm biến nhiệt độ LM35 sẽ nhận dữ liệu đầu vào và chuyển dữ liệu sang khối xử lý. Khối xử lý là vi điều khiển ESP32 nhận dữ liệu và phân tích, đưa ra kết quả sang khối hiển thị LCD, khối xử lý phân tích khi có trường hợp nhiệt độ quá mức cho phép sẽ gửi tín hiệu sang khối báo động có modul còi buzzer làm còi buzzer hoạt động. Trong suốt quá trình các khối cảm biến , xử lý và hiển thị hoạt động nhờ khối nguồn cấp nguồn.

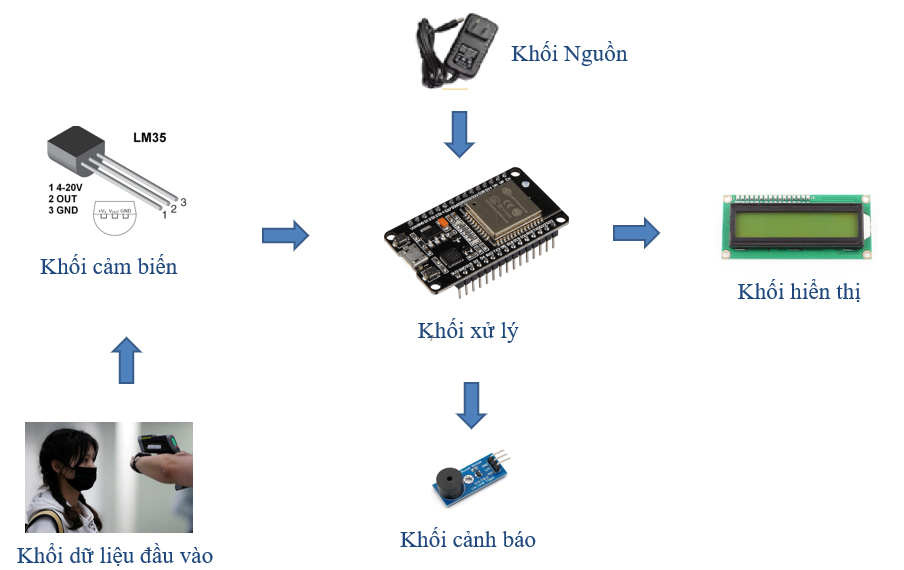
## 4.3 Lựa chọn linh kiện

Dưới đây là các linh kiện nhóm chúng em chọn để tiến hành thực hiện đề tài:

Table Danh sách linh kiện sử dụng

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên linh kiện** | **Số lượng** |
| 1 | ESP32 | 1 |
| 2 | LCD 1602 I2C | 1 |
| 3 | Cảm Biến Nhiệt Độ LM35 | 1 |
| 4 | Module còi báo buzzer | 1 |

Từ các linh kiện trên ta đưa vào sơ đồ khối như sau:

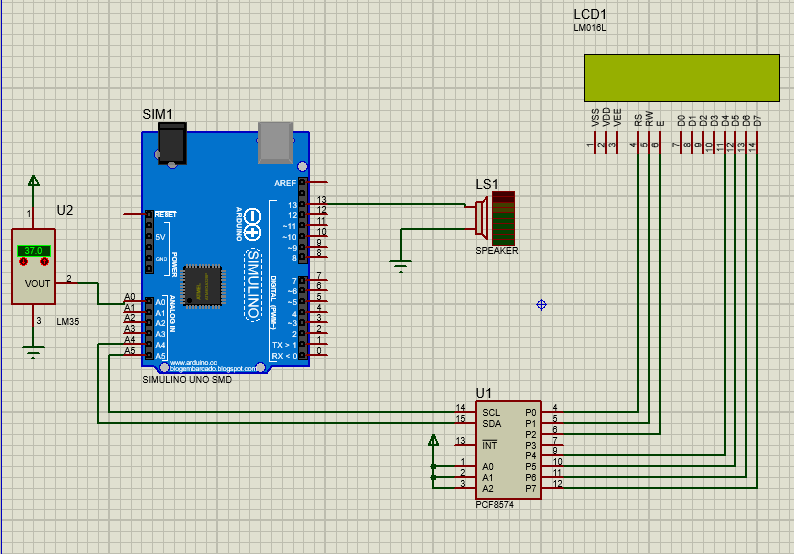


Hình Sơ đồ khối hệ thống với các linh kiện sử dụng

## 4.4 Xây dựng sơ đồ nguyên lý của mạch

Do Proteus không hỗ trợ với vi điều khiển ESP32, còi báo buzzer, nên nhóm chúng em dùng các linh kiện thay thế là simulino uno, còi báo SPEAKER.

Dưới đây là sơ đồ nguyên lý của mạch:

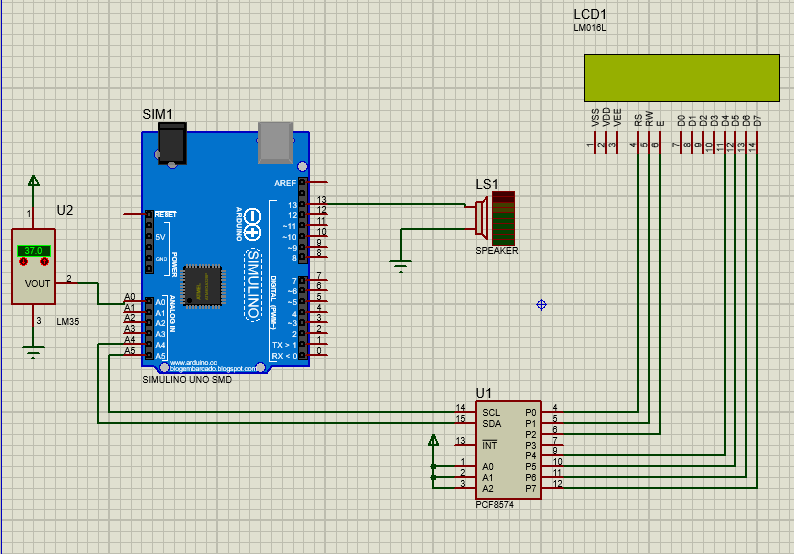


Hình 5.3 Sơ đồ nguyên lý hệ thống

# CHƯƠNG 5. SƠ ĐỒ MẠCH VÀ CODE HỆ THỐNG

## 5.1 Sơ đồ mạch

Nhóm chúng em sử dụng phần mềm Proteus để tiến hành vẽ sơ đồ mạch:



Hình 6.1 Sơ đồ mạch sử dụng phần mềm Proteus

## 6.2 Công nghệ sử dụng

Arduino [6], [7] là một nền tảng mã nguồn mở bao gồm cả phần mềm và phần cứng được thiết kế giúp làm việc với các board mạch điện tử trở nên dễ dàng hơn bao giờ hết.

Arduino che dấu đi sự phức tạp của điện tử bằng cách đơn giản hóa các khái niệm, che đi sự phức tạp của phần mềm bằng các thủ tục ngắn gọn thay phải làm việc xuống mức cấp thấp của hardware, nhưng với Arduino thì chỉ cần gọi 1 hàm.

Bởi vì tính phổ biến và dễ dùng cùng với các vô vàn thư viện được tích hợp sẵn để làm việc với rất nhiều module phần cứng khác nhau, bạn chỉ cần quan tâm đến tính năng sản phẩm mà bỏ qua các khái niệm phức tạp (protocol, datasheet …​) từ đó dễ dàng tiếp cận và làm ra các sản phẩm tuyệt vời mà không cần phải biết nhiều về điện tử.

Arduino bao gồm một phần mềm lập trình là Arduino IDE và một tập hợp rất nhiều các board mạch Arduino có thể lập trình được bằng phần mềm này với các biến thể khác nhau. Ban đầu phần lớn các board này đều dựa trên các chip họ AVR của Atmel sản xuất, nhưng sau này có rất nhiều nhà sản xuất sử dụng các chip khác nhau như ARM, PIC, STM32 gần đây nhất là ESP8266, ESP32… với năng lực phần cứng và phần mềm đi kèm mạnh mẽ hơn nhiều cũng phát hành các thư viện giúp làm việc được với Arduino như các board Arduino chính chủ.

Graphical user interface, text, application

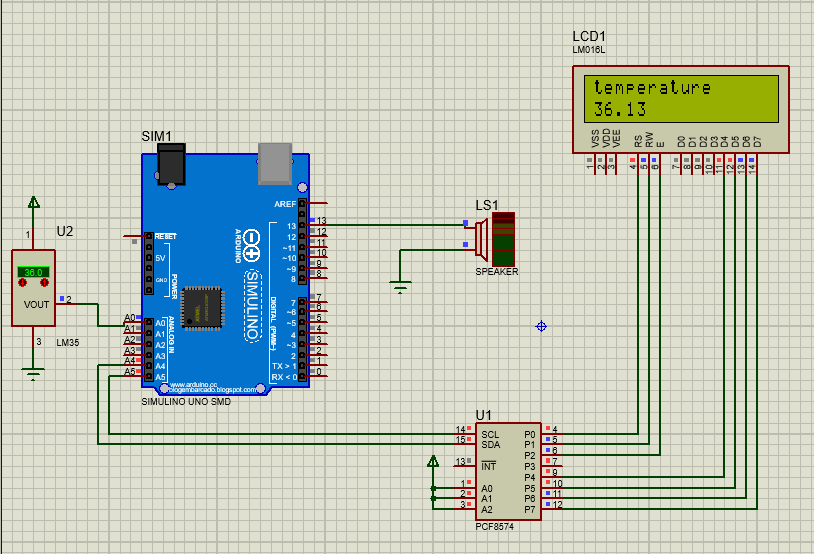
Description automatically generated

Hình Giao diện Arduino IDE

# CHƯƠNG 7. TIẾN HÀNH THỬ NGHIỆM, TEST LỖI VÀ KHẮC PHỤC HỆ THÔNG

## 7.1 Tiến hành thử nghiệm

Do tình hình dịch bệnh nhóm chúng em không có linh kiện để làm việc cùng với nhau nên nhóm em không hoàn thành được mạch thực tế. Kết quả chạy mô phỏng cho kết quả chính xác



Hình 7.1 Kết quả chạy mô phỏng bằng phần mềm Proteus

## 7.2 Kiểm thử

Trong quá trình thực hiện nhóm đã thực hiện kiểm thử hệ thống theo các kịch bản sau:

Table Bảng các bài kiểm thử hệ thống

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Mục đích kiểm thử** | **Thủ tục kiểm thử** | **Kết quả mong muốn** | **Kết quả kiểm thử** |
| 1 | Kiểm thử hiển thị trên màn hình LCD | 1. Kết nối màn hình LCD với ESP32 2. Lập trình hiển thị các thông số nhiệt độ lên màn hình | Màn hình LCD hiển thị các thông số nhiệt độ | Đạt |
| 2 | Kiểm thử đầu ra của cảm biến LM35 | 1. Kết nối cảm biến LM35 với ESP32 2. Lập trình hiển thị các thông số nhiệt độ lên màn hình LCD 3. Thay đổi nhiệt độ xung quanh cảm biến bằng các nguồn nhiệt | Các thông số nhiệt độ hiển thị trên màn hình thay đổi theo sự thay đổi của các nguồn nhiệt | Đạt |
| 3 | Kiểm thử chức năng báo còi | 1. Kết nối còi buzzer với ESP32 2. Kết nối cảm biến LM35 với ESP32 3. Thay đổi nhiệt độ đầu vào trên 38°C | Còi kêu, và sáng báo động | Đạt |

## 7.3 Test lỗi hệ thống

Do mới dừng lại ở việc mô phỏng bằng phần mềm nên chưa xuất hiện lỗi thực tế, từ việc chạy thử mô phỏng nhóm chúng em tiến hành tăng và giảm nhiệt độ ở con cảm biến, kết quả hiển thị lên LCD thay đổi tốt.Khi nhiệt độ quá ngưỡng 38 độ còi sẽ báo động. Tóm lại mạch hoạt động ổn định cho ra kết quả tốt.

# KẾT LUẬN

Sau khi tiến hành phân tích, tiến hành thiết kế máy đo, nhóm đã đạt được các chỉ tiêu đề ra lúc đầu. Tuy nhiên cho ảnh hưởng của dịch bệnh nên không được thực hiện trên thực tế, mọi thiết kế, kiểm thử chỉ mới thực hiện ở bước mô phỏng. Nên quá trình từ mô phỏng chuyển đổi thành mạch thực tế sẽ có những bất cập mà kết quả thử nghiệm cho ra không chính xác.

Qua bài tập lớn này nhóm cũng đã học tập được rất nhiều kiến thức về chuẩn giao tiếp của các con cảm biến với bộ vi xử lý ESP32, nắm rõ thêm một số các thuật toán và lập trình và nhận thấy được phần quan trọng của vi xử lý đối với cuộc sống hiện tại. Nhóm chúng em xin được cảm ơn thầy Hàn Huy Dũng đã giúp đỡ giảng dạy cho nhóm chúng em về kiến thức về bộ môn Vi Xử lý để nhóm có thể dễ nắm bắt và hoản thiện được bài tập lớn này

# TÀI LIỆU THAM KHẢO

[1] <https://epcb.vn/products/esp32-devkitc-su-dung-module-esp-wrover>

[2]<https://www.dientudat.com/combo-lap-bo-cam-bien-nhiet-do-khong-tiep-xuc-hien-thi-lcd-bao-coi>

[3]<https://machdienlythu.vn/do-nhiet-do-bang-cam-bien-lm35-su-dung-arduino-uno/>

[4] <http://dammedientu.vn/bai-11-doc-cam-bien-nhiet-lm35-voi-8051/>

[5] <https://randomnerdtutorials.com/esp32-esp8266-i2c-lcd-arduino-ide/>

[6] “Arduino Homepage.” <https://www.arduino.cc/>.

[7] “Arduino.” <https://en.wikipedia.org/wiki/Arduino>

# PHỤ LỤC

**Mã nguồn hệ thống**

Link github mã nguồn: <https://github.com/Dungquach0306/project-vxl-team12>

|  |
| --- |
| #include <Wire.h>  #include <LiquidCrystal\_I2C.h> // khai bao thu vien lcd sử dụng i2c  LiquidCrystal\_I2C lcd(0x27,16,2);  #define INTERVAL\_MESSAGE1 1000  #define INTERVAL\_MESSAGE2 2000  int fan = 13; // led chân số 13  unsigned long time1 = 0;  unsigned long time2 = 0;  void setup()  {  Serial.begin(115200);  lcd.init(); // khoi tạo lcd  lcd.backlight(); // bật đèn lcd  pinMode(fan,OUTPUT);  lcd.setCursor(0,1);  lcd.print("cam bien lm35 ");  delay(1000);  lcd.clear();    }  void loop()  {  int reading = analogRead(A0);  float temperature = (5.0\*reading\*100.0/1024.0);  if ( (unsigned long) (millis() - time1) > INTERVAL\_MESSAGE1)  {  if(temperature >= 38 )  {  lcd.clear();  lcd.setCursor(0,0);  lcd.print("temperature");  lcd.setCursor(0,1);  lcd.print(temperature);  digitalWrite(fan,HIGH);  }  time1 = millis();  }  if ( (unsigned long) (millis() - time2) > INTERVAL\_MESSAGE2)  {  if(temperature < 38 )  {  lcd.clear();  lcd.setCursor(0,0);  lcd.print("temperature");  lcd.setCursor(0,1);  lcd.print(temperature);  digitalWrite(fan,LOW);  }  time2 = millis();  }    } |