|  |
| --- |
| TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI  **VIỆN ĐIỆN TỬ - VIỄN THÔNG**  logo_128  **BÁO CÁO BÀI TẬP LỚN MÔN**  **KĨ THUẬT VI XỬ LÝ**  **Đề tài: Máy đo nhiệt độ sử dụng cảm biến**  **hồng ngoại**  Giảng viên hướng dẫn: TS. Hàn Huy Dũng  Mã lớp: 124833  Nhóm sinh viên thực hiện:  Hà Nội, 6-2021 |

**LỜI NÓI ĐẦU**

Ngày nay với sự phát triển mạnh mẽ của khoa học kĩ thuật, cuộc sống con người đã và đang ngày càng phát triển, với những trang thiết bị hiện đại phục vụ trong công cuộc công nghiệp hóa, hiện đại hóa. Chúng ta có thể tìm thấy vô vàn các thiết bị điện tử được sử dụng ở khắp mọi nơi để phục vụ cho nhiều nhu cầu các nhau. Đóng vài trò quan trọng không thể thiếu để có thể làm ra các thiết bị điện tử đó chính là các vi xử lý có khả năng lập trình thiết kế, tùy biến để tạo ra các thiết bị điện tử có mục đích sử dụng khác nhau.

Trong lĩnh vực y sinh, ngày càng nhiều máy móc được ra đời với mục đích chuẩn đoán, chăm sóc sức khỏe bệnh nhân, hỗ trợ các y bác sĩ trong quá trình khám chữa bệnh. Trong bối cảnh dịch bệnh COVID-19 đang hoành hành ở trên khắp thế giới, việc đo nhiệt độ là bước đầu tiên để chuẩn đoán và phát hiện bệnh, từ đó có thể đưa ra các biện pháp chữa trị kịp thời.

Nhận thức được tầm quan trọng của vấn đề, chúng em chọn đề tài “ Thiết kế máy đo thân nhiệt từ xa “ để làm bài tập lớn.

Chúng em xin chân thành cảm ơn thầy Hàn Huy Dũng và các anh chị Devzone đã hướng dẫn tận tình cho chúng em, trau dồi cho chúng em thêm nhiều kiến thức và các kĩ năng mềm cũng như giúp chúng em hiểu được quá trình thiết kế một sản phẩm kĩ thuật.

**MỤC LỤC**

[DANH MỤC KÝ HIỆU VÀ CHỮ VIẾT TẮT i](#_Toc9951116)

[DANH MỤC HÌNH VẼ ii](#_Toc9951117)

[DANH MỤC BẢNG BIỂU iii](#_Toc9951118)

[PHẦN 1. THIẾT KẾ 1](#_Toc9951120)

[1.1 Xác đinh yêu cầu 1](#_Toc9951121)

[1.2 Mô tả sản phẩm 2](#_Toc9951122)

[1.2.1 Yêu cầu chức năng 2](#_Toc9951123)

[1.2.2 Yêu cầu phi chức năng 3](#_Toc9951124)

[1.3 Sơ đồ khối 4](#_Toc9951128)

[1.4 Chi tiết từng khối 8](#_Toc9951143)

[1.4.1 Khối vi điều khiển 8](#_Toc9951144)

[1.4.2 Khối hiển thị 11](#_Toc9951145)

[1.4.3 Khối cảm biến 13](#_Toc9951148)

[1.4.4 Khối nguồn 13](#_Toc9951148)

[1.5 Sơ đồ nguyên lý 8](#_Toc9951143)

[CHƯƠNG 2. CÁC QUY ĐỊNH KHÁC 13](#_Toc9951146)

[2.1 Liệt kê tài liệu tham khảo 13](#_Toc9951147)

[2.1.1 Cách liệt kê 13](#_Toc9951148)

[2.1.2 Các loại tài liệu tham khảo 13](#_Toc9951149)

[2.2 Trích dẫn tài liệu tham khảo 15](#_Toc9951150)

[2.3 Nộp đồ án 16](#_Toc9951151)

[2.3.1 Nộp bản cứng 16](#_Toc9951152)

[2.3.2 Nộp bản mềm 16](#_Toc9951153)

[2.4 Bản quyền kết quả nghiên cứu 17](#_Toc9951154)

[KẾT LUẬN 18](#_Toc9951155)

[Kết luận chung 18](#_Toc9951156)

[Hướng phát triển 18](#_Toc9951157)

[Kiến nghị và đề xuất 18](#_Toc9951158)

[TÀI LIỆU THAM KHẢO 19](#_Toc9951159)

[PHỤ LỤC 20](#_Toc9951160)

[Phụ lục 1. Mẫu trang bìa chính của đồ án 20](#_Toc9951161)

[Phụ lục 2. Mẫu trang bìa phụ của đồ án 22](#_Toc9951162)

[Phụ lục 3. Mẫu nhận xét đồ án 24](#_Toc9951163)

# DANH MỤC KÝ HIỆU VÀ CHỮ VIẾT TẮT

Xem Mục 1.3.6

# DANH MỤC HÌNH VẼ

[Hình 1.1 Sơ đồ khối của hệ thống 3](#_Toc9951164)

(Xem thêm Mục 1.3.7)

# DANH MỤC BẢNG BIỂU

[Bảng 1.1 Kết quả thí nghiệm 3](#_Toc9951165)

(Xem thêm Mục 1.3.8)

# PHẦN 1. Tổng quan đề tài

## Đặt vấn đề

Với mỗi một quốc gia, sự phát triển về kinh tế luôn đi theo sự phát triển về khoa học, kỹ thuật. Trong lĩnh vực y sinh, ngày càng nhiều máy móc được ra đời với mục đích chuẩn đoán, chăm sóc sức khỏe bệnh nhân, hỗ trợ các y bác sĩ trong quá trình khám chữa bệnh. Với bối cảnh dịch bệnh COVID-19 đang hoành hành ở trên khắp thế giới, việc đo nhiệt độ là bước đầu tiên để chuẩn đoán và phát hiện bệnh, từ đó có thể đưa ra các biện pháp chữa trị kịp thời. Vì thế, đề tài “Thiết kế máy đo nhiệt độ từ xa” là một đề tài thiết thực trong cuộc sống, giúp sinh viên rèn luyện kỹ năng làm việc với mạch nhúng, trau dồi vốn kiến thức được học trên giảng đường.

## Các giải pháp, hệ thống đã có

Máy đo nhiệt độ bằng bức xạ hồng ngoại



Hình . Súng bắn nhiệt độ

## Phân tích yêu cầu thiết kế

#### Xác định yêu cầu

Thiết kế và chế tạo máy đo nhiệt độ từ xa cầm tay, nhỏ gọn, dễ sử dụng, hoạt động ổn định trong nhiều môi trường khác nhau. Kết quả hiển thị trên màn hình phải rõ nét. Thiết bị có thể duy trì hoạt động trong khoảng thời gian từ 10-12 tiếng 1 ngày. Giá thành sản phẩm không quá đắt, dễ tiếp cận đến với người tiêu dùng

#### Yêu cầu chức năng

#### Thiết bị có khả năng phát hiện năng lượng hồng ngoại phát ra từ cơ thể người, chuyển đổi thành tín hiệu điện có thể hiển thị theo đơn vị nhiệt độ, đo ở khoảng cách tối thiểu 5 cm, kết quả đầu ra hiển thị trên màn hình LCD OLED. 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. Ngoài ra thiết bị phải được tích hợp chế độ giảm tải điện năng khi thiết bị không được sử dụng.

* + 1. *Yêu cầu phi chức năng*

Thiết bị hoạt động ở điện áp từ 3.3 – 5V**,** có sai số nhỏ khoảng 0.1%, tốc độ lấy mẫu dưới 0.5 giây**.** Cảm biến thích hợp cho hầu hết các vật liệu và có thể đo được nhiệt độ cho con người thông qua vải, quần áo …Thiết kế nhỏ nhẹ và dễ sử dụng với mọi người**.** Thiết bị có khả năng chịu đựng và hoạt động tốt tại môi trường có nhiều biến động (nóng, ẩm) ở Việt Nam, chống nước, bụi.Mã nguồn sử dụng ngôn ngữ C dễ tiếp cận

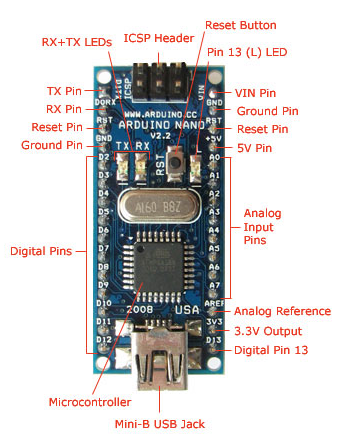
# PHẦN 2. THIẾT KẾ HỆ THỐNG

## 2.1 Thiết kế sơ đồ khối

**2.2 Chi tiết từng khối**

*2.2.1 Khối vi điều khiển*

Mạch sử dụng bo mạch Arduino Nano là khối điều khiển. Một mạch Arduino bao gồm một vi điều khiển AVR với nhiều linh kiện bổ sung giúp dễ dàng lập trình và có thể mở rộng với các mạch khác. Giống với Arduino Uno R3, Arduino Nano sử dụng vi điều khiển Atmega328p thuộc họ chip megaAVR. Cấu tạo chính của Aruino Nano được thể hiện ở hình dưới



Hình . Cấu tạo board mạch Arduino Nano

* Nguồn đầu vào: Arduino Nano có thể được cấp nguồn từ chân VIN hoặc chân 5V. Điểm khác biệt giữa Arduino Nano và Arduino Uno là Arduino Uno sử dụng cổng USB type B, còn Arduino Nano sử dụng cổng mini USB, giúp cho kích thước board giảm đi khá nhiều.
* Chân năng lượng : GND (Ground), là cực âm của nguồn điện cấp cho Arduino Uno. Các chân 5V, 3.3V là nơi cấp điện áp 5V, 3.3V đầu ra, dòng tối đa ở chân này là 500mA, 50mA. Vin là chân cấp nguồn ngoài cho Arduino. Điện áp hoạt động của vi điều khiển có thể được đo trên chân IOREF. Chân RESET dùng để reset vi điều khiển khi nối với điện áp 10kΩ.
* Thạch anh dao động 16 MHz để kiểm soát các bus, xung nhịp xử lý, cho ra tần số rất ổn định.
* Chân digital: gồm 12 chân vào/ra đánh số từ D2 đến 13, ngoài ra có 1 chân nối đất GND và một chân điện áp tham chiếu AREF. Hai chân serial 0 (RX) và 1 (RX) dùng để gửi (transmit - TX) và nhận (receive - RX) dữ liệu TTL serial. Arduino có thể giao tiếp với thiết bị khác thông qua 2 chân này. Chân PWM (~): 3,5,6,9,10 và 11 cho phép xuất ra xung PWM với độ phân giải 8bit bằng hàm analogWrite(), tức có thể điều chỉnh điện áp ra ở chân này từ mức 0V đến 5V thay vì cố định ở mức 0V và 5V.
* Chân giao tiếp SPI: 10 (SS), 11(MOSI), 12(MISO), 13(SCK).
* Chân analog (A0–A7) cung cấp độ phân giải tín hiệu 10bit (0 - 210-1) để đọc điện áp trong khoảng 0-5V. Chân A4 (SDA) và A5(SCL) hỗ trợ giao tiếp I2C/TWI.
* Vi điều khiển Atmega328 cung cấp cho người dùng 32KB bộ nhớ flash, 2KB cho SRAM, 1KB EPROM.

Chi tiết về từng chân và chức năng của nó được thể hiện ở bảng dưới

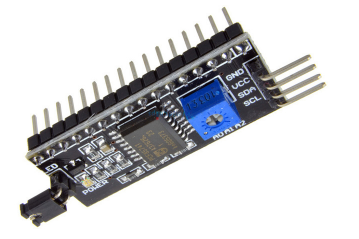
|  |  |  |
| --- | --- | --- |
| Tên Pin | Kiểu | Chức năng |
| D1/TX | I/O | Ngõ vào/ra số  Chân TX truyền dữ liệu |
| D0/TX | I/O | Ngõ vào/ra số  Chân RX nhận dữ liệu |
| RESET | Đầu vào | Chân reset, hoạt động ở mức thấp |
| GND | Nguồn | Chân nối mass |
| D2 | I/O | Ngõ vào/ra digital |
| D3 | I/O | Ngõ vào/ra digital |
| D4 | I/O | Ngõ vào/ra digital |
| D5 | I/O | Ngõ vào/ra digital |
| D6 | I/O | Ngõ vào/ra digital |
| D7 | I/O | Ngõ vào/ra digital |
| D8 | I/O | Ngõ vào/ra digital |
| D9 | I/O | Ngõ vào/ra digital |
| D10 | I/O | Ngõ vào/ra digital |
| D11 | I/O | Ngõ vào/ra digital |
| D12 | I/O | Ngõ vào/ra digital |
| D13 | I/O | Ngõ vào/ra digital |
| 3V3 | Đầu ra | Đầu ra 3.3V (từ FTDI) |
| AERF | Đầu vào | Tham chiếu ADC |
| A0 | Đầu vào | Kênh đầu vào tương tự 0 |
| A1 | Đầu vào | Kênh đầu vào tương tự 1 |
| A2 | Đầu vào | Kênh đầu vào tương tự 2 |
| A3 | Đầu vào | Kênh đầu vào tương tự 3 |
| A4 | Đầu vào | Kênh đầu vào tương tự 4 |
| A5 | Đầu vào | Kênh đầu vào tương tự 5 |
| A6 | Đầu vào | Kênh đầu vào tương tự 6 |
| A7 | Đầu vào | Kênh đầu vào tương tự 7 |
| +5V | Đầu ra hoặc đầu vào | Đầu ra 5V từ bộ điều chỉnh onboard hoặc 5V đầu vào từ nguồn điện bên ngoài |
| RESET | Đầu vào | Chân đặt lại, hoạt động ở mức thấp |
| GND | Nguồn | Chân nối mass |
| VIN | Nguồn | Chân nối với nguồn vào |

Bảng . Chi tiết các chân của Arduino Nano

Lý do chọn Arduino Nano là khối điều khiển vì bo mạch này dễ sử dụng, tiếp cận đối với người dùng chưa có nhiều kinh nghiệm sử dụng các vi điều khiển. Ngoài ra, Arduino Uno có các phương thức giao tiếp đa dạng phù hợp để kết nối với nhiều thiết bị ngoại vi. Với kích thước cực kì nhỏ gọn và khả năng tương tự như Arduino Uno, ta có thể ứng dụng nó trong các dự án yêu cầu kích thước nhỏ và khối lượng nhẹ: Giá thành của Arduino Uno cũng khá rẻ chỉ từ 45.000đ.

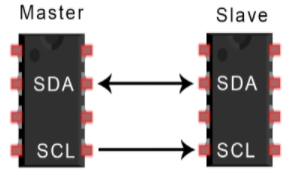
*2.2.2 Khối hiển thị*

Dùng để hiển thị kết quả nhiệt độ đo được sau khi xử lý. Hiện nay, hầu hết các mạch điện tử đều sử dụng màn hình hiển thị kết hợp với module I2C vì tính tiện lợi và tác dụng của nó.



Hình . Module I2C LCD 16x2

Giao tiếp I2C sử dụng mô hình truyền thông Master/Slave, trong đó một thiết bị hoặc một quá trình được chỉ định là Master sở hữu quyền kiểm soát các thiết bị, thiết bị hoặc quy trình khác chịu quyền kiểm soát của Master được gọi là Slave. Hướng điều khiển luôn chảy từ chủ sang nô lệ.



Hình . Mô tả mô hình truyền thông Master/Slave

I2C kết hợp các tính năng tốt nhất của SPI và UART. Với I2C, ta có thể kết nối nhiều slave với một master duy nhất (như SPI) và có thể có nhiều master điều khiển một hoặc nhiều slave. Giống như giao tiếp UART, I2C chỉ sử dụng hai dây để truyền dữ liệu giữa các thiết bị: SDA (Serial Data) - đường truyền cho master và slave để gửi và nhận dữ liệu, SCL (Serial Clock) - đường mang tín hiệu xung nhịp. I2C là một giao thức truyền thông nối tiếp, vì vậy dữ liệu được truyền từng bit dọc theo một đường duy nhất (đường SDA). Giống như SPI, I2C là đồng bộ, do đó đầu ra của các bit được đồng bộ hóa với việc lấy mẫu các bit bởi một tín hiệu xung nhịp được chia sẻ giữa master và slave. Tín hiệu xung nhịp luôn được điều khiển bởi master.

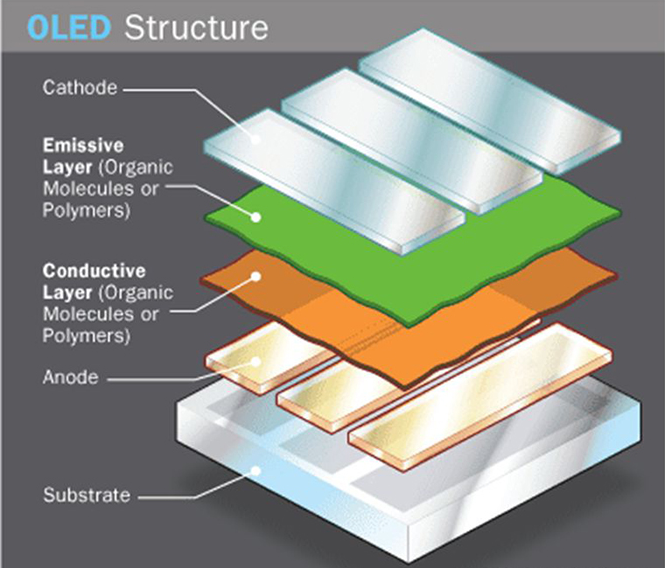
Ở trong đề tài này, chúng em quyết định sử dụng màn hình OLED 128x64 làm khối hiển thị cho thiết bị.



Hình . Minh họa màn hình OLED 128x64

Màn hình OLED SSD1306 có 0.96 inch, cho khả năng hiển thị hình ảnh tốt với khung hình 128x64 pixel. Màn hình được tích hợp sẵn giao tiếp I2C tương thích với hầu hết các vi điều khiển hiện nay. Màn hình sử dụng driver SSD1306 cùng thiết kế nhỏ gọn giúp phát triển các sản phẩm DIY hoặc các ứng dụng khác một cách nhanh chóng. SSD1306 có mức tiêu thụ điện năng thấp, chỉ 0.08W. Có thể điều chỉnh độ sáng và độ tương phản. Điện áp hoạt động từ 3V-5V DC, nhiệt độ hoạt động từ -30℃-70℃.

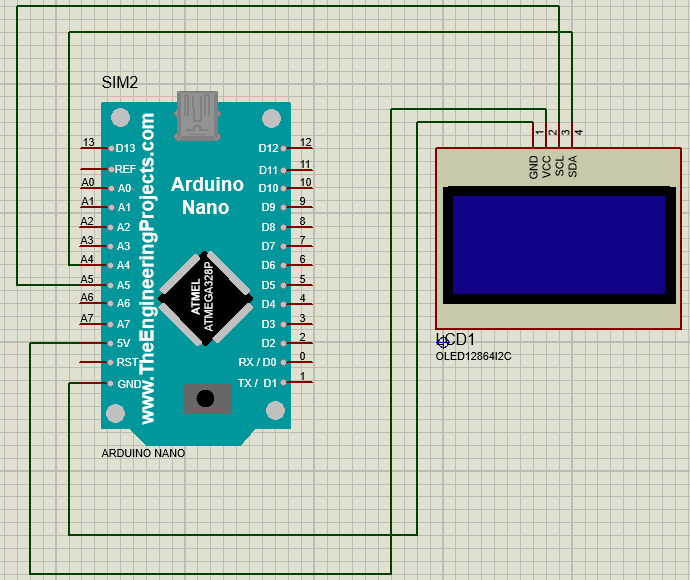
OLED viết tắt của Organic Light Emitting Diodes, là một loại điốt phát quang (LED) trong đó lớp phát xạ điện quang là một màng thuốc (film) làm bằng vật liệu từ một loại chất bán dẫn hữu cơ có khả năng phát sáng khi có dòng điện chạy qua. Đồng thời, lớp phát sáng phải được đặt giữa hai điện cực và một trong hai điện cực này là trong suốt.



Hình . Cấu tạo màn hình OLED

Cấu tạo màn hình OLED gồm có các phần sau: Tấm nền (substrate) – làm từ nhựa trong, thủy tinh, … có tác dụng chống đỡ cho OLED. Anode (trong suốt) – sẽ lấy đi các electron (hay tạo ra các lỗ trống mang điện dương) khi có một dòng điện chạy qua thiết bị. Các lớp hữu cơ được tạo thành từ các phân tử hữu cơ hay polymer. Lớp dẫn (conductive layer) – được làm từ các phân tử hữu cơ dẻo có nhiệm vụ truyền tải các lỗ trống từ anode. Trong đó có một loại polymer dẫn được sử dụng trong các OLED gọi là polyaniline. Lớp phát sáng (emissive layer) – được làm từ các phân tử hữu cơ dẻo (nhưng khác loại với lớp dẫn) có nhiệm vụ truyền tải các electron từ cathode. Trong đó có một loại polymer được dùng trong lớp phát sáng gọi là polyfluorence. Cathode (có thể trong suốt hoặc không tùy thuộc vào loại OLED) – sẽ tạo ra các electron khi có dòng điện chạy qua thiết bị.

Điểm mạnh của màn hình OLED là hình ảnh nịnh mắt, sắc nét; độ sáng và độ tương phản cao; màu đen hiển thị sâu; có khả năng tùy biến theo nhiều hình dạng khác nhau;tiết kiệm năng lượng và góc nhìn rộng hơn. Bên cạnh đó, nó cũng tồn tại những điểm yếu như: không phải là lựa chọn phù hợp với những người cần một màn hình có chất lượng hiển thị trung thực cho công việc.



Hình . Sơ đồ nối dây giữa màn hình OLED và Arduino Nano

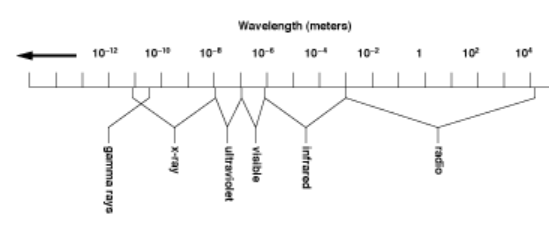
OLED lấy nguồn 5V từ Arduino Nano, GND của OLED được nối với chân GND của Arduino. Chân SDA và SCL được nối với chân A4, A5 của Arduino.

*2.2.3 Khối cảm biến*

Một trong những giải pháp xây dựng hệ thống cảm biến có thể đo nhiệt độ cao mà không làm hỏng hệ thống là sử dụng cảm biến nhiệt độ không tiếp xúc Cảm biến này có thể cảm nhận nhiệt độ của một đối tượng mà không cần chạm vào đối tượng.

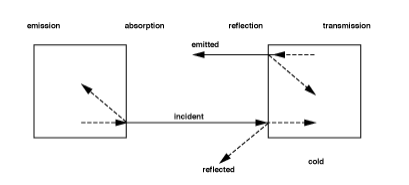
Nguyên lý hoạt động:

Năng lượng hồng ngoại (IR) được phát ra từ tất cả các vật liệu có nhiệt độ trên 0°K. Bức xạ hồng ngoại là một phần của Phổ điện từ và nằm giữa tần số ánh sáng khả kiến (Ánh sáng nhìn thấy) và sóng vô tuyến. Phần hồng ngoại của phổ kéo dài bước sóng từ 0,7μm đến 1000μm. Trong dải sóng này, chỉ các tần số từ 0,7μm đến 20μm được sử dụng để đo nhiệt độ thực tế hàng ngày. Điều này là do cảm biến nhiệt độ IR hiện có không đủ nhạy để phát hiện lượng năng lượng rất nhỏ có sẵn ở bước sóng vượt quá 20μm.



Hình . Bước sóng của một số loại sóng

Năng lượng hồng ngoại di chuyển theo đường thẳng từ nguồn và có thể bị phản xạ và hấp thụ bởi các bề mặt vật chất trên đường đi của nó. Trong trường hợp hầu hết các vật thể rắn, một phần năng lượng hồng ngoại đập vào bề mặt vật thể sẽ bị hấp thụ và một phần sẽ bị phản xạ, một số năng lượng IR cũng sẽ đi qua vật thể. Những hiện tượng này được gọi là độ phát xạ của vật thể hoặc vật liệu.

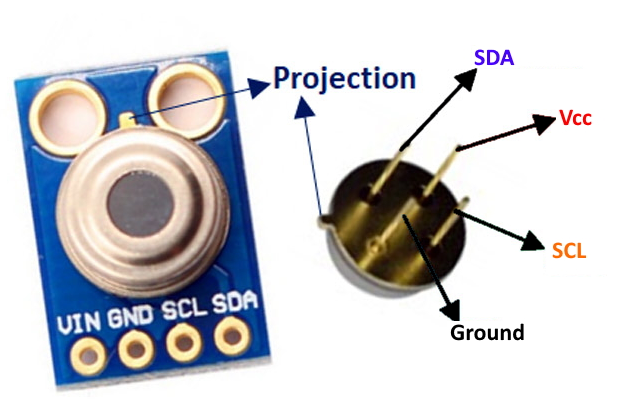


Hình . Quá trình phát xạ của vật thể

Các loại vật liệu và khí khác nhau có độ phát xạ khác nhau, và do đó sẽ phát ra IR ở các cường độ khác nhau trong một nhiệt độ nhất định. Sự phát xạ của vật liệu hoặc khí là một chức năng của cấu trúc phân tử và đặc điểm bề mặt của nó. Nó thường không phải là một chức năng của màu sắc trừ khi nguồn màu là một chất hoàn toàn khác với cơ thể chính của vật liệu.

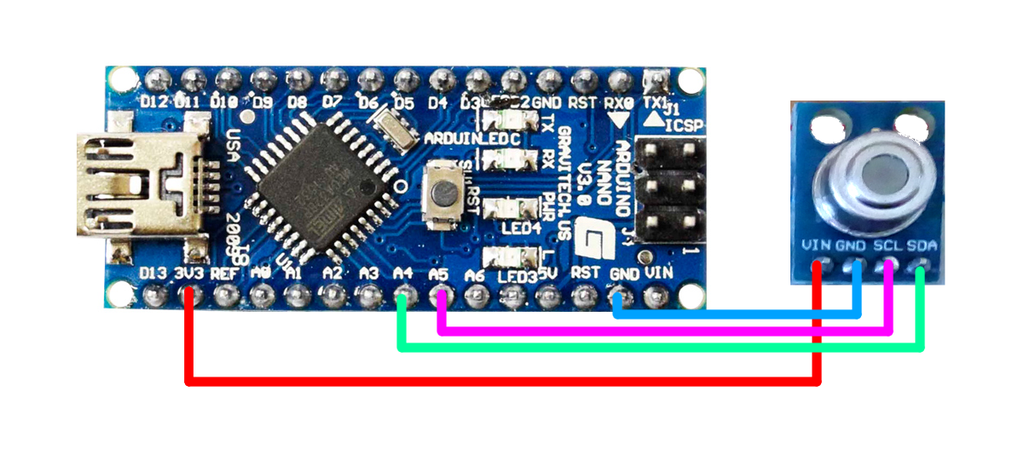
Cũng giống như trường hợp với ánh sáng khả kiến, một số bề mặt càng được đánh bóng cao thì bề mặt IR sẽ phản xạ càng nhiều. Do đó, các đặc tính bề mặt của vật liệu cũng sẽ ảnh hưởng đến độ phát xạ của nó. Trong đo nhiệt độ, điều này có ý nghĩa trong trường hợp vật liệu có độ phát xạ thấp. Ngoài cấu trúc phân tử và điều kiện bề mặt, yếu tố thứ ba ảnh hưởng đến sự phát xạ rõ ràng của vật liệu hoặc khí là độ nhạy bước sóng của cảm biến, được gọi là đáp ứng quang phổ của cảm biến. Như đã nêu ở trên, chỉ các bước sóng hồng ngoại trong khoảng 0,7μm và 20μm được sử dụng để đo nhiệt độ thực tế.

MLX90614 là một cảm biến nhiệt độ dựa trên bức xạ hồng ngoại, nó đo nhiệt độ dựa trên bức xạ hồng ngoại do một vật thể phát ra. Nó cảm nhận được các sóng điện từ trong khoảng cách từ 700 nm đến 14.000 nm. Cảm biến sử dụng tia IR để đo nhiệt độ của vật thể mà không cần tiếp xúc vật lý và giao tiếp với bộ vi điều khiển bằng giao thức I2C.



Hình . Cảm biến nhiệt độ MLX90614 và sơ đồ chân

Thông số kĩ thuật: Cảm biến MLX90614 có phạm vi đo nhiệt độ rộng, có thể đo được nhiệt độ của một đối tượng cụ thể từ -40 -> 125oC. Cảm biến có độ chính xác khá cao, với sai số 0.5oC . Kích thước của cảm biến khá nhỏ: 11.5 x 16.5mm. Cảm biến có 2 phiên bản dành cho 2 nguồn đầu vào khác nhau là 3.3V và 5V.



Hình . Sơ đồ nối dây giữa cảm biến MLX90614 dùng nguồn 3.3V và Arduino Nano

*2.2.4 Khối nguồn*

Tính toán dòng sử dụng

* Dhtt11 : V = 5v , Imax = 1mA; => P= 5mW
* LCD:

+, V=5v

+, Idd = 2.5ma

+, Iled = 16 ma

* Công suất tiêu thụ của LCD = V\*( Idd+Iled)= 0.1425 w

Năng lượng tiêu thụ dự tính

P=(Plcd+Pdht)(1+30%)~ 0.2W

Chọn pin 9v ~ 2500mAh : đáp ứng cho mạch có thể sử dụng trong một thời gian dài

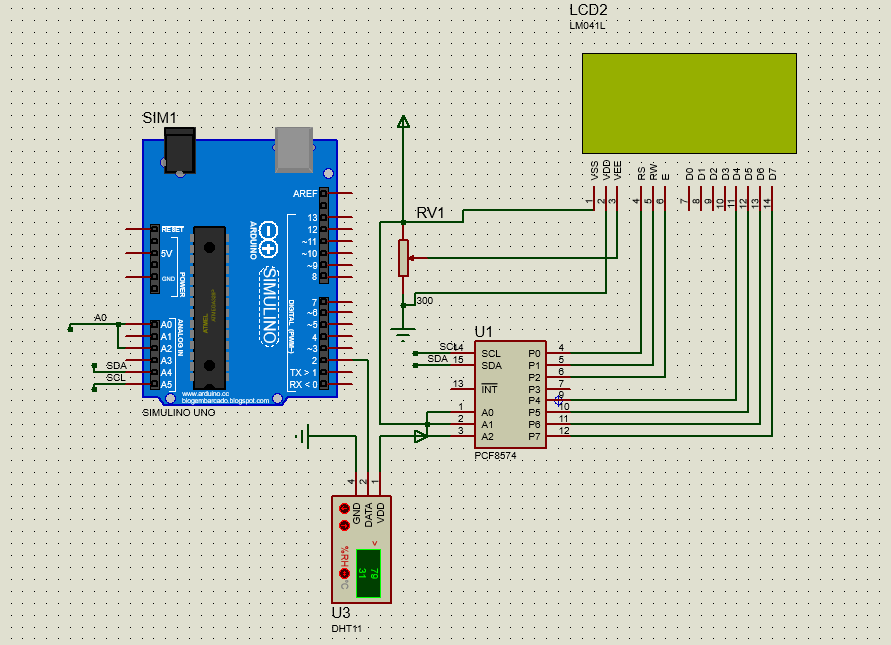


Hình . Pin 9V và dây nối

# PHẦN 3. LẬP TRÌNH VÀ MÔ PHỎNG

## 3.1 Sơ đồ mô phỏng

Mạch gồm có một board Arduino Nano, sensor DHT11, màn hình LCD 128x64, bộ giao tiếp i2c PCF8574



Hình . Sơ đồ mô phỏng mạch trên Proteus

Arduino được cấp nguồn từ Pin 9V, thông qua cổng kết nốt SIM1

Sơ đồ chân:

* DHT11:

+, VDD , GND nối lần lượt với chân 5V và GND của arduino để cấp nguồn cho cảm biến

+, Chân DATA nối với chân 2 của arduino : đường dữ liệu từ cảm biến nối đến arduino bằng giao thức onewrite

* LCD

+, Chân VSS , VDD của lcd nối với chân 5v và GND của arduino : cấp nguồn cho LCD

+, Chân VEE nối với chân 5v qua biến trở : hiện thị độ sáng tối của màn hình và điều chỉnh qua biến trở

* IC PCF8574 :

+, P0,P1,P2 nối với các chân RS, RW, E của LCD: 3 chân điều khiển chế độ ghi viết của LCD

+, P4,P5,P6,P7 nối lần lượt với các chân D4,D5,D6,D7 của LCD : gửi dữ liệu cho LCD ở chế độ 4bit

+, 2 chân SCL,SDA lần lượt nối với 2 chân A5, A4 của arduino : đường truyền giao tiếp i2c giữ arduino bằng 2 chân analog

### 3.2 Lập trình

*3.2.1 Thư viện sử dụng*

#include <LabC.h>

#include<TimerOne.h>

#include <Wire.h>

#include <LiquidCrystal\_I2C.h>

#include <DHT.h>

#include <avr/interrupt.h>

* #include<TimerOne.h> : khai báo timer 1 , dùng với mục đích tạo hàm trễ trong code
* #include <LiquidCrystal\_I2C.h> : khai báo thư viện giao thức I2c cho chân analog A4 và A5
* #include <DHT.h> : khai báo thư viện cho cảm biến 2 DHT11 và DHT12 gồm sử dụng các hàm đọc nhiệt độ từ cảm biến
* #include <avr/interrupt.h> : khai báo thư viện trình phục vụ ngắt – cụ thể là ngắt ngoài và ngắt timer

*3.2.2 Các hàm sử dụng*

Seria.begin(x): khai báo làm việc với giao tiếp serial, tốc độ truyền tải dữ liệu giữa board và PC là x bps.

- Hàm hiện thị cho LCD:

+, lcd.init() : hàm khởi tạo LCD

+, lcd.backlinght() : bật đèn nền LCD

+, lcd.print(“ nội dung “) : hàm viết nội dung lên màn hình LCD

+, lcd.setCursor (i,n) : hàm chỉnh vị trí còn trỏ trên màn hình ( i là vị trí cột trên hàng n )

- Ngắt timer :

Sử dụng timer1 của arduino để tạo hàm trễ thay hàm delay();

+, trong hàm void setup():

Timer1.initialize(TIMER\_US); // khởi động ngắt, và đặt thời gian cho nó , trong code TIMER\_US = 5000 ( thời gian thiết lập là 5ms )

Timer1.attachInterrupt(Delay); // khi xảy ra ngắt trương trình sẽ gọi đến hàm Delay

- Hàm cảm biến DHT11:

+, dht.begin() : khởi động cảm biến

+, dht.readHumidity() : đọc giá trị độ ẩm từ cảm biến

+, dht.readTemperature() : đọc giá trị nhiệt độ từ cảm biến

- Khai báo trình ngắt ngoài :

+, attachInterrupt(1,ngat,CHANGE) : khai báo ngắt ngoài 1 ở chân d3 chế độ CHANGE – chế độ thay đổi mức logic. Sau khi ngắt xảy ra trương chình sẽ gọi đến hàm ngắt

3.2.3 Khai báo biến

#define TIMER\_US 5000 // thời gian trễ = 5 ms

const int DHTPIN = 2; // khai báo chân data sensor

const int DHTTYPE = DHT11; // khai báo loại cảm biến- có 2 loại cảm biến cho thư viện dht là DHT11 và DHT22, ở đây sử dụng cảm biến DHT11

int koi = 13; // khai báo chân còi bao

int binh\_thuong = 8; // khai báo chân đèn báo trạng thái nhiệt độ bình thường

int sot = 9; // khai báo chân đèn báo trạng thái nhiệt độ khi sốt

int button = 3; // khai báo chân nút nhấn chạy của máy

int led = 7; // chân báo đèn chạy của máy

boolean nhan = false; //khai báo biến logic phụ vụ ngắt

boolean starts = true; // biến logic chạy

volatile byte s = HIGH; //khai báo biến điện áp cho ngắt

float a; // biến trung gian trễ lưu nhiệt độ

*3.2.3 Sơ đồ trạng thái và trạng thái CPU*

Nút đo

Trạng thái bắt đầu

ấn nút nguồn

Nút đo

ấn nút nguồn

Trạng thái nghỉ

Hình . Sơ đồ trạng thái máy đo nhiệt độ

Giải thích sơ đồ trạng thái

+, Trạng thái nghỉ: khi thiết bị tắt bởi nút nguồn , khi đó sẽ ngắt điện đến CPU. Màn hình tắt sáng và không hiển thị

+, Trạng thái bắt đầu: màn hình sáng và hiển thị , và đèn báo

+, Trạng thái 1:

* Sau khi đo xong nhiệt độ ( trạng thái 2 ) , màn hình sẽ hiện thị nhiệt độ sau đo và chờ hiệu lệnh khác từ bên ngoài ( đèn báo sáng

+, Trạng thái 2: Sau khi ấn giữ nút đo thiết bị sẽ chuyển từ trạng thái 1 sang trạng thái đo . Ấn giữ nút đo trong khoảng 2 giây để CPU có thể lấy dữ liệu . Thả nút sẽ chuyển từ trạng thái 2 sang trạng thái 1 để hiện thị kết quả đo . Đèn vàng tắt

- Trạng thái CPU nghỉ

+, CPU load từ 30% đến 40%

+, Ram sử dụng 512 byte (24%) , maximum là 2048 byte

+,Sketch đã sử dụng 7244 bytes (22%) trên bộ nhớ của chip , maximum 32256 byte

- Trạng thái CPU hoạt động :

+, CPU load 53% - 65%

### 3.2 Mã nguồn

#include <LabC.h>

#include <avr/interrupt.h>

#include <Wire.h>

#include <TimerOne.h>

#include <LiquidCrystal\_I2C.h>

#include <DHT.h>

LiquidCrystal\_I2C lcd(0x27,16,4);

#define TIMER\_US 5000 // thời gian trễ = 5 ms

const int DHTPIN = 2; //

const int DHTTYPE = DHT11;

int koi = 13;

int binh\_thuong = 8;

int sot = 9;

int button = 3;

int led = 7;

boolean nhan = false;

boolean starts = true;

volatile byte s = HIGH;

float a;

DHT dht(DHTPIN, DHTTYPE);

byte degree[8] = {

0B01110,

0B01010,

0B01110,

0B00000,

0B00000,

0B00000,

0B00000,

0B00000

};

float t;

void Delay(){

t = dht.readTemperature();

};

void ngat(){

s=!s;

nhan = !nhan;

starts = false;

}

void setup() {

pinMode(led,OUTPUT);

pinMode(koi,OUTPUT);

pinMode(binh\_thuong, OUTPUT);

digitalWrite(binh\_thuong,HIGH);

pinMode(sot,OUTPUT);

digitalWrite(sot,HIGH);

pinMode(button,INPUT);

digitalWrite(led,HIGH);

Serial.begin(9600);

Timer1.initialize(TIMER\_US); // khởi tạo timer 1 đến 1 giây

Timer1.attachInterrupt(Delay); // khai báo ngắt timer 1

attachInterrupt(1,ngat,CHANGE);

lcd.init(); // hàm khỏi tạo lcd

lcd.backlight();// bật đèn nền lcd

lcd.setCursor (0,0);

lcd.print("Nhiet do: ");//

lcd.setCursor(3,1);

lcd.print("bai tap lon ");

lcd.setCursor (1,3);

lcd.print(" nhom 2 ");

lcd.createChar(1, degree);

dht.begin();

}

void loop() {

digitalWrite(led,s);

if( starts == true){

a=t;

}

else{

if( nhan == true){

t=a;

}

else

{

Delay();

a=t;

}}

if (isnan(t)) { // Kiểm tra xem thử việc đọc giá trị có bị thất bại hay không. hàm isnan

}

else {

lcd.setCursor(10,0);

lcd.print(round(t));

lcd.print(" ");

lcd.write(1);

lcd.print("C");

if( a>37.5) {

digitalWrite(sot,LOW);

digitalWrite(binh\_thuong,HIGH);}

else {

digitalWrite(binh\_thuong,LOW);

digitalWrite(sot,HIGH);

}

}

}

# KẾT LUẬN

# 

# TÀI LIỆU THAM KHẢO

1. T. H. Cormen, C. E. Leiserson, and R. L. Rivet, *Introduction to Algorithm*. MIT Press, McGraw-Hill, 1990.
2. J. W. DuBois, S. Schuetze-Coburn, S. Cumming, and D. Paolino, “Outline of discourse transcription,” in *Talking Data: Transcription and Coding in Discourse Research*, J. A. Edwards and M. D. Lampert, Ed. Hillsdale, NJ: Lawrence Erlbaum Associates, 1993, pp. 45-89.
3. J. M. Airey, J. H. Rohfl, F. Brooks Jr., “Towards Image Realism with Interactive Update Rates in Complex Virtual Building Environments,” *Comptuer Graphics*, Vol. 24, No. 2, pp. 41-50, 1990.
4. S. Brandt, G. Nutt, T. Berk, M. Humphrey, “Soft Real time Application Execution with Dynamic Quality of Service Assurance,” in *Proceedings of the Sixth IEEE/IFIP International Workshop on Quality of Service*, Hawaii, USA, May 1998, pp. 154-163.
5. K. Riley, “Language theory: Applications versus practice,” presented at the Conf. of the Modern Language Association, Boston, MA, December 27-30, 1990.
6. J. Jones. (1991). *Networks* (2nd ed.) [Online]. Available: http://www.atm.com.

# PHỤ LỤC

**Mã nguồn đầy đủ**

#include <SPI.h>

#include <Wire.h>

#include <Adafruit\_GFX.h>

#include <Adafruit\_SSD1306.h>

#include <Adafruit\_MLX90614.h>

#define SCREEN\_WIDTH 120 // chiều rộng của oled

#define SCREEN\_HEIGHT 64 // chiều dài của oled

#define OLED\_RESET 4 // khai báo chân reset cho thiết bị

Adafruit\_SSD1306 display(SCREEN\_WIDTH, SCREEN\_HEIGHT, &Wire, OLED\_RESET); // khai báo biến display

float temperature; // khai báo biến temperature

Adafruit\_MLX90614 mlx = Adafruit\_MLX90614(); // đối tượng của sensor

void setup() {

// Hàm khởi tạo

Serial.begin(9600); // làm việc với giao tiếp serial, tốc độ truyền tải dữ liệu giũa board và pc là 9600 bps

mlx.begin(); // bắt đầu hoạt động hiển thị của OLED

pinMode(4, OUTPUT); // cho chân 4 của arduino làm output vì để buzzer ở chân 4

pinMode(7, INPUT\_PULLUP); // thêm nút bấm ở chân số 7

if(!display.begin(SSD1306\_SWITCHCAPVCC, 0x3C)){

Serial.println(F("SSD1306 allocation failed"));

for(;;); //

}

display.clearDisplay();

}

void loop() {

// put your main code here, to run repeatedly:

if(digitalRead(7) == LOW) { // chân số 7 ở mức thấp thì mạch hoạt động

temperature = mlx.readObjectTempC();

if(temperature > 37){ // khi nhiệt độ > 37

digitalWrite(4,HIGH); // cho chân 4 là buzzer ở mức cao

}

else {

digitalWrite(4,LOW);

}

printText(); //gọi hàm printText để hiện thị nhiệt độ ra màn hình

delay(500);

}

display.clearDisplay();

}

void printText(){ // khai báo hàm printText

//display.setFont(&FreeMonoBold18pt7b);

display.setTextColor(WHITE);

display.setCursor(0,50); //đưa con trỏ về vị trí (0,50)

display.print(temperature,1);

display.drawCircle(92, 35, 3, WHITE); // vẽ vòng tròn là kí hiệu nhiệt độ

display.setCursor(100, 50); //đưa con trỏ về vị trí (100,50)

display.print("C");

display.display(); // hiển thị tất cả các dòng text trên

}