

TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI



BÁO CÁO BÀI TẬP

THIẾT KẾ MẠCH ĐO LƯỜNG - CẢM BIẾN

NĂM 2022-2023

Sinh viên thực hiện: (Nhóm 1)

Đỗ Quỳnh Anh

Lớp: RAI K62

Khoa: Điện-Điện tử

Hoàng Việt Anh

Lớp: RAI K62

Khoa: Điện-Điện tử

Nguyễn Quý Việt Anh

Lớp: RAI K62

Khoa: Điện-Điện tử

Nguyễn Trung Bắc

Lớp: RAI K62

Khoa: Điện-Điện tử

Giảng viên hướng dẫn: Thạc sĩ Võ Quang Sơn

HÀ NỘI, 2023

TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI



BÁO CÁO BÀI TẬP

THIẾT KẾ MẠCH ĐO LƯỜNG - CẢM BIẾN NĂM 2022-2023

Sinh viên thực hiện: (Nhóm 1)

Đỗ Quỳnh Anh	Nam, nữ: Nữ	MSSV:213302442
Hoàng Việt Anh	Nam, nữ:Nam	MSSV:213332344
Nguyễn Quý Việt Anh	Nam, nữ:Nam	MSSV:213331150
Nguyễn Trung Bắc	Nam, nữ:Nam	MSSV:213304553

Lớp:RAI K62	Khoa: Điện-Điện tử	Năm thứ: 2/4.5
Ngành học: Kỹ thuật Robot& Trí tuệ nhân tạo		Dân tộc: Kinh

Giảng viên hướng dẫn: Thạc sĩ Võ Quang Sơn

HÀ NỘI, 2023

LỜI NÓI ĐẦU

Thực hiện bài tập lớn , nhóm thực hiện đã nhận được nhiều sự quan tâm và tạo điều kiện của Ban Giám hiệu nhà trường, cán bộ văn phòng khoa Điện -Điện tử , trường Đại học Giao thông vận tải. Nhóm thực hiện xin gửi lời cảm ơn sâu sắc đến sự giúp đỡ của Quý Thầy Cô.

Đặc biệt, nhóm thực hiện xin bày tỏ lòng biết ơn sâu sắc và chân thành nhất đến Thầy **Võ Quang Sơn** – Giảng viên trực tiếp hướng dẫn, chỉ bảo tận tình cùng những định hướng và tạo mọi điều kiện thuận lợi nhất để nhóm hoàn thành bài tập lớn này. Giảng viên hướng dẫn đã luôn theo sát từng giao đoạn và có những hỗ trợ, tư vấn kịp thời để tạo nên sản phẩm cuối cùng của nhóm thực hiện.

Mặc dù, nhóm thực hiện đã có nhiều cố gắng để thực hiện bài tập lớn một cách hoàn chỉnh nhất, song công trình làm bài tập khó tránh khỏi những thiếu sót. Nhóm nghiên cứu xin gửi lời cảm ơn đến Quý Thầy Cô trong Hội đồng nghiệm thu bài tập đã đưa ra những ý kiến vô cùng quý báu giúp nhóm thực hiện khắc phục được những thiếu sót trong công trình và góp phần nâng cao chất lượng đào tạo.

Chúng em xin chân thành cảm ơn!

Hà Nội, tháng 4 năm 2023

NHÓM THỰC HIỆN

ĐỀ BÀI SỐ 1

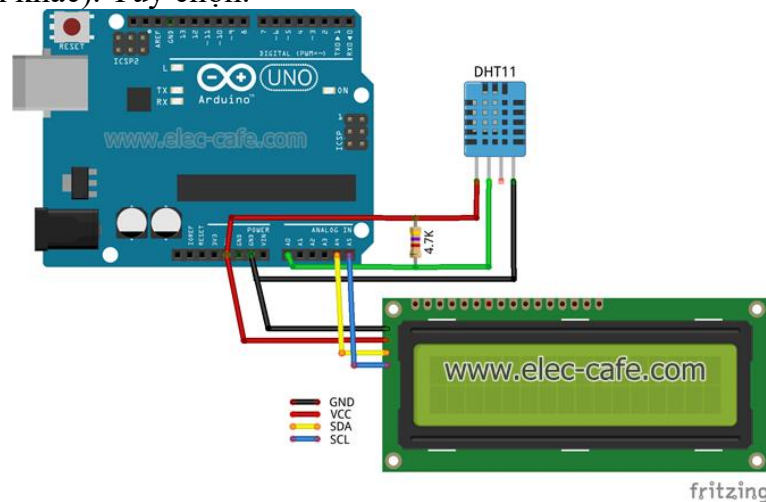
Câu 1. Thiết kế Mạch đo Nhiệt độ và cảnh báo quá nhiệt cho Vườn trồng thông minh. Thiết bị sẽ hiển thị Nhiệt độ trên LCD và phát cảnh báo bằng đèn Led/Chuông đồng thời Bật quạt khi nhiệt độ quá ngưỡng 27°C .

Thực hiện:

1a. Sử dụng Cảm biến nhiệt độ DHT11 để giám sát nhiệt độ. Bắt buộc.

- ✓ Trình bày nguyên lý đo nhiệt độ của cảm biến.
- ✓ Tín hiệu vào, tín hiệu ra.
- ✓ Trình bày các thông số kỹ thuật chính của cảm biến.
- ✓ Trình bày cách thức đọc dữ liệu cảm biến.

1b. Thiết kế mạch Vi điều khiển dùng Arduino. Vẽ mạch và Mô phỏng dùng Proteus (hoặc phần mềm khác). Tùy chọn.



Câu 2. Vẽ lại Sơ đồ mạch Thang đo DC mA của đồng hồ đo Elenco, model M1250K.

Thực hiện:

Sử dụng phần mềm vẽ mạch Altium/Proteus/Orcad (hoặc phần mềm khác).

MỤC LỤC

LỜI NÓI ĐẦU	i
ĐỀ BÀI SỐ 1	ii
MỤC LỤC	iii
PHỤ LỤC	iv
PHẦN 1: THIẾT KẾ MẠCH- SỬ DỤNG CẢM BIẾN NHIỆT ĐỘ	1
1.1. TỔNG QUAN VỀ CẢM BIẾN NHIỆT ĐỘ	1
1.1.1. Khái niệm cảm biến DHT11.....	1
1.1.2. Các loại cảm biến	1
1.1.3. Nguyên lý đo nhiệt độ của cảm biến	2
1.1.4. Tín hiệu vào, tín hiệu ra của cảm biến	3
1.1.5. Các thông số kỹ thuật chính của cảm biến	4
1.1.6. Cách thức đọc dữ liệu	4
1.2. THIẾT KẾ VI MẠCH ĐIỀU KHIỂN DÙNG ARDUINO	4
1.2.1. Tổng quan các thiết bị sử dụng trong hệ thống.....	4
1.2.1.1. Arduino	4
❖ Tổng quan Arduino.....	4
❖ Vi điều khiển	6
❖ Các cổng vào ra	7
1.2.1.2. Cảm biến DHT11 (đã được tìm hiểu ở phần 1.1).....	8
1.2.1.3. Màn hình LCD LM016L.....	8
1.2.1.4. Module giao tiếp I2C.....	8
1.2.1.5. Module Relay.....	9
1.2.1.6. Động cơ Motor DC.....	10
1.2.1.7. Đèn LED.....	1
1.2.2. Phần mềm lập trình Arduino IDE cho Arduino Uno.....	10
1.2.3. Code Arduino	12
1.2.4. Sơ đồ mạch mô phỏng.....	14
1.2.5. Kết quả.....	15
PHẦN 2: VẼ SƠ ĐỒ MẠCH THANG ĐO DC mA CỦA ĐỒNG HỒ ELENCO, MODEL M1250K	16
2.1. Đồng hồ Elenco, model M1250K... ..	16
2.2. Sơ đồ mạch thang đo DC mA.....	16
2.3. Sơ đồ mô phỏng trên Proteus.....	17
TÀI LIỆU THAM KHẢO	18

PHỤ LỤC

Danh mục hình ảnh.....	iv
Hình 1: Cảm biến DHT11.....	1
Hình 2: Sơ đồ kết nối vi xử lý.....	2
Hình 3:Sơ đồ nguyên lý hoạt động.....	2
Hình 4: Bit 0.....	3
Hình 5: Bit 1.....	3
Hình 6: Cấu trúc phần cứng của Arduino Uno.....	5
Hình 7: Vi điều khiển ATmega328.....	6
Hình 8.Sơ đồ chân của ATmega328P... ..	7
Hình 9:Các chân vào /ra của Arduino Uno.....	7
Hình 10: Màn hình LCD.....	8
Hình 11: Module I2C LCD 16x2.....	9
Hình 12: Module Relay.....	9
Hình 13: Động cơ điện.....	10
Hình 14: Đèn led.....	10
Hình 15: Giao diện của trình biên dịch.	11
Hình 16: Vùng viết chương trình biên dịch.....	11
Hình 17: Vùng thông báo Debug.....	12
Hình 18: Màn hình cài đặt thư viện.....	12
Hình 19: Sơ đồ mạch mô phỏng.....	14
Hình 20: Mạch mô phỏng khi nhiệt độ trên 27°C.....	15
Hình 21: Mạch mô phỏng khi nhiệt độ dưới 27°C.....	15
Hình 22: Đồng hồ Elenco M1250K.....	16
Hình 23: Sơ đồ mạch thang đo DC mA	16
Hình 24: Sơ đồ mô phỏng mạch thang đo DC mA.....	17
Danh mục bảng.....	iv
Bảng 1: Các chân của màn hình LCD... ..	8
Bảng 2: Các ICON chức năng thường dùng.....	11

PHẦN 1: THIẾT KẾ MẠCH- SỬ DỤNG CẢM BIẾN NHIỆT ĐỘ

1.1.TỔNG QUAN VỀ CẢM BIẾN NHIỆT ĐỘ

1.1.1.Khái niệm cảm biến DHT11

DHT11 là một loại cảm biến đo nhiệt độ và độ ẩm trong không khí, được sử dụng phổ biến trong các ứng dụng IoT và đo lường môi trường. DHT11 có thể đo nhiệt độ từ 0 đến 50 độ C và độ ẩm từ 20% đến 90%. Cảm biến này sử dụng một giao tiếp số đơn giản để truyền dữ liệu đo về nhiệt độ và độ ẩm về vi xử lý hoặc các thiết bị điều khiển khác thông qua các chân kết nối. Nó thường được sử dụng trong các ứng dụng đo lường môi trường như đo nhiệt độ và độ ẩm trong phòng, trong các trạm thời tiết, các ứng dụng tự động hóa nhà thông minh và các ứng dụng IoT khác.

Thiết bị này được tích hợp cảm biến thông minh, nên có tác dụng nhận biết các biến đổi từ môi trường. Từ đó, giúp người dùng điều chỉnh nhiệt độ phù hợp. Các thiết bị này thường được lắp đặt trên tường cố định tại một vị trí.



Hình 1: Cảm biến DHT11

Cảm biến DHT11 mang đến rất nhiều lợi ích cho chúng ta. Nó có thể thay thế cho sức người, tạo ra sự tiện lợi về các thiết bị điện, điện tử. Dưới đây là một số ứng dụng phổ biến của dòng thiết bị này:

- Đo nhiệt độ và độ ẩm
- Trạm thời tiết cục bộ
- Kiểm soát khí hậu tự động
- Giám sát môi trường

1.1.2.Các loại cảm biến

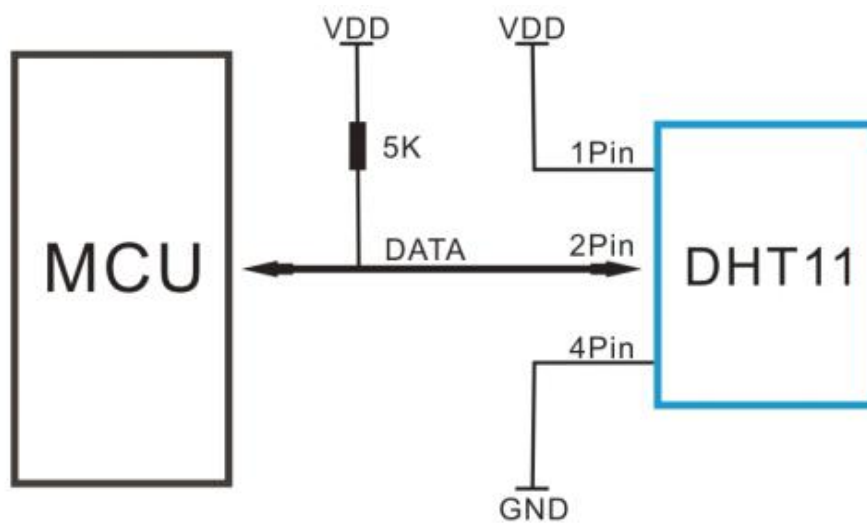
Có nhiều loại cảm biến nhiệt độ và độ ẩm khác nhau, mỗi loại có tính năng và ứng dụng riêng. Dưới đây là một số loại cảm biến nhiệt độ và độ ẩm phổ biến:

- DHT11/DHT22: Cảm biến đo nhiệt độ và độ ẩm trong không khí, được sử dụng rộng rãi trong các ứng dụng IoT và đo lường môi trường.
- LM35/LM75: Cảm biến đo nhiệt độ, thường được sử dụng trong các ứng dụng điều khiển nhiệt độ, máy tính, thiết bị y tế và xe hơi.
- DS18B20: Cảm biến nhiệt độ kỹ thuật số, được sử dụng trong các ứng dụng điều khiển nhiệt độ, điều khiển quạt, máy tính, thiết bị y tế và các thiết bị điện tử khác.

- SHT21: Cảm biến đo nhiệt độ và độ ẩm trong không khí, với độ chính xác cao, được sử dụng trong các ứng dụng đo lường môi trường, điều khiển nhiệt độ và độ ẩm.
- HTU21D: Cảm biến đo nhiệt độ và độ ẩm trong không khí, với độ chính xác cao và tiêu thụ điện năng thấp, được sử dụng trong các ứng dụng IoT và đo lường môi trường.
- BME280: Cảm biến đo nhiệt độ, độ ẩm và áp suất, với độ chính xác cao và tiêu thụ điện năng thấp, được sử dụng trong các ứng dụng đo lường môi trường, thiết bị di động và IoT.
- AM2302: Cảm biến đo nhiệt độ và độ ẩm trong không khí, tương tự như DHT11/DHT22, với độ chính xác cao và tiêu thụ điện năng thấp.

Từ các loại cảm biến trên, ta chọn sử dụng **cảm biến DHT11**.

1.1.3. Nguyên lý đo nhiệt độ của cảm biến

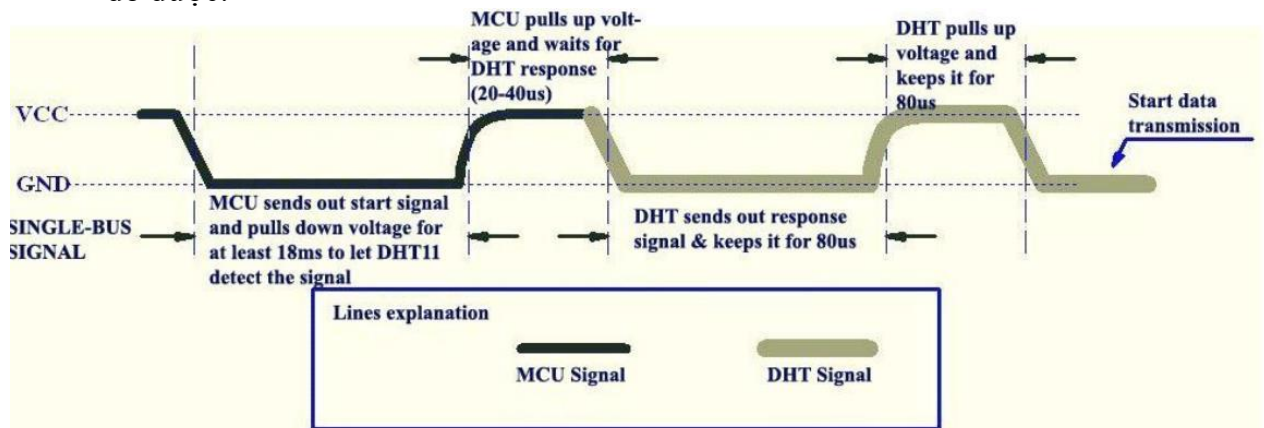


Hình 2: Sơ đồ kết nối vi xử lý

Nguyên lý hoạt động :

Để có thể giao tiếp với DHT11 theo chuẩn 1 chân vi xử lý thực hiện theo 2 bước:

- Gửi tín hiệu muốn đo (Start) tới DHT11, sau đó DHT11 xác nhận lại.
- Khi đã giao tiếp được với DHT11, Cảm biến sẽ gửi lại 5 byte dữ liệu và nhiệt độ đo được.



Hình 3: Sơ đồ nguyên lý hoạt động

❖ Bước 1: Gửi tín hiệu start

MCU thiết lập chân DATA là Output, kéo chân DATA xuống 0 trong khoảng thời gian $>18\text{ms}$.

Trong Code mình để 25ms. Khi đó DHT11 sẽ hiểu MCU muốn đo giá trị nhiệt độ và độ ẩm.

- MCU đưa chân DATA lên 1, sau đó thiết lập lại là chân đầu vào.
- Sau khoảng 20-40us, DHT11 sẽ kéo chân DATA xuống thấp. Nếu $>40\text{us}$ mà chân DATA không được kéo xuống thấp nghĩa là không giao tiếp được với DHT11.
- Chân DATA sẽ ở mức thấp 80us sau đó nó được DHT11 kéo lên cao trong 80us. Bằng việc giám sát chân DATA, MCU có thể biết được có giao tiếp được với DHT11 không. Nếu tín hiệu đo được DHT11 lên cao, khi đó hoàn thiện quá trình giao tiếp của MCU với DHT.

❖ Bước 2: Đọc giá trị trên DHT11

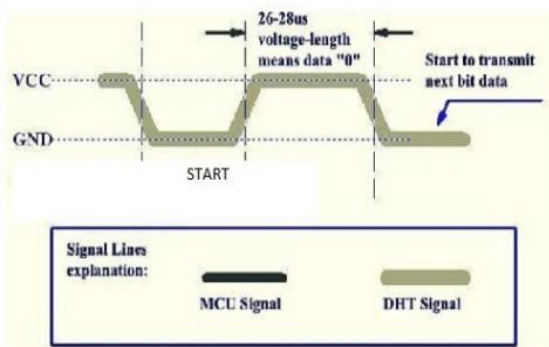
- DHT11 sẽ trả giá trị nhiệt độ và độ ẩm về dưới dạng 5 byte. Trong đó:
 - Byte 1: giá trị phần nguyên của độ ẩm (RH%)
 - Byte 2: giá trị phần thập phân của độ ẩm (RH%)
 - Byte 3: giá trị phần nguyên của nhiệt độ (TC)
 - Byte 4 : giá trị phần thập phân của nhiệt độ (TC)
 - Byte 5 : kiểm tra tổng.

Nếu Byte 5 = (8 bit) (Byte1 +Byte2 +Byte3 + Byte4) thì giá trị độ ẩm và nhiệt độ là chính xác, nếu sai thì kết quả đo không có nghĩa.

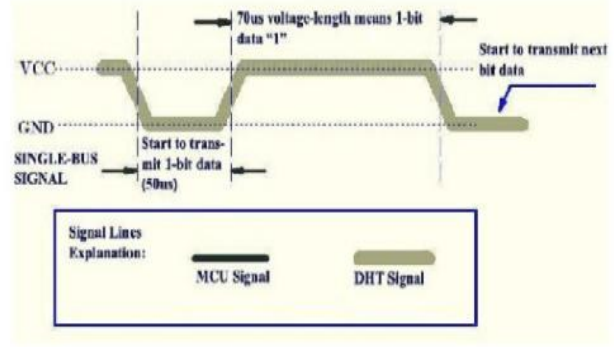
- Đọc dữ liệu : Sau khi giao tiếp được với DHT11, DHT11 sẽ gửi liên tiếp 40 bit 0 hoặc 1 về MCU, tương ứng chia thành 5 byte kết quả của Nhiệt độ và độ ẩm.

1.1.4.Tín hiệu vào, tín hiệu ra của cảm biến

Sau khi giao tiếp được với DHT11, DHT11 sẽ gửi liên tiếp 40 bit 0 hoặc 1 về MCU, tương ứng chia thành 5 byte kết quả của Nhiệt độ và độ ẩm.



Hình 4: Bit 0



Hình 5: Bit 1

Sau khi tín hiệu được đưa về 0, ta đợi chân DATA của MCU được DHT11 kéo lên 1. Nếu chân DATA là 1 trong khoảng 26-28 us thì là 0, còn nếu tồn tại 70us là 1. Do đó trong lập trình ta bắt sườn lên của chân DATA, sau đó delay 50us. Nếu giá trị đo được là 0 thì ta đọc được bit 0, nếu giá trị đo được là 1 thì giá trị đo được là 1. Cứ như thế ta đọc các bit tiếp theo.

1.1.5. Các thông số kỹ thuật chính của cảm biến

- Nguồn: 3 -> 5 VDC.
- Dòng sử dụng: 2.5mA max (khi truyền dữ liệu).
- Đo tốt ở độ ẩm 20-80% RH với sai số 5% RH.
- Đo tốt ở nhiệt độ 0°C ~ 50°C, sai số $\pm 2^\circ\text{C}$.
- Tần số lấy mẫu tối đa 1Hz (1 giây 1 lần).
- Kích thước 15mm x 12mm x 5.5mm.
- 4 chân, khoảng cách chân 0.1".
- Khoảng cách truyền tối đa: 20m.

1.1.6. Cách thức đọc dữ liệu

Để đọc dữ liệu từ cảm biến nhiệt độ và độ ẩm DHT11, bạn cần kết nối các chân của cảm biến với vi điều khiển hoặc mạch điện tử khác. Cảm biến này có 3 chân, bao gồm VCC, GND và Data.

Sau khi kết nối, để đọc dữ liệu từ cảm biến, bạn cần thực hiện các bước sau:

- Khởi động cảm biến: Trước khi đọc dữ liệu, bạn cần khởi động cảm biến bằng cách đưa tín hiệu cao (5V) đến chân VCC trong một thời gian ít nhất 18ms, sau đó đưa tín hiệu thấp (0V) đến chân Data trong ít nhất 20ms.
- Đọc dữ liệu: Sau khi cảm biến được khởi động, nó sẽ bắt đầu truyền dữ liệu đo nhiệt độ và độ ẩm thông qua chân Data. Dữ liệu được truyền dưới dạng tín hiệu số với tần số 1 bit/s và có tổng cộng 40 bit dữ liệu. Trong đó, 5 bit đầu tiên là dữ liệu đồng bộ, tiếp theo là 8 bit dữ liệu nhiệt độ, 8 bit dữ liệu độ ẩm, 8 bit kiểm tra chẵn lẻ và 8 bit cuối cùng để kết thúc truyền dữ liệu.
- Giải mã dữ liệu: Sau khi đọc dữ liệu, bạn cần giải mã các giá trị nhiệt độ và độ ẩm từ các bit dữ liệu đọc được. Để làm điều này, bạn có thể sử dụng một thư viện phần mềm hoặc viết mã đọc dữ liệu thủ công để giải mã các giá trị nhiệt độ và độ ẩm.

Tóm lại, để đọc dữ liệu từ cảm biến nhiệt độ và độ ẩm DHT11, bạn cần kết nối các chân của cảm biến với vi điều khiển hoặc mạch điện tử khác, sau đó thực hiện các bước khởi động cảm biến và đọc dữ liệu, cuối cùng là giải mã các giá trị nhiệt độ và độ ẩm.

1.2. THIẾT KẾ VI MẠCH ĐIỀU KHIỂN DÙNG ARDUINO

1.2.1. Tổng quan các thiết bị sử dụng trong hệ thống

1.2.1.1. Arduino

❖ Tổng quan Arduino

Arduino cơ bản là một mã nguồn mở về điện tử được tạo thành từ phần cứng và phần mềm. Về mặt kỹ thuật có thể coi Arduino là một bộ điều khiển logic có thể lập trình được. Đơn giản hơn, Arduino là thiết bị có thể tương tác với ngoại cảnh thông qua các cảm biến và hành vi được lập trình sẵn. Với thiết bị này việc lắp ráp và điều khiển các thiết bị điện tử sẽ dễ dàng hơn bao giờ hết.

Hiện tại có rất nhiều loại vi điều khiển và đa số được lập trình bằng ngôn ngữ C/C++ hoặc Assembly nên rất khó khăn cho những người có ít kiến thức sâu về điện tử và lập trình. Nó là trở ngại cho mọi người muốn tạo riêng cho mình một món đồ mang tính công nghệ. Song Arduino đã giải quyết được vấn đề này, Arduino được phát triển nhằm đơn giản hóa việc thiết kế, lắp ráp linh kiện điện tử cũng như lập trình trên vi điều khiển và mọi người có thể tiếp cận dễ dàng hơn với thiết bị điện tử mà không cần nhiều về kiến thức điện tử và thời gian.

Những thế mạnh của Arduino so với các nền tảng vi điều khiển khác:

- Chạy trên đa nền tảng: Việc lập trình Arduino có thể thực hiện trên các hệ điều hành khác nhau như Windows, Mac Os, Linux trên Desktop, Android trên di động.
- Ngôn ngữ lập trình đơn giản dễ hiểu.
- Mã nguồn mở: Arduino được phát triển dựa trên nguồn mở nên phần mềm chạy trên Arduino được chia sẻ dễ dàng và tích hợp vào các nền tảng khác nhau.
- Mở rộng phần cứng: Arduino được thiết kế và sử dụng theo dạng modul nên việc mở rộng phần cứng cũng dễ dàng hơn.
- Đơn giản và nhanh: Rất dễ dàng lắp ráp, lập trình và sử dụng thiết bị.
- Dễ dàng chia sẻ: Mọi người dễ dàng chia sẻ mã nguồn với nhau mà không lo lắng về ngôn ngữ hay hệ điều hành mình đang sử dụng.

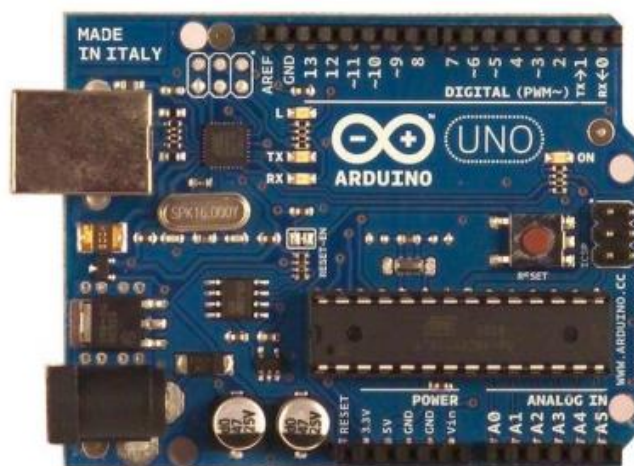
Arduino được chọn làm bộ não xử lý của rất nhiều thiết bị từ đơn giản đến phức tạp. Trong số đó có một vài ứng dụng thực sự chứng tỏ khả năng vượt trội của Arduino do chúng có khả năng thực hiện nhiều nhiệm vụ rất phức tạp.

Arduino được biết đến nhiều nhất là phần cứng của nó, nhưng phải có phần mềm để lập trình phần cứng. Cả phần cứng và phần mềm gọi chung là Arduino.

◆ Cấu trúc chung

Arduino Uno là một bo mạch vi điều khiển dựa trên chip ATmega168 hoặc ATmega328. Cấu trúc chung bao gồm:

- 14 chân vào ra bằng tín hiệu số, trong đó có 6 chân có thể sử dụng để điều chế độ rộng xung.
- Có 6 chân đầu vào tín hiệu tương tự cho phép chúng ta kết nối với các bộ cảm biến bên ngoài để thu thập số liệu.
- Sử dụng một dao động thạch anh tần số dao động 16MHz.
- Có một cổng kết nối bằng chuẩn USB để chúng ta nạp chương trình vào bo mạch và một chân cấp nguồn cho mạch, một nút reset.
- Nó chứa tất cả mọi thứ cần thiết để hỗ trợ các vi điều khiển, nguồn cung cấp cho Arduino có thể là từ máy tính thông qua cổng USB hoặc là từ bộ nguồn chuyên dụng được biến đổi từ xoay chiều sang một chiều hoặc là nguồn lấy từ pin.



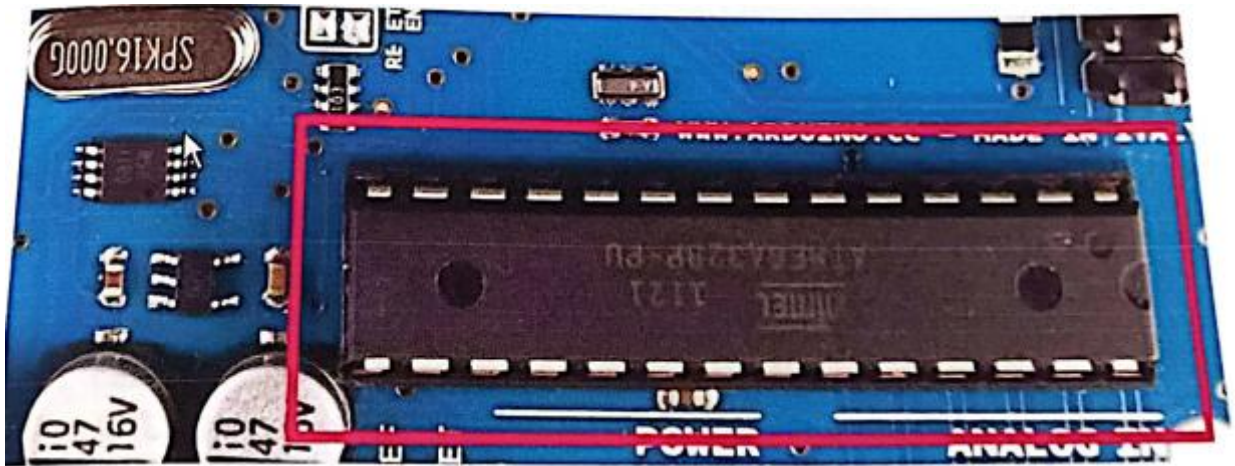
Hình 6: Cấu trúc phần cứng của Arduino Uno

◆ Thông số kỹ thuật của Arduino Uno

- Khối xử lý trung tâm là vi điều khiển Atmega328 họ 8bit.
- Điện áp hoạt động 5V.
- Điện áp đầu vào khuyến nghị là 5-12V.
- Điện áp đầu vào giới hạn 6-20V.
- Dòng điện một chiều trên các chân vào ra là 40mA.
- Dòng điện một chiều cho chân 3.3V là 50mA.
- Clock Speed 16 MHz.
- Flash Memory 16 Kb (ATmega 168) hoặc 32 Kb (ATmega 328), SRAM 1 Kb (ATmega 168) hoặc 2 Kb (ATmega 328), EEPROM 512 bytes (ATmega 168) hoặc 1 Kb (ATmega 328).

❖ Vi điều khiển

Arduino Uno R3 có thể sử dụng 3 vi điều khiển họ 8bit AVR là ATmega8, ATmega168, ATmega328. Bộ não này có thể xử lý những tác vụ đơn giản như điều khiển đèn LED nhấp nháy, xử lý tín hiệu cho xe điều khiển từ xa, ... hay những ứng dụng khác. Thiết kế tiêu chuẩn của Arduino Uno R3 sử dụng là ATmega328.

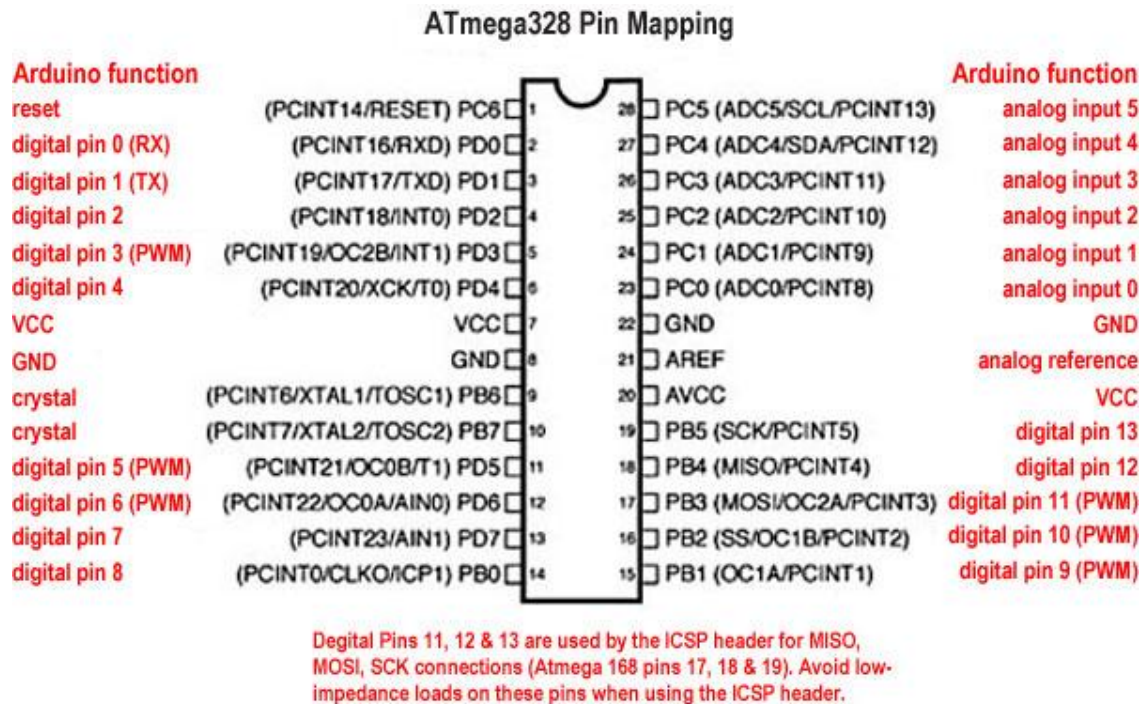


Hình 7: Vi điều khiển ATmega328

Vi điều khiển ATmega328 tiêu chuẩn cung cấp:

- **32KB bộ nhớ Flash:** những đoạn lệnh lập trình sẽ được lưu trữ trong bộ nhớ Flash của vi điều khiển. Thường thì sẽ có khoảng vài KB trong số này sẽ được dùng cho bootloader.
- **2KB cho SRAM:** giá trị các biến khai báo khi lập trình sẽ lưu ở đây. Khi khai báo càng nhiều biến thì càng cần nhiều bộ nhớ RAM. Tuy vậy, thực sự thì cũng hiếm khi nào bộ nhớ RAM lại trở thành thứ phải bận tâm. Khi mất điện, dữ liệu trên SRAM sẽ bị mất.
- **1KB cho EEPROM:** đây giống như một chiếc ổ cứng mini- nơi có thể đọc và ghi dữ liệu của mình vào đây mà không phải lo bị mất khi mất điện giống như dữ liệu trên SRAM.

Sơ đồ chân của vi điều khiển ATmega328P:



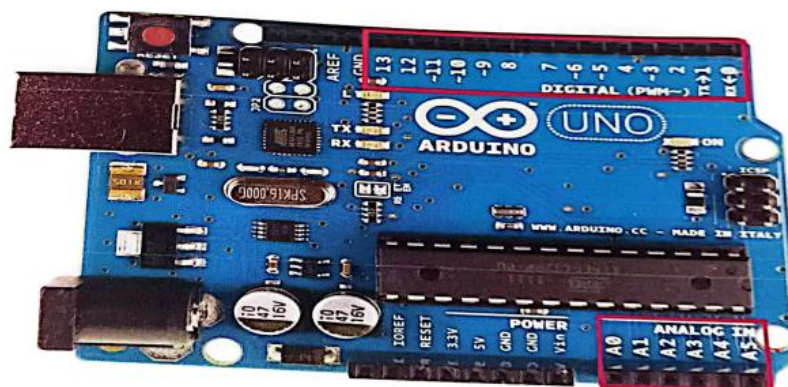
Hình 8: Sơ đồ chân của ATmega328P

❖ Các cổng vào ra

Arduino Uno có 14 chân digital dùng để đọc hoặc xuất tín hiệu. Chúng chỉ có 2 mức điện áp là 0V và 5V với dòng vào /ra tối đa trên mỗi chân là 40mA. Ở mỗi chân đều có các điện trở pull-up từ được cài đặt ngay trong vi điều khiển ATmega328 (mặc định thì các điện trở này không được kết nối).

Arduino có 6 chân analog (A0 đến A5) cung cấp độ phân giải tín hiệu 10 bit (0 đến $2^8 - 1$) để đọc giá trị điện áp trong khoảng 0V-5V. Với chân AREF trên board, có thể đưa vào điện áp tham chiếu khi sử dụng các chân analog. Tức là nếu cấp điện áp 2.5V vào chân này thì có thể dùng các chân analog để đo điện áp trong khoảng từ 0V đến 2.5V với độ phân giải vẫn là 10bit.

Đặc biệt, Arduino Uno có 2 chân A4 (SDA) và A5 (SCL) hỗ trợ giao tiếp I2C/TWI với các thiết bị khác.



Hình 9: Các chân vào /ra của Arduino Uno

1.2.1.2.Cảm biến DHT11 (đã được tìm hiểu ở phần 1.1)

1.2.1.3.Màn hình LCD LM016L

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, nối với VCC = 5V của mạch điều khiển
Chân số 3 - VEE	Điều chỉnh độ tương phản của LCD
Chân số 4 - RS	Chân chọn thanh ghi, được nối với logic “0” hoặc logic “1”
Chân số 5 – R/W	Chân chọn chế độ đọc/ghi, được nối với logic “0” để ghi hoặc logic “1” để đọc
Chân số 6 - E	Chân cho phép, sau khi cái tín hiệu đã được đặt lên Bus
DB0 – DB7	-Logic “0”: Bus DB0 – DB7 sẽ nối với thanh ghi lệnh IR của LCD (ở chế độ ghi) hoặc nối với bộ đếm địa chỉ của LCD(ở chế độ đọc) -Logic “1”: Bus DB0 – DB7 sẽ nối với thanh ghi dữ liệu DR bên trong LCD
Chân 15 - A	Cực dương Led nền
Chân 16 - K	Cực âm Led nền

Bảng 1:Các chân của màn hình LCD



Hình 10 :Màn hình LCD

1.2.1.4.Module giao tiếp I2C

- LCD có quá nhiều chân gây khó khăn trong quá trình đấu nối và chiếm dụng nhiều chân trên vi điều khiển.
- Module I2C ra đời để giải quyết vấn đề này.

- Thay vì phải sử dụng 6 chân vi điều khiển để kết nối với LCD 16x2(RS, EN, D7, D6, D5 và D4) thì module I2C chỉ cần tốn 2 chân (SCL, SDA) để kết nối.
- Module I2C hỗ trợ các loại LCD sử dụng driver HD44780(LCD 16x2, LCD 20x4,...) và tương thích với hầu hết các vi điều khiển hiện nay.



Hình 11: Module I2C LCD 16x2

1.2.1.5.Module Relay

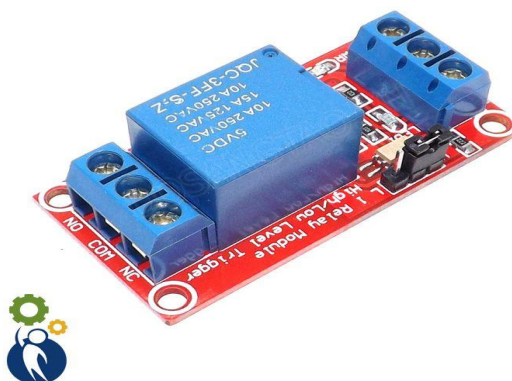
Module Relay 1 kênh 5V gồm 1 rơ le điện áp hoạt động ở mức 5VDC, đầu ra điều khiển hiệu điện tối đa ở mức 250V 10A đối với điện áp xoay chiều AC và 30V với điện áp 1 chiều DC.

Module relay 1 kênh nhỏ gọn chuyên nghiệp, khả năng chống nhiễu tốt và khả năng cách điện tốt. Trong module đã có sẵn mạch kích relay sử dụng IC cách ly quang và transistor giúp cách ly hoàn toàn mạch vi điều khiển với rơ le bảo đảm vi điều khiển hoạt động ổn định.

Có sẵn header rất tiện dụng khi kết nối với vi điều khiển. Có các lỗ bắt vít rất tiện lợi để lắp đặt trong hệ thống mạch.

Mạch điều khiển relay 1 kênh này sử dụng chân kích mức Thấp (0V), khi có tín hiệu 0V vào chân IN thì relay sẽ nhảy qua thường Mở của Relay.

Ứng dụng với chân relay module khá nhiều bao gồm cả điện DC hay AC.



Hình 12:Module Relay

Thông số kỹ thuật:

- Kích thước: 53mm (chiều dài) * 28.3mm (chiều rộng) * 19.3mm (H)
- Trọng lượng: 18g
- Màu sắc: Đen

- Bốn lỗ để bắt vít cố định có đường kính 3.1mm, dễ dàng lắp đặt trong hệ thống mạch
- Opto cô lập, tốt chống nhiễu
- Có đèn báo đóng ngắt trên Relay.
- Sử dụng điện áp nuôi DC 5V.
- Đầu ra điện thế đóng ngắt tối đa: DC 30V / 10A, AC 250V / 10A

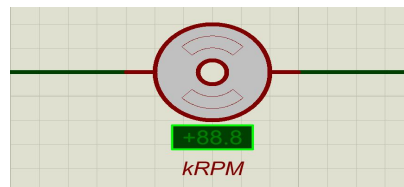
Cách đấu nối dây:

- VCC: nối với nguồn 5v
- GND: nối với 0V
- IN: nối với dây tín hiệu điều khiển relay

1.2.1.6. Động cơ Motor DC

Động cơ DC (Direct Current Motors) hay còn gọi là động cơ điện một chiều chính là động cơ được điều khiển bằng dòng có hướng xác định. Cũng có thể nói dễ hiểu hơn thì đây là một loại động cơ chạy bằng nguồn điện áp DC – điện áp 1 chiều

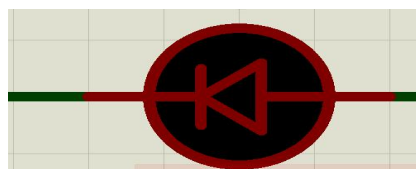
Bảng xanh: là đồng hồ đo vòng tua RPM với đơn vị là kRPM. RPM (Round Per Minute - vòng/phút) là đơn vị của vận tốc quay hoặc tần số quay xung quanh một trục cố định.



Hình 13: Động cơ điện

1.2.1.7 Đèn LED

Đèn LED, hay còn gọi bóng đèn LED (tiếng Anh: LED lamp), là đèn điện được sử dụng trong các thiết bị chiếu sáng, tạo ra ánh sáng bằng cách sử dụng một hoặc nhiều diode phát quang (LED).

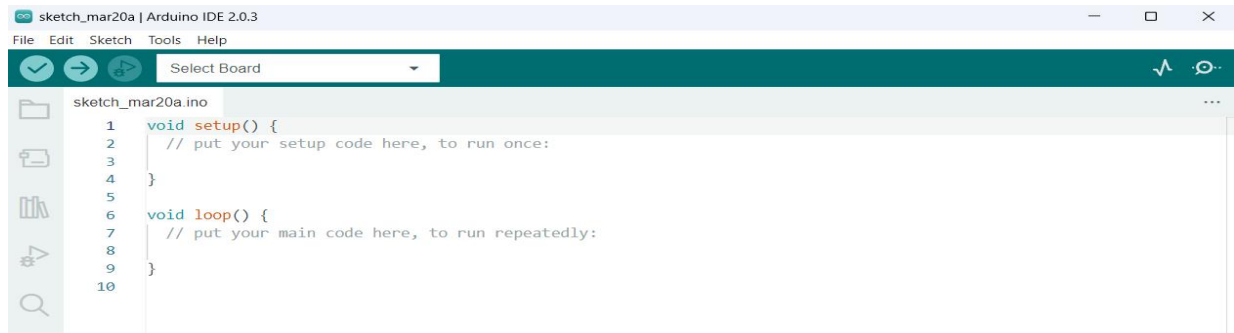


Hình 14: Đèn led

1.2.2. Phần mềm lập trình Arduino IDE cho Arduino Uno

Phần mềm Arduino IDE là trình biên dịch hỗ trợ soạn thảo và nạp code xuống các bản mạch như Arduino Uno R3, Arduino Atemega 2560, Arduino Erthernet, ESP8266...







Arduino ide được tạo ra bởi ngôn ngữ lập trình Java chính là ứng dụng đa nền tảng(cross-platform). Ngôn ngữ lập trình sử dụng cho các chương trình trong Arduino là C hoặc C++. Bản thân Arduino IDE đã được trang bị một thư viện phần mềm thường gọi là “wiring”, từ các chương trình “wiring” gốc đó sẽ giúp thực hiện thao tác lập trình dễ dàng và nhanh hơn. Một chương trình chạy trong Arduino được gọi là một sketch, chương trình được định dạng dưới dạng .ino.



Hình 15: Giao diện của trình biên dịch

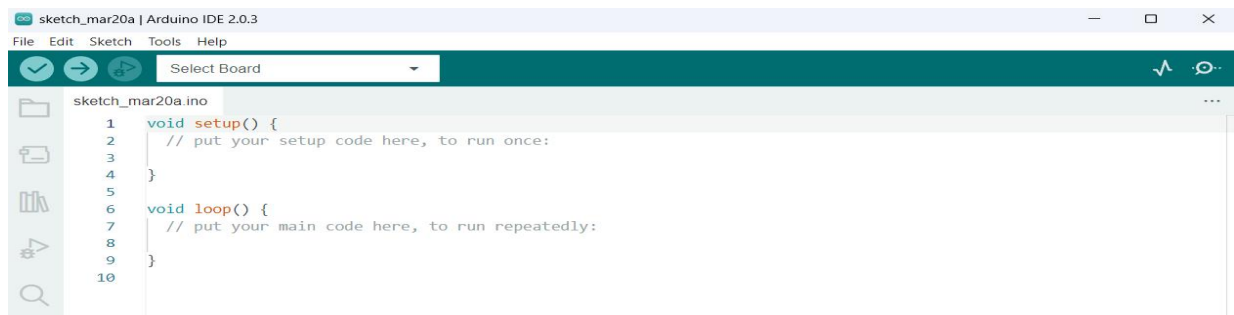
Về cơ bản thì trình biên dịch gồm 3 vùng cơ bản để người lập trình có thể thao tác và xử lý thông tin từ các vùng đó.

Vùng lệnh: Bao gồm các nút lệnh menu(File, Edit, Sketch, Tools, Help). Phía dưới là các icon cho phép sử dụng nhanh các chức năng thường dùng của IDE được miêu tả như sau:

Icon	Chức năng
	Biên dịch chương trình đang soạn thảo để kiểm tra các lỗi lập trình.
	Biên dịch và upload chương trình đang soạn thảo.
	Mở một trang soạn thảo mới.
	Mở các chương trình đã lưu.
	Lưu chương trình soạn thảo.
	Mở cửa sổ Serial Monitor để gửi và nhận dữ liệu giữa máy tính và board Arduino

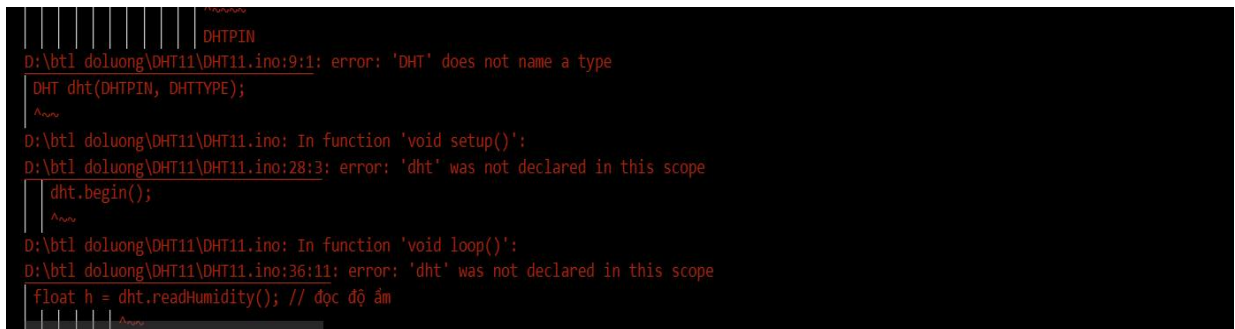
Bảng 2: Các ICON chức năng thường dùng

Vùng viết chương trình: Đây là vùng để ta viết những đoạn mã chương trình. Tên chương trình được hiển thị ngay dưới các Icon, ở đây nó tên là “Blink”. Để ý rằng phía sau tên chương trình có một dấu “§”. Điều đó có nghĩa là đoạn chương trình chưa được lưu lại.



Hình 16: Vùng viết chương trình biên dịch

Vùng thông báo (debug): Những thông báo từ IDE sẽ được hiển thị tại đây. Để ý rằng góc dưới cùng bên phải hiển thị loại board Arduino và cổng COM được sử dụng. Luôn chú ý tới mục này bởi nếu chọn sai loại board hoặc cổng COM sẽ không thể upload được chương trình lập trình.



```

D:\btl doluong\DHT11\DHT11.ino:9:1: error: 'DHT' does not name a type
  DHT dht(DHTPIN, DHTTYPE);
  ^
D:\btl doluong\DHT11\DHT11.ino: In function 'void setup()':
D:\btl doluong\DHT11\DHT11.ino:28:3: error: 'dht' was not declared in this scope
  dht.begin();
  ^
D:\btl doluong\DHT11\DHT11.ino: In function 'void loop()':
D:\btl doluong\DHT11\DHT11.ino:36:11: error: 'dht' was not declared in this scope
  float h = dht.readHumidity(); // đọc độ ẩm
  ^
  
```

Hình 17: Vùng thông báo Debug

1.2.3. Code Arduino

Để sử dụng sensor DHT11 giao tiếp với Arduino thì ta cần cài đặt thư viện DHT_Sensor:



Hình 18: Màn hình cài đặt thư viện

- Mở phần mềm Arduino → Sketch → Include Library → Add .zip Library → chọn tới file zip DHT-sensor.zip.
- Tương tự ta add tiếp → file zip Adafruit_Sensor-master.zip.
- Sau khi add thư viện nên thoát khỏi phần mềm Arduino rồi mở lại.
- Gọi thư viện DHT Sensor , LiquidCrystal_I2C.
- Mở Phần mềm Arduino → Sketch → Include Library → Kéo xuống dưới chọn DHT_Sensor.
- Mở Phần mềm Arduino → Sketch → Include Library → Kéo xuống dưới chọn LiquidCrystal_I2C.

LiquidCrystal_I2C lcd(0x27,16,2); // Khai báo LCD

- Đặt địa chỉ LCD là 0x27 cho màn hình LCD 16x2.
- 16 là số cột của màn hình.
- 2 là số dòng của màn hình.

Từ các dữ liệu và kiến thức trên, ta có mã nguồn Arduino:

```
#include <DHT.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27,16,2);
const int DHTPIN = 2;
const int DHTTYPE = DHT11;
const int FAN = 7;
const int LED = 12;
DHT dht(DHTPIN, DHTTYPE);
byte degree[8]={
  0B01110,
  0B01010,
  0B01110,
  0B00000,
  0B00000,
  0B00000,
  0B00000,
  0B00000
};

void setup()
{
  lcd.init();
  lcd.backlight();
  lcd.print("Nhiệt Độ :");
  lcd.setCursor(0,1);///Đưa con trỏ tới hàng 1, cột 0
  lcd.print("Độ Am :");
  lcd.createChar(1, degree);
  dht.begin();
  pinMode (DHTPIN,INPUT);
  pinMode (FAN,OUTPUT);
  pinMode (LED,OUTPUT);
}

void loop()
{
  // put your main code here, to run repeatedly:
  float h = dht.readHumidity(); // đọc độ ẩm
  float t = dht.readTemperature(); //đọc nhiệt độ
  //lcd.clear();
  lcd.setCursor(10, 0);
  lcd.print(round(t));
  lcd.print(" ");
  lcd.write(1);
  lcd.print("C");

  lcd.setCursor(10, 1);
  lcd.print(round(h));
```

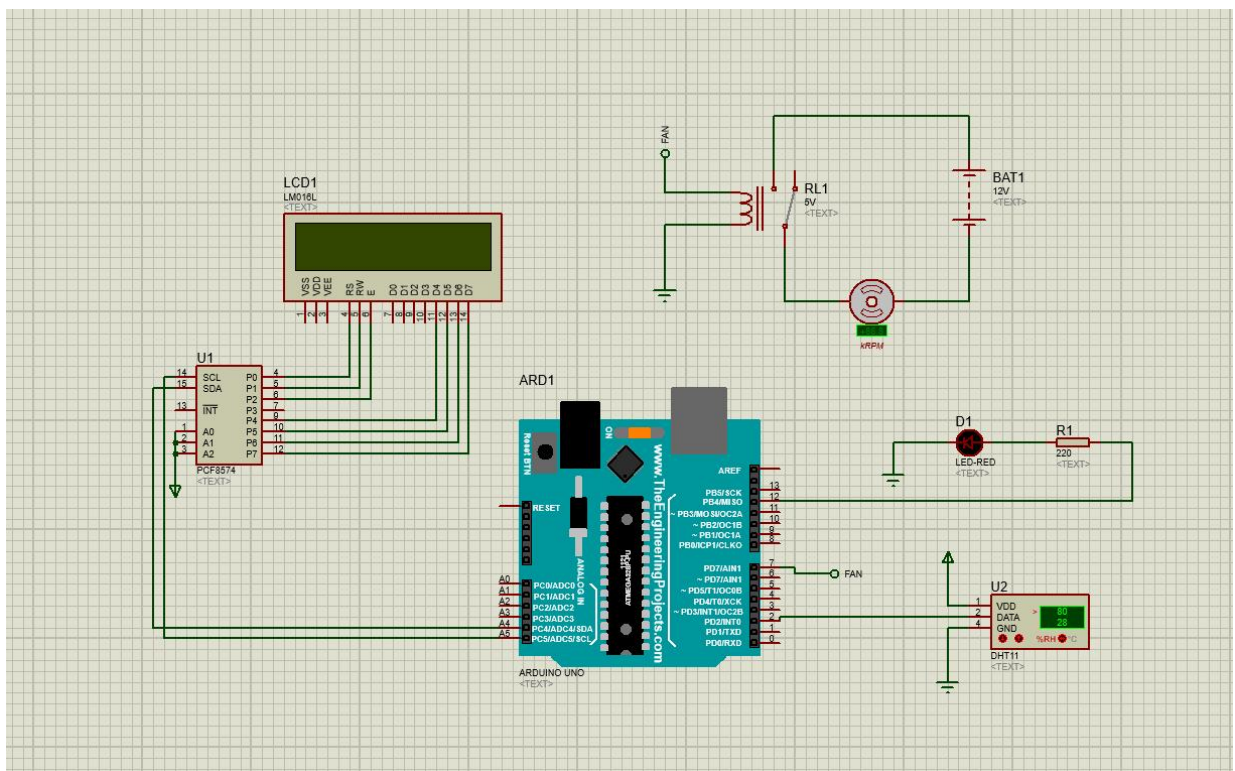
```

lcd.print(" %");

if (round(t)>27){
    digitalWrite(FAN,HIGH);
    digitalWrite(LED,HIGH);
}else{
    digitalWrite(FAN,LOW);
    digitalWrite(LED,LOW);
}
}
}

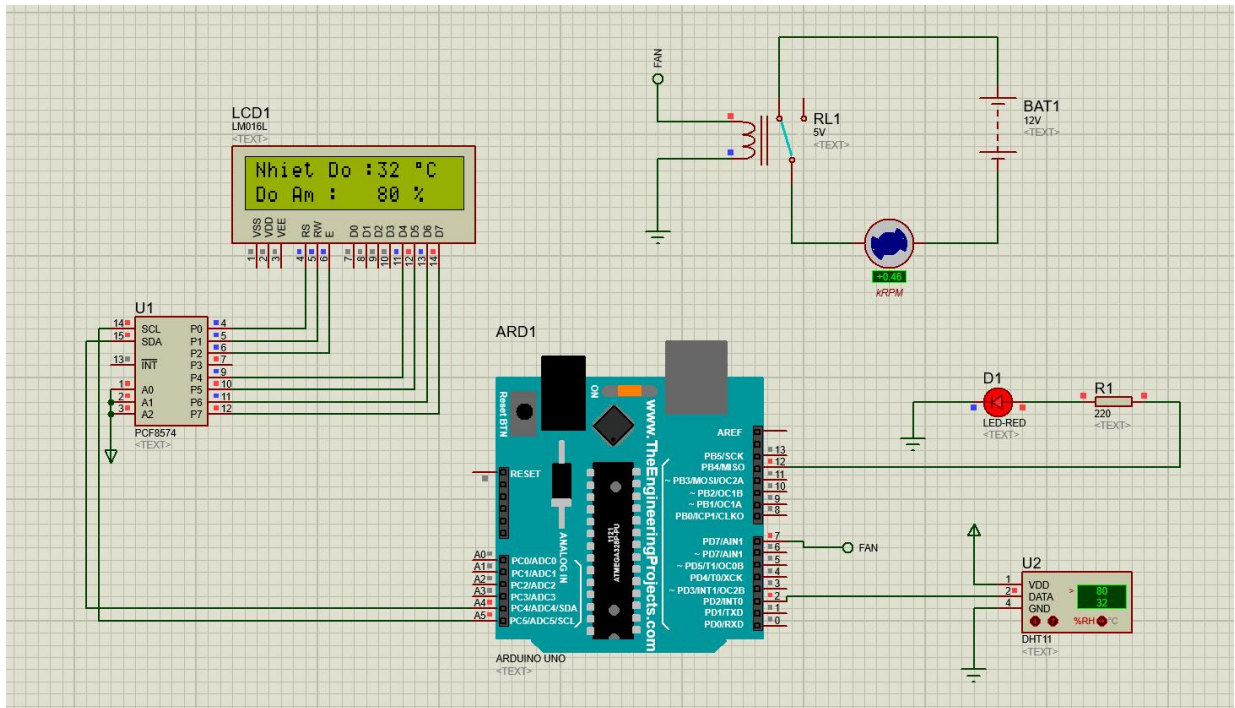
```

1.2.4. Sơ đồ mạch mô phỏng

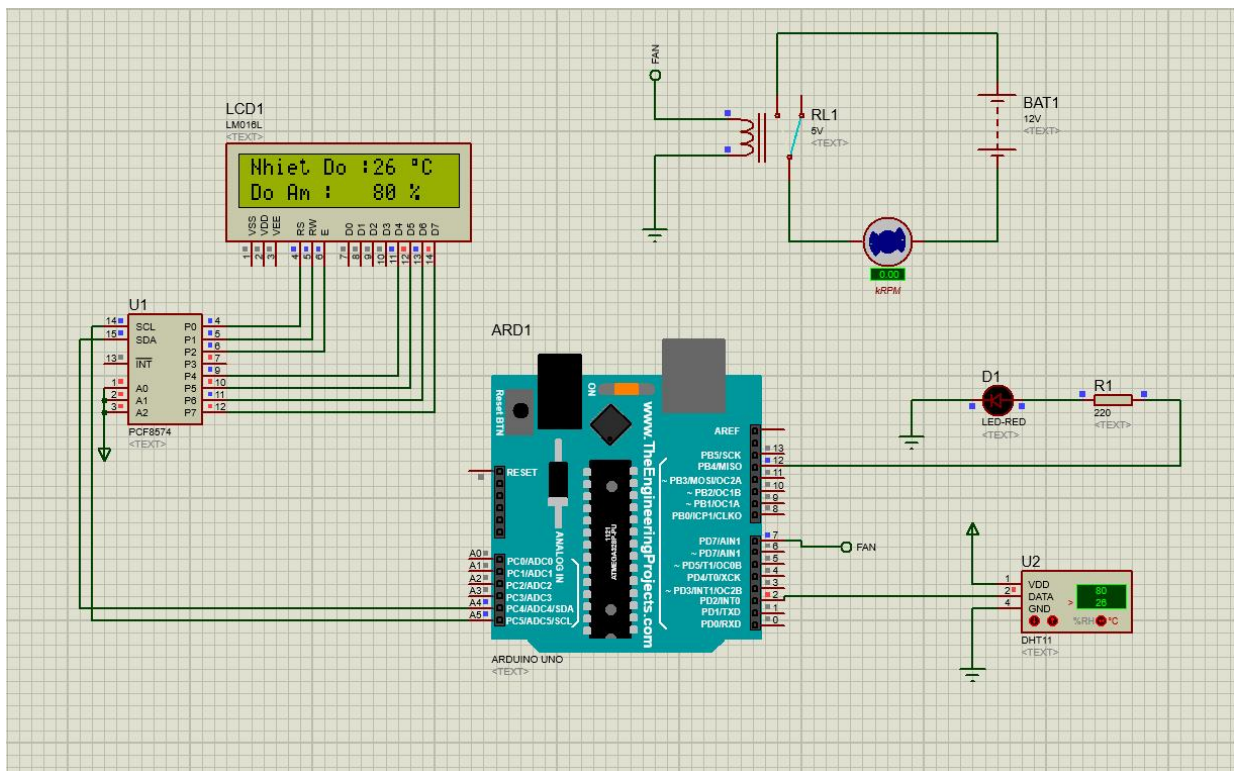


Hình 19: Sơ đồ mạch mô phỏng

1.2.5. Kết quả



Hình 20: Mạch mô phỏng khi nhiệt độ trên 27°C



Hình 21: Mạch mô phỏng khi nhiệt độ dưới 27°C

PHẦN 2: VẼ SƠ ĐỒ MẠCH THANG ĐO DC mA CỦA ĐỒNG HỒ ELENCO, MODEL M1250K

2.1. Đồng hồ Elenco, model M1250K

Đồng hồ Elenco M1250K là một loại đồng hồ đo dòng điện một chiều (DC) và dòng điện xoay chiều (AC). Đồng hồ này có thể đo các giá trị dòng điện từ 0 đến 10A và có độ chính xác cao.

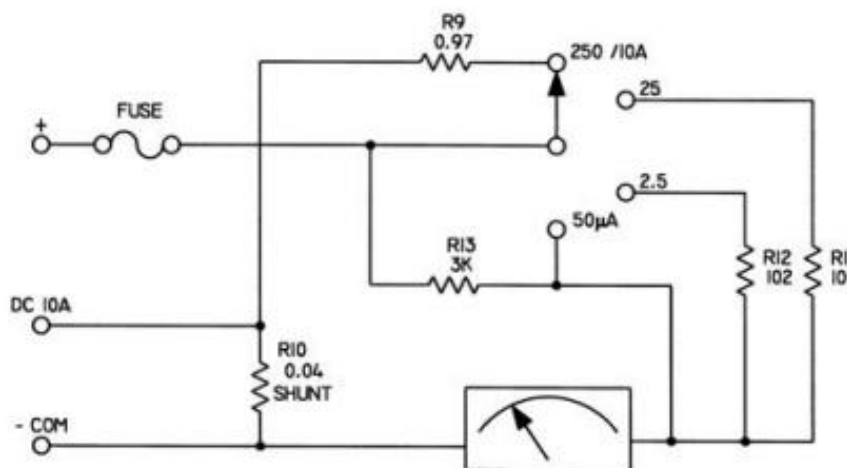
Đồng hồ Elenco M1250K có thiết kế đơn giản với một kim chỉ và một màn hình hiển thị số, giúp người sử dụng dễ dàng đọc và hiểu các giá trị đo được. Nó cũng có tính năng tự động điều chỉnh dải đo và bảo vệ quá tải, giúp bảo vệ linh kiện khỏi bị hư hỏng do quá tải.



Hình 22: Đồng hồ Elenco M1250K

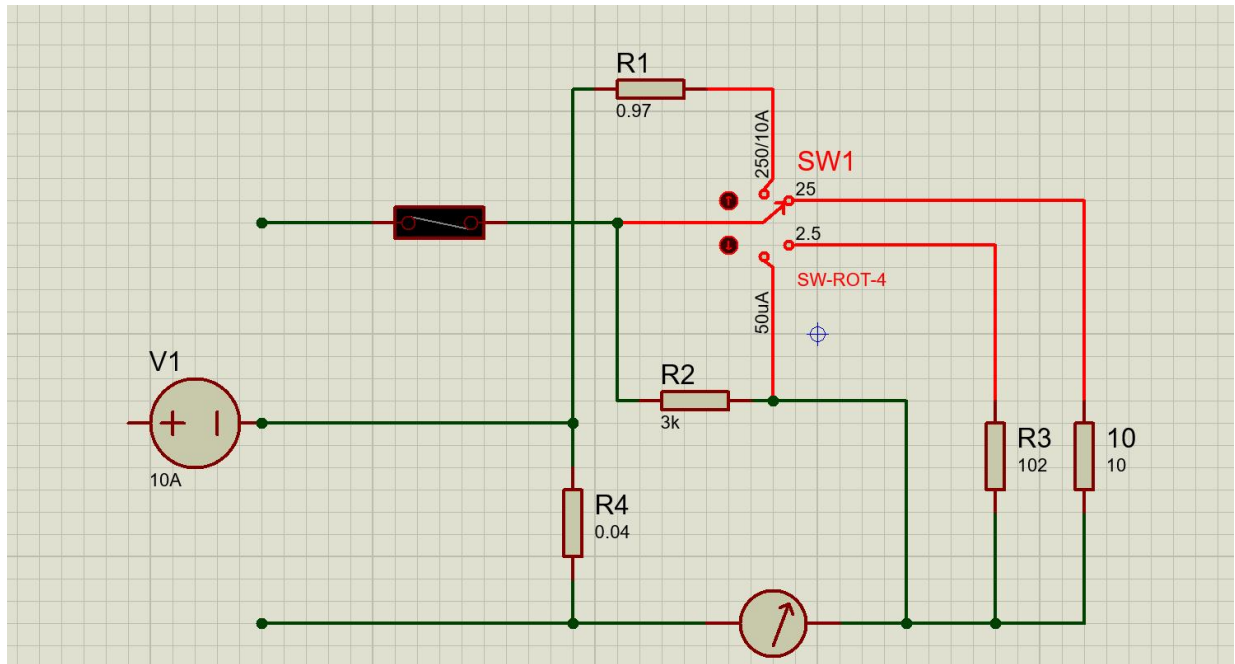
2.2. Sơ đồ mạch thang đo DC mA

Trên dải 0.01mA, dòng điện được đưa trực tiếp vào đồng hồ và điện áp rơi trên đồng hồ ở độ lệch toàn thang đo là 0,1V. Trên tất cả các dải khác, điện áp rơi trên toàn thang đo đồng hồ là 0,25V. Mạch được lắp thêm một cầu chì 0.5A để tránh quá tải. Sơ đồ mạch đơn giản của mạch đo dòng điện một chiều (DC) bao gồm 5 điện trở R1, R2, R3, R4, điện trở shunt và một cầu chì.



Hình 23: Sơ đồ mạch thang đo DC mA

2.3. Sơ đồ mô phỏng trên Proteus



Hình 24: Sơ đồ mô phỏng mạch thang đo DC mA

TÀI LIỆU THAM KHẢO

- [1].Giáo trình Kỹ thuật đo lường_Trường đại học Giao thông vận tải
- [2].Slide bài giảng Kỹ thuật đo lường_Bộ môn Kỹ thuật điện tử
- [3].Đồng hồ Elenco
<https://www.elenco.com/wp-content/uploads/2017/10/M1250K-3.pdf>
- [4].Arduino
<http://arduino.vn/>
- [5].Cảm biến DHT11