

4

HỆ THỐNG NHÚNG
MẠNG KHÔNG DÂY

Làm quen với Arduino

Biên soạn: Nguyễn Văn Bảo
Liên hệ: baonv@uit.edu.vn

Lưu hành nội bộ

A. TỔNG QUAN

1. Mục tiêu

- Giới thiệu một số khái niệm cơ bản trong lập trình nhúng.
- Giới thiệu một số thiết bị như Arduino, đèn LED, cảm biến, ...
- Sử dụng được các thiết bị đã giới thiệu để lập trình tạo các kịch bản.

2. Môi trường thực hành

- Arduino IDE.

B. GIỚI THIỆU

1. Các khái niệm cơ bản

a. Input và Output

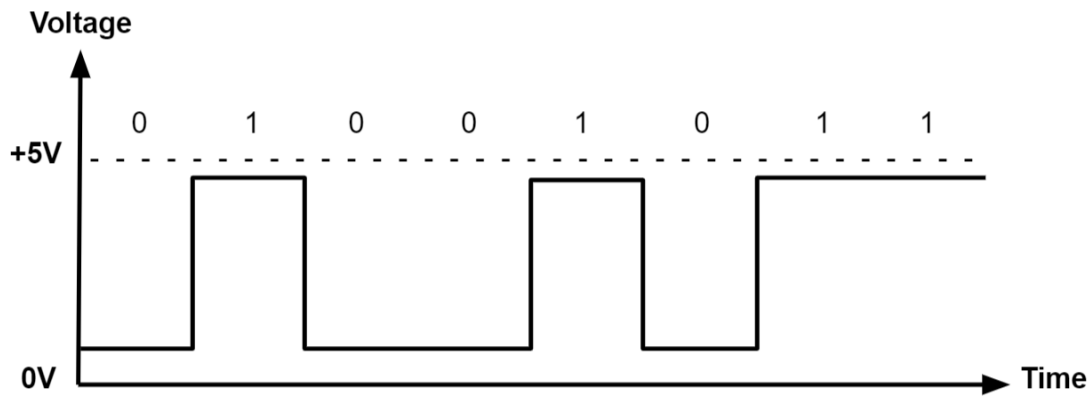
Input: những tín hiệu hoặc thông tin đi vào bo mạch. Ví dụ như tín hiệu từ một nút bấm, tín hiệu từ cảm biến nhiệt độ, cảm biến ánh sáng,...

Output: các tín hiệu xuất ra ngoài bo mạch. Ví dụ như điều khiển đèn, điều khiển động cơ, relay,...

b. Digital và Analog

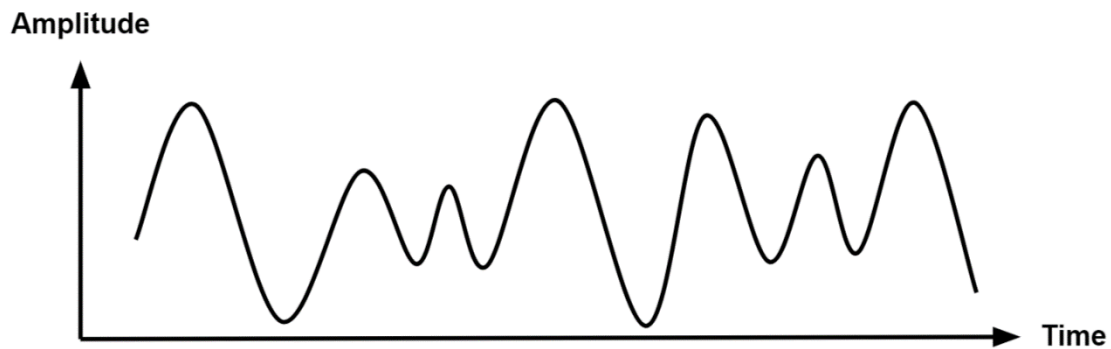
Digital (Tín hiệu kỹ thuật số): tín hiệu digital là tín hiệu biểu thị dữ liệu dưới dạng một chuỗi các giá trị rời rạc. Một tín hiệu digital chỉ có thể nhận một giá trị từ một tập hợp hữu hạn các giá trị có thể có tại một thời điểm nhất định.

Tín hiệu digital được sử dụng trong tất cả các thiết bị điện tử kỹ thuật số, bao gồm các thiết bị tính toán và thiết bị truyền dữ liệu. Khi đó trên đồ thị điện áp so với thời gian, tín hiệu digital là một trong hai giá trị và thường nằm trong khoảng 0V và VCC (thường là 3.3V hoặc 5V).

*Hình 1. Tín hiệu digital*

Analog (tín hiệu tương tự): tín hiệu analog là tín hiệu thay đổi theo thời gian và thường bị ràng buộc trong một phạm vi (ví dụ: 0 - 5V), nhưng có vô số giá trị trong phạm vi liên tục đó.

Tín hiệu analog thường là phản hồi về sự thay đổi ánh sáng, âm thanh, nhiệt độ hoặc các hiện tượng vật lý khác. Khi biểu thị trên đồ thị điện áp so với thời gian, tín hiệu analog sẽ tạo ra một đường cong mượt mà và liên tục.

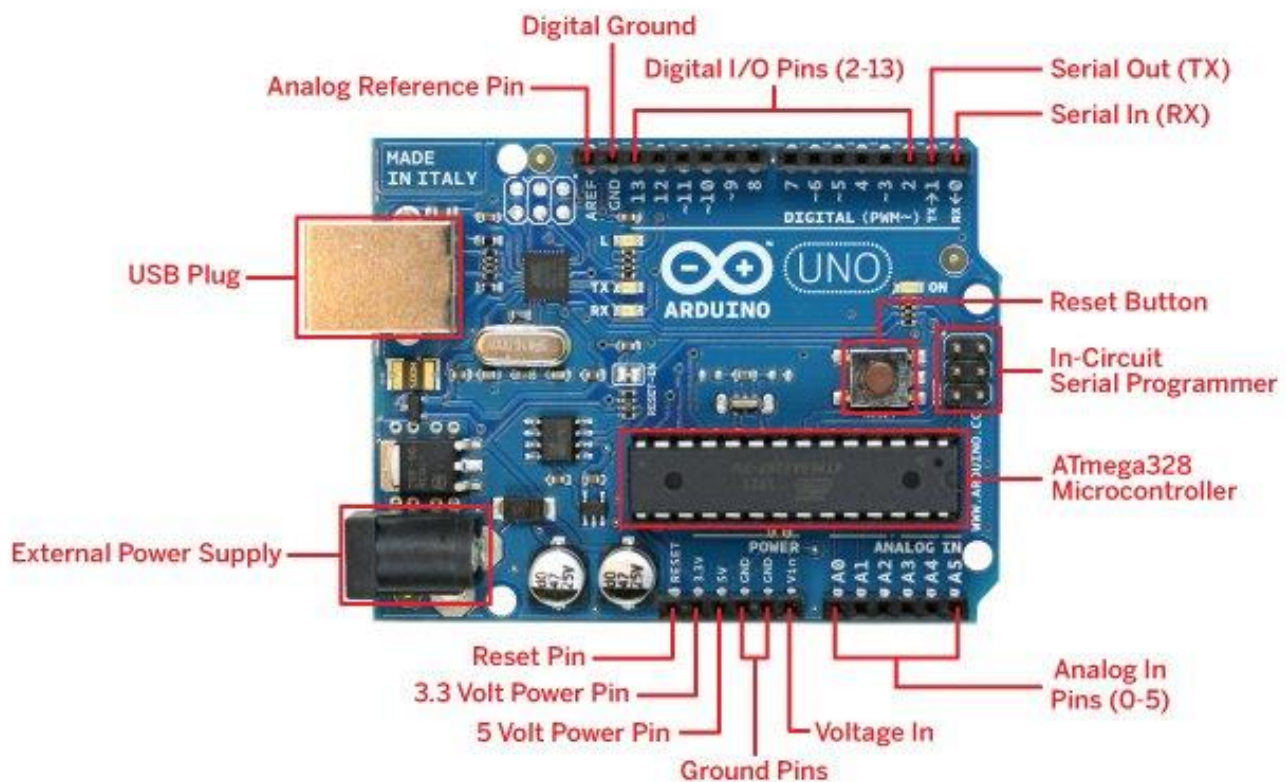
*Hình 2. Tín hiệu analog*

2. Nền tảng Arduino

Arduino là một nền tảng điện tử mã nguồn mở để sử dụng bao gồm phần cứng và phần mềm. Bo mạch Arduino có thể làm từ các việc đơn giản như bật tắt một bóng đèn, đọc các giá trị cảm biến cho tới những việc phức tạp như robot, bộ điều khiển các thiết bị trong nhà,...

Trong nhiều năm qua, Arduino đã trở thành đầu não của hàng nghìn dự án, từ những vật dụng hàng ngày cho đến những công cụ khoa học phức tạp. Một cộng đồng trên toàn thế giới bao gồm các nhà sản xuất, sinh viên, lập trình viên và các chuyên gia đã tập hợp xung quanh nền tảng này.

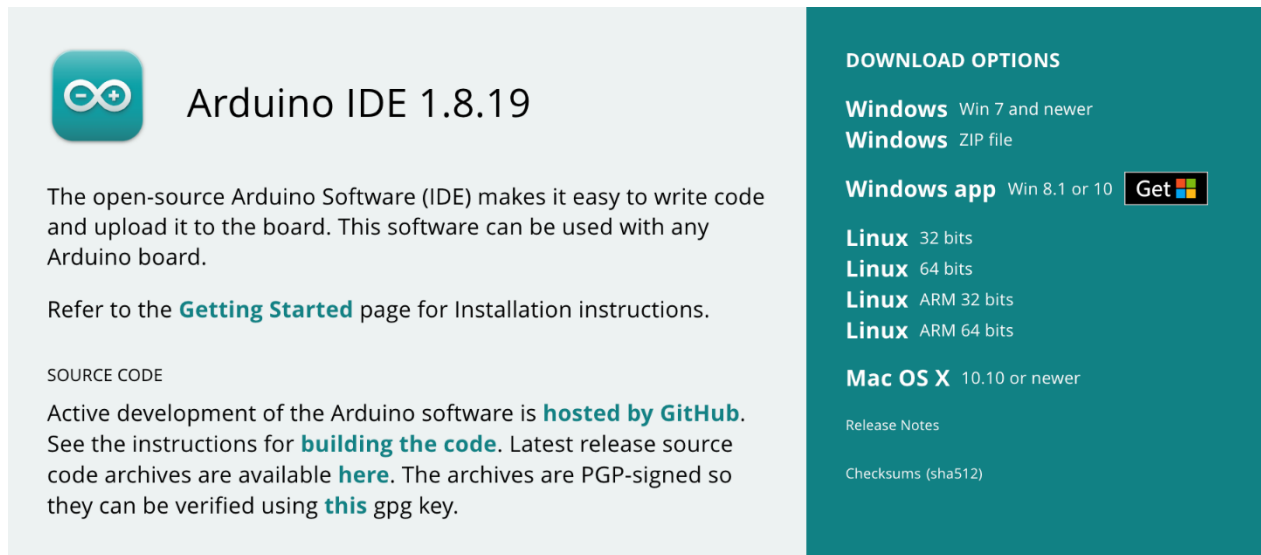
Việc học Arduino cũng không quá phức tạp, không cần nhiều kiến thức chuyên sâu về vi điều khiển cũng như kiến thức về điện tử. Do có cộng đồng phát triển mạnh nên các thư viện rất phong phú, việc đọc các giá trị cảm biến hay điều khiển thiết bị cũng trở nên đơn giản giúp quá trình xây dựng một ứng dụng trở nên nhanh chóng hơn.



Hình 3. Mô tả Arduino Uno R3

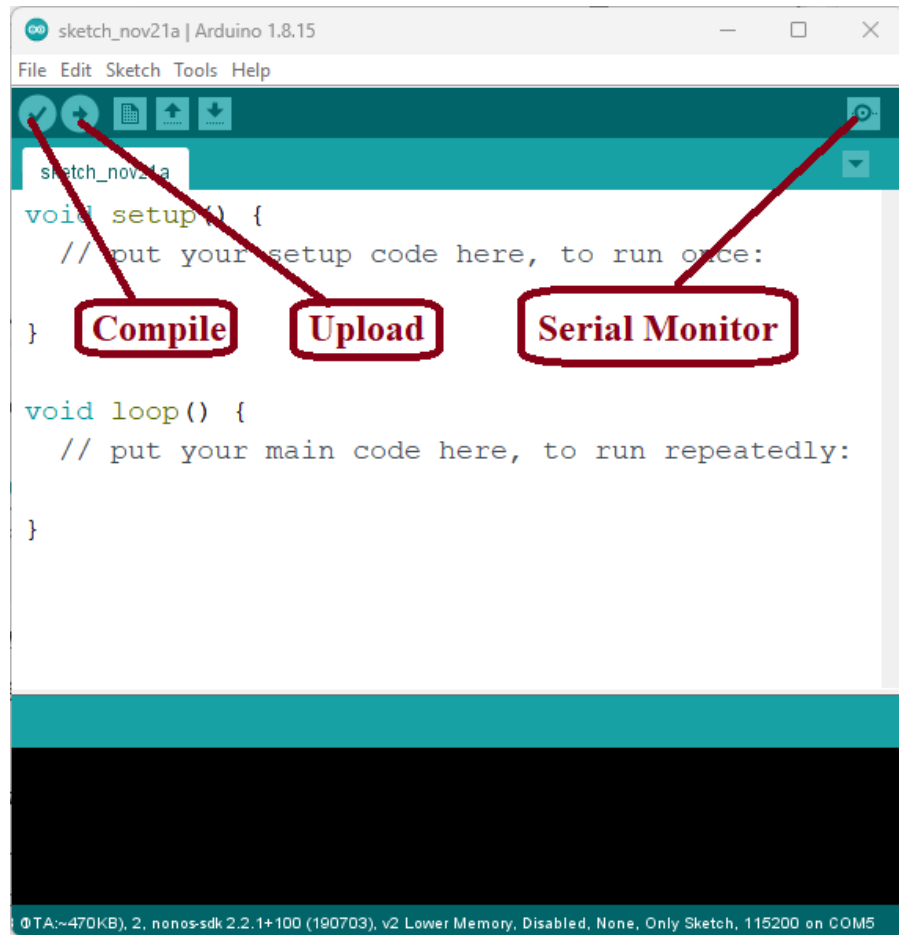
Arduino Uno R3 là một bo mạch phổ biến nhất trong dòng họ Arduino. Với chi phí rẻ nên phù hợp với việc bắt đầu làm quen với vi điều khiển.

Arduino IDE là một phần mềm miễn phí mã nguồn mở, được sử dụng chủ yếu để viết và biên dịch mã và nạp vào board mạch. Đây là một phần mềm chính thống được phát hành bởi Arduino giúp cho việc biên dịch mã trở nên dễ dàng, với giao diện dễ sử dụng giúp cho việc thao tác, lập trình với Arduino trở nên đơn giản hơn đối với người sử dụng. Các thiết bị dựa trên nền tảng Arduino được lập trình bằng ngôn ngữ Wiring – một biến thể của C/C++.



Hình 4. Tải Arduino IDE

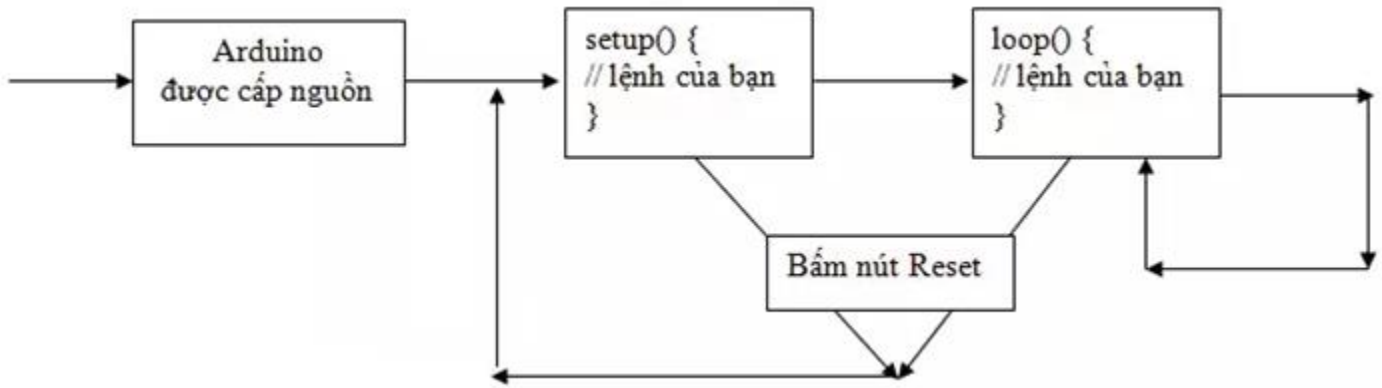
Sinh viên tiến hành download phần mềm tại: <https://www.arduino.cc/en/software>



Hình 5. Giao diện chính của Arduino IDE

Một chương trình trong trong Arduino cần phải có hai hàm chính đó là **setup()** và **loop()**:

- **setup()**: hàm này sẽ được gọi thực thi khi lần đầu tiên bo mạch được khởi động hoặc tới khi thiết bị được khởi động lại. Được sử dụng để khởi tạo các thiết lập cần thiết.
- **loop()**: sẽ được thực thi sau khi hàm setup() hoàn tất, nội dung trong hàm loop() sẽ được thực thi liên tục cho tới khi bị reset hoặc mất nguồn.



Hình 6. Mô tả một chu trình trong lập trình Arduino

Một số hàm thường hay sử dụng trong Arduino:

- ❖ **pinMode(PIN_NUMBER, MODE):** cấu hình MODE cho chân PIN_NUMBER là một đầu vào (INPUT) hay đầu ra (OUTPUT).
- ❖ **digitalWrite(PIN_NUMBER, VALUE):** xuất tín hiệu digital VALUE ra một chân trên Arduino. Có 2 loại VALUE là HIGH (hay giá trị 1) và LOW (hay giá trị 0). Nếu xuất tín hiệu HIGH thì giá trị điện áp ở chân pin sẽ có giá trị 5V ngược lại nếu xuất LOW thì giá trị điện áp sẽ là 0V. Ví dụ, nếu muốn xuất giá trị cao ra chân số 13 ta thực hiện digitalWrite(13, HIGH);
- ❖ **analogWrite(PIN_NUMBER, VALUE):** xuất tín hiệu analog VALUE ra một chân trên Arduino. Thông thường, người ta thường sử dụng hàm này để điều khiển mức sáng tối của đèn LED hay hướng quay của động cơ servo.
- ❖ **digitalRead(PIN_NUMBER):** là câu lệnh dùng để đọc tín hiệu điện từ một chân digital, giá trị trả về là 2 giá trị HIGH hoặc LOW.
- ❖ **analogRead(PIN_NUMBER):** là câu lệnh dùng để đọc giá trị điện áp từ một chân analog.
- ❖ **delay(ms):** có nhiệm vụ dừng chương trình trong thời gian mili giây.

Giao tiếp Serial

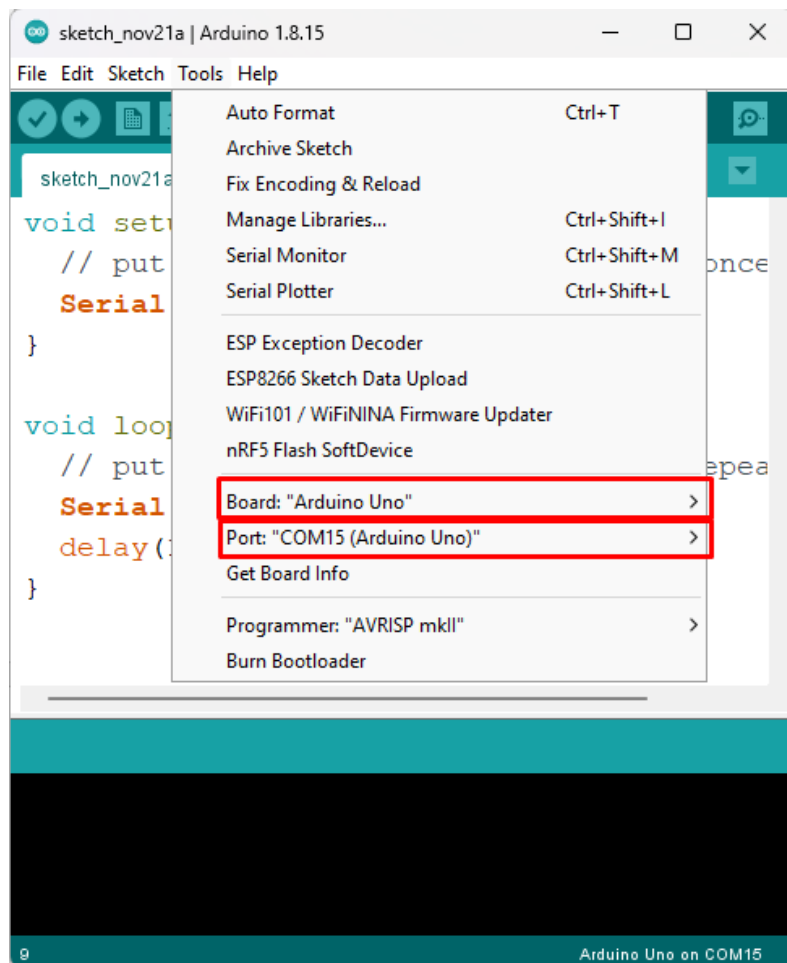
Thư viện Serial được dùng trong việc giao tiếp giữa các board mạch với nhau (hoặc board mạch với máy tính hoặc với các thiết bị khác). Tất cả các mạch Arduino đều có ít nhất 1 cổng Serial (hay còn được gọi là UART hoặc USART). Thông thường, chúng ta sẽ sử dụng giao tiếp này để debug hoặc xem các thông tin trên Arduino. Giao tiếp Serial được thực hiện qua 2

cổng digital 0 (RX) và 1 (TX) hoặc qua cổng USB tới máy tính. Vì vậy, nếu đang sử dụng các hàm của thư viện Serial này thì không thể sử dụng các chân digital 0 và digital 1 để làm việc khác được.

Nạp chương trình lên board Arduino UNO

Sau khi đã hoàn thành chương trình, để nạp chương trình vào board Arduino Uno ta thực hiện:

- B1: Trong menu **Tools->Board** chọn **Arduino Uno**
- B2: Chọn cổng giao tiếp giữa máy tính với Arduino
- B3: Bấm nút **Upload**



3. Một số loại cảm biến và thiết bị khác.

Các cảm biến đơn giản trên thị trường hiện tại hầu hết đều xuất ra tín hiệu digital và analog, các cảm biến này thường có từ 3 đến 4 chân. Trong đó có hai chân cấp nguồn (VCC, GND), một chân xuất tín hiệu digital (D0) và một chân xuất tín hiệu analog (A0).





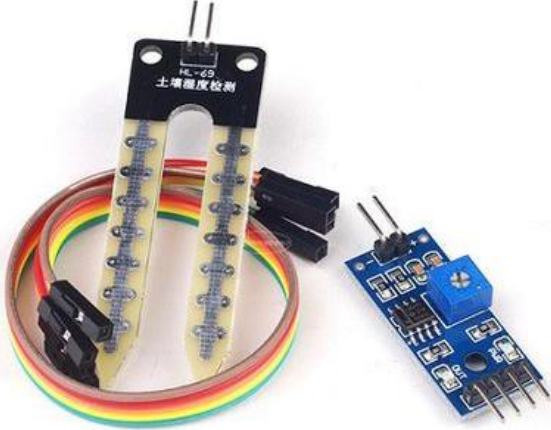
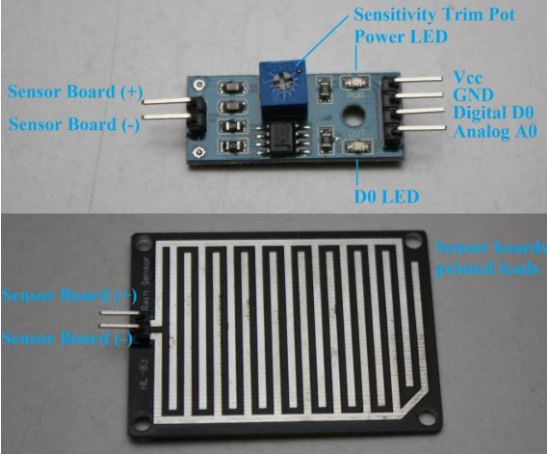
Hình 7. Mô tả các chân của cảm biến

Chân analog dùng để xuất ra tín hiệu analog, đây là tín hiệu do cảm biến xuất ra dựa vào các yếu tố mà cảm biến thu thập được. Giá trị này chỉ là giá trị điện thế, để có được giá trị đặt trưng liên quan tới cảm biến (ppm đối với khí, lux với ánh sáng hoặc dBA với âm thanh,...) thì cần một số bước chuyển đổi tính toán từ các giá trị này.



Trên cảm biến thường cũng sẽ có thêm một biến trở để điều chỉnh ngưỡng, khi giá trị cảm biến vượt ngưỡng này thì tín hiệu ở chân digital sẽ được bật lên.



Một số loại cảm biến đơn giản:

	Cảm biến chuyển động
	Cảm biến phát hiện lửa (hồng ngoại)

	Cảm biến độ ẩm đất
	Cảm biến phát hiện mưa

Ngoài các cảm biến cơ bản bên trên, để thu thập một giá trị chính xác chúng ta sẽ phải sử dụng các loại cảm biến phức tạp hơn đòi hỏi phải có kinh nghiệm để sử dụng. Và việc phức tạp đó đã được cộng đồng Arduino giải quyết, chúng ta chỉ cần biết tên cảm biến, sẽ có rất nhiều thư viện được công bố giúp chúng ta đọc được giá trị của loại cảm biến đó.

	Cảm biến siêu âm (HC-SR04), dùng để đo khoảng cách
	Cảm biến cường độ ánh sáng (BH1750)

	Cảm biến nhịp tim và SpO ₂ (MAX30100)
	Cảm biến nhiệt độ, độ ẩm (DHT22)

Một số thiết bị khác sử dụng trong bài tập, sinh viên tự tìm hiểu:

	Động cơ servo
	Màn hình LCD 1602

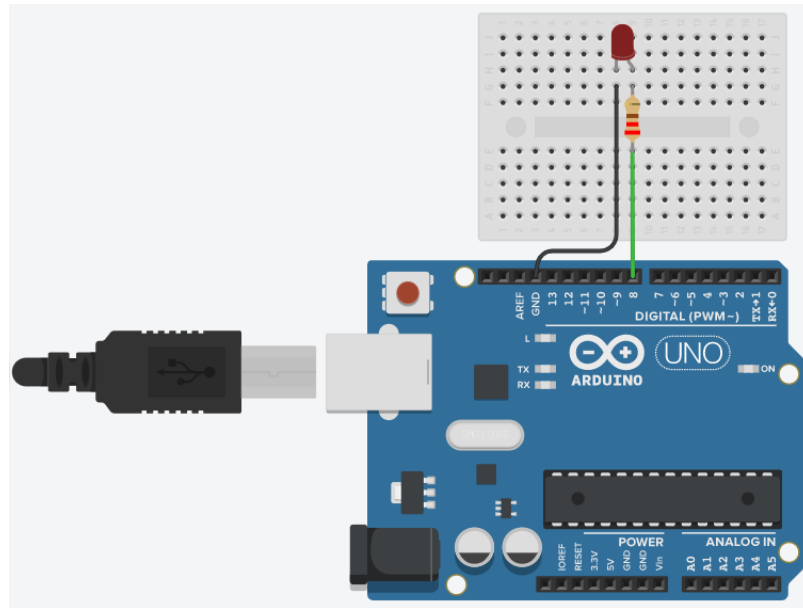
C. THỰC HÀNH

1. Tương tác đơn giản với đèn LED

a. Chớp tắt đèn LED

Trong phần này sinh viên sẽ tiến hành lập trình Arduino để điều khiển đèn LED chớp tắt định kỳ.

B1: Lắp sơ đồ như hình dưới



B2: Sử dụng Arduino IDE để lập trình

```
int LED = 8; // Điều khiển chân thứ 8 trên Arduino UNO

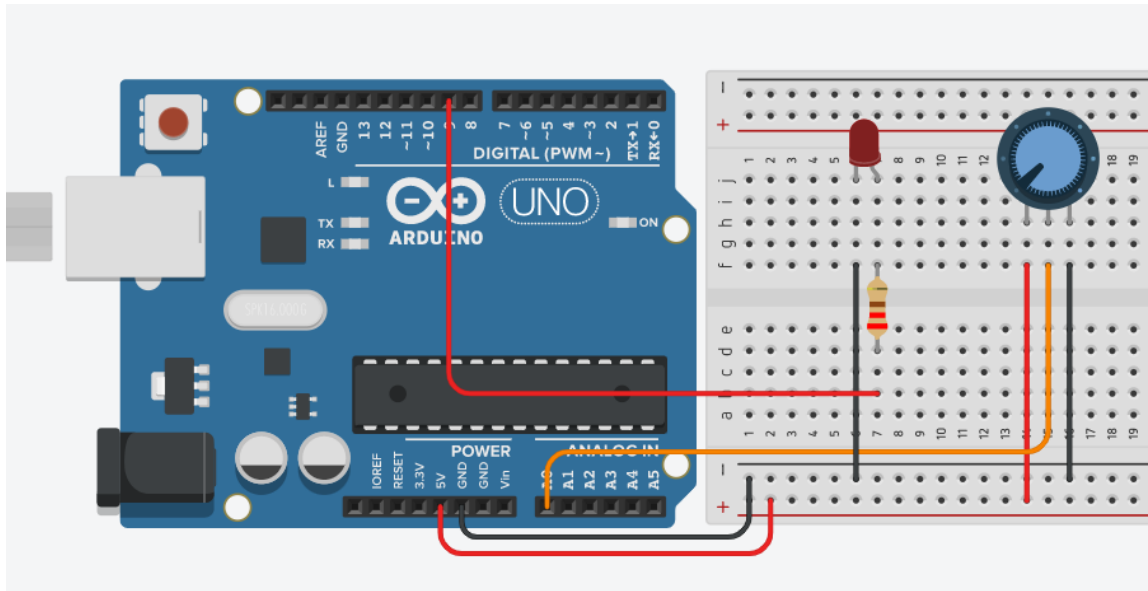
void setup()
{
    // Cài đặt chế độ cho chân điều khiển đèn LED là OUTPUT
    pinMode(LED, OUTPUT);
}

void loop()
{
    digitalWrite(LED, HIGH); // Xuất tín hiệu digital ở mức cao
    delay(1000); // Dừng 1000 mili giây
    digitalWrite(LED, LOW); // Xuất tín hiệu digital ở mức thấp
    delay(1000); // Dừng 1000 mili giây
}
```

b. Điều khiển độ sáng đèn LED thông qua biến trở

Trong phần này sinh viên lập trình Arduino để đọc giá trị của biến trở được điều khiển bởi người dùng, thông qua giá trị đó điều khiển độ sáng của đèn LED.

B1: lắp mạch như hình dưới



B2: Sử dụng Arduino IDE để lập trình.

```
int LED = 9;

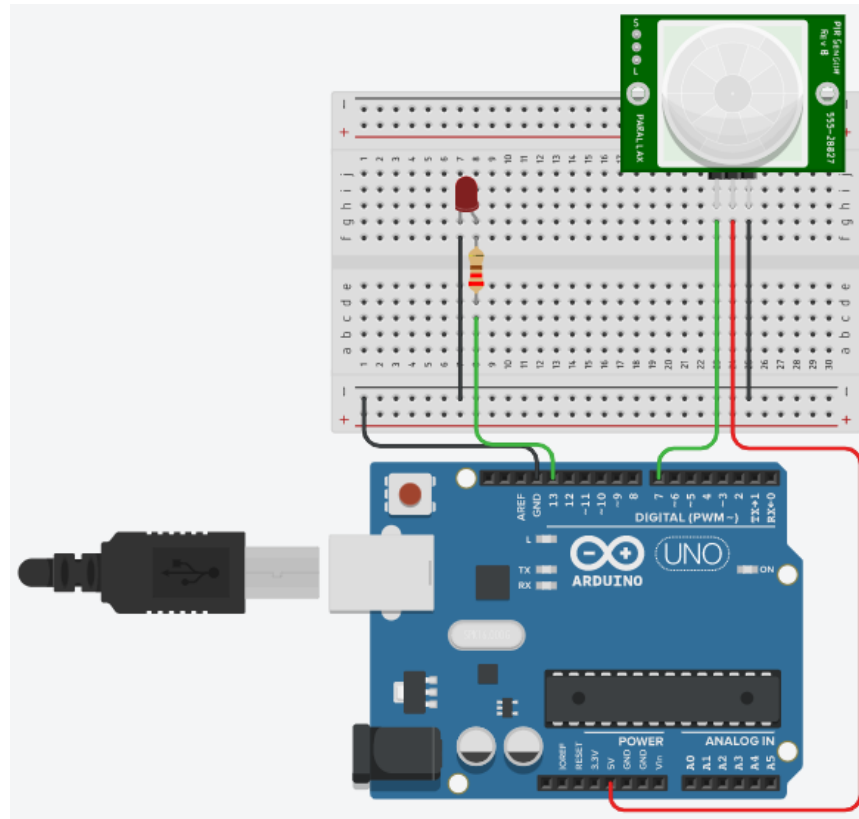
void setup()
{
  pinMode(LED, OUTPUT);
  pinMode(A0, INPUT);
}

void loop()
{
  // Đọc giá trị từ biến trở
  int value = analogRead(A0);
  // Chuyển đổi giá trị trong khoảng 0-1023 thành 0-255
  value = map(value, 0, 1023, 0, 255);
  // Xuất tín hiệu analog ra chân đèn LED
  analogWrite(LED, value);
  delay(100);
}
```

2. Làm quen với cảm biến

Sinh viên tiến hành làm quen với cảm biến chuyển động (PIR Sensor), lập trình Arduino sao cho nếu phát hiện chuyển động thì báo hiệu bằng cách chớp tắt đèn liên tục.

B1: Lắp mạch như hình dưới



B2: Lập trình với Arduino IDE

```
int sensorState = 0;
int sensorPin = 7;

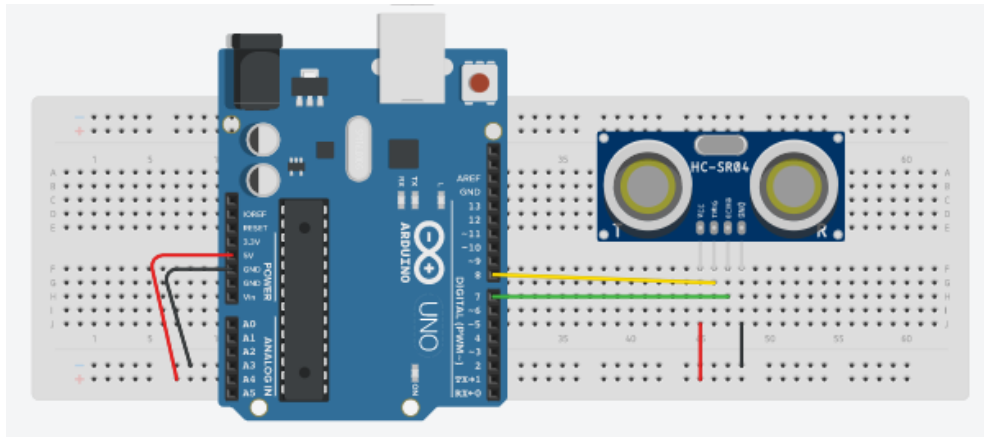
void setup()
{
  pinMode(sensorPin, INPUT);
  pinMode(LED_BUILTIN, OUTPUT);
}

void loop()
{
  sensorState = digitalRead(sensorPin);
  if (sensorState == HIGH) {
    digitalWrite(LED_BUILTIN, HIGH);
    delay(100);
    digitalWrite(LED_BUILTIN, LOW);
    delay(100);
  }
  else digitalWrite(LED_BUILTIN, LOW);
  delay(100);
}
```

Serial Monitor

Sinh viên xây dựng mạch và sử dụng cảm biến khoảng cách siêu âm để đo giá trị khoảng cách vật thể tiếp cận, sau đó in các giá trị này ra serial.

B1: lắp mạch như hình dưới



B2: lập trình đọc giá trị và tính toán khoảng cách sau đó in ra serial

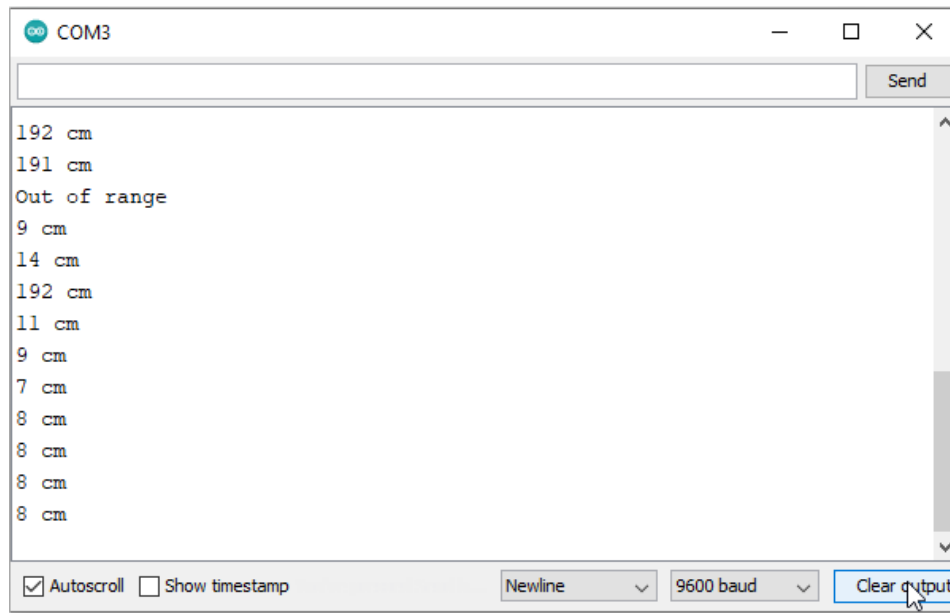
```
#define trigPin 8
#define echoPin 7

void setup()
{
  Serial.begin(9600);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
}

void loop()
{
  long duration, distance;
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  distance = (duration/2) / 29.1;
  if (distance >= 200 || distance <= 0)
  {
    Serial.println("Out of range");
  }
}
```

```
}  
else  
{  
    Serial.print(distance);  
    Serial.println(" cm");  
}  
delay(500);  
}
```

B3: nạp code, mở Serial Monitor xem kết quả



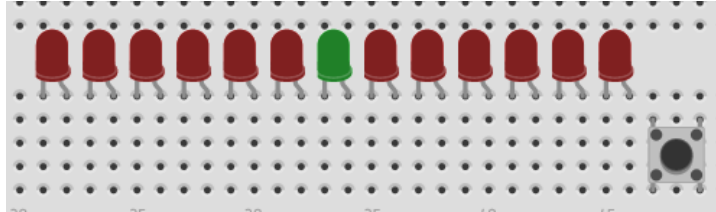
D. YÊU CẦU & NỘI BÀI

1. Yêu cầu

Sử dụng những kiến thức đã được học bên trên, sinh viên tiến hành thực hiện các kịch bản sau:

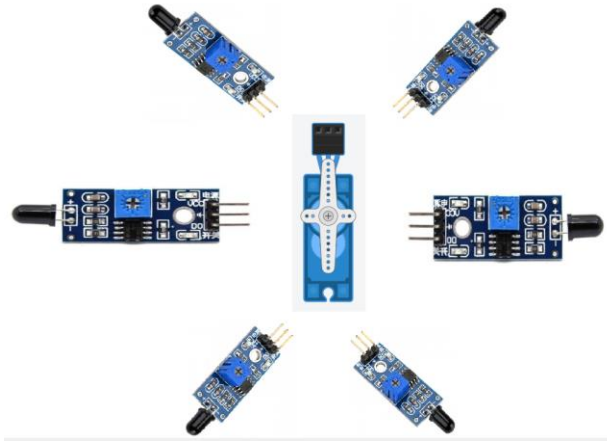
1. Xây dựng kịch bản gồm có 12 đèn LED và 1 biến trở. Điều chỉnh số lượng đèn sáng dựa vào giá trị của biến trở được điều khiển.
2. Mô phỏng trò chơi chọn đúng đèn, xây dựng kịch bản gồm có 13 đèn, trong đó có 1 đèn khác màu, 1 nút bấm. Đèn sẽ sáng lần lượt từ trái -> phải, phải -> trái và lặp lại liên tục. Nếu người chơi nhấn nút đúng vào thời điểm đèn khác màu sáng thì sẽ

được cộng điểm và đồng thời tăng tốc độ di chuyển của các bóng đèn, nếu sai thì trừ điểm và giảm tốc độ lại. Trên serial monitor cũng sẽ hiển thị thông tin cho người chơi bao gồm: số điểm hiện tại, người dùng nhấn đúng hay sai.



3. Máy tưới cây tự động. Đọc tín hiệu analog từ cảm biến độ ẩm đất, đặt một giá trị ngưỡng nếu độ ẩm đất dưới ngưỡng này sẽ bật máy bơm để tưới cây (mô phỏng máy bơm bằng đèn LED).
4. Đèn vườn tự động, khi trời tối thì đèn tự động bật lên. Sử dụng cảm biến ánh sáng hoặc quang trở để nhận biết trời sáng hay tối dựa vào đó thì bật hoặc tắt đèn.
5. Xây dựng ứng dụng cảnh báo lùi xe. Giả sử chúng ta có một cảm biến siêu âm gắn ở đuôi xe, đọc giá trị cảm biến này. Dựa vào khoảng cách mà cảm biến cho ra, biểu thị bằng 10 đèn LED (càng gần thì càng nhiều đèn sáng).
6. Hệ thống cảnh báo cháy, sử dụng cảm biến lửa, 2 đèn LED khác màu và một động cơ servo. Khi có đám cháy, 2 đèn LED sẽ thay phiên nhau chớp tắt liên tục đồng thời động cơ servo cũng sẽ quay (mô phỏng việc tự mở cửa thoát hiểm).
7. Sử dụng cảm biến nhiệt độ, độ ẩm, tiến hành đọc giá trị từ cảm biến và in ra serial kèm theo đơn vị.
8. Thiết bị theo dõi môi trường trong nhà. Đọc giá trị cảm biến nhiệt độ, độ ẩm, ánh sáng hiển thị lên màn hình LCD 1602.
9. Làm thiết bị sức khỏe. Sử dụng cảm biến nhịp tim và SpO_2 và màn hình LCD, đọc các giá trị đó và hiển thị lên màn hình LCD.

10. Hệ thống dập lửa tự động. Giả sử chúng ta có một hệ thống phun nước được gắn trên một động cơ servo, và các cảm biến lửa được gắn xung quanh. Khi phát hiện có lửa, động cơ servo sẽ quay tới hướng đó và phun nước để dập tắt nó (chỉ cần quay động cơ servo).



2. Nộp bài

- Sinh viên tìm hiểu và thực hành theo hướng dẫn. Thực hiện theo **nhóm**.
- Sinh viên báo cáo kết quả thực hiện và nộp bài bằng file. Trong đó:
 - Trình bày chi tiết quá trình thực hành và trả lời các câu hỏi nếu có (kèm theo các ảnh chụp màn hình tương ứng).
 - Giải thích, tìm hiểu, lý giải các kết quả đạt được.
 - Tải mẫu báo cáo thực hành và trình bày theo mẫu được cung cấp.

Nén **.ZIP** tất cả các file và đặt tên file theo định dạng theo mẫu:

NhomY-LabX_MSSV1_MSSV2

Ví dụ: Nhom1-Lab04_20520001_20520002

- Nộp báo cáo trên theo thời gian đã thống nhất tại website môn học.
- Các bài nộp không tuân theo yêu cầu sẽ **KHÔNG** được chấm điểm.

HẾT

Chúc các bạn hoàn thành tốt bài thực hành!