



BÁO CÁO THỰC HÀNH

CÔNG NGHỆ INTERNET OF THINGS HIỆN ĐẠI

Lab 2

TÊN BÀI THỰC HÀNH

GVHD: Phan Trung Phát

Lớp: NT532.021

Họ và tên	MSSV
Nguyễn Thành Đăng	21520683
Nguyễn Trần Bảo Quốc	21520421

ĐÁNH GIÁ KHÁC (*):

Nội dung	Kết quả
Tổng thời gian thực hiện bài thực hành trung bình (1)	1,5 tuần
Link Video thực hiện (2) (nếu có)	Bài 1 Bài 2 Bài 3 Bài 4 Bài 5 Bài 6 Bài 7
Ý kiến (3) (nếu có) + Khó khăn + Đề xuất ...	
Điểm tự đánh giá (4)	10/10

(*): phần (1) và (4) bắt buộc thực hiện.

Phần bên dưới là báo cáo chi tiết của nhóm/cá nhân thực hiện.



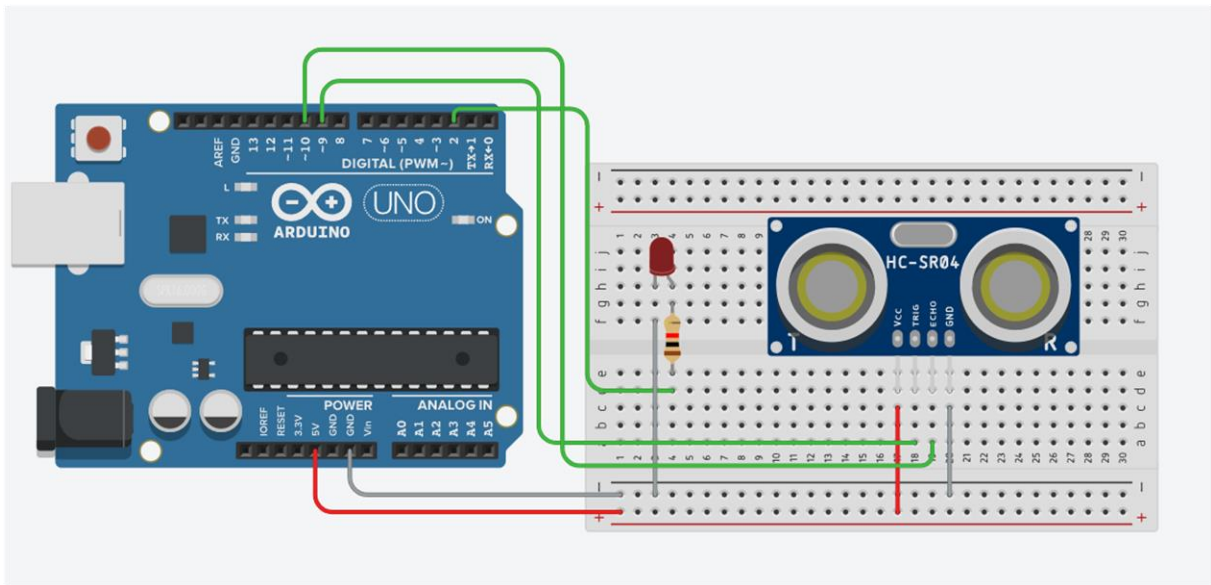
LƯU HÀNH NỘI BỘ

Câu hỏi 1: Xây dựng ứng dụng lùi xe, kịch bản gồm có 1 đèn LED và 1 cảm biến khoảng cách siêu âm. Cảm biến được gắn cố định và khi xe lùi về càng gần cảm biến thì đèn càng chớp tắt nhanh hơn để báo hiệu. Chia mức độ cảnh báo thành 5 mức từ chậm đến nhanh.

1. Minh chứng

Link demo bài 1: [Tại đây](#)

Mô hình nối mạch Arduino:



Code thực thi:

```
void flashingLed(int delayTime) {
    digitalWrite(led, HIGH);
    delay(delayTime);
    digitalWrite(led, LOW);
    delay(delayTime);
}
```

Hàm flashingLed

```
void loop() {
    digitalWrite(trigPin, LOW);
    delayMicroseconds(2);

    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);

    duration = pulseIn(echoPin, HIGH);

    distance= duration*0.034/2;

    Serial.print("Distance: ");
    Serial.print(distance);
    Serial.println(" cm");
}
```

Hàm loop

```
if (distance < 10) {
    flashingLed(200);
}
else if (distance < 20) {
    flashingLed(400);
}
else if (distance < 30) {
    flashingLed(600);
}
else if (distance < 40) {
    flashingLed(800);
}
else if (distance < 50) {
    flashingLed(1000);
}
```



2. Giải thích

Hàm `flashingLed()` có công dụng bật nháy đèn LED, với khoảng thời gian giữa 2 lần chớp tắt là `delayTime`, giá trị này càng nhỏ thì tốc độ nháy càng nhanh.

Hàm `loop()` sẽ phát xung từ chân `trigPin` trong khoảng thời gian 10ms, sau đó đo độ rộng xung HIGH ở chân `echoPin` (ms) và lưu vào biến `duration`, từ đó tính ra khoảng cách và lưu vào biến `distance`.

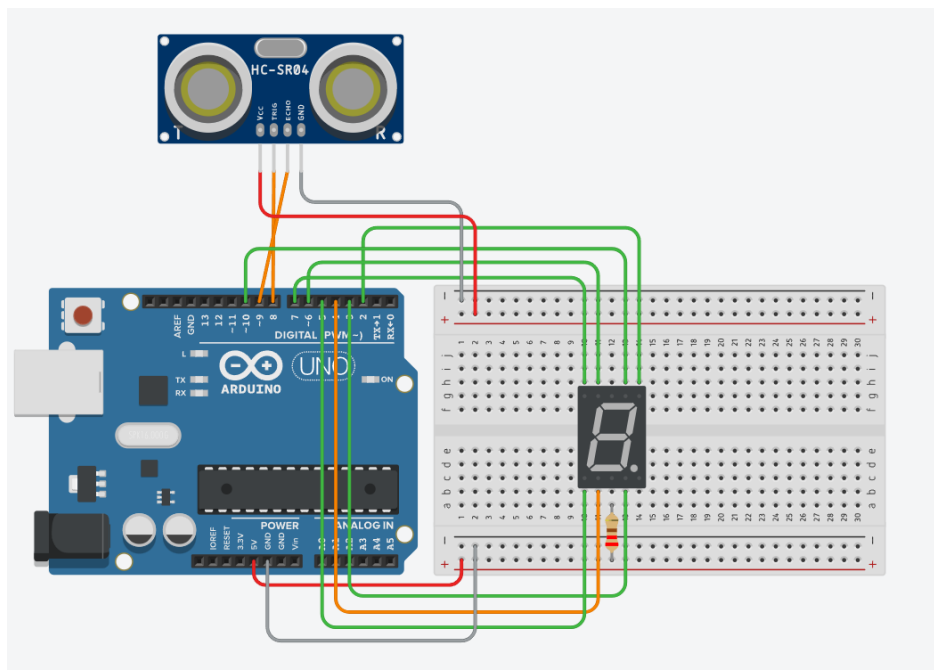
Tiếp theo, nó so sánh khoảng cách với 5 mốc giá trị khác nhau và gọi hàm `flashingLed()` để chớp nháy đèn, khoảng cách càng gần thì giá trị truyền vào hàm càng nhỏ và đèn nháy càng nhanh dần.

Câu hỏi 2: Xây dựng mô hình gồm có 1 LED 7 đoạn và 1 cảm biến khoảng cách siêu âm (Ultrasonic Distance Sensor). Dựa vào khoảng cách tiếp cận gần hay xa của vật thể thì LED 7 đoạn sẽ thể hiện bằng các con số. Nếu khoảng cách nằm ngoài vùng phủ của cảm biến hoặc bị sai thì LED 7 đoạn hiển thị số 0.

1. Minh chứng

Link demo bài 2: [Tại đây](#)

Mô hình nối mạch Arduino:



Code thực thi:



```
void loop()
{
    unsigned long duration;
    int distance;

    digitalWrite(trig, LOW);
    delayMicroseconds(2);

    digitalWrite(trig, HIGH);
    delayMicroseconds(10);
    digitalWrite(trig, LOW);

    duration = pulseIn(echo,HIGH);
    distance = int(duration*0.034/2);
    Serial.print("Distance: ");
    Serial.println(distance);

    if(distance < 10 ){
        one();
    }
    else if(distance >= 10 && distance < 20){
        two();
    }
}
```

```
    else if(distance >= 20 && distance < 30){
        three();
    }
    else if(distance >= 30 && distance < 40){
        four();
    }
    else if(distance >= 40 && distance < 50){
        five();
    }
    else if(distance >= 50 && distance < 60){
        six();
    }
    else if(distance >= 60 && distance < 70){
        seven();
    }
    else if(distance >= 70 && distance < 80){
        eight();
    }
    else if(distance >= 80 && distance < 90){
        nine();
    }
    else{
        zero();
    }
}
```

Hàm loop

```
void one()
{
    digitalWrite(b,LOW);
    digitalWrite(c,LOW);
    digitalWrite(a,HIGH);
    digitalWrite(d,HIGH);
    digitalWrite(e,HIGH);
    digitalWrite(f,HIGH);
    digitalWrite(g,HIGH);
}

void two()
{
    digitalWrite(a,LOW);
    digitalWrite(b,LOW);
    digitalWrite(g,LOW);
    digitalWrite(e,LOW);
    digitalWrite(d,LOW);
    digitalWrite(c,HIGH);
    digitalWrite(f,HIGH);
}
```

```
void three()
{
    digitalWrite(a,LOW);
    digitalWrite(b,LOW);
    digitalWrite(g,LOW);
    digitalWrite(c,LOW);
    digitalWrite(d,LOW);
    digitalWrite(e,HIGH);
    digitalWrite(f,HIGH);
}

void four()
{
    digitalWrite(f,LOW);
    digitalWrite(g,LOW);
    digitalWrite(b,LOW);
    digitalWrite(c,LOW);
    digitalWrite(a,HIGH);
    digitalWrite(d,HIGH);
    digitalWrite(e,HIGH);
}
```

```
void five()
{
    digitalWrite(a,LOW);
    digitalWrite(f,LOW);
    digitalWrite(g,LOW);
    digitalWrite(c,LOW);
    digitalWrite(d,LOW);
    digitalWrite(b,HIGH);
    digitalWrite(e,HIGH);
}

void six()
{
    digitalWrite(a,LOW);
    digitalWrite(f,LOW);
    digitalWrite(g,LOW);
    digitalWrite(c,LOW);
    digitalWrite(d,LOW);
    digitalWrite(e,LOW);
    digitalWrite(b,HIGH);
}
```

Các hàm hiển thị số trên LED 7 đoạn

2. Giải thích

Hàm loop() sẽ phát xung từ chân trig trong khoảng thời gian 10ms, sau đó đo độ rộng xung HIGH ở chân echo (ms) và lưu vào biến duration, từ đó tính ra khoảng cách và lưu vào biến distance.

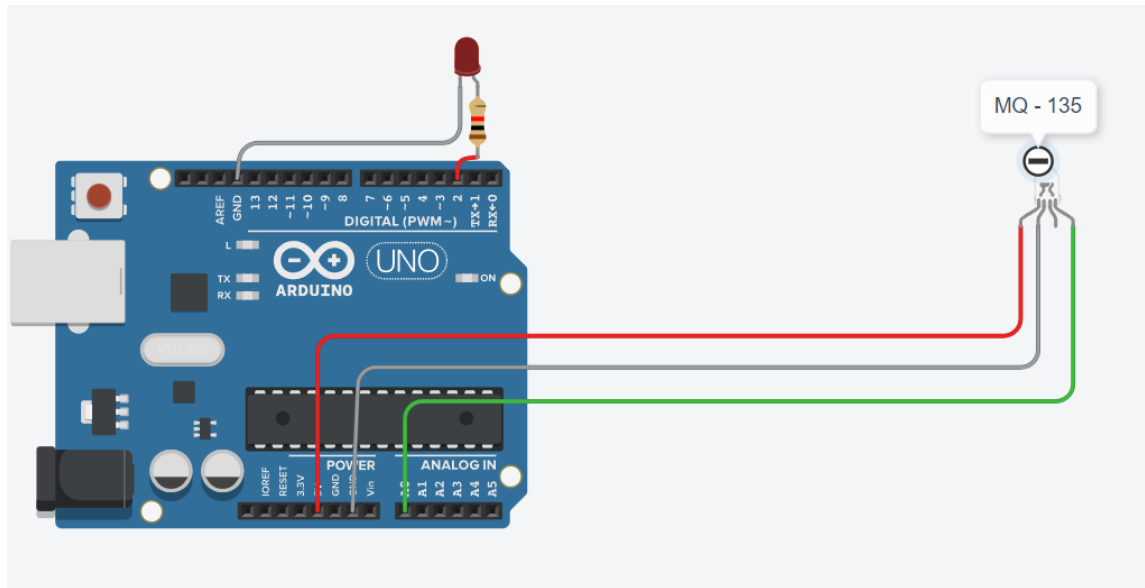
Sau đó, so sánh distance với 10 mốc khoảng cách, từ đó gọi ra các hàm để hiển thị số trên LED 7 đoạn (one(), two(), three(), ..., nine(), zero()).

Câu hỏi 3: Xây dựng kịch bản cảnh cáo chất lượng không khí, kịch bản gồm có 1 đèn LED và 1 cảm biến MQ-135. Tìm hiểu về thông số PPM để thiết đặt thông số chất lượng không khí bất thường và cảnh báo thông qua đèn. Tùy vào mật độ chất lượng không khí thì đèn sẽ chớp tắt theo 3 mức cảnh báo là nhanh - trung bình - chậm.

1. Minh chứng

Link demo bài 3: [Tại đây](#)

Mô hình nối mạch Arduino:



Code thực thi:

```
void flashingLed(int delayTime) {
    digitalWrite(led, HIGH);
    delay(delayTime);
    digitalWrite(led, LOW);
    delay(delayTime);
}
```

Hàm flashingLed

```
void loop() {
    float ppm = mq135_ss.getPPM();

    Serial.print("PPM Value: ");
    Serial.println(ppm);

    if (ppm <= 500) {
        digitalWrite(led, LOW);
    } else if (ppm <= 1000) {
        flashingLED(1000);
    } else if (ppm <= 2000) {
        flashingLED(750);
    } else {
        flashingLED(250);
    }
    delay(200);
}
```

Hàm loop

2. Giải thích

mq135_ss là sensor cảm biến được khai báo từ thư viện MQ135.h. Hàm getPPM() được gọi để thu giá trị chất lượng không khí và lưu vào biến ppm. Sau đó, so sánh

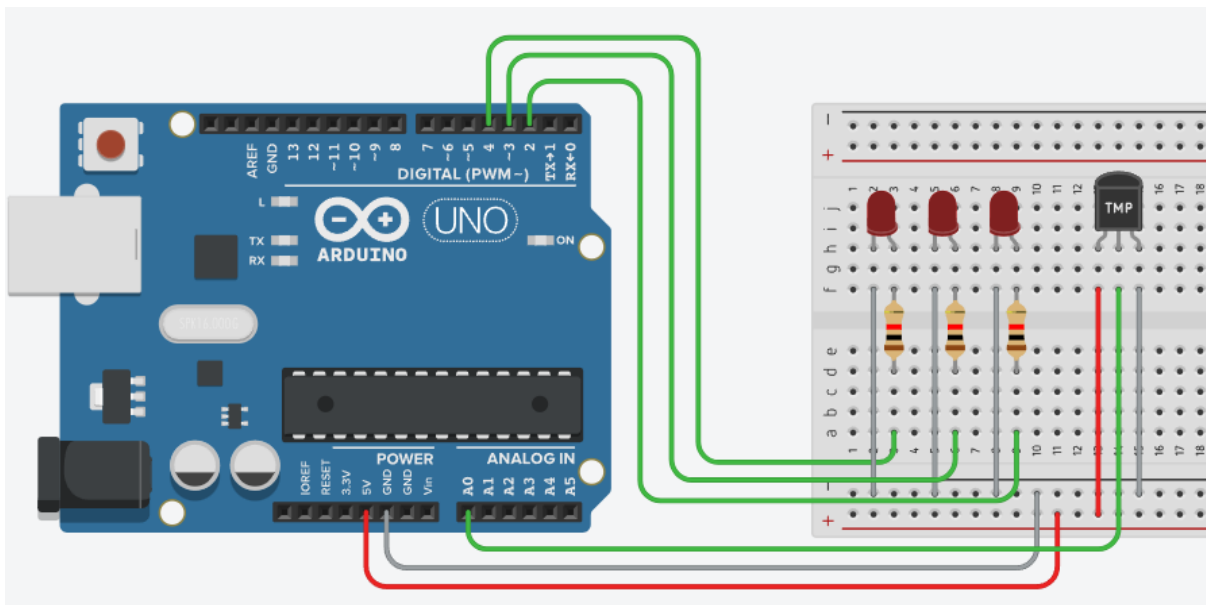
ppm với 3 mức giá trị, nếu nồng độ tạp chất thấp (dưới 500) thì tắt đèn, nồng độ càng cao thì nháy đèn càng nhanh.

Câu hỏi 4: Xây dựng ứng dụng tản nhiệt cho DataCenter, kịch bản gồm 1 cảm biến nhiệt độ, độ ẩm và 3 đèn LED. Nếu nhiệt độ quá cao thì điều khiển bật quạt (tương trưng bằng các đèn LED) để tản nhiệt cho các Server. Tương ứng với 3 mức của quạt được mô phỏng bằng 3 đèn LED, tùy theo nhiệt độ trong DataCenter. Bên cạnh đó, nếu độ ẩm trong không khí của DataCenter quá cao (khi quạt có thể đang được bật ở mức 1 và mức 2) thì điều khiển để quạt tắt đi (3 đèn tắt). Lưu ý, khi nhiệt độ ở mức cao nhất (mức 3) thì quạt sẽ phải quạt ở mức tối đa bất kể độ ẩm trong không khí.

1. Minh chứng

Link video demo bài 4: [Tại đây](#)

Mô hình nối mạch Arduino:



Code thực thi:

```
void switchFan(int level) {
  for (int i=0; i<level; i++) {
    digitalWrite(i+2, HIGH);
    delay(200);
  }
  for (int i=level; i<3; i++) {
    digitalWrite(i+2, LOW);
    delay(200);
  }
}
```

Hàm switchFan



```
void loop() {  
    float h = dht.readHumidity();  
    float t = dht.readTemperature();  
  
    Serial.print("Nhiệt độ: ");  
    Serial.println(t);  
    Serial.print("Độ ẩm: ");  
    Serial.println(h);  
  
    if (t >= 40) {  
        switchFan(3);  
    }  
    else if (t >= 30) {  
        if (h < 80)  
            switchFan(2);  
        else switchFan(0);  
    }  
    else if (t >= 24) {  
        if (h < 80)  
            switchFan(1);  
        else switchFan(0);  
    }  
    else switchFan(0);  
}
```

Hàm loop

2. Giải thích

Hàm switchFan() nhận vào tham số level tương ứng với số quạt (đèn LED) bật. Tại level n thì sẽ bật n đèn LED.

Trong hàm loop, gọi 2 hàm readHumidity() và readTemperature để đọc giá trị cảm biến và lưu vào 2 biến tương ứng là h (độ ẩm) và t (nhiệt độ).

Nếu nhiệt độ trên 40°C thì bật 3 đèn LED. Nếu nhiệt độ từ 30-39°C, kiểm tra xem độ ẩm có vượt quá 80% hay không, nếu không thì bật 2 LED. Tương tự với ngưỡng 24-29°C nhưng chỉ bật 1 LED. Nếu nhiệt độ thấp hơn 24, tắt cả 3 đèn.

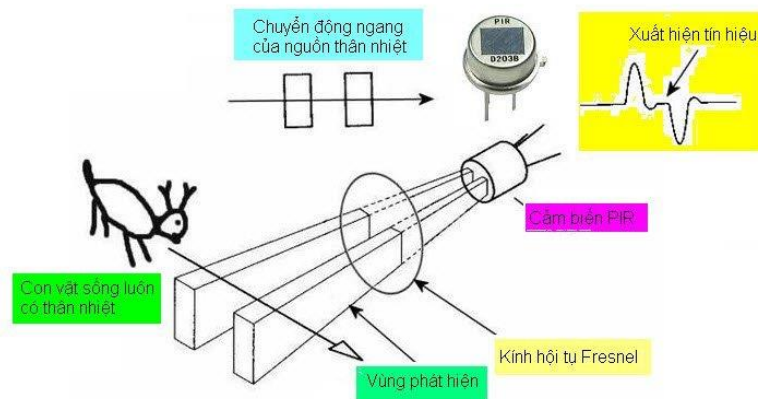
Câu hỏi 5: Sử dụng cảm biến chuyển động hồng ngoại PIR (HC-SR501), tiến hành đọc giá trị từ cảm biến và in ra serial. Tìm hiểu cách thức vận hành, sơ đồ mạch điện và cách thức sử dụng của loại cảm biến này.

1. Cách thức vận hành

Cảm biến PIR gồm 2 thành phần là một cảm biến bức xạ hồng ngoại nằm trong một thấu kính Fresnel.



Khác với một số loại cảm biến chuyển động khác, PIR là một cảm biến thụ động, nghĩa là nó không chủ động phát ra các tia phản xạ để xác định chuyển động của vật thể. PIR thu ánh sáng hồng ngoại trong phạm vi của nó và phát hiện ra những sự thay đổi trong lượng hồng ngoại lên cảm biến.

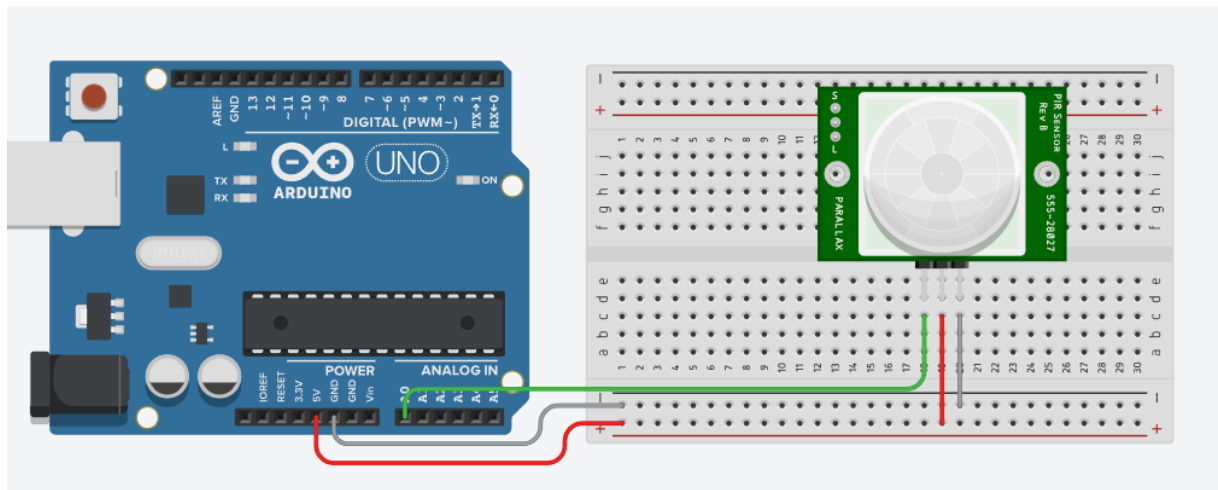


Tất cả mọi vật thể đều phát ra tia hồng ngoại dù nhiều hay ít, và lượng hồng ngoại thay đổi phụ thuộc vào nhiệt độ của vật hay đặc điểm bề mặt của vật thể. Khi một người nào đó di chuyển ngang qua cảm biến, lượng bức xạ trong vùng cảm biến sẽ tăng lên và cảm biến chuyển đổi sự thay đổi này thành điện áp đầu ra HIGH, tương ứng với “Phát hiện chuyển động”. Nếu lượng hồng ngoại không thay đổi, như khi người đó đứng yên hoặc đã rời khỏi phạm vi cảm biến, PIR sẽ phát ra điện áp LOW, tương ứng “Không có chuyển động”.

2. Minh chứng

Link video demo: [Tại đây](#)

Mô hình nối mạch Arduino:



Code thực thi:


```

int sensorState = 0;
int sensorPin = A0;

void setup()
{
  pinMode(sensorPin, INPUT);
  Serial.begin(9600);
}

void loop()
{
  sensorState = digitalRead(sensorPin);
  if (sensorState == 1) {
    Serial.println("Motion detected");
  }
  else {
    Serial.println("No Motion detected");
  }
  delay(200);
}

```

3. Giải thích

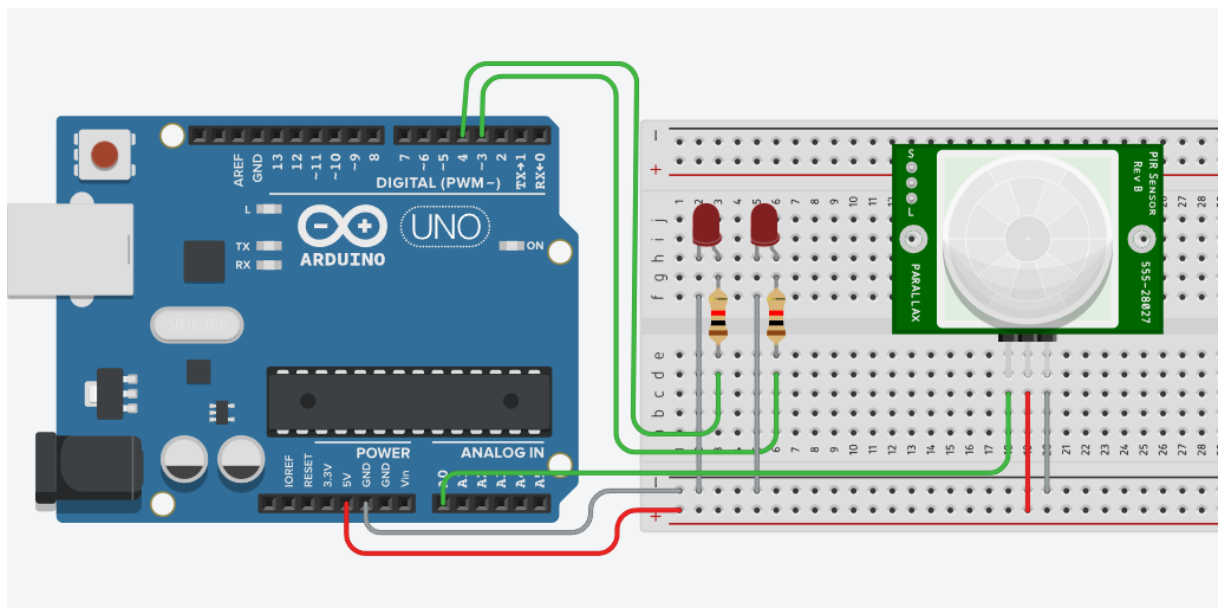
Mã nguồn rất đơn giản, dùng digitalRead để đọc điện áp ngõ ra của cảm biến PIR, nếu điện áp ở mức cao, in dòng "Motion detected" ra serial, nếu ngược lại thì in "No motion detected".

Câu hỏi 6: Mô phỏng máy giám sát vật thể, xây dựng kịch bản gồm có 1 cảm biến chuyển động, 2 đèn LED. Khi đối tượng cần quan sát chuyển động thì 2 đèn LED sẽ chớp tắt liên tục để báo động. Khi không có chuyển động thì trở lại trạng thái bình thường.

1. Minh chứng

Link video demo: [Tại đây](#)

Mô hình nối mạch Arduino:



Code thực thi:

```
void flashingLed() {
  digitalWrite(led1, LOW);
  digitalWrite(led2, HIGH);
  delay(250);
  digitalWrite(led2, LOW);
  digitalWrite(led1, HIGH);
  delay(250);
}
```

Hàm *flashingLED*

```
void loop()
{
  sensorState = 0;
  sensorState = digitalRead(sensorPin);

  if (sensorState == 1) {
    Serial.println("Motion detected");
    flashingLed();
  }
  else {
    Serial.println("No Motion detected");
    digitalWrite(led1, LOW);
    digitalWrite(led2, LOW);
  }

  delay(200);
}
```

Hàm *loop*

2. Giải thích

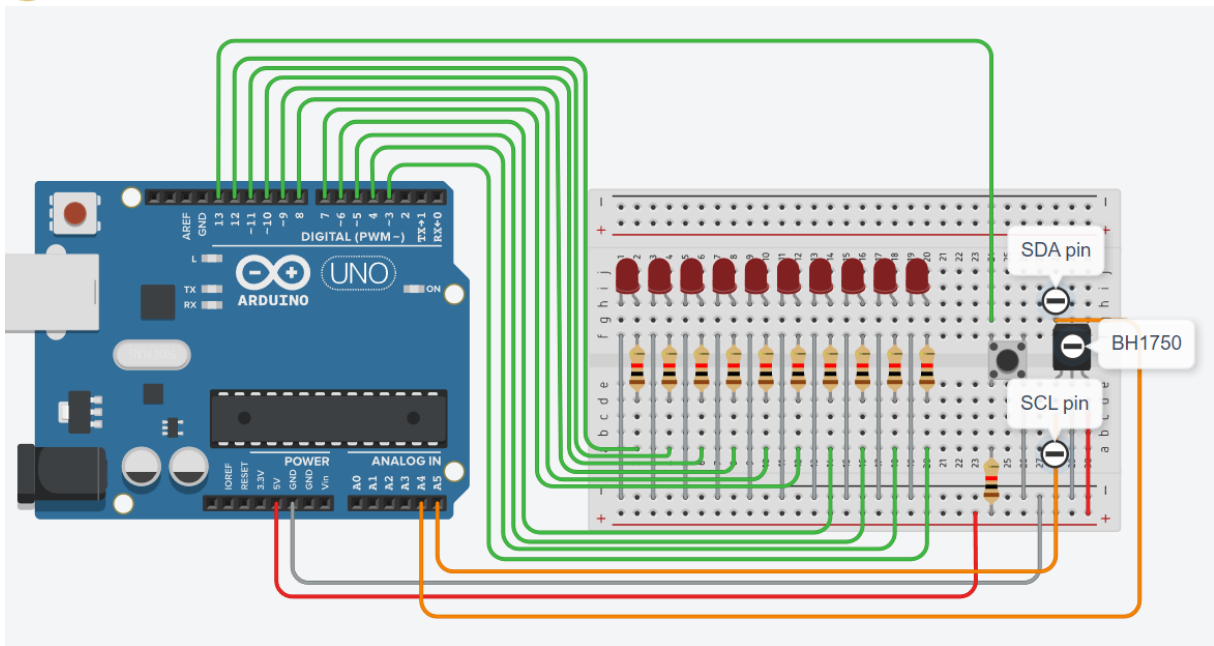
Dùng `digitalRead` để đọc giá trị của chân A0 nối với ngõ tín hiệu của cảm biến PIR. Nếu giá trị bằng 1 thì gọi hàm `flashingLed()` để chớp nháy 2 đèn LED. Nếu giá trị bằng 0, tắt cả 2 LED.

Câu hỏi 7: Mô phỏng phòng đọc sách thông minh, xây dựng kịch bản gồm có 1 cảm biến ánh sáng, 10 đèn LED, 1 nút bấm. Tùy theo cường độ ánh sáng đo được, nếu ánh sáng quá kém thì tự động bật tối đa số đèn, tương tự như thế điều chỉnh số lượng đèn sáng dựa vào cường độ ánh sáng để người đọc sách có đủ ánh sáng. Nút bấm dùng để thay đổi chế độ đèn để tiết kiệm điện: chế độ 1 là sử dụng 5 đèn khi bấm 1 lần, chế độ 2 là sử dụng tối đa 10 đèn nếu bấm liên tiếp 2 lần.

1. Minh chứng

Link video demo: [Tại đây](#)

Mô hình nối mạch Arduino (Tinkercad không có BH1750):



Code thực thi:

```
void loop() {
  float lightVal = lightSensor.readLightLevel();
  Serial.println(lightVal);

  if (lightVal > 50) {
    for (int i = 3; i <= 12; i++) {
      digitalWrite(i, LOW);
    }
  }
}
```

```
// Bật 3 đèn
else if (lightVal > 30) {
  for (int i = 3; i <= 5; i++) {
    digitalWrite(i, HIGH);
  }
  for (int i = 6; i <= 12; i++) {
    digitalWrite(i, LOW);
  }
}
```

```
// Bật 6 đèn
else if (lightVal > 20) {
  for (int i = 3; i <= 7; i++) {
    digitalWrite(i, HIGH);
  }
  if (MAX == 12)
    digitalWrite(8, HIGH);
  else digitalWrite(8, LOW);
  for (int i = 9; i <= 12; i++) {
    digitalWrite(i, LOW);
  }
}
```

```
// Bật max đèn
else {
  for (int i = 3; i <= MAX; i++) {
    digitalWrite(i, HIGH);
  }
  for (int i = MAX+1; i <= 12; i++) {
    digitalWrite(i, LOW);
  }
}

// Nhận đầu vào nút
checkButton();

delay(500);
}
```

Hàm loop

```
void checkButton() {
  int pressDelay = 200;
  int lastPressTime = 0;
  int buttonVal = digitalRead(buttonPin);

  // Nếu nút được ấn nhanh 2 lần
  if (btnState == HIGH && (millis() - lastPressTime) < pressDelay) {
    lastPressTime = millis();
    count++;
    if (count > 2)
      count = 2;
    mode = count;
  }
  // Reset bộ đếm số lần bấm
  if ((millis() - pressDelay) > lastPressTime) {
    count = 0;
    lastDebounceTime = millis();
  }

  if (mode == 2) {
    MAX = 12;
  }
  else MAX = 7;
}
```

Hàm đọc nút bấm (checkButton)

2. Giải thích

Chương trình có biến toàn cục MAX đại diện cho số đèn LED tối đa được bật (pin 3-12), ở chế độ bình thường, $MAX = 12$ (10 đèn), ở chế độ tiết kiệm điện, $MAX = 7$ (5 đèn).

Dùng `readLightLevel()` từ thư viện BH1750.h để đọc giá trị từ cảm biến ánh sáng, nếu giá trị lớn hơn 50, tắt tất cả các đèn. Nếu dưới 50, lần lượt bật 3, 6, 10 đèn tùy vào độ tối. Khi bật đèn, kiểm tra xem số đèn bật có vượt quá MAX hay không. Nếu mức ánh sáng dưới 20, bật tối đa số đèn (theo MAX).

Hàm `checkButton()` đóng vai trò xử lý thao tác bấm nút. Khi người dùng bấm nút, kiểm tra xem lần bấm này với lần bấm trước đó có cách nhau quá 200ms hay không, nếu có thì chỉ xem đó là 1 lần bấm và chuyển qua chế độ tiết kiệm điện, nếu không thì coi đó là 2 lần bấm và chuyển qua chế độ thường.