

BÁO CÁO THỰC HÀNH

Môn học: Hệ thống nhúng mạng không dây

Buổi báo cáo: Lab 04

Tên chủ đề: Làm quen với Arduino

GVHD: Nguyễn Văn Bảo

Ngày thực hiện: 02/05/2024

THÔNG TIN CHUNG:

(Liệt kê tất cả các thành viên trong nhóm)

Lớp: NT131.O21.1

Nhóm: 6

STT	Họ và tên	MSSV	Email
1	Lê Huy Hiệp	21522067	21522067@gm.uit.edu.vn
2	Nguyễn Long Vũ	21522800	21522800@gm.uit.edu.vn
3	Nguyễn Thanh Tuấn	21522756	21522756@gm.uit.edu.vn

1. ĐÁNH GIÁ KHÁC:

Nội dung	Kết quả
Tổng thời gian thực hiện bài thực hành trung bình	2 ngày
Link Video thực hiện (nếu có)	Link
Ý kiến (nếu có) + Khó khăn + Đề xuất ...	
Điểm tự đánh giá	10

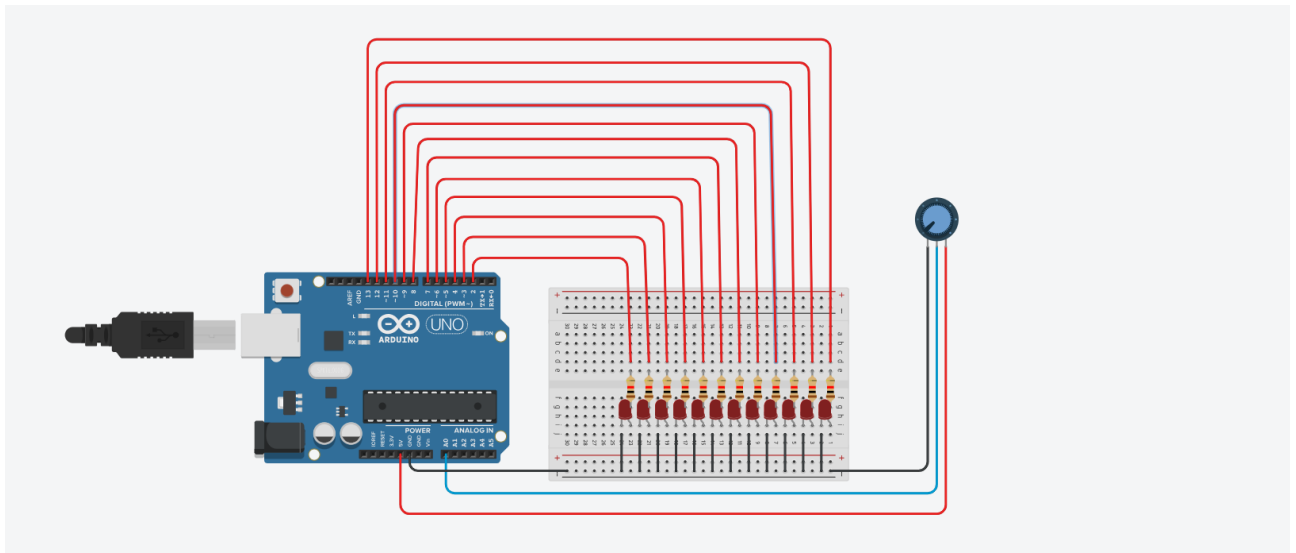
BÁO CÁO CHI TIẾT

Link Video, sơ đồ mạch, code :

<https://drive.google.com/drive/folders/1vP9CZTRyVOyTOtOdyvXfl42CJiAeP17Y?usp=sharing>

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

Sơ đồ mạch:



Code:

```
// cau 1
// C++ code
//
const int potPin = A0; // Chọn chân A0 cho biến trở
const int numLEDs = 12; // Số lượng đèn LED
const int ledPins[numLEDs] = {2,3,4,5,6,7,8,9,10,11,12,13};
void setup()
{
  for (int i = 0; i < numLEDs; i++) {
    pinMode(ledPins[i], OUTPUT);
  }
}
// Khai báo chân A0 là đầu vào
```

```
pinMode(potPin, INPUT);

}

void loop()
{
    int potValue = analogRead(potPin); // Đọc chân A0
    int numActiveLEDs = map(potValue, 0, 1023, 0, numLEDs);

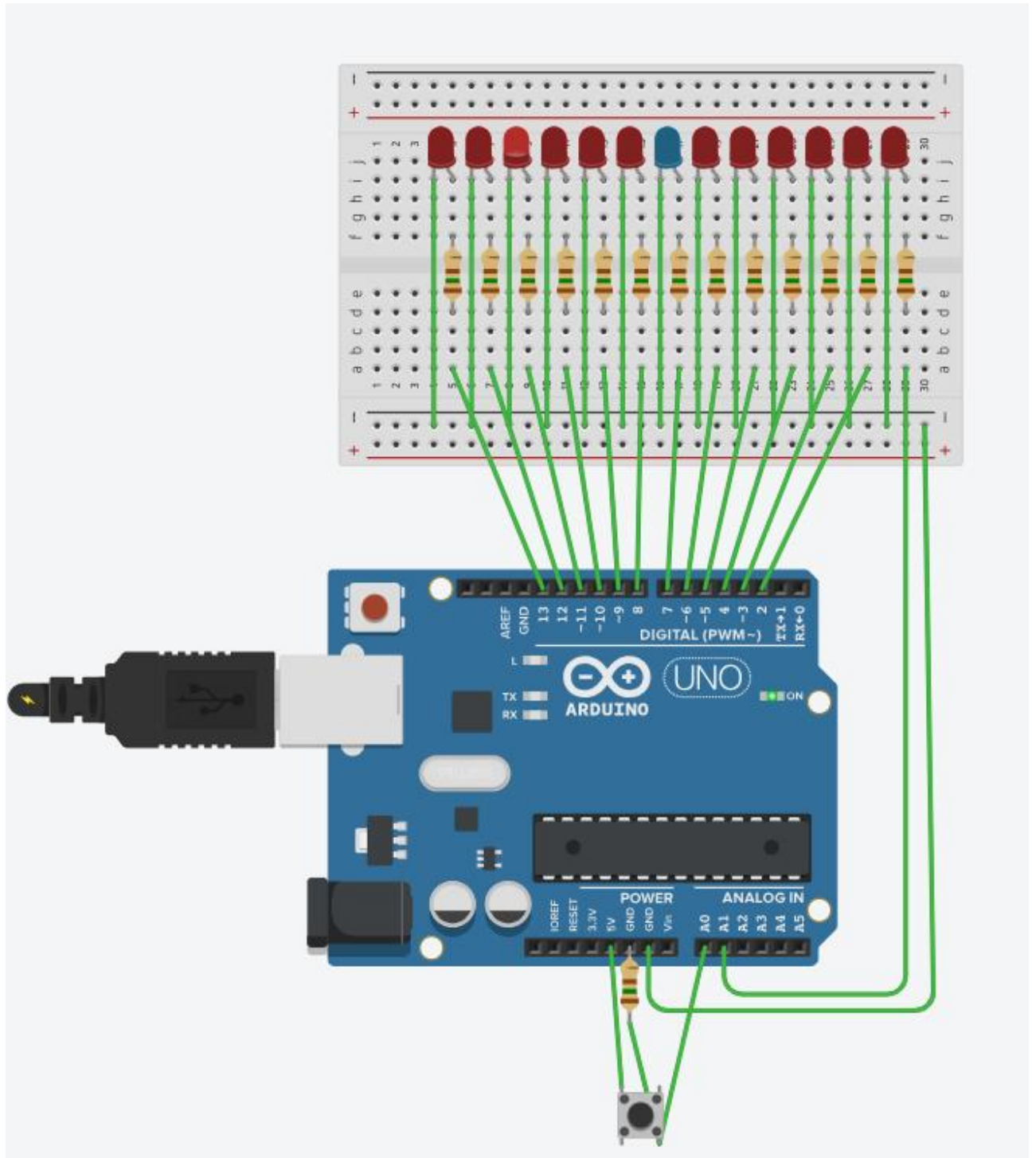
    // Bật các đèn LED tương ứng
    for (int i = 0; i < numLEDs; i++) {
        if (i < numActiveLEDs) {
            digitalWrite(ledPins[i], 1); // Bật đèn LED
        } else {
            digitalWrite(ledPins[i], 0); // Tắt đèn LED
        }
    }
}

// Đợi một khoảng thời gian trước khi đọc lại giá trị
//delay(100);
}
```

Map giá trị của biến trở từ (0,1023) về giá trị (0,12) và gán cho biến numActiveLEDs. Sau đó dùng vòng for để duyệt qua từng đèn trong 12 đèn, nếu giá trị đèn nào thấp hơn giá trị biến trở thì bật sáng đèn đó. Có nghĩa là nếu numActiveLEDs = 0 thì không đèn nào sáng nếu numActiveLEDs từ 1 đến 12 thì sẽ có số đèn sáng tương ứng

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

Sơ đồ mạch:



Code:

```
//cau 2

const int buttonPin = A0;
int led(13) = {13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, A1};
const int differentColorLED = 6;

int currentLed = 0;
int count = 0;
int score = 0;
```

```
int speed = 500;

void proc()
{
    digitalWrite(led(currentLed), HIGH);
    int buttonState = digitalRead(buttonPin);

    if (buttonState == 1) {
        if (currentLed == differentColorLED) {
            score++;
            speed -= 100; // Tăng tốc độ
            Serial.println("true!");
        } else {
            if (score > 0) {
                score--;
                speed += 100; // Giảm tốc đ
            }
            Serial.println("false!");
        }
        Serial.print("score : ");
        Serial.println(score);
        buttonState = 0;
    }

    digitalWrite(led(currentLed), LOW);
    speed = constrain(speed, 100, 1000);
}

void setup() {
    Serial.begin(9600);
    pinMode(buttonPin, INPUT);
    for (int i = 0; i < 13; i++) {
        pinMode(led(i), OUTPUT);
    }
    Serial.print("Score : ");
    Serial.println(score);
}

unsigned long previousDelayMillis = 0;
void loop() {
    unsigned long currentMillis = millis();
    // Hiển thị đèn tại vị trí hiện tại
    digitalWrite(led(currentLed), HIGH);
    delay(speed/2);

    digitalWrite(led(currentLed), LOW);

    proc();
    delay(speed/2);
    count++;
    if (count >= 24) {
        count = 0;
    }

    currentLed = abs(12-count);
}
```

Đặt led đúng là chân thứ 6 và speed ban đầu là 500

Hàm proc() dùng để cộng điểm và tăng tốc độ (thay đổi biến speed) khi người chơi nhấn trúng. Nếu không trúng thì giảm tốc độ. Sau cùng đặt lại trạng thái button = 0

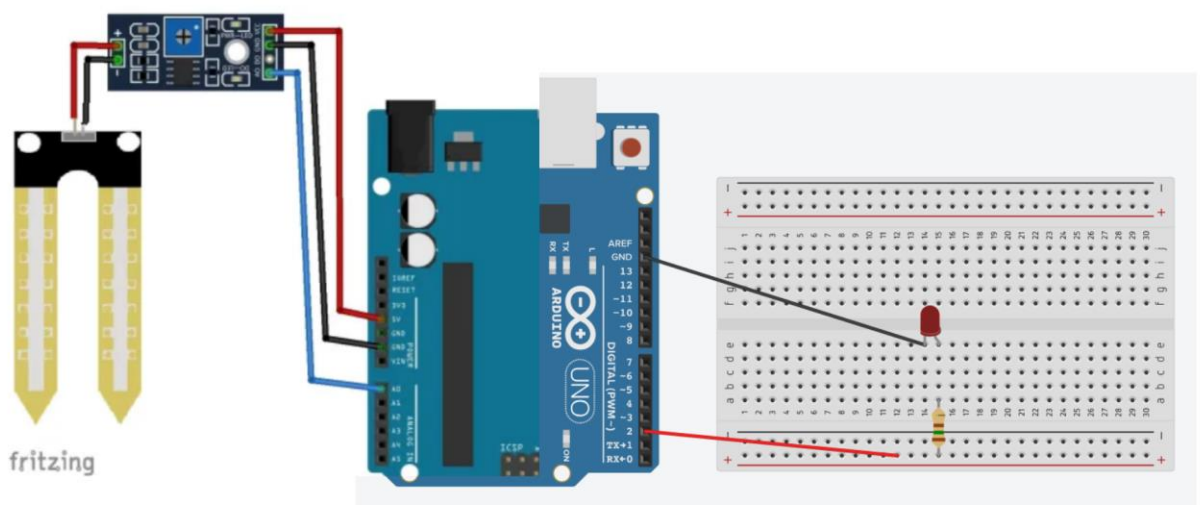
Ở hàm loop:

đưa thời gian hiện tại vào biến currentMillis bằng hàm millis()

Thực hiện tuần tự bật đèn hiện tại sử dụng biến currentLed sau đó tắt đèn đi. Thực hiện hàm proc(). Tăng chỉ số count lên 1 để cập nhật vị trí của đèn led sáng tiếp theo bằng `currentLed = abs(12-count)`; nếu count lớn hơn 24 thì đặt lại count= 0, làm như vậy để led có thể sáng tuần tự từ đèn số 1 đến đèn số 13 sau đó quay ngược lại.

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)

Sơ đồ mạch:



```
//cau 3

// C++ code
//
int maybom = 2;
const int potPin = A0;
void setup()
{
  pinMode(maybom, OUTPUT);
  pinMode(potPin, INPUT);
  Serial.begin(9600); // khai tao gui du lieu qua cong series
}
}
```

```

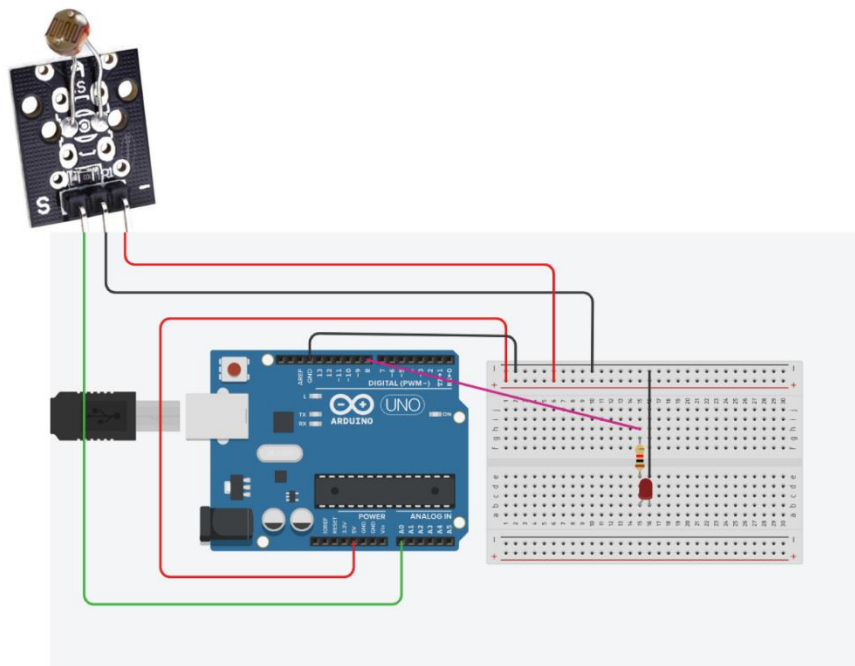
void loop()
{
  int doam = analogRead(potPin);
  int value = map(doam,0,1023,100,0);
  Serial.print("Do am ");
  Serial.print(value);
  Serial.println('%');
  if ( value >= 0 && value <= 30)
    digitalWrite(maybom, 1);
  else
    digitalWrite(maybom, 0);
  delay(500);
}

```

Set chân 2 là chân máy bơm. Đọc giá trị cảm biến độ ẩm thông qua chân analog và map về biến value có giá trị 0% đến 100%. Nếu độ ẩm từ 0% đến 30% thì máy bơm bật (Đèn led bật), nếu không thì máy bơm tắt (Đèn led tắt)

4. Đèn vườn tự động, khi trời tối thì đèn tự động bật lên. Sử dụng cảm biến ánh sáng hoặc quan trở để nhận biết trời sáng hay tối dựa vào đó thì bật hoặc tắt đèn.

Sơ đồ mạch:



Code:

```

// cau 4

void setup()
{
  Serial.begin(9600);
  pinMode(8,OUTPUT);
  pinMode(A0,INPUT);
}

```

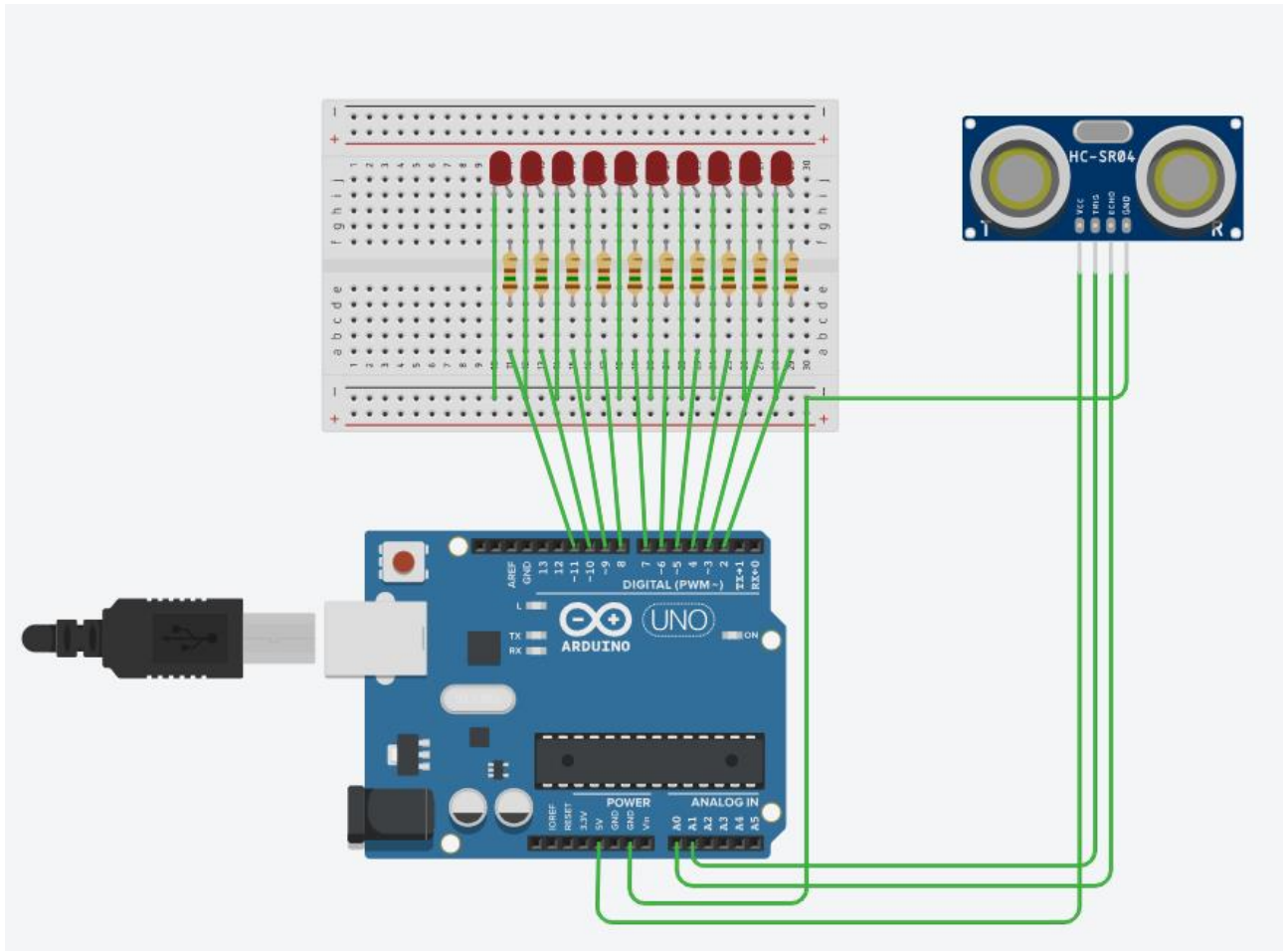
```
}  
void loop()  
{  
  int value = analogRead(A0);  
  if (value <= 300)  
    digitalWrite(8,1);  
  else  
    digitalWrite(8,0);  
  
  Serial.println(value);  
  delay(500);  
}
```

Cảm biến ánh sáng có 1 chân analog, truyền dữ liệu vào chân A0

Và đọc giá trị có được vào biến value. Trong quá trình đọc thấy khi che cảm biến lại giá trị thường bé hơn 150 tuy nhiên để chắc chắn hơn em sẽ đặt giá trị là 300, nếu value bé hơn hoặc bằng 300 thì trời tối => đèn bật, ngược lại là trời sáng => đèn tắt

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).

Sơ đồ mạch:



Code:

```
// cau 5

const int trig = A1; // chân trig của HC-SR04
const int echo = A0; // chân echo của HC-SR04

void setup()
{
  Serial.begin(9600); // giao tiếp Serial với baudrate 9600
  pinMode(trig,OUTPUT); // chân trig sẽ phát tín hiệu
  pinMode(echo,INPUT); // chân echo sẽ nhận tín hiệu
  for (int i = 2; i < 12; ++i) {
    pinMode(i, OUTPUT);
  }
}

void loop()
{
  unsigned long duration; // biến đo thời gian
  float distance; // biến lưu khoảng cách

  /* Phát xung từ chân trig */
  digitalWrite(trig,0); // tắt chân trig
  delayMicroseconds(2);
  digitalWrite(trig,1); // phát xung từ chân trig
  delayMicroseconds(5); // xung có độ dài 5 microseconds
  digitalWrite(trig,0); // tắt chân trig
```

```
/* Tính toán thời gian */
// Đo độ rộng xung HIGH ở chân echo.
duration = pulseIn(echo,HIGH);
// Tính khoảng cách đến vật.
distance = int(duration/2/29.412);
    int brightness = map(distance, 325, 0, 0, 12);
    for (int i = 2; i <= 11; i++) {
        if (i <= brightness) {
            digitalWrite(i, HIGH);
        } else {
            digitalWrite(i, LOW);
        }
    }
    Serial.print(distance);
    Serial.println("cm");
}
```

Chân trig của HC-SR04 sẽ được dùng để nhận xung sau đó phát ra sóng siêu âm => gán vào A1 và set OUTPUT

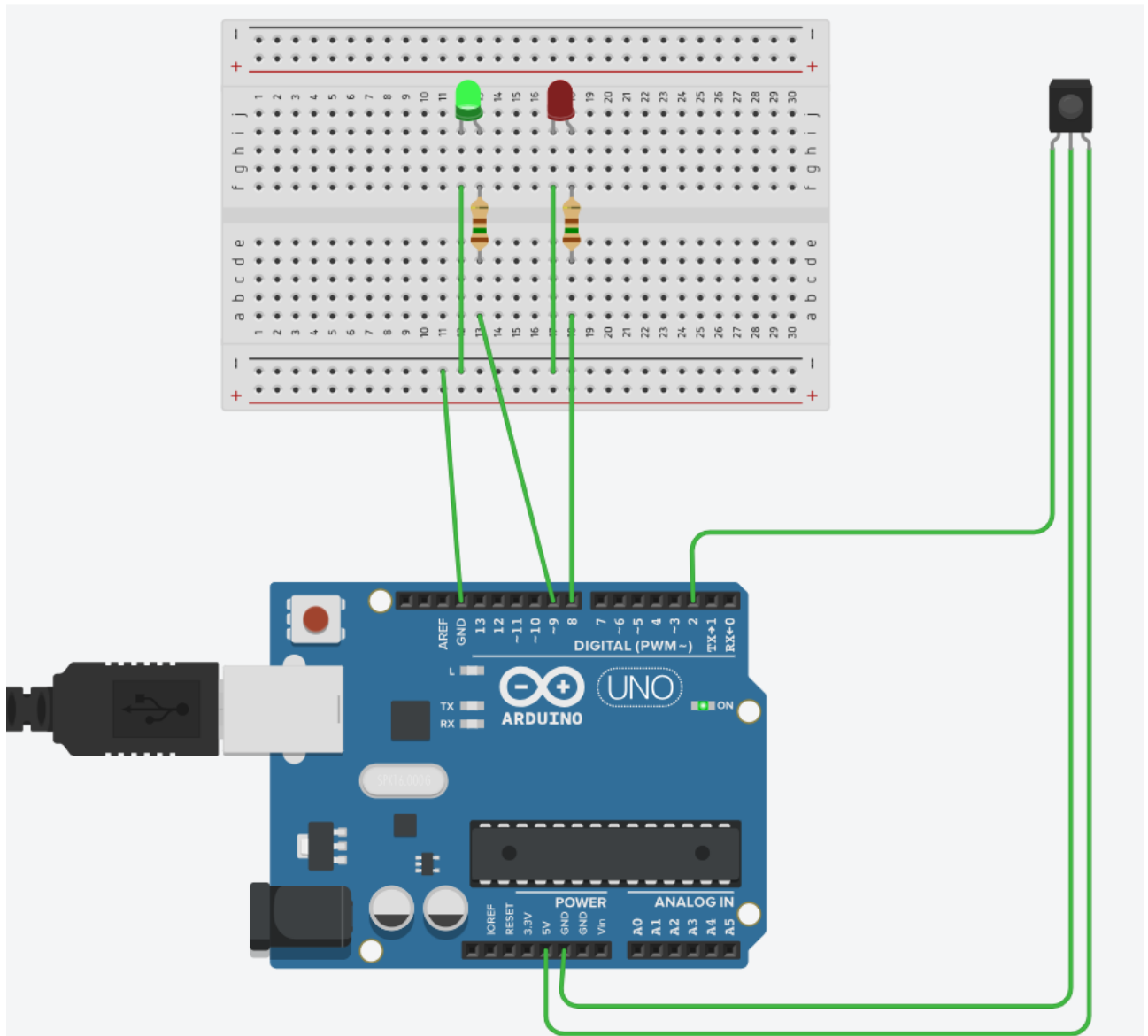
Chân echo sẽ nhận lại sóng phản xạ và gửi về => gán vào A0 và set INPUT

Đầu tiên tạo xung có độ dài 5 microseconds. Sau đó đo thời gian phản hồi về bằng hàm pulseIn do khi bắt đầu phát xung đi thì cảm biến sẽ đặt chân echo thành mức high cho đến khi nhận được sóng phản hồi về, vì vậy ta sẽ dùng pulseIn(echo,HIGH) để đo thời gian khi chân bắt đầu duy trì từ mức High về mức Low.

Sau đó tính khoảng cách bằng cách lấy thời gian sóng âm đi và về (duration) chia đôi và sau đó chia cho 29412 microseconds/cm (thời gian cần để đi được 1cm) sẽ ra được khoảng cách.

6. Hệ thống cảnh báo cháy, sử dụng cảm biến lửa, 2 đèn LED khác màu.
Khi có đám cháy, 2 đèn LED sẽ thay phiên nhau chớp tắt liên tục.

Sơ đồ mạch:



Code:

```
// cau 6
// cau 6 canh bao chay
// Khai báo các chân GPIO
const int sensor_pin = 2;
const int led1_pin = 8; // Màu đèn LED 1
const int led2_pin = 9; // Màu đèn LED 2

bool fire_detected = false;

void setup() {
  // Thiết lập chân GPIO
  pinMode(sensor_pin, INPUT);
  pinMode(led1_pin, OUTPUT);
  pinMode(led2_pin, OUTPUT);

  // Khởi động Serial Monitor để theo dõi trạng thái
```

```
Serial.begin(9600);
}

void loop() {
  // Đọc trạng thái của cảm biến lửa
  int sensor_state = digitalRead(sensor_pin);

  if (sensor_state == HIGH) {
    Serial.println("Fire detected!");
    fire_detected = true;
  } else {
    fire_detected = false;
  }

  // Thực hiện cảnh báo nếu phát hiện cháy
  if (fire_detected) {
    alert();
  }

  delay(500);
}

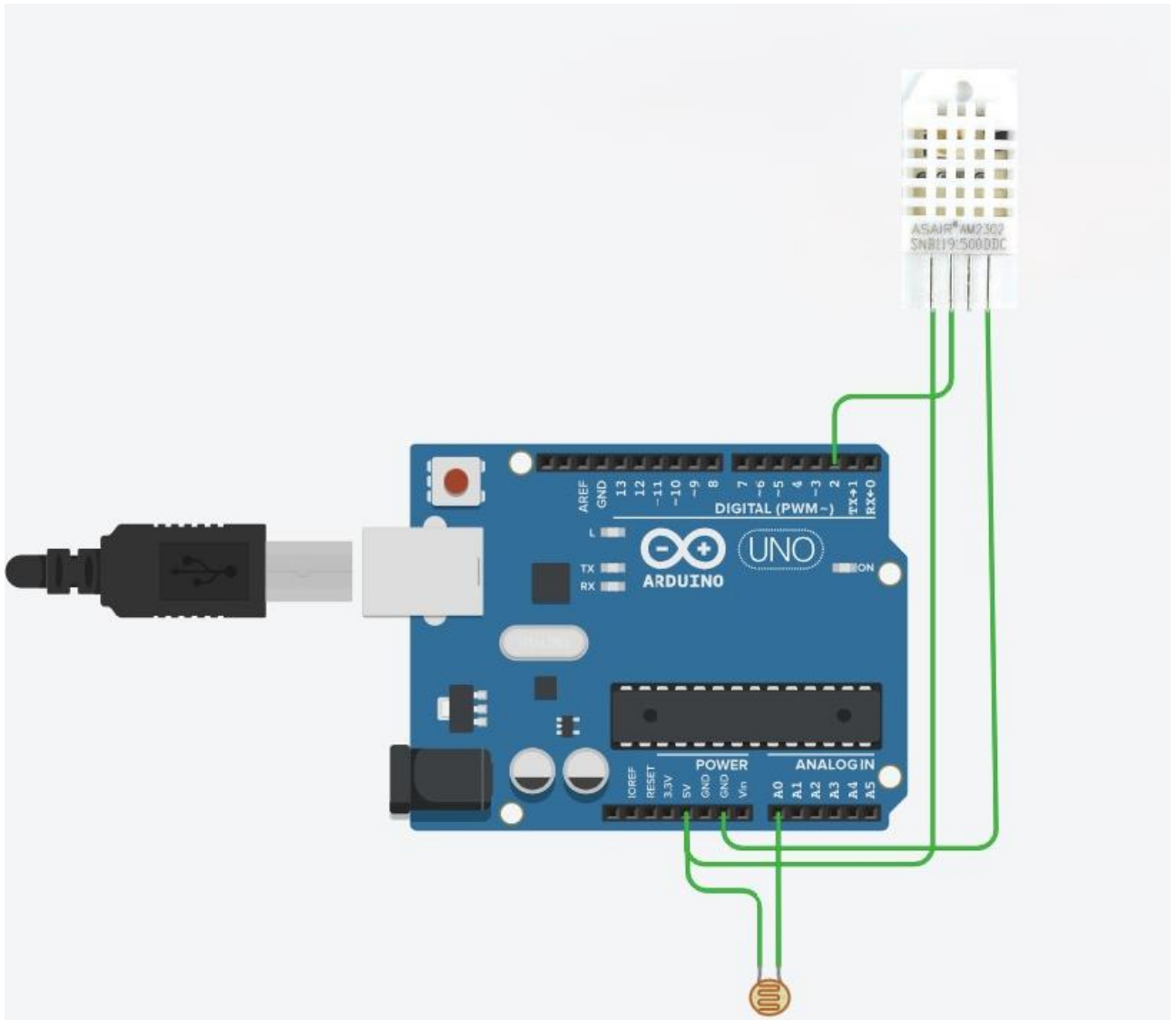
void alert() {
  digitalWrite(led1_pin, HIGH);
  delay(500);
  digitalWrite(led1_pin, LOW);
  delay(500);
  digitalWrite(led2_pin, HIGH);
  delay(500);
  digitalWrite(led2_pin, LOW);
  delay(500);
}
```

Đưa giá trị chân digital của cảm biến vào biến sensor_state nếu ở mức high => cháy thực hiện bật và tắt hai đèn đỏ và xanh. Nếu ngược lại thì bỏ qua

*Note: Do phần demo trong phòng thực hành không được sài bật lửa nên nhóm em chỉ quay mạch đã lắp ráp ☹.

7. Đọc giá trị cảm biến nhiệt độ, độ ẩm và ánh sáng và in ra serial kèm theo đơn vị.

Sơ đồ mạch:



Code:

```
// cau 7
#include <DHT22.h>
#define DHTPIN 2
#define DHTTYPE DHT22
#include <Wire.h>
#include <BH1750.h>
#define DHTTYPE DHT

DHT22 dht(DHTPIN);

void setup() {
  Serial.begin(9600);
  Wire.begin();

  pinMode(A0,INPUT);

  Serial.println("Setup complete.");
}

void loop() {
```

```
int value = analogRead(A0);
Serial.println(value);
int valuelight = map(value, 0, 1023, 0, 100);

float humDHT = dht.getHumidity(); // Read humidity from DHT sensor
float tempDHT = dht.getTemperature(); // Read temperature from DHT sensor

// Print information to Serial Monitor
Serial.print("Humidity: ");
Serial.print(humDHT);
Serial.println("%");

Serial.print("Temperature: ");
Serial.print(tempDHT);
Serial.println("°C");

Serial.print("Light: ");
Serial.print(valuelight);
Serial.println("%");

delay(1000);
}
```

Dùng thư viện DHT22.h cho cảm biến nhiệt độ độ ẩm DHT22

Khởi tạo đối tượng dht ứng với chân số 2 để lấy data

Dùng hai hàm có sẵn của thư viện dht.getTemperature() và dht.getHumidity() lấy giá trị nhiệt độ và độ ẩm

Đọc giá trị ánh sáng sử dụng cảm biến tương tự câu 4, lấy giá trị vào chân A0

YÊU CẦU CHUNG

1) Đánh giá

- Chuẩn bị tốt các yêu cầu đặt ra trong bài thực hành.
- Sinh viên hiểu và tự thực hiện được bài thực hành, trả lời đầy đủ các yêu cầu đặt ra.
- Nộp báo cáo kết quả chi tiết những đã thực hiện, quan sát thấy và kèm ảnh chụp màn hình kết quả (*nếu có*); giải thích cho quan sát (*nếu có*).
- Sinh viên báo cáo kết quả thực hiện và nộp bài.

2) Báo cáo

- File **.PDF** hoặc **.docx**. Tập trung vào nội dung, giải thích.
- Nội dung trình bày bằng Font chữ **Times New Romans/** hoặc font chữ của mẫu báo cáo này (UTM Avo)– cỡ chữ 13. Canh đều (Justify) cho văn bản. Canh giữa (Center) cho ảnh chụp.
- Đặt tên theo định dạng: LabX_MSSV1. (trong đó X là Thứ tự buổi Thực hành).

Ví dụ: Lab01_21520001

- Nộp file báo cáo trên theo thời gian đã thống nhất tại courses.uit.edu.vn.

Bài sao chép, trễ, ... sẽ được xử lý tùy mức độ vi phạm.

HẾT