



NỘI DUNG KHÓA HỌC LẬP TRÌNH ARDUINO

❖ Buổi 1 – Sáng ngày 01/7/2023

- *Bài 1: Giới thiệu sơ lược về Arduino*
 - Phần cứng Arduino UNO.
 - Phần mềm lập trình Arduino IDE.
- *Bài 2: Lập trình điều khiển LED đơn và nút nhấn.*

❖ Buổi 2 – Chiều ngày 01/7/2023

- *Bài 3: Lập trình điều khiển với cảm biến nhiệt độ - độ ẩm.*

❖ Buổi 3 – Sáng ngày 09/7/2023

- *Bài 4: Lập trình điều khiển với cảm biến siêu âm.*
- *Bài 5: Lập trình điều khiển với RFID.*

❖ Buổi 4: Chiều ngày 09/7/2023

- Thảo luận, tổng kết lớp.
- Phát chứng nhận.

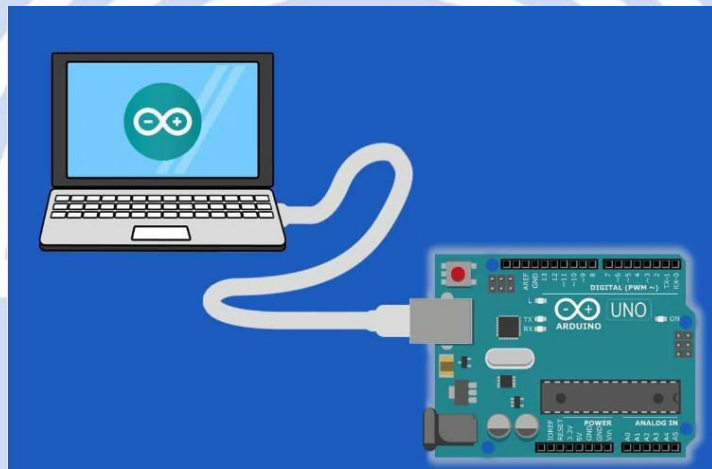
BÀI 1: GIỚI THIỆU SƠ LƯỢC VỀ ARDUINO

1.1. Arduino là gì?

Arduino là một bo mạch vi điều khiển do một nhóm giáo sư và sinh viên nước Ý thiết kế và đưa ra đầu tiên vào năm 2005. Mạch Arduino được sử dụng để lập trình và điều khiển nhiều thiết bị khác nhau như: điều khiển đèn, động cơ, cảm biến nhiệt độ... Ngoài ra mạch còn có khả năng liên kết với nhiều module khác nhau như module đọc thẻ từ RFID, điều khiển từ xa, gọi điện và nhắn tin qua điện thoại với Sim800L, điều khiển thiết bị qua mạng Internet với Ethernet shield.

Phần cứng của Arduino được thiết kế với vi điều khiển trung tâm là chip AVR Atmel 8bit, hoặc ARM, Atmel 32-bit,... Hiện tại, Arduino có rất nhiều phiên bản được bán trên thị trường. Tuy nhiên phiên bản thường được sử dụng nhiều nhất là Arduino Nano, Arduino Uno và Arduino Mega.

Để sử dụng được bo mạch Arduino thì người dùng cần phải có thêm phần mềm Arduino IDE cài đặt trên máy vi tính.



Hình 1.1: Minh họa kết nối Arduino với máy vi tính.



Hình 1.2: Một số bo mạch Arduino.

Một trong những ưu điểm nổi bật nhất của Arduino là tính tiện dụng và giá thành rẻ. Người dùng rất dễ dàng mua được bo mạch Arduino từ các trang bán hàng online. Hơn nữa, với mã nguồn mở làm cho việc lập trình điều khiển với Arduino trở nên rất dễ dàng, phù hợp với nhiều đối tượng

người dùng khác nhau. Với Arduino, người dùng không cần phải có kiến thức lập trình điều khiển chuyên sâu cũng có thể làm được.

Bên cạnh đó, Arduino cũng có nhược điểm là rất nhạy với nhiễu, khả năng chống nhiễu thấp làm cho Arduino không thích ứng trong môi trường công nghiệp. Vì thế, Arduino chỉ phù hợp với các dự án vừa và nhỏ.

1.2. Học Arduino cần những gì?

- ✓ Điều đầu tiên người dùng cần có là một bo mạch Arduino. Có nhiều loại Arduino đang được bán trên thị trường, người dùng cần chọn loại bo mạch Arduino phù hợp với dự án của mình, thường là lựa chọn dựa trên số lượng kết nối, khả năng kết nối và khả năng lưu trữ dữ liệu.
- ✓ Thứ hai, cần chuẩn bị một máy vi tính để bàn hay laptop.
- ✓ Thứ 3 là người dùng cần có kiến thức cơ bản về điện tử, tin học và tư duy lập trình.
- ✓ Cuối cùng là cần có sự đam mê khoa học, sự yêu thích tìm tòi và học hỏi.

1.3. Điện tử cơ bản

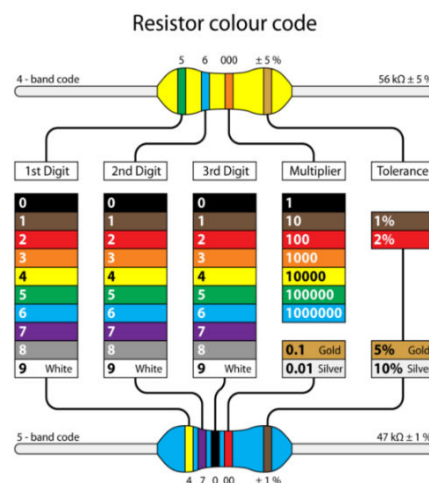
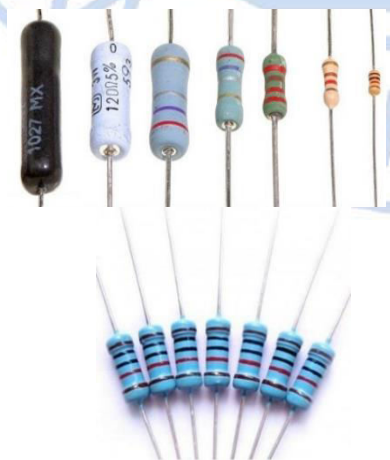
Phần này sẽ giới thiệu một số kiến thức cơ bản về điện tử và linh kiện điện tử được sử dụng trong khóa học.

❖ *Mức logic kỹ thuật số:*

- Mức logic 0: ký hiệu là [0], biểu thị cho điện áp 0 VDC, là cực âm của nguồn điện, trên bo mạch thường được ký hiệu là “GND”. Trong lập trình Arduino, mức [0] được biểu thị là bởi chữ “LOW”.
- Mức logic 1: ký hiệu là [1], biểu thị cho điện áp 5 VDC, là cực dương của nguồn điện, trên bo mạch thường được ký hiệu là “5v”. Trong lập trình Arduino, mức [1] được biểu thị là bởi chữ “HIGH”.
- Lưu ý: đối với một số loại chip, mức [1] được tạo ra bởi điện áp 3.3V.

❖ *Điện trở:*

Có 2 loại điện trở thông dụng là loại có 4 vòng màu và loại có 5 vòng màu. Dựa vào các vòng màu của điện trở, ta sẽ xác định được trị số của điện trở. Cách xác định giá trị của điện trở được mô tả như Hình 1.3.





Biến trở:

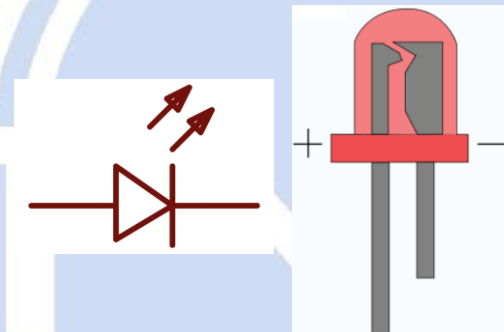
Biến trở là loại điện trở có thể thay đổi trị số tùy theo sự điều chỉnh của người dùng. Biến trở được sản xuất với nhiều hình dáng và trị số khác nhau. Trong khóa học này, biến trở được sử dụng sẽ có dạng như Hình 1.4.



Hình 1.4: Biến trở



LED: là một loại đi-ốt có thể phát ra ánh sáng khi có dòng điện chạy qua nó. Led được sản xuất với nhiều hình dạng, kích cỡ và màu sắc.



Hình 1.5: Hình dạng của Led.

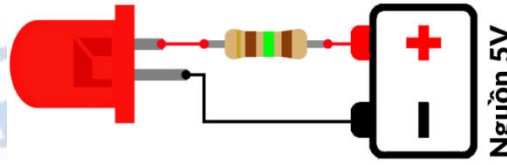
Để Led có thể phát sáng tốt nhất, ta phải cấp nguồn điện cho Led đúng định mức như Hình 1.6.



LED	ĐỎ	VÀNG	TRẮNG	XANH LÁ	XANH DƯƠNG
Điện áp	1.8-2v	1.8-2v	2.8-3.2v	2.8-3.2v	2.8-3.2v
Dòng điện	10-20mA	10-20mA	10-20mA	10-20mA	10-20mA
Cỡ LED	5 mm	5 mm	5 mm	5 mm	5 mm

Hình 1.6: Điện áp và dòng điện định mức cho Led.

Tuy nhiên, vì Arduino sử dụng điện áp là 5v (vượt quá định mức của Led), nên để an toàn cho Led khi sử dụng ta phải gắn nối tiếp với Led một điện trở như Hình 1.6.



Hình 1.6: Gắn điện trở hạn dòng cho Led.

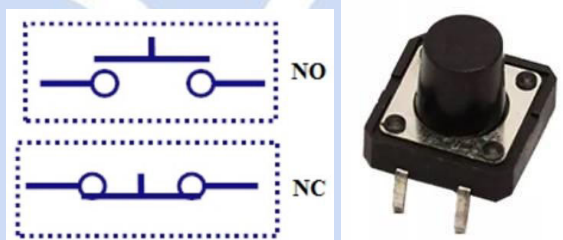
Điện trở này có tác dụng làm giảm bớt điện áp và dòng điện chạy qua Led. Giá trị điện trở được tính như sau:

Ví dụ: nếu sử dụng Led màu đỏ ($V_{\text{Led}} = 2\text{V}$, $I_{\text{Led}} = 15\text{mA}$) thì $R = \frac{5\text{V} - 2\text{V}}{15\text{mA}} = \frac{5 - 2}{0.015} = 200\Omega$.

Thông thường chúng ta sẽ chọn điện trở để bảo vệ cho một con Led sẽ có trị số vào khoảng $220\Omega - 1K\Omega$.

❖ **Nút nhấn:**

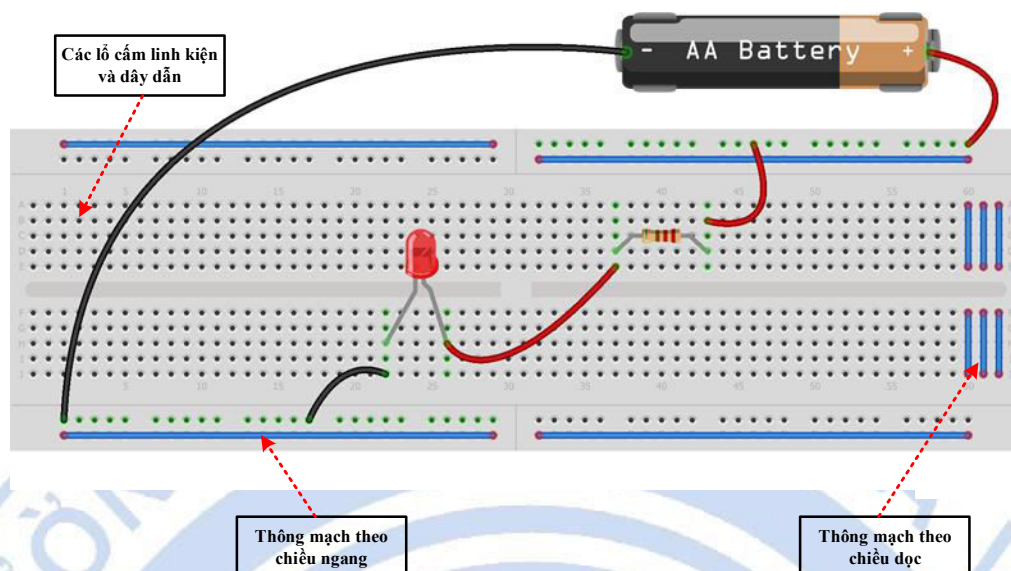
Nút nhấn có nhiều hình dạng lớn nhỏ, màu sắc khác nhau, được phân làm các loại như: thường đóng, thường hở, tự giữ và không tự giữ.



Hình 1.7: Nút nhấn

❖ **Breadboard (hay còn gọi là Testboard)**

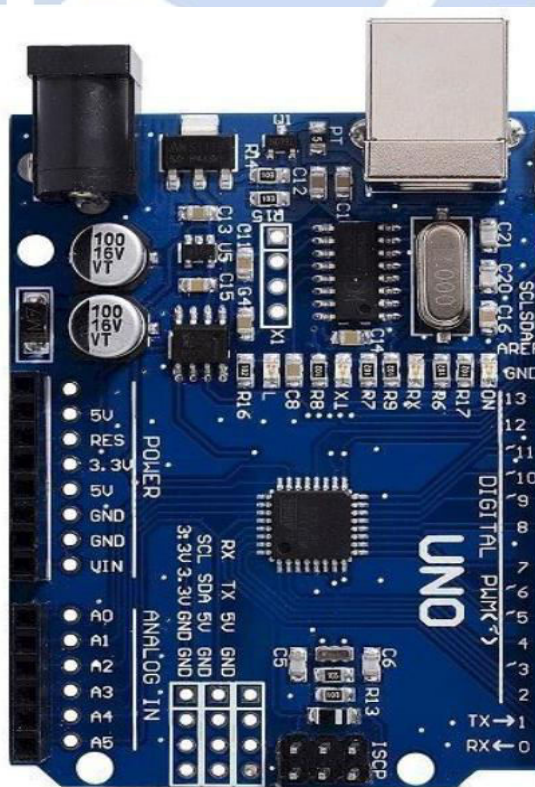
Breadboard là một bảng mạch điện với các dây điện được kết nối sẵn theo chiều ngang và chiều dọc, dùng để kết nối các linh kiện điện tử với nhau để tạo thành mạch điện ứng dụng nào đó. Việc sử dụng Breadboard chỉ là tạm thời, thường là trong quá trình học tập hoặc thử nghiệm mạch điện, cho nên Breadboard còn có tên gọi khác là Testboard. Cấu trúc của Breadboard được mô tả như hình 1.8.



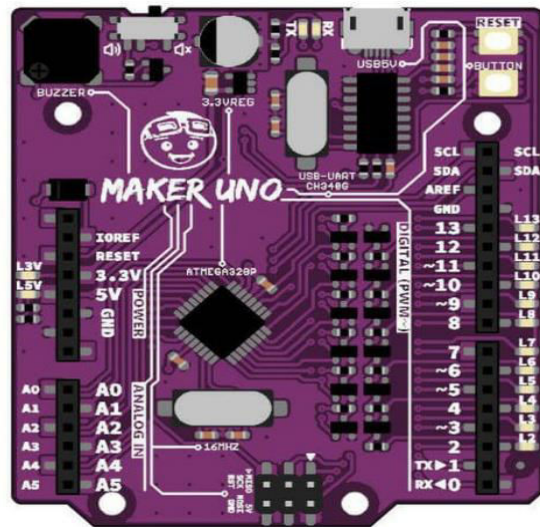
Hình 1.8: Cấu trúc Breadboard

1.4. Giới thiệu phần cứng Arduino UNO

Trong khóa học này, các bạn học viên sẽ được trải nghiệm lập trình điều khiển với bo mạch Arduino Uno, với 2 dạng bo mạch như hình 1.9 và phần cứng được mô tả như hình 1.10.



(a)



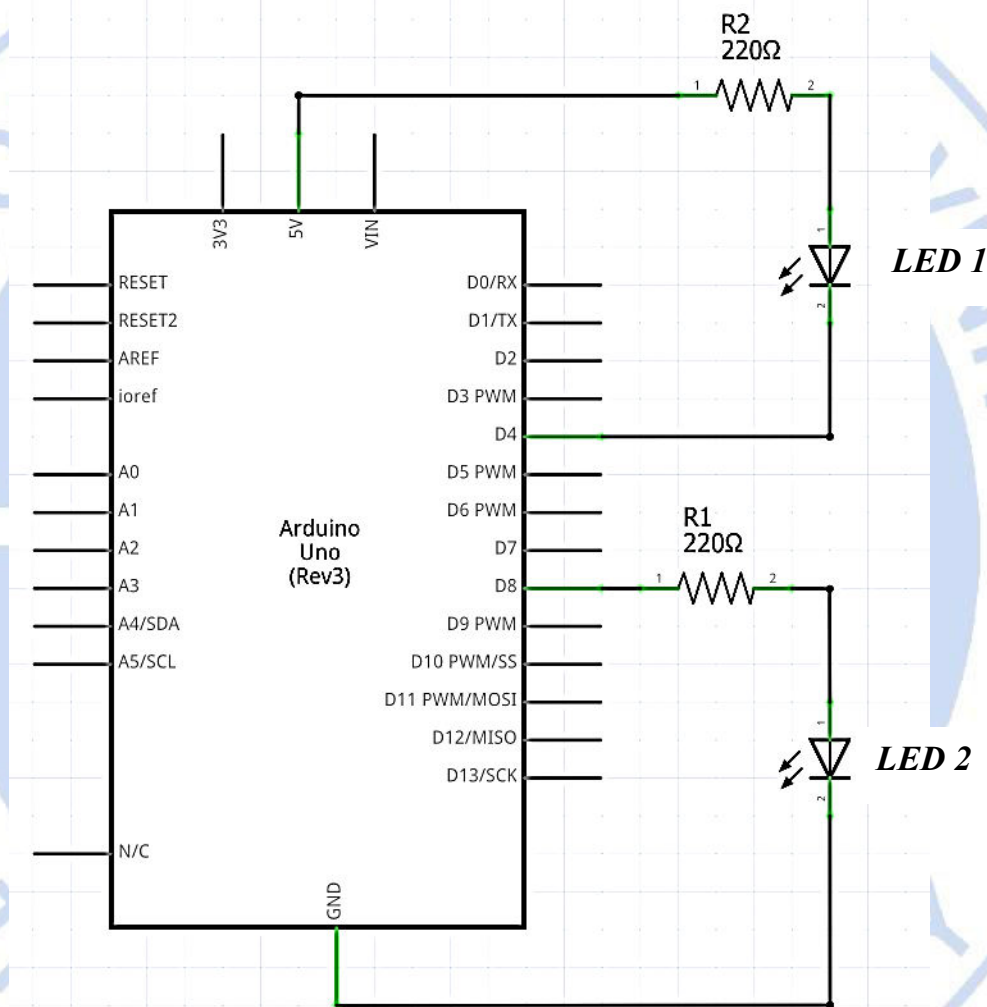
(b)
Hình 1.9: (a) – Arduino UNO
(b) – Marker UNO

Trang 8

❖ **Điều khiển thiết bị với mức logic [0] và [1]:**

Các chân input / Output của Arduino chỉ có thể làm việc với mức điện áp tối đa là 5v DC. Do đó, người sử dụng cần hiểu rõ mức điện áp định mức của các thiết bị sẽ kết nối với Arduino để không làm hư hỏng các linh kiện.

Có 2 dạng mạch kết nối thường được sử dụng với vi điều khiển là: mạch điều khiển mức [1] và mạch điều khiển mức [0] như hình 1.11.



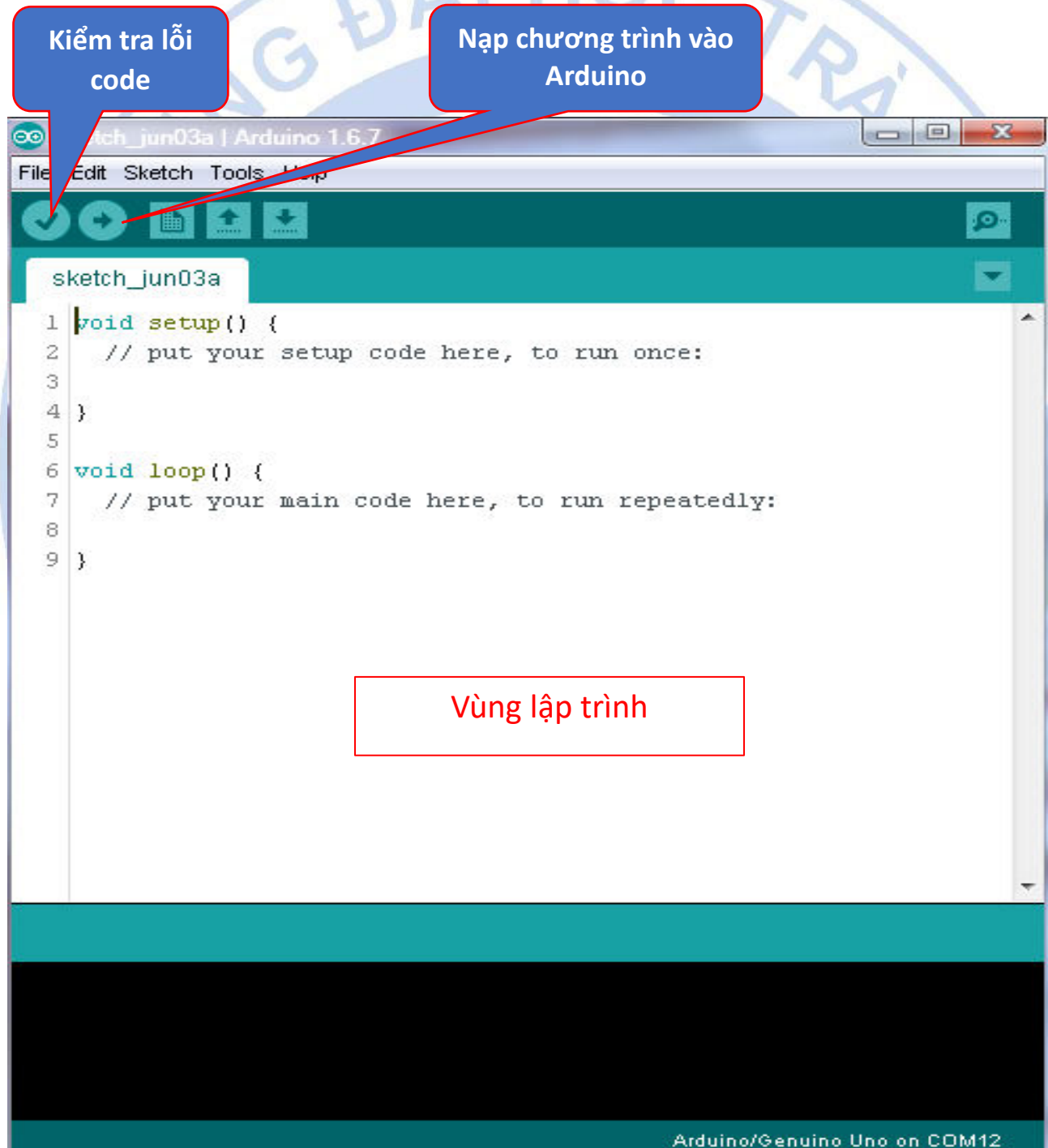
Hình 1.11: Kết nối Led với Arduino

Trong Hình 1.11, LED 1 được kết nối theo kiểu điều khiển mức [0], LED 2 được kết nối theo kiểu điều khiển mức [1]. Để LED 1 sáng thì Arduino phải xuất mức [0] ra chân số D4, muốn LED 1 tắt thì Arduino phải xuất mức [1] ra chân số D4. Ngược lại, Để LED 2 sáng thì Arduino phải xuất mức [1] ra chân số D8, muốn LED 2 tắt thì Arduino phải xuất mức [0] ra chân số D8.

1.5. Phần mềm Arduino IDE

Để lập trình cho Arduino, người dùng cần cài đặt phần mềm Arduino IDE, đây là phần mềm miễn phí. Người dùng có thể download và cài đặt Arduino IDE tại trang web: <https://www.arduino.cc/>

❖ *Giao diện phần mềm:*



Hình 1.12: Giao diện phần mềm Arduino



Chọn loại Arduino làm việc: sau khi mở chương trình Arduino IDE lên, người dùng tiến hành chọn loại Arduino đang sử dụng, ví dụ chọn Arduino Uno bằng cách:

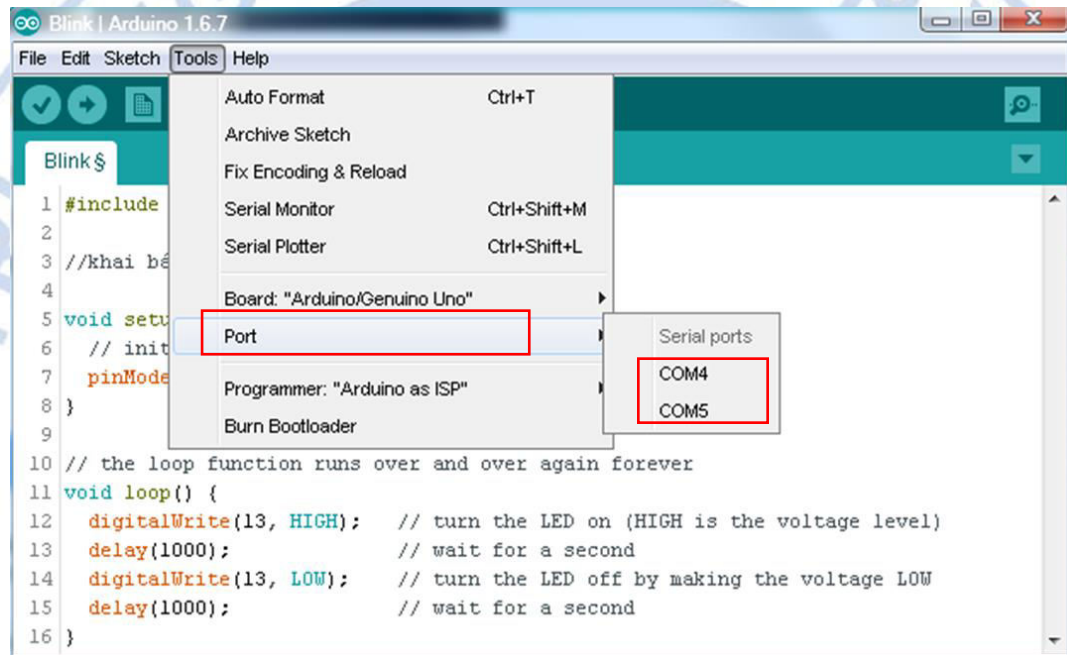
Vào **Tools / Board / Arduino Uno**



Tìm cổng kết nối Arduino với máy tính:

Vào **Tools / Port/ COMxxx**

Với xxx là số của cổng COM đã kết nối. Nếu Port bị ẩn mờ tức là Arduino chưa kết nối được với máy tính, lúc này người dùng cần kiểm tra lại dây cáp hoặc phần mềm driver USB cho Arduino.



Hình 1.13: Chọn cổng kết nối

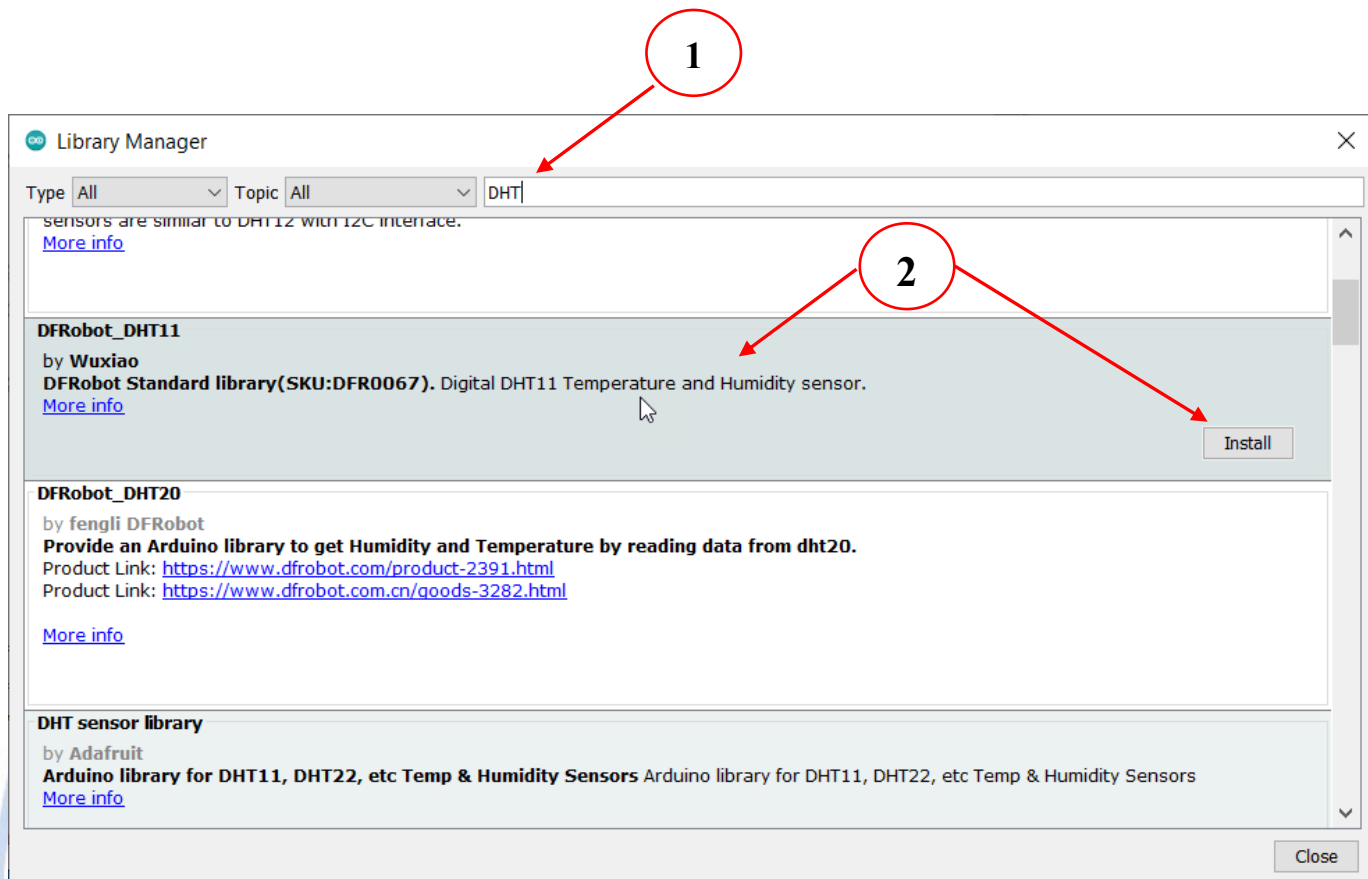


Cài đặt thêm thư viện cho Arduino IDE

Công cụ lập trình Arduino IDE đã cung cấp sẵn nhiều thư viện code mẫu hỗ trợ người dùng. Tuy nhiên, với một số ứng dụng nào đó, người dùng cần download và cài thêm thư viện ví dụ như RFID, cảm biến nhiệt độ DHT...

Có 2 cách cài đặt thêm thư viện:

- Cách 1: vào **Sketch / Include Library / Add . Zip Library...**
Với cách này, người dùng cần download file thư viện về lưu sẵn trên máy tính.
- Cách 2: vào **Sketch / Include Library / Manage Libraries...** gõ tên thư viện ô tìm kiếm
Với cách này, Arduino IDE sẽ tự tìm thư viện trực tiếp từ phần mềm, người dùng chỉ cần chọn lựa thư viện thích hợp và nhấp **“Install”** như hình Hình 1.14.

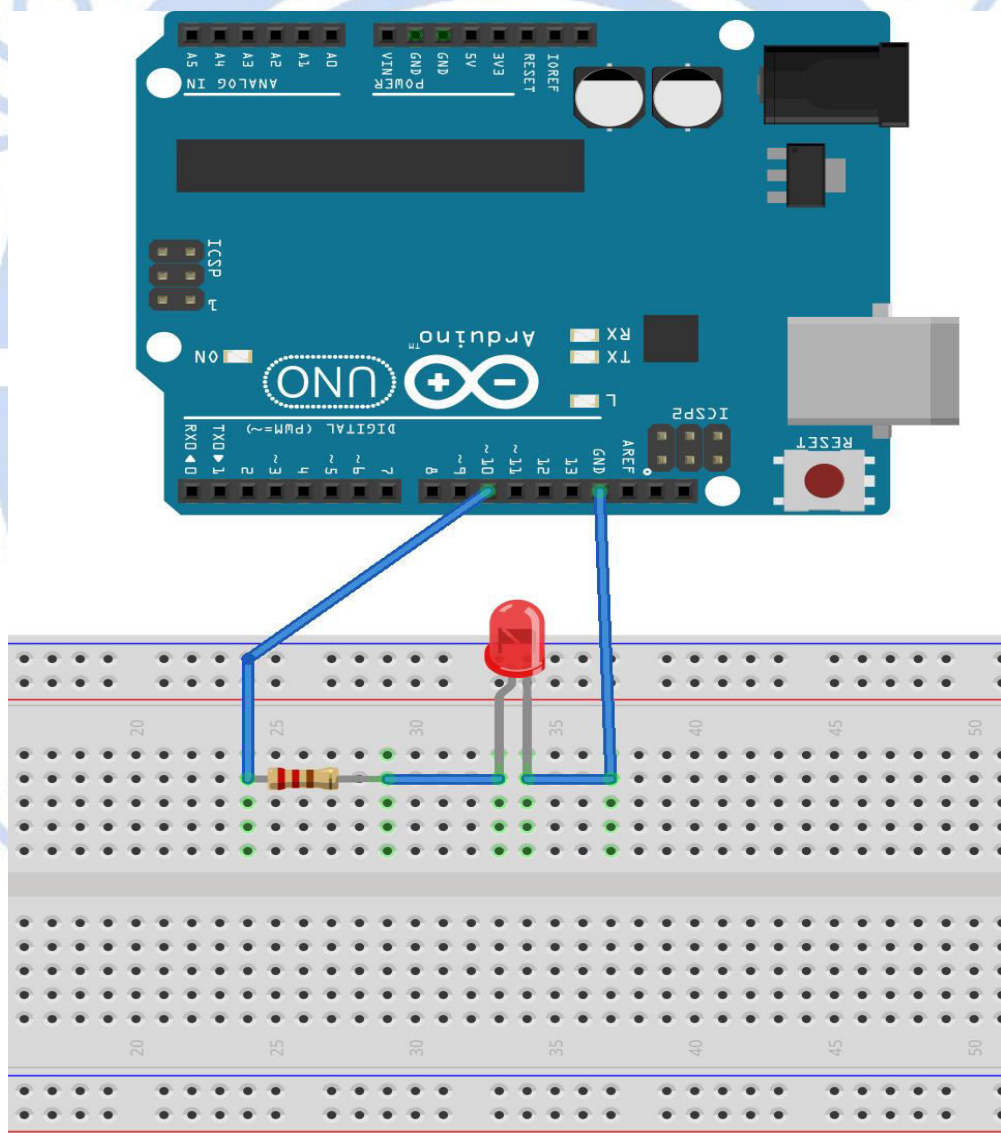


Hình 1.14: Cài thư viện với Manage Libraries

BÀI 2: LẬP TRÌNH ĐIỀU KHIỂN LED ĐƠN VÀ NÚT NHẤN

2.1. Viết chương trình chớp tắt một Led

- ❖ **Yêu cầu:** Led chớp tắt liên tục với thời gian sáng là 1s, thời gian tắt là 1s.
- ❖ **Chuẩn bị linh kiện:**
 - 1 Led đơn.
 - 1 điện trở 220Ω.
 - Dây cắm.
 - Testboard
- ❖ **Kết nối mạch như Hình 2.1 như sau:**



Hình 2.1: Mạch chớp tắt Led.



❖ **Chương trình mẫu:**

```
//-----  
  
void setup()  
{  
    pinMode(10,OUTPUT);  
}  
void loop()  
{  
    digitalWrite(10,LOW);  
    delay(1000);  
    digitalWrite(10,HIGH);  
    delay(1000);  
}  
  
//-----
```

❖ **Bài tập:** viết chương trình chớp tắt led với thời gian chớp tắt là 500ms; 200ms; 100ms.

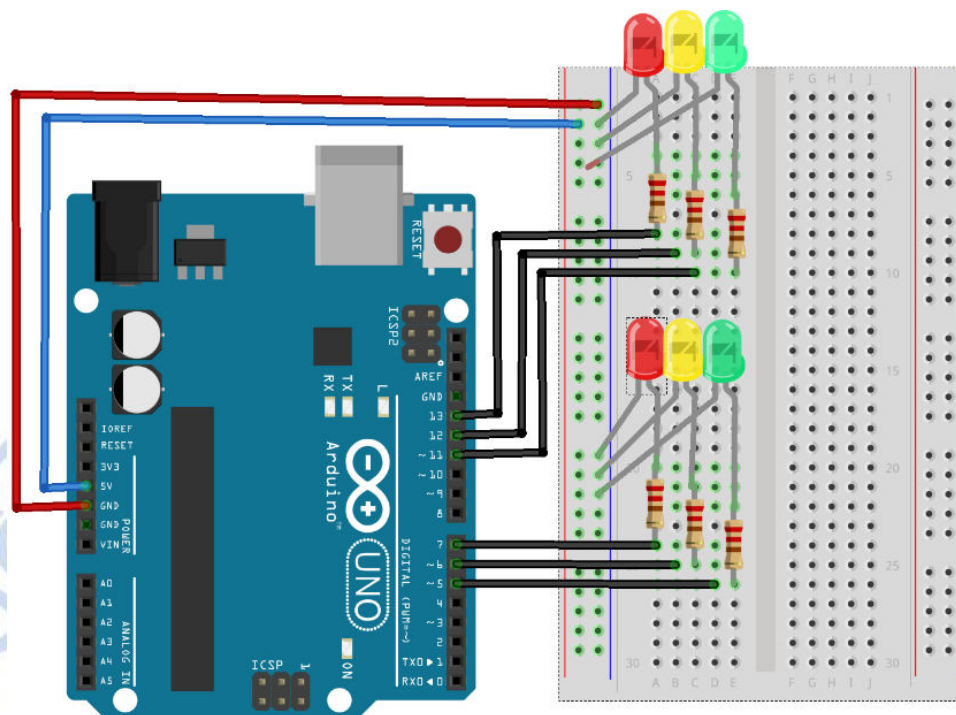
2.2. Viết chương trình điều khiển đèn giao thông ngã tư.

❖ **Yêu cầu:** thời gian sáng đèn xanh là 20s; đỏ 22s vàng 2s.

❖ **Chuẩn bị linh kiện:**

- 2 module đèn giao thông
- Dây cắm.
- Testboard

❖ **Kết nối mạch như Hình 2.2 như sau:**



Hình 2.2: Mạch đèn giao thông.

❖ **Chương trình mẫu:**

```
// -----
int redR1 = 13;
int yellowY1 = 12;
int greenG1 = 11;

int redR2 = 7;
int yellowY2 = 6;
int greenG2 = 5;

void setup()
{
    pinMode(redR1,OUTPUT);
    pinMode(yellowY1,OUTPUT);
    pinMode(greenG1,OUTPUT);
    pinMode(redR2,OUTPUT);
    pinMode(yellowY2,OUTPUT);
    pinMode(greenG2,OUTPUT);

    digitalWrite(redR1,LOW);
    digitalWrite(greenG1,LOW);
    digitalWrite(yellowY1,LOW);
    digitalWrite(redR2,LOW);
}
```



```
digitalWrite(greenG2,LOW);
digitalWrite(yellowY2,LOW);
}

void loop()
{
    // Xanh 1 và Đỏ 2 sáng
    digitalWrite(greenG1,HIGH);
    digitalWrite(redR2,HIGH);

    digitalWrite(yellowY1,LOW);
    digitalWrite(redR1,LOW);
    digitalWrite(greenG2,LOW);
    digitalWrite(yellowY2,LOW);
    delay(20000);

    //Vàng 1 và Đỏ 2 sáng
    digitalWrite(yellowY1,HIGH);
    digitalWrite(redR2,HIGH);

    digitalWrite(greenG1,LOW);
    digitalWrite(redR1,LOW);
    digitalWrite(greenG2,LOW);
    digitalWrite(yellowY2,LOW);
    delay(2000);
    // Đỏ 1 và Xanh 2 sáng

    digitalWrite(redR1,HIGH);
    digitalWrite(greenG2,HIGH);

    digitalWrite(greenG1,LOW);
    digitalWrite(yellowY1,LOW);
    digitalWrite(yellowY2,LOW);
    digitalWrite(redR2,LOW);
    delay(20000);

    // Đỏ 1 và vàng 2 sáng
    digitalWrite(redR1,HIGH);
    digitalWrite(yellowY2,HIGH);

    digitalWrite(greenG1,LOW);
    digitalWrite(yellowY1,LOW);
```

```
digitalWrite(greenG2,LOW);
digitalWrite(redR2,LOW);
delay(2000);
```

```
}
```

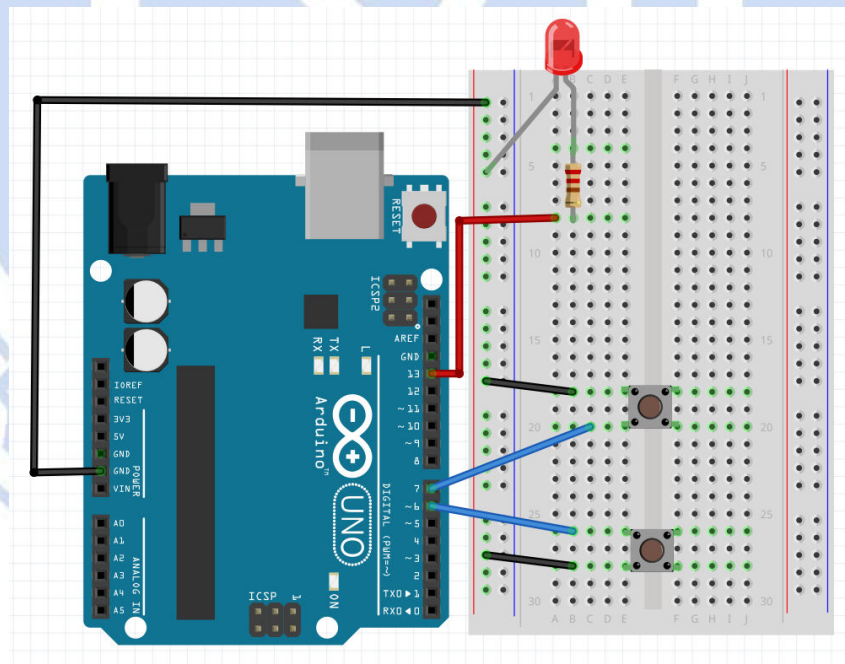
```
// -----
```

- ❖ **Bài tập:** Viết chương trình điều khiển đèn giao thông ngã 4 với yêu cầu sau:
 - Hướng đi 1: xanh sáng 30s; vàng 3s; đỏ 43s.
 - Hướng đi 2: xanh sáng 40s; vàng 3s; đỏ 33s.

2.3. Viết chương trình điều khiển Led bằng nút nhấn

- ❖ **Yêu cầu:** viết chương trình nhấn nút ON: Led sáng; Nhấn nút OFF: Led tắt.
- ❖ **Chuẩn bị linh kiện:**
 - 2 nút nhấn.
 - Dây cắm.
 - Testboard
 - Một Led (màu tùy ý).

- ❖ **Kết nối mạch như Hình 2.3 như sau:**



Hình 2.3: Mạch điều khiển Led bằng nút nhấn

- ❖ **Chương trình mẫu:**

```
// -----
```




```
const int ON = 6;
const int OFF = 7;
const int LED = 13;
int buttonState_ON=1 ;
int buttonState_OFF = 1;
void setup()
{
    pinMode(ON, INPUT_PULLUP);
    pinMode(OFF, INPUT_PULLUP);
    pinMode(LED, OUTPUT);
    digitalWrite(LED, LOW);
}

void loop()
{
    buttonState_ON = digitalRead(ON);
    buttonState_OFF = digitalRead(OFF);
    if (buttonState_ON == LOW)
    {
        digitalWrite(LED, HIGH);
    }
    if (buttonState_OFF == LOW)
    {
        digitalWrite(LED, LOW);
    }
}
// -----
```

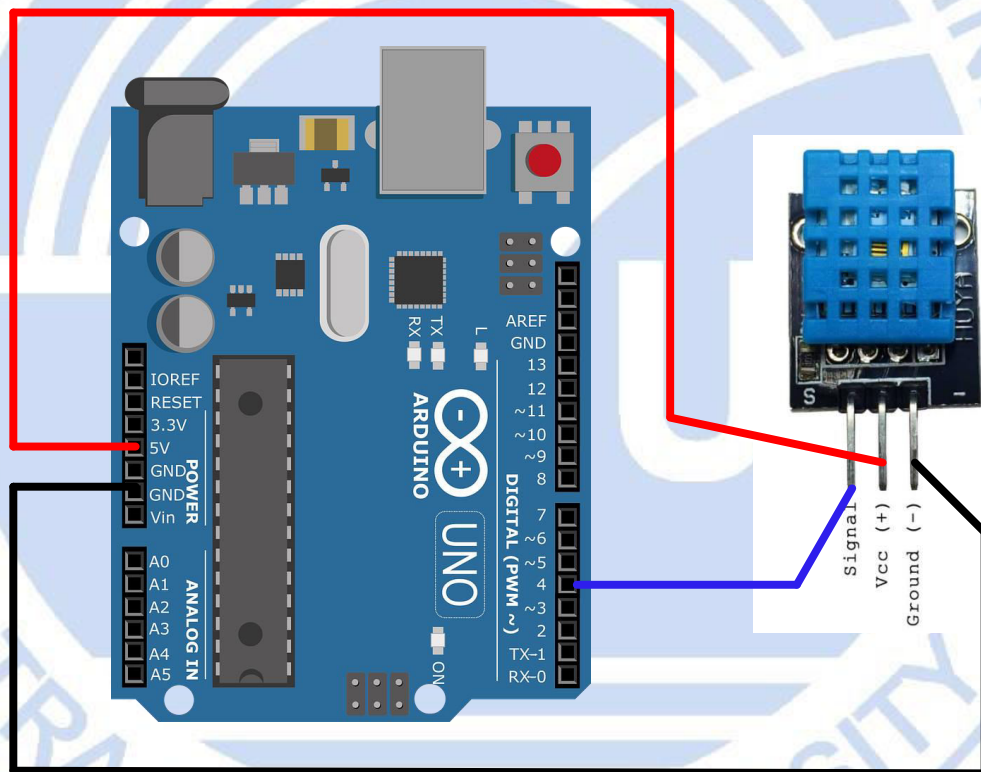
- ❖ **Bài tập:** viết chương trình điều khiển LED ON / OFF chỉ với một nút nhấn. Cụ thể là, nhấn lần 1 Led sáng, nhấn lần thứ 2 Led sẽ tắt...

BÀI 3:

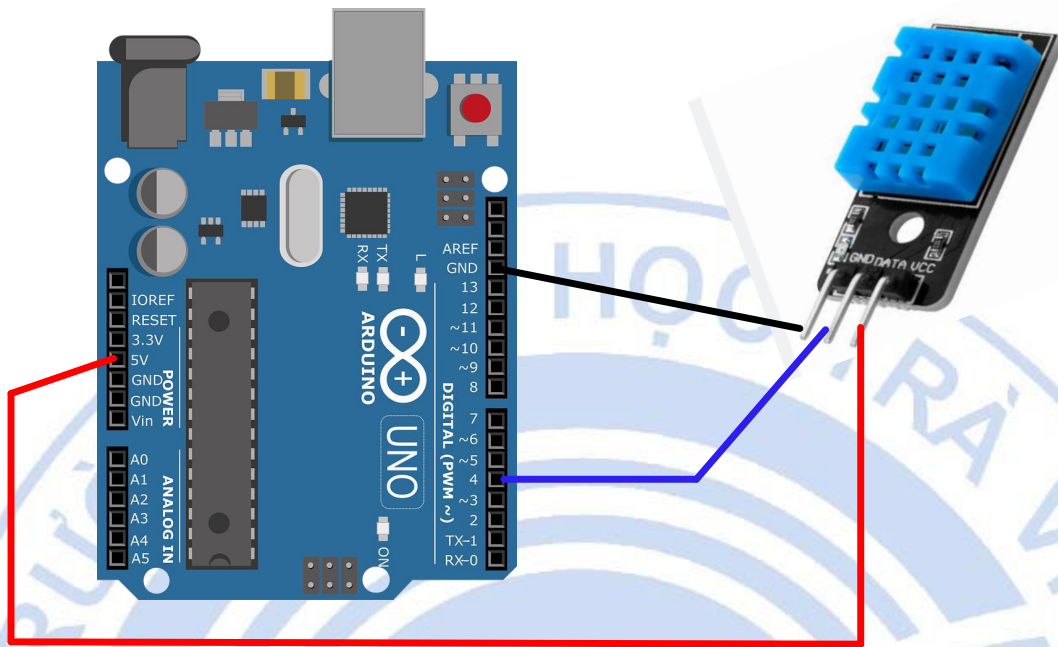
LẬP TRÌNH ĐIỀU KHIỂN VỚI CẢM BIẾN NHIỆT ĐỘ - ĐỘ ẨM

3.1. Đo nhiệt độ - Độ ẩm hiển thị lên máy vi tính.

- ❖ **Yêu cầu:** Viết chương trình đo nhiệt độ và độ ẩm trong phòng, hiển thị kết quả đo lên màn hình Serial Monitor.
- ❖ **Chuẩn bị linh kiện:**
 - 1 cảm biến DHT11 (lưu ý có 2 dạng module DHT11)
 - Dây cắm.
- ❖ **Kết nối mạch như Hình 3.1a và Hình 3.1 b như sau:**



Hình 3.1a: sơ đồ kết nối DHT11 dạng 1.



Hình 3.1b: sơ đồ kết nối DHT11 dạng 2.

❖ **Chương trình mẫu:**

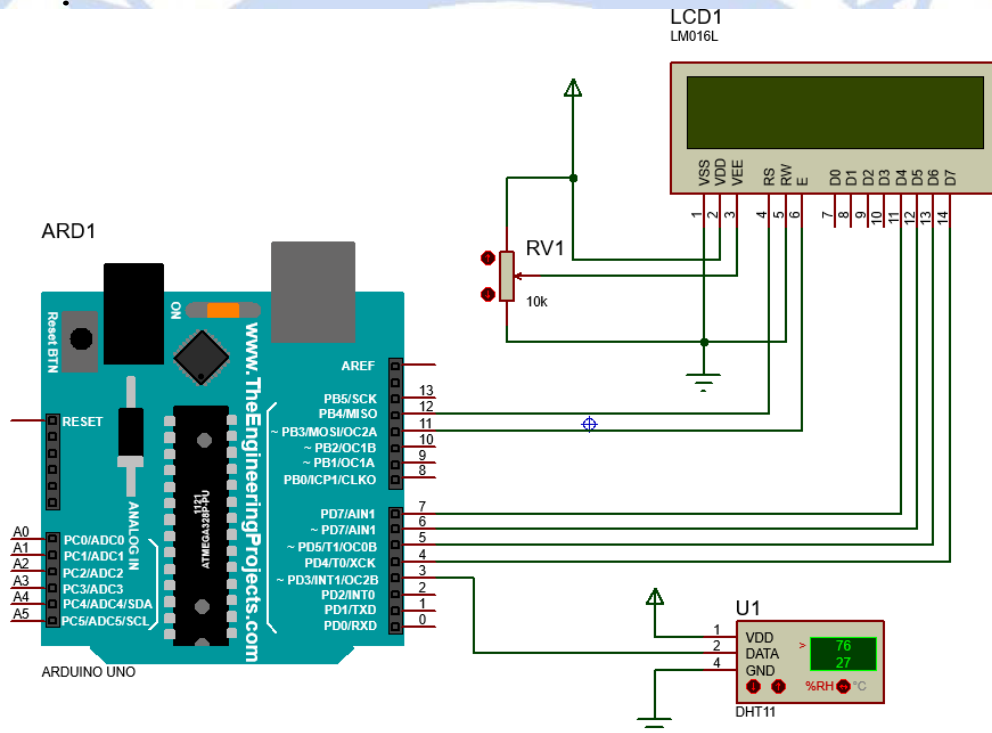
```
//-----
#include "DHT.h"
int DHTPIN = 4;
const int DHTTYPE = DHT11;
DHT dht(DHTPIN, DHTTYPE);
void setup()
{
    Serial.begin(9600);
    dht.begin();
}
void loop() {
    float h = dht.readHumidity();
    float t = dht.readTemperature();
    Serial.print("Nhiệt độ: ");
    Serial.println(t);
    Serial.print("Độ ẩm: ");
    Serial.println(h);
    Serial.println();
    delay(1000);
}
//-----
```

❖ **Bài tập:** viết chương trình đo nhiệt độ cảnh báo đèn và chuông như sau:

- Nếu nhiệt độ nhỏ hơn hoặc bằng 28°C thì đèn không sáng, chuông không kêu.
- Nếu nhiệt độ lớn hơn 28°C thì đèn sáng, chuông kêu.

3.2. Đo nhiệt độ - Độ ẩm hiển thị màn hình LCD 16*2

- ❖ **Yêu cầu:** Viết chương trình đo nhiệt độ và độ ẩm trong phòng, hiển thị kết quả đo lên màn hình LCD.
- ❖ **Chuẩn bị linh kiện:**
 - 1 cảm biến DHT11 (lưu ý có 2 dạng module DHT11)
 - Màn hình LCD 16*2.
 - 1 biến trở 10KΩ
 - Dây cắm.
- ❖ **Kết nối mạch như Hình 3.2 như sau:**



Hình 3.2: Sơ đồ kết nối LCD

- ❖ **Chương trình mẫu:**

```
// -----
#include <LiquidCrystal.h>
#include "DHT.h"
#define DHTPIN 3
LiquidCrystal lcd(12, 11, 7, 6, 5, 4);
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);

void setup()
{
    lcd.begin(16, 2);
    dht.begin();
    lcd.print("Temp: Humidity:");
}
```

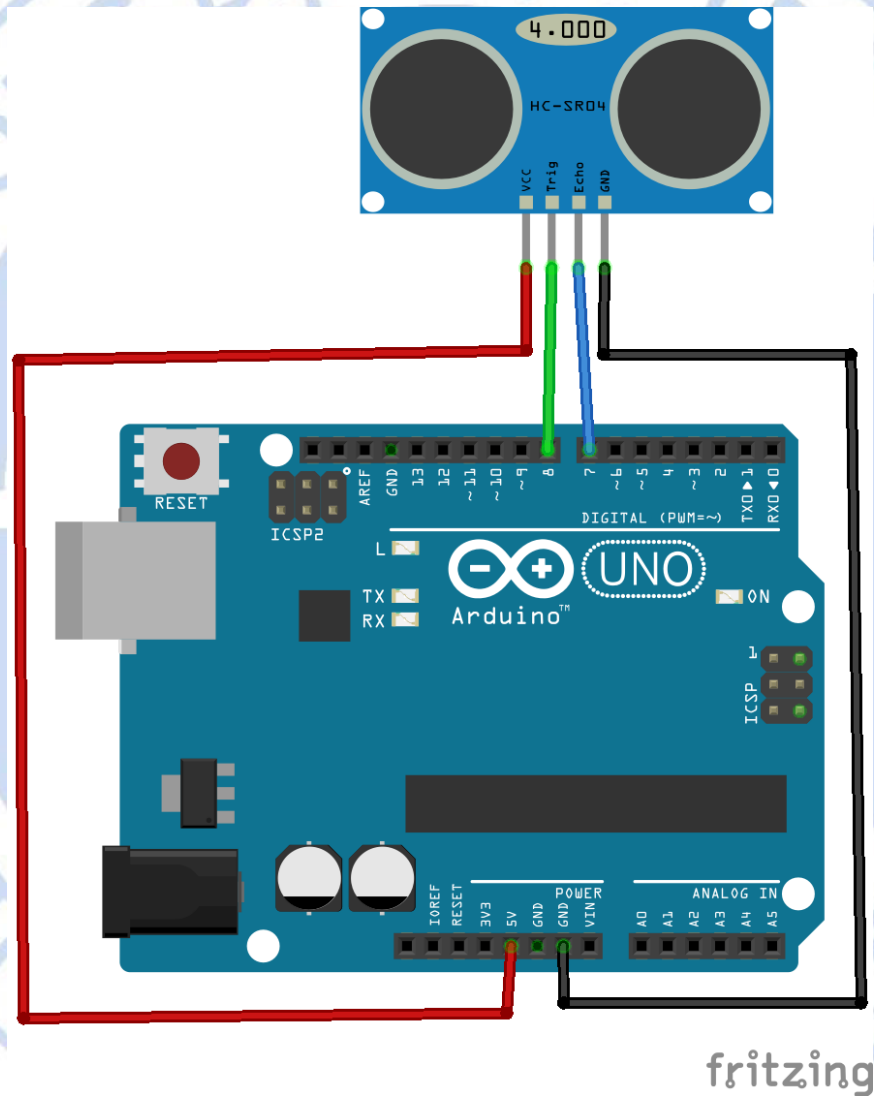


```
}  
void loop()  
{  
    lcd.setCursor(0, 1);  
    float h = dht.readHumidity();  
    float f = dht.readTemperature();  
    if (isnan(h) || isnan(f))  
    {  
        lcd.print("ERROR");  
        return;  
    }  
    lcd.print(f);  
    lcd.setCursor(7,1);  
    lcd.print(h);  
}  
// -----
```

- ❖ **Bài tập:** viết chương trình đo nhiệt độ - độ ẩm hiển thị LCD với kết quả chạy từ phải qua trái.

BÀI 4: LẬP TRÌNH ĐIỀU KHIỂN VỚI CẢM BIẾN SIÊU ÂM

- ❖ **Yêu cầu:** Viết chương trình đo khoảng cách và hiển thị kết quả đo lên máy tính.
- ❖ **Chuẩn bị linh kiện:**
 - 1 cảm biến siêu âm HC-SR04.
 - Dây cắm.
- ❖ **Kết nối mạch như Hình 4.1 như sau:**



Hình 4.1: Kết nối cảm biến siêu âm HC-SR04

- ❖ **Chương trình mẫu:**

```
// -----  
const int trig = 8;    // chân trig của HC-SR04 kết nối với chân 8 của Arduino  
const int echo = 7;    // chân echo của HC-SR04 kết nối với chân 7 của Arduino  
int distance = 0;
```




```
void setup()
{
    Serial.begin(9600);
    pinMode(trig, OUTPUT);
    pinMode(echo, INPUT);
}

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

    digitalWrite(trig,0);
    delayMicroseconds(2);
    digitalWrite(trig,1);
    delayMicroseconds(5);
    digitalWrite(trig,0);

    /* Tính toán thời gian */
    duration = pulseIn(echo, HIGH);
    distance = int(duration/2/29.412);

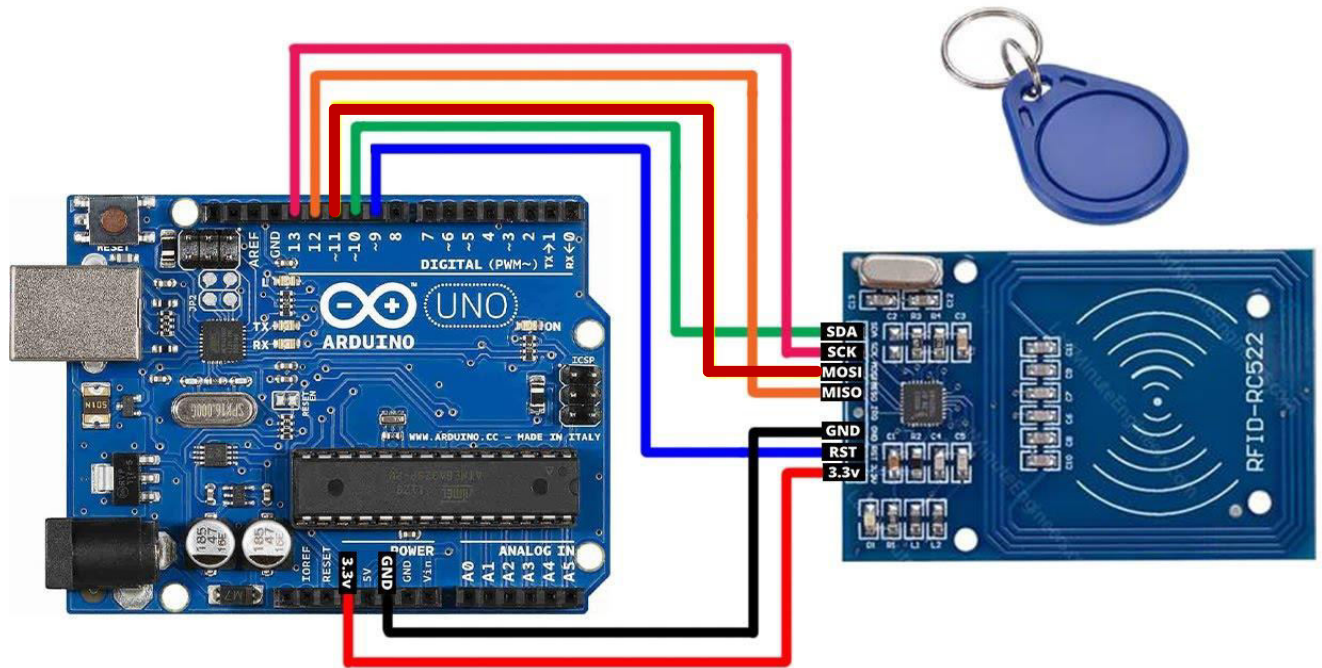
    /* In kết quả ra Serial Monitor */
    Serial.print(distance);
    Serial.println("cm");
    delay(1000);

    //-----
```

- ❖ **Bài tập:** Viết chương trình đo khoảng cách vật cản, nếu khoảng cách nhỏ hơn 20cm sẽ phát tín hiệu cảnh báo bằng đèn led nhấp nháy liên tục.

BÀI 5: LẬP TRÌNH ĐIỀU KHIỂN VỚI RFID

- ❖ **Yêu cầu:** Viết chương trình đọc mã thẻ RFID **13.56MHz** và hiện mã thẻ lên máy tính.
- ❖ **Chuẩn bị linh kiện:**
 - 1 mạch RFID-RC522
 - 2 thẻ RFID
 - Dây cắm.
- ❖ **Kết nối mạch như Hình 5.1 như sau:**



Hình 5.1: Kết nối RFID-RC522

- ❖ **Chương trình mẫu:**
- ```
// -----
#include <SPI.h>
#include <MFRC522.h> // thư viện "RFID"
#define SS_PIN 10
#define RST_PIN 9
MFRC522 mfrc522(SS_PIN, RST_PIN);
unsigned long uidDec, uidDecTemp; // hiển thị số UID dạng thập phân
byte bCounter, readBit;
unsigned long ticketNumber;

void setup()
{

```



```
Serial.begin(9600);
SPI.begin();
mfrc522.PCD_Init();
Serial.println("Cong Nghe Ky Thuat Dieu Khien & Tu Dong Hoa ");
Serial.println("Waiting for card...");
}

void loop()
{
 // Tim the moi
 if (!mfrc522.PICC_IsNewCardPresent())
 {
 return;
 }
 // Doc the
 if (!mfrc522.PICC_ReadCardSerial())
 {
 return;
 }
 uidDec = 0;
 Serial.println("=====");
 // Hien thi so UID cua the
 Serial.println("Serijnyj nomer karty / Card UID: ");
 for (byte i = 0; i < mfrc522.uid.size; i++)
 {
 uidDecTemp = mfrc522.uid.uidByte[i];
 uidDec = uidDec*256+uidDecTemp;
 }

 Serial.print(uidDec);
}

// -----
```

❖ **Bài tập:** Viết chương trình điều khiển Led Sáng / Tắt bằng thẻ RFID.