

LẬP TRÌNH CƠ BẢN VI ĐIỀU KHIỂN

Hướng dẫn tải và cài đặt phần mềm Arduino IDE

Bước 1: Truy cập vào địa chỉ: <https://www.arduino.cc/en/software>

Bước 2: Chọn 1 trong tất cả các phiên bản đã cho.

Downloads



Arduino IDE 2.2.1

The new major release of the Arduino IDE is faster and even more powerful! In addition to a more modern editor and a more responsive interface it features autocompletion, code navigation, and even a live debugger.

For more details, please refer to the [Arduino IDE 2.0 documentation](#).

Nightly builds with the latest bugfixes are available through the section below.

SOURCE CODE

The Arduino IDE 2.0 is open source and its source code is hosted on [GitHub](#).

DOWNLOAD OPTIONS

Windows Win 10 and newer, 64 bits
Windows MSI installer
Windows ZIP file

Linux AppImage 64 bits (X86-64)
Linux ZIP file 64 bits (X86-64)

macOS Intel, 10.14: "Mojave" or newer, 64 bits
macOS Apple Silicon, 11: "Big Sur" or newer, 64 bits

[Release Notes](#)

Bước 3: Chọn “Just download” để tải về.

Download Arduino IDE & support its progress

Since the 1.x release in March 2015, the Arduino IDE has been downloaded **77,969,312** times — impressive! Help its development with a donation.

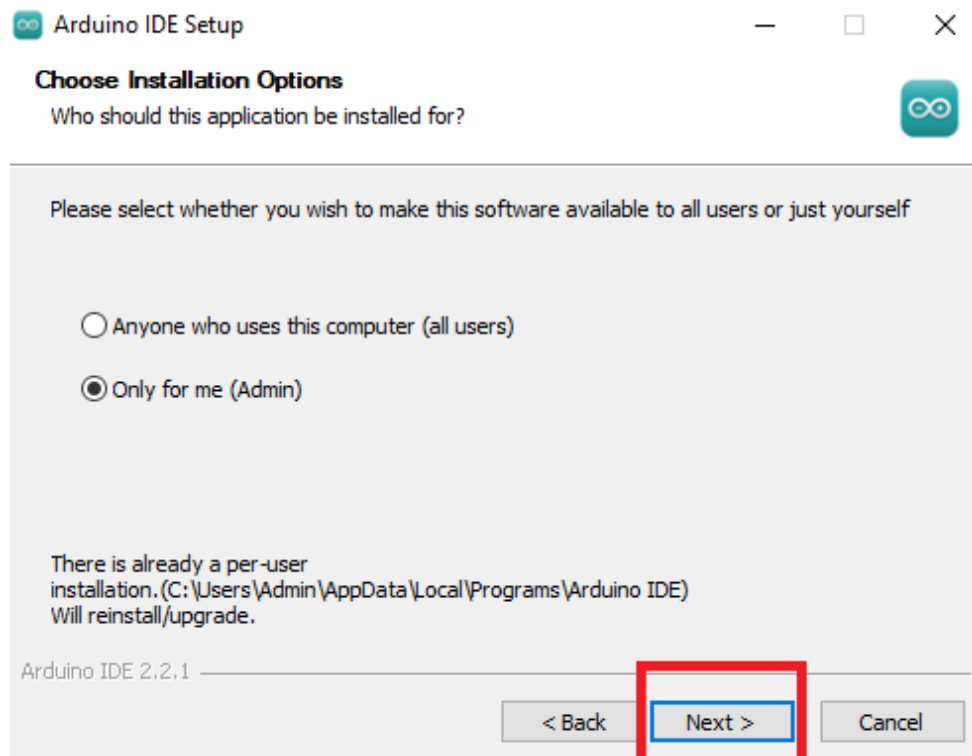
\$3	\$5	\$10	\$25	\$50	Other
-----	-----	------	------	------	-------

CONTRIBUTE AND DOWNLOAD

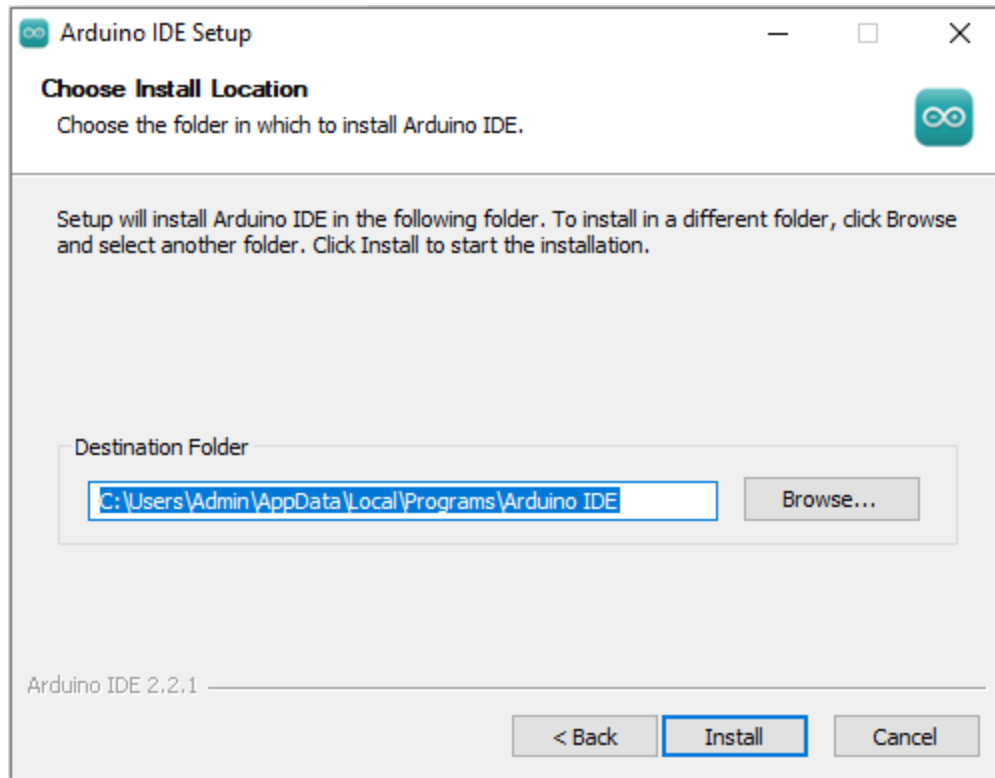
or

JUST DOWNLOAD

Bước 4: Sau khi tải về, cho chạy file mới tải



Bước 5: Cho nơi cài đặt chương trình Arduino IDE xong ấn “Install”.

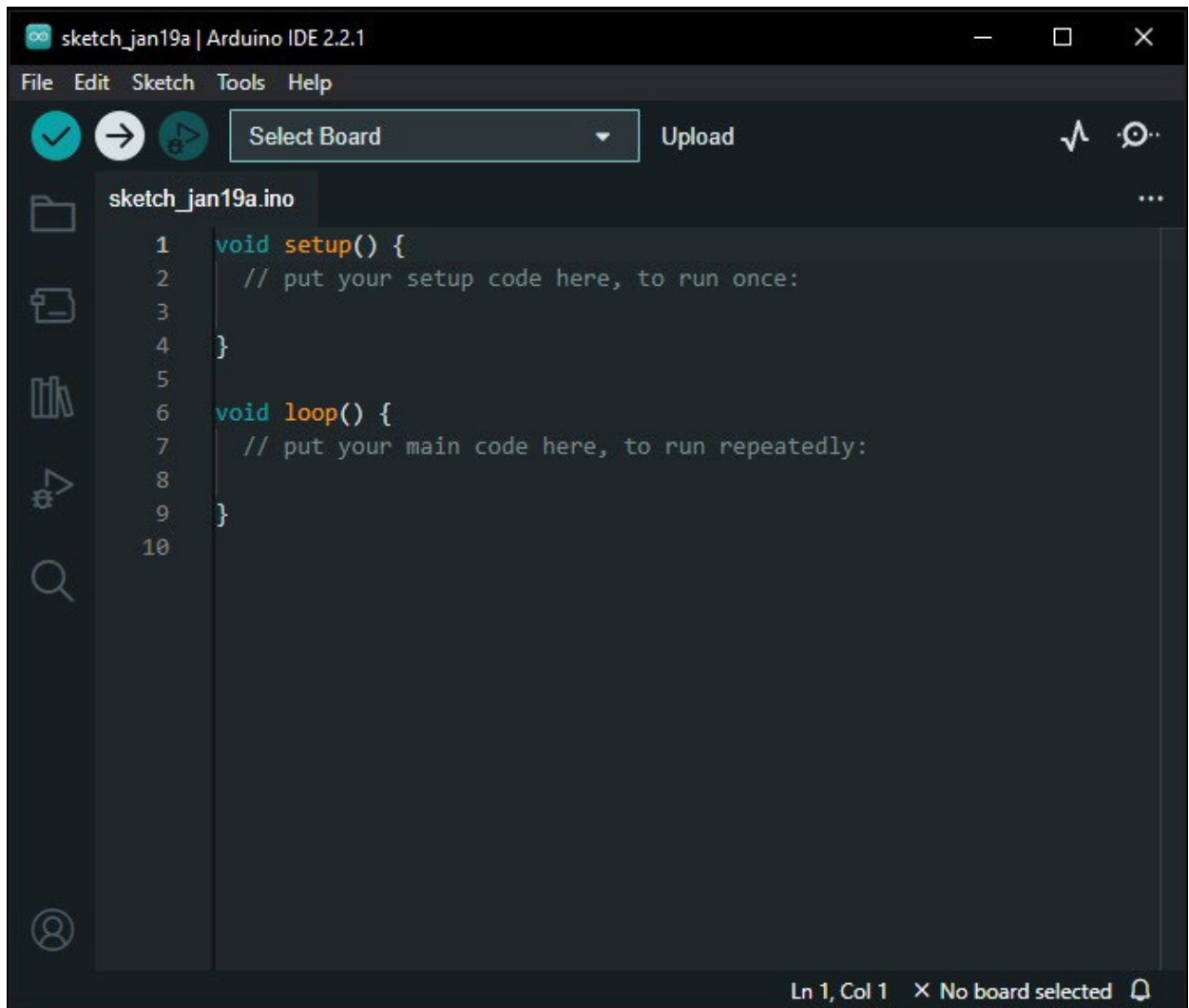


Bước 6: Sau khi đã cài đặt xong phần mềm sẽ yêu cầu Cài một số Driver để nạp chương trình cho vi điều khiển, bạn chỉ cần ấn Install tất cả Driver phần mềm đề nghị.









I.2: Viết và nạp một chương trình cho Arduino Uno

Giao diện Arduino IDE khi mới khởi động



Các chức năng nút lệnh trên Arduino:

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
	Gỡ lỗi chương trình đang soạn thảo
	Board đang được chọn để nạp chương trình vào
	Hiển thị đồ thị các đầu ra giữa máy tính và Board Arduino
	Hiển thị các dữ liệu được gửi và nhận dữ liệu giữa máy tính và board Arduino

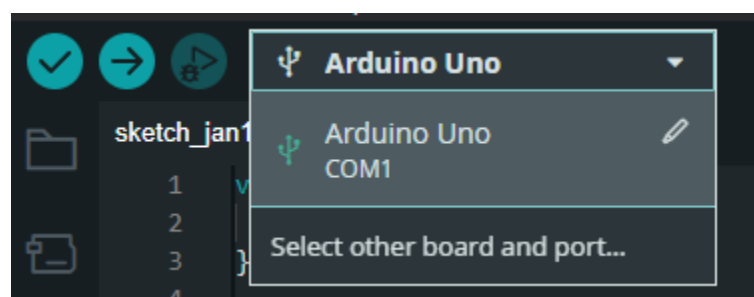
Soạn một chương trình đơn giản: Hiển thị chữ “Hello World” ra Serial Monitor:

```

1 void setup() {
2   Serial.begin(9600);
3 }
4
5 void loop() {
6   Serial.println("Hello World");
7   delay(1000); // Đợi 1000ms = 1s để in ra tiếp "Hello World"
8 }

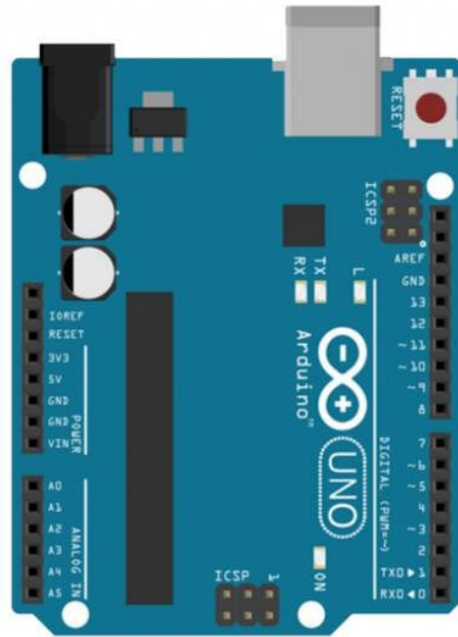
```

Chọn Cổng COM và Board đang sử dụng (Arduino UNO):



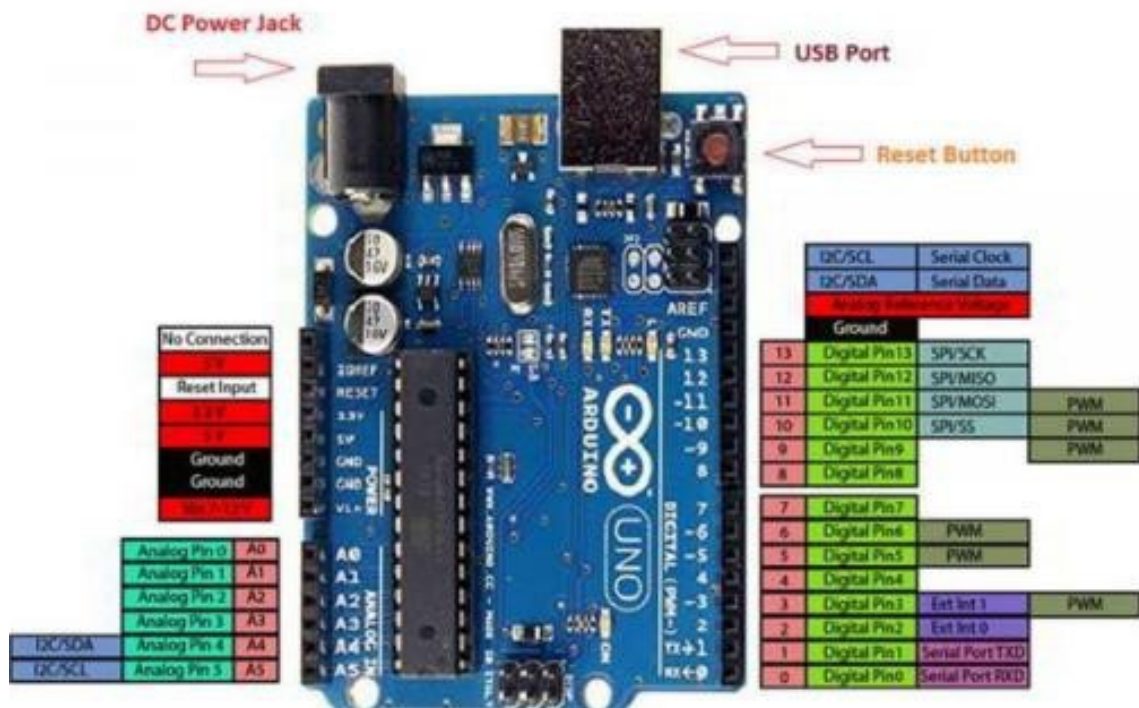
Nhấn Upload để nạp, màn hình báo Done uploading là hoàn thành:

I.3: Giới thiệu cơ bản về Arduino Uno



Arduino Uno là một vi điều khiển dựa trên ATmega328P. Nó có 14 chân đầu vào/đầu ra kỹ thuật số (trong đó 6 chân có thể được sử dụng làm đầu ra PWM), 6 đầu vào tương tự, tinh thể thạch anh 16 MHz, kết nối USB, giắc cắm nguồn, header ICSP và nút reset. Mạch chứa mọi thứ cần thiết để hỗ trợ vi điều khiển, chỉ cần kết nối với máy tính bằng cáp USB hoặc cấp nguồn bằng bộ chuyển đổi AC-to-DC hoặc qua chân nguồn để bắt đầu.

- Sơ đồ chân Arduino Uno:



Chức năng các chân:

- VIN: dùng để cung cấp nguồn khi không dùng USB hoặc các nguồn khác, điện áp 7-12VDC
- 5V: Chân này xuất ra nguồn 5V
- 3V3: Nguồn cung cấp 3.3V (dòng điện trên chân này tối đa là 50mA).
- GND: Là chân mang điện cực âm trên board.
- IOREF: chân này dùng để cung cấp tham chiếu điện áp mà bộ vi điều khiển hoạt động.
- Reset: dùng để kết nối với nút reset ra bên ngoài.

Trên Board Arduino Uno có 14 chân Digital được sử dụng để làm chân đầu vào và đầu ra, chúng sử dụng các hàm pinMode(), digitalWrite(), digitalRead(). Điện áp trên mỗi chân là 5V, dòng trên mỗi chân là 20mA và bên trong có điện trở kéo lên là 20-50 ohm. Dòng tối đa trên mỗi chân I/O không vượt quá 40mA để tránh trường hợp gây hỏng board mạch. Uno có 6 chân đầu vào Analog từ A0 đến A5, mỗi chân cung cấp 10 bit độ phân giải (tức là 1024 giá trị khác nhau).

- Serial: chân 0 (RX) và 1 (TX): Được sử dụng để nhận dữ liệu (RX) và truyền dữ liệu (TX) TTL.
- Ngắt ngoài: chân 2 và 3

- PWM (điều chế độ rộng xung): chân 3, 5, 6, 9 và 11 cung cấp đầu ra xung PWM với độ phân giải 8 bit bằng hàm `analogWrite()`.
- SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Các chân này hỗ trợ giao tiếp SPI bằng thư viện SPI.
- TWI/I2C: A4 (SDA) và A5 (SCL) hỗ trợ giao tiếp I2C/TWI với các thiết bị khác.
- AREF: điện áp tham chiếu cho các đầu vào tương tự

1. Lập trình cho vi điều khiển hiển thị dòng chữ lên màn hình LCD

1.1. Màn hình LCD 16x2:

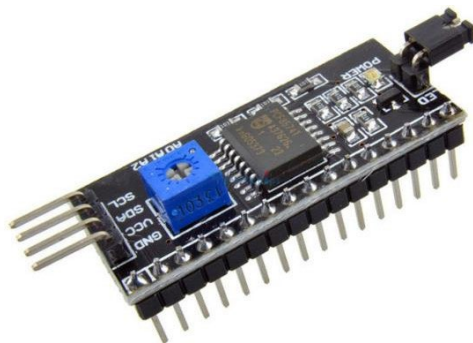


- LCD 16×2 có 16 chân trong đó 8 chân dữ liệu (D0 – D7) và 3 chân điều khiển (RS, RW, EN).
- chân còn lại dùng để cấp nguồn và đèn nền cho LCD 16×2.
- Các chân điều khiển giúp ta dễ dàng cấu hình LCD ở chế độ lệnh hoặc chế độ dữ liệu.
- Chúng còn giúp ta cấu hình ở chế độ đọc hoặc ghi.

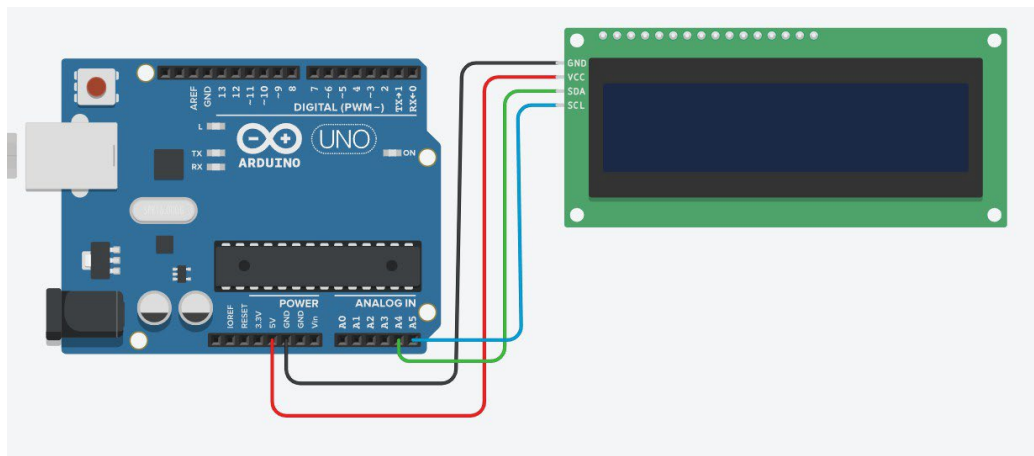
Thông số kĩ thuật:

- Điện áp hoạt động là 5 V
- Địa chỉ I2C: 0x27 (có thể thay đổi theo đơn hàng của nsx ví dụ như: 0x3F)
- Hỗ trợ màn hình: LCD1602,1604,2004 (driver HD44780).
- Có đèn led nền, có thể dùng biến trở hoặc PWM điều chỉnh độ sáng để sử dụng ít điện năng hơn.
- Có thể được điều khiển với 6 dây tín hiệu

1.2. Module I2C Arduino:



- LCD có quá nhiều nhiều chân gây khó khăn trong quá trình đấu nối và chiếm dụng nhiều chân trên vi điều khiển. Do đó Module I2C LCD ra đời để giải quyết vấn đề này. Thay vì phải mất 6 chân vi điều khiển để kết nối với LCD 16×2 (RS, EN, D7, D6, D5 và D4) thì module IC2 bạn chỉ cần tốn 2 chân (SCL, SDA) để kết nối.



1.3. Sơ đồ đấu nối:

Arduino	LCD
GND	GND
VCC	VCC
SDA/A4	SDA
SCL/A5	SCL

1.4. Code tham khảo:

- Thư viện cho LCD: “LiquidCrystal I2C” của Frank de Brabander
Link tham khảo có code mẫu:
https://github.com/johnrickman/LiquidCrystal_I2C
- Có thể mô phỏng qua đường link đây: <https://wokwi.com>
- Code:

```
#include <LiquidCrystal_I2C.h>
```

```
LiquidCrystal_I2C lcd(0x27,16,2); // Khai báo địa chỉ của LCD gồm 16 cột,  
2 dòng
```

```
void setup()
```

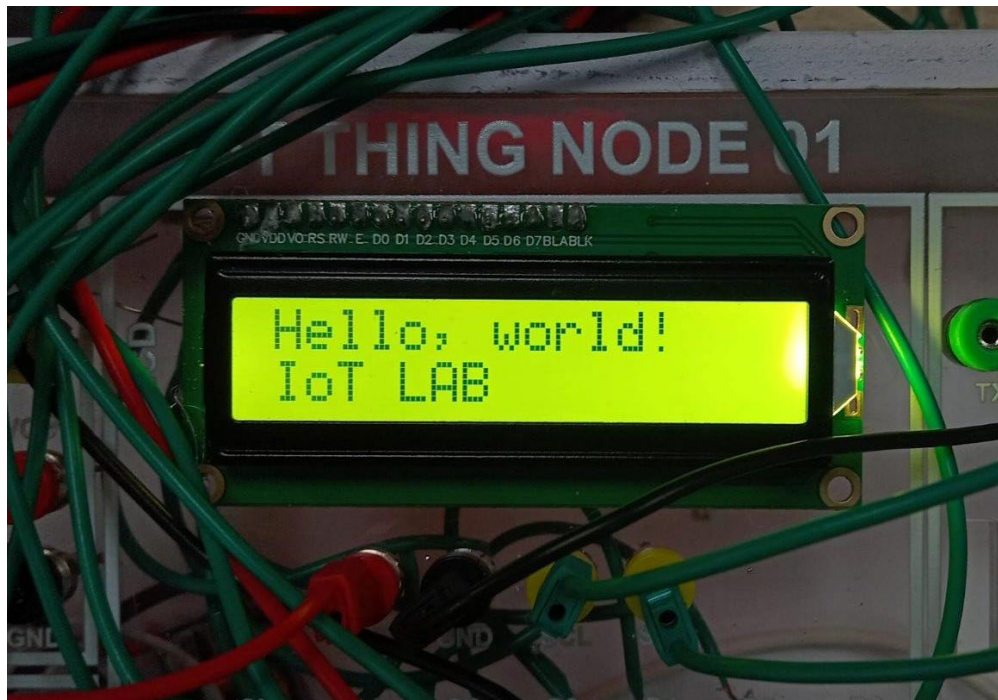
```
{  
  lcd.init();// Khởi động màn hình LCD  
  lcd.backlight();// Bật đèn nền LCD  
  lcd.setCursor(0,0); //Đưa con trỏ vị trí tới hàng 1 cột 1, nếu là (1,0)  
thì là hàng 1 cột thứ 2  
  lcd.print("Hello, world!");// In ra màn hình LCD  
  lcd.setCursor(0,1); // Đưa con trỏ tới vị trí hàng 1 cột 2  
  lcd.print("IoT LAB");// In ra màn hình LCD  
}
```

```
void loop()
```

```
{
```

```
}
```

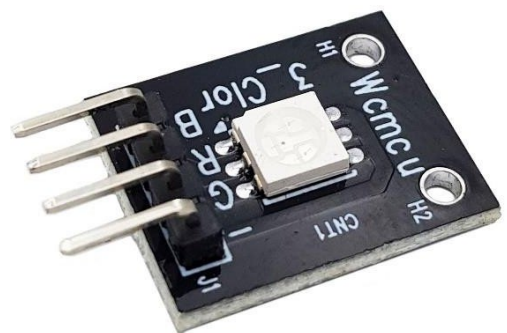
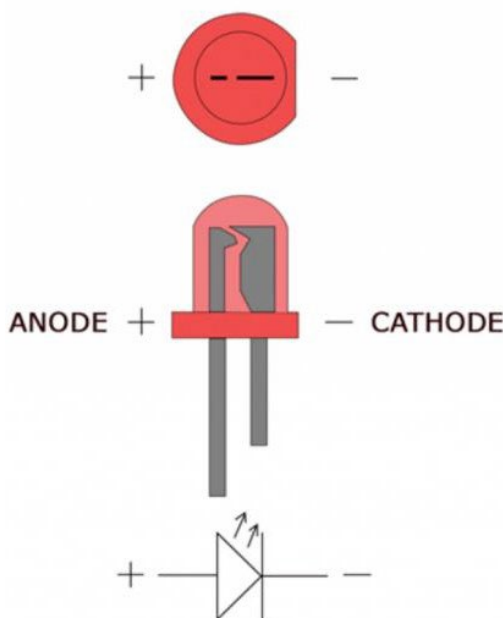
1.5. Kết quả:



2. Lập trình cho vi điều khiển Led RGB

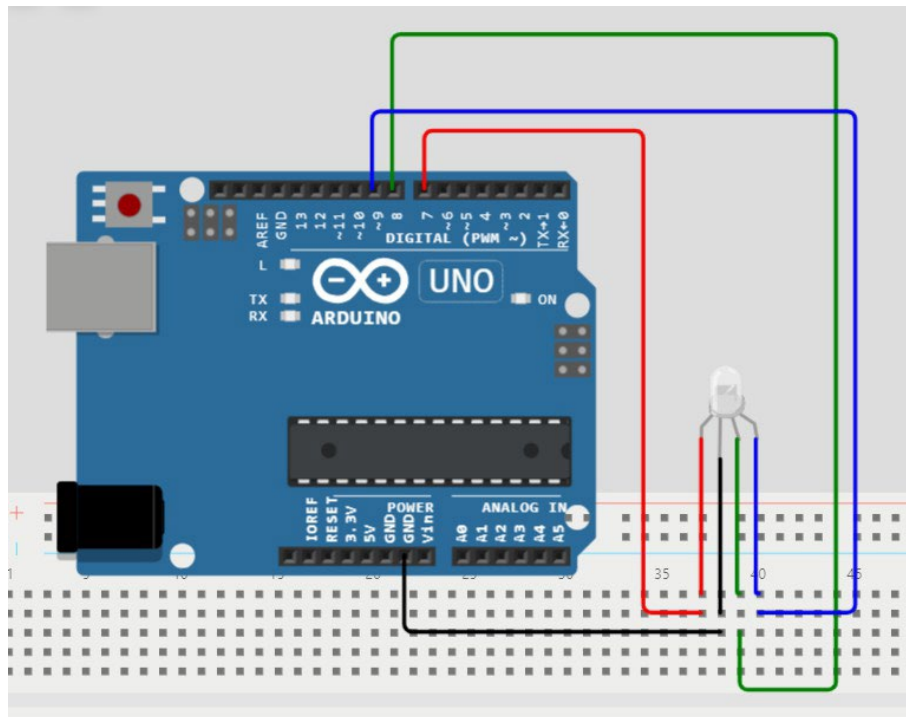
2.1. Led RGB

- Đèn led hay còn gọi là điốt phát quang, là một biến thể của diode cơ bản, nó xác định độ chênh lệch điện áp nhỏ nhất giữa Anode (+) và Cathode (-), led khác diode ở chỗ là tạo ra ánh sáng khi dòng điện đi qua. Đèn led cần được kết nối đúng âm dương của nguồn điện để hoạt động. Led RGB có 3 màu: đỏ, xanh lá, xanh dương.



- Điện áp: Mỗi đèn led đều có một mức điện áp đầu vào nhất định (led 5mm dao động 1.9 – 3.2 Vol), nếu cung cấp không đủ đèn sẽ không phát sáng hoặc cung cấp nguồn lớn hơn thì cần phải sử dụng trở để giảm điện thế vào led. Nếu bạn mắc nhiều đèn led nối tiếp, song song thì cần phải tính toán mức điện áp, dòng điện cung cấp cho tất cả các led.
- Dòng điện: Nếu bạn cung cấp dòng điện vượt qua ngưỡng cho phép, tương đương với việc làm gia tăng nhiệt độ và làm giảm tuổi thọ của led. Dòng điện phù hợp với led 5mm thường ở mức 20mA, tối đa có thể là 30mA, chúng ta kiểm soát dòng điện bằng cách mắc một điện trở nối tiếp với đèn led, nó giúp dòng điện của led luôn ở mức cho phép.

2.2. Sơ đồ đấu nối:



Arduino	LED
D7	R
D8	G
D9	B
GND	GND

2.2. Code tham khảo:

```
#define red 7
#define green 8
#define blue 9

void setup() {
  pinMode(red, OUTPUT); // Cấu hình chân D7 làm OUTPUT cho led red
  pinMode(green, OUTPUT); // Cấu hình chân D8 làm OUTPUT cho led green
  pinMode(blue, OUTPUT); // Cấu hình chân D9 làm OUTPUT cho led blue
```

```

}

void loop() {
    digitalWrite(red,HIGH); // Chân red lên mức Cao hay Bật Đèn Led
    delay(1000); // Đèn sáng tầm 1 giây
    digitalWrite(red,LOW); // Chân red xuống mức Thấp hay Tắt Đèn Led
    digitalWrite(green,HIGH); // Chân green lên mức Cao hay Bật Đèn Led
    delay(1000); // Đèn sáng tầm 1 giây
    digitalWrite(green,LOW); // Chân green xuống mức Thấp hay Tắt Đèn Led
    digitalWrite(blue,HIGH); // Chân blue lên mức Cao hay Bật Đèn Led
    delay(1000); // Đèn sáng tầm 1 giây
    digitalWrite(blue,LOW); // Chân blue xuống mức Thấp hay Tắt Đèn Led
}

```

2.3. Kết quả:



3. Lập trình cho vi điều khiển giao tiếp bàn phím

3.1. Bàn phím số (Keypad):

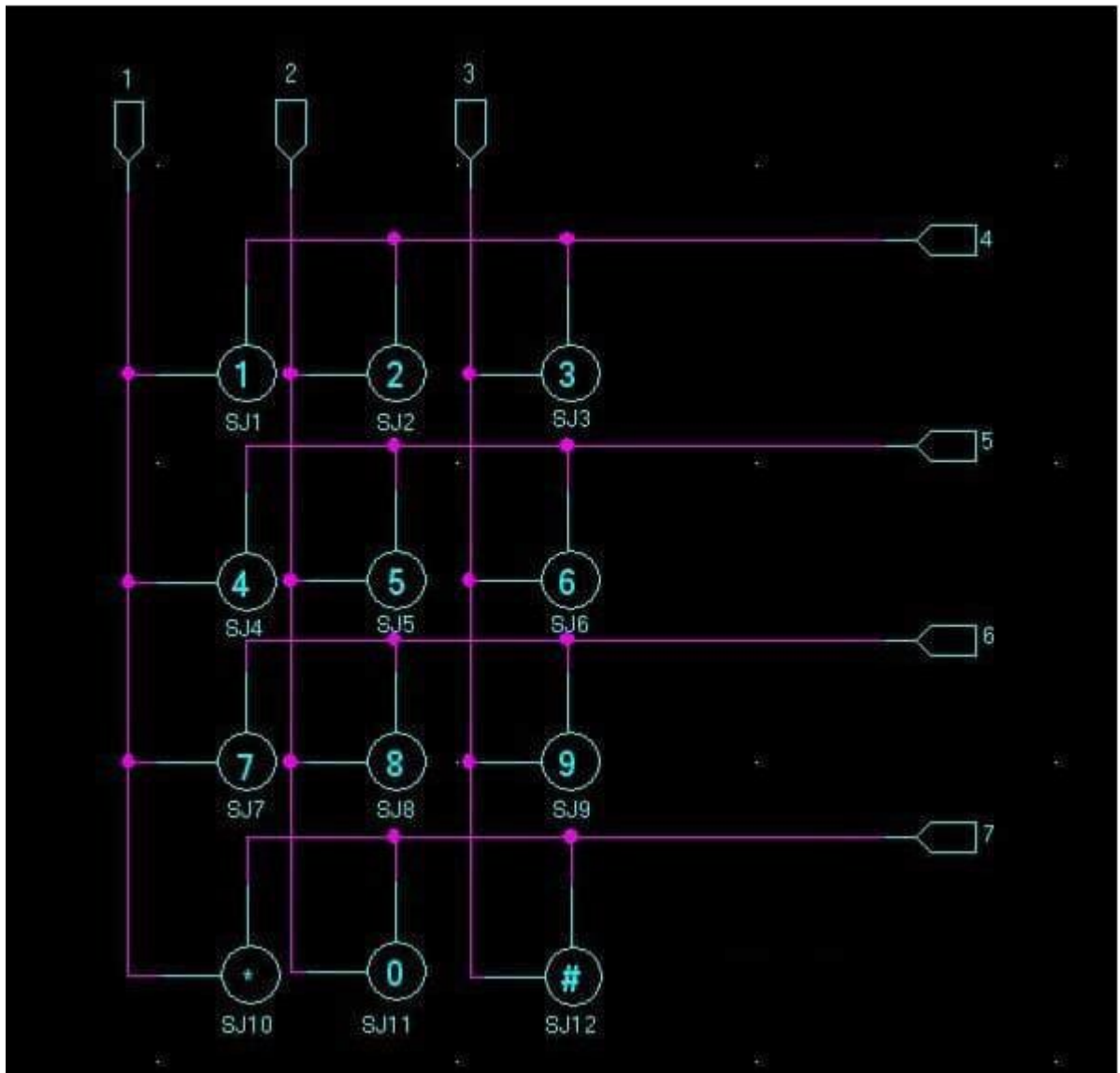


Bàn phím ma trận 3x4 nhựa cứng gồm có 16 nút bấm được sắp xếp theo ma trận 3 hàng, 4 cột. Các nút bấm trong cùng một hàng và một cột được nối với nhau.

3.2. Thông số kỹ thuật:

- Kích thước : 7×5.2×0.9cm
- Bàn Phím Ma Trận Nhựa Cứng 3x4 Keypad loại phím nhựa cứng.
- Độ dài cáp: 88mm.
- Nhiệt độ hoạt động 0 ~ 70°C.
- Đầu nối ra 7 chân.
- Kích thước bàn phím 65 x 64mm

3.3. Sơ đồ chân:



- Sơ đồ đấu nối:

Arduino	KeyPad
D3	(3,2)
D4	(2,3)
D5	(2,2)
D6	(2,1)
D7	(1,3)
D8	(1,2)
D9	(1,1)

**Note: (3,2) trong bảng chân của KeyPad ở đây tương đương với hàng thứ 3 và cột thứ 2 trong bộ Kit*

3.4. Lập trình:

```
#include <LiquidCrystal_I2C.h>

#include <Keypad.h>

#include <stdio.h>

const byte ROWS = 4; //4 hàng
const byte COLS = 3; //3 cột

String arr[8] = {};
String password[8] = {"1", "2", "3", "4", "5", "6", "7", "8"};

int place = 0;
int input = 6;

bool value = true;

char keys[ROWS][COLS] = {
    {'1','2','3'},
    {'4','5','6'},
    {'7','8','9'},
    {'*','0','#'}
};

LiquidCrystal_I2C lcd(0x27,16,2);    // Khai báo địa chỉ của LCD gồm 16
cột, 2 dòng

byte rowPins[ROWS] = {8, 3, 4, 6}; //nối hàng của keypad
byte colPins[COLS] = {7, 9, 5};    //nối cột keypad

Keypad keypad = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS );

void setup(){
    lcd.init();// Khởi động màn hình LCD
    lcd.backlight();// Bật đèn nền LCD
    lcd.setCursor(0,0); //Đưa con trỏ vị trí tới hàng 1 cột 1, nếu là (1,0)
    thì
    lcd.print("pass: ");
    lcd.setCursor(0,1);
    lcd.print("status: ");
```

```
    Serial.begin(9600); //khởi tạo serial
}
```

```
void loop(){
    //Kiểm tra nút được nhấn
    char key = keypad.getKey();

    //in nút ra màn hình
    if (key != NO_KEY){
        lcd.setCursor(input,0);
        lcd.print(key);
        arr[place] = key;
        place++;
        input++;
    }

    if(place == 8)
    {
        for(int i = 0;i<8;i++)
        {
            if(arr[i] != password[i])
            {
                value = false;
                break;
            }
        }
        if(value == true)
        {
            lcd.setCursor(0,1);
            lcd.print("    Welcome!  ");
        }
        else
        {
            place = 0;
            value = true;
            lcd.setCursor(8,1);
            lcd.print("Wrong!");
            input = 6;
            lcd.setCursor(input,0);
            lcd.print("    ");
            delay(1000);
            lcd.setCursor(8,1);
            lcd.print("    ");
        }
    }
}
```

}

3.5. Kết quả:

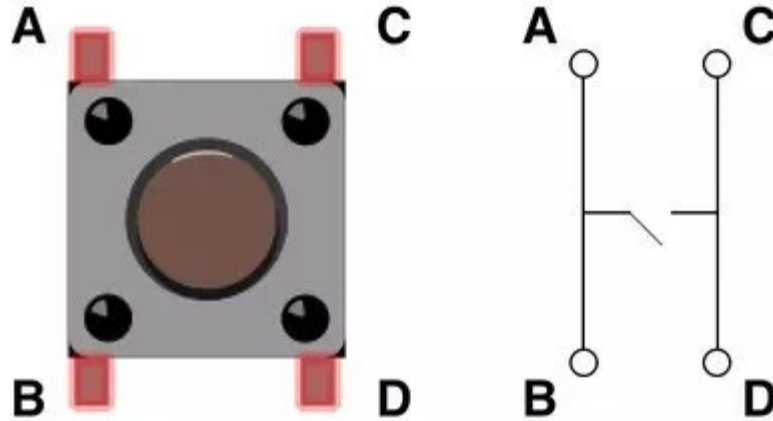


4. Lập trình Button điều khiển đèn phím

4.1. Button:



- Giống với công tắc đóng/ mở, nút nhấn cũng có cơ chế hoạt động giống như vậy. Thay vì chỉ có 2 chân như công tắc, nút nhấn có 4 chân chia làm 2 cặp. Những chân trong cùng một cặp được nối với nhau, những chân khác cặp thì ngược lại. Khi nhấn nút, cả 4 chân của nút nhấn đều được nối với nhau, cho phép dòng điện từ một chân bất kì có thể tới 3 chân còn lại, khi ngừng nhấn, 2 cặp sẽ tách rời, dòng điện sẽ không còn liên thông nữa.

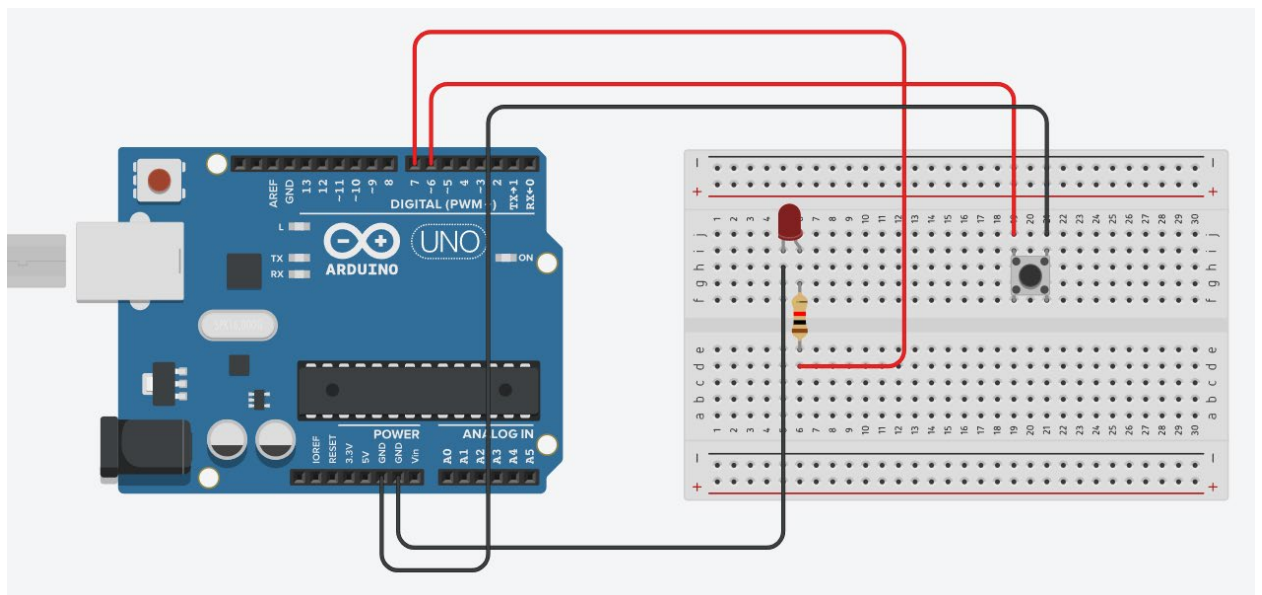


- Giờ ta sẽ thử sử dụng Button với Arduino: Làm đèn LED bật khi nhấn button đèn Led sáng, nhấn thêm lần nữa thì Đèn Led tắt.

4.2. Chuẩn bị phần cứng:

- 1 Button
- 1 Đèn Led
- 1 Điện trở tầm 220 ôm
- Bread Board

4.3. Sơ đồ đấu nối:



Arduino	Led
D7	(+)
GND	(-)

Arduino	Button
D6	Một đầu bất kỳ
GND	Đầu còn lại

4.4. Lập trình:

```

#define led 7
#define button 6
int status = 0;
void setup()
{
    pinMode(led,OUTPUT); //đặt đèn led trạng thái là đầu ra (output)
    pinMode(button,INPUT_PULLUP); //đặt nút nhấn là đầu vào PULLUP nghĩa là ban
    đầu nút nhấn ở trạng thái HIGH (mức cao)
}

void loop()
{
    if(digitalRead(button) == LOW){ // Nếu nút nhấn đang được nhấn
        delay(20); // Bỏ qua nhiễu
        if(digitalRead(button) == LOW) {// Kiểm tra thêm lần nữa để biết chắc nút
            nhân đã được nhấn
            if(status == 0){ //khi nhấn nút ở trạng thái đèn tắt
                status = 1; // Biến này để ghi nhớ trạng thái đèn
                digitalWrite(led,HIGH); // Bật đèn
            }
            else{ //Khi nhấn nút ở trạng thái đèn bật
                status = 0;
                digitalWrite(led,LOW); // Tắt Đèn
            }
        }
        while(digitalRead(button) == LOW); //Nhiệm vụ khi nào nút nhấn được thả
        ra thì thực tiếp
    }
}

```