

1

CÔNG NGHỆ
INTERNET OF THINGS
HIỆN ĐẠI

Làm quen với Arduino

Lưu hành nội bộ

A. MỤC TIÊU

- Giới thiệu một số khái niệm cơ bản trong lập trình nhúng.
- Làm quen với nền tảng Arduino.
- Giới thiệu một số thiết bị, đèn LED, nút bấm,... tại phòng thực hành để thực hiện một số kịch bản khác nhau.
- Lập trình Arduino để thể hiện các kịch bản đơn giản.

B. GIỚI THIỆU

1. Các khái niệm cơ bản

a. Input và Output

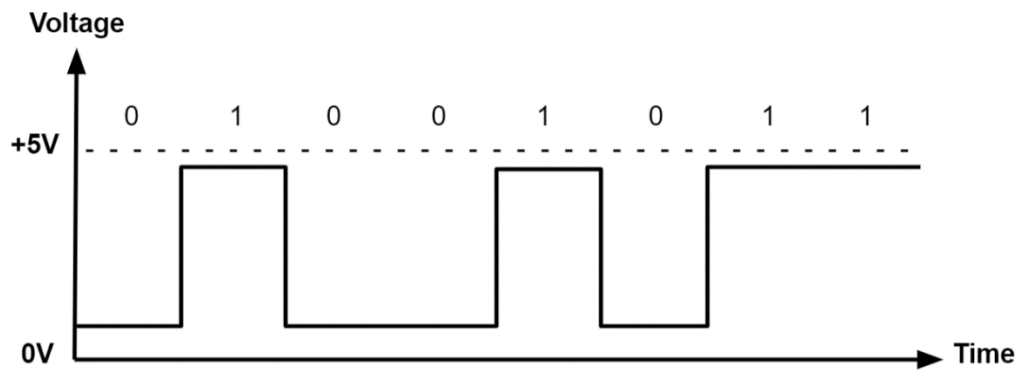
Input: những tín hiệu hoặc thông tin đi vào bo mạch. Ví dụ như tín hiệu từ một nút bấm, tín hiệu từ cảm biến nhiệt độ, cảm biến ánh sáng,...

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

b. Digital và Analog

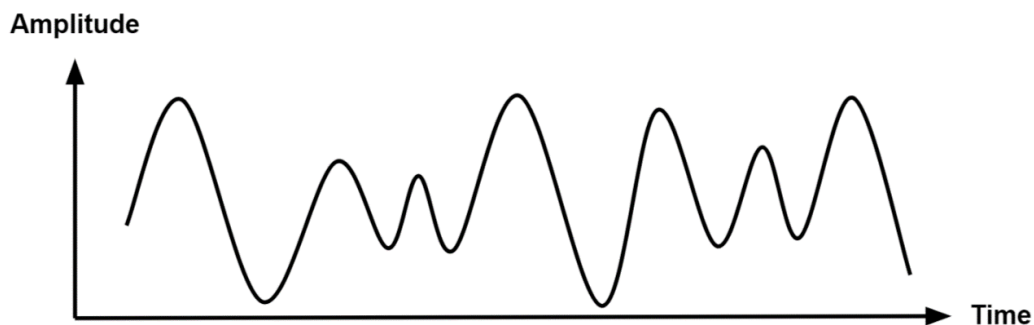
Digital (Tín hiệu kỹ thuật số): tín hiệu digital là tín hiệu biểu thị dữ liệu dưới dạng một chuỗi các giá trị rời rạc. Một tín hiệu digital chỉ có thể nhận một giá trị từ một tập hợp hữu hạn các giá trị có thể có tại một thời điểm nhất định.

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*

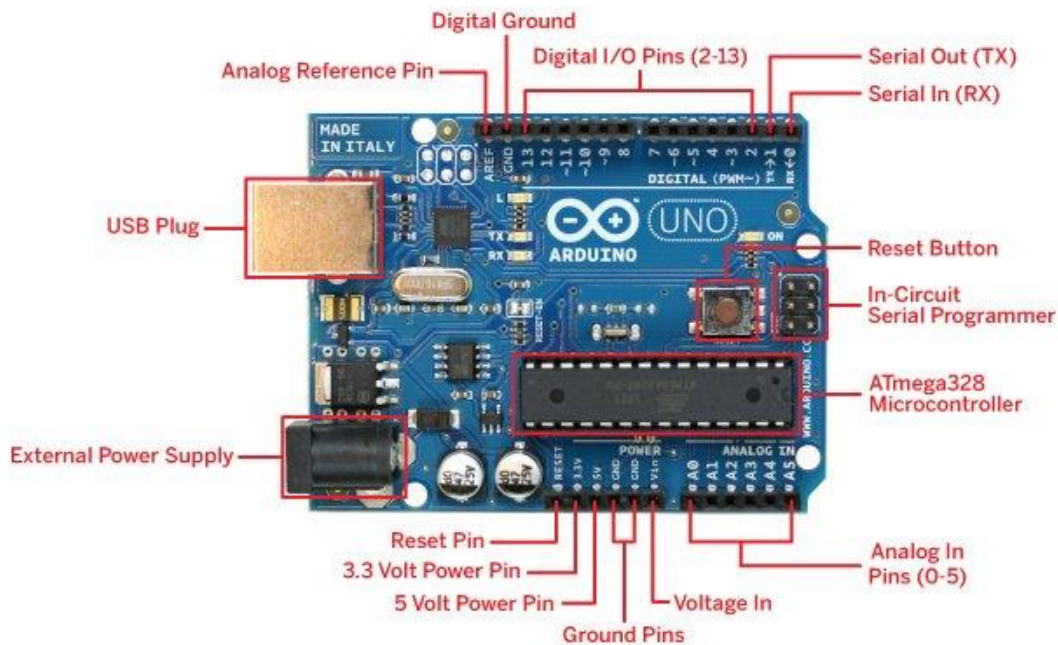
2. Arduino

a. Giới thiệu về 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.



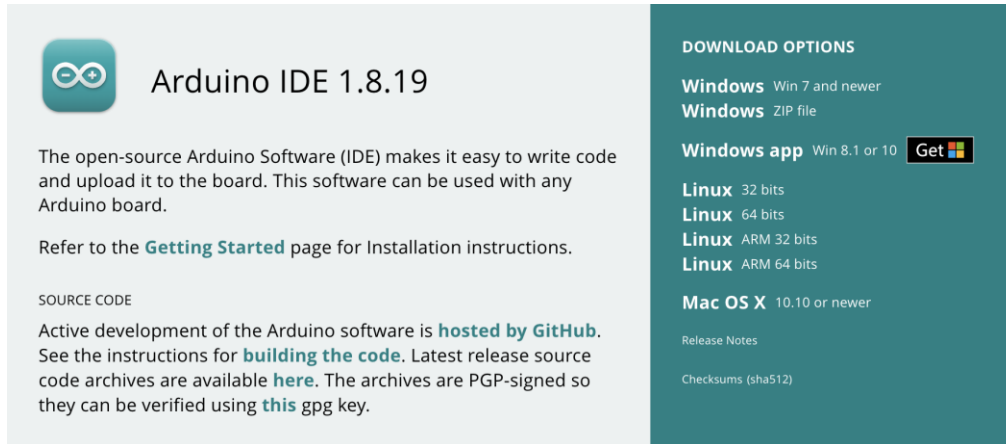
Hình 3. Mô tả Arduino Uno R3

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.

b. Arduino IDE

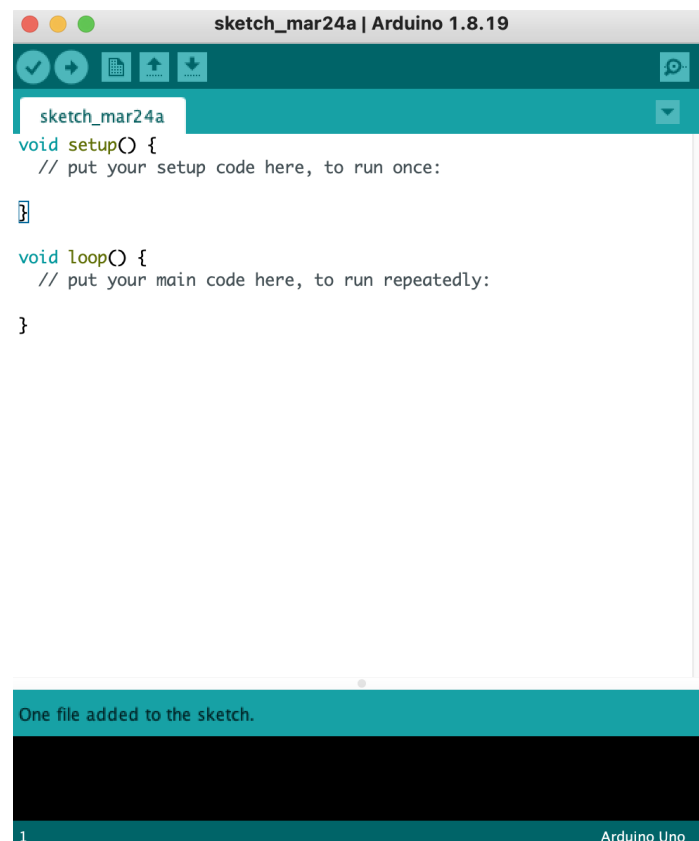
Arduino IDE là một phần mềm miễn phí mã nguồn mở, được sử dụng chủ yếu để viết và biên dịch mã và nạp vào board mạch. Đây là một phần mềm chính thống được phát hành

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>



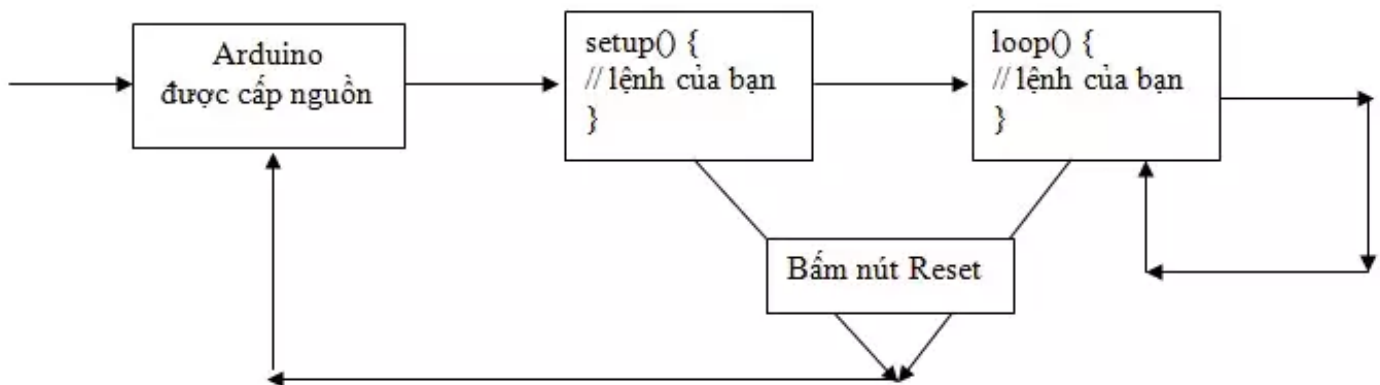
Hình 5. Giao diện chính của Arduino IDE

Hai chức năng chính cần lưu ý là **biên dịch mã** và **nạp mã** vào board mạch. Để biên dịch mã, sử dụng tính năng  trên thanh công cụ, sau khi biên dịch xong, thanh trạng thái sẽ báo “Done compiling”. Tương tự, sử dụng  để nạp code vào board mạch, sau khi chọn nút upload, Arduino IDE sẽ biên dịch mã chương trình và nạp vào thiết bị. Nếu có lỗi xảy ra, thanh trạng thái sẽ thông báo.

c. Lập trình Arduino cơ bản

Một chương trình 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 hoạt 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:

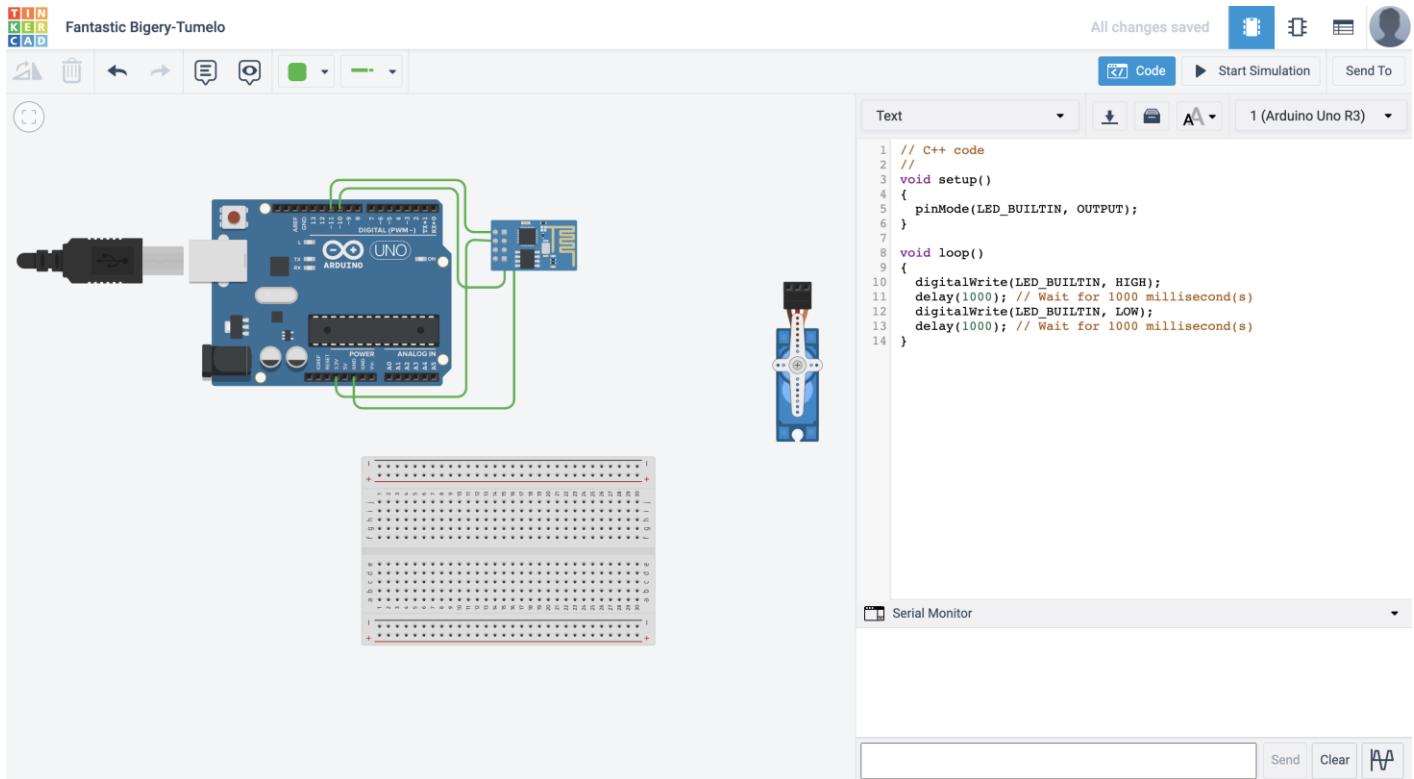
- ❖ **delay(ms)**: có nhiệm vụ dừng chương trình trong thời gian mili giây.
- ❖ **digitalWrite(PIN_NUMBER, VALUE)**: xuất tín hiệu digital VALUE ra một chân trên Arduino. Có 2 loại VALUE là HIGH (hay giá trị 1) và LOW (hay giá trị 0). Nếu xuất tín hiệu HIGH thì giá trị điện áp ở chân pin sẽ có giá trị 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.
- ❖ **`pinMode(PIN_NUMBER, MODE)`**: cấu hình MODE cho chân PIN_NUMBER là một đầu vào (INPUT) hay đầu ra (OUTPUT).

Giới thiệu Tinkercad

Tinkercad là một ứng dụng web miễn phí giúp chúng ta dễ dàng tiếp cận với các nền tảng phần ứng, mạch Arduino giả lập. Tại đây, ta có thể kéo thả các linh kiện, lập trình và chạy thử nghiệm các kịch bản tương tự như ở trên các thiết bị thật.



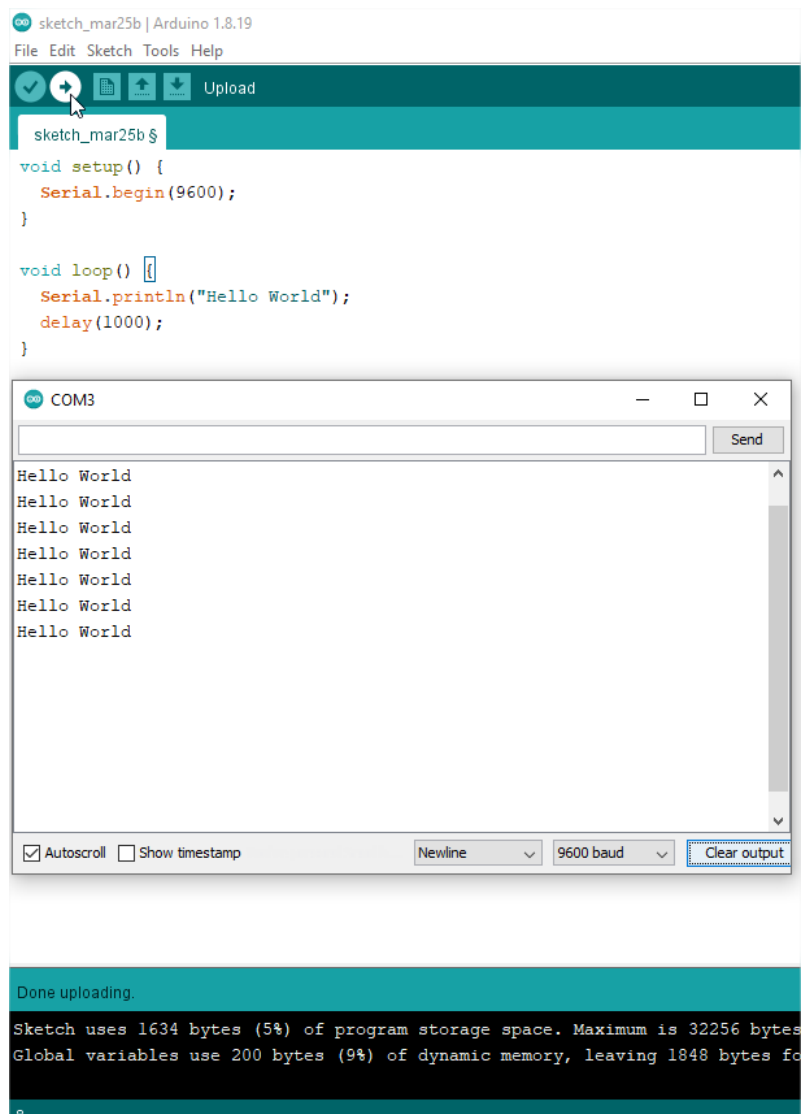
Hình 7. Giao diện của tinkercad

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.

Sau khi đã chọn đúng **Board** và **Port** (xem hướng dẫn chi tiết tại mục C).

Chọn  để vào giao diện serial.

*Hình 8. Giao tiếp Serial*

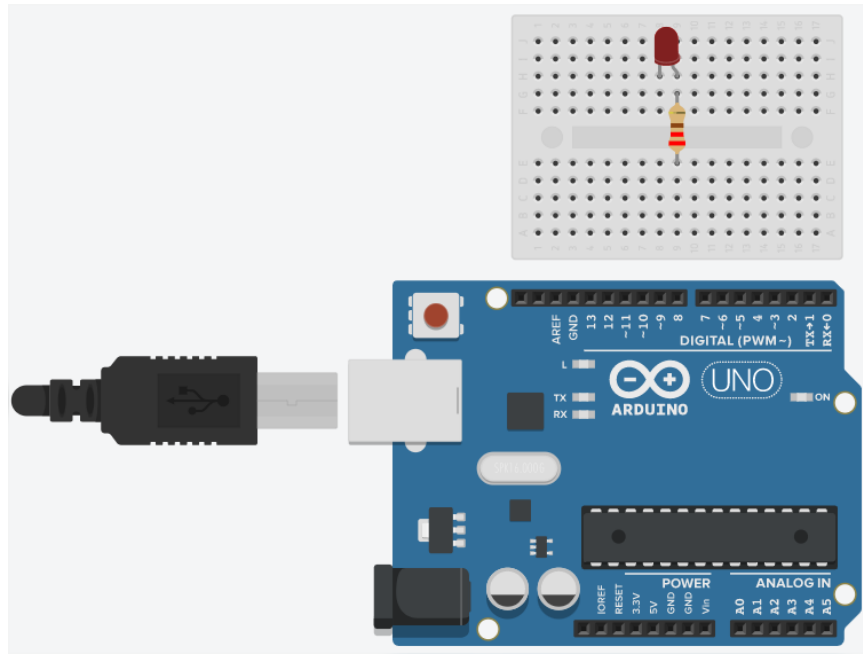
C. THỰC HÀNH

1. Tương tác đơn giản với đèn LED

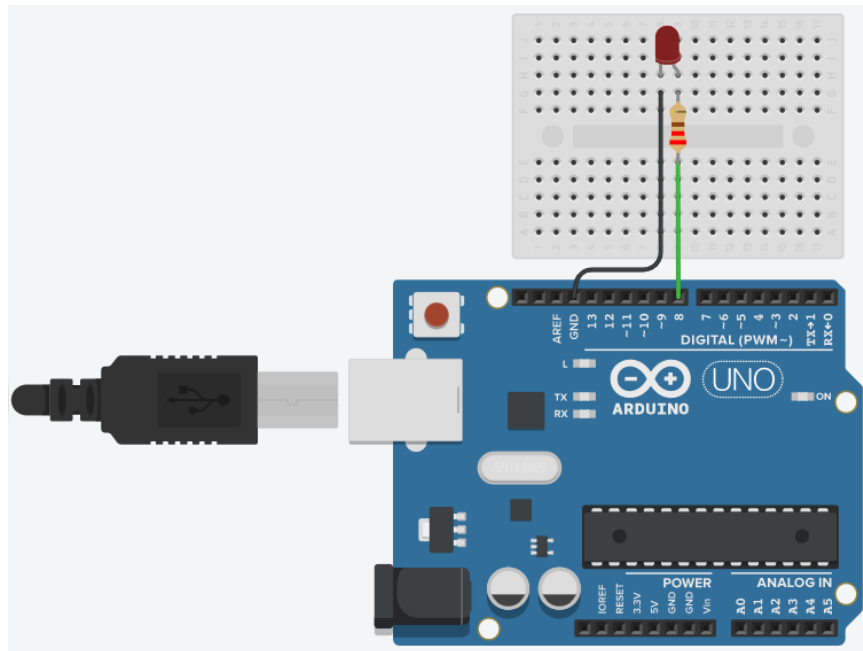
a. Chớp tắt đèn LED

Mô tả: Tại phần thực hành này, sinh viên tiến hành lập trình Arduino để điều khiển đèn LED sao cho mỗi 1 giây đèn sẽ sáng, tắt 1 lần.

- **Bước 1:** Tiến hành lựa chọn xem xét thiết bị, đấu nối với breadboard:



- **Bước 2:** Tiến hành đấu dây GND và chân điều khiển của Arduino vào đèn LED.



- **Bước 3:** Lập trình theo kịch bản được cho trước.

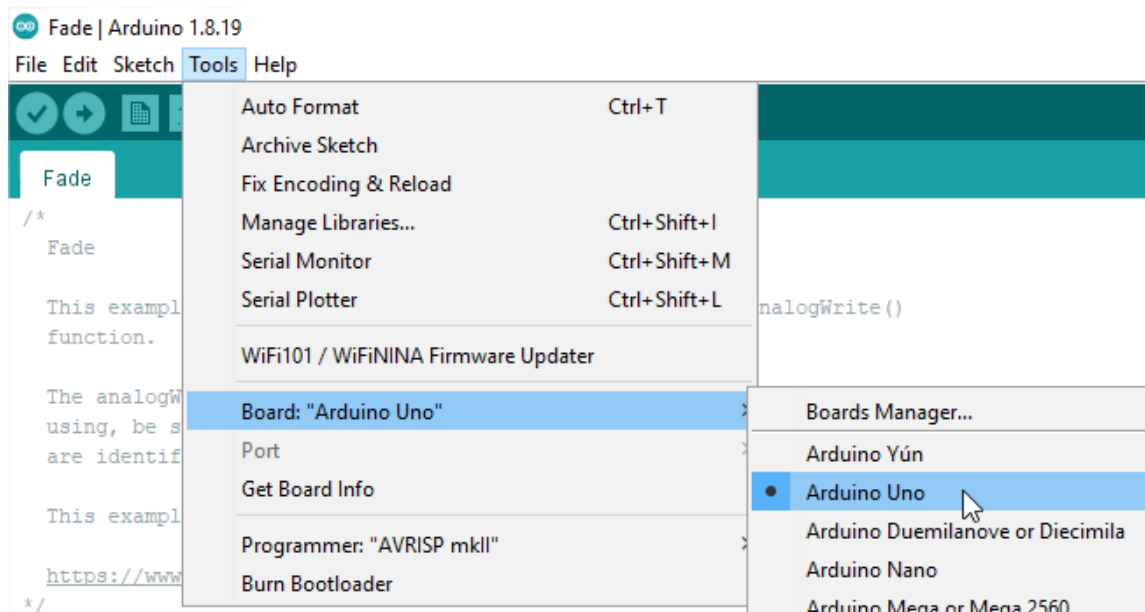
```
// C++ code
//
int LED = 8;

void setup()
{
```

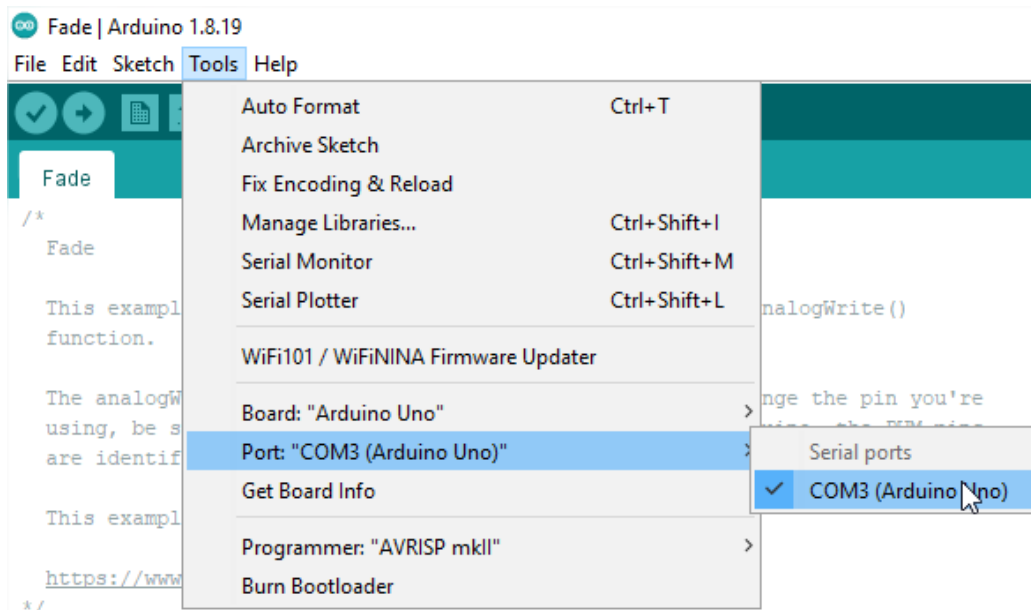
```
Serial.begin(115200);
pinMode(LED, OUTPUT);
}

void loop()
{
  digitalWrite(LED, HIGH);
  delay(1000); // Wait for 1000 millisecond(s)
  digitalWrite(LED, LOW);
  delay(1000); // Wait for 1000 millisecond(s)
}
```

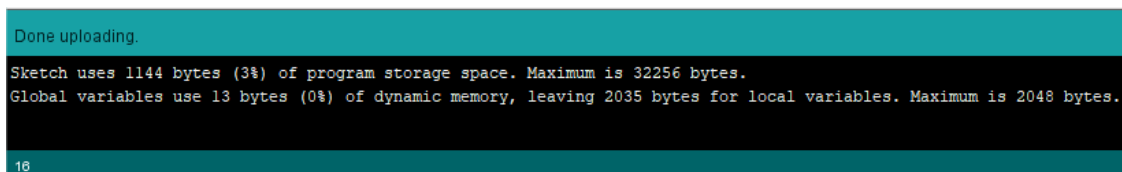
- **Bước 4:** Chọn **Tools** → **Board** để lựa chọn đúng thiết bị đang sử dụng, trong trường hợp này là “Arduino Uno”.



- **Bước 5:** Tại **Tools** → **Port**, tiến hành chọn cổng kết nối của thiết bị (cổng có dạng COMX với X là số, ví dụ COM3).



- **Bước 6:** Sử dụng  để tiến hành biên dịch và nạp code vào board mạch.



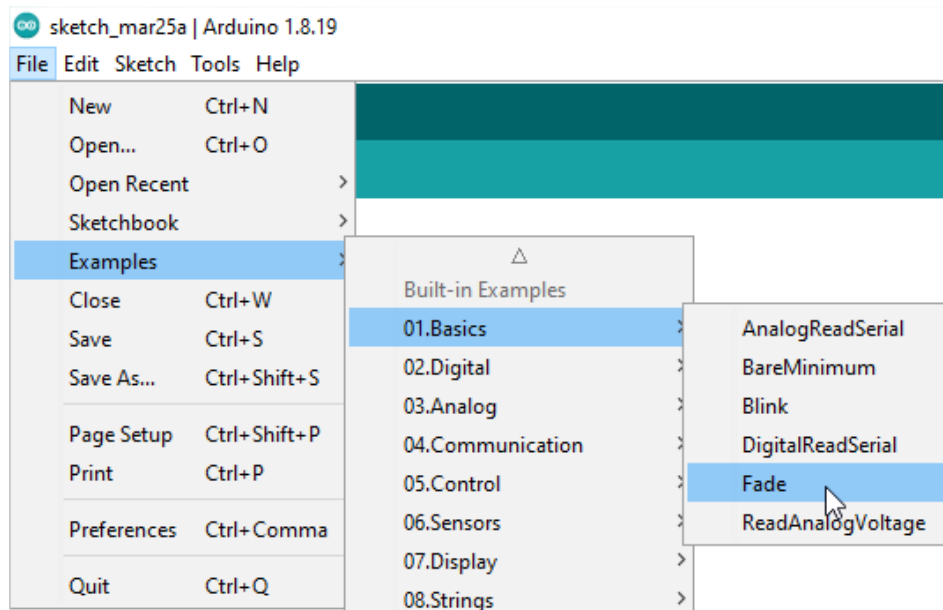
Nếu biên dịch và nạp code thành công, thanh trạng thái sẽ thông báo.

- **Bước 6:** Tiến hành quan sát kết quả trên board mạch.

b. Sử dụng code mẫu

Mô tả: Tại phần thực hành này, sinh viên sử dụng code mẫu đã có sẵn trong Arduino IDE để điều khiển đèn theo hiệu ứng fade.

- **Bước 1:** Mô hình tương tự mục a.
- **Bước 2:** Arduino IDE đã cung cấp sẵn code mẫu, thực hiện chọn **File** → **Examples** → **01. Basics** → **Fade**.

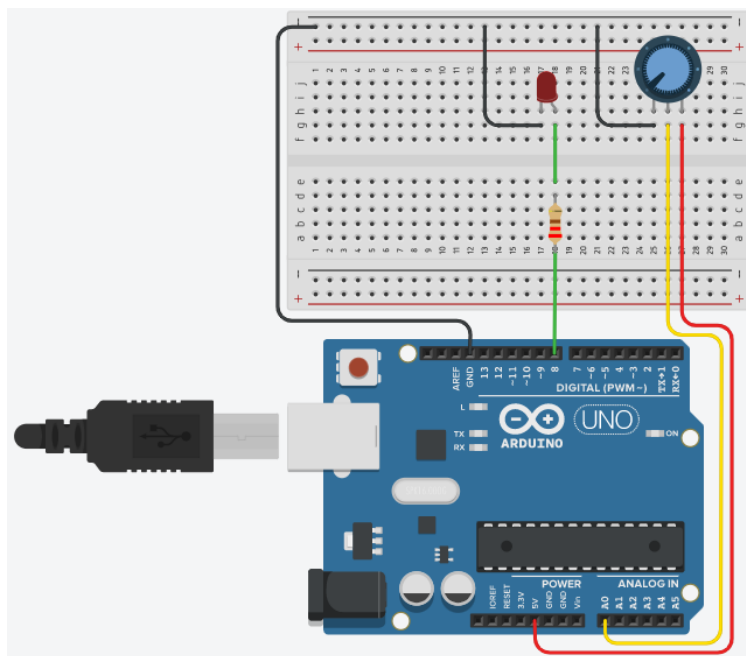


- **Bước 3:** Nạp code và kiểm tra kết quả.
- **Bước 4:** Giải thích nội dung code đã được nạp.

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

Mô tả: sinh viên tiến hành lập trình Arduino để lấy giá trị của biến trở được điều khiển từ người dùng, thông qua giá trị đó điều khiển độ sáng tối của đèn LED.

- **Bước 1:** Tương tự, lắp và đấu dây như mô hình bên dưới.



- **Bước 2:** Lập trình để đọc giá trị của biến trở và điều khiển đèn LED.

```
int LED = 8;

void setup()
{
  Serial.begin(115200);
  pinMode(LED, OUTPUT);
  pinMode(A0, INPUT);
}

int value = 0;

void loop()
{
  value = analogRead(A0);

  analogWrite(LED, value);
  Serial.print("Value of A0: ");
  Serial.print(value);
  delay(100);
}
```

- **Bước 3:** Chọn Board và Port tương ứng.
- **Bước 4:** Nạp code và mở **Serial Monitor** để quan sát kết quả thu được.

D. YÊU CẦU & NỘI BÀI

1. Yêu cầu

1. Xây dựng kịch bản gồm có 12 đèn LED và 1 nút bấm. Tùy số lượng lần bấm nút thì số lượng đèn sẽ sáng lên tương ứng. Sau khi đạt ngưỡng tối đa thì quay lại với giá trị 1 đèn sáng.
2. Xây dựng kịch bản gồm có 3 đèn LED và 1 biến trở. Điều chỉnh tốc độ đèn lần lượt sáng dựa vào giá trị của biến trở được điều khiển. Giá trị của biến trở chia thành 3 mức: chậm – trung bình – nhanh. Các kết quả này được thể hiện tại Serial.

Lưu ý: đèn được sáng theo chiều từ trái sang phải và ngược lại.

3. Mô phỏng trò chơi “**nén xúc sắc**”, xây dựng kịch bản gồm có 6 đèn và 1 nút bấm. Khi bấm lần đầu tiên thì tất cả các đèn lần lượt sáng lên và hiển thị lên số mà xúc sắc tung ra dựa vào số đèn được hiển thị. Số đèn này sẽ phải dừng lại 1 giây để người dùng theo dõi và sẽ chớp tắt cùng lúc 6 lần. Cứ tiếp tục như thế khi bấm nút lần tiếp theo.

4. Mô phỏng “**đèn giao thông**”, xây dựng kịch bản gồm có 1 đèn LED 7 đoạn, 3 đèn khác màu (đỏ, vàng, xanh). LED 7 đoạn sẽ đếm ngược từ số lượng thời gian quy định trong mỗi 1 giây, khi đến 0 thì chuyển sang sáng 1 đèn khác. Bắt đầu từ xanh → vàng → đỏ, cứ như thế và lặp lại. led 7 on

Ghi chú: đèn xanh sáng trong 9 giây, đèn vàng sáng trong 2 giây và đèn đỏ sáng trong 6 giây.

5. Mô phỏng trò chơi “**thử tài đoán số**”, xây dựng kịch bản gồm có 10 đèn, 3 nút bấm. Trò chơi gồm có 3 level, tùy vào tốc độ nháy đèn để hiển thị. Người chơi sẽ chọn 1 trong 3 nút bấm với điều kiện như sau để cộng điểm: chọn vào nút chính xác theo qui tắc (số lượng đèn sáng % 3), nếu chọn sai hoặc không chọn trong vòng 3 giây thì điểm sẽ bị trừ. Nếu chọn sai thì mặc định level sẽ reset về 1. Các thông tin về điểm số và tình trạng cộng điểm hay trừ điểm sẽ được hiển thị qua màn hình LCD. 4-level giây

2. Yêu cầu 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 các kết quả đạt được.
 - Tải mẫu báo cáo thực hành và trình bày theo mẫu được cung cấp.

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

NhomY-LabX_MSSV1_MSSV2

Ví dụ: Nhom1-Lab01_25520001_25520002

- 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!