

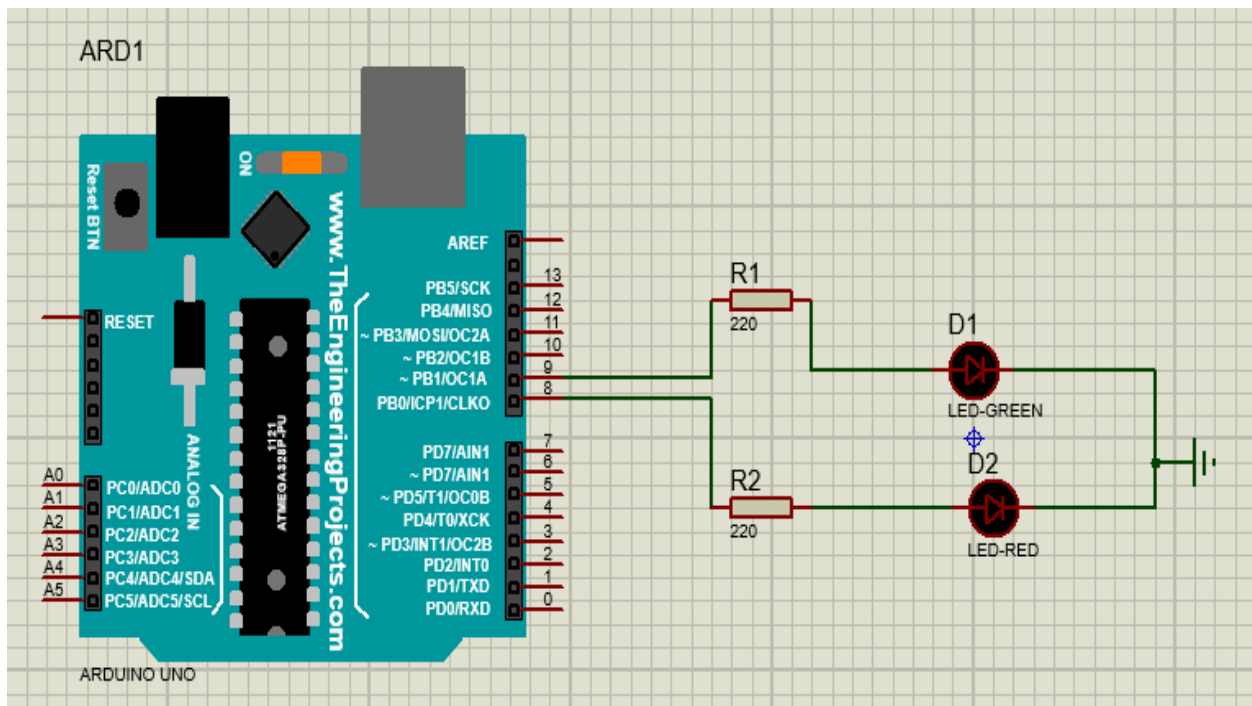
THỰC HÀNH LẬP TRÌNH NHÚNG

BÀI TẬP TỔNG HỢP

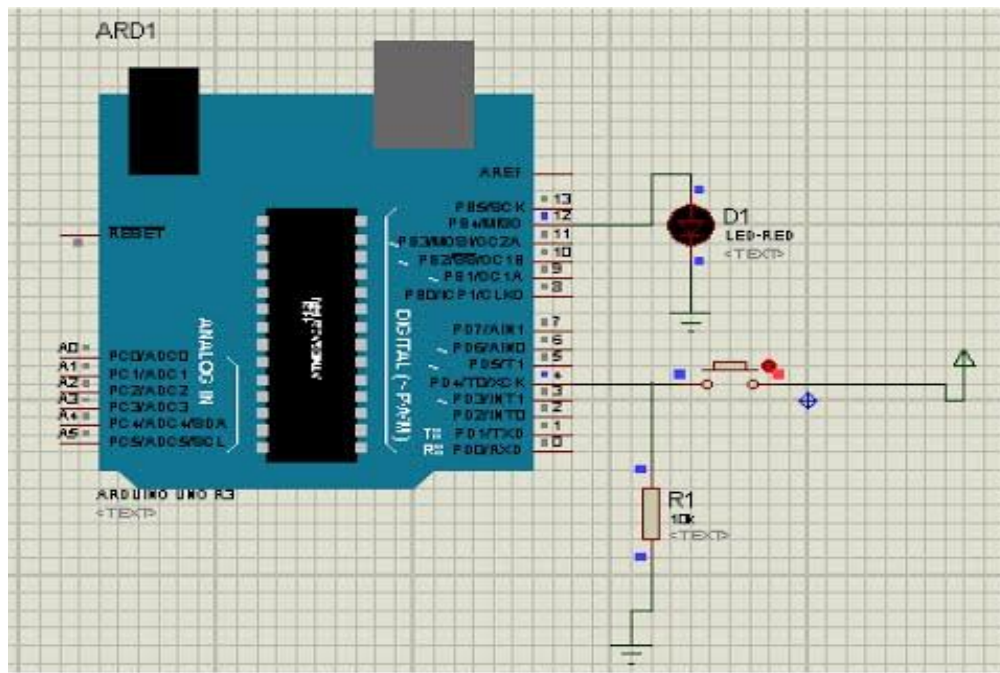
- Arduino là một bo mạch vi xử lý được dùng để lập trình tương tác với các thiết bị phần cứng như cảm biến, động cơ, đèn hoặc các thiết bị khác.
- Đặc điểm nổi bật của Arduino là môi trường phát triển ứng dụng cực kỳ dễ sử dụng, với một ngôn ngữ lập trình có thể học một cách nhanh chóng ngay cả với người ít am hiểu về điện tử và lập trình.
- Arduino được sử dụng rất phổ biến trong các ứng dụng thực tế chính là nhờ mức giá rất thấp và tính chất nguồn mở từ phần cứng tới phần mềm.
- Arduino Uno là sử dụng chip Atmega328. Nó có 14 chân digital I/O, 6 chân đầu vào (input) analog, thạch anh dao động 16Mhz.

1. Giao tiếp Arduino với LED

1.1. Sơ đồ mạch



Hình 1. Sơ đồ mạch điều khiển LED



Hình 2. Sơ đồ mạch điều khiển LED với button

1.2. Các lệnh điều khiển

 `pinMode()`

- `pinMode(8, INPUT);` /*Khai báo chân nào của Arduino là đọc dữ liệu của nút bấm*/

 `digitalWrite()`

- `digitalWrite(13,HIGH);` /*Cấp một điện thế là 5V vào chân số Digital 13. Điện thế sẽ đi qua điện trở 220ohm rồi đến đèn LED (sẽ làm nó sáng mà không bị cháy*/
- `digitalWrite(13,LOW);` /*Lúc này điện thế tại chân 13 sẽ là 0 V => đèn LED tắt.*/

 `delay()`

- `delay(1000);` /*Để thấy trạng thái bật và tắt của đèn LED, chương trình phải dừng trong một khoảng thời gian đủ lâu để mắt cảm nhận được. Hàm delay được tạo ra để làm việc này*/

 `digitalRead()`

- `digitalRead(8);` /*Đọc trạng thái của button được kết nối với chân số 8 của Arduino. Khi button được nhấn thì trả về kết quả là HIGH, khi button chưa được nhấn thì trả về kết quả là LOW*/

1.3. Chương trình điều khiển LED đơn giản

Điều khiển LED sáng tắt

sketch_dec02a §

```
1 int led = 13; // chân digital 13 của Arduino kết nối với đèn LED
2 void setup() {
3   pinMode(led, OUTPUT); // đặt 'led' là OUTPUT
4 }
5 void loop() {
6   digitalWrite(led, HIGH); // bật đèn led sáng
7   delay(1000); // dùng chương trình trong 1 giây => thấy đèn sáng được 1 giây
8   digitalWrite(led, LOW); // tắt đèn led
9   delay(1000); // dùng chương trình trong 1 giây => thấy đèn tối được 1 giây
10 }
```

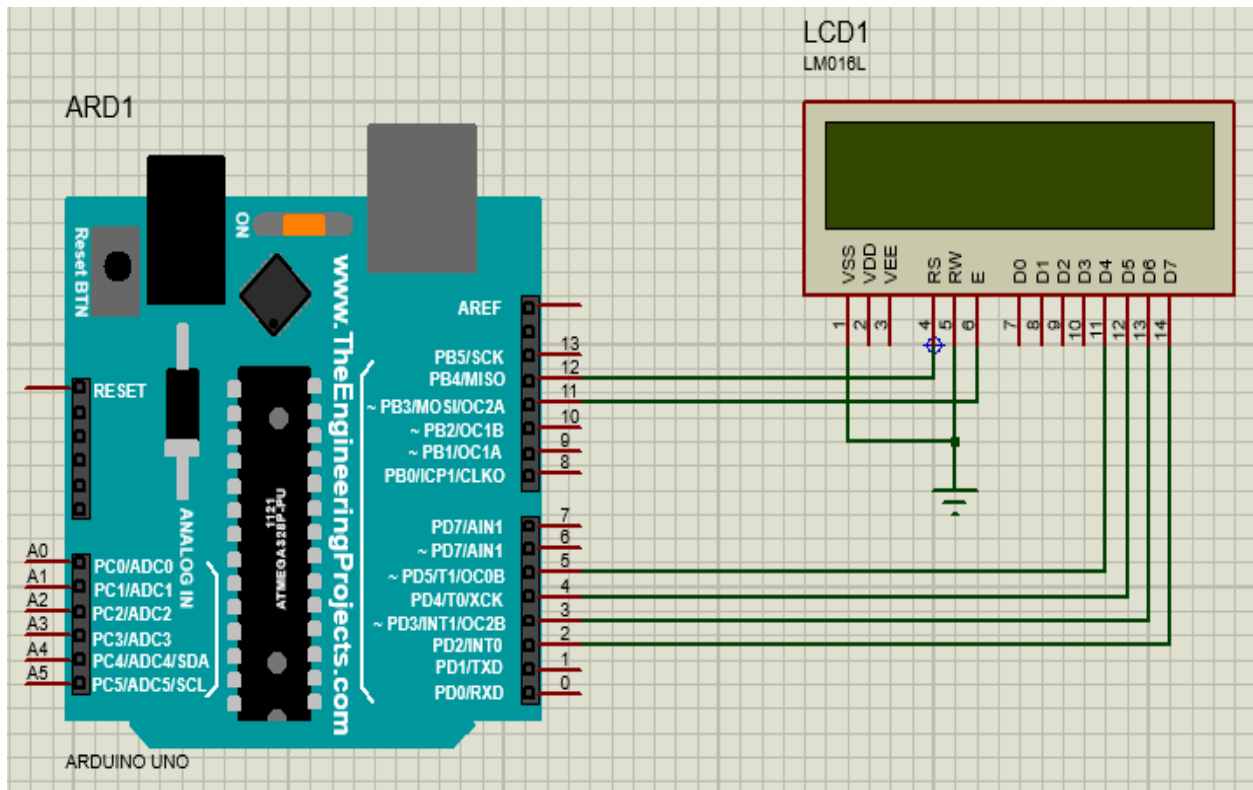
Điều khiển LED sáng tắt với button

sketch_sep17a §

```
1 int button = 8;
2 int led = 7;
3 void setup() {
4   pinMode(button, INPUT); // Cài đặt chân D11 ở trạng thái đọc dữ liệu
5   pinMode(led, OUTPUT); // Cài đặt chân D2 dưới dạng OUTPUT
6 }
7
8 void loop() {
9   int buttonStatus = digitalRead(button); // Đọc trạng thái button
10  if (buttonStatus == HIGH) { // Nếu mà button bị nhấn
11    digitalWrite(led, HIGH); // Đèn led sáng
12  } else { // ngược lại
13    digitalWrite(led, LOW);
14  }
15 }
```

2. Giao tiếp Arduino với LCD

2.1. Sơ đồ mạch



Hình 3. Sơ đồ mạch giao tiếp Arduino với LCD

2.2. Cách lệnh điều khiển

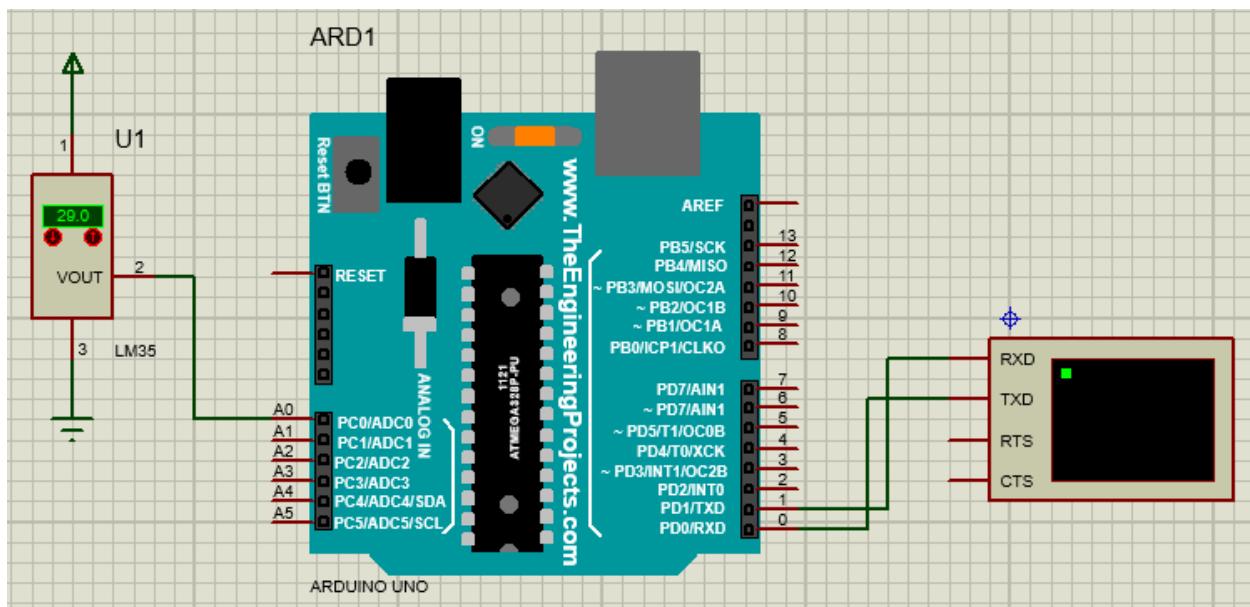
- ✚ `#include <LiquidCrystal.h>`
 - Khai báo thư viện giao tiếp Arduino với LCD.
- ✚ `LiquidCrystal lcd(12, 11, 5, 4, 3, 2);`
 - Khai báo các cổng RS, E, D4-D7
- ✚ `lcd.begin(16, 2);`
 - Khai báo loại LCD giao tiếp
- ✚ `lcd.clear();`
 - Lệnh xóa màn hình LCD
- ✚ `lcd.setCursor(6,0);`
 - Lệnh đặt vị trí con trỏ trên màn hình LCD
- ✚ `lcd.print("HELLO");`
 - Lệnh xuất một dòng dữ liệu trên màn hình LCD
- ✚ `lcd.noDisplay();`
 - Lệnh ẩn dòng dữ liệu trên màn hình LCD
- ✚ `lcd.Display();`
 - Lệnh cho hiển thị dòng chữ trên màn hình LCD

2.3. Chương trình giao tiếp LCD đơn giản

```
sketch_dec02a §  
1 //khai báo thư viện giao tiếp LCD  
2 #include <LiquidCrystal.h>  
3 // khởi tạo đối tượng LCD giao tiếp với Arduino  
4 LiquidCrystal lcd(12, 11, 5, 4, 3, 2);  
5 void setup() {  
6   lcd.begin(16, 2); //màn hình LCD có kích thước 16x2  
7 }  
8 void loop() {  
9   lcd.clear(); //xóa màn hình LCD  
10  lcd.setCursor(6,0); //đặt vị trí xuất hiện ký tự trên LCD  
11  lcd.print("HELLO"); //xuất dòng chữ HELLO trên LCD  
12  lcd.setCursor(3,1);  
13  lcd.print("ARDUINO");  
14  delay (1000); //dừng 1s để thấy dòng chữ xuất hiện trên LCD  
15 }
```

3. Giao tiếp Arduino với LM35

3.1. Sơ đồ mạch



Hình 4. Sơ đồ mạch giao tiếp cảm biến nhiệt độ LM35

3.2. Các lệnh điều khiển

```
int sensorPin = A0; //khai báo chân giao tiếp Arduino với LM35  
int reading = analogRead(sensorPin); //đọc hiệu điện thế từ chân VOUT của LM35  
float voltage = reading * 5.0/1024.0;  
float temp = voltage * 100.0; // đổi ra độ C
```

✚ $\text{float tempF} = (\text{temp} * 1.8) + 32; //$ đổi từ độ C sang độ F

3.3. Chương trình giao tiếp với LM35

```
sketch_nov30a$  
1 int val;  
2 int tempPin = A0;//chân kết nối với LM35  
3 void setup() {  
4     Serial.begin(9600);//khởi động Serial  
5 }  
6  
7 void loop() {  
8     val = analogRead(tempPin);//đọc nhiệt độ từ cảm biến  
9     float nv = (val/1024)*5000;  
10    float cel = nv/10;//đổi ra độ C  
11    Serial.print("Temperature=");  
12    Serial.print(cel);//xuất ra nhiệt độ  
13    Serial.print("*C");  
14    Serial.println();  
15    delay(1000);//cứ mỗi 1s đọc lại nhiệt độ  
16 }
```

4. Bài tập

4.1. Cấu hình đường dẫn file Hex của Arduino IDE

✚ Cấu hình đường dẫn biên dịch file Hex lưu trong đường dẫn *C:\\HexFiles*.

4.2. Bài tập giao tiếp Arduino với LCD

✚ Mỗi bài tập tạo một project trong Proteus với tên là *th_baitapx.pdsprj* và lưu trong thư mục *D:\\LTNhung_MSSV_HoTen_Lop*. Ví dụ: *th_baitap1.pdsprj* lưu trong thư mục *D:\\LTNhung_0469171001_HuynhLongAn_SCLRMT17A*.

✚ Code Arduino của mỗi bài tập lưu với tên *th_baitapx.ino* và lưu trong thư mục *D:\\LTNhung_MSSV_HoTen_Lop*. Ví dụ: *th_baitap1.ino* lưu trong thư mục *D:\\LTNhung_0469171001_HuynhLongAn_SCLRMT17A*.

Bài tập 1: Viết chương trình điều khiển 10 đèn Led. Mỗi lần random ra 10 số (mỗi số tương ứng với mỗi đèn), nếu số của đèn nào là số lẻ thì đèn đó sẽ sáng, ngược lại sẽ tắt. Ví dụ: ta có đèn 1 đến đèn 10, random một con số cho đèn 1 được số 2 (số chẵn), đèn 2 được số 5 (số lẻ), đèn 3 được số 1 (số lẻ),... thì đèn 1 tắt, đèn 2 sáng, đèn 3 sáng,... Gợi ý: sử dụng hàm random(min, max).

Bài tập 2: Viết chương trình điều khiển đèn giao thông tại ngã ba (sử dụng 6 đèn tương ứng với mỗi bên đường).

Bài tập 3: Điều khiển tốc độ chớp của đèn sử dụng 2 button. Một button khi nhấn thì điều khiển cho đèn chớp nhanh hơn, một button khi nhấn thì điều khiển cho đèn chớp chậm hơn.

Bài tập 4: Thiết kế mạch gồm 2 đèn LED (vàng và đỏ) và 1 nút nhấn. Ban đầu 2 đèn cùng tắt, nhấn nút lần 1: hai đèn cùng sáng, nhấn lần 2: hai đèn cùng tắt, nhấn lần 3: đèn vàng sáng đèn đỏ tắt, nhấn lần 4: đèn vàng tắt đèn đỏ sáng. Cứ tuần tự như vậy.

Bài tập 5: Thiết kế mạch gồm 1 LCD, 1 nút bấm và 1 đèn LED. Khi màn hình tắt thì đèn LED sáng và ngược lại.

Bài tập 6: Thiết kế mạch gồm 1 LCD và 3 nút bấm (1 nút tắt màn hình, 1 nút qua trái, 1 nút qua phải). Hiện “HELLO” lên màn hình.

- a. Xử lý bấm nút tắt màn hình
- b. Nhấn nút qua trái thì dời chữ qua trái hai vị trí
- c. Nhấn nút qua phải thì dời chữ qua phải hai vị trí

Bài tập 7: Vẽ chữ Æ, Â, Đ, Ê, Ô, Ơ và Ư trên màn hình LCD.


Bài tập 8: Thiết kế bàn phím nhập ký tự UNICODE gồm các chữ cái EOFs (EE → Ê; OO → Ô; EF → È; ES → É; OF → Ò; OS → Ó).


Bài tập 9: Thiết kế mạch gồm 1 LCD và 1 LM35, 2 nút nhấn. Viết chương trình có một nút nhấn cho phép hiển thị theo °C, một nút nhấn cho phép hiển thị độ F.

Bài tập 10: Thiết kế mạch gồm 1 LCD, 1 LM35, 1 nút nhấn, 1 đèn xanh, 1 đèn đỏ, 1 đèn vàng. Các thiết bị có các chức năng sau:

- LCD hiển thị nhiệt độ cho LM35.
- Nút nhấn cho phép chuyển chế độ hiển thị độ đo (độ F sang °C và ngược lại).
- Đèn xanh sáng khi nhiệt độ dưới 17 °C, đèn vàng sáng khi nhiệt độ từ 17 °C - 30 °C và đèn đỏ sáng khi nhiệt độ lớn hơn 30 °C.

5. Chuẩn bị bài học tiếp theo

 Bài học buổi sau: **Kiểm tra 60 phút**

 Mỗi sinh viên ôn tập:

- Sử dụng Arduino IDE.
- Sử dụng Proteus.
- Code giao tiếp Arduino với LED, LCD, LM35.