**BÁO CÁO LẬP TRÌNH NHÚNG**

Họ và tên: Trần Văn Khánh

MSSV: 61133801

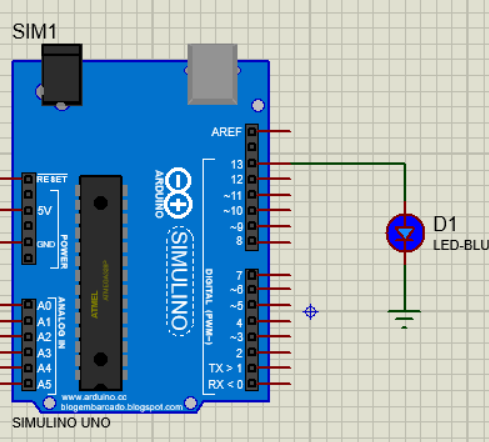
Lớp: 61-CNTT2

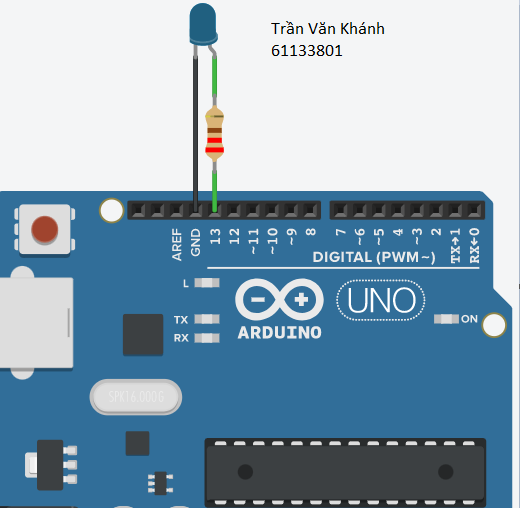
# Bài 1. Led nhấp nháy

## Mô tả

Hệ thống được thiết kế thực hiện việc lập trình điều khiển bật/tắt đèn LED trong thời gian nhất định, đèn được nối với cổng số 13 của mạch Arduino.

## Sơ đồ mạch

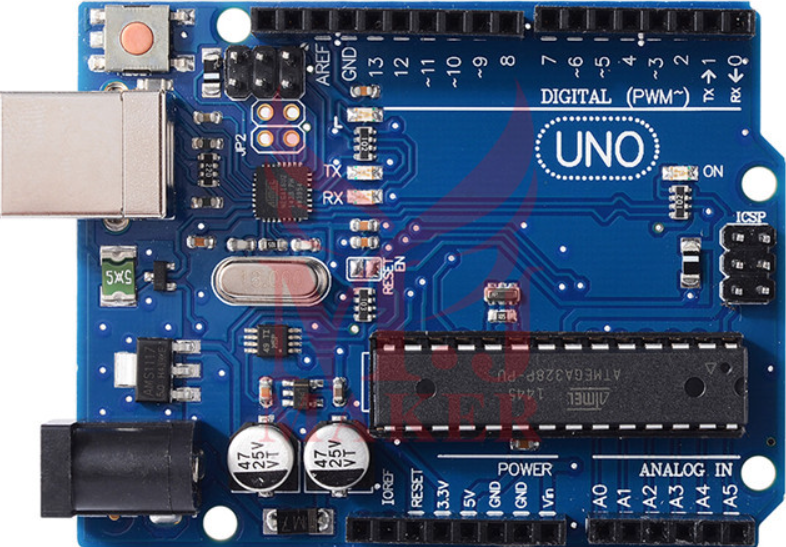


Hình 1 Sơ đồ thiết kế nối mạch hệ thống

Hình 2: Sơ đồ thiết kế nháy Led trên TinkerCard

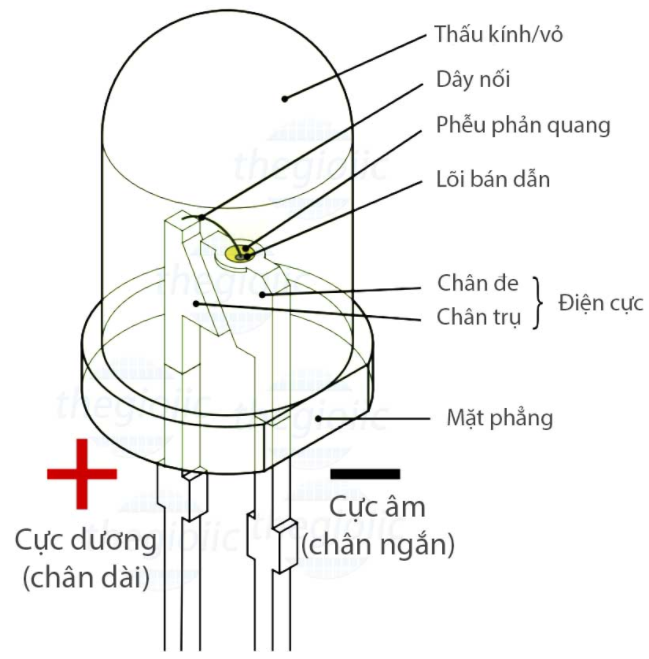
## Đặc điểm linh kiện

* 1 mạch Arduino Uno: 14 chân digital dùng để đọc hoặc xuất tín hiệu. Chúng chỉ có 2 mức điện áp là 0V và 5V với dòng vào/ra tối đa trên mỗi chân là 40mA. Ở mỗi chân đều có các điện trở pull-up từ được cài đặt ngay trong vi điều khiển ATmega328 (mặc định thì các điện trở này không được kết nối).

****

Mạch ArduinoUNO R3

* 1 đèn Led: gồm hai chân, một chân nối vào cổng Digital của Arduino, chân còn lại nối vào cổng GND (cực âm của nguồn điện cấp cho Arduino UNO thường gọi là “Tiếp đất” nghĩa là bằng 0V). Đảm bảo rằng dòng điện qua đèn Led không bao giờ lớn hơn 20mA.



Cấu tạo Đèn Led

## Code chương trình

int ledPin = 13;

void **setup**() {

pinMode(ledPin, OUTPUT); // Kết nối với cổng 13 của Arduin UNO

}

void **loop**() {

digitalWrite(ledPin, HIGH); // Bật đèn

delay(300); // Để đèn sáng trong 0.3s

digitalWrite(ledPin, LOW); // Tắt đèn

delay(300);

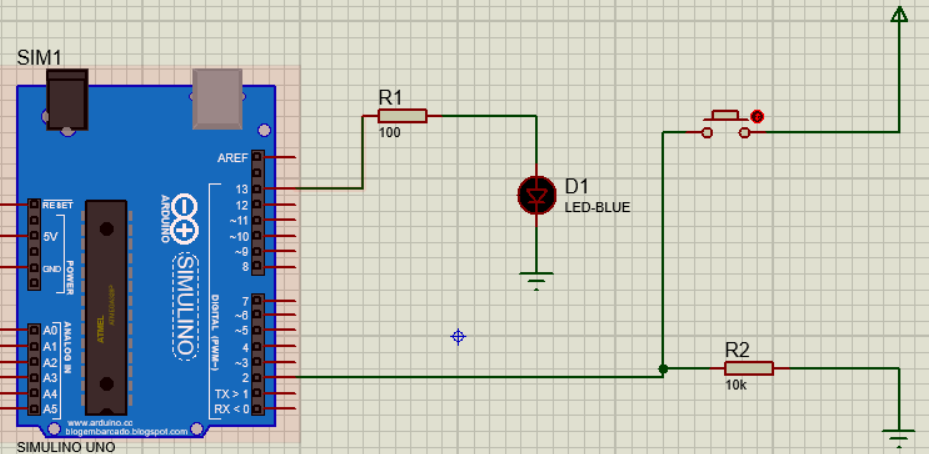
}

# Bài 2. Nháy Led có nút bấm

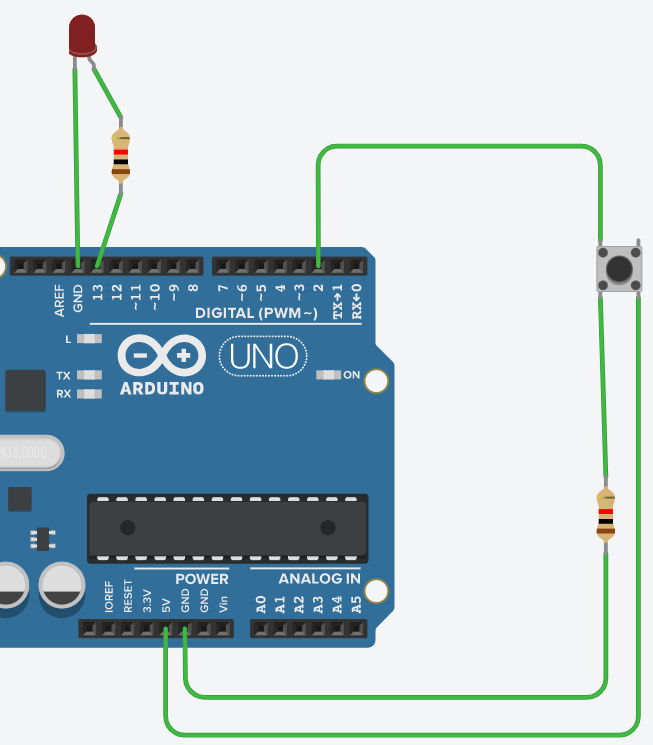
## Mô tả

Bài này thực hiện việc lập trình làm bật/tắt Led thông qua 1 nút bấm. Đèn Led được nối với cổng số 13 của mạch Arduino thông qua 1 điện trở.

## Sơ đồ mạch



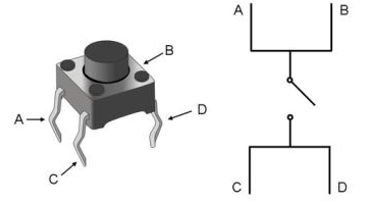
Sơ đồ thiết kế mạch hệ thống trên Proteus



Sơ đồ thiết kế hệ thống trên TinkerCard

## Linh kiện

* 1 mạch Arduino Uno
* 1 đèn Led
* 2 điện trỡ: điện trỡ dùng để hạn chế cường độ dòng điện qua đèn, giúp đèn sáng tốt không bị cháy.
* 1 nút bấm: là thiết bị đầu vào đơn giản nhất có thể được kết nối với bất kỳ bộ vi điều khiển hoặc bộ xử lý như Arduino. Một nút nhấn đơn giản nhất bao gồm bốn chân. Trong đó, các chân A và B, chân C và D được kết nối với nhau. Vì vậy, mặc dù nút nhấn có bốn chân nhưng về mặt kỹ thuật, nhưng chúng ta chỉ sử dụng hai chân.



Cấu tạo nút bấm

## Code chương trình

int x = 0;

void **setup**() {

pinMode(2, INPUT);

pinMode(13, OUTPUT);

}

void **loop**() {

x = digitalRead(2);

if( x == HIGH) {

digitalWrite(13, HIGH);

} else {

digitalWrite(13, LOW);

}

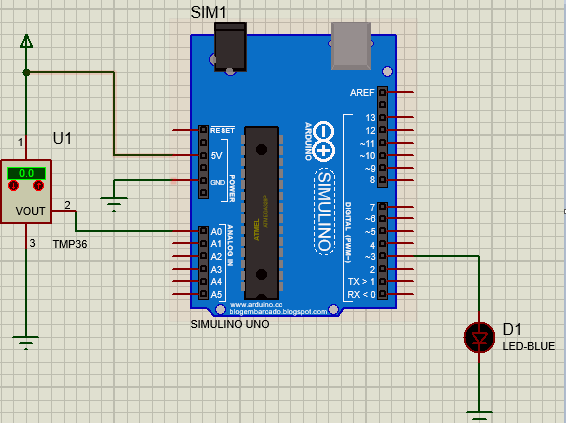
}

# Bài 3. Nháy Led với cảm biến nhiệt độ

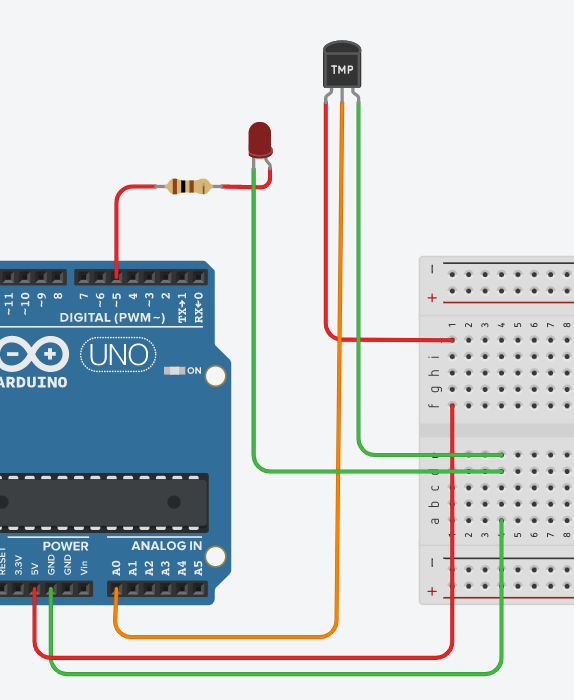
## Mô tả

Bài này thực hiện việc lập trình điều khiển Led sáng tắt với cảm biến nhiệt độ. Đèn Led sẽ nối với cổng số 11 của mạch Arduino và 1 cảm biến nhệt độ được nối với cổng A0 của mạch Arduino.

## Sơ đồ mạch điện



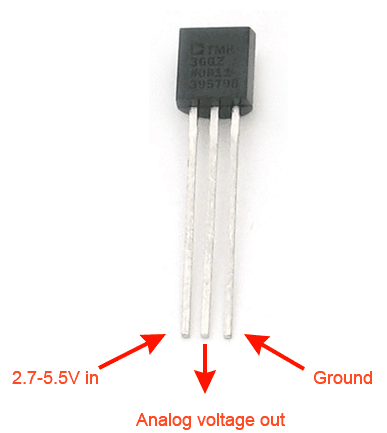
Sơ đồ thiết kế trên Proteus



Sơ đồ thiết kế trên Tinkercard

## Linh kiện

* 1 mạch Arduino Uno
* 1 đèn Led
* 1 điện trở 100 Ω
* 1 cảm biến nhiệt độ TMP36:
  + Chân số 1 là chân cấp nguồn 5V (chân này bạn có thể cắm vào nguồn 5V của Arduino khi sử dụng nó với Arduino).
  + Chân thứ 2 là chân xuất tín hiệu tương tự (tín hiệu dạng xung).
  + Chân thứ 3 là chân nối mát hay chân GND(khi sử dụng với Arduino các bạn có thể lấy từ chân Gnd từ Arduino.
  + Cảm biến TMP 36 này xuất tín hiệu là tín hiệu điện áp, tín hiệu này tỷ lệ thuận với nhiệt độ. Cứ 10mV tương đương với 1 độ C, nó có thể đo được nhiệt độ từ **-40 ==>125** độ C (hơn hẳn so với LM35 các bạn nhé). Bởi vì nó xuất một tín hiệu tương tự nên khi làm việc với Arduino chúng ta sử dụng hàm *analogRead().* Khi Arduino làm việc nó sẽ trả tín hiệu tương tự này về giá trị từ 0-1023 tùy thuộc vào điện áp từ 0-5V(5000mV). Điện áp thực tế sẽ là điện áp mà Arduino đọc được sau đó nhân với (5000/1024).



Hình Cảm biến nhiệt độ STM36

## Code chương trình

void **setup**() {

Serial.begin(9600);

pinMode(11,OUTPUT);

}

void **loop**() {

int reading = analogRead(A0);

int T= map(reading, 20, 358, -40, 125); // ánh xạ giá trị 20-358 -> -40-125

if(T>0){

digitalWrite(11, HIGH);

}

else {

digitalWrite(11, LOW);

}

delay(1000);

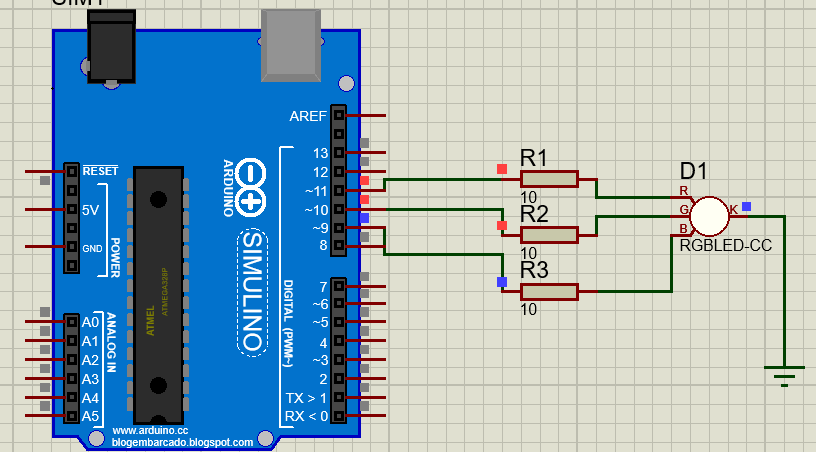
}

# Bài 4. Nháy Led RGB

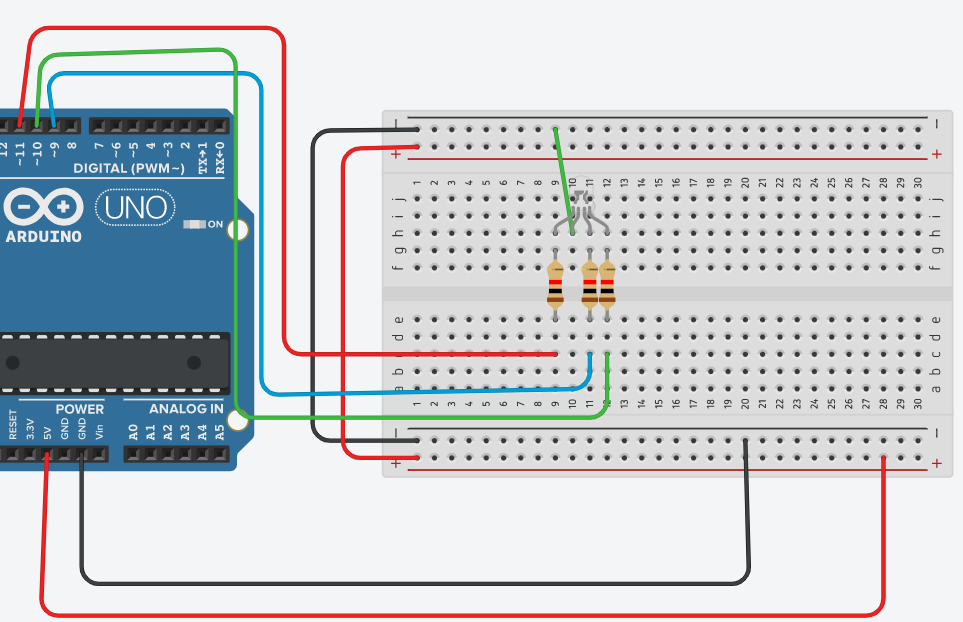
## Mô tả

Hệ thống được thiết kế điều khiển làm đèn RGB sáng ở màu đỏ, xanh lá cây, xanh dương, đèn được kết nối vào Arduino ở các cổng số 9, 10 ,11 (các chân PWM vì các cổng PWM cho phép bạn xuất ra xung PWM với độ phân giải 8bit, tức là tạo ra các giá trị từ 0 --> 28- l tương ứng với 0V --> 5V).

## Sơ đồ mạch điện



Sơ đồ thiết kế trên Proteus



Sơ đồ thiết kế trên TinkerCard

## Linh kiện

* 1 đèn led màu RGB: (Red, Green, Blue) giống như đèn LED thông thường, tuy nhiên, bên trong đèn thực sự có ba đèn LED, một màu đỏ, một màu xanh lá cây, một màu xanh lam. Bằng cách kiểm soát cường độ của từng đèn LED riêng lẻ, bạn có thể kết hợp khá nhiều màu sắc mà bạn muốn. Tổng số màu đèn RGB có thể sáng là 2563 màu.
* 3 điện trở R 10 Ω

## Code chương trình

void **setup**() {

pinMode(11, OUTPUT);

pinMode(10, OUTPUT);

pinMode(9, OUTPUT);

}

void **loop**() {

analogWrite(11,255);

analogWrite(10,24);

analogWrite(9,166);

delay(100);

analogWrite(11,144);

analogWrite(10,221);

analogWrite(9,12);

delay(100);

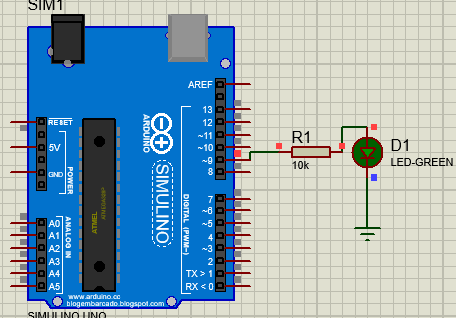
}

# Bài 5. Nháy Led sáng dần/ tắt dần

## Mô tả

Bài này sẽ thực hiện việc lập trình điều khiển led sáng dần/tắt dần liên tục. Đèn Led sẽ được nối ở cổng số 9 của mạch Arduino thông qua biến trở R.

## Sơ đồ mạch điện



Sơ đồ thiết kế trên Proteus

## Linh kiện

* 1 đèn led màu xanh
* 1 biến trỡ R

## Code chương trình

int index = 0;

void **setup**() {

pinMode(9, OUTPUT);

}

void **loop**() {

for(index = 0; index <= 255; index += 5){

analogWrite(9, index);

delay(50);

}

for(index=255; index>=0; index -= 5){

analogWrite(9, index);

delay(50);

}

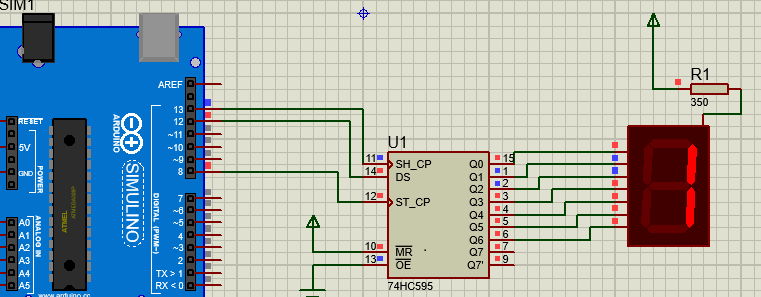
}

# Bài 6. Nháy Led 7 đoạn với IC74HC95

## Mô tả

Bài này thực hiện lập trình chương trình nháy led 7 đoạn thể hiện các số từ 0 đến 9 bằng đèn led thông qua IC74HC95

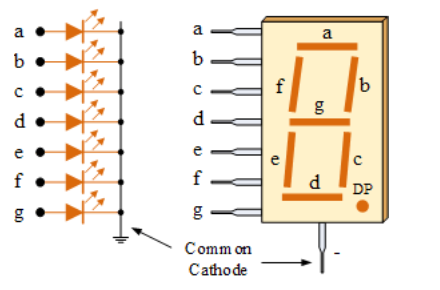
## Sơ đồ mạch



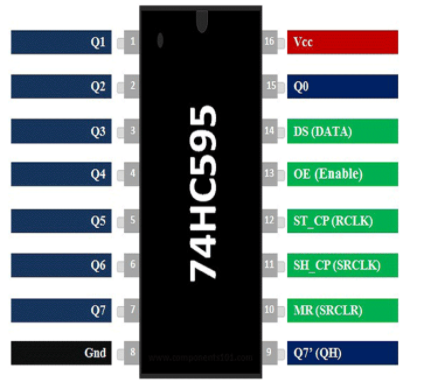
HÌnh 6 Sơ đồ thiết kế

## Linh kiện

* 1 led 7 đoạn: Mỗi đèn led 7 đoạn có chân đưa ra khỏi hộp hình vuông. Mỗi một chân sẽ được gán cho một chữ cái từ a đến g tương ứng với mỗi led. Những chân khác được nối lại với nhau thành một chân chung.



* 1 điện trở 350 Ω
* 1 thanh ghi dịch IC74HC595: 74HC595 là một thanh ghi dịch (shift register) hoạt động trên giao thức nối tiếp vào song song ra (Serial IN Parallel OUT). Nó nhận dữ liệu nối tiếp từ vi điều khiển và sau đó gửi dữ liệu này qua các chân song song. Có thể tăng 8 chân đầu ra bằng cách sử dụng chip đơn. Cũng có thể kết nối song song nhiều hơn 1 thanh ghi dịch. Giả sử đã kết nối ba thanh ghi dịch với bộ vi điều khiển, các chân đầu ra được tăng lên 8 x 3 = 24.



* Đặc tính thông số kỹ thuật 74HC595:
  + 8-bit, thanh ghi dịch nối tiếp vào song song ra.
  + Điện áp hoạt động: 2V đến 6V
  + Mức tiêu thụ điện: 80uA
  + Source ra / dòng sink: 35mA
  + Điện áp đầu ra bằng điện áp hoạt động
  + Điện áp đầu vào mức cao tối thiểu: 3,15V @ (Vcc = 4,5V)
  + Điện áp đầu vào mức thấp tối đa: 1.35V @ (Vcc = 4.5V)
  + Tần số đồng hồ tối đa: 25Mhz @ 4.5V
  + Có các gói PDIP, GDIP, PDSO 16 chân

## Code chương trình

int clockpin = 13;

int datapin = 12;

int latchpin = 8;

char LED7SEG[11] = {0xC0,0xF9,0xA4,0xB0,0x99,0x92,0x82,0xF8,0x80,0x90,0xff};

// tạo 1 mảng chứa các mã số hệ thập lục phân của các số từ 0-9

/\* Hàm: shiftOut(dataPin, clockPin, bitOrder, value)

dataPin: chân để xuất ra từng bit. Các kiểu dữ liệu được phép: int.

clockPin: chân để chuyển đổi khi dataPin đã được đặt thành giá trị chính xác. Các kiểu dữ liệu được phép: int.

bitOrder: thứ tự chuyển ra các bit; MSBFIRST hoặc LSBFIRST. (Bit quan trọng nhất đầu tiên hoặc bit ít quan trọng nhất đầu tiên).

value: dữ liệu cần chuyển ra ngoài. Các kiểu dữ liệu được phép: byte \*/

void setup() {

pinMode(clockpin, OUTPUT);

pinMode(datapin, OUTPUT);

pinMode(latchpin, OUTPUT);

}

void loop() {

for(int i=0; i<11; i++) {

digitalWrite(latchpin, LOW);

shiftOut(datapin, clockpin, MSBFIRST, LED7SEG[i]); //0

digitalWrite(latchpin, HIGH);

delay(100);

}

}

# Bài 7: Nháy Led 7 đoạn với RES16DIPIS

Mô tả: