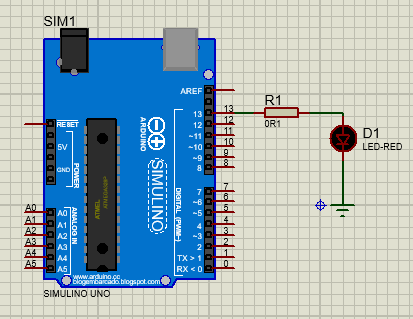
# BÀI 1: LED NHẤP NHÁY

## Mô tả:

* Thực hiện việc lập trình để làm sáng 1 đèn led
* Đèn led được kết nối vào chân số 13

## Sơ đồ thiết kế:



## Đặc điểm linh kiện:

* 1 Arduino UNO
* 1 Led red
* 1 điện trở(R1=100ohm)

## Mã lệnh chính:

*void setup()*

*{*

*pinMode(13, OUTPUT);*

*}*

*void loop()*

*{*

*digitalWrite(13, HIGH);*

*delay(1000); // Wait for 1000 millisecond(s)*

*digitalWrite(13, LOW);*

*delay(1000); // Wait for 1000 millisecond(s)*

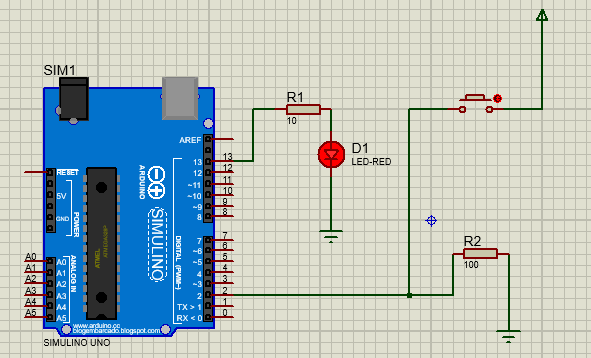
*}*

# BÀI 2: LED BẬT TẮT

## Mô tả:

* Thực hiện việc lập trình điều khiển 1 đèn led bật/tắt trong khoảng thời gian
* Đèn led được kết nối vào chân số 13 để điều khiển

## Sơ đồ thiết kế:



## Đặc điểm linh kiện:

* 1 Arduino UNO
* 1 Led red
* 2 điện trở(R1=R2=100ohm)
* 1 button

## Mã lệnh chính:

*int x = 0;*

*void setup() {*

*// put your setup code here, to run once:*

*pinMode(2, INPUT);*

*pinMode(13, OUTPUT);*

*}*

*void loop() {*

*// put your main code here, to run repeatedly:*

*x = digitalRead(2);*

*if (x== HIGH) {*

*digitalWrite(13, HIGH);*

*} else {*

*digitalWrite(13, LOW);*

*}*

*delay(1000);*

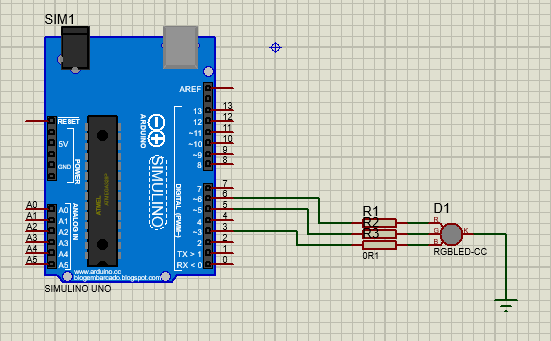
*}*

# BÀI 3: LED RGB

## Mô tả:

* Lập trình thực hiện nháy led RGB
* Chân R, G, B được nối vào lần lượt các điện trở R1=R2=R3=100ohm và điện trợ được nối vào chân 6, 5, 3 của ARDUINO UNO

## Sơ đồ thiết kế:



## Đặc điểm linh kiện:

- 1 Led RGB

- 3 Điện trở

- 1 arduino uno

## Mã lệnh chính:

*const int R = 3; const int G = 5; const int B = 6;*

*int Red = 255; int Green = 0; int Blue = 0;*

*void setup(){*

*pinMode(R, OUTPUT); pinMode(G, OUTPUT); pinMode(B, OUTPUT);*

*analogWrite(R, Red); analogWrite(G, Green); analogWrite(B, Blue);}*

*void loop() {*

*for (Blue = 0; Blue<255; Blue = Blue + 5) {*

*analogWrite(B, Blue);*

*delay(10);}*

*for (Red = 255; Red>0; Red = Red - 5) {*

*analogWrite(R, Red);*

*delay(10); }*

*for (Green = 0; Green<255; Green = Green + 5) {*

*analogWrite(G, Green);*

*delay(10); }*

*for (Blue = 255; Blue>0; Blue = Blue - 5) {*

*analogWrite(B, Blue);*

*delay(10);}*

*for (Red = 0; Red<255; Red = Red + 5) {*

*analogWrite(R, Red);*

*delay(10);}*

*for (Green = 255; Green>0; Green = Green - 5) {*

*analogWrite(G, Green);*

*delay(10);}*

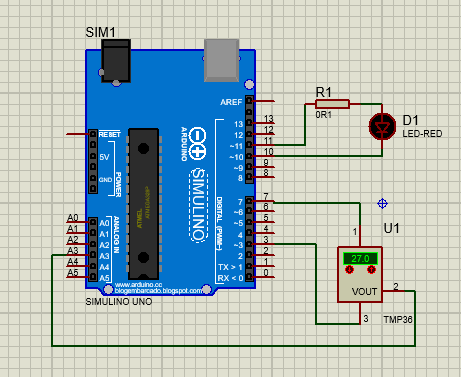
*}*

# BÀI 4: TMP36-LED

## Mô tả:

* Thực hiện điều khiển led qua TMP36
* Khi nhiệt độ thay đổi thì đèn led được sáng lên

## Sơ đồ thiết kế:



## Đặc điểm linh kiện:

- 1 Led red

- 1 Điện trở r1

- 1 arduino uno

- 1 TMP36

## Mã lệnh chính:

*void setup(){*

*Serial.begin(9600);*

*pinMode(7,1); pinMode(3,1); pinMode(11,1);*

*digitalWrite(7,1); digitalWrite(3,0);*

*digitalWrite(10,1); digitalWrite(11,0);*

*}*

*void loop(){*

*Serial.println(analogRead(A3));*

*if(analogRead(A3)>181){digitalWrite(10,0);};*

*if(analogRead(A3)<181){digitalWrite(10,1);};*

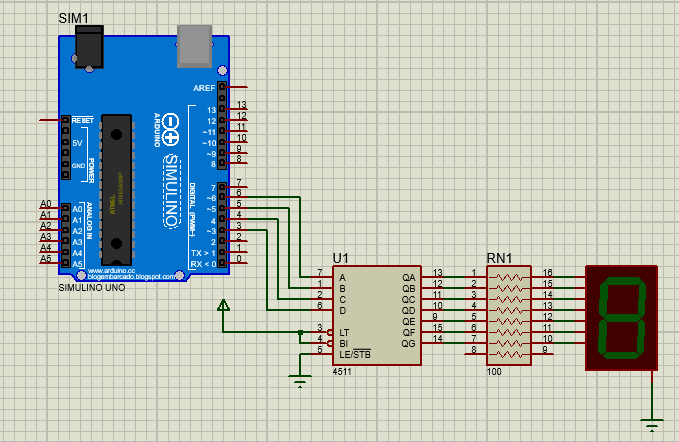
*}*

# BÀI 5: LED 7 ĐOẠN

## Mô tả:

* Thực hiện trên led 7 đoạn để led hiển thị số từ 0 -> 9
* Thông qua 4511 đọc số BCD thay thế tương đương 1 ký số ở hệ 10 bằng tổ hợp 4 bít tương ứng
* Và 1 điện trở lớn gồm 8 điện trở con bên trong

## Sơ đồ thiết kế:



## Đặc điểm linh kiện:

- 1 Led 7 đoạn

- 1 Điện trở RES16DIPIS

- 1 ARDUINO UNO

- 4511

## Mã lệnh chính:

*int a=6, b=5, c=4, d=3;*

*void setup() {*

*pinMode(a, OUTPUT); pinMode(b, OUTPUT);*

*pinMode(c, OUTPUT); pinMode(d, OUTPUT);*

*}*

*void khong(){*

*digitalWrite(a, LOW);*

*digitalWrite(b, LOW);*

*digitalWrite(c, LOW);*

*digitalWrite(d, LOW);*

*}*

*void mot(){*

*digitalWrite(a, HIGH);*

*digitalWrite(b, LOW);*

*digitalWrite(c, LOW);*

*digitalWrite(d, LOW);*

*}*

*void hai(){*

*digitalWrite(a, LOW);*

*digitalWrite(b, HIGH);*

*digitalWrite(c, LOW);*

*digitalWrite(d, LOW);*

*}*

*void ba(){*

*digitalWrite(a, HIGH);*

*digitalWrite(b, HIGH);*

*digitalWrite(c, LOW);*

*digitalWrite(d, LOW);*

*}*

*void bon(){*

*digitalWrite(a, LOW);*

*digitalWrite(b, LOW);*

*digitalWrite(c, HIGH);*

*digitalWrite(d, LOW);*

*}*

*void nam(){*

*digitalWrite(a, HIGH);*

*digitalWrite(b, LOW);*

*digitalWrite(c, HIGH);*

*digitalWrite(d, LOW);*

*}*

*void sau(){*

*digitalWrite(a, LOW);*

*digitalWrite(b, HIGH);*

*digitalWrite(c, HIGH);*

*digitalWrite(d, LOW);*

*}*

*void bay(){*

*digitalWrite(a, HIGH);*

*digitalWrite(b, HIGH);*

*digitalWrite(c, HIGH);*

*digitalWrite(d, LOW);*

*}*

*void tam(){*

*digitalWrite(a, LOW);*

*digitalWrite(b, LOW);*

*digitalWrite(c, LOW);*

*digitalWrite(d, HIGH);*

*}*

*void chin(){*

*digitalWrite(a, HIGH);*

*digitalWrite(b, LOW);*

*digitalWrite(c, LOW);*

*digitalWrite(d, HIGH);*

*}*

*void loop() {*

*// put your main code here, to run repeatedly:*

*khong(); delay(1000);*

*mot(); delay(1000);*

*hai(); delay(1000);*

*ba(); delay(1000);*

*bon(); delay(1000);*

*nam(); delay(1000);*

*sau(); delay(1000);*

*bay(); delay(1000);*

*tam(); delay(1000);*

*chin(); delay(1000);*

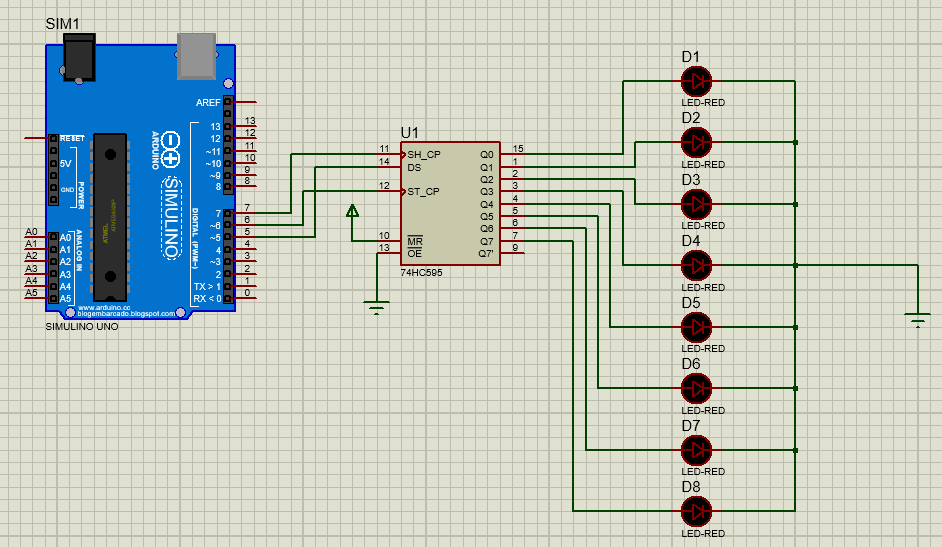
*}*

# BÀI 6: 8 LED NHẤP NHÁY

## Mô tả:

* Thực hiện điều khiển 8 led sáng tắt theo hệ số nhị phân 10

## Sơ đồ thiết kế:

****

## Đặc điểm linh kiện:

- 8 Led red

- 1 IC 74HC 595

- 1 ARDUINO UNO

## Mã lệnh chính:

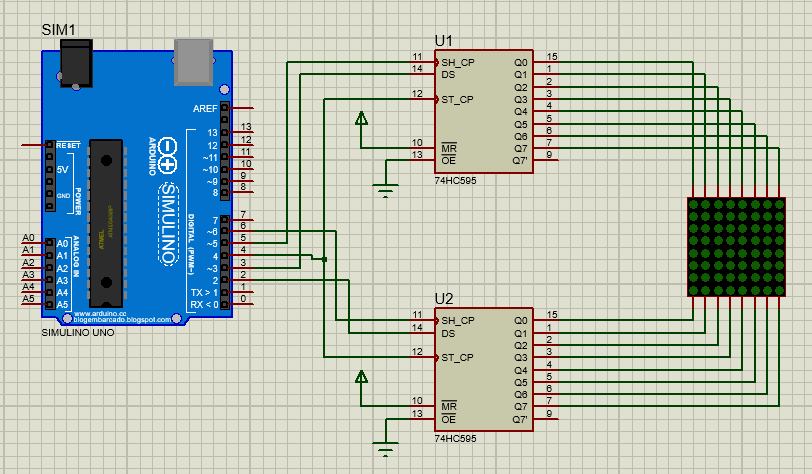
|  |
| --- |
| *#define \_clock 7* |
| *#define \_latch 6* |
| *#define \_data 5* |
|  |
| *void setup() {* |
| *// put your setup code here, to run once:* |
| *pinMode(\_latch, OUTPUT);* |
| *pinMode(\_clock, OUTPUT);* |
| *pinMode(\_data, OUTPUT);* |
| *}* |
|  |
| *void loop() {* |
| *// put your main code here, to run repeatedly:* |
| *for (int i=0;i<256;i++){* |
| *digitalWrite(\_latch, LOW);* |
| *shiftOut(\_data, \_clock, LSBFIRST, i);* |
| *digitalWrite(\_latch, HIGH);* |
| *delay(500);}* |
| *}* |

# BÀI 7: LED MA TRẬN

## Mô tả:

* Thực hiện việc lập trình làm sáng đèn led ma trận 8x8 theo hình chữ A
* Led được gắn theo hàng vào cột vào 2 IC 74HC595 để điều khiển

## Sơ đồ thiết kế:



## Đặc điểm linh kiện:

- 1 Led ma trận 8x8

- 2 IC 74HC 595

- 1 ARDUINO UNO

## Mã lệnh chính:

*#define \_data\_cot 2*

*#define \_data\_hang 3*

*#define \_clock\_hang 5*

*#define \_clock\_cot 6*

*#define \_latch 4*

*byte chu[][8] = {*

*{0xFF,0xC0,0x80,0xB7,0xB7,0x80,0xC0,0xFF}, //A};*

*byte cot = 0b10000000;*

*void setup() {*

*Serial.begin(9600);*

*pinMode(\_latch,OUTPUT);//RCLK*

*pinMode(\_data\_hang,OUTPUT);//SER hang*

*pinMode(\_clock\_hang,OUTPUT);//SRCLK hang*

*pinMode(\_clock\_cot,OUTPUT);//SRCLK cot*

*pinMode(\_data\_cot,OUTPUT);//SER cot*

*}*

*void loop() {*

*for(int i = 0;i<10;i++){*

*digitalWrite(\_latch,LOW);*

*shiftOut(\_data\_hang,\_clock\_hang,LSBFIRST,chu[0][i]);// hang 8 - hang 1*

*shiftOut(\_data\_cot,\_clock\_cot,LSBFIRST,cot >> i);// cot 8 - cot 1*

*digitalWrite(\_latch,HIGH);*

*delay(1);}*

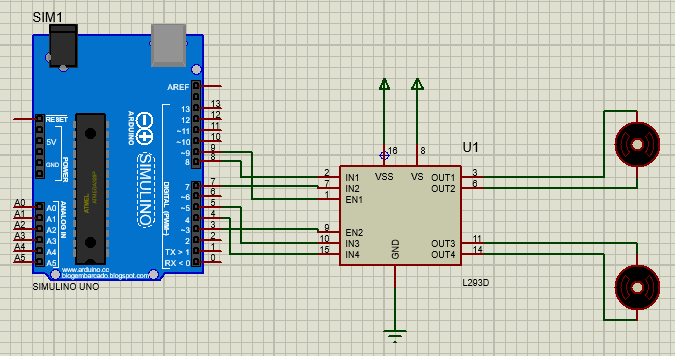
*}*

# BÀI 8: ĐIỀU KHIỂN MOTOR BẰNG IC 1293D

## Mô tả:

* Thực hiện việc lập trình để chạy 2 motor
* Chạy theo trình tự bắt đầu quay, đổi chiều quay, tăng tốc, giảm tốc,
* 2 Motor được kết nối vào IC L293D để điều khiển

## Sơ đồ thiết kế:



## Đặc điểm linh kiện:

* 1 Arduino UNO
* 1 IC L293D
* 2 Motor

## Mã lệnh chính:

// Kết nối động cơ A

int enA = 9; int in1 = 8; int in2 = 7;

// Kết nối động cơ B

int enB = 3; int in3 = 5; int in4 = 4;

void setup(){

// Set tất cả các chân điều khiển động cơ thành output

pinMode(enA, OUTPUT);

pinMode(enB, OUTPUT);

pinMode(in1, OUTPUT);

pinMode(in2, OUTPUT);

pinMode(in3, OUTPUT);

pinMode(in4, OUTPUT);

// Tắt tất cả động cơ – Trạng thái ban đầu

digitalWrite(in1, LOW);

digitalWrite(in2, LOW);

digitalWrite(in3, LOW);

digitalWrite(in4, LOW);

}

void loop(){

directionControl();

delay(1000);

speedControl();

delay(1000);

}

// Đoạn chương trình này cho phép bạn điều khiển chiều quay của động cơ

void directionControl(){

// Cho động cơ quay với tốc độ tối đa

// Giá trị của PWM thay đổi từ 0 đến 255

analogWrite(enA, 255);

analogWrite(enB, 255);

// Tắt động cơ A và B

digitalWrite(in1, HIGH);

digitalWrite(in2, LOW);

digitalWrite(in3, HIGH);

digitalWrite(in4, LOW);

delay(2000);

// Thay đổi chiều quay của động cơ

digitalWrite(in1, LOW);

digitalWrite(in2, HIGH);

digitalWrite(in3, LOW);

digitalWrite(in4, HIGH);

delay(2000);

// Tắt tất cả các động cơ

digitalWrite(in1, LOW);

digitalWrite(in2, LOW);

digitalWrite(in3, LOW);

digitalWrite(in4, LOW);

}

// Đoạn chương trình điều khiển tốc độ của động cơ

void speedControl(){

// Khởi động các động cơ

digitalWrite(in1, LOW);

digitalWrite(in2, HIGH);

digitalWrite(in3, LOW);

digitalWrite(in4, HIGH);

// Tăng tốc từ 0 đến tốc độ tối đa

for (int i = 0; i < 256; i++)

{

analogWrite(enA, i);

analogWrite(enB, i);

delay(20);

}

// Giảm tốc từ tốc độ tối đa về 0

for (int i = 255; i >= 0; i--)

{

analogWrite(enA, i);

analogWrite(enB, i);

delay(20);

}

// Tắt tất cả các động cơ

digitalWrite(in1, LOW);

digitalWrite(in2, LOW);

digitalWrite(in3, LOW);

digitalWrite(in4, LOW);

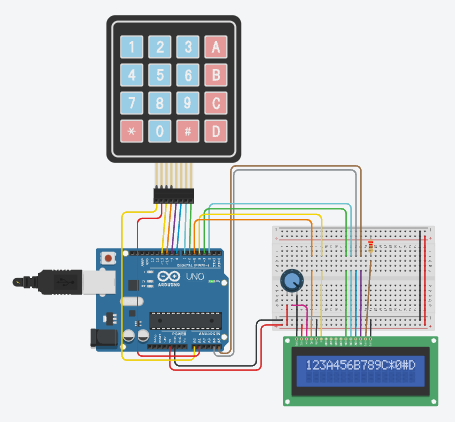
}

# BÀI 9: KEYPAD LCD

## Mô tả:

* Thực hiện việc lập trình để điều khiển keypad
* Hiển thị nội dung mà keypad được người dung thực hiện ra màn hình LCD

## Sơ đồ thiết kế:



## Đặc điểm linh kiện:

* 1 Arduino UNO
* 1 LCD 16x2
* 250 kΩ Potentiometer
* Keypad 4x4
* 220 Ω Resistor

## Mã lệnh chính:

*#include <Keypad.h>*

*#include <LiquidCrystal.h>*

*LiquidCrystal lcd(5, 4, 3, 2, A4, A5);*

*const byte ROWS = 4;*

*const byte COLS = 4;*

*char keys[ROWS][COLS] = {*

*{'1','2','3','A'},*

*{'4','5','6','B'},*

*{'7','8','9','C'},*

*{'\*','0','#','D'}*

*};*

*byte rowPins[ROWS] = {A0, A1, 11, 10};*

*byte colPins[COLS] = {9, 8, 7, 6};*

*int LCDRow = 0;*

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

*void setup(){*

*Serial.begin(9600);*

*lcd.begin(16, 2);*

*lcd.setCursor(LCDRow, 0);*

*}*

*void loop(){*

*char key = keypad.getKey();*

*if (key){*

*Serial.println(key);*

*lcd.print(key);*

*lcd.setCursor (++LCDRow, 0);*

*}*

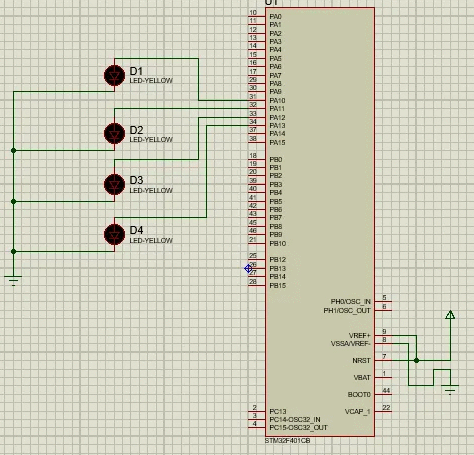
*}*

# BÀI 10: STM32-LED

## Mô tả:

* Thực hiện việc lập trình để làm sáng đèn led
* Đèn led được kết nối vào STM32 để điều khiển

## Sơ đồ thiết kế:



## Đặc điểm linh kiện:

* Led red, Led yellow, Led green
* STM32F401VE
* 3 điện trở

## Mã lệnh chính:

*/\* USER CODE BEGIN Header \*/ /\*\** ***\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\****

*\* @file : main.c*

*\* @brief : Main program body* ***\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\****

*\* @attention*

*\**

*\* <h2><center>&copy; Copyright (c) 2021 STMicroelectronics.*

*\* All rights reserved.</center></h2>*

*\**

*\* This software component is licensed by ST under BSD 3-Clause license,*

*\* the "License"; You may not use this file except in compliance with the*

*\* License. You may obtain a copy of the License at:*

*\** [*opensource.org/licenses/BSD-3-Clause*](https://l.facebook.com/l.php?u=http%3A%2F%2Fopensource.org%2Flicenses%2FBSD-3-Clause%3Ffbclid%3DIwAR2smt9NB3R-xF_64xZBddETjW2G2YgIIlxAJVQjrz3yUN3hPzYXESCK1Zk&h=AT1zd4F0G2GDUw4KWaUso6IJOl_gVNwJBvTiB6Sjih2GjETgS8HhkGp6h9o9YlMOFXdmAea9ECYPCRB5VitEjNzAq-taZOGXfXdWpK9QHjihSNyqFZ8nshe1ldYYFZeruKyKCA) *\** ***\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**** *\*/*

*/\* USER CODE END Header \*/*

*/\* Includes ------------------------------------------------------------------\*/*

*#include "main.h"*

*/\* Private includes ----------------------------------------------------------\*/*

*/\* USER CODE BEGIN Includes \*/*

*/\* USER CODE END Includes \*/*

*/\* Private typedef -----------------------------------------------------------\*/*

*/\* USER CODE BEGIN PTD \*/*

*/\* USER CODE END PTD \*/*

*/\* Private define ------------------------------------------------------------\*/*

*/\* USER CODE BEGIN PD \*/*

*/\* USER CODE END PD \*/*

*/\* Private macro -------------------------------------------------------------\*/*

*/\* USER CODE BEGIN PM \*/*

*/\* USER CODE END PM \*/*

*/\* Private variables ---------------------------------------------------------\*/*

*/\* USER CODE BEGIN PV \*/*

*/\* USER CODE END PV \*/*

*/\* Private function prototypes -----------------------------------------------\*/*

*void SystemClock\_Config(void);*

*static void MX\_GPIO\_Init(void);*

*/\* USER CODE BEGIN PFP \*/*

*/\* USER CODE END PFP \*/*

*/\* Private user code ---------------------------------------------------------\*/*

*/\* USER CODE BEGIN 0 \*/*

*/\* USER CODE END 0 \*/*

*/\*\**

*\* @brief The application entry point.*

*\* @retval int*

*\*/ int main(void)*

*{*

*/\* USER CODE BEGIN 1 \*/*

*/\* USER CODE END 1 \*/*

*/\* MCU Configuration--------------------------------------------------------\*/*

*/\* Reset of all peripherals, Initializes the Flash interface and the Systick. \*/*

*HAL\_Init();*

*/\* USER CODE BEGIN Init \*/*

*/\* USER CODE END Init \*/*

*/\* Configure the system clock \*/*

*SystemClock\_Config();*

*/\* USER CODE BEGIN SysInit \*/*

*/\* USER CODE END SysInit \*/*

*/\* Initialize all configured peripherals \*/*

*MX\_GPIO\_Init();*

*/\* USER CODE BEGIN 2 \*/*

*/\* USER CODE END 2 \*/*

*/\* Infinite loop \*/*

*/\* USER CODE BEGIN WHILE \*/*

*while (1) {*

*/\* USER CODE END WHILE \*/*

*/\* USER CODE BEGIN 3 \*/*

*HAL\_GPIO\_TogglePin(GPIOA, GPIO\_PIN\_10| GPIO\_PIN\_11 | GPIO\_PIN\_12| GPIO\_PIN\_13);*

*HAL\_Delay(1000); }*

*/\* USER CODE END 3 \*/ }*

*/\*\**

*\* @brief System Clock Configuration*

*\* @retval None*

*\*/*

*void SystemClock\_Config(void) {*

*RCC\_OscInitTypeDef RCC\_OscInitStruct = {0};*

*RCC\_ClkInitTypeDef RCC\_ClkInitStruct = {0};*

*/\*\* Configure the main internal regulator output voltage \*/ \_\_HAL\_RCC\_PWR\_CLK\_ENABLE(); \_\_HAL\_PWR\_VOLTAGESCALING\_CONFIG(PWR\_REGULATOR\_VOLTAGE\_SCALE2);*

*/\*\* Initializes the RCC Oscillators according to the specified parameters*

*\* in the RCC\_OscInitTypeDef structure. \*/*

*RCC\_OscInitStruct.OscillatorType = RCC\_OSCILLATORTYPE\_HSI; RCC\_OscInitStruct.HSIState = RCC\_HSI\_ON;*

*RCC\_OscInitStruct.HSICalibrationValue = RCC\_HSICALIBRATION\_DEFAULT; RCC\_OscInitStruct.PLL.PLLState = RCC\_PLL\_NONE;*

*if (HAL\_RCC\_OscConfig(&RCC\_OscInitStruct) != HAL\_OK)*

*{ Error\_Handler(); }*

*/\*\* Initializes the CPU, AHB and APB buses clocks \*/*

*RCC\_ClkInitStruct.ClockType = RCC\_CLOCKTYPE\_HCLK|RCC\_CLOCKTYPE\_SYSCLK |RCC\_CLOCKTYPE\_PCLK1|RCC\_CLOCKTYPE\_PCLK2;*

*RCC\_ClkInitStruct.SYSCLKSource = RCC\_SYSCLKSOURCE\_HSI; RCC\_ClkInitStruct.AHBCLKDivider = RCC\_SYSCLK\_DIV2; RCC\_ClkInitStruct.APB1CLKDivider = RCC\_HCLK\_DIV1; RCC\_ClkInitStruct.APB2CLKDivider = RCC\_HCLK\_DIV1;*

*if (HAL\_RCC\_ClockConfig(&RCC\_ClkInitStruct, FLASH\_LATENCY\_0) != HAL\_OK)*

*{ Error\_Handler(); } }*

*/\*\**

*\* @brief GPIO Initialization Function*

*\* @param None*

*\* @retval None \*/*

*static void MX\_GPIO\_Init(void) {*

*GPIO\_InitTypeDef GPIO\_InitStruct = {0};*

*/\* GPIO Ports Clock Enable \*/*

*\_\_HAL\_RCC\_GPIOH\_CLK\_ENABLE();*

*\_\_HAL\_RCC\_GPIOA\_CLK\_ENABLE();*

*/\*Configure GPIO pin Output Level \*/*

*HAL\_GPIO\_WritePin(GPIOA, LED\_OUT\_4\_Pin|LED\_OUT\_3\_Pin|LED\_OUT\_2\_Pin|LED\_OUT\_1\_Pin, GPIO\_PIN\_RESET);*

*/\*Configure GPIO pins : LED\_OUT\_4\_Pin LED\_OUT\_3\_Pin LED\_OUT\_2\_Pin LED\_OUT\_1\_Pin*

*\*/ GPIO\_InitStruct.Pin = LED\_OUT\_4\_Pin|LED\_OUT\_3\_Pin|LED\_OUT\_2\_Pin|LED\_OUT\_1\_Pin; GPIO\_InitStruct.Mode = GPIO\_MODE\_OUTPUT\_PP;*

*GPIO\_InitStruct.Pull = GPIO\_NOPULL; GPIO\_InitStruct.Speed = GPIO\_SPEED\_FREQ\_LOW;*

*HAL\_GPIO\_Init(GPIOA, &GPIO\_InitStruct); }*

*/\* USER CODE BEGIN 4 \*/*

*/\* USER CODE END 4 \*/*

*/\*\**

*\* @brief This function is executed in case of error occurrence.*

*\* @retval None \*/*

*void Error\_Handler(void) {*

*/\* USER CODE BEGIN*

*Error\_Handler\_Debug \*/*

*/\* User can add his own implementation to report the HAL error return state \*/*

*/\* USER CODE END Error\_Handler\_Debug \*/ }*

*#ifdef USE\_FULL\_ASSERT*

*/\*\**

*\* @brief Reports the name of the source file and the source line number*

*\* where the assert\_param error has occurred.*

*\* @param file: pointer to the source file name*

*\* @param line: assert\_param error line source number*

*\* @retval None*

*\*/*

*void assert\_failed(uint8\_t \*file, uint32\_t line) {*

*/\* USER CODE BEGIN 6 \*/*

*/\* User can add his own implementation to report the file name and line number,*

*tex: printf("Wrong parameters value: file %s on line %d\r\n", file, line) \*/*

*/\* USER CODE END 6 \*/ }*

*#endif /\* USE\_FULL\_ASSERT \*/*

*/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* (C) COPYRIGHT STMicroelectronics*

*\*\*\*\*\*END OF FILE\*\*\*\*/*

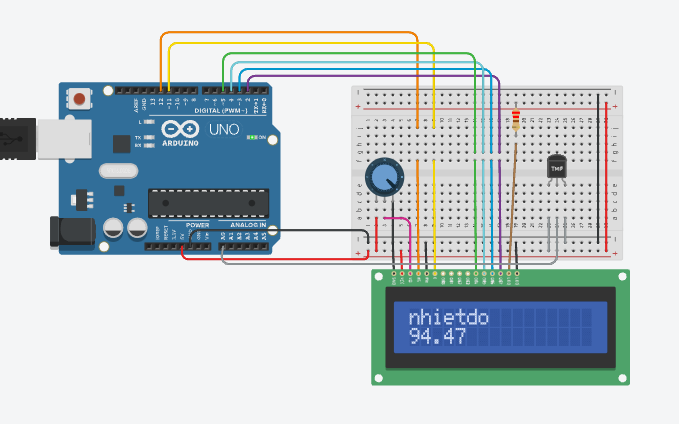
# **BÀI KIỂM TRA**

# BÀI 1A:ĐỌC NHIỆT ĐỘ MÔI TRƯỜNG VÀ IN RA LCD

## Mô tả:

* Thực hiện việc lập trình để đọc nhiệt độ của tmp36, và hiển thị nhiệt độ ra màn hình LCD
* Khi nhiệt dộ thay đổi thì liên tục đọc và hiển thị

## Sơ đồ thiết kế:



## Đặc điểm linh kiện:

* 1 Arduino UNO
* 1 LCD 16x2
* 1 Potentiometer 250kohm
* 1 Resistor 220ohm

## Mã lệnh chính:

*#include <LiquidCrystal.h>*

*#define SENSOR\_PIN A0*

*float voltage = 0;*

*float sensor = 0;*

*float celsius = 0;*

*float fahrenheit = 0;*

*// initialize the library with the numbers of the interface pins*

*LiquidCrystal lcd(12, 11, 5, 4, 3, 2);*

*void setup() {*

*// set up the LCD's number of columns and rows:*

*lcd.begin(16, 2);*

*// Print a message to the LCD.*

*Serial.begin(9600);*

*}*

*void loop() {*

*// set the cursor to column 0, line 1*

*// (note: line 1 is the second row, since counting begins with 0):*

*lcd.setCursor(0, 0);*

*// print the number of seconds since reset:*

*lcd.print("nhietdo");*

*sensor = analogRead(SENSOR\_PIN);*

*voltage = (sensor\*5000)/1024; // Chuyển đổi tín hiệu cảm biến sang mili Volt*

*voltage = voltage-400; // Trừ đi điện áp bù*

*celsius = voltage/10; // Chuyển đổi mV sang độ C*

*fahrenheit = ((celsius \* 1.8)+32);*

*lcd.setCursor(0, 1);*

*lcd.print(fahrenheit,2);*

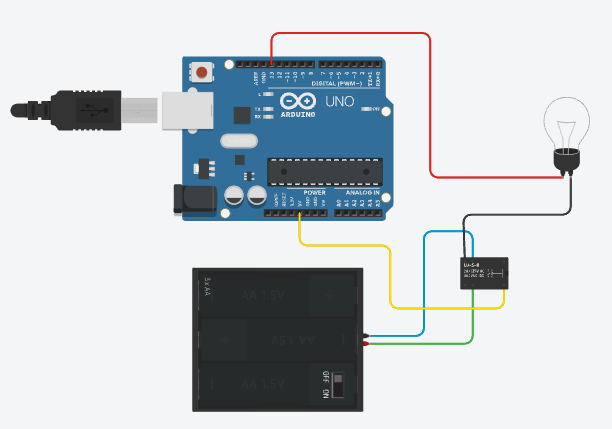
*}*

# BÀI 1B: BẬT/NGẮT ĐÈN ĐIỆN

## Mô tả:

* Thực hiện việc lập trình để làm sáng 1 đèn điện
* Điều khiển đèn điện thông qua hệ thống bật/ngắt và module Relay

## Sơ đồ thiết kế:



## Đặc điểm linh kiện:

* 1 Arduino UNO
* 1 Led bulb
* 1 Realy SPDT
* 3 batteries, AA,  yes 1.5V Battery

## Mã lệnh chính:

*// C++ code*

*//*

*void setup()*

*{*

*pinMode(13, OUTPUT);*

*}*

*void loop()*

*{*

*digitalWrite(LED\_BUILTIN, HIGH);*

*delay(1000); // Wait for 1000 millisecond(s)*

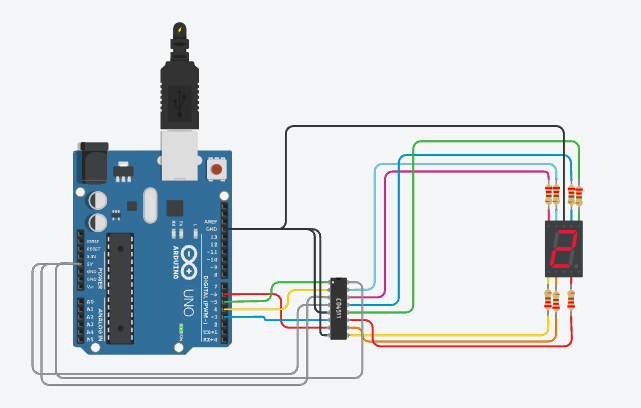
*}*

# BÀI 2A: LED 7 ĐOẠN ĐẾM SỐ TỪ 0🡪9

## Mô tả:

* Thực hiện việc lập trình để làm chạy led 7 đoạn từ 0🡪9
* Việc điều khiên thông qua IC 4511

## Sơ đồ thiết kế:



## Đặc điểm linh kiện:

* 1 Arduino UNO
* 7 Segment Display
* 220 Ω Resistor
* 1 IC 4511

## Mã lệnh chính:

*int a=6, b=5, c=4, d=3;*

*void setup() {*

*pinMode(a, OUTPUT);*

*pinMode(b, OUTPUT);*

*pinMode(c, OUTPUT);*

*pinMode(d, OUTPUT);*

*}*

*void khong(){*

*digitalWrite(a, LOW);*

*digitalWrite(b, LOW);*

*digitalWrite(c, LOW);*

*digitalWrite(d, LOW);*

*}*

*void mot(){*

*digitalWrite(a, HIGH);*

*digitalWrite(b, LOW);*

*digitalWrite(c, LOW);*

*digitalWrite(d, LOW);*

*}*

*void hai(){*

*digitalWrite(a, LOW);*

*digitalWrite(b, HIGH);*

*digitalWrite(c, LOW);*

*digitalWrite(d, LOW);*

*}*

*void ba(){*

*digitalWrite(a, HIGH);*

*digitalWrite(b, HIGH);*

*digitalWrite(c, LOW);*

*digitalWrite(d, LOW);*

*}*

*void bon(){*

*digitalWrite(a, LOW);*

*digitalWrite(b, LOW);*

*digitalWrite(c, HIGH);*

*digitalWrite(d, LOW);*

*}*

*void nam(){*

*digitalWrite(a, HIGH);*

*digitalWrite(b, LOW);*

*digitalWrite(c, HIGH);*

*digitalWrite(d, LOW);*

*}*

*void sau(){*

*digitalWrite(a, LOW);*

*digitalWrite(b, HIGH);*

*digitalWrite(c, HIGH);*

*digitalWrite(d, LOW);*

*}*

*void bay(){*

*digitalWrite(a, HIGH);*

*digitalWrite(b, HIGH);*

*digitalWrite(c, HIGH);*

*digitalWrite(d, LOW);*

*}*

*void tam(){*

*digitalWrite(a, LOW);*

*digitalWrite(b, LOW);*

*digitalWrite(c, LOW);*

*digitalWrite(d, HIGH);*

*}*

*void chin(){*

*digitalWrite(a, HIGH);*

*digitalWrite(b, LOW);*

*digitalWrite(c, LOW);*

*digitalWrite(d, HIGH);*

*}*

*void loop() {*

*// put your main code here, to run repeatedly:*

*khong(); delay(1000);*

*mot(); delay(1000);*

*hai(); delay(1000);*

*ba(); delay(1000);*

*bon(); delay(1000);*

*nam(); delay(1000);*

*sau(); delay(1000);*

*bay(); delay(1000);*

*tam(); delay(1000);*

*chin(); delay(1000);*

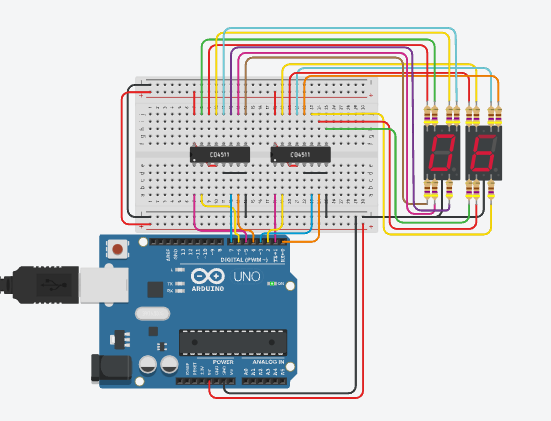
*}*

# BÀI 2B: LED 7 ĐOẠN ĐẾM SỐ TỪ 0🡪99

## Mô tả:

* Thực hiện việc lập trình để làm chạy led 7 đoạn từ 0🡪99
* Việc điều khiển thông qua 2 IC 4511

## Sơ đồ thiết kế:



## Đặc điểm linh kiện:

* 1 Arduino UNO
* Cathode 7 Segment Display
* 0.47 kΩ Resistor
* 2 IC 4511

## Mã lệnh chính:

*void setup(){*

*pinMode(0, OUTPUT);*

*pinMode(1, OUTPUT);*

*pinMode(2, OUTPUT);*

*pinMode(3, OUTPUT);*

*pinMode(4, OUTPUT);*

*pinMode(5, OUTPUT);*

*pinMode(6, OUTPUT);*

*pinMode(7, OUTPUT);*

*}*

*void show\_so(int so, int led = 0)*

*{*

*if (so == 0)*

*{*

*digitalWrite(0 + led, LOW);*

*digitalWrite(1 + led, LOW);*

*digitalWrite(2 + led, LOW);*

*digitalWrite(3 + led, LOW);*

*}*

*else if (so == 1)*

*{*

*digitalWrite(0 + led, HIGH);*

*digitalWrite(1 + led, LOW);*

*digitalWrite(2 + led, LOW);*

*digitalWrite(3 + led, LOW);*

*}*

*else if (so == 2)*

*{*

*digitalWrite(0 + led, LOW);*

*digitalWrite(1 + led, HIGH);*

*digitalWrite(2 + led, LOW);*

*digitalWrite(3 + led, LOW);*

*}*

*else if (so == 3)*

*{*

*digitalWrite(0 + led, HIGH);*

*digitalWrite(1 + led, HIGH);*

*digitalWrite(2 + led, LOW);*

*digitalWrite(3 + led, LOW);*

*}*

*else if (so == 4)*

*{*

*digitalWrite(0 + led, LOW);*

*digitalWrite(1 + led, LOW);*

*digitalWrite(2 + led, HIGH);*

*digitalWrite(3 + led, LOW);*

*}*

*else if (so == 5)*

*{*

*digitalWrite(0 + led, HIGH);*

*digitalWrite(1 + led, LOW);*

*digitalWrite(2 + led, HIGH);*

*digitalWrite(3 + led, LOW);*

*}*

*else if (so == 6)*

*{*

*digitalWrite(0 + led, LOW);*

*digitalWrite(1 + led, HIGH);*

*digitalWrite(2 + led, HIGH);*

*digitalWrite(3 + led, LOW);*

*}*

*else if (so == 7)*

*{*

*digitalWrite(0 + led, HIGH);*

*digitalWrite(1 + led, HIGH);*

*digitalWrite(2 + led, HIGH);*

*digitalWrite(3 + led, LOW);*

*}*

*else if (so == 8)*

*{*

*digitalWrite(0 + led, LOW);*

*digitalWrite(1 + led, LOW);*

*digitalWrite(2 + led, LOW);*

*digitalWrite(3 + led, HIGH);*

*}*

*else if (so == 9)*

*{*

*digitalWrite(0 + led, HIGH);*

*digitalWrite(1 + led, LOW);*

*digitalWrite(2 + led, LOW);*

*digitalWrite(3 + led, HIGH);*

*}*

*}*

*void loop()*

*{*

*static int num1 = 0;*

*static int num2 = 0;*

*show\_so(num1 % 10, 4);*

*show\_so(num2 % 10);*

*num2++;*

*if (num2 % 10 == 0)*

*{*

*num2 = 0;*

*num1++;*

*}*

*delay(1000);*

*}*