**TRƯỜNG ĐẠI HỌC NHA TRANG**

**KHOA CÔNG NGHỆ THÔNG TIN**

\*\*\*\*\*\*o0o\*\*\*\*\*\*



**BÁO CÁO LẬP TRÌNH NHÚNG**

**Giảng viên học phần: Mai Cường Thọ**

**Sinh viên thực hiện : Đặng Phương Nam**

**Mssv : 61133120**

# **Bài TMP36 đo led**

## 1.Mô tả:

Bài này quan sát nhiệt độ thay đổi trên TMP khi đèn led sáng .

## 2.Sơ đồ thiết kế:

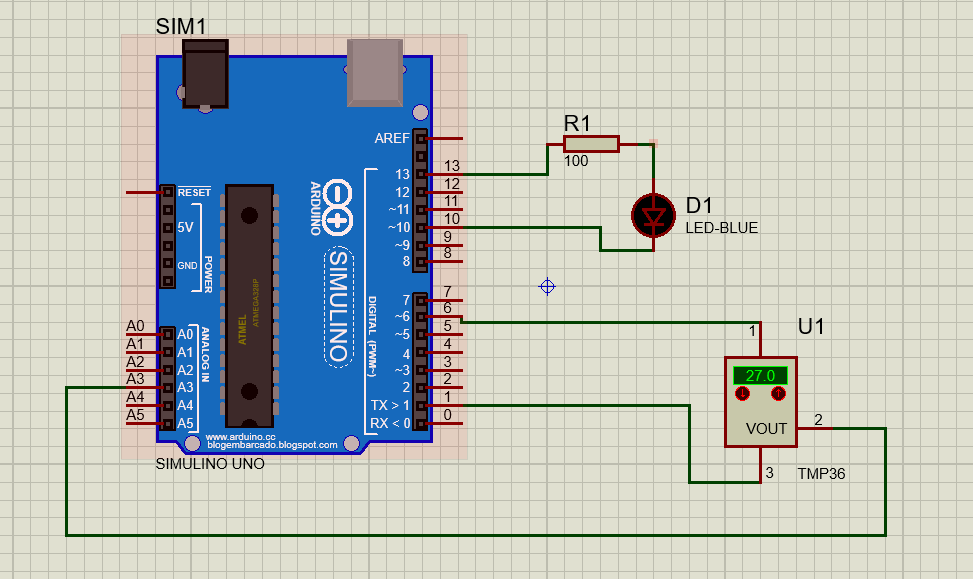


Figure 1 Sơ đồ kết nối

## 3.Đặc điểm linh kiện:

* 1 Arduino UNO
* 1 Led blue
* 1 Điện trở
* 1 TMP361.

## 4.Code:

*Void setup ()*

*{*

*Serial.begin(9600);*

*pinMode(6,1); pinMode(1,1); pinMode(13,1);*

*digitalWrite(6,1); digitalWrite(1,0);*

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

*}*

*void loop(){*

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

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

*if(analogRead(A3)<181){digitalWrite(13,1);};*

*}*

# **Bài led bật tắt**

## 1.Mô tả:

Thực hiện việc lập trình điều khiển bật tắt bằng nút bấm của đèn led được nối cổng 13 Arduino trong khoảng thời gian xác định.

## 2.Sơ đồ thiết kế:

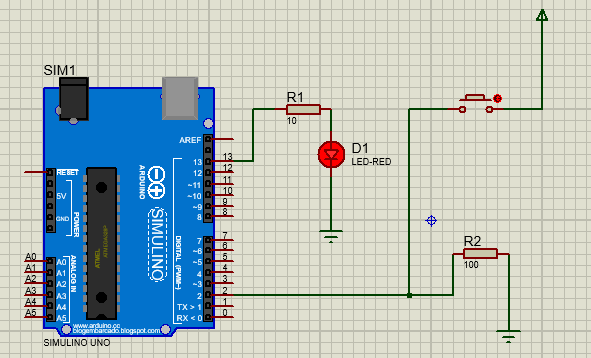


Figure 2 Sơ đồ kết nối

## 3.Đặc điểm linh kiện:

* 1 Arduino UNO
* 1 Led red
* 2 điện trở
* 1 nút bấm

## 4.Code:

*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 led 7 đoạn**

## 1.Mô tả**:**

Bài này thực hiện việc lập trình điều khiển led 7 đoạn hiện thị các chữ số từ 0 đến 9 qua mạch test-thử IC 4511

## 2.Sơ đồ thiết kế:

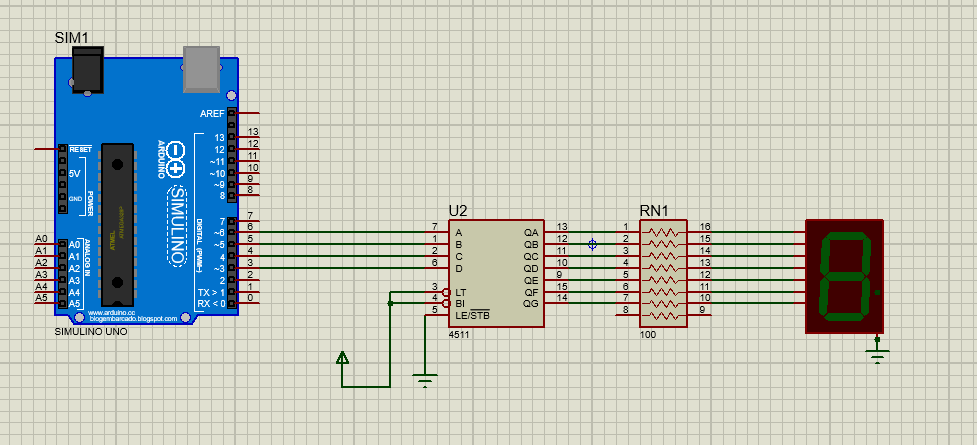
****

Figure 3 Sơ đồ kết nối

## 3.Đặt điểm linh kiện:

- 1 Mạch Arduino

- 1 Led 7 đoạn

- 1 Điện trở

- 1 4511 IC

- 2 Đất tiếp nối đèn và IC

## 4.Code :

*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() {*

*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 8 led nhấp nháy**

## 1.Mô tả**:**

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

## 2.Sơ đồ thiết kế:

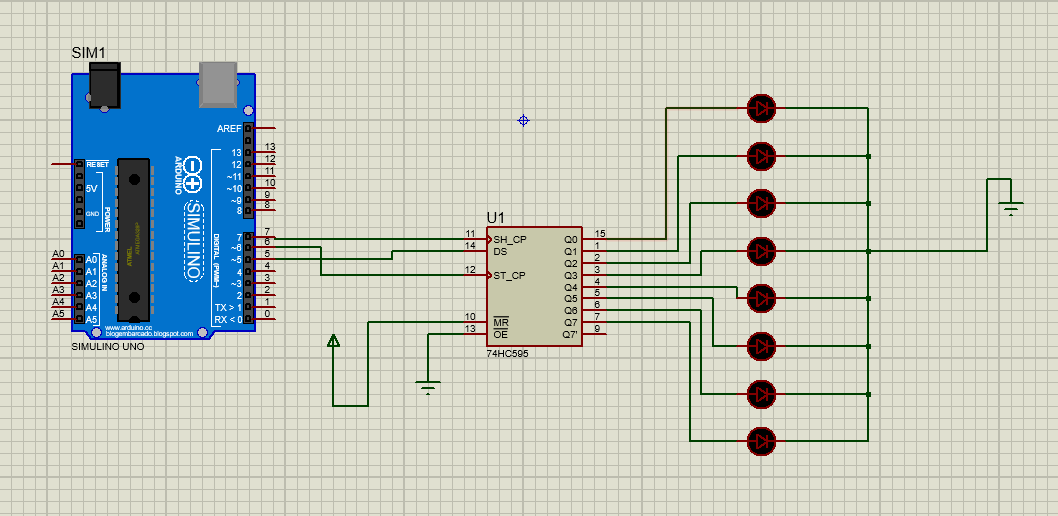


Figure 4 Sơ đồ kết nối

## 3.Đặc điểm linh kiện:

- 8 Led yellow

- 1 IC 74HC 595

- 1 ARDUINO UNO

- 2 Đất tiếp nối đèn và IC

## 4.Code:

|  |
| --- |
| *#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 led ma trận**

## 1.Môi tả :

Bài này thực hiện nháy đèn LED MATRIX 8x8 qua mạch IC 74HC595.

## 2.Sơ đồ thiết kế :

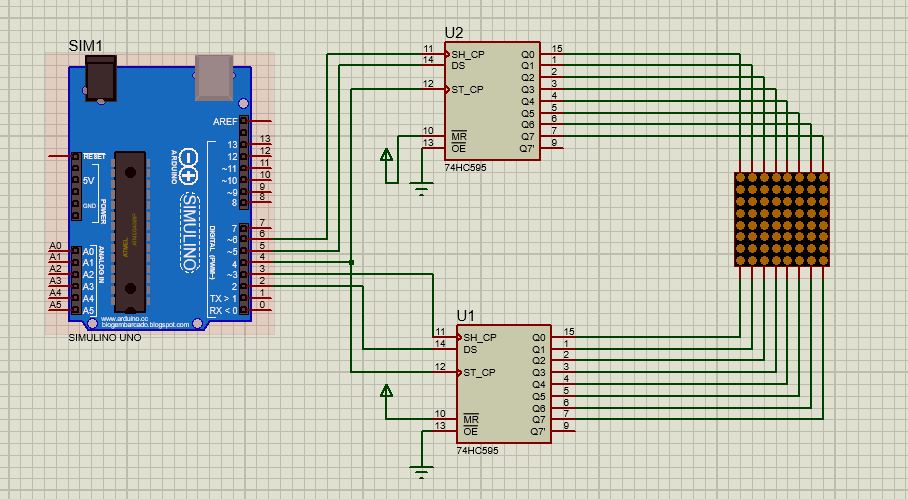


Figure 5 Sơ đồ kết nối

## 3.Đặc điểm linh kiện:

* Arduino UNO R3.
* 1 LED MATRIX 8x8.
* 2 IC 74HC595.
* 2 Đất tiếp nối đèn

## 4.Code :

*#define \_data\_cot 3*

*#define \_data\_hang 2*

*#define \_clock\_hang 6*

*#define \_clock\_cot 5*

*#define \_latch 4*

*byte chu[][8] = {*

*{0x3C,0x66,0x66,0x0C,0x18,0x30,0x7E,0x7E},//2*

*};*

*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,cot >> i);// hang 8 - hang 1*

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

*digitalWrite(\_latch,HIGH);*

*delay(100);*

*}*

# **Bài điều khiển motor bằng IC L1293D**

## 1.Mô tả :

* Thực hiện việc điều khiển làm quay 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

## 2.Sơ đồ thiết kế:

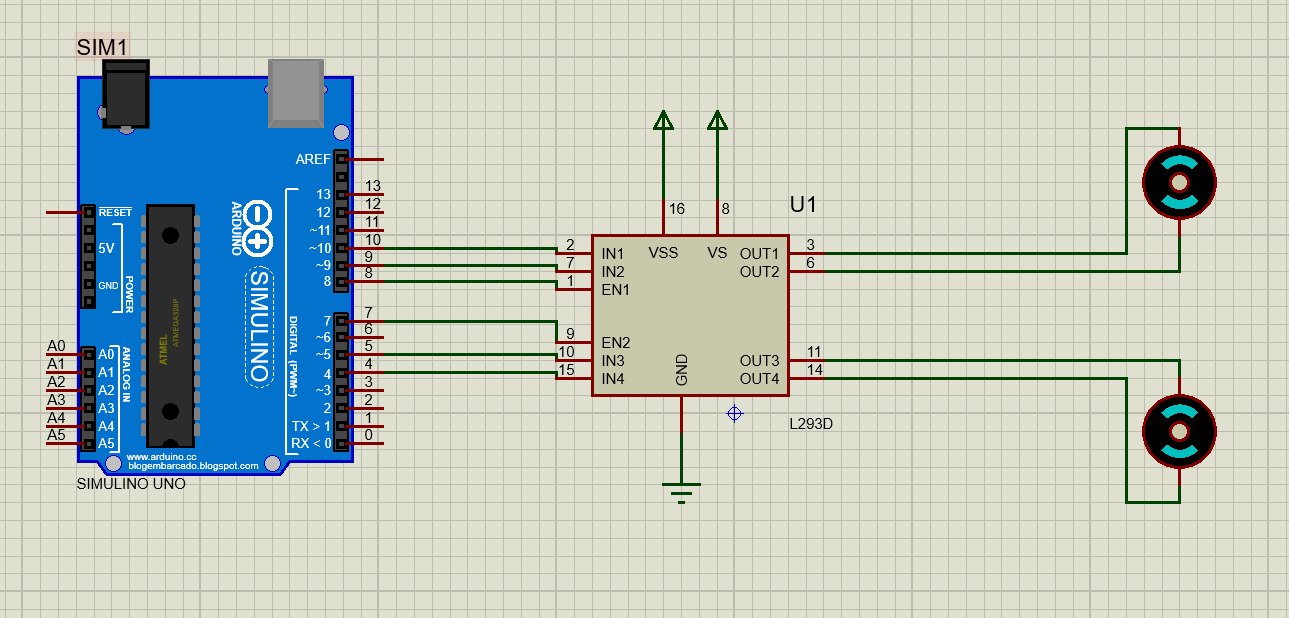


Figure 6 Sơ đồ kết nối

## 3.Đặc điểm linh kiện:

* 1 Arduino UNO
* 1 IC L293D
* 2 Motor

## 4.Code:

*// Kết nối động cơ A*

*int enA = 8;*

*int in1 = 10;*

*int in2 = 9;*

*// Kết nối động cơ B*

*int enB = 7;*

*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 nháy led 7 đoạn (0.0->9.9)**

## 1.Mô tả:

Bài này thực hiện việc lập trình điều khiển led 7 đoạn hiện thị các chữ số từ 0.0 đến 9.9 qua mạch test-thử IC 4511 , để thực hiện được ta cần dung 2 đèn led 7 đoạn kết nối với 2 IC.

## 2.Sơ đồ kết nối:

( Mô phỏng trên Tinkercard )

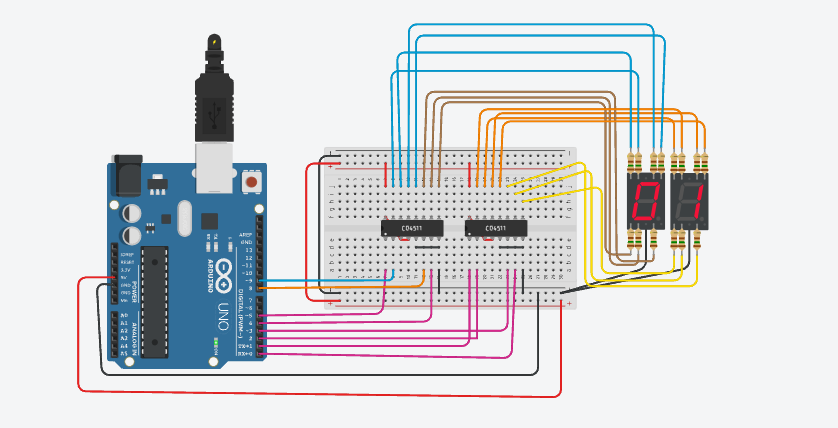


Figure 7 Mô phỏng Tinkercard

## 3.Đặc điểm linh kiện:

* 1 Mạch Arduino
* 2 Led 7 đoạn
* 14 Điện trở
* 2 IC 4511
* 1 Breadboard

## 4. Code:

*{*

*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);*

}

# **Bài đọc giá trị độ sáng và hiện ra LCD**

## 1.Mô tả:

Bài này thực hiện việc lập trình điều khiển LCD hiển thị ra độ sáng của đèn điện được gắn quang trở.

## 2.Sơ đồ kết nối:

( Mô phỏng trên Tinkercard )

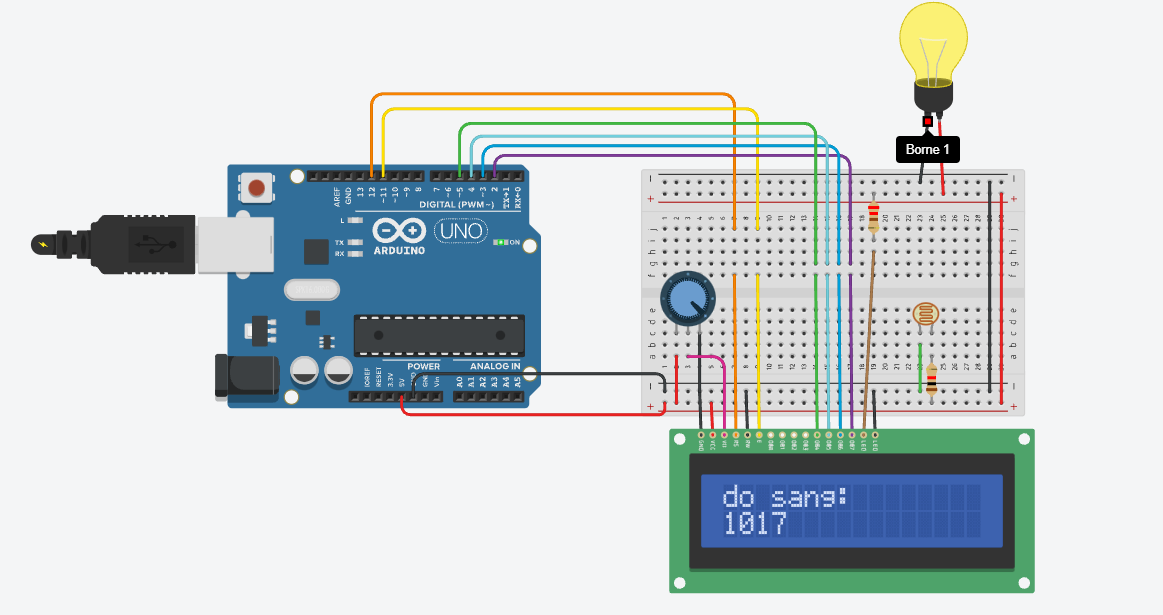


Figure 6 Mô phỏng Tinkercard

## 3.Đặc điểm linh kiện:

* 1 Mạch Arduino
* 1 LDR
* LCD
* 2 điện trở
* 1 Breadboard

## 4. Code:

*#include <LiquidCrystal.h>*

*#define SENSOR\_PIN A0*

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

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

*int LIGHT = 0;*

*int val =0;*

*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() {*

*val = analogRead(LIGHT);*

*// 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("do sang:");*

*lcd.setCursor(0, 1);*

*lcd.print(val);*

*}*

# **Bài chip STM32F401CB nháy 4 led**

## 1.Mô tả:

Bài này thực hiện lập trình điều khiển 4 đèn led sáng nhấp nháy bằng chip STM32F401VB.

## 2.Sơ đồ thiết kế:

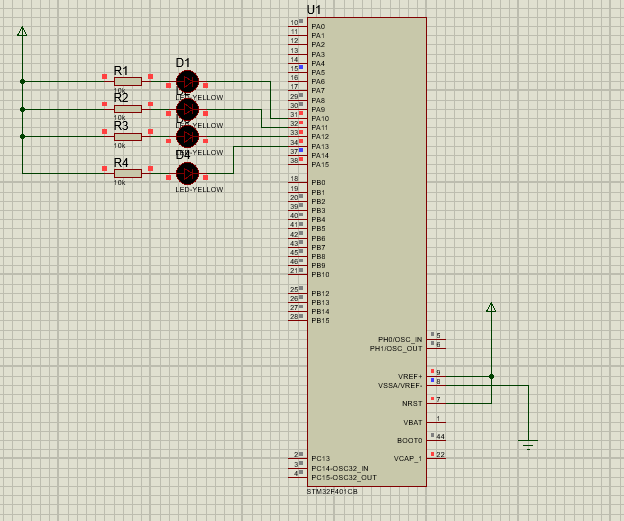


Figure 8 Sơ đồ thiết kế

## 3.Đặc điểm linh kiện:

* 1 Chip STM32F401VB
* 4 điện trở
* 4 đèn led

## 4.Code:

*#include "main.h"void SystemClock\_Config(void);*

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

*int main(void)*

*{HAL\_Init();*

*SystemClock\_Config();*

*MX\_GPIO\_Init();*

*while (1)*

*{*

*HAL\_GPIO\_TogglePin(GPIOA,GPIO\_PIN\_10);*

*HAL\_Delay(1000);*

*HAL\_GPIO\_TogglePin(GPIOA,GPIO\_PIN\_11);*

*HAL\_Delay(1000);*

*HAL\_GPIO\_TogglePin(GPIOA,GPIO\_PIN\_12);*

*HAL\_Delay(1000);*

*HAL\_GPIO\_TogglePin(GPIOA,GPIO\_PIN\_13);*

*}*

*void SystemClock\_Config(void)*

*{*

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

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

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

*\_\_HAL\_PWR\_VOLTAGESCALING\_CONFIG(PWR\_REGULATOR\_VOLTAGE\_SCALE2);*

*RCC\_OscInitStruct.OscillatorType = RCC\_OSCILLATORTYPE\_HSE;*

*RCC\_OscInitStruct.HSEState = RCC\_HSE\_ON;*

*RCC\_OscInitStruct.PLL.PLLState = RCC\_PLL\_NONE;*

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

*{*

*Error\_Handler();*

*}*

*RCC\_ClkInitStruct.ClockType = RCC\_CLOCKTYPE\_HCLK|RCC\_CLOCKTYPE\_SYSCLK*

*|RCC\_CLOCKTYPE\_PCLK1|RCC\_CLOCKTYPE\_PCLK2;*

*RCC\_ClkInitStruct.SYSCLKSource = RCC\_SYSCLKSOURCE\_HSE;*

*RCC\_ClkInitStruct.AHBCLKDivider = RCC\_SYSCLK\_DIV1;*

*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();*

*}*

*}*

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

*{*

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

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

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

*HAL\_GPIO\_WritePin(GPIOA, Led\_OUT\_Pin|Led\_OUTA11\_Pin|GPIO\_PIN\_12|GPIO\_PIN\_13, GPIO\_PIN\_RESET);*

*GPIO\_InitStruct.Pin = Led\_OUT\_Pin|Led\_OUTA11\_Pin|GPIO\_PIN\_12|GPIO\_PIN\_13;*

*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);*

*}*

*#ifdef USE\_FULL\_ASSERT*

*#endif*

# **Bài chip STM32F401VE sáng led trái tim (22led)**

## 1.Mô tả:

Bài này thực hiên điều khiển nháy sáng đèn led hình trái tim 22 led cạnh nhau và các hiệu ứng của nó qua mạch RES16DIPIS kết nối chip STM32F401VE.

## 2.Sơ đồ thiết kế

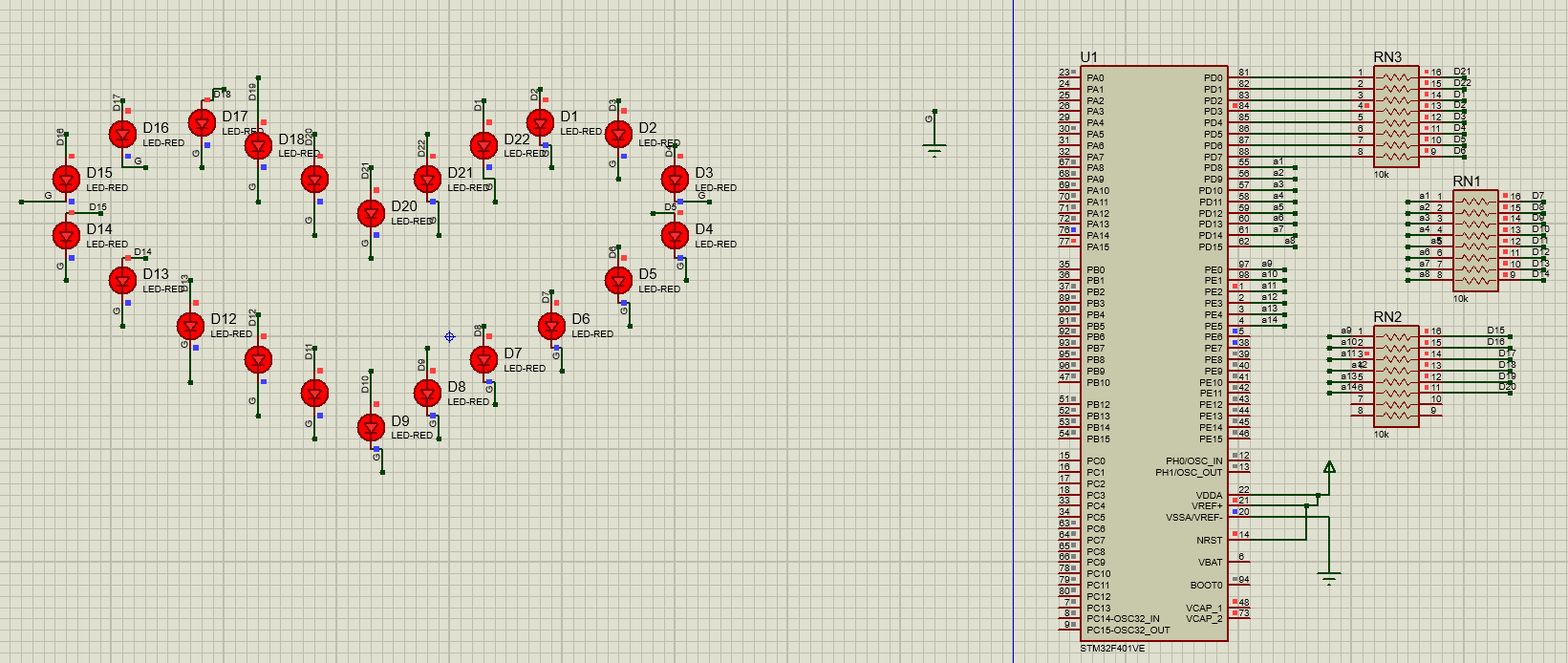


Figure 9 Sơ đồ thiết kế

## 3.Đặc điểm linh kiện:

* 22 Led-Red
* 3 mạch RES16DIPIS
* Chip STM32F401VE

## 4.Code:

*#include "main.h"*

*void SystemClock\_Config(void);*

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

*void SANGLED();*

*void TATLED();*

*void NHAPNHAY();*

*void NHAPNHAY3();*

*void NHAPNHA();*

*int main(void)*

*{*

*HAL\_Init();*

*SystemClock\_Config();*

*MX\_GPIO\_Init();*

*while (1)*

*{*

*SANGLED();*

*HAL\_Delay(1000);*

*TATLED();*

*HAL\_Delay(1000);*

*NHAPNHAY();*

*HAL\_Delay(500);*

*NHAPNHAY();*

*NHAPNHAY3();*

*NHAPNHA();*

*}*

*}*

*void TATLED()*

*{*

*SANGLED();*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_0, GPIO\_PIN\_RESET);*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_1, GPIO\_PIN\_RESET);*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_2, GPIO\_PIN\_RESET);*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_3, GPIO\_PIN\_RESET);*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_4, GPIO\_PIN\_RESET);*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_5, GPIO\_PIN\_RESET);*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_6, GPIO\_PIN\_RESET);*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_7, GPIO\_PIN\_RESET);*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_8, GPIO\_PIN\_RESET);*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_9, GPIO\_PIN\_RESET);*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_10, GPIO\_PIN\_RESET);*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_11, GPIO\_PIN\_RESET);*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_12, GPIO\_PIN\_RESET);*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_13, GPIO\_PIN\_RESET);*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_14, GPIO\_PIN\_RESET);*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_15, GPIO\_PIN\_RESET);*

*HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_0, GPIO\_PIN\_RESET);*

*HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_1, GPIO\_PIN\_RESET);*

*HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_2, GPIO\_PIN\_RESET);*

*HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_3, GPIO\_PIN\_RESET);*

*HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_4, GPIO\_PIN\_RESET);*

*HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_5, GPIO\_PIN\_RESET);*

*}*

*void SANGLED()*

*{*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_0, GPIO\_PIN\_SET);*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_1, GPIO\_PIN\_SET);*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_2, GPIO\_PIN\_SET);*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_3, GPIO\_PIN\_SET);*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_4, GPIO\_PIN\_SET);*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_5, GPIO\_PIN\_SET);*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_6, GPIO\_PIN\_SET);*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_7, GPIO\_PIN\_SET);*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_8, GPIO\_PIN\_SET);*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_9, GPIO\_PIN\_SET);*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_10, GPIO\_PIN\_SET);*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_11, GPIO\_PIN\_SET);*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_12, GPIO\_PIN\_SET);*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_13, GPIO\_PIN\_SET);*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_14, GPIO\_PIN\_SET);*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_15, GPIO\_PIN\_SET);*

*HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_0, GPIO\_PIN\_SET);*

*HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_1, GPIO\_PIN\_SET);*

*HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_2, GPIO\_PIN\_SET);*

*HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_3, GPIO\_PIN\_SET);*

*HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_4, GPIO\_PIN\_SET);*

*HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_5, GPIO\_PIN\_SET);*

*}*

*void NHAPNHAY()*

*{*

*TATLED();*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_0, GPIO\_PIN\_RESET);*

*HAL\_Delay(100);*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_1, GPIO\_PIN\_SET);*

*HAL\_Delay(100);*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_2, GPIO\_PIN\_RESET);*

*HAL\_Delay(100);*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_3, GPIO\_PIN\_SET);*

*HAL\_Delay(100);*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_4, GPIO\_PIN\_RESET);*

*HAL\_Delay(100);*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_5, GPIO\_PIN\_SET);*

*HAL\_Delay(100);*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_6, GPIO\_PIN\_RESET);*

*HAL\_Delay(100);*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_7, GPIO\_PIN\_SET);*

*HAL\_Delay(100);*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_8, GPIO\_PIN\_RESET);*

*HAL\_Delay(100);*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_9, GPIO\_PIN\_SET);*

*HAL\_Delay(100);*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_10, GPIO\_PIN\_RESET);*

*HAL\_Delay(100);*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_11, GPIO\_PIN\_SET);*

*HAL\_Delay(100);*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_12, GPIO\_PIN\_RESET);*

*HAL\_Delay(100);*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_13, GPIO\_PIN\_SET);*

*HAL\_Delay(100);*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_14, GPIO\_PIN\_RESET);*

*HAL\_Delay(100);*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_15, GPIO\_PIN\_SET);*

*HAL\_Delay(100);*

*HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_0, GPIO\_PIN\_RESET);*

*HAL\_Delay(100);*

*HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_1, GPIO\_PIN\_SET);*

*HAL\_Delay(100);*

*HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_2, GPIO\_PIN\_RESET);*

*HAL\_Delay(100);*

*HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_3, GPIO\_PIN\_SET);*

*HAL\_Delay(100);*

*HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_4, GPIO\_PIN\_RESET);*

*HAL\_Delay(100);*

*HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_5, GPIO\_PIN\_SET);*

*HAL\_Delay(100);*

*}*

*void NHAPNHAY3()*

*{*

*TATLED();*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_0, GPIO\_PIN\_SET);*

*HAL\_Delay(1000);*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_0, GPIO\_PIN\_RESET);*

*HAL\_Delay(1000);*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_1, GPIO\_PIN\_SET);*

*HAL\_Delay(1000);*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_1, GPIO\_PIN\_RESET);*

*HAL\_Delay(1000);*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_2, GPIO\_PIN\_SET);*

*HAL\_Delay(1000);*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_2, GPIO\_PIN\_RESET);*

*HAL\_Delay(1000);*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_3, GPIO\_PIN\_SET);*

*HAL\_Delay(100);*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_3, GPIO\_PIN\_RESET);*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_4, GPIO\_PIN\_SET);*

*HAL\_Delay(100);*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_4, GPIO\_PIN\_RESET);*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_5, GPIO\_PIN\_RESET);*

*HAL\_Delay(100);*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_6, GPIO\_PIN\_RESET);*

*HAL\_Delay(100);*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_7, GPIO\_PIN\_SET);*

*HAL\_Delay(100);*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_8, GPIO\_PIN\_SET);*

*HAL\_Delay(100);*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_9, GPIO\_PIN\_SET);*

*HAL\_Delay(100);*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_10, GPIO\_PIN\_RESET);*

*HAL\_Delay(100);*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_11, GPIO\_PIN\_RESET);*

*HAL\_Delay(100);*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_12, GPIO\_PIN\_RESET);*

*HAL\_Delay(100);*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_13, GPIO\_PIN\_SET);*

*HAL\_Delay(100);*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_14, GPIO\_PIN\_SET);*

*HAL\_Delay(100);*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_15, GPIO\_PIN\_SET);*

*HAL\_Delay(100);*

*HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_0, GPIO\_PIN\_RESET);*

*HAL\_Delay(100);*

*HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_1, GPIO\_PIN\_RESET);*

*HAL\_Delay(100);*

*HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_2, GPIO\_PIN\_RESET);*

*HAL\_Delay(100);*

*HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_3, GPIO\_PIN\_SET);*

*HAL\_Delay(100);*

*HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_4, GPIO\_PIN\_SET);*

*HAL\_Delay(100);*

*HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_5, GPIO\_PIN\_SET);*

*HAL\_Delay(100);*

*}*

*void NHAPNHA()*

*{*

*TATLED();*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_0, GPIO\_PIN\_SET);*

*HAL\_Delay(100);*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_0, GPIO\_PIN\_RESET);*

*HAL\_Delay(100);*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_0, GPIO\_PIN\_SET);*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_1, GPIO\_PIN\_SET);*

*HAL\_Delay(100);*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_1, GPIO\_PIN\_RESET);*

*HAL\_Delay(100);*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_1, GPIO\_PIN\_SET);*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_2, GPIO\_PIN\_SET);*

*HAL\_Delay(100);*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_2, GPIO\_PIN\_RESET);*

*HAL\_Delay(100);*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_2, GPIO\_PIN\_SET);*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_3, GPIO\_PIN\_SET);*

*HAL\_Delay(100);*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_3, GPIO\_PIN\_RESET);*

*HAL\_Delay(100);*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_3, GPIO\_PIN\_SET);*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_4, GPIO\_PIN\_SET);*

*HAL\_Delay(100);*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_4, GPIO\_PIN\_RESET);*

*HAL\_Delay(100);*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_4, GPIO\_PIN\_SET);*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_5, GPIO\_PIN\_SET);*

*HAL\_Delay(100);*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_5, GPIO\_PIN\_RESET);*

*HAL\_Delay(100);*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_5, GPIO\_PIN\_SET);*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_6, GPIO\_PIN\_SET);*

*HAL\_Delay(100);*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_6, GPIO\_PIN\_RESET);*

*HAL\_Delay(100);*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_6, GPIO\_PIN\_SET);*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_7, GPIO\_PIN\_SET);*

*HAL\_Delay(100);*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_7, GPIO\_PIN\_RESET);*

*HAL\_Delay(100);*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_7, GPIO\_PIN\_SET);*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_8, GPIO\_PIN\_SET);*

*HAL\_Delay(100);*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_8, GPIO\_PIN\_RESET);*

*HAL\_Delay(100);*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_8, GPIO\_PIN\_SET);*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_9, GPIO\_PIN\_SET);*

*HAL\_Delay(100);*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_9, GPIO\_PIN\_RESET);*

*HAL\_Delay(100);*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_9, GPIO\_PIN\_SET);*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_10, GPIO\_PIN\_SET);*

*HAL\_Delay(100);*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_10, GPIO\_PIN\_RESET);*

*HAL\_Delay(100);*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_10, GPIO\_PIN\_SET);*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_11, GPIO\_PIN\_SET);*

*HAL\_Delay(100);*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_11, GPIO\_PIN\_RESET);*

*HAL\_Delay(100);*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_11, GPIO\_PIN\_SET);*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_12, GPIO\_PIN\_SET);*

*HAL\_Delay(100);*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_12, GPIO\_PIN\_RESET);*

*HAL\_Delay(100);*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_12, GPIO\_PIN\_SET);*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_13, GPIO\_PIN\_SET);*

*HAL\_Delay(100);*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_13, GPIO\_PIN\_RESET);*

*HAL\_Delay(100);*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_13, GPIO\_PIN\_SET);*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_14, GPIO\_PIN\_SET);*

*HAL\_Delay(100);*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_14, GPIO\_PIN\_RESET);*

*HAL\_Delay(100);*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_14, GPIO\_PIN\_SET);*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_15, GPIO\_PIN\_SET);*

*HAL\_Delay(100);*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_15, GPIO\_PIN\_RESET);*

*HAL\_Delay(100);*

*HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_15, GPIO\_PIN\_SET);*

*HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_0, GPIO\_PIN\_SET);*

*HAL\_Delay(100);*

*HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_0, GPIO\_PIN\_RESET);*

*HAL\_Delay(100);*

*HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_0, GPIO\_PIN\_SET);*

*HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_1, GPIO\_PIN\_SET);*

*HAL\_Delay(100);*

*HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_1, GPIO\_PIN\_RESET);*

*HAL\_Delay(100);*

*HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_1, GPIO\_PIN\_SET);*

*HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_2, GPIO\_PIN\_SET);*

*HAL\_Delay(100);*

*HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_2, GPIO\_PIN\_RESET);*

*HAL\_Delay(100);*

*HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_2, GPIO\_PIN\_SET);*

*HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_3, GPIO\_PIN\_SET);*

*HAL\_Delay(100);*

*HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_3, GPIO\_PIN\_RESET);*

*HAL\_Delay(100);*

*HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_3, GPIO\_PIN\_SET);*

*HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_4, GPIO\_PIN\_SET);*

*HAL\_Delay(100);*

*HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_4, GPIO\_PIN\_RESET);*

*HAL\_Delay(100);*

*HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_4, GPIO\_PIN\_SET);*

*HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_5, GPIO\_PIN\_SET);*

*HAL\_Delay(100);*

*HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_5, GPIO\_PIN\_RESET);*

*HAL\_Delay(100);*

*HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_5, GPIO\_PIN\_SET);*

*SANGLED();*

*HAL\_Delay(100);*

*TATLED();*

*HAL\_Delay(500);*

*SANGLED ();*

*HAL\_Delay(500);*

*TATLED();*

*HAL\_Delay(500);*

*SANGLED ();*

*HAL\_Delay(500);*

*TATLED();*

*HAL\_Delay(500);*

*SANGLED ();*

*}*

*/\*\**

*\* @brief System Clock Configuration*

*\* @retval None*

*\*/*

*void SystemClock\_Config(void)*

*{*

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

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

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

*\_\_HAL\_PWR\_VOLTAGESCALING\_CONFIG(PWR\_REGULATOR\_VOLTAGE\_SCALE2);*

*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\_DIV1;*

*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\_GPIOE\_CLK\_ENABLE();*

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

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

*HAL\_GPIO\_WritePin(GPIOE, GPIO\_PIN\_2|GPIO\_PIN\_3|GPIO\_PIN\_4|GPIO\_PIN\_5*

*|GPIO\_PIN\_6|GPIO\_PIN\_7|GPIO\_PIN\_0|GPIO\_PIN\_1, GPIO\_PIN\_RESET);*

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

*HAL\_GPIO\_WritePin(GPIOD, GPIO\_PIN\_8|GPIO\_PIN\_9|GPIO\_PIN\_10|GPIO\_PIN\_11*

*|GPIO\_PIN\_12|GPIO\_PIN\_13|GPIO\_PIN\_14|GPIO\_PIN\_15*

*|GPIO\_PIN\_0|GPIO\_PIN\_1|GPIO\_PIN\_2|GPIO\_PIN\_3*

*|GPIO\_PIN\_4|GPIO\_PIN\_5|GPIO\_PIN\_6|GPIO\_PIN\_7, GPIO\_PIN\_RESET);*

*/\*Configure GPIO pins : PE2 PE3 PE4 PE5*

*PE6 PE7 PE0 PE1 \*/*

*GPIO\_InitStruct.Pin = GPIO\_PIN\_2|GPIO\_PIN\_3|GPIO\_PIN\_4|GPIO\_PIN\_5*

*|GPIO\_PIN\_6|GPIO\_PIN\_7|GPIO\_PIN\_0|GPIO\_PIN\_1;*

*GPIO\_InitStruct.Mode = GPIO\_MODE\_OUTPUT\_PP;*

*GPIO\_InitStruct.Pull = GPIO\_NOPULL;*

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

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

*/\*Configure GPIO pins : PD8 PD9 PD10 PD11*

*PD12 PD13 PD14 PD15*

*PD0 PD1 PD2 PD3*

*PD4 PD5 PD6 PD7 \*/*

*GPIO\_InitStruct.Pin = GPIO\_PIN\_8|GPIO\_PIN\_9|GPIO\_PIN\_10|GPIO\_PIN\_11*

*|GPIO\_PIN\_12|GPIO\_PIN\_13|GPIO\_PIN\_14|GPIO\_PIN\_15*

*|GPIO\_PIN\_0|GPIO\_PIN\_1|GPIO\_PIN\_2|GPIO\_PIN\_3*

*|GPIO\_PIN\_4|GPIO\_PIN\_5|GPIO\_PIN\_6|GPIO\_PIN\_7;*

*GPIO\_InitStruct.Mode = GPIO\_MODE\_OUTPUT\_PP;*

*GPIO\_InitStruct.Pull = GPIO\_NOPULL;*

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

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

*}*

*void Error\_Handler(void)*

*{*

*\_\_disable\_irq();*

*while (1)*

*{}*

*}*

*#ifdef USE\_FULL\_ASSERT*

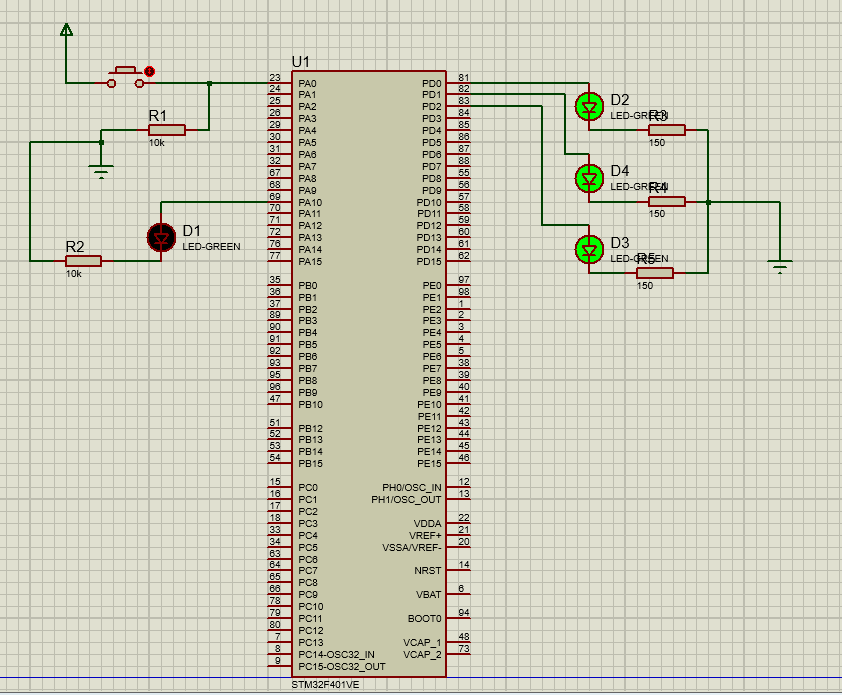
*#endif*

# **Ngắt ngoài và ngắt ưu tiên trên STM32F4**

## 1.Mô tả:

Bài này thưc hiên viêc ngắn ngoài bằng nút bấm điều khiển đèn led và mạch chính 3 đèn led vẫn hoat đông song hành đươc kết nối chim STM32F401VE.

## 2.Sơ đồ thiết kế :



## 3.Mô tả linh kiện :

* 4 đèn led
* 1 chip STM32F401VE
* 1 Button

## 4.Code :

*#include "main.h"*

*void SystemClock\_Config(void);*

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

*int main(void){*

*HAL\_Init();*

*SystemClock\_Config();*

*MX\_GPIO\_Init();*

*while (1)*

*{*

*HAL\_GPIO\_TogglePin(GPIOD, GPIO\_PIN\_0|GPIO\_PIN\_1|GPIO\_PIN\_2);*

*HAL\_Delay(1000);*

*}*

*}*

*void HAL\_GPIO\_EXTI\_Callback(uint16\_t GPIO\_Pin)*

*{*

*if (GPIO\_Pin == GPIO\_PIN\_0)*

*{*

*HAL\_GPIO\_TogglePin(GPIOA, GPIO\_PIN\_10);*

*}*

*}*

*/\*\**

*\* @brief System Clock Configuration*

*\* @retval None*

*\*/*

*void SystemClock\_Config(void)*

*{*

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

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

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

*\_\_HAL\_PWR\_VOLTAGESCALING\_CONFIG(PWR\_REGULATOR\_VOLTAGE\_SCALE2);*

*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();*

*}*

*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\_DIV1;*

*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};*

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

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

*HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_10, GPIO\_PIN\_RESET);*

*HAL\_GPIO\_WritePin(GPIOD, GPIO\_PIN\_0|GPIO\_PIN\_1|GPIO\_PIN\_2, GPIO\_PIN\_RESET);*

*GPIO\_InitStruct.Pin = GPIO\_PIN\_0;*

*GPIO\_InitStruct.Mode = GPIO\_MODE\_IT\_RISING;*

*GPIO\_InitStruct.Pull = GPIO\_NOPULL;*

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

*/\*Configure GPIO pin : PA10 \*/*

*GPIO\_InitStruct.Pin = GPIO\_PIN\_10;*

*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);*

*/\*Configure GPIO pins : PD0 PD1 PD2 \*/*

*GPIO\_InitStruct.Pin = GPIO\_PIN\_0|GPIO\_PIN\_1|GPIO\_PIN\_2;*

*GPIO\_InitStruct.Mode = GPIO\_MODE\_OUTPUT\_PP;*

*GPIO\_InitStruct.Pull = GPIO\_NOPULL;*

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

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

*/\* EXTI interrupt init\*/*

*HAL\_NVIC\_SetPriority(EXTI0\_IRQn, 0, 0);*

*HAL\_NVIC\_EnableIRQ(EXTI0\_IRQn);*

*}*

*void Error\_Handler(void)*

*{*

*\_\_disable\_irq();*

*while (1)*

*{*

*}*

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

*}*

*#ifdef USE\_FULL\_ASSERT*

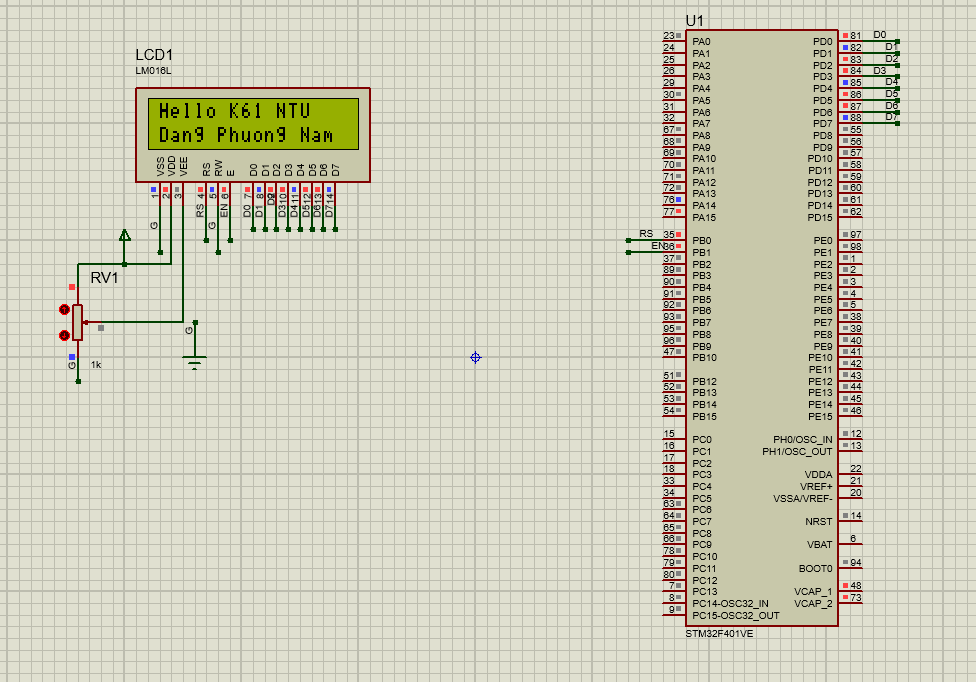
# **Bài lập trình STM32- Kết nối LCD trực tiếp**

## 1.Mô tả :

Bài này ta thực hiện việc lập trình điều khiển qua lệnh , con trỏ để hiển thị thông tin trên màn hình LCD .

LCD là một màn hình nhiều điểm ảnh, có thể coi là một Led ma trận dạng lớn.

## 2.Sơ đồ kết nối :



## 3.Đặc điểm linh kiện :

* 1 chip STM32F401VE
* 1 điện trở POT
* 1 LCD016L

## 4.Code :

*#include "main.h"*

*void SystemClock\_Config(void);*

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

*//HAM GUI LENH*

*void send8bit(char D)*

*{*

*int b0,b1,b2,b3,b4,b5,b6,b7;*

*if ((D &1)==0) b0=0; //2^0*

*else b0=1;*

*if ((D &2)==0) b1=0;*

*else b1=1;*

*if ((D &4)==0) b2=0;*

*else b2=1;*

*if ((D &8)==0) b3=0;*

*else b3=1;*

*if ((D &16)==0) b4=0;*

*else b4=1;*

*if ((D &32)==0) b5=0;*

*else b5=1;*

*if ((D &64)==0) b6=0;*

*else b6=1;*

*if ((D &128)==0) b7=0;*

*else b7=1;*

*HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_0,b0);*

*HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_1,b1);*

*HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_2,b2);*

*HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_3,b3);*

*HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_4,b4);*

*HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_5,b5);*

*HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_6,b6);*

*HAL\_GPIO\_WritePin(GPIOD,GPIO\_PIN\_7,b7);*

*}*

*void sendlcd(char cmd)*

*{*

*// Rw dat muc thap*

*//b2 dat chan rs=0,de noi rang cmd la lenh*

*HAL\_GPIO\_WritePin(GPIOB, GPIO\_PIN\_0, GPIO\_PIN\_RESET);*

*//B3 GUI 8 BIET VAO 8 PIN*

*send8bit(cmd);*

*//b4 enable cho kytu -- < lcd*

*HAL\_GPIO\_WritePin(GPIOB, GPIO\_PIN\_1, GPIO\_PIN\_RESET);*

*HAL\_GPIO\_WritePin(GPIOB, GPIO\_PIN\_1, GPIO\_PIN\_SET);*

*HAL\_Delay(10);*

*}*

*// ham gui ky tu*

*void sendchar(char kytu)*

*{*

*// Rw dat muc thap*

*//b2 dat chan rs=1*

*HAL\_GPIO\_WritePin(GPIOB, GPIO\_PIN\_0, GPIO\_PIN\_SET);*

*//B3 GUI 8 BIET VAO 8 PIN*

*send8bit(kytu);*

*//b4 enable cho kytu -- < lcd*

*HAL\_GPIO\_WritePin(GPIOB, GPIO\_PIN\_1, GPIO\_PIN\_RESET);*

*HAL\_GPIO\_WritePin(GPIOB, GPIO\_PIN\_1, GPIO\_PIN\_SET);*

*HAL\_Delay(10);*

*}*

*void sendstring(char \*str)*

*{*

*for (int i =0;str[i] !='\0';i++)*

*sendchar(str[i]);*

*}*

*int main(void)*

*{*

*HAL\_Init();*

*SystemClock\_Config();*

*MX\_GPIO\_Init();*

*//xoa noi dung*

*sendlcd(0x01);*

*// bat hien thi man hinh, tat con tro*

*sendlcd(0x0C);*

*//hello*

*sendstring("Hello K61 NTU");*

*//xuong dong*

*sendlcd(0x38);//che do 2 dong*

*sendlcd(0xC0);*

*sendstring("Dang Phuong Nam");*

*while (1)*

*{*

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

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

*}*

*/\* 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\_DIV1;*

*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\_GPIOB\_CLK\_ENABLE();*

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

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

*HAL\_GPIO\_WritePin(GPIOB, GPIO\_PIN\_0|GPIO\_PIN\_1, GPIO\_PIN\_RESET);*

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

*HAL\_GPIO\_WritePin(GPIOD, GPIO\_PIN\_0|GPIO\_PIN\_1|GPIO\_PIN\_2|GPIO\_PIN\_3*

*|GPIO\_PIN\_4|GPIO\_PIN\_5|GPIO\_PIN\_6|GPIO\_PIN\_7, GPIO\_PIN\_RESET);*

*/\*Configure GPIO pins : PB0 PB1 \*/*

*GPIO\_InitStruct.Pin = GPIO\_PIN\_0|GPIO\_PIN\_1;*

*GPIO\_InitStruct.Mode = GPIO\_MODE\_OUTPUT\_PP;*

*GPIO\_InitStruct.Pull = GPIO\_NOPULL;*

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

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

*/\*Configure GPIO pins : PD0 PD1 PD2 PD3*

*PD4 PD5 PD6 PD7 \*/*

*GPIO\_InitStruct.Pin = GPIO\_PIN\_0|GPIO\_PIN\_1|GPIO\_PIN\_2|GPIO\_PIN\_3*

*|GPIO\_PIN\_4|GPIO\_PIN\_5|GPIO\_PIN\_6|GPIO\_PIN\_7;*

*GPIO\_InitStruct.Mode = GPIO\_MODE\_OUTPUT\_PP;*

*GPIO\_InitStruct.Pull = GPIO\_NOPULL;*

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

*HAL\_GPIO\_Init(GPIOD, &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 \*/*

*\_\_disable\_irq();*

*while (1)*

*{*

*}*

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

*}*

*#ifdef USE\_FULL\_ASSERT*

**Mục lục**

[**Bài TMP36 đo led** 2](#_Toc87025568)

[1.Mô tả: 2](#_Toc87025569)

[2.Sơ đồ thiết kế: 2](#_Toc87025570)

[3.Đặc điểm linh kiện: 2](#_Toc87025571)

[4.Code: 2](#_Toc87025572)

[**Bài led bật tắt** 3](#_Toc87025573)

[1.Mô tả: 3](#_Toc87025574)

[2.Sơ đồ thiết kế: 3](#_Toc87025575)

[3.Đặc điểm linh kiện: 3](#_Toc87025576)

[4.Code: 3](#_Toc87025577)

[**Bài led 7 đoạn** 4](#_Toc87025578)

[1.Mô tả**:** 4](#_Toc87025579)

[2.Sơ đồ thiết kế: 4](#_Toc87025580)

[3.Đặt điểm linh kiện: 5](#_Toc87025581)

[4.Code : 5](#_Toc87025582)

[**Bài 8 led nhấp nháy** 9](#_Toc87025583)

[1.Mô tả**:** 9](#_Toc87025584)

[2.Sơ đồ thiết kế: 9](#_Toc87025585)

[3.Đặc điểm linh kiện: 9](#_Toc87025586)

[4.Code: 9](#_Toc87025587)

[**Bài led ma trận** 11](#_Toc87025588)

[1.Môi tả : 11](#_Toc87025589)

[2.Sơ đồ thiết kế : 11](#_Toc87025590)

[3.Đặc điểm linh kiện: 11](#_Toc87025591)

[4.Code : 11](#_Toc87025592)

[**Bài điều khiển motor bằng IC L1293D** 13](#_Toc87025593)

[1.Mô tả : 13](#_Toc87025594)

[2.Sơ đồ thiết kế: 13](#_Toc87025595)

[3.Đặc điểm linh kiện: 13](#_Toc87025596)

[4.Code: 13](#_Toc87025597)

[**Bài nháy led 7 đoạn (0.0->9.9)** 18](#_Toc87025598)

[1.Mô tả: 18](#_Toc87025599)

[2.Sơ đồ kết nối: 18](#_Toc87025600)

[3.Đặc điểm linh kiện: 18](#_Toc87025601)

[4. Code: 18](#_Toc87025602)

[**Bài đọc giá trị độ sáng và hiện ra LCD** 23](#_Toc87025603)

[1.Mô tả: 23](#_Toc87025604)

[2.Sơ đồ kết nối: 23](#_Toc87025605)

[3.Đặc điểm linh kiện: 23](#_Toc87025606)

[4. Code: 23](#_Toc87025607)

[**Bài chip STM32F401CB nháy 4 led** 25](#_Toc87025608)

[1.Mô tả: 25](#_Toc87025609)

[2.Sơ đồ thiết kế: 25](#_Toc87025610)

[3.Đặc điểm linh kiện: 25](#_Toc87025611)

[4.Code: 25](#_Toc87025612)

[**Bài chip STM32F401VE sáng led trái tim (22led)** 28](#_Toc87025613)

[1.Mô tả: 28](#_Toc87025614)

[2.Sơ đồ thiết kế 28](#_Toc87025615)

[3.Đặc điểm linh kiện: 28](#_Toc87025616)

[4.Code: 28](#_Toc87025617)

[**Ngắt ngoài và ngắt ưu tiên trên STM32F4** 43](#_Toc87025618)

[1.Mô tả: 43](#_Toc87025619)

[2.Sơ đồ thiết kế : 43](#_Toc87025620)

[3.Mô tả linh kiện : 43](#_Toc87025621)

[4.Code : 43](#_Toc87025622)