Đặng Lê Hoài Nhi 61CNTT1

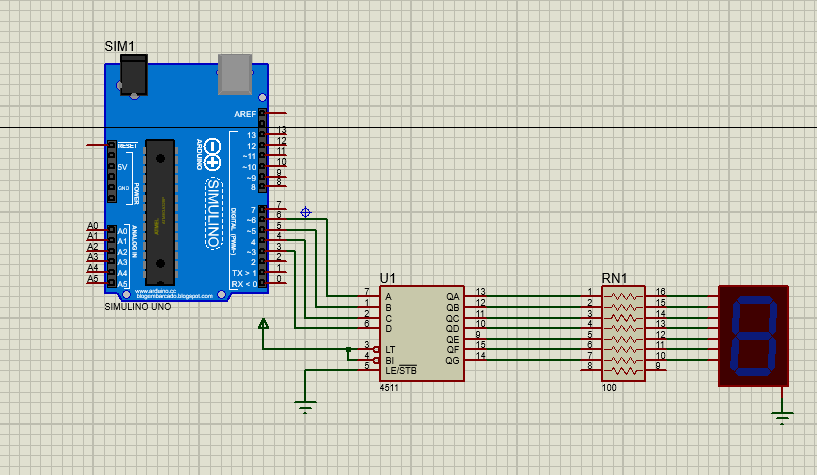
Đại học nha trang

BÁo cáo Lập trình nhúng

# Bài 1: Đèn 7 đoạn

* Mô tả nguyên lý hoạt động : Muốn LED nào sáng thì LED đó phải được phân cực thuận.Do đó muốn tạo ra chữ số nào ta chỉ cần cho LED ở các vị trí tương ứng sáng lên. Bảng mô tả cách tạo ra các chữ số để hiển thị lên LED 7 đoạn.

## Sơ đồ mạch:



Hình 1 : Sơ đồ lắp ghép

## Linh kiện

-Simulino Uno

-Thanh điện trở 10 chân RES16DIPIS

-Khối giải mã IC4511

- Led 7 đoạn

## CODE

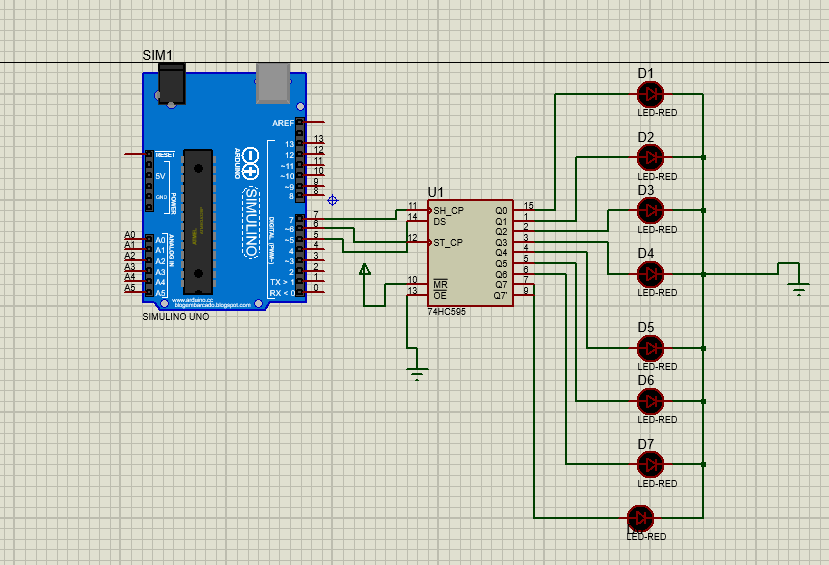
|  |
| --- |
| int a=6, b=5, c=4, d=3; |
|  |  |
|  | void setup() { |
|  | pinMode(a, OUTPUT); |
|  | pinMode(b, OUTPUT); |
|  | pinMode(c, OUTPUT); |
|  | pinMode(d, OUTPUT); |
|  | } |
|  | void khong(){ |
|  | digitalWrite(a, LOW); |
|  | digitalWrite(b, LOW); |
|  | digitalWrite(c, LOW); |
|  | 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() { |
|  | // 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 2 : 8 Led

## Mô tả :

Điều khiểu 8 bóng đèn sáng luân phiên nhau

## Sơ đồ mạch



## Linh Kiện

--Simulino Uno

- 8 bóng led đỏ

- Thanh ghi dịch 74HC595

## CODE

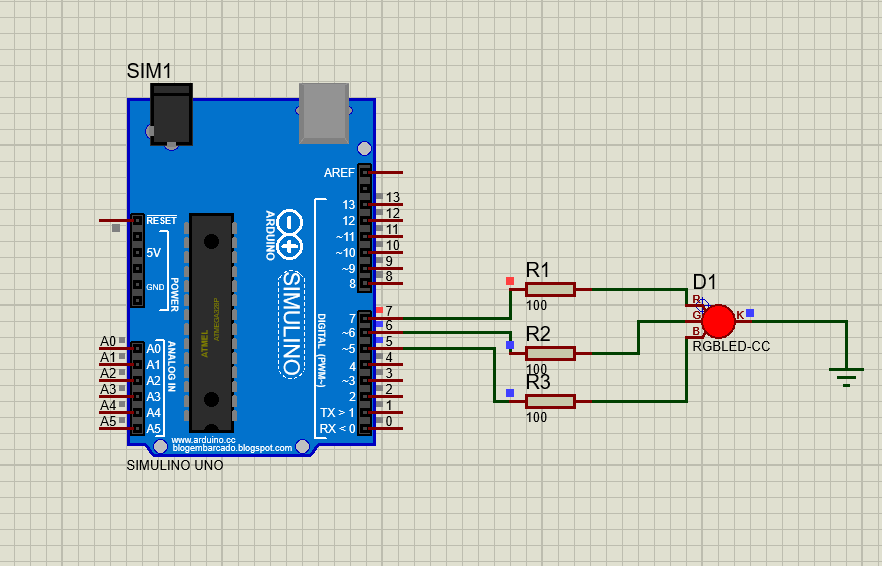
|  |
| --- |
| #define\_clock7 |
|  | #define \_latch 6 |
|  | #define \_data 5 |
|  | void setup() { |
|  | // put your setup code here, to run once: |
|  | pinMode(\_latch, OUTPUT); |
|  | pinMode(\_clock, OUTPUT); |
|  | pinMode(\_data, OUTPUT); |
|  | } |
|  |  |
|  | void loop() { |
|  | // put 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 3 : Led RGB

## Mô tả :

Thực hiện nháy led RGB, nối các chân RGB vào các điện trở và nối tiếp vào chân 6,5,3 của Arduino

## Sơ đồ mạch



Sơ đồ mạch led RGB

## Linh kiện :

-3 Điện trở 100 **Ω**

**-**1 mạch Arduino Uno để lắp

- 1 Led RGB

## Code

|  |
| --- |
| int redPin= 7;  int greenPin = 6;  int bluePin = 5;  void setup()  {  pinMode(redPin, OUTPUT);  pinMode(greenPin, OUTPUT);  pinMode(bluePin, OUTPUT);  }  void loop()  {  setColor(255, 0, 0); // Red Color  delay(1000);  setColor(0, 255, 0); // Green Color  delay(1000);  setColor(0, 0, 255); // Blue Color  delay(1000);  setColor(255, 255, 255); // White Color  delay(1000);  setColor(170, 0, 255); // Purple Color  delay(1000);  }  void setColor(int redValue, int greenValue, int blueValue)  {  analogWrite(redPin, redValue);  analogWrite(greenPin, greenValue);  analogWrite(bluePin, blueValue);  } |

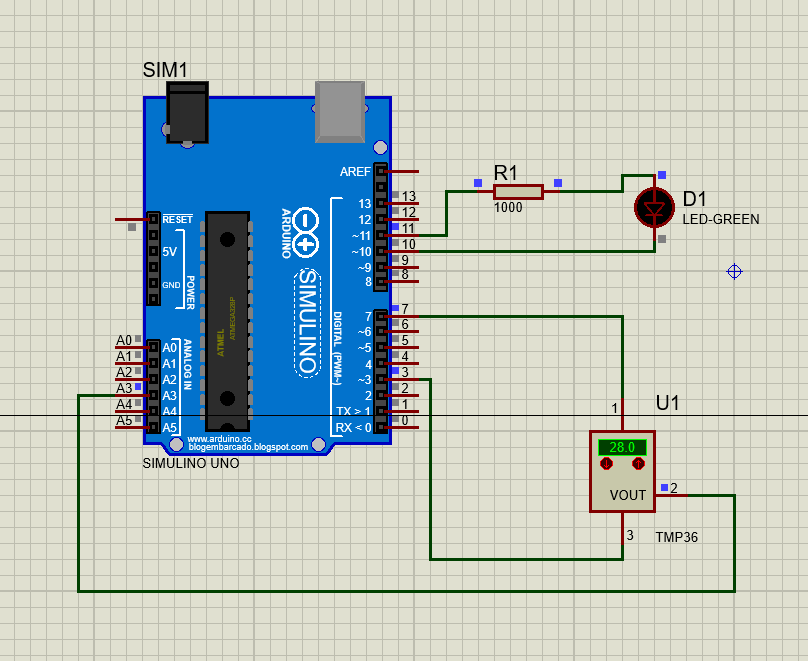
# Bài 4 : Cảm biến nhiệt độ TMP36

## Mô tả :

TMP36 có 3 chân

* Chân số 1 là chân cấp nguồn 5V
* Chân thứ 2 là chân xuất tín hiệu tương tự (tín hiệu dạng xung).
* Chân thứ 3 là chân nối mát hay chân GND

## Sơ đồ mạch



Sơ đồ mạch lắp TMP36

## Linh kiện

- 1 điện trở

-1 led

- 1 TMP36 ( cảm biến nhiệt độ)

- 1 mạch Arduino Uno

## Code

|  |
| --- |
| void setup()  {  Serial.begin(9600);  pinMode(7,1);  pinMode(3,1);  digitalWrite(7,1);  digitalWrite(3,0);  digitalWrite(10,1);  digitalWrite(11,0);  pinMode(11,1);  }  void loop()  {  Serial.println(analogRead(A3));  if(analogRead(A3)>181){digitalWrite(10,0);};  if(analogRead(A3)<181){digitalWrite(10,1);};  } |

# Bài 5 : STM32 nhấp nháy led

## Sơ đồ mạch

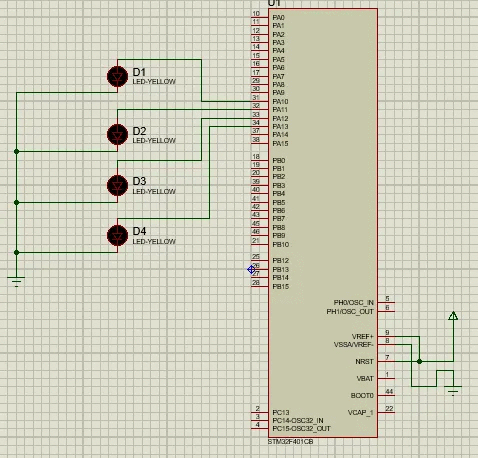


Figure Sơ đồ lắp mạch

## Linh kiện

-1 mạch stm32f4

Điện trở

Đèn Led

## Code

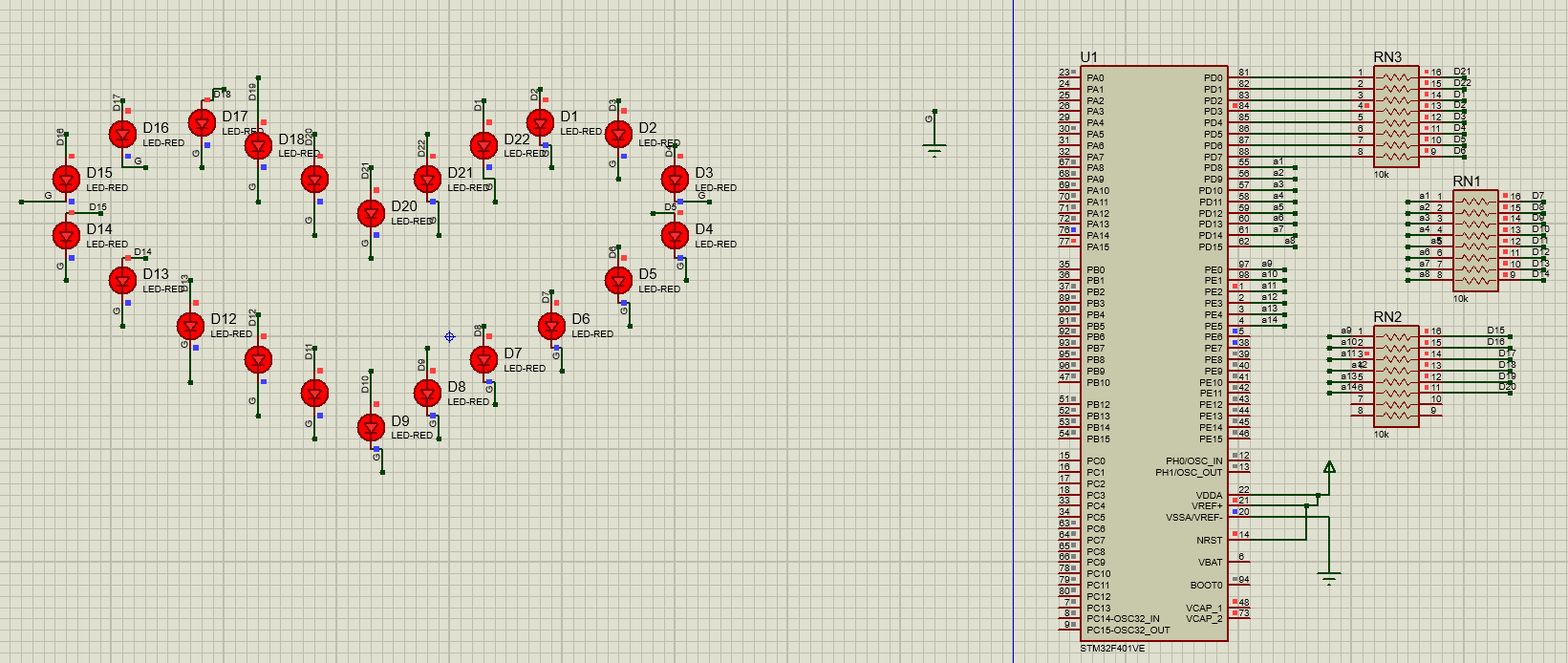
|  |
| --- |
| #include "main.h"void SystemClock\_Config(void);  static void MX\_GPIO\_Init(void);  int main(void)  {HAL\_Init();  SystemClock\_Config();  MX\_GPIO\_Init();  while (1)  { HAL\_GPIO\_TogglePin(GPIOA,GPIO\_PIN\_10);  HAL\_Delay(1000); HAL\_GPIO\_TogglePin(GPIOA,GPIO\_PIN\_11);  HAL\_Delay(1000); 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 6: STM32F41 Led trái tim

## Mô tả :

Sử dụng chip STM32F401VE để điều khiển led sáng theo hình trái tim , nạp code thông qua KeilCv5

## Sơ đồ mạch



Mạch Proteus Led trái tim

## Linh kiện

Led red

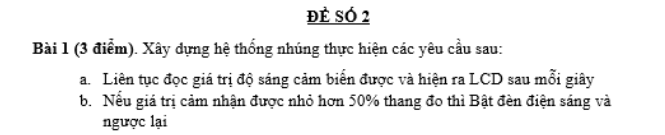
3 RES16DIPIS

1 STM32F401VE

## 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* |

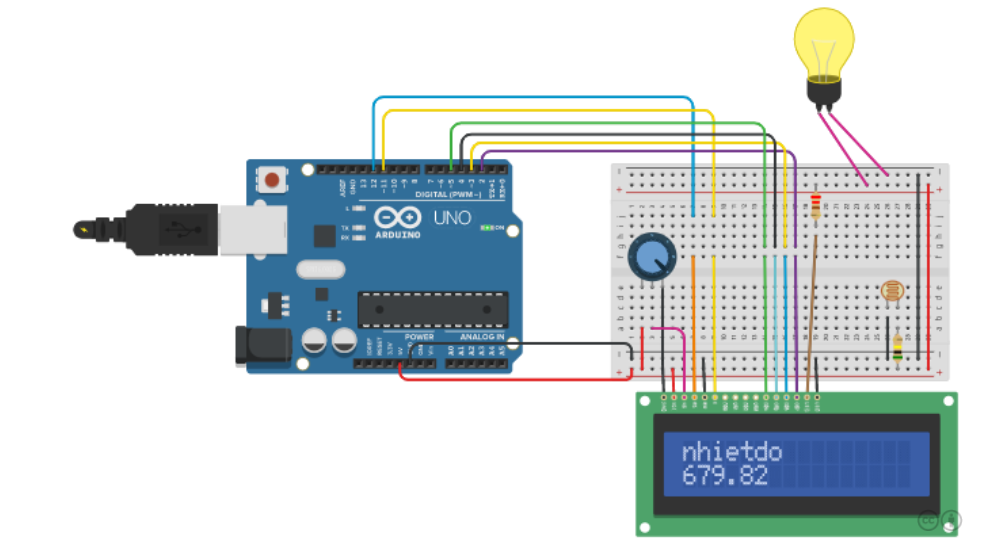
# Bài 7: Xây dựng hệ thống nhúng để đo độ sáng Led



## Mô tả

Quang trở là một loại "vật liệu" điện tử rất hay gặp và được sử dụng trong những mạch cảm biến ánh sáng[.](http://arduino.vn/bai-viet/208-cach-doc-du-lieu-tu-quang-tro-va-xay-dung-cam-bien-anh-sang) Có thể hiểu một cách dễ dàng rằng, quang trở là một loại ĐIỆN TRỞ có điện trở thay đổi theo cường độ ánh sáng. Nếu đặt ở môi trường có ít ánh sáng, có bóng râm hoặc tối thì điện trở của quang trở sẽ tăng cao còn nếu đặt ở ngoài nắng, hoặc nơi có ánh sáng thì điện trở sẽ giảm.

## Sơ đồ mạch



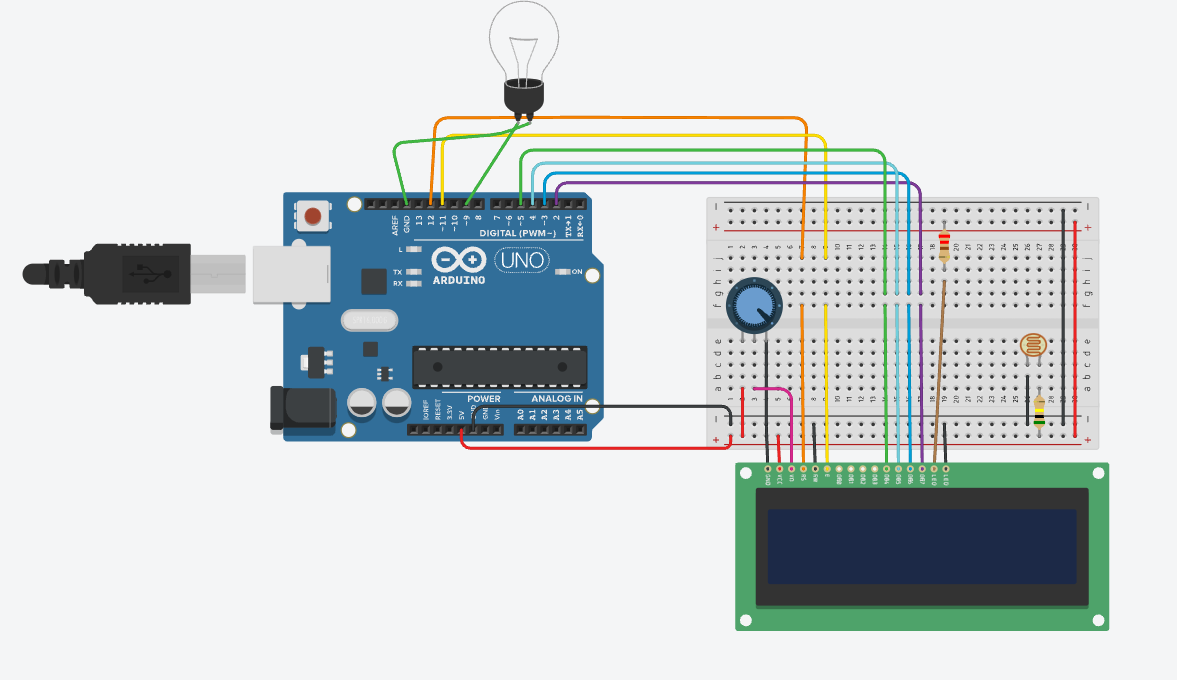
Mạch sử dụng quang trở để đo độ sáng

## Code

|  |
| --- |
| #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 8 : Bài 7 nhưng là câu b

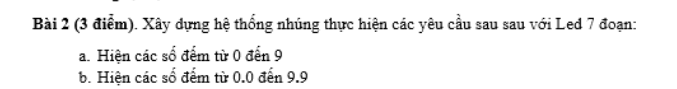
## Mạch



## 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:  pinMode(9,OUTPUT);  digitalWrite(9,LOW);  lcd.begin(16, 2);  // Print a message to the LCD.  Serial.begin(9600);  }  void loop() {  val = analogRead(LIGHT);  if (val < 100)  {  digitalWrite (9, HIGH);  delay(1000);  }  else  {  digitalWrite (9, LOW);  delay(1000);  }  // 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 cua den la :");  lcd.setCursor(0, 1);  lcd.print(val);  } |

# Bài 9 : Xây dựng hệ thống nhúng với Led 7 đoạn



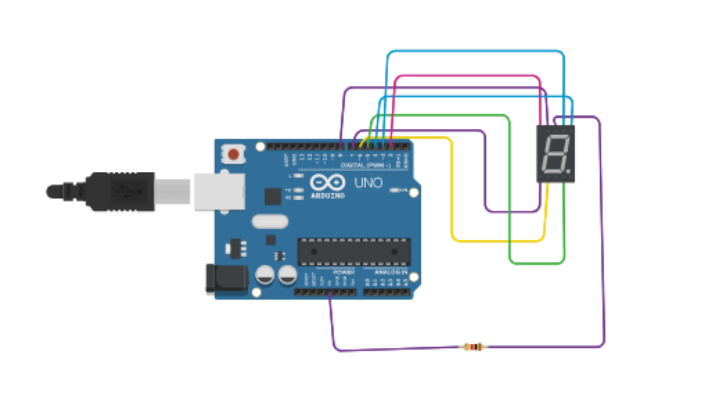
## Mô tả

LED 7 đoạn thường được dùng để hiển thị số, về cấu tạo LED 7 đoạn là các đèn LED được lắp theo thứ tự và hình ảnh giống số 8. Về phương thức hoạt động tương tự như LED bình thường.

Mỗi đoạn tương ứng một đèn LED nên cũng có cực âm và cực dương. Tuy nhiên có 2 loại đèn LED:

* Anode các chân LED có chung 1 chân cực dương.
* Cathode các chân LED có chung 1 chân cực âm.

## Sơ đồ mạch



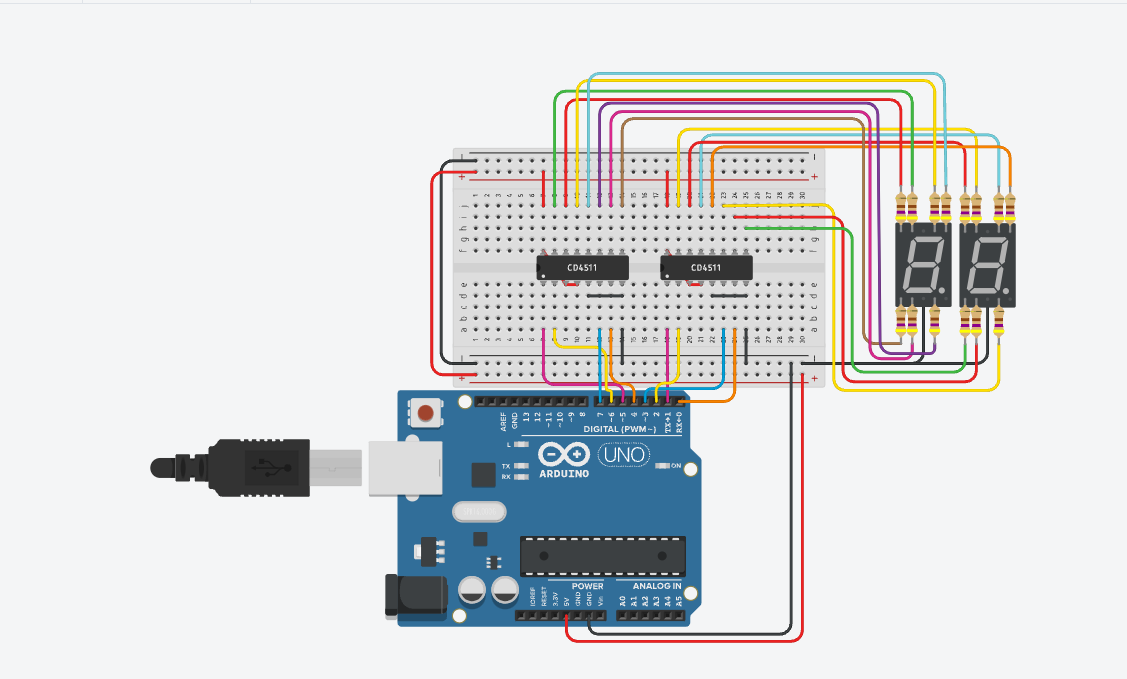
Mạch TinkerCad Led 7 đoạn hiện 1 số

## Code

|  |  |
| --- | --- |
| |  | | --- | | void setup() {  pinMode(2, OUTPUT);  pinMode(3, OUTPUT);  pinMode(4, OUTPUT);  pinMode(5, OUTPUT);  pinMode(6, OUTPUT);  pinMode(7, OUTPUT);  pinMode(8, OUTPUT);  digitalWrite(8, 0); // tắt chấm tròn “dp”  }  void loop() {  // Hiển thị số ‘0’  digitalWrite(2, 1);  digitalWrite(3, 0);  digitalWrite(4, 0);  digitalWrite(5, 0);  digitalWrite(6, 0);  digitalWrite(7, 0);  digitalWrite(8, 0);  delay(1000);  // Hiển thị số ‘1’  digitalWrite(2, 1);  digitalWrite(3, 1);  digitalWrite(4, 0);  digitalWrite(5, 0);  digitalWrite(6, 1);  digitalWrite(7, 1);  digitalWrite(8, 1);  delay(1000);  // Hiển thị số ‘2’  digitalWrite(2, 0);  digitalWrite(3, 0);  digitalWrite(4, 0);  digitalWrite(5, 1);  digitalWrite(6, 0);  digitalWrite(7, 0);  digitalWrite(8, 1);  delay(1000);  // Hiển thị số ‘3’  digitalWrite(2, 0);  digitalWrite(3, 0);  digitalWrite(4, 0);  digitalWrite(5, 0);  digitalWrite(6, 0);  digitalWrite(7, 1);  digitalWrite(8, 1);  delay(1000);  // Hiển thị số ‘4’  digitalWrite(2, 0);  digitalWrite(3, 1);  digitalWrite(4, 0);  digitalWrite(5, 0);  digitalWrite(6, 1);  digitalWrite(7, 1);  digitalWrite(8, 0);  delay(1000);  // Hiển thị số ‘5’  digitalWrite(2, 0);  digitalWrite(3, 0);  digitalWrite(4, 1);  digitalWrite(5, 0);  digitalWrite(6, 0);  digitalWrite(7, 1);  digitalWrite(8, 0);  delay(1000);  // Hiển thị số ‘6’  digitalWrite(2, 0);  digitalWrite(3, 1);  digitalWrite(4, 1);  digitalWrite(5, 0);  digitalWrite(6, 0);  digitalWrite(7, 0);  digitalWrite(8, 0);  delay(1000);  // Hiển thị số ‘7’  digitalWrite(2, 1);  digitalWrite(3, 0);  digitalWrite(4, 0);  digitalWrite(5, 0);  digitalWrite(6, 1);  digitalWrite(7, 1);  digitalWrite(8, 1);  delay(1000);  // Hiển thị số ‘8’  digitalWrite(2, 0);  digitalWrite(3, 0);  digitalWrite(4, 0);  digitalWrite(5, 0);  digitalWrite(6, 0);  digitalWrite(7, 0);  digitalWrite(8, 0);  delay(1000);  // Hiển thị số ‘9’  digitalWrite(2, 0);  digitalWrite(3, 0);  digitalWrite(4, 0);  digitalWrite(5, 0);  digitalWrite(6, 0);  digitalWrite(7, 1);  digitalWrite(8, 0);  delay(1000);  } | |

# Bài 10 : Bài 9 nhưng câu b(sử dụng 2 màn)

## Sơ đồ mạch



Led 7 đoạn hiển thị số từ 0.0 tới 9.9

## Code

|  |
| --- |
| 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);  } |

Mục lục

[Bài 1: Đèn 7 đoạn 1](#_Toc86955895)

[ Mô tả nguyên lý hoạt động 1](#_Toc86955896)

[ Sơ đồ mạch: 1](#_Toc86955897)

[ Linh kiện 1](#_Toc86955898)

[ CODE 1](#_Toc86955899)

[Bài 2 : 8 Led 3](#_Toc86955900)

[Mô tả : 3](#_Toc86955901)

[Sơ đồ mạch 4](#_Toc86955902)

[Linh Kiện 4](#_Toc86955903)

[CODE 4](#_Toc86955904)

[Bài 3 : Led RGB 5](#_Toc86955905)

[Mô tả : 5](#_Toc86955906)

[Sơ đồ mạch 6](#_Toc86955907)

[Linh kiện : 6](#_Toc86955908)

[Code 6](#_Toc86955909)

[Bài 4 : Cảm biến nhiệt độ TMP36 7](#_Toc86955910)

[Mô tả : 7](#_Toc86955911)

[Sơ đồ mạch 8](#_Toc86955912)

[Linh kiện 8](#_Toc86955913)

[Code 8](#_Toc86955914)

[Bài 5 : STM32 nhấp nháy led 9](#_Toc86955915)

[Sơ đồ mạch 9](#_Toc86955916)

[Linh kiện 9](#_Toc86955917)

[Code 10](#_Toc86955918)

[Bài 6: STM32F41 Led trái tim 11](#_Toc86955919)

[Mô tả : 11](#_Toc86955920)

[Sơ đồ mạch 11](#_Toc86955921)

[Linh kiện 11](#_Toc86955922)

[Code 11](#_Toc86955923)

[Bài 7: Xây dựng hệ thống nhúng để đo độ sáng Led 20](#_Toc86955924)

[Mô tả 20](#_Toc86955925)

[Sơ đồ mạch 20](#_Toc86955926)

[Code 20](#_Toc86955927)

[Bài 8 : Bài 7 nhưng là câu b 22](#_Toc86955928)

[Mạch 22](#_Toc86955929)

[Code 22](#_Toc86955930)

[Bài 9 : Xây dựng hệ thống nhúng với Led 7 đoạn 23](#_Toc86955931)

[Mô tả 23](#_Toc86955932)

[Sơ đồ mạch 24](#_Toc86955933)

[Code 24](#_Toc86955934)

[Bài 10 : Bài 9 nhưng câu b(sử dụng 2 màn) 29](#_Toc86955935)

[Sơ đồ mạch 29](#_Toc86955936)

[Code 29](#_Toc86955937)