Sinh viên thực hiện: Hồ Anh Nam

FIle báo cáo thực hành

**Mục lục**

[BÀI 1. NHÁY LED 6](#_Toc88264167)

[1.Mô tả 6](#_Toc88264168)

[2. Linh kiện 6](#_Toc88264169)

[3 . Sơ đồ thiết kế 6](#_Toc88264170)

[4 . Mã lệnh 6](#_Toc88264171)

[Bài 2. Nháy Led có nút bấm 7](#_Toc88264172)

[1. Mô tả 7](#_Toc88264173)

[2. Linh kiện 7](#_Toc88264174)

[3. Sơ đồ thiết kế 7](#_Toc88264175)

[4. Mã Lệnh 7](#_Toc88264176)

[Bài 3. LED sáng dần 8](#_Toc88264177)

[1. Mô tả 8](#_Toc88264178)

[2. Linh kiện 8](#_Toc88264179)

[3. Sơ đồ mạch 8](#_Toc88264180)

[4. Mã Lệnh 8](#_Toc88264181)

[Bài 4. LED sáng dần từ LED số 1 – 10 10](#_Toc88264182)

[1. Mô tả 10](#_Toc88264183)

[2. Linh kiện 10](#_Toc88264184)

[3. Sơ đồ mạch 10](#_Toc88264185)

[4. Mã lệnh 10](#_Toc88264186)

[Bài 5. LED RGB 11](#_Toc88264187)

[1. Mô tả 11](#_Toc88264188)

[2. Linh kiện 11](#_Toc88264189)

[3. Sơ đồ mạch 11](#_Toc88264190)

[4. Mã lệnh 12](#_Toc88264191)

[Bài 6. Cảm biến nhiệt độ điều khiển độ sáng của LED 13](#_Toc88264192)

[1. Mô tả 13](#_Toc88264193)

[2. Linh kiện 13](#_Toc88264194)

[3. Sơ đồ mạch 13](#_Toc88264195)

[4. Mã lệnh 14](#_Toc88264196)

[Bài 7. Điều chỉnh độ sáng của led qua chiết áp 14](#_Toc88264197)

[1. mô tả 14](#_Toc88264198)

[2. Linh kiện 14](#_Toc88264199)

[3. Sơ đồ mạch 15](#_Toc88264200)

[4. Mã lệnh 15](#_Toc88264201)

[Bài 8. Led 7 đoạn 15](#_Toc88264202)

[1. Mô tả 15](#_Toc88264203)

[2. Linh kiện 15](#_Toc88264204)

[3. Sơ đồ mạch 16](#_Toc88264205)

[4. Mã Lệnh 16](#_Toc88264206)

[Bài 9. Đọc nhiệt độ và độ ẩm 19](#_Toc88264207)

[1. Mô tả 19](#_Toc88264208)

[2. Linh kiện 19](#_Toc88264209)

[3. Sơ đồ mạch 20](#_Toc88264210)

[4. Mã lệnh 20](#_Toc88264211)

[Bài 10. Đèn giao thông 21](#_Toc88264212)

[1. Mô tả 21](#_Toc88264213)

[2. Linh kiện 21](#_Toc88264214)

[3. Sơ đồ mạch 21](#_Toc88264215)

[4 . Mã Lệnh 21](#_Toc88264216)

[Bài 11. Đo độ sáng bóng đèn 22](#_Toc88264217)

[1. Mô tả 22](#_Toc88264218)

[2. Linh kiện 22](#_Toc88264219)

[3. Sơ đồ mạch 22](#_Toc88264220)

[4. Mã lệnh 23](#_Toc88264221)

[Bài 12. Led 7 đoạn 0.0-9.9 24](#_Toc88264222)

[1. Mô tả 24](#_Toc88264223)

[2. Linh kiện 24](#_Toc88264224)

[3. Sơ đồ mạch 24](#_Toc88264225)

[4. Mã lệnh 24](#_Toc88264226)

[Bài 13. Cảm biến siêu âm + led 25](#_Toc88264227)

[1. Mô tả 25](#_Toc88264228)

[2. Linh kiện 25](#_Toc88264229)

[3. Sơ đồ mạch 26](#_Toc88264230)

[4. Mã lệnh 26](#_Toc88264231)

[Bài 14. Cảm biến vật cản và điều khiển động cơ 27](#_Toc88264232)

[1. Mô tả 27](#_Toc88264233)

[2. Linh kiện 27](#_Toc88264234)

[3. Sơ đồ mạch 27](#_Toc88264235)

[4. Mã Lệnh 28](#_Toc88264236)

[Bài 15. Led Matrix 8x8 30](#_Toc88264237)

[1. Mô tả 30](#_Toc88264238)

[2. Linh kiện 30](#_Toc88264239)

[3. Sơ đồ mạch 32](#_Toc88264240)

[4.Mã Lệnh 32](#_Toc88264241)

[Bài 16. Cảm biến chuyển động 33](#_Toc88264242)

[1. Mô tả 33](#_Toc88264243)

[2. Linh kiện 33](#_Toc88264244)

[3. Sơ đồ mạch 33](#_Toc88264245)

[4. Mã lệnh 34](#_Toc88264246)

[Bài 17. Sáng 8 Led bằng IC 74HC595 34](#_Toc88264247)

[1. Mô tả 34](#_Toc88264248)

[2. Linh kiện 34](#_Toc88264249)

[3. Sơ đồ mạch 34](#_Toc88264250)

[4.Mã lệnh 35](#_Toc88264251)

[Bài 18. Nháy 4 đèn Led thông qua STM32 35](#_Toc88264252)

[1. Mô tả 35](#_Toc88264253)

[2. Linh kiện 35](#_Toc88264254)

[3. Sơ đồ mạch 36](#_Toc88264255)

[4.Mã lệnh 36](#_Toc88264256)

[Bài 19. Ngắt ngoài trên STM32 37](#_Toc88264257)

[1. Mô tả 37](#_Toc88264258)

[2. Linh kiện 37](#_Toc88264259)

[3. Sơ đồ mạch 37](#_Toc88264260)

[4 . Mã lệnh 37](#_Toc88264261)

[Bài 20. Thay đổi màu led RGB bằng chiếc áp 38](#_Toc88264262)

[1. Mô tả 38](#_Toc88264263)

[2.Linh kiện 38](#_Toc88264264)

[3. Sơ đồ mạch 38](#_Toc88264265)

[4.Mã lệnh 38](#_Toc88264266)

[Bài 21. Điều khiển động cơ DC với IC L293D và Drduino 39](#_Toc88264267)

[1. Mô tả 39](#_Toc88264268)

[2. Linh kiện 39](#_Toc88264269)

[3. Sơ đồ mạch 39](#_Toc88264270)

[4.Mã lệnh 40](#_Toc88264271)

[Bài 21. Led trái tim bằng STM32 42](#_Toc88264272)

[1. Mô tả 42](#_Toc88264273)

[2. Linh kiện 42](#_Toc88264274)

[3. Sơ đồ mạch 43](#_Toc88264275)

[4.Mã lệnh 43](#_Toc88264276)

[Giới thiệu Appinventor. 48](#_Toc88264277)

[1.Giới thiệu 48](#_Toc88264278)

[2.Hướng dẫn sử dụng cơ bản 48](#_Toc88264279)

[Một số project Appinventor 50](#_Toc88264280)

[Nút bấm đổi màu 50](#_Toc88264281)

[DatePicker 53](#_Toc88264282)

[Hiển thị thông báo 55](#_Toc88264283)

[Caculator 57](#_Toc88264284)

[Cảm biến vật cản 59](#_Toc88264285)

[COUTING APP 61](#_Toc88264286)

[YOUR LOCATION 63](#_Toc88264287)

BÀI 1. NHÁY LED

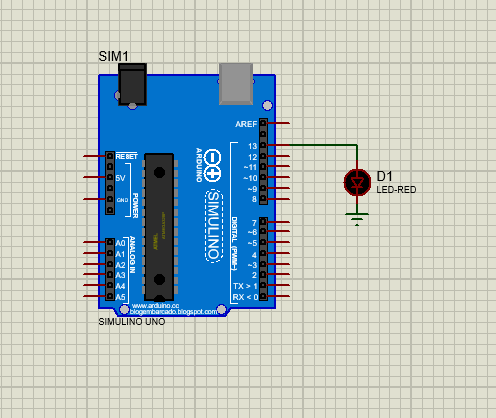
1.Mô tả

- Led tự động nháy đèn led sau khoảng thời gian 1 giây. Led được đấu vào cổng 13 của Board

2. Linh kiện

* Mạch arouno
* 1 bóng đèn

3 . Sơ đồ thiết kế



4 . Mã lệnh

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | | Int pin=13; | |  | void setup(){ | |  | pinMode(13,OUTPUT); | |  | } | |  |  | |  | void loop(){ | |  | digitalWrite(13,HIGH); | |  | delay(500); | |  | digitalWrite(13,LOW); | |  | delay(500); | |  | } | |

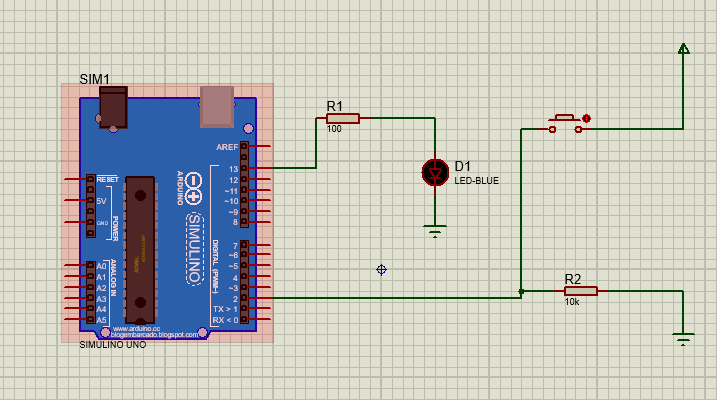
Bài 2. Nháy Led có nút bấm

1. Mô tả

* Led sáng sau khi nhấn vào nút bấm

1. Linh kiện

* Mạch arouno
* 1 điện trở
* 1 bóng đèn
* 1 bút bấm

1. Sơ đồ thiết kế
2. Mã Lệnh

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

Bài 3. LED sáng dần

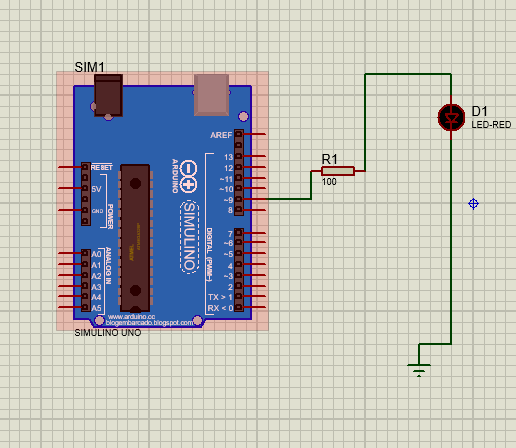
1. Mô tả

* Sau khi cấp nguồn điện, đèn led sáng dần rồi tắt và cứ thế lặp lại

1. Linh kiện

* Mạch arouno
* 1 điện trở
* 1 bóng đèn

1. Sơ đồ mạch



1. Mã Lệnh

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | | int brightness = 0 ; | |  |  | |  | void setup() { | |  | // put your setup code here, to run once: | |  | pinMode(9, OUTPUT); | |  | } | |  |  | |  | void loop() { | |  | // put your main code here, to run repeatedly: | |  | for( brightness = 0; brightness <= 255; brightness +=5) { | |  | analogWrite(9, brightness); | |  | delay(30); | |  | } | |  | for (brightness = 255; brightness >= 0 ; brightness -=5) { | |  | analogWrite(9, brightness); | |  | delay(30); | |  | } | |  |  | |  | } | |

Bài 4. LED sáng dần từ LED số 1 – 10

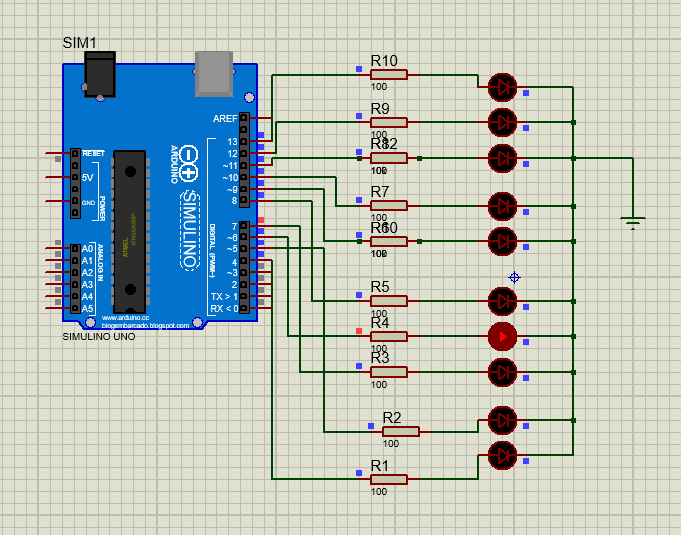
1. Mô tả

* Sau khi cấp nguồn điện, LED sẽ sáng lần lượt từ led số 1 đến led số 10 và ngược lại

1. Linh kiện

* Mạch arouno
* 10 điện trở
* 10 bóng đèn

1. Sơ đồ mạch



1. Mã lệnh

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | | byte ledPin[] = {4, 5, 6, 7, 8, 9, 10, 11, 12, 13}; | |  | int direction = 1; | |  | int currentLED = 0; | |  | void setup() { | |  | for (int x = 0; x < 10; x++) { | |  | pinMode(ledPin[x], OUTPUT); | |  | } | |  | } | |  | void loop() { | |  | for (int x = 0; x < 10; x++) { | |  | digitalWrite(ledPin[x], LOW); | |  | } | |  | digitalWrite(ledPin[currentLED], HIGH); | |  | currentLED += direction; | |  | if (currentLED == 9) { | |  | direction = -1; | |  | } | |  | if (currentLED == 0) { | |  | direction = 1; | |  | } | |  | delay(500); | |  | } | |

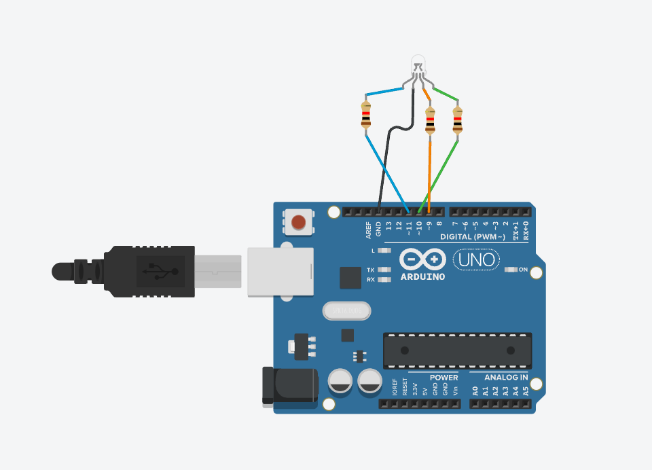
Bài 5. LED RGB

1. Mô tả

* Sau khi cấp nguồn điện chạy vào, đèn RGB sẽ sáng nhiều màu khác nhau liên tiếp

1. Linh kiện

* Led RGB
* Điện trở
* Mạch Arudro

1. Sơ đồ mạch
2. Mã lệnh

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | | void setup() { | |  | // put your setup code here, to run once: | |  | pinMode(3,OUTPUT); | |  | pinMode(2,OUTPUT); | |  | pinMode(1,OUTPUT); | |  | } | |  | const int B=0; | |  | const int R=1; | |  | const int G=2; | |  | const int Y=3; | |  | const int Bl=4; | |  | const int M=5; | |  | const int C=6; | |  | const int W=7; | |  |  | |  | void loop() { | |  | // put your setup code here, to run once: | |  | for(int i=B; i<=W; i++) | |  | { | |  | display(i); | |  | delay(500); | |  | } | |  | } | |  | void display(int colour\_number) | |  | { | |  | switch(colour\_number){ | |  | case B: digitalWrite(1,0); | |  | digitalWrite(2,0); | |  | digitalWrite(3,0); | |  | break; | |  | case R: digitalWrite(1,0); | |  | digitalWrite(2,0); | |  | digitalWrite(3,1); | |  | break; | |  | case G: digitalWrite(1,0); | |  | digitalWrite(2,1); | |  | digitalWrite(3,0); | |  | break; | |  | case Y: digitalWrite(1,0); | |  | digitalWrite(2,1); | |  | digitalWrite(3,1); | |  | break; | |  | case Bl: digitalWrite(1,1); | |  | digitalWrite(2,0); | |  | digitalWrite(3,0); | |  | break; | |  | case M: digitalWrite(1,1); | |  | digitalWrite(2,0); | |  | digitalWrite(3,1); | |  | break; | |  | case C: digitalWrite(1,1); | |  | digitalWrite(2,1); | |  | digitalWrite(3,0); | |  | break; | |  | case W: digitalWrite(1,1); | |  | digitalWrite(2,1); | |  | digitalWrite(3,1); | |  | break; | |  |  | |  | } | |  | } | |

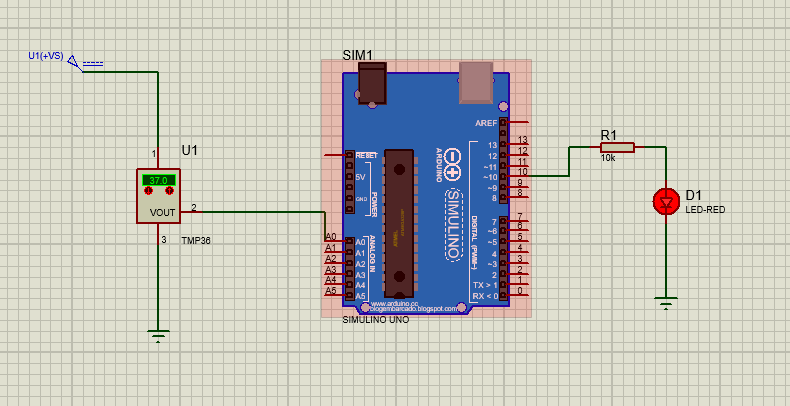
Bài 6. Cảm biến nhiệt độ điều khiển độ sáng của LED

1. Mô tả

* Sau khi cấp nguồn điện, ta điều chỉnh nhiệt độ
* Nếu nhiệt độ trên 36 độ thì đèn sẽ sáng ngược lại đèn tắt

1. Linh kiện

* Mạch Arđuo
* Bóng đèn
* Cảm biến nhiệt độ-TMP36: TMP36 là một loại cảm biến nhiệt độ. Nó có khả năng đo từ -40 đến 125 độ C
* Điện trở

1. Sơ đồ mạch

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| #define SENSOR\_PIN A0 //Chân gắn cảm biến  float voltage = 0; // Thiết lập một số giá trị ban đầu  float sensor = 0; //Giá trị cảm biến  float celsius = 0; // Độ C   |  |  | | --- | --- | | void setup() { | | |  | // put your setup code here, to run once: | |  | pinMode(10,OUTPUT); | |  | |  |  | |  | } | |  |  | |  | void loop() { | |  | // put your main code here, to run repeatedly: | |  | 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 | |  | |  | |  | if( celsius >=36)digitalWrite(10, HIGH); | |  | else digitalWrite(10, LOW); | |  | delay(100); | |  |  | |  | } | |

1. Mã lệnh

Bài 7. Điều chỉnh độ sáng của led qua chiết áp

1. Mô tả

- Ở ví dụ này ta dựa trên giá trị của biến trở mà tùy chỉnh độ sáng của đèn

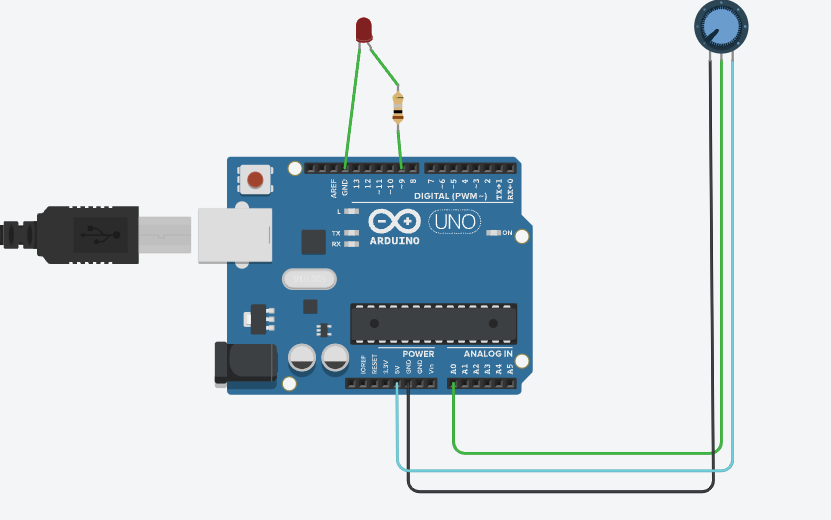
2. Linh kiện

- Mạch Arudo

- Đèn

- Biến trở

3. Sơ đồ mạch



4. Mã lệnh

|  |
| --- |
| int x = 0;  void setup() {  pinMode(A0, INPUT);  pinMode(9, OUTPUT);  }  void loop(){  //doc gia tri bien tro ghi vao bien x  x = analogRead(A0);  int brightness = map(x,0,1023,0,255);//quy doi theo do sang  //sang led theo bien tren  analogWrite(9, brightness);  } |

Bài 8. Led 7 đoạn

1. Mô tả

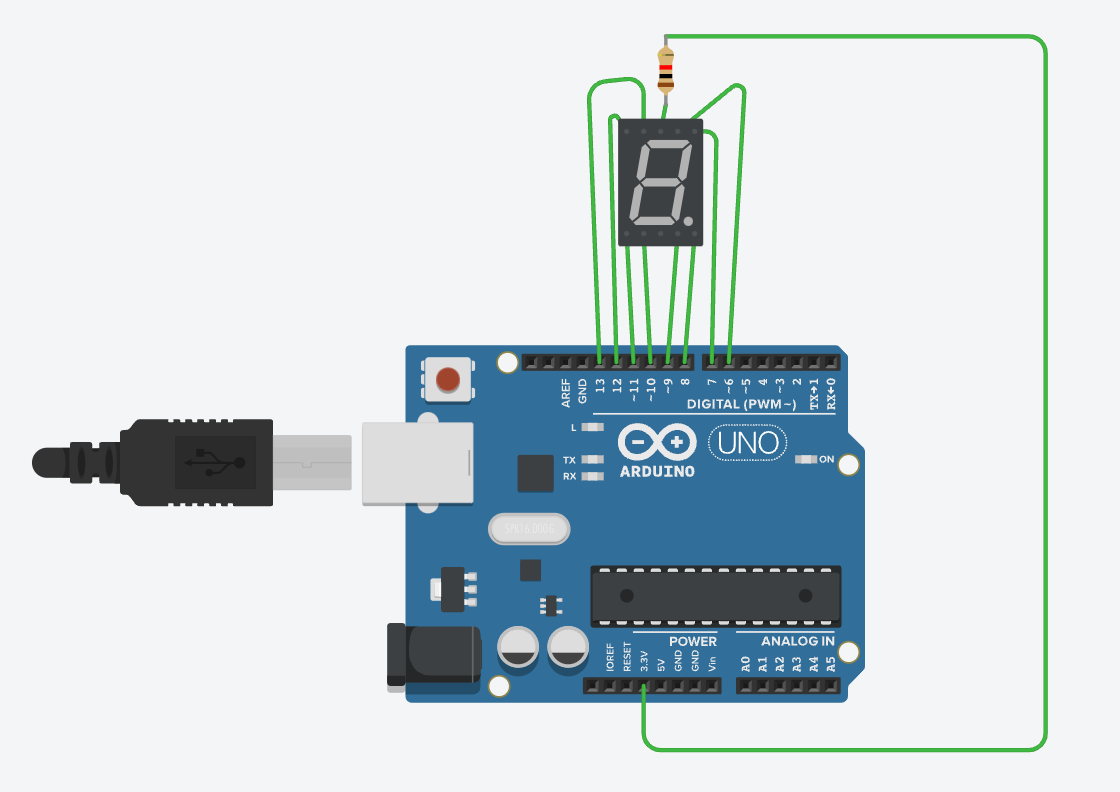
- Led sáng theo số theo thứ tự 1- 9

2. Linh kiện

- Mạch Aruno

- Led 7 đoạn

3. Sơ đồ mạch



4. Mã Lệnh

|  |
| --- |
| int x=1000;  void setup()  {  pinMode(6, OUTPUT);  pinMode(7, OUTPUT);  pinMode(8, OUTPUT);  pinMode(9, OUTPUT);  pinMode(10, OUTPUT);  pinMode(11, OUTPUT);  pinMode(12, OUTPUT);  pinMode(13, OUTPUT);  }    void loop(){  khong();  delay(x);  mot();  delay(x);  hai();  delay(x);  ba();  delay(x);  bon();  delay(x);  nam();  delay(x);  sau();  delay(x);  bay();  delay(x);  tam();  delay(x);  chin();  delay(x);  }  void khong (){  digitalWrite (6,LOW);  digitalWrite (7,LOW);  digitalWrite (8,LOW);  digitalWrite (9,LOW);  digitalWrite (10,LOW);  digitalWrite (11,LOW);  digitalWrite (12,HIGH);  digitalWrite (13,LOW);  }  void mot (){  digitalWrite (6,HIGH);  digitalWrite (7,LOW);  digitalWrite (8,LOW);  digitalWrite (9,LOW);  digitalWrite (10,HIGH);  digitalWrite (11,HIGH);  digitalWrite (12,HIGH);  digitalWrite (13,HIGH);  }  void hai (){  digitalWrite (6,LOW);  digitalWrite (7,LOW);  digitalWrite (8,LOW);  digitalWrite (9,HIGH);  digitalWrite (10,LOW);  digitalWrite (11,LOW);  digitalWrite (12,LOW);  digitalWrite (13,HIGH);  }  void ba (){  digitalWrite (6,LOW);  digitalWrite (7,LOW);  digitalWrite (8,LOW);  digitalWrite (9,LOW);  digitalWrite (10,LOW);  digitalWrite (11,HIGH);  digitalWrite (12,LOW);  digitalWrite (13,HIGH);  }  void bon (){  digitalWrite (6,HIGH);  digitalWrite (7,LOW);  digitalWrite (8,LOW);  digitalWrite (9,LOW);  digitalWrite (10,HIGH);  digitalWrite (11,HIGH);  digitalWrite (12,LOW);  digitalWrite (13,LOW);  }  void nam (){  digitalWrite (6,LOW);  digitalWrite (7,HIGH);  digitalWrite (8,LOW);  digitalWrite (9,LOW);  digitalWrite (10,LOW);  digitalWrite (11,HIGH);  digitalWrite (12,LOW);  digitalWrite (13,LOW);  }  void sau (){  digitalWrite (6,LOW);  digitalWrite (7,HIGH);  digitalWrite (8,LOW);  digitalWrite (9,LOW);  digitalWrite (10,LOW);  digitalWrite (11,LOW);  digitalWrite (12,LOW);  digitalWrite (13,LOW);  }  void bay (){  digitalWrite (6,LOW);  digitalWrite (7,LOW);  digitalWrite (8,LOW);  digitalWrite (9,LOW);  digitalWrite (10,HIGH);  digitalWrite (11,HIGH);  digitalWrite (12,HIGH);  digitalWrite (13,HIGH);  }  void tam (){  digitalWrite (6,LOW);  digitalWrite (7,LOW);  digitalWrite (8,LOW);  digitalWrite (9,LOW);  digitalWrite (10,LOW);  digitalWrite (11,LOW);  digitalWrite (12,LOW);  digitalWrite (13,LOW);  }  void chin (){  digitalWrite (6,LOW);  digitalWrite (7,LOW);  digitalWrite (8,LOW);  digitalWrite (9,LOW);  digitalWrite (10,HIGH);  digitalWrite (11,HIGH);  digitalWrite (12,LOW);  digitalWrite (13,LOW);  } |

Bài 9. Đọc nhiệt độ và độ ẩm

1. Mô tả

- Khi nhiệt độ > 30 thì đèn đỏ sáng

- Độ ẩm < 40 đèn vàng sáng

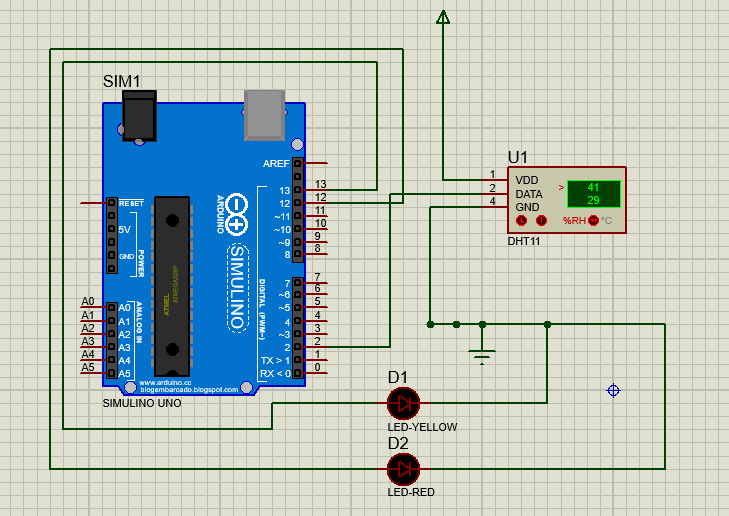
2. Linh kiện

- Mạch Arduno

- DHT11: Cảm biến nhiệt độ và độ ẩm, Cảm biến được tích hợp bộ tiền xử lý tín hiệu giúp dữ liệu nhận về được chính xác mà không cần phải qua bất kỳ tính toán nào.

- 2 led vàng và đỏ

1. Sơ đồ mạch



1. Mã lệnh

|  |
| --- |
| #include "DHT.h"  #define dht\_1 2  #define DHTTYPE DHT11  DHT dht(dht\_1, DHTTYPE);  void setup() {  dht.begin();  }  void loop() {  float doam = dht.readHumidity();  float nhietdo = dht.readTemperature();  if(nhietdo > 30) {  digitalWrite(12,HIGH);  }  else  digitalWrite(12,LOW);  if(doam <40 ) {  digitalWrite(13,HIGH);  }  else  digitalWrite(13,LOW);  } |

Bài 10. Đèn giao thông

1. Mô tả

- Hệ thông đèn giao thông mỗi đèn sẽ sáng theo 1 thời gian nhất định được sử dụng phổ biến để điều khiển giao thông

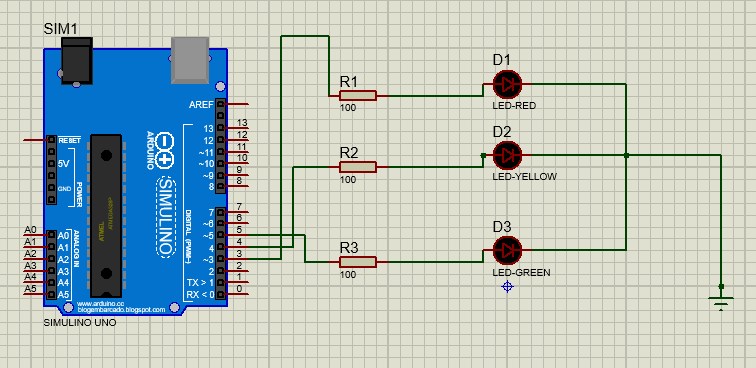
2. Linh kiện

- Mạch Arduno

- Điện trở

- Led

3. Sơ đồ mạch



4 . Mã Lệnh

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | | void setup() { | |  | // put your setup code here, to run once: | |  | pinMode(3, OUTPUT); | |  | pinMode(4, OUTPUT); | |  | pinMode(5, OUTPUT); | |  | } | |  |  | |  | void loop() { | |  | // put your main code here, to run repeatedly: | |  | int x = 2; | |  | for (int i = 1; i < x ; i++) { | |  | digitalWrite(3, HIGH); | |  | delay(25000); | |  | digitalWrite(3, LOW); | |  | digitalWrite(4, HIGH); | |  | delay(10000); | |  | digitalWrite(4, LOW); | |  | digitalWrite(5, HIGH); | |  | delay(30000); | |  | digitalWrite(5, LOW); | |  | i = 0; | |  |  | |

Bài 11. Đo độ sáng bóng đèn

1. Mô tả

- Ghi giá trị độ sáng của bóng đèn hiển thị lên LCD

2. Linh kiện

- Mạch

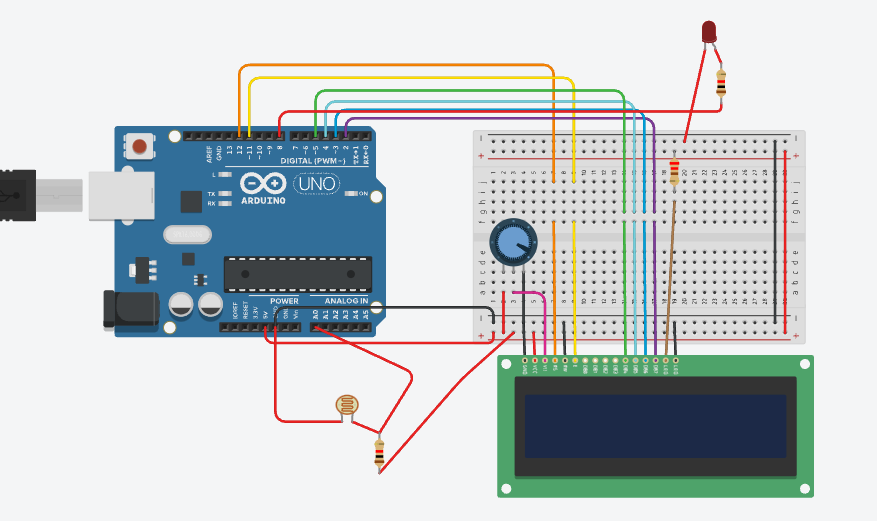
- Quang trở: cảm biến đọc giá trị độ sáng của bóng đèn

- LCD: 1 loại dạng màn hình giúp hiển thị thông tin chúng ta muốn

- Biến áp: Điều chính dòng điện

- LED

3. Sơ đồ mạch



4. Mã lệnh

|  |
| --- |
| // include the library code:  #include <LiquidCrystal.h>  // initialize the library with the numbers of the interface pins  LiquidCrystal lcd(12, 11, 5, 4, 3, 2);  int LED = 8;  int LIGHT = 0;  int val =0;  void setup() {  // set up the LCD's number of columns and rows:  lcd.begin(16, 2);  lcd.setCursor(4,0);  pinMode (LED, OUTPUT);  Serial.begin(9600);  // Print a message to the LCD.  lcd.print("DO SANG");  }  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, 1);  val = analogRead(LIGHT);  Serial.println(val);  if (val < 680)  {  digitalWrite (8, LOW);  }  else  {  digitalWrite (8, HIGH);  }  delay(10);  // print the number of seconds since reset:  lcd.print(val);  delay(1000);  } |

Bài 12. Led 7 đoạn 0.0-9.9

1. Mô tả

- Sau khi cấp dòng điện 2 led hiển thị kết hợp 0.0-9.9

2. Linh kiện

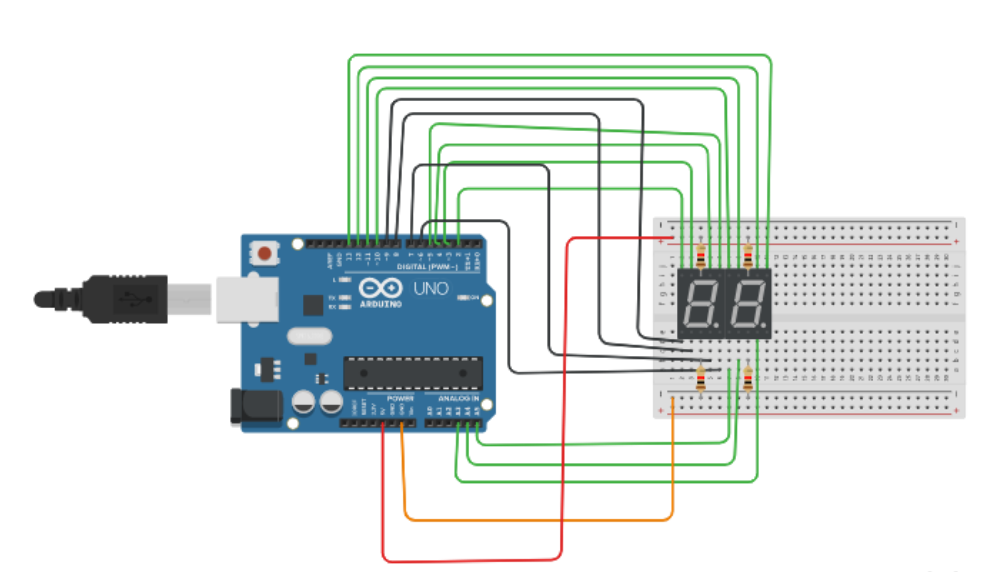
- Mạch

- Led 7 đoạn

- Điện trở

- Boarch mạch

3. Sơ đồ mạch



4. Mã lệnh

|  |
| --- |
| int display1[] = {4, 5, 7, 8, 9, 3, 2}; //man hinh 1  int display2[] = {12, 13, A3, A4, A5, 11, 10}; //man hinh 2  int nums[10][7] = {  {0, 0, 0, 0, 0, 0, 1}, // 0  {1, 0, 0, 1, 1, 1, 1}, // 1  {0, 0, 1, 0, 0, 1, 0}, // 2  {0, 0, 0, 0, 1, 1, 0}, // 3  {1, 0, 0, 1, 1, 0, 0}, // 4  {0, 1, 0, 0, 1, 0, 0}, // 5  {0, 1, 0, 0, 0, 0, 0}, // 6  {0, 0, 0, 1, 1, 1, 1}, // 7  {0, 0, 0, 0, 0, 0, 0}, // 8  {0, 0, 0, 0, 1, 0, 0} // 9  }; //thiet lap mang 2 chieu tuong ung voi cac so  const byte dot = 6;  const int showTime = 500; // time delay  void setup() {  pinMode(dot, OUTPUT); //cap dien cho mach  for(byte i = 0; i < 8; i++){  pinMode(display1[i], OUTPUT); //man 1  pinMode(display2[i], OUTPUT); //man 2  }  }  void loop() {  for(byte i = 0; i < 10; i++) {  displayNum(display1, nums[i]); //bat dau` tu 0.    for(byte j = 0; j < 10; j++) {  displayNum(display2, nums[j]); // chay tu 0-9 roi quay lai tang gia tri cua display ++  delay(showTime);  }  }  }  void displayNum(int display[], int num[]) { //cap dien cho cac dau da dc thiet lap san  for(byte i = 0; i < 8; i++)  digitalWrite(display[i], num[i]);  } |

Bài 13. Cảm biến siêu âm + led

1. Mô tả

- Cảm biến khoảng cách khi phía trước có vật cản thì đèn sẽ sáng lên báo hiệu

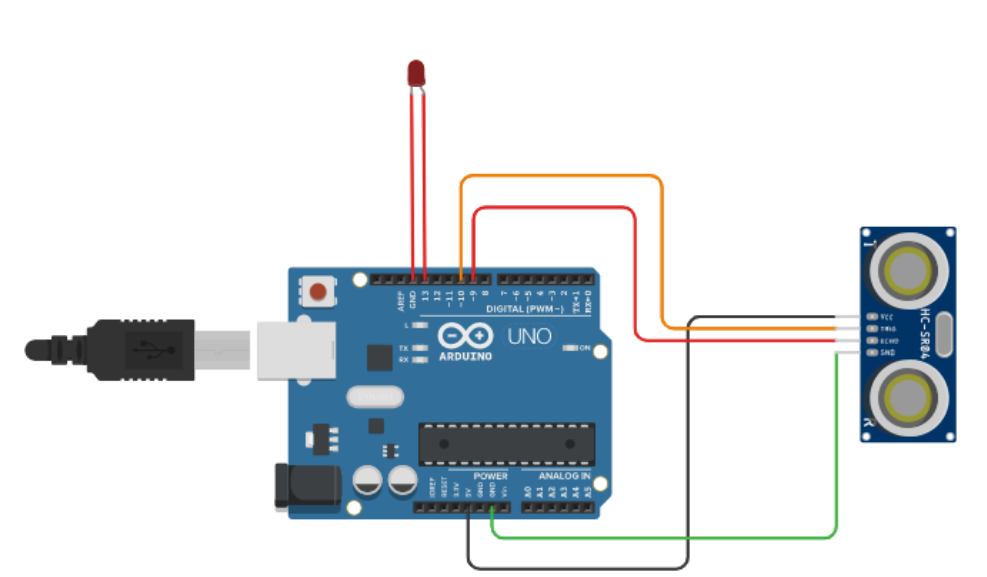
2. Linh kiện

- mạch Arudo

- LED

- Cảm biển HCR04: Dùng để cảm biến khoảng cách

3. Sơ đồ mạch



4. Mã lệnh

|  |
| --- |
| int trigPin = 10; // Trigger  int echoPin = 9; // Echo  long duration, cm, inches;  void setup()  {  pinMode(13, OUTPUT);  pinMode(trigPin,OUTPUT); // chân trig sẽ phát tín hiệ  pinMode(trigPin,INPUT); // chân echo sẽ nhận tín hiệ  }    void loop()  {  // Đầu tiên ta đặt chân trigger lên trạng thái HIGH trong 10ms hoặc hơn.  // Sau đó đặt về LOW  digitalWrite(trigPin, LOW);  delayMicroseconds(5);  digitalWrite(trigPin, HIGH);  delayMicroseconds(10);  digitalWrite(trigPin, LOW);    //Đọc giá trị từ cảm biến  // duration là thời gian được đó từ lúc phóng siêu âm ra đến lúc thu lại siêu âm phản hồi.  pinMode(echoPin, INPUT);  duration = pulseIn(echoPin, HIGH);    // Chuyển thời gian sang khoảng cách, Vì sóng siêu âm đi đến đối tượng và đi về là nhân đôi khoảng cách nên ta chia 2. 29.1 là vận tộc siêu âm tính theo cm, là 74 là tính theo inch. Ta lấy thời gian chia cho vận tốc là ra quảng đường  cm = (duration/2) / 29.1;  inches = (duration/2) / 74;  if(cm >0)  digitalWrite(13,HIGH);  else digitalWrite(13,LOW);    } |

Bài 14. Cảm biến vật cản và điều khiển động cơ

1. Mô tả

- Một xe ba bánh, trong đó có một động cơ điểu khiển 2 bánh trước, 2 động cơ bánh sau. Khi gặp vật cản:

+ Nếu xe chạy chậm, dừng xe, lùi xe ngược lại trong 10s

+ Nếu xe chạy nhanh, đánh tay lái sang trái 20 độ và giảm tốc độ Z+10 đơn vị rồi dừng hẳn khoảng thời gian 10s

2. Linh kiện

- Mạch

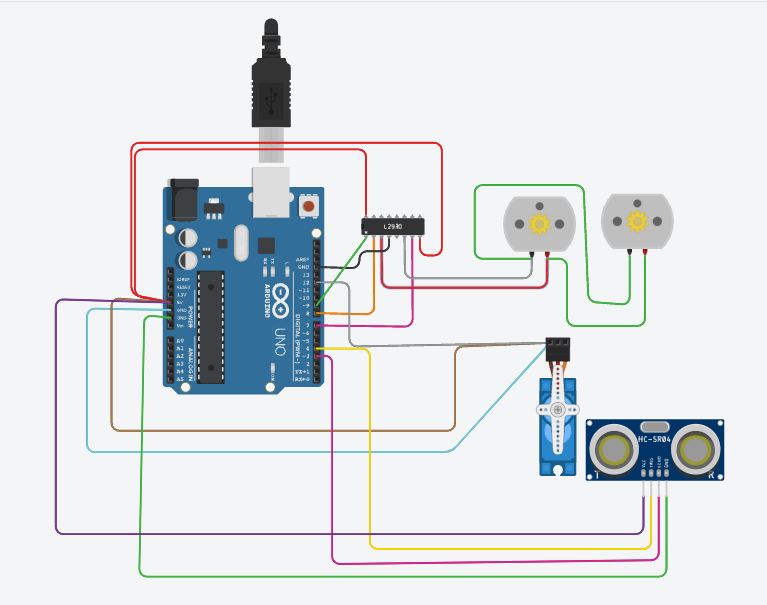
- L293D: một board mở rộng của arduino

- 2 động cơ

- 1 cảm biến siêu âm HCR04: Cảm biến khoảng cách

- 1 động cơ xoay

3. Sơ đồ mạch



4. Mã Lệnh

|  |
| --- |
| #include <Servo.h>  int speedControl =9;  int speed;  int inp1 =8;  int inp2 =7;  int trigPin = 11; // Trigger  int echoPin = 12; // Echo  long duration, cm, inches;  int pos = 0;  Servo servo\_12;  void setup()  {  pinMode(speedControl, OUTPUT);  pinMode(inp1, OUTPUT);    pinMode(trigPin, OUTPUT);  pinMode(echoPin, INPUT);    servo\_12.attach(12);  }  void loop()  {  speed = 130; // dat gia tri toc do  // Quay tới  digitalWrite(inp1,HIGH);  digitalWrite(inp2,LOW);  //với Speed  analogWrite(speedControl,speed);  // Đầu tiên ta đặt chân trigger lên trạng thái HIGH trong 10ms hoặc hơn.  // Sau đó đặt về LOW  digitalWrite(trigPin, LOW);  delayMicroseconds(5);  digitalWrite(trigPin, HIGH);  delayMicroseconds(10);  digitalWrite(trigPin, LOW);    //Đọc giá trị từ cảm biến  // duration là thời gian được đó từ lúc phóng siêu âm ra đến lúc thu lại siêu âm phản hồi.  pinMode(echoPin, INPUT);  duration = pulseIn(echoPin, HIGH);    // Chuyển thời gian sang khoảng cách, Vì sóng siêu âm đi đến đối tượng và đi về là nhân đôi khoảng cách nên ta chia 2. 29.1 là vận tộc siêu âm tính theo cm, là 74 là tính theo inch. Ta lấy thời gian chia cho vận tốc là ra quảng đường  cm = (duration/2) / 29.1;  inches = (duration/2) / 74;  delay(1000);  if(inches >0 && speed<100){  digitalWrite(inp1,LOW); //dung xe  digitalWrite(inp2,LOW);  delay(2000);  digitalWrite(inp1,LOW); //lui xe  digitalWrite(inp2,HIGH);  delay(2000);  digitalWrite(inp1,HIGH);  digitalWrite(inp2,LOW);  delay(5000);// tiep tuc chay  }  if(inches >0 && speed>100){  for (pos = 0; pos <= 180; pos += 1)  {  servo\_12.write(20);  delay(150);    }  analogWrite(speedControl,speed - 10); //giam toc do cua xe  digitalWrite(inp1,LOW); //dung xe  digitalWrite(inp2,LOW);  delay(5000);  digitalWrite(inp1,HIGH);  digitalWrite(inp2,LOW); //tiep tuc chay  }  } |

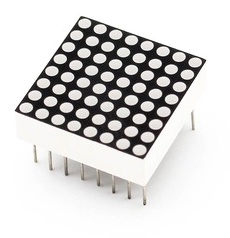
Bài 15. Led Matrix 8x8

1. Mô tả

- Hệ thống được thiết kế thực hiện việc lập trình điều đèn Led Matrix 8x8. Ở bài này ta sẽ tuần tự cho 8 đèn ở mỗi cột nháy 1 lần. 74HC595 ký hiệu là U2 sẽ là hàng, còn 74HC595 ký hiệu là U1 sẽ là cột.

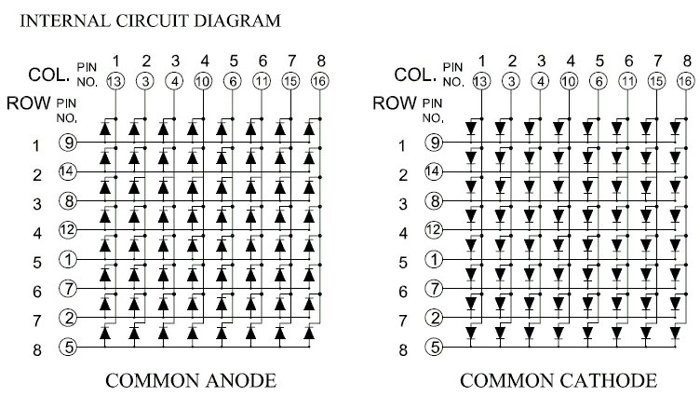
2. Linh kiện

* 1 mạch Arduino Uno
* 2 mạch IC 74HC595
* 1 bảng mạch Led Matrix 8x8 Red
* Sơ lược: Led matrix 8x8 dùng để tạo hiệu ứng hình ảnh trên Matrix, Vd: tạo chữ, số, tạo hiệu ứng chuyển động để làm biển chỉ dẫn, làm bảng quảng cáo,…
* Sơ lược: Led matrix 8x8 dùng để tạo hiệu ứng hình ảnh trên Matrix, Vd: tạo chữ, số, tạo hiệu ứng chuyển động để làm biển chỉ dẫn, làm bảng quảng cáo,…



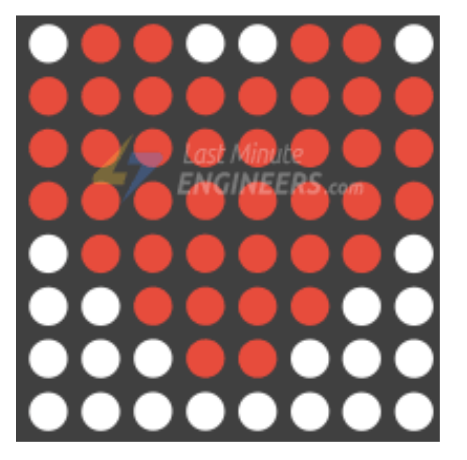
Hình 29. Hình ảnh thức tế Led Matrix 8x8

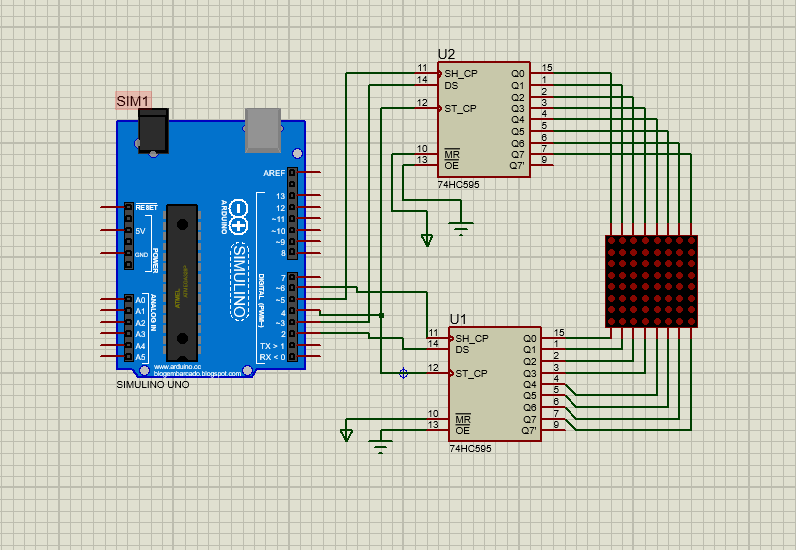
* Led matrix 8x8 đơn màu thông thường có 16 chân, 8 cho mỗi hàng và 8 cho mỗi cột. Lý do cho tất cả các hàng và cột được nối dây với nhau là để giảm số lượng chân cần thiết. Nếu không nối như vậy thì Led matrix 8x8 sẽ yêu cầu 65 chân, một chân cho mỗi đèn LED và một chân cho đầu nối cực dương hoặc cực âm chung. Vì vậy bằng cách nối dây các hàng và cột với nhau, chỉ cần 16 chân.
* Phân loại: Có 2 loại Matrix là Anode chung và Kathode chung. Ta có thể thấy 2 loại này giống nhau, điểm khác duy nhất là cách cấp nguồn cho 2 loại này là ngược nhau, ta có thể dễ dàng thấy điều này ở hình minh họa bên dưới.



Hình 30. Hình minh họa

* Mỗi cột được kích hoạt trong một khoảng thời gian rất ngắn và đồng thời các đèn LED trên cột đó sáng lên bằng cách đặt địa chỉ cho hàng tương ứng. Các cột được chuyển đổi quá nhanh (hàng trăm hoặc hàng nghìn lần một giây) đến mức mắt người vẫn nhận thấy màn hình được chiếu sáng đầy đủ. Do đó, chỉ có tối đa tám đèn LED được thắp sáng cùng một lúc.



3. Sơ đồ mạch 

4.Mã Lệnh

|  |
| --- |
| #define DS\_cot 2 //Chân data của cột kết nối chân số 5 của Arduino  #define DS\_hang 3 //Chân data của hàng kết nối chân số 3 của Arduino  #define SH\_CP\_hang 5 //Chân clock của hàng được kết nối chân số 7 của Arduino  #define SH\_CP\_cot 6 //Chân clock của cột được kết nối chân số 6 của Arduino  #define ST\_CP 4 // Khai báo chân latch được kết nối với chân số 4 của Arduino  byte cot = 0b10000000;  void setup() {  pinMode (ST\_CP,OUTPUT); //RCLK  pinMode (DS\_hang,OUTPUT);  pinMode (SH\_CP\_hang,OUTPUT);  pinMode (SH\_CP\_cot,OUTPUT);  pinMode (DS\_cot,OUTPUT);  digitalWrite(ST\_CP,LOW);  //Ta sử dụng hàm vào ra nâng cao shiftOut:  shiftOut(DS\_hang,SH\_CP\_hang,LSBFIRST,0b00000000); //hang 8 - hang 1  digitalWrite(ST\_CP,HIGH);  }  void loop() {  for(int i=0; i< 8;i++){  digitalWrite(ST\_CP,LOW);  //Ta sử dụng hàm vào ra nâng cao shiftOut:  shiftOut(DS\_cot,SH\_CP\_cot,LSBFIRST,cot >> i); //cot 8 - cot 1  digitalWrite(ST\_CP,HIGH);  delay(1000);  }  } |

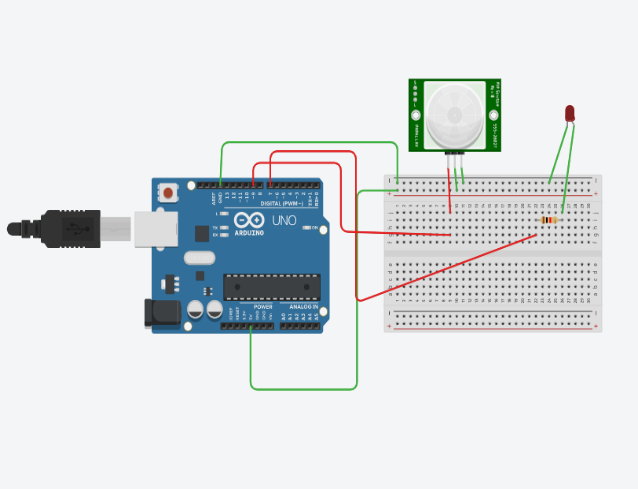
Bài 16. Cảm biến chuyển động

1. Mô tả

- Hệ thống báo hiệu khi có chuyện động, được ứng dụng rộng rãi trong các động cơ hoặc các thiết thị

2. Linh kiện

- Cảm biến PIR senor:  là chữ viết tắt của **Passive InfraRed sensor** (PIR sensor), tức là bộ cảm biến thụ động dùng nguồn kích thích là tia hồng ngoại. Tia hồng ngoại (IR) chính là các tia nhiệt phát ra từ các vật thể nóng. Trong các cơ thể sống, trong chúng ta luôn có thân nhiệt (thông thường là ở 37 độ C), và từ cơ thể chúng ta sẽ luôn phát ra các tia nhiệt, hay còn gọi là các tia hồng ngoại, người ta sẽ dùng một tế bào điện để chuyển đổi tia nhiệt ra dạng tín hiệu điện và nhờ đó mà có thể làm ra cảm biến phát hiện các vật thể nóng đang chuyển động. Cảm biến này gọi là thụ động vì nó không dùng nguồn nhiệt tự phát (làm nguồn tích cực, hay chủ động) mà chỉ phụ thuộc vào các nguồn tha nhiệt, đó là thân nhiệt của các thực thể khác, như con người con vật…

3. Sơ đồ mạch

4. Mã lệnh

|  |
| --- |
| int sensor =9;  int state;  int led = 7;  void setup()  {  pinMode(sensor, INPUT);  pinMode(led, OUTPUT);  }  void loop()  {  state = digitalRead(sensor);  if(state ==HIGH) digitalWrite(led, HIGH);  else digitalWrite(led, LOW);  } |

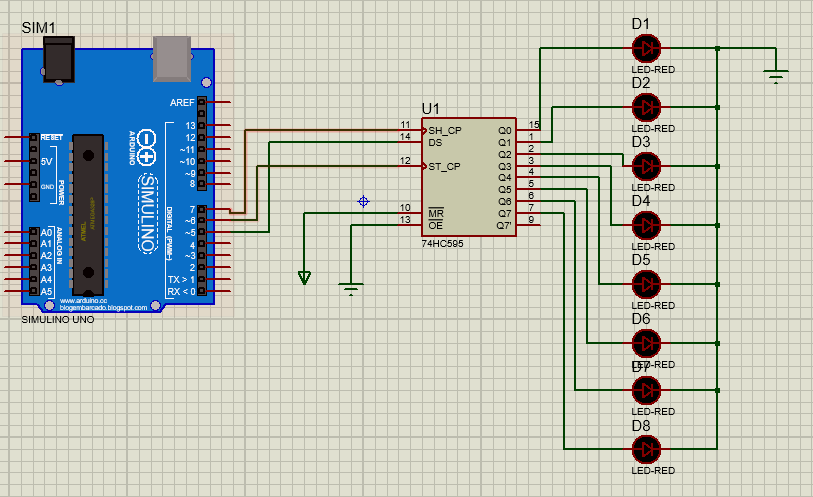
Bài 17. Sáng 8 Led bằng IC 74HC595

1. Mô tả

- Hệ thống được thiết kế thực hiện việc lập trình điều kiển bật/tắt đèn 8 Led bằng IC 74HC595, được kết nối với mạch Arduino qua các cổng 5, 6, 7.

2. Linh kiện

* 8 đèn Led
* 1 mạch Arduino Uno
* 1 mạch IC 74HC595

3. Sơ đồ mạch

4.Mã lệnh

|  |
| --- |
| #define \_latch 6 // Khai báo chân latch được kết nối với chân số 6 của Arduino  #define \_clock 7 //Khai báo chân clock được kết nối với chân số 7 của Arduino  #define \_data 5 //Khai báo chân data được kết nối với chân số 5 của Arduino  void setup()  {  pinMode(\_latch, OUTPUT);  pinMode(\_clock, OUTPUT);  pinMode(\_data, OUTPUT);  }  void loop()  {  for (int i=0; i< 256;i++){  digitalWrite(\_latch,LOW);  //Ta sử dụng hàm vào ra nâng cao:  //shiftOut(chân\_xuất\_data, chân\_clock, LSBFIRST, dữ\_liệu);  shiftOut(\_data, \_clock,LSBFIRST, i);  //LSBFIRST (bít thứ tự nhỏ trước) còn MSBFIRST là ngược lại  digitalWrite( \_latch,HIGH);  delay(500);  }  } |

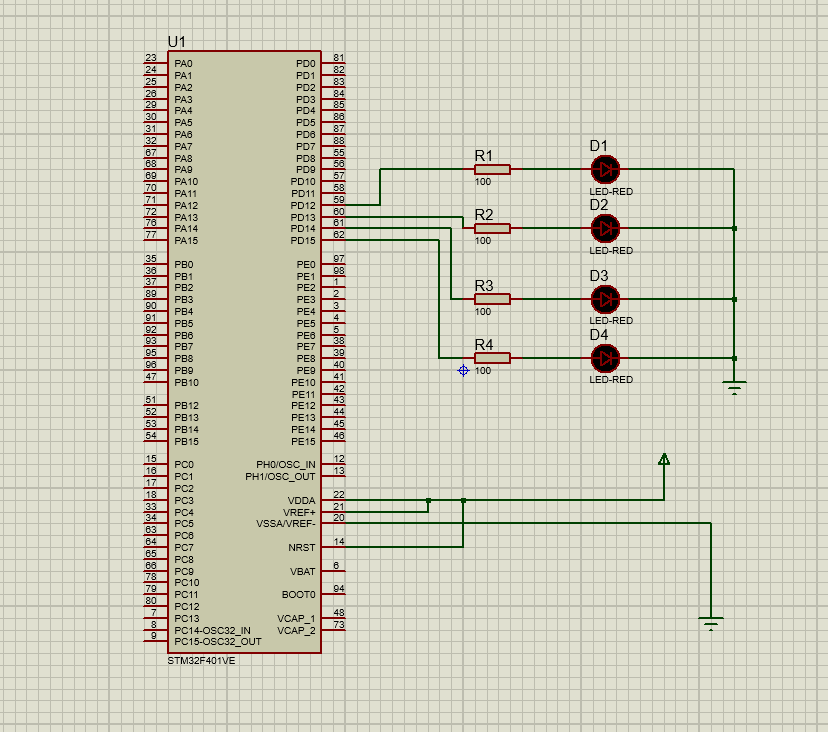
Bài 18. Nháy 4 đèn Led thông qua STM32

1. Mô tả

Hệ thống được thiết kế thực hiện việc lập trình điều kiển 4 đèn Led tự động bật/tắt sau 1 giây thông qua 4 chân PD12, PD13, PD14 và PD15 của STM32F401VE.

2. Linh kiện

* 4 điện trở: 100 Ω
* 4 đèn Led
* 1 STM32F401VE

3. Sơ đồ mạch

4.Mã lệnh

|  |
| --- |
| int main(void)  {  HAL\_Init();  SystemClock\_Config();  MX\_GPIO\_Init();  while (1)  { //Hàm đảo trạng thái các chân 12,13,14,15  HAL\_GPIO\_TogglePin(GPIOD,GPIO\_PIN\_12|GPIO\_PIN\_13|GPIO\_PIN\_14|GPIO\_PIN\_15);  HAL\_Delay(1000); //delay 1s  }  } |

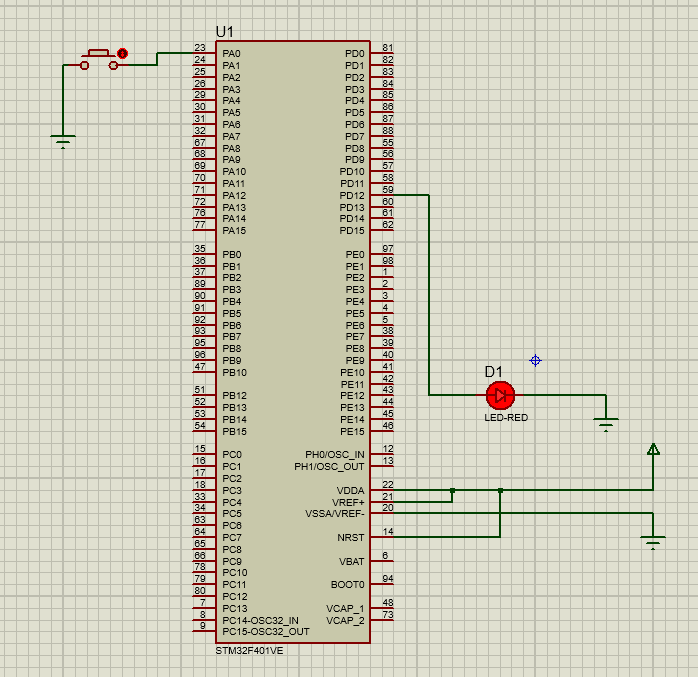
Bài 19. Ngắt ngoài trên STM32

1. Mô tả

- Hệ thống được thiết kế thực hiện việc lập trình điều kiển ngắt ngoài bật/tắt đèn Led trên STM32F401VE.Trên MCU này, nhà sản xuất đã kết nối sẵn cho chúng ta chân PA0 với User button nên ta sử dụng chân này để kết nối với nút bấm. Khi ta nhấn một nút nhấn thì phát sinh sự kiện ngắt ngoài gửi vào vi điều khiển, khi đó vi điều khiện triệu gọi một chương trình con phục vụ ngắt để thực hiện bật/tắt led ở PA12.

2. Linh kiện

* 1 đèn Led
* 1 Nút bấm
* 1 STM32F401VE

3. Sơ đồ mạch

4 . Mã lệnh

|  |
| --- |
| //Câu lệnh if (GPIO\_Pin == GPIO\_PIN\_0) sẽ giúp kiểm tra, phân luồng, phát hiện ngắt có //đúng đang sinh ra có phải ở chân 0 hay không.  void HAL\_GPIO\_EXTI\_Callback(uint16\_t GPIO\_Pin)  {  if(GPIO\_Pin == GPIO\_PIN\_0)  { HAL\_GPIO\_TogglePin(GPIOD, GPIO\_PIN\_12); }  } |

Bài 20. Thay đổi màu led RGB bằng chiếc áp

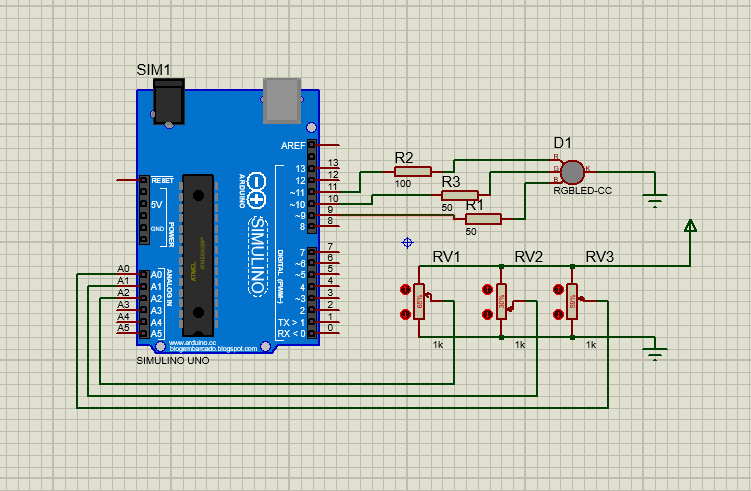
1. Mô tả

- Hệ thống được thiết kế điều khiển làm đèn RGB sáng ở các màu bất kỳ thông qua ba chiết áp được kết nối vào Arduino ở các cổng số A0, A1, A2.

2.Linh kiện

* 1 Đèn RBB
* 3 Điện trở
* 3 Chiết áp

3. Sơ đồ mạch



4.Mã lệnh

|  |
| --- |
| int led1 = 9; int led2 = 10; int led3 = 11; // Khai báo số chân Arduino  void setup() // nối với 3 đèn  {  pinMode(led1, OUTPUT);  pinMode(led2, OUTPUT);  pinMode(led3, OUTPUT);  }  void loop()  {  int val1 = map(analogRead(A0),0,1023,0,255); //Đọc giá trị chiết áp 1  int val2 = map(analogRead(A1),0,1023,0,255); //Đọc giá trị chiết áp 2  int val3 = map(analogRead(A2),0,1023,0,255); //Đọc giá trị chiết áp 3  analogWrite(led1,val1);  analogWrite(led2,val2);  analogWrite(led3,val3);  } |

Bài 21. Điều khiển động cơ DC với IC L293D và Drduino

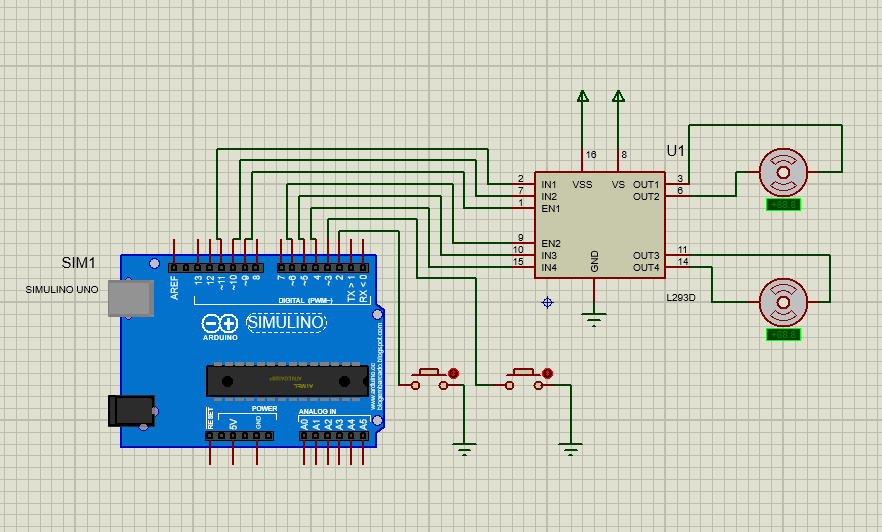
1**. Mô tả**

- Hệ thống được thiết kế thực hiện việc lập trình điều kiển động cơ DC thông qua IC điều khiển động cơ L293D

2. Linh kiện

* DC
* 2 NÚT BẤM
* Để có thể điều khiển hoàn toàn động cơ DC tức là chúng ta phải điều khiển tốc độ và chiều quay của nó. Để thực hiện diều này ta có thể kết hợp hai kỹ thuật sau. **Chế độ rộng xung (PWM)** để kiểm soát tốc độ và **Mạnh cầu H (H-Bridge)** để điều khiển hướng quay.
* PWM - Để kiểm soát tốc độ

3. Sơ đồ mạch



4.Mã lệnh

|  |
| --- |
| // Động cơ 1  int tocdo = 9; int inp1 = 11; int inp2 = 10;  // Động cơ 2  int tocdo2 = 6; int inp3 = 5; int inp4 = 4;  void setup()  {  pinMode(tocdo, OUTPUT); pinMode(inp1, OUTPUT);  pinMode(inp2, OUTPUT); pinMode(tocdo2, OUTPUT);  pinMode(inp3, OUTPUT); pinMode(inp4, OUTPUT);  digitalWrite(2, HIGH);  attachInterrupt(0, tatdongco, CHANGE);  digitalWrite(3, HIGH);  attachInterrupt(1,dieukientocdogiam, CHANGE);  //Tắt động cơ  digitalWrite(inp1,LOW); digitalWrite(inp2,LOW);  digitalWrite(inp3,LOW); digitalWrite(inp4,LOW);  }  void tatdongco()  {  digitalWrite(inp1,LOW); digitalWrite(inp2,LOW);  digitalWrite(inp3,LOW); digitalWrite(inp4,LOW);  }  void directionControl()  {  // Cho động cơ quay với tốc độ tối đa  // Giá trị của PWM thay đổi từ 0 đến 255  analogWrite(tocdo, 255); analogWrite(tocdo2, 255);  // Tắt động cơ A và B  digitalWrite(inp1, HIGH); digitalWrite(inp2, LOW);  digitalWrite(inp3, HIGH); digitalWrite(inp4, LOW);  delay(2000);  // Thay đổi chiều quay của động cơ  digitalWrite(inp1, LOW); digitalWrite(inp2, HIGH);  digitalWrite(inp3, LOW); digitalWrite(inp4, HIGH);  delay(2000);    // Tắt tất cả các động cơ  digitalWrite(inp1, LOW); digitalWrite(inp2, LOW);  digitalWrite(inp3, LOW); digitalWrite(inp4, LOW);  }  void speedControl()  {  digitalWrite(inp1, LOW); digitalWrite(inp2, HIGH);  digitalWrite(inp3, LOW); digitalWrite(inp4, HIGH);    for(int i=0; i<256 ;i++){  analogWrite(tocdo,i);  analogWrite(tocdo2,i);  delay(10);  }    for(int i=255;i >= 0; --i){  analogWrite(tocdo,i);  analogWrite(tocdo2,i);  delay(10);  }  //tat dong co  digitalWrite(inp1,LOW); digitalWrite(inp2,LOW);  digitalWrite(inp3,LOW); digitalWrite(inp4,LOW);  }  void dieukientocdotang()  {  //bat cac dong co  digitalWrite(inp1,LOW); digitalWrite(inp2,HIGH);  digitalWrite(inp3,LOW); digitalWrite(inp4,HIGH);    //tang toc do den max  for(int i=0; i<256 ;i++){  analogWrite(tocdo,i);  analogWrite(tocdo2,i);  delay(10);}  }  void dieukientocdogiam()  { //bat cac dong co  digitalWrite(inp1,LOW); digitalWrite(inp2,HIGH);  digitalWrite(inp3,LOW); digitalWrite(inp4,HIGH);    //giam toc do ve 0  for(int i=255;i >= 0; --i){  analogWrite(tocdo,i);  analogWrite(tocdo2,i);  delay(10);  }  }  void loop()  {  directionControl();  delay(1000);  speedControl();  delay(1000);  } |

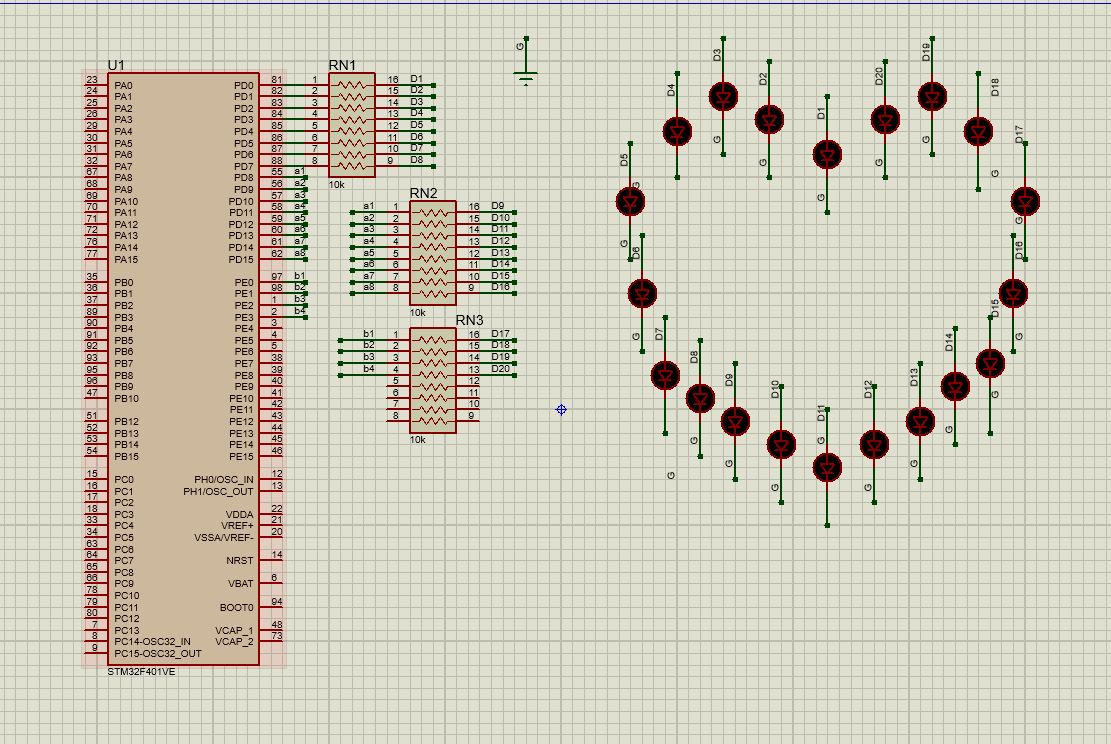
Bài 21. Led trái tim bằng STM32

1. Mô tả

- Hệ thống được thiết kế thực hiện việc lập trình điều kiển bật/tắt 20 đèn Led được sắp xếp thành hình trái tim, các đèn Led được kết nối vào STM32 thông qua 3 qua điện trở 16 cổng.

2. Linh kiện

* 20 đèn Led
* 3 bộ điện trở 16 cổng
* 1 STM32F401VE

3. Sơ đồ mạch

4.Mã lệnh

|  |
| --- |
| //Xây dựng các hàm tắt đèn, sáng đèn, sáng lần lượt  void SangHetDen(); //Bật hết tất cả các đèn  void TatHetDen(); //Tắt hết tất cả các đèn  void SangLanLuot(); //Bật sáng từng đèn, giữ nguyên không tắt  void SangLanLuot2(); //Nháy từng đèn  int main(void)  {  HAL\_Init();  SystemClock\_Config();  MX\_GPIO\_Init();  while (1)  { //Chương tình chính  SangHetDen();  HAL\_Delay(1000);  TatHetDen();  HAL\_Delay(1000);  SangLanLuot();  HAL\_Delay(1000);  SangLanLuot2();  }  }  void SangHetDen(){  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);  //Sáng đèn ở các cổng PE  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);  }  void TatHetDen(){  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);  // Tắt đèn ở các cổng PE  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);  }  void SangLanLuot(){  TatHetDen();  HAL\_Delay(100);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_0, GPIO\_PIN\_SET);  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\_SET);  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\_SET);  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\_SET);  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\_SET);  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\_SET);  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\_SET);  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\_SET);  HAL\_Delay(100);  HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_3, GPIO\_PIN\_SET);  HAL\_Delay(100);  }  void SangLanLuot2(){  TatHetDen();  HAL\_Delay(100);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_0, GPIO\_PIN\_SET);  HAL\_Delay(100);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_0, GPIO\_PIN\_RESET);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_1, GPIO\_PIN\_SET);  HAL\_Delay(100);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_1, GPIO\_PIN\_RESET);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_2, GPIO\_PIN\_SET);  HAL\_Delay(100);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_2, GPIO\_PIN\_RESET);    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\_SET);  HAL\_Delay(100);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_5, GPIO\_PIN\_RESET);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_6, GPIO\_PIN\_SET);  HAL\_Delay(100);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_6, GPIO\_PIN\_RESET);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_7, GPIO\_PIN\_SET);  HAL\_Delay(100);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_7, GPIO\_PIN\_RESET);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_8, GPIO\_PIN\_SET);  HAL\_Delay(100);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_8, GPIO\_PIN\_RESET);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_9, GPIO\_PIN\_SET);  HAL\_Delay(100);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_9, GPIO\_PIN\_RESET);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_10, GPIO\_PIN\_SET);  HAL\_Delay(100);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_10, GPIO\_PIN\_RESET);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_11, GPIO\_PIN\_SET);  HAL\_Delay(100);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_11, GPIO\_PIN\_RESET);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_12, GPIO\_PIN\_SET);  HAL\_Delay(100);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_12, GPIO\_PIN\_RESET);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_13, GPIO\_PIN\_SET);  HAL\_Delay(100);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_13, GPIO\_PIN\_RESET);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_14, GPIO\_PIN\_SET);  HAL\_Delay(100);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_14, GPIO\_PIN\_RESET);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_15, GPIO\_PIN\_SET);  HAL\_Delay(100);  HAL\_GPIO\_WritePin (GPIOD, GPIO\_PIN\_15, GPIO\_PIN\_RESET);  HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_0, GPIO\_PIN\_SET);  HAL\_Delay(100);  HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_0, GPIO\_PIN\_RESET);  HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_1, GPIO\_PIN\_SET);  HAL\_Delay(100);  HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_1, GPIO\_PIN\_RESET);  HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_2, GPIO\_PIN\_SET);  HAL\_Delay(100);  HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_2, GPIO\_PIN\_RESET);  HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_3, GPIO\_PIN\_SET);  HAL\_Delay(100);  HAL\_GPIO\_WritePin (GPIOE, GPIO\_PIN\_3, GPIO\_PIN\_RESET);  } |

Giới thiệu Appinventor.

1.Giới thiệu

App Inventor là một ứng dụng web mã nguồn mở được cung cấp bởi Google từ tháng 7 năm 2010. Sau này, App Inventor được quản lý bởi Viện Công nghệ Massachusetts hay còn gọi là MIT. Đó cũng là lý do tại sao nó hay được gọi là MIT App Inventor.

Về cơ bản, App Inventor sẽ hoạt động dựa trên nền tảng di động Android. Tức là các thành phẩm được tạo ra từ App Inventor sẽ chỉ hoạt động được trên Android. Giao diện của App Inventor bao gồm các khối hộp, bên trong là các đoạn mã. Khi sử dụng, người dùng sẽ kéo thả các khối này vào bảng mã để tiến hành lắp ghép thành một ứng dụng hoàn chỉnh. Nhìn chung, cách sử dụng App Inventor rất đơn giản, tất cả chỉ xoay quanh thao tác kéo và thả thôi.

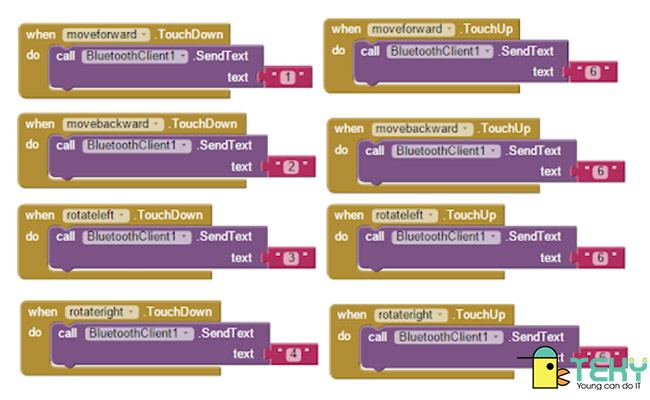
2.Hướng dẫn sử dụng cơ bản

**Bước 1 – App Inventor là gì**

Đầu tiên, ta sẽ truy cập vào MIT, đăng nhập tài khoản Google và khởi tạo một project mới.

Lúc này, trên màn hình sẽ có giao diện đơn giản gồm 3 phần chính. Bên trái là tập hợp các control bao gồm: User Interface, Media, Sensor, Social,… Để sử dụng những nút điều khiển này,  bạn chỉ cần click chuột và kéo thả vào Screen. Ở giữa là Screen mô phỏng màn hình ứng dụng của chúng ta, cũng sẽ nơi hiển thị project khi hoàn thiện. Còn bên phải là cửa sổ quản lý các Component, Media, và Property cho từng Control.

Ta sẽ dùng các khối block này để thiết kế nên phần điều khiển bên ngoài của ứng dụng.

**Bước 2**Khi ứng dụng đã có giao diện trực quan, ta bước tiếp sang phần code ứng dụng. Góc trên bên phải màn hình có nút Block, bạn cần click vào đó để đổi sang màn hình code.

Khi code, ta sử dụng các mệnh đề như if then, when do để tiến hành tạo nên các câu lệnh. Bên trái màn hình sẽ có các câu lệnh đã được soạn sẵn. Việc của bạn là ghép lệnh vào mệnh đề hoàn chỉnh.

**Bước 3**

Khi đã hoàn thiện cả phần code và phần giao diện, hãy thử chạy bản demo của ứng dụng bạn. Nếu có sai sót, bạn cần phải rà soát lại toàn bộ và chỉnh sửa lỗi. Khi ứng dụng đã chạy mượt, bạn có thể xuất ra file apk để sử dụng và lưu trữ lại nhé.

Lưu ý: Chọn Build để hiện lên màn hình lưu file. Sau đó hãy chọn dòng App nếu bạn muốn xuất phần mềm dạng QR để quét lên điện thoại. Nếu bạn muốn lưu trữ trong máy tính, hãy chọn dòng còn lại.

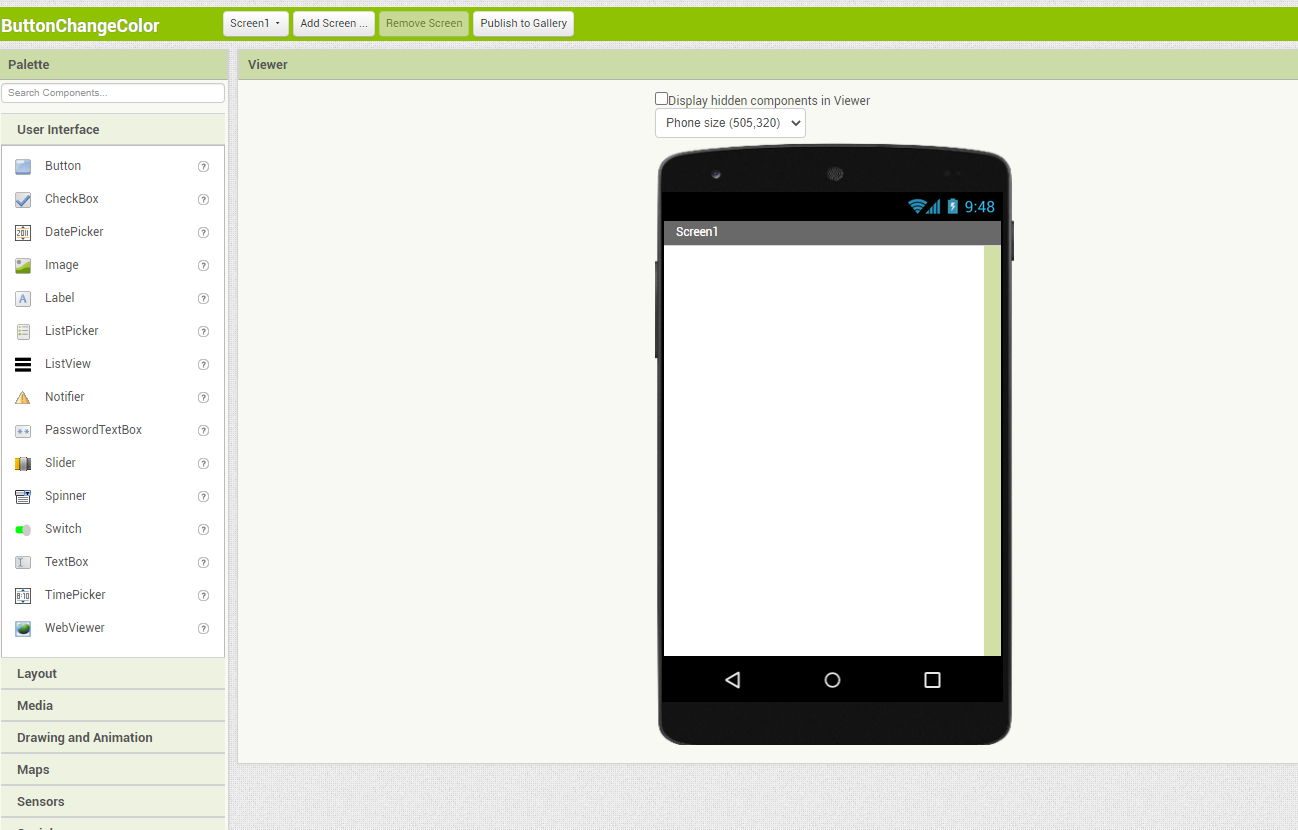
Nguồn: https://teky.edu.vn/

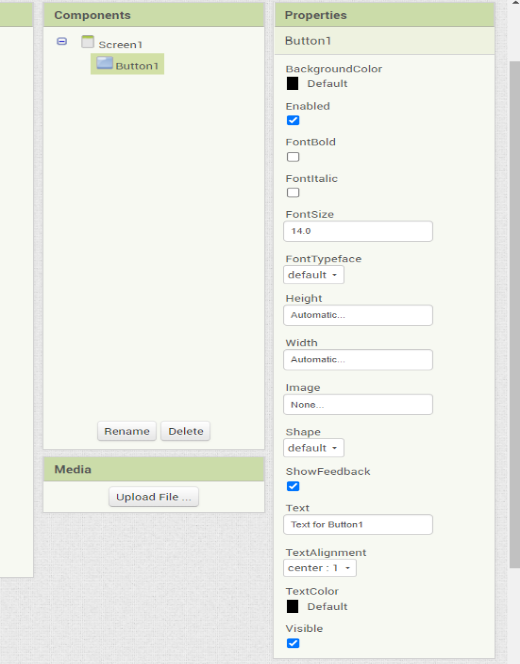
Một số project Appinventor

Nút bấm đổi màu

Ở ứng dụng này, khi ta nhấn nút 1 lần thì sẽ hiển thị 1 màu, giữ thì sẽ hiển thị 1 màu khác

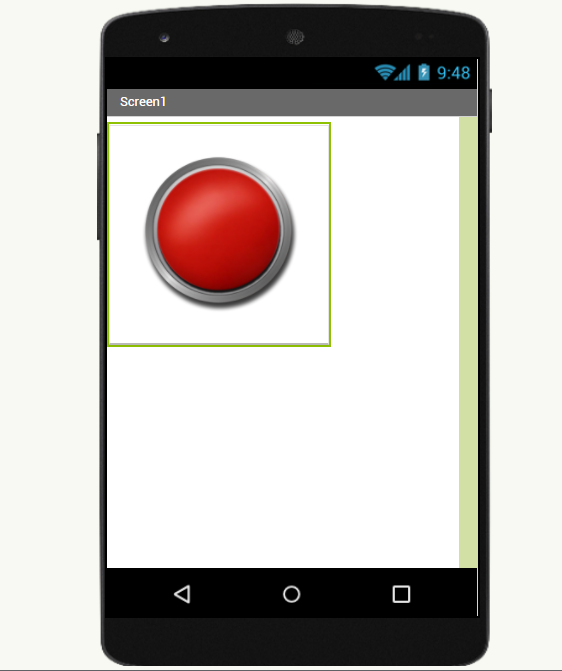
Đầu tiền bắt đầu: Ta tạo 1 project mới Project -> Start new project -> đặt tên cho dự án và Nhấn OK

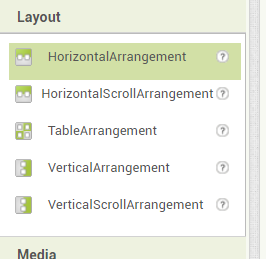


Đây là giao diện của Appinventor

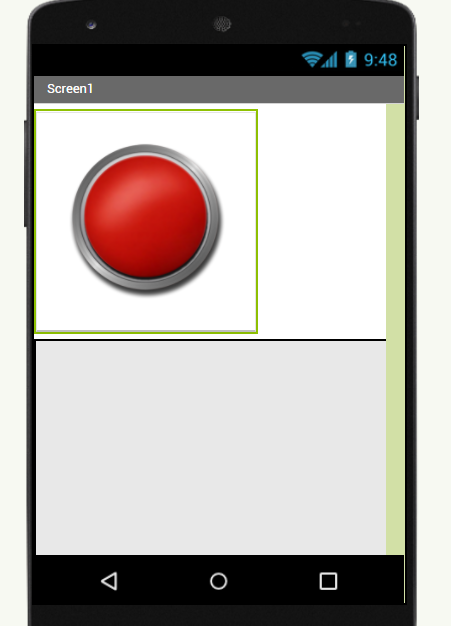
Ở thanh công cụ Use Interface ta chọn Button và kéo thả vào màn hình

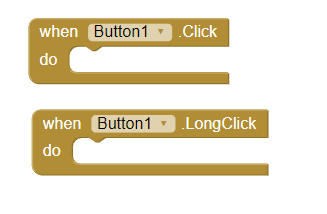
Ta có thể custom lại button bằng thanh công cụ phía bên trái bằng thay đổi các thuộc tính hoặc upload các biểu tượng hình ảnh chẳng hạn



Tiếp theo thêm 1 vùng để thực hiện thao tác đổi màu bằng thanh công cụ Layout và kéo thả vào màn hình và ta có thể điều chỉnh kích thước vùng theo ý muốn

Ta được giao diện như hình dưới:



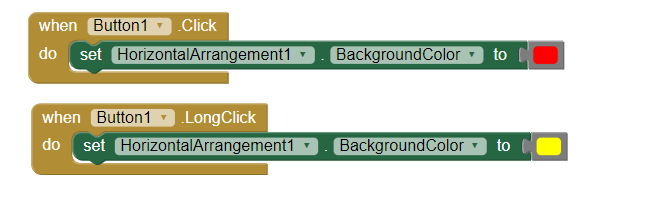


Tiếp theo ta chuyển chế độ sang block để bắt đầu cấu hình cho nút bấm

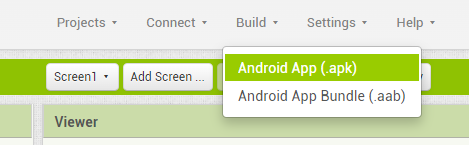
Ở Button ta kéo chọn lần lượt 2 chức năng Click và LongClick

Ở vùng ta kéo thả 2 thuộc tính Backgroud Color vào các event bấm mà bạn mong muốn và bạn có thể cấu hình màu

Ở đây khi button được nhấn hoặc giữ thì sẽ set background color cho vùng được tạo



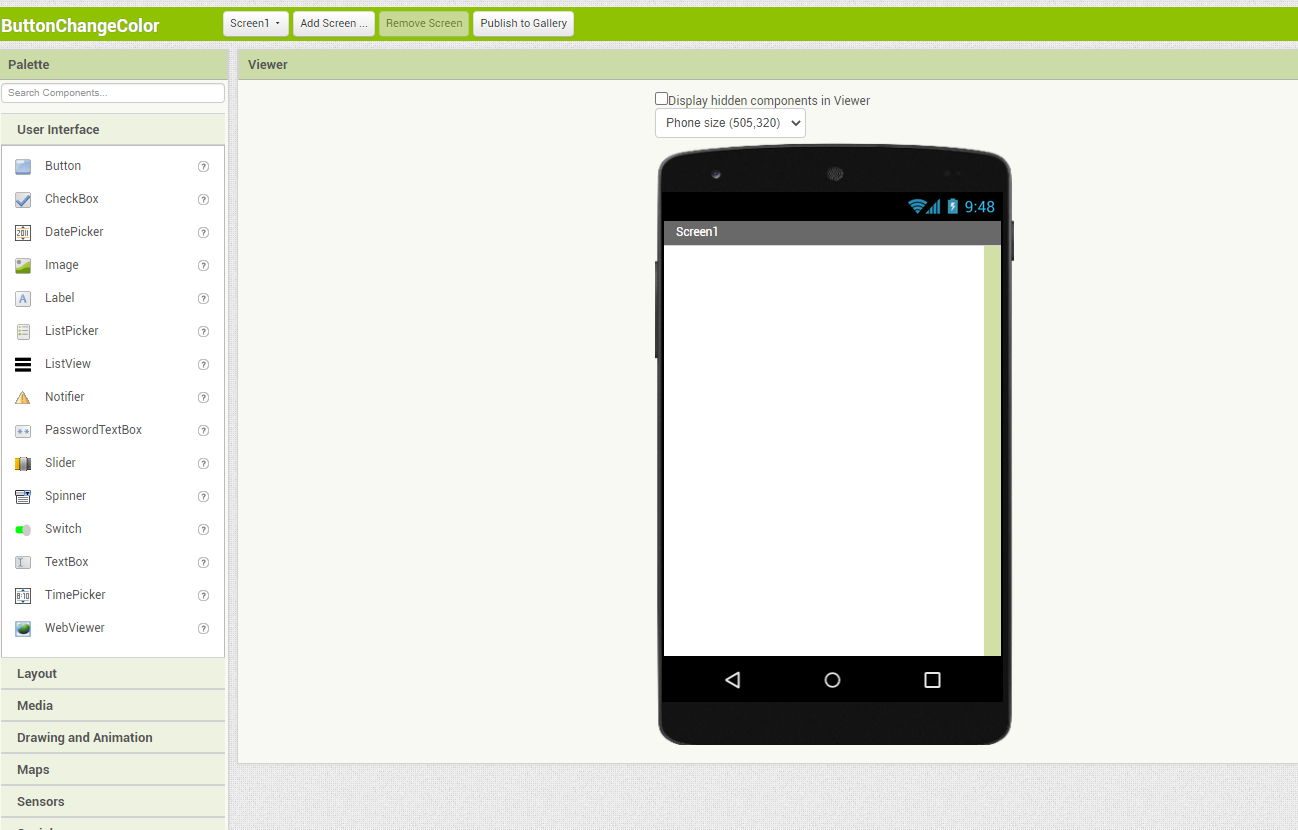
Cuối cùng bạn có thể build ra file apk và cài đặt chạy ứng dụng

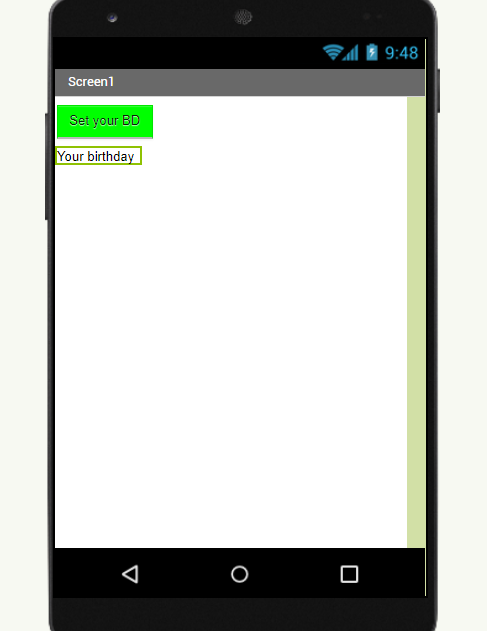


DatePicker

Ở bài tập này, ta sẽ thực hiện chức năng chọn ngày tháng và set ngày tháng như chức năng set ngày tháng trên hầu hết các thiết bị di động đều có. Ở đây ta chọn set ngày sinh nhật

Đầu tiền bắt đầu: Ta tạo 1 project mới Project -> Start new project -> đặt tên cho dự án và Nhấn OK

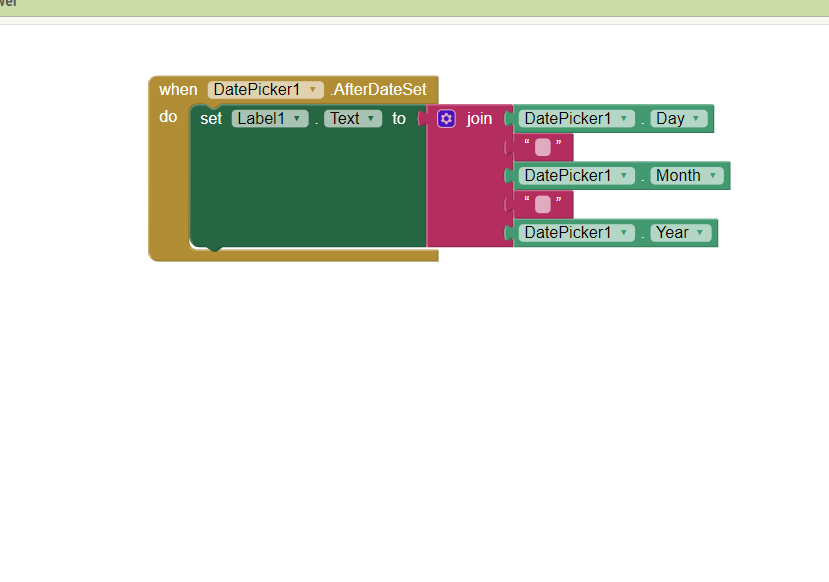


Tiếp theo ta kéo Thả Datepicker vào màn hình và style theo ý muốn

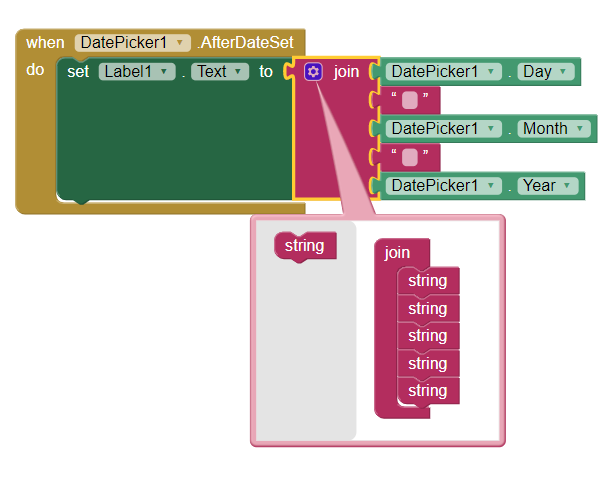
Kéo thả 1 label để hiển thị date mình đã set

Ta được kết quả như hình bên cạnh

Tiếp theo ta tiến hành cấu hình cho ứng dụng như hình dưới



Thẻ Join được lấy từ text và để thêm các chân nối ta chọn biểu tượng bánh răng và kéo thả thêm các chân nối cho thẻ



Ở đây khi sau khi ta pick các ngày tháng năm thì sẽ set ngày tháng năm vào label

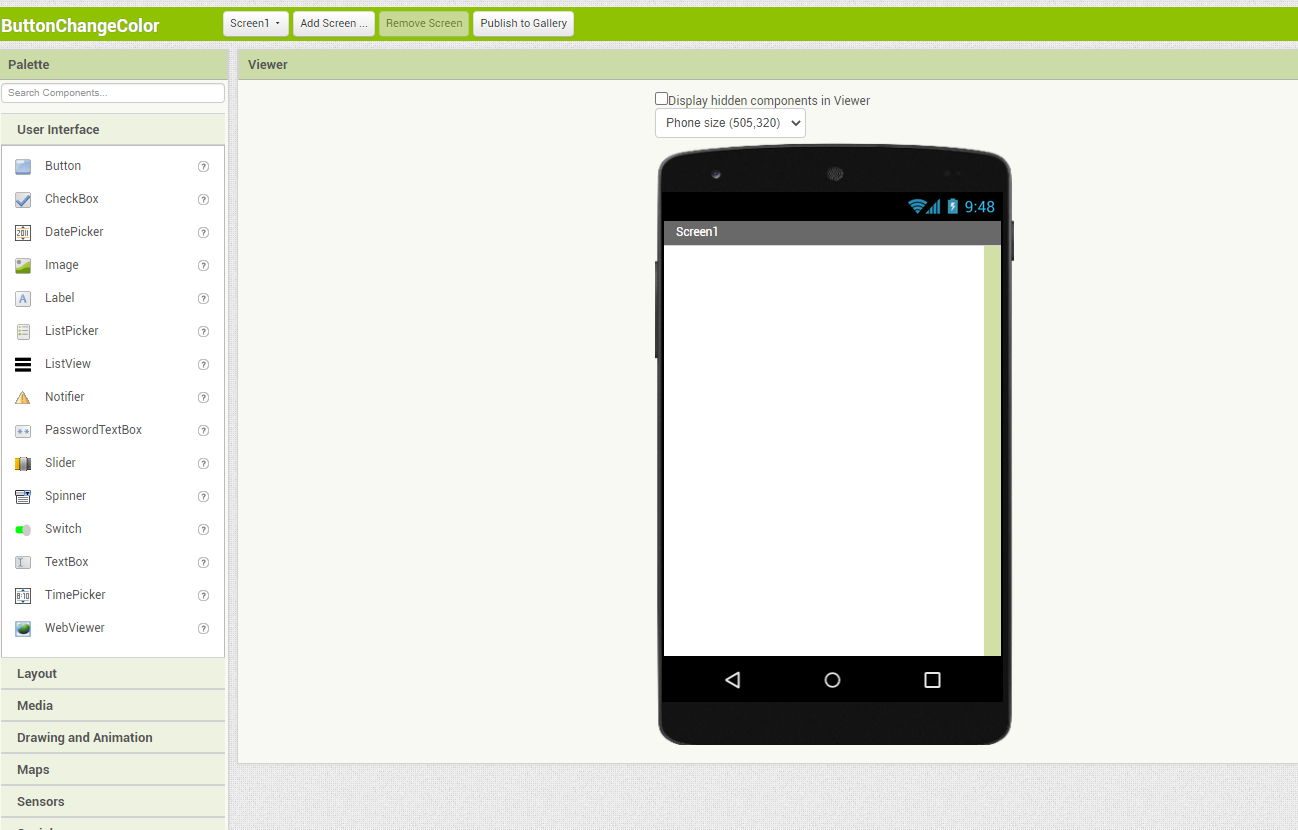
Hiển thị thông báo

Ở ứng dụng này ta sẽ xây dựng chức năng hiển thị chức năng và tùy chọn

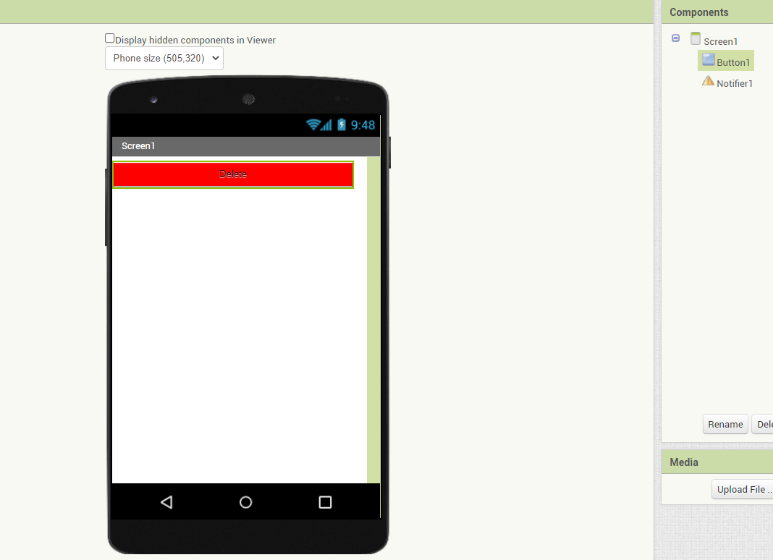
Chức năng này được ứng dụng hầu hết trên các thiết bị dùng để xác nhận 1 hành động hoặc thông báo

Sau đây ta sẽ demo chức năng xác nhận xóa 1 file

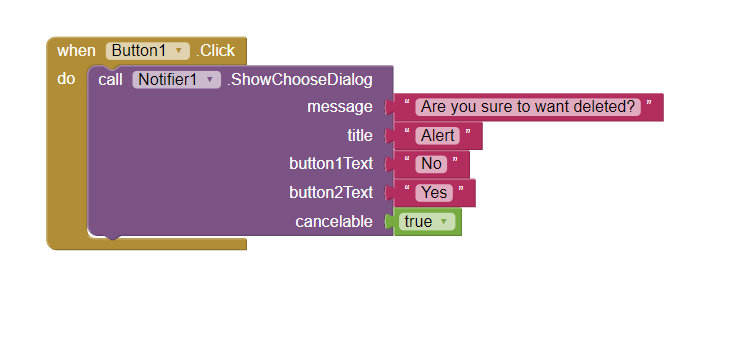
Đầu tiền bắt đầu: Ta tạo 1 project mới Project -> Start new project -> đặt tên cho dự án và Nhấn OK



Tiếp theo ta kéo thả button và notifer vào màn hình điện thoại



Tiến hành cấu hình ứng dụng như hình

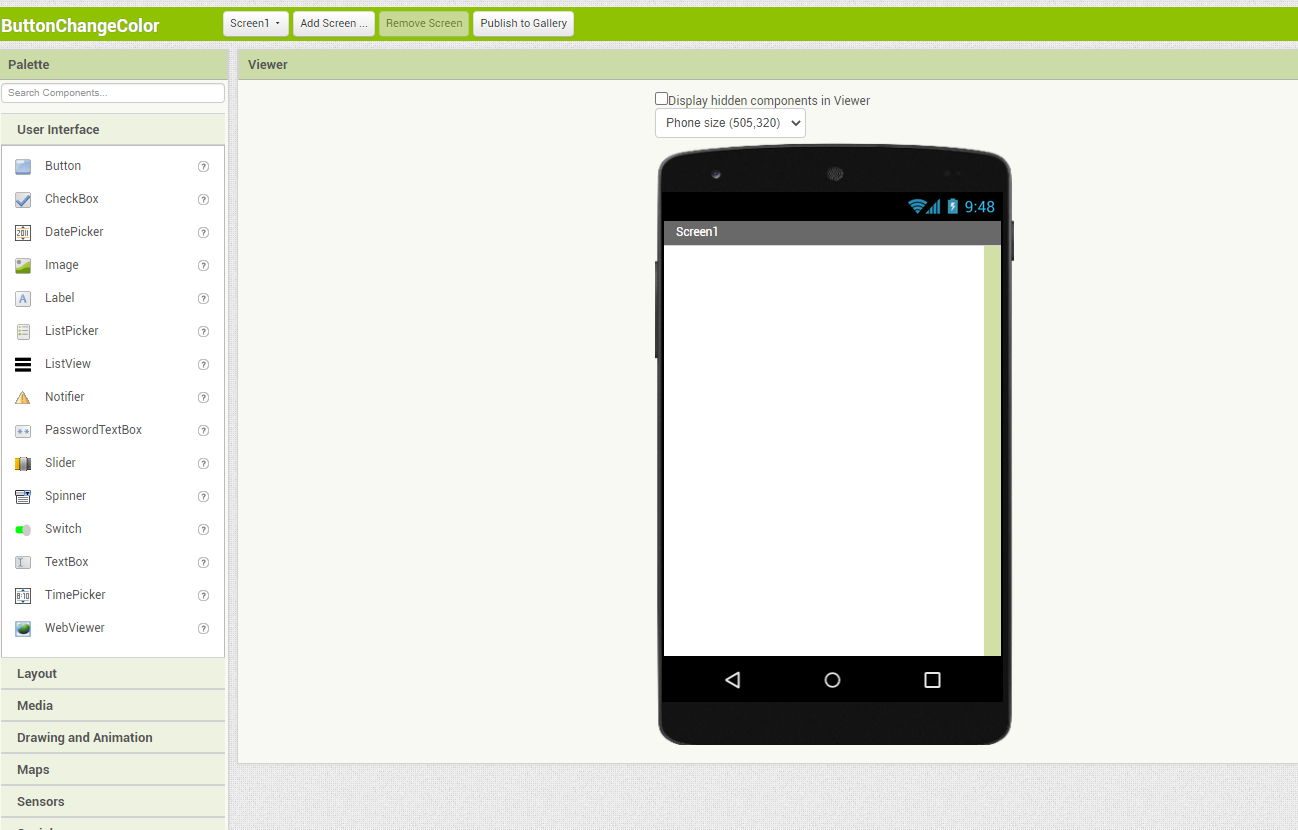


Ở đây, khi button được nhấn thì sẽ call thẻ Notifier

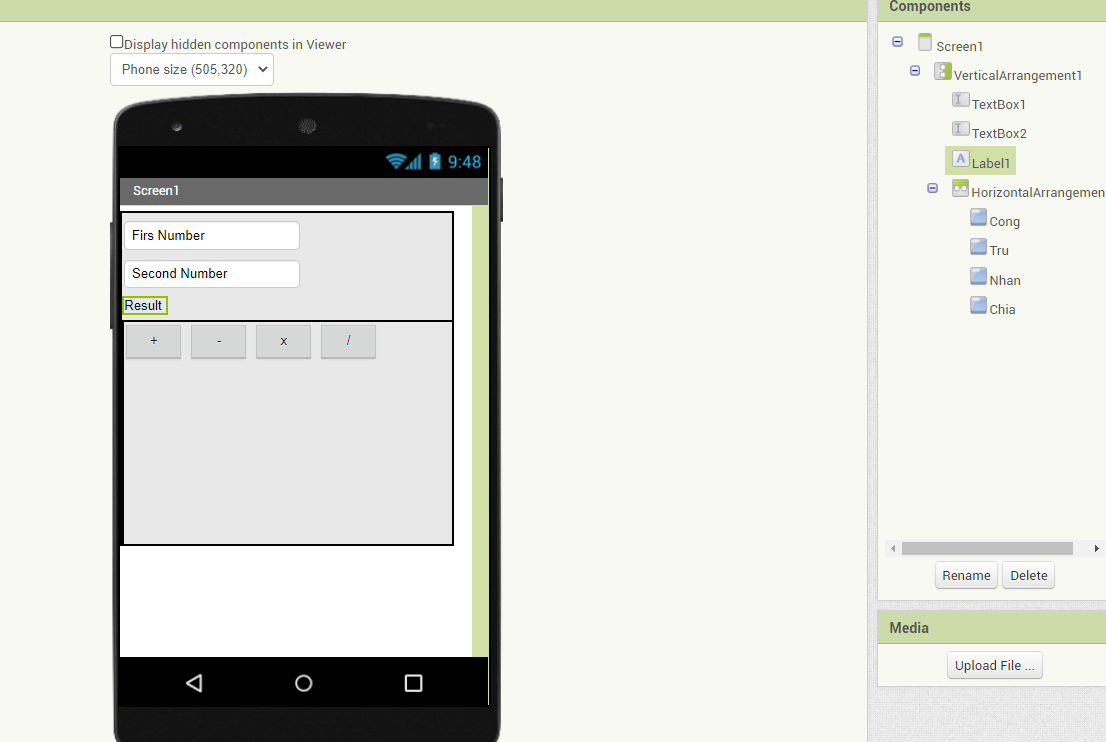
Thẻ Notifier gồm các thuộc tính Message, title, buttonText, cancelable, ta lần lược tùy chọn cho các thuộc tính này bằng cách theo thẻ text vào

Caculator

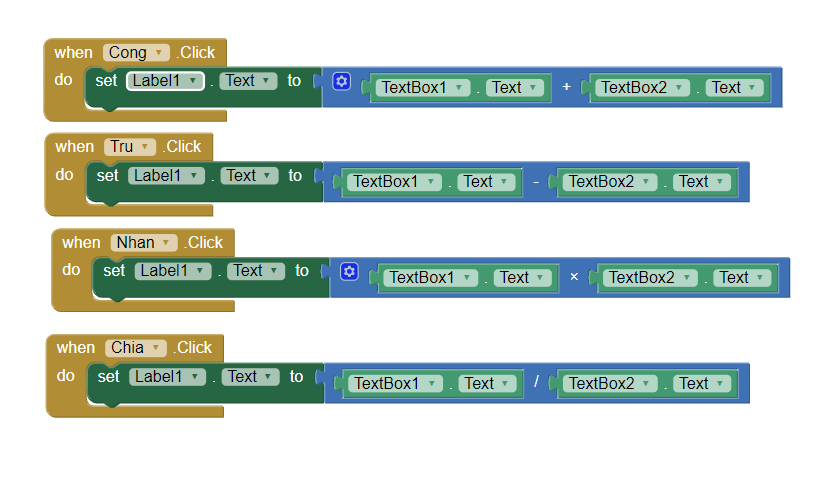
Ở ứng dụng này chúng ta sẽ xây dựng máy tính trên di động

Đầu tiền bắt đầu: Ta tạo 1 project mới Project -> Start new project -> đặt tên cho dự án và Nhấn OK

Tiến hành kéo thả theo giao diện như ảnh



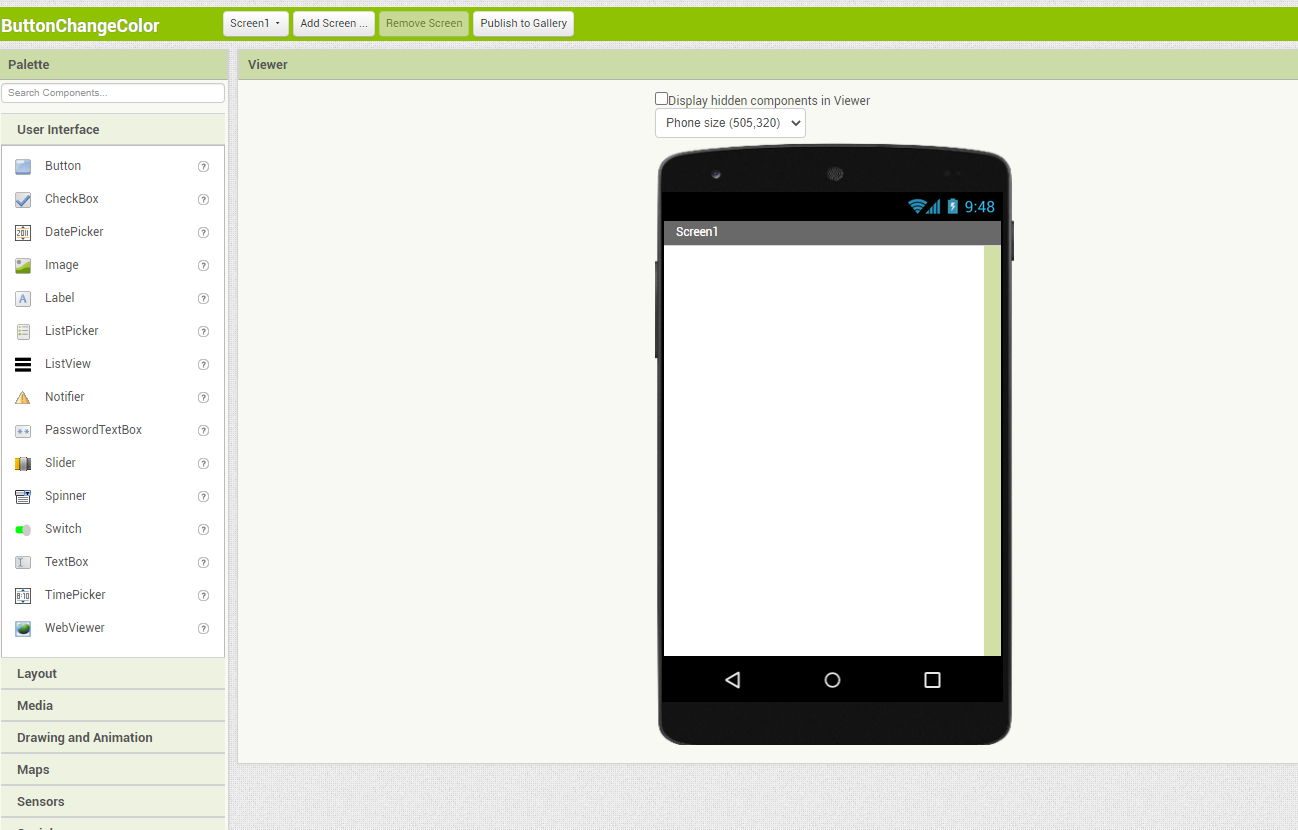
Cấu hình kéo thả các thẻ như ảnh

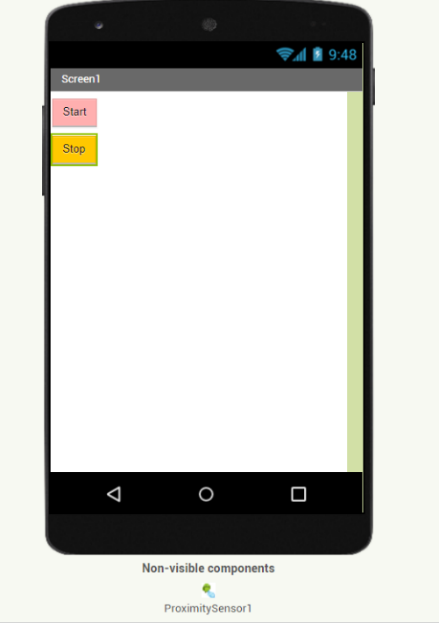


Ở đây khi ta nhấn vào button thì sẽ gọi set Label tức là sẽ nhận giá trị label từ thẻ Math giữa giá trị phép tính được định nghĩa của textBox1 và textBox2

Cảm biến vật cản

Ở ứng dụng này chúng ta sẽ xây dựng ứng dụng cảm biến vật cản trên di động. Khi phát hiện có vật cản trước mặt thì sẽ báo hiệu vật cản

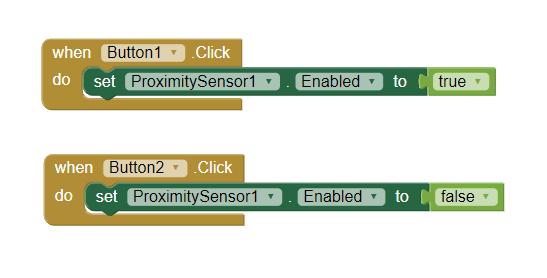
Đầu tiền bắt đầu: Ta tạo 1 project mới Project -> Start new project -> đặt tên cho dự án và Nhấn OK

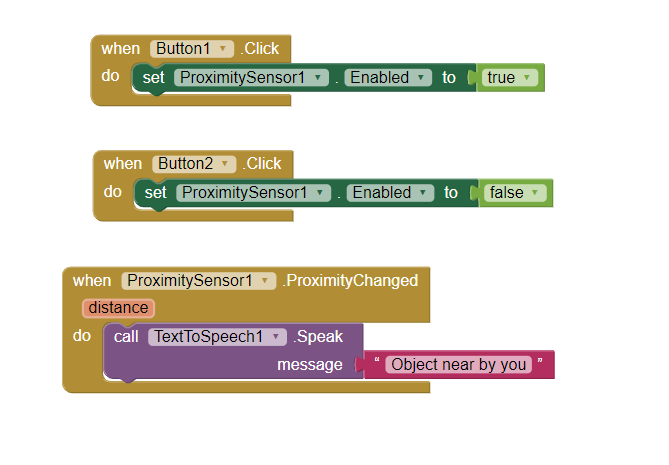


Tiếp theo ta thêm 2 button và cảm biến ProximitySensor vào màn hình điện thoại

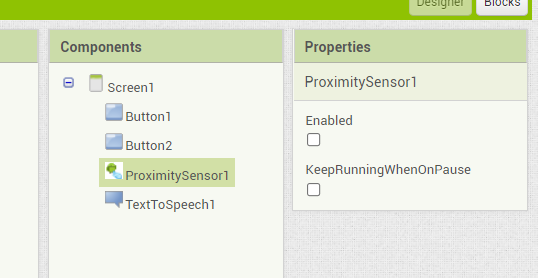
Cảm biến ProximitySensor dùng để phát hiện vật cản trước mặt

Ở đây ta có 1 nút thực hiện chức năng bắt đầu và

 1 nút dừng lại. Ta tiếp tục cấu hình bật tắt cảm biến như bên dưới

Để thực hiện chức năng cảnh báo có chướng ngại phía trước ta thêm TextToSpeech trong thẻ media vào màn hình và tiếp tục cấu hình.

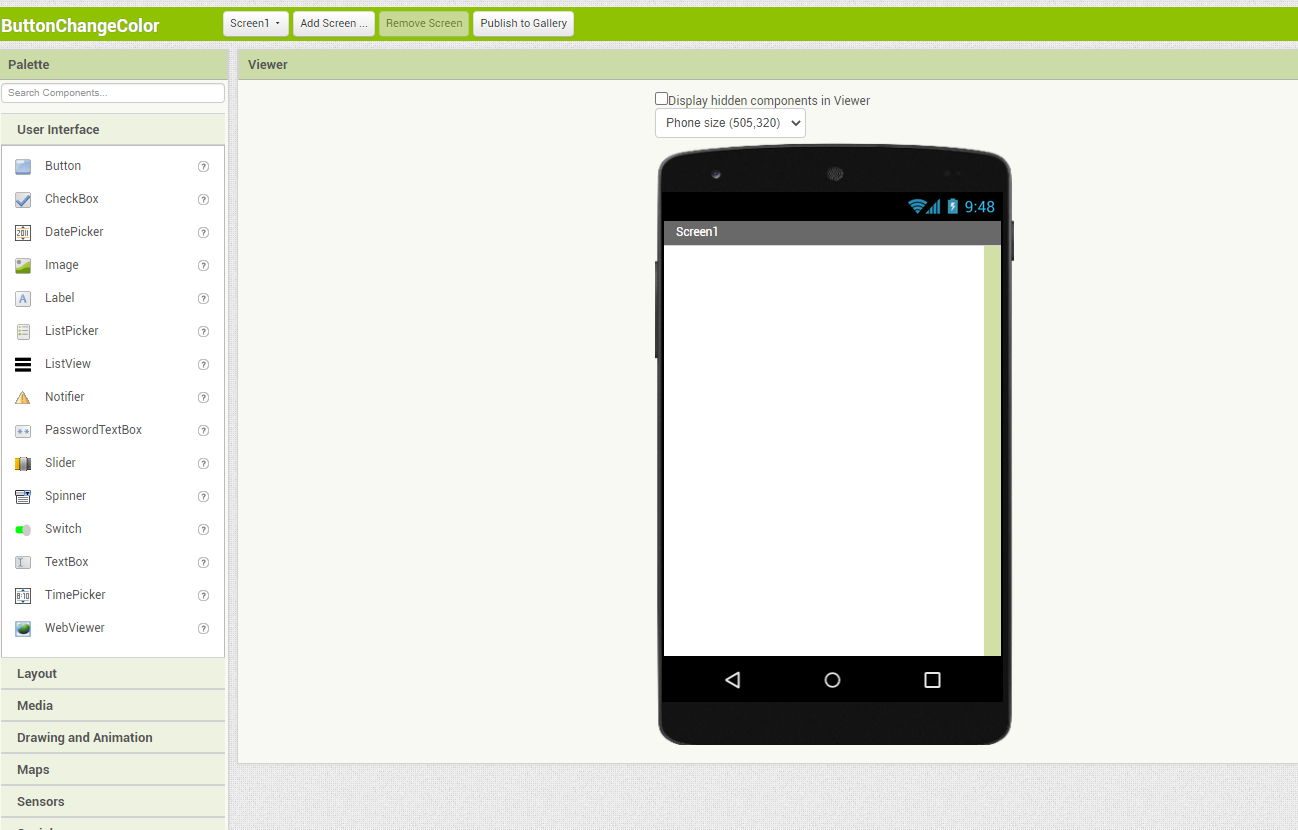
Ở đây khi ta cảm biến hoạt động sẽ gọi tên thẻ TextToSpeech và ở đây ta truyền thẻ Text vào phần message. Thẻ TextToSpeech sẽ chuyển từ text sang giọng nói thông báo vật cản ở phía trước

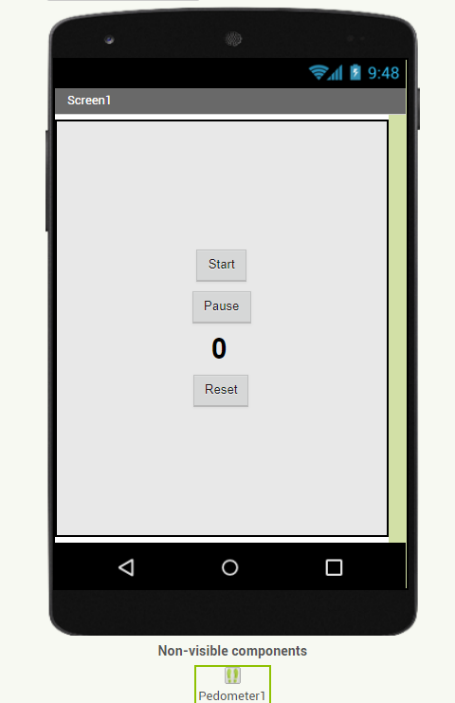


Lưu ý tắt trạng thái hoại động mặc định của cảm biến để tùy chỉnh được chức năng bật tắt của ứng dụng

COUTING APP

Ở ứng dụng này chúng ta sẽ xây dựng ứng dụng đếm khoảng cách đi bộ

Đầu tiền bắt đầu: Ta tạo 1 project mới Project -> Start new project -> đặt tên cho dự án và Nhấn OK

Tiếp theo ta thêm 3 button và 1 label để giá trị

mặc định là không

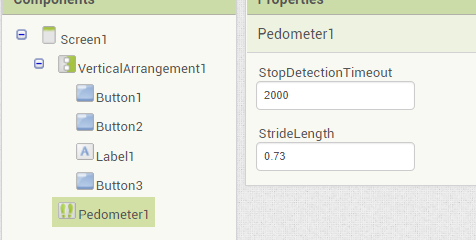
3 button còn lại thực hiện lần lược các chức

năng bắt đầu, dừng lại và reset

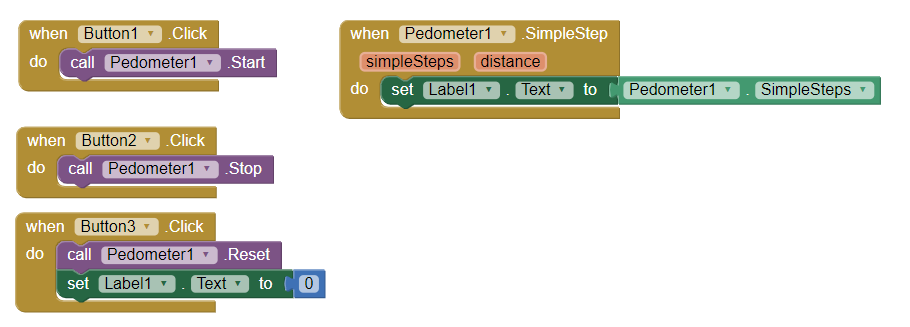
Ta thêm cảm biến Pedometer vào dùng để điểm

khoảng cách chạy và có thể cấu hình Timeout và

StrideLength



Ta cấu hình App như hình bên dưới



Khi ấn button1 tức là nút start sẽ gọi đén thẻ Pedometer bắt đầu

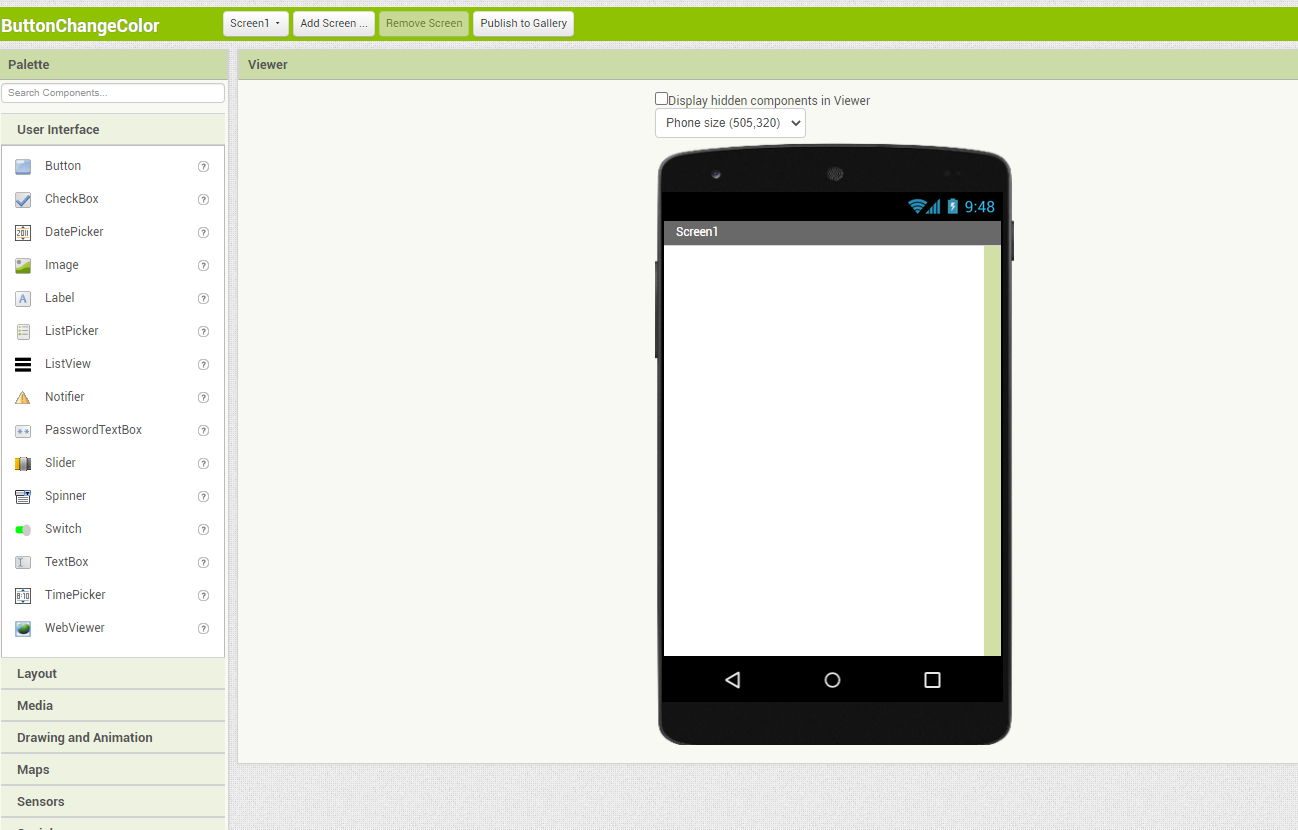
Khi ấn button2 tức là nút pause sẽ gọi đến thẻ Pedometer dừng

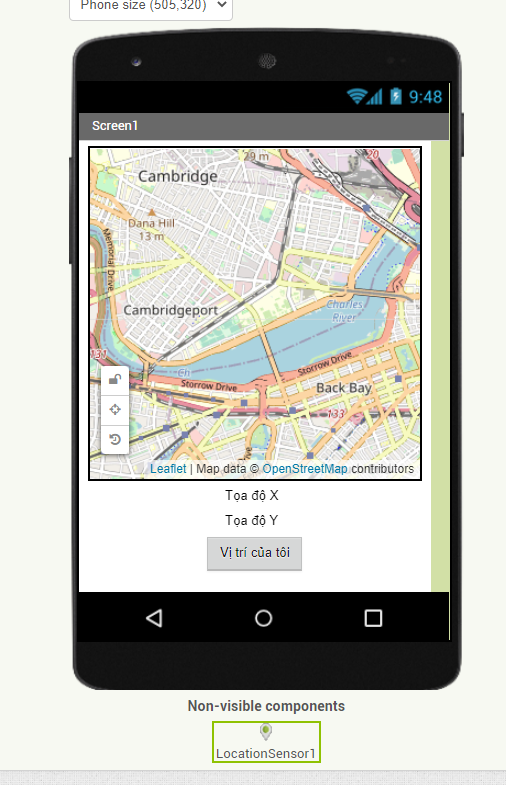
Khi ấn button3 tức là nút reset sẽ gọi đén thẻ Pedometer reset và set giá trị của label về 0

Khi cảm biến hoạt động thì gán giá trị vào thẻ label

YOUR LOCATION

Ở ứng dụng này chúng ta sẽ xây dựng ứng dụng xác định vị trí của bạn

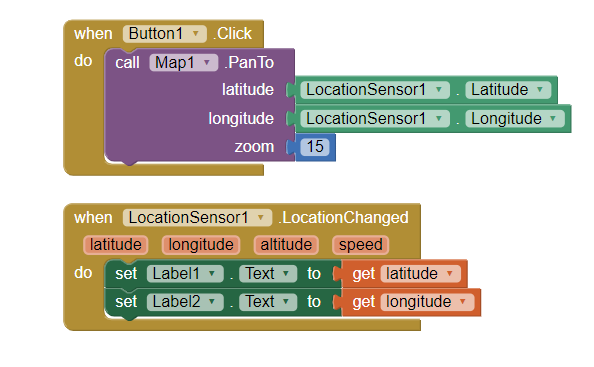
Đầu tiền bắt đầu: Ta tạo 1 project mới Project -> Start new project -> đặt tên cho dự án và Nhấn OK



Ta thiết lập giao diện như ảnh

* Bản đồ hiển thị
* Label hiển thị tọa độ X
* Label hiển thị tọa độ Y
* Button thực hiện bấm để khởi động cảm biến
* Cảm biến vị trí: Dùng để xác định vị trí của ta

Bằng giá trị tọa độ X, Y

Ta bắt đầu cấu hình ứng dụng như ảnh

* Khi ta bấm vào button thì cảm biến hoạt động và sẽ lấy giá trị tọa độ của bạn và map sẽ di chuyển đến vị trí của cảm biến và zoom 15% vị trí.
* Khi cảm biến thay đổi thì giá trị tọa độ X,Y sẽ gán vào 2 label1 và label2