# PROJECT IOT102

# Project: Radar phát hiện vật thể

|  |  |  |
| --- | --- | --- |
| Nhóm 1 | Nguyễn Trường Giang | HE163176 |
| Hà Ngọc Hiếu | HE163048 |

## I. Ý tưởng

Ý tưởng

* Các máy bay trên bầu trời hay một chiếc tàu thủy ở ngoài khơi làm thế nào để xác định đúng hướng đi của mình? Trong thực tế không phải lúc nào chúng ta cũng có thể dùng mắt thường hay camera để quan sát và đo đạc mọi thứ ở xa. Bên cạnh đó, cũng có rất nhiều môi trường nguy hiểm và đặc biệt mà con người cũng không thể tiếp cận như: dưới lòng đất, dưới đáy biển, nơi nhiễm phóng xạ,...
* Vì thế, cần có một thiết bị giúp chúng ta có thể “nhìn” trong những điều kiện hạn chế như vậy. Và radar chính là một công cụ như vậy.

Yêu cầu

* Radar tự động quay từ 0 đến 180 độ ở chế độ tự động và ngược lại hoặc tự điều chỉnh góc quay với chế độ thủ công thông qua triết áp.
* Phát hiện chính xác vật thể trong phạm vi 10-40cm với sai số nhỏ hơn 2cm.
* Thông báo bằng âm thanh qua còi khi phát hiện vật thể nằm trong khoảng nhỏ hơn 40cm.
* Hiển thị vị trí, khoảng cách vật thể quét được trên màn hình radar.

## II. Nội dung

### 1. Linh kiện (Hardware Required )

• Adruino Uno R3

• LED Display LCD 16 x 2

• Động cơ Servo (góc quay 180)

• Cảm biến siêu âm SRF05

• Còi chip 5V 9.5x12mm

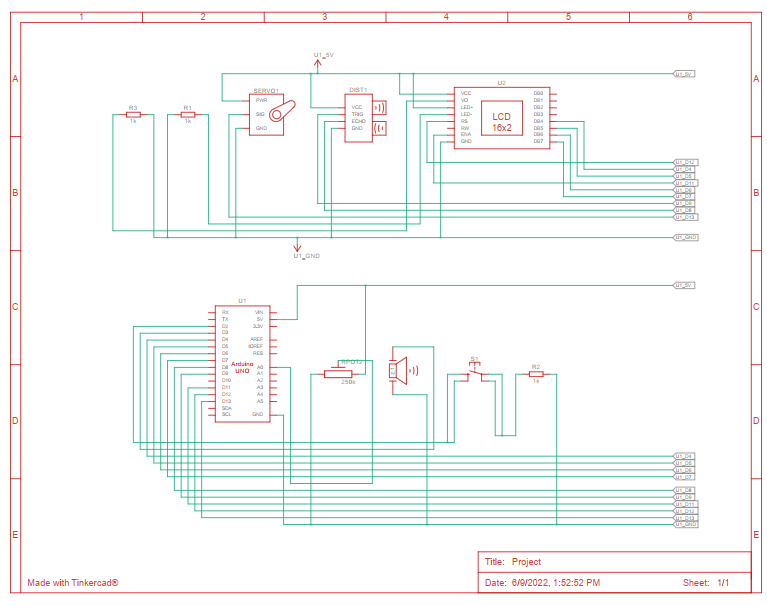
• Triết áp công tắc 10K

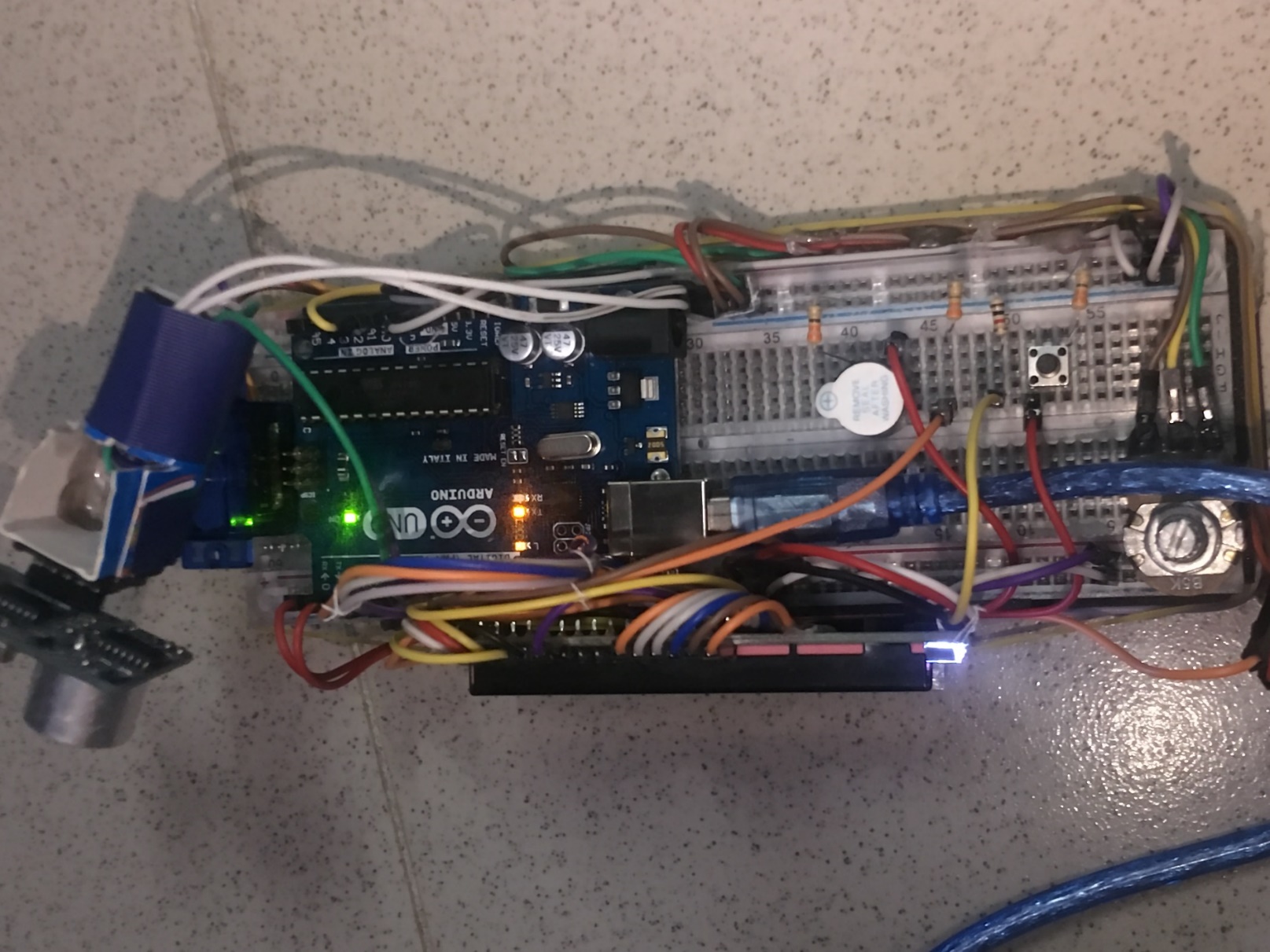
• Nút nhấn 2 chân

• Biến trở 1000 Ohm

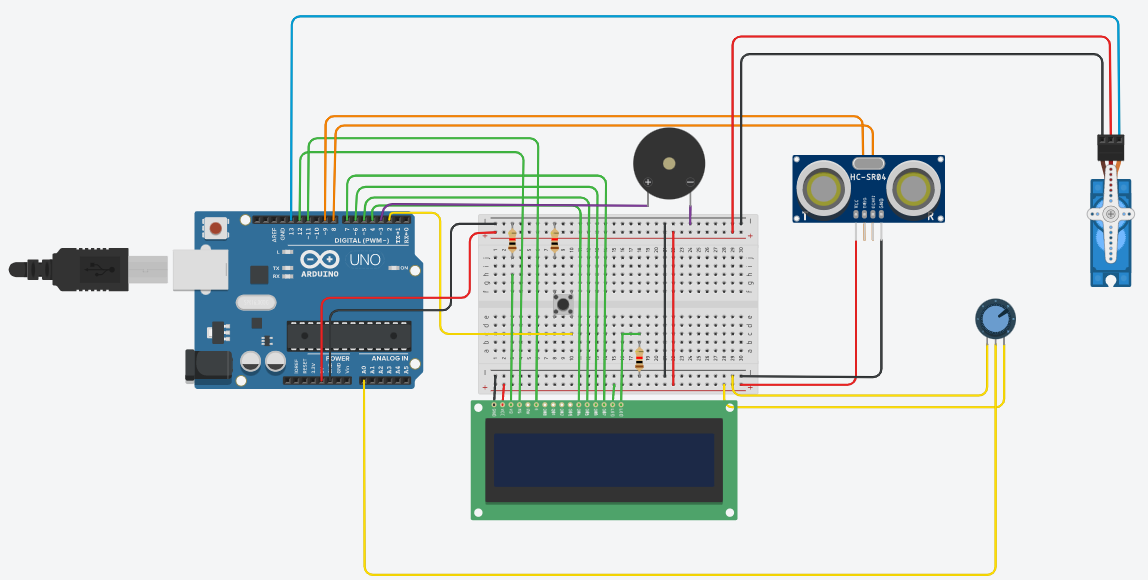
• Biến trở 330 Ohm

### 2. Mạch kết nối (Circuit )





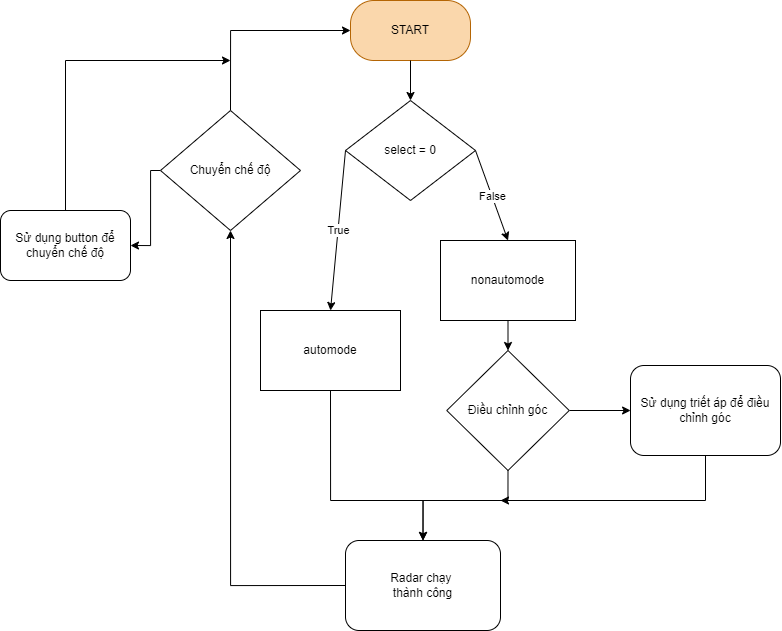
Ảnh thực tế



Ảnh tinkercad

### 3. Lưu đồ thuật toán (Flow chart)

Lưu đồ giải thuật của hàm **void loop**:



## III. Kết quả đạt được

* Sau khi tải code lên thông qua ứng dụng Ardunio, động cơ servo bắt đầu chạy từ 0 đến 180 độ và quay trở lại 0 độ. Một cảm biến siêu âm được gắn trên động cơ cũng quay cùng với servo.
* Một nút bấm được lắp trên bảng mạch cho phép chúng ta chuyển đổi chế độ quay của servo từ tự động sang quay bằng tay qua triết áp và ngược lại.
* Màn hình LCD sẽ hiển thị thông báo khi bắt đầu chạy và hiển thị chế độ đang được sử dụng.
* Mở và chạy code qua ứng dụng Processing. Một cửa sổ hiện lên, biểu diễn đồ họa của dữ liệu từ cảm biến siêu âm thu được.
* Nếu cảm biến siêu âm phát hiện bất kỳ đối tượng nào trong phạm vi của nó, chúng ta có thể thấy điều tương tự trên biểu đồ đồ họa đồng thời còi cảnh báo sẽ vang lên.

## IV. Phụ lục

### Link mô phỏng

* Link tinkercad

[Circuit design Project | Tinkercad](https://www.tinkercad.com/things/545ApiyqZnV-project/editel?sharecode=kNDb4KG2Fy1EGCKtorRqCeHxe0Jdzcbchsv8T4ao9Iw&fbclid=IwAR3RouW-nVM2npXwl-M9GkzlF60raGq1vpPzapEAeApDJWqmqM-SV7p4V9A)

* Link video

<https://drive.google.com/file/d/1K_2NyNnjgLvL0U0ywUDpFrhw_7Rtmk9n/view?usp=sharing>

<https://drive.google.com/file/d/1H1Gqp4cr1JmJ8A8u_SjRuRTdvumkZUI6/view?usp=sharing>

### 2. Code

- Phần code của dự án được chia thành 2 file chạy trên 2 ứng dụng là Ardunio và Processing (bên dưới)

### Code Ardunio

|  |
| --- |
| #include <LiquidCrystal.h>  #include <Servo.h>  const int trig = 9;  const int echo = 8;  const int rs = 12, en = 11, d4 = 4, d5 = 5, d6 = 6, d7 = 7;  int select = 0;  bool changeToManual = false;  bool flag = true;  int i = 0;  int value = 0;  int servoPos = 0;  LiquidCrystal lcd(rs, en, d4, d5, d6, d7);  Servo radar;  void setup() {  pinMode(3, OUTPUT);  pinMode(13, OUTPUT);  pinMode(2, INPUT\_PULLUP);  lcd.begin(16, 2);  radar.attach(13);  attachInterrupt(digitalPinToInterrupt(2), blink, RISING);  Serial.begin(9600);  pinMode(trig, OUTPUT);  pinMode(echo, INPUT);  }  void loop() {  if(flag){  display("WELCOME TO THE","BOAT CAPTAIN");  delay(500);  flag =false;  }  if (select == 0) {  automode();  }else {  nonautomode();  }  }  float ultrasonic() {  unsigned long duration;  float distance;  digitalWrite(trig, 0);  delayMicroseconds(2);  digitalWrite(trig, 1);  delayMicroseconds(10);  digitalWrite(trig, 0);  duration = pulseIn(echo, HIGH);  distance = duration / 2 \* 0.0344;  return distance;  }  void display(String x1, String x2) {  lcd.clear();  lcd.setCursor(0, 0);  lcd.print(x1);  lcd.setCursor(0, 1);  lcd.print(x2);  }  void blink() {  select = !select;  changeToManual = (changeToManual)?false:true;  if(!changeToManual){  i = servoPos;  }  }  void automode() {  display("Auto","");  int d;  for (i; i <= 180; i++) {  if(changeToManual){  break;  }  radar.write(i);  delay(30);  d = (int)ultrasonic();  tone(d);  Serial.print(i);  Serial.print(",");  Serial.print(d);  Serial.print(".");  }  for (i; i >= 0; i -= 1) {  if(changeToManual){  break;  }  radar.write(i);  delay(30);  d = (int)ultrasonic();  tone(d);  Serial.print(i);  Serial.print(",");  Serial.print(d);  Serial.print(".");  }  }  void nonautomode() {  value = analogRead(A0);  servoPos = map(value, 0, 1023, 0, 180);  radar.write(servoPos);  int d = (int)ultrasonic();  tone(d);  Serial.print(servoPos);  Serial.print(",");  Serial.print(d);  Serial.print(".");  display("Manual","");  }  void tone(int d) {  if (d < 40) {  tone(3, 1000);  delay(10);  noTone(3);  }  } |
|  |

### Code Processing

|  |
| --- |
| import processing.serial.\*;  import java.awt.event.KeyEvent;  import java.io.IOException;  Serial myPort;  String angle="";  String distance="";  String data="";  String noObject;  float pixsDistance;  int iAngle, iDistance;  int index1=0;  int index2=0;  void setup() {  size (1200, 700);  smooth();  myPort = new Serial(this,"COM5", 9600);  myPort.bufferUntil('.');  }  void draw() {  fill(98,245,31);  noStroke();  fill(0,4);  rect(0, 0, width, height-height\*0.065);  pushMatrix();  translate(width/2,height-50);  drawRadar();  drawLine();  drawObject();  popMatrix();  drawText();  }  void serialEvent (Serial myPort) {  data = myPort.readStringUntil('.');  data = data.substring(0,data.length()-1);    index1 = data.indexOf(",");  angle= data.substring(0, index1);  distance= data.substring(index1+1, data.length());  iAngle = int(angle);  iDistance = int(distance);  }  void drawRadar() {  pushMatrix();  //translate(width/2,height-50);  noFill();  strokeWeight(2);  stroke(98,245,31);  // ve hinh elip  for(int i = 75; i <= 1075; i+=250){  arc(0,0,(width-i),(width-i),PI,TWO\_PI);  }  // ve duong goc  line(-width,0,width,0);  for(int i = 15; i <= 165; i += 15){  line(0,0,(-width/2)\*cos(radians(i)),(-width/2)\*sin(radians(i)));  }  popMatrix();  }  void drawObject() {  pushMatrix();  strokeWeight(9);  stroke(255,10,10); // red color  pixsDistance = iDistance\*((height-height\*0.1666)\*0.025);  if(iDistance < 40){  line(pixsDistance\*cos(radians(iAngle)),-pixsDistance\*sin(radians(iAngle)),(width\*0.5)\*cos(radians(iAngle)),-(width\*0.5)\*sin(radians(iAngle)));  }  popMatrix();  }  void drawLine() {  pushMatrix();  //translate(width/2,height-50);  strokeWeight(9);  stroke(30,250,60);  line(0,0,(height\*0.85)\*cos(radians(iAngle)),-(height\*0.85)\*sin(radians(iAngle)));  popMatrix();  }  void drawText() {  pushMatrix();  if(iDistance>40) {  noObject = "Ngoài phạm vi";  }else {  noObject = "Trong phạm vi";  }    fill(0,0,0);  noStroke();  rect(0, height-height\*0.0648, width, height);  fill(98,245,31);  textSize(30);  for(int i = 1; i <= 4; i++){  text((i\*10)+"cm",width-width\*(4-i+1)/10,height-53);  }  text("Vật thể: " + noObject, width\*0.1, height-15);  text("Góc: " + iAngle +" °", width\*0.5, height-15);  text("Khoảng cách: ", width\*0.75, height-15);  if(iDistance<40) {  text("" + iDistance +" cm", width\*0.9, height-15);  }  for(int i = 30; i <= 150; i += 30){  text(i+"°",(width\*0.5)+width/2\*cos(radians(i)),(height\*0.9)-width/2\*sin(radians(i)));  }  popMatrix();  } |