Radar

Autor: Alexandru Cirlomaneanu

Grupa: 333CB

Introducere

Proiectul consta in implementarea unui sistem radar care prin rotatie va semnala existenta obstacolelor din apropiere, imitand astfel radarul pe care il poate avea un avion sau un submarin. Acesta va genera semnale acustice si luminoase in functie de distanta la care se afla un obiect de radar si va si afisa distanta la care se afla obstacolul.

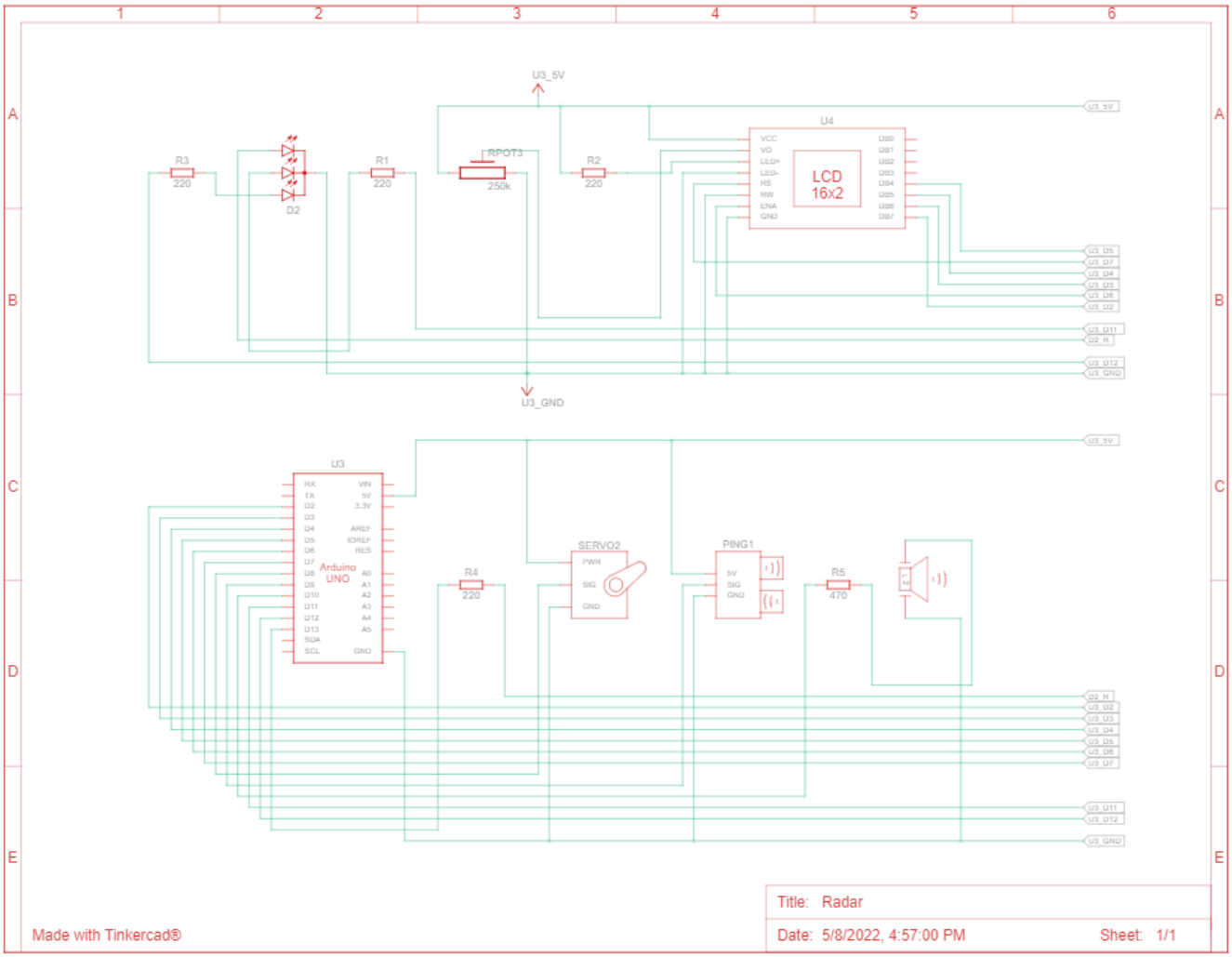
Descriere generală

Radar rotativ care afiseaza pe un ecran LCD distanta pana la un obstacol. Radarul va consta intr-un senzor de proximitate care se va roti cu ajutorul unui servomotor. De asemenea distanta va fi semnalata prin vibratia unui buzzer, iar un LED RGB isi va schimba culoarea verde-galben-rosu tot in functie de distanta.

Hardware Design

Componente necesare:

- Arduino Uno
- Senzor de Proximitate
- Servomotor
- Buzzer
- LED RGB
- Ecran LCD
- Rezistente
- Breadboard



Software Design

Pentru partea de afisare pe ecran LCD am folosit un ecran care avea deja integrat si I2C care necesita o biblioteca specifica: https://github.com/johnrickman/LiquidCrystal_I2C

In setup se pun pinii digitali pe modul de functionare, se ataseaza pinul la servomotor si se initializeaza lcd-ul.

In loop se cicleaza rotirea stanga-dreapta a servomotorului pe care sta asezat senzorul de distanta. Rotirea se face printr-o bucla repetitiva de rotire de la 0 la 180 de grade urmata de una care va roti in sens invers. De asemenea in cadrul buclei se verifica daca senzorul depisteaza vreun obstacol in apropiere.

Pentru a cauta un obstacol senzorul trimite un semnal si asteapta ecoul acestuia. Se calculeaza durata si se transforma in centimetri. Daca distanta este mai mica de 5 cm LED-ul RGB se aprinde rosu si buzzer-ul vibreaza. Daca distanta gasita este intre 5 si 20 de centimetri LED-ul se coloreaza in galben si buzzer-ul vibreaza mai putin intens. In ambele cazuri se afiseaza pe LCD distanta la care se afla obiectul si unghiul in grade hexazecimale la care era pozitionat servomotorul. In cazul in care distanta la care se afla obiectul este mai mare de 20 cm pe LCD se va scrie faptul ca suntem la o distanta sigura.

```
#include <LiquidCrystal_I2C.h>
#include <Servo.h>
#include <Wire.h>

  long duration, cm;

int red_light_pin = 13;
int blue_light_pin = 12;
int green_light_pin = 11;
int buzzer_pin = 10;
Servo servo_7;
int echo = 4;
int trigger = 5;

LiquidCrystal_I2C lcd(0x27, 16, 2);

int pos = 0;
int message = -1;
int obsFound = 0;

void setup() {
  // Initiem pinii pentru output
  servo_7.attach(7);
  pinMode(red_light_pin, OUTPUT);
  pinMode(blue_light_pin, OUTPUT);
  pinMode(green_light_pin, OUTPUT);
  pinMode(buzzer_pin, OUTPUT);
  pinMode(echo, INPUT);
  pinMode(trigger, OUTPUT);
  lcd.init();
  lcd.backlight();
  lcd.setCursor(0,0);
}

void loop() {
  for (pos = 0; pos <= 180; pos += 1) {
    // in steps of 1 degree
    servo_7.write(pos);
     getDistance();
  }
  for (pos = 180; pos >= 0; pos -= 1) {
    servo_7.write(pos);
     getDistance();
  }
}

void getDistance() {
  // Senzorul cu ultrasuente este declansat de un impuls HIGH
  // Dam un scurt impuls pe LOW pentru a ne asigura ca vom emite un semnal
  // HIGH neperturbat.
  digitalWrite(trigger, LOW);
```

```
    delayMicroseconds(2);
  digitalWrite(trigger, HIGH);
    delayMicroseconds(2);
  digitalWrite(trigger, LOW);
  // Se pune pinul pe input si se gaseste durata in care semnalul gaseste un
  // obstacol (durata impulsului care este HIGH de la emitere pana la
intoarcerea
  // ecoului)
  duration = pulseIn(echo, HIGH);

  // Obtinerea distantei
  cm = microsecondsToCentimeters(duration);

  // Culoarea rosie
  if(cm <= 5){
    digitalWrite(red_light_pin, HIGH);
    digitalWrite(green_light_pin, LOW);
    digitalWrite(blue_light_pin, LOW);
    //noTone(buzzer_pin);
    tone(buzzer_pin, 120, 200);
    message = 1;

  }
  // Culoarea galbena
  else if (cm > 5 && cm < 20){
    digitalWrite(red_light_pin, HIGH);
    digitalWrite(green_light_pin, HIGH);
    digitalWrite(blue_light_pin, LOW);
    //noTone(buzzer_pin);
    tone(buzzer_pin, 40, 50);
     lcd.clear();
    message = 1;

  }
  // Culoarea verde
  else if (cm > 20) {
    digitalWrite(red_light_pin, LOW);
    digitalWrite(green_light_pin, HIGH);
    digitalWrite(blue_light_pin, LOW);
    noTone(buzzer_pin);
  message = 0;
  }

  if (message == 1) {
    lcd.clear();
    lcd.print("Obstacle at ");
    lcd.print(cm);
    lcd.print("cm");
    lcd.setCursor(0, 1);
    lcd.print("Angle ");
    lcd.print(180 - pos);
```

```
    lcd.print(" degrees");
                delay(500);

  } else if (message == 0) {
    lcd.clear();
    lcd.print("Safe distance");
    delay(50);
  }

}

long microsecondsToCentimeters(long microseconds) {
  // Viteza sunetului in aer este 340 m/s sau 0.034 cm/microsecunda.
  // Folosim formula vitezei obtinem distanta parcursa de unda de la
  // emitere pana la intoarcere. Impartim distanta la 2 pentru a gasi
  // distanta la care se afla obstacolul.
  return microseconds * 0.034 / 2;
}
```

Rezultate Obţinute

Un scurt filmulet cu modul in care opereaza proiectul: https://youtu.be/_CwKtapNQDQ

Concluzii

In implementare am intampinat dificultati pe care nu le luasem in calcul in procesul de proiectare. In planificare initiala aveam un senzor de ultrasunete cu 3 pini (trigger si echo erau la comun), cand am construit proiectul am gasit un senzor cu 4 pini si m-am adaptat. Am gasit ca e mai eficient sa folosesc un LCD care are interfata I2C implementata salvand astfel niste pini pe placuta si niste cabluri incrucisate.

Proiectul a reprezentat o provocare intersanta mai ales in partea fizica in ceea ce priveste montarea si amplasarea componentelor. Cred ca in viitor as putea sa mai construiesc diferite aplicatii care s-ar putea dovedi utile in viata de zi cu zi.

Download

radar.zip - Arhiva cu codul Arduino + biblioteca LiquidCrystalI2C necesara ecranului LCD cu modul I2C integrat

Jurnal

- 21 aprilie: alegere proiect
- 5 mai: comandare piese
- 12-17 mai: asamblare si scriere cod
- 20-27 mai: finalizare pagina wiki

Bibliografie/Resurse

radar_1_.pdf

Export to PDF