

Memòria Projecte final IOT: Alarma intel·ligent

Introducció:

La idea del nostre projecte era crear una alarma, però no una alarma normal sinó un de molt peculiar que es desplaça per el terra, d'aquesta manera l'usuari hauria de fer l'esforç de sortir del llit per apagar-la.

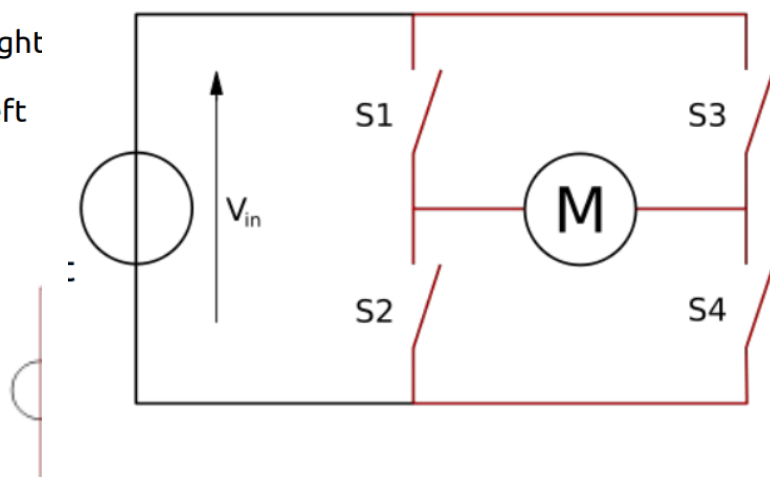
Materials utilitzats:

- Microcontrolador
- Sensor de distància x2
- Motors
- Xassís del robot
- Bateria
- Altaveu
- H-Bridge

El material final utilitzat va anar variant a mesura que vam anar construint el nostre projecte, ja que inicialment volíem utilitzar relés per governar els motors, però vam veure que amb un H-Bridge tindríem un millor control dels nostres motors i els podríem governar millor, perquè ens donaven l'oportunitat de poder fer que les rodes giressin endavant i endarrere.

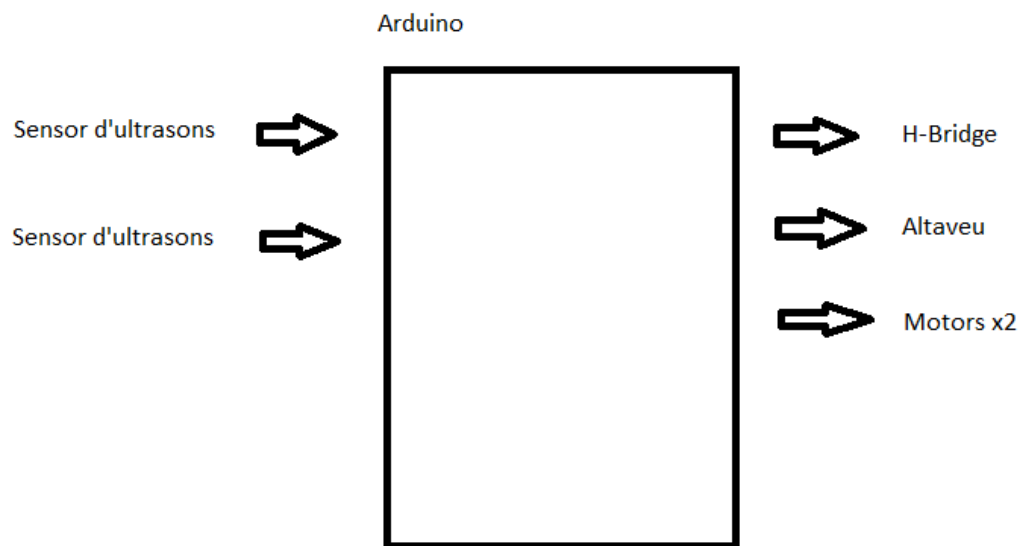
H-Bridge

S1	S2	S3	S4	Result
1	0	0	1	Motor moves right
0	1	1	0	Motor moves left
0	0	0	0	Motor coasts
0	1	0	1	Motor brakes
1	0	1	0	Motor brakes
1	1	0	0	Short circuit
0	0	1	1	Short circuit
1	1	1	1	Short circuit

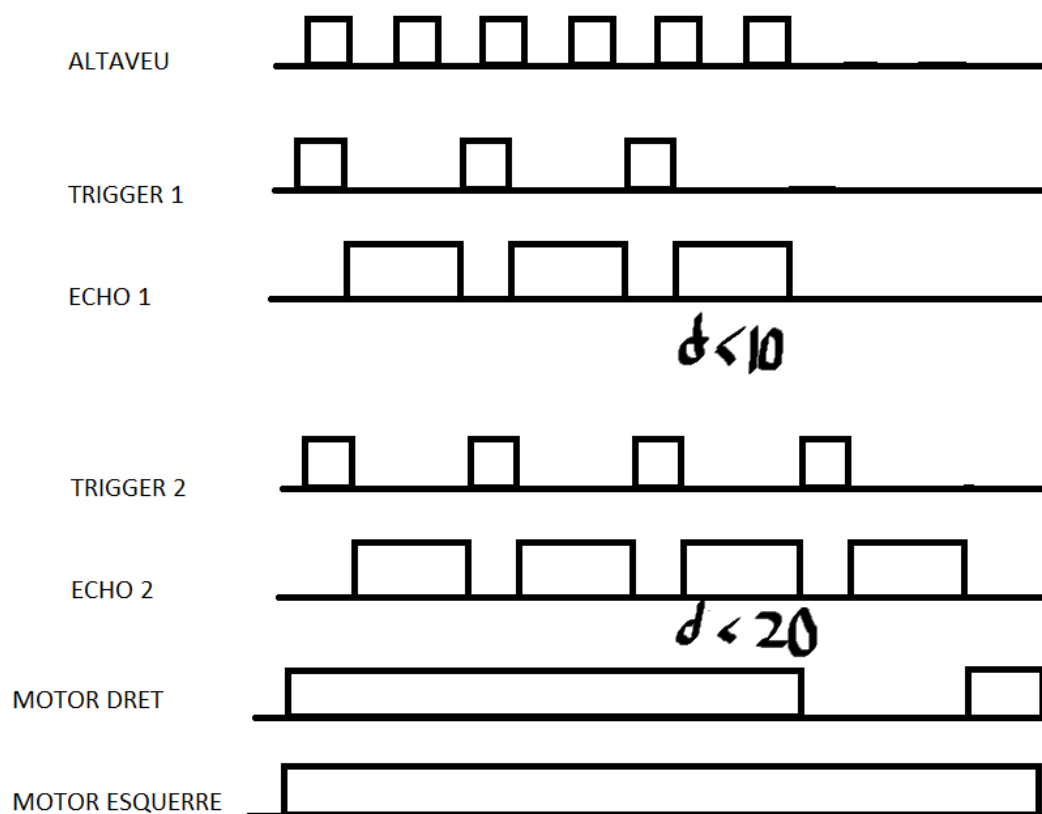


També a l'últim moment vam tenir un problema amb el xassís, ja que la fusta que teníem era massa gruixuda, els claus no travessaven del tot i els motors no quedaven ben ajuntats, així que vam decidir utilitzar la carpeta de l'Uvic tallada a mida i ens va donar un bon resultat.

Entrades i sortides:



Cronograma:



Codis:

Per fer els codis primer vam fer els funcionaments per parts un es va encarregar de la part de l'altaveu i la melodia i l'altre va investigar la mobilitat del robot i el funcionament dels motors.

Per una banda, vam començar la part dels altaveus, vam intentar trobar una melodia que ens funcionés com alarma, un cop vam tenir el codi fet i enllestit, vam incorporar el detector d'ultrasons per fer que quan acostéssis la mà l'alarma s'apagués.

I, per altra banda, vam treballar amb la mobilitat del nostre robot, teníem uns motors que s'alimentaven amb 5V i el principi utilitzàvem relés per governar-los, vam veure que necessitàvem dues piles una per a cada relé pel funcionament dels motors, i vam decidir canviar, vam deixar de banda els relés i vam utilitzar un H-Bridge que a part de tenir un ús molt fàcil ens permetia tenir governar millor els nostres motors, ja que a diferència dels relés podríem fer que les rodes gressin endavant i endarrere.

A continuació us deixarem els codis individuals i després el codi final.

Codi individual altaveus amb detector d'ultrasons:

```
#define trig 7 // Emissor de pols
#define echo 6 // Receptor de l'eco
#define buzzer 12 //Zumbador

int final=0;

void setup() {
  pinMode(trig, OUTPUT); //Emisor
  pinMode(echo, INPUT); //Receptor
  pinMode(buzzer, OUTPUT); //Emissor
}

void loop() {
  long duration, distance;
  digitalWrite(trig, LOW); // 2 microsegons en parat per un pols net
  delay(2);
  digitalWrite(trig, HIGH); //enviem un pols de 5 microsegons
  delay(5);
  digitalWrite(trig, LOW); //parem

  duration = pulseIn(echo, HIGH); //tiemps que la senyal tarda en tornar en microsegundos

  distance = (duration/2)*0.0343; //(343 m/s = 0.0343 cm/microsegons)

  int i=0;

  while(i<50 and distance >6 and final==0){ //la variable i son iteracions per posar un
número, si la distància son menys de 6cm la variable final s'actualitza i el programa no
funciona més
    tone(buzzer, 500);
    delay(100);
    tone (buzzer, 700);

    digitalWrite(trig, LOW);
    delay(2);
    digitalWrite(trig, HIGH);
    delay(5);
    digitalWrite(trig, LOW);
    duration = pulseIn(echo, HIGH);
    distance = (duration/2)*0.0343;
    i++;
    if (distance<6){
      final=1;
    }
  }

  noTone(buzzer);
  delay (5000);
}
```

Codi inicial amb relés (finalment no utilitzat):

```
int relay = 2;
void setup () {
  pinMode (relay, OUTPUT); Serial.begin (9600);
}
```

```
void loop () {
  digitalWrite(relay, HIGH); // enviar alta al rele
  Serial.println("Rele accionat");
  delay (5000);
```

```
  digitalWrite(relay, LOW); // enviar baixa al rele
  Serial.println("Rele no accionat");
  delay (1000);
```

```
}
```

```
const int Trigger = 10;
const int Echo = 9;
const int button = 7;
```

```
Serial.println("Encés");
long temps;
long distancia;
```

```
digitalWrite(Trigger, HIGH);
delayMicroseconds(10);
digitalWrite(Trigger, LOW);
```

```
temps = pulseIn(Echo, HIGH);
distancia = temps/59;
```

```
Serial.print("Distancia: ");
Serial.print(distancia);
Serial.print("cm");
Serial.println();
delay(100);
```

```
if(distancia<15){
  servo1.write(180);
  musica(160);
}
```

```
else if(distancia<30){
  servo1.write(135);
  musica(120);
}
```

```
else if(distancia<50){
  servo1.write(90);
  musica(80);
}
```

```
else if(distancia>50){
  servo1.write(0);
}
```

Codi individual motors amb detector d'ultrasons:

```
#define Trigger2 12 // Emissor de pols
#define Echo2 13 // Receptor de l'eco
#define buzzer 5 //Zumbador

int pwm_speedA = 255;
int pwm_speedB = 255;

const int Trigger = 4;
const int Echo = 2;

int final=0;
void setup()
{
  Serial.begin(9600);
  pinMode(3,OUTPUT);
  pinMode(6,OUTPUT);
  pinMode(7,OUTPUT);

  pinMode(11, OUTPUT);
  pinMode(10, OUTPUT);
  pinMode(8, OUTPUT);

  pinMode(Trigger, OUTPUT);
  pinMode(Echo, INPUT);
  digitalWrite(Trigger, LOW);

  pinMode(Trigger2, OUTPUT); //Emisor
  pinMode(Echo2, INPUT); //Receptor
  pinMode(buzzer, OUTPUT); //Emissor

}
void loop()
{
  long t;
  long d;

  long duration, distance;
  digitalWrite(Trigger2, LOW); // 2 microsegons en parat per un pols net
  delay(2);
```

```

digitalWrite(Triple2, HIGH); //enviem un pols de 5 microsegons
delay(5);
digitalWrite(Triple2, LOW); //parem

duration = pulseIn(Echo2, HIGH); //tiemps que la senyal tarda en tornar en microsegundos

distance = (duration/2)*0.0343; //(343 m/s = 0.0343 cm/microsegons)

int i=0;

digitalWrite(Triple, HIGH);
delayMicroseconds(10);
digitalWrite(Triple, LOW);

t = pulseIn(Echo, HIGH);
d = t/59;

Serial.print("Distancia: ");
Serial.print(d);
Serial.print("cm");
Serial.println();
delay(100);

if(d>20){
  digitalWrite(6,LOW);
  digitalWrite(7,HIGH);
  analogWrite(3, pwm_speedB);

  digitalWrite(8,HIGH);
  digitalWrite(10,LOW);
  analogWrite(11, pwm_speedA);
}
else{
  digitalWrite(6,LOW);
  digitalWrite(7,HIGH);
  analogWrite(3, pwm_speedB);

  digitalWrite(8,LOW);
  digitalWrite(10,LOW);
  analogWrite(11, pwm_speedA);
  delay(2000);
}
}

```

Codi final del projecte:

```
#define Trigger2 12 // Emissor de pols
#define Echo2 13 // Receptor de l'eco
#define buzzer 5 //Zumbador
```

```
int pwm_speedA = 255;
int pwm_speedB = 255;
```

```
const int Trigger = 4;
const int Echo = 2;
```

```
int final=0;
void setup()
```

```
{
  Serial.begin(9600);
  pinMode(3,OUTPUT);
  pinMode(6,OUTPUT);
  pinMode(7,OUTPUT);
```

```
  pinMode(11, OUTPUT);
  pinMode(10, OUTPUT);
  pinMode(8, OUTPUT);
```

```
  pinMode(Trigger, OUTPUT);
  pinMode(Echo, INPUT);
  digitalWrite(Trigger, LOW);
```

```
  pinMode(Trigger2, OUTPUT); //Emisor
  pinMode(Echo2, INPUT); //Receptor
  pinMode(buzzer, OUTPUT); //Emissor
```

```
}
```

```
void loop()
```

```
{
  long t;
  long d;
```

```
  long duration, distance;
  digitalWrite(Trigger2, LOW); // 2 microsegons en parat per un pols net
  delay(2);
  digitalWrite(Trigger2, HIGH); //enviem un pols de 5 microsegons
  delay(5);
  digitalWrite(Trigger2, LOW); //parem
```



```
duration = pulseIn(Echo2, HIGH); //tiemps que la senyal tarda en tornar en microsegundos
```

```
distance = (duration/2)*0.0343; //(343 m/s = 0.0343 cm/microsegons)
```

```
int i=0;
```

```
digitalWrite(Triquer, HIGH);
```

```
delayMicroseconds(10);
```

```
digitalWrite(Triquer, LOW);
```

```
t = pulseIn(Echo, HIGH);
```

```
d = t/59;
```

```
Serial.print("Distancia: ");
```

```
Serial.print(d);
```

```
Serial.print("cm");
```

```
Serial.println();
```

```
delay(100);
```

```
while(i<50 and distance >6 and final==0){ //la variable i son iteracions per posar un  
número, si la distància son menys de 6cm la variable final s'actualitza i el programa no  
funciona més
```

```
tone(buzzer, 500);
```

```
delay(100);
```

```
tone (buzzer, 700);
```

```
digitalWrite(Triquer2, LOW);
```

```
delay(2);
```

```
digitalWrite(Triquer2, HIGH);
```

```
delay(5);
```

```
digitalWrite(Triquer2, LOW);
```

```
duration = pulseIn(Echo2, HIGH);
```

```
distance = (duration/2)*0.0343;
```

```
i++;
```

```
if(d>20 && final==0){
```

```
digitalWrite(6,LOW);
```

```
digitalWrite(7,HIGH);
```

```
analogWrite(3, pwm_speedB);
```

```
digitalWrite(8,HIGH);
```

```
digitalWrite(10,LOW);
```

```
analogWrite(11, pwm_speedA);
```

```
}
```

```
else{
```

```
digitalWrite(6,LOW);  
digitalWrite(7,HIGH);  
analogWrite(3, pwm_speedB);
```

```
digitalWrite(8,LOW);  
digitalWrite(10,LOW);  
analogWrite(11, pwm_speedA);  
delay(2000);  
}
```

```
if (distance<6){  
    final=1;  
}  
}
```

```
noTone(buzzer);  
digitalWrite(6,LOW);  
digitalWrite(7,LOW);  
analogWrite(3, pwm_speedB);
```

```
digitalWrite(8,LOW);  
digitalWrite(10,LOW);  
analogWrite(11, pwm_speedA);  
delay (5000);  
}
```