

Pràctica 3: Parking Assistant

PROGRAMACIÓ DE MICROPROCESSADORS

Paula Dolado Aynie | Carles Gasia Romeu

Índex

1.1 Introducció al HC-SR04	2
Enunciat.....	2
Codi.....	2
Muntatge	2
Explicació.....	3
Link.....	3
1.2 Park Assistant.....	3
Enunciat.....	3
Codi.....	4
Muntatge	5
Explicació.....	5
Link.....	6
1.3 Park Assistant amb potenciòmetre	6
Enunciat.....	6
Codi.....	6
Muntatge	7
Explicació.....	8
Link.....	8

1.1 Introducció al HC-SR04

ENUNCIAT

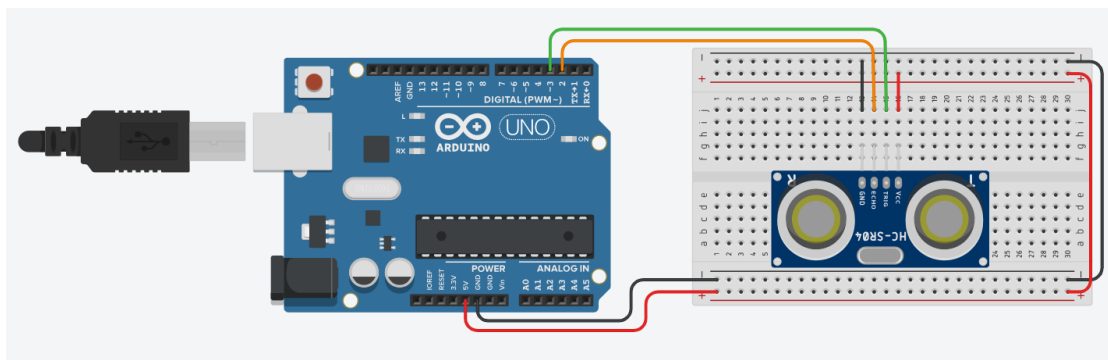
- Muntar un circuit amb un sensor de distància HC-SR04.
- Imprimir a la consola el valor de la distància en centímetres.

CODI

```
// C++ code
//
int echo = 2;
int trigger = 3;
int durada, distancia;
void setup() {
  Serial.begin(9600);
  pinMode(echo, INPUT);
  pinMode(trigger, OUTPUT);
}

void loop() {
  digitalWrite(trigger, LOW);
  delay(10);
  digitalWrite(trigger, HIGH);
  delay(0.01);
  digitalWrite(trigger, LOW);
  durada = pulseIn(echo, HIGH);
  distancia = durada / (29.2*2);
  Serial.println("Distancia: ");
  Serial.println(distancia);
  Serial.println("cm");
  delay(10);
}
```

Muntatge



Explicació

En primer lloc definirem en quins ports es connecten les dues connexions del echo i el trigger juntament amb les variables de temps i distancia.

Dins del setup declararem el trigger com a sortida y el echo al entrada.

Seguidament al iniciar el bucle partirem de que el trigger estigui en low, després d'una espera de 10ms generarem un pols i repetirem la espera però aquesta vegada de 0.01 microsegons fins tornar a disminuir el trigger en low. Al acabar aquestes operacions agafarem la durada amb el comandament de pulseIn, per calcular la distancia dividirem el temps per 29.2*2 en referencia a 29.2 dels microsegons que triga en recórrer un centímetre y 2 degut a que el so viatja d'anada i tornada.

Per acabar, imprimirem per pantalla aquesta distancia amb les seves unitats.

LINK

<https://www.tinkercad.com/things/eWMgKYNxxiX-practica3u/editel?returnTo=%2Fdashboard&sharecode=DhQsgCjcmJjeIbXuRyDN7nuwHoA8TGg8R2CMuOGVfto>

1.2 Park Assistant

ENUNCIAT

- Afegir un polsador de power on per encendre el sistema (simularia el fet de posar la marxa enrere del cotxe).
- Afegir un polsador de power off per apagar el sistema.
- Afegir un led que ajusti la seva brillantor mitjançant el PWM utilitzat en les pràctiques anteriors de manera que si es detecta una distancia igual o inferior a 30cm estigui encès al màxim i quan la distancia sigui igual o superior a 300cm aquest es mantingui al mínim (apagat). Modificar la brillantor en funció de la distancia entre aquests dos estats.
- Afegir un buzzer que amb el sistema encès: o Si la distancia és igual o superior a 300cm emetre un to de 30Hz de duració 200ms cada 1,5 segons. o Si la distancia és igual o inferior a 30cm emetre un to de 330Hz de duració 200ms cada 200ms o Que en les distàncies intermitges el delay entre els tons i la freqüència d'aquests sigui proporcional a la distància mesurada.
- Treure pel canal sèrie (consola)
- Amb això el que aconseguirem és un sistema on amb el sensor de distancia d'ultrasons i un led farem un sistema que ajudarà a detectar quan aparcuem si xocarem on no fent marxa enrere funcionant com el dels vehicles convencionals: A mesura que ens apropem a l'obstacle tindrem una senyal lluminosa més intensa (el led) i per altra banda una senyal sonora que també ens anirà avisant si podem o no tirar més marxa enrere.

CODI

```
// C++ code
//
int Trigger = 2;
int Echo = 3;
int LED = 5;
int but1 = A1;
int but2 = A0;
int zumb = 8;
int on = 0;
long temps;
long dist;

void setup(){
  Serial.begin(9600);
  pinMode(but1,OUTPUT);
  pinMode(but2,OUTPUT);
  pinMode(zumb,OUTPUT);
  pinMode(LED,OUTPUT);
  pinMode(Trigger, OUTPUT);
  pinMode(Echo, INPUT);
  digitalWrite(Trigger, LOW);
}

void loop(){
  if(digitalRead(but1)== 1){
    on = 1;
  }
  if(digitalRead(but2)== 1){
    on = 0;
  }

  if(on == 1){
    digitalWrite(Trigger, HIGH);
    delayMicroseconds(10);
    digitalWrite(Trigger, LOW);

    temps = pulseIn(Echo, HIGH);
    dist = temps/(29.2*2);

    if(dist <= 30){
      analogWrite(LED,255);
      tone(zumb,330,200);
      delay(200);
    }

    if(dist>30 && dist<300){
      analogWrite(LED,255 - (abs(dist-45)));
      tone(zumb,(330 - dist),200);
    }
  }
}
```

```

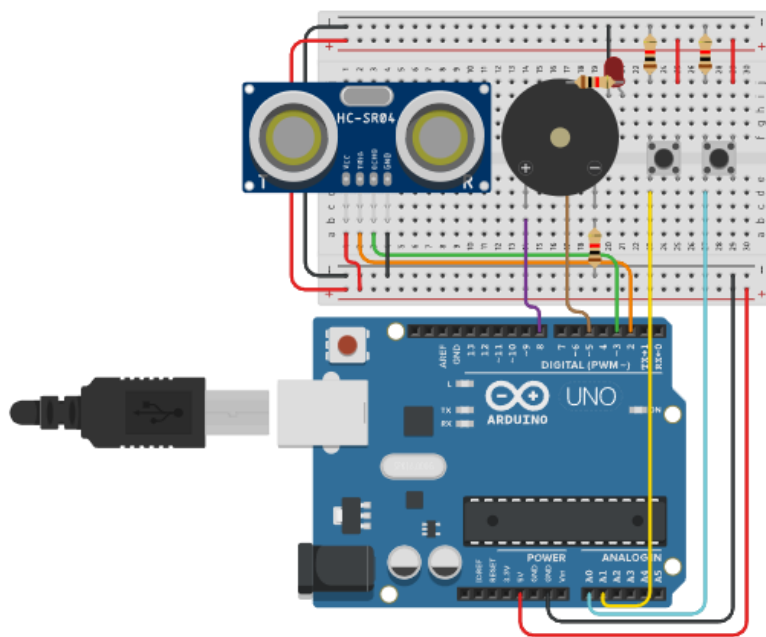
    delay(200 - (dist/6));
  }

  if(dist >= 300){
    analogWrite(LED,o);
    tone(zumb,30,200);
    delay(150);
  }

  Serial.print("Distancia: ");
  Serial.print(dist);
  Serial.print("cm");
  Serial.println();
}
}

```

Muntatge



Explicació

Com de costum iniciarem el codi del programa declarant les connexions de cada dispositiu. A continuació al setup majoritàriament se'ls hi indicarà la connexió de sortida a excepció del echo.

En primer lloc dins del bucle podem observar uns condicionals on si els polsadors son activats encendrem el sistema mentre que l'altre s'encarregarà d'apagar-ho modificat la variable "on".

Seguidament, en cas de que la variable “on” sigui 1, es a dir, volem encendre els sistema aplicarem el codi de la activitat 1.1 on inicialment volem capturar la distancia. Depenen d'aquesta distancia ens trobem 3 etapes diferents.

En cas de que la distancia sigui menor a 30cm modificarem la lluminositat del led per a que sigui la màxima, emetrem un to de 330Hz durant 200ms i farem una pausa abans de realitzar una altre operació per a poder distingir bé el so. En el segon cas obtenim que la distancia és a un rang entre 30cm-300cm on tant el led com el to serà proporcional a la distancia en la que es trobar, degut a això realitzarem una resta de la freqüència, la intensitat i el delay amb aquesta mitjançant una regla de tres. Com a últim cas, si la distància és major a 300cm observarem el led apagat i un to a 30Hz durant 200ms.

Finalment, imprimirem per pantalla la distància en cm.

LINK

<https://www.tinkercad.com/things/8f2CUoQhZHg-ex2prac3/editel?returnTo=%2Fthings%2F8f2CUoQhZHg-ex2prac3&sharecode=ajKo2YI-lGyCb3MFU62DBesbKooAYSXRlbPrPnuCHB4>

1.3 Park Assistant amb potenciòmetre

ENUNCIAT

Es tracta de fer exactament el mateix que a l'activitat 1.2 amb la diferencia que en comptes d'un sensor de ultrasons posarem un potenciòmetre on la posició ens indicarà la distància. De manera que, quan el potenciòmetre estigui al mínim tindrem una distancia de detecció de 300cm i quan estigui al màxim de 30cm. Aquesta és l'activitat que heu de muntar amb l'Arduino que teniu en cas de voler entregar el muntatge.

CODI

```
int poten = A2;
int LED = 5;
int but1 = A1;
int but2 = A0;
int zumb = 8;
int on = 0;
float dist;

void setup(){
  Serial.begin(9600);
  pinMode(but1,OUTPUT);
  pinMode(but2,OUTPUT);
  pinMode(zumb,OUTPUT);
  pinMode(LED,OUTPUT);
}
```

```

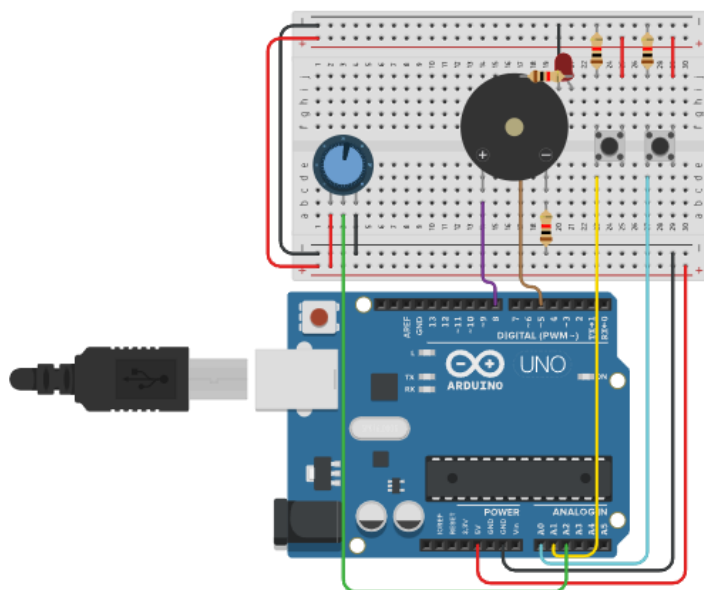
void loop(){
  if(digitalRead(but1)== 1){
    on = 1;
  }
  if(digitalRead(but2)== 1){
    on = 0;
  }
  if(on == 1){

    int v_poten = analogRead(poten);
    dist = 300-((300.0/1023.0)*v_poten);

    if(dist <= 30){
      analogWrite(LED,255);
      tone(zumb,330,200);
      delay(200);
    }
    if(dist>30 && dist<300){
      analogWrite(LED,255 - (abs(dist-45)));
      tone(zumb,(330 - dist),200);
      delay(200 - (dist/6));
    }
    if(dist >= 300){
      analogWrite(LED,0);
      tone(zumb,30,200);
      delay(150);
    }
    Serial.print("Distancia: ");
    Serial.print(dist);
    Serial.print("cm");
    Serial.println();
  }
}

```

Muntatge



Explicació

En lo referent al tercer muntatge, es molt similar al segon. En aquest s'ha de substituir l'anteriorment utilitzat sensor d'ultrasons per un potenciòmetre que realitzarà la mateixa funció.

Pel que fa al codi, hem modificat la variable distancia. Hem utilitzat una regla de tres per a fer que el rang del potenciòmetre de 1023-o sigui proporcional a la distancia de 30-300. La resta del programa manté el funcionament.

LINK

<https://www.tinkercad.com/things/29TqAbclL8C-copy-of-ex2prac3/editel?sharecode=Vd9MyqEhiOPoSFLtKUIL29CeNtDBrMuM3EhvfBCoPs>