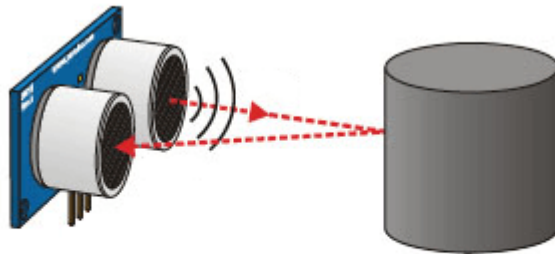


EJERCICIOS DE ARDUINO

Se desea hacer una alarma para el hogar basada en un sensor de ultrasonidos y un microcontrolador Arduino UNO. El objetivo es advertir de la presencia de intrusos dentro de nuestra casa.

Para realizar este proyecto disponemos de:

-Un sensor de ultrasonidos HC-SR04: se activa enviando un pulso por el pin Trigger, y nos ofrece un pulso de respuesta en el pin echo cuando el pulso emitido por el Trigger rebota contra un objeto. El objetivo es medir el tiempo que tarda en rebotar el pulso contra un objeto, relacionando este tiempo con la velocidad del sonido podemos obtener la distancia al objeto:



$$\text{Tiempo} = 2 * (\text{Distancia} / \text{Velocidad})$$

$$\text{Distancia} = \text{Tiempo} \cdot \text{Velocidad} / 2$$

$$\text{Distancia}(cm) = \frac{\text{Tiempo}(\mu s)}{29.2 \cdot 2}$$

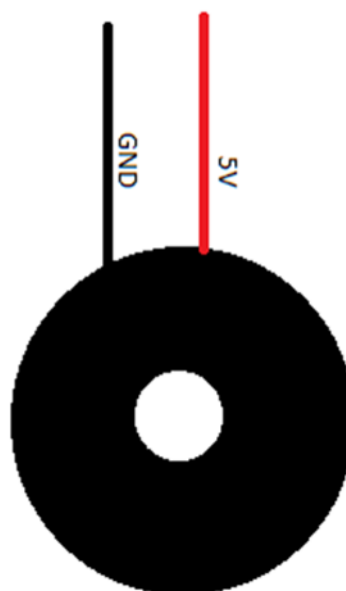
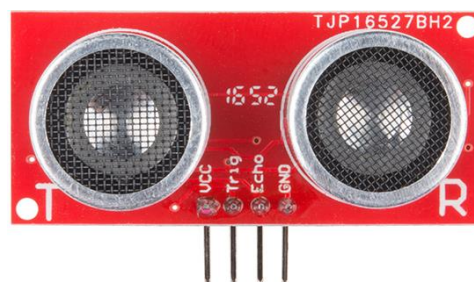
Nota: el tiempo de retardo entre pulsos se puede obtener en Arduino a través de la función **pulseIn**

- Un microcontrolador Arduino uno.
- Una sirena que se activa alimentándola a 5V.
- Fuente de alimentación de 5V

Se pide:

- Esquema eléctrico (En el espacio indicado en la hoja posterior)
- Diagrama de flujo correspondiente al código aportado.

DIAGRAMA DE CONEXIONES



```
1. #define EchoPin 5
2. #define TriggerPin 6
3. #define AlarmaPin 9
4.
5. void setup() {
6.     ConfigurarPines();
7.
8. }
9.
10. void loop() {
11.     float distanciaCm = DistanciaEnCm(TriggerPin, EchoPin);
12.     if (distanciaCm < 50) {
13.         digitalWrite(AlarmaPin, HIGH);
14.     }
15.     else {
16.         digitalWrite(AlarmaPin, LOW);
17.     }
18. }
19.
20. void ConfigurarPines() {
21.     pinMode(TriggerPin, OUTPUT);
22.     pinMode(AlarmaPin, OUTPUT);
23.     pinMode(EchoPin, INPUT);
24. }
25.
26. float DistanciaEnCm(int Trigger, int Echo) {
27.     long duracion;
28.     float distanciaCm;
29.
30.     digitalWrite(Trigger, LOW);
31.     delayMicroseconds(4);
32.     digitalWrite(Trigger, HIGH);
33.     delayMicroseconds(10);
34.     digitalWrite(Trigger, LOW);
35.
36.     duracion = pulseIn(Echo, HIGH); //medimos el tiempo entre
    pulsos, en microsegundos
37.
```

```
38. distanciaCm = duracion * 10 / 292 / 2; //convertimos a
    distancia, en cm
39. return distanciaCm;
40. }
```