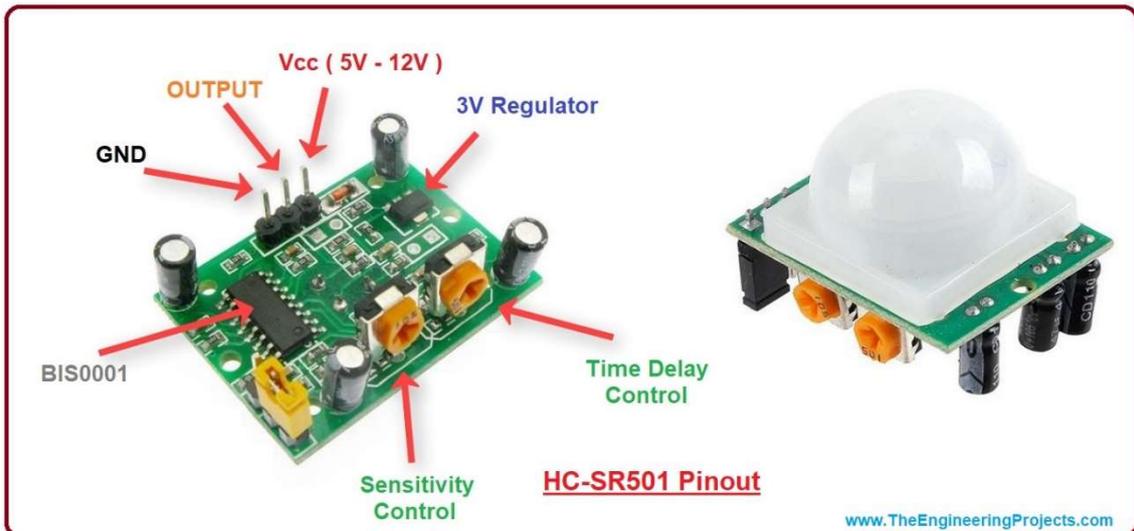


PRÁCTICA 3 ARDUINO “INSTALACIONES DOMÓTICAS”

PARTE 1: Sensor de movimiento

Se trata de programar un sensor de movimiento de tal forma que en cuanto se detecte un movimiento mantengamos la salida 3 a nivel alto un mínimo de 4 segundos y de lo contrario se mantendrá la salida a nivel bajo. En dicha salida conectaremos un diodo led con su correspondiente resistencia de protección.



(A modo de recordatorio podríamos cambiar el led por un relé para controlar algún otro dispositivo de carga mayor.)

Será necesario establecer un tiempo en el setup, de unos 20 segundos, antes de ejecutar el programa principal con el fin de estabilizar en lo posible el sensor.

PRÁCTICA 3 ARDUINO “INSTALACIONES DOMÓTICAS”

PARTE 2: Ultrasonidos

Además, en la misma práctica añadiremos un módulo de ultrasonidos como el de la imagen, que nos permitirá calcular la distancia a la que se encuentra un objeto calculando el tiempo entre la emisión y la recepción de un sonido (Para dicho cálculo se tomará el valor entregado por el dispositivo mediante un pulso alto en la entrada ECHO y se usará la función **pulseIn** para guardar el cálculo de duración en una variable DURACIÓN (int). A continuación, esa variable se dividirá por 58,2 (valor establecido por el fabricante para obtener la distancia a la que está el objeto) y se guardará en una variable llamada DISTANCIA (int)

Una vez calculado el valor deberemos mostrarlo por el monitor serie cada 2 segundos.

NOTAS:

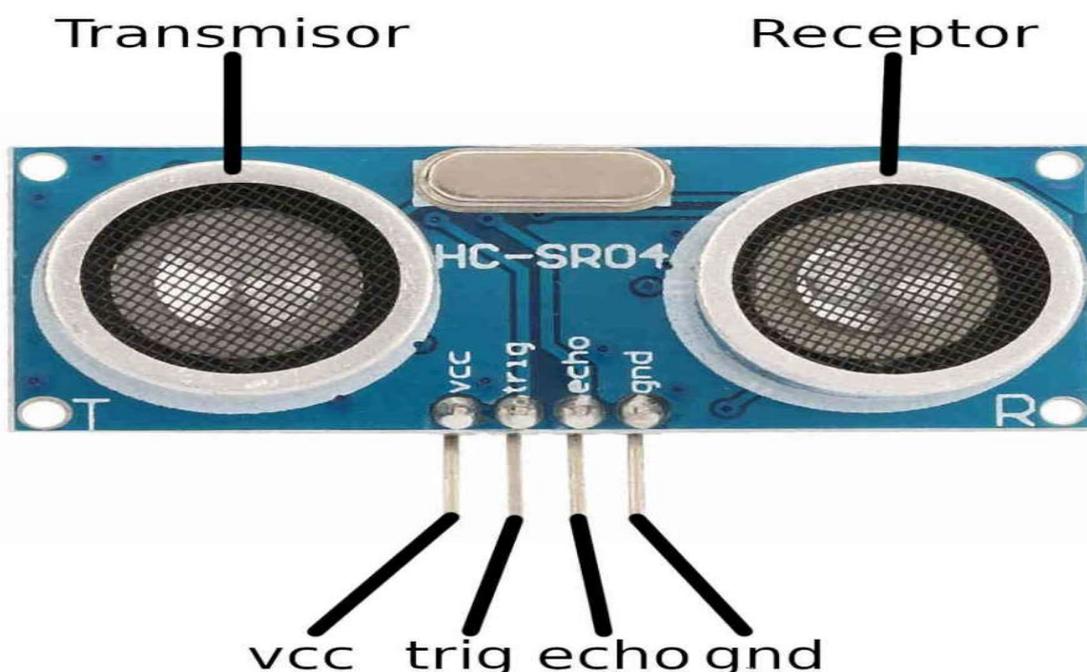
El sonido se emite a 40Khz, imperceptible para el oído humano que máximo llegaría a 20Khz.

El rango de medición será de 2 a 200cms

Trig: Será la señal que le enviaremos al dispositivo para que realice un envío

Echo: Será una señal que enviaremos de confirmación de recepción

Vcc y GND: Serán la alimentación del dispositivo.



PRÁCTICA 3 ARDUINO “INSTALACIONES DOMÓTICAS”

PARTE 3: Mixto de ambos programas

Trataremos de que sólo se lancen mediciones cuando el sensor de movimiento hay a detectado algún cambio por movimiento, si no hay movimiento no debe haber mediciones.

Usaremos tanto el IDE de Arduino como Tinkercad y Arduino Blocks

Objetivos:

- *Conexionado de un sensor de movimiento*
- *Conexionado de un dispositivo de ultrasonidos*
- *Repaso de órdenes de control de programa como: If else, delay, digitalwrite, digitalread y uso del monitor serie (activación de la comunicación e impresión de datos por pantalla)*
- *Conocer la orden pulseIn*
- *Realizar operaciones sobre variables para usar los datos calculados u obtenidos*

PRÁCTICA 3 ARDUINO “INSTALACIONES DOMÓTICAS”

PROGRAMA PARA EL SENSOR DE MOVIMIENTO

```
int PIR=7;
int LED=3;
int ESTADO=0;

void setup() {
  // put your setup code here, to run once:

  pinMode(PIR, INPUT);
  pinMode(LED,OUTPUT);
  delay(20000); // TIEMPO NECESARIO PARA ESTABILIZAR EL SENSOR SEGÚN
  FABRICANTE
}

void loop() {
  // put your main code here, to run repeatedly:
  ESTADO=digitalRead(PIR);
  if (ESTADO==HIGH){
    digitalWrite (LED, HIGH);
    delay (4000);
  }
  else {
    digitalWrite(LED, LOW);
  }
}
```

PRÁCTICA 3 ARDUINO “INSTALACIONES DOMÓTICAS”

PROGRAMA PARA EL ULTRASONIDOS

```
int TRIG=7;
int LED=3;
int ECO=9;
int DURACION;
int DISTANCIA;

void setup() {
  // put your setup code here, to run once:

  pinMode(ECO, INPUT);
  pinMode(TRIG,OUTPUT);
  pinMode(LED,OUTPUT);
  Serial.begin(9600); //
}

void loop() {
  // put your main code here, to run repeatedly:
  digitalWrite (TRIG, HIGH);
  delay (1);
  digitalWrite (TRIG, LOW); // Pulso de 1ms necesario para realizar el
envío
  DURACION = pulseIn (ECO, HIGH);
  DISTANCIA = DURACION / 58.2 ;
  Serial.println(DISTANCIA);
  delay(2000);
  if (DISTANCIA <=20 && DISTANCIA >=0) {
    digitalWrite (LED, HIGH);
    delay (DISTANCIA*10);
    digitalWrite(LED, LOW);
  }
}
```