



A.3.1 Actividad de aprendizaje

Circuito sensor de detección de objetos, utilizando un Arduino, un sensor ultrasonico y un Display I2C LCD 16x2.

Instrucciones

- Realizar un sistema simulado, capaz de detectar la presencia y/o ausencia de un objeto, a través de un circuito electrónico, utilizando un Arduino, y un **Sensor Ultrasonico HC-SR04**.
- Toda actividad o reto se deberá realizar utilizando el estilo **Markdown con extension .md** y el entorno de desarrollo VSCode, debiendo ser elaborado como un documento **single page**, es decir si el documento cuanta con imágenes, enlaces o cualquier documento externo debe ser accedido desde etiquetas y enlaces, y debe ser nombrado con la nomenclatura **A3.1_NombreApellido_Equipo.pdf**.
- Es requisito que el .md contenga una etiqueta del enlace al repositorio de su documento en GITHUB, por ejemplo **Enlace a mi GitHub** y al concluir el reto se deberá subir a github.
- Desde el archivo **.md** exporte un archivo **.pdf** que deberá subirse a classroom dentro de su apartado correspondiente, sirviendo como evidencia de su entrega, ya que siendo la plataforma **oficial** aquí se recibirá la calificación de su actividad.
- Considerando que el archivo .PDF, el cual fue obtenido desde archivo .MD, ambos deben ser idénticos.
- Su repositorio ademas de que debe contar con un archivo **readme.md** dentro de su directorio raíz, con la información como datos del estudiante, equipo de trabajo, materia, carrera, datos del asesor, e incluso logotipo o imágenes, debe tener un apartado de contenidos o indice, los cuales realmente son ligas o **enlaces a sus documentos .md**, *evite utilizar texto* para indicar enlaces internos o externo.
- Se propone una estructura tal como esta indicada abajo, sin embargo puede utilizarse cualquier otra que le apoye para organizar su repositorio.

```
- readme.md
- blog
  - C3.1_TituloActividad.md
  - C3.2_TituloActividad.md
  - C3.3_TituloActividad.md
  - C3.4_TituloActividad.md
  - C3.5_TituloActividad.md
  - C3.6_TituloActividad.md
  - C3.7_TituloActividad.md
  - C3.8_TituloActividad.md
- img
- docs
  - A3.1_TituloActividad.md
  - A3.2_TituloActividad.md
  - A3.3_TituloActividad.md
```

Desarrollo

1.Utilizar el siguiente listado de materiales para la elaboración de la actividad

Cantidad	Descripción
1	Sensor Ultrasonico HC-SR04
2	Resistencia de 1.2k
1	Diodo led Rojo
1	Fuente de voltaje de 5V
1	1 Potenciómetro 10k
1	Arduino UNO

1. Basado en la imagen que se muestra, ensamble dentro del simulador a utilizar, el circuito electrónico indicado en la **Figura 1**.

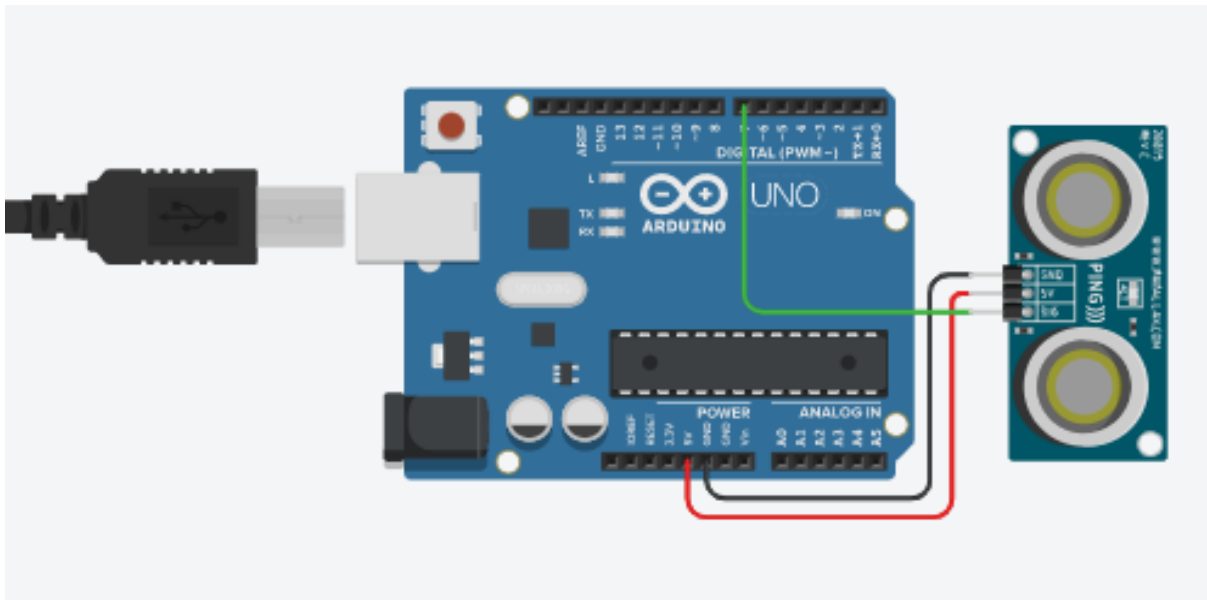


Figura 1 Sensor Ultrasonico

2. Realice el programa que permita a través de una de las entradas del Arduino, recibir el valor que registra el **Sensor Ultrasonico** al acercarse un objeto a distintas distancias.

```
// Esta constante no cambia. Es el pin que corresponde a la salida del sensor
const int pingPin = 7;

void setup() {
  // Inicializa la comunicacion serial a 9600 bits por segundo:
  Serial.begin(9600);
}

void loop() {

  // Establece variables para la duracion de ping, y la distancia resultante en
  cm:
  long duration, cm;
```

```

// El PING))) es activado por un pulso HIGH de 2 o mas microsegundos.
// Da un pulso LOW corto antes de dar el HIGH para dar el pulso HIGH limpio:
pinMode(pingPin, OUTPUT);
digitalWrite(pingPin, LOW);
delayMicroseconds(2);
digitalWrite(pingPin, HIGH);
delayMicroseconds(5);
digitalWrite(pingPin, LOW);

// El mismo pin es usado para leer la señal de PING))) : un pulso HIGH donde la
duracion
// es el tiempo en microsegundos desde el envio de la señal hasta la recepcion
del eco
// cuando toque un objeto
pinMode(pingPin, INPUT);
duration = pulseIn(pingPin, HIGH);

// Convierte el tiempo a distancia

// La velocidad del sonido es de 340 m/s o 29 microsegundos por centimetro.
// El ping viaja ida y vuelta, entonces para encontrar la distancia a la que
tenemos el
// objeto tomamos la mitad de la distancia viajada.
cm = duration / 29 / 2;;

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

delay(100);
}

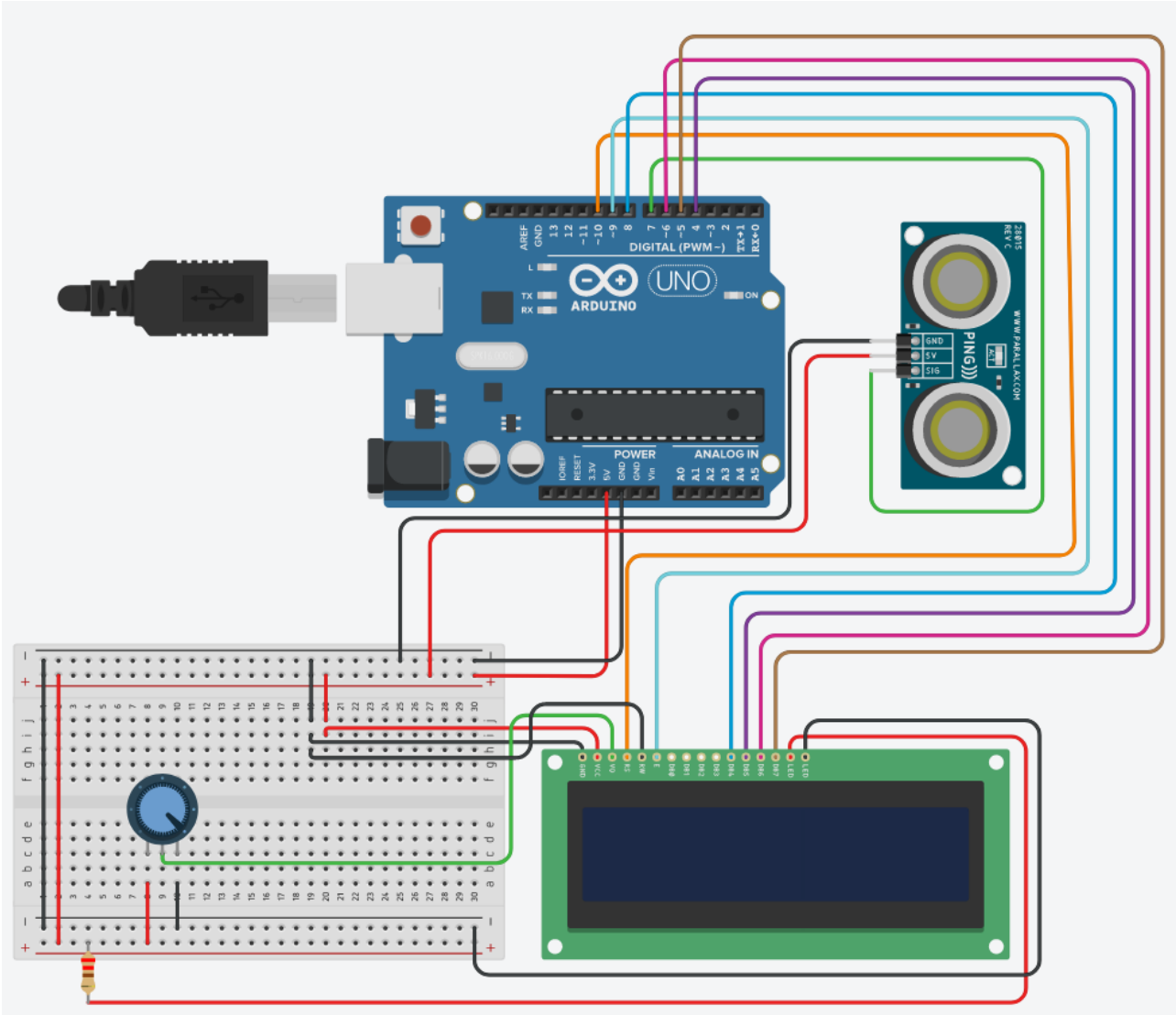
```

3. Considerando que el sensor ultrasonico tiene un rango mínimo y máximo de detección basado en el tiempo de retorno de la señal sonora, que valores se obtienen en la simulación bajo las **siguientes condiciones**:

Número	Condición 1	Condición 2	¿El objeto es detectado?
1	5 cm de distancia al sensor	0 grados al eje perpendicular del sensor	Si
2	50 cm de distancia al sensor	35 grados al eje perpendicular del sensor	Si
3	100 cm de distancia al sensor	-35 grados al eje perpendicular del sensor	Si
4	5 cm de distancia al sensor	90 grados al eje perpendicular del sensor	No

Número	Condición 1	Condición 2	¿El objeto es detectado?
5	50 cm de distancia al sensor	-60 grados al eje perpendicular del sensor	Si
6	350 cm de distancia al sensor	0 grados al eje perpendicular del sensor	No

4. Una vez completados los puntos anteriores, agregue a la Figura 1, **un Display I2C 16x2 LCD**, y coloque la imagen del circuito completado hasta este apartado.



5. Al haber completado la integración del Display I2C, ajuste el programa que le permita a través del display mostrar el siguiente mensaje, **"Objetivo detectado a ? cm"**, y en caso de no lograr la detección indicar el mensaje **"Objetivo fuera de rango"**

```
// Incluye la libreria de la pantalla
#include <LiquidCrystal.h>

// Esta constante no cambia. Es el pin que corresponde a la salida del sensor
const int pingPin = 7;
```

```
//Coloca los parametros: (rs, enable, d4, d5, d6, d7)
const int rs = 10, en = 9, d4 = 8, d5 = 4, d6 = 6, d7 = 5;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

void setup() {
  // Inicializa la comunicacion serial a 9600 bits por segundo:
  Serial.begin(9600);
  //
  lcd.begin(16,2);
}

void loop() {

  // Establece variables para la duracion de ping, y la distancia resultante en cm
  long duration, cm;

  // El PING))) es activado por un pulso HIGH de 2 o mas microsegundos.
  // Da un pulso LOW corto antes de dar el HIGH para dar el pulso HIGH limpio:
  pinMode(pingPin, OUTPUT);
  digitalWrite(pingPin, LOW);
  delayMicroseconds(2);
  digitalWrite(pingPin, HIGH);
  delayMicroseconds(5);
  digitalWrite(pingPin, LOW);

  // El mismo pin es usado para leer la señal de PING))) : un pulso HIGH donde la
  duracion
  // es el tiempo en microsegundos desde el envio de la señal hasta la recepcion
  del eco
  // cuando toque un objeto
  pinMode(pingPin, INPUT);
  duration = pulseIn(pingPin, HIGH);

  // Convierte el tiempo a distancia

  // La velocidad del sonido es de 340 m/s o 29 microsegundos por centimetro.
  // El ping viaja ida y vuelta, entonces para encontrar la distancia a la que
  tenemos el
  // objeto tomamos la mitad de la distancia viajada.
  cm = duration / 29 / 2;;

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

  delay(100);

  //Condiciion en caso de que este fuera de rango
  if(cm == 336) {

    lcd.setCursor(0, 0);
    lcd.print("No detectado");
```

```

        lcd.setCursor(0, 1);
        lcd.print("                ");
    }
    else{

        // Seccion del LCD
        lcd.setCursor(0, 0);
        lcd.print("Detectado a: ");
        lcd.setCursor(0, 1);
        lcd.print(cm);
        lcd.print("cm");
        lcd.print(" ");

    }

}

```

6. Coloque aqui evidencias que considere importantes durante el desarrollo de la actividad.

7. Conclusiones

Marquez Millan Seashell Vanessa

Garcia Rosas Ivan

Pardo Cruz Jesús Ramón



Rubrica

Criterios	Descripción	Puntaje
Instrucciones	Se cumple con cada uno de los puntos indicados dentro del apartado Instrucciones?	10
Desarrollo	Se respondió a cada uno de los puntos solicitados dentro del desarrollo de la actividad?	60
Demostración	El alumno se presenta durante la explicación de la funcionalidad de la actividad?	20
Conclusiones	Se incluye una opinión personal de la actividad por cada uno de los integrantes del equipo?	10



[Ir a GitHub - Marquez Millan Seashell Vanessa](#)



[Ir a GitHub - Garcia Rosas Ivan](#)



[Ir a GitHub - Pardo Cruz Jesús Ramón](#)