



## 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
```

---

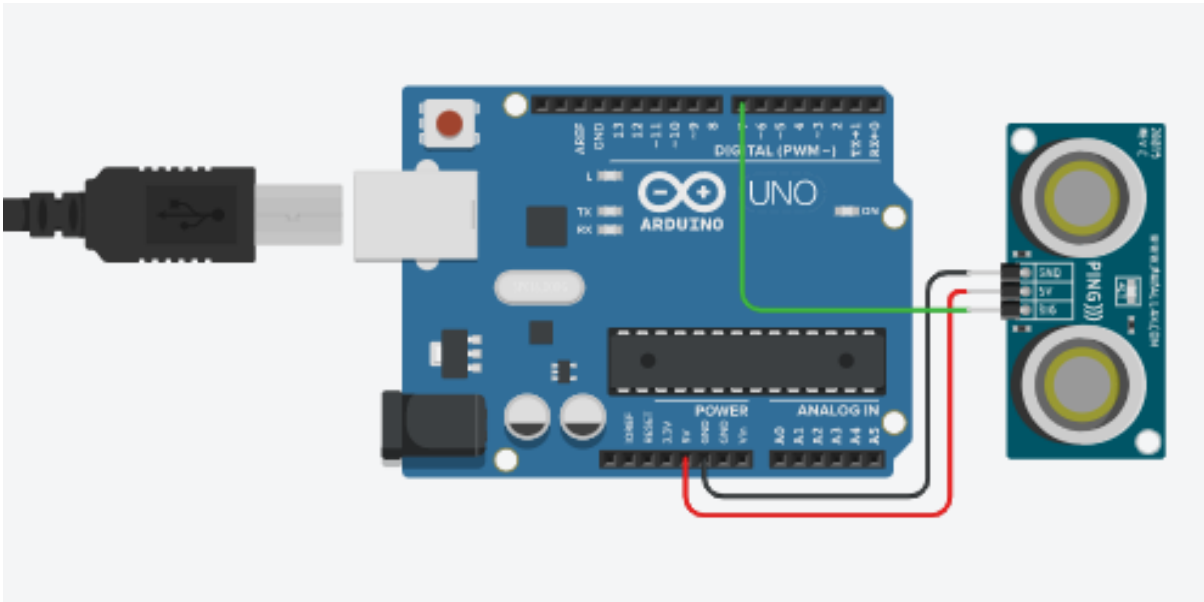
### Desarrollo

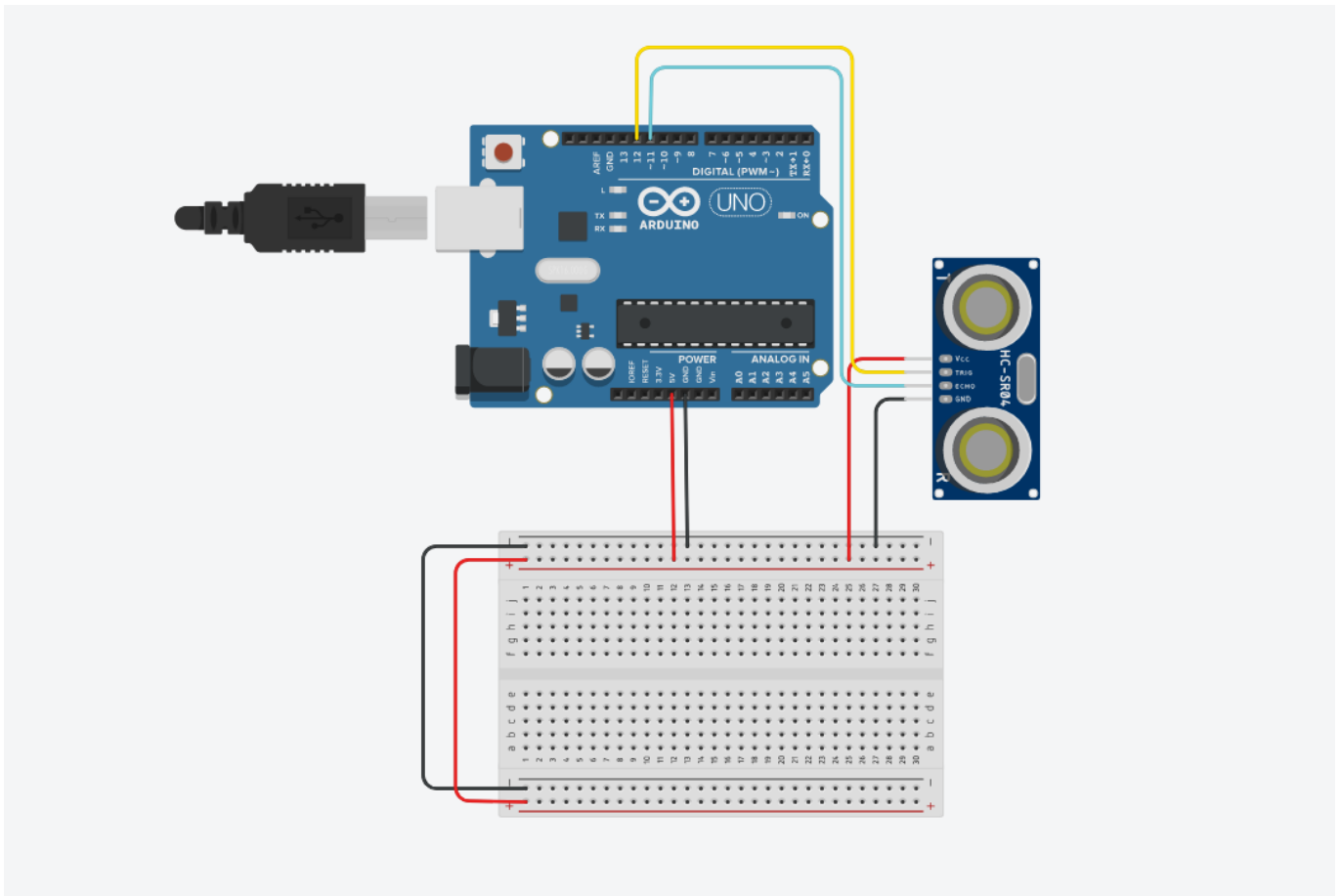
1. Utilizar el siguiente listado de materiales para la elaboración de la actividad

Cantidad	Descripción
1	Sensor Ultrasonico HC-SR04
1	Display LCD de 16x2
1	Fuente de voltaje de 5V
1	1 Potenciómetro 10k
1	Arduino UNO

2. 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





3. 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.

Texto

1 (Arduino Uno R3)

```
1 // DECLARACION DE VARIABLES PARA PINES
2 const int pinecho = 11;
3 const int pintrigger = 12;
4
5 // VARIABLES PARA CALCULOS
6 unsigned int tiempo, distancia;
7
8 void setup() {
9     // PREPARAR LA COMUNICACION SERIAL
10    Serial.begin(9600);
11    // CONFIGURAR PINES DE ENTRADA Y SALIDA
12    pinMode(pinecho, INPUT);
13    pinMode(pintrigger, OUTPUT);
14 }
15
16 void loop() {
17     // ENVIAR PULSO DE DISPARO EN EL PIN "TRIGGER"
18     digitalWrite(pintrigger, LOW);
19     delayMicroseconds(2);
20     digitalWrite(pintrigger, HIGH);
21     // EL PULSO DURA AL MENOS 10 uS EN ESTADO ALTO
22     delayMicroseconds(10);
23     digitalWrite(pintrigger, LOW);
24
25     // MEDIR EL TIEMPO EN ESTADO ALTO DEL PIN "ECHO" EL PULSO ES PROP
26     tiempo = pulseIn(pinecho, HIGH);
27
28     // LA VELOCIDAD DEL SONIDO ES DE 340 M/S O 29 MICROSEGUNDOS POR C
29     // DIVIDIMOS EL TIEMPO DEL PULSO ENTRE 58, TIEMPO QUE TARDA RECO
30     distancia = tiempo / 58;
31
32     // ENVIAR EL RESULTADO AL MONITOR SERIAL
33     Serial.print(distancia);
34     Serial.println(" cm");
35     delay(200);
36 }
```

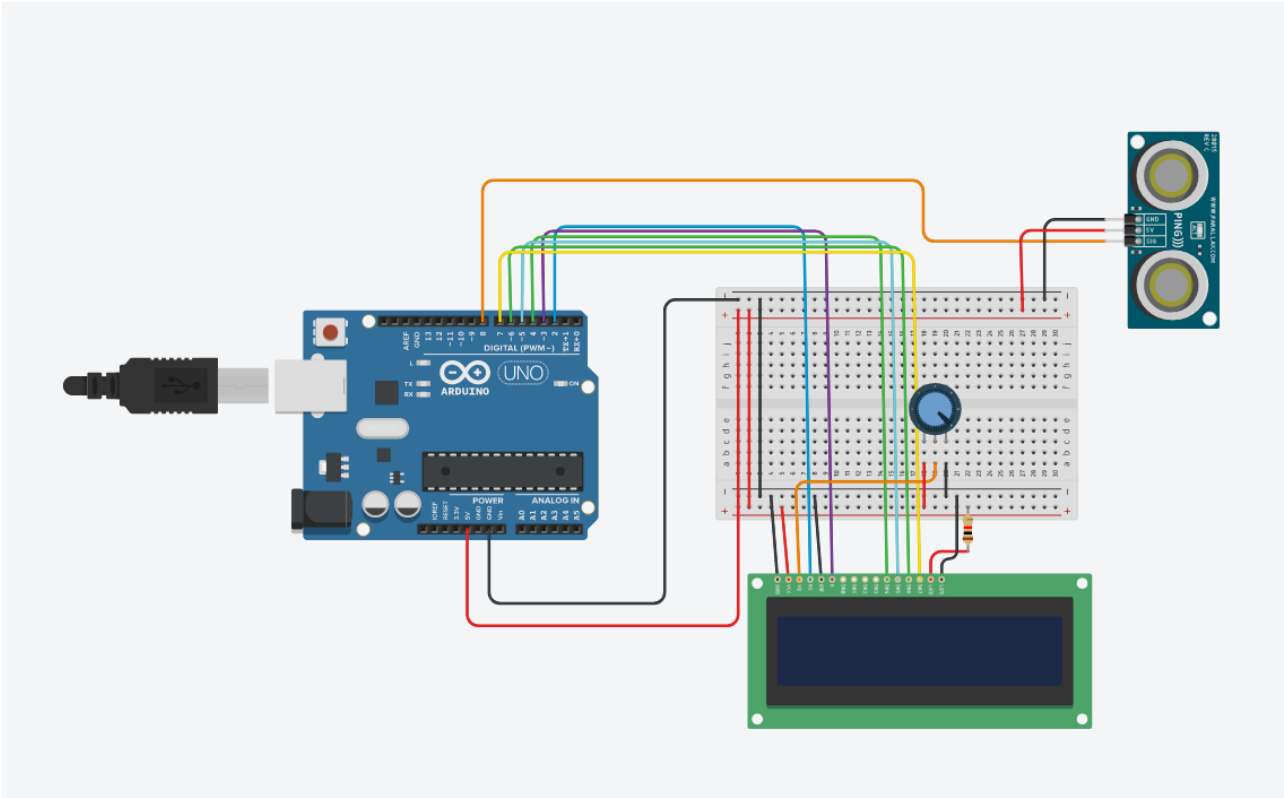
Monitor en serie

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

Numero	Condición 1	Condición2	El objeto es detectado?
1	5 cm de distancia al sensor	0 grados al eje perpendicular del sensor	El objeto si es detectado
2	50 cm de distancia al sensor	35 grados al eje perpendicular del sensor	El objeto si es detectado
3	100 cm de distancia al sensor	-35 grados al eje perpendicular del sensor	El objeto si es detectado




Numero	Condición 1	Condición2	El objeto es detectado?
4	5 cm de distancia al sensor	90 grados al eje perpendicular del sensor	El objeto no es detectado
5	50 cm de distancia al sensor	-60 grados al eje perpendicular del sensor	El objeto si es detectado
6	350 cm de distancia al sensor	0 grados al eje perpendicular del sensor	El objeto no es detectado

5. 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.



6. 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"**




Texto



1 (Arduino Uno R3)

```
1 // Incluye la libreria de la pantalla
2 #include <LiquidCrystal.h>
3
4 // Esta constante no cambia. Es el pin que corresponde a la salida
5 const int pingPin = 8;
6
7 //Coloca los parametros: (rs, enable, d4, d5, d6, d7)
8 const int rs = 2, en = 3, d4 = 4, d5 = 5, d6 = 6, d7 = 7;
9 LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
10
11 void setup() {
12     // Inicializa la comunicacion serial a 9600 bits por segundo:
13     Serial.begin(9600);
14     //Tamaño del display
15     lcd.begin(16,2);
16 }
17
18 void loop() {
19
20     // Establece variables para la duracion de ping, y la distancia
21     long duration, cm;
22
23     // El PING es activado por un pulso HIGH de 2 o mas microsegundos
24     // Da un pulso LOW corto antes de dar el HIGH para dar el pulso
25     pinMode(pingPin, OUTPUT);
26     digitalWrite(pingPin, LOW);
27     delayMicroseconds(2);
28     digitalWrite(pingPin, HIGH);
29     delayMicroseconds(5);
30     digitalWrite(pingPin, LOW);
31 }
```

Texto



1 (Arduino Uno R3)

```
34 // cuando toque un objeto
35 pinMode(pingPin, INPUT);
36 duration = pulseIn(pingPin, HIGH);
37
38
39
40 // La velocidad del sonido es de 340 m/s o 29 microsegundos por
41 // El ping viaja ida y vuelta, entonces para encontrar la dista
42 // objeto tomamos la mitad de la distancia viajada.
43 cm = duration / 29 / 2;;
44
45 Serial.print(cm);
46 Serial.print("cm");
47 Serial.println();
48
49 delay(100);
50
51
52 //Fuera de rango
53 if(cm == 336) {
54
55     lcd.setCursor(0, 0);
56     lcd.print("No Identificado");
57     lcd.setCursor(0, 1);
58     lcd.print("                ");
59 }
60 else{
61
62     // Seccion del LCD
63     lcd.setCursor(0, 0);
64     lcd.print("Identificado a: ");
65     lcd.setCursor(0, 1);
66     lcd.print(cm);
67     lcd.print("cm");
68     lcd.print(" ");
69
70 }
71
```

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

https://app.slack.com/client/T018PEE2NLD/G01AV2KBNQL

Buscar en ITN\_SistemasProgramables

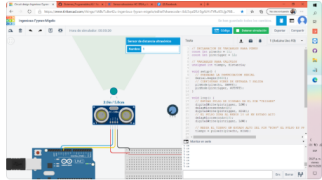
ITN\_SistemasProgram... Añadir un tema

Menciones y reacciones  
Más  
Canales  
# general  
# varios  
# zerox  
Añadir canales  
Mensajes directos  
Slackbot  
VANESSA MARLENNE RO...  
Añadir compañeros de equi...

**Jueves, 5 de noviembre**

**ALEJANDRO DIAZ NAVA** 18:14  
<https://www.tinkercad.com/things/1WBvTu8w42u-ingenious-fyran-migelo/editel?sharecode=JbIL5quGfUr5gAbvTYRuK5Ujp76BQv6njcB6Br8IXk>  
**Tinkercad**  
Tinkercad | From mind to design in minutes  
Tinkercad is a free, easy-to-use app for 3D design, electronics, and coding.

**Viernes, 6 de noviembre**

**VANESSA MARLENNE RODRIGUEZ BAEZ** 17:32  
Image.png  


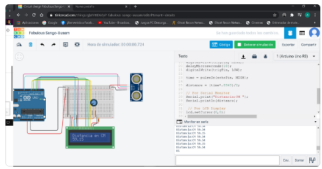
**Sábado, 7 de noviembre**

**ALEJANDRO DIAZ NAVA** 13:45  
<https://www.tinkercad.com/things/gjVWtID68yP-fabulous-sango-uusam/editel?sharecode=758UkgLXcy5EnQDo5bXiiguQn4eK-sMC8BLQYAaOLgM>  
**Tinkercad**  
Tinkercad | From mind to design in minutes  
Tinkercad is a free, easy-to-use app for 3D design, electronics, and coding.

Enviar mensaje a @zerox

**Sábado, 7 de noviembre**

**ALEJANDRO DIAZ NAVA** 13:45  
<https://www.tinkercad.com/things/gjVWtID68yP-fabulous-sango-uusam/editel?sharecode=758UkgLXcy5EnQDo5bXiiguQn4eK-sMC8BLQYAaOLgM>  
**Tinkercad**  
Tinkercad | From mind to design in minutes  
Tinkercad is a free, easy-to-use app for 3D design, electronics, and coding.

Image.png  


**VANESSA MARLENNE RODRIGUEZ BAEZ** 13:47  
Gracias por la ayuda 😊

**Ayer**

**VANESSA MARLENNE RODRIGUEZ BAEZ** 11:50  
Hola chicos buenos días  
Ya quedo la primera etapa de la practica solo falta conectarnos para mandarnos las fotos y todo eso, en la segunda etapa ya esta solo falta que identifique que no identifico el objeto  
Alejandro eso que esta trabajando es lo que me va a ayudar

Enviar mensaje a @zerox



ITN\_SistemasProgram... ▾
+ Añadir canales

Menciones y reacciones

Más

Canales

- # general
- # varios
- # zerox**

Añadir canales

Mensajes directos

Slackbot

VANESSA MARLENNE RO...

Añadir compañeros de equi...

**zerox**

Buscar en ITN\_SistemasProgramables

Añadir un tema

Ayer ▾


CodigoF1.png ▾


```





1 // DECLARACION DE VARIABLES PARA PULSO
2 const int pincho = 12;
3 const int pinterreg = 12;
4
5 // VARIABLES PARA CALCULO
6 unsigned int tiempo, distancia;
7
8 void setup() {
9   // INICIALIZO LA COMUNICACION SERIAL
10  Serial.begin(9600);
11  // CONFIGURO PINES DE ENTRADA Y SALIDA
12  pinMode(pincho, INPUT);
13  pinMode(pinterreg, OUTPUT);
14
15 }
16
17 void loop() {
18   // ENVIA PULSO DE CORRIENTE EN EL PIN "PINTERREG"
19   digitalWrite(pinterreg, LOW);
20   delayMicroseconds(1);
21   digitalWrite(pinterreg, HIGH);
22   // LE DADO OTRO AL BOMBO LO DE SE ESTADO ALTO
23   delayMicroseconds(1);
24   digitalWrite(pinterreg, LOW);
25
26   // MIDE EL TIEMPO DE BOMBEO ALTO DEL PIN "BOMBO" EL PULSO ES PERO
27   tiempo = pulseIn(pincho, HIGH);
28
29   // LA VELOCIDAD DEL SONIDO ES DE 340 M/S O 29 MICROSEGUNDOS POR CM
30   // DIVIDIMO EL TIEMPO DEL PULSO ENTRE 50, TIEMPO QUE TAMAÑA HACER
31   distancia = tiempo / 50;
32
33   // ENVIA EL RESULTADO AL MONITOR SERIAL
34   Serial.println(distancia);
35   Serial.println("\n");
        
```

Enviar mensaje a @zerox

🔗 | B I ↺ ↻ 🔍 ⌨️ 📄
Aa @ ☹️ 👤 ➡️

ITN\_SistemasProgram... 

Buscar en ITN\_SistemasProgramables 


zerox  Añadir un tema  4  


Menciones y reacciones  
: Más



Canales  
# general  
# varios  
zerox


+ Añadir canales

Mensajes directos  
Slackbot  
VANESSA MARLENNE RO...  
+ Añadir compañeros de equi...

Ayer 

Hoy 

 VANESSA MARLENNE RODRIGUEZ BAEZ 13:43  
Codigoetapa2.png 



Texto  1 (Arduino Uno R3)


```


1 // Incluye la libreria de la pantalla
2 #include <LiquidCrystal.h>
3 // Esta constante no cambia. Es el pin que corresponde a la salida
4 const int pinLed = 8;
5 //Coloca los parametros: (rs, enable, d4, d5, d6, d7)
6 const int rs = 2, en = 3, d4 = 4, d5 = 5, d6 = 6, d7 = 7;
7 LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
8
9 void setup() {
10 // Inicializa la comunicacion serial a 9600 bits por segundo
11 Serial.begin(9600);
12 //Punto de inicio del display
13 lcd.begin(16,2);
14 }
15
16 void loop() {
17 // Establece variables para la duracion de ping, y la distancia
18 long duration, cm;
19
20 // El PINO es activado por un pulso HIGH de 2 o mas microsegundos
21 // De un pulso LOW entre antes de dar el HIGH para dar el pulso
22 pinMode(pinLed, OUTPUT);
23 digitalWrite(pinLed, LOW);
24 delayMicroseconds(2);
25 digitalWrite(pinLed, HIGH);
26 delayMicroseconds(3);
27 digitalWrite(pinLed, LOW);
28 }





```

Enviar mensaje a @zerox

ITN\_SistemasProgram... 

Buscar en ITN\_SistemasProgramables 


zerox  Añadir un tema  4  



Menciones y reacciones  
: Más


Canales  
# general  
# varios  
zerox

+ Añadir canales

Mensajes directos  
Slackbot  
VANESSA MARLENNE RO...  
+ Añadir compañeros de equi...

Hoy 


 VANESSA MARLENNE RODRIGUEZ BAEZ 13:43  
CodigoEtapa2-1.png 


Texto  1 (Arduino Uno R3)

```

1 // Incluye la libreria de la pantalla
2 #include <LiquidCrystal.h>
3 // Esta constante no cambia. Es el pin que corresponde a la salida
4 const int pinLed = 8;
5 //Coloca los parametros: (rs, enable, d4, d5, d6, d7)
6 const int rs = 2, en = 3, d4 = 4, d5 = 5, d6 = 6, d7 = 7;
7 LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
8
9 void setup() {
10 // Inicializa la comunicacion serial a 9600 bits por segundo
11 Serial.begin(9600);
12 //Punto de inicio del display
13 lcd.begin(16,2);
14 }
15
16 void loop() {
17 // Establece variables para la duracion de ping, y la distancia
18 long duration, cm;
19
20 // El PINO es activado por un pulso HIGH de 2 o mas microsegundos
21 // De un pulso LOW entre antes de dar el HIGH para dar el pulso
22 pinMode(pinLed, OUTPUT);
23 digitalWrite(pinLed, LOW);
24 delayMicroseconds(2);
25 digitalWrite(pinLed, HIGH);
26 delayMicroseconds(3);
27 digitalWrite(pinLed, LOW);
28 }

```

CodigoEtapa2-1.png 

Texto  1 (Arduino Uno R3)



```

1 // cuando se enciende el sensor
2 pinMode(pinLed, OUTPUT);
3 duration = pulseIn(pinLed, HIGH);
4
5 // La velocidad del sonido es de 340 m/s o 29 microsegundos por
6 // el pin vuela la y vuelve, entonces para encontrar la dista
7 // en = duration / 29 / 2;
8
9 Serial.print(cm);
10 Serial.print("\n");
11 Serial.println();
12 delay(100);
13
14 //Punto de inicio del display
15 lcd.begin(16,2);
16 lcd.setCursor(0,0);
17 lcd.print("M: Identificado");
18 lcd.setCursor(0,1);
19 lcd.print(" ");
20 }
21
22 // Seccion del loop
23 lcd.setCursor(0,0);
24 lcd.print("Identificado a: ");
25 lcd.setCursor(0,1);
26 lcd.print(cm);
27 lcd.print("\n");
28 lcd.print(" ");
29 }

```

Fotos del código de la fase 2

Enviar mensaje a @zerox

## 8. Conclusiones

- **Diaz Navarro Alejandro:** Está práctica si me fue algo más complicada, puesto que nunca había utilizado Arduino y mucho menos programado en el, así que con ayuda de mis compañeros y unos videos pude ir entendiendo un poco más cómo se maneja esté, la primera etapa fue un tanto confusa puesto que no sabíamos en qué posición tomar las medidas pero al final lo resolvimos, en la segunda etapa tuvimos problemas en cuanto a la pantalla LCD en cuestión de la conexión y los mensajes que mandaría una vez se detectará el objeto, al final se resolvió esto.. Así que solo me queda aprender más sobre el funcionamiento y la programación de estos componentes.
- **Rodríguez Báez Vanessa Marlenne:** El uso del sensor ultrasónico normalmente es utilizado en robótica en la creación de carros de esqui obstáculos y otros proyectos, Ya había utilizado el sensor el cual no se me complicó mucho el codificarlo en la primera etapa lo único que se me complicó fue a la hora de querer hacer la medición pero mi compañero guillermo nos ayudó con eso, el que nunca había utilizado es el display el cual si tuve que investigar más de él y su funcionamiento con videos para

poder programarse y crear la condición para que el display hiciera su función, de ahí en fuera esta practica reforzó mis conocimientos con el sensor y creo nuevos sobre el display.

- **Soria Márquez Guillermo:** Esta práctica estuvo muy interesante, yo ya había utilizado este sensor anteriormente pero no de la manera que lo plantea la práctica. Al principio cuando comenzamos con la etapa 1 no tuvimos problema alguno, colocamos el objeto variando el ángulo a nuestra perspectiva , no utilizamos ninguna herramienta para posicionar el ángulo. En la etapa 2 tuvimos problemas porque en el LCD siempre mandaba el mensaje indicando que el objeto siempre estaba identificado incluso cuando lo posicionábamos fuera de rango, nos dimos cuenta que teníamos mal los cálculos y los modificamos. Al final la simulación funciono correctamente



## 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



[Link Díaz Navarro Alejandro](#)



[Link Rodríguez Báez Vanessa Marlenne](#)



[Link Soria Márquez Guillermo](#)