

NOMBRE DE LA PRÁCTICA	SENSOR DE DISTANCIA			No.	UNIDAD 4
ASIGNATUR A:	LENGUAJE INTERFAZ	CARR ERA:	ISIC	PLAN:	ISIC-2010-204

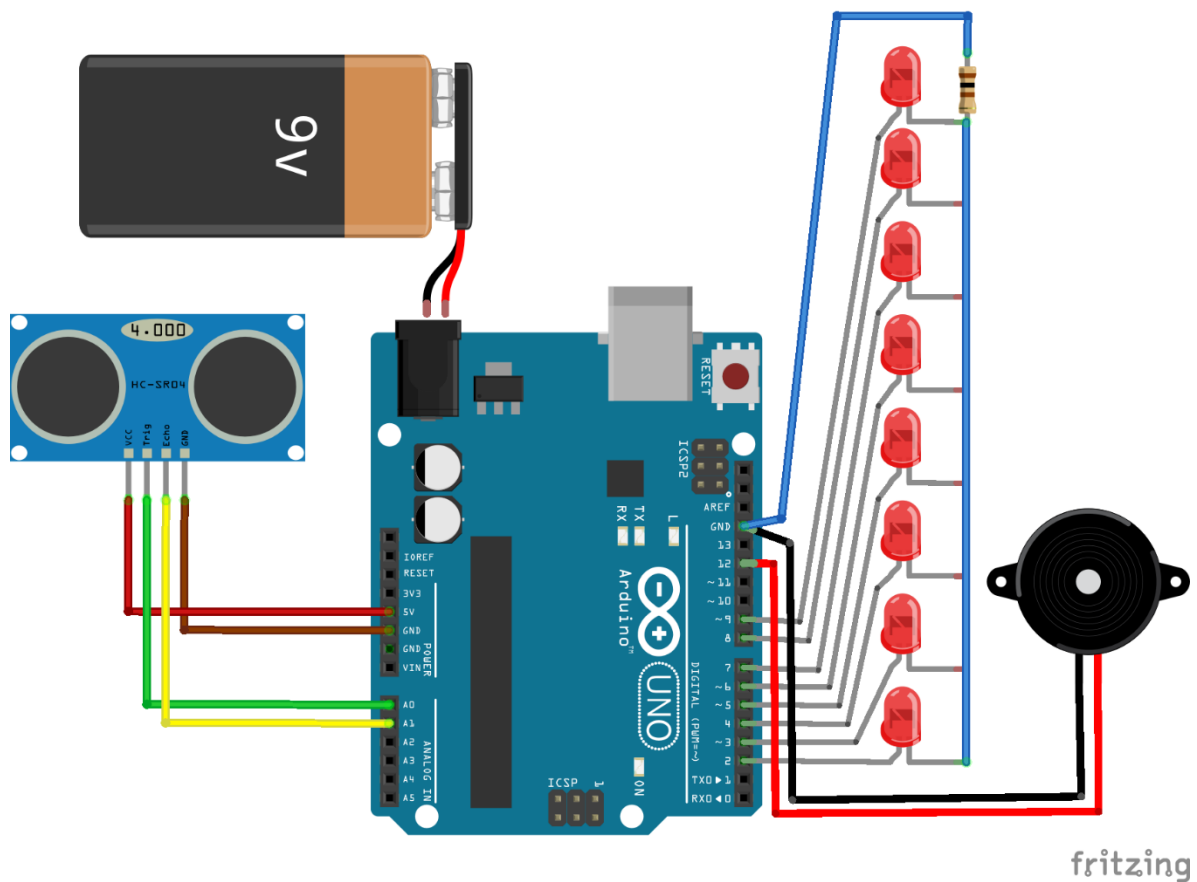
**Nombre:** Jesús Navarrete Martínez

**Grupo:** 3501

**Objetivo:** Realizar un programa que conecte un sensor de ultra sonido a través de la interface Arduino para medir el nivel del agua.

1. Realiza un programa en Arduino para sensor distancia con la tarjeta Arduino.

### Circuito a realizar:



## Código desarrollado:

```
sketch_mar05a Arduino 1.8.19 (Windows Store 1.8.57.0)
Archivo Editar Programa Herramientas Ayuda

sketch_mar05a $
/*
 * medir nivel de agua con ultrasonido
 */
const int echo=A1;
const int trigger=A0;
long tiempo;
float distancia;
int leds[]={1,2,3,4,5,6,7,8,9,10};

void setup()
{
  for(int pin=2;pin<=11;pin++)
    pinMode(pin,OUTPUT);

  pinMode(echo,INPUT);
  pinMode(trigger,OUTPUT);
  Serial.begin(9600);
  delay(100);
}

void loop()
{
  medir();

  Serial.print("Distancia: ");
  Serial.print(distancia); //Enviamos serialmente el valor de la distancia
  Serial.print("cm");
  Serial.println();

  indicador_nivel();
}

sketch_mar05a Arduino 1.8.19 (Windows Store 1.8.57.0)
Archivo Editar Programa Herramientas Ayuda

sketch_mar05a $

void medir()
{
  digitalWrite(trigger,LOW);
  delayMicroseconds(2);
  digitalWrite(trigger,HIGH);
  delayMicroseconds(10);
  digitalWrite(trigger,LOW);

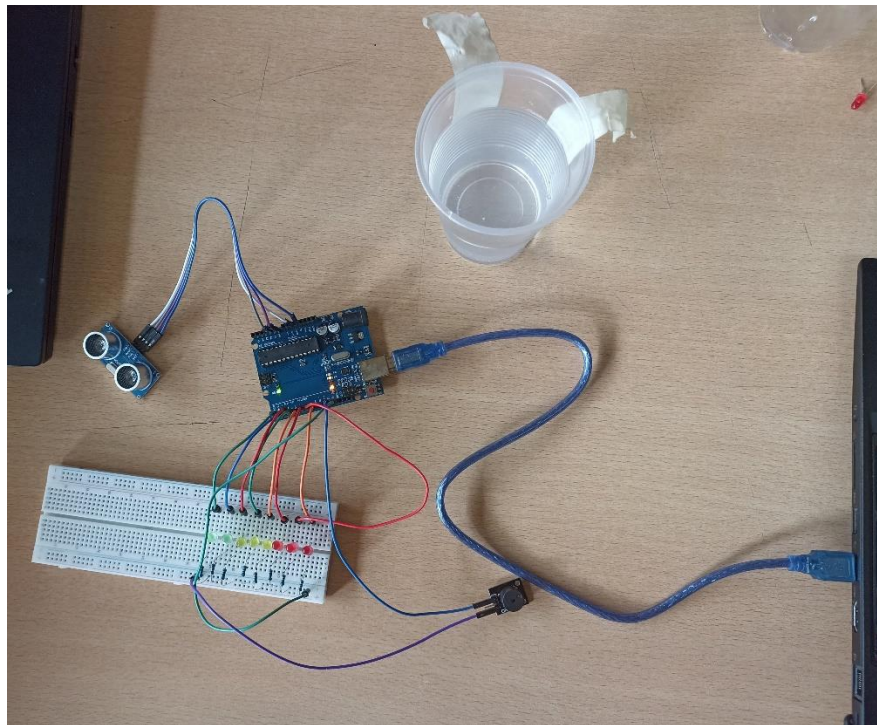
  tiempo=pulseIn(echo,HIGH);//us=microsegundos
  distancia = float(tiempo*0.0343)/2;
  delay(10);
}

void indicador_nivel()
{
  if(distancia>11) //distancia maxima , apaga los LEDs
  {for(int k=0;k<=9;k++)
    {digitalWrite(leds[k],LOW);}
  }
  else if(distancia>10) // primer nivel encender LED 1.
  {for(int k=0;k<=0;k++)
    {digitalWrite(leds[k],HIGH);}
    for(int k=1;k<=9;k++)
    {digitalWrite(leds[k],LOW);}
  }
  else if(distancia>9) // 2do nivel encender LED 2 y asi con los demas LED etc...
  {for(int k=0;k<=1;k++)
    {digitalWrite(leds[k],HIGH);}
    for(int k=2;k<=9;k++)
    {digitalWrite(leds[k],LOW);}
  }
}
```

```
sketch_mar05a Arduino 1.8.19 (Windows Store 1.8.57.0)
Archivo Editar Programa Herramientas Ayuda

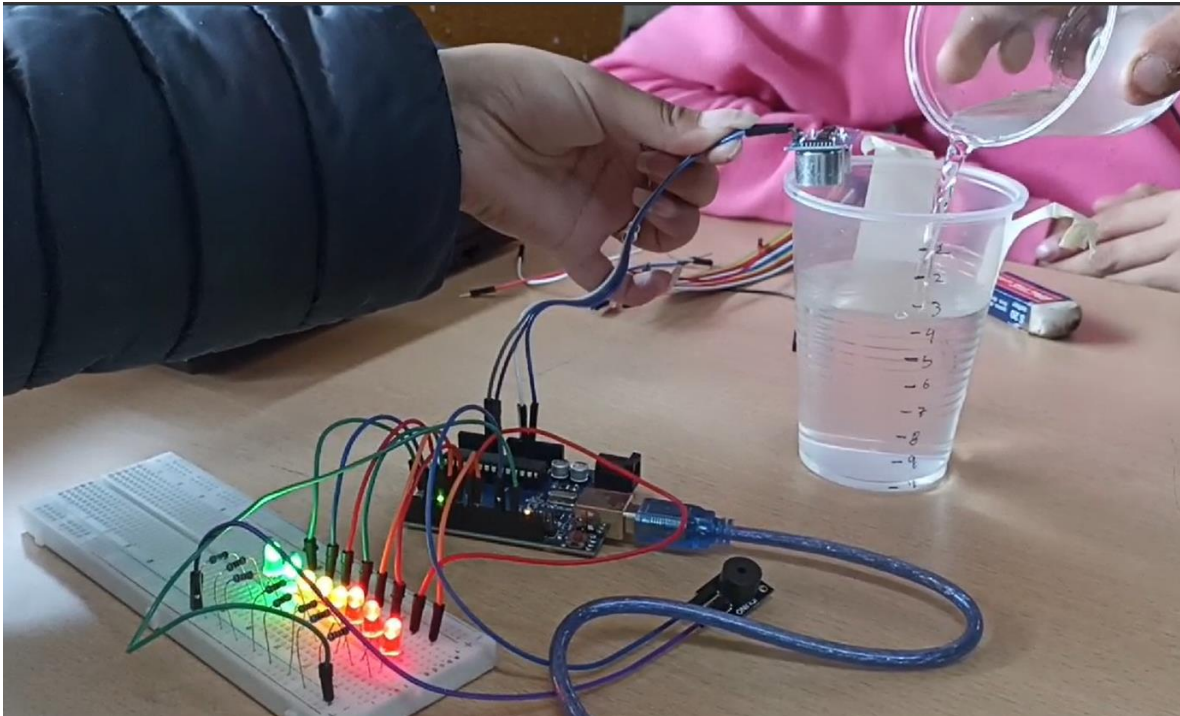
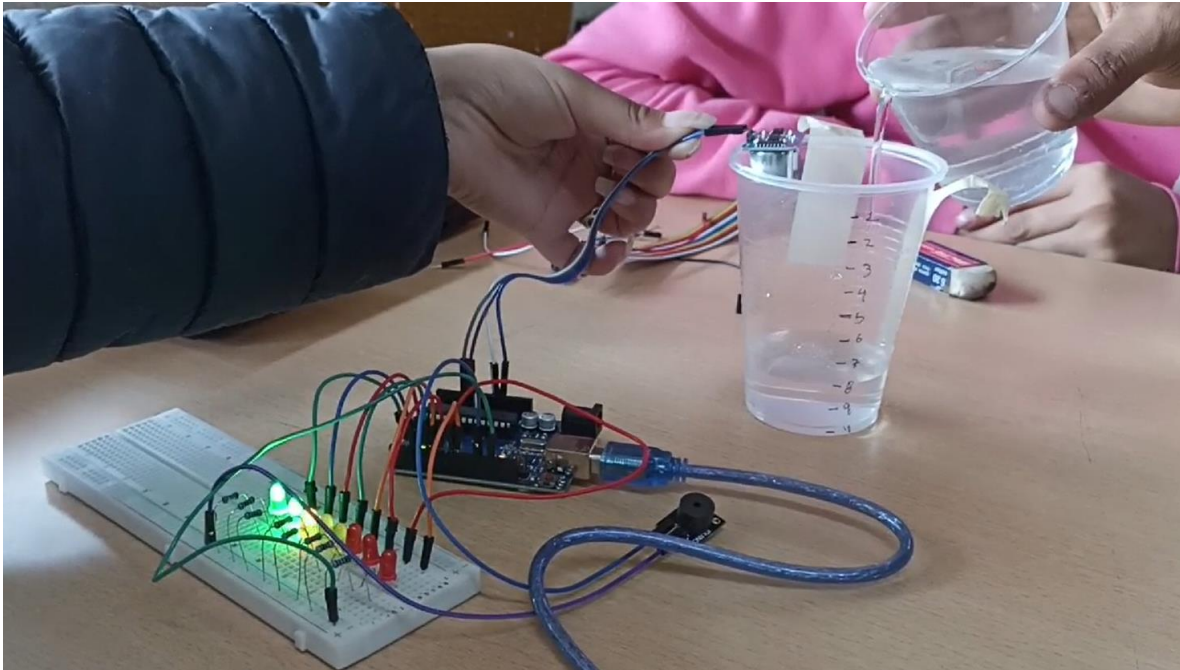
sketch_mar05a §
else if (distancia>8)
{
  for (int k=0;k<=2;k++)
  {digitalWrite (leds[k],HIGH);}
  for (int k=3;k<=9;k++)
  {digitalWrite (leds[k],LOW);}
}
else if (distancia>7)
{
  for (int k=0;k<=3;k++)
  {digitalWrite (leds[k],HIGH);}
  for (int k=4;k<=9;k++)
  {digitalWrite (leds[k],LOW);}
}
else if (distancia>6)
{
  for (int k=0;k<=4;k++)
  {digitalWrite (leds[k],HIGH);}
  for (int k=5;k<=9;k++)
  {digitalWrite (leds[k],LOW);}
}
else if (distancia>5)
{
  for (int k=0;k<=5;k++)
  {digitalWrite (leds[k],HIGH);}
  for (int k=6;k<=9;k++)
  {digitalWrite (leds[k],LOW);}
}
else if (distancia>4)
{
  for (int k=0;k<=6;k++)
  {digitalWrite (leds[k],HIGH);}
  for (int k=7;k<=9;k++)
  {digitalWrite (leds[k],LOW);}
}
else if (distancia>3) //ultimo nivel indicador LED 8.
{
  for (int k=0;k<=8;k++)
  {digitalWrite (leds[k],HIGH);}
  tone(12,523, 100);
  tone(12,450, 100);
}
```

## Demostración de su funcionamiento:



## PRÁCTICA 11

Ing. y Esp. Rodolfo Guadalupe Alcántara  
Rosales



### Conclusiones:

La realización del programa para conectar un sensor de ultrasonido a través de la interfaz Arduino y medir el nivel del agua permitió integrar conceptos clave de electrónica, programación y procesamiento de datos en un contexto práctico. Esta actividad proporcionó una comprensión profunda del funcionamiento de sensores de ultrasonido, su capacidad para medir distancias mediante el uso de ondas acústicas y su utilidad en aplicaciones de monitoreo en tiempo real.

Durante el desarrollo de la práctica, se abordaron varios aspectos técnicos importantes, como la correcta configuración de la comunicación entre el sensor y el Arduino mediante pines digitales, la calibración del sistema para garantizar mediciones precisas, y la interpretación de las señales de retorno del sensor. Esto incluyó traducir los datos en tiempo de eco a valores de distancia y relacionarlos con los niveles de agua en un recipiente, destacando la importancia de las conversiones matemáticas y las fórmulas físicas en proyectos de este tipo.

Además, el proyecto permitió reforzar habilidades de programación en el entorno de Arduino, incluyendo la estructuración del código, el manejo de bibliotecas específicas para sensores de ultrasonido y la implementación de algoritmos que aseguren la fiabilidad del sistema. La práctica también hizo énfasis en la importancia del diseño de un circuito eléctrico seguro y funcional, considerando aspectos como la alimentación del sensor y la integración de componentes adicionales, como LEDs o alarmas, para mejorar la funcionalidad.

En un contexto más amplio, este ejercicio demostró la relevancia de la tecnología de sensores en aplicaciones industriales y domésticas, como el monitoreo de tanques de agua, sistemas de riego automatizados y dispositivos de seguridad. Además, resaltó la importancia de validar los resultados mediante pruebas prácticas, ajustando parámetros como el umbral de detección y las frecuencias de medición para garantizar precisión y adaptabilidad a diferentes escenarios.