

# **Tecnologías de Automatización y Robotización para Empresas**

## **IoT**

### **Placa Arduino Uno**

## **Práctica de Laboratorio 04**

### **Percepción**

### **Parte I**

**Profesor Eladio Dapena Gonzalez**

**Curso 2024-2025**

## Práctica 04: Percepción Parte I.

### 1 Introducción

Durante el desarrollo de la práctica se experimentará con dispositivos que permiten obtener información del entorno.

En particular se presentan tres ejercicios con sensores de: temperatura, de luminosidad y medición de distancias.

### 2 Objetivos

- a) Leer el valor de mediciones realizadas desde diferentes sensores.
- b) Conocer el funcionamiento y conectar correctamente diferentes sensores.

### 3 Materiales Requeridos

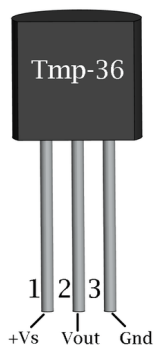
- 1 Arduino UNO
- 1 protoboard
- 1 Sensor de temperatura TMP36.
- 1 Photosensitive sensor. HW5P-1
- 1 Sensor de distancia ultrasonidos HC-SR04.
- Resistencias de  $220\Omega$
- 1 Resistencia de  $50K\Omega$
- 1 potenciómetro de  $10K\Omega$
- 1 Pantalla LCD 16x2
- Cables para conexiones.

### 4 Marco teórico

#### 4.1 Sensor TMP36

El sensor de temperatura TMP36 es un dispositivo analógico que convierte la temperatura ambiente en una señal de voltaje.

Es conocido por su precisión, estabilidad y facilidad de uso, siendo ampliamente utilizado en proyectos de monitoreo y control de temperatura.



**Características.**

Sensor TPM36.	
<b>Rango de temperatura</b>	-40°C a 125°C
<b>Sensibilidad</b>	Aproximadamente 10 mV/°C. La salida aumenta en 10 mV por cada grado Celsius de aumento de temperatura.
<b>Precisión</b>	De ±1°C en el rango de -40°C a 125°C
<b>Calibración de fábrica</b>	No requiere calibración adicional ya que está calibrado en fábrica
<b>Tensión de alimentación</b>	El rango de voltaje de operación típico es de 2.7V a 5.5V
<b>Corriente de alimentación</b>	50 µA a 2.7V y 60 µA a 5.0V
<b>Salida analógica</b>	Salida de voltaje linealmente proporcional a la temperatura medida. Valores de salida de referencia del sensor: 0 °C → 500 mV. 25 °C → 750 mV.

### Funcionamiento.

El TMP36 opera suministrando un voltaje de referencia interno que se convierte en un voltaje de salida que es directamente proporcional a la temperatura en grados Celsius.

A diferencia de otros sensores como los termistores, que requieren cálculos más complejos para convertir la resistencia en temperatura, el TMP36 proporciona una salida directa que puede ser leída fácilmente por un microcontrolador.

### Fórmula de Conversión

Para convertir el voltaje de salida del TMP36 a grados Celsius de una lectura de temperatura realizada desde un pin analógico de la placa Arduino, se utiliza la siguiente fórmula:

$$V_{out}(mV) = \frac{Lectura (Pin Analógico A_i) * 500 mV}{1024}$$

$$Temperatura ( ^\circ C) = \frac{(V_{out} - 500)}{10}$$

## 4.2 Photosensitive sensor. HW5P-1

Un fotodiodo es un dispositivo semiconductor que convierte la luz en corriente eléctrica.



### Modos de funcionamiento.

Modo Fotoconductorivo: El fotodiodo se polariza inversamente (aplicando un voltaje negativo en el ánodo en comparación con el cátodo).

Cuando la luz incide sobre el fotodiodo, los fotones de la luz excitan los electrones del material semiconductor, generando pares de electrones-huecos.

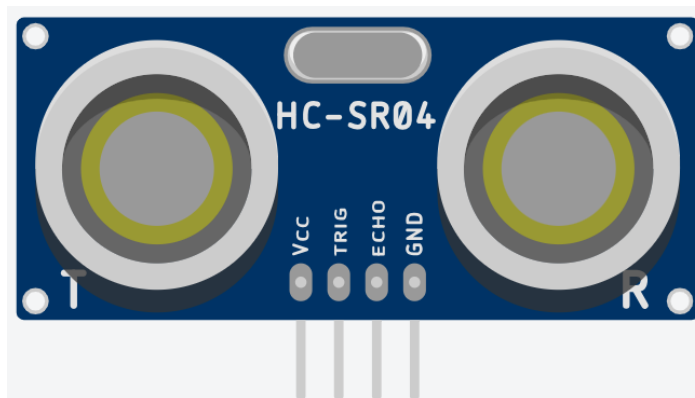
Esto aumenta la corriente inversa a través del fotodiodo, que es directamente proporcional a la intensidad de la luz incidente.

Modo Fotovoltaico: Similar al funcionamiento de una célula solar, donde el fotodiodo genera un voltaje o corriente cuando se expone a la luz, sin necesidad de polarización externa.

En aplicaciones prácticas, el fotodiodo generalmente se utiliza en modo fotoconductorivo para detectar la presencia o intensidad de la luz.

### 4.3 Sensor HC-SR04

El HC-SR04 es un sensor de medición de distancias mediante la emisión y recepción de ondas ultrasónicas.



#### Componentes y Características

El HC-SR04 consta de los siguientes componentes principales:

##### Transductor emisor (Trigger).

Este componente emite una onda ultrasónica a 40 kHz.

Cuando se aplica un pulso al pin "Trigger" del sensor, este genera la onda sonora que se propaga en línea recta en forma de un cono.

##### Transductor receptor (Echo).

Recibe la onda ultrasónica que se refleja en el objeto y vuelve al sensor.

El tiempo que tarda en volver la onda es proporcional a la distancia entre el sensor y el objeto.

##### Circuito de control.

Se encarga de procesar el tiempo transcurrido entre la emisión y la recepción de la onda sonora, generando un pulso en el pin "Echo" cuyo ancho es proporcional a la distancia medida.

##### Pines de conexión.

- VCC : Alimentación del sensor, típicamente 5V.
- GND : Conexión a tierra.
- Trigger : Pin de entrada donde se envía un pulso de activación.
- Echo : Pin de salida donde se recibe el pulso de salida, que es proporcional al tiempo de viaje de la onda ultrasónica.

#### Funcionamiento

**Emisión del pulso:** Para iniciar la medición, se envía un pulso alto (de al menos 10 microsegundos) al pin Trigger. Esto hace que el sensor emita una ráfaga de 8 pulsos de ultrasonido a 40 kHz.

**Reflexión de la onda:** La onda ultrasónica viaja a través del aire, y si encuentra un objeto en su camino, se refleja y regresa hacia el sensor.

**Recepción del pulso:** El transductor receptor detecta la onda reflejada. El tiempo entre la emisión y la recepción de la onda se mide a través del pin Echo, que permanece en estado alto durante ese tiempo.

**Cálculo de la distancia:** El microcontrolador (como un Arduino) mide el tiempo que el pin Echo permanece en alto, lo que corresponde al tiempo de viaje de la onda sonora. La distancia se calcula usando la fórmula:

$$Distancia = \frac{Tiempo * Velocidad\ del\ Sonido\ en\ el\ Aire}{2}$$

Donde la velocidad del sonido en el aire es aproximadamente 343 m/s (esto puede variar con la temperatura y la humedad).

Así, si el tiempo medido es en microsegundos, la distancia en centímetros se calcula como:

$$Distancia(cm) = \frac{Tiempo(\mu seg) * 0,0343}{2}$$

**Nota.** Cuando el sensor no recibe el rebote de la onda emitida genera como salida el valor máximo de detección, esto significa que en su rango de medición no hay objetos.

#### 4.4 Funciones.

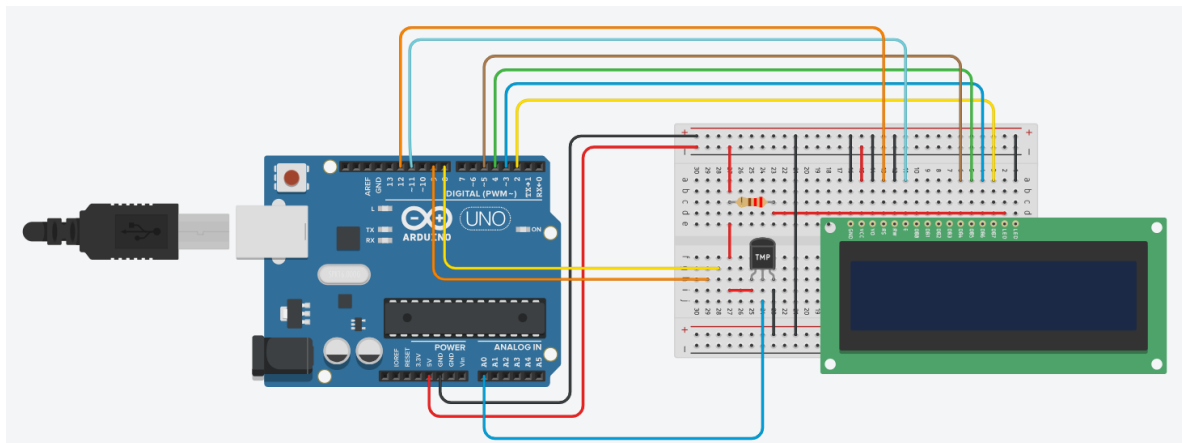
Función	Descripción
<i>micros()</i>	Retorna el número de microsegundos desde que la placa Arduino comenzó a ejecutar el programa actual. El tipo de dato asociado a la función micros es <b>unsigned long</b> . Este número se desbordará (volverá a cero) después de aproximadamente 70 minutos Ejemplo: T= <i>micros()</i>
<i>delayMicroseconds()</i>	Pausa el programa durante el tiempo (en microsegundos) especificado por el parámetro. Hay mil microsegundos en un milisegundo y un millón de microsegundos en un segundo. Esta función funciona con mucha precisión en el rango [ 3 , 16383 ] microsegundos. No se garantiza que funcione con precisión para tiempos de retardo más pequeños. Los tiempos de retardo más largos pueden demorarse durante un tiempo extremadamente breve. Para retrasos mayores a unos pocos miles de microsegundos, debería usar delay() en su lugar. Ejemplo: <i>delayMicroseconds(50);</i>

## 5 Ejercicio 1. Sensor de Temperatura TMP36 (1 punto).

El objetivo de este ejercicio es conocer, conectar y utilizar el sensor de temperatura TMP36 y mostrar su valor de salida en una pantalla LCD.

- Determinar el valor de temperatura asociado al valor de voltaje que genera el sensor.
- Conocer el funcionamiento del sensor TMP36.
- Conectar y leer correctamente el sensor TMP36.

Conectar los componentes como se muestra en el siguiente esquema:



**Nota:** **Verifique** sus conexiones con el **técnico del laboratorio** antes de continuar con la actividad.

### 5.1 Programa.

Abrir en su entorno de desarrollo (IDE Arduino) el programa [IoT04E01.ino](#) descargado desde el material de la práctica para el ejercicio 01.

### 5.2 Prueba.

Cargar el programa en la placa Arduino para su ejecución.

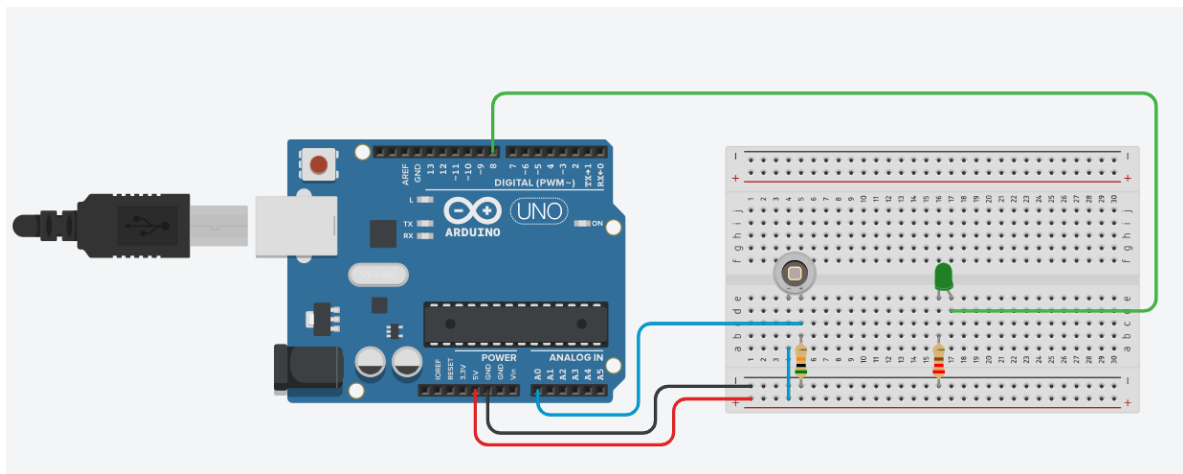
## 6 Ejercicio 2. Photosensitive sensor. (1 punto)

El objetivo de este ejercicio es utilizar un Fotosensitivo y mostrar su valor de salida en la terminal.

Cuando el valor obtenido desde el sensor superar el valor del umbral de voltaje de salida se encenderá un led.

- Lectura del valor de voltaje de salida del fotodiodo conectado a un pin analógico.
- Conocer el funcionamiento, conectar y leer correctamente un sensor Fotosensitivo.

### 6.1 Conectar los componentes como se muestra en el siguiente esquema:



**Nota:** **Verifique** sus conexiones con el **técnico del laboratorio** antes de continuar con la actividad.

### 6.2 Programa.

Abrir en su entorno de desarrollo (IDE Arduino) el programa [IoT04E02.ino](#) descargado desde el material de la práctica para el ejercicio 01.

### 6.3 Prueba.

Cargar el programa en la placa Arduino para su ejecución.

### 6.4 Validar Ejercicio.

Mostrar al técnico del laboratorio el funcionamiento del modelo para el **registro en la plantilla de valoración** de la práctica.



## 7 Ejercicio 03. Medir distancias con Sensor HC-SR04. (1 punto)

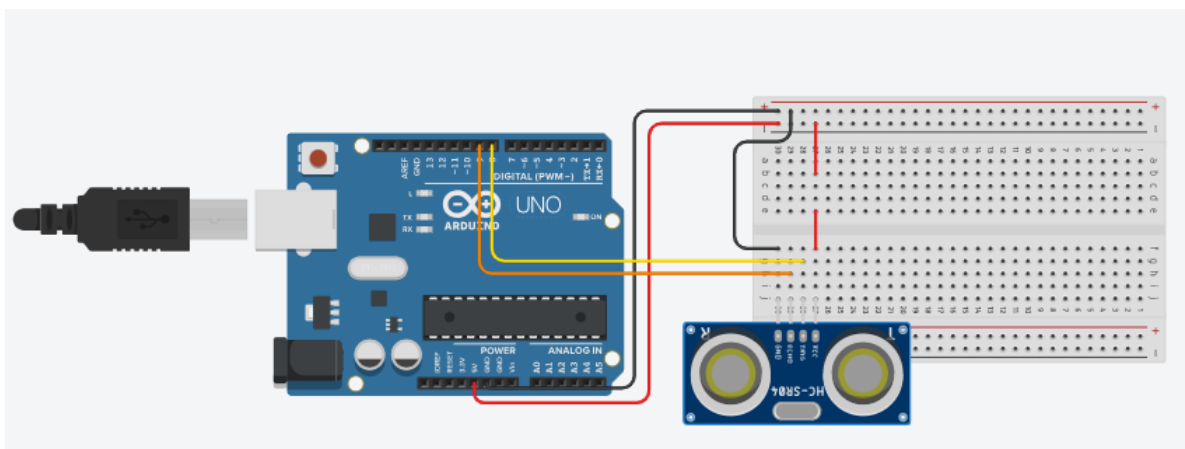
El objetivo de este ejercicio es medir distancias con el sensor de ultrasonidos HC-SR04.

Además se muestra en la terminal el valor de la distancia.

Nota. Si no hay elementos en el rango de la señal de ultrasonidos la distancia será el valor máximo del sensor.

- Determinar la distancia a la que el sensor HC-SR04 detecta un objeto.
- Conocer el funcionamiento del sensor HC-SR04.
- Conectar y activar correctamente el sensor HC-SR04.

### 7.1 Conectar los componentes como se muestra en el siguiente esquema:



**Nota:** **Verifique** sus conexiones con el **técnico del laboratorio** antes de continuar con la actividad.

### 7.2 Programa.

Abrir en su entorno de desarrollo (IDE Arduino) el programa [IoTP04E03.ino](#) descargado desde el material de la práctica para el ejercicio 01.

### 7.3 Prueba.

- Cargar el programa en la placa Arduino para su ejecución.
- Colocar su mano a diferentes distancias del sensor y observar la medición en la Pantalla LCD.
- Defina una posición fija de medición para replicar mediciones en las mismas condiciones.

### 7.4 Validar Ejercicio.

Mostrar al técnico del laboratorio el funcionamiento del modelo para el **registro en la plantilla de valoración** de la práctica.

## 8 Actividad Práctica **Autónoma A. (4 puntos)**

### 8.1 Sensor de Distancia y Señal luminosa.

**Objetivo:** Encender un led cuando se detecte un obstáculo a una distancia de **x** centímetros.

- Utilice el esquema del Ejercicio 03, para crear un nuevo esquema que incluya sus componentes y además una Pantalla LCD y un diodo led.
- Realizar las conexiones necesarias para cumplir con el objetivo.
- Crear un nuevo programa **IoTP04E04.ino** utilizando como referencia los diferentes programas de la práctica para mostrar en la pantalla LCD el valor de la distancia obtenido desde el sensor HC-SR04 e indicar el Estado del Led (Encendido, Apagado).

**Nota:** **Verifique** sus conexiones con el **técnico del laboratorio** antes de continuar con la actividad.

### 8.2 Programa.

Ejecutar el programa que ha creado **IoTP04E04.ino**.

### 8.3 Prueba.

- Cargar el programa en la placa Arduino para su ejecución.
- Definir una distancia **X** de encendido del Led luego de verificar diferentes valores obtenidos desde el sensor.
- Utilice la distancia X fija de medición para realizar repetidas ejecuciones del programa y observa la lectura del sensor manteniendo las mismas condiciones.

### 8.4 Validar Ejercicio.

Mostrar al técnico del laboratorio el funcionamiento del modelo para el **registro en la plantilla de valoración** de la práctica.

## 9 Actividad Práctica **Autónoma B. (3 puntos)**

### 9.1 Sensor medidor de distancia SHARP GP2Y0A21.

- Investigar el funcionamiento general del sensor SHARP GP2Y0A21 y su forma de conexión.
- Sustituya el sensor de ultrasonidos HC-SR04 del esquema del ejercicio anterior por un sensor SHARP GP2Y0A21.
- Crear un nuevo programa **IoT04E05.ino** a partir del programa **IoT04E04.ino** de la práctica para mostrar en la pantalla LCD el valor de la distancia obtenido desde el sensor SHARP GP2Y0A21 e indicar el Estado del Led (Encendido, Apagado).

**Nota:** **Verifique** sus conexiones con el **técnico del laboratorio** antes de continuar con la actividad.

### 9.2 Programa.

Ejecutar el programa que ha creado **IoT04E05.ino**.

### 9.3 Prueba.

- Cargar el programa en la placa Arduino para su ejecución.
- Definir una distancia **X** de encendido del Led luego de verificar diferentes valores obtenidos desde el sensor.
- Utilice la distancia **X** fija de medición para realizar repetidas ejecuciones del programa y observa la lectura del sensor manteniendo las mismas condiciones.

### 9.4 Validar Ejercicio.

Mostrar al técnico del laboratorio el funcionamiento del modelo para el **registro en la plantilla de valoración** de la práctica.