



**UNIVERSIDAD DE SANTIAGO DE CHILE**  
FACULTAD DE INGENIERÍA  
Departamento de Ingeniería Mecánica  
Ingeniería Ejecución Mecánica

---



# **Tópico de la Especialidad**

## **Tercer informe de laboratorio:** **Programación en Arduino**

---

Estudiante: Vicente Vásquez Bornhardt

Profesor: Marcelino Araya Espinoza

Fecha de entrega:  
viernes 04 de septiembre de 2020

## **Tabla de contenidos**

Resumen.....	1
Objetivo general.....	2
Objetivos específicos.....	2
Equipos utilizados.....	2
Metodología Experimental.....	3
Análisis de resultados y conclusiones.....	4
Referencias.....	5
Anexo.....	5
Código Arduino.....	5
Captura de pantalla repositorio.....	7

---

## **Resumen**

En este trabajo se utilizó un sensor de proximidad PING, en conjunto con una placa Arduino Uno y tres diodos led. Se analizó y validó el rango operacional del instrumento de medición y se definieron tres segmentos equidistantes, dentro de los cuales un objeto puede ser detectado por el dispositivo. Según la distancia medida hacia el objeto sea dentro del primer, segundo, o tercer segmento (siendo el primero el más cercano), se definió que se encienda el led verde, el amarillo, o el rojo respectivamente.

Para validar el rango operacional del dispositivo, se utilizó la comunicación serial entre la placa Arduino y el computador. De esta manera, se definió un rango de trabajo con límites muy cercanos a los del rango operacional del sensor, para asegurarse de que los led se encendieran exclusivamente dentro del rango de trabajo establecido.

En este trabajo se hizo uso de elementos básicos de la programación en Arduino. Estos fueron el definir funciones dentro del programa, el uso de variables locales y globales, las funciones Setup y Loop, además del uso de la comunicación serial para estudiar el funcionamiento del sensor de proximidad utilizado.

## **Objetivo general**

Realizar programación en la plataforma Arduino a partir de proyectos de la comunidad.

## **Objetivos específicos**

1. Reconocer elementos de programación y ser capaz de modificarlos para dar un uso personalizado.
2. Interactuar con la comunidad, obteniendo información y compartiendo el desarrollo versionado.

## **Equipos utilizados**

### **Placa Arduino Uno:**

- Microcontroller: ATmega328P
- Operating Voltage: 5V
- Input Voltage (recommended): 7 - 12V
- Input Voltage (limit): 6 - 20V
- Digital I/O Pins: 14 (of which 6 provide PWM output)
- PWM Digital I/O Pins: 6
- Analog Input Pins: 6
- DC Current per I/O Pin: 20 mA
- DC Current for 3.3V Pin: 50 mA
- Flash Memory: 32 kB (ATmega328P) of which 0.5 kB used by bootloader
- SRAM: 2 kB (ATmega328P)
- EEPROM: 1 kB (ATmega328P)
- Clock Speed: 16 MHz
- LED BUILTIN: 13
- Size: 68.6 mm × 53.4 mm

### **Sensor de proximidad ultrasónico:**

- Range: 2 cm to 336 cm (0.8 in to 3.3 yd)
- Supply voltage: +5 VDC
- Supply current: 30 mA typ; 35 mA max
- Communication: Positive TTL pulse
- Package: 3-pin SIP, 0.1" spacing (ground, power, signal)
- Operating temperature: 0 – 70° C
- Size: 22 mm H x 46 mm W x 16 mm D (0.84 in x 1.8 in x 0.6 in)
- Weight: 9 g (0.32 oz)

**Diodo LED:**

- Diámetro: 5 mm
- Corriente: 20 mA
- Voltaje: 3.2 a 3.4V
- Milicandelas: 13 000 a 15 000
- ángulo de emisión: 25°

**Resistencia eléctrica:**

- Resistencia Eléctrica: 200Ω
- Potencia de disipación: 0,25 vatios
- Tecnología de inserción (through hole)
- Fabricante: Genérico
- Disposición: Tipo Axial
- Tolerancia: 5%

**Metodología Experimental**

1. Para definir el circuito electrónico usado en esta experiencia se tomó el circuito montado en la experiencia anterior y se le añadió un led adicional. También se cambiaron las conexiones con los pines, para mantener más ordenadas las conexiones. Se definió usar los pines 2, 3 y 4 para los diodos led y el pin 7 para el sensor de proximidad.
2. Una vez definido el circuito, se utilizó el código anterior (basado en el código de muestra que viene asociado al sensor PING, disponible en la página tinkercad) para reconocer el rango de operación del sensor y estudiar su precisión. Para esto, se utilizó el comando «Serial.print», con el cual se observaron los valores reales medidos por el dispositivo.
3. Se definieron valores máximos y mínimos para utilizar, los cuales son muy cercanos a los extremos del rango de operación. Estos valores se escogieron debido a las limitaciones del emulador de tinkercad, por lo que son cercanos pero no iguales al rango de operación del sensor PING real. Los valores utilizados fueron 2.5 cm y 327.0 cm.
4. Se definieron dos variables «distMin» y «distMax», las cuales contienen los valores de la distancia mínima y máxima que se escogieron en el punto N°3. Con estas variables, se diseñaron los rangos con los cuales se trabajó dentro del código, a partir de los cuales se encienden las luces led.
5. Para aplicar los límites del rango de operación al código, se dividió el valor del rango operacional en tres segmentos. Utilizando tres funciones IF, se verificó si el valor de la distancia medida por el dispositivo se correspondía con alguno de estos segmentos. En caso de que esta condición se cumpliera, se definió que se encendiera un led verde, para el segmento más cercano, un amarillo para el intermedio y un led rojo para el segmento ubicado a más distancia del sensor.

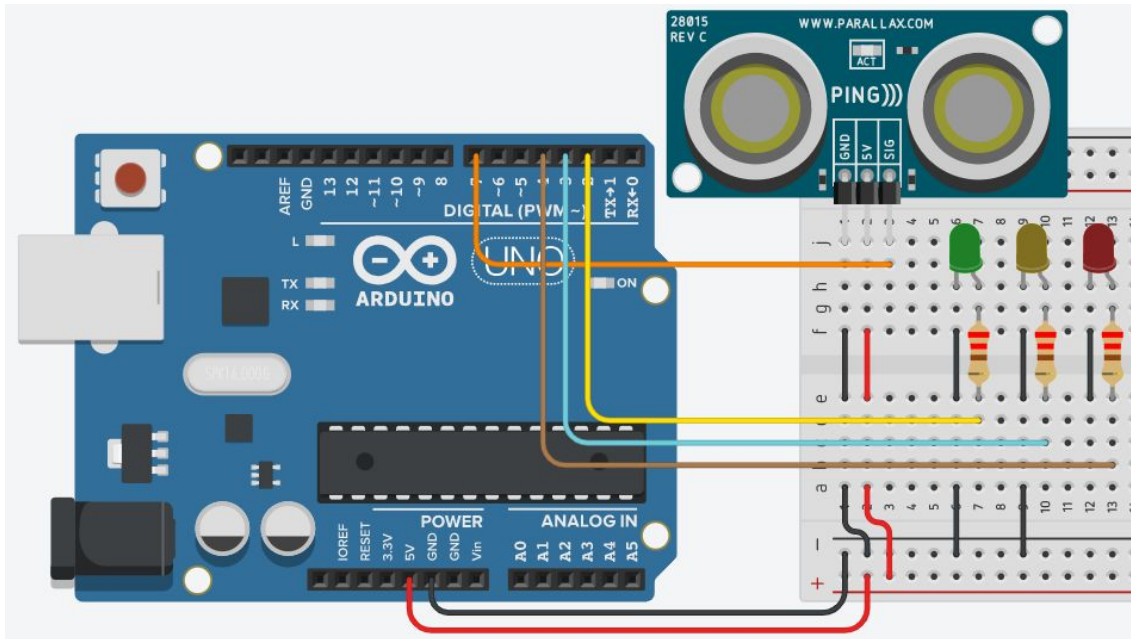


Figura 1: Esquema de montaje del sensor de proximidad PING con la placa Arduino

### **Análisis de resultados y conclusiones**

Se definió que los diodos led solo se enciendan cuando la distancia medida se ubica dentro del rango de operación del dispositivo. Cuando el objeto medido por el sensor se ubica más cerca que 2.5 cm, o más lejos que 327 cm, ninguno de los led se enciende.

Este trabajo se realizó haciendo uso de un emulador del sensor de proximidad PING, el cual se encuentra en la página «Tinkercad.com». La precisión y operación de este dispositivo es similar, pero no exactamente igual al sensor real. Por este motivo, los valores escogidos para definir el rango de trabajo son muy cercanos, pero no iguales al rango de operación definido para este dispositivo. En caso de replicar este trabajo con un sensor PING real, los valores de las variables «distMin» y «distMax» deberán ser modificados para que sean más cercanos todavía al rango de operación real.

Se consiguió que las luces led se encendieran cuando la distancia medida estaba dentro del rango. Para esto, se definieron tres segmentos que dividen al rango de operación en tres tercios y se logró que cada uno de los led se encienda según la distancia medida por el sensor esté dentro del rango de alguno de los tres segmentos. De esta manera, el led verde se enciende cuando el valor de la distancia medida se ubica en el segmento más proximo al sensor, el led amarillo se enciende cuando la distancia medida se ubica dentro del segundo segmento y el led rojo se enciende cuando la distancia se ubica dentro del segmento mas distante.

## **Referencias**

1. Ficha técnica sensor de proximidad PING:  
<https://datasheet.octopart.com/28015-Parallax-datasheet-28530511.pdf>
2. Características técnicas placa Arduino Uno:  
<https://afel.cl/producto/arduino-uno-rev3-original/>

## **Anexo**

### **Código Arduino**

/\*

Este programa usa un sensor de proximidad PING para medir la distancia entre un objeto y el sensor. Se escogen una distancia máxima y una distancia mínima de 327.0 cm y 2.5 cm respectivamente, las cuales son muy cercanas al rango de operación del instrumento.

Cuando el objeto detectado se encuentra dentro de un rango previamente definido, se enciende un led de un color específico.

- Led verde: entre 2.5 cm y 109.0 cm
- Led amarillo: entre 109.0 cm y 218.0 cm
- Led rojo: entre 218.0 cm y 327.0 cm

El programa también ejecuta comunicación serial e imprime continuamente el valor de la distancia mínima, la distancia máxima y la distancia actual a la que se encuentra un objeto detectado por el sensor.

```
// ***** //
```

Ping))) Sensor

This sketch reads a PING))) ultrasonic rangefinder and returns the distance to the closest object in range. To do this, it sends a pulse to the sensor to initiate a reading, then listens for a pulse to return. The length of the returning pulse is proportional to the distance of the object from the sensor.

The circuit:

- \* +V connection of the PING))) attached to +5V
- \* GND connection attached to ground
- \* SIG connection attached to digital pin 7

<http://www.arduino.cc/en/Tutorial/Ping>

This example code is in the public domain.

```
*/

float distancia = 0.0;
int ledVerde = 2;
int ledAmarillo = 3;
int ledRojo = 4;
float distMin = 2.5; // distancia mínima de operación
float distMax = 327.0; // distancia máxima de operación

long readUltrasonicDistance(int triggerPin, int echoPin){
  pinMode(triggerPin, OUTPUT); // Clear the trigger
  digitalWrite(triggerPin, LOW);
  delayMicroseconds(2);
  // Sets the trigger pin to HIGH state for 10 microseconds
  digitalWrite(triggerPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(triggerPin, LOW);
  pinMode(echoPin, INPUT);
  // Reads the echo pin, and returns the sound wave travel time in microseconds
  return pulseIn(echoPin, HIGH);
}

void setup(){
  pinMode(ledVerde, OUTPUT);
  pinMode(ledAmarillo, OUTPUT);
  pinMode(ledRojo, OUTPUT);
  Serial.begin(9600);
}

void loop(){
  // measure the ping time in cm
  float distancia = 0.01723 * readUltrasonicDistance(7, 7);

  if ((distancia >= distMin) && distancia <= distMax/3){
    digitalWrite(ledVerde, HIGH);
  }
  else {
    digitalWrite(ledVerde, LOW);
  }
  if ((distancia > distMax/3) && (distancia <= distMax*2/3)){
    digitalWrite(ledAmarillo, HIGH);
  }
  else {
    digitalWrite(ledAmarillo, LOW);
  }
}
```

```


if ((distancia > distMax*2/3) && (distancia <= distMax)){
    digitalWrite(ledRojo, HIGH);
}
else {
    digitalWrite(ledRojo, LOW);
}

Serial.print("distancia minima: ");
Serial.print(distMin);
Serial.print("\t distancia maxima: ");
Serial.print(distMax);
Serial.print("\t distancia: ");
Serial.println(distancia);

delay(10);
}

```

### Captura de pantalla repositorio


 master

[Go to file](#)
[Add file](#)
[Code](#)

[About](#)

This branch is 5 commits ahead of dimecusach:master.

[Pull request](#)
[Compare](#)


**VicenteHub**
Add files via upload

40 seconds ago

9

apendice	se agrega telemetroUltrasonico-pingSensor.ino	13 days ago
code	Update sensorPing-semaforoLed.ino	3 hours ago
plano	Add files via upload	2 hours ago
referencias	se agrega telemetroUltrasonico-pingSensor.ino	13 days ago
LICENSE	Initial commit	13 days ago
Lab 3 TOPICO.pdf	Add files via upload	40 seconds ago
README.md	se agrega telemetroUltrasonico-pingSensor.ino	13 days ago

[Readme](#)

[GPL-3.0 License](#)

Proyecto de uso de sensor de proximidad PING acoplado a una placa Arduino Uno. Este proyecto es parte de la experiencia N°3 de laboratorio de la asignatura de Robótica (Tópico de la especialidad), para la carrera de Ingeniería de Ejecución Mecánica en la Universidad de Santiago de Chile