AMERICAN INTERNATIONAL UNIVERSITY-BANGLADESH

Faculty of Engineering

Lab Report

Experiment # 07

Title: Interfacing the Arduino with an external sensor using serial communication protocol for implementing an obstacle detection system.

Date of Perform	12 March 2023	**Date of Submission**	19 March 2023
Course Title	MICROPROCESSOR AND EMBEDDED SYSTEMS		
Course Code	COE3104	**Section**	C
Semester	Spring 2022-23	**Degree Program**	BSc in CSE
Course Teacher	**RICHARD VICTOR BISWAS**		

Declaration and Statement of Authorship:

1. I/we hold a copy of this Assignment/Case-Study, which can be produced if the original is lost/damaged.
2. This Assignment/Case-Study is my/our original work and no part of it has been copied from any other student's work or from any other source except where due acknowledgment is made.
3. No part of this Assignment/Case-Study has been written for me/us by any other person except where such collaboration has been authorized by the concerned teacher and is clearly acknowledged in the assignment.
4. I/we have not previously submitted or currently submitting this work for any other course/unit.
5. This work may be reproduced, communicated, compared, and archived for the purpose of detecting plagiarism.
6. I/we give permission for a copy of my/our marked work to be retained by the Faculty Member for review by any internal/external examiners.
7. I/we understand that Plagiarism is the presentation of the work, idea, or creation of another person as though it is your own. It is a form of cheating and is a very serious academic offense that may lead to expulsion from the University. Plagiarized material can be drawn from, and presented in, written, graphic and visual forms, including electronic data, and oral presentations. Plagiarism occurs when the origin of the source is not appropriately cited.
8. I/we also understand that enabling plagiarism is the act of assisting or allowing another person to plagiarize or copy my/our work.

* Student(s) must complete all details except the faculty use part.
** Please submit all assignments to your course teacher or the office of the concerned teacher.

Group # 01

Sl No	Name	ID	Department
1	Zaid Amin **Rawfin**	20-42459-1	CSE
2	Abzana Sultan **Ira**	20-41973-1	CSE
3	MD. Sanjid Bin Karim **Sezan**	19-41702-3	CSE
4	Abdur Rahman **Swapnil**	18-38950-3	CSE

Faculty use only		
FACULTY COMMENTS	**Marks Obtained**	
	Total Marks	

Table of Contents

Experiment Title

Interfacing the Arduino with an external sensor using serial communication protocol for implementing an obstacle detection system.

Abstract

This experiment aims to interface an Arduino microcontroller with an external HCSR04 ultrasonic sensor using serial communication protocol to implement an obstacle detection system. The system detects the distance between the sensor and an obstacle and illuminates one or more LEDs based on the measured distance. The HCSR04 sensor emits a 40 kHz pulse and detects whether the pulse signal is reflected back by an object. The distance of the object from the sensor is calculated based on the time taken by the pulse signal to travel and reflect back. The Arduino Uno R3 board is used to program and control the system. The experiment involves writing code for the system in Arduino IDE, interfacing the sensor with the Arduino board, and testing the system to detect obstacles.

Introduction

The Arduino microcontroller is a popular and versatile platform used for creating interactive electronics projects. One of the key features of the Arduino is its ease of use and the ability to interface with a wide range of sensors and peripherals. In this experiment, we will use the Arduino to implement a simple obstacle detection system using an ultrasonic sensor (HCSR04). The sensor will detect the distance between the sensor and an obstacle, and based on the distance, the Arduino will control one or more LEDs to indicate the presence of an obstacle. This experiment provides an excellent opportunity to learn about the principles of serial communication and interfacing with external sensors, which are essential skills for many applications in the field of robotics and automation.

Theory and Methodology

Arduino is an open-source platform used for creating interactive electronics projects. Arduino consists of both a programmable microcontroller and a piece of software, or IDE (Integrated Development Environment) that runs on your computer used to write and upload computer code to the microcontroller board. Arduino Uno also doesn't need a hardware circuit (programmer/ burner) to load a code into the board. We can easily load a code into the board just using a USB cable and the Arduino IDE (which uses an easier version of C++ to write codes).

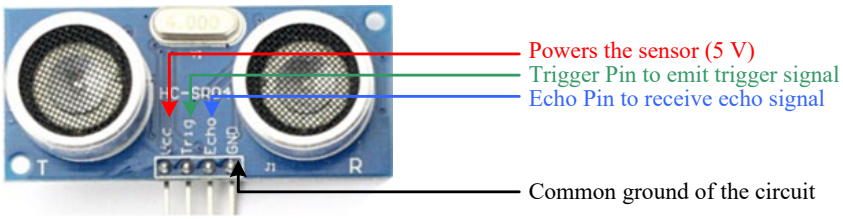
In this experiment, we will use a sonar sensor (HCS04) to detect the distance of an obstacle. Based on the distance between the sensor and the object being detected, one or more LEDs will glow as soon as it detects the obstacle.

The HCSR04 ultrasonic sensor uses a sonar signal to determine the distance to an object. This sensor reads from 2 cm to 400 cm (0.8 inches to 157 inches) with an accuracy of 0.3 cm (0.1 inches). The HCSR04 module consists of a transmitter, receiver, and control circuit. It has four pins, such as VCC, GND, Trigger, and Echo. A list of some features and specifications of this sensor is given below, but for more information, you should consult the sensor's datasheet:

- Power Supply: +5 V DC

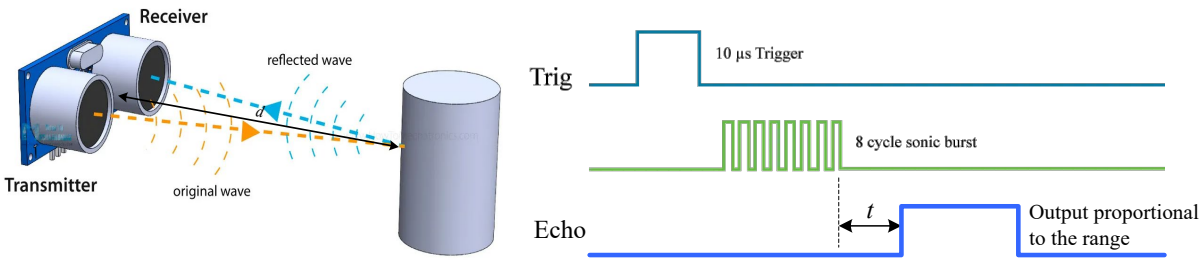
- Quiescent Current: < 2 mA
- Working Current: 15 mA
- Effective Angle: < 15°
- Ranging Distance: 2 cm – 400 cm/1″ – 13 ft
- Resolution: 0.3 cm
- Measuring Angle: 30°
- Trigger Input Pulse width: 10 μs TTL pulse
- Echo Output Signal: TTL pulse proportional to the distance range

Here is the pin configuration of the sensor:



You can easily interface it with Arduino boards. Using the output Trigger pin, the module automatically sends eight 40 kHz pulse signals and detects whether there is a pulse signal back at the Echo pin. The Trigger pin of the sensor is connected to digital pin 11 and the Echo pin to digital pin 12 of the Arduino Uno R3 board with connecting wires. An LED is connected to pin 2 to show that an obstacle is detected. Here, pins 11 and 2 will act as output pins because the trigger will be generated from Arduino, and the LED state (HIGH/LOW) will also be changed by the Arduino board. If more than one LED is to be connected then we can use more digital pins, for example, 3 and 4. All these pins must be declared as output pins in the setup function.

The ultrasound transmitter (Trigger pin) emits a high-frequency ultrasonic sound wave (40 kHz) that travels through the air. If it finds an object, it bounces back to the module. The ultrasound receiver (Echo pin) receives the reflected sound wave (echo) as shown in the following schematic diagram:



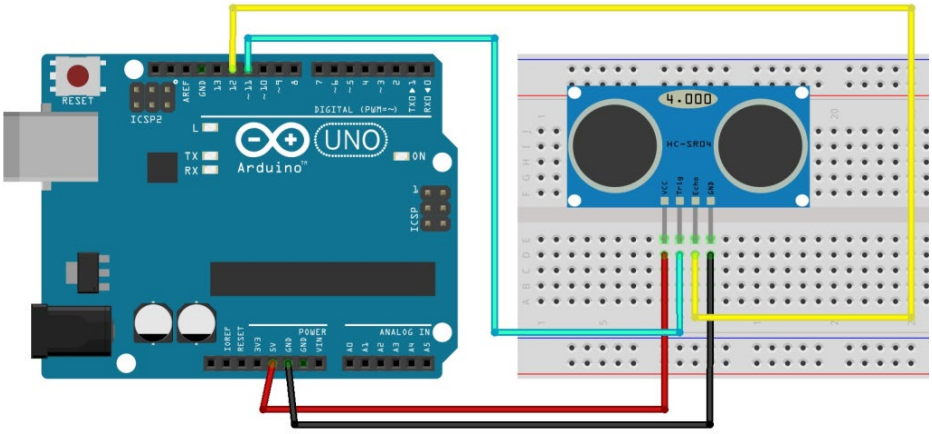
As the trigger signal generated from the Arduino board travels out from the Trigger pin and comes back to the Echo pin, the travel distance of the signal from the microcontroller to the object and back again to the microcontroller is double the distance (d) between the microcontroller and the board. We know that the sound velocity (v) in the air is 340 m/s.

$$2d = vt \text{ or simply, } d = \frac{t}{2}v, \text{ where the } t = \text{travel time.}$$

Experimental Procedure

1. Firstly, the HCSR04 ultrasonic sensor was connected to the Arduino board as per the pin configuration given above.
2. A LED was also connected to pin 2 of the Arduino board.
3. The Arduino IDE was then opened and a new sketch was created.
4. The code was written in the sketch to read data from the sensor using the serial communication protocol and display it on the Serial Monitor.
5. The code was then uploaded to the Arduino board using the USB cable.
6. The Serial Monitor was opened in the Arduino IDE to check if the sensor readings were being displayed correctly.
7. The code was then modified to control the LED based on the distance measured by the sensor.
8. The modified code was uploaded to the Arduino board, and the sensor was tested to detect the presence of obstacles.
9. The test was performed by placing an object at varying distances from the sensor and observing the LED behavior.
10. The experiment was repeated with multiple LEDs connected to the Arduino board to indicate different distance ranges.
11. The results were recorded and analyzed to evaluate the performance of the obstacle detection system..

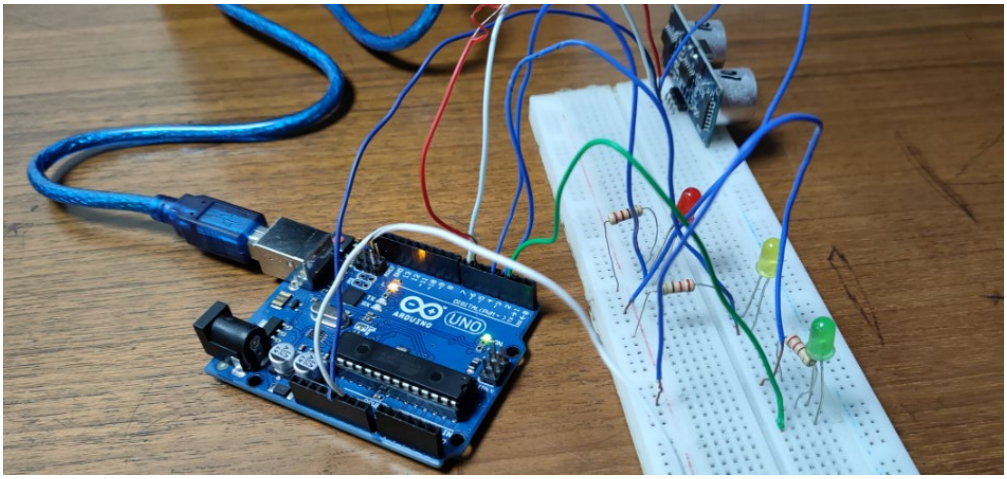
Circuit Diagram



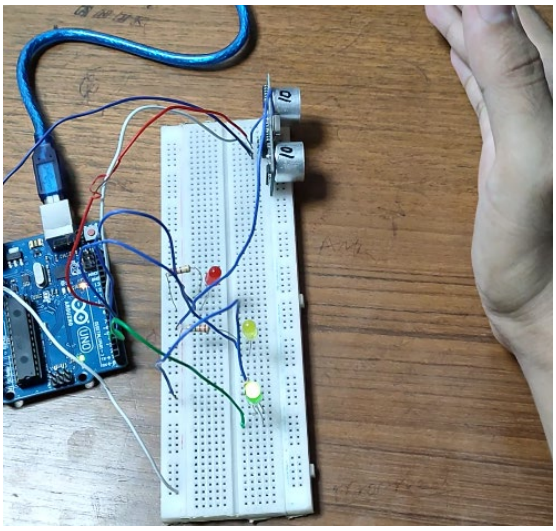
Apparatus

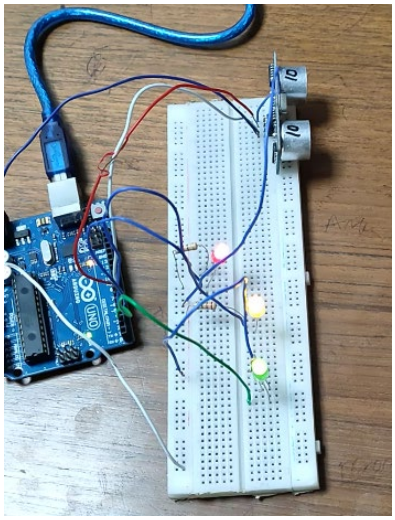
- Arduino IDE
- Arduino Uno (R3) board
- Sonar Sensor (HCSR04)
- LED

Hardware Output Results



Hardware Setup





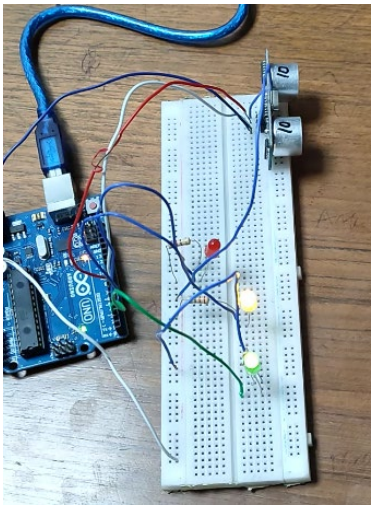
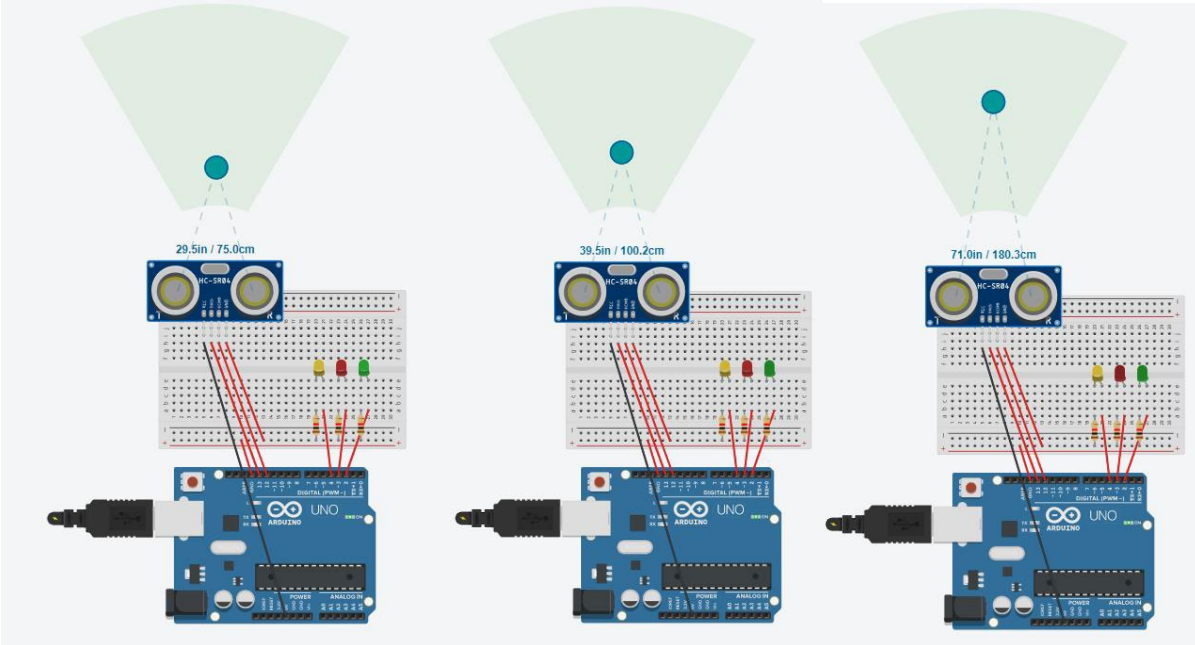


Fig: Lights changes when moving obstacle

Simulation Output Results



Code/Program

```
const int trigPin = 11;
const int echoPin = 12;
// define variables
long duration;
float distance;
float distanceInches;
float distanceThreshold;

void setup() {
  Serial.begin(9600); // Starts the serial communication
  pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output
  pinMode(echoPin, INPUT); // Sets the echoPin as an Input
  pinMode(2, OUTPUT); // Sets pins 2, 3, and 4 as the Output pin
  pinMode(3, OUTPUT);
  pinMode(4, OUTPUT);
}

void loop() {
  // Clears the trigPin
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);

  // Sets the trigPin on HIGH state for 10 microseconds
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  // Reads the echoPin, returns the sound wave travel time in microseconds
  duration = pulseIn(echoPin, HIGH);
```

```
  // Calculating the distance
  distance = (duration/2) * 1e-6 * 340 * 100;
  distanceInches = (distance / 2.54);

  Serial.print("Distance = ");
  Serial.print(distance);
  Serial.print("cm; ");
  Serial.print("Distance = ");
  Serial.print(distanceInches);
  Serial.println("inches");

  // set threshold distance to activate LEDs
  distanceThreshold = 80;

  if (distance > distanceThreshold) {
    digitalWrite(2, LOW);
    digitalWrite(3, LOW);
    digitalWrite(4, LOW);
  }

  if (distance < distanceThreshold && distance > distanceThreshold-30) {
    digitalWrite(2, HIGH);
    digitalWrite(3, LOW);
    digitalWrite(4, LOW);
  }

  if (distance < distanceThreshold-30 && distance > distanceThreshold-50) {
    digitalWrite(2, HIGH);
    digitalWrite(3, HIGH);
    digitalWrite(4, LOW);
  }

  if (distance < distanceThreshold-50 && distance > distanceThreshold-70 ) {
    digitalWrite(2, HIGH);
    digitalWrite(3, HIGH);
    digitalWrite(4, HIGH);
  }

  delay(200);
}
```

Code Explanation

This code sets up and uses an ultrasonic sensor to measure distance and activate LEDs based on the measured distance. The sensor is connected to pins 11 and 12 of the Arduino board, and the LEDs are connected to pins 2, 3, and 4. The program continuously loops and measures the distance using the sensor, then converts the duration of the returned sound wave to distance in centimeters and inches. It then compares the measured distance to a threshold distance and activates the LEDs based on the measured distance. If the distance is greater than the threshold, all LEDs are off, if it's less than the threshold, the LEDs light up gradually depending on the distance from the sensor. The program then waits for 200ms before looping again.

Discussion & Conclusion

In conclusion, the experiment successfully demonstrated the use of an ultrasonic sensor and Arduino to measure distances and activate LEDs based on the distance readings. The results showed that the sensor accurately measured the distance and the LEDs were activated based on the set threshold distances. However, there were some limitations to the experiment, such as the maximum range of the sensor and the accuracy of the measurements at longer distances. Overall, this experiment highlights the potential of using ultrasonic sensors in various applications, such as robotics and automation. Further improvements could be made to increase the accuracy and range of the sensor, as well as incorporating more complex functions into the code.

References

- https://www.arduino.cc
- ATMega328 manual
- https://www.tutorialspoint.com/arduino/arduino_ultrasonic_sensor.htm