American International University- Bangladesh
Department of Electrical and Electronic Engineering
EEE 4103: Microprocessor and Embedded Systems Laboratory

Title: Interfacing the Arduino with an external sensor to establish communication using the RS-232 protocol with implementing an obstacle detection system.

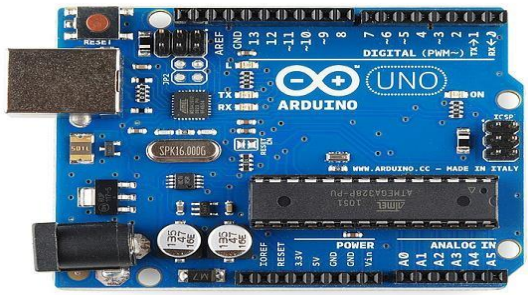
Objectives: In this lab, we will learn –

(i) How to code a simple Obstacle Detection System in Arduino IDE.
(ii) Implement a simple Obstacle Detection System in Hardware.

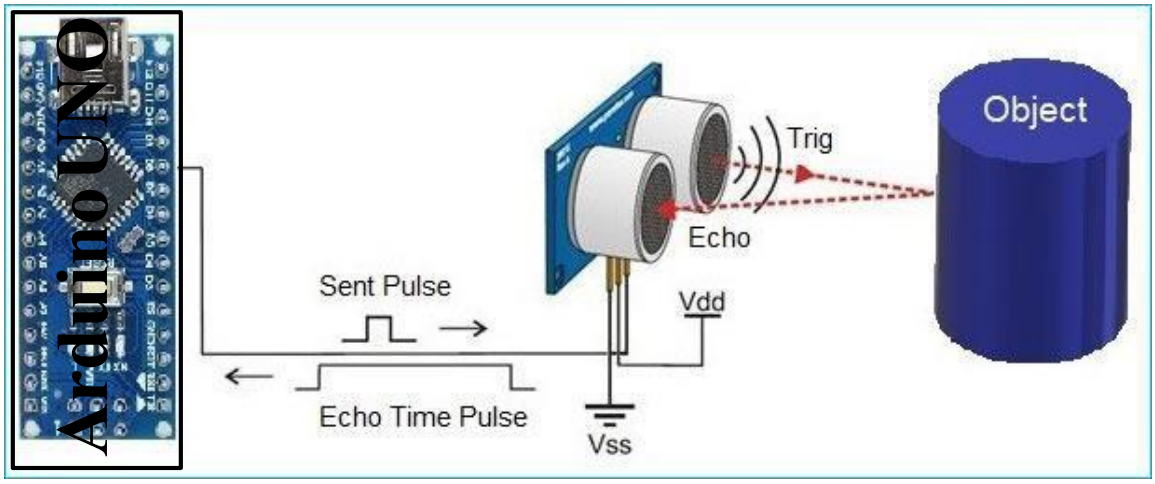
Theory and Methodology:

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 new code into the board. We can easily load a code into the board just using a USB cable and the Arduino IDE (that uses an easier version of C++ to write a code).

Apparatus:

1) Arduino IDE (any version)	Software
2) Arduino Uno (R3) board	
3) Sonar Sensor (HCSR04)	
4) LED	

Overview of Sonar Sensor:



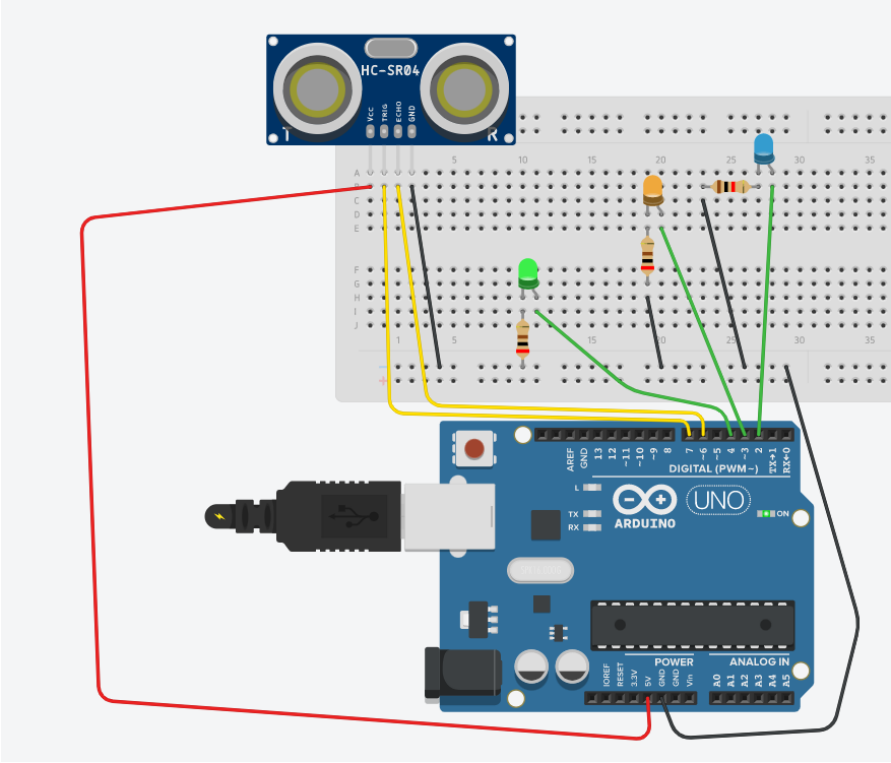
HC-SR04 Sensor Features:

Operating voltage: +5V
Theoretical Measuring Distance: 2cm to 450cm
Practical Measuring Distance: 2cm to 80cm
Accuracy: 3mm
Measuring angle covered: <15°
Operating Current: <15mA
Operating Frequency: 40Hz

Experimental Procedure:

Setting up the Circuit

The main task of our lab is to use a sonar sensor to detect the distance of an obstacle.



How It Works:

Here we are using a sonar sensor (HCS04) to detect the distance of an obstacle and will also glow an LED as soon as it detects the obstacle. HCSR04 is an Ultrasonic ranging module that consists of a transmitter, receiver, and control circuit. It has four pins for VCC, GND, Trigger, and Echo. You can easily interface it with Arduino boards. Using the I/O trigger for at least 10 microseconds of high-level signal, the Module automatically sends eight 40 kHz and detects whether there is a pulse signal back. The Trigger pin of the sensor is connected to digital pin 7 and the Echo pin at digital pin 6 of the Arduino Uno R3 board with connecting wire. And LEDs are connected to pins 2,3 and 4 to show that an obstacle is detected as per conditions applied to the program mentioned below. Here pins 7 and 6 will act as output pins because the trigger will be generated from Arduino and the LED states (HIGH/LOW) will also be changed by the Arduino board. As the ping generated from the Arduino board travels out from the trigger and comes back to the echo, so to find the distance of the object we take half of the distance travelled. As we all know, the speed of sound is 340 m/s or 29 microseconds per centimeter or 74 microseconds per inch. So, the distance covered in centimeter by the trigger will be calculated by the following equation: -

distance_cm=microseconds / 29/ 2;
distance_in=microseconds / 74 /2;

Program:

```
int distanceThreshold = 0;
int cm = 0;
int inches = 0;

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()
{
Serial.begin(9600);
pinMode(2, OUTPUT);
pinMode(3, OUTPUT);
```

```
pinMode(4, OUTPUT);
}

void loop()
{
// set a threshold distance to activate LEDs
//considering the features of ultrasonic sensor
distanceThreshold = 80;
// measure the ping time in cm, 340m/s=0.034cm/µs, therefore 0.034/2=0.017 mainly as the signal is working as echo
cm = 0.017 * readUltrasonicDistance(7, 6);

  // convert to inches by dividing by 2.54

inches = (cm / 2.54);
Serial.print(cm);
Serial.print("cm, ");
Serial.print(inches);
Serial.println("in");

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

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

 if (cm < distanceThreshold-50 && cm > distanceThreshold-100 ) {
digitalWrite(2, HIGH);
digitalWrite(3, HIGH);
digitalWrite(4, LOW);
}

if (cm < distanceThreshold-100 && cm > distanceThreshold-300 ) {
digitalWrite(2, HIGH);
digitalWrite(3, HIGH);
digitalWrite(4, HIGH);
}
delay(100); // Wait for 100 millisecond(s)
}
```

Questions for Report Writing:

Include all codes with explanations into lab report by following the writing template mentioned in appendix A of Laboratory Sheet Experiment 6.

Perform the obstacle detection system in tinkercad and Proteus simulation platform.

Reference(s):

[1] Arduino IDE, https://www.arduino.cc/en/Main/Software accessed on May 3, 2019.
[2] Arduino and Proteus Library, https://etechnophiles.com/add-simulate-ultrasonic-sensorproteus-2018-edition/ accessed on May 3, 2019.
[3] Ultrasonic Distance Sensor in Arduino With Tinkercad https://www.instructables.com/id/Ultrasonic-Distance-Sensor-Arduino-Tinkercad/ accessed on May 3, 2019.