MIT World Peace University

Internet of Things
Second Year B. Tech, Semester 2

Obstacle Detection and Notification using Arduino

Assignment 5

Prepared By

Krishnaraj Thadesar
Cyber Security and Forensics
Batch A2, PA 20

May 2, 2023

Contents

1 Aim

To simulate an operation of obstacle detection and notifying it with a buzzer or LED using Raspberry-Pi/Beagle bone black board/ Tinker CAD Arduino etc.

2 Objectives

1. To develop an obstacle detection system using Arduino, ultrasound sensor, buzzer, and LED and or DC Motor.
2. To demonstrate when the ultrasonic sensor detects an obstacle at an distance(3cm) the Arduino orders the buzzer to ring and the red LED to light up.

3 Equipments Used

Equipment Name	**Quantity**
Raspberry Pi Model 3 B	1
DC Motor	1
Ultrasonic Sensor	1

4 Theory

The objective of this project is to develop an obstacle detection system using an Arduino microcontroller, an ultrasonic sensor, a buzzer, and a LED or a DC motor. The ultrasonic sensor is used to detect obstacles within a certain range, and when an obstacle is detected, the Arduino will activate the buzzer and the LED or the DC motor. This system can be useful in various applications, such as in automobiles, robots, and security systems, where obstacle detection is important.

In this project, the ultrasonic sensor is used to measure the distance between the sensor and the obstacle. The sensor emits high-frequency sound waves and then listens for the echo of those waves bouncing back from the obstacle. The time between the transmission and reception of the sound waves is used to calculate the distance between the sensor and the obstacle.

The ultrasonic sensor used in this project is the HC-SR04, which has four pins: VCC, GND, Trigger, and Echo. The VCC and GND pins are connected to the 5V and GND pins of the Arduino, respectively. The Trigger pin is used to send the ultrasonic sound wave, and the Echo pin is used to receive the echo of the sound wave.

The LED or the DC motor is used to provide a visual or a physical indication of the obstacle detection. The LED is connected to a digital output pin of the Arduino, and it can be turned on or off by writing a high or low value to that pin.

The DC motor is connected to a motor driver module, which is connected to two digital output pins of the Arduino. By varying the voltage and the polarity of the output pins, the direction and the speed of the DC motor can be controlled.

In the context of the Internet of Things (IoT), this project can be extended to include wireless communication between the Arduino and other devices or systems. For example, the Arduino can be connected to a Wi-Fi or a Bluetooth module, which can be used to send the obstacle detection data to a cloud server or a smartphone application. The data can then be processed and analyzed to provide insights and actions based on the detected obstacles.

This can be useful in smart city applications, where the traffic flow can be optimized based on the real-time obstacle detection data. The use of wireless communication in IoT also enables remote monitoring and control of the obstacle detection system, making it more efficient and convenient.

5 Circuit Diagram

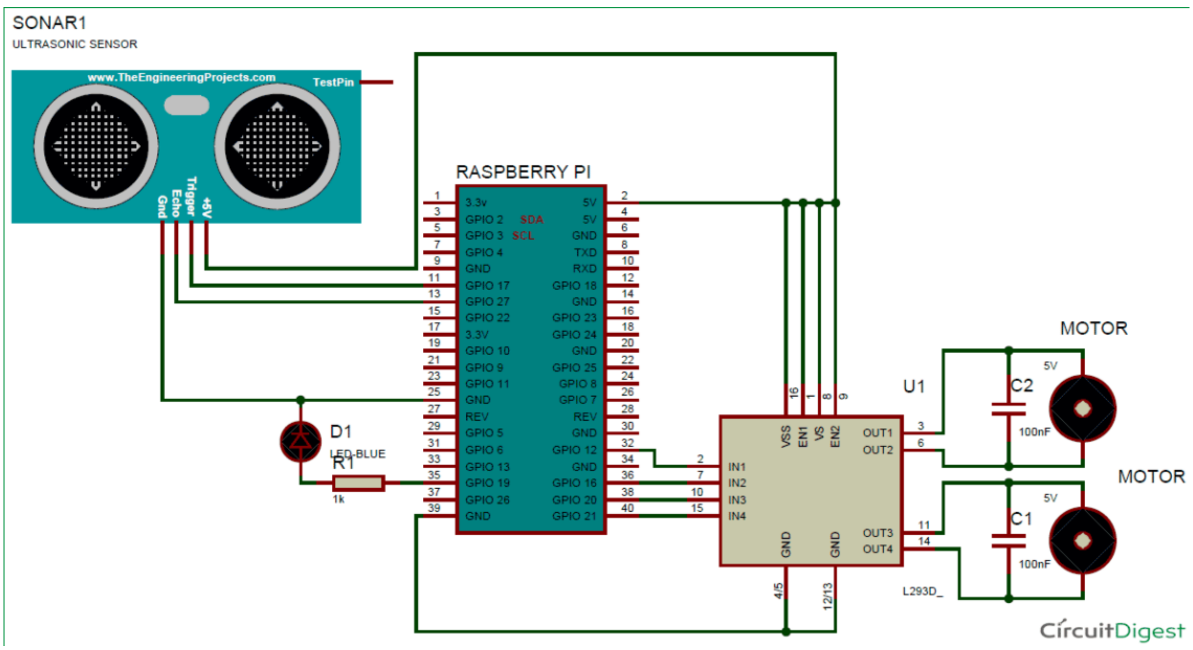

Figure 1: Circuit Diagram

6 Platform

Operating System: Arch Linux x86-64
IDEs or Text Editors Used: Visual Studio Code and Tinkercad, thonny on Pi

7 Input

Data from the Ultrasonic sensor is given to pi.

8 Output

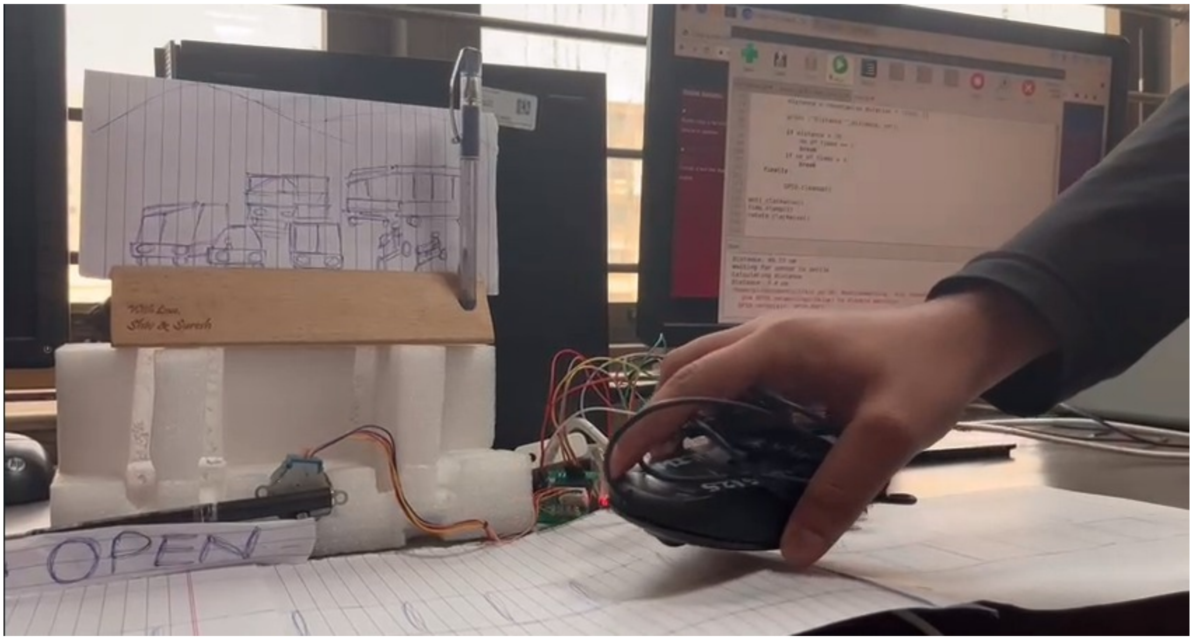

Figure 2: Demonstration of the Project

9 Code

```
import RPi.GPIO as GPIO
import time
GPIO.setmode(GPIO.BOARD)
control_pins = [7,11,13,15]

#Clockwise direction
for pin in control_pins:
  GPIO.setup(pin, GPIO.OUT)
  GPIO.output(pin, 0)
halfstep_seq = [
  [1,0,0,0],
  [1,1,0,0],
  [0,1,0,0],
  [0,1,1,0],
  [0,0,1,0],
  [0,0,1,1],
  [0,0,0,1],
  [1,0,0,1]
]
for i in range(512):
  for halfstep in range(8):
  for pin in range(4):
    GPIO.output(control_pins[pin], halfstep_seq[halfstep][pin])
  time.sleep(0.002)

#Anti Clockwise direction
for pin in control_pins:
```

```
    GPIO.setup(pin, GPIO.OUT)
    GPIO.output(pin, 0)
  halfstep_seq = [
    [1,0,0,1],
    [0,0,0,1],
    [0,0,1,1],
    [0,0,1,0],
    [0,1,1,0],
    [0,1,0,0],
    [1,1,0,0],
    [1,0,0,0]
  ]
  for i in range(512):
    for halfstep in range(8):
    for pin in range(4):
      GPIO.output(control_pins[pin], halfstep_seq[halfstep][pin])
    time.sleep(0.002)
  GPIO.cleanup()

  print ("Script for Controlling Distance Sensor")

try:
      GPIO.setmode(GPIO.BOARD)
      PIN_TRIGGER = 7
      PIN_ECHO = 11
      GPIO.setup(PIN_TRIGGER, GPIO.OUT)
      GPIO.setup(PIN_ECHO, GPIO.IN)
      GPIO.output(PIN_TRIGGER, GPIO.LOW)
      print ("Waiting for sensor to settle")
      time.sleep(2)
      print ("Calculating distance")
      GPIO.output(PIN_TRIGGER, GPIO.HIGH)
      time.sleep(0.00001)
      GPIO.output(PIN_TRIGGER, GPIO.LOW)
      while GPIO.input(PIN_ECHO)==0:
            pulse_start_time = time.time()
      while GPIO.input(PIN_ECHO)==1:
            pulse_end_time = time.time()
      pulse_duration = pulse_end_time - pulse_start_time
      distance = round(pulse_duration * 17150, 2)
      print ("Distance:",distance,"cm")

finally:

      GPIO.cleanup()
```

10 Conclusion

Thus, we have successfully simulated an operation of obstacle detection and notifying it with a DC motor.

11 FAQ

Details of the main components used in this project and their pin specifications:

1. Raspberry Pi Model 3 B:
   The Raspberry Pi has a total of 40 pins, including 26 GPIO (General Purpose Input/Output) pins, 3.3V and 5V power pins, and ground pins. The pinout diagram for the Raspberry Pi 3 B can be found on the official Raspberry Pi website.

   - Pin 1: 3.3V
   - Pin 2: 5V
   - Pin 3: GPIO 2
   - Pin 4: 5V
   - Pin 5: GPIO 3
   - Pin 6: Ground
   - Pin 7: GPIO 4
   - Pin 8: GPIO 14
   - Pin 9: Ground
   - Pin 10: GPIO 15
   - Pin 11: GPIO 17
   - Pin 12: GPIO 18
   - Pin 13: GPIO 27
   - Pin 14: Ground
   - Pin 15: GPIO 22
   - Pin 16: GPIO 23
   - Pin 17: 3.3V
   - Pin 18: GPIO 24
   - Pin 19: GPIO 10
   - Pin 20: Ground
   - Pin 21: GPIO 9
   - Pin 22: GPIO 25
   - Pin 23: GPIO 11
   - Pin 24: GPIO 8
   - Pin 25: Ground
   - Pin 26: GPIO 7
   - Pin 27: ID SD
   - Pin 28: ID SC
   - Pin 29: GPIO 5
   - Pin 30: Ground
   - Pin 31: GPIO 6

- Pin 32: GPIO 12
- Pin 33: GPIO 13
- Pin 34: Ground
- Pin 35: GPIO 19
- Pin 36: GPIO 16
- Pin 37: GPIO 26
- Pin 38: GPIO 20
- Pin 39: Ground
- Pin 40: GPIO 21

2. DC Motor:
   A DC motor typically has two wires for power and two wires for control. The power wires are typically red and black, and the control wires can be any other color. The control wires are used to vary the voltage and polarity of the applied power to control the speed and direction of the motor.

3. Ultrasonic Sensor (HC-SR04): The HC-SR04 ultrasonic sensor has four pins: Vcc, Trig, Echo, and Gnd. The pin specifications are as follows:

   - Vcc: This pin is used to provide power to the sensor. It typically requires a voltage of 5V, but it can also be powered using 3.3V.
   - Trig: This pin is a digital output pin that is used to trigger the sensor. It sends a 10µs pulse to the sensor to start the measurement.
   - Echo: This pin is a digital input pin that receives the echo signal. It measures the time taken for the ultrasonic waves to travel to the object and back, and sends a pulse back to the Raspberry Pi to indicate the distance.
   - Gnd: This pin is used to connect the sensor to ground.