

EXPERIMENT NO: 1**1. Using raspberry pi**

- a. Calculate the distance using a distance sensor.
- b. Basic LED functionality.

AIM:**Using raspberry pi**

- a. Calculate the distance using a distance sensor.
- b. Basic LED functionality.

PROGRAM:

A. Use a Raspberry Pi to calculate the distance using a distance sensor:

- Connect the distance sensor to the Raspberry Pi's GPIO pins.
- Install the required libraries and packages (e.g., RPi.GPIO, pigpio, time, etc.).
- Write a script in Python to read the sensor data and calculate the distance using a formula based on the sensor's specifications.
- Use the formula to convert the sensor data into a distance value and display the output on the screen or send it to a remote server for storage and processing.
- Run the script on the Raspberry Pi to get the distance measurements.

Here's an example code that demonstrates this process:

```
import RPi.GPIO as GPIO
import time
# Set the GPIO mode
GPIO.setmode(GPIO.BCM)
# Define the GPIO pin for the sensor
TRIG = 23
ECHO = 24
# Set up the GPIO pins for the sensor
GPIO.setup(TRIG, GPIO.OUT)
GPIO.setup(ECHO, GPIO.IN)
# Set the trigger pin to low
GPIO.output(TRIG, False)
# Wait for the sensor to settle
```

III-II B.TECH CSE (IoT)**IoT LAB MANUAL**

```
time.sleep(2)
# Send a 10uS pulse to trigger the sensor
GPIO.output(TRIG, True)
time.sleep(0.00001)
GPIO.output(TRIG, False)
# Measure the pulse duration
while GPIO.input(ECHO)==0:
    pulse_start = time.time()
while GPIO.input(ECHO)==1:
    pulse_end = time.time()
# Calculate the distance
pulse_duration = pulse_end - pulse_start
distance = pulse_duration * 17150
distance = round(distance, 2)
# Display the distance
print("Distance:", distance, "cm")
# Clean up the GPIO pins
GPIO.cleanup()
```

OUTPUT: