Experiment 02

Range Detection

In this experiment, we will look at different sensors that we can use to measure distance. We will learn their working principles and discuss the applications where a certain type of sensor is preferred over the other.

Also, we will send the sensor data to the ThingSpeak platform for remote monitoring

Background

Range measurement sensors are widely used in day-to-day applications. These sensors are used to determine the distance between two bodies. There are different types of range measurement sensors available in the current market. Commonly available types include ultrasonic sensors, IR Triangulation sensors, LIDARs, and IR time of flight sensors. One of the aspects of distance measurement sensors is proximity assessment. Proximity detection sensors only require a body to be present in its sensing range to be activated. They are not used to extract the distance between the body and the sensor, but rather a binary value to represent the presence of an object within the configured sensing distance. Range measurement sensors on the other hand are used to accurately measure the distance between the body and the sensor.

Ultrasonic Sensors

Ultrasonic sensors are very simple, cheap, and very commonly used. They work by transmitting an ultrasonic sound signal. The sound wave propagates till it reaches an obstacle. As the sound wave hits the obstacle, a certain portion of it is absorbed and the rest is reflected. The sensor detects this reflected sound wave. The sensor now calculates the total time taken by the sound wave from the moment it is emitted till it reaches back. The velocity of the sound wave traveling in the air when combined with the time taken for the wave to travel across gives the distance between the sensor and the object.

IR Triangulation Sensors

These sensors have a transmitting IR LED, which emits the IR signal/wave. Unlike the ultrasonic sensor, on the receiving side of the IR sensor, a position-sensitive IR detector is used. This detector measures the angle at which the reflected wave reaches back. As the distance between the sensor and the object varies, the angle of the reflected wave proportionally varies. The sensor triangulates the distance based on the angle the reflected wave makes at the detector.

IR Time of Flight Sensor

Time of Flight (ToF) sensors work similarly to ultrasonic sensors. The ToF sensors also measure the elapsed time from the emission of the light pulse till the reflected pulse is detected at the receiver. Since the speed of light is constant, measuring the time delay will directly give you the distance between the sensor and the object. These types of sensors are accurate, with very high refresh rates.

LiDAR – Light Detection and Ranging

LiDAR uses the same principle of time of flight sensors but instead of using a single beam, LiDARs are capable of emitting multiple beams at the same time. This gives them the ability to analyze depth information by comparing the individual beam times, therefore, allowing the user to accurately map a 3D surface. The variation of distance at the individual point of beam incidence is realized to estimate the texture in 3D of the measuring surface. Scaling this up can help map an entire area accurately to generate 3D views.

💻Experiment Set-up: Configuration

The **HC-SR04 ultrasonic module** uses high-frequency sound signals that are out of the hearing range (greater than 20KHz). The sensor can read distances from 2cm to 400cm with an accuracy of up to 0.3cm. The module consists of an ultrasonic transmitter, receiver, and the circuitry necessary to generate the required pulses. Unlike most other circuits on the board, this module does not use I2C communication. The ultrasound transmitter generates a signal at 40KHz when we provide an input trigger at the trig pin. When the receiver detects the reflected wave a trigger signal is generated by the sensor at the echo pin. Now the Node MCU calculates the time between these two trigger pulses. This time information is used to calculate the distance.

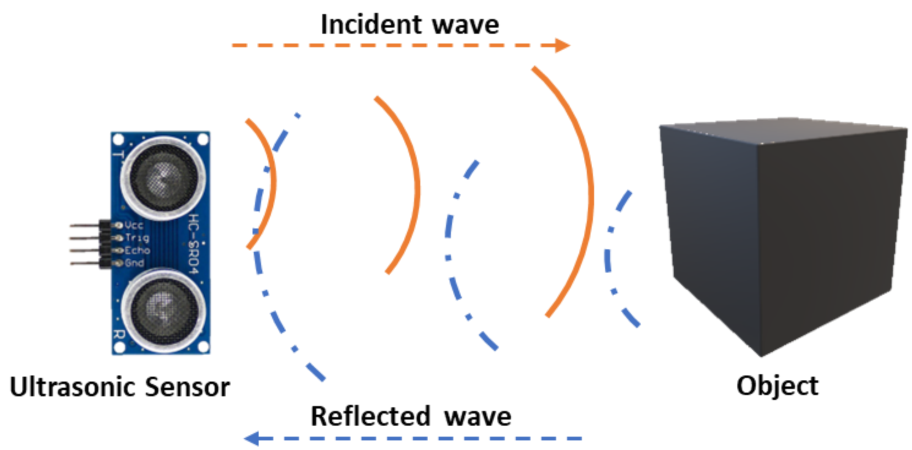

Fig1: The sensor emits an incident sound wave which travels till it hits the surface of the object. Then a portion of the wave gets reflected to the sensor. The receiver detects the reflected wave and generates the output trigger signal.

The trigger pulse is set high for at least 10us. In this interval, eight 40kHz signals are transmitted. The receiver will detect the reflected waves when the transmitted waves hit an object and reflect. The distance measured is equal to the product of the high-level time of received pulse and velocity of sound divided by 2. We divide the time by 2 because the time measured includes both the incident and the reflected periods.

For testing the sensor, place a piece of cardboard facing the ultrasonic sensor a few feet away from the board. Measure the distance using the Arduino output console. Keep moving the cardboard towards the board and away from it. Notice the distance values displayed on the output console.

Instructions - Part 1

1. Look at the board schematic in the manual and determine the pin connections. Identify which pins of the Node MCU are connected to the Trig and Echo pins of the sensor module.
2. Open Arduino and click on “File” from the Menu bar. Click on “Open” to open the “distance.ino” file. The “distance.ino” file can be found in the course material for this module. The IDE now loads the program. Edit the code to add the pin connections.
3. Once you’ve made changes to the code, flash the program. Refer to the IoT Board Manual for flashing instructions.
4. Open the Serial Monitor from the dropdown menu under “Tools” on the Menu bar. Refer to the IoT Board Manual for setting up the Serial Monitor window. In the serial monitor, the distance value measured is refreshed constantly.
5. Now start moving the cardboard sheet closer to the sensor. Notice how the values are changing. This can help you understand the resolution of the sensor. Estimate the resolution and report it in the submission.

Instructions - Part 2

1. Create an account on ThingSpeak.com. Use your UF email and select the free trial
2. Create a new channel. Add a name to identify it with the project. Add a description - since we are measuring cms, add a field called Distance (cm)
   a. All other details are not necessary
3. Obtain your API Writing key and channel ID number
4. Open the "distance_thingspeak.ino" file from the course materials
5. Install the libraries ThingSpeak (by MathWorks) and WiFi (Arduino Uno WiFi Dev Ed Library by Arduino). Go to Tools→Manage Libraries→ and use the search engine
6. In the code, change your SSID and password, and add the API writing key and channel ID number noted in step 2
7. Flash the program and see what happens in the Serial Monitor and in your channel in the ThingSpeak website

☞ Deliverables

Demonstration:

1. Record a video demonstration explaining the outcome of the experiment. Refer to the title page for a brief description of the expected outcome. Make sure you talk over all observations and the video is presentable. Also, don't forget to show the data updating in real time on the serial monitor and in your ThingSpeak channel.
2. Address the following items in your recording or add it as text in the submission:
   a. Determine the range of the ultrasonic sensor experimentally.
   b. How does the sensor respond when the receiver signal is tampered with? (Join with another team and try placing sensors opposite to each other)
   c. Consider a situation where the environment of the experiment site is changed. Let's assume that we are on the surface of Mars. Now assume we are going to use the same ultrasonic sensor. What changes in the provided code are to be made and why?
   d. Following the previous question, what type of distance-measuring sensor would you use when the medium between the sensor and the object keeps changing? Provide reasons as to why one option is superior to the other.

References and Further Reading

[1] https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf
[2] https://global.sharp/products/device/lineup/data/pdf/datasheet/gp2y0a21yk_e.pdf
[3] https://static.garmin.com/pumac/LIDAR-Lite%20LED%20v4%20Instructions_EN-US.pdf
[4] https://oceanservice.noaa.gov/facts/lidar.html
[5] https://www.st.com/en/imaging-and-photonics-solutions/vl53l0x.html