Possible Collision Sensing Device

By Abeed Arefin
Student # 216485435

Abstract/Executive Summary

This project is about a device whose main objective is to detect any object in its path and alert everyone if the object gets too close to the device to avoid any possible collision. The device detects any object from two sides (front and back) and alerts everyone using 2 sets of LEDs and a buzzer. As can be seen in the graphical abstract below, five LEDs and one buzzer is used to alert from one side and another five LEDs, and one buzzer is used to alert from another side. When any object gets closer to any one of the ultrasonic sensors, the LED for that region lights up keeping any previous LEDs (if any) lighted up. At the same time, the buzzer starts beeping with a certain frequency. If the object gets closer the buzzer beeping frequency increases and vice-versa. If the object goes far away, the LEDs go off one by one. This way the device alerts everyone for any possible collision. Arduino DUE microcontroller is used to control all the appliances and ultrasonic sensors are used to detect any object and measure its distance by the written program. The program is written by Arduino IDE. The project is successful and meets all the requirements. The author plans to further improve this project and make use of the experience gathered from this project for future projects.

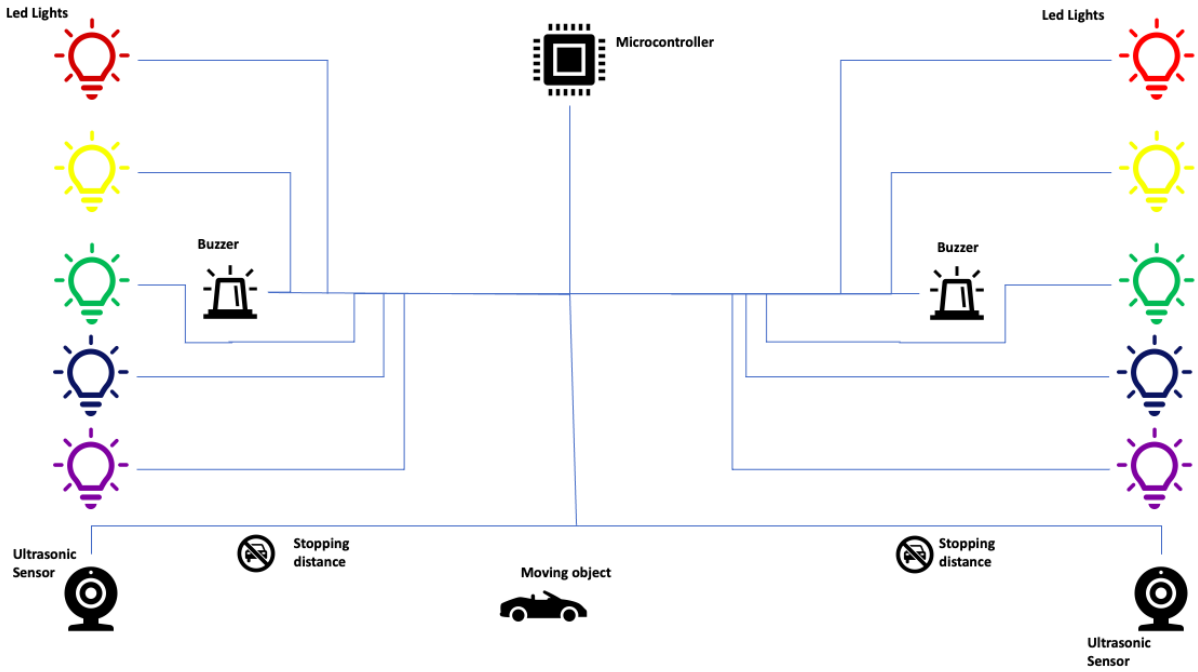

Figure 1 : Graphical Abstract

Table of Contents

Table of Figures

Introduction

This project was made as a requirement for EECS 2032 course. This project was selected because through this project an interest in software coding was showcased. A motion sensor device is built and programmed from scratch. In the future, the knowledge and experience gathered from this project can be easily utilized to work on a motion-sensing device for an automated car or robot. This device detects any object in its path from the two sides and measures the distance and decides whether to alert about the possible collision with the object or not.

Automated emergency braking (AEB) widely used in recent automated cars like the cars manufactured by Tesla worked as the motivation for this project. There is various kind of sensors and technologies used to detect any object and operate automated emergency braking. The idea of using ultrasonic sensors and Arduino DUE to make a distance tracking device is inspired by a video whose link [1] is provided in the first reference in the reference section. This device can be compared to a motion-sensing device where the Arduino microcontroller is used, and this type of project can be easily found on the internet. Radar can also be compared to this device. A radar also detects an intruder or a possible collision.

An electrical circuit was needed to be designed for this device and this was done from the scratch. For this electric circuit design and implementation, the learnings and experience gathered from EECS 2200, Electrical Circuits, course was helpful. And as for the software component, Arduino software is used where C is the main compiled language.

From this project, the experience of using sensors to give inputs to a microcontroller and coding the microcontroller for appropriate behavior can be easily gathered. This project is also one of the first project done by the author where complete circuit design and implementation was done. The experience gathered from this type of project and this course will surely help anyone in their future embedded systems projects and eventually help in their future career prospects.

In the time of the COVID-19 pandemic, the completion of this project gave a very vivid and clear concept of Embedded Systems. Though it sounds interesting, there were various challenges associated with it. The hardware design and budget problems were huge factors for the success of this project. This project might not have any market value but the various implementation of this concept of tracking and alerting and taking a decision based on that is widely used in modern times.

Proposed Project

An Arduino DUE microcontroller was used in this device to get the input from the sensor (ultrasonic sensors in this case) and make a decision and further action based on the input from the sensor. Here, after getting the input from the two ultrasonic sensors, the Arduino DUE microcontroller takes a decision with the help of the code inputted to it by Arduino software and then takes further actions.

The device is set up in such a way that five LEDs and one buzzer are assigned for each ultrasonic sensor. Each LED represents a particular distance. When an object passes a particular distance, the LED assigned for that distance lights up, which demonstrates that the device is being able to sense the object and track its distance. The buzzer also starts ringing with a specific frequency. The closer an object goes to any of the ultrasonic sensors, the more LEDs blinks keeping any previous LEDs (if any) lighted up. The beeping frequency of the buzzers also increases. When the object goes closest to the device, the LEDs constantly blink, and the buzzer beeps with the highest frequency. If the object goes far from any of the ultrasonic sensors, the LEDs goes off one by one. The frequency of the buzzer beeping also goes down at the same time. This way it alerts for any possible incoming collision. This project showcases the use of sensors and the use of software to make decisions based on that. An Arduino DUE microcontroller was used in this device to get the input from the sensor (ultrasonic sensors in this case) and make a decision and further action based on the input from the sensor. Here, after getting the input from the two ultrasonic sensors, the Arduino DUE microcontroller takes a decision with the help of the code inputted to it by Arduino software and then takes further actions.

Apart from an Arduino microcontroller and two HC-SR04 ultrasonic sensors, two buzzers, ten LEDs, ten resistors, two breadboards, and connecting cables are also used in this device.

A. Program

In the following program flowchart inserted on the next page, we can see that in the beginning all the constants and variables are defined. Here all the input and output portals are defined and declared. Then in the void setup section, as it can be seen in the flowchart below, the ultrasonic sensors are set up. The ports for the trigger and echo of the ultrasonic sensors are set up for output and input respectively. In this section, all the LEDs and buzzers are also set up. To do this pinMode() function of Arduino IDE is used. The use of this function in this program can be seen in the inserted picture in the Appendix section of this report. Then by using the Serial function, the communication between the ultrasonic sensor and the

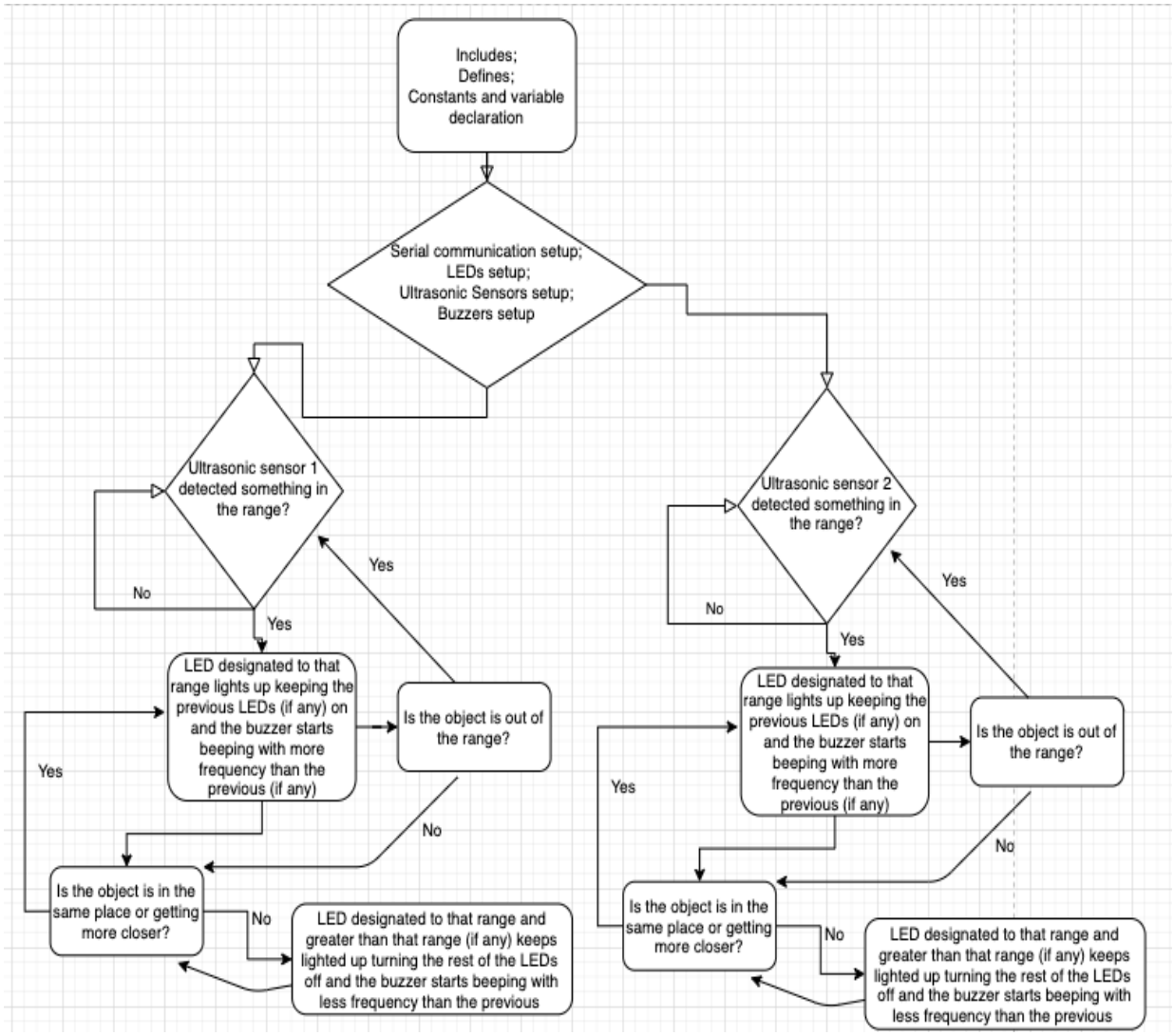

Figure 2 : Program Flowchart

microcontroller is established.

A method, named “measure”, is then added to the program. This method takes the trigger and echo port of a particular ultrasonic sensor and returns the distance of the object from the sensor. This method is added at the end of the void loop section as seen in the inserted picture in the Appendix section of this report.

In the void loop section of this program, the distance of any object is calculated from both the ultrasonic sensors. The distance of the second ultrasonic sensor is termed as ‘distance2’ as seen in the image in the Appendix section. Then the application or the functionality is programmed with the help of few if-else statements. The first five if-else statements are written for the behavior of the device for the object detected by the first ultrasonic sensor. The functionality of this program can be

explained by the help of the program flowchart inserted above.

As it can be seen in the program flowchart, if nothing is detected in the set range of the first ultrasonic sensor, then nothing happens and again the sensor searches for any object. And if the sensor detects any object, then the LED designated for that distance or region lights up and the buzzer starts to beep at a certain frequency. The frequency of beeping is lowest for the farthest distance and highest for the closest distance. If the object gets closer, the LED of that region lights up keeping previous LEDs lighted up, and also the beeping of the buzzer increases. When the object gets closest, the LEDs constantly blink and the buzzer beeps with the highest frequency. If the object goes far from the sensor, the LEDs go off one by one as seen in the flowchart. The same is or the second ultrasonic sensor.

The programming of distance tracking or distance measuring is taken from the second link [2] in the reference section. The use of pinMode and digitalWrite function is learned from the third link [3] and fourth link [4] in the reference section. The connection setup between the microcontroller and the sensor is taken from the fourth link [5] in the reference section. And all the other programming and problem solving is done by the author himself.

The most challenging part was to create different buzzing frequencies for different distances. As there Arduino DUE does not support the tone function, appropriate use of delay and digitalWrite function was done in this program as seen in the image in the Appendix section. A different method for measuring the distance was programmed to eliminate any confusion and for better readability.

B. Hardware

An Arduino microcontroller and two HC-SR04. Apart from an Arduino microcontroller and two HC-SR04 ultrasonic sensors, two buzzers, ten LEDs, ten resistors, two breadboards, and connecting cables are also used to build this device.

As can be seen in the hardware schematic below, the LEDs are connected with the resistor and the microcontroller in series. Each LED and resistor group uses one port from the microcontroller. And all the LEDs and resistors are also connected to the ground. The two buzzers are also connected to the microcontroller using two separate ports and they are also connected to the ground.

Then two ultrasonic sensors are connected to the 5V VCC supply of the Arduino DUE microcontroller. Both of them are connected through the same 5V VCC port of the microcontroller as seen in the schematic. For this,

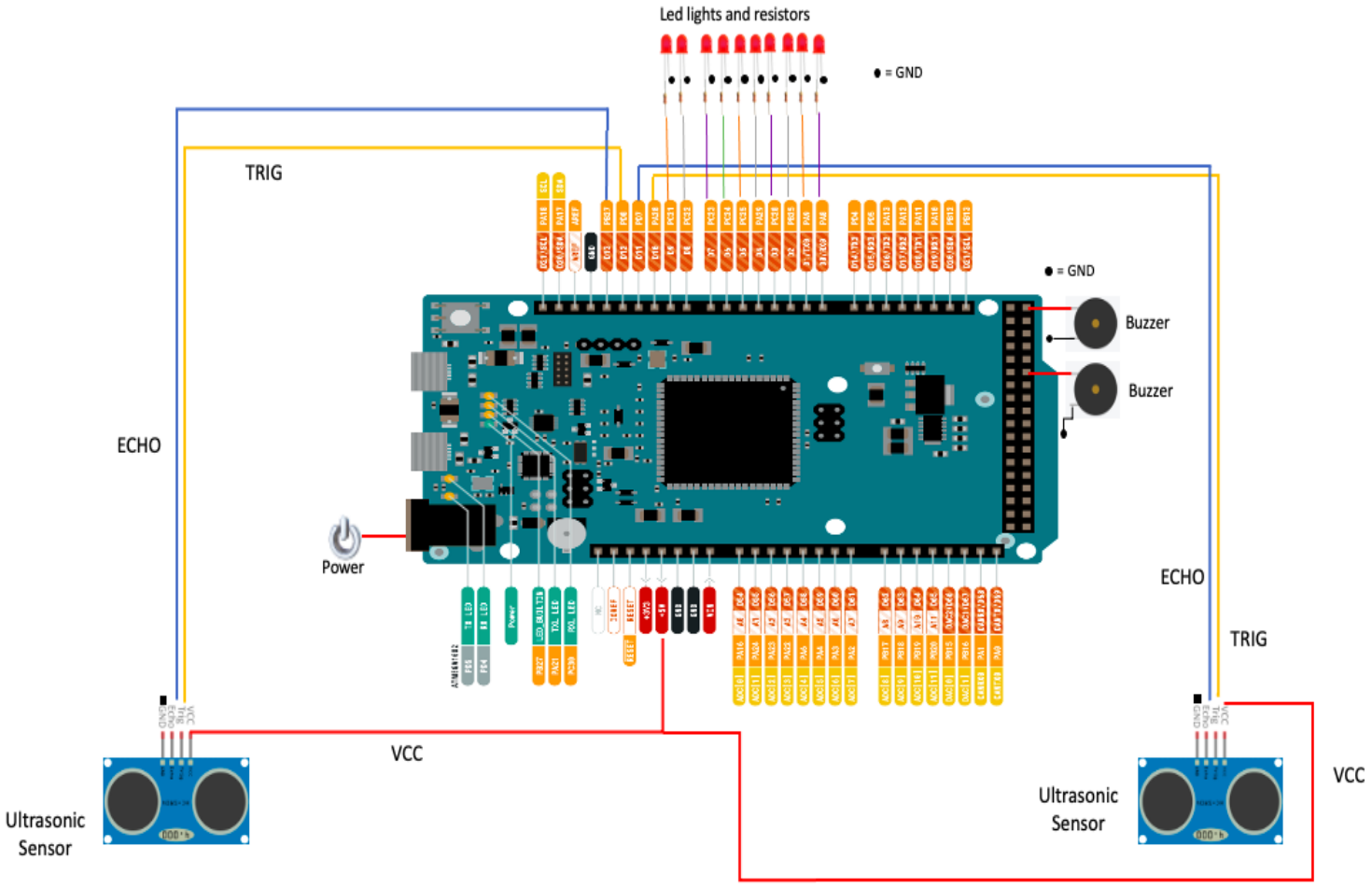

Figure 3 : Hardware Schematic

one of the power rails of a breadboard is connected to the 5V VCC port of the microcontroller. Then power is supplied to both the sensors using connecting wires. Their trigger and echo ports are also connected to the 4 separate ports of the microcontroller as there are two sensors. The two sensors are also connected to the ground. The ground is connected to one of the ground ports of the microcontroller. Here one of the power rails is also used to connect the grounds to all the appliances. And then connecting wires are used to connect the appliances to the ground. Appropriate use of breadboard and connecting wires is one of the important design tricks which helped to make this device within a limited budget. Four groups of long connecting wires are used to place the second breadboard and second ultrasonic sensor far away from the microcontroller and the first ultrasonic sensor.

Results

This project is the result of the project being divided into small parts and then built. At first, the useability of the ultrasonic sensor was tested. To do that one ultrasonic sensor was connected with the Arduino DUE microcontroller and then the distance was calculated and printed on the screen by using the System.print function. Then LEDs were connected with the microcontroller and resistors in series and the microcontroller is programmed to build a device that detects and alerts when an object is near only from one side.

After that buzzer was added to it and a constant sound to the buzzer was added. Then it was a challenge to come up with a way to give different frequencies in the buzzer for different distances. This challenge was easily solved by the appropriate use of the delay function. After the complete completion of the application of one ultrasonic sensor and LEDs and buzzer associated with it, the second ultrasonic sensor was connected with the Arduino DUE microcontroller. And other LEDs and buzzer was connected to the microcontroller. After confirming the complete functionality of the device, long connecting wires were used to place the second ultrasonic sensor far away from the first one and thus the successful completion of the project is achieved.

If an object is detected in the range of the first LED, then the LED lights up, as shown in the picture below, and also the buzzer starts beeping with the lowest frequency.

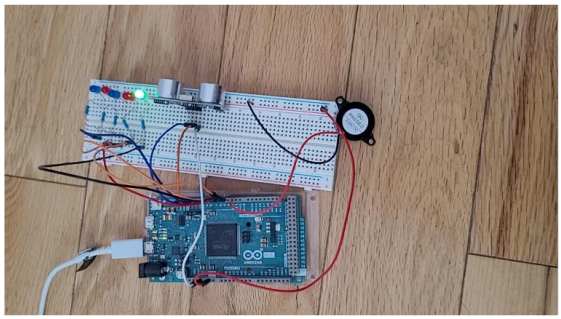

Figure 4 : Picture showing what happens when an object is detected

Then when the object gets closer to the sensor, the LEDs continue to light up keeping the previous LEDs lighted up as shown below in the picture. The buzzer also starts beeping with a higher frequency.

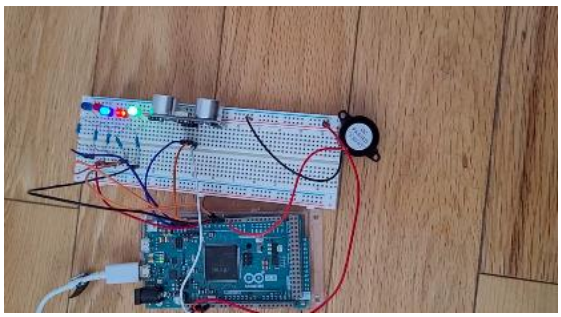

Figure 5 : Picture showing what happens when object gets closer to the device

When it reaches the closest to the device or the sensor, all the LEDs constantly blinks as shown below in the picture and the buzzer starts beeping with the highest frequency.

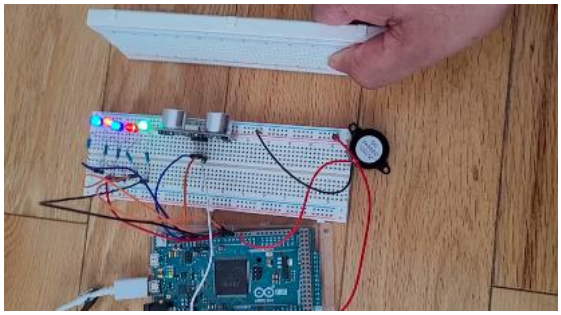

Figure 6 : Picture showing what happens when an object is closest to the device

When the object goes far from the device or the sensor, the LEDs go off one by one until there is none when the object is out of the set range. The buzzer beeping sound also decreases until it goes off. The same is also applicable for the second side also. This way the device detects any object in its path and alerts everyone for a possible collision.

Discussion

The device works perfectly as it is expected. But the ultrasonic sensor sometimes detects the object at the wrong distance because the ultrasonic sensor is never 100% accurate. Its measurements vary normally up to 1 cm. It will not be a big problem as the range of collision detection is far more than 1 cm.

The buzzers used are of two types to distinguish the sounds of the buzzers. There were a few syntax and logical errors while coding the program which has been taken care of with the help of a debugger. There was also a problem with placing the second ultrasonic sensor quite far away from the first ultrasonic sensor. This also has been resolved by the appropriate use of connecting wires. This type of technology is being used in automated cars today whose future is noteworthy.

Conclusion

The main objective of this project was to create a device that detects any object in its path and alerts everyone about a possible incoming collision. Overall, the project can be termed as a successful one. For the detection of the object, normal HC-SR04 ultrasonic sensors were used. It cost less than ten dollars and this way the budget was conserved. If the budget was more, far better ultrasonic sensors could have been used and thus giving far better results. This device can be easily assembled for less than 200 dollars.

To make this project a successful one, despite of the COVID-19 pandemic, this project was divided into small sub-projects as it is mentioned earlier in the results section. The experience of breaking a project into smaller parts and completing them to have a successful project will help anyone in their future academic project management. This project is also one of the first project done by the author where complete circuit design and implementation was done. The experience gathered from this type of project and this course will surely help anyone in

their future embedded systems projects and eventually help in their future career prospects.

Acknowledgement

Professor Ebrahim Ghafar-Zadeh has contributed to this project by teaching the basics of Arduino and hardware settings. Special thanks go to the teaching assistant, Parham Mohammedi, of this course. He has tracked the progress of this project throughout the term and provided valuable feedback whenever needed.

References

1. "Ultrasonic Sensor HC-SR04 and Arduino Tutorial," 26-Jul-2015. [Online]. Available: https://www.youtube.com/watch?v=ZejQOX69K5M. [Accessed: 25-Apr-2021].
2. Distance Measurement using Ultrasonic Sensor and Arduino," *Circuitdigest.com*, 27-Jun-2015. [Online]. Available: https://circuitdigest.com/microcontroller-projects/arduino-ultrasonic-sensor-based-distance-measurement. [Accessed: 25-Apr-2021].
3. "digitalWrite()," *Arduino.cc*. [Online]. Available: https://www.arduino.cc/reference/en/language/functions/digital-io/digitalwrite/. [Accessed: 25-Apr-2021].
4. "pinMode()," *Arduino.cc*. [Online]. Available: https://www.arduino.cc/reference/en/language/functions/digital-io/pinmode/. [Accessed: 25-Apr-2021].
5.. "Serial.begin()," *Arduino.cc*. [Online]. Available: https://www.arduino.cc/reference/en/language/functions/communication/serial/begin/. [Accessed: 25-Apr-2021].

Abeed Arefin (Author)

(Oct.'98) is currently pursuing Software Engineering at York University. He is now in the second year of his studies. He is attending as an international student at York University. He is also working as a part timer in McDonald's.

Appendix

The code or the programming part of the device is attached below.

```
// All the constants are declared here

const int trig = 12; // This is the trigger port for ultrasonic sensor 1
const int echo = 13; // This is the echo port for ultrasonic sensor 1

const int trig2 = 22; // This is the trigger port for ultrasonic sensor 2
const int echo2 = 23; // This is the echo port for ultrasonic sensor 2

const int buzzer1 = 2; // This is the port for buzzer for ultrasonic sensor 1
const int buzzer2 = 3; // This is the port for buzzer for ultrasonic sensor 2

const int LED1 = 8; // This is LED 1 port for ultrasonic sensor 1
const int LED2 = 7; // This is LED 2 port for ultrasonic sensor 1
const int LED3 = 6; // This is LED 3 port for ultrasonic sensor 1
const int LED4 = 5; // This is LED 4 port for ultrasonic sensor 1
const int LED5 = 4; // This is LED 5 port for ultrasonic sensor 1

const int LED6 = 24; // This is LED 1 port for ultrasonic sensor 2
const int LED7 = 25; // This is LED 2 port for ultrasonic sensor 2
const int LED8 = 26; // This is LED 3 port for ultrasonic sensor 2
const int LED9 = 27; // This is LED 4 port for ultrasonic sensor 2
const int LED10 = 28; // This is LED 5 port for ultrasonic sensor 2

int distance = 0; // This is used to store the distance of the object from ultrasonic sensor 1

int distance2 = 0; // This is used to store the distance of the object from ultrasonic sensor 2

void setup() // This method is used to do the initial setup
{
  // The trigger and echo ports are initialized for ultrasonic sensor 1
  pinMode(trig , OUTPUT);
  pinMode(echo , INPUT);

  // The trigger and echo ports are initialized for ultrasonic sensor 2
  pinMode(trig2 , OUTPUT);
  pinMode(echo2 , INPUT);

  pinMode(buzzer1,OUTPUT); // The buzzer for ultrasonic sensor 1 is initialized
  pinMode(buzzer2,OUTPUT); // The buzzer for ultrasonic sensor 2 is initialized


  // The LEDs for ultrasonic sensor 1 is initialized
  pinMode(LED1 , OUTPUT);
  pinMode(LED2 , OUTPUT);
  pinMode(LED3 , OUTPUT);
  pinMode(LED4 , OUTPUT);
  pinMode(LED5 , OUTPUT);

  // The LEDs for ultrasonic sensor 2 is initialized
  pinMode(LED6 , OUTPUT);
  pinMode(LED7 , OUTPUT);
  pinMode(LED8 , OUTPUT);
  pinMode(LED9 , OUTPUT);
  pinMode(LED10 , OUTPUT);

  Serial.begin(9600); // A connection is being setup with the sensors and microcontroller

}

void loop()
{
    // This is used to store the distance of the object from ultrasonic sensor 1
    distance = measure(trig, echo);
    // This is used to store the distance of the object from ultrasonic sensor 2
    distance2 = measure(trig2, echo2);

  // These conditions and codes are for ultrasonic sensor 1
  // This if-else statement is used to generate the alert when a object is closest
  if ( distance <= 10){
    digitalWrite(buzzer1,HIGH); // Buzzer is set active
    // The LEDs are lighted up
    digitalWrite(LED1, HIGH);
    digitalWrite(LED2, HIGH);
    digitalWrite(LED3, HIGH);
    digitalWrite(LED4, HIGH);
    digitalWrite(LED5, HIGH);
    delay(50); // A short delay to show constant LED blinking and generate high beeping frequency
    // The LEDs are turned off
    digitalWrite(LED1, LOW);
    digitalWrite(LED2, LOW);
    digitalWrite(LED3, LOW);
    digitalWrite(LED4, LOW);
    digitalWrite(LED5, LOW);
    delay(50); // A short delay to show constant LED blinking and generate high beeping frequency
```

```
  // The LEDs are lighted up
  digitalWrite(LED1, HIGH);
  digitalWrite(LED2, HIGH);
  digitalWrite(LED3, HIGH);
  digitalWrite(LED4, HIGH);
  digitalWrite(LED5, HIGH);
}
else
{
  // The LEDs are turned off
  digitalWrite(LED1, LOW);
  digitalWrite(LED2, LOW);
  digitalWrite(LED3, LOW);
  digitalWrite(LED4, LOW);
  digitalWrite(LED5, LOW);
  digitalWrite(buzzer1,LOW); // Buzzer is turned off
}

// This if-else statement is used to generate the alert when a object is a little bit far
if ( distance <= 20 && distance > 10)
{
  // The LEDs are lighted up
  digitalWrite(LED2, HIGH);
  digitalWrite(LED3, HIGH);
  digitalWrite(LED4, HIGH);
  digitalWrite(LED5, HIGH);
  digitalWrite(buzzer1,HIGH); // Buzzer is set active
  delay(100); // A delay to get desired beeping frequency
  digitalWrite(buzzer1,LOW); // Buzzer is turned off
  delay(100); // A delay to get desired beeping frequency
}
else
{
  // The LEDs are turned off
  digitalWrite(LED2, LOW);
  digitalWrite(LED3, LOW);
  digitalWrite(LED4, LOW);
  digitalWrite(LED5, LOW);
  digitalWrite(buzzer1,LOW); // Buzzer is turned off
}



// This if-else statement is used to generate the alert when a object is a more far
if ( distance <= 30 && distance > 20)
{
  // The LEDs are lighted up
  digitalWrite(LED3, HIGH);
  digitalWrite(LED4, HIGH);
  digitalWrite(LED5, HIGH);
  digitalWrite(buzzer1,HIGH); // Buzzer is set active
  delay(200); // A delay to get desired beeping frequency
  digitalWrite(buzzer1,LOW); // Buzzer is turned off
  delay(200); // A delay to get desired beeping frequency
}
else
{
  // The LEDs are turned off
  digitalWrite(LED3, LOW);
  digitalWrite(LED4, LOW);
  digitalWrite(LED5, LOW);
  digitalWrite(buzzer1,LOW); // Buzzer is turned off
}

// This if-else statement is used to generate the alert when a object is far
if ( distance <= 40 && distance > 30)
{
  // The LEDs are lighted up
  digitalWrite(LED4, HIGH);
  digitalWrite(LED5, HIGH);
  digitalWrite(buzzer1,HIGH); // Buzzer is set active
  delay(300); // A delay to get desired beeping frequency
  digitalWrite(buzzer1,LOW); // Buzzer is turned off
  delay(300); // A delay to get desired beeping frequency
}
else
{
  // The LEDs are turned off
  digitalWrite(LED4, LOW);
  digitalWrite(LED5, LOW);
  digitalWrite(buzzer1,LOW); // Buzzer is turned off
}
```

```
// This if-else statement is used to generate the alert when a object is farthest
if ( distance <= 50 && distance > 40)
{
  // The LED is lighted up
  digitalWrite(LED5, HIGH);
  digitalWrite(buzzer1,HIGH); // Buzzer is set active
  delay(400); // A delay to get desired beeping frequency
  digitalWrite(buzzer1,LOW); // Buzzer is turned off
  delay(400); // A delay to get desired beeping frequency
}
else
{
  // The LED is turned off
  digitalWrite(LED5, LOW);
  digitalWrite(buzzer1,LOW); // Buzzer is turned off
}

// These conditions are for ultrasonic sensor 2
// This if-else statement is used to generate the alert when a object is closest
if ( distance2 <= 10){
  digitalWrite(buzzer2,HIGH); // Buzzer is set active
  // The LEDs are lighted up
  digitalWrite(LED6, HIGH);
  digitalWrite(LED7, HIGH);
  digitalWrite(LED8, HIGH);
  digitalWrite(LED9, HIGH);
  digitalWrite(LED10, HIGH);
  delay(50); // A short delay to show constant LED blinking and generate high beeping frequency
  // The LEDs are turned off
  digitalWrite(LED6, LOW);
  digitalWrite(LED7, LOW);
  digitalWrite(LED8, LOW);
  digitalWrite(LED9, LOW);
  digitalWrite(LED10, LOW);
  delay(50); // A short delay to show constant LED blinking and generate high beeping frequency
  // The LEDs are lighted up
  digitalWrite(LED6, HIGH);
  digitalWrite(LED7, HIGH);
  digitalWrite(LED8, HIGH);
  digitalWrite(LED9, HIGH);
  digitalWrite(LED10, HIGH);
}


else
{
  // The LEDs are turned off
  digitalWrite(LED6, LOW);
  digitalWrite(LED7, LOW);
  digitalWrite(LED8, LOW);
  digitalWrite(LED9, LOW);
  digitalWrite(LED10, LOW);
  digitalWrite(buzzer2,LOW); // Buzzer is turned off
}

// This if-else statement is used to generate the alert when a object is a little bit far
if ( distance2 <= 20 && distance2 > 10)
{
  // The LEDs are lighted up
  digitalWrite(LED7, HIGH);
  digitalWrite(LED8, HIGH);
  digitalWrite(LED9, HIGH);
  digitalWrite(LED10, HIGH);
  digitalWrite(buzzer2,HIGH); // Buzzer is set active
  delay(100); // A delay to get desired beeping frequency
  digitalWrite(buzzer2,LOW); // Buzzer is turned off
  delay(100); // A delay to get desired beeping frequency
}
else
{
  // The LEDs are turned off
  digitalWrite(LED7, LOW);
  digitalWrite(LED8, LOW);
  digitalWrite(LED9, LOW);
  digitalWrite(LED10, LOW);
  digitalWrite(buzzer2,LOW); // Buzzer is turned off
}
```

```
// This if-else statement is used to generate the alert when a object is a more far
if ( distance2 <= 30 && distance2 > 20)
{
  // The LEDs are lighted up
  digitalWrite(LED8, HIGH);
  digitalWrite(LED9, HIGH);
  digitalWrite(LED10, HIGH);
  digitalWrite(buzzer2,HIGH); // Buzzer is set active
  delay(200); // A delay to get desired beeping frequency
  digitalWrite(buzzer2,LOW); // Buzzer is turned off
  delay(200); // A delay to get desired beeping frequency
}
else
{
  // The LEDs are turned off
  digitalWrite(LED8, LOW);
  digitalWrite(LED9, LOW);
  digitalWrite(LED10, LOW);
  digitalWrite(buzzer2,LOW); // Buzzer is turned off
}

// This if-else statement is used to generate the alert when a object is far
if ( distance2 <= 40 && distance2 > 30)
{
  // The LEDs are lighted up
  digitalWrite(LED9, HIGH);
  digitalWrite(LED10, HIGH);
  digitalWrite(buzzer2,HIGH); // Buzzer is set active
  delay(300); // A delay to get desired beeping frequency
  digitalWrite(buzzer2,LOW); // Buzzer is turned off
  delay(300); // A delay to get desired beeping frequency
}
else
{
  // The LEDs are turned off
  digitalWrite(LED9, LOW);
  digitalWrite(LED10, LOW);
  digitalWrite(buzzer2,LOW); // Buzzer is turned off
}


// This if-else statement is used to generate the alert when a object is farthest
if ( distance2 <= 50 && distance2 > 40)
{
  // The LED is lighted up
  digitalWrite(LED10, HIGH);
  digitalWrite(buzzer2,HIGH); // Buzzer is set active
  delay(400); // A delay to get desired beeping frequency
  digitalWrite(buzzer2,LOW); // Buzzer is turned off
  delay(400); // A delay to get desired beeping frequency
}
else
{
  // The LED is turned off
  digitalWrite(LED10, LOW);
  digitalWrite(buzzer2,LOW); // Buzzer is turned off
}
}

/*
 *  This is a method which calculates the distance of any object from any ultrasonic sensor.
 *  It takes the trigger port and echo port and do some internal calculations to give the distance.
 */
long measure(int trigger, int echo){

  long dur = 0; // This is used to store the time to travel from ultrasonic sensor to object and back

  // These three lines of codes is used to send a pulse to the object
  digitalWrite(trigger , HIGH);
  delayMicroseconds(2);
  digitalWrite(trigger , LOW);

  // This line receives the pulse back from the object
  dur = pulseIn(echo,HIGH);
  return (dur/2) / 28.5; // This line does some calculation to return the distance of the object

}
```