EECS 1021 Major Project

Kiet Le, 218052787

Video: https://youtu.be/5S1wmfVnGJ8

Introduction:

While trying to come up with a major project, I went through a lot of thinking, trial and error to get to what I now call the Ultrasonic Buzzer (UB). Or what other people call a motion sensor. My major project is a buzzer and LED that activates when an ultrasonic sensor detects a change in distance. The UB sends the distance data to Java to then create a Distance/Time chart.

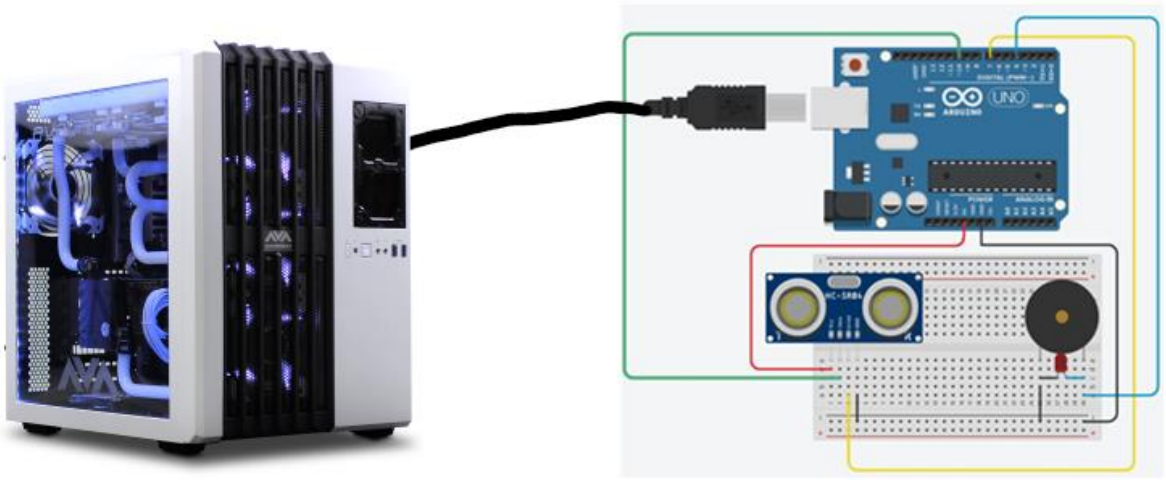
Context:

Motion sensors are used for many different reasons and in many different fields. Your house probably has multiple motion sensors setup for reasons that could include automatic lights and a security system. The ultrasonic sensor is my take on motion sensors using an ultrasonic sensor that will detect if there is a change in distance and then activate actuators of my choosing, which for me are an LED and a buzzer. This data can then be viewed via computer software. You can see the changes in distance over time. This will show you can be extremely useful for remote surveillance.

Technical Requirements/Specifications:

Project abilities:

- Senses change in distance in front of device using ultrasonic sensor.
- Makes a buzzer sound when change in distance is sensed.
- Displays distance data from Java GUI.
- Blinks an LED when activated from GUI.



Component List:

- Arduino Grove Board (1x)
- Ultrasonic sensor HC-SR04 (1x)
- Buzzer (1x)
- Half Breadboard (1x)
- Wires (6x)
- Sick gaming PC (not strictly necessary)

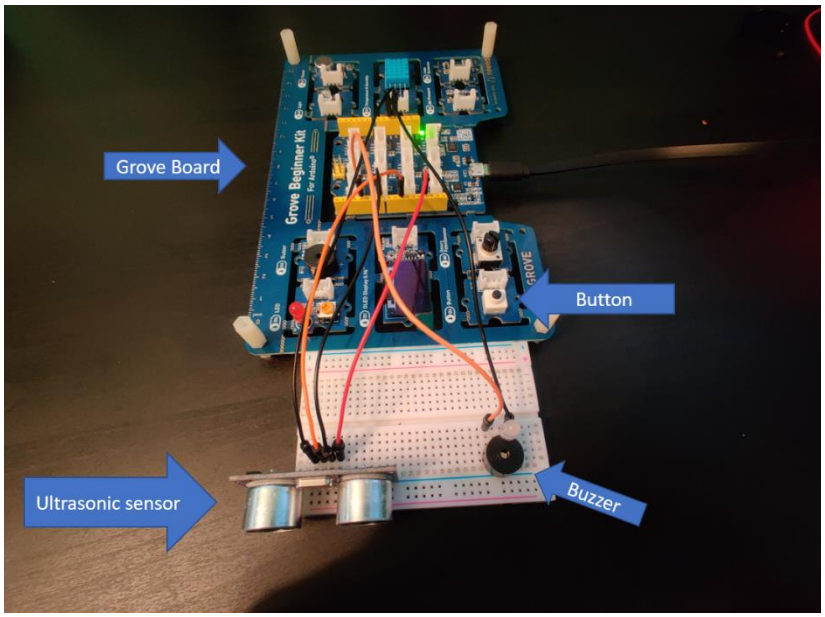


Figure 1: Or is it?

Procedure:

When I was testing out my idea and its feasibility, I did all my initial coding in the Arduino IDE. At first, I was experimenting with the ultrasonic sensor and was learning how to use it. Unfortunately, it was a lot more complicated to code than I expected. There was a bunch of calculations and processes that were involved in getting the sensor to operate. Some of the calculations included using the speed of sound when calculating distance in front of the sensor.

The next obstacle was to somehow get the buzzer to be activated when the ultrasonic senses a change in distance. I did this by adding an if/else statement that sent a signal to the buzzer when there was a substantial change in the distance (either larger to smaller).

Linking the system to Java was not too difficult. I created a method in Arduino IDE that would send the distance data using serial. This data would then be used in my Java program where it would build the graphs using the data. The java program also has the ability to send data to the Arduino to activate the LED.

Test:

My testing method was simple: just wave my hand in front of the sensor and see if the buzzer goes off. The Ultrasonic Buzzer was designed to buzz when sensing a change in distance and stop when the distance in front of it stays constant. So, when I went to test it, it had to complete that criteria. One thing that I found is that the measurements are not very accurate at long ranges. Anything over a meter had various results.

Contingency:

Before I decided on the ultrasonic buzzer as my project, I tried out two other ideas. The first idea was to build a sumobot competition robot. In sumobot, the goal is to push your opponent's robot off the circular arena. The robot is composed of a multitude of wheels, different sensors, batteries, a case, and much more depending on how ambitious you are. When I was pursuing this idea, I programmed the robot to be able to activate its motor and wheels. But I ran into the roadblock of the object sensing. I did not know how to program my robot to sense objects and move towards them. Because of the complexity of the project and my basic knowledge of Arduino, I made the decision to try something simpler. But even though I did not successfully complete a sumobot doesn't mean that I won't try again. One of my goals is to build a fully functional sumobot that can be used in competitions.

The next idea that I tried was to do an automatic streetlight. Basically, the project had a photoresistor and a LED. The photoresistor that would sense how bright the area was. If it was bright, then the LED would stay off. If the area was dark, then the LED would turn on. The code composed of a simple if-else statement. Although the project worked and would have been suitable to use for the major project, I felt as if it was too simple and felt that I could do better. The device also did not meet the requirements of the sensors and actuators that the major project's rubric required.

Additional Materials:

The most basic thing the Ultrasonic Buzzer can be used for is as an electric door chime. But my sensor and code can do more than just activate buzzers (as was displayed in this project). With small modifications it can be implemented into many different situations. Automatic doors can be done by placing the sensor on top of it and changing the code to output a signal to open the door, similar to the current code that activates the buzzer/LED. Security systems can use the sensor to detect movement and activate an alarm. Or it can signal a security camera to start recording when it detects movement. When going back to see the history of the movement, the program can show them how the sensor reacted. The use of my sensor can be implemented in many different ways and benefits society with things that we take for granted and never think about.

Conclusion

I think my major project was successful all things considered. Being able to use my newfound knowledge, I feel like I implemented most of what I've learned into this project.

Video: https://youtu.be/5S1wmfVnGJ8

Works Cited

fazecast. (n.d.). *fazecast*. Retrieved from Github: https://fazecast.github.io/jSerialComm/javadoc/com/fazecast/jSerialComm/SerialPort.html#setComPortTimeouts-int-int-int-

Jabbaar, A. A. (2019, September 17). *Ultrasonic Sensor HC-SR04 with Arduino Tutorial*. Retrieved from Arduino Project Hub: https://create.arduino.cc/projecthub/abdularbi17/ultrasonic-sensor-hc-sr04-with-arduino-tutorial-327ff6

Robinson, R. (2021, March 11). *LabI*. Retrieved from GitHub: https://github.com/rr-codes/LabI