Computer Engineering Culminating Assessment Task Design Report

Luka Spinoti

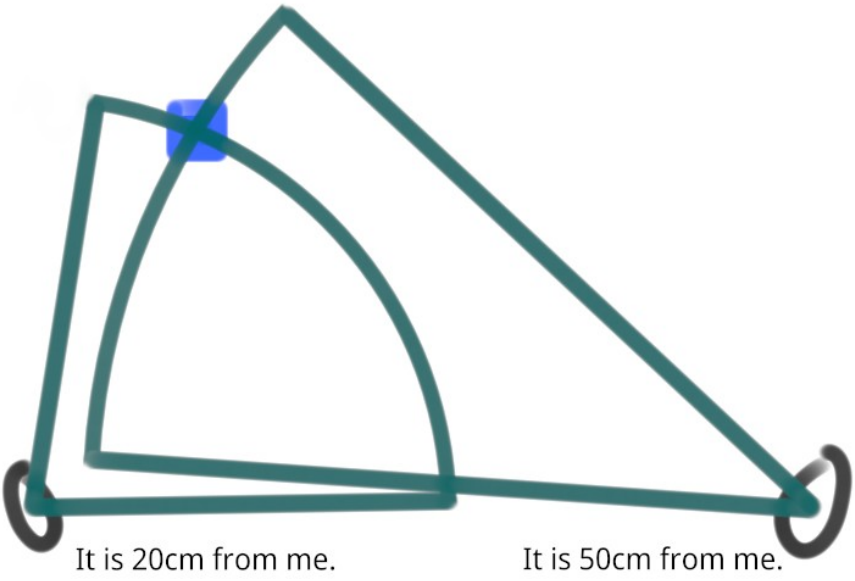
TER 3M1

Mr. Teacher

January 11th, 2023

PART 1

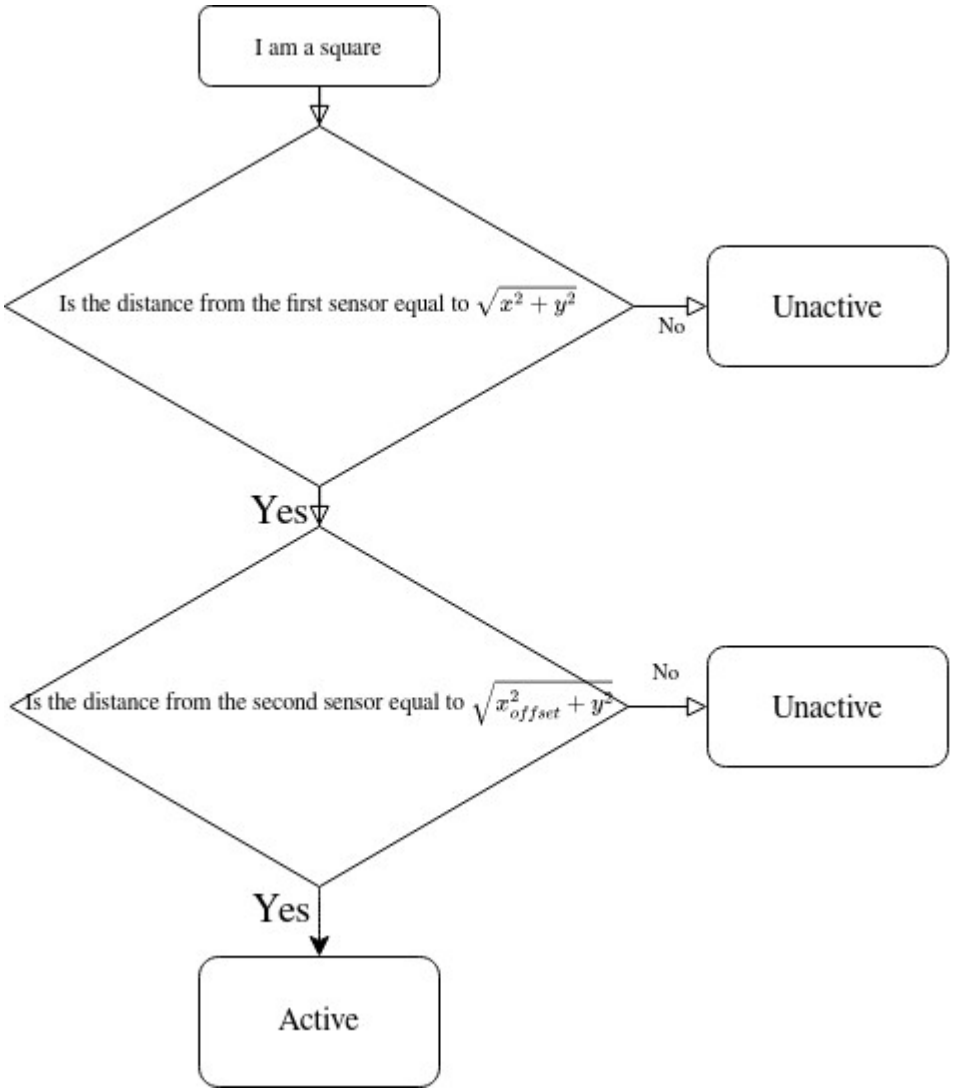
For this course's culminating task, I will be designing, constructing, and testing a positional sensor based on two ultrasonic distance sensors. Such sensor would be able to be used by autonomous vehicles for detecting and avoiding obstacles, which may potentially even save lives. The materials needed are: (1) two ultrasonic sensors, (2) an Arduino, (3) a flat board to be the area the sensors detect on. Position will be detected by having two ultrasonic sensors whose ranges overlap, and every point that is in the distance detected by both sensors will shown on the serial monitor. View the diagram below:



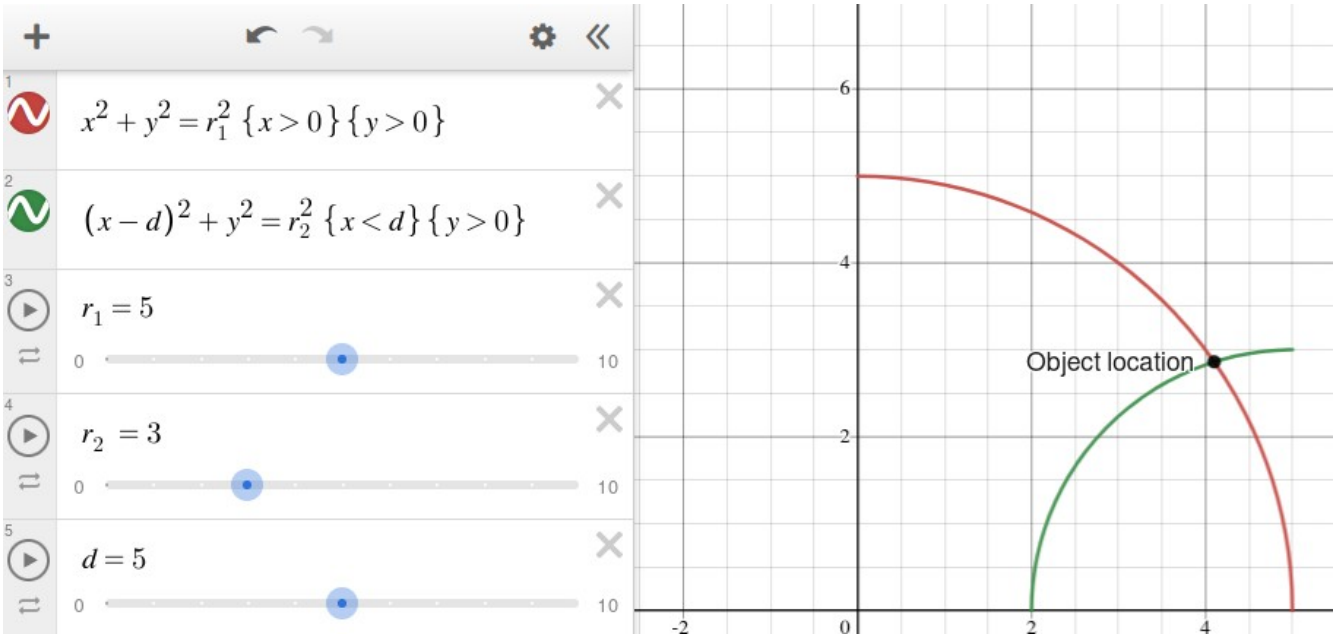
If those two conditions are met, the blue square can only exist in one place (implying the sensors are set up like in the diagram). The serial monitor will output a 10x10 board with squares only active if they satisfy the two conditions mentioned above (I intend to use the Pythagorean theorem / circle equation to find these). I can foresee a few issues I may face in creating this, one being that the sensors must be spaced far apart, meaning I need long cables. I will also have to find the width of the sensors range, which I predict to be tedious.

PART 2

This ultrasonic technology is nothing new, and has been used in the automotive industry for years. Nowadays, it is hard to find a car that does not have circular parking sensors scattered around them. Ultrasonic sensors are also superior to infrared sensors when it comes to detecting transparent objects like glass, as the IR beam will just pass through it. For my project though, I will need two of them to find position in 2D space. The calculation for if a square will be activated is as follows:

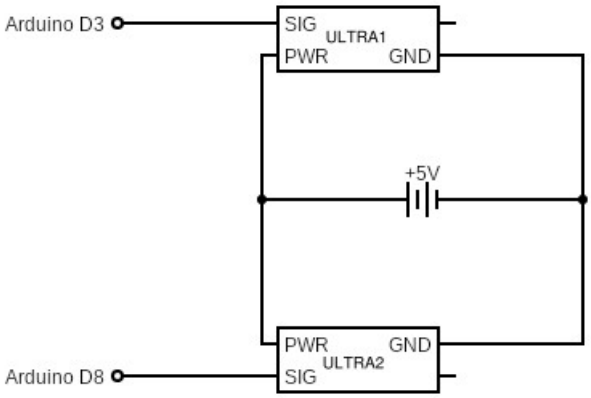


Note the x offset, because the second sensor will act like its coming from the top right instead of the top left. See another diagram:



PART 3

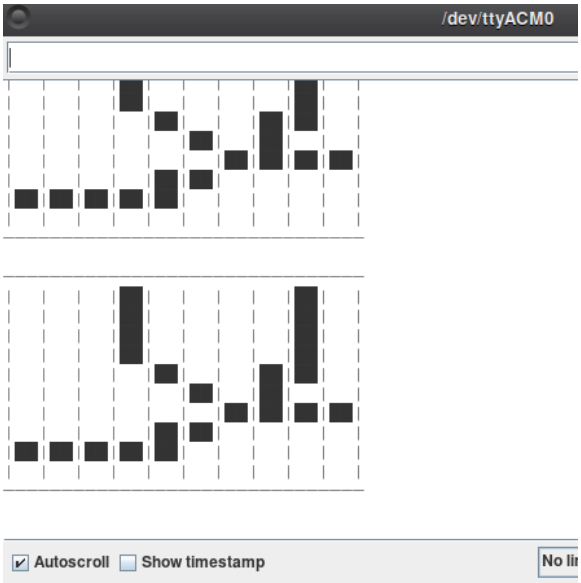
Now that I have decided on my solution, I will begin the next part of this project. I have already written down the parts I need, but I forgot to include the breadboards and my laptop, both of which are critical to my project's success. I plan to first write the Arduino code for three days (Wednesday, Thursday, and Friday), and then to construct it in three days (Monday, Tuesday, and Wednesday), which will give me an extra two days in case I cannot get it working on Wednesday. The circuit schematic seems very simple, but in real life, the sensors will be positioned far apart from each other, which may cause the cables to look cluttered.



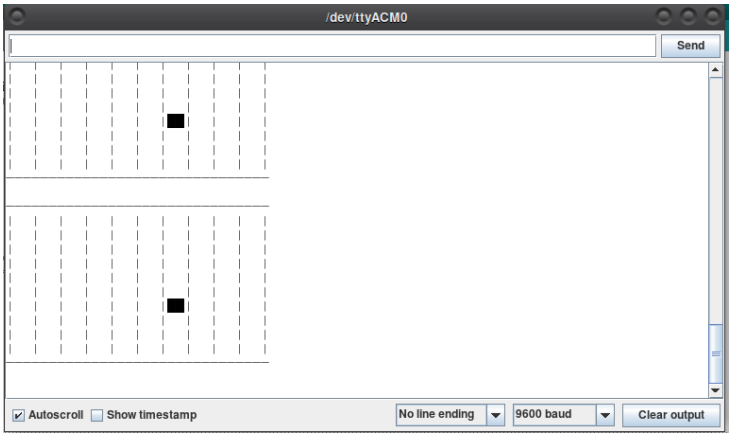
PART 4

To code the project, I used the Arduino IDE. I then defined the constants I would need in the program (the board size, the pin numbers, etc). Since I wanted the whole process to be repeated, I put everything I wanted wanted to run in the **loop** function, except for the **Serial.begin** command. I made the function **printBoard** to print the board first because I knew it would be simpler. It simply loops through a 2D boolean array and prints the squares black if they are true and white otherwise. The next function, **detectSquare**, uses the Pythagorean theorem to check if a square contains the object. I also made a function named **detectSquareDebug** to set the squares that are found by either circle (OR instead of AND).

Output when using **detectSquareDebug**



Output when using **detectSquare**



After all of the code was written, I started to build it. The first thing I did was set up the first ultrasonic sensor in a mini breadboard near the Arduino. I used my hand to determine the range and width of the sensor to ensure that no other objects outside of the board would be detected. Next, I cut long wires to be able to connect the sensor that was to be positioned far from the rest of the apparatus. I connected the wires to the second sensor through another mini breadboard, and calibrated it as well.

PART 5

One of the largest problems I faced during the development of this project was that only the squares on the edges of the board would be detected. Initially, I thought it was the calculation that was wrong, because it looked to complicated to work perfectly the first time. It turns out that when comparing the distance of the ping to the position on the board, I was comparing a float value (which 99% of the time was going to be a value with a decimal) to an integer (no decimal places), so it would only detect the sides of the board where the values are exact. I fixed this by casting the distance to be an integer, so the values were much more likely to be the same. The first time I built the apparatus, it worked perfectly. The sensor's cone width was narrower than I had imagined, but it nonetheless functioned well. I could not get the same level of precision as the first time for every time after that. No matter how well I calibrated it, I couldn't get a single box to display while using **detectSquare**. For my presentation, I will probably have to use **detectSquareDebug** and show both circles instead of just the intersection. I didn't have to make any major changes to the concept during the building process.