ELECTRONICS LAB REPORT
2D Topology using Ultrasonics

Amey Gaikwad : 15D260002
Rahul Dandwate : 15D260006

November 9 2017

Abstract

An Arduino microprocessor can be used for a variety of purposes. In this project we have tried to implement a 2D mapping device. We have programmed the Arduino to power up a servo motor on which is mounted an ultrasonic sensor. As the servo motor moves through an angle of 180 degrees, the ultrasonic sensor measures the distance. 180 readings are recorded by the ultrasonic sensor over half a rotation. The Arduino code is synced with a python file which takes this data and plots the 2D demography. The ultrasonic sensor has its own resolution issues, however efforts have been made to minimise stray errors as much as possible.

Contents

1 Introduction

The use of ultrasonics in calculating the distance and mapping out the 2D demography is not new. Radars and Sonars have already shown the numerous applications of using sound for mapping purposes. Even in the real world, we see dolphins and bats using ultrasonics for mapping out the environment.

The principle used is echo location. The sound signal is sent out by the ultrasonic sensor. It hits the object whose distance is to be measured. Suppose the velocity of sound in dry air is given by v and the distance to the object is d. Then the time t required to come back to the sensor is given by:

$$t = \frac{2d}{v} \tag{1.1}$$

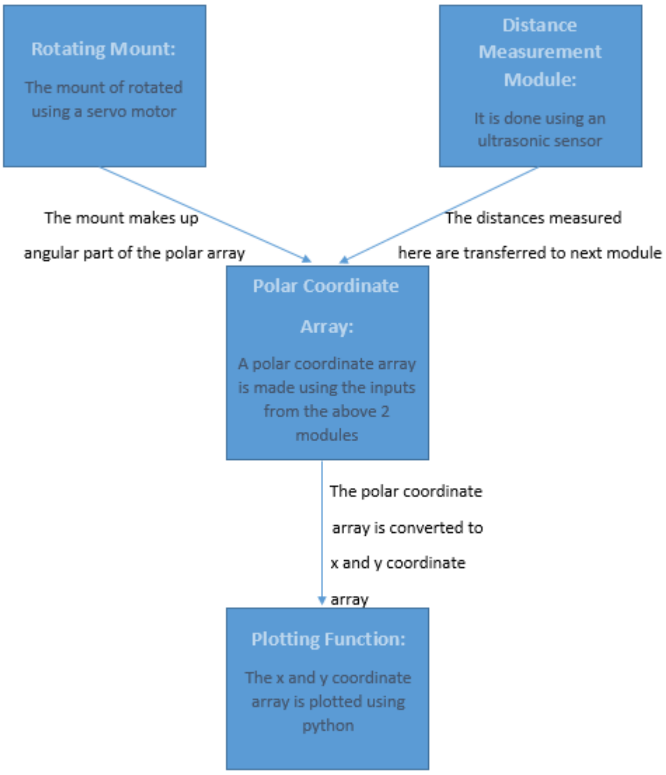
$$d = \frac{vt}{2} \tag{1.2}$$

This is the basic principle of echo location. By finding out the time required to come back to the sensor, the distance to the object can be calculated by (1.2). This is precisely the principle we are going to exploit to map out the 2 D surroundings.

In this 5 weeks long project, we have modelled a simplistic two dimensional mapping device. An ultrasonic sensor is mounted on a servo motor. The mounting mechanism used is also kept simplistic : a double sided tape.The Arduino powers the servo motor.The servo has been programmed to perform just one half rotation. While rotating, the ultrasonic sensor takes 180 readings, one per degree. This data is then sent to a python code which is synced with the Arduino code. The data points obtained are radial and angular. These are then converted to Cartesian arrays. The X and the Y data points obtained are then plotted.

The project is a very crude version of a 2D mapping device. It comes with its own set of limitations, but we believe that this project can be further improvised upon in the future and it can be used as a prototype for a more efficient mapping device

2 Block Diagram


Rotating Mount:
The mount of rotated using a servo motor
Distance Measurement Module:
It is done using an ultrasonic sensor
The mount makes up angular part of the polar array
The distances measured here are transferred to next module
Polar Coordinate Array:
A polar coordinate array is made using the inputs from the above 2 modules
The polar coordinate array is converted to x and y coordinate array
Plotting Function:
The x and y coordinate array is plotted using python

3 The Modules and Implementation

The project was divided into three modules modules which were implemented independently. They were sewed together in three different stages and test runs were carried out after each module was completed. The description of individual modules is given below:

3.1 Distance measurement module

Ultrasonic sensor was used to measure distance. The sensor can measure distances from 2 cm to 4 m. The distances are measured by echo detection. The sensor sends a ultrasonic pulse and measure the time for the pulse to return, the distances are measured using the speed of sound in air. The main equation governing this is given by (1.2).

3.2 Rotating mount

The ultrasonic sensor was mounted on to a rotating mount. The rotation was controlled by a servo motor, which has a geared mechanism which precisely controls the angle of rotation of the sensor.

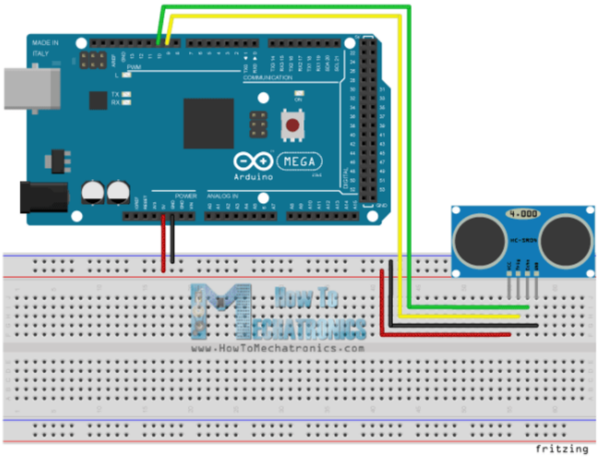
The servo motor rotates by angle of 1° in each step. The distances are measured at each step. These measured distance are then passed from Arduino to the communication port of the laptop, from where they are stored in an array using python. Syncing this to the python file needs the implementation of PySerial to be done properly.

3.3 Plotting function

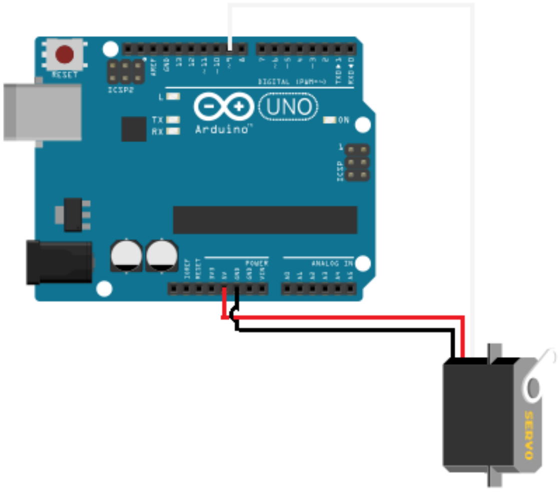
The array of distances measured for each step is now converted to array of x and y coordinates. These x and y coordinates of obstacles placed around the sensor are now plotted using python. Some stray values are also picked up by the sensor. These may lead to the graph looking skewed.

4 Circuit Diagram

- Circuit for measuring the distances using the ultrasonic sensor



- Circuit for controlling the servo motor



5 Testing and Verification

Various trial runs were carried out at regular stages of the project in order to check if everything was in order upto the penultimate stage.

- After setting up the ultrasonic sensor. The sensor was exposed to different environments. This was done to check if the data collected by the sensor was accurate.
- After setting up the servo motor. In order to check the working of the servo motor, various codes were uploaded.
- Before mounting the sensor on the motor, the combined code was checked for consistency. Again the sensor was exposed to different environments to see if in the combined code too, it mapped the distances accurately.
- Final test runs were carried out after attaching the sensor to the motor. Efforts were made to eliminate the limitations as much as possible. More complex environments were introduced.

6 Work Distribution

Amey

- Transfer the data obtained from the Arduino code to the Python file for plotting the data points.
- Perfecting the mount.
- Sync the Python and the Arduino codes.
- Ready the angular and the distance arrays.
- Plot the final data points

Rahul

- Setup the ultrasonic sensor
- Setup the Servo motor.
- Combine the codes for the servo motor and the ultrasonic sensor.
- Run trial runs.
- Try to cut down on the limitations of the equipment.

7 Weekly Distribution

Week 1

- Ready the code for the ultrasonic sensor.
- Sync the ultrasonic sensor with the Arduino code.

Week 2

- Run trial runs to check whether the ultrasonic sensor is picking up data in a consistent manner.
- Ready the arrays in Python for storing the radial distances calculated by the sensor.

Week 3

- Set up the servo motor.
- Makes sure that the speed of the motor is properly set. It should not be too fast.

Week 4

- Prepare the mounting device for the sensor on the servo motor.
- Makes sure that the combined code runs without glitches.
- Prepare the angle and the radial distance arrays
- Plot the graph.

Week 5

- Try to improve upon the mounting mechanism.
- Run trial runs to check whether the graph obtained conforms to the environment or demography being mapped

8 Challenges along the way

- Syncing python and Arduino files was initially found to be tricky. However after having learnt how to implement PySerial, the job became easier.

- Problems occur when perfect right angles are encountered and the readings seem scuppered. This is mainly due to the resolution limitation of the ultrasonic sensor.

- The speed of the motor has to be optimum. It has to be much lesser than the speed of sound, so that the distances are read correctly. This arises due to the finite speed of sound and the resolution errors of the ultrasonic sensor.

- Initial attempts were made to use a stepper motor instead of a servo motor. However, the current limitations of the hair-bridge capacitor led to the replacement of the stepper with the servo. While the stepper did provide a 360 degree rotation, the hair-bridge motor had to be used along with it.

- The syncing of the python and the Arduino files led to many teething problems. The motor used to behave in an erratic fashion sometimes. After a few trial runs, we managed to find the way to run both of them together.

9 Limitations

- The resolution of the ultrasonic sensor is a problem. While mapping the corners, it can lead to some pretty strange readings. It offers a measuring angle of 15 degree. This can lead to some stray values and errors, which can throw the graph into chaos. After having filtered out these values, we get a rough estimate of how the demography is mapped.

- The wires attached to the ends of the ultrasonic sensor move the alignment of these sensor which is attached to the roaring motor. So as the motor rotates the wires somehow constrain the motion, which leads to a slight tilt in the frame of the ultrasonic sensor. This leads to stray values.

- The COM4 port to which the values are being read is sometimes not properly picked up by Python. Hence the Python code may start reading the values from the port even before the ultrasonic sensor starts reading the values.

- The distance sensitivity of the ultrasonic sensor is 2cm to 4m, so distances outside this range can lead to errors in readings. Also due to the instability of the ultrasonic sensor, distances outside this range can be encountered leading to errors in the values being read.

- While efforts have been made to keep the servo motor as stable as possible during operation, random fluctuations can lead to a tilt in the sensor's frame leading to stray values. This limitation arises due to the crude mounting device that is being used. While a double sided tape does a good job of holding everything together, vibrations of the motor can lead to instability in the sensor's frame leading to wrong readings.

10 Conclusion

We have built a basic prototype of a 2 D mapping device. This can be modified and used for may interesting applications in the future. The almost immediate step would be to increase the number of dimensions which it can map. A 3 Dimensional mapping device would truly replicate a radar or a sonar. Albeit the fact that the mounting device would have to be greatly modified, it is still something that can be thought about It has interesting applications in robotics. It can be used as the sensing paraphernalia of the robot. In surveillance cameras, too, for a quick scan of the surroundings, this prototype can be perfected to map out the dangers quickly. The concept can be further generalised to mapping out heat signatures in the nearby surroundings.

While still being just a prototype, it packs into it the basics that are needed for the aforementioned applications. It does come with a fair share of limitations, but the hope is that they could be overcome by a more sturdier construction and a much more sensitive ultrasonic sensor. We hope that you will have as much fun in exploring our 2D mapping device as much as we have had in making it !!

11 Acknowledgements

We would like to thank our Prof. Pradeep Sarin for giving us an opportunity to do this project. We also want to thank Mr. Nitin Pawar and other TAs in the electronics lab for helping us along the way. Their suggestions went a long way in helping us to do this project in the most efficient way possible.

12 Bibliography

- Tutorial for ultrasonic sensor
- Tutorial for servo motor
- Tutorial for reading data from the communication port for collecting data from the Arduino board

13 Link for video demo

The following is the link to video demo: Video demo
The following was the output of the trial run: Output