

“Door unlocking using Face Detection”

Aishwary Jagetia
adjagetia@wpi.edu

1. Summary of Accomplishments:

1.1. Did you complete all of the basic requirements?

1.1.1. Download latest Arduino IDE from arduino.cc.

I have downloaded ARDUINO 1.8.5 which is open-source software, compatible with my Linux as well as Windows operating system.

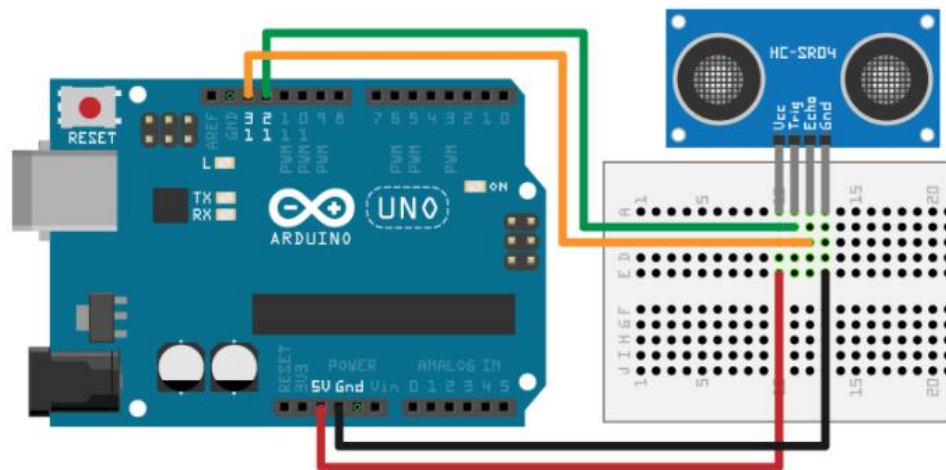
1.1.2. Learn Arduino C/C++ and its IDE.

We have used Arduino C in serial communication and standard libraries for experimentation.

1.1.3. Learn about interfacing with a variety of sensors. (Analog, PWM, Interrupts, SPI, I2C).

We have interfaced the Ultrasonic sensor with analog and digital interrupts as well as with I2C for interrupts communication.

1.1.4. Write a simple sketch to get an Arduino to interface with a sensor.



1.1.5. Show that Arduino can properly handle interrupts.

When it is not using serial communication it is working properly, while communicating the serially the interrupts are messing up and not behaving as intended.

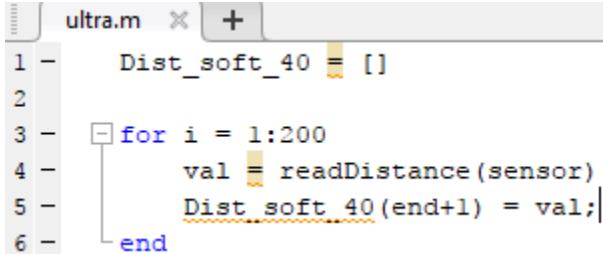
1.1.6. Get Arduino to communicate with a laptop using the Arduino serial class.

I have also used MATLAB library JRodrigoTech/HCSR04 to set up communication between Arduino with the laptop. Using a simple command as follows I have collected the data :

```
sensor = addon(a, 'JRodrigoTech/HCSR04', 'D12', 'D13')
val = readDistance(sensor);

sprintf('Current distance is %.4f meters\n', val)
```

The MATLAB Code for this is as follows:



```
1 -   Dist_soft_40 = []
2 -
3 -   for i = 1:200
4 -       val = readDistance(sensor)
5 -       Dist_soft_40(end+1) = val;
6 -   end
```

1.1.7. Get Arduino to send ROS messages of sensor outputs using Arduino ros-serial library.

Following the official instruction given in <http://wiki.ros.org/> we used ros_lib in order to connect with Arduino in order to send ROS messages of sensor outputs.

1.1.8. Learn Ubuntu (Linux) and its tools.

I have learned to operate Ubuntu operating system, and developed my skills in Command Prompt and perform tasks using it. I have installed many developer tools including: IDE Python, Arduino IDE etc. with a Simulation environment “Anaconda” using which I have learnt to run coding using Tensorflow and Keras.

1.1.9. Learn the basics of Python.

I have opted for a basic course from codecademy (www.codecademy.com) to learn the basics of python. During this course I have earned 19 badges so far. I am not coding in python using atom interface.

1.1.10. Learn the basics of the Robot Operating System (ROS).

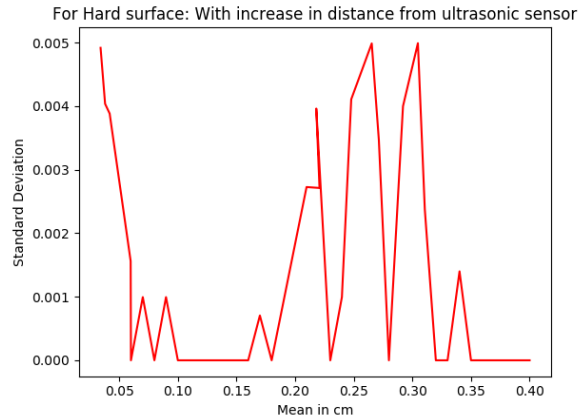
Using <http://wiki.ros.org/> the documentation and tutorials available for learning ROS as well as referred to the book “A Gentle Introduction to ROS” by Jason M. O’Kane gave me a strong foundation to ROS basics.

1.1.11. Install Ubuntu (14.04 works) and ROS (Indigo) on Laptop

I have installed Ubuntu 16.04.3 on sides with my Windows Operating system. I have also installed ROS (Indigo) on my Ubuntu and tested it with turtle game.

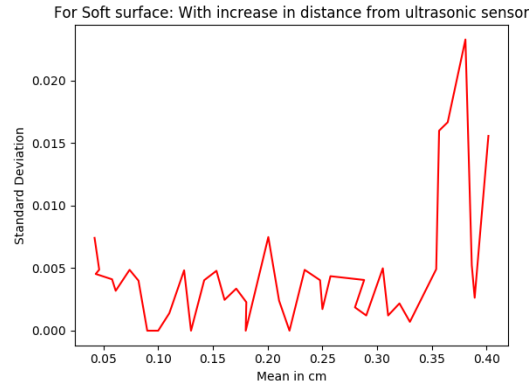
- 1.1.12. [Have ROS store sensor data for later analysis. Probably use ROS “bags”.](#)
We have not used ROS “bag” instead we have implemented the node system, because we are implementing the network in real time not from the stored data.
- 1.1.13. [Analyze the data using Python.](#)

```
Sensor Analysis_Hard surface.py | Sensor Analysis_Soft surface.py
1  """
2  Created on Wed Nov 29 17:27:25 2017
3
4  @author: Aishwary
5  """
6  import numpy as np
7  import glob
8  import pandas
9  import math
10 import os
11 import matplotlib.pyplot as plt
12
13 def sortKeyFunc(s):
14     return int(os.path.basename(s)[-4])
15
16 mean = []
17 std = []
18 files = sorted(glob.glob('Hard_surface\*.csv'), key=sortKeyFunc)
19
20 for i in files:
21     dataframe = pandas.read_csv(i, header=None)
22     dataset = dataframe.values
23     a_list = np.reshape(dataset, (200,1))
24     i = 0
25     time1 = []
26     for i in range(0,len(a_list)):
27         time1.append(i/10)
28         mean.append(np.mean(a_list))
29         std.append(np.std(a_list))
30
31 plt.plot(mean,std, color='red')
32 plt.title('For Hard surface: With increase in distance from ultrasonic sensor')
33 plt.xlabel('Mean in cm')
34 plt.ylabel('Standard Deviation')
35 plt.show()
```



```

Sensor Analysis_Hard surface.py  x  Sensor Analysis_Soft surface.py
1  """
2  Created on Wed Nov 29 17:27:25 2017
3
4  @author: Aishwary
5  """
6  import numpy as np
7  import glob
8  import pandas
9  import math
10 import os
11 import matplotlib.pyplot as plt
12
13 def sortKeyFunc(s):
14     return int(os.path.basename(s)[-4])
15
16 mean = []
17 std = []
18 files = sorted(glob.glob('Soft_surface\*.csv'), key=sortKeyFunc)
19
20 for i in files:
21     dataframe = pandas.read_csv(i, header=None)
22     dataset = dataframe.values
23     a_list = np.reshape(dataset, (200,1))
24     i = 0
25     time1 = []
26     for i in range(0,len(a_list)):
27         time1.append(i/10)
28         mean.append(np.mean(a_list))
29         std.append(np.std(a_list))
30
31 plt.plot(mean,std, color='red')
32 plt.title('For Hard surface: With increase in distance from ultrasonic sensor')
33 plt.xlabel('Mean in cm')
34 plt.ylabel('Standard Deviation')
35 plt.show()
  
```



We have collected the data at every cm distance from the ultrasonic sensor taking 200 reading at every position. We have used hard as well as soft surface and examined our sensor reading at different angles.

Based on these observations we found that the threshold distance from the ultrasonic sensor is less than 40cm, which we have used in our project. We also, found that the sensor is sensitive only for 10-degree variation of the surface.

1.1.14. Understand the characteristics of sensors. Don't just connect them up.

Document them:

1.1.14.1. Their interface characteristics and code.

Interface characteristics:

- Power Supply: +5V DC
- Quiescent Current: <2mA
- Working Current: 15mA
- Effectual Angle: <15°
- Ranging Distance: 2cm – 400 cm/1" – 13ft
- Resolution: 0.3 cm
- Measuring Angle: 30 degrees
- Trigger Input Pulse width: 10uS

1.1.14.2. Their error characteristics and statistics.

The specifications of the HC-SR04 ultrasonic sensors are:

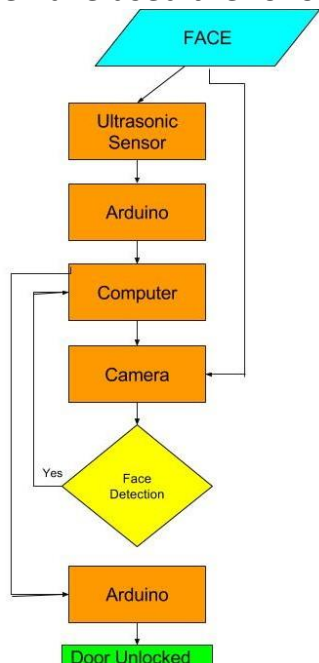
- FOV (full cone): horizontal ~21°, vertical ~4°
- Spatial resolution (full cone): ~0.6-1.4°
- Range: tested from 5 to 200 cm
- Accuracy: absolute error ~0.035 cm/cm.
- Precision: standard deviation ~0.1-0.5 cm

1.1.15. Keep notes as you progress.

We have notes for Sensor analysis data, Filter data and Output log data as reflected in the presentation.

1.2. Did you accomplish other objectives?

Our project objective was to successfully implement a door unlocking system with face detection capability. For which we have used the following approach:



We have successfully implemented this and accomplished our objective. This includes to work on sensor data analysis to find the threshold distance, to establish communication using ROS interface and to successfully implement Open-cv for face detection.

2. Problems Encountered:

During the project we encountered some problems as listed below.

- Interrupts break ros_serial
- Filtering added a large delay
- Filter was initially erratic
- OpenCV and ROS integration difficult
- Program flow different, mixing the two was interesting
- Sensor Range limited
- Trouble maintaining ROS connection



3. Solutions to (or work-arounds for) problems encountered:

During the project we encountered some problems as listed below.

- We removed the interrupts from the code which made everything work perfectly.
- For Filtering the program collects the data using the ROS communication. So, the delay was due to consideration of multiple reading being taken/collected before processing. So, reducing the number of readings to be collected we can reduce the delay.
- Using while loop in OpenCV and simultaneously using `ros::isOk()` loop was solved.
- Sensor Range limited which was solved by using python instead of MATLAB with different libraries.