Title
Third Eye: Assistive Technology for the Visually Impaired

Author
Qazi, Tasmiyah

Publication Date
2019-07-01

THIRD EYE
(ASSISTIVE TECHNOLOGY FOR THE VISUALLY IMPAIRED)

By

Tasmiyah S Qazi

A capstone project submitted for
Graduation with University Honors

June 6, 2019

University Honors
University of California, Riverside

APPROVED

____

Dr. Richard Cardullo
Department of Biology

____

Dr. Richard Cardullo, Howard H Hays Jr. Chair, University Honors

Abstract

Third Eye project is a wearable aid device for assisting anyone with visual impairments. It contains multiple sonar sensors to track approaching objects and a camera that identifies faces according to a user-entered database. Facial detection is used in order to store people in a contact list and inform the user who is approaching. The intent is for the visually impaired individual to know who they are talking to or who is approaching them. This concept can eventually be expanded upon to recognize strangers through reverse image-search for future prototypes. These devices will communicate about who/what is approaching with the user through separate speakers. For example: “Bob is approaching you”, “obstacle 5 feet ahead”.

The system is embedded within a box as a proof of concept. There are two speakers that the user listens to for sound cues with important directional and facial information.

This project is inspired by helping those who are disabled. Visual impairment is one disability that **may** be troublesome and can potentially affect quality of life. If this project can make possible problems a little more avoidable like bumping into objects or not knowing who is approaching you to talk, it would be beneficial for the user. The project's benefit is not just for the blind and would allow for numerous possibilities to expand in the future such as other disabilities.

In terms of technicalities, Third Eye will be using an Arduino Uno for the distance sensor functionalities and sound as well as a Raspberry Pi for the facial detection and sound. The project outcome included: the two microcontrollers to run concurrently, the Raspberry Pi to be able to initiate a sound cue using a Python library (OpenCV) when the camera detected a familiar face and the Arduino to be able to initiate a sound cue when an obstacle is detected within the vicinity using sonar sensors.

Table of Contents

Document Revision History

Revision Number	Revision Date	Description	Rationale
0	6/6/19	Document creation	----------

Definitions, Acronyms, and Abbreviations

Acronyms/Abbreviations:

- GB: Gigabyte
- OS: Operating System
- SD Card: Secure Digital Memory Card
- PWM: Pulse Width Modulation
- SPI: Serial Peripheral Interface

Definitions:

- SPI: is a synchronous serial communication interface specification used for short-distance communication. Common applications are for communicating with SD Cards.
- Ultrasonic wave: Ultrasonic waves are sound waves transmitted above the human-detectable frequency range, usually above 20,000 Hz.
- Sonar: A technique that uses sound propagation to navigate, communicate with or detect objects on or under the surface of the water, such as other vessels.
- Raspbian: A Debian-based computer operating system for Raspberry Pi.
- Debian: A Unix-like operating system consisting entirely of free software.
- PWM: A technique for getting analog results with digital means.

Assumptions and Dependencies

We are assuming the user will be of average height in order for collision and face detection accuracy. We are also assuming that they will not wear the device in excessively wet or hot conditions.

Even though we have changed the mock to a box, we still need to base it on an average height. The box also assumes the user's waistline is small enough for the 4 sonars to effectively cover the user's sides and front/back.

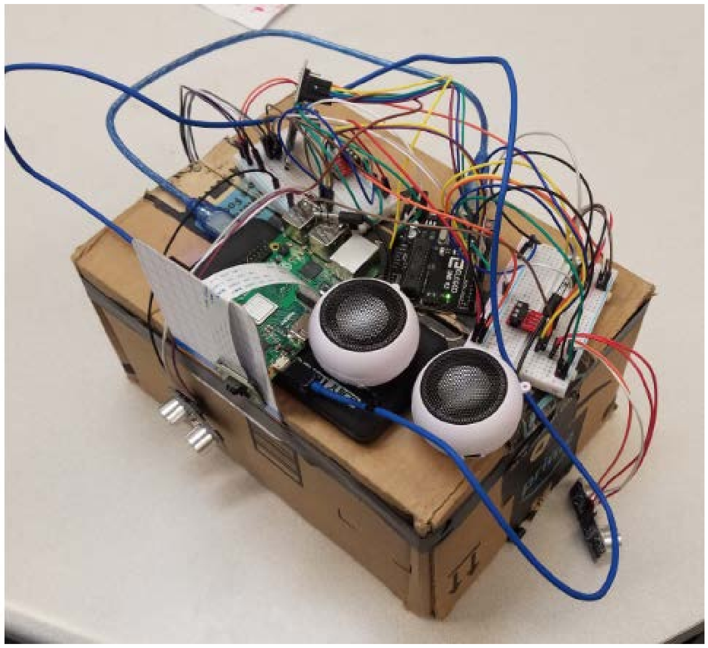
Our system does not depend on any external systems as everything is supported and run via our Raspberry Pi and Arduino Uno.

Project Description

Problem Statement:

The inspiration for designing this project comes from the idea of assisting individuals with visual impairments. There are two main problems we are trying to solve: object detection and recognizing familiar faces. We would like to help these individuals know the direction of surrounding objects that are approaching them. We would also want the user to be able to know who is approaching them. The end goal of Third Eye is to provide visually impaired individuals with the ability to navigate more independently.

System Purpose:



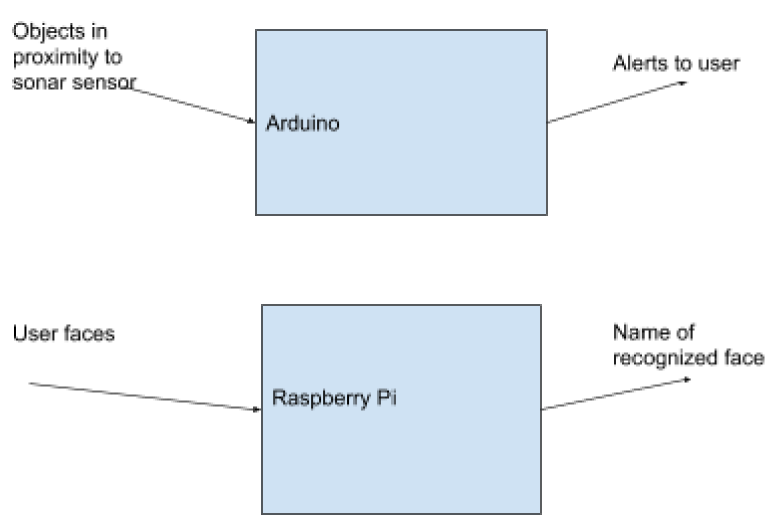
Third Eye is a system that assists people with impaired vision by using obstacle detection, facial detection and sound cues. The Third Eye prototype is embedded in a box representing a mock human.

The first use case for Third Eye is to alarm the user of the direction of any incoming objects that are either approaching or being approached by the user. Third Eye is equipped with four sonar sensors connected to an Arduino Uno that detect four directions: front, behind, left, and right. There is a speaker embedded on the system that provides the sound cue for the user. All of these components are currently embedded on a box that emulates a wearable device for the user.

As the user approaches surrounding objects, the sonar sensors will detect the direction to alarm the user based on the distance detected. For example, as the user approaches a wall in front of them within 150 centimeters, Third Eye will then initiate a sound cue that says "front" to the user. If the user continues to approach this wall, Third Eye will once again send another sound cue saying "front". If the user turns left, Third Eye will then indicate to the user that this wall is now on their right by saying "right". This benefits the user because it now expands the radius in which a visually impaired individual can detect objects from and allows the user to have enough time avoid the object if necessary. If the user is still, there is no sound cue.

The second use case of Third Eye is to alarm the user of the familiar faces in front of them. This subsystem is also embedded on the box and emulates the user's vision. This subsystem uses a camera to determine the recognized face and a speaker to say the name of the person. As the user approaches a familiar person named "Richard," The system will then initiate a sound cue that says "Richard". This would also work as the user approaches "Richard.". This benefits the user because it will allow them to know the name of the person they are having a conversation with. There is no sound cue if the face is unrecognized or if the last face Third Eye recognized is the same.

System Context:



The components connected to the Arduino is one subsystem. The output of the Arduino are the alerts to the user. The input of the Arduino are potential obstacles in the environment. The components connected to the Raspberry Pi is another subsystem. The input of the Raspberry Pi are potential human faces.

Our system will be used in indoor and outdoor environments. It is useful anywhere the user needs to navigate or recognize faces. However, the lighting must be relatively similar to the lighting used in the contact list images in order for facial detection to work accurately.

User Characteristics:

The users will be those who are visually impaired, specifically those who have defects in the ability to see or born without/with limited sight. It can also be used for anyone who would like to use the system's ability to detect obstacles or faces.

This system can be used in almost any location or climate. The only constraints are that the system cannot be used in wet or excessively humid environments. The system can also not detect faces in darkness.

The user's behavior and use of the system should specifically be for its intended use of aiding everyday walking in already familiar surroundings.

Operational Scenarios**:**

Bob is a 25 year old who contracted a disease that affects his ability to see. He is in need of a device that can help him adjust to his impairment.

Bob is given a Third Eye to help adjust to his impairment. Bob holds the device in front of him and begins to go about his day. As he walks, the device alarms him of approaching objects and of familiar faces. The Third Eye alarms Bob of incoming objects such as when he approaches the door to get outside of his house. The Third Eye will alarm Bob through a speaker and will tell him “front” to warn him.

The Third Eye also informs Bob of familiar people stored in its contacts list. As Bob’s friend Richard approaches him, the Third Eye notices him and says “Richard”.

Bill of Materials (BoM):

Part #	**Description**	**Cost per**	**Quantity**	**Total**
EL-SM-001	Ultrasonic Module Distance Sensor for Arduino UNO	$7.99	2	$15.98
B0033	Pi 3 camera for facial/image recognition	$13.44	1	$13.44
AU-EL-CB-001	Arduino to use for the sonar	$11.86	1	$11.86
AE1223	Gikfun TRRS 3.5mm Jack Breakout Headphone Video Audio MP3 Jack for Arduino	$2.53	1	$7.58
Kingston SDC4/8GBET	2, 8GB micro sd cards with adapter	$3.99	2	$7.98
SDREADWRITE_BO	SD Card Reader for microcontrollers	$6.59	1	$6.59
299006	Electrical tape	$2.86	1	$2.86

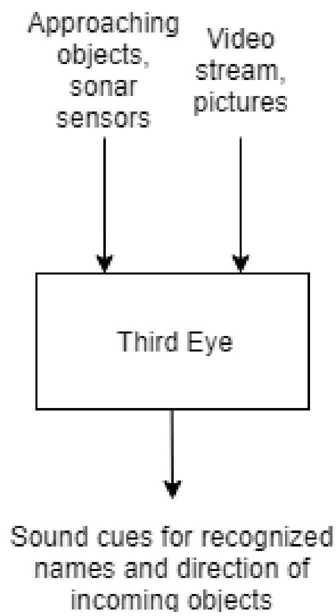
B072BNLC1M	Hamburger Mini Speaker	$5.95	2	$16.90
LYSB01IMNVZDC-ELECTRNCS	4 small breadboards	$8.49	1	$8.49
ALLEU 4330582961	jumper wires M/M, M/F, F/F	$6.49	1	$6.49
RPI3B+HS	Raspberry Pi 3 Model B+ used for facial recognition software	$37.75	1	$37.75
				Total: $135.92

Engineering Standards:

Description	**Standard**	**Governing body**
JPEG	ISO/IEC 10918	ISO/IEC
3.5mm audio jack	P.382	ITU-T
BGR Color Scheme	OpenCV	Intel Corporation
MP4	MPEG-4 14	ISO/IEC
Facial Recognition	OpenCV	Intel Corporation
Storing Raspbian on the Raspberry Pi, and storing WAV files on the Arduino	SD Card	SD Card Association(SDA)

Serial communication with the SD card	SPI	SPI bus is not governed by any standard organization
Audio file being outputted by the Arduino	WAV	Microsoft and IBM

System Overview

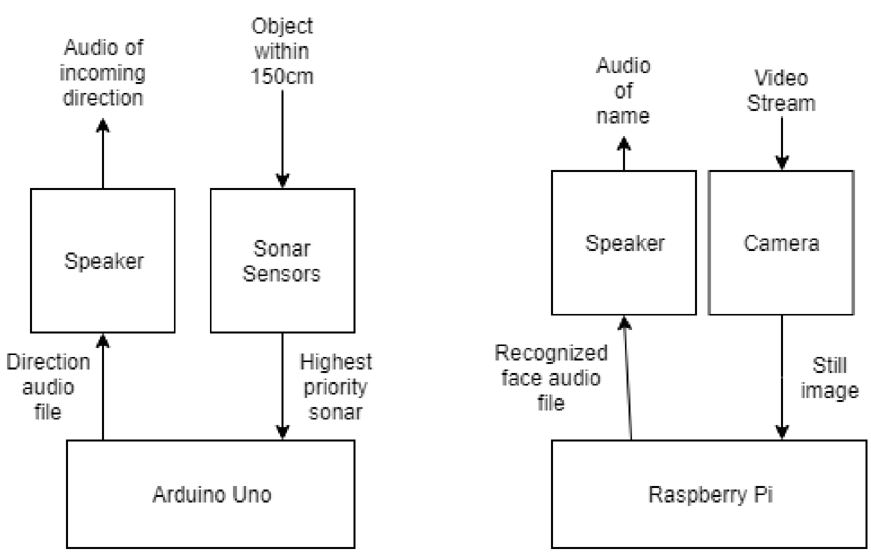


Input/Output:

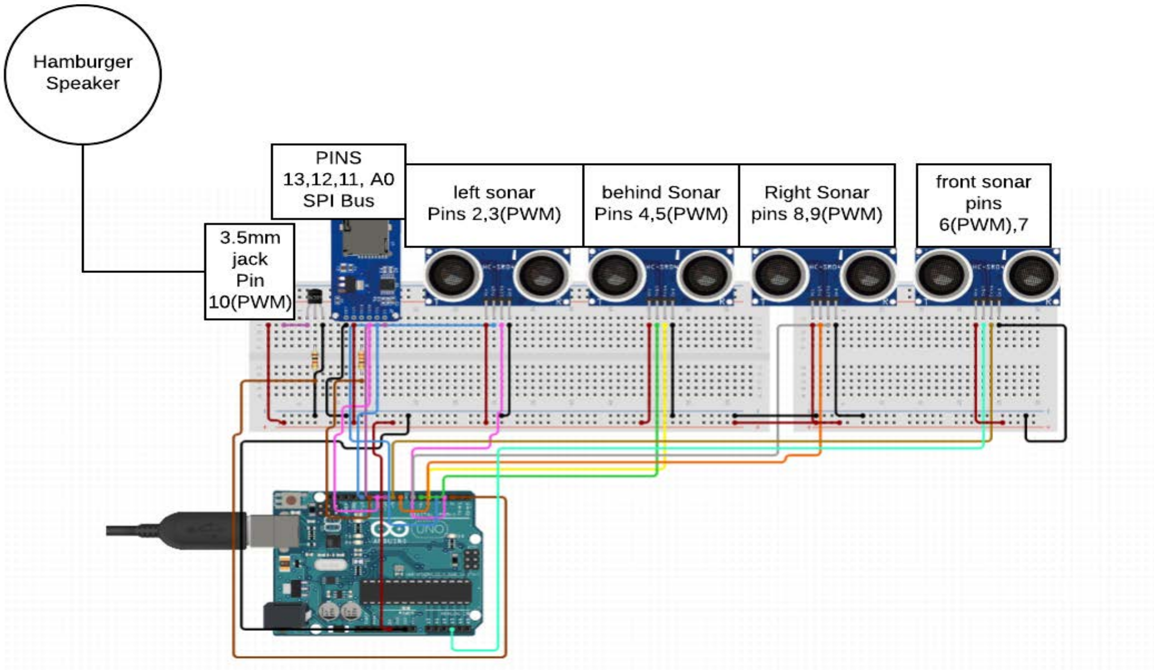
Name	**I/O**	**Description**	**Use**
Surrounding Objects	Input	Walls, humans, objects	Surrounding objects will be used as input to notify user of surrounding obstacles

Human faces	Input	Camera stream looking for faces for facial detection algorithm	Input to camera for facial detection (output)
Camera pictures	Input	Images to give to facial detection algorithm	Used to run the facial detection algorithm
Hamburger speakers	Output	Used for audio cues	Speaker will notify user of direction and recognized faces (left, right, behind, front)
Sonar sensors	Input	Detects objects within a 150cm range	Used to determine the highest priority sonar and the direction

Subsystems:



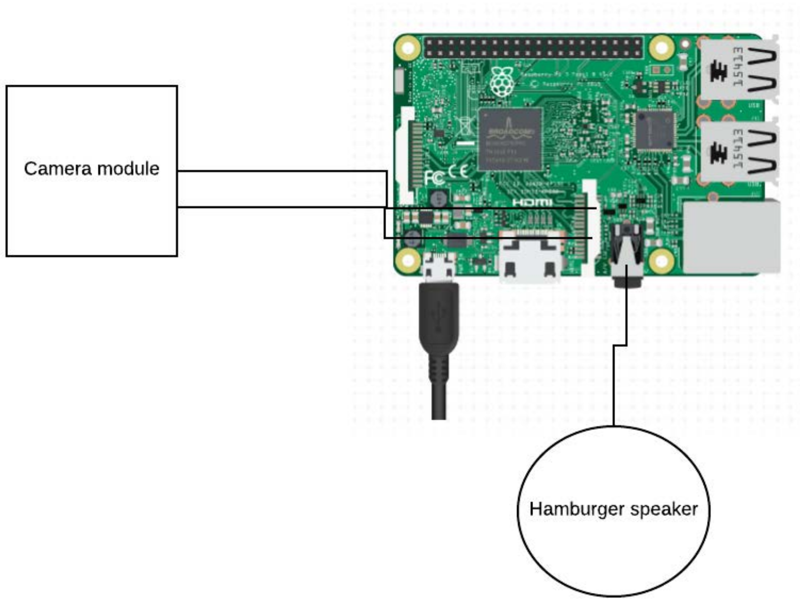
Subsystem 1: Arduino Uno and Sonars:



Name	I/O/Comm	Description	Use
Pins 2,3(PWM)	I/O	Communicate with Echo and Trigger pin of Ultrasonic wave sensor to send and receive distance and direction from the sonar sensor	Informs the Arduino subsystem That an obstacle was detected to the left of the user if distance is less than 150cm
Pins 4,5(PWM)	I/O	Communicate with Echo and Trigger pin of Ultrasonic wave sensor to send and receive	Informs the Arduino subsystem That an obstacle was detected

		distance and direction from the sonar sensor	behind the user if distance is less than 150cm
Pins 8,9(PWM)	I/O	Communicate with Echo and Trigger pin of Ultrasonic wave sensor to send and receive distance and direction from the sonar sensor	Informs the Arduino subsystem That an obstacle was detected to the right of the user if distance is less than 150cm
Pins 6(PWM),7	I/O	Communicate with Echo and Trigger pin of Ultrasonic wave sensor to send and receive distance and direction from the sonar sensor	Informs the Arduino subsystem That an obstacle was detected in front of the user if distance is less than 150cm
PINS SPI(13,12,11), A0	Comm/Out-put	Communicate with SD card reader. Communicate with SD card to read files stored in SD card	Read WAV files to output audio file to the speaker
Pin 10(PWM)	Output	Communicate with 3.5mm jack. WAV file passed through PWM to send signal to speaker	Used for outputting WAV files which are the directions “front, left, right, behind” on the hamburger speaker

Subsystem 2: Raspberry Pi and Pi Cam:



Name	I/O/Comm	Description	Use
Camera Port	I/O	Communication with Camera Module connected to the Raspberry pi 3 B+	Used for facial detection in order to determine who the name of the individual approaching the user
3.5mm jack	Output	Communicate with hamburger speaker	Outputs the name of the detected face

System Capabilities, Conditions, and Constraints

Capabilities:

1. Sonar Sensors

- (Met) The sonar sensors should recognize incoming objects or people less than 150cm from the system.
- (Met) The sonar sensors should be interchangeable in case of repair.

2. Pi Camera

- (Met) The Pi Camera should record a video stream.
- (Met) The Pi Camera should be able to capture images.

3. Portability

- (Met) The Third Eye system should be embedded onto a box as a mock for a human torso.
- (Met) The system should be able to handle walking and general human movement

Conditions:

The system will not function properly in the rain because our system is not waterproof. Wind should not affect the system because it will be strapped down to a box mock. The system should also be able to handle the motion of the user walking. It should be able to survive dropping on hard floor from average human height but in the future better housing would be implemented.

Another key environmental condition is the height of the system, which is dependent on the height of the user. We need to have the sonars and camera be adjustable depending on a person's height and size.

These conditions were met in our testing and development.

Constraints:

Budget - (Met) $150.

Water Durability - (Met) The device cannot get wet.

Size/Aesthetic - (Met) The device will be embedded onto a box to hold the systems together.

PiCam Video Frame Rate - (Met) The video frame rate is limited to 1-2 frames per second due to the Raspberry Pi's processing power.

SD Card Size - (Met) The Raspberry Pi has a 32GB SD Card. The Arduino has an 8GB SD Card.

Battery Life - (Met) The system is powered via an 8000 mAh battery pack. However, the system is restricted to about 2 hours of use due to the speaker's battery life.

Sonar Range - (Met) The sonar sensors are constrained to 150 cm because we only want objects within the user's immediate vicinity.

Number of PWM Pins - (Met) The number of sonars we have is limited by the amount of PWM pins on the Arduino, because we need 2 PWM pins for SPI and the 3.5mm audio jack. 4 PWM pins are used for each of the directional sonars (forward, backward, left, right)

Testing Strategy

Milestone I:

- (Hardware) Sonar Sensor
  - **Test:** We will grab an object and move it within the range of the sonar to better understand the outputs. Outputs and inputs are described below.

- **Short description**: We will test 2 sonars using the Arduino Uno. We are using an output terminal feature on the Arduino IDE to output how far away. If the object is away from the sonar and then use a console output as a mock representation that a signal has been sent to the speaker. We will also be analyzing the interference between 2 sonars.
- **Input:** Waves from surrounding objects(detected from the ultrasonic distance sensor/sonar)
- **Expected output:** Arduino IDE output terminal display will show the distance between the object detected and the sonar sensor. LED light will be used as an indicator for when an object is detected (PASSED)
- **Mocks needed:** LED to represent the speaker
- **Type of test (performance, load, stress, etc.):** Performance, Usability
- **Reference to any requirement validated by the test:** We want to identify the distance from the sonar to an object using an output terminal. We will also use an LED as an indicator that an object is in the sonar's range.

- (Hardware) Speaker
  - **Test:** Use an LED to indicate object detection. Make sure the LED lights appropriately when an object appears to be in the range.
  - **Short description:** For milestone 1, we originally planned to have the speaker working with the Arduino as well with conjunction with the sonar sensors as they send signals. Due to the delay of part ordering, we will be using an LED as a mock part. Otherwise, the speaker would be playing sounds after getting signals from the Arduino.

- **Input:** WAV files from the SD card.
- **Expected output:** Sound (PASSED)
- **Mocks needed:** LED to indicate a signal has been sent to the "speaker"
- **Type of test (performance, load, stress, etc.):** Performance, Usability
- **Reference to any requirement validated by the test:** Using an LED as a mock component for the speaker.

- (Hardware) Camera
  - **Test:** make sure the camera is capable of streaming while connected to the raspberry pi.
  - **Short description:** For the first milestone, our goal is to connect the camera to the raspberry pi and just get the camera to stream an image. For this test we just want to make sure that the equipment will be ready to be used for further testing.
  - **Input:** Environment
  - **Expected output:** Live camera feed (PASSED)
  - **Mocks needed:** camera image in place of video feed
  - **Type of test (performance, load, stress, etc.):** Usability
  - **Reference to any requirement validated by the test:** Setting up the Camera on to the raspberry pi and making sure that it can function.

Milestone II:

- (Software) Face Detection for a Single Face
  - **Test:** Make sure the algorithm using OpenCV is capable of detecting faces.
  - **Short description:** Testing successful face detection through python script and openCV library.

- **Inputs:** Frames captured from a video stream
- **Expected Output:** Identifying a face and putting a box around it to indicate a face has been detected (PASSED)
- **Mocks needed:** None because we have all the parts needed
- **Type of test (performance, load, stress, etc.):** Performance
- **Reference to any requirement validated by the test:** This test utilizes the MP4 and OpenCV standards. We will have facial recognition completed by this milestone.

- (Hardware) Connecting Multiple Sonars
    - **Test:** Measuring outputs of many sonar sensors and priority code
    - **Short description:** Once we have the sonar tests from milestone working, we will begin connecting multiple sonars and seeing how all of them communicate. As a test, we will use multiple objects and move them around the multiple sonars and see the potential outputs from the sonars. We will also test if the priority code is capable of changing due to specific object movements.
    - **Inputs:** Sound waves from external objects
    - **Expected Output:** A pitch from a buzzer (PASSED)
    - **Mocks needed:** Output Terminal Arduino IDE and LED
    - **Type of test (performance, load, stress, etc.):** Performance, usability
    - **Reference to any requirement validated by the test:** Getting multiple sonars functioning.

- (Hardware/Software) Speaker and audio jack breakout (Arduino)
  - **Test:** Begin testing if the speaker can output through the audio jack breakout board
  - **Short description:** The audio jack was wired to read from a PWM pin and send that PWM signal to the speaker and output it.
  - **Input:** input PWM frequency and audio jack
  - **Expected output:** PWM Frequency (PASSED)
  - **Mocks needed:** PWM frequency instead of audio files from SD card since we have not tested the SD card reader yet.
  - **Type of test (performance, load, stress, etc.):** Performance, Usability
  - **Reference to any requirement validated by the test:** Getting the speaker to read from the SD card reader

- (Hardware/Software) Speaker and SD card reader (Arduino)
  - **Test:** Begin connecting the SD card reader and the speaker. In this test, we want to make sure that the SD card reader can read the SD card and output the sound file out to the speaker.
  - **Short description:** Once we have tested the speaker's ability to play WAV files from the SD card we can begin adding multiple WAV files that out the direction of incoming objects.
  - **Input:** input audio file from SD card
  - **Expected output:** Sound (PASSED)
  - **Mocks needed:** Song in place of directions

- **Type of test (performance, load, stress, etc.):** Performance, Usability
- **Reference to any requirement validated by the test:** Getting audio files to work alongside the sonar sensors

Final Demo:

- (Hardware/Software) Speaker and Raspberry Pi Interface
    - **Test:** Begin connecting the Raspberry Pi and the speaker. In this test, we want to make sure that when an object is recognized, the speaker sounds an alert indicating whose face is detected.
    - **Short description:** Once we have tested the facial detection for multiple faces, we will begin adding the speaker component to sound alerts.
    - **Input:** Whose face is detected from a python script
    - **Expected Output:** Sound cue (e.g. a name or direction) (PASSED)
    - **Mocks needed:** None because we now have the speaker
    - **Type of test (performance, load, stress, etc.):** Performance, Usability
    - **Reference to any requirement validated by the test:** This test utilizes the MP4 and OpenCV standards.

- (Software) Multiple Sonar Sensor Priority Algorithm
    - **Test:** Measuring outputs of many sonar sensors and priority code
    - **Short description:** We will use multiple objects and move them around the multiple sonars and see the potential outputs from the sonars. We will also test if the priority code is capable of changing due to specific object movements.
    - **Inputs:** Sound waves from external objects

- **Expected Output:** Buzzer Tone (PASSED)
- **Mocks needed:** Output Terminal Arduino IDE and LED
- **Type of test (performance, load, stress, etc.):** Performance, usability
- **Reference to any requirement validated by the test:** Getting multiple sonars functioning.

- (Hardware/Software) Multiple Sonar Sensors, Speaker and SD card reader (Arduino)
  - **Test:** Begin connecting the speaker to Arduino with multiple sonar sensors. Replacing the buzzer that was in place of the speaker. In this test, we want to make sure that the speaker can take the place of the buzzer without interfering with previous implementation.
  - **Short description:** We want to output the specific direction the object is coming towards the user. The directions being left, right, front, and behind. They are all assigned to a sonar sensor. We want to make sure the speaker works in the same way as the buzzer tones did before in the Multiple sonar sensor algorithm test.
  - **Input:** input audio file from SD card and sonar sensors
  - **Expected output:** Direction Audio file will be output depending on what sonar was triggered (PASSED)
  - **Mocks needed:** N/A
  - **Type of test (performance, load, stress, etc.):** Performance, Usability
  - **Reference to any requirement validated by the test:** Getting audio files to work alongside the sonar sensors

- (Software) Face Detection for a Multiple Faces
  - **Test:** Make sure the algorithm using OpenCV is capable of detecting multiple faces using a pre-trained model.
  - **Short description:** Testing successful face detection to specific person. We will supply our algorithm with 4 sets of images, each with our faces. This will serve as the "contacts list" and we will test to make sure the algorithm successfully matches faces. The OpenCV implementation uses an already trained model.
  - **Inputs:** Frames captured from a video stream, images of group members
  - **Expected Output:** Matching face from an image to their name (PASSED)
  - **Mocks needed:** None because we have all the parts needed
  - **Type of test (performance, load, stress, etc.):** Performance
  - **Reference to any requirement validated by the test:** This test utilizes the MP4 and OpenCV standards. We will have facial recognition completed by this milestone.

- (Hardware) Completed Subsystems
  - **Test:** Make sure the both subsystems are capable of meeting the requirements proposed in our project proposal.
  - **Short description:** The completed subsystems will have the two subsystems working concurrently. The Raspberry Pi will **not** communicate with the Arduino to send information.
  - **Inputs:** Information about recognized faces or incoming collisions with objects will be passed between microcontrollers so we can alert the user

- **Expected Outputs:** A warning or name in the form of audio on two different speakers. (PASSED)
- **Mocks needed:** None
- **Type of test (performance, load, stress, etc.):** Compliance
- **Reference to any requirement validated by the test:** The primary requirement is having the subsystems work together.

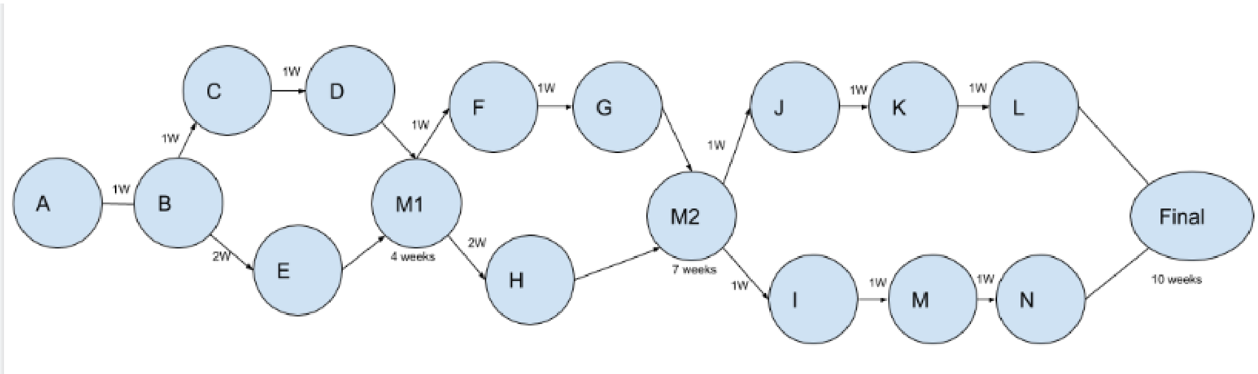
Integration Tests:

Both devices will be integrated so that they run concurrently. They will **not** communicate with each other and exchange information as originally intended. The two systems will be embedded onto the same cardboard box as a mock for a human torso.

For the Arduino subsystem, as an integration test, we wanted to make sure that we could be able to play the audio file for the appropriate direction. To do this we made sure we could get a reading from the appropriate sonar sensor and made sure we could hear the direction "front." We moved the device towards a wall in front of us to see if this would work, and the result was the system saying "front" once the distance detected was less than 150 cm. We tested this for all the directions.

Future Plans

Updated PERT Chart:



A- Proposal	E - Raspbian setup, get Pi to take a picture with the camera	H - Facial detection for 1 face.	K - Fix facial detection accuracy (replaced Pi due to hardware malfunction)	Final - Mount on box portability final demo
B - Standards communication	M1 - Milestone 1	M2 - Milestone 2	L - Add Speaker to function with raspberry pi	
C - Begin object detection with sonar	F - Get a reading from the sonar and output a sound	I - Fix Documentation (Testing Strategy, project	M - Added directions to speaker code to let the user know	

	cue (buzzer as a mock) to indicate direction	requirements) + organize Github repo	where the object is coming from	
D - Multiple sonars (two functioning sonar sensors)	G - Ability to read SD card and output audio file to the speaker, on the second arduino	J - Multiple Sensor Priority Algorithm refinement + multiple facial detection	N - Add 4 sonar sensors and speaker, audio jack, SD card to the same arduino	

Future Directions:

If we were to continue development, we'd like to add social media integration combined with reverse image searching in order to try and guess the identities of strangers. We would also like to develop a better case for the device, ideally one a user can wear instead of carry.

Another major change to our project would be upgrading the processing power of the Raspberry Pi, perhaps by using a more powerful microcontroller in order to expand upon our facial detection algorithm. This could help us achieve more than 1-2 frames per second on the video feed. We would also buy better sensors to provide better accuracy. As well as add more sensors to reduce blind spots.

End of Life (EoL) Plan:

We do not plan to work on this project further. Johnathan Murad is currently planning to hold onto the project and its components for the foreseeable future. Our plan is to finalize our Github (https://github.com/jmura003/cs179-j) by organizing the code into milestone 1, 2, and final demo sections. We will use both the GitHub and the Video demo (https://drive.google.com/file/d/1aquOGJp2uNfdFD0eqieC-O1JrreNR6BF/view?usp=sharing)in order to build our personal project portfolios.

Project Source Code:

Project source code can be found on our GitHub project Page:

https://github.com/jmura003/cs179-j

Team

Esmeralda Mendoza is a fourth year Computer Science major. She has a passion for innovation and problem solving in the field of embedded systems. She primarily worked on the integration of the multiple ultrasonic sensors to determine incoming people or object to the user. As well as the sound component of the device for both the raspberry pi and arduino using audio jack breakout, speaker, and SD card reader. Esmeralda also has excellent interpersonal, communication and team building skills. Her contribution to the documentation, presentation, and github (Kanban) kept the team on track to the ultimate completion goal of the project.

Richard Rangel is a fourth year Computer Science major. He is interested in data science and its applications. He worked primarily on the sonar sensor algorithm that determined sonar priority of an object or individual within 150cm of the Third Eye system. He also implemented zoning in the case of repeated movements on same sonar. In addition to his work with the algorithm, he contributed to the creation of group presentations and keeping the documentation consistently updated to reflect changes to the system over the course of the quarter.

Johnathan Murad is a fourth year Computer Science Major. He is interested in embedded systems and data science. His first contribution to the project was setting up the operating system on the Raspberry Pi and setting up the camera. His primary contribution was implementing the algorithm for prioritising sonar sensors based on the shortest distance recorded in order to output the appropriate direction. He also assisted in zoning, which initiates another sound cue when the user takes step forward or backward. He also assisted in making sure the system stops initiating a sound cue when the sensor with the highest priority stays in the same position. Along with working on implementation, he also assisted in updating documentation and making sure the documentation is as clear and descriptive as possible. He also made sure there were no grammatical or logical errors in the documentation.

Tasmiyah Qazi is a fourth year computer engineering major. She has a passion for embedded systems along with integrated machine learning. She worked primarily on the Raspberry Pi facial recognition with OpenCV and camera functionality. In addition to the speaker interface along with the Python code with the audio files on the Raspberry Pi, she also worked on the algorithm for the functionality of a single sonar sensor. She also helped with Arduino interface with the SD card module. Her contributions to keep the GitHub Kanban board updated as well as issuing tasks for everyone to complete weekly, have been a major aid in keeping the group on track to reach necessary project milestones. Her making sure that all the documentation on Google Team Drive is updated, consistent as well as follow engineering standards also drove the group for stellar documentation.

References

Tutorials and Online Resources:

Raspberry Pi facial recognition: https://www.hackster.io/mjrobot/real-time-face-recognition-an-end-to-end-project-a10826

Single Ultrasonic Sensor: https://howtomechatronics.com/tutorials/arduino/ultrasonic-sensor-hc-sr04/

converted text to speech: https://soundoftext.com/

converted mp3 to wav: https://audio.online-convert.com/convert-to-wav

Arduino Library to play from audio files: https://github.com/TMRh20/TMRpcm

Audio Jack Breakout Wiring: https://learn.sparkfun.com/tutorials/mbed-starter-kit-experiment-guide/experiment-9-pwm-sounds

Datasheets:

SD Card Reader: https://drive.google.com/open?id=15BFKYefoCTteq3tZyVTLO78tOxJqBBz7

Elegoo (Arduino): https://drive.google.com/open?id=1F_056oRuO2ErJQrG9kTYhfin2TzCbXca

Raspberry Pi Camera: https://drive.google.com/open?id=1fT7m5C7362ybgnsyQKIUaq0wEtbqQ1eS

Micro SD Card: https://drive.google.com/open?id=1nEE2HS2JsM3XBX1eCawFNfS9wtaQ41np

Sonar Sensor: https://drive.google.com/open?id=1sByxRy82kSzY1wvPNeA6N-evUp1ATw9v

Speaker: https://drive.google.com/open?id=1x1nOP6aBUWblD77_Ew-s7uA2CBoZJ8t-

Raspberry Pi: https://drive.google.com/open?id=188Q0pGg-d1CiIrrBA16RSVng2zRF6_AI

Standards:

Jpeg: https://drive.google.com/open?id=1erQI8bjDX9xZ-py55iZWJhqnenZOPmQ7

3.5mm Audio Jack: https://drive.google.com/file/d/1aCN2bhg6XC7-NKvn4TwGY83wkM8nzw7T/view?usp=sharing

SPI:

https://drive.google.com/file/d/15vRjo5oomw0pjn7SSPA3jFpmvaB3wfXm/view?usp=sharing

MP4:

https://drive.google.com/file/d/1YC3ncY765iU8kwpXWxws1IsQ6FyZRLDA/view?usp=sharing

Acknowledgments

- Sankha Dutta (TA)
  - Provided feedback throughout the quarter and helped by giving us edge cases to consider while developing our system
- Dr. Philip Brisk (Professor)
  - Helped us understand importance of engineering standards as well as practical aspects such as good documentation and presentation practices.

Picture of your team with your final product