EEE3097S 2023
ASSIGNMENT 1: PAPER DESIGN

1 TABLE OF CONTENTS

2 CONTRIBUTIONS

Student Name	Student Number	Contribution
Md Shaihan Islam	ISLMDS002	Acceptance Test Procedure
		Development Timeline
Tilal Mukhtar	MKHTIL001	Subsystem Design
Aimee Simons	SMNAIM002	Requirement Analysis

3 REQUIREMENT ANALYSIS

3.1 REQUIREMENTS

The objective of this project is to design and implement an acoustic triangulation system using Time Difference of Arrival (TDoA) to accurately locate the position of a sound source within a rectangular grid.

The following system requirements have been identified:

1. The system shall be capable of determining the position of a stationary sound source within a rectangular grid.
2. The system shall provide two-dimensional coordinates relative to the rectangular grid.
3. The system shall utilize two Raspberry Pi (RPi) microcontrollers.
4. The system shall operate the RPi microcontrollers in parallel.
5. The system shall ensure that the RPi microcontrollers are time synchronized.
6. The system shall utilize four microphones.
7. The system shall be capable of simultaneously capturing audio signals from all microphones.
8. The system shall incorporate noise reduction techniques to improve the signal-to-noise ratio (SNR) of the captured audio signals.
9. The system shall calculate the TDoA of the audio signals between all microphones.
10. The system shall employ an appropriate triangulation algorithm to convert the TDoA data into two-dimensional coordinates.
11. The system shall provide a graphical user interface (GUI) for displaying the predicted location of the sound source.
12. The system should ensure that the time synchronization error between the RPi microcontrollers and calculated TDoA values are accurate within 10 microseconds.
13. The system should be capable of capturing audio signals within the audible spectrum.
14. The system should ensure that the sample rate of the microphones is greater than 40kHz.
15. The system should ensure that SNR of the captured audio signals are greater than 60dB.
16. The system should provide the location of the sound source within a 1cm accuracy.
17. The GUI should be intuitive and user-friendly.

3.2 SPECIFICATIONS

The following specifications can be derived from the system requirements:

1. The system will use two Raspberry Pi Zero W modules.
2. The system will use four Adafruit I2S MEMS microphone breakout boards.
3. The system will use an A1 size (59.4 x 84.1 cm) printed grid.
4. The sound source will be generated using an Android smartphone.

5. The RPi microcontrollers will be powered via their micro-USB ports.
6. The RPi microcontrollers will receive an input voltage of 5V DC at an input current of 2.5A.
7. The RPi microcontrollers will be connected to the local network of a host device via Wi-Fi.
8. The RPi microcontrollers will communicate with the host via the Secure Shell (SSH) and Secure Copy (SCP) protocols.
9. The microphone breakout boards will be powered via power connections to the RPi microcontrollers.
10. The microphone breakout boards will receive an input voltage of 3.3V DC.
11. The microphone breakout boards will communicate with the RPi microcontrollers via the I2S serial communication protocol.
12. A Generalized Cross-Correlation Phase Transform (GCC-PHAT) algorithm will be applied to pairs of audio recordings, to determine TDoA data.
13. A Least Squares Estimation (LSE) algorithm will be applied to the TDoA data to determine the estimated coordinates of the sound source
14. The system will be programmed using the Python programming language.
15. The GUI will be programmed using the Tkinter library for Python.

3.3 Possible Implementations

A possible implementation includes:

- Setting up each of the four microphones at each of the four corners of the A1 grid provided.
- Creating an acoustic sound within the grid.
- Obtaining and recording the audio received from each the microphones.
- Passing these audio recordings to the RPi's.
- Filtering the audio recordings to reduce the noise present in the sound.
- Determining the TDoA between each microphone by performing a cross-correlation function on two microphones at a time.
- Set up 2-4 equations in the form:

  $$\sqrt{(x-xi)^2+(y-yi)^2}-\sqrt{(x-xii)^2+(y-yii)^2} - ct_{TDoA}=0$$

  Where (xi,yi) and (xii,yii) represent the coordinates of two separate microphones, (x,y) represent the coordinates of the sound source, c which is the speed of sound and t_{TDOA}, which is the time delay between two microphones.
- Solve for (x,y) using matrix calculations.

3.4 Feasibility Analysis

3 Points of Feasibility will be discussed:

1. Technical Feasibility:
   Insofar as technical resources available, two RPi Zero W modules were provided, along with 4 microphones, 2 SD cards and an A1 grid. It can therefore be said that this design project is technically feasible as the majority of the equipment/materials needed were, in fact, provided. The members of the project are also equipped with the skills needed in order to solve the problem.
2. Financial Feasibility:
   As mentioned previously, two RPi Zero W modules, 4 microphones, 2 SD cards and an A1 grid were all provided and, as such, did not require any financial input from the members of the design team. The software available that will be used for the algorithms are free to use

and readily available. The only expense is the connectors need for the RPi microcontrollers, in order to provide power to the boards as well as display the operating system on a monitor. As such, the project is financially feasible as not a lot of funds are needed to execute the solution.

3. Scheduling Feasibility:
   With reference to the fact that there are no required class test or exams scheduled for the design course, it will allow the members to focus solely on the design and implementation, with minimal distractions. It will, therefore, ensure that scheduling the milestones are more conducive to everyone's schedule. The project is, therefore, feasible with regards to scheduling and time constraints.

3.5 Possible Bottlenecks

1. Hardware Limitations:
   - 1.1. The system is constrained to make use of two RPi Zero W microcontrollers which limits the options available to synchronise the microcontrollers.
   - 1.2. The system is constrained to make use of four Adafruit I2S MEMS microphone breakout boards. This limits the possible accuracy of the TDoA and Triangulation algorithms.
   - 1.3. The system is constrained to an A1 grid size which limits the accuracy of the system.
2. Software Limitations:
   - 3.4. The system will make use of the Python programming language which has worse performance than other suitable programming languages such as C++.
3. Time Limitations:
   - 3.1. The project must be completed over the course of a semester.
4. Information Limitations:
   - 4.1. As this is a relatively new topic to the some of the members, it might be initially difficult to grasp the concept and implement the solution correctly.

4 SUBSYSTEM DESIGN

4.1 SUBSYSTEM AND SUB-SUBSYSTEM BREAKDOWN

The following subsystems and sub-subsystems were identified:

1. Pi Synchronization
   1.1. Pi Communication
   1.2. Pi Timing
2. Signal Acquisition
   2.1. Signal Capture
   2.2. Signal Preprocessing
3. Time Delay Estimation
4. Triangulation
5. User Interface
   5.1. User Interface Functionality
   5.2. User Interface Design

4.2 SUBSYSTEM AND SUB-SUBSYSTEM REQUIREMENTS

Subsystem	Requirements
1. Pi Synchronization	
1.1. Pi Communication	The Pi Communication sub-subsystem shall provide communication between the RPi microcontrollers and user device.
	The sub-subsystem shall utilize bidirectional communication
	The sub-subsystem should ensure minimal latency in data transmission and reception between the microcontrollers.
	The sub-subsystem should include mechanisms to detect and recover from communication failures.
1.2. Pi Timing	The Pi timing sub-subsystem shall establish precise time synchronization between the two RPi microcontrollers.
	The sub-subsystem should ensure that the time synchronization error between the RPi microcontrollers is within 10 microseconds.
	The sub-subsystem should include mechanisms to detect and recover from synchronization failures.
2. Signal Acquisition	
2.1. Signal Capture	The signal capture sub-subsystem shall capture audio signals from the 4 microphones simultaneously.
	The sub-subsystem shall ensure synchronization between microphone sampling and microcontroller timing.
	The sub-subsystem should be capable of capturing audio signals within the audible spectrum.
2.2. Signal Preprocessing	The signal preprocessing sub-subsystem shall provide a timestamp for all captured audio signals
	The system shall incorporate noise reduction techniques to improve the SNR of the captured audio signals.
	The sub-subsystem should ensure that SNR of the captured audio signals are greater than 60dB.

	The sub-subsystem should handle variations in sound source intensity and frequency without significant degradation.
3. Time Delay Estimation	The time delay estimation subsystem shall implement Time-Difference-of-Arrival (TDoA) algorithms to calculate the time differences between audio signal captures from different microphones.
	The subsystem should be computationally efficient.
	The subsystem should ensure that the calculated TDoA values are accurate within 10 microseconds.
4. Triangulation	The triangulation subsystem shall employ an appropriate triangulation algorithm to convert the TDoA data into two-dimensional coordinates.
	The subsystem shall be capable of determining the position of a stationary sound source within a rectangular grid.
	The subsystem should provide the location of the sound source within a 1cm accuracy.
5. User Interface	
5.1. User Interface Functionality	The user interface functionality sub-subsystem should provide real-time updates to the coordinate representation of the sound source’s location.
	The sub-subsystem shall be capable of accepting user input.
	The sub-subsystem should be capable applying user input to the system.
5.2. User Interface Design	The user interface design shall be graphical.
	The user sub-subsystem shall display a coordinate representation of the sound source's location.
	The sub-subsystem should be intuitive and user-friendly.
	The sub-subsystem should be designed to facilitate user interaction

4.3 Subsystem and Sub-Subsystem Specifications

Subsystem	Specifications
1. Pi Synchronization	
1.1. Pi Communication	The RPi microcontrollers will be connected to the local network of a host device via Wi-Fi.
	The host device will use the SSH protocol to transmit Unix commands simultaneously to the RPi microcontrollers
	The host device will use the SCP protocol to retrieve data from the RPi microcontrollers.
1.2. Pi Timing	The audio signals captured by the RPi microcontrollers will be time synchronised using a calibration signal.
	The calibration audio signal will be positioned at the centre of the grid.
2. Signal Acquisition	
2.1. Signal Capture	The audio signals will be captured by the four Adafruit I2S MEMS microphone breakout boards.
	The microphones are omnidirectional.
	The microphones have a frequency range of 50Hz - 15KHz.

	The microphones will be set a sampling rate of 48kHz
	The microphones will be placed on the corners of the rectangular grid.
	The microphones breakout boards will be connected to the RPi microcontrollers via the I2S serial communication protocol.
	Two microphones breakout boards will be connected to each RPi microcontroller.
	The captured audio signals will be stored on microSD cards connected to the RPi microcontrollers.
2.2. Signal Preprocessing	The microphone breakout board has a rated SNR of 65dB.
	The captured audio signals will be passed through a lowpass filter, to filter out any frequencies higher than 15kHz
	The audio signal data will be timestamped using the start of audio capturing as a reference point.
3. Time Delay Estimation	A Generalized Cross-Correlation Phase Transform (GCC-PHAT) algorithm will be applied to pairs of audio recordings, to determine the time delay between the arrival time of sound at each microphone.
	The four audio signals will be synchronised using the calibration signal as a reference.
4. Triangulation	A Least Squares Estimation (LSE) algorithm will be applied to the TDoA data using the coordinates of the microphones as reference points.
5. User Interface	
5.1. User Interface Functionality	The GUI will be programmed using the Python programming language.
	The GUI will be programmed using the Tkinter library for Python.
	The GUI will allow the user to start and stop the acoustic triangulation process.
	The GUI will allow the user to set the coordinates of the reference grid.
	The GUI will allow the user to set the positions of the microphones within the reference grid.
	The GUI will allow the user to set the recording time of the audio signals.
5.2. User Interface Design	The GUI will display a coordinate grid with the predicted location of the sound source.
	The GUI coordinate grid will reference the coordinates of the physical grid used.
	The GUI will display the predicted two-dimensional Cartesian coordinates of the sound source.
	The GUI will allow the user to view the audio signals captured by the microphone.

4.4 Inter-Subsystem and Inter-Sub-Subsystems Interactions

Subsystem	Interactions
1. Pi Synchronization	
1.1. Pi Communication	The sub-subsystem depends on sub-subsystems 2.2 and 5.1.
	Sub-subsystem 2.2. transmits the audio signal data of the audio signals captured and preprocessed by subsystem 2. This transfer is done via SCP.
	Sub-subsystem 5.1 transmits the user input data captured by subsystem 5. This transfer is done wirelessly via the SSH protocol over a local network.
1.2. Pi Timing	The subs-subsystem depends on sub-subsystem 1.1.
	Sub-subsystem 1.1. transmits the time synchronization data. This data is used to time synchronize the two RPi microcontrollers.
2. Signal Acquisition	
2.1. Signal Capture	The subs-subsystem depends on subsystem 1.2.
	Sub-subsystem 1.2. transmits the command to simultaneously capture the audio signals from all four microphones. The transmission to the microphones is done via the I2S protocol.
2.2. Signal Preprocessing	The subs-subsystem depends on sub-subsystem 2.1.
	Sub-subsystem 2.1 transmits the captured audio signal data for preprocessing on the RPi microcontrollers. This transmission is down via the I2S protocol.
3. Time Delay Estimation	The subsystem depends on sub-subsystem 5.1.
	Sub-subsystem 5.1 transmits the audio signal data captured and preprocessed by subsystem 2. The audio signal data is used to calculate the TDoA of the captured audio signals at each microphone.
4. Triangulation	The subsystem depends on subsystem 3 and sub-subsystem 5.1.
	Subsystem 3 transmits the TDoA data of the captured audio signals at each microphone. The TDoA data is used to calculate the two-dimensional Cartesian coordinates of the sound source.
	Subsystem 5.1 transmits the user input data. The user input data is used to set the coordinates of the grid and microphones.
5. User Interface	
5.1. User Interface Functionality	The sub-subsystem depends on subsystem 4 and sub-subsystem 1.1.
	Subsystem 4 transmits the localization data. This data provides the estimated position of the sound source.
	Sub-subsystem 1.1 transmits the audio signal data captured and preprocessed by subsystem 2. This transfer is done wirelessly via the SCP protocol over a local network. The audio signal data is then retransmitted to subsystem 3 for further processing.
5.2. User Interface Design	The subsystem depends on sub-subsystem 5.1.
	Sub-subsystem 5.1 transmits the data to be displayed to the user by the GUI.

4.5 UML Diagram

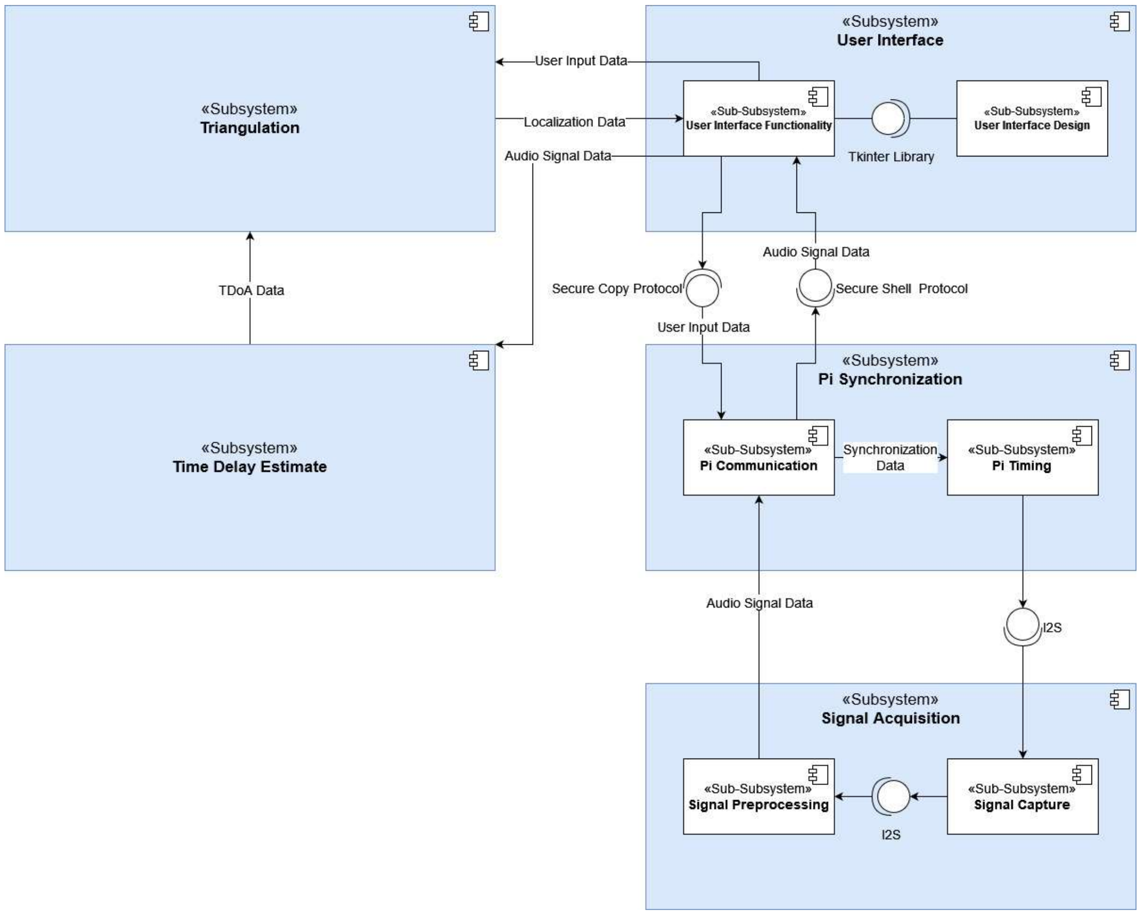


Figure 1: Component UML diagram of the subsystems and sub-subsystems

5 ACCEPTANCE TEST PROCEDURE

For this project, the defined requirements listed previously will each undergo an acceptance test procedure, to ensure that these requirements are met to an acceptable degree. Acceptance test procedures are also vital for problem identification as well as debugging. After each acceptance test procedure, the test will be rated on a scale of 1 – 10, with a rating of 1 representing the test not achieving the desired outcome or behaviour at all, and with a rating of 10 representing the test working perfectly. The table below lists, in detail, the acceptance tests that will be performed, observed and critiqued:

Subsystem	Figures of Merit	Acceptance Test	Acceptable Performance
1. Pi synchronization			
1.1. Pi Communication	Test message	To ensure bidirectional communication, a test message will be sent from each of the RPi's to each other.	Both RPi modules can read messages sent to them from the other module and display the message.
1.2. Pi Timing	Response messages sent by the microcontroller when it has connected with the other microcontroller, as well as an indication of start of operation.	To ensure time synchronisation of the microcontrollers, two acceptance tests will be performed: • A test will be performed to ensure the microcontrollers only start the detection process once both have established bidirectional communication with each other. • A test will be performed to ensure that both the microcontrollers are functioning simultaneously to locate the sound source once the communication has been established.	If the appropriate response messages were returned at the appropriate times, performance will have been deemed acceptable.
2. Signal acquisition			

2.1. Signal Capture	A switch combined with the microphone output can be used to keep an LED on when the microphone is able to record a sound.	For signal capture, an acceptance test will be performed to determine whether the microphone was able to capture the sound being emitted from the source. Furthermore, a test can be conducted to ensure the microcontroller sends a response message once it has been able to capture the sound signal.	If the LED is lit when the sound source is emitting sound, and the LED is off when the sound source is not emitting sound, acceptable performance is met. It is also met when the appropriate response message is outputted by the microcontroller.
2.2. Signal Preprocessing	Timestamps Signal to noise ratio (SNR)	This sub-subsystem requires the timestamps to be captured for the audio signals, and this will be tested using the Pi microcontrollers, which will be outputting these timestamps. Tests will also be conducted to measure the signal-to-noise ratio (SNR). Information regarding the noise in the system will be taken from the filter that will be used to attenuate the noise from the signal, which can then be used to calculate the SNR.	Timestamps accurately align with the audio signals SNR must be less than 60dB
3. Time delay estimation	Timestamps	Correctness of the time delay will be tested for. Information regarding the time delay will be computed using the RPi microcontrollers, and the distance that was calculated using the Time-Difference-of-Arrival equations will be outputted by the microcontrollers.	Timestamps that produce an accurate distance estimation will be considered acceptable performance.
4. Triangulation	Sound Source Coordinates	This subsystem will be involved in a detailed	For this experiment,

	outputted by the RPi microcontroller.	acceptance test, stated as follows: • The sound source will be placed at various positions within the grid, for example: close to a microphone, at the center of the grid, as well as outside the grid. The system will then be tested to see if provides the appropriate coordinate information	both the x-axis and y-axis coordinates must be calculated and outputted correctly. The coordinates must also have a maximum tolerance of 1 centimetre to ensure acceptable test performance.
5. User interface			
5.1. User Interface Functionality	Elements of the user interface, such as buttons, text boxes and display panels.	Each input and output element will be tested for functionality and correctness. The inputs will be executed numerous times, and the output of these inputs will also be recorded, to determine whether the input achieves the desired output/function.	Each element in the interface needs to behave as desired, and it must do so in a responsive, fast manner.
5.2. User Interface Design	Overall layout and outlook of the user interface, as well as ergonomics	The user interface will be tested in its ergonomic aspect and ease of use. A volunteer, with no prior knowledge of the project, will be asked to use the program and comment/rate on how easy or difficult the interface is to use and understand.	An average rating of 7 or above given by the volunteers will be deemed acceptable performance.

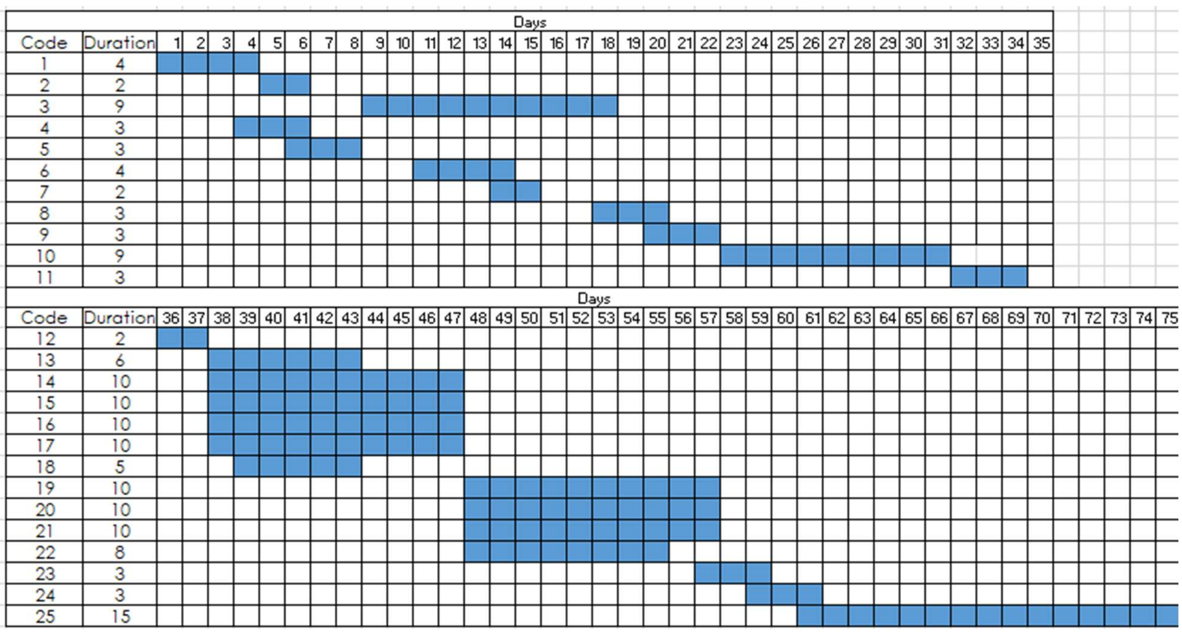
6 DEVELOPMENT TIMELINE

The Work Breakdown Structure (WBS) of the project is displayed below:

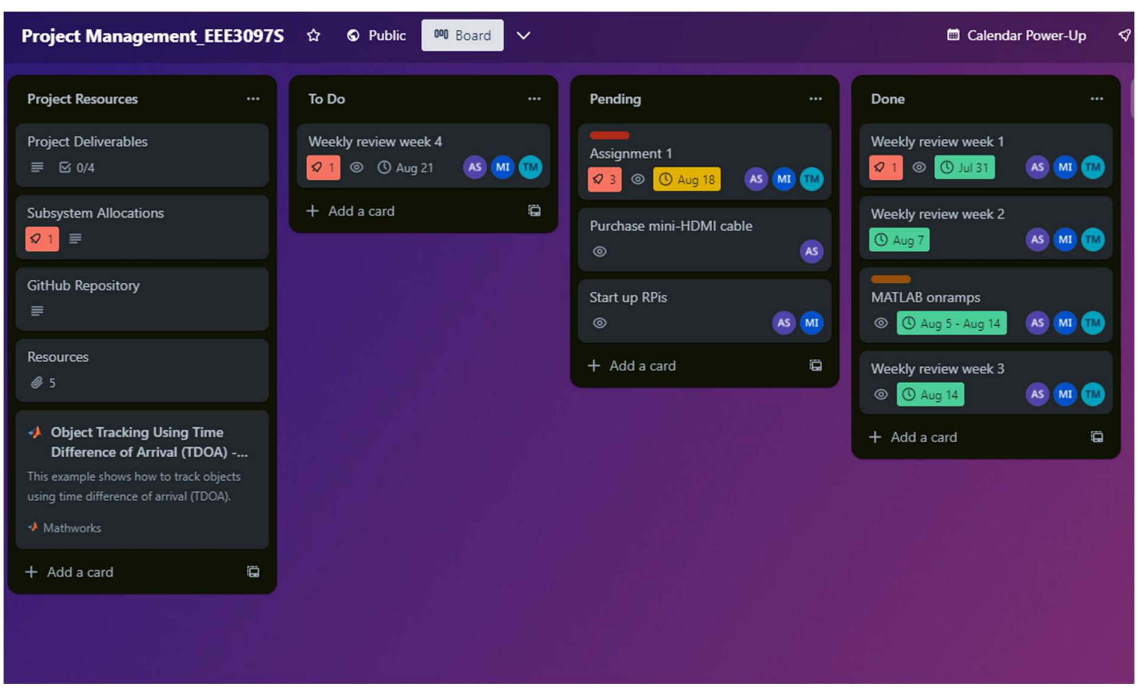
Code	Activity	Duration	Dependency
1	Proposition of project idea and commencement of planning	28/07/2023 - 31/07/2023	None
2	Begin background research on aspects of project	31/07/2023 - 01/08/2023	1
3	Complete MATLAB Onramp courses	05/08/2023 - 14/08/2023	1
4	Identification of project requirements	31/07/2023 - 02/08/2023	1
5	Analysis of project requirements	02/08/2023 - 04/08/2023	4
6	Development of specifications	07/08/20223 - 10/08/2023	4, 5
7	Subsystem Breakdown	10/08/2023 - 11/08/2023	4, 5, 6
8	Write up of inter- and intra-subsystem interactions	14/08/2023 - 16/08/2023	7
9	Define acceptance test procedures	16/08/2023 - 18/08/2023	8
10	Setup of simulation using MATLAB	21/08/2023 - 29/08/2023	3, 6, 8
11	Testing of MATLAB simulation	30/08/2023 - 01/09/2023	10
12	Setup of physical system	11/09/2023 - 12/09/2023	11
13	Pi synchronisation setup	13/09/2023 - 18/09/2023	12
14	Development of code for signal acquisition and preprocessing	13/09/2023 - 22/09/2023	11
15	Development of code for time delay estimation	13/09/2023 - 22/09/2023	11
16	Development of code for triangulation/localisation algorithm	13/09/2023 - 22/09/2023	11
17	Development of user interface	13/09/2023 - 22/09/2023	3
18	Testing and debugging of Pi synchronization	14/09/2023- 18/09/2023	13
19	Testing and debugging of signal acquisition and preprocessing	22/09/2023 - 31/09/2023	14
20	Testing and debugging of time delay estimation	22/09/2023 - 31/09/2023	15
21	Testing and debugging of triangulation/localisation algorithm	22/09/2023 - 31/09/2023	16
22	Testing and debugging of user interface	22/09/2023 - 29/09/2023	17

23	Overall performance evaluation	31/09/2023 - 02/10/2023	13, 14, 15, 16, 17
24	Final Testing	02/10/2023 - 04/09/2023	23
25	Final Report Write-up	04/09/2023 - 18/09/2023	24

The Gantt chart for the development timeline is shown below:



The Project Management Page is shown below:



7 BIBLIOGRAPHY

[1] Da Costa, D.G. (2022) Time Difference of Arrival Acoustic Triangulation Using a Distributed Sensor Network. Dissertation

[2] "Raspberry pi documentation," Raspberry Pi. [Online]. Available: https://www.raspberrypi.com/documentation/

[3] "Adafruit I2S MEMS microphone breakout," Adafruit Learning System, [Online]. Available: https://learn.adafruit.com/adafruit-i2s-mems-microphone-breakout/overview

[4] B. Jin, X. Xu, and T. Zhang, "Robust time-difference-of-arrival (tdoa) localization using weighted least squares with cone tangent plane constraint," Sensors (Basel, Switzerland), vol. 18, 03 2018. [Online]. Available: https://www.mdpi.com/1424-8220/18/3/778

[5] D. Dalskov and S. K. Olesen, 'Locating acoustic sources with multilateration', Master's, 2014. [Online]. Available: https://vbn.aau.dk/ws/files/198526294/Measurements/Source%20signals

[6] M. Pollefeys and D. Nister, 'Direct computation of sound and microphone locations from time-difference-of-arrival data', in 2008 IEEE International Conference on Acoustics, Speech and Signal Processing, 2008, pp. 2445–2448. [Online]. Available: https://cvg-pub.inf.ethz.ch/WebBIB/papers/2008/001_PollefeysICASSP08.pdf

[7] T. Ho et al., "Acoustic Source Localization," Georgia Institute of Technology. [Online]. Available: https://eceseniordesign2022spring.ece.gatech.edu/sd22p11/finalwrittenreport.pdf

[8] N. R. Kumarasiri, Development of novel algorithms for localization in wireless sensor networks. The University of Toledo, 2014.