ECE 53800: Digital Signal Processing

Final Project Report - Group 1

Emergency Vehicle Detection Using Microphone Array and Sound Localization

Priyank Kalgaonkar – Team Member
Kavyashree Prasad S P – Team Member
Arjun Narukkanchira Anilkumar – Team Member

Department of Electrical and Computer Engineering,
Purdue School of Engineering and Technology at IUPUI.

Submitted to:

Course Instructor
Dr. Mohamed El-Sharkawy
Department of Electrical and Computer Engineering, IUPUI.

Submitted as partial fulfillment for the requirement of Fall 2019 - ECE 53800: Digital Signal Processing course.

Date of Submission
December 09, 2019

Executive Summary

Emergency Vehicle Detection Using Microphone Array and Sound Localization system uses a microcontroller unit, an array of microphones point in different directions and LED bulbs. The main objective of Emergency Vehicle Detection Using Microphone Array and Sound Localization project is to inform and alert the direction of approaching emergency vehicle(s) to the drivers, cyclists and pedestrians on the road and/or at an intersection. The result is to demonstrate this idea of sound localization and hence, enable new means of alerting drivers and pedestrians of approaching emergency vehicle beforehand.

As per the United States Access Board, a federal agency that promotes equality for people with disabilities through leadership in accessible design and the development of accessibility guidelines and standards for the built environment, transportation, communication, medical diagnostic equipment, and information technology, there are over 300,000 signaled intersections in the USA that roughly states to one signalized intersection per 1,000 living population. Furthermore, based on the United States Census Bureau's forecast of future population growth of 0.85% per annum, approximately 2,500 new signalized intersections are introduced every year[1]. However, very few intersections utilize a single white flashing light. If this white flood light at the intersection is flashing, an emergency vehicle is approaching; it does not specify the direction of travel of the approaching emergency vehicle.

Our design of Emergency Vehicle Detection Using Microphone Array and Sound Localization prototype utilizes an array of microphones pointing in different directions and LED bulbs to display the direction of travel of emergency vehicles. The assumption is that the array of microphones should be receiving the same auditory signals but at different intervals of time. The sound captured by an array of microphones will be sampled. It will then be filtered and angle of the receiving soundwave that corresponds to the frequency of the emergency vehicles' sirens will be used to determine the direction of the corresponding soundwave and light up the LEDs accordingly.

This report provides detailed information that will be useful for companies interested in this concept and the engineering team(s) responsible for further research and development.

Acknowledgement

This project consumed an immense amount of work, research, and dedication. Nonetheless, implementation would not have been possible without the support of many individuals and organizations.

We are highly indebted to our professor: Dr. Mohamed El-Sharkawy, and our teaching assistants: Dewant Katare and Niranjan Ravi, at the Purdue School of Engineering and Technology at IUPUI for their guidance and support in completing this project. During the course of this work, their constant association with my team has been most pleasurable. Therefore, on behalf of all my team members, I would like to extend our sincere gratitude to all of them.

We are also grateful to Mr. Jeffrey Sears who provided us with access to the SL111 lab in the Engineering Science & Technology (SL) Building and with necessary tools to build, develop, solder and test hardware components for this project. Without his time and support in providing necessary tools and components, our hardware team would have been greatly challenged to develop, integrate and test the hardware subsystems and components for this project.

My thanks and appreciations also go to my colleagues involved in developing this project and especially to those who have willingly helped me out with abilities at their best. I would like to extend my special gratitude to my team members: Arjun Narukkanchira Anilkumar and Kavyashree Prasad S P, the members of Group #1 actively involved in this project.

We would like to express our sincere thanks towards volunteer helpers, family, and friends who provided us with essential support and additional resources in the implementation of this project.

Without the help and counsel of the aforementioned individuals and organizations, always generously and unstintingly given, the completion of this work would have been immensely difficult and time consuming.

On behalf of Group 1,
Priyank Kalgaonkar.

Table of Contents

1: Introduction 8

2: Problem Statement 9

2.1: Need 9

2.2: Objective 9

2.3: Marketing Requirements 9

3: System Design 11

3.1: System Architecture 11

3.2: Data Flow 12

3.3: Engineering Requirements 13

3.4 Validation 13

4: Standards Compliance 15

4.1: Hardware Standards 15

4.2: Software Standards 16

5: Detailed Design 17

5.1: Physical Design 17

5.2: Component Dimension 17

5.3 Software Design 18

5.4 Hardware Design 21

6: Integration and Testing 23

6.1: Integration Plan 23

6.1.1: Software Integration Sequence 23

6.1.2: Hardware Integration Sequence 23

6.1.3: Full Hardware-Software Integration Sequence 24

6.2: Unit Testing - Hardware 25

6.3: Integration and Testing - System Acceptance Test 27

6.4 Final Functional Test 28
7: Economics 29
7.1: Cost of Materials 29
7.2: Engineering Labor Hours 30
7.3 Specialized Facilities and Resources 30
8: Project Management Plan 31
8.1: Pre-defined Schedule (Gantt Chart) 31
8.2: Risk Management 31
9: Conclusion 33
10: References 34
11: Appendices 35
11.1 Appendix A - Component Pinouts 35
11.2 Appendix B - Source Code 36

List of Tables

Table 3.3.1: Engineering Requirements Justification Table ………………………… 13

Table 5.1.2: Table for Component Dimensions ……………….………..……………… 17

Table 6.2.1: Unit Test for microphone modules ……………….…………………..... 25

Table 6.2.2: Unit Test for LED bulbs ……………………………………………..... 26

Table 6.2.3: Unit Test for FRDM K64F MCU ……………………………….…..... 26

Table 6.3.1: System Acceptance test ..………………………………………………. 27

Table 7.1.1: Cost of billable material ………………………………………………. 29

Table 7.2.1: Engineering labour hours……………………………………………….. 30

Table 7.3.1: Table of specialised facilities and resources…………………………….. 30

List of Figures

Figure 1.1.1: Prototype Photo ……...……………………..……………………….. 8

Figure 3.1.1: System Architecture ………….……………………………...…… 11

Figure 3.2.1: Data flow diagram..……………………...…………………….....….... 12

Figure 5.1.1: Prototype Photo ………………………………………………………... 17

Figure 5.3.1: Software design flowchart……………………………………….... 20

Figure 5.4.1: System schematics ……………………………………………………... 21

Figure 6.1.2: Hardware Integration Order ....…………………………………………. 24

Figure 6.1.4: Full hardware software integration order ………………………………. 25

Figure 8.1.1: Project schedule gantt chart……………………………………………... 31

Figure 11.1.1: NXP-FRDM K64f header pinout diagram…………………………….. 35

Figure 11.1.2: MAX 9814 microphone pinout………………………………………..... 35

1.0 INTRODUCTION

Very few intersections in the United States utilize a single white flashing light. If this white flood light at the intersection is flashing, an emergency vehicle is approaching the intersection. However, it does not specify from which direction the emergency vehicle is approaching. This can result in confusion between drivers and bystanders to make way for the emergency vehicle to safely pass by and can cause the vehicle to slow down or worse, come to a complete stop until the intersection is clear or a new route is created for the emergency vehicle to pass by. A solution is needed to mitigate this issue.

The idea of the Emergency Vehicle Detection Using Microphone Array and Sound Localization project relates to use of hardware and Digital Signal Processing techniques such as sampling and filtering. In this project, we utilize an array of microphones which are pointing in different directions so that each microphone will detect the same sound (emergency vehicles' sirens) but at different intervals of time. This sound captured by an array of microphones will be sampled. It will then be filtered and angle of the receiving soundwave that corresponds to the frequency of the emergency vehicle's siren will be used to determine the direction of the corresponding soundwave and light up the LEDs accordingly.

For building a prototype of this system, we chose to implement two microphones with Automatic Gain Control and Low-Noise Microphone Bias, each placed on the far ends of breadboard (a construction base for prototyping of electronics), a NXP FRDM K64F microcontroller and five Light-Emitting Diodes (LED) as pictured below.

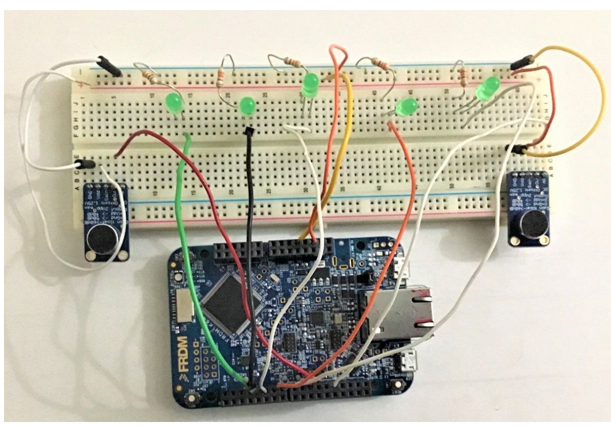

Figure 1.1: Prototype Photo

2.0 PROBLEM STATEMENT

2.1 Need

As stated in the introduction, very few intersections in the United States utilize a white flashing flood light system to warn motorists, cyclists and pedestrians on the road of an approaching emergency vehicle such as an ambulance, a fire truck, a police vehicle, etc. Most of the intersections do not have any kind of mechanism in place at all and hence, drivers have to rely on their ability to see and hear in order to become aware of the approaching vehicle. This ability can be hindered due to various reasons, for example: opaque objects such as trees, small structures or a building on the corner, to name a few.

2.2 Objective

To inform motorists, cyclists and pedestrians on the road of an approaching emergency vehicle, we are introducing the idea Emergency Vehicle Detection Using Microphone Array and Sound Localization that utilizes digital signal processing techniques such as sampling and filtering to process the sound captured by an array of microphones to provide a real-time output. This will result in vehicles and people around the intersection and/or where this system is implemented, be better informed of the exact direction of the approaching emergency vehicle and make way beforehand for it to pass by safely and quickly.

2.3 Marketing Requirements

The marketing requirements are focused on making this system easy to implement by retrofitting the existing signaling system at an intersection or including in the design and planning phase of the new intersections. They are as follows:

1. Processing should be done in real-time.
2. Few modifications should be required to retrofit to existing signaled intersections.
3. This system should be less expensive to maintain.

The detailed engineering requirements derived from these marketing requirements are listed in section 3.3. Further discussion on system design will follow in the coming sections.

3.0 SYSTEM DESIGN

3.1 System Architecture

The overall architecture of the Emergency Vehicle Detection Using Microphone Array and Sound Localization system has at its center the NXP FRDM K64F Microcontroller Unit (MCU) featuring an ARM Cortex M4 core running up to 120MHz, embedding 1024KB Flash and 256KB RAM. The FRDM-K64F MCU receives sound signals from two MAX9814 microphones which features Automatic Gain Control and Low-Noise Microphone Bias, processes the sound signals and output is given out to the corresponding LED. The functional architecture is drawn in the figure below:

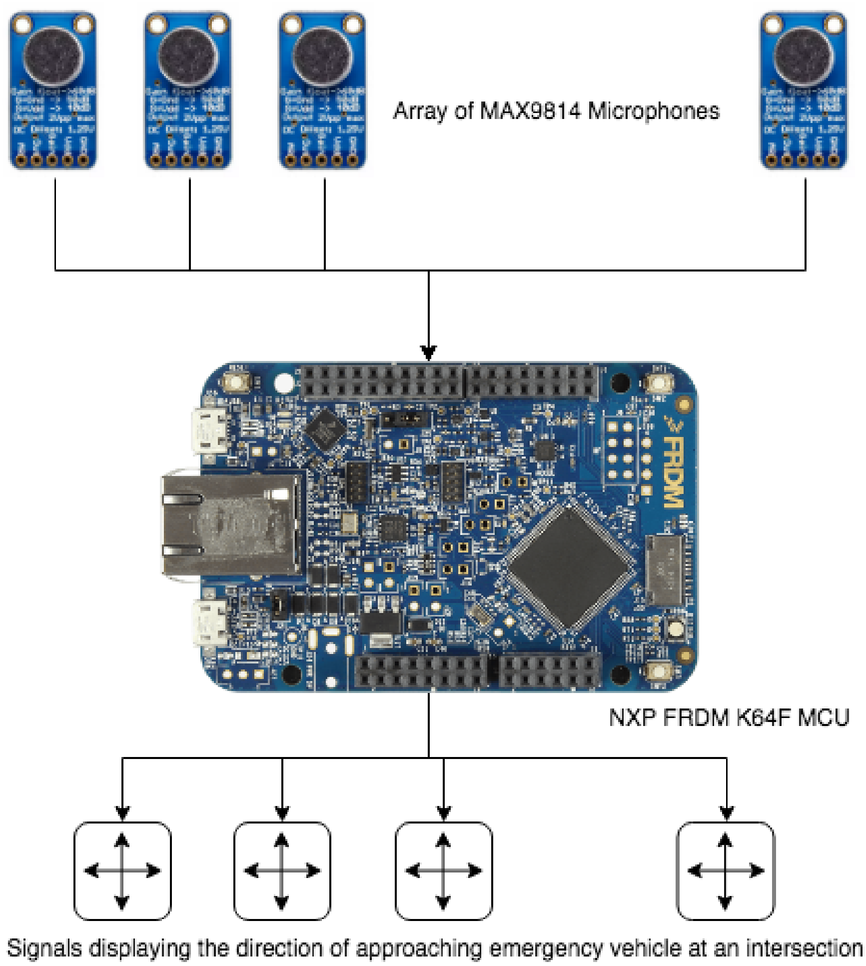

Figure 3.1.1: System Architecture

3.2 Data Flow

The system consists of four stages audio acquisition, audio processing, angle estimation and signalling/ indication. Audio acquisition is done via microphone array which receives analogue signal and is converted to digital after sampling. Audio processing includes the band pass filter to filter out ambulance sound and use cross correlation among the signals received from two different microphones. Angle estimation is done with calculations using cross correlated values along with other parameters. Finally, the output is indicated using led bulbs each indicating specific angle.

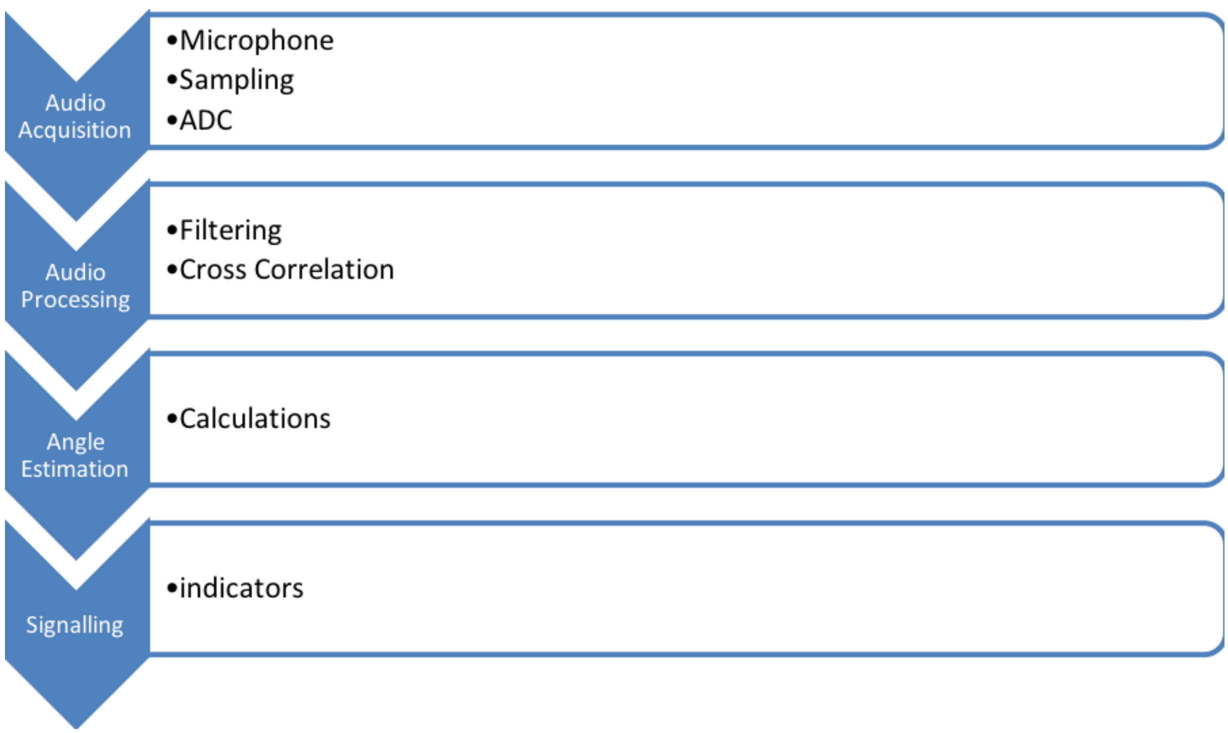

Figure 3.2.1: Data Flow Diagram

3.3 Engineering Requirements

To satisfy the marketing requirements explained in section 2.3, we have set the engineering requirements listed below.

Marketing Require-ment	Engineering Requirement	Justification
1	System should process sound signals acquired in real-time.	This will facilitate the LEDs that will signal of the approaching emergency vehicle in real time and alert motorists and pedestrians before the actual arrival of fast approaching emergency vehicles at the intersection.
2	System implementation should require minimal modifications to existing signaled intersections.	This will allow the installation costs and signaled intersection downtime to be very less and hence, many counties will be able to afford to implement this system.
3	Cost of maintenance should be less.	Since the size of components chosen for this system is relatively small, it can be integrated inside the existing weather protective covers used for electronics at the signaled intersection.

Table 3.3.1: Engineering Requirements Justification Table

3.4 Validation

The following analyses explain how our design is expected to meet the engineering requirements. Each explanation is listed under the corresponding engineering requirement, reprinted below for convenience:

1. **System should process sound signals acquired in real-time.**

   Since FRDM-K64F is powered by an ARM Cortex M4 core running up to 120MHz, the sound should be processed with acceptable delay.

2. **System implementation should require minimal modifications to existing signaled intersections.**

   Since we propose to use an array of small microphones, a microcontroller and dedicated LED bulbs/display, cost of implementation can be kept at a minimum.

3. **Cost of maintenance should be less.**

   Since the size of components chosen for this system is relatively small, it can be integrated inside the existing weather protective covers used for electronics at the signaled intersection.

4.0 STANDARDS COMPLIANCE

4.1 Hardware Standards

All the components used in this system are bought off-shelf from the electronics store, which are manufactured in compliance with the following standards and hence, the overall system, when fully assembled, will be in compliance with the standards described below in detail:

A. Safety and Health Compliance Standards:

EN 60950-1:2006 + A1:2010 + A11:2009 + A12:2011: The safety standard EN 60950-1 applies to mains powered equipment such as the NXP FRDM-K64F, MAX 9814 Microphone, and Light-Emitting Diodes. “Requirements within this standard address normal equipment operating conditions, likely and consequential faults, foreseeable misuse and external influences such as temperature, altitude, pollution, moisture and over voltages. In general, these requirements are intended to minimize the risk of fire, electric shock and injury for the operator, layman, service person and anyone else that may come into contact with the equipment during installation, operation and maintenance.” [4][5][6]

B. Electromagnetic Compatibility (EMC) Compliance Standards:

EN 55022:2010: The safety standard EN 55022 applies to the Radio disturbance characteristics of the components in the system. All the components of the system have been tested by the manufacturer and found to comply with the limits for Class B Information Technology Equipment according to the European Standard except the touch screen display which complied with the limits for a Class A Information Technology Equipment according to the European Standard. [4][5][6]

C. Federal Communications Commission (FCC) Compliance Standards:

FCC Emissions Compliance Statement: All the components of the system have been tested by their respective manufacturers and found to comply with the limits for a Class A digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. All the components of the system comply with Part 15 of the FCC Rules. Operation is subject to the following two conditions:

a) This system might not cause harmful interference

b) This system must accept any interference received, including interference that may cause undesired operation. [4][5][6]

D. Restriction of Hazardous Substances Directive (RoHS) Compliance Standards:

RoHS Directive Compliance Statement: NXP FRDM-K64F MCU board and other components of this system complies with the relevant provisions of the RoHS Directive for the European Union. In common with all Electrical and Electronic Equipment (EEE), the NXP FRDM-K64F MCU or any components of the Emergency Vehicle Detection Using Microphone Array and Sound Localization system should not be disposed of as household waste. Alternative arrangements may apply in other jurisdictions. [4][5][6]

4.2 Software Standards

Since our project includes custom software developed in the C programming language in Keil Microcontroller Development Kit (MDK) which includes includes the µVision IDE and debugger, Arm C/C++ compiler, and essential middleware components, it is expected to adhere to Arm Holding's standards. The MDK is licensed under the permissive Apache 2.0 (Open-Source) license and so it can be used in both commercial and personal projects. The software was developed by us.

5.0 DETAILED DESIGN

5.1 Physical Design

Since our prototype is designed to be retrofitted on to existing signalized intersection with minimal modification or to be incorporated in the design of new signalized intersection, we have setup a prototype of our system by utilizing a simple breadboard and connecting wires as shown below:

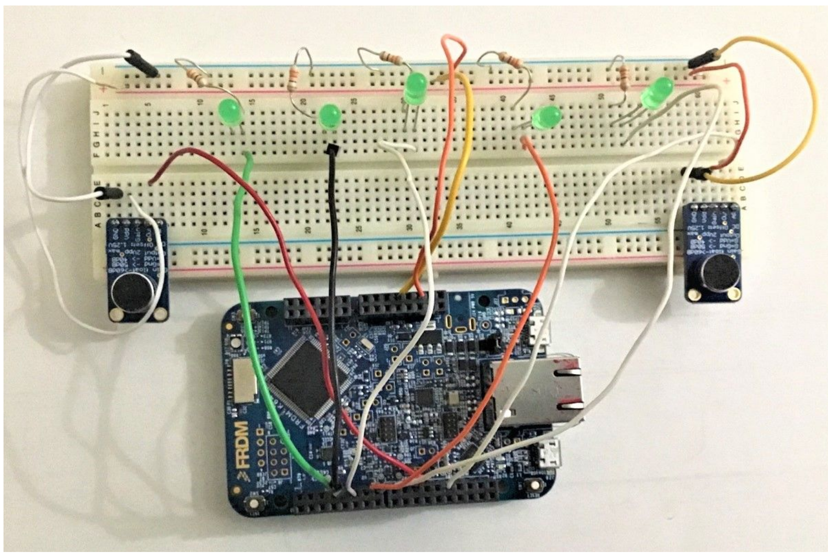

Figure 5.1.1: Emergency Vehicle Detection Using Microphone Array and Sound Localization Prototype

5.2 Component Dimension

Measurements of the main components (excluding wires) are shown in Table 5.1.2 below:

Component	Height (mm)	Width (mm)	Depth (mm)
NXP FRDM-K64F MCU	81	53	7
MAX9814 Microphone	18	12	7
LED Bulb	12	3	3

Table 5.1.2 Table for Component Dimensions.

5.3 Software Design

The software was developed in Keil µVision 5.28 MDK utilizing C Programming Language, which is one of the recommended programming languages. The software design consists of:

1. Starting up the code.
2. Checking C Library integration.
3. Checking Peripheral libraries.
4. Executing the code.

MVC stands for Model, View and Controller — the three main elements of an MVC design as follows:

- Model: The model is a representation of the data. In this case, the model is receiving data from the microphone sensor.
- View: The view is represented by Light-Emitting Diodes that glow indicating the direction of source of sound or, in this case, the direction of the approaching emergency vehicle.
- Controller: The controller coordinates the model and the view. In this case, Controller is FRDM-K64F MCU.

Our software design contains the following steps:

1. <u>Initialization of ADC'S</u>: The incoming sound acquired by the microphones was passed through the ADC's. ADC present in the boards were initialized and configured to appropriate settings by registers CFG1 and CFG2 for ADC0 and ADC1 respectively.16 bit mode was enabled with a short sample time. We used the default bus clock value of 20.97 MHz. After referring to the FRDM K64F manual it was inferred that, SFC ladder consumes 5 ADCK cycles and 5 bus clock cycles. Basic ADC conversion time is 25 ADCK cycles. HSC Adder uses 2 ADCK cycles and LST Adder takes 0 ADCK cycles.

Hence, 37 cycles in overall is need for conversion. Based on this, sampling frequency was found to be 20.97MHz/37 = 566.76 kHz. Therefore sampling time is 1.76 microseconds.

2. Design of Band pass Filters: Since the ambulance sound falls in the frequency range 500-1000 KHz[9]. A band pass filter was incorporated to avoid interference from other sources. A digital FIR filter with an order of 40 was designed in matlab and the determined coefficients were passed as a header file to the main code.
3. Time delay determination: Incoming sound from the two microphones are first filtered out. They are then cross-correlated to determine the time delay in reception of sound between them. This is achieved by determining the maximum cross correlation value and the index at which it occurs. This index is subtracted from the total length of input array containing the incoming sound to find out the time delay.
4. Calculation of incoming sound angle: The angle of incoming sound is calculated using the formula given below:

$$\theta = sin^{-1} \frac{\Delta t}{d}$$

where θ is the angle in radians, Δt is the sampling period ,v is the velocity (343 m/s) and d is the distance between the two microphones. Our code contains range of angles which light up certain LEDs if the calculated angle falls in that range. Based on the angle obtained, corresponding LED lights up to indicate the direction of sound.

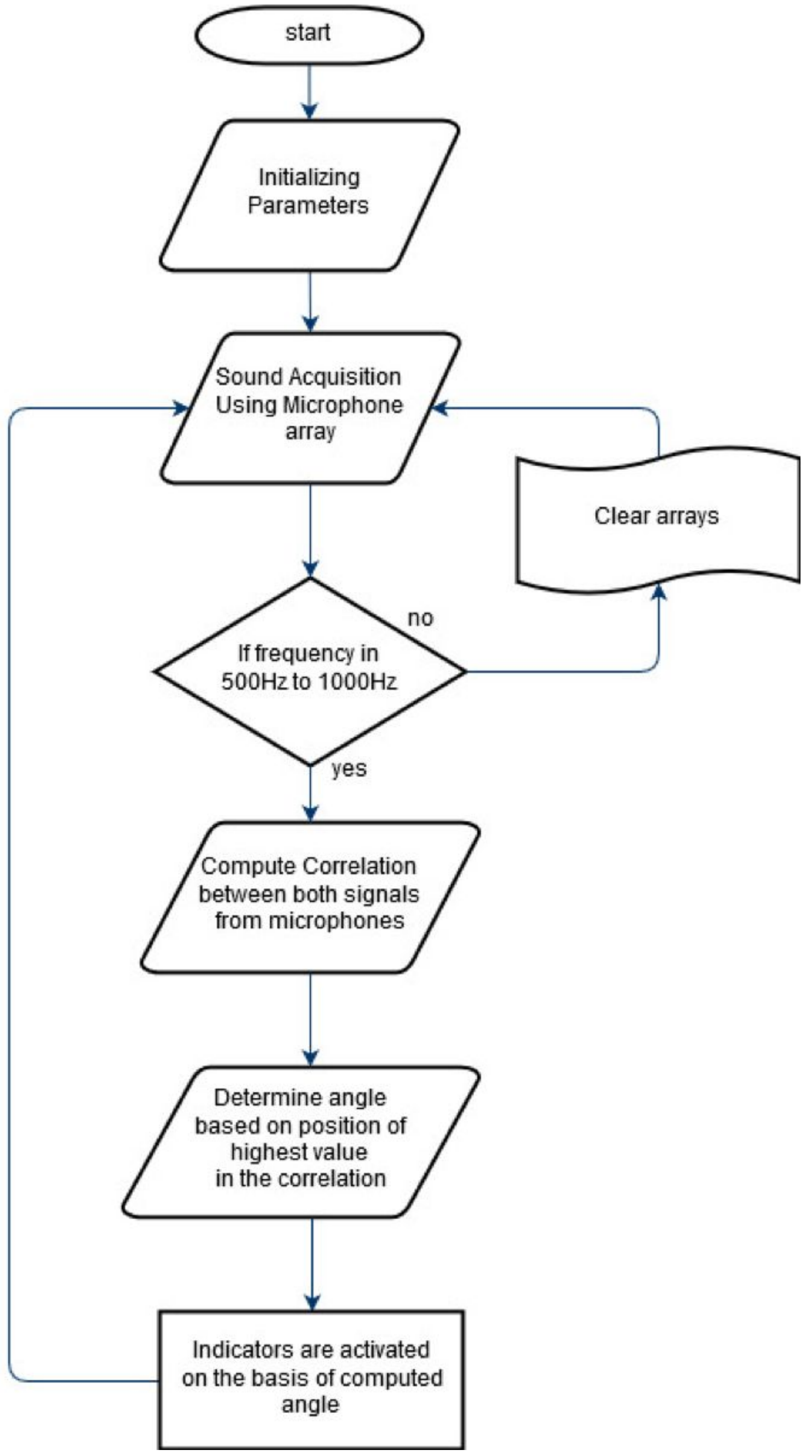

Figure 5.3.1: Software Design Flow Chart.

5.4 Hardware Design

Schematic for our prototype is shown in fig. It has been designed using EagleCAD.

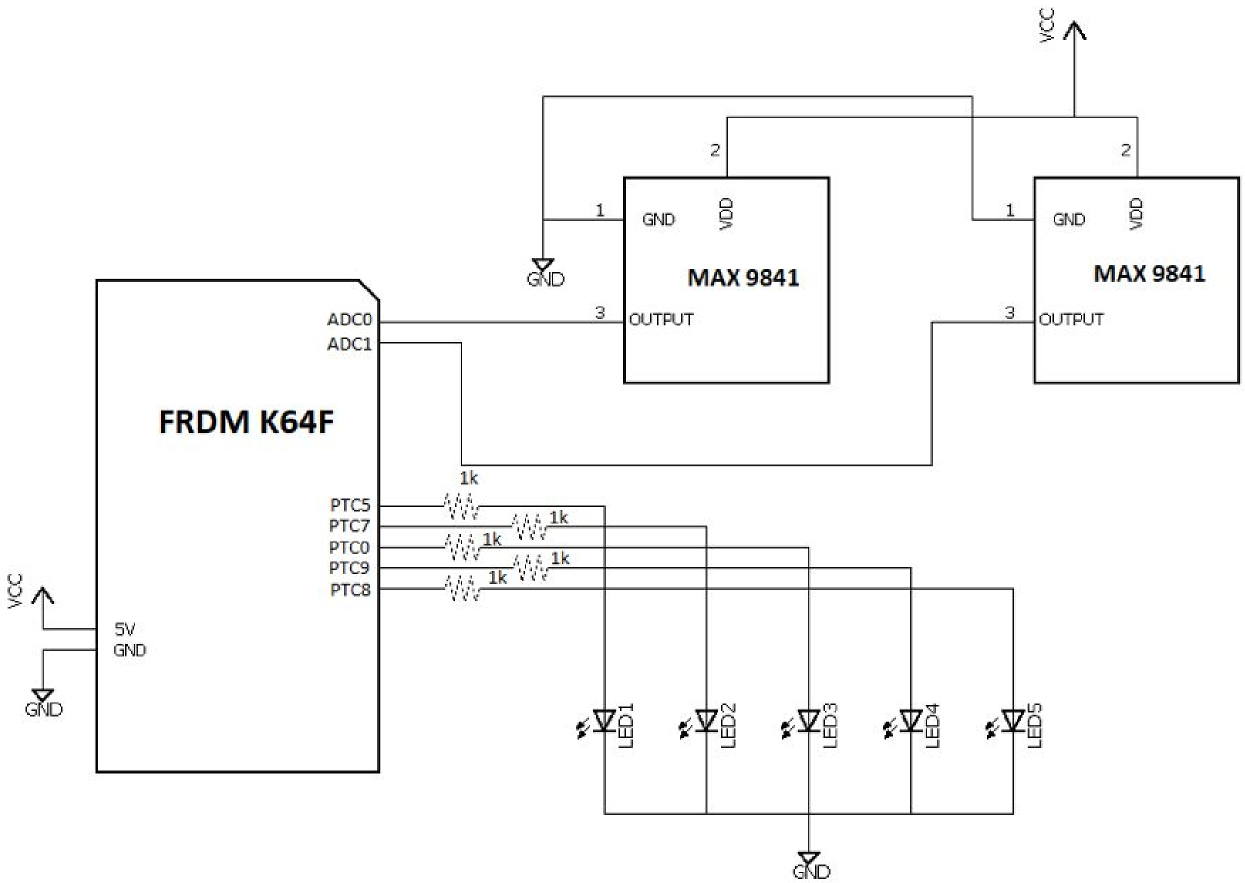

Figure 5.4.1: System Schematics

As seen the figure 5.4.1 above, following are detailed specifications of the components that we have used in our project:

1. **MAX 9814 Microphone:** Max 9814 is a cost efficient, high quality microphone with an inbuilt amplifier having automatic gain control (AGC). 2 MAX 9814 Microphones have been used in the project for sound localization. Input to this microphone is amplified in three stages. First, it is passed through a low noise preamplifier with a gain of 12dB. Second, it passes through a variable gain amplifier controlled by AGC. Lastly it passes

through an output amplifier with a fixed gain of 8db, 18db and 20db programmed through a single tri level logic unit. Features of MAX9814 are as follows:

- Automatic gain control.
- Low power device (2.7 to 5.5V).
- Low power shutdown mode.
- Low noise microphone bias of 2V.
- Temperature range: -40°C to 85°C.

2. **FRDM K64F Freedom Module:** It is a low cost, power efficient with ARM Cortex M4 32 bit core with DSP instructions. Following are the specifications of this module:
    - 1MB Flash memory, 256KB RAM with 120MHz clock.
    - 100 Low profile Quad Flat Package.
    - Flexible power supply options-OpenSDAv2 USB.
    - Kinetis K64 USB and External source.
    - Ethernet connectivity onboard.
    - 12-bit DAC.
    - 16-bit ADC.
    - 68 GPIO pins.

3. **LED Bulbs:** 5 Green LED bulbs have been used in the design to indicate the direction of incoming sound. It has a forward voltage of 3-3.2V and a current of 20mA.

An example of the code to calculate the distance using the formula above is as follows:

```
time_Delay = sampling_T*(maxVal_Index - input_len);
theta = (343*time_Delay/distance)*180/3.14;
```

Where `maxVal_Index` is the index of maximum value in the correlation array, `samplint_T` is the sampling time period, `input_len` is the length of input sequence and distance is the distance between microphone array.

6.0 INTEGRATION AND TESTING

6.1 Integration Plan

The items tested consist of the integration of the software code modules developed as well as the hardware components. For testing software code modules, we choose the middle-out approach whereas for testing the hardware components of the subsystem, we choose the top-down approach. The integration sequence is described below, while the individual tests are in sections 6.2 and 6.3.

6.1.1 Software Integration Sequence

Although we did not do any explicit unit tests on our software, we did incremental tests of the components as we built the software to ensure all of the modules were working together as they should. This helped us remove defects early and focus on future integrations.

6.1.2 Hardware Integration Sequence

The integration approach chosen for this system was a top-down approach. This means that the designer has an overall vision of what the final system must do, and the problem is partitioned into components, or subsystems that work together to achieve the overall goal of the system for this project. Then each subsystem is successively refined and partitioned as necessary.

In the case of our Emergency Vehicle Detection Using Microphone Array and Sound Localization project, the overall objective was determined as aforementioned in the previous sections; the major components are defined such as the NXP FRDM-K64F, MAX9814 Microphone Modules, and Light-Emitting Diodes.

Before we integrated the various components of the subsystem, we performed individual testing of each hardware component. This way the project was built down starting from the top level because this method was more suitable, where it is unlikely that bringing together pieces in an ad-hoc fashion would have successfully solved the problem. This integration test, as described, is at the component level in section 6.2.

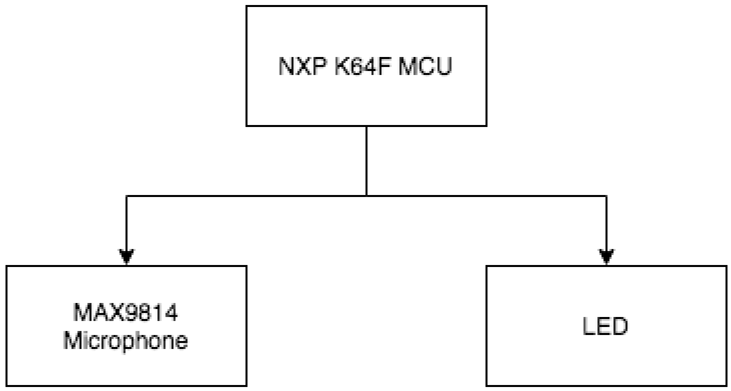

Figure 6.1.2: Hardware Integration Order.

6.1.3 Full Hardware-Software Integration Sequence

The integration approach chosen for this system was a top-down approach. This means that the designer has an overall vision of what the final system must do, and the problem is partitioned into components, or subsystems that work together to achieve the overall goal of the system for this project. Then each subsystem is successively refined and partitioned as necessary.

Before we integrated various software components as aforementioned in section 6.1.1, we assembled all hardware components as per section 6.1.2 because integration of software build(s) depends on the hardware modules. In particular, all the hardware modules had to function properly before we could integrate software modules with hardware modules.

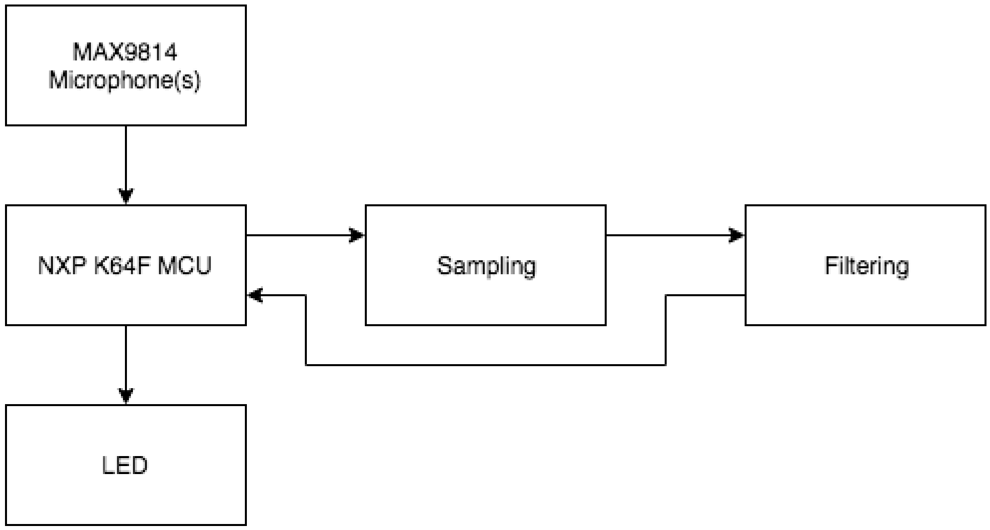

Figure 6.1.4: Full Hardware-Software Integration Order

6.2 Unit Testing - Hardware

Test Writer: Priyank Kalgaonkar.						
Test Case Name:		Test MAX9814 Microphone(s).			**Test ID #:**	SL-UT-01
Description:		This test shall be performed on the MAX9814 using a microphone amp. No microcontroller or programming required. 3AA batteries, a headphone jack, a breadboard, and a 1 to 100uF electrolytic capacitor to protect your headphones from the DC bias voltage.				
Tester Information						
Name of Tester:		Arjun and Kavya			**Date:**	11/30/2019
System Ver:		N/A			**Time:**	12:30 PM
Setup:		Connect the Gain pin to VDD. Connect the OUT pin to a 1uF-100uF capacitor. Connect the Left and Right pins together to the negative side of the capacitor. Make VDD to positive and GND to negative of power supply.				
Step	**Action**	**Expected Result**	**Pass**	**Fail**	**N/A**	**Comments**
1	Try listening to sounds on the headphones.	Should notice a strange effect (similar to echo) where you can hear people from further away than your hearing.	✓			Both microphone modules work as intended.
	Overall test result:		✓			Modules have passed the unit tests.

Table 6.2.1: Unit Test for Microphone Modules

Test Writer: Priyank Kalgaonkar.						
Test Case Name:		Test Light-Emitting Diodes (LED) Bulbs			**Test ID #:**	SL-UT-02
Description:		This test shall be performed on five LED bulbs.				
Tester Information						
Name of Tester:		Arjun and Kavya			**Date:**	11/30/2019
System Ver:		N/A			**Time:**	12:45 PM
Setup:		Test all the LED bulbs by supplying 4.5V to 5V using a Triple Output Power Supply. Check polarity of LED bulbs during connection.				
Step	**Action**	**Expected Result**	**Pass**	**Fail**	**N/A**	**Comments**
1	Connect the LED bulb to a power supply.	LED bulbs show glow.	✓			All LED bulbs are working.
	Overall test result:		✓			No issues were found.

Table 6.2.2: Unit Test for LED Bulbs

Test Writer: Priyank Kalgaonkar.						
Test Case Name:		Test NXP FRDM K64F Microcontroller			**Test ID #:**	SL-UT-03
Description:		This test shall be performed to check the proper functioning of the K64F MCU.				
Tester Information						
Name of Tester:		Arjun and Kavya			**Date:**	11/30/2019
System Ver:		N/A			**Time:**	1:00 PM
Setup:		Connect the K64F MCU to a PC using K64F's OpenSDAv2 USB port (J26).				
Step	**Action**	**Expected Result**	**Pass**	**Fail**	**N/A**	**Comments**
1	Hold the Reset button on K64F MCU while plugging in the cable.	Upon successful connection, green LED should glow on the K64F MCU and should show as 'Bootloader' in Windows Explorer.	✓			Show's Bootloader in Windows Explorer. Green LED lights up.
	Overall test result:		✓			K64F MCU powers on.

Table 6.2.3: Unit Test for NXP FRDM K64F MCU

6.3 Integration and Testing - System Acceptance Test

Test Writer: Priyank Kalgaonkar.						
Test Case Name:		Sound localization system final test			**Test ID #:**	SL-AT-01
Description:		Test the sound capture, sampling and filtering process on the final system.				
Tester Information						
Name of Tester:		Arjun, Kavya, Priyank			**Date:**	12/06/2019
System Ver:		Sound Localization 1.1			**Time:**	05:05 PM
Setup:		This test shall be performed on the final system after the NXP FRDM K64F is powered on, software from Keil MDK is flashed to it and all external connections to microphones and LED bulbs are properly made as planned. No special setup is required.				
Step	**Action**	**Expected Result**	**Pass**	**Fail**	**N/A**	**Comments**
1	Play an ambulance siren on cell phone and slowly move from left to right.	LED bulbs should glow and indicate which direction the sound is coming from.	✔			LEDs glow as expected.
2	Play random music on level 5 out of 10 on cell phone.	LED bulbs should not glow since they do not correspond to the high pitched frequency of emergency vehicles' sirens.		✔		Problem: Sound captured by an array of microphones is not being filtered. Root Cause: Due to hardware limitations of the K64F MCU, the MCU is giving memory errors, possibly due to lack of memory to filter sound signal in real-time.
	Overall test result:					Sampling of the signals passed. However, the MCU is unable to handle Bessel filtering technique.

Table 6.3.1: System Acceptance Test

This test covers all of the functionality of the Emergency Vehicle Detection Using Microphone Array and Sound Localization system. Step 1 verifies that the sampling technique is being performed by the algorithm design by us. Steps 2 verifies the filtering technique works without as intended. However, during the testing process and after troubleshooting the issue, we concluded that limited memory is one of the hardware limitations of the MCU.

6.4 Final Functional Test

The final functional tests were completed after software was fully developed, all of the hardware had been assembled, and the two had been integrated together. The tests that were carried out are outlined as follows:

1. <u>Action</u>: **Reset the board to see if indicators work proper.**

<u>Expected Result</u>:All indicators should glow simultaneously for a small time duration .

2. <u>Action</u>: **Producing low frequency voices .**

<u>Expected Result</u>: No indicators should glow.

3. <u>Action</u>: **Test with ambulance siren sound from different angles.**

<u>Expected Result</u>: LEDs corresponding to different angles should glow.

These tests are intended to confirm that the system has been successfully integrated, and now is ready for full system test by engineering. Section 6.4 provides the test results of the integration and testing carried out as aforementioned.

7.0 ECONOMICS

The system development life cycle of our Emergency Vehicle Detection Using Microphone Array and Sound Localization system project is composed of a number of clearly defined work phases, such as: brainstorming, design, implementation, testing and delivering the final product. Like anything that is prototyped, the system development life cycle aims to produce high quality systems to meet professional engineering standards, based on the need for an easy-to-use product within scheduled time frame and budget estimates.

7.1 Cost of Material

The budget includes costs for all hardware requirements for building the prototype including the NXP FRDM-K64F MCU, sensors and other related components. Following is a table of cost of individual components:

Component	Price/Component	Qty. Purchased	Total
NXP FRDM K64F	$59.37	1	$59.37
MAX9814	$10.45	2	$20.90
LED Bulb	$1.07	5	$5.35
Total Billable Material		8	$85.62

Table 7.1.1: Cost of Billable Material

7.2 Engineering Labor Hours

In determining overall cost estimates, engineering labor hours also need to be taken into consideration. Table 7.2.1 gives an estimate on the labor hours for each activity below:

Activity	(In Hours)
Meeting 4 days/week (3 group members)	48
Critical Design Review	5
Software Development	25
Hardware	5
System Integration and Testing	12
Finalize Implementations	15
Total Hours	110

Table 7.2.1 Table of Engineering Labor (In Hours)

7.3 Specialized Facilities and Resources

Implementation of this project requires resources and a location where these resources are available. Table 9.4 shows the specialized facilities that was used for research and development of the project.

Facility	Location	Resources	Use
SL111: ECE538 Lab	Engineering Science & Technology (SL) Building at IUPUI.	Hardware such as voltmeters, oscilloscopes, soldering station, wires and breadboards.	Testing purposes.

Table 7.3.1 Table of Specialized Facilities and Resources

8.0 PROJECT MANAGEMENT

8.1 Pre-Defined Schedule (Gantt Chart)

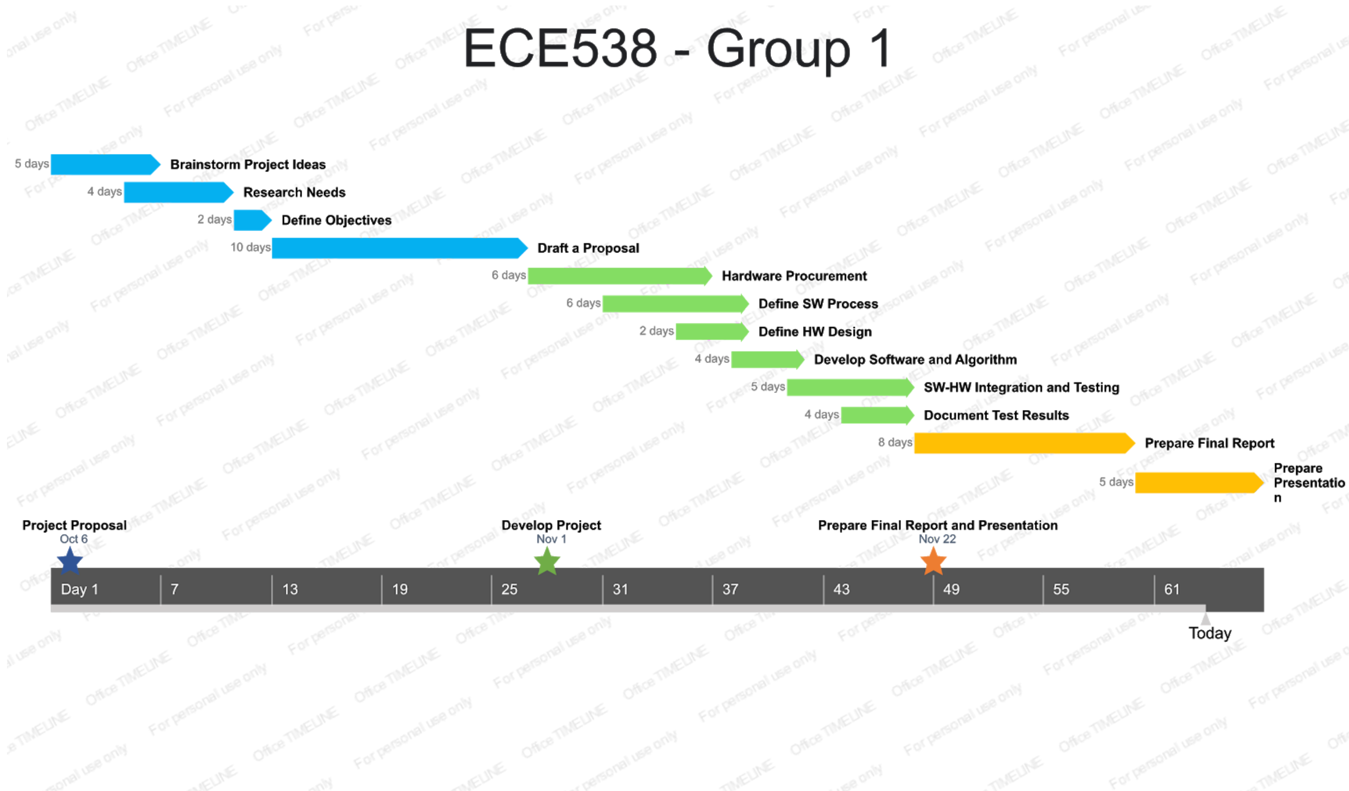

Figure 8.1.1: Project Schedule - Gantt Chart

8.2 Risk Management

There were several risks identified with our project. The first major risk was that various project deadlines such as development, integration, testing, and final building of the prototype would not be met. This was mitigated by creating a detailed plan including a Gantt chart as shown above, and setting up efficient communication channels between the team members to manage our time correctly and efficiently.

Another risk was not having the critical components available in a timely fashion. Shipments could have easily been delayed, and parts could get damaged during shipping. This risk was avoided by organizing and selecting all the components necessary for our design as early as possible. Once this was complete, all the components were ordered at once. This gave enough time for complications and troubleshooting, which included parts that did not work. At first, one of the MAX9814 sensor was dead on arrival (DoA). Faulty components had to be replaced from the seller with the same new components. Again, we avoided real complications by using good time management.

Additionally, our group ran the risk of not complying with specific standards for hardware components. We were able to avoid this risk by purchasing all hardware components off the shelf.

9.0 CONCLUSION

The objective of the Emergency Vehicle Detection Using Microphone Array and Sound Localization system is to inform motorists, cyclists and pedestrians on the road, at an intersection, the direction of an approaching emergency vehicle(s) and make way beforehand for it to pass by safely and quickly .

The design of this system utilizes digital signal processing techniques such as sampling and filtering to process the sound captured by an array of microphones to provide a real-time output. This will result in vehicles and people around the intersection and/or where this system is implemented, be better informed of the exact direction of the approaching emergency vehicle and make way beforehand for it to pass by safely and quickly.

The Emergency Vehicle Detection Using Microphone Array and Sound Localization prototype requires the system to process the sound captured by microphones in real-time, be easy to implement and maintain, and low cost. The requirement was partially fulfilled since FRDM-K64F MCU is powered by an ARM Cortex M4 core running up to 120MHz, the sound should be processed with acceptable delay. The MCU was able to sample sound in real-time but failed to filter the sound due to limited amount of memory (RAM) available onboard. Regarding the requirement of cost of implementation and maintenance, since we propose to use an array of small microphones, a microcontroller and dedicated LED bulbs/display, cost of implementation can be kept at a minimum.

In conclusion, the Emergency Vehicle Detection Using Microphone Array and Sound Localization prototype utilizes hardware and Digital Signal Processing techniques such as sampling and filtering to provide output by glowing corresponding LEDs in real-time. Motorists and pedestrians will be better altered beforehand of the direction of the approaching emergency vehicle(s), making it safe for both, people on the road and inside the emergency vehicle(s).

10.0 REFERENCES

1. Regulatory Assessment. (n.d.). Retrieved from https://www.access-board.gov/guidelines-and-standards/streets-sidewalks/144-public-rights-of-way-guidelines/regulatory-assessment.
2. FRDM-K64F. (n.d.). Retrieved from https://os.mbed.com/platforms/FRDM-K64F/.
3. FRDM-K64F: Freedom Development Platform for Kinetis® K64, K63, and K24 MCUs. (n.d.). Retrieved from https://www.nxp.com/design/development-boards/freedom-development-boards/mcu-boards/freedom-development-platform-for-kinetis-k64-k63-and-k24-mcus:FRDM-K64F.
4. Industries, A. (n.d.). Electret Microphone Amplifier - MAX9814 with Auto Gain Control. Retrieved from https://www.adafruit.com/product/1713.
5. Environmental Certifications. (n.d.). Retrieved from http://www.nxp.com/company/our-company/about-nxp/corporate-responsibility/environmental-compliance-organization/environmental-certifications:ENV_PBROAD.
6. "Certification: Espressif Systems." Certification | Espressif Systems, www.espressif.com/en/certificates.
7. Standards Compliance. (n.d.). Retrieved from http://www.sparkfun.com/compliance.
8. Keil. (n.d.). Retrieved from https://www2.keil.com/mdk5/docs.
9. Acentech. (2019). *Cutting Through the Noise: Sirens and Emergency Vehicle Detection - Acentech*. [online] Available at: https://www.acentech.com/blog/cutting-through-the-noise-sirens-and-emergency-vehicle-detection/ [Accessed 8 Dec. 2019].

11.0 APPENDICES

11.1 Appendix A - Component Pinouts

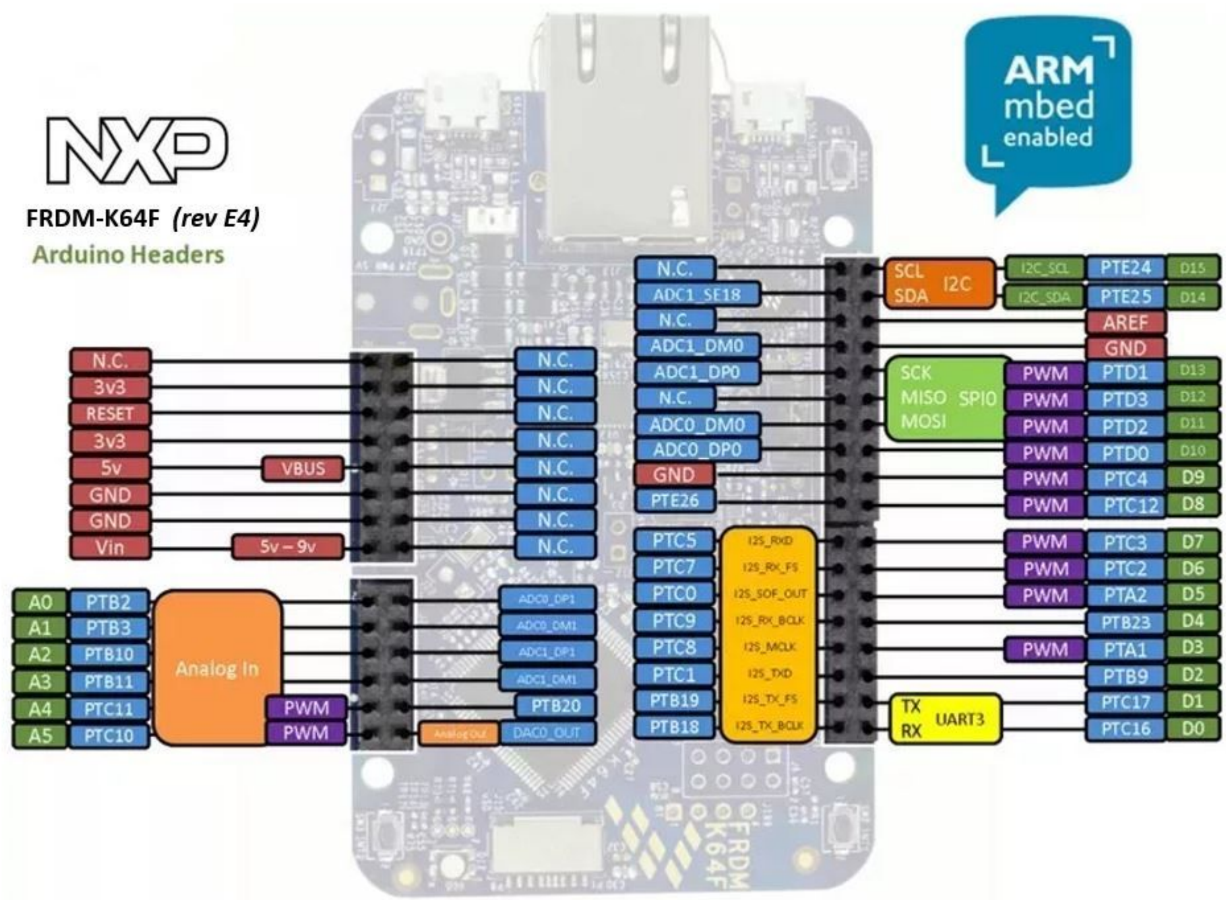

Figure 11.1.1: NXP FRDM K64F Header Pinout Diagram[2]

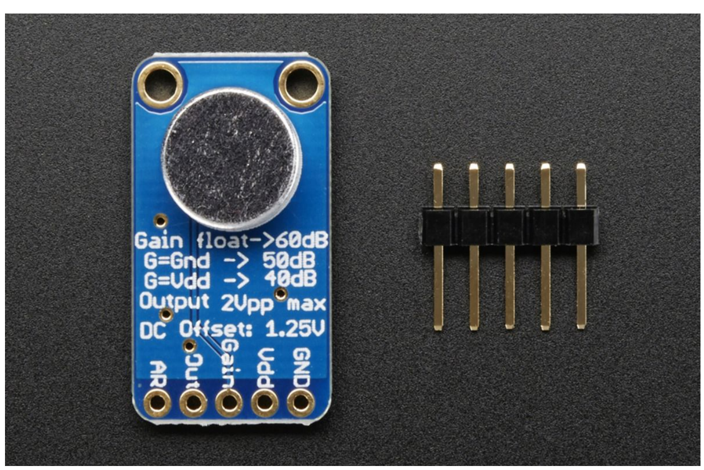

Figure 11.1.2: MAX9814 Microphone Pinout[3]

11.2 Appendix B - Source Code

```
#include <fsl_device_registers.h>
#include "ADC.h"
#include <stdlib.h>
#include "board.h"
#include <math.h>
#define ARM_MATH_CM4
#include <arm_math.h>
#include "fsl_debug_console.h"

uint32_t maxVal_Index;
unsigned int INPUT_LEN = 2000;
unsigned int Correlation_length = 3999;
float32_t maxVal;
float32_t time_Delay;
float32_t sampling_T = 1.7644E-6f;
float32_t theta;
float32_t dist = 0.05f;

void AudioAcquire(float32_t * mic0, float32_t * mic1);
void printTerminal(float32_t* data);
void LED_init();
void LED_out(float32_t theta);

int main()
{
    hardware_init();
    initADC();
    LED_init();

    //init data arrays
    float32_t *mic0Data = (float32_t*)
    malloc(INPUT_LEN*sizeof(float32_t));
    if (!mic0Data)
    {
        debug_printf("memory allocation failed!");
        return -1;
    }

    float32_t *mic1Data = (float32_t*)
    malloc(INPUT_LEN*sizeof(float32_t));
    if (!mic1Data)
    {
        debug_printf("memory allocation failed!");
        return -1;
    }
```

```c
    float32_t *correlationData = (float32_t*)
    malloc(Correlation_length*sizeof(float32_t));
    if (!correlationData)
    {
        debug_printf("memory allocation failed!");
        return -1;
    }

    while(1)
    {
        AudioAcquire(mic0Data, mic1Data);
        arm_correlate_f32(mic0Data, INPUT_LEN, mic1Data, INPUT_LEN,
        correlationData);
        arm_max_f32(correlationData, Correlation_length, &maxVal,
        &maxVal_Index);
        debug_printf("%d %d\r\n", (unsigned)maxVal,
        (unsigned)maxVal_Index);
        time_Delay = sampling_T*((int)maxVal_Index - (int)INPUT_LEN);
        theta = asinf(343*time_Delay/dist)*180/3.14159265;
        debug_printf("angle: %d \r\n", (int)theta);
        LED_out(theta);
    }
}

void LED_init()
{
    SIM->SCGC5 |= SIM_SCGC5_PORTC_MASK;
    PORTC->PCR[8] = PORT_PCR_MUX(001);
    PTC->PDDR |= (1<<8);
    PORTC->PCR[9] = PORT_PCR_MUX(001);
    PTC->PDDR |= (1<<9);
    PORTC->PCR[0] = PORT_PCR_MUX(001);
    PTC->PDDR |= (1<<0);
    PORTC->PCR[5] = PORT_PCR_MUX(001);
    PTC->PDDR |= (1 << 5);
    PORTC->PCR[7] = PORT_PCR_MUX(001);
    PTC->PDDR |= (1<<7);
    PTC->PTOR |= 0x00003A1;
    DelayFunction();
    PTC->PTOR |= 0x00003A1;
}

void LED_out(float32_t theta)
{
    PTC->PCOR |= 0xFFFFFFFF;
    if(theta < -50)
      PTC->PSOR = (1 << 8);
```

```c
    else if(theta < -20)
      PTC->PSOR = (1 << 9);
    else if(theta < 20)
      PTC->PSOR = (1 << 0);
    else if(theta < 50)
      PTC->PSOR = (1 << 7);
    else if (theta < 90)
      PTC->PSOR = (1 << 5);
}

void AudioAcquire(float32_t* mic0, float32_t * mic1)
{
    int i;
    float32_t mic0Mean;
    float32_t mic1Mean;

    for (i = 0; i < INPUT_LEN; i++)
    {
      ADC0_SC1A = 0 & ADC_SC1_ADCH_MASK;
      while(ADC0_SC2 & ADC_SC2_ADACT_MASK);
      while(!(ADC0_SC1A & ADC_SC1_COCO_MASK));
      mic0[i] = (((float32_t)ADC0_RA)/1000);

      ADC1_SC1A = 0 & ADC_SC1_ADCH_MASK;
      while(ADC1_SC2 & ADC_SC2_ADACT_MASK);
      while(!(ADC1_SC1A & ADC_SC1_COCO_MASK));
      mic1[i] = (((float32_t)ADC1_RA)/1000);
    }
    arm_mean_f32(mic0, INPUT_LEN, &mic0Mean);
    arm_mean_f32(mic1, INPUT_LEN, &mic1Mean);
    arm_offset_f32(mic0, -mic0Mean, mic0, INPUT_LEN);
    arm_offset_f32(mic1, -mic1Mean, mic1, INPUT_LEN);
}

void printTerminal(float32_t * data)
{
    int i;
    for (i = 0; i < Correlation_length; i++)
      debug_printf("%d\r\n", (unsigned)data[i]);
}
```