EMBEDDED SYSTEMS 2

CSE 5342 – Spring 2024

Angle of Arrival on Continuous Audio

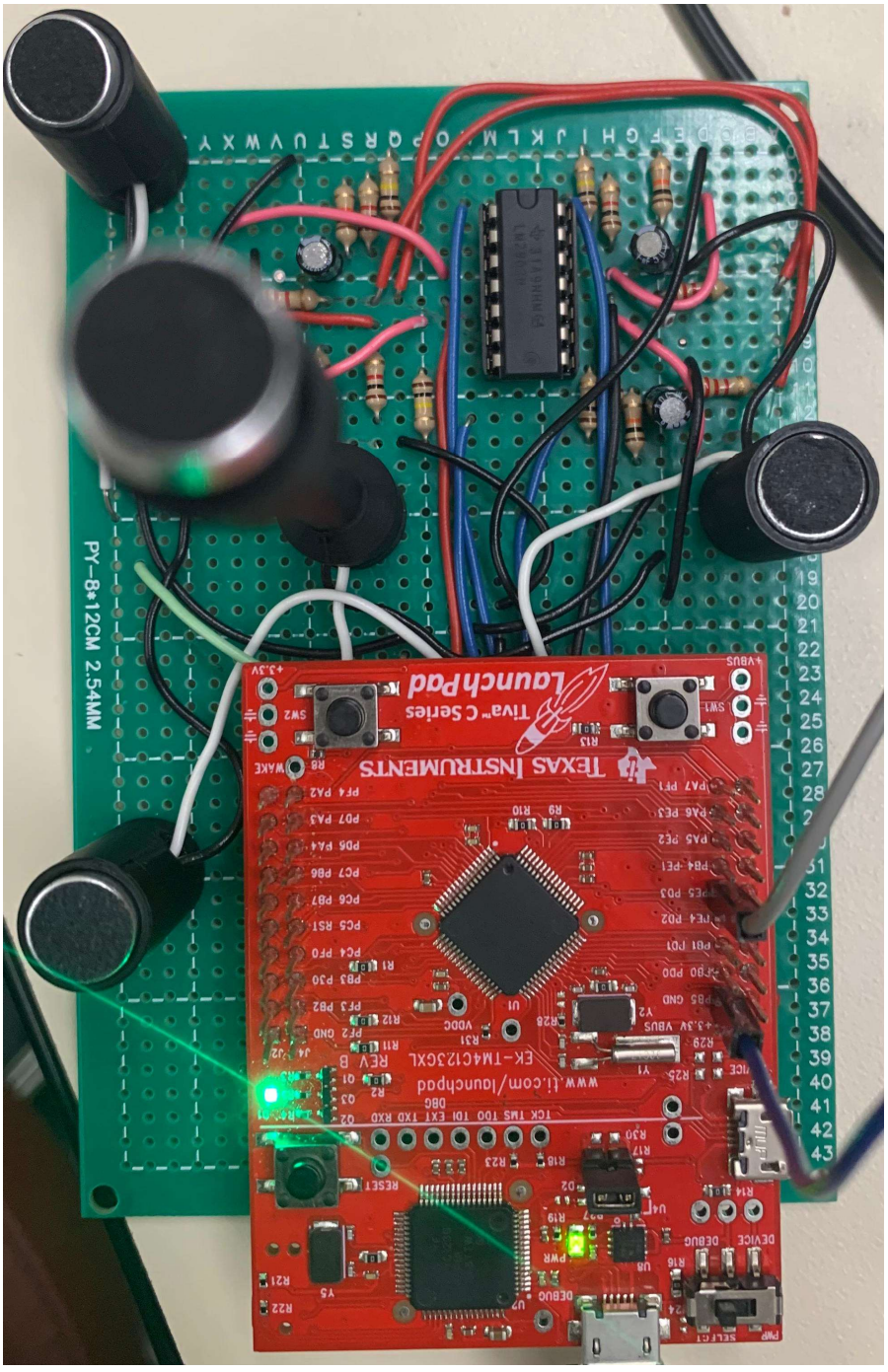
By,

Padmanathan Kannan (1002011190)

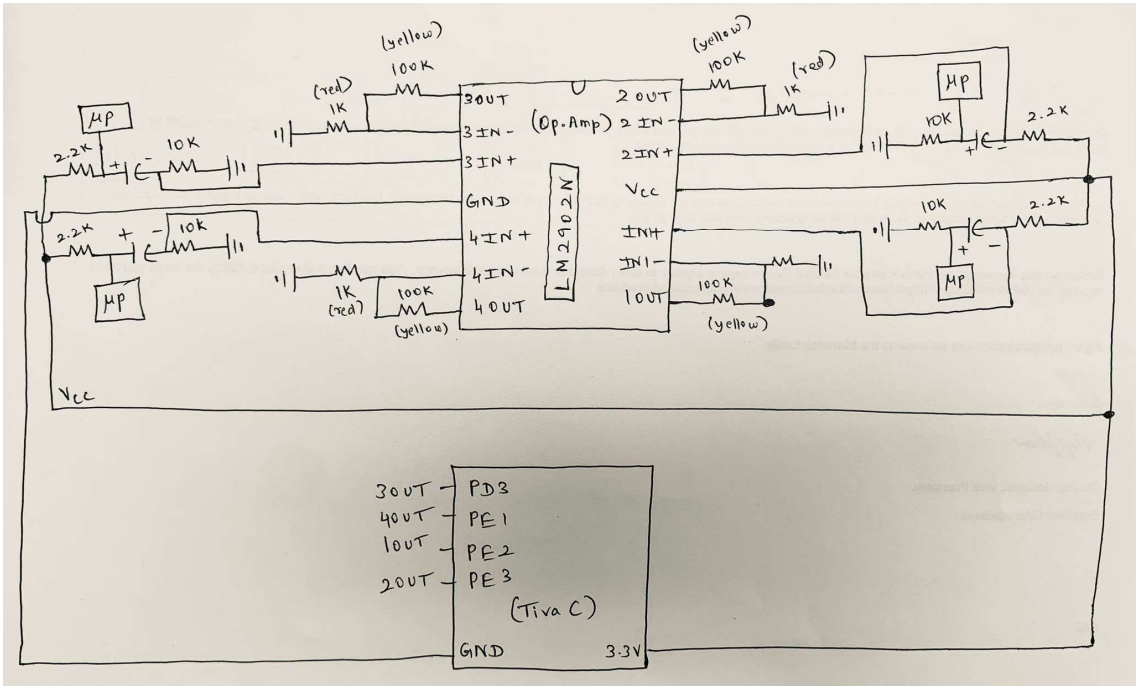
Introduction:

The objective of this project is to design a low cost and low power device to solve the angle of arrival problem with 4 electret microphones connected to op amp. The op amp output is directed to the ADC module of ARM M4F microcontroller that samples the signal at 1 million samples per second. The device uses cross-correlation algorithm to determine the angle of arrival of sound sources and provides angle information via a command line interface. The project also implements DMA to limit the CPU processing load.

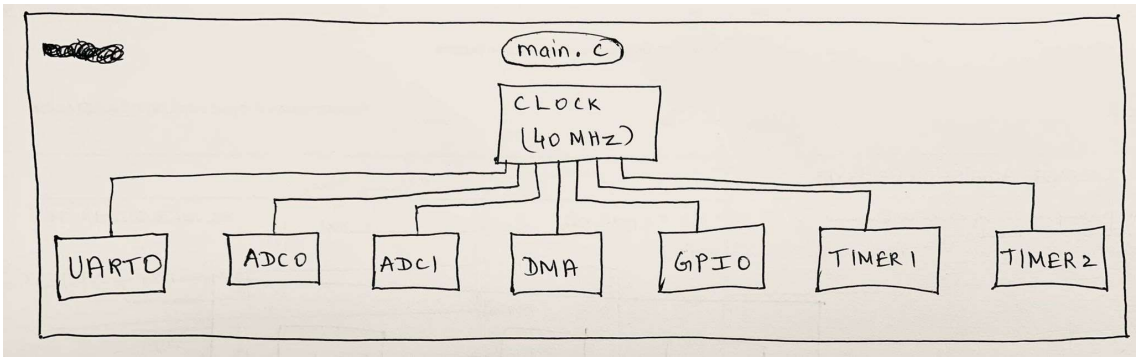
Hardware Setup:



Hardware Block diagram:



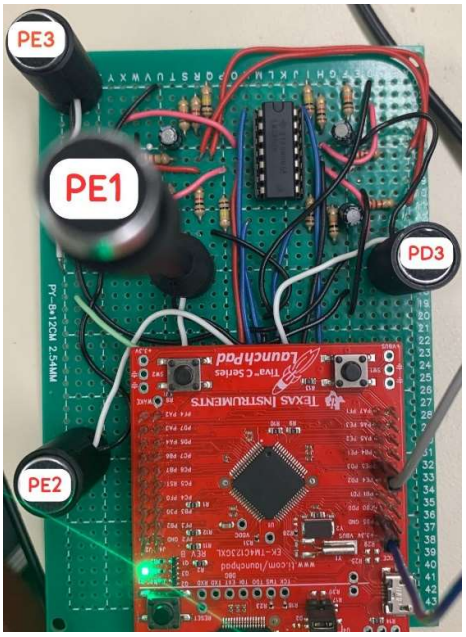
Software Block Diagram:



Peripherals:

Peripheral	Purpose
ADC0 SS1	It was used for Digital Comparator 0,1,2,3 and for trigger purposes.
ADC1 SS1	It was used for DMA and this peripheral is at channel 25 of DMA.
GPIO	It was used to configure AIN0, AIN1, AIN2 ,AIN4.
UART 0	It was used for putty commands.
Timer1	It was used for the purpose of valid event.
Timer2	It was used to implement Holdoff command where it waits for the timer's load time before detecting next sound source.

Design Approach:



The microphones are connected to the AIN2, AIN1, AIN0, AIN4 of the Tiva C Red board. The hardware uses ADC0 SS1(Sample Sequencer) for Digital Comparator logic and event trigger and ADC1 SS1 for DMA.

The primary reason for using digital comparator was to decrease the noise susceptibility and set well defined threshold values. In this project, the microphones threshold values were empirically set based the values received from the ADC modules.

In this project, a "valid event" is when the ADC value is above the threshold value, that is the COMP1 value of the register ADC0_DCCMP0_R (ADC Digital Comparator Range) is crossed. The high band bit in the ADC0_DCCTL0_R (ADC Digital Comparator Control) was set to achieve this functionality. It generates interrupt when COMP1 value is crossed.

Algorithm:

The main objective of the algorithm is to find the theta value. However, there are a lot of steps involved in it.

1) The first step was to extract the samples from the DMA Buffer and stored them in arrays of size 16 exclusively for each of the 4 microphones.

2) Since this project's objective was to find the theta value using the 2D approach, the microphone in the middle (PE1) was not used for the correlation purpose.

3) The cross correlation was found using the sliding window method between the signal pairs PE2-PE3, PE3-PD3, PD3-PE2. The results were stored in 3 arrays of size 31 exclusive for each correlation pair, as the signals were of size 16. The resultant formula is 2*n – 1. In this case, n =16.

4) The time stamps T1, T2, and T3 were found using multiplying it with the time period. In this project, the sampling rate is 1000000. So, the time period = 1/sampling rate. The T1,T2,T3 values decide which microphones received the sound first.

5) The theta formulas for all the microphones are:

THETA_FINAL_1 = (K1 *(T3-T2)) + (K2 * (T3-T2) * (T3-T2));

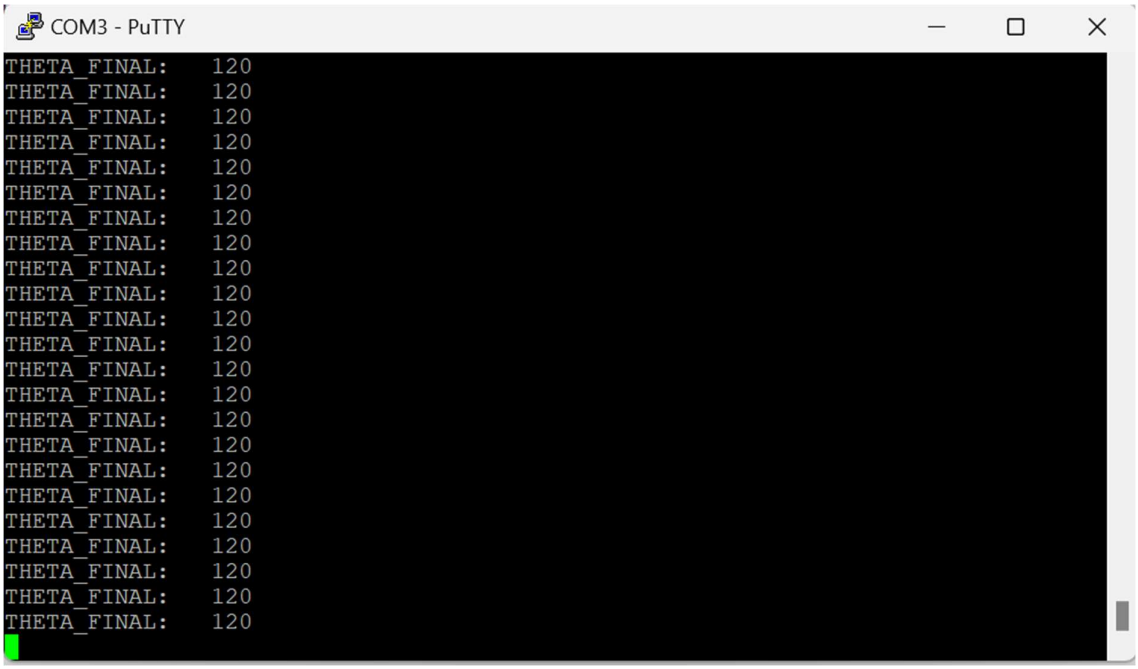
THETA_FINAL_2 = -120 + (K1 *(T3-T2)) + (K2 * (T3-T2) * (T3-T2));

THETA_FINAL_3 = 120 + (K1 *(T3-T2)) + (K2 * (T3-T2) * (T3-T2));

6) The K1,K2 values were assumed to be 0.5 in this project.

Readings:

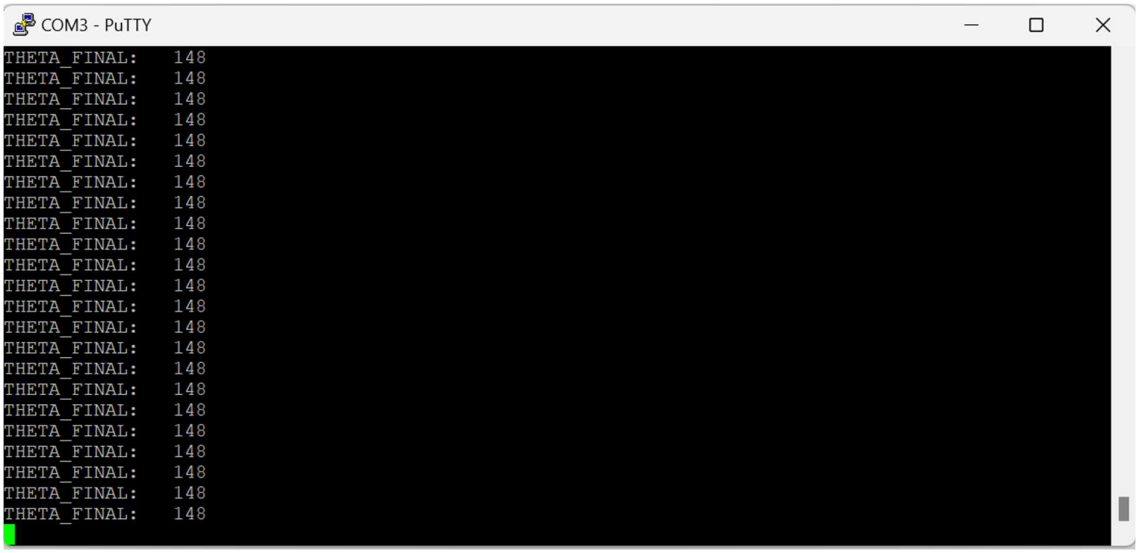
1) In this case the output angle was read to be 120 degrees.



Expressions

Expression	Type	Value	Address
ping_buffer	unsigned short[64]	[2640,228,2643,252,2632...]	0x20001400
pong_buffer	unsigned short[64]	[2646,180,2635,191,2642...]	0x20001480
mic_pe1_samples	unsigned int[16]	[1460,1680,1951,2301,2624...]	0x200016B4
mic_pe2_samples	unsigned int[16]	[21,55,79,116,160...]	0x200016F4
mic_pe3_samples	unsigned int[16]	[1582,1788,2112,2555,2639...]	0x20001734
mic_pd3_samples	unsigned int[16]	[47,44,85,175,180...]	0x20001674
mic_pe2_pe3_xcorr	unsigned int[31]	[0,9,106,7,0...]	0x2000157C
mic_pe3_pd3_xcorr	unsigned int[31]	[41,0,0,17,16...]	0x200015F8
mic_pd3_pe2_xcorr	unsigned int[31]	[41,59,0,6,37...]	0x20001500
T1	float	1.0	0x200017C0
T2	float	1.0	0x200017C4
T3	float	1.0	0x200017C8
THETA_FINAL_1	unsigned short	0	0x200017DE
THETA_FINAL_2	unsigned short	65417	0x200017E0
THETA_FINAL_3	unsigned short	120	0x200017E2
max_value_pe2_pe3	unsigned int	83777	0x200017B0
max_value_pe3_pd3	unsigned int	62201	0x200017B4
max_value_pd3_pe2	unsigned int	62014	0x200017B8
max_index_pe2_pe3	unsigned int	1	0x200017A0
max_index_pe3_pd3	unsigned int	1	0x200017A4
max_index_pd3_pe2	unsigned int	1	0x200017AC
THETA_FINAL	unsigned short	120	0x200017DC
Add new expression			

2) In this case, the angle was read to be 148 degrees.



T1	float	4.0	0x200017C0
T2	float	2.0	0x200017C4
T3	float	2.0	0x200017C8
THETA_FINAL_1	unsigned short	15	0x200017DE
THETA_FINAL_2	unsigned short	0	0x200017E0
THETA_FINAL_3	unsigned short	148	0x200017E2
max_value_pe2_pe3	unsigned int	12704	0x200017B0
max_value_pe3_pd3	unsigned int	15552	0x200017B4
max_value_pd3_pe2	unsigned int	13517	0x200017B8
max_index_pe2_pe3	unsigned int	4	0x200017A0
max_index_pe3_pd3	unsigned int	2	0x200017A4
max_index_pd3_pe2	unsigned int	2	0x200017AC
THETA_FINAL	unsigned short	148	0x200017DC

Design Failings:

The angle readings had an error of at least 30 degrees and up to 50 degrees. The below mentioned ways can resolve the error:

1) The 4th microphone present in the middle can be used to increase the accuracy of the readings. This methodis basically the 3D method of solving Angle of Arrival problem.

2) The sampling rate can be increased from 1 Msps to 2 Msps by using another ADC module that samples at 1 Msps. A single ADC module supports sampling rate only up to 1 Msps.

3) By adding a bypass capacitor to the hardware, we can further reduce the noise interference.

Conclusion:

Thus, a low cost and low power device to solve the angle of arrival problem, was designed and solved using the 2D method with 3 microphones on a ARM M4F platform.