Acoustic Distance Measurement using Nexys A7-100T

Aishwarya Manickam Rameshbabu
9926445
UC Santa Barbara
Santa Barbara, CA, 93106
a_manickamrameshbabu@ucsb.edu

Radha Vaidya
9866765
UC Santa Barbara
Santa Barbara, CA, 93106
radhavaidya@ucsb.edu

Abstract

This report describes designing an acoustic distance measurement system using various functionalities of the Nexys A7-100T board. The most common method for measurement of distance is the time of arrival method. In this project, we explore the implementation of this method using the phase shift conversion for calculation of distances between 1-10 feet.

1 Introduction

Distance measurement using acoustics is an age-old concept. Recently, such kind of contact-less distance sensing finds application in smart-cars industry. For applications in such an industry as well as military/airforce fields, accuracy of distance measurement is crucial. In this project, we have implemented an acoustic distance sensor using the phase-shift method of distance measurement as it has proven to improve accuracy in such calculations[1].

In this method, we have 2 modules, the transmitter and receiver. In our project, both modules exploit different functionalities of the Nexys A7-100T board. The transmitter, which in our case is a speaker attached to the mono output of the the Nexys A7-100T, outputs a short acoustic tone of desired frequency. The receiver, which is the on- board microphone, is always listening. When the microphone hears the acoustic tone the system measures the time difference between the moment the acoustic tone is generated and the moment the microphone receives the start of the acoustic tone. Multiplying the time difference by the speed of sound provides an estimate of the distance. This time difference is know as time of flight or time of arrival, and is determined by using phase difference between the sent and received signal in our system.

The time of arrival technique relies on accurate identification of the start of an acoustic pulse which can be difficult in noisy environments. This is why the use of the phase difference between the transmitted and received signals is an elegant way of determining the time difference required to compute the distance [2].

2 Hardware Requirements

The Nexys A7-100T board acts as a complete distance sensor module. This is because both the transmitter and receiver modules of the sensor are designed by using different functionalities of the Nexys A7-100T board. In addition to the Nexys A7-100T board, we use various other peripherals to build this distance sensor.

Our input peripherals are the microphone and push buttons on the board. The output peripherals are a speaker connected to the mono output and an LCD connected to a PMOD port.

Given below is a picture of the Acoustic Distance Measurement System, with all the peripherals used followed by the description of the peripherals.

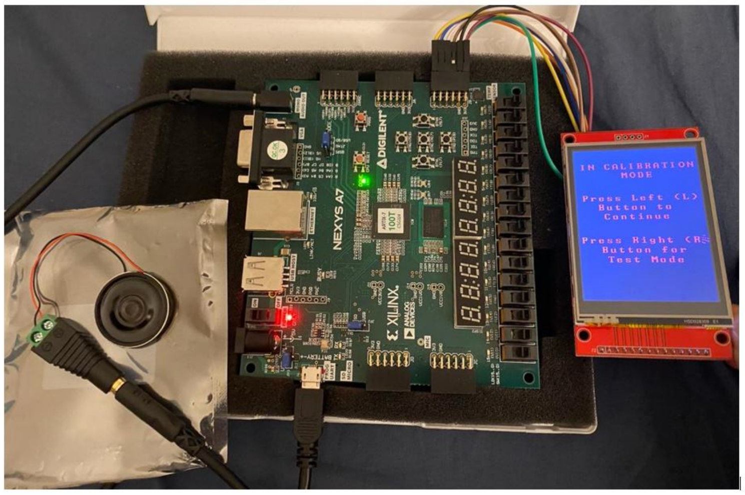
NEXYS A7
DIGILENT
XILINX
IN CALIBRATION MODE
Press Left (L) Button to Continue
Press Right (R Button for Test Mode

Figure 1: Picture of Complete Setup

1. MEMS Microphone

   We make use of the on-board MEMS omnidirectional microphone in our receiver module. The microphone uses an Analog Device ADMP421 chip which has a high signal to noise ratio (SNR) of 61dBA and high sensitivity of -26 dBFS. It also has a flat frequency response ranging from 100Hz to 15 kHz. The digitized audio is output in the pulse density modulated (PDM) format[3].

   We use this microphone to receive the signals and proceed with further processing.

2. Push Buttons

   We make use of the push buttons available on the board to navigate our UI. The push buttons essentially serve as signals to our QP Nano based -HFSM, which will be discussed in a later section.

3. Speaker connected to Mono Output

   We connect an 8Ω mini loudspeaker to the mono output on the board. This jack is driven by a Sallen-Key Butterworth low-pass 4th order filter [3]. The speaker is connected to the mono output using a screw terminal and a mono interconnect cable.

4. LCD Connected to PMOD output

   We connect the LCD to one of the PMOD ports (JB) available on the board using male-female jumper wires. The LCD is the output component of our UI that displays all the instructions as well as the results of our measurements.

3 Design Challenges

1. The MEMS Microphone on the board has frequency response of 100Hz to 15KHz. This restricted us from not being able to operate in the ultrasonic frequency range.
2. We chose 8Ω mini loudspeaker instead of 16Ω after considering the mono audio output constraints.
3. Simultaneous working of speaker and microphone.

4 Methodology

4.1 Approach

First we set up hardware and made the block design. We then proceeded to make the software blocks.Our approach was modular. We primarily have three blocks: PWM Audio Generation, Input via microphone Signal Processing and Distance Measurement, and User Interface blocks. The modular representation of our setup, Block design built on Vivado and detailed description of each of the blocks is given below.

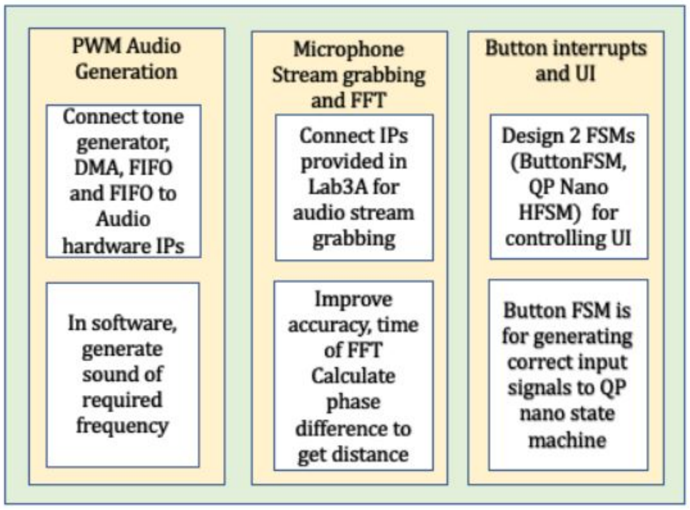
PWM Audio Generation
Connect tone generator, DMA, FIFO and FIFO to Audio hardware IPs
In software, generate sound of required frequency
Microphone Stream grabbing and FFT
Connect IPs provided in Lab3A for audio stream grabbing
Improve accuracy, time of FFT Calculate phase difference to get distance
Button interrupts and UI
Design 2 FSMs (ButtonFSM, QP Nano HFSM) for controlling UI
Button FSM is for generating correct input signals to QP nano state machine

Figure 2: Modular Representation of the Workflow

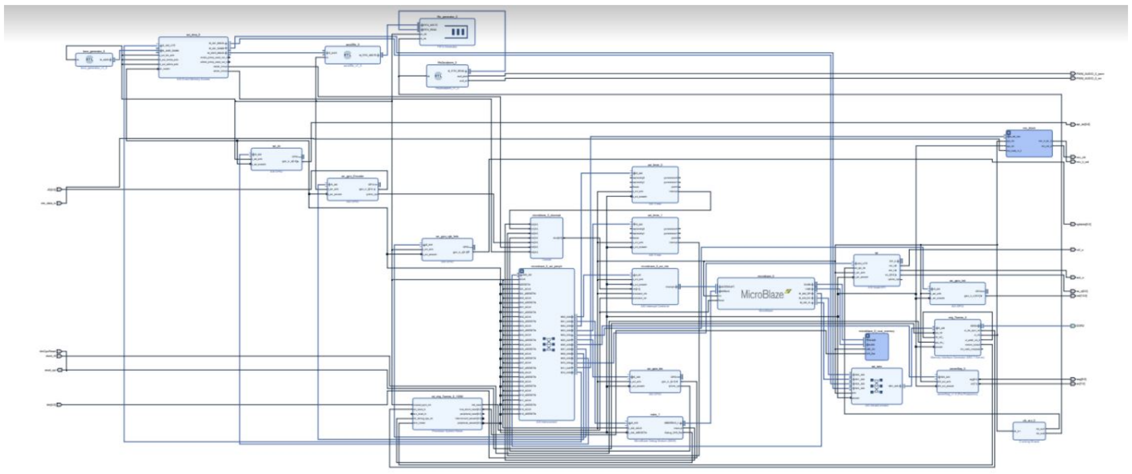

Figure 3: Block Design Built on Vivado

4.2 PWM Audio Generation

For the sound generation and output through the Mono output, we followed the Block design of Digilent Audio Demo[4], and used their IPs as well as portions of their available code after modifying it for our application.

We generate a PWM signal and then amplify and filter it to produce the output signal. A PWM signal is a chain of pulses of varying width. This digital signal is passes through a Sallen-Key Butterworth Low-pass 4th Order Filter to give an analog signal with amplitude proportional to the width of the pulse. A pulse-width modulated (PWM) signal is a chain of pulses at some fixed frequency, with each pulse potentially having a different width. This digital signal can be passed through a simple low-pass filter that integrates the digital waveform to produce an analog voltage proportional to the average pulse-width over some interval (the interval is determined by the 3dB cut-off frequency of the Sallen-Key Butterworth Low-pass 4th Order Filter and the pulse frequency)[3].

The generation of this PWM signal is done in the Tone generator block with a sampling frequency of 96KHz. We use the value to be fed in the accumulator to determine the tone frequency. This is done using the relation:

$$\text{accumulator value} = \frac{2^{\text{accumulator depth}} * \text{Tone Frequency required}}{\text{Audio Sampling Frequency}}$$

Here the accumulator depth is 32 bits, and the sampling frequency used is 96KHz. We then find the accumulator value to obtain a Tone Frequency of 19KHz. The audio data generated is an 8 bit stream.

Then the output of the tone generator is given to the FIFOs and ultimately to the A11 pin using the DMA. The DMA or Direct Memory Access is used to directly connect different modules to memory, bypassing the processor.

Once a signal is generated, we then send focus on receiving this signal in the Input module.

4.3 Microphone Setup

We followed instructions for Lab 3A to set the microphone. The output of the microphone is Pulse Density Modulated (PDM) single byte stream. A decimator block filters the incoming PDM stream to a 48KHz stream. 48 KHz is the sampling frequency we use to compute the FFT for the incoming signal. This stream is then sent to the processor using a stream grabber block. The stream grabber fills up its buffer(size 4096) with the incoming stream. When enough samples are received, the processor polls the grabber to then use the values.

Once we receive the incoming signal in this block, we then proceed to the Signal Processing and Distance Measurement Block.

4.4 Signal Processing and Distance Calculation

This project uses acoustic waves to measure distance. It uses the phase-shift conversion method to determine the flight time which is then used to compute the distance. Time of flight is the time difference from when the sound leaves the source (the speaker setup pictured above) to when it reaches the receiver(microphone on the Nexys A7-100T board). We know that distance can be computed using the following formula:

$$d = v.t$$

Here d is the distance to be computed, v is the speed of sound which we have considered 340 ms^{-1}, t is the time of flight or time of arrival described above. In our implementation, we are making use of phase shift between the sent and received signals to compute this time of flight. We make use of this relation to compute time of flight:

$$\phi = 2\pi f t$$

Where ϕ is the phase difference (in radians) between the signal sent from the loudspeaker set-up and the signal received by the microphone on board and f is the frequency of the signal used.

Combining the above equations together, the final equation we use for distance computing in our model is

$$d = \frac{v.\phi}{2\pi f}$$

There are various methods to determine the phase shift between two signals, we however use the FFT method. The FFT or the Fast Fourier Transform method is a very easy and elegant way of determining the phase shift between signals in the following steps

- Find the product of the sent and received signals
- Find the Fast Fourier Transform of this product signal
- Determine the FFT peak

- Determine phase at the FFT peak by using the following formula:

$$\phi = tan^{-1}\left(\frac{\text{Imaginary part of FFT peak}}{\text{Real part of FFT peak}}\right)$$

- Find the distance between the points using the phase shift just computed and the distance formula used above. The distance computed is in metres, we multiply it with a conversion factor to convert the distance to feet. We use the conversion that 1 metre is equal to 3.281 feet.

We follow the above-mentioned steps to compute distance in our model. This method had better accuracy over other methods because we take into consideration the shift in the signal with respect to the frequency, rather than many other inconsistent ways to just take the time difference between sending and receiving the signals.

4.5 Calibration Mode

The Calibration Mode assists us in calibrating the distances in our system. The distance measured by the system is bound to have some errors due to factors like noise, temperature, humidity. When we enter the Calibrate Mode, we are prompted by the UI to stand at 5 ft and 10 ft distances. At each distance, when we are ready, we press a button. Once the button is pressed, the tone is generated, which then is captured by the microphone, and the steps discussed in the Signal Processing and Distance Measurement Module are performed to determine the distance. We now find the error margin between the computed distance and actual distance. These correction constants are taken into account when distance is measured in the test module.

We performed the experiment for 100 trials to find the distances between 1-10 feet and this is the graph for an arbitrary trial. From this we obtain our correlation constants by plotting the actual distances versus the calculated distances.

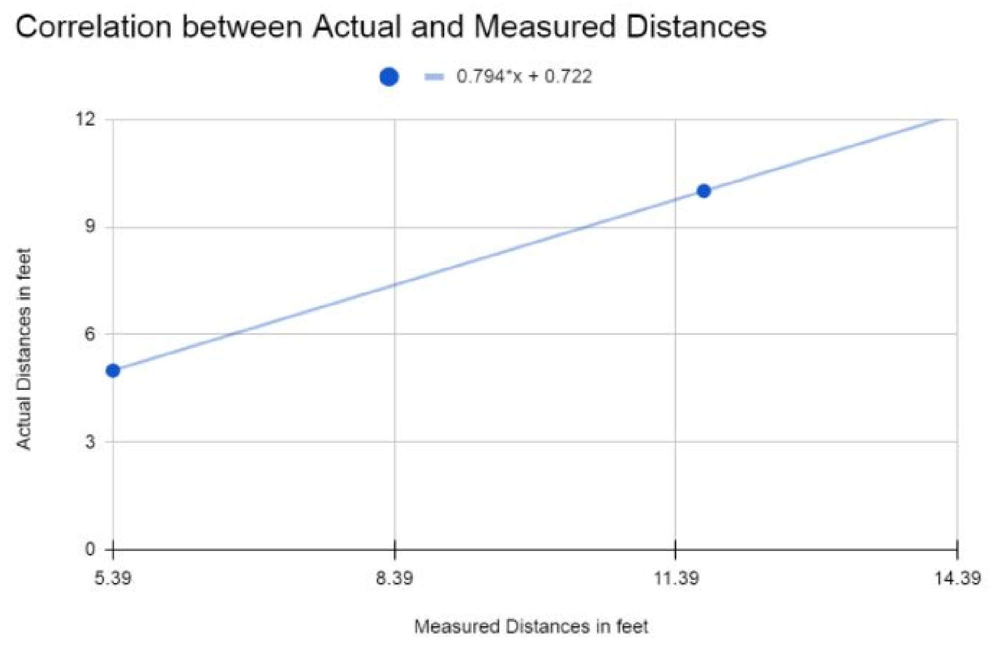

Figure 4: Graph used to find the correction values in Calibration module

We use the slope and intercept of the line of best fit to calibrate the measured distance using the formula

$$\text{corrected distance} = \text{slope of line} * \text{measured distance} + \text{intercept of line}$$

4.6 Test Mode

The Test Mode is essentially demonstrates the functionality of our system. When we enter the Test Mode, we are prompted by the UI to stand at any distance and press the button to indicate that we are ready. Once the button is pressed, the tone is generated, which then is captured by the microphone, and the steps discussed in the Signal Processing and Distance Measurement Module are performed again to determine the actual distance. We take into account the correction constants from calibration module and use them to rectify the distance we get from our distance function.

4.7 Graphical User Interface

The State Machine for our system's GUI is given below.

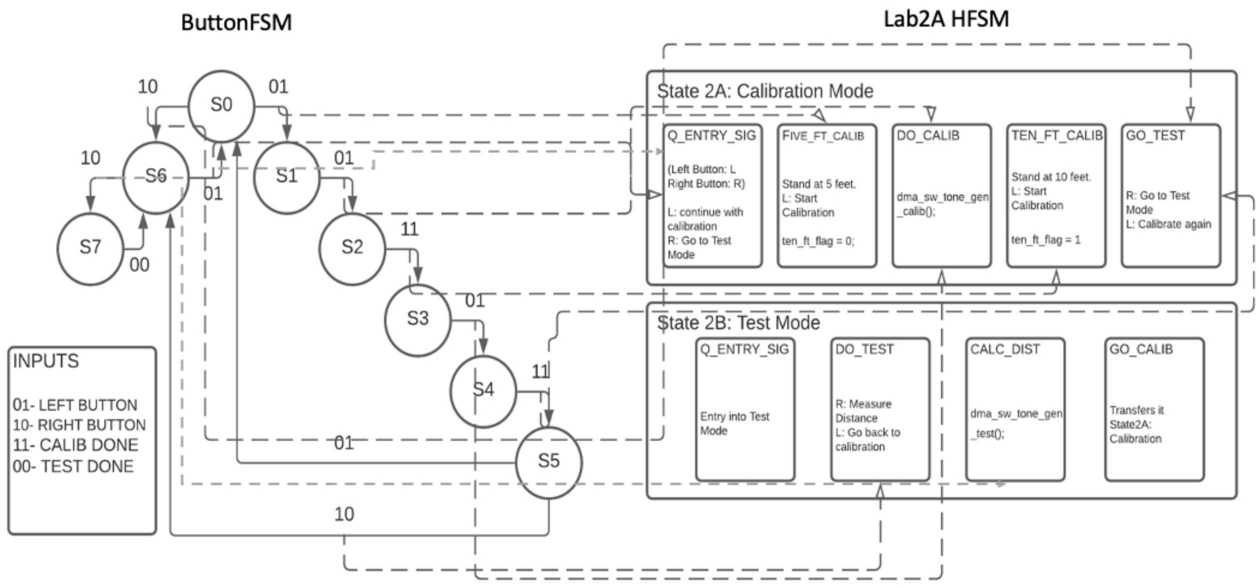

Figure 5: State machine of the Acoustic Distance Measurement System

Here we use two interconnected FSMs, The inputs FSM and the GUI FSM.

The Input FSM (ButtonFSM) has States S0 through S7 that used for the generation of signals to activate State2A and State2B. States S0 through S7 have two kinds of inputs: inputs triggered by the press of a push button, and inputs that signify the completion of the Calibrate mode or the Test mode. Inputs 01 is triggered by the Left Push Button (L), 10 is triggered by the Right Push Button(R), 11 Signifies that the Calibration Mode is done and 00 signifies that Test Mode is done. These signals are inputs to state S0 to S7, which in-turn output signals that serve as inputs to the GUI FSM.

The GUI FSM is an HFSM designed Using QP Nano. Inputs to this FSM are obtained from the Input FSM. Here we have 2 child states: State2A: Calibration and State2B: Test Mode. Here we perform all GUI based operations for both Calibration and Test Modes, such as prompting instructions, performing computations and displaying results. We toggle states in this FSM based on the input received from the Input FSM.

5 Results

We performed 100 trials to find the distances between 1-10 feet, to get a trend of how accurate the system is. Based on these trials, we calculated the mean squared error for all the distance calculations which came out to be 3.32 feet. We have proposed in Future Scope section how to reduce the error in determining the distances and get better accuracy.

The video demo captures the working of our system. We go over the different modules of our system and demonstrate the actual calibration and calculation of distances using our system.

6 Conclusion

This project proposes a simple method for Distance Measurement. Phase Shift can be used to calculate the time difference between the signal transmitted and the signal received, which in turn can be used to measure the distance between two points. We make use of functionalities of the Nexys A7-100T board and low cost peripherals, so it is a very cost effective system as well. We learned a lot about integrating different peripherals, particularly the use of PWM Audio output and MEMS Microphone through this project. Limited availability of resources and difficulty of signal processing in Embedded C restricted our ability to get more accurate results in given time frame. However, in future, we hope to improve it using certain ideas mentioned below.

7 Future Scope

1. Implementation of chirp (we figured out the implementation but didn't present it due to lack of time) and appropriate processing on the microphone for it.
2. Use more points for calibration. We could do this easily except it increases complexity of our FSM so we have presented our basic working model for now.
3. Use of different frequencies concurrently or broadcasting white noise for simple correlation.
4. Account for multi-path propagation of the signal
5. Determine the changes in velocity of the transmitter module as it moves.
6. Built on the GUI to be more interactive and graphic.
7. Account for factors like temperature and humidity while calculating distance.

References

[1] Stephane Poujouly, Bernard Journet, Dominique Placko., Digital Laser Range Finder: Phase-Shift Estimation by Undersampling Technique. *IECON'99. Conference Proceedings. 25th Annual Conference of the IEEE Industrial Electronics Society (Cat. No.99CH37029)*, 1999, pp. 1312-1317 vol.3, doi: 10.1109, 1999.

[2] B. P. Flanagan, "Self Localization Using A Modulated Acoustic Chirp, *Technical report, The MITRE Corporation,* , 2007.

[3] Nexys A7 Reference Manual, https://reference.digilentinc.com/reference/programmable-logic/nexys-a7/reference-manual.

[4] Nexys-A7-100T-DMA-Audio Demo, https://github.com/Digilent/Nexys-A7-100T-DMA-Audio.

[5] M. Chongchamsai, S. Sinchai, P. Wardkein and S. Boonjun, Distance Measurement Technique Using Phase Difference of Two Reflected Ultrasonic Frequencies, *2018 3rd International Conference on Computer and Communication Systems (ICCCS)*, pp. 372-376, doi: 10.1109/CCOMS.2018.8463283, 2018.