Multipurpose Acoustic Localization System

**B.Vikram, Chonben Lotha,**

**R.S.D.Wahidabanu,**

**Government College of Engineering, Salem,**

**Salem, Tamil Nadu, India**

**vikramslm@rocketmail.com**

***Abstract*—** The Multipurpose Acoustic Localization System is a solution that can be used to accurately locate loud sounds like gunshots and explosions in real time. The principle of Multilateration, which locates the sound source using the time difference of arrival (TDOA) of the sound wave at three or more predefined points, is used here. The system implementation involves four Piccolo based slave sensor modules placed around the area to be monitored, connected via a wireless star network to a master module having the Delfino DSC, which acquires the sound data and computes the location of the sonic event. The entire system is implemented as stand-alone, portable and cost effective Embedded Solution, without the usage of expensive Personal or Industrial Computers. Extensive DSP techniques are employed in the implementation of the system.

***Keywords — Time Difference of Arrival (TDOA); DSP; FFT; Wireless networks; Piccolo; Delfino.***

**I. INTRODUCTION**

The Multipurpose Acoustic Localization System is a solution that can be used to accurately pinpoint the source locations of loud, low to medium frequency sounds like gunshots, explosions and even serious vehicular accidents and relay the location to the user in real time.

It works on the principle of Multilateration by which, the time difference of arrival (TDOA) of the sound wave at three or more predefined points, is used to determine the location of the sound source.

The project was selected because it involves many futuristic areas like DSP, Wireless Networks, Acoustic Engineering and Analog Electronics. Moreover, it has many applications in the areas of law enforcement and military, which is the need of the hour for our country.

The aim of the project is to create a working solution which can be easily scaled up to monitor large areas for both civilian and military purposes.

*A. Technical Background*

The most relevant work found was “Acoustic Triangulation Device” by J.Sanders et. al. of the University of Central Florida, [1] which uses Signal Triangulation. We decided to build a similar system by using a simpler, more efficient and possibly more accurate technique, Multilateration, as proved by [5] and [6]. Unlike numerous other references ([1], [2], [3] and [4]), where lion's share of the work is done in the PC environment and all sensing parts are connected with wire, **we decided to create a completely stand-alone embedded solution using DSP technology with various sub-systems interconnected by wireless networks.**

*B. Proposed Solution*

The system consists of a Master module having the Delfino - TMS320F28335 Digital Signal Controller (DSC) which is connected to four slave modules by a wireless Star Network implemented using Xbee Zigbee modules. The slave modules contain the microphones for listening to the sound events along with the necessary amplifier, filter and signal conditioning circuits. The slave modules also contain the Piccolo DSCs which sample the sound data and transmit it to the master module using Zigbee modules.

The Master module receives the sound data from the slave module and calculates the position of the sound source after verifying the authenticity of the sound signals using Fourier analysis. The final output is displayed in the LCD mounted on the master module.

Thus, the entire system is completely stand-alone and is implemented using Embedded Systems without the usage of expensive Personal or Industrial Computers unlike the references cited previously. Further, wireless networks are used here to connect the various sensors instead of using wired connections.

*C. Organization of the Paper*

The paper is organized as follows. Section II has a brief description of various components present in both the master and slave modules. Section III gives a detailed working description of various hardware and software components and gives an idea about where all of them fit in the big picture. Section IV gives a detailed description about the implementation of hardware and software parts and it has two subsections for the same. Section V explains the results obtained from testing the system and Section VI explains the conclusions derived from our work.

**II. PROPOSED SOLUTION**

The system is implemented with many sub-modules interconnected by wireless networks as shown in Figure 1. The major components of the system are as follows.

A. Master Module – 1

1. Main Digital Signal Controller (DSC) – Delfino – TMS320F28335
2. Wireless Transceiver – Zigbee Module
3. Temperature sensor – LM35
4. LCD display
5. Li-Ion battery pack with voltage regulators.

B. Slave Modules – 4

* 1. Slave DSC – Piccolo – TMS320F28027
  2. Wireless Transceiver – Zigbee Module
  3. Sensitive Omnidirectional Microphone
  4. Analog Signal Conditioning Circuit (Audio amplifier, filter and voltage shifting circuits)
  5. Li-ion battery pack with voltage regulators and positive to negative voltage convertor.

*A. Master Module*

The master module contains the TMS320F28335 – Delfino microcontroller. A Zigbee wireless module, an LM35 temperature sensor and a character LCD display are connected to the Delfino microcontroller. The master module can be placed anywhere within the range of the wireless transceivers of the slave modules. It need not be placed inside the area enclosed by the slave modules. The master module is powered by a 7.4V Li-ion battery pack. 5V and 3.3V voltage regulators are used to generate the required power supply voltages.

*B. Slave Module*

Each slave module has an omnidirectional microphone to detect loud sounds. The output of the microphone is connected to an analog signal conditioning circuit consisting of a Microphone pre-amplifier, a low pass filter and a difference amplifier used for scaling and shifting the filtered signal. The output of this circuit is given as the input to the ADC port of the TMS320F28027 – Piccolo DSC. A Zigbee wireless module is connected to the Piccolo DSC. Four slave modules are placed around the area in which the sonic event is expected. The power supply is taken from a 7.4V Li-ion battery pack. A set of voltage regulators are used to generate +5V and +3.3V for power supply. An adjustable positive to negative voltage convertor is used to generate the required -3.3V for providing dual supply to the Op-Amps.



**Figure 1: Block Diagram of the System**

The slave modules are placed in a rectangular or square shaped pattern around the area to be monitored. The sound source location is displayed as distances in x and y coordinates from the first slave module whose location is considered as the origin.

**III. WORKING DESCRIPTION**

When a sonic event occurs within the specified area, the sound waves are detected by the microphones present in the slave modules. The sound signals are given to ADC ports of the Piccolo DSCs after signal conditioning. The Piccolo DSCs record the signal and also note the time of reception of the signal. These signals are then sequentially transmitted to the main microcontroller along with the time of reception data.

The Delfino DSC receives the signal samples and converts the data into frequency domain using FFT. Then, the Delfino DSC analyses the frequency domain signals to check whether all of them have the same frequency components.

After verifying authenticity of the sonic event, the Delfino DSC measures the time of difference of arrival of the sound wave with respect to the four microphones.

To ensure that all the four slave microcontrollers have synchronized clocks for the detection of the time of reception of the sound wave, the Delfino DSC transmits a synchronizing signal to all the slave Piccolo DSCs so that they can start counting at the same time. This is done in the beginning initialization stage before listening for sonic events.

The locations of the various microphones are predefined and are known to the Delfino DSC as shown in Figure 2. The speed of sound in air at ground level varies with respect the atmospheric temperature. So, the temperature sensor is used to compensate for this temperature difference. The speed of sound in air is given by **V = 331.3 + 0.606xT m/s**. (Where T is the temperature in degree Celsius).



**Figure 2: Position of the Slave Modules around the area to be monitored**

To determine the exact location of the sound source from differences in distance of arrival readings, a look up table is created with distance of arrival values for various discrete points in the area surrounded by the microphones. The point with minimum error in difference of distance of arrival readings for all the microphones is selected as the location of the sound source. The coordinates of the sound source with respect to the first microphone are displayed using the LCD. The size of the rectangular/square area and the resolution of the look up table points can be specified earlier in software.

**IV. IMPLEMENTATION**

*A. Hardware Implementation*

The hardware implementation has been completely done using project boards and all the components are hand soldered. The only PCBs used were the DSC evaluation boards, the analog EVMs for the filters and the breakout boards for the XBee Zigbee modules. The complete project hardware, including the master and slave modules can be seen in Figure 3.



**Figure 3: Project Hardware**

*a. Slave Module – Hardware*

*i. Microphone and Analog Signal Conditioning Circuits*

An omnidirectional, high sensitivity, low power condenser microphone (AOM-4544P) was selected for the slave modules. The microphone also has a minimum operating voltage less than 5V which makes it convenient to be used with the battery powered power supply.

The Analog signal conditioning circuits consists of three parts – Mic preamplifier, Low Pass Filter and a Difference amplifier for scaling and shifting the signals.

The entire circuit has been implemented using two TL082 – Dual JFET Input Operational Amplifiers. The particular IC has been used because of its high input impedance and its popularity as an audio amplifier. The IC also has two channels thereby reducing component count and cost. Moreover, this Op-Amp was used because it had better performance characteristics at low power supply voltages ±3.3V than other ICs we evaluated.

The Microphone is biased with a power supply voltage of ±3.3V. The output is coupled to an inverting amplifier circuit having a gain of 56. The gain of the amplifier is set such that the circuit has right sensitivity; high enough to detect all loud sonic events but low enough to reject low intensity noise sources.

The output of the amplifier is fed to a low pass filter which is used as an anti-aliasing filter. Due to the unavailability of firearms for testing, we had to test our system with sound sources like exploding balloons and whistles. All of them had a frequency range of 0 – 5kHz. However, to be on the safer side, we are sampling the analog signals at a frequency of 30 kHz. In order to prevent aliasing, the signals are filtered by a 4 pole Chebyshev low pass having a cutoff frequency of 7.5kHz. The filter has zero attenuation pass band of about 7Khz and an attenuation of 28dB at 15kHz which is adequate to prevent aliasing. We selected the Chebyshev filter response to get high stop band attenuation. The pass band ripple of 1dB is not a significant issue for our application. The filter was designed using FilterPro software. The filters have been implemented using TI Analog Evaluation Modules (EVMs) as they provided soldering facilities for both through-hole and SMD components.

The filtered output signal varies for approximately ±2.8V. Since the ADC of the Piccolo DSC can only sample signals varying from 0 to +3.3V, The signals are passed through a difference amplifier circuit which scales and shifts the signals such that it varies between 0 and +2.8V. The Filter output is applied to the non-inverting input, while an adjustable negative voltage bias (-1V) generated using a potentiometer is applied to the inverting input. The output results in an attenuated signal that has a mean value of +1V and having a peak to peak voltage of 2.8V. This signal is now given to the ADC port of the DSC.

*ii. Piccolo – Digital Signal Controller (DSC)*

Each slave module has a TMS320F28027 - Piccolo DSC. The controller was selected because of its in-built High Resolution 12 Bit ADC peripheral and adequate RAM Size (12kB). The RAM size is necessary to store the ADC samples and transmit it to the master module. The DSC was primarily selected because of its availability as a Piccolo Launchpad, an inexpensive evaluation platform with an inbuilt USB Emulator and 3.3V Voltage regulator. The voltage regulator is also used to power the analog circuits of the slave module. The output of the analog signal conditioning circuit is given to an ADC port and the XBee module is connected to the SCI (Serial Communication Interface) pins of the DSC. The availability of Control Suite, a support software containing software libraries and examples made the Piccolo DSC, a natural choice for our application.

*b. Master Module – Hardware*

*i. Delfino – Digital Signal Controller(DSC)*

The master module has the TMS320F28335 - Delfino DSC, a high end device in the C200 family with all the bells and whistles required for DSP processing. The device has everything one can think of for any medium complexity application including full DSP capability supported with a full-fledged DSP library, a Floating Point unit with a complete math library along with other features that normally comes with a DSC. The availability of Control Suite, a software containing all the necessary libraries and examples makes the implementation of very difficult tasks much easier. Needless to say, the DSC has a large fat RAM space(64kB), adequate for storing the sound samples from the slave modules, FFT buffers and outputs, look up tables and other data necessary for our application.

The DSC has been used in the form of an Experimenter kit with an inbuilt USB Emulator and 3.3V Voltage regulator. The XBee module is connected via the SCI (Serial Communication Interface) pins of the DSC. An LCD module is connected to the GPIO pins and the Temperature sensor is connected to an ADC pin.

*ii. 2×16 Character LCD Module*

A 2×16 Character LCD module is used to display the output of the system. The LCD module is also used for debugging the application by displaying information about the part of the code being executed.

*iii. Temperature Sensor – LM35*

A Temperature sensor (LM35) is connected to the Delfino's ADC port. The LM35 has been selected because it gives linear output in degrees Celsius without any need for any external circuits like Instrumentation Amplifiers. The temperature sensor is used to compensate the sound speed values for temperature variations.

*c. Common Hardware (Master and Slave modules)*

*i. Wireless Modules*

Zigbee Wireless modules are used in both master and slave modules. The XBee modules made by Digi International were selected because of their ease of use and the ability to transmit a signal from the Master(Coordinator) module to all the Slave (Router) modules simultaneously (Broadcast Mode). This ability is very essential in order to synchronize the timers of the slave Piccolo DSCs. The Xbee modules in the slaves are configured as Routers and the Xbee module in the master is configured as the Coordinator for the star network. The modules are configured in AT(Transparent) mode where the data given through the UART (SCI) inputs is exactly transmitted to the correct destination node. Since the XBee modules have pins with 2mm pitch, separate adapter modules have been used to solder the modules in project boards having hole pitches of 2.5mm.

*ii. Battery and Power Supply Circuits*

One of our primary targets was to make the entire system completely portable. That makes battery power supply a necessity for our application. A 7.4V, 300mAh Lithium Ion battery pack with inbuilt charging and protection circuits, was selected for use in all the sub-systems, the master and slave modules. The selection was primarily due to the convenience of the inbuilt charging circuits which eliminates the need for separate charging circuits.

A 5V low dropout regulator (UCC283-5) to power up the Piccolo Launchpad (Slave Module), the Delfino Experimenter Kit (Master Module), the Zigbee Modules and the Character LCD (Master Module). As mentioned earlier, the DSC evaluation boards have their own 3.3V regulators to power up the respective DSCs. In the slave modules, the power supply from these 3.3V voltage regulators are used to power up the microphone and analog signal conditioning circuits. For supplying negative voltages to the Op-Amps, an adjustable positive to negative voltage convertor (PTN04050A) is used. The output is manually set using a potentiometer at -3.3V.

*B. Software Implementation*

*a. Slave Modules – Piccolo Digital Signal Controllers.*

The software implementation in the Piccolo DSCs can be better explained in two parts. The first part is the main program code and the second one is the ADC EOC(End-of-Conversion) interrupt routine which runs once every 33 milliseconds as the ADC sampling frequency is 30kHz. The following on-chip peripherals have been used; GPIO, SCI (Serial Communication Interface), ADC and ePWM (for giving ADC SOC(Start-of-Conversion) Trigger). The operation is briefly explained here.

After executing all initialization routines, the DSC waits until the master initiates communication. The DSC gives an acknowledgment and waits until the master gives the command to start listening for sounds. Then, the ePWM timer is started to enable ADC SOC. The timer counter is also started at the same time. Then DSC waits until it hears any loud sound. If it does, the sound detection time and 1024 continuous sound samples are sent to the master module. Now, the DSC awaits command from the master to start listening for the next sonic event and the cycle goes on.

Sound detection and storing of the sound samples take place in the ADC EOC interrupt routine in order to free up the DSC to look after the serial communication (Zigbee Module). When the routine is executed, the current ADC sample is checked whether the magnitude level beyond 2V or below 0.4V. When this happens, a flag variable is set which will enable the storage of the next 1024 samples and the current timer counter value is saved as the Sound Detection Time. Then, the DSC will start reporting to the master that a sound has been received.

*b. Master Module – Delfino Digital Signal Controller.*

In the Master Delfino DSC, the following on-chip peripherals have been used; GPIO, SCI (Serial Communication Interface) and ADC. We also made extensive use of the DSP and Math libraries for our application. The complete operation is briefly explained in the following paragraphs.

After executing all initialization routines, the DSC sequentially checks whether all slaves are online. Next, it sends a synchronization command and instructs them to start listening for sounds. Then, the DSC keeps on checking whether any of the slaves have detected any loud sounds. If all four slaves report the detection of the sound, the master asks them to transmit the Sound Detection Time along with the 1024 sound samples. After receiving the data, the DSC checks whether the data is authentic by comparing the FFT outputs of the signals. If the data is authentic, the DSC computes the sound position using the generated look-up table data and displays it on the LCD. Otherwise, the data is discarded and an appropriate message is displayed in the LCD. Then, the DSC asks the slaves to listening for sounds again.

The exact technique used for finding out the authenticity of the received signals is described here. The Delfino DSC receives 1024 sound samples from each slave module. The samples are processed on the fly with the same arrays used for storing the samples and the FFT outputs. The samples are divided into two equal halves and two 512 point Real FFTs are computed. Then, the magnitude values of the FFTs are averaged and stored. This technique yields better and more usable values than a single 1024 point FFT [7] and also consumes less RAM space. The same procedure is followed for sound samples from all the slave modules.

Then, the FFT values are passed through a 9 point moving average filter to reduce the number of peaks and ridges to make the values more usable. Then, the frequency values at which top five magnitude peaks occur, are found for each FFT reading. If at least one of the values coincide (lie within a deviation of 5 points) for all the four FFT readings, the signal is assumed to have the same major frequency component and is declared authentic. Otherwise, it is considered as a false detection.

For finding out the point of origin of the sound, we initially tried to implement a Hyperbolic method. By correlating the TDOA values from a pair of microphones, we can get a Hyperbolic equation. By using the values from two or more pairs, we can get multiple equations which can be solved to find out the point of origin of sound wave. This technique required the usage of complicated numerical techniques which are extremely difficult to implement in a microcontroller environment.

So, we considered the problem in a reverse approach and indigenously developed an extremely simple technique with adequate accuracy, which is very simple to implement. We created a look-up table containing the distance values from the four microphones for a number of points inside the area covered by the microphones. We then check each value with the actual difference in the distance values with respect to the first microphone. The point having the least error between the assumed and actual values is taken as the point of origin of the sonic event. Naturally, the accuracy depends upon the number of points in the look up table which can be adjusted as per user needs.

Since, storing the look-up table values in the RAM would take up a lot of memory space, the assumed distance values are created on the fly as the DSC compares the actual distance values with the assumed values for various points inside the area under observation. The size of the area and the density of points are specified earlier.

**V. RESULTS**

Due to the inability to test with firearms, we had to test our system by bursting balloons and using whistles. The testing area had to be reduced to sizes within which, the test sound can travel without much reduction in intensity. So, we tested our system in an outdoor test area of size 16'×16'. The performance reliability of various tasks are listed here as a probability of successful operation.

**Table 1:**

**Success Rate of Various Tasks**

|  |  |  |
| --- | --- | --- |
| **S.No** | **Task** | **Success Rate** |
| **1.** | Wireless transmission of sound data | 95% |
| **2.** | Authentication of a proper sonic event | 90% |
| **3.** | Rejection of an improper sonic event (Microphone triggered individually by using local sounds) | 80% |
| **4.** | Successful sound localization with an error of less than 2 feet. | 60% (Improvement under progress) |

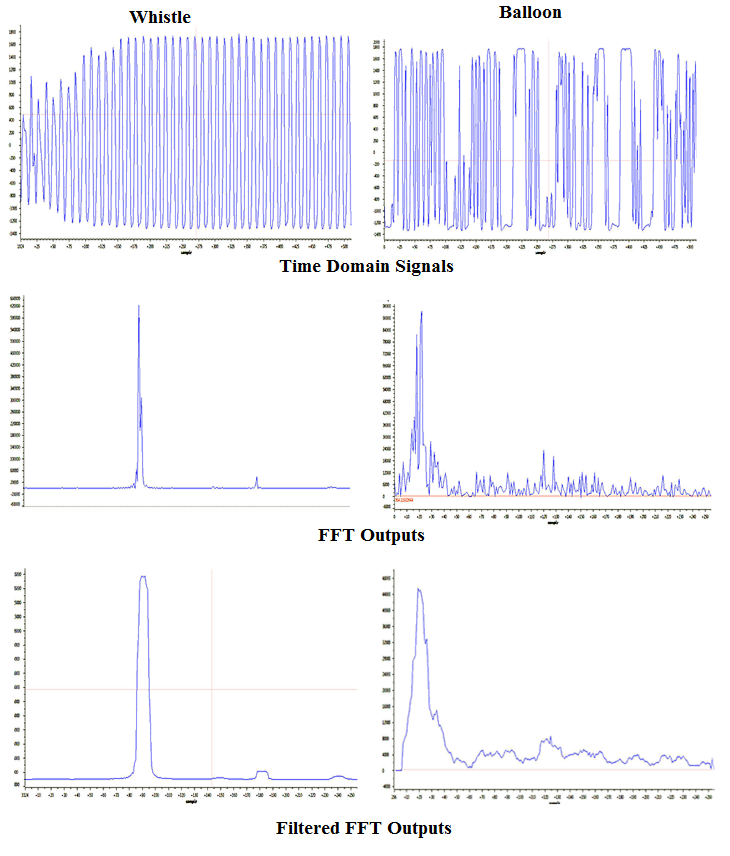
Samples of the sound data in time and frequency domains (FFT output after averaging two 512 point FFTs) for both test sounds (Whistle and Exploding Balloons) are shown in Figure 4. The output of the moving average filter on the FFT readings is also shown. The importance of the moving average filter can be clearly seen from the figure. The plots are taken from the graph function in Code Composer Studio.

**VI. CONCLUSIONS**

From the previous section, it is obvious that a number of improvements must be made. The reliability of the wireless transmission and sound authentication is acceptable, but the reliability of the localization process is not. Further, the accuracy can be improved greatly.

The problem seems to come from the insufficient acoustic isolation for the slave modules. We think the primary problem is due to the fact that the sound travels much faster in solids than air. The sound waves traveling through the ground and the walls seem to reach the microphones faster than the sound traveling through air, despite the fact that the later path is shorter. This has to be the problem because the synchronization process is near perfect in testing the Xbee modules.

We are working to solve the problem by experimenting with sound absorbing materials for the slave modules. We hope to see significant improvements soon both in reliability and accuracy of the sound localization. We didn't yet include the temperature sensor readings in our calculations. We are testing with a preset temperature measured with a thermometer. We will include the temperature sensor readings soon.



**Figure 4: Sound Data samples in various stages of processing.**

If the accuracy and reliability can be improved, the system provides an excellent, cost efficient alternative to complex implementations using Industrial and Personal Computers with acceptable accuracy for most applications.

If greater accuracy is needed, suitable Numerical techniques can be implemented, given enough time. The technique we developed can be used to provide an approximate solution, which can be given as input to iterative non-linear equation solving algorithms to give a more accurate result.

ACKNOWLEDGMENTS

We thank Mr.M.Balaji, Technical Director, Frontline Electronics Pvt. Ltd., India, for his guidance.

**REFERENCES**

1. J.Hopfinger, B.Noble and J.Sanders,“Acoustic Triangulation Device”, University of Central Florida.
2. A.K.Tellakula, “Acoustic Source Localization Using Time Delay Estimation”, Supercomputer Education and Research Center, Indian Institute of Science, Bangalore, India.
3. A.Bahai, M.Ergen and P.Minero, “A Distributed Localization System for Sensor Networks”, Distributed Sensing Lab, Department of EECS, University of California, Berkeley, Berkeley, CA, U.S.A
4. J.Lantto, “Sound Source Detection System for Control of an Autonomous Mobile Robot – A Behaviour based Approach.”, Romansy 16 Robot Design, Dynamics and Control. Pg 339 – 346, June 21-24, 2006.
5. H.I.Ahmed, Y.Du, I.Memon, P.Wei, W.Xie, “Estimation of Time Difference of Arrival(TDOA) for the Source Radiates BPSK Signal”, presented in IJCSI International Journal of Computer Science Issues, Vol. 10, Issue 3, No 2, May 2013.
6. J.C.Hamann, “Exploring the Mathematics of Multilateration”, Department of Electrical and Computer Engineering, University of Wyoming, February 9, 2007.
7. S.W.Smith, “Digital Signal Processing, A Practical Guide to Scientists and Engineers”, Elsevier Publications, 2003, Pg 170.

**APPENDIX A**

The Schematics for the Master and Slave modules are included in this section.

*A. Analog Signal Conditioning Circuits – Slave Module*

**1. Microphone Preamp**



**2. Four Pole Low Pass Filter**



**3. Voltage Scaling and Shifting Circuit**



*B. XBee Modules – Master and Slave modules*

**XBee Module – Slave XBee Module – Master**



*C. 2×16 Character LCD – Master module*



**APPENDIX B**

The Flowcharts for the master and slave codes are included here along with the code.

1. *Flowcharts*

**1. Piccolo DSC – Main Code**



**2. Piccolo DSC – ADC EOC Interrupt Routine**



**1. Delfino DSC**

