A Compact Ultrasonic Airflow Sensor

For Clinical Monitoring of Pediatric Tracheostomy Patients

Thomas Hall Ruscher

Thesis submitted to the faculty of the Virginia Polytechnic Institute and State University in partial fulfillment of the requirements for the degree of

Master of Science in

Mechanical Engineering

Alfred L Wicks, Chair

Andre Muelenaer

Kathleen Meehan

December 11, 2012

Blacksburg, Virginia

Keywords: Ultrasonic, Time-of-flight, Airflow, Sensor, Tracheostomy

A Compact Ultrasonic Airflow Sensor

For Clinical Monitoring of Pediatric Tracheostomy Patients

Thomas Hall Ruscher

ABSTRACT

Infants and young children with tracheostomies need better respiratory monitors. Mucus in the tracheostomy tube presents a serious choking hazard. Current devices indirectly detect respiration, often yielding false or delayed alarms. A compact ultrasonic time-of-flight (TOF) airflow sensor capable of attaching directly to the tracheostomy tube has been developed to address this need. The ultrasonic flow sensing principle, also known as transit time ultrasound, is a robust method that correlates the timing of acoustic signals to velocity measurement. The compact prototype developed here can non-invasively measure all airflow into and out of a patient, so that breath interruption can easily be detected.

This paper concerns technical design of the sensor, including the transducers, analog/digital electronics, and embedded systems hardware/software integration. Inside the sensor's flow chamber, two piezoelectric transducers sequentially transmit and receive ping-like acoustic pulses propagating upstream and downstream of flow. A microcontroller orchestrates measurement cycles, which consist of the transmission, reception, and signal processing of each acoustic pulse. The velocity and direction of airflow influence transit time of the acoustic signals. Combining TOF measurements with the known geometry of the flow chamber, average air velocity and volumetric flow rate can be calculated. These principles have all been demonstrated successfully by the prototype sensor developed in this research.

Acknowledgements

I am grateful for the opportunity to pursue engineering research on a medical device with potential to improve quality of life for young children with respiratory defects. In addition to the satisfaction of knowing that my work could directly help others, I have found great joy in the nature of the work itself.

I owe these opportunities to my primary advisor, Al Wicks, PhD, as well as Andre Muelenaer, MD, and Joseph Tamez, MD, for help and guidance during this research work. The Pediatric Medical Device Institute, a non-profit organization based in Roanoke, VA, provided funding for research and development.

Thank you Mom and Dad -- you let me bring home cool gadgets, including surplus industrial electronics, from Coiner's scrap yard when I was young and tear that stuff apart on the back porch. I learned so much from those experiences, and I owe part of my early engineering inspiration to you.

Table of Contents

List of Figures

List of Tables

Chapter 1. Introduction

"Children with a chronic tracheostomy constitute an important subgroup of children who are at risk for potentially devastating airway compromise. There have been no standards published for their care and disappointingly little research."

- Official Statement from the American Thoracic Society Board of Directors [1]

1.1 Problem statement

Andre Muelenaer, MD, and Joseph Tamez, MD, pediatric pulmonologists at the Carilion Clinic Children's Hospital in Roanoke, VA, know that children with chronic tracheostomies face serious challenges. Tracheostomy tubes have internal diameters narrower than the child's natural airway. Despite routine maintenance, tubes can easily become obstructed with mucus, saliva, or blood. During their practice, Muelenaer and Tamez have witnessed five deaths and multiple instances of hypoxic brain injury in the past 5 years. [2]

Most children with chronic tracheostomies are healthy enough to return home under careful supervision. Respiratory monitors are an essential part of outpatient care to warn caregivers when breathing ceases. However, there are no commercially available technologies designed specifically for tracheostomy patients. Instead, physicians must prescribe apnea monitors designed for children without artificial airways, who constitute a much larger population. These monitors indirectly monitor respiration using probes affixed to the child. An American Academy of Pediatrics parental guide book describes the frustration parents experience with these devices [3]:

> *"When babies are sent home on monitors, parents are often relieved. It makes them feel more secure. After about a week, though, they're ready to throw the monitor out of the window because false alarms are driving them crazy."*

The two most common apnea monitors sense changes in chest wall movement or blood oxygen saturation. A comprehensive 2007 study found false positive rates greater than 85%. [4] Poor electrode-to-skin contact can contribute to reliability issues with chest impedance monitors,

while motion artifact can cause a pulse oximetry probe to lose its signal. Desensitized caregivers may be slower to respond to alarms, or tempted to ignore them.

But for pediatric tracheostomy patients, the situation is worse. A child asphyxiating on his own mucus will heave as he gasps for air, and chest impedance alarms may mistake the event for a cough, sneeze, or normal respiration. [5] In such an instance, pulse oximetry used in conjunction with thoracic impedance may be the only way to detect such an event before chest movement ceases. If the thoracic sensor continues to register breathing, then the alarm will be delayed until saturated oxygen in the blood stream drops below a preset threshold.

There is clearly a need for better respiratory monitoring of children with chronic tracheostomies. Chest impedance monitors perform poorly at detecting airway obstructions, a primary cause of apnea in children with tracheostomies. Because these children constitute a smaller subgroup, they have traditionally been underserved by technologies designed for monitoring children without an artificial airway.

1.2 Evaluating potential solutions

In 2007, Dr. Muelenaer decided to survey the medical community. He was interested in seeing if colleagues had developed unpublished strategies for respiratory monitoring of pediatric tracheostomy patients. The question was posed to an online community of pediatric pulmonologists, "What is the best way of monitoring patients sent home with a tracheostomy?" The collective answer was "there is not really a good method." [2] An American Thoracic Society report published several years prior echoes that sentiment *[1]*:

> *"Ideally, a monitor should provide an early, reliable warning signal of critical airway compromise so that emergency airway stabilization might be rapidly instituted. Unfortunately, most commercially available monitors provide only indirect evidence of airway occlusion, sometimes resulting in a delayed warning signal."*

Muelenaer decided to investigate monitoring technologies through his connection to the Pediatric Medical Device Institute (PMDI), a non-profit organization that partners with the engineering community to address clinical needs. In a 2009 proposal, an excerpt of which is shown in Table 1-1, PMDI evaluates currently available monitoring technologies:

Table 1-1: Summary of monitoring methodologies and potential flaws (adapted from PMDI proposal, [6])

Possible Solution	Issues with Technology
Impedance monitor of chest wall movement	- Chest still moves if tracheostomy is blocked or removed - Heart slows with hypoxia which is a late response
Thermistor (detects changes in temperature of inspired vs. expired gas)	- Mucus [coating or fouling sensor] - Heated circuits
End tidal CO_2 (measures inspired vs. expired concentration of carbon dioxide)	- Expensive - Mucus and moisture interfere with accurate measurements - Two types: mainstream & sidestream. - Mainstream sensors heavy - Sidestream sensors susceptible to clogging of sampling tubing - Marked delay in response of sidestream due to length of tubing
Pulse oximeter	- Susceptible to motion artifact - Expensive [particularly for higher resolution monitors]
In-line pressure monitor for ventilators/Continuous Positive Airway Pressure	- In small infants, resistance to flow in small tracheostomy tubes may create enough back pressure that decannulation is not detected

1.3 Direct air flow monitoring

Direct monitoring of tidal flow (air flow in and out of the airway) became the most attractive solution. This style of monitoring would be too intrusive for traditional patients, as it would require wearing a facial mask. But for tracheostomy patients, access to the airway is already well established. All tracheostomy tube sizes, neonatal through adult, are equipped with an industry standard 15mm diameter endotracheal connection. Direct monitoring has the potential to provide the most reliable, robust indication of respiration and to mitigate problems with alarm latency found in indirect monitors.

Thermistor-based airflow monitors were briefly considered, but their susceptibility to mucus and condensed saliva was a considerable drawback. Likewise, any airflow monitoring technique requiring moving parts (such as the micro-turbine based monitors found in traditional spirometers) was considered too unreliable for continuous monitoring.

A sensor based on ultrasonic detection of airflow was determined to be the best solution due to the fact that such a sensor could:

- Be made compact and lightweight.
- Operate with no moving parts, eliminating a point of failure.
- Present minimal additional resistance to airflow.
- Attach directly to the end of the tracheostomy tube.

1.4 Scope of work presented in this thesis

1.4.1 Summary of accomplishments

This thesis details the successful development of a prototype ultrasonic flow sensor, intended for the medical monitoring application described previously. Relevant to a customer perspective, the primary *functional claims* are summarized below. These are qualities and functionalities of the device that would be relevant to a clinician, physician, or home care giver (the end user). The sensor:

- Is compact, approximately the size of a passive Heat Moisture Exchanger (HME) device.
- Has 15mm endotracheal connections, making it universally compatibly with all medical respiratory equipment, including neonatal tracheostomy tubes.
- Detects inspired and expired air flow in a range of normal human respiratory rates.
- Exhibits immunity to humidity & temperature variations.
- Connects to interface circuitry with a single cable.

Select, novel *engineering claims* about the device are that it:

- Achieves a compact size using high Q air-coupled ultrasonic transducers by
    - Reducing channel crosstalk (primarily through shielding).
    - Active transmit transducer clamping.
- Measures TOF of contra-propagating acoustic signals by
    - Amplifying, filtering, and discriminating the received signal
    - Triggering on one particular acoustic peak within the signal
    - Using a microcontroller hardware timer in capture mode to detect rising and falling edges of the acoustic peak with 40ns resolution
- Completes a TOF measurement cycle in as little as 200µs

Challenges of transducer excitation, transducer ring down and interference, precision timing, and weak signal reception and amplification were all addressed to enable this device.

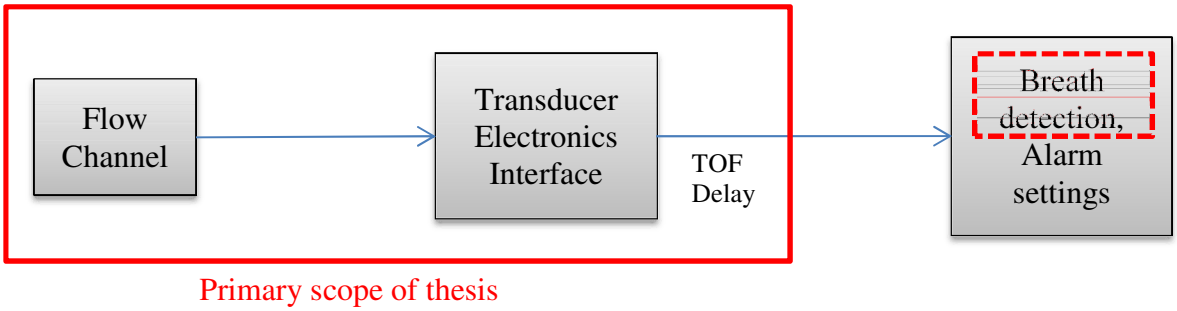


Figure 1-1: Primary focus of thesis work is on the flow channel and interface electronics. Breath detection is included to verify electronics operation. Alarm development is not included.

1.4.2 Limitations of the device, and work beyond the scope of this thesis

Likewise, there are claims that the author does not make about the capabilities of this device. This is work beyond the scope of this thesis, and fertile ground for future device characterization and optimization. Important limitations are summarized below:

- The sensor is a respiratory monitor, NOT a spirometer and, thus, it does not necessarily measure flow with a high degree of precision or accuracy. Instead, it presents a measurement output that is indicative of flow and breathing in the channel.
- The sensor has not been tested extensively for flow accuracy or linearity.
- High flow velocities and/or changes in air temperature can that cause deviations in signal TOF greater than 1.5us (approximately one half of a 300 kHz acoustic wavelength in air) from a preset value will cause erroneous peak detection and flow readings. This is rarely observed under normal operating conditions of human respiration at room and body temperature.

1.5 Final prototype sensor

The flow sensor and electronics (sometimes collectively referred to as “the sensor”) developed in this thesis are shown complete in Figure 1-2. All custom electronics were designed, created, and programmed by the author. Design is discussed in Chapter 5.

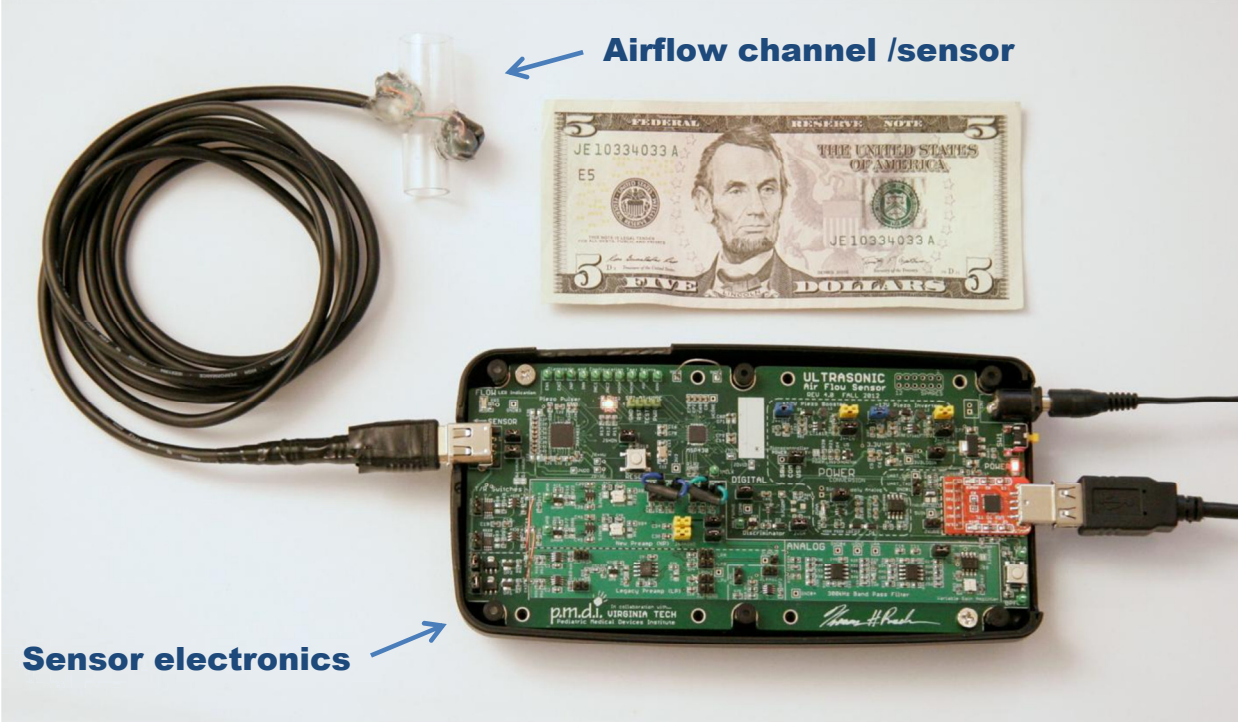


Figure 1-2: Ultrasonic airflow sensor developed by author

1.6 Thesis chapter-by-chapter overview

Chapter 2 is a literature review covering the background of tracheostomies, hazards to the patients, and currently used monitoring methods, including chest impedance and pulse oximetry methods. Indications for a tracheostomy in children are included. At the conclusion of chapter 2, we discuss how ultrasonic TOF can address shortcomings in current sensing methods.

Chapter 3 explains the ultrasonic TOF velocity measurement method. Equations are derived from first principles. Next acoustics are discussed: what it takes to cause an acoustic transmission, how that signal propagates, and how the received signal is sensed. Other theoretical background relevant to ultrasonic TOF is also included.

Two precursor proof-of-concept prototypes that were built prior to the final sensor are presented in Chapter 4. These prototypes were part of the sensor development stage and key stages in an iterative development towards a final device. Electronics architecture is discussed, and incremental stage validation data is presented.

Chapter 5 describes in detail the final prototype design. An overview / block diagram of the electronics architecture is shown, which includes transmit, receive, and digital (microcontroller) stages. Details of the software (firmware) are also discussed, including the measurement routine essential to controlling all transducer interface electronics.

Results of the device are presented in chapter 6. Special attention is given to waveforms and oscilloscope captures, which validate sections of the code and operation. At this time, there has been no detailed performance analysis of the final device, such as testing it against a calibrated airflow source. Therefore, respiration data is instead presented, which shows that the device is capable of monitoring 100mL-500mL tidal flows from a ventilator. Future work will be necessary to fully test the accuracy and precision of the device.

The conclusion in chapter 7 starts with a clinical recap which reiterates the purpose of this research – to design an improved flow sensor for monitoring pediatric tracheostomy patients. A summary of accomplishments is presented, and ideas for future work. Although further work will be needed to test and improve the device, this research has successfully demonstrated a compact ultrasonic TOF airflow sensor.

Chapter 2. Clinical Review

This literature review covers tracheostomies, hazards to the pediatric patient, indirect respiratory monitoring of compromised patients, and direct methods of airflow monitoring. We will begin with the medical aspects of a tracheostomy and discuss indications and prognoses for pediatric patients. Statistics on children with tracheostomies will be provided. The core of the chapter is devoted to current respiratory monitoring methods. End-tidal CO_2, chest impedance, and pulse oximetry will be discussed. Next, we will consider direct monitoring of respiration using a miniature sensor attached directly to the tracheostomy tube.

2.1 Pediatric Tracheostomy

2.1.1 What is a tracheostomy? Why are tracheotomies performed on children?

A tracheotomy is a surgical procedure that creates a hole, known as a tracheostomy, through the wall of the trachea. By creating the tracheostomy, the entire upper respiratory tract, including the larynx, throat, nasal passages, and sinuses are bypassed. Tracheotomies are one of the earliest documented surgical procedures, dating back over four thousand years. [7] In modern times, children may require tracheostomies for a variety of reasons. Although intubation of the airway, a less invasive procedure, is preferred when possible, long term conditions or emergency situations may necessitate tracheostomy.

The indications for a tracheostomy include various congenital and acquired disorders, traumatic injury, and infection. [8] A study of nearly 200 infants and young children, published in 2003, describes some of the most common reasons children get tracheostomies. A list of these reasons, summarized from source data, is given below, in decreasing order of prevalence: [9]

- Upper airway anomalies
- Multiple disorders / diagnoses
- Central nervous system and neuro-muscular disorders
- Prematurity
- Trauma
- Tumors
- Respiratory diseases / infections

Figure 2-1: Shiley™ 3.0mm ID neonatal tracheostomy tube and insertion stylet

A tracheostomy *tube*, such as that shown in Figure 2-1, is often inserted into the stoma created by surgery. The tracheostomy tube can be used to support the airway, provide access to the lower respiratory tract, protect from aspiration, or to administer positive airway pressure ventilation. [10] However, the tube often irritates the tracheal tissue. This irritation can cause a variety of reactions, ranging from excess mucus generation to bleeding. [11] Body fluids have the potential to clog or foul the airway, especially in the narrower tracheostomy tube, which can lead an episode of obstructive apnea.

2.2 Apnea monitoring methods

Apnea is the interruption of normal breathing. It is a problem for many infants and young children, not just those with tracheostomies. Physicians currently use the same types of devices to monitor children with tracheotomies as they do for children without the surgery. Most home monitors are based on thoracic (or transthoracic) impedance and pulse oximetry. However, a common complaint is that these current monitoring methods have an unacceptable level of false alarms. In a three year study published in 2007, only 13% of 5000 apnea alarms were true alarms; the false positive rate was 87%. [4]

The American Academy of Pediatrics parental guide book describes a frequency cause of false alarms in thoracic impedance monitors[3]:

> *"False alarms are usually set off by abdominal breathing or by a loose belt or incorrectly placed monitor electrodes (leads). The frequency of false alarms tends*

to increase as a baby grows older and becomes more active. Ignoring the alarms or assuming them to be false can be potentially dangerous."

2.2.1 Thoracic impedance monitoring

Regular expansion and contraction of the chest wall is a positive indication for respiration. Thoracic impedance monitoring is a form of plethysmography, or estimating the volume change within the body. Detecting chest wall cavity movement requires adhesive probes on the chest, similar to EKG leads, or a band wrapped around the chest with the electrodes implanted.

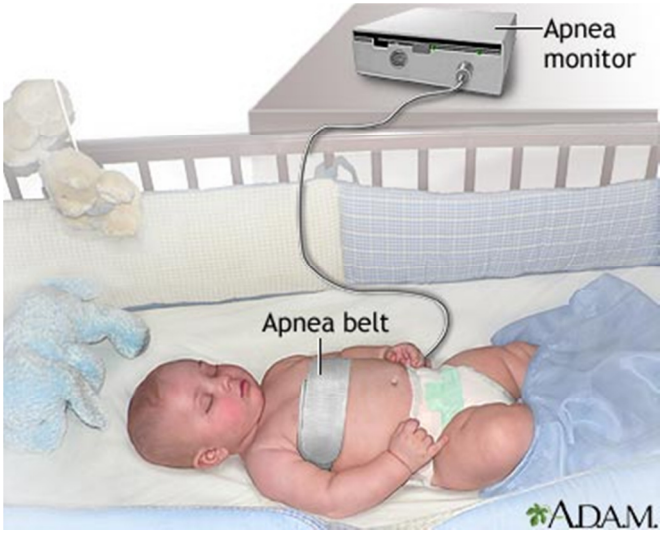


Figure 2-2: Thoracic impedance based apnea belt and monitor (image from [12])

2.2.1.1 Parental guidelines and use of thoracic impedance monitors

Since thoracic impedance works by detecting small changes in resistance across the chest cavity, proper electrode contact is crucial to obtain a valid reading from the device. The electrodes can be either embedded in a chest belt or standalone (see Figure 2-2). Improper electrode placement, contamination, or loose or shifting apnea belts is frequently the source of false alarms. [3] Parents must adhere to a set of use guidelines, procedures, and wash regimen that make the use of these monitors cumbersome. A partial listing of these, excerpted from manuals on infant apnea monitoring, is given below.

Electrode / chest belt placement and application for parents (from [13] and [14]):

- Make sure the surface of the skin where the electrodes will make contact is clean, dry & free of lotion, powder or oil.
- Electrodes must be regularly cleaned and free of any build-up.

- For better quality signal, caregivers should wet the carbon/rubber surface of each electrode with a drop of water and rub it in before attaching to the chest.
- Proper electrode alignment with the chest is important; electrodes should contact the area just below the nipple line. (Electrodes or belts that shift position can trigger alarms)
- Belts must have the proper tension –not so tight as to impair respiration, but not too loose as to shift position
- Foam belts and/or standalone electrodes must be removed a minimum of twice daily for 30 minutes or more to prevent skin irritation (children with sensitive skin may require more frequent removal)

Chest impedance plethysmography is commonly used to monitor respiration in infants and young children. However, electrode care and placement is often temperamental and slight variations can trigger false alarms. This is an undesired burden for caregivers.

2.2.2 Pulse oximetry based monitoring

Consistent oxygen perfusion in tissue extremities is another positive, though indirect, indicator of respiration. Pulse oximetry is a method of determining arterial blood oxygen saturation by measuring red and infrared light absorbance in tissue.[15] Pulse oximetry requires a small probe to be attached to the child, typically around the foot or the toe. Pulse oximetry is sensitive to motion artifact, which can be a trigger for a false alarm.

2.3 Direct airflow monitoring

Combination thoracic impedance / pulse oximetry monitors are the most widely used available tool for home monitoring of pediatric tracheostomy patients, despite the fact that these devices were originally designed to accommodate children without artificial airways (such as those at risk for sudden infant death syndrome, or SIDS). Tracheostomy patients have a direct access to the airway. If it is possible to build an appropriate *direct* airway monitor for these patients, then many of the problems with *indirect* respiratory monitoring may be ameliorated.

The ideal respiratory monitor for tracheostomy patients must meet several clinical use requirements. It must directly measure airflow in and out of the air passage, consistently measure airflow, and resist partial contamination from bodily fluids, including saliva condensation. A user-friendly mechanism should also be compact and easy to clean.

Volumetric airflow measurements are obtained by measuring the velocity of air, then multiplying by the cross-section of the measurement channel. Total or inspired volume can be found by integrating volumetric airflow. Conventional methods of measuring small airflows are based on thermal (hot wire), differential pressure (Pitot tube-based), turbines, and ultrasonic (both TOF and Doppler).[16] With the exception of ultrasonic-based methods, each of the former methods requires a mechanism or probe inside the flow chamber. Thermal wires, Pitot tubes, and turbines have the potential of fouling from bodily fluids. This is the primary reason why ultrasonic time-of-flight (TOF) was chosen.

2.4 Clinical summary

In pediatric pulmonology, there is no satisfactory way to detect tracheostomy tube obstructions that can lead to hypoxia. [1] Without proper monitoring, patients are at risk for complications or death. Despite routine maintenance of the tracheostomy tube, dislodged or copious mucus can obstruct the airway. Young children have difficulty tending to their own tubes and are particularly vulnerable to blockages. In a hospital environment, ventilators, end-tidal CO_2 monitors, thermistors, and other auxiliary equipment provide sufficient monitoring of respiration. However, outpatient monitoring methods, such as thoracic impedance and pulse oximetry, are indirect and prone to false positives. Desensitization of caregivers to frequent false alarms has been cited in medical literature as a contributing factor in cases of child death. [17],[18] A robust, compact airflow sensor capable of attaching directly to the tracheostomy tube is needed for improved home respiratory monitoring of young children.

Chapter 3. Theoretical Background

Ultrasonic time-of-flight or TOF (also known as transit-time ultrasound) uses an arrangement of two piezoelectric transducers to measure flow velocity in a channel. The measurement principle is considered non-invasive compared to other flow measurement techniques, because the flow channel is largely unobstructed – there are no thermal wires, Pitot tubes, or micro turbines. In the first section of this chapter, ultrasonic TOF will be derived from first principles, using analogies and simple geometries. Ultrasonic TOF is inherently a robust method when acoustic signals are transmitted bi-directionally in the "pitch-catch" scheme. The canceled effects of temperature and humidity will be shown mathematically. For clarification, ultrasonic TOF will also be contrasted with Doppler ultrasound, the other primary ultrasonic flow measurement principle that relies on frequency shift, rather than arrival time. We will also discuss why the Doppler technique is not suitable for our application.

Piezoelectric transducers are key elements to the sensor design. In the second half of chapter 4, we discuss basic physics of ultrasound in materials, how it couples from solid to air, and its propagation across a flow channel. We start with an overview of fundamental concepts in acoustics: the wave equation, sound pressure, intensity, propagation speed, and acoustic impedance. Transmission and reflection at boundaries and matching layers will be discussed next, as they are important to transducer efficacy, especially where there is a large acoustic impedance mismatch between solids and air. Abstracting the transducer as a complete unit, we have the link budget which is a convenient means of calculating energy loss along the entire acoustic path.

3.1 Ultrasonic flow measurement using the TOF principle

Velocity of fluid, either liquid or gas, can be determined using acoustic signals propagating within that fluid – with direction parallel to flow. In this subsection, we begin with a conceptual description of the time of flight (TOF) velocity measurement, proceed to a small idealized model, then derive equations, and finally discuss how the bidirectional TOF method has inherent immunity to environmental factors such as temperature and humidity.

For the individual unfamiliar with ultrasonic TOF, a video animation can clearly explain the basic principle. An excellent animation is available from Endress-Hauser, manufacturers of industrial ultrasonic TOF flow meters. The video, entitled "The Ultrasonic Flow Measuring Principle," is available online at: http://www.products.endress.com/ultrasonic and at the address provided in [19]. These links are active as of December 2012.

3.1.1 Clarification of terminology

Terms used in this chapter may have a slightly different meaning from those in other ultrasound resources. Definitions are provided below.

Time-of-flight (TOF): The exact propagation time of an acoustic signal from the moment it is transmitted to the moment it is received.

Transit-time: Synonymous with TOF.

Pulse-echo: A single transducer arrangement where an acoustic signal is transmitted and then later received by the same transducer after reflection from a distant surface, particle, or acoustic discontinuity. Pulse-echo can be used in Doppler ultrasound velocimetry, but not TOF.

Pitch-catch: A dual transducer arrangement, where one transducer transmits a signal that is received by the second transducer. Pitch catch is inherent to TOF flow measurement techniques. There are two implementations of pitch-catch:

> *Unidirectional:* One transducer is a dedicated transmitter, the second a dedicated receiver. Signals of interest only propagate from the transmitter to the receiver.
>
> *Bidirectional:* Both transducers act in dual roles, as both transmitters and receivers. A transducer can either generate or listen for a signal, but practically not at the exact same time. Bidirectional pitch-catch is by far the most common implementation of a TOF setup, for mathematical reasons explained later.

3.1.2 Swimmer analogy

The basic concept of acoustic TOF velocity measurement is analogous to a swimmer in a slow-moving river. From the swimmer's perspective, he/she always swims at the same rate. When swimming downstream, the swimmer traverses a given distance slightly faster because of being carried with the river current. Conversely, when swimming upstream, the swimmer must swim against the current, and takes longer to traverse a given distance. As long as the swimmer always swims at a consistent rate, the "transit time" of the swimmer between fixed points (such

as upstream and downstream rocks in the river) can be measured and exploited to determine the velocity of water flow. Conceptually, this is the same principle behind an acoustic TOF velocity measurement.

3.1.3 Equations for a simple unidirectional TOF

Let's begin with a simple, unidirectional pitch-catch velocity measurement using TOF. To start with, this simplifies the math and eventually leads to the conclusion of why a bidirectional TOF measurement is more robust. Consider the model shown in Figure 3-1. A tubular flow chamber contains ideal, point-source transducers located directly upstream and downstream. The velocity of airflow in the chamber is given by ***v***, and is assumed to be uniform throughout the cross sectional area. The speed of sound in air is constant ***c***. Two point-like transducers **A** and **B** are separated distance ***d*** apart. At time t=0, transducer A emits a brief acoustic pulse (similar to a sonar ping). This pulse propagates along the dashed line, where it is received some time later at point B.

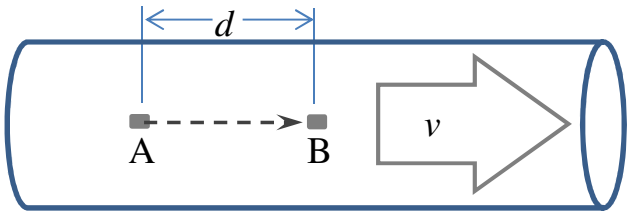


Figure 3-1: Simplified model of unidirectional TOF measurement with arbitrarily small transducers oriented parallel to flow

The time it takes for the acoustic pulse to travel from point A to point B *is the signal* ***time of flight****, or TOF*. That time is given by:

$$t_{A\to B} = \frac{d}{c+v} \qquad \text{Eq. 3-1}$$

Assume $v << c$. All flow velocities considered in this paper will be small compared to the speed of sound in the medium to permit for linear approximation. Rearranging to solve for velocity:

$$v = \frac{d}{t_{A\to B}} - c \qquad \text{Eq. 3-2}$$

As long as c remains constant, a unidirectional TOF measurement from A to B is sufficient to calculate velocity. However, this is rarely the case. The speed of sound in air changes slightly from a particular reference condition, such as standard temperature pressure (STP) depending on environmental factors, such as temperature and humidity.

$$c = c_{air@ref\_cond}\left(1 + \Delta_{temp.} + \Delta_{humid.} + \Delta_{gas-comp.} + \cdots\right) \quad \text{Eq. 3-3}$$

The change in velocity due to temperature and humidity is small, but enough to distort velocity measurements in a unidirectional TOF scheme. The data in table 3-2 shows that sound velocity in air changes by several percent as a function of temperature and humidity fluctuations.

Table 3-1: Sound velocity (m/s) in air as a function of temperature and humidity. Data calculated from [20]

(Note: velocity given in m/s)	Relative Humidity (%)				
Temperature (deg. C)	**30**	**40**	**50**	**80**	**100**
15	340.7	340.8	340.9	341.2	341.3
20	343.7	343.9	344.0	344.4	344.6
30	349.8	350.1	350.3	351.0	351.4
40	356.1	356.4	356.8	358.1	358.9

Only using unidirectional TOF, there is no way to know if a changing TOF measurement is due to flow in the chamber or to environmental changes in the air.

3.1.4 Diagonal transducer bidirectional TOF

The previous section demonstrated that information from a unidirectional TOF cannot alone compensate for environmental air changes within the chamber. A mathematically *informal* way to conceptualize this principle is that the unidirectional TOF change could be due to *either* a velocity or an environmental change:

$$\Delta t_{A\to B} = \Delta t_{velocity} + \Delta t_{temp.} + \Delta t_{humid.} + \Delta t_{gas-comp.} + \cdots$$
$$\Delta t_{A\to B} = \Delta t_{velocity} + \Delta t_{environmental} \quad \text{Eq. 3-4 (a,b)}$$

Measuring two acoustic TOFs in a symmetric, or bi-directional, pitch catch scheme can cancel out changes in propagation speed of sound due to environmental effects (or even contaminants in the flow channel along a shared acoustic path). Consider Figure 3-2. Two changes are evident from the previous model. The first, and most important change, is the fact that acoustic signals are sent between the transducers – a bidirectional pitch catch arrangement -- denoted by the double-arrow ended dashed line connecting A and B. Secondly, the orientation of the transducers is altered to reflect a more realistic and typical sensor geometry – transducers are oriented opposite one another, in the walls of the flow chamber, with an angular component **α** parallel to flow velocity *v*. This only affects the algebra of our calculations. The transducers remain separated by distance d. And the speed of sound in the air (including environmental effects) is still c.

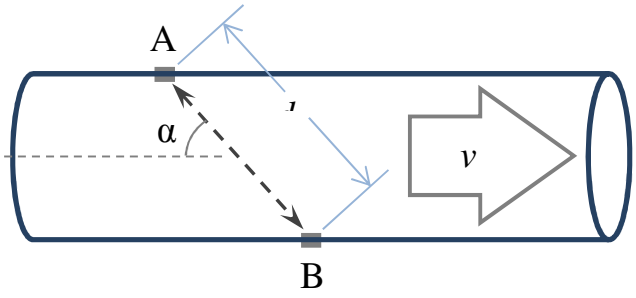


Figure 3-2: Bidirectional TOF arrangement with transducers embedded in opposite flow chamber walls

There are now two TOF equations:

$$t_{A\to B} = \frac{d}{c + v\cos\alpha} \qquad t_{B\to A} = \frac{d}{c - v\cos\alpha} \qquad \text{Eq. 3-5 (a,b)}$$

The significance is that with the extra information from a bidirectional pitch catch TOF measurement, c (and small variations of c) no longer affect the measurement. We have the information to solve for velocity in terms of both TOFs and the known geometry:

$$c = \frac{d}{t_{A\to B}} - v\cos\alpha \qquad c = \frac{d}{t_{B\to A}} + v\cos\alpha \qquad \text{Eq. 3-6 (a,b)}$$

$$\frac{d}{t_{A\to B}} - \frac{d}{t_{B\to A}} = 2v\cos\alpha \qquad \text{Eq. 3-7}$$

$$\frac{d}{2\cos\alpha}\left(\frac{1}{t_{A\to B}}-\frac{1}{t_{B\to A}}\right)=v \qquad \text{Eq. 3-8}$$

$$v=\frac{d}{2\cos\alpha}\left(\frac{t_{B\to A}-t_{A\to B}}{t_{A\to B}t_{B\to A}}\right) \qquad \text{Eq. 3-9}$$

The *most general form* of the bidirectional ultrasonic TOF measurement principle is expressed by Eq. 3-9. Note that c has been decoupled from the expression. Only the geometric orientation of the transducers and the two measured TOFs are known to calculate average velocity in the flow chamber.

Note that as long as environmental factors remain small, the acoustic signal *TOF difference* (the numerator of Eq. 3-9) has the most influence. Thus, Eq. 3-9 can be further approximated to:

$$v\approx\frac{d}{2\cos\alpha}\left(\frac{\Delta t}{t_{avg}^2}\right) \qquad \text{Eq. 3-10}$$

Where: Δt TOF difference, equal to $t_{B\to A}-t_{A\to B}$
t_{avg} average TOF transit time

Eq. 3-10 is consistent with the approximations discussed in [21] and [22]. This approximated form of ultrasonic TOF flow measurement is used in the current final prototype software (see Appendix E). The approximation permits simplification in coding for the first firmware as t_{avg} is taken to be a fixed value (45us), and the TOF difference Δt is the measurement passed to the PC from the sensor electronics in a single byte. The final prototype described in Chapter 5 operates well and exhibits immunity to humidity and temperature for small airflows – although the flow measurements have not yet been tested with a calibrated airflow source (see section 1.4.2). If necessary, the more general flow relationship (Eq. 3-9) can be included in a future firmware upgrade. At present, Eq. 3-10 is sufficient to measure airflow without significant environmental distortion.

3.1.5 Simultaneous vs. Sequential Pitch-Catch

The bi-directional measurement discussed above consists of two acoustic signals that are "pitched" and "caught" in opposite directions. In the case of *simultaneous* pitch catch, both transducers are activated at the same time. The acoustic signals propagate across the flow

channel, cross near the middle, and are received at approximately the same time (separated by only a Δt). *This simultaneous_pitch catch_method is used by the first two prototypes detailed in Chapter 4.* It works well when the transducers are spaced far apart. The method is easier to implement electrically (no microcontroller required), and conceptually easier to understand. Figure 4-7 is a beautiful illustration of simultaneous pitch-catch.

The problem is that the transducers act like tuned, high-Q resonators. Once activated by an electrical impulse, they require substantial time to "ring down." If two transducers are placed close together and pulsed simultaneously, then they will still be in the process of ringing down when they need to be receiving an acoustic signal from the opposite transducer. (This problem is also exacerbated by the 80dB disparity between the transmit signal and the receive signal.)

The solution is *sequential* pitch catch. This is employed by the final prototype (discussed in Chapter 5), and allows the transducers to be placed much closer together. A microcontroller coordinates pitching and catching. Only one transducer transmits at a time. Assume transducer A starts first. While A is still ringing down, A's signal is received by B. Soon after B catches the signal, B transmits a signal back to A, which by this time has finished ringing down and is ready to receive. Those two pitch catch sequences, from A to B and B to A, constitute a complete TOF measurement with enough information to satisfy Eq. 3-9 or Eq. 3-10. The final prototype can complete each pitch catch in 100μs, for a total measurement time of 200μs. Although not simultaneous, the two pitch catch sequences happen in such rapid succession that any change in environmental factors or air velocity is negligible.

3.2 TOF in contrast to Doppler Ultrasound

Ultrasonic TOF is distinctly different from Doppler ultrasound. A brief explanation is included here to distinguish the two flow measurement technologies, which both use ultrasound, but rely on different physics. TOF probes the velocity of a flowing medium by measuring the *propagation time* of ultrasound *through a medium*. This principle is shown in Figure 3-3. Doppler ultrasound measures frequency shifts of the backscatter or *reflection* of an ultrasonic signal from *particulates carried in the medium*. Because of this important difference, TOF is better suited for more uniform "clean" flows, such as clean liquids and most gasses. Doppler is

preferred in situations where the flow is acoustically contaminated, by solid particulates or even air bubbles in a liquid.

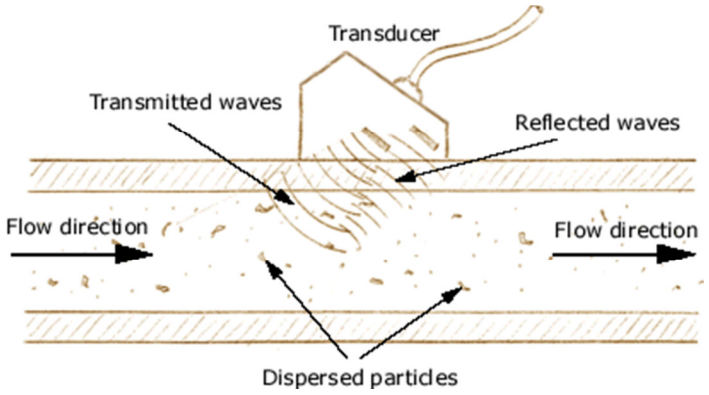


Figure 3-3: Principle of operation of Doppler ultrasonic flow meter (image courtesy [23])

In TOF measurements, the frequency of the acoustic signal remains constant. In Doppler ultrasound, the frequency of signal is shifted in proportion to the velocity of the object from which it is reflected. This frequency shift is given by the classical Doppler shift formula:

$$\Delta f = \frac{2 v f_0 \cos\theta}{c} \qquad \text{Eq. 3-11}$$

Where:

	Δf	frequency shift of reflected signal
	v	velocity of particles in the flow stream
	c	speed of sound in the medium
	f_0	transmitted carrier frequency
	θ	orientation angle of transducer relative to flow direction

Doppler ultrasound has been used extensively for measuring blood flow, where red blood cells scatter high frequency ultrasound. [24] However, this method is unsuitable for measuring flow in uniform fluids, which is why ultrasonic TOF was chosen as a sensing method.

3.3 Mechanical background – acoustics

Mechanical and electrical systems are analogous. Both concern energy transfer. Energy, power (energy per time), intensity (power per area), are common features both systems. Much of the material presented within this chapter about mechanical phenomena, as it relates to acoustics, can also be used to understand electrical systems.

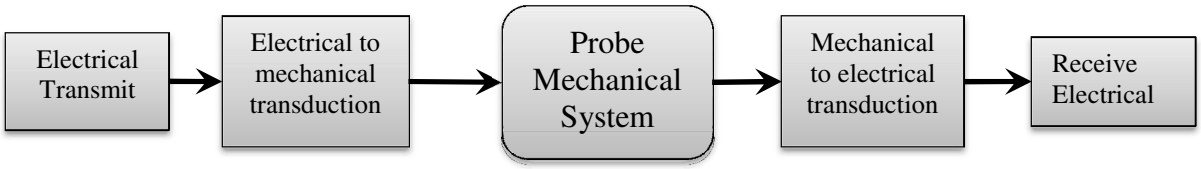


Figure 3-4: Stages of energy transfer within the ultrasonic measurement

Figure 3-4 shows the stages of energy transfer as the ultrasonic system probes air velocity. An electrical signal in the form of an impulse is first issued from the transmit electronics stage. The transducer, whose active element is a disc-shaped piezoelectric crystal, resonates at a given frequency in response to the electrical impulse. The majority of energy is dissipated within the transducer, but a small fraction couples to the air (primarily due to the transducer's matching layers) and propagates across the flow chamber where it is incident upon the second transducer. The transduction process is reversed, from mechanical to electrical. The receive electrical signal is generated. Since it is much weaker than the original transmit signal, the receive signal must be amplified for further processing.

3.3.1 Motivation for understanding acoustic principles

Ultrasound is the probing mechanism that allows us to query the flow channel and measure velocity using time of flight principles. Transducer size will be a limiting factor in further reducing sensor profile. Custom transducers will be needed. Thus, it is necessary to review acoustics and transducer design to not only to understand how the device currently works, but also to plan for a future device. Of additional interest are the effects of contaminants in the flow channel, such as condensed saliva, mucus, and blood.

3.3.2 Acoustic wave equation

Ultrasound is a mechanical wave propagating through a physical medium with a frequency above that of human hearing (> 20kHz). Sound travelling through a solid medium with constrained particles can propagate according to multiple modes such as longitudinal (pressure), transverse (shear), and Rayleigh (surface). [25] Fluid substances with unconstrained particles only support pressure waves, which are the focus of this discussion.

The linear wave equation provides an adequate starting point to establish properties of pressure waves. A particle constrained to move in one dimension in a lossless medium satisfies the 1D wave equation [26]:

$$c^2 \frac{\partial^2 u}{\partial x^2} - \frac{\partial^2 u}{\partial t^2} = 0 \qquad \text{Eq 3-12}$$

The familiar sine and cosine functions, and the complex exponential are solutions to the wave equation [26]:

$$u = A \sin a(x - ct)$$
$$u = A \cos a(x - ct) \qquad \text{Eq 3-13 (a,b,c)}$$
$$u = Ae^{\pm i\omega(x-ct)}$$

Many acoustic problems, including transducer design and matching layers, can be solved by finding analytic or numeric solutions to the wave equation (which may be modified with additional terms to suit the problem at hand) affixed to appropriate boundary equations. For problems with intricate geometry, solving the three dimensional wave equation is best left to computer numerical solutions using a technique such as finite element analysis. In our application, which is primarily concerned with narrow beam acoustic propagation between two aligned transducers, the one-dimensional (1D) wave equation is sufficient and easier to work with. Issues such as beam spread can be modeled as a path loss.

3.3.3 Sound pressure, velocity, and acoustic impedance

Mechanical waves propagating through an otherwise undisturbed material at equilibrium can also be characterized by the magnitude of the pressure change in zones of compression and rarefaction. This is *sound pressure*, and it usually specified by the root-mean-squared (RMS) amplitude of the sound wave in the SI unit Pascal (N/m^2). A common reference sound pressure is 20μPa, which is considered the threshold for human hearing.

Sound pressure differences are perceived according to a change in the order of magnitude, making sound logarithmic in nature. *Sound pressure level* (SPL) is the measure of the ratio of sound pressure relative a given reference level. To calculate the SPL of an acoustic signal with RMS pressure amplitude relative to a reference pressure:

$$L_{sig}(dB) = 10 \log_{10}\left(\frac{{p_{sig-rms}}^2}{{p_{ref}}^2}\right) = 20 \log_{10}\left(\frac{p_{sig-rms}}{p_{ref}}\right) \qquad \text{Eq 3-14}$$

The velocity of sound c is dependent on the properties of the material. It can be derived from a state equation relating the partial derivative of pressure p with respect to density ρ [27]:

$$c = \left(\frac{\partial p}{\partial \rho}\right)^{\frac{1}{2}} \quad \text{Eq 3-15}$$

A related property is acoustic impedance, Z. Acoustic impedance (which has common CGS units of Rayls) is a measure of how sonically hard a material is, and is given by the equation 3-6 [26]:

$$Z = \rho c \quad \text{Eq 3-16}$$

Acoustic properties of relevant materials and substances are given in Table 3-2.

Table 3-2: Acoustic impedance and sound velocity in select materials

Material	Sound Velocity (m/s)	Acoustic Impedance (Rayls)
Air	343	415
Water	1500	1.5×10^6
Neoprene (backing) [28]	1600	2.1×10^6
PZT5H Series[28]	3800	28×10^6
PZT500 Series[28]	4350	33.7×10^6
PZT800 Series[28]	4600	35×10^6

The characteristic acoustic impedance of a material is analogous to characteristic impedance in electrical systems. A propagating wave that encounters an impedance change will be partially reflected and partially transmitted.

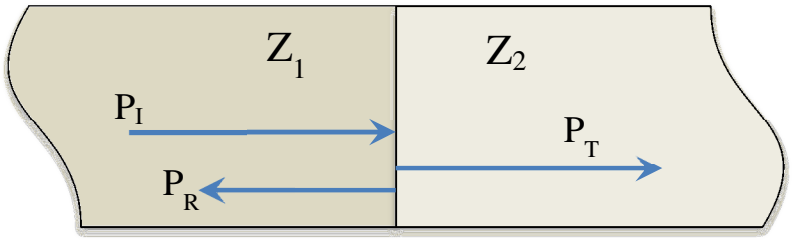


Figure 3-5: Incident acoustic power transmitted and reflected at an impedance boundary

Two materials of different acoustic impedances meet at a boundary, as shown in Figure 3-5. The transmission and reflection coefficients, or fraction of incident energy transmitted and reflected, for an incident wave such are given by the following equations [29]:

$$T = \frac{4Z_1Z_2}{(Z_1+Z_2)^2} \quad \text{Eq 3-17}$$

$$R = 1 - T = \left(\frac{Z_1 - Z_2}{Z_1 + Z_2}\right)^2 \quad \text{Eq 3-18}$$

Using the above equations and Figure 3-5, the transmission coefficients expressing the fraction of acoustic power propagating from a solid PZT5H series material to water and air are:

$$T_{PZT\rightarrow Water} = \frac{4 \times (28 \times 10^6) \times (1.5 \times 10^6)}{(28 \times 10^6 + 1.5 \times 10^6)^2} = 0.19 \quad \text{Eq 3-19}$$

$$T_{PZT\rightarrow Air} = \frac{4 \times (28 \times 10^6) \times 415}{(28 \times 10^6 + 415)^2} = 0.000059 \quad \text{Eq 3-20}$$

$$T_{Water\rightarrow Air} = \frac{4 \times (28 \times 10^6) \times 415}{(28 \times 10^6 + 415)^2} = 0.0011 \quad \text{Eq 3-21}$$

Using Eq 3-14, it is possible to determine the *transmission loss*, or attenuation of each acoustic interface in decibels (dB). The reference power value is normalized, and the signal value is the fraction of energy transmitted across the interface calculated above (note that since power rather than amplitude is being used, it is appropriate to use the first form of Eq 3-14 without squaring the arguments).

$$A_{PZT\rightarrow Water} = 10\log_{10}(0.19) = -7.2dB \quad \text{Eq 3-22}$$

$$A_{PZT\rightarrow Air} = 10\log_{10}(0.000059) = -42dB \quad \text{Eq 3-23}$$

$$A_{Water\rightarrow Air} = 10\log_{10}(0.0011) = -30dB \quad \text{Eq 3-24}$$

There is a several order of magnitude difference in attenuation between acoustic transmissions from a PZT solid to water vs. PZT solid to air. The acoustic efficiency of coupling from either solid to air or liquid to air is fundamentally hampered by the blunt fact that air is vastly mismatched to either of these two substances. However, it is important to note that the

transmission line model we have been working with so far is a worst-case scenario, because it assumes a single incident wave upon the boundary of two semi-infinite materials. A practical transducer design, shown in Figure 3-6, demonstrates how a half-wavelength active element and a quarter-wavelength resonant acoustic matching layer can partially overcome these limitations, and achieve better than -40dB performance.

3.3.4 High frequency air ultrasound transducer example design

An ultrasonic transducer is an engineered device that converts energy between electrical and mechanical domains. Through manipulation of active and passive materials and their associated geometries, it is possible design for properties such as center frequency, bandwidth, resonance factor (Q), acoustic directivity (beam pattern). [30] It is also possible to partially overcome the coupling limitation of -40dB calculated in Eq 3-10 and Eq 3-13. Transducer design is a complex and involved topic beyond the scope of this thesis. However, a simplified example transducer design will be presented here for reference.

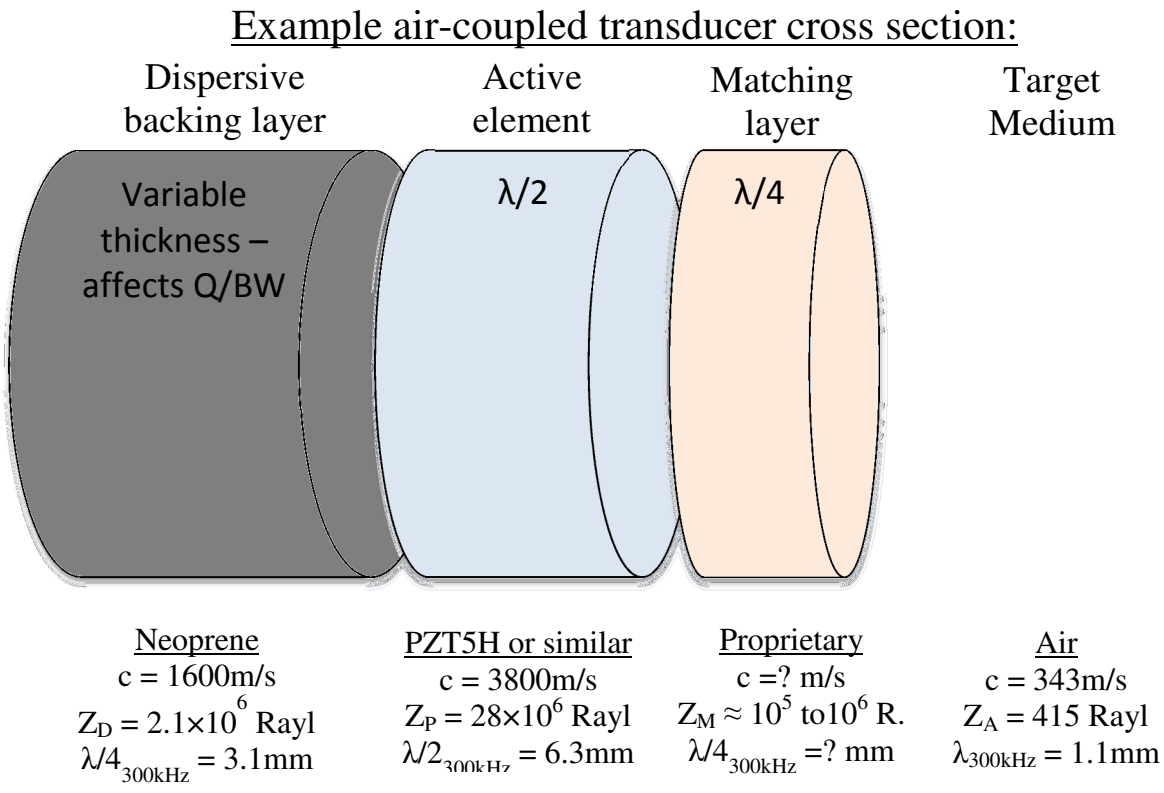


Figure 3-6: Simplified ultrasonic transducer exploded view, with plausible material values given for construction of a narrow bandwidth 300kHz transducer

Three layered elements common to many air-coupled transducers are shown in Figure 3-6. Example material values and calculations are listed below the figure that would allow the transducer to operate at a frequency of 300 kHz with a relatively narrow -3dB bandwidth (similar to the Airmar model AT300 transducer used in this research, explained in Chapter 5). At the center of the transducer is the active element, a piezoelectric crystal disc operating in thickness mode. An electric field applied across the flat, capacitor-like ends of the disc causes axial strain along the same orientation of the electric field (known as piezoelectric 33-mode) With no other layers, the half-wavelength thick crystal disc exhibits resonant compression expansion behavior (high Q factor) for a very narrow bandwidth. [28] The quarter wavelength matching layer has acoustic impedance with an ideal value that is the geometric mean between the active element and the target medium. [29] Finally, the dispersive backing layer defines the tradeoff between Q and bandwidth of the transducer. Generally speaking, closer impedance matching of this layer to the impedance of the active element will yield wider bandwidth / less Q and vice versa for greater impedance mismatching of the layers.

3.3.5 Acoustic link budget

Given the acoustic coupling of ultrasound to air, it is important to ensure that sufficient signal can be transmitted across the flow channel. In this work, we rely on transmit and receive sensitivity data provided by the manufacturer for calculation of the link budget. This is analogous to using antenna specifications without getting into antenna design theory. An acoustic link budget abstracts the transducer parameters to two main factors:

- Transmit sensitivity: A transducer's sound pressure level (SPL) output for a given electrical input voltage, at a specified distance.
- Receive sensitivity: A transducer's electrical voltage output for a given SPL, at a given distance.

Link budget calculations were performed for AT300 transducers using specifications available from [31]. This calculation was verified with experimental observations. A +/-12V input to Figure 3-4 resulted in a +/-1.2mV output, a link loss on the order of -80dB. The weak receive signal can still be amplified, filtered, and used for acoustic TOF measurement.

Chapter 4. Early sensor development

Chapter 4 details two early design stages and their verification, which were crucial to the iterative design process. Design of final sensor is featured in Chapter 5.

The two prototyping stages preceding the final ultrasonic airflow sensor both use a *simultaneous* pitch catch scheme (explained in section 3.1.5). Each stage will be discussed with a presentation of the goal, a description of engineering steps taken to accomplish that goal, and verification of functionality.

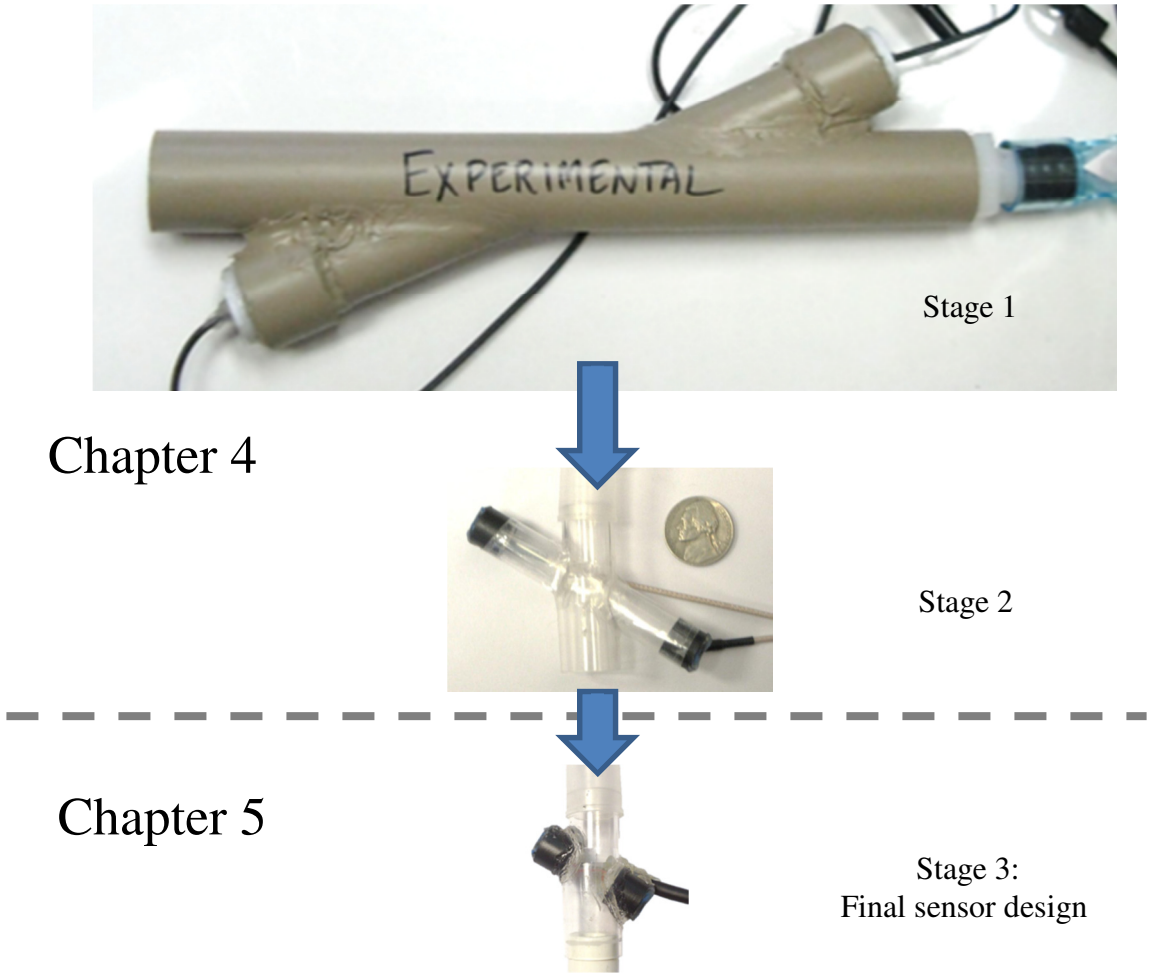


Figure 4-1: Reducing sensor size from two early prototypes to the final, compact prototype

4.1 Developmental stage overview

Stage 1: The proof-of-concept prototype was approached with the goal of learning ultrasonic TOF. The author sought rapid verification that he would be capable of interfacing with transducers and make observations of TOF on an oscilloscope. Using a large flow chamber with oversized transducers, and circuits prototyped on a breadboard, this goal was accomplished.

Stage 2: For the ultrasonic TOF device to have clinical usefulness, dimensions of the flow sensor needed to be reduced by at least an order of magnitude. Compact air-coupled ultrasonic transducers were incorporated into a flow chamber with standard 15mm endotracheal connections, making the new sensor compatible with medical respiratory equipment. A new electronic architecture was developed for transducer interface; including a high-voltage pulser chip driven by a logic circuit, transmit/receive switches, preamplifiers, and filters. Using a simultaneous pitch-catch scheme, acoustic signals were amplified/filtered in parallel and passed to an external oscilloscope which measured the Δt in arrival time. A host computer connected to the oscilloscope used the Δt values to calculate velocity, volumetric flow, and breath detection.

Stage 3 (Final): In the last iteration (discussed in chapter 5) the goals were to reduce transducer spacing and eliminate the oscilloscope as part of the TOF measurement process. In order to do this, a sequential pitch catch scheme was developed to address transducer ring down. An MSP430 microcontroller with the capability of deterministic, clock-cycle-accurate I/O was added to control pulsing, clamp channels, route analog signals, measure TOF using a hardware timer, and communicate with a host computer.

Table 4-1: Three stages to sensor development

	Stage 1	Stage 2	Stage 3 - Final (Ch. 5)
Primary goals	Proof of concept Familiarization with interface electronics	Reduce sensor size by an order of magnitude Modernize electronics	Reduce transducer spacing On-board TOF measurement (eliminate oscilloscope)
Sensor	Oversized flow channel with large transducers	Use tiny AT300 transducers Modify end-tidal CO_2 channel	Sequential pitch-catch to mitigate ring down
Electronics	Rely on reference designs Breadboard everything	Develop new digital and analog circuitry to interface to AT300s – but keep the phase measurements on an oscope	Embedded control – incorporate an MCU

4.2 Stage 1: Large proof-of-concept prototype

The first prototype, shown in Figure 4-2, was an important stage to become familiar with ultrasonic TOF and transducer interfacing. This stage uses a PVC flow chamber constructed by a former student, and transducer interface electronics loosely inspired by designs such as [32].

The electronics architecture required for a simultaneous pitch catch scheme is reflected in the three separate breadboards shown in Figure 4-2. A single pulse generator circuit excites both transducers. Identical amplifier & filter circuits (on their own separate breadboards) process received acoustic signals from each transducer in parallel.

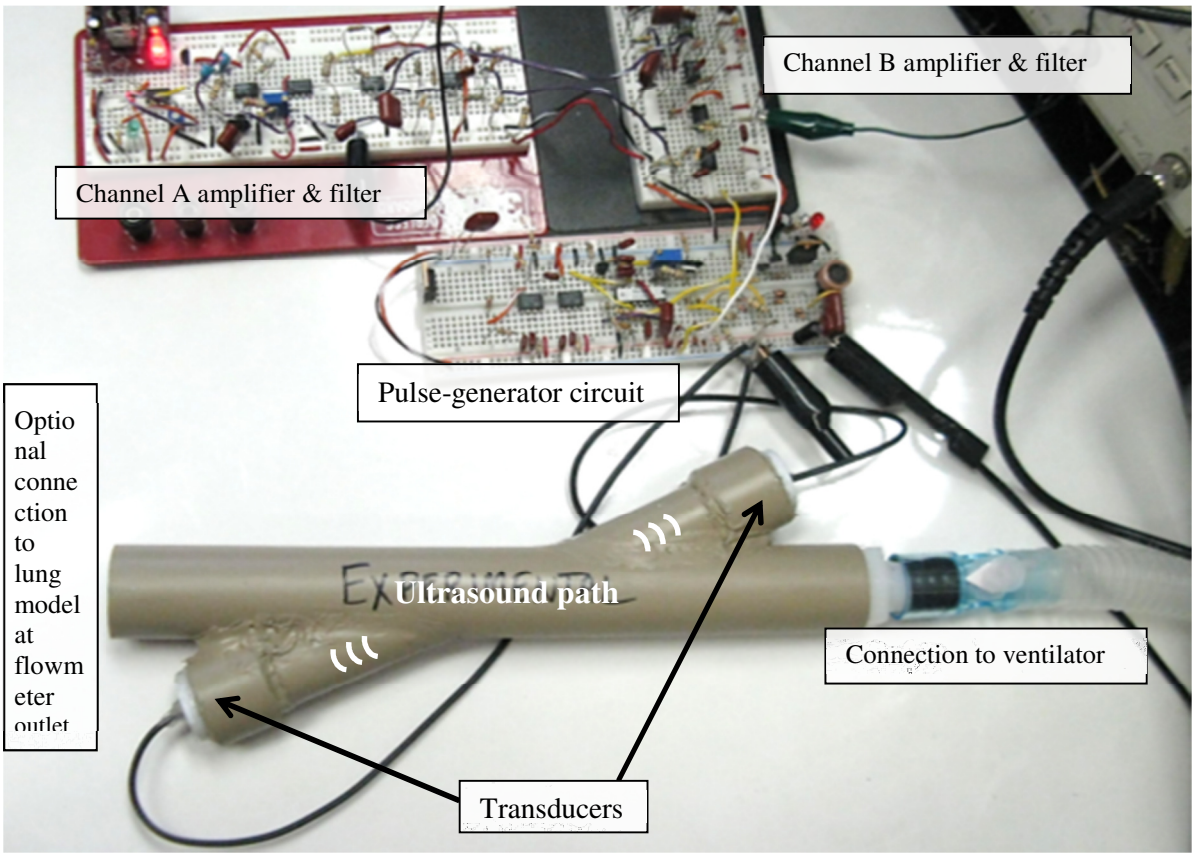


Figure 4-2: Proof-of-concept first generation prototype

The transmit circuit consisted of a cascaded pulse generator connected to a MOSFET, which switched the primary of a fly-back transformer. The receive circuits were protected from transmit stage HV pulses by resistor/diode clamps. Weak receive pulses, which passed through the clamp, were pre-amplified and processed through a 4th order band pass filter.

4.2.1 Stage 1 - oscilloscope waveform verification.

Figure 4-3 (a) and (b) illustrate the output of each receive circuit channel, shown together on the same graph. In the left portion of (a), one can see the start of the interference from the transducer excitation, followed by each transducer's ring down, and then the arrival of the acoustic signal of interest. The oscilloscope graph in (b) is a magnification of the acoustic signals of interest. These signals shift relative to one another during airflow. This observation was a qualitative verification of ultrasonic TOF; however at the time there was no way to quantitatively measure the difference in time of flight.

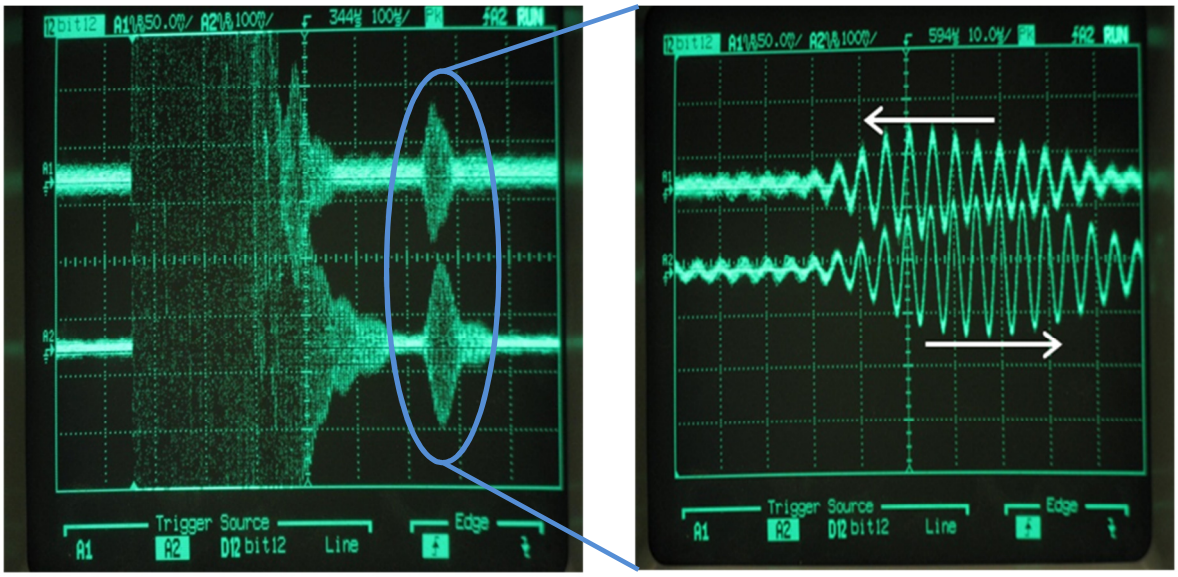


Figure 4-3: (a) Analog receive circuit output for both transducer channels, (b) Acoustic signal Δt shift

In summary, the proof-of-concept achieved its stated goals of qualitatively demonstrating a working ultrasonic TOF system and familiarizing the author with electronics related to ultrasonic TOF. After confirmation that we could excite, receive, and amplify signals from ultrasonic transducers simultaneously, and observe phase shift on an oscilloscope, we preceded to planning the next prototype.

4.3 Stage 2: Smaller transducers, improved interface electronics

After demonstrating ultrasonic TOF with the proof-of-concept prototype, a second generation prototype was designed with numerous improvements to both the sensor channel and electronics. These goals are summarized in the following lists.

The primary goals set for the second generation prototype were to:

- Reduce sensor size by one order of magnitude
- Develop quality transducer interface electronics
- Quantitatively measure ultrasonic TOF

Compromises were still made at this stage, namely:

- No embedded controller yet – all pulsing performed by discrete timer chips.
- External oscilloscope & PC connection required for TOF difference / flow measurement.

The second generation prototype was undertaken as an iterative step towards the ultimate goal of creating a self-contained, compact sensor, capable of functioning independently and reporting TOF or flow measurements to an external computer.

4.3.1 Sensor Physical Design

The flow chamber sensor was constructed by modifying an existing end-tidal CO_2 adaptor tube, which measured 1.8cm overall diameter by 4.7cm in length. The existing adaptor is a small polycarbonate attachment designed to be placed in-line with a ventilator circuit. It comes with standard male/female 15mm endotracheal connections at both ends. The adaptor tube was modified by drilling a 1cm diameter hole through the walls of the flow chamber at a 60° angle parallel to direction of airflow.

Initially, the AT300 transducers (explained in Chapter 5) were mounted directly in the tube, until it was clear that the current electronic circuit and high transducer Q value (also a product of the simultaneous pitch catch scheme) would set a boundary on the minimum transducer spacing to 5.5cm. To accommodate the extended acoustic pathway, tubular extensions were added to the holes already in place. The second-generation sensor device is shown in Figure 4-4.

Figure 4-4: Second-generation prototype sensor connected to a doll with tracheostomy tube

4.3.2 Stage 2 electronics

The main objectives of the electronic circuit for the stage 2 prototype are to miniaturize the circuit, improve the reliability and repeatability of the transmit stage, and increase the sensitivity of response and adjustability of the receive circuit. It was decided to keep the post processing (phase/delay detection of contra-propagating acoustic signals) and further calculation of volumetric flow rate external to the prototype circuit.

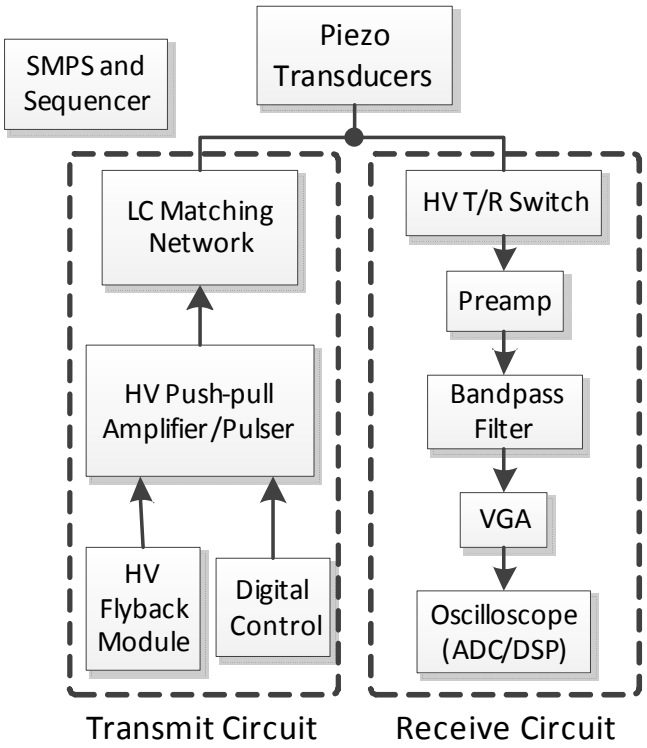


Figure 4-5: High-level block diagram of 2nd generation sensor circuit

A block diagram of the second generation sensor circuit is shown in Figure 4-5. Details of the circuit will not be discussed here – many components, including the digital pulser, transmit receive switches, band pass filter, etc. are common to the final sensor circuit discussed in Chapter 5. The stage 2 was prototyped on a four layer PCB, shown in Figure 4-6.

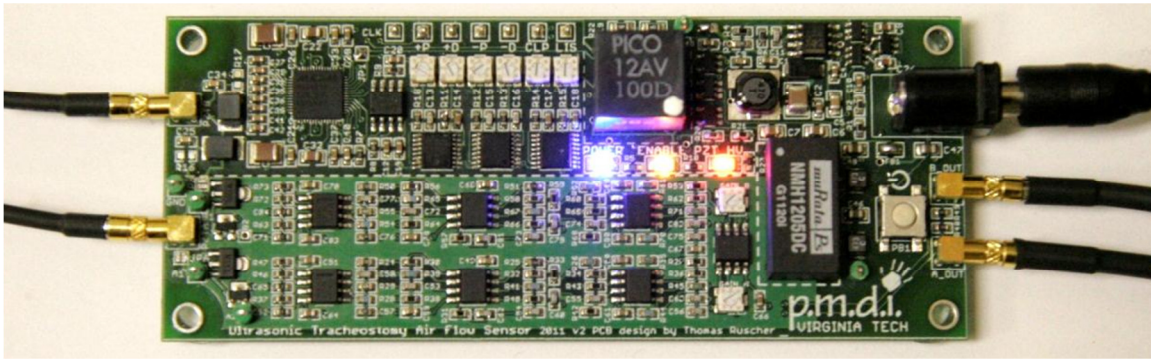


Figure 4-6: Stage 2 prototype circuit PCB.

4.3.3 Response and ring down

Figure 4-7 shows inputs and outputs from the second generation circuit. One can clearly see the simultaneous transducer excitation – the clipping on the preamplifiers, followed by ring down, then near simultaneous arrival of the acoustic pulses. The ring down period is approximately 150µs in duration due to the high Q design (minimal acoustic dampening) of the transducers, an issue described in literature [33], and in Chapter 3.

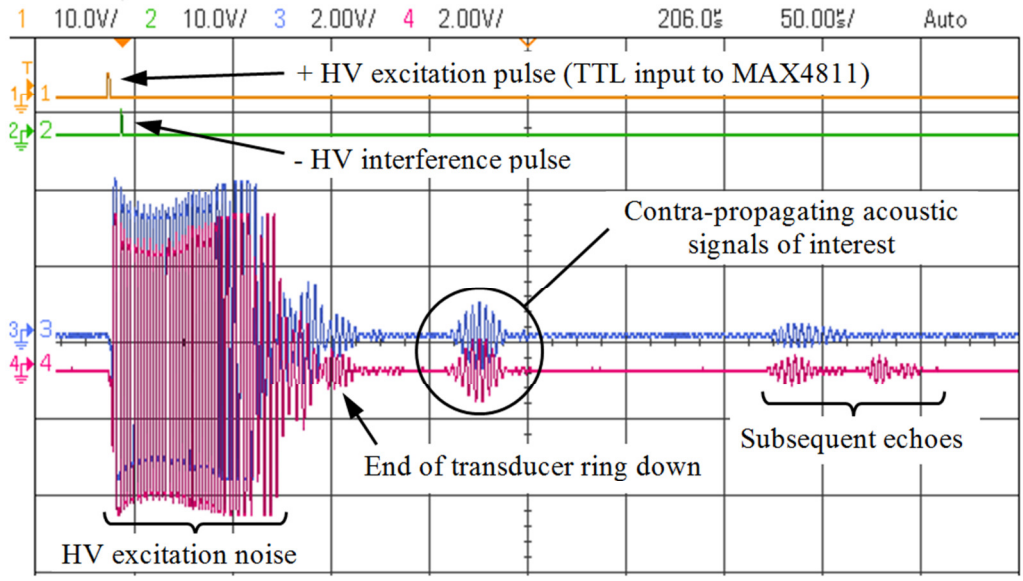


Figure 4-7: Signals measured from the second generation prototype sensor and circuit (channels 1 and 2 are digital inputs to the MAX4811, channels 3 and 4 are signal-conditioned waveforms from transducers A and B, respectively)

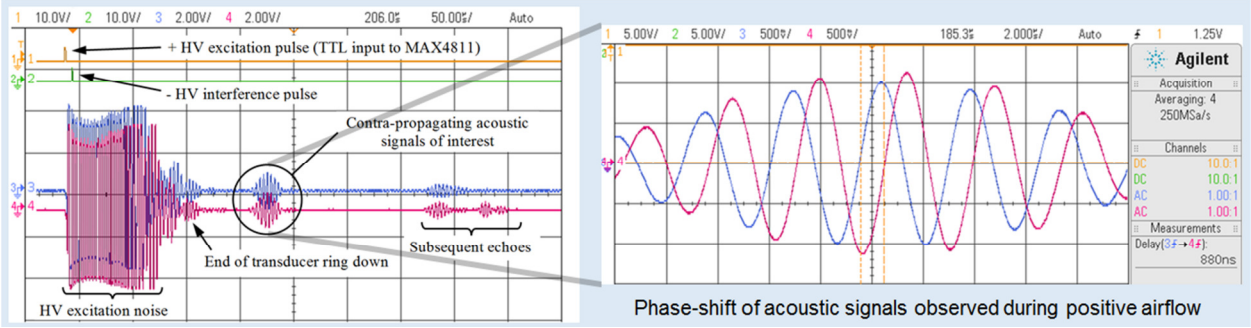


Figure 4-8: Phase shift of simultaneously arriving acoustic signals

Approximately 175ns after the initial impulse, the contra-propagating pulses are received by the opposite transducer. Due to near-simultaneous reception, the Δt delay between pulses appears as a phase shift, shown in Figure 4-8 and magnified in Figure 4-9. An oscilloscope performs the delay measurement, shown in the red box in Figure 4-9.

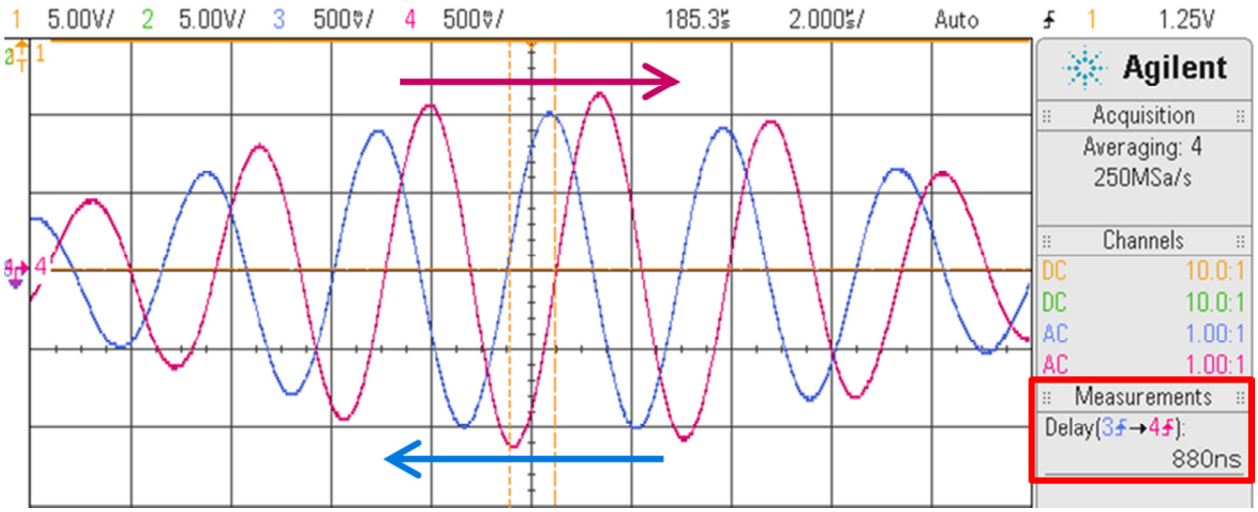


Figure 4-9: Close up of acoustic signal shift and measurement of Δt (boxed in red)

The oscilloscope is connected to a computer running LabVIEW. Δt delay measurements are passed from the oscilloscope to the computer. Eq. 3-10 is used to compute velocity, which is multiplied by the cross sectional area of the sensor channel to obtain volumetric airflow. An integration can be performed over time to detect breaths.

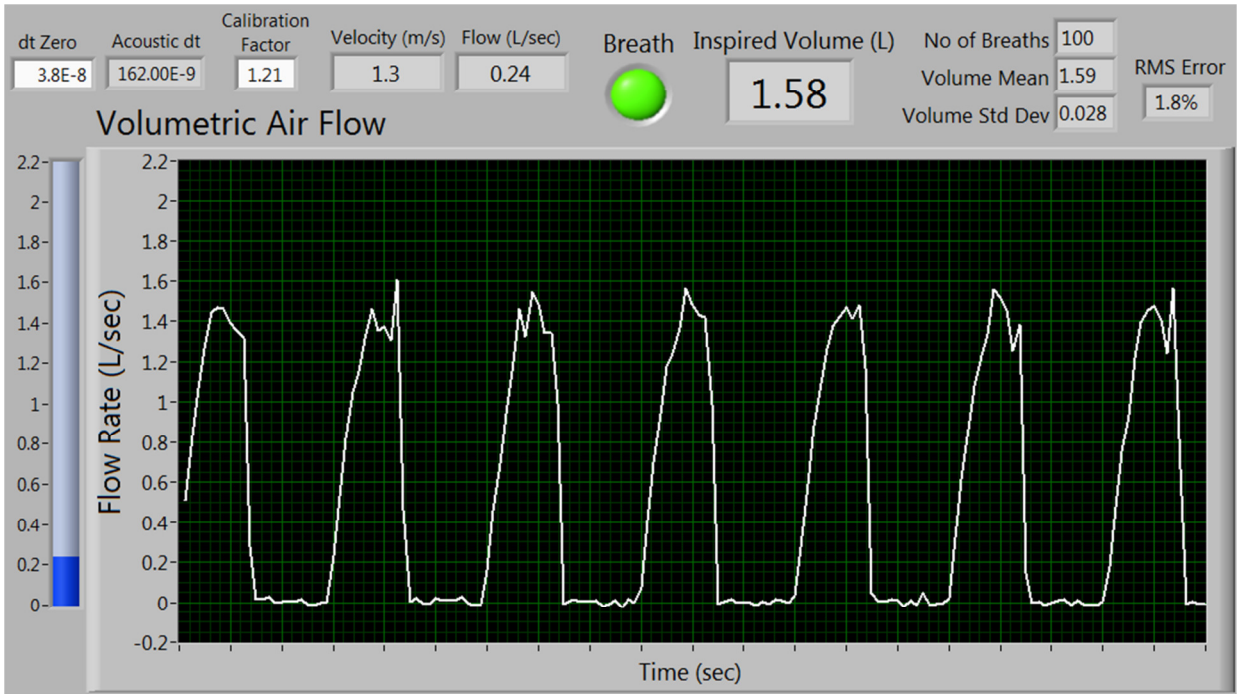


Figure 4-10: A graph of real time airflow through the stage 2 sensor

The stage 2 prototype was connected to a respiratory ventilator, capable of delivering calibrated volumes of air in one direction. The prototype device was able to measure sequences of repeated ventilator "breaths". A sample result is shown in Figure 4-10.

4.4 Summary of early prototype accomplishments

In conclusion, two successful early prototypes demonstrated that an ultrasonic airflow sensor based on the TOF principle could be constructed. Both prototype sensors used a simultaneous pitch catch scheme, which permitted easy verification of TOF – but restricted minimum transducer spacing. In the second prototype, an oscilloscope quantitatively measured TOF and passed Δt values to a computer running a LabVIEW program. The computer used the data and the ultrasonic TOF equation to calculate flow velocity, volumetric flow rate, and breath detection by time integration of volumetric flow rate. The computer also generated a real-time graph displaying airflow through the sensor.

Chapter 5. Final Sensor Design

Chapter 5 details the design of the complete sensor, including the interface electronics, the transducers, and embedded software. The final sensor's major improvement over previous prototypes is the use of a *sequential* pitch catch ultrasonic TOF measurement method (described in 3.1.5), which allows for reduced transducer spacing and overall sensor miniaturization. An MSP430 series microcontroller unit (MCU) was incorporated to coordinate sequential pitch catch sequences, record acoustic arrival times, and communicate Δt TOF measurement values to a host computer. Embedded software for the MCU was developed in C. A photo of the complete sensor layout, including the compact flow chamber and electronics, was presented at the end of Chapter 1 (Figure 1-2).

5.1 Electronics Hardware - Overview

The sensor electronics are shown in Figure 5-1. Detailed schematics and layout can be found in Appendices A and B. The design and architecture of the electronics will be discussed in the following pages.

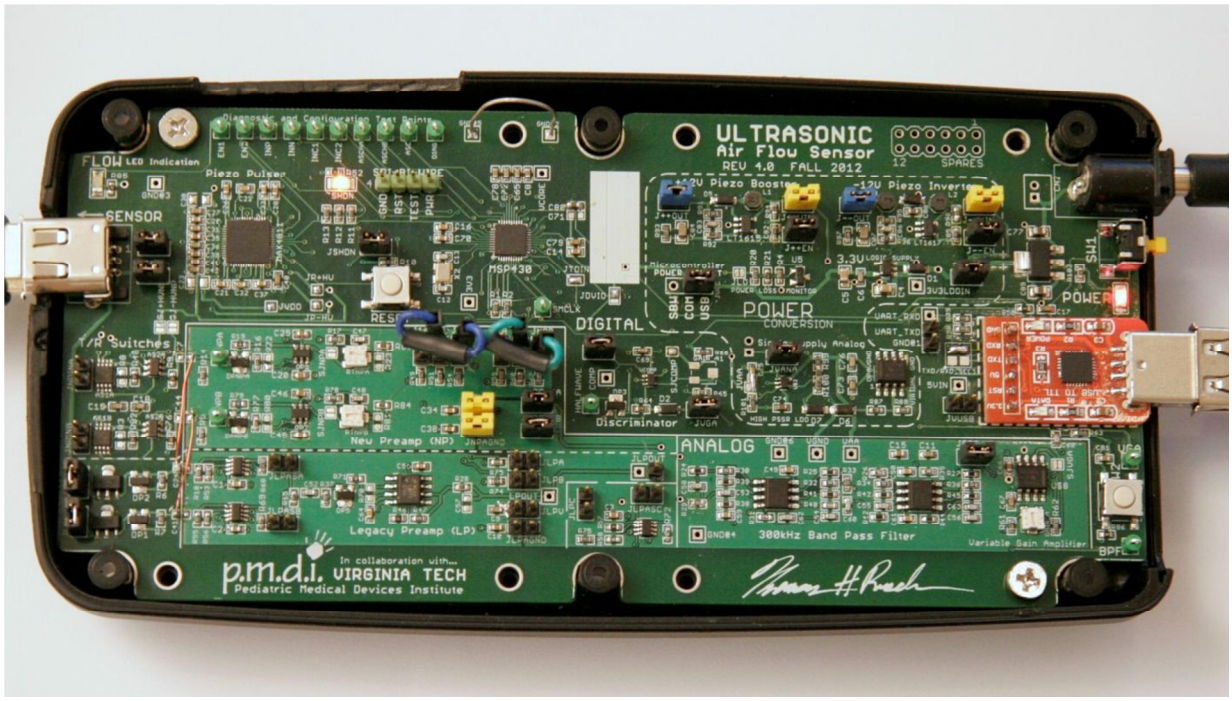


Figure 5-1: Sensor interface electronics PCB

A high-level overview of the electronics is shown in the block diagram in Figure 5-2. The primary components of the system are power, the transmit circuit, the receive circuit, and the embedded controller which coordinates all operations. The microcontroller controls initiation of a measurement, routing of the analog signal by enabling the preamplifiers, and measurement of arrival time an acoustic pulse using a hardware timer.

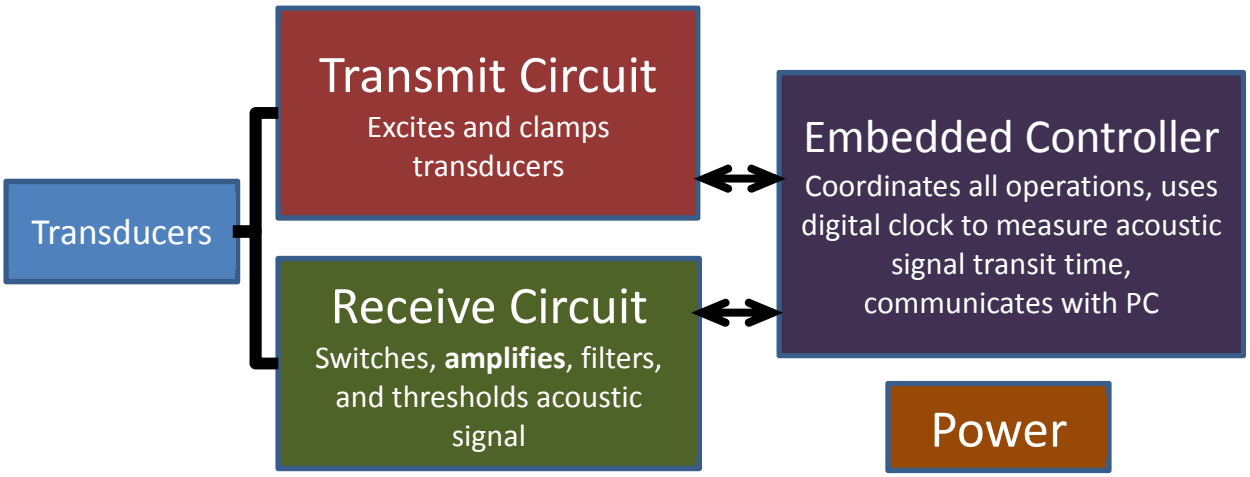


Figure 5-2: High-level overview of sensor architecture

A detailed, technical overview of the electronics is shown in Figure 5-3. The architecture resembles front-end stages found in radar and radio transceivers.

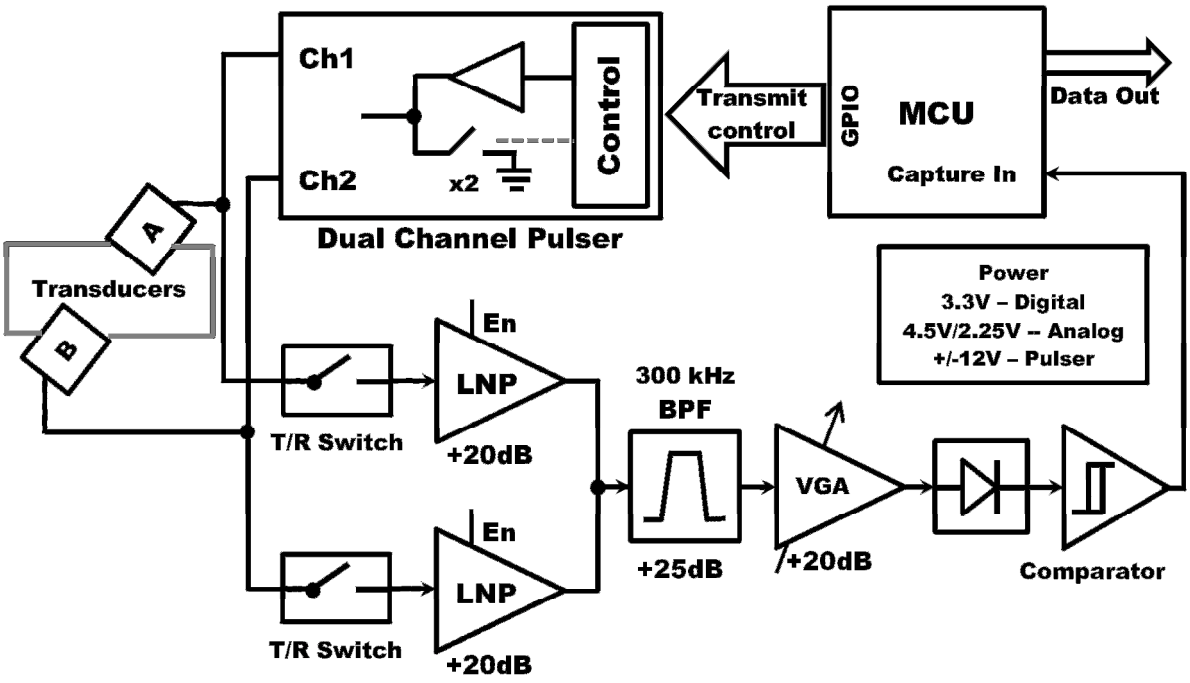


Figure 5-3: Technical electronics architecture for final prototype

5.2 Transmit Circuit

The primary component in the acoustic transmission circuit is the high voltage digital pulser (Maxim Integrated part MAX4811). This IC translates digital control signals into high voltage pulses, capable of operating up to +/-100V (+/-12V pulses are currently used with the AT300 transducers). The driven transducers appear as a capacitive load.

Figure 5-4 shows the internal schematic of the MAX4811. Internally, it consists of a MOSFET push-pull amplifier stage and a MOSFET active clamp. Appropriate signal level shifting, glue logic, MOSFET gate driving, etc., is incorporated. The chip achieves encapsulated, black-box like performance by allowing the user to control all HV circuitry through the use of standard 3.3V CMOS logic signals.

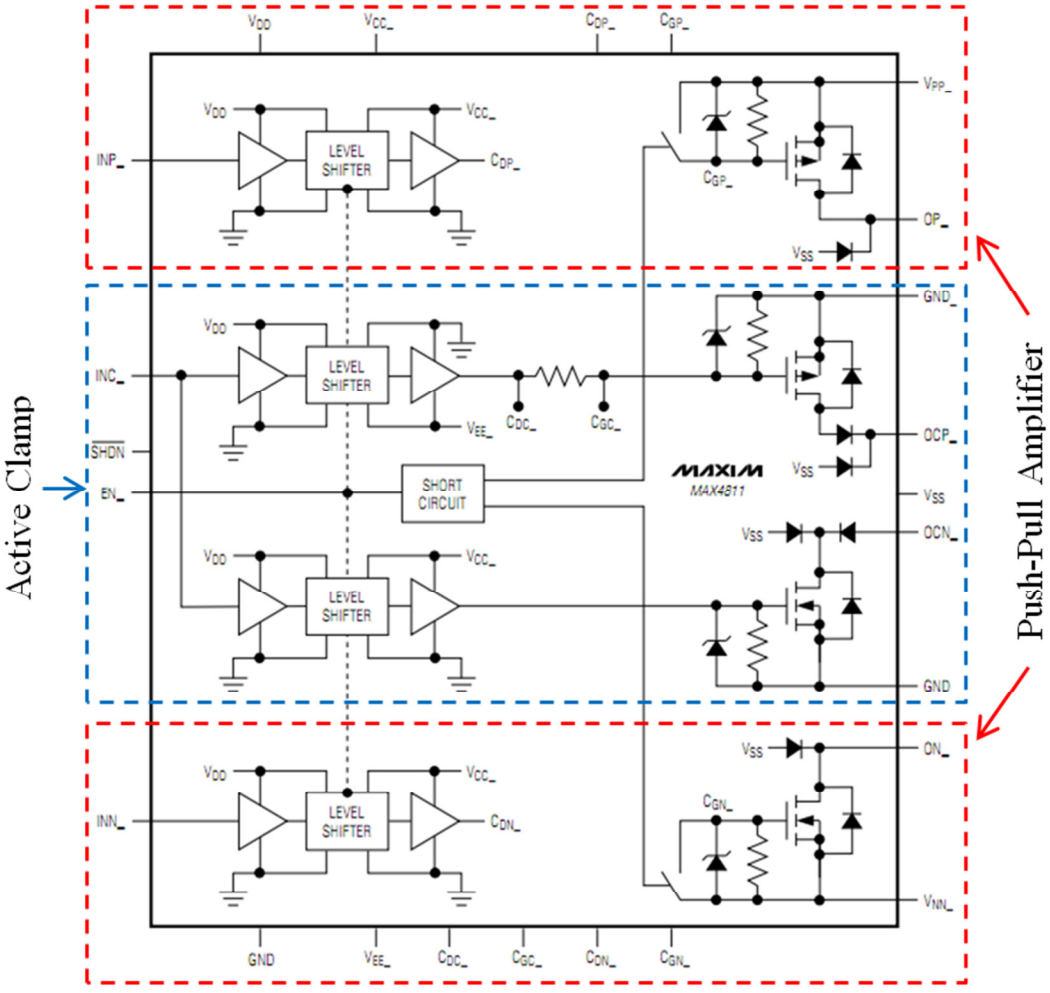


Figure 5-4: Schematic of a single channel of the MAX4811 digital HV pulser [34]

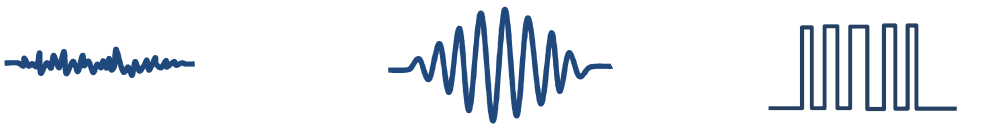
The pulser accepts standard digital control line inputs. These include the abilities to:

- Shutdown the entire pulser
- Enable each transducer channel separately
- +HV pulse each channel
- -HV pulse each channel
- Clamp each channel (creating an internal 27Ω connection to ground)

The active clamp circuit feature helps dampen transducer ringing after excitation. The +/-12V supply inputs come from separate boost and inverting switch-mode power supplies. Decoupling capacitors near the pulser ensure that voltage sag on these lines is minimal when driving the transducers. The MAX4811 is controlled through general purpose input output (GPIO) pins from the MSP430 microcontroller.

5.3 Receive Circuit

The receive circuit amplifies, filters, and transforms acoustic signals. The circuit takes as its input a direct electrical connection to the transducers. After HV blocking and selective pre-amplification, acoustic signals of interest are amplified and band-pass-filtered to reject spurious out-of-band noise. Then, the signal is rectified and an amplitude threshold is applied to select the strongest part of the waveform. After a comparator, the strongest peaks of the acoustic waveform resemble a digital pulse train. This signal is passed to the digital timer module inside the microcontroller, which measures rising and falling edges and calculates the peak arrival time of the acoustic signal. A signal overview shown in Figure 5-5, and an explanation of each stage of the receive circuit follows.



Noisy, weak signal (INPUT from transducer) | Amplified and filtered signal (Intermediate stages) | Timed pulse train (OUTPUT to MCU)

Figure 5-5: Signal processing at various stages of the receive circuit

5.3.1 Transmit-receive (T/R) switches

The first elements in the receive circuit are the HV transmit/receive (T/R) switches, which protect the sensitive analog preamp from high voltage pulses (refer to Figure 5-3). Since the receive circuit is directly connected to the transducers, it must block HV drive signals that would otherwise damage amplifier components. A specialized T/R switch has a major advantage over the diode clamp used in the previous design in that it enters high impedance mode when the voltage exceeds a certain threshold, which permits stronger driving of the transducers. Supertex model MD0100 is a two terminal device (shown in Figure 5-6) that requires only a few components for operation. It was chosen for simplicity of use and low on-state resistance (15Ω).

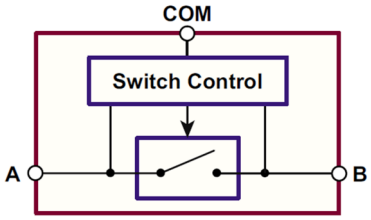


Figure 5-6: Functional block diagram of Supertex MD0100 High Voltage Protection T/R Switch [35]

Figure 5-7 shows that the MD0100 behaves like a resistor with a linear V-I relationship (the slope of this line is R, by virtue of Ohm's law) for low voltages. However, current begins to taper near the saturation current (I_{peak}) and then falls to a steady state value of 200uA (a leakage or operational current), regardless of applied voltage.

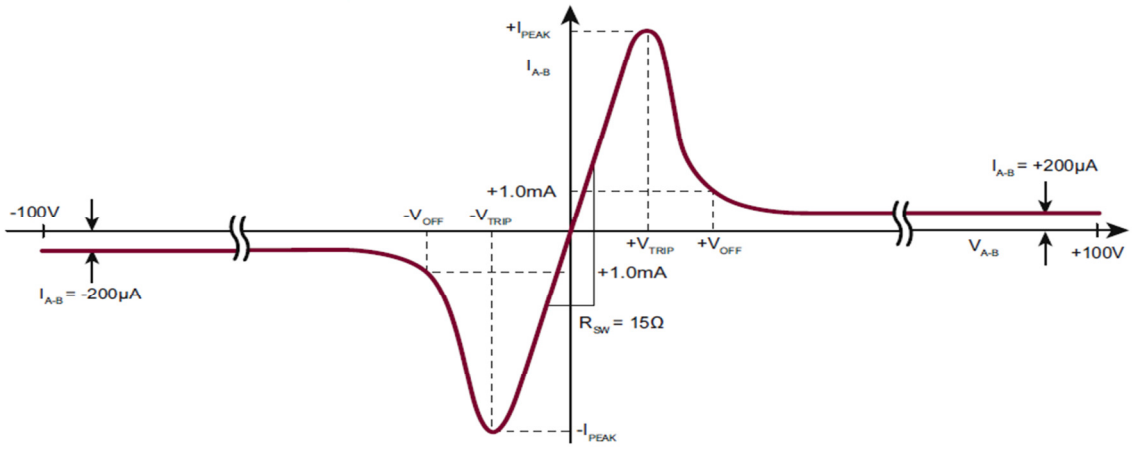


Figure 5-7: V-I curve showing cutoff for MD0100. [35]

In practice, the T/R switch was found to transition to high impedance mode at an approximate level of +/- 1V. Figure 5-8 shows the application reference circuit for the device, which was employed in the sensor electronics.

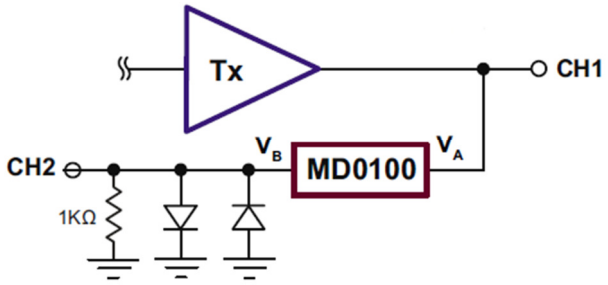


Figure 5-8: Application circuit for MD0100, where Tx is the HV pulser, CH1 connects to transducer, CH2 connects to sensitive receive electronics. [35]

5.3.2 Low-Noise Preamps (LNPs)

In the analog signal chain, the T/R switches are followed by low noise preamplifiers (LNPs), shown circled in Figure 5-9. There is a dedicated LNP circuit for each transducer channel. Since a sequential acoustic transmission/reception scheme is used, rather than a simultaneous, it is possible to multiplex the BPF following the LNPs. Thus, the LNPs function not only as amplifiers, but also as analog switches by virtue of their enable feature.

When an acoustic signal is expected on one of the transducer channels, the LNP for that channel is enabled, so that it can pre-amplify that signal and pass it on to the BPF. Otherwise, the LNPs are left disabled, so that spurious signals (especially the blocked HV signals from the T/R switches, which are still +/-1V in amplitude) do not excite the BPF.

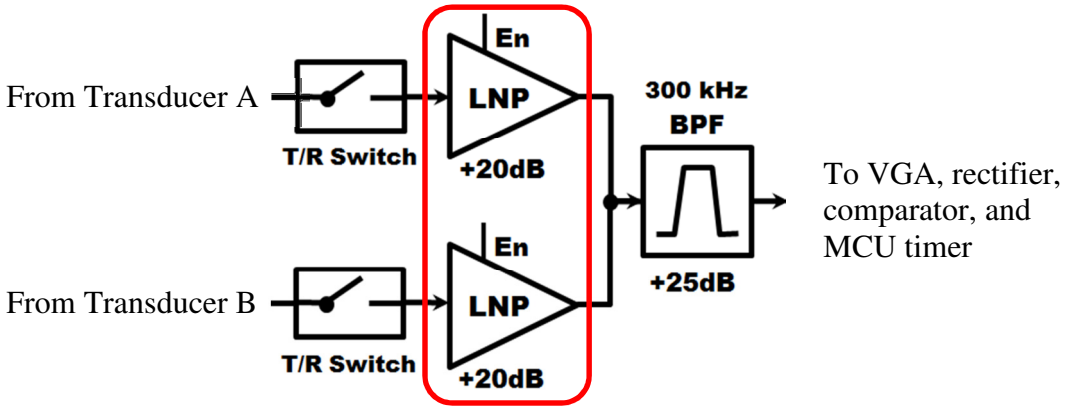


Figure 5-9: Highlight of LNPs in the analog receive circuit

The central part of the LNPs are op-amps (National Semiconductor model LMV791 [36]) configured as standard non-inverting voltage amplifiers. The single channel circuit is shown in Figure 5-10, with a frequency response shown in Figure 5-11. The received acoustic signal is approximately 2mVp-p in amplitude. It is attenuated through the T/R switch stage, and then amplified by the LMV791s. Overall gain for the switch stage and LNP combined is approximately +20dB.

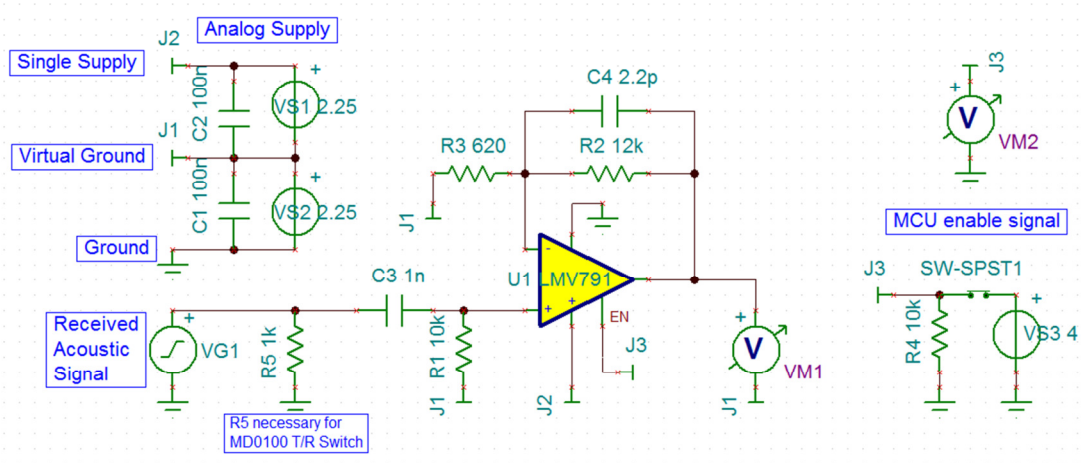


Figure 5-10: Design of analog circuit low noise preamp in TINA (Texas Instruments SPICE software)

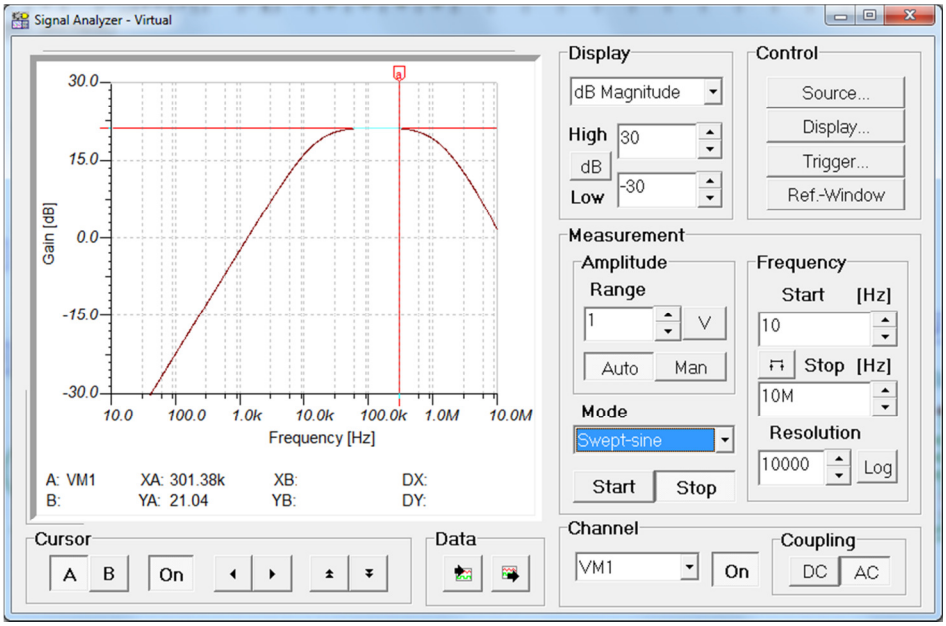
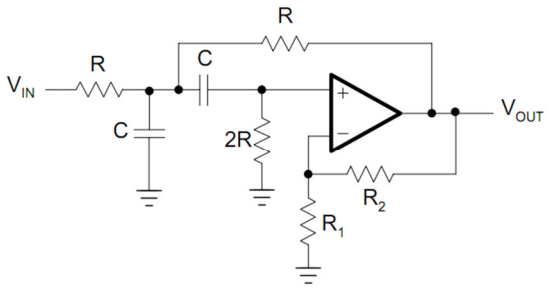


Figure 5-11: TINA simulation frequency response of MD0100 T/R switch and LMV791 LNP circuit

The LNP and much of the other analog circuitry operate on a completely separate power source from all the other electronics – a *single-supply* low noise analog power supply with virtual ground. Shown in Figure 5-10, the signal entering the LNP is DC-decoupled through capacitor C3.

5.3.3 Band Pass Filter (BPF)

A BPF was selected to minimize stray electrical noise at frequencies outside of the narrow frequencies produced by the transducer. Figure 5-12 shows one of four stages in the 8th order linear phase, Sallen-Key topology BPF. The BPF has a pass-band of 300 kHz +/- 30 kHz, pass-band gain of 20dB, and roll off rate of -80dB/decade. Design of the BPF was accomplished using [37] and TI FilterPro software (the design was based on Ref. [37]).



(x4)

Figure 5-12: Single stage in the Sallen-Key band pass filter topology (four cascaded stages with different pole locations comprise the 8th order filter)

For constructing the filter, Texas Instruments OPA2980 op amps were chosen. Details can be found in [38]. These op-amps have sufficiently high gain-bandwidth product for the BPF application. In practice, they exhibited excellent (near ideal) performance.

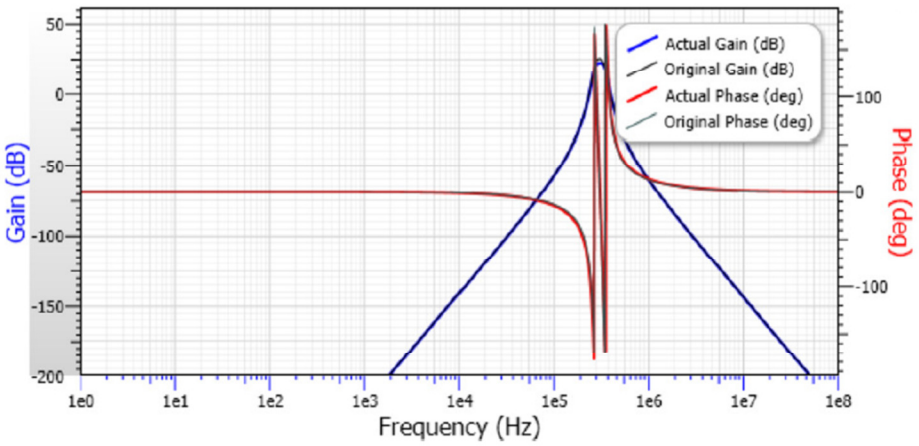


Figure 5-13: Frequency response of receive circuit band-pass filter, from simulation

However, the Q of the filter was found to be an overlooked parameter. Charge injection (brief, glitch-like impulse events) from enabling of the LNP op amps caused ringing of the filter and interfered with the received signal of interest. Future work must correct this.

5.3.4 Variable Gain Amplifier (VGA), Rectifier, and Discriminator

The VGA, rectifier, and discriminator are the final signal processing stages to prepare the signal for input to the microcontroller. The microcontroller hardware timer peripheral module is capable of separately detecting rising and falling edges, but requires standard 3.3V CMOS inputs with fast rising and falling edges. Functionality of these stages is shown in Figure 5-14, and exact component details and circuit architecture area available in the schematic in Appendix A.

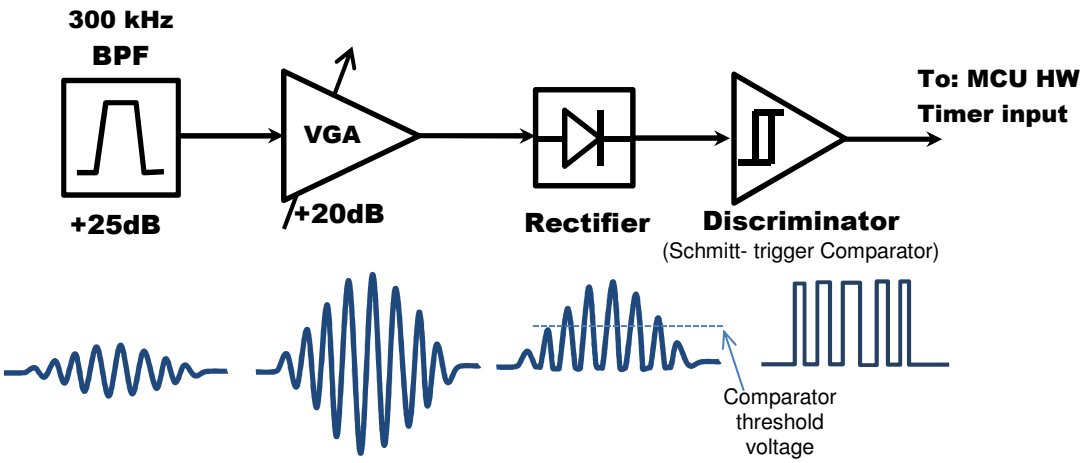


Figure 5-14: VGA, rectifier, and comparator stages, generating the timed pulse train

The VGA was used to tune the received signal amplitude before rectification. It consists of an OPA890 (single version of the OPA2890 used in the BPF – datasheet found at [39]) configured as a standard inverting amplifier, with a trimmer resistor in the feedback stage. For a future revision, an automatic gain control (AGC) amplifier may yield improved results.

Output of the VGA is capacitively coupled to the rectifier and discriminator stage, which are powered through digital (3.3V) rails. The rectifier consists of a single Schottky diode (with low forward voltage drop). The discriminator is a Schmitt trigger comparator (TLV3201 [40]) with fixed reference level. It effectively converts the half-sinusoid waveform to a precision pulse train, with fast rising edges suitable for MCU input.

5.4 Transducers and flow chamber

A comprehensive manufacturer search was performed to find the most compact, commercially available, off-the-shelf, air-coupled piezoelectric transducer suitable for the new prototype sensor. The Airmar Corporation model AT300, shown in Figure 5-15, was found to be the only available air-coupled transducer with dimensions on the order of 1cm. The transducers were originally procured for the stage 2 prototype, before the concept of the link budget was fully developed. However, being the smallest available off-the-shelf transducers, they were kept for the final design and driving electronics were modified to accommodate.

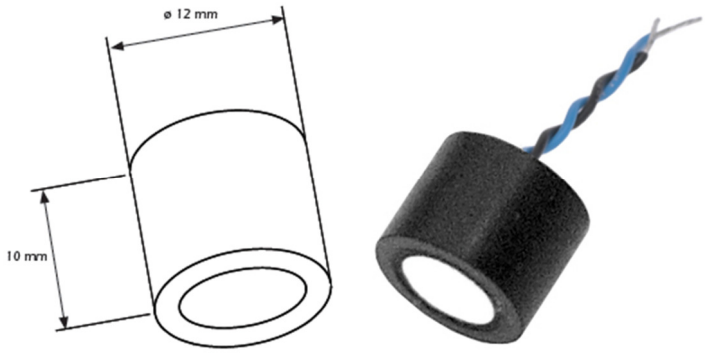


Figure 5-15: Compact Airmar model AT300 transducer [31]

To construct the prototype flow chamber sensor, shown in Figure 5-16, the transducers were mounted in a modified end-tidal CO_2 polycarbonate chamber using a thermosetting plastic (hot glue). Per Figure 3-2 and Eq. 3-9 (which describe ultrasonic TOF geometry), oblique angle to the flow α is 60°, and transducer separation distance d is 15.0 mm.

Figure 5-16: AT300 transducers shown embedded in the final sensor flow chamber, which is attached to a respiratory tube and neonatal tracheostomy tube

5.4.1 Transducer cable shielding and crosstalk reduction

According to the link budget, there is an approximate 70dB discrepancy between the transmit signal level used to drive the transducers and the expected magnitude of the receive signal. The primary concern in a pitch-catch sequence is that the transmit signal will couple to the receive line. This can spuriously excite the receive transducer, causing it to ring, and contaminating or overwhelming the actual signal when it arrives. Also, the ring-down from the transmit channel will couple into the analog preamp on the receive channel, which then excites the BPF.

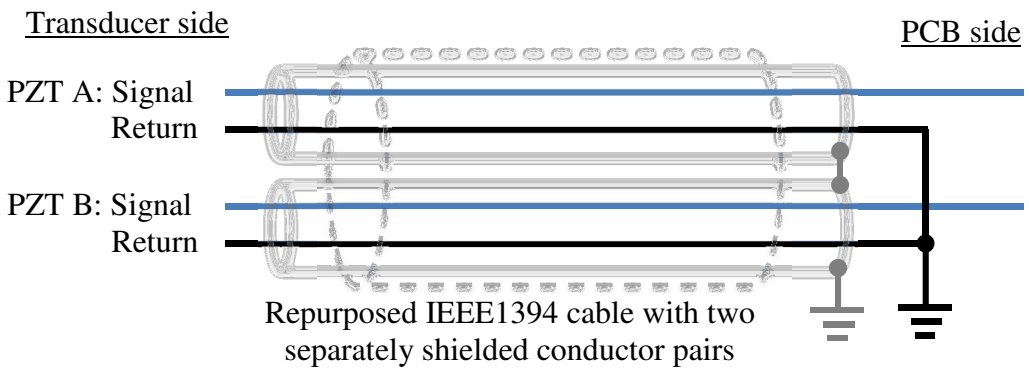


Figure 5-17: Cable connection and shielding scheme used to minimize crosstalk

5.5 MSP430 Microcontroller

The microcontroller is responsible for interfacing with the MAX4811 digital piezoelectric pulser, routing signals with analog switches, and measuring the time-of-flight of the ultrasound signal. These tasks require precise and flexible timing.

To orchestrate the measurement process, a microcontroller from Texas Instruments (TI) MSP430 family was selected. The MSP430F5172 is a 16-bit Von-Neumann architecture MCU with the appropriate peripherals and features to interface with digital and analog ultrasound electronics developed in this thesis. It can be programmed in C using TI's Code Composer Studio (CCS) Integrated Development Environment (IDE). Code is downloaded to the MCU through an USB emulator which is part of a development board called the TI Launchpad.

The MSP430F5172 operates at 25MHz (providing 40ns clock cycle resolution). It features several hardware timer modules, purpose input output (GPIO) pins, and a UART for communication. One reason for choosing this specific MCU model above others in the MSP430 family is that the two Timer_D peripherals can be converted to high resolution 200 MHz timers through the use of an internal phase-locked-loop (PLL) clock multiplier. The idea was that this could be used to get ultra-precise TOF measurements. However, in practice, 40ns clock resolution proved sufficient for TOF measurements.

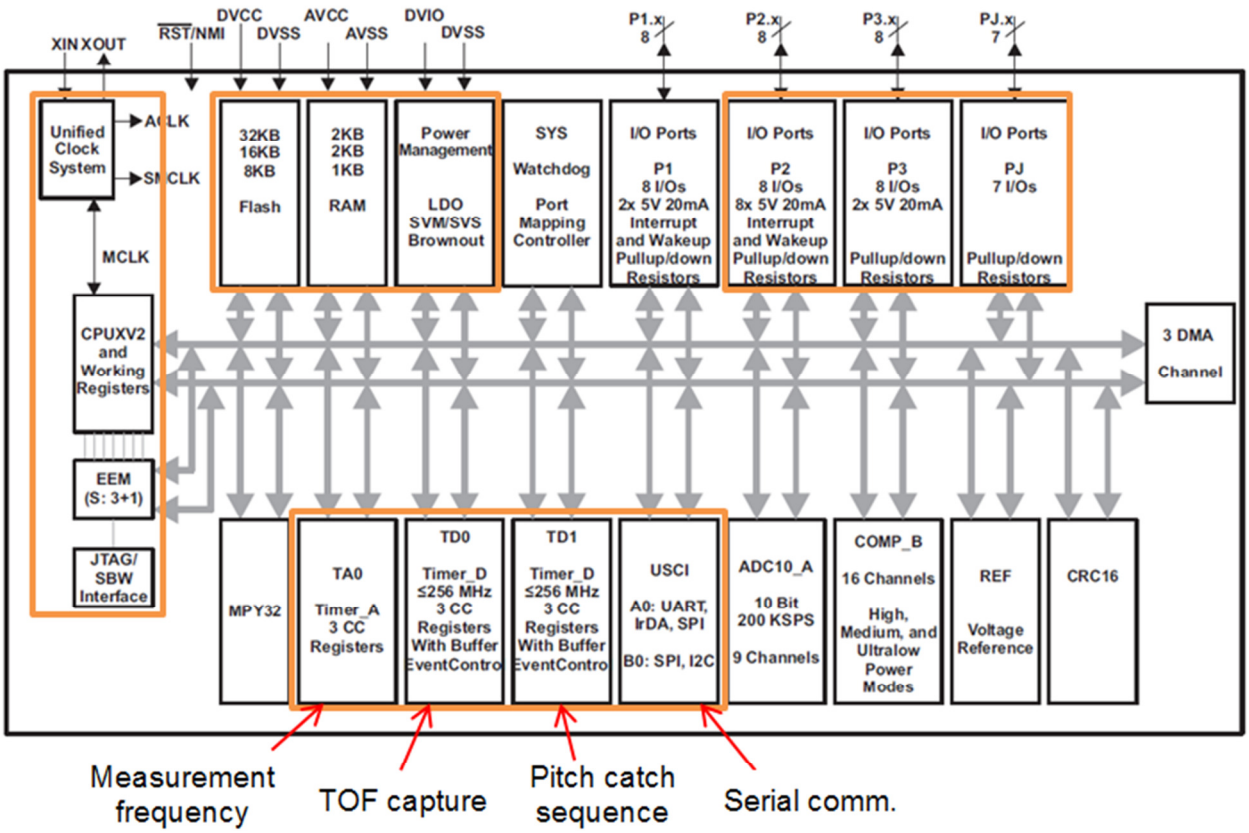


Figure 5-18: Architecture of the TI MSP430F5172 MCU, emphasizing peripheral components that are used in the sensor (diagram from [41])

5.6 MCU firmware

The code performs a variety of tasks. It controls the hardware, captures acoustic input, calculated Δt, and communicates with an external PC. The firmware discussed in this section is TOF_Firmware_4_0, and is provided in Appendices B and C.

Figure 5-19 is a high-level flowchart of the software. The overall mission is for the system to boot itself and enter a super loop, where it continually executes a measurement routine and streams measurements out to a PC. The sequential pitch catch measurement process is emphasized in the "make measurement" block.

After the general software architecture overview, the writing in this chapter becomes quite technical. Familiarity with embedded programming and CPU architecture is suggested.

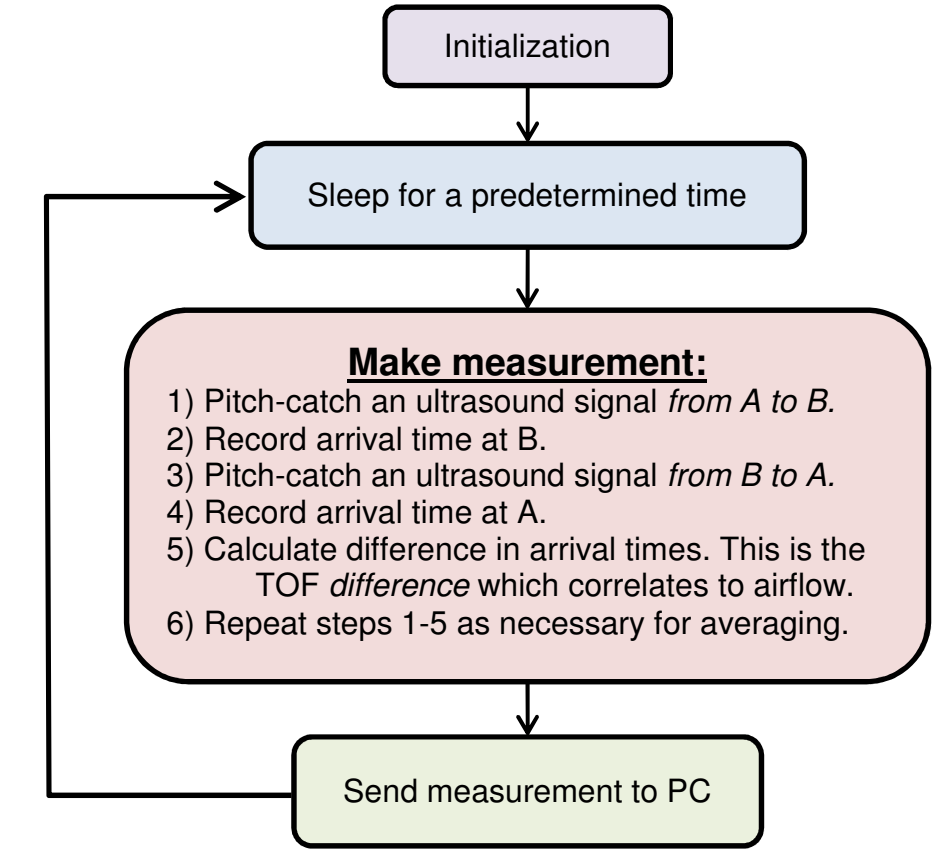


Figure 5-19: High-level software flowchart

The TOF code cannot tolerate any latency – not even single clock cycle jitters. Such jitter would corrupt the time-of-flight measurement. For this reason, the coding style is imperative. Code execution is highly deterministic. Timer interrupts, which signal a timing execution or the reception of an acoustic signal, are always handled from a CPU dormant state (clock disabled gating to the CPU – set by a status register bit), which allows the CPU to wake in a deterministic number of cycles. When timer and CPU-generated GPIO pulses are viewed on an oscilloscope, they are clock-cycle deterministic.

5.6.1 Initialization Routine

The Power-on-reset (POR) initialization routine is responsible for starting the entire device from a complete powered down condition or a system reset. During this time, the system must assert control over critical digital lines that would cause problems if they were permitted to indeterminate states (for instance, simultaneously pulsing the positive and negative inputs of the MAX4811 chip would cause a destructive shoot-through, according to [34]).

The initialization steps accomplish the following sequence:

1. Disable watchdog timer
2. Initialize outputs on general purpose input output (GPIO) Ports 2, 3, and J
    a. First clear output registers (by writing 0x00)
    b. Set pin directions to output, starting with the most critical lines. Those lines would be the ones controlling the MAX4811 Shutdown, enable, and the +/-12V power supplies. Although these lines also have appropriate external pull down resistors, as a matter of good form we assert control over them early.
3. Slowly increase the CPU core voltage in preparation for MCLK increase
4. Start the +/-12V power supplies
5. Initialize the external 25MHz crystal
6. Increase the CPU clock (called Main Clock or MCLK) from >1MHz to 25MHz
7. Declare select TimerD0 parameters related to input capture mode
8. Set up serial (UART) communication
9. Enable momentary pushbutton control
10. Start Timer A, responsible for setting measurement rate through interrupts
11. Enable all mask-able interrupts and wait for event driven execution

The entire initialization procedure occurs in less than one second, which meets user expectations for rapid system start.

5.6.2 Usage of Timer Modules

Hardware timers are critical to the operation of the sensor. They are versatile peripherals capable of triggering periodic interrupts, generating pulse width modulated outputs, or precisely recording the input time of an event. They operate autonomously from the rest of the MCU and are not affected by CPU latency.

A timer can be configured in a variety of modes to suit the needs of its mission. At the core of the timer is the TxR register (where x corresponds to the letter of the timer, such as TAR for timer A and TDR for timer D). This register increments (or decrements) its value by 0x01 on each clock cycle. Other registers, called capture compare registers (CCRs), can compare their value to the value of the TAR register and trigger events when the values match. Examples of

events are an output state change, an interrupt, or even an affect to TAR itself, causing it to overflow or switch from increment to decrement. CCRs can also function in the capture mode. In this mode, they wait for a trigger event, such as a rising pulse edge, and copy the value of TAR when this event occurs.

Timers are used in two main ways in this device. Timer A, a general purpose timer, and Timer D1 are used for precise timing and scheduling. Timer A runs off of the Auxiliary Clock (ACLK). In this application, ACLK is sourced from the main 25MHz crystal and divided by a factor of 32, such that its period is 1.28us. With this period, timer A can be used to set measurement frequency. Timer D0 is used to *capture* the incoming acoustic signals.

Table 5-1: Usage of timers, CCRs, and latches in TOF_Firmware

Timer	CCRs used	Latches used	Mode	Primary function
A0	0	n/a	compare	Set the sensor measurement acquisition rate
D0	0,1	0,1	capture	Record TOF of received acoustic signals
D1	0,1,2	n/a	compare	Schedule the events necessary to a single TOF

5.6.3 Main Super loop execution

In order to make measurements of airflow velocity, the device must measure the TOF difference between upstream and downstream ultrasonic propagation. The basic steps include initiating a transmit, adjusting parameters (such as clamping, switching, etc.) while the acoustic signal is propagating, and finally detecting the arrival time of a single peak of the received acoustic signal. The arrival time is recorded. Then the transducers reverse transmit and receive roles, and the sequence is repeated. At the end of the sequence, the difference between the two arrival times of contra-propagating acoustic signals is calculated.

If measurements are to be averaged for the purposes of noise mitigation before sending a value to the computer, this process can be repeated in a loop. A single *measurement* of the device is defined as the combination of two pitch catches for the calculation of a TOF difference between upstream and downstream acoustic signals.

The entire measurement sequence is contained within the MakeMeasurement() function. The function performs multiple round trip pitch catches and calculates an average air flow TOF

difference. The first step in the MakeMeasurement() function is to disable all maskable interrupts not related to TimerD0 and TimerD1, which are used in measuring TOF. This is an important step because the MakeMeasurement() function contains imperative sequences of time critical code. These sequences rely on the CPU to execute deterministically. Servicing an interrupt during these sequences would obviously introduce unwanted lag and uncertainty into the program, producing unpredictable results. A Global Interrupt Disable intrinsic operation would also disable TimerDx interrupts. So, the only way to disable unwanted interrupts it to manually keep track of those being used other parts of the program, then modify the corresponding interrupt enable field in each control register.

As these interrupts are disabled by the InterruptsOff() function at the beginning of PitchCatch(), one must remember to manually turn each one on again after the measurement has been completed. This is accomplished in the InterruptsOn() function.

A diagnostic pulse is issued next on PJ.0 (the same pin used to output SMCLK for clock cycle profiling, if it is not being used for that function). This pulse signifies the start of a measurement routine. It is also intended to serve as an oscilloscope trigger. The diagnostic pulse is performed by successive writes to the output register (high, then low). Each write consumes five clock cycles and the output changes on the last assembly instruction. Therefore, the pin is high for 5 clock cycles, or 200ns.

TD1Config() is a function responsible for setting up TimerD1. Recall that TimerD1 is used to schedule a sequence of events during the pitch catch, while TimerD0 is actually responsible for capturing the input time of the signal. As the program is currently written (version TOF_firmware_4_0), the values contained within TD1Config are not modified, so the function could be moved to the Initialization routine. However, if the phase of the return waveform shifts significantly, then without adaptation the sensor will begin to give erroneous readings since it is designed to detect a particular peak within the acoustic waveform sequence. In this case, the timing values of the TD1 CCR registers (particularly CCR register 0, which sets the time at which TimerD0 starts) must be appropriately modified to compensate for the signal phase shift. This feature is not currently implemented in TOF_firmware_4_0.

The first write in the TD1Config() is to the timer's main control register (TD1CTL0) to select the 25MHz SMCLK as the source for TimerD1. Each of the interrupts for CCR0-2 are then enabled. Finally, we set registers CCR0-2, which control timing. (The "function label" or purpose of the sequence that each CCR triggers is discussed in the PitchCatch() section.) Since TimerD1 is sourced from SMCLK, the value written to each of the CCR registers represents the number of 40ns increments that must elapse before the CCR sets its corresponding flag, CCIFG. The values loaded into TD1CCR0-2, as well as the corresponding decimal and microsecond delay translations are given in Table 5-2. The corresponding functional routine that each CCR triggers is discussed in section 5.6.3.1.

Table 5-2: Typical timing values preloaded into TD1 CCRs

CCR	2	0	1
Functional Label	Prepare Analog Receive	Setup Capture	Tidy Up
Value (hex)	0x00FA	0x0674	0x07B7
Value (decimal)	250 SMCLK cycles	1652 SMCLK cycles	1975 SMCLK cycles
Value (microseconds)	10.00us	66.08us	79.00us

After setting values in TD1Config(), the program enters a *for* loop that will orchestrate sequences of round trip pitch catches. Each of these iterations will complete a measurement of TOF difference from transducer A to B and B to A. The number of times the *for* loop executes determines the number of TOF difference samples that are averaged into one complete measurement and passed on to an external instrument.

The first step in the *for* loop is to declare the roles of the transducers for the first pitch catch operation. This is done in a static void function called SetAtxBrx_m1(), which is an mnemonic for "Set transducer A to transmit (Tx) and transducer B to receive (Rx), a state known as mode 1 (m1)." In this function, the int variables txEN, txClamp, rxClamp, and rxOPA are used as representations. They are assigned to the value of the appropriate bit mapped to the channel of the digital pulser (MAX4811) and analog preamp enable. In the definitions.h header file, these mappings are declared. Refer to the source code in Appendix B for a listing of the definitions.

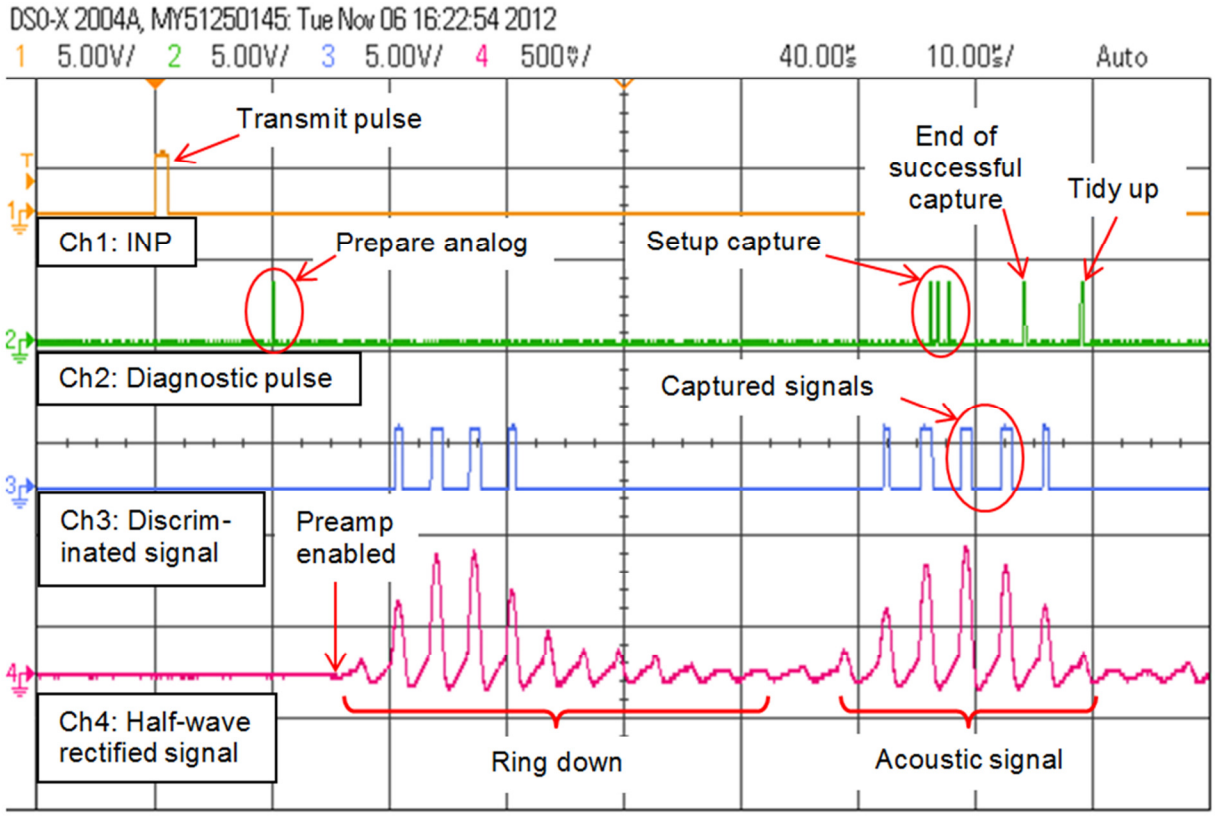


Figure 5-20: Sequences within a single pitch-catch routine

5.6.3.1 PitchCatch()and TimerDx ISRs

The PitchCatch() function and the TimerD1/ TimerD0 ISRs that execute within it are at the core of each TOF measurement. This sequence of five subroutines is responsible for multiple actions which result in a single pitch catch TOF measurement. By the end of execution, which lasts approximately 100us, the MCU has captured two successive waveform peaks within the TD0 CCRs and latches.

5.6.3.1.1 PitchCatch()

PitchCatch() begins the sequence. The first step is a write to Port 3 to enable the appropriate channel of the MAX4811 for transmit (txEN), and to turn on both transmit and receive clamps (txClamp and rxClamp). According to its datasheet, the MAX4811 contains internal logic that disables a channel's clamp while that channel is being pulsed. This allows us to leave txClamp enabled for the entire pitch catch sequence.

The next step in PitchCatch() is to start TimerD1 in continuous mode by modifying the TD1CTL0 register. TDR1, as it increments, will eventually activate each of the CCR interrupts according to the values set in TD1Config(). A brief intrinsic delay of 9 clock cycles following the start of TimerD1 acts as a correction factor that allows us to reference the beginning of each ISR to the start of the first transmit pulse. This improves timing debugging using an oscilloscope that triggers on the INP signal.

The acoustic pitch begins with excitation of the Tx transducer using successive +12V and -12V pulses. INP is first driven high with a write to the P3OUT, an intrinsic delay sets the positive pulse duration, and then the INP bit is cleared. There is another delay before the process is repeated for the negative pulse, by setting then clearing INN. It is important to note that simultaneous high activation of INN and INP causes an illegal input combination explicitly prohibited by the MAX4811 documentation found in [34]. It creates a destructive shoot-through in the pulser push pull MOSFET output stage, which shorts +12V to -12V. The MAX4811 does not contain internal logic to prevent this condition.

After pulsing is complete, there is no need to leave the Tx channel enabled. P3OUT is modified, clearing txEN, but leaving txClamp and rxClamp high. The 27Ω active clamps expedite transducer ring down.

The last instruction in the PitchCatch() subroutine is an intrinsic function to leave the MCU in low power mode 0 (LPM0) with interrupts enabled. Recall that at this point, TimerD0 and TimerD1 are the only unmasked interrupts. In LPM0, the CPU is disabled by gating MCLK. After this instruction, the MCU waits for the ISRs to execute sequentially. As part of the low power philosophy in the MSP430 line, the CPU is designed to automatically wake and service an interrupt then return to LPM0 without the need for any additional C instructions from the developer.

Table 5-3: ISRs that execute after PitchCatch()

Functional Label:	Prepare Analog Receive	Setup Capture	Capture	Tidy Up
Executes?	Always	Always	Only if acoustic signal detected	Always
Timer	D1	D1	D0	D1
Vector	TIMER1_D1	TIMER1_D0	TIMER0_D0	TIMER1_D1
Flag Location	TD1CCTL2-CCIFG	TD1CCTL0-CCIFG	TD0CCTL0-CCIFG	TD1CCTL1-CCIFG
CCR	2	0	0	1
CCR mode	Compare	Compare	Capture	Compare

5.6.3.1.2 Prepare Analog Receive

This is the first subroutine to execute approximately 10us after the start of TimerD1 in the PitchCatch() function. The information pertinent to this subroutine is summarized in Table 5-3. Note that CCR2 and CCR1 share an interrupt vector, so we must check the flags with an *if* statement to determine which routine executes. The first step is to disable the +/-12V switching power supplies to minimize a source of noise. Then, the receive channel is prepared by turning off rxClamp and enabling the appropriate receive preamplifier by setting high the rxOPA value. The final step is to clear the interrupt flag.

5.6.3.1.3 Setup Capture

The timing of the Setup Capture subroutine is important. The responsibility of this ISR is to start the capturing process of the discriminated acoustic waveform at a precise time. That time is chosen such that the same peak within the pulse train is always being detected. In a no airflow condition, the timing must be calibrated such that the rising edge and falling edge captures begin ahead of the strongest peak in the amplified acoustic signal. Furthermore, the enabling of each CCR should be equidistant from the respective rising and falling edge events so that the sensor triggers on the same peak within the acoustic signal, unless the signal shifts more than 180° out of phase.

Figure 5-21 shows how this is accomplished. The oscilloscope waveform is a magnification of the "Setup Capture" area of Figure 5-20. The traces are all the same signals: red (CH4) is the rectified acoustic signal, blue (CH3) the comparator output, green (CH2) the MSP430 diagnostic pulse output, and yellow (CH1) INP. The purpose of the figure is to show the exact time CCR0 and CCR1 are activated, to capture CH3 rising and falling edges.

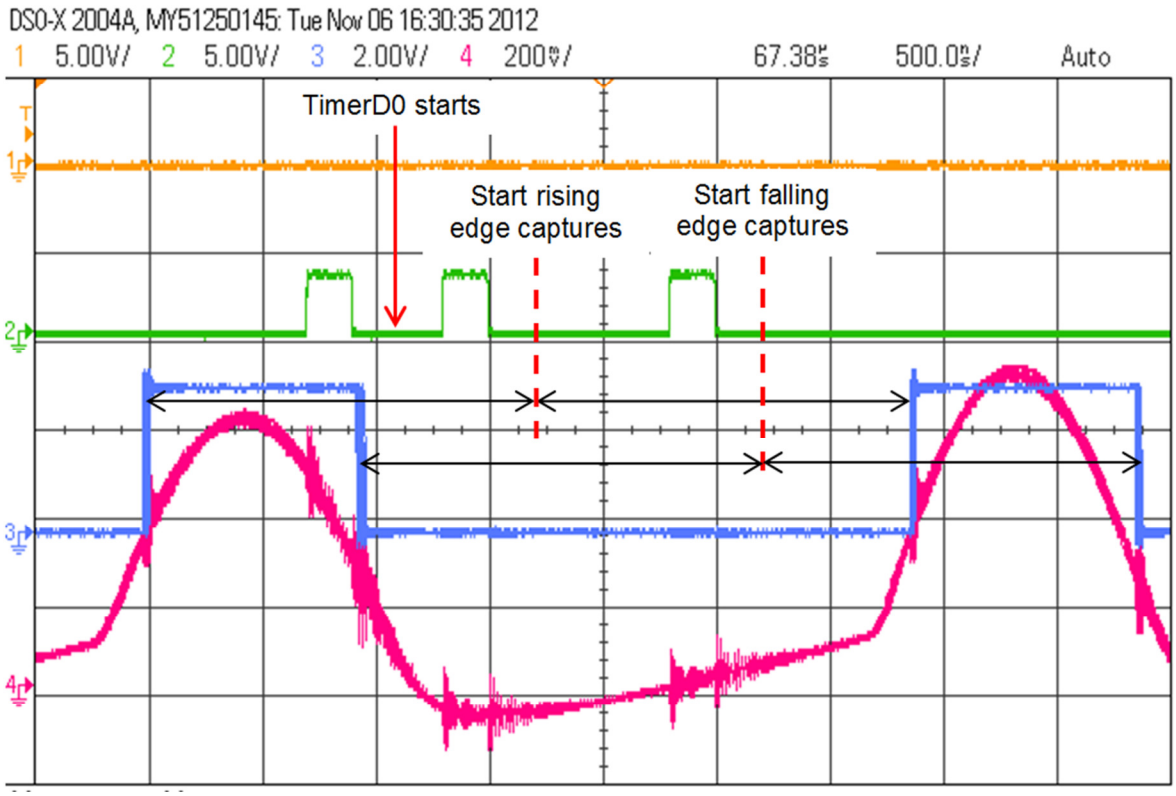


Figure 5-21: Setup capture routine: starting captures 180° out of phase with rising edge (RE) and falling edge (FE) events. TimerD1's CCR0 captures the RE and CCR1 captures the FE.

The TD0CCRInit() function, which executed during initialization, configured the CCR0 and CCR1 registers in TimerD0 such that each register is in dual capture mode. This enables the capture of two successive rising and falling edges, as according to the state diagram shown in Figure 5-22. The first capture for either a rising or falling edge is stored in the TDCCRx register. The second capture displaces this value and shifts the first capture to the associated capture latch, TDCLx. At this time, only the first discriminated peak will be used later in the calculation of TOF difference. A future firmware improvement will permit the sensing detection of a shift of more than 180°.

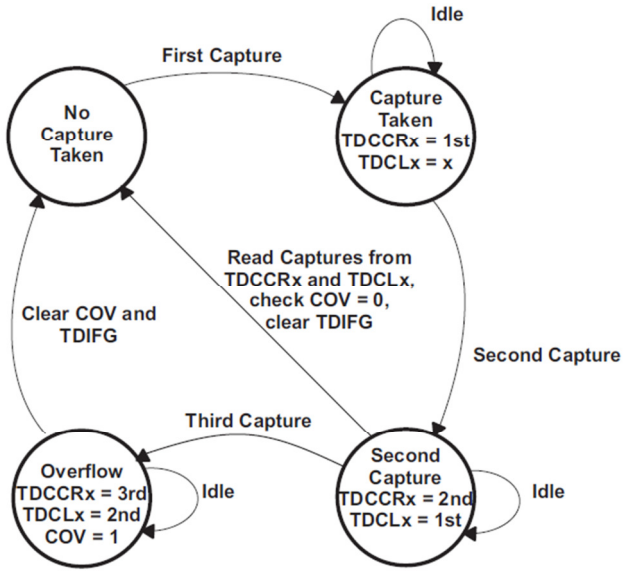


Figure 5-22: Operation of TimerD0 CCR0/1 dual capture mode, showing that two discriminated waveform pulses can be captured by two rising and two falling edges. (image courtesy [41])

5.6.3.1.4 Capture

The Capture code is located in an ISR that executes after the second capture of falling edge by TD0CCR0. The primary purpose of this routine is to stop the capture of falling edges by CCR0 and rising edges by CCR1 before a third capture occurs on either, which would cause an overflow condition. This routine does not read the capture registers or latches, it simply puts them into a safe state so that they can be read by another function later.

The event immediately following the falling edge that triggers this ISR is a rising edge, so as a matter of good form CCR1 is halted first by clearing its capture mode field. (This is just one detail of a design philosophy that would enable us to use higher frequency transducers, for example.) Next, CCR0 is halted, a diagnostic pulse is generated for oscilloscope timing debugging, and the interrupt flags are cleared. Note that the TD0CCTL1 flag must be cleared, even though this interrupt is not enabled, because otherwise the CCR1 overflow bit would be set on the following iteration.

If the acoustic waveform is not present because the acoustic pathway is blocked or the transmitter is disabled, then the Capture subroutine will not execute. This can be observed on an oscilloscope by the absence of the diagnostic pulse that would otherwise execute within the ISR.

5.6.3.1.5 Tidy Up

This is the fourth ISR within the pitch catch measurement sequence. Its purpose is to restore all outputs and timers to their original states prior to the start of PitchCatch(), with the exception of the rising and falling edge values stored within the TimerD0 CCRs and latches. These are extracted in function CheckAndTransferCCRs(). By resetting the appropriate values, Tidy Up supports the overall modularity of firmware components.

Tidy Up shares a common vector with the Prepare Analog Receive subroutine (refer to Table 5-3), so the condition for execution is to first verify the source of the interrupt flag. After verification, the first step is to stop CCR1 and CCR0 captures and to clear their associated flags, in the event that the TimerD0 ISR never executed. Next, both TimerD0 and TimerD1 are stopped and reset. All digital pulser lines and receive preamps are disabled by clearing the entire port 3 output register. The ±12V power supplies are re-enabled and the interrupt flag is cleared. The last step in the routine is to modify the status register with an intrinsic function, which stops the CPU from returning to LPM0 after interrupt exit. Rather than returning to the last line in the PitchCatch() function, the CPU can proceed to the next instruction, CheckAndTransferCCRs().

5.6.4 Calculating acoustic arrival times and TOF difference (Δt)

At the end of the PitchCatch() sequence, discriminated acoustic waveforms are stored in the TimerD0 capture compare registers and latches, in the form of rising and falling edge capture times. The center of the acoustic peak can be determined by averaging the rising and falling edge – a value which becomes the representative TOF for that pitch catch. The TOFs for each pitch catch cycle (from A to B, and B to A) are recorded, and at the end of the measurement cycle, a TOF *difference* (Δt) is computed. This can be averaged over several measurement cycles for noise rejection. Refer to the flowchart in Figure 5-19 for clarification.

5.6.5 Transmitting TOF difference (Δt) to the computer

After a string of measurements has completed, the averaged Δt value is compressed to a single byte. This byte is an unsigned char, where ΔTOF=128 represents no flow, ΔTOF>128 represents positive direction airflow, ΔTOF<128 represent negative airflow. The number (offset from 128) represents how many 40ns clock cycles are in the Δt measurement. (For example, 130 would represent a Δt of 80ns, with velocity in the positive airflow direction.) Using the MCU

communications UART, the TOF measurement is passed over serial/USB to a connected computer.

A LabVIEW program (see Appendix D for code) running on the PC accepts the byte from a COM port. It decompresses the data to recover Δt, and computes Eq. 3-10 to determine flow velocity. The velocity values are graphed. Volumetric flow rate is calculated by multiplying velocity by the cross section of the flow chamber, and also graphed. An integration scheme, similar to that used in the stage 2 prototype, can be used to determine volume of breaths.

5.7 Concluding remarks on the final prototype design

The final prototype integrates hardware and software to achieve a superior ultrasonic TOF flow measurement. A transmit stage uses a digital pulser to excite the transducers, while a receive stage switches, amplifies, filters, and conditions acoustic signals before routing them to the MCU hardware timer. In software, the MCU uses sequential pitch catch scheme to mitigate problems with ring down and allow close transducer spacing. Although the electronic design is more complex than previous prototypes, the final airflow chamber is compact.

Chapter 6. Sensor Validation & Discussion

The results presented for the final prototype demonstrate functionality of various stages of the electronics, a look at pitch-catch waveforms, and finally real time graphs of flow measured by the sensor. Oscilloscope waveforms are first shown which characterize the major analog components of the circuit, such as ring down, crosstalk, amplification, and filtering. Next, the complete measurement cycle of a pitch-catch is characterized. Measurement readings from manual and ventilator tests are also presented, which show that the prototype is capable of sensing inspired volume as low as 100mL.

6.1 Transducer ring down received acoustic signal

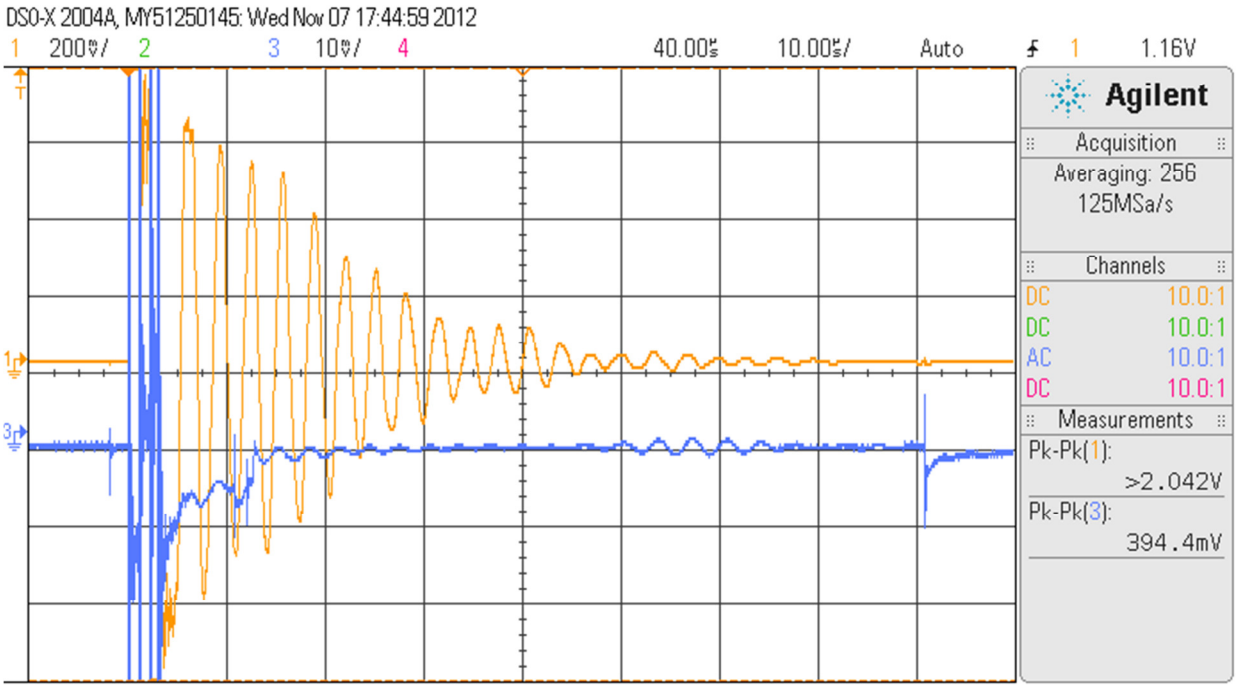


Figure 6-1: Transmit transducer ring down (yellow- CH1) and received acoustic signal

Figure 6-1shows one transducer (yellow trace – CH1) transmitting and ringing down, while the other transducer (blue trace – CH3) waits for and receives a 2mV p-p acoustic signal approximately 45µs after initial excitation. Further observation of the traces reveals a 60-70dB coupling / crosstalk from the transmit to the receive channel. Note that ring down coupling on the received channel just barely subsides before the arrival of the acoustic signal.

6.2 Analog preamp and BPF performance

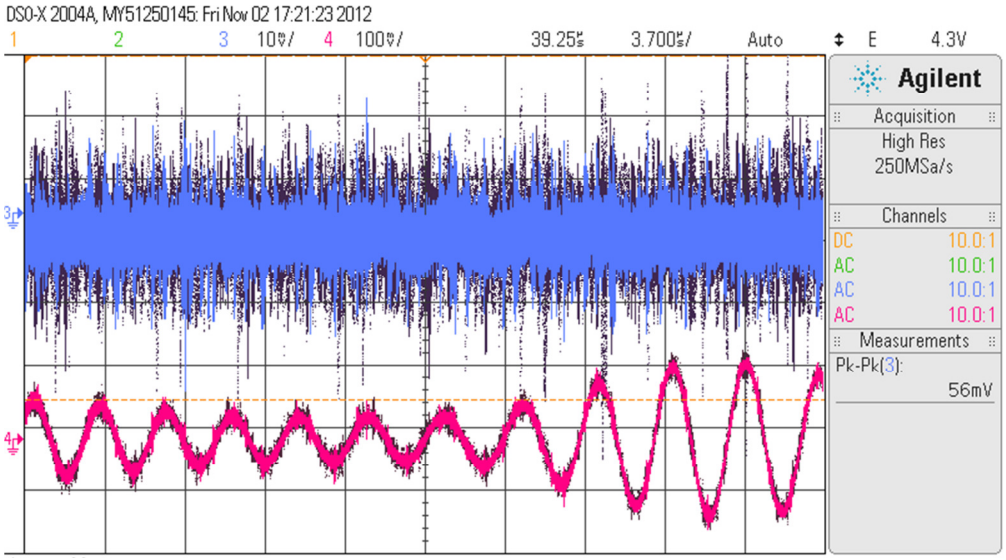


Figure 6-2: Raw received signal (blue-CH3) and post BPF filtered signal (red–CH4)

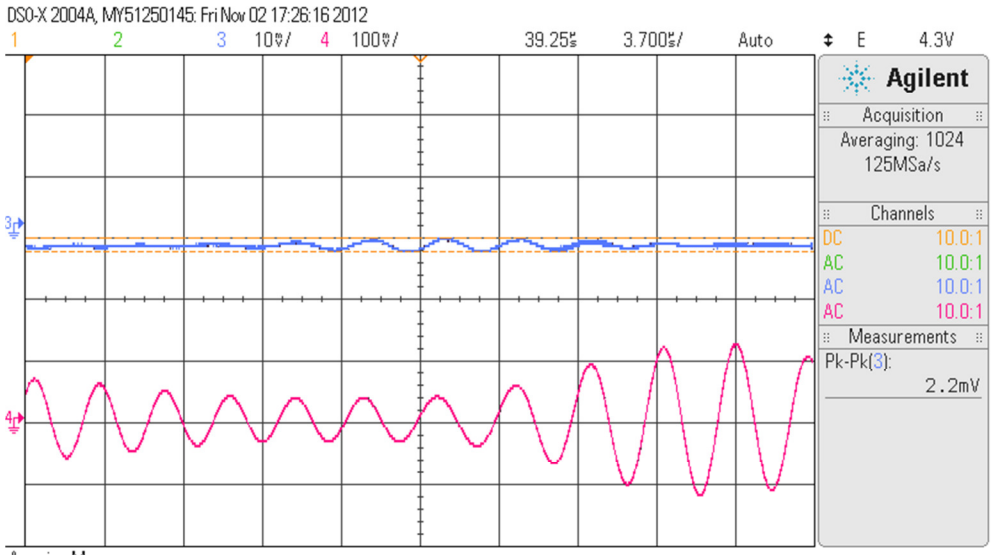


Figure 6-3: Averaged receive signal for comparison

The above two figures show how the input acoustic signal is conditioned by the preamplifier and band pass filter. Figure 6-2 was generated using the oscilloscope's high resolution capture mode and no averaging. The input signal is hidden within noise, but then extracted and amplified by the BPF. The bottom scope waveform, Figure 6-3, is a 1024 sample average for comparison, which shows both the input and BPF in a much cleaner form, because random noise is averaged out.

6.3 Pitch catch analog waveforms

Figure 6-4 focuses on the analog signal conditioning stages of a single pitch catch. Similar to Figure 6-1, one can see ring down from the original transducer excitation on CH1 and a 70dB coupling to the receive channel, which dies out before the contra-propagating acoustic signal arrives approximately 50us after the excitation pulse. The analog observations on CH2, 3, and 4 are consistent with parts of the analog stages explained graphically the Chapter 5 design discussion (refer to Figure 5-5 and Figure 5-14).

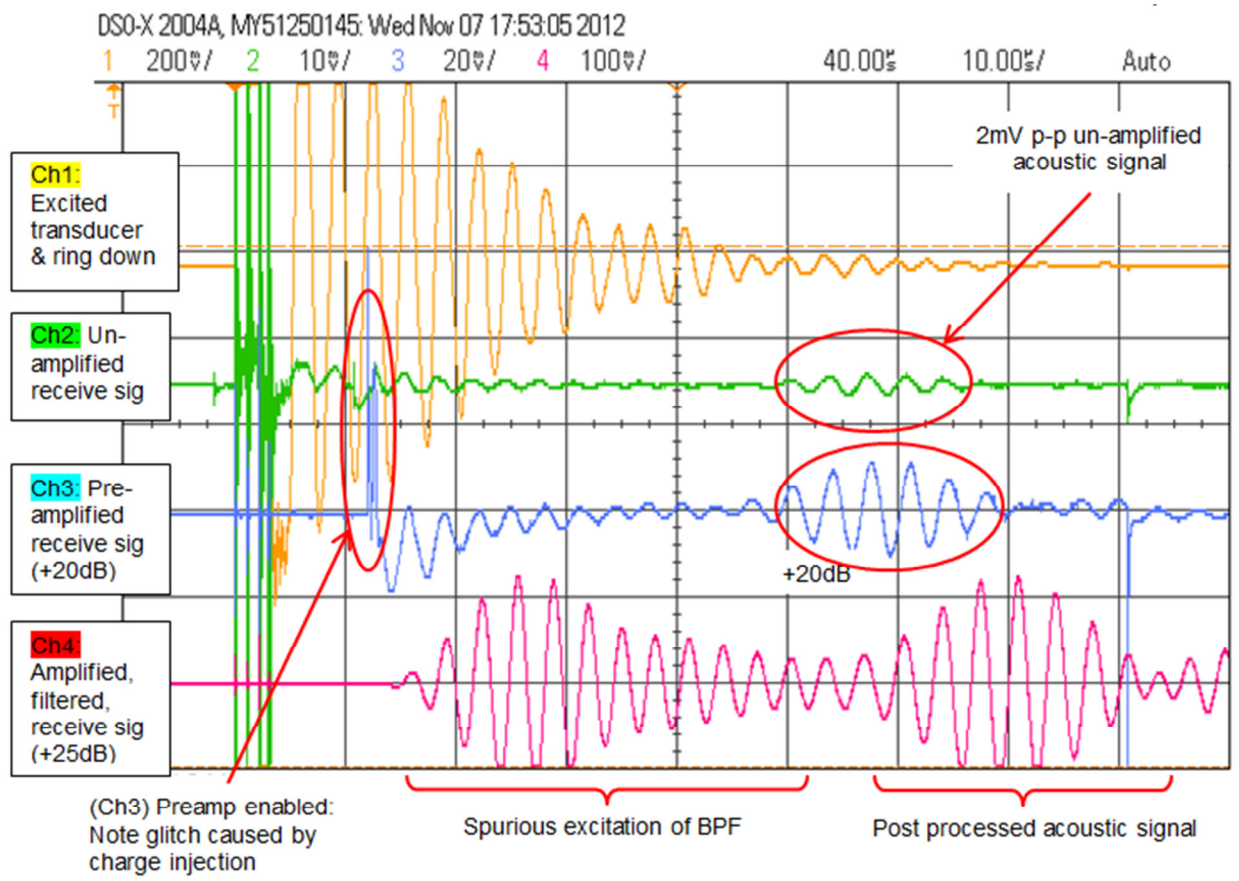


Figure 6-4: Select traces from analog signal chain. Of importance is the charge injection glitch resulting from the enabling of the LMV791 preamp on Ch3, which acts as an impulse to the BPF.

Other items of interest: The preamp, when enabled, generates a glitch (charge injection) which excites the BPF along with the residual ring down. One can also see the group delay in the post-BPF processed signal, a characteristic of the BPF. High Q in the BPF is problematic, as it begins to smear into the acoustic signal. Further investigation of this phenomenon is necessary, and the BPF may need to be redesigned in future hardware upgrades.

6.4 Pitch catch digital waveforms

The MCU initiates and controls the pitch catch sequence. This begins with a digital transmit pulse to one of the transducers, shown in yellow (CH1) in Figure 6-5. Other pulses, called diagnostic pulses, are programmed into software to highlight the execution of PitchCatch() subroutines explained in Chapter 5. These pulses are the green trace (CH2) in Figure 6-5. The red trace (CH4) is the rectified acoustic signal from the receive channel. The blue trace (CH3) shows the discrimination of this waveform using a comparator (the last stage in Figure 5-14).

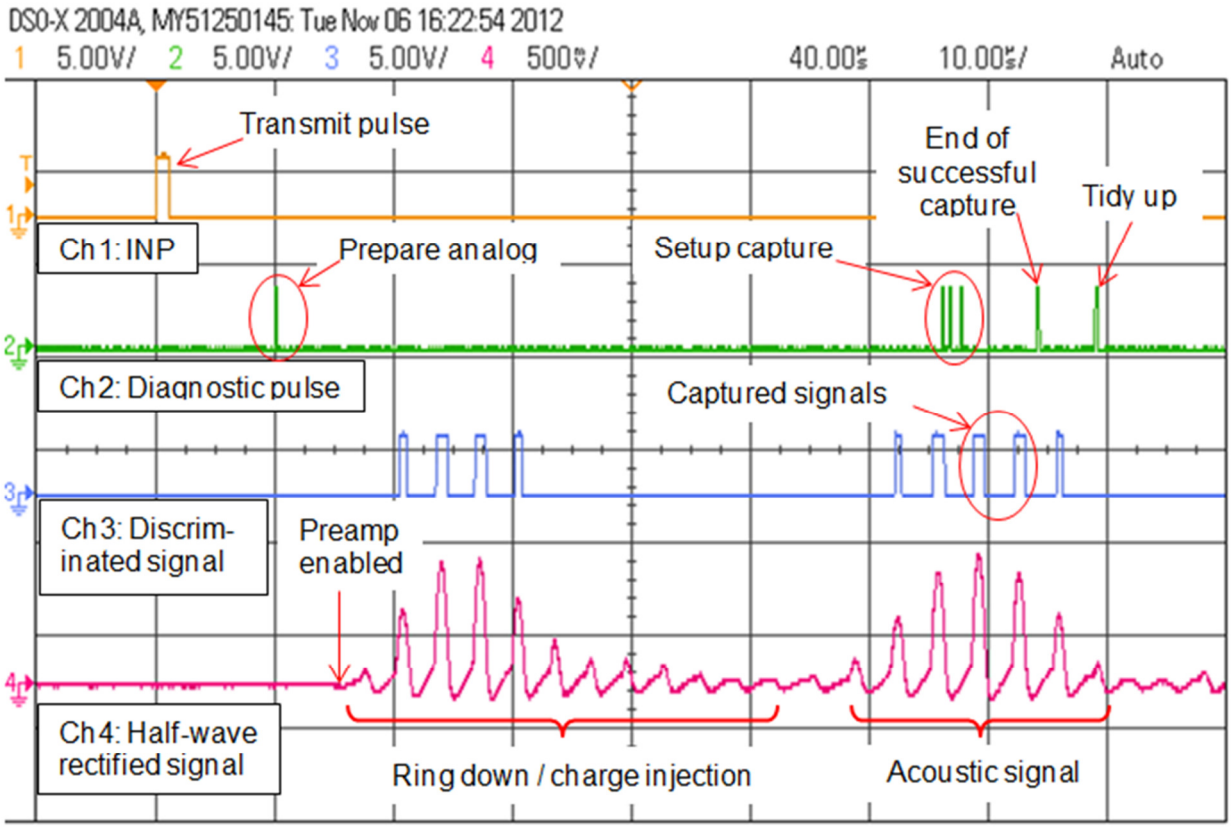


Figure 6-5: MCU diagnostic pulses and analog signals

A close-up, time-magnified view of the "setup capture" and "end of successful capture" routines can be seen in Figure 6-6. This is a persistence, or "motion blur", oscilloscope capture. The acoustic signal and its discriminated waveform do indeed time shift during airflow – but not by more than a single 300 kHz wave cycle. The MCU capture routine, shown by the 4th green CH2 pulse, locks onto the rising and falling edges of the discriminated waveform. Thus, the MCU software is confirming that its hardware received an acoustic pulse.

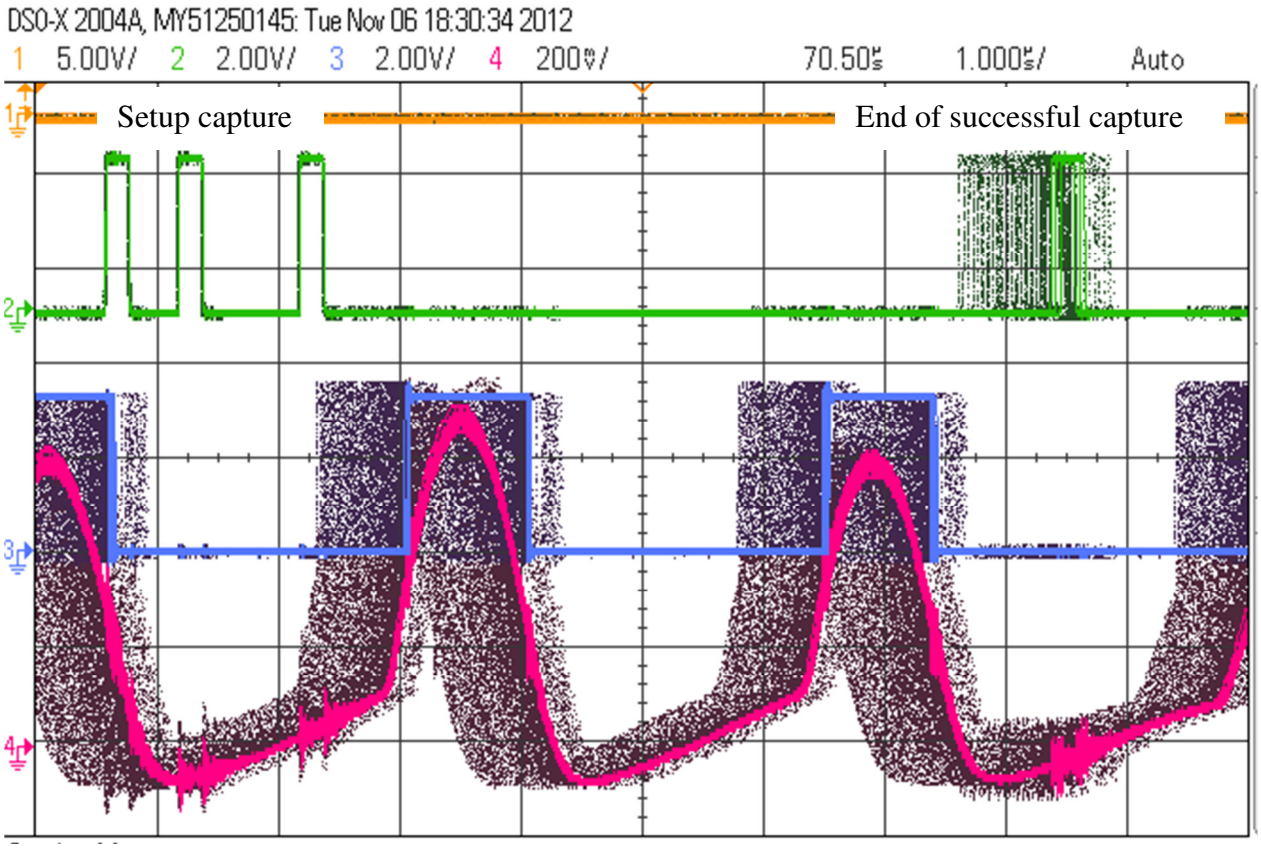


Figure 6-6: Persistence view confirming dynamic capture of acoustic waveform by MCU hardware timer

6.5 LabVIEW graphs of airflow data

The device has been verified both by manually breathing through the tube and by connecting it to a ventilator designed to administer repeated breaths of known air volume. Figure 6-7 shows a qualitative verification of airflow and sensor operation, moving and blowing through the sensor and observing airflow in real time on a LabVIEW graph.

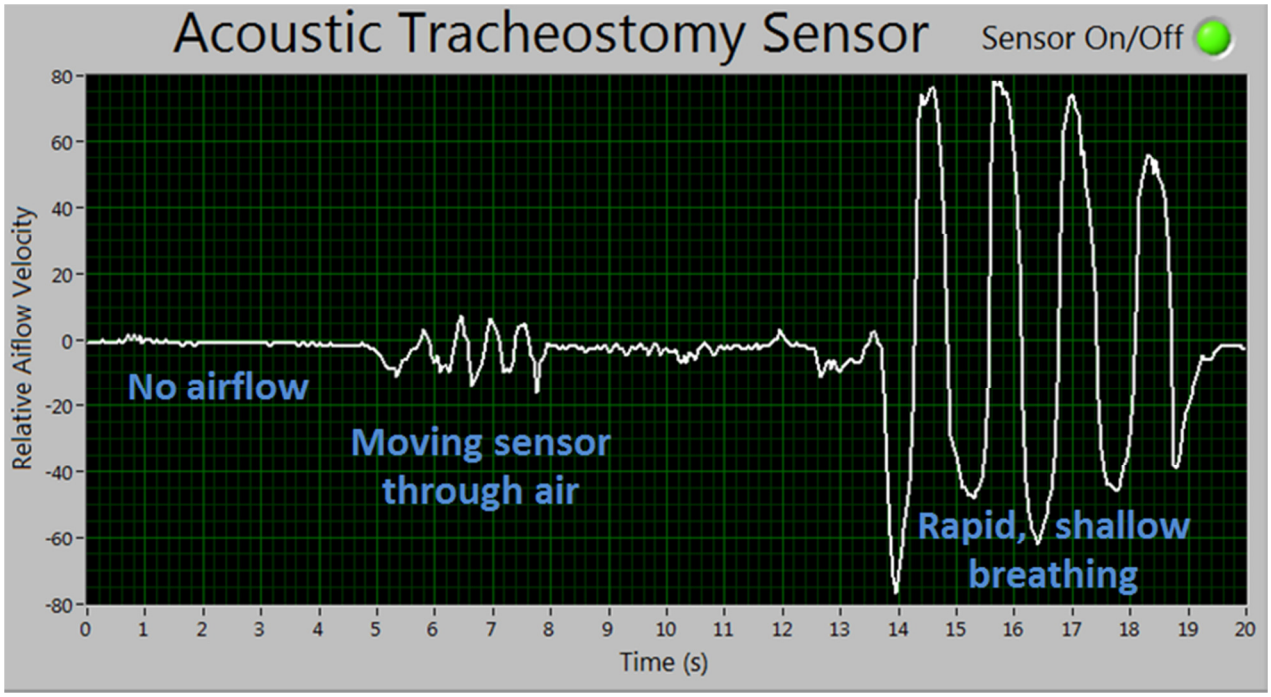


Figure 6-7: Qualitative functionality test of flow sensor by graphing flow output in LabVIEW

In Figure 6-8 and Figure 6-9, repeated one-way breaths were delivered by a ventilator through the flow chamber. Ventilator breath detection for several tidal flow settings, ranging from 100mL – 500mL, was performed. The graphs show a sequence of periodic tidal flow, which can be threshold detected in software. The device had success in detecting 100mL (the lowest setting for the ventilator), 200mL, 300mL, and 500mL breaths with the same calibration settings.

Further work and more detailed characterization of the device will be needed. However, these initial results, both qualitative and quantitative, demonstrate functionality of the airflow sensor.

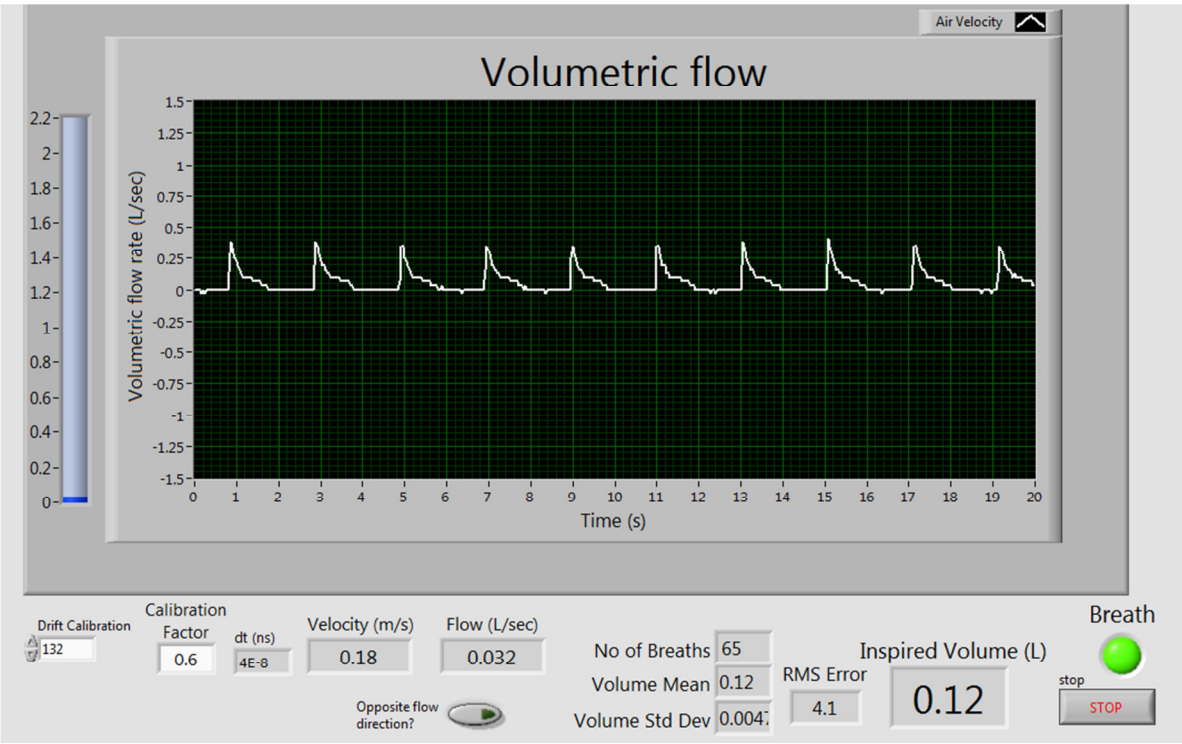


Figure 6-8: 100mL breath detection

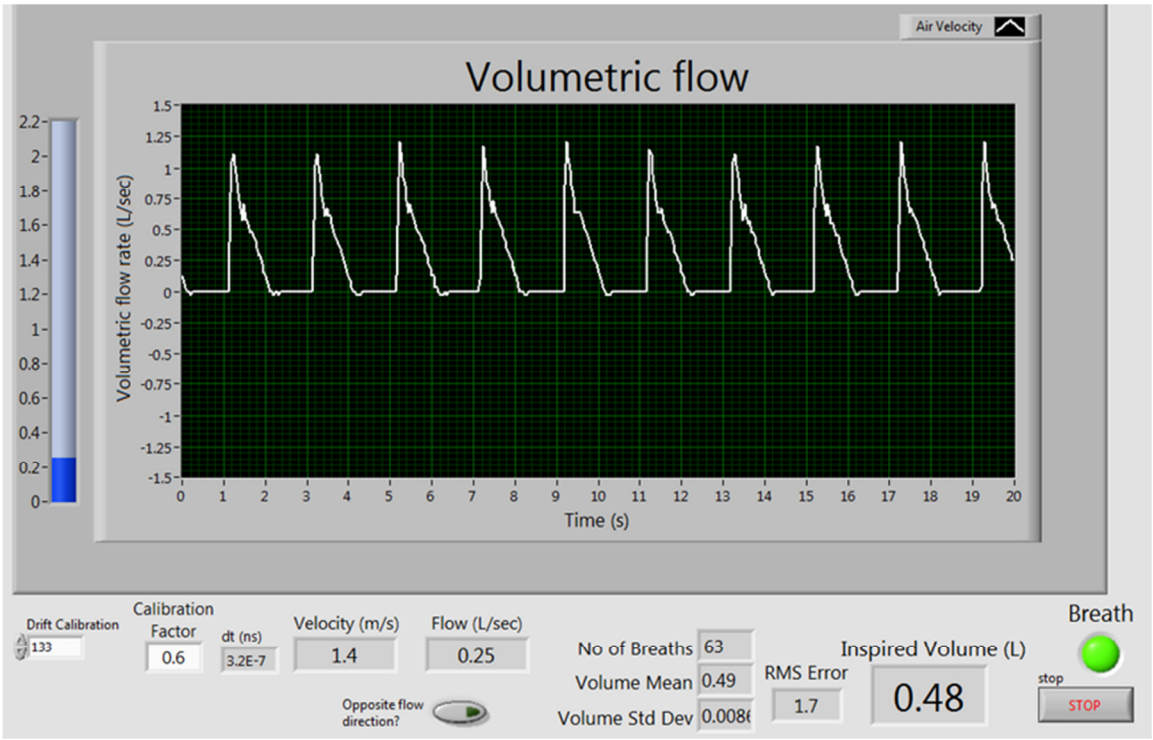


Figure 6-9: 500mL breath detection

6.6 Acoustic echo interference – an interesting issue

Recently, problems with erroneous, slowly drifting measurements were observed in the device. This was a troublesome bug that emerged in Nov 2012. The flow output, as seen in the LabVIEW program, would appear to drift, even in a no-flow condition. This had not been observed previously.

The root cause was determined and corrected after several weeks. Behavior was consistent with a timing error between PitchCatch() routines that causes a distant, weak echo (in this case, the seventh echo) to interfere with an acoustic signal. Multiple echoes from a single acoustic signal can be seen in Figure 6-10, bouncing back and forth between the two transducers. The problem was when one of these echoes aligned with a measurement.

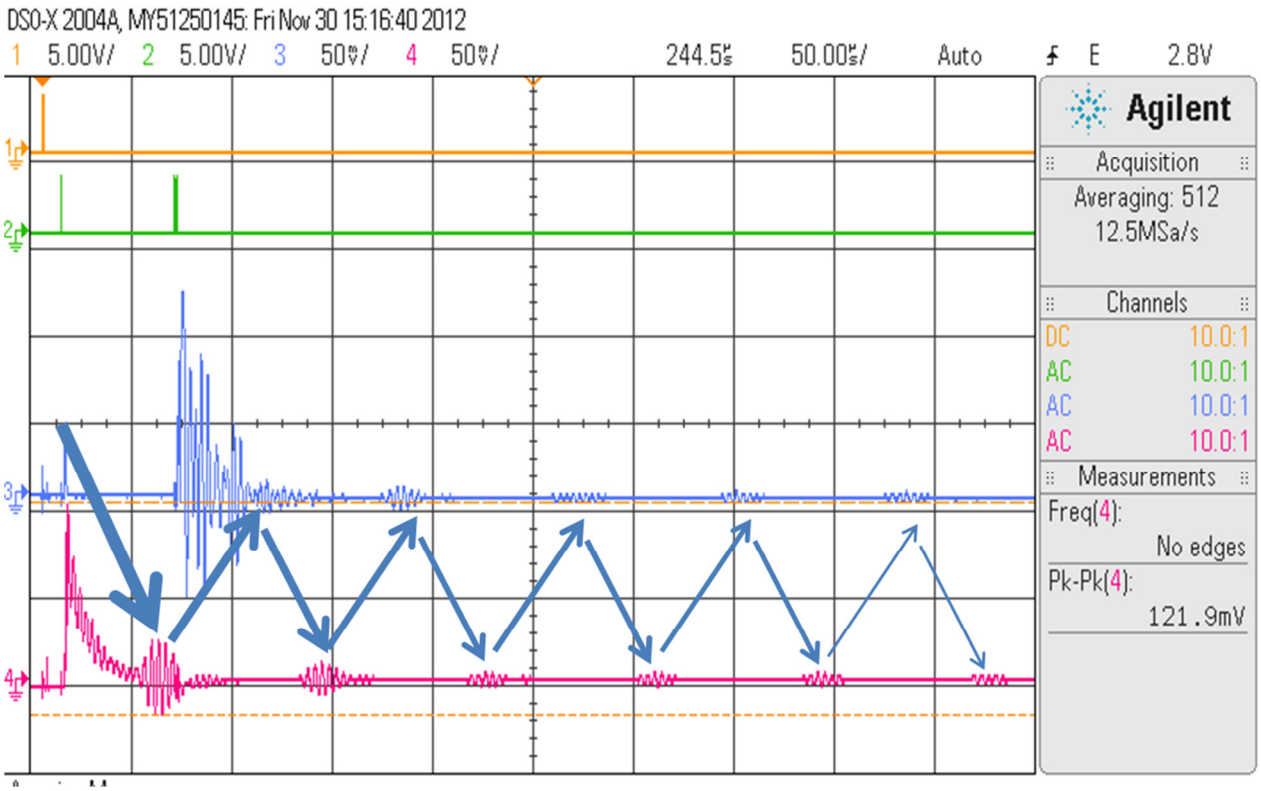


Figure 6-10: A single acoustic signal echoing between transducer A (CH3) and B (CH4)

The bug was originally introduced through a timing change to PitchCatch(). When timing was re-modified through software so that echoes do not coincide with signals of interest, the problem was no longer observed. Meaningful and consistent flow results, such as the flow graphs shown in Figure 6-7 through Figure 6-9, were once again observed.

Chapter 7. Conclusions and Future Work

7.1 Clinical recap

Pediatric tracheostomy patients constitute a subgroup of children who are at-risk for airway complications. Apnea monitors are critical for early detection when children are unattended at home (especially during the night). The ideal monitor should consistently and reliably alert caregivers of a suspension of breathing. Currently, however, physicians have no choice but to prescribe commercial monitors, which indirectly monitor respiration through chest impedance and pulse oximetry. The false positive rate on these monitors is greater than 85%, and they are particularly ineffectual in the detection of artificial airway obstructions, the most common issue with tracheostomy children.

7.2 Accomplishments of this work

The work presented in this thesis details the design and verification of a functional ultrasonic flow sensor. The sensor uses closely spaced air-coupled transducers to make an ultrasonic TOF flow measurement using a sequential pitch-catch scheme.

Primary components of the circuit include transmit, receive, and control stages. The transmit stage generates electrical pulses necessary to initiate an acoustic signal through a digital pulser. The pulser can independently enable, pulse, and clamp each transducer. The receive stage includes low noise preamplifiers, a band pass filter, variable gain amplifier, and discriminator. The analog stage translates a weak acoustic signal into a digital pulse train which is delivered to an MSP430 microcontroller, which measures arrival time.

The device has been validated on a ventilator. Observations show that it is capable of measuring tidal flow down to 100mL and can generate a display of subsequent breaths. However, the accuracy and linearity of measurements has not yet been tested.

7.3 Future Research

Of immediate interest is testing the device for accuracy and linearity across a range of flow rates. To accomplish this, the sensor needs to be placed in line with a spirometer and/or mass air flow meter and connected to a calibrated airflow source.

There is always ample opportunity for device improvement, to the flow chamber and the electronics hardware. A few ideas to improve the device are:

- CAD modeling and rapid prototyping of the sensor flow channel
- Design of custom, smaller transducers
- Better understanding of transducer equivalent circuit models, particularly the models discussed in [42] and [43].
- Using transducer equivalent circuit models to develop a matching electrical network
- Automatic gain control (AGC) to compensate for attenuated acoustic signals
- Possible redesign of the band pass filter stage
- Noise analysis on received signal – both amplitude and phase
- Incorporation of the ultrasonic time-of-flight algorithm into MCU firmware

7.4 Final remarks

In this research, a novel airflow sensor based on ultrasonic time of flight was developed to address the clinical need for improved respiratory monitoring of young children. After two initial prototyping stages, a final design emerged that enabled a compact flow sensor driven by a custom embedded electronics platform. Although future work will be needed to fully characterize the device, it has already demonstrated functional operation and shows promise in upcoming applications.

References

[1] J. Sherman, S. Davis, S. Albamonte-Petrick, R. Chatburn, C. Fitton, C. Green, *et al.*, "Care of the child with a chronic tracheostomy. This official statement of the American Thoracic Society was adopted by the ATS Board of Directors, July 1999," *American journal of respiratory and critical care medicine,* vol. 161, p. 297, 2000.

[2] M. Andre Muelenaer, "Personal Communication," ed, 2012.

[3] R. Jeanette Zaichkin, MN, NNP-BC. (2009, November 2012). *Newborn Intensive Care: What Every Parent Needs to Know (American Academy of Pediatrics) (3rd ed.).* Available: http://www.healthychildren.org/English/ages-stages/baby/preemie/pages/Apnea-Monitors.aspx

[4] N. Nassi, R. Piumelli, E. Lombardi, L. Landini, G. Donzelli, and M. de Martino, "Comparison between pulse oximetry and transthoracic impedance alarm traces during home monitoring," *Archives of disease in childhood,* vol. 93, pp. 126-132, 2008.

[5] M. Marzocchi, R. T. Brouillette, D. E. Weese-Mayer, A. S. Morrow, and L. P. Conway, "Comparison of transthoracic impedance/heart rate monitoring and pulse oximetry for patients using diaphragm pacemakers," *Pediatric pulmonology,* vol. 8, pp. 29-32, 1990.

[6] J. T. John L Robertson, Andre Muelenaer, "Research Proposal: An Acoustic Sensor for Airflow in Pediatric Artifical Airways (unpublished)," 2009.

[7] C. R. Basil Matta, *Tracheostomy: A Multi-Professional Handbook*. London: Greenwich Medical Media Limited, 2003.

[8] D. Trachsel and J. Hammer, "Indications for tracheostomy in children," *Paediatric Respiratory Reviews,* vol. 7, pp. 162-168, 2006.

[9] W. Tantinikorn, C. M. Alper, C. D. Bluestone, and M. L. Casselbrant, "Outcome in pediatric tracheotomy," *American journal of otolaryngology,* vol. 24, p. 131, 2003.

[10] D. R. Hess, "Tracheostomy tubes and related appliances," *Respiratory care,* vol. 50, pp. 497-510, 2005.

[11] L. C. e. al., "Tracheotomy in Pediatric Patients: A National Perspective," *Archives of Otolaryngology, Head & Neck Surgery* pp. 523-529, 2003.

[12] (2012, December). *Apnea Monitor*. Available: http://www.walgreens.com/marketing/library/contents.jsp?docid=19846&doctype=2

[13] M. Patricia Cole Stivrins, "Home Monitoring Manual: A Guide for Monitoring Infant Apnea," BryanLGH Medical Infant Apnea Center.

[14] A. H. Medical, "Apnea Monitors: Product Information for Customers (PIC-AP)," T. M. Group, Ed., ed, 2009.

[15] J. G. Webster, *Design of Pulse Oximeters*. New York, 1997.

[16] *Sensor Technology Handbook*. Boston: Elsevier, 2005.

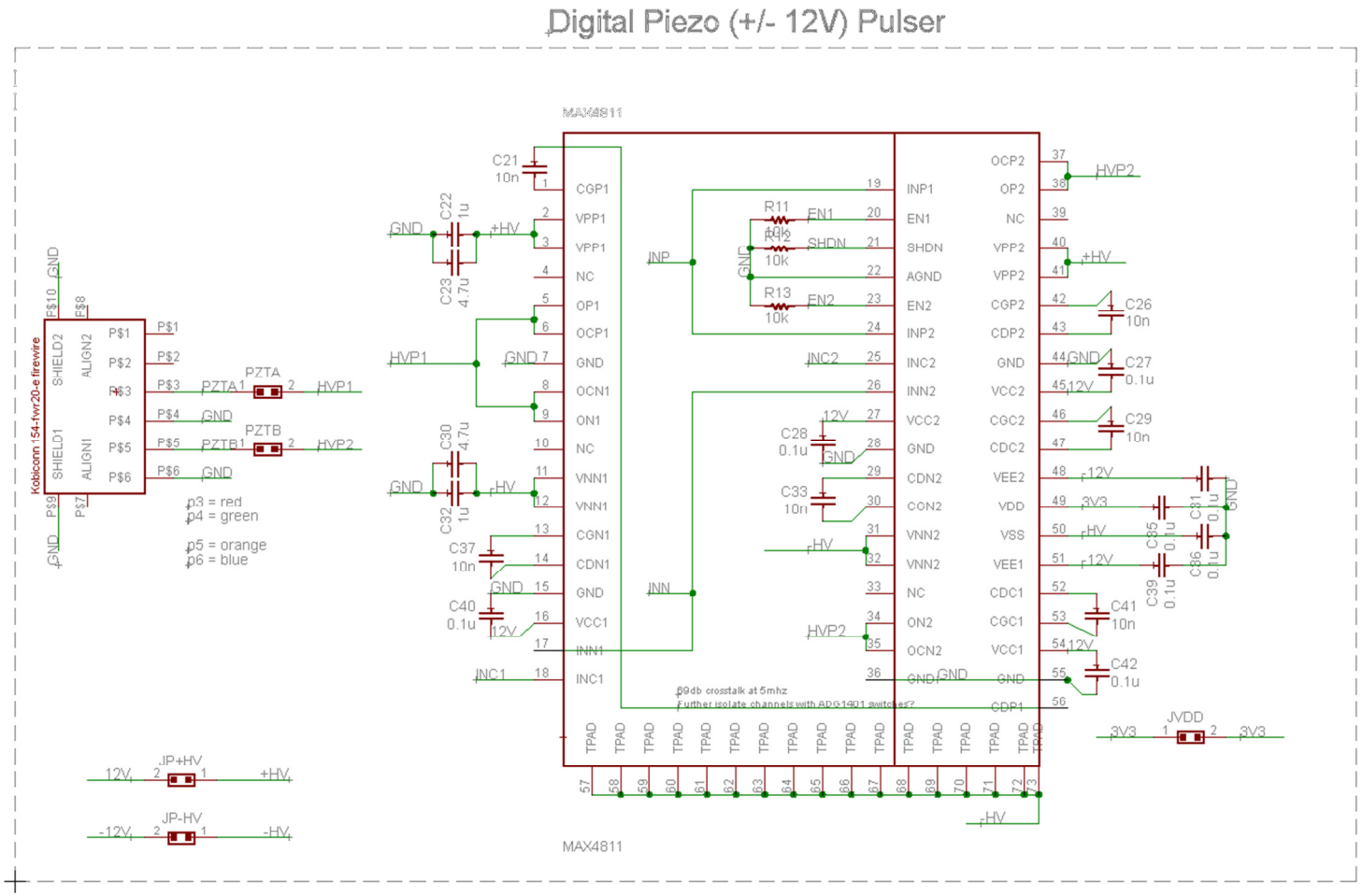
[17] R. G. Meny, J. L. Carroll, M. T. Carbone, and D. H. Kelly, "Cardiorespiratory recordings from infants dying suddenly and unexpectedly at home," *Pediatrics,* vol. 93, pp. 44-49, 1994.

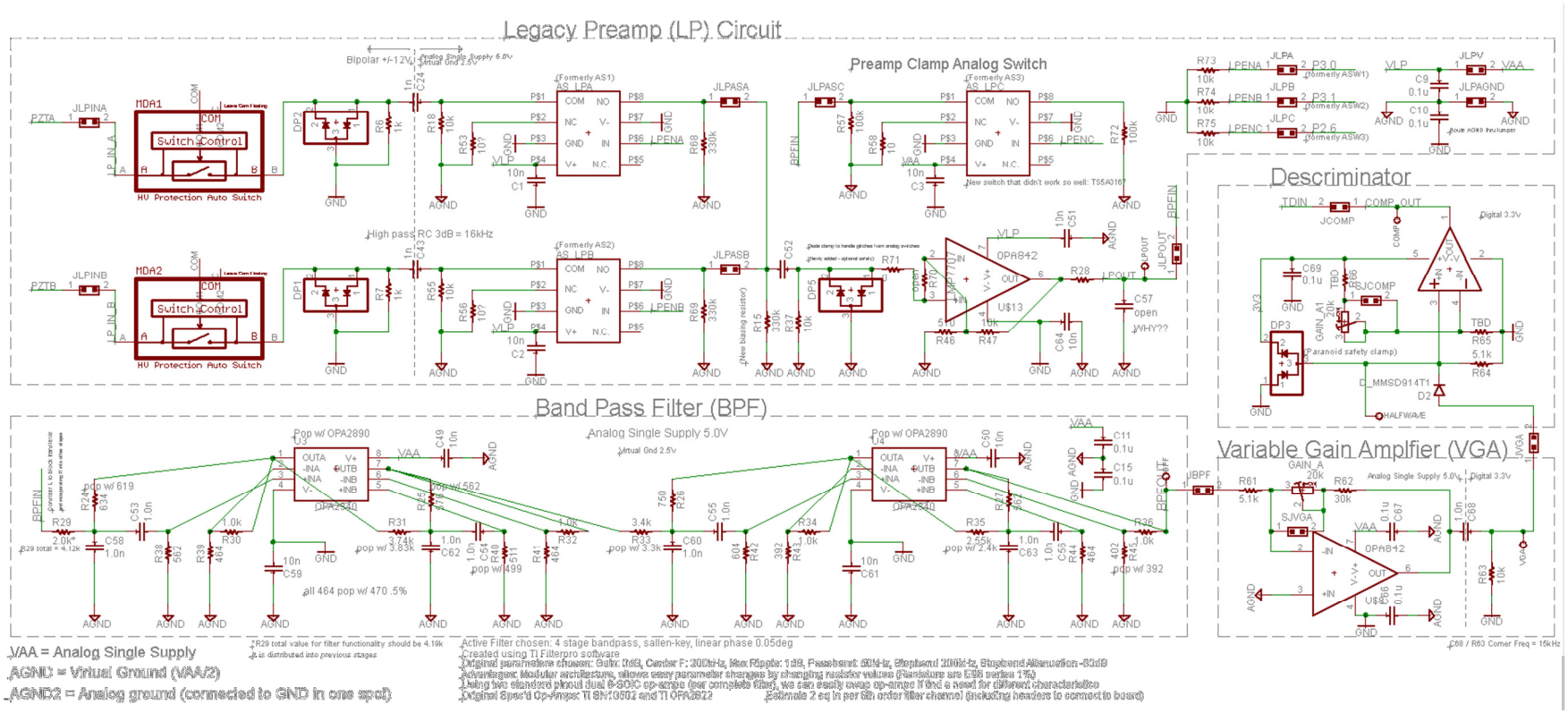
[18] B. Bohnhorst, C. S. Peter, and C. F. Poets, "Pulse oximeters' reliability in detecting hypoxemia and bradycardia: comparison between a conventional and two new generation oximeters," *Critical care medicine,* vol. 28, pp. 1565-1568, 2000.

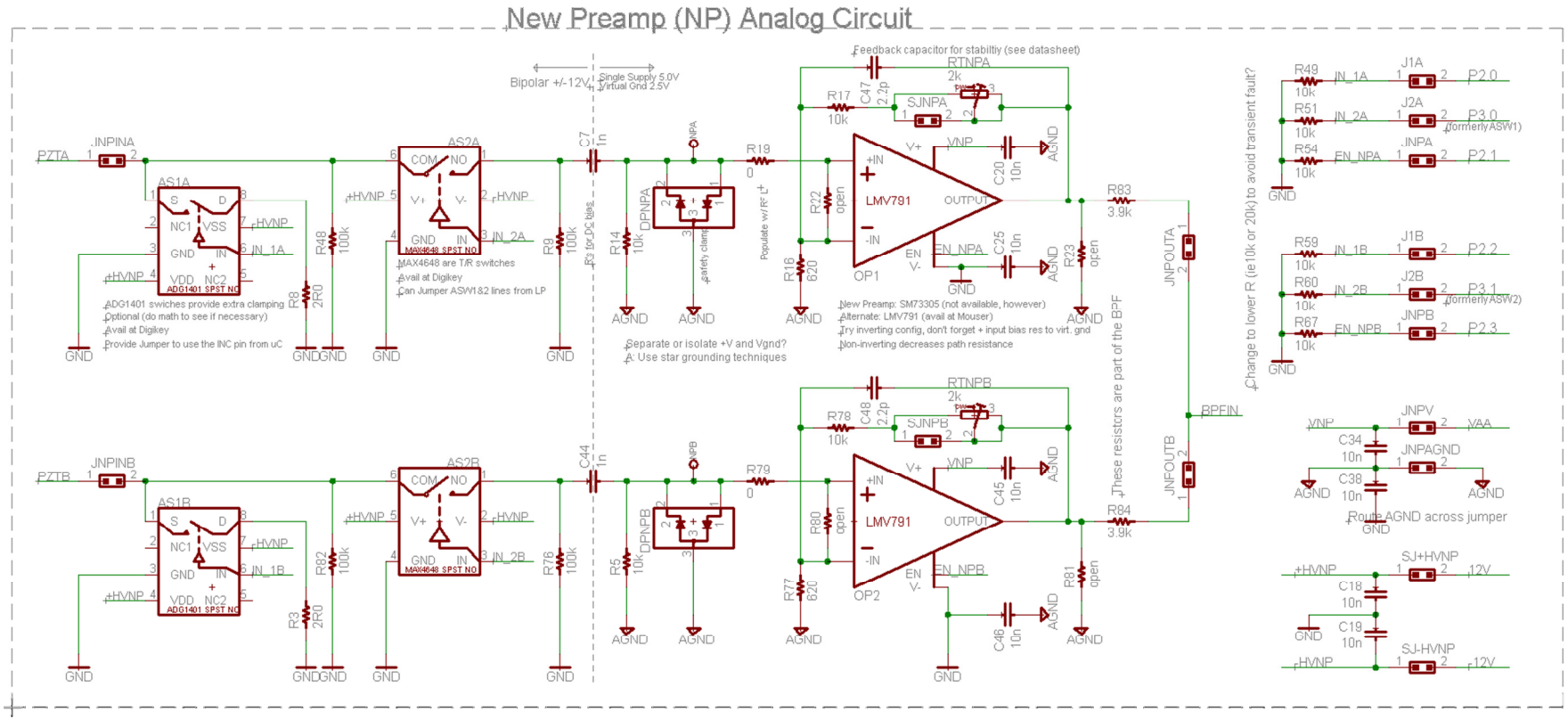
[19] I. Endress-Hauser. (2009, December). *The Ultrasonic Flow Measuring Principle (video explanation).* Available: http://www.youtube.com/watch?v=Bx2RnrfLkQg

[20] D. A. Bohn, "Environmental effects on the speed of sound," *Journal of the Audio Engineering,* 2012.
[21] R. C. Baker, *Flow Measurement Handbook: Industrial Designs, Operating Principles, Performance, and Applications*: Cambridge University Press, 2005.
[22] C. L. Smith, *Basic Process Measurements*: John Wiley & Sons, 2011.
[23] (October). *Doppler Ultrasonic Flowmeters*. Available: http://www.efunda.com/designstandards/sensors/flowmeters/flowmeter_usd.cfm
[24] D. W. Baker, "Pulsed ultrasonic Doppler blood-flow sensing," *Sonics and Ultrasonics, IEEE Transactions on,* vol. 17, pp. 170-184, 1970.
[25] C. f. N. Testing. Available: http://www.ndt-ed.org/EducationResources/CommunityCollege/Ultrasonics/Physics/modepropagation.htm
[26] D. T. Blackstock, *Fundamentals of Physical Acoustics*. New York: John Wiley & Sons, 2000.
[27] T. Rossing, *Springer Handbook of Acoustics*. New York: Springer Science and Business Media, 2007.
[28] "Piezoelectric Ceramics Brochure," in *Online*, M. ElectroCeramics, Ed., ed, 2007, p. 21.
[29] L. C. Lynnworth, "Ultrasonic impedance matching from solids to gases," *Sonics and Ultrasonics, IEEE Transactions on,* vol. 12, pp. 37-48, 1965.
[30] I. BioSono. (2007-2012, November). *Ultrasound Physics*. Available: http://biosono.com/UltrPhys/UltrPhysMain.htm
[31] A. T. Corporation, "AT300 300 kHz Ultrasonic Transducer Datasheet," 2010.
[32] C. Buess, P. Pietsch, W. Guggenbuhl, and E. A. Koller, "Design and Construction of a Pulsed Ultrasonic Air Flowmeter," *IEEE transactions on Biomedical Engineering,* vol. BME-33, August 1986.
[33] V. Magori, "Ultrasonic Sensors in Air," *IEEE Ultrasonics Symposium,* vol. 1, p. 471, 1994.
[34] M. I. Products, "MAX4811: Dual, Unipolar/Bipolar High-Voltage Digital Pulser (Datasheet)," 2008.
[35] I. Supertex, "MD0100: High Voltage Protection Transmit/Receive Switch (Datasheet)," 2010.
[36] N. Semiconductor, "LMV791: 17 MHz, Low Noise, CMOS Input, Operational Amplifier with Shutdown (Datasheet)," 2008.
[37] "Active Filter Design Techniques," in *Op Amps for Everyone*, ed Dallas, TX, 2008.
[38] T. Instruments, "OPA2890: Dual Low-Power, Wideband, Voltage Feedback Operational Amplifier (Datasheet)," 2009.
[39] T. Instruments, "OPA890: Low-Power, Wideband, Voltage Feedback Operational Amplifier (Datasheet)," 2009.
[40] T. Instruments, "TLV3201: 40-ns, Micropower, Push-pull Output Comparators (Datasheet)," 2012.
[41] T. Instruments, "MSP430f51x2: Mixed Signal Microcontroller (Datasheet, technical document SLAS619D)," 2010-2012.
[42] W. P. Mason, *Electromechanical transducers and wave filters*: Van Nostrand Reinhold, 1946.
[43] R. Krimholtz, D. Leedom, and G. Matthaei, "New equivalent circuits for elementary piezoelectric transducers," *Electronics Letters,* vol. 6, pp. 398-399, 1970.

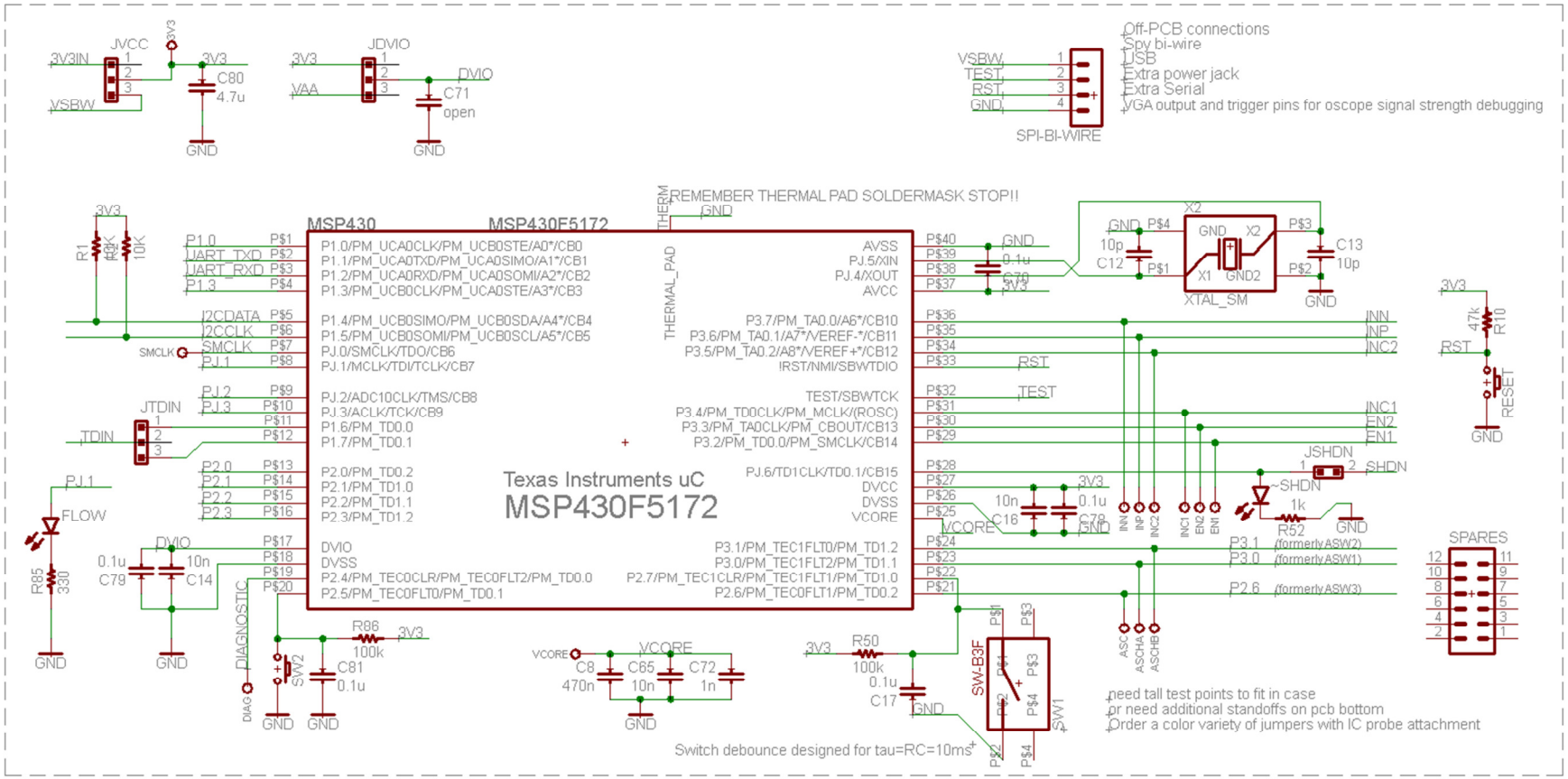
Appendix A: Final Prototype Schematics



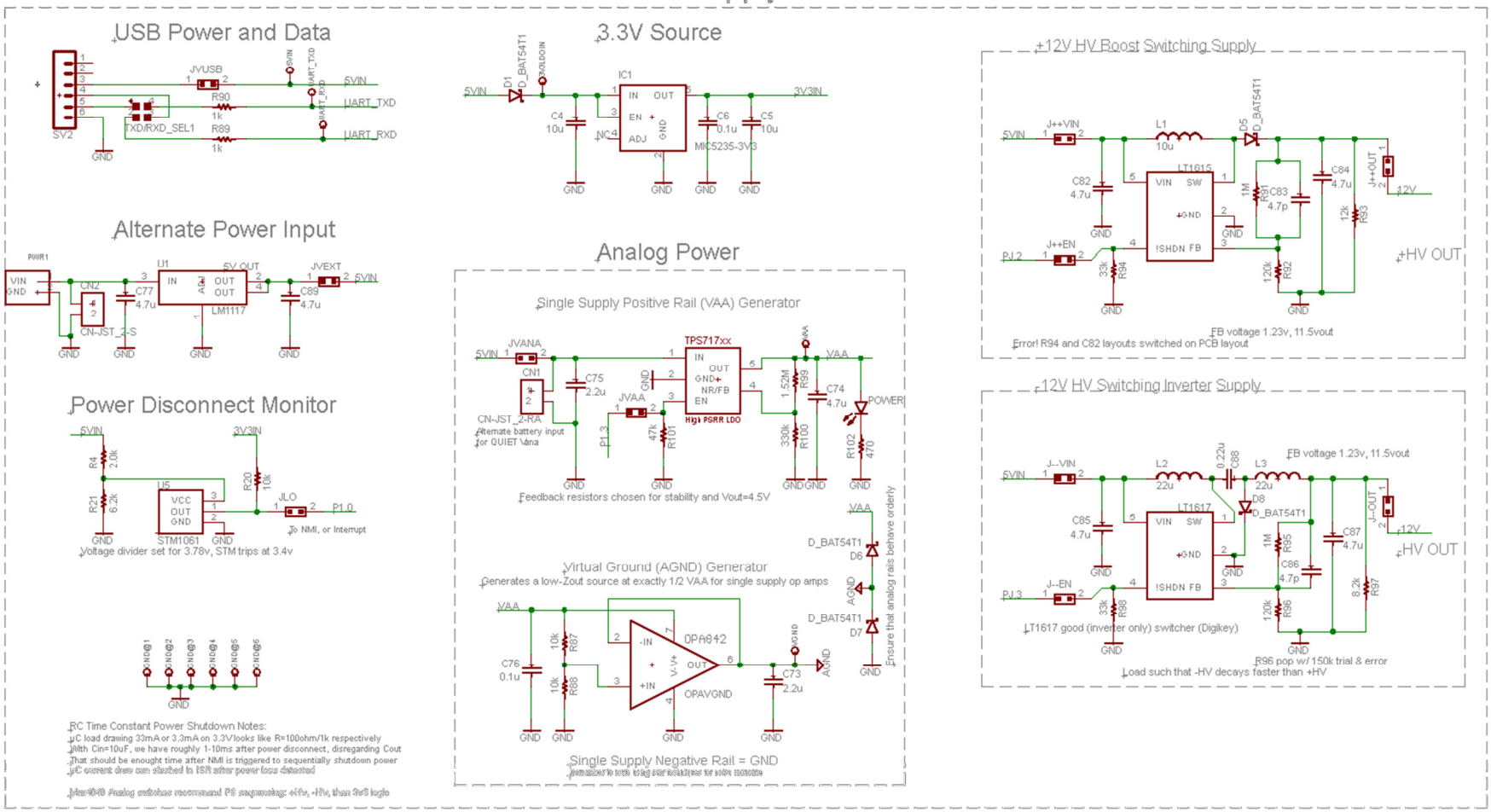

Legacy Preamp (LP) Circuit
Preamp Clamp Analog Switch
Descriminator
Band Pass Filter (BPF)
Variable Gain Amplfier (VGA)
Analog Single Supply 5.0V
Virtual Gnd 2.5v
VAA = Analog Single Supply
AGND = Virtual Ground (VAA/2)
AGND2 = Analog ground (connected to GND in one spot)

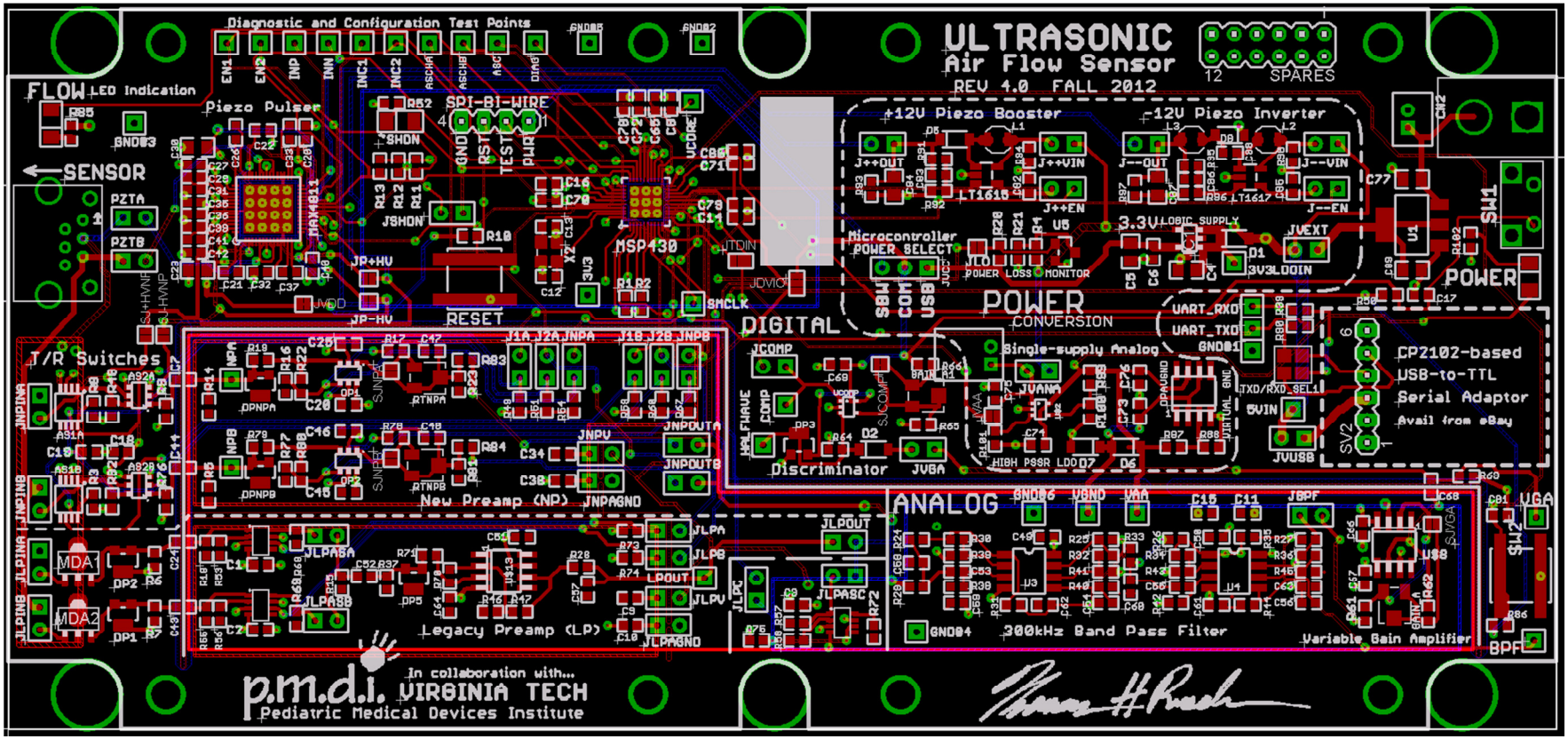

New Preamp (NP) Analog Circuit
Bipolar +/-12V
Single Supply 5.0V
Virtual Gnd 2.5V
Feedback capacitor for stabiltiy (see datasheet)
PZTA
JNPINA
AS1A
AS2A
ADG1401 SPST NO
MAX4648 SPST NO
ADG1401 swiches provide extra clamping
Optional (do math to see if necessary)
Avail at Digikey
Provide Jumper to use the INC pin from uC
MAX4648 are T/R switches
Avail at Digikey
Can Jumper ASW1&2 lines from LP
Separate or isolate +V and Vgnd?
A: Use star grounding techniques
LMV791
OP1
OP2
New Preamp: SM73305 (not available, however)
Alternate: LMV791 (avail at Mouser)
Try inverting config, don't forget + input bias res to virt. gnd
Non-inverting decreases path resistance
These resistors are part of the BPF
Change to lower R (ie10k or 20k) to avoid transient fault?
Route AGND across jumper
PZTB
JNPINB
BPFIN
JNFOUTA
JNFOUTB
J1A
J2A
J1B
J2B
JNPV
JNPAGND
SJ+HVNP
SJ-HVNP

MCU and auxilary components



Power supply chain




ULTRASONIC
Air Flow Sensor
REV 4.0 FALL 2012
Diagnostic and Configuration Test Points
FLOW
LED Indication
Piezo Pulser
SENSOR
PZTA
PZTB
SPI-BI-WIRE
MAX4011
MSP430
RESET
+12V Piezo Booster
+12V Piezo Inverter
LT1615
Microcontroller
POWER SELECT
3.3V
LOGIC SUPPLY
POWER LOSS MONITOR
POWER
CONVERSION
DIGITAL
Single-supply Analog
T/R Switches
New Preamp (NP)
Discriminator
Legacy Preamp (LP)
ANALOG
300kHz Band Pass Filter
Variable Gain Amplifier
CP2102-based
USB-to-TTL
Serial Adaptor
Avail from eBay
In collaboration with...
p.m.d.i. VIRGINIA TECH
Pediatric Medical Devices Institute

Appendix B: MCU Firmware Source Code - C files

TOF_Firmware_4_0 : main.c

```
/*****************************************************************************
* main.c
*
* This is the main program for operating the ultrasonic time-of-flight
* flowmeter based around TI's MSP430F5172 microcontroller with high-resolution
* timerD peripheral.
*
* Created by: Thomas Ruscher, Version 4.0
*
* Rev. 4.0, 4th Gen prototype, modified Oct, 2012
*
*****************************************************************************/

#include "msp430f5172.h"
#include "definitions.h"
#include "clock_setup.h"
#include "HVIO.h"
#include "Comm.h"

void main(void) {
    WDTCTL = WDTHOLD + WDTPW;           // Stop Watchdog
    DigitalOutputInit();                // De-float digital outputs early (good habit, though critical lines have pull
resistors)

    SetVCoreUpShortcut(1);              // uC requires highest core voltage in order to run at 25MHz
    SetVCoreUpShortcut(2);              // VCore MUST be changed one level at a time with delays
    SetVCoreUpShortcut(3);
    PJOUT |= (PosVen + NegVen);         // Start +/- 12V power supplies
    PJDIR |= BIT0;                      // Set pin PJ.0 to output
//  OutputSMCLK();                      // Un-comment to enable OutputSMCLK to measure clock tick execution times
    ClockInit25Mhz();                   // Boot the 25MHz crystal, and switch MCLK to it
    TD0HClockInit();
    TD0CCRInit();
    SerialInit();
    PushbuttonInit();
    // TimerA0 controls measurement frequency. Enter number of ACLK cycles (ACLK period should be 1.28us)
    TA0Init(31250);                     // E.g. 31250=exactly40.00ms(25samp/s for data streaming); 781=1ms (1ksamp/s)
    __bis_SR_register(GIE);             // Intrinsic function to enable interrupts
```

```
    while(1) {
        MakeMeasurement();
        __bis_SR_register(CPUOFF + GIE);        // Wait in LPM0.
    }
}

// Momentary Pushbutton ISR
#pragma vector = PORT2_VECTOR
__interrupt void PORT2_ISR(void) {
    __delay_cycles(65000);              // 4ms delay (crude debounce)
    if (P2IFG & SW1) {
        P2IFG &= ~SW1;
    }
    if (P2IFG & SW2) {                  // Switch 2 pressed?
        PJOUT ^= SHDN;                  // Toggle MAX4811 SHDN
        P2IFG &= ~SW2;                  // Clear SW2 interrupt flag
    }
}

#pragma vector = TIMER0_A0_VECTOR
__interrupt void TIMER0_A0_ISR(void) {
    TA0CCTL0 &= ~CCIFG;                 // Clear TimerA interrupt flag
    __bic_SR_register_on_exit(CPUOFF);  // Exit LPM0
}
```

Source File: HVIO.c

(Key program file to implementing acoustic TOF measurement)

```
/*********************************************************************
 * HVIO.c
 * (Originally HVIO stood for "High Voltage and Input/Output")
 *
 * This important source file enables key functionality of the pitch catch schemes
 *
 *  Created on: Jun 5, 2012
 *  Last revised for Trach v4 on Oct 2012
 *      Author: Thomas Ruscher
 *********************************************************************/
// Pulser API

#include "msp430f5172.h"
#include "definitions.h"
#include "HVIO.h"
#include "Comm.h"

static unsigned int txEN = 0;
static unsigned int txClamp = 0;
static unsigned int rxClamp = 0;
static unsigned int rxOPA = 0;
unsigned int validData = 0;             // Flag for valid data
static int i = 0;

// TOF measurement variables (TODO: Determine if these variables should be static or volatile)
volatile unsigned int RE1_temp = 0;         // 1st Rising Edge
volatile unsigned int RE2_temp = 0;
volatile unsigned int FE1_temp = 0;         // 1st Falling Edge
volatile unsigned int FE2_temp = 0;
volatile unsigned int Peak_temp = 0;        // Peak of one
volatile unsigned int Peak_m1 = 0;
volatile unsigned int Peak_m2 = 0;
volatile signed int DeltaT = 0;
signed int DeltaTCalib = 133;  // Average offset (ns) of mode 1 and mode 2 acoustic peaks with stagnant airflow
// Calibration offset measured 6/27/12 by sealing ends of sensor and altering code to take 500 samples
volatile signed int DeltaTAvg = 0;
int FlowDir = 0;                            // Flow Direction initialized to zero
volatile signed int DeltaTshort = 0;
volatile unsigned char DeltaT8bit = 0;
```

```
// DigitalOutputInit configures output pins related to the Pulser and Analog Switches
void DigitalOutputInit(void) {
    // Assert port output registers LOW to begin with (remember pins are not outputs until PxSEL is set)
    PJOUT =  0x00;
    P3OUT =  0x00;
    P2OUT =  0x00;
    // Write to port selection registers to set pins to output (which is now ensured to start LOW)
    PJDIR |= SHDN + PosVen + NegVen;        // Set SHDN, power pins to output
//  P2OUT &= ~ASW3; // Assert ASW3 starts low
    P3DIR |= 0xFF;                          // Entire P3
    P2DIR |= ASW3 + BIT4;                   // ASW3 and P2.4 (used for diagnostic pulsing)
    PJDIR |= FlowLED;                       // Set FlowLED to output
}

void TD0CCRInit(void) {
    // Setup CCR registers here
    P1SEL |= (BIT6 + BIT7);                 // Set P1.6 and P1.7 as CCRO and CCR1 inputs
    TD0CTL2 |= (TDCAPM0 + TDCAPM1);         // Dual capture mode for both CCR registers
    TD0CCTL0 |= (CAP + SCS + CM_0);         // CCR0 = Capture mode, sync, assert no capture mode (will be FE later)
    TD0CCTL1 |= (CAP + SCS + CM_0);         // CCR1 = Capture mode, sync, assert no capture mode (will be RE later)
    TD0CCTL0 |= CCIE;                       // Interrupt Enable TD0CCR0
}

// MakeMeasurement() is an important function
void MakeMeasurement(void) {
    InterruptsOff();
    PJOUT |= BIT0;                          // Start of diagnostic pulse for loop
    PJOUT &= ~BIT0;
    TD1Config();
    for (i = 0; i <30; i++) {  // Careful setting i not to cause DeltaTAvg int to overflow
        // The sequence contained within this FOR loop constitutes the heart of the measurement system.
        // Essentially, we are acoustically Tx'ing from one transducer, Rx'ing on the other.
        // We measure and record the arrival time of an acoustic peak at Rx transducer.
        // Then we reverse roles of the transducers and repeat the measurement.
        // We calculate the difference in Time Of Flight (DeltaT) between the transducers.
        Set_AtxBrx_m1();
            PitchCatch();
            CheckAndTransferCCRs();
            CalcAcousticPeak();
        Set_BtxArx_m2();
            PitchCatch();
            CheckAndTransferCCRs();
            CalcAcousticPeak();
```

```
        CalcDeltaT();
        DeltaTAvg += DeltaT;
    }
    DeltaTAvg /= i;
    DeltaTshort = DeltaTAvg; // was  + 1000
//  DeltaTshort /= 8;
    DeltaT8bit = DeltaTshort;               // casts from int to byte for uart data transmission
    //  CalcFlowRate();
    //  if TimeOutFlag == 1, return error or special value of CalcDelay signifying blocked channel / low signal
    DeltaTAvg = 0;                          // Reset DeltaTAvg for next measurement
    InterruptsOn();
    SerialTX(DeltaT8bit);
}


//void CalibDeltaTOffset(void)   // An optional future program method to calibrate no-flow offset

// User must manually disable all CCIE registers in use that are not related to TimerD0 and TimerD1
// so that there is not interruption during time critical code.  If more interrupts are added for peripheral functions
// (e.g. communication or display) they must be disabled/enabled here so that they do not interrupt time critical code.

static void InterruptsOff(void) {           // Disables all interrupts that could affect time critical pulsing
    P2IE = 0x00;                            // Port 2 (Pushbutton Switch) interrupts
    TA0CCTL0 &= ~CCIE;                      // TimerA CCR0 interrupt
    // UCB0IE = 0; ??
}

static void InterruptsOn(void) {            // Restores all interrupts to previous states
    P2IE  |= (SW1 + SW2);                   // SW1 + SW2 Interrupt re-enable
    TA0CCTL0 |= CCIE;                       // TimerA CCR0 interrupt re-enable
    // UCB0IE |= UCTXIE;                    // Enable TX interrupt
}

static void TD1Config(void) {
    TD1CTL0 |= TDSSEL_2;                        // Select SMCLK as source for Timer_D1
    TD1CCTL2 |= CCIE;                           // Interrupt Enable TD1CCR2
    TD1CCTL1 |= CCIE;                           // Interrupt Enable TD1CCR1
    TD1CCTL0 |= CCIE;                           // Interrupt Enable TD1CCR0
    // TD1CCR2 sets the time to turn off RxClamp, turn on rxOPA and ASW3 (if needed)
    TD1CCR2 = 0x00FA;                           // 0x00FA = 250 SMCLK cycles = 10.0us
    // TD1CCR0 sets the time to start TD0 (CCR0 is used b/c it has its own high priority interrupt vector)
    // Calibrate such that TD0R starts incrementing and TD0CCR0/1 start capturing
    // during the middle of analog signal pulse train (see thesis for more info)
    TD1CCR0 = 0x0674;                           // 0x0674 = 1652 SMCLK cycles = 66.08us
    // TD1CCR1 sets the end of measurement and acts as a timeout
```

```
    TD1CCR1 = 0x07B7;                           // 0x0753 = 1975 SMCLK cycles = 79.00us
}

// Pitch Catch Mode 1: Set PZTA TXing, PZTB RXing
static void Set_AtxBrx_m1(void) {
    txEN = EN1;
    txClamp = INC1;
    rxClamp = INC2;
    rxOPA = ASW2;
}

// Pitch Catch Mode 2: Set PZTB TXing, PZTA RXing
static void Set_BtxArx_m2(void) {
    txEN = EN2;
    txClamp = INC2;
    rxClamp = INC1;
    rxOPA = ASW1;
}

static void CheckAndTransferCCRs(void) {
    // Future improvements: Need code to insure that all captured values are in range and valid
    // And check overflow registers.  If all readings OK, set validData = 1;
    //  if (validData) {        // Multiply following values by 5 to get delay in nanoseconds when using TimerD HR mode
    FE1_temp = TD0CL0;
    FE2_temp = TD0CCR0;
    RE1_temp = TD0CL1;
    RE2_temp = TD0CCR1;
//  }
    __delay_cycles(400);            // Placeholder for additional data processing execution time
}

static void CalcAcousticPeak(void) {
    // Calculates midpoint of a single pulse (which is the acoustic peak), given rising edge (RE) and falling edge (FE)
    // Works by converting measurement from clock cycles to ns, summing RE and FE and dividing by two.
//  RE1_temp *= 40;
//  FE1_temp *= 40;
    Peak_temp = FE1_temp + RE1_temp;
//  Peak_temp /= 2;  // can possibly optimize timing by leaving out division
    if (txEN == EN1) {
        Peak_m1 = Peak_temp;
    }
    if (txEN == EN2) {
        Peak_m2 = Peak_temp;
    }
}
```

```
static void CalcDeltaT(void) {
    DeltaT = Peak_m2 - Peak_m1 + 128; // - DeltaTCalib;  // DeltaT > 128 : Inhaling;    DeltaT < 128 : Exhaling
}

/************************ START TIME CRITICAL CODE SEQUENCES ************************/
// The functions and ISRs contained herein constitute one side of the pitch-catch measurement

// PitchCatch() always executes first and ends in LPM0.  It leaves room for 4 ISRs to execute afterwards.
// The function controls TX enable, TX and RX clamps, and timing of INP and INN pulses.  Exits in LPM0 in preparation
for interrupts
static void PitchCatch(void) {
    // Typical execution time on register setting commands is 5 clock cycles (5*40ns = 200ns)
    P3OUT = (txEN + rxClamp);       // Enable Tx channel, clamp Rx channel to minimize crosstalk.
    __delay_cycles(25);                     // EN requires 1us to activate
    // ***WARNING: Simultaneous activation of INP and INN will cause a destructive shoot-through
    // condition which will cause the the MAX4811 chip to be destroyed.***
    TD1CTL0 |= MC_2;                        // Start TimerD1 (non-high res) in Continuous mode
    __delay_cycles(9);                      // Corrects ISR offset for for an oscope triggering on INP rising edge
    P3OUT |= INP;                           // Begin +HV pulse (MAX4811 automatically ignores txClamp input per
datasheet)
    __delay_cycles(25);                     // +HV pulse duration (in addition to above)
    P3OUT &= ~INP;                          // End +HV pulse
    __delay_cycles(15);
    P3OUT |= INN;                           // Begin -HV pulse
    __delay_cycles(10);                     // -HV pulse duration
    P3OUT &= ~INN;                          // End -HV pulse
    P3OUT = (txClamp + rxClamp);            // Disable Tx channel, but leave clamps in place
    __bis_SR_register(CPUOFF + GIE);        // Wait in LPM0.  CPU execution returns here after all ISRs finish
executing.
}

// TO CALIBRATE ISR timing, set an oscilloscope to trigger on the INP rising edge, then look at P2.4 diagnostic signal
// Leave diagnostic pulses in place b/c they affect timing sequence

#pragma vector = TIMER1_D1_VECTOR
__interrupt void TIMER1_D1_ISR(void) {
    P2OUT |= BIT4;                  ////// Diagnostic pulse HIGH
    P2OUT &= ~BIT4;                 ////// Diagnostic pulse LOW
    // PREPARE ANALOG RECEIVE -- 1ST HW Interrupt to execute (when TD1 reaches TD1CCR2)
    // Turns on appropriate analog switches routing signal from the RX channel, adjusts Rx clamp, clears TD0CCRs
    if (TD1CCTL2 & CCIFG) {         // Check TD1CCR2 Interrupt Flag
        PJOUT &= ~(PosVen + NegVen);// Turn off noisy HV power supplies
        P3OUT = txClamp;            // Release rxClamp in preparation to receive signal, leave txClamp activated
        __delay_cycles(25);         // Pause 1us to allow transients from release of rxClamp to settle
```

```
        P3OUT |= rxOPA;             // Turn on receive channel op amp

        // Reset TD0CCRs to 0x00 to enable detection of low/no acoustic signal
        // Problem: these registers act like they are read only, they don't seem to clear as seen by debugging
        TD0CL0 = 0x00;
        TD0CCR0 = 0x00;
        TD0CL1 = 0x00;
        TD0CCR1 = 0x00;

        TD1CCTL2 &= ~CCIFG;         // Clear CCR2 interrupt flag
    }

    // TIDY UP -- 4TH and last HW Interrupt routine to execute (when TD1 reaches TD1CCR1)
    // Asserts that TD0CCRs are stopped, resets TDR0 and TDR1, turns off all pulser outputs and analog switches
    if (TD1CCTL1 & CCIFG) {         // Check TD1CCR1 Interrupt Flag
        TD0CCTL1 &= ~(CM_1 + CCIFG);// Stop CCR1 captures and clear flag in case TIMER0_D0_ISR never executed
        TD0CCTL0 &= ~(CM_2 + CCIFG);// Do the same for CCR0
        TD0CTL0 &= ~MC_2;           // Stop TDR0
        TD0CTL0 |= TDCLR;           // Reset TDR0
        TD1CTL0 &= ~MC_2;           // Stop TDR1
        TD1CTL0 |= TDCLR;           // Reset TDR1
        P3OUT = 0x00;               // Turn off all pulser inputs and rxOPA
        PJOUT |= (PosVen + NegVen); // Turn back on HV power supplies
        TD1CCTL1 &= ~CCIFG;         // Clear CCR1 interrupt flag (TIMER1_D1_ISR)
        __bic_SR_register_on_exit(CPUOFF);  // Exit LPM0 to return to last line in PitchCatch()
    }
}

// SETUP CAPTURE -- 2ND HW Interrupt to execute (when TD1 reaches TD1CCR0)
// Starts TimerD0 in middle of incoming analog waveform (just before strongest acoustic pulse peak)
// Also enables TD0 capture registers at precise moments with the following rationale:
// Starting TD0 CCRs 180deg from their respective REs and FEs insures max flow measurement range w/o error
#pragma vector = TIMER1_D0_VECTOR
__interrupt void TIMER1_D0_ISR(void) {
    P2OUT |= BIT4;                  //////
    P2OUT &= ~BIT4;                 //////
    TD0CTL0 |= MC_2;                // Start TimerD0: TDR0 in Continuous mode
    P2OUT |= BIT4;                  //////
    P2OUT &= ~BIT4;                 //////
    TD0CCTL1 |= CM_1;               // CCR1 = Capture on rising edge
    __delay_cycles(9);              // Calibrate the setting of CCR0 to be 1.5us before the first rising edge
    P2OUT |= BIT4;                  //////
    P2OUT &= ~BIT4;                 //////
    TD0CCTL0 |= CM_2;               // CCR0 = Capture on falling edge
    TD1CCTL0 &= ~CCIFG;             // Clear CCR0 interrupt flag (TIMER1_D0_ISR)
```

```
}

// CAPTURE -- 3RD HW Interrupt executes on falling edge #2 which triggers TD0CCR0 CCIFG
// (This ISR only executes if an acoustic signal is present)
// Immediately stops the CCR registers from capturing
#pragma vector = TIMER0_D0_VECTOR
__interrupt void TIMER0_D0_ISR(void) {
    TD0CCTL1 &= ~CM_1;              // CCR1 = Stop capturing on rising edges first
    TD0CCTL0 &= ~CM_2;              // CCR0 = Stop capturing on falling edges
    P2OUT |= BIT4;                  //////
    P2OUT &= ~BIT4;                 //////
    TD0CCTL1 &= ~CCIFG;             // Clear CCR1 Interrupt flag (Prevents CCR1 COV overflow bit from being set next
iteration)
    TD0CCTL0 &= ~CCIFG;             // Clear CCR0 Interrupt flag (TIMER0_D0_ISR)
}

/************************ END TIME CRITICAL CODE SEQUENCES ************************/
```

Source file: clock_setup.c

```
/*
 * clock_setup.c
 *
 *  Created on: Jun 6, 2012
 *  Last revised Oct 2012
 *      Author: Thomas
 */

#include "msp430f5172.h"
#include "definitions.h"
#include "clock_setup.h"

void OutputSMCLK(void) {
    PJDIR |= BIT0;                          // Set pin PJ.0 to output
    PJSEL |= BIT0;                          // Select PJ.0 to SMCLK
}

void ClockInit25Mhz(void) {
//  UCSCTL3 |= SELREF__REFOCLK;             // FLLREF source = REF0CLK aka (SELREF_2)
  UCSCTL4 = (SELM_2 + SELS_2 + SELA_2);     // All clocks = REF0CLK
  UCSCTL6 |= XT1OFF;                        // Turn off XT1 while configuring
  UCSCTL6 |= XTS;                           // HF mode
  PJSEL |= BIT4+BIT5;                       // Port select XT1
  UCSCTL6 &= ~(XT1OFF);                     // XT1 On
  // Loop until XT1 & DCO stabilizes - In this case loop until XT1 and DCo settle
  do {
      UCSCTL7 &= ~(XT1LFOFFG + XT1HFOFFG + DCOFFG);   // Clear XT1,DCO fault flags
      SFRIFG1 &= ~OFIFG;                    // Clear fault flags
  }while (SFRIFG1&OFIFG);                   // Test oscillator fault flag
  UCSCTL4 = (SELM_0 + SELS_0 + SELA_0);     // MCLK & SMCLK & ACLK = XT1
  UCSCTL5 |= DIVA__32;                      // ACLK divider = 32
}

// TimerA is used to set the measurement rate, the rate at which each sequence of pitch catches
// that constitutes one measurement is performed
void TA0Init(unsigned int timerPeriod) {
    TA0CTL |= TASSEL_1;                     // Select TimerA0 clock source as ACLK
    TA0CCTL0 |= CCIE;                       // TimerA0 CCR0 interrupt enable
    TA0CCR0 = timerPeriod;                  // timerPeriod is number of ACLK cycles to count to
    TA0CTL |= MC_1;                         // Start TimerA0 in UP mode (counts to CCR0 value)
}
```

```
void TD0HClockInit(void) {
    TD0CTL0 = TDSSEL_2 + TDCLR;             // SMCLK source and TD clear for good measure
    if(0) {
        TD0HCTL1 |= TDHCLKCR;               // TimerD high res clock input >15MHz
        TD0CTL1 = TDCLKM_1;                 // Select high res local clock source
        TD0HCTL0 = TDHRON + TDHREGEN;       // TDHR on all MCx modes, regulation enabled
        TD0HCTL0 |= TDHEAEN;                // High accuracy mode enabled
        TD0HCTL0 |= TDHEN;                  // Hen what?! JK. High res mode enabled
        while (TDHLKIFG == 0);              // Wait for High-resolution to lock
        __delay_cycles(20000);
    }
}

void SetVCoreUp(unsigned int level) {
// Open PMM registers for write access
PMMCTL0_H = 0xA5;
// Make sure no flags are set for iterative sequences
while ((PMMIFG & SVSMHDLYIFG) != 0);
while ((PMMIFG & SVSMLDLYIFG) != 0);
// Set SVS/SVM high side new level
SVSMHCTL = SVSHE + SVSHRVL0 * level + SVMHE + SVSMHRRL0 * level;
// Set SVM low side to new level
SVSMLCTL = SVSLE + SVMLE + SVSMLRRL0 * level;
// Wait till SVM is settled
while ((PMMIFG & SVSMLDLYIFG) == 0);
// Clear already set flags
PMMIFG &= ~(SVMLVLRIFG + SVMLIFG);
// Set VCore to new level
PMMCTL0_L = PMMCOREV0 * level;
// Wait till new level reached
if ((PMMIFG & SVMLIFG))
while ((PMMIFG & SVMLVLRIFG) == 0);
// Set SVS/SVM low side to new level
SVSMLCTL = SVSLE + SVSLRVL0 * level + SVMLE + SVSMLRRL0 * level;
// Lock PMM registers for write access
PMMCTL0_H = 0x00;
}

void SetVCoreUpShortcut(unsigned int level) {  // Assume 1MHz DCO = MCLK
// Open PMM registers for write access
PMMCTL0_H = 0xA5;

// Set SVS/SVM high side new level
SVSMHCTL = SVSHE + SVSHRVL0 * level + SVMHE + SVSMHRRL0 * level;
// Set SVM low side to new level
```

```
SVSMLCTL = SVSLE + SVMLE + SVSMLRRL0 * level;

// Wait till SVM is settled
__delay_cycles(50000);

// Clear already set flags
PMMIFG &= ~(SVMLVLRIFG + SVMLIFG);
// Set VCore to new level
PMMCTL0_L = PMMCOREV0 * level;

// Wait till new level reached
__delay_cycles(50000);

// Set SVS/SVM low side to new level
SVSMLCTL = SVSLE + SVSLRVL0 * level + SVMLE + SVSMLRRL0 * level;

// Delay for good measure:
__delay_cycles(50000);

// Lock PMM registers for write access
PMMCTL0_H = 0x00;
}
```

Source file: Comm.c

(communications)

```
/*
 * Comm.c
 * A file about communication.
 *  Created on: Jun 19, 2012
 *  Last modified for v4: Oct 2012
 *      Author: Thomas
 */

#include "msp430f5172.h"
#include "definitions.h"
#include "Comm.h"

unsigned char* PTxData;                     // Pointer to TX data
unsigned char TXByteCtrB;

// Write 0x00 0xC0 to change lines
void SerialInit(void) {
    UCA0CTL1 |= UCSSEL__SMCLK;              // SMCLK sources UCA0's BRCLK
    // Setting baud rate: 25MHz BRCLK, Target baud rate = 230400bps. Per formulae on datasheet pg 758:
    // N = 108.507, therefore UCBRx = 108 and UCBRSx = 4
    UCA0BR0 |= 0x6C;                        // Set UCBRx low byte to 0x6C = 108
    UCA0MCTL |= UCBRS_4;                        // Set second stage modulation value to 4
    P1DIR |= BIT1;                          // P1.1 Output
    P1SEL |= BIT1;                          // P1.1 UCA0TXD
}

void SerialTX(int TXdata) {
    UCA0CTL1 &= ~UCSWRST;                   // Clear Reset bit to enable UART
    UCA0TXBUF = TXdata;                     // Write to TX buffer
//  __delay_cycles(1500);
//  UCA0CTL1 &= ~UCSWRST;                   // Clear Reset bit to enable UART
//  UCA0TXBUF = 0x2F;
//  UCA0CTL1 |= UCSWRST;                    // Reset to disable (sleep) UART
}
```

```
void PushbuttonInit(void) {
    P2IFG =  0x00;                          // Insure all interrupt flags are cleared
    P2IES |= (SW1 + SW2);                   // SW1 + SW2 Falling edge interrupts
    P2IE  |= (SW1 + SW2);                   // SW1 + SW2 Interrupt enable
}

//-------------------------------------------------------------------------
// USCI B, not A.... left for reference
// The USCIAB0TX_ISR is structured such that it can be used to transmit any
// number of bytes by pre-loading TXByteCtrB with the byte count. Also, TXData
// points to the next byte to transmit.
//-------------------------------------------------------------------------
#pragma vector = USCI_B0_VECTOR
__interrupt void USCI_B0_ISR(void) {
  if (TXByteCtrB)                           // Check TX byte counter
  {
    UCB0TXBUF = *PTxData++;                 // Load TX buffer
    TXByteCtrB--;                           // Decrement TX byte counter
  }
  else
  {
    UCB0CTL1 |= UCTXSTP;                    // I2C stop condition
    UCB0IFG &= ~UCTXIFG;                    // Clear USCI_B0 TX int flag
    __bic_SR_register_on_exit(CPUOFF);      // Exit LPM0
  }
}
```

Appendix C: MCU Firmware Source Code - Header files

Definitions.h

```
/*
 * definitions.h
 *
 *  Created on: Jun 22, 2012
 *      Author: Thomas
 */

#ifndef DEFINITIONS_H_
#define DEFINITIONS_H_

// PORT 2 ASSIGNMENTS:

//Include assignments for NP

#define SW1         (BIT7)                  // Momentary Pushbutton Switch #1   Pin 22
#define SW2         (BIT5)                  // Momentary Pushbutton Switch #2   Pin 20
#define ASW3        (BIT6)                  // Analog Switch 3                  Pin 21

// PORT 3 ASSIGNMENTS:
#define ASW1        (BIT0)                  // Analog Switch 1                  Pin 23
#define ASW2        (BIT1)                  // Analog Switch 2                  Pin 24
#define EN1         (BIT2)                  // Enable Ch1                       Pin 29
#define EN2         (BIT3)                  // Enable Ch2                       Pin 30
#define INC1        (BIT4)                  // Clamp Control Ch1                Pin 31
#define INC2        (BIT5)                  // Clamp Control Ch2                Pin 34
#define INP         (BIT6)                  // Positive HV (+HV) pulse          Pin 35
#define INN         (BIT7)                  // Negative HV (-HV) pulse          Pin 36

// PORT J ASSIGNMENTS:
#define FlowLED     (BIT1)                  // FLOW Indicator LED               Pin 8
#define PosVen      (BIT2)                  // +12V boost supply enable         Pin 9
#define NegVen      (BIT3)                  // -12V inverter supply enable      Pin 10
#define SHDN        (BIT6)                  // MAX4811 Shutdown (Active LOW)    Pin 28
```

```
// OTHER DEFINES:
// Flow Direction: Transducer (PZT) A is always more distal to the patient, PZT B always closer to patient
#define INHALE          11          // Patient is inhaling (PZT A is upstream of flow direction, PZT B downstream)
#define EXHALE          12          // Patient is exhaling (PZT B is upstream of flow direction, PZT A downstream)
// Note that above definitions for INHALE and EXHALE are arbitrarily chosen to avoid confusion with mode1/mode2
// Analog Preamp designations:
#define LP              1                   // Legacy Preamp (LP)
#define NP              2                   // New Preamp (NP)


// OTHER NOTES:
// Port J has no select line (or register?) in MSP430x5xxx datasheet, though this may be typo??

#endif /* DEFINITIONS_H_ */
```

HVIO.h

```
/*
 * HVIO.h
 * This file maps hardware I/O pins from the TI MSP430F5172 uC
 * to the Maxim IC MAX4811 dual channel pulser IC.
 * It is critical to have a clear and easy-to-use pin definitions to write
 * time-consistent code and avoid shoot-through conditions on the pulser.
 * Rationale for bit masks: while bitfields would make this code much more
 * readable, there are some bitfield compiler issues with CCSv5, which is why
 * this code was written to use bit masks.
 *
 *  Created on: Jun 5, 2012
 *      Author: Thomas Ruscher
 *      Rev 1.0: Setup
 */

#ifndef HVIO_H_
#define HVIO_H_


void DigitalOutputInit(void);
void TD0CCRInit(void);
void MakeMeasurement(void);
static void InterruptsOn(void);
static void InterruptsOff(void);
static void TD1Config(void);
static void Set_AtxBrx_m1(void);
static void Set_BtxArx_m2(void);
static void PitchCatch(void);
static void CheckAndTransferCCRs(void);
static void CalcAcousticPeak(void);
static void CalcDeltaT(void);

//void DiagnosticTimingPulse(void);

#endif /* HVIO_H_ */
```

Clock_setup.h

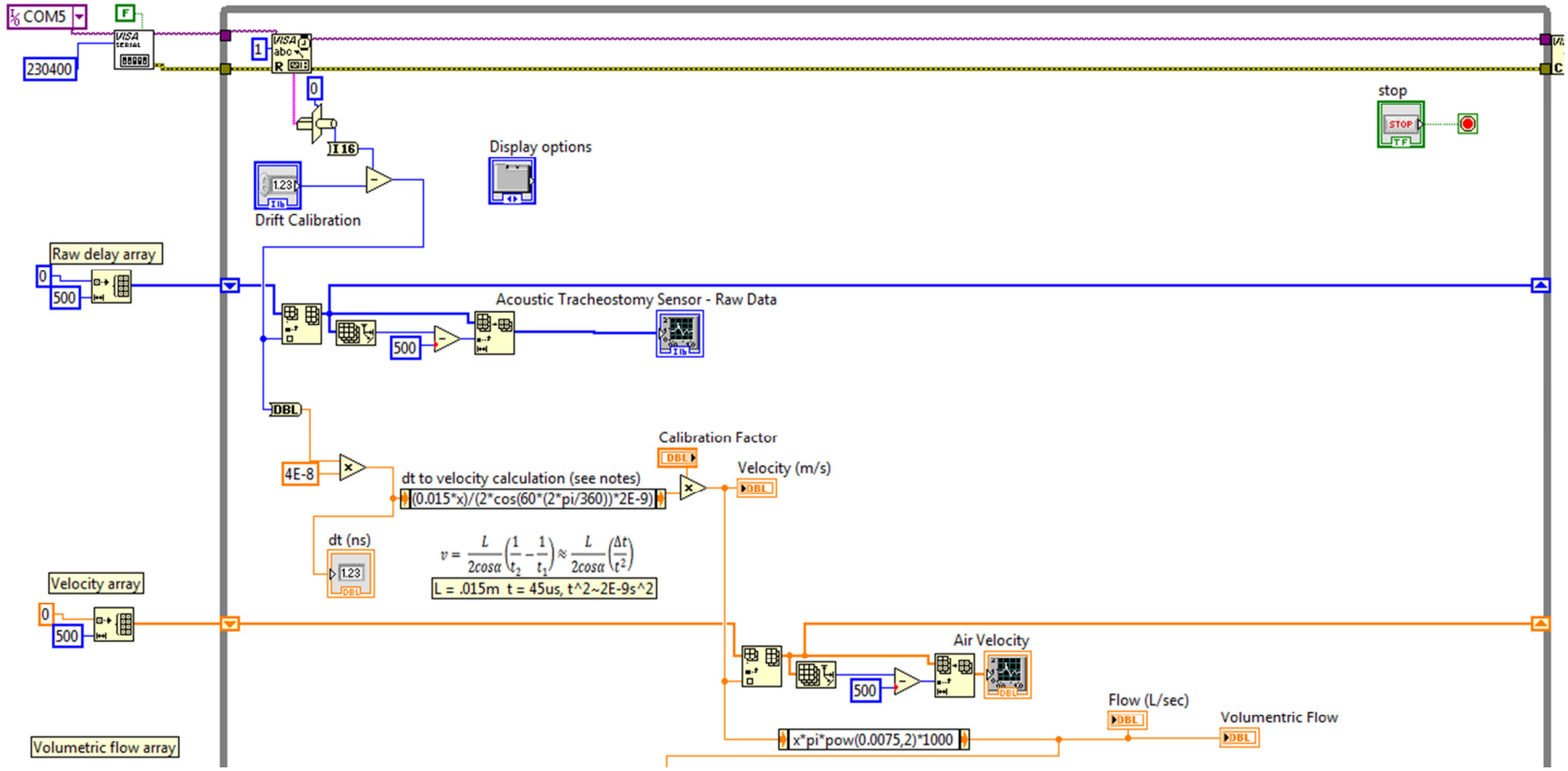
```
/*
 * clock_setup.h
 *
 *  Created on: Jun 7, 2012
 *      Author: Thomas
 */

#ifndef CLOCK_SETUP_H_
#define CLOCK_SETUP_H_


void OutputSMCLK(void);
void ClockInit25Mhz(void);
void TA0Init(unsigned int timerPeriod);
void TD0HClockInit(void);
void SetVCoreUp(unsigned int level);
void SetVCoreUpShortcut(unsigned int level);

#endif /* CLOCK_SETUP_H_ */
```

Comm.h

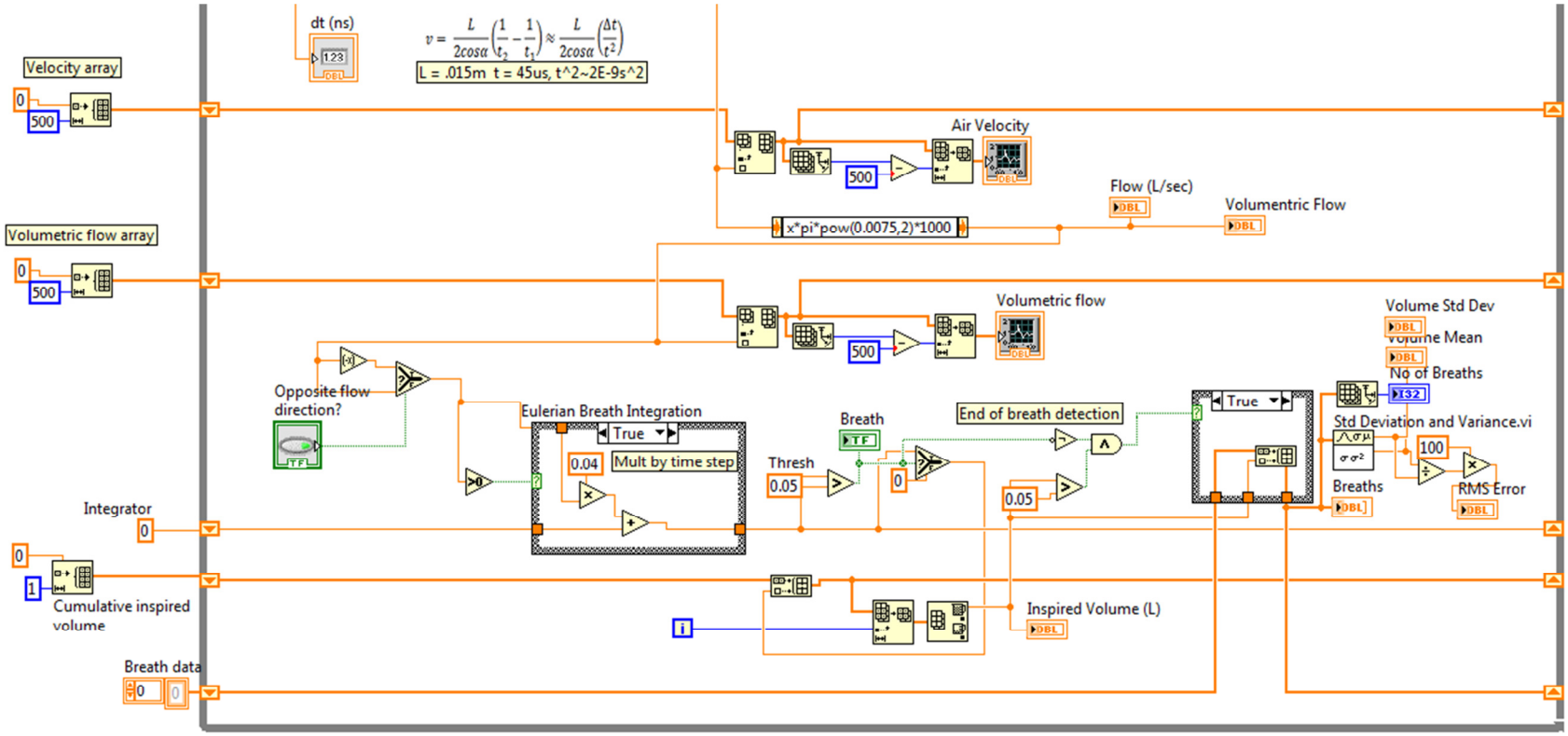
```
/*
 * Comm.h
 *
 *  Created on: Jun 19, 2012
 *      Author: Thomas
 */

#ifndef Comm_H_
#define Comm_H_


void SerialInit(void);
void SerialTX(int TXdata);
void PushbuttonInit(void);

#endif /* Comm_H_ */
```

Appendix D: LabVIEW source code for PC-side calculations

This code reads in one-byte ΔTOF (dt) values (in 40ns increments) data values from the flow sensor, converts the values to air velocity, and multiplies by the cross section of the flow chamber (and sample period) to achieve volumetric flow rate. Arrays of past values are maintained in order to present a visual graph. Also, breath detection is achieved through volumetric flow rate integration.



(continued)


dt (ns)
L = .015m t = 45us, t^2~2E-9s^2
Velocity array
Air Velocity
Flow (L/sec)
Volumetric Flow
x*pi*pow(0.0075,2)*1000
Volumetric flow array
Volumetric flow
Volume Std Dev
Volume Mean
No of Breaths
Opposite flow direction?
Eulerian Breath Integration
True
Mult by time step
0.04
Thresh
0.05
Breath
End of breath detection
Std Deviation and Variance.vi
Breaths
RMS Error
Integrator
Cumulative inspired volume
Inspired Volume (L)
Breath data