12-2017

Design, development, and characterization of breathforce : a respiratory training system for patients with spinal cord injuries.

Kevin L Tran
University of Louisville

Recommended Citation

DESIGN, DEVELOPMENT, AND CHARACTERIZATION OF *BREATHFORCE*, A
RESPIRATORY TRAINING SYSETEM FOR PATIENTS WITH SPINAL CORD INJURIES

Kevin Ly Tran

Bioengineering B.S., University of Louisville, 2016

A Thesis

Submitted to the Faculty of the

University of Louisville

J.B. Speed School of Engineering

As Partial Fulfillment of the Requirements

For the Professional Degree

MASTER OF ENGINEERING

Department of Bioengineering

December 2017

DESIGN, DEVELOPMENT, AND CHARACTERIZATION OF *BREATHFORCE*, A RESPIRATORY TRAINING SYSETEM FOR PATIENTS WITH SPINAL CORD INJURIES
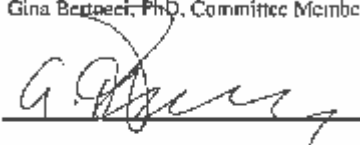
Submitted by: _____

Kevin Tran

A Thesis Approved on:

12 DEC 2017

(Date)

By the Following Reading and Examination Committee:

_____

Thomas Roussel, PhD, Thesis Director

_____

Gina Bertocci, PhD, Committee Member

_____

Alexander Ovechkin, MD, PhD, Committee Member

ii

ACKNOWLEDGMENTS

I would like to thank my family and my girlfriend, Thu Nguyen, for the love and support through this long year. I would also like to thank my thesis director, Thomas Roussel, for being such a great mentor to work with. His knowledge and guidance are extraordinary and I could not ask for a better mentor. Without all of these people, I do not know where I would be with this project.

ABSTRACT

Pulmonary and cardiovascular dysfunction are consistently reported as the leading causes of morbidity and mortality among the 1,275,000 people who are living with chronic spinal cord injury (SCI) in the United States. Respiratory-cardiovascular complications from neurological disorders (primarily COPD and sleep apnea) are currently the number one cause of death and disability in the US and, in 2020, are projected to be ranked first worldwide in terms of burden of disease.

The main goal of this project is to develop an inspiratory-expiratory training device for use in the rehabilitation of patients with respiratory motor and cardiovascular deficits that incorporates existing technologies and promotes successful training methodologies performed at the clinic and at home.

An embedded microprocessor was to convert pressure from a physiological range sensor into appropriate units and guide the user through a therapy session, while saving the data for later use by the clinician. Rechargeable batteries were used to allow for portability. A bi-directional breathing apparatus to accompany the microprocessor was developed using FDA approved, off-the-shelf parts.

Two therapy modes were programmed into the microprocessor to 1) find the max expiratory pressure (MEP) and max inspiratory pressure (MIP) of the user and 2) function as a spirometer to track and display user data during respiratory muscle training (RMT).

A transducer tester was used to apply a calibrated pressure to the device to validate the measurement accuracy. Measured values differed from the tester by 1.91%-3.78%. No drift was noticed in the device when left running for an extended period of time and humidity, moisture, and temperature effects did not affect the accuracy of the sensor measurement. A SCI test subject showed an average pressure deviation from target values (10-18.51%) that were less than that of a healthy subject (~40%).

The prototype device that was given the name BreathForce, met all the initial design criteria. Validation studies are underway for accuracy and effectiveness and future studies include hardware upgrades such as wireless communication and electronic proportional valves.

Table of Contents

LIST OF FIGURES

LIST OF TABLES

I. INTRODUCTION

Spinal cord injuries (SCI) occur across a wide variety of patients regardless of health and age. Accidents, in particular motor vehicle accidents and falls, are the most prevalent cause of SCI [1]. By 2010, an average of 17,000 new cases were being documented each year [1]. Incidence and prevalence of traumatic SCI is higher in the US than the rest of world [2]. Of the patients with SCIs, 58.3% have tetraplegia (spinal cord injuries between C1-C8 of the spinal cord where all four limbs suffer some degree of paralysis) and 41.3% have paraplegia (spinal cord injuries between T1-L5 of the spinal cord where patients may still have full function of their arms and torso, but legs suffer some degree of paralysis). More than one-half of all SCIs (56.4%) occur at the cervical level [3]. Figure 1 graphically explains the breakdown of causes of SCIs in 2010. A *complete* spinal cord injury can be described when there is total lack of sensory and motor function below the level of injury while *incomplete* is when some sensation and/or movement is possible below the level of injury.
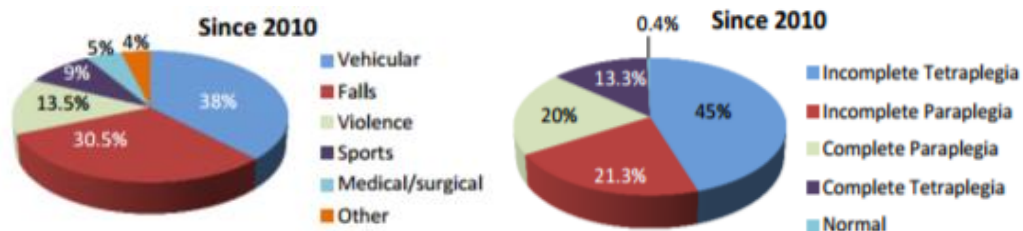
Figure 1 – The percentage breakdown of causes of SCIs (left) and type of paralysis since 2010 (right) [1]

Average yearly expenses covering health care costs as well as living expenses and the estimated lifetime costs for a patient has been shown to increase in relation to the location of the injury and age of the patient (Table 1). Costs increase from paraplegia to tetraplegia. This trend can be generally attributed to the extra care a tetraplegic patient requires when compared to a paraplegic patient. Estimated lifetime costs also increase as age of the patient decrease. As there are currently no cures for SCIs, patients will continue to pay for medical supplies and medication throughout their lifetime. A decrease in costs is seen in elderly patients compared to younger ones as cost is adjusted by remaining estimated lifespan.

Table 1 – Table of costs for patients with SCIs [1]

	Average Yearly Expenses (in 2015 dollars)		Estimated Lifetime Costs by Age at Injury (discounted at 2%)	
Severity of Injury	First Year	Each Subsequent Year	25 years old	50 years old
High Tetraplegia (C1-C4)	$1,065,980	$185,111	$4,729,788	$2,599,411
Low Tetraplegia (C5-C8)	$770,264	$113,557	$3,455,879	$2,125,674
Paraplegia	$519,520	$68,821	$2,312,846	$1,517,851
Motor Functional at Any Level	$347,896	$42,256	$1,580,148	$1,115,312

1.1 Clinical Problem

Spinal cord injuries (SCIs) can result in many symptoms, with severity and range depending on location of the SCI along the spinal cord. General SCI symptoms can include: loss of muscle movement, loss of sensation, exaggerated reflex activities, and difficulty breathing, coughing or clearing secretions from the lungs [4].

Cervical SCIs are spinal cord injuries occurring in the neck region of the spinal cord. These types of SCIs have a direct negative effect on respiratory pathways and can cause respiratory muscle paresis and/or paralysis. These forms of paralysis often lead to respiratory insufficiencies such as difficulty in breathing, as well as the inability to

cough. A cough is a short expulsion of air when foreign chemicals or objects stimulate sensory nerve fibers that line the lung walls. Failure to remove irritants can lead to respiratory infections and possibly lethal infections such as pneumonia. These respiratory infections are the main causes of morbidity and mortality of patients with cervical SCIs [5].

Currently, there are no rehabilitative therapy methods that have yet been proven to be clinically effective in restoring respiratory function after spinal cord injury. There are only potential strategies in helping prevent respiratory failure and infections in patients [6]. Methods like exercise and training to improve fitness have shown to improve respiratory function and pressure. These existing methods occur mostly in a clinical environment where the patient is under supervision by the clinician who analyzes the patient's data. These methods should be extended to home use to allow patients to regularly exercise at home while also saving data for clinicians to analyze and track patient improvements.

1.2 Project Goals

The main goal of this project is to develop an inspiratory-expiratory training device for use in the rehabilitation of users with respiratory motor and cardiovascular deficits that combines modern technology with the advanced training methodologies developed at the University of Louisville [7]. This project will provide a cheaper alternative to large, costlier medical training devices for clinicians, therapists, and consumers. A simple user interface will be designed to allow all users and/or therapists, who assist the user, to easily operate the device.

Specific Aim 1: Develop a breathing apparatus that accurately measures maximum

inhalation and exhalation pressure. A single apparatus that can handle both functionalities

eliminates the need for two separate breathing devices. While spirometry is most directly

correlated with lung volume and function, max expiratory pressure (MEP) and max

inspiratory pressure (MIP) are more closely related to respiratory motor control of the

muscles and neurological function [6].

Specific Aim 2: Develop a system that is applicable in both a clinical and home setting.

The device must blend the benefits of existing clinical and home devices: the device must

be small and battery-powered for home/remote use, and include advanced measuring

features as those available in existing clinical devices.

Specific Aim 3: Develop software to measure and set threshold levels for training sessions

and guide users through the session. Target threshold levels are calculated from a

percentage of MEP/MIP values and users will breathe through the apparatus to meet

those target threshold values. To guide the user, a screen-type device with an integrated

user interface (UI) will be used to provide visual feedback and display individualized

training session information. During and after a training session, data of the session will

be saved to a SD card for clinicians to retrieve and analyze later that will provide both a

verification of prescribed therapy compliance as well as technique to track user

improvements over time.

This training system will be designed for use by an extremely broad population

including patients with spinal cord injury, pulmonary disease (pneumonia, COPD and

others), sleep apnea, neurological disease, stroke, cardiovascular disease, and other

physiological problems associated with pulmonary, cardiovascular, and/or neurological

deficits. An advanced, compact, respiratory trainer will also be attractive to a broad

population of healthy individuals including athletes looking to improve respiratory

performance and cardiovascular function or vocal professionals seeking to improve

respiratory motor-related performances.

II. BACKGROUND

2.1 Physiology of the Spinal Cord

There are several bones and nerves that support and innervate the cervical section of the spinal cord. These bones are numbered from C1 to C8; the high-cervical region being C1-C4 and the low-cervical region being C5-C8 (Figure 2).

The C1-C3 (Figure 2) nerves play key roles in breathing. Spinal nerve C4 helps control the shoulders and the diaphragm for breathing. Damage to this section is the most severe of the spinal cord injuries as SCI here can result in paralysis of both arms, hands, trunk, and legs (quadriplegia). The ability to speak is sometimes impaired or reduced and the patient may not be able to breathe, cough, or control bowl or bladder movement [8].
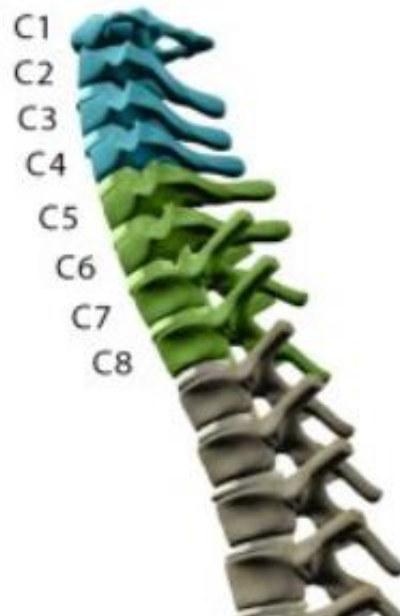
Figure 2 – Labeled cervical sections of a spinal cord [8]

The C5 (Figure 2) nerve controls the muscles of the upper body (deltoids) and biceps to allow flexion of elbow and rotation of the forearm. The C6 nerve controls wrist extensors and provides some innervation to the biceps, while spinal nerves C7 and C8 control the triceps and hands, respectively. Injury to the lower cervical section of the spinal cord can also result in similar severe symptoms like the high cervical section: Patients will either have some or total paralysis of arms and legs and will have little to no voluntary control of bowel or bladder. However, a patient with this level of injury may be able to breathe on their own and speak normally [8].

2.2   Physiology of Spinal Cord Injury

A SCI is caused when there is damage to any part of the spinal cord or nerve endings of the spinal canal. Changes of the body and severity of symptoms vary depending on the location of the SCI along the spine. A SCI often causes permanent changes in strength, sensation, and other body functions below the site the injury [4].

A SCI damages the nerve pathways that connect the body's physiological systems to the brain. For example, SCIs can affect bladder control where the brain is unable to properly communicate with the bladder. Changes in bladder control increases risk of urinary tract infections and may also cause kidney infections and kidney or bladder stones. In the circulatory system, a SCI may cause circulatory problems resulting in low blood pressure or high blood pressure (orthostatic hypotension). These circulation changes may increase risk of developing blood clots like deep vein thrombosis or pulmonary emboli [4]. Focusing primarily on respiratory complications, SCI in the cervical region can make breathing and coughing more difficult by affecting control of the abdominal and chest muscles.

When a section of the spinal cord is damaged (Figure 3), the afferent and efferent pathways (ascending and descending the spinal cord, respectively) are damaged which prevents nerve impulses from passing beyond the point of injury. In afferent pathways, respiratory centers in the brain do not receive impulses from parts of the body via peripheral chemoreceptors and baroreceptors that stimulate movement of the thoracic cage and lungs. Correspondingly, in efferent pathways, phrenic nerves and intercostal
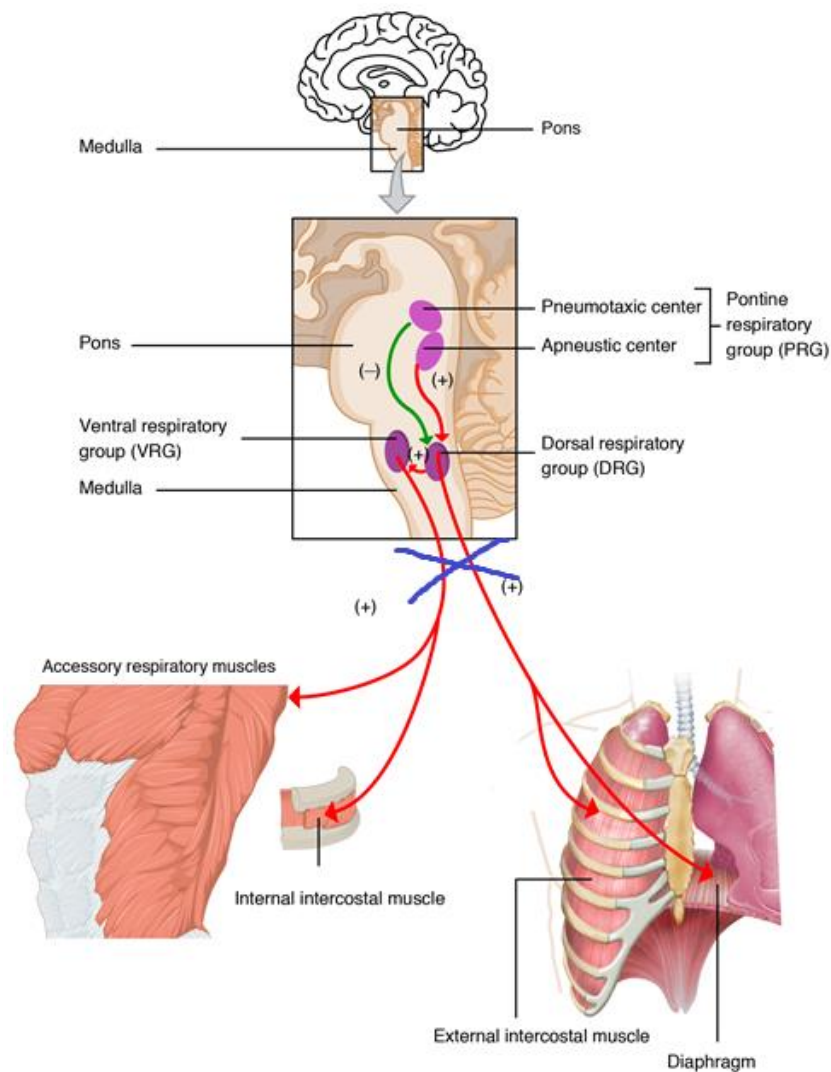
Figure 3 – Diagram of how SCI (blue x) blocks impulse signals from mid-brain to internal and external intercostal muscles and diaphragm [5]

nerves are not able to supply impulses to the diaphragm and intercostal muscles, respectively, because motor neurons in the anterior horn cells of the cervical segments cannot synapse with nerve fibers past the location of the SCI.

2.3   Physiology of Respiration

There are three phases that dictate the mechanics of respiration: inspiration, exchange of gases, and expiration. Inspiration is defined as the process of drawing air into the lungs. This occurs when muscles of the diaphragm contract, moving the diaphragm downward. External intercostal muscles contract and raise the rib cage forward (Figure 4, left). Due to these activities, the volume of chest cavity increases and results in decrease of air pressure inside the chest cavity. Oxygenated air, present outside the body, is at a higher pressure and flows rapidly into the lungs passing through the nasal cavity, pharynx, larynx, trachea, bronchi, and bronchioles.
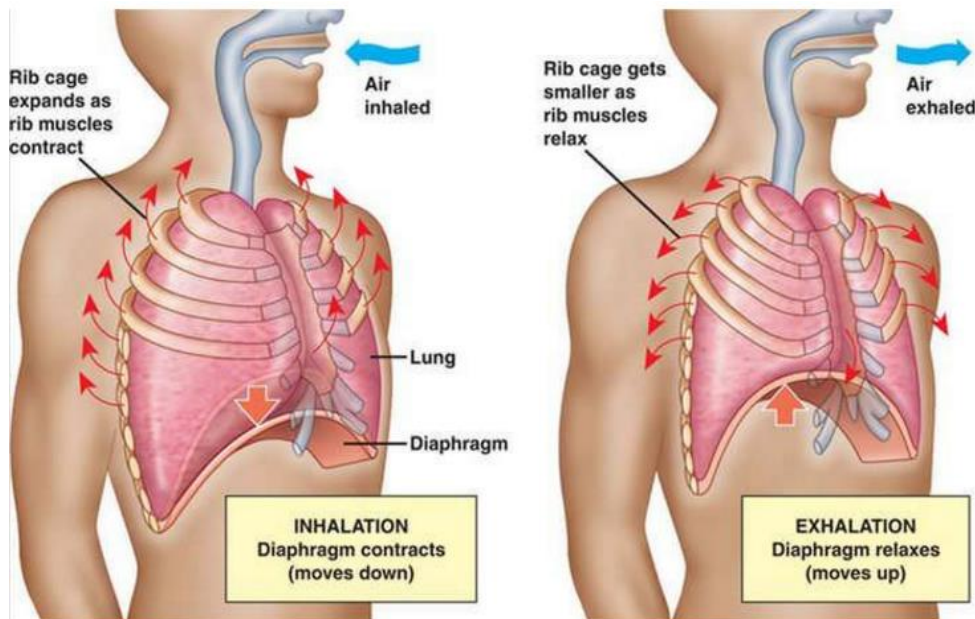
Figure 4 – Diagram showing physiology of inhalation (left)
and exhalation (right) [5]

During gas exchange in the lungs, oxygenated air reaches alveoli by passing through bronchioles. Oxygen passes through the walls of alveoli into blood present in blood capillaries. This blood is returned to the heart to be supplied to all tissues of the body. In the tissues, oxygen is used for the oxidation of food and is converted into carbon dioxide. From tissues, carbon dioxide is absorbed by blood and carried to alveoli of lungs for expiration.

During expiration, the muscles in the diaphragm relax, and the diaphragm moves back upward into its original position (Figure 4, right). External intercostal muscles relax, and internal intercostal muscles contract, lowering the rib cage backwards. This decrease in volume of the chest cavity causing the chest cavity to contract, pushing out carbon dioxide through bronchi, trachea, larynx, pharynx, nasal cavity, and nostrils.

Maximum inspiratory pressure (MIP) and maximum exhalatory pressure (MEP) represent the greatest airway pressure forced effort against an occluded airway sustained over a one second period. MEP and MIP are used to evaluate the strength of inspiratory and respiratory muscles during inspiration and respiration [6]. Doctors analyze these values to track and trend patient respiratory performance and functionality.

2.4   Spinal Cord Injury and Respiratory System Compromise

Frankel, et al. (1998, [7]), examined long-term survival in patients with SCIs and identified risk factors contributing to deaths and explored trends in causes of death over the past 50 years pertaining to SCIs patients. The study found that there was a higher mortality risk associated with location and severity of SCIs. Problems with the respiratory system was the leading cause of death, followed by urinary, then heart disease complications [5].

Depending on the vertebral section in which the SCI occurs, there is a direct correlation to the severity of respiratory system compromise. In high-cervical or low-cervical nerve damage, patients are likely to have paralysis, muscle weakness, or spasticity in the muscles used to perform normal respiration. Weakness of respiratory muscles have been linked to a ==paradoxal== breathing pattern in patients with cervical or high thoracic SCI. The upper anterior rib cage moves inward during inspiration that results from a lack of spinal motor activation of the external intercostal muscles combined with excessive compliance of the abdominal wall due to weak muscle contraction [6].

This category of patients with cervical nerve damage are frequently unable to breathe on their own and are also unable to cough. Difficulty taking a deep breath can cause anxiety and may result in dyspnea. Retained secretions from not being able to cough may cause collapse of alveoli and pooling of secretions which increases the risk of pneumonia and lung infections [6]. Weakness in respiratory muscles commonly results in the use and eventual dependency on a mechanical ventilator for breathing support.

Lung volumes can be analyzed graphically in a lung capacity chart that compares volume of lung capacity over time (Figure 5). In a lung capacity chart, tidal volume (TV) is the amount of air inspired during normal, relaxed breathing. Inspiratory reserve volume (IRV) is the additional air that can be forcibly inhaled after the inspiration of a normal TV. Expiratory reserve volume (ERV) is the additional air that can be forcibly exhaled after the expiration of a normal TV. Residual volume (RV) is the volume of air remaining in the lungs after ERV is exhaled. This kind of chart is usually obtained through a spirometer.
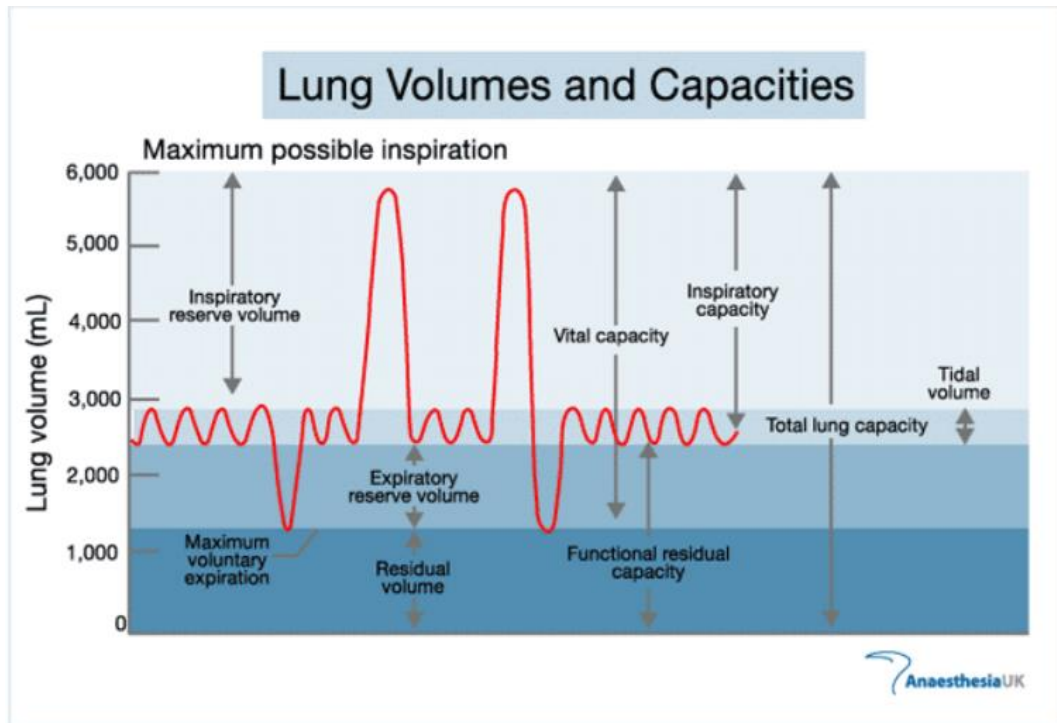
Figure 5 – Respiratory capacity of an uninjured patient [13]

In a flow volume loop curve, graphs provide a graphical illustration of a patient's spirometric effort (Figure 6). Flow is plotted against volume to display a continuous loop for inspiration to expiration. Overall shape of flow volume loop is used to interpret spirometric results. In a healthy individual, after a small amount of gas is exhaled, flow is limited by airway compression and is determined by the elastic recoil of the lung and resistance upstream of that point.

Effect of SCI on respiratory muscle function is related to a decrease in functional residual capacity, total lung capacity, expiratory reserve volume, and increased residual volume. These decreased values are due to the obstructed airways that result from the inability to cough and remove irritants that have accumulated in the lungs. Airway obstruction can be seen in the flow volume charts (Figure 6) where flow rate is very low in relation to lung volume, and a scooped-out appearance is often seen following the
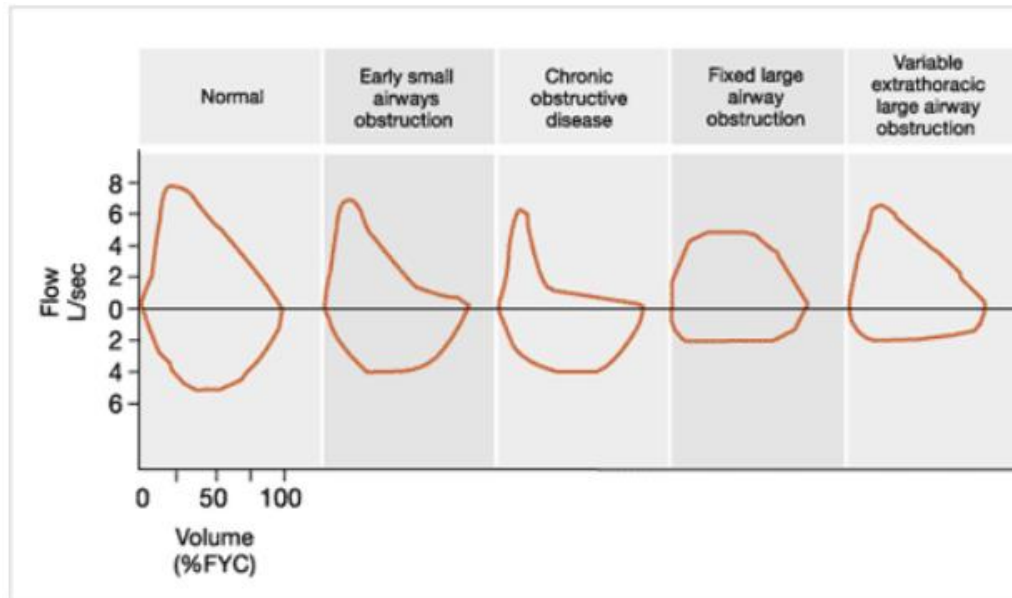
Figure 6 – Flow-volume curves that illustrate changes in respiratory capacity (area under curve) of an uninjured patient (normal) to a patient with an obstructive disease (pneumonia) [13]

point of maximal flow. Areas under the curve for obstructive diseases are generally smaller than that of a healthy individual, indicating a lower lung capacity.

Garshick, et al. (2005, [14]), examined how failures in respiratory systems were causes of death in SCI patients. He found causes of death relating to reduced pulmonary function and circulatory systems and assessed the relationship between medical conditions and other health-related factors, such as diabetes, heart disease, and reduced pulmonary function, to SCI patient mortality rates. It was found that the common underlying and contributing causes of death were diseases of the circulatory system (40%) and diseases of the respiratory system (25%) [14].

The inability to cough effectively and efficiently is a major problem for patients with SCIs. A cough is a short expulsion of air when foreign chemicals or objects stimulate sensory nerve fibers that line the lung walls. Impulses travel from the lungs to

13

the vagus nerve and into the medulla of the lower brain [15]. At the medulla, impulses synapse with neurons of efferent pathways to trigger effector muscles to inhale and cause the glottis to close and compress respiratory muscles to build pressure inside the lungs [15]. When the glottis opens, air and mucus with foreign irritants are pushed out because of high pressure in the lungs [15]. Although the vagus fibers bypass the spinal cord and should not be affected by SCI, patients still produce ineffective coughs [15].

Because there are no cures for muscle paralysis caused by a SCI, regardless of severity, treatment/therapy methods are used to get patients to work their lungs and keep the associated muscles active to prevent atrophy. There are two common treatment methods used to keep respiratory muscles active: respiratory muscle training/therapy (RMT) and passive electrical pacing with specialized equipment. RMT involves patients to actively train and inhale or exhale into a machine to reach certain target goals that reflect improvement in a patient's respiratory capabilities. Passive electrical pacing stimulates the diaphragm and intercostal muscles with electric current to assist in breathing or coughing and force inactive respiratory muscles to contract at the level the muscles should be contracting during normal breathing or coughing.

Aside from therapy training, there are methods that provide patients with an instant, temporary relief of symptoms but do not help strengthen respiratory muscles. To help with breathing, positive-pressure ventilation can be used to compensate for weak muscle control by increasing the patient's airway pressure through an endotracheal or tracheostomy tube. Positive pressure allows air to flow into the airway until the ventilator breath is terminated. To help with coughing, there are assistive cough techniques to help generate an adequate cough due to lack of abdominal strength (Figure 7Figure 7) [16].

Figure 7 – An example of an assisted cough [16]

Terson de Paleville, et al. (2011, [6]), reported that exercise in general has shown to increase fitness and improve ventilatory function of individuals with acute and chronic SCI. The study reviewed that treadmill locomotor exercises in individuals with chronic SCI elicit a metabolic response characterized by an increase in oxygen consumption, pulmonary ventilation, and heart rate. Additionally, strength training of the pectoralis muscle by doing repetitive, strenuous, isometric contraction in complete lower cervical cord-injured patients had shown to significantly improve expiratory function by significantly increasing expiratory reserve volume and decreasing residual volume of the lungs. The load from dynamic exercises resulted in an increase in breathing that are associated with the excitation of the cerebral cortex, limbic and reticular activating systems, hypothalamus, and central chemoreceptors. Training designed to specifically target and activate respiratory muscles (RMT) have shown to be successful at improving respiratory and expiratory pressures, total lung capacity, peak oxygen consumption, and

one-minute ventilation by inducing central neural remodeling through activity-dependent plasticity [6].

2.5   Training Non-Compliance

While in-clinic therapy has obvious positive benefits, significant effort is required to encourage participants to continue exercise once they are no longer under the supervision of the clinic and are at home. When patients are home, they are out of sight of their doctor and therapists. When unsupervised, patients lose motivation to continue prescribed exercise routines, and look to other options of treatment that do not require exercise, such as phrenic nerve pacing devices.

Phrenic nerve pacing devices and diaphragm pacing devices can be considered electronic pacemakers for the lungs (Figure 8). The devices stimulate the diaphragm and intercostal muscles with electric current. It is a very effective treatment that requires less exercise for patients restricted to wheelchairs. The negative aspects of this device are the
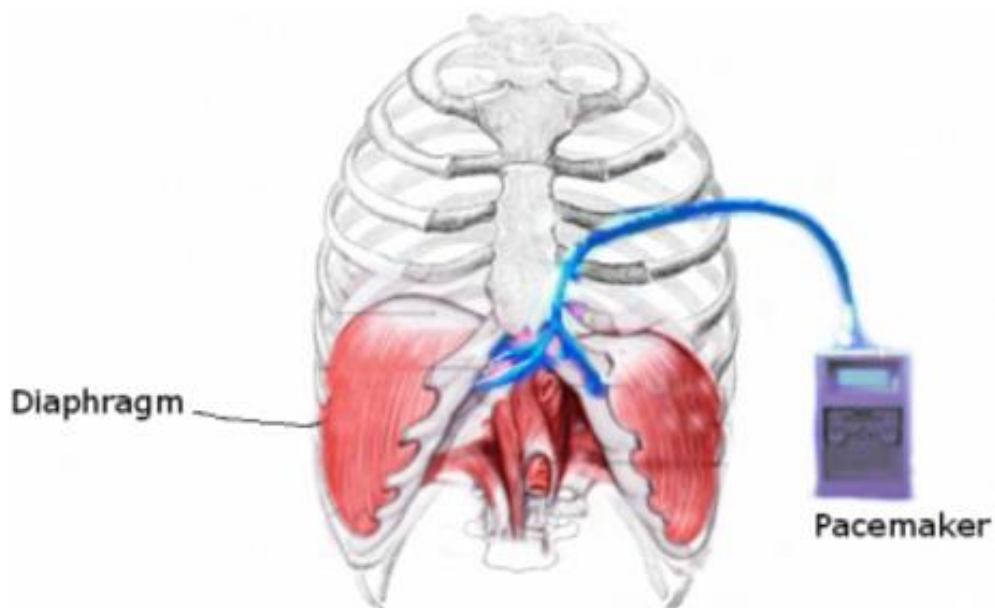
Figure 8 – An example of a diaphragm pacing device [18]

16

high risk of nerve damage from implantation, nerve deterioration associated with device implantation, high surgery costs and device costs, and exiting wires through the skin that carry risk of infection [18].

2.6   Existing Training Technologies

The main limitation of existing devices for training is that they are designed for inspiratory *or* expiratory training by loading inspiratory or expiratory muscles. This simple approach is mostly aimed to increase the strength of the associated muscles, not to restore neuro-muscular function or spinal neural circuitry. Finally, existing multi-featured training devices are not portable for use beyond the clinic.

Small, personal devices exist on the market that focus on training and are designed to specifically target and activate respiratory muscles. These devices can be used at home when users are not required to visit the clinic for respiratory muscle training (RMT). Passive (non-electric) spirometers (Figure 9a) are available for home-use and provide measurement of a user's expiratory capacity during one steady deep exhale. For

a)   b)
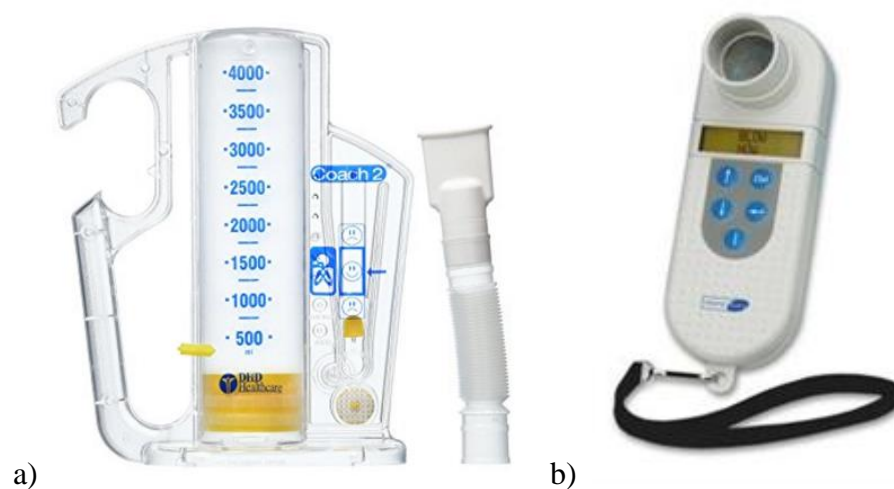
Figure 9 – Basic training devices. From left to right: a) non-electric spirometer, b) Carefusion Micro Diary Spirometer (electric spirometer) [19,20]

feedback during an exhalation, users typically see a ball or indicator rise and fall during a full exhale. Unfortunately, these devices don't show the full capacity of the lungs. It is limited to expiratory measurements, and therefore does not quantify inspiratory capacities. Measurements must be recorded manually and these devices have a price range from $8-$40 (Figure 9a) [19]. Electronic versions of spirometers can range from $120-$650 (Figure 9b) [20] who measuring capabilities are not limited to just expiratory flow.

In medical facilities, training devices are more advanced and have more features than handheld devices. With a dedicated computer system and software, these medical devices also vary in functionality and testing capabilities, sometimes requiring medical facilities to purchase two or more devices to be able to retrieve certain types of data. Price range of these systems can range from $800 - $32,000 (Table 2). For example, an Ultima CPX metabolic stress testing system (Figure 10) provides some features such as true breath-by-breath metabolic analysis for complete assessment of an individual's

Table 2 – A table of commercial respiratory monitoring devices on the market comparing their cost and testing capabilities

System	TransAir3M	Ultima Series PF	Quark PFT
Company	Morgan Sci	MGC Diagnostics	Cosmed
Estimated Cost	$18 K	~$32 K	$20K
Test Capability	Spirometry (FVC, FEV, SVC, MVV) Cough Peak Flow, and Bronchial Challenge, MEP/MIP, SNIP	Spirometry (FVC, SVC, MVV), MEP/MIP, Diffusing Capacity, Nitrogen Washout, Single Breath N2, Arterial Blood Gas Entry	Base: Spirometry (SVC, FVC, Bronchial Challenge, Bronchial Dilator) Optional: Lung Volume (Nitrogen Washout), Diffusing Capacity, MEP/MIP, Body Plethysmography, Forced Oscillation Technique, Interrupter Technique, Metabolic
Software	ComPAS	BreezeSuite	Omnia
Pneumotach	Lily type	Lily type	Lily type
Pressure	Solid state pressure transducer		

Figure 10 – Advanced training device: Medical facility device (Ultima Series PF MGC Diagnostics) [21]

functional capacity and cardiorespiratory system, an array of software for additional testing capabilities like cardiac output and spirometry, and a fully adjustable desktop for maximum testing comfort for technician and participant. All of these features and hardware factor into the devices high price tag of $32,000.

An obvious limitation to these medical facility devices is that they are not suitable to own by a typical SCI patient without incurring significant financial commitment. Additionally, due to their large, bulky size, these medical facility devices do not fit in a home environment. Not owning these devices forces patients to go to a clinic to have access to these medical-grade devices for RMT. These clinics which could possibly be a long distance away from home and not worth it for a patient, who typically only does one session of data is recorded per visit. In one session, a participant tests and reports their blood pressure, inhalation pressure, and exhalation pressure values to the doctor. Time between periodic visits to the hospital may be too broad and give an inaccurate evaluation and interpretation of the overall participant's respiratory capabilities. Regular, more

frequent RMT sessions provides more data over time and can fully describe and evaluate

a patient's respiratory capability.

2.7 Mechanical Valve Resistance Adjustment

Passive (non-electric) training devices include a mechanical valve with a contact

spring that is adjustable to modulate the threshold opening pressure. As the spring is

incrementally compressed, force on the mechanical valve increases, making it more

difficult for the valve to open. This is due to Hooke's law

$$F = -k * X$$

which describes how restoring force F scales linearly with respect to that distance the

spring is either compressed or extended. The negative sign indicates that the equilibrium

force is in the opposite direction of X.

As X decreases, the spring compresses, which increases the force on the valve. When

this compressive force increases, a greater force is required to open the one-way valve.

This larger force applied over the surface area of the valve is also known as the *cracking*

pressure at which a valve begins to open and pass fluid or air. Once flow is established

through the valve, the valve continuously remains open as long as the force from the flow

of air remains above the restorative force created by the spring. The pressure (and

therefore force) to hold the valve open can be slightly less than the cracking pressure,

similar to the differences between static and kinetic (or dynamic) friction.

III.   MATERIALS & METHODS

To meet specific aims set out by this project, hardware to develop specified device features was obtained and assembled into a working prototype. Once functionality and prototype design were established, software was developed to follow the specified training regimen. Verification and validation tests were conducted after software was transferred into the BreathForce prototype.

3.1   Device – Hardware

3.1.1   Hardware Design Criteria

The hardware design criteria (Table 3) established includes the following: 1) The device must have a breathing apparatus that can modulate between inspiratory and expiratory airflows. 2) The device must be small in comparison to advanced respiratory devices while 3) handling high data acquisition rates (100 pts/sec) with a limited power supply that 4) must last for ten 30-minute sessions before the next doctor visit or recharge. 5) Device UI must be customized to be aesthetically appealing to users based off user feedback. The device must be able to convert voltage readings from the signal conditioning circuit to cm H_2O and have 6) calibration test results with an accuracy within 5% of a verification tester device [23]. 7) Prototype value accuracy must be within +/- 1-2 cm H_2O of threshold valves [22]. These seven criteria were selected as the they fit with the specific aims of the project to make a small, more compact respiratory training device that mirrors features of larger, more advanced respiratory training devices.

Table 3 – Design Criteria - Hardware

Hardware	Design Criteria	Criteria Value
1	Breathing apparatus has inspiratory and expiratory airflows	2
2	Device must be smaller than advanced respiratory devices	70-53.3-124.5 cm
3	Device can handle high data acquisition rates	100 pts/sec
4	Power supply must last for ten 30-minute sessions	10 sessions
5	Customized UI to satisfy users	Yes
6	Calibration test accuracy within 5% of verification tester.	< 5%
7	Prototype accuracy within +/- 1-2cm H_2O of threshold valves	1-2 cm H_2O

3.1.2 Breathing apparatus

Initially, a bi-directional breathing apparatus device capable of airway resistance

adjustment was developed using off-the shelf parts. A bi-directional breathing apparatus

is important to the design to eliminate the need for two separate pressure resistance

mechanisms. Having one apparatus that can handle both inhalation and exhalation will

make it easier on the user, where typically a user had to remove their mouthpiece to

switch in between therapy devices for inhalation and exhalation.

Figure 11 – Inspiratory Muscle Trainer (IMT, left), and Positive Expiratory Pressure (PEP, right) devices [22]

After removing the mouth pieces of a regulated threshold positive expiratory pressure device (HS735-010, Philips Respironics, Andover, MA, Figure 11, Right) and a threshold inspiratory muscle trainer (HS730-010, Philips Respironics, Andover, MA, Figure 11, Left), the two parts were combined with a tee connector (Carefusion – 004051, Figure 12, Left) to form an integrated device that is manually adjustable in pressure increments that increase airway resistance measured in centimeters of water (cm-H_2O) (Figure 13).

Figure 12 - Tee connector with integrated one way valves and pressure port (Left). Disposable mouthpiece (Right)
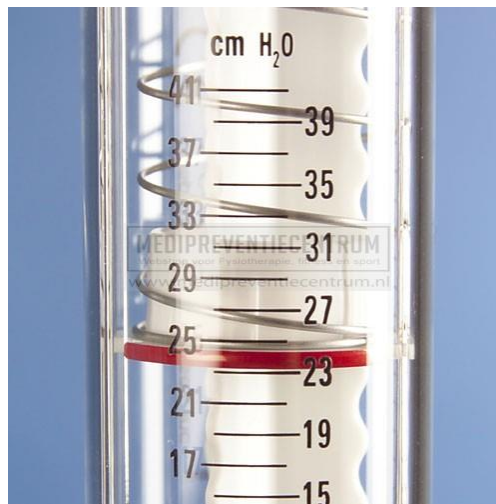
Figure 13 – Close-up of IMT device indicating gradations of cm H20 pressure values

Threshold device settings are adjusted by manual manipulation of compression springs inside the airway valves. As the springs become more compressed, higher pressure is required to crack (open) the internal pressure valves. Increased pressure setting will require increased effort by the user to open the internal valves, and this increasing difficulty represents the fundamental basis of therapy.

Figure 14 – (Top) PendoTech Pressure sensor. (Bottom) Pressure sensor and threshold devices attached to the breathing apparatus tee connector

The tee has two one-way valves on opposite sides, a built-in Luer adapter pressure port, and a flanged, disposable mouthpiece used for breathing (Figure 12, Right). The fully assembled breathing apparatus and direction of airflow while using the breathing apparatus is shown in Figure 15. As the user inhales, air passes through the one-way valves of the IMT and the tee connector. During exhalation, one-way valves block air from going through IMT and allow air flow through the other set of one-way valves through the tee connector and PEP.
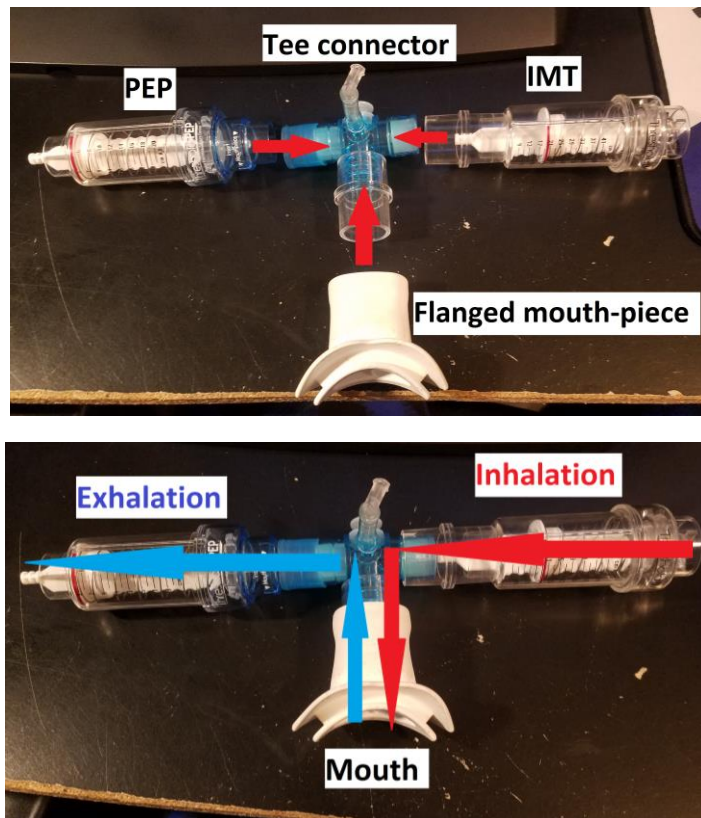
Figure 15 – (Top) Schematic of how to assemble IMT and PEP with tee and mouth-piece. (Bottom) Fully assembled breathing apparatus showing direction of air flow during device usage

3.1.3 Physiological Pressure Sensor

A disposable, sterilized/sterilizable pressure sensor with integrated female Luer connector (PRESS-S-000, PendoTECH, Princeton, NJ) was attached to the pressure port of assembled breathing apparatus (Figure 14). The sensor has a molded body with a PendoTECH High Accuracy Pressure (MEMS-HAP) pressure sensing chip affixed into a compartment in the molded body. This chip contains a micro-diaphragm that detects changes in pressure. The micro-diaphragm and fluid path are separated by a "dielectric silicon" that 1) can transmit force (pressure) from the fluid path to the micro-diaphragm while 2) provide isolation of the fluid path from the micro-diaphragm. The sensor

assesses a fluid path opening of about ~4mm diameter that runs throughout the entire molded body.

The pressure sensor operates by generating a differential voltage output that is proportional to pressure (APPENDIX IV). A 4-pin male connector circular plug (CP-7040-ND, CUI Inc., SD-40) was soldered to the ends of the sensor wires (APPENDIX I**Error! Reference source not found.**) to allow for a secure and consistent electrical connection with its 4-pin female connector mate (CP-1240-ND, CUI Inc., SD-40SN) panel mounted to the device's case that connects internally to corresponding pins on the PCB (Figure 16). This pressure sensor was chosen based on its low cost ($35.00, [23]), accuracy, and low-weight. The PendoTech family of disposable pressure sensors are qualified for use from -7 to 75 psi (-492.149 to 5273.02 cm H_2O) and have an operating temperature of 2 to 50 degrees Celsius (35.6 to 122 Fahrenheit). The pressure sensor accuracy is rated by the manufacturer at better than +/- 2% of reading in the range of 0 to 6 psi (0-421.842 cm H_2O). The accuracy rating and range of the pressure was deemed

Figure 16 – (Left) Pressure sensor connected to male 4-pin connector. (Center) Male 4-pin connector and female 4-pin connector. (Right) Breathing apparatus (male) connected to BreathForce processing unit (female)

acceptable as the average lung pressure that a typical person can exert is 1.4 psi (98.4 cm H_2O) with an upper end of 2.8 psi (196.8 cm H_2O) that can be exerted without injury for brief periods of time [23].

The sensor is biocompatible and meets USP Class VI requirements, a standard set by United States Pharmacopoeia (USP) to safeguard the quality of medicines and other health care technologies. Passing class VI requirements means the sensor exhibits very low levels of toxicity and is more likely to produce acceptable biocompatibility results. Finally, IMT and PEP devices, three-way connector mouthpiece, and PendoTECH Pressure Sensor are all individually FDA approved devices, and therefore acceptable for clinical use.

3.1.4   Pressure Sensor Signal Conditioning Circuit

A custom instrumentation circuit was developed to amplify and filter the signal generated by the pressure sensor. An INA122UA instrumentation amplifier (Texas Instruments, INA122UA/2K5) with a ground resistor of 402-ohms provided a gain of 500 (Figure 17). A high gain of 500 was selected to amplify signal from the pressure sensor so that measurements could be analyzed accurately. This particular op-amp IC was chosen because it was designed to operate on a single power supply (Voltage supply range of 2.2-36V), and was therefore suitable for battery operation. Since the amplified voltage (ADC0 of Figure 17) would ultimately be read by a battery-operated microcontroller (MCU) with a measurement input range of only 0-5 Volts, a 2.5 V low-impedance output voltage reference was created using an OPA237UA op-amp (Texas Instruments, OPA237UA/2K5) configured as a voltage-follower (Figure 18 – Voltage follower configuration to provide a low-impedance, 2.5V reference voltage to the

27

instrumentation amplifier in Figure 17Figure 18). A schematic of the complete circuit

(APPENDIX II) was created in Multisim (National Instruments, Austin, TX, V14.1) and

simulated to confirm correct operation of the design, as well as to establish connectivity

of the headers of the pressure sensor to the MCU described in the following section.
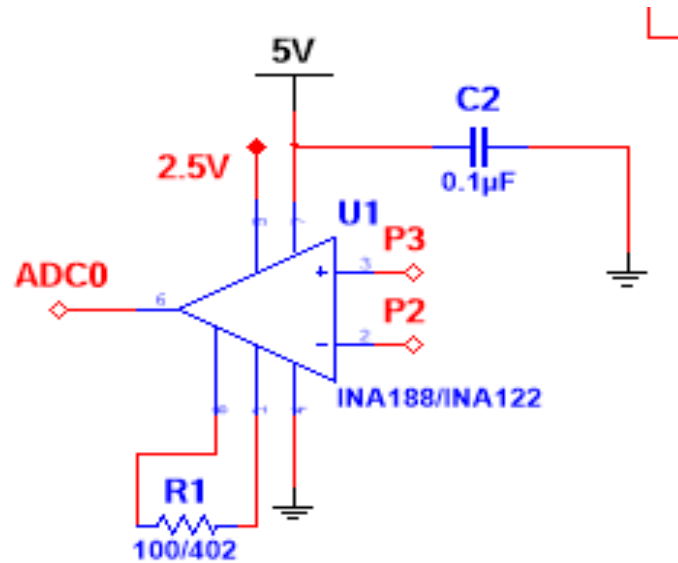
Figure 17 – Differential Op-Amp configuration, where P2 and P3 are inputs from
the pressure sensor into the INA122UA instrumentation amplifier. A single
resistor, R1, sets the gain of the device.
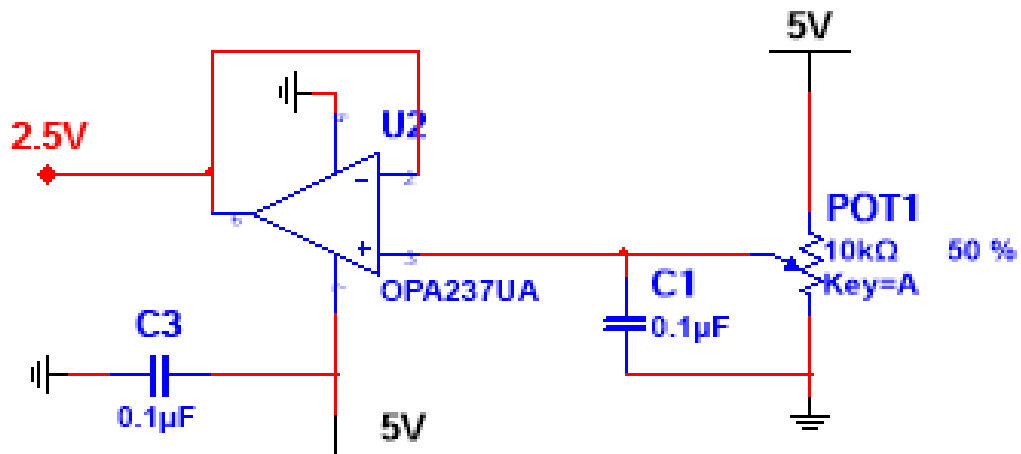
Figure 18 – Voltage follower configuration to provide a low-impedance, 2.5V
reference voltage to the instrumentation amplifier in Figure 17

3.1.5   Printed Circuit Board Design and Fabrication

The circuit schematic created in Multisim established the physical connections of
components and nodes of a custom printed circuit board (PCB) (Figure 19). Ultiboard
(National Instruments, Austin, TX, V14.1) was used to position all parts and connectors
on a two-layer, standard FR4 circuit "shield", which is a name given to PCB's that were
designed specifically to plug easily into an Arduino platform. All op-amps, resistors, and
capacitors were surface mounted onto the fabricated PCB according to the schematic
layout (Figure 20). Stackable headers were soldered to the custom board to allow other
shield devices to be stacked on top of the custom board and connect to the pins of the
MCU. A four-pin male header with friction lock (Molex, 0470533000, WM9797-ND)
was soldered on to the custom board and was mated to the 4-pin female connector
(Molex, WM3285-ND, 0470541000) that was soldered to the attached panel-mounted
female plug connector for the pressure sensor connection shown previously in Figure 16
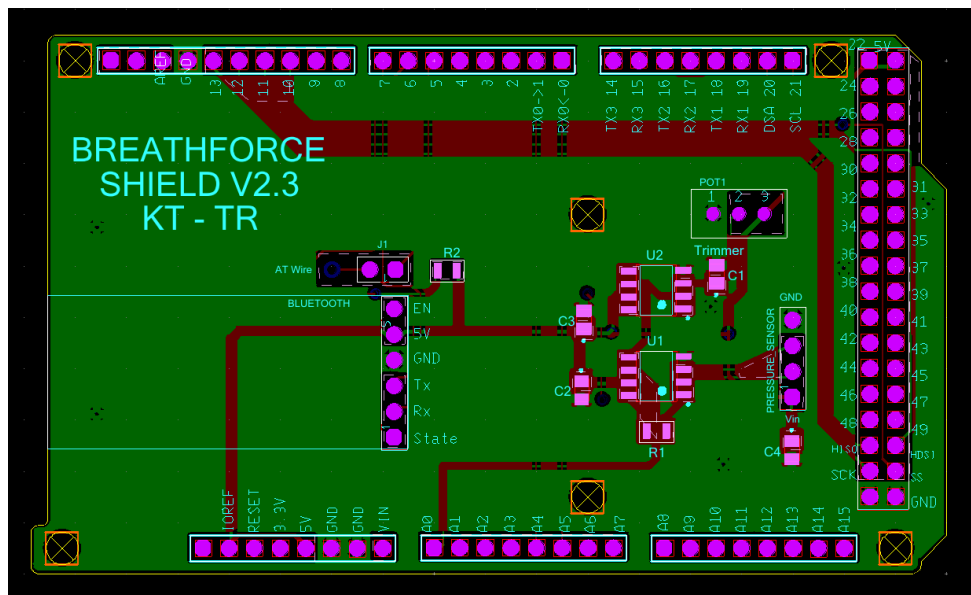
Figure 19 – Ultiboard schematic showing placement of all electrical components

(Figure 21). A Bluetooth module (HC-05, NewZoll) was also soldered onto the board for future feature expansions of the system (wireless operation).
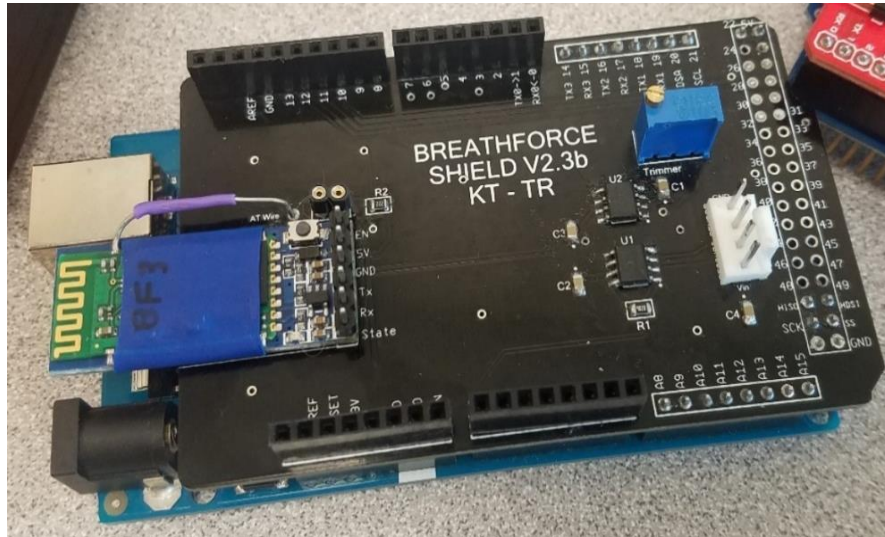
Figure 20 – Image of populated printed circuit board
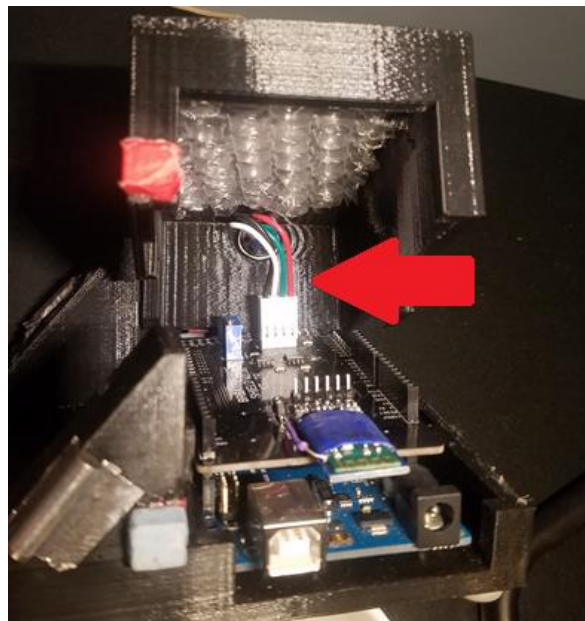
Figure 21 – Connection of panel mounted plug connector of breathing apparatus to the PCB

3.1.6 Data logging and SD card reader

An Arduino compatible data logging shield module with an integrated SD card slot (product 1141, www.adafruit.com, Figure 22) was included to log therapy session data. This shield contains a real-time clock (RTC) that was used to time-stamp data. Time-stamping data allows confirmation of individual data points in the plot of pressure
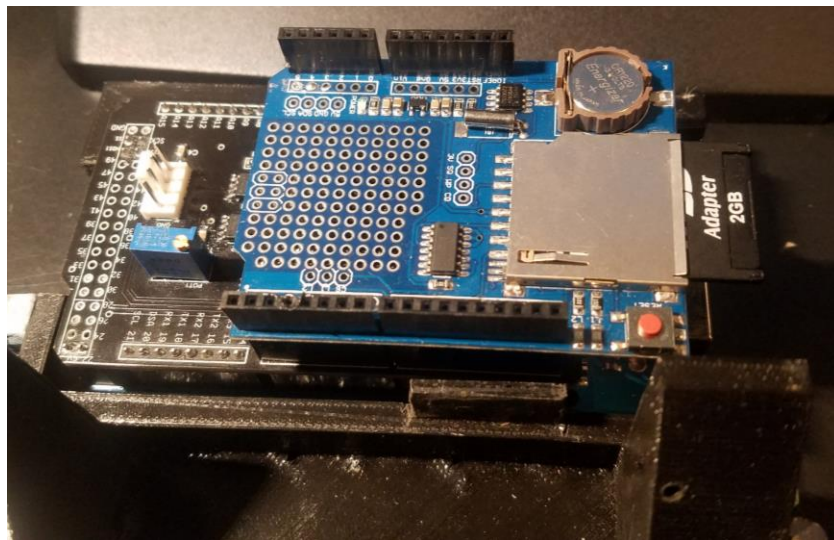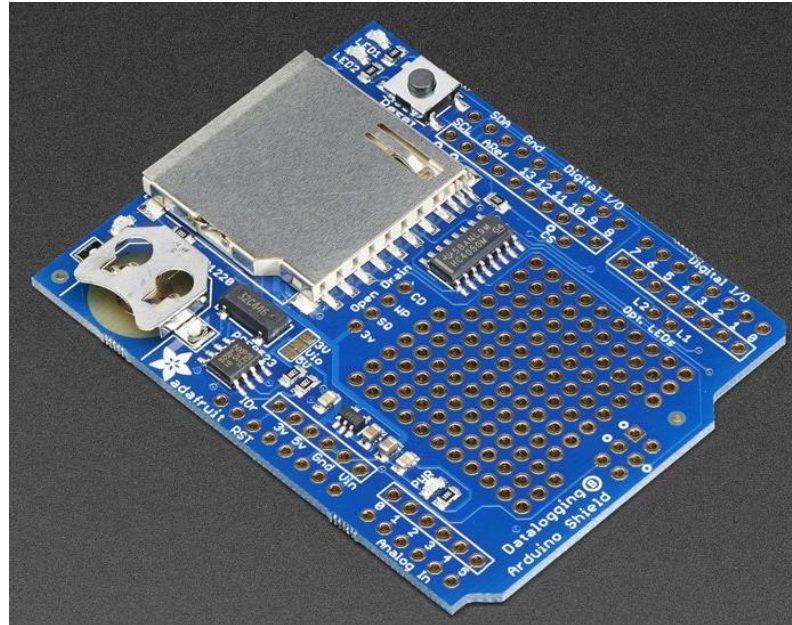
Figure 22 – (Top) Adafruit Industries SD card datalogger shield [24]. (Bottom) Datalogger shield stacked on top of the PCB.

measurements to track how long device-users are doing RMT and to calculate how many measurements the device is saving per second. Additionally, this index also allows clinicians to see breathing patterns over a duration of time. Clinicians can see breathing patterns graphically, such as a sharp, short, high-pressured breath or a long, flat-line, low-pressured sustained breath, and be able to interpret the respiratory capabilities of the user (similar to how flow-volume curves are analyzed). Stackable shield compatibility allows the SD card shield to be placed on top of the same headers of the custom shield and connect to the MEGA (Figure 22). This allows the device to remain more compact without any unnecessary external wiring.

3.1.7   Touchscreen Display

A 3.2-inch (diagonal) display (uLCD-32PTU-AR, 4D Systems, Minchinbury, NSW, Figure 23) with integrated, resistive touch input and display resolution of 240 x 320 was used to display data and provide touch control and visual feedback for the BreathForce prototype device. The touchscreen provided a high-resolution interface and simple method of interaction with the device. The development kit provided with the display a
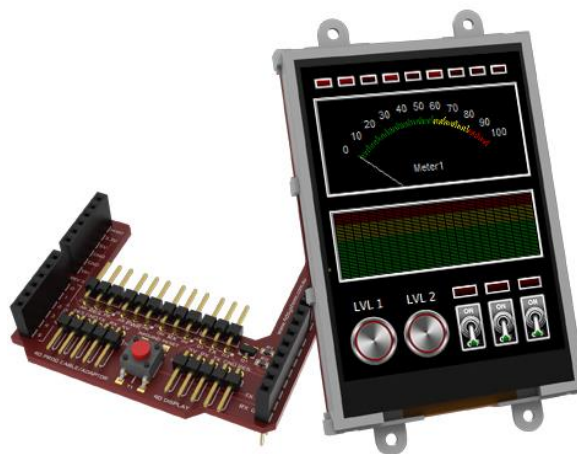
Figure 23 – 4D Systems Touchscreen Display [25]

plug-in shield (red board, Figure 24) to connect directly to the MEGA and a five-pin wire

connection from the shield to the display (Figure 25).
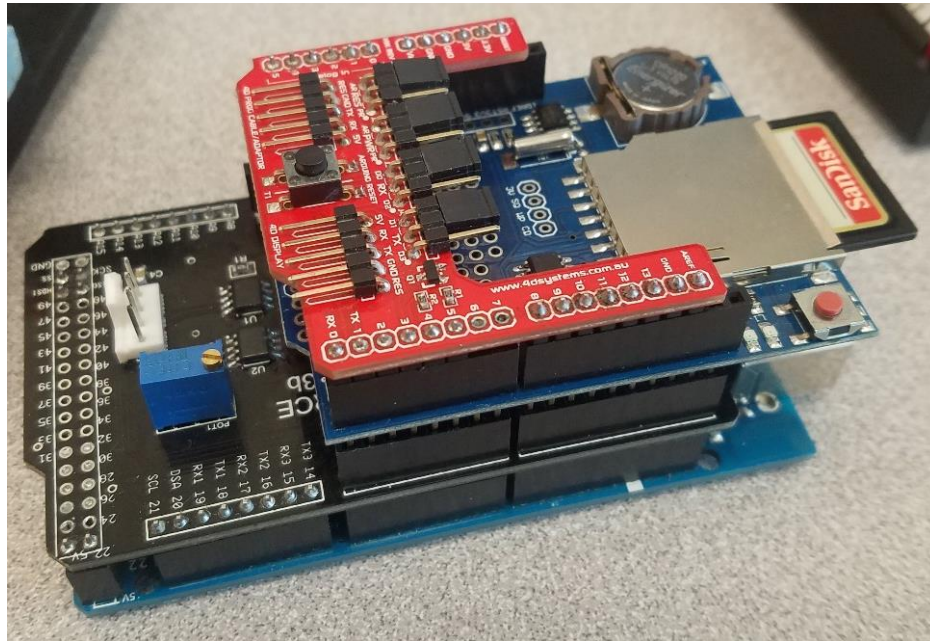
Figure 24 – 4D touchscreen display shield (red) stacked on top of the Arduino
MEGA
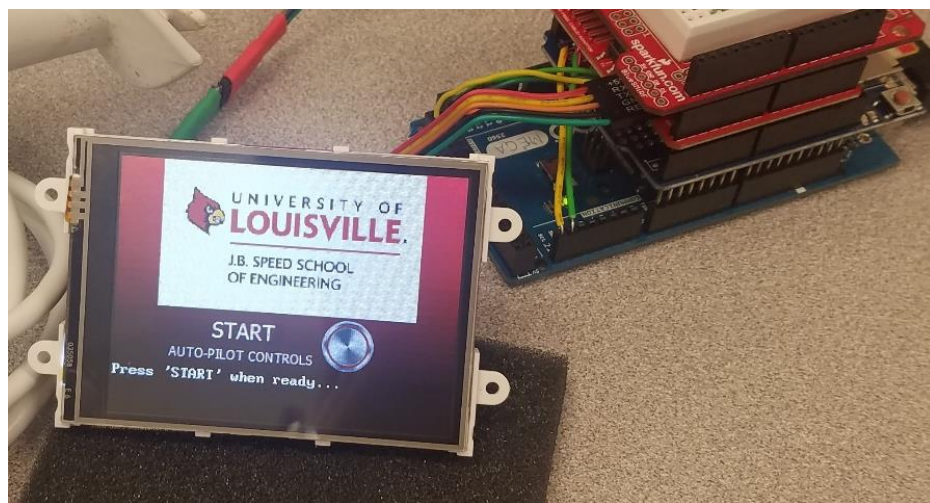
Figure 25 – 4D Systems touchscreen display connected to the plug-in shield,
which is stacked with other devices connected to the Arduino MEGA

3.1.8 Microcontroller

In order to measure the output from the pressure sensor and provide a method to scale and use these values in a therapeutic session, an Arduino MEGA (2560 REV3, Arduino.cc, Figure 26) was used as the microcontroller (MCU) for the device (herein referred to simply as the MEGA). This embedded development platform is based on the ATmega2560 microprocessor (Atmel) that provides 54 digital input/output pins, 16 analog inputs, 256 KB of flash memory, 8 KB of SRAM, 4 KB EEPROM, and a clock speed of 16 MHz. Bench tests on initial prototypes with the more common Arduino UNO (Rev 3, Arduino.cc, Figure 27), revealed that a MCU with more program memory space
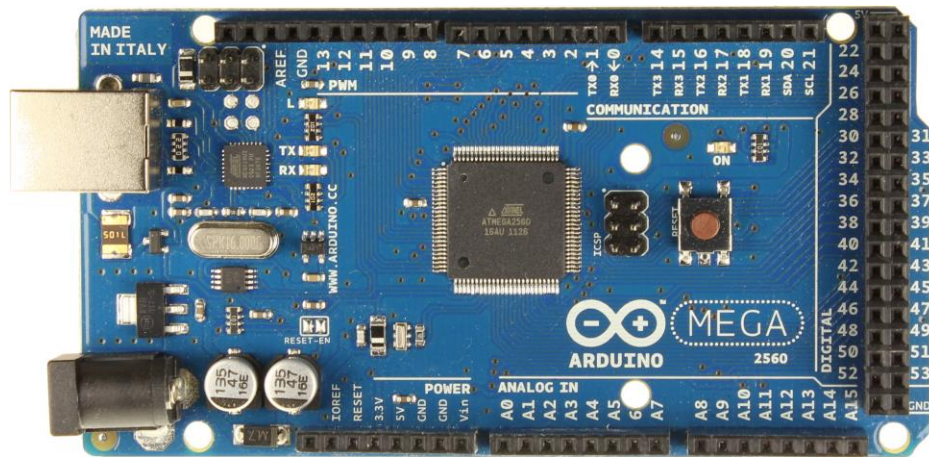
Figure 26 – Arduino MEGA 2560 microcontroller development platform [26]
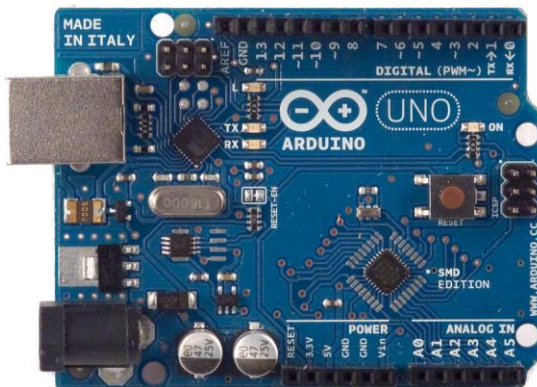
Figure 27 – Arduino UNO microcontroller development platform [26]

would be required to support all the features of the proposed device. An insufficient amount of flash (storage) memory (32 KB) resulted in low acquisition rates of data saved onto the SD card (6 samples per second) and visibly slowed refresh rates of the external display. The MEGA, on the other hand, provided affluent memory to store and run all the necessary function libraries (SD, DS1307RTC, genieArduino) required to handle all the features of the device. Bench tests using the MEGA showed increased refresh rates on the external display with no visible lag and a significant increase in data acquisition (22 samples per second). Table 4 shows a side-by-side comparison of selected specifications of the UNO and MEGA versions of the Arduino platform.

Table 4 – Comparison of tech specs of the Arduino UNO REV 3 vs
Arduino MEGA 2560 REV3. [26]

Feature	UNO	MEGA 2560
Microcontroller	ATmega328	AVR ATmega2560
Operating/Input Voltage	5V / 7-12V	5V / 7-12V
Digital I/O Pins	14	54
Analog Input Pins	6	16
Flash Memory	32 KB	256 KB
SRAM	2 KB	8 KB
EEPROM	1 KB	4 KB

3.1.9   Custom Device Enclosure

A custom 3D printed enclosure was developed in SolidWorks (V17, Dassault Systèmes, Waltham, MA) to enclose the electronics and touchscreen display (Figure 28). An opening on the right side of the case allowed for retrieval and insertion of the SD
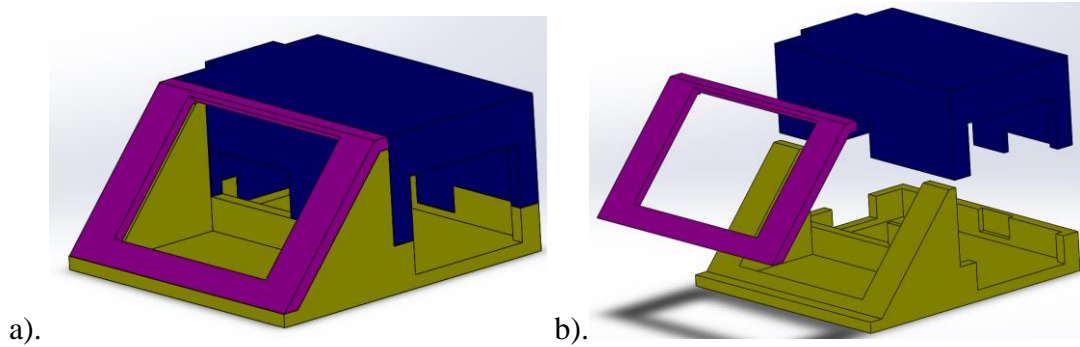
Figure 28 – a) 3D rendering of case designed in Solidworks. b) Exploded view shows case consists of three separate parts

card. The same opening also allowed access to power input and USB programming ports of the MEGA. A hole on the left side of the case was designed to hold the 4-pin female panel mount connector and provide stable connection with the outside 4-pin male plug connector of the breathing apparatus. A window opening was designed to hide mounting hardware on the left and right edges of the display while allowing users access to the touch screen. Schematics of case parts are in APPENDIX III.

Magnets were used to hold the parts together (Figure 29). The individual case parts were 3D printed in a Makerbot Replicator (makerbot.com) with ABS plastic. Printer settings were set to a 30% infill density, layer height of 0.2mm, two shells, and a
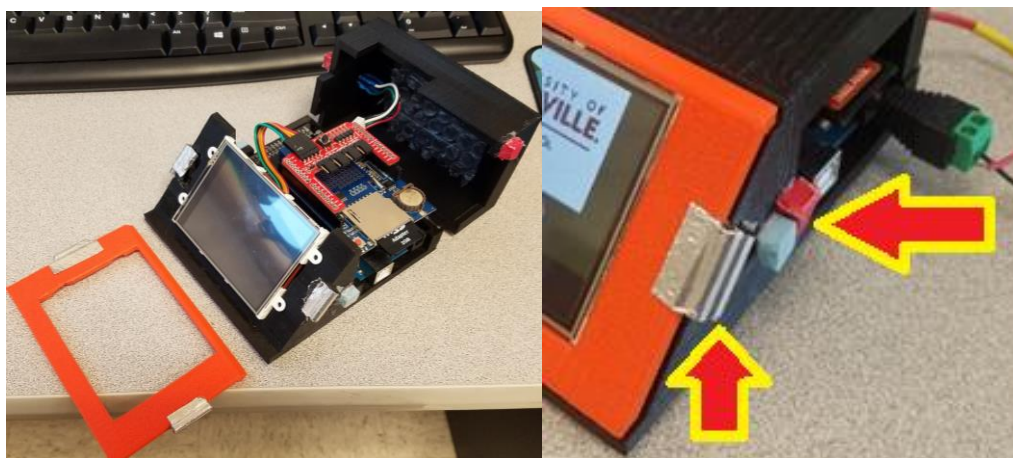
Figure 29 – Arrows indicating where magnets are located that hold the parts together

diamond fill infill pattern. These settings were selected as a balance between fast printing times and a rigid, durable case.

3.1.10 Power Supply

Two 3.7-volt rechargeable Li-ion batteries (BRC 18650 type, 3000mAh, EBLmall.com, Los Angeles, CA) were used to power the device and to offer portability to the system (Figure 30). The two batteries were inserted into a battery holder that would be Velcroed to the back of the device enclosure. A 2.1mm center-positive plug was attached to the wires of the battery holder to plug into the MEGA to power the device.
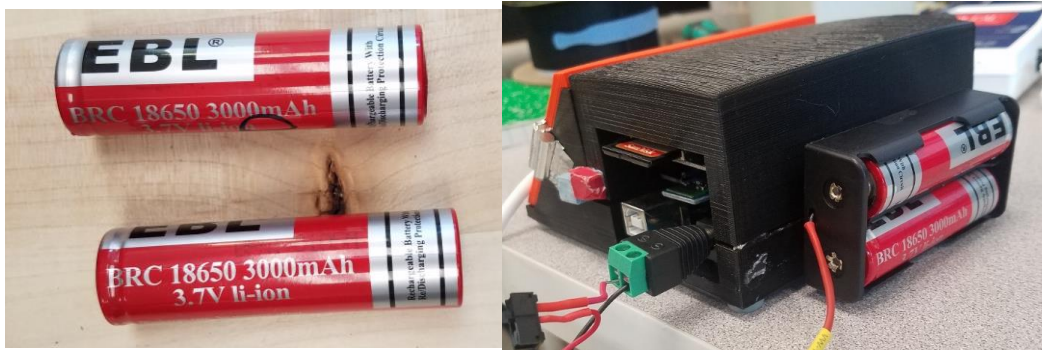
Figure 30 – (Top) Two BRC 18650 3000mAh 3.7V rechargeable Li-ion batteries. (Bottom) Batteries in holder attached to the back of the enclosure, plugged into the MEGA

By providing a total voltage of 7.4V, two cells were appropriate to power the device. The Arduino MEGA requires a voltage supply from 7 to 12 volts and regulates this supply voltage down to 5 volts via an onboard linear voltage regulator. With the MEGA using 14 I/O pins with each pin using 40 mA [26], it is calculated to take 10.7 hours to drain both batteries. If the device-user is assigned to do 30-minute training sessions by the doctors, two cells are expected to last the around 21 sessions before the batteries must be recharged.

Dimensions of the fully assembled (W-D-H) measure out to be 115-120.75-53.37 mm which is a lot smaller compared to advanced systems, like Ultima (70-53.3-124.5 cm, [21]). After the hardware for BreathForce was fully assembled (Figure 31), the system software developed to measure pressure, perform MEP/MIP evaluation, and time and record data during a therapy session.

Figure 31 – Fully assembled device of printed parts enclosing device hardware

3.2   Device – Software

3.2.1   Software Criteria for Design and Functionality

For software, design criteria is if the protocol for training participant and clinician are followed.

3.2.2   Coding the Touchscreen Display

The BreathForce prototype required two programming environments to program the system to enable therapy: the 4D Workshop IDE and Arduino IDE. The 4D Workshop IDE (**I**ntegrated **D**evelopment **E**nvironment) (v4.5.0.17) was used to program the

touchscreen display with separate pages of controls and indicators, while the Arduino

IDE (v1.8.2) was used to code the operational control program on the MEGA.

The touchscreen input features were designed and programmed in the 4D Workshop

IDE. The touchscreen device was programmed to have user interface (UI) "pages" to

increase working space for multiple user inputs and extend the touchscreen functionality

beyond a single, static UI. Virtual buttons were programmed onto these pages to provide

user input and report these activities to the code running on the MEGA. These messages

are commands that request the MEGA to perform certain functions of the device such as
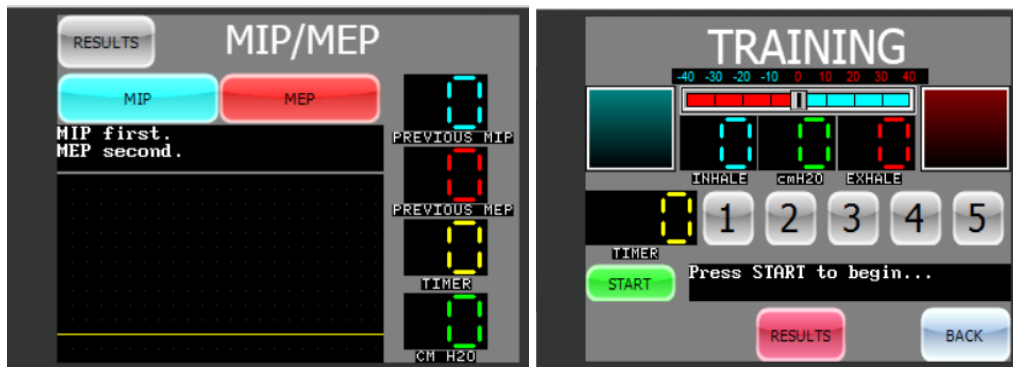
Figure 32 – Pages of BreathForce showcasing virtual buttons, LEDs, level
gauges and trace plots

running a particular test, saving data to the SD card, or performing a calibration of the

device. Virtual level gauges and data trace plots were programmed onto pages of the

display to receive bytes of pressure and operational data from the Arduino MEGA and

report them on the screen (Figure 32).

While performing subject testing, test subjects will be asked for their opinion about

the UI of the device. They will be asked what they like and don't like about the current

UI and what graphical display do they prefer on the UI. Once feedback is received, the UI

can be re-designed in the 4D Workshop IDE to satisfy user demands.

3.2.3 Coding the Microcontroller

The Arduino IDE programming environment was used to develop code to measure

pressure inside the breathing device, perform calculations to convert pressure to correct

units, and control both the SD card shield and the touchscreen display. The Arduino IDE

contains several built-in software libraries to provide access to microcontroller functions

used to operate the device's features. The analog input voltage from the pressure sensor

was converted from PSI to cm H_2O using a software algorithm. Additional software

would tell the microcontroller to display the converted cm H_2O to the 4D display and

save to the SD card. Finally, the software included a function to wait for messages from

button inputs of the 4D display to initiate programmed functions.

3.3 Device Verification and Validation

3.3.1 Pressure Sensor Measurement Validation

To ensure that the device was measuring pressure accurately, results displayed on

the prototype were tested to match reported pressures of a standardized pressure

calibrator. Independently, PEP and IMT Threshold devices have an accuracy of +/- 1 cm

H2O and +/- 2 cm H_2O, respectively, with reproducibility of +/- 0.5 cm H_2O and +/- 1 cm

H_2O, respectively [26]. Pressure measurements using the prototype were compared to a

Delta-Cal Transducer Simulator and Tester (Utah Medical Products, Salt Lake City, UT).

The Delta-Cal tester was connected directly to the PendoTECH pressure sensor (Figure

33) and pressure adjusted to compare with the output and conversion of pressure in the

41
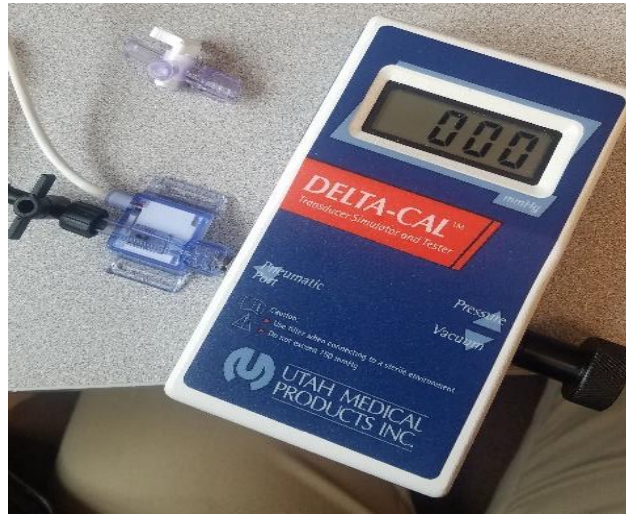
Figure 33 – Delta-Cal Transducer Tester connected to pressure sensor

BreathForce prototype. Although Delta-Cal reports in mmHg and the BreathForce reports in cm H_2O, a conversion factor from cm H_2O to mmHg was used to compare the results.

3.3.2   Humidity Test

A humidity test was performed on the pressure sensor to investigate for effects of increased accumulation of moisture on the pressure measurements during a therapy session. While operating the breathing apparatus for and extended amount of time, there is a noticeable amount of condensation in the breathing apparatus (Figure 34). It is important to account for variations in pressure measurements if the condensation/moisture built up in the apparatus will affect the pressure sensor and pressure measurements. Future designs will incorporate a disposable inline filter between the mouthpiece and pressure sensor to minimize accumulation of moisture inside the air passages.

The humidity tests were similar to the pressure range verification method, except tests were performed after inhaling and exhaling into the breathing apparatus connected

42
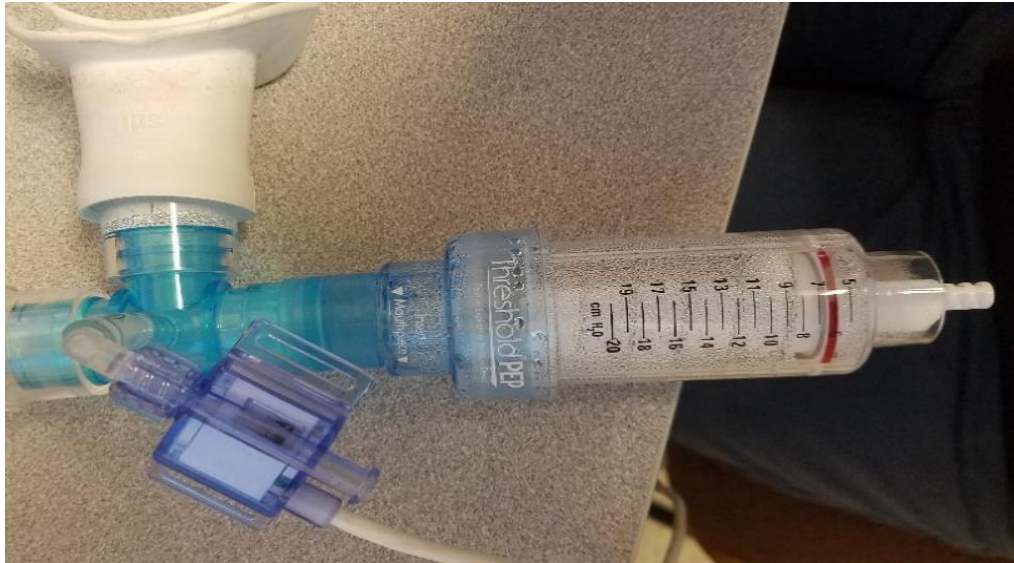
Figure 34 – Threshold device with condensation

to the pressure sensor for one minute before disconnecting the sensor and connecting it to

the Delta-Cal. This ensured the presence of moisture/humidity in the sensor prior to each

measurement. Humidity test values are expected to not to show any variability compared

to the pressure measurement validation test as the pressure sensor was manufactured with

a di-electric silicon barrier to isolate the micro-diaphragm of the sensor from moisture

effects.

3.3.3   Moisture Evaluation

While the breathing apparatus is designed to be reusable, it will still need to be

washed/rinsed before each use. Because water will come in direct contact with the sensor

during cleaning, a moisture test was performed on the prototype for reasons similar to the

humidity test. The moisture test would provide data to indicate that the device can be

used immediately after cleaning or that the device must be dried before being used after

cleaning. Moisture tests were similar to the humidity test except instead of blowing into

the pressure sensor, water was run through the sensor to simulate the cleaning step.

Results are expected to behave similarly to the humidity test results: showing no change in data acquisition behaviors due to how the pressure was manufactured.

3.3.4 Temperature Evaluation

Because the device is portable, it is important to consider effects of temperature variations while using the pressure sensor. The sensor datasheet includes information that reports the sensor has a safe operating temperature range of 35.6 to 122 F. It is important to verify that ambient or temperature changes during use will not affect the accuracy of the sensor. As a user breathes into the device, heat from breathing warms up the sensor. It must be verified that breathing warm air into a cold sensor does not produce drift in measurements versus an already warmed up sensor. Temperature tests were done by initially placing the temperature sensor in a cold environment (13.2 ⁰F) for one hour, followed by a pressure range calibration using the Delta-Cal. Calibration readings were taken from the sensor every five minutes as it warmed to room temperature.

3.3.5 User Data Evaluation

Two subjects, one with SCI and one without SCI, will be tasked to operate the BreathForce prototype. User results are saved onto the BreathForce prototype and will be evaluated to see if device is operating as intended, device is measuring pressure accurately, and if interpretation of data represents the health condition of the user.

3.3.6 Test of Manual Valves for Cracking Pressure

Test of manual valves vs cracking pressure was conducted to directly compare the pressure labels on the threshold valves to measured values using BreathForce. In increments of 5 cm H_2O on both inhalation and exhalation threshold valves, a control subject inhaled and exhaled into the breathing apparatus until the respective threshold

valve opened. The purpose of this test was to see if the BreathForce displayed the same values as the threshold devices at the moment of valves cracking.

3.4   Device Operational Features

Two primary modes were included in the control program: MEP/MIP mode and Training mode.

3.4.1   MEP/MIP Evaluation

The MEP (Max Expiratory Pressure)/MIP (Max Inspiratory Pressure) mode was developed to record the max pressures of inhalation and exhalation that a patient can produce. Traditionally, MEP and MIP are some of the parameters that quantify respiratory motor function and are one of the easiest measurements to track for respiratory motor improvement [6]. It is also one of the common measurement features listed in the table of expensive devices listed in Table 2.

For this mode, the breathing apparatus is adjusted to accommodate a one-way adjustable valve attachment (Figure 35) that was attached to an opening on either side of the tee connector that would restrict air flow and facilitate airway pressure collection.

Figure 35 – Attachment to replace threshold valves for MEP/MIP mode

Which opening the apparatus was attached to depended on which direction of airflow was being examined. If MIP was being calculated, the valve attachment was attached to the inspiratory side of the tee restricting air flow entering the apparatus. If MEP was being calculated, the valve attachment was attached to the expiratory side of the tee restricting air flow exiting the apparatus. Once MEP and MIP are obtained and calculated, training target values would be calculated from the MEP and MIP values and used in Respiratory Muscle Training.

3.4.2   Respiratory Muscle Training

In Training Mode, the user inhales and exhales through the breathing apparatus with the pressure threshold devices attached to the breathing apparatus. In this mode, the device functions like a spirometer. Expiratory and inspiratory pressure threshold devices are manually adjusted to match training target values determined from MEP/MIP mode, respectively. While performing RMT, users inhale and exhale through the breathing apparatus, cracking the one-way flow valves, and work to maintain set target values. The user interface provides visual feedback to the user during training mode and will light on-screen LEDs if the user is reaching the training target values.

3.5   Device Operating Procedure

When initially powering on the device, the display will start on the "Starting" page (Figure 36). On this page, the device will run checks on the status of the prototype. It will verify that the RTC is running and if an SD card is inserted into the SD slot. Once those checks pass, the user is notified to press "Start" which will move the user to the "Directory" page (Figure 36).

46
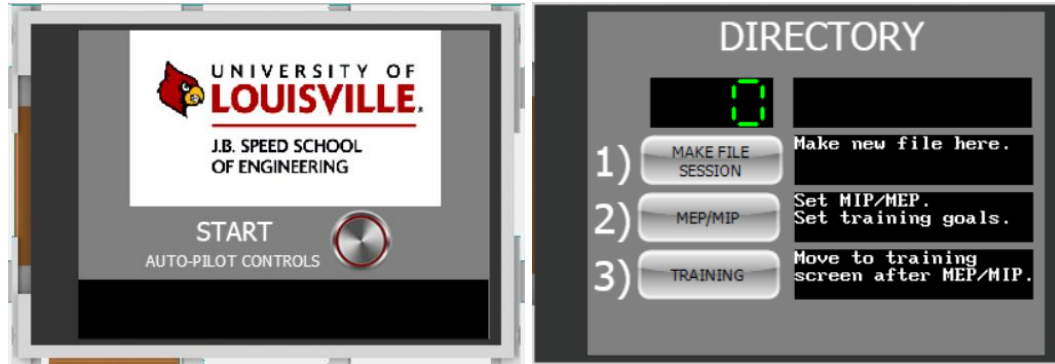
Figure 36 – "Starting" page and "Directory" page of display

On the "Directory" page, the device runs an initial calibration step to zero any initial pressure sensor offset which accounts for the current location of the device and environmental effects on the sensor. After the calibration step, the user will press "MAKE FILE SESSION" on the "Directory" page to create a new file for the current session. The MEGA performs a file-check on file names that exist on the SD card. If the latest number in the file name does not exist (i.e. Name07 exists but Name08 does not exist), a new file will be created with the latest number (i.e. Name08).

After the new data file is generated, the user will press "MEP/MIP" to go into MEP/MIP mode. The user will be taken to a "Training % Setting" page where the user or clinician will set a training percent level for the session (Figure 37). This number will be multiplied to the MEP/MIP at the end of the test and will set the training pressure value for RMT. The page also displays the "Previous % Setting" from the previous session for user comparison to the session about to be done. The user may skip picking a training percent setting if the previous percent setting satisfies the user. By skipping this page, the

device will automatically set the previous percent setting to the current setting. After

pressing "NEXT", the user will be sent to the MIP/MEP page.
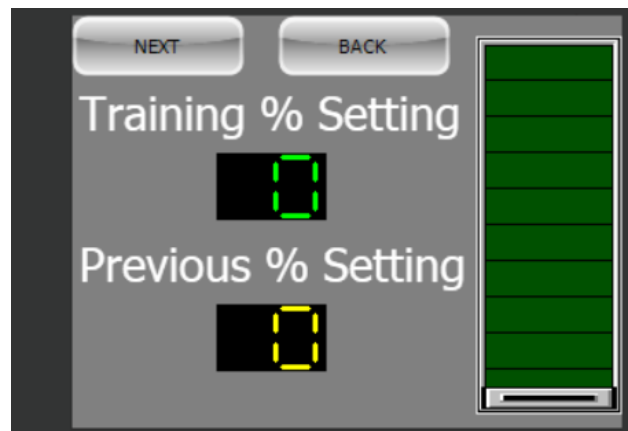
Figure 37 – Touchscreen display of "Training % Setting" page that sets the training percent for the session
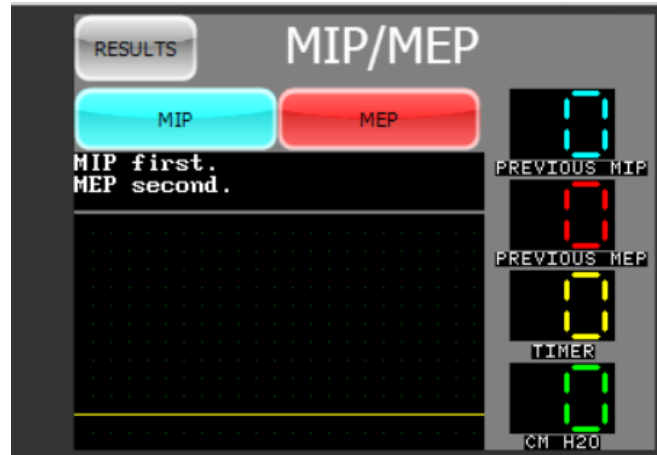
Figure 38 – "MEP/MIP mode" page to conduct MEP/MIP test
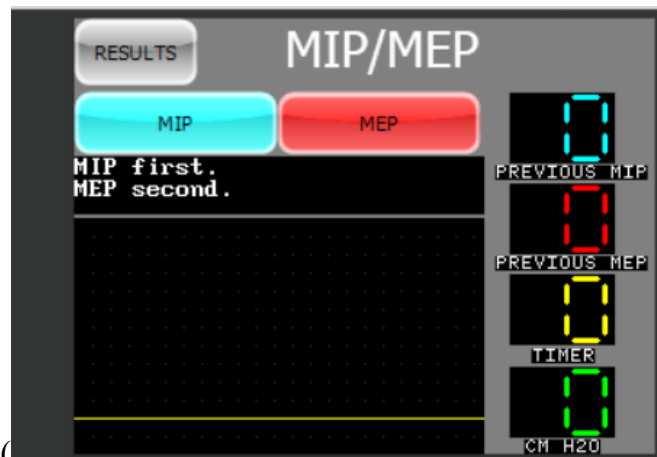
On the "MEP/MIP" page (

Figure 38), the user will conduct the MEP/MIP test to find MEP and MIP. At a

clinic, the clinician will prep the breathing apparatus with the one-way adjustable valve

attachment for this test and instruct the device-user on how to inhale for the test. The

clinician or user will press "MIP" and follow the devices instruction on when to inhale.

Once inhalation and the test is complete, the clinician will switch the one-way adjustable

valve attachment and instruct the user on how to exhale through the apparatus. Either the

user or clinician can press "MEP" to begin the test.

Once both tests are completed, the user or clinician can press "RESULTS" to be

taken to the "RESULTS" page of the MEP/MIP test (Figure 39). On this page it will

49

display MEP and MIP values from the test and the calculated training values for RMT

from the training % setting.

Once Training Values have been confirmed, the user will press "MAIN

DIRECTORY" to be taken back to the directory page. On the next page, the user will

Figure 39 – "Results" page of MEP/MIP mode that will display Training EP
and IP for RMT
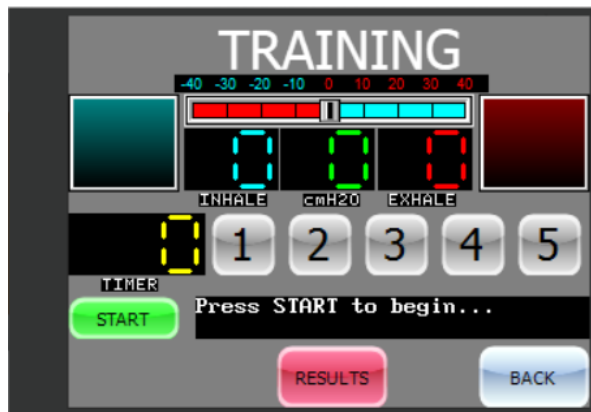
Figure 40 – Touchscreen display of Respiratory Muscle Training Mode

press "TRAINING" to be taken to the training page for RMT (Figure 40). Training

values (blue and red) are displayed next to the LED indicators. The clinician or user will

change settings on the threshold pressure devices corresponding to inhale and exhale

directions to match the training values. The user will then detach the one-way adjustable

valve and attach both threshold pressure devices. LED indicators will light up if the user's current cm H_2O reading (green number) reaches 90% of the training value. A sliding gauge provides an additional visual feedback for the user during RMT. Before pressing "START", the user or clinician will decide on the duration of RMT ranging from 1-5 minutes and press the corresponding button. Once the timer is set, the user can press "START" to begin RMT. Once RMT is over, the device will give the option for the user to repeat the training, repeat the training with more time, or view the session results page (Figure 41) by pressing "RESULTS."

On the "SESSIONS RESULTS" page (Figure 41), the device will display the file name that was just created and display all relevant information pertaining to the RMT that was just completed and saved onto the SD card. The device will also display information from the previous session by opening an older file to show the user's
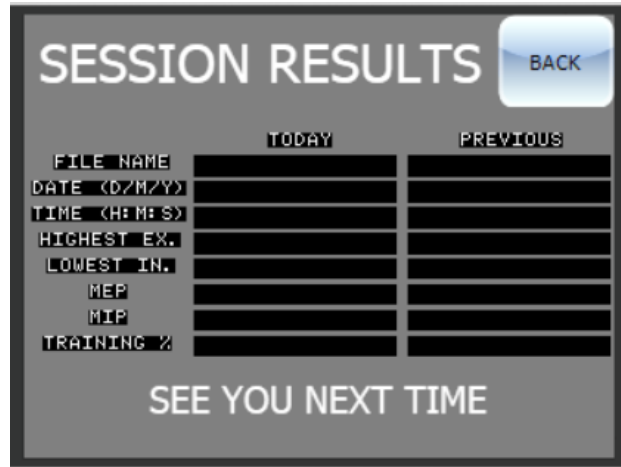
Figure 41 – Results page of RMT software

progress from the last session. The "SESSION RESULTS" page signals the user they have completed using the device for MEP/MIP tests and RMT and can be stored away until the next training session.

51

3.6   Breathing Apparatus Cleaning and Replacement

The breathing apparatus was designed to be disassembled easily for cleaning (Figure 42, Left). The device should be cleaned often to avoid contamination and to extend the lifetime of the mouthpiece portion of the breathing apparatus. Over time, constant use of the breathing apparatus will result in buildup of dead skin cells and saliva residue on the mouthpiece (Figure 42, right). When device-users return to the clinic for check-ups, clinicians will examine the state of the breathing apparatus to determine if any parts need replacement. The estimated life span of the disposable device components is around one year before a replacement part is issued. Even if there are no faulty parts, replacement of an old part may be performed to ensure accurate measurement.

Figure 42 – Disassembled breathing apparatus for easy cleaning (left). Dead skin cells and mouth build up on mouthpiece compared to a brand new clean mouthpiece (right).

3.7   Clinician Protocols

The following outline describes the pre-session clincan protocol required to prepare for a respiratory training session. The software was designed to follow this protocol as closely as possible.

Initial set-up:

- At the start of the test, clinicians will construct the breathing apparatus with IMT, PEP, pressure sensor and flanged mouthpiece and connect it to the MCU.

- The clinician will inspect the breathing apparatus to ensure that there are no leaks between the pressure sensor and the tee connector and between the tee connector and threshold valves before powering on the device.

Session set-up:

- When the device is turned on, the user and clinician will wait for the system check tests to pass before pressing START. When completed, press START to proceed to the DIRECTORY page.

- Wait for the calibration step on DIRECTORY page before pressing MAKE NEW FILE to make a new file for the training session about to be done.

- Once the file has been made, press the MEP/MIP icon to go to the % TRAINING page to setup MEP/MIP.

MEP/MIP Testing:

- When on % TRAINING page, the clinician will discuss with the participant on a percentage value to be multiplied to the MEP/MIP values and set the scale to that agreed percentage. Press NEXT to go to the MEP/MIP page.

- The clinician will attach the valve attachment, specific for MEP/MIP test, on the inspiratory path of the tee connector and instruct the participant on how to inhale very strongly for five (5) seconds through the device.

- Once the participant understands the procedure, the clinician will press MIP and the participant will inhale when instructed to by the device. After MIP is done, the clinician will change the valve attachment to the expiratory path and instruct the

participant how to exhale strongly for five (5) seconds. Once the subject understands, the clinician will press MEP and the subject will exhale strongly when instructed by the machine.

- Once MEP/MIP tests have been completed, the clinician will press RESULTS to see what values to adjust the threshold valves to based on the result screen displays. IMT valve will be adjusted to the number in the Target Insp box and PEP valve will be adjusted to the number in the Target Exp box.

Training mode:

- After the results have been confirmed, the clinician press MAIN DIRECTORY button to go back to the DIRECTORY page and press then press TRAINING for training mode.

- The clinician will remove the MEP/MIP valve attachment and attach the level-adjusted PEP and IMT threshold valves to the respective locations on the tee connector.

- On the TRAINING screen, the clinician and participant will agree on a duration of time to conduct the training and will press the corresponding number on the screen of the device. Once the time has been set, the clinician will press START and allow the participant to perform respiratory cycles to try and match the target values set by MEP/MIP tests and light up corresponding on-screen LEDs. Clinician needs to keep an active watch on the participant to ensure that they are performing full, complete respiratory cycles and giving 100% effort.

- After time has expired, the clinician will press RESULTS to display the results of the just conducted training session. Training is then concluded once the results have been confirmed and device can be powered down to be stored away.

- The breathing apparatus must also be disassembled by the clinician to be washed and cleaned before storage.

3.8   Participant Protocols

The following outline describes the subject protocol required to perform a respiratory training session. The software was designed to follow this protocol as closely as possible.

MEP/MIP Testing:

- On the % TRAINING page, discuss with the clinician the percentage level that the participant feels the most comfortable with for RMT while clinician attaches MEP/MIP valve attachment to breathing apparatus.

- On the MEP/MIP page, listen to the clinician's instruction on how to inhale strongly through the apparatus for five (5) seconds for MIP test with 100% effort.

- Participant must put on breathing apparatus and wait for device countdown before inhaling strongly into the device for five (5) seconds.

- After MIP test, listen to the clinician's instruction on how to exhale strongly through the apparatus for five (5) seconds for MEP test while the clinician attaches the valve attachment to the opposite side of the tee.

- Participant must put on breathing apparatus and wait for device countdown before exhaling strongly into the device for five (5) seconds with 100% effort.

Training mode:

- When on TRAINING page and clinician has attached the threshold valves with adjusted settings, discuss with clinician time duration of training session.

- After agreeing on duration, patient must put on breathing apparatus and wait for device countdown before inhaling and exhaling full, complete breath cycles with 100% effort for the duration of time agreed with clinician.

- After training is complete, remove the breathing apparatus and give the device to the clinician for cleaning.

- Observe session results on RESULTS page and compare current results to past results to track for any growth.

IV. RESULTS/DISCUSSION

4.1   Device Verification and Validation Results

With the system pressure sensor connected inline with the Delta-Cal sensor calibration

device, pressure on the calibrator was adjusted in increments of 10 mmHg and the

prototype display reported the internally scaled pressure reading. Measurements were

recorded every five minutes, for twenty minutes to characterize any drift in the

measurement. Both Delta-Cal and BreathForce readings were recorded for comparison

and device validation in APPENDIX VI.

Table 5 reports the averages of data points (n=5) obtained for pressure measurements using the prototype. The average standard deviation of the prototype pressure measurement versus Delta-Cal measurement was 1.16 mmHg (n=11). Average percent error of the BreathForce measurements versus calibrator was 3.44% across the span of pressures measured. This percent error is larger than the 2% accuracy range rated by pressure sensor's manufacturer that covered an even larger range (0-310.29 mm Hg) versus bench testing (-50 to 50 mm Hg). This error was deemed acceptable as errors could have stemmed from calculation errors in the software.

Table 5 – Statistics of Average BreathForce measurements compared to
Delta-Cal measurements

Delta-Cal (mmHg)	Average BF (mmHg)	Std. BF	% Error
-50	-48	2.33	4.12
-40	-38	1.97	4.41
-30	-29	1.48	4.41
-20	-19	0.89	3.68
-10	-10	0.49	4.41
0	0	0.00	0.00
10	10	0.49	4.41
20	19	0.77	2.94
30	29	0.89	2.45
40	38	1.84	4.04
50	49	1.64	2.94
	Average	1.16	3.44

Figure 43 shows plotted averages of BreathForce measurements versus Delta-Cal values. The slope of BreathForce average was 0.96283 with an R-squared value of 0.99997, indicating the prototype and Delta-Cal has an almost 1:1 linear relationship with a nearly perfect regression analysis.

4.2   Humidity Sensitivity Test

An evaluation of sensor performance in the presence of increased humidity was

performed by breathing in the apparatus to simulate a normal RMT session for one

minute to build up moisture inside the airflow path. After breathing through the

apparatus, the sensor was then disconnected from the breathing apparatus and connected

to the Delta-Cal to provide a standardized, consistent pressure. The procedure of

breathing into the sensor was repeated every five minutes for each measurement session

Table 6 – Statistics of BreathForce measurements with humidity compared to
Delta-Cal measurements

Humidity			
Delta-Cal (mmHg)	Average BF (mmHg)	Std. BF	% Error
-50	-49	1.64	2.94
-40	-38	1.82	4.04
-30	-29	1.19	3.43
-20	-20	0.16	0.74
-10	-10	0.49	4.41
0	0	0.00	0.00
10	10	0.49	4.41
20	19	0.99	4.41
30	29	1.22	3.43
40	39	1.69	3.68
50	48	2.18	3.82
	Average	1.08	3.21

to ensure fresh moisture was in the sensor. The pressure applied by the Delta-Cal were adjusted in 10 mmHg increments, from -50 to 50, for every five minutes up to 20 minutes total. Table 6 reports the averages of measured pressure in mmHg.

The average standard deviation of the BreathForce measurement with increased humidity versus Delta-Cal reported pressure was 1.08 mmHg. The average percent error of the prototype versus Delta-Cal measurement was 3.21%. Again, the slight deviation from accuracy rating could have been due to calculation error in the software. Additionally, it was expected that humidity would not affect the pressure measurement due to the manufacturer including a membrane that separates the micro-diaphragm from the fluid path in the pressure sensor.

Figure 44 shows plotted averages of BreathForce values in the presence of increased humidity versus Delta-Cal values. The slope of BreathForce pressure
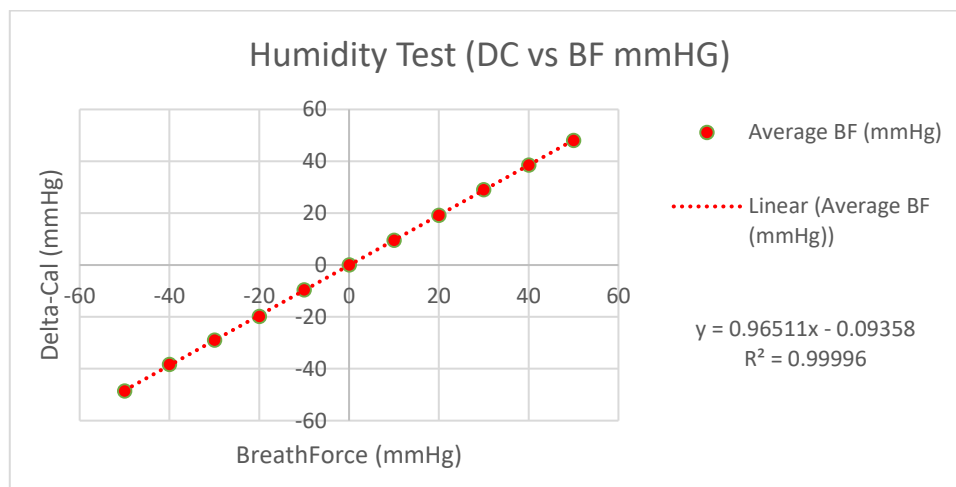
Figure 44 – Graph showing comparison of average BreathForce measurements with humidity to Delta-Cal measurements

measurements over the experiment range was 0.96511 with an R-squared value of 0.99996, indicating the prototype did not appear to be affected by humidity as expected. The calibrations plot shows an almost 1:1 linear relationship and that the regression model fits observations.

4.3   Water Sensitivity Test

A moisture test was performed by removing the sensor from the prototype and running it under water to simulate a cleaning/rinsing step. Before drying, the sensor was reconnected to the prototype and to the Delta-Cal. This procedure was repeated every 5 minutes for each measurement session. Pressures in Delta-Cal were adjusted in 10 mmHg increments, from -50 to 50, for every five minutes up to 20 minutes total.

Table 7 shows the averages of measured pressure in mmHg. The average standard

deviation of the BreathForce measurements after rinsing versus Delta-Cal measurement

was 0.92 mmHg. Average percent error of BreathForce pressure measurements versus

Delta-Cal measurement was 2.41%. Like in the humidity test, the dielectric silicon barrier

prevents moisture from entering the micro-diaphragm and unaffecting its measurement

capabilities. Although the manufacturer of the pressure sensor initially designed the

pressure sensor to be used only once, finding that a washed sensor producing a similar, or

Table 7 – Statistics of BreathForce measurements with moisture after
cleaning compared to Delta-Cal measurements

Moisture			
Delta-Cal (mmHg)	Average BF (mmHg)	Std. BF	% Error
-50	-49	1.72	2.94
-40	-40	0.59	1.10
-30	-29	0.66	1.96
-20	-20	0.64	2.21
-10	-10	0.37	1.47
0	0	0.00	0.00
10	10	0.47	2.94
20	19	0.77	2.94
30	29	1.48	4.41
40	39	1.62	3.31
50	48	1.84	3.24
	Average	0.92	2.41

close, percent error as the manufactured accuracy rating shows that this sensor can be used multiple times. This reduces the amount of pressure sensors needed to be kept on hand in case of need of pressure sensor replacements.

Figure 45 shows plotted averages of BreathForce with Moisture values versus Delta-Cal values. Slope of BreathForce average was 0.9723 with an R-squared value of 0.99999, indicating BreathForce with Moisture and Delta-Cal have an almost 1:1 linear relationship and that regression model fits observations.
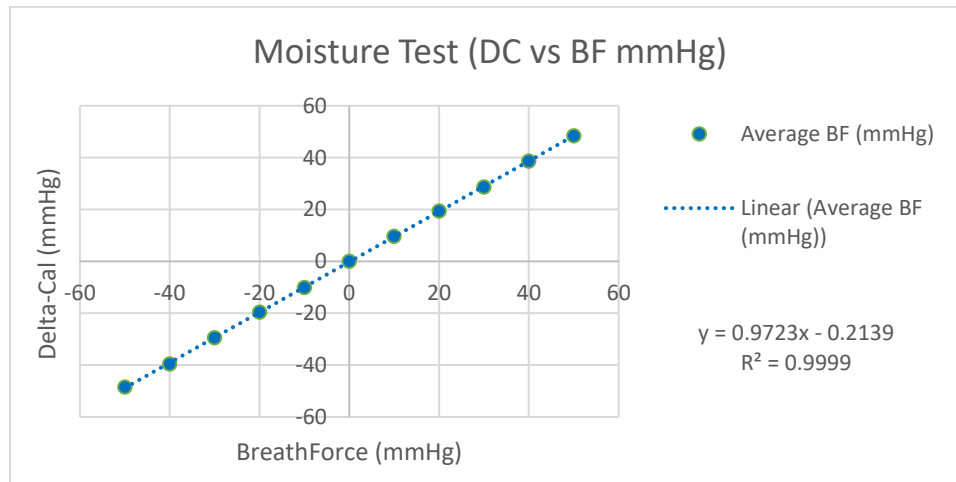
Figure 45 – Graph showing comparison of average BreathForce measurements with moisture after cleaning to Delta-Cal measurements

4.4   Temperature Sensitivity Test

A temperature test was performed to investigate the effect of temperature variations on the accuracy of the pressure sensor. The sensor was placed in a freezer for one hour. The sensor was then placed back into the prototype and calibration tests performed every 5 minutes for 20 minutes. Before each test, temperature of the sensor

was recorded using a laser temperature sensor (Fluke IR Thermometer, 561). The sensor was allowed to warm-up to room temperature to see if pressure readings exhibited any drift in the pressure measurement. The pressures in the calibrator were again adjusted in 10 mmHg increments, from -50 to 50. Table and plot of Temperature vs Pressure is in Figure 46.

Table 8 – Statistics of BreathForce measurements with colder temperatures compared to Delta-Cal measurements

Temperature	Temperature Range	13.2F	50.8F

Delta-Cal vs. BreathForce with different Temperature

	-50	-40	-30	-20	-10	0	10	20	30	40	50
13.2F	-46	-37	-28	-18	-8	0	11	21	30	40	49
34.7F	-48	-38	-29	-19	-9	0	10	20	29	39	49
45.5F	-48	-38	-29	-19	-10	0	10	19	29	39	49
47.5F	-48	-39	-29	-19	-10	0	10	19	29	39	49
50.8F	-48	-38	-29	-19	-10	0	10	19	29	38	49
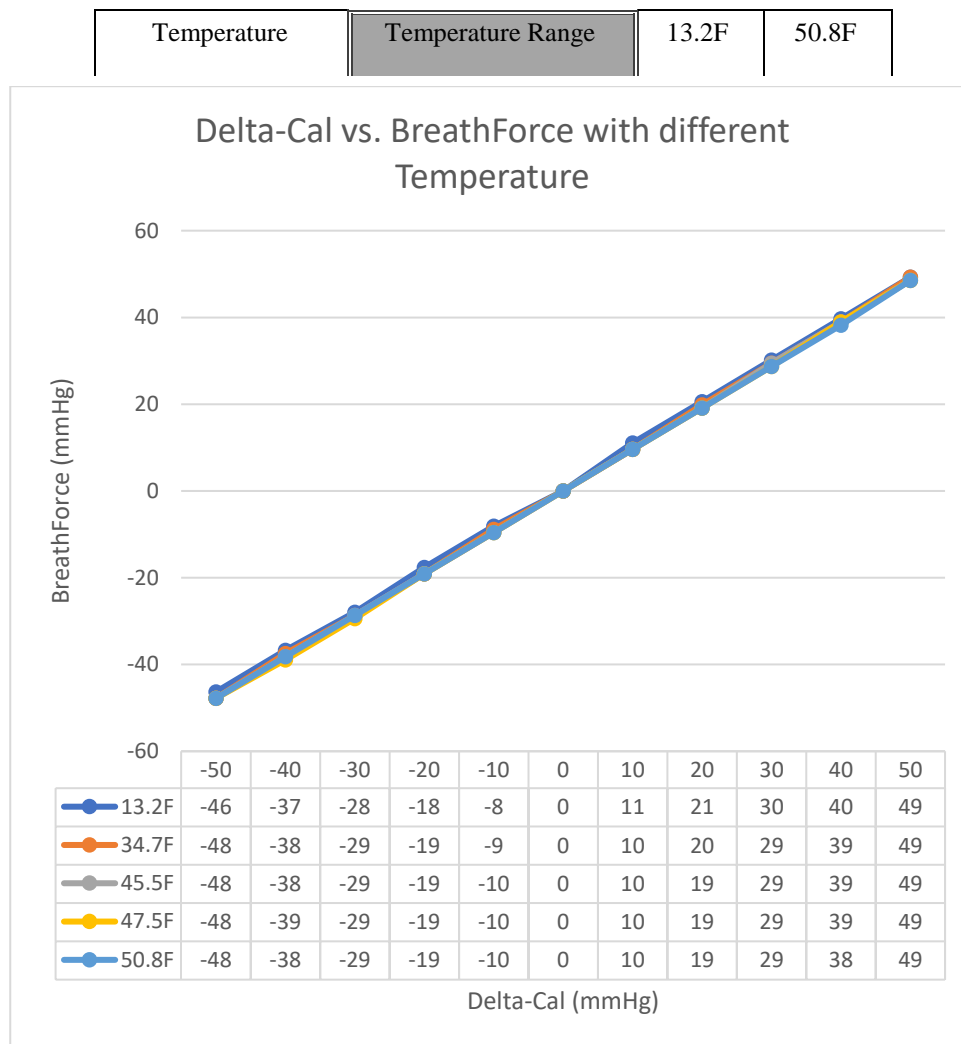
Delta-Cal (mmHg)

Figure 46 – Comparison of Delta-Cal vs BreathForce measurements over different temperatures

Table 8 shows the averages of measured pressure in mmHg. The average standard deviation of BreathForce measurement with temperature versus Delta-Cal measurement was 1.34 mmHg. The average percent error of the BreathForce pressure measurements

versus Delta-Cal measurements was 3.67%. Percent error deviation from 2% could be due to calculation error in the software as the temperature range that the sensor was being tested in (13.2-50.8F) was mostly within the operating temperature range stated by the manufacture (35.6-122F).
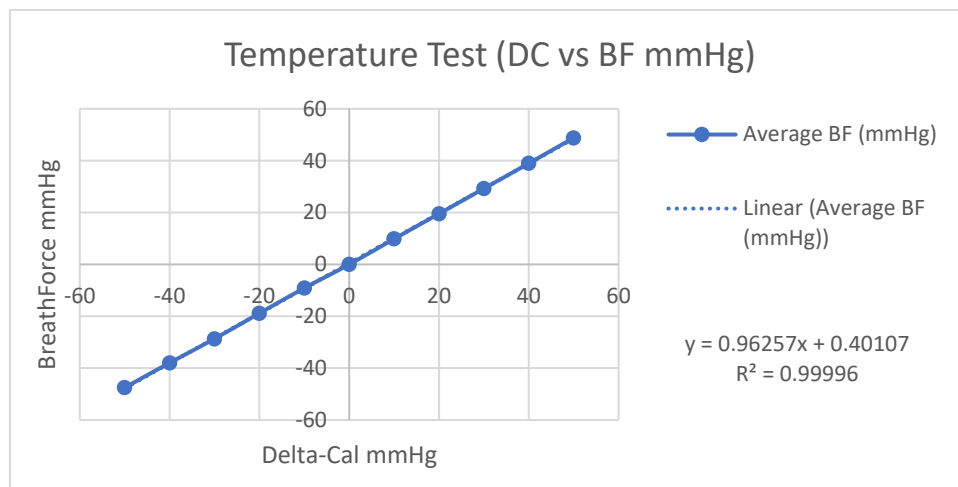
Figure 47 – Graph showing comparison of average BreathForce measurements with colder temperatures after cleaning to Delta-Cal measurements

Figure 47 shows plotted averages of BreathForce with Temperature values versus Delta-Cal values. The calculated slope of the BreathForce pressure measurement linear fit equation was 0.96257 with an R-squared value of 0.99996, indicating the prototype was not sensitive to variations in temperature and had an almost 1:1 linear relationship and a regression model that fit observations for this experiment.

4.5   Interpretation of Pressure Sensor Tests

The BreathForce pressure sensor produced near 1:1 relationships compared to the Delta-Cal pressure standard across all tests. Despite an average standard deviation of

1.16, 1.08, 0.92, and 1.34 (normal, humidity, moisture, and temperature, respectively)

and percent errors of 3.44%, 3.21%, 2.41%, and 3.67%, respectively, the larger standard

deviations occurred mostly in the higher-pressure ranges of 30-50 mmHg, consistent with

most common pressure and force sensors with accuracies reported as a percent of full

scale. The farther the measurement from zero, the larger the error. Regardless, these

larger values of pressures exceed normal inspiratory and expiratory training regimens set

by RMT procedures. Regular training settings for SCI subjects usually range from 5-20

mmHg for both expiratory and inspiratory activates. When considering only that range,

deviation average drop to 0.53, 0.43, 0.45, and 0.83 (normal, humidity, moisture,

temperature, respectively) and percent errors fall to 3.09%, 2.79%, 1.91%, and 3.78%,

respectively. With an R-squared of greater than 0.99 for all tests (normal, humidity,

moisture, and temperature), it can be confidently claimed that the pressure sensor used in

the BreathForce prototype and the conversion calculations (APPENDIX IV) in the

system software produces accurate pressure readings regardless of the state of the sensor

and produce similar results to that of the calibrated pressure reference device.

4.6   Experimental Subject Testing

Only two volunteers were used to obtain subject data. Because this test was only

conducted on two participants, each at a single sitting, it was not possible to see long-

term effects of using the prototype device and track improvements in respiratory function

over many sessions. Although the main purpose of the device is to improve respiratory

function, the purpose of these tests was for prototype evaluation.

Subject A had lower extremity paralysis while subject B had no history of SCI or paralysis. Subject B was used as the control for the experiment. Both participants had pressure threshold settings set to 10% of their individual MEP/MIP as this was the initial training percentage set by established RMT procedures. Subject A performed two 10% threshold sessions. The control subject B performed three 10% tests and an additional 20% and 30% test to examine for deviations in measurement recording and device behavior as threshold settings increased by manual manipulation of compression springs inside the airway valves.

Table 9 and Table 10 show results of expiratory and inspiratory RMT of both subjects A and B, respectively. Deviation of expiratory pressure was calculated by finding the average pressure for all complete expiratory cycles and subtracting it from the

target expiratory pressure. Deviations of inspiratory pressures were calculated in the same way.

Table 9 – Subject A data

	Summary of Participant Data		
	10%		
MEP	93.00	93.00	
Target Exp	9.30	9.30	Average
Average Exp	8.77	7.97	8.37
Deviation	-0.53	-1.33	-0.93
MIP	-94.00	-94.00	
Target Insp	-9.40	-9.40	Average
Average Insp	-8.23	-7.06	-7.65
Deviation	1.17	2.34	1.76

Table 10 – Subject B data

	Summary Control Data			
	10%			
MEP	85.00	85.00	85.00	
Target Exp	8.50	8.50	8.50	Average
Average Exp	19.46	19.62	19.55	19.54
Deviation	10.96	11.12	11.05	11.04
MIP	-111.00	-111.00	-111.00	
Target Insp	-11.10	-11.10	-11.10	Average
Average Insp	-23.47	-23.23	-23.34	-23.34
Deviation	12.37	12.13	12.24	12.24
	20%	30%		
MEP	85.00	85.00		
Target Exp	17.00	25.50	Grand Average	
Average Exp	24.86	28.03	22.30	
Deviation	7.86	2.53	7.14	
MIP	-111.00	-111.00		
Target Insp	-22.20	-33.30	Grand Average	
Average Insp	-32.06	-37.93	-28.00	
Deviation	9.86	4.63	8.91	

Figure 48 and Figure 49 show graphical results of three full breath cycles taken from Subject A and B's RMT data. Graphs labeled "Trial" represent Subject A and graphs labeled "Control" represent Subject B. Inhalation and exhalation cycles of a test subject can be determined from the graphs when the data measurements cross the x-axis: data crossing from positive values to negative values indicate expiration while data

Figure 48 – Graphs of RMT data for subject A. Exhalation generates positive data
values, while inhalation generates negative data values.
Time values removed for clarity.
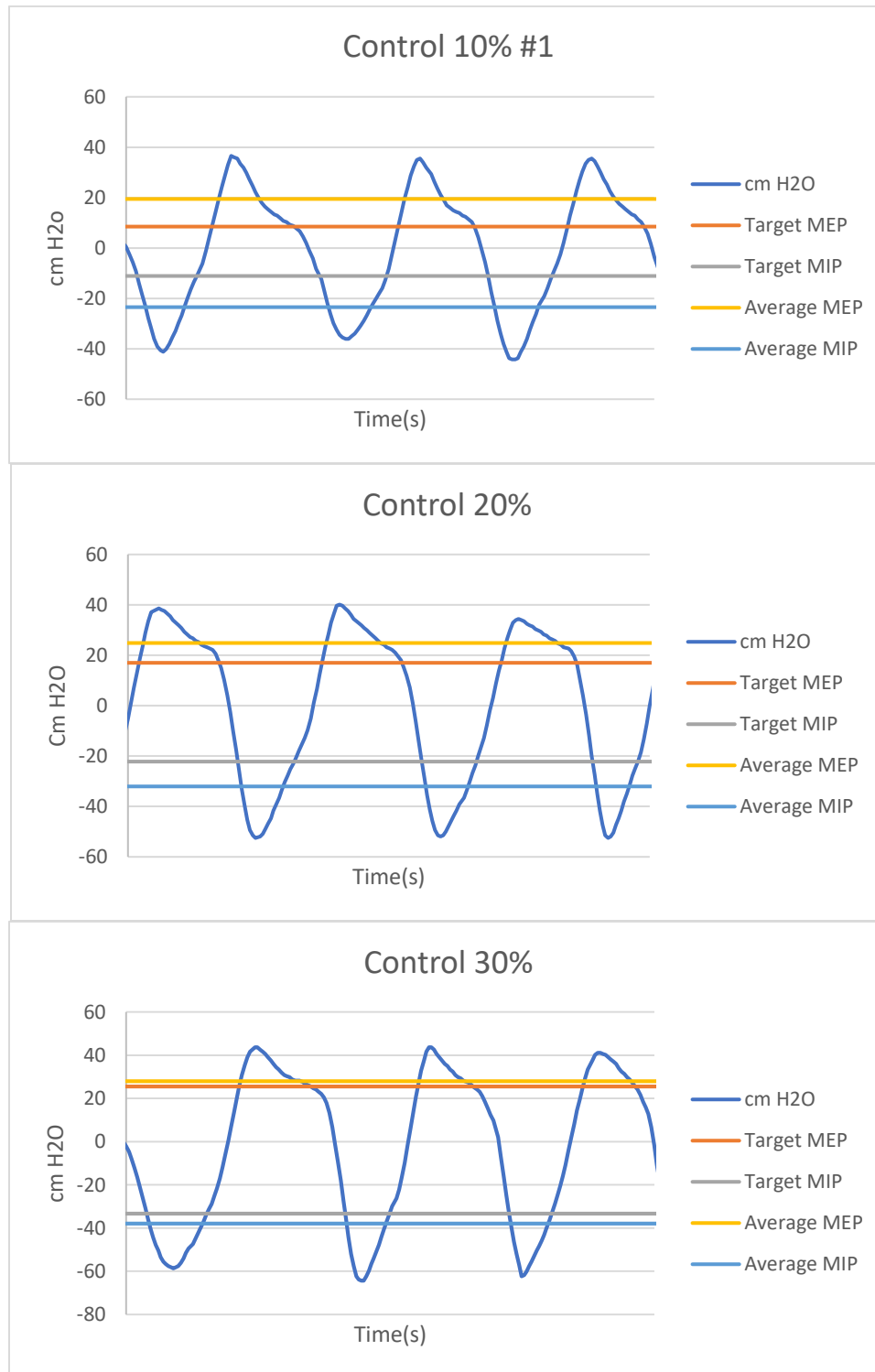
Figure 49 – Graphs of RMT for Control 10%-30% (Subject B) to examine decreasing deviation with increasing target pressures. Time values removed for clarity.

crossing from negative values to positive values indicate inspiration. During inspiration, the IMT threshold device is activated and builds pressure inside the pressure sensor that had been calibrated to zero in the initial startup phase of the device. When the user finishes inspiration and is about to begin exhalation, the one-way valve from the IMT threshold device to the pressure sensor closes and deactivates the IMT device. PEP threshold device is activated when the particpant begins exhalation and exhales the just inspired volume of air plus the expiratory reserve volume of air previously present in the lung. Exhalation of inspired volume plus expiratory reserve volume is detected as a negative change by the sensor, resulting in negative pressure values. Target expiratory and inspiratory values, calculated from MEP/MIP test, are represented by orange and grey lines on graphs, respectively. Average expiratory and inspiratory pressures were calculated by identifying and taking the average of all expiratory and inspiratory amplitudes, respectively, and are represented by yellow and blue lines on the graphs, respectively.

4.7   Interpretation of Experimental Subject Testing

Because the pressure sensor was validated previously by Delta-Cal, it was safe to assume that pressure values displayed on BreathForce were accurate. After MEP/MIP tests were conducted, threshold valve levels were set to match target values derived from MEP/MIP. Subject A, who had a SCI and was set to 10% training levels, had an average deviation of average expiratory BreathForce values from target expiratory pressures was 0.93 cm H_2O and a deviation from target inspiratory pressures of 1.76 cm H_2O.

73

RMT results of subject B, who had no SCI and was set to multiple training levels, displayed larger deviations and percent errors than those in subject A. Target values measured exceeded the specified threshold and resulted in larger percent errors. Average deviation of average expiratory pressures from target expiratory pressures across multiple training levels was 7.14 cm H_2O. Average deviation from target inspiratory pressures across the same training levels was 8.91 cm H_2O. These large average deviations and percent errors represent a disconnection in conversion between the target pressures calculated from MEP/MIP mode to the pressure sensor values printed on the threshold devices. When users crack the threshold valves during RMT, there is no feedback of the valves to maintain this pressure (and therefore resistance) in steady state. Future studies will address this limitation replacing the threshold pressure devices with electronic valves that will be calibrated to BreathForce's standard of measurement and provide active feedback to adjust valve opening to create a more consitant resistance throughout the inhalation/exhalation cycle.

Comparing only 10% settings (Figure 50), Subject B had a larger range between MEP and MIP values (85 to -111 cm H_2O respectively) than subject A (93 to -94 cm H_2O respectively). No definitive reason could explain this difference except possibly that subject B is healthy and has a wider range of respiratory capability than subject A who has an SCI. Variability between patients prevents a true one-to-one comparison as subject B also produced more average pressure values (19.54 cm H2O average expiration and -23.34 cm H2O average inspiration) that were greater than subject A (8.37 cm H2O

average expiration and -7.65 cm H2O average inspiration). This difference can be

explained by subject B does not have a SCI and has full respiratory capability while

subject A has a SCI and would have less respiratory capability. Subject A also had
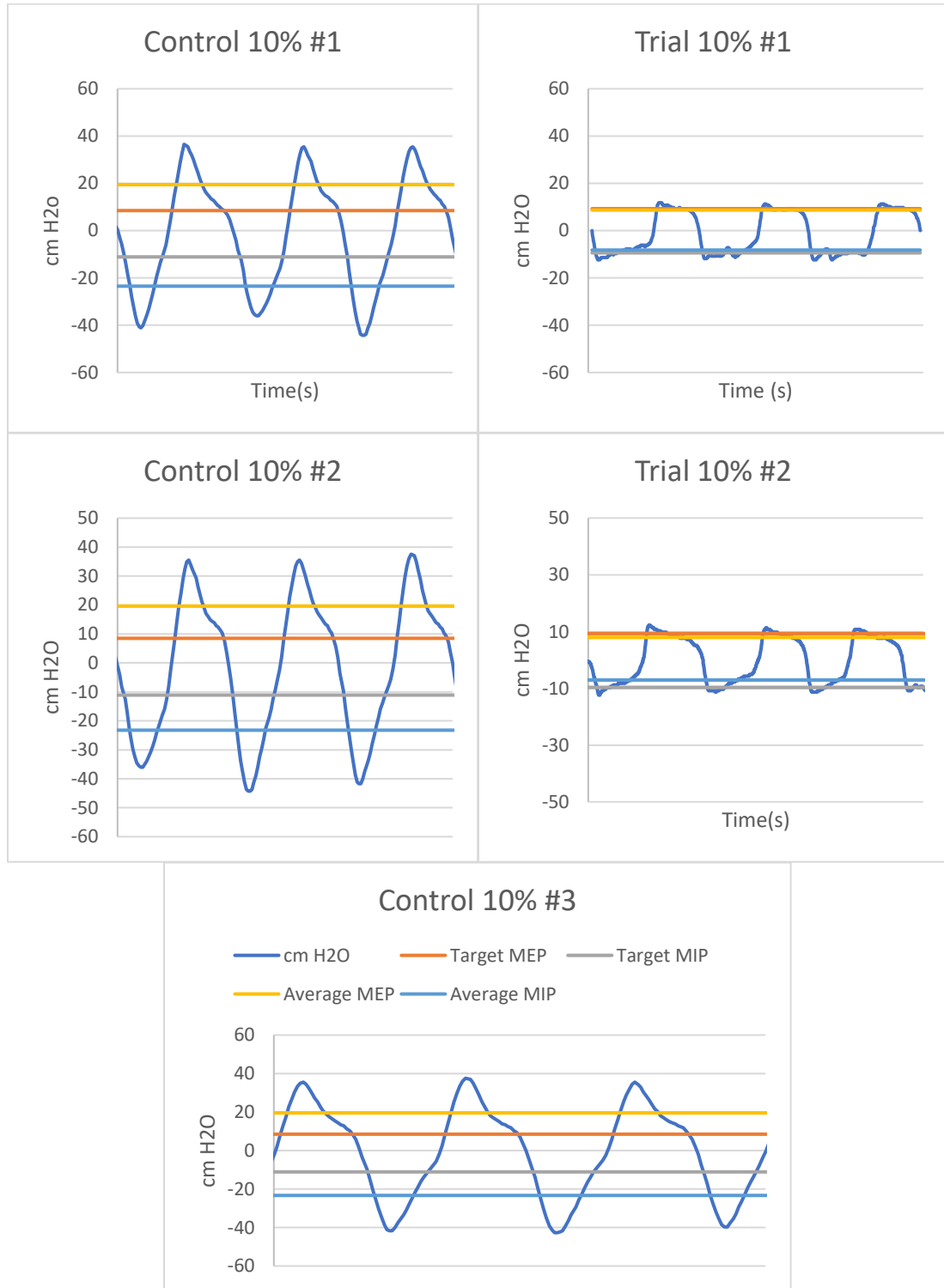
Figure 50 – Graphs of RMT for Control 10%-30% (Subject B). Time values removed for clarity. Legend in Control #3 can be applied to all graphs.

smaller deviations between average pressure values and target pressure values (0.93 cm H_2O difference in expiratory and 1.76 cm H_2O in inspiratory) than subject B (11.04 cm H_2O difference in expiratory and 12.24 cm H_2O in inspiratory). This difference between the two subjects can be explained by examining across the entire range of subject B's tests.

Aside from ten percent threshold data, there is a noticeable trend in deviations across all different percent training levels for subject B. As training percent levels increased (resulting in increased target expiratory/inspiratory values and increased device threshold levels), deviation from target values decreased. An explanation for such larger deviations and average pressures exceeding target pressures at lower levels for subject B versus subject A is that subject B is a healthy individual with a fully functional respiratory system that can easily exceed lower percent target training levels. As threshold level settings increased and deviation from target values decreased, subject B was approaching current maximum respiratory capabilities reflecting the subject's true respiratory capabilities.

There is a noticeable sharp peak in most of the RMT graphs at the beginning of each inhalation and exhalation cycle across both test subjects. This transient response represents the initial pressure build up in the pressure sensor that is required to "crack" (open) the threshold valve. This cracking pressure is always greater than the pressure required to sustain the valve to remain open in a steady state. A brief period of sustained

breathing was observed after the initial cracking pressure peak before crossing back to the opposite side of the x-axis.

Inconsistencies can be seen in RMT in charts of both test subjects where there are fluctuations after the initial cracking pressure peak. These inconsistencies represent the user's struggle to maintain appropriate pressure to keep the valves open as respiratory muscle stamina decreases. These minor fluctuations/inconsistencies create a graphical representation of a user's breathing cycle where smooth, consistent graphs are preferred.

4.8   Test of Manual Valves for Cracking Pressure

The data in Table 11 shows an average inspiration deviation between threshold valves and BreathForce was 6.34 cm H_2O and average exhalation deviation between threshold valves and BreathForce was 8.64 cm H_2O. Figure 51 shows a graphical

Table 11 – Deviation between average cracking pressure vs target values
on threshold valves

Deviation from Valve Value								
Inspiration								
Average	9.60	15.78	22.29	29.15	22.11	16.13	9.28	
Target	5.00	10.00	15.00	20.00	15.00	10.00	5.00	**Average**
Deviation	4.60	5.78	7.29	9.15	7.11	6.13	4.28	6.34
% Error	92.07	57.77	48.62	45.77	47.40	61.30	85.67	62.66
Expiration								
Average	-15.26	-16.98	-20.24	-28.47	-36.01	-37.90	-43.05	-50.25
Target	-5.00	-10.00	-15.00	-20.00	-25.00	-30.00	-35.00	-40.00
Deviation	-10.26	-6.98	-5.24	-8.47	-11.01	-7.90	-8.05	-10.25
% Error	205.27	69.77	34.91	42.33	44.04	26.34	22.99	25.63
	-45.19	-38.08	-36.23	-29.61	-20.77	-17.33	-14.30	
	-35.00	-30.00	-25.00	-20.00	-15.00	-10.00	-5.00	**Average**
	-10.19	-8.08	-11.23	-9.61	-5.77	-7.33	-9.30	-8.64
	29.11	26.92	44.92	48.07	38.47	73.27	186.00	61.20

representation of the deviation between BreathForce values and threshold device values. Inspiratory graph has less points than expiratory graph as IMT only scaled to 20 cm H_2O while expiratory scales to 40 cm H_2O.
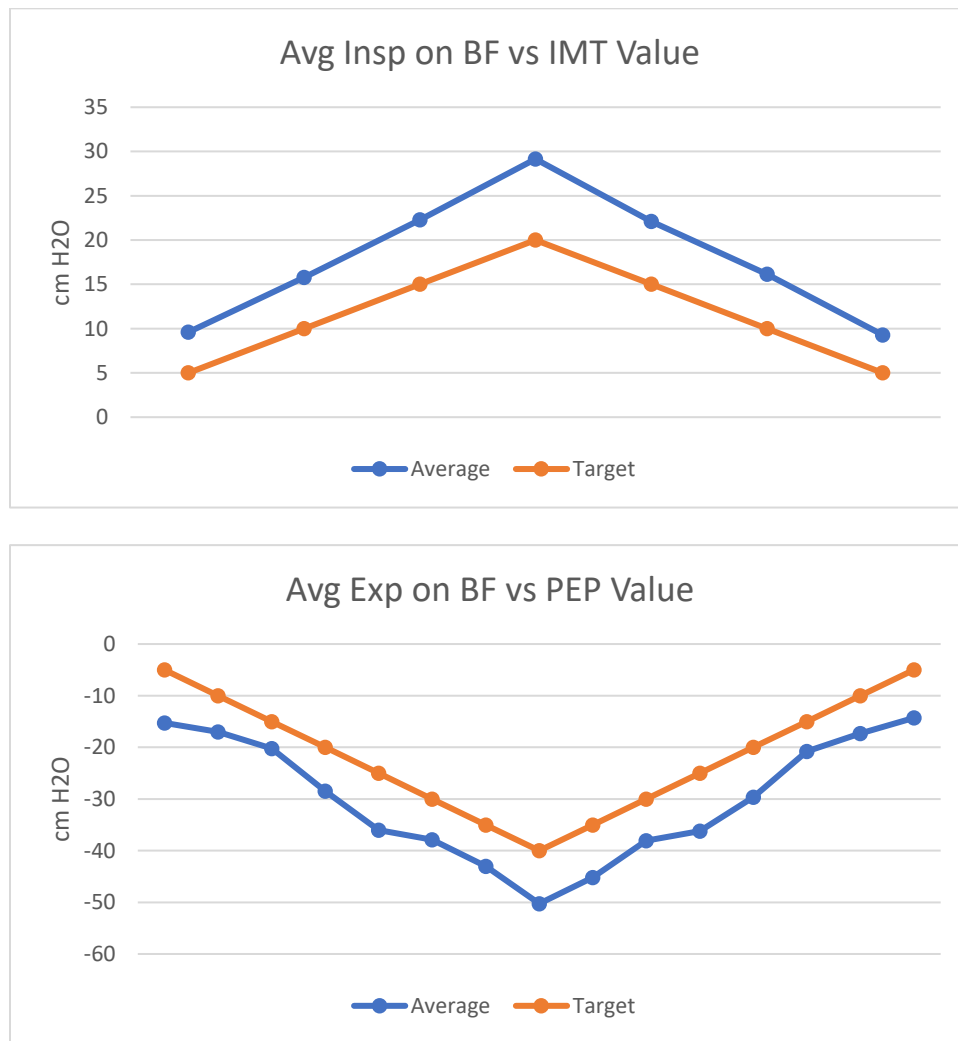
Figure 51 – Graphs of deviations of BreathForce prototype from IMT and PEP threshold valve values

4.9   Data Acquisition Rate Verification

Table 12 – Data acquisition rates across all subject tests (points per second)

Control 10% #1	Control 10% #2	Control 10% #3	Average
21.47	21.3	21.82	21.53
Trial 1	Trial 2	Average	T. Average
26.53	26.69	26.61	23.82

Data acquisition rates (Table 12) differed between preliminary and subject testing as some software changes were made to the device in between tests. These changes were minor as hardware and conversion factors remained the same. At an average of 21.53 points per second, BreathForce was able to produce acceptable graphical representations of the user's lung capacity. For future studies, it is suggested that improvements be made in data acquisition rates as larger, more expensive medical devices can acquire rates of 100 times per channel, per second [26].

4.10   Prototype Development Cost

The hardware budget to produce the prototype was set at $250.

APPENDIX VII lists the cost for the individual components used to fabricate the BreathForce prototype. This table excludes the Delta-Cal calibration device as it is not part of the device used for consumers. The grand total cost to manufacture one unit of BreathForce was $242.64.

4.11   User UI Feedback

After subject testing was completed, feedback on device UI was obtained. Comments were compiled based on user/therapist input to questions such as: legibility and organization of user input buttons and displays of feedback data; adherence of program to protocol; and suggestions for additional features/controls. For example, users

Figure 52 – Comparison of early version of UI of BreathForce (Left) compared to the final (Right)

appreciated the original software bar graph to visualize pressure readings that was included, users suggested the addition of a discrete indicator to identify target pressure values that the users needed to meet. Figure 52 (left) shows the initial version of the UI where no target pressure indicator was included the display. On this touchscreen page, LED bars were removed to increase available space on the page for a more detailed combination of feedback parameters. Instead, a new slider feature, two pressure LED indicators, two LED digits to display target pressures, and timer functions were programmed onto the page (Figure 52, right). Secondary interviews provided feedback from users indicating that the UI had improved from a very simplistic display to a very informative display for users during training sessions.

4.12   Limitations of Project

4.12.1 Test Subject population

Only two test subjects were used to generate subject test data. Given the short time frame of this project, physiological effects and long-term benefits of respiratory therapy were not obtained. Subject tests were performed only to evaluate the temporal response of the prototype and to obtain feedback to aid in development of BreathForce software with hopes for future clinical applications. With more time and subject testing, BreathForce will be evaluated as a clinic and in-home method for RMT.

4.12.2 Statistical Limitation

Because of the low subject population, it is impossible to establish statistical significance between the two subjects and there is minimal chance of detecting the true physiological benefits of device-users using BreathForce. As stated previously, given the short time frame of project, subject tests were only for prototype evaluation and feedback purposes. More time and subject testing can increase chances of statistical significance and performance analysis to examine the efficacy of the BreathForce system in a clinical setting.

4.12.3 Device Limitations

4.12.3.1   Data acquisition

With an average sample rate of approximately 20 points per second ($dT = 0.05$ sec), the data acquisition was limited by the microprocessor performance while updating the touchscreen. Existing larger and more powerful medical devices can easily achieve ~100 to 1000 points per second and have faster microprocessors that can handle a larger quantity of data. Having more points per second results in higher temporal resolution

data, and therefore increased frequency content to fully depict respiratory breathing patterns. Most of the expensive medical devices are computer-based, and have a fast and powerful x86 processor (or equivalent) running the equipment. The MEGA is a low power, 16MHz platform with relatively low-resolution A/D inputs (10-bit). The processing speed is also limited by the display hardware connected to the MEGA, which occupies more processing cycles, and reduced the DAQ performance. One solution would be to have two separate microprocessors in the device, one dedicated to sampling and one dedicated to user interaction and data storage.

4.12.3.2   Device Features

While only calculating MEP/MIP and working as a bi-directional spirometer, BreathForce lacks extra testing capabilities that other medical devices possess, such as true breath-by-breath metabolic analysis for complete assessment of a user's functional capacity and cardiorespiratory system, an array of software for additional testing capabilities like cardiac output. Lack of features can be attributed to a fixed time for development, single microprocessor limitations, and funding. Adding more features to the current prototype of BreathForce would have degraded sampling performance even further.

4.12.3.3   Mechanical Threshold Valves

While all the off-the-shelf respiratory products used were FDA approved, these threshold valves and pressure sensors were not originally designed to work with one another. With the breathing apparatus being a culmination of off-the-shelf products, these threshold valves separately may produce different results when used in a manner for which they were originally designed.

Mechanical threshold valves also create an inconsistent and therefore inaccurate pressure resistance compared to the tube markings indicators, by an average of 6.34 cm H_2O for IMT and -8.64 cm H_2O for PEP, when the spring force is adjusted. The springs press against one-way valves which will then only open at increasing pressures as the spring is increasingly pre-compressed. The noticeable sharp peak at the beginning of some breath cycles is typically the cracking pressure threshold, which is typically higher than the pressure required to maintain the valve in an open position. Since each breath is slightly different, it is hard to capture the actual cracking pressure for each breath. An accurate method to establish an increasing linear resistance relation over many sessions is needed. Electronically controlled proportional valves that open incrementally should be able to provide an accurate and repeatable flow resistance by simply altering the diameter of a short portion of the airflow path. This will be the subject of a future study.

4.13   Design Criteria Comparison

Based results shown in Table 13, the BreathForce prototype met a few of the

established criteria, but did not meet others. Most significantly, the data acquisition rate

of approximately 23.82 data points per second is significantly less than the minimum 100

points per second criteria. However, the worst calibration data was 4.41 % compared to

the standard used, which was acceptable. Deviations from verification tool were greater

than 2% across all evaluation tests, and prototype deviations of subject B when

comparing BreathForce prototype to IMT and PEP threshold devices' accuracy ranges

were greater than 2 cm H_2O and 1 cm H_2O, respectively. These failures stemmed from

the limitations that were just described and can be improved on in future studies.

Table 13 – Pass or fail of Design Criteria

Hardware	Design Criteria	Criteria Value	Actual Value	Pass/Fail
1	Breathing apparatus has inspiratory and expiratory airflows	2	2	PASS
2	Device must be smaller than advanced respiratory devices	70-53.3-124.5 cm	115-120.75-53.37 mm	PASS
3	Device can handle high data acquisition rates	100 pts/sec	23 pts/sec	FAIL
4	Power supply must last for ten 30-minute sessions	10 sessions	21 sessions	PASS
5	Customized UI to satisfy users	Yes	Yes	PASS
6	Calibration test accuracy within 5% of verification tester.	< 5%	< 4%	PASS
7	Prototype accuracy within +/- 1-2cm H_2O of threshold valves	1-2 cm H_2O	6-8 cm H_2O	FAIL
Software	Criteria	Criteria Value	Actual Value	Pass/Fail
1	Participant and Clinician protocol are followed	Yes	Yes	PASS

V.   CONCLUSION

5.1   Proof of Concept

Based on results and validation, BreathForce can replicate two features of existing RMT training devices at a lower cost while also providing additional benefits. Validation by comparing BreathForce pressure values to Delta-Cal pressure values suggest that the device can acquire and save accurate pressure measurements comparable to that of other pressure devices existing on the market. At a manufacturing cost of just under $250, the device is significantly cheaper than even the least most expensive device on the market (Powerbreathe K5, $672.23, [27]). Additionally, BreathForce is a portable alternative that has quick setup time for both clinical and home use. Having a battery powered system with touchscreen control helps keep BreathForce small and compact, eliminating the need for large computing units that would add to the size of the device and add additional costs.

5.2   Considerations for Future Development

While still in the prototype stage, BreathForce is fully functional, yet remains incomplete, with much potential for more testing capabilities and improved features. There are many considerations for future development. The transition from mechanical threshold valves to electronic proportional valves that restrict airflow should result in more consistent resistance during RMT. Another adaptation would be to eliminate the wired connection between the breathing apparatus and controller by using a wireless

connection. This would eliminate the clutter that comes with protruding wires and free the user from feeling tied to the processing unit. Another step past making the breathing apparatus wireless is to develop a fully wireless device, able to be paired to a smartphone and companion app that functions just like the 4D system's touchscreen display. This consideration would shift all electrical components from the processing unit to the mouth piece, eliminating extra hardware and making the device even more compact and portable.

VI.   REFERENCES

1. "Spinal Cord Injury (SCI) Facts and Figures at a Glance." Web. 8 Dec. 2016.

2. Devivo, M J. "Epidemiology of traumatic spinal cord injury: trends and future implications." *Spinal cord, U.S*. National Library of Medicine, May 2012, www.ncbi.nlm.nih.gov/pubmed/22270188.

3. Zimmer, M. Beth, et al. "Effect of Spinal Cord Injury on the Respiratory System: Basic Research and Current Clinical Treatment Options." *The Journal of Spinal Cord Medicine*, American Paraplegia Society, 2007, www.ncbi.nlm.nih.gov/pmc/articles/PMC2031930/. Accessed 27 Sept. 2017.

4. "Spinal cord injury: Coping with a life-Changing event." *Mayo Clinic*, Mayo Foundation for Medical Education and Research, 8 Oct. 2014, www.mayoclinic.org/disease-conditions/spinal-cord-injury/basics/symptoms/con-20023837

5. "Regulation of Respiration." *Intranet.tdmu.edu.ua*. itranet.tdmu.edu.ua/data/kafedra/internal/normal_phiz/classes_stud/en/med/lik/2%20course/4%20Cycle%20Physiology%20of%20breathing/02%20%20Regulation%20of%20breathing.htm.

6. Daniela G. L. Terson de Paleville, et al. "Respiratory Motor Control Disrupted by Spinal Cord Injury: Mechanisms, Evaluation, and Restoration." *Translational stroke research. U.S*. National Library of Medicine, 1 Dec. 2011, www.ncbi.nlm.nih.gov/pmc/articles/PMC3297359/.

7. Ovechkin, Alexander V., et al. "Respiratory motor training and neuromuscular plasticity in patients with chronic obstructive pulmonary disease: a pilot study." *Respiratory physiology & neurobiology, U.S.* National Library of Medicine, 15 July 2016, www.ncbi.nlm.nih.gov/pmc/articles/PMC4887410/.

8. "Levels of Injury." *Understanding Spinal Cord Injury*, www.spinalinjury101.org/details/levels-of-injury.

9. Frankel, H L, et al. "Long-Term survival in spinal cord injury: a fifty year investigation." *Spinal cord, U.S.* National Library of Medicine, Apr. 1998, www.ncbi.nlm.nih.gov/pubmed/9589527.

10. Lawrence James, Smoke at Management and Science University Follow. "Regulation of Respiration." *Share and Discover Knowledge on LinkedIn SlideShare*. 2011. Web. 08 Dec. 2016.

11. *Michigan Medicine | University of Michigan*. http://www.med.umich.edu/lrc/coursepages/m1/anatomy2010/html/modules/spinal_cord_module/Files/latsc_diagram.jpg

12. HCPro, Inc. *Complications from immobility by Body System*, 25 Nov. 2017, www.hcpro.com/LTC-286850-10704/Comlications-from-immobility-by-body-system.html.

13. Editorial, AnesthesiaUK. "Tests of pulmonary function." *Anaesthesia UK: Tests of pulmonary function*, www.frca.co.uk\article.aspx?articleid=100023.

14. Garshick, E, et al. "A prospective assessment of mortality in chronic spinal cord injury." *Spinal cord, U.S.* National Library of Medicine, July 2005, www.ncbi.nlm.nih.gov/pubmed/15711609.

15. John E. Hall. Textbook of Medical Physiology. 13th edition.

16. "Assisted Cough for a Person Who Is Obese." *WebMD*. WebMD. Web. 08 Dec.

    2016.

17. "Mechanism of Breathing - Fun Science." Web. 8 Dec. 2016.

18. "Diaphragm Pacing." Medscape. Web. 16 Dec 2015. 08 Dec 2016.

19. *Amazon*. www.amazon.com

20. "Micro Diary Spirometer." *MedicalDeviceDepot.com*,

    www.medicaldevicedepot.com/Micro-Diary-Spirometer-p/

21. *MGC Diagnostics*: Ultima PF™ pulmonary function system with RTD.

    http://mgcdiagnostics.com/products/view/ultima-pf-with-rtd

22. "Overcome the Pressure for Consistent PEP and IMT Therapy." *Respironics*,

    Respironics,

    www.caremed.co.nz/Modules/LSDocumentManager/DocumentDownload.aspx?Doc
    umentId=165.

23. "PendoTech PRESS-S-000 PressureMAT Single-Use Sensor, Luer Fitting,

    Polycarbonate." *Cole-Parmer*, www.coleparmer.com/i/pendotech-press-s-000-
    pressuremat-single-use-sensor-luer-fitting-
    polycarbonate/1940632?PubID=UX&persist=true&ip=no&gclid=Cj0KCQiAjO_
    QBRC4ARIsAD2FsXPerwbpTKWazdmVEF2eqHv_T7JtDs2RlRAZFo_rAWW
    QrgsPwj9JcMsaAleLEALw_wcB.

24. Industries, Adafruit. "Adafruit Assembled Data Logging shield for

    Arduino." *Adafruit industries blog RSS*,

    www.adafruit.com/product/1141?gclid=Cj0KCQiAjO_QBRC4ARIsAD2FsXNil

OqsffnAR5Ft_-

BgvqGe21vboRHhHBVOmHh1AG4ui1YwMA8sb1caAr2JEALw_wcB.

25. "ULCD-32PTU." *4D Systems*, www.4dsystems.com.au/product/uLCD_32PTU/.

26. *Arduino*. https://www.arduino.cc/en/Products/Compare

27. *Powerbreathe Kinetic K5 Breathing Trainer*.

https://www.powerbreathe.com/powerbreathe-k5

PendoTECH Cable Adapter for Single Use Pressure Sensor (12 feet)
Part# PDKT-650-298

```
4  BLACK  (-EXCITATION)
3  WHITE  (+SIGNAL)
2  GREEN  (-SIGNAL)
1  RED    (+EXCITATION)
```
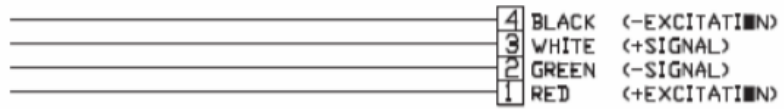
Wire connection schematic for PendoTech Pressure sensor for 4-pin plug connection

VIII. APPENDIX II
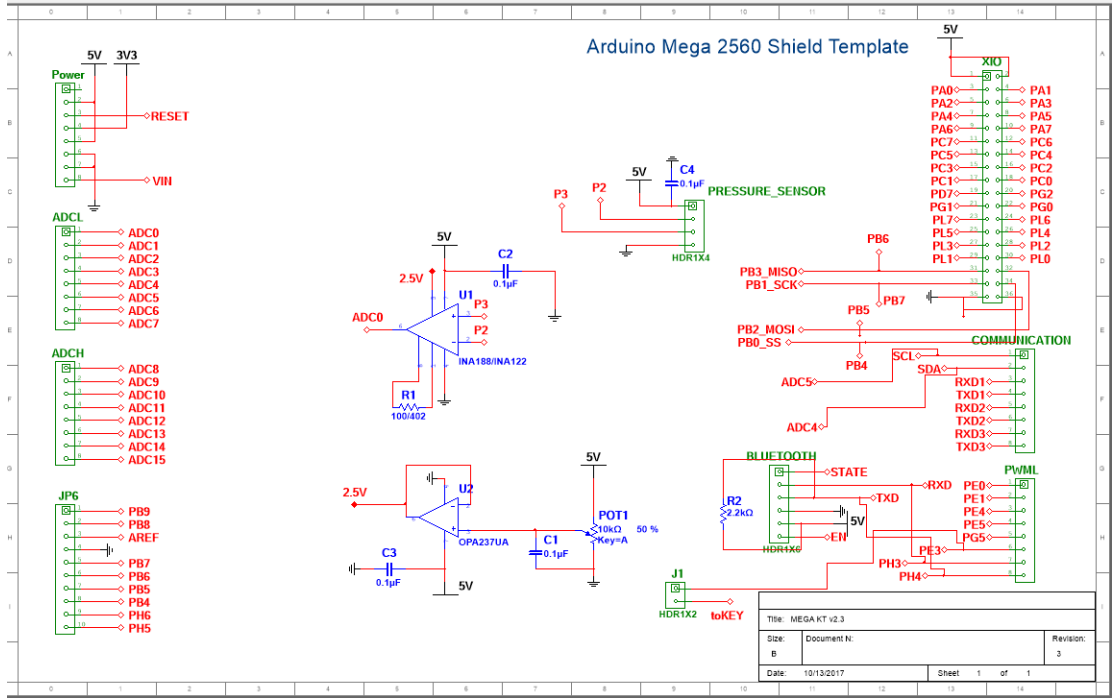
BreathForce Circuit Schematic:

IX. APPENDIX III

Schematic drawings of case enclosure:

Screen cover

Screen cover

Case Bottom + Screen Holder Bo
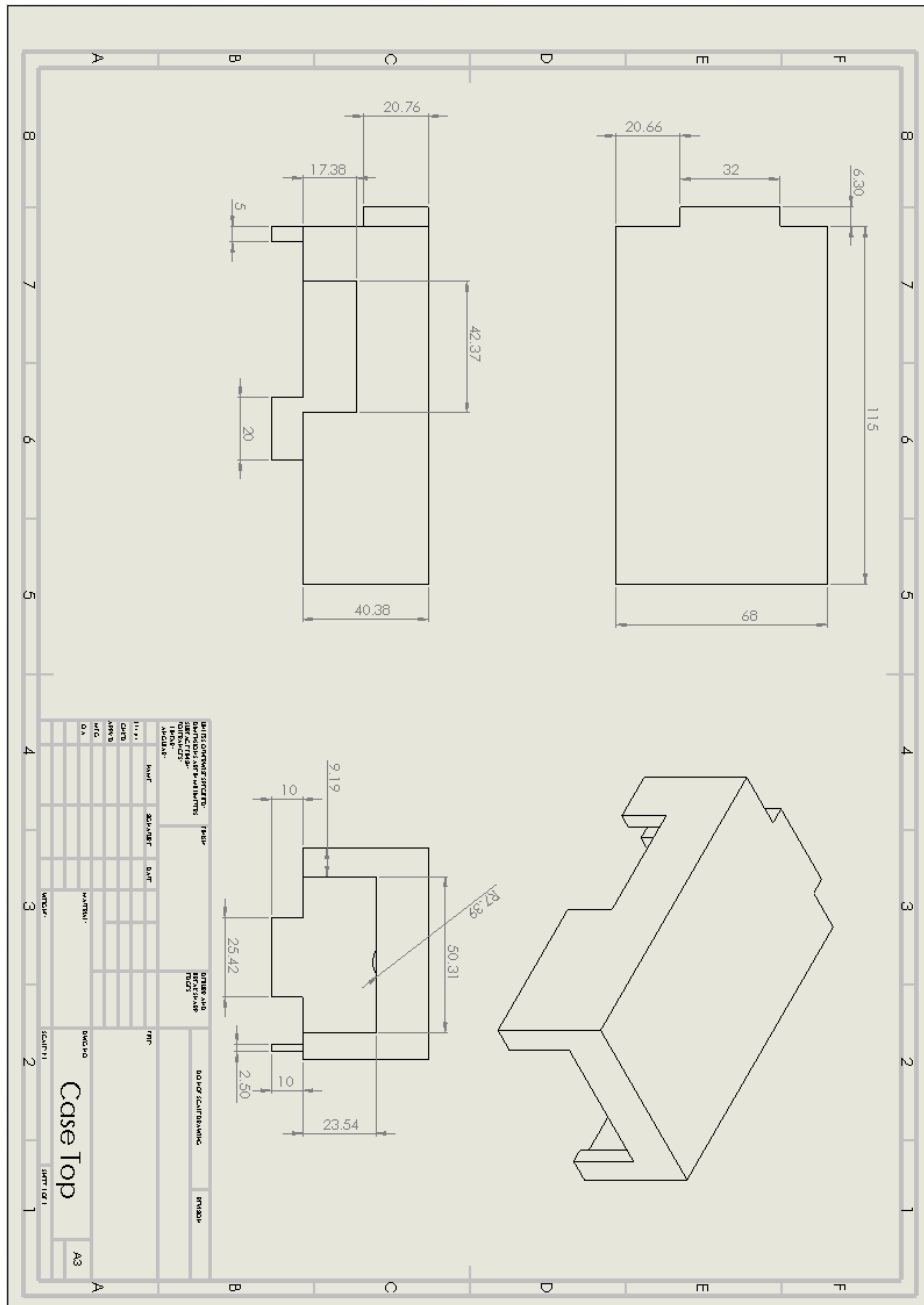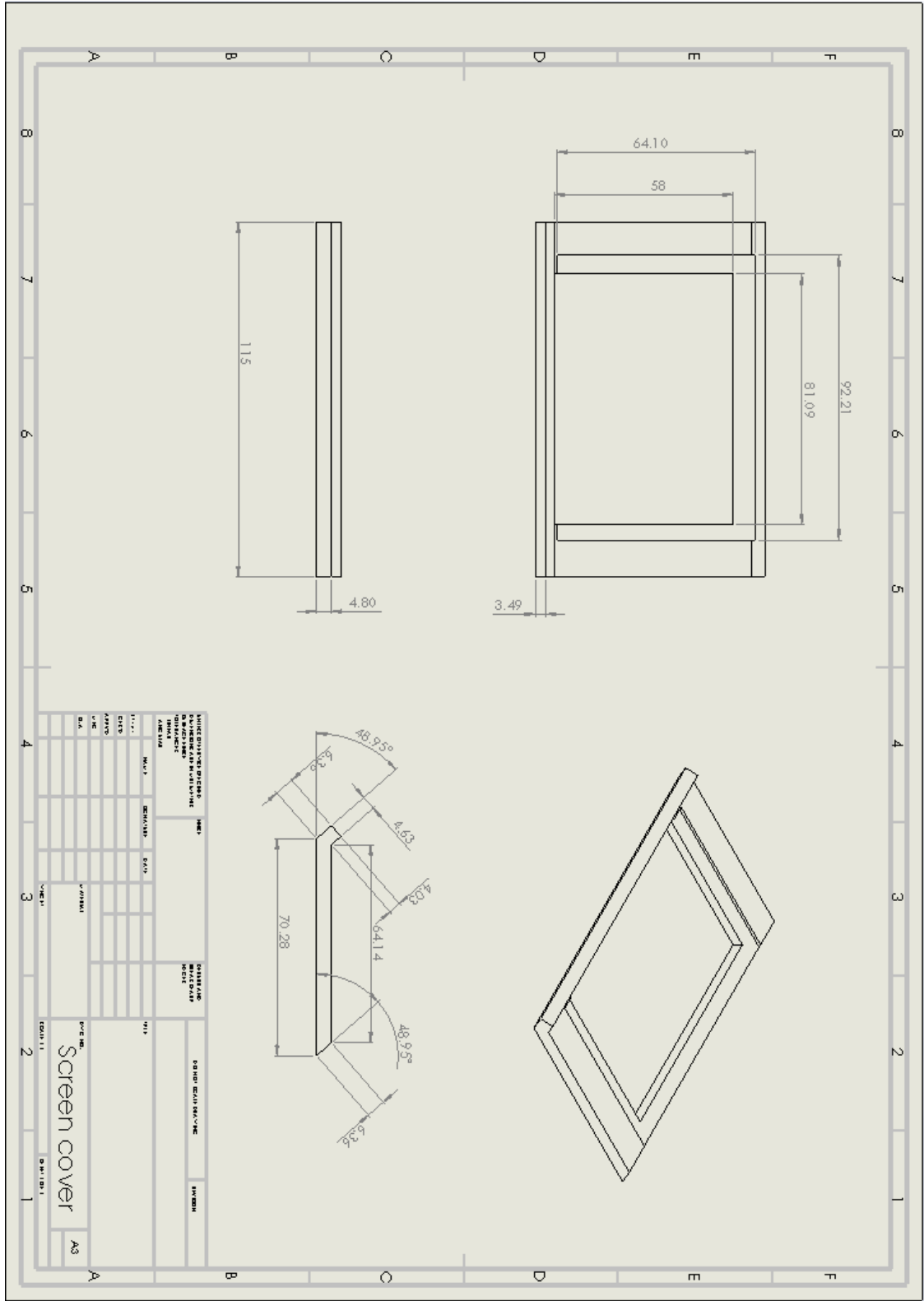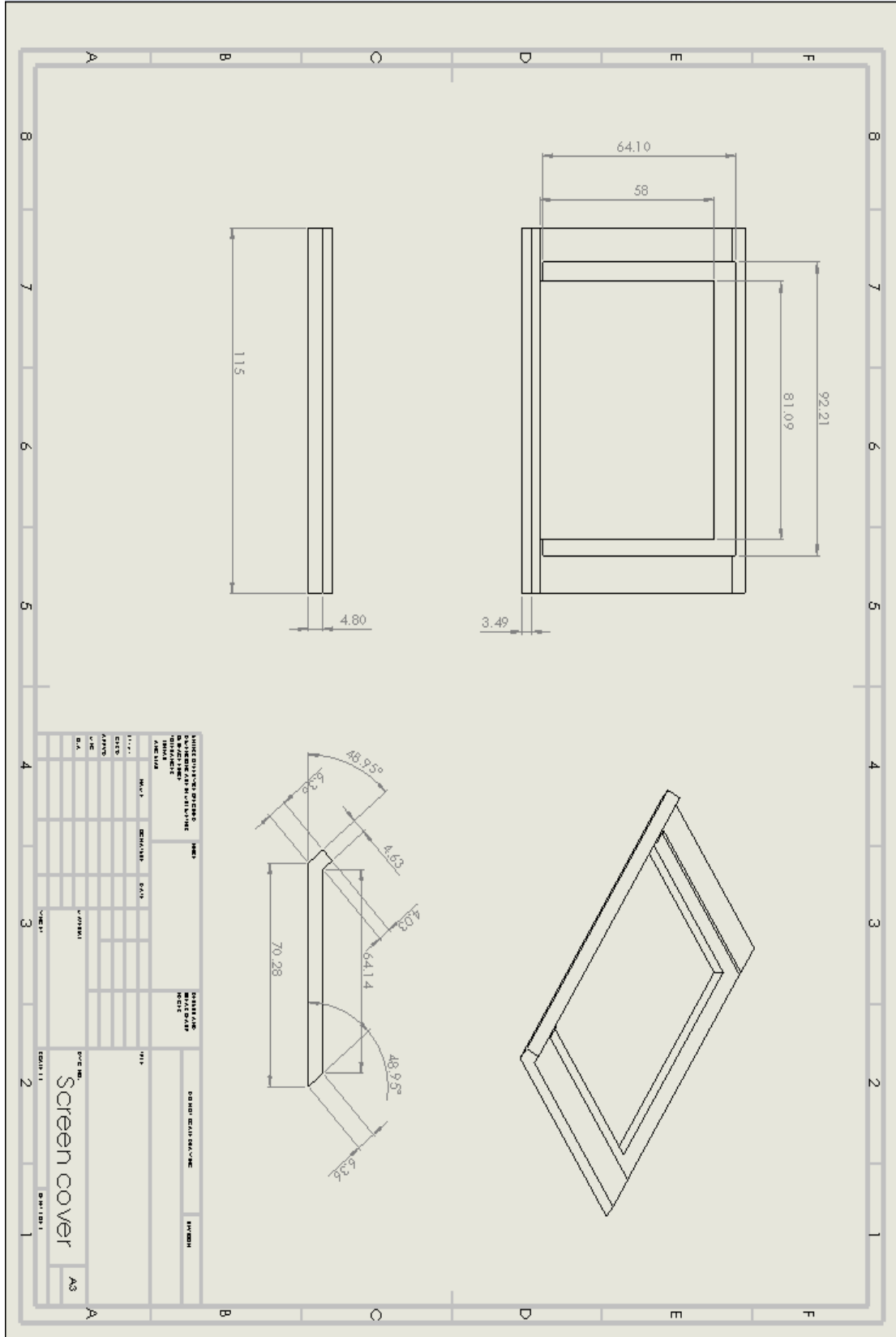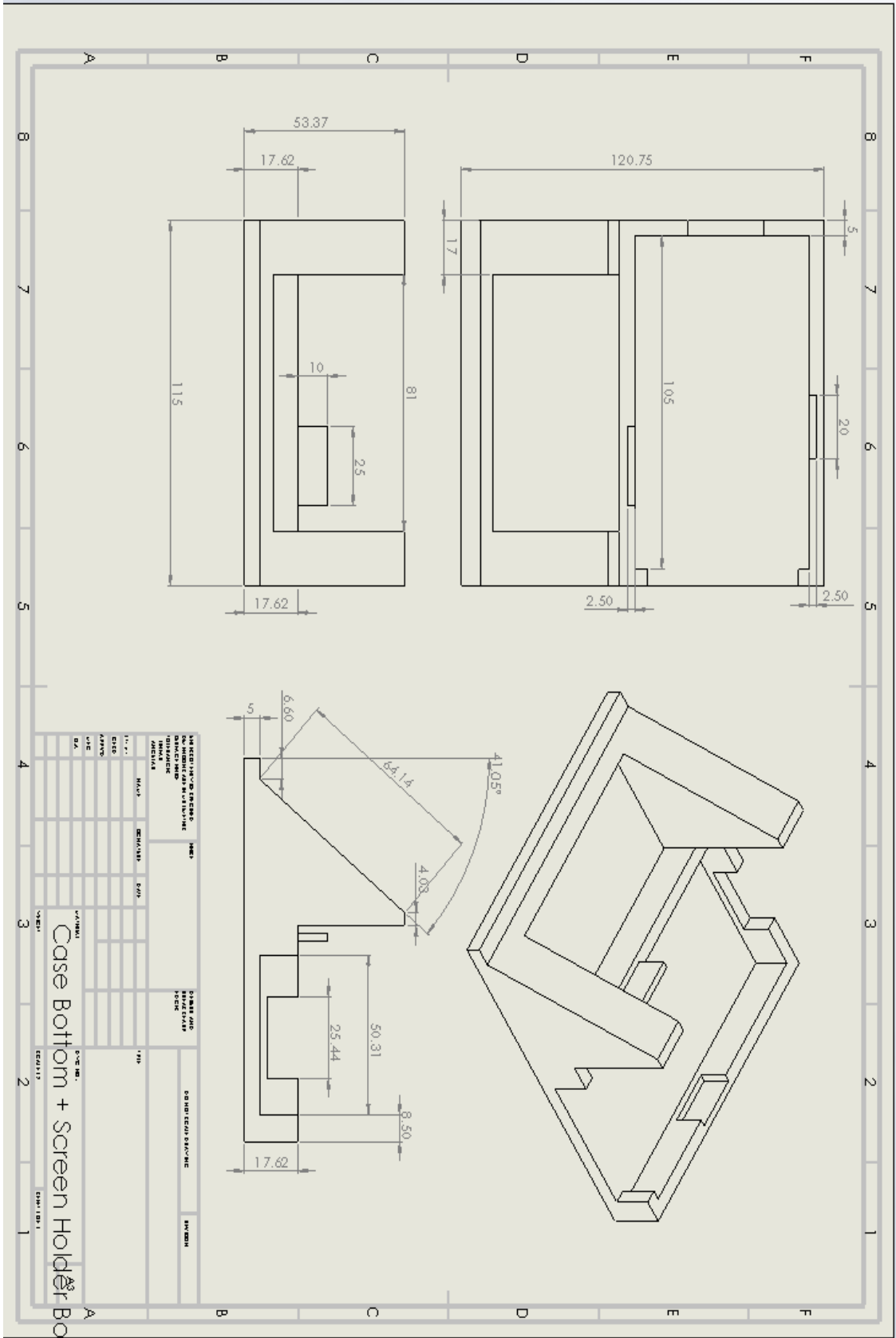
X. APPENDIX IV

Signal conversion from kPa from pressure sensor to cm H_2O in Arduino code:

```
psi = total / numReadings;
psi = psi / bits;          //bits = 1023.0
psi = psi * Vcc;           //Vcc=5.0
psi = psi - offset;        //offset = 2.5
psi = psi / resolution;    //resolution = 0.0002584
psi = psi / GAIN;          //GAIN
psi = psi * 10 * 1.36;     //1.36 is conversion from 1 mmHg = 1.36 cmH2O
psi = psi - calibration;
```

XI.   APPENDIX V

Complete BreathForce system code installed in the Arduino MEGA MCU:

```
#include <SD.h>
#include <Wire.h>
#include <TimeLib.h>
#include <DS1307RTC.h>  //Run RTClib -> SetTime example to re-calibrate time
tmElements_t tm; //Time things
#include <genieArduino.h>
Genie;
#define RESETLINE 4

//psi calculations
const int numReadings = 10;
int readings[numReadings];      // the readings from the analog input
int readIndex = 0;              // the index of the current reading
int total = 0;               // the running total
int average = 0;               // the average
float GAIN = 500.0;
float bits = 1023.0;
float resolution = 0.0002584;
float psi;
float offset = 2.5;
float Vcc = 5.0;
float calibration = 0.00;

int PSIhi;        //to help display on 4d display
int PSIlow;

int MEP = 0;
int MIP = 0;
int TargetMEP;
int TargetMIP;
int TrainingPercent;

int MEParray[105];
int MEPsumarray[95];
int E;
```

```
int MIParray[105];
int MIPsumarray[95];
int I;
long int PSITimer;

//Pressure sensor
int inputPin = A0;

//SD Card
File logfile;
const int chipSelect = 10;
char filename[] = "XXXXXX00.CSV";   //change patient name here. Format: (First thE
letters of first and last name)00.CSV
char filename2[] = "XXXXXX00.CSV";
int x;
int xx;                     //helps display previous file
int xxx;

/////////////////////////////////////////////////////////////////////////////////////////////////////////////////

void setup() {
 //for PSI calculation
 for (int thisReading = 0; thisReading < numReadings; thisReading++) {
  readings[thisReading] = 0;
 }

 //for 4d display to boot up
 Serial.begin(200000);                      // Serial0 @ 150000 (200K) Baud
 genie.Begin(Serial);                    // Use Serial0 for talking to the Genie
Library, and to the 4D Systems display
 genie.AttachEventHandler(myGeniEventHandler);        // Attach the user function
Event Handler for processing events
 pinMode(RESETLINE, OUTPUT);               // Set D4 on Arduino to Output
(4D Arduino Adaptor V2 - Display Reset)
 digitalWrite(RESETLINE, 1);               // Reset the Display via D4
 delay(100);
 digitalWrite(RESETLINE, 0);               // unReset the Display via D4
 delay (3500);                       //let the display start up after the reset (This is
important)
 genie.WriteContrast(1); //1=ScrEn on, 0 = scrEn off
 delay(500);

 //Starting process
 genie.WriteStr(0, "STARTING UP...");
 delay(300);
```

```
  //RTC Boot Check
  genie.WriteStr(0, "Checking RealTimeClock (RTC) status:...");
  delay(300);
  if (RTC.read(tm)) {
   genie.WriteStr(0, "RTC is running.");
   delay(300);
  }
  else {
   if (RTC.chipPresent()) {
     genie.WriteStr(0, "The DS1307 is stopped. Please run the SetTime example
DS1307RTC to recalibrate");
     while (1) {};
   }
   else {
     genie.WriteStr(0, "DS1307 read error! Please check the circuitry/battery.");
     while (1) {};
   }
  }

  //SD Card Check
  genie.WriteStr(0, "Initializing SD card:...");
  delay(300);

  pinMode(53, OUTPUT);
  if (!SD.begin (SPI_FULL_SPEED, chipSelect)) {
   genie.WriteStr(0, "Failed. SD Card not present: 1) Insert SD. 2) Press reset.");
   while (1) {};
  }
  else {
   genie.WriteStr(0, "Success.");
   delay(300);
  }
  genie.WriteStr(0, "Press 'START' when ready...");
}

//////////////////////////////////////////////////////////////////////////////////////////////////////////////////

void loop() {
  genie.DoEvents(); // This calls the library each loop to process the queued responses
from the display
}

//////////////////////////////////////////////////////////////////////////////////////////

void myGeniEventHandler(void) {
  genieFrame Event;
```

```
genie.DequeueEvent(&Event); // Remove the next queued event from the buffer, and
process it below

int EVENT_val = 0;
int slider_val = 0;

EVENT_val = genie.GetEventData(&Event);

//If the cmd received is from a Reported Event (Events triggered from the Events tab of
Workshop4 objects)
if (Event.reportObject.cmd == GENIE_REPORT_EVENT)
{
 //For Calibration on directory screen
 if (Event.reportObject.object == GENIE_OBJ_4DBUTTON)
 {
  if (Event.reportObject.index == 0)
  {
   genie.WriteObject(GENIE_OBJ_FORM, 0, 1);
   delay(1000);
   genie.WriteStr(26, "Calibrating...");
   delay(500);
   genie.WriteObject(GENIE_OBJ_LED_DIGITS, 3, 333);
   delay(1000);
   genie.WriteObject(GENIE_OBJ_LED_DIGITS, 3, 222);
   delay(1000);
   genie.WriteObject(GENIE_OBJ_LED_DIGITS, 3, 111);
   delay(1000);
   long int CALendtime = 0;
   CALendtime = millis() + 1500;
   while (millis() < CALendtime)
   {
    PSIcalculation(); // PSI calculation
    genie.WriteObject(GENIE_OBJ_LED_DIGITS, 3, psi);
   }
   delay(500);
   calibration = psi;
   PSIcalculation();
   genie.WriteObject(GENIE_OBJ_LED_DIGITS, 3, psi);
   delay(500);
   genie.WriteStr(26, "Calibrated.");
   delay(1000);
   genie.WriteStr(26, "Make File Session.");
  }
 }

 //To create new file for session
```

```
if (Event.reportObject.object == GENIE_OBJ_WINBUTTON)
{
 if (Event.reportObject.index == 13)
 {
  genie.WriteStr(18, "Checking INFO file...");
  delay(1000);
  if (Event.reportObject.index == 13)
  {
   MakeNewFile();
  }
 }
}

//Puts in High (e) and Low (i) psi into file. Then displays results of session and
displays previous session
if (Event.reportObject.object == GENIE_OBJ_WINBUTTON)
{
 if (Event.reportObject.index == 10)
 {
  PullFromNewAndOldFiles();
 }
}

//Controls Training% for PSItraining %
if (Event.reportObject.object == GENIE_OBJ_SLIDER)
{
 if (Event.reportObject.index == 0)
 {
  slider_val = genie.GetEventData(&Event);
  genie.WriteObject(GENIE_OBJ_LED_DIGITS, 9, slider_val);
  TrainingPercent = slider_val;
 }
}

//For MIP
if (Event.reportObject.object == GENIE_OBJ_WINBUTTON)
{
 if (Event.reportObject.index == 6)
 {
  TimerMIP(8, 1, 4, 17);
  genie.WriteStr(17, "Done. Proceed to MEP...");
 }
}

//For MEP
if (Event.reportObject.object == GENIE_OBJ_WINBUTTON)
```

```
{
  if (Event.reportObject.index == 14)
  {
    TimerMEP(8, 1, 4, 17);
    genie.WriteStr(17, "Done. See RESULTS...");
  }
}

//PSI Training Timers
if (Event.reportObject.object == GENIE_OBJ_WINBUTTON)
{
  if (Event.reportObject.index == 7)
  {
    PSITimer = 60500;
    genie.WriteObject(GENIE_OBJ_LED_DIGITS, 6, 60);
  }
}
if (Event.reportObject.object == GENIE_OBJ_WINBUTTON)
{
  if (Event.reportObject.index == 12)
  {
    PSITimer = 120500;
    genie.WriteObject(GENIE_OBJ_LED_DIGITS, 6, 120);
  }
}
if (Event.reportObject.object == GENIE_OBJ_WINBUTTON)
{
  if (Event.reportObject.index == 15)
  {
    PSITimer = 180500;
    genie.WriteObject(GENIE_OBJ_LED_DIGITS, 6, 180);
  }
}
if (Event.reportObject.object == GENIE_OBJ_WINBUTTON)
{
  if (Event.reportObject.index == 16)
  {
    PSITimer = 240500;
    genie.WriteObject(GENIE_OBJ_LED_DIGITS, 6, 240);
  }
}
if (Event.reportObject.object == GENIE_OBJ_WINBUTTON)
{
  if (Event.reportObject.index == 17)
  {
    PSITimer = 300500;
```

```
      genie.WriteObject(GENIE_OBJ_LED_DIGITS, 6, 300);
   }
  }

  //Shows results of MEP/MIP on MEP/MIP result
  if (Event.reportObject.object == GENIE_OBJ_WINBUTTON)
  {
   if (Event.reportObject.index == 8)
   {
    delay(500);
    genie.WriteObject(GENIE_OBJ_FORM, 7, 1);
    delay(500);
    genie.WriteStr(12, MEP);
    delay(500);
    TargetMEP = MEP * TrainingPercent / 100;
    genie.WriteStr(13, TargetMEP);
    genie.WriteObject(GENIE_OBJ_LED_DIGITS, 0, TargetMEP);
    delay(500);
    genie.WriteStr(14, abs(MIP));
    delay(500);
    TargetMIP = MIP * TrainingPercent / 100;
    genie.WriteStr(15, abs(TargetMIP));
    genie.WriteObject(GENIE_OBJ_LED_DIGITS, 2, abs(TargetMIP));
   }
  }

  //LUNG TRAINING button takes to Form 3
  if (Event.reportObject.object == GENIE_OBJ_WINBUTTON)
  {
   if (Event.reportObject.index == 5)
   {
    genie.WriteObject(GENIE_OBJ_FORM, 3, 1);
    delay(100);
   }
  }

  //For PSI training scrEn
  if (Event.reportObject.object == GENIE_OBJ_WINBUTTON)
  {
   if (Event.reportObject.index == 11)
   {
    genie.WriteStr(2, "Starting...");
    delay(1000);
    genie.WriteStr(2, "3...");
    delay(1000);
    genie.WriteStr(2, "2...");
```

```
        delay(1000);
        genie.WriteStr(2, "1...");
        delay(1000);
        genie.WriteStr(2, "Now...");
        delay(500);

        logfile = SD.open(filename, FILE_WRITE);
        TimerAndRecordPSI ();
        logfile.close();

        delay(500);
        genie.WriteStr(2, "Done.");
        delay(1000);
        genie.WriteStr(2, "RESULTS to see results \nor START to repeat training.");
      }
    }
  }
}

//PSI calculation///////////////////////////////////////////////////////////////////////
int PSIcalculation() {
  total = total - readings[readIndex];        //psi readings + calculations
  readings[readIndex] = analogRead(inputPin);
  total = total + readings[readIndex];
  readIndex = readIndex + 1;
  if (readIndex >= numReadings) {
    readIndex = 0;
  }
  psi = total / numReadings;
  psi = psi / bits;            //bits = 1023.0
  psi = psi * Vcc;             //Vcc=5.0
  psi = psi - offset;          //offset = 2.5
  psi = psi / resolution;      //resolution = 0.0002584
  psi = psi / GAIN;            //GAIN
  psi = psi * 10 * 1.36;       //1.36 is conversion from 1 mmHg = 1.36 cmH2O
  psi = psi - calibration;
}

//Makes new file name based off INFO file and setsup CVS file///////////////////////////
void MakeNewFile() {

  genie.WriteStr(18, "Patient Info:...");
  delay(1000);
  logfile = SD.open("INFO.TXT");
  filename[0] = logfile.read();
  filename[1] = logfile.read();
```

106

```
filename[2] = logfile.read();
filename[3] = logfile.read();
filename[4] = logfile.read();
filename[5] = logfile.read();
filename2[0] = filename[0];
filename2[1] = filename[1];
filename2[2] = filename[2];
filename2[3] = filename[3];
filename2[4] = filename[4];
filename2[5] = filename[5];
logfile.close();
genie.WriteStr(18, filename);
delay(1000);

genie.WriteStr(18, "Making new file:...");
for (uint8_t i = 1; i < 100; i++) {
 filename[6] = i / 10 + '0';
 filename[7] = i % 10 + '0';
 if (! SD.exists(filename)) {
  logfile = SD.open(filename, FILE_WRITE);
  logfile.print(",,");
  logfile.print("Date(D/M/Y)/Time:,");          //A1:B1
  logfile.print(tm.Day);
  logfile.print("/");
  logfile.print(tm.Month);
  logfile.print("/");
  logfile.print(tmYearToCalendar(tm.Year));
  logfile.print(",");
  logfile.print(tm.Hour);
  logfile.print(":");
  logfile.print(tm.Minute);
  logfile.print(":");
  logfile.println(tm.Second);
  logfile.print(",,");
  logfile.print("Highest Exp. PSI:");           //A2:B2
  logfile.print(",");
  logfile.print("       ");
  logfile.print(",");
  logfile.println("cm H2O");
  logfile.print(",,");
  logfile.print("Lowest Insp. PSI:");           //A3:B3
  logfile.print(",");
  logfile.print("       ");
  logfile.print(",");
  logfile.println("cm H2O");
  logfile.print(",,");
```

```
        logfile.print("MEP:");              //A2:B2
        logfile.print(",");
        logfile.print("       ");
        logfile.print(",");
        logfile.println("cm H2O");
        logfile.print(",,");
        logfile.print("MIP:");              //A3:B3
        logfile.print(",");
        logfile.print("       ");
        logfile.print(",");
        logfile.println("cm H2O");
        logfile.print(",,");
        logfile.print("Training %");
        logfile.print(",");
        logfile.print("     ");
        logfile.print(",");
        logfile.println("%");
        logfile.print(",,");
        logfile.print("# of points:");             //A4:B4
        logfile.print(",");
        logfile.print( "=COUNT(B:B)");
        logfile.print(",");
        logfile.println("points");
        logfile.print(",,");
        logfile.print("Time Duration");            //A5:B5
        logfile.print(",");
        logfile.print("=(MAX(A:A)-MIN(A:A))/1000");
        logfile.print(",");
        logfile.println("seconds");
        logfile.print(",,");
        logfile.print("Points per second:");        //A6:B6
        logfile.print(",");
        logfile.print("=D7/D8");
        logfile.print(",");
        logfile.println("points/s");
        logfile.print("Time(ms)");     //A8:B8
        logfile.print(",");
        logfile.println("cm H2O");
        logfile.close();
        break; // leave the loop!;
      }
    }
    delay(1000);
    genie.WriteStr(18, filename);
    delay(1000);
```

```
  genie.WriteStr(18, "Checking old file...");
  delay(1000);
  filename2[6] = filename[6];
  filename2[7] = filename[7];
  xx = filename2[6] - '0';   //Renames filename to go back 1 previous session to display on
result scrEn
  xxx = filename2[7] - '0';
  x = (xx * 10) + xxx - 1;
  filename2[6] = x / 10 + '0';
  filename2[7] = x % 10 + '0';
  logfile = SD.open(filename2, FILE_READ);
  logfile.seek(20);
  logfile.readStringUntil(',');
  logfile.readStringUntil(',');
  logfile.readStringUntil(',');
  logfile.readStringUntil(',');
  logfile.readStringUntil(',');
  logfile.readStringUntil(',');
  logfile.readStringUntil(',');
  logfile.readStringUntil(',');
  logfile.readStringUntil(',');
  logfile.readStringUntil(',');
  logfile.readStringUntil(',');
  logfile.readStringUntil(',');
  MEP = logfile.readStringUntil(',').toInt();
  logfile.readStringUntil(',');
  logfile.readStringUntil(',');
  logfile.readStringUntil(',');
  MIP = logfile.readStringUntil(',').toInt();
  logfile.readStringUntil(',');
  logfile.readStringUntil(',');
  logfile.readStringUntil(',');
  TrainingPercent = logfile.readStringUntil(',').toInt();
  logfile.close();

  genie.WriteObject(GENIE_OBJ_LED_DIGITS, 12, MEP);    //MEP
  genie.WriteObject(GENIE_OBJ_LED_DIGITS, 13, abs(MIP));    //MIP
  genie.WriteObject(GENIE_OBJ_LED_DIGITS, 10, TrainingPercent);    //% Training

  genie.WriteStr(18, filename2);
  delay(1000);

  genie.WriteStr(18, "Proceed to MEP/MIP.");
  delay(100);
}
```

```
//MIP calculating/////////////////////////////////////
void TimerMIP (int a, int b, int c, int d) {
 genie.WriteStr(d, "Starting...");
 delay(1000);
 genie.WriteStr(d, "3...");
 delay(1000);
 genie.WriteStr(d, "2...");
 delay(1000);
 genie.WriteStr(d, "1...");
 delay(1000);
 genie.WriteStr(d, "Now.");
 delay(500);

 long int MIPendtime = 0;
 I = 0;
 MIP = 0;
 MIPendtime = millis() + 10500;
 while (millis() < MIPendtime)
 {
  PSIcalculation();
  genie.WriteObject(GENIE_OBJ_LED_DIGITS, a, (MIPendtime - millis()) / 1000);
  genie.WriteObject(GENIE_OBJ_SCOPE, b, abs(psi));
  genie.WriteObject(GENIE_OBJ_LED_DIGITS, c, abs(psi));
  MIParray[I] = psi;
  I++;
 }
 for (I = 0; I < 96; I++)
 {
  MIPsumarray[I] = (MIParray[I] + MIParray[I + 1] + MIParray[I + 2] + MIParray[I +
3] + MIParray[I + 4] + MIParray[I + 5] + MIParray[I + 6] + MIParray[I + 7] +
MIParray[I + 8] + MIParray[I + 9]) / 10;
  delay(5);
 }
 int idy;
 for (byte idy = 0; idy != 96; idy++)
 {
  if (MIPsumarray[idy] < MIP) {
   MIP = min(MIPsumarray[idy], MIP);
  }
  delay(5);
 }
}
//MEP calculating/////////////////////////////////////////////////////
void TimerMEP (int a, int b, int c, int d) {
 genie.WriteStr(d, "Starting...");
 delay(1000);
```

110

```
  genie.WriteStr(d, "3...");
  delay(1000);
  genie.WriteStr(d, "2...");
  delay(1000);
  genie.WriteStr(d, "1...");
  delay(1000);
  genie.WriteStr(d, "Now.");
  delay(500);

  long int MEPendtime = 0;
  E = 0;
  MEP = 0;
  MEPendtime = millis() + 10500;
  while (millis() < MEPendtime)
  {
   PSIcalculation();
   genie.WriteObject(GENIE_OBJ_LED_DIGITS, a, (MEPendtime - millis()) / 1000);
   genie.WriteObject(GENIE_OBJ_SCOPE, b, abs(psi));
   genie.WriteObject(GENIE_OBJ_LED_DIGITS, c, abs(psi));
   MEParray[E] = psi;
   E++;
  }
  for (E = 0; E < 96; E++)
  {
   MEPsumarray[E] = (MEParray[E] + MEParray[E + 1] + MEParray[E + 2] +
MEParray[E + 3] + MEParray[E + 4] + MEParray[E + 5] + MEParray[E + 6] +
MEParray[E + 7] + MEParray[E + 8] + MEParray[E + 9]) / 10;
   delay(5);
  }
  int idx;
  for (byte idx = 0; idx != 96; idx++)
  {
   if (MEPsumarray[idx] > MEP) {
    MEP = max(MEPsumarray[idx], MEP);
   }
   delay(5);
  }
}
//Timer and PSI session calculating//////////////////////////////////////////////////////////////////
void TimerAndRecordPSI () {
 long int Pendtime = 0;
 Pendtime = millis() + PSITimer;
 while (millis() < Pendtime)
 {
  PSIcalculation();
  genie.WriteObject(GENIE_OBJ_LED_DIGITS, 6, (Pendtime - millis()) / 1000);
```

```
     genie.WriteObject(GENIE_OBJ_LED_DIGITS, 1, abs(psi));
     genie.WriteObject(GENIE_OBJ_SLIDER, 1, psi + 40);
    if (psi > PSIhi) {
     PSIhi = psi;
    }
    if (psi < PSIlow) {
     PSIlow = psi;
    }
    if (psi > (TargetMEP * 0.90)) {
     genie.WriteObject(GENIE_OBJ_USER_LED, 0, 1);
     //genie.WriteObject(GENIE_OBJ_SOUND, 0x00, 0);
     //genie.WriteObject(GENIE_OBJ_SOUND, 0x01, 100);
    }
    else {
     genie.WriteObject(GENIE_OBJ_USER_LED, 0, 0);
     //genie.WriteObject(GENIE_OBJ_SOUND, 0x04, 0);
    };

    if (psi < (TargetMIP * 0.90)) {
     genie.WriteObject(GENIE_OBJ_USER_LED, 1, 1);
     //genie.WriteObject(GENIE_OBJ_SOUND, 0x00, 0);
     //genie.WriteObject(GENIE_OBJ_SOUND, 0x01, 100);
    }
    else {
     genie.WriteObject(GENIE_OBJ_USER_LED, 1, 0);
     //genie.WriteObject(GENIE_OBJ_SOUND, 0x04, 0);
    };
    logfile.print (millis());
    logfile.print (",");
    logfile.println(psi);
  }
}

//Pulls information from old file to display on session results scrEn////////////////
void PullFromNewAndOldFiles() {
  delay(500);
  logfile = SD.open(filename, FILE_WRITE);
  logfile.seek(20);
  logfile.readStringUntil(',');
  logfile.readStringUntil(',');
  logfile.readStringUntil(',');
  logfile.readStringUntil(',');
  logfile.print(PSIhi);
  logfile.readStringUntil(',');
  logfile.readStringUntil(',');
  logfile.readStringUntil(',');
```

```
logfile.readStringUntil(',');
logfile.print(PSIlow);
logfile.readStringUntil(',');
logfile.readStringUntil(',');
logfile.readStringUntil(',');
logfile.readStringUntil(',');
logfile.print(MEP);
logfile.readStringUntil(',');
logfile.readStringUntil(',');
logfile.readStringUntil(',');
logfile.readStringUntil(',');
logfile.print(MIP);
logfile.readStringUntil(',');
logfile.readStringUntil(',');
logfile.readStringUntil(',');
logfile.readStringUntil(',');
logfile.print(TrainingPercent);
logfile.close();

genie.WriteObject(GENIE_OBJ_FORM, 4, 1);
delay(1000);
genie.WriteStr(5, filename);
delay(500);

logfile = SD.open(filename, FILE_READ);
logfile.seek(20);
genie.WriteStr(1, logfile.readStringUntil(','));    //DATE
delay(500);
genie.WriteStr(4, logfile.readStringUntil(','));    //TIME
delay(500);
logfile.readStringUntil(',');
logfile.readStringUntil(',');
genie.WriteStr(6, logfile.readStringUntil(','));    //EXHALE
delay(500);
logfile.readStringUntil(',');
logfile.readStringUntil(',');
logfile.readStringUntil(',');
genie.WriteStr(3, logfile.readStringUntil(','));    //INHALE
delay(500);
logfile.readStringUntil(',');
logfile.readStringUntil(',');
logfile.readStringUntil(',');
genie.WriteStr(19, logfile.readStringUntil(','));    //MEP
delay(500);
logfile.readStringUntil(',');
logfile.readStringUntil(',');
```

```
logfile.readStringUntil(',');
genie.WriteStr(21, logfile.readStringUntil(','));    //MIP
delay(500);
logfile.readStringUntil(',');
logfile.readStringUntil(',');
logfile.readStringUntil(',');
genie.WriteStr(23, logfile.readStringUntil(','));    //% Training
delay(500);
logfile.close();

xx = filename[6] - '0';   //Renames filename to go back 1 previous session to display on
result scrEn
xxx = filename[7] - '0';
x = (xx * 10) + xxx - 1;
filename[6] = x / 10 + '0';
filename[7] = x % 10 + '0';
genie.WriteStr(7, filename);
delay(500);

logfile = SD.open(filename, FILE_READ);
logfile.seek(20);
genie.WriteStr(8, logfile.readStringUntil(','));
delay(500);
genie.WriteStr(9, logfile.readStringUntil(','));
delay(500);
logfile.readStringUntil(',');
logfile.readStringUntil(',');
genie.WriteStr(10, logfile.readStringUntil(','));
delay(500);
logfile.readStringUntil(',');
logfile.readStringUntil(',');
logfile.readStringUntil(',');
genie.WriteStr(11, logfile.readStringUntil(','));
delay(500);
logfile.readStringUntil(',');
logfile.readStringUntil(',');
logfile.readStringUntil(',');
genie.WriteStr(20, logfile.readStringUntil(','));    //MEP
delay(500);
logfile.readStringUntil(',');
logfile.readStringUntil(',');
logfile.readStringUntil(',');
genie.WriteStr(22, logfile.readStringUntil(','));    //MIP
delay(500);
logfile.readStringUntil(',');
logfile.readStringUntil(',');
```

```
  logfile.readStringUntil(',');
  genie.WriteStr(24, logfile.readStringUntil(','));    //% Training
  delay(500);
  logfile.close();
}
```

XII.   APPENDIX VI

Comparison of Delta-Cal to raw data BreathForce pressure measurements over a span of 20 minutes.

Delta-Cal (mmHg)	BreathForce (mmHg)				
	0:00	5:00	10:00	15:00	20:00
-50	-49	-48	-48	-48	-48
-40	-38	-38	-38	-38	-38
-30	-29	-29	-29	-29	-29
-20	-20	-19	-19	-19	-19
-10	-10	-10	-10	-10	-10
0	0	0	0	0	0
10	10	10	10	10	10
20	20	20	19	19	19
30	29	29	29	29	29
40	39	38	38	38	38
50	49	49	49	49	49

XIII. APPENDIX VII

Bill of Materials

Component	Price	Manufacturer
Positive Expiatory Pressure device	$25.00	Phillips Respironics
Inspiratory Muscle Trainer device	$23.00	Phillips Respironics
Ventilator monitoring adaptor circuit	$5.00	AirLife
Single Use Pressure Sensors	$30.00	PendoTech
Arduino MEGA 2560 REV 3	$35.75	Arduino
Custom circuit board	$5.00	China-company
HC-05 Arduino Bluetooth Receiver	$10.95	NewZoll
Electrical components	$5.00	Various
Micro-SD and SD card	$5.00	Various
XD-05 data logging shield	$3.95	Arduino
uLCD-32 PTU Touchscreen Display	$79.00	4D Systems
Custom 3D printed case	$6.00	Makerbot
2x 18650 3.7 3000mAh Li-Ion Batteries	$8.99	EBL
Total Cost	$242.64	