WORCESTER POLYTECHNIC INSTITUTE

Flexible Sensor for Measurement of Skin Pressure and Temperature for the Prevention of Pressure Ulcers

by

Matthew D. Crivello

A thesis submitted in partial fulfillment for the
degree of Master of Science
in
Electrical and Computer Engineering

October 2016

APPROVED:

Professor John McNeill

Professor Yitzhak Mendelson

Professor Xinming Huang

Professor Shamsur R Mazumder

WORCESTER POLYTECHNIC INSTITUTE

Abstract

Electrical and Computer Engineering

Master of Science

by Matthew D. Crivello

With the prolonged lifespan of the average person, the number of hospital stays have increased. Currently, pressure ulcers are one of the most severe complications associated with prolonged hospital stay. The protocol in today's hospital is to rotate bedridden patients once every two hours to prevent pressure ulcers. This puts a strain on attending nurses as the risk of a pressure ulcer for a patient is not universal and therefore, a universal preventative protocol is not the most effective solution.

This thesis describes the circuit design and physical implementation of a device to address the issue of pressure ulcers. The device has the form factor of a patch to be placed on specific, at risk areas of the human body. The device was designed and prototyped first on a rigid structure and then on a flexible printed circuit board substrate. A calibration procedure was developed to reduce part to part variability inherent to the pressure sensor. The resistance measurement was achieved through a novel approach including the use of a timer removing the need for an analog-to-digital converter. A seven hour experiment was conducted with live, animal subjects to measure the pressure and temperature of at risk areas of the body. The results of the experiment successfully prove the fundamental approach outlined in this thesis and justify continued research and refinement into the product design.

Acknowledgements

It has been a great honor to work under Professor McNeill. His technical insights and intuition about analog circuits helped guide me through some of the more challenging design aspects and made me realize how little I actually know. His occasional pep-talks helped me realize that in research, failure is a learning process.

I would like to thank Yitzhak Mendelson Ph.D., Raymound Dunn M.D., and Kelli Hickle M.D. They're help with the medical aspects of pressure ulcers proved to be instrumental in the device design. Raymound Dunn M.D. specifically was vital to this thesis and without him, the final live animal experiment would not be possible. Bill Appleyard provided great insights into the physical construction of the device and provided assistance during critical times of the project. Additionally, I would like to thank Devdip Sen for helping me through this entire design and testing process. If Devdip moves on with this project, I wish him the best of luck.

Finally, I would like to thank my family without which this thesis would not be possible. They're support and encouragement was greatly appreciated especially through the more stressful moments. I can't express my gratitude and appreciation for all the times you have been there for me. Thank you.

Contents

List of Figures

List of Tables

Dedicated to My Friends and Family

Chapter 1

Introduction

Traditional protocols in the medical community for lowering pressure ulcer risk is to rotate long term bed ridden patients every 2 hours [1]. An observational study conducted in 1973 [27] is the medical foundation for the protocol without specifically taking into account any of the many risk factors that would put a patient at an increased risk of developing a pressure ulcer. Many competitors in this field have opted to determine the risk of a pressure ulcer through full body pressure mapping [15], through algorithms based on body position [14], and through actively reducing the overall pressure exerted upon the body [30]. Many of these solutions are very expensive or do not take into account many of the other factors affecting a patients' risk of pressure ulcer development (such as height, weight, and age). Additionally, these solutions are aimed directly to patients who are bedridden and ignore patients who may develop pressure ulcers in day to day life.

The overall goal of this work is to design a novel device to be attached to the skin for measuring pressure in a localized region. That data will then be streamed to another central, device for pressure ulcer risk analysis. Typical devices in this area work to prevent pressure ulcers in a hospital setting. This device aims to prevent pressure ulcers in both a hospital setting as well as everyday life through a robust, skillful design.

Chapter 2

Background

2.1 Pressure Ulcer Characterization

Pressure ulcers (more commonly known as bedsores), are a major health problem in the United States. Approximately 3 million patients are affected by pressure sores each year in the United States alone [1]. Pressure Ulcers incur an average charge of $37,800 per stay [2]. With aging populations around the word and increasing nursing shortages, it is likely that that the occurrence of pressure ulcers will continue to rise.

2.1.1 Causes

The fundamental cause of pressure ulcers is the inability of capillaries to supply skin and subcutaneous tissue with adequate perfusion causing tissue necrosis. The pressure within capillaries is known to range from 20 to 40 mmHg with an accepted average of

32 mmHg [3]. With this in mind, keeping external pressure less than 32 mmHg should be able to prevent the formation of pressure ulcers. Some risk factors for pressure ulcers include age, current smoking history, low body mass index, and impaired mobility.

2.1.2 Consequences

A patient with a severe pressure ulcer is comparable to that of a burn victim. Just as with a burn victim, the time frame for healing is almost unknown as it is different for all patients and depending on the severity, could require a skin flap [4]. With many patients, the quality of life for that patient decreases significantly. Many patients feel that the pressure ulcer owns their life. Patients need a special bed, have limited sleeping positions, require frequent dressing changes, home health care, have significant odor, drainage, and clothing limitations just to name a few of the hardships faced by patients [4]. Many patients also describe the pain from a pressure ulcer as excruciating even at rest [4].

2.1.3 Prevention

Because of limited pressure ulcer prevention techniques, pressure ulcers have become an accepted part of bedridden patients [4]. The primary technique for prevention is to move patients every two hours whether they are at risk of a pressure ulcer or not. This technique was determined through an observational study conducted at the Rancho Los Amigos Hospital, Downey, California in 1973 [27]. While this technique has proven to be effective in preventing pressure ulcers, it does not take into account age, body mass index, or any other proven contributing risk factors. An alternate solution is to implement a bed that increases air flow around the body and reduces the pressure on the skin as a patient rests.

2.2 Device Characteristics

There are many characteristics to consider when designing a product to measure the pressure applied to the skin and the temperature of the skin. One of the main considerations is the pressure sensor itself. It must have a low profile in order to not bias measured pressure and it must be able to measure a range of pressures. Another consideration, is the way in which that that sensor measurement is actually read. Once gathered, the data must be analyzed and recorded. Finally, the power requirements of the device must be addressed. All of these design constraints must be given proper attention to achieve a successful device.

2.2.1 Force Sensor

There are many options for force sensors on the market. Many manufactures emphasize accuracy and precision over footprint. For most applications, this is very logical and beneficial however, this is not the case for the design of this device. Because the device will be attached directly to the skin, the device must be slim in order to get unbiased measurements when attached to the patient's skin. This is not to say that accuracy and precision are not important to the design; it is simply necessary to place more significance on the device footprint rather the precision and accuracy. Along with the footprint, the measurable pressures of the device must be within a certain range. If pressure exceeds a threshold of approximately 32 mmHg [4], capillary blood flow can be reduced or stopped, denying oxygen to tissue in the area. Over time, this will lead to tissue necrosis in the affected area. Pressures exceeding this amount only increase the risk of pressure ulcers. The amount of time that pressure is applied to the skin plays a vital role in the determination of the risk of a pressure ulcer. In other words, various pressure levels applied for various amounts of time will lead to different risk levels in the formation of a pressure ulcer. With this in mind, the sensor must be able to measure pressures lower than 32 mmHg as well as pressures reasonably higher.

2.2.2 555 Timer Resistance Measurement Theory

Many force sensors work as a variable resistor. This varying resistance value must be interpreted by a micro-controller leading to the need to convert this resistance value into a digital value. The most common measurement technique is to put the sensor in a voltage divider with a known resistor and a known power source to measure the change in voltage across the sensor with an analog-to-digital convert (ADC). The resolution of the ADC is directly proportional to the price of the component (the higher the resolution, the higher the price). An ADC converts the measured voltage into a bit value that is dependent on the ADC (8-bit, 10-bit, 16-bit, etc.). This can then be understood by the micro-controller and further manipulated to get meaningful results.

Another method can be achieved through the use of a timer such as the 555 timer. A 555 timer utilizes the discharge and recharge rate of two resistors and a capacitor to trigger a flip-flop logic circuit. This creates a standing square wave where the frequency and duty cycle are dependent on the resistance and capacitor values. Figure 2.1 shows the circuit diagram for a 555 timer in the astable configuration. Either resistor can be replaced with the force sensor to affect the duty cycle in a measurable way.

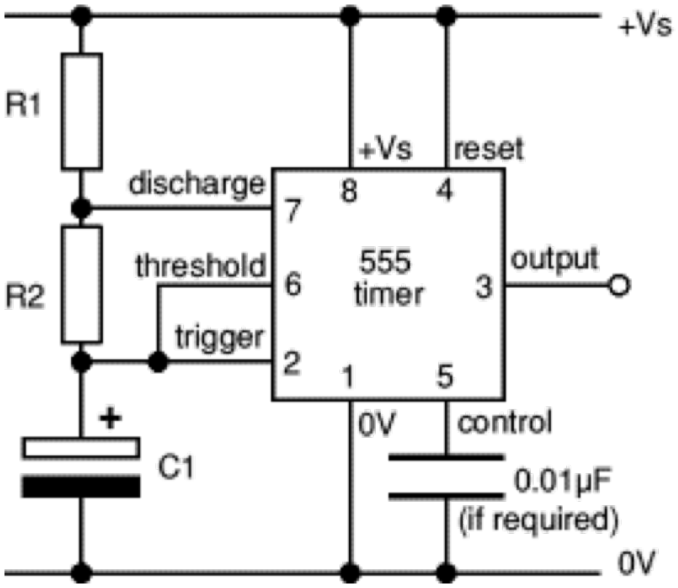


FIGURE 2.1: 555 Astable Circuit Diagram [5]

The frequency of the output can be determined from equation 2.1 [6]:

$$f = \frac{1}{ln(2)(R_1 + 2R_2)C} \tag{2.1}$$

Looking at equation 2.1, it is important to notice that the frequency of the output is directly dependent on R_1, R_2, and C. This means that the precision with which R_1 and C are known will directly affect the precision of the determination of R_1 from a frequency measurement. It is possible to improve the precision of the R_1 measurement if the dependence upon C is removed. This can be done by looking at the equations for the high and low time (T_{high} and T_{low} respectively) [6]:

$$T_{high} = ln(2)(R_1 + R_2)C \tag{2.2}$$

$$T_{low} = ln(2)R_2C \tag{2.3}$$

Both equations 2.2 and 2.3 are directly related to the value of C. Dividing equation 2.2 by equation 2.3 results in the duty cycle of the signal (δ) removing the dependence on C as seen in equation 2.4 [6]:

$$\delta = \frac{R_1 + R_2}{R_1 + 2R_2} \tag{2.4}$$

This removes the dependence of C from the measurement technique as well. Further implications are the that the tolerance of the capacitance value of C can be more lenient thereby reducing cost and a possible source of error. The tolerance of value of R_2 becomes very important but advances in resistor manufacturing have made very low tolerance resistors very inexpensive.

2.2.3 Wireless Transmission

Wireless transmission of data is a very mature field. With the advances in the last few decades to push for going completely wireless, many options have developed for various applications. There are five main protocols that are applicable to this device: WiFi, Bluetooth, ZigBee, ANT, and RF.

Of the four protocols mentioned, WiFi offers the highest bandwidth. The primary application for WiFi has been for computers to access the internet for browsing, streaming video, streaming music, etc. A large advantage of WiFi is its vast adoption. WiFi converge has become expected in all public spaces and private homes. Many IoT (Internet of Things) devices are taking advantage of this established network and are pushing live data to the cloud for further analysis and monitoring. WiFi offers a lot of advantages but also has drawbacks which include the relatively high power consumption and the high computational power required in order to handle the high bandwidth protocols and handshakes.

Bluetooth offers another option that has become very mature with the advent of the cell phone peripherals. Bluetooth is a peer-to-peer connection strategy and works primarily on a local network level. Usually a master device (i.e. cell phone) communicates with slave devices (i.e. headset, speakers, smart watches, etc.). This provides small, manageable networks that perform very specific tasks that do not need the robust infrastructure provided through WiFi. The range of Bluetooth is considerably smaller than WiFi. There are 3 classes of Bluetooth transmitters with various power consumptions and ranges. Class 1 devices consume 100mW of power and can reach a range of 100 meters, class 2 devices consume 2.5mW of power and have a range of 10 meters, and class 3 devices consume 1mW of power and have a range of less than 10 meters [7]. Class 1 devices are reserved for those which have plentiful power sources (laptops, desktops, cars, etc.) where as class 2 and 3 devices contain a limited power supply. Bluetooth also offers a protocol known as Bluetooth LE (Low Energy) that is designed specifically for IoT devices [8]. This offers a communication protocol that is specifically designed for low power devices such as the one that is the focus of this thesis.

Like Bluetooth, Zigbee offers a protocol that has been adopted by many low-power devices in the industry. Of the many uses for ZigBee, the primary applications involve sensing and monitoring. Currently, major companies such as Philips and Samsung are utilizing ZigBee in their IoT products such as smart home lighting [28]. There are many configurations in the way ZigBee can be implemented on a device. The three specifications are known as ZigBee Pro, ZigBee RF4CE, and ZigBee IP [28]. ZigBee Pro offers a one-way communication between two devices that is proven to be highly reliable and uses extremely low power. ZigBee RF4CE offers two-way device to device communication that is low power and requires a small amount of memory for operation. ZigBee IP is a full mesh network standard that allows many ZigBee devices to connect into a network utilizing IPv6 network addressing [28].

ANT is a low-power protocol that was developed for low-power devices. The primary use of ANT is sports and fitness devices. Traditionally, ANT radios are treated as black boxes with minimal development necessary in order to make the device operational. ANT does not have the same market reach at Bluetooth LE but does include many of the benefits seen with Bluetooth LE. ANT+ is an updated protocol for the devices that guarantees interoperability between devices. This new protocol does require more hardware and increases the hardware costs slightly [29].

A fifth option for wireless data transmission is RF. This is a very general option that encompasses the remainder of wireless options available but the flexibility of the option makes it important to investigate. RF offers the option to develop a low overhead protocol utilizing the free spectrum utilized by both WiFi and Bluetooth (2.4 GHz and 5 GHz). Both WiFi and Bluetooth are very robust, mature protocols that allow for multiple, universal connections as well as high bandwidth. For the device being discussed in this paper, this could prove to be overly complex and unnecessary for the application. An RF solution would require the implementation of a communication protocol as well as a custom hardware solution. This has the potential to result in an extremely efficient solution optimized specifically for our device.

The cost implementing each of these different protocols is relatively the same. Many of the components vary only slightly (e.g capacitor size, antenna length, etc.) with the primary variation coming from the processor used to handle the protocol. With this in mind, Table 2.1 shows the cost of implementing a Bluetooth LE device [29].

Component	**Quantity**	**Cost ($)**
Battery	1	0.325
Antenna	1	0 (Printed Antenna)
EEPROM	1	0.89
Decoupling Cap	6	0.002
Signaling Cap	5	0.002
Resistor	4	0.0001
Crystal	2	0.243
Bluetooth Low Energy IC	1	$\tilde{1}$
Total		**$2.72**

Table 2.1: A BOM or a Bluetooth Low Energy Device [29]

Various aspects of each protocol must be compared in order to determine which protocol is the right fit for this application. Some important qualities are Power, Range, bytes per second, energy per bit, and Peak Current. All of these qualities greatly effect the efficiency of the devices as well as the capabilities of each protocol. Table 2.2 compares these various characteristics.

	Power (mW)	**Range (m)**	**byte/second (bps)**	**energy/bit (μJ/bit)**	**Peak Current (mA)**
WiFi	210	150	5,000,000	0.00525	116
Bluetooth LE	0.147	280	120	0.153	12.5
ZigBee	36	100	192	185.9	40
ANT	0.183	30	32	0.71	17

Table 2.2: Comparing Different Characteristics of the Various Protocols [29]

This data gives insight into the strengths and weaknesses of the various protocols. WiFi has the highest bit rate and the lowest energy per bit but the highest peak current. This shows the priority WiFi places on throughput rather and energy per bit rather than the peak power used as many devices utilizing WiFi will have high data requirements and would be able to provide those power levels. Bluetooth LE has the highest range, the lowest peak current, and lowest power but does not have the highest bit rate nor the lowest energy/bit. Bluetooth LE shows the emphasis placed on total power used rather than throughput. This protocol was designed for a low power device that does not require high data rates but does have a limited power source. An RF solution was not included in Table 2.2 as the solution would be custom and the various characteristics cannot be determined without a basis design.

2.2.4 Power

Power is an extremely important aspect to any mobile device and is often seen as an afterthought in design. When designing any device, the primary goal in regards to power is to reduce power consumption as much as possible. This is especially prevalent in the design of this device because it must be worn on the skin and completely wireless. This leads to two primary options: a coin cell battery and a combination film battery with wireless power harvesting. As revisions to prototypes are made, the specific power requirements will be determined leading to a more concrete decision on the power system design.

A coin cell battery design lends to a more traditional power design. A coin cell has limiting power capacities which increase with larger coin cell batteries. This is a problem in the design of the device because it would add extra thickness and size to the device which would make it more uncomfortable for the patient to wear and could potentially cause adverse effects on the pressure measurements. It is very important to keep the components of the device as thin as possible in order to achieve the most accurate pressure measurements. With these drawbacks in mind, coin cells do offer the highest battery capacity for our device and can also be found in eco-friendly solutions [9]. This would allow for a disposable, eco-friendly design that would have ample amounts of power.

An alternative to a traditional battery approach would be to incorporate a thin film battery with wireless power harvesting technology. A thin film battery is extremely thin and made from eco-friendly material allowing for a disposable design like the coin cell approach. The main drawback to thin film batteries is capacity. Battery capacities are on the order of 10s of micro amp-hours which does not leave much room for long term wireless transmission [10]. These batteries do have the advantage of being rechargeable allowing them to be possibly utilized through a wireless power harvesting solution. Wireless power harvesting technology is widely incorporated with RFID technology and could be adapted for the design of this device. An antenna is used to take the energy from an electromagnetic signal and that energy is then stored for later use or used for a very small application [11]. This could be utilized in the device to charge the thin-film battery which would later be used to periodically send data to a base station for further analysis. This could also potentially increase the lifetime of the device in that the power source becomes essentially external. A custom base station that would flood the spectrum with wireless power would have to be developed. This has the potential to effect some of the other hardware found in a patients hospital room and would have to be further investigated.

2.2.5 Software

Part of the design of the device is the software providing the device logic which has many aspects to consider. The on board device software must be minimal in order to reduce computation time and thereby reducing power consumption. Another factor is reducing the amount of time necessary for transmitting data wirelessly. The transmission of data will be one of the most power intensive processes conducted by the device. A balance will have to be made between on board computation and off board transmission. The more that can be computed by the on board micro-controller, the less data that must be transmitted. This also leads to longer computation times which may be a problem because this must all be completed before the next measurement to ensure that no data is missed.

Another aspect of the software is the software off board from the primary device. This software will contain the primary algorithm for the determination for the risk of a pressure ulcer. Many parameters will be factored into the algorithm. Age, time of applied pressure, amount of applied pressure, region of the body where pressure is applied, and temperature are some of the primary contributing factors to the formation of pressure ulcers [12]. Through many clinical trials, the role in which each of these factors play in the cause of the of a pressure ulcer will be determined. The off board software will have to have a way to incorporate all of these key factors into an algorithm for determining the risk of a pressure ulcer in the patient. Another aspect to the off board software is the way in which the data is presented to doctors and other pertinent staff. The data may be kept on a server and made accessible through the internet from many various devices. A software platform such as ThingWorx [13] would allow for a robust system as it is specifically designed for IoT devices.

2.2.6 Micro-Controller

To interpret the electrical signals from the force sensor, a micro-controller is needed. The micro-controller must take the data from either an ADC or the 555 timer approach and pull out the resistance measurement that corresponds to the applied force on the force sensor. The data must then be transmitted to a central location or must be stored for a chunked style transmission. The micro-controller must also be able to compensate for any calibration that must be conducted on the sensor to compensate for any manufacturer variations between sensors.

The many real time calculations involved in the measurement and interpretation of signals required for this device leads to using an FPGA. This would allow for many simultaneous calculations at a much faster rate than a typical micro-controller could handle. FPGAs tend to use more power than standard micro-controllers however, which is a very important aspect in the design of this device. An embedded micro-controller, such as an MSP430 or an ATTINY should be able to provide enough processing power and very low power requirements.

2.3 Competition

When developing any product, it is important to look at the current competition in the pertaining market. Currently, there are no products available on the market today that directly compete with the product we are creating. Companies have taken different approaches to preventing pressure ulcers in patients and these implementations will be explored in further detail in the following sections. Looking into patents specifically, there are multiple device concepts that have been patented but none of the devices mentioned here have made it to market nor do these devices meet all the design requirements of our product.

2.3.1 Market Competitors

For pressure ulcer prevention, there are various competitors. A few examples include the Leaf Patient Sensor by Leaf Healthcare [14], the M.A.P (Monitor. Alert. Protect.)

system by Wellsense [15] and the Alternating Pressure Mattress by Vive [30]. Each of these systems take a very different approach to pressure ulcer prevention. The Leaf Patient Sensor device consists of a gyroscope, or some other form of position sensing sensor, that when placed on the patients torso, can determine the orientation of the patient as he or she lies in bed. Figure 2.2 shows the Leaf Patient sensor and Figure 2.3 shows the typical application of the sensor to the torso of a patient.

FIGURE 2.2: Leaf Patient Sensor [14]

FIGURE 2.3: Application of Leaf Patient Sensor [16]

The data from the Leaf Patient Sensor is streamed through a network of proprietary antennas that then push the data to a central server. This data can then be viewed on a computer, tablet, or other internet capable device. Figure 2.4 shows a typical network structure. The Leaf Patient Sensor is meant to be disposable and contains a battery which is claimed to last 2 weeks [14]. Clinical trials have been conducted using the sensor and the results have proven to reduce the occurrence of pressure ulcers in patients caused by prolonged hospital care [16]. Through a proprietary algorithm, the Leaf Patient Sensor can determine if a patient is at risk of a pressure ulcer based upon their position in bed.

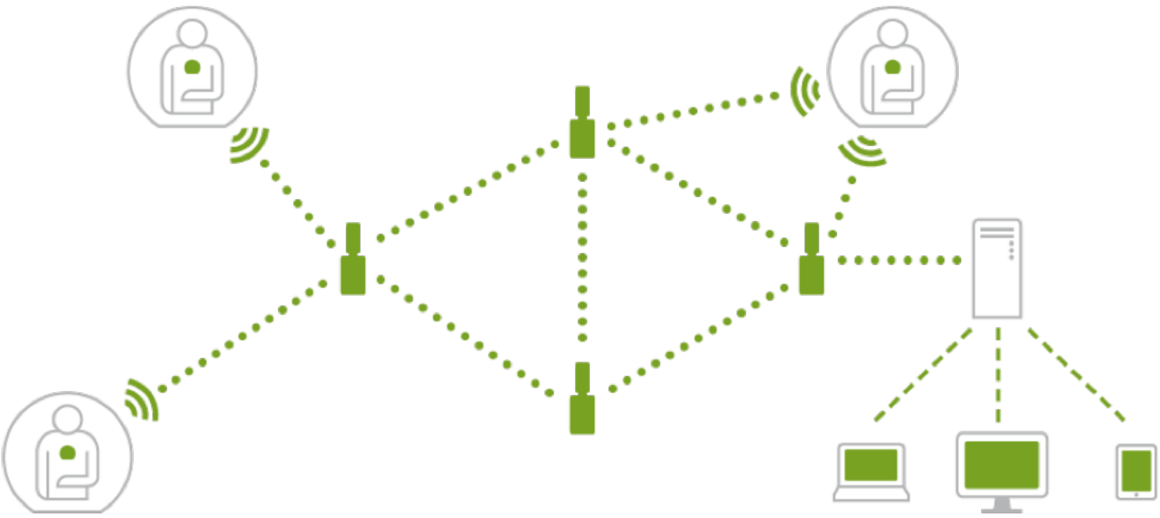


FIGURE 2.4: Leaf Patient Sensor Network Structure [14]

The MAP system takes an entirely different approach to pressure ulcer prevention. The MAP system directly measures the pressure extricated on the patient as he or she lays in bed whereas the Leaf Patient Sensor indirectly determines if a person is at risk of a pressure ulcer. The MAP system consists of a pressure sensing blanket that measures the pressure exerted on the body as the patient lies in bed. Figure 2.5 shows the components of the system (the pressure sensing mat as well as the handle monitoring device). This system has had a number of clinical trials and has proven to reduce pressure ulcers in hospital bedridden patients and has shown to save hospitals $250K to $650K over 6 months [15]. This system does provide a highly detailed pressure map of the patient in bed but takes a more traditional approach to monitoring the patient in that the data can only be access through the bedside monitor itself. This does improve the prevention of pressure ulcers but still requires a nurse to periodically check and interpret the monitor as opposed to alerting the nurse that a patient must be moved or a pressure ulcer will develop.

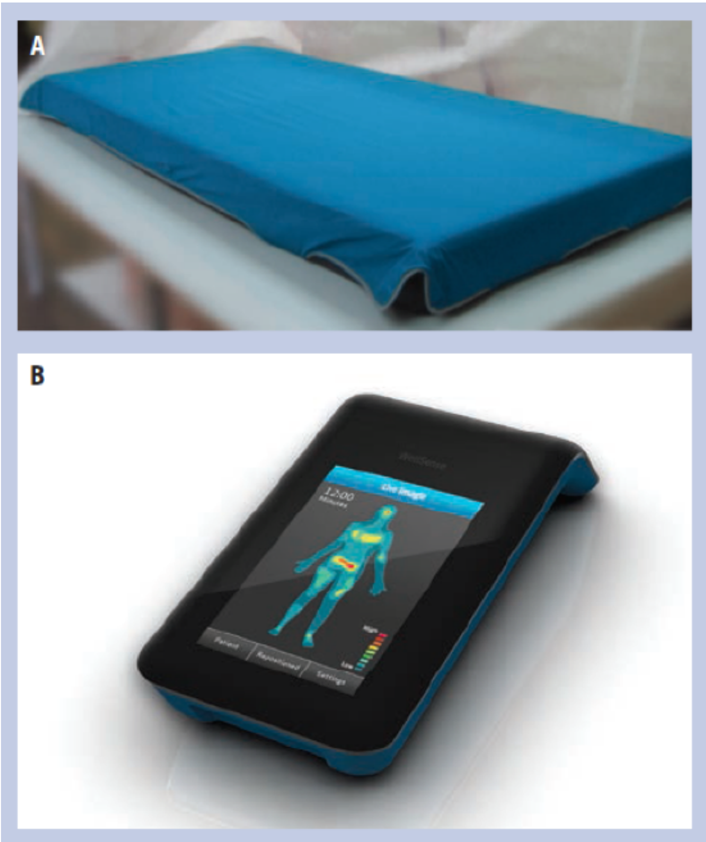


FIGURE 2.5: MAP System (A) Pressure-sensing Mat (B) Handheld Monitoring Unit [17]

The Vive alternating pressure mattress actively works to prevent pressure ulcers from forming. This is accomplished through alternating the amount of pressure applied to various parts of the body as the patient rests. Air inflates and deflates air pockets throughout the mattress to vary the pressure load on the patient. This prevents concentrated areas of pressure from occurring as this is the leading cause of pressure ulcers in bedridden patients. This system can be applied to any pre-existing as a typical mattress cover [30]. The Vive alternating pressure mattress can be seen in Figure 2.6.

FIGURE 2.6: Vive Alternating Pressure Mattress [30]

Each of these devices work to prevent pressure ulcers in creative ways however, both have benefits and drawbacks which become more apparent when comparing them to each other and to our proposed product. The following table, Table 2.3, gives light to some of the drawbacks and benefits of each product as well the future features of our product. Part of the main benefit to our product design is its ability to measure localized pressure at the location of application and then stream this data to a centralized server where the data can be accessed by a physician, by a nurse, or by the patient. This allows for real time monitoring of a patient from anywhere and allows for the ability to warn the appropriate parties of the risk of an impending pressure ulcer.

	Leaf Patient Sensor	M.A.P.	Vive Pressure Mattress	This Work	Future Work
Directly Measures Pressure	×	✓	×	✓	✓
Highly Detailed Pressure Map	×	✓	×	×	×
Pressure Ulcer Warning	✓	×	×	×	✓
Wireless Connectivity	✓	✓	×	×	✓
Accessible Over the Internet	✓	×	×	✓	✓
Battery Powered	✓	×	×	×	✓
Mobile	×	×	×	×	✓
Actively Prevents Pressure Ulcers	✓	×	✓	×	✓

Table 2.3: Comparison of Products on the Market

The MAP system gives an extremely detailed pressure map of a patient but keeps that data localized, does not intelligently warn nurses or physicians of impending pressure ulcers, and requires an outside power source. The Leaf Patient Sensor streams data to a centralized server where the data can be accessed and alerts are triggered when a patient is at a high risk of developing a pressure ulcer. This system is also battery powered but does not measure pressure directly. The position of the patient is used to determine current risk of the patient developing a pressure ulcer. The Vive pressure alternating mattress does not measure the pressure on the patient nor does it record any data about the patient. The device only works to alleviate pressure. Our product will measure localized pressure, stream the data to a centralized server for remote access, work off an internal battery, and actively warn the patient or caregiver in order to prevent future pressure ulcers. The current prototype does not incorporate all of these features that will be apparent in later sections.

2.3.2 Patented Competitors no yet on the Market

With any new product design, it is important to check previously awarded patents. This gives insight into what others in the field have been developing as well as ensuring patient infringement is avoided. Two hardware patents in particular were very comparable to the focus of our device.

A patent issued by the name of System and Method of Reducing Risk and/or Severity of Pressure Ulcers utilizing a traditional design [18]. Pressure sensors are incorporated into sensors that attach to the body like the electrodes of a EKG. These sensors are then connected through wire to a central monitoring unit. Figure 2.7 is an artist rendition of this system.

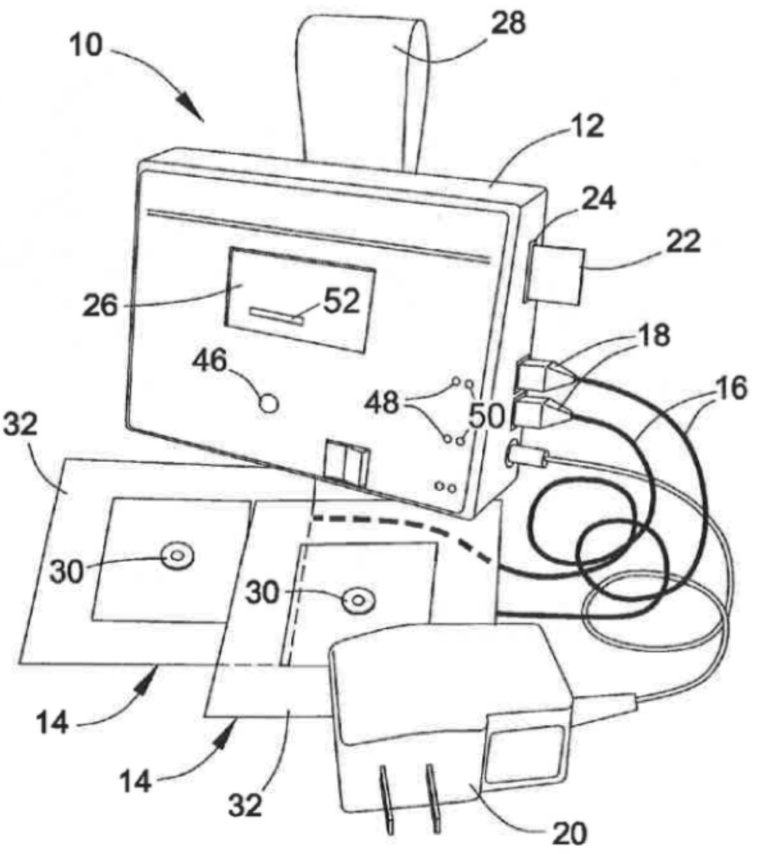


FIGURE 2.7: System and Method of Reducing Risk and/or Severity of Pressure Ulcers Artist Rendition [18]

This system requires an external power source and does not have a method for transmitting the information to a database for remote access. This design is similar to the electrical prototype that was realized for our product in the initial experiments. The main difference in our prototype is that other components (the 555 timer as well as its respective circuitry) were included in the sensor pads that are attached to the patient. The final design of our product will have all of the components of the sensor embedded in the patch and the data will be streamed wirelessly to a central database for analysis. The determination of a pressure ulcer for this patent is accomplished through a predetermined pressure threshold. This is very different from how our product as we will have a dynamic, adaptive risk assessment algorithm.

Another patent of interest is the patent Active On-Patient Sensor, Method, and System [19]. This patent focuses primarily on hardware design. A pressure sensor is attached to the sensor with the components used to read the sensor and transmit the data are enclosed in a separate bandage. Figure 2.8 shows a rendition of the design. The data is sent to a wireless hub and the data is then interpreted by a physician to determine if the patient is at risk of a pressure ulcer. This system does not have any way to intelligently determine if a patient is at risk which will be a main feature of our product. Another major difference between our system and this system is the placement of the sensor. In this design, the sensor is external to the main circuitry of the device. In our design, the sensor will be surrounded by extremely low profile components in order to achieve a single, completely enclosed package.

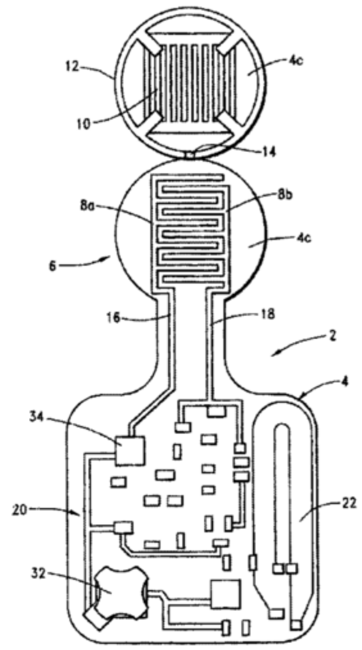


FIGURE 2.8: Active On-Patient Sensor, Method, and System Artist Rendition

Chapter 3

Device Design

3.1 Electrical Prototype Design

With any device, it is important to create electrical prototype designs in order to demonstrate the validity of concepts. This involves showing that the core design will work however, this does not involve making a device that meets all of the specific design criteria. For this device, it is important to show successful measurements of the force sensor and a thermistor. Both of these sensors convert a physical phenomenon into a varying resistance so they can be measured in a similar manner. The electrical prototype will attempt to show successful measurement of both temperature and pressure with a single FR4 board design. External to the board will be the microcontroller as well as the power supply and the method of transmitting the data to a computer for further analysis using MATLAB.

3.1.1 Component Selection

Components for the device were chosen for various reasons. The method of measuring the change in resistance will be determined through the use of a 555 timer rather than an ADC. A 555 timer was chosen over an ADC because of its novel and original approach to measuring resistance. A 555 timer will be a cheaper component to an ADC and allows for varying resolutions of resistance measurements. By measuring multiple periods of the square wave resulting from the 555 timer, it is possible to determine a more accurate value of the resistance of the varying resistor. A duel package 555 timer [20] was chosen for the electrical prototype as both temperature and pressure were measured.

A force sensor by Interlink Electronics (FSR 402 Short) was chosen as the pressure sensor for the device [21]. This device allows for force measurements of 0 to 1kg. Figure 3.1 shows a graph of typical voltages across the FSR 402 Short when placed in a voltage divider of varying resistor values and a power source (V+) of 5V.

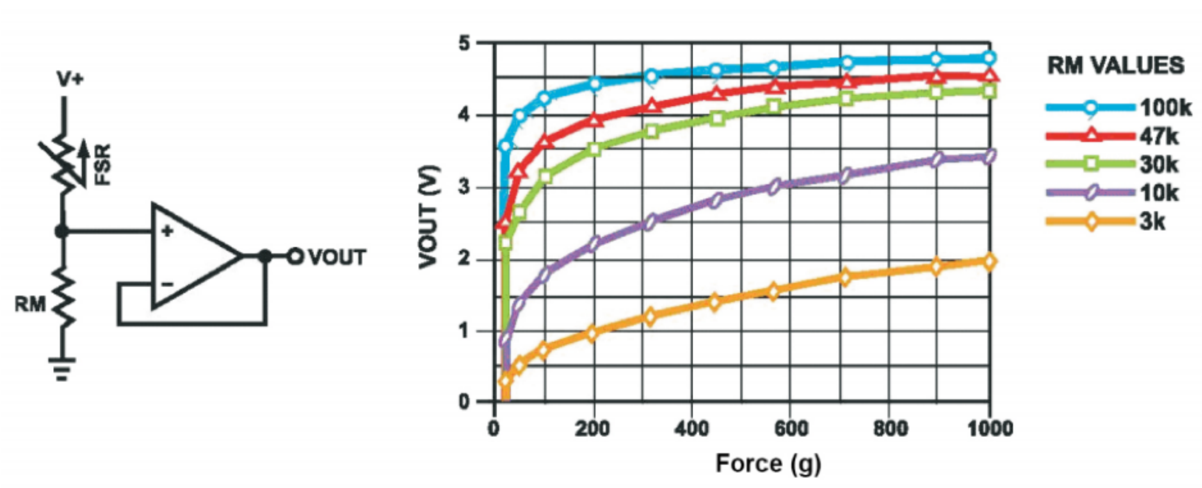


FIGURE 3.1: Circuit and Voltage Curves of FSR 402 Short in a Voltage Divider with Varying Resistors [21]

From Figure 3.1, it is apparent that placing a 10k in series with the FSR 402 Short would provide the greatest range of measurable voltage values for various applied force on the sensor. From the datasheet, it is also known that the active surface area of the device is $1.27cm^2$ which can then be used to determine the pressure exerted on the sensor. Explained in the FSR Integration Guide, there is a part-to-part variability of about ±15% to ±25% as shown in Figure 3.2 [22]. This will give rise to a calibration technique for each component in order to reduce this variability. In Figure 3.2, the dashed lines represent the ranges of of part variability where the red line represents the results of the average part.

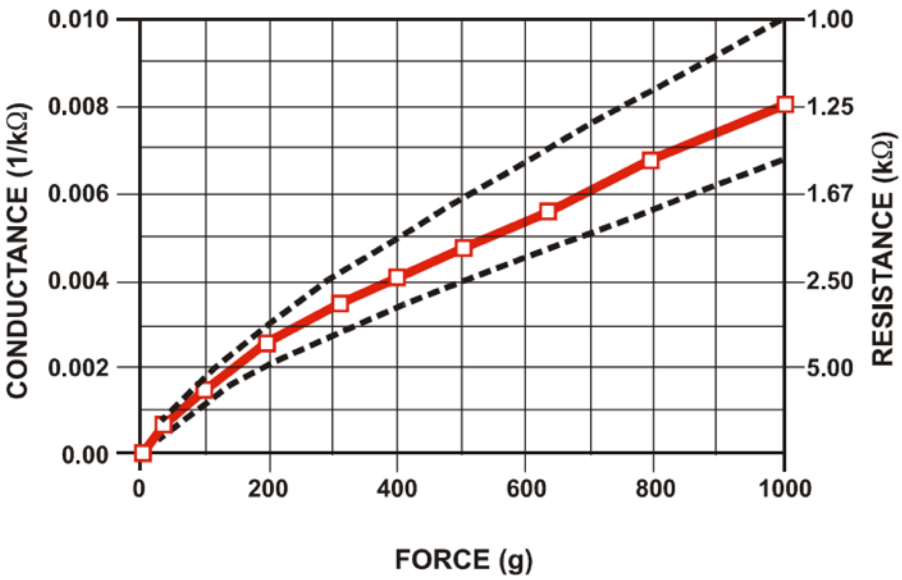


FIGURE 3.2: Conductance and Resistance vs. Force for FSR 402 Short [22]

The micro-controller was chosen largely on what was on hand and what provided the most straightforward platform for prototype development. An Arduino UNO [23] was chosen as it has a robust development platform and rapid prototyping capabilities. The thermistor was chosen primarily based on the FSR 402 Short. Because it was determined that the FSR 403 Short would be placed in series with a $10k\Omega$ resistor, a thermistor with nominal resistance of $10k\Omega$ was chosen. This would be placed in series with another $10k\Omega$ resistor in its own separate measurement circuit. The remainder of the resistors and capacitors were chosen as necessary and that selection is explained in the following section describing the circuit.

3.1.2 Explanation of Circuit

The 555 timer circuit was designed using the astable 555 timer design. This allowed for the predictable nature of the timer allowing for the determination of the resistance of the FSR 402 Short and the thermistor. Figure 3.3 shows the circuit diagram utilized in the electrical prototype. A resistor was placed in parallel with the FSR 402 Short because when no pressure is applied, the resistance is extremely high and can be considered an open circuit in this application. By adding the resistor in parallel, the 555 timer is always producing a valid square wave and is stable. This leads to greater stability in the micro-controller operation in that there is always a measurable signal.

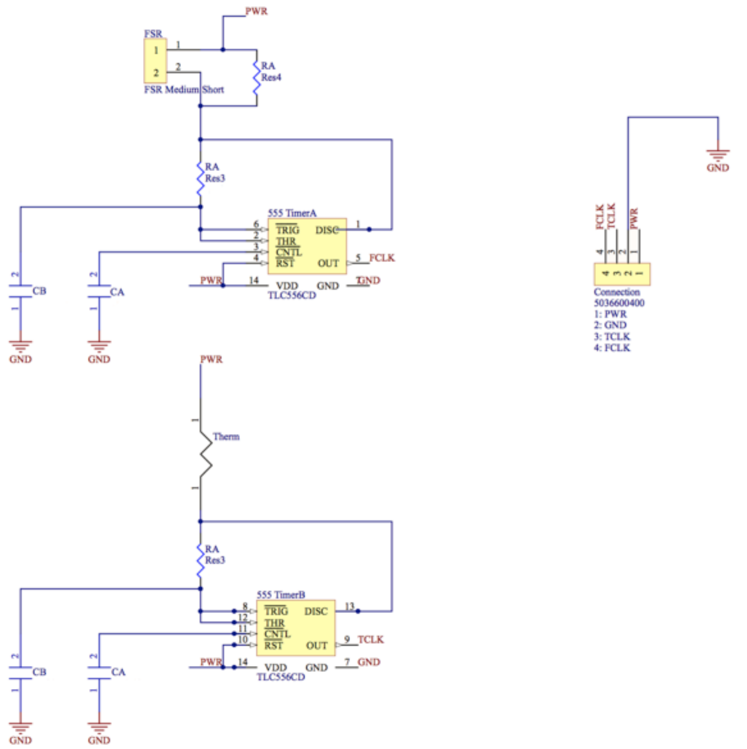


FIGURE 3.3: Electrical Prototype Circuit Diagram

The purpose of this prototype was to validate the measurement technique of using 555 timers as well as the pressure sensor. For this reason, an external power source was incorporated and the clock signals corresponding to each signal were sent through a hard wire to an external micro-controller.

3.1.3 PCB Design

The pcb design was intended to incorporate the sensors in a small package that would simulate the likely final design of the device. A dual 555 timer was used as there are two sensors to measure. The FSR 402 Short was placed on the board so that no components were behind the sensor. It is important that the sensor lays on a flat surface to get valid readings. For the electrical prototype, FR4 material was used for the board however, the final design will incorporate a flex board as this will be more comfortable for the patient and will result in a more accurate measurement of the pressure applied to the skin. Figure 3.4 and Figure 3.5 show the net routing and the 3D model for the PCB design for the electrical prototype. The board incorporates a 2-layer design where the sensors are on one side and the other various components (555 timers, resistors, capacitors, etc.) are on the other.

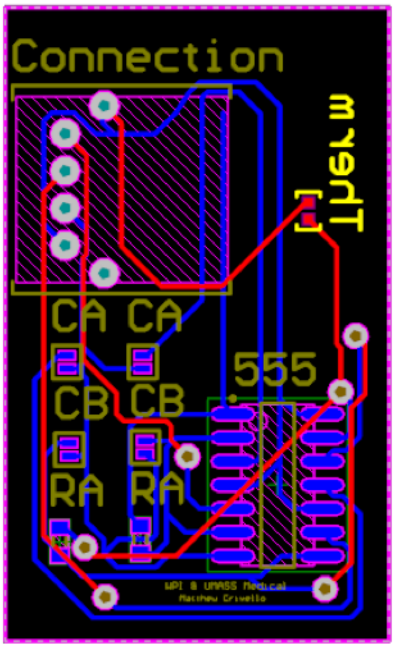
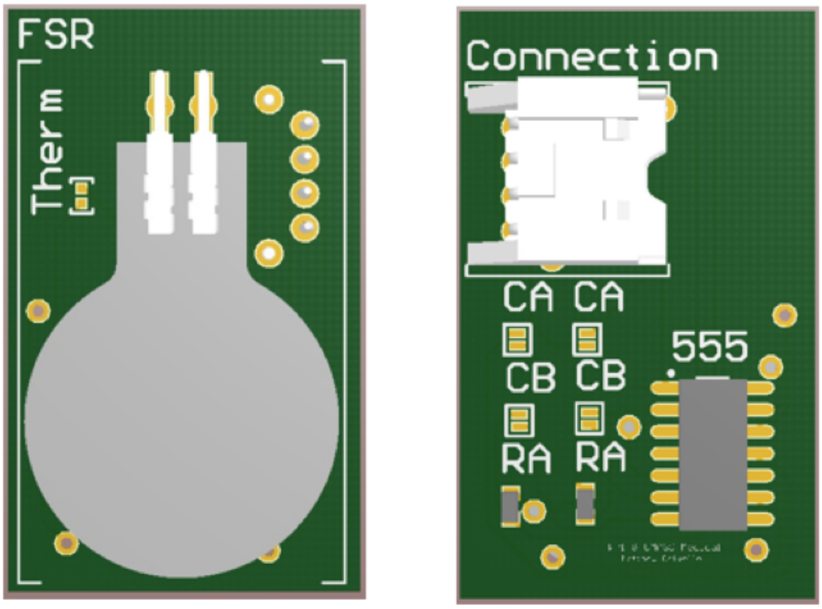


FIGURE 3.4: Electrical Prototype PCB (net routes)



a) b)

FIGURE 3.5: Electrical Prototype 3D Rendering where a) is the Front and b) is the back

From the digital designs, a physical prototype was created. This product can be seen in Figure 3.6. A slight modification was made to the design after it was constructed. The connector height was much higher than the surrounding components creating an unlevel platform for the force sensor. This affected the pressure measurements and it was therefore determined that it needed to be removed. Wires were soldered directly to the vias used by the connector instead. This created the level surface needed for the calibration process development.

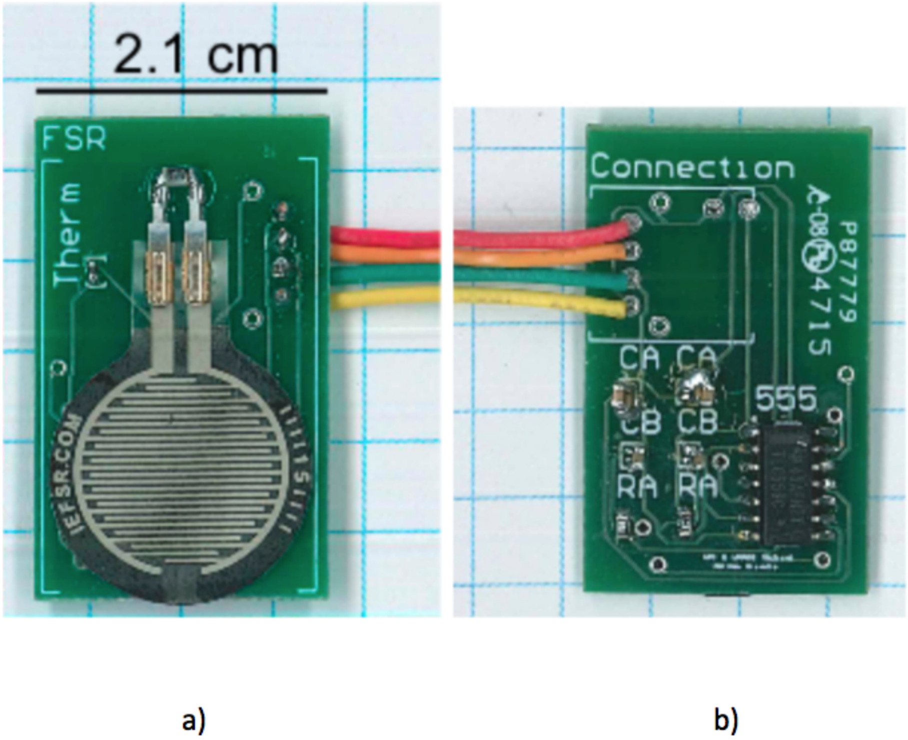


FIGURE 3.6: Electrical Prototype Assembled where a) is the Front and b) is the back

3.1.4 Software Design

The software for this device must be able to record and interpret the signal sent from the 555 timers. The device sends a square wave to the micro-controller and which can be measured by a digital pin because of the on off nature of the signal. The device must be able to measure the signals accurately and send the data to a computer for further analysis. There are two primary approaches to accomplishing these requirements: using a digital pin and using an interrupt pin. Both approaches were investigated.

3.1.4.1 Digital Pin Approach

The digital pin approach implemented took advantage of predefined function pulseIn [24]. This function records the amount of time is HIGH or LOW at the choice of the user. This function was used to measure both the HIGH time and LOW time of the signal on a specified pin. This data was then used to determine the duty cycle of the signal and thereby the resistance of the sensor. The function will only work on signals that are 10 microseconds to 3 minutes in length and the function will wait for a change before it will start recording time [24]. Once a script was established, the program had to be tested against known duty cycle values. This was accomplished by connecting a function generator to the specified pin rather than the output of the 555 timer. The percent error of this test can be seen in Figure 3.7 for various frequencies and duty cycles.

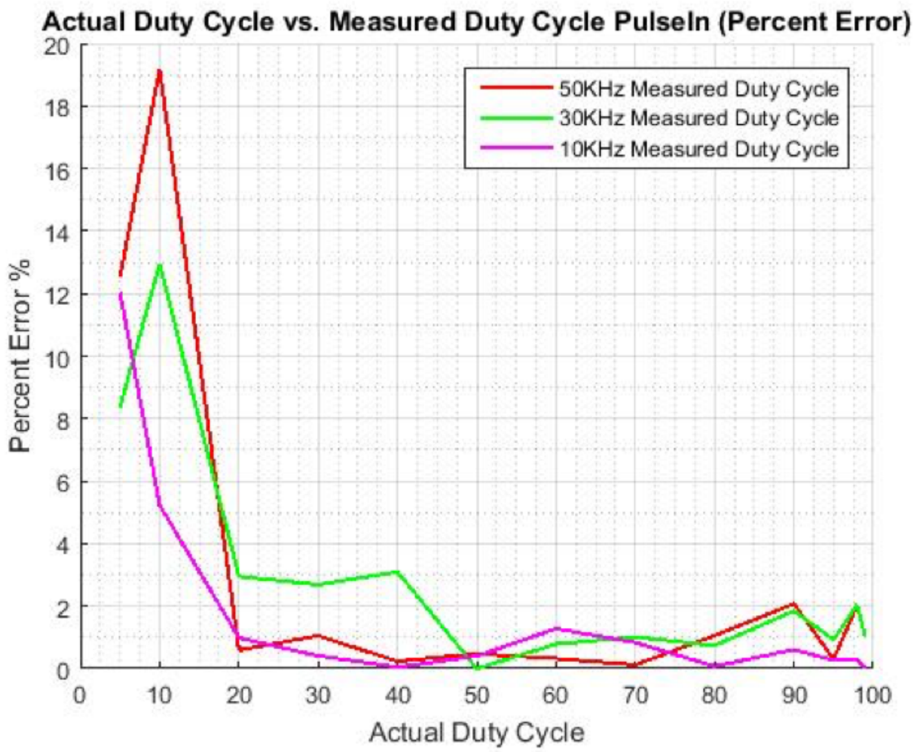


FIGURE 3.7: Actual Duty Cycle vs. Measured Duty Cycle PulseIn (Percent Error)

It is clear that there is a lot of error between the actual duty cycle and the measured duty cycle. This error would lead to drastic errors in the determination of the resistance value of the sensors. The FSR 402 Short already has a high variability from part to part so it is crucial to reduce the error in all other aspects of the device. For this reason, a different approach was taken utilizing the external interrupts of the Arduino.

3.1.4.2 Interrupt Approach

Another approach to measuring the duty cycle of the signal from a 555 timer is through an interrupt based approach. Through this approach, interrupts are used to count the time in between pulses rather than polling pins continuously. This method proved to be highly accurate and greatly improved over the previous implementation of polling pins. The results of the same test with the function generator are shown in Figure 3.8. The percent error has been reduced dramatically. This is the preferred method chosen for further tests with the electrical prototype design.

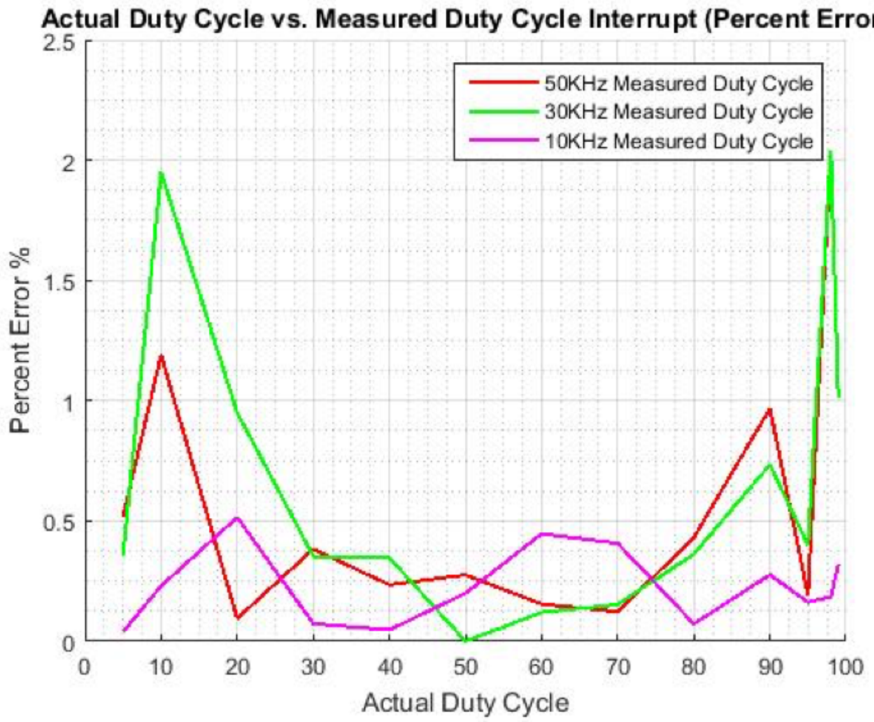


FIGURE 3.8: Actual Duty Cycle vs. Measured Duty Cycle Interrupts (Percent Error)

3.1.5 Thermistor Measurement

The thermistor works in a similar way to the FSR 402 Short where a change in temperature corresponds to a change in the resistance of the NTC thermistor. With this in mind, the resistance value of the thermistor is determined in the same way through the use of a 555 timer. With the resistance value, the Beta and Steinhart-Hart equations [25] can be utilized to determine the measured temperature value of a NTC thermistor.

$$\frac{1}{T} = \frac{1}{T_0} + \frac{1}{\beta} ln(\frac{R}{R_0}) \quad (3.1)$$

In equation 3.1, R represents the measured resistance, T represents the calculated temperature value, T0 represents room temperature in kelvin, is the beta parameter provided in the data sheet for the thermistor, and R0 is the nominal value of the resistor which is provided in the datasheet of the thermistor as well. This method allows for a temperature measurement with a tolerance of $\pm 0.2^{o}$C [25].

3.2 Apparatus for Calibrating Sensors

To test the force sensor, it was important to develop an apparatus to create repeatable measurements in a controlled environment. Figure 3.9 shows the apparatus developed to conduct the experiment.

Figure 3.9: Force Sensor Calibration Apparatus

The device consists of a platform with a flat surface to place the sensor. The sensor is then surrounded by a long transparent tube that has an opening on the side to allow pass through of wires to the sensor. There is an opening at the top as well to allow for a secondary tube to be placed within. The platform was created from a 18" x 4.75" x 0.75" plank of plywood. The attached orange base and clear tube resulted in a height of 13.25". The secondary tube (the black tube in Figure 3.9) has a flat bottom with a peace of sponge attached to the bottom. The length of this tube was 12.5" with a diameter of 1.75". The sponge allows for complete and direct contact with the sensor and the tube. On top of the tube, a 3D printed platform was attached to provide an adequate surface for the weights to be placed repeatedly. The weights were stacks of 13 United States quarters that were weighed to be 75 grams each. This allowed for a repeatable, known weight to be applied to the sensor. The tubes were required because it was desired the weight be applied directly to the sensor. With the long tubes, the inner black was able to lean against the outer transparent tube allowing for a very small amount of the weight to the be transferred to the outer tube but the vast majority of the weight to be applied directly to the sensor itself. The weight of the black tube had to be taken into account as well in the measurements taken with this apparatus. For weights less than the weight of the black tube, a paper tube was created and lose quarters were placed on top of the tube. The length of this tube was 1' with a diameter of 1.5". The paper tube can be seen in Figure 3.10.

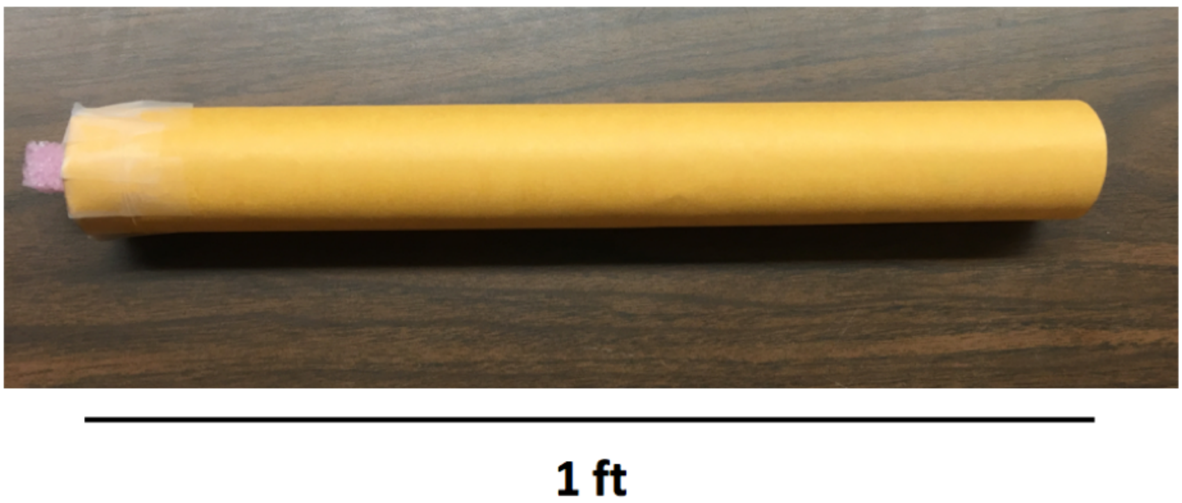


FIGURE 3.10: Paper Tube for Calibration

With the creation of the apparatus, it was possible to test and develop a calibration procedure for the sensor. This setup allowed for the development of a calibration procedure for the device as well.

3.2.1 Variability in FSR 402 Short

With a proper testing apparatus developed, it is possible to test the variability in the FSR 402 Short. With this data, it is also possible to determine a calibration method. For this experiment, varying weights were placed on the tube for a duration of 20 seconds allowing for 60 readings to be taken at each weight. This number was then averaged and finally plotted versus the actual weight. Figure 3.11 shows the results of this experiment.

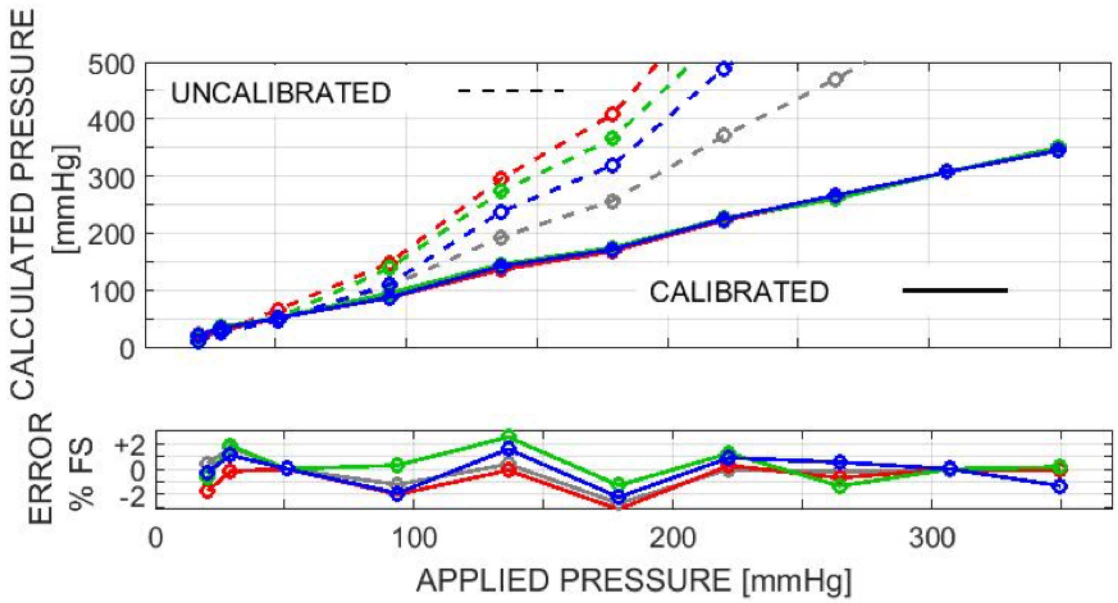


FIGURE 3.11: Measurement results, before (dashed) and after (solid) calibration

In Figure 3.11, the upper plot shows the calculated pressure from four tested sensors, represented in different colors, as a function of the known applied pressure. The dashed lines indicate calculated pressure using the nominal FSR parameters. The solid lines show the results after calibration. Clearly shown is the variability of the components which confirms the data from the manufactures in (Figure 3.2). Also shown, is the reduction of force measurement variability to ±3% of full scale for the calibrated output.

3.2.2 Calibration Methodology

To create the calibration procedure, it is first important to note the corresponding conversion factor for determining applied pressure from applied weight. For many applications within the medical community, the applied pressure on the skin is more important than the weight applied. The conversion is accomplished through a few simple unit conversion steps. It is known that 1 gram per square centimeter is equal to 0.7356 mmHg (millimeters of mercury) and from the datasheet, it is also true that the active area of the FSR 402 Short is 1.27 cm^2. Therefore, 1 gram of force is equal to 0.579 mmHg of pressure for the FSR 402 Short. With this conversion, a range of 30g — 600g is equal to 20mmHg — 350mmHg. Looking at this range in Figure 3.2, it is clear that there is a power-law relationship that follows the equation

$$R_{FSR} = R_0 F^x \tag{3.2}$$

In equation 3.2, R_0 and x represent the results of the two-point calibration, F represents the current measured resistance before calibration, and R_{FSR} represents the calibrated resistance result. This allows for the determination of the force applied to the sensor through the measurement of the resistance of the sensor. Through a least-squares fit of the 30g — 600g range of data in Figure 3.2, nominal values of R_0 and x are $R_0 =$ 200$k\Omega$ and $x =$ -0.738 where the following equations (equations 3.3 and 3.4) were used for the least-squares fit:

$$x = \frac{log({}^{da}/_{db})}{log({}^{da_{actual}}/_{db_{actual}})} \tag{3.3}$$

$$R_0 = da_{actual}(da)^{\frac{1}{x}} \tag{3.4}$$

In equations 3.3 and 3.4, da and db are the corresponding resistant values for the two calibration points. For the two calibration points, 98.6 grams and 548.6 grams were chosen. These values were used as they fell within the 30g — 600g range well and represented easily repeatable applied weights (the weight of the black tube and the weight of the black tube plus 6 stacks of quarters where each stack consists of 13 quarters). Where da and db were the measured calibration values, da_{actual} and db_{actual} represent the actual applied weights. This reduces the part-to-part variability of the force sensor considerably (from ±25% to ±3% as shown in Figure 3.11). This method of fitting data is known as a power fit and was first implemented within MATLAB and then incorporated into real time calculations on the Arduino.

3.3 Prototype Design for Pig Experiment

With any design, it is important to test prototypes through multiple experiments. For this prototype, it is imperative to determine the validity of the design with live subjects as the device will be used on live subjects in its final implementation. With the help and generosity of Dr. Raymond Dunn, we were able to test the product design on live pig subjects. These animals were part of another study unrelated to this project involving the animals abdomen. This left the rest of the animal free for some experimentation involving our device. The electrical prototype would not be able to support such an experiment and therefore, a new prototype would have to be designed and fabricated.

3.3.1 Design Choices

There are many aspects to consider when developing a device that will be used upon a live subject. A major restriction placed upon the design is a time restriction. The involvement in the project was only known about 2 months before the experiment would take place. This drove much of the design as the previous prototype was in no way designed for testing on subjects. It was only designed to show the validity of the electrical design.

A primary consideration for the design is the structure the components will be mounted upon. A standard FR4 board is too rigid to be placed upon the body and will adversely affect the pressure measurements of the device. Another consideration is the micro-controller interpreting the signals. Ultimately, the micro-controller will be included directly on the device however, in the interests of time, it was determined that an Arduino Mega will be connected through a ribbon cable to each device. This alleviated the need to include an on board power source as well as the Arduino Mega can provide a maximum of 500mA which is directly from the USB port powering the Arduino Mega. From the datasheet, each single 555 timer draws a maximum of 6mA of current to operate with a Vcc of +5V when the output is low with no load [26]. It is desired that there will be three different devices per Arduino Mega so with two timers per device, that results in 36mA of current at +5V which much lower than the maximum output current of 500mA. To connect the devices to the Arduino Mega, ribbon cable was split by the various devices and then attached to the Arduino through keyed connector. A mezzanine board was constructed in order to map the ribbon cable connector to the specific pins utilized on the Arduino Mega as shown in Figure 3.12.

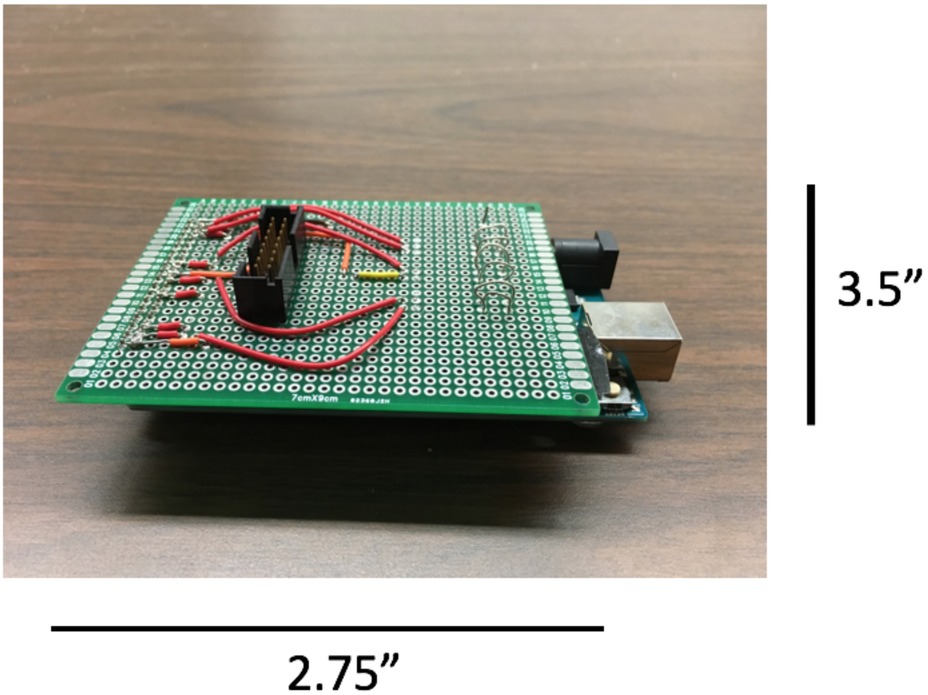


Figure 3.12: Arduino Mega Board with Mezzanine Board

Initially, it was thought that there would be four subjects at once in the experiment. This changed to one subject in the final stages of the development process however, the design decisions were made to accommodate many subjects. For this reason, it was determined that a local network would be needed to collect all of the data streamed from the devices in one location. A raspberry pi would take the data from the Arduino and stream that data to a server where the data would be stored for further post experiment analysis. This alleviates the need to include a nonvolatile storage solution with the Arduino and this setup more closely simulates the system that would be implemented in the final design.

3.3.2 Flex PCB Design

The physical design of the prototype must include all of the necessary components for the proper functionality of the device. This must be done in such a way as to not affect the pressure measurement. The temperature measurement is not as susceptible to the package size of the various components. As another constraint, it was desired that the device itself be the size of a standard electrode patch as a device this size is very common in the medical world. Figure F.2 and Figure F.3 show the net trace of the device as well as a 3D rendering of the device with attached components.

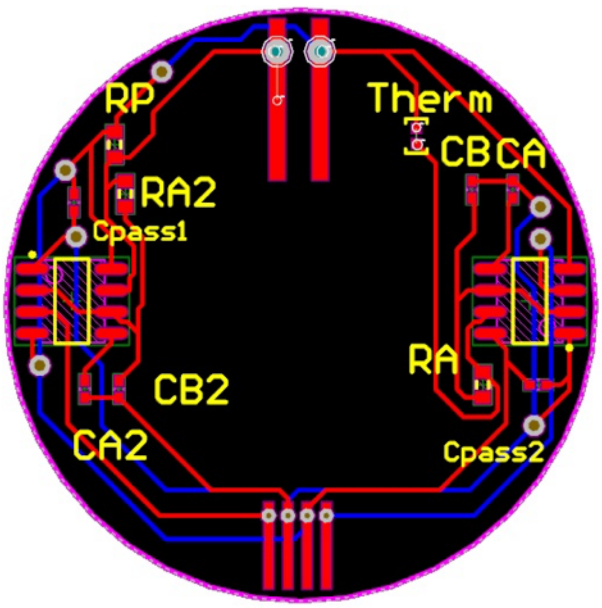


FIGURE 3.13: Flex PCB Prototype (net routes)

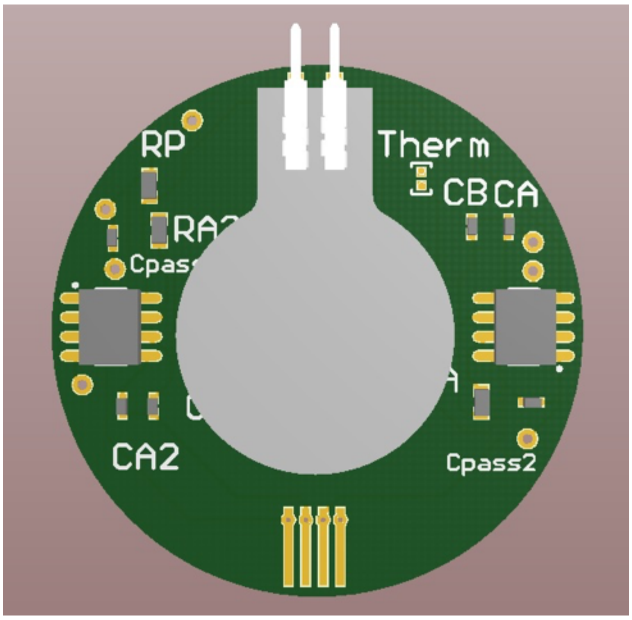


FIGURE 3.14: Flex PCB Prototype 3D Rendering

Figure F.3 shows the FSR 402 Short as the main focus of the device and is surrounded by the corresponding components for the circuit. In the previous electrical prototype, a dual package 555 timer was used for creating the two signals. Because of the size constraint of the device, two single package 555 timers were used. This allowed for the timers to be placed around the sensor along the edges of the devices. Another aspect of the design is the routes of the nets. This was kept away from the underside of the FSR 402 Short to rule out any interference this may cause with the sensor. Figure 3.15 shows an image of the finished prototype.

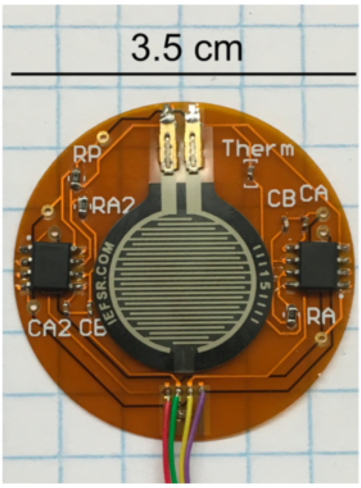


FIGURE 3.15: Completed Flex PCB Prototype

In Figure 3.15, it is apparent that an external power source is applied and the signals from the 555 timers leaves the device. These cables were attached through vias designed into the device previously. Through testing it was determined that these vias did not hold up under repeated use and would sever. This affected specifically the ground line so an additional wire was grafted onto the board to bridge the gap where the connection was severed. A small amount of glue was added to the incoming wires as well to help prevent this from occurring again. Figure 3.16 shows the results of this modification.

FIGURE 3.16: Completed Flex PCB Prototype with Modification

For future designs that are not wireless, the incoming wires should be multi-core wire and should be soldered directly to surface pads rather than vias. The flexible nature of the flex pcb leads to repeated strain and eventual failure of the connection with vias. This should not be a problem for designs that include a wireless solution.

3.3.3 Software Changes for Multiple Signals

In previous testing, only one device (two 555 timer signals) was attached to an Arduino. This allowed for the interrupt method of measuring the duty cycles of the signals to be highly accurate. With the addition of two more devices (four additional 555 timer signals) for one Arduino, accuracy was reduced. With essentially 6 separate interrupts, there is a high probability of multiple interrupts coming in at the same time. This would then lead to missed rising and falling edges as the device would have to assign priority levels to the various interrupts. Each device signal should be treated with equal priority levels as no device is more important than another. This would lead to problems in the software logic when dealing with multiple interrupts coming in at the same time. Another potential problem with this setup is if the micro-controller is handling a specific interrupt and another interrupt is thrown with a higher priority interrupt, then that interrupt would be handled before the previous interrupt. This would lead to miscalculations as the determination of duty cycle is extremely time sensitive.

To alleviate the problem with using interrupts, a different solution was devised. The new approach looks at each signal individually in succession. This insures that no data is missed and that calculations are not interrupted and corrupted. A drawback of this method is that none of the other signals can be measured while one signal is being measured. This method of cycling through multiple signals is known as time-division multiplexing. This is a common technique in networking for sending multiple signals through a common signal path. In addition to implementing time-division multiplexing, averaging over a specific number of periods was also utilized. This increases the accuracy of the duty cycle measurement and increases the accuracy of the determination of the resistance. Averaging over a specific number of periods rather than a specific amount of time was chosen so that partial periods were not included within the average. This opens the software up to a potential bug if no signals periods are received. This could happen in the event of a device failing. To account for this situation, a timeout period was implemented that would force the micro-controller to move on to another device if no periods were received in a reasonable amount of time (predetermined). This timeout also forces the micro-controller to move on if a signal is slower than expected. Because the averaging is based on the number of periods, a slower signal will cause more data to be missed in a faster signal. The timeout accounts for this and forces the micro-controller to adapt accordingly.

3.3.4 Experiment Setup

Pigs of a mass of about 80kg were used in the experiment. Because of the nature of the surgery being performed on the pig, anesthesia was applied prior to placing the sensors on multiple sites along the back. Figure 3.17 shows the relative locations the sensors were placed on the pig. Each location was determined to be in proximity to a bony prominence by the lab technicians. Figure 3.18 shows a close up application of a sensor at one of the locations on the pig. These locations along the back of the pig replicate a similar situation of a human on his or her back during a lengthy surgery.

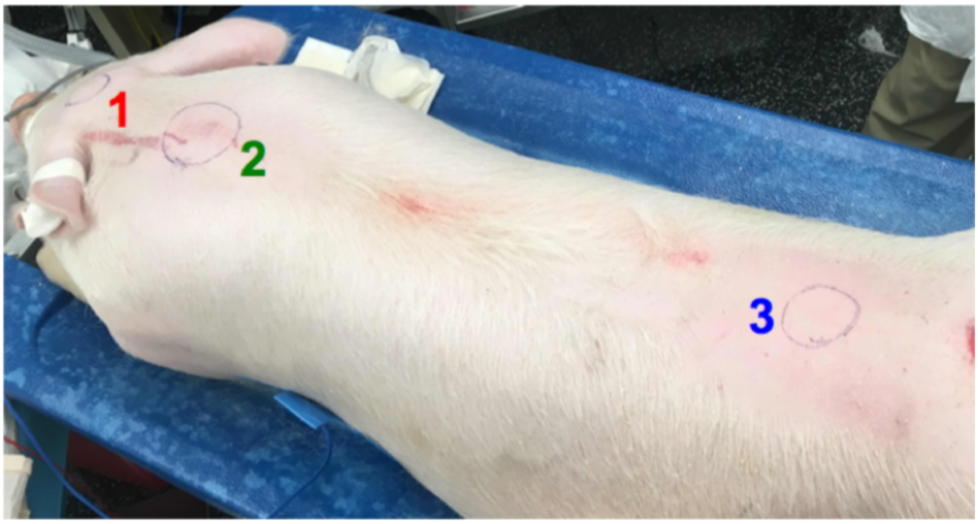


FIGURE 3.17: Sites Instrumented on Anesthetized Pig

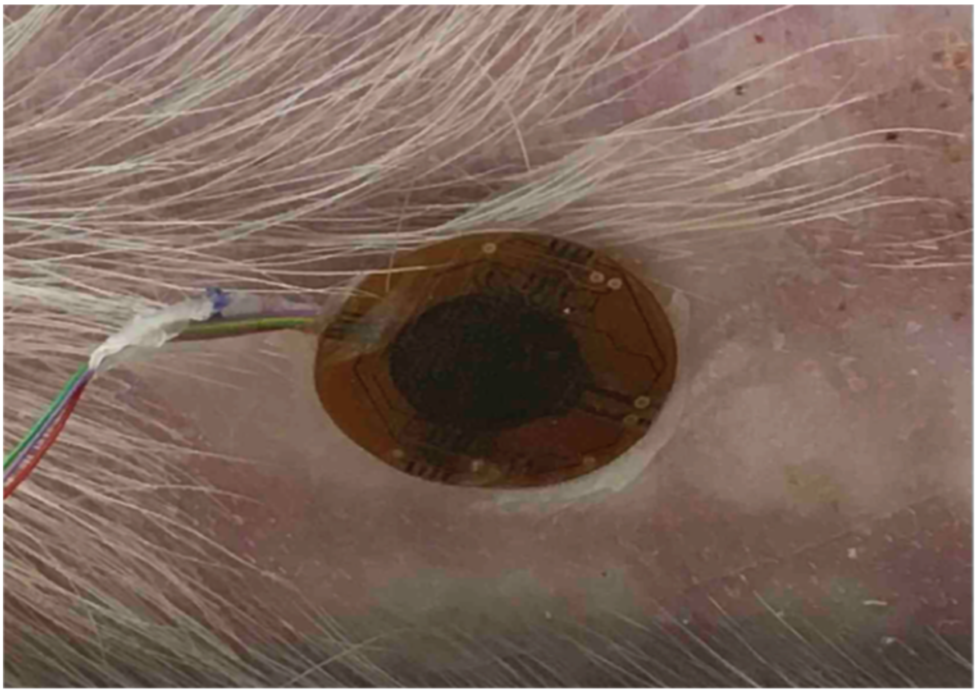


FIGURE 3.18: Close-up of Site After Sensor Application

3.3.4.1 Different Network Components

After it became apparent that only one pig would be tested, the networking portion of setup became unnecessary however, it was still utilized as a test of the kind of setup that may be implemented in the future. The network consisted namely of three components: a raspberry pie 3, a router, and a server. The three sensors placed on the pig were attached to a single Arduino mega. This Arduino was then connected to a raspberry pie using a USB Type B cable. The Arduino communicated with the raspberry pie through a serial connection and that data was captured and forwarded to a ThingWorx server using a java application known as the Academic Edge Connector [13]. Figure 3.19 shows a screenshot of the Academic Edge Connector application.

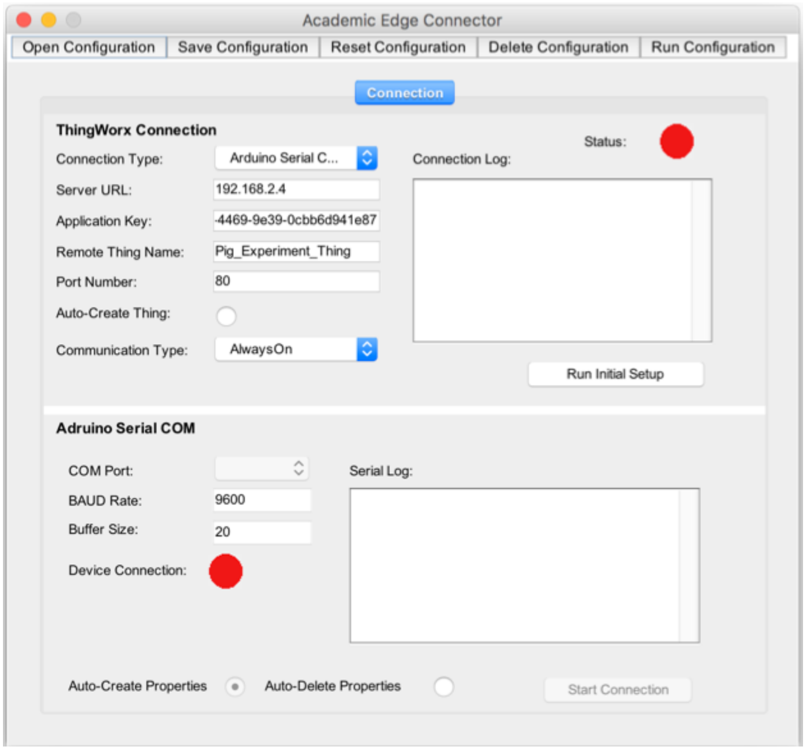


FIGURE 3.19: Screenshot of the Academic Edge Connector

Multiple connection types are possible through this application including REST calls and an always on protocol connection through web sockets. The always on connection was chosen for this application as it offers a more stable connection with less packet loss between the client and host.

The raspberry pie was connected to the router through a wireless connection and the server was connected to the router through a wired Ethernet connection. The ThingWorx server was hosted on laptop pc as this experiment does not put much strain on the server and a laptop allows for enough mobility to utilize a simple, closed local area network rather than pushing the data across the internet. Figure 3.20 shows a diagram of the experiment setup for one subject (a diagram for three subjects is located in Appendix G). Figure 3.21 shows an image of the full network setup of the experiment.

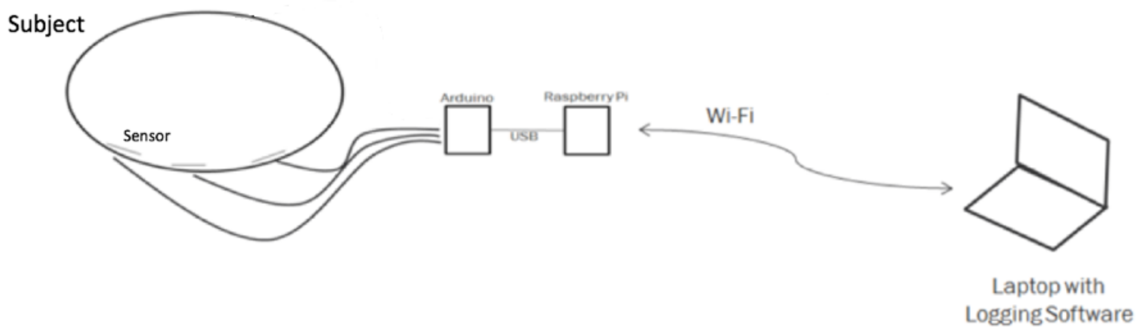


FIGURE 3.20: Experiment Setup Diagram

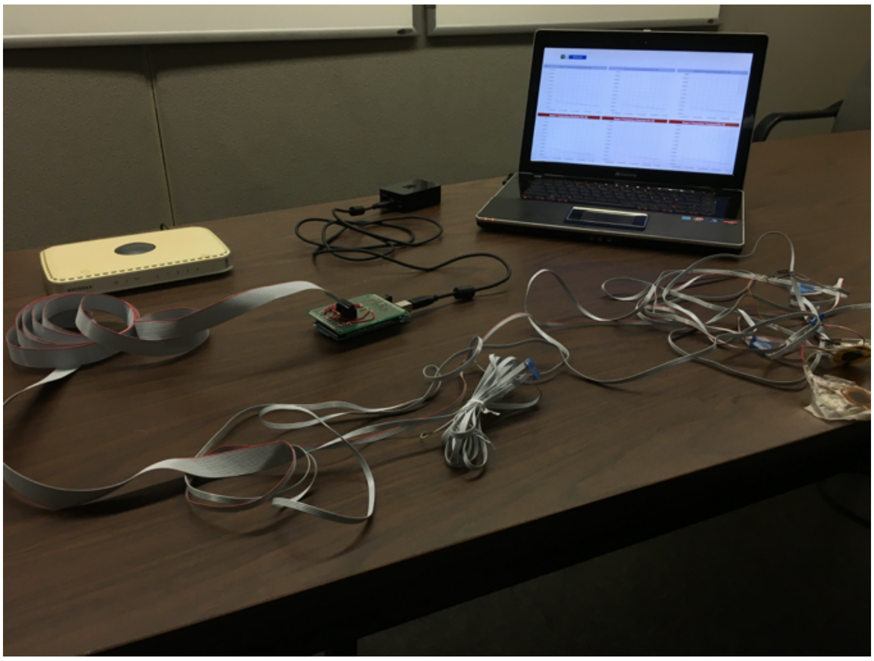


FIGURE 3.21: Image of all Experiment Components

3.3.4.2 ThingWorx Software

The software used on the server to record all of the data coming from the sensors is called ThingWorx. ThingWorx is a platform designed specifically for the IoT development and deployment space. Multiple data types through multiple data communication protocols are possible with the use of this software. Thingworx also allows for the real time display of data as it is received. For this experiment, a mashup was used to display and graph the data as it is received by the server as seen in Figure 3.22.

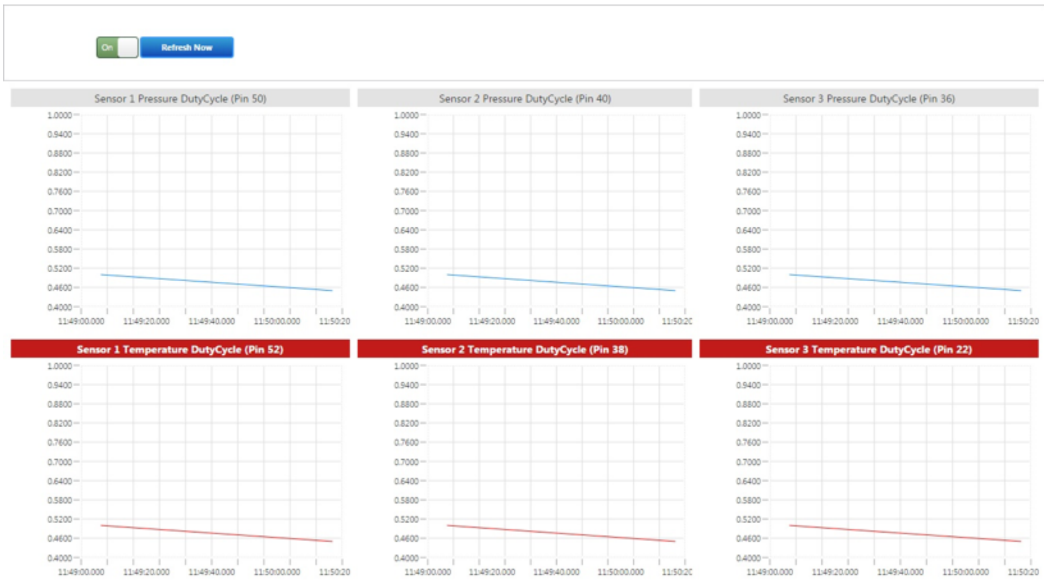


FIGURE 3.22: Screenshot of ThingWorx Mashup used in the Experiment

The real time display gives a visual display of the data as it comes in providing diagnostic information. If there is a major problem with the sensor, then it would be immediately apparent through the display. In addition to visually showing the data as it arrives at the server, ThingWorx allows for real time analytics on the data. This was not implemented however, as it was more practical to use MATLAB for data analysis. ThingWorx was primarily used to display data and aggregate that data into a specifically formatted CSV file including timestamps as the data was received by the server.

3.3.5 Results

The experiment lasted approximately 7 hours over which pressure and temperature measurements were taken throughout. Over the course of the experiment, it became apparent that sensor 1 had a mechanical failure and became intermittent. Figure 3.23 shows the results across the entire experiment.

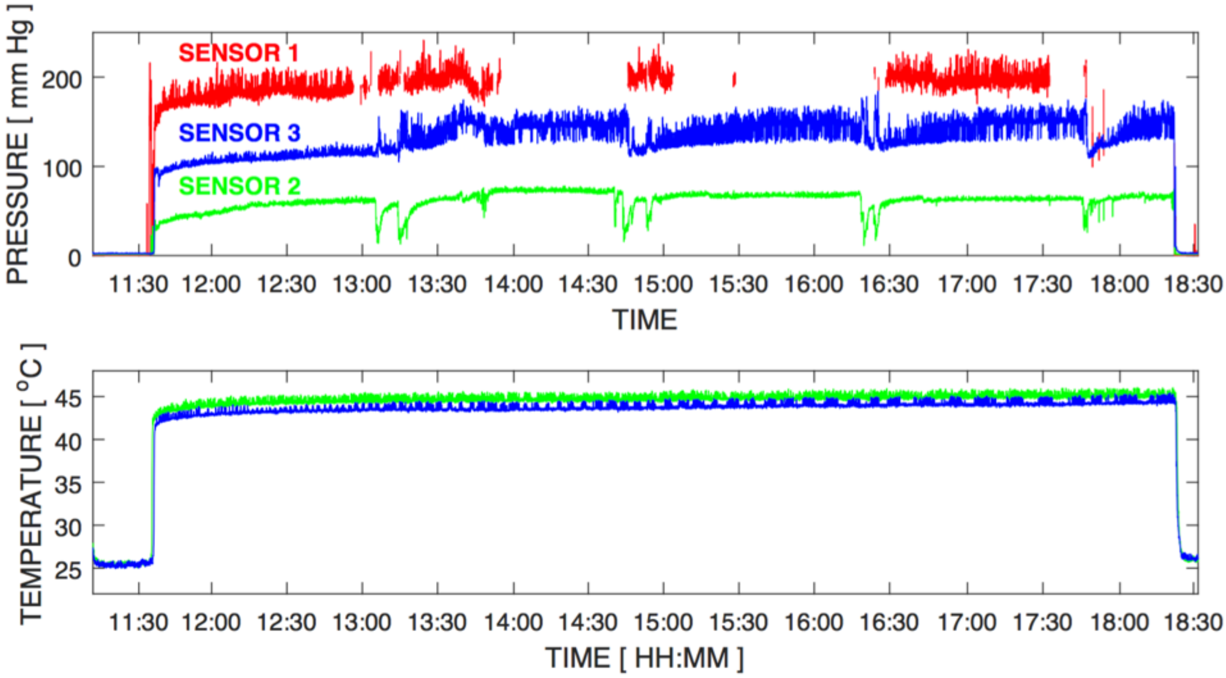


FIGURE 3.23: Pressure and Temperature Data Over Duration of Surgery, Including Pre and Post Attachment Control Times

The loss of data in the readings from sensor 1 are clearly shown in Figure 3.23 leading to the conclusion that the sensor failed during the experiment. The sensors took samples in approximately 2 second intervals with the sensors being attached between 11:35 and 11:37 and detached at 18:22. The readings from sensor 2 and 3 are accurate and consistent with the procedure of the surgery with a range of 30 to 80 mmHg for sensor 2 and a range of 100 to 180 mmHg for sensor 3. Part of the surgery protocol required the pig to worked on in approximately 90 minute intervals. This resulted in the movement of the pig at these intervals and the adjustment of the pig resulted in variations of the pressure. These times correspond with dips in pressure at times 13:06 to 13:14, 14:39 to

14:55, 16:18 to 16:26, and 17:46 to 17:50. The dips in pressure are more pronounced for the readings taken from sensor 2 than in the readings taken from sensor 3. This is likely due to the location of the individual sensors.

Temperature measurements were taken throughout the experiment and are shown in the lower graph in Figure 3.23. Throughout the experiment a temperature measurement of 42^{o}C was observed. This is consistent with the procedure of the surgery as the surgery protocol required the pigs to be placed on a thermal blanket which was set to 42^{o}C.

The data shown in Figure 3.23 does not include any filtering of the data received from the sensors. The data also includes a pre and post application period. These periods act as a control period for the sensors showing a period of approximately zero pressure and a temperature measurement of approximately the ambient temperature of 25^{o}C.

Chapter 4

Conclusion

This thesis presented the design, layout, and test of a device which address a prominent need in the medical field. Chapter 2 discussed the background behind the severity of pressure ulcers as well as previous technical solutions to addressing this problem. Additionally, chapter 2 worked to develop a list of device characteristics that would result in a product which would solve the prominent issues in pressure ulcer prevention. Chapter 3 delved into the design and test of two prototype devices including a clinical test with live pig subjects.

The ultimate goal of this thesis was to create a device which could be applied directly to a patient's skin and measure pressure in high risk areas. The novel idea of this device was to measure pressure and temperature using 555 timers removing the need to include an ADC. Through a calibration procedure, the part to part variability of the pressure sensor was reduced from $\pm 25\%$ to $\pm 3\%$ of the full scale resolution. A flexible pcb was designed, fabricated, and tested with live subjects producing promising results for further prototypes.

4.1 Future Work

If work is to move forward with this device, further design work and testing will have to be conducted. Namely, the power and wireless transmission of data will have to be addressed and implemented. The device must include solutions for these design aspects in order to remove the need external wiring. This would allow the device to be completely enclosed in a bandage for easy application upon the patient's skin. Additionally, further clinical trials and patient data must be conducted and collected in order to create a sophisticated algorithm for the determination of pressure ulcer risk. Specially, this data is necessary for determining the rate at which pressure measurements and temperature measurements should be taken.

Appendix A

Arduino Code (PulseIn)

```
int pin = 5;
float durationON;
float durationOFF;
float Period;
float Frequency;
float dutyCycle;
float one = 1000000;

long double avgdurationON = 0;
long double avgdurationOFF = 0;
float finaldurationON = 0;
float finaldurationOFF = 0;

int i = 0;
int j = 1;

void setup(){
  Serial.begin(57600);
  pinMode(pin, INPUT);
}
```

```

void loop(){

  durationON = pulseIn(pin, HIGH);
  durationOFF = pulseIn(pin, LOW);

for(int idx = 0; idx < 20; idx++){
  if(i<j) {
    avgdurationON = avgdurationON + durationON;
    avgdurationOFF = avgdurationOFF + durationOFF;
    i++;
  }
  else{
    //Serial.println("Before averaging: " + String(avgR2));
     finaldurationON = avgdurationON/(j);
     finaldurationOFF = avgdurationOFF/(j);
     Period = finaldurationON + finaldurationOFF;
     dutyCycle = (finaldurationON / Period) * 100;
     Frequency = one / Period;
     Serial.println("DuractionON: " + String(finaldurationON));
     Serial.println("DurationOFF: " + String(finaldurationOFF));
     Serial.println("Period: " + String(Period));
     Serial.println("Frequency: " + String(Frequency));
     Serial.println("DutyCycle: " + String(dutyCycle) + "%");
     Serial.println(" ");
    i=0;
    avgdurationON = 0;
    avgdurationOFF = 0;
  }
}
delay(5000000);
}
```

Appendix B

Arduino Code (Interrupt)

```
#define runMode 0
#define calMode 1
#define MainPeriod 100
#define calPeriod 1000
#define onTime555 100

long previousMillis = 0; // will store last time of the cycle end
volatile unsigned long durationRising = 0; // accumulates pulse width
volatile unsigned int pulseCountRising = 0;
volatile unsigned long previousMicrosRising = 0;
volatile unsigned long durationFalling = 0; // accumulates pulse width
volatile unsigned int pulseCountFalling = 0;
volatile unsigned long previousMicrosFalling = 0;

int mode = 0;
boolean calFinished = false;
boolean powerSave = false;
int cal_state = 0;
int onOffCounter = onTime555;

```

```
int lowCalPinState = 0;
int highCalPinState = 0;

static int interruptPin3 = 3;
static int interruptPin2 = 2;
static int lowCalPin = 4;   // pushbutton connected to digital pin 6
static int highCalPin = 5;
static int resetPin555 = 6;
static int redLED = 12;
static int yellowLED = 11;
static int greenLED = 10;

static float Rref = 10000;  //reference resistor
static float Rmax = 300000;  //maximum resistance value (parallel resistor)
float cal_param_1 = 0.6360;   //using nominal values
float cal_param_1_actual = 98.6365462;
float cal_param_2 = 0.5287; //using nominal values
float cal_param_2_actual = 548.6365462;
float dutyCycle_FSR = 0;
float xfit = -1.01291;  //using nominal values
float Rofit = 590476.81;  //using nominal values

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  attachInterrupt(digitalPinToInterrupt(interruptPin3), runModeHandler, CHANGE);
  attachInterrupt(digitalPinToInterrupt(interruptPin2), risinginthandler, CHANGE);
  //using pin D2 on bluno
  pinMode(lowCalPin, INPUT);      // sets the digital pin 6 as input
  pinMode(highCalPin, INPUT);      // sets the digital pin 6 as input
  pinMode(resetPin555, OUTPUT);   // sets the digital pin 7 as output
  pinMode(redLED, OUTPUT);  //red
  pinMode(yellowLED, OUTPUT);  //yellow
  pinMode(greenLED, OUTPUT);  //green
```

```
}

void loop() {
  // put your main code here, to run repeatedly:
  switch (mode) {

    case calMode:
      calFinished = true;
      lowCalPinState = digitalRead(lowCalPin);   // read the lowCal input pin
      highCalPinState = digitalRead(highCalPin);   // read the highCal input pin
      if (cal_state == 0) {        //button has been released after first calibration
      point taken
        cal_state = 1;
        digitalWrite(redLED, HIGH);
        digitalWrite(yellowLED, LOW);
        digitalWrite(greenLED, LOW);
      }
      if (lowCalPinState == HIGH && cal_state == 1) {          //button pressed for first
      calibration point measurement
        cal_param_1 = DutyCycleMeasurement();
        if (cal_param_1 != 0) {
          cal_state = 2;
          Serial.println("Calibration point 1 measured: " + String(cal_param_1));
          digitalWrite(redLED, HIGH);
          digitalWrite(yellowLED, LOW);
          digitalWrite(greenLED, HIGH);
          delay(500);
        }
      }
      if (lowCalPinState == LOW && cal_state == 2) {        //button has been released
      after first calibration point taken
        cal_state = 3;
        Serial.println("Ready to measure calibration point 2");
```

```
      digitalWrite(redLED, LOW);
      digitalWrite(yellowLED, HIGH);
      digitalWrite(greenLED, LOW);
    }
    if (highCalPinState == HIGH && cal_state == 3) {        //button pressed for second
    calibration point measurement
      cal_param_2 = DutyCycleMeasurement();
      if (cal_param_2 != 0) {
        cal_state = 4;
        Serial.println("Calibration point 2 measured: " + String(cal_param_2));
        digitalWrite(redLED, LOW);
        digitalWrite(yellowLED, HIGH);
        digitalWrite(greenLED, HIGH);
        delay(500);
      }
    }
    if (highCalPinState == LOW && cal_state == 4) {        //button has been released after
    second calibration point taken completing the calibration process
      float da = (((1 / Rref) * ((1 - cal_param_1) / (2 * cal_param_1 - 1))) - (1 / Rmax));
      float db = (((1 / Rref) * ((1 - cal_param_2) / (2 * cal_param_2 - 1))) - (1 / Rmax));
      xfit = log(db / da) / log(cal_param_1_actual / cal_param_2_actual);
      Rofit = cal_param_1_actual * pow(da, (1 / xfit));
      Serial.println("Calibration complete with point 1: " + String(cal_param_1, 5) +
      " and point 2: " + String(cal_param_2, 5));
      Serial.println("da: " + String(da, 10));
      Serial.println("db: " + String(db, 10));
      Serial.println("xfit: " + String(xfit, 5));
      Serial.println("Rofit: " + String(Rofit, 10));
      cal_state = 5;
    }
```

```

    while (mode == calMode && cal_state == 5) {
      digitalWrite(redLED, HIGH);
      delay(100);
      digitalWrite(redLED, LOW);
      delay(100);
      digitalWrite(yellowLED, HIGH);
      delay(100);
      digitalWrite(yellowLED, LOW);
      delay(100);
      digitalWrite(greenLED, HIGH);
      delay(100);
      digitalWrite(greenLED, LOW);
      delay(100);
    }
    break;

  case runMode:
    digitalWrite(redLED, LOW);
    digitalWrite(yellowLED, LOW);
    digitalWrite(greenLED, HIGH);
    if (powerSave == true) {
      calFinished = false;
      if (onOffCounter == 0) {
        digitalWrite(resetPin555, LOW);
        onOffCounter++;
      }
      else if (onOffCounter == onTime555) {
        digitalWrite(resetPin555, HIGH);
        for (int idx = 0; idx <= 20; idx++) {
          dutyCycle_FSR = DutyCycleMeasurement();
          float Fpredfit = Rofit * pow((((1 / Rref) * ((1 - dutyCycle_FSR) /
          (2 * dutyCycle_FSR - 1))) - (1 / Rmax)), (-1 / xfit));
          Serial.println("Fpredfit " + String(idx) + ": " + String(Fpredfit, 4));
```

```
            if (mode == calMode) {
              break;
            }
          }
          onOffCounter = 0;
        }
        else {
          onOffCounter++;
          Serial.println("Saving Power: " + String(onOffCounter));
          delay(100);
        }
      }
      else {
        for(int i = 1; i < 301; i++){
        calFinished = false;
        digitalWrite(resetPin555, HIGH);
        dutyCycle_FSR = DutyCycleMeasurement();
        float Fpredfit = Rofit * pow((((1 / Rref) * ((1 - dutyCycle_FSR) /
        (2 * dutyCycle_FSR - 1))) - (1 / Rmax)), (-1 / xfit));
        Fpredfit = Fpredfit * 0.579;
        Serial.println(String(i) + "   " + String(Fpredfit, 5) + " " +
        String(dutyCycle_FSR, 5));
        }
        delay(20000);
      }
      break;
  }
}

void runModeHandler() {
  int temp = digitalRead(3);
  if (temp == HIGH && calFinished == false) {
    mode = calMode;
    cal_state = 0;
```

```
    Serial.println("CalMode!");
  }
  else if (temp == LOW && calFinished == true) {
    mode = runMode;
    Serial.println("RunMode!");
  }
}

float DutyCycleMeasurement() {
  if (mode == runMode) {
    unsigned long currentMillis = millis();
    boolean measuring = true;
    float dutyCycle;
    while (measuring) {
      currentMillis = millis();
      if (currentMillis - previousMillis >= MainPeriod)       //this waits until
      the MainPeriod time has past. This allows for a sufficient number of
      measurements to be taken
      {
        previousMillis = currentMillis;
        // need to bufferize to avoid glitches
        unsigned long _durationRising = durationRising;
        unsigned long _pulseCountRising = pulseCountRising;
        unsigned long _durationFalling = durationFalling;
        unsigned long _pulseCountFalling = pulseCountFalling;
        durationRising = 0; // clear counters
        pulseCountRising = 0;
        durationFalling = 0; // clear counters
        pulseCountFalling = 0;
        float riseTime = float(_durationRising) / float(_pulseCountRising);
        float fallTime = float(_durationFalling) / float(_pulseCountFalling);
        dutyCycle = (fallTime / (fallTime + riseTime));
        measuring = false;
      }
```

```
  }
  return dutyCycle;
}

if (mode == calMode) {
  unsigned long currentMillis = millis();
  int i = 0;
  float dutyCycle;
  while (i <= calPeriod) {
    currentMillis = millis();
    if (currentMillis - previousMillis >= MainPeriod)        //this waits until
    the MainPeriod time has past. This allows for a sufficient number of
    measurements to be taken
    {
      previousMillis = currentMillis;
      // need to bufferize to avoid glitches
      unsigned long _durationRising = durationRising;
      unsigned long _pulseCountRising = pulseCountRising;
      unsigned long _durationFalling = durationFalling;
      unsigned long _pulseCountFalling = pulseCountFalling;
      durationRising = 0; // clear counters
      pulseCountRising = 0;
      durationFalling = 0; // clear counters
      pulseCountFalling = 0;
      float riseTime = float(_durationRising) / float(_pulseCountRising);
      float fallTime = float(_durationFalling) / float(_pulseCountFalling);
      dutyCycle = dutyCycle + (fallTime / (fallTime + riseTime));
      Serial.println("DutyCycle: " + String(dutyCycle,5 ));
      i++;
    }
  }
  dutyCycle = dutyCycle / calPeriod;
  return dutyCycle;
}
```

```
}

void risinginthandler() // rising interrupt handler
{
  unsigned long currentMicros = micros();                    //has a resolution of 4
  microseconds (always returns in multiples of 4). Will go back to zero after 70 minutes
  if ( digitalRead(2) == HIGH) {
    durationRising += currentMicros - previousMicrosFalling; //determine the amount of
    time that has passed since the last measured pulse
    previousMicrosRising = currentMicros;
    pulseCountRising++;                               //count the number of pulses seen
  }
  else {
    durationFalling += currentMicros - previousMicrosRising; //determine the amount of
    time that has passed since the last measured pulse
    previousMicrosFalling = currentMicros;
    pulseCountFalling++;                               //count the number of pulses seen
  }
}
```

Appendix C

MATLAB Calibration Code

```
%Actual Vs Estimated Residual
%12th Feb
%Pressure Ulcer
clear

% nominal parameters for fit
% R scale factor
Ro=200000;
% exponent
x=-0.738;
% Reference resistor
Rref=10000;
% parallel (max) resistor
Rmax=1000000;

Nsensors=4;
Nforces=10;
% Dimension array to hold all force readings to plot for all sensors
F=zeros(Nsensors,Nforces);
% F array in general with N sensors
```

```
%
%     --- colunns are number of weight readings to keep -->
%     [  - - -   sensor 1 readings     - - -      ]
%     [  - - -   sensor 2 readings     - - -      ]
% F = [                  :                        ]
%     [                  :                        ]
%     [  - - -   sensor N readings     - - -      ]

% READ IN DATA FROM .csv FILES

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% FIRST DATA SET
kd=1                   % data identifier
force_range=[1:10];    % range of weights to keep MUST BE SAME NUMBER OF POINTS
read_range=[40:50];    % range of readings to average - can be different
pa=3;pb=9;             % indices of points to use in 2 point fit for this kd

% force: row vector of forces applied
Fin = xlsread('Work_19Feb/BlockReadings_19Feb_sensor2.csv','','B1:P1');
% duty cycle measurements
DutyCyclein = xlsread('Work_19Feb/BlockReadings_19Feb_sensor2.csv','','B2:P60');
% Remove  points above 600 g (350 mmHg)
F(kd,1:length(force_range))=Fin(force_range);
DutyCycle = DutyCyclein(:,force_range);
% Duty cycle to plot is average of 90 - 100 (after eyeballing plot)
d(kd,:)=mean(DutyCycle(read_range,:));
% get max and min for error bars
dmax(kd,:)=max(DutyCycle(read_range,:));
dmin(kd,:)=min(DutyCycle(read_range,:));
% Predicted force from duty cycle, NOMINAL PARAMETERS
% Fpred1 all one equation
Fpred(kd,:)=(Ro*((1/Rref)*((1-d(kd,:))./(2*d(kd,:)-1))-(1/Rmax))).^(-1/x);

% 2-point fit
```

```
da=((1/Rref)*((1-d(kd,pa))/(2*d(kd,pa)-1))-(1/Rmax));
db=((1/Rref)*((1-d(kd,pb))/(2*d(kd,pb)-1))-(1/Rmax));
Fa=F(kd,pa);
Fb=F(kd,pb);
% Fit parameters
xfit(kd)=(log(db/da)/log(Fa/Fb))
Rofit(kd)=Fa*da^(1/xfit)

% Predicted from fit data
Fpredfit(kd,:)=   Rofit(kd)*((1/Rref)*((1-   d(kd,:))./(2*   d(kd,:)-1))-(1/Rmax)).^(-1/xfit(kd));
% max and min for error bars
Fpredfitmax(kd,:)=Rofit(kd)*((1/Rref)*((1-dmax(kd,:))./(2*dmax(kd,:)-1))-(1/Rmax)).^(-1/xfit(kd));
Fpredfitmin(kd,:)=Rofit(kd)*((1/Rref)*((1-dmin(kd,:))./(2*dmin(kd,:)-1))-(1/Rmax)).^(-1/xfit(kd));



% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% SECOND DATA SET
kd=2                   % data identifier
force_range=[1:10];    % range of weights to keep MUST BE SAME NUMBER OF POINTS
read_range=[40:50];    % range of readings to keep - can be different
pa=3;pb=9;             % indices of points to use in 2 point fit for this kd

% force: row vector of forces applied
Fin = xlsread('Work_19Feb/BlockReadings_19Feb_sensor3.csv','','B1:P1');
% duty cycle measurements
DutyCyclein = xlsread('Work_19Feb/BlockReadings_19Feb_sensor3.csv','','B2:P60');
% Remove  points above 600 g (350 mmHg)
F(kd,1:length(force_range))=Fin(force_range);
```

```
DutyCycle = DutyCyclein(:,force_range);
% Duty cycle to plot is average of 90 - 100 (after eyeballing plot)
d(kd,:)=mean(DutyCycle(read_range,:));
% get max and min for error bars
dmax(kd,:)=max(DutyCycle(read_range,:));
dmin(kd,:)=min(DutyCycle(read_range,:));
% Predicted force from duty cycle, NOMINAL PARAMETERS
% Fpred1 all one equation
Fpred(kd,:)=(Ro*((1/Rref)*((1-d(kd,:))./(2*d(kd,:)-1))-(1/Rmax))).^(-1/x);

% 2-point fit
da=((1/Rref)*((1-d(kd,pa))/(2*d(kd,pa)-1))-(1/Rmax));
db=((1/Rref)*((1-d(kd,pb))/(2*d(kd,pb)-1))-(1/Rmax));
Fa=F(kd,pa);
Fb=F(kd,pb);
% Fit parameters
xfit(kd)=(log(db/da)/log(Fa/Fb))
Rofit(kd)=Fa*da^(1/xfit(kd))

% Predicted from fit data
Fpredfit(kd,:)=   Rofit(kd)*((1/Rref)*((1-   d(kd,:))./(2*   d(kd,:)-1))-(1/Rmax)).^(-1/xfit(kd));
% max and min for error bars
Fpredfitmax(kd,:)=Rofit(kd)*((1/Rref)*((1-dmax(kd,:))./(2*dmax(kd,:)-1))-(1/Rmax)).^(-1/xfit(kd));
Fpredfitmin(kd,:)=Rofit(kd)*((1/Rref)*((1-dmin(kd,:))./(2*dmin(kd,:)-1))-(1/Rmax)).^(-1/xfit(kd));



% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% THIRD DATA SET
```

```
kd=3                  % data identifier
force_range=[1:10];   % range of weights to keep MUST BE SAME NUMBER OF POINTS
read_range=[40:50];   % range of readings to keep - can be different
pa=3;pb=9;            % indices of points to use in 2 point fit for this kd

% force: row vector of forces applied
Fin = xlsread('Work_19Feb/BlockReadings_19Feb_sensor4.csv','','B1:P1');
% duty cycle measurements
DutyCyclein = xlsread('Work_19Feb/BlockReadings_19Feb_sensor4.csv','','B2:P60');
% Remove  points above 600 g (350 mmHg)
F(kd,1:length(force_range))=Fin(force_range);
DutyCycle = DutyCyclein(:,force_range);
% Duty cycle to plot is average of 90 - 100 (after eyeballing plot)
d(kd,:)=mean(DutyCycle(read_range,:));
% get max and min for error bars
dmax(kd,:)=max(DutyCycle(read_range,:));
dmin(kd,:)=min(DutyCycle(read_range,:));
% Predicted force from duty cycle, NOMINAL PARAMETERS
% Fpred1 all one equation
Fpred(kd,:)=(Ro*((1/Rref)*((1-d(kd,:))./(2*d(kd,:)-1))-(1/Rmax))).^(-1/x);

% 2-point fit
da=((1/Rref)*((1-d(kd,pa))/(2*d(kd,pa)-1))-(1/Rmax));
db=((1/Rref)*((1-d(kd,pb))/(2*d(kd,pb)-1))-(1/Rmax));
Fa=F(kd,pa);
Fb=F(kd,pb);
% Fit parameters
xfit(kd)=(log(db/da)/log(Fa/Fb))
Rofit(kd)=Fa*da^(1/xfit(kd))

% Predicted from fit data
Fpredfit(kd,:)=   Rofit(kd)*((1/Rref)*((1-   d(kd,:))./(2*   d(kd,:)-1))-(1/Rmax)).^(-1/
    xfit(kd));
% max and min for error bars
```

```
Fpredfitmax(kd,:)=Rofit(kd)*((1/Rref)*((1-dmax(kd,:))./(2*dmax(kd,:)-1))-(1/Rmax)).^(-1/
    xfit(kd));
Fpredfitmin(kd,:)=Rofit(kd)*((1/Rref)*((1-dmin(kd,:))./(2*dmin(kd,:)-1))-(1/Rmax)).^(-1/
    xfit(kd));



%
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% FOURTH DATA SET
kd=4                    % data identifier
force_range=[1:10];     % range of weights to keep MUST BE SAME NUMBER OF POINTS
read_range=[40:50];     % range of readings to keep - can be different
pa=3;pb=9;              % indices of points to use in 2 point fit for this kd

% force: row vector of forces applied
Fin = xlsread('Work_19Feb/BlockReadings_19Feb_sensor5.csv','','B1:P1');
% duty cycle measurements
DutyCyclein = xlsread('Work_19Feb/BlockReadings_19Feb_sensor5.csv','','B2:P60');
% Remove  points above 600 g (350 mmHg)
F(kd,1:length(force_range))=Fin(force_range);
DutyCycle = DutyCyclein(:,force_range);
% Duty cycle to plot is average of 90 - 100 (after eyeballing plot)
d(kd,:)=mean(DutyCycle(read_range,:));
% get max and min for error bars
dmax(kd,:)=max(DutyCycle(read_range,:));
dmin(kd,:)=min(DutyCycle(read_range,:));
% Predicted force from duty cycle, NOMINAL PARAMETERS
% Fpred1 all one equation
Fpred(kd,:)=(Ro*((1/Rref)*((1-d(kd,:))./(2*d(kd,:)-1))-(1/Rmax))).^(-1/x);

% 2-point fit
da=((1/Rref)*((1-d(kd,pa))/(2*d(kd,pa)-1))-(1/Rmax));
```

```matlab
db=((1/Rref)*((1-d(kd,pb))/(2*d(kd,pb)-1))-(1/Rmax));
Fa=F(kd,pa);
Fb=F(kd,pb);
% Fit parameters
xfit(kd)=(log(db/da)/log(Fa/Fb))
Rofit(kd)=Fa*da^(1/xfit(kd))

% Predicted from fit data
Fpredfit(kd,:)=   Rofit(kd)*((1/Rref)*((1-   d(kd,:))./(2*   d(kd,:)-1))-(1/Rmax)).^(-1/xfit(kd));
% max and min for error bars
Fpredfitmax(kd,:)=Rofit(kd)*((1/Rref)*((1-dmax(kd,:))./(2*dmax(kd,:)-1))-(1/Rmax)).^(-1/xfit(kd));
Fpredfitmin(kd,:)=Rofit(kd)*((1/Rref)*((1-dmin(kd,:))./(2*dmin(kd,:)-1))-(1/Rmax)).^(-1/xfit(kd));

% Plot all data

figure(901)
plot(DutyCycle)
xlabel('Reading')
ylabel('Duty Cycle')
title('all duty cycle readings - look and check for strange behavior')

figure (1)
hold off
plot(F(kd,:),d(kd,:),'o')
hold on
plot(Fpred(kd,:),d(kd,:))
xlabel('Force')
ylabel('Duty Cycle')
```

```
title('Predicted force from duty cycle, nominal Ro=200K, x=-0.738')

figure (2)
hold off
loglog(F(kd,:),d(kd,:),'o')
hold on
loglog(Fpred(kd,:),d(kd,:))
xlabel('Force')
ylabel('Duty Cycle')
title('Predicted force from duty cycle, nominal Ro=200K, x=-0.738')




figure(9)
hold off
plot(F,Fpredfit(kd,:),'o')
hold on
plot(F,Fpred(kd,:),'*')
xlabel('Force')
ylabel('calc Force')
title('Predicted force from duty cycle, nominal Ro=200K, x=-0.738')

figure (10)
hold off
loglog(F,Fpredfit(kd,:),'o')
hold on
loglog(F,Fpred(kd,:),'o')
xlabel('Force')
ylabel('calc Force')
title('Predicted force from duty cycle, nominal Ro=200K, x=-0.738')


figure(11)
```

```
hold off
plot(F(kd,:),Fpredfit(kd,:)-F(kd,:),'o')
hold on
%plot(F,Fpred2,'*')
xlabel('Force')
ylabel('calc Force error')
title('Predicted force from duty cycle, nominal Ro=200K, x=-0.738')

% Time stamp so output file can be uniquely identified
savestr=datestr(now,'yyyy-mm-dd-HH-MM-SS')


%%%%%%%%%%%%%%%%%%DISPLAY RESULTS: LINEAR AXES
%
% Select which of the following to use
fnum=802;
figure(fnum)
clf
set(fnum, 'PaperOrientation', 'Portrait');
set(fnum, 'PaperUnits', 'inches');
set(fnum, 'PaperPosition', [1 0.5 7 10]);
set(fnum, 'Units', 'inches');
set(fnum, 'Position', [5.7 0.02 7 10]);
axes_height=.14;    % try different values to get best fit in paper
axes_spacing=0.14;
axes_offset=0.05;

% Normalize force data to pressure
P=0.579*F;
Ppred=0.579*Fpred;
Ppredfit=0.579*Fpredfit;
% Calculate error bars from fractional errors, readings
% Factor of 0.5 since drawn bar is +/- value, and difference is pk-to-pk
Perrorbars=(0.5)*(0.579)*(Fpredfitmax-Fpredfitmin);
```

```

%%%%%%%%%%%%%%%%
%
% Plot pressure data
axes('OuterPosition',[0 3.9*axes_spacing+axes_offset 1 2.5*axes_height])
hold off
kd=1
% Uncalibrated data
plot(P(kd,:),Ppred(kd,:),'--o','Color',[0.5 0.5 0.5],'LineWidth',2)
hold on
errorbar(P(kd,:),Ppredfit(kd,:),Perrorbars(kd,:),'-o','Color',[0.5 0.5 0.5],'LineWidth'
    ,2)
kd=2
plot(P(kd,:),Ppred(kd,:),'--o','Color',[1 0  0  ],'LineWidth',2)
errorbar(P(kd,:),Ppredfit(kd,:),Perrorbars(kd,:),'-o','Color',[1 0 0 ],'LineWidth',2)
kd=3
plot(P(kd,:),Ppred(kd,:),'--o','Color',[0 .8  0  ],'LineWidth',2)
errorbar(P(kd,:),Ppredfit(kd,:),Perrorbars(kd,:),'-o','Color',[0 .8 0 ],'LineWidth',2)
kd=4
plot(P(kd,:),Ppred(kd,:),'--o','Color',[0 0 1  ],'LineWidth',2)
errorbar(P(kd,:),Ppredfit(kd,:),Perrorbars(kd,:),'-o','Color',[0 0 1 ],'LineWidth',2)

% Annotate
text(10,450,'UNCALIBRATED              ', ...
    'BackgroundColor',[1 1 1],'FontSize',14)
plot([120 160],[450 450],'--','Color',[0 0 0 ],'LineWidth',2)
text(190,100,' CALIBRATED                ', ...
    'BackgroundColor',[1 1 1],'FontSize',14)
plot([290 330],[100 100],'-','Color',[0 0 0 ],'LineWidth',2)


%xlabel('APPLIED FORCE [g]','FontSize',16)
set(gca,'XTick',0:50:500)
```

```
%set(gca,'XTickLabel',{'0',' ','100',' ','200',' ','300',' ','400',' ','500'},'FontSize',14)
% Following for no labels since just above error plot
set(gca,'XTickLabel',{' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' '},'FontSize',14)
set(gca,'XLim',[0 370])

ylabel([ 'CALCULATED PRESSURE'; ...
         '       [mmHg]      ' ],'FontSize',16)
set(gca,'YTick',0:50:500)
set(gca,'YTickLabel',{'0',' ','100',' ','200',' ','300',' ','400',' ','500'},'FontSize',14)
set(gca,'YLim',[0 500])
grid on

% Maximum full scale value for reporting % error
Pmax=350;
%%%%%%%%%%%%%%%%
%
% Plot error
axes('OuterPosition',[.01 3.05*axes_spacing+axes_offset-.08 .99 1.3*axes_height])
hold off
kd=1;
plot(P(kd,:),(100/Pmax)*(Ppredfit(kd,:)-P(kd,:)),'-o','Color',[ 0.5 0.5 0.5 ],'LineWidth',2)
hold on
kd=2;
plot(P(kd,:),(100/Pmax)*(Ppredfit(kd,:)-P(kd,:)),'-o','Color',[1 0 0 ],'LineWidth',2)
kd=3
plot(P(kd,:),(100/Pmax)*(Ppredfit(kd,:)-P(kd,:)),'-o','Color',[0 .8 0 ],'LineWidth',2)
kd=4
plot(P(kd,:),(100/Pmax)*(Ppredfit(kd,:)-P(kd,:)),'-o','Color',[0 0 1 ],'LineWidth',2)

%plot(0:20:500,2*randn(1,26),'Color',[0 0 1 ],'LineWidth',2)

```

```
xlabel('APPLIED PRESSURE [mmHg]','FontSize',16)
set(gca,'XTick',0:50:350)
set(gca,'XTickLabel',{'0',' ','100',' ','200',' ','300',' ','400'},'FontSize',14)
set(gca,'XLim',[0 370])

ylabel(['ERROR';'% FS '],'FontSize',16)
set(gca,'YTick',-3:1:+3)
set(gca,'YTickLabel',{' ',' -2',' ',' 0',' ',' +2',' ',' +4'},'FontSize',14)
set(gca,'YLim',[-3.1 3.1])
grid on

% % % Save in .fig and .pdf formats
%saveas(gcf,[ 'results_' savestr '.fig' ])
%saveas(gcf,[ 'results_' savestr '.pdf' ])
```

Appendix D

Pig Experiment Arduino Code

```
#include "ecPlatform.h"
#include "ECPParser.h"
#include "ecUtil.c"

ecPlatform platform(&Serial); //The argument here can be any valid Serial port. On UNOs
    there is only one port.
ECPParser parser;
ECPMessage lastMsg;
char serialBuffer[64];
int bytesRead = 0;

int pin[] = {50, 52, 40, 38, 36, 22};
//float frequency[] = {0, 0, 0, 0, 0, 0};
//float dutyCycle[] = {0, 0, 0, 0, 0, 0};
int initialReading[] = {0, 0, 0, 0, 0, 0};
int currentStatePin[] = {0, 0, 0, 0, 0, 0};
int previousStatePin[] = {0, 0, 0, 0, 0, 0};
//float risingEdge[21] = {0, 0, 0, 0, 0, 0, 0, 0, 0, 0,0,0,0,0,0,0,0,0,0,0};
//float fallingEdge[20] = {0, 0, 0, 0, 0, 0, 0, 0, 0, 0,0,0,0,0,0,0,0,0,0,0};
unsigned long risingEdge[101];
```

```
unsigned long fallingEdge[101];
unsigned long sumRising = 0;
unsigned long sumFalling = 0;

int numSignalsMeasure = 6;
int counter = 0;
int numPeriods = 100;

String fallingEdgeString;
String risingEdgeString;

int pinReset = 12;
int timeoutFlag = 0;
unsigned long timeout = 487800;  //timeout time in micros (corresponds to 200 periods at
    410Hz)
int refreshflag = 0;
String refreshflagName = "refreshflag";

void setup()
{
  Serial.begin(9600);
    pinMode(pinReset, OUTPUT);
    digitalWrite(pinReset, HIGH);
  for (int i = 0; i < numSignalsMeasure; i++) {
    pinMode(pin[i], INPUT);
  }
  for (int i = 0; i < numSignalsMeasure; i++) {
    int statePin = digitalRead(pin[i]);
    previousStatePin[i] = statePin;
  }
  platform.sendDataItem(refreshflagName, refreshflag);
//    Serial.println(" ");
}

```

```
void loop()
{
    for (int i = 0; i < numSignalsMeasure; i++) {
      unsigned long timeoutMicros = micros();

      while (counter < numPeriods && timeoutFlag != 1) {
        unsigned long currentMicros = micros();
        //Serial.println("CurrentMicros: " + String(currentMicros));
        currentStatePin[i] = digitalRead(pin[i]);
        if (currentStatePin[i] != previousStatePin[i]) {
          if (currentStatePin[i] == 1) {
            if (initialReading[i] == 1) {
              counter++;
              risingEdge[counter] = currentMicros;
            }
            else {
              initialReading[i] = 1;
              risingEdge[counter] = currentMicros;
              //counter++;
            }
          }
          else {
            if (initialReading[i] == 1) {
              fallingEdge[counter] = currentMicros;
            }
            else {
            }
          }
          previousStatePin[i] = currentStatePin[i];
        }
        if((currentMicros - timeoutMicros) > timeout){
          timeoutFlag = 1;
          if ((i % 2) == 0){
            i++;
```

```
          }
        }
      }
    for (int j = 0; j < numPeriods; j++) {
      sumRising = sumRising + risingEdge[j];
      sumFalling = sumFalling + fallingEdge[j];
    }
    float dutyCycle = ((sumFalling - sumRising) / (float)(risingEdge[numPeriods] -
    risingEdge[0]));

    float frequency = (numPeriods * 1E6) / (risingEdge[numPeriods] - risingEdge[0]);

    String propNameFrequency = "Pin_" + String(pin[i]) + "_Frequency";
    String propNameDutyCycle = "Pin_" + String(pin[i]) + "_DutyCycle";

    if(timeoutFlag == 1){
      frequency = -10;
      dutyCycle = -10;
    }
    platform.sendDataItem(propNameFrequency, frequency);
    Serial.println(" ");
    platform.sendDataItem(propNameDutyCycle, dutyCycle);
    Serial.println(" ");

    fallingEdgeString = " ";
    risingEdgeString = " ";
    counter = 0;
    initialReading[0] = 0;
    sumRising = 0;
    sumFalling = 0;
    timeoutFlag = 0;
}
    refreshflag = 1;
    platform.sendDataItem(refreshflagName, refreshflag);
```

```
    platform.sendDataItem("test", 1);
    swhwReset();
}

void swhwReset() {
  delay(100);
  digitalWrite(pinReset, LOW);
}
```

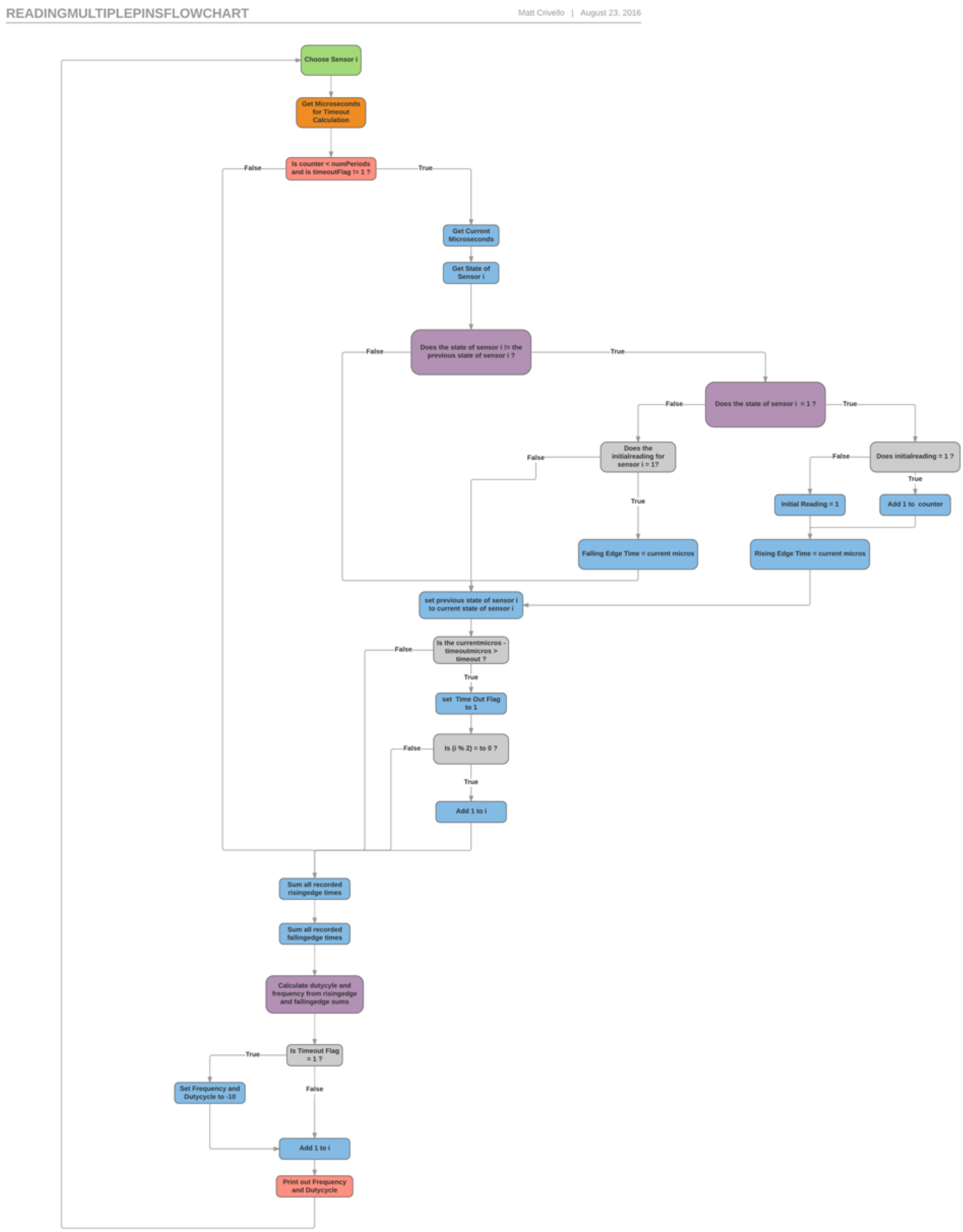


Figure D.1: Flowchart of the Arduino Code used in the Pig Experiment

Appendix E

PCB Documentation (Electrical Prototype)

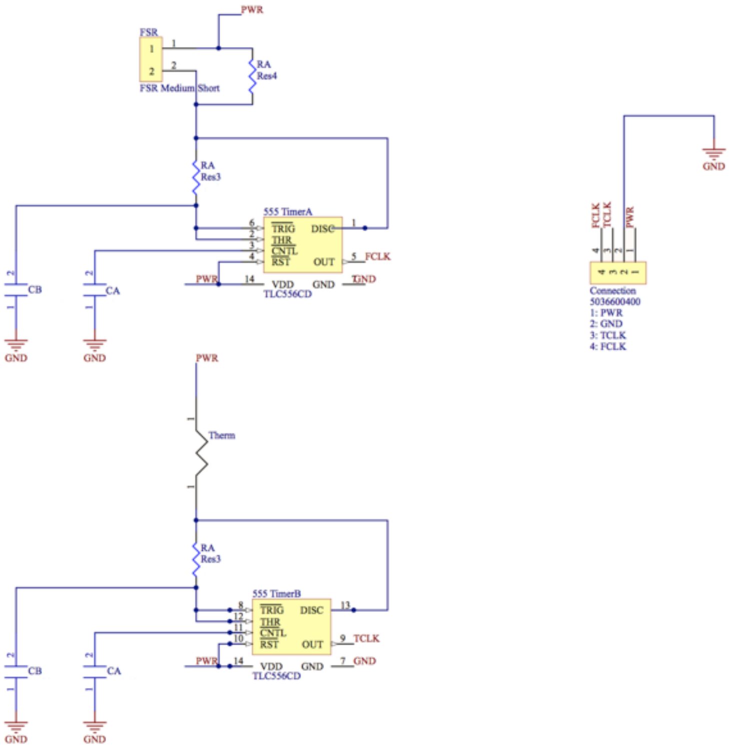


Figure E.1: Electrical Prototype Circuit Diagram

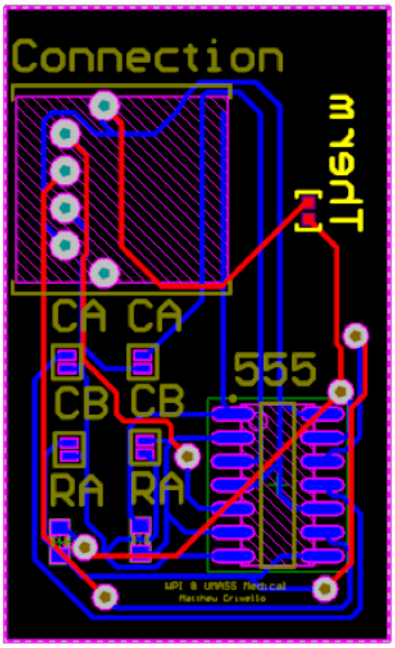


FIGURE E.2: Electrical Prototype PCB (net routes)

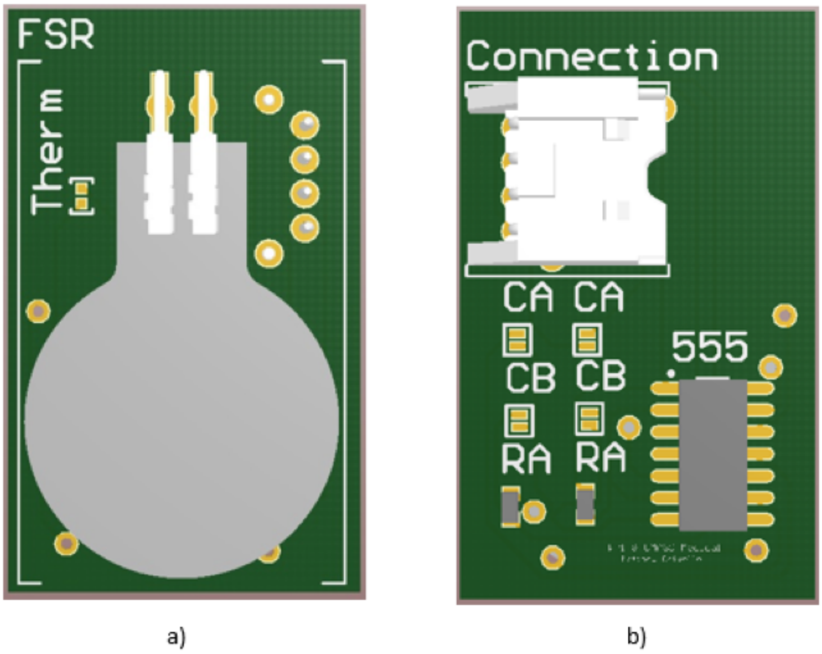


FIGURE E.3: Electrical Prototype 3D Rendering where a) is the Front and b) is the back

Appendix F

PCB Documentation (Flex Design)

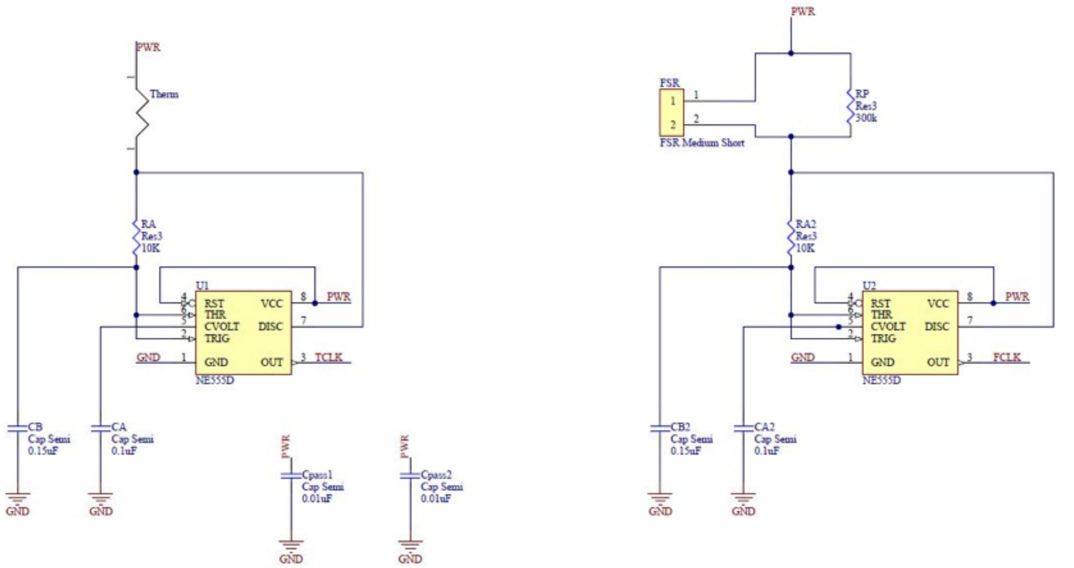


Figure F.1: Flex PCB Design Circuit Diagram

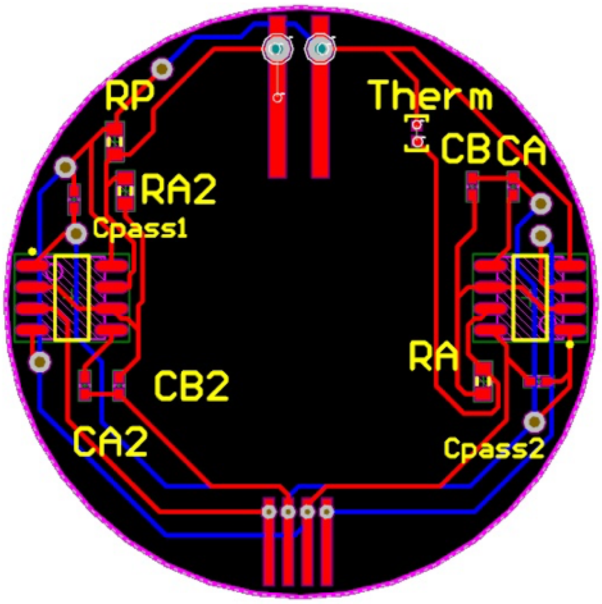


FIGURE F.2: Flex PCB Prototype (net routes)

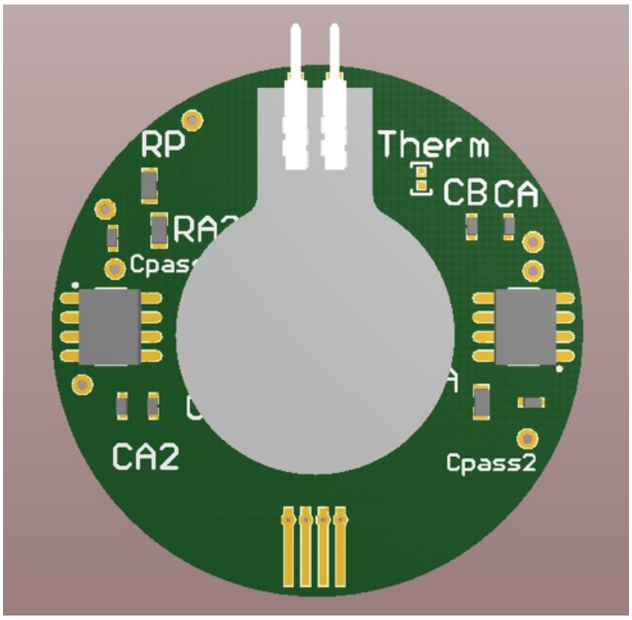


FIGURE F.3: Flex PCB Prototype 3D Rendering

PCB UNIVERSE QUOTE

Company : Worcester Polytechnic Institute
Contact : Matthew Crivello
Part Number : Thesis Flex Rev V3
Email : mdcrivello@wpi.edu
Phone : (508) 831-5000
Description : Flex PCB

Real time PCB quoting and ordering

Incredible Pricing
High Quality
Fast Turn Times
Even Faster Quotes
Aluminum
Flex / Rigid Flex

Always Free Tooling and Free Electrical Test!

We make it easy to move your legacy parts to us. We will match your stencil so you can start saving money now!

RoHS Compliant
UL & ISO 9001 Certified
Up to 42 Layers
Free DFM File Review
Prototypes to
Full Production

Assembly Specifications

Place your order as late as 9pm EST / 6pm PST !

Quote Number EW160505-1002 Date 5/24/2016 10:34
Gerber File Name Manufacturer Files.zip

PCB Quoted Specifications

Layers	2
Board Size	1.377 x 1.377 in
Array	No
Scoring	No
Jump Scoring	No
Tab Rout	No
Material	Polyimide (Flex)
Final Thickness	0.008in / 0.20mm
Plating	Immersion Gold
Outer Cu Weight	1 oz
# of Stiffeners	1w/m.k
Sides Cover Lay	Both
Cover Lay Color	Green
Sides Silkscreen	Top
Silkscreen Color	White
Gold Fingers	No
Min Trace/Space	10mil / 0.25mm
Min Hole Size	15mil / 0.38mm
# Holes Under 12 mil	0
Plated Slots	No
Rout Method	Yes

RoHS	Yes
ITAR	No
Class 2	Yes
Class 3	No
UL/Date Code/94V-0 Mark	Not Required
Counter Sinks/Bores	No
Controlled Dielectric	No
Castellated Holes	No
Carbon Ink	No
Blind/Buried Vias	No
Via in Pad	No
Tented Vias	No
Mask Plugged Vias	No
Conductive Filled Vias	No
Controlled Impedance	No

Special Notes

New files received on 5/24/2016

Lead Times in Business Days (Mon-Fri)

Bare PCB LOT Price

Qty	8 Day							
5	$450.27							
10	$523.77							
20	$661.77							

Bare Board Tooling: ~~$150~~ *(Free)* *Bare Board Electrical Test:* ~~$150~~ *(Free)*

Assembly Labor Pricing Per Lot						Additional Costs (per Lot)		
Qty	3 Day	5 Day				Parts		
5	$600.00	$540.00				$138.82		
10	$750.00	$675.00				$205.14		
20	$975.00	$825.00				$361.49		

Assembly NRE: $0

Free Bare Board Tooling and Electrical Test!

Plus Free DFM (Design For Manufacturing) Review on Every Order!

PCB Universe Acct Mgr Eric Willson
11818 SE Mill Plain Blvd. Phone 303-432-1476
Suite 208 e-Fax 503-296-2925
Vancouver, WA 98684 USA email eric@pcbuniverse.com

- Quote valid for 30 days.
- Daily order cut off time is 9pm EST / 6pm PST. Day zero begins once files are approved by engineering.
- Costs and lead times do not include shipping. Standard UPS Domestic Rates Apply.
- If certain specs were not specifically stated in your files, we assumed our industry standard specs. If you see any specs that are inaccurate, let us know immediately. It may not only affect the cost but also how your boards are constructed ! 0 v-16.05.04

FIGURE F.4: Flex PCB Quote from PCB Universe [30]

Appendix G

Pig Experiment Setup Diagram (Multiple Subjects)

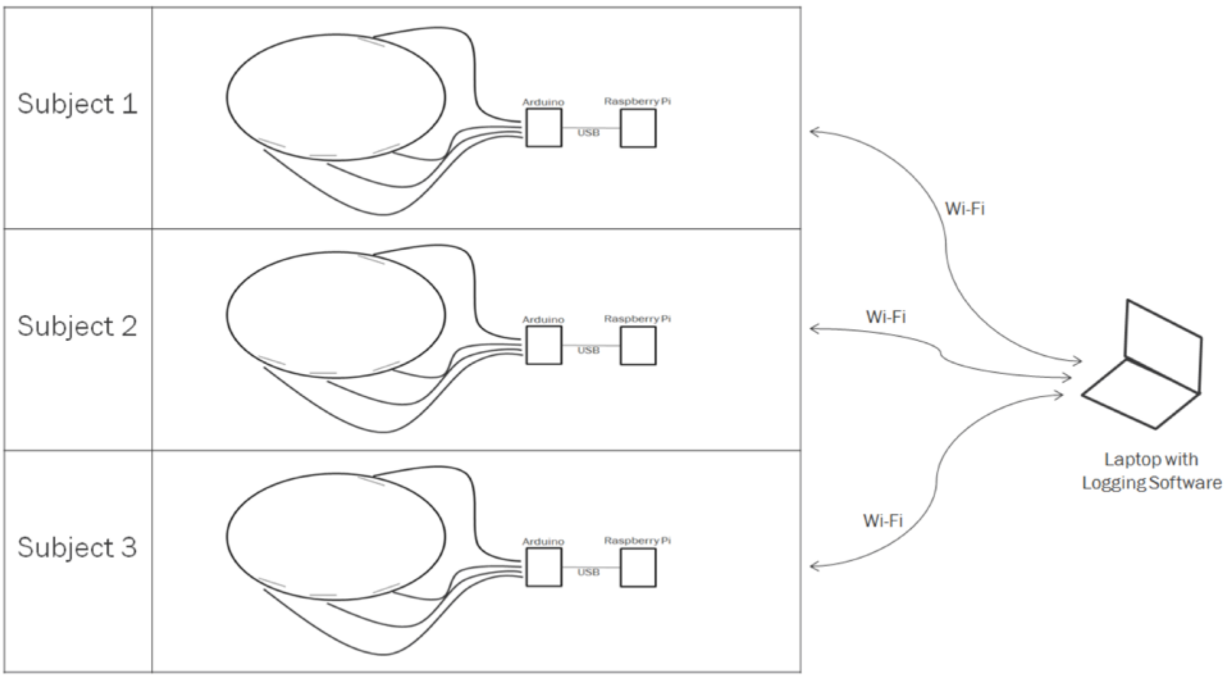


FIGURE G.1: Diagram of the Experiment Setup for Three Subjects

Appendix H

NEBEC Conference Paper

Modeling of Force Sensor Nonlinearity for Time-Domain-Based Pressure Measurement in Biomedical Sensors

Matthew Crivello, Devdip Sen, John McNeill, Yitzhak Mendelson, Raymond Dunn, M.D., Kelli Hickle, M. D.
Worcester Polytechnic Institute
100 Institute Rd, Worcester, MA 01609 e-mail: mcneill@wpi.edu
University of Massachusetts Medical School
55 N Lake Ave, Worcester, MA 01655

***Abstract*—An interfacing technique for resistive force sensors allows determination of resistance using a time-based measurement approach. Modeling and correction of force sensor nonidealities enables measurement accuracy of $\pm3\%$ after calibration over a pressure range of 20 to 350 mmHg.**

I. Introduction

Accurate measurement of force and/or pressure is required in many biomedical applications such as fingertip pressure in an active exoskeleton [1], automated personal safety equipment [2], and rehabilitation [3]. This paper describes a measurement technique applicable to resistive pressure sensors such as [4]. Unlike voltage-based techniques requiring a resistive divider and analog-to-digital converter, the time-based technique described in this work has the advantages of being more "digital-friendly" and allowing reconfigurable system accuracy by increasing measurement time.

This paper is organized as follows: Section II provides background information on the sensor and a brief review of previous measurement techniques. Section III provides an overview of the proposed time-based measurement technique, with measured results provided in section IV.

II. Background

A. Resistive Force Sensor

The basis of Polymer Thick Film (PTF) sensing is the increased conductance of a polymer layer subjected to a compressive force. Figure 1 shows the force-conductance characteristic for the FSR-40 sensor [4], with the region of forces between 30g - 600g highlighted. Given the sensor active area of $A = 1.27\text{cm}^2$, this region corresponds to a range of pressures from 20 - 350 mm Hg, as expected in applications such as [1]–[3]. Over this range, the resistance-to-force F (in grams) relationship shows a power-law characteristic [4] as

$$R_{FSR} = R_0 F^x \tag{1}$$

Thus a measurement of resistance R can be used to determine force on the sensor using the model of (1). A least-squares fit in log-log space for the nominal data in Fig. 1 gives values of $R_0 = 200k\Omega$ and $x = -0.738$.

A challenge associated with this type of pressure sensor is part-to-part variability of up to $\pm25\%$, indicated by the dashed lines of Fig. 1 [4]. The calibration procedure described in Section III enables linear force measurement with accuracy approaching the single-part repeatability of $\pm2\%$.

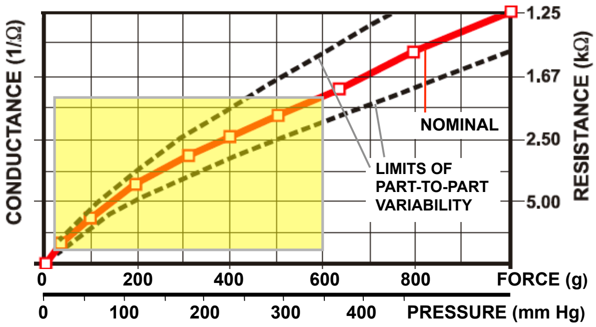


Fig. 1. Conductance vs. force for FSR-402 sensor [4].

B. Voltage-based Resistance Measurement

In a voltage-based resistance measurement technique [4] the force sensing resistor R_{FSR} is placed in a voltage divider configuration with a reference resistance. With a known reference voltage applied, the voltage divider output is digitized by an analog-to-digital converter (ADC). The value of R_{FSR} can be inferred from the measured voltage using the voltage divider relationship. With the value of R_{FSR}, (1) can be used to determine force, and pressure can be determined using the sensor active area. Disadvantages of this approach include the need for a reference voltage and accurate ADC, each of which introduces additional error and power dissipation.

III. System Design

A. Time-based Resistance Measurement

Figure 2 shows the proposed time-based resistance measurement technique, based on the LM555 timer. For the output digital waveform DCLK, the frequency f and duty cycle $\delta = T_H/T$ (fractional "high" time T_H relative to the waveform period T) are given by

$$f = \frac{1.44}{(R_A + 2R_B)C} \qquad \delta = \left(\frac{R_A + R_B}{R_A + 2R_B}\right) \tag{2}$$

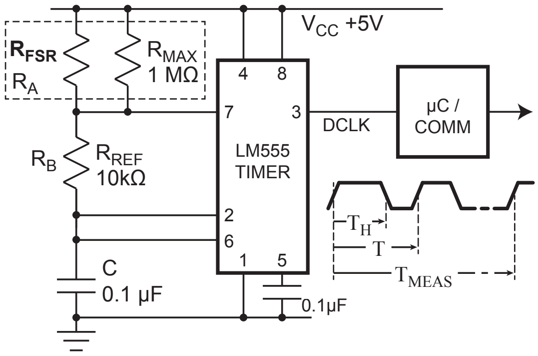


Fig. 2. System block diagram with time-based resistance measurement.

Either expression in (2) could be used to determine resistance from a time domain measurement. Duty cycle is chosen since it is independent of the capacitor value C. The duty cycle δ is calculated by the microcontroller, which measures T_H and T in the digital domain over interval T_{MEAS}, covering many T periods. The microcontroller also implements the calibration algorithm described below. If desired, the resolution of the measurement can be improved by increasing the T_{MEAS} time measurement interval.

In Fig. 2 the force sensing resistor R_{FSR} is placed in parallel with resistor R_{MAX}, giving

$$R_A = R_{FSR} || R_{MAX} = \frac{R_{FSR} R_{MAX}}{R_{FSR} + R_{MAX}} \tag{3}$$

This limits the maximum value of R_A, as $R_{FSR} \to \infty$ for zero force, which would result in a waveform period T exceeding T_{MEAS}. A known reference resistance R_{REF} is used for R_B.

B. Calibration

Combining (1), (2), and (3), solving for force F gives

$$F = \left[R_0 \left(\frac{1}{R_{REF}} \left[\frac{1-\delta}{2\delta - 1} \right] - \frac{1}{R_{MAX}} \right) \right]^{-1/x} \tag{4}$$

in which R_{REF} and R_{MAX} are known, and best-fit parameters R_0 and x are determined from initial measurements.

IV. Results

The design of Figure 2 was tested for accuracy over forces corresponding to a pressure range of 20 to 350 mm Hg. To investigate tolerance of this approach to variation in the FSR characteristic, four different sensors were tested. System parameters and results are summarized in Table I.

The upper plot in Figure 3 shows calculated pressure from (4) as a function of the known applied pressure. Each of the four tested sensors is represented in a different color. The dashed lines indicate calculated pressure using the nominal FSR parameters of (1); the wide sensor-to-sensor variability of parameters is apparent. The solid lines show results after calibration, using a least-squares determination of best-fit values for R_0 and x from (1) for each sensor. The lower plot in Figure 3 shows the measurement error (the difference between the calculated pressure and actual applied pressure) as a fraction of the 350 mm Hg full scale. Despite the wide variation in initial uncalibrated performance, the model of (1) enables accuracy within $\pm 3\%$ for calibrated output.

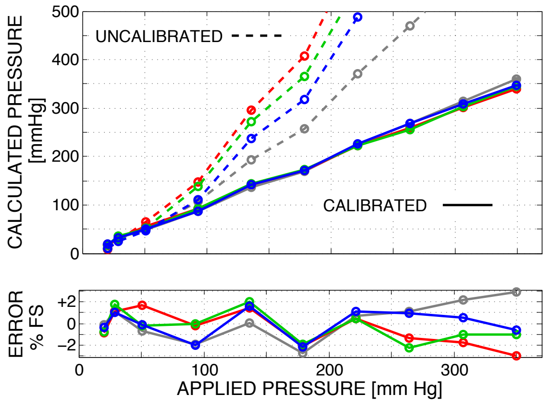


Fig. 3. Measurement results, before (dashed) and after (solid) calibration.

TABLE I
PROTOTYPE SYSTEM PARAMETERS / RESULTS

PARAMETER / RESULT		VALUE	UNITS
Sensor	Active Area	1.27	cm^2
	Force Range	20 - 500	g
LM555	Frequency Range	114Hz 470Hz	kHz
	δ Range	51 - 97	%
Pressure	Resolution	1	mm Hg
Measurement	Sample Rate	1	Hz
System	Accuracy Calibrated	$\pm 3\%$	%
Performance	Power Dissipation	2.5	mW

V. Conclusion

An interfacing technique for resistive pressure sensors has been presented which enables determination of resistance using a time-based measurement approach. Unlike traditional voltage-based approaches, the technique requires no reference voltage or analog-to-digital converter, is more digital-friendly by moving measurement into the time domain, and allows reconfigurable system accuracy by increasing measurement time. Measured results show accuracy of $\pm 3\%$ after calibration. Although presented in the context of a resistive pressure sensor, the technique is applicable to any resistive sensor.

Acknowledgment

This material is based upon work supported by a grant from the UMMS/WPI Collaborative Seed Funding Initiative.

References

[1] J. M. Nagasako et al., "Closed-Loop Electromyography Controller for Augmented Hand Exoskeleton Gripping," NEBEC2015.
[2] N. E. Miller et.al., "Saving Navy SEALS: Pressure-Activated System for Personal Flotation Device," NEBEC2015.
[3] K. Dyer et.al., A New Body Weight Supported Treadmill Device to Measure Kinetic Response from Spinal Cord Injury Animals," NEBEC2015.
[4] Interlink Electronics, "FSR Integration Guide,"

Appendix I

SENSORS Conference Paper

Flexible Sensor for Measurement of Skin Pressure and Temperature in a Clinical Setting

John McNeill, Matthew Crivello, Yitzhak Mendelson, Devdip Sen
Worcester Polytechnic Institute
100 Institute Rd, Worcester, MA 01609 e-mail: mcneill@wpi.edu

Raymond Dunn, M.D., Kelli Hickle, M. D.
University of Massachusetts Medical School
55 N Lake Ave, Worcester, MA 01655

***Abstract*—A flexible, wearable sensor patch for simultaneous monitoring of local skin pressure and temperature is described. Measurement can be collected for a period of time extending over several hours, suitable for monitoring in a clinical setting, for example during surgery, in the home, or in a long-term care facility. Experimental results are presented demonstrating pressure and temperature measurement at multiple locations on an anesthetized animal during a seven hour surgical procedure.**

I. Introduction

Accurate measurement of force and/or pressure is required in many biomedical applications such as medical diagnostics [1], fingertip pressure in an active exoskeleton [2], automated personal safety equipment [3], and rehabilitation [4]. This paper describes a sensor patch and measurement technique applicable to sensing of localized skin pressure, for example during surgery when a patient may be immobilized for an extended period of time. While the patient is immobilized, susceptible points on the body can experience a localized elevation of pressure. If pressure exceeds a threshold of roughly 30 mm Hg, capillary blood flow can be reduced or stopped, denying oxygen to tissue in the area. Over time the reduced blood flow can result in injury or in extreme cases tissue death (necrosis). Since local temperature is also an indicator for tissue damage, measuring temperature as well as pressure is desirable for preventing injury.

The novel contribution described in this paper is a flexible, wearable sensor patch for simultaneous monitoring of local skin pressure and temperature. The measurement techniques presented are suitable for ultimate implementation in a wireless patch system. Measured results are presented showing that information from multiple sensors can be combined and displayed to alert a caregiver to a condition requiring intervention.

This paper is organized as follows: Section II provides background information on the application and an overview of the sensor patch design. Section III provides details on the system design, including the force and temperature sensors used. Measured results are provided in section IV.

II. Application

Figure 1 shows a common condition that can lead to tissue damage in at-risk areas such as the heel, sacrum, scapula, ischium (sitting patient), or occiput (comatose patient). When a sufficient fraction of the patient's weight is supported in a region with a bony prominence, the resulting localized concentration of pressure reduces the cross-sectional area of blood vessels, restricting blood flow and limiting oxygen supply to the at-risk tissue. If pressure is maintained for sufficient time, the lack of oxygen leads to tissue necrosis.

To prevent tissue injury, the aim of this work is to develop a sensor system that would alert a caregiver to a potentially harmful level of pressure. The alert would enable immediate direct intervention to prevent a problem in the at-risk body area. The ultimate goal is to develop a disposable sensor suitable for any at-risk body area, as part of a communication and monitoring system that will measure local pressure and temperature at points on the patient's body known to be vulnerable to damage, communicate results to a monitoring station, and alert a caregiver if intervention is necessary.

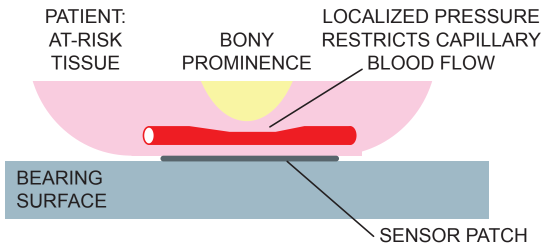


Fig. 1. Application of patch (cross-sectional view) in measurement of local pressure and temperature for at-risk tissue.

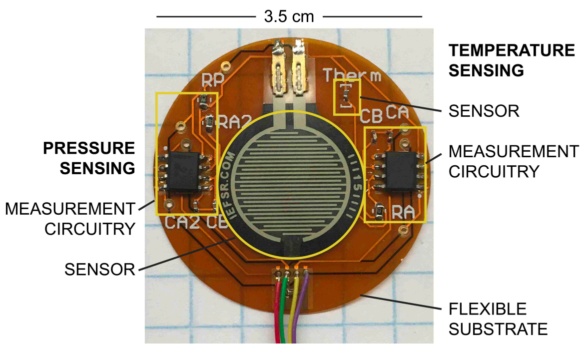
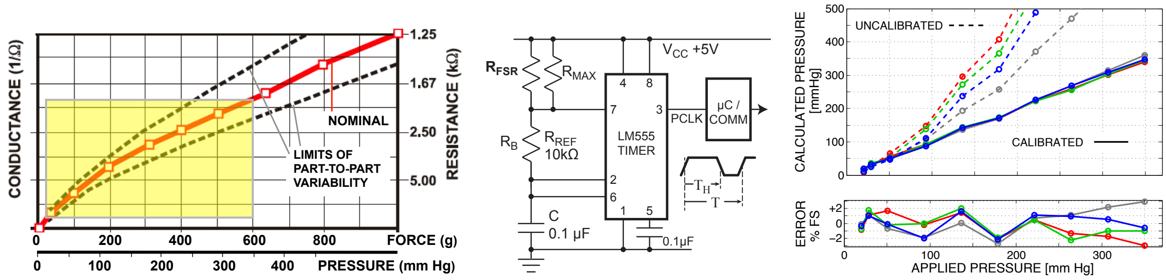


Fig. 2. Wired prototype of sensor patch.



(a) FSR-402 conductance, resistance vs. force [5] (b) R_{FSR} measurement circuitry. (c) Calibrated performance

Fig. 3. Measurement technique for resistive force sensor.

III. System Design

Ultimately the sensor system will incorporate wireless communication; the work presented in this paper uses a wired prototype shown in Figure 2 to verify the physical design of the temperature and pressure sensor measurement techniques required for the eventual wireless implementation.

A. Pressure Measurement

Pressure is measured using a Polymer Thick Film (PTF) Force Sensitive Resistor (FSR) and inferring pressure using the FSR active area. The basis of PTF sensing is the increased conductance of a polymer layer subjected to a compressive force. Figure 3a shows the force-conductance characteristic for the FSR-40 sensor [5], with the region of forces between 30g - 600g highlighted. Given the sensor active area of $A = 1.27\text{cm}^2$, this region corresponds to a range of pressures from 20 - 350 mm Hg, as expected in applications such as [2]–[4]. Over this range, the resistance-to-force F (in grams) relationship shows a power-law characteristic [5] as

$$R_{FSR} = R_0 F^x \tag{1}$$

in which R_0 and x are calibration parameters.

In [6], the authors presented a time-based measurement procedure (shown in Figure 3b) based on the LM555 timer. For the output digital waveform PCLK, the duty cycle $\delta = T_H/T$ (fractional "high" time T_H relative to the waveform period T) is determined solely by resistance ratios. The microcontroller can use the known values of R_{REF} and R_{MAX} to calculate R_{FSR} from the measured duty cycle δ; in turn the measured value of resistance R_{FSR} can be used to determine force on the sensor using the model of (1).

A challenge associated with this type of pressure sensor is part-to-part variability of up to $\pm 25\%$, indicated by the dashed lines of Fig. 3a [5]. In [6], the authors also presented a calibration procedure; results for four different sensors are shown in Figure 3c. The procedure enables linear force measurement with accuracy approaching the FSR's single-part repeatability of $\pm 2\%$.

B. Temperature Measurement

The same time-based resistance measurement technique was used for temperature sensing, using a resistive temperature sensor [7]. For this negative temperature coefficient (NTC) sensor, the resistance-vs.-temperature (R-T) characteristic is given by

$$R_{RTS} = R_{T=T_0} e^{B(1/T - 1/T_0)} \tag{2}$$

in which $R_{T=T_0}$ is the resistance at reference temperature T_0 and B is a calibration constant. In similar fashion the duty cycle was measured by the microcontroller; from the inferred resistance R_{RTS} the R-T characteristic in (2) was used to calculate temperature.

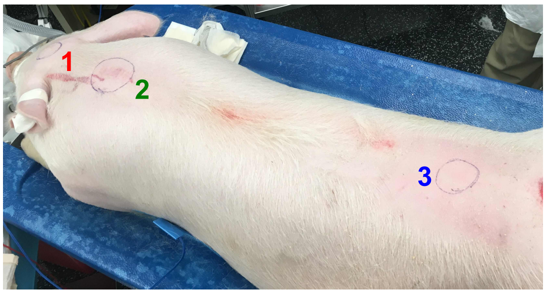


Fig. 4. Sites instrumented on anesthetized pig.

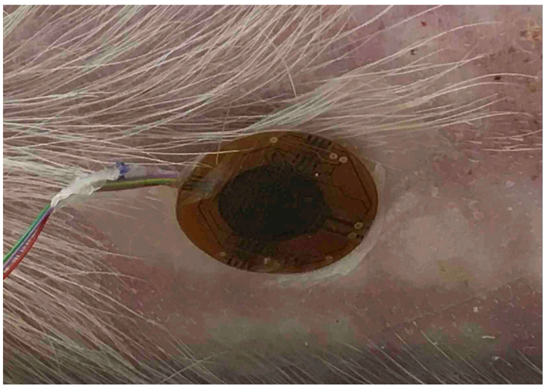


Fig. 5. Closeup of site after sensor applied.

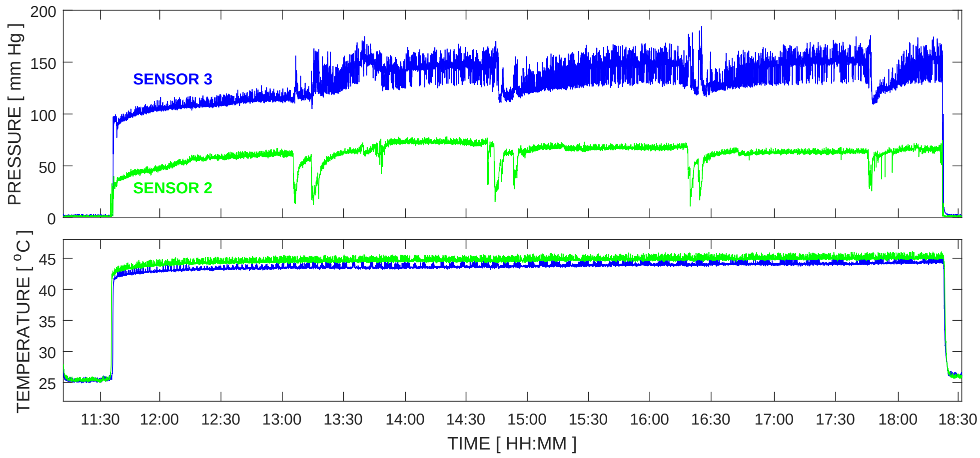


Fig. 6. Pressure and temperature data over duration of surgery, including pre-and post attachment control times.

IV. Results

An anesthetized pig of mass $\approx$ 80kg was instrumented with three different sensors using sites as shown in Figure 4 (prior to surgery). The sites were chosen to be in proximity to a bony prominence as indicated in Figure 1. Figure 5 shows a closeup of the site after sensor application. During the surgical procedure the pig was on its back; given its weight the resulting load on the bearing surface is similar to what would experienced by a human subject.

Figure 6 shows measured plots of pressure and temperature data over the duration of surgery, a time interval of nearly seven hours. During this time, operation of sensor 1 became intermittent due to a mechanical failure, so its data is not presented in the figure. Samples were taken at intervals of approximately two seconds. No filtering or other signal processing was applied to the data. The sensors were attached between times 11:35 and 11:37; the sensors were removed at the conclusion of the surgical procedure at 18:22. The plots also includes measured data from pre-and post attachment times, which serve as a control showing pressure roughly equal to zero and temperature roughly equal to the ambient temperature of 25^oC.

Pressure data shows a range of pressure from 30-80 mm Hg for sensor 2, and 100-180 mm Hg for sensor 3. The surgical protocol required that the animal be moved somewhat at intervals of roughly 90 minutes. This is shown in the brief relief of pressure in sensor 2 at times 13:06 to 13:14, 14:39 to 14:55, 16:18 to 16:26, and 17:46 to 17:50. Note that the relief in pressure is much less pronounced in the results for sensor 3.

The bottom plot shows nearly constant temperature data in the range of $42 - 43^o$C. While local skin temperature can indicate blood flow, we were unable to observe any variation in the case of this experiment due to the surgical protocol which required that the animal's temperature be stabilized using a thermal blanket. We were able to verify the accuracy of the temperature measurement function, as the thermal blanket was set to maintain a temperature of 42^oC.

V. Conclusion

A flexible, wearable sensor patch for simultaneous monitoring of local skin pressure and temperature has been presented. The measurement techniques used are suitable for ultimate implementation in a wireless patch system. Results are presented showing that pressure and temperature measurements from multiple sensors can be combined and displayed, providing information that can alert a caregiver to a condition requiring intervention.

Acknowledgment

This material is based upon work supported by a grant from the UMMS/WPI Collaborative Seed Funding Initiative.

The authors thank William Appleyard of WPI for assistance with sensor fabrication, and Heather Tessier of UMMS for coordination of access to experimental resources and assistance in compliance with the UMMS Institutional Animal Care and Use Committee (IACUC)-approved protocol.

References

[1] X. Mi and F. Nakazawa. "A Multipoint Thin Film Polymer Pressure/Force Sensor to Visualize Traditional Medicine Palpations," *IEEE SENSORS 2014*.

[2] J. M. Nagasako et al., "Closed-Loop Electromyography Controller for Augmented Hand Exoskeleton Gripping," *IEEE Northeast Biomedical Engineering Conference (NEBEC 2016)*.

[3] N. E. Miller et.al., "Saving Navy SEALS: Pressure-Activated System for Personal Flotation Device," *NEBEC2015*.

[4] K. Dyer et.al., "A New Body Weight Supported Treadmill Device to Measure Kinetic Response from Spinal Cord Injury Animals," *NEBEC2015*.

[5] Interlink Electronics, "FSR Integration Guide."

[6] M. Crivello, D. Sen, J. McNeill, Y. Mendelson, R. Dunn, M.D., K. Hickle, M. D., "Modeling of Force Sensor Nonlinearity for Time-Domain-Based Pressure Measurement in Biomedical Sensors," *NEBEC2016*, Binghamton, NY, April 2016.

[7] muRata Electronics, "NTC Thermistors, NCP15XH103"

References

[1] C. H. Lyder and E. A. Ayello, "Pressure Ulcers: A Patient Safety Issue," 2008/04 2008.

[2] C. A. Russo, "Hospitalizations Related to Pressure Sores," Agency for Healthcare Research and Quality, Healthcare Cost and Utilization Project2006.

[3] L. EM, "Micro-injection studies of capillary blood pressure in human skin," ed, 1930.

[4] D. K. Langemo. (2005) Quality of Life and Pressure Ulcers: What is the Impact? WOUNDS. Available: http://www.woundsresearch.com/article/3625

[5] jmorgan. (2013). 555 Timer Oscillator Circuit. Available: http://ehelion.net/projects/digitalclock/555timer.html

[6] J. McNeill, "ECE3204 Lecture 15," ed: Worcester Polytechnic Institute, 2015.

[7] J. Wright. (2016). Dispelling Common Bluetooth Misconceptions. Available: http://www.sans.edu/research/security-laboratory/article/bluetooth

[8] Bluetooth, "Bluetooth Low Energy — Bluetooth Technology Website," 2016.

[9] S. I. Inc., "SEIZAIKEN(Mercury-Free)," 2016.

[10] (2015). EnerChip Rechargeable Solid State Batteries — Thin Life-of-Product Energy Storage — Cymbet Corporation. Available: http://www.cymbet.com/products/enerchip-solid-state-batteries.php

[11] "What is RFID Technology — How RFID Works — RFID Applications," 2015-08-16 2015.

[12] T. A. Krouskop, "A synthesis of the factors that contribute to pressure sore formation," Medical hypotheses, vol. 11, pp. 255-267, 1983.

[13] PTC. (2016). ThingWorx — An Enterprise IoT Solutions Company. Available: https://www.thingworx.com/

[14] (2016). Wireless Patient Monitoring ... Leaf Healthcare. Available: http://www.leafhealthcare.com/

[15] wellsense. (2016). M.A.P. Available: http://themapsystem.com/

[16] M. Levin-Epstein, "Boise VA Hospital Finds Leaf Healthcare Wearable Sensor Improves Compliance With Pressure Ulcer Prevention Efforts," Journal of Clinical Engineering, vol. 40, pp. E1-E2, 2015.

[17] A. Siddiqu, R. Behrendt, M. Lafluer, and S. Craft, "A Continuous Bedside Pressure Mapping System for Prevention of Pressure Ulcer Development in the Medical ICU: A Retrospective Analysis," Wounds, A Compendium of Clinical Research and Practice, vol. 25, 2013.

[18] D. B. Drennan and D. W. Southard, "System and method of reducing risk and/or severity of pressure ulcers," ed: Google Patents, 2013.

[19] G. M. Ortega and M. J. Sciarra, "Active on-patient sensor, method and system," ed: Google Patents, 2012.

[20] (2016). TLC556CDR Texas Instruments — Integrated Circuits (ICs) — DigiKey. Available: https://www.digikey.com/product-detail/en/texas-instruments/TLC556CDR/296-1338-1-ND/276606

[21] I. Electronics, "FSR 402 Datasheet," ed.

[22] I. Electronics, "FSR Force Sensing Resistor Integration Guide and Evaluation Parts Catalog," 90-45632 Rev. D ed.

[23] Arduino. (2016). Arduino - ArduinoBoardUno. Available: https://www.arduino.cc/en/Main/ArduinoBoardUno

[24] Arduino. (2016). Arduino - PulseIn. Available: https://www.arduino.cc/en/Reference/PulseIn

[25] G. Lavenuta, "An Explanation of The Beta and SteinHart-Hart Equations for Representing The Resistance vs. Temperature Relationship in NTC Thermistor Materials," ed: Qti Sensing Solutions.

[26] (2016). NA555D Texas Instruments — Integrated Circuits (ICs) — DigiKey. Available: https://www.digikey.com/product-detail/en/texas-instruments/NA555D/NA555D-ND/1571931

[27] THOMAS A. KROUSKOP, The Institute for Rehabilitation and Research, 1333 Moursund Avenue, Houston, Texas 77030, U.S.A.

[28] (2016). The ZigBee Alliance. Available: http://www.zigbee.org/

[29] (2016). Comparing Low-Power Wireless Technologies — DigiKey. Available: http://www.digikey.com/en/articles/techzone/2011/aug/comparing-low-power-wireless-technologies

[30] V. Health. (2016). Alternating Pressure Mattress. Available: http://www.vivehealth.com/products/vive-alternating-pressure-mattress

[31] I. PCB Universe. (2016). PCB Universe - Printed Circuit Boards - Custom PCB Prototypes and Production Printed Circuit Boards - Quote On-line with our Instant PCB Quote. Available: http://www.pcbuniverse.com/