Smart Brace

Taylor Ellsworth
ECE 499: Capstone Design Project
Advisor: Takashi Buma
March 17, 2016

Report Summary

The main goal of the Smart Brace is to design and build a knee brace that an ACL patient could use to track and monitor their rehab progress. Using a potentiometer to measure range of motion, the Smart Brace will send the data over Bluetooth to a user's computer, and display the results for the user to see. In order to be considered helpful to the user, the Smart Brace has to meet many design requirements. The system has to be light and small enough to not affect the user's movement during exercise. In addition, a user interface has to be created that is easy for the patient to use. They have to be able to see the max values, as well as a graph showing the results. Finally, the brace has to contain enough memory and battery life to last throughout an entire rehab workout. After taking all of these things into consideration and performing necessary calculations, all the components were finalized and purchased. During the winter term, a functioning prototype was created that ACL patients could use. After transmitting the data over to a computer, the patients can see their max range of motion value, as well as a graph showing the data.

Table of Contents

Table of Figures and Tables

1. Introduction

Knee injuries are prevalent in today's society. For example, over 200,000 ACL tears occur each year (1). In order to return to daily and recreational activities, ACL patients first have to go through an exercise rehabilitation program. These programs include stretches and exercises for the patient to complete every single day. Each week, the program outlines goals that the patient should try to reach. However, a huge problem with these rehab programs is that they are not geared towards the specific individual recovering from injury. Everyone recovers at a different pace. Some people make faster progress and are ahead in schedule in their rehabilitation process. On the other hand, others take a longer time to reach the weekly goals. This raises the question: Is there a way to track how a person is progressing during their rehabilitation program, so they can alter their workout plan accordingly?

To fix this problem, I have come up with the Smart Brace. This wearable knee brace will have sensors to measure the range of motion of the user's knee during rehab exercises. According to Emory Healthcare, regaining full range of motion is very important and is one of the main goals in an ACL rehabilitation program (2). Through Bluetooth, the Smart Brace will transmit data to a computer for storage and analysis. By analyzing this information over a period of several days, the user can determine if they are progressing at a good pace, or if they are not progressing at all. Subsequent changes to their workout plan can be made accordingly.

The remainder of this report is organized into four sections. Section two reviews ACL injuries and discusses previous smart knee braces. In addition, it also discusses the ethical issues for the project. Section three describes the requirements that the Smart Brace must meet in order for it to be a helpful product to the user. Section four talks about the design alternatives for all the hardware and software systems. Furthermore, section five explains the proposed final design for the Smart Brace, and section 6 provides the final design and implementation. The next section examines the cost analysis for the product, while section 8 gives the performance estimates and results. Section 9 outlines the production schedule for the winter term. Section 10 offers a user manual for the operation and maintenance of the system. Finally, section 11gives the discussion and conclusion, and section 12 acknowledges the people who advised me while creating this product.

2. Background

2.1 What is the ACL?

ACL is an acronym for the anterior cruciate ligament. It is one of the four main ligaments in the knee that connects the tibia to the femur. It is located diagonally in the middle of the knee. The purpose of this ligament is to keep the knee stable. It provides rotational stability to the knee, as well as prevents it from hyperextending.

The ACL can be torn in many ways. The main causes of injury are when the knee is twisted, bent side to side, or is hyperextended. These things can occur when changing direction rapidly, landing incorrectly from a jump, or when there is a collision directly to the knee.

ACL injuries are prevalent in sports that require a lot of stop and go movements. For example, this injury is prevalent in sports like soccer, football, basketball, and tennis (3).

2.2 Treatment of Injury

Without treatment, the injured ACL is less able to control knee movements. As a result, there exists two ways to treat an ACL tear. If the patient is older and not active, they may choose the nonsurgical treatment for their injury. However, the ACL will not completely heal this way. Thus, the doctor will recommend that the patient brace their injured knee. Bracing it will defend the knee from instability.

Choosing this type of treatment allows the patient to return to a much quieter lifestyle.

On the other hand, majority of patients choose the surgical treatment for their injury. This treatment will ultimately allow them to return to their normal way of life. The surgery consists of the doctor replacing the torn ligament with a tissue graft. This tissue is often taken from either the patella tendon in the knee, or the hamstring tendon. Surgery to the ACL is performed using an arthroscope that is inserted into the joint using small incisions (4).

2.3 Recovery Process

After surgery, the doctor assigns every patient a rehab program to follow. Emory Healthcare states that the goal in every post operation ACL rehabilitation program is "to return the patient to a normal and complete level of function in as short a time possible, without compromising the integrity of the surgically reconstructed knee" (3). In addition, the patient should strive to retain full range of motion, reduce swelling, and retain muscle size and strength. To achieve these goals, a normal rehab program consists of stretches and exercises for the patient to complete. Some examples of exercises include leg extensions, flexions, and lifts. Ideally, by successfully completing the program, a patient can return to full activity in six to nine months (2).

2.4 Previous Work

Previous attempts have been made to track a person's rehab process from knee injuries. Researchers from the Fraunhofer Institute for Manufacturing Engineering and Automation IPA in Stuttgart, Germany have come up with a knee brace that uses sensors to record knee movement. Even though the brace is not commercially available, it is able to archive the degree to which the knee rotates (5).

Figure 1: Knee brace developed by researchers from Fraunhofer Institute

Bend Labs is another company that has attempted to track a person's rehab progress. Like the Smart Brace, Bend Labs' Smart Knee uses sensors to track knee movements and uses Bluetooth to send the data to a smartphone. The data includes the knee angle, hip angle, and altitude of the knee. In addition, the knee brace can track complex human movements in real time (6).

Figure 2: Smart Knee developed by Bend Labs

Both these devices, however, do not use the knee brace that the doctor prescribes after ACL reconstruction surgery. Instead, they use a soft brace that does not provide much support and protection. A person who has torn their ACL is 15 times more likely to re-tear their ACL within 12 months after their first surgery. Wearing the brace helps protect the recently healed knee from outside forces that occur during activity and everyday life (7). This assigned brace is used during rehab exercises. Thus, the Smart Brace is advantageous because it allows patients to track their rehab progress safely while wearing the same protective brace they are given after surgery.

2.5 Effects on Society

2.5.1 Social

Sports and exercise are two huge social aspects of today's society. People use these activities as ways to meet new friends, hang out with peers, and as a source of entertainment. However, when a person tears their ACL, they can no longer participate in these two events. Until they fully rehab and recover from their injury, they will not be able to be as social as they previously were. By using the Smart Brace, the patient can track their progress while rehabbing. They will be able to increase their rehab workload, which will possibly allow them to be able to return the social activities of sports and exercise in a shorter period of time.

2.5.2 Health and Safety

The Smart Brace will have a positive impact on the health and safety of individuals. Exercise is extremely important. According to Lets Move, an exercise program geared towards raising a healthier generation, "Physical activity is an essential component of a healthy lifestyle. In combination with healthy eating, it can help prevent a range of chronic diseases, including heart disease, cancer, and stroke" (8). ACL injuries prevent people from exercising and living healthy lifestyles. By using the Smart Brace to rehab, patients will have a greater chance to recover from the injury sooner, allowing them to return to physical activity.

In addition, if a person comes back from an ACL injury too soon, they are at a higher risk of injuring it again (9). By being able to track the range of motion of their knee, a patient will know how far along they are in their recovery. This will prevent them from jeopardizing their safety by returning too early.

2.5.3 Political

Over the next ten years, the cost of health care is expected to rise 5.8% each year (10). Consequently, the cost of going to rehab is going to increase as well. As stated earlier, rehab programs are often not geared towards specific individuals. Patients are often placed on preset rehabilitation programs. Consequently, they attend a preset number of rehab sessions, and progress at a predetermined rate. The Smart Brace will allow a patient to possibly recover at a faster, more individually based rate. Thus, the patient will possibly be able to attend less rehab sessions, saving a lot of money.

2.6 Ethics

Ethical issues naturally surround any product that claims to aid a patient's recovery. For example, it would be unethical to try to market and sell the Smart Brace without it first being cleared by the FDA. In order for this device to be put on the market, it first has to get cleared by a 510(k). A 510(k) requires that the new device be compared to similar lawfully marketed devices. It must be compared for safety and effectiveness (11). As the designer and developer, I would make sure that the Smart Brace is cleared by a 510(k) before selling it to any consumer.

In addition, it would be unethical for the Smart Brace to display inaccurate results. It cannot exaggerate a patient's recovery pace, with the hope that consumers will be satisfied and recommend the product to more people. As the designer, I have a responsibility to the clients and their health. Therefore, I am ethically obligated to ensure that the Smart Brace yields accurate results. To ensure accurate results, I would do several things. To confirm that the angle of rotation is accurate, I will measure the angle of a patient's knee using a potentiometer. I will then measure the angle of the patient's knee using a protractor. Testing will be performed with the knee at many different angles. If the angles measured using the potentiometer and the protractor are the same, the data obtained for angle of rotation is accurate.

Furthermore, another ethical issue surrounding the Smart Brace involves the pricing of the product. In hopes of making a sizable profit off of this device, it would be unethical to raise the price to a level that is significantly higher than market value of a comparable knee brace. The knee brace that is commonly prescribed to patients is the Breg Fusion Knee Brace. According to The Brace Shop, the retail price is $530 (12). A high-end alternative to the Breg Fusion Knee Brace is the Breg Custom Fusion XT with AirTech knee brace (13). Valued at $799, some people purchase this brace because the extra padding provides more comfort to the user. To ensure that the Smart Brace is priced fair and reasonably, I will make sure that I price the brace appropriately according to these range of prices.

3. Design Requirements

The Smart Brace must meet certain requirements in order to be a useful product to the user. The development of a wearable smart brace will allow people to be able to measure the range of motion of their knee. Through Bluetooth, the Smart Brace will be connected to a computer. The microcontroller will send the taken data to the user's computer, and these values will be displayed to the user and stored for analysis. A block diagram of the overall project can be seen below.

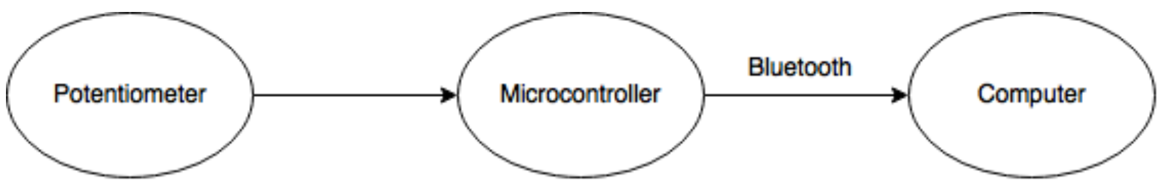


Figure 3: Model of overall project

Since the system is being attached to an existing knee brace, there must be constraints on dimensions. The potentiometer, Bluetooth module, battery, and microcontroller all have to be small enough to fit on the knee brace, and not interfere with the user's movements. After measuring the dimensions of the knee brace, the components should be no more than 50mm x 30mm (1.97" x 1.18"). In addition, there must be restrictions on the weights of the equipment. The weights must be negligible, so the user cannot feel the added weight to the knee brace. A good value to strive for is for each component to weigh less than 5 grams.

Battery life, number of samples per second, and memory are three more requirements that need to be quantified. A daily rehab session takes about an hour to complete. In this hour, a user typically completes 15 exercises. Each one of these exercises consists of the patient bending and straightening their knee for 3-5 sets of 10-15 repetitions. Each exercise takes approximately one to two minutes to complete. In between sets, a person usually rests for a couple of minutes. Since the total rehab exercises take roughly an hour, the battery power source must provide enough current to support at least one hour of continuous use. In addition, the number of samples taken per second need to be quantified. The angle of the knee joint and angular velocity of the user's knee will constantly be changing. As a result, the number of samples taken per second has to be great enough to ensure that no values are missed. After much thought, I decided 300 samples per second would ensure this. This is because the maximum knee angle velocity in a normal gait is roughly 300 degrees per second (14). Furthermore, the memory of the microcontroller has to be quantified as well. The microcontroller has to have enough memory to capture and store the data before it sends it to the computer. 2 KB of SRAM memory would be sufficient. This is because the 50 bytes of data will be stored in a buffer at a time. The 2 KB of SRAM memory is more than enough for the 50 bytes at a time.

Furthermore, the desired values for range of motion is a design requirement as well. The range of motion of a knee is up to 135 degrees (15). Therefore, the desired values should be between 0 and 135 degrees. When designing the knee, I will use the following image as a model.

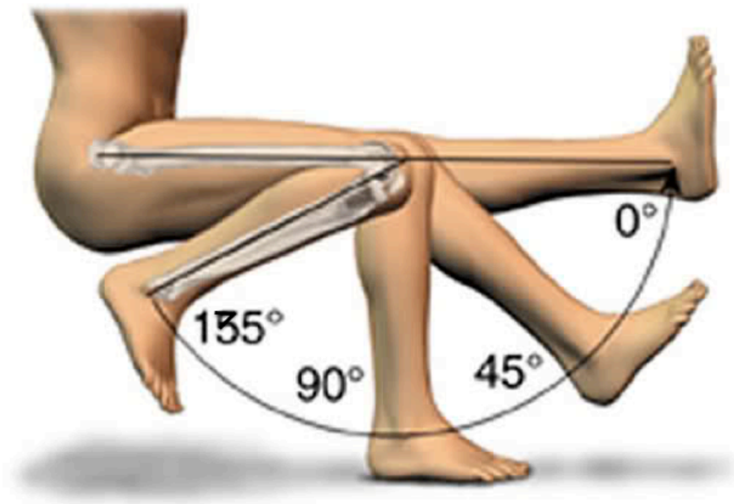


Figure 4: Range of Motion model when designing the Smart Brace

I will consider zero degrees of rotation to be when the knee is completely straight, and 135 degrees of rotation when the knee is fully bent.

Also, the range accuracy should be within 1 degree. Since a patient will be tracking their results on a day-to-day basis, any slight increase or decrease in their range of motion needs to accurately be recorded.

Another design requirement is the user interface. On the computer, the user needs to see the current max values for angle of rotation of their knee. Also, it would be extremely helpful for them to see all the values throughout every exercise. Furthermore, they need a way to see a graph of the data, so they can see a visual representation of the movement of their knee.

The Smart Brace also must be affordable. Even if the product is extremely functional and useful, it must be cheap enough for a consumer to buy it. As stated earlier, the knee brace that is commonly prescribed to patients is the Breg Fusion Knee Brace. The retail price of it is $530. As a result, the Smart Brace must be priced at an amount where the cost of buying it does not outweigh the usefulness of it.

Table 1 below summarizes all the design requirements.

Objective	Requirement
Size of Components	< 50mm x 30mm (1.97" x 1.18")
Mass	<= 5 grams per component
Battery Life	> 1 Hour of use
Samples Per Second	~300 samples per second
Memory	>= 2 KB
Range of Motion	0-135 degrees
Range of Motion Accuracy	Within 1 degree
User Interface	Display max values Graph of results
Price	Not a lot greater than $530

Table 1: Table summarizing design requirements

4. Design Alternatives

4.1 Microcontroller Alternatives

When choosing what microcontroller to use, there were several options. After much consideration, I decided to settle on the Arduino family of microcontrollers. This is because Arduino prides themself on being an open source platform used for building electronic projects. Unlike many other microcontrollers, the Arduino does not need a separate piece of hardware to load code onto the board. A USB cable is all it needs. Furthermore, the piece of software that the microcontrollers use is called the Integrated Development Environment. The Integrated Development Environment uses a basic version on C++, making it easier to program (16).

The first microcontroller in the Arduino family that we considered using is called the Arduino Uno. The Uno consists of 14 Digital I/O pins, and has 2 KB of SRAM memory. Also, it can be powered by a 9V external power supply, making it portable so it could be attached to the knee brace. However, one of the main design requirements was that the microcontroller had to be small enough to fit on the knee brace. With a length of 68.6mm and a width of 53.4mm, the Arduino Uno is too large to be placed on the knee brace, without interfering with the user's movements. In addition, another major design requirement was the user could not feel the weight of the microcontroller added to the knee brace. The Uno has a weight of 25 grams. Ideally, we were looking for microcontroller weighing much less than that (17).

The next microcontroller that we considered using was the LilyPad Arduino USB. According to the Arduino website, "The LilyPad Arduino USB is the perfect board for e-textiles and wearable projects" (18). With only a 50mm diameter, and a weight of 0.8 grams, the LilyPad Arduino USB would pass the size requirements for the microcontroller. However, similar to the Arduino Uno, the LilyPad presented one big problem. It is powered with a 3.7V battery (18). This battery would not have enough power to support the devices for an hour of continuous use.

On the other hand, the Arduino Nano offers the same specifications as the Arduino Uno, but is in size of the LilyPad Arduino. With dimensions of 45 mm x 18 mm and a mass of 5 grams, it meets the size design requirements. The Arduino Nano is able to fit on the Smart Brace without interfering with the user's movements. In addition, it can be powered with a 9V battery. This battery will be able to support the Smart Brace for at least one hour. As a result, the decision was made to not use the Arduino Nano.

4.2 Sensor Alternatives

In addition to the microcontroller, there were also design alternatives for the sensor. In the beginning of the design process, a rotary encoder was thought to be the best way to measure range of motion. However, one of the design requirements for the Smart Brace is that the price cannot be greater than the $799 Breg Custom Fusion AirTech Knee Brace. In order to keep the final cost in an acceptable range, the components cannot cost a lot of money. However, the cost of a rotary encoder tends to be moderate to high compared to other devices. For example, the Omron

Rotary Encoder costs $262.20. This would definitely cause the Smart Brace to be way too expensive (19).

Another way to measure range of motion is by using a potentiometer. Precision potentiometers are designed for control applications where accuracy is important. These types of potentiometers are cheap. For example, the ETI Systems Precision Potentiometers are sold for $17.35 (20). Consequently, it was decided to use a cheaper potentiometer rather than a very expensive rotary encoder to measure range of motion,

4.3 Software Alternatives

Originally, the decision was made to use MATLAB to read in data from the potentiometer on the computer. This was due to the fact that I have used MATLAB in a majority of my electrical engineering courses here at Union. However, I wanted to challenge myself by using a language that I was not as familiar with. Python is a programming language that I had only used once in my college career. I have not had much experience using it. Therefore, it would require more work to become as familiar with Python as I am with MATLAB. As a result, I decided to use Python.

4.4 Communication Alternatives

When selecting a way to transmit the data to the computer, I decided on using Wi-Fi. At first, it seemed like the perfect method. However, it has a range of 32 meters indoors and 95 meters outdoors. For the application of the Smart Brace, this long range is unnecessary and impractical. When performing rehab exercises, the computer will not be a more than a couple meters away from the user. Other reasons why Wi-Fi wasn't the best method is because it has high power consumption and is not easy to use. It is extremely complex and requires both configuration of the hardware and software. On the other hand, Bluetooth has a range of zero to thirty meters. This is much more appropriate. Also, Bluetooth has a much lower power consumption than W-FI (21). Thus, it was decided that Bluetooth was the best way to transmit the data to the computer.

After deciding on Bluetooth, I then had to decide whether to send one byte at a time from the Arduino to the computer, or to use buffers to send several bytes of data at a time instead. Latency is the amount of time it takes a packet to travel from source to destination (22). The latency of Bluetooth is 200ms (21). If the data were to be sent one byte at a time, the Arduino Nano would have to read in one value from the potentiometer, and send this one byte over to the computer. During the 200ms of downtime of waiting for the byte to arrive to the computer, the Arduino reads in another byte of data. When the computer receives the first byte of data, it converts it to degrees. Once this byte of data is converted to degrees, it repeats this process with the next byte of data. This is highly inefficient. By using a buffer technique, the Arduino will read in several values from the potentiometer, and store them in a buffer. Instead of transmitting only one byte of data to the computer at a time, the Arduino will now transmit several bytes at a time. The computer will receive the buffer, and convert the values to degrees. The buffer method allows the Arduino to read in another set of values into another buffer during the 200ms of downtime that exists while the previous buffer is being sent to the computer. This is much more efficient than the Arduino only reading one byte of data during the latency period. This way, when the computer is done converting the data in one buffer to degrees, they can immediately start the same process with the next buffer that was already sent by the Arduino. As a result, it was decided to use the buffer technique to transmit the data.

5. Proposed Design During Fall Term

The goal of this project is to make a prototype where a microcontroller correctly obtains data from a potentiometer, and sends it to a computer. As the data comes in, the computer will update the angle of the knee joint, and display it to the user.

A diagram of the system is shown below in Figure 4. In this section, each component of the system will be discussed, as well as a timeline schedule for the Winter Term.

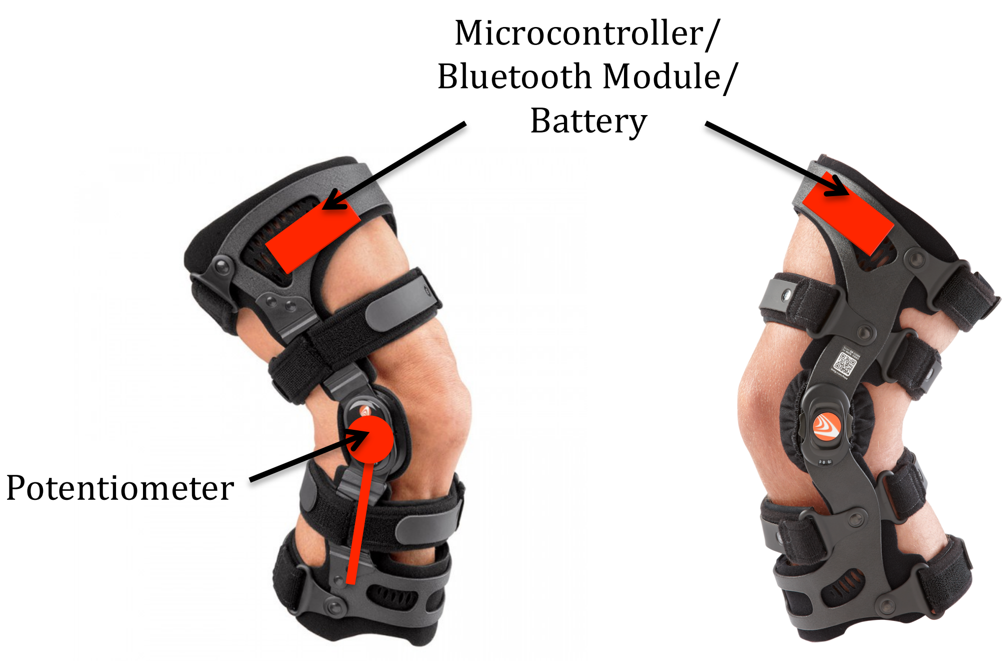


Figure 5: Diagram locating location of the parts

5.1 Microcontroller

Ultimately, we decided to use the Arduino Nano 3.0 (23). The specifications are given below in Table 1.

Input Voltage	7-12 Volts
Operating Voltage	5 Volts
Digital I/O Pins	14
DC Current per I/O Pin	40 mA
SRAM	2 KB
Dimensions	45 mm x 18 mm
Weight	5 grams

Table 2: Specifications of the Arduino Nano used in the Smart Brace

The Arduino Nano was chosen because there is more than enough I/O pins to support the other devices in this product. Also, unlike the Arduino Uno, the Arduino Nano is small enough to fit on the brace without impacting the user's performance. With dimensions 23.6 mm x 35.4 mm smaller than that of the Arduino Uno, the microcontroller is small enough for the Smart Brace. Furthermore, the Arduino Nano weighs a very light 5 grams, 20 grams less than the Arduino Uno. Even though the user may still feel this added weight on the knee brace, it is nowhere near extra weight that the Uno provided. Finally, this microcontroller was chosen because the input voltage is between 7-12 Volts. Unlike the 3.7 Volts used to power the LilyPad Arduino USB, a battery of 9 Volts could provide enough current to power the Smart Brace for more than an hour. This calculation will be shown in section 5.5.

5.2 Potentiometer

The potentiometer being used is a 1 W 5KOhms 5% precision potentiometer manufactured by ETI Systems (20). Table 2 gives the specifications of the potentiometer.

Resistance	5K Ohms
Linearity	+/- 0.25%
Tolerance	5%
Number of Turns	3
Weight	0.8 oz
Size	1.57" x 0.88"

Table 3: Specifications of the Potentiometer used in the Smart Brace

As you can see, the potentiometer has a resistance of 5K Ohms. Using the 5V supplied from the Arduino Nano, the power dissipation is calculated below in Figure 5.

$$P = I^2 R$$

$$R_{pot} = 5000\ \Omega$$

$$I = \frac{V}{R}$$

$$I = \frac{5V}{5000\Omega} = 0.001\ A = 1\ \text{mA}$$

$$P = (0.001A)^2(5000\Omega) = 0.005\ W = 5\ \text{m}W$$

Figure 5: Calculations for Power Dissipation across the Potentiometer

As the calculations show, a resistance of 5K Ohms is not too low where the potentiometer will get warm and blow out. This is due to the fact that not a lot of Watts are dissipated. Also, the resistance is not too high where it would load down the signal, causing the voltage to drop. Thus, a potentiometer with a resistance of 5K Ohms is sufficient to use.

Another reason why this potentiometer was chosen was because it is called a precision potentiometer. This is due to the fact that its linearity was +/- 0.25%. Linearity is the relationship between the output voltage and rotation of the shaft. In this case, the output voltage will be within 0.25% of the correct amount, based on the rotation of the shaft. Since the angle of rotation of the user's knee needs to be as accurate as possible, this potentiometer ensures that the readings will be very accurate.

Ideally, the number of turns needed for the potentiometer is one. Since the range of motion of a knee is up to 135 degrees, the potentiometer does not need to rotate more than one turn. However, there is no potentiometer that had one turn, with a resistance of 5K Ohms, and a linearity of +/- 0.25%. Both the resistance and linearity are more important in obtaining accurate data, so I chose the number of turns to be the next smallest number. This value happened to be 3.

Finally, the weight and size of the potentiometer fit under our restrictions, so I decided to use it.

5.3 Bluetooth Modem

After contemplating between using Wi-Fi or Bluetooth, Bluetooth was ultimately chosen. This is because Bluetooth has shorter range, low power consumption, and is much easier to use than Wi-Fi. The Bluetooth modem that was chosen is called the SparkFun Bluetooth Modem - BlueSMiRF Gold (24). I chose this modem, because all the specifications support the application of the Smart Brace. Also, it is known for being easy to use with Arduino microcontrollers. Table 4 includes the specifications for this device.

Supply Voltage	3.0V – 3.3V
Serial Communication	2400 – 115200 bps
Frequency	2.402 ~ 2.480 GHz
Power Consumption: Standby/Idle Connected (Normal Mode) Connected (Low Power)	 25 mA 30 mA 8 mA
Size	42mm x 16.5 mm x 5.6mm

Table 4: Specifications for the Bluetooth Modem used in the Smart Brace

5.4 Battery and Battery Life

The battery I decided on is an Energizer LA522 9V Lithium Battery (25). By using the currents used by the Arduino Nano, potentiometer and Bluetooth modem, as well as the data sheet for the battery, it will take roughly 3.75 hours to discharge the battery from 9V to 5.4V. This calculation is shown below in Figure 6. On top of this, the battery will be stored in a battery holder case (26). The case includes an on/off switch for the battery. A user can extend the battery life if they turn the battery off in between exercises. As stated earlier, a normal rehab session takes about an hour to complete. Therefore, the LA522 9V Battery is able to support the Smart Brace without having to worry about it dying during a rehab exercise.

Arduino Nano: 40 mA x 5 pins = 200 mA (Shown in Table 2)

Potentiometer: 1 mA (Calculated in Figure 5)

Bluetooth Modem: 30 mA (Shown in Table 4)

200 mA
1 mA
+ 30mA
----
231 mA being discharged

Using data sheet, 231 mA ≈ 750 mA$\cdot$ Hr

$$750\ mA \cdot Hr \circ \frac{1}{200\ mA} = 3.75 \text{ Hours to Discharge to 5.4 Volts}$$

Figure 6: Calculations to find battery life of Lithium LA522

5.5 Data Transmission

After much thought, I decided that 300 samples would be taken every second. Since the max angular velocity of a user's knee is roughly 300 degrees per second, taking 300 samples per second guarantees that no values will be missed. Like a stated earlier, sending the data in buffers is more efficient. This is because this technique allows the Arduino to read in another set of values into another buffer during the 200ms of downtime that exists while the previous buffer is being sent to the computer. When the computer is done converting the data in one buffer to degrees, they can immediately start the same process with the next buffer that was already sent by the Arduino. I decided that 50 Bytes of data should be packaged and sent at a time. To figure out if the delivery time is acceptable for this project, the amount of time it takes to transmit a buffer of data from the Arduino to the computer was calculated. The calculations are shown below in Figure 7. The throughput of Bluetooth is 2.1 Mbps (21). As a result, when dividing it by the size of the buffer, we get0.19048 ms. When added to the latency of Bluetooth, the delivery time that it takes to send one buffer of data from the Arduino to the computer is 200.19048ms.

Delivery Time of One Buffer of Data = Latency + Size/Throughput

Latency = 200ms

Size = 50 Bytes * 8 = 400 bits

Throughput = 2.1 Mbps = 2100000 bps

Delivery Time One Buffer of Data = 200 ms + 400 bits / 2100000 bps

= 200 ms + 0.00019048 s

= 200 ms + 0.19048 ms

= 200.19048 ms

Figure 7: Calculations for Transferring Data

5.6 User Interface

The computer that will be used to test the Smart Brace will be my MacBook Pro. However, any computer that has Bluetooth capabilities can be used. When the program is run and the patient begins their rehab exercise, the Arduino will send the range of motion values in packets to the computer. Once a packet of data is transferred to the computer, it will store these values in an array. Once all the buffers arrive at the computer, the computer will print the max range of motion value. In addition, it will display a graph of the results for the exercise. This will be similar to a stock market ticker.

5.7 Ordered Parts

Table 6 below shows the parts ordered to complete the Smart Brace.

QTY	ITEM(S) ORDERED	UNIT PRICE	TOTAL
	Robotshop.com		
1	Arduino Nano USB Microcontroller v3 Product Code: RB-Gra-01 http://www.robotshop.com/en/arduino-nan-v-3.html	$34.99	$34.99
1	USB to Mini B Cable 1.3m Product Code: RB-All-33 http://www.robotshop.com/en/usb-mini-b-cable-1-3m.html	$2.40	$2.40
	Sparkfun.com		
1	SparkFun Bluetooth Modem-BlueSMiRF Gold Part Number: WRL-12582 https://www.sparkfun.com/products/12582	$34.95	$34.95
1	Break Away Headers – Straight Part Number: PRT-00116 https://www.sparkfun.com/products/116?_ga=1.42913389.993224844.1447628979	$1.50	$1.50
	Amazon.com		
4	Lithium LA522 9V Battery ASIN Number: B0046TV5VO http://www.amazon.com/Energizer-LA522-Industrial-Lithium-Detectors/dp/B0046TV5VO	$9.15	$36.60
1	Pair 9V Battery Holder Storage Case On/Off Switch With Cap 2 Pcs ASIN Number: B00FHJTOVU http://www.amazon.com/Pair-Battery-Holder-Storage-Switch/dp/B00FHJTOVU/ref=sr_1_6?ie=UTF8&qid=1447705000&sr=8-6&keywords=9V+battery+pack	$2.58	$2.58
	Mouser.com		
1	Potentiometer Mouser Part Number: 882-MW22B-3-5K Manufacturer Part Number: MW22B-3-5K Manufacturer: ETI Systems http://www.mouser.com/ProductDetail/ETI-Systems/MW22B-3-5K/?qs=sGAEpiMZZMvygUB3GLcD7ngsAX7l9Vbcje%2fGy1c6G2Q%3d	$17.35	$17.35

Table 5: Parts ordered and their price

5.8 Winter Term Schedule

The schedule for next term is should below in Table 7.

Week 1	Set up potentiometer and use laboratory equipment to obtain data
Week 2	Continue to set up potentiometer and use laboratory equipment to obtain data
Week 3	Attach potentiometer to knee brace, and read data into microcontroller
Week 4	Continue to attach potentiometer to knee brace, and read data into microcontroller
Week 5	Set up Bluetooth module, and send data to computer (easier to see data and make necessary changes to how data is being transmitted)
Week 6	Adjust code to make sure data is being presented correctly
Week 7	Make any final changes, and test the device
Week 8	Prepare for Presentations
Week 9	Finalize any tests that need to be done
Week 10	Make last minute changes and write Final Report

Table 6: Timeline for Winter Term

6. Final Design and Implementation

In this section, each component of the final system will be discussed. After testing each item that was proposed in the previous fall term proposed design section, it was decided that some of the items needed to be changed. The reasons for choosing a new component will be stated. An image of the Smart Brace is shown below in Figure 5.

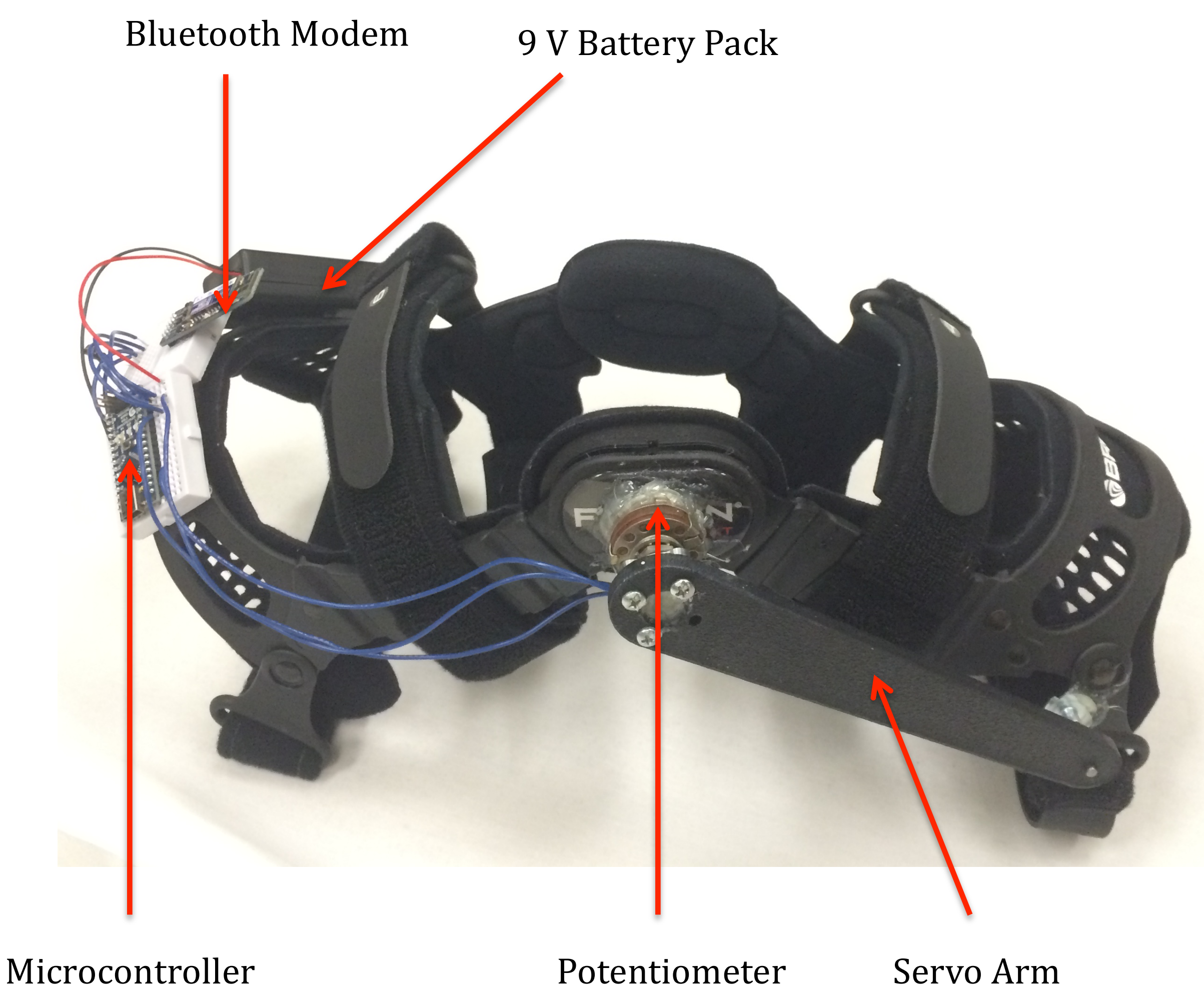


Figure 8: Smart Brace with components

The microcontroller, Bluetooth modem, and battery pack are located on the top section of the knee brace (Left side of Figure 8). Furthermore, the potentiometer is placed on the hinge of the brace. Using a Bore clamping hub, three screws, and a servo arm, the shaft of the potentiometer is attached to a metal rod located at the bottom of the knee brace (Right side of Figure 8). This allows the shaft of the potentiometer to rotate and measure the range of motion when the user bends their knee.

6.1 Microcontroller

In the end, we decided to stick with our preliminary idea and use the Arduino Nano 3.0. As stated earlier, the Arduino Nano is perfect because it is small enough to fit on the brace without interfering with the user's movements. With dimensions of 44mm x 18mm and a mass of 5 grams, it satisfies the design requirement of each component having to be smaller than 50mm x 30mm and no greater than 5 grams. In addition, the Arduino Nano is the best choice because the input voltage is between 7-12 volts. Unlike the LilyPad Arduino USB, the Arduino Nano could use a 9 Volt battery to provide enough current to power the Smart Brace for more than an hour.

6.2 Potentiometer

In the fall, we planned on using a 1 W 5KOhms 5% precision potentiometer manufactured by ETI Systems. However, this is not the potentiometer we ultimately ended up using. This is because a 3-turn potentiometer lowered the ability to obtain accurate results. The Arduino Nano contains a 10-bit analog to digital converter. Therefore, it maps the voltages between 0 to 5 volts into integer values between 0 and 1023. This means that the smallest resolution between readings is 5 volts/1024 units = 4.9 mV per unit. When using the 3-turn potentiometer, each unit of reading results in a change of 1.06 degrees. The calculations are shown below in Figure 9.

(0 volts, 0 degrees)   (5 volts, 1080 degrees)

$$m = \frac{1080-0}{5-0} = 216 \text{ degrees/volts}$$

$$216 \times 0.0049 \text{ volts} = 1.06^\circ$$

Figure 9: Calculations for change in degrees per one unit of reading (Old Potentiometer)

One of the design requirements is that range of motion accuracy was to be within one degree. If we used the ETI Systems potentiometer, 4.9 mV would produce over one degree of rotation. As a result, we needed to use a different potentiometer to produce more accurate results.

The potentiometer we used in the final design was a Honeywell Type J Hot-Molded Panel Potentiometer (27). The specifications are given below in Table 8.

Resistance	5K Ohms
Linearity	+/- 1.5%
Tolerance	10%
Number of Turns	1 (311 degrees)
Size	29.36mm diameter

Table 7: Specifications of the Potentiometer used on the Smart Brace

The biggest advantage of using this potentiometer compared to the potentiometer in the preliminary design is that it rotates 311 degrees instead of 1080 degrees. This is extremely important, because each unit of reading now only results in a change of 0.30 degrees.

(0 volts, 0 degrees)   (5 volts, 311 degrees)

$$m = \frac{311-0}{5-0} = 62.2 \text{ degrees/volts}$$

$$62.2 \times 0.0049 \text{ volts} = 0.30°$$

Figure 10: Calculations for change in degrees per one unit of reading (New Potentiometer)

By using the Type J potentiometer, it takes 16.33 mV to produce one degree of rotation. Therefore, using this potentiometer yields more accurate results that met the accuracy design requirement.

Since this potentiometer still has a resistance of 5K Ohms, the power dissipation is still the same as the original potentiometer in the preliminary design. The calculations can be seen in Figure 5.

6.3 Bluetooth Modem

In the preliminary proposed design, the Smart Brace was going to use the SparkFun Bluetooth Modem – BlueSMiRF Gold. However, the computer never could connect to the Bluetooth modem. After testing the component several times using many different computers, we determined that the device was broken. As a result, we decided to switch to a different Bluetooth modem. The Bluetooth modem that is used on the Smart Brace is called the Adafruit Bluefruit EZ-Link Breakout (28). Table 9 includes the specifications for this device.

Supply Voltage	5V
Serial Communication	Detects and Changes Baud Rate
Power Consumption:	30 mA
Size	40.64 mm x 20.32 mm

Table 8: Specifications of the Bluetooth Modem used on the Smart Brace

We chose this Bluetooth modem, because it operates at a supply voltage of 5 volts. The SparkFun Bluetooth Modem – BlueSMiRF Gold that we planned on using originally used a supply voltage of 3.3 volts. This required the use a voltage divider when hooking it up to the Arduino, to bring down the voltage from 5 volts to 3.3 volts. However, the Adafruit Bluefruit EZ-Link Breakout can connect straight to the Arduino without setting up a circuit to bring down the voltage.

Another reason why we decided on using this Bluetooth modem is because it detects and changes the baud rate on its own. The baud rate does not have to be manually set to agree with the baud rate of the Arduino Nano.

Finally, with the dimensions of 40.64 mm x 20.32 mm, it sits on the Smart Brace without interfering with the user's movements.

6.4 Battery and Battery Life

The Smart Brace uses the Energizer LA522 9V Battery that was proposed in the preliminary proposed design (Section 5.5). As calculated in Figure 6, it will take roughly 3.75 hours to discharge the battery from 9V to 5.4V. This is perfect, because the LA522 9V Battery is able to support the Smart Brace during an entire hour-long rehab session.

6.5 Additional Components

In order for the potentiometer to rotate when the user bends their knee, a component had to be attached to the shaft of the potentiometer. After searching for different solutions to this problem, I ultimately decided on using a servo arm. The servo arm being used is called a 6-inch Gearbox Arm from ServoCity (29). Also, a 5/16 inch Bore Clamping Hub and 3 6-32x3/8 inch Pan Head Phillips Machine Screws are used to tightly attach the servo arm to the shaft of the potentiometer (30, 31). Finally, a metal rod connected the servo arm to the bottom of the knee brace, allowing the potentiometer to rotate with the knee brace.

6.6 Memory and Data Transferring

As stated in Section 5.6, the Smart brace records values 300 times per second. This is ensures that no values will be missed, because the angular velocity of a person's knee when they bend it is 300 degrees per second. In addition, the Arduino Nano reads in a value from the potentiometer, and stores it in a buffer. This process loops until the buffer is filled with 50 values. Then, the buffer is sent over Bluetooth to the computer. Once the buffer is sent, the Arduino reads in 50 more values into another buffer and sends it to the computer. This process continues until the rehab exercise is over and all the data is sent. A block diagram is shown below in Figure10. Also, the Arduino code is shown in Figure 11 in Appendix A.

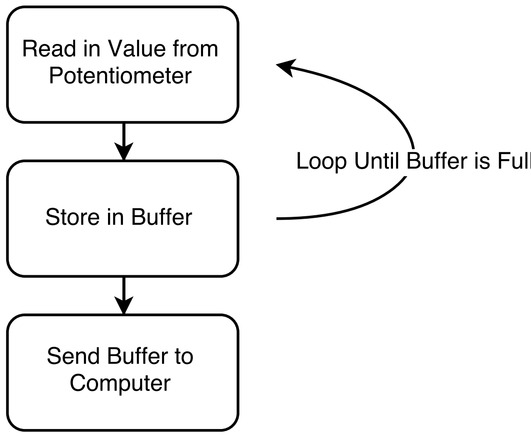


Figure 11: Block Diagram of Data Transmission (Arduino)

On the computer side, the Smart Brace uses Python to read in the values. Python was chosen, because I became familiar with the programming language during my introduction to computer science class. I wanted to further gain experience programming in Python as a result of completing the project. When the computer reads in the incoming buffer from the Arduino, it converts the voltage readings into degrees. This process repeats itself for every buffer. A block diagram of this is shown in Figure 11. Once all the buffers arrive, the max range of motion value and a graph of all the data are printed to the screen. In addition, the Arduino code is shown in Figure 12 in Appendix A.

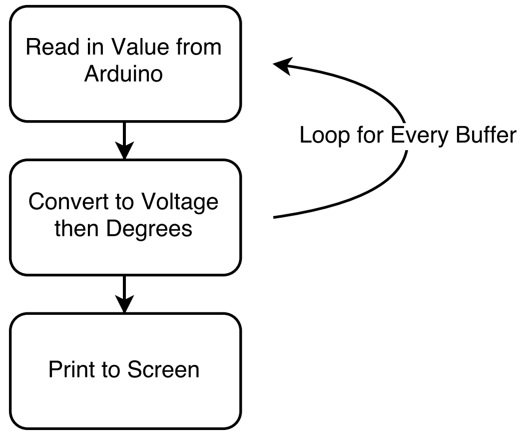


Figure 12: Block Diagram of Data Transmission (Python)

7. Cost Analysis

Table 10 contains the cost of the items that are apart of the Smart Brace.

QTY	ITEM(S) ORDERED	UNIT PRICE
1	Arduino Nano USB Microcontroller v3	$34.99
1	Bluefruit EZ-Link – Bluetooth Serial Link & Arduino Programmer – v.13	$22.50
1	Honeywell Type J Hot-Molded Panel Potentiometer	$34.88
1	6 inch Gearbox Arm (Single)	$7.95
1	5/16 inch Bore Clamping Hub (0.770 inch)	$7.99
3	6-32x3/8 inch Pan Head Phillips Machine Screws (Zinc-Plated)	$0.06
1	Metal Rod	$2.50
3	Breadboard – Mini Modular (White)	$3.95
1	Break Away Headers – Straight	$1.50
1	Lithium LA522 9V Battery	$9.15
1	9V Battery Holder Storage Case On/Off Switch	$2.58
		Total: $136.07

Table 9: Parts ordered and their price

The grand total of $136.07 does not include the components that were bought, but not used. In addition, I had my own ACL knee brace, so we did not need to purchase it. The Breg Fusion Knee Brace costs around $530. Therefore, the project has an estimated cost of about $666. This value is extremely reasonable. Even though the Breg Fusion Knee Brace costs around $530, there are other knee braces that are much more expensive. As stated, the Breg Custom Fusion XT with AirTech brace has a retail value of $799.99. The only added benefit to this brace is that contains more padding. The Smart Brace is much more useful than this brace, because people can use it to track their rehab progress. Thus, valuing this project at $666 is appropriate.

8. Performance Estimations and Results

Figure 11 is the graph of a 60 second rehab exercise. Each oscillation of the graph represents the one cycle of the user bending and straightening their knee. In addition, the max range of motion value for this exercise is 110.73 degrees. This value is located beneath the graph.

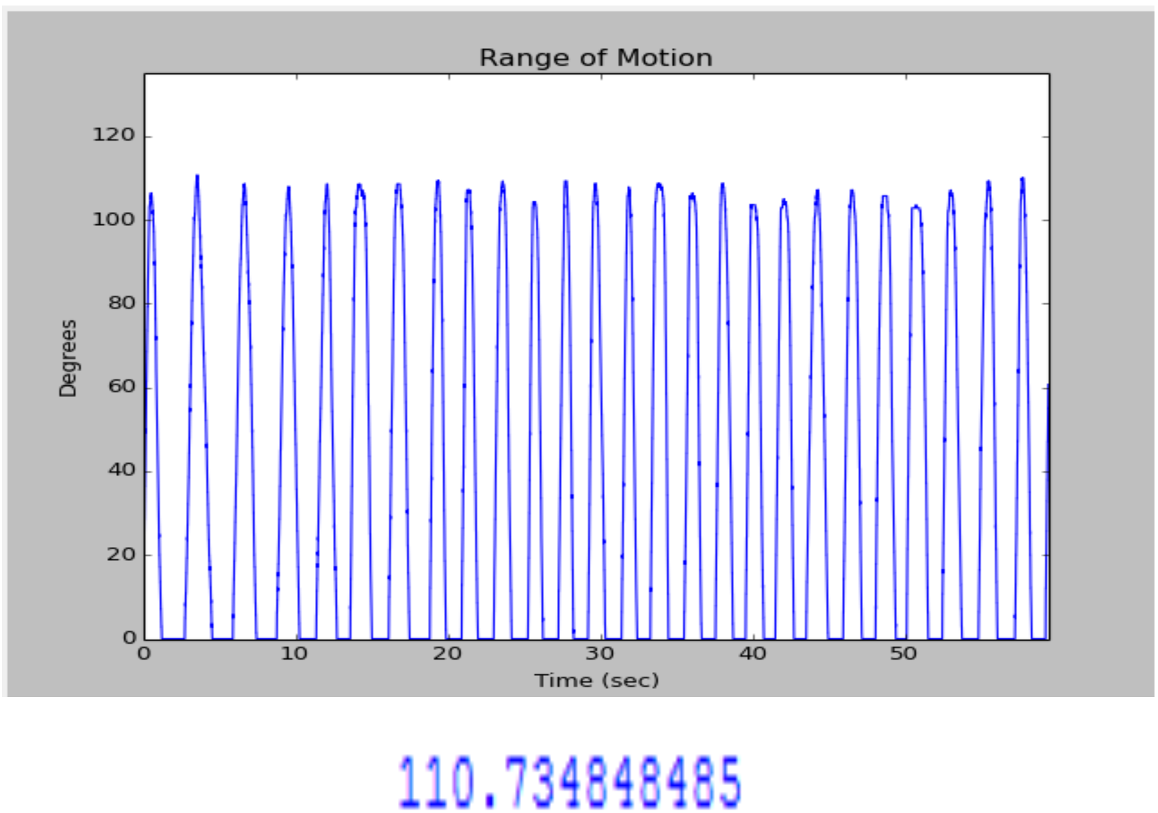
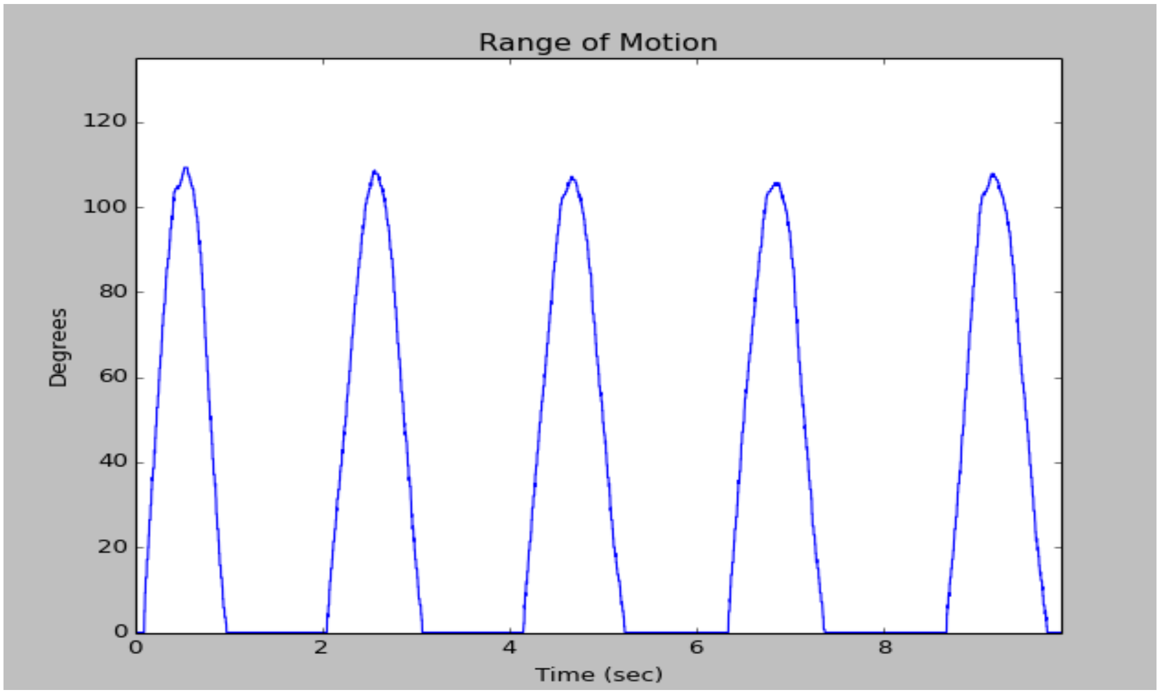


Figure 13: Graph of 60 second rehab exercise/max range of motion value

Also, Figure 12 is the graph of a 10 second rehab exercise. I included this plot in the report, so the plot of the cycles of the user bending and straightening their knee can be seen more clearly. In this exercise, the max range of motion value was 109.31 degrees



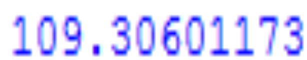


Figure 14: Graph of 10 second rehab exercise/max range of motion value

To test the accuracy of the angle measurement, I measured eight separate angles of rotation using the potentiometer. After receiving the angle of rotation value, I measured the same angle using a protractor, and compared the results. Table 7 shows the results of the test.

Angle Measured with Potentiometer	Same Angle Measured with Protractor	Difference
17.8604	21.5	4.1396
30.0056	33.5	3.4944
41.4363	45	3.5637
68.5842	72	3.4158
82.8725	86.5	4.1275
100.7330	104	3.267
112.1732	115.5	2.8268
119.5102	123	3.4898
		Average Difference: 3.54

Table 10: Results for range of motion accuracy

As you can see, the average difference between the angle measured with the potentiometer and the same angle measured with a protractor is 3.54 degrees.

The following table states the design requirement goals that were determined at the beginning of the project. Also, it states whether or not these goals were met.

Objective	Requirement	Met
Size of Components	< 50mm x 30mm (1.97" x 1.18")	Yes
Mass	<= 5 grams per component	Yes
Battery Life	> 1 Hour of use	No
Samples Per Second	~300 samples per second	Yes
Memory	>= 2 KB	Yes
Range of Motion	0-135 degrees	Yes
Range of Motion Accuracy	Within 1 degree	No
User Interface	Display max values Graph of results	Yes
Price	Not a lot greater than $530	Yes

Table 11: Assessment of Design Requirement Goals

As you can see, almost all of the design requirement goals were met. The sizes of all the components of the Smart Brace were less than 50mm x 30mm, and the mass of the components was no more than 5 grams Furthermore, the Smart

Brace records 300 values per second, ensuring that no data is missed. Also, the memory of the Arduino Nano is 2 KB, and the potentiometer can successfully measure the range of values from 0 to 135 degrees. Finally, the goal of displaying the max value as well as a graph of the results was met, and the final price of the Smart Brace is roughly $666. This cost is correctly marketed within the price range of $530 to $799. Unfortunately, the Smart Brace fails to have a range of motion accuracy within one degree. When comparing the angle of rotation measured using the potentiometer to the same angle measured using a protractor, the angle was off by roughly 3.5 degrees. After trying to figure out the source of this error, I ultimately determined that the potentiometer was not lined up accurately when attached to the brace. When the knee brace was perfectly straight at zero degrees, the potentiometer was not at zero degrees. To account for this human error, an offset is included in the Python Code. As a result, the Smart Brace now can read the range of motion values with an accuracy of less than one degree. The other design requirement that was not met was having a battery life of greater than an hour. Even though the calculations in Figure 6 state that the battery can support the Smart Brace for 3.75 hours, I did not test this calculation. I did not actually run the system for an hour to see if the battery life is okay. While the Smart Brace is promising, it cannot be considered a success until all the necessary tests are performed.

9. Production Schedule

The following section will discuss the phases of the design and implementation of the project. Also, it will recommend any improvements that could have been made in the scheduling and planning.

Like all things in life, the phases of the design did not go as preliminary planned. Table 11 provides the weekly progress of the Smart Brace for winter term.

Week 1	Set up ETI Systems potentiometer and used laboratory equipment to obtain voltage readings
Week 2	Programmed the Arduino to read in values from the potentiometer, and displayed the data on Arduino's serial monitor
Week 3	Determined that this potentiometer did not produce data that was accurate enough. Repeated week 1 and week 2 with Honeywell Type J potentiometer
Week 4	Tested SparkFun Bluetooth Modem-BlueSMiRF Gold on several different computers. Determined that it was not working. Ordered Bluefruit EZ-Link Breakout and waited for it to arrive

Week 5	Set up the new Bluetooth modem and worked on Python code to receive buffered data from Arduino
Week 6	No progress. Had a lot of work in other courses
Week 7	Attached all the components to the knee brace
Week 8	Made some last changes to Python code, eliminating spikes in data. Tested the device and obtained results
Week 9	Prepared for presentation and designed the poster
Week 10	Worked on the final report

Table 12: Winter Term production schedule

One improvement I could have made in the scheduling and planning is to have given myself more time to test the device. Ideally, an extra week or two would have allowed me to complete all the tests I intended to. For example, I wanted to test the battery life calculation performed in Figure 6. To test this, I would have turned on the battery and let it power the Smart Brace, until the battery drained to the point that it could not power the device anymore. Unfortunately, having to choose a new potentiometer and Bluetooth module took up time that could have been spent testing the battery. If I had an extra week, the test could have easily been performed.

10. User's Manual

This section provides a "user's manual" for the operation and maintenance of the Smart Brace. We detail the wiring of all the components and program downloading.

10.1 Wiring

In order to use the Smart Brace, the connections of all the components must be verified. Figure 11 shows the correct wiring.

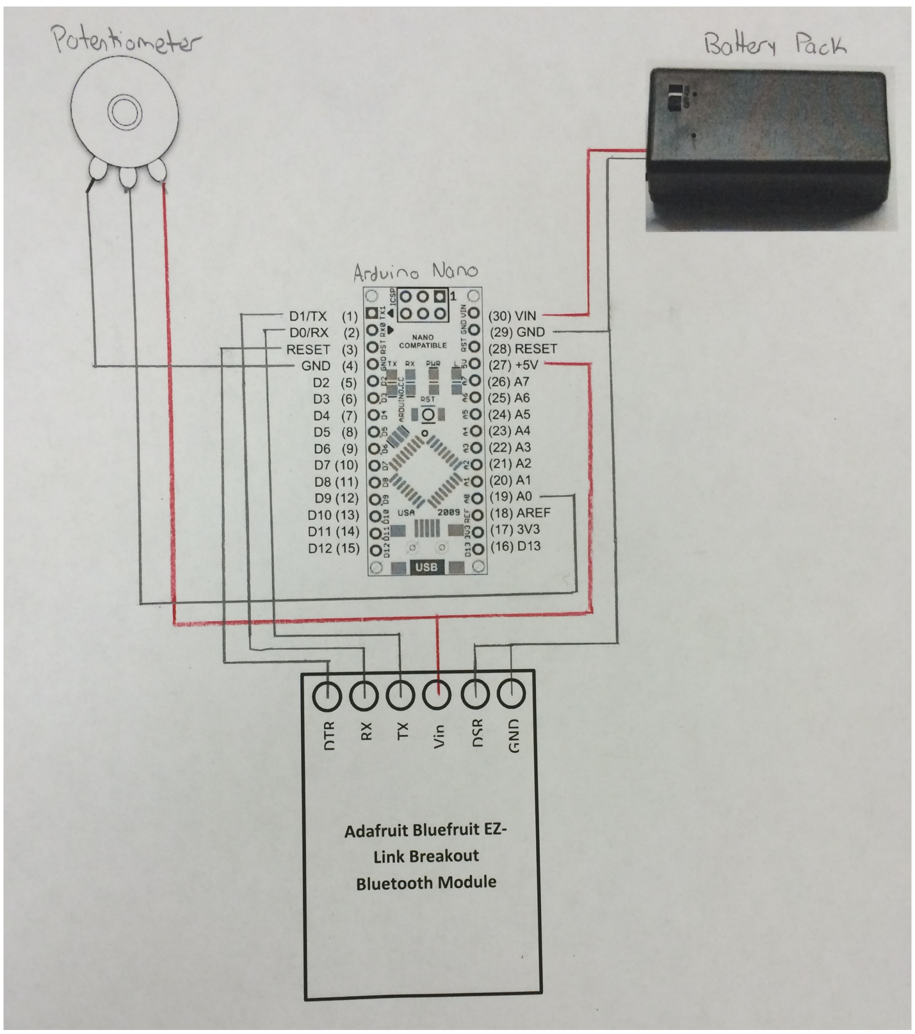


Figure 15: Wiring of the Smart Brace

10.2 Program Downloading

Before the rehab session begins, the following steps should be performed to make sure the Smart Brace and computer are ready to receive the data.

1. Make sure the power switch on the battery pack is in the ON position.
2. Turn Bluetooth on the computer ON, and connect to the Adafruit Bluefruit EZ-Link Breakout (Will appear as "Adafruit EZ-Link 879e").
3. Open Arduino IDE. Under Tools, choose the board to be "Arduino Nano", processor to be "ATmega328", and port to be "/dev/cu.AdafruitEZ-Link879e-SPP".
4. Copy in the Arduino code from Figure ___, compile the sketch, and then upload to the board.
5. Open Python IDLE, and create a new file with the python code from Figure __
6. Run the module.
7. Begin performing the rehab exercises!

11. Conclusions and Recommendations

Tearing an ACL is a heartbreaking moment in many people's lives. In order to return back to daily activities, they first have to go through an intensive exercise rehabilitation program. These programs include stretches and exercises for the patient to complete every single day. Sadly, a huge problem with these programs is that they are not geared towards the specific patient. Everyone recovers at a different pace, and thus deserves their own personalized rehab program. However, this is not the case.

To solve this problem, I have come up with the Smart Brace. This knee brace uses a Honeywell J Type potentiometer to measure the range of motion of the user's knee during rehab exercises. Powered by a Lithium 9 volt battery, an Arduino Nano reads in 300 values per second from the potentiometer. Using an Adafruit Bluefruit EZ-Link Bluetooth modem, the Smart Brace transmits buffered data to the user's computer. The computer uses a Python script to display the max range of motion value, and a graph of the results to the user. By analyzing this information over an extended period of time, the patient can determine how their rehab is progressing.

The Smart Brace was a success. It met almost all the design requirements that were set during the preliminary stage of this assignment. The one requirement that it did not meet right away was the accuracy within one degree. However, this failure was corrected by implementing an offset into the Python code.

Over the last year, I have learned lessons as a result of completing this project. One of the major takeaways I have is that when designing a project, every decision has to be backed up with a valid explanation. For example, I first decided to record values 100 times per second. There was no reason behind this. I just felt that 100 values would have been sufficient enough. I was wrong. Eventually, I conducted some research and found out that the normal angular velocity when a person bends their knee is around 300 degrees per second. Thus, recording 100 values per second would have resulted in the Smart Brace missing 200 data points. Therefore, I changed the Arduino to record 300 values per second. If I realized I needed to back up all my design decisions with facts, perhaps I would have had the extra week to complete all the tests.

Even though I spent two terms working on the Smart Brace, there is still work left to complete. Once again, I want to test the battery life calculations, to make sure they are accurate. In addition, I would love to design an Android application for the Smart Brace. While transmitting the data to a computer is appropriate for this project, I believe receiving the data on a smartphone would be even more beneficial to the patient. In 2015, two thirds of Americans own a smartphone (32). If the patient had the phone on them, they would be able to see the results while performing the rehab exercises. Therefore, they would be able to get immediate feedback on how the are progressing.

12. Acknowledgement

I would like to thank Professor Buma, for his active guidance throughout the completion of the project. Without his help, the Smart Brace would not have been completed to the level that it is today.

13. References

1 "Value of Orthopaedic Treatment: ACL - A Nation in Motion." *A Nation in Motion*. N.p., n.d. Web. 12 Nov. 2015. <http://www.anationinmotion.org/value/acl/>.

2 "ACL Rehabilitation Program." *Emory Healthcare*. N.p., n.d. Web. 12 Nov. 2015. <http://emoryhealthcare.org/acl-program/index.html>.

3 Tion., and 6 2 1 S C I E N C E D R I V E • M A D I S O N , W I 5 3 7 1 1 • U Ws P O R T Sm E D I C I. *Rehabilitation Guidelines for Knee Multi-ligament Repair/Reconstruction* (n.d.): n. pag. Web. 18 Nov. 2015.

4 "Anterior Cruciate Ligament (ACL) Injuries-OrthoInfo - AAOS." *Anterior Cruciate Ligament (ACL) Injuries-OrthoInfo - AAOS*. N.p., n.d. Web. 15 Nov. 2015. <http://orthoinfo.aaos.org/topic.cfm?topic=a00549>.

5 "Knee Brace." *My Blog RSS*. N.p., n.d. Web. 15 Nov. 2015. <http://www.marklfuerst.com/blog/tag/knee-brace/>.

6 "The Most Intelligent Knee Brace Available." *Bend · Intelligent Knee Brace Platform for Healthcare*. N.p., n.d. Web. 15 Nov. 2015. <http://www.bendlabs.com/smartknee/#home>.

7 "ACL Bracing." - *Helping With Prevention, Protection & Healing*. N.p., n.d. Web. 15 Nov. 2015. <http://www.djoglobal.com/education/patient-education/acl-bracing>.

8 "Let's Move." *Get Active*. N.p., n.d. Web. 24 Nov. 2015. <http://www.letsmove.gov/get-active>.

9 "ACL Surgery: Return to Play Not Guaranteed." *ACL Surgery: Return to Play Not Guaranteed*. N.p., n.d. Web. 24 Nov. 2015. <http://www.momsteam.com/health-safety/acl-reconstructive-surgery-return-to-play-not-guaranteed>.

10 "SocialHeadline." *POLITICO*. N.p., n.d. Web. 24 Nov. 2015. <http://www.politico.com/story/2015/07/health-care-spending-hike-prediction-120740>.

11 "U.S. Food and Drug Administration." *Device Approvals, Denials and Clearances*. N.p., n.d. Web. 15 Nov. 2015. <http://www.fda.gov/MedicalDevices/ProductsandMedicalProcedures/DeviceApprovalsandClearances/>.

12 "Breg FUSION Knee Brace." *Breg Fusion Knee Brace*. N.p., n.d. Web. 24 Nov. 2015. <http://www.braceshop.com/breg-fusion-knee-

brace.htm?gdftrk=gdfV28018_a_7c2568_a_7c10961_a_7c28608&gclid=CMuh1rzUqskCFdgPgQodJVUI0Q>.

13 "Breg Custom Fusion W/AirTech Knee Brace." *Breg Custom Fusion W/AirTech Knee Brace*. N.p., n.d. Web. 12 Mar. 2016. <http://www.bracestore.com/breg-custom-fusion-w-airtech-knee-brace.htm>.

14 "MVS 330 Project Title." *MVS 330 Project Title*. N.p., n.d. Web. 19 Nov. 2015. <http://www.umich.edu/~mvs330/w99/strike/results2.html>.

15 "What Is the Normal Range of Motion of the Knee?" *LIVESTRONG.COM*. LIVESTRONG.COM, 16 Aug. 2013. Web. 17 Nov. 2015. <http://www.livestrong.com/article/40176-normal-range-motion-knee/>.

16 "What Is an Arduino?" *What Is an Arduino?* N.p., n.d. Web. 16 Nov. 2015. <https://learn.sparkfun.com/tutorials/what-is-an-arduino>.

17 "Arduino - ArduinoBoardUno." *Arduino - ArduinoBoardUno*. N.p., n.d. Web. 16 Nov. 2015. <https://www.arduino.cc/en/Main/ArduinoBoardUno>.

18 "Arduino - ArduinoBoardLilyPadUSB." *Arduino - ArduinoBoardLilyPadUSB*. N.p., n.d. Web. 16 Nov. 2015. <https://www.arduino.cc/en/Main/ArduinoBoardLilyPadUSB>.

19 "Omron E6B2CWZ5B360PR05M Rotary Encoder, 100, 200 Pulses Per Revolution, 6000 Rpm." *Buy from Cross*. N.p., n.d. Web. 16 Mar. 2016. <https://www.shopcross.com/product/omron-e6b2cwz5b360pr05m-rotary-encoder-100-200-pulses-revolution-6000-rpm?gclid=CJjDocnYyssCFdU7gQodJnUNIg>.

20 "MW22B-3-5K ETI Systems | Mouser." *Mouser Electronics*. N.p., n.d. Web. 17 Nov. 2015. <http://www.mouser.com/ProductDetail/ETI-Systems/MW22B-3-5K/?qs=sGAEpiMZZMvygUB3GLcD7ngsAX7l9Vbcje%2FGy1c6G2Q%3D>.

21 "Bluetooth vs Wi-Fi." - *Difference and Comparison*. N.p., n.d. Web. 16 Nov. 2015. <http://www.diffen.com/difference/Bluetooth_vs_Wifi>.

22 "Latency." *What Is Latency? Webopedia*. N.p., n.d. Web. 17 Mar. 2016. <http://www.webopedia.com/TERM/L/latency.html>.

23 "Arduino - ArduinoBoardNano." *Arduino - ArduinoBoardNano*. N.p., n.d. Web. 17 Nov. 2015. <https://www.arduino.cc/en/Main/ArduinoBoardNano>.

24 "SparkFun Bluetooth Modem - BlueSMiRF Gold." - *WRL-12582*. N.p., n.d. Web. 17 Nov. 2015. <https://www.sparkfun.com/products/12582>.

25 "Robot Check." *Robot Check*. N.p., n.d. Web. 17 Nov. 2015. <http://www.amazon.com/Energizer-LA522-Industrial-Lithium-Detectors/dp/B0046TV5VO>.

26 "Pair 9V Battery Holder Storage Case ON/OFF Switch W Cap 2 Pcs." *Amazon.com : : Two Way Radio Battery Chargers : GPS & Navigation*. N.p., n.d. Web. 18 Nov. 2015. <http://www.amazon.com/Pair-Battery-Holder-Storage-Switch/dp/B00FHJTOVU/ref=sr_1_6?ie=UTF8&qid=1447705000&sr=8-6&keywords=9V%2Bbattery%2Bpack>.

27 "Honeywell J Series Potentiometers." *Honeywell J Series Potentiometers*. N.p., n.d. Web. 29 Feb. 2016. <http://www.mouser.com/Honeywell/Passive-Components/Potentiometers-Trimmers-Rheostats/Potentiometers/J-Series/_/N-9q0yp?P=1yzv72tZ1z0sf9t>.

28 "Introducing Bluefruit EZ-Link Breakout." *Overview*. N.p., n.d. Web. 28 Feb. 2016. <https://learn.adafruit.com/introducing-bluefruit-ez-link/overview>.

29 "Servo Power Gearbox Arms." *Servo Power Gearbox Arms*. N.p., n.d. Web. 20 Feb. 2016. <https://www.servocity.com/html/servo_power_gearbox_arms.html#.VufxpOnBH0s>.

30 "Clamping Hub - 5/16" Bore." - *ROB-12449*. N.p., n.d. Web. 20 Feb. 2016. <https://www.sparkfun.com/products/12449>.

31 "Machine Screws, Phillips Pan Head, Zinc Plated Steel, #6-32 X 3/16"" - *Bolt Depot*. N.p., n.d. Web. 20 Feb. 2016. <https://www.boltdepot.com/Product-Details.aspx?product=9225>.

32 "U.S. Smartphone Use in 2015." *Pew Research Center Internet Science Tech RSS*. N.p., 01 Apr. 2015. Web. 11 Mar. 2016. <http://www.pewinternet.org/2015/04/01/us-smartphone-use-in-2015/>.

12. Appendix A

```
int dt = 3.33;  //sampling interval is 3.33 ms (100 samples per second)

int N = 18000;  //number of samples is 18000 (60 seconds)

const int VALUES = 50;

byte buffer_array[VALUES];

int index;


void setup() {

  Serial.begin(9600);

}


void loop() {

  for (int n=0; n < N; n++){

    for(int index = 0; index < VALUES; index = index + 2){

        //Read analog pin 0

        int voltage = analogRead(A0);

        //Store 1st byte in array

        buffer_array[index] = voltage / 256;

        //Store 2nd byte in array

        buffer_array[index+1] = voltage % 256;

    }
```

```
    //Write out value to computer
    Serial.write(buffer_array, 50);
    //wait for all data to send
    Serial.flush();
    delay(dt);
  }
}
```

Figure 16: Arduino code for the Smart Brace

```
import serial
import numpy as np
import matplotlib.pyplot as plt

#initiates serial connection
potentiometerData = serial.Serial('com8', 9600)

#totalValues/bufferSize
totalLoops = 720

#size of incoming buffer from arduino
bufferSize = 50

#total numbers in buffer
numbersInBuffer = 4

#ratio of potentiometers turn and voltage
potRotation = 311/5

#Number of samples
N=18000
#Total Values in respect to number of samples
totalValues = 2*N
#Number of samples per second (300)
dt=0.0033

#array of zeros to store final value
finalArray = np.zeros([N])

#off set due to failed accuracy of potentiometer
offset = 1.035

i=0

#loop for total number of buffer
for n in range(0, 720):

    #while waiting for buffer to arrive, do nothing
    while (potentiometerData.inWaiting() < bufferSize):
        pass

    #read in buffer
    inputData = potentiometerData.read(bufferSize)
```

```
#loop for every value in buffer
for index in range(0, numbersInBuffer):

    #combine high and low bytes back together, convert to voltage
    finalArray[i] = inputData[2*index] * 256 + inputData[2*index+1]

    finalArray[i] = (finalArray[i]*(5/1023)) * potRotation * offset

    i = i+1

  potentiometerData.flushInput()


#print
print(finalArray)
print(max(finalArray))


#set up the graph
Graph, = plt.plot(60,finalArray)
plt.title('Range of Motion')
plt.xlabel('Time (sec)')
plt.ylabel('Degrees')
plt.xlim(0,N*dt)
plt.ylim(-0.1,135)


potentiometerData.flushInput()

#Plot the graph
plt.ion()
plt.show()
Graph.set_ydata(finalArray)
plt.draw()
```

Figure 17: Python Code for Smart Brace