**Critical Design Review**

**Active Assist Elbow Orthosis**

**Engineering 498**



Table of Contents

1.0 Introduction

2.0 System Description including System Block Diagram

3.0 System Design Details

3.1 Brace Design Details

3.2 Locking Mechanism Design Details

3.3 Motor Design Details

3.4 Motor Driver Design Details

3.5 Clutch Design Details

3.6 Arduino Design Details

3.7 Sensor Design Details

3.8 Power Supply Design Details

3.9 iOS Application Design Details

4.0 Acceptance Test Procedures (ATP)

4.1 Test Procedure Verification and Data Sheet: AEO Lock (1.0)

4.2 Test Procedure Verification and Data Sheet: Motor Torque (2.0)

4.3 Test Procedure Verification and Data Sheet: Position Tolerance (3.0)

4.4 Test Procedure Verification and Data Sheet: AEO Sensor (4.0)

4.5 Test Procedure Verification and Data Sheet: Angular Velocity (5.0)

4.6 Test Procedure Verification and Data Sheet: Sensor App Accuracy (6.0)

4.7 Test Procedure Verification and Data Sheet: Temperature Range (7.0)

4.8 Test Procedure Verification and Data Sheet: Weather (8.0)

4.9 Test Procedure Verification and Data Sheet: Battery Life (9.0)

4.10 Test Procedure Verification and Data Sheet: Battery Recharge (10.0)

5.0 Models and Analyses

5.1 Torque Model

5.2 Motor Power Model

5.3 Sensor Model

5.4 Weight Analysis

5.5 Temperature Analysis

5.6 Sensor Analysis

5.7 Step Angle Analysis

5.8 Sampling Analysis

6.0 Risk And Cost Analysis

6.1 Mitigation Plan

6.2 Cost Analysis

7.0 Appendix

7.1 Detailed Parts and Assembly Drawing

7.2 Wireless Application Instruction Document

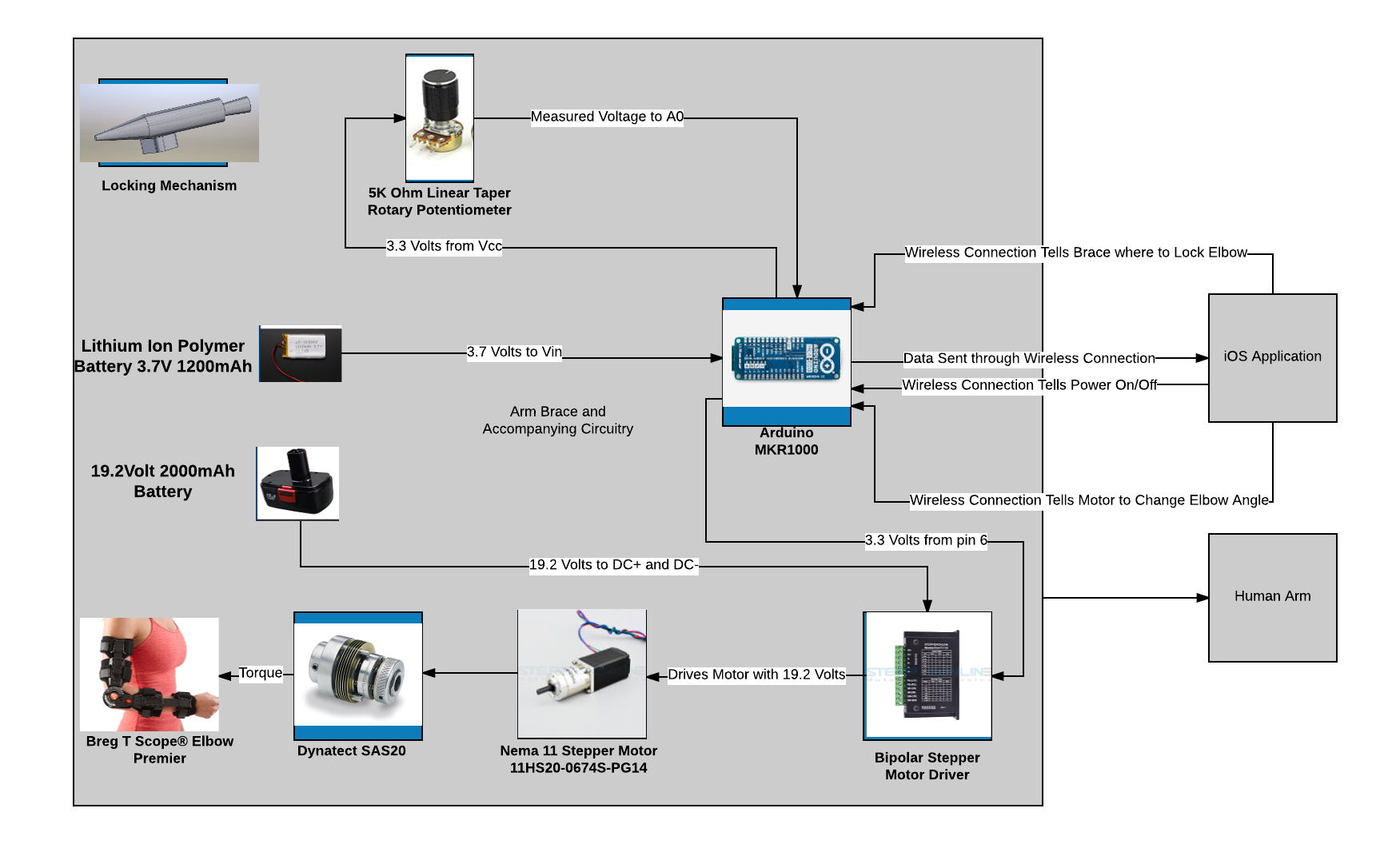
**1.0 Introduction**

Elbow stiffness after surgery is a common and debilitating problem. It is caused by the accumulation of fibrous tissue in the joint and surrounding structures during the healing process and can occur after many types of elbow surgery. It is made worse by the rigid postoperative bracing that is used to stabilize the elbow after surgery. It is commonly treated with range of motion exercises performed daily, but the exercises can be painful and are often ineffective. Thus, there exists an unmet need for a brace that simultaneously stabilizes the elbow after surgery and provides frequent motion to prevent the buildup of scar tissue and maintain the range of motion of the elbow joint.

The goal of this project is to design and create a motor-hinged elbow orthosis that will maintain as much range of motion in the joint throughout the six-month healing process while also breaking down the scar tissue in the joint. This device should aid recovery after surgery, having the advantage over physical therapy of being wearable and user controllable. While performing the flexion/extension movement in the arm, the orthotic will actively assist the patient into moving the arm to a predetermined angle per doctor’s orders. This elbow orthosis will be a staple in the orthopedic surgical department.

Previously, we had met with our sponsor to discuss functional requirements, non functional requirements, and how they should be verified. These requirements and verification methods are all documented in our SRD. These, together with concept trades, analyses, and test methods, were reviewed at the PDR on Nov 15, 2016. This CDR covers our multi-disciplined technical design to ensure that the AEO could proceed into system fabrication, demonstration, and test; and could meet the stated performance requirements within cost, schedule, risk, and other system constraints. While transitioning from the PDR, we discovered a few improvements to the design we had created. After careful consideration of our many design options to choose from, we have optimized our final design.

**2.0 System Description including System Block Diagram**



The Active Elbow Orthosis is broken up into two main distinctions: the arm brace including the mechanical and electrical components on the left and the iOS application on the right. The arm brace is composed of a Breg T Scope Elbow Premier, an Arduino MKR1000, a Nema 11 Stepper Motor, a Bipolar Stepper Motor Driver, a Dynatect SAS20 clutch, a 5K ohm potentiometer (sensor), locking pin, and a power supply consisting of a lithium ion polymer battery and a 19.2 Volt battery. The iOS application will include pages for login, schedule/calendar of exercises, exercise history, and exercise initiation.

The iOS application will interface with the patient-user. Instructions for use of the iOS application will be given in the Technical Data Package. The iOS application will connect through wifi with the Arduino MKR100. The Arduino will send the application information on the angle of the elbow brace. The application will process and store that information. Based on the application exercise schedule, the application will send instructions to the Arduino on when and to what extent the motor will act in assisting elbow flexion or extension.

The Arduino is the focal point of the arm brace electromechanical system. The Vcc pin on the Arduino will be connected to the 5K ohm potentiometer in series with a 1K ohm resistor. The GND pin of the Arduino will be connected to the other side of the potentiometer-resistor series. In between the potentiometer and the resistor, the A0 pin will read the voltage at that node and convert it to a digital signal. The A0 pin is a 12 bit analog-to-digital converter. The Arduino reads the voltage between the resistor and potentiometer and converts that voltage to an angle reading with the equation from analysis. The relationship between the voltage at the node between the resistor and potentiometer and the elbow angle is: V = 88(. The angle reading is sent to the iOS application through the wifi connection. The sampling rate for the ADC is 10,000 Hz, which is well above the minimum required sampling rate.

The Arduino MKR1000 is powered by the lithium ion polymer battery which will supply 3.7 volts to the Vin pin of the Arduino. The lithium ion polymer battery has 1200 mA-hours. During research on the Arduino, the minimum recommended battery size was 700 mA-hours, which means we are well beyond the minimum.

Upon instruction from the iOS application, the Arduino MKR1000 connects 3.3 volts from pin 6 to control the bipolar stepper motor driver. The bipolar stepper motor is controlled by the Arduino and powered by the 19.2 volt battery with 2000 mA-hours. The stepper motor driver will drive the motor with the 19.2 volts from the battery. The motor has a 14:1 gearbox on its end to increase torque. The end of the gearbox will interface with the Dynatect clutch to prevent motor burnout. The clutch and motor will interface with the elbow brace. Upon instruction from the Arduino the motor driver will drive the motor which will add torque at the elbow brace joint. The added torque will aid the patient in flexion and extension of the arm.

The locking mechanism is in place to prevent the patient from feeling pain. The locking mechanism is always accessible to the patient in order to lock the elbow brace in place. Upon consideration, the locking system was kept purely mechanical so that electrical failure is not a problem.

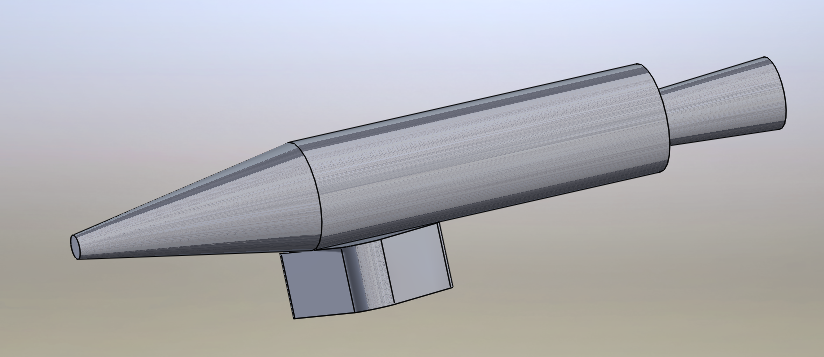
**3.0 System Design Details**

**3.1 Brace Design Details**



A Breg T Scope Premier Brace was chosen for the device. The brace has a weight allocation of 0.81 pounds and a maximum length allocation of 464 millimeters. It allows for extension control of -10 to 110 degrees and flexion control of 10 to 120 degrees. There are four velcro straps attached to brace which tighten around the arm and forearm: two on the main frame of the brace and two connected to the extending struts.

**3.2 Locking Mechanism Design Details**



A locking mechanism will be implemented in case the patient experiences any pain or discomfort. The mechanism consists of a spring-action piece that will be constructed to connect to the outer casing of the device. There will be an inner tube and inner-locking parts similar to those found in pens so that the pin will stay locked when pushed and then retract when pushed again. A bracket will secure the piece to provide stability and allow it to latch onto the case at a point where the tip of the mechanism faces downward toward the gear of the brace. The pointed end (shown in the figure above) will thrust into the main gear of the device and prevent it from rotating more. The lock will be located in an easily accessible spot so that the patient can press it and the device will immediately stop extending/flexing.

**3.3 Motor Design Details**

A NEMA 11 Stepper Motor containing an attached planetary gearbox with a 14:1 gear ratio was chosen for this device. The manufacturer part number for this item is 11HS20-0674S-PG14. The motor requires a weight allocation of 0.68 pounds of the allotted 10 pounds for the entire device. The motor with the gearbox will require a space allotment of 30x30x85.5 millimeters. It is recommended that this motor is powered with a battery of 12-24 volts. The gear efficiency of the of the contained gearbox is 81%. The maximum holding torque of the motor is 1.34 N-m and it requires a maximum current of 0.67 amperes to operate. This motor was chosen over other models, because it weighed less which reduces the risk of injury while still delivering the necessary torque.



**3.4 Motor Driver Design Details**

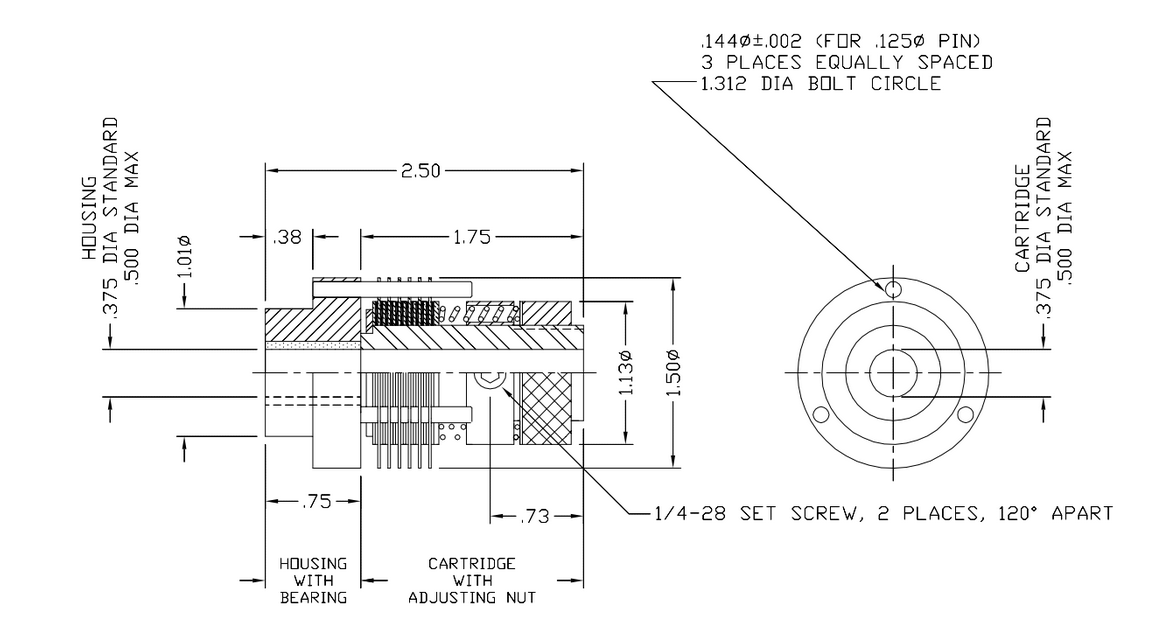
The motor driver chosen for this device is a bipolar stepper motor driver. The item is identified as a 128 High Subdivision ST-7128. This motor driver requires a weight allocation of 0.66 pounds of the allotted 10 pounds for the entire device. The motor driver will require a space allotment of 116x76x26 millimeters. It is recommended that the motor driver is powered with a battery of 10-32 volts. The maximum current this motor driver is able to output is 3 amperes.



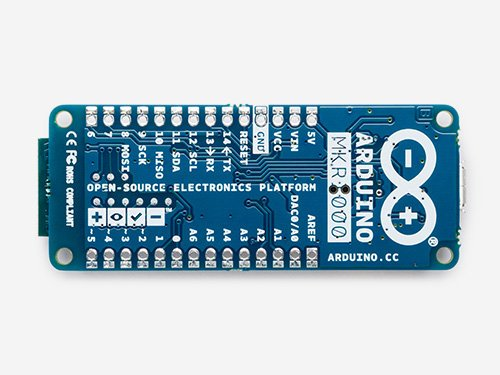
**3.5 Clutch Design Details**

The main reason for choosing a Clutch to be part of this system is to make sure external forces applied by the patient when performing exercises would not risk a stalling of the motor. The clutch chosen is from Dynatect company series No. SAS20. The main qualities of this device are the bidirectional performance, an adjustable maximum torque of 1.35 N-m (falling under the needed torque for the stepper motor). Up to maximal performance, this clutch have the possibility of heating up to 6 Watts. Equation below demonstrates the variables responsible for the clutch to heat up, therefore since RPM in our system is minimal the implementation of a SAS20 in our device shall not contain any harm.

Watts = RPM x torque (in-lbs) x .011 x duty cycle (%)

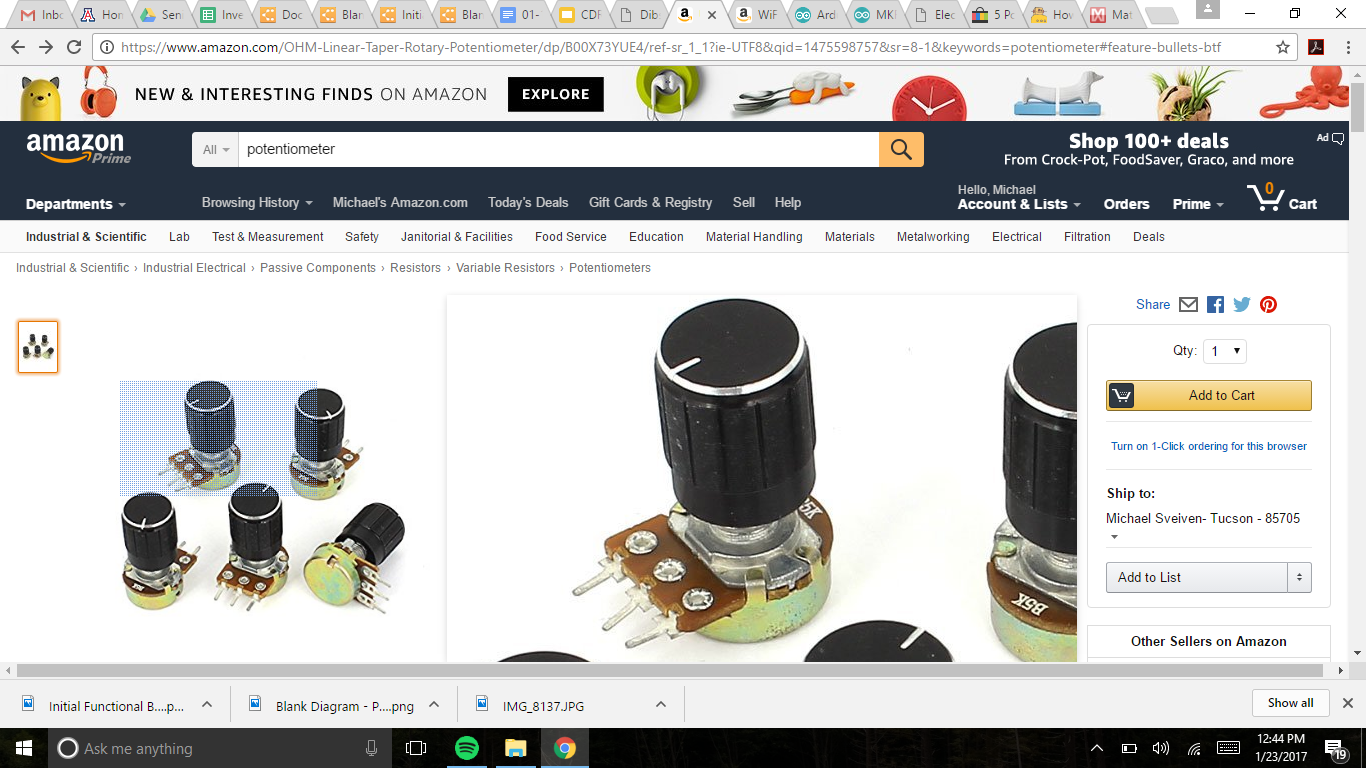
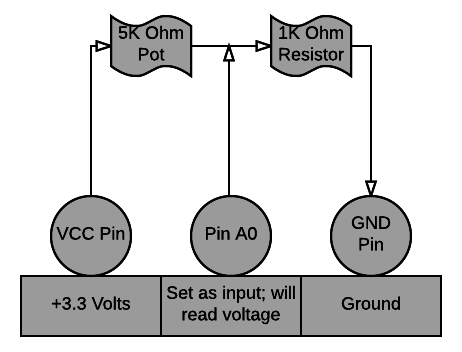


**3.6 Arduino Design Details**



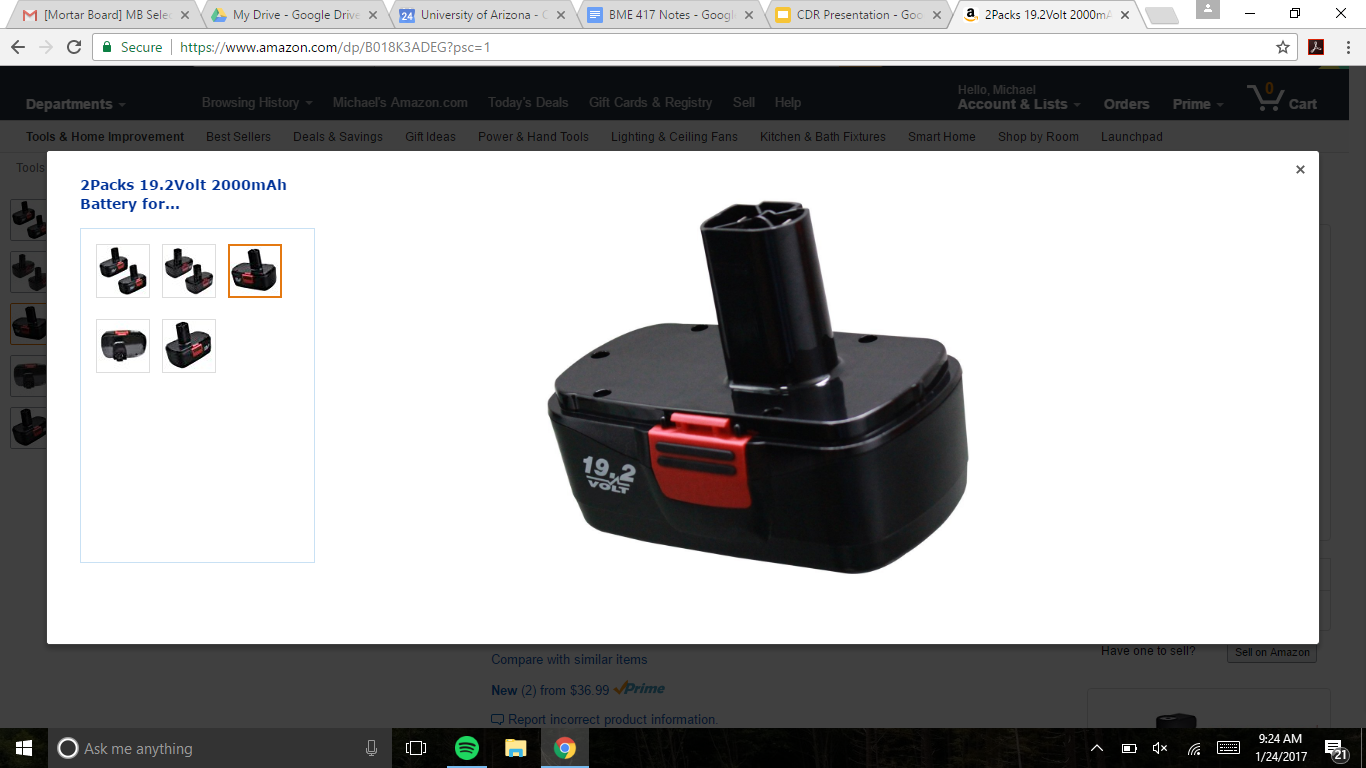
A single-board computer was needed to control the motor and read the angle sensor. The Arduino MKR1000 was chosen because of its size, ADC pins, output pins, and wifi capability. The Arduino MKR1000 is only 46 grams. The size of the Arduino is quite small with respect to the rest of the Active Elbow Orthosis. The length of the Arduino is 61.5 mm and the width is only 25 mm. The Analog-to-Digital Converter pins are 12 bit, which means they have 4096 discrete steps, which is well beyond the system requirements. The Arduino output pins give a maximum voltage of 3.3 V. The Arduino MKR1000 was specifically made to work with the Internet of Things. The Arduino’s wifi connectivity is based on the Atmel ATSAMW25 SoC (System on Chip).

**3.7 Sensor Design Details**



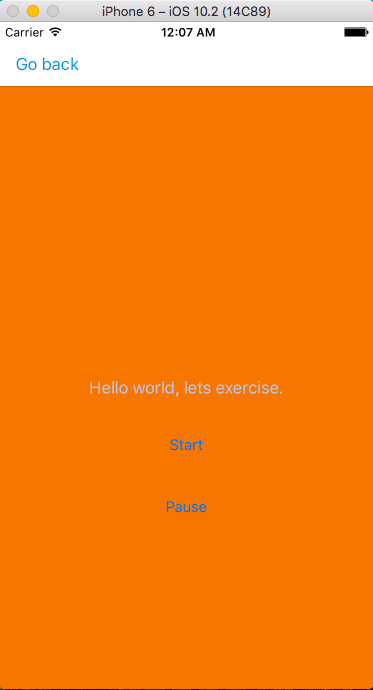
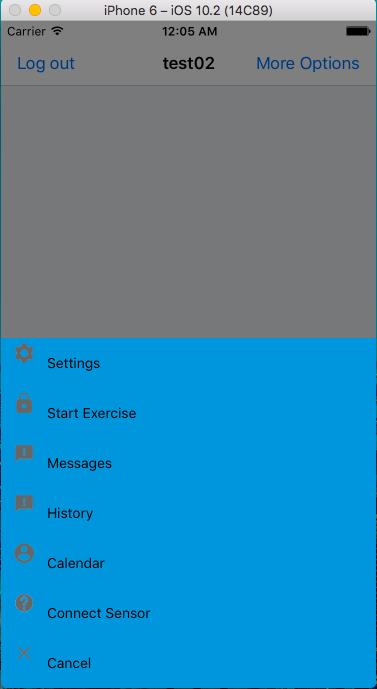
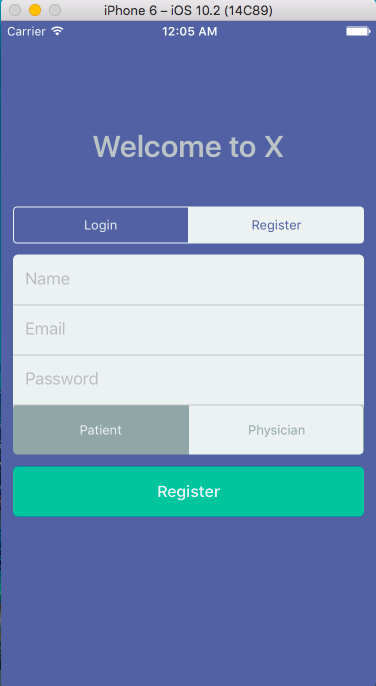
The sensor is a 5K Ohm Linear Taper Rotary Potentiometer. When the knob on the potentiometer is turned, the resistance between the end terminals and the middle terminal changes. The 5K ohm potentiometer was chosen because it is light and simple. The potentiometer is only 46 grams. The potentiometer is also a simple sensor because it can be assembled in a simple circuit that allows the Arduino to read the voltage between the potentiometer and the resistor. The voltage that will be read at node between the resistor and potentiometer is dependent on the resistance of the potentiometer. Because the resistance of the potentiometer changes with the turning of the knob, the potentiometer can be connected to the brace so that the knob turns with the elbow brace. The voltage read by the Arduino is therefore related to the elbow angle. The relationship between the voltage at the node between the resistor and potentiometer and the elbow angle is: V = 88(

**3.8 Power Supply Design Details**



Both the Arduino MKR1000 and the motor need power to perform the functions. The Lithium Ion Polymer battery was chosen because of its small size and because it surpassed the power requirement for the Arduino. The Arduino battery is only 0.3 ounces and 61x33x5 mm. In addition, [www.arduino.cc](http://www.arduin.cc) states that “We strongly recommend that you select a Li-Po battery of at least 700mAh capacity”. The 19.2 V battery was chosen because it was the smallest available battery that also met the necessary power requirement. The 19.2 V battery is 2 lbs and 99x79x71 mm. The 19.2 V battery has 2000 mA-hours, which will be shown to be appropriate later in the analysis/models section.

**3.9 iOS Application Design Details**



The iOS application will control the AEO device and there are several key features that must be included in its programming. There will be a login feature for each the patient and the physician. There will be a schedule/calendar of the patient’s exercises. The application will contain a log of the patient’s previous exercise history. The application will contain a start and stop button for the patient to begin and end an exercise. A timer will track how long the patient has been doing a given exercise. Only the physician will be able to make changes to the exercise schedule of the patient via the physician login. It is required that the application is able to communicate with the microcontroller. The application will be written in English. It has not yet been determined how much storage the application will require on a phone, since it is still undergoing development.

**4.0 Acceptance Test Procedures (ATP)**

**Acceptance Test Procedures**



|  |  |  |
| --- | --- | --- |
| **Issue Date:** | **Revision Date:** | **Contract Name:** |
|  |  |  |

**ATP Introduction:**

The purpose of this Acceptance Test Plan and set of procedures is to establish the overall approach, testing sequence, and specific tests to be conducted to demonstrate successful operation of the AEO prototype. The AEO prototype provides the user with an advanced toolset for maintaining the range of motion post surgery with a state-of-the-art iPhone application. The plan and procedures described in this document are designed to verify stable and safe operational performance of the AEO prototype.

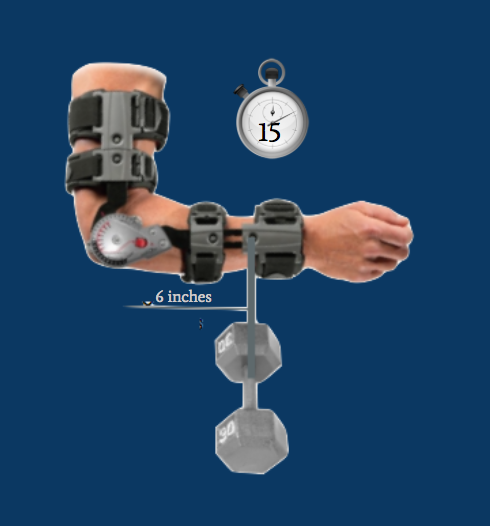
**4.1 Test Procedure Verification and Data Sheet: AEO Lock (1.0)**

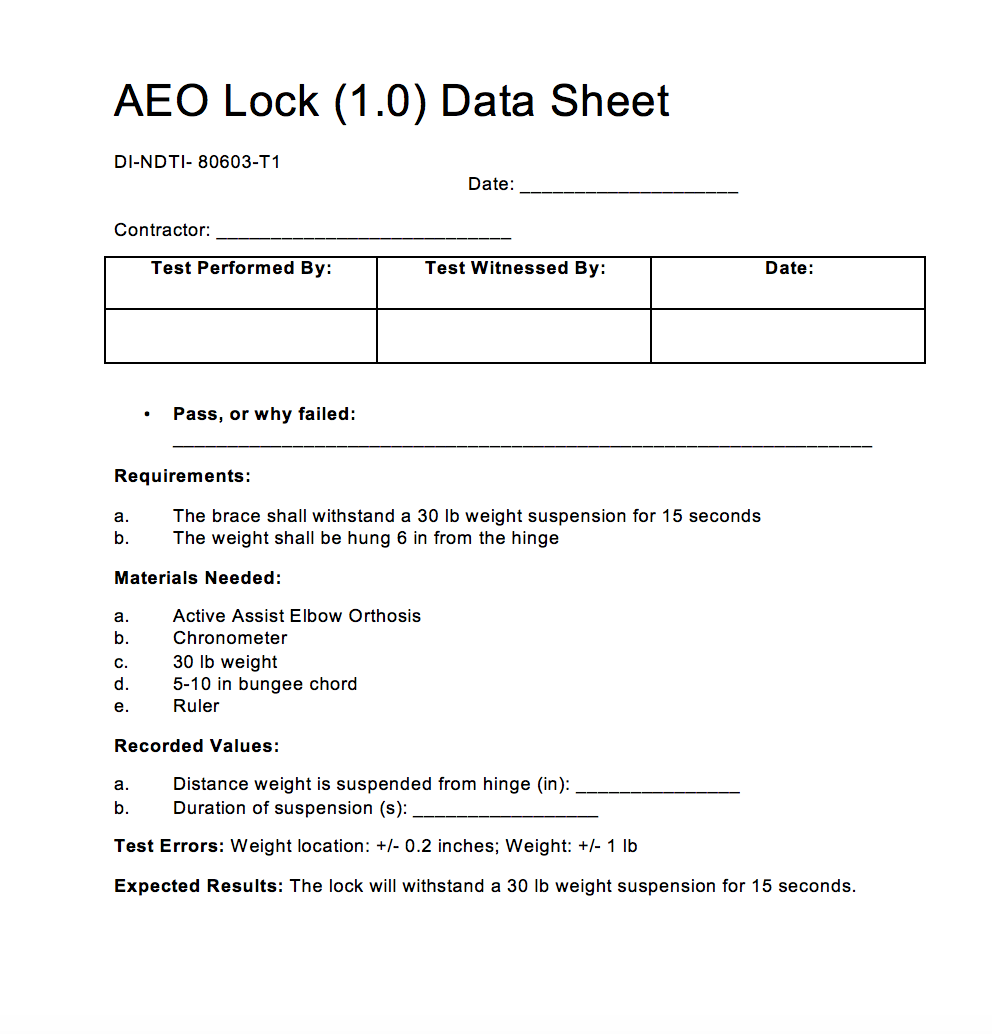
Test Description: This test will verify that the AEO lock will be able to withstand a substantial amount of weight for a specific period of time.

Procedure:

1. The device shall either be worn by a healthy individual, or tightly secured so that the arm attachment is vertical and the forearm attachment is horizontal
2. The lock shall be applied to keep the AEO fixed at the 90˚ angle
3. A 30 lb weight (+/- 1 lb) shall be suspended from the forearm attachment 6 inches (+/- 0.2 inches) from the hinge
4. A chronometer will be used to time the duration of the weight suspension
5. The brace shall withstand this amount of weight for 15 seconds

Diagram:





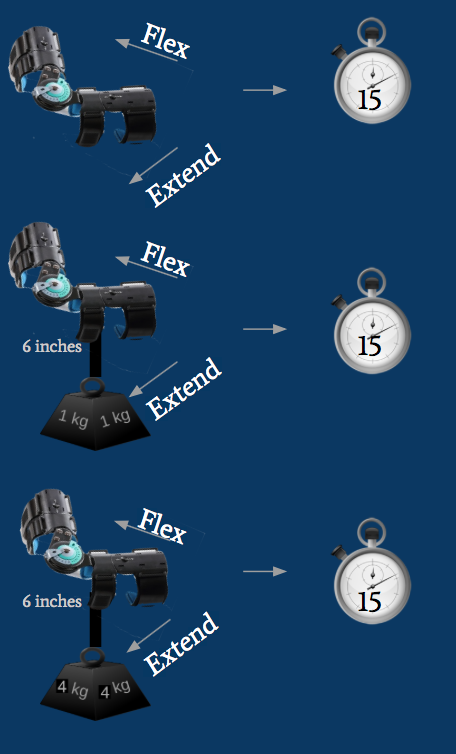
**4.2 Test Procedure Verification and Data Sheet: Motor Torque (2.0)**

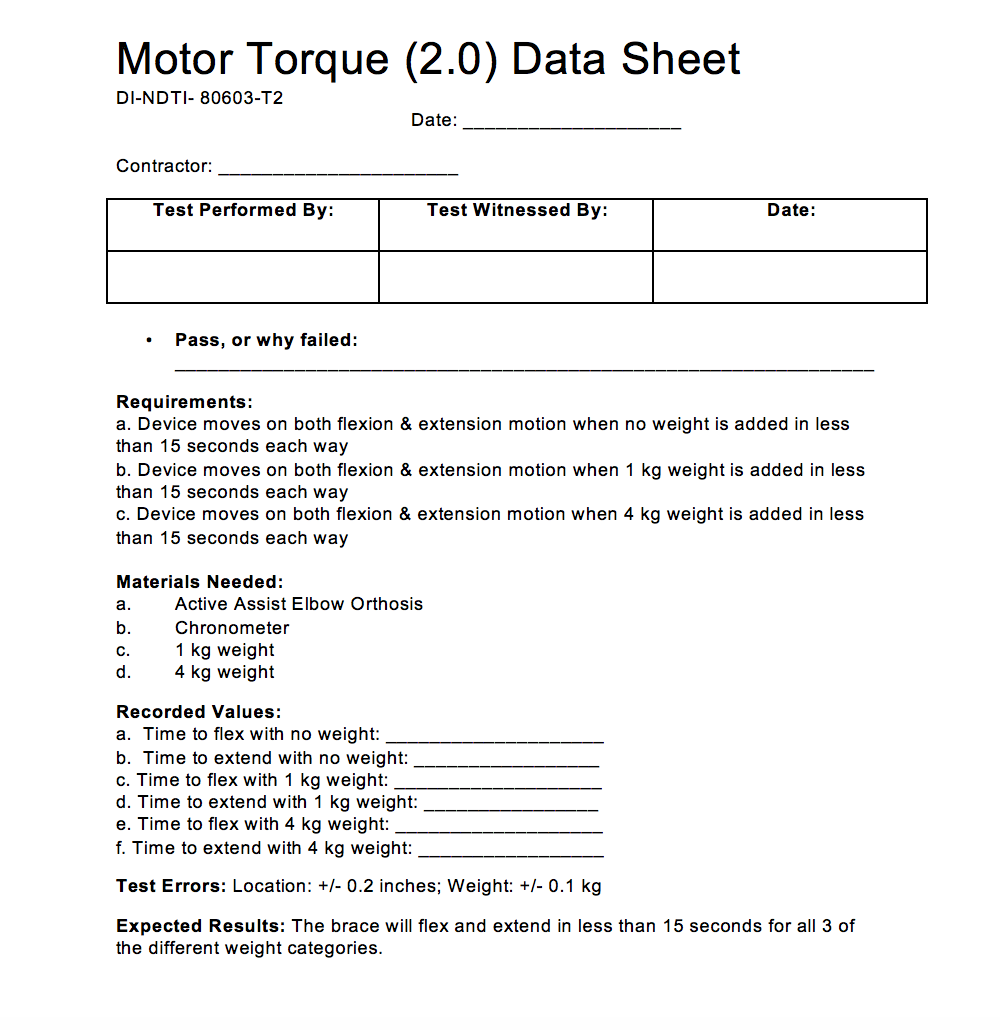
Test Description: This purpose of this test is to verify that the motor is able to actively assist the patient in movement as the patient exhibits a certain amount of resistance. The weights will be used to mimic this resistance, and the motor should flex and extend in less than 15 seconds.

Procedure:

1. The upper section of the AEO device shall be fixed in a vertical position.
2. With iOS app, motor shall be instructed to perform flexion movement, from around 0° to 130°
3. With iOS app, motor shall be instructed to perform extension movement, from around 130° to 0°
4. A chronometer will be used to measure the time it takes for the device to move from 0˚ to 130˚, which should be less than 15 seconds. The chronometer will be used to measure the time it takes the device to move from around 130˚ to 0˚, which should also be under 15 seconds.
5. Step 2 and 3 shall be repeated by adding no weight, followed by 1 kg weight (+/- 0.1 kg) added 6 inches from the hinge on the forearm attachment, and 4 kg weight (+/- 0.1 kg) attached 6 inches (+/- 0.2 inches) from the hinge on the forearm attachment.
6. The chronometer will be used to measure these times the device will flex and extend, which will be under 15 seconds each way.

Diagram:





**4.3 Test Procedure Verification and Data Sheet: Position Tolerance (3.0)**

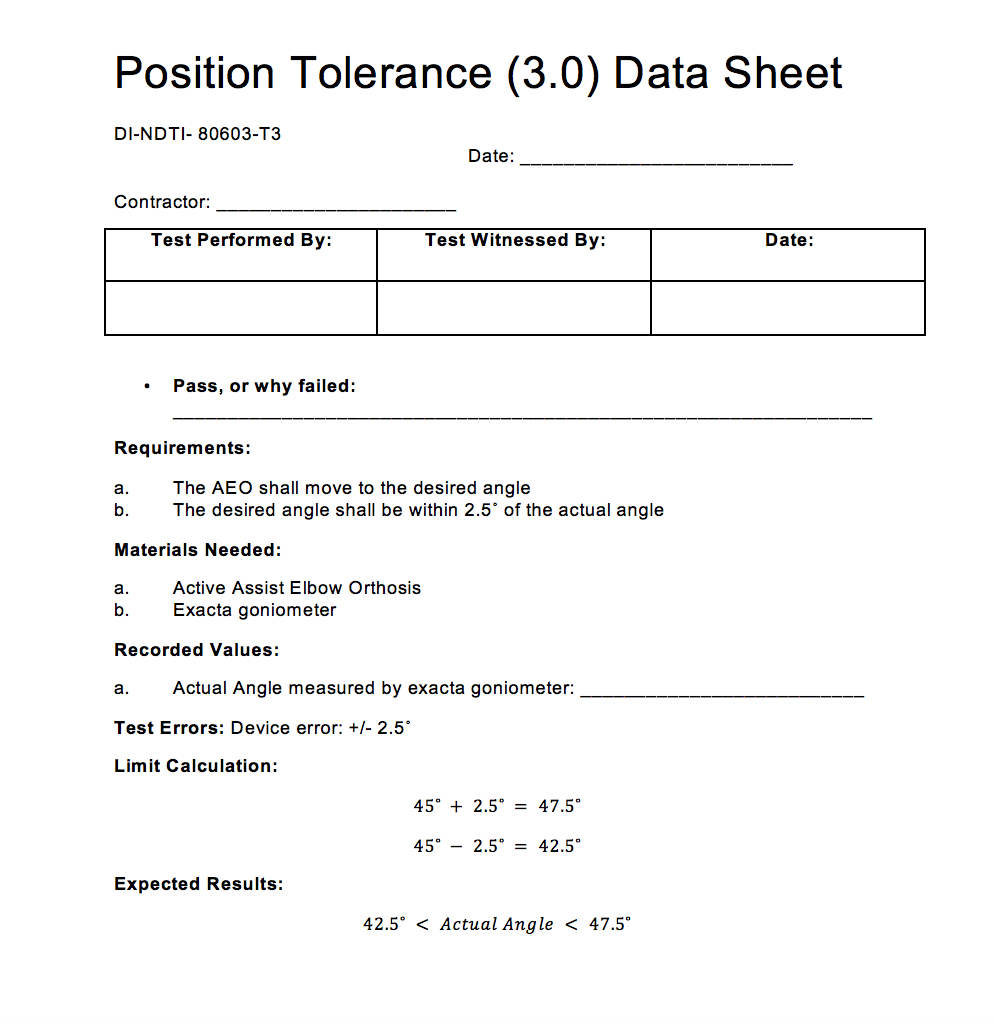
Test Description: The purpose of this test is to verify that the device moves to the correct position that it is instructed to via the iOS application. The position of the brace is necessary to further verify that the sensor works correctly.

Procedure:

1. The device shall be securely worn on a healthy individual, and powered on via iOS app
2. The subject will instruct the device to move to desired angle (45˚) via iOS app
3. Once the device stops at the instructed angle, an exacta goniometer will be used to measure the angle
4. Add collected data to the data sheet
5. If the angle measured is within 2.5˚ of the desired angle, the device passes the test

Diagram:



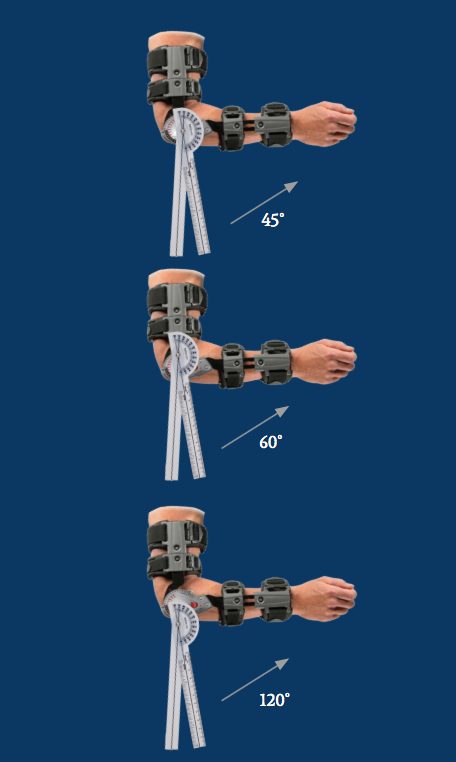


**4.4 Test Procedure Verification and Data Sheet: AEO Sensor (4.0)**

Test Description: The purpose of this test is to ensure that the AEO sensor works at a variety of angles, and reads within 2.5˚ of the desired angle.

Procedure:

1. The device shall be securely worn on a healthy individual, and powered on via iOS app
2. The subject will instruct the device to move to desired angle (45˚)
3. Once the device stops at the instructed angle, an exacta goniometer will be used to measure the angle
4. The angle will be recorded on the data sheet
5. The subject will instruct the device to move to desired angle (60˚)
6. Steps 3 and 4 are repeated
7. The subject will instruct the device to move to desired angle (120˚)
8. Steps 3 and 4 are repeated
9. If the angle measured is within 2.5˚ of the programmed angle, the device passes the test

Diagram:



**4.5 Test Procedure Verification and Data Sheet: Angular Velocity (5.0)**

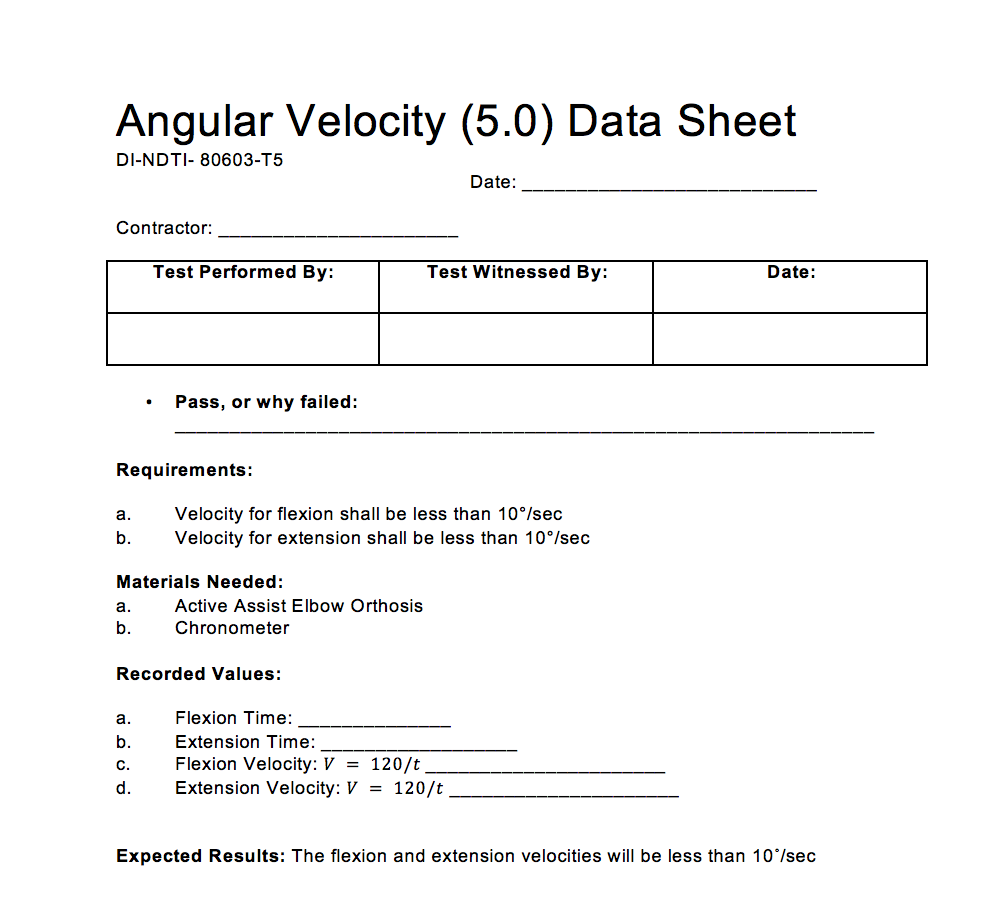
Test Description: The purpose of this test is to verify that the flexion and extension movement has a velocity of less than 10˚ per second.

Procedure:

1. The device shall be securely worn on a healthy individual, and powered on via iOS app
2. The AEO device shall be commanded by the app to move from around 0° to 120°. The patient shall not exert any external force.
3. With the use of a chronometer, the time shall be measured accurately.
4. Step 2 and 3 shall be performed now extending the arm, from 120° to 0°.
5. Equation to calculate the velocity is V=120/t , t being time in seconds.

Diagram:





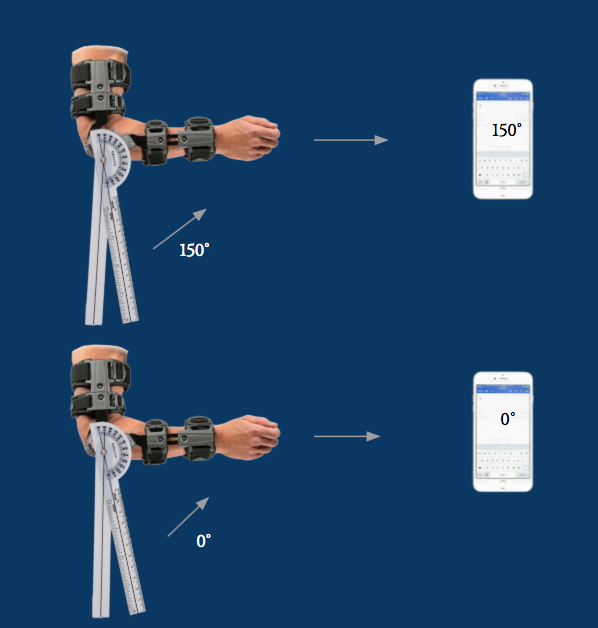
**4.6 Test Procedure Verification and Data Sheet: Sensor App Accuracy (6.0)**

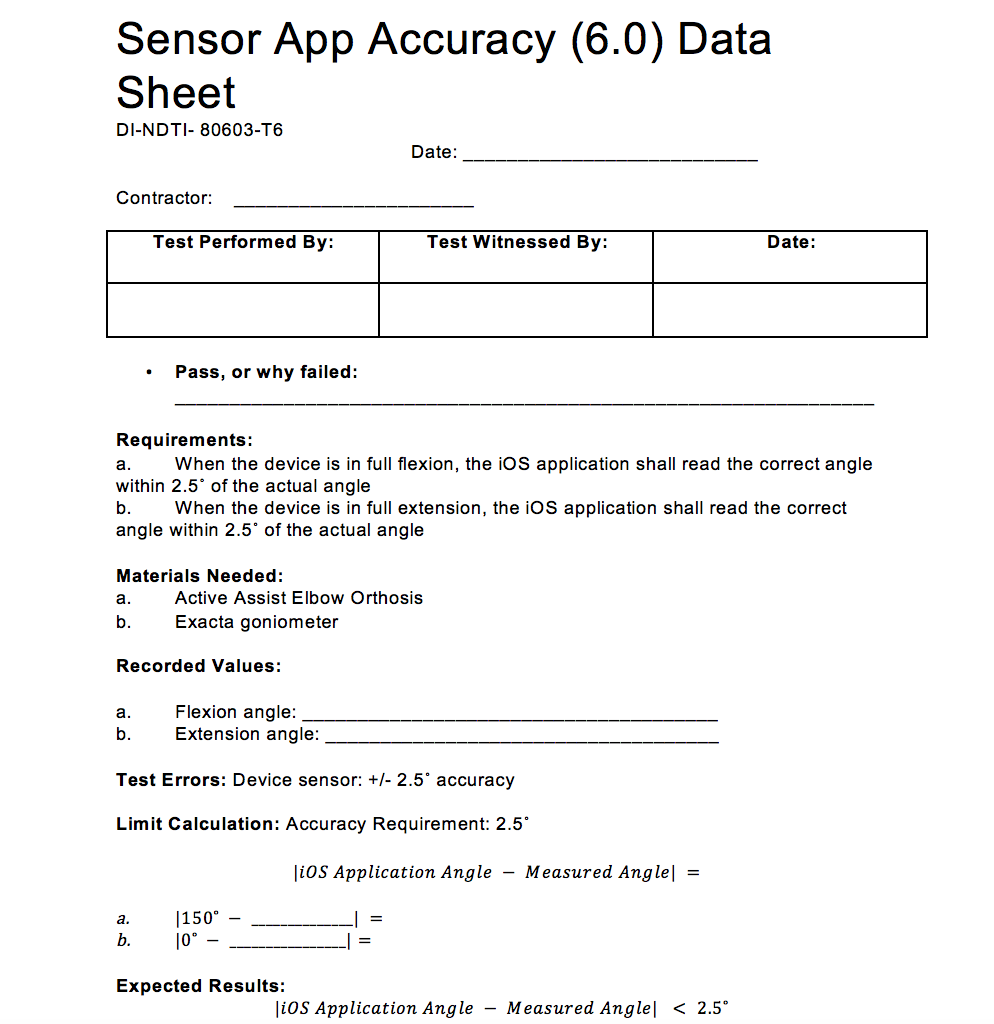
Test Description: The purpose of this test is to verify that the sensor on the AEO device will measure the angle of the elbow within 2.5˚ of the actual angle. It is necessary to test that the measurements at full extension and full flexion are within 2.5˚ of the actual angle to ensure the device is capable of accurately moving within a healthy individual’s full range of motion.

Procedure:

1. A healthy individual will use the brace for angle measurements at **maximum extension** and **maximum flexion**.
2. The individual test subject will fully extend their elbow while wearing the AEO
3. iOS application shall display the angle of extension within 2.5˚ of the actual angle
4. Exacta goniometer shall measure the angle to verify the app reads within 2.5˚ of the actual angle
5. Record sensor accuracy on data sheet
6. Test subject will fully flex their elbow while wearing the AEO
7. iOS application shall display the angle of flexion within 2.5˚ of the actual angle
8. Exacta goniometer shall measure the angle to verify the app reads within 2.5˚ of the actual angle
9. Record sensor accuracy on data sheet

Diagram:





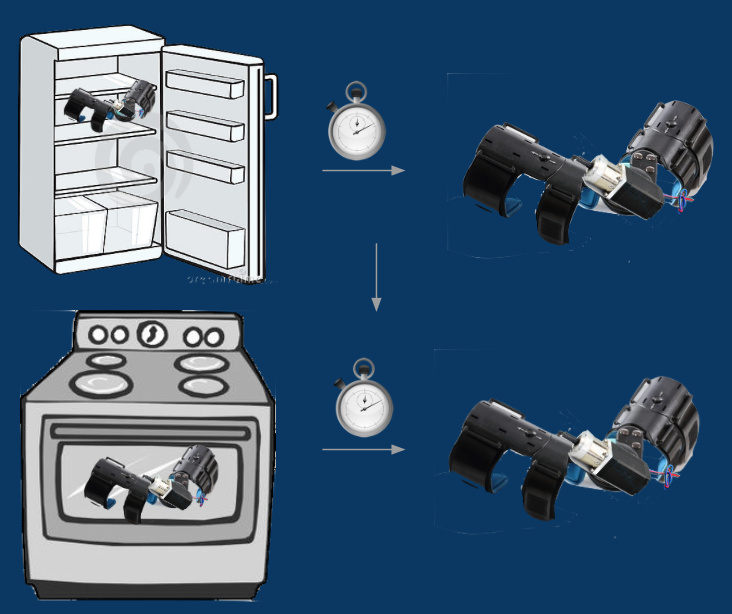
**4.7 Test Procedure Verification and Data Sheet: Temperature Range (7.0)**

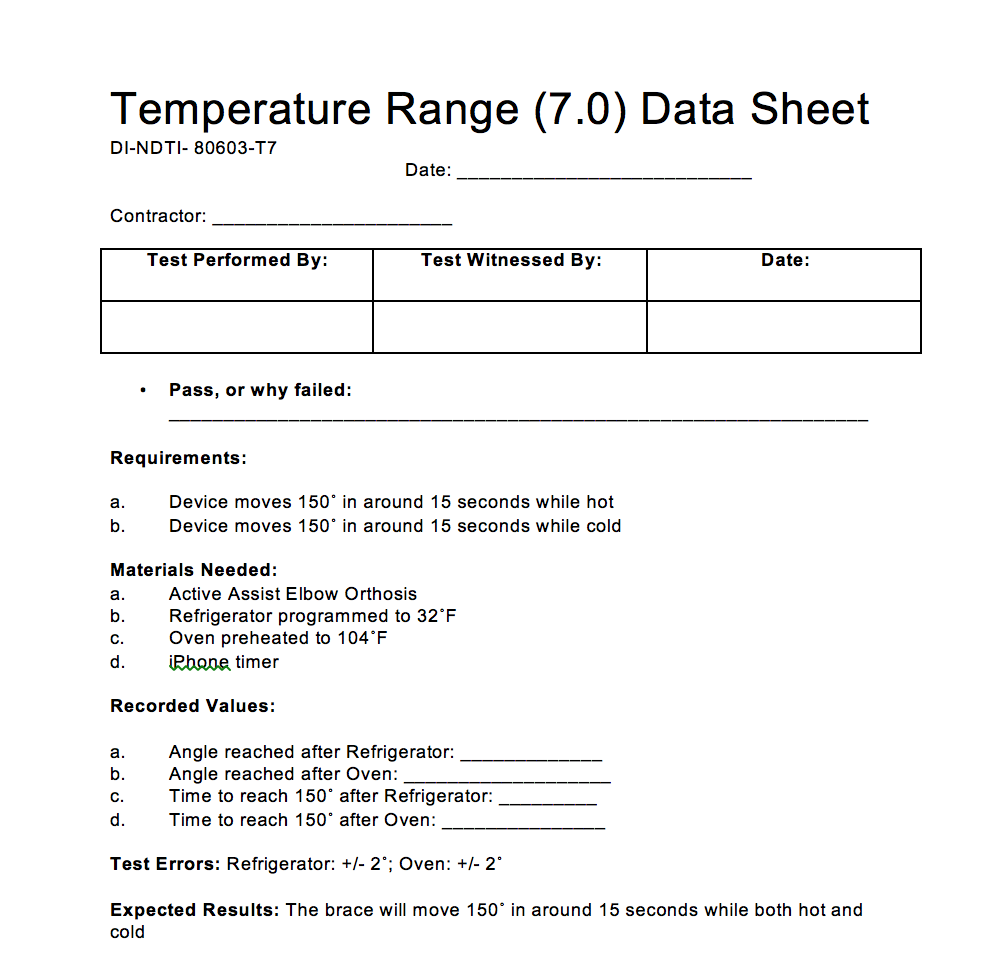
Test Description: The purpose of this test is to verify that the entire AEO device can withstand extreme temperatures and still operate a full range of motion in less than 15 seconds.

Procedure:

1. Place the AEO inside a preprogrammed refrigerator set to 32˚F (0˚C)
2. Set a timer for 30 minutes
3. Remove the AEO from the refrigerator and begin testing within 5 minutes of removal
4. Power on the AEO using the iOS application, and program the AEO to move the full 150˚ range of motion in less than 15 seconds.
5. Let the AEO come to room temperature
6. Place the AEO inside a preheated oven set at 104˚F (40˚C)
7. Set a timer for 30 minutes
8. Remove the AEO from the oven and begin testing within 5 minutes of removal
9. Repeat Step 4
10. Record the pass/fail results in the data sheet

Diagram:





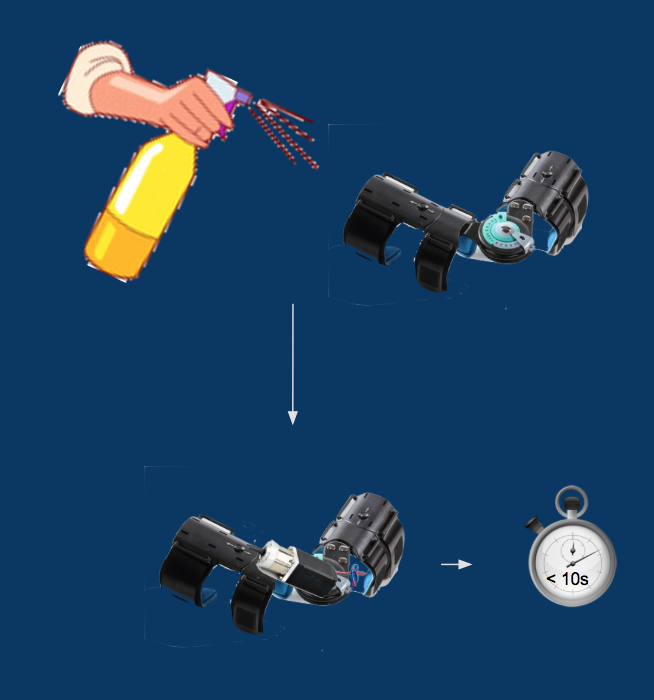
**4.8 Test Procedure Verification and Data Sheet: Weather (8.0)**

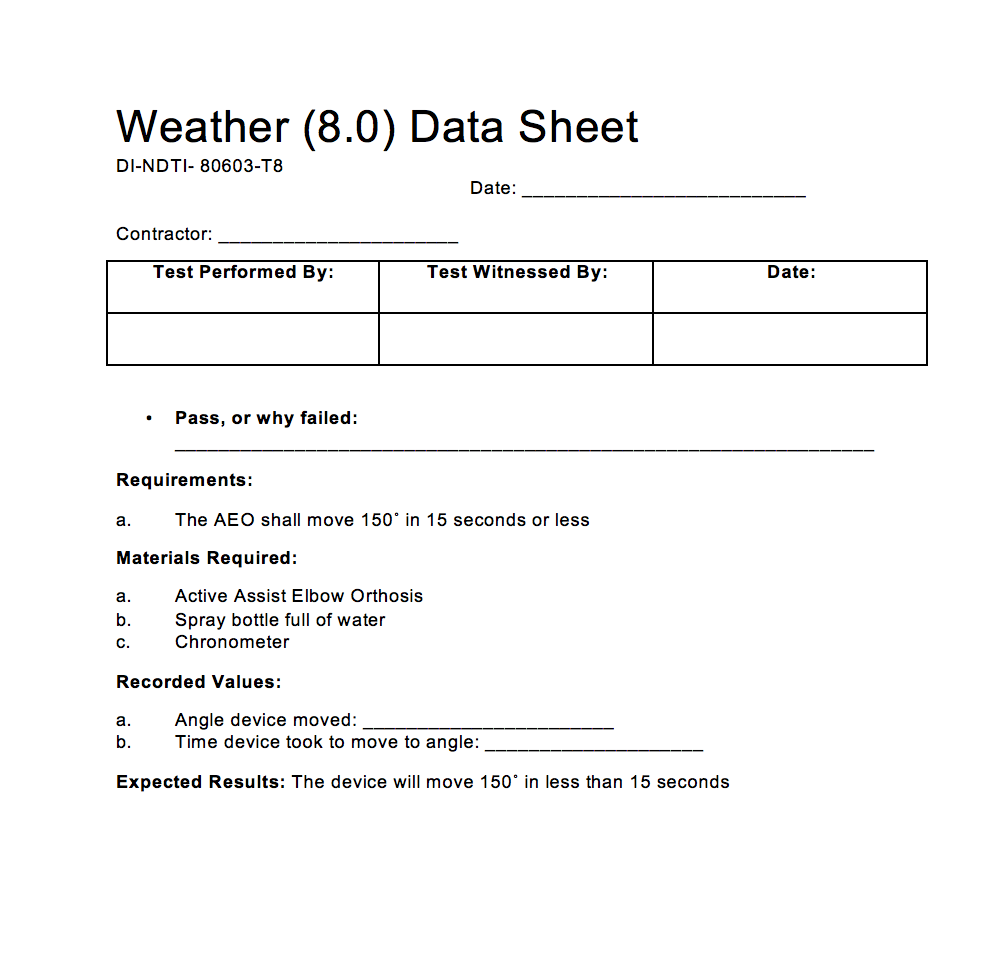
Test Description: The purpose of this test is to mimic a light rainy condition and verify that the AEO device can operate under such weather conditions.

Procedure:

1. Fill a squirt bottle with water
2. Lay the AEO on an empty, flat surface
3. Make sure casing is attached with motor not visible to water
4. Spray the device 5 times beginning at one end of the AEO, and slowly spraying to the other end of the AEO
5. Power on the AEO using the iOS application, and program it to move the full 150˚ range of motion in 15 seconds or less
6. Use chronometer to measure the time of the movement
7. Record the pass/fail results of the AEO moving with the water droplets mimicking light rain into the data sheet

Diagram:





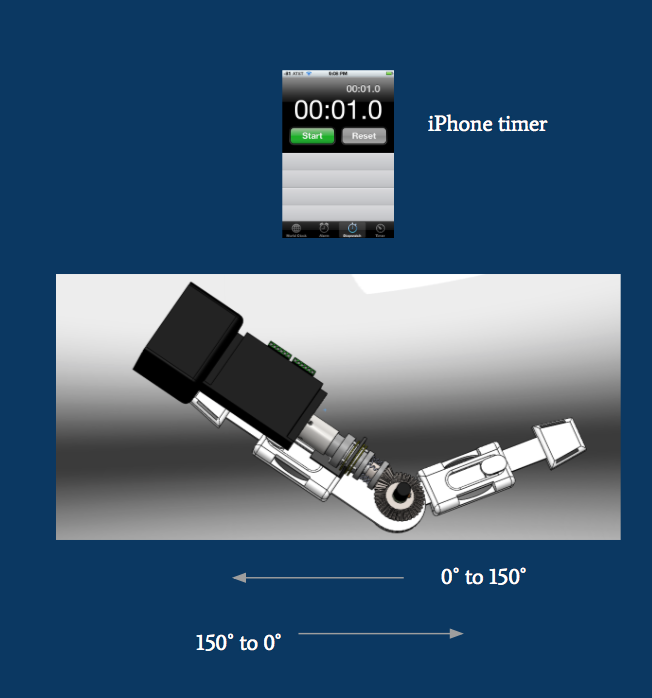
**4.9 Test Procedure Verification and Data Sheet: Battery Life (9.0)**

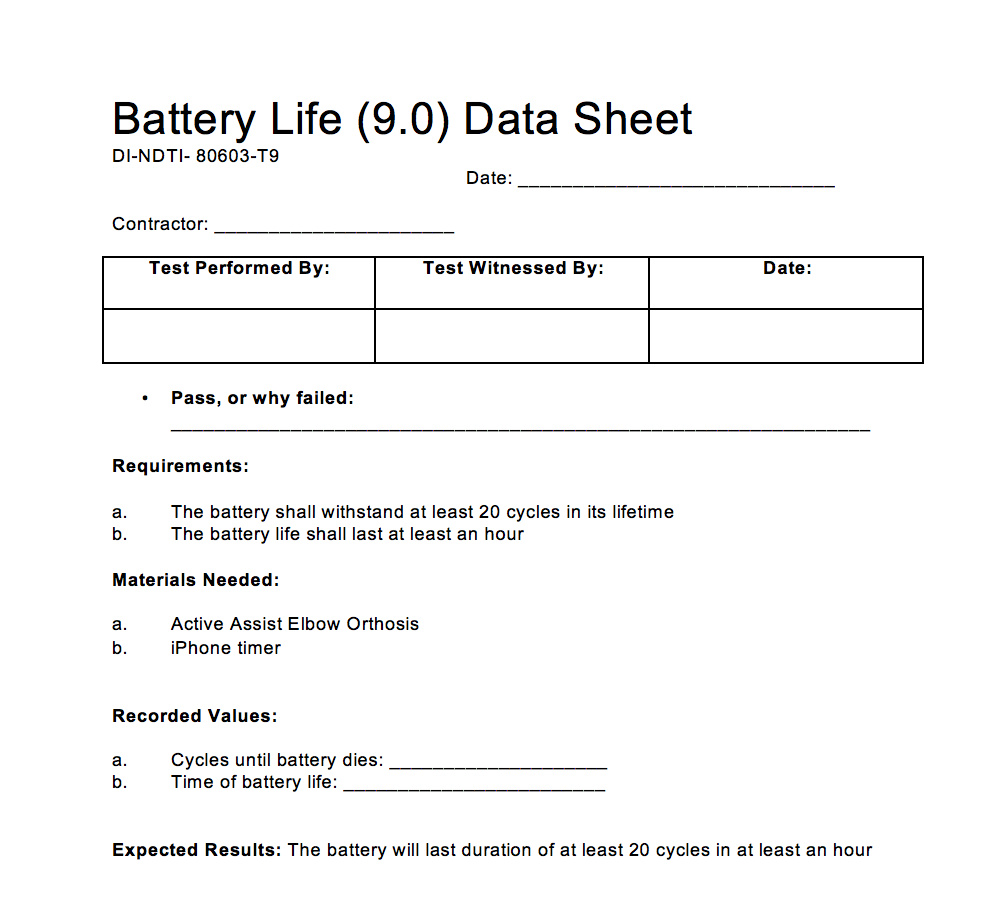
Test Description: The purpose of this test is to ensure that the AEO device can operate at least 20 cycles before the battery dies. The user should be able to perform at least two full exercises during the lifetime of the battery.

Procedure:

1. Attach the 19.2V battery, motor, and casing unit onto the brace, and power the device on via the iOS application
2. Program the device to flex from 0˚ to 150˚ and to extend from 150˚ to 0˚
3. Perform this movement as many times until the battery is completely dead
4. Use an iPhone timer to determine the duration of battery life. Begin the timer as soon as the first flexion movement begins. Battery life should last at least an hour
5. The device shall flex and extend to complete one cycle, and a total of 20 cycles should be completed before the battery dies
6. Record the number of complete cycles that occur on the data sheet
7. If the device completes at least 20 cycles in the battery life, it passes the test

Diagram:





**4.10 Test Procedure Verification and Data Sheet: Battery Recharge (10.0)**

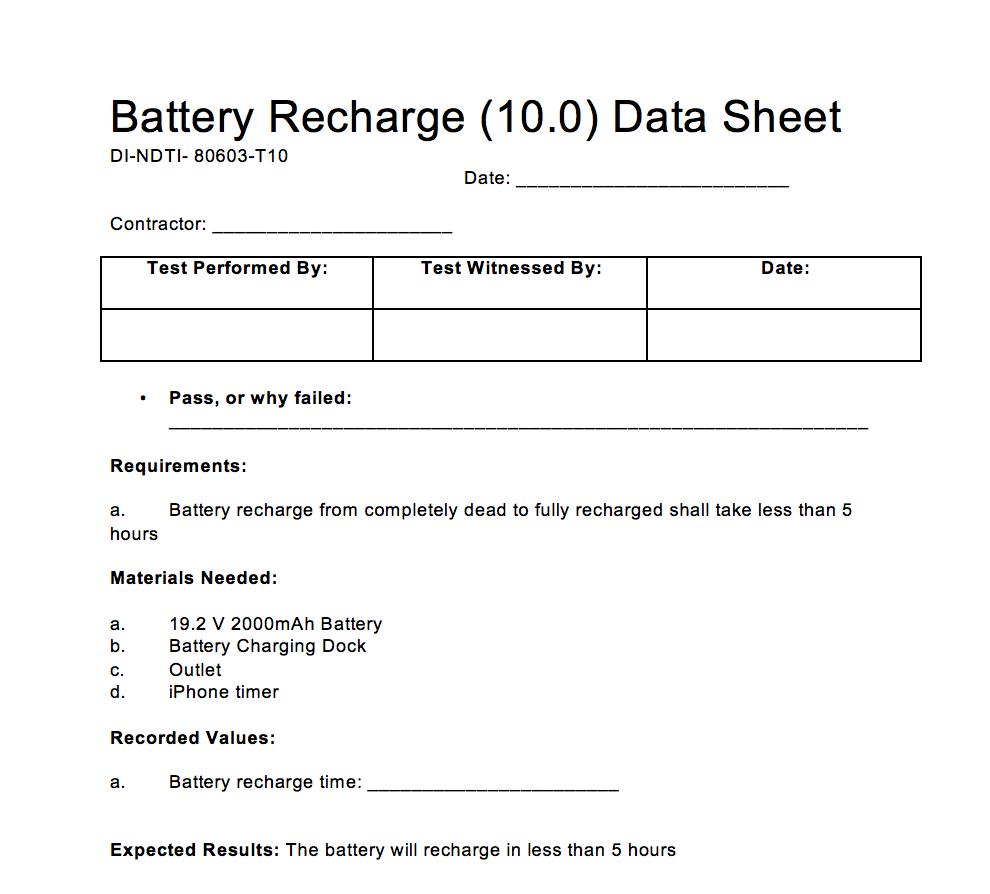
Test Description: The purpose of this test is to verify that the time it takes the battery to recharge is less than 5 hours. This way, the user will be able to perform his/her exercises multiple times a day.

Procedure:

1. Run the 19.2V battery until it is completely dead.
2. Plug the battery into the charging dock in the nearest outlet
3. Using an iPhone timer, time the duration it takes for the battery to be fully recharged
4. Battery recharge time should take less than 5 hours for the device to pass.
5. Record time on the data sheet

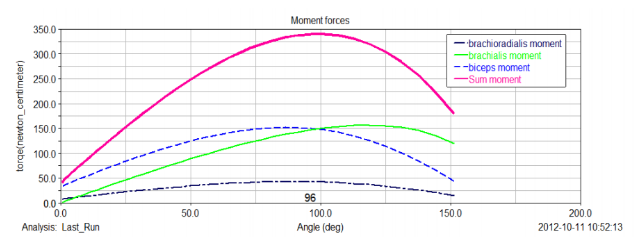
Diagram:





**5.0 Models and Analyses**

**5.1 Torque Model**



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **θ** | 25 | 50 | 96 | 125 |
| **𝝉 (N-cm)** | 150 | 250 | 345 | 300 |

The torque model was used to analyze the amount of torque exerted by the arm muscles at each angle of flexion from 0 to 150 degrees. It was found that there was a maximal torque of 345 N-cm exerted at 96 degrees. Since we do not want the This model helped us determine how much torque our brace should exert. We do not want the device to exert the full amount, but rather, somewhere around half the maximum torque. The information gathered was key in helping us to pick out the motor.

**5.2 Motor Power Model**

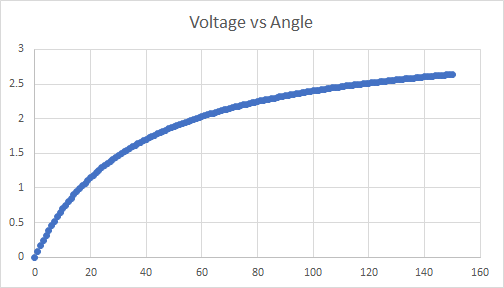
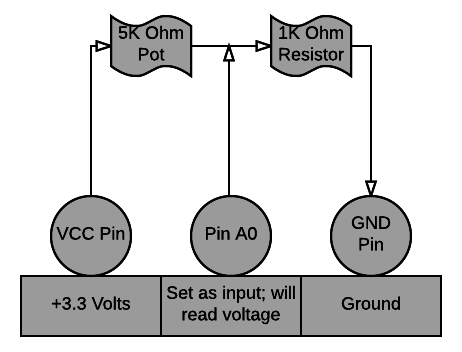
Operating time = (time of one full flexion) \* (20 cycles)   
 = (15 seconds) \*20 = 300 seconds

Battery capacity = current \* operating time = 0.67 Amps (max) \* 300 seconds  
 = 55.8 mAmp-Hrs

The battery capacity needed for the motor operation was calculated by first finding the operating time and then multiplying that time by the current needed to run the motor. The operating time was determined as the time required to complete 20 cycles of full 150 degree flexion at 10 degrees per second. The maximum current that can be run into the motor is 0.67 amps. The battery capacity required for one exercise (20 cycles) is 55.8 mAmp-Hrs.

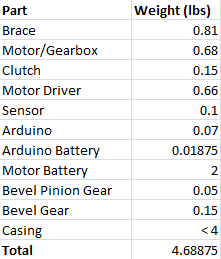
**5.3 Sensor Model**





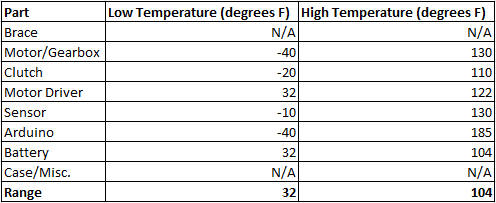
The relationship between the voltage read by pin A0 of the Arduino MKR1000 is the voltage between the 5K ohm potentiometer and the 1K ohm resistor. That voltage depends on the resistance of the potentiometer. The resistance of the potentiometer is dependent on the turning of its knob. The resistance of the potentiometer at 0 degrees is 0 ohms and the resistance of the potentiometer at 150 degrees is 4000 ohms. Because the elbow brace joint and the potentiometer are connected, the relationship between elbow angle and potentiometer resistance is: R = 26.6667 . The circuit is set-up as a voltage divider. The relationship between the voltage at the node between the resistor and potentiometer and the resistance of the potentiometer is: V = 3.3 R / (1000 + R). Substituting the relationship between potentiometer resistance and elbow angle. he relationship between the voltage at the node between the resistor and potentiometer and the elbow angle is: V = 88(.

**5.4 Weight Analysis**



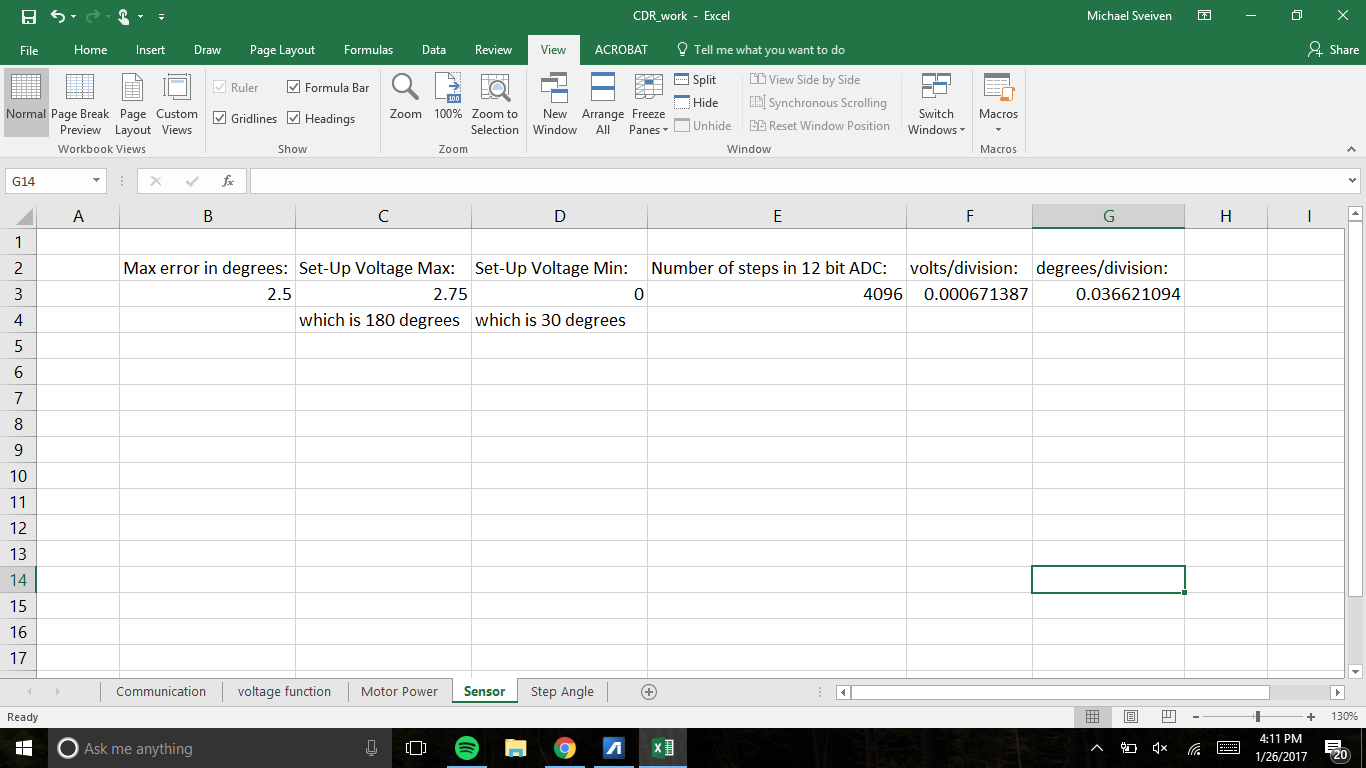
This is the weight analysis for the AEO device. It is broken-down into the weight of each part utilizing excel as a tool. Here it is analyzed and verified that the AEO device, entailing all its individual parts and casing, will weigh less than ten pounds.

**5.5 Temperature Analysis**



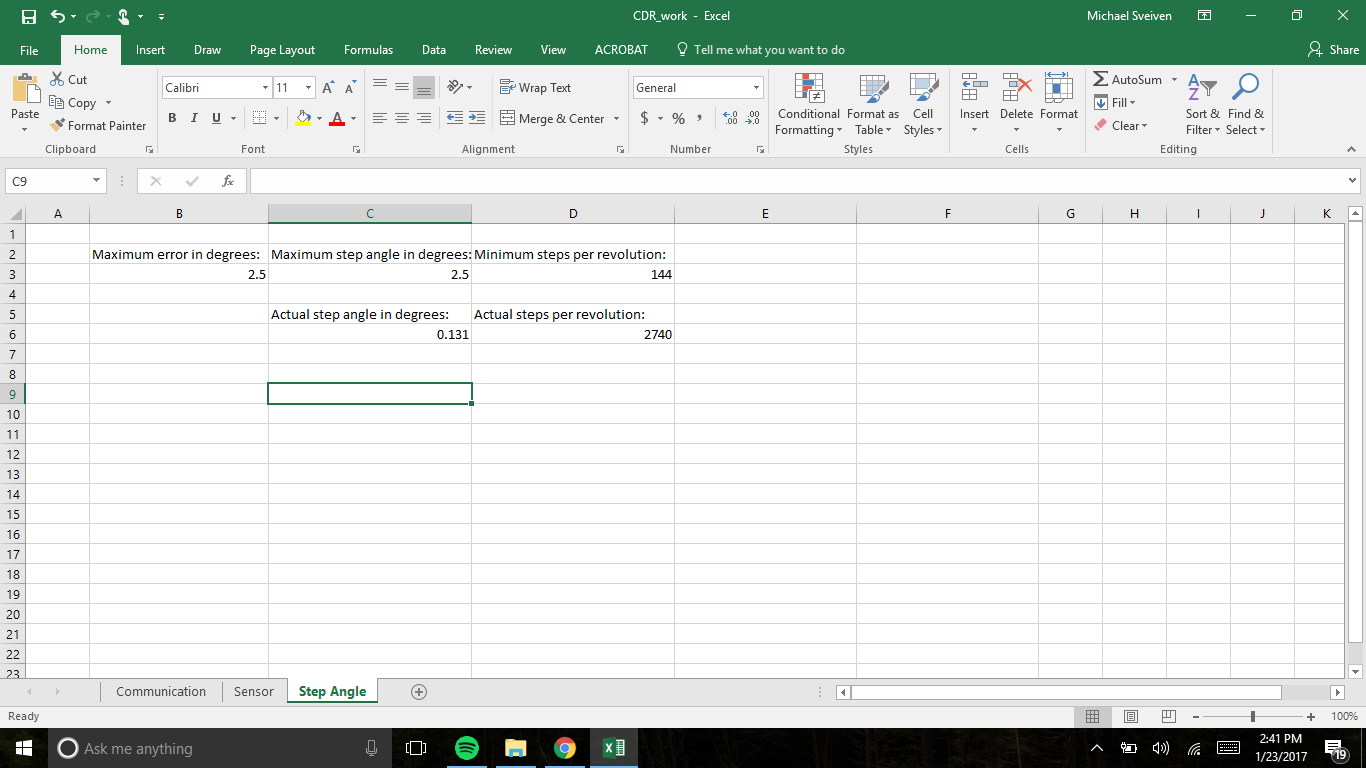
This is the temperature analysis for the AEO device. Excel is used as a tool to analyze the lowest and highest operating temperatures for each individual part. This analysis allows us to determine at which temperatures the AEO device shall be operational. According to this, the battery is the limiting part when measuring temperature due to the fact that it has the highest low operating temperature and the lowest high operating temperature. In result, the range of operating temperature is from 32 degrees Fahrenheit to 104 degrees Fahrenheit.

**5.6 Sensor Analysis**



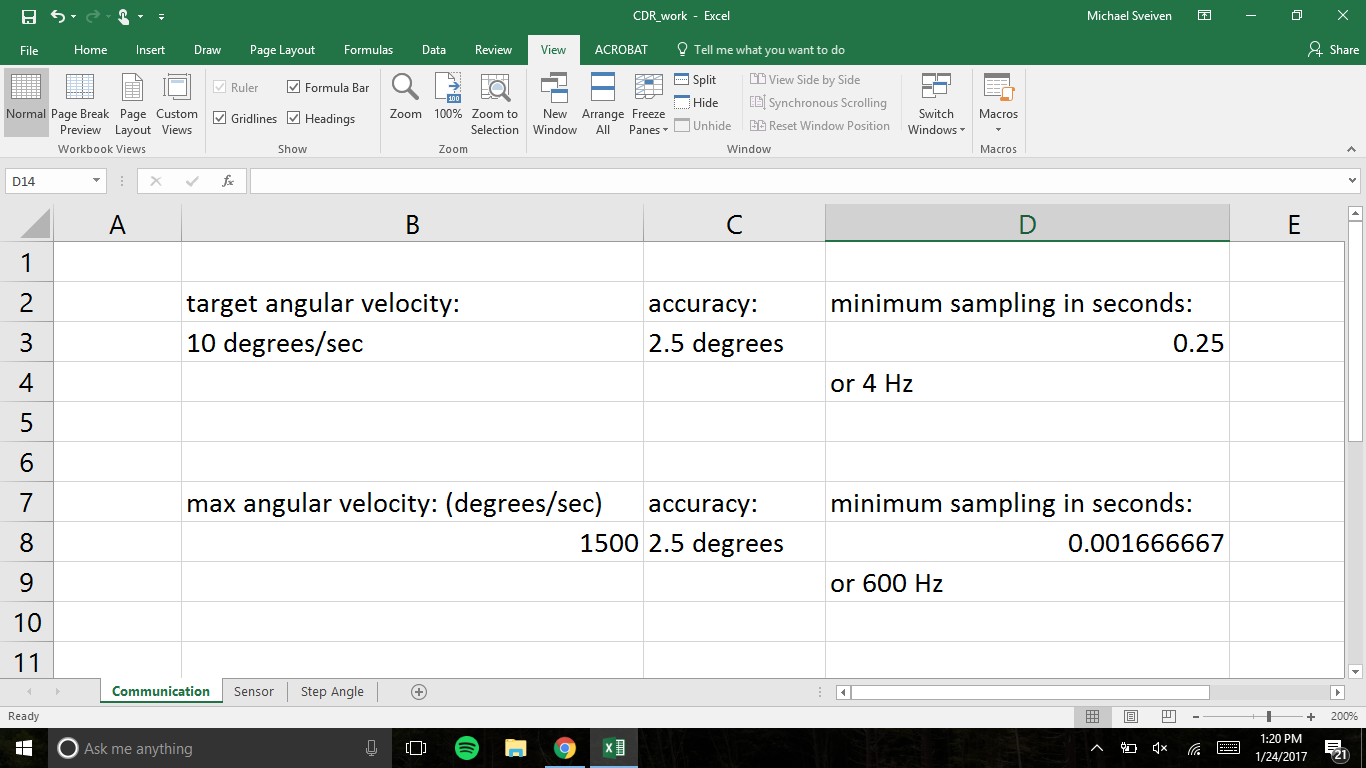
The maximum voltage to be read at the A0 pin of the Arduino MKR1000 is 2.75 V and the minimum is 0 V. The maximum error in degrees is 2.5. The smallest degrees/division is that 2.5 degrees. Because of the 12 bit ADC of pin A0, the degrees per division is 0.0366, which is much better than necessary.

**5.7 Step Angle Analysis**



The step angle analysis was simple. The maximum error for the AEO motor movement is 2.5 degrees, therefore, the largest step size for our motor would be 2.5 degrees per step. The NEMA 11 stepper motor that we chose has a 0.131 step angle, which is much better than the 2.5 step angle required.

**5.8 Sampling Analysis**



The target angular velocity for the AEO is 10 degrees per second. At 10 degrees per second angular velocity, the sensor would have to sample every 0.25 seconds in order to stay within that 2.5 degree accuracy required. A 4 Hz sampling rate is required at 10 degrees per second. The likely maximum angular velocity that the AEO would exhibit is 1500 degrees per second. In order to stay within that 2.5 degree accuracy, the sensor would need to sample every 0.001667 seconds, which is 600 Hz. According to the [www.arduino.cc](http://www.arduino.cc) site, the sampling rate of the Arduino MKR1000 is 10,000 Hz, which is well above what is necessary.

**6.0 Risk and Cost Analysis**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Likelihood | Consequences | | | | |
|  | Insignificant  (easily handled) | Minor (some  disruption) | Moderate (significant time/resources required) | Major (project  severely damaged) | Catastrophic  (project destroyed) |
| Almost certain (>90%) | High | High | Extreme | Extreme | Extreme |
| Likely (50%-90%) | Moderate | High 4 | High | Extreme | Extreme |
| Moderate  (10%-50%) | Low | Moderate | High 3, 5 | Extreme 1, 2 | Extreme |
| Unlikely (3%-10%) | Low | Low | Moderate | High | Extreme |
| Rare (<3%) | Low | Low | Moderate | High | High |

1. Motor and battery placement 4. Brace sizing
2. Battery overheat 5. Weight of brace
3. Emergency shut off mechanism

**6.1 Mitigation Plan**

1. Motor and battery placement- Construct a cover that will protect motor, battery, and any wires that will be exposed on the active elbow orthosis device, especially from water. Verify placement does not restrict any movement of arm. If unacceptable, attach motor and battery in a different position on the brace..
2. Battery overheat- Buy rechargeable lithium-ion battery and attach to motor and active

orthosis device. Run the prototype iOS application that commands the battery powered motor on and running smoothly. If unacceptable, replace battery. Make sure to place in area with air circulation. Rerun prototype.

1. Emergency Shut Off- Connect the shut off mechanism to the active orthosis device. Run the prototype test and verify that once the emergency shut off is used, the device is free from any active assist and in free motion controlled by the patient. If unacceptable, replace shut off mechanism and rerun prototype.
2. Brace Sizing- Obtain the prototype brace that is adjustable in both the arm and forearm area. Place it on an adult and adjust to fit their arm length. Place it on a child and adjust to fit their arm length. If unacceptable, add secondary adjustment mechanism.
3. Weight of brace- Obtain the prototype brace and attach the motor and battery in its full

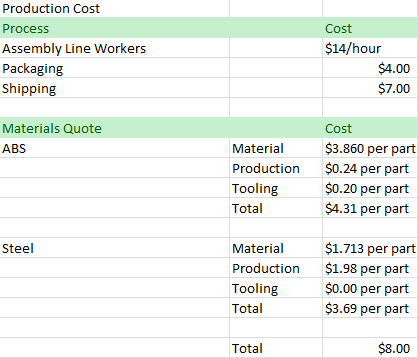
weight. Place on the patient to verify no overbearing weight or shoulder strain. If unacceptable, remove battery and motor from device and place in new area on the body that is not the arm.

**6.2 Cost Analysis**

There were an estimated 150,000 elbow surgeries in US in 2006. In response, estimated number of AEO devices to be produced in the span of the first 6 months would be 10,000 to observe sales and competition. After this production will be adjusted accordingly. These estimates were made in theory and are not permanent.

The following is an estimation of the cost of the AEO prototype and assembly cost.





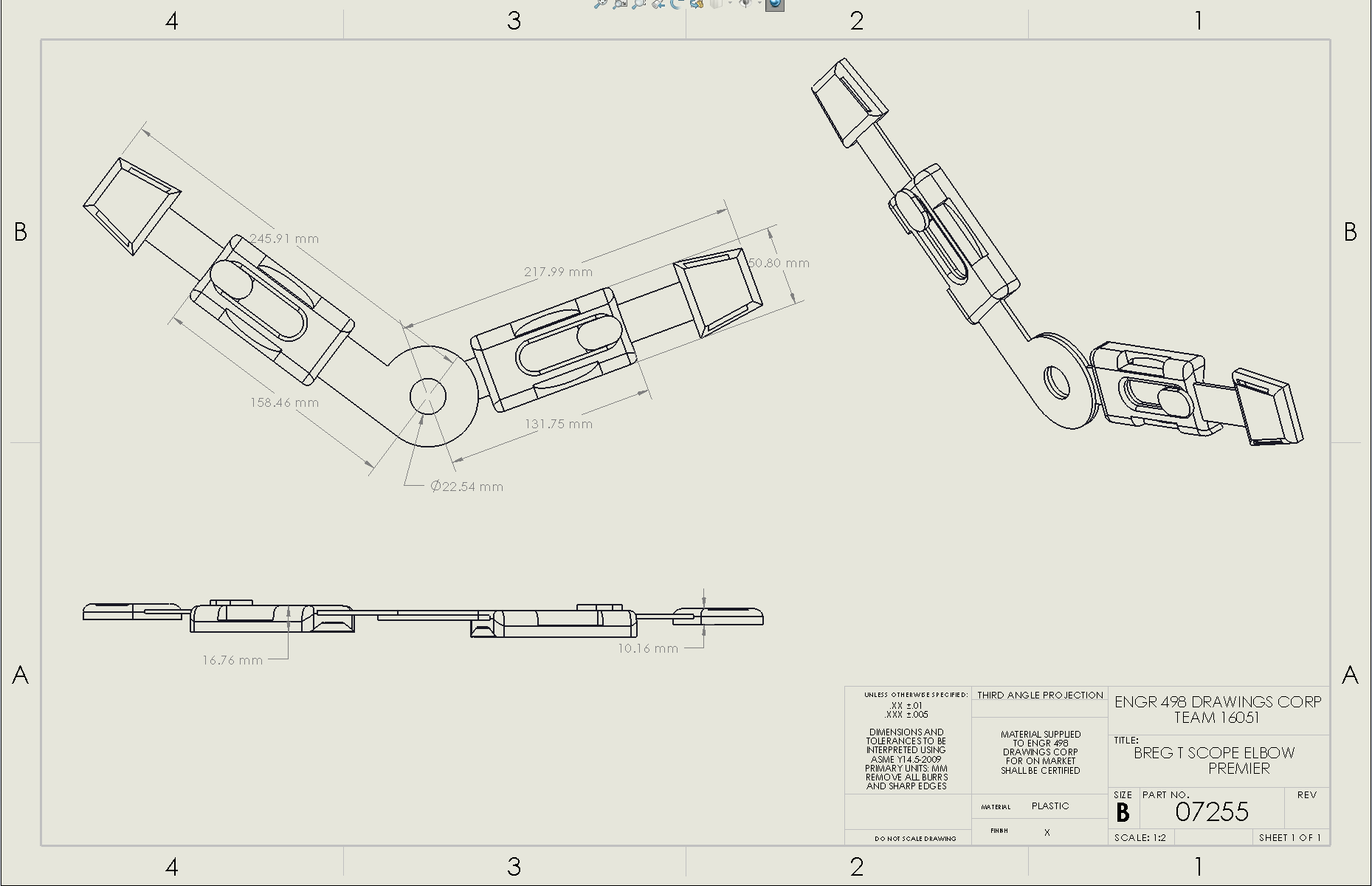
Current braces in the market with no active assist range from $300 to $600. In the future our portable device would be competitive at a price ranging from $700 to $1000 with the downloadable iOS application free of cost. At the moment, the prototype cost would be valued around $400.

The following is a summary of the costs spent thus far.

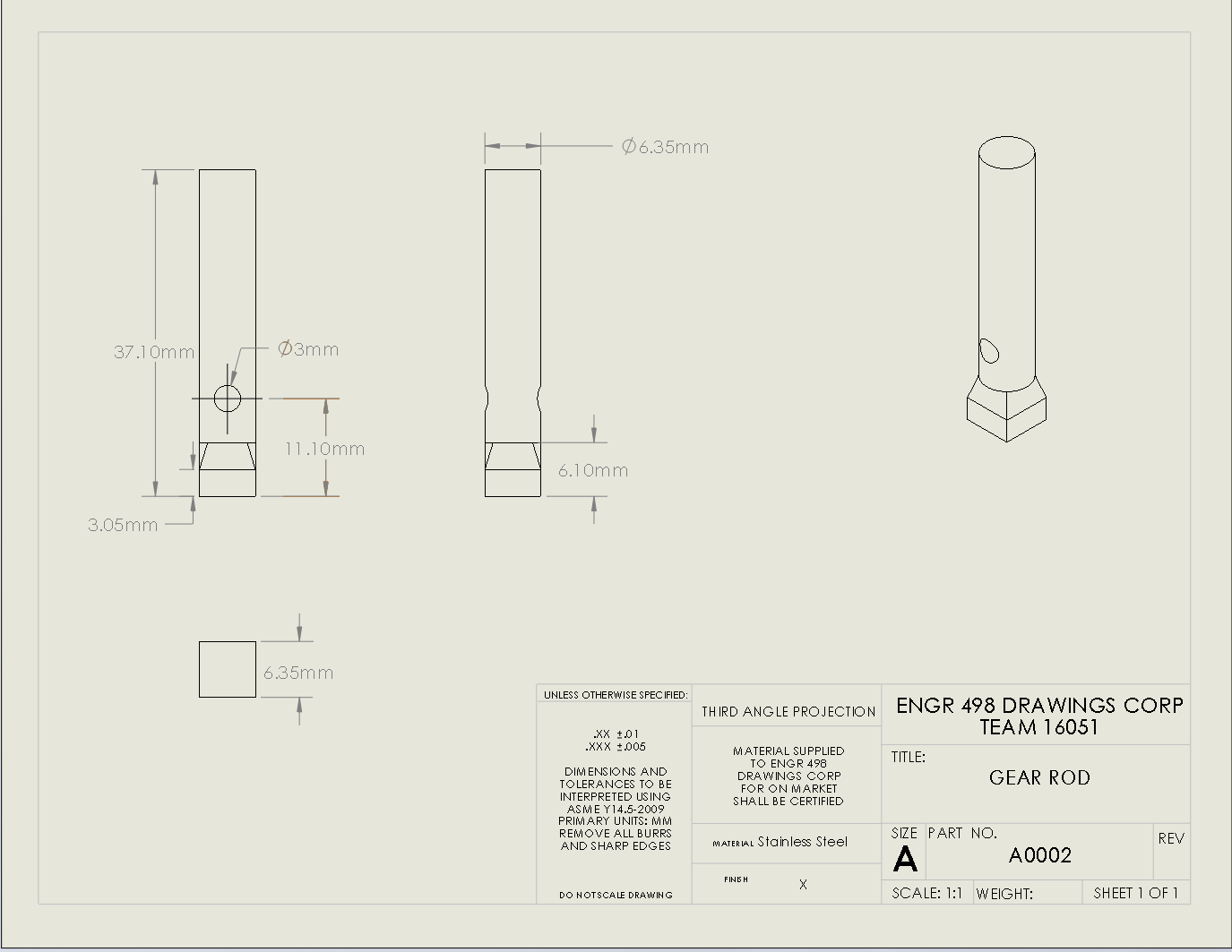
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Item | Cost Ea. | Qty. | Supplier | Lead Time | When Needed |
| Arduino | $34.99 | 1 | Arduino LCC/ Amazon | Obtained | 2/2 |
| Potentiometers | $7.71 | 5 pc. | Amazon | Obtained | 2/2 |
| Stepper Motor (including gear box) | $34.68 | 1 | StepperOnline Motors and Electronics | Obtained | 2/2 |
| Arduino Battery | $16.43 | 1 | Amazon | 1 week | 2/2 |
| Motor Battery | $22.99 | 1 | Amazon | 1 week | 2/2 |
| Brace | Free | 1 | Sponsor | Obtained | 2/2 |
| Stepper Motor Driver | $32.17 | 1 | StepperOnline Motors and Electronics | 5 days | 2/2 |
| Slipper Clutch | $24.99 | 1 | Dynatect | 1 week | 2/2 |
| Boston Gear Spiral Bevel Gear | $14.76 | 1 | Amazon | 1 week | 2/2 |
| Boston Gear Spiral Bevel Pinion Gear | $10.98 | 1 | Amazon | 1 week | 2/2 |
| Casing | $5.35 | 1 | UA Machine Shop | 2-3 weeks | 3/1 |
| iOS Application Student Consultant | ~$1000 | - | - | - | - |
| Team Shirts | $50 | 6 | Aztec Embroidery | 2 weeks | 3/10 |
| Design Day Poster | $20 | 1 | University of Arizona | 2 weeks | 4/20 |
| Total | $1,525.05 | - | - | - | - |
| Misc. | ~10% | - | - | - | - |
| Extended Cost | $1,677.55 | - | - | - | - |

**7.0 Appendix**

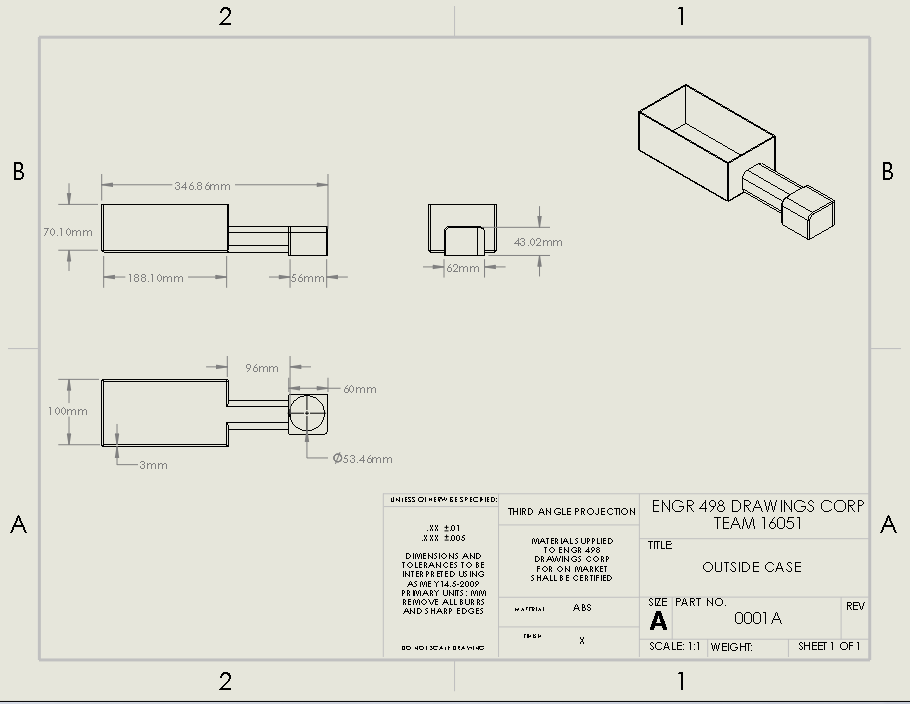
**7.1 Detailed Parts and Assembly Drawing**



**Figure 1.** Drawing of elbow brace following ASME Y14.5 -2009 guidelines



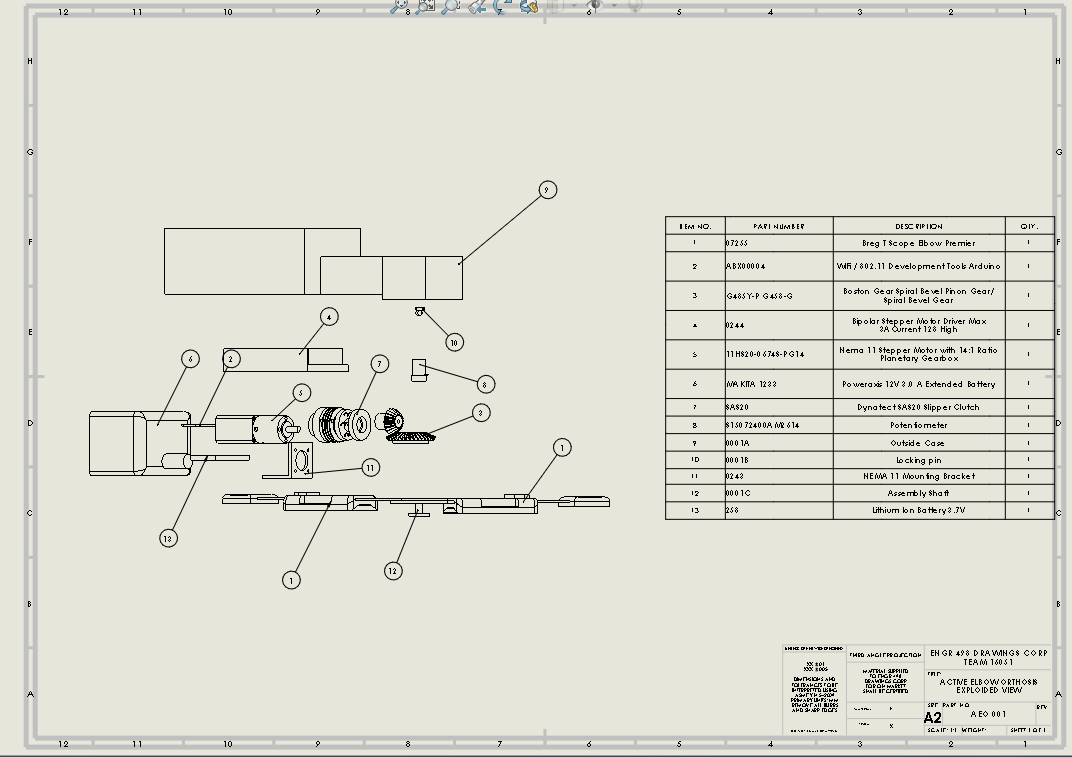
**Figure 2.** Drawing of stainless steel shaft that needs to be build to aid the stability of the gear system, following ASME Y14.5 -2009 guidelines



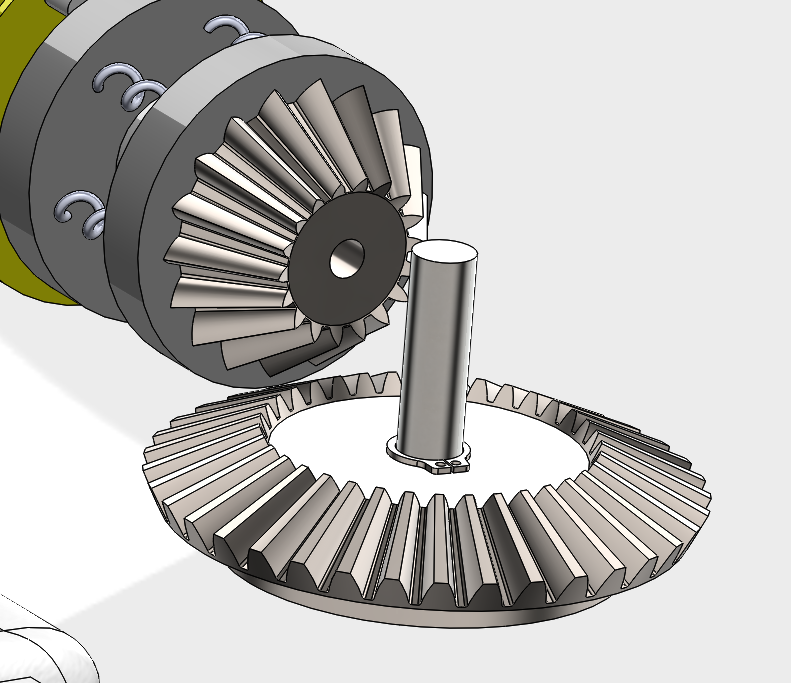
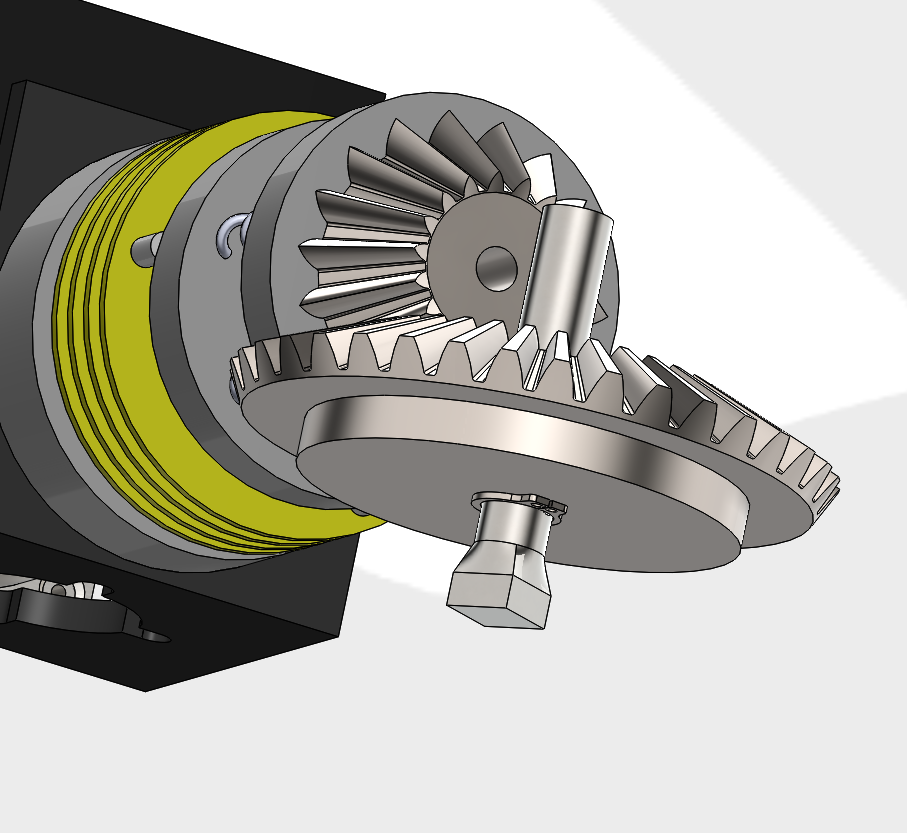
**Figure 3.** Drawing of external envelop to protect and stabilize all the pieces for the device, following ASME Y14.5 -2009 guidelines



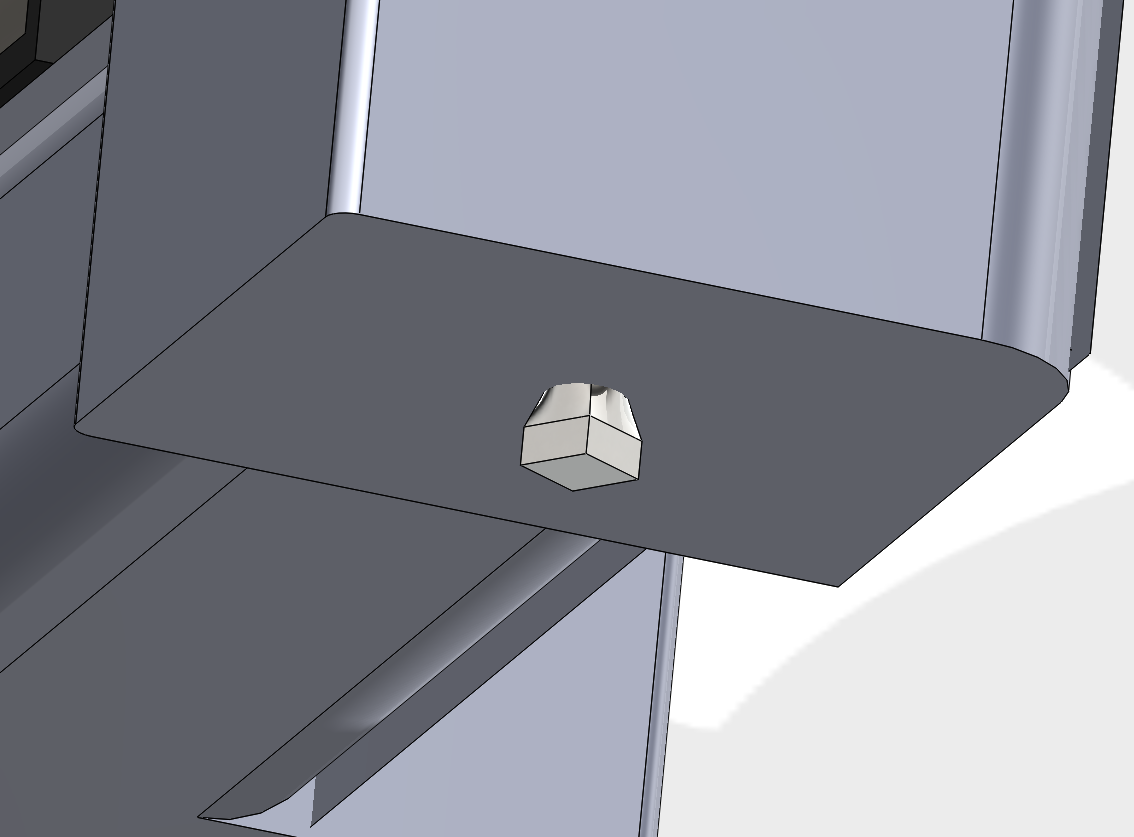
**Figure 4.** Drawing of device with external case attached, following ASME Y14.5 -2009 guidelines.



**Figure 5.** Drawing of AEO exploded view, following ASME Y14.5 -2009 guidelines.



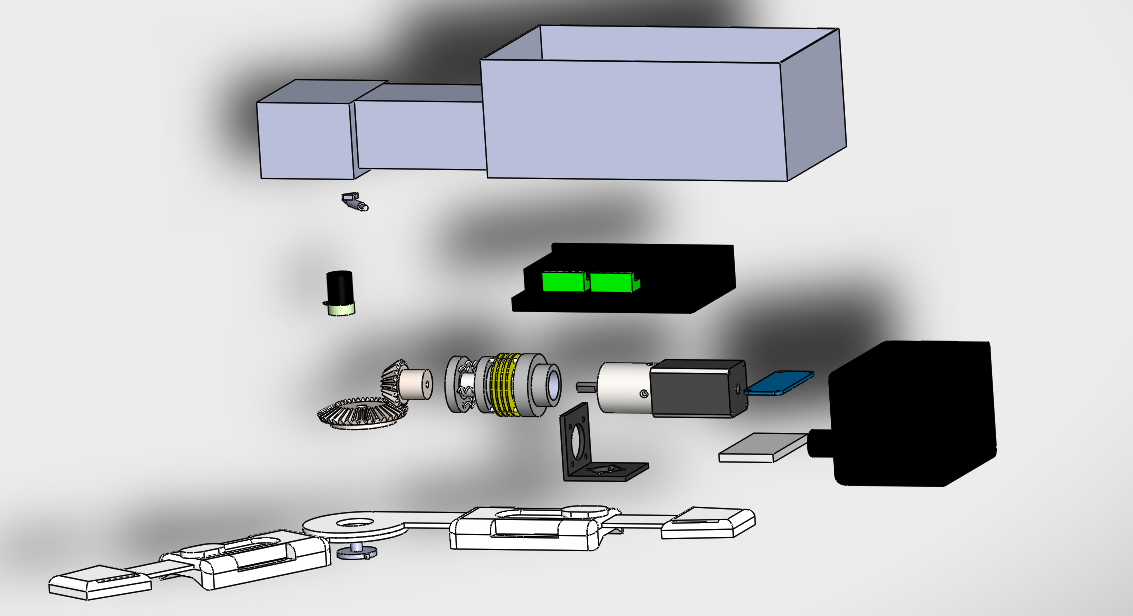
**Figure 6.** Assembly of AEO exploded view, following ASME Y14.5 -2009 guidelines.



**Figure 7.** Assembly of join point bottom view, this square shaped shaft allows the join between the brace and motor mechanism allowing for the rotation of only the forearm extension.



**Figure 8.** Assembly of the entire AEO device, enclosing all the parts inside the ABS case.



**Figure 9.** Assembly of AEO exploded view, showing all parts. List of the parts include, NEMA 11 stepper motor, potentiometer, clutch SAS20, arduino, pinion bevel gear, bevel gear, motor driver, shaft for gear stability, batteries (main and arduino), brace, motor mounting bracket, arduino mounting bracket, emergency stop, and case.

**7.2 Wireless Application Instruction Document**

