RECOVERY-MONITORING KNEE BRACE

By

Dennis Ryu [dryu3]

Dong Hyun Lee [dlee134]

Jong Yoon Lee [jlee642]

Final Report for ECE 445, Senior Design, Fall2017

TA: Dongwei Shi [dshi9]

13 December 2017

Project No. 40

Abstract

The Recovery-Monitoring Knee brace has three main features. It notifies the user about maintaining a secure brace fit through multiple pressure sensors, helps the user refrain from excessive leg use by showing an EMG signal on the calves, and monitors the injury directly by measuring the temperature of the swelling of the knee compared to its vicinity. These features can be accessed both on the computer and mobile device.

Contents

1. Introduction

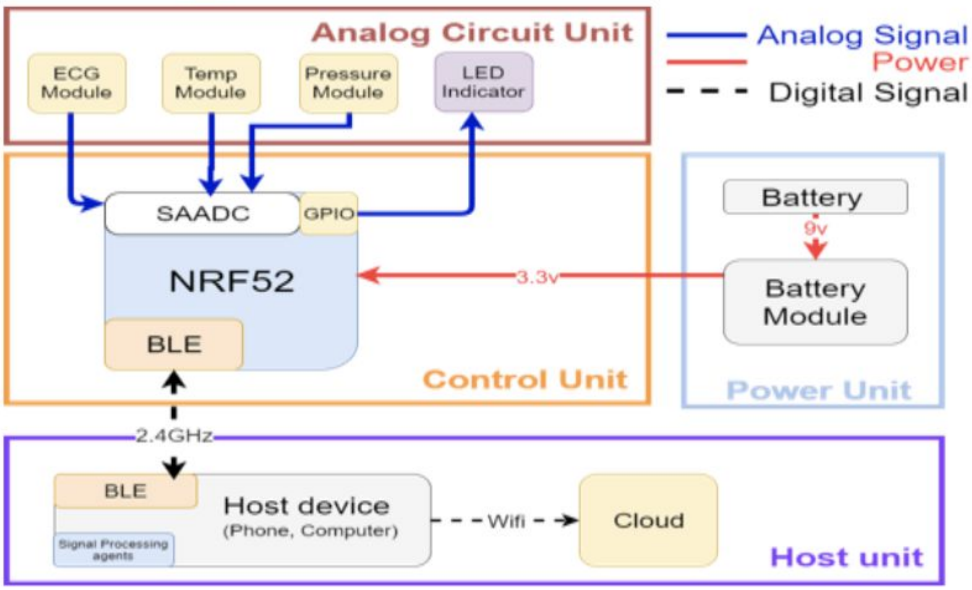
Thanks to modern technology, it is easy to encounter a wide variety of wearable fitness devices such as Fitbit and Apple Watch in the market. Such devices are designed for average consumers who wish to track their lifestyle by counting steps or measuring heartbeats. But it is rare to find a product for the actual patients who require both the real-time monitoring of a wearable device and the hard protection of a brace.

Personally, one of our teammates ruptured his front knee ACL and received reconstruction surgery a few years ago. After ACL surgery, it is common to wear a knee brace for about two to three months for protection from outside impacts, fast recovery, and restriction of movement. For a patient who is situated in rehabilitation after surgery, knee protection is an imperative recovery stage, but is often overlooked. One cannot deny that a brace is also cumbersome to put on in the first place. There should be a viable alternative for such people in need.

Our prototype is displayed in Appendix B.

2. Design Procedure

The Recovery-Monitoring Knee Brace uses a knee brace that is commonly used for knee-injured patients. The basic frame is a lightweight metal with multiple straps that go around the patient's knee. There are three straps on each side of the brace with an angle control dial at the middle. The high-level design of our project is shown in Figure 1.



ECG: Electrocardiography
SAADC: Successive Approximation A/D Converter
GPIO: General-purposeinput/output
NRF52: Name of our microcontroller
BLE: Bluetooth Low Energy

Figure 1 Block diagram of our project.

We collect array of bio-signals from the **Analog Circuit Unit** by utilizing the ADC peripheral on the NRF52 SoC. The **Control Unit** then orchestrates the conversion of the analog signal and the BLE data transfer to the Host device. **Power Unit** provides the necessary power to the **Control Unit** and the **Analog Circuit Unit.** The streamed data from the NRF52 chip is received from the **Host Unit** and signal processing is done via various methods.

2.1 Control Unit

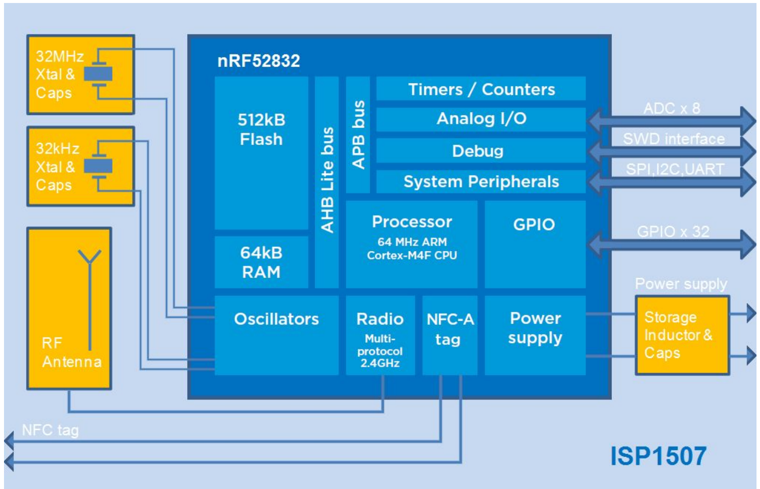


Figure 2 High-level diagram of nRF52 microcontroller.

We will be taking advantage of the nRF52 for the purposes of our project. It includes an Analog I/O, GPIO, timer, and Bluetooth communication module, which are all required. A block diagram of the microcontroller is displayed in Figure 2.

2.2 Electromyography Sensor

Electromyography (EMG) is a signal that measures the the muscle activity utilizing electrodes. Figure 3 is the full pipeline of the circuit

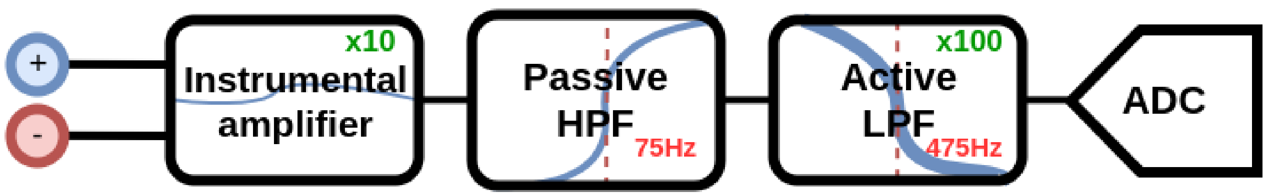


Figure 3 High-level diagram of nRF52 microcontroller.

The purpose of the EMG sensor is to measure the degree of muscle activity by the patient and give feedback on whether it was under or over the safe threshold. Figure 4 displays our circuit. We used the passive high pass filter and active low pass filter to cut out the unnecessary frequency ranges. Equation 1 was used to calculate the cutoff frequency in our filter. Throughout the multiple try out, we found out that the optimal frequency range for leg was 75Hz to 475Hz. In addition to that, the signal gain was 1000 from the initial signals from the leg. First gain is from the INA333 and second gain is from the active high pass filter. The INA333 gain is calculated with equation 2 and our inverse operational amplifier's gain is calculated with equation 3.

$$f_c = \frac{1}{2\pi RC} \tag{1}$$

$$G_1 = 1 + \frac{100k\Omega}{R_1} \tag{2}$$

$$G_2 = -\frac{R_2}{R_1} \tag{3}$$

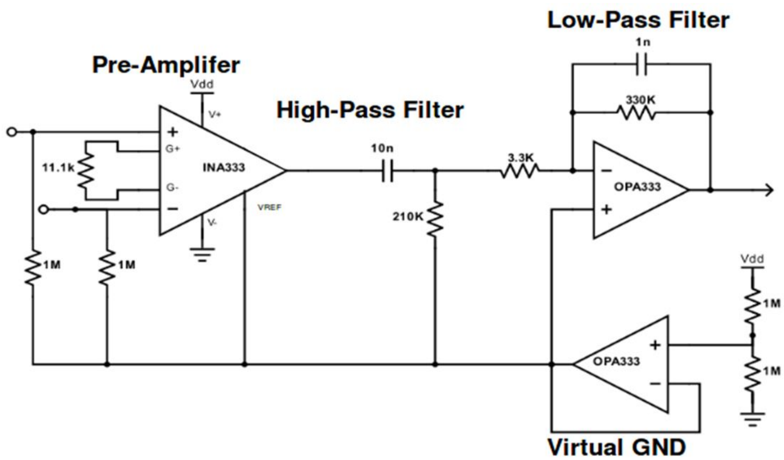


Figure 4 Schematic of EMG circuit

We capture three different types of muscle activities from our EMG signal. Strong muscle activity such as running, walking up the stairs, or stamping, weak muscle activity such as leaning, and no muscle activity.

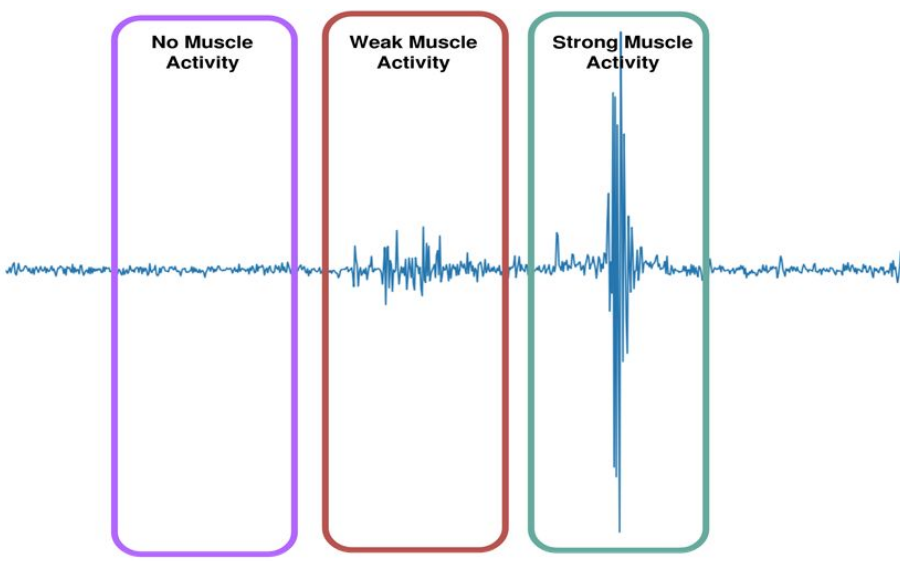


Figure 5 The Host User Interface on PC tablet

This was done through a neural network. The network consists of 3 layers 4 in width. The network's input is 200 data points. 200 points was large enough to capture all the features while small enough to run the algorithm in real-time. The network was trained with labeled data that we generated by simulating the three different muscle activities we want.

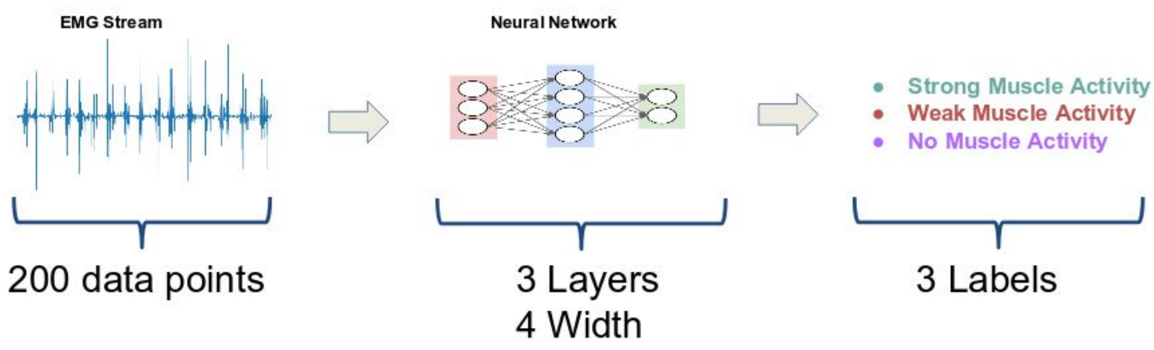


Figure 6 The Host User Interface on PC tablet

We also implemented a lightweight version of the EMG signal processing code and can be seen in Appendix D.

2.3 Temperature/Pressure Sensor

The temperature sensor for the device is to show the user that their knee is swollen up or not. Our device uses two different temperature point, one on the knee and one on the reference point to compare the temperature. (Appendix C shows temperature sensor location) Our host user interface will show the temperature difference between these two points.

The pressure sensors under the strap of the knee brace will monitor whether the user has properly worn the knee brace or not. The degree of tightness of the strap will be measured through pressure sensors, and this analog signal(voltage) will be fed into the NRF52 to post signal processing for the user.

Both the temperature and pressure sensors operate under the bridge circuit formation, as shown in Figure 7. Both pressure and temperature sensors will be resistor sensors. We are going to use a voltage divider equation using known resistor values to figure out the voltage for each sensor. This voltage output will be fed into the microcontroller to find out the result that we desired for each sensor.

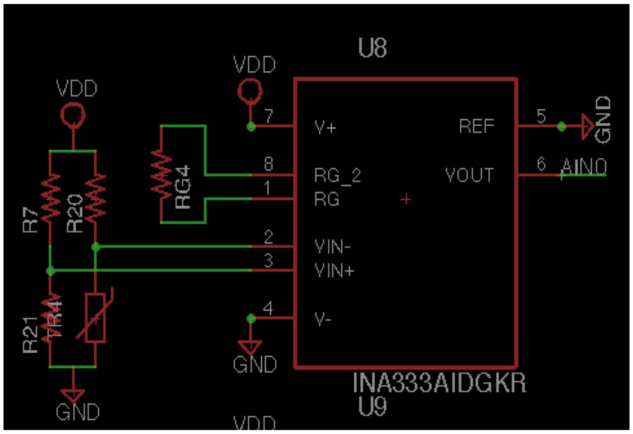


Figure 7 Schematic of Pressure/Temperature circuit

$$V_{out} = V_{in}\left(\frac{R_1}{R_1+R_2} - \frac{R_T}{R_3+R_T}\right) \quad (4)$$

2.4 Status LED Indicator Circuit

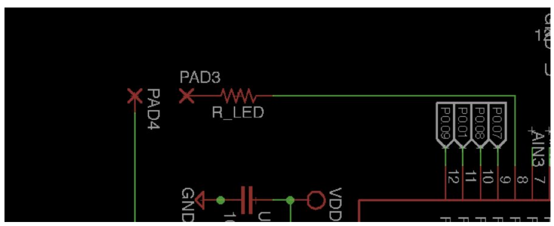


Figure 8 Schematic of LED Indicator circuit

The Status LED indicator, as shown in Figure 8, will be a simple LED circuit with the functionality of displaying whether device is turned on or off. Also, it will indicate that the device is connected into host device via bluetooth or not. The duty cycle will change when the device is connected into the host device. The LED will be connected into the GPIO from the microcontroller and power will be supplied from the power circuit which is 3.3V. Only one resistor (220Ω) will be connected between LED and Microcontroller as nRF52 requires on its reference design.

2.5 Power Circuit

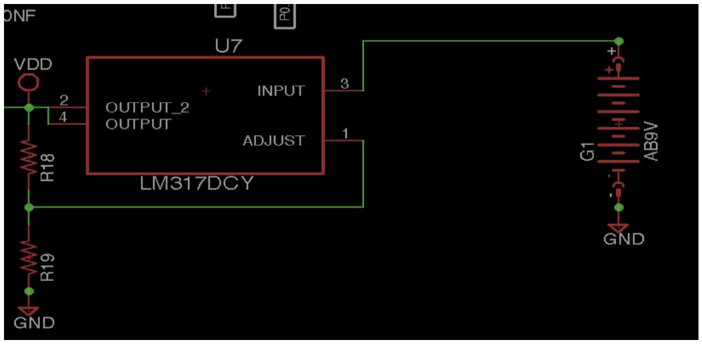


Figure 9 Schematic of Voltage Regulator circuit

The power supply has to provide a steady 3.3 voltage input for the microcontroller and multiple biosignal capturing sensor when the device is turned on. Our main power source will be a 9-volt battery and using a LM317 voltage regulator chip will provide steady 3.3V. Its top-level diagram is shown in Figure 9.

$$V_{out} = V_{ref}(1 + \frac{R_2}{R_1}) \tag{5}$$

In the NRF52, we calculate the voltage level using the 14 bit A/D converter. The ADC has a gain of ⅙ and a reference to internal .6V. Using this information we were able to derive the following equation to get the voltage level in millivolts. The converted battery level is then sent to the host device

$$V_{nrf52} = ADC_{battery} * \frac{V_{internal\_ref}}{ADCResultion} * \frac{1}{Gain} \tag{6}$$

$$V_{nrf52} = ADC_{battery} * \frac{600}{16384} * 6 \tag{7}$$

3. Design Verification

3.1 Control Unit Verification

3.1.1 Current Consumption Verification

Since our knee brace system is a battery powered embedded system, it is important to maintain a low current consumption to ensure a stable long lifetime of the device. We wanted current consumption less than 5mA.

Setup:

In order to measure the current consumption of the system, we use the **Power Profiler Kit**[1] to capture real time current consumption as shown in Figure 10. I measure the current consumption in the **two modes**. First is when the device is connected to the host device and sending data streams.Second is when the device is in not connected and just advertising for a host device. We average five seconds of current measurement as our result.

We also compare different configurations for the chip. We tried the two different power regulation which are **DCDC** and **LDO** and different low frequency clock which are **External Crystal** and **Internal RC**

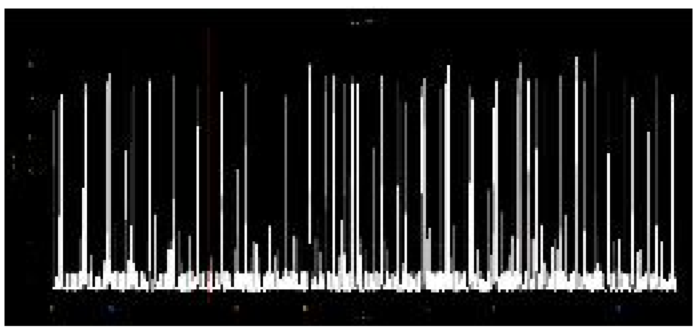


Figure 10. Current graph when the device is connected

Result:

We tried the 4 different configurations and concluded that using DCDC with external crystal resulted in the lowest current consumption. The results are shown below.

Table 1 Current consumption for two modes

	Not connected	Connected
DCDC - External Crystal	25 uA	2.11 mA

We conclude that we are able to achieve our goal current consumption as we wanted the system to have less than 5mA of current consumption.

3.1.2 BLE Data Rate Verification

The robust data transfer from the knee brace to the host device is critical to the operation of the system. This test verifies that we are able to send the necessary data(EMG, Temperature, Pressure, Battery Life) to the host device using Bluetooth Low energy. Our current required data transfer rate is 2123 Bytes/sec.

Setup:

2 NRF52 Development boards **3 Meters** apart. one of the device is acting as a **client** and other as a **server**. After the server allows the client to send notifications, the **client** sends a full package (20 bytes) every **4 ms** which translates to **5000 Bytes/sec**. This is more than enough since our system needs. The connection interval between the two is **10 ms** in order to have the lowest impact from it. The server side collects the packages sent and displays how many packages are received every 10 seconds through the J Link RTT[3] console.

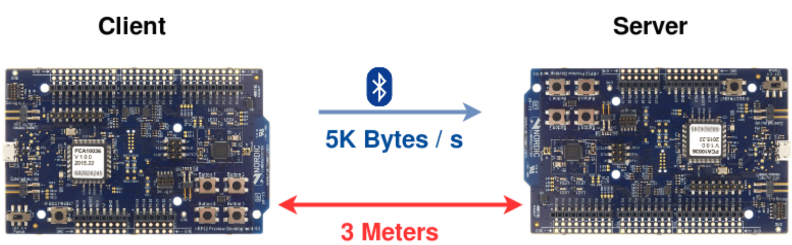


Figure 11 Setup of the Data Rate Test

Result:

Table 2 Package received

Time(s)	# Packages Received	Packages per second (Packages/ sec)	Bytes per second (Bytes/ sec)
10	2498	249.8	4996
20	5001	250.3	5006
30	7499	249.8	4996
40	10000	250.1	5002
50	12495	249.5	4990
60	14998	250.3	5000

When we were sending 5000 Bytes/sec from the client, we were able to receive average of 4497 Bytes/sec with a standard deviation of 4.32 Bytes/sec.

We conclude that we are able to use BLE as our communication protocol to send our data as it is able to support the data transfer rate we desire.

3.1.3 LED Indicator GPIO Verification

We use the an LED indicator to notify the status of the system. We control the duty cycle and the duration of the LED utilizing the GPIO. We verify that we are able to do this with the GPIO peripheral

Setup:

We use the NRF52 development board to test the GPIO module. We use LED2 on the development board that is connected to P0.18. The LED circuit is configured the same way we do on our final product. We try out different duty cycles. We deployed customized firmware for this test.

Result:

We were able to control the LED to have 10% 50% and 90% duty cycle. The test was verified with our eyes. This is shown in Figure 12.

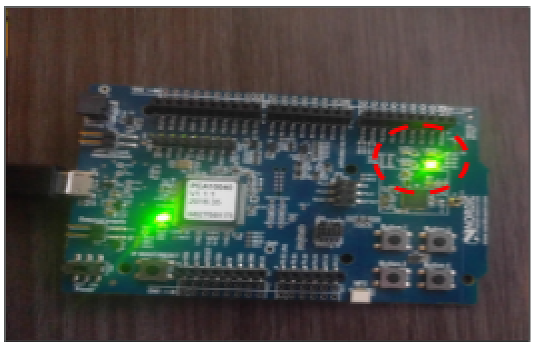


Figure 12 Setup of the Data Rate Test

We conclude that we are able to control the GPIO module to control the LED.

3.2 Analog Circuit Unit Verification

Analog Circuit verification also can be seen on the host user interface clearly. Our device sends all of the variables which NRF52 receives from the sensors to our host device via bluetooth. The microcontroller, NRF52, can do the post signal processing as we want and user can see the information that they need. As Figure 13 shows, EMG signal, two temperature signal and four pressure signal can be seen.

Figure 13 The Host User Interface on PC tablet

3.2.1 EMG Sensor Verification

The EMG circuit proved to be the most challenging to configure of the electrical components. Because we had to configure both the passive high pass filter and the active low pass filter, we had to accurately decided where the set the cutoff frequency and how much gain to give the signal. Its primary function was to be able to detect subtle muscle movements.

Setup:

We hooked up our EMG sensor to the nRF52 analog I/O port and measured the voltage readings that the electrodes picked up. Consequently, we use our trained machine learning model (Neural Network) to see if the model can detect different types of muscle movements .

Result:

Figure 14 shows the different signals that the EMG sensor was able to detect. This test was verified by differentiating between strong and weak muscle activity, and training the device to do so likewise.

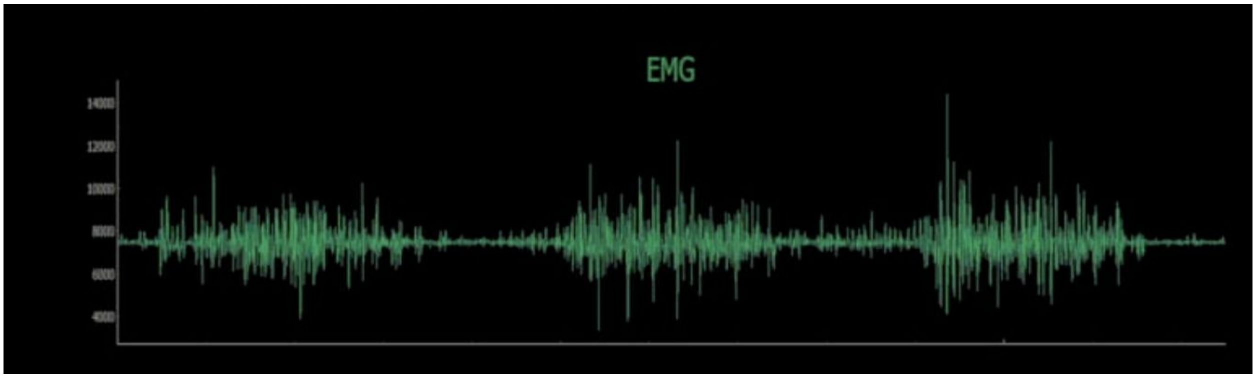


Figure 14 The Actual EMG signal Capturing from the EMG Sensor from the Host User Interface.

We also was able to verify the accuracy of the model by using a 5-Fold method on the neural network we trained and got an 73.7% accuracy.

3.2.2 Temperature Sensor Verification

Temperature sensor is to acknowledge the user that the knee is swollen up or not. To figure out the swolleness, we compare the temperature between the knee and reference body part.

Setup:

To calibrate the Temperature sensor, we used the comparison method. We left the temperature sensor and the knee brace out in the open cold for roughly 10 minutes. After that we bring it inside and measure its temperature by using a heat gun, which measures the temperature of a remote object via blackbody radiation. We both record the reading of the temperature sensor voltage and its actual surface temperature.

Result:

After the calibration using the setup above, we were able to capture the temperature difference between the knee and the reference body parts successfully. We took the logged data from the sensor and the thermometer readings to match the graph voltage to actual temperatures. The error was around $\pm 0.5°C$ with our fitted temperature model which was tolerable.

Future Calibration:

Since we used a comparison method for our calibration, it is hard to ensure ultimate temperature accuracy. If we have a more time for the calibration, it is better to use a fixed point calibration for the temperature sensor. Using a water as a source, its freezing or melting point can be used for calibration. This can provide us a $\pm 0.001\,°C$ accuracy.

3.2.3 Pressure Sensor Verification

The optimal goal for our pressure sensor is to distinguish the point where strap is loosely fitted and tightly fitted. After we configure the bridge circuit and fed our analog data into the NRF52, we tried to test out the voltage point when the strap is loose and tight.

Setup:

We connected our device into the host device. After connection is successful, put all the straps perfectly to the user and ensure that the fitted status is “Fitted”. The next step is to loosen the strap that we put tightly and see how fitted status on the host device changes.

Result:

As we expected, our host device changed the fitted status into “Adjust” whenever we did not put strap tightly enough. When we put all the strap back in tightly, it changed back into “Fitted” status. This was demonstrated in the demo.

3.3 Power Unit Verification

3.3.1 9V Battery

Our device use 9V battery as a primary power source from the Duracell. Its expected yield plot is displayed in Figure 15.

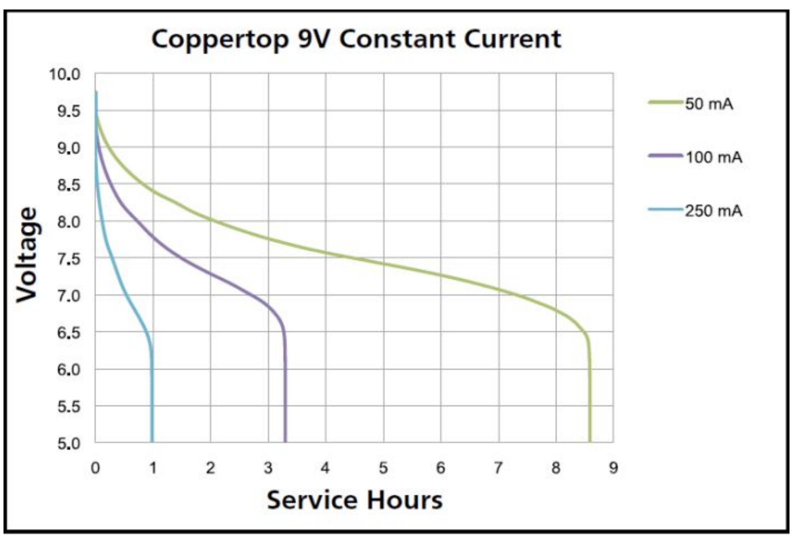


Figure 15 The Voltage output within the service Hours with different Constant Current (Duracell provided)

During the process of building our device, we realized that the power source drain into 7.7V in some point. As we can see on the Figure 15, the 9 volt battery goes down to 7.7 V really fast and starts to lose its power (voltage output) instantly after some service hours from the 7.7 Voltage input point.

3.3.2 Voltage Regulator Verification

We had planned to run our device with a 9V battery as the power source. Although we only needed about 3V to power our device, a larger capacity meant that the lifetime of our device is increased.

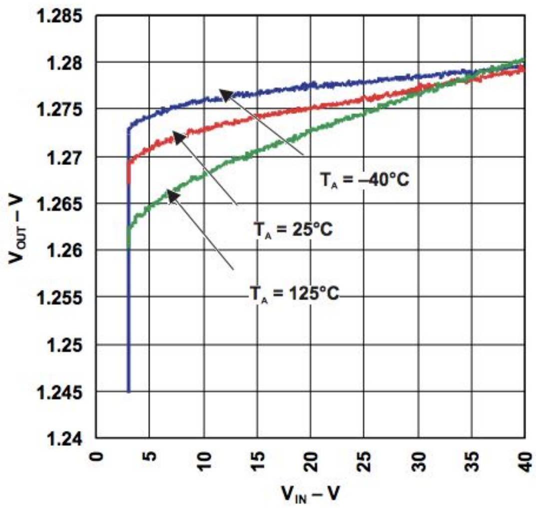


Figure 16 Vref VS Vin graph within different temperature (TI provided)

Figure 16 shows us that the Reference voltage from the equation 5 does not change until the input voltage goes under the 5 volte. We can ensure that the the our voltage regulator can provide steady 3.3V until the battery goes under the 5V.

Setup:

We connected the output of the 9V battery to the input of our voltage regulator unit: the LM317 chip. The output of this chip is what will be powering our entire project. The resistors value for the power circuit designed based on the equation 5.

Result:

We first tested the functionality of our power unit by hooking up the input feed to a digital power supply. Upon sweeping the input from a range of 5V to 9V, we observed that the output was being held constant at the value we desired. A fresh 9V battery turned out to have a usable voltage of around 7.7V when used for under an hour. By using a multimeter, we visually confirmed that the power unit was releasing a steady stream of around 3.3V, which was exactly what we needed. The output reading was also displayed on our GUI.

4. Costs

4.1 Parts

Table 3 Parts Costs

Part	Manufacturer	Unit Cost	Quantity	Total Cost
Temperature Sensor 32208548	Heraeus Sensor Technology	$_4.53	3	$13.59
Pressure Sensor 30-73258	Interlink Electronics	$_7.95	4	$31.80
INA333AIDGKR	Texas Instruments	$_4.60	7	$32.20
OPA2333AIDR	Texas Instruments	$_4.34	2	$_8.68
Skin-contact Electrodes	Covidien Kendall	$11.86	2	$23.72
Various Resistors, Capacitors, Inductors (1206)	Digikey	$_0.75	19/4/1	$18.00
			Analog Circuit Unit	$127.99
LM317	Texas Instruments	$_1.44	1	$_1.44
9V Alkaline Battery	Newegg	$_7.47	1	$_7.47
			Power Unit	$_8.91
nRF52 Microcontroller	Nordic Semiconductor	$_2.11	2	$_4.22
			Control Unit	$_4.22
			Total	$141.12

4.2 Labor

Table 4 Work Costs

Name	Hourly Rate	Total Time Invested	Total Cost (Total hourly pay x 2.5)
Dong Hyun Lee	$34.31	200	$17,155
John Lee	$34.31	200	$17,155
Dennis Ryu	$34.31	200	$17,155
		Total	$51,465

5. Conclusion

The Recovery-Monitoring Knee Brace was able to successfully detect and distinguish between different muscle activity, provide feedback based on the group of pressure sensors, and measure the temperature difference between two nodes. The main difficulties we had were soldering the components onto our comparatively small PCB and calibrating the sensors prior to building the system itself. If we had more time, then we could have also trained our signal processing models better with more data.

One uncertainty we had was with the temperature sensor, which was unable to pick up an input as accurate as the other sensors. The component we had selected was subject to a change of $1\,\Omega$ for every $1\,^{\circ}C$ increase in temperature. But the range of our temperature detection was to be within $\pm 2^{\circ}C$ of the reference temperature. One improvement we could have made early on was the selection of a more optimal temperature sensor for our needs. The one we implemented did not yield a strong enough output in that small range of voltages, so if we had used another thermistor instead, then we would have been more successful in detecting a small range of output voltages.

We can extend our project to be applied to not just the knee, but also to different parts of the body. Because clinics are looking for wireless substitutions that will assist the process of diagnosing patients, this holds true for any other ligament. Our implemented system can be used on the arm, back, or neck. Additionally, we can mount our circuit onto a flexible substrate and hide it between the sleeves of the leg brace, making it more aesthetically appealing.

References

[1] Anon, (2017). [online] Available at: https://www.nordicsemi.com/eng/Products/Power-Profiler-Kit [Accessed 7 Nov. 2017].

[2] *Devzone.nordicsemi.com. (2017). Tutorials - Nordic Developer Zone. [online] Available at: https://devzone.nordicsemi.com/tutorials/11/ [Accessed 1 Nov. 2017].*

[3] SEGGER - The Embedded Experts. (2017). *RTT Viewer*. [online] Available at: https://www.segger.com/products/debug-probes/j-link/technology/rtt-viewer/ [Accessed 13 Dec. 2017

[4] Devzone.nordicsemi.com. (2017). Maximum RSSI value of -42 dBm - Nordic Developer Zone. [online] Available at: https://devzone.nordicsemi.com/question/64899/maximum-rssivalue-of-42-dbm/ [Accessed 6 Oct. 2017].

[5] Infocenter.nordicsemi.com. (2017). Nordic Semiconductor Infocenter. [online] Available at: https://infocenter.nordicsemi.com/index.jsp [Accessed 3 Oct. 2017].

[6] interlinkelectronics. (2017). Pressure Sensor FSR400 Series Datasheet. [online] Available at: http://www.interlinkelectronics.com/datasheets/Datasheet_FSR.pdf [Accessed 3 Oct. 2017].

[7] Punchthrough.com. (2017). Maximizing BLE Throughput on iOS and Android - Punch Through. [online] Available at: https://punchthrough.com/blog/posts/maximizing-blethroughput-on-ios-and-android [Accessed 6 Oct. 2017].

[8] Sensing.honeywell.com. (2017). Installation Instructions for the 700 Series Platinum RTD Temperature Sensors. [online] Available at: https://sensing.honeywell.com/index.php?ci_id=50017 [Accessed 6 Oct. 2017].

[9] Sensing.honeywell.com. (2017). 700 Series Platinum RTDs Data Sheet. [online] Available at: https://sensing.honeywell.com/index.php?ci_id=50019 [Accessed 6 Oct. 2017].

[10] Texas Instruments. (2017). LM3671/-Q1 2-MHz, 600-mA Step-Down DC-DC Converter. [online] Available at: https://cdn-shop.adafruit.com/product-files/2745/P2745_Datasheet.pdf [Accessed 4 Oct. 2017].

[11] Texas Instrument. (2017). INA333 Data Sheet. [online] Available at: http://www.ti.com/lit/ds/sbos445c/sbos445c.pdf [Accessed 3 Oct. 2017].

[12] Duracell Coppertop. (2017). MN1604 Data Sheet. [online] Available at: https://d2ei442zrkqy2u.cloudfront.net/wp-content/uploads/2016/03/MN1604_US_CT1.pdf [Accessed 13 Dec. 2017].

Appendices

A - Requirement and Verification Table

Table 5   System Requirements and Verifications

Requirement	Verification	Verification status (Y or N)
1. Maximum of 5 mA of current consumption.	1. Utilize a power analyzer(Function in the Nrf52)to measure average currentconsumption.It will ensure us the usage of current on the device.	**Y**
2. Nrf52 is going to use Status LED indicator , it has to provide digital signal to control LED indicator.	2. We are going to check on the LED indicator when we turn the device on and connect the device with host device. These two actions have to show correct indication action of LED. (For example, Green light on when we turn the device on and constant blink when the device is correctly connected with host device.)	**Y**
3. Support minimum of 2162Bytes/second of data transfer to the host device	3. Data transfer rate test and package loss test will be done by checking the nRF52 status with computer via USB and create a measurement test and check Nrf52 is delivering proper data into the host device.	**Y**
4. Temperature Sensor has to capture precise temperature of user's body in real time with precision of±0.5° Celsius with actual real world temperature.	4. a. Measure the temperature of User's Knee with temperature sensor. b. Use external thermometer and measure the Knee's temperature. c. Compare the knee temperature from the temperature sensor and actual thermometer to ensure that temperature sensor is working properly.	**Y**
5. Pressure Sensor has to distinguish between perfect fit and loose fit of straps within user's knee.(The pressure quantity on the pressure sensor is not going to be constant depend on the user's usage and their own feeling of tightness on the knee.However,loosely put straps will be detect no pressure.)	5. a. Tightly put straps of Knee brace into the user and check the host device for a status ofthe straps. b. Loosely put straps of Knee brace into the ser and warning has to show up into our host device because pressure sensor indicates low pressure.	**Y**
6. EMG Sensor has to capture the signal increases with an increase in muscle activity/effort.	6. a. Using a host device within the EMG sensor connected into one of our team member and compare the rectified signal's amplitude between flexed muscle and relaxed muscle.	**Y**
7. Status LED has to show power on power off.	7. a. Take out the battery and check the LED and put battery back into power module to see that LED light kicks in.When it kicks in, the blink time will be short. (blink time means the time interval of light on and off.) b. When we connect the microcontroller with host device, the LED blink time will belonger.	**Y**
8. The 9V battery has to provide fixed 9V inthe tolerance of±3%	8. a. Use multimeter to ensure that the 9 Volt battery providing proper voltage. b. Check the voltage value on the multimeter and make sure that battery is providing fixed voltage value in the tolerance range.	**Y**
9. a. Buck Converter has to provide steady 3.3v with ±3% from the battery. b. Buck Converter has to provide about 600mA into the device..	9. a. First connect the power supply into the Buck Converter as a power source and change the input voltage from 5V to 9V within the range of DC/DC converter allows. b. Measure the value from voltage output of the Buck converter and verify that the voltage is providing fixed 3.3Volt. c. Our software in the host device will show input voltage which microcontroller is getting from the power circuit.	**Y**

B - Device picture

Figure 17 Product picture 1

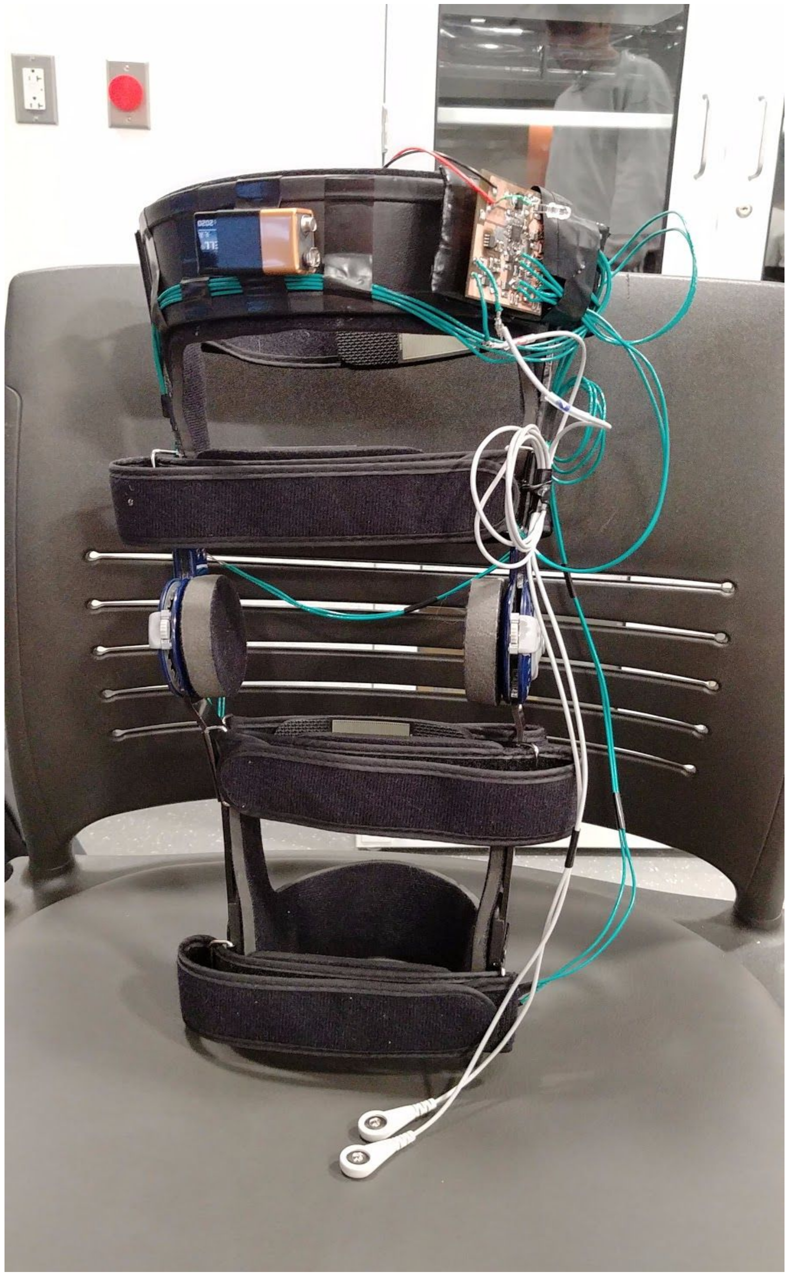


Figure 18 Product picture 2

C - Sensor Location

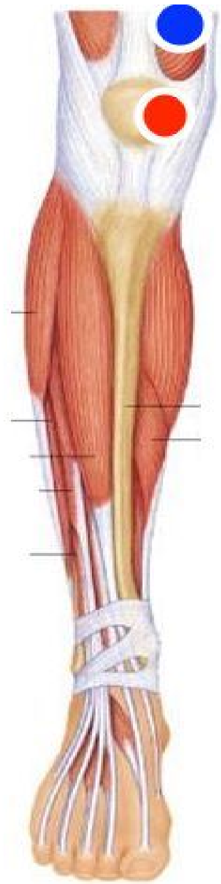


Figure 19 Temperature Sensor Location: Blue is reference and Red is Knee.

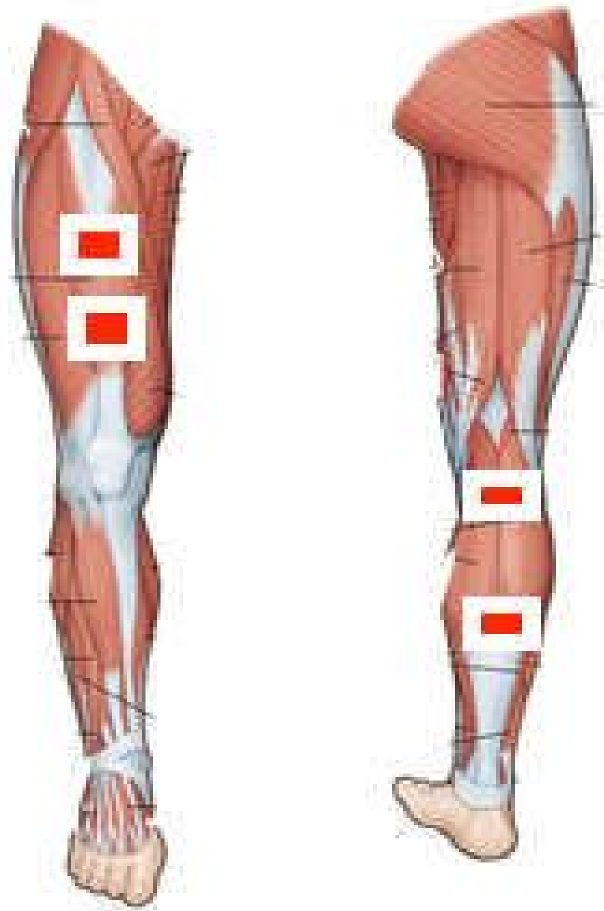


Figure 20 Four Pressure Sensor Location

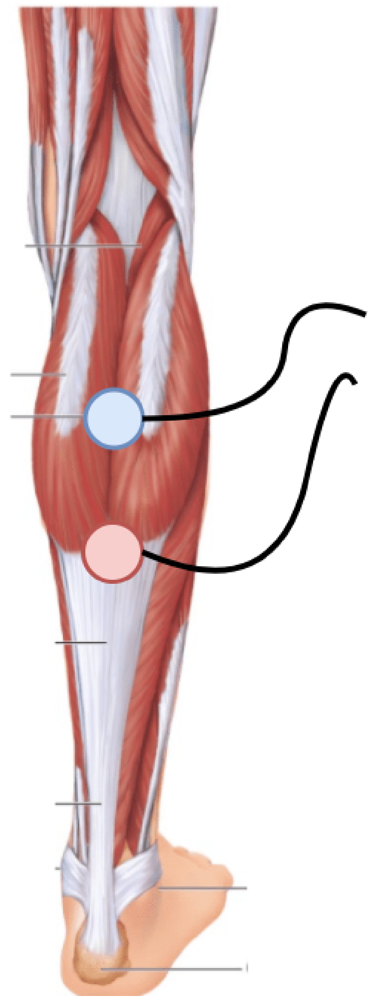


Figure 21 EMG Sensor Location.

D - EMG Signal processing visualization

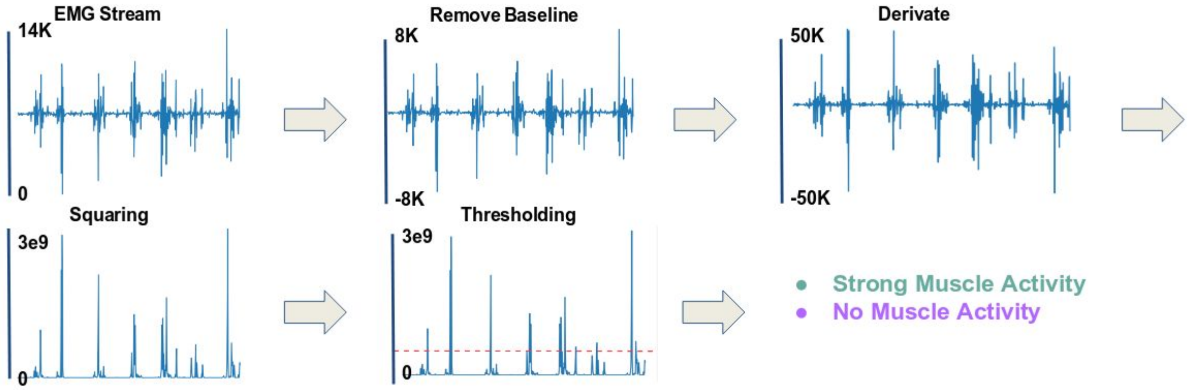


Figure 22 Lightweight signal processing to detect muscle activity