Wearable Posture Corrector

By

Shubham Agarwal
Vishal Guntupalli
Vignesh Sridhar

Final Report for ECE 445, Senior Design, Spring 2016

TA: James Norton

3 May 2016

Project No. 39

Abstract

The focus of this project is to build a wearable device to detect wearer's posture; especially lower-back posture and provide haptic feedback as well as feedback in graphical format to user's device.

A gyroscope and an accelerometer are used to measure the angle of the wearer's lower back; and this information is processed and tracked by a microcontroller which relays it to a Bluetooth module to be sent over to a paired device as well as turn on / off the vibration motor to indicate poor posture to be corrected immediately.

The device successfully alerted the user via the vibration motor if the offset from the calibrated position was greater than a pre-defined threshold angel; in our case 30 degrees. Also, it relayed the angle values that the microcontroller calculated to a paired device via Bluetooth.

Contents

1. Introduction

1.1 Purpose

According to the American Chiropractic Association, 31 million Americans experience low-back pain at any given time [1]. Lower back pain is the second most common reason for visits to a doctor, outnumbered only by upper-respiratory infections. Americans spend at least $50 billion each year on back pain [1]. A study done by the Agency for Healthcare Research and Quality showed that most cases of back pain are mechanical or non-organic, meaning they are not caused by some major illness [1]. National Center for Biotechnology Information states that acute episodes of back pain are associated with muscle strain. Poor posture, as will be defined in a moment is a major player in making 50% of people in the industrialized world suffer from some form of back complaint [2]. 8 out of 10 Americans have at some point been affected by lower back pain, accounts for 93 million days of lost work annually at an estimated cost of $11 billion [3].

Our posture affects the health of our spine. The vertebrae in the lumbar region are the largest in the spine; with L4 and L5 being damaged most frequently [2]. The human body is not made to sit in front of computer screens for hours, the body weight gets distributed predominantly to the lower back and pelvic areas. Figure 1 shows a depiction of the spine.

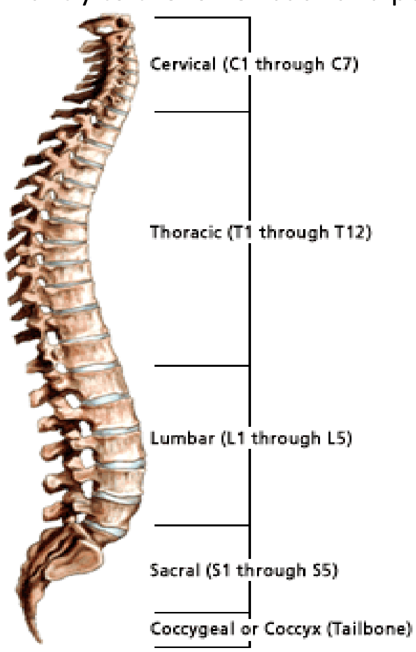


Figure 1: The Spinal Cord

Our wearable which is to be at the top of the top of the lumbar spine(small of the back close to L1) will maintain posture within a range while accepting slouching as a way of relaxing muscles depending on the users past data. The solution we created was a small wearable device which will maintain posture within a range of angle while accepting a small form of slouching as leeway.

1.2 Objectives

Goals:

- Provide discreet real-time haptic feedback and also data on a personal device regarding the posture of the wearer via credit card sized device.

Functions:

- Collect data on the position of the lower back through an accelerometer and gyroscope.
- Convert the raw values from the sensors to angular values of the lower back.
- Send feedback through the vibration motor indicating the user needs to correct their posture.

Benefits:

- Helping people maintain good back posture so that they don’t suffer from back pain later on in their life.
- Potentially the first step in developing wearable devices which will provide feedback regarding a user’s movement and health.
- It also teaches better posture and potentially a routine for people to follow through their day.

Features:

- Relaying the data regarding angles via a Bluetooth to a front end application.
- Keeping track of various movements by the user
- Rechargeable battery
- Mobile and discrete

2 Design

2.1 Block Diagram

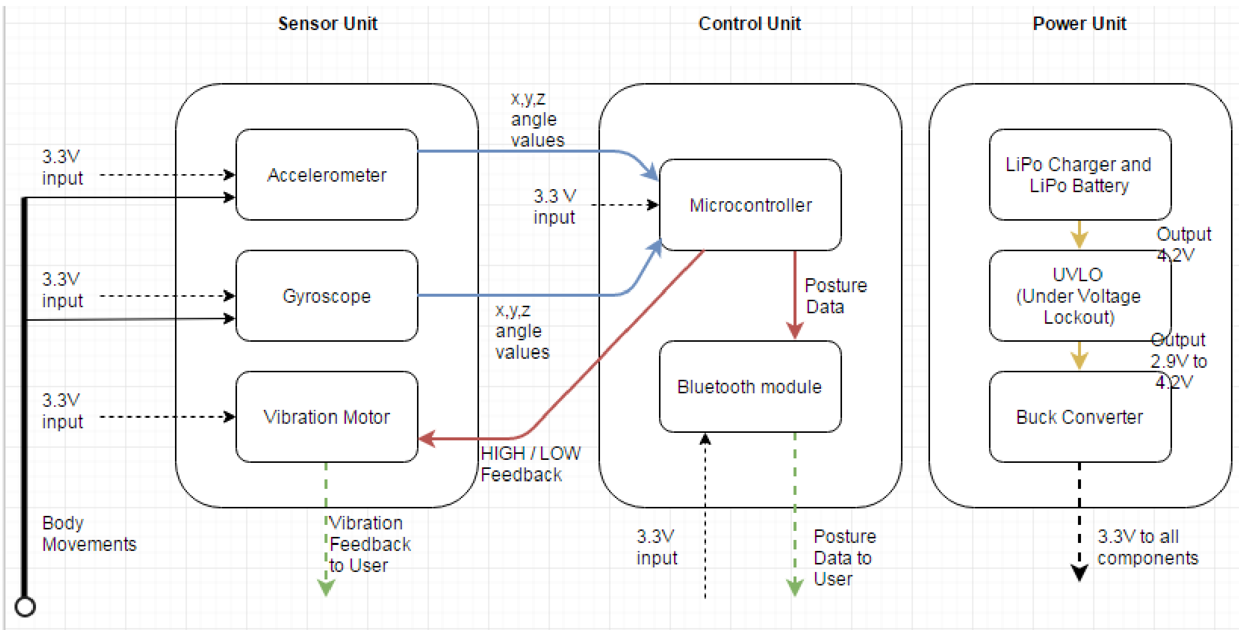


Figure 2: Block Diagram of Our System

Yellow lines / black lines	Power from battery to components
Blue lines	From sensors to microcontroller
Red lines	From microcontroller to components
Green lines	From components to user

Table 1: Legend for block Diagram

2.2 Block Descriptions

2.2.1 High Level Block Description

This project consists of 3 units which are integral for the functioning of the device. The sensor unit is responsible for interacting directly with the wearer measuring raw values of posture and providing the hepatic feedback. The control unit is where the raw data is processed into angles of the lower back and also relay that to a Bluetooth module which will send it to the front end application of our design for the user to keep track of their posture.

Finally, the power unit is composed of the rechargeable battery along with an Under Voltage Lockout(UVLO) circuit and also a Buck Converter to enable a steady output to the other components.

2.2.2 Sensor Unit

Accelerometer (MMA8452Q): Three vectors which correspond to the acceleration values in the x, y, and z axis. It is powered at 3.3V which is provided by the battery. I^2C bus is used to pass the information to the microcontroller. Has its own unique address.

Gyroscope (ITG-3200): Three vectors which correspond to the rotational value in the x, y, and z axis. It is also powered by 3.3V which is provided by the battery. I^2C bus is used to pass the information to the microcontroller. Has its own unique address.

Vibration Motor: The motor is switched on or off via the microcontroller sending a high or low signal from the digital pin of the microcontroller.

2.2.3 Control Unit

Microcontroller (ATMega328p): The microcontroller acquires data from the sensors via the I^2C bus, converting the raw values into angles. Changes in angle from a calibrated 'ideal' back posture will be used to determine whether or not a signal has to be given to the vibration which will indicate to the user of their poor posture. The microcontroller will also relay this information to the Bluetooth module which will allow for data to be sent to a paired device where the user will be able to track their progress.

Bluetooth Module(JY-MCU): The Bluetooth module receives digital values from the microcontroller and sends it to a paired device. The data appears on the serial monitor which can then be processed via a front end application to track through the day. The module is powered by the battery at 3.3V.

2.2.4 Power Unit

LiPo Battery and Charger: We are using a very small, extremely lightweight Polymer Lithium battery which can output 3.95 ± 0.25 V at 400mAh. These batteries require special charging and thus we use a Spark fun LiPo charger. This charger is a simple charging circuit to charge 3.7 V to 4.2 V LiPo batteries. It features a status LED to indicate when the battery is completely charged and that voltage and current to the battery is completely cut-off. This charger uses a micro-USB cable plugged in from the wall, and maintains an output to a charging battery to 4.2 ± 0.03 V and 400mA ± 10mA.

UVLO: This UVLO (Under Voltage lockout) circuit will turn off the output power to the circuit if the operational voltage of the LiPo battery drops below 3.1V.

It also turns on the output power to the circuit if the operational voltage of the LiPo battery rises above 3.5V, due to hysteresis. The start voltage is a lot higher than the cut-off voltage to avoid rapid switching on/off when battery reaches 3.1V. This prevents from battery from damage.

Buck Converter: The ideal voltage that is required by our components is 3.3V and the buck converter is used for the purpose of pushing down the voltage by our battery. It has an input of 3.95 ± 0.25 V with a maximum of 400mA and the output is approximately 3.3 ± 0.03V and a maximum current of 400mA

2.3 Device Flow Chart

Figure 3 below depicts the basic process the microcontroller follows, using data from the accelerometer and gyroscope, and determining whether the user is in a good or bad position.

2.3.4 Device Flow Chart Description

Start – Device is turned on.

Calibrate and Set Timer - User wears the device and stands still for 10 seconds while the microcontroller builds a reference. Timer is set up so that the user has to maintain the position for 30 minutes after which the user will be encouraged to move.

Measure Back Angle – The microcontroller polls the data from the accelerometer and gyroscope and uses it to calculate an angle.

Send to Bluetooth - The angular data that has been acquired will be sent to the Bluetooth so that it may be transmitted to the front end application.

Good Posture? - Microcontroller compares the newly measured angle with the calibrated reference and will determine if the user is in a good posture or not.

Vibrate - If the microcontroller decides the user is in a poor posture then a high signal will be sent to the motor which will cause it to vibrate and the user will be notified of the need to change their posture.

Timer = 0? – This would indicate that the user can move and not have to maintain the previously calibrated posture.

End - Device is turned off

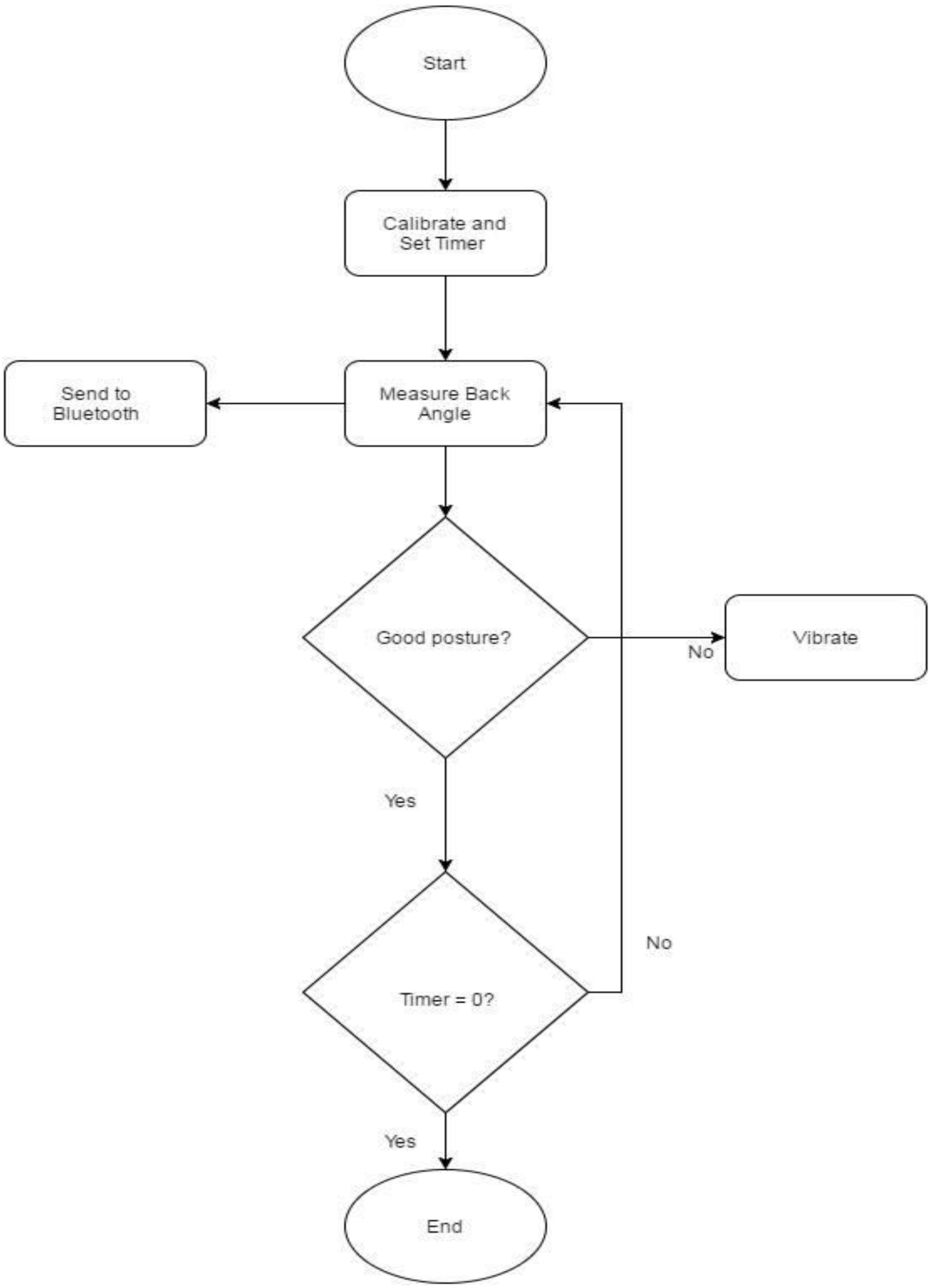


Figure 3 – Flowchart of the Microcontroller

3. Calculation and Simulation

3.1 Back Angle Measurement

3.1.1 Accelerometer

For measuring the lower back angle, we use a three axis accelerometer and a three axis gyroscope. With it we can a get a complete 360^{o} of motion on each of the axis. The accelerometer gives raw g-force values while the gyroscope gives rotational values along each axis. The raw g-force values are not that helpful in acquiring back angles, but they can be converted into an orientation with respect to gravity using the inverse tangent function.

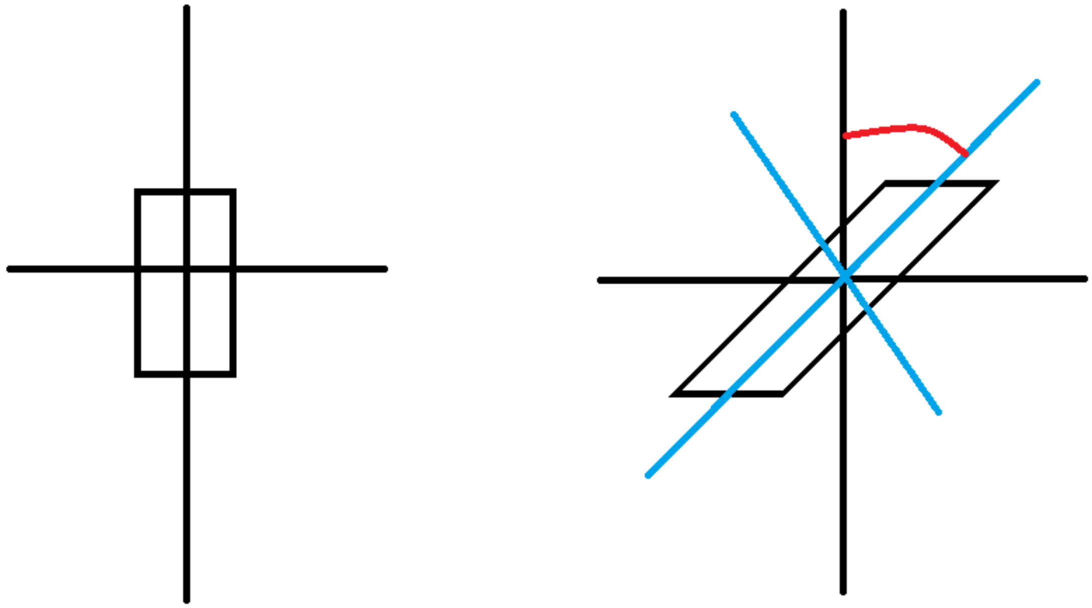
Equations 1 and 2 can be used to find the angles about the x-axis and the y-axis.

$$A_x = tan^{-1}\left(\frac{x}{\sqrt{y^2+z^2}}\right) \quad (1)$$

$$A_y = tan^{-1}\left(\frac{y}{\sqrt{x^2+z^2}}\right) \quad (2)$$

During the calibration period, the angles in the x-axis and y-axis are averaged over a period of 15 seconds. After that time, if either of the angles in the x-axis or y-axis deviate by more than 15^{o} relative to the calibration value, then it can be said that the wearer is out of position. We are only focused on the angular difference in 2 out of the 3 axis as normal back movement. The calibration is something offsets the effects of potential drift. This is so because of the fact that we look for deviation from the calibrated position more so than acquiring exact back angles.

In Figure 4 we have marked in red the angle we wish to measurement. On the left it depicts the measurement approximately at mean position. When a person bends forward, the calibrated position is represented by the black axis and then the red angle, measure the displacement. We can imagine that as the person bends forward more and more, the angle increases and when it is beyond a certain threshold then it is said that the wearer is in bad posture and it needs to be rectified.



`Figure 4 –Accelerometer and Deviation

3.1.2 Gyroscope

We used a three axis gyroscope which gives us the rotational value along all three axis, and it is the change in angle over change in time. To be able to get the angle value from gyroscope we have to integrate them, but unfortunately we don't have a continuous function which can be used to integrate. But it can be modified into a series which we can use to find the angles as shown in equation 3. The summation is over the period of time the gyroscope is active and sending data to the microcontroller. T_s represents the sampling time, for our design we were sampling at 20Hz, which would place the sampling time at 0.05 seconds.

$$\int_0^T d\theta \ dt \ = \ \sum_0^t d\theta \, T_s \qquad (3)$$

Our gyroscope does have a form of drift to it which is common to the gyroscopes to it. During the calibration process we take an average of the values of the gyroscope so that they can be used to zero it. Furthermore, there is a high pass filter implemented as that will ignore any small values which is due to the drift that may occur post calibration while in use.

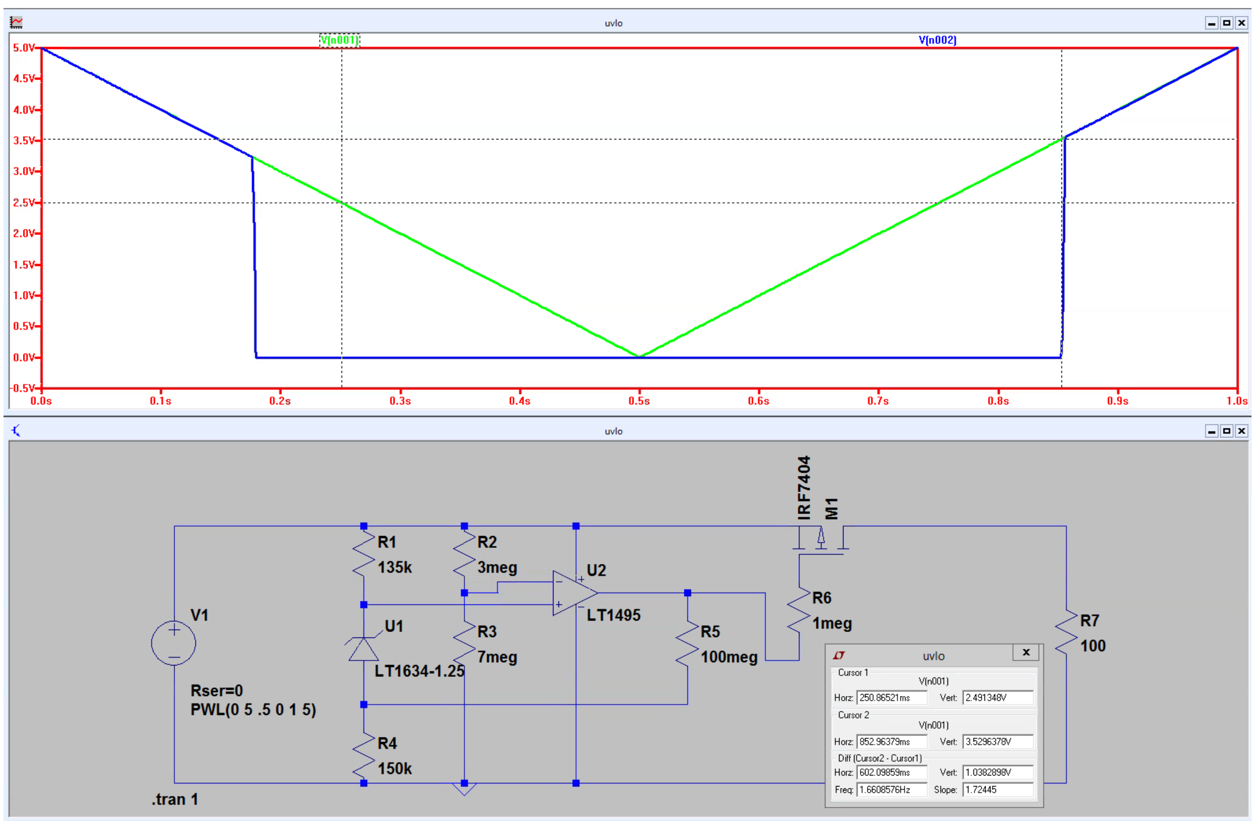
Overall both the gyroscope and accelerometer work together to detect whether there is any movement away from the mean position which has been calibrated as a zero. The accelerometer is prone to random spikes which give false data, to offset this the gyroscope which does not have the same problems ensure the minimization of false positives.

During our simulated runs we placed the device on a flat surface and allowed it to calibrate. Then we proceeded to tilt in different axis at multiple speeds. We could see changes on our devices which were getting the data via Bluetooth. Plus the vibration motor had a good response which we could feel easily.

13.2 Power

3.2.1 UVLO

We design a handmade circuit to meet our exact power requirement. Thus, we had to do the resistor calculations. A LTSpice simulation is in figure 5.



Green line	PWL (piecewise linear) of source voltage
Blue line	Output voltage across R7 (Load)

Figure 5 –Under voltage Lockout

We have a shunt voltage resistor in the chip LT1634-1.25 and thus can get a voltage difference of 1.25V. We will define the point above U1 as point 'B' which goes to the positive input of our op amp comparator. Thus, we can get a voltage equation at point 'B'[4]. Let V_t be the lockout voltage. Thus, we can get our equations as follows:

$$V_b = 1.25V + (I * R_4) = 1.37 \tag{4}$$

$$I = \frac{V_t - 1.25V}{R_4 + R_3} = 800nA \tag{5}$$

$$1.37 = 1.25V + (800nA * R_4) \tag{6}$$

$$R_4 = \frac{(0.12 * 10^9)}{800} = 150{,}000\Omega \tag{7}$$

Thus, we are able to achieve an output of 0V on different input ranges depending on whether the circuit if active or charging.

3.2.2 Buck Converter

To be able to step down the voltage, we designed a handmade asynchronous buck converter. Figure 6 is a basic circuit diagram of the implemented circuit. For a switch we used a nMOS transistor.

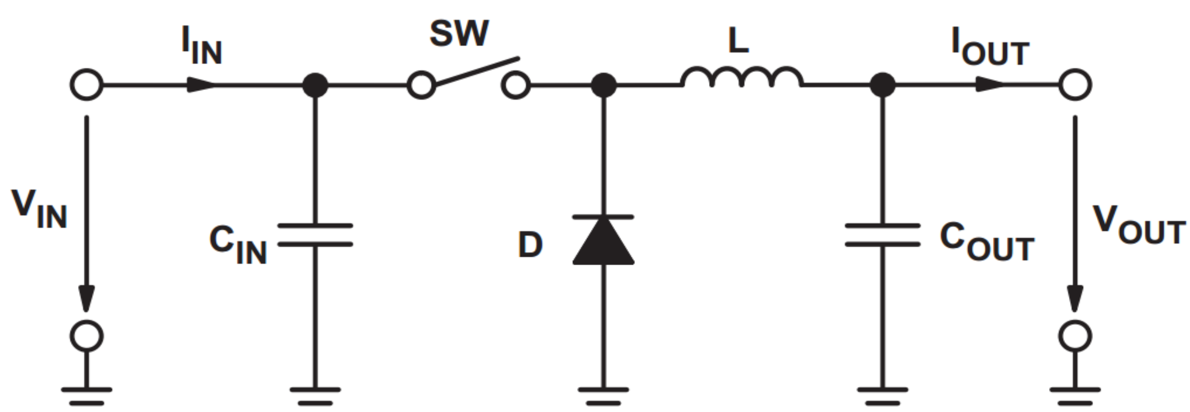


Figure 6 – Buck Converter Circuit Diagram

The nMOS was controlled by a square wave that we sourced from a waveform generator for the simulation. The duty cycle was calculated as shown in equation 8, which was calculated to 84.61%. The V_{in} is approximately 3.9 as that is the input from the battery and the V_{out} is 3.3V as that is where all the components have their functioning sweet spot.

$$V_{out} = D * V_{in} \tag{8}$$

For the PWM we drove the waveform generator to generate a square wave at 200kHz, 10V peak-to-peak, with no offset and a duty cycle of 84.61%. Despite our circuitry we ordered a chip to replace the circuitry and this was due to the power efficiencies. For our simulation we used a 30Ω resistor as this would draw approximately the same amount of current as our components which we measured to be 120mA. This resistance value was gotten to using Ohm's Law which is shown in equation 8. The input power was calculated at 0.4485W using equation 9 where V_{in} is 3.9V and the I_{in} is 115mA.

$$V = I * R \tag{8}$$

$$P_{in} = I_{in} * V_{in} \tag{9}$$

The output power of our buck converter was calculated with a load of 30ohms and the method shown in equation 10.

$$P_{out} = \frac{{V_{out}}^2}{R} \tag{10}$$

The output voltage (V_{out}) was measured via a multimeter to be 3.38V and the power consumed was 0.3808. We calculated the efficiency of the buck converter we built to be 84.91% using equation 11.

$$Efficieny\ = \frac{P_{out}}{P_{in}} * 100\% \tag{11}$$

The chip we decided on was LM3671 and with specifications of our battery and current draw, its efficiency is significantly better. The difference in efficiency is approximately 10%. Figure 7 shown below is an efficiency graph that was provided with the chip we ordered. The red shows approximately where our circuits efficiency would lie in a graph of that sort while the green arrow shows the efficiency of the chip when it is used as part of our device.

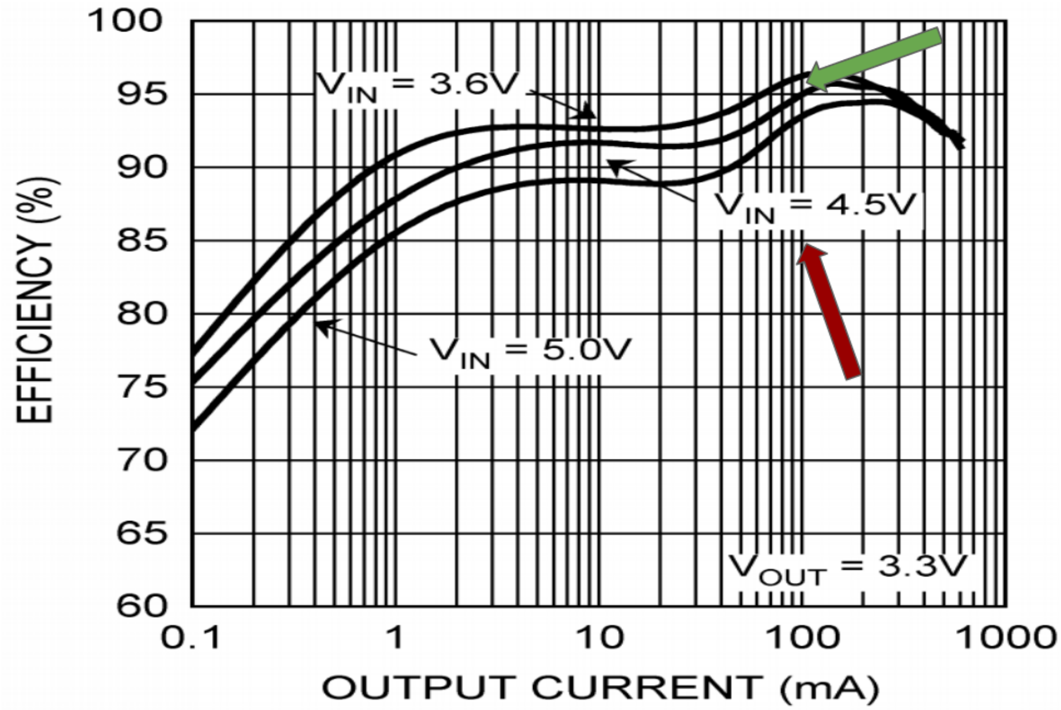


Figure 7 – Efficiency Graph of LM3761

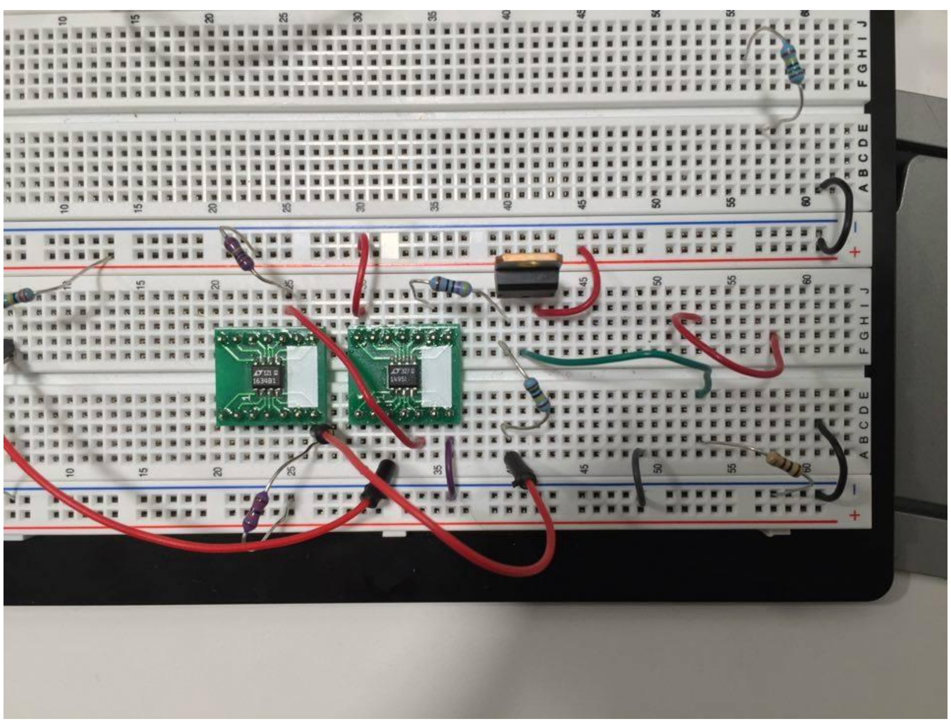


Figure 8 – UVLO circuit Implemented on Bread Board

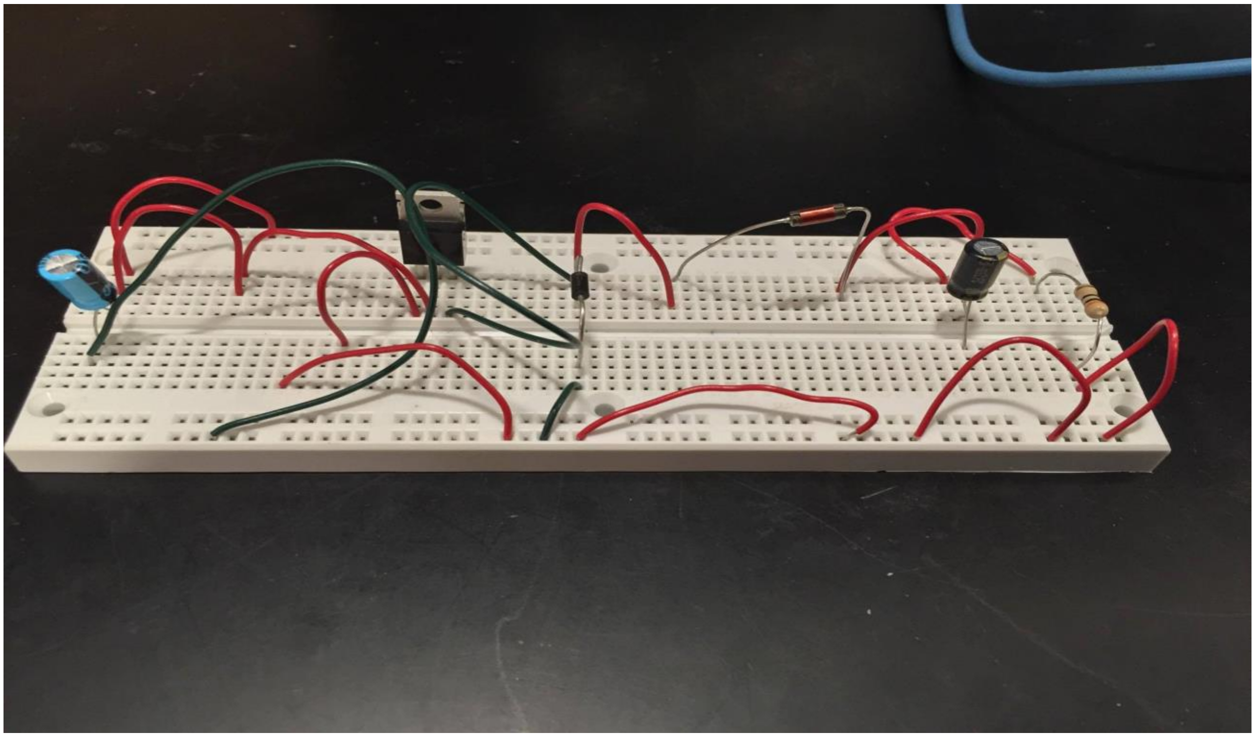


Figure 9 – Buck Converter Implemented on a Bread Board

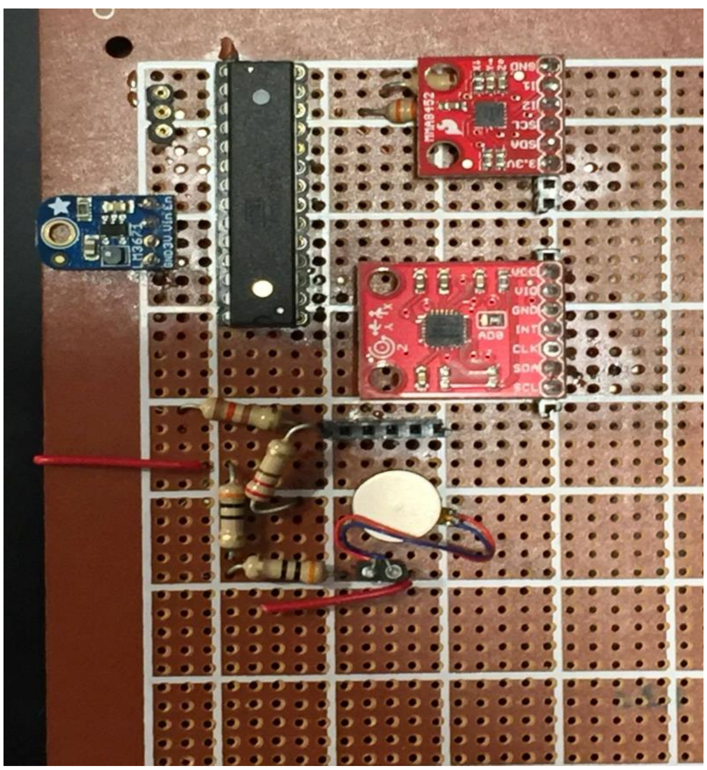


Figure 10 – Final Product on Prototype Board

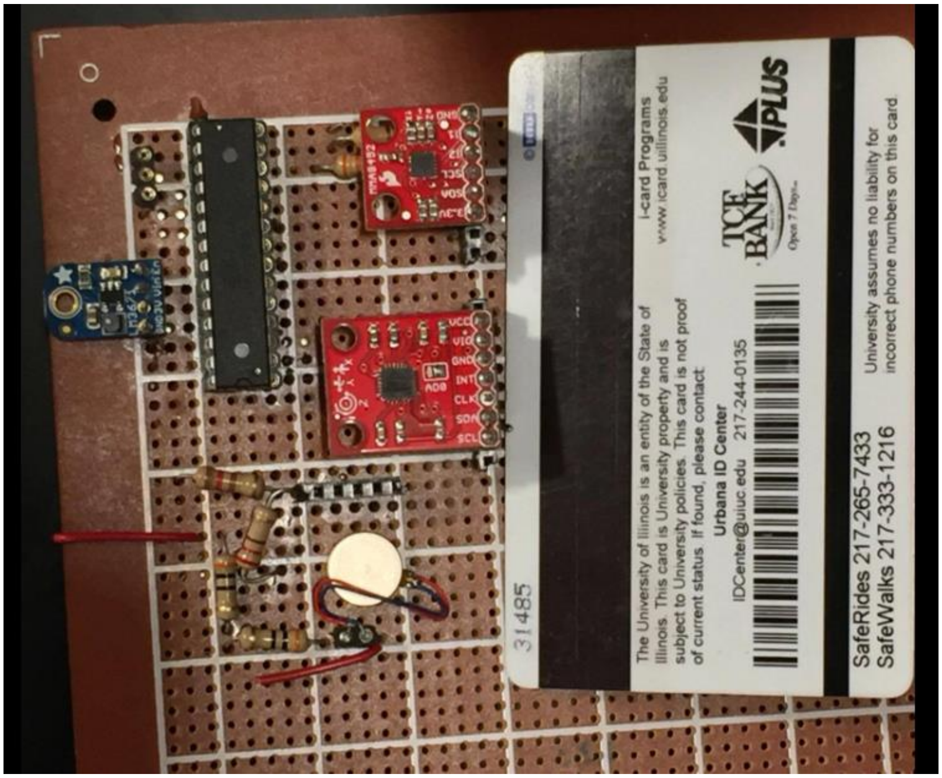


Figure 11 – Final Product on Prototype Board Relative to a card.

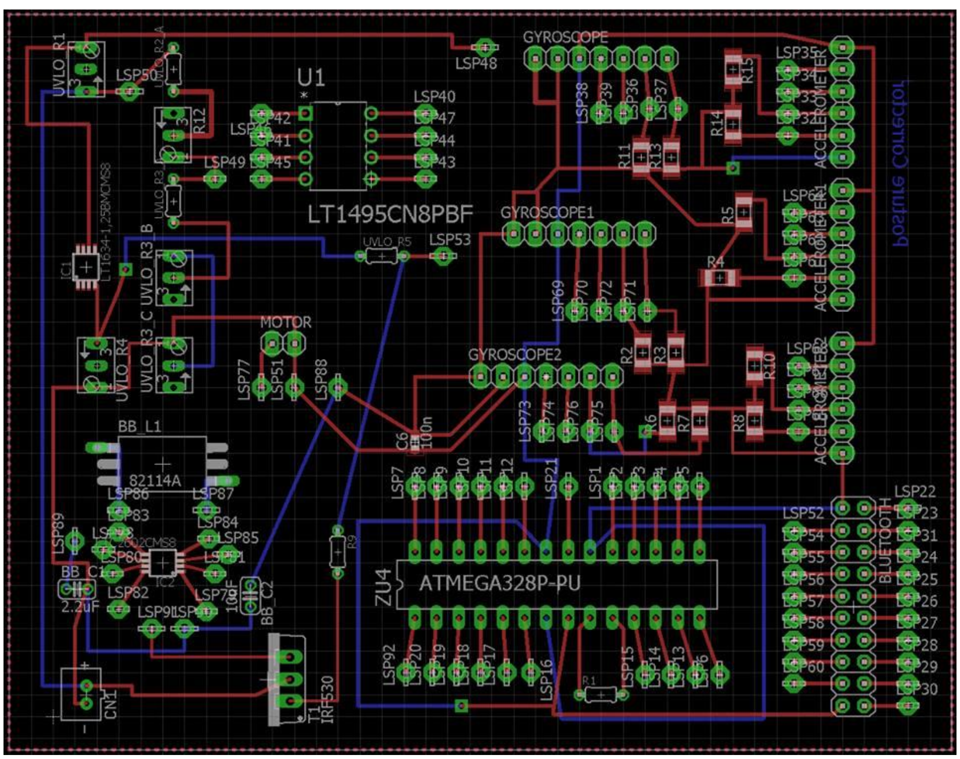


Figure 12 – Final Board Design for PCB

4. Costs

4.1 Parts

Table 2 – Costs for Parts/Services

Part/Service	Part Number	Quantity	Unit Cost	Total
Microcontroller	ATmega328p	1	$30.00	$30.00
Accelerometer	MMA8452Q	1	$9.95	$9.95
Gyroscope	ITG-3200	1	$24.95	$24.95
Vibration motor	B1034.FL45-00-015	1	$4.95	$4.95
Bluetooth	JY-MCU	1	$7.99	$7.99
LiPo charger	MCP73831/2	1	$7.95	$7.95
LiPo battery	GSP652535	1	$6.95	$6.95
Buck converter	LM3671	1	$14.00	$14.00
Shunt Voltage Reference	LT1634-1.25	1	$5.10	$5.10
Op Amp Comparator	LT1495	1	$3.20	$3.20
Prototype Board		2	$6.37	$12.74
Wires		2	$6.87	$13.74
PCB Printing		1	$55.00	$55.00
Assorted Resistors		1	$10.69	$10.69
Assorted Capacitor		1	$10.99	$10.99
Header Pins		1	$4.70	$4.70
Solder and Flux		1	$5.00	$5.00
Assorted Inductors		1	$10.90	$10.90
N-channel and P-channel MOSFETs		1	$8.69	$8.69
Assorted Diodes		1	$4.94	$4.94
	Total Cost:			$252.43

4.2 Labor

Table 3 – Costs for Labor

Name	Hourly Rate	Total Hours	Cost
Shubham Agarwal	$35	180	$6,300.00
Vishal Guntupalli	$35	180	$6,300.00
Vignesh Sridhar	$35	180	$6,300.00
	Total	**540**	**$18,900.00**
	Overhead	**2.5 * Total**	**$47,250**

5. Conclusion

5.1 Accomplishments

We were able to acquire reliable and accurate sensor (accelerometer / gyroscope) data, and a very responsive motor. We tested the data by moving the device into different angles, and the motor by continuously and pushing the device beyond the programmed thresholds. Our overall device was about the size of a credit card despite having it on a perfboard with non-optimized parts. Examples of this would be the large resistors and capacitors we used. We were able to get all our modules to work, and the control unit and sensor unit to work together. This we feel is a good test model, which will provide a good platform to make improvements.

5.2 Uncertainties

A major uncertainty, which caused an issue was the question of posture. This question was raised consistently, and after large amounts of research we were able to answer what good posture is. Early on we were only able to answer what good posture isn't. This is an area which we need to improve upon to be able to make a solid device.

Another uncertainty is the ability to successfully market this as a viable wearable. Wearable devices have a certain novelty to them but it is hard to get people to consistently use them as many are just discarded soon after purchase. Since our device is one that consistently vibrates to alert the user it might be perceived as nagging or annoying and may not see much success.

5.3 Ethical considerations

5.3.1 Safety Considerations

Making a wearable comes with a lot of responsibility. We have to make sure that even if something fails in our product, it either recovers gracefully; else stays intact and limited to the device such that it does not harm the user.

The battery has to be properly packed and isolated to prevent leakage, and the device has to be kept away from people using pacemakers as electrical and magnetic components in the circuit may interfere with the pacemaker.

5.3.2 Ethical Considerations

Our project follows the IEEE code of ethics. We accept responsibility in making decisions with the safety and welfare of the public. Our project also acknowledges errors and credits properly the contribution of others.

5.4 Future work

In general, we expect the entire product should be smaller than the size of a credit card and placed into a pocket stitched on the inside of the back of a shirt. The front end application needs to be improved so that it keeps track of the posture through a day and suggest potential improvements other than the constant vibration. It would also help alleviate the feeling of the device nagging at the wearer.

Currently the code used is not optimized, the Bluetooth constantly sends data which will lower the battery life. Optimizing it so that aggregate data is sent at regular intervals will help extend the battery life. The algorithm for detection can be improved that it is possible to suggest routines the user has to follow and then have the device track if the user is following them properly.

An example of a good routine during an interval of 3 hours:

- Sitting while leaning back around 110 degrees to 150 degrees for 30 minutes
  Can do this while taking phone calls or reading a paper.
  This position leads to strain in the neck and is thus not advised.
- Leaning in front from 0 to 78 degrees for 20 minutes.
  This creates movement in the muscles and can help relax them as well.
- Sit upright from around 79 degrees to around 105 degree for 1 hour 50 minutes.
  Use this for doing work.
- Walk around for 20 minutes.

References

[1]"Back Pain Facts and Statistics", *Acatoday.org*, 2016. [Online]. Available: http://www.acatoday.org/Patients/Health-Wellness-Information/Back-Pain-Facts-and-Statistics. [Accessed: 2016].

[2]"Ergonomics of sitting", *Ergo.human.cornell.edu*. [Online]. Available: http://ergo.human.cornell.edu/DEA3250Flipbook/DEA3250notes/sitting.html. [Accessed: 2016].

[3]J. Van, "Study finds leaning back at 135-degree body-thigh angle is best", *Pittsburgh Post-Gazette*, 2016. [Online]. Available: http://www.post-gazette.com/news/health/2006/12/06/Study-finds-leaning-back-at-135-degree-body-thigh-angle-is-best/stories/200612060260. [Accessed: 2016].

[4]A. Lee, "4.5µA Li-Ion Battery Protection Circuit", *LT-Journal*, pp. 36-37, 1999.

Appendix A Requirement and Verification Table

Table 4 Requirements and Verifications

1. Li Po Charger

Requirement	Verification	Points
1. When connected to a wall charger micro USB, the output of the charger should be 4.2 V ± 0.03 V and 400mA ± 10mA. ----------------------------------- 2. When the battery is charged to a maximum capacity of 3.9 ± 0.03 V; the charger cuts off voltage supply to the battery, and the voltage and current across the terminals of the battery go to 0V and 0A respectively.	• Lithium Ion batteries need a special type of charger as they are very sensitive and can sustain only a certain range of voltage and current. They tend to overheat if charged beyond a certain limit and explode if discharged beyond a certain voltage depending on the properties of the particular battery. Thus, we will use a particular charger which has been made and tested by spark fun. • We will use an oscilloscope (one terminal of oscilloscope on positive terminal of charger and other terminal of oscilloscope on negative terminal of the charger) to measure voltage changes across charger. • We cannot directly use an oscilloscope to measure electric current. So, we will attach a resistor of a known value to the oscilloscope's probes (one terminal of oscilloscope to one end of resistor and other terminal of oscilloscope to other terminal of resistor. The resistor itself is in parallel to the charger output terminal.). If we measure the voltage across the resistor, then we can use Ohm's law to calculate the electric current. ----------------------------------- • A fully discharged battery of power capacity 400mAh when charged	[0/100]

----------------------------------- 3. When battery of capacity 400mAh is charged to max capacity, the status led should go on. At this point, the current and voltage across the output terminals of the charger should also go to 0V and 0A respectively. 4. Unplug the battery from the charger when the led lights on.	from this charger (charges at rate of 400mA per hour) takes anywhere between 50 minutes to 70 minutes to charge fully (1.48Wh is maximum capacity of battery). • Now, if we probe the oscilloscope (one terminal of oscilloscope on positive terminal of charger and other terminal of oscilloscope on negative terminal of the charger) across the output terminals of the charger, the voltage should be 0V. • Use an ammeter (positive or red terminals of ammeter at positive output terminal of charger and negative or black terminal of ammeter at negative output terminal of charger) to measure the output current across the output terminals of the charger. This current should also be 0A. ----------------------------------- • The status led on the charger is a feature that lights up when the battery is fully charged. This is also an indication that the voltage and current to the battery from the charger is cut off. • LiPo batteries are very sensitive to voltage and current levels. Even though the charger cuts off current and voltage to the battery when the led lights up; there might still be current in nAmps still flowing to the battery. Thus, the led is a signal to unplug the battery from the charger.	

2. Polymer Lithium Ion Battery

Requirement	Verification	Points
1. When fully charged, must be able to output a voltage of 3.7 ± 0.2 V and should be able to deliver a maximum current of 400mA. ----------------------------------- 2. If all components of the circuit are functioning at the same time; the circuit should be drawing a maximum of 300 ± 20 mA. ----------------------------------- 3. A battery of power capacity 400mAh should be used with output 3.7 ± 0.2 V and a current of 400mA ± 20 mA. The battery should last 4 ± 1 hour of continuous usage with these specs.	• We will use an oscilloscope (one terminal of oscilloscope on positive terminal of battery and other terminal of oscilloscope on negative terminal of battery) to measure and graph the output voltage and current of the battery. • Connect the battery to a 10 ohm resistor, and use the oscilloscope (already connected as described in above point) to measure and graph the current output of the battery. • Thus current output should be close to 400mA to ensure that all the components will get required power when needed from the battery. ----------------------------------- • Load Arduino program on microcontroller with all the components to be always active by making them 'HIGH' in the loop function of the Arduino. • Use an oscilloscope (one terminal of oscilloscope on positive terminal of battery and other terminal of oscilloscope on negative terminal of battery) to measure and graph the output current of the battery. • Connect the pcb / circuit to the battery, this turns all components on. Now check the graph of current on the oscilloscope to check if the current drawn is 300± 20 mA and battery is meeting the maximum needed current requirement comfortably. -----------------------------------	[0/100]

	• Our circuit uses 100mA of current. This will last 4 ± 0.2 hours of continuous use with a 400mAh battery. With a little more efficient algorithm, we can improve this. Moreover, since we are aiming at a wearable, using a physically bigger battery is not feasible.	

3. UVLO

Requirement	Verification	Points
1. For an active (not charging) circuit with an input of 3.5 ± 0.4 V and 200 ± 200 mA; the output of the UVLO should be same as the input. 2. For an active (not charging) circuit with an input of 1.55 ± 1.55 V and 200 ± 200 mA; the output of the UVLO should be 0V and 0A. ----------------------------------- 3. For an inactive (charging) circuit with an input of 3.7 ± 0.2 V and 200 ± 200 mA; the output of the	• Use a function generator to supply 3.7 ± 0.2 V to the UVLO and slowly drop it by 0.1V every 5 seconds. Use an oscilloscope (one terminal of oscilloscope on positive output terminal of UVLO and other terminal of oscilloscope on negative output terminal of UVLO) to measure and graph voltage changes across the UVLO. • The oscilloscope should show the same output as the function generator for 3.5 ± 0.4 V; and 0V for the function generator at 1.55 ± 1.55 V. • Use 2 ammeters (positive of 1st ammeter on positive output of UVLO and negative of 1st ammeter on negative output of UVLO; similarly, positive of 2st ammeter on positive input of UVLO and negative of 2nd ammeter on negative input of UVLO). • Both ammeters should show same current values for function generator values 3.5 ± 0.4 V and the 1st ammeter should show 0.1 ± 0.1 mA when the function generator shows	[25/100]

UVLO should be same as the input. 4. For an inactive (charging) circuit with an input of 1.75 ± 1.75 V and 200 ± 200 mA; the output of the UVLO should be 0V and 0A.	1.55 ± 1.55 V. ----------------------------------- • Use a function generator to supply 1.75 ± 1.75 V to the UVLO and slowly raise it by 0.1V every 5 seconds. Use an oscilloscope (one terminal of oscilloscope on positive output terminal of UVLO and other terminal of oscilloscope on negative output terminal of UVLO) to measure and graph voltage changes across the UVLO. • The oscilloscope should show the same output as the function generator for 3.7 ± 0.2 V; and 0V for the function generator at 1.75 ± 1.75 V. • Use 2 ammeters (positive of 1st ammeter on positive output of UVLO and negative of 1st ammeter on negative output of UVLO; similarly, positive of 2st ammeter on positive input of UVLO and negative of 2nd ammeter on negative input of UVLO). • Both ammeters should show same current values for function generator values 3.7 ± 0.2 V and the 1st ammeter should show 0.1 ± 0.1 mA when the function generator shows 1.75 ± 1.75 V.	

4. Buck converter

Requirement	Verification	Points
1. For input voltage between 3.5 ± 0.4 V; the buck boost should output a voltage of 3.3 ± 0.05 V and the same as the input	• Use a function generator to supply 3.5 ± 0.4 V to the Buck. Use an oscilloscope (one terminal of oscilloscope on positive output	[25/100]

current. ----------------------------------- 2. When all components are functioning, the Buck-Boost must be able to output a voltage of 3.3 ± 0.03 V and should be able to deliver a maximum current of 400mA.	terminal of Buck-Boost and other terminal of oscilloscope on negative output terminal of Buck-Boost) to measure and graph voltage changes across the Buck-Boost. • The oscilloscope should show 3.3 ± 0.03 V output on the oscilloscope. • Use 2 ammeters (positive of 1st ammeter on positive output of Buck-Boost and negative of 1st ammeter on negative output of Buck-Boost; similarly, positive of 2st ammeter on positive input of Buck-Boost and negative of 2nd ammeter on negative input of Buck-Boost). • Both ammeters should show same current values for function generator values 3.5 ± 0.4 V. ----------------------------------- • Load arduino program on microcontroller with all the components to be always active by making them 'HIGH' in the loop function of the arduino. • Connect a resistor (10.5 to 11.5 ohm) in parallel to output of Buck-Boost and check current with an ammeter (positive terminal of ammeter at positive output terminal of Buck-Boost and negative terminal of ammeter at negative output terminal of Buck-Boost). The output current should be about 380 ± 20 mA. • Now connect 4 more registers (all in parallel) of the same value between 10.5 and 11.5 ohm to output of Buck-Boost. They all should be drawing 100 ± 0.1 mA of current as seen on the ammeter. This ensures the maximum current we need for our circuit is available.	

5. **Accelerometer**

Requirement	Verification	Points
1. Maximum current input should be 10 ± 2 mA when active. ----------------------------------- 2. Should have steady output without variation of more than 2 degrees when placed in a steady position for more than 1 second.	• Load program on microcontroller to keep accelerometer always on at 20 Hz. • Use an ammeter (positive terminal of ammeter at positive input terminal of accelerometer(Vcc) and negative terminal of ammeter at negative input terminal of accelerometer(GND)) to measure the input current across the terminals of the accelerometer. This current should also be 10 ± 2 mA. ----------------------------------- • Power the accelerometer and connect it to an arduino so that values are printed on the serial monitor of the arduino. • Place accelerometer on a flat surface and see 0 ± 1 degrees on the serial monitor. Now place accelerometer at 10, 20, ...90, 100, ….180 degrees with the help of a protractor and see a value accurate within 2 degrees on the serial monitor. • Now place accelerometer at 180+, 190, 200, ….360 degrees with the help of a protractor and see negative values corresponding to 170, 160, ….0 degrees accurate within 2 degrees on the serial monitor.	[12/100]

6. Gyroscope

Requirement	Verification	Points
1. Maximum current input should be 10 ± 2 mA. ----------------------------------- 2. Should have steady output without variation of more than 2 degrees when placed in a steady position for more than 1 second.	• Load program on microcontroller to keep gyroscope always on at 20 Hz. • Use an ammeter (positive terminal of ammeter at positive input terminal of gyroscope(Vcc) and negative terminal of ammeter at negative input terminal of gyroscope(GND)) to measure the input current across the terminals of the gyroscope. This current should also be 10 ± 2 mA. ----------------------------------- • Power the gyroscope and connect it to an arduino so that values are printed on the serial monitor of the arduino. • Place gyroscope on a flat surface and see 0 ± 1 degrees on the serial monitor. Now place gyroscope at 10, 20, ...90, 100, ….180 degrees with the help of a protractor and see a value accurate within 2 degrees on the serial monitor. • Now place gyroscope at 180+, 190, 200, ….360 degrees with the help of a protractor and see negative values corresponding to 170, 160, ….0 degrees accurate within 2 degrees on the serial monitor.	[12/100]

7. Bluetooth module

Requirement	Verification	Points
1. Maximum current input should be 80 ± 20 mA when active. ----------------------------------- 2. Bluetooth should be able to pair with a computer/phone at least 5 meters away. It should recognize a device and automatically reconnect with it when it is looking for potential devices to connect with. ----------------------------------- 3. Bluetooth should be able to send angle values to the computer which reflect as continuously updating graphs on the computer.	• Load program on microcontroller to keep bluetooth always on and transmitting data. • Use an ammeter (positive terminal of ammeter at positive input terminal of bluetooth(Vcc) and negative terminal of ammeter at negative input terminal of bluetooth(GND)) to measure the input current across the terminals of the bluetooth. This current should also be 80 ± 20 mA. • There are huge variations in bluetooth power consumption because of various modes like finding a device, connecting to device, transmitting data to a device, and sleeping momentarily when no data to transmit. ----------------------------------- • Power the bluetooth module with 3.3 ± 0.03 V and also ground it with pins 1 and 10 respectively. At a distance of 5 ± 2 meters away from the bluetooth module, a computer should be able to establish a connection. • Now take this computer more than 25 meters away from the bluetooth module (range is 20 ± 2 meters) so that it disconnects from the bluetooth module. Bring it back in range and it should start receiving data immediately. ----------------------------------- • Program microcontroller to send certain predefined values to the bluetooth device through TXD pins	[12/100]

	which the bluetooth sends to the connected computer. • These values should be seen on the computer where they are processed and data is represented as continuously updating graphs.	

8. Vibration motor

Requirement	Verification	Points
1. Maximum current input should be 80 ± 20 mA when active and 2 ± 2 mA when inactive. ----------------------------------- 2. When the accelerometer values cross a predefined thresholds (in our test 30 degrees in the x axis) in the microcontroller; the microcontroller sends a HIGH signal to motor and the motor vibrates to indicate bad posture. ----------------------------------- 3. An inductor in the motor charges when motor is turned on and current can spike to 90 ± 20 mA.	• Program the microcontroller to send a constant high to the digital pin controlling the vibration motor. • Probe the motor using a ammeter. (Positive terminal of ammeter to positive terminal(red wire) of motor and negative terminal of ammeter to negative terminal(black wire) of motor). The ammeter should show an input current of 80 ± 20 mA. • Now, program the microcontroller to send a constant low to the digital pin controlling the vibration motor. • Probe the motor using a ammeter. (Positive terminal of ammeter to positive terminal(red wire) of motor and negative terminal of ammeter to negative terminal(black wire) of motor). The ammeter should show an input current of 2 ± 2 mA. ----------------------------------- • Write an arduino program to turn digital pin high when accelerometer angles crosses for 5 seconds and then low for 5 seconds. Record the time of	[4/100]

4. Sudden current and voltage fluctuations can affect performance and can introduce a time lag which should not be observable.	vibration of the motor with a hand clock to see if the vibration actually lasts for 5 seconds. ---------------------------------- • Connect the oscilloscope across the motor (one terminal of oscilloscope to one end of the motor and other terminal of oscilloscope to other end of the motor) and plot the voltage and current graph. • See if the voltage and current vs time varies as the graph of an inductor would and then stabilizes. • Also, see if highest voltage and current spikes remain the same always to ensure the motor fluctuation is predictable. This is important to know if the motor works correctly and does not have time delays.	

9. Microcontroller

Requirement	Verification	Points
1. For input voltage as 3.5 ± 0.4 V; all pins of microcontroller should be able to output 3.3 ± 0.03 V.	• Upload arduino code to set all pins to high and load it on microcontroller. • Supply 3.3 ± 0.03 V to microcontroller and with an ammeter(positive of ammeter on pins of microcontroller and negative of ammeter on ground of microcontroller), and check if the output of each pin of microcontroller gives 3.3 ± 0.03 V, which is a digital high. • Now upload arduino program to set	[10/100]

----------------------------------- 2. The microcontroller should be able to get values from the sensors (accelerometer and gyroscope), calculate the angles, and output it to the bluetooth module. 3. Thus, microcontroller should have enough computation and memory to hold all of this and receiver / transmit data via I2C communication, and analog pins.	all pins to low and load it on microcontroller. • Supply 3.3 ± 0.03 V to microcontroller and with an ammeter(positive of ammeter on pins of microcontroller and negative of ammeter on ground of microcontroller), check if the output of each of the pin of microcontroller gives 1.15 ± 1.15 V, which is a digital low. • Now upload arduino program to make a few randomly selected digital pins high and the remaining digital pins low. • Supply 3.3 ± 0.03 V to microcontroller and with an ammeter(positive of ammeter on pins of microcontroller and negative of ammeter on ground of microcontroller), check if the digital pins are high or low as written in the program. ----------------------------------- • The Atmega328p has 32 kBytes Flash memory, 14 digital pins for I2C communication (we need 3 pins), and operates at 20MHz (we need about 20 Hz). • Thus, this chip is more than sufficient for our use; and since it came with the arduino, we used the Atmega328p.	