

“LIFE” Jacket

Final Report

May 12, 2015

[Team 7]

Elliott Tan

Andy Choi

Cindy Lam

Minn Tun

Abstract

Our project focuses on building a wearable embedded system that helps the users to track their sitting posture and receive feedback of its correctness in real-time. By creating custom PCBs that can hold various motion sensors and can attach to any existing jacket, we were able to monitor the user's posture based on our pre-trained machine learning algorithm and provide tactile feedback whenever the user was in an unhealthy state. Among the five or more different types of posture that the algorithm could identify, the user would be notified about which posture he or she is currently in so that the user could sit up straight until the posture was deemed correct. We also provide an Android application that can receive the posture data from the jacket wirelessly and show visual feedback of how the user is sitting wrong along with the history of all past postures that the user had been in. In the end, we believe that our solution will be able to help anyone who has a need of fixing his sitting posture and remain in a healthy state.

Table of Contents

1. Project Description
2. Design Requirements
3. Functional Architecture
4. Design Trade Studies
5. System Description/Depiction
6. Project Management
 - a. Schedule
 - b. Team Member Responsibilities
 - c. Budget
 - d. Risk Management
7. Conclusions
 - a. Lessons Learned
 - b. What Would've Been Done Differently
 - c. Future Work
8. Related Work
9. References

1. Project Description

Poor posture has negative effects in people's lives in many ways. There are several medical issues that stem from poor posture. For example, the lungs are unable to expand fully, resulting in less oxygen flow in the body. Also, prolonged poor posture can result in back, shoulder, and neck pain. Posture is an issue that must constantly be monitored and adjusted. Most people, however, tend to forget about their posture, settling into poor posture for extended periods of time. Our project is motivated by our realization that many people with bad posture do not realize that they have bad posture. Based on these facts, we've aimed to create a product that would help those people correct and improve their posture without intruding on daily activities. Therefore, we came up with the idea to create a posture monitoring jacket that would subtly alert the wearer when posture is poor.

The "LIFE" Jacket is a wearable tech jacket that helps detect and correct your posture in real time. Correcting posture is important amongst the early youth to prevent some posture related health complications. The system consists of a removable hardware module that can be placed onto any existing jacket. Also, the Jacket is able to function for any typical amount of time a person spends on the desk and can operate on rechargeable battery. Accelerometers and gyroscopes are used for movement detection, while vibrating motors will allow wearers to receive feedback on bad posture. The jacket is designed to give feedback on postures that are primarily focused on stationary sitting postures. The jacket also has a bluetooth low energy module and an Android application to allow the wearer to view current posture and history of his posture data on a mobile device.

2. Design Requirements

Explicit Requirements

- Hardware
 - For each sub-module that is in charge of detecting the user's motion, the minimum requirement is to have at least one accelerometer and one gyroscope. Any one of the sub-module is also required to have a vibrating motor for providing tactile feedback to the user.
 - For the main-module that will process the sensor data and send the result over Bluetooth, a microprocessor, a BLE module, and a power source is required.
- Software

- All software that will power the hardware on the jacket will be programmed in c/c++ with libraries from Arduino and BlueCreation. The smartphone application will be based on Android and will be built using the latest Android SDK.
- Functional
 - Our design shall allow for accurate posture detection, and notifying the wearer of the areas that are out of alignment. Also, the data shall be stored locally if there is no mobile phone paired with the jacket. Once the jacket is paired with a phone, the stored data and any new data will be transmitted via BLE. The mobile application should be visually appealing and straightforward to use.

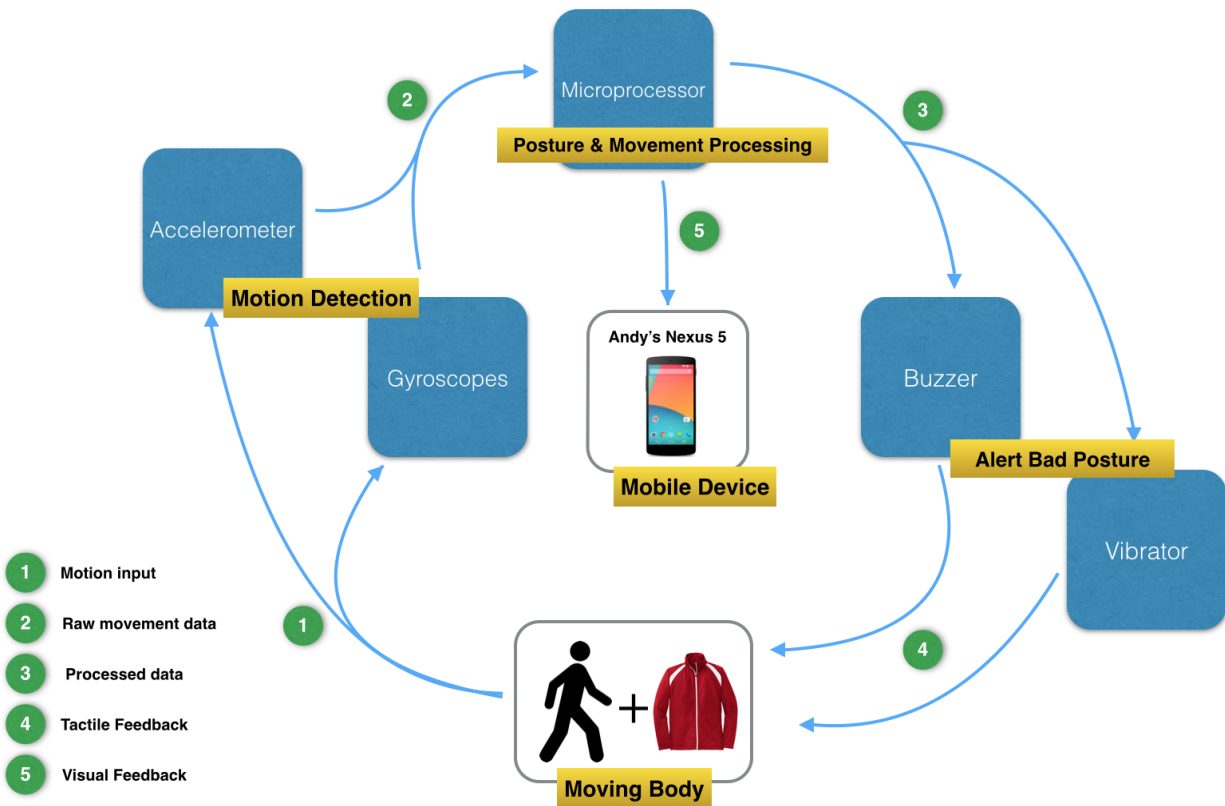
Implied Requirements

- Our design requires us to calibrate and store data from calibration regarding estimated thresholds of posture movement detection.
- We need the user to pose in different postures before using the system so that the calibration is correctly performed for the user.

Aesthetic Requirements

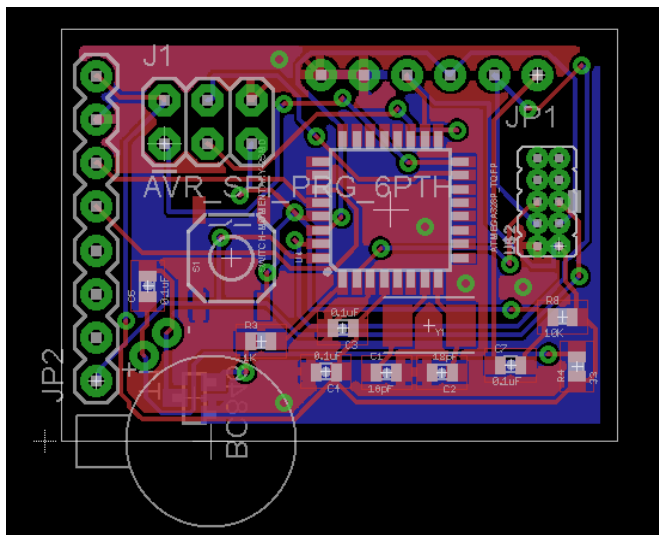
- We need the user to be wearing any lightweight sports jacket with our custom pcb modules that are designed to be small and flat and lay comfortably on a person's back.
- We need the jacket to have enough pouches or pockets if the user wants to have the custom hardware unnoticeable from other people.

3. Functional Architecture

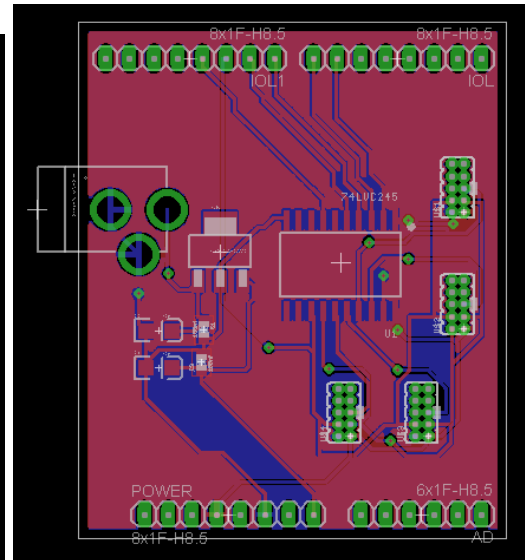


[Figure 3.1] High Level Architecture of Proposed System

Figure 3.1 depicts our functional architecture for our project. The wearer will don the jacket and then continue to move in daily life. The Life-Jacket will have a set of motion detection modules that would measure the movements of the wearer. The raw data from each of the motion detection modules will be fed into the Microprocessor for it to start measuring the user's posture data and then alert the user in case an incorrect posture is detected. The feedback system will include vibration motors, and an Android phone application.



[Figure 3.2] Submodule PCB



[Figure 3.3] Main Module PCB

To go further in depth on the functional architecture, we decided on two different modules each with their own processors. First we have a submodule, as seen in Figure 3.2, which is a dedicated PCB with the gyro, accelerometer and motor attached to an atmega. The module has built in connectors which can readily be attached to the main module. This module would be powered by 3V to have longer battery life.

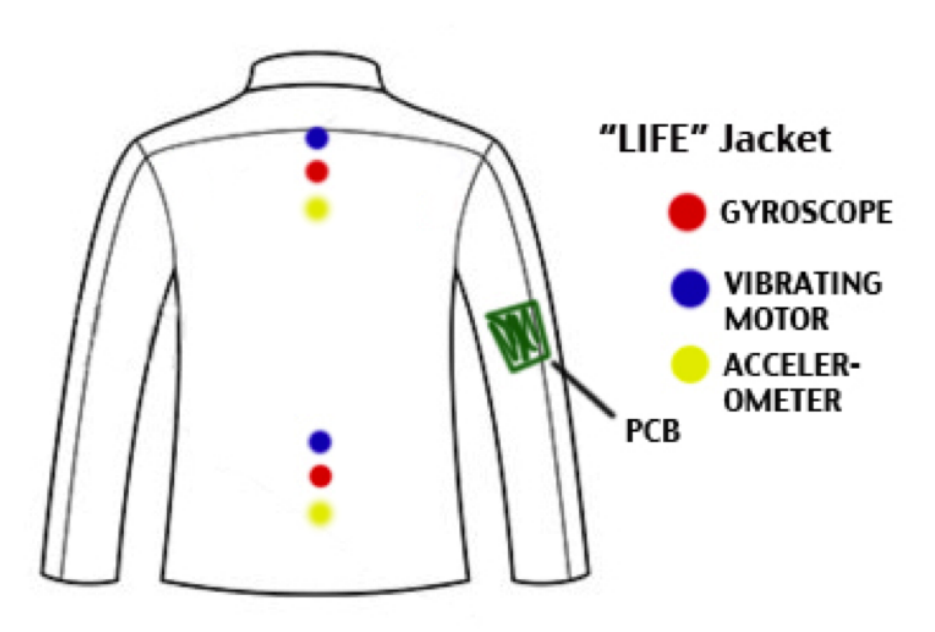
The main module is an Arduino Yun which has a more powerful processor to run ML scripts. On top of the Yun sits another custom PCB, seen in Figure 3.3, which connects required connections (mainly power, gnd and serial) into a automatic bidirectional logic level shifter due to the shift from 5V powered Yun to the 3V powered submodule. Then these connections connect to four independent connectors to form a single master/multiple slave network allowing the main module to communicate with each submodule (designed for up to four). These built in connectors allow greater modularity in our design and usage.

4. Design Trade Studies

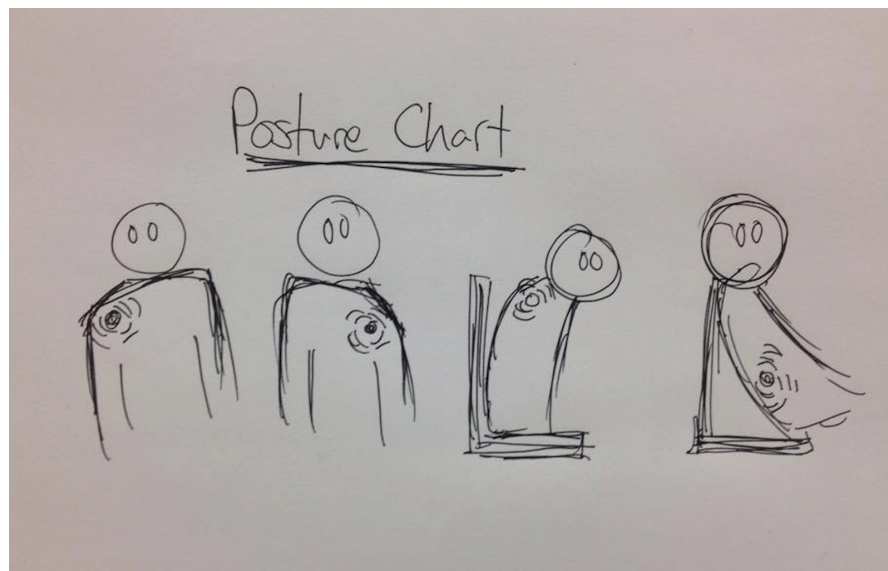
1) Number and Location of Gyroscopes

The very first design choice and the most important aspect was on what components we were going to use to enable the functionality of our product, or specifically, how our gyroscope placements will impact posture detection. We decided that in order to detect different postures, we would need to place four gyroscopes on four different parts of the jacket (e.g. left shoulder, right shoulder, lower spine, upper spine) as shown in Figure 5.1. Though having all four would be optimal for identifying postures more accurately, we also thought that having at least two

would be enough for identifying several distinct postures. These sensors will provide unique data of each body part to help us determine different bad posture positions in real time (e.g. left shoulder lean, right shoulder lean, hunch forward, hunch back) as shown in Figure 5.2. Using anymore than four might become redundant as these body part locations are the places that give enough data to determine what your current posture is.



[Figure 4.1] Sensor & Actuator Positions



[Figure 4.2] Different postures detected

2) Requirement of Separate Stand-alone Accelerometer

A second major design choice we had to make was whether to include a separate stand-alone accelerometer in our design. The benefits of including this separate accelerometer was that our product would be able to tell whether the person was in motion or stationary. This would allow the jacket to filter out potential incorrect posture alerts while the person was in motion. Also, the system would be able to start real time posture detection only when the person was stationary and sitting. We could also set a stretch goal of having our product provide an additional functionality of basic activity tracking (e.g. being able to differentiate between sitting, walking, and running). However, we decided not to have this accelerometer due to resource and time constraints.

3) Choice of Actuator (i.e. type of feedback)

Once an incorrect posture is detected during real time posture monitoring, we needed to somehow alert the user of this. Our main choices for the actuator were an led (a visual feedback type which lights up on a posture flag), a buzzer (an audible feedback type which beeps), and a vibrator motor (a tactile feedback type which vibrates when the posture is identified to be incorrect). We decided to stick with the vibrator motors that would be positioned at any location where the gyroscopes would be, as shown from Figure 5.1. Our reason for choosing this type of feedback method was that audio feedback through a buzzer disturbs those nearby and decreases the functionality of our jacket to be something that can be worn at anytime and anywhere.

4) Wireless vs. Wired Connection Among Components

The decision of having components, specifically the sensors and actuators, connect to each other either wired or wirelessly to the main PCB's microcontroller was an easy decision to make. The benefits of making these components wirelessly connected is ease of access, better aesthetics, and modularity of these components. However, the costs outweigh the benefits of this design. Running wires through the jacket was thought to be a doable job without sacrificing aesthetics. A second reason for choosing on having the components wired to each other was the amount of power needed to power the communication between the components. We felt that power capacity would be one of the big issues we would have to face in our design, and using wires instead of wireless was believed to provide better power efficiency.

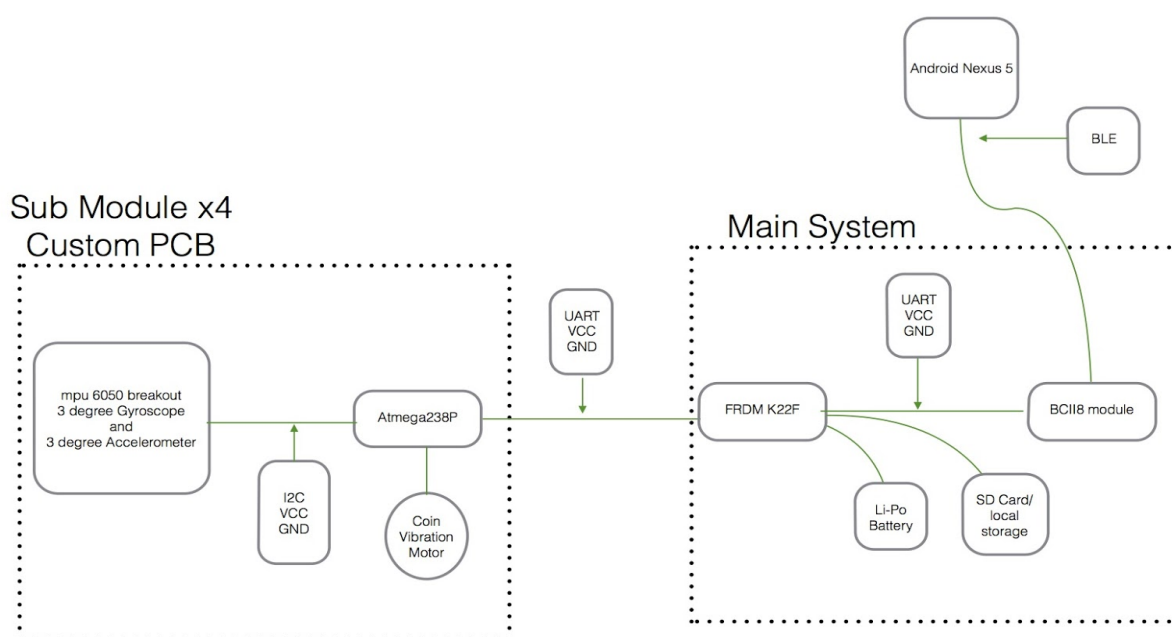
5) Conductive Thread vs. Ordinary Wires

Conductive threads are much thinner than ordinary wires and are easily sewn into wearables. But because we are using lithium polymer ion battery as a power source, a short in the thread could result in a spark or heat. Thus we decided to use ordinary wires, which has less of a form factor, for initial testing but conductive thread for our finalized product if possible.

6) Wireless vs. Wired Data Communication

We decided to provide a smartphone application that could provide extra features and have our device communicate data to it through BLE. However, this posed a dilemma as to what data we would be transmitting and how frequently this will be done, as additional communication would have an impact on battery life. However, we found that a well-controlled BLE module would not drain the battery too much, and having the user to frequently connect the smartphone to the jacket through wires did not seem to provide good user experience.

5. System Description/Depiction



[Figure 5.1] Overall System Diagram with Technical Specifications

○ Training Data Collection

In our design, a button would be provided to the end user for initiating or ending the calibration stage. However, we did not get to implement this piece of hardware. The current system requires the user to be connected to the main processor (Arduino Yun) via WiFi. The calibration stage was required in order to accommodate the different physical properties that depends on each system's user. Since people have various sitting postures and have different interpretations of what a correct sitting posture is, one single criteria

for processing the collected sensor data might be accurate for one particular person but not for the others. Thus, we decided to include a calibration stage whenever the user of the system changes.

The calibration stage starts when the Arduino Yun runs a data collection script. Once the calibration stage starts, the user should be sitting in a posture that he/she thinks is the correct way of sitting. During this stage, data from the sensor system will be collected and sent over to the data processing system via serial communication so that the system can label them as being correct. A similar process is done for each of the different postures. This data is stored in a CSV file, which then will be fed as a training set into a machine learning algorithm using logistic regression so that the data processing system can derive a decision function that will be used for labeling future data.

After the user believes that enough data has been collected, the user should terminate the training data collection script. When actually running the main code, the sensor data that will be provided to the processing system from now on will act as raw data that the machine learning algorithm will have to label using logistic regression. From this point, whenever the algorithm labels a stream of data to be incorrect, it will trigger the feedback system to alert the user of his/her incorrect posture.

- Posture Detection System

Our posture detection system consists of sensor submodules that each have an ATmega328P microcontroller on a custom PCB. Each submodule has a MPU-6050 triple axis accelerometer and gyroscope breakout board connected via I2C to the ATmega328P. Since this sensor is a accelerometer and gyroscope combination. Each submodule will have six data points about the orientation of the

The four gyroscopes will also be sending three axis data to the microprocessor on each of the gyroscope's current orientation. Because we are sewing the gyroscopes to become attached to the jacket itself, the only way we would see changes in orientation of each gyro is through changes in posture which is the exact data that we are looking for. This data will also be sent through 3 channel multi bit communication for each gyroscope. All these data will be constantly collected and in slow real time be sent to the microprocessor at a constant rate. Doing so, the microprocessor can keep track of the person's current "posture" and based on the state of either in motion or stationary, output feedback based on this received data.

- Data Processing System

The Arduino Yun will handle all the processing of the data received from the posture detection system. The microcontroller collects the six data points from each of the sensor submodules and classifies them against the set of training data using logistic regression. Depending on the classification of the input data, the user's posture will be

determined. A signal will be sent to the sub-module that has a vibration motor attached to it so that it can activate the motor and alert the user.

- Feedback System

The feedback system is made up of vibration motors on each of the submodules. After the Arduino Yun has processed the incoming orientation data of the sensor submodules, it decides on if the user is currently in bad back posture. If the user is in a bad posture, a message is sent over serial communication to the submodule to vibrate the motor for one second. The ATmega328P on the submodule then outputs a voltage to the corresponding vibrating motor to alert the user of the position of bad posture.

- BLE System

Our device is also capable of syncing up to a phone, which will receive locally logged posture data and visually represent the data. After the Arduino Yun runs the machine learning algorithm and classifies the raw data into one of five postures, the classification is sent over serial communication to the BLE Mate 2. The BLE Mate 2 is a breakout board which houses the BC118 chip, which forwards the data over BLE to a connected smartphone if one is connected. The smartphone app will then process this data into visual feedback to the user.

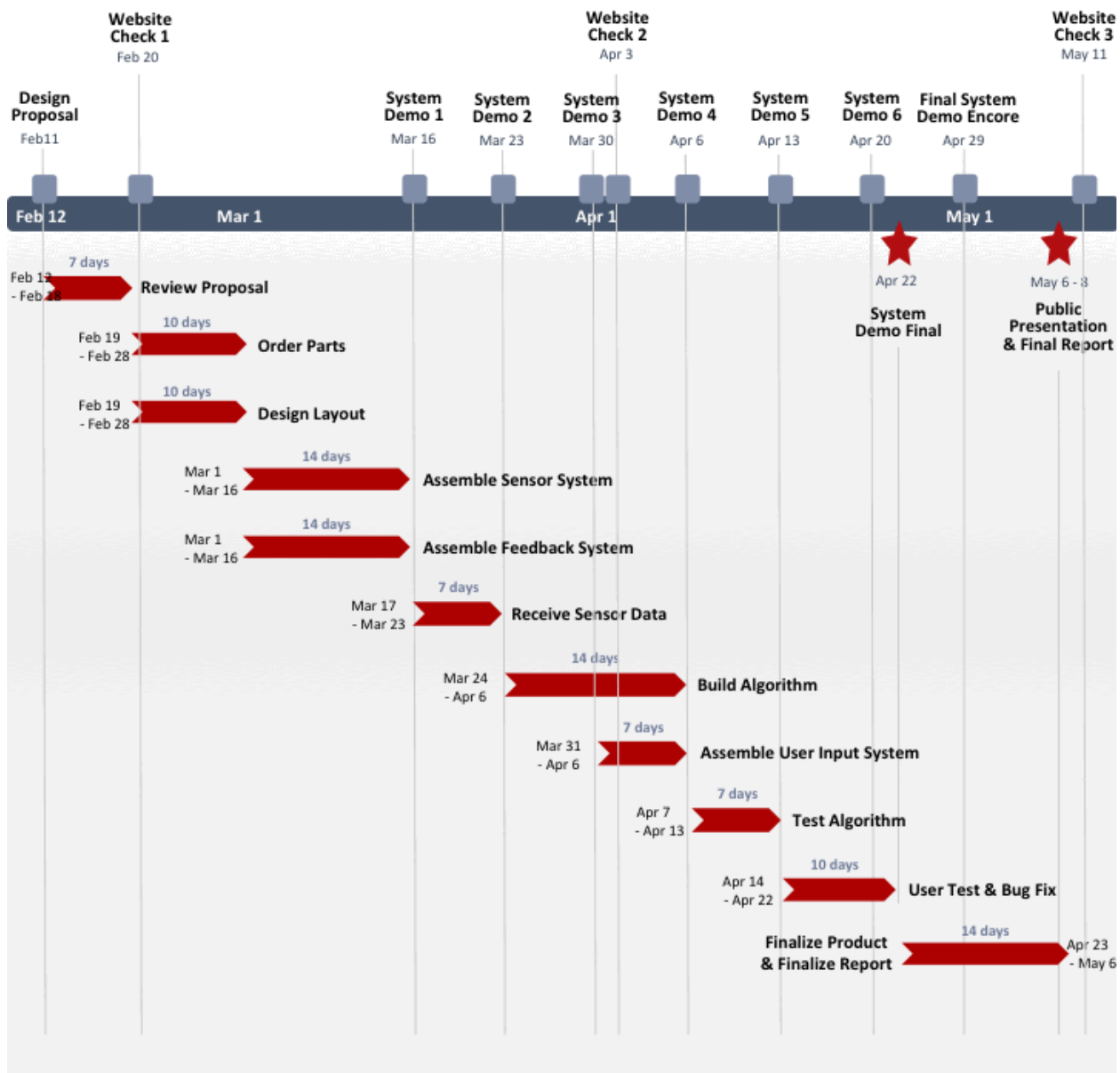
- Power System

Our main microprocessor, the Arduino Yun, is powered using a rechargeable battery that connects via micro USB. This battery supplies 5V with a power of 4000mAh. Each of the sensor submodules has is powered by a wire in the connector which supplies 3.3V that comes from the main Arduino Yun.

6. Project Management

a. Schedule

- Semester Schedule



[Figure 6.1] Gantt Chart for Semester Schedule

- Website Schedule

Checkpoints	Tasks	Deadline
Checkpoint 1	Basic team info / Concept / Competitive analysis / Motivation / Requirements / Tech specs (components)	Feb 20
Checkpoint 2	Architecture / Use cases / Interaction diagrams	Apr 3
Checkpoint 3	All documents (slides, diagrams, etc.) / Media (videos, etc.)	May 12

[Table 6.1] Website Schedule and Corresponding Tasks

- Team Member Responsibilities

Name	Primary Responsibility	Secondary Responsibility
Cindy Lam	Developing the machine learning software for data processing	Testing the validity of the input data and labeling training data
Andy Choi	Developing the data communication software for sensor and feedback IO	Designing the protocol of the communications among components
Minn Tun	Designing the layout of the components on the physical product	Wiring the physical components and perform physical tests
Elliott Tan	Developing the core software of the microprocessor	Documenting the project history and managing the project website

[Table 6.2] Primary and Secondary Responsibilities per Member

- Budget

Part	Description	Quantity	Cost
ATmega328P TQFP	Microcontroller	2	\$7.62
BLE Mate 2	Microcontroller + Bluetooth LE	1	\$29.95
USB Battery Pack	Power Source	1	\$14.95
Arduino Yun	Microprocessor	1	\$71.95
Kootek MPU-6050 Module	Accelerometer/Gyroscope	2	\$11.98

10mm Vibration Motor	Tactile Feedback	4	\$15.24
74LVC245 (SMD)	Logic Level Shifter	2	\$1.04
Crystal TXC 7A (8 MHz)	Crystal Oscillator	2	\$1.58
SanDisk 8GB MicroSD	Local Data Storage	1	\$6.00
Jumper Wires Standard 7"	Material for Prototype	1 pack of 30	\$4.95
5 Pin Cables	Connectors	2	\$19.68
Jacket	Material for Prototype	1	\$20.00
Custom PCB (Submodule)	Design and Construction Cost	4	\$50.00
Custom PCB (Main Module)	Design and Construction Cost	1	\$50.00
		Total	\$304.94

[Table 6.3] Parts List with Description and Cost

d. Risk Management

Potential Risks

- Since the product has to integrate into fabric, the layout of the wires might serve as the critical point from which the product's durability is determined.
- As many people can behave differently or have different standards regarding what a correct posture is, the machine learning algorithm might require more time to be developed in order to accommodate various features test cases.
- Depending on the team's experience with hardware, the board design or component requirements might change unintentionally and thus would require more time to meet the proposed deadline.

Fallback Designs

- Use actual wires instead of conductive thread and have them over-stitched so that the amount of the surface that the wires get exposed to is minimized.
- Decrease the size of the product to reduce the amount of variables that can stem from a real environment (e.g. vest to belt), which would help decrease the number of features needed for the algorithm.
- Reduce the amount of feedbacks that are given to the end user (e.g. buzzer & vibrator to buzzer only, fewer sets of sensors, etc.) to make the overall product design more simplified and achievable by the deadline.

7. Conclusions

a. Lessons Learned

- We learned that in case UART was to be used, we had to be careful about how many UART ports would be available based on our board decision. In case of our system, having only one UART port required us to create virtual UART ports through Arduino's SoftwareSerial library, and even with this, there could be only one software UART port working at a time.
- In terms of hardware, designing the pcb to be error free was something difficult. We had issues with clock speeds (due to running atmegas at different voltages) and logic levels. Parts were also mistakenly chosen such as having a bidirectional logic level shifter which was not automatic when our design required bidirectional automatic. A lot of these issues had to be bypassed manually such as soldering wires to the PCB or connecting a second logic level shifter on a breadboard.

b. What Would've Been Done Differently

- Due to the fact that most Arduinos usually have only one UART interface, we probably might have chosen I2C or SPI interfaces over UART if we needed to have one module talking to other modules simultaneously again.
- Since the machine learning algorithm provided from the library required a long time for the training phase to complete, we could've tried implementing our own version of the algorithm to see if any optimizations were available.
- Since all of the modules are placed on the back of the jacket, it is currently inconvenient for the user to lean back on a chair, since the modules would bulk out and poke the user. We would probably locate the modules elsewhere so that the modules won't intervene with any normal sitting posture.



c. Future Work

- Since the data produced while the jacket and the smartphone are not connected is discarded, we would probably have the data stored locally and have it sent later when the two connect to each other. In order to do this, the microcontroller will first determine if the bluetooth system is paired with the mobile phone. If there is no active connection, the microcontroller will store the data to an external SD card. Once the microcontroller detects that the bluetooth system is connected to the phone, it will send the logged data from the SD card to the bluetooth system for transmission before sending out any new data.
- Since the current system only has two sub-modules communicating with the main-module, we could increase the total number of sub-modules so that more data can be collected for better posture detection precision.

8. Related Work

Currently, the only similar commercial products are produced by Lumo. One product that they produce is the Lumo Back. This is a device that you can wear around your lower back to detect your posture. It also vibrates when it detects that you are slouching. This band may be uncomfortable for the user, especially when sitting down against a chair where the device presses against the back. Additionally, since the band is pressing against your body, issues with heat and sweat could arise in that area. Another available product is the Lumo Lift, which is a magnetic clasp that you attach to your clothing on your upper chest.

The advantage that our product has over these two devices is that the entire system is integrated into a jacket. Users do not need to worry about putting on additional devices. They simply put on one article of clothing. Also, our product can determine which parts of the back are out of alignment. The feedback from the device can be programmed to vibrate at specific locations, indicating to the user where they need to adjust their posture.

Lumo Back	Lumo Lift
	

[Table 8.1] Pictures of Primary Competitors

9. References

- <http://www.lumobodytech.com/>
- <http://www.lumobodytech.com/lumoback/>
- Dogărescu, Octavian and Sever Pașca. “Abnormal Posture Detection Device with Audible Feedback.” *Advanced Topics in Electrical Engineering*: 2011.
- Mattmann, Corinne. “Body Posture Detection Using Strain Sensitive Clothing.” 2008.

- <https://punchthrough.com/bean/>