Fun Times Part 4 / SmartBell

XiaoChuan Ding, Hamza Kakeh, Nathan Park, Daniel Wu, Kevin Zheng

12/16/2019

Capstone Design ECE 4440 / ECE4991

Signatures

XiaoChuan Ding

Hamza Kakeh

Nathan Park

Daniel Wu

Kevin S. Zheng

Statement of Work:

David Ding

David was in charge of interfacing with the SmartBell's LCD screen. The LCD screen is an important part of the project for two reasons. First, it is used as part of the debugging process to output variable and information without the need to pause execution of the program. Second, the LCD screen is used as the interface for the user of the SmartBell, displaying information such as the number of reps completed, the number of bad reps completely, the number of calories burned, or the time between or during sets to name a few. It was essential this was completed quickly early on. In addition, modification was needed after board designs were changed to enable the use of more pins.

David was also in charge of researching how to count calories and the code to determine the calorie count. Recent gym equipment has become smarter and one common feature included is the calories burned per exercise. The group wanted to mimic this recent trend, so the SmartBell includes the ability to check how many calories the user has burned.

Hamza Kakeh

Hamza was in charge of interfacing with the inertial measurement unit (IMU) which included a gyroscope and accelerometer. The IMU can be considered the heart of the SmartBell as it is used to determine when a repetition is complete and whether it was completed with good form. Hamza had to write the SPI code as hardware SPI was not enabled with the wiring. He also figured out how to set up the IMU so that the different contained devices operated at the desired frequency and range. This was part of the midterm review and was completed on time.

Hamza was also in charge for the CAD design of the outer casing. He worked through multiple iterations and in the end achieved a product that was very compact, fit tightly with the PCB board, and could snap closed. The SmartBell needed some sort of casing so that it could be attached to a barbell as well as to protect the PCB components.

Nathan Park

Nathan was in charge of interfacing with the array of LEDs. The array of LEDs is important to the project because it was the user's primary feedback during the actual exercise. In addition, it was also used as another debugging tool for the group. For the user, if the user is exerting too much force on one side compared to the other, the array of LEDs would light up a specific pattern so that the user to correct his or her form. Nathan needed to figure out how to get I2C communication working between the MSP430 and the array of LEDs. This was achieved through hardware I2C.

Nathan was also in charge note taking, reaching deadlines, creating progress slides, scheduling, and primary report writing. This was important so that all the checkpoints were reached, and something was submitted in Collab's assignment page. Also, in case group member missed something important, it was his job to catch them back up on what was happened. He provided the organization that the group needed.

Daniel Wu

Daniel was in charge of implementing the algorithm for counting repetitions and determining bad form. This was the most software intensive portion of the project. Daniel had to experiment with several different algorithms in order to get the repetition counting to work

before the final demo. This was important to the project because counting repetitions, good or bad, is arguably the most important feedback the user receives during the exercise.

Daniel also played a major role in parts research and selection. He helped select parts that were right for this project. He was also in charge of wiring up all of the components in Multisim. This was essential as this was needed for the board layout.

Kevin Zheng

Kevin was in charge of code integration. Much of the software was written completely separate from one another. Kevin went through all other group members' code and put it all together making sure everything functioned after the code integration. Kevin also was very good and helpful at debugging other bugs in the code. This was important to the project because all the SmartBell's different features needed to work at the same time.

Kevin was also in charge of the Ultiboard design as well as part selections and ordering. The board had to be extremely compact and dealt with a variety of surface mount items that were on the scale of millimeters. This was important as the SmartBell should not be intrusive while doing the workout. This required multiple iterations before achieving a design that was roughly a third of the original size. Making sure the right parts were chosen was also important to ensure that integration was possible.

Table of Contents

Table of Figures

Abstract

The SmartBell is a device to aid both the beginner and advanced weightlifter in their barbell exercises. The SmartBell is to be attached to any barbell and provide feedback for common barbell exercises, such as deadlifts, shoulder presses, bench press, and many more. The SmartBell provides feedback by detecting correct repetitions and incorrect repetitions, counting calories, counting average rest time in between sets, and notifying the user if the barbell is parallel to the floor. These specific features are included in the SmartBell because knowing this information can help the weightlifter have a more efficient, productive, and safer workout. The SmartBell has a small LCD screen and an array of LEDs to display the aforementioned forms of feedback. The SmartBell also allows for user input to determine how many repetitions per set and sets the user plans on completing. This will allow for computations such as average rest time between sets. All this information cannot be shown at once on the LCD screen, so the user will have to navigate between displays using two buttons right next to the LCD screen.

Background

Current gym equipment is, and looks, more advanced than equipment in the past. For example, the first ever treadmill had very limited features and looked like an industrial conveyor belt with railings [1]. After many years, iterations, and market competition, the treadmill has evolved to what we have today. Recent variations have included a big digital screen allowing the user to navigate to different workouts and settings. Some notable statistics given as feedback by treadmills are the runner's pace, calories burned, distance ran, and amount of time ran [2]. Other gym equipment have followed this trend of evolving and providing extra feedback for its users. The first reason Fun Times Part 4 chose to create the SmartBell was because the current ways of

providing feedback are too expensive for barbell exercises, which will be further discussed below.

The second reason Fun Times Part 4 chose to create the SmartBell is to prevent injuries. Back in 2009, One thousand and five hundred people were treated in the emergency room due to gym related activities [3]. One way of injury is if the weightlifter has muscle imbalances [4]. For clarification, muscle imbalances occur when the user asserts more pressure on either side of the lift, thus causing the barbell to tilt. Fun Times Part 4 wanted to create a device that notifies the user of any tilt during the lift, so that the user can correct their form and lower the probability of injury.

The most common technology for barbell form correction is the Smith machine. It was invented in the 1950's and is still popular to this day due to its stabilized vertical movement and incorporated safety measures [5]. Although, the Smith machine has many positive attributes, it is expensive, and gyms usually don't have many Smith machines due to their space requirements.

There have been three devices, two in market and one in the pre-sale phase, that provide similar feedback as the SmartBell. The first of these devices is called the BaziFit [6] which costs $199. This is the one that is in the pre-sale phase and has not been released yet. The device has the capability of attaching to almost any workout equipment and providing useful statistics to the user. The device claims to be catered toward barbell, balance, and suspension exercises. It also possesses the ability to connect to a mobile device via bluetooth and record the user's metrics there. The device looks like a relatively small square that lights up to different colors during the progression of the exercise.

The second device is called thisisbeast which costs $289 [7]. The device is a small rectangular magnet that can be directly attached to metal plates, held in the user's wristband, or

kept in a shirt pocket. It is used to give feedback on the user's power and speed of the workout. The device includes an accelerometer, gyroscope, and compass to determine user statistics. Like the BaziFit described above, thisisbeast, can connect via bluetooth to show relevant data. An interesting design decision made by thisisbeast is that the connection to the bluetooth device is essential for its use. This is due to the fact that the user of the device must start the recording process from their bluetooth device. For example, before the user starts their set, they must click a button on their phone, which starts a countdown, which then informs the user to start their set.

The third device is called Bowflex SelectTech 560 Dumbbells which costs $549 [8]. This device is a smart dumbbell. The technology to count repetitions and other statistics are built into the dumbbell instead of being an attachable like the products described above. The Bowflex SelectTech 560 comes with additional plates that can be easily attached to the smart dumbbell in order to increase the weight. The product also has the ability to connect via Bluetooth, where user statistics can be analyzed and shown through a screen display. The big advantage of having the Bowflex SelectTech 560 instead of regular dumbbells is due to its ability to adjust weight. This means there is no need to buy an entire rack of dumbbells, since the user would only need one Bowflex SelectTech and adjust the weights according to the exercise.

The SmartBell is both similar and different from the three products described above. Most importantly, the SmartBell is cheaper to produce, then buying the BaziFit. Like the devices described above, the SmartBell has the capability to detect repetitions and completed sets. It also has the general purpose of providing user feedback, so that the user can have a more effective workout. The SmartBell differs from the other products because it does not interface to another device via Bluetooth. Instead, the feedback is delivered directly onto a screen attached to the device. That way the user does not have to configure the workout device from a separate entity

just to use it. One other interesting feature of the SmartBell is the live feedback in the form of an array of LEDs. Out of the three devices above, only the BaziFit utilizes some sort of lighting mechanism to provide feedback. The BaziFit changes colors throughout the workout and lights up the entire square like structure. On the other hand, the SmartBell has a row of LEDs that intuitively lights up to inform the user to correct their form, or in other words, making sure the user lifts the bar parallel to the ground without tilt.

In order to successfully complete the prototype of the SmartBell, Fun Times Part 4 had to draw knowledge from previous courses. In terms of choosing all the various parts and designing the board, the three Fundamentals of Electrical Engineering courses were drawn upon. Multisim and Ultiboard were used to make all the wired connections and finalize the board layout. Fun Times Part 4 had to rely heavily on their knowledge learned from Intro to Embedded Systems. The class was taught using an MSP430 and the SmartBell used the same CPU, so much of the material taught in that course was helpful. One other course material Fun Times Part 4 used was knowledge gained from various Science and Technology courses. The team wanted to create a device that would actually be similar to what is trending currently, in whatever field chosen. In this case, the team had to figure out how has gym equipment evolved in the past few years and what should be included in the feedback to the weightlifter.

Constraints

Design Constraints

One major concern about the design was the need for it to be low power. If the SmartBell is to be attached to a barbell it must be relatively small and unobtrusive. If it were large due to a battery pack or needed wires coming out of it that would make it hard to use. In line with this

thinking, the entire design of the SmartBell needed to be small and compact. Thus the PCB board had to be as compact as possible, with the outside shell being as fitting as possible as well.

Stemming out of the need for low power, it is logical to expect that there will be limited computing power that can be provided from a small CPU package such as the MSP430. Utilizing data from an inertial measurement unit most likely will require a lot of processing and that will need to be optimized for the MSP430 to handle.

Economic and Cost Constraints

The goal for the team was to make the total cost of production less than the market price of similar products. In this case, Fun Times Part 4 wanted all the external labor costs and component costs to be less than $200, which is the market price of BaziFit. Fun Times Part 4 believes to have reached this constraint, but are not completely certain. The cost of all parts including the PCB even in the prototyping stage is below $90. However, they are unsure of the cost of 3D printing, and the labor of 3W, a Charlottesville based company the team frequented for soldering assistance.

In discussing the cost for the development of the board, Fun Times Part 4 did not break any constraint set by the Capstone class budget. The team believes this to be true as they were not contacted by instructors warning that any particular hardware was too costly for the class budget. All parts submitted to any parts order form were delivered without concern.

External Standards

The SmartBell's use during exercise will likely result in sweat or water splashes on the

unit. Due to this, NEMA enclosure standards 1 and 2 should provide a courteous level of dust, splash, and drip resistance [9]. The product does not need to be watertight or submersible by any means but should be able to withstand splashes in an indoor or normal outdoor condition.

Smartbell will also need to abide by NEMA ICS 16-2001 for Motion/Position Controls and Feedback devices [10]. The product will encode the motion of the barbell for balance and spatial position and can be considered a medical feedback device.

If the SmartBell was taken to market, the IEEE 82079-1-2019 standard for instructions for use of products would be required [11]. Instructions would include how to setup the product, interact with the system, and how to power/charge the device.

Battery power standards are necessary as the SmartBell will contain an internal battery capable of charge via USB. Therefore, we will be referencing the Battery Charge Specification for USB 2.0 as our power standard [12].

For general PCB design in electronic industry, IPC-2221 [13] was used. This standard defines recommended minimum spacing between internal and external conductors as a function of peak working voltage level. Power conversion circuits use an additional standard IPC-9592 [14]. This standard defines linear circuit board spacing recommendations equal to (0.6 + Vpeak*.005).

Tools Employed

Tools in three main areas were used in order to create the SmartBell. It consisted of tools to create the physical board, tools to create the software, and tools to create the enclosure.

National Instruments' Multisim and Ultiboard were used to create the circuit schematic and PCB layout respectively. All of the selected parts were first entered into Multisim, ensuring

that the packaging and pinouts were correct and were wired correctly. This was then transferred over to Ultiboard where the physical packages were entered in order to create a footprint for the parts. Team members who were in charge of these areas had extensive knowledge using the programs to design, create, and test circuitry.

In addition, physical tools such as National Instruments' Virtual Bench, soldering irons, and heat guns were necessary to assemble and test the PCB design. The Virtual Bench allowed the use of an oscilloscope to see if digital signals were outputted correctly while the multimeter included allowed the test of power systems. Soldering irons and heat guns were needed simply to physically assemble the parts.

To develop the software for the SmartBell, Texas Instruments' Code Composer Studio along with their MSP430 debugger were used. Code Composer Studio is an integrated development environment which makes coding for the MSP430 simple because it provides all the required libraries and builds the code according to the needs of the MSP430. In addition, when used in tandem with the MSP430 debugger, it allows real-time monitoring of software execution which allows for bug fixing of the code much more easily. All members of Fun Times Part 4 had taken and passed Intro to Embedded, a course where programming and debugging on the MSP430 is taught.

Finally, in designing the outside enclosure for the SmartBell, AutoDesk Inventor 2017 was used. This software allowed CAD design of the enclosure which was later exported the IGS file type for 3D printing.

Ethical, Social, and Economic Concerns

Environmental Impact

The SmartBell enclosure is also printed out of recyclable hard plastic. Plastic waste has been a concern on the rise, but the use of plastic as an outside case is also necessary. By making the case recyclable, we hope to reduce the impact on the environment.

All other parts used are mass produced with the SmartBell taking up much less than even a fraction of all materials. Therefore we see the SmartBell causing little impact on the manufacture of these electronic devices.

Sustainability

The SmartBell can be considered to be sustainable for two reasons. First, as mentioned above, we believe that the impact on the environment is minimal, and made of recyclable products in the device needs to be disposed of. However, that is not all.

The SmartBell is also designed to be able to be used for an extremely long period of time. During workout, the SmartBell rarely comes into contact with the human body. This should vastly increase the lifespan of the device and gifting the tool to another person should be easily possible. Some people might outgrow the SmartBell as they become more experienced or instead seek out a personal trainer. However, the sport of weightlifting does not change, and thus the SmartBell will stay relevant for an extremely large amount of time.

Health and Safety

Electrical failure could always be an issue with every electronic device, but the low electrical conductivity of cast iron and stainless-steel barbells should ensure no injury is sustained to a user while lifting with the SmartBell. In addition, the highest operating voltage will be 5V. Current regulation is more important than voltage management for electrocution safety [15]. However, a rubber component could be used as traction to grip the bar and provide a medium to prevent any current flowing to the bar in the event of a malfunction. For the SmartBell, the plastic case should provide more than enough insulation.

The largest safety risk when using the SmartBell is the distraction the tool provides from completing a lift. Weightlifting and trying to maximize reps or total weight require complete focus to perform correctly and not result in injury. The more simplified our device can be the better as a quick glance should offer all the information the user needs to know about balance of the bar and number of reps completed correctly.

Manufacturability

The SmartBell is highly manufacturable due to its printed circuit board and attached internals making up the majority of the manufacturing process. The PCB is simply placed inside the SmartBell's case with some poly-urethane foam surrounding it to prevent unnecessary vibrations and movement within the case. The limited number of parts and largely automatable processes of case and PCB creation make large scale manufacturing a definite possibility.

Ethical Issues

A possible ethical issue for the SmartBell is the collection of personal data of health and exercise habits. The product is meant to give instantaneous feedback on correct form, number of repetitions, and rest time. No large data sets will be stored or accessed on any server or application. In addition, there is no persistent storage of data and powering off the device will erase all data on it. If data collection was integrated into the project, Health Care Data Standards would be consulted to ensure the information we obtain can dissipate via national health information infrastructure [16]. The only point of interaction for a user's data is their device itself.

Intellectual Property Issues

As mentioned in the background section, there are several devices that aim count repetitions and electronically control exercise routines. US patent 20170000386A1 is assigned to the Bazifit. The patent describes a method and system for monitoring and analyzing position, motion, and equilibrium of body parts [17]. Form correction is a main objective of the SmartBell. Specifically, the Bazifit patent claims use of the " A system of evaluation comprising … at least one sensor removably attachable to at least one or more of: an object and a user's body" in claim 15 and "at least one subcomponent comprising at least one of: an accelerometer, a gyroscope, a magnetometer, and a computer processor" in claim 18. Utilizing our inertial sensor in this fashion would be dependent on these claims.

CN patent 207024475U is described as a kind of dumbbell for possessing exercise data acquisition function [18]. The Bowflex SmartTech 560 is dependent on this patent. The patent

claims to “[possess] exercise data acquisition function [characterized by] at least two kinds of infrared temperature sensor, humidity sensor, pressure sensor, acceleration transducer, gyroscope and magnetometer”. Due to our use of accelerometer and gyroscope in tandem, the SmartBell would be dependent on this patent.

US patent 20080090703A1 is described as an automated personal exercise regimen tracking apparatus [19]. Several key functionalities described in claim 1 of this patent would make our project dependent on it including “determining each of a plurality of exercise repetition events from the received motion signals”;“determining a cumulative repetition count for the selected exercise activity set from the determined exercise repetition events”; and “notifying the user via the user interface when the determined cumulative repetition count has achieved the prescribed number of exercise repetitions for the selected exercise activity.” The SmartBell would encapsulate all these functions with the addition of dynamic balance feedback. Due to the SmartBell’s main functionality and user interface depending on repetition counting, the technology would be dependent on this patent.

Detailed Technical Description of Project

The SmartBell is designed to give weightlifters instantaneous feedback about their form when performing a repetition. It also gathers statistics such as number of sets and repetitions, calories burned, timing information, and information about bad reps. This device is meant to be small and attach to a barbell where a user can see it.

The SmartBell is a device centered around a MSP430 CPU. The MSP430 is interfaced with two inputs, the inertial measurement unit which includes an accelerometer and gyroscope to gather physical data and buttons to interface with the human user. There are also three outputs,

an LED array, an LCD display, and a buzzer to interface with the user. Software is then implemented to gather and process statistics. A block diagram representing the overall design is shown below.

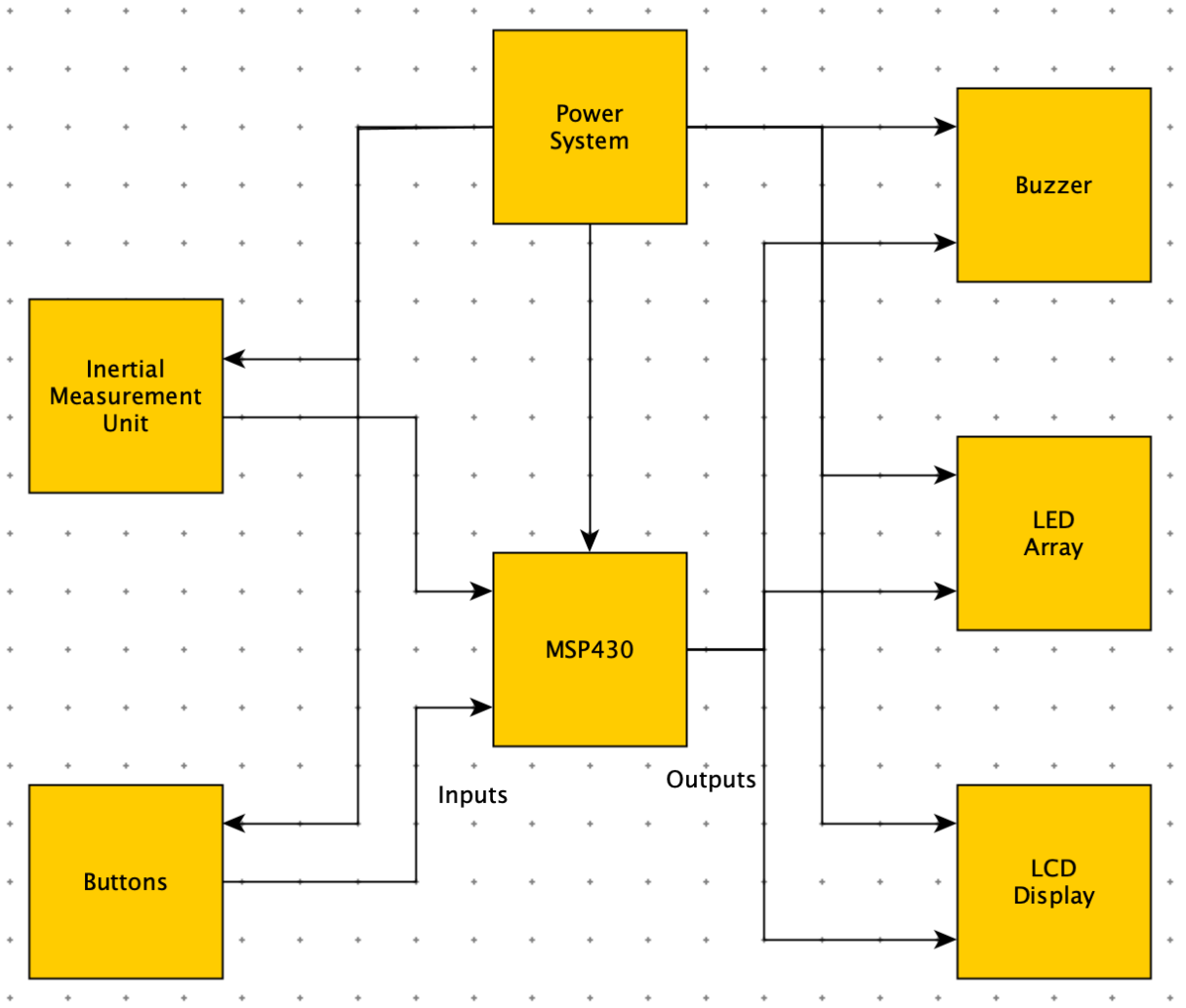

Figure 1 System Overview

There is also a power system which steps various voltages down to 3.3V which is used as power by all other components. The center of the power system is a Texas Instrument battery charger, the BQ24166RGER [20]. This allows input from two external sources in addition to a battery input. The main power source for the SmartBell is a 3.7V, 1000mAh, lithium ion battery

by SparkFun Electronics. This provides a measured operating time of 15-20 hours, with the option of selecting a smaller capacity batter to save costs. A workout will usually last no longer than 2 hours. This battery is connected directly to the battery charger chip instead of to the system board voltage regulator. To charge the battery, the SmartBell incorporates a micro USB-B port [21] to provide power. This port does not transfer any data. This 5V input is again connected directly to the battery charger chip. The chip also allows the connection of a 12V input, but that is unused and unconnected for the SmartBell. The chip contains a buck converter that outputs a stable 3.7V. In the case the battery voltage drops below, it will use the 5V input to charge the battery. This battery charger was specifically designed for lithium ion batteries. The battery charger chip also includes two LED outputs, one for "Power Good" indicating either the 12V or 5V external source was connected and one for "Charge" indicating that the battery was being charged. In addition, the chip also had multiple mosfet outputs to drive loads but were not utilized in the SmartBell. The battery chip did require a current setting to limit the USB charging current, and that was hardwired using either the system output voltage to signify high or system ground to signify low. This was selected by referencing the datasheet and set at 1A. This selection is there to prevent excessive current draw, but with modern USB specifications having a minimum of 1A output, this was an easy decision. The output of the battery chip at 3.7V was then run through a low dropout voltage regulator, the Diodes Incorporated AP2120N-3.3TRG1 [22], to drop the voltage down to 3.3V. This decision was twofold. First it allowed extended operation in case the battery voltage dropped below 3.7V all the way until 3.4V, which is a safe lower bound to prevent excessive discharge of the battery. Second 3.3V is a good upper limit for the board components as it was on the higher end of operating voltages without being the maximum voltage accepted by any board component. Below is a diagram of the power section.

U18 is the USB input near the top, U13 is the battery near the bottom, and U29 on the right is the voltage regulator feeding the board VCC.

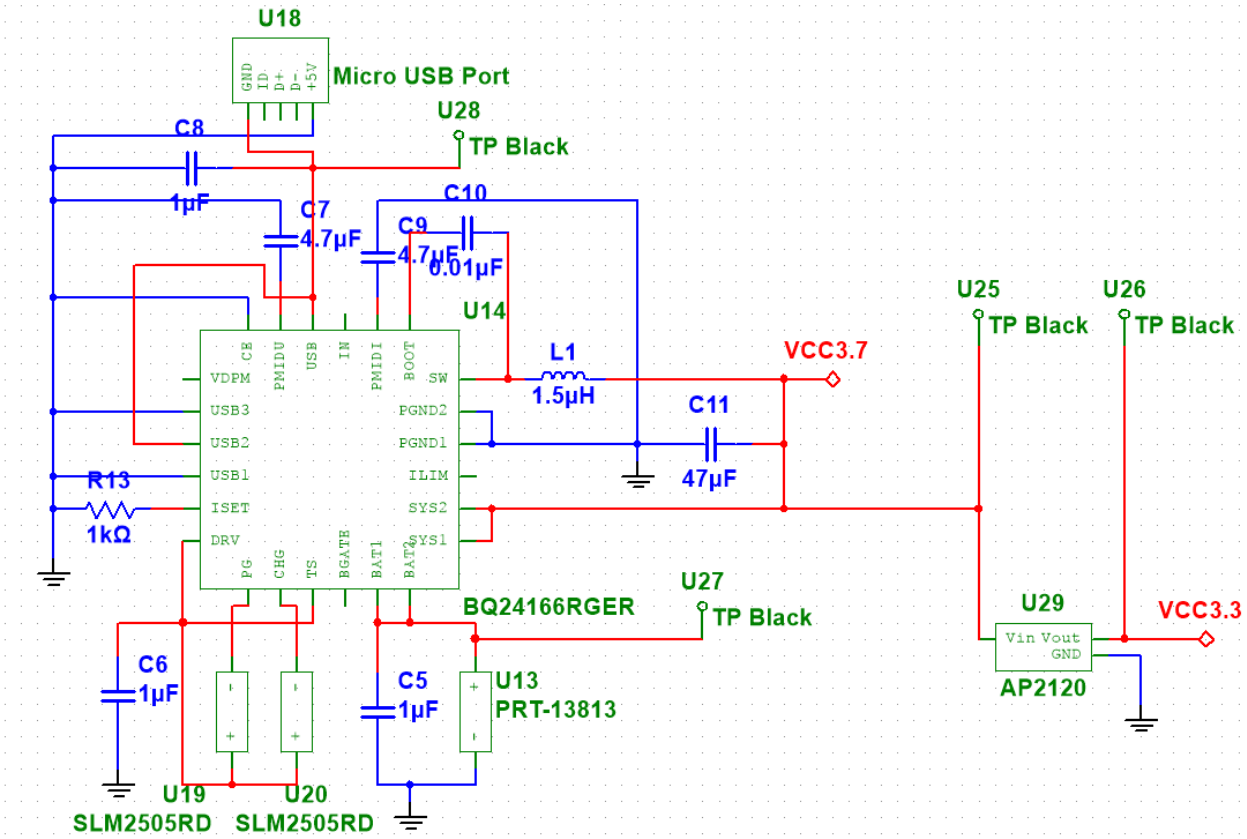

Figure 2 Power Supply Schematic

The MSP430 is relatively simple with only power at 3.3V and a Spy Bi Wire interface requires to work. The MSP430 chosen was the G2553 [23] as that had the largest amount of flash at 16KB and ram at 512 bytes. Originally a MSP430 with lower specs were chosen but a decision to move to this version was made after seeing how much space the code took up. The two wire Spy Bi Wire interface is the simplest connection possible which allows for programming and debugging. It was decided that the additional UART connections not be used due to the limited I/O pins available. If a MSP432 were used instead, the decision might have been made to create more connections to help debug the software. A small diagram of the JTAG header [24] is shown

below.

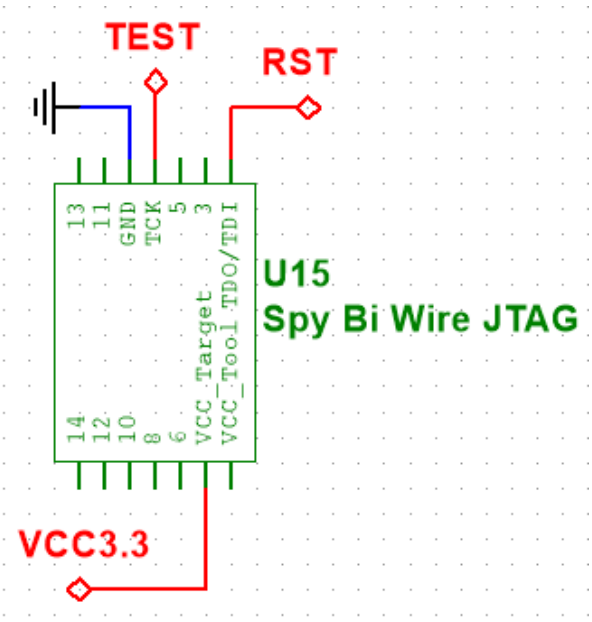

Figure 3 JTAG Header Schematic

The inertial measurement unit is a Bosch Sensortec BMI160 [25] which includes an accelerometer, gyroscope, and magnetometer. The SmartBell only utilizes the gyroscope and accelerometer to determine 3D orientation as well as axis and direction of movement. The accelerometer communicates using SPI and is implemented in software instead of hardware. Being surface mount parts, it was import the traces were routable and using hardware SPI would have made that extremely difficult. SPI was also chosen over I2C for ease of implementation. The response time of the IMU was also long enough that software SPI was sufficiently fast. To communicate with the IMU, various values were written to registers to configure the device. Values were then read from other registers which corresponded to the measured physical values. A frequency of 2000 hertz was selected to provide the maximum amount of accuracy and reduce integration error. The simple wiring can also be seen below; all wires are directly connected to the pins.

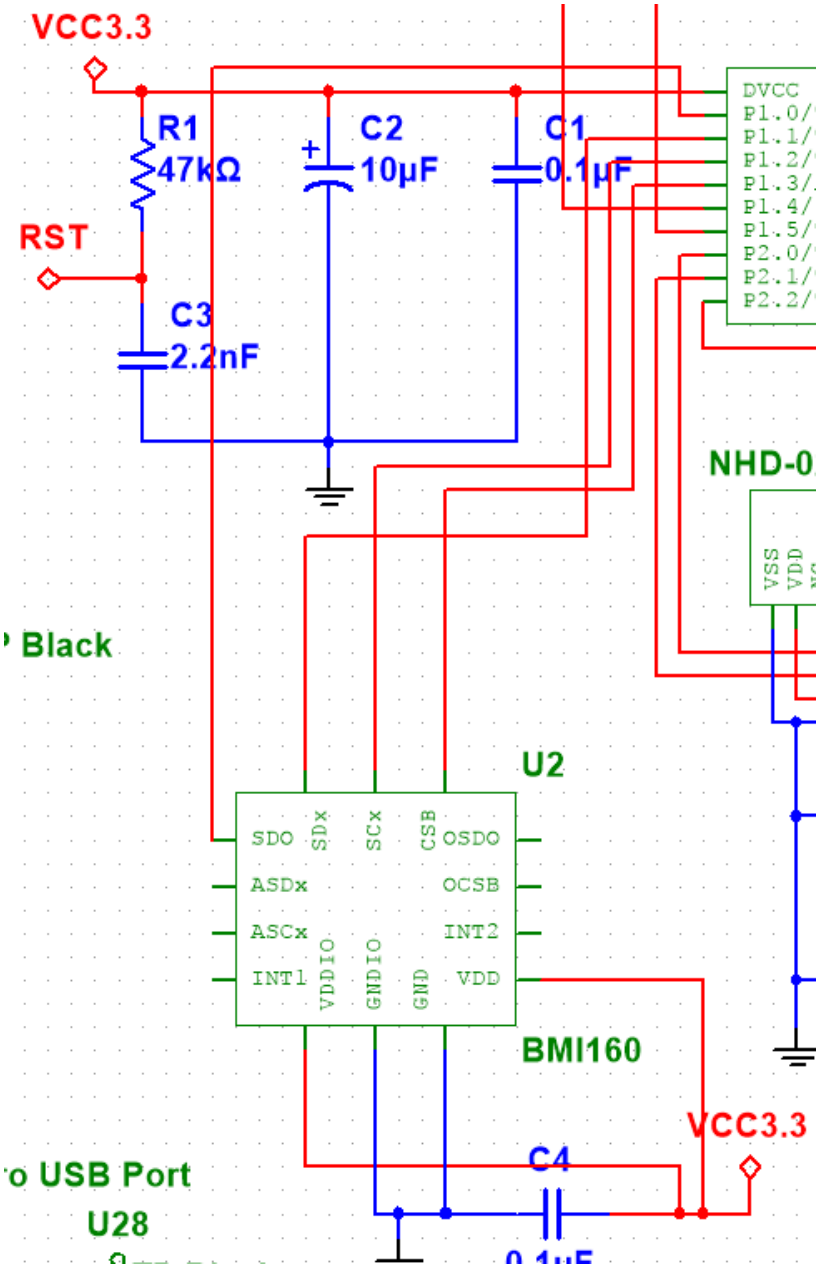

Figure 4 Inertial Measurement Unit Schematic

The second input to the SmartBell is the pair of buttons [26]. These buttons function as an up and down arrow when used individually and an enter key when pressed in tandem. Each button is de-bounced separately to ensure good and consistent presses. These buttons are connected directly to the MSP430 through pull up resistors of 35k ohms.

One visual output device was the LED array. A decision was made to choose an odd number of LEDs so that a center could be indicated, and that the number be less than 8 so that one byte could represent the entire array of LEDs. 7 LEDs fit the bill. It was made sure that the LEDs were all bright enough while at the same time drawing little power. Some trial and error was done before settling on the Kingbright WP710A10SRD/J4 [27] which was visibly bright enough at a distance of 2 feet and had a rated current draw of 20mA. The LED driver was the Texas Instrument TCA6507PWR [28] which supports 7 LEDs and communicates with the I2C protocol. In interfacing with the MSP430, hardware I2C was chosen to provide the fastest communication possible with little set up. The seven LEDs were directly connected to the LED drivers while the LED driver I2C lines were connected to the MSP430 with pull up resistors of 1K ohm as seen in the schematic below. The internal MSP430 resistors at 35k ohms were too large to support fast communication. Instead, values of 1k ohms were used.

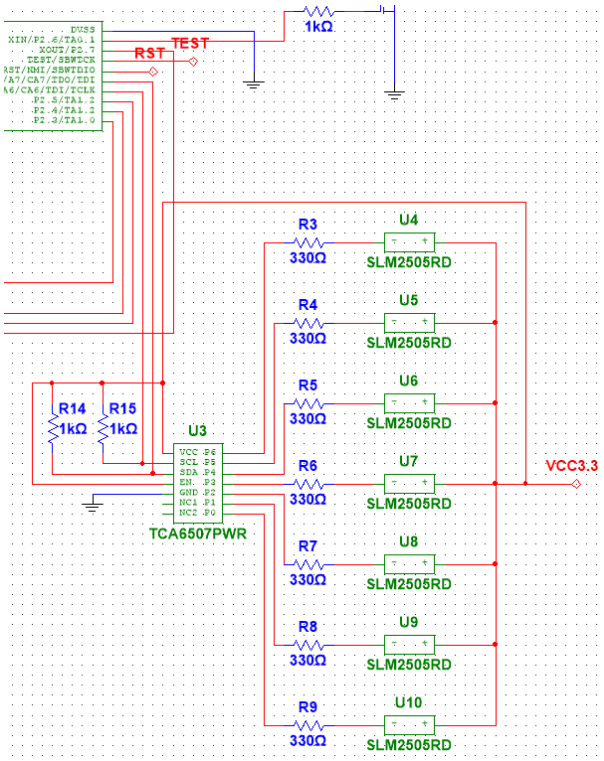

Figure 5 LED Array Schematic

The second visual output device is the LCD display NHD-0216HZ-FSW-FBW-33V3C by New Haven [29]. This LCD was chosen for the two-line 16 character size which fit all the menu items perfectly along with operating fine at 3.3V. It was also chosen because it could operate with only four data lines as a minimum, but 5 was used to help debug. This LCD screen could operate with 9 connection 8-bit communication but that would have overwhelmed the MSP430s I/O ports. It is interesting to note that this LCD display did not use a normal communication bus such as I2C or SPI but had its own communication protocol which was implemented using software and matched to the data sheet specifications. Data was sent to the LCD 4 bits at a time. If an MSP with more ports was used, 8-bit communication would have been used to reduce latency and increase responsiveness of both the LCD and SmartBell overall. Again, the wiring can be seen below.

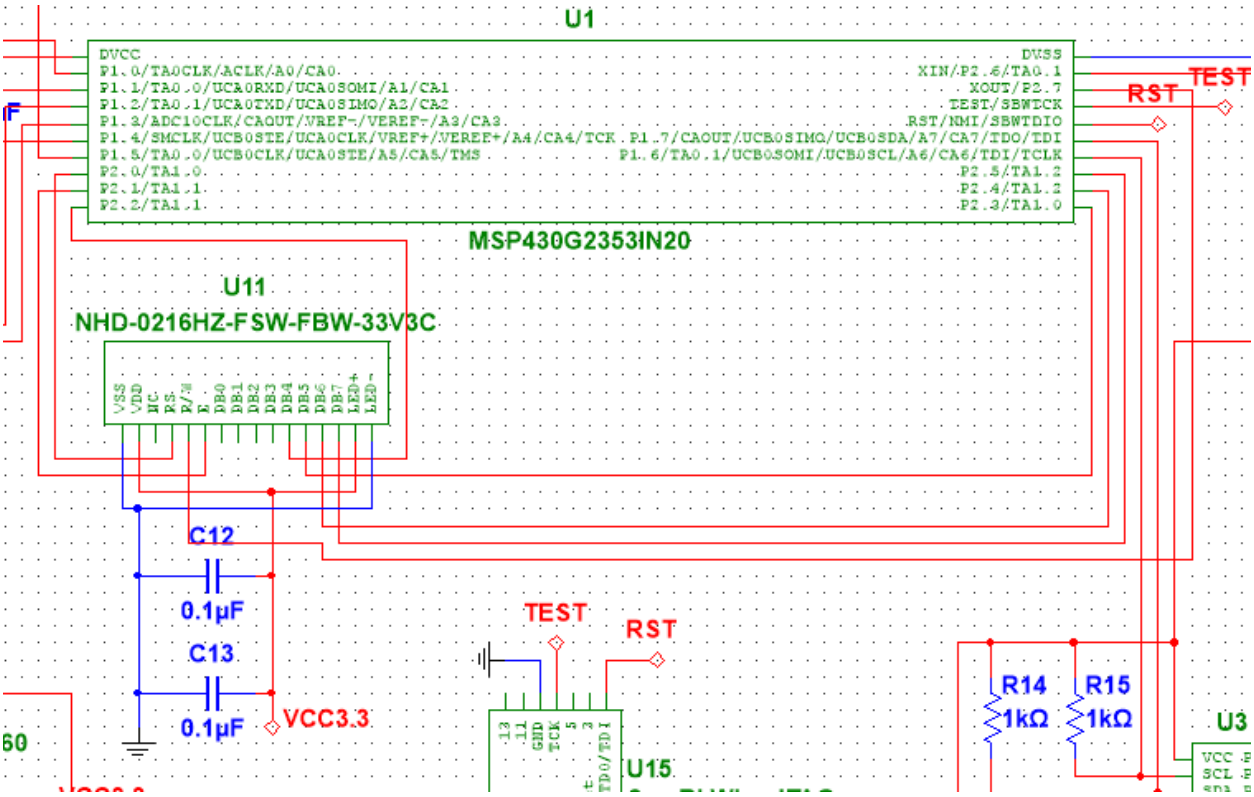

Figure 6 LCD Display Schematic

Finally the buzzer, a PKMCS0909E4000-R1 by Murata Electronics [30], is the third and final output of the SmartBell. Originally designed to provide feedback when catastrophically bad form is detected, this is later connected but unused during operating due to how annoying and distracting it is. The output port of the MSP430 for the buzzer is connected to an N-MOSFET [31] which then drives the piezo electric buzzer using the board's 3.3V power supply. This buzzer doubles as a signal output for debugging and was very useful for hooking up an oscilloscope. Below is a schematic which shows both the connection of the buzzer and buttons.

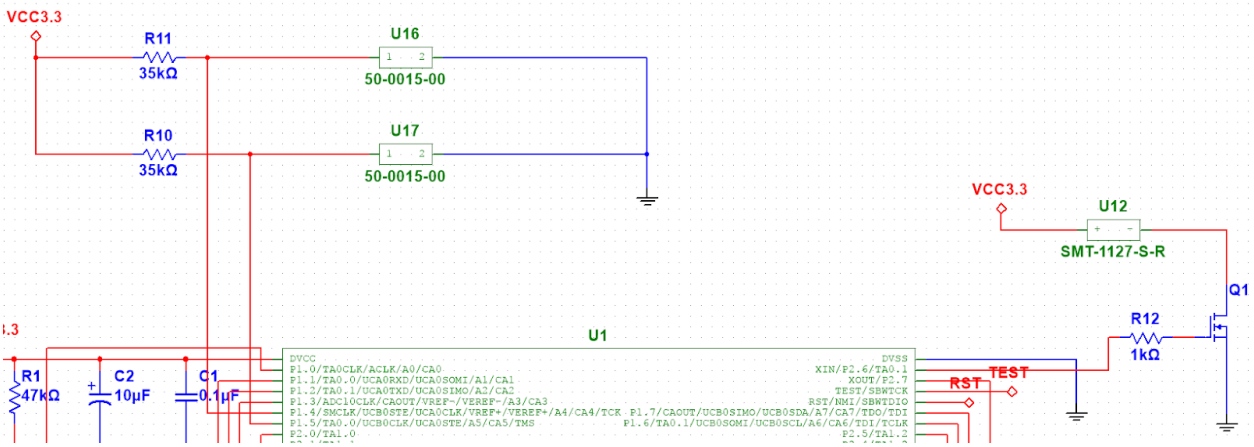

Figure 7 Buzzer and Button Schematic

When the SmartBell is powered on, a series of setup screens are shown. In order, these screens allow the setting of weight in kilograms, number of reps per set, and number of sets. The upper button increases the count by one while the lower button decreases the count by one. A double press indicates to move onto the next screen. Preselected values such as 50 kg and 3 sets are initialized to reduce the number of button presses needed to get to the desired value. After these screens, a calibration screen instructing the user to not move is then displayed and IMU values for the accelerometer and gyroscope in all three axis are taken to be used to give a point of reference. Once this is done, the device enters normal operation. An example of a setting menu is shown below.

Figure 8 Weight Setup Menu Screen

During normal operation, the first thing the user sees is that the LED array will change depending on the degree of tilt. This value is determined using the accelerometer and the difference of the physical data value to the calibration data in the calibrated axis. Depending on the intensity of tilt, one, two, or three LEDs in that same direction will light up. An if statement in the code checks the values and then determines the LEDs to light up. A switch statement then send using I2C the instruction to light up the LEDs depending on the desired output. This separation allows easy change of the threshold values, which are set at intervals of 10 degrees at the time of this report. Below is a picture with the device at a tilted angle.

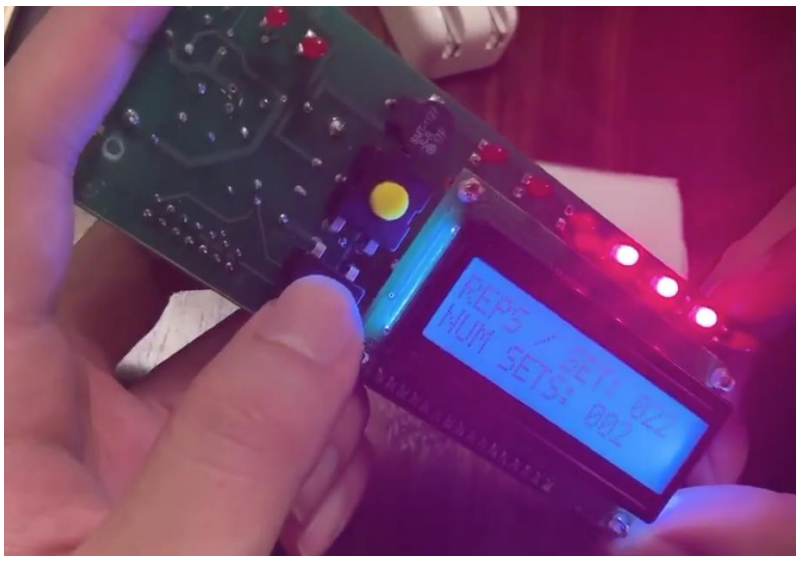

Figure 9 LED Array Responding to Tilt

The menu will display one page of information and update every time a repetition is completed, or if a button is pressed. The two buttons again serve as two different directions of page changes. Each page serves different information and rolls over. The counts of the repetitions, good or bad, update on the menu after each rep. In the figure above, it can be seen that information is written to the screen.

The most important piece of code is the IMU code to determine orientation in space. The sensors are first calibrated by reading steady values done during initialization. The gyroscope data is then run through integration to find direction in space. This is required as the accelerometer cannot differentiate between acceleration due to gravity and acceleration due to movement of the device. The accelerometer data is run through CORDIC to determine angle without the need for computationally intensive multiplication and division which the MSP430 does not support with hardware. This, however, leads to a reduction in accuracy. Once a direction and magnitude are determined from the accelerometer, the calibration offset off gravity is then subtracted in the downwards direction which the integrated gyroscope data provides. This results in the acceleration data of the device due to movement only. This data can then be used to determine whether a rep has been completed, and whether it was completed with good form as a good rep should have minimal deviation from one axis and move straight up and down.

Ideally for best results, the acceleration data should also be integrated to provide speed and even positional data. However, due to the limited computational capabilities of the MSP430, a heuristic model utilizing timing intervals, threshold values, and movement direction was instead used to determine repetition count and repetition form. In the heuristic model, a repetition is only counted if it lasts a certain amount of time with an acceleration above a threshold which would indicate enough distance traveled. The rep ends when values stabilize, and movement stop

after detecting acceleration at the top and bottom of a repetition. Currently, the method for determining a bad repetition is excessive tilt in any direction. This criterion however can be easily modified using available data to include things such as excessive spinning, excessive jitter, or even excessive speed. Most of this will rely on the ability of the chosen CPU to process all the data in real time.

Below is a diagram that shows the operation of the repetition counting algorithm. The LED array and LCD are omitted as it is assumed that they each operate independently.

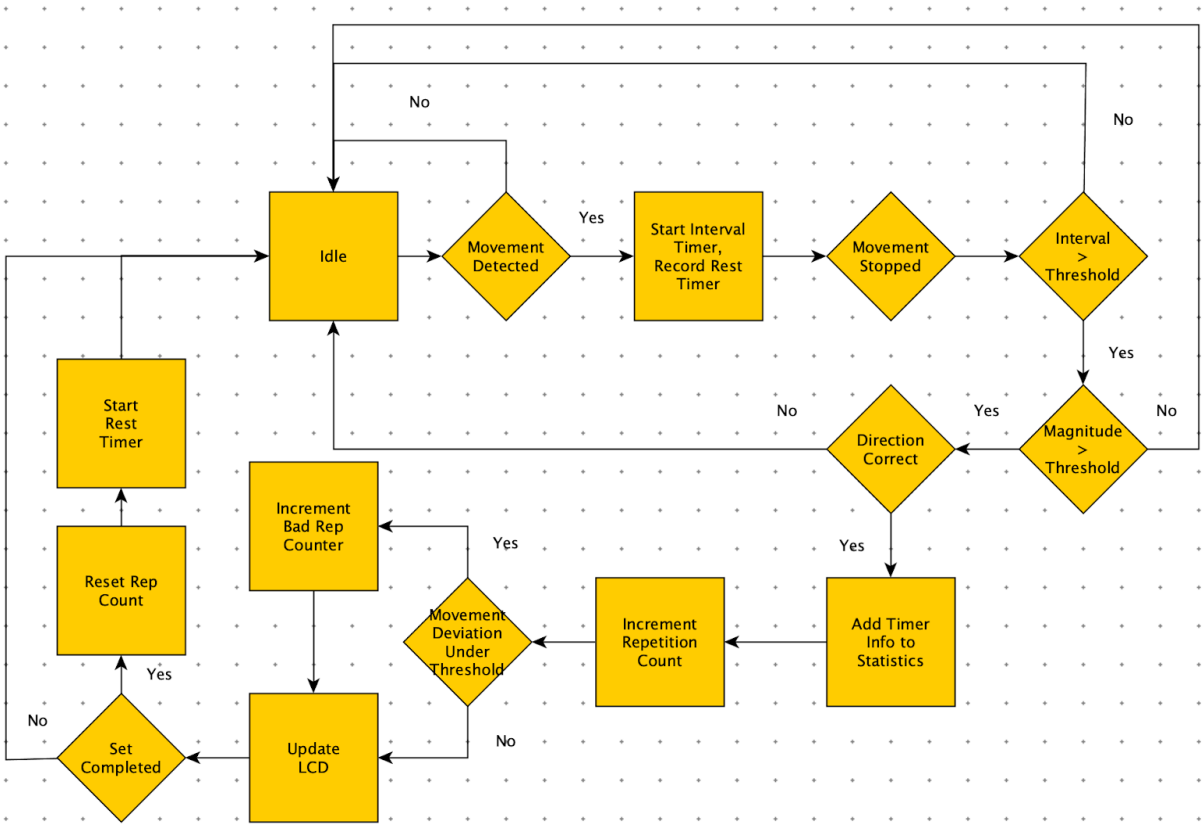

Figure 10 Repetition Counting Algorithm

Finally, below is the Ultiboard file representing the final PCB layout. This PCB was made to be as compact as possible to fulfill design requirements.

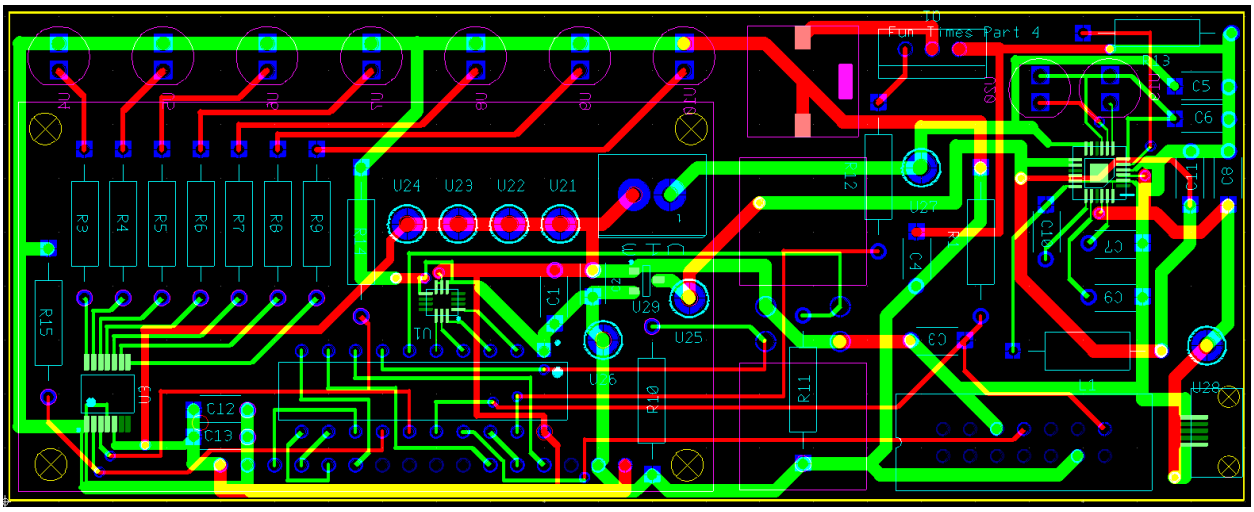

Figure 11 Ultiboard PCB Design

There are two interesting things to note. The first is the use of both sides of the board, with many components such as the LED array, MSP430, and IMU mounted on the opposite side of the LCD display. This created a huge reduction in footprint as the resistors took up a huge amount of space due to their through-hole mounting style. This can be seen in the bottom left hand side of the diagram.

The second interesting thing to note is the power traces. Everything related to power input is concentrated on the right-hand side of the board to reduce interference with signal traces. In addition, to ensure good power and ground distribution, loops of both the 3.3V and ground were created so that each component could easily connect to a very thick trace and reduce the resistance of copper. Due to the small chip sizes, traces near the chips often had to be extremely small at 9mils. This can be seen at the battery charger chip in the upper right and the IMU near the center. However, once the trace left the chips, no trace was less than 15mils with those at 15 having parallel traces. This ensures good connections. Most traces were at 25mils with power traces being parallel traces of 50mils.

In addition, each chip was placed in an orientation where minimal vias and turns are

needed to connect to the MSP430. This greatly made routing easier as surface mount chips meant that all traces had to start on the top copper.

Finally, the outer case is shown below, encompassing the PCB. There are cutouts for each separate LED as well as the LCD screen and USB port. The case is meant to be as form fitting as possible to reduce bulkiness.

Figure 12 SmartBell in the Case

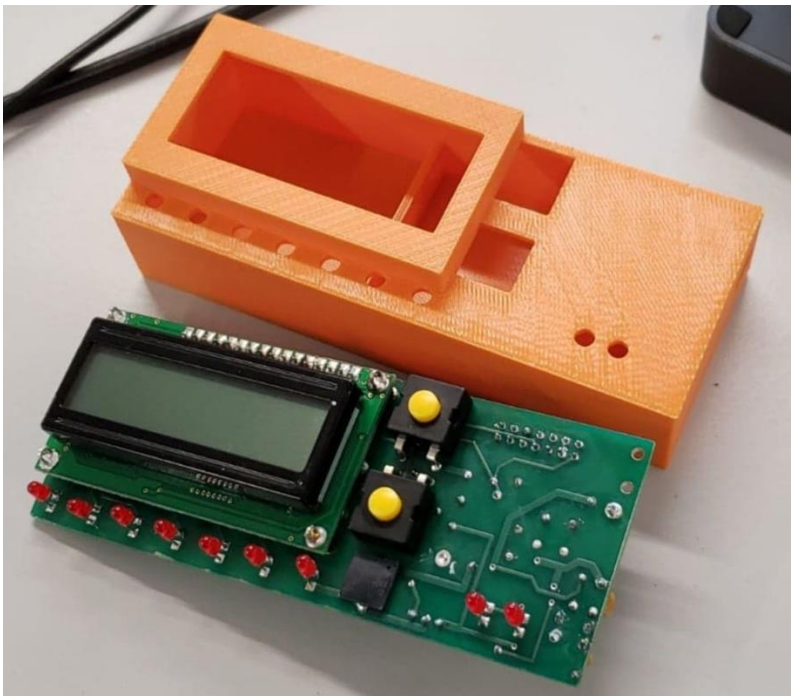

Figure 13 PCB and Case Side by Side

Project Time Line

Fun Times Part 4 used google sheets to create a Gantt chart to keep track of project schedule. The team followed the original project timeline very well, and so the final timeline is nearly identical. Two noteworthy changes made in the timeline were the completion of calorie /rest time code and PCB testing. These two coding tasks were pushed back two weeks in order to make sure that the midterm design review tasks were complete. It was also more important to try and get both the LCD and LEDs working as they can be use as debugging tools.

The start of the PCB testing was also pushed back one and a half weeks because the team did not have any board to test at the original set time. Instead, wiring was tested through the use of breadboards. The original Gantt chart as well as the final Gantt chart can be seen below.

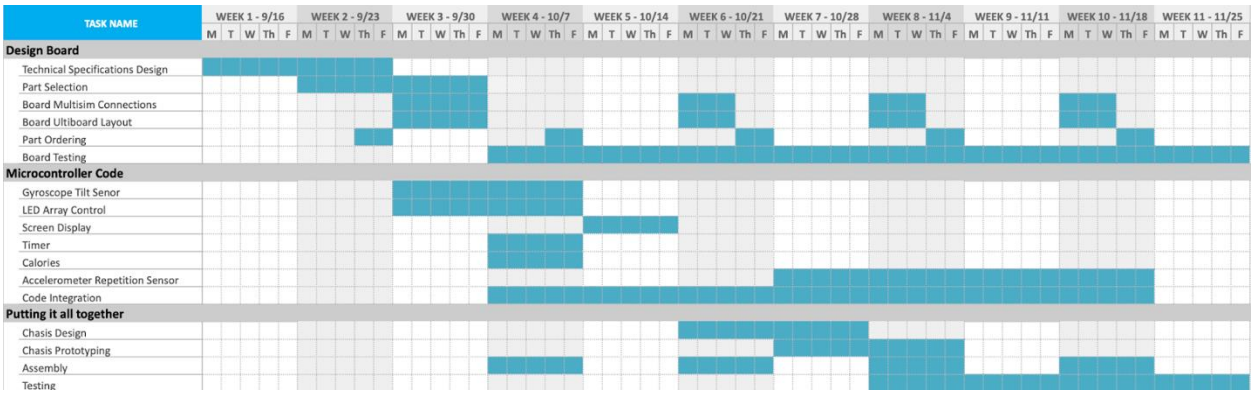

Figure 14 Original Gantt Chart

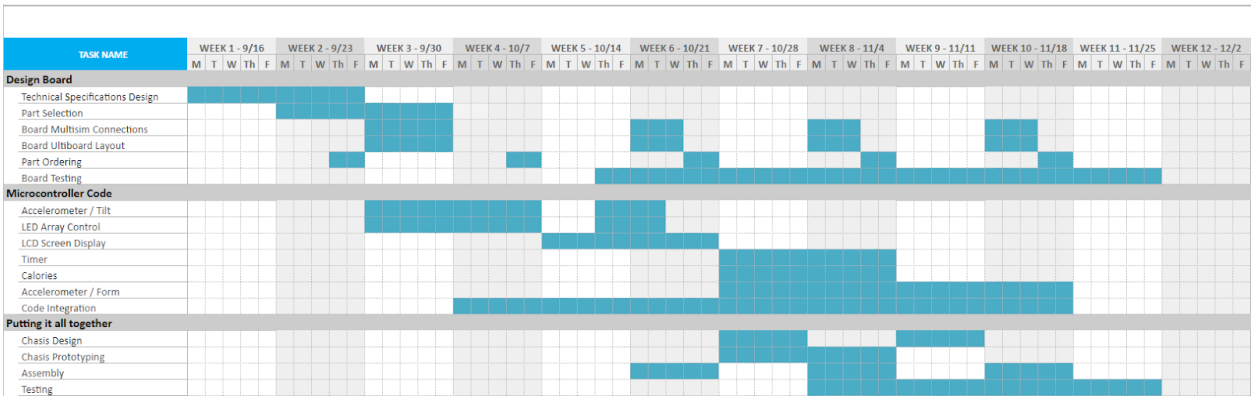

Figure 15 Updated Gantt Chart

It’s also important to note that the team finished the physical board design ahead of schedule, and so did not utilize the last two board send-outs or parts orders.

Only a few tasks had to be serialized, and they were at the beginning of the project when designing the PCB. First, hardware parts had to be selected, then connected in Multisim, and finally placed in Ultiboard. Every other task was done in parallel. Since the team had many iterations of the board, multiple team members could work on multiple tasks at the same time. For example, one person could work on accelerometer code while someone else worked on LED array code. This was also helped by the fact that a breakout board was ordered for the IMU which meant the team could breadboard our design.

The tasks were split up quite evenly. In the two stages of the design, every person had a task to be executed in parallel. At the beginning David’s task was to get the LCD screen working; Hamza’s task was to get the IMU interface working; Daniel’s task was to select parts and wire it up in Multisim; Nathan’s task was to interface with the LED Array; Kevin’s task was to select parts and design the board in Ultiboard. During the second phase, David’s task was to get the calories code working’ Hamza’s task was to design the outer casing; Daniel’s task was to get the repetition counting algorithm done; Nathan’s task was to help with the repetition counter; Kevin’s task was to integrate all the code and update board designs.

Test Plan

The SmartBell is mainly a software implemented device with minimal complex hardware implemented components. Testing can be separated into two parts, first, testing of board-wide components such as the power system that powers everything and the MSP430 which is the brain of the device. Second, testing of all of the peripherals such as the LCD display, LED array, and

inertial measurement unit to name a few.

Due to the integrated nature of the SmartBell's design as well as most physical connections going directly from one component into another, the board was fully assembled before any testing was conducted.

The first system to be tested was the power system. There were five test points provided for the testing of voltages representing the four different voltages used plus a ground point. There is a 5 volt input for a USB, and a 3.7 volt input for the battery. These are connected directly to the battery charge chip which then outputs a separate 3.7 volt output for the system. This output is then run through a voltage regulator which drops the voltage to 3.3V.

To test this system, first only the battery was connected. Multimeter measurements were then taken at all of the test points. At first, the battery voltage was too low and so it was charged up to maximum capacity of 4.1V using the Virtual Bench carefully to ensure that no overcharging was done which would damage the battery. After charging, testing was done again which did yield the correct battery voltage at the battery test point, a 3.7V output at the battery charger chip output, and 3.3V at the voltage regulator output. This was all relative to the ground of the board. Once this was working, a 5V was applied to the USB input test point. The testing of this input relied on the battery charger's output. An LED which signifies "Power Good" lit up, which indicates the 5V was connected correctly and recognized. In addition, it was observed that as the voltage of the battery dipped below 3.7 the LED representing a charge cycle lit up, indicating that charging was working correctly as well. Testing of the actual micro-usb port was done as well. This however revealed that the power wiring in the USB port was flipped. This voltage resulted in a puff of smoke from the battery charger chip, and a redesign of the wiring. This was as simple as switching around two traces.

After power, testing of the MSP430 commenced. This was as simple as making sure that we could interface with the MSP430 through the Spy Bi Wire interface. The MSP430 debugger was connected to the JTAG header port and Code Composer was used to try to load code onto the MSP430. This was always successful and indicated that the interface was correct and that the MSP430 could function correctly.

The next part to be tested was the LED array. This actually consisted of two parts, the LED driver and the LEDs themselves. The LEDs were first tested individually by applying a voltage to ensure that the LEDs were mounted correctly. They were all mounted correctly. Next, a test program was loaded onto the MSP430 to cycle through all of the LEDs to determine whether communication through the I2C bus was working. In testing for the later versions of the SmartBell, this was all successful. However, during the first iteration of the SmartBell there were some issues with the I2C communication which was noticed by hooking up an oscilloscope to the I2C pins. It was noticed that signals were being applied too fast to be accepted. Once this was found, simply slowing down the rate of sending was enough to see the LED array function as intended.

The next thing that was tested was the LCD display. Once the board was powered the LCD screen immediately lit up which represented that all of the power wiring was correct. A different test program was then loaded onto the MSP430 which was intended to write the alphabet along with some punctuation characters to the screen. This was always successful in all iterations of the SmartBell.

The next thing that was tested was the buttons. These buttons were toggle buttons. A multimeter was connected to the button leads. The MSP430 debugger was then used to check the value the MSP430 was reading from the buttons. The values always matched, with 3.3V

corresponding to logical high and 0V corresponding to logical low.

The buzzer was tested next. This was done by outputting a PWM signal from the MSP430 to the buzzer. It was during this testing that it was discovered that the 6mA output of the MSP430 was not enough to power the buzzer. This lead to a design change where the MSP430 would output a signal to a N-Mosfet which would then drive the buzzer using the 3.3V board power. This design was then verified in the same way as the original design and had no issues.

The final physical piece of the board that was tested was the inertial measurement unit which included the gyroscope and accelerometer. The MSP430 debugger was used to read and write values to various registers. At first, this testing was unsuccessful. After using the oscilloscope to look at the SPI signals, it was seen that the signals being sent were not correct. This resulted in an examination of the software code. Once that was fixed, reading and writing to registers were successful. However, reading the actual physical measurements was still incorrect. The values did not change during our tests. This was later attributed to incorrectly setting up the device and not allowing enough delay for the device to initialize.

Integration testing was then done to ensure the parts worked together. The LED array code was changed to light up the array based on the IMU readings. At first this did not work, but it was then attributed to the lack of calibration, which was added. The buttons were also tested to see if they would change the menu functions like designed. Finally, physical reps were done to determine whether the counter was successful and whether the timer worked correctly. This was all successful once implemented.

Final Results

The SmartBell achieved all the goals that were set out. Some went beyond expectations while others were met with a minimal functional product. At this point it should be obvious that the most central modules of the SmartBell such as the power system and use of the MSP430 was successful as that is the foundation for all other functionality.

One requirement was to display the level of tilt in real time using an LED array. This was completely perfect, and the update rate was even adjusted to 100 hertz to allow for smoothness in changing lights.

The LCD along with the buttons were designed to allow interface with a human user. The buttons not only work by themselves individually but a double press of both buttons at the same time was also successfully implemented to allow the setting of variables such as weight, number of sets, and number of reps. The LCD display also was able to display all of the wanted menus and information, using the buttons as cues to cycle through the pages.

As a very simple feedback method, the buzzer also worked perfectly. However, we decided not to include that as it's believed that the buzzing would cause too big of an annoyance for everyday use in the gym.

The most complex issue with the SmartBell is detecting reps performed and determining whether they were good or bad ones. The SmartBell successfully is able to count reps. This utilizes not only the accelerometer like we imagined before but also the gyroscope which requires integration. The two put together allowed the distinction between force on the SmartBell by a user and by gravity. However, because of this, a lot more error in measurement was introduced. This made detecting bad reps extremely difficult.

There was difficulty coming up with how to define what a bad repetition was. Due to this difficulty, our determination was a bit arbitrary. If the angle exceeds a certain threshold, the rep would be considered a bad one. However, the technology and software needed to determine acceleration and direction is completely functional. It relies on the accelerometer data in addition to the integral of the gyroscope data to determine 3D orientation and can be calibrated. Thus, with more testing and more processing power and memory the system should be able to be modified to a much higher degree of accuracy. It was also hard to specify what a bad rep was because there was little ability to grab real time data from the MSP430 without pausing execution, which would disrupt the timing and calibration. In hindsight, a serial connection to the MSP430 might have helped.

Although the SmartBell would need some improvements to become a truly accurate device, in its current form it checks the requirements set forth at the beginning of the project. As a proof of concept, it goes beyond showing that such a device can be made cheaply and perform efficiently enough to be used in the gym.

Costs

During prototyping, each SmartBell costs roughly $53 for parts, another $30 for the PCB board, and an estimated cost of $10 for the casing. This translates to just below $100. There is also additional labor involved to assemble the board, but this can be done under one hour per board by group members. This means a specialized worker in the field should be able to finish assembly in much less time.

When manufactured in quantities of 10,000, significant savings can be made on the SmartBell from parts only. The chart for savings can be seen in Figure 16 in the appendix. The

cost for 10,000 units is $324553 which translates to roughly $33 dollars per part. If we factor in the costs of the PCB board and plastic casing up at the scale of 10,000 unit range, we estimate the total costs to be $50. This is significantly lower than the prototyping board costs. Even with the added labor, the cost should be much lower than $100.

Manufacturability is high for the SmartBell as all components could be easily mounted to the PCB by machines instead of human labor. The casing is also completely automated when it comes to manufacture. Thus, the SmartBell could be easily mass-produced.

Future Work

The SmartBell in its current state can be said to be the first iteration of the product. Although it achieves all of its goals of monitoring a workout and providing feedback, there can be improvements to both the physical design as well as software implementation.

First, the physical design. The majority of the components used currently are still through hole parts. This means that in the goal of creating the smallest possible board this can be drastically improved. In addition, one huge footprint currently is the LCD screen. This is a package consisting of both a display and a display driver on a separate PCB board. If these parts are mounted directly onto the SmartBell then the footprint and vertical height can be reduced. It is estimated that the board can be reduced by another 30% in size.

Software is however where the largest improvement can be done. The largest issue comes from the selection of the CPU in this case the MSP430. It has limited computational capabilities as well as a limited number of hardware timers. The first issue we ran into was how to schedule all of the asynchronous systems such as the LED array, LCD display and IMU. In this regard, having a CPU capable of executing multiple threads would be a huge benefit. More hardware

timers would also help as this way execution of computationally intensive or long delay code will not affect much if any of the other systems on board. We also ran into issues of the lack of computing power. At first, we were expecting only to need to do filtering on signals. However, we required multiple integrations at the end.

It is definitely recommended that for all future project sufficient headroom be designed into the system. During the design of the SmartBell, we went for “just enough” which ended up being a constraint on our software. Both flash and ram were also a concern as our code for complex and to a point where storing it took up most of the flash and execution variables filled up the majority of ram. Having more space would allow more accurate detection of movement as we can store time series.

One other recommendation is to have some low-latency feedback which can be used for debugging. In the SmartBell it was the LED array. While the LCD display could provide much more detailed information, the delay caused tens of milliseconds of delay which affected the operation of other components. The LED array gave just enough information to help debug without creating unnecessary strain on the system.

Finally, it is advised that if there is space to implement a serial port connection from the MSP430 to a laptop using the debugger. This would help gather data, especially physical measurements such as acceleration without needing to pause code execution. This data can be dumped into excel files which can be easily viewed and parsed as proof of concepts. In the SmartBell, due to the use of a MSP430 the number of I/O ports was limited. Although the use of a serial port was not needed in the end to finish the project, it would have made debugging as well as selection of the repetition counting algorithm much easier.

References

[1] T. R. Measom and S. R. Watterson, “Treadmill”, U.S. Patent 3 156 65S, March 26, 1991.

[2] “Treadmill Buying Guide,” TreadmillReviews. [Online]. Available: https://www.treadmillreviews.net/treadmill-buyers-guide/. [Accessed: 16-Dec-2019].

[3] K. Johnson, “Gym Accident Statistics,” LegalMatch Law Library, 11-May-2018. [Online]. Available: https://www.legalmatch.com/law-library/article/gym-accident-statistics.html. [Accessed: 16-Dec-2019].

[4] S. F. Nadler, G. A. Malanga, J. H. Feinberg, M. Prybicien, T. P. Stitik, and M. DePrince, “Relationship between hip muscle imbalance and occurrence of low back pain in collegiate athletes: a prospective study,” *American journal of physical medicine & rehabilitation*, Aug-2001. [Online]. Available: https://www.ncbi.nlm.nih.gov/pubmed/11475476. [Accessed: 16-Dec-2019].

[5] C. Heffernan, “The History of the Smith Machine,” Physical Culture Study, 15-Dec-2015. [Online]. Available: https://physicalculturestudy.com/2015/12/15/the-history-of-the-smith-machine/. [Accessed: 16-Dec-2019].

[6] “How it works,” BaziFit. [Online]. Available: https://bazifit.com/device. [Accessed: 16-Dec-2019].

[7] “Shop Now,” Beast Sensor. [Online]. Available: https://www.thisisbeast.com/en/products. [Accessed: 16-Dec-2019].

[8] “Bowflex SelectTech 560 Dumbbells,” Bowflex. [Online]. Available: https://www.bowflex.com/selecttech/560/100581.html. [Accessed: 16-Dec-2019].

[9] NEMA Enclosure Types. (2005). [ebook] Rosslyn, VA: National Electrical Manufacturers Association. Available at: https://www.nema.org/products/documents/nema-enclosure-types.pdf [Accessed 23 Sep. 2019].

[10] Industrial Control and Systems Motion/Position Control Motors, Controls, and Feedback Devices. (2001). [ebook] Rosslyn, VA: National Electrical Manufacturers Association. Available at: https://www.nema.org/Standards/SecureDocuments/ICS16.pdf. [Accessed: 25- Sep- 2019].

[11] IEEE 82079-1-2019 - IEEE/IEC International Approved Draft Standard Preparation of Information for Use (Instructions for Use) of Products - Part 1: Principles and General Requirements, Standards.ieee.org, (2019). [Online]. Available at: https://standards.ieee.org/content/ieee-standards/en/standard/82079-1-2019.html. [Accessed: 25- Sep- 2019].

[12] Battery Charging Specification. (2010). [ebook] USB Implementers Forum. Available at: https://www.usb.org/document-library/battery-charging-v12-spec-and-adopters-agreement. [Accessed: 25- Sep- 2019].

[13] “Generic Standard on Printed Board Design,” *IPC*. [Online]. Available: http://www.ipc.org/TOC/IPC-2221.pdf. [Accessed: 16-Dec-2019].

[14] “Performance Parameters for Power Conversion Devices,” *IPC*. [Online]. Available: http://www.ipc.org/3.0_industry/3.5_councils_associations/3.5.0_ipc/spvc/0607/ipc-9592-final-draft-0407.pdf. [Accessed: 16-Dec-2019].

[15] Giovinazzo, P. (1987). *The Fatal Current.* [online] Available at: https://www.asc.ohio-state.edu/physics/p616/safety/fatal_current.html [Accessed: 25- Sep- 2019].

[16] Aspden, P. (2004). *Patient Safety: Achieving a New Standard for Care.* [online] Available at: https://www.ncbi.nlm.nih.gov/books/NBK216088/ [Accessed: 25- Sep- 2019].

[17] “US20170000386A1 - Method and system for monitoring and analyzing position, motion, and equilibrium of body parts,” *Google Patents*. [Online]. Available: https://patents.google.com/patent/US20170000386?oq=bazifit. [Accessed: 16-Dec-2019].
[18] “CN207024475U - A kind of dumbbell for possessing exercise data acquisition function,” *Google Patents*. [Online]. Available: https://patents.google.com/patent/CN207024475U/en?q=dumbell&oq=Bowflex+dumbell. [Accessed: 16-Dec-2019].
[19] “US20080090703A1 - Automated Personal Exercise Regimen Tracking Apparatus,” *Google Patents*. [Online]. Available: https://patents.google.com/patent/US20080090703A1/en?q=counter&page=1. [Accessed: 16-Dec-2019].
[20] “BQ24166 | BQ24166RGER | Battery Charger ICs | Description & parametrics.” [Online]. Available: http://www.ti.com/product/BQ24166. [Accessed: 16-Dec-2016].
[21] “10118193-0001LF Amphenol ICC (FCI) | Connectors, Interconnects | DigiKey.” [Online]. Available: https://www.digikey.com/product-detail/en/amphenol-icc-fci/10118193-0001LF/609-4616-1-ND/2785380. [Accessed: 16-Dec-2019].
[22] “AP2120N-3.3TRG1 Diodes Incorporated | Integrated Circuits (ICs) | DigiKey.” [Online]. Available: https://www.digikey.com/product-detail/en/diodes-incorporated/AP2120N-3.3TRG1/AP2120N-3.3TRG1DICT-ND/4505261. [Accessed: 16-Dec-2019].
[23] “MSP430G2553 | MSP430G2x/i2x | MSP430 ultra-low-power MCUs | Description & parametrics.” [Online]. Available: http://www.ti.com/product/MSP430G2553. [Accessed: 06-Dec-2016]. https://www.digikey.com/product-detail/en/diodes-incorporated/AP2120N-3.3TRG1/AP2120N-3.3TRG1DICT-ND/4505261. [Accessed: 16-Dec-2019].
[24] “5103308-2 TE Connectivity AMP Connectors | Connectors, Interconnects | DigiKey.” [Online]. Available: https://www.digikey.com/product-detail/en/te-connectivity-amp-connectors/5103308-2/A33161-ND/1114899. [Accessed: 16-Dec-2019].
[25] “BMI160 Bosch Sensortec | Sensors, Transducers | DigiKey.” [Online]. Available: https://www.digikey.com/product-detail/en/bosch-sensortec/BMI160/828-1057-1-ND/6136308. [Accessed: 16-Dec-2019].
[26] “50-0015-00 Judco Manufacturing Inc. | Switches | DigiKey.” [Online]. Available: https://www.digikey.com/product-detail/en/judco-manufacturing-inc/50-0015-00/519PB-ND/307997. [Accessed: 16-Dec-2019].
[27] “WP710A10SRD/J4 Kingbright | Optoelectronics | DigiKey.” [Online]. Available: https://www.digikey.com/product-detail/en/kingbright/WP710A10SRD-J4/754-1895-ND/4098608. [Accessed: 16-Dec-2019].
[28] “TCA6507PWR Texas Instruments | Integrated Circuits (ICs) | DigiKey.” [Online]. Available: https://www.digikey.com/product-detail/en/texas-instruments/TCA6507PWR/296-22777-1-ND/1739925. [Accessed: 16-Dec-2019].
[29] “NHD-0216HZ-FSW-FBW-33V3C Newhaven Display Intl | Optoelectronics | DigiKey.” [Online]. Available: https://www.digikey.com/product-detail/en/newhaven-display-intl/NHD-0216HZ-FSW-FBW-33V3C/NHD-0216HZ-FSW-FBW-33V3C-ND/2773591. [Accessed: 16-Dec-2019].
[30] “PKMCS0909E4000-R1 Murata Electronics | Audio Products | DigiKey.” [Online]. Available: https://www.digikey.com/product-detail/en/murata-electronics/PKMCS0909E4000-R1/490-9647-1-ND/4878401. [Accessed: 16-Dec-2019].

[31] “IRF840PBF Vishay Siliconix | Discrete Semiconductor Products | DigiKey.” [Online]. Available: https://www.digikey.com/product-detail/en/vishay-siliconix/IRF840PBF/IRF840PBF-ND/812044. [Accessed: 16-Dec-2019].

Appendix

Item Name	Single Unit (USD)	10000 Unit (USD)	Savings (USD)
MSP430	2.69	11843	15057
ED20DT	0.26	1056	1544
BMI160	4.98	22176	27624
LCD (NHD)	12.15	87505.2	33994.8
LCD pins	1.38	6626.75	7173.25
LED Driver	1.58	6604.4	9195.6
LED	0.67	1372	5328
Battery	9.95	99500	0
Battery Header	0.17	604.35	1095.65
Battery Charger	4.68	23507.5	23292.5
JTag Header	1.36	6259.51	7340.49
Button	1.48	8910	5890
USB Port	0.43	2699.4	1600.6
Voltage Regulator	0.37	791.7	2908.3
330 Ohm Resistor	0.1	138.51	861.49
36k Ohm Resistor	0.1	138.51	861.49
1k Ohm Resistor	0.1	138.51	861.49
47k Ohm Resistor	0.1	138.51	861.49
1uF Capacitor	0.55	1881.4	3618.6
0.1uF Capacitor	0.25	603.8	1896.2
0.01uF Capacitor	0.26	627.9	1972.1
4.7uF Capacitor	0.67	2268.8	4431.2
47uF Capacitor	0.33	776.9	2523.1
10uF Capacitor	0.22	449.7	1750.3
2.2nF Capacitor	0.29	702.9	2197.1
1.5uH Inductor	5.76	30709.6	26890.4
MOSFT	1.62	6398.6	9801.4
Buzzer	1.5	8222.5	6777.5
		Total Savings	Savings Per Device
		207348.05	20.734805

Figure 16 Part Prices