Head Impact Telemetry
Data Logging System

ECE 445 - Design Review - Spring 2017

Contributors:
Matt Hebard | Matt Schafer | Evan Qi

TA:
John A. Capozzo

Table of Contents

Introduction

In the United States, the entertainment industry is vastly dominated by sports. Many of these sports are high impact sports, and the risk of injury must be considered. Football, lacrosse, and hockey, as well as many others, are among these high impact sports. According to the British Journal of Sports Medicine, there are an estimated two to three million cases of concussions in the States alone per year [1]. Most of these concussions are unreported and untreated. If the risk of concussion for each player were calculated in real time, then coaches and officials could be notified of any potential injuries as they happen, and medical attention could be given immediately. Many products introduced today analyze only the force applied to the head in these collisions, but do not focus on how the collision affect the different parts of the body. This data detection cannot give any insight to what is actually happening inside the body to better understand the effects of collisions to the head.

The purpose of the Head Impact Telemetry System (HITS) is to monitor the acceleration of the cranium. Upon a sharp impact, the accelerometers will show a spike in acceleration. The device will have a preset threshold beyond which any acceleration higher than the threshold poses a risk for a concussion. At the same time, the HITS has a heart rate monitor as well to track the heart rate before and after impacts to provide insight on the effects of concussions on the heart. These two sensors are to be time locked before being stored to the SD card, allowing the two data sets to be relatable to each other when reviewed at a later time. These two data sets could be collected separately, for example using a *Garmin Heart Rate Monitor* as well as current head acceleration equipment already in commercial use, however, by collecting these data sets on a single device we can reduce size and cost, as well as eliminating additional technology needed to analyze these separate data sets together. We will be creating a device that coordinates acceleration data with heart rate data on a single compact device to make analysis immediate and simple for the user to study this information. Once the player has received medical attention, the time locked data can be transferred to an interface for a researcher to study the variations in pulse. Ideally we would like to look at the data in real time, but for the purposes of our project, we are aiming to detect changes in head acceleration and how these impacts can affect the health of the subject.

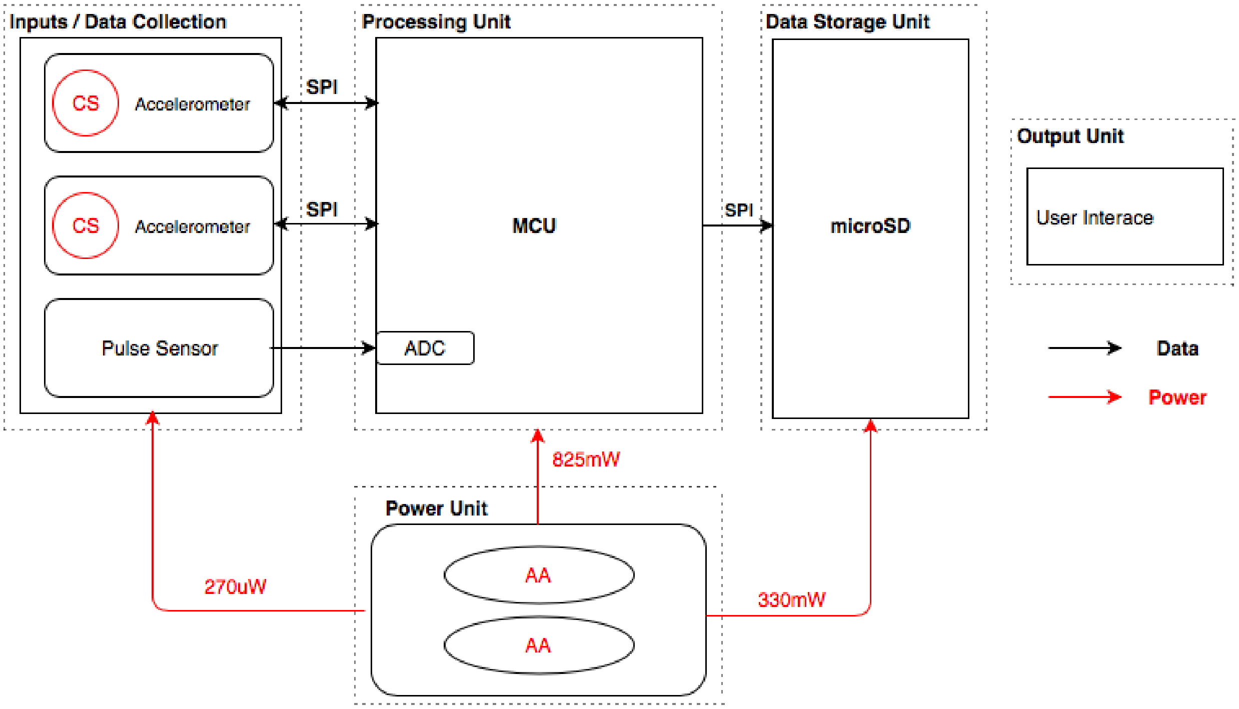
- **High-level requirements list:**
    - Accuracy: The accelerometer must detect acceleration with an accuracy above 85%. Heart Rate monitors in current commercial use have been tested to be no more than 4 bpm error when compared to a medically graded ECG Q-test machine[10]. We want our heart rate sensor to be able to have just as accurate if not better accuracy as these commercial devices. By comparing our heart rate data to a simultaneously equipped commercial product, we can see our heart rates accuracy.
    - Wearability: The entire device will be worn on the body and must be reasonably small and light as to not obstruct movement or throw off the player's center of gravity. The PCBs will not exceed 5.8x8.6x1.9 cm and weigh less than 300 grams.
    - Scalability: The design will be compatible for the user to add more accelerometers to the sensor array for more accurate and detailed data. Our project will act as a proof-of-concept for two accelerometers, but it will support as many as needed, as long as all

sensors in the array feed into the same microcontroller so that all data streams can be time locked together.

- Power: The device must, at the minimum, carry enough battery life to last the duration of a professional sports game, so a minimum of three hours.

Design

Figure 1: HIT Data Logging Block Diagram



Block Diagram Overview

The system above can be broken down into four separate subsystems: inputs/data collection, processing, storage, and output. Moving from left to right: the input/data collection module consists of three sensors: two accelerometers (placed behind each ear), and a pulse sensor (that will be placed on the individual's neck or ear). The processing unit consists of our microcontroller. The third unit in the block diagram is the storage unit. This is where all the data will be located. This is a removable micro SD card mounted on the *printed circuit board*. The last section of the block diagram is not entirely in our design, but we wanted to include it to indicate that we will have an output monitor when we show our *proof-of-concept* to display all of the data we capture.

1. Sensors

The HIT system will use a total of three sensors: two accelerometers and one heart rate sensor. The accelerometers will be used to gather information on the linear acceleration of the head at 2 different points on the head (behind each ear). The heart rate sensor is used for gathering an athlete's pulse in real time. These sensor’s outputs will then be fed as inputs into the PCB’s MCU where they will be processed using DSP filtering algorithms. The sensors will also be powered by the battery located on the printed circuit.

1.1 Accelerometers

The accelerometers that we will be utilizing in our design is Analog Devices's ADXL375 3-Axis, ±200g Digital MEMS Accelerometer. One reason this particular accelerometer was chosen was due to its high range of sensing. Studies show that concussions can occur at cranial accelerations as low as 60*g*. Most other accelerometers do not reach such high values, and thus an accelerometer that can sense beyond this value is essential to this design. This device also contains an internal ADC and a built in digital filter, which will prove to be valuable, saving us time as we do not need to filter the signal out ourselves. The device has a controllable output data rate and bandwidth of up to 1600 Hz. Sampling acceleration for concussions requires a bandwidth of at least 600 Hz [4]. For our desired design, power consumption is greatly reduced by limiting our ODR to 1600 Hz and bandwidth to 800 Hz. This bandwidth is an ample speed to detect all necessary data for logging a concussive blow and will also effectively reduce accelerometer current draw by 40%, from 145 uA to 90 uA. In addition, by decreasing the ODR of our accelerometer, we can also reduce a lot of the noise in our accelerometer data being sent to the MCU, allowing for less filtering required by MCU, further lowering power consumption. With a sample rate of 1600Hz, we need to communicate over SPI at a clock speed of at least 3MHz in order to not lose necessary samples from the accelerometers. We chose to use SPI communication due to our need to log data from multiple slave devices in real time. The timing diagrams for the SPI communication are shown below in Figure 2.

Figure 2: SPI Timing Diagram [3]

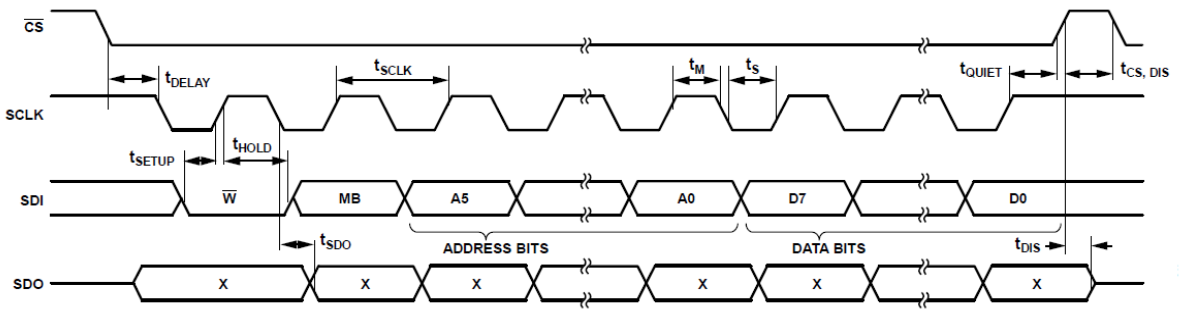


Figure 25. SPI 4-Wire Write Timing Diagram

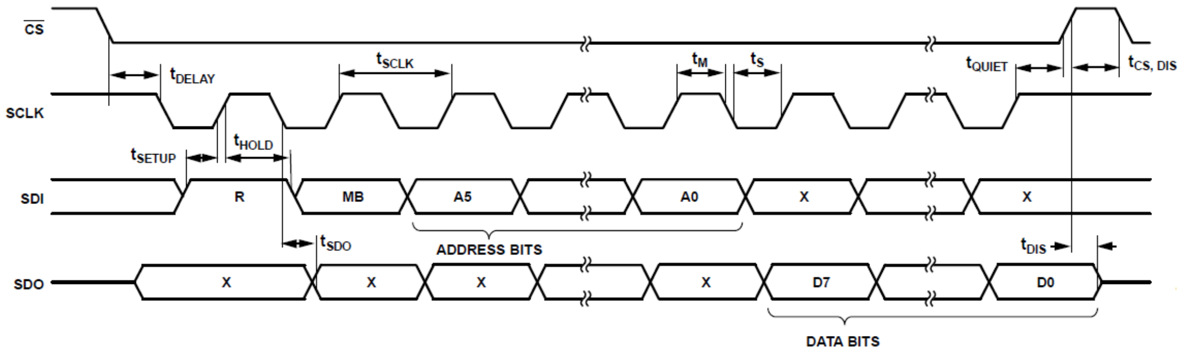
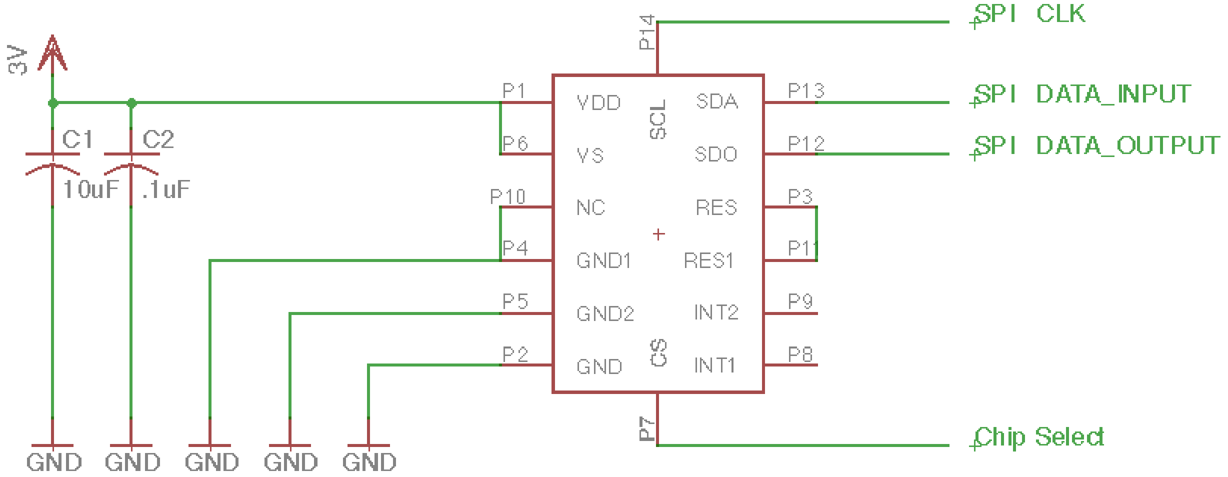


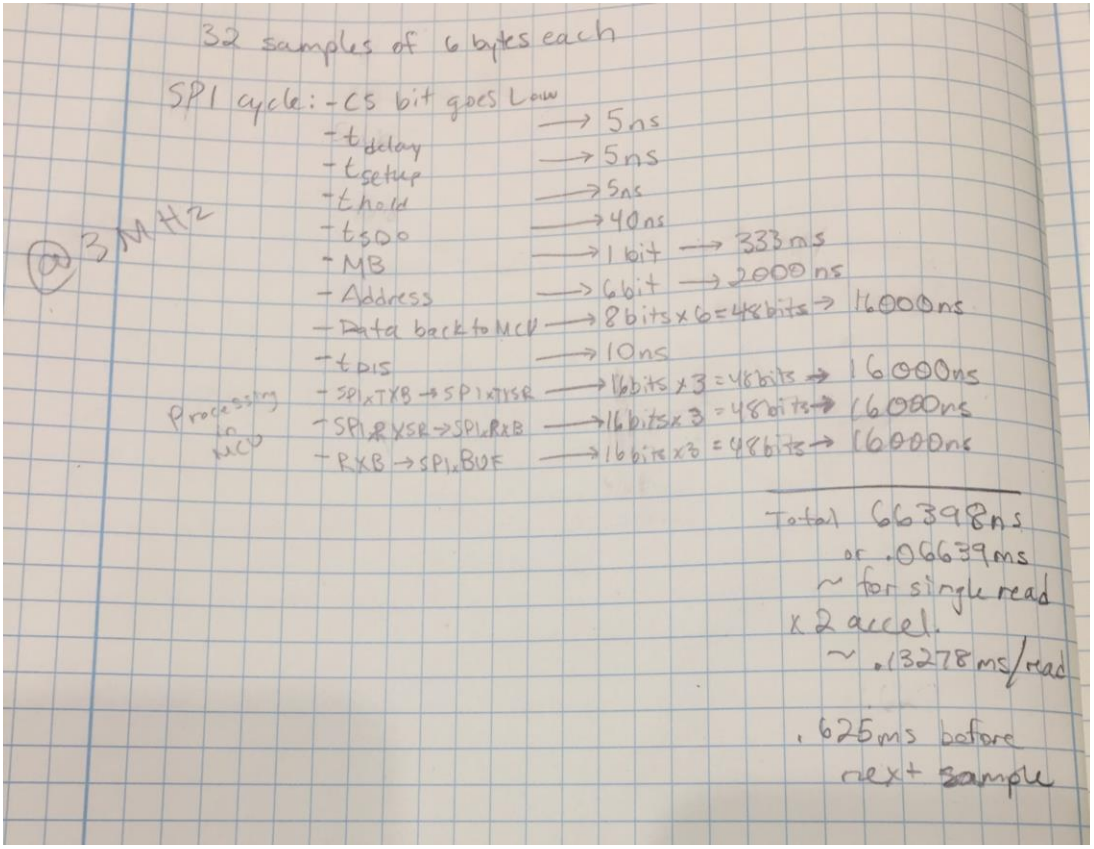
Figure 26. SPI 4-Wire Read Timing Diagram

Figure 3: Accelerometer Schematic



The calculations for timing and collection rate have been done based on the timing diagram for SPI protocol for the accelerometers shown above in figure 2. Below in figure 4 shows the calculations to determine that both our accelerometers should be able to read to the MCU in roughly .133ms.

Figure 4: Timing Calculations for SPI reads on both Accelerometers



32 samples of 6 bytes each

SPI cycle: - CS bit goes Low → 5ns
- t_{delay} → 5ns
- t_{setup} → 5ns
- t_{hold} → 40ns
- t_{SDO} → 1 bit → 333ns
- MB → 6bit → 2000ns
- Address → 8bits x 6 = 48bits → 16000ns
- Data back to MCU → 10ns
- t_{DIS}

@3MHz

Processing in MCU
- SPIxTXB → SPIxTXSR → 16bits x 3 = 48bits → 16000ns
- SPIxRXSR → SPIxRXB → 16bits x 3 = 48bits → 16000ns
- RXB → SPIxBUF → 16bits x 3 = 48bits → 16000ns

Total 66398ns
or .06639ms
~ for single read
x 2 accel.
~ .13278 ms/read

.625ms before
next sample

Requirements:	Verification:
(1) Accuracy: The accelerometers are 85% accurate with noise no higher than 5mg/sqrt(Hz). A concussive blow has been suggested to occur around roughly 60gs and above. If we have a tolerance of 85% and above we can account for any blow above 60gs by setting our threshold to 50gs.	(1) We will build a spring-based device to oscillate the sensor back and forth. We will compare the calculated acceleration with the detected acceleration and verify that error is no less than 15%.
(2) Data Collection: Collect data from each accelerometer in roughly .1ms or a rate of 10000Hz	(2) Use an arduino with same code as on MCU to output time stamped data to show that our code is designed to properly output 3,000,000 sets of data in 300 seconds (Amount of time required in our design).

1.2 Pulse Sensor

The Pulse Sensor we would like to utilize is the Amped Pulse Sensor, SEN 11574, provided by adafruit. Our reasoning behind choosing this device is its low cost and low power consumption. The pulse sensor relies on an LED and a surface mount light photo sensor to detect the reflected light off the surface of the skin to produce a photoplethysmographic signal. The output is sent through an op-amp where it then outputs an analog signal to our MCU's ADC module. Based on research studied it is necessary to sample heart rate sensors at a minimum of 500Hz to collect viable data [3]. For our design we have chosen a higher sample rate of 800Hz due to our MCU's high speed capability.

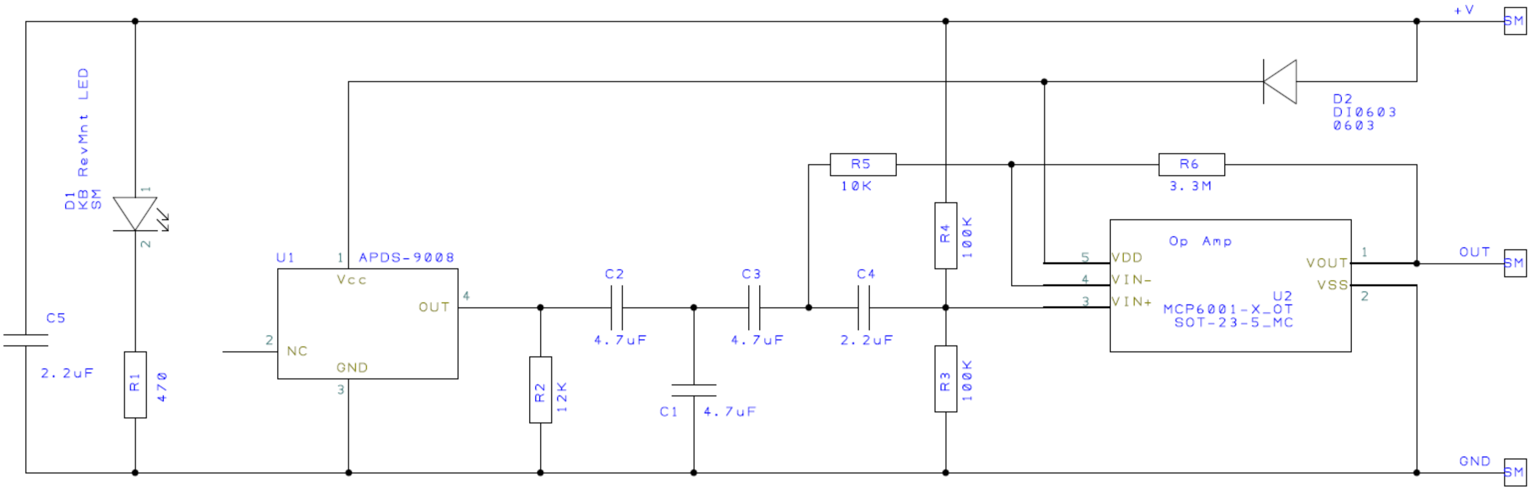
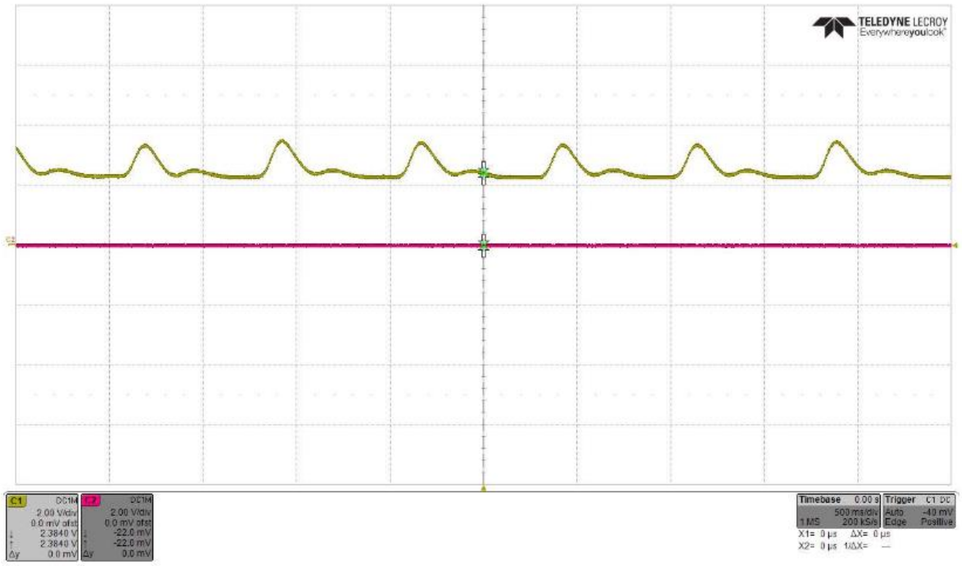


Figure 5: Pulse Sensor Schematic [3]

For processing our pulse sensor data, our goal is to calculate and log beats per minute (BPM) based on averaged Inter-Beat Interval time (IBI) calculated from the pulse sensor output signals. Before we can do any BPM calculations, we will be applying a periodic moving average filter (PMAF) on the MCU to remove motion artifacts, since our project will be containing a large amount of motion from impacts[8]. Figure 6 below shows the waveform generated from sensing pulse on the ear, while figure 7 shows simulation data of motion artifacts and eliminating it through PMA filtering to allow for a more accurate BPM calculation.

Figure 6: Pulse Waveform (X-axis: 500ms/div, Y-axis: 2V/div)

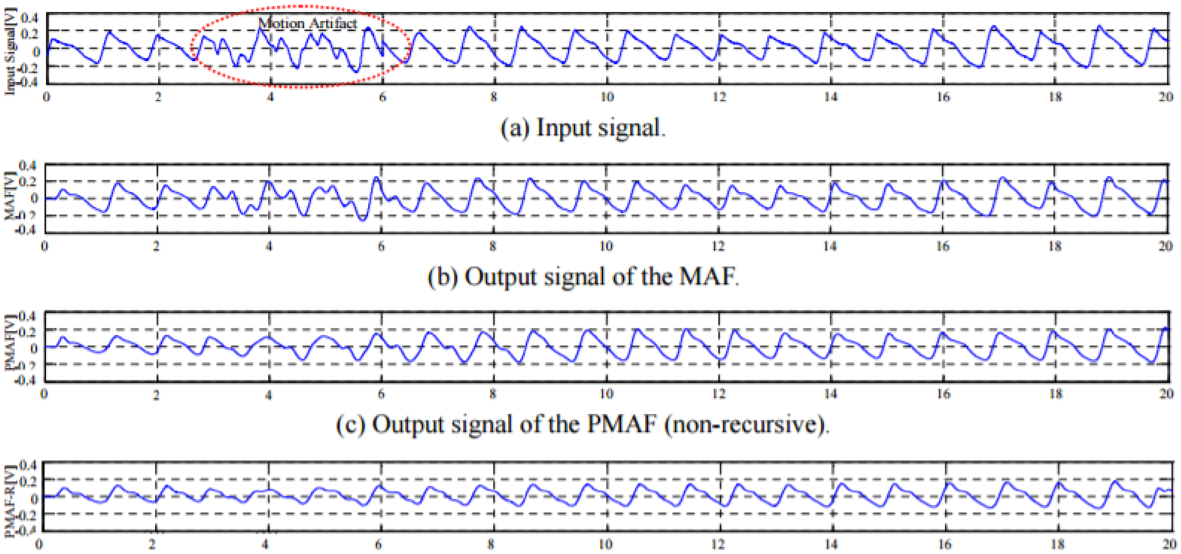


C1 DC1M
2.00 V/div
0.0 mV ofst
2.3840 V
2.3840 V
Δy 0.0 mV

C2 DC1M
2.00 V/div
0.0 mV ofst
-22.0 mV
-22.0 mV
Δy 0.0 mV

Timebase 0.00 s
500 ms/div
1 MS 200 kS/s
X1= 0 µs ΔX= 0 µs
X2= 0 µs 1/ΔX= —

Trigger C1 DC
Auto -40 mV
Edge Positive



(a) Input signal.

(b) Output signal of the MAF.

(c) Output signal of the PMAF (non-recursive).

(d) Output signal of the PMAF (recursive).

Figure 7: MatLab Simulation of PMAF Filtering [8]

For our BPM algorithm we look to calculate the time between actual pulse events, which will be detected as 50% of the amplitude wave peak. Initially this 50% amplitude will be set at V/2, but will be continually updated by averaging previous peak and trough values and recalculating the midpoint between these two averages. When the pulse sensor surpasses this 50% amplitude threshold, we will have a delay of roughly 240ms before looking to detect the next pulse, this will allow for a maximum heart rate of 250 BPM, well outside the viable heart rate range.

Requirements:	Verification:
(1) Data Collection: Sample HR sensor output at a rate of 800Hz. (2) Accuracy: Heart Rate Sensor must be within 4 bpm of a commercially used Heart Rate Sensor.	(1) Use an arduino with same code as on MCU to output time stamped data to show that our code is designed to properly output 240,000 samples in 300 seconds (Amount of time required for our design). (2) We will connect our sensor as well as a commercial HR sensor to a subject and record heart rate throughout a 10 minute running exercise. The recorded information will be analyzed along side our HR sensor to ensure 4 bpm accuracy.

2. Processing

2.1 Microcontroller

The microcontroller unit (MCU) serves as the processing unit, temporary storage, and communications master needed in this design. The accelerometers and pulse sensor will be hooked up to the MCU through SPI protocol. Given that the accelerometers are 16-bit, we will utilize a 16-bit MCU as well. When the threshold is detected by the accelerometers, it will send a signal to the microcontroller, which will read the data from the accelerometers' 32-level FIFO buffers, which each contains 32 of its most recent samples. This corresponds to 20ms of data, which exceeds the 8ms pre-collision time suggested by literature to acquire sufficient acceleration data [4]. At the same time, the MCU will pull heart rate samples from the previous 20ms from the cache and time lock it accordingly to the acceleration, and store the results to the micro SD card.

For the next 300 seconds, the MCU will continue to pull data from the accelerometers and heart rate sensor and store to flash memory after processing in real time. At a rate of 1600Hz, or a period of 20ms between each read, the microcontroller must read 32 samples of 3 dimensional points, or 96 16-bit numbers. Then, we need to process the heart rate data. The HR sensor will be sampled at a rate of 800Hz (after the 240 ms cooldown time), or a period of 1.25ms, which corresponds to 16 samples and read cycles per accelerometer read. These 16 samples will be filtered through fast convolution, so 256 cycles are needed for processing. Then, after these data values are calculated, they must be written to flash memory. Writing 32 samples of 88 8-bit characters each requires 1408 cycles. Adding all this together, we need 1776 operations per read, or a grand total of 88800Hz. Since we do not want to saturate the timing but rather aim for 70% usage, 130kHz is the bare minimum for our processor. However, since the accelerometers require at least 2MHz for an output data rate of 1600Hz, we will use one that is at least 3MHz to allow for tolerance.

The MCU that we chose in the end was Microchip's PIC24FJ128GA202. This unit has a clock speed of up to 8MHz, and carries three SPI I/Os for the accelerometers, as well as an analog peripheral for the heart rate sensor. This device also contains 8kB of RAM that we will be using for calculations and cache storage.

Requirements:	Verification:
(1) The MCU must be able to store all information from the accelerometers and process heart rate data quickly enough in between each accelerometer read.	Each row of the final stored data must be time locked within a 200 µs window, with a 100% sample throughput. We will manually give a triggering concussive blow to a ball with the device attached and ensure that 192,000 rows are recorded in the storage at the end of 120 seconds.

Figure 8: Microcontroller Schematic

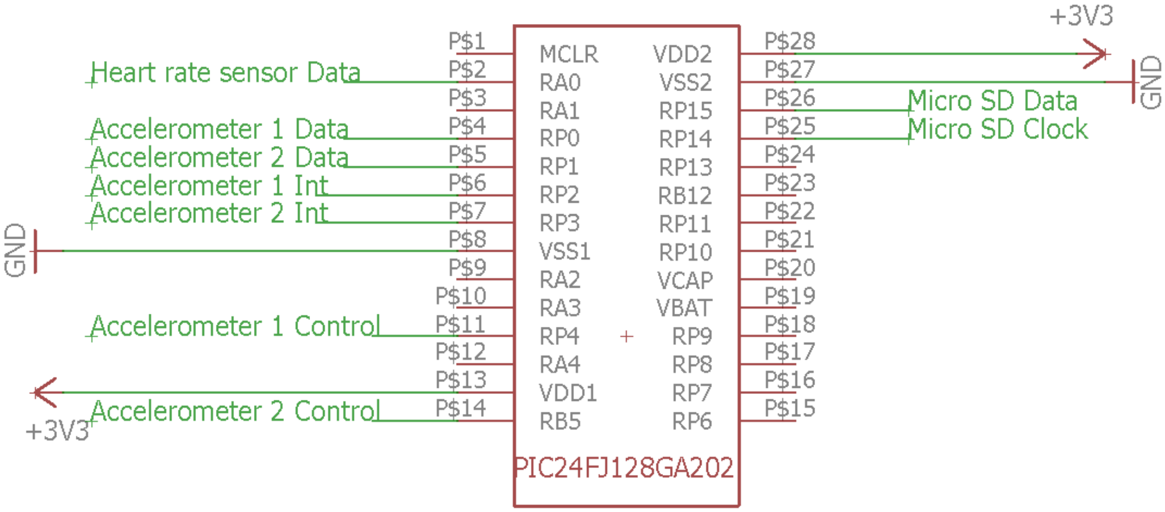
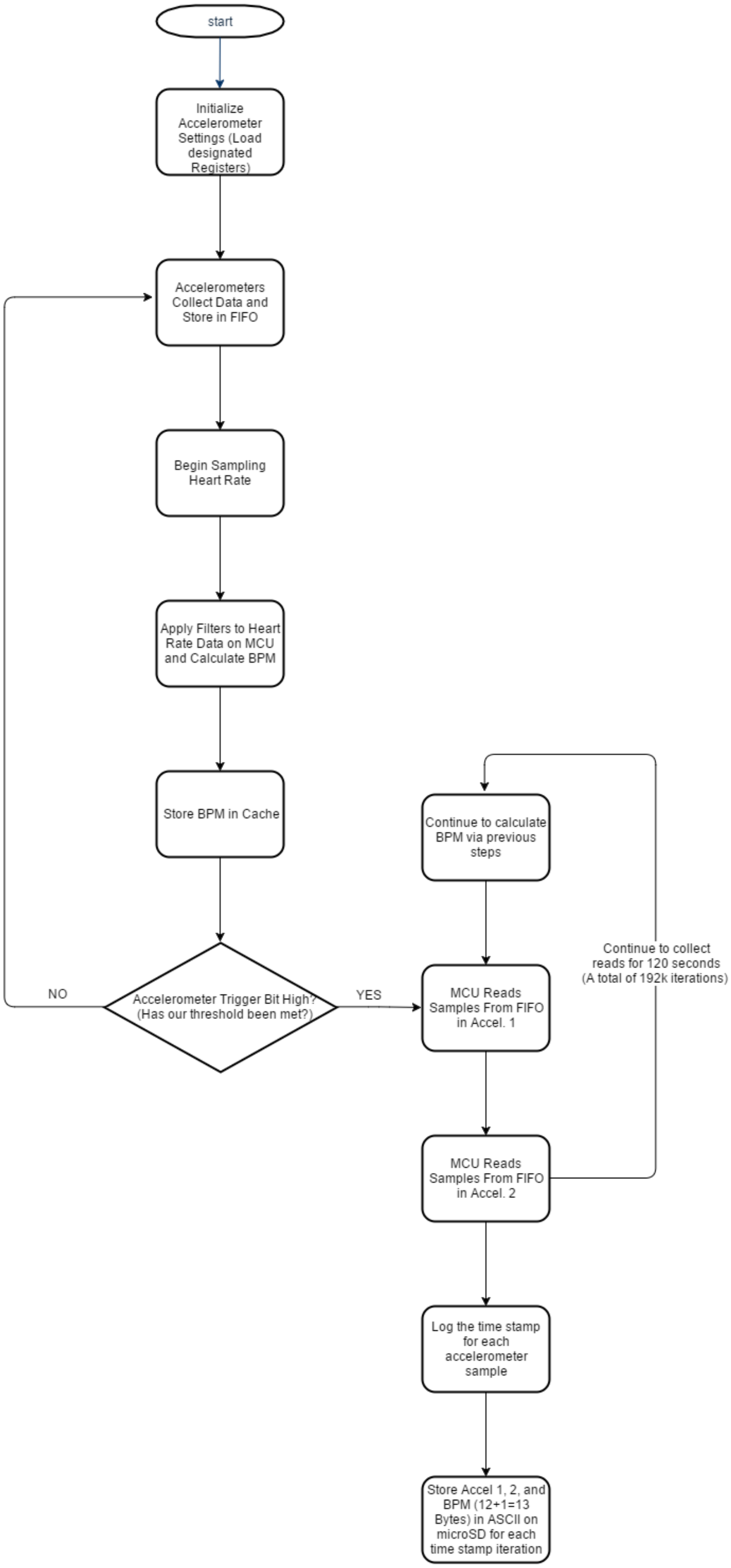


Figure 9: System Flow Chart


start
Initialize Accelerometer Settings (Load designated Registers)
Accelerometers Collect Data and Store in FIFO
Begin Sampling Heart Rate
Apply Filters to Heart Rate Data on MCU and Calculate BPM
Store BPM in Cache
Accelerometer Trigger Bit High? (Has our threshold been met?)
NO
YES
Continue to calculate BPM via previous steps
Continue to collect reads for 120 seconds (A total of 192k iterations)
MCU Reads Samples From FIFO in Accel. 1
MCU Reads Samples From FIFO in Accel. 2
Log the time stamp for each accelerometer sample
Store Accel 1, 2, and BPM (12+1=13 Bytes) in ASCII on microSD for each time stamp iteration

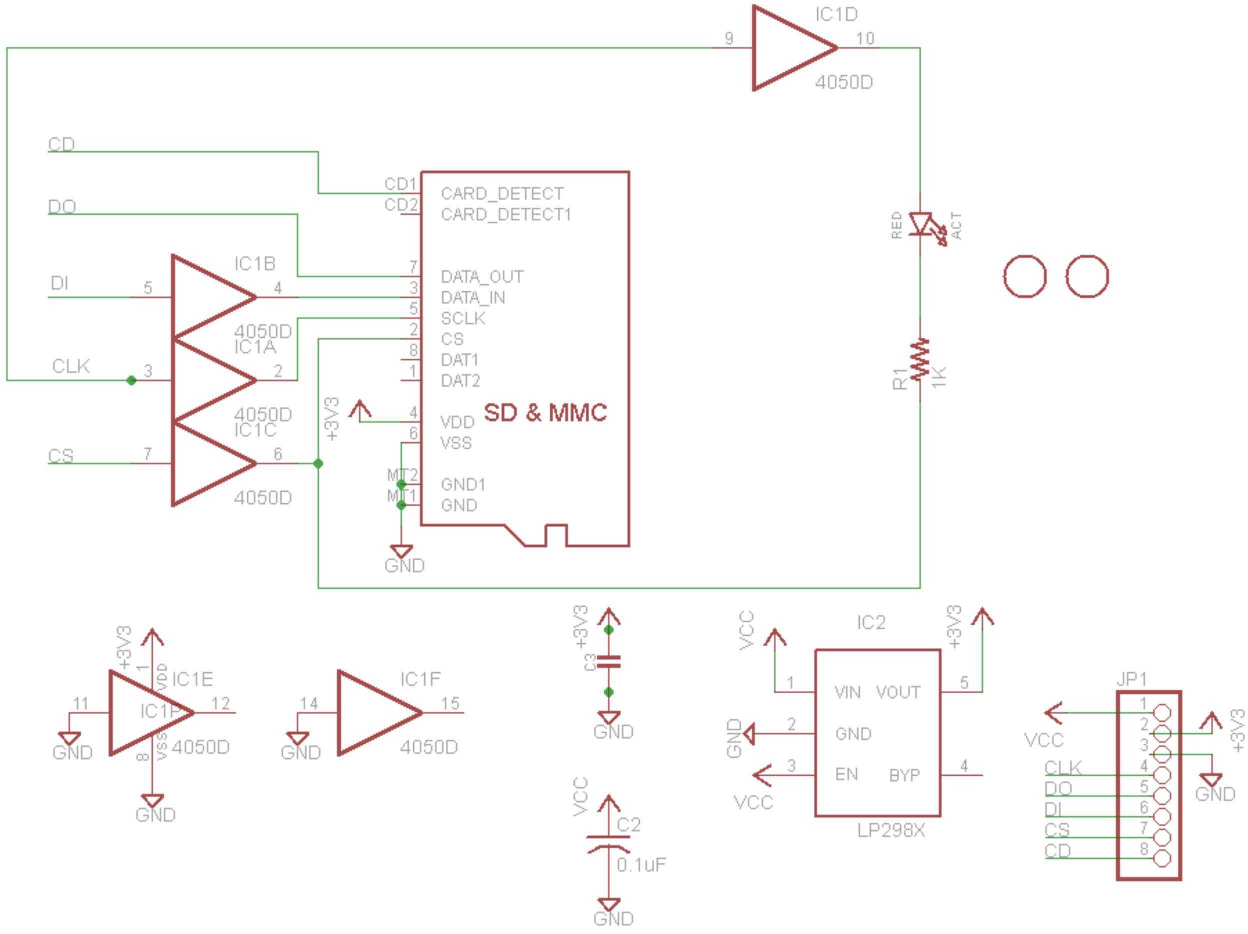
3. Data Storage

Given that our device will be a data logging system, storage will be a necessary component. The purpose of our system is to store all the information from the sensors to flash memory, so that it may be reviewed at a later time. A micro SD card with FAT16 format will be used.

Micro SD card holder

The card will be securely held in place and interfaced with using a breakout board. We will design this part based off of Adafruit's breakout board+. This part will communicate with the card through SPI protocol. Since SD cards require a strict 3.3V input, we will have to include a regulator to ensure that the voltage does not fluctuate.

Figure 10: MicroSD card holder schematic [8]



Requirements:	Verification:
(1) The micro SD card breakout board must be capable of writing to the micro SD with 100% accuracy. (2) The board must be able to write at 138kB/s.	We will write to the micro SD with a test file for the required amount of time at the calculated speeds and ensure that the stored data in the card is 100% matched with input data.

Micro SD card

The storage device we are using is going to be a micro SD card. We predict that 1GB will be more than enough, but larger and more costly cards are usable. The data will be stored in a .log file, with the sensors' data recorded in rows alongside the date and time in the following format: "hh:mm:ss.ssss - A +xxx.xxxx +yyy.yyyy +zzz.zzzz B +xxx.xxxx +yyy.yyyy +zzz.zzzz H hhh.h". The accelerations A and B would store the dimensional accelerations sensed by accelerometers a and b, respectively, in the units of g-force. The heart rate H would be stored alongside the accelerations in units of beats per minute. Since the micro SD will be written to only after the threshold acceleration is met, the data will be stored in portions of 300 seconds. The micro SD will be written to in real time, and so a row of 88 ASCII characters will be written at a rate of 1600Hz, or a data rate of 138kB/s. A Sandisk 4GB micro SD card will be used in our design, which will be enough for over 30,000 seconds or 500 hours of data.

Requirements:	Verification:
(1) Storage space must be able to hold data from the entire required interval of time.	After testing the writing speed of the breakout board above, we will make sure that there is storage room to spare.

4. Power

In figure 1 we have a block diagram of our HIT data logging system. The goal of this device is to secure enough power to endure the entire cycle of an athletic sports game, so a minimum of 3 hours. The two accelerometers will each carry with them their own CR-1025 coin cell to reduce the number of wires running up and down the body. The MCU, micro SD, and heart sensor all need to be supplied with power. We will be implementing a power system with the starting supply of two double AA Lithium (Energizer) batteries in series, with a nominal voltage of 3 volts. The MCU, micro SD, and pulse sensor all need a minimum voltage of at least 3 volts and preferably around 3.3 volts. In figure 11 below you can see the system we are implementing (without the three loads that would be connected in parallel to the 3.3 volt output). We are using a linear technology boost converter chip to step up our 3 volt input to supply our three loads with a consistent 3.3 volt output. The chip enables high load current at the output end, which is essential for our MCU and microSD because these two loads draw a significant amount of current (approximately around 100mA-200mA each). With an absolute maximum power draw of around 1.2 watts (see below for power breakdown), the AA batteries would be able to hold for about 6 hours, according to its datasheet. Since we will likely not be functioning at full power for the entire duration of the game, the battery life is expected to survive for much longer.

Figure 11: Power Schematic For Processing-Storage-Pulse Sensor Loads [11]

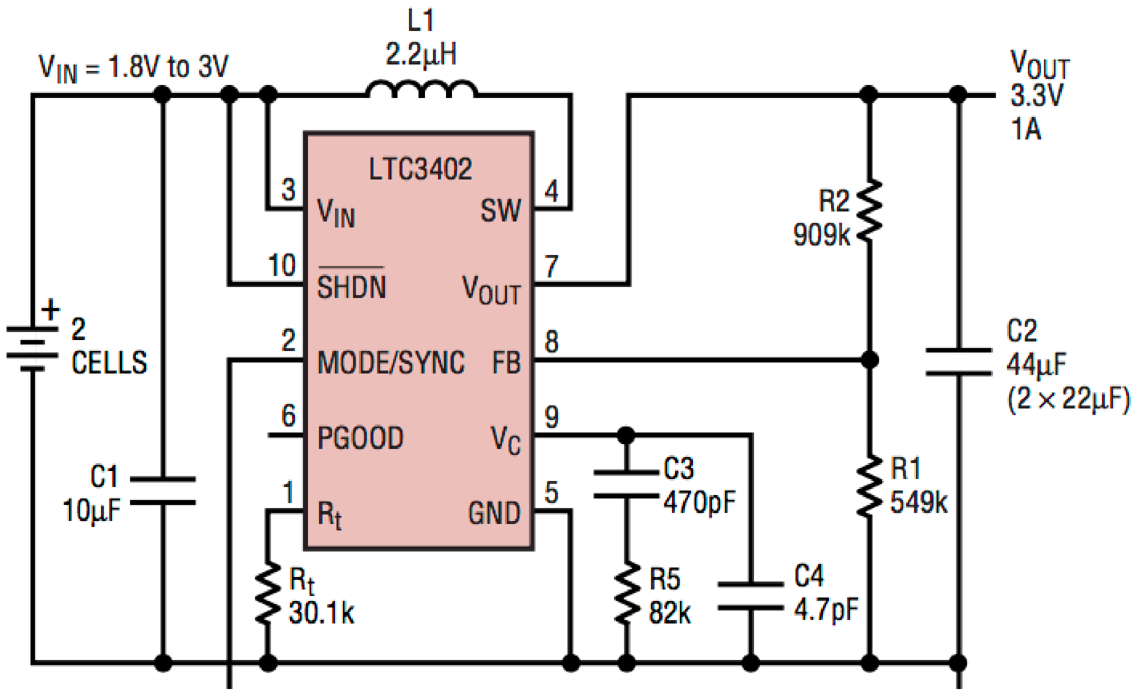
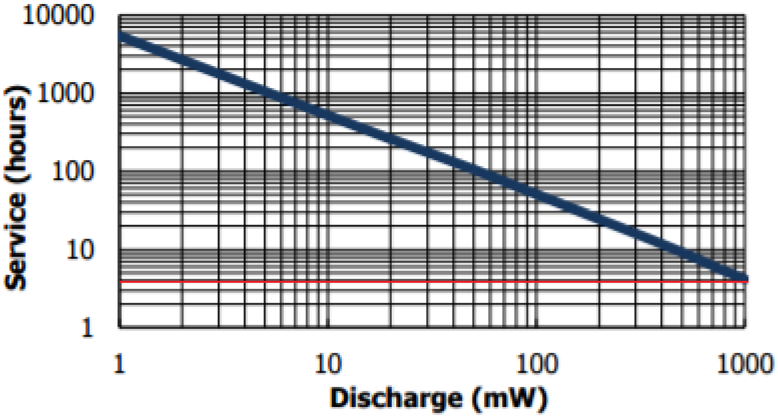


Figure 12: Battery Bank Expected Lifetime [4]



Sensors

Each accelerometer will be powered with its own CR-1025 coin cell. This coin cell carries a charge of 30mAh at 3V. Since the accelerometers running at 1600Hz will require 90uA of current at 3V, each accelerometer should theoretically consume 270uW and last for over 300 hours.

The heart rate sensor will pull power from bank of 2 AA batteries that will be mounted with the PCB. The heart rate monitor intakes 4mA at 5V, but has an option for a 3V intake. After fully testing the capabilities of the pulse sensor at 3V, we found that there were no drawbacks to using 3V versus 5V. Thus, we will power the pulse sensor directly from the 3.3V regulator, and expect a maximum power consumption of around 13mW.

Processing Unit

It is difficult to predict the power consumption of the microcontroller, but the datasheet claims an absolute maximum current of 250mA. Running at 3.3V, this gives an absolute maximum power consumption of 825mW. Given that we will not be using the microcontroller at its maximum speeds, since we do not need its full 8MHz potential in a two accelerometer case, the actual power consumption should be significantly less. The microcontroller will be pulling power from the battery bank through the voltage regulator.

Storage

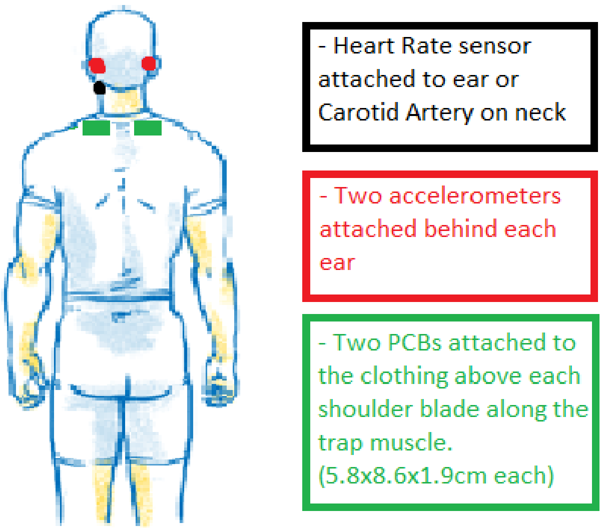
The SD breakout board that we choose to implement has two modes to choose from: default mode, which runs at 25MHz, and high speed mode, which runs at 50MHz. The default mode will be used, since the extra speed is not needed. A current draw of approximately 100mA is expected at a voltage of 3.3V, giving a maximum dissipation of 330mW for the storage unit.

5. Wearability

The first three blocks in the diagram above consist of our overall wearable design module. The first module (inputs / data collection) consists of three sensors: two accelerometers and one pulse sensor. The wearable design - specifically the three sensors - need to be small enough to fit comfortably yet strongly attached to the individual's skull. Each accelerometer will be mounted on its own miniature printed circuit board and will be powered by a coin cell. The accelerometers will be glued or taped on either side of the head, behind the ears, to catch linear acceleration as well as yaw and roll of the head. Our pulse sensor will be pinned to the ear with a clip. The HIT Data Logging System would ideally be able to scale to any athlete on this planet, but for the scope of this class we are going to focus on a few essential qualities of the device.

The second thing we are going to focus on is making this wearable design as light as possible. The main printed circuit board is going to contain the microcontroller unit for processing and storage. The power unit will be carried on a different platform. Design should allow for player to achieve full range of motion of the head, neck, and upper body. According to research, the optimal location to wear this device would be on the back, on the upper shoulder blades. The units on the shoulder blades should not exceed 5.8x8.6x1.9 cm. To prevent throwing off the player’s balance, the entire system should weigh less than 300 grams, or approximately the weight of an iPhone 6 on either shoulder.

Figure 13: Wearable Design Layout


- Heart Rate sensor attached to ear or Carotid Artery on neck
- Two accelerometers attached behind each ear
- Two PCBs attached to the clothing above each shoulder blade along the trap muscle. (5.8x8.6x1.9cm each)

Requirements:	Verification:
(1) The x-, y-, and z-dimensions of any unit carried on the shoulder blades should not exceed 5.8cm, 8.6cm, and 1.9cm respectively. (2) The entire device must have a mass of less than 300 grams. Each PCB should be at least 95% similar in weight to maintain athlete's balance.	The units will be measured with a ruler to determine the dimensions. The device will also be weighed using a scale.

6. Tolerance Analysis

The purpose of our device is to timelock accurate sets of acceleration and heart rate data. Since our highest sampling rate will be from the accelerometers at 1600Hz, or a period of 625 μs, in order for the data to be considered to be within the same timestamp, an absolute maximum in error would be ∓312 μs. Our device will aim to time lock the data within an error range of ∓100 μs, or within a period of 200 μs. Since each read of the accelerometer reads the buffer of 32 samples of 6 bytes each, an 8MHz processor should theoretically read the entire buffer in 12 μs. This means that the two accelerometers would have no issue in being time locked within the desired error range.

The heart rate sensor proves to be easier to implement within a tolerable error. The trigger mode on the accelerometer sends an interrupt signal once the threshold acceleration is met, which will be used to signal the microcontroller. At this moment in time, the current calculated BPM is logged along with the accelerations. Since the BPM is continuously updated in real time with every heartbeat, or approximately every 500 ms, the time lock can take the current BPM at any moment and consider it to be accurate.

Cost Analysis

Part	Cost
MCU	$4
Micro SD	$5
Accelerometer (4 x $8)	$32
Pulse Sensor	$0 (Acquired in ECE 445 Lab)
Lithium Double AA Batteries (pack of 4)	$15
Coin Cell Battery (x2 x $1.50)	$3
TOTAL COST:	**$59.00**

Name	Hourly Rate	Hours	Cost
Matthew Schafer	30 USD/Hr	250 Hrs	7,500 USD
Matthew Hebard	30 USD/Hr	250 Hrs	7,500 USD
Evan Qi	30 USD/Hr	250 Hrs	7,500 USD
Total	**90 USD/Hr**	**250 Hrs**	**22,500 USD**

Risk Analysis

The module with the highest risk within our design would be the sensors module. Whereas the other modules would likely have terminal consequences in case of failure, buggy sensors would be much harder to detect and troubleshoot. Despite the verification of the integrity of the sensors before implementing them into the system, continuous impact may cause deterioration. A failure in the sensors would cause the entire system to be useless, as the data it collects may not only be incorrect, but may even sabotage all prior research in the subject. To minimize this risk, we must decrease the probability of failure in the sensor module as much as possible. The sensors have to be secured as tightly as possible in place to prevent wiggling loose in extreme physical impacts.

Ethics and Safety

One ethical issue associated with our system is the intentional destruction of the device to prevent monitoring, a violation of code 9 in the IEEE Code of Ethics [4]. Given that the device will be mainly used for high impact sports, opposing players may specifically target the device in an impact. To avoid this issue, we will make the PCB as flexible yet sturdy as possible.

Since the device will be worn on the head, there is a risk of electrical shock. Our device will be powered by a battery, so the shock will not be lethal in any way, but the risk is still possible. The sensors we plan to use has a logic interrupt that we will ensure is functional to reduce the risk of electric shock.

Another possible source of injury from our device is burns, especially because of its wearable design. The device will be dissipating a maximum of around 1 watt, which will not be nearly enough thermal energy to burn the skin. In the case of an unexpected electric surge, we will implement a sacrificial fuse to break the circuit.

Schedule

Week	Matt Schafer	Evan Qi	Matt Hebard
2/27/17	Start running accelerometer lab tests	Finalize MCU schematics and finish PCB layout	Complete accelerometer PCB design and order boards
3/6/17	Complete Pulse Sensor tests	Make sure all parts have been ordered	Test power circuit design module
3/13/17	Update input / data all modules in design that need update	Run verification tests on MCU boards when they arrive	Power printed circuit boards should be ordered and shipped
3/20/17	Keep up to date on functionality of design throughout break!	Keep up to date on functionality of design throughout break!	Keep up to date on functionality of design throughout break!
3/27/17	Continue testing input / sensor unit to make sure we are hitting required accuracy	Start designing the DSP code for MCU and accelerometer functionality	Help with the C/HDL code for DSP protocol with Evan
4/3/17	Complete functionality testing should be done within each unit	MCU testing should be coming to completion	Help with the testing of complete functionality of the design

4/10/17	Functionality testing	Complete software algorithm	Functionality testing
4/17/17	Modularity testing	Continue help with functionality testing	Work on second half of modularity testing
4/24/17	Start Final Report	Start Final Report	Start Final Report
5/1/17	Finish Report	Finish Report	Finish Report

References

[1] Harmon, K. *et al*. "American Medical Society for Sports Medicine position statement: concussion in sport". *British Journal of Sports Medicine*. 47 (1): 15–26. doi:10.1136/bjsports-2012-091941

[2] Ramasamy, V. *et al*. "The Basics of Designing Wearable Electronics with Microcontrollers."

[3] Rijnbeek, P. *et al*. "Minimum Bandwidth Requirements for Recording of Pediatric Electrocardiograms." *Circulation* (2001). 3087-3090. doi:10.1161/hc5001.101063

[4] Wu, L. *et al*. "Bandwidth and sample rate requirements for wearable head impact sensors." *Journal of Biomechanics* **49 (**2016). 2918-2924. doi:10.1016/j.jbiomech.2016.07.004

[5] Rowson, S. *et al*. "Head Acceleration Measurements during Head Impact in Pediatric Populations."

[6] Gemperle, F. *et al*. "Design for Wearability."

[7] Life Span Fitness. Target Heart Rate Calculator: https://www.lifespanfitness.com/fitness/resources/target-heart-rate-calculator

[8] Lee, H. *et al*. "The Periodic Moving Average Filter for Removing Motion Artifacts from PPG Signals." *International Journal of Control, Automation, and Systems* **5** (2007). 701-706.

[9] "Target Heart Rates." *Target Heart Rates*. N.p., 12 Oct. 2016. Web. 28 Feb. 2017.

[10] Prospero, Mike. "Who Has The Most Accurate Heart Rate Monitor?" *Tom's Guide*. Tom's Guide, 01 June 2016. Web. 09 Mar. 2017.

[11] Corporation, Linear Technology. "LTC3402." *LTC3402 - 2A, 3MHz Micropower Synchronous Boost Converter* (n.d.): n. pag. *Http://cds.linear.com/docs/en/datasheet/3402fb.pdf*. Linear Technology. Web.