RCA: Real-Time Concussion Analyzer

Timothy E. Coyle, EE, Justin D. Kober, EE, Scott A. Rosa, CSE, and Kenneth W. Van Tassell, EE

Abstract—**We introduce RCA (Real-Time Concussion Analyzer), a real-time system that will allow football coaches to remotely monitor the forces of impacts experienced by players during game play. This system will provide the likelihood that a player has experienced a concussion, allowing coaches to make more informed decisions pertaining to player safety. RCA incorporates an array of accelerometers placed inside each player's helmet. The sensor data from each helmet is wirelessly transmitted to an Android device, where an application will query a player database on a server, and determine the likelihood of concussion.**

I. INTRODUCTION

CONCUSSIONS in sports have become a growing concern in recent years, despite advancements in safety equipment. According to the Center for Disease Control and Prevention, a concussion is a type of traumatic brain injury, or TBI, caused by a bump, blow, or jolt to the head that can change the way your brain normally works. "While all concussions are serious, most occur without loss of consciousness." [1] Recognition and proper response to concussions when they first occur can help prevent further injury or even death. Current concussion detection relies on a coach to constantly check a player's reported symptoms after each collision. They monitor a wide array of symptoms from memory to balance. However, many players will try to mask their symptoms [2], [3]. This problem is not localized to any age of player, though RCA is aimed at the high school level. In June of 2012, over 2000 former players filed a lawsuit again the NFL claiming that the league hid the link between football-related head trauma to permanent brain injuries [4]. This lawsuit brought concussion awareness to the forefront of sports medicine.

Preventing concussions in football seems to be an extremely challenging goal, but detecting them is slightly less challenging. RCA is a system intended on detecting concussions in the hopes that a coach will remove an injured player before they may suffer any further damage. This problem has been addressed within the last decade by not only researchers but also private companies. There have been a few systems developed to address this, most notably the Head Impact Telemetry (HIT) System [5]. The HIT System takes a similar approach to this problem, utilizing an array of linear accelerometers with wireless transmission to an off-field base station computer. This is a real-time system, though it lacks portability at the base station, is not cost efficient and provides only data on forces; there is no likelihood of a concussion presented. A system such as RCA could possibly influence how the game of football is played. If such a system was to be employed, players may find themselves removed from games more frequently. This could lead to opposing teams targeting key players knowing that the key player would have to be removed after receiving a certain concussive force.

To begin the design of RCA, we developed system specifications by reviewing the current solutions and their limitations. None of the current systems had a way to adapt for multiple sub-concussive impacts, which may be as dangerous as a single serious concussive impact. With this in mind, RCA will not only have wireless transmit capability, but also have the capability for wireless receiving. From their data sheets, the combination of the sensor network, processor, and radio in the helmet are restricted to using 302 mA and 5 V [6], [7], [8]. This means RCA will have a maximum power consumption of 1.51 W. The Android device has its own internal battery and is not included in this power analysis. RCA sensors will be installed in a player's helmet before the game in such a way that the player will not notice the sensors. There will be a threshold setting in a microcontroller, imbedded in the helmet, to account for minor accelerations that are not related to impacts. In the event of an impact, this threshold will be broken and trigger transmission of the impact data to the Android device. The device will then query a player database to retrieve all the history of recent impacts the player suffered, and then will compute the probability of a concussion. The coach will receive an alert with the player name and number, as well as the probability of concussion and show the impact location. This application on the device will have a user menu for the coach to add or remove players, as well as make notes and query the data after the games. Table I shows a list of specifications.

TABLE I
SPECIFICATIONS

Specification	Value
Weight	<5% increase (typically 102 grams)
Range	25 m
Response Time	<2 s
Battery Life	>5 hours
Cost	<$5000 for full team of 52 players
Power Consumption	<2 W
Acceleration Range	+/- 70 g
Sensitivity	Only measure actual collisions
Durable Packaging	Stable and waterproof

T. E. Coyle from Easthampton, Ma (e-mail: tcoyle@student.umass.edu)
J. D. Kober from Easthampton, Ma (e-mail: jkober89@gmail.com)
S. A. Rosa from Peabody, Ma (e-mail: sarosa@student.umass.edu)
K. W. Van Tassell from Chelmsford, Ma (e-mail: kvantass@student.umass.edu)

II. DESIGN

A. *Overview*

Our approach to this problem is to develop a rugged sensor network to be deployed in a helmet. This network will consist of six accelerometers, a microcontroller, a wireless radio, and a battery to supply power. We are currently evaluating single-axis accelerometers for the sensors, and have ruled out the use of gyroscopes after speaking with Professor Steven Rowson from Virginia Tech. Professor Rowson explained keeping the gyroscopes oriented was proving to be technically challenging, and he suggested not including them due to the amount of time it would take to figure out how to best stabilize them. From the linear accelerations that these sensors collect, we will be able to calculate rotational accelerations, which are needed for the calculations. We decided on using an 8-bit microcontroller as our impact data processor instead of an FPGA or other means, due to our familiarity with these devices. Microcontrollers have the ability to interface with analog sensors, process information, and transmit data to other devices. We will interface a Bluetooth radio with the microcontroller, as this type of radio can easily interface with an Android device. A Bluetooth radio provides a proof of concept, but with range limitations of these radios we are considering XBee radios, which have longer ranges, but are not readily interfaced with an Android device. As mentioned previously, we have selected an Android device for our user interface, as Android is the most prevalent operating system for smartphones [9] and it supports open source programming. To store the data and player history, we will develop a server with two separate databases, as to not overload the device's memory. We had also initially considered a base station type of approach with a receiver and a laptop, but with the portability requirement we decided to make the system as mobile as possible.

RCA includes three main blocks Fig.1:

1) Impact Data Collection: An array of sensors will be placed in the helmet, along with a microcontroller, Bluetooth radio and power supply. Together, these devices will detect an impact and transmit the data to be processed.

2) Data Analysis: This block is responsible for the calculations of risk, as well as storing and receiving the raw data and player information. It needs to communicate with the Android device and the backend server. By having the data analysis preformed on the Android device, we can minimize the processing done in the Impact Data Collection block. This leads to a low-profile system on both the player and coaches side, keeping the increase in helmet weight under five percent

3) User Interface: The user interface will be an Android device. Here, the coach can input player rosters, retrieve information about the impacts a player has sustained, and receive real-time alerts with the probability that a player has suffered a concussion. It needs to be able to communicate with the Impact Data Collection and Data Analysis blocks. It will do this over Bluetooth and the Internet respectively.

By utilizing Bluetooth and Android devices we can keep the cost to a minimum, as most users will have a capable Android device. The microcontroller and Bluetooth draw the most current, and even these devices can only draw 300 mA combined when transmitting [7], [8]. These three main blocks are then divided among the team into four sub-blocks.

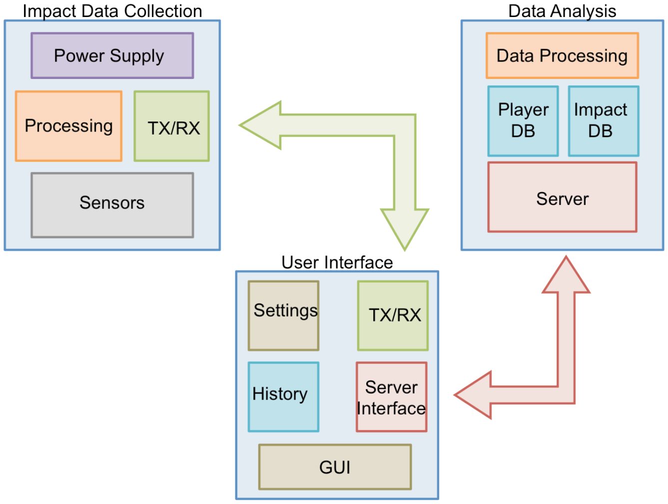


Fig. 1. This RCA Block Diagram shows the organization of the project.

B. *Sensor Network and Power Supply*

This sub-block of the Impact Data Collection block is concerned with measuring forces and powering the electrical components in the helmet. Sensors will be placed inside the helmet as close to the player's head as possible, for the most accurate measurements. The sensors we will be using are micro-electro-mechanical systems (MEMS) accelerometers. The ADXL78 [6] is a low powered, single-axis, MEMS accelerometer, with a tolerance of +/- 70 g. Six of these sensors will be strategically placed to capture the x-axis, y-axis, and z-axis of a player's head. These sensors need to be tested thoroughly to learn everything possible about how the sensor works, our initial concern is response time of the sensor. During testing, we will also learn how the sensor responds to multiple impacts of the same force, impacts from different directions, and what kind of realistic forces we can apply for demo day. Preforming the same test multiple times will also prove the sensor is functioning properly. We need to understand how one sensor works in order for us to build the network of sensors in the helmet.

The sensor we are currently testing is the ADXL 193 [10]. This sensor is a low powered, single-axis, MEMS accelerometer, with a tolerance of +/- 250 g. We had initially selected this sensor due to the large range or its tolerance. From research at Virginia Tech [15] they found the maximum linear acceleration a player experienced over the two year study was slightly less than 200 g's. The 193 and 78 differ in sensitivity, tolerance, and power levels. We decided to switch to the ADXL 78 due to its lower sensitivity and range. For our prototype we will not be inflicting extremely high g forces and thus do not need the ADXL 193. With the sensitivity level of the ADXL 78 being over three times lower, 27 mV/g as opposed to the 8 mV/g, than the ADXL 193 [6] [10], our system will be less susceptible to noise.

The experiment we conducted utilizes a basic pendulum to generate an impact of known magnitude. The sensor is placed, in the helmet, so it will measure in the direction of the swinging pendulum. The pendulum will impact the helmet giving us a measurement of the impact. This measurement is sent to the processor and transmitted to a computer for analysis, which will show that we can repeatedly inflict an impact with the same magnitude.

Using a power budget and energy analysis we can determine a type of battery to use; our initial thought is to use a coin cell battery because of it small weight and size. We confirm this energy analysis by measuring the changing voltage across a small resistor in series with our system, and then using Ohms law we found the current and calculated power. While we are connected to the circuit, we also can connect the sensor directly to the oscilloscope to measure how long the sensor takes to reach its peak acceleration value, see Fig. 2 below.

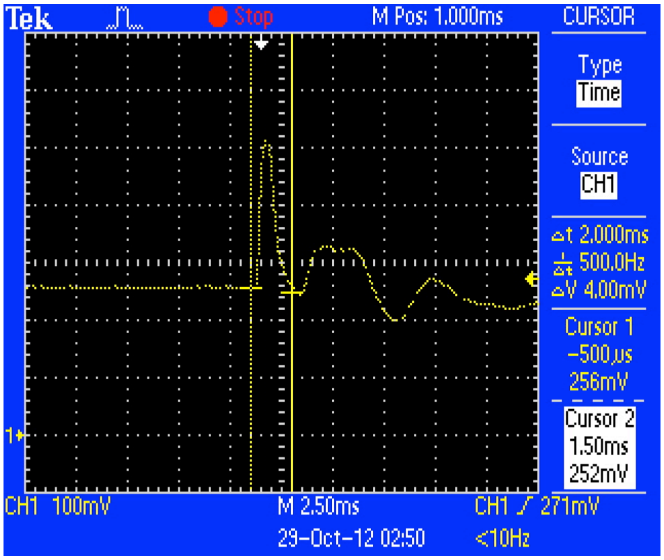


Fig. 2. This shows the peak acceleration of ADXL193 takes 1 ms.

C. *Impact Processing and Communication*

As a sub-block of the Impact Data Collection block, the impact processing and communication block will be responsible for detecting a collision and then transmitting the sensor data to the User Interface block. At the heart of this sub-block is a microcontroller that is used to process the incoming signals from the accelerometers described above. Once the signals have been processed by the microcontroller they will be either discarded, or in the case of a collision, they will be serially transmitted to a Bluetooth radio. This radio will communicate the impact data to the user interface.

The microcontroller selected for this block is the ATmega32U4 by ATMEL [7]. This microcontroller was selected for RCA after careful review of its features. There will be up to six analog signals coming from the sensor array that will need to be converted to a digital signal for transmission. This microcontroller has 12 10-bit ADC (analog to digital converter) channels, with six being used for the analog to digital conversions of the sensor signals. The ATmega32U4 also has a programmable serial USART, which will allow the communication between the microcontroller and the Bluetooth radio. There are multiple types of storage available if we need to hold the impact data on the processor for some unforeseen reason. All this is in a TQFP package, which will allow us to have a small circuit to fit inside the helmet.

The programming of this microcontroller in C is familiar, as we have use a similar 8-bit AVR microcontroller, the ATmega32, in previous course work. There are many features of this microcontroller that we will utilize for RCA: the ADC, USART, Timers, and Interrupts. While we had previous course work, we found online tutorials essential in configuring some elements of the ATmega32U4 [11], [12]. We use an external 16 MHz clock so that we can utilize the fastest prescaler of the ADC. With this prescaler, a single analog to digital conversion can be completed in 8 us, thus we can read all six sensors once every 48 us. To actually read the converted values in real-time, we interfaced an LCD with the ATmega32U4 so that we could output the conversion and read the value. To achieve maximum granularity the ADC was set to have all 10-bits of resolution, thus the outputs range from 0-1023. With 5 V applied to both the sensors and ADC each incremental value of the analog to digital conversion is weighted at 4.88 mV.

Initially the design for communication was going to be XBee radios, but due to our decision to have a scaled down prototype we chose to start the project with Bluetooth. The Bluetooth radio selected for this block is the BlueSMiRF Gold by Sparkfun [13], which uses the Roving Networks RN-41 Bluetooth module [8]. This Bluetooth radio is a class one Bluetooth device with an advertised range of approximately 100 m. We found this range to be quite exaggerated once the device was configured, and could only connect at a maximum range of 25 m. This is a Bluetooth version 2.1+EDR with built in error correction for the 8-bit packet transmission and 128-bit encryption. This module has a standard data rate of 300 Kbps.

To interface this radio with the microcontroller, we first had to initialize the USART feature within the ATmega32U4. Once we configured the Bluetooth radio and microcontroller, we tried to read the serial stream on a laptop terminal window. We could communicate between the microcontroller and the laptop over Bluetooth, although the symbols were being distorted. We eventually corrected the baud rates to 9600 and the Bluetooth connection was working. We plan on optimizing this connection to achieve the maximum throughput possible for RCA over the course of the project. We utilized our knowledge from ECE 563 to help understand how the Bluetooth modulation schemes could improve throughput, as well as ECE 374 to understand how to network the devices using MAC addresses.

Once we had completed the basics of this sub-block, an experiment was conducted to determine the accuracy to which the user can receive the sensor data. The output of the sensor was measured directly, while at the same time, the sensor data was recorded on the laptop after the Bluetooth transmission.

From this experiment we were able to learn that our sample rate was too low, one sample every 6 ms. We had conducted measurements of the sensor directly with an oscilloscope and found that the average response time to reach peak acceleration was approximately 1 ms. By enabling double speed operation of the USART and optimizing the baud settings of the ATmega32U4 and the BlueSMiRF we were able to achieve a sample rate of 1 ms. This sample rate is not optimal as we would like to have a sample rate of 50 us for the entire completed sensor array. With the current sample rate we achieve an accuracy of +/- 4 g's, this was determined through statistical analysis of theses experiments. We have a proposed solution to increase the sample time and that is to buffer the samples in the memory of the microcontroller. The Bluetooth link is the bottleneck of our system and if we buffer the data before we begin any transmission this should allow for further sample rate improvements, and still maintain a real-time feel to the user.

We also measured the power consumption for the helmet electronics: one sensor, microcontroller, and Bluetooth transmitter. Without using the timer and counter features of the microcontroller, but reading a single sensor, using the ADC, and transmitting we found a maximum current draw of 70 mA. This corresponds to a power consumption of approximately 350 mW, far below our 1.51 W maximum allowed. However, this is not the entire sensor network so the actual power consumption will increase slightly as we incorporate the other five accelerometers.

D. Data Analysis

The purpose of this block is to determine the probability of a concussion for a specific player based on the hits the player has taken. It is broken into two main parts, the database and the risk algorithm. The database stores the maximum magnitude linear acceleration measured for each impact. The risk algorithm queries the linear acceleration magnitude table for all the impacts, calculates the risk of concussion for each impact, and adds the risks together. Once the risk algorithm was written, it was tested using black box testing methods. The data processing and storage, acts as the backbone for the Android application, which displays the risk of a concussion for each hit and the cumulative risk.

The database is a MySQL, My Structured Query Language, database that is on a remote server. The database is controlled by PHP scripts, which can connect, read, and write to the database [14]. The risk algorithm gets the impact data from the helmet and finds the max magnitude linear acceleration. Then the risk algorithm uses a PHP script to store the max magnitude linear acceleration, the player name, player age, and the player number into a database. The magnitude database also keeps track of the time the entry is entered into the database. With all this information stored into the database, a risk algorithm [15] can use this information.

To continue to develop the risk algorithm, we needed a way to correlate linear acceleration to probability of concussion. We found a study [15] done by Steve Rowson, a professor of the School of Biomedical Engineering & Sciences at Virginia Tech. Rowson performed a study on concussive impacts by equipping 314 collegiate players' helmets with the HIT System, which consists of six accelerometers and calculates a resultant linear head acceleration at the CG, center of gravity, of the head. Also, 21 players' helmets were equipped with the 6DOF, Six Degrees of Freedom, system that had twelve accelerometers instead of six. The 6DOF is able to calculate linear and rotational acceleration. The study had monitored players between 2007 and 2009 and noted when players were diagnosed with a concussion [15].

The study derived an equation that converts linear acceleration to rotational acceleration from the equations of motion, modeling force acting on a head [See Equation (1)].

$$\alpha = \frac{m\sqrt{ax^2 + ay^2}}{I} d \qquad (1)$$

In Equation (1), α is rotational acceleration, m is the mass on the head, ax is the peak acceleration along the anterior-posterior axis of the head, ay is the peak acceleration along the medial-lateral axis of the head, I is the moment of inertia of the head, and d is the perpendicular distance from the head CG to the impact vector [15]. The unknown variables of m, d, and I were determined through a regression model analysis of recorded 6DOF acceleration data and confirmed with laboratory validation experiments. A least squares technique was used to equate (m*d/I) to 6.48 m^{-1} [15].

The study also developed an equation that correlated rotational acceleration and risk. To develop the risk equation, the study performed a statistical analysis on occurrences of sub-concussive impacts and concussive impacts reported. Recorded accelerations were correlated with either sub-concussive or concussive impacts. A logistic regression analysis based on weighted sub-concussive and concussive head acceleration distributions was used to express risk as a function of rotational head acceleration [See Equation (2)] [15].

$$\text{risk} = \frac{1}{1 + e^{-(c_1 + c_2\alpha)}} \qquad (2)$$

In Equation (2), α is rotational acceleration; c_1 and c_2 are regression coefficients. The regression coefficients were determined using a generalized linear model technique. The c_1 coefficient was calculated to be -12.531, and the c_2 coefficient was calculated to be 0.002 [15].

After reviewing his study and speaking with Steven Rowson over the phone, the risk algorithm was developed to model the equations derived in the study. When the user requests the cumulative risk, the risk algorithm first retrieves and displays the player number and name. When the user selects a player, the risk algorithm queries the database for all of the peak linear accelerations for that player. The algorithm then converts the peak linear acceleration that was returned from the query into a rotational acceleration [See Equation (1)]. For testing purposes, ay is equal to 0 and ax is the peak linear acceleration from a single accelerometer. The rotational acceleration is then used to calculate a risk [See Equation (2)]. The risk algorithm then sums up all the individual risks and returns the cumulative risk for the user interface code to display. This summation of risk is currently our way to incorporate the possible dangers of multiple sub concussive impacts; this may be revised further.

The entire risk algorithm was black box tested by developing a Java test program that would parse input text files containing set impact data, clear the database, and run the impact data through the risk algorithm. The Java test program then writes the rotational accelerations and risks produced by the risk algorithm to output text files. A python script was written to generate 200 input linear accelerations into the Java test program. Another python script was written to parse the 200 output text files and store the calculated rotational accelerations and calculated risks into a datasheet. The results of this test were graphed and matched the graph of the risk function from Rowson's study [See Equation (2)].

E. User Interface and Communication Block

This block's purpose is to display concussion data to the user and allow user input to control the application and system settings. The two main parts of this block are the Android application and the application's interface with the Android device's internal Bluetooth module. Currently the ability for the application to receive a data stream from the helmet-processing unit is working and the data is displayed. The application uses its main screen as the display for incoming concussion data and a set of buttons for connecting and disconnecting from the Bluetooth radio in the helmet. The next task is implementing a method for transmitting an updated threshold setting to the ATmega32U4 based on previous impact history. This will allow RCA to become more sensitive to multiple sub-concussive impacts for that player.

The application is written in the Android programming language, using the Android Development Tools plug-in for Eclipse IDE. For programming, we took advantage of prior coding experience and the development tools available to write the application. The application currently has three main activities. First is the main screen activity, where the user can connect and disconnect to the Bluetooth module housed in the helmet. The next activity is the graph view, which allows the user to view a plotted graph of the received acceleration data. This graph activity comes from an open source graphing library that is available for use in Android development. The final activity is the cumulative risk activity, which retrieves the impact data from the server and shows a comprehensive analysis of the player, their impact history, and their cumulative risk of concussion. This interface will continue to evolve and will be refined for CDR. As we were testing the application, we noted reliability issues with the application, as sometimes it would be forced to close as a result of some uncaught error. Our main focus for the continued development of this application is functionality and reliability.

The application's Bluetooth interface was created in reference to the tutorial on Android device and Arduino Bluetooth Communication [16]. This tutorial provided the necessary code examples to detect, connect, and disconnect an Android to an Arduino. We were able to adapt this code to connect to our ATmega microcontroller and interface it with its USART capabilities. As we tested the Bluetooth module for the distance as outlined earlier in the document, we also took that opportunity to test the robustness of the Bluetooth data transfer itself. We observed that at distances before losing connection we saw a significant drop in the rate in which data was being acquired. We are continuing to debug this situation and determine if it is a programming error or a limitation of the components themselves. We also tested the response time of each individual data transfer across the Bluetooth to characterize the system response time. The measured time between data reception was 16.8 ms, which describes the time it takes for the application to receive a data point via Bluetooth and perform the necessary computations to make that data meaningful and display it. This response is acceptable and meets our overall system specification of having a 500 ms response time.

III. PROJECT MANAGEMENT

Table II shows the list of the MDR goals for our project.

TABLE II
MDR GOALS

Specification	Value
Sensor/Microcontroller interfaced	100
Microcontroller Bluetooth interfaced	100
Process and transmit sensor data	100
Display sensor data on Android device	100
Configure database and server	100
Android device/Server interfaced	100

We have accomplished all of the most challenging goals, what remains are to find server space to host our server online, and to develop a consistent and known impact force for our testing apparatus. Both a single sensor and Bluetooth radio have been successfully integrated with the microcontroller. We have a working first iteration of the application on the Android device, which can receive the sensor data wirelessly and display results. The server and required databases are constructed and are currently being hosted on a local machine. We have a test apparatus constructed and are conducting trials now to better understand how to inflict our desired force.

The team members of RCA compliment each other very well, and there is a great group dynamic. Every member brings something unique to the table and all help each other out and try to become involved with every facet of the project. On more than one occasion a team member may have been at a low point in their workload and has helped offset another's. We have regular team meetings once per week, and one team meeting with our advisor, Professor Hollot per week. Scott Rosa is the CSE of the group and is the data processing and server expert, as well as maintains the website. Kenneth Van Tassell is the Android device programmer of the group, and working on the wireless communications on the Android device end. Justin Kober is responsible for the sensor network and power for the Impact Data Collection block. He is also in charge of the validation testing of the project. Tim Coyle is responsible for the microcontroller and wireless transmission of data to the Android device; he is also the team manager. Fig 3. below shows the Gantt chart for our project.

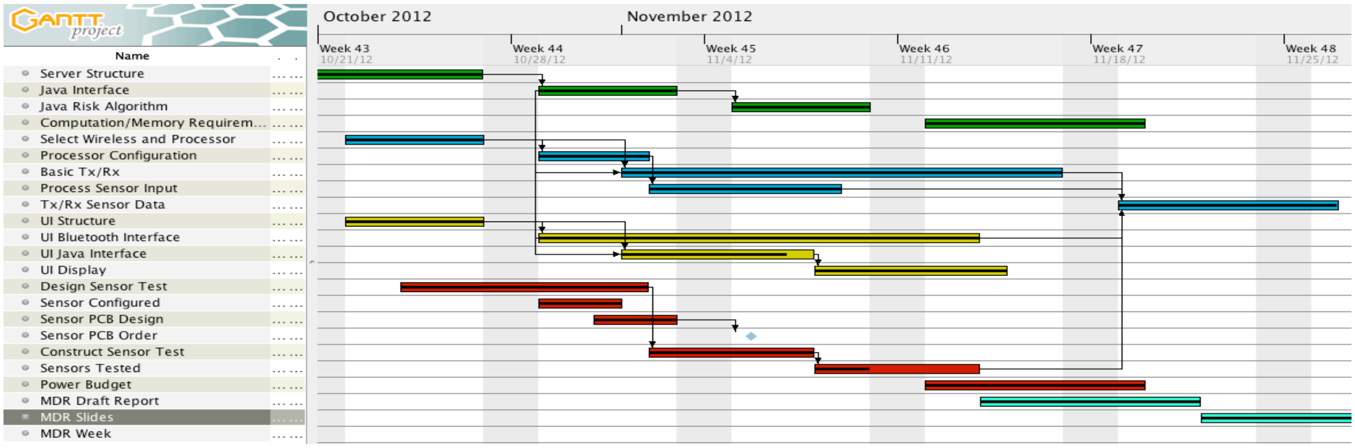


Fig. 3. This RCA Gantt chart shows our progress.

I. CONCLUSION

We have currently met our main MDR goals, but still have some important optimization to work on. Currently we can inflict a known force and RCA will accurately, +/- 4 g, transmit that force to the Android device where the user will see the correctly measured force. We were able to stay on track with our Gantt chart and weekly meetings, keeping the lines of communication open. Between now and CDR we will be optimizing the sample rate, working on the sensor network and revising our algorithms and application to process an array instead of a single sensor.

The next phase of RCA is to interface an entire sensor network of six accelerometers. The microcontroller will need to cycle through each sensor one at a time as it can only read one ADC channel at a time since they are all multiplexed. The code for this is ready for testing once we have more sensors configured. We need to decide the means of triggering our transmission; currently we plan on using the timer and interrupt once a force threshold has been reached. The user interface will need to be configured to have a way to input players and recall their information at a user's request. There will need to be two-way communication between the Android device and helmet so we can have adjustable thresholds to account for multiple sub-concussive impacts. Trying to establish this network and packaging it may prove to be one of the more challenging pieces of RCA. The sensors will need to maintain a fixed orientation and the circuitry in the helmet needs to be protected, but not feel any different to the player. We plan on reaching all of our goals for RCA by continuing to stay ahead of deadlines and keep communication lines open with the team and advisor.

APPENDIX

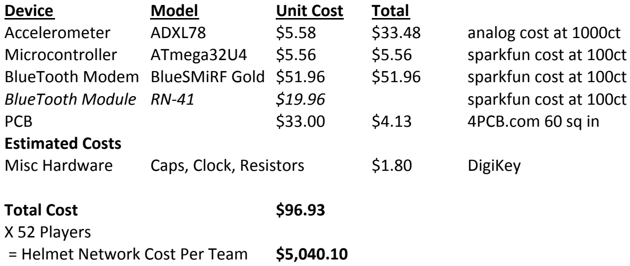
A. *Application of Engineering*

There are many areas of math, science, and engineering that apply to RCA, most notably: data structures and algorithms, classical mechanics, circuit analysis, electronics, computer networks, hardware organization, communications, and signal processing. For our software development portion of RCA the server, databases and Android development were all done in Java, while the software for the microcontroller and Bluetooth control are written in C. We have exposure to these programming languages through the coursework in ECE 122, ECE 242, ECE 353 and ECE 354. The sensor network and the power regulation circuitry design were both essential in our data collection block. These elements of RCA would not have been implemented correctly had we not had previous knowledge of circuit design, which we gained in courses like ECE 211, ECE 212, ECE 323 and ECE 324. Courses like ECE 313, ECE 314, ECE 333, ECE 374 and ECE 563 helped us to better understand the fundamentals needed for successful wireless communication.

B. *RCA Cost*

Below is our initial cost analysis for the project. All of the costs are in the helmet network, as this system assumes the user already has an android device.

TABLE III
HELMET NETWORK COST



Device	**Model**	**Unit Cost**	**Total**	
Accelerometer	ADXL78	$5.58	$33.48	analog cost at 1000ct
Microcontroller	ATmega32U4	$5.56	$5.56	sparkfun cost at 100ct
BlueTooth Modem	BlueSMiRF Gold	$51.96	$51.96	sparkfun cost at 100ct
BlueTooth Module	*RN-41*	*$19.96*		sparkfun cost at 100ct
PCB		$33.00	$4.13	4PCB.com 60 sq in
Estimated Costs				
Misc Hardware	Caps, Clock, Resistors		$1.80	DigiKey
Total Cost		**$96.93**		
X 52 Players				
= Helmet Network Cost Per Team		**$5,040.10**		

ACKNOWLEDGMENT

Team RCA would like to thank Holyoke Catholic High School and the University of Massachusetts Amherst Football team for supplying football helmets to our project. We would also like to thank Professor Steven Rowson for allowing us to use his risk function and speaking with us about our project and his research.

REFERENCES

[1] CDC Injury Prevention & Control: Traumatic Brain Injury. (2012, January 30). *Concussion and Mild TBI* [Online]. Available: http://www.cdc.gov/concussion/ [Accessed Web. 17 Nov. 2012.]

[2] Health Day News. (2012, January 30). *Many High School Football Players Ignore Signs of Concussion: Survey* [Online]. Available: http://consumer.healthday.com/Article.asp?AID=669798 [Accessed Web. 5 Dec. 2012.]

[3] McCrea, M., T. Hammeke, G. Olsen, P. Leo, and K. Guskiewicz. *Unreported concussion in high school football players: implications for prevention.* Clin. J. Sport Med. 14:13–17, 2004.

[4] B. Wilner. (2012, June 7). *NFL Concussions Mega-Lawsuit Claims League Hid Brain Injury Links From Players* [Online]. Available: http://www.huffingtonpost.com/2012/06/07/nfl-concussion-brain-trauma-lawsuit-players_n_1577497.html [Accessed Web. 17 Nov. 2012.]

[5] System and method for measuring the linear and rotational acceleration of a body part, by J.J. Crisco, III and R.M. Greenwald. (2001, Oct 10). *Patent US6826509B2* [Online]. Available: http://www.freepatentsonline.com/6826509.pdf [Accessed Web. 30 Sept. 2012.]

[6] Analog Devices, "ADXL78 Datasheet and Product Info.,*"ADXL78: Â Single-Axis, High-g, IMEMSÂ® Accelerometers.* [Online]. Available: http://www.analog.com/en/mems-sensors/mems-accelerometers/adxl78/products/product.html. [Accessed Web. 17 Nov. 2012.]

[7] ATMEL. (2010, Nov.). ATmega16U4/32U4 Preliminary Datasheet. [Online]. Available: http://www.atmel.com/Images/doc7766.pdf [Accessed Web. 6 Oct. 2012.]

[8] Roving Networks. (2012, Oct.). RN-41 Datasheet. [Online]. Available: http://www.rovingnetworks.com/resources/download/18/RN_41 [Accessed Web. 16 Oct. 2012.]

[9] IDC – Press Release. (2012, November 1). *Android device Marks Fourth Anniversary Since Launch with 75.0% Market Share in Third Quarter, According to IDC* [Online]. Available: http://www.idc.com/getdoc.jsp?containerId=prUS23771812 [Accessed Web. 17 Nov. 2012.]

[10] Analog Devices, "ADXL193 Datasheet and Product Info.,*"ADXL193: Â Single-Axis, High-g, IMEMSÂ® Accelerometers.* [Online]. Available: http://www.analog.com/en/mems-sensors/mems-accelerometers/adxl193/products/product.html. [Accessed Web. 17 Nov. 2012.]

[11] P. Hood-Daniel. (2012, October 10). Microcontroller Tutorial. [Online]. Available: http://newbiehack.com/MicrocontrollerTutorial.aspx [Accessed Web. 20 Oct. 2012.]

[12] HeKilledMyWire. (2011, January 5). Using the USART/serial. [Online]. Available: http://hekilledmywire.wordpress.com/2011/01/05/using-the-usartserial-tutorial-part-2/#more-31 [Accessed Web. 26 Oct. 2012.]

[13] Sparkfun. (2011, Nov.). Bluetooth Mode – BlueSMiRF Gold. [Online]. Available: https://www.sparkfun.com/products/10268 [Accessed Web. 16 Oct. 2012.]

[14] Tamada, R. (2011). *How to Connect Android device with PHP, MySQL* [online]. Available: http://www.Android devicehive.info/2012/05/how-to-connect-Android device-with-php-mysql/ [Accessed Web. 16 Nov. 2012.]

[15] Rowson, S, et al. “Rotational Head Kinematics in Football Impacts: An Injury Risk Function for Concussion,” *Annals of Biomedical Engineering,* vol. 40, no. 1, pp. 1-13, Jan. 2012. [Accessed Web. 21 Sept. 2012.]

[16] Bell, M. (2012). *Android device and Arduino Bluetooth Communication* [online]. Available: http://bellcode.wordpress.com/2012/01/02/Android device-and-arduino-bluetooth-communication/ [Accessed Web. 16 Nov. 2012.]