MaxProtect – Early Concussion Detection System

Osbaldo Vera

R11364327

Texas Tech University

ECE 3332-301

Mark Storrs, Ph.D., P.E.

Crystal Nelson, Christian Jensen

March 15, 2018

Abstract

This paper describes the hardware and software characteristics of *MaxProtect,* a system designed to wirelessly transmit data that can detect a concussion during athletic events. Utilizing a low power microcontroller's ADC and UART serial communication with an accelerometer and Bluetooth module, this system measures rotational and linear acceleration to detect a trauma head injury based on a Virginia Tech Study on concussions. This paper is written to discuss the system's five modules: Data Collection, Wireless Communication, Android User Interface, Power and Charging system. The purpose of this device is to minimize the occurrence of head trauma in athletes.

Table of Contents

List of Figures

List of Tables

1. Introduction

In recent years concussions have become a growing concern in sports, from the high school to professional level. A concussion is a traumatic brain injury, TBI, caused from a shock to the head that moves the head and brain back and forth. According to the CDC states that symptoms for concussion can take weeks to surface, this makes detecting a concussion in real time a problem during athletic events [1]. Another level of difficulty presents itself when most high school to college level athletes do not report common symptoms out of fear of not being able to play [2]. This puts athletes in a dangerous position as each repeated concussion increases the risk of severe brain injury by a factor of 4 each concussion.

The most advance system to date is the Head Impact Telemetry(HIT), it uses an array of accelerometers to detect multiple angles of shock and acceleration of the head thus creating a more reliable system for monitoring head movement at a high price [3]. The HIT system is also only for raw data collection and does not provide users with a user interface, rather one must be built before it could ever be used. MaxProtect is designed to be a low cost reliable alternative with an Android App user interface that allows one to easily see the risk of a concussion based off data gathered by a Virginia Tech Study [3]. See figure 1 for the complete system diagram.

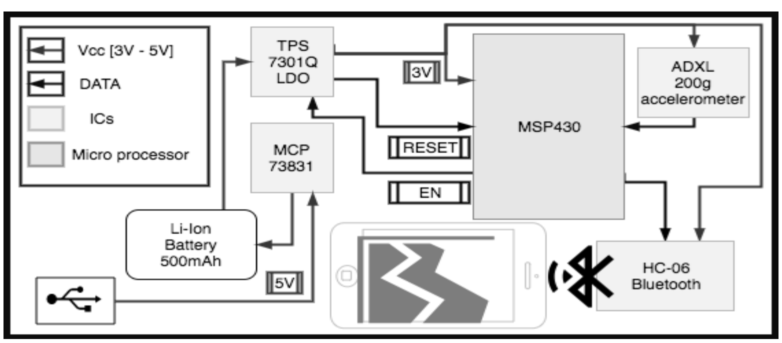

Figure 1: System Diagram of MaxProtect, By Christian Jensen

2. Data Collection

The purpose of this module is to collect the data used to determine whether a hard hit on a specific player caused a concussion. The main component used to measure the linear acceleration of the players head will be a 3-axis 200g accelerometer, the ADXL377, chosen for the accuracy in high g measurements [4]. The accelerometer is positioned at the back of the helmet with the positive Z direction facing towards the head, while the positive Y direction is down from the head, and the positive X direction is to the left of the head. The resulting voltages from the accelerometer will be polled by a built in 10bit ADC Sequence of Channels in the Msp430 microcontroller, then converted in linear acceleration, and only transmitted if it hits a threshold of 15g's (147m/s^2) to filter out any normal activities [5].

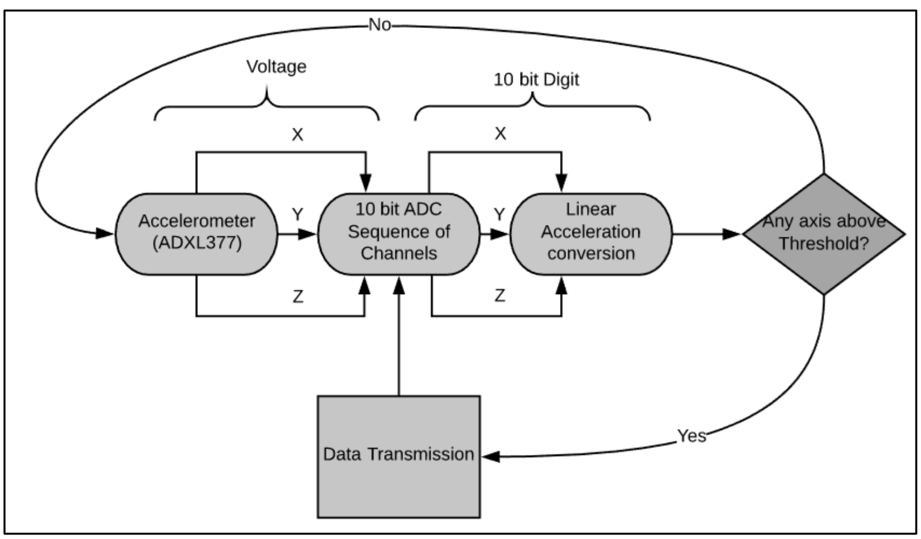

Figure 2: Software Flowchart for Data Collection, by Osbaldo Vera

Table I: ADXL377 Characteristics [4]

Parameter	Test Conditions/Comments	Min	Typ	Max	Unit
SENSOR INPUT	Each axis				
Measurement Range			±200		*g*
Nonlinearity	% of full scale up to 180 *g*		±0.5		%
Cross-Axis Sensitivity[1]			±1.4		%
SENSITIVITY, RATIOMETRIC[2]	Each axis				
Sensitivity at X_{OUT}, Y_{OUT}, and Z_{OUT}	$V_S = 3$ V	5.8	6.5	7.2	mV/*g*
Sensitivity Change Due to Temperature[3]	$V_S = 3$ V		±0.02		%/°C
ZERO *g* BIAS LEVEL, RATIOMETRIC					
Zero *g* Voltage	$V_S = 3$ V, $T_A = 25$°C	1.4	1.5	1.6	V
Zero *g* Offset vs. Temperature					
X-Axis and Y-Axis			±12		mg/°C
Z-Axis			±30		mg/°C

Using 3V to power the accelerometer the reference voltage of the ADC can be set to 3V as well. Using the given characteristics in Table 1, the linear acceleration is calculated with the fallowing equation on the microcontroller.

$$A = 4(ADC - 512) + \frac{(ADC-512)}{2} \tag{1}$$

Where ADC represents the 10bit ADC value, and A being the linear acceleration of the specified axis. The derivation of equation 1 is in appendix A. For the microcontroller to calculate the accurate acceleration in a reasonable amount of time, this exact equation must be used to minimize the execution time. Using a Timer A on the Msp430 the execution time was measured to be 53µs to convert all 3 axis from ADC conversion to linear acceleration conversion. The sampling rate is then 18kHz allowing the system to more precisely detect hits above the threshold, to start recording data into the phone.

Initial testing was performed with a 3-axis 3g accelerometer, ADXL337, to verify correct conversion between output voltages into Linear Acceleration as shown in Figure 3.

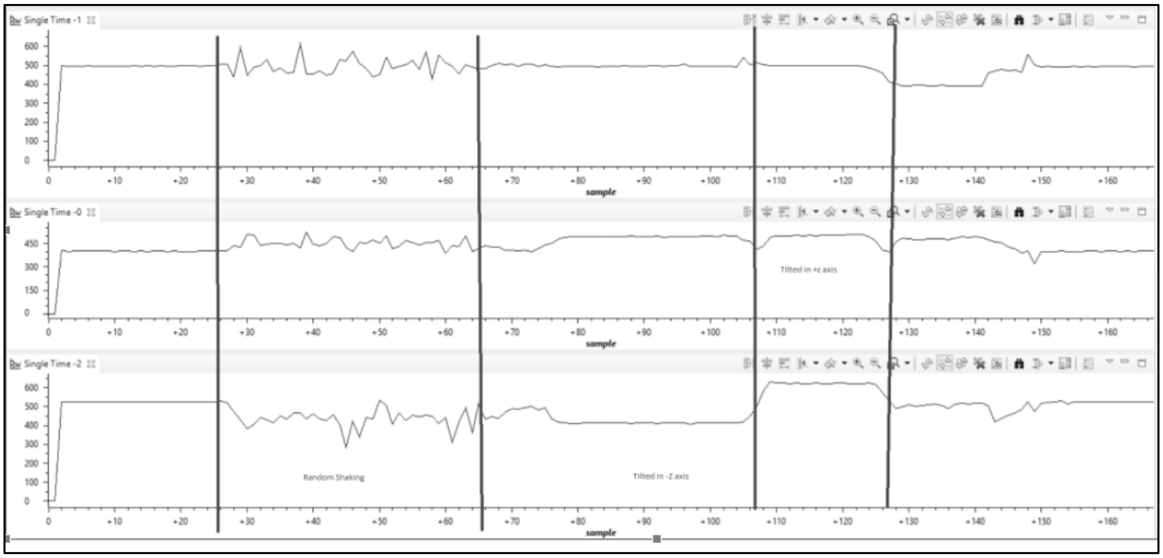

Figure 3: 3g Accelerometer Testing, by Osbaldo Vera

The figure above shows test results of testing the linear acceleration calculation by flipping the accelerometers on different axis, where 9.8 m/s^2 is the direction of gravitational acceleration. Linear Acceleration was successfully calculated and can now be used to collect the linear acceleration used to calculate rotational acceleration in the Android App. Without having to test higher moments of Linear Acceleration, since lower moments are less accurate on the 200g accelerometer, which is design specifically for high impact detection instead of tilt detection like the 3g accelerometer.

The data will only transmit through the Bluetooth module if it hits a certain threshold. This threshold is the max acceleration of sprinting, jumping and anything else that is not a high impact. The threshold is calculated by gathering data on these common actions and setting a high enough limit. In the Virginia Tech Study, this threshold in athletes was calculated to be 15g's (147m/s^2), this threshold is high enough that it will not be triggered by non-impact data and low enough to not miss any high-impact hits [3].

3. Wireless Communication

Once the Data has been collected, the data must be sent through a Bluetooth module, so it can wirelessly communicate the data while the athletes are in game. Using an 8-bit, no-parity, 1-stop bit UART-configurated Bluetooth module like the HC-06, it is possible to transmit data up to 100m, plenty of space between a coaching staff and a specific player. The Msp430 has a built in UART Communication system that allows one to use a vast variety of Bluetooth chips. See Table II for the modulation settings for varying baud rates and clock speeds for different types of Bluetooth modules.

Table II: Modulation settings with corresponding Frequency and Baud Rate [5]

BRCLK Frequency [Hz]	Baud Rate [Baud]	UCBRx	UCBRSx	UCBRFx	Maximum TX Error [%]		Maximum RX Error [%]	
12,000,000	115200	104	1	0	-0.5	0.6	-0.9	1.2
12,000,000	128000	93	6	0	-0.8	0	-1.5	0.4
12,000,000	256000	46	7	0	-1.9	0	-2.0	2.0
16,000,000	9600	1666	6	0	-0.05	0.05	-0.05	0.1
16,000,000	19200	833	2	0	-0.1	0.05	-0.2	0.1
16,000,000	38400	416	6	0	-0.2	0.2	-0.2	0.4
16,000,000	56000	285	6	0	-0.3	0.1	-0.5	0.2
16,000,000	115200	138	7	0	-0.7	0	-0.8	0.6
16,000,000	128000	125	0	0	0	0	-0.8	0
16,000,000	256000	62	4	0	-0.8	0	-1.2	1.2

Table II also shows the maximum RX and TX error rates for each specific modulation setting. For the HC-06 Bluetooth module using a 16MHz clocking frequency and a baud rate of 9600 the maximum error rates for the transmission line is negative .05 to positive .05. Ensuring our data is accurately transferred onto the user interface.

To transfer the data successfully, two packets of information must be sent because 8-bits can only transmit values up to 255. Figure 3 shows the software implementation for successfully sending 16-bits of data, plenty for what the system needs.

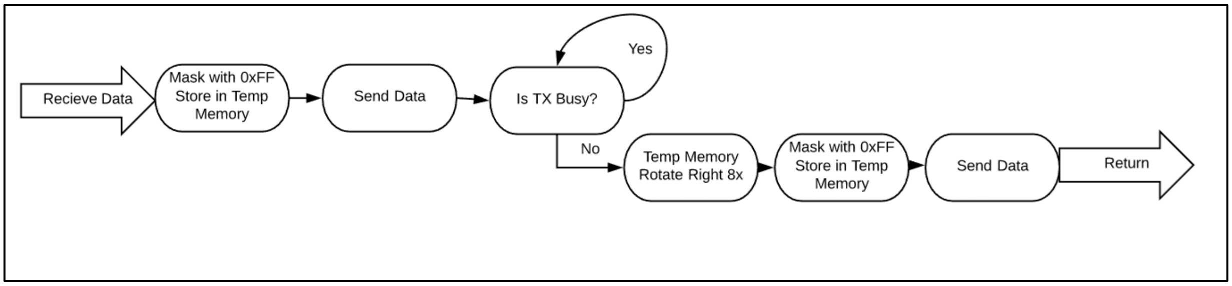

Figure 4: Software Flowchart for sending 16bits of data, by Osbaldo Vera

Masking the first byte of the data being transferred allows the packet to be sent through UART. Performing eight arithmetic rotate right on the data, the second half can be sent through UART. These two different bytes will need to be combined in the user interface to become readable. This function must be performed for all three axes of the accelerometer once the specified threshold has been detected by the microcontroller. Using the built in Timer A on the microcontroller the execution time for all three axes was measured to be 1.125ms. With this execution time and the data conversion and comparison execution time the update time was calculated to be 1.178ms, giving about 849 updates per second to the android application. This allows for more precise results, presenting accurate data to the user.

4. Android User Interface

The user interface utilizes an easy and familiar environment of a java application on Android devices. Chosen for its mobility and support of open source programming. The Application is designed to alert the user of the risk of a concussion, it also stores and analyze the data received through Bluetooth from the Msp430. Java also includes the option to use SQLite database, which allows for multiple parties to view the data. Since this product is design to let a variety of people utilize it, SQLite has been chosen as the storage database. Figure 5 shows some screenshots of the working app's home screen, alert screen and graph screen.

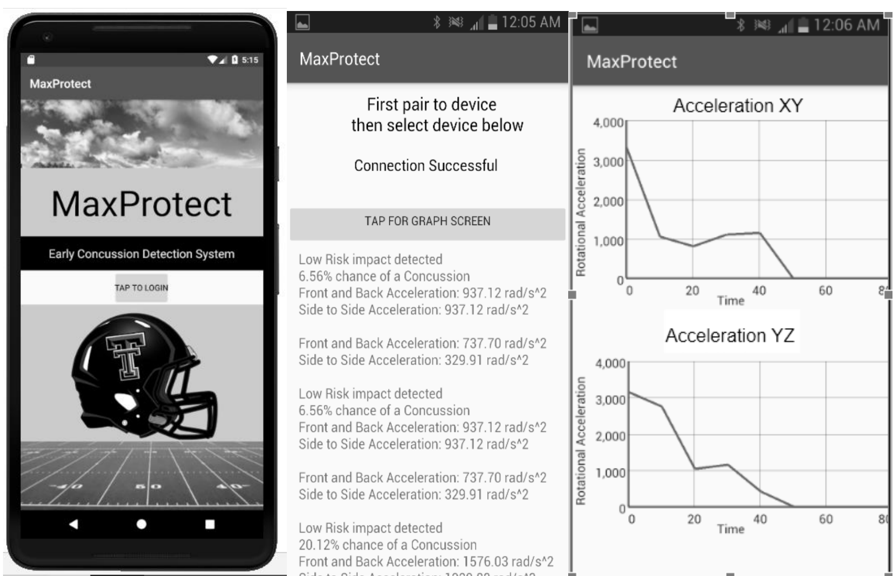

Figure 5: Working App Screenshots by Crystal Nelson

Setting up a separate thread to handle the Bluetooth data being received only updating the its database when information is being transferred. This avoids the application other functions to work as intended instead of waiting for the Bluetooth to receive data constantly.

```
public void Accelerations() {
    int num = 10;
    // arrays for storing linear accelerations in x, y, and z directions
    int a_x[] = new int[(num)];
    int a_y[] = new int[num];
    int a_z[] = new int[num];

    Arrays.fill(alpha_xy,0);
    Arrays.fill(alpha_zy,0);


    int a = 0;
    for (int i = 0; i <= 64;i++) {



        if((rawBytes[i] == 0x42)) {
            a_x[a] = rawBytes[i+1] + (rawBytes[i+2]<<8) ;
            a_y[a] = rawBytes[i+3] + (rawBytes[i+4]<<8) ;
            a_z[a] = rawBytes[i+5] + (rawBytes[i+6]<<8) ;

            double n = a_x[a] * a_x[a] + a_y[a] * a_y[a];
            double m = a_y[a] * a_y[a] + a_z[a] * a_z[a];
            alpha_xy[a] = 6.48 * Math.sqrt(n);
            alpha_zy[a] = 6.48 * Math.sqrt(m);
            a++;
        }
```

Figure 6: Parsing and Converting Rotational Acceleration by Crystal Nelson

The Parsing algorithm, see Figure 6, is used to decipher the data sent to the application, where the first byte received is stored in a variable, and the second byte is rotated left 8 times and then added onto the variable where the first byte was stored. This algorithm is performed repeatedly always going in the same order the data is sent, X axis, Y axis, Z axis, until the data stops being sent.

The deciphered data can now be used to calculate the front to back and side to side rotational acceleration of the players head, so we can use the graphs provided by the Virginia Tech study shown in Figure 7 [3].

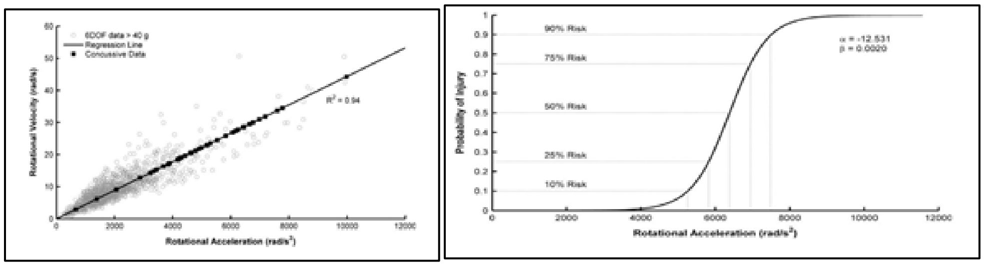

Figure 7. Virginia Tech concussive data analysis [3]

The Data analyzed by the paper shows the risk of a concussive injury depending on the rotational acceleration on the head of the player, specifying 10% to 90% risk model. The paper also provides the calculations for the rotational acceleration given in equation 2 and the risk calculation given in equation 3 [3].

$$\alpha = 6.48\sqrt{ax^2 + by^2} \tag{2}$$

$$risk = \frac{1}{1 + e^{-(c_1 + c_2\alpha)}} \tag{3}$$

In equation 2, ax and by are the linear rotational acceleration acted on the head of the player while the constant 6.48 is the calculated average of the average college football athlete’s Mass times distance from the head and the heads pivot point all over the moment of inertia typically experienced by the players head. In equation 3 c_1 and c_2 are constants calculated to be -12.531 and .002. Alpha representing the rotational acceleration calculation in equation 2. With these calculations performed, the data can display the risk of a concussive injury of a certain hit an athlete takes. Figure 8 Shows the full software flowchart. The output was thoroughly tested by attaching it to a wheel and spinning it gathering an encoder's rotation in a set amount of time.

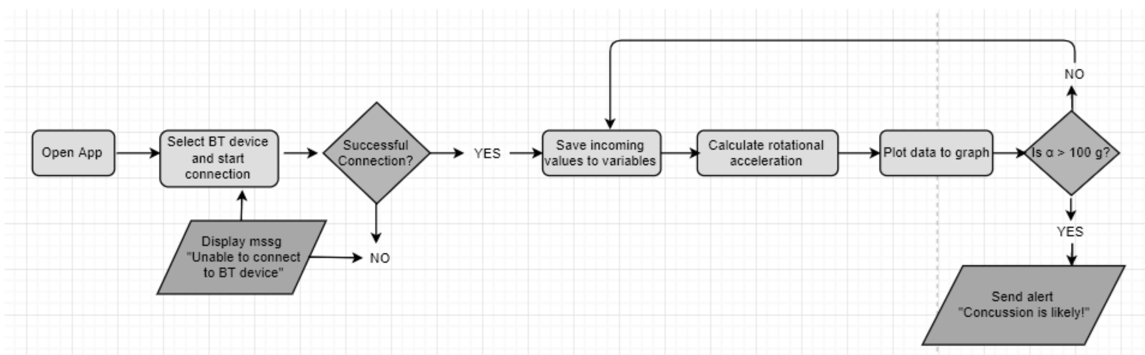

Figure 8: Software Flowchart of Android User Interface, by Crystal Nelson

5. Power and Charging system

Powering the Data collection and transmission is a 3.75V 500mAh Lithium-ion rechargeable battery chosen because of its size and its ability to be charged. Allowing users to charge their device before an event ensuring that the individuals components stay powered throughout an event. The current drawn by all individual components: ADXL377 (.3mA), HC-06 (8mA), MSP430G2553 at 16MHz(5mA), power system IC's (1mA). The total current draw is calculated to be around 14.3mA, meeting the goal of powering it for more than 5 hours, keeping it running for 34hours nonstop before being unable to power the individual components [4,5,7,8].

A Charge management controller IC, MCP73831, was chosen to handle the charging of the Lithium-ion battery, Figure 9 shows the circuit diagram from the voltage source to the battery.

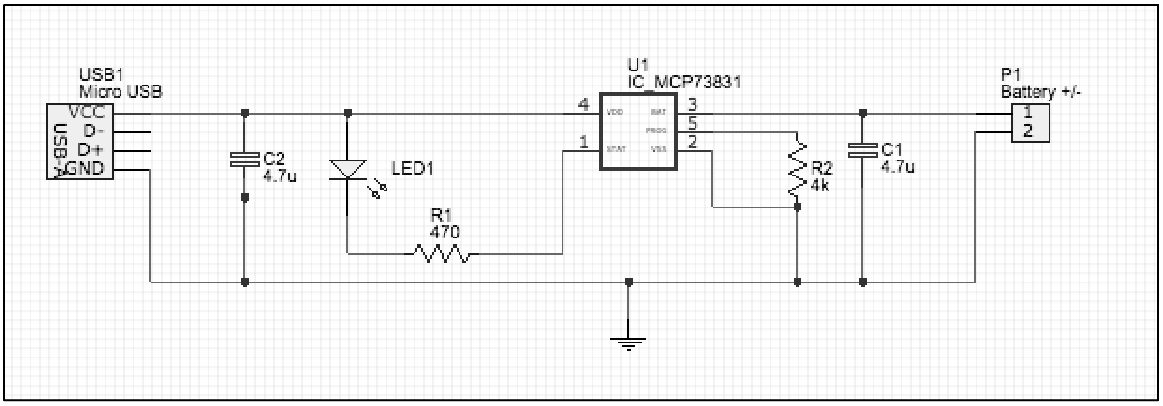

Figure 9: Charging circuit from USB source to Battery, Christian Jensen

This controller fallows the procedure for charging a lithium-ion battery, is outlined in Figure 10, where the charge current is set to 250mA, and a fixed voltage regulation of 4.2V represented.

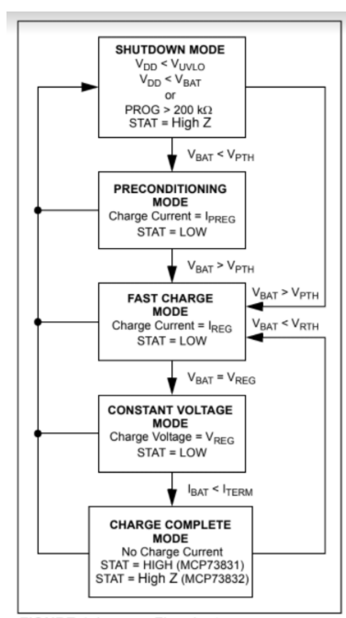

Figure 10: Charging Procedure of charging controller [7]

When not being charged the system is powered through a Low-Dropout Voltage Regulators with and integrated delayed reset Function, TPS7301QP. This allows the system to be powered by a constant voltage of 3V disabling the output voltage once the input voltage drops below 3V. Figure 11 shows the Circuit diagram with this specific IC.

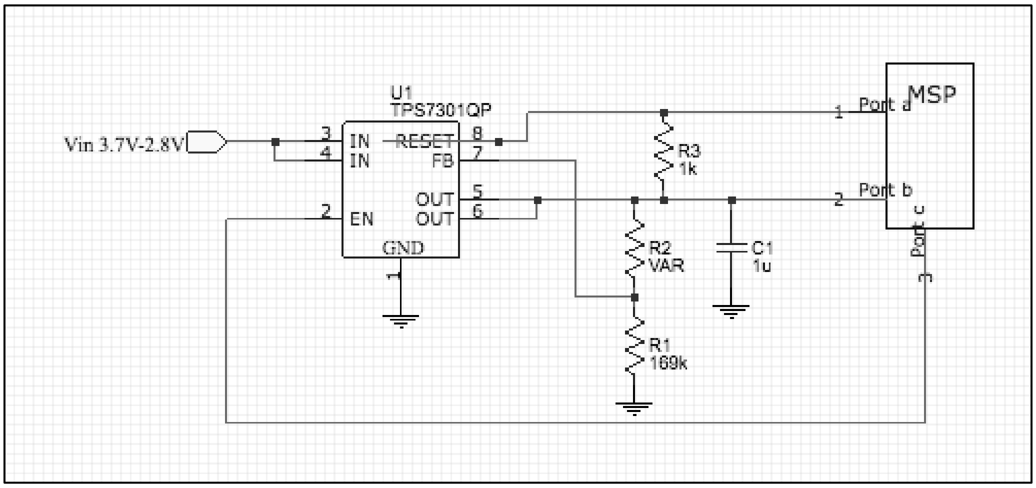

Figure 11: Circuit diagram of voltage regulation [8]

6. Conclusion

In all, MaxProtect, fills a well demanded product for inexpensive concussion detectors that accurately detects the risk of a concussive injury based on a long-term Virginia Tech Study. The product is tailored for use for both in a professional and high school environment as well as personal monitoring for multitudes of sports. The product is integrated with a simple to use User Interface, that any android device with a Bluetooth module can use. Also gives ample amount of battery life and the ability to be charged through a USB connection.

References

1. Center of Disease Control and Prevention, National Center for Injury Prevention and Control. “Heads Up,” January 2016, https://www.cdc.gov/headsup (February,2018).

2. Dotinga, Randy, “Many High School Football Players Ignore Signs of Concussion: Survey,” *HealthyDay*, October 22, 2012, https://consumer.healthday.com/fitness-information-14/football-health-news-250/many-high-school-football-players-ignore-signs-of-concussion-survey-669798.html (February, 2018).

3. Rowson, Steven, “Rotational Head Kinematics in Football Impacts: An Injury Risk Function for Concussion,” Virginia Tech University, 2012.

4. “ADXL377 Analog Devices Datasheet,” http://www.analog.com/media/en/technical-documentation/data-sheets/ADXL377.pdf (Febuary, 2018).

5. “Msp4302x53 Datasheet,” http://www.ti.com/lit/ds/symlink/msp430g2553.pdf (January 2018).

6. Wingood, Harry. “Android Send/Receive Data with Arduino using Bluetooth – Part 2,” 15 April, 2014, https://wingoodharry.wordpress.com/2014/04/15/android-sendreceive-data-with-arduino-using-bluetooth-part-2/ (18 March, 2018).

7. “MCP73832 Microchip Datasheet,” https://www.sparkfun.com/datasheets/Prototyping/Batteries/MCP73831T.pdf (21 March, 2018).

8. “TPS7301 Texas Instruments Datasheet,” http://pdf1.alldatasheet.com/datasheet-pdf/view/29268/TI/TPS7301.html (21 March, 2018).

Appendix A

A full Explanation of the derivation of equation 1.

For a 10-bit ADC the step size can be calculated with the fallowing equation:

$$Step(v) = \frac{V_{ref}}{1023} \tag{4}$$

Using a reference voltage of 3V the step size comes out to 2.9mv per bit.

Taking the information from Table() the sensitivity is shown to be 6.5mv per g-force.

Dividing them together will leave us with 2.24bit per g-force as shown below

$$\frac{6.5\frac{mv}{g}}{2.9\frac{mv}{bit}} = 2.24\frac{bit}{g} \tag{5}$$

Using the constant acceleration of g-force, $9.8\frac{m}{s^2}$

One can calculate the acceleration per bit by dividing 1 g force by 2.24 bit per g force the result is $4.375\frac{m}{s^2}$. Unfortunately, the Msp430 does not include floating point multiplication or division, a much efficient approach is to accept some error and round to 4.5 and rotate right once, luckily the GCC compiler interprets the division of 2 as 1 Rotate Right this allows the update frequency to be set to 849 Hz instead of 661Hz.

The accelerometer will be 1.5v at 0g according to Table(), at 1.5v the ADC value is 512, using this as the pivot point we can use this to get the equation.

$$A = 4(ADC - 512) + \frac{(ADC-512)}{2} \tag{6}$$

Appendix B

Gant Chart

		Engineers	01/30	02/06	02/13	02/20	02/27	03/06	03/13	03/20
1	**Research**									
1.1	Concussion Thresholds	All		100%						
1.2	Research	All		100%						
1.3	Wireless Comm (Bluetooth?)	All		100%						
1.4	Android App Development	All			100%					
2	**Programming**									
2.1	Msp430- Concusion Detector	Ozzie					100%			
2.2	Msp430-Wireless Transmission	Ozzie								
2.3	Java Tutorial	Crystal		100%						
2.4	Android Development Tutorial	Crystal				100%				
2.4	Andriod App- G.U.I	Crystal								
2.5	Android App - BT communication	Crystal/Ozzie								
2.6	Android App - Data Processing	Crystal								
3	**Circuits**									
3.1	Power Supply	Christian					95%			
3.1.1	- Charge Controller	Christian					90%			
3.1.2	- Voltage Regulator	Christian					90%			
3.2	Accelerometer- Analog Input	Christian								15%
3.2	Main Board	Christian								
3.3	Bluetooth Module	Christian								
4	**Misc**									
4.1	Design 3d Printable Packaging	All								0%
4.2	3D Print Packaging	All								
4.3	Troubleshooting	All								

Appendix C

Budget

LAB GROUP 5	Running Total			Total Estimate				Start Date	1/18/2018
Direct Labor:								Today	2/13/2018
Category or individual:	*Rate/Hr*	*Hrs*		*Rate/Hr*	*Hrs*			End Date	5/5/2018
Christian	20	59	$1,180.00	20	160	$3,200.00	Notes:		
Oz	20	90	$1,800.00	20	160	$3,200.00			
Crystal	20	87	$1,740.00	20	160	$3,200.00			
Total Labor:			$4,720.00			$9,600.00			
Consulting Fees:									
Category or individual:	*Rate/Hr*	*Hrs*		*Rate/Hr*	*Hrs*				
Lab I	15	0	$0.00	15	0	$0.00			
Lab II	20	0	$0.00	20	2	$40.00			
Lab III	25	0	$0.00	25	1	$25.00			
Lab IV & V	35	0	$0.00	35	0	$0.00			
Lab Tutors	40	0	$0.00	40	2	$80.00			
Lab Assistants	40	0	$0.00	40	0	$0.00			
Mr. Woodcock	100	0	$0.00	100	0	$0.00			
Instructor	200	0	$0.00	200	5	$1,000.00			
Total Contract Labor:			$0.00			$1,145.00			
Direct Labor Cost:		Subtotal:	$4,720.00		Subtotal:	$9,600.00			
Indirect Labor Cost	*rate:*	100%	$4,720.00	*rate:*	100%	$9,600.00			
Total Direct Labor:			$9,440.00			$19,200.00			
Supplies And Materials:			$177.09			$225.80			
(from Materials Cost worksheet)									
Total Direct Material Cost:			$177.09			$226			
Equipment Rental:	Value	Rental Rate (Per Week)		Value	Rental Rate		Date begin	Today/End Date	Total days
Oscilloscope	$2,851.00	1.00%	$4.07	$2,851.00	1.00%	$435.80	2/5/18	2/6/18	1
Function Generator	$292.50	1.00%	$0.42	$292.50	1.00%	$44.71	2/5/18	2/6/18	1
DMM	$259.00	1.00%	$0.37	$259.00	1.00%	$39.59	2/5/18	2/6/18	1
Power Supply	$699.00	1.00%	$1.00	$699.00	1.00%	$106.85	2/5/18	2/6/18	1
Total Rental Costs:			$5.86			$626.94			
Total Supplies Materials & Equipment:			$182.95			$852.74			
Total TDL+TCL+TDM+TRM			$9,622.95			$20,052.74			
Business Overhead:		55%	$5,292.62		55%	$11,029.01			
								Amount Under:	$16,166.18
Total Cost:		**Current**	$14,915.57		**Estimate**	$31,081.75		***Amount Over:***	-$16,166.18

Appendix B

WRITTEN LAB PAPER EVALUATION FORM

Student Name: Course Number:

Please score the student by circling one of the responses following each of the statements.

1) The student's writing style (clarity, directness, grammar, spelling, style, format, etc)

A B C D F Zero

2) The quality and level of technical content of the student's paper

A B C D F Zero

3) The quality of results and conclusions

A B C D F Zero

4) Quality of measurements planned/ taken

A B C D F Zero

Grade: