Self-Contained Analytical Skating Form Tracker

Design Review

Team 38 — Charles Hu, Jonathan Wang, Qian Ma

ECE 445 — Spring 2017

TA — Yuchen He

Contents

1 Introduction

1.1 Objectives

As an essential part of figure skating and hockey, ice skating is not only the foundation of the professional plays, but also a widely popular recreational sport for the public all over the world. The increased speed and reduced friction on ice often brings about rising level of safety issues. Therefore, ice skating calls for extreme precision and intricate body position. However, most skaters have not received formal training on their skating forms. Even for people with professional instructors, the traditional ways of coaching they get relies heavily on visual inspection and depends on the instructor's experience. Thus, many skaters, including some professionals, are unaware of their bad habits in skating form, which might lead to more risks of injury and will overall waste energy.

Our goal is to design an affordable, easy-to-wear device that can help users increase efficiency in skating while reducing the chances of injury. This device can track the skating forms and provide feedback to the user. It will assist the process of evaluating the skater performance and identify some subtle, unwanted habits that are hard to notice through visual inspection. We aim to make this design small and straightforward to use, while being affordable so the general public can utilize its benefits and improve upon their skating forms.

1.2 Background

In the past decade, researchers have worked on the analysis of on-ice biomechanics measurements. A team from University of Calgary, Canada implemented a system to investigate the forward skating technique [1]. Their measurement system includes EMG electrodes, electrogoniometers (to measure knee and hip angles), accelerometers, instrumented insoles, and data acquisition unit. Another project carried out at McGill University, Canada focused on analyzing dynamic forces during skater landing using specially modified skates and a force transducer system [3]. The main issue with the previous projects on skating data collection is the high cost to replicate these devices. They require either a multitude of expensive sensors or customized skates with special design [2]. In addition, the convoluted connection between devices make the system cumbersome and inconvenient for frequent use. All these factors make the existing systems prohibitive to the public.

Based on the above analysis, we aim to make our skate form tracker easy to use and highly affordable. It can be easily attached to the front of the skates near the laces. Since it's lightweight and small, it will be as non-invasive as possible to the user. It will also eliminate the complex wire connection so it's straightforward to use for the public. Ideally, the price of the device should be below $20.

In terms of technique, ice skating involves a series of complex motions. To begin with, ice skating is very unintuitive, as all people naturally learn to walk and run on frictional ground. The complexity of skating can be broken down into three general steps. The first motion is opening the hip. When striding with a leg, opening the hip allows for a skater to extend their leg farther, which increases the efficiency of each stride. The second motion is straightening the knee. This maximizes the stride length and thus increasing its efficiency. Last is the ankle push-off. This motion is the final general motion, where force is transferred from the middle or heel region of the foot towards the toe. The transfer of force adds extra propulsion to increase speed. These three general motions encompass much of the skating form. On top of that the symmetry of these motions in regards to the right and left leg is very important for proper skating technique. Asymmetry will cause imbalances and make skating more difficult. Thus, our device will collect data on the movement of the skater and then analyze it to determine if there are any asymmetries involved in the user's skating form.

1.3 High-Level Requirements

- The device must be reusable, water resistant, and must be less than 1 pound.
- The device must be able to capture the translational movement and rotational orientation of the user's skate. The movement of the skate must be accurate within 1 cm and the orientation of the skate must be accurate within ± 5 degrees.
- The device must provide graphical and numerical feedback on the user's skating form, based on the collected data.

Physical Design

As shown in Fig.1, the housing of the device is a rectangular box with dimension of 70mm*50mm*27mm. There are four holes on the top side of the housing. Slot 1 and 2 are for LED display, slot 3 and 4 are for push-buttons. There are two handles on the bottle of the housing for a strap to go through so the device can be securely fastened to the skates.

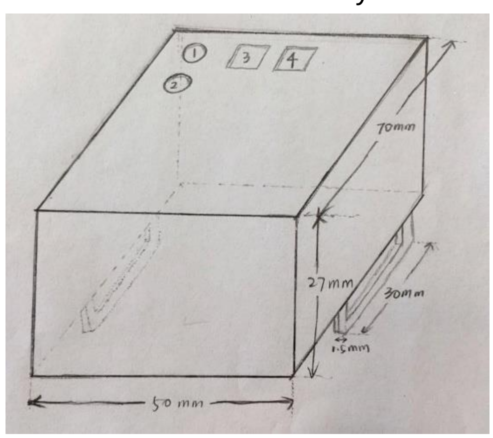


Figure 1: Device Housing Physical Design

2 Design

2.1.1 High Level Description

Our project features 2 almost-identical units, one for each foot. The device will be placed on top of the laces of the boot and secured by a strap that wraps around the outsole through the hole of the blade. For the device on each foot, one is designated as the master unit, and the other the slave. The only hardware difference is the lack of a physical button that is used for pair/sync/start/stop (depending on the current device state) on the slave device. Upon startup, the units should be in a rest state until the master unit initiates the pair up/syncing procedure (which in turn is initiated by the user pressing the button). The success of this sequence will be indicated by the status LED. Once paired and synced, the device will begin data collection, during which the microcontroller will sample the IMU and the force sensor in specified time intervals, and store it on the on-board microSD card. After a period of inactivity, or when the stop button is pressed on the master device, the data collection ends, and the slave unit will transmit all its collected data to the master unit. The master unit will combine the data collected from both devices and store them into the microSD card for post processing on the computer.

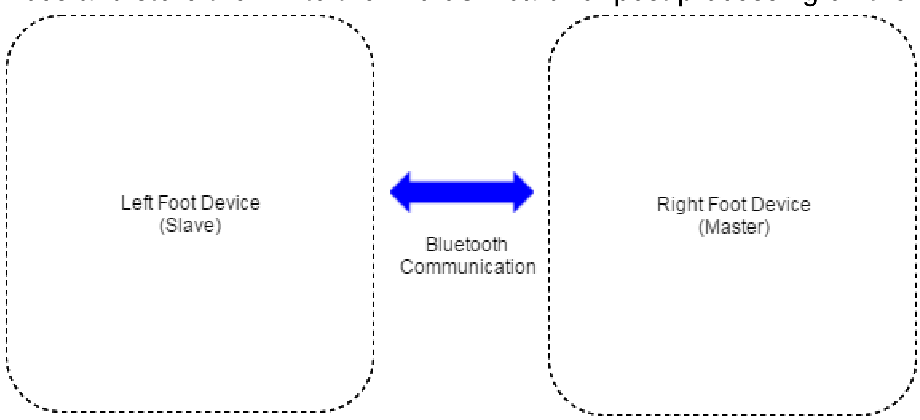


Figure 2: High-level Overview

2.1.2 Device Software Finite State Machine

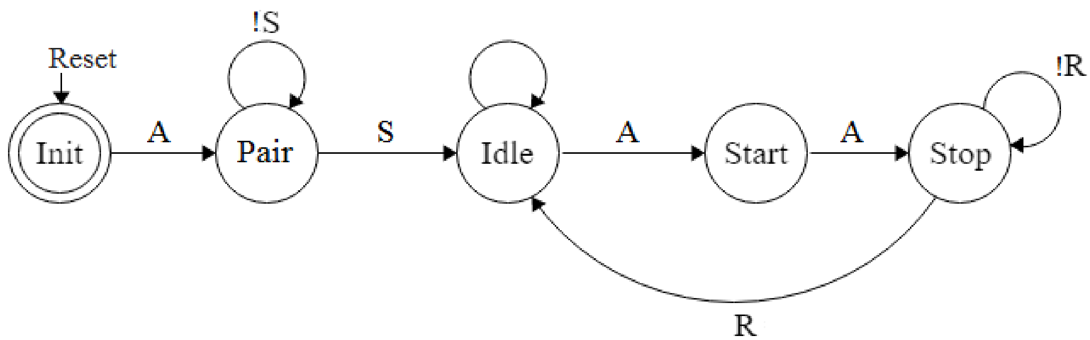


Figure 3: Device Software Finite State Machine (Master Side)

Input	Explanation
Reset	Triggered by pressing the Reset button
A	Triggered by pressing the Sync button
S	Bluetooth pairing process success
R	Bluetooth data transmission success

Table 1: Master FSM Input Explanation

State	Description
Init	Initial setup state when reset is pressed
Pair	Pairing state, master sends out pairing signal P to slave
Idle	Wait state in between Bluetooth pairing and data collection process, or after slave data transmission at Stop state is finished; Status LED2 on
Start	Data collection in progress; Status LED2 on
Stop	Initiate slave data transmission; Status LED2 off

Table 2: Mater FSM State Description

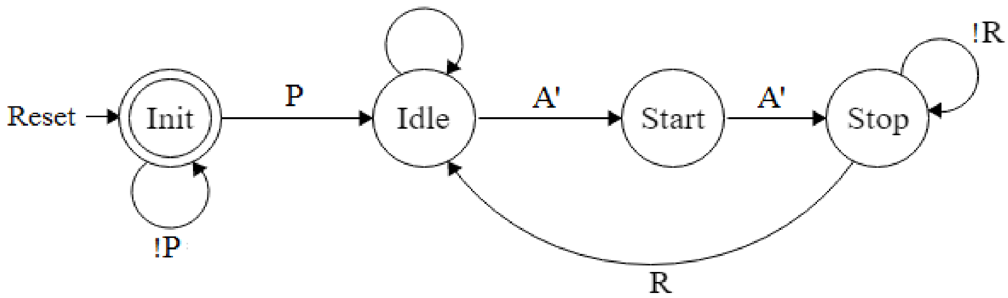


Figure 4: Device Software Finite State Machine (Slave Side)

Input	Explanation
Reset	Triggered by pressing the Reset button
P	Pairing signal from master module
A’	Signal A transmitted through Bluetooth
R	Bluetooth data transmission success

Table 3: Slave FSM Input Explanation

State	Description
Init	Initial setup and pairing state
Idle	Wait state in between Bluetooth pairing and data collection process, or after slave data transmission at Stop state is finished; Status LED2 on
Start	Data collection in progress; Status LED2 on
Stop	Data collection finished; Initiate slave data transmission; Status LED2 off

Table 4: Slave FSM State Description

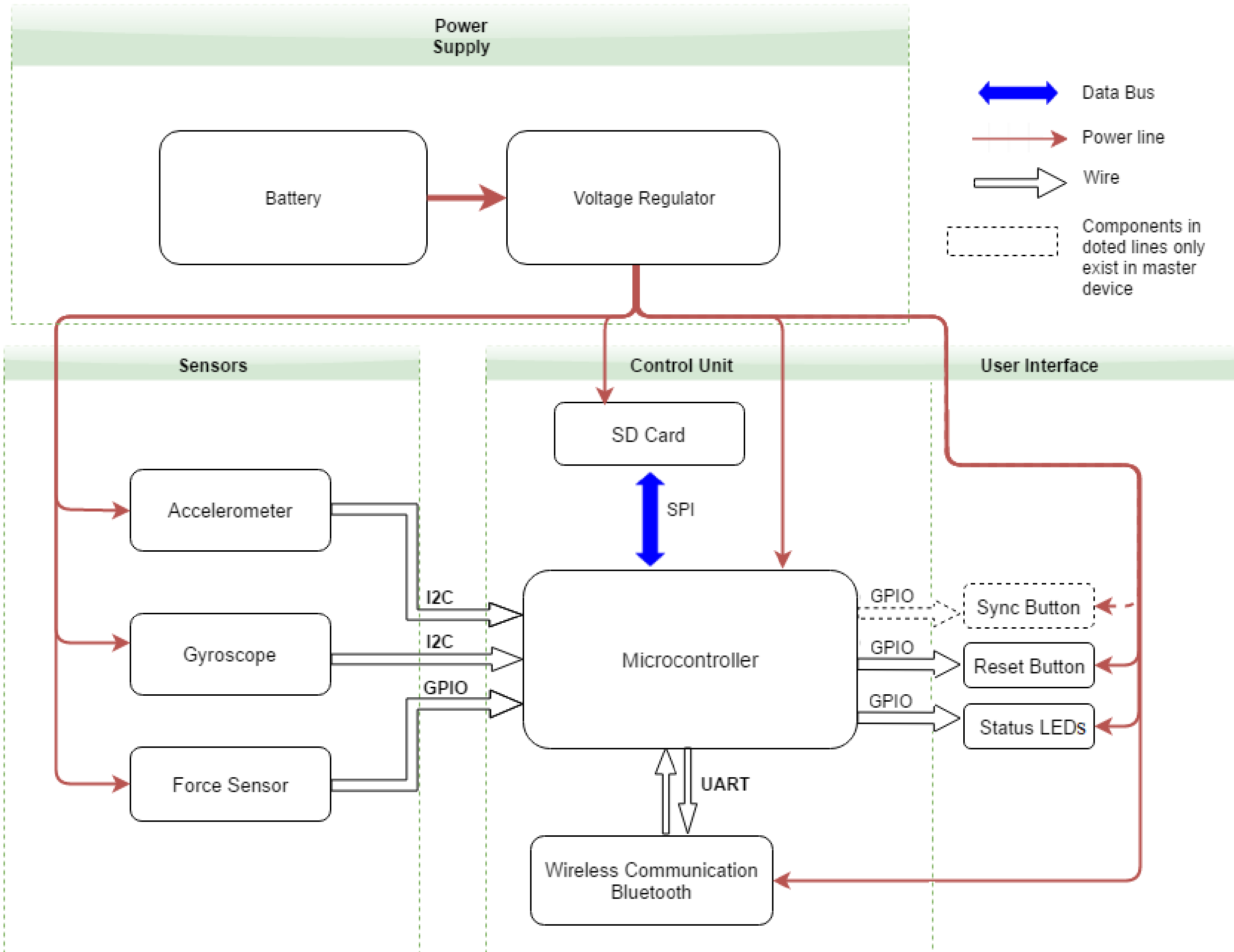


Figure 5: Hardware System Block Diagram

2.2 Block Descriptions

2.2.1 Power Supply

Input: 9V Alkaline Battery

Output: 9V directly to Microcontroller analog pin 25, 3.3V ± 5% to Microcontroller, force sensitive resistors, Bluetooth Transceiver, and IMU unit

The voltage regulator regulates the 9V battery input voltage to the 3.3V operating voltage for the individual modules. As shown in figure 8, capacitor C1 is connected between the input pin and ground to improve the power supply rejection ratio (the ability to maintain output voltage as input voltage is varied). Capacitor C5 is connected between output and ground for noise reduction.

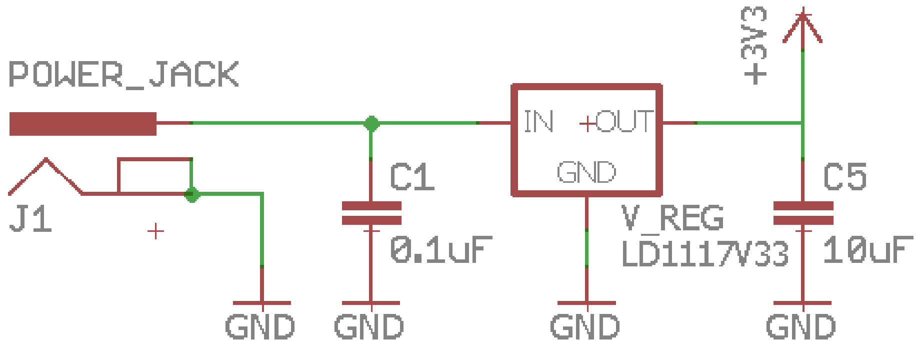


Figure 6: Power Supply Circuit

Since we expect our device to operate solely on the power supply unit for an extended amount of time, the durability of the battery is essential and requires careful testing. The circuit shown in Fig.9 could be used to test the power supply. For detailed steps refer section 2.3 and calculation 2.5.5.

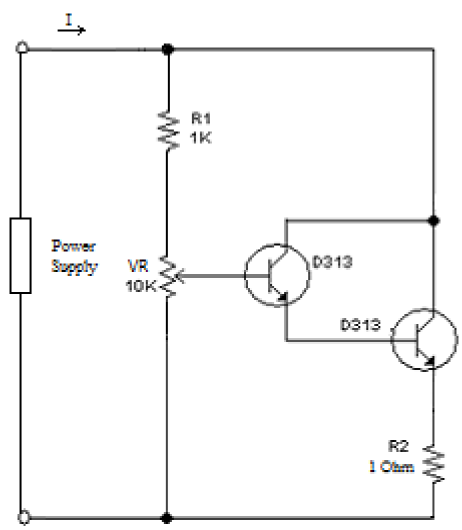


Figure 7a: Variable Load Circuit for Power Supply Testing

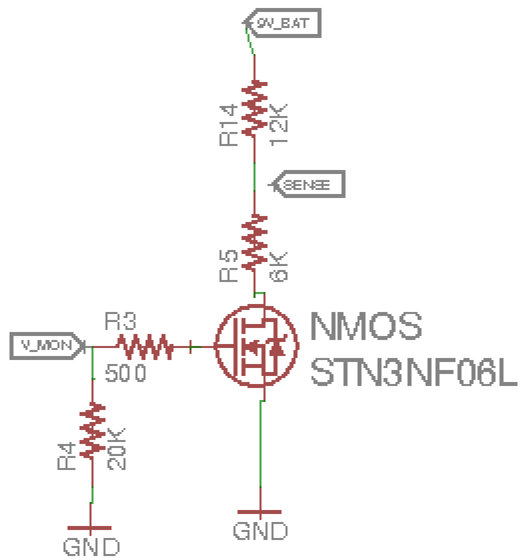


Figure 7b: Circuitry to Enable Battery Monitor

Figure 7b shows a voltage divider circuit that will be probed by pin 25 of the MPU to periodically monitor the voltage level, and therefore the battery capacity, of the 9V battery. Since keeping this circuit on at all times drains a significant amount of power, an NMOS is attached to act as a switch, so that the circuit is only active during probing.

9V Alkaline Battery (3 Points)	
1. Must supply voltage that is above the dropout range of the voltage regulator at a current draw of up to 100mA (2pt)	1. Design a variable load test circuit with the multimeter monitoring the voltage across the supply 2. Tune the load resistance to adjust the load current to 100mA, measure the supply voltage drop and check if it is above the 4.3V margin(regulator voltage plus dropout voltage);
2. The battery needs to support normal operation of the device for at least 1 hour (1pt)	1. Connect a load to the battery that draws consistent current; 2. Check if the voltage of the battery is above 5V after 1 hour.
Voltage Regulator (5 Points)	
1. Output voltage must be regulated to +3.3V	1. Connect the multimeter across the voltage

± 5% at a range of current draw (up to 100mA) as required by the modules (5pts)	regulator; 2. Check if the voltage drop between the input and output of the voltage regulator matches specification; 3. Use multimeter to track if the output voltage is stable; 4. Adjust the load current to 15mA and 30mA, respectively, repeat the above process.

Table 5: Power Supply Unit RV Table

2.2.2 Status LEDs

Input: 3.3V power supply input, digital signals from the microcontroller

Output: Red light emitted from the LEDs

Referring back to the microcontroller circuit, there are two LEDs connected on pins PD6 and PD7. Status LED1(PD6) is used to indicate power (ON/OFF of the overall system), and status LED2(PD7) is used to indicate device FSM data collection state. The values of the current-limiting resistor are calculated by:

$\frac{(3.3-2)}{0.01} = 130 \text{ Ohm}$

Status LED (0 Points)	
1. At forward current of 10mA, the LED should emit red light and visible at direct viewing angle	1. Connect the LED in series with a 330 Ohm resistor; 2. Use the multimeter to measure current through the LED; 3. At 10mA, make sure the LED light is visible from direct viewing angle

Table 6: Status LED RV Table

2.2.3 Input Buttons

Input: 3.3V power supply input

Output: Digital signal to the microcontroller

There are two physical buttons connected to the microcontroller on the master device. The one on the INT0 pin serves as a reset button, utilizing the special interrupt hardware in the microcontroller. The one on pin PC3 acts as the pair/start/stop button, depending on the current state of the device.

Sync Button (0 Points)	
1. Buttons should be debounced and should indicate correct signal transition upon pressed	1. Connect the button in series with a 330 Ohm resistor; 2. Supply 3.3 V;

	2. Use the oscilloscope to monitor the voltage transition across the resistor, make sure the signal is debounced and correspond to correct digital value (digital “High” when pressed and “Low” when released)

Table 7: Buttons RV Table

2.2.4 Microcontroller(ATmega328P)

Input: 3.3V power supply input, 3-axis acceleration data, 3-axis angular velocity data, 2 force-sensitive resistor voltage measurements, buttons, data from microSD card reader, data and signals from Bluetooth module

Output: Signals to sample the data from the combined 6-axis IMU, signal and data output to Bluetooth module

This module manages the data collection of the device by polling the IMU (inertial measurement unit; accelerometer + gyroscope) via the I2C interface as well as measuring the voltages across the two FSRs (force sensitive resistor). During collection, the data is buffered in the internal EEPROM and written into the microSD card through SPI in chunks of 512 bytes. During data transfer, the slave device reads the recorded data from its microSD card and transfers it to the master device via Bluetooth through the UART interface. The microcontroller is connected to a reset button on the hardware interrupt pin. In addition, the master device has a multipurposed button used to control its operating state.

This module is chosen based on its ability to handle serial communication of UART, SPI, and I2C at the transfer speeds shown in the calculations section. [7]

The module will be powered by 3.3V from a voltage regulator drawing power from a 9V battery. The circuit diagram of the microprocessor is shown below.

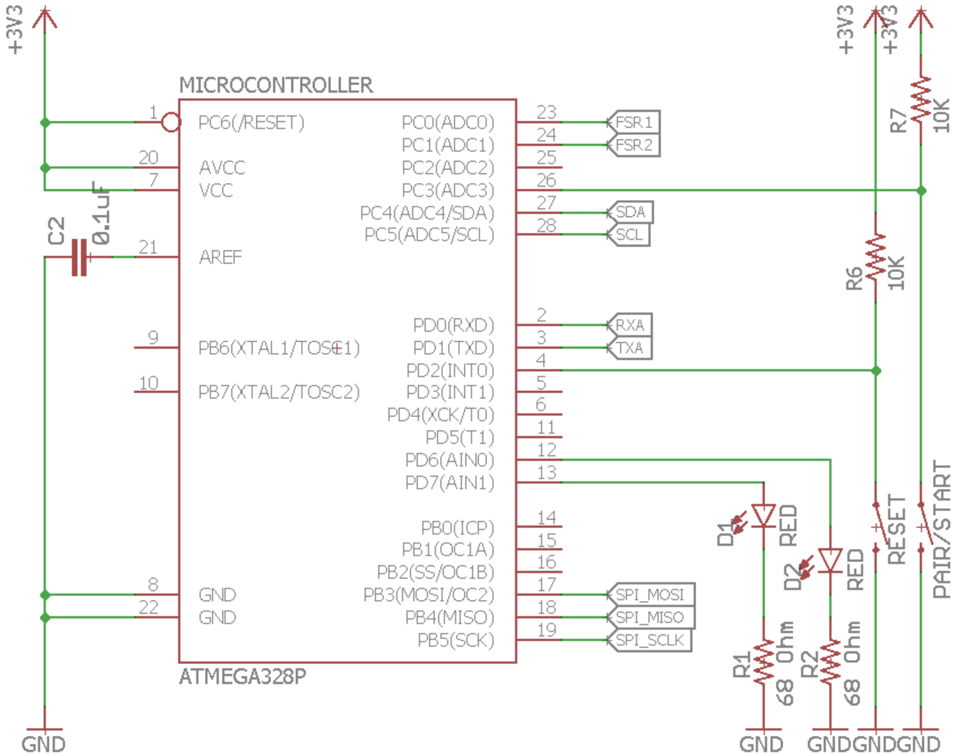


Figure 8: AVR Microcontroller Schematic

2.2.5 microSD Reader

Input: 3.3V power supply input, signal and data from microcontroller

Output: signal and data from the microSD card

The microSD card will act as the main storage device for the data collected. It connects to the microcontroller via SPI, and allows write to the SD card in 512 byte-chunks of data. This is a component external to our PCB, and will be connected via a 7-pin pin head port, as shown figure 6.

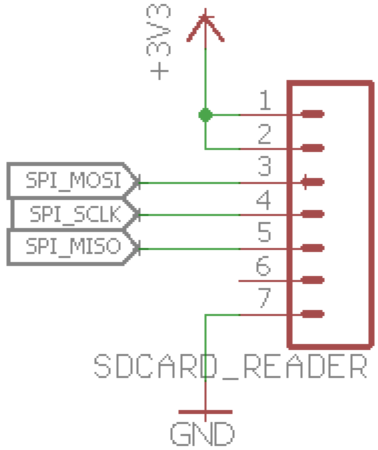


Figure 9: MicroSD Reader Schematic

Requirements	**Verification**
AVR Microcontroller (15 Points)	
1. Module is capable of running UART at a baud rate of 115200 (3pts)	1. Program the AVR to send 0x55 repetitively through the UART Tx pin 2. Probe the Tx pin with the oscilloscope 3. Verify the presence of a square wave with each pulse lasting ~8.68µs
2. Device is capable of writing text files to the microSD card that can be read from a PC (7pts)	1. Write a simple test program onto the AVR that uses the roland-riegel MMC/SD/SDHC card library to write 16-bit signed integers onto the microSD card 2. Verify on the PC that the number are readable and consistent with what was written
3. Device is capable of monitoring the voltage of the battery when the MOSFET is ON. The reported voltage should be within ± 5%.(2pt)	1. Get readings from the analog pin 25 that reports the voltage of the battery by using the formula Vbattery = 3*Vadc; 2. Check if the voltage matches voltmeter readings.

4. Device is capable of receiving data from the IMU via I2c at a minimum bitrate of 5580 bps (calculated requirement for maintaining the target 180 Hz sampling rate) (3pts)	1. Orient the IMU such that the gyroscope has a constant non-zero reading 2. Pull data from the IMU at 180 Hz for a second 3. Verify that the quantity and validity of data received meets expectations

Table 8: Microcontroller RV Table

2.2.6 Bluetooth Transceiver (HC-05)

Input: 3.3V power supply input, signal and data from microcontroller,
Output: Device-to-device signals between master and slave, data from microcontroller
The Bluetooth transceiver serves as the communication bridge between the master and the slave devices. The module runs on the UART ports of the microcontroller at 115200 baud in order to ensure a fast-enough data transfer rate during the data transfer state of both the master and the slave devices. The justification for this data rate is explained in the calculations section. This is a component external to the PCB and will be connected via a 4-pin pin head port, as shown in figure 5.

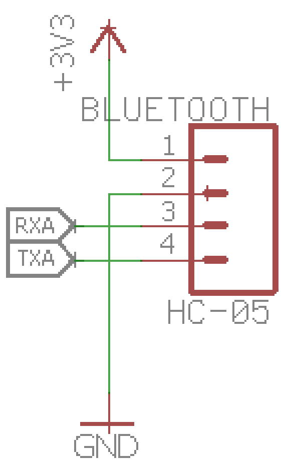


Figure 10: Bluetooth Transceiver Schematic

Bluetooth Module (5pts)	
1. Module is capable of receiving and transmitting over UART at a baud rate of 115200 (4pts)	1. Program an Arduino to start serial communication at the baud rate of 115200 2. Use the Arduino to program the Bluetooth module to operate at baud rate 115200 3. Connect the Bluetooth module to an Arduino (Tx->Rx, Rx->Tx)

	4. Short the Tx and Rx pins 5. Pair computer with Bluetooth module 6. Use a terminal program (eg. puTTY), connect to the correct serial line at baud rate 115200 7. Verify that what's typed into the terminal is echoed back. If so, the loopback test is successful
2. Two of such modules are capable of performing at the speed mentioned above when placed 2 ± 5% meters apart (1pt)	1. Repeat steps 1-6 from the verification above 2. Pre-populate a string field of length 115200kb 3. Move the HC-05 2 meters away from the testing PC 4. Send the pre-populated string and check for accurate echo response

Table 9: Bluetooth RV Table

2.2.7 Inertial Measurement Unit (IMU)

Input: 3.3V power supply input, polling signal from the microprocessor
Output: 3-axis acceleration data, 3-axis angular velocity data
This module contains a digital 3-axis accelerometer and a 3-axis gyroscope. For our purposes, we require the accelerometer to have a range of ± 16g, and the gyroscope to have a range of ± 125 dps (degrees per second). The targeted sampling rate for both the accelerometer and the gyroscope is 180 Hz each. This is to ensure high enough granularity for accurate data. The process of transferring data from the IMU to the microcontroller includes reading data value from one dimension of the IMU at a time. Thus, the microcontroller will read from the IMU at a frequency of 180 Hz; the reads from the IMU will rotate through all of its dimensions. When one set of all six dimensions has been read from the IMU, that will constitute to a successful poll. Thus, the microcontroller will poll at a rate of 30 Hz. Each read from the IMU will transfer two bytes of data. Data will be transferred via the I2C protocol.

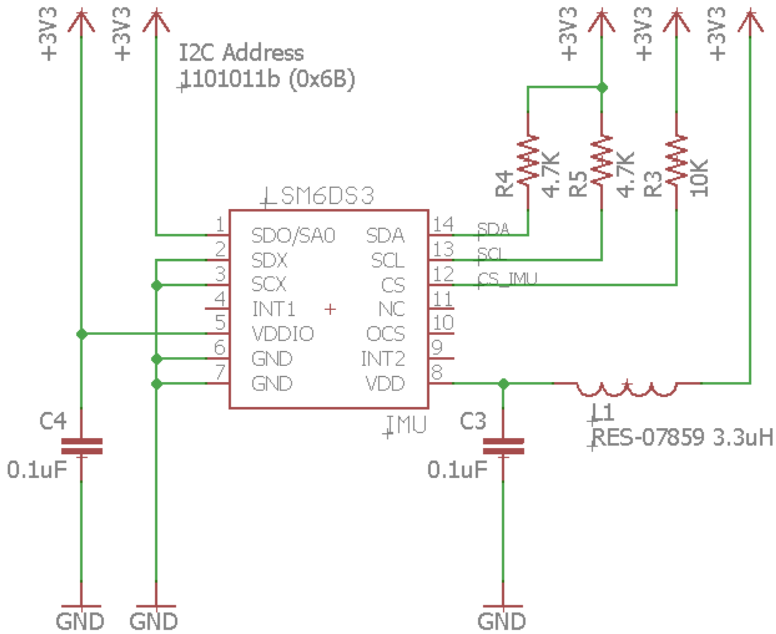


Figure 11: Inertial Measurement Unit Schematic

Accelerometer (5 Points)	
1. Accurate data output such that location tracking is within 5 cm of accuracy. Ensures accurate data for analysis	1. Download the phone app Sensor Kinetics, by INNOVATIONS, Inc. from iOS's app store or Android's Play Store. The Pro version is required 2. Connect the accelerometer to a microcontroller. Load test code onto the microcontroller. The test code will collect and store data from the accelerometer 3. Open the Sensor Kinetics app and press the graph button to the right of the "Accelerometer Sensor" header. This will bring the app to the accelerometer tracking mode. 4. Attach the accelerometer to the phone so it that it will not move when the phone is in motion. 5. Start the data collection for the phone app and the microcontroller at the same time. Then grab the phone and move around with

	varying speeds; shaking the phone and sweeping the phone in all directions. 6. After about 10 seconds stop the data collection for the phone and the microprocessor at the same time. 7. On the phone app, press the downward arrow at the top right and then press” Files & Sharing” to view the data collected from the phone’s accelerometer 8. Compare the data collected from the phone and accelerometer to verify that the specification is met
3. Samples at a rate of at least 180 Hz in order to give enough resolution to represent movement	1. Connect the accelerometer to the microprocessor and load a test code that will collect and log the data from the accelerometer 2. Obtain a stopwatch to record the time data is collected. Collect data for 10 seconds. 3. Analyze the data collected to ensure that there are enough entries to meet the requirement of 180 Hz (for 10 seconds example there should be at least 1800 entries)
Gyroscope (5 Points)	
1. Accurate up to ± 7 degrees in order to assure accuracy in data	1. Download the phone app Sensor Kinetics, Pro version, by INNOVATIONS, Inc. available from iOS’s app store or Android’s Play Store. 2. Connect the gyroscope to a microcontroller. Load test code onto the microcontroller. The test code will collect and store data from the gyroscope 3. Open the Sensor Kinetics app and press the graph button to the right of the “Gyroscope Sensor” header. This will bring the app to the gyroscope tracking mode. 4. Attach the gyroscope to the phone so it that it will not move when the phone is in

	motion 5. Start the data collection for the phone app and the microcontroller at the same time. Then grab the phone and move around with varying speeds; twisting the phone and sweeping the phone in all directions. 6. After about 10 seconds stop the data collection for the phone and the microprocessor at the same time. 7. On the phone app, press the downward arrow at the top right and then press” Files & Sharing” to view the data collected from the phone’s gyroscope 8. Compare the data collected from the phone and gyroscope to verify that the measurement is within ± 5 degree difference
2. Samples at a rate of at least 180 Hz in order to give enough resolution to represent movement	1. Connect the gyroscope to the microprocessor and load a test code that will collect and log the data from the gyroscope 2. Obtain a stopwatch to record the time data is collected. Collect data for 20 seconds. 3. Analyze the data collected to ensure that there are enough entries to meet the requirement of 180 Hz (for 20 seconds example there should be at least 3600 entries)

Table 10: IMU RV Table

2.2.8 Force Sensitive Resistors with Passive Low-pass Filter

Input: 3.3V power supply input, physical force applied to the sensor area
Output: Resistance change reflected by voltage level to the microcontroller
The force sensitive resistors (FSR) serve as the force sensor in our design. The amplitude of the pressure will be reflected by the resistance of the FSR. There will be two FSRs installed beneath the insole of each skate. FSR1 will be placed between the third and fourth toe mounds and FSR2 will be placed on the heel area. The sensors will each be connected in series with a 20kOhm reference resistor. The output terminal is connected to the output port of an Op-Amp to improve accuracy of the force sensor. In reality, because the ADC(analog input) pins of the microcontroller have a load impedance of only around 10 kOhms. This finite load impedance is connected in parallel with the reference resistor, making the equivalent reference resistance

less than the 20kOhm nominal value. Therefore, it is necessary to include the Op-Amp in our design. In addition, the natural vibration during skating due to the uneven ice surface that causes high frequency noises of the analog data requires denoising of the data. As a result, we decided to design a low-pass filter with a cut-off frequency of 20Hz(this is a theoretically value we are using). More force calculation will be discussed in depth in section 2.5.3.

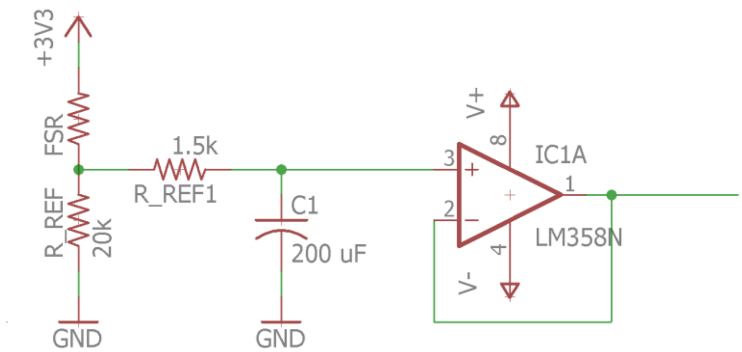


Figure 12: FSR Schematic with Low-Pass Filter

Force Sensitive Resistor and Low-pass Filter (12 Points)	
1. Pressure measurement range needs to be from 1psi to at least 6.409psi (Equivalent: force measurement range needs to be between 14-90N); The measurement needs to be within ±10% of the theoretical value at the two boundary values. (4pts) *see derivation of the values used here in the calculation and simulation section	1. Place the FSR on a flat surface and connect the FSR in series with a 20 kOhm reference resistor R_{ref}; 2. Supply 3.3V voltage across the two resistors; 3. Place a 1.4kg weight on top of the FSR; 4. After the readings stabilize, measure the voltage drop across the reference resistor, V_{ref}; 5. Find V_{FSR} by using the relation V_{FSR} = 3.3V - V_{ref} and then find the current across the components I = V_{ref} / R_{ref}; 6. Use the voltage and current found to calculate FSR resistance, compare with theoretical value and check if it is within 10% error margin; 7. If the value does not match the standards, change the weight until the value meets the requirement, record the adjusted value and check if it is among a reasonable range; 8. If the value matches the standards, move on to replace the 1.4kg weight with a 9.2kg

	weight and repeat the above steps.
2. Measurement result needs to be accurate within ±10% of the theoretical value (3pts)	1. Select 5-10 weight values between the two threshold values; 2. Repeat the same steps as specified in part 1 of FSR verification and record the corresponding measurements; 3. Use the values obtained to plot the Pressure vs Force curve, fit a trend line; 3. Check if the trendline matches theoretical curve as shown in Fig. 4 and the points have less than ±10% error.
3. Must draw current less than 1uA at resting state (2pt)	1. Supply 3.3V voltage across the resistor; 2. Measure the current drawn from the supply using the multimeter;
4. The low-pass filter must have -3dB frequency of 20Hz ± 10%; (3pt)	1. Use the function generator to generate a sinusoidal input with amplitude of 50mV at frequency of 5Hz and slowly increase the frequency to 30Hz while using the oscilloscope to monitor the amplitude; 3. When the amplitude dropped to $35.34mV$, verify that it is within ± 10% of 20Hz.

Table 11: FSR RV Table

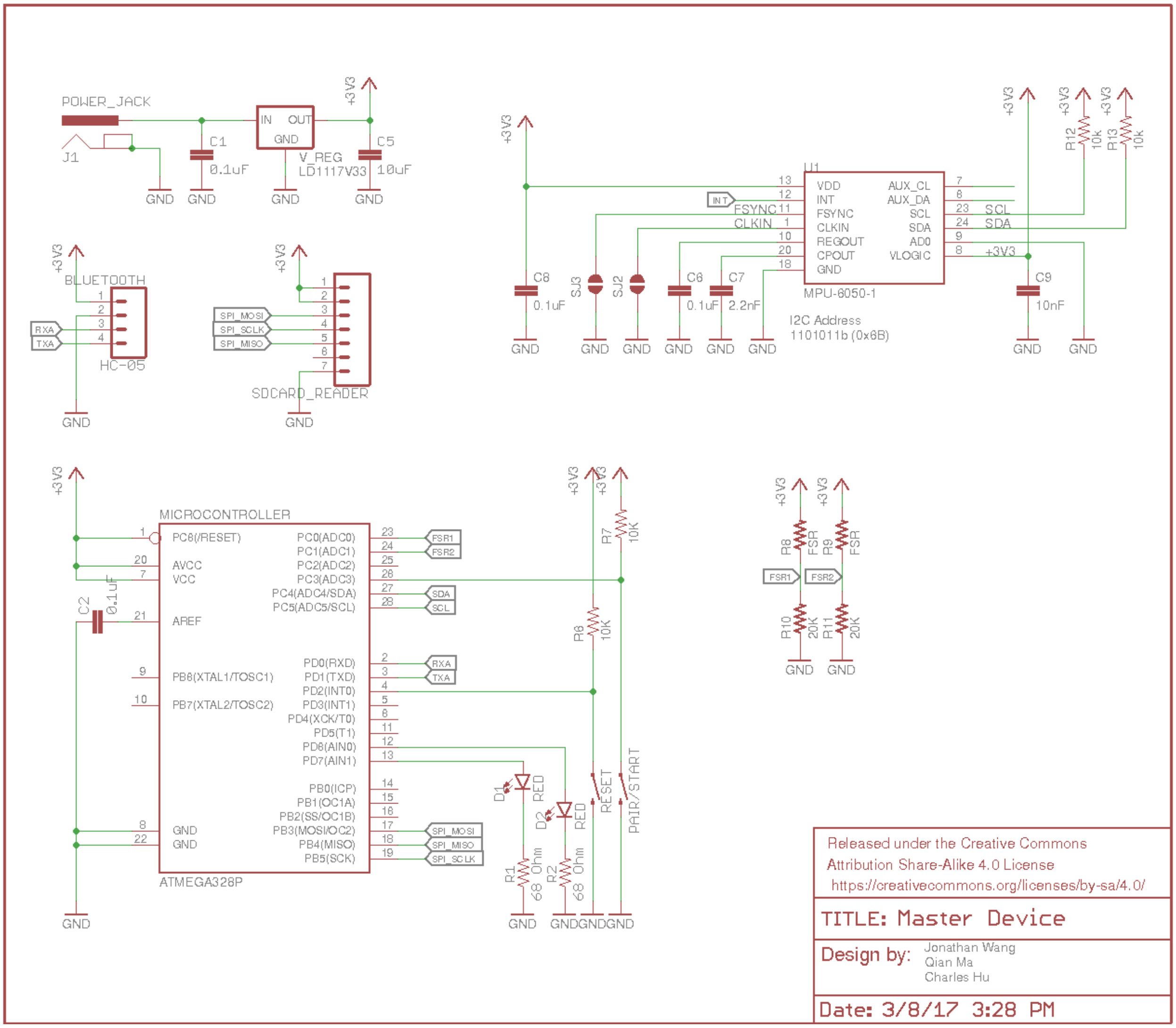


Figure 13: Overall System Schematic

2.3 Software Description

2.3.1 Post Processing Software

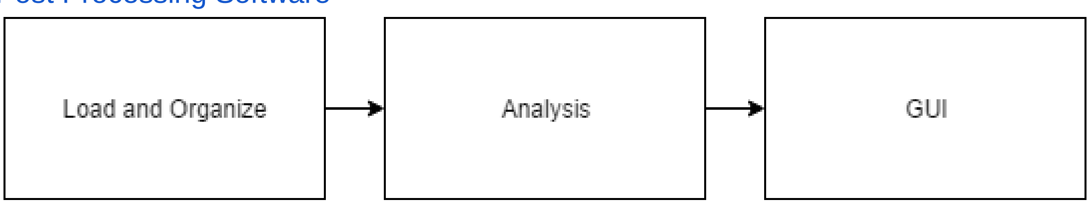


Figure 14: Software Flowchart

General

The above flow chart shows the general procedure of the post processing software. This procedure is meant to analyze the data obtained by the device and give meaningful feedback to user on their skating form. The base code will detect differences between the user's left and right leg skating form and notify the user accordingly. This procedure will be prototyped in MATLAB.

Load and Organize

The SD from the master device will first be inserted into a computer, and then a script will be run to facilitate the whole process. Initially the data from the SD card will be loaded off and organized into array form, two for data collected from the accelerometers, two for data collected from the gyroscopes, and four for data collected from the force sensors. All of the arrays will be timestamped or have a time array in relation to sensor collected data. Once the data is successfully loaded off the SD card, the analysis algorithm will be run.

Analysis Algorithm

The analysis algorithm is meant to detect asymmetry between the right and left legs and also determine if the user's form fulfills the metrics that determine efficient skating. The analysis algorithms described below will calculate metrics for both right and left leg. These metrics will then be compared in order to determine if the user's skating form is symmetric. Symmetry is characterized with the following proposed equation, the Weighted Extracted Asymmetric Ratio (WEAR). x_{left} defines the left metric that is being compared and x_{right} defines the right leg of the same metric. If the WEAR is true then the user is asymmetric for the compared metric.

$$\frac{\left|\mathrm{x}_{\mathrm{left}} - \mathrm{x}_{\mathrm{right}}\right|}{\min\{x_{left}, x_{right}\}} > 0.1$$

There are certain quantitative metrics that the algorithm will look for in order to determine efficient skating. The first quantity deals with the skate angle. During a stride, where a leg pushes to propel the user forward, the skate should make a 45 degree angle with the ice. This ensures that the skate has the maximum grip on the ice and therefore maximizes the efficiency of the stride. The second quantity being looked at is the largest translational displacement of skate. This is information will be compared to the length of the user's leg to make sure that the user is fulling extending during the strides. The length of the leg will either be inferred based on the user's height input or through calibration.

The overall algorithm can be split up into three phases. Phase 1 will set time markers for the data set. The purpose of the markers is to set boundaries between each stride. Phase 2 involves analyzing the weight transfer from the heel to the toe during the stride. Phase 3 will calculate the translational and angular displacement within each stride. The markers set in phase 1 will be used to determine the rough beginning and end of each stride.

<u>Phase 1</u>

This phase breaks down the data that is collected into sets of data; each set of data represents one stride. This is necessary for calculating displacement from the acceleration data.

Calculating displacement strictly from acceleration data introduces inaccuracies, which gets compounded the more the calculation goes on. Therefore the displacement calculation will be done strictly within each stride data set. We believe this way the inaccuracies will not affect the results that much.

The algorithm for this phase is proposed based on the force data collected from previous research [1]. We assume that heel force we plan to collect will look similar to the researches force data. Thus the algorithm will first calculate the peaks of the waveform. MATLAB has a library for detecting peaks in noisy signals. Then “period markers” will be placed at times in between each peak. These period markers will essentially split the data into sections; the data in each section will represent one stride.

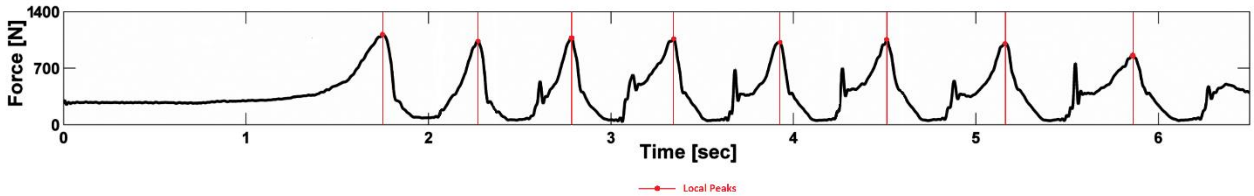


Figure 15: Using Force Data for Stride Definition [1]

<u>Phase 2</u>

This phase also analyzes the force data, except the analysis will compare the heel force data with the toe force data. The purpose is to quantify weight transfer from the heel to the toe during a stride. Then weight transfer for the left leg and right leg is checked for symmetry.

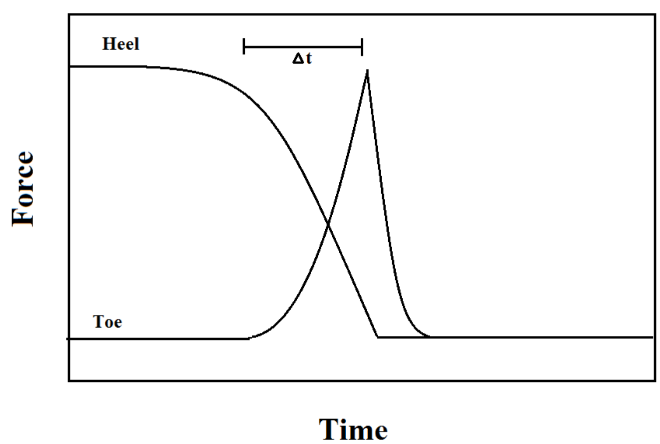


Figure 16: Theoretical Overlay of Force Plot

The analysis for one foot will overlay the heel force data with the toe force data. Theoretically, the force from the heel should peak first and then decrease. At the same time the force from the toe would increase and peak around when the heel force is at a minimum. When skating efficiently the weight should be transferred from heel to toe during a stride. The time difference between the heel peak and toe peak is then calculated. This metric will be used to compare the left side to the right side. The force data is also periodic, so the metric calculation will be done for each stride, defined by the markers, and then averaged. The averaged left and right time difference is then used in the WEAR to determine symmetry. We believe that if the user has an asymmetrical habit then averaging the time difference for all the strides of one foot will not compromise the comparison of the left and right legs. This is because the habit theoretically should be evident in all the user's strides.

Phase 3

In this phase, the algorithm will calculate the translational and rotational displacement based on the collected acceleration data. Both translational calculations will be done within each stride data set. The result of the calculations will be used for symmetric comparison and to compare with the quantitative metrics that were described above.

For symmetry analysis, the max translational displacement for each stride will be gathered and then averaged for each leg. Then the averaged values will be used in the WEAR to determine if the user's leg extension is symmetrical. The same is done with the rotational data; the feedback is used to determine if the user is opening his/her hips enough during the stride and if their skate is gripping the ice at the same angle.

Then the translational displacement will be parsed to check if the user fully extended his/her knee during each stride. The same thing will be done with the rotational data to make sure that the user's skate is at about 45 degrees to the ice during each stride.

GUI

After data analysis is performed, the calculated output is transferred to the GUI. The GUI will output the calculated displacements on time related graphs. It will also display a force graph where the heel and toe force data are overlaid over each. The outputs of the WEAR will also be displayed as feedback to the user.

2.4 Calculation and Simulation

2.4.1 Current and Power Calculation

Load	Operating Voltage	Current Draw (Active Mode)	Current Draw (Sleep Mode)
Microcontroller	3.3V	4.0 mA	4.0 uA
IMU Unit	3.3V	1.25 mA	425 uA
Force-Sensitive Resistor	3.3V (in series with R_{ref})	Max: 0.066m A	0 mA
Bluetooth Module	3.3V	40 mA(pairing) 8 mA(active)	3 mA
LED (2)	3.3V	10 mA	0 mA
Voltage Regulator	9V	5mA	0 mA
microSD Card	3.3V	30mA	200 uA

Table 12: Component Current Draw

Max total Current Draw (active) = 100.316 mA
Dynamic Power (active max) = V*I = 331.043 mW

Normal total current Draw (active) = 62.316 mA
Dynamic Power(active) = V*I = 205.643mW

Total Current Draw (sleep) = 3.629 mA
Dynamic Power(sleep) = 11.976 mW
9V Alkaline Battery Capacity = 500 mAh
Battery life for max active usage: 500/100.316 = 4.84 h
Battery life for normal active usage: 500/62.316 = 8.02 h
Battery life for sleep mode: 500/3.629 = 137.78 h

2.4.2 Sensor Resolution

IMU acceleration:

The LSM6DS3 provides the configuration of ±2g, ±4g, ±8g, and ±16g range. With the intensity of ice skating activity [1], it is necessary that our design uses the ±16g scale where g is the gravitational acceleration with value 9.81 m/s^2. Therefore, the resolution of the acceleration data is:

$$\text{Acceleration Resolution} = \frac{|\text{Measurement Range}|}{2^{\text{number of data bits}}} = \frac{16\text{g}}{2^{15}} = 4.79\ \text{mm}/\text{s}^2 \quad (1)$$

Force measurement Resolution:

The output of FSR is a continuous analog voltage, which will be converted to an 8-bit unsigned value. Data loss will occur during conversion to 256 digital levels. The voltage ranging from 0 to 3.3V will be converted into an unsigned integer from 0 to 255. Therefore, the lowest voltage interval that can be resolved is:

$$\text{FSR Voltage Resolution} = \frac{|\text{Measurement Range}|}{2^{\text{number of data bits}}} = \frac{3.3}{2^{8}} = 12.89\ \text{mV} \quad (2)$$

Due to the non-linearity of the voltage and pressure relationship, each voltage interval will represent different pressure difference. For detailed discussions refer to section 2.4.3.

2.4.3 Force Calculation

In the below calculation, we are making the following assumptions:

(1) The maximum force during acceleration is 1000N [1], the value accounts for both the weight of the skater and the force exerted during acceleration;
(2) The pressure applied is evenly distributed over the area of the foot;

To find the approximate value for area of the foot, we will be using the average American shoe size of 8.5 for women and 10.5 for men. Since the area is inversely proportional to pressure, to find maximum pressure, we will use size 8.5 in the calculation.

Approximate foot area:

$A_f \; = \; 24.6\text{cm} \times 9.2\text{cm} \; = \; 226.32 \, cm^2 \; = \; 0.0226 \, m^2$ (1)

Maximum pressure:

$P_{max} \; = \; \frac{F}{A_f} = \; \frac{1000N}{0.0226m^2} = \; 44185.2 \, Pa = 6.409 \, psi$ (2)

1 inch Diameter Sensor Area:

$A_{sensor} \; = \; \pi \times (0.5)^2 \; inch^2 = 0.00202677m^2$ (3)

Maximum force applied to the sensor:

$F_{max} \; = \; P_{max} \times A_{sensor} \; = 44185.2 \times 0.00202677 = 89.55 \; N$ (4)

With the known values from the datasheet [5] provided by the manufacturer, we were able to fit a trendline as shown in the figure below. The trendline has a coefficient of determination $R^2 = 0.9991 = \; 99.91\% < 5\%$ error margin.

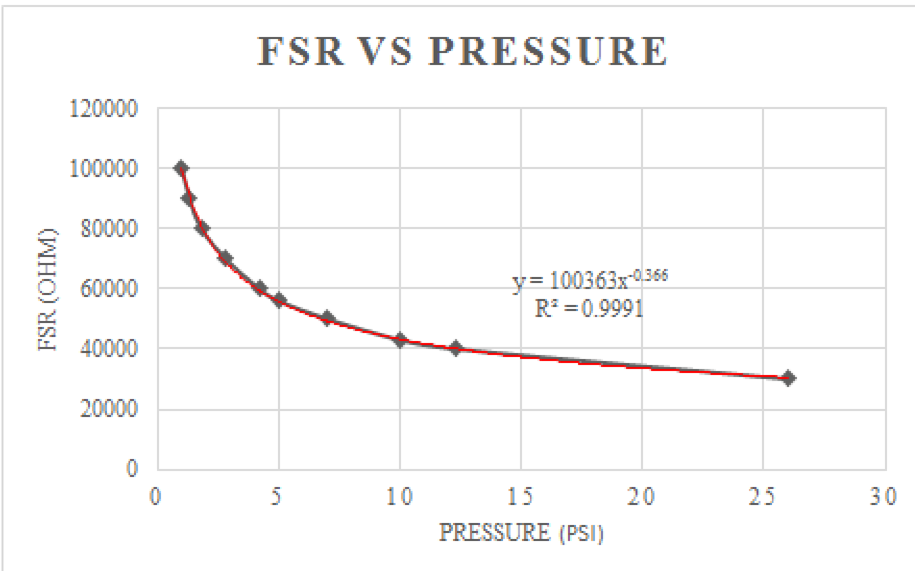


Figure 17: Low Pressure FSR Values with Trendline

Within the range of 1psi to 25psi, the FSR behavior follows a power curve of:

$FSR = 100362 * P^{-0.366}$ (5)

With equation (5), we can get the following relation:

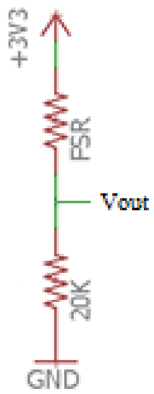
$\text{P} = (\frac{100362}{\text{FSR}})^{\frac{1}{0.366}}$ (6)

Based on figure shown below, the calculation from analog voltage to FSR value is as follows:

Voltage devider rule:

$\frac{\text{V}_{\text{OUT}}}{3.3\text{ V}} = \frac{20\text{k}}{20\text{k}+\text{FSR}}$ (7)

$\text{FSR} = \frac{66\text{k}}{\text{V}_{\text{OUT}}} - \; 20\text{k Ohm}$ (8)



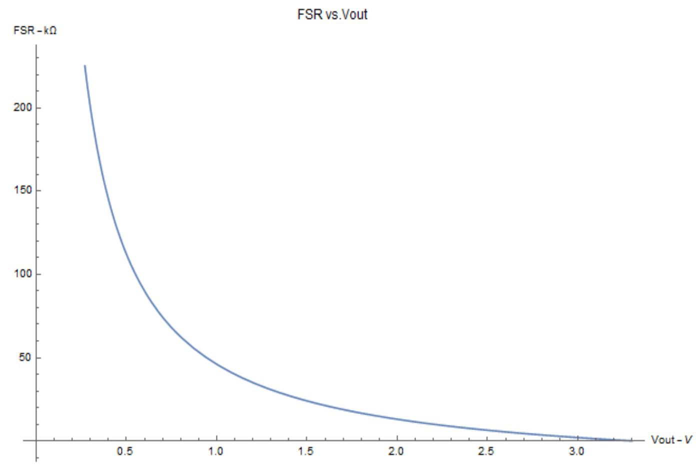


Figure 18: FSR Configuration Figure 19: FSR vs Vout

Combining (6) and (8), then we can derive the formula to calculate pressure from the analog voltage:

$$P = \left(\frac{100362}{\frac{66k}{V_{OUT}} - 20k}\right)^{\frac{1}{0.366}} \text{psi} \qquad (9)$$

Notice that there are a few factors that contribute to the non-linearity of the relationship. Firstly, the relationship between pressure and analog output voltage is non-linear. Secondly, the analog voltage will be converted to 256 digital levels, which implies that each voltage interval represents different pressure change.

2.4.4 Data Collection Rate Justification

Since our device functions as a data logger, the data collection and transfer rates must be carefully considered.

From our research and observation on typical IMU outputs, we've determined that a sampling rate of 30 Hz from all sensors provides the best overall balance between ease of process, usefulness, and economizing the data storage used.

At 30 Hz, we must finish polling all sensors within:

$$\frac{1}{f} = \frac{1}{30} = 33.3 \text{ ms} \qquad (1)$$

IMU data via I2C:

Since we're polling the data one axis at a time in sequence, we must poll 6 times from the IMU to form one complete set of data. From each poll, we expect a 2-byte data [6] from each axis. Combining data from all six axes:

$$2 \text{ bytes} \times 6 = 12 \text{ bytes} \qquad (2)$$

Since the I2C transfer rate is 400 kb/s (50 B/ms), all the data is expected to be transferred within:

$$\frac{12 \text{ bytes}}{50 \text{ B/ms}} = 0.24 \text{ ms} \qquad (3)$$

This is a lower bound, as we expect to write in timestamp information along with the data collected.

Since 0.24 ms < 33.3 ms, I2C is capable of transferring all the data in time.

Storage with SD Card:

We expect two 1-byte data from the analog pin connected to the FSR. Therefore, we must store 14 bytes of data at 30 Hz, which requires a rate of:

$$(12 + 2)\text{bytes} \times 30\ \text{Hz} = 420\ \text{Bps} \quad (4)$$

The microSD card is written in blocks of 512 bytes, and will take an average of 20 ms to write. Since our 30 Hz gives us ~33 ms between each poll, the data must be buffered in 512-byte blocks in the internal EEPROM, so as to not disturb with the flow of data collection.

Bluetooth Baud Rate:

We specified the Bluetooth to be working at a higher baud rate than default due to the amount of data that needs to be transferred. For a 10-minute session, we expect 25.2 MB of data. At 115200 baud, we have 115.2 kb/s, which converts to 14400 KB/s. With 25.2 MB of data, it takes 1.75s to complete the transfer, without taking into account bookkeeping bits. The default 9600 baud is only 8% of this speed, and is therefore unsuitable for this task.

2.4.5 Variable Load Current Testing Circuit Analysis

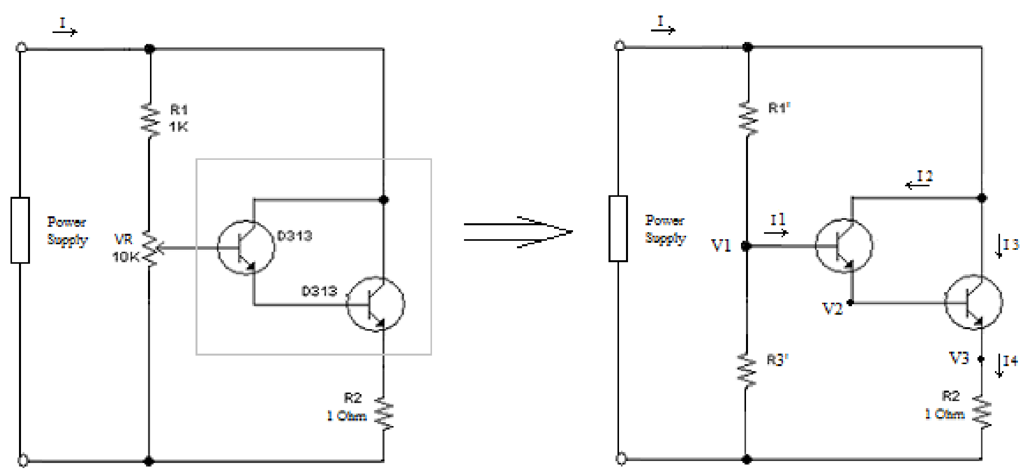


Figure 20: Variable Load Circuit Analysis

We designed a variable load circuit to test the stability of our power supply by utilizing the current amplifying properties of Bipolar Junction Transistors. As shown in Fig.17, we can draw the equivalent circuit to simplify the analysis.

The two BJTs form a Darlington pair, which is commonly used to amplify small current [8].

We know from the characteristics of BJTs that:

$$V_1 - V_2 = V_2 - V_3 = 0.7\text{V} \quad (1)$$

If we use two identical BJTs with DC current gain β, then:

$I_2 = \beta I_1$ (2)

$\frac{I_4}{I_3} = \beta+1$ (3)

Combining (2) and (3), we get:

$I_4 = (\beta + 1)^2 I_1$ (4)

Using Ohm's Law,

$V_3 = I_4 R_2$ (5)

$V_1 = I_4 R_2 + 1.4$ (6)

Using KCL:

$I = I_4 + \frac{V_1}{R_3'} \approx I_4$ (7)

Once we know the voltage V_1, we can calculate I_1 accordingly. By varying the ratio of R_1' and R_3', we can tune the total current I.

2.5 Tolerance Analysis

Most of the components in our device are durable and can withstand small fluctuations in accuracy within a defined margin. However, the IMU data contains measurements of six axes, which increases the probability that errors exceeding the ±5% threshold. In addition, accuracy of the IMU data is essential because any errors in the measurement data will propagate to the post-processing software program, which will lead to failure in providing accurate feedback. Thus, we conclude that our inertia measurement unit is the critical component. The IMU must have a certain level of accuracy to successfully analyze the user's skating form.

We define the tolerance levels for the IMU to be the following:

- Accelerometer's tolerance allows data to be accurate if within 1 cm.
- Gyroscope's tolerance allows for ±5 degrees in any direction

These tolerance standards must be met for the data to be deemed accurate for analytical processing.

Inaccuracy of the sensor can propagate throughout the analysis algorithm. For instance, let one IMU's gyroscope overshoot its accuracy by 5 degrees. Then for example average angle obtained from the faulty gyroscope is 45 degrees and the non-faulty IMU outputs an angle of 40 degrees. However, both IMU's should have outputted 40 degrees. If both outputted 40 degrees, then the analysis would say that the skater has symmetry for the angular rotation metric. But with the faulty IMU the calculation for symmetry goes as such:

|45 - 40| / 40 = .125

This result would dictate asymmetry, which is false.

The accelerometer and gyroscope both contain three axes. The data for each axis is stored as a 16-bit signed integer with 1 sign bit store the sign of the signal and 15 data bits storing the magnitude. Therefore, if we assume each data bit has a probability of p to fail, then the 15-bit sequence forms a Binomial Distribution B (15, p).

The mean of the distribution: 15p
The variance of the distribution: 15p(1-p)
For either gyroscope or accelerometer, the probability that all three axes, a total of 45 data bits are error free is:
$P_{zero-error} = (1-p)^{45}$
Therefore, the probability that some error will occur would be:
$P_{error} = 1 - P_{zero-error} = 1 - (1-p)^{45}$

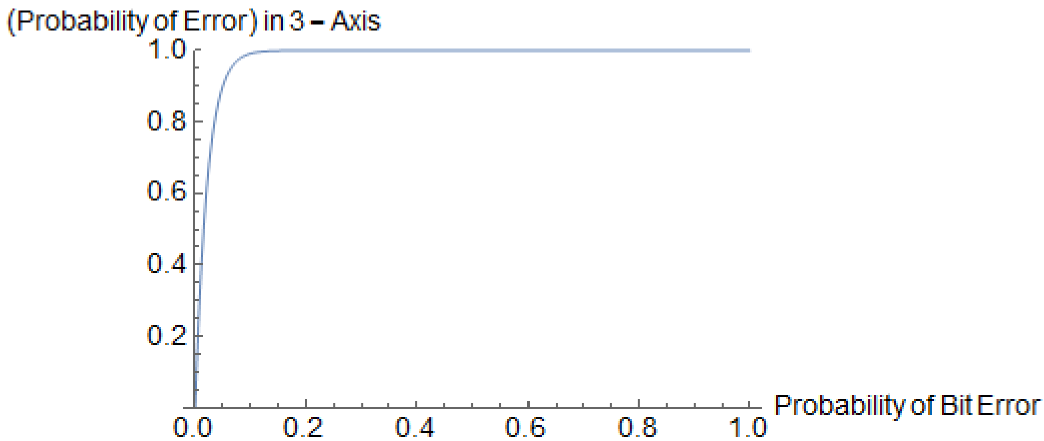


Figure 21: Probability of Error in 3-Axes of Data

3 Ethics

3.1 Ethics Justification

This product is purposed to help people maximize their efficiency while skating. We believe that this purpose and the device's implementation is strictly following the IEEE Code of Ethics, #9:" to avoid injuring others, their property, reputation, or employment by false or malicious action;" [4]. We plan to stick to our purpose during the whole process, and everything will be documented for complete transparency. The documentation will not only act as proof showing that we did not deviate from our original goal but also be part of our effort to follow the IEEE Code of Ethics, #3: "to be honest and realistic in stating claims or estimates based on available data;" [4]. Throughout our design process, we will take responsibility to address the potential safety issues and make every effort to minimize them. We believe that this is our commitment to the IEEE Code of Ethics, #1:" to accept responsibility in making decisions consistent with the safety, health, and welfare of the public, and to disclose promptly factors that might endanger the public or the environment;" [4].

3.2 Safety

Based on the design components, there are a few potential safety concerns:

3.2.1 Device Overheating

As with most electronics there is always a possibility that a device could overheat based on unforeseen circumstances. In our case, the sealed packaging might lead to increased temperature. As mentioned in section 2.5.1, the power consumption during active mode is around 330 mW and less than 12 mW during sleep mode. For safety precautions, we would recommend that the user switch the device to sleep mode after 10 minutes of active use. In the unlikely chance that our device overheats, the safety concern would be sustained burns due to handling the device. The overheating device could also damage the skates or the laces but safety issue regarding that would be minimal.

3.2.2 Battery Safety

Considering that the device will mostly operate in ice rink with potential water contact, the sealed housing is essential to reduce risks related to battery. Among the battery options available to supply a small low powered device for an extended period of time, we decided that the 9V alkaline battery is a suitable choice. Compared to Li-Po batteries, alkaline batteries are much less likely to cause fire hazards. In addition to fire hazards, we do need to address that the electrolyte in these cells is potassium hydroxide (KOH), which is caustic and can cause tissue damage when in contact with skin. Therefore, one should avoid direct contact with the battery if there is potential leakage.

3.2.3 Hardware failure

In the case of a hardware failure, everything will most likely be contained due to the sealed packaging. There will also be no shorting issues because the boot of the skate will act as an insulator between the device and the user's foot.

3.2.4 Mounting failure

There is a possibility that mount could fail and the device falls off while the user is skating. This could cause injury to the user or those around him/her should they accidently trip over the device.

4 Cost and Schedule

4.1 Cost Analysis

4.1.1 Labor

Team Member	Hourly Rate	Hours	Total	Total x 2.5
Charles Hu	$30	200	$6,000	$15,000
Qian Ma	$30	200	$6,000	$15,000
Jonathan Wang	$30	200	$6,000	$15,000

Total Labor				$45,000

Table 13: Labor Cost

4.1.2 Parts

Item	Quantity	Vendor	Cost/unit	Cost
ATMega328P	2	SparkFun	$4.3	$8.6
Wireless Bluetooth RF Transceiver Module Serial RS232 TTL HC-05	2	Amazon	$6.61	$13.22
IMU – MPU6050	2	InvenSense	$5.45	$10.9
1 Inch ShuntMode FSR	4	Sensitronics	$6	$24
Voltage Regulator - 3.3V	1	SparkFun	$1.95	$1.95
LEDs	3	Amazon	$0.175	$0.525
9V Energizer 522 Battery	2	Amazon	$3.82	$7.64
microSD Reader	2	SparkFun	$4.95	$9.90
Total				**$72.705**

Table 14: Parts Cost

4.1.3 Grand Total

Grand total = Parts Cost + Labor = $45,072.975

4.2 Schedule

Week	Task	Responsibility
1/16/17	**Initial Post Due**	All
1/23/17	Come up with a project idea	All
	Apply for RFA	All
1/30/17	**Project Approval Due**	All
	Work on proposal	All
2/6/17	**Project Proposal Due**	All

	Work on design documentation	All
2/13/17	**Mock Design Sign-up Closes**	Qian
	Work on design documentation	All
2/20/17	**Design Review Sign-up Closes**	Jonathan
	Design Document Due	All
	Attend mock design review	All
2/27/17	**Attend Design Review**	All
	Purchase Parts	All
	Build power supply and test	Qian, Charles
	Check microprocessor for RV and upload basic code	Jonathan
3/6/17	Purchase parts; Integrate microprocessor and interface with IMU	Jonathan
	Write test code for IMU, load onto microprocessor	Qian
	Test IMU for RV, integrate with microprocessor	Charles
3/13/17	Design PCB	All
	Test Force Sensor for RV, integrate with microprocessor	Qian
	Test SD card for RV, integrate with microprocessor	Jonathan, Charles
	Collect specification for casing to give to machine shop	Qian
3/20/17	**Spring Break**/ Buffer Week	All

	Last Day for First Revision of PCB	All
	Work on individual progress reports	All
3/27/17	**Individual Progress Reports Due**	All
	Last Day for Revisions to Machine Shop	All
	Test Bluetooth module for RV and integrate into device	All
	Finalize PCB design	All
4/3/17	**Last Day for Revised PCB**	
	Implement processing algorithm	All
4/10/17	Design GUI for user feedback	Qian, Charles
	Solder components onto PCB and test	Jonathan
4/17/17	**Mock Demo**	All
	Demonstration Sign-up Closes	All
	Mock Presentation Sign-up Closes	All
	Presentation Sign-up Closes	All
	Verify device, debug problems, buffer week	All
	Prepare presentation and demonstration	All
4/24/17	**Mock Presentation**	All
	Demonstration	All
	Finalize verification	All

	Start writing final report	All
5/1/17	**Final Papers Due**	All
	Lab Notebook Due	All

Table 16: Weekly Schedule

Bibliography

[1] E. Buckeridge, M. C. LeVangie, B. Stetter, S. R. Nigg, and B. M. Nigg, "An on-ice measurement approach to Analyse the Biomechanics of ice hockey skating," PLOS ONE, vol. 10, no. 5, p. e0127324, May 2015.

[2] J. Pearson, "Researchers at Brigham young developed a smart skate," in MOTHERBOARD, Motherboard, 2014. [Online]. Available: https://motherboard.vice.com/en_us/article/this-smart-skate-measures-how-hard-figure-skaters-land. Accessed: Feb. 8, 2017.

[3] X. Robert-Lachaine, "Force measurement and ankle motion of the forward skating and crossovers with a standard hockey skate and a modified hockey skate," 2010.

[4] "IEEE code of ethics," in IEEE.org. [Online]. Available: http://www.ieee.org/about/corporate/governance/p7-8.html. Accessed: Feb. 8, 2017.

[5]"1" Shunt Resistance Resistance vs. Pressure," Sensitronics. [Online]. Available: http://www.sensitronics.com/product-res/1-inch-shunt-mode-fsr/docs/1-inch-shunt-mode-medium-res.pdf. Accessed: Feb. 25, 2017.

[6]"LSM6DS3: Always-on 3D accelerometer and gyroscope - Application Note," STMicroelectronics, 2017. [Online]. Available: http://www.st.com/content/ccc/resource/technical/document/application_note/12/98/b4/44/a5/bf/4e/c5/DM00157511.pdf/files/DM00157511.pdf/jcr:content/translations/en.DM00157511.pdf. Accessed: Feb. 25, 2017.

[7]"ATmega328/P DATASHEET SUMMARY," Atmel, 2016. [Online]. Available: http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-42735-8-bit-AVR-Microcontroller-ATmega328-328P_Summary.pdf. Accessed: Feb. 25, 2017.

[8]Hamuro, "Variable dummy load for power supply testing," 2013. [Online]. Available: http://www.deeptronic.com/electronic-circuit-design/variable-dummy-load-for-power-supply-testing/. Accessed: Feb. 27, 2017.