AUTOMATIC HANDSHAKE CONTACT INFO EXCHANGER

By

WILLIAM HANLEY

KUANYSH SAMIGOLLAYEV

AMBIECA SAHA

DESIGN REVIEW for ECE 445, Senior Design, Spring 2013

TA: Justine Fortier

06 February 2013

Project No. 13

Contents

1. Introduction

1.1 Title

Automatic Handshake Contact Info Exchanger

The goal of this project is to produce a wearable device that is capable of transmitting and receiving contact information from other users. This device will improve the way people exchange their contact information with each other - instead of taking time to exchange information and wasting paper with business cards, one could automatically receive and transmit contact information electronically with the shake of a hand. This project is exciting because it would be useful for a lot of people in the business world and might be able to revolutionize the process of business card exchange and personal information exchange in general.

1.2 Objectives

Goals

- To create a device that will be able to wirelessly transmit and receive contact information with a handshake
- To make the device as small as possible and be able to be worn by the user
- To make the product marketable

Functions

- Wirelessly transmit data between two devices on the detection of a handshake
- Data can be uploaded to a personal computer

Features

- Device is completely wireless for the user
- An accelerometer to detect handshake
- 4 modes of operations: Off, Transmit Only, Receive Only, Receive & Transmit
- Information being transmitted includes name, phone number, email, company name, position/title and URL

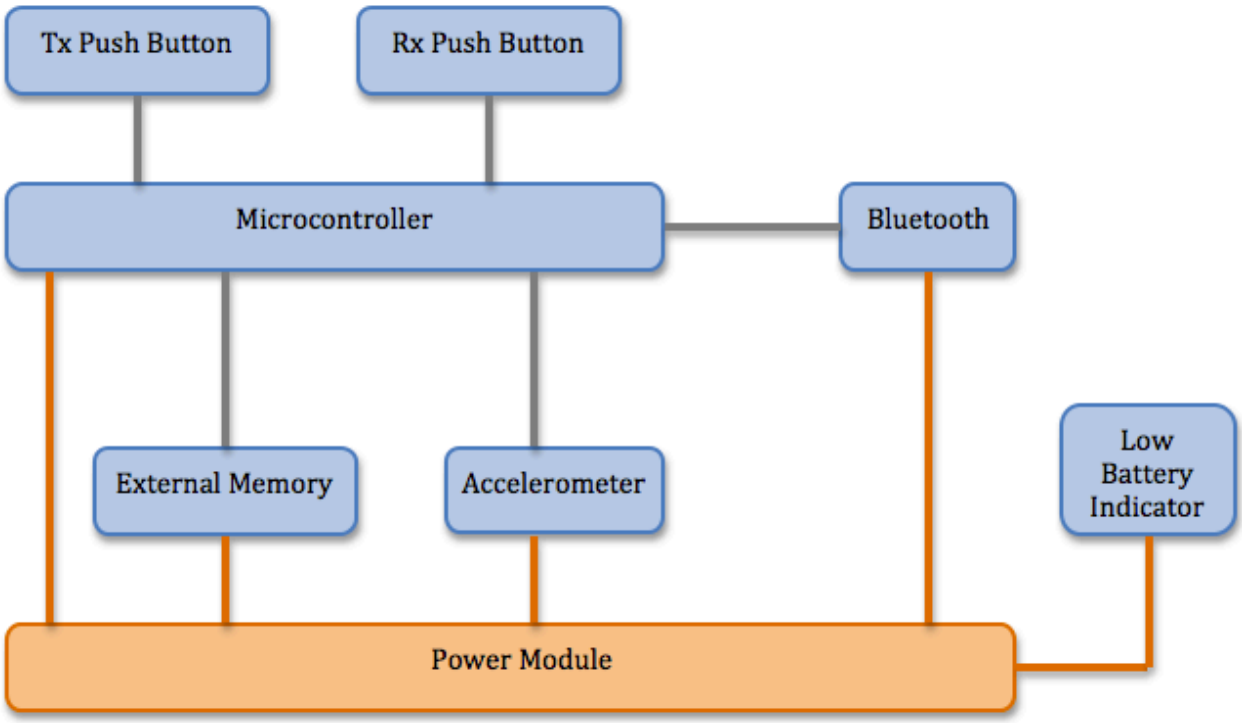
Benefits

- Quick exchange of contact information
- Eliminate the need to print new business cards when information is updated
- Easy organization and access of contact information (reduces the need to carry around huge numbers of business cards; reduces risk of accidentally losing someone's business card)

2. Design

2.1 Block Diagrams

a. Top Level Block Diagram (single device)



___	Power Flow
___	Data Connection

b. Communication Block Diagram

2.2 Block Description

Microcontroller Unit: Arduino Mini Pro

The Arduino microcontroller will be used to control the functionality of the entire device. The Arduino will accept the output of the accelerometer (Analog Input Pin, A_0) and analyze the data in order to detect when a handshake has taken place. The Arduino will also have two inputs from manual push button switches that the user can press. One of these buttons will be pressed to transmit your contact information, and the other will be pressed for receiving the other user's contact information on a handshake. The Arduino microcontroller will be used to write data that it receives from the Bluetooth to the external memory of the device. It will also be responsible for controlling the flow of data throughout the device. Prototyping and testing will be done with an Arduino Uno and the actual device will use an Arduino Mini Pro. These two microcontrollers are almost functionally equivalent besides the Mini Pro operating at 8 MHz instead of the Uno's 16 MHz. Both use the ATmega328P chip as the actual microcontroller.

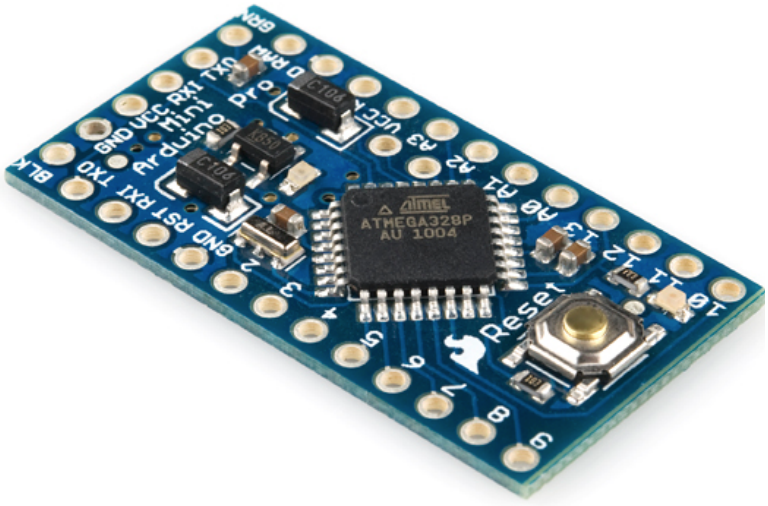


Figure 1: Arduino Pro Mini 328 [1]

Accelerometer: ADXL335 3-axis accelerometer

The accelerometer will be the part of the circuit that detects when a handshake between two users has occurred. It will be able to sense the acceleration of the user's hand in 2 directions while the device is worn. The outputs of the two axes involved in the motion of a handshake (hand extension and shaking) will each send analog data to the Arduino microcontroller (Pins A_0 and A_1). A picture showing the orientation of the

accelerometers axes is shown below in Figure 2. The accelerometer will be powered from the regulated 3.3V output on the Arduino.

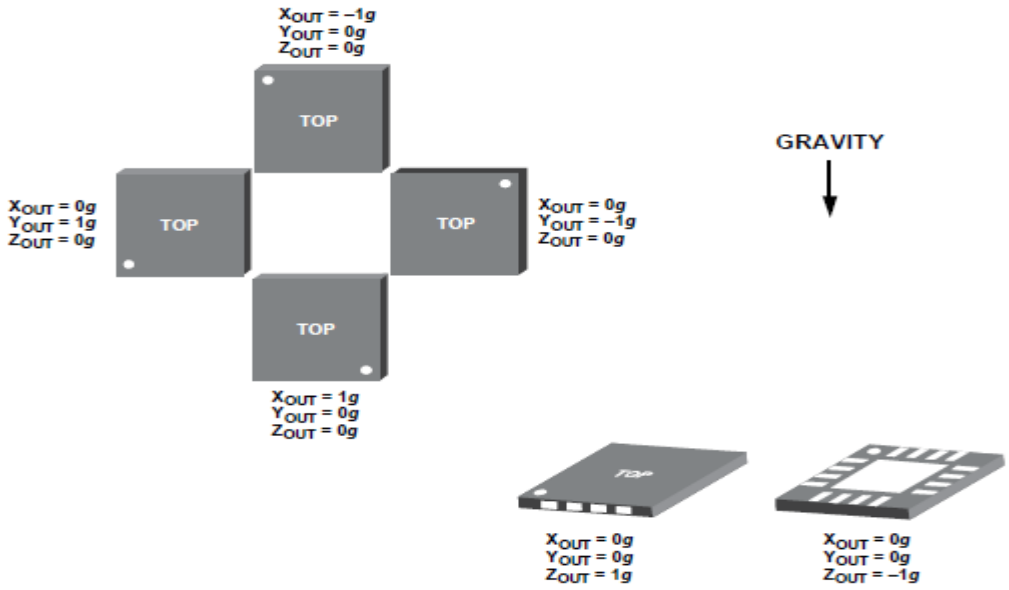


Figure 2: ADXL335 3-axis Accelerometer [2]

External Memory Unit: 24LC256

The external memory will hold the contacts the user receives when data is transferred. The microcontroller will be responsible for writing to this memory when information is received from another device. The data the user sends upon a handshake will be stored on the microcontroller's EEPROM (Electrically Erasable Programmable Read-Only Memory) so the Arduino will not have to read from this memory to transmit data. The size of the memory is 32kBytes, which is enough storage to hold over 100 business cards (see Data Requirement Calculations). The information will be able to be read off of this chip by using the Arduino's serial communication port to display it on a computer monitor.

Manual Switches: Push Button Switches

The circuit design will include two manual switches that the user can press if he wants to transmit/receive/or swap contact info with other users. The outputs of these sensors will either pass V_{cc} or GND to inputs on the microcontroller.

➔ Tx (transmitter) Push Button Switch:

When this switch is pressed down by the user, upon the detection of a handshake the users uploaded data will be transferred to the other device.

➔ Rx (receiver) Push Button Switch:

When this switch is pressed down by the user, upon the detection of a handshake the user will receive data from the other device and read it into the external memory.

The table below summarizes the outcome of communication depending on which switch(es) are pressed:

Tx Button	Rx Button	Action Upon Handshake
Not Pushed	Not Pushed	Do Nothing
Not Pushed	Pushed	Receive data from other device
Pushed	Not Pushed	Transmit data to other device
Pushed	Pushed	Swap data

Bluetooth Mate Silver- RN-42 Module

The Bluetooth transceiver will be used to send and receive the data being transmitted wirelessly between the two devices. This device is low power because it only requires 26 μA in sleep mode while still being discoverable and connectable. [7] The communication range between two devices will always be between about 0.2-5 meters so the data will be transferred reliably because it is well within the range of 20 meters. The Bluetooth Tx will be connected to the microcontroller's Rx and the microcontroller's Tx will be connected to the Bluetooth's Rx. Since the files being transmitted are generally small, duration of communication time will also be low. The transceiver will use the microcontroller as a host to process the data.

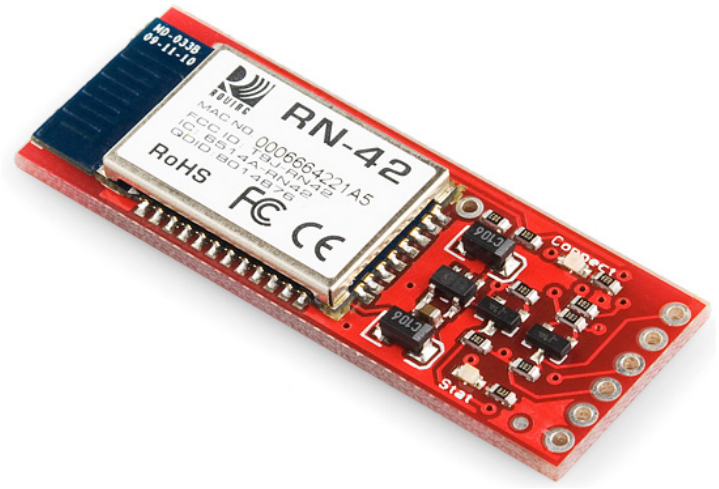


Figure 3: Bluetooth Mate Silver (RN - 42 Module) [3]

Power Supply

The power source for the circuit components will be two SONY CR2477 3V button cell batteries connected in series (Figure 5). The total capacity of these batteries is 1000mAh, which is a reasonable lifetime for our device (see Power Calculations). The power requirement for the components in this circuit will require 3.5-12V or a regulated 3.3V source. The regulated 3.3V source can be obtained from the V_{cc} pins of the Arduino that output a regulated 3.3V.

Battery Life Indicator

The circuit will include a battery life indicator that is designed to detect when the power supply batteries are running low. When the hardware detects a power supply voltage of about 4.5V it will light an LED informing the user that they need to change the batteries soon. The warning cutoff is set at 4.5V because referencing the data sheet [8], the individual 3V batteries start steeply discharging at ~2.25V.

3. Schematics and Flow Chart

3.1 Schematics

- Overall Circuit Schematic for one device

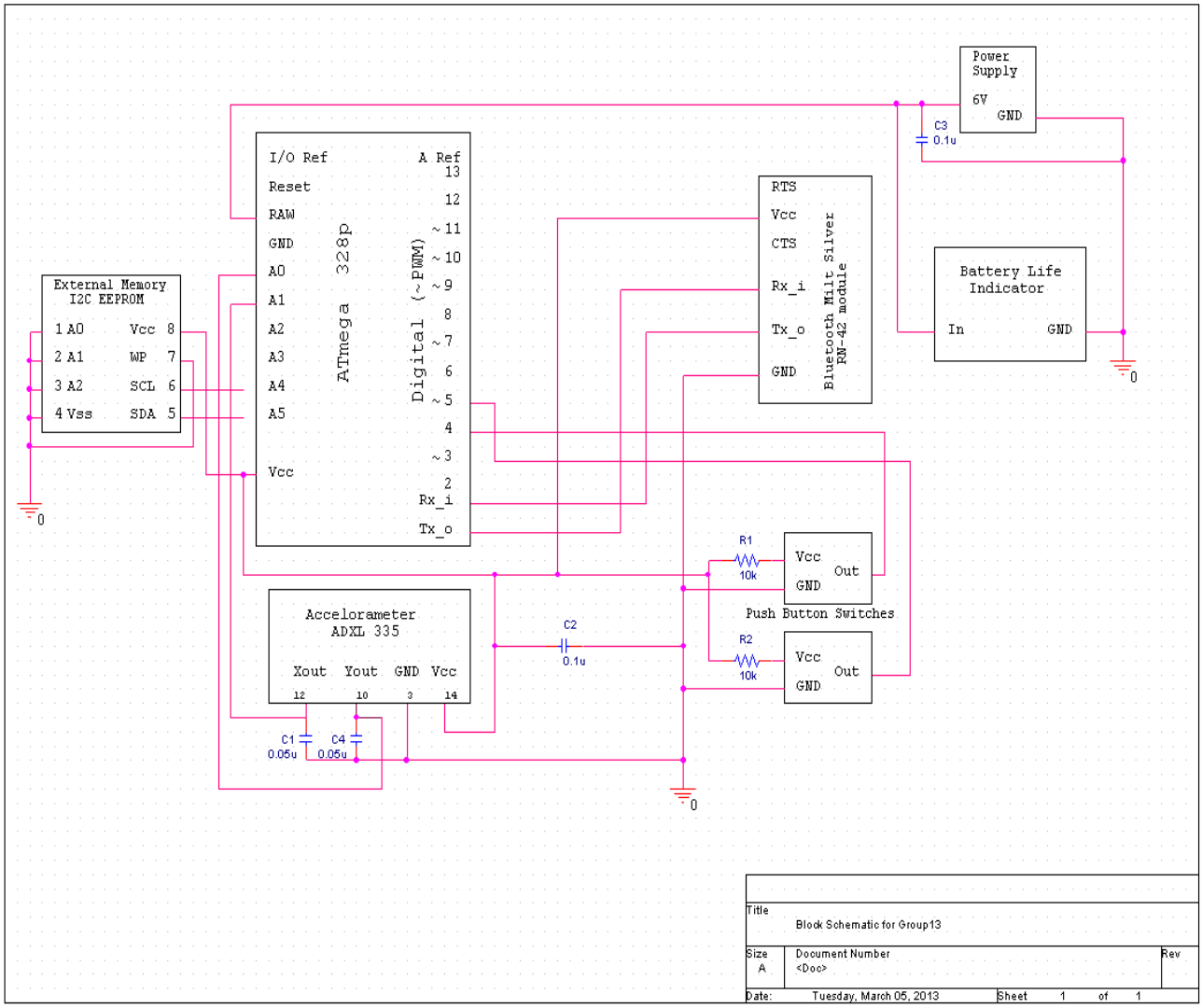


Figure 4: Overall Circuit Schematic

- Arduino to ATmega 328 Pin Mapping

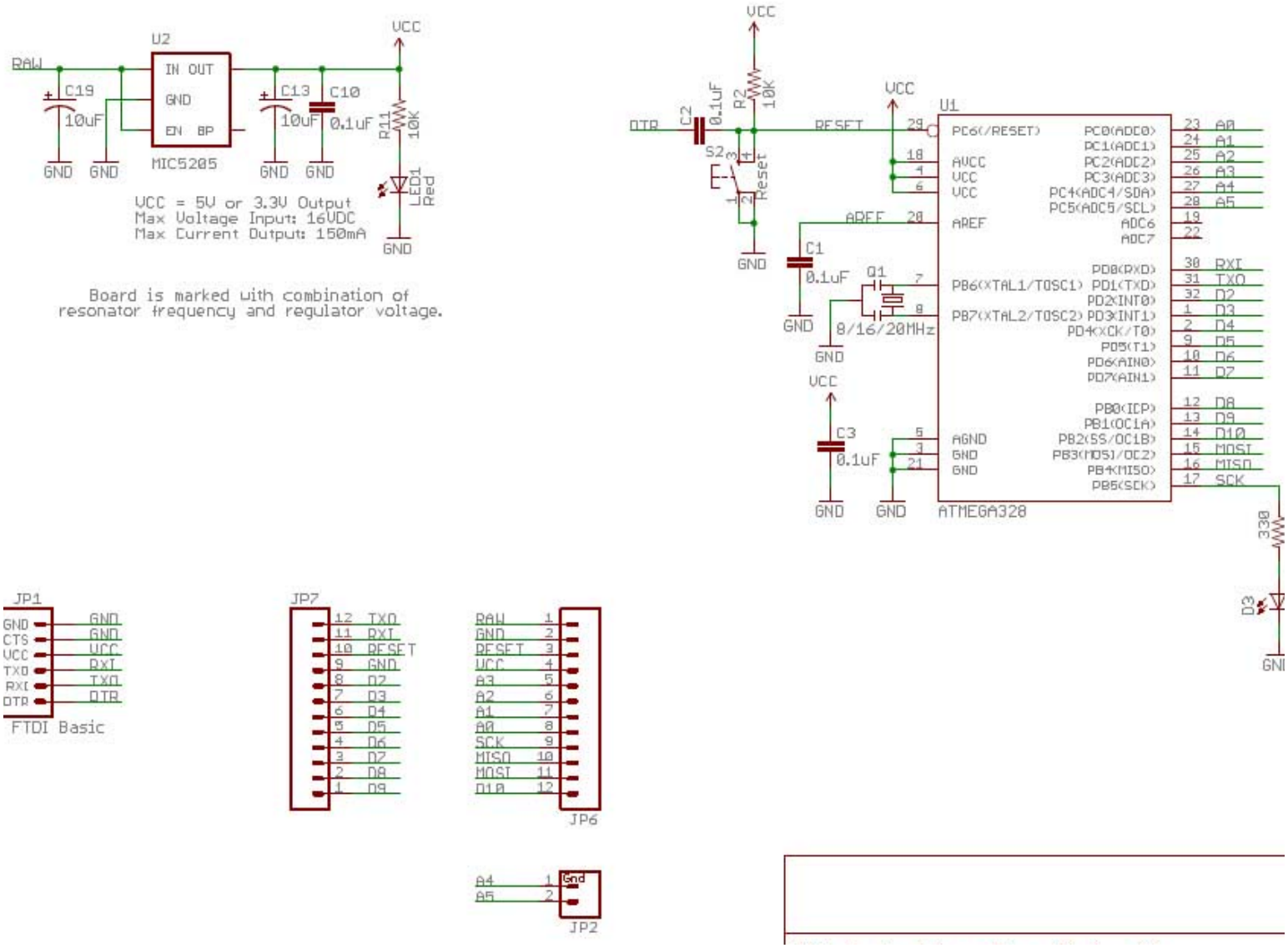


Figure 5: Arduino Pro Mini to ATmega328 Pin Mapping [9]

➢ Power Supply Schematic

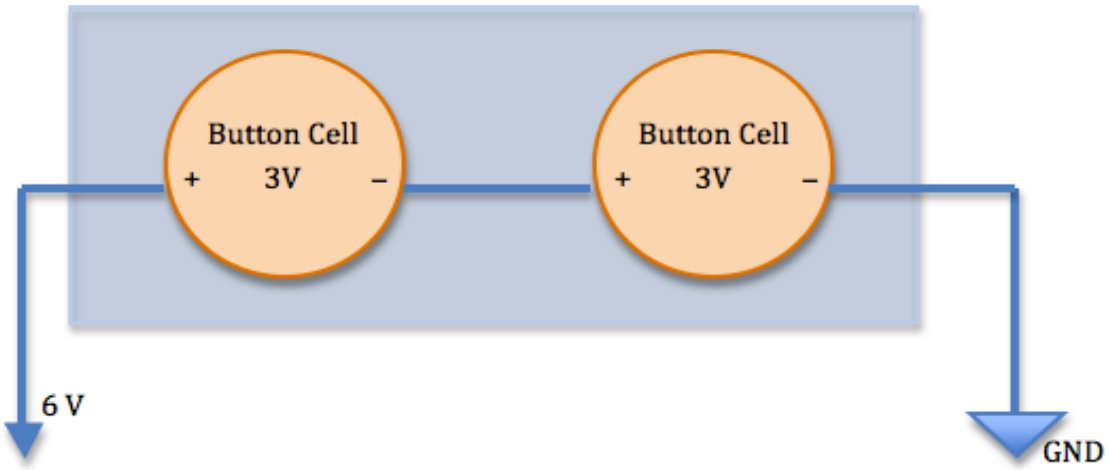


Figure 6: Power Supply Schematic

➢ Battery Indicator Schematic

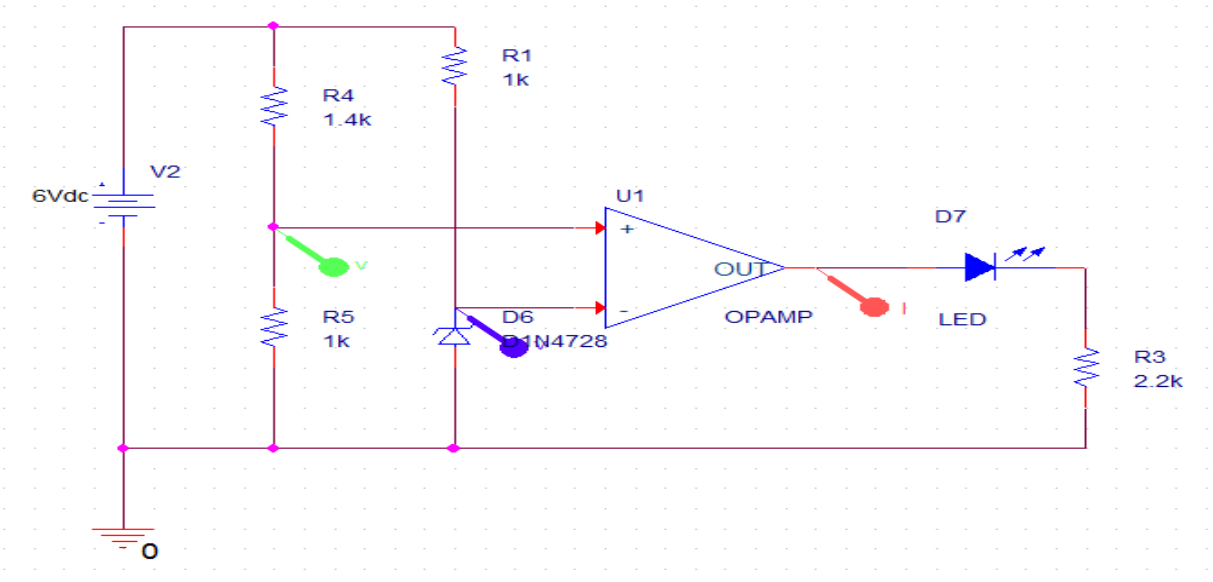


Figure 7: Battery Life Indicator Schematic

3.2 Overall System Flowchart

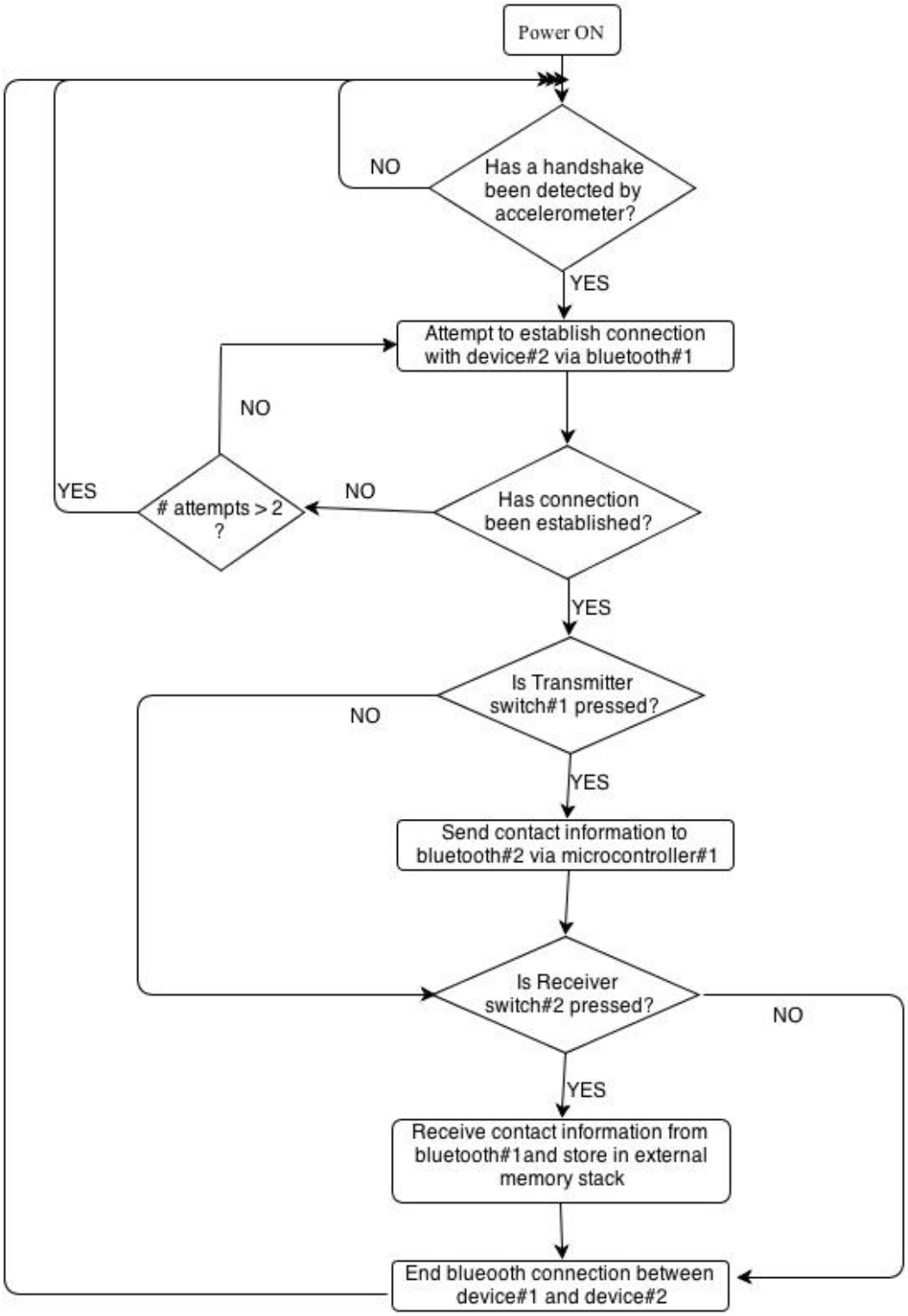


Figure 8: System Flowchart

4. Calculations

4.1 Power Requirement Calculations

To calculate the battery lifetime of our power supply we will need to know the power consumption of our circuit. The series combination of our two 3V button cells will have a capacity of 1000mAh [8]. In our unit, the following components will draw power from the batteries in the power module:

- The Arduino Pro Mini
- The Bluetooth Mate Silver
- The Accelerometer
- The external memory

The 3.3V, 8 MHz Arduino Pro Mini will require about 4mA of current [4]. To be conservative, for our estimations we will say that the Arduino consumes a current of about 7 mA.

The accelerometer, ADXL335 will require about 350 µA of current [2].

The external memory unit, in its standby state, typically requires a current of about 0.1 µA, at our operating voltage. During the time that we are writing to the memory it will require about 400 µA [5].

The Bluetooth Mate Silver has 3 different operating modes. It requires 26 µA in its sleep state, 3 mA when connected, and 30 mA for transmitting data [7].

Device	I_{Sleep}	I_{Active}	$I_{Transmitting}$
Arduino Pro Mini	-	7 mA	-
Bluetooth Mate Silver	26 µA	3 mA	30 mA
Accelerometer	-	350 µA	-
External Memory	0.1 µA	400 µA	-

For estimating the current required by the Bluetooth Module we will assume that:

- 95% of the time it will be sleeping (when a handshake is not detected)
- 3% of the time it will be connected
- 2% of the time it will be transmitting

$$I_{Bluetooth} = 0.95*(26\ \mu A) + 0.03*(3\ mA) + 0.02*(30mA) \approx 715\ \mu A$$

For estimating the current the external memory will require we will assume that:

- 98% of the time it will be in sleep mode
- 2% of the time it will be written to

$$I_{Memory} = 0.98*(0.1\ \mu A) + 0.02*(400\ \mu A) \approx 8.098\ \mu A$$

The total amount of current can be expressed as:

$$I_{total} = \sum I_{Components}$$

$$I_{total} \approx 8.073 \text{ mA}$$

The battery lifetime (τ) for our circuit can be expressed as:

$$\tau = \frac{Battery\ Capacity\ (mA\hbar)}{I_{total}\ (mA)}$$

Therefore,

$$\tau = \frac{1000}{8.073} \approx \mathbf{123.87\ \boldsymbol{hours}}.$$

4.2 Data Requirement Calculations

We created some basic test files using Notepad. Since notepad allows us to code only available ASCII characters, these files will be the most basic of their kind and each text character will require 1 byte of data.

In order to estimate how many bytes a typical file will require, we created 3 simple files in Notepad containing the typical information that constitute a business card.

File 1	225 bytes
File 2	196 bytes
File 3	205 bytes

From these results, we can assume that a typical business card will be approximately 200 bytes.

The external memory that we are using, has a storage capacity of 32 kB. The number of files that can be stored on this device can be estimated as follows:

$$\#\ files \approx \frac{32{,}000\ bytes}{200\ bytes} = \mathbf{160\ files}$$

5. Requirements and Verification

5.1 Requirements & Verifications Table

REQUIREMENTS	VERIFICATIONS
Power Supply	
1. The button cell batteries need to provide a stable voltage of to the Arduino board a) The batteries must provide the Arduino RAW input pin with a voltage between 3.5-12V, which it can regulate to 3.3V. 2. The Arduino's on-board voltage regulator needs to output a stable voltage from its V_{CC} pin a) A regulated 3.3V must be output from the Arduino V_{CC} pin to the other circuit components	1. Measure the voltage across the battery using a multimeter. a) We expect the power supply to input 6 ± 0.5V when the batteries are fully charged. 2. Measure the voltage out of the Arduino V_{CC} pin using a multimeter. a) We expect the multimeter to read 3.3 ± 0.2V while the input voltage on the Arduino RAW pin is between 3.5-12V.
Manual Switches	
1. Should output a logic LOW (0 ± 0.3V) when not pressed. 2. Should output a logic HIGH of (3.3 ± 0.5V) when pressed.	1. Connect terminals V_{cc} and GND to 3.3V supply and ground respectively. Using a multimeter, measure the output voltage when switch is not pressed. 2. Using a multimeter, measure the output voltage when switch is pressed.

Accelerometer	
1. Functions properly when powered by regulated 3.3V output from Arduino a) The X and Y accelerometer outputs should sense the acceleration due to gravity when the chip is held stationary on their respective axes. b) The X and Y outputs of the accelerometer should sense acceleration due to motion on their respective axes. 2. Should correctly output data at a sufficient number of measurements per second (100 Hz), in order to detect a handshake from the user a) The capacitor attached between the accelerometer's X_{OUT} or Y_{OUT} and GND should set the sample frequency to 100 Hz ± 25%. [2]	1. Use a multimeter to verify that the voltage on the accelerometer's V_{CC} pin is in the required range of 1.8-3.6V a) While powered, connect the X output to the oscilloscope and orient the accelerometer's X-axis parallel to gravity. Verify that the oscilloscope is reading a set value when the accelerometer is held stationary. Do the same for the Y output. b) Move the accelerometer in the X and Y directions while viewing their outputs on the oscilloscope. Their outputs should oscillate around their offset (from gravity) corresponding to motion on their axes. 2. Connect sensor output pin to A_0/A_1 on the Arduino and write software to print the data the Arduino is receiving from the accelerometer on the computer monitor. a) If the frequency of samples is too low then decrease the value of the capacitor between X_{OUT} or Y_{OUT} and GND until the samples are being output at 100 Hz ± 25%.
Microcontroller	
1. The ATmega328P chip should function appropriately when the voltage supply on the Arduino's RAW pin is in the range 3.5V – 12V. a) The logic high output should be	1. Supply RAW pin with a voltage of 6V. Verify the Arduino chip with a simple test code setting one digital output HIGH and another one LOW. a) Connect multimeter to the output

3.3V ± 0.5V with a logic low of 0V ± 0.5V.	of the chip and verify that the output high is 3.3V ± 0.5V and output low is 0V ± 0.5V.
2. Required to correctly detect when a handshake has occurred by using mock data from it's Analog Input pins A_0 and A_1	2. Write a software algorithm to process the mock data. Hook up a simple resistor and LED between an Arduino digital output pin and GND. Turn the LED on when the algorithm detects a handshake from the user.
a) The Arduino must not detect a "false handshake" and initiate unwanted data transfer from this mock input data	a) Test this algorithm by inputing data created to simulate the accelerometer data when a user is wearing it. Verify the LED only turns on during a handshake.
3. Microcontroller should complete communications based on which manual push button switches are pressed	3. Tie the switch inputs to the Arduino to either HIGH or LOW and have the Arduino detect a mock handshake to initiate communication
a) If the TRANSMIT switch is pressed then data should be sent to the serial input of the Bluetooth	a) When the TRANSMIT switch is pressed view the microcontroller output to the Bluetooth on the computer and verify that it matches the mock data to be sent
b) If the RECEIVE switch is pressed the microcontroller should take data input from the Bluetooth and send it to the external memory	b) When the RECEIVE switch is pressed view the microcontroller output to the external memory on the computer screen and verify that it matches the mock data input from the Bluetooth
4. After data transfer between two devices is finished end connection between Bluetooth modules.	4. Check that after communication between devices is over the blinking LED on board the Bluetooth module turns off, signaling the connection is over.

Bluetooth	
1. The Bluetooth Mate Silver should function properly when supplied with the regulated 3.3V from the Arduino supply. 2. The Bluetooth modules should be able to communicate without error when located within a range of 0.2-5m. a) Bluetooth Module can correctly and efficiently (<2s) transmit data from these distances b) Bluetooth Module can correctly and efficiently (<2s) receive data from these distances	1. Use a multimeter to measure the voltage on the Bluetooth's V_{CC} pin and verify that it is between 3.3-6V. 2. With Bluetooth Module kept 0.2m, 1m, 3m, and 5m away from the computer's Bluetooth, establish connection between the two modules: a) Transmit 200 bytes of mock data from Bluetooth module to the computer's Bluetooth. Verify that the correct data is transmitted at each distance by viewing the received data on the computer screen. b) Receive 200 bytes of mock data from the computer's Bluetooth to the Bluetooth Module. Verify that the correct data is received at each distance by viewing the received data on the computer screen.
External Memory	
1. The external memory should function properly when powered with regulated 3.3V output from Arduino. a) When preforming a write operation the data should be stored in the correct address which is specified by the address byte	1. Use a multimeter to test that the input voltage is within the required range (2.7-5.5V) a) Testing code will send known random data to be written to specified external memory addresses

b) When reading from a memory address the correct data should be printed to the computer screen	b) Will use software to read data contents from these addresses and display it on the computer monitor to verify that the memory read and write operations are functioning properly.
Battery Life Indicator	
1. The Battery Life Indicator should properly detect when the voltage on its input falls below the reference threshold a) The red LED on the output of the Battery Life Indicator should turn on when the batteries are low (~15% charge remaining)	1. Provide the Battery Life Indicator circuit with a range of sample input voltages and test the output of the red LED a) Use a multimeter to sweep the battery input voltage from 6V to 2V in increments of 0.1V. Verify that the red LED turns on when the input voltage is about 4.5V.

5.2 Tolerance Analysis

In order for our device to function properly it is essential for our accelerometer unit to properly output data on each axis that is sending data to the microcontroller for handshake detection. If the accelerometer fails to output data at a correct frequency, < 75 Hz (25% tolerance), the peak acceleration can be missed and the user's handshake might not be detected. The accelerometer is also required to output the correct value of acceleration on each axis with maximum 10% error. If this error is any larger than 10%, then data transfer could be initiated when no handshake happens or it could fail to detect a handshake that did happen. We will test that it is correctly measuring acceleration values due to gravity within 10% by viewing the output of each axis on the oscilloscope. When accelerated in certain directions, proper output should also be observed with a maximum 10% error.

6. Cost Analysis & Schedule

6.1 Cost Estimate

Parts

Table 1: Parts Costs

Part	Estimated Cost/Unit ($)	Quantity	Total Cost ($)
ADXL335 Accelerometer	24.95	2	49.90
Microcontroller Arduino Mini Pro	18.95	2	37.90
Bluetooth Mate Silver	39.95	2	79.90
External Memory	1.95	2	3.90
Power Supply (Lithium 3V button cells)	4.50	4	18.00
LED	0.35	2	0.70
Zener Diode	0.25	2	0.50
Op Amp	0.40	2	0.80
Resistors	0.05	14	0.70
Capacitors	0.10	6	0.60
Total			**$ 192.90**

Production Cost

Table 2: Production Cost for 1 Device

Part	Bulk Price ($)	Quantity	Cost/unit ($)
ADXL335 Accelerometer	7.96	1	7.96
Microcontroller Arduino Mini Pro	15.96	1	15.96
RN-42 Bluetooth	15.95	1	15.95
External Memory	1.56	1	1.56
Power Supply (Lithium 3V button cells)	2.49	2	4.98
LED	0.28	1	0.28
Zener Diode	0.20	1	0.20
Op Amp	0.30	1	0.30
Resistors	0.05	7	0.35
Capacitors	0.10	3	0.30
Total			**$ 47.84**

Labor

Table 2: Labor Costs				
Name	**Rate/hour**	**Overhead (x 2.5)**	**Hours ***	**Total ($)**
W. Hanley	35	87.5	180	15,750
K. Samigollayev	35	87.5	180	15,750
A. Saha	35	87.5	180	15,750
Total				**$ 47,250**

* Assuming a 15 hour work week for 12 weeks

Grand Total

Table 3: Total Costs	
Section	**Total ($)**
Parts	192.90
Labor	47,250.00
Total	**$ 47,442.90**

6.2 Schedule

Week	Task	Person in-charge
February 04	Research and think about handshake detection methods	Kuanysh
	Finalizing proposal	Ambieca
	Research circuit components / decide on parts to use	William
	NOTE: proposal due, mock DR sign-up	Ambieca
February 11	Design microcontroller unit	William
	Design components for the rest of the circuit (Accelerometer, Bluetooth, External Memory)	Kuanysh
	Ordering required parts	Ambieca
February 18	Research and design power supply that will be best for our device	Ambieca
	Come up with software flow chart for entire device	William
	Start initial battery capacity and data requirement calculations	Kuanysh
	NOTE: DR sign-up	
February 25	Write and organize Design Review Document	Ambieca
	Learn to program microcontroller unit	William
	Create overall circuit schematic for device	Kuanysh
	NOTE: Design Reviews	
March 04	Begin programming microcontroller unit; work on handshake detection with Kuan	William
	Accelerometer testing and verification; handshake detection	Kuanysh
	Learn how to make PCB layout & footprints in Eagle; design and build Battery Life Indicator circuit	Ambieca

March 11	Create PCB footprints for each hand device	Ambieca
	Interface Bluetooth Mate with microcontroller; test for proper communication between modules	William
	Testing and verification of external memory unit and power supply	Kuanysh
	*NOTE: Individual progress reports due	
March 18	SPRING BREAK	
March 25	Put overall circuit together; make sure all components of the circuit are working with each other correctly	Kuanysh
	Submit PCB for manufacture; Prepare for mock-up demo / presentation	Ambieca
	Write Arduino code to write received data onto external memory upon a handshake	William
	*NOTE: Mock-up demo, mock-up presentation sign-up	
April 1	Learn to solder components onto PCB; Revise PCB	Ambieca
	Interface transferred data from memory to the computer	William
	Debug overall circuit issues; help with interfacing data to computer	Kuanysh
	NOTE: Mock-up presentation, last day for first revision of PCB	
April 8	Revise and resubmit PCB	Ambieca
	Complete assembly of device	William
	Help with PCB board revisions and device assembly	Kuanysh
	NOTE: last day for final revision of PCB	
April 15	Document final paper / Make device wearable	Ambieca
	Overall Project Debugging	Kuanysh
	Start preparation for final demo and presentation	William
	NOTE: demo sign-up & presentation sign-up	

April 22	Prepare presentation	William
	Final Paper work	Ambieca
	Prepare and assemble final demo	Kuanysh
	NOTE: demos and presentations	
April 29	Presentation preparation	William
	Final paper editing & finishing touches	Ambieca
	Finishing touches on overall project	Kuanysh
	NOTE: final paper due, lab notebook due, checkout	

7. Ethical Considerations

The purpose of the *Automatic Handshake Info Exchanger* is to simplify and enhance the user's ability to exchange personal contact information. We strive to complete this project bearing in mind the IEEE Code of Ethics in Section 7.8 of IEEE Policies [6]. The most relevant policies pertaining to our project are detailed as follows:

3. *To be honest and realistic in stating claims or estimates based on available data;*

Throughout the development of the device, we will follow the 3rd code closely, and only make claims and estimates based on real data acquired from our design. We will be honest and will not falsify the data acquired from our test procedures.

5. *To improve the understanding of technology; its appropriate application, and potential consequences*;
6. *To maintain and improve our technical competence and to undertake technological tasks for others only if qualified by training or experience, or after full disclosure of pertinent limitations*;

After this project, we will have learned a great deal about various real-world technologies such as Bluetooth, the ATMega328p microcontroller, external Memory, motion sensors, accelerometers, gyroscopes. This will improve our understanding of these technologies and their applications, and also improve our technical competence, as directed in the 5th and 6th codes of the IEEE Code of Ethics.

7. *To seek, accept, and offer honest criticism of technical work, to acknowledge and correct errors, and to credit properly the contributions of others*;
10. *To assist colleagues and co-workers in their professional development and to support them in following this code of ethics*.

Furthermore, while working on the project, we will build an environment that promotes engineer professionalism, which welcomes constructive and honest criticisms, acknowledges errors, assists peer workers with their professional and academic developments, and credits appropriate contributions, as cited in the 7th and 10th codes of the IEEE Code of Ethics.

9. *To avoid injuring others, their property, reputation, or employment by false or malicious action*;

We will make sure that the data transmitted through this device will offer the users with data that is as accurate as possible, and will not provide them with false information, in order to avoid damaging.

In addition to IEEE Code of Ethics, we will also ensure that privacy of the users is protected and that the device is transmitting the authorized data only.

8. Safety Issues

1. Although working with low levels of voltage and current we will take necessary precautions when working in the lab around our equipment as well as other group's equipment.
2. We will ensure that there are no exposed wires on the device, which could be potentially dangerous for the user.
3. Because this device will be worn on hand, we will ensure that there are no rough edges that can cause any physical harm to the user.
4. We will ensure that the components used in our circuit will not become too hot so that it is not a hazard to the user's skin.

9. References

1. Sparkfun Electronics. (n.d.). *Arduino pro mini 328 - 5v/16mhz*. Retrieved from https://www.sparkfun.com/products/9218

2. Analog Devices. (n.d.). *"Small, low power, 3-axis ±3 g accelerometer" ADXL335 datasheet*. Retrieved from http://www.analog.com/static/imported-files/data_sheets/ADXL335.pdf

3. Sparkfun Electronics. (n.d.). *Bluetooth mate silver*. Retrieved from https://www.sparkfun.com/products/10393

4. Talbott, R. (2009, February 25). Re: Power consumption [Online forum comment]. Retrieved from http://forums.adafruit.com/viewtopic.php?f=25&t=9638

5. Microchip. (n.d.). *"256k I2C™ CMOS Serial EEProm" 24LC256 datasheet*. Retrieved from http://ww1.microchip.com/downloads/en/DeviceDoc/21203S.pdf

6. Institute of Electrical and Electronics Engineers, Inc. (2012). *IEEE Policies*. Retrieved from http://www.ieee.org/documents/ieee_policies.pdf

7. Roving Networks. (n.d.). *"Class 2 Bluetooth® module" RN-42-N data sheet*. Retrieved from http://dlnmh9ip6v2uc.cloudfront.net/datasheets/Wireless/Bluetooth/Bluetooth-RN-42-DS.pdf

8. MicroBattery. (n.d.). *: Lithium Manganese Diode Battery" CR2477 datasheet*. Retrieved from http://www.sony.net/Products/MicroBattery/cr/pdf/cr2477.pdf

9. (n.d.). Retrieved from http://www.arduino-freeduino.de/wp-content/uploads/2011/05/Arduino_Pro_Mini_schematic.jpg