Investigating an Accelerometer to Produce a Step Measurement Glove

Research Question: *How do accelerometers detect motion through algebraic manipulation of physics concepts and equations and how do they transmit that information into computational data. How do we represent this using 3-D vectors on a plane & how do we communicate with the accelerometer using binary.*

Subject: Mathematics HL

Word count: 3,427

Table of Contents

1- Introduction:

From the phone that each of us holds in their pockets, to the breaking innovation of the Nintendo Wii controllers and even virtual reality headsets, accelerometers have made game-changing advancements in our everyday lives and we might not even notice it. Accelerometers have also made their way in helping within medical fields as it helps with studies and research that involve physical activities, which in our case will help us measure the number of steps someone takes.

I have always been interested in engineering and its integration of math, physics, and computer science in order to solve problems through its unique solutions. This is why I utilized the components I had available and the skills I learned through the IB program to build a glove that measures a person's steps. Most smartphones and smartwatches use this functionality to tell their users how many steps they took throughout the day. Many training instructors use this piece of technology to keep track of their clients health. So due to my curiosity within the field of engineering and being an athletic student, I pursued this project. In this project, I will engineer and program a glove capable of counting the number of steps a person makes.

Before we build our glove however we must analyze and investigate the devices used and their components. Accelerometers help us determine changes in movement in the x, y & z planes, which can be represented as a 3-D vector and they do this by using capacitors and Newton's second law of motion and manipulating both formulas algebraically. Moreover, data is transmitted using binary which can be represented using a geometric series, from which we can determine numerical values for each byte.

2- Accelerometers and MEMS

2.1 - Measuring the acceleration of an object in a spring, mass system

A force(measured in Newtons, N) is a push or a pull on a mass system forcing that object to accelerate in the direction. The acceleration due to gravity pulls an object with mass towards the center of gravity with a force equivalent to,

$$F = ma$$

A spring system can also apply a force. That force depends on the stiffness of the spring. To measure the force the spring applies on a mass system we use the following equation.

$$F = -k\Delta x$$

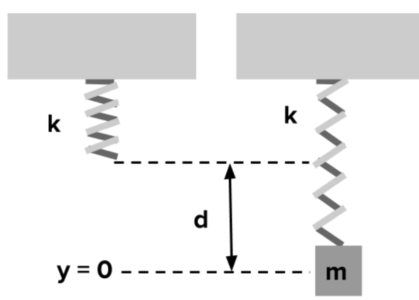


Figure 1- Effect and displacement of spring with and without a mass

where k is the spring constant and Δx is the displacement of the object from the origin of the spring to the place where it rests. The negative sign just tells us that the force applied is opposite to the direction of the displacement. Equating the two equations you get,

$$ma = -k\Delta x$$

To measure the acceleration we divide both sides by m and get

$$a = -\frac{k\Delta x}{m}$$

where k and m are known constants. But to measure acceleration we need to figure out the displacement, and we need to measure this displacement using voltage readings. This can be done with microelectromechanical systems(abbreviated to MEMS).

2.2 MEMS

MEMS are used in our everyday lives and we don't even notice it, it helps us build mechanical and electrical systems to transmit information on a microscale. Systems such as micro resistors, gyroscopes, and our focus in this essay, accelerometers.

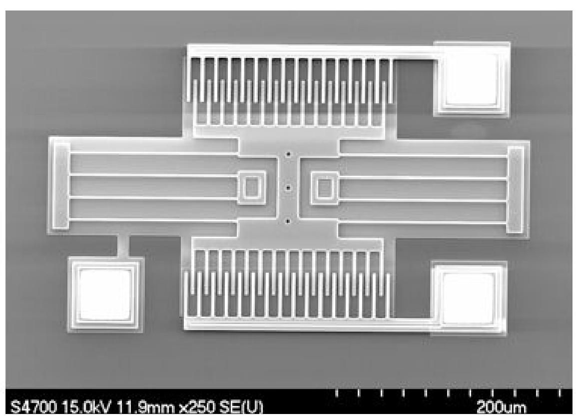


Figure 2- A MEMS system. This device is used as a microsensor ("What Is MEMS Technology?")

MEMS is a growing field in engineering that has multiple applications and billions of dollars in investment from large corporations. These systems are so small and sensitive that they are only researched and produced inside a special room known as a clean room which filters out any particles that could destroy the system. So how are mechanical and electrical systems used in order to create an accelerometer? We discussed the mechanical aspect in 2.1, the electrical component will require the use of capacitors.

2.3 Capacitors

Capacitors are electronic systems that involve two charged plates each with an area of A, separated by a distance of x. When a potential difference of magnitude V is connected between the two plates, each plate becomes charged by a value of q, one plate is positive while the other is negative. The capacitance of a capacitor(measured if farads, F) is defined by the charge on the plates per unit volume which can be found by the following equation

$$C = \frac{q}{\Delta V}$$

Since the area of the plates increase the capacitance we can say that the following relation is true

$$C \ \alpha \ A$$

and since the distance between the plates decreases the capacitance we can say that the following relation is true

$$C \;\alpha\; \frac{1}{x}$$

between the two charged places is a medium (air, vacuum, etc) which is represented by a constant known as the permittivity of the medium, given a value of ε. So another way of calculating the capacitance of an object is by using the following formula

$$C = \varepsilon \frac{A}{x}$$

2.4 Relating the Capacitance and the Force Equations

Our objective is to relate the acceleration to capacitance. To do that we can manipulate the capacitance equations in order to form the following equation.

$$x = \varepsilon \frac{A}{C}$$

However, we must find the change in displacement between the two plates so the change in displacement is represented by the following equation

$$\Delta x = \varepsilon \frac{A}{C_f} - \varepsilon \frac{A}{C_i}$$

$$\Delta x = \varepsilon A \left(\frac{1}{C_f} - \frac{1}{C_i}\right)$$

Where C_f is the value of the capacitance when a force is applied on the mass and C_i is the value of the capacitance when no force is applied, now utilizing and manipulating the following equation

$$a = -\frac{k\Delta x}{m}$$

$$\Delta x = -\frac{am}{k}$$

now we substitute the displacement value into the spring-mass system equation

$$-\frac{am}{k} = \varepsilon A(\frac{1}{C_f} - \frac{1}{C_i})$$

$$a =- \frac{k\varepsilon A}{m}(\frac{1}{C_f} - \frac{1}{C_i})$$

$$a = \frac{k\varepsilon A}{m}(\frac{1}{C_i} - \frac{1}{C_f})$$

We can use this equation by finding the value of C_i, or we could represent C_i in the form of the initial displacement

$$a = \frac{k\varepsilon A}{m}(\frac{x_i}{\varepsilon A} - \frac{1}{C_f})$$

$$a = \frac{kx_i}{m} - \frac{k\varepsilon A}{mC_f}$$

where the values of A,ε,m and C_i/x_i are known. Calculating the acceleration is the whole premise of an accelerometer. The capacitance of an object could be found using microcontrollers in order to calculate the acceleration.

2.5 An accelerometers MEMS system

We discussed and analyzed the relationship between the change in capacitance and the acceleration. But how does this look physically on a MEMS system

An accelerometer involves a fixed system with a plate and a movable internal assembly that holds the second plate and a mass system, the movable internal assembly is connected to a fixed plate with springs. As follows

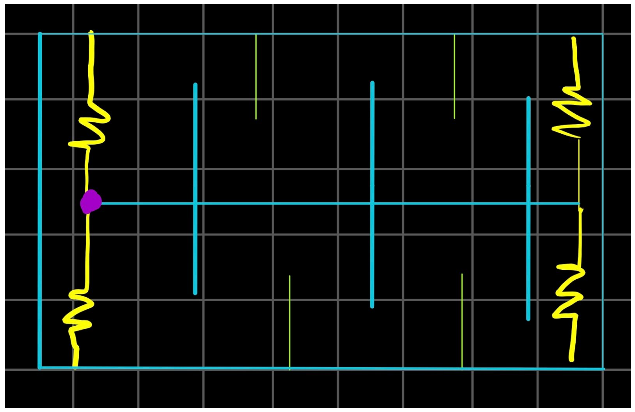


Figure 3- An accelerometers spring-mass system

and a labeled figure

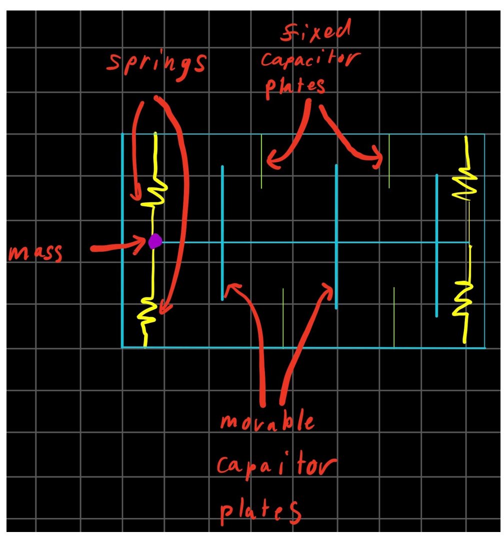


Figure 4- A Labeled figure of the accelerometers spring-mass system

So every time the mass experiences a force due to gravity it pulls the system down and the distance between the plates increases but the distance between the plates towards the mass system decreases, and measuring this capacitance helps us determine the acceleration.

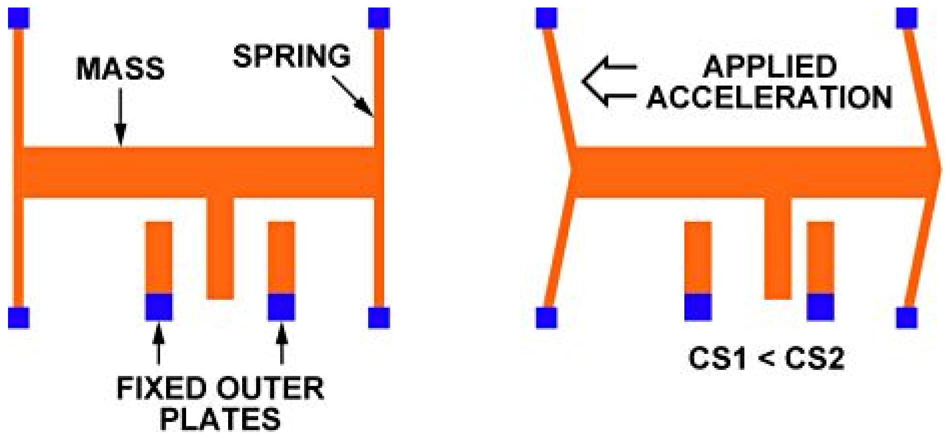


Figure 6 - The spring-mass system at rest and when a force is applied (Khenkin)

Increased capacitance means an acceleration in the direction of that capacitor, giving us a vector direction to work with. There are three of these MEMS systems within a capacitor, one on each axis

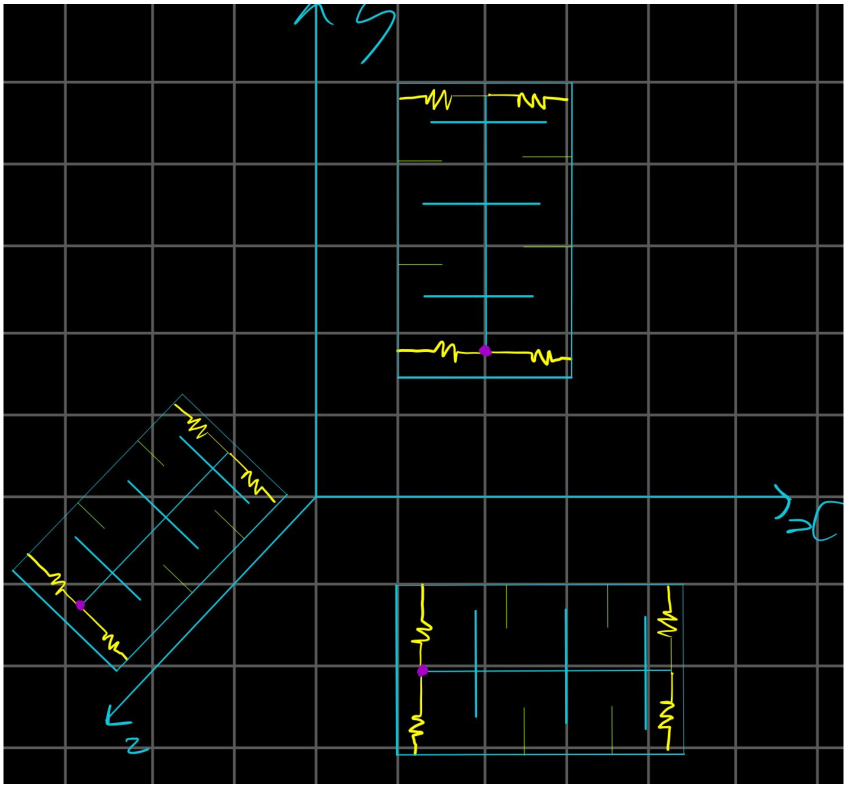


Figure 7- Three spring-mass systems, each coincident to specific axes to make an accelerometer

so if the force of gravity acts in one of these planes, that force of gravity will pull the mass and the capacitors with it displacing it by a factor of Δx and measuring the capacitance give us the acceleration value on each axis, we will represent these values using vectors. However, how do we determine the acceleration on each axis? More importantly, how do we represent this mathematically using vectors?

<u>3 - Representing models on a 3-D plane</u>

3.1- The force of gravity

The force of gravity acts from the center of mass towards the center of gravity (in this case the center of the earth). We can identify the force of gravity with a vector equation.

$$F = m\,(0\widehat{i} + 0\widehat{j} + 9.81\widehat{k})$$

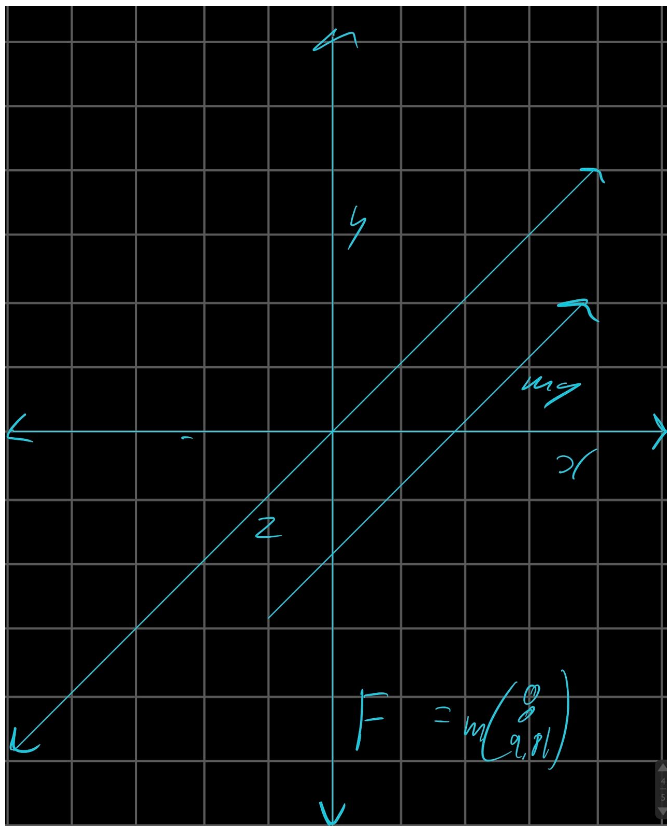


Figure 8- The force of gravity acting on the z-axis and its vector equation

However when we rotate an object, in our case the accelerometer, such that the force of gravity will no longer act in the z-axis, but rather has three-component, an x, y, and z components. The force of gravity only applies when the direction of the force is parallel to an axis

3.2- The Components of a Gravity Vector

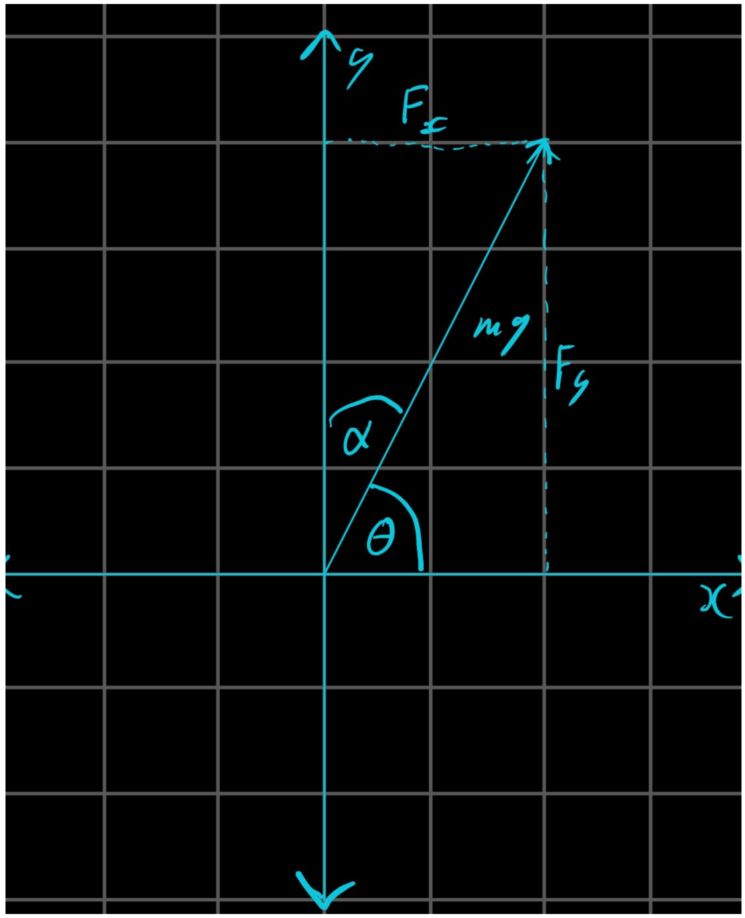


Figure 9- The force of gravity acting on a 2-D plane with x and y components

To help us visualize this, we will start off in a 2-D plane. So in figure 9, the force of gravity has an x- component and has a y-component which can be determined by

$$F_x = mg\ cos(\theta)$$

$$F_y = mg\ cos(\alpha)$$

Where θ and α are the angles between the axis and the force due to gravity in order to determine the component vector parallel to that axis. This concept helps us to represent force vectors on a

3D plane. We can represent the force due to gravity and acceleration due to gravity(g) by the following vector equations

$$F_g = m\,(9.81cos(\alpha)\widehat{i} + 9.81cos(\beta)\widehat{j} + 9.81cos(\gamma)\widehat{k})$$

$$\frac{F_g}{m} = (9.81cos(\alpha)\widehat{i} + 9.81cos(\beta)\widehat{j} + 9.81cos(\gamma)\widehat{k})$$

$$g = (9.81cos(\alpha)\widehat{i} + 9.81cos(\beta)\widehat{j} + 9.81cos(\gamma)\widehat{k})$$

$$9.81ms^{-1} = (9.81cos(\alpha)\widehat{i} + 9.81cos(\beta)\widehat{j} + 9.81cos(\gamma)\widehat{k})$$

But what is each angle, more importantly, how do we determine each angle.

3.3 Angles Between Vectors

Let's assume that there are three separate vector equations coincident to each axis.

$$b_x = (1\widehat{i} + 0\widehat{j} + 0\widehat{k}) \quad b_y = (0\widehat{i} + 1\widehat{j} + 0\widehat{k}) \quad b_z = (0\widehat{i} + 0\widehat{j} + 1\widehat{k})$$

So the angle α is the angle between b_x and vector g, β is the angle between b_y and vector g, and finally, γ is the angle between b_z and vector g. Now to find the value of an angle between two vectors, we will use the following equation for each angle.

$$cos(\theta) = \frac{|a \bullet b|}{|a||b|}$$

Let ‘a’ be the acceleration due to gravity and let ‘b’ be the vector coincident to the axis whose angle you want to find. We also discussed that the magnitude of a is 9.81m/s (later on we'll be using another value) so that gives us

$$cos(\theta) = \frac{|a \bullet b|}{|9.81||b|}$$

$$cos(\theta) = \frac{|a \bullet b|}{|9.81||1|}$$

Now for each angle.

$$cos(\alpha) = \frac{|a \bullet b_x|}{9.81}$$

$$cos(\alpha) = \frac{\left|a \bullet (1\widehat{i} + 0\widehat{j} + 0\widehat{k})\right|}{9.81}$$

$$cos(\alpha) = \frac{\left|a\widehat{i}\right|}{9.81}$$

$$cos(\beta) = \frac{\left|a \bullet b_y\right|}{9.81}$$

$$cos(\beta) = \frac{\left|a \bullet (0\widehat{i} + 1\widehat{j} + 0\widehat{k})\right|}{9.81}$$

$$cos(\beta) = \frac{\left|a\widehat{j}\right|}{9.81}$$

$$cos(\gamma) = \frac{|a \bullet b_z|}{9.81}$$

$$cos(\gamma) = \frac{\left|a \bullet (0\widehat{i} + 0\widehat{j} + 1\widehat{k})\right|}{9.81}$$

$$cos(\gamma) = \frac{\left|a\widehat{k}\right|}{9.81}$$

3.4 Real-Life Application

Let's try and apply this in a real-life situation. Let's say you rotate your hand, and thus rotating the accelerometer with a mass system of 1 kg, let's tilt the accelerometer in a way such that the force vector makes a 30° angle with the x-axis and makes an angle of 45° with the y-axis. So we know that

$$F = 9.81N$$

$$9.81N = 1kg\ (9.81cos(\alpha)\widehat{i}\ + 9.81cos(\beta)\widehat{j} + 9.81cos(\gamma)\widehat{k})$$

$$9.81m/s^2 = \ (9.81cos(45)\widehat{i}\ + 9.81cos(60)\widehat{j} + 9.81cos(\gamma)\widehat{k})$$

$$g = (6.94\widehat{i}\ + 4.91\widehat{j} + 9.81cos(\gamma)\widehat{k}) = 9.81m/s^2$$

So we deduce that the acceleration due to gravity in the x component is $6.94m/s^2$ and the y component of the acceleration due to gravity is $4.91m/s^2$. To determine the angle between the z-axis and the acceleration due to gravity, and magnitude of the z component we shall find the magnitude of a vector

$$\left|(6.94\widehat{i}\ + 4.91\widehat{j} + 9.81cos(\gamma)\widehat{k})\right| = 9.81m/s^2$$

$$9.81^2 = \ 6.94^2 + 4.91^2 + (9.81cos(\gamma))^2$$

$$96.24 = \ 48.12 + 24.06 + (9.81cos(\gamma))^2$$

$$96.24 = \ 72.18 + (9.81cos(\gamma))^2$$

$$96.24 - 72.18 = \ (9.81cos(\gamma))^2$$

$$24.06 = \ (9.81cos(\gamma))^2$$

$$9.81cos(\gamma) = \sqrt{24.06}$$

$$9.81cos(\gamma) = 4.91\ m/s^2$$

$$cos(\gamma) = 0.5$$

$$\gamma = arcos(0.5)$$

$$\gamma = 60^\circ$$

So the magnitude of the force acting in the z-direction is $4.91 m/s^2$ and the angle between the z-axis and the gravity acceleration vector is 60°.

3.5 Computational Data Values

However, we are not writing the accelerometer values on a paper, we are obtaining this data from a computer, and in a computer, we measure the force due to gravity by LSB sensitivity. LSB sensitivity is simply 1 volt per 1 acceleration due to gravity(1.00g). LSB sensitivity changes the number of volts that are transmitted based on your configuration. A higher LSB sensitivity means that acceleration values are more prone to change as the accelerometer is sensitive to a voltage change. The acceleration measured mV/g the mV is millivolts and the g is the acceleration due to gravity as $1g = 9.81 m/s^2$. The value of 1g in milli-volts provided by the capacitors in our configuration is 16384mV so the LSB sensitivity is $16384 mV/g$ as given by the MPU-6050 instruction manual. Keep in mind we chose this value as it is the most sensitive and the most appropriate configuration to use. We could have set up the max voltage to 8,192mV or 4,096mV which are less sensitive but less accurate in our circumstances. So if we divide the obtained voltage values gained by the computer by 16384 mV/g this gives us the acceleration value in g's. The following line of code applies what we just explained.

```
gX = accX/16384.0;
gY = accY/16384.0;
gZ = accZ/16384.0;
```

gX is the x component of the acceleration vector and accX is the voltage provided by the mass-spring system coincident to the x-axis. gY is the y component of the acceleration vector and accY is the voltage provided by the mass-spring system coincident to the y-axis. Finally, gZ is the z component of the acceleration vector and accZ is the voltage provided by the mass-spring system coincident to the z-axis

So in our previous example, we can translate the vector equation to g as follows

$$9.81 m/s^2 = (6.94\widehat{i} + 4.91\widehat{j} + 4.91\widehat{k})$$

$$\frac{9.81 m/s^2 = (6.94\widehat{i} + 4.91\widehat{j} + 4.91\widehat{k})}{9.81 m/gs^2}$$

$$1g = (0.71\widehat{i} + 0.5\widehat{j} + 0.5\widehat{k})$$

The following figures are data values that were obtained and represented by the Arduino serial monitor (keep in mind that there is a small degree of error, around $\pm 0.05g$, due to the change of acceleration due to gravity in different locations and the systematic error of the equipment)

```
0.95 = gravX: 0.01 gravY: -0.02 gravZ: 0.95
0.95 = gravX: 0.01 gravY: -0.02 gravZ: 0.95
0.96 = gravX: 0.02 gravY: -0.01 gravZ: 0.96
0.95 = gravX: 0.01 gravY: -0.01 gravZ: 0.95
0.95 = gravX: 0.01 gravY: -0.01 gravZ: 0.95
0.99 = gravX: 0.05 gravY: 0.00 gravZ: 0.99
0.95 = gravX: 0.00 gravY: -0.02 gravZ: 0.95
0.95 = gravX: -0.00 gravY: -0.02 gravZ: 0.95
0.96 = gravX: -0.00 gravY: -0.02 gravZ: 0.96
0.95 = gravX: 0.00 gravY: -0.02 gravZ: 0.95
0.95 = gravX: 0.00 gravY: -0.01 gravZ: 0.95
```

Figure 10- Vector data values when the accelerometer is nearly flat

```
0.95 = gravX: -0.46 gravY: -0.63 gravZ: 0.55
0.96 = gravX: -0.45 gravY: -0.62 gravZ: 0.57
0.96 = gravX: -0.45 gravY: -0.61 gravZ: 0.59
0.95 = gravX: -0.44 gravY: -0.61 gravZ: 0.58
0.96 = gravX: -0.43 gravY: -0.61 gravZ: 0.61
0.95 = gravX: -0.42 gravY: -0.61 gravZ: 0.60
0.96 = gravX: -0.44 gravY: -0.60 gravZ: 0.60
0.95 = gravX: -0.42 gravY: -0.61 gravZ: 0.60
0.95 = gravX: -0.42 gravY: -0.61 gravZ: 0.59
0.94 = gravX: -0.42 gravY: -0.62 gravZ: 0.56
0.96 = gravX: -0.41 gravY: -0.61 gravZ: 0.61
```

Figure 11- - Vector data values when the accelerometer is tilted on all axes

In figure 10, the accelerometer is laid flat on the table, and that's why the acceleration is, more or less, acting on the z-axis and the vector sum of the different components is around

1.00g. While in figure 11 the accelerometer was rotated on all the axis and each mass-spring system experiences a component of acceleration due to gravity. Once again, the vector sum of each component is around 1.00g. So no matter how we rotate the object, the magnitude of the acceleration will always be around 1.00g.

Lets calculate the angle between each axis and the first vector equation in figure 11.

$$0.95g = (-0.46\widehat{i} - 0.63\widehat{j} + 0.55\widehat{k})$$

We use the equation mentioned in section 3.3 but rather than using vector 9.81m/s we will use 0.95g

$$cos(\alpha) = \frac{|a\widehat{i}|}{0.95}$$

$$cos(\alpha) = \frac{|-0.46|}{0.95}$$

$$cos(\alpha) = 0.48$$

$$\alpha = arccos(0.48)$$

$$\alpha = 61.0^\circ$$

$$cos(\beta) = \frac{|a\widehat{j}|}{0.95}$$

$$cos(\beta) = \frac{|-0.63|}{0.95}$$

$$cos(\beta) = 0.66$$

$$\beta = arccos(0.66)$$

$$\beta = 48.5^{\circ}$$

$$cos(\gamma) = \frac{|\widehat{ak}|}{0.95}$$

$$cos(\gamma) = \frac{|0.55|}{0.95}$$

$$cos(\gamma) = 0.58$$

$$\gamma = arccos(0.58)$$

$$\gamma = 54.6^{\circ}$$

4- Movement

4.1- A person's acceleration

If you begin to walk you will start from rest (0m/s) and speed up, you accelerate and generally while moving, us humans will rarely move at constant speeds, whether it is us moving our hands, speeding up or slowing down the general vibration that happens when you take a step can all be detected by an accelerometer, which is why we set the accelerometer to the highest LSB sensitivity value. So unless there is a perfect human with suspensions that can maintain a constant speed for a long period of time, the accelerometer will calculate an increase in acceleration. Let us say that the acceleration due to movement is given the value μ and we are accelerating with a magnitude of $2m/s^2$ in the positive x coordinate, we could refer to this as

$$\mu = (2\widehat{i} + 0\widehat{j} + 0\widehat{k}) = 2m/s^2$$

$$\mu = (0.203\widehat{i} + 0\widehat{j} + 0\widehat{k}) = 0.2g$$

4.2 Total Acceleration Acting on the Accelerometer

So to calculate the sum of the magnitude of the total acceleration vector we must add the acceleration of the person and the acceleration due to gravity, then find the magnitude of their total acceleration. Once again, μ is the acceleration when moving and g is the acceleration due to gravity.

$$\mu + g = (0.203\widehat{i} + 0\widehat{j} + 0\widehat{k}) + (0\widehat{i} + 0\widehat{j} + 1\widehat{k})$$

$$\mu + g = (0.203\widehat{i} + 0\widehat{j} + 1\widehat{k})$$

$$|\mu + g| = \sqrt{0.203^2 + 0^2 + 1^2}$$

$$|\mu + g| = \sqrt{0.042 + 1}$$

$$|\mu + g| = \sqrt{1.042}$$

$$|\mu + g| = 1.02g$$

$$|\mu + g| = 10.01 m/s^2$$

We code the system such that if the acceleration is above the average resting value, a step is measured, so if the Arduino gains a value above 1.00g, a step is measured.

4.3 Design Flaw

However, this has a major design flaw that even large companies face. When a drastic change in the net acceleration (generally when it is greater than 1.00g such as the example above) a step is counted. This is problematic that people can stimulate their steps by moving their hands as if they were moving without actually moving. That's why this is not very effective and why companies like Fitbit must design their products in a way so that steps can’t be simulated. Later on we will discuss how this problem is resolved.

5- Activating and Using the Components using Binary

5.1 Chip Communication

The MPU-6050 has a separate system and configuration from that of the Arduino. This means that the MPU-6050 can’t be programmed without the help of an external configuration that must be imported that can translate the code written digital signals which the MPU-6050 can understand. This external configuration also allows the Arduino to translate signals into data, which the Arduino can translate to data, in our case, acceleration.

5.2 Libraries and classes

A library is a function that can hold an 8-bit data value called a class. Each function controls parts of the MPU-6050 chip. Based on the class you call in the library, it can initiate or deactivate systems or change sensitivity values. The MPU-6050 has specific configurations within libraries that can only be used by referring to the classes using binary. The user manual of the device shows the bits that must be referred to.

4.28 Register 107 – Power Management 1
PWR_MGMT_1

Type: Read/Write

Register (Hex)	Register (Decimal)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
6B	107	DEVICE_RESET	SLEEP	CYCLE	-	TEMP_DIS	CLKSEL[2:0]		

Figure 12 - A table from the MPU-6050 manual on the library 'Read/Write' ("MPU-6000 And MPU-6050 Product Specification Revision 3.4")

Figure 12 is a table of the library 'Read/Write' and shows you what each bit controls. So if bit 3 is on, more precisely, is a 1, the temperature sensor is disabled.

5.3 Translating Binary

In order to code the functions of the accelerometer effectively, we must understand how to count in binary. There is a geometric pattern observable in binary, each bit u_i (where i is the position of the bit) is equal to the value of the bit before them u_{i-1} multiplied by the common ratio r which is 2 in our case. We can write this relationship as

$$u_i = 2u_{i-1}$$

We know that $u_1 = 1$ so we can represent this is the geometric series equation

$$u_i = u_1 r^{i-1}$$

$$u_i = (1)2^{i-1}$$

$$u_i = 2^{i-1}$$

However, a bit is only counted if its represented as a 1, if the bit i is 1 we give the bit value of u_i if the bit i is a 0 however, we give it a value of 0. That in mind we use the following equation to read binary numerals

$$x = \sum_{i=1}^{n} u_i(b_i) \quad n \in Z^+$$

$$x = \sum_{i=1}^{n} 2^{i-1}(b_i) \quad n \in Z^+$$

Where b is bits, n is the number of bits (n is usually equal to 8 because 1 byte = 8bits) and i is the position of the bit. So if we were to take an 8-bit data value, such as 0b10110011, this is how we would calculate it.

$$x = \sum_{i=1}^{n} 2^{i-1}(b_i)\ n \in Z^+$$

$$x = 2^0(1) + 2^1(1) + 2^2(0) + 2^3(0) + 2^4(1) + 2^5(1) + 2^6(0) + 2^7(1)$$

$$x = 1 + 2 + 0 + 0 + 16 + 32 + 0 + 128$$

$$x = 179$$

5.4 Determining Class Numbers

This is needed to understand the transmissions of the board. The following lines of my code

```
Wire.beginTransmission(0b1101000);
Wire.write(0x6B);
Wire.write(0b00000000);
Wire.endTransmission();
```

Each one of these wire commands calls upon a library(begin transmission, write), each library has different classes and using bits you are able to call the class you want. Take for instance the first line, Wire.beginTransmission(0b1101000); which calls upon the library begin transmission, and the class

$$x = 2^0(0) + 2^1(0) + 2^2(0) + 2^3(0) + 2^4(1) + 2^5(0) + 2^6(1) + 2^7(1)$$

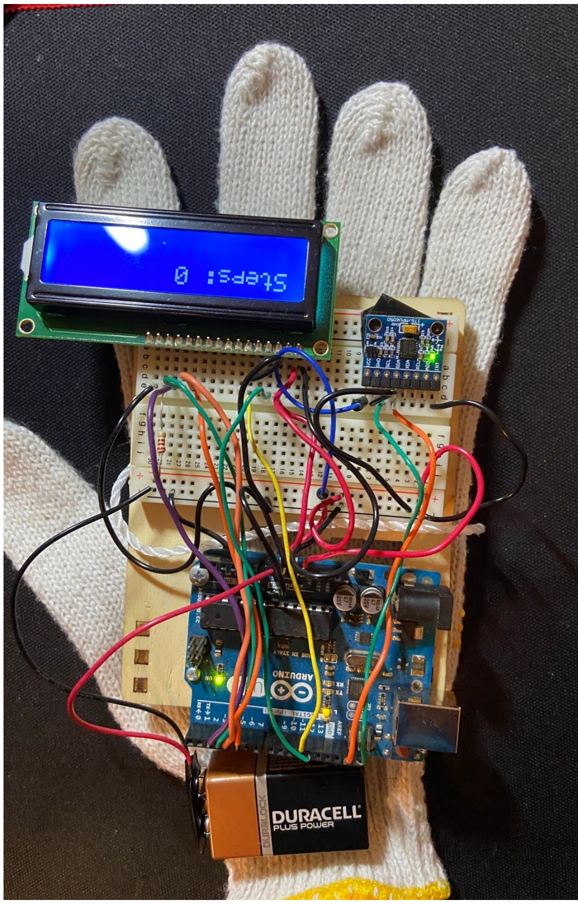
$$x = 0 + 0 + 0 + 0 + 32 + 0 + 64 + 128$$

$$x = 224$$

So class 224 in the library is called upon. This class does the following: the MPU-6050's transmission and the clocking are initiated, it also activates the accelerometer and the gyroscope.

6- Final Product:

6.1- The glove



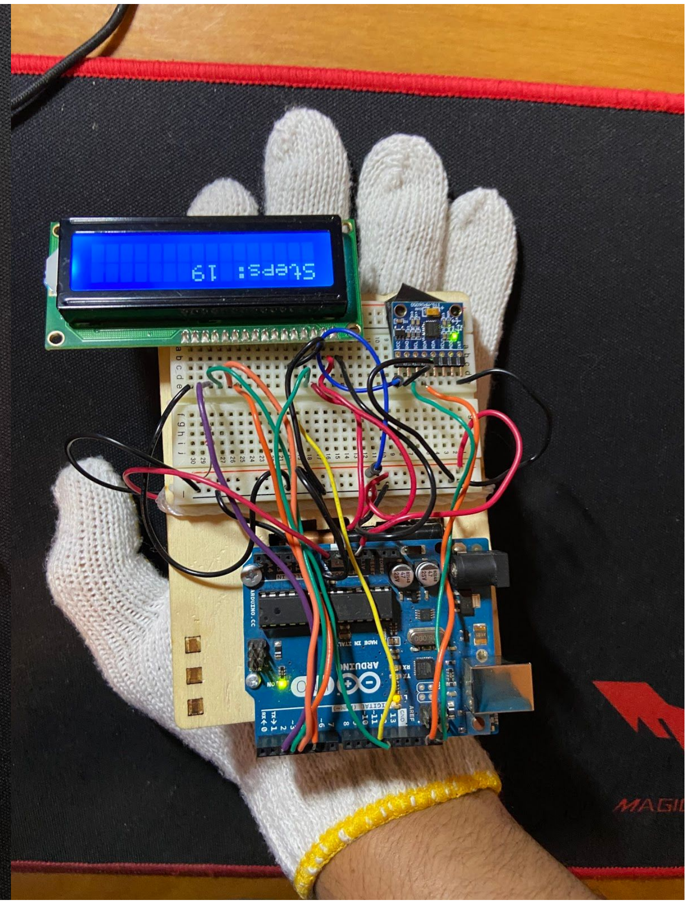


Figure 13 (left) A glove that measures your steps (not worn)

Figure 14 (right) A glove that measures your steps (worn and used)

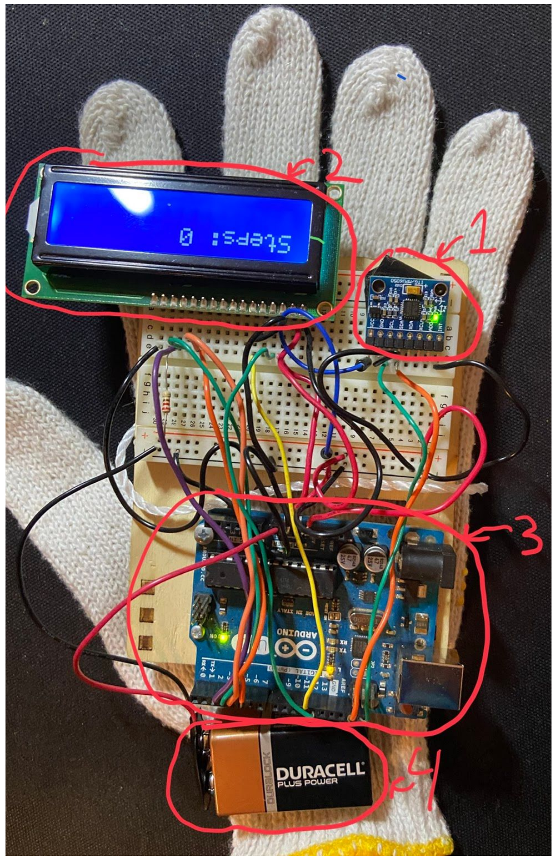


Figure 15: A Labeled figure of the main components of the glove.

These are the following components that were labeled in figure 15

1. MPU-6050 chip
2. LCD screen
3. Arduino UNO board
4. 9V battery

Figure 13 gives us a photo of the glove I built. The Arduino board and the breadboard (Which Contains the accelerometer and the LCD screen) is attached to a glove, the easiest way to measure a person's acceleration with the available equipment. The screen doesn't affect any of the results, however, it's used to display the number of steps taken.

6.2- The MPU-6050's and Arduino's Communication Terminals

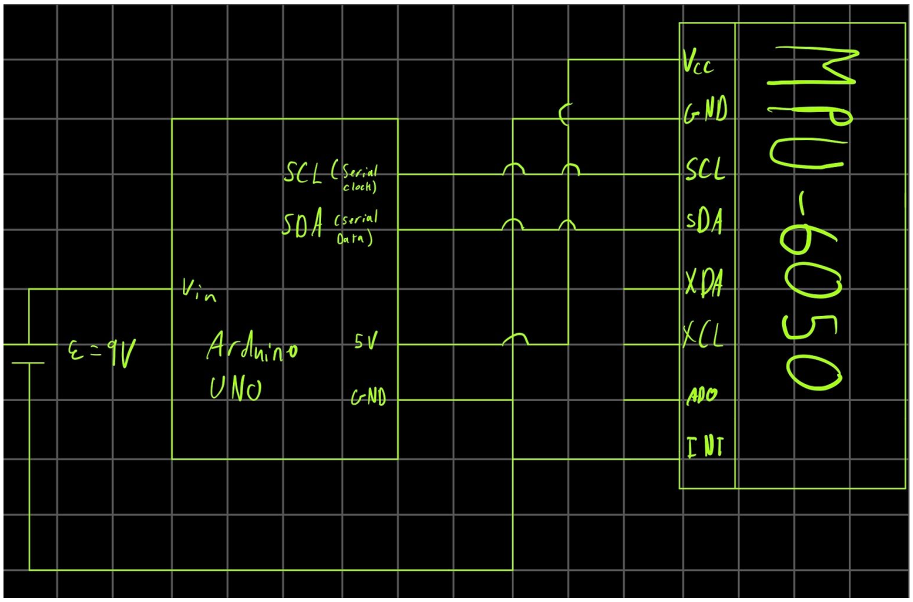


Figure 15 A simplified schematic view of the Arduino and MPU-6050 chip

We discussed in part 5 that the MPU-6050 chip and the Arduino need to communicate using the binary. This is done through two critical ports, the serial clock (SCL) and the serial data (SDA), which are connected to one another as seen in figure 15.

The SCL is a port that synchronizes the MPU-6050 with the Arduino board. All the libraries and the classes are called upon from the SCL, think of it as an input for the MPU-6050 and an output for the Arduino.

The SDA, on the other hand, is a port that obtains data from the accelerometers, gyroscope and temperature sensor that are in the MPU-6050 and send these data values to the Arduino. Think of it as an input from the Arduino and output for the MPU-6050.

6.3 Reflection

This glove utilizes some of the most important technology of our time, however, it’s far from perfect. As discussed in part 4.3, it can stimulate steps and a person’s hand isn’t the best part to place the device. It’s differently bulky, and while the code and the devices utilized can be built upon and developed in a more complex manner, the glove itself wouldn’t be used by a person's daily routine as its better to use a smartwatch or a phone. This product has a lot of competition making it inefficient in the market. However, with the equipment available this is the most effective way to develop a device that can measure your steps. If I were to work on a similar project, I would find a better way to transmit the information to a phone rather than having a bulky LCD screen and place the device on a person’s wrist rather than their hand to try and measure their heart rate to detect changes in their heart rate when moving. During this project, I learned more about the MEMS system and their mathematical manipulation of physics equations in order to produce complex systems. I aspire to become a robotics engineer and this project showed me the mathematical and physical approach engineers have when developing a product. This project could be looked on from a different approach as I want to produce a glove that works as a virtual reality controller where your hand and thumb movements are recreated in a video game.

7- Conclusion:

Micro-electrical mechanical systems (MEMS) implement Newtonian and electro-physics ideas in order to integrate mechanical and electrical components from which it can create electrical data values from a change in the mechanical systems on a micro-scale. Accelerometers implement Hooke's law, Newton's second law of motion & the formula of capacitance to create a relationship between the change in capacitance & acceleration. I applied the knowledge I learned in physics and used the fundamental ideas of algebra to explain how an accelerometer works.

The acceleration due to gravity can be represented using a 3-D vector equation. This helps us relate the acceleration to each and every component. The acceleration due to gravity on each and every spring system was measured by the cosine of the angle between said axes of that component and the acceleration vector.

Using the Arduino serial monitor we were able to get a value of each of the vector components using the unit g. Where 1g=9.81m/s. We discussed the fact that the sensitivity could be increased by giving the value of 1g a higher LSB voltage value. We saw that no matter how we rotate the accelerometer, the magnitude of the vector is 1.00g.

To determine a step we would have to detect a spiking and gradual increase in the absolute value of the acceleration, generally somewhere above 1.00g as an average walking speed. We discussed however that this is problematic since users could stimulate steps without actually taking any, so to prevent this, companies either track heart rates, with steps or design them so they are put in more optimal locations, such as your wrist.

To code the MPU-6050 chip, and the Arduino, we needed to import an external configuration. This configuration used binary in order to communicate different commands and data values between the two devices. So we call upon a library, then we chose on or off (1s or 0s) for specific bits. The final 8-bit binary code, which we represented as a geometric series, represents a numbered class which has specific configurations such as turning systems on or off, or increasing or decreasing a sensitivity value.

All of our research and implementation allowed us to build a portable glove capable of measuring steps. Its major components being the Arduino board and the MPU-6050 chip. Their ability to communicate is achieved through the serial clock (SCL) and the serial data (SDA). The components used in the project are impressive but its bulkiness causes issues in movement, to develop this project, we must try to decrease its size and locate it on the wrist whilst using another sensor to measure changes in the heart rate.

8-Bibliography:

1. Keim, Robert. "Introduction To MEMS (Microelectromechanical Systems)". *All About Circuits*, 2018, https://www.allaboutcircuits.com/technical-articles/introduction-to-mems-microelectromechanical-systems/. Accessed 10 July 2019.
2. Khenkin, Alex. "Sonic Nirvana: MEMS Accelerometers As Acoustic Pickups In Musical Instruments". *Fierceelectronics*, 2009, https://www.fierceelectronics.com/embedded/sonic-nirvana-mems-accelerometers-as-acoustic-pickups-musical-instruments. Accessed 6 Aug 2019.
3. SantoPietro, David. "Simple Harmonic Motion In Spring-Mass Systems Review". *Khan Academy*, 2016, https://www.khanacademy.org/science/ap-physics-1/simple-harmonic-motion-ap/spring-mass-systems-ap/a/simple-harmonic-motion-of-spring-mass-systems-ap. Accessed 28 July 2019.
4. Romanov, Vladimir. "Ep. 57 Arduino Accelerometer & Gyroscope Tutorial MPU-6050 6DOF Module". *Youtube.Com*, 2016, https://www.youtube.com/watch?v=M9lZ5Qy5S2s&list=WL&index=108&t=0s. Accessed 17 August 2019.

5. "How Accelerometer Works? | Working Of Accelerometer In A Smartphone | MEMS Inside Accelerometer". *Youtube.Com*, 2017, https://www.youtube.com/watch?v=T_iXLNkkjFo. Accessed 10 July 2019.
6. "MPU-6000 And MPU-6050 Product Specification Revision 3.4". *Invensense.Com*, 2019, https://www.invensense.com/wp-content/uploads/2015/02/MPU-6000-Datasheet1.pdf. Accessed 14 July 2019.
7. "What Is MEMS Technology?". *Memsnet.Org*, 2019, https://www.memsnet.org/mems/what_is.html. Accessed 9 Sept 2019.

9-Appendices

9.1 Appendix A: Arduino Code

```
#include <Wire.h>
#include <LiquidCrystal.h>

LiquidCrystal inp(12,11,5,4,3,2);

long accX,accY,accZ;
float gX,gY,gZ;
float gXs,gYs,gZs;

float vector;
float speedM;
int steps;



void setup()
{
  Serial.begin(9600);
  Wire.begin();
  MPUsetup();
  inp.begin(16,2);
  }

void loop()
{

  accRec();
  gXs = sq(gX);
  gYs = sq(gY);
  gZs = sq(gZ);
 vector = sqrt(gXs + gYs + gZs);
 speedM = (((vector - 0.96)*9.81)*0.250);

 if (speedM<0)
```

```
 {
  speedM=0;
  }
  if (vector > 1.60)
  {
    steps++;
    }


 //inp.setCursor(0,1);
// inp.print("Speed: ");inp.print(speedM); inp.print("m/s");

 inp.setCursor(0,0);
 inp.print("Steps: ");inp.print(steps);

    printD();

    delay(250);
}

void accRec()
{
  Wire.beginTransmission(0b1101000);
  Wire.write(0x3B);
  Wire.endTransmission();
  Wire.requestFrom(0b1101000,6);

  accX = Wire.read()<<8|Wire.read();
  accY = Wire.read()<<8|Wire.read();
  accZ = Wire.read()<<8|Wire.read();

  gX = accX/16384.0;
  gY = accY/16384.0;
  gZ = accZ/16384.0;
  }

void printD()
{ Serial.print(vector);
  Serial.print(" = gravX: "); Serial.print(gX);
  Serial.print(" gravY: "); Serial.print(gY);
  Serial.print(" gravZ: "); Serial.print(gZ);
  Serial.println();

  }
```

```
void MPUsetup()
{
 Wire.beginTransmission(0b1101000);
 Wire.write(0x6B);
 Wire.write(0b00000000);
 Wire.endTransmission();
 Wire.beginTransmission(0b1101000);
 Wire.write(0x1C);
 Wire.write(0b00000000);
 Wire.endTransmission();
 }
 }
```

9.2 Appendix B: Schematic of the Device