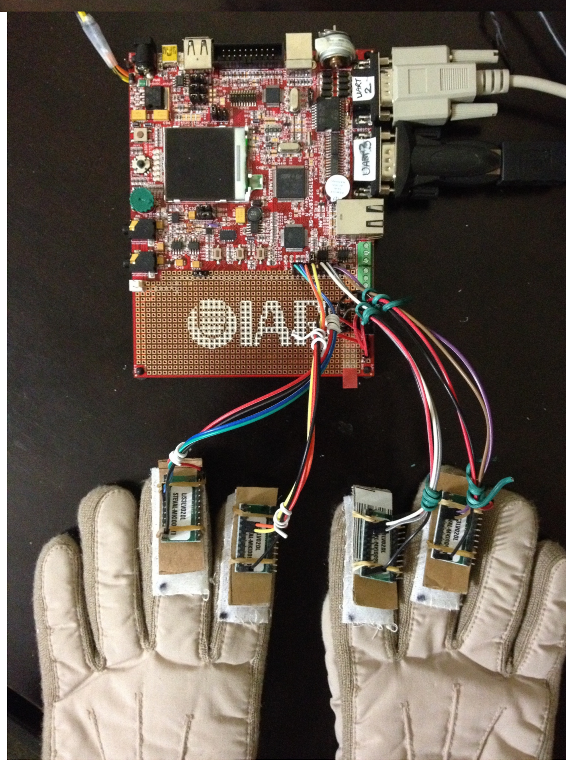
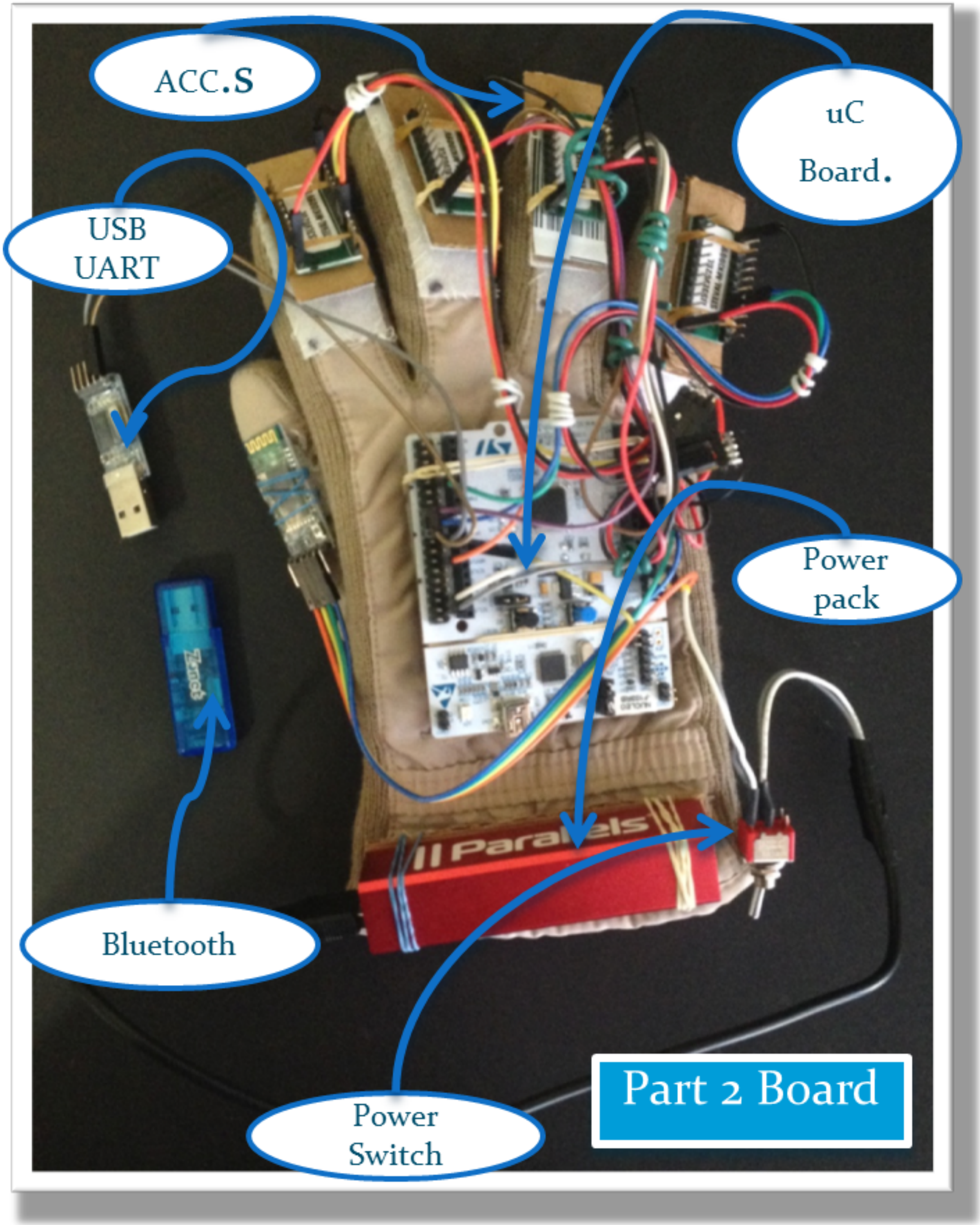
# KEY-BOARDLESS TYPING

YEAR:
2014-15

Using accelerometer sensors to program a virtual typing system. – Computer Science

Science Fair


IAR


ACC.S
uC
Board.
USB
UART
Power
pack
Bluetooth
Power
Switch
Part 2 Board

# TABLE OF CONTENTS

## INTRODUCTION

In 1868, the QWERTY keyboard was invented into the type writer by Christopher Latham Sholes. Since then, revolutions have been made to keyboards thin as paper, twistable keyboard and portable keyboard through lasers. But what they all have in common is, one, take space on a cluttered desk, two, do not allow typing to be done on different surfaces, different distances of hands etc.

The goal behind the project is to create a typing method that does not involve a hard keyboard typing surface, mobility of hands, and free to type on any surface.

Using accelerometers on each finger, the code determined the movement of the finger to logically determine the character wanting to be typed.

This revolutionary concept can achieve efficient use of space, easy mobility between writing, typing, and other hand movement. It also has a potential to help disabled type on any surface, with personalized positioning. Along with this many concepts that require the movement of fingers could be done virtually (explained more areas in Future Application).

The software and hardware made incorporates the capability to type on any surface, and at any distance, making the typing *virtual*.

# HYPOTHESIS

## *GOALS*

The goal of the project is to create a *virtual* typing system that does not require the use of a physical keyboard. Using accelerometer sensors, the program should be able to compute a code based on specific acceleration and movement of the accelerometer.

The code should detect movements, and compute for a specific key that would be pressed on a QWERTY keyboard if the movements of the fingers were corresponding.

Another goal is for the product to increase efficiency of typing and workspace. Also certain paralysis limit the movement of the hands to just fingers, and the goal is that this should enable them to type.

## *EXPEXTED OUTCOMES*

The final product is expected to be a glove that is easy to wear and convenient that increases the efficiency of the workspace, tying into the goal.

It is expected that the typing can be done on any surface, any distance between hands, thus allowing the paralyzed type without much movement of their hands. This will occur due to the typing only requiring the movement of the fingers, and reducing the movement needed for the whole arm. For example, if the disabled cannot move there arm beyond a certain point, they can still type, as through this software only movement of the fingers is needed.

## *CONSTRAINTS AND CRITERIA:*

Several constraints were present:

- Cost constraints were the biggest constraint and the biggest limiting factor, it limited the type of accelerometer devices, microcontrollers I could buy, thus reducing the possibility of meeting some of the goals and expected outcomes.
- Mobility. This was a constraint present for the glove, that required the mobility of the fingers while wearing the device, thus allowing the fingers to type.
- Number of accelerometers. Due to the cost constraint, the number of accelerometers was limited to only four and that too not very accurate. Thus this limited possibilities for a full keyboard.
- Time. Many of the small problems in the code could not be fixed due to the little time that was available.
- Coding knowledge, due to the lack of high level knowledge many options for the flow of the code was limited. But with high level knowledge, the coding and software could become smoother and more accurate.

# MATERIALS

## *MAJOR*

These are materials absolutely needed, specifically for coding.

1. Microcontroller – STM32F107VC-SK (IAR systems)
2. Accelerometer – 4 - More is wanted - (STEVAL-MKI009V1)
3. Computer
    a. IAR Embedded workbench
    b. Dock-light System
    c. USB connectors
4. Attachment wires – 25
5. Millimeter
6. Header pins
7. External Power Supply

## *MINOR*

These are materials that are not as important, mostly for the glove platform:

1. Glue-Gun
2. Soldering Iron
3. Cardboard
4. Velcro
5. Rubber bands; Plastic ties.
6. Scissors
7. Sewing Kit
8. Two gloves

# PROCEDURE

## *SUMMARY OF PHASES*

### *PHASE 1*

Pre-experiment, downloading the needed software's, and testing microcontrollers are done in this phase.

### *PHASE 2*

Attaching accelerometers on the microcontroller is done in this phase using GPIO pins on the Microcontroller.

### *PHASE 3*

This phase included building the platform for the accelerometers onto the glove, and making a base for the final product.

### *PHASE 4*

Attaching accelerometers onto the glove, to ensure ability to move freely in the glove, and being able to stimulate typing was/is done in this phase.

### *PHASE 5*

Logic and coding, the coding for the program to compute certain keys corresponding to the QWERTY keyboard are major components of this phase.

## *DETAILED PROCEDURE*

### *PHASE 1*

1. Download the following:
    I. IAR work bench – with the CD provided with the microcontroller kit.
    II. Dock light – from docklight.de
2. Download the necessary software and programs needed for data transfer to the microcontroller.
    I. J-link will be available inside the CD, or will download with the attaching of microcontroller

3. Test the following on the microcontroller
    I. Using the pre-made codes given with the CD, and in IAR workbench, build, compile and run them on the microcontroller.
    II. Test for each of the codes to ensure that the microcontroller is properly functioning.
4. Make sure to get familiar with the IAR workbench (How to build, How to Compile etc). Also get familiar with the coding and know some basics.

### *PHASE 2*

1. For this phase, gather the following:
    a. Extension wires
    b. Microcontroller
    c. Four accelerometers
    d. Header pins
    e. Power-supply – This can be a wire with a plug for the outlet. Refer to photos for details.
2. If the accelerometer did not come with header pins attached, solder the header pins onto the sockets if necessary. Take precaution from burning as the soldering is in a small space. Take notice to not shot any pins together.
3. Using the solder and soldering iron, shot pins 1 and 2 on the accelerometer. And connect a small wire to pin 19 on the accelerometer. Use caution and supervision of adult.
4. Connect an external power source to the microcontroller, as the USB power is not enough. This is located towards the top left of the microcontroller. Will require shouldering and power connection. Be careful for voltage to prevent too much flow of power.
5. Connect header pins onto the GPIO pins on the microcontroller, on PE2, PE3, PE4, PE5, PE8, PE9, PE10, and PE11.
6. Connect extension wires onto the pins above. Make sure to use different colors and note what color is on what pin.
7. Connect the wires to the accelerometer. Refer to Figure 1 chart.
8. Connect headers to the ground and power pins on the microcontroller. Using schematic of the microcontroller (located in the Extras tab of this binder), find 3.3V or ground connection. Connect header pins on any of these, and extend them to Pin 24 and Pin 1 on accelerometer.

9. Using Dock light, set the correct communication port, and tests the accelerometers one by one. See if the X, Y and Z change, if they do, the connections are correct, and if they do not, check connections and try again. Make sure to have the same signal speed.
    a. This step requires the library to be written, look at PHASE 5 for this.

Figure 1

| PE on Microcontroller | Connection Type | Accelerometer Pin | Color (Used in this project) |
|---|---|---|---|
| RST | ----- | ----- | ----- |
| PE2 | SCL | P20 – ACC0 | BROWN |
| PE3 | SDA | P21 – ACC0 | PURPLE |
| PE4 | SCL | P20 – ACC1 | GREY |
| PE5 | SDA | P21 – ACC1 | WHITE |
| PE6 | ----- | ------ | ------- |
| PE8 | SCL | P20 – ACC2 | YELLOW |
| PE9 | SDA | P21 – ACC2 | ORANGE |
| PE10 | SCL | P20 – ACC3 | BLUE |
| PE11 | SDA | P21 – ACC3 | GREEN |

## *PHASE 3*

1. Gather the materials needed
    I. Velcro
    II. Cardboard
    III. Scissors
    IV. Glue Gun
    V. Sewing kit
    VI. Pair of gloves
    VII. Accelerometers.
    VIII. Rubber Bands
2. Cut the Velcro the same size as the cardboard for both sides. Leave one of the Velcro sides longer by about 1-1.5 inches.
3. Sew the long side of the Velcro onto the tip of the glove. And the other (smaller one) onto the cardboard.

*<u>PHASE 4</u>*

1. Taking the cardboard, cut a rectangle big enough to hold the accelerometer, and small enough to fit on the glove.
2. Taking the shorter Velcro, using the hot glue gun, glue it onto the cardboard. Take precaution, and make sure no plastic leaks out.
3. Take the 4 accelerometers, and rubber bands (2 – small). With the pins facing up, put the rubber bands to hold the accelerometer in place. Put the accelerometer opposite side of the Velcro.
4. Now, simply, using the Velcro put the accelerometer base onto the glove base.
5. Test the mobility of the glove by putting the glove on, and mimicking typing. Make sure the accelerometers to no come into contact with each other while moving your fingers. If it does, then simply move the accelerometer to a position where it is safe.

*<u>PHASE 5</u>*

1. Write the library (API) for the accelerometers. Make sure to include:
   a. Specify the hardware connection of the accelerometer.
   b. Provide simple interface for sending commands and reading data from accelerometers.
   c. Control the I2C signals (SDA and SCL). And control high and low for the accelerometer from the microcontroller.
   d. Refer to the Code section
2. Write the code that displays the X, Y, and Z on Docklight through UART cable. This will require the library parts. Refer to the code section for the code (this code has parts for step 4).
3. Test the library and code (from sample code of IAR systems) by using Docklight system, and making sure the change in X,Y, and Z can be seen for each accelerometer, this ensures the connections are correct. If nothing is shown, then check for mistakes in library and connections.
4. In addition to the code written in step 3, make criteria and logic for X and Z (Y not used), such that two values are displayed for two different positions (Finger typing positions) for each of the accelerometers.

5. Test the code by doing the following
    a. Determine a sequence wanted (ex. 12345678)
    b. Try to type that sequence and record results.
    c. Count the number of mistakes, and record these as well. Find the percentage of wrong and right (divide by 8).
    d. Repeat 5 trials for the sequence.
    e. Repeat steps a-d for a different sequence.

## FLOW CHART OF CODE AND LOGIC

This is attached in the following page. The flowchart shows how the code is structured, what it does, and the process it follows to distinguish the keys.

For the actual code, refer to the Extra Tab.

# DATA AND ACCURACY

## *GRAPHS*

The Graphs and Data are on the following page. They are all considered one page on the table of contents.

## *DATA ANALYSIS*

- Many of the inaccuracy seen were due to human errors. The following are problems that led to the reasons behind this data.
    - Human unwanted finger movement errors (Further explained in error analysis).
    - Minor problems in the coding that cannot be fixed.
- Finding the mistake posed many questions. It was hard to determine what a mistake was and what was not. The following are some examples of questions that occurred during the project in the process of determining an error.
    - For example, if the sequence tested is 12345678, and the result outputted was 412345678. Is the first four an error?
    - For example, if the sequence tested is 12345678, and the result outputted was 13245678. Is the 3 then 2 an error?
    - Similarly, if the finger is tapped and then no number was received, was it an error?
    - Is human mistake, such as forgetting where the key is, and resulting into a wrong digit an error? (This is further discussed in Error Analysis Tab)

This was major problem in determining a mistake or not. In the project, it was considered that any number mess-up was an error, but the taps were not taken into consideration.

- When trying to type, typing had to be done at a slow speed (not the finger movement, but the time between each finger movement). This does not exactly meet the expected outcome, but with more accurate and smaller accelerometer with a faster signal transition speed in microcontroller and accelerometer can be done. This was not done due to cost constraints.

# <u>**CONCLUSION**</u>

The goal of the project was to create a method of typing without a physical keyboard. The expected outcomes were a wearable device that makes typing efficient and helps the productivity in the workspace *(Look at Goals and Hypothesis Tab for details*). This would be a method to help paralysis by reducing the movement needed to type. Using accelerometers, and a logic code, eight typing positions were developed. The testing method included determining a specific sequence wanted, and testing the positions to see accuracy (*See Procedure and Data for further detail.*) Five trials were done on each sequence, giving a total of 40 tests. The results later were analysis and several problems were found (*Discussed with greater detail in Data and Accuracy.*)

Using the sequence and the trials, accuracy rate was found, and recorded. Alongside with the coding, debugging was done to increase the accuracy by changing minor values of variables such as delay time, criteria threshold etc. This resulted into a final product with 83.3% accuracy. When the data was analyzed, one major factor that was decreasing the accuracy was human error (*Error Analysis/Improvements to Make*). When the mistakes that were due to human error were taken out, accuracy increased by about seven percent, making the accuracy 90.83%, and the rest of the inaccuracy was due to coding error that could not be fixed (*For data of all trials, see Data and Accuracy*). Many of the limitations were due to the big size of the accelerometer, and the decrease in mobility due to the glove.

Looking at the goal of the project, if taken 90 percent accuracy as the final accuracy, it can be acknowledged that the goal has been met. Although the expected outcome was a full keyboard, this could not be done due to cost constraints, lack of enough technology to do so (such as more accelerometers, and more accurate accelerometers). An expected outcome was a wearable object, which was met; however it did not exactly meet the efficiency criteria. The typing had to be done very slowly due to the slow speed of signal transmission between microcontroller and accelerometer. But without a cost constraint, and better accelerometer with faster microcontroller, these goals and expected outcomes can be met with the same coding and logic present in this project. As mentioned, many of the errors or mistakes in the trials was due to human error, and without this, the accuracy was enough to meet the expected goal.

The reason of this project was to help the disabled, and with the factors mentioned, better microcontroller etc, the full keyboard can definitely be achieved with the same code that was written in this project. Many other places can also be benefited from this concept of virtual typing or the concept of simplicity by reducing equipment to simple body movement (*Future Application)*. For example playing an instrument could be done by simple finger movements. The more obvious application is the keyboard. Without the cost constraint, the full keyboard can be made with the same concept and coding done in this project.

## ERROR ANALYSIS/ IMPROVEMENTS TO MAKE

### *ERRORS*

- The reason for the errors in the data or mistakes in the sequence was mostly due to human errors. When human errors were taken out, the accuracy increased by about 7-10% (by taking the wrong values out of the result that were supposedly due to human error).
- The following are areas where human error occurred
    - When moving and trying to press one position, the finger next to it also moved. As pressing requires the finger to move down, tendency of the human hand is the move the whole wrist down, causing the other finger on the wrist to move down as well.
    - While moving fingers on one wrist, for example the right fingers, the left fingers tend to move due to the habit of typing on a regular keyboard.
    - Movement of the fingers was limited while having the gloves on due to the bulkiness of the accelerometers and the thick gloves. This made it harder to move fingers fast enough and to move just one finger.
    - On a normal keyboard, while typing, one tends to lift other fingers while only pressing one key. This was a human error in the data as the code computes all values of all accelerometers, thus when one finger is lifted to type, other finger data is also calculated, computed and displayed.
- There were errors in the coding and software side of this, but they were fixed while debugging the code.

### *IMPROVEMENTS THAT COULD BE MADE*

- Improving accelerometers. With the bulky accelerometers, the movement of the hand is restrained to a slow speed, and limited finger movements. With smaller accelerometers, the movement could be free, and allow for faster typing.
- Accurate accelerometers. With more accurate accelerometers, better and numerous detections of movements could be added. For example the 4-5 movements the index finger is responsible for on the keyboard, and with these accelerometers, the movement and detection of 5 is not possible.

- Adding a sensor on the palm. Adding a muscle sensor on the palm to further identify movement of the finger would result into a coding for the complete keyboard (numbers, symbols, punctuation etc).
- More accelerometers. With more accelerometers, more keys could be coded for, resulting into a complete keyboard.
- Many smaller changes in the code could also be done, however the most were done while debugging the code and typing.

# FUTURE APPLICATION

### *FUTURE APPLICATION*

The following are places that this project can be applied:

- Workspace. With the ease and convenience of typing, the workspace efficiency can be increased. By being able to type on any surface, easy transitions between typing, and writing, for example, can be seen. As this method does not need a hard keyboard, one could increase the space on the desk. With this concept, people can type at a comfortable position that a keyboard does not allow, for example hands spread apart. It allows for the user to type on any surface and any location. But overall, despite these applications, this concept can be a great advancement in using the computer if there were less cost constraints.
- Paralyzed. People with limited hand movements could apply this concept of this keyboard and be able to use a standard computer at their will. This concept of the keyboard does not need a huge amount of hand movements, and does not require the stationary keyboard that creates a distance between the two hands that cannot be changed, the paralyzed could benefit. By simply having a glove, and simply moving your fingers, one can easily type and maneuver a computer despite being paralyzed.

### *FUTURE RESEARCH*

The following are places that further research on this concept can be done (excluding Keyboards):

- More accurate keyboard. With further testing and detailed coding, a fully equipped keyboard can be made, but this would require more accelerometer and advanced technology, that was not available in this project. Due to cost constrains this could not be done, but with the data of many accelerometers located in moved spaces while typing could yield to greater data and accurate combinations to compute a key. With many accelerometers, many movements can be sensed, including many minor finger and wrist movements.

- Piano. With better coding, and advanced accelerometers this concept can be applied to making a *virtual* piano. This would require more specific coding, and several accelerometers to detect the movement of the whole end to reach the ends of the keyboard. With more accurate sensors, piano sounds could be stimulated based on different finger movements and wrist movements. This again would require great deal of coding and a bigger budget to supply for these accurate sensors.
- This can be applied to many other places that involve the moving of hands and fingers, for example playing an instrument. With further coding and research with multiple accelerometers, playing an instrument could become and would only require the movement of your fingers.  Or gaming, where you would move fingers with no hardware remote.

# EXTRAS

These are references and resources used in the project.

## *RESEARCH PLAN*

### *Problem/Objective/Experiment:*

a. Purpose:
   1. To make a virtual keyboard, and a way to type, without a physical keyboard.
   2. Using sensors to detect the movement of the fingers, code a program that computes the activated sensors on each finger to result and output of a character (A-Z).
   3. Programming will be dome using a microcontroller, attached will be the sensors using the specific Bus.
   4. The final outcome will have a wearable device (glove-like), that has sensors and one can type with.
      1) The plan is to have a glove, with sensors on it, and connections to a central microcontroller with a display or a display connected to the computer to show the characters typed.
   5. The typing should be able to be done on any type of surface; convenient; etc.

b. Issue:
   1. Make a more convenient and efficient way to type. This will be done by being able to type without a hard board keyboard, and type on any surface.

c. Goal/Expected Outcome:
   1. The goal of the project is the create a virtual typing system that does not require the use of a physical keyboard. Using accelerometer sensors, the program should be able to compute a code based on specific acceleration and movement of the accelerometer.
   2. The code should detect movements, and compute for a specific key that would be pressed on a QWERTY keyboard if the movements of the fingers were corresponding.
   3. Another goal is for the product to increase efficiency of typing and workspace. Also certain paralysis limit the movement of the hands to just fingers, and the goal is that this should enable them to type.

4. The final product is expected to be a glove that is easy to wear and convenient that increases the efficiency of the workspace, tying into the goal.
5. It is expected that the typing can be done on any surface, any distance between hands, thus allowing the paralyzed type without much movement of their hands.

## *Possible Solutions/Experiments*

d. Solution Ideas

1. These sensors can be used on each finger to sense the motion
    1) Accelerometers – axis on which it is moved to sense what finger moved where in relation to the keyboard.
    2) Buttons – different areas on the finger to sense which one is pressed and activated, each would be linked to one character.
    3) Role-Ball – Direction of rolling, coded to one character per direction.
    4) Piezoelectricity – Positive and Negative energy output on different locations on fingers to sense which one is pressed/activated.
    5) Muscle Senses – Using muscle sensors, EMG, to sense what tendon is moved, and relate that to what character is wanting to be typed.

e. Best Possibilities

1. Accelerometers : Accurate, cheaper, but harder to program – Best Choice
2. *Muscle Sensors : Accurate, but costly and hard to program*
3. *Buttons: Many needed, easy to program, very small needed*
4. *Role Ball: Easy to program, somewhat impractical*
5. *Piezoelectric: Minute needed, only yes and no results and external (outside of microcontroller) programming needed.*

## *MATERIALS/PROCEDURE*

f. Materials – May change based on needs as project goes onward.

1.Microcontroller - Used in Project: STM32F107VC-SK (IAR Systems)

2.Sensors - Accelerometers – 4 (STEVAL-MKI009V1)

3.Computer - IAR embedded workbench; Docklight; USB bus.

4.Soldering Iron

5.Millimeter

6.Connection wires – 25

7.Multi-meter

8.Glue Gun

9.Cardboard; Velcro; Rubber bands; Plastic Ties.

g. Location

1. Building Location – Home

2.Analysis Location – Home

3.Testing Location – Home

h. Risk/Safety

1. Soldering Iron/Glue gun – Temperature and burn risk

    1) Protection: Supervision of adult, careful usage.

2.Minor risk of electric shock – Only 3-5 volts being used

    1) Protection: Minor risk, even if contacted with charged wire, little to no shock. Will be prevented by making sure for rubber coverings on wires.

i. Testing Procedure

1. Plan for testing:

    1) Using the product, type a specific sequence
    2) Count the characters that were presented to be wrong and how many were right based on what was wanted to be typed
    3) Find accuracy, and try to improve accuracy through debugging of the program and editing the program
    4) Repeat the steps, to see the increase or decrease in accuracy, and use to change program and code.

2. Measuring

1) Very small measuring will be done for the lengths of the wires needed for the fingers, connections to accelerometers etc. Not very accurate data is required; +/- 0.5 inch.

3.Data Validation

1) Multiple trials will be done with the "sequence testing."
    a. Five trials for each sequence will be done.
2) The experiment will be separated into 5 phases, starting from installation and progressing to 4 sensors on one Microcontroller.
3) The more accurate the characters are, the better the system of coding is.

4.Debugging

1) Debugging the code can be done by the following:
    a. Outcomes verses code; change the code by seeing the outcome and trying to make it better.
    b. Syntax; The minor syntax errors in coding (ex. Brackets, colons etc); usually programming software detects these errors.
    c. Not working; code is wrong, some mistake to be fixed. Use stop and go markings in code to find mistake.

## *BIBLIOGRAPHY*

The bibliography is located in the next tab.

## *BIBLIOGRAPHY*

A beginner's guide to accelerometers. (n.d.). Retrieved November 9, 2014, from http://www.dimensionengineering.com/info/accelerometers

*This website provides the uses, logic, and method behind the accelerometers. It explains the difference between analog and digital, an essential part of the project.*

Accelerometer, Gyro and IMU Buying Guide. (n.d.). Retrieved October 7, 2014, from https://www.sparkfun.com/pages/accel_gyro_guide

*This website shows what an accelerometer is, what to look for when buying an accelerometer and specifies certain options for accelerometer to buy from.*

Brian, M. (n.d.). HowStuffWorks "How Microcontrollers Work" Retrieved October 7, 2014, from http://electronics.howstuffworks.com/microcontroller1.htm

*This website explains what a microcontroller is, the basic components of the microcontroller, and how it works. Not giving specifics to certain microcontroller, a general concept of them.*

Dachis, A. (2011, January 31). Learn to Code: The Full Beginner's Guide. Retrieved October 7, 2014, from http://lifehacker.com/5744113/learn-to-code-the-full-beginners-guide

*This website gives detailed instruction on how to write code with variables, if statements etc. This will be useful for the code for the accelerometers.*

Docklight RS232 Terminal / RS232 Monitor - A Development Tool For Serial Communication Protocols. (2014, September 1). Retrieved November 25, 2014, from http://docklight.de/

*This website is the source for the Docklight display program will be used, it shows how it works, and messaging to and from it.*

Gabay, J. (2011, December 1). Fundamentals of Piezoelectric Shock and Vibration Sensors. Retrieved October 7, 2014, from http://www.digikey.com/en/articles/techzone/2011/dec/fundamentals-of piezoelectric-shock-and-vibration-sensors

*This website explains the fundamentals of piezoelectricity, and how it works in regard to circuits and voltage.*

Introduction to Strain Gages. (n.d.). Retrieved October 5, 2014, from http://www.omega.com/prodinfo/straingages.html

*This website is information on basic concept of strain gages, and how they work.*

Lowe, D. (n.d.). Electronics Components: Introducing Microcontrollers. Retrieved October 7, 2014, from http://www.dummies.com/how-to/content/electronics-components-introducing-microcontroller.html.

*This website has a detailed explanation of the components of a microcontroller and what they do, and what purpose they serve in regard to their significance.*

MICROCONTROLLER PROGRAMMING LANGUAGES. (n.d.). Retrieved October 7, 2014, from http://www.esacademy.com/assets/faqs/primer/7.htm

*Shows variations of different codes, and what they are called. It explains the differences in types of codes, and what they are used for.*

Trapani, G. (2009, November 11). Programmer 101: Teach Yourself How to Code. Retrieved October 7, 2014, from http://lifehacker.com/5401954/programmer-101-teach-yourself-how-to-code

*Although focused on coding for a website, it gives explanations of different types of coding based softwares for example C++.*

UART explainedA guide for new imp programmers. (n.d.). Retrieved November 11, 2014, from https://electricimp.com/docs/resources/uart/

*This website focusus on UART, a connection bus, and how to code for it. Although the coding language is not the same, it still can apply to the project.*

What are Accelerometers? (n.d.). Retrieved October 7, 2014, from http://www.omega.com/prodinfo/accelerometers.html

*This shows specifics on different options present about accelerometers, and what certain specifications mean.*

## ***MATERIAL INFO***

The pages following are receipts, and information about the materials used.

## CODE:

The first code is the library written, the next is the code with the logic and criteria. They are counted as one page in the Table of Contents.

***LIBRARY***

----Next page.

**THIS API IS FOR ONE ACCLEOROMETER. THE OTHER ACCLEROMTERS HAD A SIMILAR API, BUT WITH MINOR CHANGES TO IT.**

***CODE WITH LOGIC AND CRITERIA***