# High Six – The Sign Language Glove

Kirk Chan, Ali Mizan, Brian Troili, and Laura Rubio-Perez

Dept. of Electrical Engineering and Computer Science, University of Central Florida, Orlando, Florida, 32816-2450

***Abstract* — High Six is a sign language glove that provides text and voice translation of the 26 letters of the American Sign Language. This glove is composed of flex sensors, pressure sensors, a gyroscope, an accelerometer and a Bluetooth module. Once a hand gesture is made, the data obtain from the sensors and other hardware components will be sent to the Android device. The Android device will then compare the data with a built in library and it will translate it into a letter of the American Sign Language. This translation will be done via text and voice and it will be displayed on an Android application.**

***Index Terms* — Accelerometers, force sensors, gesture recognition, gyroscopes.**

## I. Introduction

High Six opens the scope of mobile devices to become mediums of translating hand gestures through a glove peripheral. When paired with a compatible smartphone, this product improves sign language user's ability to readily communicate to those unfamiliar with the language. One may use the language glove, capable of detecting distinct hand gestures, to make appropriate letters for the message. This glove shall support wireless transmission to the High Six's Android mobile application to ultimately display the translated message.

High Six aims to help people with hearing and speech impairment. To show the power of the glove, it supports the fundamental letters and numbers of the American Sign Language (ASL). The language glove is designed to capture the position of each finger as well as the movement and orientation of the hand system. By analyzing the specific properties of each gesture, a library is created within the mobile application to allow further communication outside the language-impaired community.

With the power of today's smart phones, High Six conveniently recognizes and translates the hand signs to an audible and readable message for their respective audience. The project's overall system architecture consists of three main subsystems. These subsystems are as follows: unified hand gesture detection, Bluetooth communication, and the Android translation application.

## II. The Gesture Library

The project starts by understanding the fundamental gestures of 26 letters and 10 numbers before discussing the design of each subsystem. Looking the figures below, Figure 1 and Figure 2, these represent the supported gesture library.

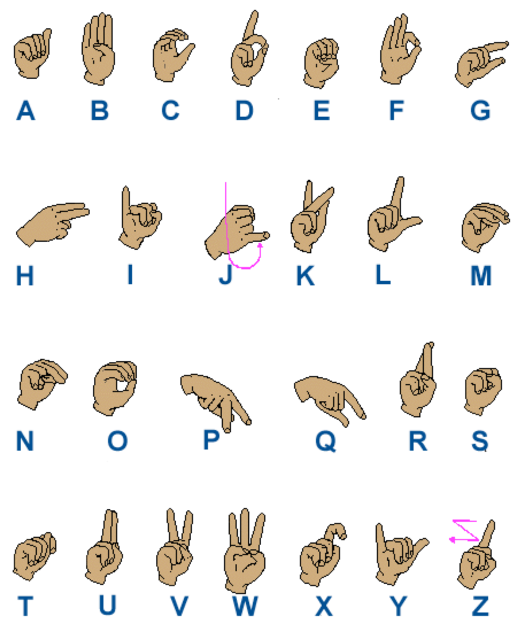



**Figure 1: ASL Alphabet**

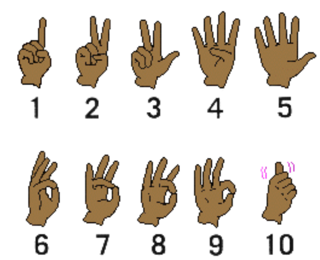



**Figure 2: ASL Numbers**

Within these simply 36 characters, it shows the complexity of the American Sign Language. These characters are static and unique characters from one another while others require special attention to distinguish them. Characters 'R' and 'U' have similar finger placement, but the index and middle fingers cross. This suggests sensors at particular points of contact.

Characters 'I' and 'J' are the same gesture, but the 'J' introduces movement and orientation of the hand. With these examples, the gesture library proposes the challenge of capturing hand gestures and movement. In these next sections, they shall discuss the group's solution to the problem now that there is a better understanding of the data set.

## III. Subsystem Components

As mentioned earlier, this project is composed of three main subsystems that are interfaced to create the overall design. Figure 3, shown below, depicts the functionality within each subsystem and Bluetooth wireless communication held between the two devices:

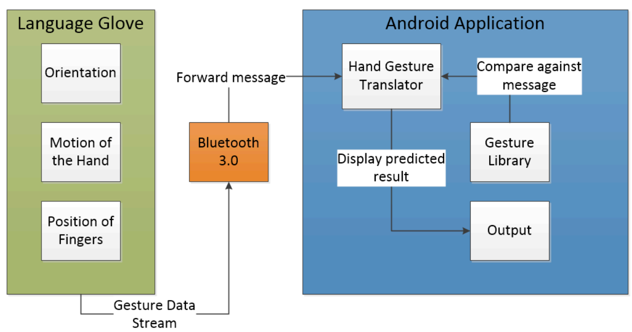



**Figure 3: Subsystem Components**

### *A. Language Glove*

The unified hand gesture detector, or language glove, is one of the main components of this project. The design focus surrounds the design concepts for the language glove. The first factor is low power. To give feasibility to the idea, the glove shall demonstrate little power usage, ultimately allowing for the everyday use. One charge is expected to support glove functionality for up to 16 hours. While 16 hours may not seem long, an average conversation is assumed to be less than or equal an hour. Thus, the glove may hold up to 16 conversations, well within the range of a day's use.

The next key concept is the physical weight of the language glove. As an additional device overlaying the user's hand, the glove should not stress or strain the user while fingerspelling phrases. From this thought, the glove shall weigh less than 1.5 lbs.

The last feature is the wireless communication. The glove shall communicate wirelessly to the mobile device of the user and alleviate the user of struggling with a limited range of movement. Using Bluetooth technology, the user may move freely with the glove activated and translating their message for up to 50 meters,

With these concepts, the unified hand gesture detector's key function is to capture the variable data for each hand sign. The considered factors into each gesture include the following: orientation of the hand, position of the fingers, and the motion of the entire hand. After initial setup needed to pair the two devices, both smartphone and glove, the glove shall seamlessly deliver the data.

### *B. Bluetooth*

To attain the information of the hand gesture, the glove subsystem forwards raw data through the Bluetooth module. The microcontroller first synchronizes the signals from the sensors before the data is sent wirelessly. The Bluetooth communication subsystem serves as the bridge for the wireless data transmission between the Android application and the language glove. To establish this connection, the Android application transmits a request signal when in use. Meanwhile, the glove subsystem waits in standby until it hears the request. Next, the language glove will confirm the connection request. Upon receiving the approval from the glove, the Android application establishes a connection, ready to receive data.

### *C. Android Application*

The Android translation application is the primary language tool. It focuses on the management of the gesture library and the graphical user interface of the language tool. The default gesture library supports the American Sign Language alphabet and numbers zero to nine. With the system capable of capturing one hand's motion and hand shape, the word bank available is extremely limited. Thus the default library provides only the alphanumeric characters.

## IV. Language Glove Hardware Detail

This section provides a summarized technical report of each component used in the language glove's final design. The following figure, Figure 4, shows how the components interact with each other.

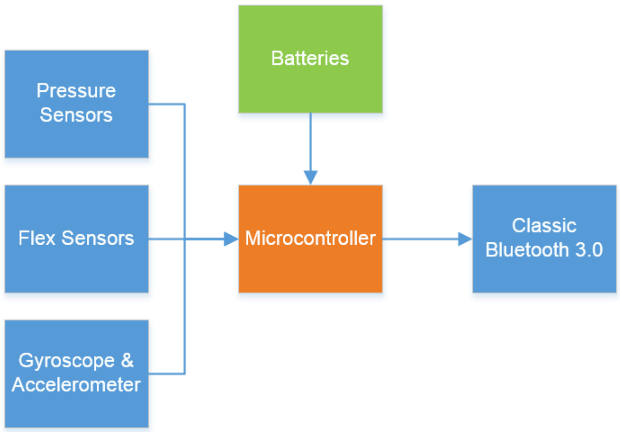



**Figure 4: Language Glove's Components**

*A. Microcontroller*

The microcontroller used in the language glove circuit is the Atmega328 from Atmel. The Atmega328 is a low power 8-bit microcontroller that features advanced RISC architecture. It executes powerful instructions in a single clock cycle and achieves throughputs of 1 MIPS per MHz, which allows the optimization of power consumption versus processing speed. This microcontroller is in charge of gathering the data not only from the flex sensors and pressure sensors but also from the gyroscope and accelerometer using I²C.

Furthermore, this microcontroller combines 32KB ISP flash memory with read-write capabilities; it has 23 general-purpose I/O lines, 32 general-purpose registers, three flexible timer/counters with compare modes, internal and external interrupts, and a serial programmable USART. Besides offering advance flash technology, this microcontroller provides binary code level compatibility and pin-to-pin compatibility with other devices.

The Atmega328 has the perfect balance between a low power consumption and performance, which is the main reason of why this microprocessor is used in this subsystem. Moreover, the Atmega328 has cross platform environments, where programs can be developed in Linux, Windows, and OSX environments, as well as having a boot loader easily available which helps ease sketching by allowing the update of new code.

*B. Flex Sensors*

The most common sensors on our language glove are the flex sensors. Flex sensors change their resistance depending on their radius of curvature. This feature can be used in a voltage divider to detect changes in flexion. The language glove has a total of 9 flex sensors: two on each finger and one on the thumb. The language glove uses the 2.2" flex sensors "**SEN-10264**" from sparkfun.com. According to the data sheets shown in Figure 5 and Figure 6, this flex sensor has a flat resistance of 25k Ohms, a resistance tolerance of ±30%, a power rating of 0.50 watts continuous, and a blend resistance range from 45k Ohms to 125k Ohms.

Mechanical Specifications

-Life Cycle: >1 million
-Height: ≤0.43mm (0.017")
-Temperature Range: -35°C to +80°C

**Figure 5: Flex Sensor's Mechanical Specifications**

Electrical Specifications

-Flat Resistance: 25K Ohms
-Resistance Tolerance: ±30%
-Bend Resistance Range: 45K to 125K Ohms (depending on bend radius)
-Power Rating : 0.50 Watts continuous. 1 Watt Peak

**Figure 6: Flex Sensor's Electrical Specifications**

*C. Pressure Sensors*

When making letters in ASL, there are several unique points in which contact is made. Subtle differences in hand flexion would make an appropriate interpretation difficult. By adding pressure sensors in these locations the glove may more accurately read the hand gestures. Like the flex sensors, pressure sensors change their resistance. When pressure is applied the resistance decreases. In the language glove, there are four pressure sensors: one between the index and middle fingers, and one at the tip of the index and middle fingers, and one at the tip of the thumb. The placement of each pressure sensor is significant distinguishing certain letters with these additional data values.

The "**Standard FLEXIFORCE Sensors: A201**" from Tekscan are used in the language glove. These types of sensors are a 3-pin male connector; it has dynamic range that can be modified by changing its drive voltage and adjusting the feedback resistor. Their response time is less than 5ms and they have an operating temperature range from -40°F - 140°F (-40°C - 60°C).

*D. Analog to Digital Converter*

The Atmega328 has a total of 23 GPIOs with 16 of those being analog input capable. The language glove requires 21 sensor pins between the accelerometer, gyroscope, flex and pressure sensors. Since there was a lack of pin availability for the Bluetooth module, it was desired to implement an external Analog to Digital Converter (ADC) in the language glove as well. The language glove uses the "**ADS7828**" from Texas Instruments. This ADC has 8 channels, I²C capability, and 12-bit resolution. It too is a low power device and operates between 2.7 and 5.5V. For this project the higher resolution and larger number of channels are desirable. Although this device has two address select lines, the 8 channels are more than enough to compensate.

*E. Accelerometer & Gyroscope*

For this project, the "**IMU Digital Combo Board -6 Degrees of Freedom**" from Sparkfun is used to detect to orientation of the hand in 3-D space. This breakout is composed of the ADXL345 accelerometer and the ITG-3200 MEMS gyroscope. This IMU uses I²C to communicate with a microcontroller and requires 3.3V input.

The ADXL345 is a low power, 3-axis accelerometer with resolution measurements up to ±16*g*. This device is suited for mobile device applications. It measures static acceleration of gravity and dynamic acceleration resulting from motion or shock. Moreover, the ADXL features low current use with 40 µA while in measurement mode and 0.1 µA while in standby. The low power mode enables intelligent motion-based power management with threshold sensing and active acceleration with extremely low power dissipation.

The ITG-3200 is a 3-axis digital gyroscope. It uses InvenSense's proprietary MEMS technology with vertically driven vibrating masses, which produce a low-cost motion sensor. The ITG-3200 features 16-bit analog-to-digital converters for digitizing the gyro ouputs, a user-selectable internal low pass filter bandwidth, as well as a fast I²C interface. This gyroscope has digital-output X-, Y-, and Z-axis angular rate sensors. It has a low 6.5mA operating current consumption for long battery life and a supply voltage range of 2.1V to 3.6V.

*F. Bluetooth Module*

Bluetooth is utilized for the language glove for mobility. Unlike Wi-Fi, which requires a demanding configuration of hardware, Bluetooth is on mobile phones and uses less power to conserve battery life. The larger range that Wi-Fi offers is not necessary. This language glove requires low energy communication and independent of stationary hardware.

Although the optimal solution for the language glove would be the Bluetooth Low Energy (BLE), this language glove uses classical Bluetooth. BLE was in the initial design, but due to the setbacks in shipment, the glove stepped down to Classic Bluetooth. Based on the requirements mentioned above, the HC-06 Bluetooth module also meets design requirements. This module operating voltage is 3.6 – 6V, as well as Bluetooth Specification V2.0+EDR compliant. It also has a working current of 40mA and a sleep current of less than 1mA. The HC-06 module is a class 2-type output device, which allows wireless connection to a computer.

The HC-06 Bluetooth module can readily pair with other Bluetooth devices, which in this case is an Android smart phone, following two main conditions: Their communication must be between master and slave and the password must be correct. The device is can only be used as a slave device which suits the needs of the design. The downside of this module is that the work mode of the Bluetooth cannot be modified and only few commands can be used such as changing the name of the Bluetooth, resetting the password, and resetting the baud rate.

*G. Power Supply*

The design of the power supply for the language glove consists of a Polymer Lithium Ion Battery and a Power Cell-LiPo Charger/Booster. This battery is extremely light weight. The cell outputs a 3.7V at 850 mAh. This type of battery features 2C continuous discharge, robust power source under extreme conditions, and excellent long-term self-discharge rates. The selected charger is a single cell LiPo boost converter to 3.3V and 5V and a micro-USB charger all in one.

The boost converter is based on the TPS61200 from Texas Instrument. It is a low input voltage synchronous boost converter; it outputs current up to 600 mA at 5V while using a single-cell Li-Ion battery and discharges it down to 2.5 V. With low load currents, the converter enters the Power Save mode to maintain a high efficiency over a wide load current range.

Furthermore, the micro-USB charger uses the MCP73831, management controller that allows charging 3.7V LiPo cells at a rate of 100mA. It utilizes a constant-current/ constant-voltage configuration. The constant-voltage regulation has four options: 4.20V, 4.35V, 4.40V, and 4.50V. The MCP73831 limits the charge current based on the die temperature during high power. This thermal regulation optimizes the charge cycle time while maintaining device reliability.

*H. Glove Aesthetics*

The gloves used for this project will be a tennis glove. We selected this type of glove since it provides moisture resistance, as well as it maintains the hand at a cool temperature and it has a comfortable fit. There will be two flex sensors attached to each finger and one on the thumb. The pressure sensors will be located as follows: one between the index and middle finger, one at the tip of the index and middle finger, and one at the tip of the thumb. Since the hand rests naturally in a mild curve, we believe that the best position to put the circuitry will be at the top of the hand. The sensors will be mounted inside on the top of each finger on the glove and thumb, as shown on Figure 8.

Besides the flex sensors and pressure sensors, on the master circuitry we will have the gyro/accelerometer, and the battery. The Bluetooth will be attached to a wristband to minimize PCB size and not impede hand gestures. The master circuitry will be sewed on to the glove. Through the micro-USB charger the battery is easily connected to the device and has no short term need to be removed.


**Figure 7. Mounting of Glove**

*I. Glove Printed Circuit Board*

The printed circuit board used in the language glove was order through Sunstone Circuits' website. Sunstone provides free CAD software for schematics and circuit board layout. Since the design needs to be small in order to be able to fit on the top of the hand and be lightweight, the printed circuit board has only 2 layers. Most of the components are surface mounted except for the Bluetooth module since it was too complex to recreate. The main reason for selecting this manufacturer was the build time in their Quickturn. Their build time is as fast as 24 hours, which allows the group to re-order the PCB if needed with a fast turnaround.

## V. SOFTWARE DETAIL

This section summarizes the software choices and details of the mobile application that supports the translations algorithm and the graphical user interface.

*A. Android Application*

Android is the chosen mobile environment for the application. This operating system is ideal for several reasons: there is significant amount of support and documentation and low-cost to develop in compared to its competitors, Windows and Macintosh; the group's prior experience lays mainly in Java; Android is the most popular mobile platform. All things considered, the language application is designed for Android devices. Although many devices support Android, the test environment includes only one device.

The software modules are tested with the Samsung Galaxy Note III provided by one of the group members. Performance with like Android devices and tablets is outside the test environment.

*B. Bluetooth*

Classic Bluetooth 3.0 is the means of establishing a connection between the language glove and the Android mobile application. Setup of the connection with the Bluetooth device must be simple for the user. Enabling on the Bluetooth adapter, discovering devices and finally connecting to the device may not be a familiar process to the user. Ideally, Bluetooth connection setup should be done without leaving the application itself.

The mobile device must also initiate the connection as a client when connecting to the glove. Within every Bluetooth connection, there are two roles: server and client. Looking at the flow of data, the glove sends a data stream to the Android application with the gesture values of each sensor. In this application, the mobile device acts as a client where it creates a client socket with its universally unique identifier, a distinguishing number to identify an application's Bluetooth service, while the glove opens a server socket listening for a connection request containing a valid UUID known to the server.

To implement the Bluetooth connection successfully in the hands of any user, the mobile application shall establish the connection with the click of a button.

Below is a figure representing the decision logic tree when establishing the Bluetooth connection with the language glove given that glove peripheral is on.

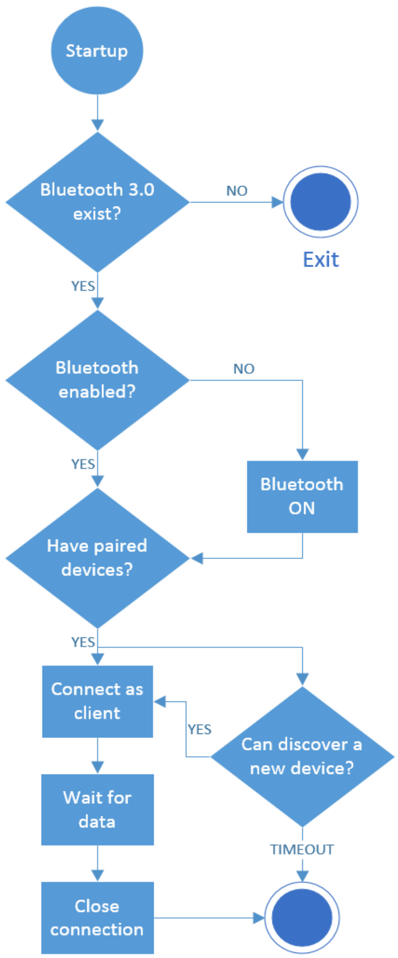



**Figure 8: Bluetooth Setup Decision Tree**

At the Startup, the user must give permission to utilize the Bluetooth device, but after granting the permission, the application may take the necessary steps to setup Bluetooth. The process may even query for previously paired devices or discover new discoverable devices. The setup may fail either by not having the appropriate hardware specifications, Classic Bluetooth 3.0, or there is no device available within the default time window of 120 seconds. In most cases, the Bluetooth connection shall be established first, ready for the user to translate hand gestures.

## VI. GESTURE CLASSIFIER ALGORITHM

The gesture translation faces a dilemma of associating message to the gesture. The glove defines the gesture by an array of data values representing the variety of sensor readings observed. These observed values are expected to vary with every gesture made as well as with every glove user. Considering the variances in the input data, the software solution must be robust enough to overlook these minor differences while being able to recognize the unique gestures. In these next sections, they shall discuss the process of learning and recognizing gestures.

### *A. Gesture Data Stream*

The gesture recognition algorithm starts with the data stream sent from the glove to the mobile application. It shall capture the sensor data at a fixed sample rate. With the server-client connection setup by Bluetooth, devices may only assume one role, and data flows in one direction from server to client device. Hence, the sampling rate is a fixed speed as a user to maintain a simple user friendly system.

When the Android's Bluetooth module receives the data stream, the information is raw and unprocessed by the glove. Due to the limited resources of the language glove, its sole responsibility is to read the sensor values. Then, forward it to the Android application, where it shall interpret the data. Within one complete sample cycle from the glove peripheral, the Android application shall see a data object.

In each cycle, the total flex and pressure sensors are divided between the two analog-to-digital converters; the inertial mass unit has two readings of the accelerometer and gyroscope. The IMU is read twice the sampling rate of the ADC's because movement requires a more precise measurement. With this data stream, a gesture model may be created from this to populate the library or translate gestures to a message.

### *B. Discrete Gesture Data*

This section shall discuss the process of identifying a gesture from the data stream. Before any analysis for a new or existing gesture, the system must know where to start looking within a potentially continuous stream. Looking at the data stream, there exist worthless values at the start of or in-between gestures. If the data is read directly from start to end, the noise surrounding the real gestures may delay the results. The algorithm must be able to differentiate the nonsensical data from the gestures themselves.

The solution looks at the end of each gesture. When fingerspelling a character, the user is expected to hold the gesture for roughly 0.5 seconds. By firmly holding the sign before spelling the next character, this creates a point where the hand system does not experience change. Knowing there are brief moments where there are no changes in the data allows the application to discover a valid starting point to analyze for an American Signed Language character.

*C. Gesture Data Model*

Given a valid starting point, a gesture data model is ready to be created. This model represents a collection of observed values. It shall duplicate the value set of the four inputs mentioned above from the data stream. One model start from one sampling cycle's result and send it to the classifier to determine the associated message. The process shall repeat looking for the next gesture with a new starting point until the connection closes.

*D. Gesture Analysis*

From each gesture data model, a gesture classifier is ready to identify if there is a recognizable gesture within the library. The analysis starts by determining how to interpret the expected values of each gesture. The glove has flex sensors to measure how flexed a user's fingers are. Out of the 36 characters that the application will support, 32 of them have unique hand shapes, which indicate that they have unique ranges for the flex sensor values. There are only 4 hand gestures that share the same hand shapes. Those letters are 'J' and 'Z', which have the same hand shapes as 'I' and '1' respectively, as shown in figure 7. The only way to differentiate between these 4 particular gestures is to also read in data from the accelerometer and gyroscope.

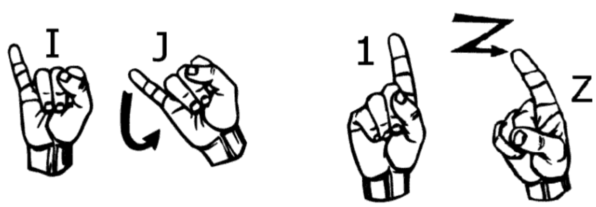



**Figure 9. Gestures with same hand shape**

Based on these observations, the group developed the approach for the deterministic algorithm. With the exception of the 4 characters that need the accelerometer and gyroscope values, all other gestures can be recognized just by looking at the flex and pressure sensors, which indicate the shape of the hand. The logic behind recognizing these 32 gestures is described below.

There is a set X, which is the set of all 32 hand gestures that have unique hand shapes. There is also a vector V, which has 13 elements; each element corresponds to the value of one of the flex and pressure sensors. Since each gesture in X corresponds to a unique hand shape, and since flex/pressure sensors indicate hand shape, this means each gesture has a unique range of values V can have. If a user is trying to gesture a character, then a vector V will be generated. There is a gesture library L which contains the range of vector values for each of the 36 characters. The library will take in the generated vector and see if it fits within the ranges of any of the 36 characters. If it does, then the output delivers the associate character. These gestures recognized by the algorithm uses a fixed library, discussed in the next section.

*E. Gesture Library*

This next section shall discuss the method behind the implementation of gesture library using boundaries. There shall be data models that represent the boundaries of each gesture. These boundaries capture the range of the data vector and shall be determined by a hand shape training algorithm. There will be an additional step in the algorithm for recognizing the characters 'J', 'I', '1', and 'Z'.

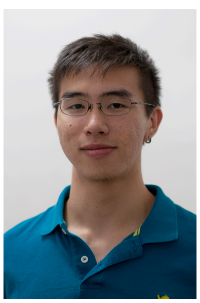
Within the Android application, there shall be training sessions for all 36 possible characters to train. When the session starts, the minimum and maximum of the flex and pressure sensors are defined by the trainer wearing the glove is making the gesture. If the trainer tries to create the model of the letter 'A', then he flexes his hand into the shape of the gesture for 'A'. At that point, he shall start the session to train the hand shape. The flex and pressure values are recorded at that point. The trainer shall adjust the gesture for a designated period to account for gesture variability. The minimum and maximum flex and pressure values are continuously being updated as the developer does this. Of the rare cases, the letters 'J' and 'Z' with similar hand shapes as 'I' and '1' shall require additional attention. These special models shall be introduced to orientation data. With the additional information, these characters may be distinguished. After the high and low boundaries have been set in place, the trainer can press "Stop Training" to complete the model for the character. There is also the option to clear out the training data and rebuild the gesture model, update the high and low thresholds by redoing the training for the character or even delete the entire library.

For the training of the 4 special characters that share the same hand shape, hand orientation is used to differentiate them. When training these gestures, the trainer would keep their hand still at the end of the gesture for about half a second. This allows the glove to recognize when the gesture is complete and obtain the additional orientation data. The 'J' and 'Z' have different orientations than 'I' and '1' at the end of the gesture. The 'J' ends with the palm facing the user's torso; the 'Z' ends with a tilted orientation facing away from the user and toward the ground. When the force of gravity acts on the accelerometer of the IMU, the orientation data produced shall distinguish between the special characters. This shall ultimately eliminate the overlap of common hand gestures once orientation is introduced.

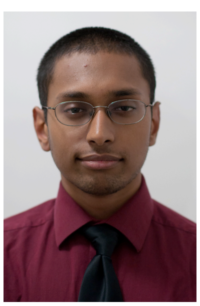
Once the training sessions are complete, the gesture models are associated with the character from the defined set of 36 characters. The library from the user's perspective is fixed and requires an update in the library from the developer to expand the supported gestures. The device and glove now are ready to translate messages.

## BIOGRAPHY

Laura Rubio-Perez is a 25-year old graduating Computer Engineering and Electrical Engineering student who is taking a job with Intel Corporation as a SSD Validation Intern. She will continue her graduate studies at Georgia Institute of Technology in the field of Computer Architecture in the fall of 2014. She is currently interested in the system architecture of computing devices, as well as the importance of performance's maximization versus power consumption in computing devices.



Kirk Chan is a 22-year old Computer Engineering graduate taking a job position as a Program Analyst Associate with FedEx Services after previously holding an internship position. His work will involve the development of Web Services Technologies. Graduate school is a possibility, but he currently wishes to pursue work experience before further advancing his education.



Ali Mizan is 22-year old graduating Computer Engineering student. He is going into the graduate program at UCF in fall 2014. He was a grader at UCF for Computer Organization and Computer Architecture. He also had an internship at Delphi Corporation in Troy Michigan.

Brian Troili is a 23-year old Electrical Engineering student

and hobbyist. He is pursing interests in computer communications over the summer of 2014, during which he will graduate. He held seasonal contract positions with several engineering firms. After graduating he wishes to pursue a career in peripheral equipment design.


## ACKNOWLEDGEMENT

The authors wish to acknowledge the assistance and support of Dr. S. Richie, as well as Boeing Corporation for its generous sponsorship.


## REFERENCES


[1] "8-bit Microcontroller with 4/8/16/32K Bytes In-System Programmable Flash ATmega 328," September 2013. Web. <http://www.atmel.com/Images/doc8161.pdf>

[2] "FlexiForce Standard Force & Load Sensors Model #A201," September 2013. Web. <http://www.tekscan.com/pdf/FLX-A201-force-sensor.pdf>

[3] "Flex Sensors Special edition Length," September 2013. Web. <https://www.sparkfun.com/datasheets/Sensors/Flex/flex22.pdf>

[4] "HC Serial Bluetooth Products User Instructional Manual," March 2014. Web. <http://www.exp-tech.de/service/datasheet/HC-Serial-Bluetooth-Products.pdf>

[5] InvenSense, "ITG-3200 3-Axis Gyro Evaluation Board," March 2014. Web. <http://dlnmh9ip6v2uc.cloudfront.net/datasheets/Sensors/Gyros/3-Axis/EB-ITG-3200-00-01.1.pdf>

[6] Analog Devices, "3-Axis, ±2 g/±4 g/±8 g/±16 g Digital Accelerometer," September 2013. Web. <https://www.sparkfun.com/datasheets/Sensors/Accelerometer/ADXL345.pdf>

[7] "12-Bit, 8-Channel Sampling ANALOG-TO-DIGITAL CONVERTER with I$^2$C Interface," November 2013. Web. http://www.ti.com/lit/ds/symlink/ads7828.pdf