SignIt

\*Make-A-Difference\*

By: Rouaa Diab

Meyar Shehab

Contents

[Background Information 3](#_Toc500111621)

[Purpose 3](#_Toc500111622)

[Goals 3](#_Toc500111623)

[System Design 3](#_Toc500111624)

[Components 4](#_Toc500111625)

[Arduino Micro 4](#_Toc500111626)

[Flex Sensors 5](#_Toc500111627)

[Pressure Sensors 5](#_Toc500111628)

[Emic-2 Module 6](#_Toc500111629)

[Accelerometer 6](#_Toc500111630)

[System Architecture 7](#_Toc500111631)

[Implementation 8](#_Toc500111632)

[Future plans 11](#_Toc500111633)

[Summary 11](#_Toc500111634)

[References 12](#_Toc500111635)

[Appendix A 13](#_Toc500111636)

[Appendix B 14](#_Toc500111637)

# Background Information

SignIt is a project that aims to help the deaf and mute community by translating the movements made by the dead/mute individual into spoken English. The latest prototype for this project was implemented at the Make-A-Difference workshop held at Helmut Schmidt University. The details of the project design and implementation will be provided in this document.

# Purpose

The aim of SignIt is to be able to allow deaf/mute individuals feel more understood by their communities and the people around them. We believe that Sign Language is simply just another language, not a disability. We hope that the individuals our project aims to help would actually benefit from it and that it would help them feel more included in their communities.

# Goals

* Be able to translate Sign Language effectively and efficiently.
* Allow our system to be used in different scenarios, portability.
* Allow for translation into speech.
* Allow for output to be given in real-time.

# System Design

After analyzing the different ways we could design our glove, such as making it completely software-based by using a technology the Kinect, or by the use of different sensors and circuit models, we have chosen to make a glove that integrates the uses of several diverse types of sensors.

The SignIt glove uses flex sensors, pressure sensors, an accelerometer, and a speech module. The processing of the results obtained through these sensors is done using the Arduino Micro, which typically uses the Arduino IDE for a coding environment.

We chose such a design in order to diversify the abilities of the components. The design is simple, yet dense, but it is meant to be budget-friendly. Each component used in the making of SignIt will be explained in upcoming section.

# Components

## Arduino Micro

The Arduino technology we decided to use is the Arduino Micro, due to its small size, but its convenient number of analog inputs/outputs. It has 12 analog pins, which is all what we need to be able to translate a large amount of gestures. Its size also makes it look presentable and light on the glove.



Figure . Arduino Micro

## Flex Sensors

Flex sensors are used for measuring the amount of bending of the fingers in contact with the sensor. Flex sensors can vary in terms of their range of resistance and their lengths. The size we chose was 5.588 cm, we chose our flex sensors to be this size because it covers the spectrum of the joints we need at this stage. For our current prototype, five flex sensors were being used.

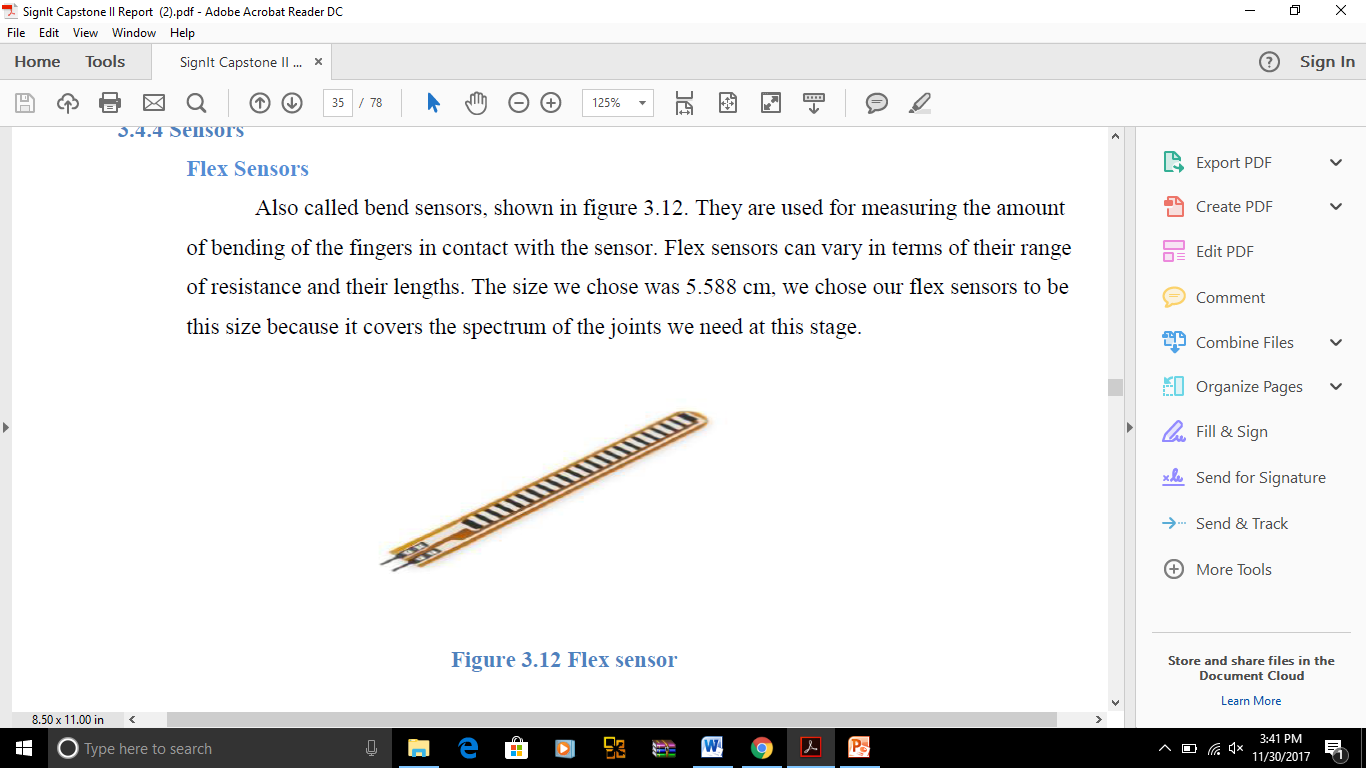


Figure .Flex Sensor

## Pressure Sensors

Pressure sensors are used to detect whether or not a certain finger has pressure being exerted on it; as well as how high the pressure being exerted on it. It is sensitive from 0.1N to 10N of pressure. It has a diameter of 4mm and an overall length of 4.445 cm; we chose these sizes because they are the most suitable for each finger’s width.

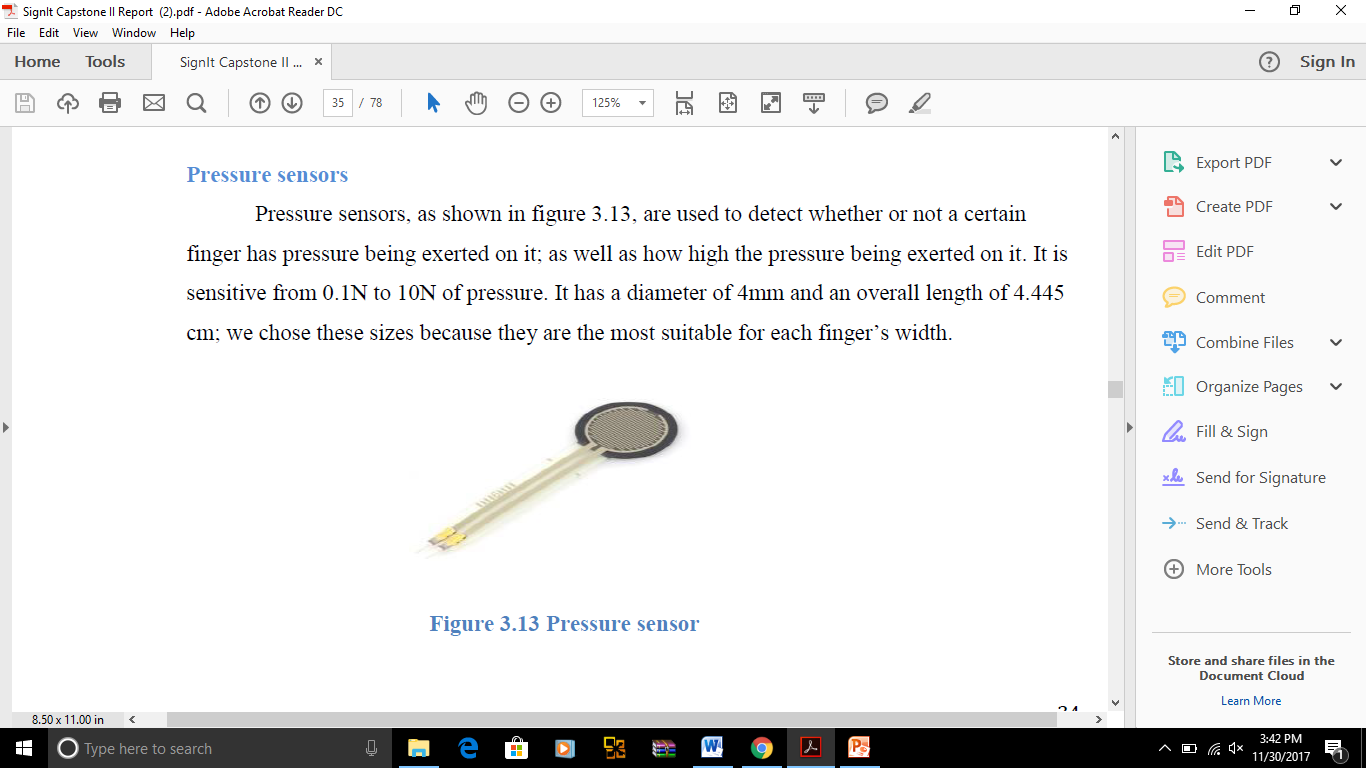


Figure .Pressure Sensor

## Emic-2 Module

The Emic-2 Module is a sophisticated device, which is used to output speech on a speaker connected to the Arduino. It has six different tones of voice, including male, female and child. Volume and speech pace can also be controlled within’ the Arduino code.



Figure 4.Emic-2 Module

## Accelerometer

The MMA8452Q is the accelerometer that we decided on using. It helps us determine the hand’s position within’ 3D axes.



Figure .Accelerometer

# System Architecture

The overall system architecture and the way the system functions are shown in the figures below. Please note that this is the simplified version, it eliminates the connections with the breadboard that was used.

The connections for the components used are as follows:

* Flex sensors each use a connection to ground, connection to Arduino analog pin, connection to voltage source, and a 10KΩ resistor in between all these connections.
* Pressure sensors each use a connection to ground, connection to Arduino analog pin, connection to voltage source, and a 3.3KΩ ohm resistor in between all these connections.
* Accelerometer uses a connection to the SDA and SCL pins of the Arduino, which are connected to the accelerometer’s corresponding pins. It also uses two 330Ω resistors, one between each of the connections between the Arduino and the accelerometer.
* Emic Module is directly connected to the Arduino.

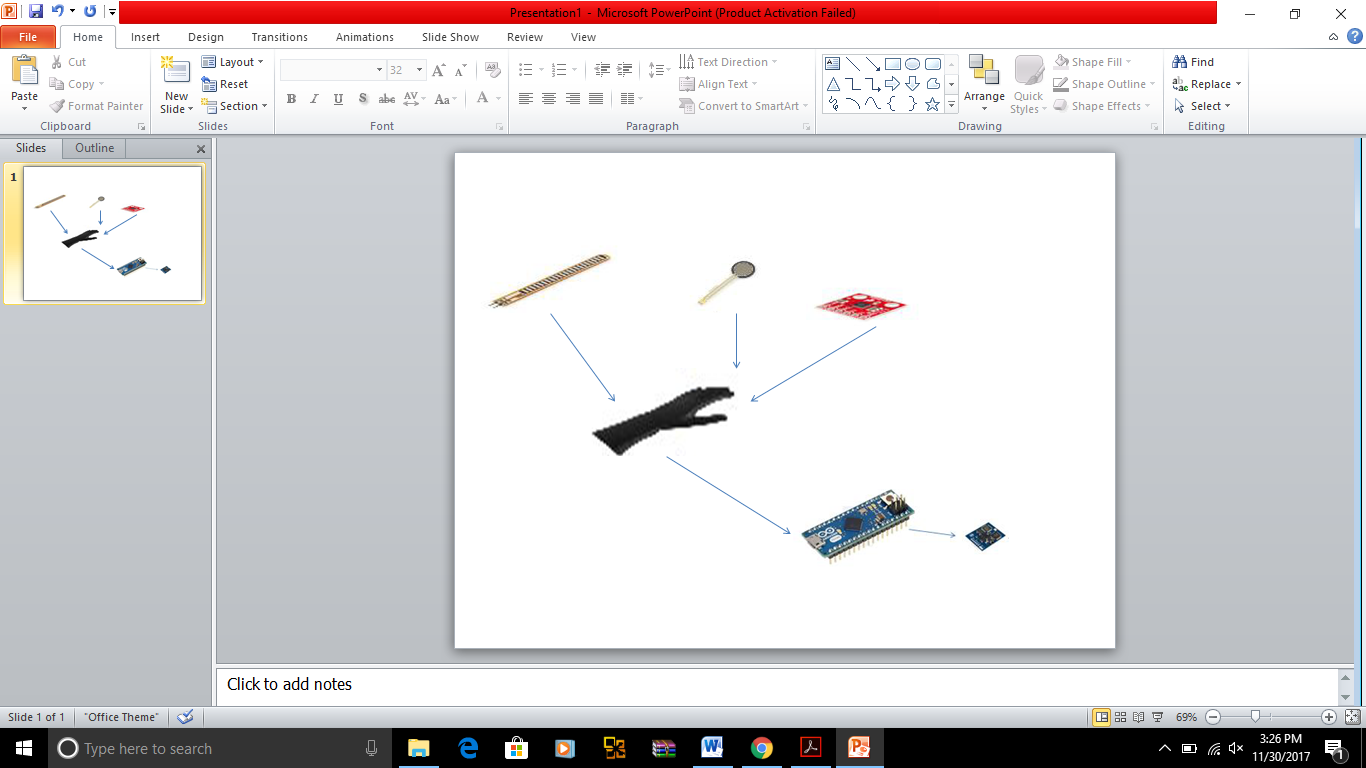


Figure .SignIt System Connections



Figure 7.SignIt Flow Diagram

# Implementation

The first stage of the implementation was to test the components that we chose to put on our glove and check to see if they perform the way we would like them to. After testing each component, we placed them on the glove gradually. The order of the readings taken and the component placement is as follows:

1. Flex sensors
2. Pressure sensors
3. Accelerometer
4. Emic-2 Module

The code is provided in the Appendix of this document, as well as the diagram to clarify the way the flex sensors were numbered in the code.

The latest prototype of SignIt is shown in the figure below. In this prototype we implemented two words, two phrases, and eight letters. We chose to implement these gestures in order to show the capabilities of the glove.



Figure .SignIt Prototype

The following is a table showing the gesture made and the types of components used to detect this movement accurately. It can be noted that with the addition of the more components, more movements can be detected.

|  |  |
| --- | --- |
| **Gesture** | **Sensors used** |
| Thank you | Flex sensors, accelerometer. |
| Hello | Flex sensors, accelerometer. |
| Sorry | Flex sensors, accelerometer. |
| I love you | Flex sensors. |
| D | Flex sensors. |
| E | Flex sensors, pressure sensors. |
| F | Flex sensors. |
| L | Flex sensors. |
| I | Flex sensors, accelerometer. |
| J | Flex sensors, accelerometer. |
| M | Flex sensors, pressure sensors. |
| N | Flex sensors, pressure sensors. |

The table shows the basic types of sensors/components used to detect the gestures, more of the sensors in our current implementation can be used in order to make the readings more accurate. For example, our accelerometer can be used in letters like D, E etc.

The final step in implementing the prototype was to print 3D components to place the mini circuit board in, as well as a wristband to keep the 3D component attached to the hand. The 3D components allow for the freedom of movement and flexibility.

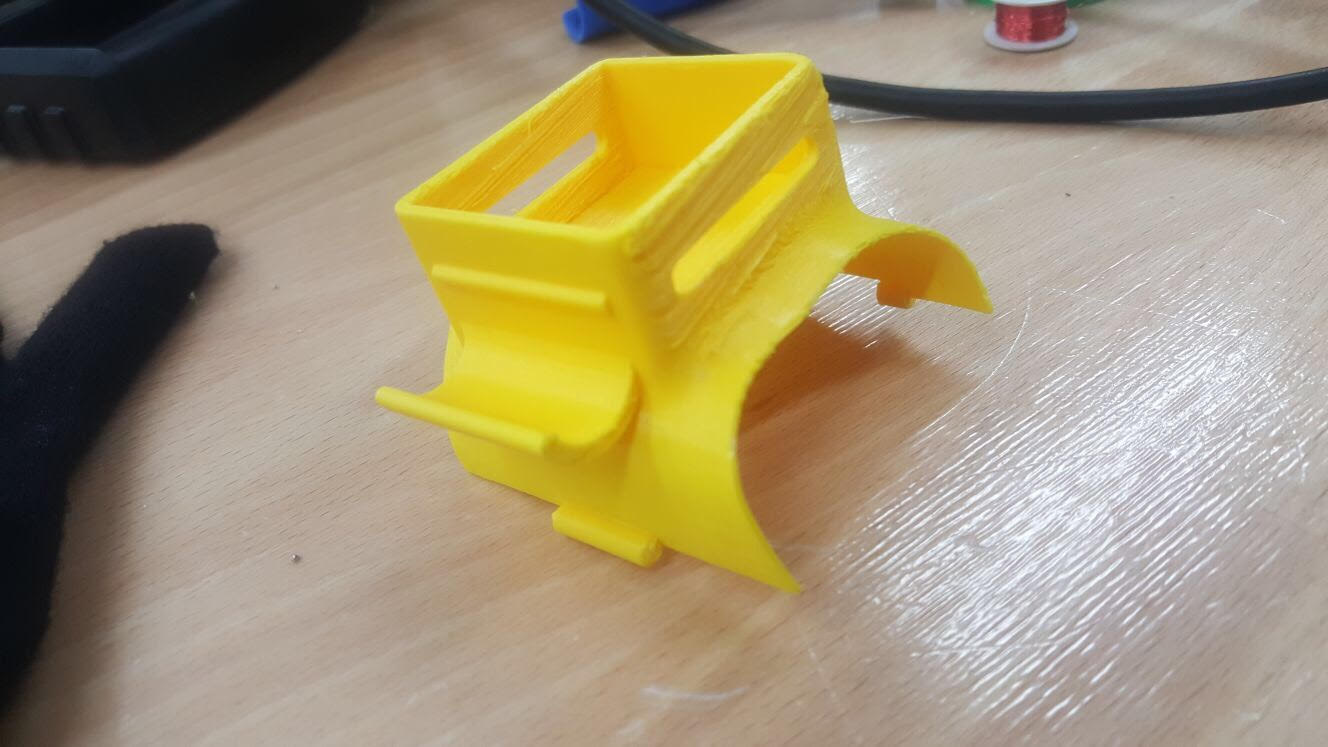
****

Figure . 3D Component

# Future plans

There are certain enhancements we would like to make in order to make SignIt a project usable by the deaf/mute community. Many words and phrases need to be added, perhaps in the form of a database. Another feature we would like to incorporate is the addition of an application, which will allow the user to control the settings for the Emic-2 Module, such as volume and voice option. We would like to add the mentioned features, but we will try to do so in a budget-friendly manner.

# Summary

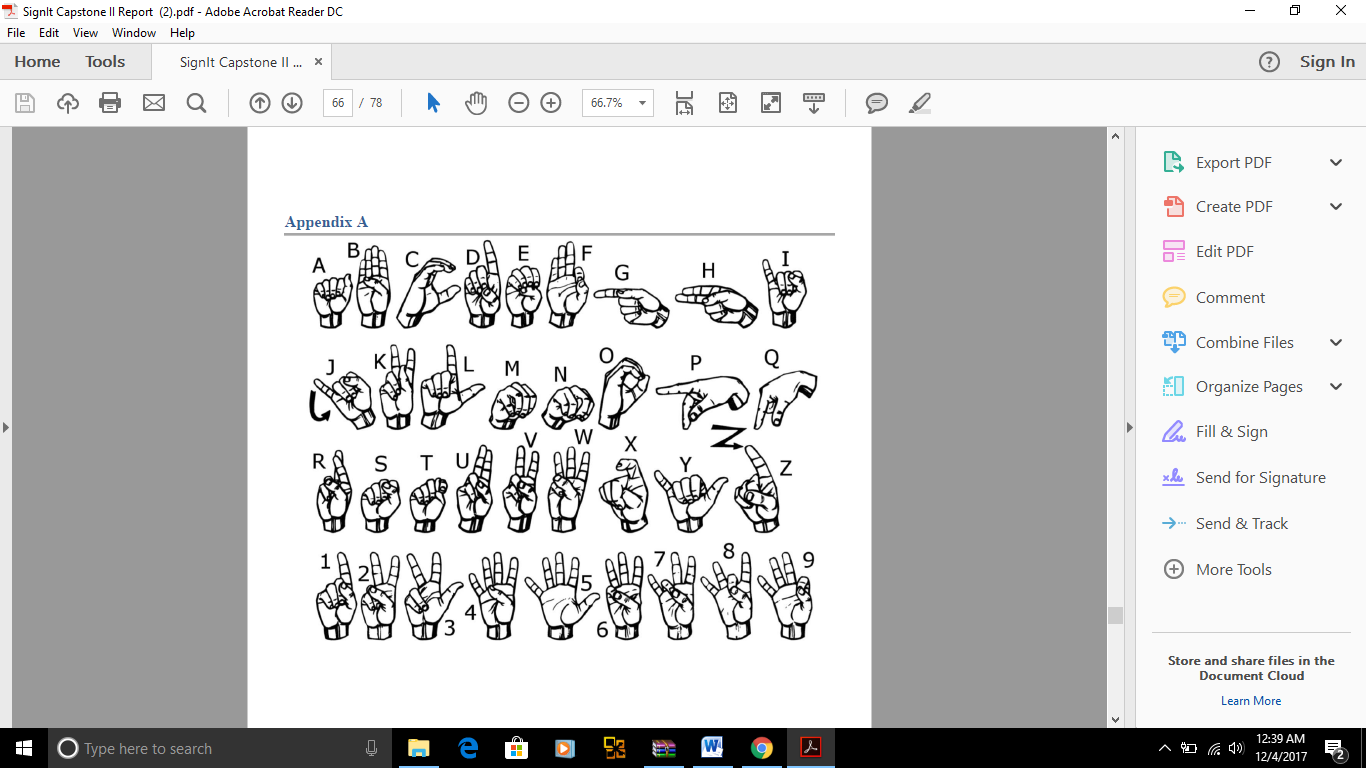
SignIt is a glove that uses diverse technologies in order to keep track of certain English sign language movements in order to translate them into English speech. Its latest prototype was documented in this report with the details of the stages and components used.

# References

[1] Arduino.cc. [Online]. Available: <https://www.arduino.cc/>.

[2] SparkFun. [Online]. Available: <https://www.sparkfun.com/>.

# Appendix A



# Appendix B

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* \*/

/\* Project: SignIt \*/

/\* Developers: Rouaa Diab, Meyar Shehab \*/

/\* Version: 1.0 \*/

/\* Purpose: Translate sign language movements into English speech. \*/

/\* \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*libraries\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <Wire.h> // Must include Wire library for I2C

#include <SFE\_MMA8452Q.h> // Includes the SFE\_MMA8452Q library

#include <SD.h> // Needed by the EMIC2 library, though not utilized in this example

#include "EMIC2.h"

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*definitions\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

//accelerometer

MMA8452Q accel;

double xaxis,yaxis,zaxis;

//emic2

#define RX\_PIN 10 // Connect SOUT pin of the Emic 2 module to the RX pin

#define TX\_PIN 11 // Connect SIN pin of the Emic 2 module to the TX pin

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*globals\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

EMIC2 emic; // Creates an instance of the EMIC2 library

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*function prototypes\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void setup(void);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void loop()

{

accel.read();

xaxis = accel.cx;

Serial.println(xaxis);

yaxis = accel.cy;

Serial.println(yaxis);

zaxis = accel.cz;

Serial.println(zaxis);

unsigned short flexThumb = 0,

flex2 = 0,

flex3 = 0,

flex4 = 0,

flex5 = 0,

pressureN = 0,

pressureM = 0;

flexThumb = analogRead(0);

flex2 = analogRead(1);

flex3 = analogRead(2);

flex4 = analogRead(3);

flex5 = analogRead(4);

pressureN = analogRead(5);

pressureM = analogRead(6);

Serial.print("Thumb: ");

Serial.println(flexThumb,DEC);

Serial.print("Flex2: ");

Serial.println(flex2,DEC);

Serial.print("Flex3: ");

Serial.println(flex3,DEC);

Serial.print("Flex4: ");

Serial.println(flex4,DEC);

Serial.print("Flex5: ");

Serial.println(flex5,DEC);

Serial.print("PressureN: ");

Serial.println(pressureN,DEC);

Serial.print("PressureM: ");

Serial.println(pressureM,DEC);

if(flexThumb<850 && flex3<810 && flex4<810 && flex5<810 && xaxis>0 && yaxis<0 && zaxis>0 ){

emic.speak("Thank you");

Serial.println("Thank you");}

else if(flexThumb>900 && flex3<810 && flex4<810 && flex5<810 && xaxis>0 && yaxis<0 && zaxis<0.2 ){

emic.speak("Hello");

Serial.println("Hello");}

else if(flex2>850 && flex3>820 && flex4>820 && flex5>820 && xaxis>0 && yaxis<0 && zaxis<0.1 ){

emic.speak("Sorry");

Serial.println("Sorry");}

else if(flexThumb<850 && flex2<820 && flex3>820 && flex4>800 && flex5<820 ){

emic.speak("I love you");

Serial.println("I love you");}

else if(flexThumb>850 && flex2<850 && flex3>850 && flex4>850 && flex5>850 && pressureM > 1000 && pressureN>1000){

emic.speak("D");

Serial.println("D");}

else if(flexThumb>850 && flex2>850 && flex3>850 && flex4>850 && flex5>850 && pressureM>1000 && pressureM>1000 ){

emic.speak("E");

Serial.println("E");}

else if(flexThumb>850 && flex2>850 && flex3<850 && flex4<850 && flex5<850 ){

emic.speak("F");

Serial.println("F");}

else if(flexThumb<850 && flex2<850 && flex3>850 && flex4>850 && flex5>850 ){

emic.speak("L");

Serial.println("L");}

else if(flexThumb>850 && flex2>850 && flex3>850 && flex4>850 && flex5<850 && xaxis > 0 && yaxis>0 && zaxis>0 ){

emic.speak("I");

Serial.println("I");}

else if(flexThumb>850 && flex2>850 && flex3>850 && flex4>850 && flex5<850 && xaxis > 0 && yaxis<0 && zaxis>0 ){

emic.speak("J");

Serial.println("J");}

else if(flexThumb>820 && flex2>820 && flex3>820 && flex4>820 && flex5>820 && pressureM<1000 ){

emic.speak("M");

Serial.println("M");}

else if(flexThumb>820 && flex2>820 && flex3>820 && flex4>820 && flex5>820 && pressureN<1000){

emic.speak("N");

Serial.println("N");}

else

Serial.println("No output");

delay(2000);

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*setup function\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void setup()

{

// initialize serial communications

Serial.begin(9600);

accel.init();

emic.begin(RX\_PIN, TX\_PIN);

emic.setVolume(18);

emic.setVoice(2); // Sets the voice (9 choices: 0 - 8)

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/