TRIBHUVAN UNIVERSITY

INSTITUDE OF ENGINEERING

Kathmandu Engineering College, Kalimati

Major Project Report

on

SMART GLOVES USING FLEX SENSORS

By:

Bipin Godar (75005)

Niraj Kushwaha (75014)

To:

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

KATHMANDU ENGINEERING COLLEGE

Kathmandu, Nepal

Jestha, 2079

ACKNOWLEDGEMENT

First and foremost, we offer our sincerest gratitude towards the **Department of Electronics Communication, and Information Engineering, Kathmandu Engineering College, Kalimati** and **Institute of Engineering (IOE)** for providing us the wonderful opportunity to undertake this project. We would like to show our gratitude to our supervisor, **Er. Suramya Sharma Dahal** for guiding us through this project. We would also like to show our gratitude to our Head of Department of Electronics and Communication, **Er. Rajan Lama** for providing us this opportunity to showcase our major project. We extend our gratitude to our Major Project Coordinator, **Er. Dipen Manandhar** for his support and helping us to tackle different problems faced in this project.

We would also like to send our regards to faculty members, teachers and individuals who helped us with their support and suggestions for the making of this report.

ABSTRACT

According to World Health Organization (WHO), about 5%, approximately 70 million, of the world's population are mute and deaf and only a faction from this number know how to communicate well using correct sign language. Sign language is a method of non-verbal communication that is used by deaf and dumb people. Normal people do not learn sign language. The problem arises here and this problem becomes a barrier between them. So, we have suggested a wearable glove for people who face difficulty in communicating verbally due to various different reasons (be it deaf or dumb), so that with the possession of this device, they can exhibit their basic requirements via their gestures and those gestures will be converted to speech for the hearer to understand what is he or she trying to say. This device is constructed by mounting multiple flex sensors on a glove and connecting them to an Arduino. Depending on different hand signs the resistance value throughout the flex sensor changes and certain messages are shown and read aloud using a speaker.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATION

AAC	: Augmentative and Alternative communication
AC	: Alternate Current
ADC	: Analog to Digital Conversion
ANN	: Artificial Neural Network
ARQ	: Automatic Repeat Request
CSV	: Comma Separated Values
DC	: Direct Current
DOF	: Depth of Field
DTMF	: Dual Tone Multi Frequency
EMG	: Electromyography
FPGA	: Field Programmable Gate Array
LED	: Light Emitting Diode
MEMS	: Micro electromechanical System
MSA	: Mathematical Sound Architecture
PIC	: Peripheral Interface Controller

Chapter 1: INTRODUCTION

1.1 Background

A human being is a social animal and has the natural ability to see, listen, speak and interact with the external environment. Unfortunately, there are some people who do not have the ability to interact by speaking. The deaf and dumb population is a result of the physical disability of hearing and speaking. In the recent years, there has been a rapid increase in the number of hearing impaired and speech disabled victims due to birth defects, oral diseases and accidents. When a speech impaired person speaks to a normal person, the normal person finds it difficult to understand and asks the deaf-dumb person to show gestures for his/her needs. Dumb persons have their own language to communicate with us; the only thing is we need a translator in between.

Sign language is used by deaf and mute people and it is a communication skill that uses gestures instead of sound to convey meaning simultaneously combining hand shapes, orientation and movement of the hands, arms or body and facial expressions to express fluidly a speaker's thoughts. But most of the time normal people find it difficult to understand this sign language. This presents a major roadblock for people in the deaf and dumb communities when they try to engage in interaction with others, especially in their educational, social and professional environments. Therefore, it is necessary to have an advance gesture recognition or sign language detection system to bridge this communication gap.

The people who cannot speak or have lost their ability to speak in some accident, it becomes difficult for them to convey their message within the society. To overcome this, a project called 'SMART GLOVE' has been designed. Giving a voice to the voiceless has been a cause that many have championed throughout history, but it's safe to say that none of those efforts involved packing a bunch of sensors into a glove. The main objective of this project is to help deaf and dumb people by removing communication barrier so they are not restricted in a small social circle and are able to convey their feelings and emotions whenever they want.

Smart glove is based on the wearable technology. It is basically a device which has some specific wearable sensors with phenomenal temperature stability. All the sensors are fitted on a glove which measures the different analog parameters associated with the movement of fingers and orientation of the hand during any particular gesture. These sensors read those particular analog values and coding is done in the microcontroller according to these values to recognize the corresponding sign language. The goal of this project is to develop a portable communication system having multiple sensors for Sign Language Recognition and to translate these gestures into text and sound.

1.2 Problem Statement

Deaf and normal person communication is as same as two different persons from different countries using two different languages for communication without any common language which leads to problem in communication. Sign language is the only communication tool used by deaf people to communicate to each other. However, normal people do not understand sign language and this creates a large communication barrier between deaf people and normal people. In addition, the sign language is also not easy to learn due to its natural differences in sentence structure and grammar. Therefore, there is a need to develop a system which can help in translating the sign language into text and voice in order to ensure the effective communication can easily take place in the community.

1.3 Objective

The objectives of the project are

- To build a glove embedded with sensors to read the sign language and convert it into text and speech.
- Help to deaf and dumb people to communicate with normal people especially during emergency situation.

1.4 Scope or Application

The scope or application of the project are

- For all deaf and dumb people
- Institution for deaf and dumb people

1.5 Organization of report

Chapter 1 deals with the introductory part of this project report. It deals with the background, objectives, scope and application of this project along with the problem statement of the project. Chapter 2 deals with the literature review that describes the past works that were undertaken related to this project and also the components that were used in the past. Chapter 3 deals with the conceptual design and outline of the project. Chapter 4 explains the system block diagram, algorithm and flowchart. It gives detailed explanation of the methods and steps in making of our project both in terms of hardware and software. And finally, chapter 5 includes the work progress; work completed till now and remaining work.

Chapter 2: LITERATURE REVIEW

Enable Talk is a student project, whose main idea is to translate sign language into speech. The project was presented at the Microsoft Imagine Cup competition in 2012 at Sydney, Australia and won the first prize for software design competition [1]. The team was from country Ukraine with city Donetsk and school Computer Academy Step. The concept of the project consisted of two sensor embedded gloves and a mobile device, which entailed the recognition process.

Glove- based system is composed of an array of sensor, electronics for data acquisition or processing, power supply & a support for sensors that can be worn on user's hand [2]. LED glove, data glove, Sayre glove, cyber glove are the different types of gloves used here. Glove based system helps user for selecting a particular glove for particular application.

Glove Talk II is a system which translates hand gestures to speech, which is based on the gesture to format model developed by Sidney Fels and Geoffrey Hinton, Department of Computer Science of University of Toronto [3]. Neural networks were used to implement an adaptive interface, called Glove Talk II, which contains hand gestures to control the parameters of a parallel format speech synthesizer to allow a user to speak with his hands. It is used to implement an artificial vocal tract. Glove-Talk-II is a system which translates hand gestures to speech through an adaptive interface. Hand gestures are mapped continuously to 10 control parameters of a parallel format speech synthesizer. The mapping allows the hand to act as an artificial vocal tract that produces speech in real time. This gives an unlimited vocabulary, multiple languages in addition to direct control of fundamental frequency and volume. Currently, the best version of Glove-Talk II uses several input devices (including a Cyberglove, a Contact glove, a polhemus sensor, and a foot-pedal), a parallel formant speech synthesizer and 3 neural networks [4]. The gesture to speech task is divided into vowel and consonant production by using a gating network to weight the outputs of a vowel and a consonant neural network. The gating network and the consonant network are trained with examples from the user. The vowel network implements a fixed, user-defined relationship between hand-position and vowel sound and does not require any training examples from the user. Volume, fundamental frequency and stop consonants are produced with a fixed mapping from input devices.

Bend sensor modeling us used for motion recognition. The model is used to track human joint movement and it recovers the original signal waveforms, which shows the joint rotation for the fastest human speed. Bend sensor modeling is demonstrated that bend sensor can be applied for human posture recognition.

Harneet Kaur, et al. in their paper, presented a brief description about the past attempts that were made to convert sing language to understandable form. In their paper, they have thoroughly scrutinized the previous attempts over this technology and also suggested various possible ways to implement the design of a simple smart glove [5].

Speak jet is sound synthesizer which is used to convert text data into voice [6]. It uses mathematical Sound Architecture technique to control five channel sound synthesizers to generate a speech signal. It is having 72 speech elements, 43 sound effects and 12 DTMF touch tones by using MSA component and also pitch, rate, bend and volume parameter user can generate various sound effects. They tried to develop Electronic Speaking Glove, designed to facilitate an easy communication through synthesized speech for the benefit of speechless patients. Generally, a speechless person communicates through sing language which is not understood by the majority of people. The proposed system is designed to solve this problem. Gestures of fingers of a user of this glove will be converted into synthesized speech to convey and audible message to others. For example, in a critical communication with doctors. The glove is internally equipped with multiple flex sensors that are made up of "bend-sensitive resistance elements". For each specific gesture internal flex sensors produce a proportional change in resistance of various elements. The processing of this information sends a unique set of signals to the PIC microcontroller and speaks jet IC which is pre-programmed to speak desired sentences.

In a P5 Glove from Essential reality was used. It is an inexpensive (~50 Euro) glove with integrated 6 DOF tracking designed as a game controller [7]. 6 DOF means six degrees of freedom, in fact the ability to move forward/backward, up/down, left/right (translation in three perpendicular axes) combined with rotation about three perpendicular axes (pitch, yaw, roll). The glove consists of five bend sensors to track the flexion of the wearer's fingers. An infrared-based optical tracking system is used to compute the glove position and orientation without the need for additional hardware. The glove is connected with a cable to the base station.

Tushar Chouhan et al. implemented wired interactive glove, interfaced with a computer running MATLAB or Octave, with a high degree of accuracy for gesture recognition [8]. The glove maps the orientation of the hands and fingers with the help of bend sensors, Hall Effect sensors and an accelerometer. The data is then transmitted to a computer using automatic repeat request (ARQ) as an error controlling scheme. The system is modelled for the differently abled section of the society to help convert sign language to a more human understandable form such as textual messages. The hardware section of their proposed design has its constituent electronic components as bend sensor, hall-effect sensor, accelerometer and Machine Learning Algorithms used for Gesture Recognition. The bend sensors outputs are fed to the analog multiplexer (HEF4051B by NXP Semiconductors). The output of this multiplexer is given to a current to voltage converter circuit. Since the voltage output of the Hall sensor is low, an amplifier is needed. Sensor outputs obtained are given to the inbuilt ADC (analog to digital converter) of MSP430G25553(by Texas Instruments) for sampling the values given by the sensors, which is also used for interfacing the glove with a computer running the machine learning algorithms. The data acquisition process starts with the processor sending control signals to multiplexer for receiving values from the different sensors sequentially and temporarily storing it in an array. These stored values from the different sensors sequentially and temporarily storing it in an array [9].

Chapter 3: RELATED THEORY

3.1 Machine Learning

Machine learning is a branch of artificial intelligence (AI) and computer science which focuses on the use of data and algorithms to imitate the way that humans learn, gradually improving its accuracy.

Machine learning is an important component of the growing field of data science. Through the use of statistical methods, algorithms are trained to make classifications or predictions, uncovering key insights within data mining projects. These insights subsequently drive decision making within applications and businesses, ideally impacting key growth metrics. As big data continues to expand and grow, the market demand for data scientists will increase, requiring them to assist in the identification of the most relevant business questions and subsequently the data to answer them.

The typical supervised machine learning algorithm consists of roughly three components:

1. **A decision process:** A recipe of calculations or other steps that takes in the data and “guesses” what kind of pattern your algorithm is looking to find.
2. **An error function:** A method of measuring how good the guess was by comparing it to known examples (when they are available). Did the decision process get it right? If not, how do you quantify “how bad” the miss was?
3. **An updating or optimization process:** A method in which the algorithm looks at the miss and then updates how the decision process comes to the final decision, so next time the miss won’t be as great.

Machine learning uses two types of techniques;

1. **Supervised Learning**
   The dataset being used has been pre-labeled and classified by users to allow the algorithm to see how accurate its performance is.
2. **Unsupervised Learning**
   The raw dataset being used is unlabeled and an algorithm identifies patterns and relationships within the data without help from users.

3.2 Random forest classification

Random forest or random decision forests are an ensemble learning method or supervised classification algorithm for classification, regression and other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees [9]. This algorithm creates a forest with a number of trees and the more trees in the forest the more robust the forest looks like. Random decision forests correct the decision trees' habit of over fitting to their training set. Ensemble algorithms are those, which combine more than one algorithm of same or different kind for classifying objects [11]. For example, running a prediction over Naive Bayes, SVM and Decision Tree and then taking vote for final consideration of class for test object.

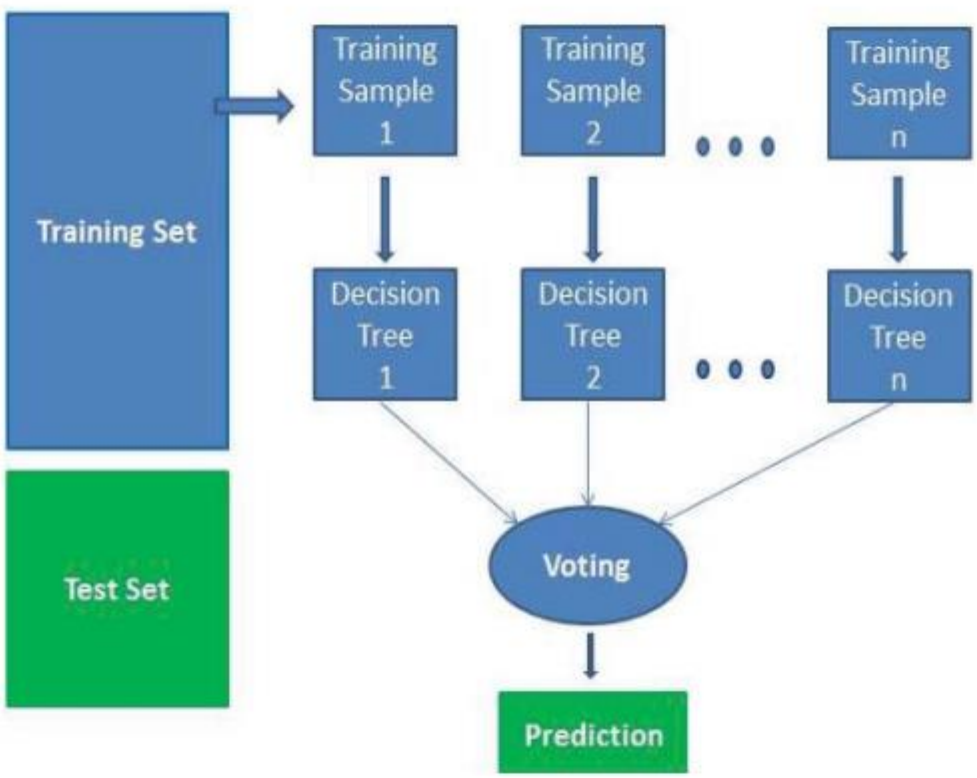

Figure 3.1: Working of Random Forest algorithm

Similarly, in the random forest classifier, higher number of trees in the forest results in higher accuracy of the outputs. Random forest classifier creates a set of decision trees from randomly selected subset of training set and then it aggregates the votes from different decision trees to decide the final class of the test object.

3.3 Arduino Mega

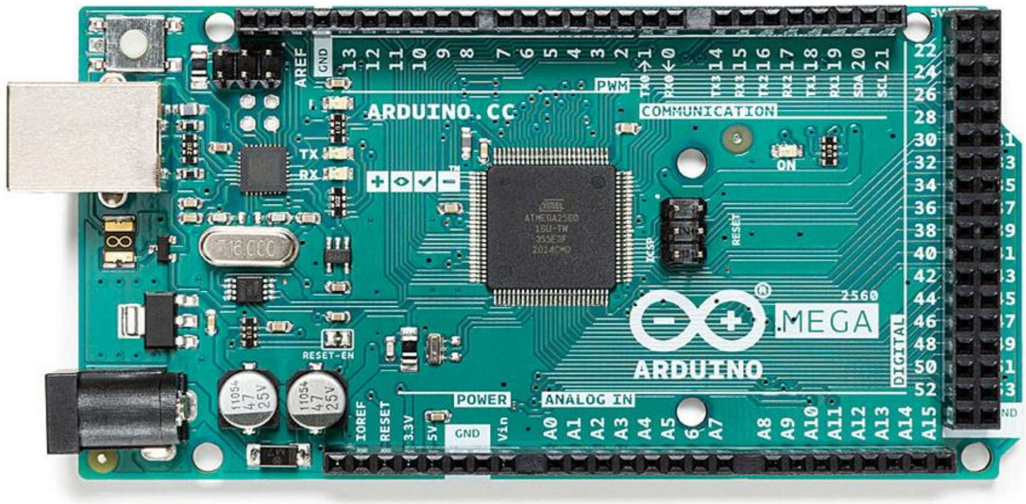

Figure 3.2: Arduino Mega 2560

The Arduino Mega 2560 is a microcontroller board based on the ATmega2560. It has 54 digital input/output pins (of which 15 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller.

Table 3.1: Tech Specifications

MICROCONTROLLER	ATmega2560
OPERATING VOLTAGE	5V
INPUT VOLTAGE (RECOMMENDED)	7-12V
INPUT VOLTAGE (LIMIT)	6-20V
DIGITAL I/O PINS	54 (of which 15 provide PWM output)
ANALOG INPUT PINS	16
DC CURRENT PER I/O PIN	20 mA
DC CURRENT FOR 3.3V PIN	50 mA

FLASH MEMORY	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
CLOCK SPEED	16 MHz
LED_BUILTIN	13
LENGTH	101.52 mm
WIDTH	53.3 mm
WEIGHT	37 g

3.4 Flex sensors

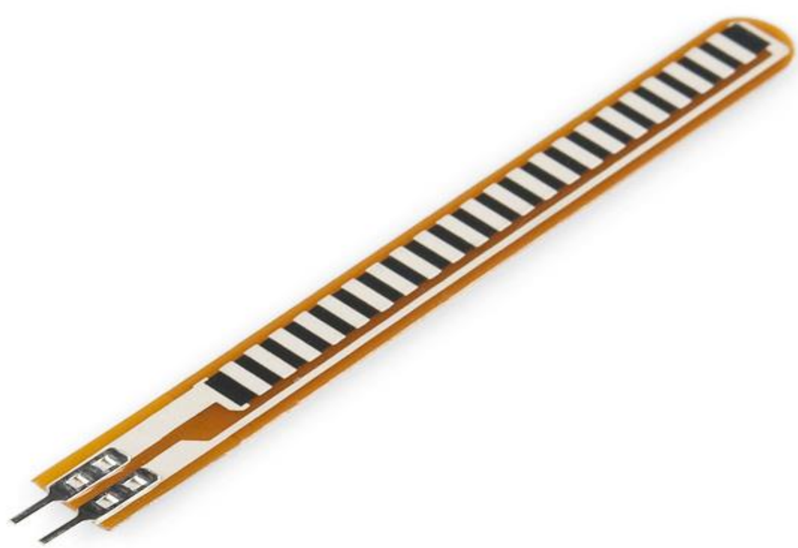

Figure 4.3: Flex sensor

Flex sensors are sensors that changes the resistance depending upon the amount of bend on the sensor. They convert the change in bend to electrical resistance-the more the bend, the more the resistance value. The designing of this sensor can be done by using materials like plastic and carbon. The carbon surface is arranged on a plastic strip as this strip is turned aside then the sensor's resistance will be changed. Thus, it is also named a bend sensor.

These sensors are classified into two types based on its size namely 2.2-inch flex sensor & 4.5-inch flex sensor. The size, as well as the resistance of these sensors, is dissimilar except the working principle. Therefore, the suitable size can be preferred based on the necessity. This type of sensor is used in various applications like computer interface, rehabilitation, servo motor control, security system, music interface, intensity control, and wherever the consumer needs to modify the resistance throughout bending.

The features of this sensor are:

- Operating voltage of this sensor ranges from 0V to 5V
- It can function on low-voltages.
- Power rating is 1 Watt for peak & 0.5Watt for continuous.
- Operating temperature ranges from -45ºC to +80ºC
- Flat resistance is 25K Ω
- The tolerance of resistance will be ±30%
- The range of bend resistance will range from45K -125K Ohms

3.5 MPU-6050

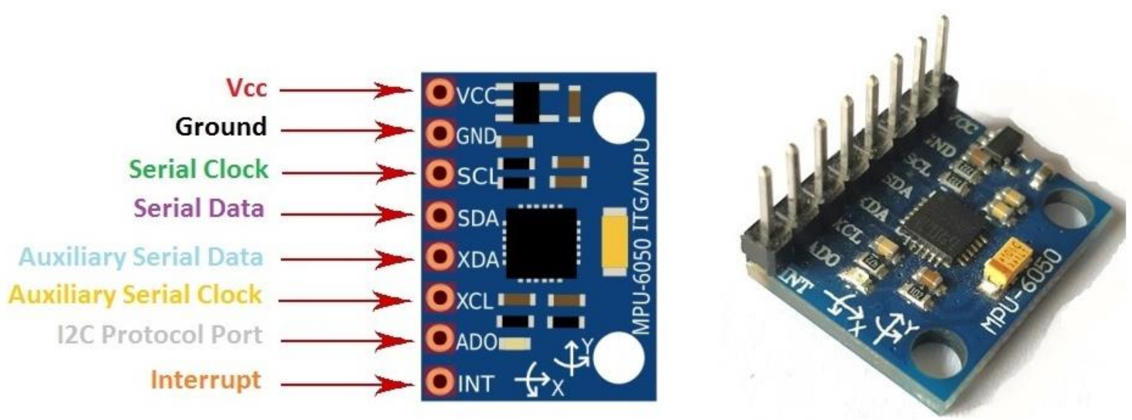

Figure 3.4: MPU-6050 Module

MPU-6050 is basically a sensor for motion processing devices. It is the world first six-dimension motions tracking device. It was designed for low cost and high performances smartphones, tablets and wearable sensor. It is capable of processing nine-axis algorithms, it captures motion in X, Y and Z axis at the same time. It is used in different industrial projects and electronic devices to control and detect the 3-D motion of different objects. It consists of three-axis **accelerometer** and three-axis **gyroscope**. It helps us to measure velocity, orientation, acceleration, displacement and other motion like features. The MPU-6050 is a 6 DOF (degrees of freedom) or six-axis IMU sensor, which means it gives six values as output: three values from the accelerometer and three from the gyroscope. It is a sensor based on MEMS (micro electro mechanical systems) technology. Both the accelerometer and the gyroscope are embedded inside a single chip.

Table 3.2

MPU-6050 Pinout		
Pin#	**Pin Name**	**Description**
01	VCC	This pin used for Supply Voltage. Its input voltage is +3 to +5V.
02	GND	This pin use for ground
03	SCL	This pin is used for clock pulse for I2C compunction
04	SDA	This pin is used for transferring of data through I2C communication.
05	Auxiliary Serial Data (XDA)	It can be used for other interfaced other I2C module with MPU6050.
06	Auxiliary Serial Clock (XCL)	It can also be used for other interfaced other I2C module with MPU6050.
07	AD0	If more than one MPU6050 is used a single MCU, then this pin can be used to vary the address.
08	interrupt (int)	This pin is used to indicate that data is available for MCU to read.

3.5.1 Accelerometer

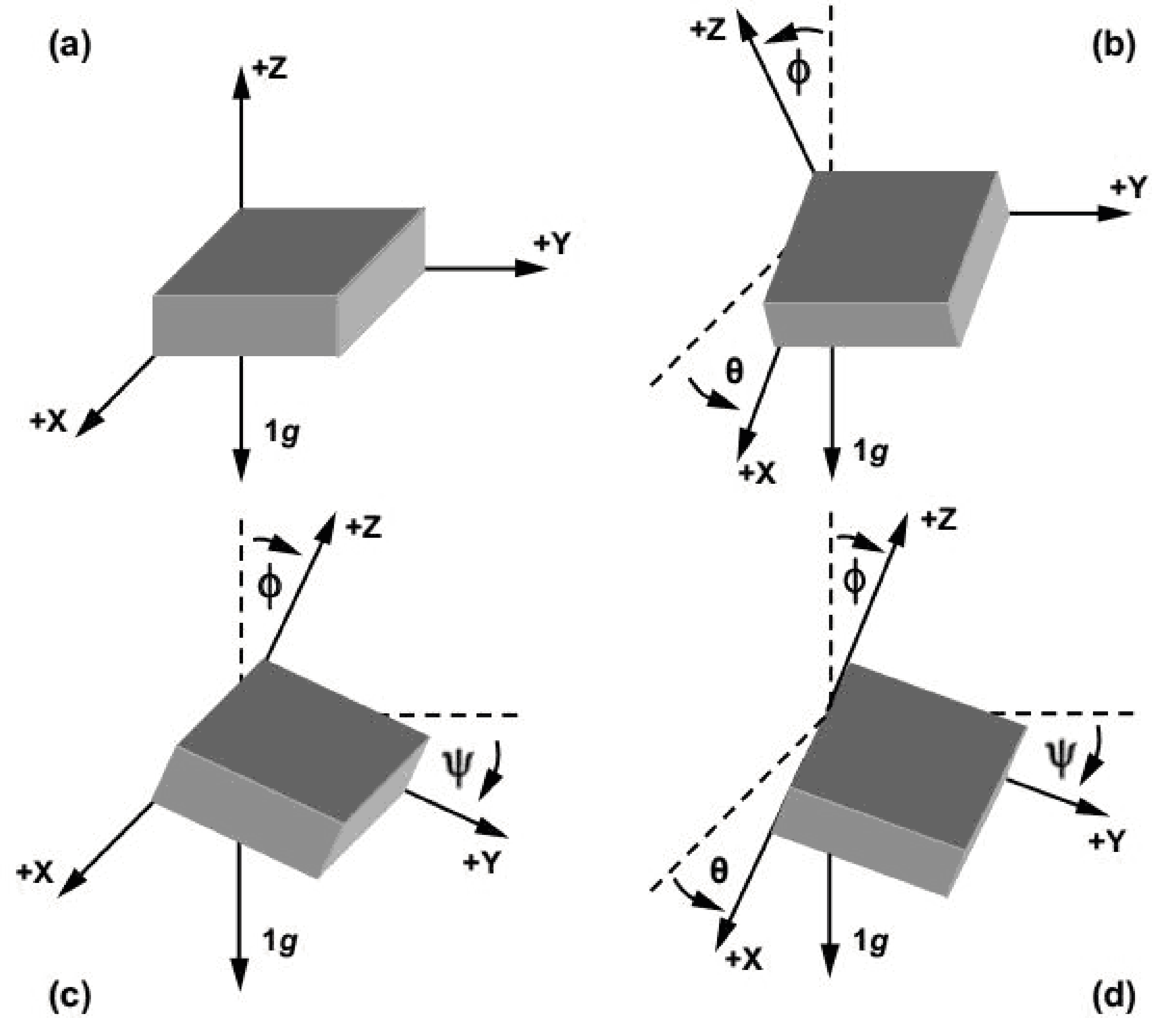

Figure 3.5: Tilt or inclination of MPU-6050 accelerometer

The MPU-6050 consists of 3-axis accelerometer with MEMs technology. It is used to detect angle of tilt or inclination along X, Y and Z axes as shown in the figure.

An accelerometer is an electronic sensor that measures the acceleration forces acting on an object, in order to determine the object's position in space and monitor the object's movement. Acceleration, which is a vector quantity, is the rate of change of an object's velocity (velocity being the displacement of the object divided by the change in time).

There are three different types of accelerometers, and they are each designed to efficiently function in their intended environments. The three types are: piezoelectric, piezo resistance and capacitive.

Most accelerometers are miniscule, and they are often referred to as Micro-Electro-Mechanical Systems (MEMS) accelerometers. Because of their size and affordability, they are embedded in a myriad of hand-held electronic devices (such as phones, tablets, and video game controllers). In phones and tablets, the accelerometer is responsible for "flipping" the screen when the device is rotated. Accelerometers are also used by zoologists (to track the movement of animals in the wild), engineers (especially in collision experiments) and factories (to monitor the vibration of machinery).

3.5.2 Gyroscope

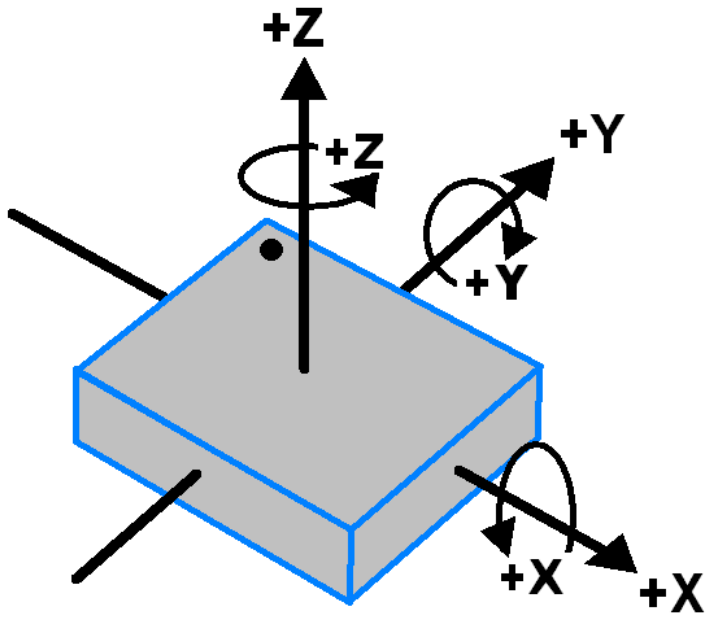

Figure 3.6: Orientation and Polarity of MPU-6050 Gyroscope

The MPU-6050 consists of 3-axis gyroscope with MEMs technology. It is used to detect rational velocity along the X, Y and Z axes as shown in the figure.

A gyroscope is a device used for measuring or maintaining orientation and angular velocity. It is a spinning wheel or disc in which the axis of rotation (spin axis) is free to assume any orientation by itself. When rotating, the orientation of this axis is unaffected by tilting or rotation of the mounting, according to the conservation of angular momentum. It works on the principle of Coriolis acceleration.

3.6 Bluetooth module

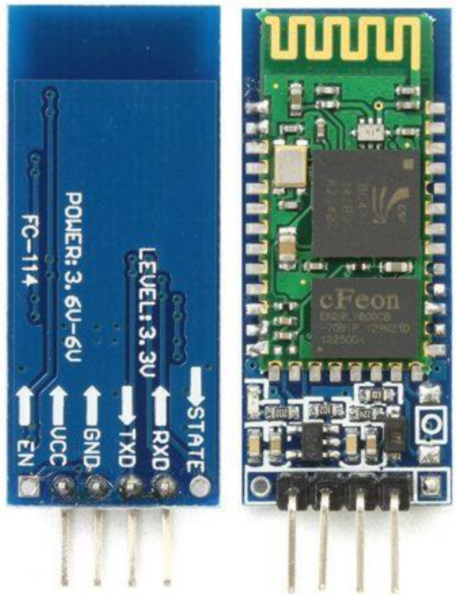

Figure 3.7: Bluetooth module

Bluetooth module (Bluetooth module) refers to the basic circuit set of the chip with integrated Bluetooth function, used for short-range 2.4G wireless communication module. For the end user, the Bluetooth module is a semi-finished product. Through the process of functional redevelopment and packaging of the shell based on the module, the final product capable of utilizing Bluetooth communication is realized. Generally, it refers to the module that supports the Bluetooth protocol below 4.0, which is generally used for relatively large data transmission, such as voice, music and other high data transmission.

3.7 LCD

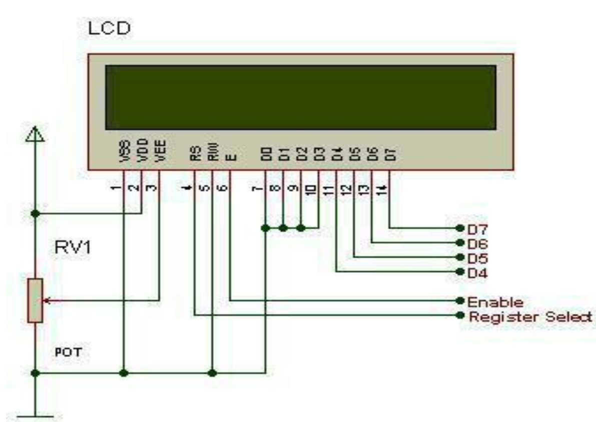

Figure 3.8: LCD

Liquid Crystal Display (LCDs) provide a cost-effective way to put a text output unit for a microcontroller. As we have seen in the previous tutorial, LEDs or 7 Segments do not have the flexibility to display informative messages. This display has 2 lines and can display 16 characters on each line. Nonetheless, when it is interfaced with the microcontroller, we can scroll the messages with software to display information which is more than 16 characters in length.

3.8 I2C Module

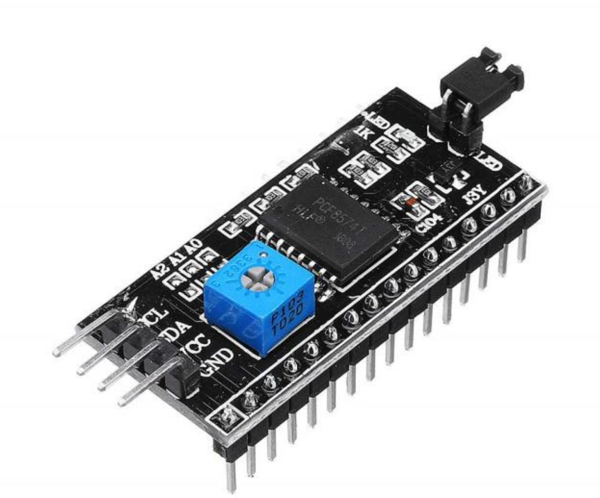

Figure 3.9: I2C Module

I2C is a synchronous, multi slave, multi master packet switched, single-ended serial bus i.e., multiple chips can be connected to the same bus. Typical voltages used are +5 V or +3.3 V, although systems with other voltages are permitted.

It is also known as I2C Module. It has total of 20 male pins. 16 pins are faced to rear side and 4 pins faced towards front side. The 16 pins for connect to 16x2 LCD and the 2 pins out of 4 pins are SDA and SCL. SDA is the serial data pin and SCL is the clock pin. The rest 2 pins for power supply (VCC and ground). There is a POT on the I2C Module. We can control the contrast of the LCD display by rotating this POT. And there is a jumper fixed on the module. When we remove the jumper, the backlight of the LCD display will go OFF.

Chapter 4: METHODOLOGY

4.1 Hardware assembling

Six flex sensors are required. They are to be attached to thumb, index, middle, ring and pinky fingers respectively to measure the bend of the fingers. The final flex sensor is to be attached to the palm to measure the clench of hand. The MPU6050 connected with accelerometer and gyroscope is to be attached on the back of the hand. This determines the position and movement of the hand on the space. Flex sensors and MPU6050 is to be interfaced with Arduino Mega 2560.

All flex sensors are to be powered with 5V source voltage. Flex sensors with resistor is to be provided as input to the analog pins of the Arduino.

MPU6050 and Arduino Mega 2560 is to be interfaced with each other. I2C protocol is required which connects the SDA and SCL pin of the MPU6050 to pin 20 and 21 of Arduino Mega.

Arduino Mega is to be programmed to send data from flex sensors (estimated 6 data from 6 flex sensors), 3 accelerometer data and 3 gyroscope data. So, on a single period 12 data is to be collected. This data is to be received in the laptop and processable by python programming language.

4.2 Dataset Preparation

4.2.1 Dataset Collection

The dataset for training the machine is to be collected manually as this dataset couldn't be found online. Search for dataset is ongoing but it is hard to say if the data acquired will be applicable as the hardware configurations might be completely different. It is preferable to collect data from the received hardware with the realistic parameters.

4.2.2 Data Preprocessing

The collected data is to be cleanly arranged in excel spreadsheet which will generate the csv file. This data also needs to be shuffled, randomized to generate more dataset. The risk of overfitting is expected in the training model in this process.

4.2.3 Dataset Description

The data is then categorized as to what it indicates and what output it should generate. This is a must require process for supervised learning models like Random Forest Classifier.

4.3 System Block Diagram

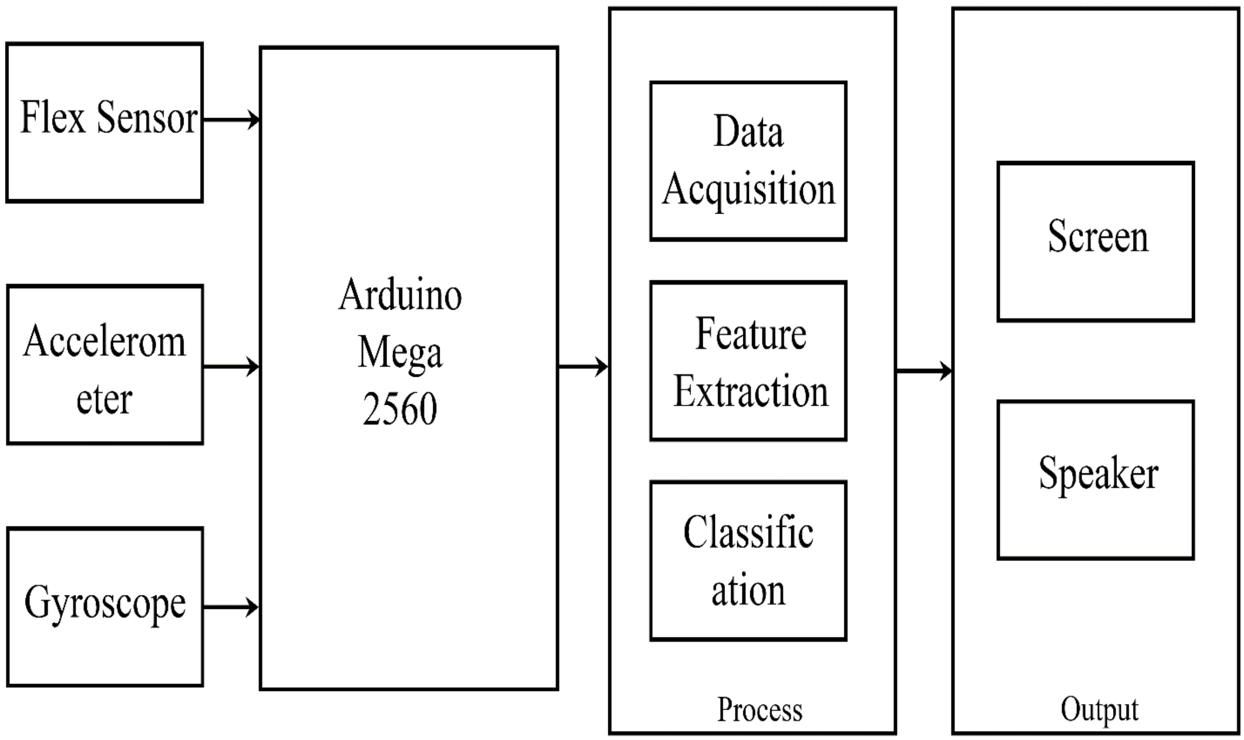

Figure 4.1: System Block Diagram

4.4 Algorithm

4.4.1 Algorithm for data set preparation

Step 1: Start

Step 2: Set the baud rate similar to Arduino

Step 3: Is there data at serial port

If no, wait for the data.

Else go to step 4.

Step 4: Collect data from serial port at real time.

Step 5: Convert collected data to CSV file and write to file.

Step 6: Flush the Arduino buffer.

Step 7: count=count+1

Add delay of 1 second

Step 8: Is Count<=350?

If yes, go to step 9.

Else go to step 3.

Step 10: Close the file

Step 11: End

4.4.2 Algorithm for real time application

Step 1: Start

Step 2: Train the model using training dataset.

Step 3: Set the baud rate similar to Arduino.

Step 4: Is there data at serial port?

If no, wait till the data is available.

Else go to step 5.

Step 5: Collect data from the serial port at real time.

Step 6: Convert collected data to CSV file and write to file.

Step 7: Flush the Arduino buffer.

Step 8: Close the file.

Step 9: Pass data through the model.

Step 10: trained machine predicts the output.

Step 11: Display the prediction on the screen and play the audio of the predicted word.

Step 12: Sleep for 1 second.

Step 13: Go to step 4.

4.5 Flowchart

4.5.1 Flowchart for data set preparation

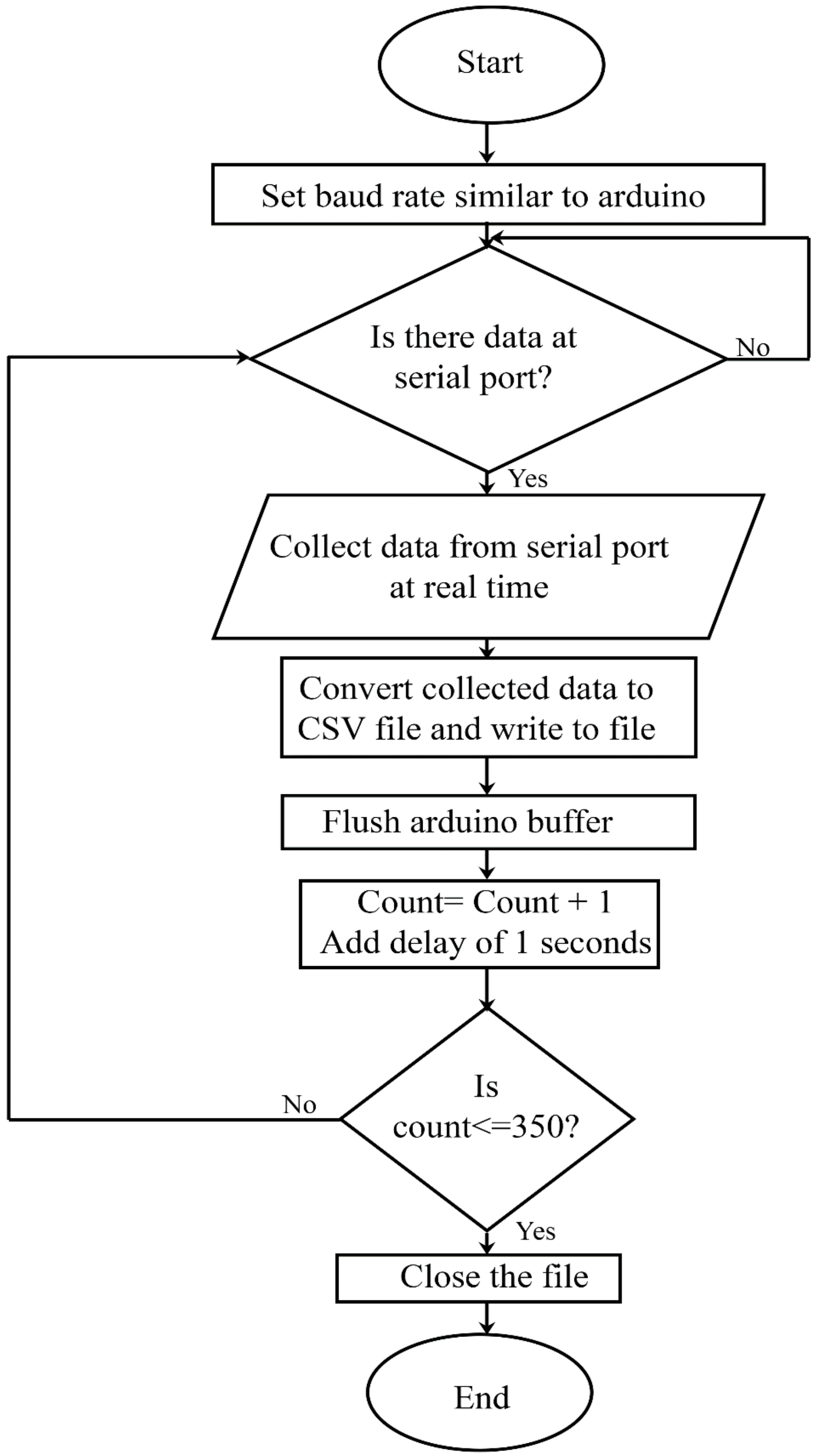

Figure 4.2 Flowchart for dataset preparation

4.5.2 Flowchart for real time application

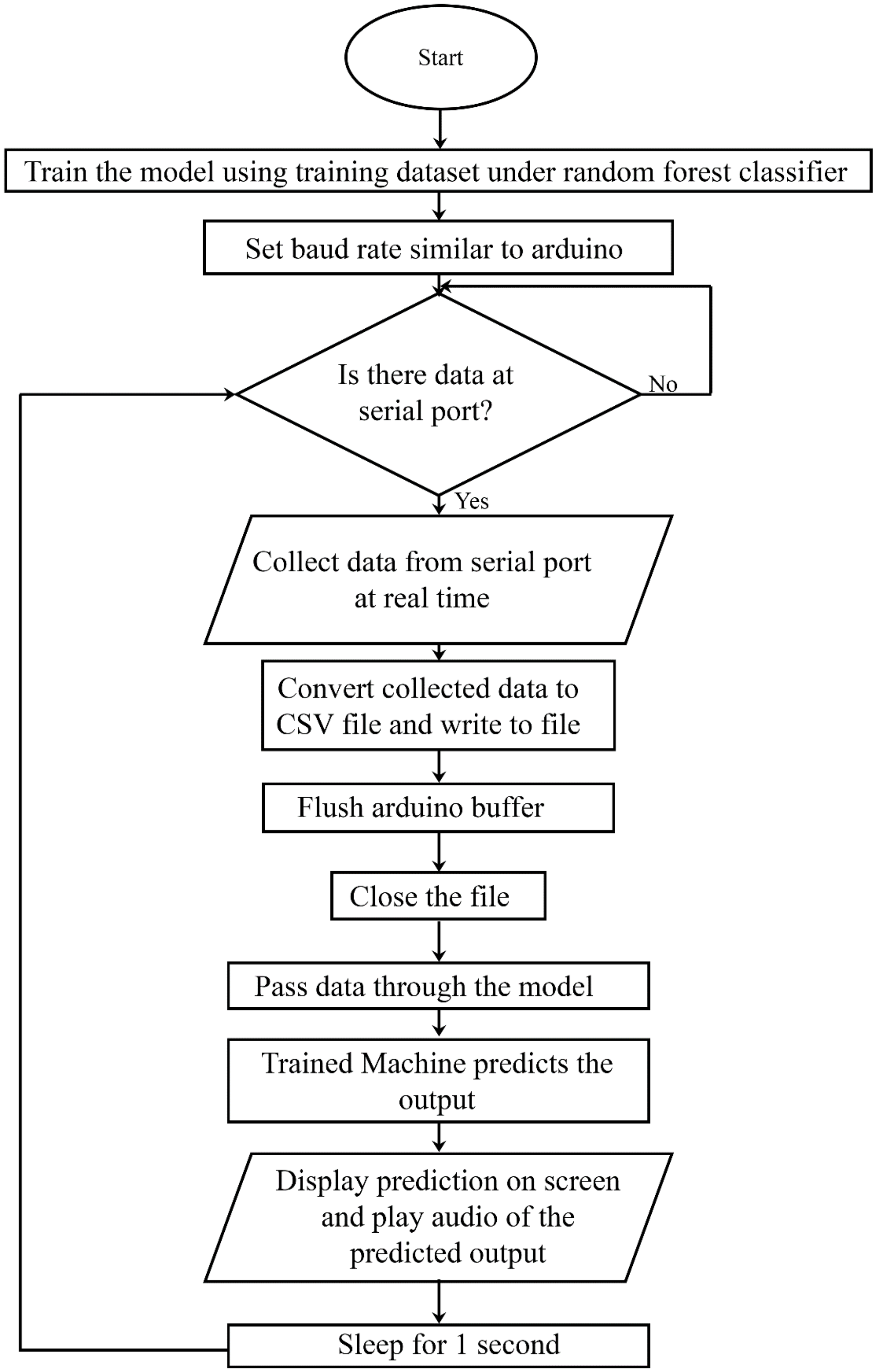

Figure 4.3: Flowchart for real time application

Chapter 5: WORK PROGRESS

5.1 Completed Work

The works that were done are as follows:

5.1.1 Research Works

For our research, we needed to analyze different supervised learning models and select the best one. For this we couldn't pick any regression models, as they would be highly unsuitable for uncorrelated dataset. So neural network and regression models couldn't be used. Naïve Bayes and K-Nearest Neighbor were the other choices. Naïve Bayes assumes Bayes theorem, that probabilities are independent, but for our dataset, that isn't always the case. As for K-Nearest Neighbor, we could technically use it as it seems excellent in correlating between all lengths of hands and data types which will definitely be closer in numerical proximity. However, for our project, the datasets are not correlated between various inputs but only during the glove being used by multiple people. So, for better classification, we have to use Random Forest classifier where inputs can coexist independently from each other.

5.1.2 Simulation Works

Hardware Works:

We interfaced the flex sensor with the Arduino MEGA 2560. We connected the 1st pin of the flex sensor to the Arduino GND (ground) pin and the 2nd flex sensor pin was connected to the A0 port of the Arduino and a resistor was connected between the 5V supply of the Arduino and the 2nd flex sensor pin. We checked the working of the flex sensor by running a simple code to turn on a buzzer and turn on a LED when the resistance value of the flex sensor exceeds a certain threshold. We collected the resistance values of the flex sensor before and after it was bent.

The following figure shows the interfacing of the flex sensor with Arduino.

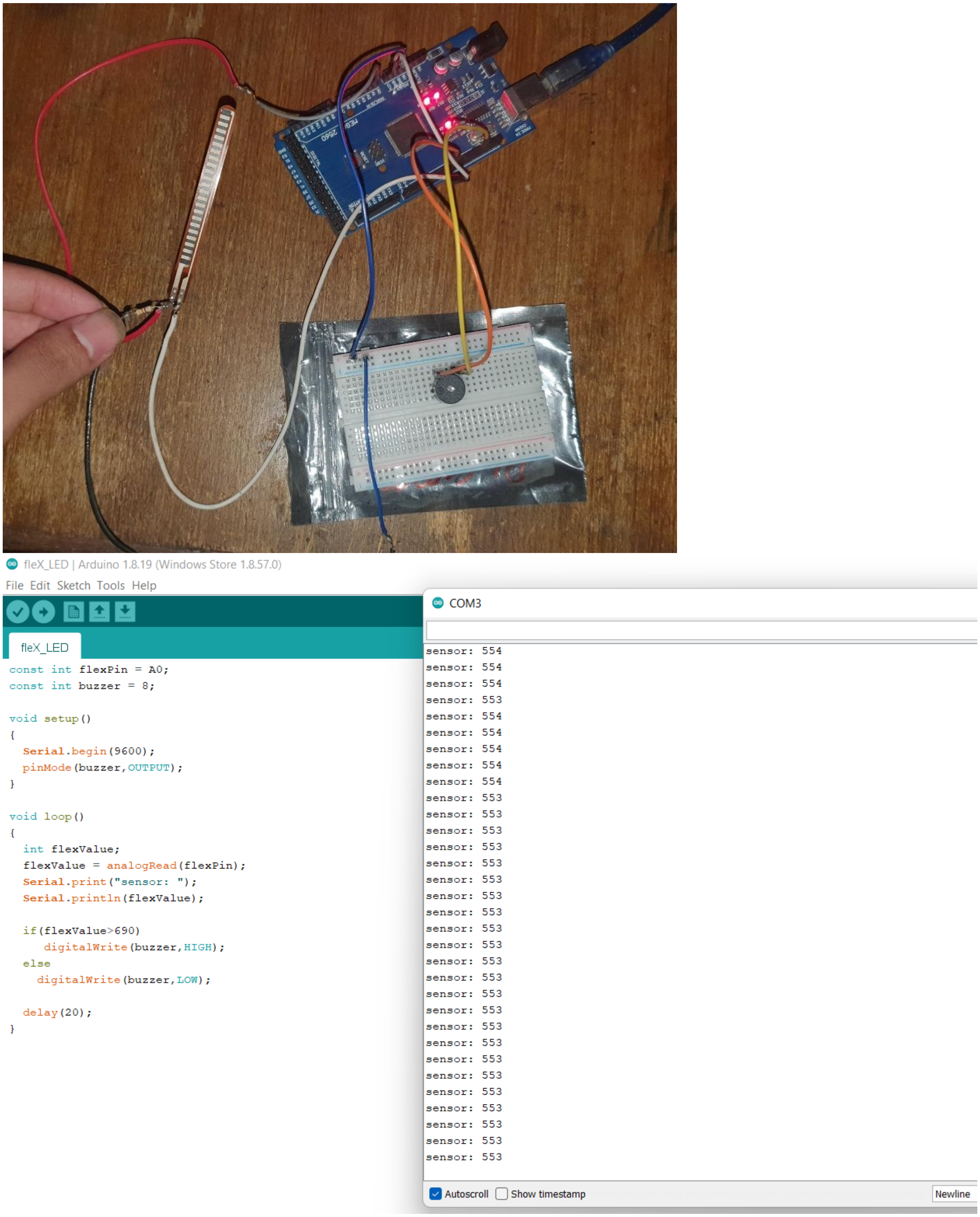

Figure 5.1: Before Bending

As shown in the above Figure 5.1, those are the resistance values of the flex sensor at a default state.

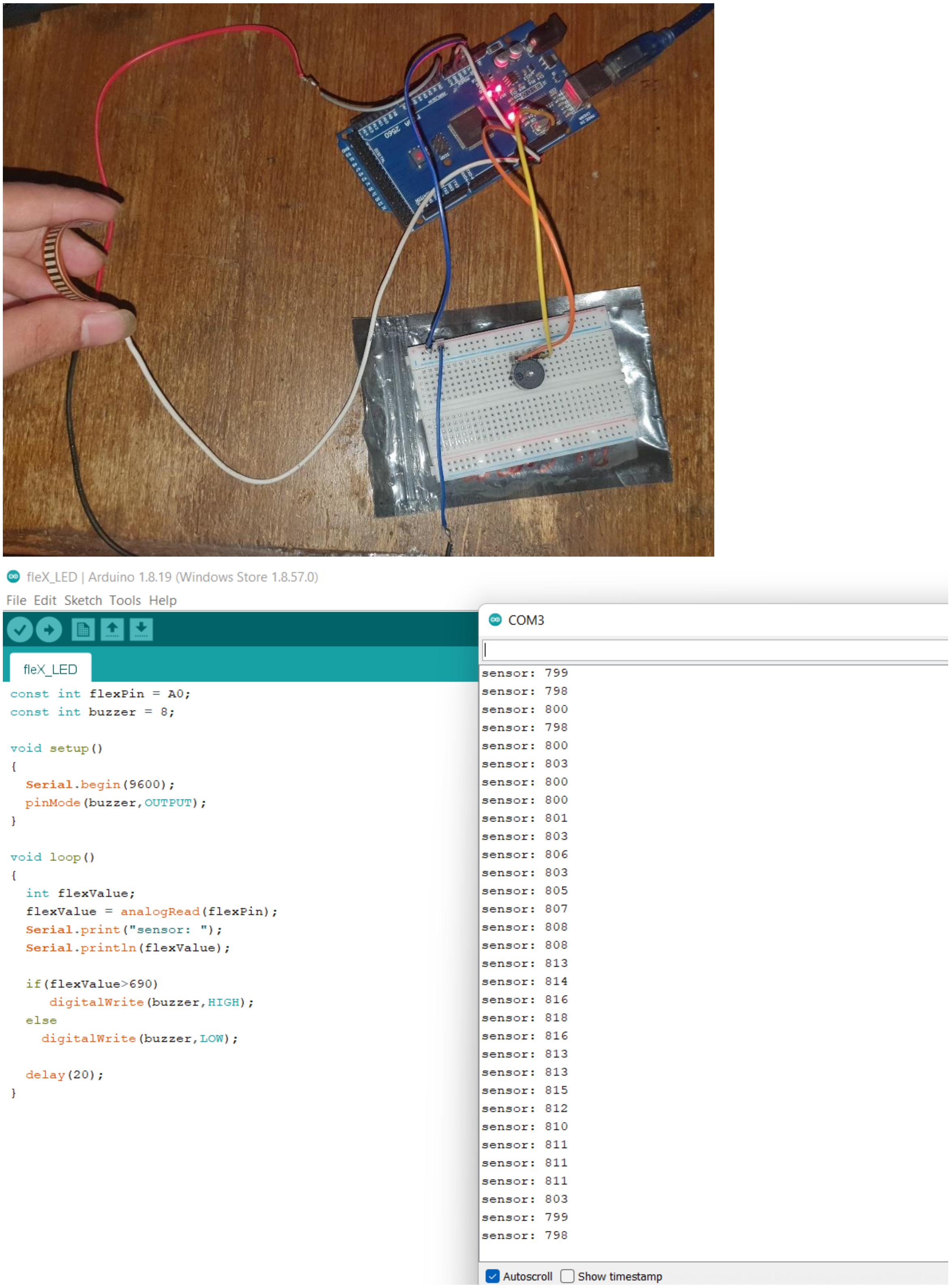

Figure 5.2: After Bending

And after bending the flex sensor to a certain degree the resistance values change as shown in the above Figure 5.2.

Software Works:

We successfully simulated the Random Forest classifier on a similar dataset which will be necessary during implementation. This also made us understand the python implementation through the sklearn library and pandas library which we used for training. By using pickle, we were able to export the trained model saving us the hassle of retraining to run the code. We also were able to play sound of the inputs using the playsound library. We also were able to send a custom data into the model and the model would predict it. By sending different data's, we were able to manipulate the program into playing different sounds. Following is the dataset containing approx. 100 tuples that we used for short implementation of Random Forest. The model gives output 0 if the user isn't purchasing a car and 1 if purchasing. The code is given in appendix.

1	User ID	Gender	Age	AnnualSalary	Purchased
2	385	Male	35	20000	0
3	681	Male	40	43500	0
4	353	Male	49	74000	0
5	895	Male	40	107500	1
6	661	Male	25	79000	0
7	846	Female	47	33500	1
8	219	Female	46	132500	1
9	588	Male	42	64000	0
10	85	Female	30	84500	0
11	465	Male	41	52000	0
12	686	Male	42	80000	0
13	408	Male	47	23000	1
14	790	Female	32	72500	0
15	116	Female	27	57000	0
16	118	Female	42	108000	1
17	54	Female	33	149000	1
18	90	Male	35	75000	0
19	372	Male	35	53000	0
20	926	Male	46	79000	1
21	94	Female	39	134000	1
22	338	Female	39	51500	0
23	134	Female	49	39000	1
24	821	Male	54	25500	1
25	294	Female	41	61500	0
26	597	Female	31	117500	0
27	567	Male	24	58000	0

Figure 5.3: Simulation dataset for random forest classifier

5.2 Remaining Work

1. Data collection and Parsing
2. Writing Data into excel file
3. Hardware assembling
4. Code based on real data model
5. Research and adjustment as required
6. Testing and Output Processing

REFERENCE

[1] D. Hubbell, “Blogs: Microsoft,” 9 January 2014. [Online]. Available: https://blogs.msdn.microsoft.com/accessibility/2014/01/19/ukranian-students-design-gloves-that-translate-sign-into-spoken-words/. [Accessed 25 May 2022]

[2] Laura Dipietro, Angelo M.Seior, member IEEE, “A Survey of Glove-Based System and their Applications,” *IEEE transaction on System, Man and Cybernetics-Part C: Application and Review*, vol. 38, no. 4, 2008.

[3] X Chai et al., "Sign Language Recognition and Translation with Kinect," Institute of Computing Technology, CAS, Microsoft Research Asia, Beijing, China, October 2013.

[4] Sidney Fels and Geoffrey Hinton, “Glove Talk II,” Department of Computer Science of University of Toronto, Toronto, 6 November 1992.

[5] Y Quan and P Jinye, "Application of improved sign languages recognition and synthesis technology," Industrial Electronics and Application, 2008, ICIEA 2008, 3rd IEEE Conference, pp. 1629- 1634, June 2008.

[6] “Speakjet,” Magnevation Speak Jet Inc. USA, 4 June 2010. [Online]. Available: http://www.speakjet.com/. [Accessed 25 May 2022]

[7] Ata Ur Rehman, Salman Afghani, Muhammad Akmal, and Raheel Yousaf, "Microcontroller and Sensors Based Gesture Vocalizer," Rachna College of Engg. and tech., Air University Islamabad, Islamabad, Research Paper 2008.

[8] Prerana K.C, Nishu Mahato, and Bishruti Satyal, "Hand Gesture Recognition using KINECT," Nepal Engineering College, Baktapur, Bachelor Thesis P3-I.0-14-4, 2015.

[9] Avinash Navalani. (2018, May) datacamp. com. [Online]. https://www.datacamp.com/community/tutorials/random-forests-classifier-python.

[10] Shubham Mishra, Saurabh Takkar and Tripti Garg (15 March 2018), “Arduino based Gestures to Speech Conversion System” [Online]. International Journal of Advance Research and Innovation.

[11] Tony Yiu (Jun 12, 2019), “Understanding Random Forest” [Online]. https://towardsdatascience.com/understanding-random-forest-58381e0602d2.

[12] Sparkfun. (2015) learn.sparkfun.com. [Online]. https://learn.sparkfun.com/tutorials/flex-sensor-hookup-guide/all

APPENDIX

Working Codes:

```python
import pandas as pd
from matplotlib import pyplot as plt
import numpy as np
import os

df=pd.read_csv("car_data.csv")
# print(df.head())
# sizes=df['Attribute1'].value_counts(sort=1)   #counts the no. of data for each class/label
# print(sizes)



df.drop(['User ID'], axis=1, inplace=True) #Drop the irrelevant attribute

# print(df.head())
# df= df.dropna()  #If some data is missing it will drop the whole tuple

# #Convert non-numeric data to numeric
df.Gender[df.Gender=='Female']=0
df.Gender[df.Gender=='Male']=1


#Define dependent variable the output value that it should give
Y=df['Purchased'].values #values of the attribute column
Y=Y.astype('int') #changes the values to list of integers
# print(df.head())

# #Define independent variable, this value will include all flex sensor value
X=df.drop(labels=['Purchased'], axis=1)
# print(Y)
# print(df.head())

from sklearn.model_selection import train_test_split
# #Split Data into training and testing datasets
X_train, X_test, Y_train, Y_test = train_test_split(X,Y,test_size=0.3, random_state=20)
# print(X_test)
from sklearn.ensemble import RandomForestClassifier
model = RandomForestClassifier(n_estimators=12, random_state=30)  #n-estimators no. of trees before selecting best tree
model.fit(X_train, Y_train)

prediction_test = model.predict(X_test)
from sklearn import metrics
print("Accuracy=",metrics.accuracy_score(Y_test,prediction_test))

import pickle
with open('model_pickle','wb') as f:
    pickle.dump(model,f)
```

```python
import pandas as pd
from matplotlib import pyplot as plt
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn import metrics
import pickle
import os
from playsound import playsound

data=[["0","80","45000"]] #random data custom creation

#load model
with open('model_pickle','rb') as f:
    mp=pickle.load(f)


a=mp.predict(data)
#prediction outputs array so convert to int
r=a[0]
print(r)
if r==1:
    playsound('Hello.mp3')
else:
    playsound('Thank_You.mp3')
```