ANALYSIS OF FINGERSPELLING RECOGNITION PROTOTYPE ACCURACY USING SENSOR-BASED SIGNAL

MUHAMMAD HAZMIRUL HAQIM BIN HAZRI

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

ANALYSIS OF FINGERSPELLING RECOGNITION PROTOTYPE ACCURACY USING SENSOR-BASED SIGNAL

MUHAMMAD HAZMIRUL HAQIM BIN HAZRI

This report is submitted in partial fulfillment of the requirements for the degree of Bachelor of Computer Engineering with Honours

Faculty of Electronic and Computer Technology and Engineering
Universiti Teknikal Malaysia Melaka

2023/2024

DECLARATION

I declare that this report entitled “Analysis of Fingerspelling Recognition Prototype Accuracy Using Sensor-Based” is the result of my own work except for quotes as cited in the references.

Signature : *Hazmirul*

Author : Muhammad Hazmirul Haqim Bin Hazri

Date : 14/6/2024

APPROVAL

I hereby declare that I have read this thesis and in my opinion, this thesis is sufficient in terms of scope and quality for the award of Bachelor of Computer Engineering with Honours.

Signature : ……………………………………………

Supervisor Name : DR. ZARINA BINTI MOHD NOH
Pensyarah Kanan
Fakulti Teknologi Dan Kejuruteraan Elektronik Dan Komputer (FTKEK)
Universiti Teknikal Malaysia Melaka (UTeM)

Date : 14/06/2024

DEDICATION

This thesis is dedicated to my mum, Umi Sipak Binti Fakharudin and my father, Hazri Bin Zamzuri, who is constantly giving me encouragement and supporting me in everything I do,

ABSTRACT

Individuals with hearing and speech impairments often rely on sign language, which use specific hand gestures combined with facial expressions to communicate. These gestures are precise movements of the hands, forming the basis of sign languages used by the deaf community. Current research aims to translate these hand gestures into spoken language using electronic devices, bridging the communication gap between deaf and hearing individuals. This study presents a prototype glove equipped with one accelerometer and five flex sensors, designed to recognize and convert fingerspelling into an alphabet. The glove is fitted with flex sensors along the length of each finger capturing the movements and bends of the fingers. These sensors measure changes in resistance caused by finger bends, while the accelerometer tracks the hand's movement for the alphabet "J" and "Z". The collected data is processed by a microcontroller, which then transmits the information to an OLED display. This system allows deaf individuals to make various hand gestures while wearing the glove, which are then translated into the corresponding alphabet for easy recognition by hearing individuals. The flex sensors play a crucial role in accurately detecting finger movements. The integration of flex sensor data with accelerometer readings provides a comprehensive understanding of hand gestures, enabling precise conversion into alphabet. This project demonstrates the potential of sensor-based gloves to facilitate communication for the deaf and hard-of-hearing communities, enhancing their ability to interact with the hearing world. The accuracy of this project's first characterization for user-based is 62% and alphabet-based is 63%. For final characterization after improvement from the first characterization for user-based is 76% and alphabet-based also 76%.

ABSTRAK

Individu dengan gangguan pendengaran dan ucapan sering bergantung kepada bahasa isyarat, yang menggunakan gerakan tangan tertentu dikombinasikan dengan ungkapan muka untuk berkomunikasi. Tindakan ini merupakan pergerakan tangan yang tepat, yang membentuk asas kepada bahasa isyarat yang digunakan oleh masyarakat yang tuli. Penyelidikan semasa bertujuan untuk menterjemahkan gerakan tangan ini ke dalam bahasa lisan menggunakan peranti elektronik, menampung jurang komunikasi antara orang yang tuli dan pendengaran. Kajian ini memperkenalkan sarung tangan prototype yang dilengkapi dengan satu akselerometer dan lima sensor fleks, yang direka untuk mengenali dan menukarkan jari-jari ke dalam alfabet. Sarung tangan ini dilengkapi dengan sensor flex sepanjang panjang setiap jari yang menangkap pergerakan dan kemiringan jari-jari. Sensor ini mengukur perubahan dalam rintangan yang disebabkan oleh kemiringan jari, manakala akselerometer menjejaki pergerakan tangan untuk huruf "J" dan "Z". Data yang dikumpulkan diproses oleh mikrokontroler, yang kemudian menghantar maklumat kepada skrin OLED. Sistem ini membolehkan individu tuli untuk membuat pelbagai gerakan tangan semasa memakai sarung tangan, yang kemudian diterjemahkan ke dalam alfabet yang sesuai untuk dikenali dengan mudah oleh individu yang mendengar. Sensor flex memainkan peranan penting dalam mengesan pergerakan jari dengan tepat. Integrasi data sensor fleks dengan pembacaan akselerometer menyediakan pemahaman yang komprehensif gestur tangan, membolehkan penukaran yang tepat kepada alfabet. Projek ini mendemonstrasikan potensi sarung tangan berbasis sensor untuk memudahkan komunikasi bagi masyarakat yang tuli dan berisiko mendengar, meningkatkan keupayaan mereka untuk berinteraksi dengan dunia pendengaran. Ketepatan karakterisasi pertama projek ini untuk berasaskan pengguna ialah 62% dan berasaskan alfabet ialah 63%. Untuk karakterisasi akhir selepas peningkatan daripada karakteristik pertama untuk pengguna berasaskan adalah 76% dan berasaskan alfabet juga 76%.

ACKNOWLEDGEMENTS

First and foremost, I would like to express my gratitude to the administration of Universiti Teknikal Malaysia Melaka (UTeM), especially Fakulti Teknologi Kejuruteraan Elektronik dan Kejuruteraan Komputer (FTKEKK) for ensuring an ambiance study environment and complete facilities for me to complete this research.

Besides that, with immense gratitude, I would like to acknowledge the support and help I got from my supervisor, DR. ZARINA BINTI MOHD NOH. She has given me plenty of valuable guidance and advice throughout the process of completing this research. She is constantly encouraging me and complimenting my work, and it helps in motivating me to contribute to this project. Plenty of useful resources and reading materials were provided by her and it aided me in completing this study.

Furthermore, I would also like to thank my classmates who are always helpful and provide guidance as well as advice to me when I need help.

Lastly, a special mention goes to my parents, family, and friends for their support and encouragement in me completing this project. Without the assistance of the mentioned above, I would not be able to complete this project smoothly.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS AND ABBREVIATIONS

ASL	:	American Sign Language
ADC	:	Analog-to-Digital Converter
BIM	:	Bahasa Isyarat Malaysia
BSL	:	British Sign Language
DTW	:	Dynamic Time Warping
LSTM	:	Long-Short Term Memory
PCB	:	Printed Circuit Board

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

This chapter includes six sub-chapters which are project overview, problem statement objectives, scope of work and the report structure.

1.1 Project Background

This project will explore the fingerspelling of the alphabet using a flex sensor. Incorporating flex sensors into specialized gloves to capture finger movements for sign language fingerspelling gestures is a promising initiative that can significantly enhance accessibility and communication for the deaf and hard-of-hearing community. In this case, interactions between deaf people and normal people have always been a troublesome task. This is because not every ordinary person can comprehend sign language. This increases their life difficulty as communication is one of the necessities. This will affect their integration into society. However, this Project focuses on analyzing the accuracy of the fingerspelling recognition prototype in converting the alphabet. Flex sensors and microcontrollers will be used in the designed prototype to obtain good accuracy in the fingerspelling alphabet. At the end of this project, the processed data generates a user-friendly interface, which could be displayed on a screen or other output devices. By using this technology, deaf and hard-of-hearing individuals can communicate more efficiently and accurately in sign language.

1.2 Problem Statement

Statistics show that around 9 billion people are deaf. In Malaysia, there are about 2.8 million people who have disabilities [1]. However, interactions between deaf people and normal people have always been a troublesome task to communicate. This is because not every ordinary person can comprehend sign language. This increases their life difficulty as communication is one of the necessities. This willaffect their integration into society. To overcome this issue, a sign language recognition system must be developed with a specific end goal to bridge the gap between the disabled and normal individuals. The main goal of this project is to develop a prototype of analysis fingerspelling recognition based on a sensor that translates hand gestures into the alphabet. Since not every typical person is educated in sign language communication, this system will help them comprehendthe language of deaf people to ease their daily tasks. Fingerspelling is a key component of sign language. This research addresses this problem by developingand analyzing a fingerspelling prototype using flex sensors. The issue is the lack of an effective and user-friendly fingerspelling interpretation system, which creates barriers to effective communication and inclusivity for the deaf and hard-of-hearing community.

1.3 Project Objectives

The objectives of this project are to:

1. Configure a prototype of a fingerspelling recognition system using flex sensors.
2. Characterize the flex sensor reading according to the alphabet that the fingerspelling represents.
3. Analyse the accuracy of the fingerspelling recognition prototype in converting the alphabet based on five different users.

1.4 Scopes of Work

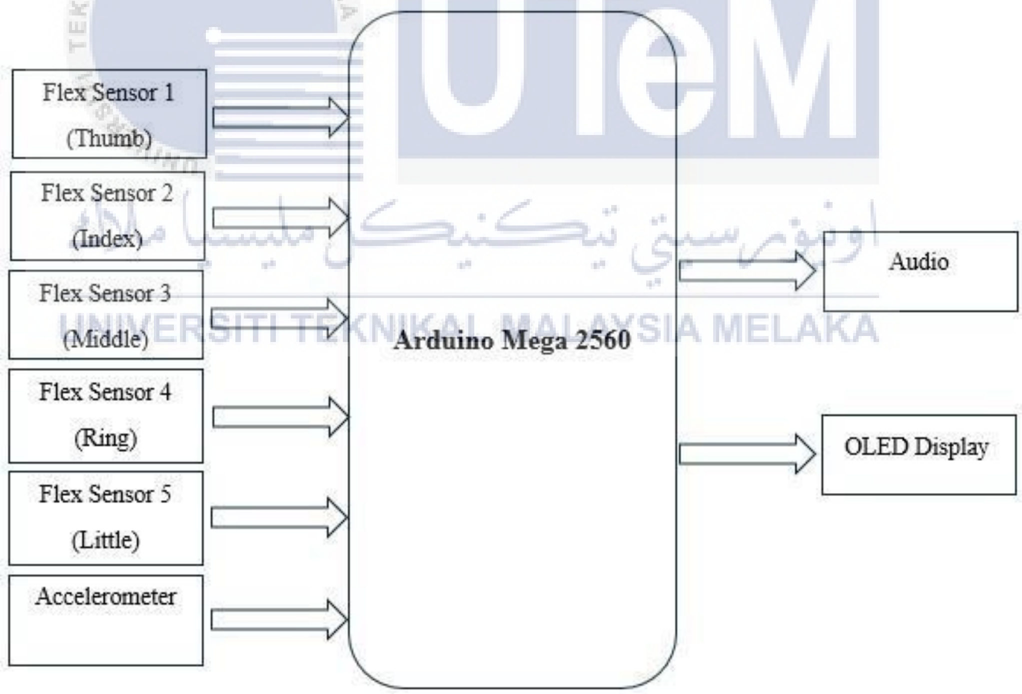


Figure 1.1: Block Diagram Scope of Work.

Figure 1.1 shows a block diagram of the scope of work. This project consists of five flex sensors and one accelerometer. The function of the flex sensor is to do the hand gesture alphabet and accelerometer to detect movement for the alphabet "J" and

"Z". Each finger will be attached to a flex sensor. Flex sensor 1 is equivalent to the thumb, flex sensor 2 is for the index finger, flex sensor 3 for the middle, flex sensor4 for the ring, and flex sensor 5 for little. The flex sensor will send the input resistance value to Arduino Mega. Then, Arduino Mega will convert the resistance value to an ADC value. The alphabet output from the Arduino Mega will be delivered through audio and OLED display.

1.5 Report Structure

This report was structured into five chapters, including an introduction to the fingerspelling recognition system in the first chapter. The project background, objectives of this study, the problem statement, and the scope of work of the project are discussed in the first chapter. The background research for this study, which covered a review of the literature on an overview of recognition in fingerspelling, fingerspelling in recognition systems, different vision-based and sensor-based sign language recognition systems, and a summary of the literature review was included in the second chapter. The project methodology was thoroughly outlined in the third chapter, and the results were analyzed in the fourth. Finally, the fifth chapter is future work and concludes all the findings from the previous chapters.

CHAPTER 2

BACKGROUND STUDY

2.1 Important of Sign Language Recognition System

Deaf people can learn Sign Language more easily with the help of Finger's alphabet. This technique, which is rarely used alone, involves representing the alphabetic letters with one hand. It is also useful for introducing new phrases and concepts [2]. It can also be applied to names that lack a sign that the deaf population is familiar with. The significance of the alphabet of the fingers is more apparent since each of the twenty-six letters may be made with a particular hand motion. For instance, the letter A and the other letters have distinct vowels.

Sign language recognition systems are important because they provide a critical means of communication for the hearing impaired or especially abled. These technologies allow sign language users to efficiently communicate, express themselves, and share their emotions with others. Sign language recognition systems bridge the communication gap between sign language users and non-users by identifying and converting sign language gestures into frequently spoken languages such as English, allowing for seamless interaction and understanding. Furthermore, these technologies play an important role in empowering the hearing-impaired community by giving them the ability to communicate more freely and effectively with a broader spectrum of people.

One technique for visual physical communication is finger spelling, which displays a manual manner and symbolizes written language [3]. It works by having the letters drawn out and then written in the air rather than on paper. The alphabetical approach is regarded as one of the most crucial ways to communicate with the deaf since it is an essential component of general communication, particularly when discussing words, names, or places without agreed-upon signs. Fingerspelling is the quickest way for the deaf to see words in front of their eyes and is appropriate for reading. Because the fingerspelling method is an application of the writing skills the hearing-impaired student has already acquired, training in it will not require much effort from the student.

2.2 Fingerspelling in Sign Language Recognition

A fingerspelling recognition system is a type of sensor-based or vision-based sign language recognition system that specifically focuses on recognizing and translating fingerspelled words and letters in sign language. Fingerspelling recognition systems aim to automatically detect and translate these rapid hand movements and finger positions into text, enabling more efficient communication for the deaf and hard-of-hearing community. Communication between human beings is not only through speech or voice. However, such differently abled people who try to communicate with the normal world find it difficult. Hence, they use sign language to communicate.

Without using voice or speech as a communication medium with one another there are several processes of communication like hand shape, body and arm movements, and facial expression. This mode of communication is called Sign Language. Mute community for example deaf people use sign language as the medium of communication. As it is very tough for normal people to recognize and analyze sign language. Thus, the barrier between normal people and the mute community is created which can be broken through the study of sign language.

All words used by normal people can be derived through fingerspelling. The fingerspelling also guides different signs of letters to construct words that are correctly unavailable in a sign language dictionary. Sentence construction can be done using sign letters and the wing sign of the word is faster. The current research work uses “Bahasa Isyarat Malaysia (BIM)” gestures as shown in Figure 2.1.

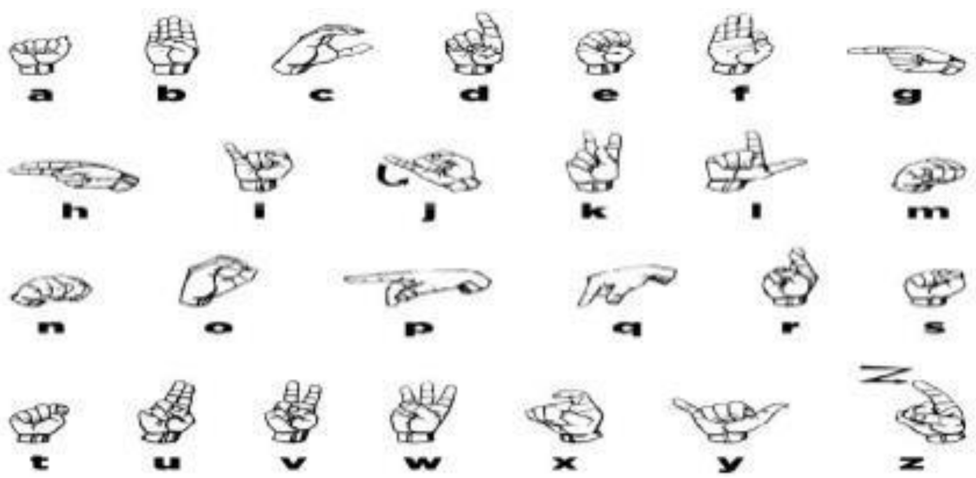


Figure 2.1: Sign Language Gestures [4].

In this category of gestures, the position of the hand does not affect the gesture's meaning and the gestures are time-independent [4]. Bahasa Isyarat Malaysia, static one-hand gestures refer to the alphabet, as shown in Figure 2.1. Therefore, these signs are very important because signers use them to spell out names such as their names, names of places they want to go to, names of food they want to eat, or to name anything that does not have special signs.

The differences between Bahasa Isyarat Malaysia and British Sign Language (BSL) in fingerspelling recognition systems, particularly focusing on the alphabet J and Z, are notable. In BSL, fingerspelling involves the use of two hands, which introduces more variation between signs compared to ASL, where fingerspelling is typically one-handed [5]. This difference in hand usage can impact the recognition system's ability to accurately interpret and classify the gestures for letters like J and Z. In the context of the fingerspelling recognition system for BSL, the involvement of both hands overlapping each other poses a challenge, making it more complex to differentiate between letters like J and Z due to the additional hand interactions and movements required.

On the other hand, BIM fingerspelling recognition systems, which are typically one-handed, may have a more straightforward recognition process for letters like J and Z, as the gestures are performed with a single hand, allowing for clearer and more distinct recognition patterns. The recognition system for BIM fingerspelling may be able to achieve higher accuracy by leveraging the one-handed nature of the gestures and the ability to better account for variations between signers. The two-handed BSL fingerspelling may introduce more signer-specific variations, making it more challenging for the recognition system to generalize and maintain high accuracy across different users [5]. Therefore, the key difference lies in the complexity introduced by the two-handed nature of BSL fingerspelling compared to the typically one-handed nature of BIM fingerspelling, impacting the recognition and classification of letters like J and Z in the respective sign languages.

2.3 Vision-Based and Sensor-Based Sign Language Recognition System

A vision-based approach uses visual data, typically captured through cameras or other imaging devices, to interpret and recognize sign language gestures. Vision-based systems analyze images or video footage of hand and arm movements to identify and translate sign language signs into text or speech. These systems often involve image pre-processing, feature extraction, and gesture classification stages to interpret the gestures made during sign language communication accurately. By leveraging computer vision techniques and algorithms, vision-based sign language recognition systems can effectively bridge communication gaps for individuals with hearing and speaking impairments by enabling the recognition and translation of sign language gestures into meaningful content.

Vision-based sign language recognition systems typically work in three main phases: image pre-processing, feature extraction, and gesture classification. Figure 3 is an example of vision-based sign language. The feature extraction phase is crucial for the performance of various classifiers used in these systems. Researchers have explored a variety of feature extraction techniques for vision-based sign language recognition, such as hand direction, wrist orientation, and joint angle detection [6].

The limitations of vision-based fingerspelling recognition systems stem from various factors. These systems face challenges such as occlusion, where the hand or fingers may be obstructed during the gesture, impacting recognition accuracy. Additionally, variations in lighting conditions can affect image quality, making it difficult to segment and extract features from hand gestures. Complex backgrounds can also interfere with hand segmentation, leading to feature extraction and recognition errors. Moreover, the wide range of possible hand poses and orientations during fingerspelling poses a challenge for developing robust recognition algorithms [7].

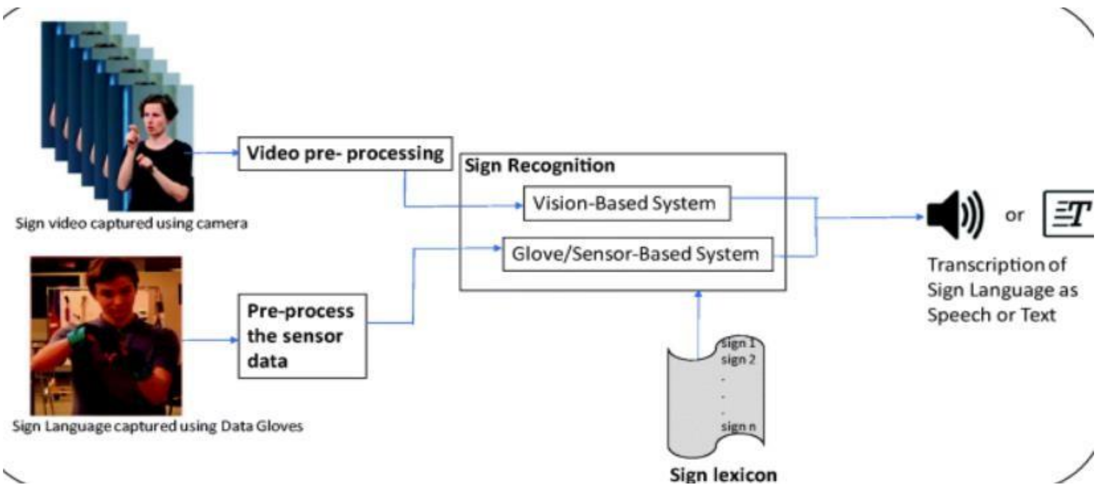


Figure 2.2: Vision-based Sign Language Recognition System [8].

In Figure 2.2 the Vision-based sign language recognition systems can be more user-friendly and less intrusive compared to sensor-based approaches, as they do not require the user to wear specialized equipment [9]. In summary, vision-based sign language recognition is a promising research area that aims to enable natural and intuitive communication for the deaf and mute community, leveraging advances in computer vision and machine learning techniques.

Sensor-based sign language recognition systems involve the utilization of various sensors to capture and interpret the gestures and movements made during sign language communication. These systems rely on sensor data, such as data from accelerometers, flex sensors, or wearable devices, to detect and analyze the hand and arm movements involved in sign language gestures. By processing this sensor data using machine learning algorithms, sensor-based systems can recognize and translate sign language signs into text or speech, facilitating communication for individuals with hearing and speaking impairments.

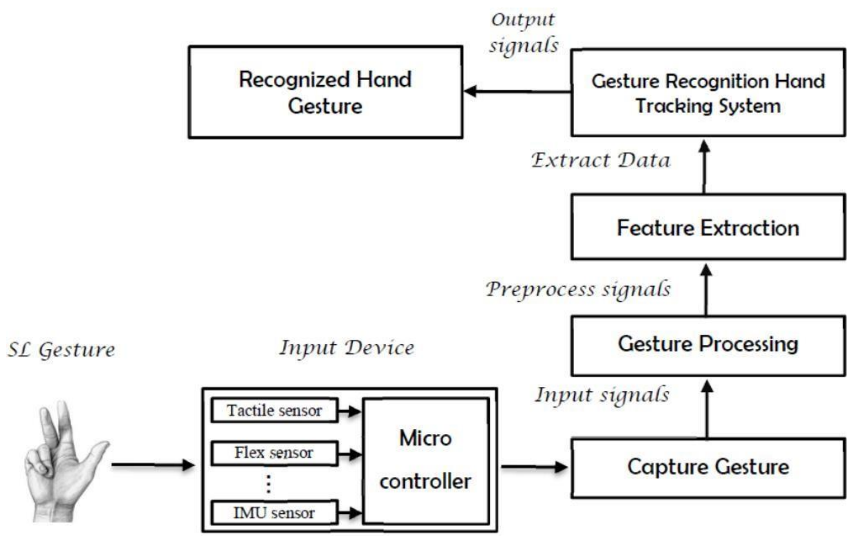


Figure 2.3: Block Diagram of Sensor-Based [10].

Figure 2.3 shows the block diagram of the sensor-based. The system uses flex sensors attached to the fingers to detect the bending of the fingers during fingerspelling gestures. An accelerometer is also used to measure the dynamic and static gestures of the hand. The signals from the flex sensors and accelerometer are processed by a microcontroller, which converts the analog sensor outputs to digital data. Based on the sensor data, the microcontroller then uses lookup tables or algorithms to recognize the fingerspelling gestures. Once the letter is identified, the microcontroller sends the data to a computer or display device where the recognized letter is displayed.

Flex sensors are commonly used in sensor-based sign language recognition systems to detect finger bending and hand gestures. The resistance of the flex sensors changes as the fingers bend, allowing the system to capture the intricate movements involved in sign language [11]. Sensor-based systems often combine flex sensors with other sensors like accelerometers to capture both finger movements and hand/arm motions, providing a more comprehensive representation of sign language gestures.

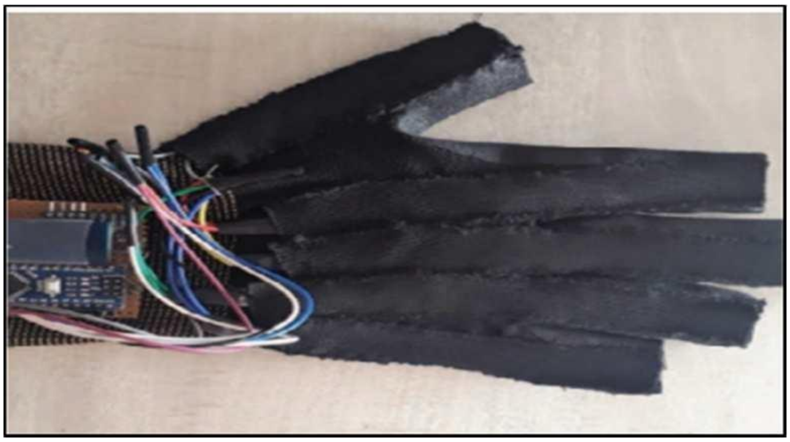


Figure 2.4: Sensor-Based Glove for Sign Language Recognition [12].

In this approach, the input is given using hardware sensors in the form of gloves in Figure 2.4. The sign language detection process gets simplified and faster when using sensor-based input. In gloves, multiple sensors are embedded to track every movement of the fingers, palm, and their location. Sensor-based gloves can be connected using Bluetooth. Sensor-based recognition has the advantage it being thin and flexible, so there is no such problem of space or location of placement. A portable sensor-based glove was designed by [12] that could translate real-time signs into speech using LSTM (Long-Short Term Memory) networks. It had a Testing accuracy of 98%. The designed glove having wireless capabilities (Bluetooth) used low-cost hardware making it affordable for many of its users. Figure 5 shows the prototype of a sensor-based glove.

These sensor-based approaches aim to provide a cost-effective and user-friendly solution for individuals with speech and hearing impairments, enabling them to communicate more effectively by converting sign language into a format understandable by the general population [13]. In summary, flex sensors play a crucial role in sensor-based sign language recognition systems, allowing for the accurate detection of finger movements and hand gestures, which are then processed and translated into text or speech output to facilitate communication for the deaf and mute community.

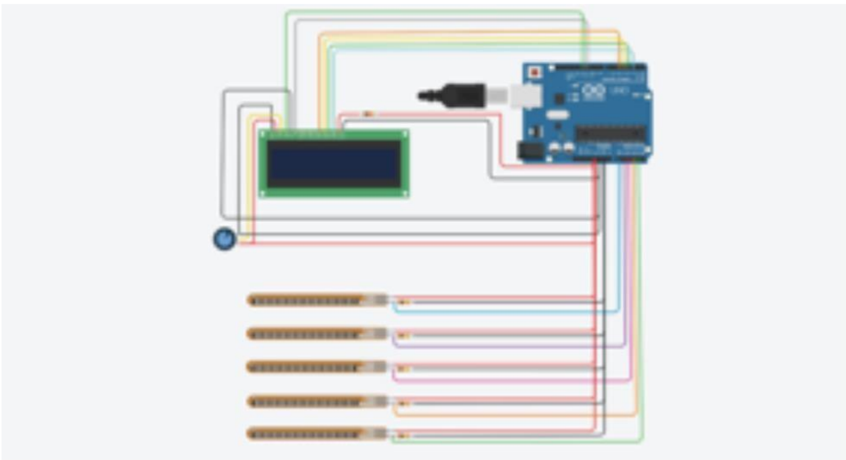


Figure 2.5: Full Circuit Diagram of Sensor-Based [14].

Figure 2.5 shows the technology exhibited a commendable level of precision in identifying and transforming sign language motions into written text. The flex sensors successfully recorded the finger motions and bending, enabling precise identification of hand gestures. The Arduino Uno microcontroller evaluated the sensor data and accurately correlated the motions with their corresponding textual representations.

This paper proposes a system to aim for using the Arduino Uno and flex sensors in the sign-to-text system, which shows encouraging outcomes in converting sign language motions into text. Technology provides a viable remedy to bridge the communication divide between those with speech problems and the broader populace. Subsequent research and development efforts should prioritize the improvement of the system's precision, extending its capacity to recognize gestures, and investigating other functionalities to augment its usefulness and efficacy in practical situations [14].

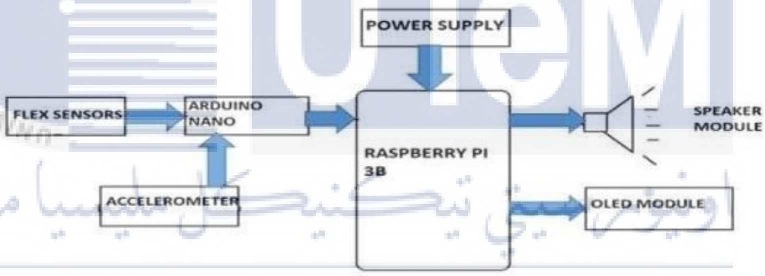


Figure 2.6: Block Diagram of Purposed System [15].

The ultimate prototype is created in Figure 2.6 using the block diagram and it incorporates flex sensors that are put into the glove. The sign's unique value is determined by combining the acceleration values of the axes from the accelerometer to the ADC converter. This is necessary because the acceleration values are analog and need to be converted to digital format for processing by the Raspberry Pi. The output is then displayed as text on the OLED module and as speech through the audio output connected to the Raspberry Pi 3B via the 3.5mm jack**.**

This paper introduces the flex sensor, together with the accelerometer, ADC converter, and Raspberry Pi, effectively and precisely translates American Sign Language (ASL) into text and voice. By affixing these sensors onto a glove, a very handy and user-friendly wearable device is created that is not only effective but also provides comfort for our everyday activities. It offers an effective approach to addressing the challenges faced by those with speech impairments. It gives individuals the ability to communicate effectively and enables them to articulate their thoughts and feelings more effectively [15].

2.4 Accelerometer in Sign Language Recognition

The accelerometer plays a significant role in sign language recognition, particularly for dynamic gestures like the letters "J" and "Z" in American Sign Language (ASL). These letters involve movements that are not easily captured by static sensors like flex sensors or resistive sensors. The accelerometer, which measures the acceleration of the hand or fingers, helps to detect the dynamic movements involved in these signs. Figure 2.7 shows that example of a fingerspelling prototype using the accelerometer.

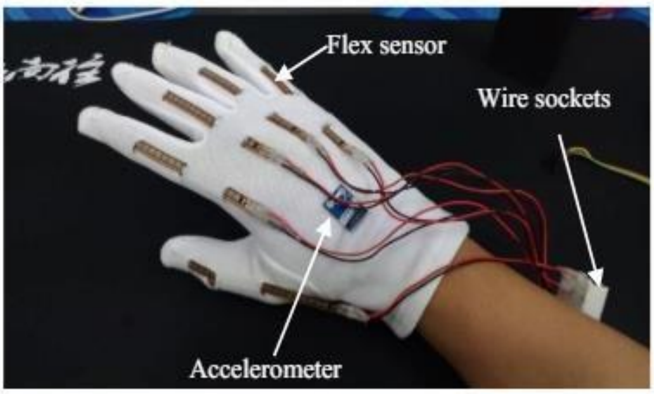


Figure 2.7: Prototype using Accelerometer [16]

Similarly, in Figure 2.7 the development of a smart glove for sign language recognition, the accelerometer was used in conjunction with flex sensors to capture both the bending of fingers and the movement of the hand. This combination of sensors allowed for more accurate recognition of dynamic signs like "J" and "Z"[17]. In summary, the accelerometer is an essential component in sign language recognition systems, especially for dynamic gestures like "J" and "Z", as it helps to capture the movement and acceleration of the hand or fingers involved in these signs.

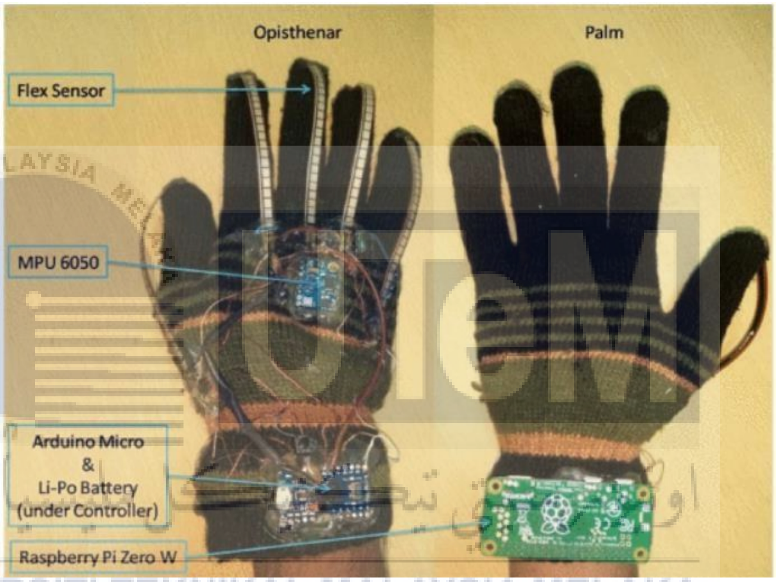


Figure 2.8: Design of Smart Glove with Accelerometer [18].

Figure 2.8 has affixed an accelerometer and gyroscope (MPU6050) to the central region of the Opisthenar, which will track the hand's orientation. The flex sensors are positioned on each finger and provide analog data depending on the bending moments of the fingers to the Arduino Micro board, which is located on the wrist. The Arduino board collects data from all sensors, which are originally analog. The data is converted into digital format using an Analog-to-Digital Converter (ADC), undergoes pre-processing, and is then sent over UART to a Raspberry Pi Zero W board for internal calculation. The Dynamic Time Warping (DTW) and Nearest Mapping Algorithm are implemented in a Python script on the Raspberry Pi Zero W. The data is transformed into English Alphabet based on the training database.

2.5 Summary of Literature Review

Table 2.1: Summary of Literature Review.

SYSTEM	Automated Sign Language Interpreter Using Data Gloves [19].	American Sign Language Translation via Smart Wearable Glove Technology [20].	Gesture Recognition Glove for American Sign Language Using Accelerometer s [21].	Development of Sign Language using Flex Sensors [22].	Sensor-Based Hand Gesture Recognition System for English Alphabets Used in Sign Language of Deaf-Mute People [23].
INPUT	Flex sensors, Contact sensors.	Flex sensors, Accelerometer s.	Accelerometer s, Flex sensors.	flex sensors, acceleromete r.	Flex Sensor, MPU 6050
OUTPUT	PC display	Android Application Wirelessly	Text And Audio	Android Application, Text and Speech.	Data in Matlab.
PLATFORM	Ardui no UNO, Arduino IDE.	Arduino Leonardo, android application.	Arduino IDE, Arduino.	Arduino,	Matlab, Arduino Micro board, Raspberry Pi Zero W.
LIMITATION	The system still has room for improvement and further optimization.	The failures are attributed to sensor errors, particularly while changing the hand signals.	Gestures for the letters 'J' and 'Z' in American Sign Language are dynamic and involve motion of the wrist while making the gesture and hence are out of the scope of the project.	May be replaced by a customized ASIC which may take care of all required functionality s including computation.	The accuracy value of translation can be increased by increasing the number of sensors.

ADVANTAGE	It is an economically feasible, easy-to-use device that can help the deaf and dumb communities to overcome communicate on barriers.	Friendly mobile Android application for displaying and listening to the output.	The gloves operate with a 2-second delay between each gesture, allowing the user to gesture easily.	Sensors directly give the glove the bend value, and the output is available in a fraction of a second.	The device is wearable, mobile, and user-friendly.

Table 2.1 shows the summary of the Literature Review from another article. Most of the article focuses on fingerspelling prototypes using flex sensors. The platform that most authors use is Arduino board. Finger gestures from the glove will produce data and be sent to the microcontroller to determine the command to be executed [9]. The output from the article focuses on displaying the alphabetical using a PC display, an Android application, and text. All the article reviews have limitations and advantages of the project. The article can improve the project to become more accurate and more relevant to the industry.

CHAPTER 3

METHODOLOGY

Figure 3.1 below shows the flowchart of the project. To start the project, gathered all the needed components to make the prototype. Then, the resistance of each flex sensor is determined. Flex sensors are typically resistive sensors that change their resistance in response to the degree of bending or flexing. After that, the circuit of the prototype is configured by determining the component's layout and its connection and making it into a circuit diagram. In the process of program development for the prototype, a code is written to read the sensor signal for each alphabet. Subsequently, the prototype is tested to test the functionality. If the prototype is not working, then the prototype needs to be troubleshooted by going back to the circuit configuration step. However, if the prototype is working well, the next step is to characterize the fingerspelling by involving the use of a formula to convert the analog signal to a digital signal.

Then, the prototype's program is improved by rewriting the code to convert the analog signal to a digital signal. Recognition system testing is important after improving the program. It is for making sure the recognition accuracy of the alphabet. If there are issues arise, the characterization of the flex sensor reading step is repeated. Once the recognition system is operational, it is tested with 5 users. The 5 users performed fingerspelling from alphabet A to alphabet Z. The data from system testing is used to do system analysis and improvement. The system's effectiveness, functionality, and efficiency were assessed to align it with the intended objectives. From the analysis, the accuracy of each alphabet was calculated based on the performance of the 5 users. The alphabet was considered accurate if more than 3 users completed fingerspelling with the correct output. In case the system accuracy was unacceptable, the system testing step was repeated. Once the system accuracy met expectations, the project was concluded and officially closed.

3.1 Project Implementation Flowchart

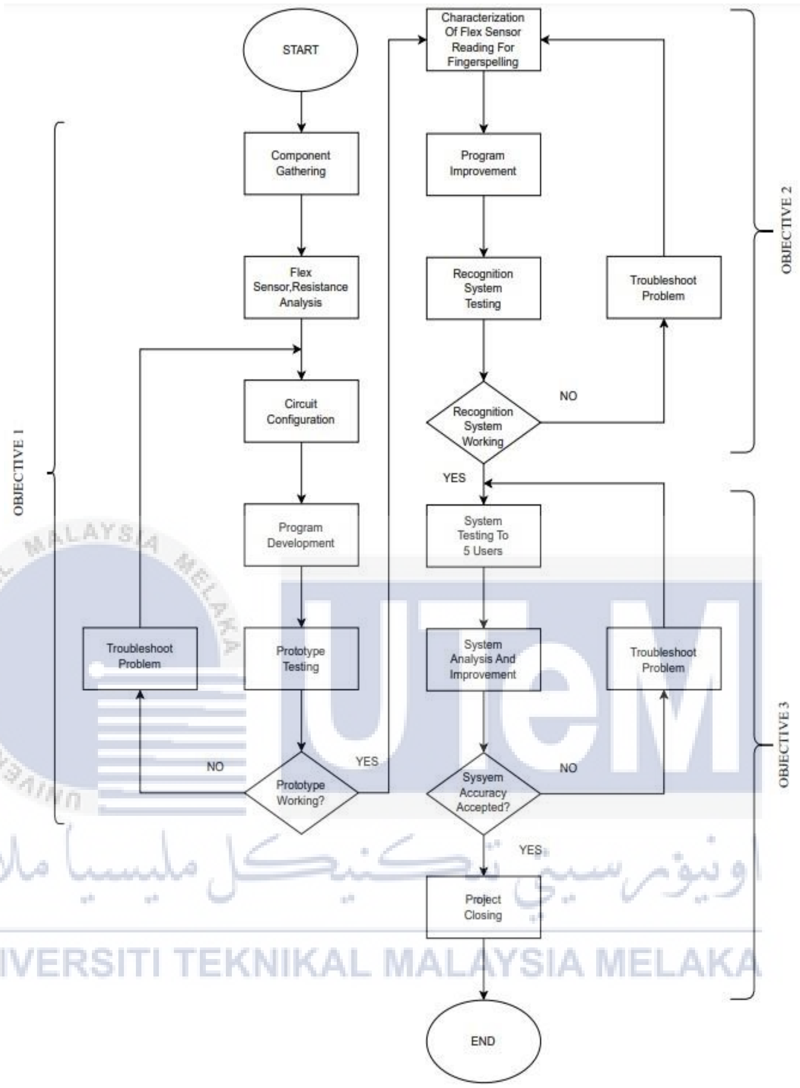


Figure 3.1: Flowchart of the Project.

3.2 Development of Fingerspelling Recognition System Prototype

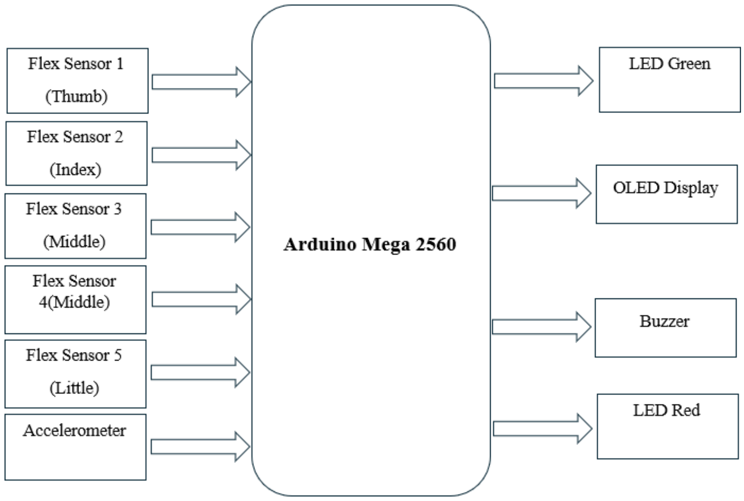


Figure 3.2: Block Diagram of Fingerspelling Recognition System.

This project interprets and displays hand movements using an Arduino Mega 2560 connected to several sensors and output devices shown in Figure 3.2. Five flex sensors, mounted to the thumb, index, middle, ring, and little fingers, give analog signals to the Arduino when each finger bends. An accelerometer gives additional motion information. The application uses these inputs to determine specific movements. Depending on the recognized motion, it sends data to an OLED display and initiates an auditory response. This configuration provides real-time feedback on hand movements, which might be useful in applications such as sign language interpretation or gesture-based control systems.

3.2.1 Hardware Development

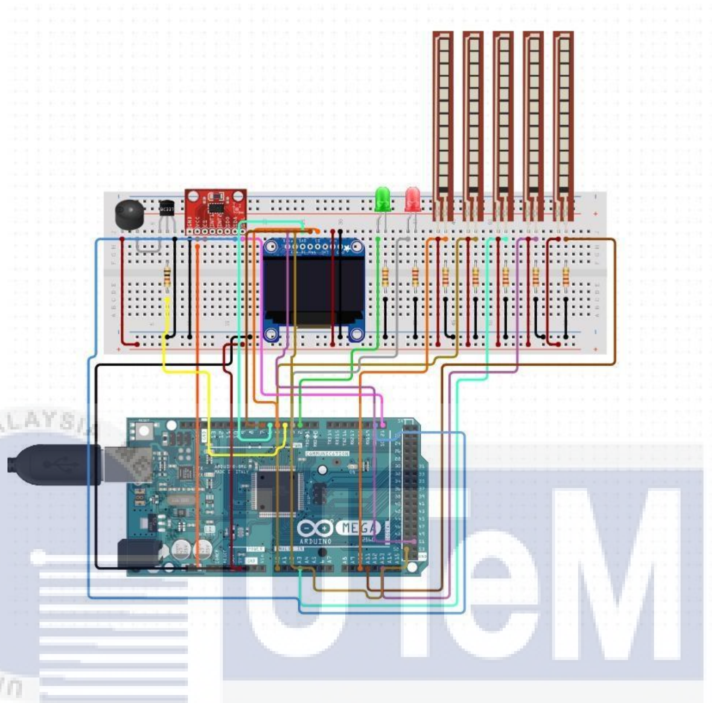


Figure 3.3: Full Circuit Diagram

Figure 3.3 shows the full circuit diagram of the analysis fingerspelling prototype recognition system. 5 flex sensors that represent each finger are used in this project. Each sensor's positive lead connects to an assigned analog input pin which is A0, A1, A2, A3, and A4 on the Arduino. Their negative leads share a common ground connection. OLED Display is connected via four wires SCL to pin SCL20, SDA to pin SDA21, VCC to 5V, and GND to ground. This enables communication and data transfer between the Arduino and the display. Both LEDs connect positively to separate pins which are green to pin 8 and red to pin 9. Their negative leads share a common ground connection. At the same time, the buzzer is directly connected to pin 13 for power and to ground for completion of the circuit, enabling its operation when triggered by the Arduino. Lastly, the accelerometer connects with GND, VCC, SDA 20, and SDL 21.

Table 3.1: List Pin Output Component.

NO.	COMPONENT	PIN	ARDUINO MEGA PINOUT
1	FLEX SENSOR	Flex Sensor1 Flex Sensor2 Flex Sensor3 Flex Sensor4 Flex Sensor5	A0 A1 A2 A3 A4
2	LED GREEN	Positive Negative	8 GND
3	LED RED	Positive Negative	9 GND
4	OLED DISPLAY	SCL SDA VCC GND	SCL 21 SDA 20 5V GND
5	BUZZER	Positive Negative	13 GND
6	ACCELEROMETER	VCC GND SDA SCL	5V GND SDA 20 SCL 21

Table 3.1 shows the list of pin output components of the analysis fingerspelling prototype recognition system. 5 flex sensors that represent each finger are used in this project. Each sensor's positive lead connects to an assigned analog input pin which is A0, A1, A2, A3, and A4 on the Arduino. Their negative leads share a common ground connection. OLED Display is connected via four wires SCL to pin SCL20, SDA to pin SDA21, VCC to 5V, and GND to ground. This enables communication and data transfer between the Arduino and the display. Both LEDs connect positively to separate pins which are green to pin 8 and red to pin 9. Their negative leads share a common ground connection. At the same time, the buzzer is directly connected to pin 13 for power and to ground for completion of the circuit, enabling its operation when triggered by the Arduino. Lastly, the accelerometer connects with GND, VCC, SDA 20, and SDL 21.

Table 3.2: List of Cost

COMPONENT	DESCRIPTION	QUANTITY	PRICE PER ITEM	TOTAL PRICE
Glove	-	1	RM 2.40	RM 2.40
Arduino Mega 2560	-	1	RM 219.00	RM 219.00
Flex Sensor	4.5”	5	RM 95.00	RM 475.00
LED	RED GREEN	1 1	RM 0.10 RM 0.10	RM 0.10 RM 0.10
Resistor 10k	10KΩ	5	RM 0.20	RM 1.00
Resistor 1k	1KΩ	2	RM 0.10	RM 0.20
Jumper Wire	MALE TO MALE MALE TO FEMALE	20 20	RM 0.25 RM 0.30	RM 5.00 RM 6.00
Buzzer	-	1	RM 1.00	RM 1.00
OLED display	128x64 12C	1	RM 35.00	RM 35.00
Accelerometer	ADXL345	1	RM 8.00	RM 8.00

Table 3.2 shows a comprehensive cost of different components necessary for a project, including the component's name, description, price, and overall cost. The kit comprises a glove, an Arduino Mega 2560 microcontroller, flex sensors, red and green light-emitting diodes (LEDs), 10kΩ and 1kΩ resistors, male-to-male and male-to-female jumper cables, a buzzer, an OLED display, and an accelerometer (ADXL345). The flex sensors and the Arduino Mega 2560 are the most costly elements, and the LEDs and resistors are the least expensive items. The total cost is determined by multiplying the amount by the price per item for each component.

3.2.1 Hardware development

3.2.1.1 Arduino Mega 2560

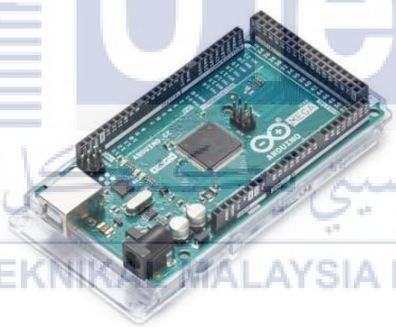


Figure 3.4: Arduino Mega 2560.

Figure 3.4 shows Arduino Mega 2560 is the main controller for this project. All the components will relate to the Arduino mega 2560 pinout. 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.

3.2.1.2 Flex Sensor

Figure 3.5: Flex Sensor.

Figure 3.5 shows the flex sensor of the 4.5-inch flex sensor. It has two terminals, which are denoted as p1 and p2. This sensor does not have any polarized terminals, such as a diode or a capacitor, hence there is no positive or negative terminal. The needed voltage for this sensor to operate varies from 3.3V to -5V DC, which can be obtained by any form of interface. This project focuses on the resistance value produced by flex sensors. The flex sensor will attach to the glove for every finger thumb, index, middle, ring, and little. The hand gestures for every alphabet have a different resistance value depending on the degree of bending the hand gesture of the alphabet.

3.2.1.3 LED

Figure 3.6: Green and Red LED

Figure 3.6 shows the Green and Red LEDs are semiconductor devices that emit light when an electric current flows through them. They are often constructed of materials like gallium arsenide or gallium phosphide. When a current flows through an LED, it excites the electrons in the material, causing them to emit energy in the form of light. The hue of light emitted by an LED is determined by the semiconductor's band gap.

3.2.1.4 Buzzer

Figure 3.7: Buzzer.

The buzzer has two wires. These cables link the buzzer to a circuit board. The circuit board generates the electrical current that causes the buzzer to vibrate. The voltage required to power the buzzer varies depending on the model, although they commonly operate between 3 and 12 volts shown in Figure 3.7.

3.2.1.5 Accelerometer

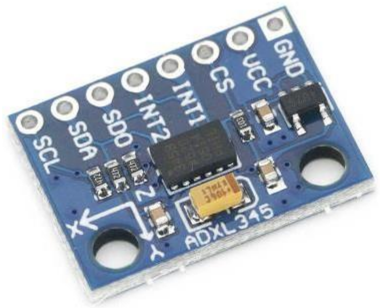


Figure 3.8: Accelerometer.

Figure 3.8 shown in the graphic shows a 3-axis accelerometer (ADXL345) that can measure acceleration in three dimensions (X, Y, and Z). The wording on the module's back marks the pins that can be connected to a microcontroller board like an Arduino or Raspberry Pi and programmed to read sensor data. The accelerometer will connect to the main board Arduino mega 2560 pin GND, VCC (5V), SDA (SDA20), and SCL (SCL21) only.

3.2.1.6 Printed Circuit Board (PCB) Design

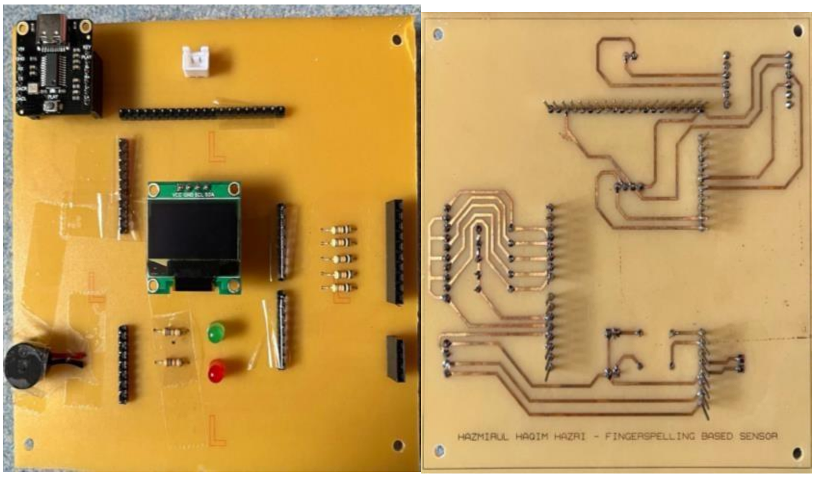


Figure 3.9: Complete Printed Circuit Board Design (PCB).

Figure 3.9 shows a PCB developed for a project that incorporates a variety of electronic components, including an OLED display, a green and red LED also connected to resistor 1kΩ, a buzzer, and headers for adding extra sensors such as a flex sensor connected with resistor 10kΩ for each flex sensor and an ADXL345 accelerometer. The primary OLED display produces a visual output of alphabets, while the LEDs offer status indicators green for true condition and red for false condition. The buzzer on the bottom left functions as an audible alert system if the alphabet is in true condition the buzzer will turn on. The board features various headers for connecting to an Arduino Mega 2560, allowing full control and data processing of the integrated components.

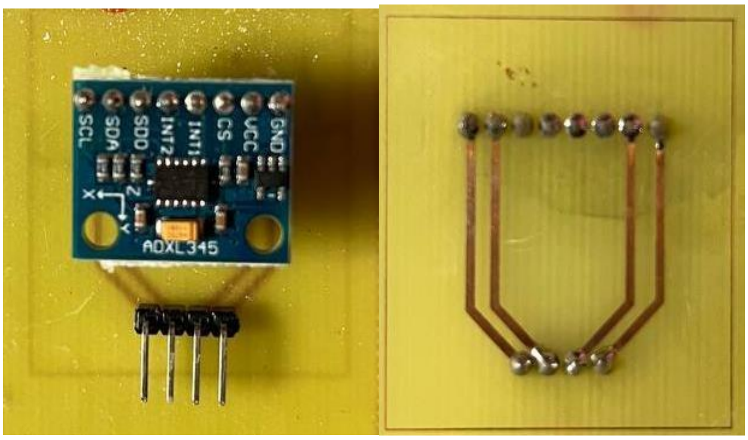


Figure 3.10: Printed Circuit Design Accelerometer.

Figure 3.10 shows a tiny PCB with an ADXL345 accelerometer module on it. The ADXL345 is a three-axis (x-axis, y-axis, and z-axis) accelerometer that measures acceleration with high resolution (13-bit) up to ±16 g. The module is connected to the PCB, with the pins extending below to connect to the main circuit. This arrangement is most commonly employed to detect the alphabet "J" and "Z". The ADXL345 communicates via I2C or SPI, allowing for simple integration with microcontrollers and other digital systems. On the back PCB, the accelerometer will connect to the main board Arduino mega 2560 pin GND, VCC (5V), SDA (SDA20), and SCL (SCL21) only.

3.2.2 Software Development

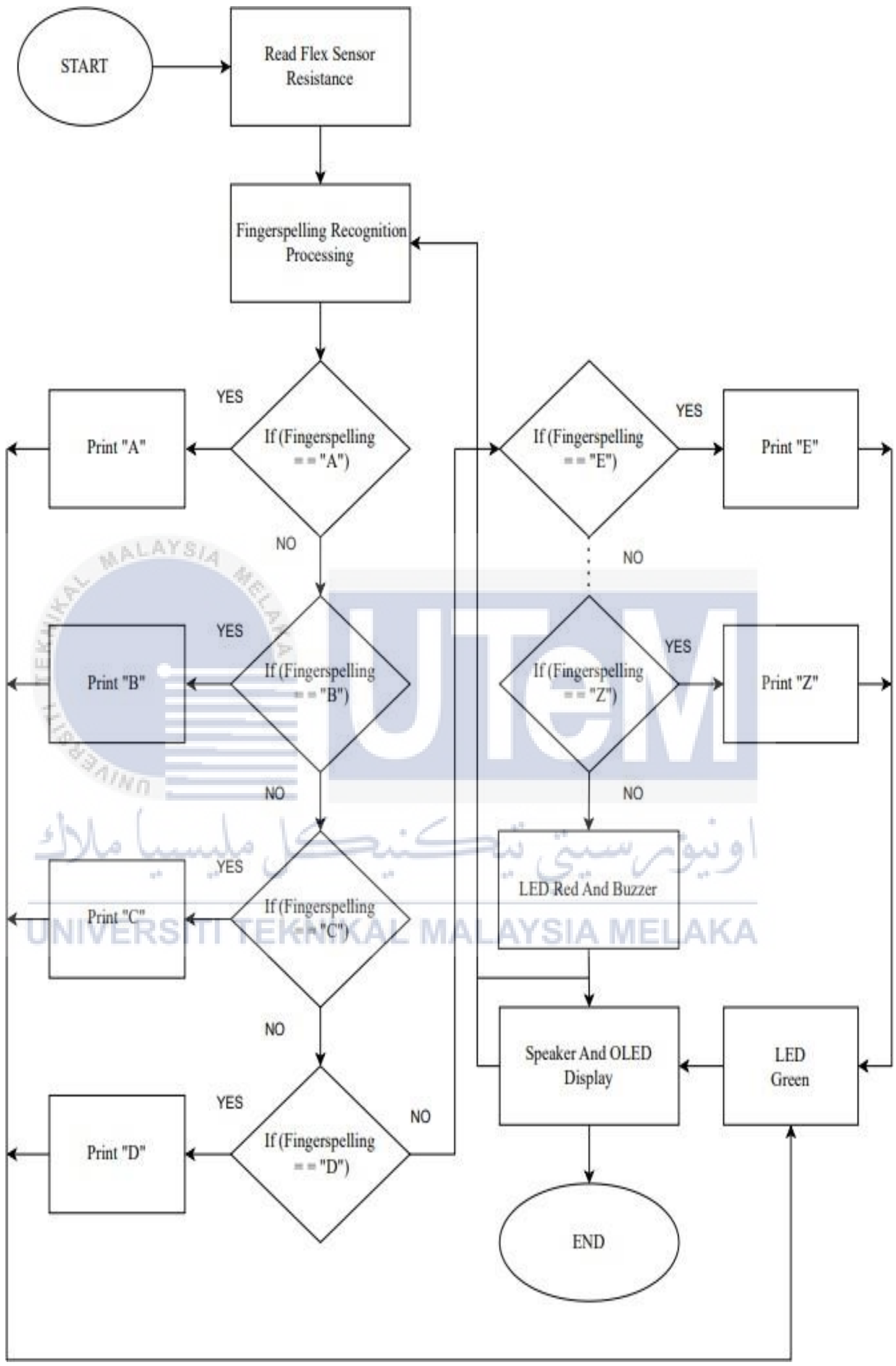


Figure 3.11: Flowchart of the Software Development

Figure 3.11 shows the flowchart of the software development for a fingerspelling prototype. The first step of the software development read the flex sensor. The flex sensor will produce a resistance value for each flex sensor. After that, the resistance value is determined by following the hand gesture of the fingerspelling prototype. The alphabet will print out depending on the resistance value of each finger. If the resistance value of the flex sensor for each finger is the same as the value of the fixed value for each finger for every alphabet written in the code, the green LED will be turned on and the output will display on the OLED display and audio on speaker. But if the resistance value is not the same as the value resistance for each alphabet, the red LED is turned on. The process will be turned back to determine the resistance value for each finger to detect the alphabet. Full coding can be seen in the appendix part.

```
if (a < 500 && b < 300 && c < 300 && d < 350 && e < 300 && a > 400) {
  display.setTextSize(2.0);
  display.setTextColor(SSD1306_WHITE);
  display.setCursor(50, 0);
  display.println("A");
  display.display(); // Update the OLED display with the new content
  digitalWrite(GREEN_LED_PIN, HIGH); // Turn on green LED for true condition
  digitalWrite(BUZZER_PIN, HIGH); // Turn on buzzer for true condition
  delay(2500); // Delay to display "A"
  display.clearDisplay();
} else {
  digitalWrite(RED_LED_PIN, HIGH); // Turn on red LED for false condition
}
```

Figure 3.12: Program for Display Alphabet "A".

Figure 3.12 shows the program for the Alphabet "A".This Program will continue depending on each of the alphabet. From this program, every finger has an ADC value to indicate the value for each alphabet. This program evaluates a series of conditional statements containing five variables (a, b, c, d, and e) to see if certain conditions are met. If all requirements ($a < 500$, $b < 300$, $c < 300$, $d < 350$, $e < 300$, and $a > 400$) are met and displays the letter "A" on an OLED display. It then activates a green LED that will turn on, wait 2.5 seconds, and clear the screen. If any of the requirements are false, the program disables the display activities and instead illuminates a red LED to indicate that the criteria were not met. This program same for all alphabets but not for the alphabet “J” and “Z” because these two alphabets have movement hand gestures using the accelerometer.

```
  if(!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
    Serial.println(F("SSD1306 allocation failed"));
    for(;;);
  }
  display.clearDisplay();
  display.setTextSize(1.5);
  display.setTextColor(SSD1306_WHITE);
  display.setCursor(50, 0);
  display.println("HELLO");
  display.display();
  delay(2500);
  display.clearDisplay();
  display.setTextSize(1.5);
  display.setTextColor(SSD1306_WHITE);
  display.setCursor(0, 0);
  display.println("PLEASE WEAR GLOVE");
  display.display();
  delay(2500);
  display.clearDisplay();
  display.setTextSize(1.5);
  display.setTextColor(SSD1306_WHITE);
  display.setCursor(0, 0);
  display.println("YOU CAN START MAKE   HAND GESTURE");
  display.display();
  delay(2500); // Pause for 2.5 seconds
  display.clearDisplay();

  for (int i = 0; i < num_ADC_pins; i++) {
    pinMode(ADC_pins[i], INPUT);
  }
}

void loop() {
  display.clearDisplay(); // Clear the display buffer
```


Figure 3.13: Program Instruction for User.

Figure 3.13 shows the program of OLED display. This program informs the user to start the hand gesture using the prototype. The words “HELLO”, “PLEASE WEAR GLOVE”, and “YOU CAN START TO MAKE HAND GESTURE” will be displayed on an OLED display to make sure the user can start to make hand gestures. This word will display for 2.5 seconds for each of the words.

```
  // Display ADC values on OLED
  for (int i = 0; i < num_ADC_pins; i++) {
    int flex_ADC = analogRead(ADC_pins[i]);
    float flex_voltage = flex_ADC * VCC / 1024.0;
    float flex_resistance = (fixed_resistor * (VCC / flex_voltage - 1.0)) / 1000.0;

    // Print to serial
    Serial.print("ADC value for pin ");
    Serial.print(i);
    Serial.print(": ");
    Serial.println(flex_ADC);
    Serial.println();

    // Print to OLED display
    display.setTextSize(1);
    display.setTextColor(SSD1306_WHITE);
    display.setCursor(0, 10*i);
    display.print("ADC");
    display.print(i);
    display.print(": ");
    display.print(flex_ADC);

  }
  display.display(); // Update the OLED display
  delay(5000); // Delay for 5 seconds
}
```

Figure 3.14: Program Display ADC Value for each Flex Sensor.

Figure 3.14 shows the program for ADC Value for each flex sensor. This program reads and displays ADC data from several pins on flex sensors. It iterates through the number of ADC pins (num_ADC_pins), reading each pin's analog value using analogRead() and converting it to a voltage (flex_voltage). It then calculates the flex sensor's resistance (flex_resistance) using a preset resistor value. The application displays the ADC values on the serial monitor for debugging or logging purposes. It simultaneously configures the OLED display, altering text size and color, and places the cursor to print the ADC value for each pin. After iterating through all pins, it updates the OLED display with a 5-second delay before restarting the procedure.

```
  else if (a > 250 && b > 330 && c < 330 && d < 320 && e > 170 && a < 330 && b < 400 && c > 250 && d > 260 && e < 250 ==
event.acceleration.z > 2.0) {
    display.setTextSize(2.0);
    display.setTextColor(SSD1306_WHITE);
    display.setCursor(50, 0);
    display.println("Z");
    display. display(); // Update the OLED display with the new content
    digitalWrite(GREEN_LED_PIN, HIGH);
    digitalWrite(BUZZER_PIN, HIGH); // Turn on buzzer for true condition
    delay(3000); // Delay to display "Z"
    display.clearDisplay();
  }
```

Figure 3.15: Program Accelerometer for Display Alphabet “Z”.

Figure 3.15 shows the program for the alphabet “Z”. This program also use for the alphabet “J” because these two alphabets use an accelerometer to detect the movement of the hand gesture. This code snippet is part of an Arduino program that takes actions based on numerous sensor readings and an accelerometer's Z-axis value. It first determines whether the values of variables a, b, c, d, and e are within specific ranges and if the Z-axis acceleration (event.acceleration.z) is larger than 2.0. If these requirements are met, it shows the letter "Z" on the OLED display, illuminates a green LED, and sounds a buzzer for 2.5 seconds. Following this delay, the display is cleared. If the conditions are not met, it illuminates a red LED to indicate a false condition. This logic is presumably part of a bigger system that monitors and responds to certain sensor statuses.

3.3 Characterization of the Flex Sensor Reading According to Alphabet Sign

Figure 3.16: ADC Value for each Flex Sensor.

Characterization of the flex sensor reading according to alphabet sign is the ADC value for each flex sensor. Every flex sensor has a different ADC value depending on the resistance value shown in Figure 3.16. This value can be measured by coding using the Analog pin read at Arduino Mega 2560. The flex sensor will read the output pin in Arduino Mega 2560 (A0, A1, A2, A3, and A4). Figure 25 shows the resistance value for every flex sensor for the thumb, index, middle, ring, and little. From the resistance value, all alphabets can be characterized based on their resistance value when doing the hand gesture.

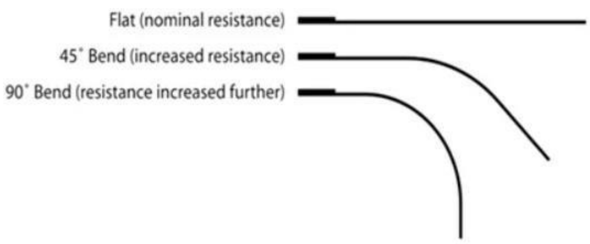


Figure 3.17: The Bent of Flex Sensor.

The main functioning principle of a flex sensor is that its electrical resistance changes with bending shown in Figure 3.17. When the sensor is flat, it displays the nominal resistance. As it turns, the resistance increases. When in the flat position, the sensor provides a baseline or nominal resistance. This is the lowest resistance condition. When the sensor is bent at a 45-degree angle, the conductive particles within it spread out or compress, increasing the resistance. The degree of growth is determined by the sensor's materials and design. At a 90-degree curve, the sensor receives even greater distortion. The conductive channels are stretched or compressed to a larger extent, which increases resistance.

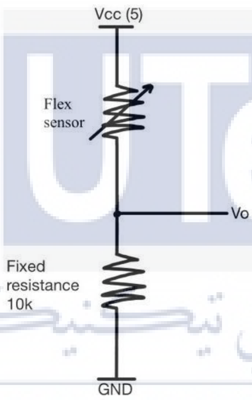


Figure 3.18: Voltage Divider Circuit.

Figure 3.18 shows the voltage divider circuit consisting of the flex sensor and a 10kΩ resistor. When the flex sensor is unbent (low resistance), the voltage across it (Vo) is larger according to the voltage divider principle. In contrast, when the sensor is bent (high resistance), Vo decreases. Equations 3.1 and 3.2 show the formula for calculating resistance value using Arduino IDE coding. The Arduino Mega 2560 Analog-to-Digital Converter (ADC) measures the voltage on the flex sensor output pin. When the sensor is unbent (with low resistance), the voltage increases. As the sensor bends (resistance increases), the voltage drops. The Arduino Mega 2560 ADC value will be higher for a straighter sensor (lower resistance) and lower for a bent sensor (higher resistance). This enables the translation of the sensor's bending into a digital value. This formula will show the resistance value for each flex sensor for every degree of bent.

$$\text{Flex voltage} = \text{flex ADC} * \text{VCC} / 1024.0 \tag{3.1}$$

$$\text{Flex resistance} = (\text{fixed resistor} * (\text{VCC} / \text{flex voltage} - 1.0)) / 1000 \tag{3.2}$$

The characterization has two parts first characterization ADC values and final characterization ADC values. The first characterization is to try for 5 different users and get the accuracy for 5 users. After that, the final characterization is the improvement from the first characterization. The improvement depends on the ADC value from the 5 different users and then from that the accuracy has been taken from 10 different users. All results can be seen in the appendix parts.

3.4 Analysis of the Accuracy of the Fingerspelling Recognition Prototype in Converting the Alphabet Sign

Analysis of the accuracy of the fingerspelling recognition prototype in converting the alphabet sign is the resistance value will be collected for each finger to demonstrate the alphabet. The resistance value of the flex sensor is taken for each finger for each alphabet. This is because, for every alphabet, the resistance value of each finger will be different. As the hand gesture for every alphabet is different, the value of the resistance will be different. The resistance value is different based on the degree of bent for each finger. From that resistance value, the programming will develop using the resistance value for each alphabet. The output alphabet will depend on the resistance value for each finger. The method to analyze the accuracy is by testing the prototype on 5 users. The output will have the information of the accuracy. For example, when user 1 makes the hand gesture for alphabet A, the monitor shows the same alphabet A which means the prototype is accurate. Other than that, it shows a different alphabet. Therefore, the alphabet A for user 1 is not accurate.

The accuracy has 2 parts first characterization and final characterization. First characterization testing to 5 users to get the accuracy of the alphabet and the user. For final characterization testing test 10 users to get the accuracy of the alphabet and the user. Final characterization is an improvement from the first characterization to check the accuracy of the alphabet and the user increase or decrease.

CHAPTER 4

RESULTS AND DISCUSSION

4.1 Prototype of Fingerspelling Recognition System

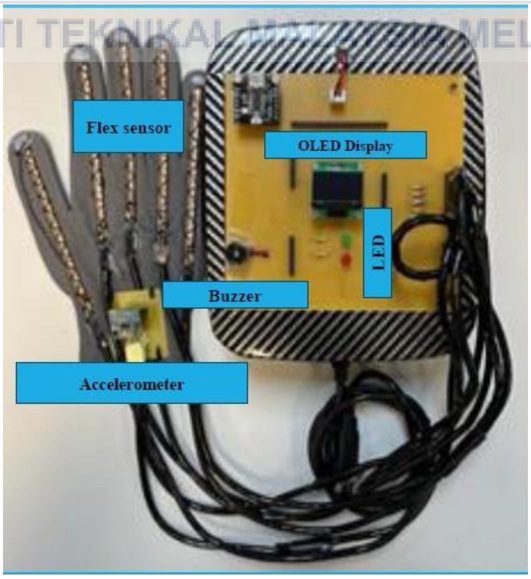


Figure 4.1: Complete prototype for fingerspelling recognition system.

Figure 4.1 depicts a wearable electronic glove system that is comprised of several different components. It contains a buzzer that will activate if the alphabet is in true condition, an OLED display that provides visual output, green and red LEDs that provide status indications for true or false situations, and an OLED display that provides visual output. The system is equipped with a header that allows for the connection of flex sensors and an accelerometer. This allows for the motion alphabet "J" and "Z" to be detected. This is because these two alphabets require the user to move their hands to identify the letter. The Arduino Mega 2560 microcontroller, which acts as the central processing unit for the management of sensor data, display output, and other operations, is the device that these components interface with.

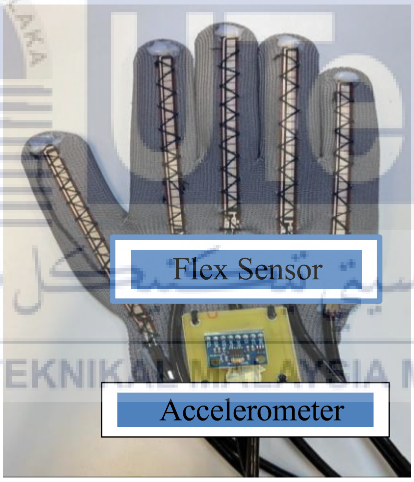
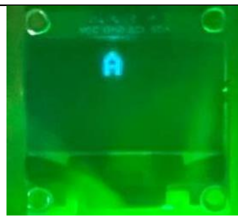
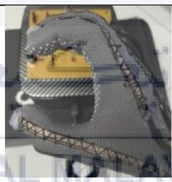
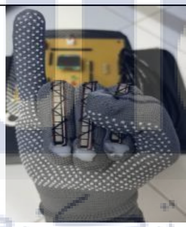
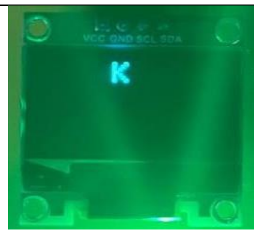


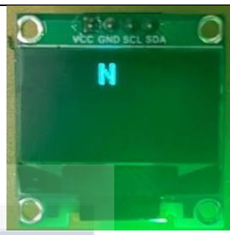
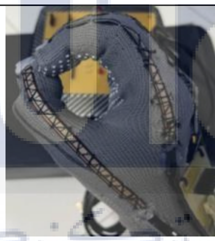
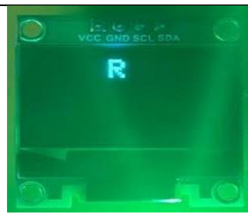
Figure 4.2: Prototype Glove Flex Sensor and Accelerometer.

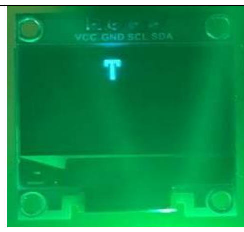
Figure 4.2 shows the flex sensors already attached to the glove and accelerometer. Five fingers thumb, index, middle, ring, and little have their flex sensor. Each flex sensor has two jumper wires that are soldered to it. One jumper wire is connected to VCC 5V and another jumper wire is connected to Arduino Mega A0, A1, A2, A3, and A4 to send input of resistance value. The function of this flex sensor that is attached to the glove is to produce the resistance value of each bend's finger. From the resistance value, all the alphabets will be determined by characterizing the resistance value for each fingerspelling alphabet. The function of the accelerometer is to detect the alphabet J and Z. These two alphabets need to do movement for the fingerspelling recognition system.

Table 4.1: Output on an OLED Display.

ALPHABET	HAND GESTURE	OUTPUT
A		
B		
C		
D		
E		

F		VCC GND SCL SDA F
G		VCC GND SCL SDA G
H		VCC GND SCL SDA H
I		I
J		VCC GND SCL SDA J
K		VCC GND SCL SDA K

L		 VCC GND SCL SDA L
M		 VCC GND SCL SDA M
N		 VCC GND SCL SDA N
O		 VCC GND SCL SDA O
P		 VCC GND SCL SDA P
Q		 VCC GND SCL SDA Q
R		 VCC GND SCL SDA R

S		VCC GND SCL SDA S
T		VCC GND SCL SDA T
U		VCC GND SCL SDA U
V		VCC GND SCL SDA V
W		VCC GND SCL SDA W
X		VCC GND SCL SDA X

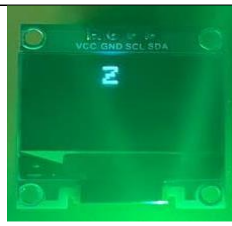
Y		
Z		

Table 4.1 shows the hand gestures for the alphabet A until Z. This table shows the output of the alphabet will be displayed on an OLED display. From Table 4 all alphabets have their hand gestures all users will make the hand gesture and the output will be displayed on an OLED display.

4.2 Flex Sensor Reading According to the Alphabet that the Fingerspelling Represents.

Table 4.2: First ADC Value for each alphabet

ALPHABET	THUMB	INDEX	MIDDLE	RING	LITTLE
A	433	253	272	283	224
B	316	489	511	501	446
C	369	386	379	420	427
D	343	469	305	336	276
E	289	261	277	308	249
F	286	290	476	494	440
G	432	418	254	268	199

H	322	476	479	304	268
I	288	246	280	336	423
J	367	306	333	367	446
K	331	481	502	342	277
L	431	468	281	293	248
M	296	264	316	306	228
N	364	270	282	291	221
O	343	344	337	380	387
P	397	480	482	431	202
Q	456	443	299	311	247
R	355	477	501	341	295
S	331	279	330	370	268
T	399	274	305	313	267
U	356	481	512	354	309
V	350	483	509	383	333
W	336	491	507	501	298
X	344	444	289	307	261
Y	432	333	372	394	448
Z	376	488	321	363	296

Table 4.2 shows the ADC value for each alphabet 1. This table for objective number 2 characterizes the flex sensor reading according to the alphabet that fingerspelling represents. This value will be set in range on the code to represent all the alphabet. If the user makes a hand gesture of the alphabet “A” and the ADC value

in a range of the value as in Table 4.2 the alphabet "A" will be displayed on an OLED display. This table shows how the output of the alphabet will be detected using the sensor-based. The Arduino mega 2560 will convert input from the flex sensor and become the ADC value.

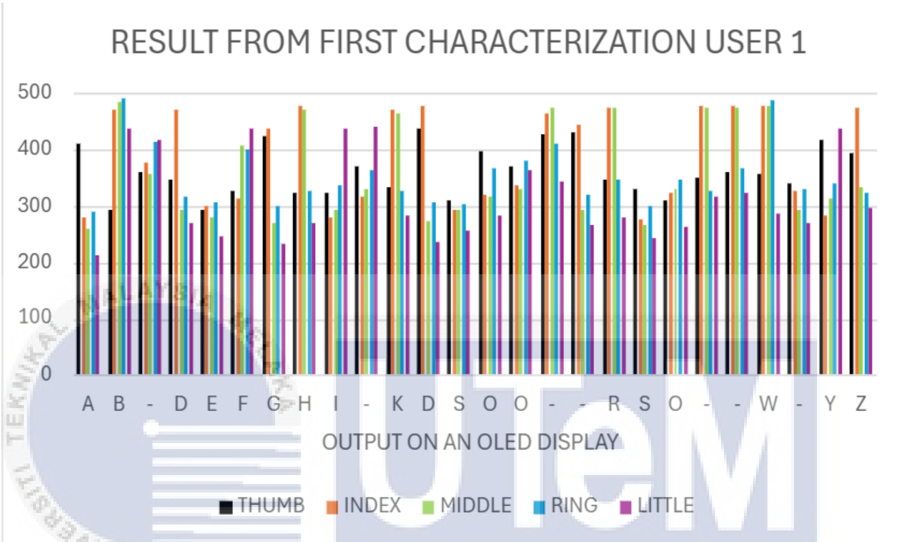


Figure 4.3: Result from First Characterization of User 1.

Figure 4.3 shows the graph result for user 1. The output on an OLED Display is not perfect from A to Z. Alphabet C, J, L, M, N, P, Q, T, U, V, and X are not on the graph. For the alphabet L, the output displays the alphabet D. Alphabet M the output displays the alphabet S. Alphabet N the output displays the alphabet O. Alphabet T the output displays the alphabet O. This is because the ADC Value for the alphabet is different from the ADC Value in Table 4.2. So, the alphabet display on an OLED Display is different with the user's hand gesture because of the ADC Value.

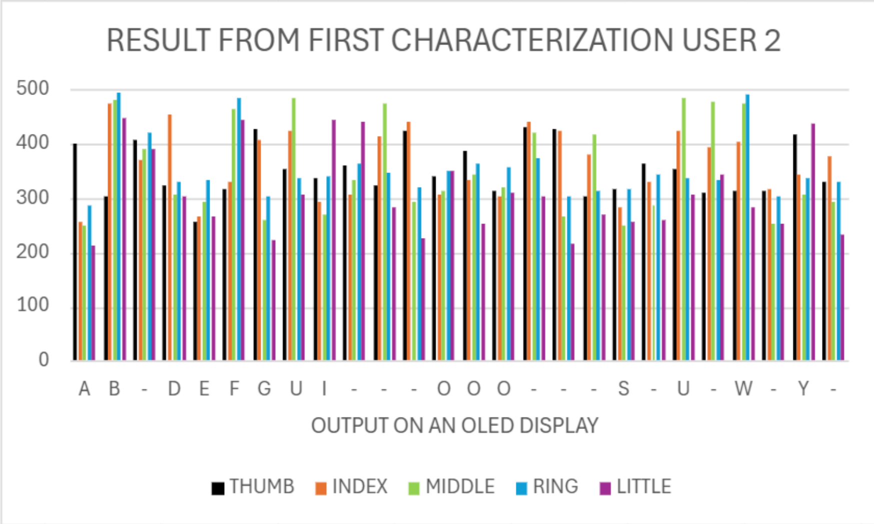


Figure 4.4: Result from First Characterization of User 2.

Figure 4.4 shows the graph result for user 2. The output on an OLED Display is not perfect from A to Z. Alphabet C, H, J, K, L, M, N, P, Q, R, T, V, X, and Z are not on the graph. For the alphabet H, the output displays the alphabet U. Alphabet M the output shows the alphabet O. Alphabet N the output displays the alphabet O. This is because the ADC Value for the alphabet differs from the ADC Value in Table 4.2. So, the alphabet display on an OLED Display differs from the user's hand gesture because of the ADC Value.

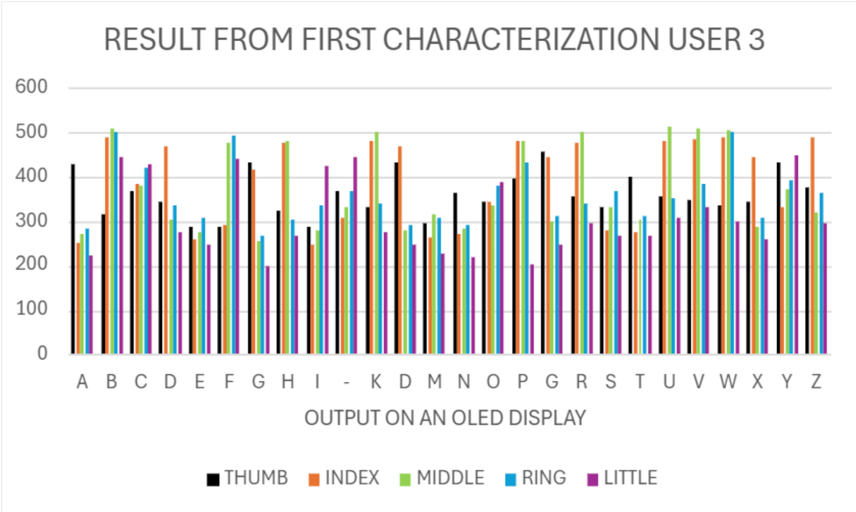


Figure 4.5: Result from First Characterization of User 3.

Figure 4.5 shows the graph result for user 3. The output on an OLED Display is not perfect from A to Z. Alphabet J, L and Q are not on the graph. For the alphabet L, the output displays the alphabet D. Alphabet Q the output shows the alphabet G. This is because the ADC Value for the alphabet differs from the ADC Value in Table 4.2. So, the alphabet display on an OLED Display differs from the user's hand gesture because of the ADC Value.

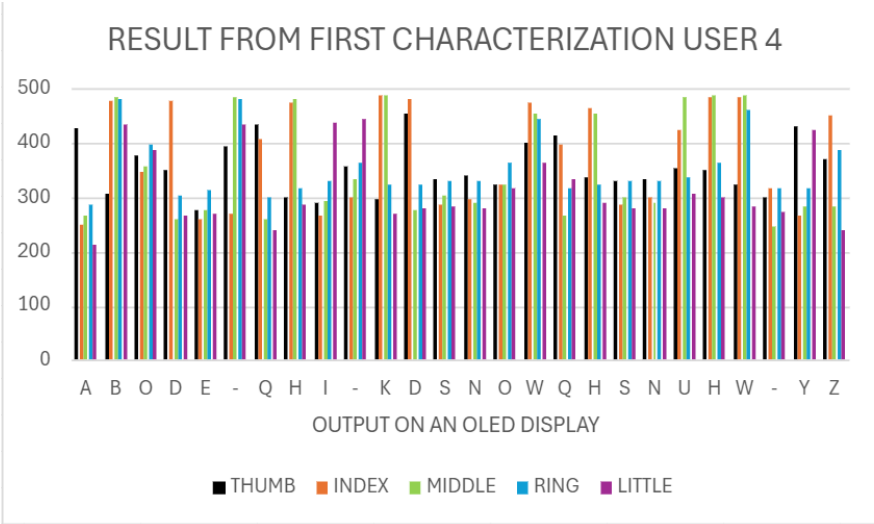


Figure 4.6: Result from First Characterization of User 4.

Figure 4.6 shows the graph result for user 4. The output on an OLED Display is not perfect from A to Z. Alphabet C, F, G, J, L, M, P, R, T, V and X are not on the graph. For the alphabet C, the output displays the alphabet O. For the Alphabet G the output shows the alphabet Q. Alphabet L the output displays the alphabet D. For the Alphabet M the output shows the alphabet S. For the Alphabet P the output shows the alphabet W. For Alphabet R the output shows the alphabet H. For Alphabet T the output shows the alphabet N. For Alphabet V the output shows the alphabet H. This is because the ADC Value for the alphabet differs from the ADC Value in Table 4.2. So, the alphabet display on an OLED Display differs from the user's hand gesture because of the ADC Value.

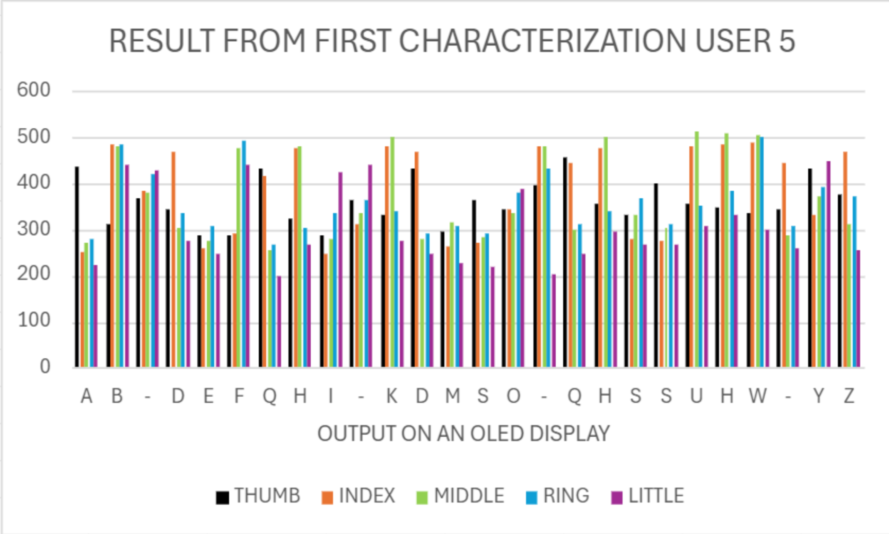


Figure 4.7: Result from First Characterization of User 5.

Figure 4.7 shows the graph result for user 5. The output on an OLED Display is not perfect from A to Z. Alphabet C, G, J, L, N, P, R, T, V, and X are not on the graph. For the alphabet G, the output displays the alphabet Q. Alphabet L the output shows the alphabet D. Alphabet N the output displays the alphabet S. Alphabet R the output on an OLED displays the alphabet H. Alphabet T the output on an OLED displays the alphabet S. Alphabet V the output shows the alphabet H. This is because the ADC Value for the alphabet differs from the ADC Value in Table 4.2. So, the alphabet display on an OLED Display differs from the user's hand gesture because of the ADC Value.

Table 4.3: Final ADC Value for each alphabet.

ALPHABET	THUMB	INDEX	MIDDLE	RING	LITTLE
A	403	250	254	287	209
B	320	479	482	488	443
C	387	351	346	357	377
D	341	467	311	279	260
E	314	271	283	300	243
F	340	281	478	486	439
G	411	390	245	257	206
H	378	474	477	315	252
I	323	299	300	327	437
J	320	315	317	340	438
K	361	477	472	282	236
L	448	452	252	278	224
M	328	261	282	284	223
N	364	252	271	241	216
O	323	308	316	307	293
P	368	439	361	307	227
Q	427	439	254	290	213
R	328	468	466	234	242
S	290	275	274	277	209
T	374	288	249	270	208

U	319	425	479	282	289
V	318	422	485	324	285
W	330	418	483	479	270
X	318	300	256	258	212
Y	441	368	304	337	438
Z	322	384	271	275	216

Table 4.3 shows the ADC value for each alphabet after improvement from Table 5. This table for objective number 2 characterizes the flex sensor reading according to the alphabet that fingerspelling represents. This value will be set in range on the code to represent all the alphabet. If the user makes a hand gesture of the alphabet "A" and the ADC value in a range of the value as in Table 6 the alphabet "A" will be displayed on an OLED display. This table shows how the output of the alphabet will be detected using the sensor-based. The Arduino mega 2560 will convert input from the flex sensor and become the ADC value.

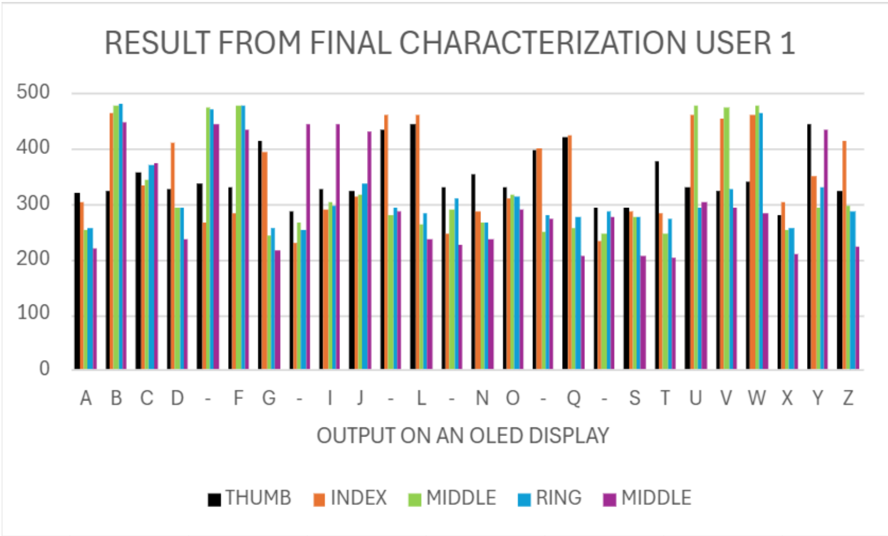


Figure 4.8: Result from Final Characterization of User 1.

Figure 4.8 shows the graph result for user 1 from the final characterization after the improvement shown in Table 4.3. The output on an OLED Display is not perfect from A to Z. Alphabet E, H, K, M, P, and R are not on the graph. This is because the ADC Value for the alphabet is different from the ADC Value in Table 4.3. So, the alphabet display on an OLED Display is different from the user's hand gesture.

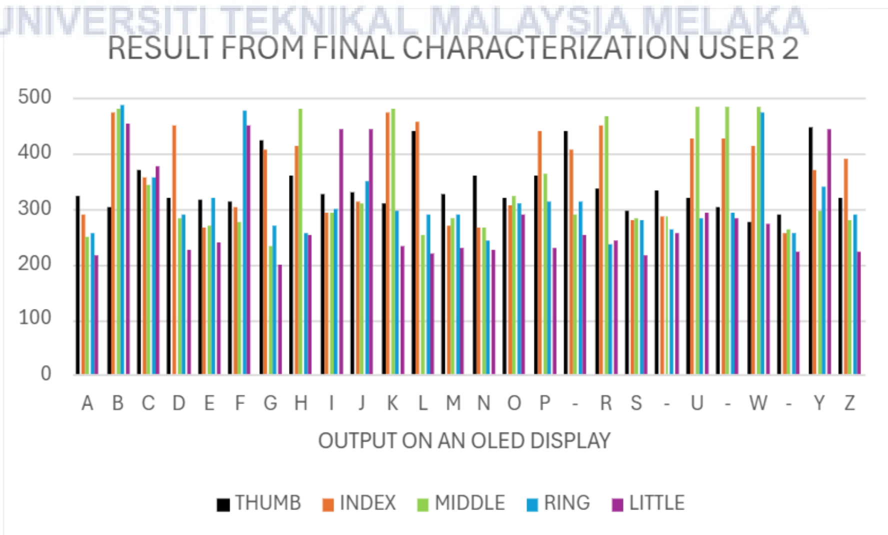


Figure 4.9: Result from Final Characterization of User 2.

Figure 4.9 shows the graph result for user 2 from the final characterization after the improvement shown in Table 4.3. The output on an OLED Display is not perfect from A to Z. Alphabet Q, T, V, and X are not on the graph. This is because the ADC Value from user 2 differs from the ADC Value in Table 4.3. So, the alphabet display on an OLED Display is different from the user's hand gesture.

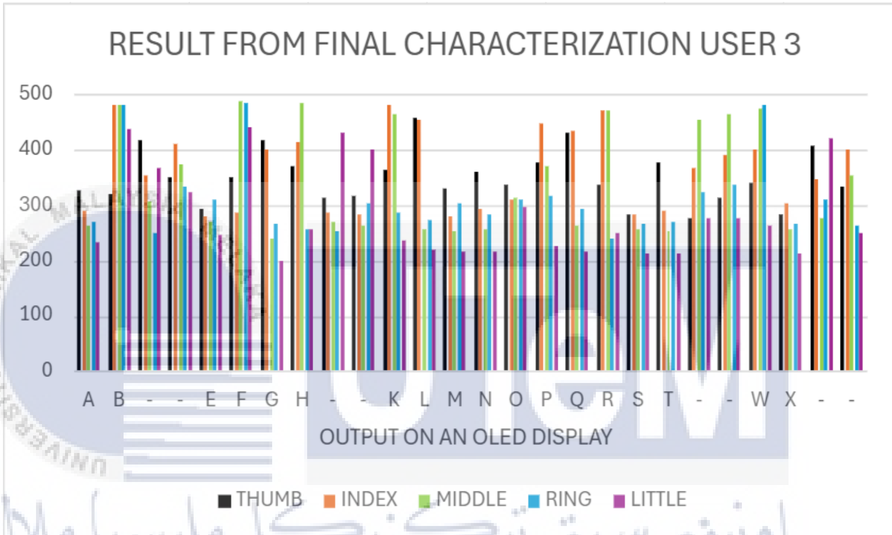


Figure 4.10: Result from Final Characterization of User 3.

Figure 4.10 shows the graph result for user 3 from the final characterization after the improvement shown in Table 4.3. The output on an OLED Display is not perfect from A to Z. Alphabet C, D, I, J, U, V, Y, and Z are not on the graph. This is because the ADC Value from user 3 differs from the ADC Value in Table 4.3. So, the alphabet display on an OLED Display differs from the user's hand gesture.

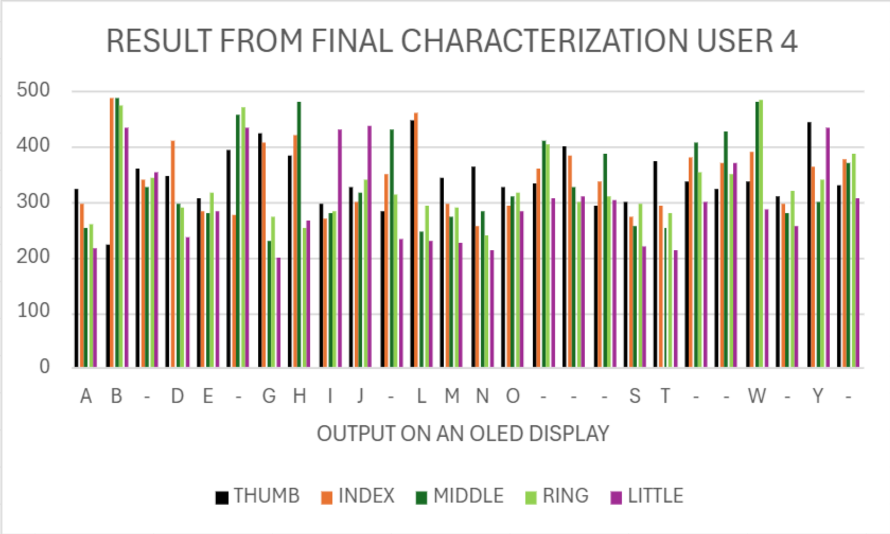


Figure 4.11: Result from Final Characterization of User 4.

Figure 4.11 shows the graph result for user 4 from the final characterization after the improvement shown in Table 4.3. The output on an OLED Display is not perfect from A to Z. Alphabet C, F, K, P, Q, R, U, V, X, and Z are not on the graph. This is because the ADC Value from user 4 differs from the ADC Value in Table 4.3. So, the alphabet display on an OLED Display differs from the user's hand gesture.

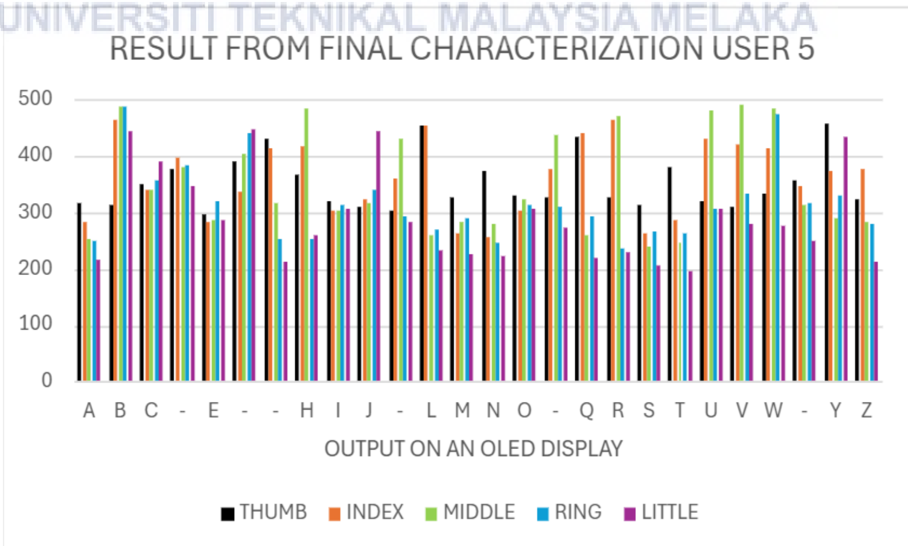


Figure 4.12: Result from Final Characterization of User 5.

Figure 4.12 shows the graph result for user 5 from the final characterization after the improvement shown in Table 4.3. The output on an OLED Display is not perfect from A to Z. Alphabet D, F, G, K, P, and X are not on the graph. This is because the ADC Value from user 5 differs from the ADC Value in Table 4.3. So, the alphabet display on an OLED Display differs from the user's hand gesture.

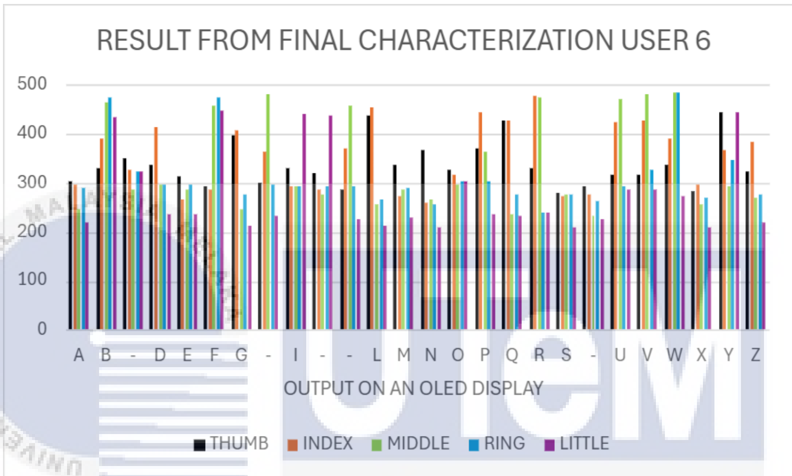


Figure 4.13: Result from Final Characterization of User 6.

Figure 4.13 shows the graph result for user 6 from the final characterization after the improvement shown in Table 4.3. The output on an OLED Display is not perfect from A to Z. Alphabet C, H, J, K, and T are not on the graph. This is because the ADC Value from user 6 differs from the ADC Value in Table 4.3. So, the alphabet display on an OLED Display differs from the user's hand gesture.

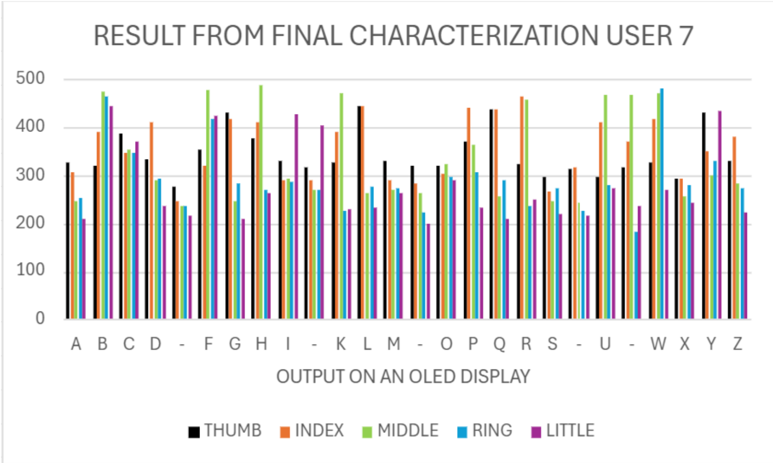


Figure 4.14: Result from Final Characterization of User 7.

Figure 4.14 shows the graph result for user 7 from the final characterization after the improvement shown in Table 4.3. The output on an OLED Display is not perfect from A to Z. Alphabet E, J, N, T, U, and V are not on the graph. This is because the ADC Value from user 7 differs from the ADC Value in Table 4.3. So, the alphabet display on an OLED Display differs from the user's hand gesture.

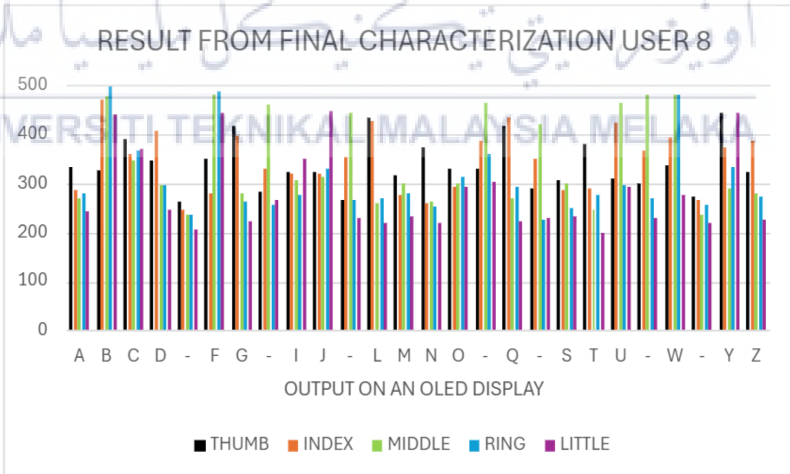


Figure 4.15: Result from Final Characterization of User 8.

Figure 4.15 shows the graph result for user 8 from the final characterization after the improvement shown in Table 4.3. The output on an OLED Display is not perfect from A to Z. Alphabet E, H, K, P, R, V, and X are not on the graph. This is because the ADC Value from user 8 differs from the ADC Value in Table 4.3. So, the alphabet display on an OLED Display differs from the user's hand gesture.

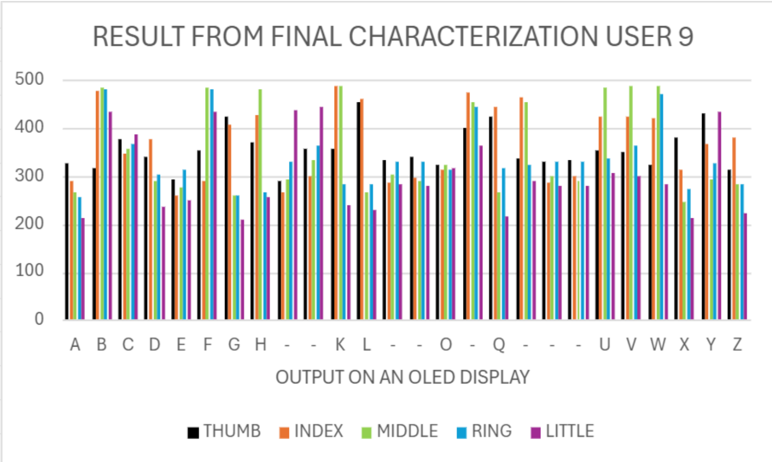


Figure 4.16: Result from Final Characterization of User 9.

Figure 4.16 shows the graph result for user 9 from the final characterization after the improvement shown in Table 4.3. The output on an OLED Display is not perfect from A to Z. Alphabet I, J, M, N, P, R, S, and T are not on the graph. This is because the ADC Value from user 9 differs from the ADC Value in Table 4.3. So, the alphabet display on an OLED Display differs from the user's hand gesture.

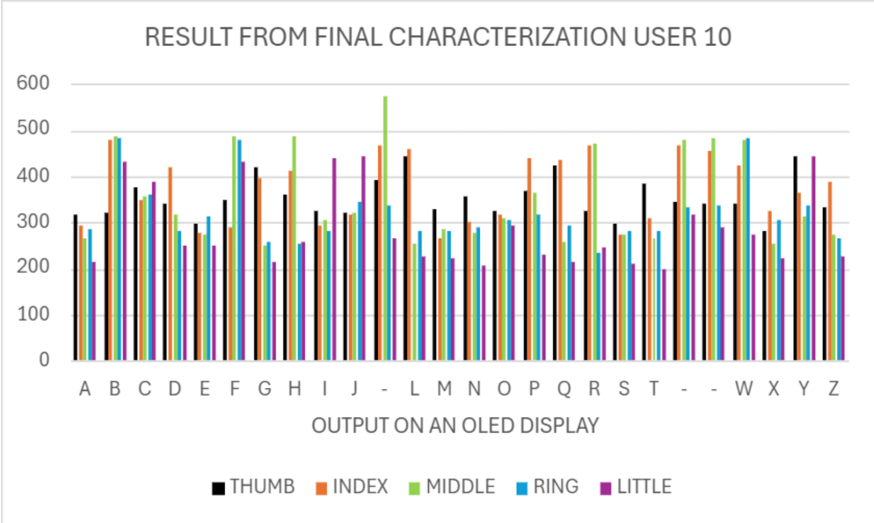


Figure 4.17: Result from Final Characterization of User 10.

Figure 4.17 shows the graph result for user 10 from the final characterization after the improvement shown in Table 4.3. The output on an OLED Display is not perfect from A to Z. Alphabet K, U, and V are not on the graph. This is because the ADC Value from user 10 differs from the ADC Value in Table 4.3. So, the alphabet display on an OLED Display differs from the user's hand gesture.

4.3 Accuracy of the Fingerspelling Recognition Prototype in Converting the Alphabet Based on Five User

Table 4.4: First Accuracy for each user.

USER	ACCURACY
User1	58%
User2	42%
User3	96%
User4	50%
User5	62%

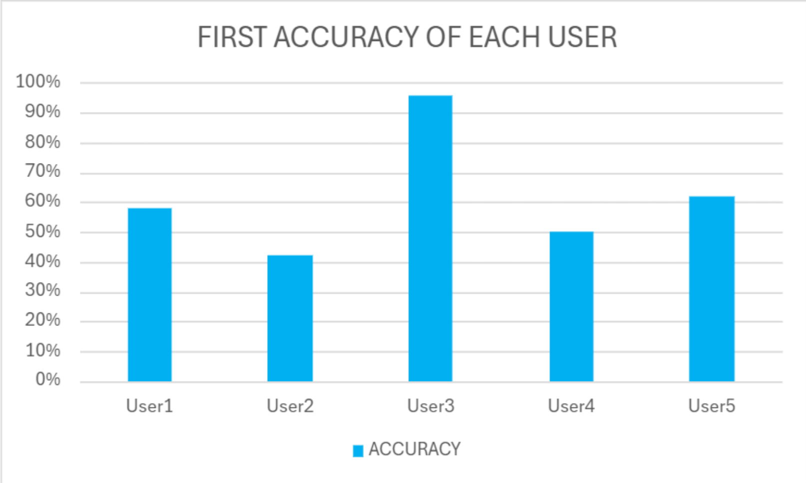


Figure 4.18: First Graph for Accuracy of each user.

The accuracy percentages of five distinct users, identified as Users 1 through User 5, are shown in Figure 4.18. Accuracy percentages are shown on the Y-axis and range from 0% to 120%; however, accuracy is usually limited to 100%. User 1 accuracy is around 60%, User 2 is about 30%, User 3 is the highest at almost 100%, User 4 is about 40%, and User 5 is about 60%. The accuracy levels of the users are shown in this bar chart, where User 3 performs noticeably better than the rest.

Table 4.5: First Accuracy of Alphabet.

ALPHABET	USER1	USER2	USER3	USER4	USER5	ACCURACY
A	A	A	A	A	A	100%
B	B	B	B	B	B	100%
C	-	-	C	O	-	20%
D	D	D	D	D	D	100%
E	E	E	E	E	E	100%
F	F	F	F	-	F	80%
G	G	G	G	Q	Q	60%
H	H	V, U	H	H	H	80%
I	I	I	I	I	I	100%

J	-	-	-	-	-	0%
K	K	-	K	-	K	60%
L	D	-	D	D	D	80%
M	S	O	M	S	M	40%
N	O	O	N	N	S	40%
O	O	O	O	O	O	100%
P	-	-	P	W	-	20%
Q	-	-	G	Q	Q	40%
R	R	-	R	H	H	40%
S	S	S	S	S	S	100%
T	O	-	T	N	S	20%
U	-	-	U	U	U	60%
V	-	-	V	H	H	20%
W	W	W	W	W	W	100%
X	-	-	X	-	-	20%
Y	Y	Y	Y	Y	Y	100%
Z	Z	-	Z	Z	Z	60%

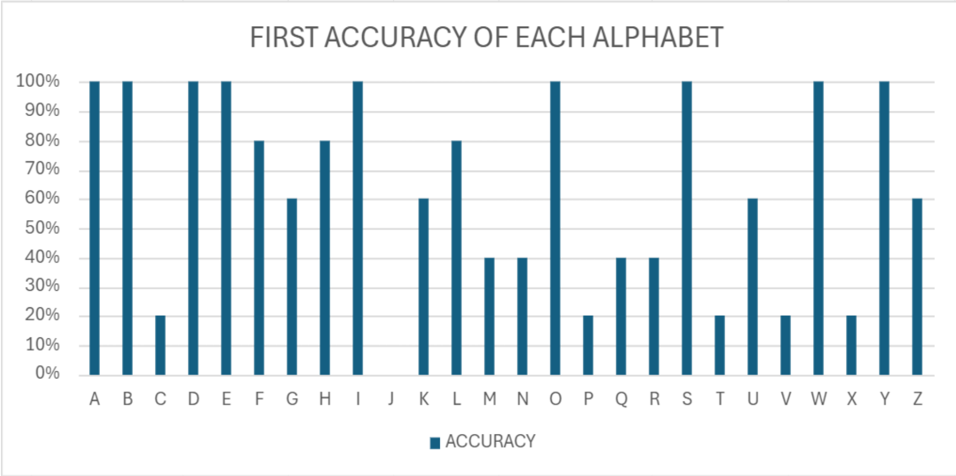


Figure 4.19: First Graph of Accuracy for Alphabet.

Figure 4.19 below displays the precision rates for different alphabetical letters. Each bar represents the correctness of a certain letter in the alphabet, ranging from A to Z. The vertical axis reflects accuracy, with values ranging from 0% to 100%, while the horizontal axis displays the alphabet. The data illustrates that certain letters, such as A, B, D, E, I, O, S, W, and Y, achieve an accuracy close to perfection. Conversely, the accuracy of letters such as C, P, T, V, and X is significantly lower, approximately 40% or less. The accuracy values for other letters vary, indicating a diverse range of performance levels throughout the alphabet.

Table 4.6: The Factor of Accuracy.

USER	MALE	FEMALE	SMALL HAND	BIG HAND
User 1	/		/	
User 2	/		/	
User 3	/			/
User 4		/	/	
User 5		/	/	

Table 4.6 shows the factor of accuracy that the user cannot get 100% accuracy of hand gestures. For users 1 and 2 small size of the hand can impact the accuracy of the

alphabet. This is because the ADC value of characterization for the small size of the hand differs from the big size of the hand. User 3 the accuracy is 96% because of the big size of the hand. So, the hand gesture of user 3 can be determined because the ADC value same as the value from chacterization. For users 4 and 5 same with users 1 and 2.

Table 4.7: Final Accuracy of each user.

USER	ACCURACY
USER 1	78%
USER 2	85%
USER 3	69%
USER 4	62%
USER 5	78%
USER 6	81%
USER 7	81%
USER 8	73%
USER 9	69%
USER 10	88%

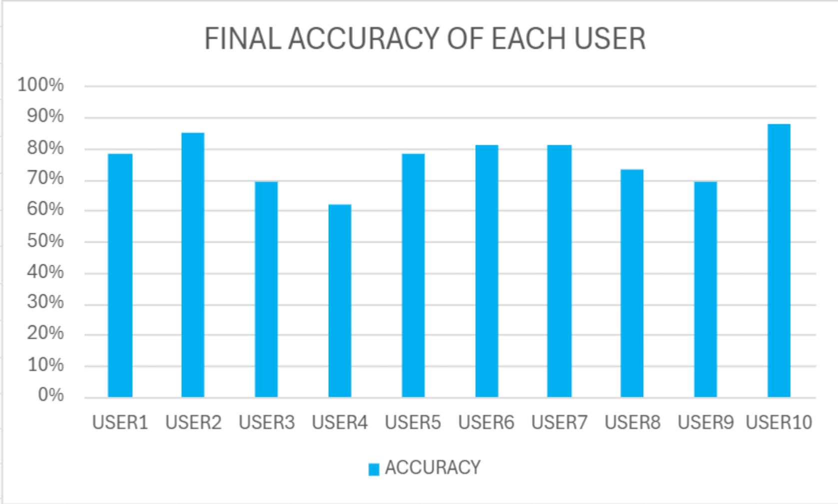


Figure 4.20: Final Graph for Accuracy of each user.

Figure 4.20 shows the "Final Accuracy of Each User". The bar chart presents the accuracy percentages for ten different users, numbered from User 1 to User 10. A blue bar is used to show the accuracy of each user. In terms of accuracy, User 10 achieves the maximum level, which is slightly around 80%, while User 4 achieves the lowest level, which is approximately 60%. Additionally, User 2 and User 6 demonstrate high levels of accuracy that are greater than 80%, although the majority of users fall somewhere between 60% and 80%. In general, the pattern indicates that there is a degree of variety in performance across the users, with User 10 exhibiting a noticeable peak. Accuracy is represented in percentage terms along the vertical axis, which ranges from 0% to 100%.

Table 4.8: Final Accuracy of Alphabets.

ALPHABET	USER										ACCURACY
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	
A	A	A	A	A	A	A	A	A	A	A	100%
B	B	B	B	B	B	B	B	B	B	B	100%
C	C	C	-	-	C	-	C	C	C	C	70%
D	D	D	-	D	-	D	D	D	D	D	80%
E	-	E	E	E	E	E	-	-	E	E	70%
F	F	F	F	-	-	F	F	F	F	F	80%
G	G	G	G	G	-	G	G	G	G	G	90%
H	-	H	H	H	H	-	H	-	H	H	70%
I	I	I	-	I	I	I	I	I	-	I	80%
J	J	J	-	J	J	-	-	J	-	J	60%
K	-	K	K	-	-	-	K	-	K	-	40%
L	L	L	L	L	L	L	L	L	L	L	100%
M	-	M	M	M	M	M	M	M	-	M	80%
N	N	N	N	N	N	N	-	N	-	N	80%
O	O	O	O	O	O	O	O	O	O	O	100%
P	-	P	P	-	-	P	P	-	-	P	50%
Q	Q	-	Q	-	Q	Q	Q	Q	Q	Q	80%
R	-	R	R	-	R	R	R	-	-	R	60%
S	S	S	S	S	S	S	S	S	-	S	90%
T	T	-	T	T	T	-	-	T	-	T	60%
U	U	U	-	-	U	U	U	U	U	-	70%
V	V	-	-	-	V	V	-	-	V	-	40%
W	W	W	W	W	W	W	W	W	W	W	100%
X	X	-	X	-	-	X	X	-	X	X	60%
Y	Y	Y	-	Y	Y	Y	Y	Y	Y	Y	90%
Z	Z	Z	-	-	Z	Z	Z	Z	Z	Z	80%

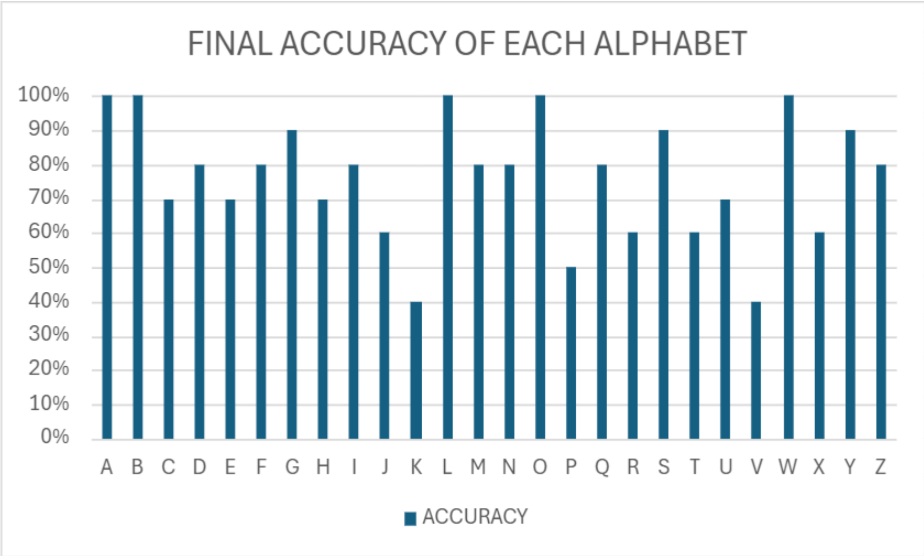


Figure 4.21: Final Graph of Accuracy for Alphabet.

Figure 4.21 displays a bar chart titled "Final Accuracy of Each Alphabet." The accuracy percentages are recorded for each letter of the alphabet, starting with A and ending with Z. Each letter is accurately depicted by a blue bar, while the vertical axis shows a percentage scale ranging from 0% to 100%. The letters A, B, L, O, and W exhibit the highest level of accuracy, achieving a flawless score of one hundred percent, indicating that there are no errors in their identification or categorization. In contrast, the letters K, P, and V have much lower accuracies ranging from forty percent to fifty percent, suggesting a high frequency of inaccuracy. Various letters demonstrate varying levels of precision, with the majority ranging from sixty percent to ninety percent, indicating a combination of successful and unsuccessful outcomes. The data clearly illustrates a significant disparity in accuracy among different letters, with certain letters being recognized with much greater dependability than others.

Table 4.9: Average Accuracy.

AVERAGE ACCURACY	FIRST CHARACTERIZATION	FINAL CHARACTERIZATION
USER-BASED	62%	76%
ALPHABET-BASED	63%	76%

Table 4.9 shows the accuracy of two characterization techniques, user-based and alphabet-based, at two distinct stages: initial characterization and final characterization. Both methods exhibit comparable accuracy at the initial characterization step, with the user-based method obtaining 62% accuracy and the alphabet-based method achieving 63% accuracy. At the end of the characterization step, both techniques achieve a comparable accuracy level of 76%, demonstrating a progressive enhancement over time. This implies that irrespective of the initial approach employed, the ultimate precision reaches a common value.

CHAPTER 5

CONCLUSION AND FUTURE WORKS

5.1 Conclusion

In conclusion, the first objective of the study, which aimed to configure a prototype of a fingerspelling recognition system using flex sensors, has been accomplished. The first objective is to develop a prototype of a fingerspelling recognition system using sensor-based. This prototype uses a flex sensor as a sensor-based, and the Arduino mega 2560 as a microcontroller and output OLED display, green LED, and Buzzer.

Additionally, the second objective of the study, which aimed to Characterize the flex sensor reading according to the alphabet that the fingerspelling represents, has been accomplished. For second objective has 2 parts which are the first characterization and the final characterization of ADC value. The final characterization is an improvement from the first characterization to increase accuracy.

The final objective of the experiment, which aimed to Analyse the accuracy of the fingerspelling recognition prototype in converting the alphabet based on different users has been accomplished. The third objective is the analysis of fingerspelling prototypes for different users. The analysis part focuses on the accuracy of the alphabet and the users. The accuracy of the alphabet and the user depends on the first characterization and final characterization. The accuracy of the alphabet and the user increases from the first characterization to the final characterization.

In conclusion, integrating flex sensors with fingerspelling recognition systems provides a practical and efficient alternative for assisting speech-impaired people and improving communication with the deaf community. When integrated into gloves or wearable devices, Flex sensors enable the translation of hand motions into text data, allowing for real-time communication using sign language recognition systems. These systems use the analog signals produced by flex sensors to detect finger movements and gestures and turn them into intelligible text or audio output. Using sensor-based flex sensors in fingerspelling recognition systems represents a significant step forward in assistive technology, providing a practical and cost-effective solution for improving communication and interaction for people with speech impairments or who use sign language. The research and development in this area aims to improve these systems' accuracy, efficiency, and use, thereby increasing the quality of life and communication for people with hearing and speech problems.

5.2 Future Works and Limitations

The limitation of this project is the size of the hand can affect the accuracy of each alphabet. For future work, to obtain good accuracy researchers could investigate mixing a mixture of flex sensors for small hands and big hands using 2 types of flex sensors 2.2inch and 4.5inch and adding other sensor types such as accelerometers or gyroscopes to capture a more comprehensive range of hand movements and gestures, leading to more accurate recognition systems. Furthermore, the goal is to optimize the processing speed of fingerspelling recognition systems to enable real-time gesture recognition, allowing users to communicate seamlessly and instantly. Lastly, may concentrate on making fingerspelling recognition systems more accessible and affordable for a larger range of users, ensuring inclusivity and widespread use of assistive technologies.

REFERENCES

[1] A new data glove approach for Malaysian sign language detection. (n.d.). https://cyberleninka.org/article/n/591340/viewer.

[2] M. S. Altememe and N. K. El Abbadi, "A Review for Sign Language Recognition Techniques," 2021 1st Babylon International Conference on Information Technology and Science (BICITS), Babil, Iraq, 2021, pp. 39-44, doi: 10.1109/BICITS51482.2021.9509905.

[3] M. A. Almasre and H. Al-Nuaim, "A comparison of Arabic sign language dynamic gesture recognition models," Heliyon, vol. 6, no. 3, p. e03554, 2020, doi: 10.1016/j.heliyon.2020.e03554.

[4] Chouhan, Tushar, et al.Smart glove with gesture recognition ability for the hearing and speech impaired. 2014 IEEE Global Humanitarian Technology Conference-South Asia Satellite (GHTC-SAS). 2014.

[5] Kumar, K. (2022). DEAF-BSL: Deep Learning Framework for British Sign Language recognition. ACM Transactions on Asian and Low-resource Language Information Processing, 21(5), 1–14. https://doi.org/10.1145/3513004.

[6] Tyagi, A., & Bansal, S. (2020). Feature extraction technique for Vision-Based Indian Sign Language Recognition System: a review. Advances in Intelligent Systems and Computing, 39–53. https://doi.org/10.1007/978-981-15-6876-3_4.

[7] Sarma D, Bhuyan MK. Methods, Databases and Recent Advancement of Vision-Based Hand Gesture Recognition for HCI Systems: A Review. SN Comput Sci. 2021;2(6):436. doi: 10.1007/s42979-021-00827-x. Epub 2021 Aug 29. PMID: 34485925; PMCID: PMC8403257.

[8] Aloysius, N., & Geetha, M. K. (2020). Understanding vision-based continuous sign language recognition. *Multimedia Tools and Applications*, *79*(31–32), 22177–22209. https://doi.org/10.1007/s11042-020-08961-z .

[9] Gupta, H., Ramjiwal, A., & Jose, J. T. (2018). Vision-based approach to sign language recognition. International Journal of Advances in Applied Sciences, 7(2), 156. https://doi.org/10.11591/ijaas.v7.i2.pp156-161.

[10] Ahmed, Mohamed Aktham & Bahaa, Bilal & Zaidan, A. & Salih, Mahmood & Lakulu, Muhammad Modi. (2018). A Review on Systems-Based Sensory Gloves for Sign Language Recognition State of the Art between 2007 and 2017. Sensors. 18. 2208. 10.3390/s18072208.

[11] F, A. T. L. &. O. V. &. N. E. (2023). Sign Language Recognition System using Flex Sensor Network. ideas.repec.org. https://ideas.repec.org/a/bjf/journl/v8y2023i9p182-187.html.

[12] Korhan Mutludogan and M Bilgin Metin, "American sign-language character recognition with capsulenet works", International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT), pp. 1-6, 2019.

[13] Ijraset. (n.d.-b). Sign language to text conversion using flex sensors. IJRASET. https://www.ijraset.com/research-paper/sign-language-to-text-conversion-using-flex-sensors.

[14] Ijraset. (n.d.). Sign language to text conversion using flex sensors. IJRASET. https://www.ijraset.com/research-paper/sign-language-to-text-conversion-using-flex-sensors.

[15] Shareef, S. R., Hussain, M. M., Gupta, A., & Aslam, H. A. (2020). Hand gesture recognition system for deaf and dumb. https://www.semanticscholar.org/paper/Hand-Gesture-Recognition-System-for-Deaf-and-Dumb-Shareef-Hussain/a7476a619595dd768715542d7dabc5e9ccb23858.

[16] Ambar, R., Fai, C. K., Wahab, M. H. A., Jamil, M. M. A., & Ma'radzi, A. A. (2018). Development of a wearable device for sign language recognition. Journal of Physics. Conference Series, 1019, 012017. https://doi.org/10.1088/1742-6596/1019/1/012017.

[17] Amin MS, Rizvi STH, Hossain MM. A Comparative Review on Applications of Different Sensors for Sign Language Recognition. J Imaging. 2022 Apr 2;8(4):98. doi: 10.3390/jimaging8040098. PMID: 35448225; PMCID: PMC9027924.

[18] A. B. Jani, N. A. Kotak and A. K. Roy, "Sensor Based Hand Gesture Recognition System for English Alphabets Used in Sign Language of Deaf-Mute People," 2018 IEEE SENSORS, New Delhi, India, 2018, pp. 1-4, doi: 10.1109/ICSENS.2018.8589574. keywords: {Gesture recognition;Assistive technology;Thumb;Flexible printed circuits;Accelerometers;Gyroscopes;Hand gesture recognition;Sign language recognition;Human computer interface;Accelerometer;Sensors;Sensing based device;Flex Sensor;Gyroscope},

[19] H. S. Anupama, B. A. Usha, S. Madhushankar, V. Vivek and Y. Kulkarni, "Automated Sign Language Interpreter Using Data Gloves," 2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS), Coimbatore, India, 2021, pp. 472-476, doi: 10.1109/ICAIS50930.2021.9395749.

[20] S. B. Rizwan, M. S. Z. Khan and M. Imran, "American Sign Language Translation via Smart Wearable Glove Technology," 2019 International Symposium on Recent Advances in Electrical Engineering (RAEE), Islamabad, Pakistan, 2019, pp. 1-6, doi: 10.1109/RAEE.2019.8886931.

[21] S. Sa, M. R. Chowdary, M. Satvika, K. Kalidindi, S. Bj and P. Kokila, "Gesture Recognition Glove For American Sign Language Using Accelerometers," 2023 International Conference on Advancement in Computation & Computer Technologies (InCACCT), Gharuan, India, 2023, pp. 784-789, doi: 10.1109/InCACCT57535.2023.10141835.

[22] A. Suri, S. K. Singh, R. Sharma, P. Sharma, N. Garg and R. Upadhyaya, "Development of Sign Language using Flex Sensors," 2020 International Conference on Smart Electronics and Communication (ICOSEC), Trichy, India, 2020, pp. 102-106, doi: 10.1109/ICOSEC49089.2020.9215392.

[23] A. B. Jani, N. A. Kotak and A. K. Roy, "Sensor-Based Hand Gesture Recognition System for English Alphabet Used in Sign Language of Deaf-Mute people," 2018 IEEE SENSORS, New Delhi, India, 2018, pp. 1-4, doi: 10.1109/ICSENS.2018.85.

APPENDICES

APPENDICES A

ALPHABET OF USER HAND GESTURE	OUTPUT AN ON-OLED DISPLAY	ADC VALUE				
		THUMB	INDEX	MIDDLE	RING	LITTLE
A	A	410	280	260	290	214
B	B	292	471	486	493	439
C	-	360	378	356	414	417
D	D	346	471	295	316	269
E	E	292	299	279	307	247
F	F	327	312	407	401	437
G	G	425	437	271	300	233
H	H	325	478	471	327	270
I	I	322	279	295	338	437
J	-	370	316	329	365	442
K	K	333	471	465	327	282
L	D	439	479	273	308	237
M	S	310	292	294	303	258
N	O	396	321	317	368	285
O	O	371	336	329	380	365
P	-	428	464	476	411	345
Q	-	430	446	295	320	267
R	R	347	476	476	348	281
S	S	329	275	268	301	244

T	O	309	325	329	347	263
U	-	350	478	474	326	317
V	-	361	477	475	368	323
W	W	357	479	477	489	288
X	-	342	327	294	329	271
Y	Y	419	285	314	341	439
Z	Z	395	474	334	322	296

APPENDICES B

ALPHABET OF USER HAND GESTURE	OUTPUT AN ON-OLED DISPLAY	ADC VALUE				
		THUMB	INDEX	MIDDLE	RING	LITTLE
A	A	402	256	249	288	213
B	B	305	475	481	494	447
C	-	409	371	392	420	391
D	D	323	453	308	329	304
E	E	257	268	293	334	266
F	F	318	331	465	485	446
G	G	427	407	261	303	223
H	U	354	425	484	338	308
I	I	337	295	269	339	446
J	-	360	306	335	363	440
K	-	324	413	473	347	284
L	-	424	441	292	319	227
M	O	342	308	312	350	349
N	O	387	335	345	364	253
O	O	314	304	321	357	311
P	-	431	442	421	375	302
Q	-	429	424	268	305	217
R	-	302	381	419	314	271
S	S	317	283	251	316	258
T	-	365	331	286	345	260
U	U	354	425	484	338	308
V	-	310	395	478	334	345
W	W	315	403	473	493	282
X	-	313	318	254	304	253
Y	Y	417	344	306	337	439
Z	-	330	377	293	330	234

APPENDICES C

ALPHABET OF USER HAND GESTURE	OUTPUT AN ON-OLED DISPLAY	ADC VALUE				
		THUMB	INDEX	MIDDLE	RING	LITTLE
A	A	430	253	272	283	224
B	B	316	489	511	501	446
C	C	369	386	379	420	427
D	D	343	469	305	336	276
E	E	289	261	277	308	249
F	F	286	290	476	494	440
G	G	432	418	254	268	199
H	H	322	476	479	304	268
I	I	288	246	280	336	423
J	-	367	306	333	367	446
K	K	331	481	502	342	277
L	D	431	468	281	293	248
M	M	296	264	316	306	228
N	N	364	270	282	291	221
O	O	343	344	337	380	387
P	P	397	480	482	431	202
Q	G	456	443	299	311	247
R	R	355	477	501	341	295
S	S	331	279	330	370	268
T	T	399	274	305	313	267
U	U	356	481	512	354	309
V	V	350	483	509	383	333
W	W	336	491	507	501	298
X	X	344	444	289	307	261
Y	Y	432	333	372	394	448
Z	Z	376	488	321	363	296

APPENDICES D

ALPHABET OF USER HAND GESTURE	OUTPUT AN ON-OLED DISPLAY	ADC VALUE				
		THUMB	INDEX	MIDDLE	RING	LITTLE
A	A	426	250	266	286	214
B	B	308	478	485	483	433
C	O	376	347	358	396	388
D	D	350	478	261	305	268
E	E	275	260	275	314	270
F	-	394	269	485	480	433
G	Q	435	407	259	299	240
H	H	301	476	482	316	287
I	I	290	265	292	331	439
J	-	357	300	333	364	444
K	K	296	489	488	325	271
L	D	456	480	276	324	279
M	S	334	288	302	329	284
N	N	339	296	289	331	281
O	O	325	325	323	364	317
P	W	402	474	453	444	364
Q	Q	415	396	268	317	335
R	H	336	466	454	323	291
S	S	330	288	300	330	281
T	N	335	301	290	329	281
U	U	354	425	484	338	308
V	H	352	484	487	363	300
W	W	324	485	489	461	285
X	-	300	316	246	316	272
Y	Y	432	267	283	317	423
Z	Z	370	450	284	388	240

APPENDICES E

ALPHABET OF USER HAND GESTURE	OUTPUT AN ON-OLED DISPLAY	ADC VALUE				
		THUMB	INDEX	MIDDLE	RING	LITTLE
A	A	438	253	270	280	224
B	B	310	485	480	485	440
C	-	369	386	379	420	427
D	D	343	469	305	336	276
E	E	289	261	277	308	249
F	F	286	290	476	494	440
G	Q	432	418	254	268	199
H	H	322	476	479	304	268
I	I	288	246	280	336	423
J	-	365	310	335	364	439
K	K	331	481	502	342	277
L	D	431	468	281	293	248
M	M	296	264	316	306	228
N	S	364	270	282	291	221
O	O	343	344	337	380	387
P	-	397	480	482	431	202
Q	Q	456	443	299	311	247
R	H	355	477	501	341	295
S	S	331	279	330	370	268
T	S	399	274	305	313	267
U	U	356	481	512	354	309
V	H	350	483	509	383	333
W	W	336	491	507	501	298
X	-	344	444	289	307	261
Y	Y	432	333	372	394	448
Z	Z	376	468	311	373	256

APPENDICES F

ALPHABET OF USER HAND GESTURE	OUTPUT AN ON-OLED DISPLAY	ADC VALUE				
		THUMB	INDEX	MIDDLE	RING	LITTLE
A	A	319	303	252	258	220
B	B	324	464	478	481	447
C	C	358	333	343	370	374
D	D	327	412	294	294	238
E	-	337	266	474	470	446
F	F	331	285	478	478	435
G	G	415	393	244	255	216
H	-	287	231	265	252	444
I	I	326	291	302	297	444
J	J	323	313	316	337	432
K	-	433	462	279	295	288
L	L	446	461	262	283	237
M	-	331	246	289	311	226
N	N	354	287	267	268	236
O	O	329	310	317	314	289
P	-	396	401	251	280	273
Q	Q	421	425	258	278	205
R	-	292	232	245	286	275
S	S	294	287	277	278	206
T	T	377	284	245	273	204
U	U	331	460	479	293	305
V	V	323	453	476	327	292
W	W	340	461	479	463	284
X	X	279	303	252	258	210
Y	Y	444	351	293	330	433
Z	Z	325	413	296	286	222

APPENDICES G

ALPHABET OF USER HAND GESTURE	OUTPUT AN ON-OLED DISPLAY	ADC VALUE				
		THUMB	INDEX	MIDDLE	RING	LITTLE
A	A	323	290	251	255	217
B	B	305	473	482	487	455
C	C	370	356	345	358	378
D	D	321	450	282	289	226
E	E	317	266	270	320	240
F	F	314	304	275	479	452
G	G	425	409	232	270	200
H	H	361	414	482	256	254
I	I	326	292	294	299	443
J	J	329	312	310	352	446
K	K	311	474	482	296	234
L	L	442	457	254	290	218
M	M	327	270	283	291	228
N	N	360	266	268	244	226
O	O	321	307	323	310	291
P	P	361	442	363	315	228
Q	-	441	407	291	312	253
R	R	336	452	468	236	244
S	S	296	280	284	280	216
T	-	333	288	288	264	255
U	U	320	426	484	284	293
V	-	302	428	486	292	282
W	W	278	415	486	475	272
X	-	291	258	262	256	224
Y	Y	447	371	297	339	445
Z	Z	320	392	280	289	224

APPENDICES H

ALPHABET OF USER HAND GESTURE	OUTPUT AN ON-OLED DISPLAY	ADC VALUE				
		THUMB	INDEX	MIDDLE	RING	LITTLE
A	A	326	290	262	271	232
B	B	321	482	483	483	438
C	-	416	353	308	249	366
D	-	350	410	373	333	323
E	E	294	279	271	311	248
F	F	349	287	488	484	440
G	G	416	400	240	265	201
H	H	369	415	486	255	255
I	-	314	286	270	254	431
J	-	316	282	264	305	401
K	K	364	481	464	287	237
L	L	458	455	257	273	221
M	M	329	281	254	305	217
N	N	361	292	256	283	216
O	O	336	310	312	311	297
P	P	376	449	369	317	226
Q	Q	432	436	263	295	216
R	R	336	472	470	239	249
S	S	285	283	258	268	214
T	T	376	290	252	271	212
U	-	275	368	455	323	275
V	-	312	391	465	337	275
W	W	339	401	476	481	263
X	X	285	303	258	268	214
Y	-	408	348	277	310	422
Z	-	335	400	354	262	251

APPENDICES I

ALPHABET OF USER HAND GESTURE	OUTPUT AN ON-OLED DISPLAY	ADC VALUE				
		THUMB	INDEX	MIDDLE	RING	LITTLE
A	A	324	296	253	260	216
B	B	224	489	488	475	433
C	-	360	341	328	344	354
D	D	348	411	297	291	236
E	E	308	283	280	318	284
F	-	395	275	459	471	436
G	G	424	407	230	272	201
H	H	385	422	483	253	265
I	I	296	271	281	285	432
J	J	327	300	316	342	437
K	-	282	352	431	312	234
L	L	448	460	246	294	229
M	M	344	296	273	290	226
N	N	365	255	285	241	214
O	O	328	293	310	318	284
P	-	335	359	410	404	308
Q	-	401	383	328	299	310
R	-	293	338	388	309	304
S	S	301	272	255	297	220
T	T	375	292	252	281	213
U	-	336	382	406	353	300
V	-	323	371	426	350	370
W	W	337	392	483	484	286
X	-	309	298	279	321	255
Y	Y	445	364	301	342	434
Z	-	332	377	372	389	306

APPENDICES J

ALPHABET OF USER HAND GESTURE	OUTPUT AN ON-OLED DISPLAY	ADC VALUE				
		THUMB	INDEX	MIDDLE	RING	LITTLE
A	A	316	282	252	251	216
B	B	315	465	489	487	446
C	C	352	342	342	357	390
D	-	378	398	382	383	348
E	E	296	282	288	321	287
F	-	390	337	405	440	447
G	-	431	413	316	253	212
H	H	368	417	485	252	261
I	I	319	302	303	314	306
J	J	310	325	316	342	446
K	-	303	362	431	293	282
L	L	456	456	260	270	233
M	M	326	262	282	291	226
N	N	374	257	280	248	223
O	O	329	302	323	314	306
P	-	328	378	437	311	272
Q	Q	434	441	260	295	218
R	R	326	466	472	238	230
S	S	313	264	241	266	207
T	T	380	288	248	262	197
U	U	321	432	480	306	307
V	V	311	421	490	334	280
W	W	334	414	485	474	275
X	-	357	346	312	318	250
Y	Y	459	373	291	330	435
Z	Z	323	378	283	281	213

APPENDICES K

ALPHABET OF USER HAND GESTURE	OUTPUT AN ON-OLED DISPLAY	ADC VALUE				
		THUMB	INDEX	MIDDLE	RING	LITTLE
A	A	305	298	246	291	220
B	B	331	391	465	473	433
C	-	350	327	286	323	323
D	D	337	413	297	297	235
E	E	312	267	287	297	235
F	F	292	288	457	475	448
G	G	397	406	245	276	213
H	-	300	364	482	296	232
I	I	330	292	294	292	441
J	-	321	288	276	295	438
K	-	288	371	459	293	226
L	L	437	453	258	265	214
M	M	338	273	286	290	230
N	N	366	259	265	256	210
O	O	326	318	297	305	302
P	P	371	443	363	305	236
Q	Q	428	427	235	278	234
R	R	332	478	474	240	241
S	S	281	272	277	278	210
T	-	294	276	233	264	227
U	U	318	425	472	295	286
V	V	318	428	482	327	288
W	W	336	391	486	484	272
X	X	285	298	256	271	210
Y	Y	446	368	295	347	443
Z	Z	323	384	271	278	218

APPENDICES L

ALPHABET OF USER HAND GESTURE	OUTPUT AN ON-OLED DISPLAY	ADC VALUE				
		THUMB	INDEX	MIDDLE	RING	LITTLE
A	A	328	307	248	253	210
B	B	320	390	473	465	443
C	C	389	347	353	346	369
D	D	333	411	291	295	235
E	-	278	247	237	235	217
F	F	355	321	478	418	423
G	G	432	416	248	284	209
H	H	376	411	488	269	262
I	I	329	290	292	286	429
J	-	317	289	269	270	403
K	K	327	392	472	225	231
L	L	446	446	263	275	233
M	M	329	291	271	274	262
N	-	319	285	264	223	199
O	O	320	303	322	296	291
P	P	369	442	365	306	234
Q	Q	438	437	255	291	211
R	R	323	463	459	236	249
S	S	297	265	248	272	220
T	-	313	316	242	227	217
U	U	296	411	468	279	272
V	-	317	371	467	182	238
W	W	328	417	470	480	269
X	X	292	293	257	279	244
Y	Y	430	352	299	330	433
Z	Z	329	381	285	273	222

APPENDICES M

ALPHABET OF USER HAND GESTURE	OUTPUT AN ON-OLED DISPLAY	ADC VALUE				
		THUMB	INDEX	MIDDLE	RING	LITTLE
A	A	334	287	270	280	244
B	B	328	471	478	497	440
C	C	392	361	347	367	370
D	D	346	407	297	297	247
E	-	263	245	235	235	205
F	F	350	280	482	488	444
G	G	417	396	279	264	224
H	-	282	331	461	255	266
I	I	322	321	307	277	350
J	J	324	319	312	331	448
K	-	267	354	444	267	231
L	L	434	429	259	270	221
M	M	316	278	301	281	233
N	N	375	260	262	253	221
O	O	329	292	299	312	293
P	-	332	387	463	360	305
Q	Q	417	436	269	294	224
R	-	289	350	421	226	228
S	S	306	288	301	251	233
T	T	379	291	247	278	201
U	U	311	425	466	296	292
V	-	301	366	480	271	231
W	W	338	394	480	482	277
X	-	274	267	235	255	220
Y	Y	445	373	291	335	446
Z	Z	323	389	279	273	226

APPENDICES N

ALPHABET OF USER HAND GESTURE	OUTPUT AN ON-OLED DISPLAY	ADC VALUE				
		THUMB	INDEX	MIDDLE	RING	LITTLE
A	A	326	290	266	256	214
B	B	318	478	485	483	433
C	C	376	347	358	366	388
D	D	340	378	291	305	238
E	E	295	260	275	314	250
F	F	354	289	485	480	433
G	G	425	407	259	259	210
H	H	371	426	482	266	257
I	-	290	265	292	331	439
J	-	357	300	333	364	444
K	K	356	489	488	285	241
L	L	456	460	266	284	229
M	-	334	288	302	329	284
N	-	339	296	289	331	281
O	O	325	315	323	314	317
P	-	402	474	453	444	364
Q	Q	425	446	268	317	215
R	-	336	466	454	323	291
S	-	330	288	300	330	281
T	-	335	301	290	329	281
U	U	354	425	484	338	308
V	V	352	424	487	363	300
W	W	324	420	489	471	285
X	X	382	312	246	274	214
Y	Y	432	367	293	327	433
Z	Z	315	380	284	282	222

APPENDICES O

ALPHABET OF USER HAND GESTURE	OUTPUT AN ON-OLED DISPLAY	ADC VALUE				
		THUMB	INDEX	MIDDLE	RING	LITTLE
A	A	316	295	266	286	214
B	B	322	478	485	483	433
C	C	376	347	358	362	388
D	D	340	418	315	283	248
E	E	298	278	275	314	250
F	F	348	289	485	480	433
G	G	421	395	248	259	216
H	H	361	412	485	255	256
I	I	325	295	307	282	440
J	J	322	318	319	345	442
K	-	392	467	575	336	264
L	L	445	459	255	280	226
M	M	330	266	286	281	221
N	N	357	301	276	288	208
O	O	325	318	311	305	295
P	P	370	440	365	315	229
Q	Q	425	437	258	292	215
R	R	324	466	469	236	245
S	S	298	275	275	280	212
T	T	383	309	265	281	199
U	-	343	465	479	334	315
V	-	342	456	483	335	290
W	W	339	422	480	482	274
X	X	282	325	255	305	222
Y	Y	445	365	314	335	442
Z	Z	332	389	275	265	226

APPENDICES P

```
#include <Wire.h>

#include <Adafruit_GFX.h>

#include <Adafruit_SSD1306.h>

#include <Adafruit_ADXL345_U.h> // Include the accelerometer library

#include <SoftwareSerial.h>



// Define constants

#define SCREEN_WIDTH 128

#define SCREEN_HEIGHT 64

#define OLED_RESET    -1

#define GREEN_LED_PIN 8

#define RED_LED_PIN 9

#define BUZZER_PIN 13

#define VCC 5.0

#define fixed_resistor 10000.0
```

```
// Initialize objects

Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire,
OLED_RESET);

Adafruit_ADXL345_Unified accel = Adafruit_ADXL345_Unified(12345);



// Define ADC pins

const int ADC_pins[] = {A0, A1, A2, A3, A4};

const int num_ADC_pins = 5;



// Define gesture recognition variables

int a = 0;

int b = 0;

int c = 0;
```

```
int d = 0;

int e = 0;



void setup() {

  pinMode(A0, INPUT); //thumb

  pinMode(A1, INPUT); //index

  pinMode(A2, INPUT); //middle

  pinMode(A3, INPUT); //ring

  pinMode(A4, INPUT); //little



  pinMode(GREEN_LED_PIN, OUTPUT);

  pinMode(RED_LED_PIN, OUTPUT);

  pinMode(BUZZER_PIN, OUTPUT);



  digitalWrite(GREEN_LED_PIN, LOW);

  digitalWrite(RED_LED_PIN, LOW);

  digitalWrite(BUZZER_PIN, LOW);
```

```
Serial.begin(9600);



if(!accel.begin()) {
  Serial.println("Could not find a valid ADXL345 sensor, check wiring!");
  while(1);
}



if(!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
  Serial.println(F("SSD1306 allocation failed"));
  for(;;);
}
display.clearDisplay();
display.setTextSize(1.5);
display.setTextColor(SSD1306_WHITE);
display.setCursor(50, 0);
display.println("HELLO");
```

```
display.display();

delay(2500);

display.clearDisplay();

display.setTextSize(1.5);

display.setTextColor(SSD1306_WHITE);

display.setCursor(0, 0);

display.println("PLEASE WEAR THE GLOVE");

display.display();

delay(2500);

display.clearDisplay();

display.setTextSize(1.5);

display.setTextColor(SSD1306_WHITE);

display.setCursor(0, 0);

display.println("YOU CAN START MAKE   HAND GESTURE");

display.display();

delay(2500); // Pause for 2.5 seconds

display.clearDisplay();
```

```
}

void loop() {

  display.clearDisplay(); // Clear the display buffer

  // Read analog inputs and perform gesture recognition
  a = analogRead(A0); //thumb
  b = analogRead(A1); //index
  c = analogRead(A2); //middle
  d = analogRead(A3); //ring
  e = analogRead(A4); //little

  sensors_event_t event;
  accel.getEvent(&event); // Read accelerometer data
```

```
digitalWrite(GREEN_LED_PIN, LOW);

digitalWrite(RED_LED_PIN, LOW);

digitalWrite(BUZZER_PIN, LOW);



// Gesture recognition logic



if (a < 400 && b < 350 && c < 270 && d < 270 && e < 220 && a > 350 && b >
300 && c > 200 && d > 200 && e > 150) {

  display.setTextSize(2.0);

  display.setTextColor(SSD1306_WHITE);

  display.setCursor(50, 0);

  display.println("A");

  display.display(); // Update the OLED display with the new content

  digitalWrite(GREEN_LED_PIN, HIGH); // Turn on green LED for true condition

  digitalWrite(BUZZER_PIN, HIGH); // Turn on buzzer for true condition

  delay(3000); // Delay to display "A"

  display.clearDisplay();
```

```
}

else if (a < 300 && b > 200 && c > 400 && d > 400 && e > 400 && a > 200 &&
b < 400) {
  display.setTextSize(2.0);
  display.setTextColor(SSD1306_WHITE);
  display.setCursor(50, 0);
  display.println("B");
  display.display(); // Update the OLED display with the new content
  digitalWrite(GREEN_LED_PIN, HIGH);
  digitalWrite(BUZZER_PIN, HIGH); // Turn on buzzer for true condition
  delay(3000); // Delay to display "B"
  display.clearDisplay();
}

else if (a < 360 && b < 360 && c < 360 && d < 360 && e < 390 && a > 300 &&
b > 300 && c > 300 && d > 320 && e > 320) {
  display.setTextSize(2.0);
```

```
    display.setTextColor(SSD1306_WHITE);

    display.setCursor(50, 0);

    display.println("C");

    display.display(); // Update the OLED display with the new content

    digitalWrite(GREEN_LED_PIN, HIGH);

    digitalWrite(BUZZER_PIN, HIGH); // Turn on buzzer for true condition

    delay(3000); // Delay to display "C"

    display.clearDisplay();

  }

  else if (a > 250 && b > 300 && c < 360 && d < 350 && e > 150 && a < 350 &&
b < 400 && c > 260 && d > 250 && e < 280) {

    display.setTextSize(2.0);

    display.setTextColor(SSD1306_WHITE);

    display.setCursor(50, 0);

    display.println("D");

    display.display(); // Update the OLED display with the new content
```

```
    digitalWrite(GREEN_LED_PIN, HIGH);
    digitalWrite(BUZZER_PIN, HIGH); // Turn on buzzer for true condition
    delay(3000); // Delay to display "D"
    display.clearDisplay();
  }

  else if (a < 290 && b > 240 && c < 290 && d < 310 && e > 220 && a > 250 &&
b < 280 && c > 240 && d > 270 && e < 260) {
    display.setTextSize(2.0);
    display.setTextColor(SSD1306_WHITE);
    display.setCursor(50, 0);
    display.println("E");
    display.display(); // Update the OLED display with the new content
    digitalWrite(GREEN_LED_PIN, HIGH);
    digitalWrite(BUZZER_PIN, HIGH); // Turn on buzzer for true condition
    delay(3000); // Delay to display "E"
    display.clearDisplay();
```

```
  }

  else if (a > 230 && b < 300 && c < 500 && d > 410 && e > 400 && a < 300 &&
b > 250 && c > 450 && e < 460) {
    display.setTextSize(2.0);
    display.setTextColor(SSD1306_WHITE);
    display.setCursor(50, 0);
    display.println("F");
    display.display(); // Update the OLED display with the new content
    digitalWrite(GREEN_LED_PIN, HIGH);
    digitalWrite(BUZZER_PIN, HIGH); // Turn on buzzer for true condition
    delay(3000); // Delay to display "F"
    display.clearDisplay();
  }

  else if (a > 350 && b > 350 && c < 320 && d < 280 && e < 200 && a < 400 &&
b < 400 && c > 220 && d > 240 && e > 150) {
    display.setTextSize(2.0);
```

```
display.setTextColor(SSD1306_WHITE);

display.setCursor(50, 0);

display.println("G");

display.display(); // Update the OLED display with the new content

digitalWrite(GREEN_LED_PIN, HIGH);

digitalWrite(BUZZER_PIN, HIGH); // Turn on buzzer for true condition

delay(3000); // Delay to display "G"

display.clearDisplay();

}

else if (a < 310 && b > 340 && c > 400 && d > 260 && e < 250 && a > 260 &&
b < 400 && c < 450 && d < 300 && e > 200) {

display.setTextSize(2.0);

display.setTextColor(SSD1306_WHITE);

display.setCursor(50, 0);

display.println("H");

display.display(); // Update the OLED display with the new content
```

```
    digitalWrite(GREEN_LED_PIN, HIGH);

    digitalWrite(BUZZER_PIN, HIGH); // Turn on buzzer for true condition

    delay(3000); // Delay to display "H"

    display.clearDisplay();

  }



  else if (a < 320 && b > 270 && c < 300 && d < 320 && e > 410 && a > 270 &&
b < 310 && c > 250 && d > 280 && e < 450) {

    display.setTextSize(2.0);

    display.setTextColor(SSD1306_WHITE);

    display.setCursor(50, 0);

    display.println("I");

    display.display(); // Update the OLED display with the new content

    digitalWrite(GREEN_LED_PIN, HIGH);

    digitalWrite(BUZZER_PIN, HIGH); // Turn on buzzer for true condition

    delay(3000); // Delay to display "I"

    display.clearDisplay();
```

```
}

else if (a < 320 && b > 250 && c < 350 && d < 350 && e > 400 && a > 230 &&
b < 330 && c > 250 && d > 280 && e < 460 == event.acceleration.x > 1.0 &&
event.acceleration.y < 1.0 &&  event.acceleration.z > 1.0) {
  display.setTextSize(2.0);
  display.setTextColor(SSD1306_WHITE);
  display.setCursor(50, 0);
  display.println("J");
  display.display(); // Update the OLED display with the new content
  digitalWrite(GREEN_LED_PIN, HIGH);
  digitalWrite(BUZZER_PIN, HIGH); // Turn on buzzer for true condition
  delay(3000); // Delay to display "J"
  display.clearDisplay();
}

else if (a > 300 && b > 340 && c > 400 && d > 280 && e < 280 && a < 350 &&
b < 400 && c < 500 && d < 330 && e > 200) {
```

```
    display.setTextSize(2.0);

    display.setTextColor(SSD1306_WHITE);

    display.setCursor(50, 0);

    display.println("K");

    display.display(); // Update the OLED display with the new content

    digitalWrite(GREEN_LED_PIN, HIGH);

    digitalWrite(BUZZER_PIN, HIGH); // Turn on buzzer for true condition

    delay(3000); // Delay to display "K"

    display.clearDisplay();

  }
```



```
  else if (a < 460 && b > 410 && c < 360 && d < 280 && e < 230 && a > 420 &&
b < 450 && c > 320 && d > 240 && e > 170) {

    display.setTextSize(2.0);

    display.setTextColor(SSD1306_WHITE);

    display.setCursor(50, 0);

    display.println("L");
```

```
    display.display(); // Update the OLED display with the new content

    digitalWrite(GREEN_LED_PIN, HIGH);

    digitalWrite(BUZZER_PIN, HIGH); // Turn on buzzer for true condition

    delay(3000); // Delay to display "L"

    display.clearDisplay();

  }



  else if (a < 330 && b < 300 && c < 300 && d < 300 && e < 250 && a > 290 &&
b > 250 && c > 250 && d > 260 && e > 180) {

    display.setTextSize(2.0);

    display.setTextColor(SSD1306_WHITE);

    display.setCursor(50, 0);

    display.println("M");

    display.display(); // Update the OLED display with the new content

    digitalWrite(GREEN_LED_PIN, HIGH);

    digitalWrite(BUZZER_PIN, HIGH); // Turn on buzzer for true condition

    delay(3000); // Delay to display "M"
```

```
    display.clearDisplay();
  }

  else if (a < 380 && b < 330 && c < 300 && d > 250 && e < 250 && a > 320 &&
b > 270 && c > 250 && d < 300 && e > 180) {
    display.setTextSize(2.0);
    display.setTextColor(SSD1306_WHITE);
    display.setCursor(50, 0);
    display.println("N");
    display.display(); // Update the OLED display with the new content
    digitalWrite(GREEN_LED_PIN, HIGH);
    digitalWrite(BUZZER_PIN, HIGH); // Turn on buzzer for true condition
    delay(3000); // Delay to display "N"
    display.clearDisplay();
  }

  else if (a < 350 && b < 330 && c < 330 && d < 340 && e < 300 && a > 300 &&
b > 280 && c > 280 && d > 290 && e > 250) {
```

```
    display.setTextSize(2.0);
    display.setTextColor(SSD1306_WHITE);
    display.setCursor(50, 0);
    display.println("O");
    display.display(); // Update the OLED display with the new content
    digitalWrite(GREEN_LED_PIN, HIGH);
    digitalWrite(BUZZER_PIN, HIGH); // Turn on buzzer for true condition
    delay(3000); // Delay to display "O"
    display.clearDisplay();
  }
```



```
  else if (a > 320 && b > 350 && c > 350 && d > 280 && e > 200 && a < 360 &&
b < 400 && c < 400 && d < 330 && e < 250) {
    display.setTextSize(2.0);
    display.setTextColor(SSD1306_WHITE);
    display.setCursor(50, 0);
    display.println("P");
```

```
  display.display(); // Update the OLED display with the new content

  digitalWrite(GREEN_LED_PIN, HIGH);

  digitalWrite(BUZZER_PIN, HIGH); // Turn on buzzer for true condition

  delay(3000); // Delay to display "P"

  display.clearDisplay();

 }


 else if (a > 350 && b > 340 && c < 350 && d < 300 && e < 210 && a < 430 &&
b < 430 && c > 220 && d > 250 && e > 150) {

  display.setTextSize(2.0);

  display.setTextColor(SSD1306_WHITE);

  display.setCursor(50, 0);

  display.println("Q");

  display.display(); // Update the OLED display with the new content

  digitalWrite(GREEN_LED_PIN, HIGH);

  digitalWrite(BUZZER_PIN, HIGH); // Turn on buzzer for true condition

  delay(3000); // Delay to display "Q"
```

```
  display.clearDisplay();

 }


 else if (a < 310 && b > 320 && c > 450 && d < 330 && e < 260 && a > 250 &&
b < 400 && d > 280 && e > 220) {

  display.setTextSize(2.0);

  display.setTextColor(SSD1306_WHITE);

  display.setCursor(50, 0);

  display.println("R");

  display.display(); // Update the OLED display with the new content

  digitalWrite(GREEN_LED_PIN, HIGH);

  digitalWrite(BUZZER_PIN, HIGH); // Turn on buzzer for true condition

  delay(3000); // Delay to display "R"

  display.clearDisplay();

 }


 else if (a < 310 && b < 280 && c < 260 && d < 300 && e < 210 && a > 270 &&
b > 240 && c > 220 && d > 250 && e > 160) {
```

```
    display.setTextSize(2.0);

    display.setTextColor(SSD1306_WHITE);

    display.setCursor(50, 0);

    display.println("S");

    display.display(); // Update the OLED display with the new content

    digitalWrite(GREEN_LED_PIN, HIGH);

    digitalWrite(BUZZER_PIN, HIGH); // Turn on buzzer for true condition

    delay(3000); // Delay to display "S"

    display.clearDisplay();

  }

  else if (a < 400 && b < 350 && c < 300 && d < 280 && e < 230 && a > 350 &&
b > 300 && c > 250 && d > 230 && e > 160) {

    display.setTextSize(2.0);

    display.setTextColor(SSD1306_WHITE);

    display.setCursor(50, 0);

    display.println("T");
```

```
  display.display(); // Update the OLED display with the new content

  digitalWrite(GREEN_LED_PIN, HIGH);

  digitalWrite(BUZZER_PIN, HIGH); // Turn on buzzer for true condition

  delay(3000); // Delay to display "T"

  display.clearDisplay();

 }

 else if (a < 340 && b < 400 && c > 400 && d < 330 && e < 280 && a > 280 &&
b > 340 && c < 500 && d > 260 && e > 200) {

  display.setTextSize(2.0);

  display.setTextColor(SSD1306_WHITE);

  display.setCursor(50, 0);

  display.println("U");

  display.display(); // Update the OLED display with the new content

  digitalWrite(GREEN_LED_PIN, HIGH);

  digitalWrite(BUZZER_PIN, HIGH); // Turn on buzzer for true condition

  delay(3000); // Delay to display "U"
```

```
    display.clearDisplay();
  }

  else if (a < 330 && b > 350 && c > 400 && d < 350 && e < 300 && a > 280 &&
b < 400 && c < 500 && d > 280 && e > 220) {
    display.setTextSize(2.0);
    display.setTextColor(SSD1306_WHITE);
    display.setCursor(50, 0);
    display.println("V");
    display.display(); // Update the OLED display with the new content
    digitalWrite(GREEN_LED_PIN, HIGH);
    digitalWrite(BUZZER_PIN, HIGH); // Turn on buzzer for true condition
    delay(3000); // Delay to display "V"
    display.clearDisplay();
  }

  else if (a < 330 && b > 350 && c > 400 && d > 450 && e < 270 && a > 270 &&
b < 400 && c < 450 && d < 500 && e > 220) {
```

```
display.setTextSize(2.0);

display.setTextColor(SSD1306_WHITE);

display.setCursor(50, 0);

display.println("W");

display.display(); // Update the OLED display with the new content

digitalWrite(GREEN_LED_PIN, HIGH);

digitalWrite(BUZZER_PIN, HIGH); // Turn on buzzer for true condition

delay(3000); // Delay to display "W"

display.clearDisplay();

}
```



```
else if (a < 330 && b < 310 && c < 280 && d < 300 && e < 220 && a > 270 &&
b > 260 && c > 240 && d > 250 && e > 170) {

display.setTextSize(2.0);

display.setTextColor(SSD1306_WHITE);

display.setCursor(50, 0);

display.println("X");
```

```
display.display(); // Update the OLED display with the new content

digitalWrite(GREEN_LED_PIN, HIGH);

digitalWrite(BUZZER_PIN, HIGH); // Turn on buzzer for true condition

delay(3000); // Delay to display "X"

display.clearDisplay();

}

else if (a > 400 && b < 400 && c < 330 && d < 330 && e > 410 && a < 470 &&
b > 320 && c > 280 && d > 290 && e < 450) {

display.setTextSize(2.0);

display.setTextColor(SSD1306_WHITE);

display.setCursor(50, 0);

display.println("Y");

display.display(); // Update the OLED display with the new content

digitalWrite(GREEN_LED_PIN, HIGH);

digitalWrite(BUZZER_PIN, HIGH); // Turn on buzzer for true condition

delay(3000); // Delay to display "Y"
```

```
  display.clearDisplay();

 }



 else if (a > 250 && b > 330 && c < 330 && d < 320 && e > 170 && a < 330 &&
b < 400 && c > 250 && d > 260 && e < 250 == event.acceleration.z > 2.0) {

  display.setTextSize(2.0);

  display.setTextColor(SSD1306_WHITE);

  display.setCursor(50, 0);

  display.println("Z");

  display. display(); // Update the OLED display with the new content

  digitalWrite(GREEN_LED_PIN, HIGH);

  digitalWrite(BUZZER_PIN, HIGH); // Turn on buzzer for true condition

  delay(3000); // Delay to display "Z"

  display.clearDisplay();

 }



 else {
```

```
  digitalWrite(RED_LED_PIN, HIGH); // Turn on red LED for false condition
}

// Display ADC values on OLED
for (int i = 0; i < num_ADC_pins; i++) {
  int flex_ADC = analogRead(ADC_pins[i]);
  float flex_voltage = flex_ADC * VCC / 1024.0;
  float flex_resistance = (fixed_resistor * (VCC / flex_voltage - 1.0)) / 1000.0;

// Print to serial
  Serial.print("ADC value for pin ");
  Serial.print(i);
  Serial.print(": ");
  Serial.println(flex_ADC);
  Serial.println();
```

```
    // Print to OLED display
    display.setTextSize(1);
    display.setTextColor(SSD1306_WHITE);
    display.setCursor(0, 10*i);
    display.print("ADC");
    display.print(i);
    display.print(": ");
    display.print(flex_ADC);
  }
  display.display(); // Update the OLED display
  delay(10000); // Delay for 10 seconds
}
```