DEVELOPMENT OF SIGN LANGUAGE GLOVE (SIGN GLOVE) USING ARDUINO

ZA'IM HAKIMI BIN ROZILAN

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

DEVELOPMENT OF SIGN LANGUAGE GLOVE(SIGN GLOVE) USING ARDUINO

ZA'IM HAKIMI BIN ROZILAN

This report is submitted in partial fulfilment of the requirements for the degree of Bachelor of Computer Engineering Technology (Computer Systems) with Honours

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

2025

UNIVERSITI TEKNIKAL MALAYSIA MELAKA
FAKULTI TEKNOLOGI DAN KEJURUTERAAN ELEKTRONIK DAN KOMPUTER

BORANG PENGESAHAN STATUS LAPORAN
PROJEK SARJANA MUDA II

Tajuk Projek : DEVELOPMENT OF SIGN LANGUAGE GLOVE(SIGN GLOVE) USING ARDUINO

Sesi Pengajian : 2024

Saya ZA'IM HAKIMI BIN ROZILAN mengaku membenarkan laporan Projek Sarjana Muda ini disimpan di Perpustakaan dengan syarat-syarat kegunaan seperti berikut:

1. Laporan adalah hakmilik Universiti Teknikal Malaysia Melaka.
2. Perpustakaan dibenarkan membuat salinan untuk tujuan pengajian sahaja.
3. Perpustakaan dibenarkan membuat salinan laporan ini sebagai bahan pertukaran antara institusi pengajian tinggi.
4. Sila tandakan (✓):

☐	**SULIT***	(Mengandungi maklumat yang berdarjah keselamatan atau kepentingan Malaysia seperti yang termaktub di dalam AKTA RAHSIA RASMI 1972)
☐	**TERHAD***	(Mengandungi maklumat terhad yang telah ditentukan oleh organisasi/badan di mana penyelidikan dijalankan.
/	**TIDAK TERHAD**	

Disahkan oleh:

____

(COP DAN TANDATANGAN PENYELIA)

Alamat Tetap: ....
....
....
....
....

Tarikh : 28 Januari 2025

Tarikh : 28 Januari 2025

*CATATAN: Jika laporan ini SULIT atau TERHAD, sila lampirkan surat daripada pihak berkuasa/organisasi berkenaan dengan menyatakan sekali tempoh laporan ini perlu dikelaskan sebagai SULIT atau TERHAD.

DECLARATION

I declare that this project report entitled "Project Title" is the result of my own research except as cited in the references. The project report has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

Signature	:	
Student Name	:	ZA'IM HAKIMI BIN ROZILAN
Date	:	28 Januari 2025

APPROVAL

I hereby declare that I have checked this project report and in my opinion, this project report is adequate in terms of scope and quality for the award of the degree of Bachelor of Computer Engineering Technology (Computer Systems) with Honours.

Signature : ....

Supervisor Name : MA TIEN CHOON

Date : 28 Januari 2025

Signature : ....

Co-Supervisor Name (if any) : ....

Date : ....

DEDICATION

To my beloved mother, Khuznita Binti Mohd Yatim, and father, Rozilan Bin Abdul Rahim, who have been the source of inspiration and strength when I'm feeling down and giving up, who continually provide their moral, spiritual, emotional, and financial support.

And

To dearest families, Nur Athirah Hazwani and Adam Hatim who shared their words of advice and encouragement to finish this project.

And

To my friends, Roobakkanthan, Arif, Qayyum, Wan, Ammar, Haziq, Hafiz, Nazrul and Saifudin whose has contributed and guide me in completing my work, giving their thought, idea and solution that greatly help me in solving all the problems that occur in my journey to finish this project.

And lastly,

To the Almighty God, thank you for the guidance, strength, power of mind, protection, skills and for giving me a healthy life. It is because of your mighty power; I was able to finish this project successfully.

ABSTRACT

The design of a Sign Language Glove (Sign Glove) using Arduino technology, with the objective of increasing communication for those people who have hearing and speech problems. The Sign Glove is was designed to transform hand motions from American Sign English (ASL) into written text and spoken English, thus bridging the communnication divide between the deaf people and others who are not acquainted with sign language. The glove is combined with flexible flex sensors that can detect finger movements and accelerometer to capture hand posture and motion. The sensors are connected to Arduino microcontroller, which analyzes the sensor data and translate it into text presented on the screen and also, can produces audio output through a connected speaker. Other than that, the development method will use the design of the glove's hardware, the programming of the Arduino that allows excellent gesrture recognition, and the implementation of an intuitive user interface. The system's accuracy was tested by a series of test employing unique ASL signals, showing dependability and speed in translating in real-time. The results are going to show an incredible degree of precision and a user-friendly interface, suggesting that the Sign Glove has the capacity to considerably improve the daily contact for those persons who are deaf or hearing difficult. This work being done not only demonstrates the practical use of Arduino in assistive technology, but also shows the potential of technological devices in developing inclusive communication solutions. In the future, the sign glove might be improved with a larger vocabulary, more portability and a combanation of machine learning techniques to increase the accuracy of gesture detection.

ABSTRAK

Reka bentuk sarung tangan bahasa isyarat (Sign Glove) menggunakan teknologi Arduino, dengan objektif untuk meningkatkan komunikasi bagi orang-orang yang mempunyai masalah pendengaran dan pertuturan. Sign Glove direka untuk mengubah gerakan tangan daripada American Sign English (ASL) kepada teks bertulis dan pertuturan Bahasa Inggeris, sekali gus merapatkan jurang komunnicasi antara orang pekak dan orang lain yang tidak mengenali bahasa isyarat. Sarung tangan digabungkan dengan sensor flex fleksibel yang boleh mengesan pergerakan jari dan pecutan untuk menangkap postur tangan dan gerakan. Sensor disambungkan ke mikropengawal Arduino, yang menganalisis data sensor dan menterjemahkannya ke dalam teks yang dibentangkan pada skrin dan juga, boleh menghasilkan output audio melalui pembesar suara yang disambungkan. Selain itu, kaedah pembangunan akan menggunakan reka bentuk perkakasan sarung tangan, pengaturcaraan Arduino yang membolehkan pengiktirafan gesrture yang sangat baik, dan pelaksanaan antara muka pengguna intuitif. Ketepatan sistem ini diuji oleh satu siri ujian yang menggunakan isyarat ASL yang unik, menunjukkan kebolehgunaan dan kelajuan dalam menterjemah dalam masa nyata. Hasilnya akan menunjukkan tahap ketepatan yang luar biasa dan antara muka yang mesra pengguna, menunjukkan bahawa Sign Glove mempunyai keupayaan untuk meningkatkan hubungan harian bagi orang-orang yang pekak atau sukar mendengar. Kerja yang dilakukan ini bukan sahaja menunjukkan penggunaan praktikal Arduino dalam teknologi bantuan, tetapi juga menunjukkan potensi peranti teknologi dalam membangunkan penyelesaian komunikasi inklusif. Pada masa akan datang, sarung tangan tanda mungkin diperbaiki dengan perbendaharaan kata yang lebih besar, lebih mudah alih dan gabungan teknik pembelajaran mesin untuk meningkatkan ketepatan pengesanan gerak isyarat.

ACKNOWLEDGEMENTS

First and foremost, I would like to express my gratitude to my supervisor, Sir Ma Tien Choon for his precious guidance, words of wisdom and patient throughout this project.

I am also indebted to Universiti Teknikal Malaysia Melaka (UTeM) and my parents for the financial support through PSM 1 and PSM 2 which enables me to accomplish the project. Not forgetting my fellow colleague, Roobakkanthan, Arif, Qayyum, Wan, Ammar, Haziq, Hafiz, Nazrul and Saifudin for the willingness of sharing his thoughts and ideas regarding the project.

My highest appreciation goes to my parents and family members for their love and prayer during the period of my study. An honourable mention also goes to Roobakkanthan for all the motivation and understanding.

Finally, I would like to thank all the fellow colleagues and classmates, the faculty members, as well as other individuals who are not listed here for being co-operative and helpful.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

δ	-	Voltage angle

LIST OF ABBREVIATIONS

V	-	Voltage

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Background

In over the last few years technology has grown to be something that is almost a necessity, it has provided incredible breakthroughs for those who are struggling. One of them is the assistive technology designed to help bring together people using sign language and those who know nothing about it. While for those people who are deaf or have some hearing impairments these technologies can enhance their life and give them a possibility to communicate and be integrated to society. The Sign Language Glove (Sign Glove) project, which applies Arduino technology, is the creative effort intending to improve communication of the deaf or hard of hearing individuals. By combining hardware components like flex sensors, Arduino Nano, ESP8266, and software like the Blynk app, create Sign Glove that can translate sign language gestures into text in real-time. This wearable device detects finger movement, then processes the data with Arduino algorithms, and displays the translated output on the Blynk app. Sign Glove aims to empower people with different communication needs, promote diversity and close communication gaps by developing hardware and software solutions. With the use of technology of facilitate simple communication between sign language users and non-speakers, accessibility solutions like Sign Glove can create a revolutionize society by promoting inclusivity and improving quality of life.

1.2 Problem Statement

For many people who are deaf or hard of hearing, communication can be difficult because most people do not know sign language. This can cause major social difficulties for them in normal life interactions with others, for instance in school or at work or common social activities. This consequently makes them feel left out, all alone as well as failure to get the understanding that they would want from others. Of course, there are some tools such interpreters or written messages, but these solutions are not always applicable or effective enough to provide real-time coverage of the conversations. This means that there is a general need to establish a more effective means of assisting sign language users to interact with people who do not understand sign language. A solution that will help the two groups to better explain themselves to each other could enhance their lives as well as help them become more functional in society. The Sign Glove project aims to solve this problem by creating a wearable glove that can translate sign language into text in real-time. This device will make it possible for people with hearing impairment to effectively converse with non-signing persons thus improving the flow, speed and inclusiveness of communication processes.

1.3 Project Objective

The purpose is one of the most crucial things that need to be thought out and acknowledged effectively as it shall serve as a guideline in designing a project. For this project, the main goal of the Sign Glove project is to create a wearable glove that helps people who are deaf or hard of hearing communicate more easily with others who don't know sign language. The specific objectives of the project are:

a) By Create a wearable glove that is comfortable and has sensors to detect hand and finger movements.

b) Can translate sign language gestures into text or speech that others can understand.

c) Can connect the glove to the Blynk app to show the translated text on a mobile device.

1.4 Scope of Project

The scope of this project are as follows:

a) The project would construct a comfortable glove that enhances the detection of finger movements.

b) The glove will incorporate flex sensors, and electronic devices such as the Arduino Nano and ESP8266 to capture and analyze movement.

c) The sign glove is designed to convert sign language signals into an appropriate text in real-time with the help of the Blynk application.

d) The sign glove will be interfaced with Blynk app to show result on a portable gadget.

e) Checking the sign glove to ensure it accurately translates gestures and would work with efficiency to offer good results.

f) The project will focus on basic sign language gestures.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

Sign language is visual language that uses hand shape, facial expression, gestures, and body language that is used by people who are deaf to communicate with normal people. However, normal people do not understand sign language and this will create a huge problem for them to communicate with deaf people. Other than that, sign language is not easy to learn due to its different language and level. As a result, there is a need to create a system that can help in converting sign language into voice and voice to sign language to provide efficient and easy communication between deaf people and normal people.

2.2 Design And Development Of Hand Gesture Recognition System For Speech Impaired People

Lots of method have been proposed for hand gesture recognition by using image processing. “Hand Gesture Recognition System using Image Processing” makes use of modified SIFT algorithms, which are digital image processing techniques. With the SIFT algorithm, the decoding of the sign language is successful. The advantages of algorithm is its fast-processing speed, which allow the real-time results. Despite the system is fast, it still needs expensive components[1].

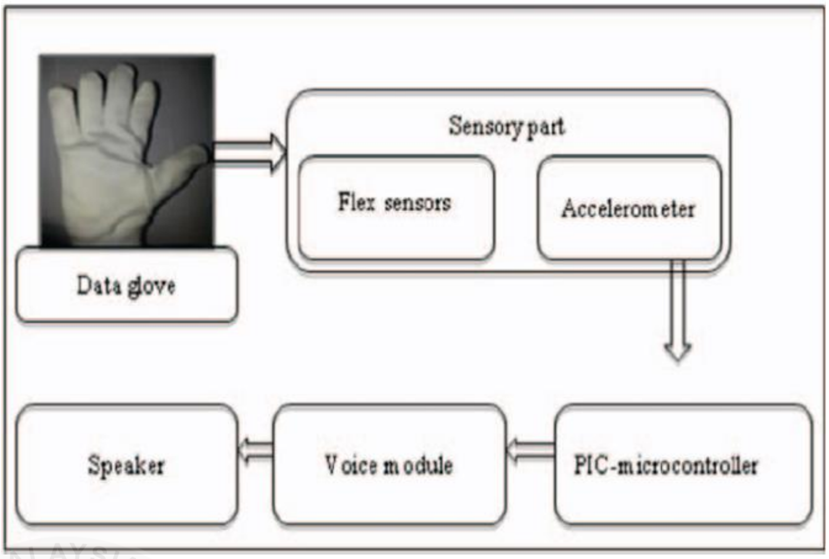

Figure 2.1 Block Diagram of the ISL hand gesture recognition system[1]

2.3 Sign Language Recognition Using Image Processing

"Sign Language Recognition using Image Processing" can help deaf people to communicate by using SURF (Speeded Up Robust Features) algorithms for hand motion recognition. Hand gestures will be recorded by a video camera and resulting the footage which it will be divided into frames. The SURF methods are used to identify and extract features for every frame. By using MATLAB, the application of system can make features extraction, matching and detection for sign language translation[2].

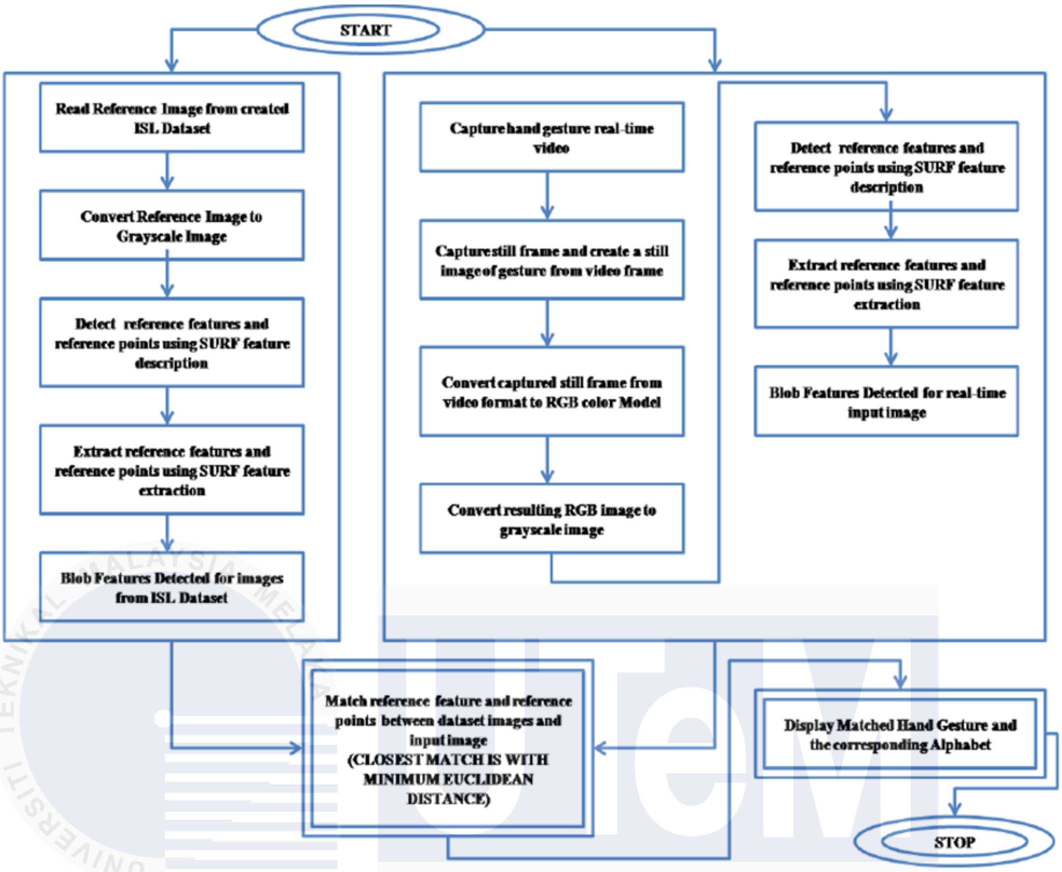

Figure 2.2 Hand Gesture Recognition System for ISLS[2]

2.4 A Smart Hand Glove That Converts Gesture Into Text & Speech To Assist The Handicapped (Hand Talk)

"A Smart Hand Glove that Convert Gesture into Text & Speech to Assist the Handicapped (Hand talk) " is created for those people who are deaf or mute, so they can easily to communicate. It will detect hand motion using flex sensors that are attached to a glove. An Arduino board will detect hand gestures and produce text and audio outputs. The device also links to a server to enable emergency replies and control appliances, which makes it a practical and user-friendly option for speech impaired people who suffer communication challenges[3]

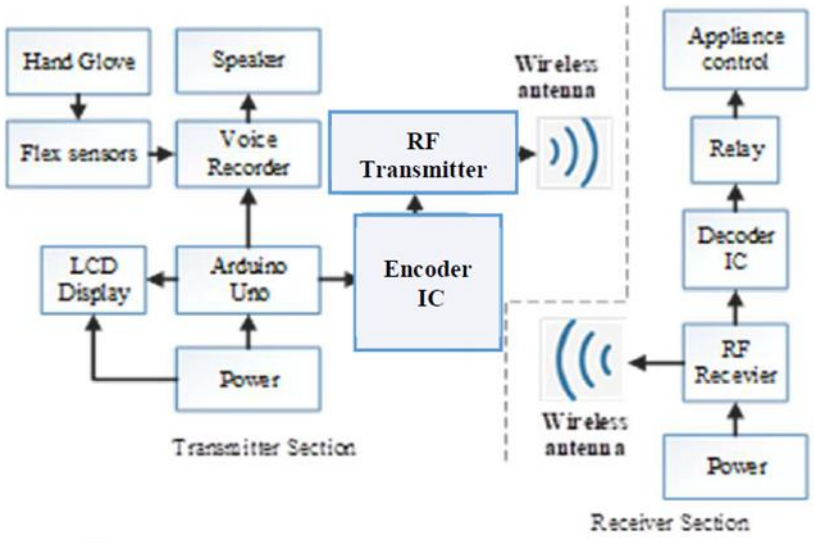

Figure 2.3 Block diagram of the proposed method[3]

2.5 Microcontroller Based Sign Language Glove

The "Microcontroller Based Sign Language Glove" project design to solve the communication problem faced by those who are dumb, deaf or have speech impairments. This device can convert Sign Language alphabets into digital signal by using flex sensors that are attached to a glove. The data will be processed by a microcontroller and then will show result on the LCD panel. Through this accessible technology, the project strives to increase the independence and quality of life of persons with disabilities by assisting them to communicate with members of the community and with each other[4].

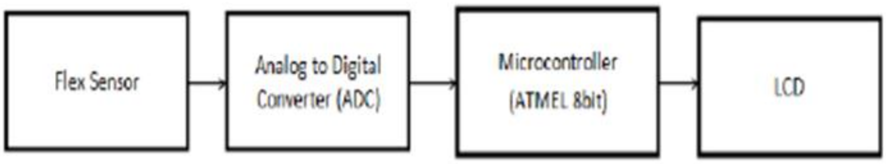

Figure 2.4 Block diagram of prototype project[4]

2.6 Smart Gloves For People With Speech Disability

The "Smart Gloves for People with Speech Disability" project will be using device that combines gyroscopes, accelerometers, and flex sensors to record hand motions and translate voiced into sign language. This helps deaf or speech-impaired people to communicate effectively. By using CNN (convolutional neural networks) on a Raspberry Pi, the system will be carefully developed to reach excellent identification accuracy of 88.97% for the prototype. By using this device will help close the communication gap between people disability hearing and normal people. Also, this not only makes it easier for people to find better jobs, but also gives them the opportunity to live more independent lives[5].

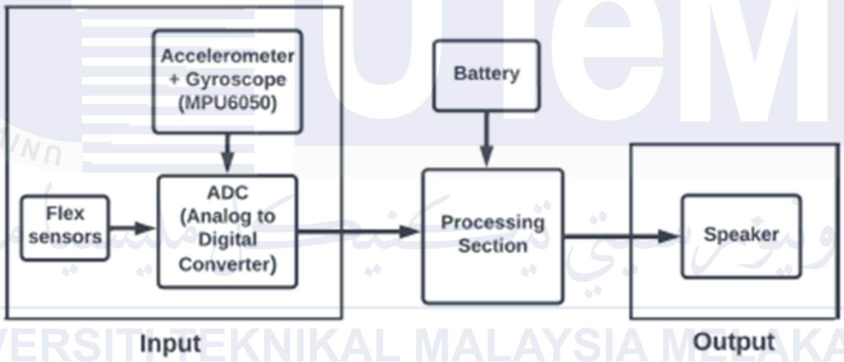

Figure 2.5 Blocked diagram of the proposed system[5]

2.7 Development Of A Wearable Device For Sign Language Recognition

The "Development of a Wearable Device for Sign Language Recognition" can translate text and audio from ASL (American Sign Language) using flex sensors and a motion sensor. This glove will be based on gadget recognizes hand and finger movements, also it translates them into appropriate words and sentences by using an Arduino microcontroller. The translated output is then shown on LCD screen and spoken by a speaker. This will help the deaf people to communicate more effectively. The project demonstrates how wearable technology can improve accessibility and help close communication gaps for those who have hearing loss[6].

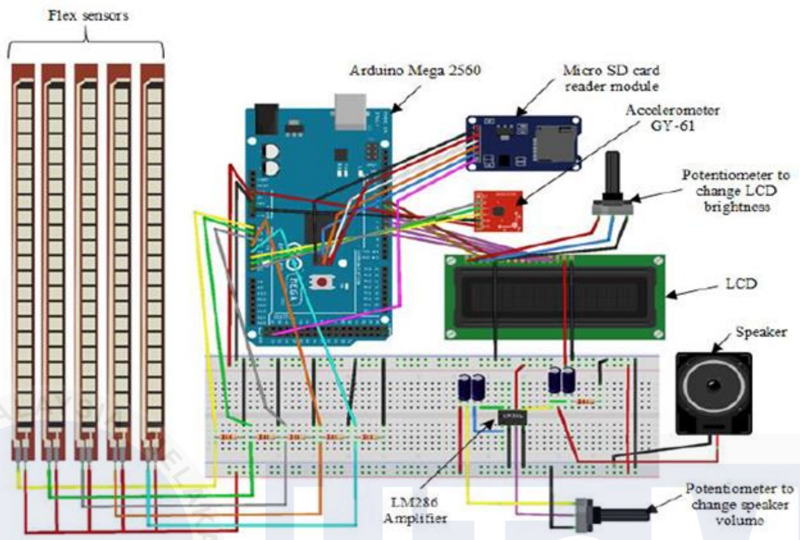

Figure 2.6 Circuit diagram for sign language translator device[6]

2.8 Design Of Smart Gloves

The "Design of Smart Gloves" focuses on developing an affordable solution to help people communicate who have speech and hearing impairments. This project will use flex sensors that are built into gloves to record hand motions. A microcontroller then analyzes the gestures to produce speech that is synthesized. By overcoming the communication gap between normal people and without disabilities people, the system hopes to improve more relationships and better job prospects for people with disabilities[7].

Figure 2.7 American Sign Language Alphabet[7]

2.9 Deaf-Mute Communication Interpreter

The “Deaf-Mute Communication Interpreter” translates ASL (American Sign Language) motion to text and audio sound by using a sensor-based device built up of flex, tactile and accelerometer sensors. The letters "A," "B," "C," "D," "F," "I," "L," "O," "M," "N," "T," "S," and "W" were the only thirteen signs that could successfully translate into their appropriate alphabets. Three letters, M, N, and T were made more accurate by the use of a touch sensor[8].

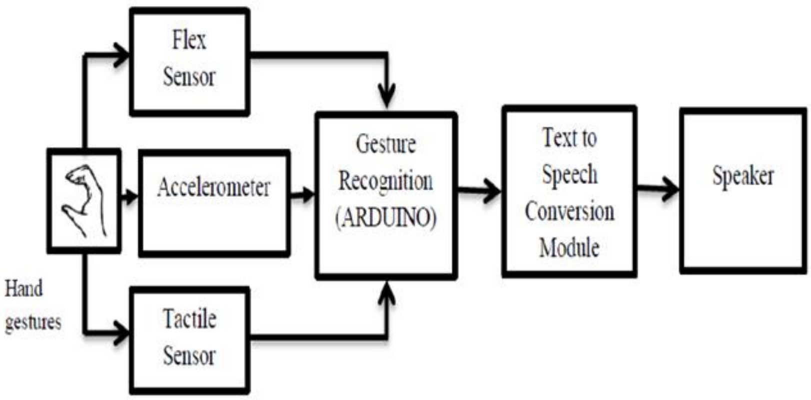

Figure 2.8 Block diagram of device[8]

2.10 Sign Language To Speech Translation System Using PIC Microcontroller

The "Sign Language to Speech Translation System using PIC Microcontroller" is built up of gyro sensors that produce a signal matching to orientation of the hand motion and flex sensors that recognize finger motions. Through the automatic translation of sign language into speech, this device help improve the ability of people with speech impairments to express themselves. Other than that, the device will use the APR9600 speech module and a PIC microcontroller to improve the quality of life for deaf people with speech disorders by allowing real-time communication. Furthermore, it can communicate over long distances utilizing RF wireless transmission, which boosts its mobility and accessibility for everyday use[9].

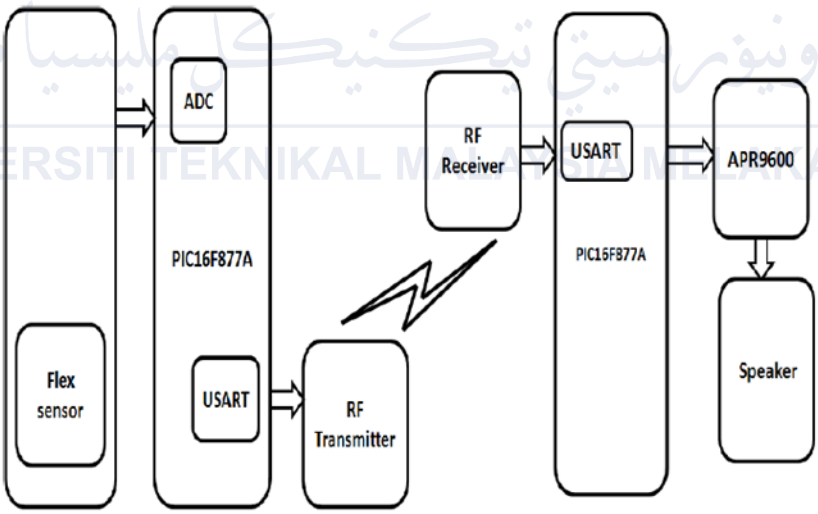

Figure 2.9 Block diagram of Sign Language Translation[9]

2.11 Smart Glove Using ARDUINO With Sign Language Recognition System

The “Smart Glove Using Arduino with Sign Language Recognition System" aids communication for the deaf and speech impaired by using flex sensors to translate hand motions into text and recorded audio. The Arduino powered device will record sensor data that corresponds to hand movements, shows the numbers on LCD, and uses a voice recording module to play back sounds. The goal of this affordable, portable gadget is to improve the independence and communication skills of individuals with disabilities[10].

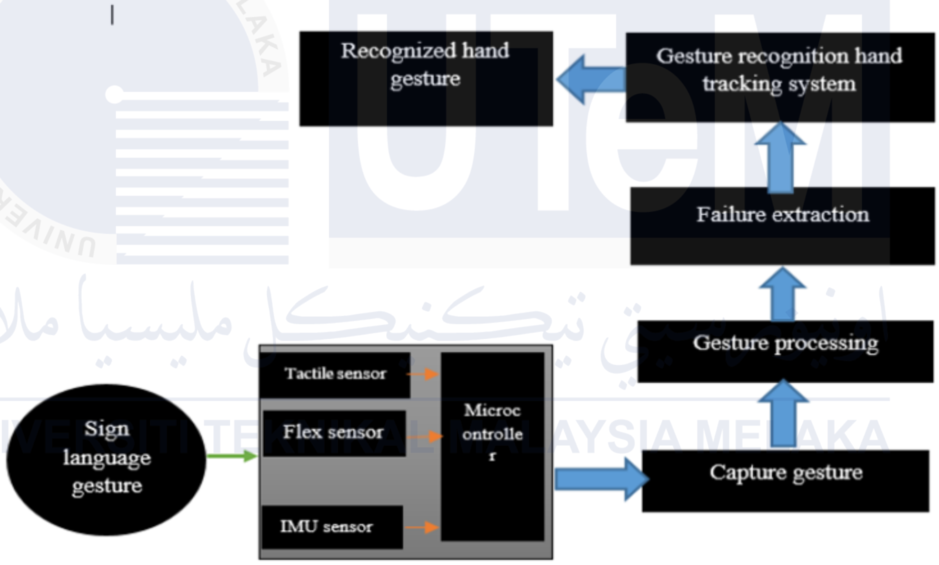

Figure 2.10 SL gesture data collection and recognition using a glove-based system[10]

2.12 Smart Glove For Sign Language Translation Using IoT

Based on this research, it introduces a method for translating sign language by using intelligent gloves with Internet of Things(IoT) technology. The gloves will be used with flex sensors and accelerometers so it can capture hand movements. After that, the movements that have been captured will be translated in real-time using a smartphone app. The introduction of IoT enables smooth wireless data transmission, enhancing accessibility for people with hearing loss while showcasing the revolutionary capacity of IoT in tackling practical obstacles[11].

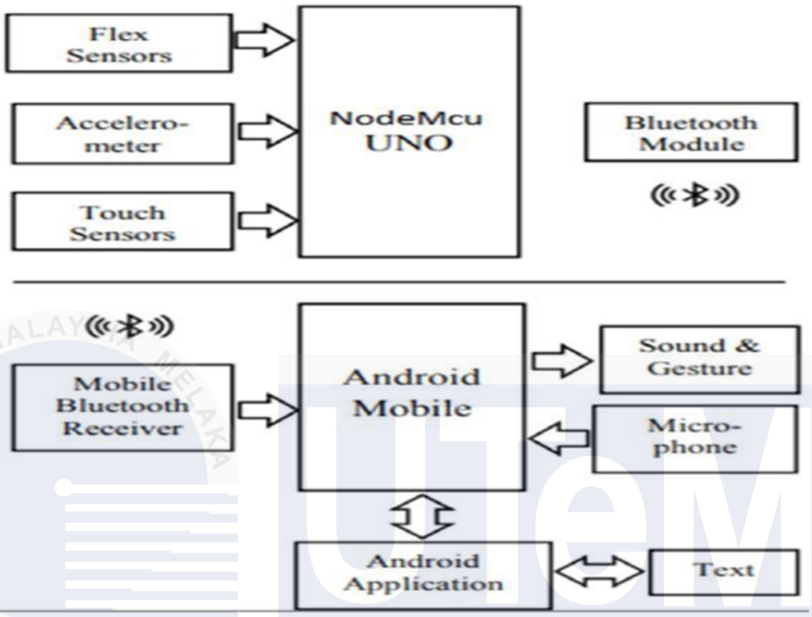

Figure 2.11 Block diagram of smart glove[11]

2.13 Development Of A Wearable Sensor Glove For Real-Time Sign Language Translation

This project will be focusing on creating a wearable sensor glove that can translate a sign language movement into text and speech. With the combination of sensors to detect movement and motion, which allows for correct analysis. This device is created to help people with hearing loss communicate by providing a portable and efficient substitute for sign language translation. Finally, with the improvement of wearable technology sensor glove, this ensures accessibility and inclusion in a variety of setting by allowing for real-time translation[12].

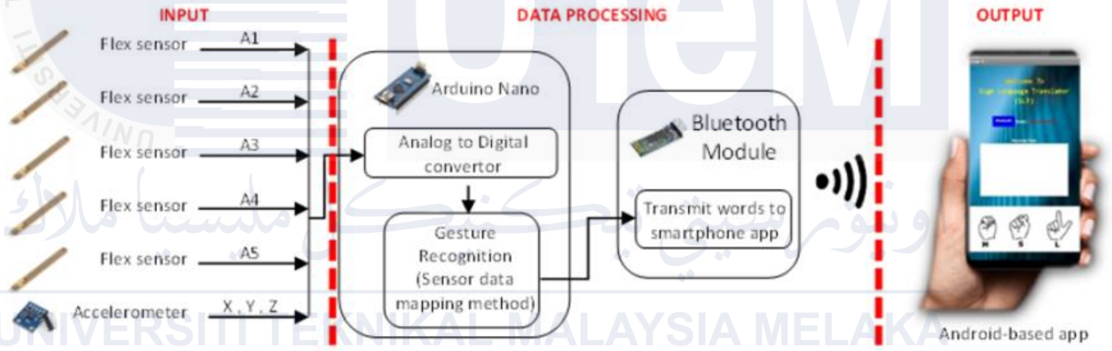

Figure 2.12 Overview of the proposed device[12]

2.14 Advances In Machine Translation For Sign Language

The research project about advances in machine translation for sign language: Approaches, Limitations and Challenges is providing a detailed overview of methodologies and challenges in building this sign language translation system. Other than that, by reviewing several strategies, it talks about their advantages and disadvantages and emphasizes how critical it is to solve the obstacles to improve the accessibility for people with hearing loss. This study can help and be useful for practitioners and scholars who are interested in creating sign language translation using machine learning[13].

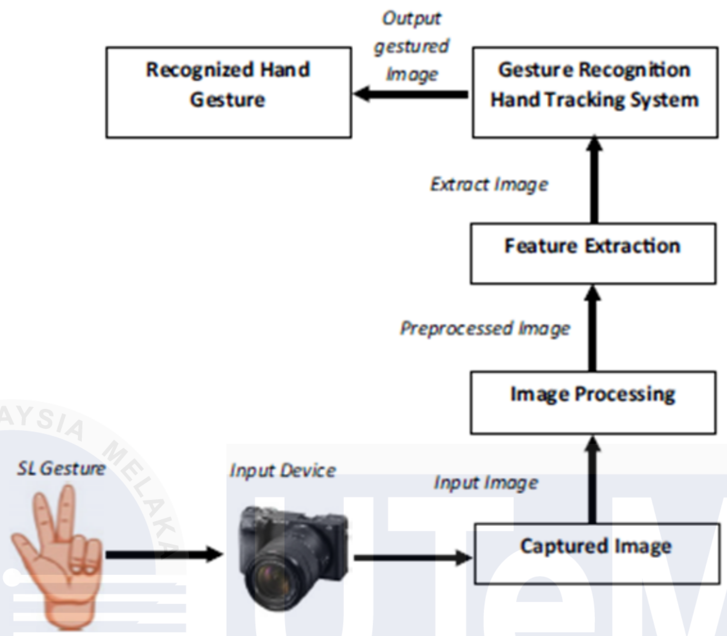

Figure 2.13 Step in software-based gesture[13]

2.15 Development Of A Sign Language Recognition System Using Machine Learning

This project is about system that can help understanding sign language hand movement by using machine learning. Researchers are trying to develop algorithms that can accurately identify and transform hand gestures into understandable forms by using sign language data. Next, is using machine learning that can help provide aids for those people who have hearing loss by translating sign language in real-time. Finaly, with assistive technology this can help more people with hearing loss to have better communication and inclusion[14].

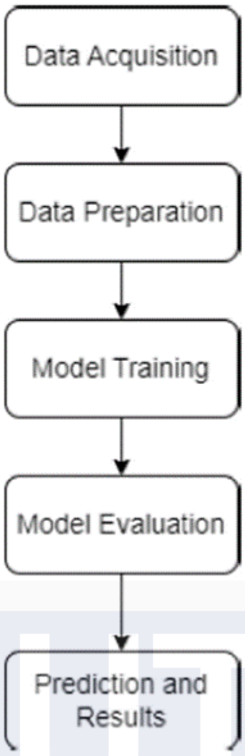

Figure 2.14 Block diagram of system design[14]

2.16 Design And Implementation Of Smart Gloves For The Specially Privileged

This research is about producing special gloves call a “smart glove”. It is for people who have experience issues with their hands, especially for those who are different skilled. By creating these gloves, it will help them to communicate better. By using special equipment such as a sensor, it can help understand hand movement and gestures. Other than that, these gloves will help more people to communicate using sign language and even help to translate these signs into text and speech. This project was very successful because it helps people who have trouble hearing and speaking to communicate more freely and efficiently[15].

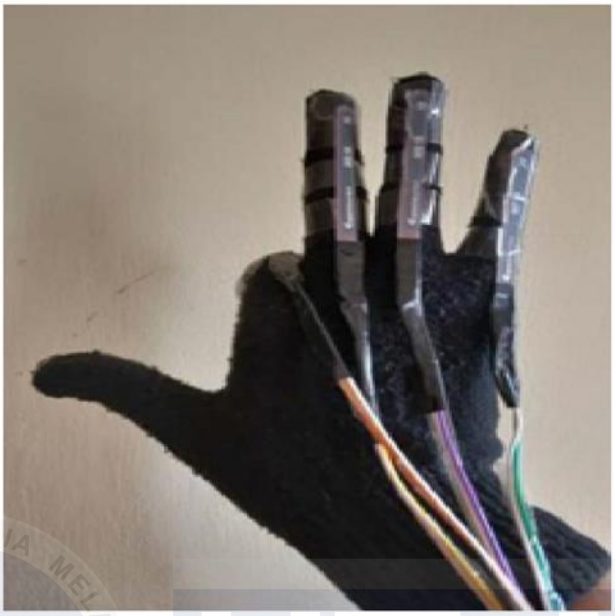

Figure 2.15 Gesture indicate food[15]

2.17 Sign Language Recognition For Arabic Alphabets Using Transfer Learning Technique

The research focus on teaching a computer to interpret sign language for the Arabic alphabet using a technique that call transfer learning. The objective is to use the computer to recognize other languages to translate Arabic sign language. By doing this, it will make a computer recognize Arabic sign language gesture more effectively. Finally, this will help many people to use sign language to communicate with Arabic people easier by using technology[16].

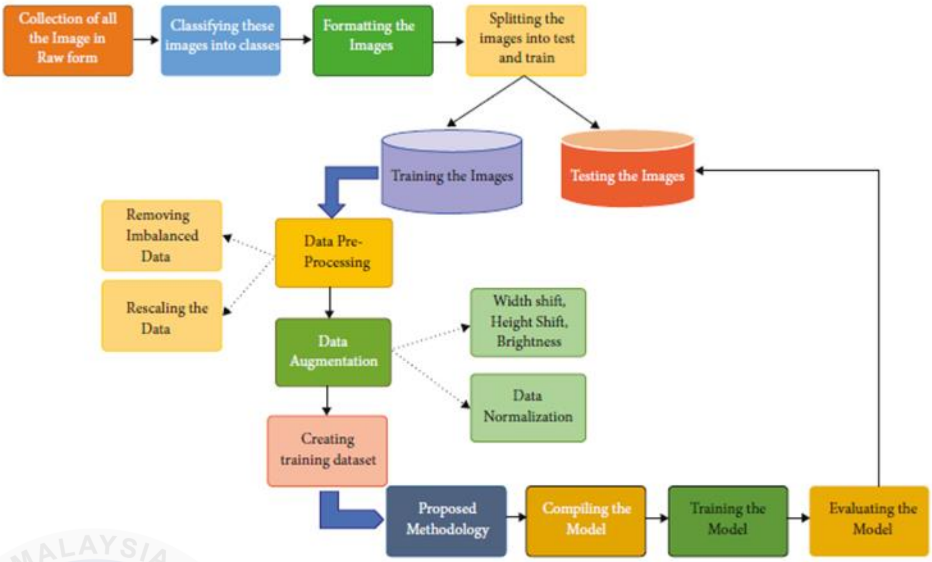

Figure 2.16 The flow of the work[16]

2.18 Real Time Static And Dynamic Sign Language Recognition Using Deep Learning

The research project is about developing a system that can recognize real time static and dynamic sign language recognition in real time by only using Deep learning. By using advanced Artificial Intelligence(AI), the system can read these hand gestures more quickly and precisely. This real-time capability is vital for successful communication, particularly in fast-paced circumstances. The technology intends to improve communication for persons who use sign language, making their interactions with others and with technology more fluid and efficient. This can dramatically boost accessibility and inclusion for the hearing impaired[17].

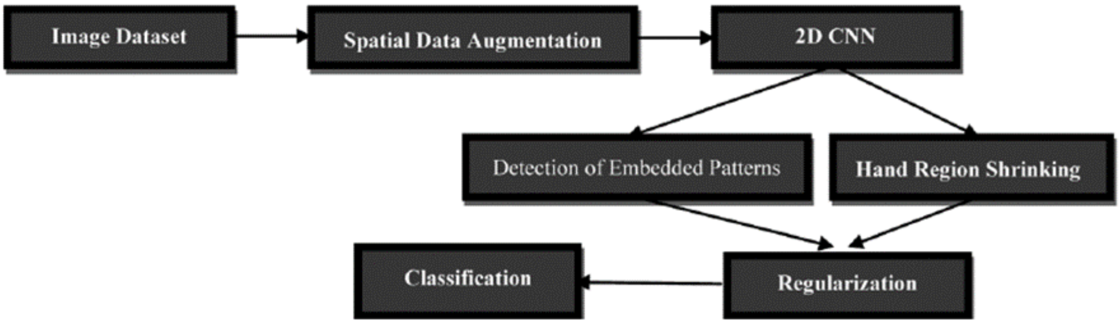

Figure 2.17 Image dataset processing model with 2D-CNN regularization to interpret gestures in both simple and complex backgrounds[17]

2.19 Intelligent Sign Language Recognition Using Image Processing

In this project, will be using image processing to construct a smart system that identifies sign language. By employing the technology, we can try to snap a photo of hand motion and then try to interpret by figuring out what is being signed. Next, the technology will convert sign language into text or speech by examining the form and movement of the hand that has just been taken picture. With this, it will make communicate more effectively with others for those who use sign language. The project is to increase accessibility and communication for those people who hear impaired by using advanced picture analysis technology[18].

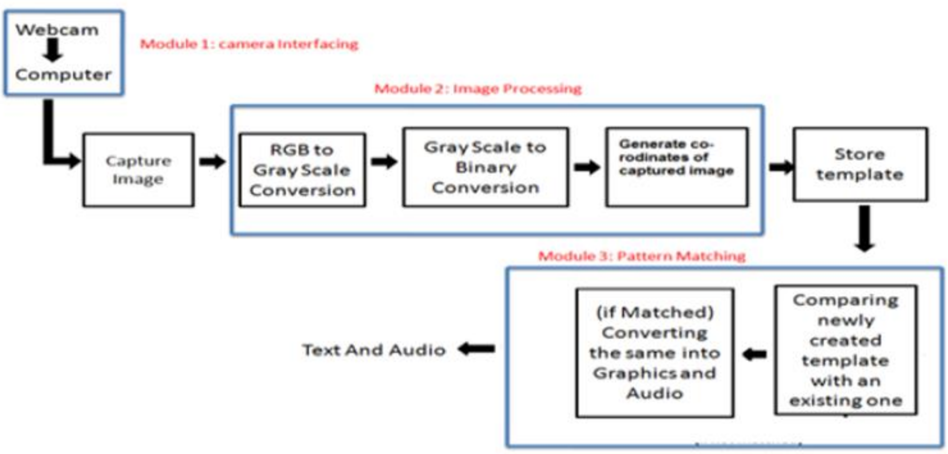

Figure 2.18 System diagram flow work[18]

2.20 Sign Language Recognition Based on Intelligent Glove Using Machine Learning Techniques

In this study, an intelligent glove using machine learning techniques is used to develop a system for understanding sign language. It will have a feature sensor that will detect hand movements and gestures. By analyzing these movements, machine learning algorithms will interpret and transform sign language into text and speech. Finally, any person who uses sign language should be able to communicate more easily, which will boost social engagement when they try to use this device. By using machine learning, the system can read wide range of action, increasing its usefulness and dependability[19].

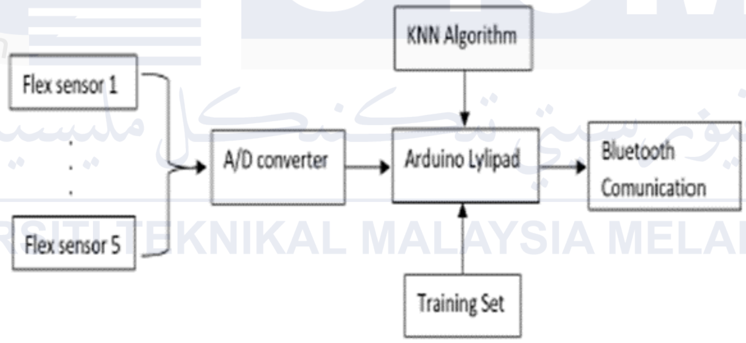

Figure 2.19 Block diagram of Intelligent Glove[19]

2.21 Systems-Based Sensory Gloves For Sign Language Recognition State Of The Art

This research study is on the evolution and progress of sensory gloves used for sign language between 2007 and 2017. It studies several methods that use sensors in glove to detect hand movements and gestures that will translate into text and speech. The assessment focuses on cutting-edge technology and developments achieved over the previous decade, demonstrating how these sensory gloves have gotten more precise and efficient. This research is crucial since it highlights the progress in making communication easier for persons who use sign language, opening the door for more advanced and accessible options in the future[20].

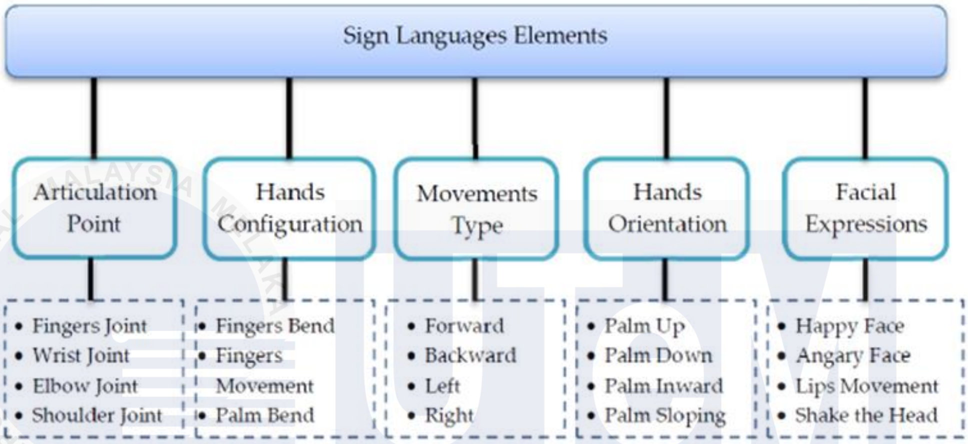

Figure 2.20 The element connected to sign language gesture formation[20]

2.22 Comparison Table

Based on table 2.1 is the comparison table from all literature review about the previous project Sign Glove that has been created.

Table 2.1 Comparison Table Literature Review

Reference	Component	Methodology	Advantages	Challenges
[1]	Modified SIFT algorithms	Hand gesture identification using SIFT algorithms is achieved by the application of digital image processing techniques.	Fast processing speed, real-time results.	Requires expensive components.
[2]	SURF algorithms	Hand motion identification utilizing the Speeded Up Robust Features (SURF) algorithm for feature extraction and matching in MATLAB.	Effective feature extraction for sign language translation.	Relies on video footage, potentially slower real-time performance.
[3]	Flex sensors, Arduino	A glove-based system is used to convert hand motions into text and speech. This system is also capable of connecting to a server to access other services.	Practical, user-friendly for speech impaired individuals.	Dependency on reliable server connection for extended functionality.

Reference	Component	Methodology	Advantages	Challenges
[4]	Flex sensors, Microcontroller	Translates sign language gestures into digital impulses that are visually represented on an LCD panel.	Enhances communication independence for users with disabilities.	Limited to displaying simple alphabets, may not capture nuanced gestures.
[5]	Gyroscopes, accelerometers, flex sensors, CNN	Utilizes Convolutional Neural Network (CNN) on Raspberry Pi to identify gestures.	High accuracy in gesture identification (88.97%).	Complex setup with multiple sensors and CNN implementation.
[6]	Flex sensors, motion sensor, Arduino	This device utilizes sensors and a microcontroller to convert American Sign Language (ASL) into both written text and audio. The converted information is then displayed on an LCD screen and communicated through a speaker.	Real-time translation, improves communication effectiveness.	Hardware integration challenges, usability in diverse environments.
[7]	Flex sensors, Microcontroller	Generates spoken language by analyzing hand movements captured by flex sensors embedded in gloves.	Affordable, portable solution for speech and hearing impaired.	Accuracy in gesture recognition and speech synthesis may vary.

Reference	Component	Methodology	Advantages	Challenges
[8]	Flex, tactile, accelerometer sensors	This application converts American Sign Language (ASL) gestures into written text and audio. It improves accuracy by utilizing touch sensors.	Successful translation of specific ASL letters.	Limited gesture vocabulary, effectiveness across diverse gestures.
[9]	Gyro sensors, flex sensors, PIC microcontroller	This technology enables the conversion of sign language into audible speech, facilitating long-distance communication using RF wireless transmission.	Real-time communication enhancement for speech-impaired individuals.	Complexity in RF transmission setup, potential interference issues.
[10]	Flex sensors, Arduino	The system captures manual movements, exhibits numerical values on a liquid crystal display, and reproduces audio via a voice recording module.	Affordable, improves independence and communication skills.	Limited to simple gesture recognition and playback capabilities.
[11]	Flex sensors, accelerometers, IoT	Translates hand movements using IoT-enabled gloves, real-time translation via smartphone app.	Enhanced accessibility with IoT connectivity.	Reliability on smartphone app, dependency on internet connectivity for real-time translation.

Reference	Component	Methodology	Advantages	Challenges
[12]	Wearable sensor glove	Converts sign language gestures into text and speech, focusing on real-time translation.	Portable, efficient communication aid for hearing-impaired individuals.	Ensuring accuracy and reliability in real-time translation.
[13]	Machine translation	Reviews approaches in machine translation for sign language, stresses AI-based developments.	Explores various methodologies for improved accessibility.	Challenges in achieving real-time translation and accuracy with AI.
[14]	Machine learning algorithms	Develops algorithms to identify and transform sign language gestures into understandable forms.	Real-time translation aid for hearing-impaired individuals.	Complexity in training machine learning models for diverse gestures.
[15]	Sensor-based gloves	Assists communication for people with hand disabilities, translates gestures into text and speech.	Enhances communication for differently abled individuals.	Ensuring robustness and adaptability across various hand impairments.
[16]	Transfer learning	Teaches computers to recognize Arabic sign language, increases gesture recognition success.	Improves accessibility for Arabic-speaking individuals using sign language.	Adaptability across different sign languages and cultural gestures.
[17]	Deep learning	Recognizes static and dynamic sign language motions in real-time using deep learning algorithms.	High precision and real-time gesture interpretation.	Resource-intensive training and deployment of deep learning models.

Reference	Component	Methodology	Advantages	Challenges
[18]	Image processing	Constructs a system using image processing to understand sign language, translates motions to text or speech.	Advanced picture analysis technology for effective communication.	Challenges in real-world implementation and environmental variability.
[19]	Machine learning techniques	Develops intelligent gloves for sign language recognition, turns motions into text and voice.	Enhances social engagement and communication for sign language users.	Ensuring adaptability and accuracy across a wide range of gestures and environments.
[20]	Sensor gloves	Studies developments in sensor gloves for sign language, focuses on enhancing precision and efficiency.	Evolution in technology for enhanced communication accessibility.	Addressing historical limitations in accuracy and real-time responsiveness.

CHAPTER 3

METHODOLOGY

3.1 Introduction

This chapter describes the methodology used for the project. The overview, device, and design of the project are covered in this chapter. Furthermore, a full list and description of the hardware and software employed in this project will be presented.

3.2 Selecting and Evaluating Tools for a Sustainable Development

Effective tool choosing and evaluation were critical to the Sign Language Glove project to ensure the effectiveness of the system. For the finger motions, flex sensors are used in the project and an Arduino Nano is used as the core processor in view of its easy programming and sufficient processing ability. Also, information was transmitted through ESP8266 module and was received by an Android application (Blynk) which has great number of applications in it's library and has excellent programming environment. The output text rephrased the result.

For this purpose, the functionality of each tool, its ability to integrate with others and compatibility of each tool was critically reviewed to ensure that it met the specific needs of this project and was suitability for data processing and transmission. These tools were chosen to keep the project sustainable in the long term and, in addition to increasing the speed of development, brought a significant boost to the performance of the end product and minimal future revisions and maintenance. Figure 3.1 show the development work.

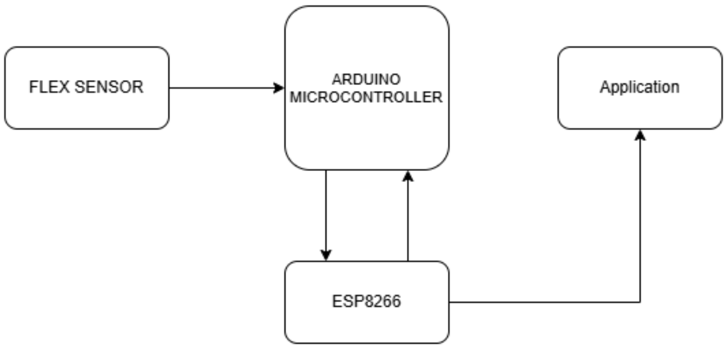

Figure 3.1 Block diagram of Sign Glove

3.3 Methodology

The procedures of the Android application (Blynk) and the sign glove are both illustrated in this documentation, which describes the operational methodology of the Sign Language Glove project. Firstly, the glove begins by detecting finger movements using flex sensors. Once the sensors detect the movement, it will record the data to identify the specific sign. The gloves can identify seven different signs. When a sign is detected it produces the result and forwards it to the Android application Blynk. After going through each step, the next step is the Blynk application. Firstly, turn on the Blynk application by connecting to the Wi-Fi connection. After that it connects and then waits for the data to be transmitted from the glove. Finally, the result will be display the text after the information has been received by the application. This system is an effective means by which hand gestures moveable and recognizable output that can enhance communication. Figure 3.3 and 3.4 shows the project's working system to better comprehend the project constructure flow.

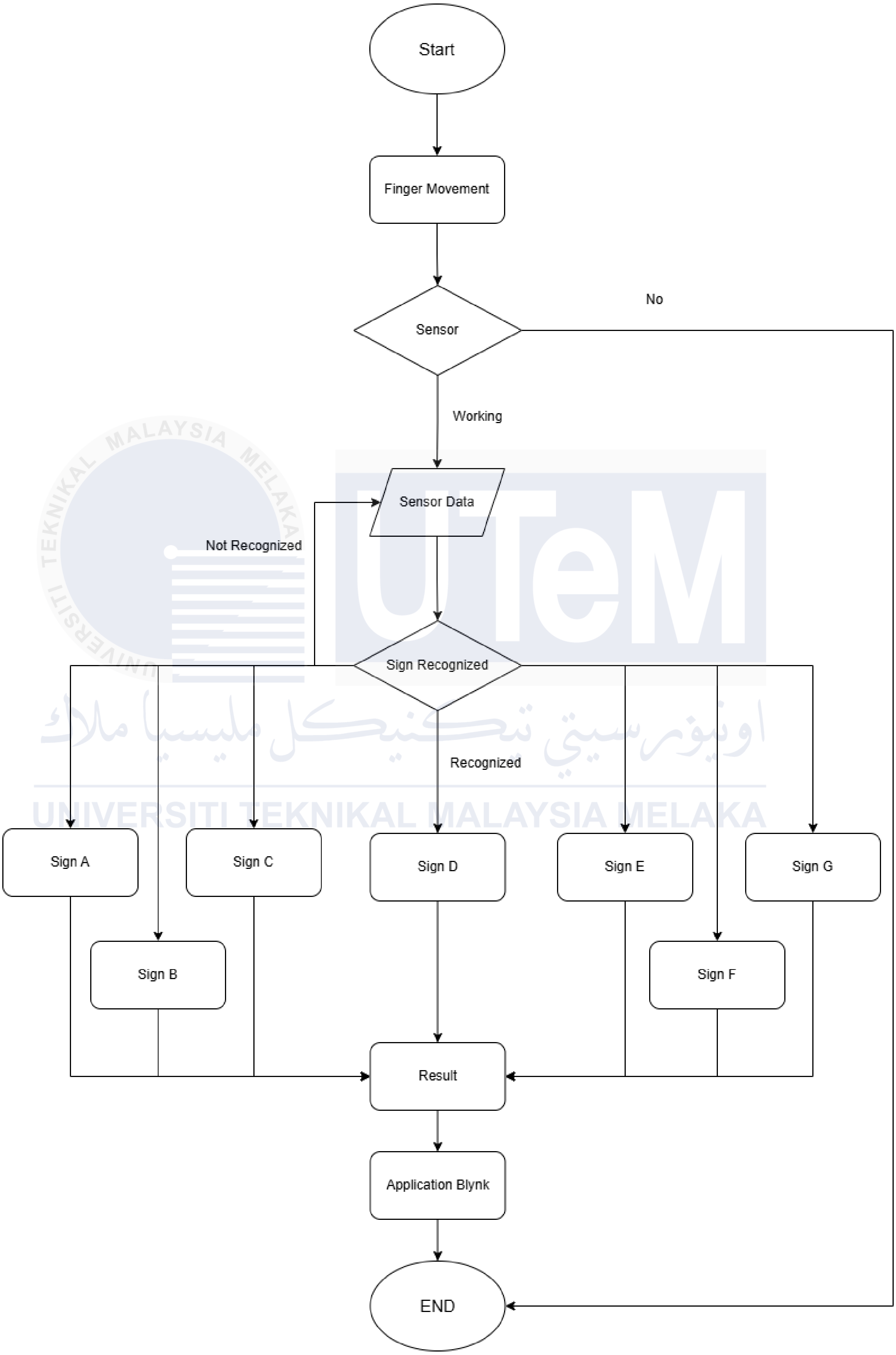

Figure 3.3 Flowchart of Sign Glove project

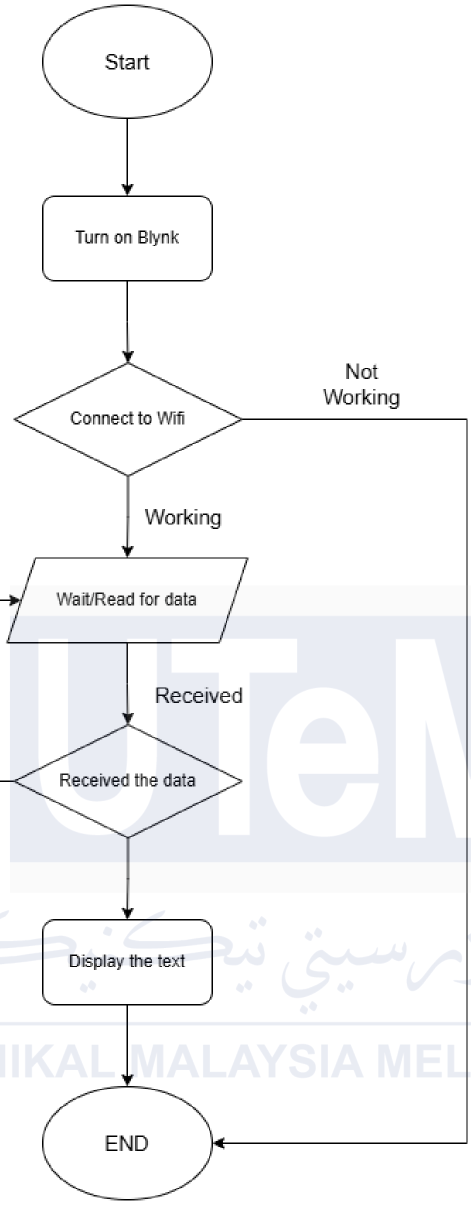

Figure 3.4 Flowchart of Android application

3.4 Component

For the Sign Language Glove to be effective, it will be necessary to incorporate several components that are critical to its normal operation. Some of these unit involve the Arduino Nano which acts as the major controller and the ESP8266 module that handles the internet connection. Five flex sensors are employed throughout the project in order to read every finger flexing and thus properly interpret the gesture. For the sensor circuits, 10K ohm resistors have been incorporated into the circuit design. For power supply, the glove has a 18650 Li-ion battery and a push on/off switch button whereby the glove can work separately without a USB connection. Furthermore, a buck converter is incorporated to translate a higher voltage input to a lesser voltage output so that all components are supplied with the correct voltage levels. Lastly, the project adds a donut board, a large donut shaped board, which is used instead of the breadboard or soldering to map the circuit designs. These combine enable proper functioning of a Sign Language Glove that translates movements of the hands into texts.

3.4.1 Arduino NANO

Based around the ATmega328P or ATmega168 microprocessor, the Arduino Nano is a compact and versatile microcontroller board. Its 14 digital I/O pins (6 of which may offer PWM output) and 8 analog input pins make it ideal for small-scale projects requiring a multitude of inputs and outputs. The Arduino IDE can be used to program the board through a mini-USB interface, and it can accept input voltages between 7 and 12 volts. It is also capable of being powered by a USB cable. The Arduino Nano is better suited for prototype and mobile applications because to its significantly smaller form size when compared to the Arduino Uno. It is intended for use with breadboards.

Figure 3.5 Arduino Nano

3.4.2 NodeMCU ESP8266

The NodeMCU ESP8266 is a low-cost and compact microcontroller board which has Wi-Fi on board hence much suitable for jobs that require wireless connectivity. It enables the devices to connect to the Internet, or a particular local area network and it can send or receive data wirelessly . The board has input/output pins for connection of sensors and to control devices like LEDs or motors. They operate on relatively basic code which can be coded using the Arduino Integrated Development Environment (IDE). The NodeMCU also has a USB port to program the board and to provide power and a voltage regulator that will provide the correct voltage to the ESP8266 chip. This component can be used in the Sign Language Glove project, where it processes data from connected sensors and sends this data wirelessly to another device such as Blynk.

Figure 3.6 NodeMCU ESP8266

3.4.3 Buck Converter

A buck converter is the power converter that has the property of converting high input voltage towards the low output voltage while the efficient energy is retained. It does this automatically by regulating how much power is supplied to the circuit to other components, and protect from power surges which may be too powerful for any component to handle. In Sign Language Glove project, the buck converter guarantees that the needed components, consisting of the microcontroller, sensors, and other electronics, take the required voltage to run safely and efficiently. This makes the glove to function as required without breakdown hitches or power problems, especially when using a powerful battery such as the 18650 Li-ion battery.

Figure 3.7 Buck Converter

3.4.4 Flex Sensors

Flex sensor is an electrical component that detects bending or flexing since its electrical resistance alters with bending. The more one bends, the more likely it will create a resistance level which can be measured to determine the amplitude. In the Sign Language Glove project, flex sensors are attached to each finger to detect their movements. These sensors enable the glove to capture hand gestures, based on the extent of each finger bend. This data is then used to determine specific signs, making it possible for the glove to translate hand movements into text for better communication.

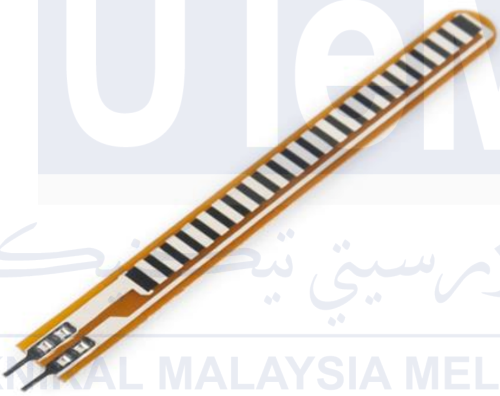

Figure 3.8 Flex Sensor

3.4.5 Resistor 10k Ohm

The component of 10k ohm resistor has a passive electronic component that can control the flow of electricity current in a circuit. Other than that, this device also has a resistance of 10,000 ohms and can be used to control current and voltage levels in electrical circuits. This show, a resistor is vital component in electrical design, which ensures that the circuit can be run safely and successfully while restricting the high current flow. So, any sensor-based project can use this component in voltage divider topologies to connect analog sensors to microcontrollers, which ensures constant and stable readings.

Figure 3.9 Resistor 10k

3.4.6 Push Button Switch ON/OFF

A push button on/off switch is actually a basic mechanism to regulate the flow of power in circuit of an electrical circuit. As simple as that it just switches on or off thus making it easy to control a device without unplugging it. In Sign Language Glove project, this design has provided a convenient way of ON/OFF control through a push button switch. This feature is of great help in extending the battery times when the glove is not in usage, and guarantees the system will only run when required, making the glove more effective and easy to control.

Figure 3.10 Push Button Switch ON/OFF

3.4.7 18650 Li-Ion Battery

The 18650 Li-ion battery is a battery designed for its capacity to be recharged and reused to power the various electronic gadgets. Again, it is portable, light in weight, and can be designed to deliver a high energy which is suitable for portable application. To that end, the 18650 Li-ion battery installed in the Sign Language Glove project powers the entire unit to eliminate the need for a hanging USB cable and an external power source. This makes the glove portable and wage friendly in that one can use it in any location, making the translating of finger movement to text.

Figure 3.11 18650 Li-Ion Battery

3.4.8 Donut Board

A donut board or a perfboard for short is a flat piece of material on which circuits are constructed for use in various applications. It has small contact holes which are surrounded by copper pads and to which components can make fixed connections when soldered. In the Sign Language Glove project, the donut board is used to organize and connect components like the microcontroller, sensors, and resistors in a neat and secure way. This maintains reliability and small size for each circuit and strengthens the structure of the glove while guaranteeing proper interaction of all components.

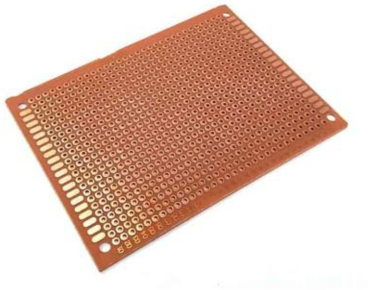

Figure 3.12 Donut Board

3.5 Software

The software development aspect of the Sign Language Glove project is to develop straightforward and optimized algorism for the sign glove. The Arduino Nano is used to interface with the flex sensors, take in the finger movements and transmit this data wirelessly through the NodeMCU ESP8266 module. This data is then to be fed to the Android application which will translate these finger movements into typed text. The system is tested and debugged carefully to make sure everything works correctly and without problems.

3.5.1 Arduino IDE

The Arduino IDE is an adaptable software tool that allows developers to write code in C and C++ following special structure principles intended for Arduino projects. It features a complete software library called the Wiring project, which provides a number of standard input and output capabilities, easing the development process. Next then, the Arduino IDE can connects easily with Arduino and Genuino hardware, which allows developers to upload their code straight to the microcontroller and build a connection with components that has been linked. Also, the Arduino projects can be emulated with Proteus. It is a simulation-based program that can help in testing and debugging the code in a simulation before deploying it to the actual hardware. Finally, a detailed process guarantees that the coding structure is effectively established, tested, and implemented. Figure 3.10 shows Arduino IDE software.

Figure 3.10 Arduino IDE software

3.5.2 Proteus 8

Proteus 8 is a software where at is can simulate, designing and sketch electricity circuit that options to our need. Compared to the other educational tools, this software has multiple orientations with the different microcontrollers it supports. It can help in general for those who are new to learning to build electronic circuits.

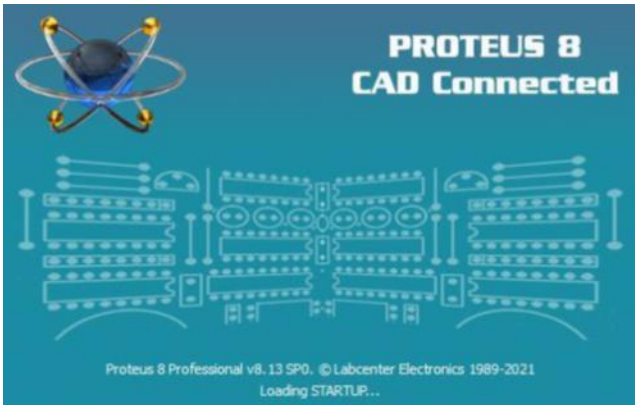

Figure 3.11 Proteus 8 Software

3.5.3 Blynk

Blynk is a solution that shows several steps to make the device Internet-compatible and provides a smartphone application for its control or observation. Blynk is suitable for a sign language glove in a way that it can receive data from the sensors in the glove. When a finger bends to create sign language, the information goes to Blynk where it is interpreted as text and can be displayed on the app to allow better and easier communication.

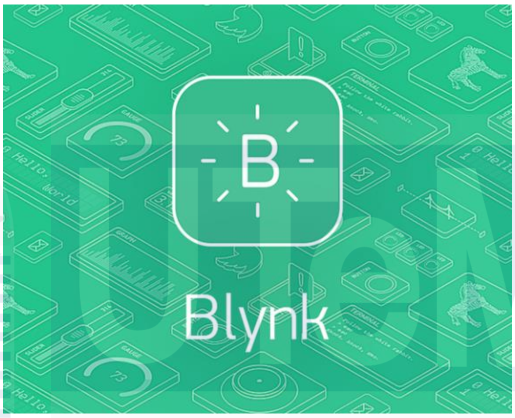

Figure 3.12 Blynk

3.6 Summary

Overall, therefore, the Sign Language Glove involves both hardware and software components in order to operate fs smoothly. The components of the hardware are Arduino Nano, NodeMCU ESP8266, buck converter, flex sensors, 10k Ohm resistor, push switch on/off, 18650 Li-Ion battery and donut board. These together serve as an efficient input signal and processing tool to capture finger movement. On the software side, a basic software like Arduino IDE is used for programming of microcontroller to decide its behaviour with the respective components. Furthermore, data from each of the glove's sensors is obtained using Blynk; when a finger is bent in a particular way to form a sign, the data is processed and translated to text on the Blynk app. This makes it easier to turn sign language into text for better communication.

CHAPTER 4

RESULTS AND DISCUSSIONS

4.1 Introduction

This chapter shall provide the results and discussions of the Sign Glove project. It is as an essential activity in the project trying to assess how far the set objectives were achieved and investigating on achievements. In this section, the data collected will be presented and it will be explain what these results affect and finally how the project was successful in achieving its aim.

4.2 Project Code

This system reads the resistance value from a flex sensor. When fingers bend, the sensor produces an analog signal, which is sent to a microcontroller. The Arduino Nano collects the flex sensor data, processes it to recognize hand gestures, and sends the information to a NodeMCU. The NodeMCU analyses the data and transmits gesture information to the Blynk app, in which it is visualized on a smartphone. This system's source code is in C++, developed using the Arduino Integrated Development Environment (IDE). Arduino Development Environment offers a library, which contains input and output operations, which help in programming of the system.

Figure 4.1 Arduino Integrated Development Environment (IDE)

The coding for this project was done in the C++language. Sometimes, some libraries have to be installed to guarantee the Arduino Nano and NodeMCU ESP8266 components will operate coherently with the code. The following libraries are declared right at the start of the code because they are necessary for the proper running of the project. The downloaded libraries were declared as in figure 4.2 and figure 4.3.

```
#include <SoftwareSerial.h>
#include <ArduinoJson.h>
```

Figure 4.2 Declaration of the library for Arduino Nano

```
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include <SoftwareSerial.h>
#include <ArduinoJson.h>
```

Figure 4.3 Declaration of the library for NodeMCU ESP8266

Figure 4.5 shows the Flex Sensor Configuration, where the analog pins for the flex sensors connected to each finger are defined. Figure 4.5 shows the process of reading Flex Sensor Values, where the system reads the analog values from each flex sensor to measure how much each finger is bent. Figure 4.6 illustrates Determining Hand Signs, where the system checks if the flex sensor values fall within a specific range for a particular gesture. Similar conditions are applied to recognize other gestures.

```
// Define the analog pins for the flex sensors
const int pinkyPin = A0;
const int ringPin = A1;
const int middlePin = A2;
const int indexPin = A3;
const int thumbPin = A4;
```

Figure 4.4 Flex Sensor Configuration

```
// Read the analog values from the flex sensors
int pinkyValue = analogRead(pinkyPin);
int ringValue = analogRead(ringPin);
int middleValue = analogRead(middlePin);
int indexValue = analogRead(indexPin);
int thumbValue = analogRead(thumbPin);
```

Figure 4.5 Reading Flex Sensor Values

```
// Determine the sign based on the values
if (pinkyValue > 700 && pinkyValue <= 800 &&
    ringValue > 700 && ringValue <= 800 &&
    middleValue > 800 && middleValue <= 900 &&
    indexValue > 700 && indexValue <= 800 &&
    thumbValue > 700 && thumbValue <= 800) {
    Serial.println("Sign: Hi, Hello");
    sign1 = 1;
}
```

Figure 4.6 Determining Hand Signs

4.3 Result text

This sign language glove in the project has the capability to identify 7 signs. The coding uses if-else conditions to identify the gestures. The Arduino Nano collects data from the flex sensors, processes it to detect finger movements, and sends the information to the NodeMCU. The NodeMCU next analyzes this data and relays the gesture picked to the Blynk app, where the gesture appears on a Smartphone.

The first condition, if the values of the flex sensor are with the specified range then the message displayed on Blynk app will be "Hi Hello" which corresponds to sign 1. This is evident from the figure 4.7 below which demonstrate that the obtained meet values satisfies the meet range and the figure 4.8 below indicates that "Hi, Hello" appears on the Blynk.

```
if (pinkyValue > 700 && pinkyValue <= 800 &&
    ringValue > 700 && ringValue <= 800 &&
    middleValue > 800 && middleValue <= 900 &&
    indexValue > 700 && indexValue <= 800 &&
    thumbValue > 700 && thumbValue <= 800) {
    Serial.println("Sign: Hi, Hello");
    sign1 = 1;
```

Figure 4.7 Flex Sensor values match a specific range for a gesture

Figure 4.8 Blynk app display "Hi, Hello"

The second condition, if the values of the flex sensor are with the specified range then the message displayed on Blynk app will be "Peace" which corresponds to sign 2. This is evident from the figure 4.9 below which demonstrate that the obtained meet values satisfies the meet range and the figure 4.10 below indicates that “Peace” appears on the Blynk.

```
else if (pinkyValue > 800 && pinkyValue <= 900 &&
         ringValue > 800 && ringValue <= 900 &&
         middleValue > 800 && middleValue <= 900 &&
         indexValue > 700 && indexValue <= 800 &&
         thumbValue > 800 && thumbValue <= 900) {
    Serial.println("Sign: Peace");
    sign2 = 2;
```

Figure 4.9 Flex Sensor values match a specific range for a gesture

Figure 4.11 Blynk app display “Peace”

The third condition, if the values of the flex sensor are with the specified range then the message displayed on Blynk app will be “Love You” which corresponds to sign 3. This is evident from the figure 4.11 below which demonstrate that the obtained meet values satisfies the meet range and the figure 4.12 below indicates that “Love You” appears on the Blynk.

```
else if (pinkyValue > 700 && pinkyValue <= 800 &&
         ringValue > 800 && ringValue <= 900 &&
         middleValue > 800 && middleValue <= 900 &&
         indexValue > 700 && indexValue <= 800 &&
         thumbValue > 700 && thumbValue <= 800) {
    Serial.println("Sign: Love You");
    sign3 = 3;
```

Figure 4.11 Flex Sensor values match a specific range for a gesture

Figure 4.12 Blynk app display “Love You”

The forth condition, if the values of the flex sensor are with the specified range then the message displayed on Blynk app will be “Rock On” which corresponds to sign 4. This is evident from the figure 4.13 below which demonstrate that the obtained meet values satisfies the meet range and the figure 4.14 below indicates that “Rock On” appears on the Blynk.

```
else if (pinkyValue > 700 && pinkyValue <= 800 &&
         ringValue > 800 && ringValue <= 900 &&
         middleValue > 800 && middleValue <= 850 &&
         indexValue > 700 && indexValue <= 800 &&
         thumbValue > 800 && thumbValue <= 900) {
    Serial.println("Sign: Rock On");
    sign4 = 4;
```

Figure 4.13 Flex Sensor values match a specific range for a gesture

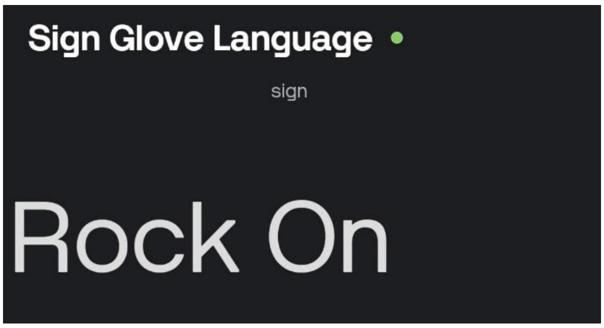

Figure 4.14 Blynk app display “Rock On”

The fifth condition, if the values of the flex sensor are with the specified range then the message displayed on Blynk app will be “Call Me” which corresponds to sign 5. This is evident from the figure 4.15 below which demonstrate that the obtained meet values satisfies the meet range and the figure 4.16 below indicates that “Call Me” appears on the Blynk.

```
else if (pinkyValue > 700 && pinkyValue <= 800 &&
         ringValue > 800 && ringValue <= 900 &&
         middleValue > 800 && middleValue <= 900 &&
         indexValue > 800 && indexValue <= 900 &&
         thumbValue > 700 && thumbValue <= 800) {
    Serial.println("Sign: Call Me");
    sign5 = 5;
```

Figure 4.15 Flex Sensor values match a specific range for a gesture

Figure 4.16 Blynk app display “Call Me”

The sixth condition, if the values of the flex sensor are with the specified range then the message displayed on Blynk app will be “Thumbs Up” which corresponds to sign 6. This is evident from the figure 4.17 below which demonstrate that the obtained meet values satisfies the meet range and the figure 4.18 below indicates that “Thumbs Up” appears on the Blynk.

```
else if (pinkyValue > 800 && pinkyValue <= 900 &&
         ringValue > 800 && ringValue <= 900 &&
         middleValue > 800 && middleValue <= 900 &&
         indexValue > 800 && indexValue <= 900 &&
         thumbValue > 700 && thumbValue <= 800) {
    Serial.println("Sign: Thumbs Up");
    sign6 = 6;
```

Figure 4.17 Flex Sensor values match a specific range for a gesture

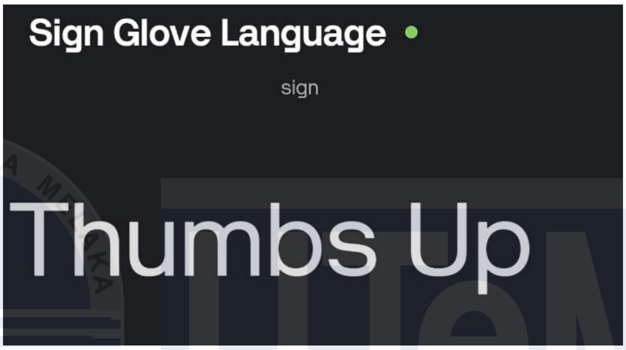

Figure 4.18 Blynk app display "Thumbs Up"

The seventh condition, if the values of the flex sensor are with the specified range then the message displayed on Blynk app will be "Loser" which corresponds to sign 7. This is evident from the figure 4.19 below which demonstrate that the obtained meet values satisfies the meet range and the figure 4.20 below indicates that "Loser" appears on the Blynk.

```
else if (pinkyValue > 800 && pinkyValue <= 900 &&
         ringValue > 800 && ringValue <= 900 &&
         middleValue > 800 && middleValue <= 900 &&
         indexValue > 700 && indexValue <= 800 &&
         thumbValue > 700 && thumbValue <= 800) {
    Serial.println("Sign: Loser");
    sign7 = 7;
```

Figure 4.19 Flex Sensor values match a specific range for a gesture

Figure 4.20 Blynk app display “Loser”

4.4 Finger bending activity

The flex sensor detects human's finger flexing movement by pinned the flex sensor on back of the glove. The figure 4.21 show Finger Bending Angles Over Time activity, which shows how each finger bends during a gesture, helping to recognize specific sign language gestures. At the start time, which is 0s, all fingers are straight. As the gesture progresses from time to time the fingers will bend at different rates and some fingers bending more than others. When at the peak, the fingers reach their maximum bending will completing the gesture. As the gesture ends, the fingers straighten back to their original position. These bending patterns, show their timing and angles can help to recognize the specific gesture being made, which is important for sign language translation systems to accurately convert the gesture into text.

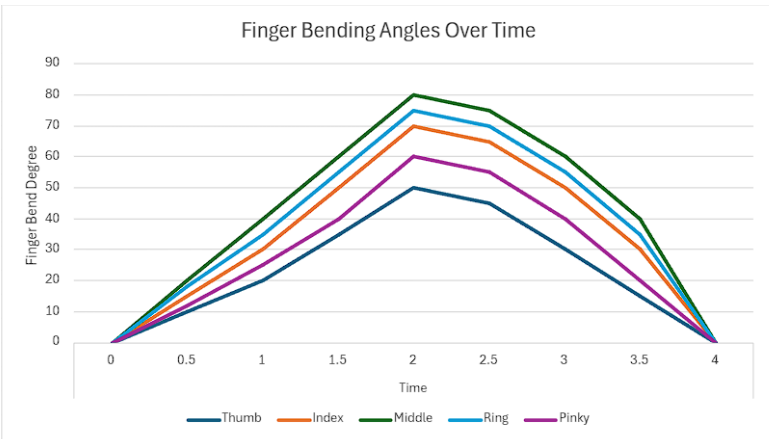

Figure 4.21 Finger Bending Angles Over Time activity

4.5 Flex Sensor Circuit Operation and Voltage analysis

Flex sensor circuit is employed to determine how much a finger may be bent. It uses a flex sensor and a fixed resistor, combined in what is called a voltage divider circuit. When the flex sensor is bent, the amount of resistance in the circuit alters resulting in the voltage level. This voltage is then read by a microcontroller to identify the amount at which the finger has bend. Figure 4.22 shows the schematic diagram of the voltage divider of the flex sensor.

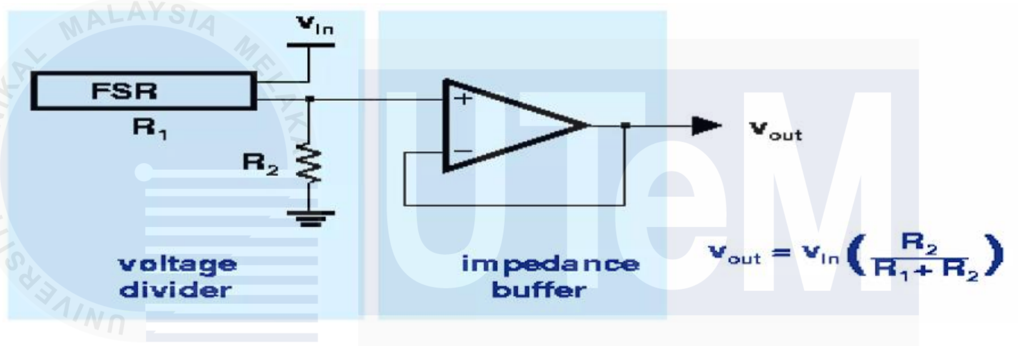

Figure 4.22 Basic Flex Sensor Circuit

To calculate the voltage output, can used this formula $Vout = Vin \times \frac{R1}{R1+R2}$, let assume Vin is 5V, flex sensor has resistance of 2kΩ (R1) when bend 90 degree. The fixed resistor is 10 kΩ (R2), and the power supply is 5V. The value of Vout is 0.83V. The table 4.1 shows the flex sensor output voltage.

Table 4.1 Flex Sensor Output Voltage

Bending Angle	Resistance of Flex Sensor	Output Voltage
0°	10 kΩ	2.5V
10°	9.5 kΩ	2.4V
20°	9 kΩ	2.3V
30°	8.5 kΩ	2.2V
40°	8 kΩ	2.1V
50°	7.5 kΩ	2.0V
60°	7 kΩ	1.8V
70°	5.5 kΩ	1.5V
80°	3 kΩ	1.0V
90°	2 kΩ	0.83V

4.6 Hardware Actual Design

The figure 4.24 shows the actual design of the sign language glove. This glove has flex sensors attached on each finger. These sensors have the capacity of assessing the bending movement of the fingers. When a specific finger movement matches a predefined sign, the sensors send the data to the Arduino Nano. Using the Arduino Nano, the data produced by the sensors is accumulated and analyzed to determine the hand gesture. It then sends this information to the ESP8266. The ESP8266 then captures the data, analyzed it and transmits the detected gesture to the Blynk application. The idea is implemented in the smartphone by showing the result of the gesture as the text, in order to let the user, know what sign language gesture is currently being done. This specific design allows a person to type using their hand gestures in real time in hopes of making communication much simpler.

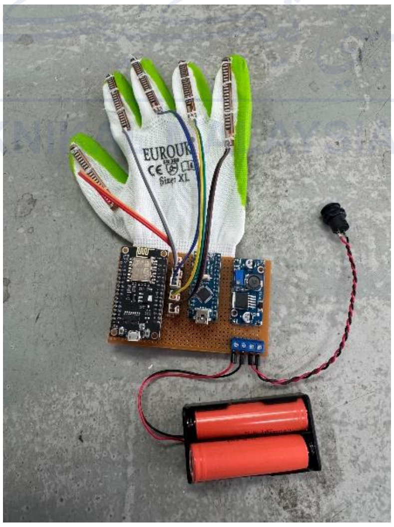

Figure 4.23 Actual Design of Sign Language Glove

4.7 Summary

In this chapter, the details of Sign Glove project are discussed in terms of project code, finger bending activity, and the functional understanding of flex sensor circuit. I also investigated the voltage changes and demonstrated the hardware design that completes the Sign Glove system. I also discovered how to go about the coding of the project using Arduino IDE Stretch sensors and how to type a sign language to display content. I also contributed to the finger bending activity to see how the flex Sensors measured the fingers' motion. With voltage analysis, I got to see the effects of shift in finger positions as well as how the system functioned. Besides, I integrated the Sign Glove hardware since I also developed it, making connections to guarantee that all the parts were working correctly. During the work on the project I have developed the practical experience in the circuit design, coding of the system and testing. It took some time before I was able to define the process, but it let me step by step make progress and gain new knowledge.

CHAPTER 5

CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

In the nutshell, the Sign Glove project has been able to design and develop a wearable glove whose function is to translate finger movement gestures into text. With the help of sensors, Arduino Nano, ESP8266, and the Blynk app, the glove offers people with hearing impairments an efficient method of interacting with others. This project demonstrates how technology can be utilized to help the society overcome barriers and the limitations to effective communication. The use of this innovation can greatly assist persons with hearing impediments as they take part in daily conversations with the rest of society without feeling left out. The project reveals that when equipped with proper instruments it is quite possible to improve the quality of individuals' lives who suffer from communication difficulties.

5.2 Future Works

For future improvement, the Sign Glove could be enhanced as follows:

1. Recognizing More Gestures: The project can be extended to recognize a variety of sign language gestures, thus better to serve other sign language systems.
2. Adding Voice Recognition: The system could be expanded to include voice command or output so the glove would be able to say things, or take spoken instructions from the user.

3. Enhancing Comfort and Design: Subsequent versions of the glove would be more comfortable and even wearable for extended periods.

5.3 Project Commercialize

The Sign Glove project can be a product which would seem helpful in daily usage for deaf and hard of hearing people. In schools, hospitals and workplaces, this can be employed to simplify communication. It can also be presented as an educational product for learning sign language; it can also be adapted to a particular sign language. Adding features like connecting to laptop or voice output can make it even more useful and attract more users. Working with governments, charities, or disability organizations can help reach more people. A subscription for updates or extra features can also make it more valuable. This makes the Sign Glove as a product which aids inclusion, and address and solve actual functional living requirements among users.

REFERENCES

[1]N. Harish and S. Poonguzhali, "Design and development of hand gesture recognition system for speech impaired people," *2015 Int. Conf. Ind. Instrum. Control. ICIC 2015*, no. Icic, pp. 1129–1133, 2015, doi: 10.1109/IIC.2015.7150917.

[2]K. P. Kour and L. Mathew, "Sign Language Recognition Using Image Processing," *Int. J. Adv. Res. Comput. Sci. Softw. Eng.*, vol. 7, no. 8, p. 142, 2017, doi: 10.23956/ijarcsse.v7i8.41.

[3]R. Raen, S. M. R. Islam, and R. Islam, "A Smart Hand Glove that Converts Gesture into Text & Speech to Assist the Handicapped-(Handtalk)," *2022 Int. Conf. Adv. Electr. Electron. Eng. ICAEEE 2022*, no. February, 2022, doi: 10.1109/ICAEEE54957.2022.9836392.

[4]S. K. M, "Microcontroller Based Sign Language Glove," vol. 1, no. 4, pp. 31–33, 2013.

[5]O. Chowdhury, M. S. Rahman Hridoy, F. Amin, S. Arif, and H. Begum, "Smart Gloves for People with Speech Disability," *2023 26th Int. Conf. Comput. Inf. Technol. ICCIT 2023*, no. March, pp. 1–5, 2023, doi: 10.1109/ICCIT60459.2023.10441257.

[6]F. Pezzuoli, D. Corona, M. L. Corradini, and A. Cristofaro, "Development of a Wearable Device for Sign Language Translation," *Springer Proc. Adv. Robot.*, vol. 7, pp. 115–126, 2019, doi: 10.1007/978-3-319-89327-3_9.

[7]P. Verma, S. L. Shimi, and S. Chatterji, "Design of Smart Gloves," *Int. J. Eng. Res. Technol.*, vol. 3, no. 11, pp. 210–214, 2014.

[8]A. Rajamohan, R. Hemavathy, M. Dhanalakshmi, S. Sivasubramania, N. College, and T. Nadu, "Deaf-Mute Communication Interpreter," vol. 2, no. 5, pp. 336–341, 2013.

[9]K. Gunasekaran and R. Manikandan, "Sign language to speech translation system using PIC microcontroller," *Int. J. Eng. Technol.*, vol. 5, no. 2, pp. 1024–1028, 2013.

[10] V. A. P, S. B. P, K. P. sudheesh C, and M. T. S, "Smart Glove Using Arduino With Sign Language Recognition System," *Int. J. Adv. Res. Innov. Ideas Educ.*, vol. 6, no. 2, pp. 1691–1701, 2020, [Online]. Available: www.ijariie.com

[11] A. J. Abougarair, "Smart glove for sign language translation," *Int. Robot. Autom. J.*, vol. 8, no. 3, pp. 109–117, 2022, doi: 10.15406/iratj.2022.08.00253.

[12] R. Ambar, S. Salim, M. H. A. Wahab, M. M. A. Jamil, and T. C. Phing, "Development of a Wearable Sensor Glove for Real-Time Sign Language Translation," *Ann. Emerg. Technol. Comput.*, vol. 7, no. 5, pp. 25–38, 2023, doi: 10.33166/AETIC.2023.05.003.

[13] U. Farooq, M. S. M. Rahim, N. Sabir, A. Hussain, and A. Abid, *Advances in machine translation for sign language: approaches, limitations, and challenges*, vol. 33, no. 21. Springer London, 2021. doi: 10.1007/s00521-021-06079-3.

[14] H. Orovwode, I. D. Oduntan, and J. Abubakar, "Development of a Sign Language Recognition System Using Machine Learning," *6th Int. Conf. Artif. Intell. Big Data, Comput. Data Commun. Syst. icABCD 2023 - Proc.*, pp. 1–8, 2023, doi: 10.1109/icABCD59051.2023.10220456.

[15] R. Anisha, A. S. Nair, A. Shreya, C. Tharun, and K. N. Neethu, "Design and Implementation of Smart Gloves for the Specially Privileged," *Proc. 2020 3rd Int. Conf. Adv. Electron. Comput. Commun. ICAECC 2020*, no. February, pp. 4–9, 2020, doi: 10.1109/ICAECC50550.2020.9339475.

[16] M. Zakariah, Y. A. Alotaibi, D. Koundal, Y. Guo, and M. Mamun Elahi, "Sign Language Recognition for Arabic Alphabets Using Transfer Learning Technique," *Comput. Intell. Neurosci.*, vol. 2022, 2022, doi: 10.1155/2022/4567989.

[17] P. Jayanthi, P. R. K. Sathia Bhama, K. Swetha, and S. A. Subash, "Real Time Static and Dynamic Sign Language Recognition using Deep Learning," *J. Sci. Ind. Res. (India).*, vol. 81, no. 11, pp. 1186–1194, 2022, doi: 10.56042/jsir.v81i11.52657.

[18] S. Pramada, "Intelligent Sign Language Recognition Using Image Processing," *IOSR J. Eng.*, vol. 03, no. 02, pp. 45–51, 2013, doi: 10.9790/3021-03224551.

[19] P. D. Rosero-Montalvo *et al.*, "Sign Language Recognition Based on Intelligent Glove Using Machine Learning Techniques," *2018 IEEE 3rd Ecuador Tech. Chapters Meet. ETCM 2018*, no. December, 2018, doi: 10.1109/ETCM.2018.8580268.

[20] M. A. Ahmed, B. B. Zaidan, A. A. Zaidan, M. M. Salih, and M. M. Bin Lakulu, "A review on systems-based sensory gloves for sign language recognition state of the art between 2007 and 2017," *Sensors (Switzerland)*, vol. 18, no. 7, 2018, doi: 10.3390/s18072208.

APPENDICES

No	Week / Activity	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	BDP 2 BRIEFING														
2	BUY COMPONENT														
3	TESTING COMPONENT														
4	WRITE CODE														
5	DO PROTOTYPE														
6	MEETING WITH SUPERVISOR														
7	LOGBOOK SUBMISSION														
8	REPORT SUBMISSION														
9	PRESENTATION														

Code for Arduino Nano and NodeMCU ESP8266

Arduino Nano:

```
#include <SoftwareSerial.h>
#include <ArduinoJson.h>

// Define the analog pins for the flex sensors
const int pinkyPin = A0;
const int ringPin = A1;
const int middlePin = A2;
const int indexPin = A3;
const int thumbPin = A4;

// Initialize SoftwareSerial for NodeMCU communication
SoftwareSerial NodeMCU(2, 3);

void setup() {
  // Initialize Serial Monitor
  Serial.begin(9600);
  NodeMCU.begin(9600);
}

void loop() {
  // Read the analog values from the flex sensors
  int pinkyValue = analogRead(pinkyPin);
  int ringValue = analogRead(ringPin);
  int middleValue = analogRead(middlePin);
  int indexValue = analogRead(indexPin);
  int thumbValue = analogRead(thumbPin);

  // Print the values to the Serial Monitor
  Serial.print("Pinky: ");
  Serial.print(pinkyValue);
  Serial.print(", Ring: ");
  Serial.print(ringValue);
  Serial.print(", Middle: ");
  Serial.print(middleValue);
  Serial.print(", Index: ");
  Serial.print(indexValue);
  Serial.print(", Thumb: ");
  Serial.println(thumbValue);

  // Initialize sign variables
  int sign1 = 0;
```

```
int sign2 = 0;
int sign3 = 0;
int sign4 = 0;
int sign5 = 0;
int sign6 = 0;
int sign7 = 0;

// Determine the sign based on the values
if (pinkyValue > 700 && pinkyValue <= 800 &&
   ringValue > 700 && ringValue <= 800 &&
   middleValue > 800 && middleValue <= 900 &&
   indexValue > 700 && indexValue <= 800 &&
   thumbValue > 700 && thumbValue <= 800) {
   Serial.println("Sign: Hi, Hello");
   sign1 = 1;
}
else if (pinkyValue > 800 && pinkyValue <= 900 &&
       ringValue > 800 && ringValue <= 900 &&
       middleValue > 800 && middleValue <= 900 &&
       indexValue > 700 && indexValue <= 800 &&
       thumbValue > 800 && thumbValue <= 900) {
   Serial.println("Sign: Peace");
   sign2 = 2;
}
else if (pinkyValue > 700 && pinkyValue <= 800 &&
       ringValue > 800 && ringValue <= 900 &&
       middleValue > 800 && middleValue <= 900 &&
       indexValue > 700 && indexValue <= 800 &&
       thumbValue > 700 && thumbValue <= 800) {
   Serial.println("Sign: Love You");
   sign3 = 3;
}
else if (pinkyValue > 700 && pinkyValue <= 800 &&
       ringValue > 800 && ringValue <= 900 &&
       middleValue > 800 && middleValue <= 850 &&
       indexValue > 700 && indexValue <= 800 &&
       thumbValue > 800 && thumbValue <= 900) {
   Serial.println("Sign: Rock On");
   sign4 = 4;
}
else if (pinkyValue > 700 && pinkyValue <= 800 &&
       ringValue > 800 && ringValue <= 900 &&
       middleValue > 800 && middleValue <= 900 &&
       indexValue > 800 && indexValue <= 900 &&
       thumbValue > 700 && thumbValue <= 800) {
   Serial.println("Sign: Call Me");
   sign5 = 5;
}
else if (pinkyValue > 800 && pinkyValue <= 900 &&
       ringValue > 800 && ringValue <= 900 &&
```

```
          middleValue > 800 && middleValue <= 900 &&
          indexValue > 800 && indexValue <= 900 &&
          thumbValue > 700 && thumbValue <= 800) {
    Serial.println("Sign: Thumbs Up");
    sign6 = 6;
  }
  else if (pinkyValue > 800 && pinkyValue <= 900 &&
          ringValue > 800 && ringValue <= 900 &&
          middleValue > 800 && middleValue <= 900 &&
          indexValue > 700 && indexValue <= 800 &&
          thumbValue > 700 && thumbValue <= 800) {
    Serial.println("Sign: Loser");
    sign7 = 7;
  }
  else {
    Serial.println("Sign: Unknown");
  }

  // Create JSON buffer and object
  StaticJsonBuffer<200> jsonBuffer;
  JsonObject& data = jsonBuffer.createObject();

  // Add data to JSON object
  data["sign1"] = sign1;
  data["sign2"] = sign2;
  data["sign3"] = sign3;
  data["sign4"] = sign4;
  data["sign5"] = sign5;
  data["sign6"] = sign6;
  data["sign7"] = sign7;

  // Send the JSON data to NodeMCU
  data.printTo(NodeMCU);
  NodeMCU.println();

  // Clear the JSON buffer
  jsonBuffer.clear();

  // Small delay to prevent flooding
  delay(100);
}
```

NodeMCU ESP8266:

```
#define BLYNK_TEMPLATE_ID "TMPL6MX47PC7O"
#define BLYNK_TEMPLATE_NAME "Sign Glove Language"
#define BLYNK_AUTH_TOKEN "-ksV0m6c-jUUtDxroZ_pmiy8lr2jt39w"
#define BLYNK_PRINT Serial
```

```
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include <SoftwareSerial.h>
#include <ArduinoJson.h>

// Blynk authentication token
char auth[] = BLYNK_AUTH_TOKEN;

// WiFi credentials
// char ssid[] = "studiogambar-2.4G@unifi";
// char pass[] = "maanstraat";

char ssid[] = "RumahKami";
char pass[] = "Selamat000";

// Initialize Blynk timer
BlynkTimer timer;

// Initialize SoftwareSerial for Arduino communication (D5 = Rx & D6 = Tx)
SoftwareSerial nodemcu(D5, D6);

// Buffer for incoming data
const int BUFFER_SIZE = 256;
char inputBuffer[BUFFER_SIZE];
int bufferIndex = 0;

void processJson(char* json) {
  // Create JSON buffer
  StaticJsonBuffer<200> jsonBuffer;

  // Parse JSON object
  JsonObject& data = jsonBuffer.parseObject(json);

  // Test if parsing succeeds
  if (!data.success()) {
    Serial.println(F("parseObject() failed"));
    return;
  }

  // Extract values
  int sign1 = data["sign1"] | 0;  // Use | operator for default value
  int sign2 = data["sign2"] | 0;
  int sign3 = data["sign3"] | 0;
  int sign4 = data["sign4"] | 0;
  int sign5 = data["sign5"] | 0;
  int sign6 = data["sign6"] | 0;
  int sign7 = data["sign7"] | 0;

  // Print received values for debugging
```

```
  Serial.print(F("Received values: "));
  Serial.print(sign1); Serial.print(" ");
  Serial.print(sign2); Serial.print(" ");
  Serial.print(sign3); Serial.print(" ");
  Serial.print(sign4); Serial.print(" ");
  Serial.print(sign5); Serial.print(" ");
  Serial.print(sign6); Serial.print(" ");
  Serial.println(sign7);

  // Update Blynk
  if (sign1 == 1) {
    Blynk.virtualWrite(V0, "Hi, Hello");
  }
  else if (sign2 == 2) {
    Blynk.virtualWrite(V0, "Peace");
  }
  else if (sign3 == 3) {
    Blynk.virtualWrite(V0, "Love You");
  }
  else if (sign4 == 4) {
    Blynk.virtualWrite(V0, "Rock On");
  }
  else if (sign5 == 5) {
    Blynk.virtualWrite(V0, "Call Me");
  }
  else if (sign6 == 6) {
    Blynk.virtualWrite(V0, "Thumbs Up");
  }
  else if (sign7 == 7) {
    Blynk.virtualWrite(V0, "Loser");
  }
  else {
    Blynk.virtualWrite(V0, "No Sign Detected");
  }
}

void send_data() {
  // Read data from Arduino while available
  while (nodemcu.available() > 0) {
    char c = nodemcu.read();

    // Add character to buffer if there's space
    if (bufferIndex < BUFFER_SIZE - 1) {
      inputBuffer[bufferIndex] = c;
      bufferIndex++;

      // If newline received, process the message
      if (c == '\n' || c == '\r') {
        // Null terminate the string
        inputBuffer[bufferIndex - 1] = '\0';
```

```
            // Only process if buffer contains data
            if (bufferIndex > 1) {
               processJson(inputBuffer);
            }

            // Reset buffer
            bufferIndex = 0;
         }
      }
      else {
         // Buffer overflow, reset buffer
         bufferIndex = 0;
      }
   }
}

void setup() {
   // Initialize Serial Monitor
   Serial.begin(9600);

   // Initialize NodeMCU-Arduino communication
   nodemcu.begin(9600);

   // Connect to Blynk
   Blynk.begin(auth, ssid, pass);

   // Set timer interval for data processing
   timer.setInterval(100L, send_data);

   Serial.println(F("System initialized"));
}

BLYNK_CONNECTED() {
   Serial.println(F("Blynk connected"));
}

void loop() {
   if (Blynk.connected()) {
      Blynk.run();
      timer.run();
   } else {
      Serial.println(F("Blynk not connected"));
      delay(1000);
   }
}
```