VIMAL JYOTHI
ENGINEERING COLLEGE

APPROVED BY AICTE AND AFFILIATED TO APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

Jyothi Nagar, Chemperi P.O, Kannur Dt. Kerala - 670632 | www.vjec.ac.in

PROJECT WORK

DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

Signature Not Verified

Principal VJEC

Contents

Signature Not Verified

Principal.VJEC

VIMAL JYOTHI ENGINEERING COLLEGE
DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING
LIST OF UG STUDENTS UNDERTAKING PROJECT WORK

SL NO	REGISTER NUMBER	NAME
1	VML19EE001	ABDUL JABBAR YOUNUS
2	VML19EE002	ABHILASH JOSEPH
3	VML19EE003	ABHINAV V
4	VML19EE004	AJIN MATHEW JOSEPH
5	VML19EE005	AJITH SAJI
6	VML19EE006	AKHIL GEORGE
7	VML19EE007	ALBIN SAJI
8	VML19EE008	ALEENA JAISON
9	VML19EE009	ALEN VARGHESE
10	VML19EE011	AMRITHA P
11	VML19EE012	ASHLYN WILSON SASTHAMPADAVIL
12	VML19EE013	ASWANTH RAMESHAN
13	VML19EE014	ASWIN K
14	VML19EE015	ASWIN SURESH M S
15	VML19EE016	DILNA MARIA SHIBU
16	VML19EE018	DWITHI SHIVAKUMAR
17	VML19EE019	ELTTIN JOY
18	VML19EE020	GOKUL ARIYIL
19	VML19EE021	HRITHWIK SREEJITH
20	VML19EE022	JOEL M JACOB
21	VML19EE023	JOHN TOMY
22	VML19EE024	KIRAN JOSEPH
23	VML19EE025	MUHAMMED HANNAN FAZAL
24	VML19EE026	NAKUL GANESH
25	VML19EE027	NOYAL JOSE
26	VML19EE029	P ROMA ULLAS
27	VML19EE028	PRANAV TV
28	VML19EE030	RENITHA RAMAKRISHNAN

Signature Not Verified

Principal.VJEC

29	VML19EE031	SAYOOJ DEVAN M.B
30	VML19EE032	SRADHA ALEX
31	VML19EE033	VAISHALI PRABHAKARAN
32	VML19EE034	VISHNU SREEKUMAR K M
33	VML19EE035	ZIJAH TK

Signature Not Verified

Principal.VJEC

A SMART GLOVE THAT TRANSLATES SIGN LANGUAGE INTO TEXT AND SPEECH

PROJECT PHASE-II REPORT

Submitted by

DILNA MARIA SHIBU(VML19EE016)

DWITHI SHIVAKUMAR(VML19EE018)

VAISHALI PRABHAKARAN(VML19EE033)

In the partial fulfilment for the award of the Degree of

BACHELOR OF TECHNOLOGY IN

ELECTRICAL AND ELECTRONICS ENGINEERING

VIMAL JYOTHI ENGINEERING COLLEGE

A P J ABDUL KALAM TECHNOLOGICAL UNIVERSITY

MAY 2023

ACKNOWLEDGEMENT

All glory and honors of God, the Almighty without his grace our project would not have seen the light of success. Motivation is the driving source behind every successful venture. With regard to the completion of this project, we wish to acknowledge the following people who are the source of motivation behind it.

We express our thanks to **Dr. BENNY JOSEPH**, Principal, **Mrs. LALY JAMES**, Head of the Department, Tutors and all the other staffs of Department of Electrical and Electronics Engineering, Vimal Jyothi Engineering College, Chemperi, for their valuable suggestion and co-operation.

We wish to extend our heartfelt thanks to our guide **Dr. R. SENTHILKUMAR**, Professor in the department of EEE, for his guidance, support and cooperation during the course of this project.

Thanks to **Mrs. TINTU GEORGE**, Associate Professor, who have been to co-ordinate the project. Last but not the least we would thank our parents and friend swhose encouragement and advice helped us a lot during the period of this work.

DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that the project report entitled **"A SMART GLOVE THAT TRANSLATES SIGN LANGUAGE INTO TEXT AND SPEECH"** *is a bonafide record of the EED 416 Project Phase II Report preliminary done by* ***DILNA MARIA SHIBU, DWITHI SHIVAKUMAR and VAISHALI PRABHAKARAN*** *under our guidance towards the partial fulfilment of the requirements for the award of the Degree of Bachelor of technology in Electrical& Electronics Engineering of the APJ Abdul Kalam Technological University through Vimal Jyothi Engineering College, Chemperi, Kannur.*

PROJECT GUIDE	PROJECT COORDINATORS	HEAD OF THE DEPARTMENT
Dr. R Senthilkumar	**Mrs. Tintu George**	**Mrs. Laly James**
Professor	Associate Professor	Professor
Department of EEE	Department of EEE	Department of EEE
VJEC	VJEC	VJEC

DEPARTMENT OF ELECTRICAL & ELECTRONICS ENGG. V. J.E.C CHEMPERI Date:............

Signature Not Verified

Principal.VJEC

CONTENTS

ABSTRACT

Speech is the easiest way for communication in the world. It becomes difficult for speech impaired people to communicate with normal people as they use sign language for communication. When a speech-impaired person communicates with normal person, the communication gap between speech impaired and normal masses is too much to fill. The gesture recognition can be done in two ways, Image processing based and sensor-based.

The Objective of the project is to design a smart glove for sign language translation that helps an easy way of communication for speech impaired or hearing-impaired people.

In this project, glove need to be equipped with sensors such as Flex sensor and Accelerometer which sense different sign language gestures. Flex sensors are placed on fingers which measure the bending of fingers according to a gesture made. An accelerometer is placed on the palm which measures the location of the hand in X, Y, Z axes. The sensed data from sensors is sent to Arduino UNO board for further processing and transfer data to an android phone via Bluetooth module. The data we get will be in the form of text. This text data is then converted into speech.

LIST OF FIGURES

ABBREVIATIONS

Abbreviation	Expansion
ASL	American Sign Language
MPU	Memory Protection Unit
DMP	Digital Motion Processor
IDE	Integrated Development Environment
USB	Universal Serial Bus
LED	Light Emitting Diode
EDA	Electronic Design Automation
PCB	Printed Circuit Board
IC	Integrated Circuit
SPI	Serial Peripheral Interface
PWM	Pulse Width Modulation
GSM	Global System for Mobile communication
GPS	Global Positioning System

LIST OF TABLES

CHAPTER 1

INTRODUCTION

In the present world it is very complicated for the deaf & dumb people to talk with the ordinary people as impaired people lacks the amenities which a normal person should own. It actually becomes the same problem of two persons which knows two different languages, no one of them knows any common language so its becomes a problem to talk with each other and so they require a translator physically which may not be always convenient to arrange and this same kind of problem occurs in between the Normal Person and the Deaf person or the Normal Person and the Dumb person.

Sign language is one of the most natural and expressive ways of communication for the hearing impaired and dumb people. According to the World Federation of the Deaf, there are approximately 720 million deaf people globally and they use 300 different types of sign languages. As per the 2011 census, about 50 lakh people in India are deaf or mute or have poor hearing abilities.

Although technology has been evolving rapidly in this information age, deaf/mute people still use sign language as their only way of communication. Using sign language as a communication tool can be beneficial among those who are familiar with this language, but the problem remains when communicating with the wider community. Sign Language Translator is the appropriate solution that enables deaf/mute people to communication fluently through technology in different languages. As sign language is a formal language employing a system of hand gesture for communication (by the deaf).

With the impetuous advancement of informatics, human-computer interaction is paving the way for new eras. There have been several advancements in technology and a lot of research has been done to help the people who are impaired using sign language (gesture) recognition which includes both image-based and sensor-based approaches. Hand gesture recognition systems provide us with an innovative, natural, user-friendly way of communication with the computer which is more familiar to human beings nowadays. Automatic sign language recognition offers enhancement of communication capabilities for the speech and hearing impaired. It promises improved social opportunities and integration into society for these people.

1.1 PROJECT OBJECTIVE

The aim of the project is to develop a hand glove equipped with sensors such as Flex sensor and Accelerometer which sense different sign language gestures. Flex sensors are placed on fingers which measure the bending of fingers according to a gesture made. An accelerometer is placed on the palm which measures the location of the hand in X, Y, Z axes. Firstly, sensors were simulated to extract the sensed data. Secondly the sensed data from sensors is sent to Arduino UNO board for further processing and transfer data to an android phone via Bluetooth module. The data will be in the form of text. This text data is then converted into speech.

1.2 PROJECT OUTLINE

This project report is presented over the four remaining chapters. Chapter 2 describes the sign languages. Chapter 3 presents the principle of operation of sensors and various components used in the project. Chapter 4 explains the concepts of Arduino programming. Chapter 5 presents the simulation results of the various signs simulated using the Arduino simulator. Finally, conclusions are drawn in chapter 6.

CHAPTER 2

SIGN LANGUAGE DESCRIPTION

Sign languages are visual languages that use hand, facial and body movements as a means of communication. There are over 135 different sign languages all around the world including American Sign Language (ASL), Australian Sign Language (Auslan) and British Sign Language (BSL). There are also signed representations of oral languages such as Signed Exact English (SEE) and mixes such as Pidgin Signed English (PSE). Sign language is commonly used as the main form of communication for people who are Deaf or hard of hearing, but sign languages also have a lot to offer for everyone. Sign languages are an extremely important communication tool for many deaf and hard-of- hearing people. Sign languages are the native languages of the Deaf community and provide full access to communication. Although sign languages are used primarily by people who are deaf, they are also used by others, such as people who can hear but can’t speak. People who know a sign language are often much better listeners. When using a sign language, a person must engage in constant eye contact with the person who is speaking. Unlike spoken language, with sign languages a person cannot look away from the person speaking and continue to listen. This can be an extremely beneficial habit to have for spoken language as well as sign language. By maintaining eye contact in spoken language, it shows that a person is genuinely interested in what the other is saying.

2.1 Different Sign languages

American Sign Language (ASL):

American Sign Language (ASL) is a complete, natural language that has the same linguistic properties as spoken languages, with grammar that differs from English. ASL is expressed by movements of the hands and face. It is the primary language of many North Americans who are deaf and hard of hearing and is used by some hearing people as well. Fig 2.1 shows 26 alphabets and 10 numbers in the ASL.

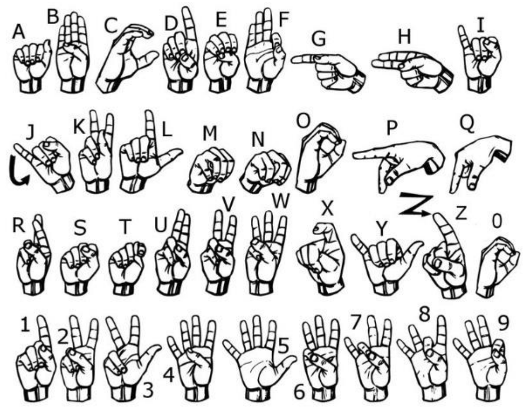


Fig 2.1 : American Sign Language

British, Australian and New Zealand Sign Language (BANZSL):

Sharing a sign language alphabet is British Sign Language, Australian Sign Language (Auslan) and New Zealand Sign Language. Unlike ASL, these alphabets use two hands, instead of one and it is shown in Fig.2.2.

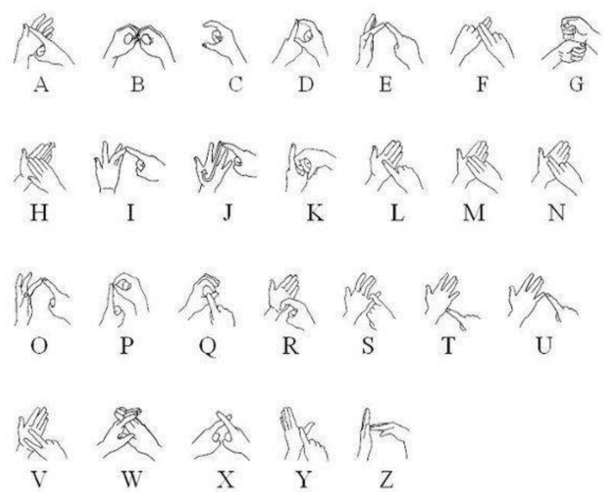


Fig 2.2 : Mexican Sign Language

Chinese Sign Language (CSL):

CSL’s signs are visual representations of written Chinese characters, they use a_one handed alphabet as shown in Fig.2.3. There are many CSL dialects but the Shanghai dialect is the most common. The language has been developing since the late 1950’s and The Chinese National Association of the Deaf, is working hard to raise awareness and promote use of the language throughout the country.

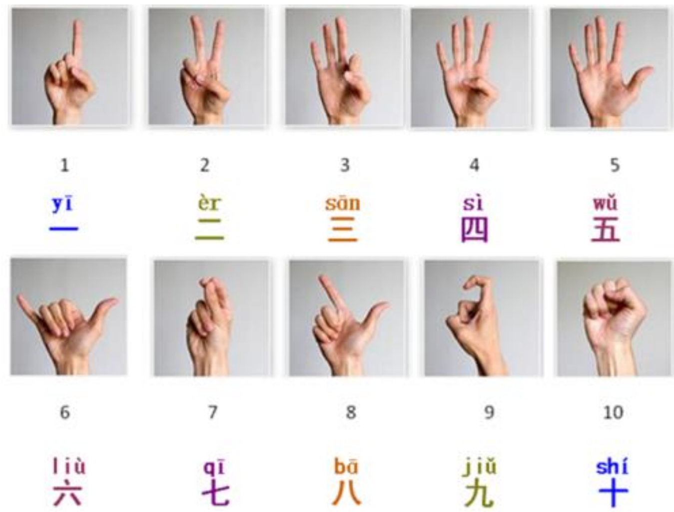


Fig 2.3 Chinese Sign Language

French Sign Language (LSF):

French Sign Language is similar to ASL – since it is in fact the origin of ASL – but there are minor differences throughout. LSF also has a pretty fascinating history. LSF is shown in Fig.2.4.

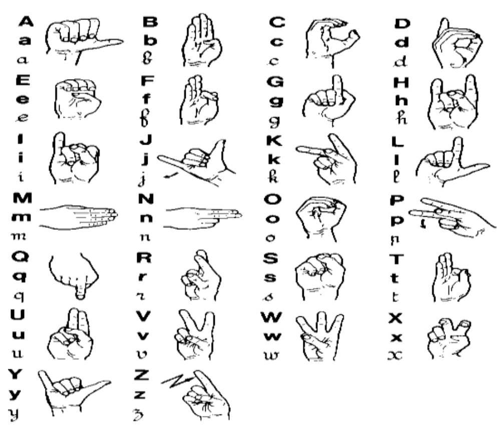


Fig 2.4 French Sign Language

Japanese Sign Language (JSL) Syllabary:

The Japanese Sign Language (JSL) Syllabary is based on the Japanese alphabet, which is made up of phonetic syllables. JSL is known as Nihon Shuwa in Japan and as shown in Fig.2.5.

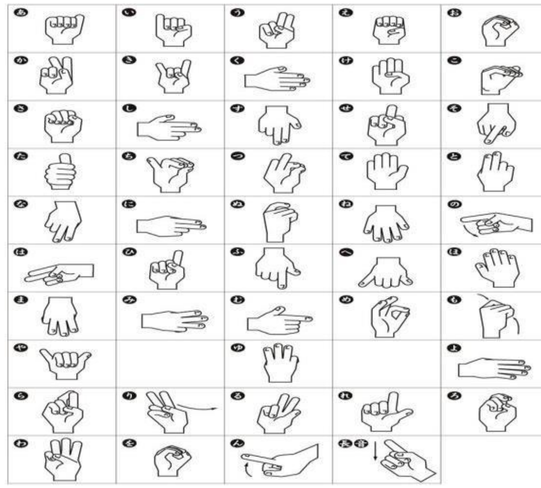


Fig 2.5 Japanese Sign Language

Arabic Sign Language:

The Arab sign-language family is a family of sign languages across the Arab Mideast. Data on these languages is somewhat scarce, but a few languages have been distinguished, including Levantine Arabic Sign Language. Arabic sign language is shown in Fig.2.6.

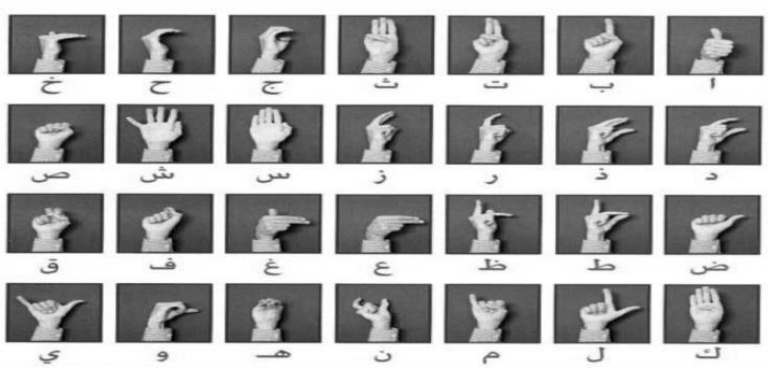


Fig 2.6 Arabic Sign Language

Spanish Sign Language (LSE):

Spanish Sign Language is officially recognized by the Spanish Government. It is native to Spain, except Catalonia and Valencia. Many countries that speak Spanish do not use Spanish Sign Language! (See Mexican Sign Language below, for example.) SSL is mainly used in Spain and there is an estimated 100,000 signers of SSL. SSL is completely different from ASL, in the same way that English is different from Spanish. SSL is used across all of Spain, except in Catalonia which uses Catalan Sign Language and Valencia which uses Valencian Sign Language. LSE is shown in Fig.2.7.

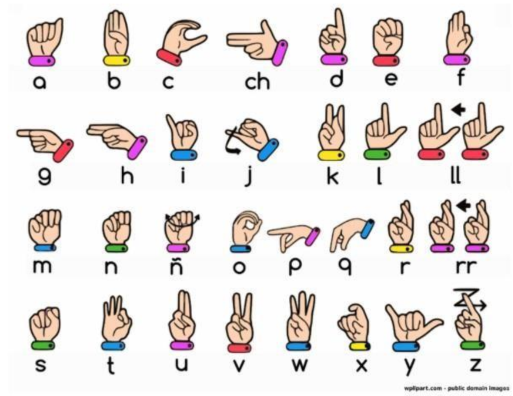


Fig 2.7 Spanish Sign Language

Mexican Sign Language (LSM):

Mexican Sign Language ('lengua de señas mexicana' or LSM) is different from Spanish, using different verbs and word order. The majority of people who use Mexican Sign Language reside in Mexico City, Guadalajara and Monterrey. Variation in this language is high between age groups and religious backgrounds. LSM is shown in Fig.2.8.

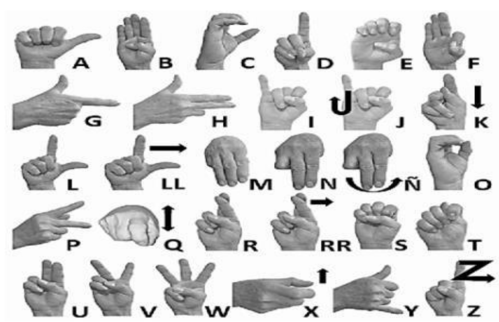


Fig 2.8 Mexican Sign Language

Ukrainian Sign Language (USL):

Ukrainian Sign Language is derived from the broad family of French Sign Languages. It uses a one-handed manual alphabet of 33 signs, which make use of the 23 handshapes of USL andis shown in Fig.2.9

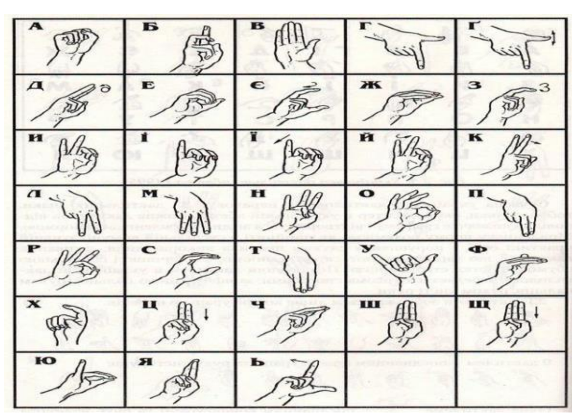


Fig 2.9 Ukrainian Sign Language

2.2 STANDARD SIGNS

There is no universal sign language. Different sign languages are used in different countries or regions. For example, British Sign Language (BSL) is a different language from ASL, and Americans who know ASL may not understand BSL. Some countries adopt features of ASL in their sign languages.

Fig 2.10 Standard Signs

It's not always practical to spell out words for everyday interactions. That's where these expressions come in handy! And are as shown in Fig.2.10. We can use common expressions to meet people, show your appreciation, and communicate with friends. It becomes easy forimpaired persons to communicate with normal persons.

2.3 ADVANTAGES

- It reduces frustration.
- It increases self-esteem.
- It enhances languages and listening skills.
- It enriches relationships.
- It provides a window into your child's world.
- It increases their IQ.

CHAPTER 3

LITERATURE REVIEW

There have been various works dealing with sign language translation and its different application. Mainly the following five papers were reviewed to get insight about sign language recognition.

[1] The Language of Glove Wireless gesture decoder with low-power and stretchable hybrid electronics (2017) written by Timothy F. O'Connor, Matthe E. Fach, Rachel Miller, Samuel E.Root, Patrick P. Mercier, Darren J. Lipoma.

This communication describes a glove capable of wirelessly translating the American Sign Language (ASL) alphabet into text displayable on a computer or smartphone. The key components of the device are strain sensors comprising a piezoresistive composite of carbon particles embedded in a fluor elastomer. These sensors are integrated with a wearable electronic module consisting of digitizers, a microcontroller, and a Bluetooth radio.

[**2] Sensor Based Hand Gesture Recognition System for English Alphabets used in Sign Language of Deaf and Mute People, IEEE Sensors (2018)** written by Abhishek B. Jani, Nishith A. Kotak, and Anil K. Roy.

Hand gesture and sign language are significant ways of communication for deaf mute people. It puts a barrier in comprehension of conversation between a mute person and a normal person, because a normal person does not understand the sign language. In this paper we developed a sensor-based device which deciphers this sign language of hand gesture for English alphabets. We propose that if this wearable device, which is a hand glove, is put on by a mute person, the device would recognize the 26 letters almost accurately. We discuss the challenges and future potential of this device so that it would completely be able to facilitate communication of such class of people. Hand gesture recognition is also a challenging problem of the human computer interface area.

[3] Hear Sign Language: A Real-Time End-to-End Sign Language Recognition System (2022) written by prepared by Zhibo Wang, Tengda Zhao, JinxiMa , Hongkai ChenKaixin Liu, Huajie Shao, Qian Wang and Ju Ren.

In this paper the design of a novel real-time end-to-end SLR system, called Deep SLR, to translate sign language into voices to help people “hear” sign language. Specifically, two armbands embedded with an IMU sensor and multi-channel sEMG sensors are attached on the forearms to capture both coarse-grained arm movements and fine-grained finger motions and propose an attention-based encoder-decoder model with a multi-channel convolutional neural network (CNN) to realize accurate, scalable, and end-to-end continuous SLR without sign segmentation.

[4] Electromyography - Based Gesture Recognition: Is It Time to Change Focus from the Forearm to the Wrist? (2022) done by Fady S. Botros (Graduate Student Member, IEEE) Angkoon Phinyomark (Member, IEEE,) and Erik J. Scheme Senior Member, IEEE).

The direct comparison of signal and information quality is con- ducted between concurrently recorded wrist and forearm signal. The study proved that the EMG signals from the forearm signals a better than wrist signals. People are more used to wrist-worn devices. Hence the intrusion of electromyography(EMG) sensors into light weight and feasible wearable designs has gained the attention of many. The main reason for this is the use of EMG signals. This study is having a comprehensive and systematic investigation of the feasibility of hand gesture recognition using EMG signals recorded at the wrist. The tests done using wrist EMG signals achieved average accuracy levels of 92.1% for single-finger gestures, 91.2%for multi- finger gestures, and 94.7% for the conventional wrist gestures.

[5]A 3D Printed Soft Robotic Hand With Embedded Soft Sensor for Direct Transition Between Hand Gesture and Improved Grasping Quality and Diversity (2022) written by Hao Zhou, Charbe Tawk, and Gursel Alici, Member, IEEE

A three-dimensional (3D) printed soft robotic hand with embedded soft sensors, intended for prosthetic applications is designed and developed to efficiently operate with new-generation myoelectric control systems, e.g., pattern recognition.

All these papers helped to extract ideas like using a real-time end-to-end sign language recognition system to use low power and stretchable hybrid electronics, to use robotic hand with embedded soft sensor.

CHAPTER 4

METHODOLOGY

The system that we have developed is basically a smart glove which consist of flex sensors that recognizes the sign (as per American sign language) and shows the corresponding alphabet in textual and speech format. Each sign represents a English alphabet which can be shown using single hand.

4.1 BLOCK DIAGRAM

The block diagram of the system is as shown above. The flex sensors sense real time movements of the fingers. An accelerometer sensor is a tool that measures the acceleration of any body or object in its instantaneous rest frame. These are then connected to Arduino UNO board and Bluetooth module. The Arduino board converts hand movements into corresponding signals and these are transmitted using Bluetooth. The mobile application converts the signal received into corresponding text and speech.

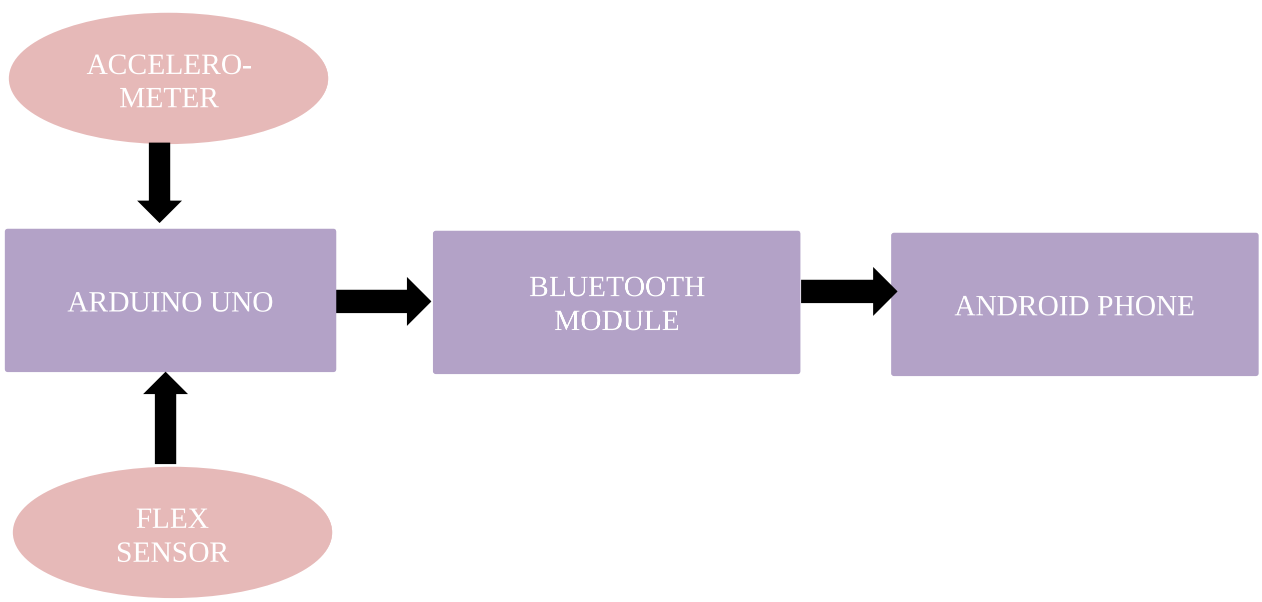


Fig 4.1 Block Diagram

Signature Not Verified

Principal.VJEC

4.2 CIRCUIT DIAGRAM

When the person who wears the glove performs a symbol, the values of the flex sensors varies. The values from the five flex sensors attached to the five fingers are recorded accordingly and transmitted to the analog pins of Arduino-UNO. Then the Arduino receives these values and corresponding text is transferred to the Android phone through the Bluetooth module attached in between them.

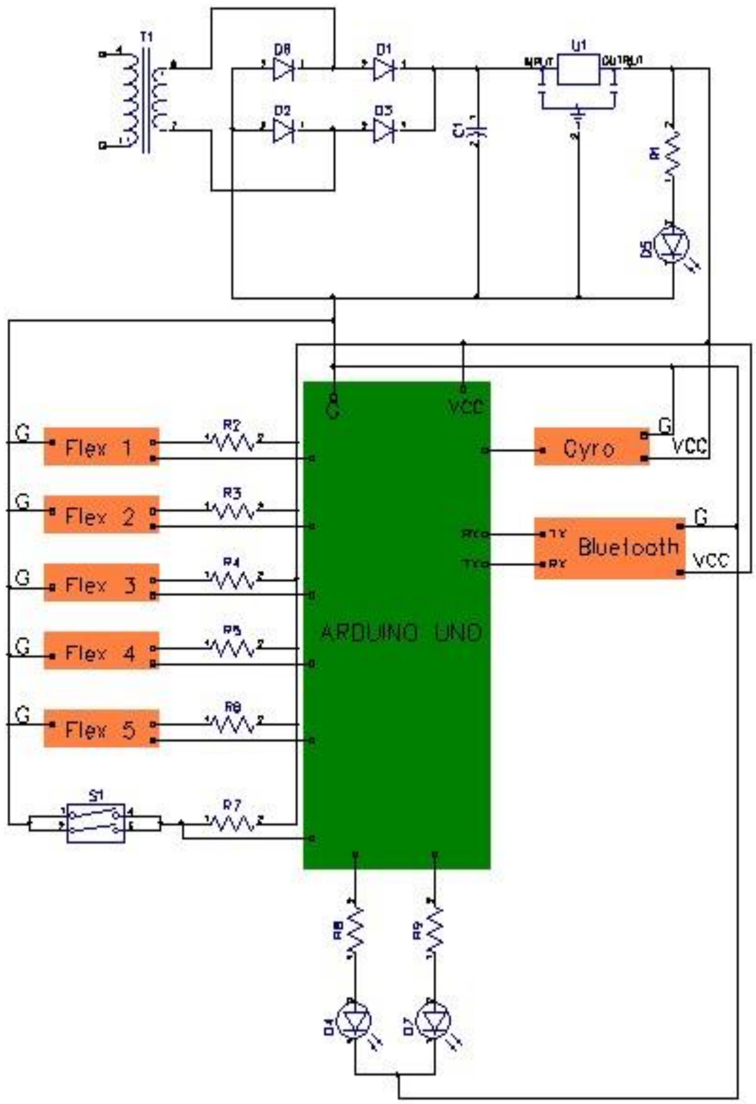


Fig 4.2 Circuit diagram

Signature Not Verified

Principal.VJEC

CHAPTER 5

COMPONENTS

5.1 HARDWARE REQUIREMENTS

The proposed system consists of primarily two sections:

1. Transmitter Section
2. Receiver Section

The devices contained in the transmitter section are:

1. Flex sensors
2. Accelerometer Sensor (MPU6050)
3. HC-05 Bluetooth Module
4. Arduino Uno Microcontroller.

The gloves contain flex sensors which are the main sensors for this product. They are devices which can show variable resistance based on various bend angles. The sensors are connected in a voltage divider circuit such that the resultant analog voltage is sent to one analog port of the micro-controller. The glove is mounted with 5 flex sensors, each on a finger.

5.1.1 FLEX SENSORS

A **flex sensor** or **bend sensor** is an easy-to-use sensor specifically designed to measure the amount of bending or deflection People have been using it as a goniometer to determine joint movement, a door sensor, a bumper switch for wall detection or a pressure sensor on robotic grippers, etc.

5.1.1.1Flex Sensor overview

A flex sensor is basically a variable resistor that varies in resistance upon bending. Since the resistance is directly proportional to the amount of bending, it is often called a **Flexible Potentiometer**. Flex sensors are generally available in two sizes: one is 2.2″ (5.588cm) long and another is 4.5″ (11.43cm) long. They are also divided based on resistance. There are LOW resistance, MEDIUM resistance and HIGH resistance types. Choose the appropriate type depending on requirement.

Flex sensors are designed to flex in only one direction. Bending the sensor in another direction may damage it. Also take care not to bend the sensor close to the base, because the bottom of the sensor (where the pins are crimped on) is very fragile and can break when bent over.

Fig 5.1 Flex Sensor

5.1.1.2 Flex Sensor Working

The conductive ink printed on the sensor acts as a resistor. When the sensor is straight, this resistance is about 25k as shown in Fig.3.2

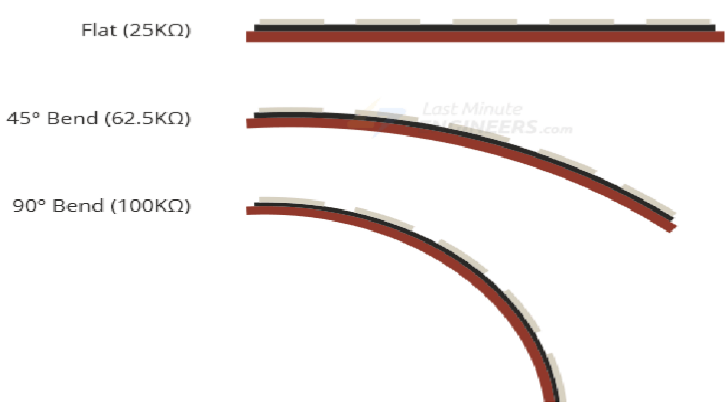


Fig 5.2 Flex sensor working

When the sensor is bent, conductive layer is stretched, resulting in reduced cross section (imagine stretching a rubber band). This reduced cross section results in an increased resistance. At 90° angle, this resistance is about 100KΩ. When the sensor is straightened again, the resistance returns to its original value. By measuring the resistance, you can determine how much the sensor is bent.

5.1.2 ACCELEROMETER SENSOR: (MPU6050)

In recent years, some crafty engineers successfully made micro machined gyroscopes. These MEMS (microelectromechanical system) gyroscopes have paved the way to a completely new set of innovative applications such as gesture recognition, enhanced gaming, augmented reality, panoramic photo capture, vehicle navigation, fitness monitoring and many more, no doubt the gyroscope and accelerometer are great in their own way. But when we combine them, we can get very accurate information about the orientation of an object. This is where the **MPU6050** comes in. The **MPU6050** has both a gyroscope and an accelerometer, using which we can measure rotation along all three axes, static acceleration due to gravity, as well as motion, shock, or dynamic acceleration due to vibration.

5.1.2.1 MPU6050 Module Hardware Overview

At the heart of the module is a low power, inexpensive 6-axis Motion Tracking chip that combines a 3-axis gyroscope, 3-axis accelerometer, and a Digital Motion Processor (DMP)all in a small 4mm x 4mm package as shown in Fig.3.5. It can measure angular momentum orrotation along all the three axes, the static acceleration due to gravity, as well as dynamic acceleration resulting from motion, shock, or vibration.

Fig 5.3 MPU6050

The module comes with an on-board LD3985 3.3V regulator, so you can use it with a 5V logic microcontroller like Arduino without worry. The MPU6050 consumes less than 3.6mA during measurements and only 5µA during idle. This low power consumption allows the implementation in battery driven devices. In addition, the module has a power LED thatlights up when the module is powered.

5.1.2.2 MPU6050 Module Pinout

The pin diagram of the module is as shown in Fig.3.6 below

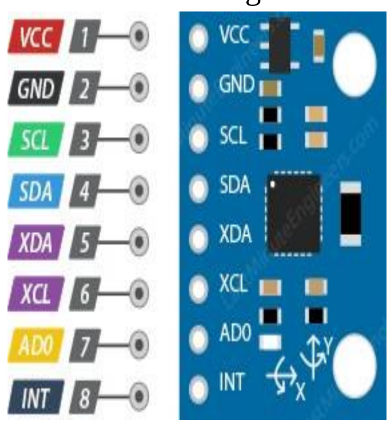


Fig 5.4 MPU6050 Module Pinout

VCC is the power supply for the module. Connect it to the 5V output of the Arduino.

GND should be connected to the ground of Arduino.

SCL is a I2C Clock pin. This is a timing signal supplied by the Bus Master device. Connect to the SCL pin on the Arduino.

SDA is a I2C Data pin. This line is used for both transmit and receive. Connect to the SDA pin on the Arduino.

XDA is the external I2C data line. The external I2C bus is for connecting external sensors.

XCL is the external I2C clock line.

AD0 allows you to change the internal I2C address of the MPU6050 module. It can be used if the module is conflicting with another I2C device, or if you wish to use two MPU6050s on the same I2C bus. When you leave the ADO pin unconnected, the default I2C address is 0x68HEX and when you connect it to 3.3V, the I2C address becomes 0x69HEX.

INT is the Interrupt Output. MPU6050 can be programmed to raise interrupt on gesture detection, panning, zooming, scrolling, tap detection, and shake detection.

5.1.3.HC-05 BLUETOOTH MODULE

HC-05 Bluetooth Module is an easy-to-use Bluetooth SPP (Serial Port Protocol) module, designed for transparent wireless serial connection setup. Its communication is via serial communication which makes an easy way to interface with controller or PC.

The table 5.1 below gives the pin description of HC-05 Module.

Table 5.1 : HC-05 pin description

Sl. No	Pin Name	Pin Description
1	Enable Key	This pin is used to toggle between Data Mode (set low) and AT command mode(set high). By default, it is in Data mode
2	Vcc	Powers the module. Connect to +5V Supply voltage
3	Ground	Ground pin of module, connect to system ground.
4	TXD– Transmitter	Transmits Serial Data. Everything received via Bluetooth will be given out by thispin as serial data.
5	RXD– Receiver	Receive Serial Data. Every serial data given to this pin will be broadcasted viaBluetooth
6	State	The state pin is connected to on board LED, it can be used as feedback to check if Bluetooth is working properly.
7	LED	Indicates the status of Module Blink once in 2 sec: Module has entered Command ModeRepeated Blinking: Waiting for connection in Data Mode Blink twice in 1 sec: Connection successful in Data Mode
8	Button	Used to control the Key/Enable pin to toggle between Data and command Mode

Signature Not Verified

Principal.VJEC

5.1.3.1 HC-05 BLUETOOTH MODULE PINOUT

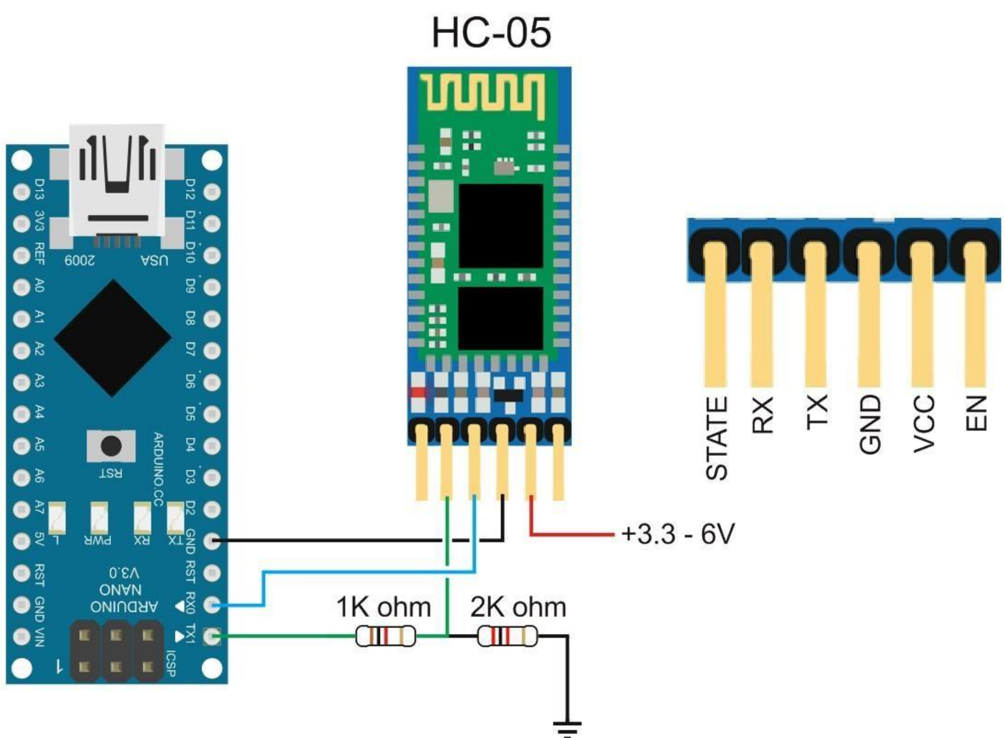


Fig 5.5 HC-05 BLUETOOTH MODULE

5.1.4 ARDUINO UNO

The Arduino Uno is an open-source microcontroller board based on the Microchip ATmega328P microcontroller and developed by Arduino.cc. The board is equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards (shields) and other circuits. The board has 14 digital I/O pins (six capable of PWM output), 6 analog I/O pins, and is programmable with the Arduino IDE (Integrated Development Environment), via a type B USB cable. It can be powered by the USB cable or by an external 9-volt battery, though it accepts voltages between 7 and 20 volts. It is similarto the Arduino Nano and Leonardo. The hardware reference design is distributed under a Creative Commons Attribution Share-Alike 2.5 license and is available on the Arduinowebsite. Layout and production files for some versions of the hardware are also available.

The word "uno" means "one" in Italian and was chosen to mark the initial release of Arduino Software. The Uno board is the first in a series of USB-based Arduino boards; it and version 1.0 of the Arduino IDE were the reference versions of Arduino, which have now evolved to newer releases. The ATmega328 on the board comes preprogrammed with a bootloader that allows uploading new code to it without the use of an external hardware programmer. While the Uno communicates using the original STK500 protocol, it differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it uses the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter.

5.1.4.1ARDUINO UNO PINOUT:

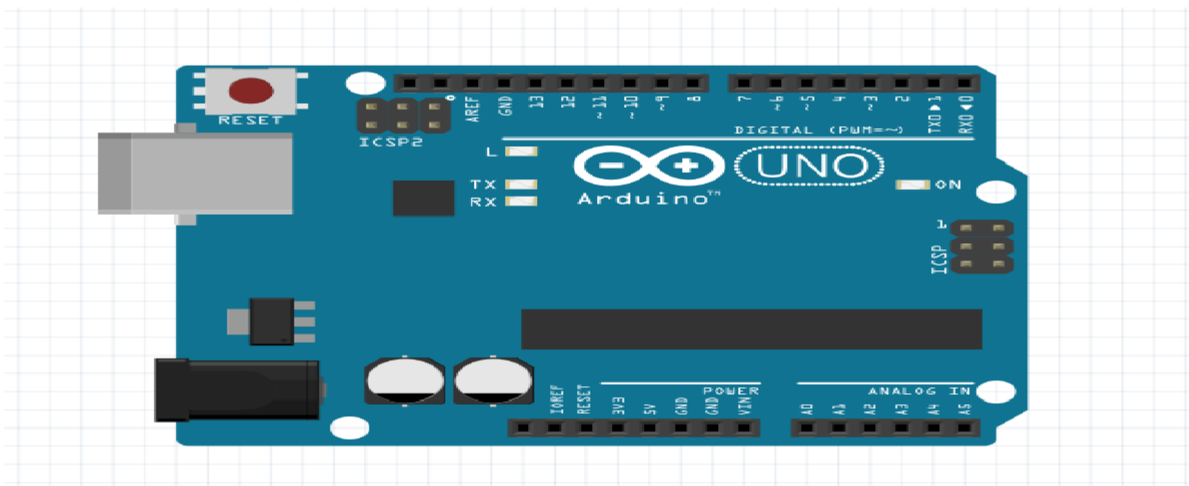


Fig 5.6 ARDUINO UNO

General pin functions

- **LED**: There is a built-in LED driven by digital pin 13. When the pin is high value, the LED is on, when the pin is low, it is off.
- **VIN**: The input voltage to the Arduino/Genuino board when it is using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- **5V**: This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 - 20V), the USB connector (5V), or the VIN pin of the board (7-20V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage the board.
- **3V3**: A 3.3-volt supply generated by the on-board regulator. Maximum current draw

is50 mA.

- **GND**: Ground pins.
- **IOREF**: This pin on the Arduino/Genuino board provides the voltage reference with which the microcontroller operates. A properly conFigured shield can read the IOREF

  pin voltage and select the appropriate power source, or enable voltage translators on theoutputs to work with the 5V or 3.3V.
- **RESET**: Typically used to add a reset button to shields that block the one on the board.

5.2 SOFTWARE REQUIREMENT

5.2.1 DIPTRACE

DipTrace is a software suite for electronic design automation (EDA) to create schematic diagrams and printed circuit board layouts. DipTrace has four modules: schematic capture editor, PCB layout editor with built-in shape-based autorouter and 3D preview, component editor, and pattern editor. The PCB board design are done using this software.

Fig 5.7 : DipTrace

5.2.2 ARDUINO IDE

The Arduino IDE (Integrated Development Environment) is used to write the computer code and upload this code to the physical board. The Arduino IDE is very simple and this simplicity is probably one of the main reason Arduino became so popular. We can certainly state that being compatible with the Arduino IDE is now one of the main requirements for a new microcontroller board. Over the years, many useful features have been added to the Arduino IDE and you can now managed third-party libraries and boards from the IDE, and still keep the simplicity of programming the board. The main window of the Arduino IDE is shown below,

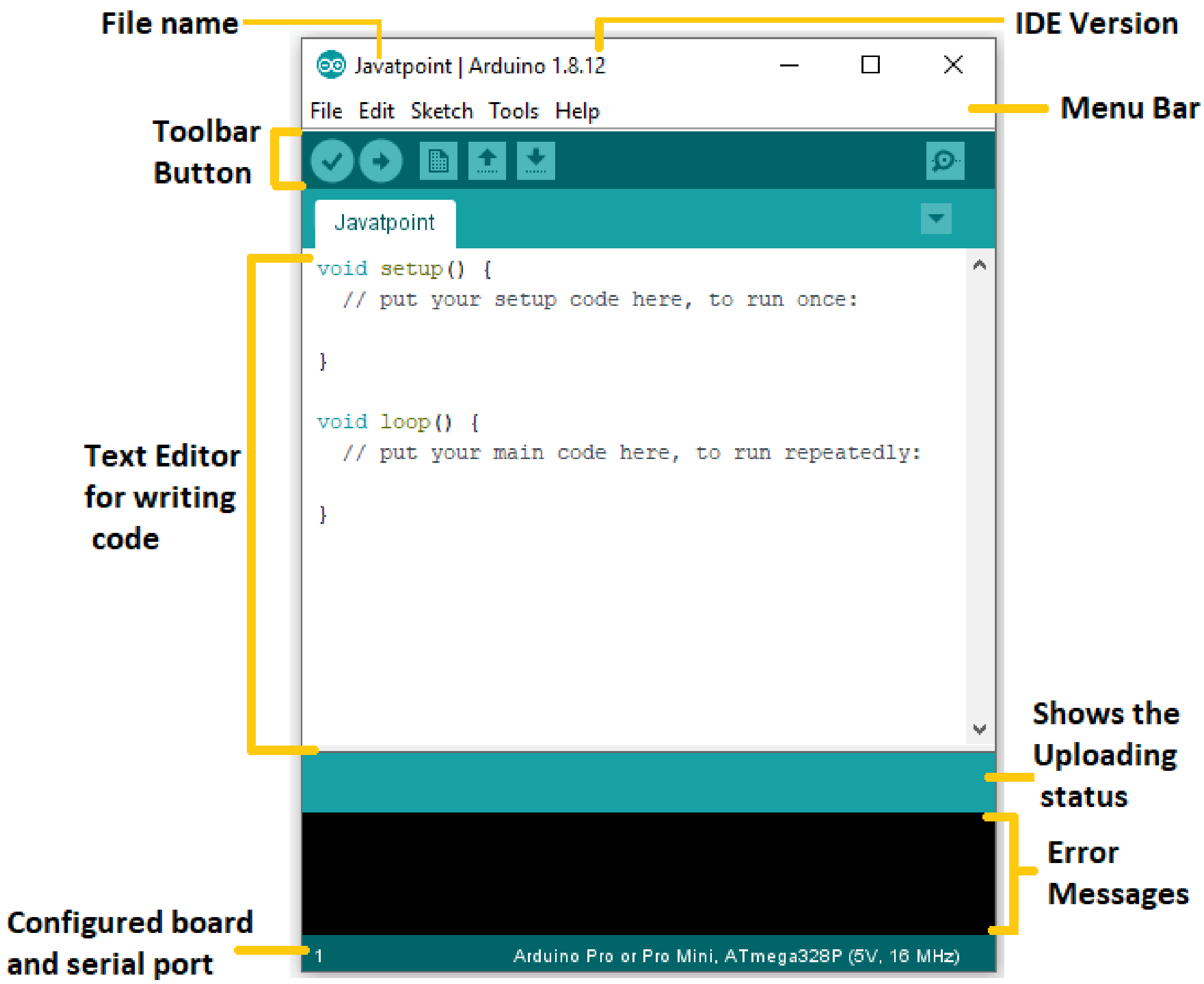


Fig 5.8 Arduino IDE

CHAPTER 6

ARDUINO PROGRAMMING

Arduino is a prototype platform based on an easy-to-use hardware and software. It consists of a circuit board, which can be programed (referred to as a microcontroller) and a ready-made software called Arduino IDE (Integrated Development Environment), which is used to write and upload the computer code to the physical board.

The key features are:

- Arduino boards are able to read analog or digital input signals from different sensors and turn it into an output such as activating a motor, turning LED on/off, connect to the cloud and many other actions.
- You can control your board functions by sending a set of instructions to the microcontroller on the board via Arduino IDE (referred to as uploading software).
- Unlike most previous programmable circuit boards, Arduino does not need an extra piece of hardware (called a programmer) in order to load a new code onto the board. You can simply use a USB cable.
- Additionally, the Arduino IDE uses a simplified version of C++, making it easier to learn to program.
- Finally, Arduino provides a standard form factor that breaks the functions of the microcontroller into a more accessible package.

6.1 ARDUINO BOARD

Various kinds of Arduino boards are available depending on different microcontrollers used. However, all Arduino boards have one thing in common: they are programed through the Arduino IDE.

The differences are based on the number of inputs and outputs (the number of sensors, LEDs,and buttons you can use on a single board), speed, operating voltage, form factor etc. Some boards are designed to be embedded and have no programming interface (hardware), which you would need to buy separately. Some can run directly from a 3.7V battery, others need atleast 5V.

6.2 BOARD DESCRIPTION

1- Power USB:

Arduino board can be powered by using the USB cable from your computer. All you need to do is connect the USB cable to the USBconnection (1).

2- Power (Barrel Jack):

Arduino boards can be powered directly from the AC mains power supply by connecting it to the Barrel Jack (2).

3- Voltage Regulator:

The function of the voltage regulator is to control the voltage given to the Arduino board and stabilize the DC voltages used by the processor and other elements.

4- Crystal Oscillator:

The crystal oscillator helps Arduino in dealing with time issues. The Arduino calculates time by using the crystal oscillator. The number printed on top of the Arduino crystal is 16.000H9H. It tells us that the frequency is 16,000,000 Hertz or 16 MHz .

5, 17-Arduino Reset:

You can reset your Arduino board, i.e., start your program from the beginning. You can reset the UNO board in two ways. First, by using the reset button (17) on the board. Second, you can connect an external reset button to the Arduino pin labelled RESET (5).

6,7,8,9 -Pins (3.3, 5, GND, Vin) :

3.3V (6): Supply 3.3 output volt

5V (7): Supply 5 output volt

Most of the components used with Arduino board works fine with 3.3 volt and 5 volts.

GND (8) (Ground): There are several GND pins on the Arduino, any of which can be used toground your circuit.

Vin (9): This pin also can be used to power the Arduino board from an external power source, like

AC mains power supply.

10- Analog pins:

The Arduino UNO board has five analog input pins A0 to A5. These pins can read the signal from an analog sensor like the humidity sensor or temperature sensor and convert it into a digital value that can be read by the microprocessor.

11-Main microcontroller:

Each Arduino board has its own microcontroller (11). You can assume it as the brain of your board. The main IC (integrated circuit) on the Arduino is slightly different from board to board. The microcontrollers are usually of the ATMEL Company. You must know what IC your board has before loading up a new program from the Arduino IDE. This information is available on the top of the IC. For more details about the IC construction and functions, you can refer to the data sheet.

12-ICSP pin:

Mostly, ICSP (12) is an AVR, a tiny programming header for the Arduino consisting of MOSI, MISO, SCK, RESET, VCC, and GND. It is often referred to as an SPI (Serial Peripheral Interface), which could be considered as an "expansion" of the output. Actually, you are slaving the output device to the master of the SPI bus.

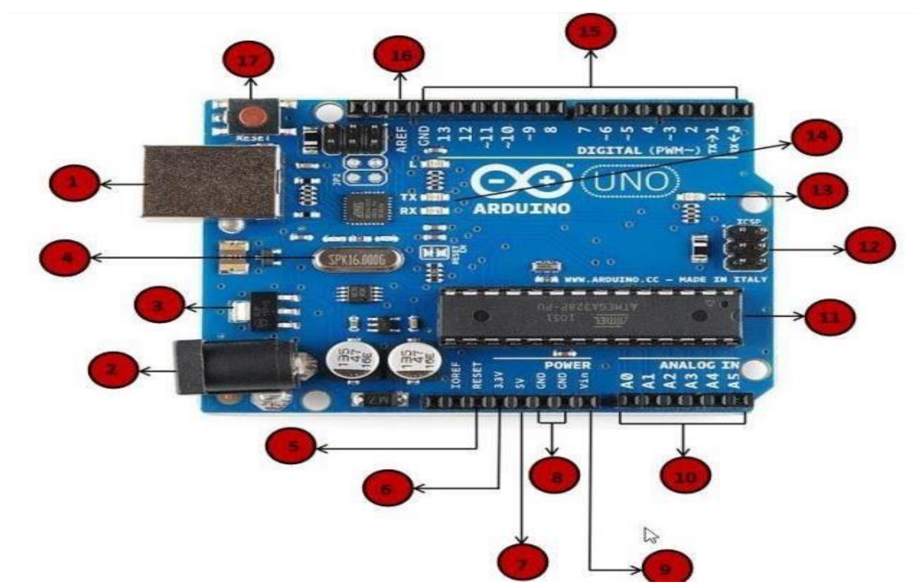


Fig 6.1 Arduino Uno Board

13-Power LED indicator:

This LED should light up when you plug your Arduino into a power source to indicate that your board is powered up correctly. If this light does not turn on, then there is something wrong with the connection.

14-TX and RX LEDs:

On your board, you will find two labels: TX (transmit) and RX (receive). They appear in two places on the Arduino UNO board. First, at the digital pins 0 and 1, to indicate the pins responsible for serial communication. Second, the TX and RX led (13). The TX led flashes with different speed while sending the serial data. The speed of flashing depends on the baud rate used by the board. RX flashes during the receiving process.

15-Digital I / O:

The Arduino UNO board has 14 digital I/O pins (15) (of which 6 provide PWM (Pulse Width Modulation) output. These pins can be configured to work as input digital pins to read logic 14 values (0 or 1) or as digital output pins to drive different modules like LEDs, relays, etc.The pins labeled “~” can be used to generate PWM.

16-AREF:

AREF stands for Analog Reference. It is sometimes, used to set an external reference voltage (between 0 and 5 Volts) as the upper limit for the analog input pins.

6.3 ARDUINO PINOUT

The Arduino UNO is a standard board of Arduino, which is based on an ATmega328P microcontroller. It is easier to use than other types of Arduino Boards.

The Arduino UNO Board, with the specification of pins, is shown in Fig 6.2:

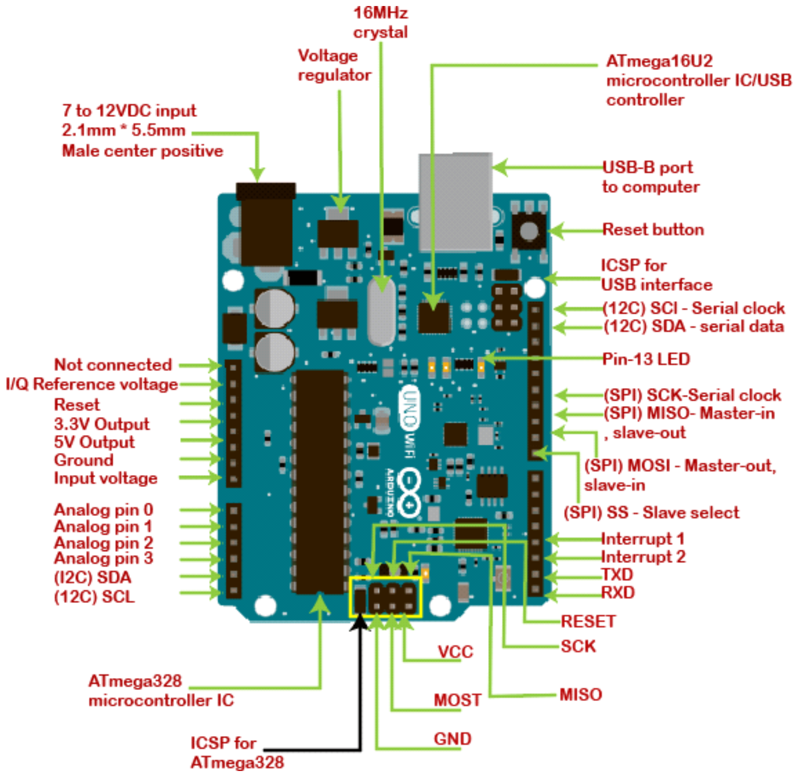


Fig 6.2 Pins Specifications

Signature Not Verified
Principal.VJEC

6.4 PROGRAMMING

6.4.1: BASICS

Coding Screen:

The coding screen is divided into two blocks. The 'setup' is considered as the preparation block, while the 'loop' is considered as the execution block. It is shown below:

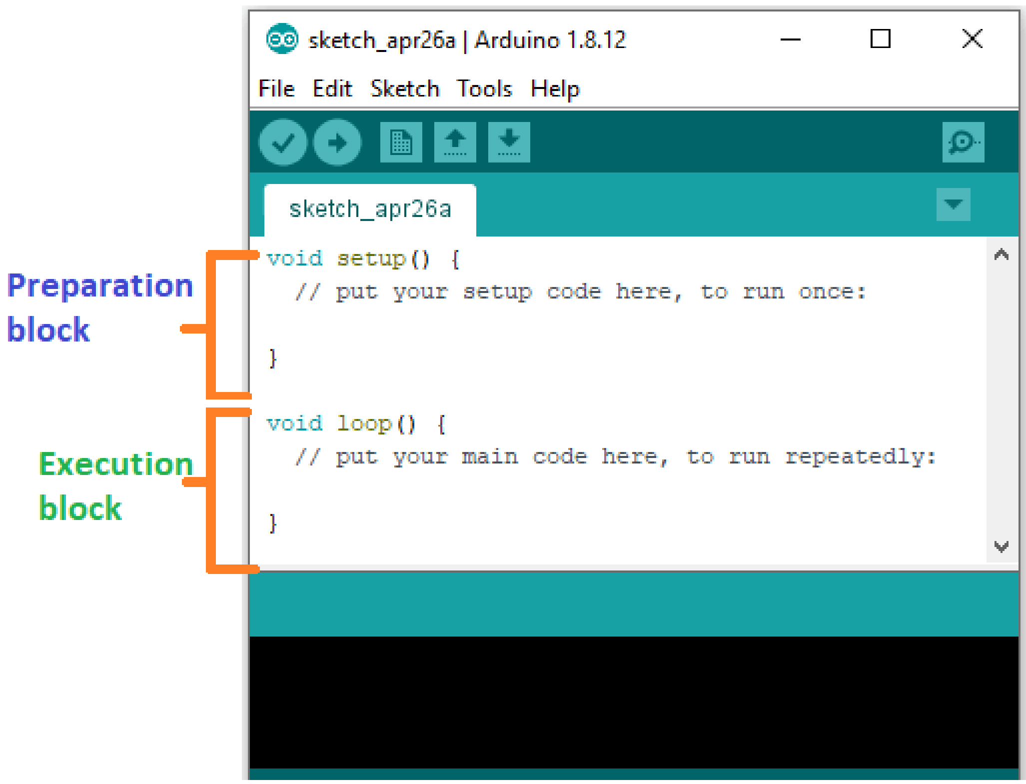


Fig 6.3 Coding Screen

The set of statements in the setup and loop blocks are enclosed with the curly brackets. We can write multiple statements depending on the coding requirements for a particular project.

Set Up ():

It contains an initial part of the code to be executed. The pin modes, libraries, variables, etc., are initialized in the setup section. It is executed only once during the uploading of the program and after reset or power up of the Arduino board.

Loop ():

The loop contains statements that are executed repeatedly. The section of code inside the curly brackets is repeated depending on the value of variables.

Pin Mode ():

The specific pin number is set as the INPUT or OUTPUT in the pinMode () function. The Syntax is: **pinMode (pin, mode)** where, **pin** is the pin number and we can select the pin number according to the requirements, and **Mode** is INPUT or OUTPUT according to the corresponding pin number.

The OUTPUT mode of a specific pin number provides a considerable amount of current to other circuits, which is enough to run a sensor or to light the LED brightly. The output state of a pin is considered as the low-impedance state.

Digital Write ():

The digitalWrite( ) function is used to set the value of a pin as HIGH or LOW. HIGH sets the value of the voltage. For the 5V board, it will set the value of 5V, while for3.3V, it will set the value of 3.3V. LOW sets the value = 0 (GND).

If we do not set the pinMode as OUTPUT, the LED may light dim.

The syntax is: **digitalWrite(pin, value HIGH/LOW)**

The digitalRead () function will read the HIGH/LOW value from the digital pin, and the digitalWrite () function is used to set the HIGH/LOW value of the digital pin.

Delay ():

The delay () function is a blocking function to pause a program from doing a task during the specified duration in milliseconds.

For example,

delay (2000) where, 1sec= 1000millisecond

Hence, it will provide a delay of 2 seconds.

6.4.2 Syntax and Program Flow

Syntax:

Syntax in Arduino signifies the rules to be followed for the successful uploading of the Arduino program to the board. The syntax of Arduino is similar to the grammar in English. It means that the rules must be followed in order to compile and run our code successfully. If we break those rules, our computer program may compile and run, but with some bugs.

Functions:

- The functions in Arduino combine many pieces of lines of code into one.
- The functions usually return a value after finishing execution. But here, the function does not return any value due to the presence of void.
- The setup and loop function have **void** keyword present in front of their function name.
- The multiple lines of code that a function encapsulates are written inside curly brackets.
- Every closing curly bracket ' **}** ' must match the opening curly bracket '**{** ' in the code.
- Arduino ignores the white spaces and tabs before the coding statements.

Tools Tab:

- The verify icon present on the tool tab only compiles the code. It is a quick method to check that whether the syntax of our program is correct or not.
- To compile, run, and upload the code to the board, we need to click on the Upload button.

Uses of Parenthesis ():

- It denotes the function like void setup ( ) and void loop ( ).
- The parameter's inputs to the function are enclosed within the parentheses.
- It is also used to change the order of operations in mathematical operations.

Semicolon :

- It is the statement terminator in the C as well as C++.
- A statement is a command given to the Arduino, which instructs it to take some kind of action. Hence, the terminator is essential to signify the end of a statement.
- We can write one or more statements in a single line, but with semicolon indicating the

end of each statement.

- The compiler will indicate an error if a semicolon is absent in any of the statements.
- It is recommended to write each statement with semicolon in a different line, which makes the code easier to read.
- We are not required to place a semicolon after the curly braces of the setup and loopfunction.

Arduino processes each statement sequentially. It executes one statement at a time before moving to the next statement.

Program Flow:

The program flow in Arduino is similar to the flowcharts. It represents the execution of a program in order.

The Arduino coding process in the form of the flowchart is shown below in Fig 6.4:

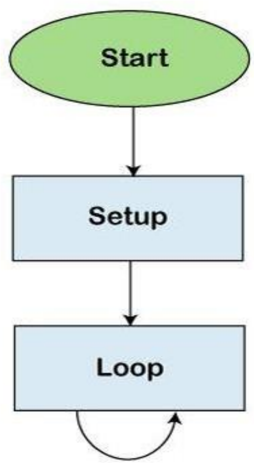


Fig 6.4 Program Flow Chart

6.4.3 Serial Functions

Serial.begin():

The serial.begin( ) sets the baud rate for serial data communication. The **baud** rate signifies the data rate in bits per second.

The default baud rate in Arduino is **9600 bps (bits per second**). We can specify other baudrates as well, such as 4800, 14400, 38400, 28800, etc.

Fig 6.5 Loop ( )

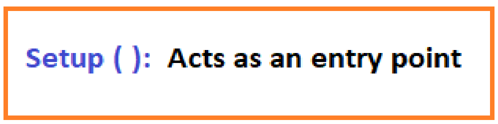


Fig 6.6 Setup ( )

The Serial.begin( ) is declared in two formats, which are shown below:

- begin(speed)
- begin(speed, conFig)

where, serial signifies the serial port object and speed signifies the baud rate or bps (bits per second) rate. conFig: It sets the stop, parity, and data bits.

Serial.print:

The serial.print ( ) in Arduino prints the data to the serial port. The printed data is stored in the ASCII (American Standard Code for Information Interchange) format, which is a human-readable text.

Each digit of a number is printed using the ASCII characters.

The printed data will be visible in the serial monitor, which is present on the right corner on the toolbar.

The Serial.print( ) is declared in two formats, which are shown below:

- print(value)
- print(value, format)

serial: It signifies the serial port object.

print: The print ( ) returns the specified number of bytes written.

value: It signifies the value to print, which includes any data type value.

format: It consists of number base, such as OCT (Octal), BIN (Binary), HEX (Hexadecimal), etc. for the integral data types. It also specifies the number of decimal places.

6.4.4 analogRead()

The **analogRead( )** function reads the value from the specified analog pin present on the particular Arduino board. The ADC (Analog to Digital Converter) on the Arduino board is a multichannel converter. It maps the input voltage and the operating voltage between the values 0 and 1023. The operating voltage can be **5V or 3.3V.** The values from 0 to 1023 are the integer values. The time duration to read an analog input signal on the boards (UNO, Mega, Mini, and Nano) is about 100 microseconds or 0.0001 seconds. Hence, the maximum reading rate of analog input is about 10000 times per second.

6.4.5 Arduino Data Types

The data types are used to identify the types of data and the associated functions for handling the data. It is used for declaring functions and variables, which determines the bit pattern and the storage space. The data types that we will use in the Arduino are listed below:

- void Data Type
- int Data Type
- Char Data Type
- Float Data Type
- Double Data Type
- Unsigned int Data Type
- short Data Type
- long Data Type
- Unsigned long Data Type

- byte data type
- word data type

6.4.6 Arduino Variables

The variables are defined as the place to store the data and values. It consists of a name, value, and type. The variables can belong to any data type such as int, float, char, etc.

Ex: **int pin=8**

Here, the **int** data type is used to create a variable named **pin** that stores the value **8**. It also means that value 8 is initialized to the variable **pin**.

6.4.7 Arduino Operators

The operators are used to solve logical and mathematical problems. For example, to calculate the temperature given by the sensor based on some analog voltage.

The types of Operators classified in Arduino are:

- Arithmetic Operators
- Boolean Operators
- Comparison Operators
- Bitwise Operators

1. **Arthematic Operators:**

There are six basic operators responsible for performing mathematical operations in Arduino, which are listed below:

- Assignment Operator (=):The Assignment operator in Arduino is used to set the variable's value. It is quite differentfrom the equal symbol (=) normally used in mathematics.
- Addition (+):The addition operator is used for the addition of two numbers.
  For example, P + Q.
- Subtraction (-):Subtraction is used to subtract one value from the another.
  For example, P - Q.

- Multiplication (*):The multiplication is used to multiply two numbers. For example, P * Q.
- Division (/): The division is used to determine the result of one number divide with another. Forexample, P/Q.
- Modulo (%):The Modulo operator is used to calculate the remainder after the division of one number by another number.

2. Boolean Operators:

The Boolean Operators are NOT (!), Logical AND (& &), and Logical OR (| |).Let's discuss the above operators in detail.

- Logical AND (& &): The result of the condition is true if both the operands in the condition are true.
- Logical OR (| |): The result of the condition is true, if either of the variables in the condition is true.
- NOT (!): It is used to reverse the logical state of the operand.

3. Comparison Operators:

The comparison operators are used to compare the value of one variable with the other. The comparison operators are listed below:

- less than (<):The less than operator checks that the value of the left operand is less than the right operand.The statement is true if the condition is satisfied.
- greater than (>): The less than operator checks that the value of the left side of a statement is greater than theright side. The statement is true if the condition is satisfied.
- equal to (= =): It checks the value of two operands. If the values are equal, the condition is satisfied.
- not equal to (! =): It checks the value of two specified variables. If the values are not equal, the condition willbe correct and satisfied.
- less than or equal to (< =): The less or equal than operator checks that the value of left side of a statement is less orequal to the value on right side.
  The statement is true if either of the condition is satisfied.
- greater than or equal to (> =): The greater or equal than operator checks that the value of the left side of a statement is greater or equal to the value on the

right side of that statement. The statement is true if the condition is satisfied.

4. Bitwise Operators:

The Bitwise operators operate at the binary level. These operators are quite easy to use.There are various bitwise operators. Some of the popular operators are listed below:

- bitwise NOT (~):The bitwise NOT operator acts as a complement for reversing the bits. bitwise XOR (^):The output is 0 if both the inputs are same, and it is 1 if the two input bits are different.
- bitwise OR (|): The output is 0 if both of the inputs in the OR operation are 0. Otherwise, the output is 1. Thetwo input patterns are of 4 bits.
- bitwise AND (&):The output is 1 if both the inputs in the AND operation are 1. Otherwise, the output is 0. Thetwo input patterns are of 4 bits.
- bitwise left shift (< <):The left operator is shifted by the number of bits defined by the right operator.
- bitwise right shift (> >):The right operator is shifted by the number of bits defined by the left operator.

6.4.8 Arduino IF statement

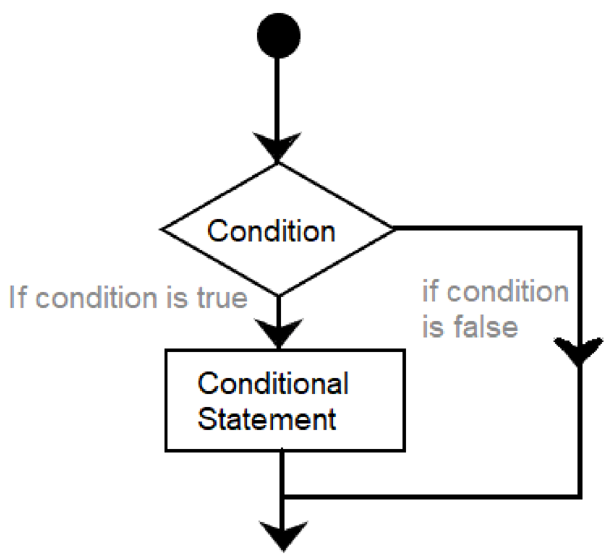


Fig 6.7 Flow chart of IF statement

1. If( ):

The if ( ) statement is the conditional statement, which is the basis for all types of programming languages. If the condition in the code is true, the corresponding task or function is performed accordingly. It returns one value if the condition in a program is **true**.It further returns another value if the condition is **false**. It means that if ( ) statement checks for the condition and then executes a statement or a set of statements.

2. If-Else:

The if-else condition includes if ( ) statement and else ( ) statement. The condition in the else statement is executed if the result of the If ( ) statement is false.

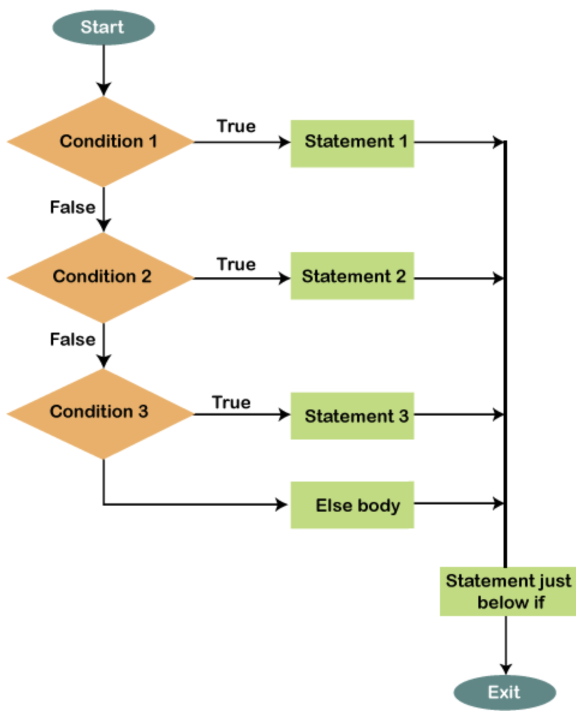


Fig 6.8 Flow Chart of If-Else

The statements will be executed one by one until the true statement is found. When the true statement is found, it will skip all other if and else statements in the code and runs the associated blocks of code.

3.Else-if:

The else if statement can be used with or without the else ( ) statement. We can include multiple else if statements in a program. The else if ( ) statement will stop the flow once its execution is true.

CHAPTER 7

RESULT

During the implementation, all the three sensors are connected to Arduino Uno board using jumper wires. Once the connections are made perfectly, then Arduino takes inputs from two sensors (Flex sensor, accelerometer). Flex sensors are placed on fingers which measure the bending of fingers according to the gesture made with the glove. An accelerometer is placed on the palm which measures the location of the hand in X, Y, Z axes. Initially, only flex sensors are implemented in this sign language transition. But, some hand gestures are similar to other gestures. To distinguish these types of gestures an additional sensor Accelerometer is also implemented. This is very important in distinguishing two signs when they have the same bend in the fingers but different bends in the palm. Similarly in the case of touch sensor.

The designed circuit has been connected and tested with many Hand gesture where the voice was clear for all the gestures, some examples are shown below:

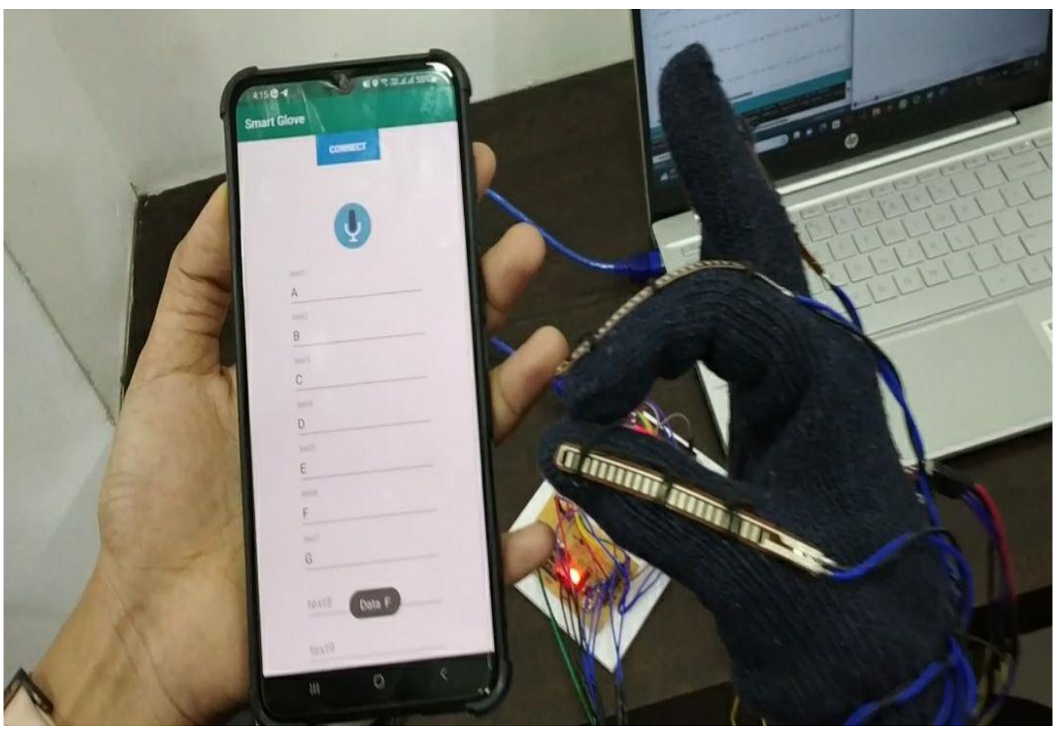


Fig 7.1 Gesture and output for F

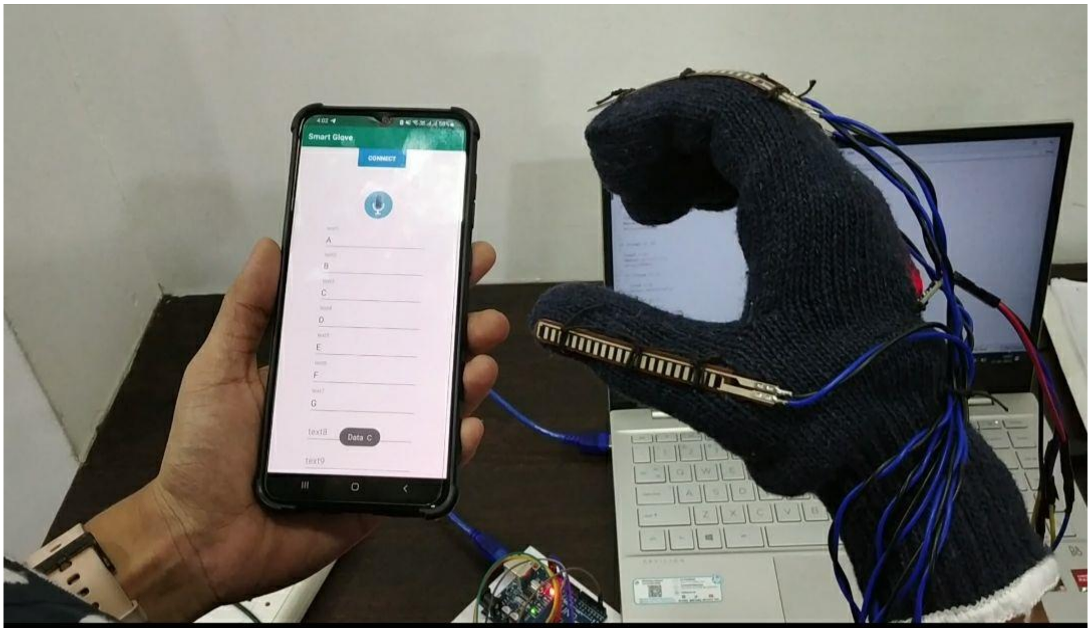


Fig 7.2Gesture and output for C

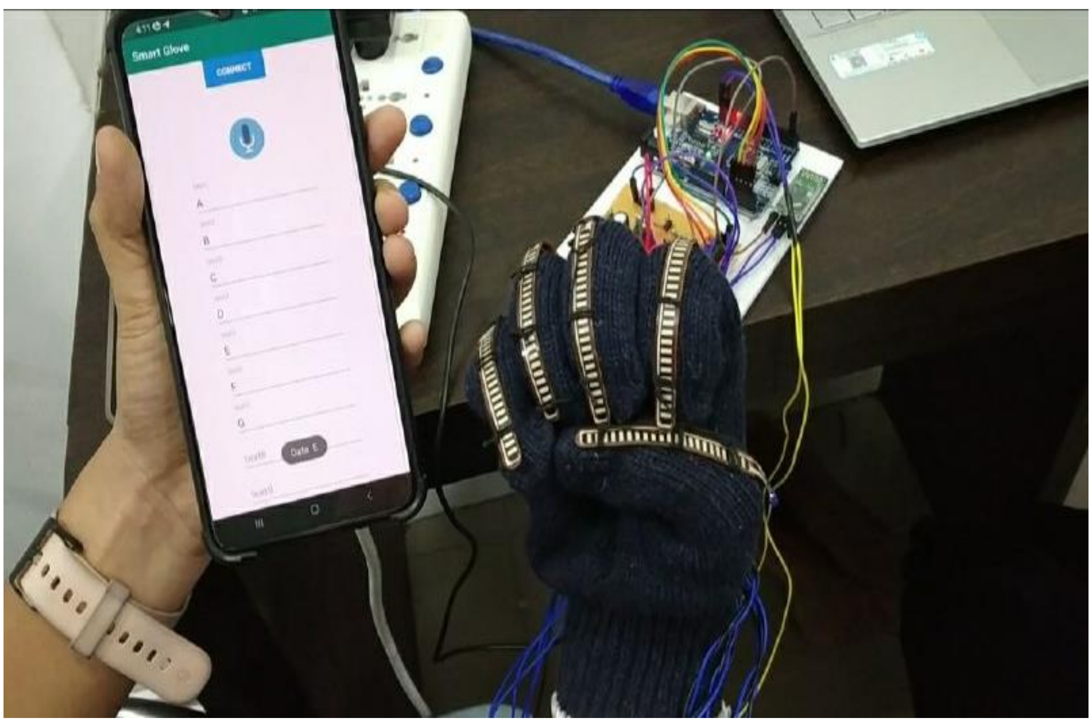


Fig 7.3 Gesture and output for A

The gesture manager is the principal part of the recognition system. It contains data to match with incoming data. The system tries to match incoming data with existing posture. The bend values of the fingers and for each posture definition the distance to the current data is calculated. From the convenience of simple flex sensors, a user is able to interact with others in more comfortable and easier manner. This makes it possible for the user to not only interact with their community but with others also and they can also live normal life.

CHAPTER 8

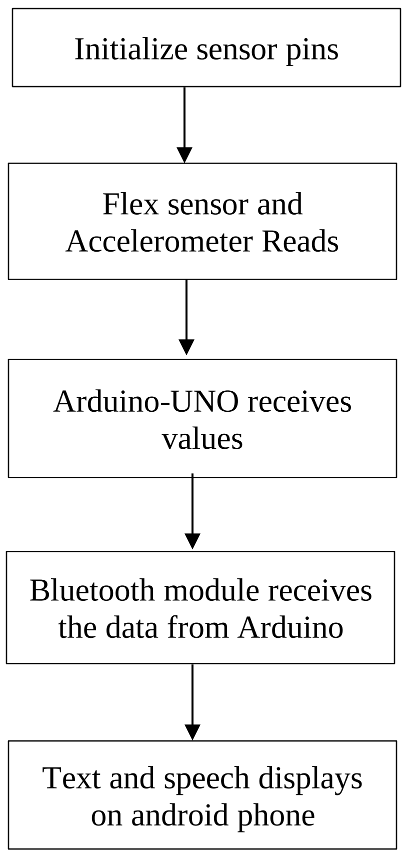
CONCLUSION

In this project, we developed a Smart-Glove for supporting deaf and mute people in communicating with normal people who are not familiar with sign language. The Smart-Glove is able to connect to Android mobile and facilitate exchange of messages. Whereas the android application is able to receive messages from Smart-Glove and the Smart-Glove is able to send corresponding values to the application through the Bluetooth module. The Smart-Glove is light, cheap, easy to use and no risk. We believe that the project is an effective and very useful for deaf and mute people if they know the sign language where they can communicate with their families and people around them.

As future scope of the project, the system may be extended to support other languages, can include home automation and the system can use several ways to communicate. It can use Wi-Fi connection, which enables a faster connection and better range from the base station or GSM module (Global System for Mobile communication). GSM is the most widespread and it's a cellular technology used for transmitting mobile data services, the most obvious advantage of it is widespread use throughout the world. Besides that, by making use of the GPS, it is possible to locate the position of user and it will make easier to locate them in the case of emergency.

APPENDIX

FLOW CHART:



PROGRAM:

```
int flex1 = A0;

int flex2 = A1;
int flex3 = A2;
int flex4 = A3;
int flex5 = A4;
bool flag1 = 0, flag2 = 0, flag3 = 0, flag4 = 0, flag5 = 0, flag6 = 0, flag7 = 0, flag8 = 0, flag9 = 0,
flag10 = 0, flag11 = 0, flag12 = 0, flag13 = 0, flag14 = 0;
bool flag15 = 0, flag16 = 0, flag17 = 0, flag18 = 0, flag19 = 0, flag20 = 0, flag21 = 0, flag22 = 0,
flag23 = 0, flag24 = 0, flag25 = 0, flag26 = 0;
bool flag = 0;
#include<Wire.h>
const int MPU_ADDR = 0x68;
int16_t AcX, AcY, AcZ, Tmp, GyX, GyY, GyZ;
void setup() {
  Serial.begin(9600);
  pinMode (flex1, INPUT);
  pinMode (flex1, INPUT);
  pinMode (flex2, INPUT);
  pinMode (flex3, INPUT);
  pinMode (flex4, INPUT);
  pinMode (flex5, INPUT);
  Wire.begin();
  Wire.beginTransmission(MPU_ADDR);
  Wire.write(0x6B);
  Wire.write(0);
  Wire.endTransmission(true);
  // put your setup code here, to run once:

}
```

```
void loop()
{
  int val1 = analogRead(flex1);
  int val2 = analogRead(flex2);
  int val3 = analogRead(flex3);
  int val4 = analogRead(flex4);
  int val5 = analogRead(flex5);
 Wire.beginTransmission(MPU_ADDR);
  Wire.write(0x3B);
  Wire.endTransmission(false);
  Wire.requestFrom(MPU_ADDR, 14, true);
  AcX = Wire.read() << 8 | Wire.read(); // 0x3B (ACCEL_XOUT_H) & 0x3C (ACCEL_XOUT_L)
  AcY = Wire.read() << 8 | Wire.read(); // 0x3D (ACCEL_YOUT_H) & 0x3E (ACCEL_YOUT_L)
  AcZ = Wire.read() << 8 | Wire.read(); // 0x3F (ACCEL_ZOUT_H) & 0x40 (ACCEL_ZOUT_L)
  float x = AcX / 16384.0;
  float y = AcY / 16384.0;
  float z = AcZ / 16384.0;
  //  Serial.print("Accelerometer: ");
  //  Serial.print("X=");
  //  Serial.print("g Y=");
  //  Serial.print("g Z=");
  //  Serial.println("g");
//  Serial.println(val1);
//  Serial.println(val2);
//  Serial.println(val3);
//  Serial.println(val4);
//  Serial.println(val5);
//  Serial.println("///////////////////////////////");
  //  Serial.println(x); // Conversion factor for 16-bit values
```

```
  //  Serial.println(y);
  //  Serial.println(z);
if (val1 > 750 && val1 < 770 && val2 > 745 && val2 < 760 && val3 > 745 && val3 < 760 && val4 >
740 && val4 < 760 && val5 > 710 && val5 < 740 )
  {
 flag = 1;
  }
 if (val1 > 730 && val1 < 790 && val2 > 800 && val2 < 850 && val3 > 770 && val3 < 820 && val4 >
780 && val4 < 830 && val5 > 700 && val5 < 790 )
  {
    flag1 = 1;
  }
 if (val1 > 775 && val1 < 820  && val2 > 730 && val2 < 770 && val3 > 740 && val3 < 770 && val4 >
740 && val4 < 780 && val5 > 700 && val5 < 760 )
  {
    flag2 = 1;
  }
  if (val1 > 770 && val1 < 795 && val2 > 770 && val2 < 815 && val3 > 775 && val3 < 800 && val4 >
760 && val4 < 810 && val5 > 700 && val5 < 750)
    {
      flag3 = 1 ;
    }
 if (val1 > 770 && val1 < 810 && val2 > 740 && val2 < 780 && val3 > 790 && val3 < 820 && val4 >
790 && val4 < 840 && val5 > 700 && val5 < 760)
  {
    flag4 = 1 ;
  }
 if (val1 > 780 && val1 < 820 && val2 > 820 && val2 < 850 && val3 > 790 && val3 < 830 && val4 >
780 && val4 < 830 && val5 < 900)
  {
    flag5 = 1 ;
  }
```

```
 if (val1 > 770 && val1 < 805 && val2 > 790 && val2 < 820 && val3 > 730 && val3 < 750 && val4 >
720 && val4 < 790 && val5 < 900)
  {
    flag6 = 1 ;
  }
  if (val1 > 740 && val1 < 835 && val2 > 740 && val2 < 800 && val3 > 750 && val3 < 810 && val4 >
760 && val4 < 810 && val5 > 700 && val5 < 780 )
    {
      flag7 = 1 ;
    }
 if (val1 > 780 && val1 < 810 && val2 > 750 && val2 < 770 && val3 > 730 && val3 < 770 && val4 >
760 && val4 < 810 && val5 > 700)
  {
    flag8 = 1 ;
  }
if (val1 > 805 && val1 < 830 && val2 > 800 && val2 < 860 && val3 > 765 && val3 < 840 && val4 >
740 && val4 < 860 && val5 > 710 && val5 < 780)
  {
    flag9 = 1 ;
  }
if (val1 > 770 && val1 < 795 && val2 > 810 && val2 < 860 && val3 > 770 && val3 < 810 && val4 >
770 && val4 < 820 && val5 > 720 && val5 < 760)
    {
      flag10 = 1 ;
    }
 if (val1 > 720 && val1 < 765 && val2 > 720 && val2 < 760 && val3 > 730 && val3 < 780 && val4 >
750 && val4 < 790 && val5 > 840 && val5 < 885)
    {
      flag11 = 1 ;
    }
if (val1 > 720 && val1 < 785 && val2 > 730 && val2 < 770 && val3 > 780 && val3 < 820 && val4 >
780 && val4 < 820 && val5 > 840 && val5 < 885)
    {
```

```
    flag12 = 1 ;
  }
 if (val1 > 780 && val1 < 825 && val2 > 800 && val2 < 840 && val3 > 760 && val3 < 800 && val4 >
780 && val4 < 840 && val5 > 840 && val5 < 885)
  {
    flag13 = 1 ;
  }
 if (val1 > 780 && val1 < 815 && val2 > 800 && val2 < 830 && val3 > 800 && val3 < 830 && val4 >
760 && val4 < 790 && val5 > 840 && val5 < 875)
  {
    flag14 = 1 ;
  }
 if (val1 > 760 && val1 < 790 && val2 > 790 && val2 < 820 && val3 > 760 && val3 < 790 && val4 >
765 && val4 < 795 && val5 > 830 && val5 < 870)
  {
    flag15 = 1 ;
  }
 if (val1 > 740 && val1 < 775 && val2 > 730 && val2 < 760 && val3 > 730 && val3 < 760 && val4 >
740 && val4 < 780 && val5 > 820 && val5 < 865)
  {
    flag16 = 1 ;
  }
 if (val1 > 740 && val1 < 765 && val2 > 740 && val2 < 770 && val3 > 770 && val3 < 800 && val4 >
780 && val4 < 810 && val5 > 830 && val5 < 855)
  {
    flag17 = 1 ;
  }
 if (val1 > 770 && val1 < 795 && val2 > 730 && val2 < 750 && val3 > 755 && val3 < 775 && val4 >
780 && val4 < 815 && val5 > 850 && val5 < 885)
  {
    flag18 = 1 ;
  }
```

```
if (val1 > 780 && val1 < 815 && val2 > 840 && val2 < 865 && val3 > 780 && val3 < 820 && val4 > 790 && val4 < 820 && val5 > 850 && val5 < 885)
  {
   flag19 = 1 ;
  }
if (val1 > 740 && val1 < 785 && val2 > 790 && val2 < 840 && val3 > 765 && val3 < 785 && val4 > 755 && val4 < 785 && val5 > 835 && val5 < 870)
  {
   flag20 = 1 ;
  }
if (val1 > 760 && val1 < 795 && val2 > 730 && val2 < 760 && val3 > 735 && val3 < 760 && val4 > 775 && val4 < 800 && val5 > 820 && val5 < 855)
  {
   flag21 = 1 ;
  }
if (val1 > 760 && val1 < 795 && val2 > 730 && val2 < 760 && val3 > 735 && val3 < 760 && val4 > 775 && val4 < 800 && val5 > 820 && val5 < 855)
  {
   flag22 = 1 ;
  }
if (val1 > 765 && val1 < 795 && val2 > 735 && val2 < 760 && val3 > 735 && val3 < 760 && val4 > 720 && val4 < 760 && val5 > 870 && val5 < 910)
  {
   flag23 = 1 ;
  }
if (val1 > 790 && val1 < 820 && val2 > 790 && val2 < 820 && val3 > 790 && val3 < 820 && val4 > 790 && val4 < 820 && val5 > 840 && val5 < 885)
  {
   flag24 = 1 ;
  }
if (val1 > 740 && val1 < 775 && val2 > 810 && val2 < 845 && val3 > 770 && val3 < 790 && val4 > 775 && val4 < 790 && val5 > 805 && val5 < 825)
  {
```

```
      flag25 = 1 ;
    }
 if (val1 > 765 && val1 < 800 && val2 > 720 && val2 < 780 && val3 > 780 && val3 < 820 && val4 >
770 && val4 < 810 && val5 > 730 && val5 < 775)
    {
      flag26 = 1 ;
    }
if (flag1 == 1 )
  {
    Serial.write('A');
    flag1 = 0;
    delay(3000);
  }
 if (flag2 == 1)
  {
    flag2 = 0;
    Serial.write('B');
    delay(3000);
  }
 if (flag3 == 1)
  {
    flag3 = 0;
    Serial.write('C');
    delay(3000);
  }
 if (flag4 == 1)
  {
    flag4 = 0;
    Serial.write('D');
    delay(3000);
```

```
}
if (flag5 == 1)
{
  flag5 = 0;
  Serial.write('E');
  delay(3000);
}
if (flag6 == 1)
{
  flag6 = 0;
  Serial.write('F');
  delay(3000);
}
if (flag7 == 1)
{
  flag7 = 0;
  Serial.write('G');
  delay(3000);
}
if (flag8 == 1)
{
  flag8 = 0;
  Serial.write('H');
  delay(3000);
}
if (flag9 == 1)
{
  flag9 = 0;
  Serial.write('I');
  delay(3000);
```

```
}
 if (flag10 == 1)
  {
   flag10 = 0;
   Serial.write ('J');
   delay(3000);
  }
if (flag11 == 1)
  {
   flag11 = 0;
   Serial.write('K');
   delay(3000);
  }
  if (flag12 == 1)
  {
   flag12 = 0;
   Serial.write('L');
   delay(3000);
  }
  if (flag13 == 1)
  {
   flag13 = 0;
   Serial.write('M');
   delay(3000);
  }
  if (flag14 == 1)
  {
   flag14 = 0;
   Serial.write('N');
   delay(3000);
```

```
}
if (flag15 == 1)
{
  flag15 = 0;
  Serial.write('O');
  delay(3000);
}
if (flag16 == 1)
{
  flag16 = 0;
  Serial.write('P');
  delay(3000);
}
if (flag17 == 1)
{
  flag17 = 0;
  Serial.write('Q');
  delay(3000);
}
if (flag18 == 1)
{
  flag18 = 0;
  Serial.write('R');
  delay(3000);
}
if (flag19 == 1)
{
  flag19 = 0;
  Serial.write('S');
```

```
  delay(3000);
 }
 if (flag20 == 1)
 {
  flag20 = 0;
  Serial.write('T');
  delay(3000);
 }
 if (flag21 == 1)
 {
  flag21 = 0;
  Serial.write('U');
  delay(3000);
 }
 if (flag22 == 1)
 {
  flag22 = 0;
  Serial.write('V');
  delay(3000);
 }
 if (flag23 == 1)
 {
  flag23 = 0;
  Serial.write('W');
  delay(3000);
 }
 if (flag24 == 1)
 {
  flag24 = 0;
  Serial.write('X');
```

```
  delay(3000);
 }
 if (flag25 == 1)
 {
  flag25 = 0;
  Serial.write('Y');
  delay(3000);
 }
 if (flag26 == 1)
 {
  flag26 = 0;
  Serial.write('Z');
  delay(3000);
 }
 }
```

REFERENCES

1. Zhibo Wang; Tengda Zhao; Jinxin Ma; Hongkai Chen; Kaixin Liu; Huajie Shao Department of Computer Science, University of Illinois Urbana-Champaign, Urbana, IL, USA “Hear Sign Language: A Real-Time End-to-End Sign Language Recognition System” IEEE Transactions on Mobile Computing (Volume: 21, Issue: 7, 01 July 2022).

2. Sevgi Z. Gurbuz; Ali C. Gurbuz; Evie A. Malaia; Darrin J. Griffin; Chris Crawford “A Linguistic Perspective on Radar Micro-Doppler Analysis of American Sign Language” 2020 IEEE International Radar Conference (RADAR).

3. Fady S. Botros; Angkoon Phinyomark; Erik J. Scheme “Electromyography-Based Gesture Recognition: Is It Time to Change Focus From the Forearm to the Wrist?” IEEE Transactions on Industrial Informatics (Volume: 18, Issue: 1, January 2022).

4. Hao Zhou , Charbel Tawk , and Gursel Alici , Member, IEEE (2022) “A 3D Printed Soft Robotic Hand With Embedded Soft Sensors for Direct Transition Between Hand Gestures and Improved Grasping Quality and Diversity VOL. 3” IEEE Transactions on Neural Systems and Rehabilitation Engineering (Volume: 30) 02 March 2022.

5. Zhicheng Yang; Shenggang Yan “Hand-Finger Pose Estimation Using Inertial Sensors, Magnetic Sensors and a Magnet” IEEE Sensors Journal (Volume: 21, Issue: 16, 15 August 2021)

6. Chen Zhu; Jianyu Yang; Zhanpeng Shao; Chunping Liu IEEE/CAA Journal of Automatica Sinica “Vision Based Hand Gesture Recognition Using 3D Shape Context” (Volume: 8, Issue: 9, September 2021)Leyla Khaleghi; Unal Artan; Ali Etemad; Joshua A. Marshall “Touchless Control of Heavy Equipment Using Low Cost Hand Gesture Recognition (Volume: 5, Issue: 1, March 2022)

7. Daniel Skomedal Breland; Aveen Dayal; Ajit Jha; Phaneendra K. Yalavarthy "Robust Hand Gestures Recognition Using a Deep CNN and Thermal Images" (Volume: 21, Issue: 23, 01 December 2021)

8. Chuanxin M. Niu;Jiayue Liu;Chih-Hong Chou;Manzhao Hao;Ning Lan "Evaluation of Model-Based Biomimetic Control of Prosthetic Finger Force for Grasp" IEEE Transactions on Neural Systems and Rehabilitation Engineering (Volume: 29), 20 August 2021

9. Jibo Bai; Baojiang Li; Haiyan Wang; Yutin Guo "Tactile Perception Information Recognition of Prosthetic Hand Based on DNN-LSTM" IEEE Transactions on Instrumentation and Measurement (Volume: 71) 11 July 2022

10. Q. Wang, B. Zheng, Q. Li, C. Shen, and Z. Ba, "Towards query-efficient adversarial attacks against automatic speech recognition systems," IEEE Trans. Inf. Forensics Secur., vol. 16, pp. 896–908, 2021

.

CERTIFICATE

*This is to certify that the Project Phase-II report entitled "**AUTOMATIC WASTE GATHERING AND DISPOSAL** " is a bonafide record of the EED416 Project Phase-II done by **ALBIN SAJI, NAKUL GANESH, ELTTIN JOY, VISHNU SREEKUMAR KM** under our guidance towards the partial fulfillment of the requirements for the award of the Degree of Bachelor of Technology in ELECTRICAL AND ELECTRONICS ENGINEERINGof the APJ Abdul Kalam Technological University through Vimal JyothiEngineering College, Chemperi.*

Place: Chemperi

Date:

Project Guide	**Project Coordinator**	**Head of Department**
Dr.G Justin Sunil Dhas	Mrs.Tintu George	Mrs.Laly James
Professor	Assistant Professor	Assistant Professor
Department of EE	Department of EE	Department of EE
VJEC Chemperi	VJEC Chemperi	VJEC Chemperi

DEPARTMENT OF ELECTRICAL & ELECTRONICS ENGG. ★ V. J.E.C CHEMPERI Date:...............

Signature Not Verified

Principal.VJEC

CERTIFICATE

This is to certify that the Project Phase-II report entitled ***"AUTONOMOUS UNDERWATER VEHICLE"*** *is a bonafide record of the EED 416 Project Phase-II done by* ***ALEENA JAISON , AMRITHA P, KIRAN JOSEPH, NOYAL JOSE*** *under our guidance towards the partial fulfillment of the requirements for the award of the Degree of Bachelor of Technology in Electrical And Electronics Engineering of the APJ Abdul Kalam Technological University through Vimal Jyothi Engineering College, Chemperi.*

Place: Chemperi

Date 02-06-2023

Project Guide	**Project Coordinator**	**Head of Department**
Ms.Shelma George	Ms.Tintu George	Ms.Laly James
Assistant Professor	Associate Professor	Associate Professor & HOD
Department of EEE	Department of EEE	Department of EEE
VJEC, Chemperi	VJEC,Chemperi	VJEC,Chemperi

CERTIFICATE

This is to certify that the Project Phase-II report entitled ***"DESIGN AND ANALYSIS OF BIDIRECTIONAL BATTERY CHARGER FOR ELECTRIC VEHICLE"*** *is a bonafide record of the EED 416 Project Phase-II done by* ***ABDUL JABBAR YOUNUS, ABHINAV V,ASWANTH RAMESHAN,ASWIN SURESH MS*** *under our guidance towards the partial fulfillment of the requirements for the award of the Degree of Bachelor of Technology in Electrical and Electronics Engineering of the APJ Abdul Kalam Technological University through Vimal JyothiEngineering College, Chemperi.*

Place: Chemperi

Date:

Project Guide	**Project Coordinator**	**Head of Department**
Mrs. Tintu George	Mrs. Tintu George	Mrs. Laly James
Associate Professor	Associate Professor	Associate Professor
Department of EEE	Department of EEE	Department of E&I
VJEC Chemperi	VJEC Chemperi	VJEC Chemperi

Signature Not Verified

Principal.VJEC

CERTIFICATE

This is to certify that the Project Phase-II report entitled ***"DUAL AXIS SOLAR TRACKING SYSTEM WITH HOME AUTOMATION"*** *is a bonafide record of the EED416 Project Phase-II done by* ***ASHLYN WILSON SASTHAMPADAVIL,GOKUL ARIYIL, MUHAMMAD HANAN AND P ROMA ULLAS*** *under our guidance towards the partial fulfillment of the requirements for the award of the Degree of Bachelorof Technology in Electrical and Electronics Engineeringof the APJ Abdul Kalam Technological University through Vimal Jyothi Engineering College, Chemperi.*

Place: Chemperi

Date:02.06.2023

Project Guide	**Project Coordinator**	**Head of Department**
Ms. Tinu Francis	Ms. Tintu George	Ms. Laly James
Assistant Professor	Associate Professor	Associate Professor
Department of EEE	Department of EEE	Department of EEE
VJEC Chemperi	VJEC Chemperi	VJEC Chemperi

DEPARTMENT OF ELECTRICAL & ELECTRONICS ENGG. V. J.E.C CHEMPERI Date:...............

Signature Not Verified

Principal.VJEC

CERTIFICATE

This is to certify that the Project Phase-II report entitled ***"ENERGY AUDIT OF VIMAL JYOTHI ENGINEERING COLLEGE"*** *is a bona fide record of the EED416 Project Phase-II done by* ***AJIN MATHEW JOSEPH, AJITH SAJI, AKHIL GEORGE, T K MUHAMMED ZIJAH*** *under our guidance towards the partial fulfillment of the requirements for the award of the Degree of Bachelor of Technology in Electrical and Electronics Engineering of the APJ Abdul Kalam Technological University through Vimal Jyothi Engineering College, Chemperi.*

Place: Chemperi

Date:

Project Guide	**Project Coordinator**	**Head of Department**
Mr. Prabin James	Mrs. Tintu George	Mrs. Laly James
Assistant Professor	Associate Professor	Associate Professor
Department of EEE	Department of EEE	Department of E&I
VJEC Chemperi	VJEC Chemperi	VJEC Chemperi

DEPARTMENT OF ELECTRICAL & ELECTRONICS ENGG. ★ V. J.E.C CHEMPERI Date:..................

Signature Not Verified

Principal.VJEC

DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

VIMAL JYOTHI ENGINEERING COLLEGE

CHEMPERI, KANNUR

CERTIFICATE

This is to certify that the report entitled **"POWER GENERATION & HARVESTING OF ELECTRIC ENERGY USING PIEZOELECTRIC ENERGY"** submitted by **PRANAV TV (VML19EE028), SRADHA ALEX (VML19EE032), RENITHA RAMAKRISHNAN (VML19EE030)** to the APJ Abdul Kalam Technological University in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Electrical and Electronics Engineering through Vimal Jyothi Engineering College is a bonafide record of the project work carried out by him under our guidance and supervision.

Project Guide	**Project Coordinator**	**Head of Department**
Dr. Teena George	Mrs. Tintu George	Mrs. Laly James
Associate Professor	Associate Professor	Associate Professor
Department of EEE	Department of EEE	Department of EEE
VJEC Chemperi	VJEC Chemperi	VJEC Chemperi

DEPARTMENT OF ELECTRICAL & ELECTRONICS ENGG. V. J. E. C CHEMPERI Date:..........

Signature Not Verified

Principal.VJEC

DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

VIMAL JYOTHI ENGINEERING COLLEGE

CHEMPERI, KANNUR

CERTIFICATE

This is to certify that the report entitled **"POWER GENERATION & HARVESTING OF ELECTRIC ENERGY USING PIEZOELECTRIC ENERGY"** submitted by **PRANAV TV (VMLEE028), SRADHA ALEX (VML19EE032), RENITHA RAMAKRISHNAN (VML19EE030)** to the APJ Abdul Kalam Technological University in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Electrical and Electronics Engineering through Vimal Jyothi Engineering College is a bonafide record of the project work carried out by him under our guidance and supervision.

Project Guide	Project Coordinator	Head of Department
Dr. Teena George	Mrs. Tintu George	Mrs. Laly James
Associate Professor	Associate Professor	Associate Professor
Department of EEE	Department of EEE	Department of EEE
VJEC Chemperi	VJEC Chemperi	VJEC Chemperi

DEPARTMENT OF ELECTRICAL & ELECTRONICS ENGG. ★ V. J. E. C CHEMPERI Date:...............

Signature Not Verified

Principal.VJEC

CERTIFICATE

This is to certify that the Project Phase-II report entitled ***"SMART ENERGY METER WITH BILL PREDICTION USING AI"*** *is a bonafide record of the EED416 Project Phase-II done by* ***HRITHWIK SREEJITH,SAYOOJ DEVAN MB,JOHN TOMY*** *under our guidance towards the partial fulfillment of the requirements for the award of the Degree of Bachelor of Technology in Electrical and Electronics Engineering of the APJ Abdul Kalam Technological University through Vimal JyothiEngineering College, Chemperi.*

Place: Chemperi

Date:02.06.2023

Project Guide	**Project Coordinator**	**Head of Department**
Ms. Athira M Thomas	Ms. Tintu George	Ms. Laly James
Assistant Professor	Associate Professor	Associate Professor
Department of EEE	Department of EEE	Department of EEE
VJEC Chemperi	VJEC Chemperi	VJEC Chemperi

DEPARTMENT OF ELECTRICAL & ELECTRONICS ENGG.
V. J.E.C
CHEMPERI
Date:..................

Signature Not Verified

Principal.VJEC

DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that the project report entitled **"A SMART GLOVE THAT TRANSLATES SIGN LANGUAGE INTO TEXT AND SPEECH"** *is a bonafide record of the EED 416 Project Phase II Report preliminary done by* ***DILNA MARIA SHIBU, DWITHI SHIVAKUMAR and VAISHALI PRABHAKARAN*** *under our guidance towards the partial fulfilment of the requirements for the award of the Degree of Bachelor of technology in Electrical& Electronics Engineering of the APJ Abdul Kalam Technological University through Vimal Jyothi Engineering College, Chemperi, Kannur.*

R. Senthilkumar 17/6/23

PROJECT GUIDE	PROJECT COORDINATORS	HEAD OF THE DEPARTMENT
Dr. R Senthilkumar	**Mrs. Tintu George**	**Mrs. Laly James**
Professor	Associate Professor	Professor
Department of EEE	Department of EEE	Department of EEE
VJEC	VJEC	VJEC

DEPARTMENT OF ELECTRICAL & ELECTRONICS ENGG. ★ V. J.E.C CHEMPERI Date:...........

Signature Not Verified

Principal.VJEC

CERTIFICATE

This is to certify that the Project Phase-II report entitled **"SMART TRAFFIC CONTROL SYSTEM USING RASPBERRY PI"** *is a bonafide record of the EED416 Project Phase-II done by* **ABHILASH JOSEPH, ALEN VARGHESE, ASWIN K, and JOEL M JACOB** *under our guidance towards the partial fulfillment of the requirements for the award of the Degree of Bachelor of Technology in Electrical and Electronics Engineering of the APJ Abdul Kalam Technological University through Vimal JyothiEngineering College, Chemperi.*

Place: Chemperi

Date:

Project Guide	Project Coordinator	Head of Department
Mr. Rojith K	Ms. Tintu George	Ms. Laly James
Assistant Professor	Associate Professor	Associate Professor & HOD
Department of EEE	Department of EEE	Department of EEE
VJEC Chemperi	VJEC Chemperi	VJEC Chemperi

Signature Not Verified

Principal.VJEC