A

Mini Project Report on

**"DEVELOPMENT OF SIGN LANGUAGE TRANSLATOR BASED ON GESTURES-TO-WORDS USING IOT".**

**Submitted in partial fulfillment of the requirements for the award of degree**

**BACHELOR OF TECHNOLOGY**

**in**

**ELECTRICAL AND ELECTRONICS ENGINEERING**

**by**


| | |
|---|---|
| **G. PRAVEEN** | **23E45A0257** |
| **K. PRASANTH** | **23E45A0233** |
| **B. SRISILAM** | **23E45A0289** |
| **V. SRIKANTH** | **23E45A0273** |

Under the Esteemed Guidance of

**Mr.M.V. Santhosh kumar** M. Tech.

**Assistant Professor, Department of EEE**

**DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING**

**SREE DATTHA INSTITUTE OF ENGINEERING & SCIENCE**

**(Accredited by NAAC, Approved by AICTE, Affiliated to JNTU, Hyderabad)**


**(2024-2025)**

**SREE DATTHA INSTITUTE OF ENGINEERING & SCIENCE**

**(Accredited by NAAC, Approved by AICTE, Affiliated to JNTU, Hyderabad)**

**Sheriguda, Ibrahimpatnam, R.R. Dist. Hyderabad.**

## Department of Electrical and Electronics Engineering

# CERTIFICATE

**Certified that this is a bonafide record of the dissertation work entitled, "DEVELOPMENT OF SIGN LANGUAGE TRANSLATOR BASED ON GESTURES-TO-WORDS USING IOT", done by** G. PRAVEEN (23E45A0257), K. PRASANTH (23E45A0233), B. SRISILAM (23E45A0289) and E. SRIKANTH (23E45A0273) **submitted to the faculty of Electrical & Electronics Engineering, in partial fulfillment of the requirement of Degree of BACHELOR OF TECHNOLOGY in ELECTRICAL AND ELECTRONICS ENGINEERING from SREE DATTHA INSTITUTE OF ENGINEERING & SCIENCE, Hyderabad during the period 2024-2025.**

**Internal Guide**
Mr.M.V. Santhosh Kumar

**Head of the Department**
Dr. P. UMAPATHI REDDY

**Internal Examiner**

**External Examiner**

**SREE DATTHA INSTITUTE OF ENGINEERING AND SCIENCE**
**(Accredited by NAAC, Approved by AICTE, Affiliated to JNTU Hyderabad)**
**Sheriguda, Ibrahimpatnam, R.R. Dist.**

DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

# DECLARATION

We U.SAISRI (23E45A0213) , E.SAIPRAKASH (23E45A0208), D.SHIVAJI (22E45A0215) and B. SANTHOSH (22E45A0221)hereby declare that the mini project report titled "**DEVELOPMENT OF SIGN LANGUAGE TRANSLATOR BASED ON GESTURES-TO-WORDS USING IOT**" under the guidance of Mr.M.V.Santhosh Kumar, Assistant Professor, Department of Electrical and Electronics Engineering, **Sree Dattha Institute of Engineering & Science, Ibrahimpatnam** is submitted in partial fulfillment of the requirement for the award of the degree of B. Tech. in Electrical and Electronics Engineering is a record of bonafide work carried out by us and the results embodied in this project have not been reproduced or copied from any source. The results embodied in this project report have not been submitted to any other University or Institute for the award of any Degree or Diploma.

| | |
|---|---|
| **G. PRAVEEN** | **23E45A0257** |
| **K. PRASANTH** | **23E45A0233** |
| **B. SRISILAM** | **23E45A0289** |
| **V. SRIKANTH** | **23E45A0273** |

# ACKNOWLEDGMENT

We would like to thank Chairman **Sri. G. Panduranga Reddy** garu, Vice Chairman **Sri G. N. V. Vibhav Reddy** garu and Director **Sri. G. Vikram Reddy** garu for providing all the facilities to complete our project with in time.

We would like to express our deep sense of gratitude to **Dr. S. Venkata Achuta Rao**, Dean-Academics, Sree Dattha Institute of Engineering & Science and **Dr. E. Madhusudhana Reddy**, Principal, Sree Dattha Institute of Engineering and Science, for their continuous effort in creating a competitive environment in our college and encouraging through this course.

We would like to give a special vote of thanks to **Dr. P. Umapathi Reddy**, Vice Principal & Head of the Department, Electrical and Electronics Engineering for his unique way of inspiring students through clarity of thoughts, enthusiasm, and care. His constant encouragement and assistance were very helpful and made our efforts a success.

Working and writing our thesis in exchange at Sree Dattha Institute of Engineering and Science was a great opportunity and we would like to thank from bottom of our heart to our internal guide **Mr.M.V. Santhosh Kumar**, Assistant Professor, Electrical and Electronics Engineering for providing it to us. There was never such a resourceful and enriching time in our life. We truly appreciate and value him esteemed guidance and encouragement from the beginning to the end of this thesis.

We would like to thank the Teaching & Non -Teaching staff of department of Electrical & Electronics Engineering for sharing their knowledge with us.

Finally, we deeply acknowledge all our friends and family members who have encouraged us during the preparation of our industry oriented mini project.

| | |
|---|---|
| **G. PRAVEEN** | **23E45A0257** |
| **K. PRASANTH** | **23E45A0233** |
| **B. SRISILAM** | **23E45A0289** |
| **V. SRIKANTH** | **23E45A0273** |

# ABSTRACT

Sign language is one of the world's languages, but instead of using speech or voice, it communicates by hand signals and other body parts. Speech and hearing-impaired people need to be able to communicate in sign language. A speech impaired person is someone who is unable to talk due to a reluctance or inability to speak. A person who has been hearing-impaired since birth or loss may develop later. Malaysian Sign Language (MSL) will be used by Malaysians speech and hearing-impaired population marto communicate, but it will be an obstacle as they need to converse with normal people. Currently, sign language translators are mostly focused on general sign language and American Sign Language (ASL).

This project aimed to develop a sign language translator system based on MSL with Node MCU ESP8266, flex sensors and MPU6050 accelerometer gyroscope sensor module that would enable speech and hearing-impaired people to communicate with regular people who are unfamiliar with sign language. It used a data glove approach method that can convert a hand gesture into words. In this context, the flex sensors defined the sensor values based on finger bending, meanwhile, the accelerometer gyro sensor module defined the hand position.

This project implements IoT technology to display the meaning of sign language gestures on a smartphone. A Node MCU ESP8266 module was also used as a source IoT platform by integrating with the Blynk application in the smartphone to display the output of the system. The integration of the Node MCU ESP8266 makes the system capable of transferring data via Wi-Fi. The data captured from the sensors is processed, and the gesture's meaning is displayed on a smartphone screen via the Blynk app, offering a real-time translation of the sign into text. This IoT functionality is crucial as it allows the system to be portable, user-friendly, and flexible for various use cases, especially in everyday situations where a smartphone can easily be used to read the translation.

# INDEX

# LIST OF TABLES

# LIST OF ABBREVATIONS

| | | |
|---|---|---|
| **LCD** | **:** | Light crystal diode |
| **LED** | **:** | Light emitting diode |
| **RPS** | **:** | Regulated power supply |
| **NODE MCU** | **:** | Node micro controller unit |

# CHAPTER – 1

## INTRODUCTION

Communication is a fundamental aspect of human interaction, enabling individuals to express thoughts, emotions, and needs. However, for speech and hearing- impaired individuals, communication becomes a significant challenge, especially when interacting with people who are unfamiliar with sign language. Sign language serves as a primary mode of communication for the deaf and mute community, using hand gestures, facial expressions, and body movements instead of spoken words.

In Malaysia, Malaysian Sign Language (MSL) is widely used by speech and hearing-impaired individuals. Despite its importance, many people outside the community do not understand MSL, creating a communication gap that hinders daily interactions, education, and employment opportunities. Current technological advancements have introduced sign language recognition systems, but most focus on general sign language or American Sign Language (ASL), leaving a need for localized solutions tailored to specific regional sign languages like MSL.

This project aims to develop a gesture-to-word sign language translator based on Internet of Things (IoT) technology to bridge this communication gap. The system intilizes a data glove approach embedded with flex sensors and an MPU6050 accelerometer-gyroscope sensor module to capture and interpret hand gestures. The flex sensors detect finger bending, while the accelerometer and gyroscope measure hand orientation and motion. The collected gesture data is processed using Node MCU ESP8266, which transmits the recognized signs to a smartphone interface via the Blynk IoT platform, displaying the translated words in real time.

# CHAPTER - 2

## LITERATURE AND SURVEY

Sign language is a crucial mode of communication for individuals with speech and hearing impairments, enabling them to express themselves using hand gestures and body movements. However, challenges arise when communicating with individuals who do not understand sign language. Various research efforts have been made to get this communication gap by developing sign language translation systems that convert hand gestures into text or speech.

Several studies have explored different technologies and methodologies for translating sign language into text. Early systems relied on image processing and computer vision techniques, where cameras captured hand movements, and machine learning algorithms recognized gestures. However, these systems had limitations such as lighting dependency, complex background processing, and high computational costs.

To overcome these challenges, researchers shifted towards wearable sensor-based approaches, such as data gloves embedded with flex sensors, accelerometers, and gyroscopes. These sensor-based systems provide higher accuracy in gesture recognition, as they can precisely track finger bending and hand orientation. For example, studies have demonstrated that flex sensors can effectively detect the degree of finger bending, while MPU6050 accelerometer and gyroscope modules can determine hand movements and orientation in 3D space.

In recent years**,** IoT-based solutions have gained prominence in sign language translation. Systems integrated with Wi-Fi-enabled microcontrollers like the Node MCU ESP8266 allow real-time data transmission to smartphones or cloud platforms. Researchers have successfully implemented IoT platforms like Blynk and Firebase to display translated text on a mobile application, making the system more accessible and user-friendly.

# CHAPTER-3

## INTRODUCTION TO EMBADDED SYSTEMS

Existing methodologies for sign language translation include computer vision-based and wearable sensor-based approaches, each with advantages and limitations in accuracy and usability. Hybrid models and AI-NLP integration enhance performance but increase system complexity

1. **Computer Vision-Based Approach:** Uses cameras and deep learning models (e.g., CNNs) to recognize hand gestures and translate them into text or speech.

2. **Wearable Sensor-Based Approach:** Utilizes gloves equipped with flex sensors, accelerometers, and gyroscopes to track hand and finger movements for accurate gesture recognition.

3. **EMG (Electromyography)-Based Approach:** Employs EMG sensors to detect muscle activity in the hand and forearm, interpreting gestures based on muscle contractions.

4. **Hybrid Approach:** Sensor + Vision-Based Combines wearable sensors and computer vision to enhance accuracy by capturing both movement data and visual context.

5. **AI-Based NLP Integration for Gesture Translation:** Integrates Natural Language Processing (NLP) and AI models to convert recognized gestures into structured, meaningful sentences.

## 3.1 Proposed Methodology:

The proposed methodology uses flex sensors and an MPU6050 accelerometer- gyroscope for gesture recognition, with Node MCU ESP8266 enabling real-time IoT- based translation via a smartphone app. The system is programmed using Embedded C in Arduino IDE, displaying output on an I2C LCD and Blynk app, ensuring accessibility and accuracy.

**3.1.1 Sensor-Based Gesture Recognition:** Utilizes flex sensor sand anMPU6050 accelerometer-gyroscope to detect finger bending and hand movements for accurate gesture identification**.**

**3.1.2 IoT Integration with Node MCU ESP8266:** Employs Node MCU ESP8266 to transmit gesture data to a smartphone via Wi-Fi, displaying translated text using the Blynk application.

**3.1.3 Embedded System Development:** Programs the system using Embedded C in Arduino IDE, ensuring efficient data processing and real-time translation**.**

**3.1.4 Real-Time Display and User Interface:** Outputs recognized gestures on an I2C LCD display and a smartphone app, enhancing accessibility for users unfamiliar with sign language.

**3.1.5 System Testing and Performance Evaluation**: Conducts unit testing, integration testing, and user testing to validate accuracy, response time, and usability, making improvements based on feedback.

# CHAPTER-4

# PROJECT DISCRIPTION

## 4.1 BLOCK DIAGRAM

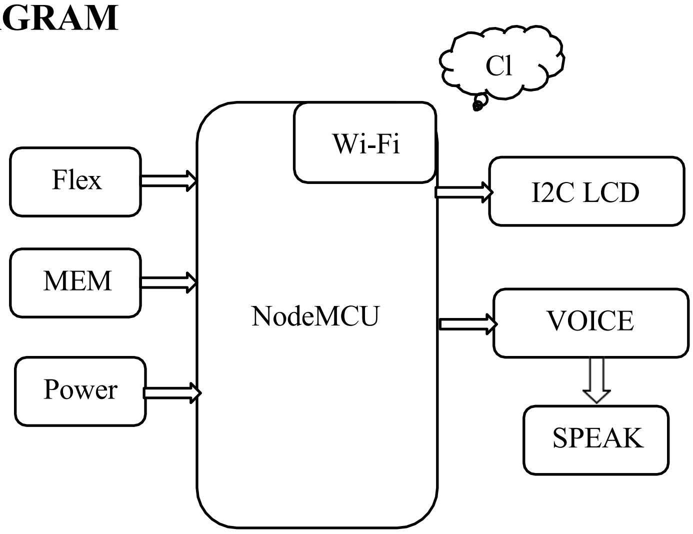



**Figure4.1.1BlockDiagram**

Fig 4.1.1 shows block diagram of "**DEVELOPMENT OF A SIGN LANGUAGE TRANSLATOR BASED ON GESTURES TOWORDS USING IOT**". The block diagram of the Sign Language Translator System illustrates the flow of data from gesture detection to output display. It provides a clear representation of how different hardware and software components interact to achieve real-time sign language translation. The system primarily consists of sensors, a microcontroller (Node MCU ESP8266), an LCD display, and IoT connectivity for smartphone integration.

At the input stage, flex sensors are attached to the fingers to detect their bending positions, while the MPU6050 accelerometer-gyroscope module captures hand orientation and movement. These sensors generate analog and digital signals corresponding to different hand gestures

The processing unit, Node MCU ESP8266, acts as the core of the system, converting raw sensor values into meaningful words. It is programmed using Embedded C in the Arduino IDE, ensuring efficient data handling and real-time translation. The microcontroller also enables Wi-Fi connectivity, allowing the processed data to display not only on an I2C LCD screen but also on a smartphone via the Blynk application.

At the output stage, the translated words are displayed on an LCD screen, providing instant feedback. Simultaneously, the Blynk app on a smartphone receives the translated text, making it accessible for communication with non-sign language users. This IoT-based approach enhances usability, as the system can be remotely monitored and updated.

## 4.2 WORKING PRINCIPLE:

The Sign Language Translator functions as follows:

**1. Hand Gesture Detection**

The user wears a data glove equipped with flex sensors and anMPU6050 accelerometer gyroscope sensor.

Flex sensors detect finger bending, while the MPU6050 captures hand position and movement.

**2. Data Processing**

The Node MCUESP8266 microcontroller collects and processes sensor values. The system maps

The detected gesture to a predefined word or phrase.

**3. IoT-Based Transmission**

The processed data is sent wirelessly via Wi-Fi to the Blynk application on a smartphone.

The system also supports an I2CLCD for direct display of the recognized word.

**4. Output Display**

The interpreted gesture is converted into readable text.

The text is displayed on the smartphone and LCD, enabling communication between speech- and hearing-impaired individuals and non-sign-language users.

This system bridges the communication gap by converting Malaysian Sign Language (MSL) gestures into words using IoT technology.

## 4.3 System architecture and design implementation

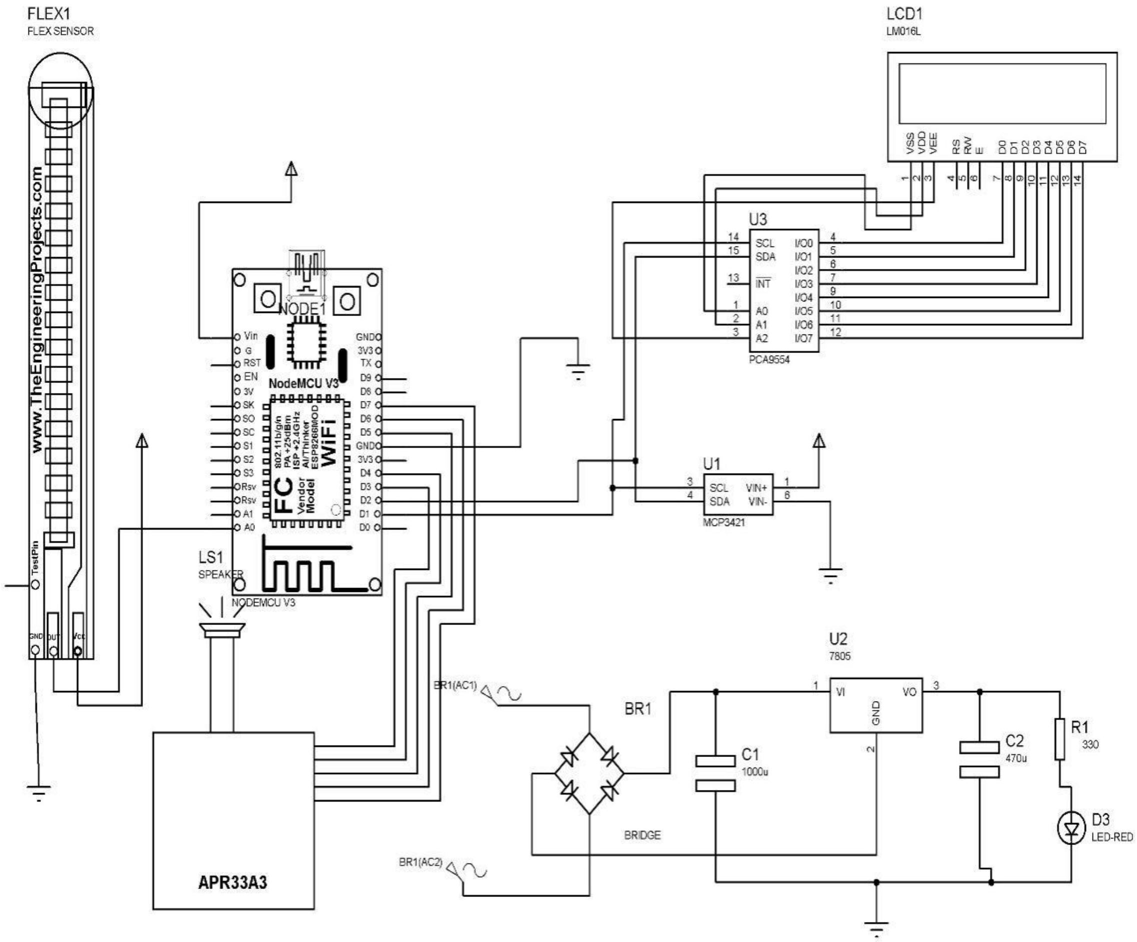



**Fig4.3: System Architecture**

This circuit schematic represents a flex sensor-based system using a Node MCU ESP8266, an LCD display, and power regulation components. The power supply section consists of a transformer that steps down AC voltage, a bridge rectifier to convert AC to DC, and voltage regulators (7805) ensuring a stable 5V output. Capacitors help in filtering, while an LED indicator confirms power availability. The flex sensor detects bending and changes its resistance accordingly, which is then processed by the MCP3421 ADC module for precise analog-to-digital conversion. The Node MCU ESP8266 microcontroller reads this data and communicates via I2C protocol with a 16x2 LCD display using a PCF8574 I2C module to minimize pin usage. A potentiometer is included to adjust the display contrast. The system functions by detecting hand movements through the flex sensor, converting these signals into digital values, and displaying the corresponding information on the LCD. This setup is useful for gesture-based communication, sign language translation, and IoT applications.

This flex sensor-based system using a Node MCU ESP8266 is designed to capture hand movements and translate them into meaningful digital outputs, which are then displayed on an LCD screen. The circuit integrates various key components, including a power supply section, signal processing units, and a display module.

The power section consists of a step-down transformer, a bridge rectifier, and voltage regulators (7805), ensuring a steady5V DC supply for the circuit. Capacitors are used for filtering, while an LED indicator provides a visual confirmation of power availability. The flex sensor, a crucial component in this system, works by varying its resistance in response to bending, which is then processed by the MCP3421 ADC module for accurate conversion into digital data

# CHAPTER 5
# HARDWARE REQUIREMENTS

## 5.1 Node MCU(ESP8266-12E)

The Node MCU (*N*ode *M*icro *C*ontroller *U*nit) is an open-sources of software and hardware development environment built around an inexpensive System-on-a-Chip (SoC) called the ESP8266. The ESP8266, designed and manufactured by Es press if Systems, contains the crucial elements of a computer: CPU, RAM, networking (Wi Fi), and even a modern operating system and SDK. That makes it an excellent choice for Internet of Things (IoT) projects of all kinds.

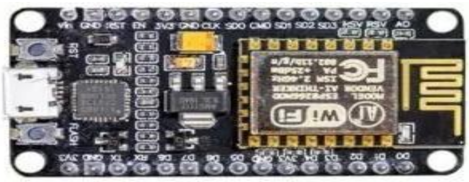


**Fig5.1: NodeMcu**

However, as a chip, the ESP8266 is also hard to access and use. You must solder wires, with the appropriate analog voltage, to its pins for the simplest tasks such as powering it on or sending a key stroke to the “computer” on the chip

## 5.2 Node MCU Specifications

The Node MCU is available in various package styles. Common to all the designs is the base ESP8266 core. Designs based on the architecture have maintained the standard 30-pin layout. Some designs use the more common narrow (0.9″) footprint, while others use a wide (1.1″) footprint – an important consideration to be aware of.

**Technical Specification of Node MCU (ESP8266-12E)**

| | |
|---|---|
| Microcontroller | Esp8266–32bit |
| Flash memory (ROM) | 4MB |
| RAM | 64Kb |
| Wi-Fi Built-In | 802.11b/g/n |
| Digital I/O pins | 11 |
| Analog Input Pins | 1 |
| UART | 1 |
| SPI | 1 |
| I2C | 1 |
| Operating voltage | 3.3v |
| Input Voltage | 4.5–10V |
| ADC voltage Range | 0-3.3V |

**Table : 5.2 technical specifications of node mcu**

## 5.3 Resistor

Resistors restrict the flow of electric current, for example a resistor is placed in series with a light-emitting diode (LED) to limit the current passing through the LED.Resistors may be connected either way round. They are not damaged by heat when soldering.

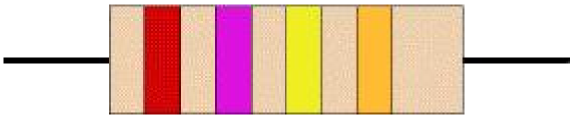


**Fig: 5.3.1resister**

## 5.4 Light Emitting Diode (LED)

LEDs are like tiny lightbulbs. Low energy consumption, small size, rapid switching and long lifespan makes them ideal for mobile devices and other low-power applications.

## 5.5 LED Working

Like an ordinary diode, the LED operates only in forward bias condition. When the LED is forward biased, the free electrons cross the PN junction and recombine with holes. Since these electrons fall from a higher to a lower energy level, they radiate energy in the form of photons (light).

## 5.6 REGULATED POWER SUPPLY(RPS)

In mains-supplied electronic systems the AC input voltage must be converted into a DC voltage with the right value and degree of stabilization. The common

DC voltages that are required to power up the devices are generally in the range Of 3V DC to 30 V DC.

**Regulated Power Supply Block diagram:**

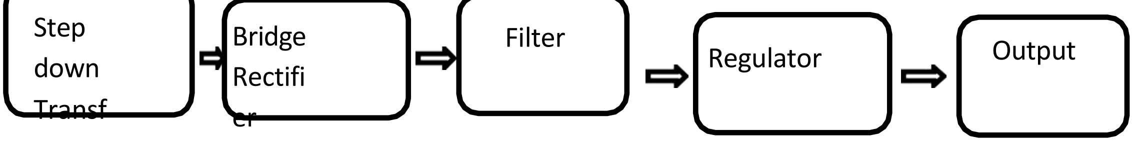



**Fig5.6.1: Block diagram of RPS**

**Transformer**

Transformers convert AC electricity from one voltage to another with little loss of power. Transformers work only with AC and this is one of the reasons why mains electricity is AC of110 VAC or 220 VAC I.e, it converts higher voltage at the input side(Vin) to a lower voltage at the output (V out).

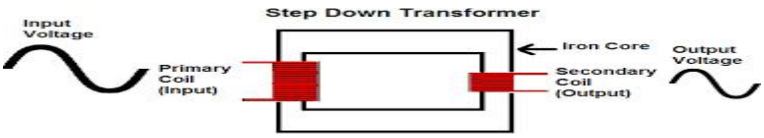



**Fig5.6.2: Transformer**

**REGULATOR**

This is a simple DC regulated supply project using 7805 voltage regulator to obtain a variable DC voltage range from 5V to 15V.

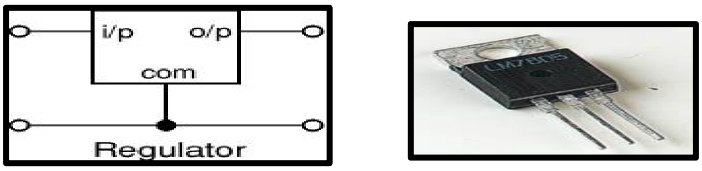



**Fig 5.6.3: Regulator**

**I2C16X2LCD**

The classic parallel LCD consumes a lot of pins on the Arduino. Even in the 4-bit mode, it requires at least 6digital I/O pins on the Arduino. So in many projects where you use the classic parallel LCD, you will run out of pins very easily.

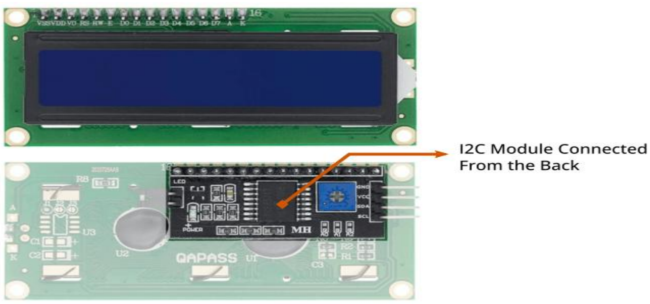
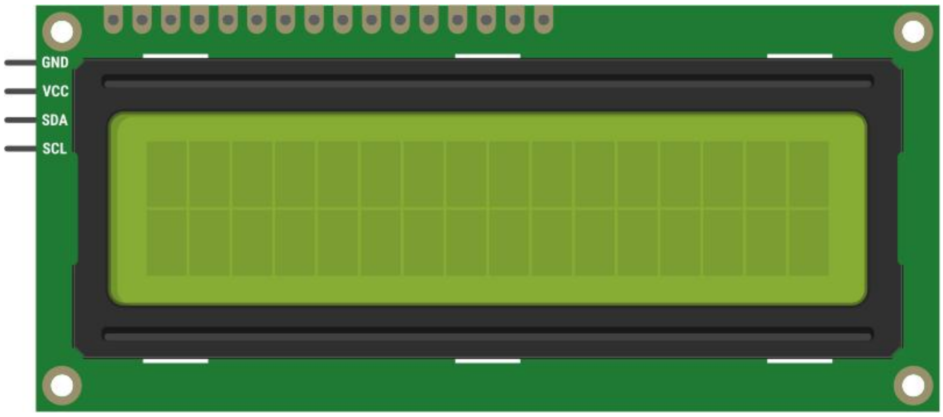



**Fig5.6.4 I2CLCD**

**I2CLCD Pin-out**

The I2CLCD display has four pins coming out from the I2CLCD adapter. These four pins are – Ground, VCC, SDA, and SCL.You can see the pin out of an I2CLCD display in the image below




**Wiring an I2CLCD Display to an ESP8266**

Connecting I2C LCD to ESP8266 is very easy as you only need to connect 4 pins. Start by connecting the VCC pin to the VIN on the ESP8266 and GND to ground. Now we are left with the pins which are used for I2C communication. We are going to use the default I2C pins (GPIO#4 and GPIO#5) of the ESP8266. Connect the SDA pin to the ESP8266's D2 (GPIO#4) and the SCL pin to the ESP8266's D1 (GPIO#5).

The following table lists the pin connections:

| I2CLCD | | ESP8266 |
|---|---|---|
| VCC | | VIN |
| GND | | GND |
| SCL | | D1 |

**Table 5.6.1 pin connections**

## Flex Sensor Working and Its Applications

We know that there are different types of sensors available in the market where each sensor can be used based on the application. Likewise, a bend sensor or flex sensor is one kind of sensor used to measure the quantity of bending otherwise deflection.

### What is a Flex Sensor?

A flex sensor is a kind of sensor which is used to measure the amount of defection otherwise bending. The carbon surface is arranged on a plastic strip as this strip is turned aside then the sensor’s resistance will be changed. Thus, it is also named abend sensor. As its varying resistance can be directly proportional to the quantity of turn thus it can also be employed like a goniometer.

**Fig5.6.5: Flex-sensor**

## ADXL345(3-AXIS ACCELEROMETER)

Inertial sensors are used to detect linear and rotational motion of an object. There are two types of inertial sensors – accelerometers that detect linear acceleration and gyro scopes that detect rotational motion. Accelerometers and gyros are widely used in.

### Interfacing ADXL345 accelerometer with Arduino Uno

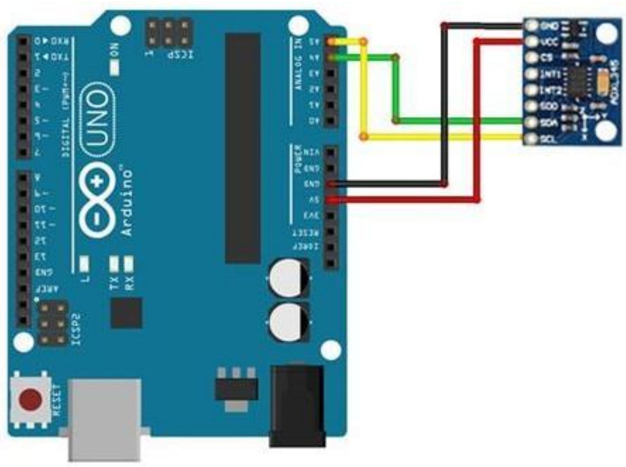



**Fig5.6.6: Arduino UNO**

# CHAPTER-6

# SOFTWARE REQUIREMENT

Arduino IDE (Integrated Development Environment) is required to program the NODE MCU board.

## 6.1 PROGRAMMING ARDUINO

Once Arduino IDE is installed on the computer, connect the board with computer using USB cable. Now open the Arduino IDE and

choose the correct board by selecting Tools>Boards>Arduino/Genuine Uno, and choose the correct Port by selecting Tools>Port. NODE MCU is programmed using Arduino programming language based on Wiring. To get it started with NODE MCU board and blink the built-in LED, load the example code by selecting Files>Examples>Basics>Blink.

## 6.2 ARDUINO– INSTALLATION

After learning about the main parts of the NODE MCU board, we are ready to learn how to set up the Arduino IDE. Once we learn this, we will be ready to upload our program on the Arduino board. In this section, we will learn in easy steps, how to set up the Arduino IDE on our computer and prepare the board to receive the program via USB cable.

**Step1: About Arduino board**

First you must have your Arduino board (you can choose your favorite board) and a USB cable. In case you use NODE MCU, Arduino Due milanove, Nano, Arduino Mega2560 ,or Diecimila, you will need a standard USB cable (A plug to B plug), the kind you would connect to a USB printer as shown in the following image.

**Fig.6.2.1 USB Cable**

In case you use Arduino Nano, you will need an A to Mini-B cable instead as shown in the following image

**Fig.6.2.2 A TO MINI-B CABLE**

**Step2: Download Arduino IDE Software**

You can get different versions of Arduino IDE from the Download page on the Arduino Official website. You must select your software, which is compatible with your operating system. After your file download is complete , unzip the file

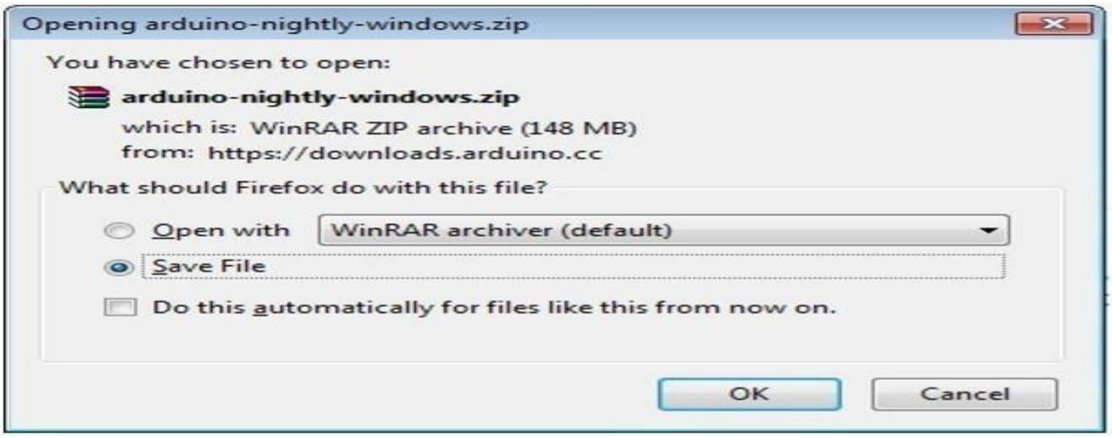



**Fig.6.2.3 Opening Arduino**

**Step 3: Power up your board**

The NODE MCU, Mega, Due milanove, Arduino, and nano technology automatically draw power,USB connection to the computer or an external power supply.

**Step4: Launch Arduino IDE**

After your Arduino IDE software is downloaded, you need to un zip the folder.

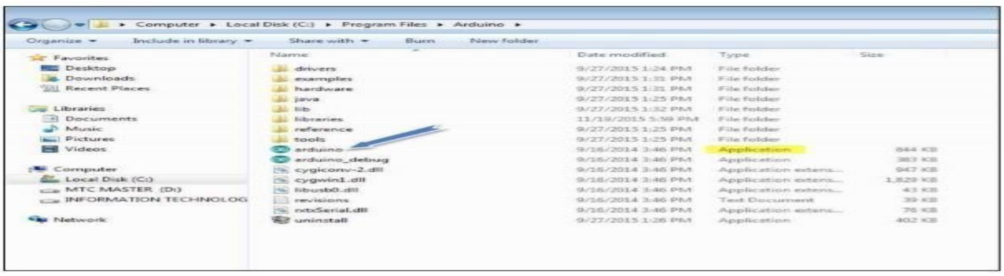


**Fig.6.2.4 Launching Arduino**

**Step 5: Open your first project**

Once the software starts, you have two options:

- Create a new project.
- Open an existing project example.

To create an e w project, select File-->New

**Fig.6.2.5 Arduino Software Opening New File**

To open an existing project example, select File ->Example->

Basics->Blink.

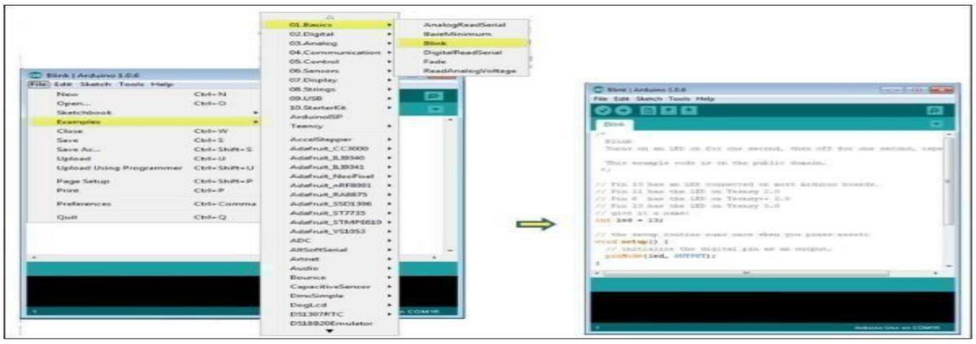


**Fig:6.2.6 Selecting File Type**

Here, we are selecting just one of the examples with the name Blink. It turns the LED on and off with some time delay. You can select any other example from the list.

**Step6: Select your Arduino board**

To avoid any error while uploading your program to the board, you must select the correct Arduino board name, which matches with the board connected to your computer.

Goto Tools->Board and select your board.

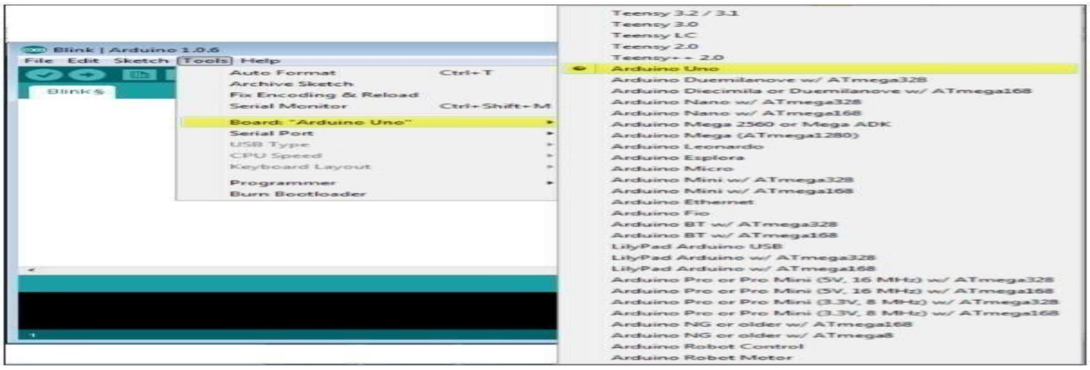


**Fig 6.2.7Selecting Board Type**

**Step7: Select your serial port**

Select the serial device of the Arduino board. Go to Tools ->Serial Port menu. This is likely to be COM3 or higher (COM1 and COM2 are usually reserved for hardware serial ports). and select that serial port.

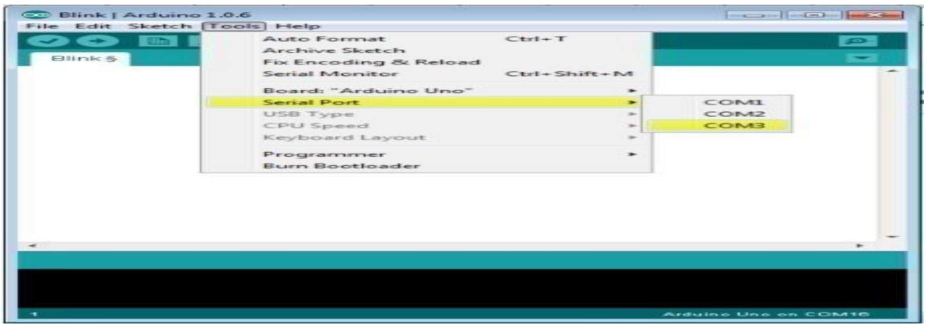


**Fig 6.2.8 Selecting Serial Port**

**Step8: Upload the program to your board**

Before explaining how we can upload our program to the board, we must demonstrate the function of each symbol appearing in the

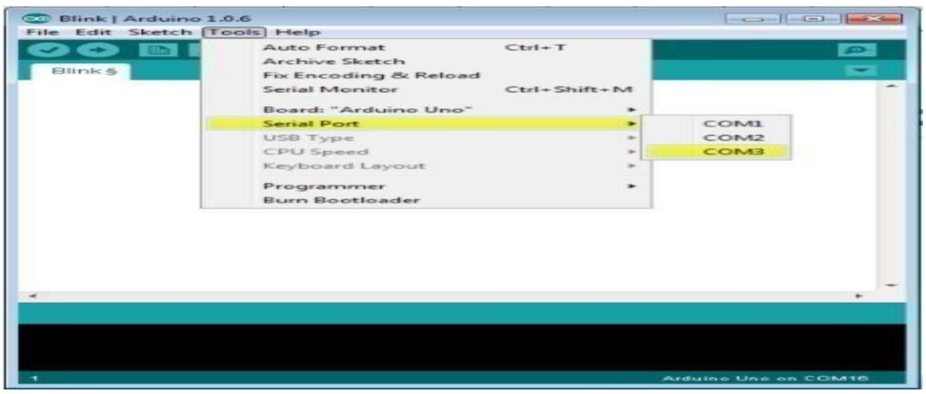
Arduino IDE toolbar


**Fig 6.2.9Uploading Program To Board**

A-Used to check if there is any compilation error.

B-Used to upload a program to the Arduino board.

C-Shortcut used to create a new sketch.




D-Used to directly open one of the example sketch.

E-Used to save your sketch.

F-Serial monitor used to receive serial data from the board and send the serial data to the board.

Now, simply click the "Upload" button in the environment. Wait a few seconds; you will see the RX and TXLEDs on the board, flashing. If the upload is successful, the message "Done uploading" will appear in the status bar.

We will study in depth, the Arduino program structure and we will learn more new terminologies used in the Arduino world. The Arduino software is opensource. The source code for the Java environment is released under the GPL and the C/C++ microcontroller libraries are under the LGPL. Sketch: The first new terminology is the Arduino program called "sketch". In this tutorial, we will learn about the Arduino software program, step by step, and how we can write the program without any syntax or compilation error. Let us start with the Structure. Software structure consist of two main Function:

- Setup( ) function
- Loop( ) function

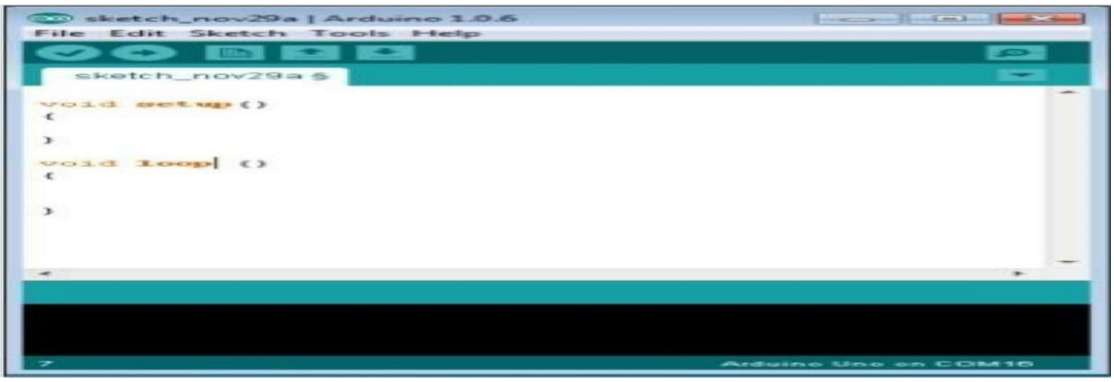



**Fig 6.2.10 Program Structure**

```
Void setup ( )
{

}
```

PURPOSE: The setup () function is called when a sketch starts. Use it to initialize the variables, pin modes, start using libraries, etc. The setup function will only run once, after each power up or reset of the Arduino board.

INPUT:

OUTPUT:

RETURN:

## 6.3 Software Description

### 6.3.1 Arduino Software:

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

**Download Arduino IDE Software:**

You can get different versions of Arduino IDE from the Download page on the Arduino Official website. You must select your software, which is compatible with your operating system (Windows, IOS, or Linux). After your file download is complete, unzip the file.

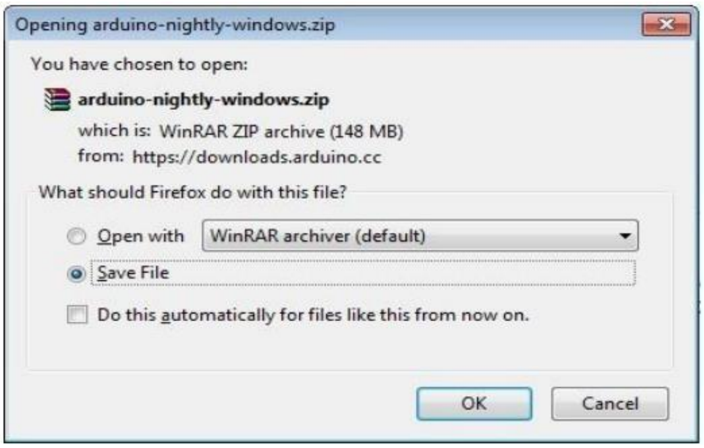



**Fig6.3.1: Opening Arduino nightly windows zip**

**Launch Arduino IDE:** After your Arduino IDE software is downloaded, you need to unzip the folder. Inside the folder, you can find the application icon with an infinity label (application.exe). Double click the icon to start the IDE.

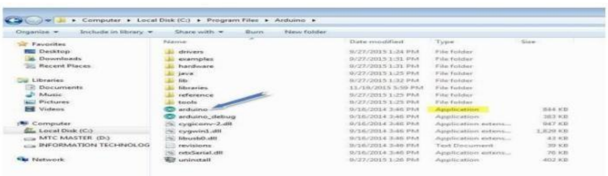


**Fig 6.3.2 Launching of Arduino IDE**

**Open your first project:** Once the software starts, you have two options: • Create a new project.

- Open an existing project example. To create an e w project, select File-

->New. To open

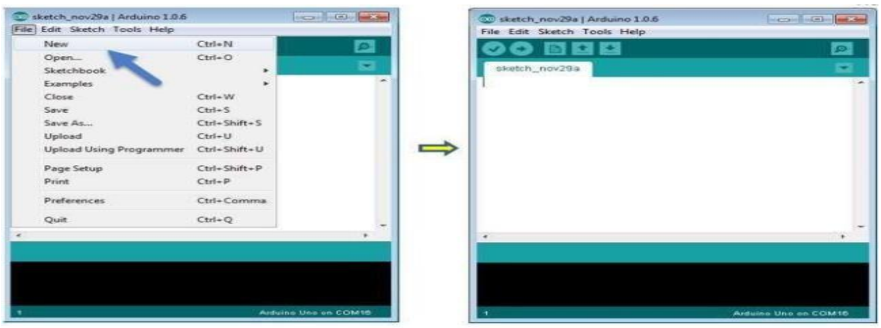



**Fig6.3.3 Opening the project**

**Write the Code:**

According to the project write the necessary code required for the execution of output.

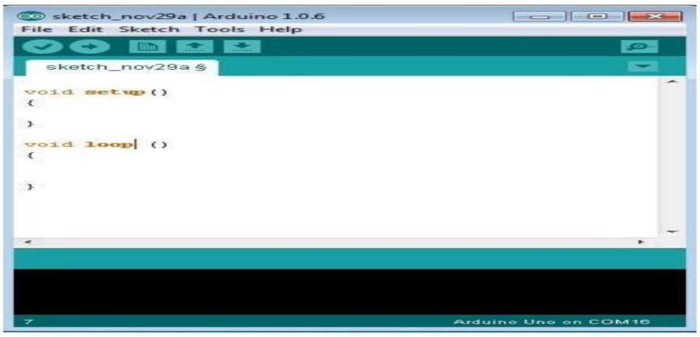



**Fig6.3.4 Representation of writing the code Upload the program to your board:**

Before explaining how we can upload our program to the board, we must demonstrate the function of each symbol appearing in the Arduino IDE toolbar.

**Fig 6.3.5 Uploading of program**

A-Used to check if there is any compilation error.

B-Used to upload a program to the Arduino board.

C-Short cut used to create a new sketch.

D-Used to directly open one of the example sketch.

E-Used to save your sketch.

F-Serial monitor used to receive serial data from the board and send the serial data to the board.

Now, simply click the "Upload" button in the environment. Wait a few seconds; you will see the RX and TX LEDs on the board, flashing. If the upload is successful, the message "Done uploading" will appear in the status bar.

**Fig 6.3.6USB Cable**

## 6.4 Proteus Software:

Proteus is a widely used software tool primarily for electronic design automation (EDA) and simulation of electronic circuits. It's developed by Lab center Electronics Ltd., based in the UK.

**Designing a circuit in Proteus:**

The process of designing a circuit in Proteus typically involves several steps. Here's a general outline of the typical workflow:

**Launch Proteus:** Start by launching the Proteus software on your computer.


**Fig 6.4.1 Launching Proteus**

**Create a New Project:** In Proteus, circuits are organized within projects. Create a new project by selecting "File"> "New Project"fromthe menu bar. Give your project a name and specify a location to save it.


**Fig 6.4.2Creating a New Project**

**Open Schematic Capture:** With in your newly created project, open the schematic capture module by selecting "File" > "New Schematic" from the menu bar.

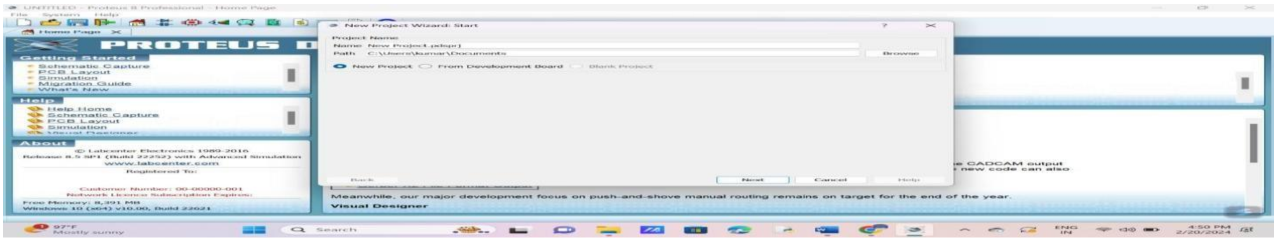
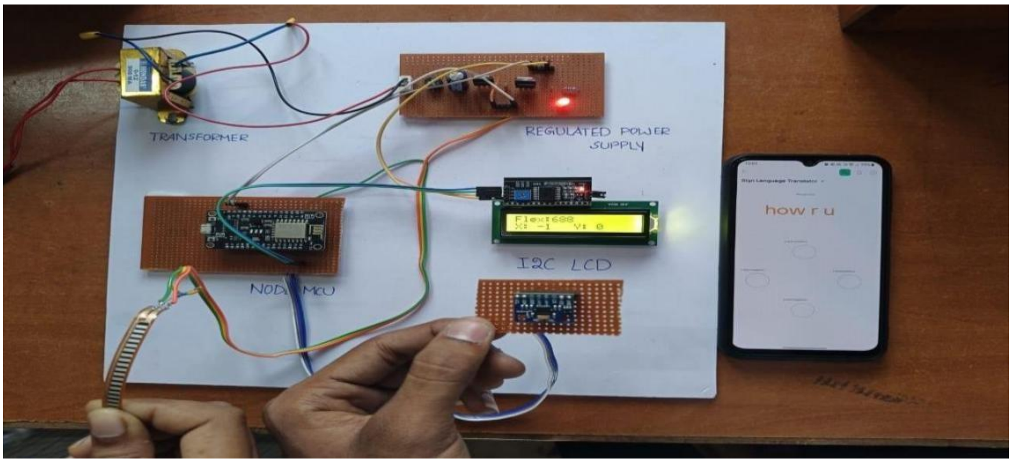


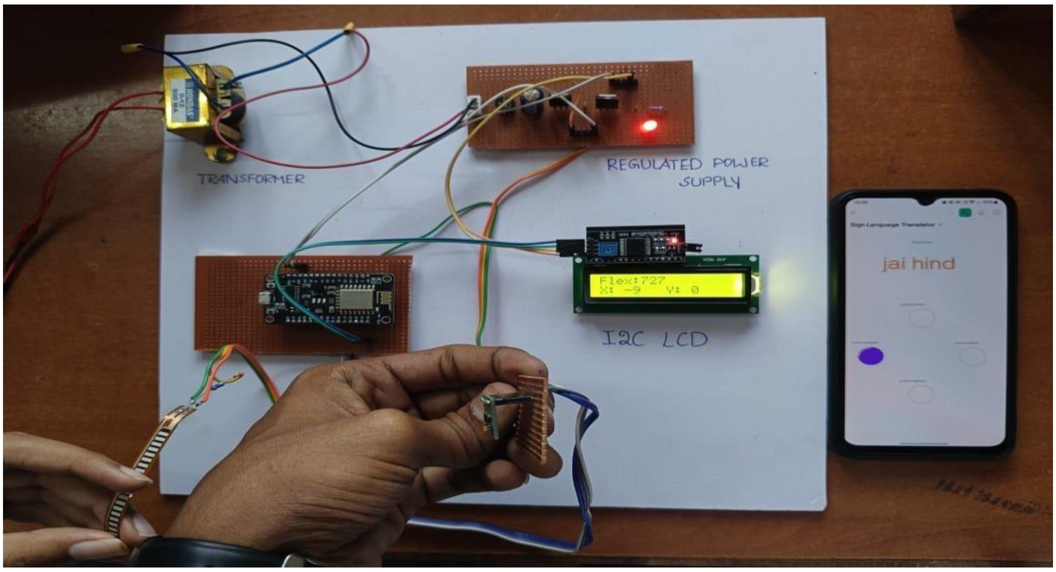

**Fig 6.4.3Schematic Capture**

# CHAPTER-7
# RESULTS AND ANALYSIS

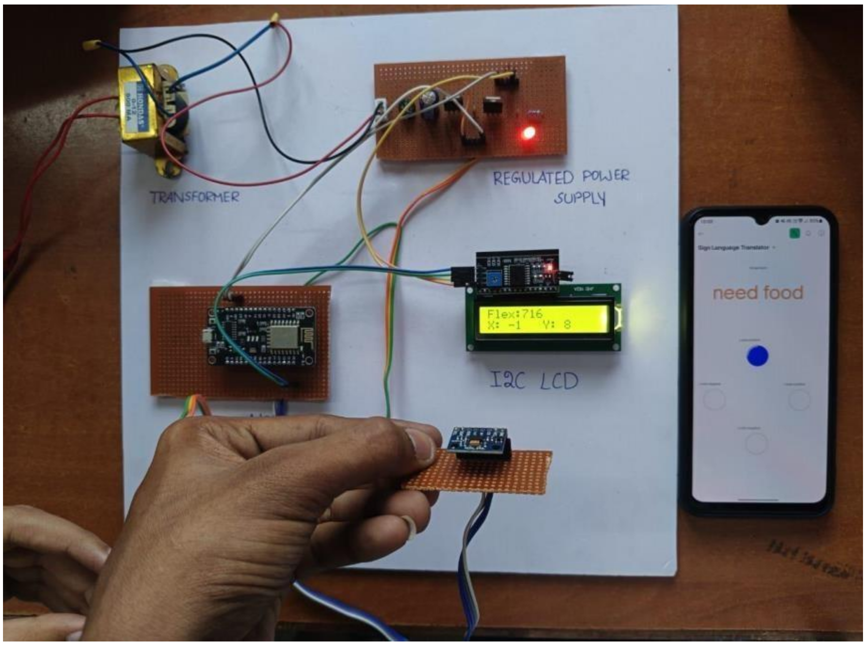
The developed Malaysian Sign Language (MSL) translator system successfully converts hand gestures into corresponding words, facilitating communication between speech and hearing-impaired individuals and those unfamiliar with sign language. The system effectively utilizes flex sensors to measure finger bending and an MPU6050 accelerometer gyroscope sensor to determine hand positioning. This combination enables accurate gesture recognition, ensuring precise interpretation of MSL gestures into readable text.




The integration of the Node MCU ESP8266 with the Blynk application allows for seamless IoT-based real-time display of translated text on a smartphone. The system demonstrates low latency and high accuracy in processing gestures, making it a reliable tool for real-time communication. The wireless data transmission between the data glove and the smartphone ensures user convenience, eliminating the need for complex wired connections




Testing results indicate that the system performs efficiently under different hand movement conditions, maintaining an accuracy rate suitable for effective translation. The flex sensors provide stable and repeatable readings, while the MPU6050 sensor accurately detects hand orientation, ensuring minimal misinterpretation of gestures. The system also allows for scalability, additional words and gestures used .



# CHAPTER-8

# ADVANTAGES AND APPLICATIONS

## 8.1 ADVANTAGES

**Social Benefits**

1. Social Integration: Facilitates social interactions and relationships.
2. Community Engagement: Encourages participation in community events and activities.

**Educational Benefits**

1. Better Learning Outcomes: Enhances understanding and engagement in educational settings.
2. Increased Accessibility: Provides equal access to educational resources and materials.

## 8.2 DISADVANTAGES

1. Cost: Sign language translators can be expensive, limiting accessibility for some users.
2. Limited language support : many systems are designed for specific sign languages (example.,ASL) limiting global accessibility and inclusivity .

## 8.3 APPLICATIONS

**Education**

1. Inclusive Classrooms: Enhances learning experiences for students with hearing impairments.
2. Accessibility in Online Courses: Provides equal access to educational resources and materials.

**Healthcare**

1. Effective Communication: Facilitates accurate communication between healthcare providers and patients with hearing impairments.
2. Improved Patient Care: Enhances patient outcomes and satisfaction.

# CHAPTER – 9
# CONCLUSION AND FUTURESCOPE

## 9.1 CONCLUTION

In conclusion, this project successfully developed a Malaysian Sign Language (MSL) translator system using Node MCU ESP8266, flex sensors, and the MPU6050 accelerometer gyroscope sensor module. By utilizing a data glove approach, the system effectively converts hand gestures into words, allowing speech and hearing- impaired individuals to communicate with those unfamiliar with sign language. The combination of flex sensors for finger bending detection and an accelerometer- gyroscope module for hand position tracking ensures accurate gesture recognition, making the system a reliable tool for real-time translation. Furthermore, integrating IoT technology into the project enhances its accessibility and usability. By using the Node MCUESP8266 as an IoT plat form and connecting it with the Blynk application, the translated gestures are displayed on a smartphone, providing a seamless and user-friendly experience. This innovation bridges the communication gap between the speech and hearing-impaired community and the general public, offering a practical solution for daily interactions. The system's wireless connectivity and real-time processing further improve its efficiency and convenience.

## 9.2 FURURE SCOPE

**Multi-Language Support** –The system can be expanded to support multiple sign languages, including ASL and British Sign Language (BSL), making it accessible to a broader audience.

**AI-Powered Real-Time Translation** – Incorporating AI-powered real-time translation can help convert sign language gestures directly into text or speech on digital platforms like video calls or smart assistants.

# REFERENCE

## 1. Research Papers and Articles on Sign Language Translation:

This paper reviews existing sign language recognition systems, providing a solid background for the challenges and technologies involved in developing a sign language translation system.

## 2. Development of Gesture Recognition Systems:

This article reviews various gesture recognition systems, focusing on flex sensors and accelerometer-based technologies, which are key components in your project.

## 3. Internet of Things in Sign Language Translation:

Sharma, R., &Singh, This paper discusses how IoT technologies, including Node MCU and cloud integration, are being used in sign language translation systems to bridge communication gaps.

## 4. Flex Sensors and Gesture Recognition:

International Journal of Engineering and Technology. This article provides insight into how flex sensors are employed in gesture-based systems and their advantages in detecting finger and hand movements A relevant article that highlights the use of accelerometer and gyroscope modules in wearable gesture recognition systems for various applications, including sign language translation.