SENSOR BASED SIGN LANGUAGE TO SPEECH CONVERTER

A project report submitted in partial fulfillment of the requirements for the award of the degree of

Bachelor of Engineering

In

Electronics and Communication Engineering

Submitted by

H. Vishal 160119735058

A. Vishnu 160119735059

Ch. Supriya 160119735301

Under the esteemed guidance of

Dr M. RAMANA REDDY

Assistant Professor, Dept. of ECE

Department of Electronics and Communication Engineering

Chaitanya Bharathi Institute of Technology(A)

Gandipet, Hyderabad-500075

DECLARATION

This is to certify that the work reported on the present project entitled **SENSOR BASED SIGN LANGUAGE TO SPEECH CONVERTER** is a record of work done by us in the Department of Electronics and Communication Engineering, Chaitanya Bharathi Institute of Technology, Osmania University. This report is based on the project work done entirely by us, it doesn't involve plagiarism and any information procured from resources has been mentioned in references.

Hridwin Vishal (1601-19-735-058)

Alakuntla Vishnu (1601-19-735-059)

Chakilam Supriya (1601-19-735-301)

Department of

Electronics and Communication Engineering

Chaitanya Bharathi Institute of Technology(A)

Gandipet, Hyderabad-500075

CERTIFICATE

This is to certify that the Project work entitled "**SENSOR BASED SIGN LANGUAGE TO SPEECH CONVERTER**" is a bonafide work carried out by H.Vishal (160119735058), A. Vishnu (160119735059), Ch. Supriya (160119735301) in partial fulfillment of the requirements for the degree of Bachelor of Engineering in Electronics and Communication Engineering, Osmania University, Hyderabad during the academic year 2022-23. The results embodied in this report have not been submitted to any other University or Institution for the award of any diploma or degree.

Supervisor
Professor Dr M. Ramana Reddy
Department of ECE

Professor D. Krishna Reddy
Head of Department, ECE

ACKNOWLEDGEMENT

At the very outset of this report, we would like to thank with utmost gratitude and sincerity to all the people who have helped us in this effort. Without their guidance, support, and encouragement we wouldn't have made it through this project.We are gratified and beholden to our guide **Dr. M. Ramana Reddy Sir** , Professor, Department of ECE - CBIT, for his constant guidance and support throughout our work, attending to our needs, and helping us in finding solutions to various problems encountered. We sincerely thank our project coordinator **Dr. K. Suman**, Professor, **Dr. M.L.N. Charyulu** , Associate Professor, and **Dr. M. Vinodh Kumar** , Assistant Professor Department of ECE - CBIT, for various suggestions which have benefitted our work in various ways. We express our profound gratitude to **Dr. D. KRISHNA REDDY**, Head of the Department ECE - CBIT, for this valuable guidance and enormous support. We express our profound gratitude to **Dr. P. RAVINDER REDDY**, Principal of CBIT, for his valuable guidance and facilities provided.

ABSTRACT

Communication is the only medium by which we can share our thoughts or convey a message but deal and dumb people are facing difficulty in communication with normal people. Generally , dead and dumb people use sign language for communication , but they find difficulty in communication with normal people, who did not understand sign language. Hence, there is a barrier to communication between these two communities and with the outside world. To overcome the above problem, a smart glove is proposed for deaf and dumb ed or communicating with normal people. It converts sign language (hand gestures) into a text message and auditory voices. This project proposes the Sensor-based technology approach. Flex sensors and accelerometer are used along with Arduino Uno, which can program by using Embedded C language. Whenever a hand gesture is given, the sensor senses the angle of tilting and bending, then it produces output voltage according to the input gesture. This voltage converts into text messages as well as audio output.

Table of Contents

List of figures

List of tables

CHAPTER 1

1. INTRODUCTION

1.1 Introduction

In the recent years there has been a rapid increase in the number of hearing impaired and speech disabled victims due to birth defects, oral diseases, and accidents. If a speech impaired person wants to speak to a normal person, the normal person finds it difficult to understand and asks the deaf-dumb person to show some gestures for understanding. Dumb persons have their gesture language to communicate with us, but barrier is we need to understand their language. The language used by deaf and dumb people for communication is known as sign language. The sign language used by deaf and dumb people do not have a common origin. As regular people aren't trained on hand gestures i.e., sign language, so the communication becomes very difficult for them. In emergency situations or sometimes communicating with new people, it becomes difficult in conveying their message to normal people.

To overcome the barrier, here we ar proposing a smart glove that helps deaf and dumb people in conveying the message to nova: ~eople by using hand signs or gestures. The main aim of this project is to facilitate protic by means of a glove-based communication interpreter system. The smart hand glove i= mormal cloth glove which is fitted with flex sensors along with the length of each finger to the hand. The sensors give an output that is a stream of data that varies with the degree of bend. In this project flex sensors play a major role, flex sensors can change the resistance depending on the amount of bend on the sensor. They convert the change in the bend of the hand to electrical resistance. The Accelerometer sensor is also used and placed on the top of the glove to measure the acceleration or change velocity of the hand.

1.2 Aim

The aim of our project is to implement a sign language to voice converter in real time using a wearable hand glove.

1.3 Objectives

- To develop a sign language to speech converter that accurately translates American Sign Language (ASL) letters into English letters, with a focus on clarity and audibility.
- To develop a sign language to speech converter that accurately translates some common ASL signs into English words, with a focus on usability and practicality.
- To optimize the device's performance in terms of minimizing latency and maximizing accuracy, through the selection of appropriate hardware components and software algorithms.
- To collect and analyze sensor data to create a comprehensive dataset for the letters and words translated by the device, with a focus on improving the accuracy and precision of the translation process.
- To design and implement a power management system that enables the device to function continuously for several hours without requiring battery replacement or charging

1.4 Motivation

Communication between deaf an: dunt people can be problematic and inconvenient. This project attempts to bridge the communication gap by designing a portable glove that captures the user's gestures and outputs the translated text message. The existing projects designed by using IOT technology. In which it uses wireless communication technologies which may cause an obstacle in signal prone areas. To overcome this problem wired Arduino Uno based technology is used and it also provides the cost effectiveness. In the existing system only alphabets are displayed, but in the proposed system sentences are also displayed.

1.5 Literature survey

Sign Language Recognition

Anup Kumar, Karun Thankachan and Mevin M. Dominic National Institute of Technology, Calicut, India

- The objective of this paper is sign language recognition and conversion of speech to signs. which is able to recognize static and dynamic gestures of American Sign Language using computer vision and image processing.
- Static and dynamic gestures are recognized and analyzed in the paper
- The paper uses HSV thresholding for isolating the hand and calculate its center of gravity. Further finger tips are detected and the angular data is analzed and signs are dtermined.

Patience assistance using flex sensor

Kollu Jaya Lakshmi, Akshada Muneshwar, A Venkata Ratnam and Prakash Kodali

- The objective of this paper is to help people with facial and limb paralysis to control their surroundings like control air conditioning, TV or any other device using NodeMCU module and ask for assistance through simple operations . It picks up bending in finger and translates to a selective control.
- The paper uses voltage divider circuitary to read and analyze flex sensors.

Development of Sign Language using Flex Sensors

Ajay Suri,Dr. Sanjay Kumar Singh,Rashi Sharma ,ABES Engineering College

- The objective of this paper is developing a glove-based Sign Language translator It uses adaptable glove senses the client's American Sign Language developments and gives the output in a text or speech form at a screen.
- Flex sensors for each finger, accelerometer, Arduino as a primary controller, Bluetooth module and glove as hardware .

- An average accuracy of 92% is obtained with an response time of 1.1s-1.5s for sentences.

1.6 Organization of Thesis :

The thesis is organized into five chapters, they are listed below:

Chapter 1 Describes introduction to smart glove, literature Survey, motivation, aim of the

thesis, objectives, and technical approach.

Chapter 2 Deals with various Hardware components used in the Smart Glove.

Chapter 3 Describes the Methodology, hardware interfacing and flow chart of Smart Glove.

Chapter 4 Details about the Results, and Discussions obtained.

Chapter 5 Describes conclusion and future scope

CHAPTER 2

DESCRIPTION OF HARDWARE COMPONENTS

2.1 Arduino Uno

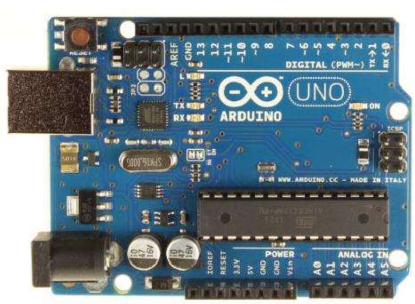
Arduino is a prototype and open-source stage dependent on a simple equipment and programming. It comprises of a circuit board, which can be customized (referred to as a microcontroller) and an instant programming called Arduino IDE (Integrated Development Environment), which is utilized to compose and transfer the PC code to the actual board. Arduino gives a standard structure factor that breaks the elements of the miniature regulator into a more open bundle. There are some prerequisites for Arduino to get started. Arduino needs basics of C language and C++, understanding of Micro-controllers and electronics.

2.1.1 Introduction to Arduino Uno

Arduino Uno is a microcontroller board dependent on the ATmega328P. It has 14 progressed input pins (of which 6 can be used as PWM), 6 straight forward datasources, a 16 MHz artistic resonator (CSTCE16M0V53-R0), a USB affiliation, a force jack,an ICSP header and a reset button. The Arduino Uno contains all the contents that help the microcontroller. Fundamentally, user need to interface Arduino board to a PC with a USB connection or force it with an AC-to-Dc conector. An AC-to-DC connector or battery 1s 2! to start the Arduino. User need not to stress about the results of the Arduino board, int worst-case scenario user can replace the chip with a new one and start over again.

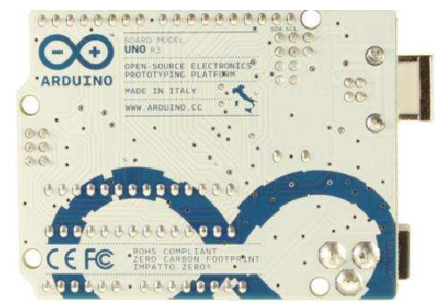
2.1.2 Board Description

The figure 2.1 is the Arduino uno board, The length of the board is 68.6 mm, and the width is 53.4 mm, weight 25g. The board contains the power USB, power barrel back, voltage regulator, crystal oscillator along with Arduino Reset button, main Microcontroller. The below is the pin description of the Arduino uno board.



Arduino Uno R3 Front

Fig 2.1.1: Arduino Uno Front side



Arduino Uno R3 Back

Fig 2.1.2 Arduino Uno Back side

2.1.3 Pin Description of Arduino Uno

The figure 2.1 shows pin description of Arduino Uno.

PIN 1 (USB Type B): This pin is used to upload the program from the Arduino IDE Software to the Arduino Board. Also used as Power supply to the Board.

PIN 2 (Power socket): This pin is used to power on the Board.

PIN 3 (Voltage Regulator): The voltage regulator is used to regulate the power supplied to the board. It provides continuous power supply within the power limit range of the controller.

PIN 4 (Crystal Oscillator): The crystal oscillator provides the oscillation pulses to the controller for working purposes. The number on top of the Arduino crystal is 16.000H9H. 6,000,000 Hertz or 16 MHz

PIN 5&17 (Reset): To reset the Arduino, i.c., start execution of the program from the beginning. Either user can use the reset button (17) or user can externally connect the button to the pin labeled reset (5).

PIN 6,7.8.9 : 3.3V (6) -Supply 3.3V output volt. SV (7) — Supply 5V output volt

GND (8) (Ground) : There are many ground pins provided with Arduino to connect with ground .

Inputs & Output s: Vin (9) — This pin can be also used to power the Arduino.

PIN 10 (Analog pins): There are Six analog input pins (AO to AS). These Pins are used to read the analog input values. These Pins can also be used as digital output pins.

PIN 11(Microcontroller): Each Arduino has a microcontroller that controls all inputs and outputs and processes the entire information. These controllers are usually of ATMEL company. the information of the controller is written on the IC.

PIN 12(ICSP): This pin contains MOS], MISO, SCK, RESET, VCC, and GND Pins. It also referred as an SPI (Serial Peripheral Interface).

PIN 13: Power LED indicator. This LED indicates whether the Arduino is connected to the power supply or not.

PIN 14: TX and RX LEDs These LEDs blink when there is serial communication. It is mostly seen when user upload the program to the Arduino board.

PIN 15: Digital I/O. The Arduino uno is as 14 Digital I/O pins (15). In that 6 of them provides PWM (Pulse Width Modulation} output. Pins with the "~" symbol indicate PWM.

PIN 16: AREF means Analog Reference. sometimes it is used to set external reference

between 0 to 5V and upper limit to analog input pins.

2.1.4 Features of Arduino Uno

The below are the features of Arduino Uno board

- Microcontroller Atmega328P
- The operating voltage is 5v
- Input Voltage recommended ranges from (7V-12V)
- Input Voltage ranges (6V-20V)
- I/O digital PWM pins are 6
- Analog input pins are 6
- DC current for I/O Pin is 20mA
- DC current per 3.3 V Pin is 50mA
- The Flash Memory of 32KB(ATmega328P) of which 0.5 KB used by bootloader
- The SRAM of 2KB (ATmega328P)
- The EEPROM of 1KB{ATmega328P)
- Clock Speed of Arduino Uno is 16MHZ
- LED BUILTIN 13
- Weight 25g

2.2 Arduino Uno Architecture

Architecture describes the complex or carefully designed structure of Board. The ATmega328/P is a low-power CMOS 8-bit micro-controller based on the AVR enhanced RISC (reduced instruction set computer) architecture. To increase performance of Arduino Uno, the Arduino microcontroller uses Harvard architecture with separate memories and buses for program and data. The instruction in the program memory is executed with a single level of pipe-lining, and clock is controlled by an external 16MHz Crystal Oscillator. The basic working of CPU of ATmega328 is, to upload the data the data in serial port (uploaded from the computer's Arduino IDE). The information is decoded, afterward the instructions are shipped off instruction register, and it interprets the instructions on a similar clock beat.

On the following clock beat the following arrangement of instructions is stacked in the instruction register. There are 3, 16-bit registers present in the Arduino Uno.

A) 8-bit registers are used for storing the data, for normal calculations and results.

B) 16-bit registers are used to store data of timer counters in 2 different registers.

Example: X-low & X-high. They are fast and are used to store specific hardware functions. EEPROM stores data permanently even if the power is cut out. Programming inside an EEPROM is slow. Interrupt uit checks whether there is an interrupt for the execution of the instruction to be executed ISR (Interrupt Service Routine). Serial Peripheral Interface (SPI) is an interface transport generally used to send information among microcontrollers and little peripherals, for example, camera, Display, SD cards, and so forth It utilizes separate clock and information lines, alongside a select line to pick the gadget you wish to converse with. Guard dog clock is utilized to recognize and recuperate from MCU breaking down. A simple comparator looks at the info esteems on the positive and negative pin, when the worth of the positive pin is higher the yield is set. Status and control are utilized to control, the progression of execution of orders by checking different squares inside the CPU at ordinary spans. ALU(Arithmetic and Logical unit) the elite AVR ALU works, indirect association with all the 32 broadly useful working registers. Inside a solitary clock cycle, math tasks b/w universally useful registers are executed. The ALU operations are isolated into 3 primary classes arithmetic, logical and bit-function. The digital inputs and outputs (digital I/O) on the Arduino are, what permit user to associate the Arduino sensors, actuators, and different IC's. Figuring out how to utilize them will permit user to utilize the Arduino to do some truly helpful things, for example, perusing switch inputs, lighting markers, and controlling relay outputs.

2.2.1 Communication

The Arduino Uno has a number of facilities for communicating with a computer, another Arduino or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega16U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The '16U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, on Windows, a file is required.

The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (but not for serial communication on pins 0 and A Software Serial library allows for serial communication on any of the Uno's digital pins. The ATmega328 also supports I2C (TWI) and SPI communication.

The Arduino software includes a Wire library to simplify use of the I2C bus; see the documentation for details. For SPI communication, use the SPI library.

2.2.2 Programming

The Arduino Uno can be programmed with the Arduino software. The ATmega328 on the Arduino Uno comes pre burned with a boot loader that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol (reference, C header files).You can also bypass the boot loader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see these instructions for details.

The ATmega16U2 (or 8U2 in the rev1 and rev2 boards) firmware source code is available. The ATmega16U2/8U2 is loaded with a DFU boot loader, which can be activated by: On Rev1 boards: connecting the solder jumper on the back of the board (near the map of Italy) and then resetting the 8U2.

On Rev2 or later boards: there is a resistor that pulling the 8U2/16U2 HWB line to ground, making it easier to put into DFU mode. You can then use Atmel's FLIP software (Windows) or the DFU programmer (Mac OS X and Linux) to load a new firmware. Or you can use the ISP header with an external programmer (overwriting the DFU boot loader). See this user-contributed tutorial for more information.

2.2.3 Automatic (Software) Reset

Rather than requiring a physical press of the reset button before an upload, the Arduino Uno is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of theATmega8U2/16U2 is connected to the reset line of the ATmega328 via a 100Nano farad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment.

This means that the boot loader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload. This setup has other implications. When the Uno is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB).

For the following half-second or so, the boot loader is running on the Uno. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened.

If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data. The Uno contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line.

2.2.4 USB Over current Protection

The Arduino Uno has a resettable poly fuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

2.2.5 Physical Characteristics

The maximum length and width of the Uno PCB are 2.7 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Four screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16") not an even multiple of the 100 mil spacing of the other pins.

2.3 AT MEGA 328

The **ATmega328** is single-chip microcontroller created by Atmel in the megaAVR family (later MicrochipTechnology acquired Atmel in 2016). It has a modifiedHarvardarchitecture 8-bit RISC processor core.

Fig 2.3.1 ATmega328P in 28-pin narrow dual in-line package (DIP-28N)

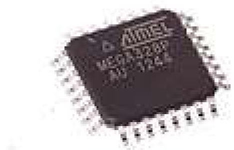


Fig 2.3.2 ATmega328P in 32-pin thin quad flat pack (TQFP-32)

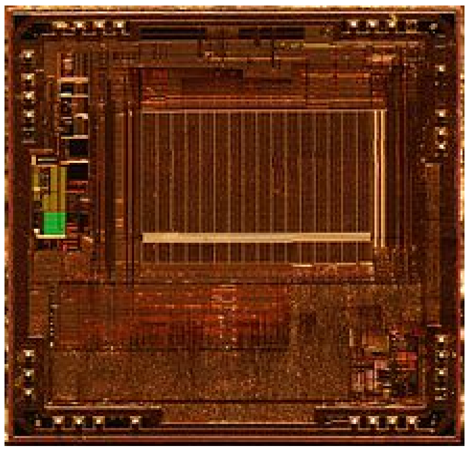


Fig3.6 Die of ATmega328P

You can use microcontroller “328P” separately, there is no need to use it on Arduino board with its own IDE.

2.3.1What's the meaning of ATmega328p in Arduino UNO?

Arduino UNO is the name of development board which has "microcontroller ATmega 328pu" mounted on it. A microcontroller is an IC which performs all the tasks but to get hold of the pins for easy use and uploading code, Arduino UNO board is made. Apart from just making pins available it provides extra 5v and 3.3v pins for use, LED on pin 13 of board for debugging etc. Unlike other series of microcontrollers you don't need extra connections from microcontroller to upload your code. Programmer is already built in Arduino board which can be connected to USB port via cable. Simply other software's apart from Arduino like Atmel studio, caver can be used to upload the code like we do with other microcontrollers but it will just make the process more hectic and time consuming. Arduino UNO board supports Arduino IDE in which the code is written and tests are performed. Arduino IDE provides multiple functionalities and huge community support. Serial monitor and plotter can be used for easy debugging and data plotting. Almost for every sensor/module to be used by hobbyists there is a library already built in or can be added separately. It is great for those who don't want to do direct register coding like in other microcontroller

2.3.2 Specifications

The Atmel 8-bit AVR RISC-based microcontroller combines 2kb ISP flash memory with read-while-write capabilities, 1kb EEPROM, 2kb SRAM, 23 general purpose I/O lines, 32 general purpose working registers, three flexible timer/counters with compare modes, internal and external interrupts, serial programmable USART,a byte oriented 2-wire serial interface, SPI serial TQFP and QFN/MLF packages), programmable watchdogtimer with internal oscillator, and five software selectable power saving modes. The device operates between 1.8-5.5 volts. The device achieves throughput approaching 1 MIPS per MHz

2.3.3 Key parameters

Table

Parameter	**Value**
CPU type	8-bit AVR
Performance	20 MIPS at 20 MHz[2]
Flashmemory	32 kb
SRAM	2 kb
EEPROM	1 kb
Pin count	28-pin PDIP, MLF, 32-pin TQFP, MLF[2]
Maximum operating frequency	20 MHz
Number of touch channels	16
Hardware QTouch Acquisition	No
Maximum I/O pins	23
External interrupts	2
USB Interface	No
USB Speed	–

Table 2.3.1: Key parameters

A common alternative to the ATmega328 is the "pic power" ATmega328P. A comprehensive list of all other members of the mega AVR series can be found on the Atmel website.

2.4 Arduino Uno IDE

The Arduino Integrated Development Environment or Arduino IDE incorporates a text editor for writing code, a message display arca, a text console, a toolbar with buttons for common functions and a series of menus. It connects to the Arduino and hardware to upload programs and communicate with them.

The code written in Arduino IDE is called as “Sketch” or “Sketches”. Software can be downloaded from https://www.arduino.cc/en/software and is compatible with the macOS, Windows and Linux. Arduino IDE consists of several libraries: serial monitor and various tools.

2.4.1 Libraries

Libraries provide extra functionality for use in sketches, for instance working with hardware or manipulating data. To use a library in a sketch, select it from the Sketch by then Import Library menu. This will call at any rate one #include articulations at the most noteworthy place of the sketch and assemble the library with your sketch. Since libraries are transferred to the board with your sketch, they increment the measure of room it takes up. On the off chance that a sketch not, at this point needs a library, basically erase its #include statements from the highest point of code.

2.4.2 Serial Monitor

Serial screen shows sequential sent from the Arduino over USB or chronic connector. To send information to the board, enter text and snap on the "send" catch or press enter. Select the baud rate from the spring up menu that matches the rate passed to Serial (by and large 9600).

To start the sketch, note that on Windows, Mac, or Linux Arduino board need to reset (it will rerun your sketch) when you interface with the monitor screen. Please note that the Serial Monitor does not process control characters, if the user sketch needs complete management of the serial communication with control characters, can use an external terminal program and connect it to the COM port assigned to your Arduino board.

2.4.3 Arduino Uno Installation:

In this we will get know of the process of installation of Arduino IDE and connecting Arduino Uno to Arduino IDE.

Step 1

First we must have our Arduino board (we can choose our favorite board) and a USB cable. In case we use Adriana UNO, Arduino Duemilanove, Nano, Arduino Mega 2560, or Diecimila, we will need a standard USB cable (A plug to B plug), t In case we use Arduino Nano, we will need an A to Mini-B cable.

Step 2 − Download Arduino IDE Software:

We can get different versions of Arduino IDE from the Download page on the Arduino Official website. We must select wer software, which is compatible with wer operating system (Windows, IOS, or Linux).

After wear file download is complete, unzip the file.

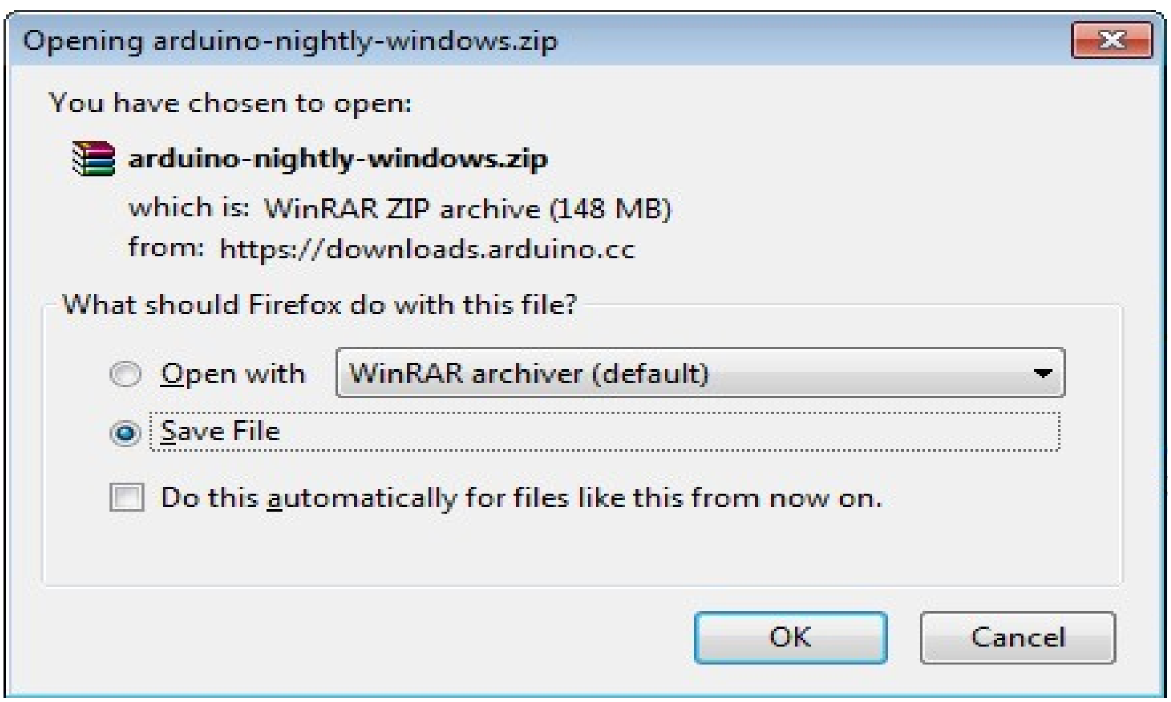


Fig. 2.4.3.1: Arduino Installation

Step 3 Power up our board:

The Arduino Uno, Mega, Duemilanove and Arduino Nano automatically draw power from either, the USB connection to the computer or an external power supply.

If we are using an Arduino Diecimila, we have to make sure that the board is configured to draw power from the USB connection. The power source is selected with a jumper, a small piece of plastic that fits onto two of the three pins between the USB and power jacks.

Check that it is on the two pins closest to the USB port.Connect the Arduino board to wer computer using the USB cable. The green power LED (Labeled PWR) should glow.

Step 4 − Launch Arduino IDE:

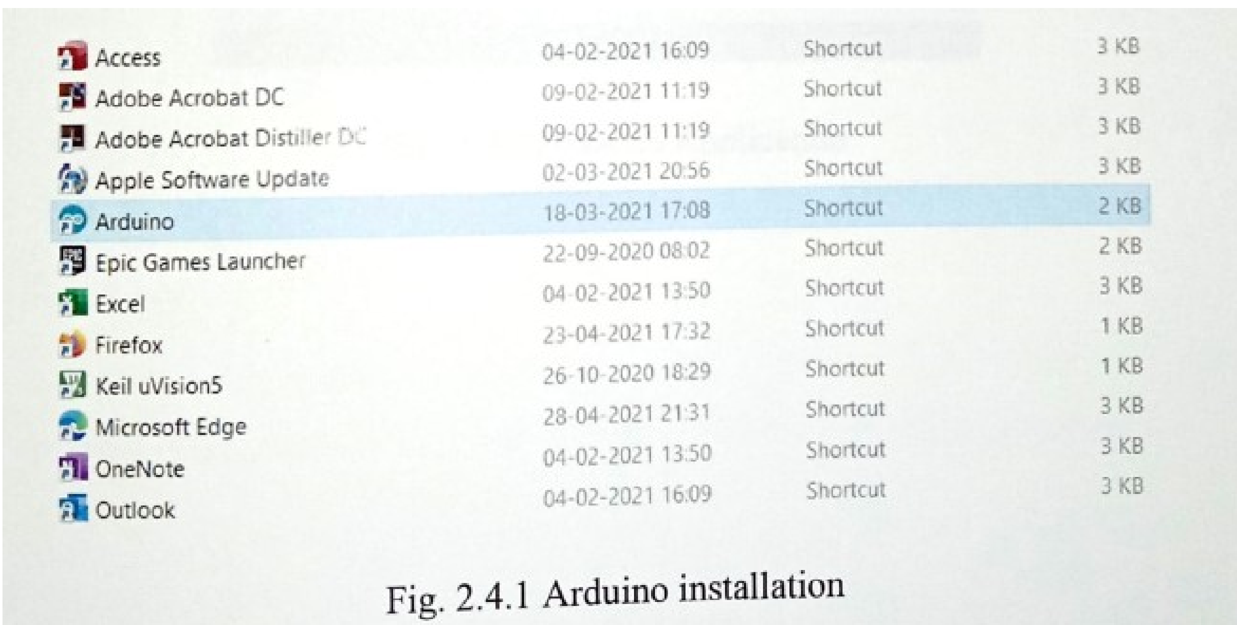


Fig. 2.4.2 : Arduino Installation-2

Step 5: Open the first project.

Once the software starts, there are two options available to Create a new project. First open an existing project example. If user need to create a new project, select File → New.

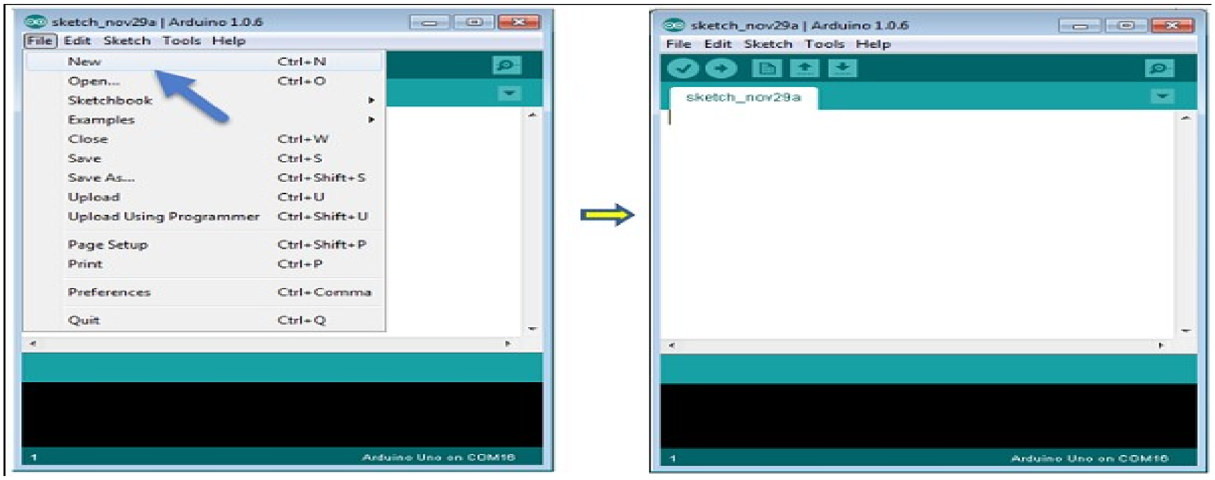
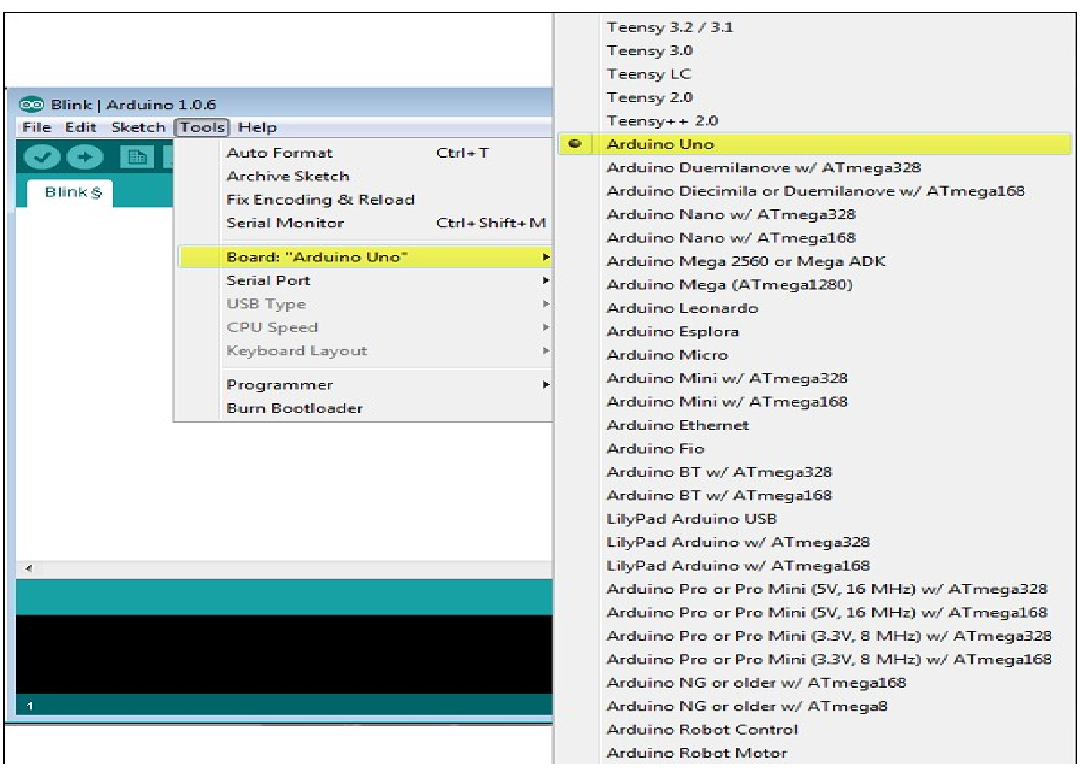


Fig. 2.4,3 Arduino setup

If user need to open an existing project, select File → Example → Basics → Blink

Here, by choosing just one of examples from the name Blink. It turns LED ON/OFF, it has time delay and user can select any other example from the list.

Step 6: Select our Arduino board:



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

Go to Tools → Board and select wear board.

Here, we have selected Arduino Uno board according to our tutorial, but we must select the name matching the board that we are using.

Step 7 Select wer 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).

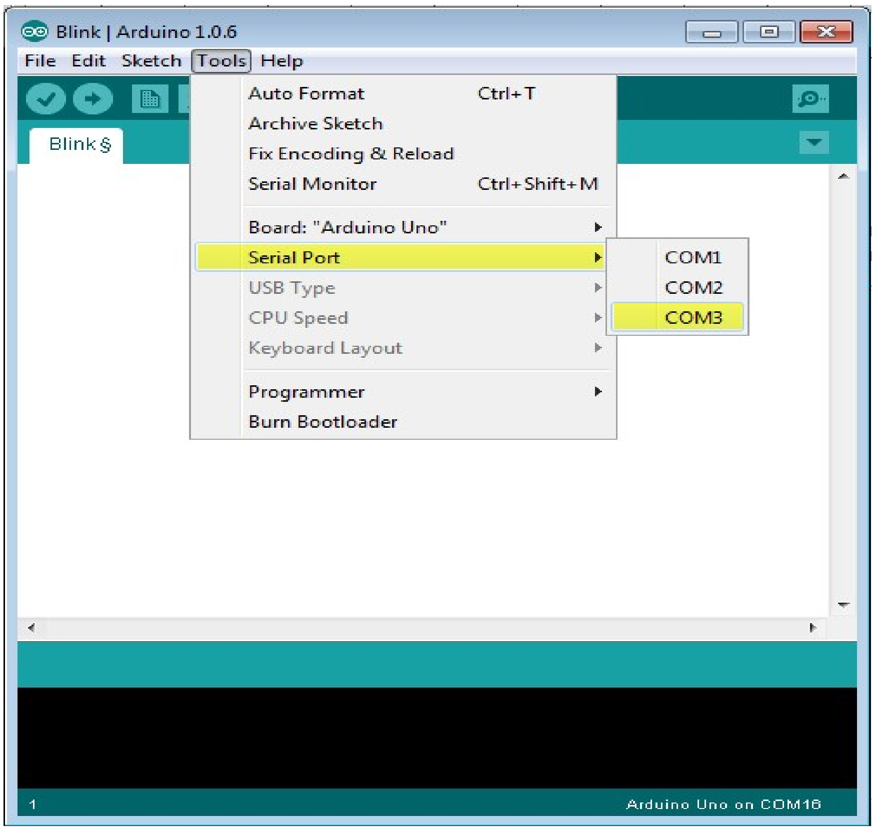


Fig. 2.4.4 Port selection

To find out, we can disconnect wer Arduino board and re-open the menu, the entry that disappears should be of the Arduino board. Reconnect the board and select that serial port.

Step 8 − Upload the program to the 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.

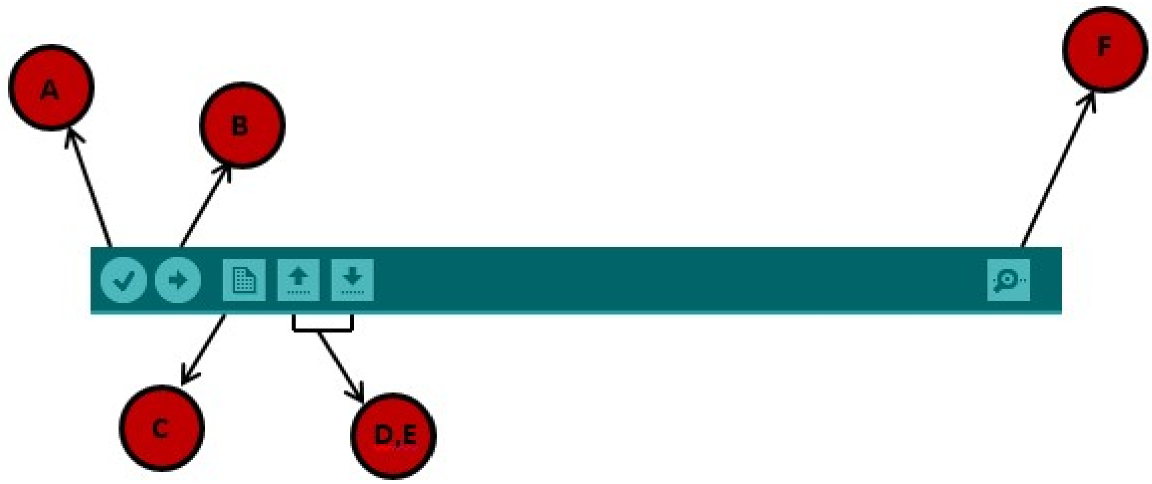


Fig. 2.4.5 : Uploading Sketch

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 wer 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; we 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.

Note − If we have an Arduino Mini, NG, or other board, we need to press the reset button physically on the board, immediately before clicking the upload button on the Arduino Software18

2.5 Accelerometer (ADXL345)

The below fig 2.5 is the Accelerometer sensor. accelerometer is a small, low power, complete 3- axis module. The accelerometer supports both 12C and SPI interfaces, The accelerometer board paving the features, like on-board 3.3V voltage regulator and level shifter. This makes it easy to interface with 5V microcontrollers or with Arduino Uno.

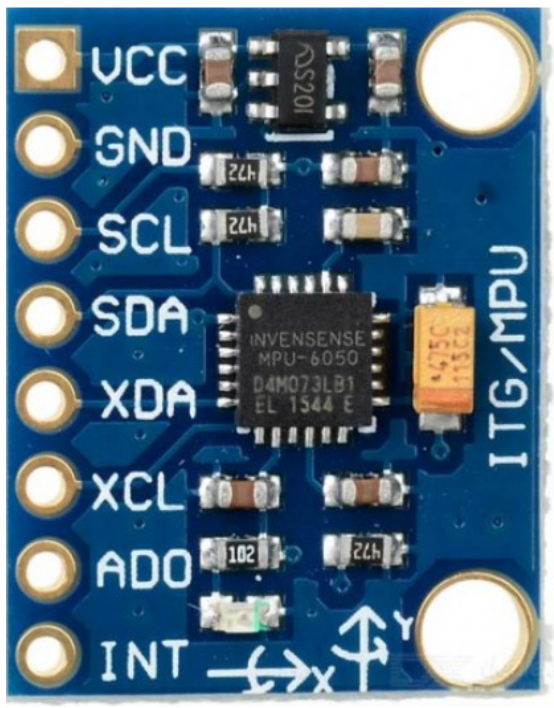


Fig. 2.5.1 ADXL345 accelerometer sensor

Formula for acceleration and Roll calculations:

The below is the formula for measuring the Roll from the accelerations variables

accelerometerX =(signed int)(((signed int) rawData_X D319)

accelerometer Y=(signed int)(((signed int) rawData_Y) *3.9);

accelerometerZ =(signed int)(((signed int) rawData_Z) *3.9),

Roll = (((atan2(Y,Z) * 180) / 3.14) + 180);

2.5.1 Pin Configuration od ADXL345

The figure 2.5 show the accelerometer sensor module, below is the pin description of Accelerometer Sensor (ADXL345)

PIN 1: GND: - Ground pin connected to the ground

PIN 1: VCC: - Power supply pin ranges (3V to 6V)

PIN 1: CS: - Chip select pin to enable the chip

PIN 1: INT1: - output pin of interrupt 1

PIN 1: INT2: - output pin of interrupt 2

PIN 1: SDO: - Pin for serial data output

PIN 1: SDA: - Pin for serial data output and input

PIN 1: SDL: - Pin for serial communication clock

2.5.2 Specifications and Features of ADXL.345

The below are the specifications and features of the ADXI.345.

1. Supply Voltage 3V-6V DC.
2. On-board LDO voltage regulator
3. In built Voltage level converior (MOSFET based)
4. Can be interface with Mic roller of 3.3V or 5V.
5. Ultra-Low Power of 40u/ i sezasurement mode, 0.1uA in standby at 2.5V
6. Free-Fall Detection
7. Can interface for SPI and 12C interfaces
8. Measuring Range up to +16g
9. Measuring Values range (-16g to +16g)
10. X axis value ranges: (-235 to +270)
11. Y axis value ranges: (-240 to +260)
12. Z axis value ranges: (-240 to +270)

2.6 FLEX SENSOR:

This sensor can detect flexing or bending in one direction. They were popularized by being used the Nintendo Power Glove as a gaming interface. These sensors are easy to use they are basically resistors that change value based on how much they're flexed. If they're inflexed, the resistance is about ~10KΩ. When flexed all the way the resistance rises to ~20KΩ. They're pretty similar to FSRs so following this tutorial will get you started. You can use an analog input on a microcontroller (with a pull up resistor) or a digital input with the use of a 0.1uF capacitor for RC timing.The bottom part of the sensor (where the pins are crimped on) is very delicate so make sure to have strain relief such as clamping or gluing that part so as not to rip out the contacts. A flex sensor or bend sensor is a sensor that measures the amount of deflection or bending. Usually, the sensor is stuck to the surface, and resistance of sensor element is varied by bending the surface. Since the resistance is directly proportional to the amount of bend it is used as goniometry, and often called flexible potentiometer.

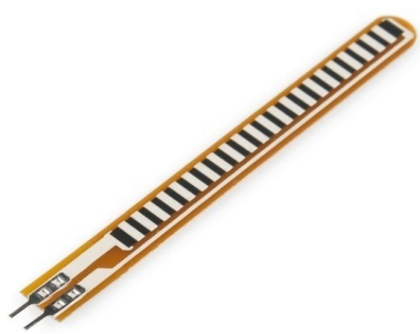


Fig 2.6.1 flex sensor

2.6.1 Flexion sensors, (from Latin *flexure*, 'to bend') also called **bend sensors**, measure the amount of deflection caused by bending the sensor. There are various ways of sensing deflection, from strain-gauges to hall-effect sensors.

The three most common types of flexion sensors are:

- conductive ink-based
- fiber-optic
- conductive fabric/thread/polymer-based

A property of bend sensors worth noting is that bending the sensor at one point to a prescribed angle is not the most effective use of the sensor. As well, bending the sensor at one point to more than 90˚ may permanently damage the sensor. Instead, bend the sensor around a radius of curvature. The smaller the radius of curvature and the more the whole length of the sensor is involved in the deflection, the greater the resistance will be (which will be much greater than the resistance achieved if the sensor is fixed at one end and bent sharply to a high degree). In fact, Infusion Systems define the sensing parameter as "flex angle multiplied by radius".

2.6.2 Specifications

A typical bend sensor has the following basic specifications:

- range of deflection
- uni- vs. bi-directional sensing
- uni- vs. bi-polar sensing
- range of resistance (nominal to full-deflection)

Range of deflection: Determines the maximum angle of deflection that can be measured (as opposed to the maximum angle the sensor can be bent).

Uni- vs. bi-directional sensing: Some flexion sensors increase the resistance when bent in either of two opposing directions; however there is no difference in the measurement with respect to the direction.

Uni- vs. bi-polar sensing: A bi-polar flexion sensor measures deflection in two opposing directions yielding different measurements.

Range of resistance: Bend sensors can vary largely (even the same product) in terms of their range of resistance, measured as the difference from nominal resistance to resistance at full deflection.

2.6.3 Properties

- hysteresis/noise neglectible
- resistance is function of radius of curvature, not angle at one point
- high temperature and humidity-tolerance
- relatively low cost
- customizable (coatings, laminating materials)

2.6.4 Applications

- automotive applications
- industrial applications, e.g. safety switches, shipping, machine control
- medical applications (e.g. "SmartBed")
- gaming devices
- measuring devices
- assistive technology
- robotics

CHAPTER 3
METHODOLOGY

3.1 Hardware Connections

The project contains the hardware modules. The interfacing of this modules will play a key role in the development of this project.

3.1.1 Flex sensors Interfacing with Arduino Uno

A Flex Sensors are also called as Bend Sensor. Flex sensor measures the amount of pend. Usually, the amount of resistance is varied by bending the sensor. The Fig 3.1.1 shows the interfacing Flex sensors with Arduino uno. It is about 0.6cm wide and 8cm long and contains the two pins. The flex sensors 1,2,3 and 4 are connected to the Al, A2, A3 and A4 analog pins in the Arduino uno. The flex sensor 5 is connected to the Digital pin D7 of the Arduino Uno. These all- flex sensors are connected to the Arduino Uno Board by using the 1K Q resistors, which is a fixed value pulldown resistor. The main use of the pulldown resistor is to control the current flow to avoid damage to the Arduino Uno board.

The Fig 3.12 shows the interfacing Accelerometer sensor with Arduino uno, and accelerometer sensor is used to measure the acceleration (or) change in the velocity in X, Y, and Z-axis. Generally, these small sensors are utilized in cars and bikes to detect accidents to install the airbags and employed in mobile phones for a variety of applications like compass and location tracking.

3.1.2.1 Accelerometer Connections with Arduino Uno:

The below are the pin connections of Accelerometer with Arduino Uno

- The A4 pin (SDA) of Arduino Uno is connecting to the SDA pin of ADXL345

- The AS pin (SCL) of Arduino Uno is connecting to the SCL pin of ADXL345
- The GND of Arduino Uno is connecting to the GND pin of ADX1.345
- The 5V power supply of Arduino to the Vc of ADXL345

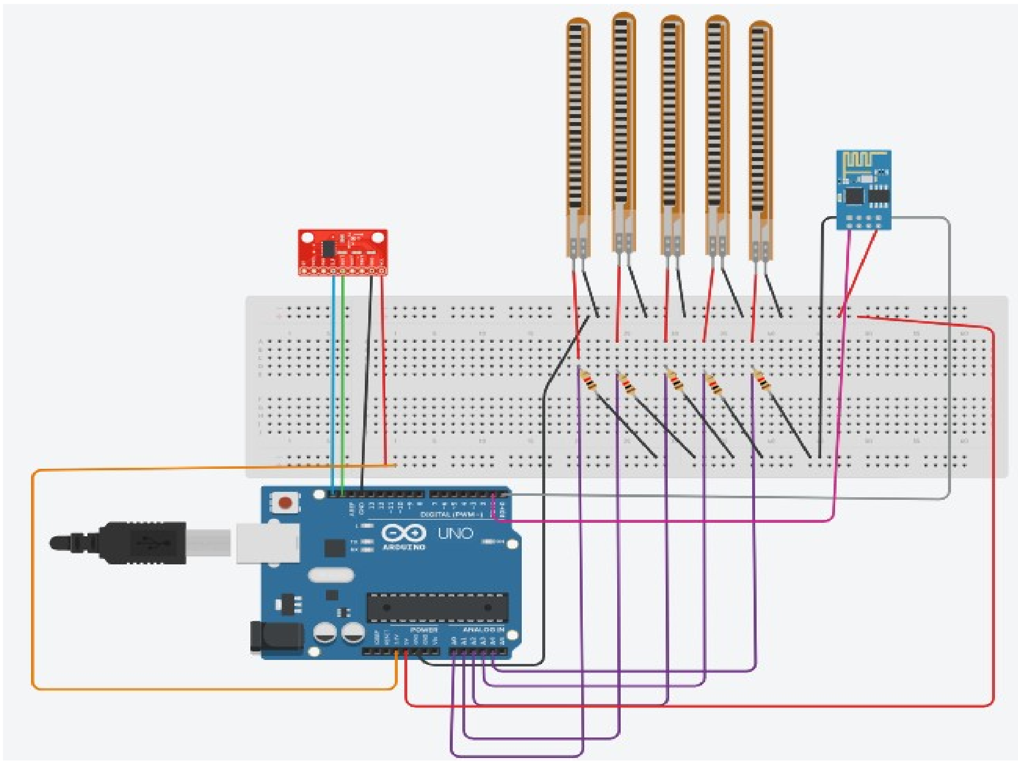


Fig 3.1.1 Interfacing sensors with Arduino uno

3.2 Working of Smart Glove

To achieve the objectives, the scope of this project is determined. For the hardware, Flex sensors will be used as a sensor to detect the bent of the gesture, and it sends a signal to Arduino Uno which acts as a microcontroller. The microcontroller processes the data and sends the signal to the voice module, which zuides the speaker.

The design of the circuit is done using Fritzing software and the progrars are run using Arduino software tool through Arduino IDE.

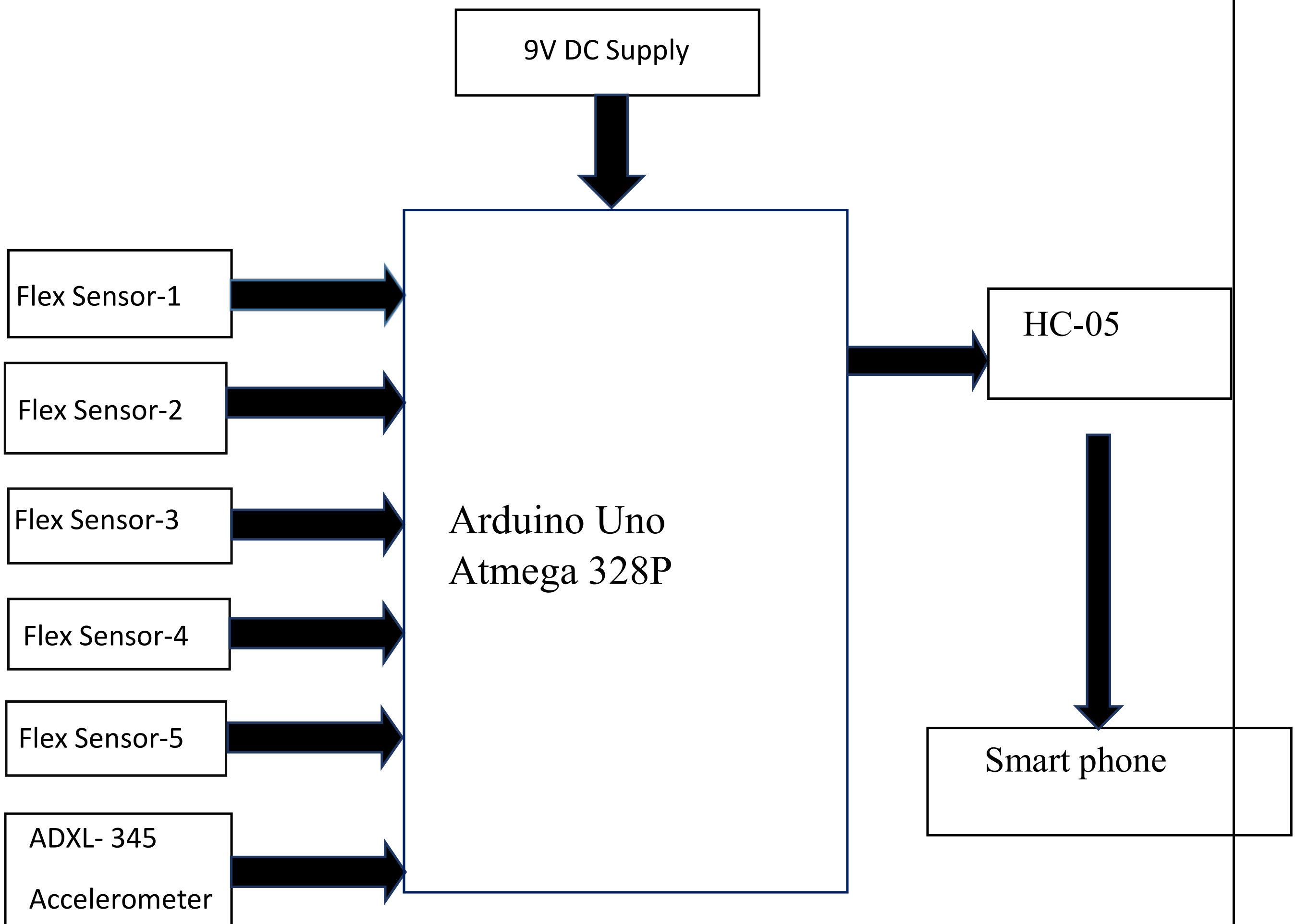


Fig 3.2 Block Diagram of Smart Glove

The Fig 3.2 shows the block diagram of smart glove system. Whenever user give the hand gestures, then the Arduino process the signal from the flex sensors and acceicrometer then gives output to the voice module. Further, it produces the voice related to i" hand gesture. It is helpful for deaf and dumb people to communicate with other people.

3.3 Flow Chart

The fig 3.3 shows the flow chart of smart glove. Whenever the hand gestures given by the user the sensors produce the output with respect to the input, these sensors are flex sensors and accelerometer sensors. Whenever the hand gesture is given the flex sensor produces the output voltage related to the input gesture. These values of flex sensors are given to the Arduino uno board, in the same way the accelerometer sensors sense the acceleration or change in velocity, then given to the Arduino Uno. The Arduino will process these signals and produces the commands related to the input gestures in the form of voice and display

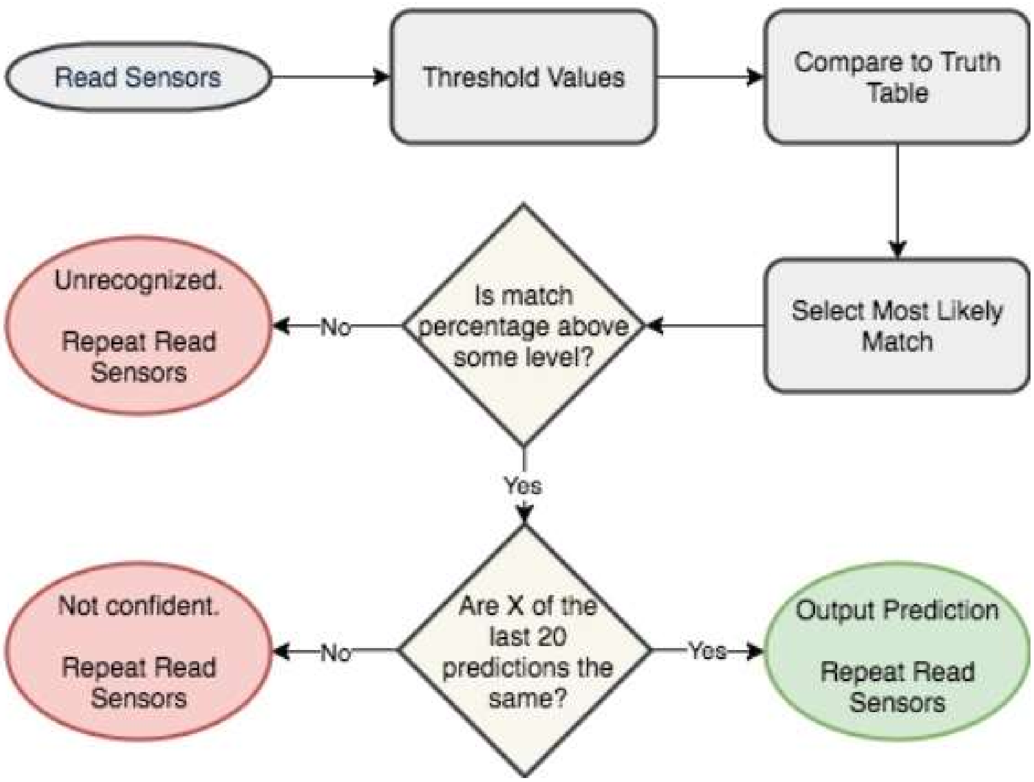


Fig 3.3 Flow Chart of Smart Glove

3.4 Experimental Hardware Setup

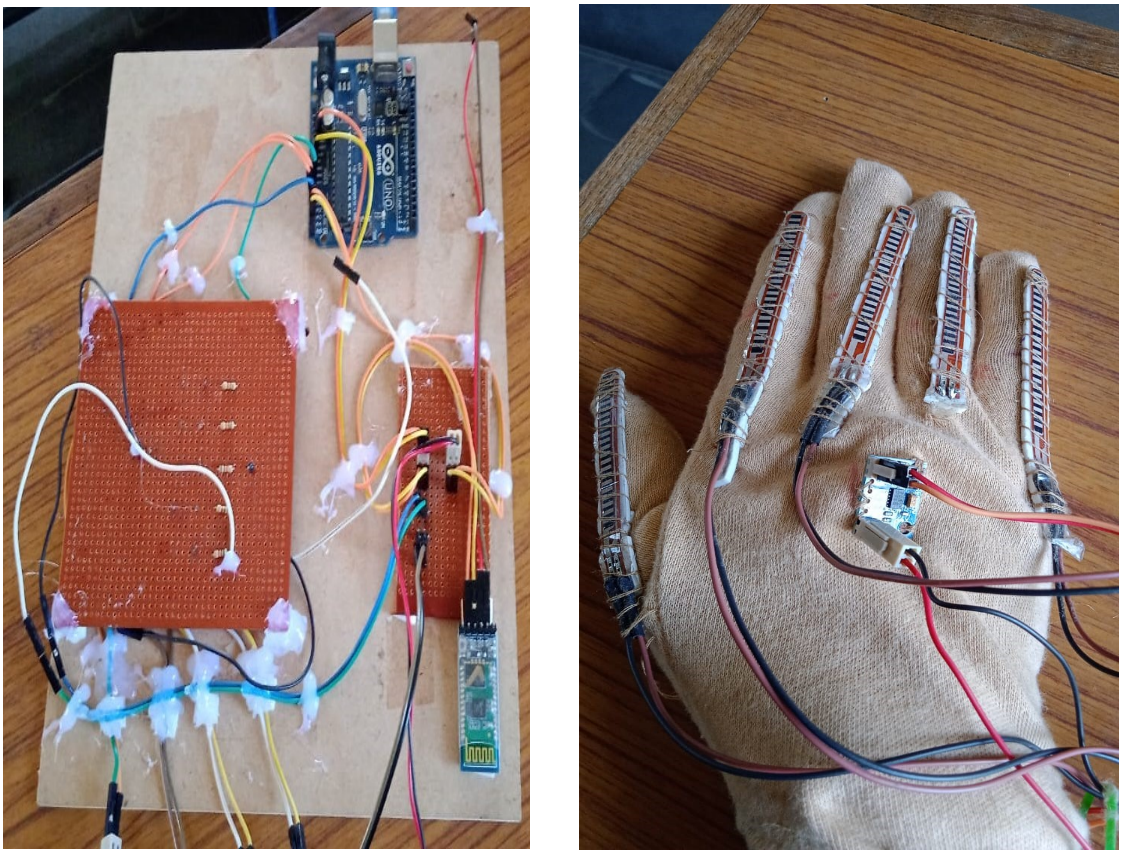


Fig 3.4 Experimental setup

The above fig 3.4 is the smart glove developed for the deaf and dumb people to communicate among them and also with normal people. In the system the sensors, Accelerometer sensor and flex sensors are connected/ mounted on the hand glove. The remaining all the modules are mounted on a wood board with the help of glue and double tape plaster. To use the smart glove user nees to wear the glove to the right hand

3.5 Conclusion

These are the conclusions based on our objectives:

- In conclusion, our final year project on sign language detection using flex sensors and ADXL sensor has shown promising results in improving communication between hearing and hearing-impaired individuals. We have developed a system that accurately recognizes and translates sign language gestures, and we plan to continue improving and expanding the system in the future.
- We believe that our system has the potential to make a significant impact on the lives of hearing-impaired individuals, allowing them to communicate more easily and effectively with others.

CHAPTER 4
RESULTS AND DISCUSSIONS

4.1 Introduction

In this project , all the components are attached to the glove and glove is as shown in the below figure 4.1. Flex sensors is attached to the glove such that the sensors measure the bent of the figures when hand gestures given. Hence the sensors can detect the bent of the hand produces the related output voltage

Fig 4.1 View of Smart Glove

Arduino uno board is the processor, which controls all the units that are connected to it . The Flex sensors and Accelerometer system which are integrated on the top of the glove . Which helps to provide the voice message and test message as output

4.2 Results

From all the information, we developed the circuit diagram and the embedded c code for Arduino Uno. The below are the commands which we included in the smart glove. Whenever user give hand gestures, the sensors produce the voltage which can be processed by the microcontroller, and each gesture provides the output as follows.

Table:

1. Alphabets:

Alphabet	Thumb		Index		Middle		Ring		Pinky				
	min	max	min	max	min	max	min	max	min	max	X	Y	Z
A	15	30	8	12	10	15	10	15	10	15	-8	-2	-3
B	10	15	20	28	20	28	20	28	20	28	-8	-2	-3
C	15	25	10	18	10	21	10	25	16	23	-6	-1	-3.5
D	13	20	21	30	8	15	10	14	8	15	-8	-2	-3
E	10	15	8	10	10	15	8	14	10	12	-8	-2	-3
F	10	20	7	10	24	30	23	30	25	30	-8	-2	-3
G	19	26	19	22	10	14	10	14	10	17	-6	-8	-2.5
H	16	22	19	22	24	30	10	14	10	17	-6	-8	-2.5
I	10	17	10	13	10	14	10	14	25	30	-8	-2	-3
J	10	17	10	13	10	14	10	14	25	30	-6	-9	-4.5
K	17	24	19	23	25	30	11	16	10	17	-8	-2	-3
L	13	20	18	26	10	15	10	15	10	14	-8	-2	-3
M	10	15	8	10	10	15	8	14	10	12	-6	-9	-4.5
N	13	18	8	10	10	15	8	14	10	12	-6	-9	-4.5
O	18	25	11	18	10	21	10	25	16	23	-6	-1	-3.5
P	15	26	18	25	10	15	10	15	8	15	3	-8	-5
Q	15	26	18	25	10	19	10	15	8	15	3	-8	-5
R	21	30	21	30	23	35	10	15	10	18	-8	-2	-3
S	15	30	8	12	10	15	10	15	10	15	-8	-2	-3
T	10	19	8	12	10	15	10	15	10	15	-8	-2	-3
U	10	20	7	10	24	30	23	30	25	30	-8	-2	-3
V	10	20	20	15	24	30	23	30	25	30	-8	-2	-3
W	17	24	20	30	20	30	20	30	10	15	-8	-2	-3
X	13	20	21	30	8	15	10	14	8	15	-8	-2	-3
Y	15	25	10	14	10	14	10	14	15	26	-8	-2	-3
Z	13	20	21	30	8	15	10	14	8	15	-8	2	5

2.Words:

MISTAKE	18	23	17	20	20	30	20	30	22	30	-8	-5	-4
NAME	17	24	19	23	22	30	10	14	10	15	-8	-5	-4
--- +THINK	10	15	15	23	10	15	10	15	10	15	-8	-5	-4
HELLO	10	15	20	28	20	28	20	28	20	28	-8	-2	-3
PLEASE	15	25	20	28	20	28	20	28	20	28	-8	-2	-3
I need water	20	25	10	15	10	15	10	15	10	15	-4	-3	-5
I am ready to do	20	25	8	10	10	15	10	15	10	15	-4	-7	-5

3.Numbers:

4.2 Applications of smart glove

The below are the applications of the smart glove

1. Useful for physically challenged (deaf and dumb) people.
2. Conveying information related operations for deaf and dumb people
3. Provide easy communication among the speech impaired people and the normal people

CHAPTER 5

CONCLUSION AND FUTURE SCOPE

5.1 Conclusion

It has been a major problem for deaf and dumb people to communicate among themselves and with outer world. To mitigate the above-mentioned problem, a system names "SENSOR BASED SIGN LANGUAGE TO SPEECH CONVERTER" was proposed. Smart Glove aims to solve real-world problems faced by dead and dumb people in their daily life and makes them communicate easily with others by using an Accelerometer, Flex Sensors, Arduino uno, Voice module, Speaker, and LCD. The Sensor Based sign language to speech converter was introduced by developing a circuit diagram and used Embedded C code for converting the electrical Signal to test messages and audio output. By giving the hand gestures which are Embedded in the System, the glove takes the input of all the voltages values from the hand movements and gives the output command accordingly. The benefit of the framework lies in the way that it can demonstrate to minimal expense answers for many deaf an dumb people around the world.

5.2 Future scope of the project

The future extent of the current glove manages the outwardly impeded individual who is deaf and dumb to communicate in a flexible and efficient way. The glove is specifically designed for a particular hand . As every hand differs in size, shape, and motion, The glove is flexible to work for any hand which is different size by giving it learning capabilities. In the proposed project, the output of the hand glove is displayed on Smart Phone. To improve portability, any Android application can also be developed for displaying the test and the speech output on an Android device. In another way, this algorithm can be implemented on a Raspberry Pi for compactness. This project is produces on large scale, in the future the entire circuit can be embedded into a single chip.

The project can also be modifies for the benefit of visually impaired children by adding ultrasonic sensors to the smart glove.

APPENDIX:

CHAPTER 6

REFERENCES: