SPEAKING SYSTEM FOR MUTE PEOPLE USING HAND GESTURES

A PROJECT REPORT
Submitted in Partial Fulfillment for the degree of
Bachelor of Technology in
ELECTRICAL ENGINEERING
From
Maulana Abul Kalam Azad University of Technology, West Bengal
(MAKAUT, WB)

Submitted by
RITABRATA CHAKRABORTY
NEHA PRASAD
ARITRA PAUL
TANMOY BAIDYA
MANISHA KUMARI
KSHITIJ SINGH

Under the guidance of
Mr. SANTANU MAITY

DEPARTMENT OF ELECTRICAL ENGINEERING
MCKV Institute of Engineering
243, G.T. Road (North), LILUAH, HOWRAH

MAY 2022

DEPARTMENT OF ELECTRICAL ENGINEERING
MCKV Institute of Engineering
Affiliated to
Maulana Abul Kalam Azad University of Technology, West Bengal
(MAKAUT, WB)

CERTIFICATE

This is to certify that the project report entitled ***"SPEAKING SYSTEM FOR MUTE PEOPLE USING HAND GESTURES "*** *submitted by* ***"RITABRATA CHAKRABORTY (11601618037), NEHA PRASAD(11601618049), ARITRA PAUL(11601618061), TANMOY BAIDYA(11601618014), MANISHA KUMARI(11601618052), KSHITIJ SINGH(11601618053)"*** *for 8th semester examination have been prepared following the guidelines of B.Tech degree in Electrical Engineering, awarded by the Maulana Abul Kalam Azad University of Technology, West Bengal (MAKAUT, WB).*

They have carried out the project work under my supervision.

Dr. Tamal Roy

HEAD OF THE DEPARTMENT

Electrical Engineering Department
(NBA Accredited),

MCKV Institute of Engineering
(NAAC Accredited "A" Grade Autonomous Institute, Affiliated to MAKAUT, W.B.)
243, G.T.Road(N), Liluah. Howrah-711204.

Mr. Santanu Maity

PROJECT GUIDE

Electrical Engineering Department
(NBA Accredited),

MCKV Institute of Engineering
(NAAC Accredited "A" Grade)
Autonomous Institute, Affiliated
To MAKAUT, W.B)
243, G.T.Road(N), Liluah, Howrah- 711204

TABLE OF CONTENTS

1) OBJECTIVE- It's very difficult for mute people to convey their message to regular people. Since regular people are not trained on hand sign language, the communication becomes very difficult. In emergency or other times when a mute person travelling or among new people communication with nearby people or conveying a message becomes very difficult. Here we propose a smart speaking system that help mute people in conveying their message to regular people using hand motions and gestures.

2) INTRODUCTION- This system makes use of a hand motion reading system equipped with motion and flex sensors along with a speaker unit. A raspberry pi is used for processing the data and operating the system. The system consists of around 10 stored messages like "need help", "where is the toilet/washroom" and so on that help mute people convey basic messages. The system reads persons hand motions for different variations of hand movement. It also consists of a trigger sensor in order to indate that the person wishes to activate the system and speak something. This ensures the system does not speak when the person is just involuntarily making hand motions. The raspberry pi processor constantly receives input sensor values and then processes it. Now it searches for matching messages for the set of sensor values. Once it is found in memory this message is retrieved and is spoken out using text to speech processing through the interfaced speaker. Thus, we have a fully functional smart speaking system to help mute people communicate with regular people using a simple wearable system.

3) REQUIRED COMPONENTS-

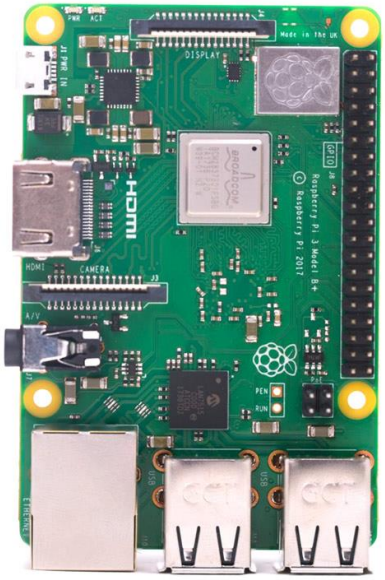
I. **Raspberry PI 3 Model B+**
II. **Flex Sensor**
III. **Accelerometer(ADXL345)**
IV. **Arduino UNO**
V. **Bluetooth module(HC-06)**
VI. **Battery 9v**
VII. **Resistor-1k & 10k**
VIII. **Speaker**
IX. **Gloves**

4) DETAILS OF COMPONENTS:-

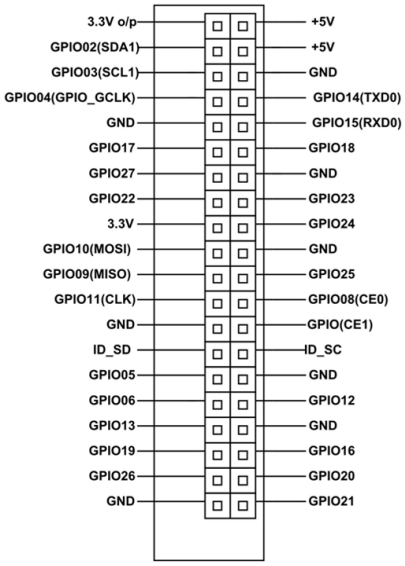
(4.1)RASPBERRY PI 3 MODEL B+

The Raspberry Pi is a low cost, credit-card sized computer that plugs into a computer monitor or TV, and uses a standard keyboard and mouse. It runs Linux, but it also provides a set of GPIO (general purpose input/output) pins, to control electronic components for physical computing and explore the Internet of Things (IoT).
The current generations of regular Raspberry Pi boards are Zero, 1, 2, 3, and 4. Basically, in this project we use Raspberry Pi 3 B+ model.

Raspberry Pi 3 B+:-



Raspberry Pi 3 B



Raspberry Pi 3 B pin out

Raspberry Pi-3 B Pin Configuration:-

PIN GROUP	PIN NAME	DESCRIPTION

POWER SOURCE	+5V, +3.3V, GND and Vin	+5V -power output +3.3V -power output GND – GROUND pin
COMMUNICATION INTERFACE	UART Interface(RXD, TXD) [(GPIO15,GPIO14)]	UART (Universal Asynchronous Receiver Transmitter) used for interfacing sensors and other devices.
SPI Interface(MOSI, MISO, CLK,CE) x 2 [SPI0-(GPIO10 ,GPIO9, GPIO11 ,GPIO8)] [SPI1--(GPIO20 ,GPIO19, GPIO21 ,GPIO7)]	SPI (Serial Peripheral Interface) used for communicating with other boards or peripherals.	
TWI Interface(SDA, SCL) x 2 [(GPIO2, GPIO3)] [(ID_SD,ID_SC)]	TWI (Two Wire Interface) Interface can be used to connect peripherals.	
INPUT OUTPUT PINS	26 I/O	Although these some pins have multiple functions they can be considered as I/O pins.
PWM	Hardware PWM available on GPIO12, GPIO13, GPIO18, GPIO19	These 4 channels can provide PWM (Pulse Width Modulation) outputs.
EXTERNAL INTERRUPTS	All I/O	In the board all I/O pins can be used as Interrupts.

Raspberry Pi 3 Technical Specifications:-

Microprocessor	Broadcom BCM2837 64bit Quad Core Processor
Processor Operating Voltage	3.3V

Raw Voltage input	5V, 2A power source
Maximum current through each I/O pin	16mA
Maximum total current drawn from all I/O pins	54mA
Flash Memory (Operating System)	16Gbytes SSD memory card
Internal RAM	1Gbytes DDR2
Clock Frequency	1.2GHz
GPU	Dual Core Video Core IV® Multimedia Co-Processor. Provides Open GLES 2.0, hardware-accelerated Open VG, and 1080p30 H.264 high- profile decode. Capable of 1Gpixel/s, 1.5Gtexel/s or 24GFLOPs with texture filtering and DMA infrastructure.
Ethernet	10/100 Ethernet
Wireless Connectivity	BCM43143 (802.11 b/g/n Wireless LAN and Bluetooth 4.1)
Operating Temperature	-40ºC to +85ºC

Board Connectors:-

Name	**Description**
Ethernet	Base T Ethernet Socket
USB	2.0 (Four sockets)
Audio Output	3.5mm Jack and HDMI
Video output	HDMI

Camera Connector	15-pin MIPI Camera Serial Interface (CSI-2)
Display Connector	Display Serial Interface (DSI) 15-way flat flex cable connector with two data lanes and a clock lane.
Memory Card Slot	Push/Pull Micro SDIO

NOOBS Software:-

NOOBS means new-out-of-box software and it is the easiest way to get started with the Raspberry Pi. It is easy to copy NOOBS to your SD or MicroSD card. Once copied, it provides us with a simple menu for installing various operating systems. It is an open-source software. There is an option to buy a card with NOOBS already installed on it.

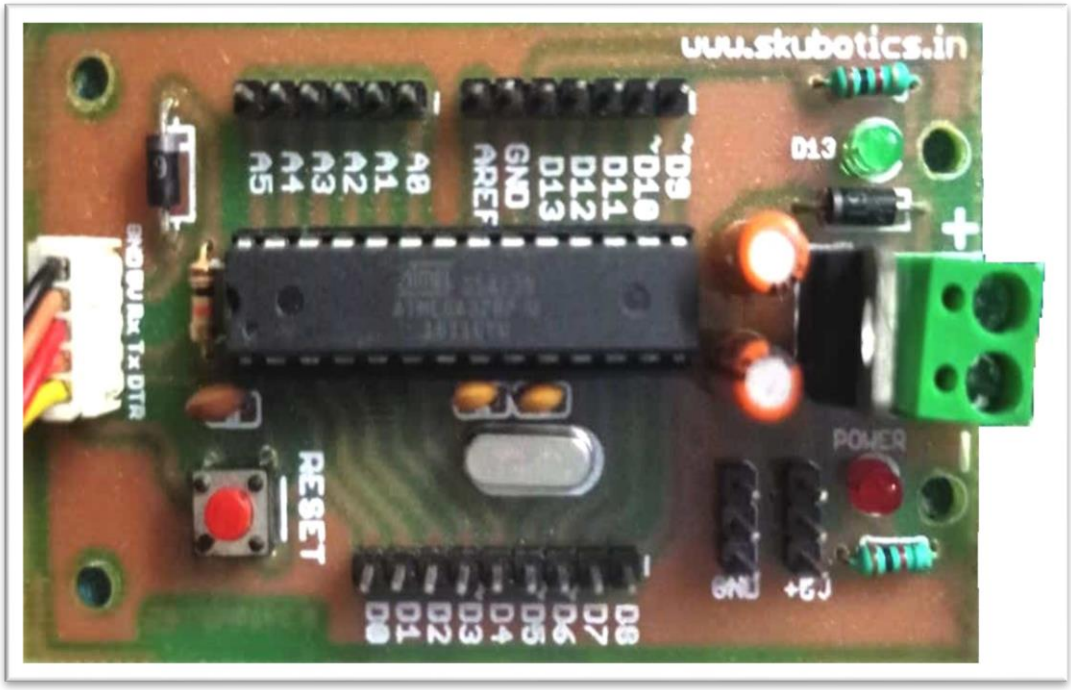
Applications of Raspberry Pi:-

The raspberry pi boards are used in many applications like Media streamer, Arcade machine, Tablet computer, home automation, Carputer, Internet radio, controlling robots, Cosmic Computer, hunting for meteorites, Coffee and also in raspberry pi-based projects.

(4.2)ARDUINO UNO

Basic Details:-

The Arduino UNO is a low-cost, flexible, and easy-to-use programmable open-source microcontroller board that can be integrated into a variety of electronic



projects. This board can be interfaced with other Arduino boards, Arduino shields, Raspberry Pi boards and can control relays, LEDs, servos, and motors as an output. Arduino UNO features AVR microcontroller Atmega328, 6 analogue input pins, and 14 digital I/O pins out of which 6 are used as PWM output.

This board contains a USB interface i.e. USB cable is used to connect the board with the computer and Arduino IDE (Integrated Development Environment) software is used to program the board.

The unit comes with 32KB flash memory that is used to store the number of instructions while the SRAM is 2KB and EEPROM is 1KB.

The operating voltage of the unit is 5V which projects the microcontroller on the board and its associated circuitry operates at 5V.

Components:-

The Arduino UNO board contains the following components and specifications:

1. **ATmega328:** This is the brain of the board in which the program is stored.
2. **Ground Pin:** there are several ground pins incorporated on the board.
3. **PWM:** the board contains 6 PWM pins. PWM stands for Pulse Width Modulation, using this process we can control the speed of the servo motor, DC motor, and brightness of the LED.
4. **Digital I/O Pins:** there are 14 digital (0-13) I/O pins available on the board that can be connected with external electronic components.
5. **Analogue Pins:** there are 6 analogue pins integrated on the board. These pins can read the analogue sensor and can convert it into a digital signal.
6. **Aref:** It is an Analog Reference Pin used to set an external reference voltage.
7. **Reset Button:** This button will reset the code loaded into the board. This button is useful when the board hangs up, pressing this button will take the entire board into an initial state.
8. **USB Interface:** This interface is used to connect the board with the computer and to upload the Arduino sketches (Arduino Program is called a Sketch)
9. **DC Power Jack:** This is used to power up the board with a power supply.
10. **Power LED:** This is a power LED that lights up when the board is connected with the power source.

11. **3.3V:** This pin is used to supply 3.3V power to our project.
12. **5V:** This pin is used to supply 5V power to our project.
13. **VIN:** It is the input voltage applied to the UNO board.
14. **SPI:** The SPI stands for Serial Peripheral Interface. Four Pins 10(SS), 11(MOSI), 12(MISO), 13(SCK) are used for this communication.
15. **TX/RX:** Pins TX and RX are used for serial communication. The TX is a transmit pin used to transmit the serial data while RX is a receive pin used to receive serial data.

..How to Program Arduino UNO

Arduino UNO is easy to program and a person with little or no technical knowledge can get hands-on experience with this board. The Arduino UNO board is programmed using Arduino IDE software which is an official software introduced by Arduino.cc to program the board. The Arduino program is called a sketch which you need to unload into the board. The sketch is nothing but a set of instructions that allow the board to perform certain functions as per your requirements.

Each Arduino sketch comes with two main parts:

void setup() – this sets up the things that need to be done once and they don't happen again in the running program.

void loop() – this part comes with the instructions that get repeated again and again until the board is turned off.

Why it is necessary for our project:-

In this project Arduino is used to process the input signal from from flex sensor,accelerometer and send the processed signal via Bluetooth module to the raspberry pi.

(4.3)FLEX SENSOR

Basic Details:-

A flex sensor is a kind of sensor which is used to measure the amount of defection otherwise bending. The designing of this sensor can be done by using materials like plastic and carbon. 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 a bend sensor. They are usually in the form of a thin strip from 1"-5" long that vary in resistance from approximately 10 to 50kiloohms. The specifications and features of this sensor include the following.

The features of this sensor are as follows:

- Operating voltage of this sensor ranges from 0V to 5V.
- It can function on low-voltages.
- Power rating is 1 Watt for peak & 0.5Watt for continuous.
- Operating temperature ranges from -45ºC to +80ºC
- Flat resistance is 25K Ω
- The tolerance of resistance will be ±30%

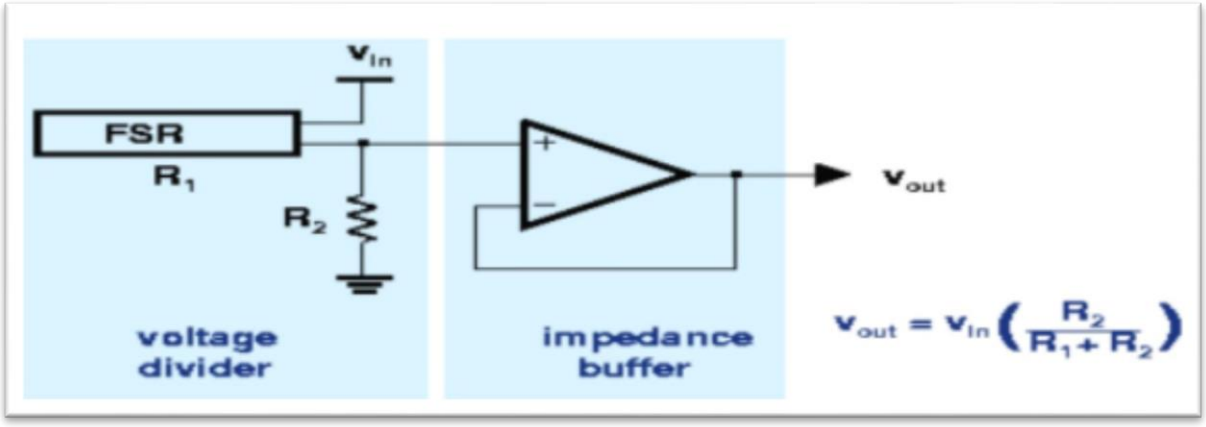
Configuration:-

The pin configuration of the flex sensor is shown below. It is a two-terminal device, and the terminals are like p1 & p2. This sensor doesn't contain any polarized terminal such as diode otherwise capacitor, which means there is no positive & negative terminal. The required voltage of this sensor to activate the sensor ranges from 3.3V -5V DC which can be gained from any type of interfacing.

Pin P1: This pin is generally connected to the +ve terminal of the power source.
Pin P2: This pin is generally connected to GND pin of the power source.

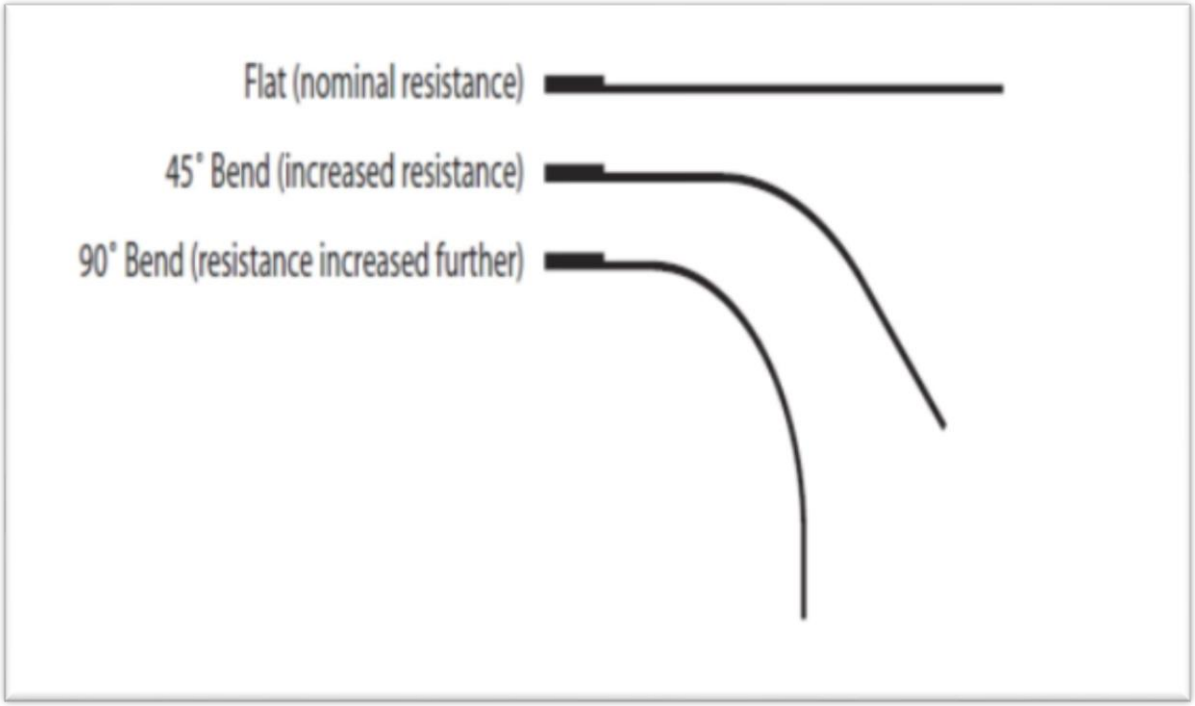
Basic Circuitry:-

Flex Sensor as Voltage Divider - Output voltage increases with the bend. The impedance buffer in the circuit reduces error due to source impedance of the flex sensor as voltage divider. Suggested op amps are the LM358 or LM324.



Working Principle:

This sensor works on the bending strip principle which means whenever the strip is twisted then its resistance will be changed. This can be measured with the help of any controller. This sensor works similar to a variable resistance because when it twists then the resistance will be changed. The resistance change can depend on the linearity of the surface because the resistance will be dissimilar when it is level. When the sensor is kept straight with no force acting on it, it has a resistance of around 10 K Ohms. With a typical flex sensor, a flex of 0 degrees will give 10K resistance will a flex of 90 will give 30-40 K ohms.



Necessity of Flex sensor in our project:-

We propose a smart speaking system that helps mute people in conveying their message to regular people using hand motions and gestures. The system makes

use of a hand motion reading system equipped with motion and flex sensors along with a speaker unit. If the Raspberry Pi could be interfaced with sensors it can get to know about the real-world parameters and interact with it which is the primary motive of our project for mute people. Flex Sensor is used along with a fixed resistance as a voltage divider and the analog values can be read by a micro controller. To connect the flex sensor to the Raspberry pi which as only digital pins. So, we cannot do it without an Analog to digital converter (ADC) which convert the analog values of flex senor into digital values and gives it to the raspberry pie.

(4.4)ACCELEROMETER(ADXL345)

Accelerometers are devices that measure acceleration, which is the rate of change of the velocity of an object. They measure in meters per second squared (m/s^2) or in G-forces (g). A single G-force for us here on planet Earth is equivalent to 9.8 m/s^2, but this does vary slightly with elevation (and will be a different value on different planets due to variations in gravitational pull). Accelerometers are useful for sensing vibrations in systems or for orientation applications.

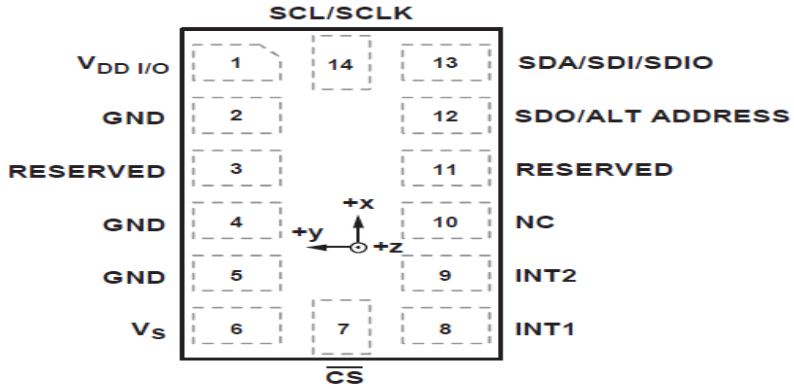
Basic Details:-

ADXL345 is a small, ultra-low power, 3-axis accelerometer sensor that is used in electronic control systems of mobile device applications like mobile handsets, smartphones, gaming devices, pointing devices, personal navigation devices, hard drive protection, medical and industrial instrumentation.
ADXL345 is a 3-axis MEMS capacitive digital accelerometer sensor. It has a user-selectable range up to +/-16g, a maximum output resolution of 13 bits, sensitivity of 3.9 mg/LSB, and a maximum output data rate of 3200 Hz. It is a digital sensor in a 14-lead package that outputs sensor data through I2C and SPI interfaces. ADXL345 measures static acceleration due to gravity as well as dynamic acceleration resulting from motion or shock. It can be used to sense linear acceleration in 3 axes, tilt and free fall of an object. The sensor also has two interrupt pins and supports built-in sensing functions like free-fall detection and tapping by mapping sensing functions to its interrupt pins. ADXL345 can detect the presence or lack of motion by comparing acceleration values to user-defined thresholds.
The ADXL345 sensor is a digital sensor, so it has several built-in registers that can be read and written over I2C or SPI. By reading and writing data to these

registers, a controller/computer can configure the sensor (like set different measurement range, data transfer rate, sensitivity, and resolution), read acceleration values, set thresholds, map interrupts, and use built-in sensing functions.
ADXL345 offers four measurement range: +/-2g, +/-4g, +/-8g, and +/-16g. The default measurement range is +/-2g, which can be used to sense acceleration up to 19.6 m/s^2 in either direction along each axis. The available resolutions are 10-bit for +/-2g, 11-bit for +/-4g, 12-bit for +/-8g and 13-bit for +/-16g range. The default resolution is 10-bit, which for +/-2g (default) range allows sensitivity of 3.9mg/LSB. The default data rate is 100 Hz.

Configuration:-

ADXL345 comes in a 14-lead package having the following pin diagram:



ADXL345 Accelerometer sensor has the following pin description:-

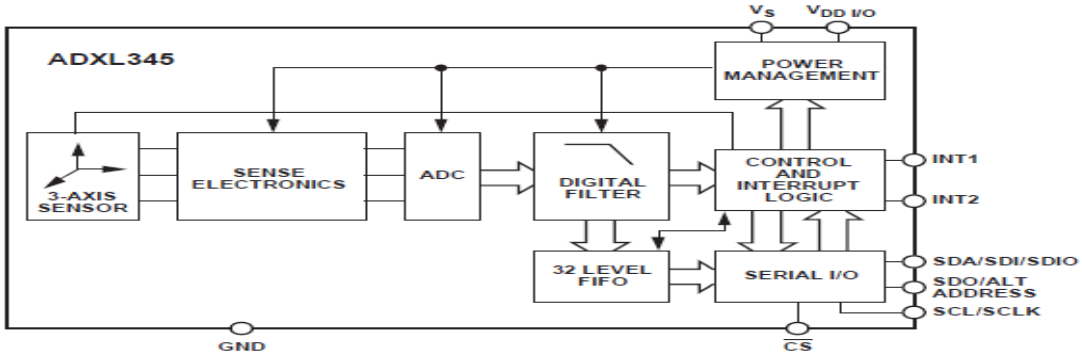
Pin No.	Pin Name	Pin Description
1	VDD I/O	Digital Interface Supply Voltage
2	GND	Ground
3	RESERVED	Reserved; Must be connected to Vs or left open.
4	GND	Ground
5	GND	Ground
6	VS	Supply Voltage
7	CS	Chip Select
8	INT1	Interrupt 1 Output
9	INT2	Interrupt 2 Output
10	NC	Not Internally Connected.
11	RESERVED	Reserved; Must be connected to Ground or left open.
12	SDO/ALT ADDRESS	Serial data output (SPI 4-wire)/Alternate I2C Address Select (I2C)
13	SDA/SDI/SDIO	Serial data (I2C)/Serial Data Input (SPI 4-Wire)/Serial Data Input Output (SPI 3-Wire)
14	SCL/SCLK	Serial Communications Clock. SCL (I2C)/SCLK (SPI)

Working Function of an ADXL345 Accelerometer:-

ADXL345 is a 3-axis accelerometer that senses both static acceleration (due to gravity) as well as dynamic acceleration (due to motion or shock). So, it can be used as a tilt sensor or to detect free fall. It is a MEMS accelerometer consisting of a polysilicon surface-micro-machined structure built on the top of a Polysilicon wafer. It is a capacitive accelerometer sensor. The polysilicon springs suspend the proof mass, and differential capacitors are used between the proof mass and fixed frame to measure the acceleration. Any acceleration along an axis deflects the proof mass and unbalances the differential capacitor, resulting in a sensor response that is directly proportional to the acceleration. Phase-sensitive demodulation is used to determine the magnitude and polarity of the acceleration.

The sensor can be interfaced with a controller/embedded computer using I2C or SPI interface. Using serial interfaces (I2C/4-wire SPI/3-wire SPI), a controller/computer can read and write to internal registers of the sensor.

Block Diagram of ADXL345 Accelerometer:-



FUNCTIONAL BLOCK DIAGRAM

The sensor has the following registers to which a controller/computer can read/write over serial interfaces:-

Register Address		Register Name	Type	Reset Value	Description
Hex	Dec				
0x00	0	DEVID	R	11100101	Device ID
0x01 to 0x1C	1 to 28	Reserved	-	-	Reserved; do not access;
0x1D	29	THRESH_TAP	R/W	00000000	Tap threshold
0x1E	30	OFSX	R/W	00000000	X-axis offset
0x1F	31	OFSY	R/W	00000000	Y-axis offset
0x20	32	OFSZ	R/W	00000000	Z-axis offset
0x21	33	DUR	R/W	00000000	Tap duration
0x22	34	Latent	R/W	00000000	Tap latency
0x23	35	Window	R/W	00000000	Tap window
0x24	36	THRESH_ACT	R/W	00000000	Activity threshold
0x25	37	THRESH_INACT	R/W	00000000	Inactivity threshold
0x26	38	TIME_INACT	R/W	00000000	Inactivity time
0x27	39	ACT_INACT_CTL	R/W	00000000	Axis enable control for activity and inactivity detection
0x28	40	THRESH_FF	R/W	00000000	Free-fall threshold
0x29	41	TIME_FF	R/W	00000000	Free-fall time
0x2A	42	TAP_AXES	R/W	00000000	Axis control for single tap/double tap
0x2B	43	ACT_TAP_STATUS	R	00000000	Source of single tap/double tap
0x2C	44	BW_RATE	R/W	00001010	Data rate and power mode control
0x2D	45	POWER_CTL	R/W	00000000	Power-saving features control
0x2E	46	INT_ENABLE	R/W	00000000	Interrupt enable control
0x2F	47	INT_MAP	R/W	00000000	Interrupt mapping control
0x30	48	INT_SOURCE	R	00000010	Source of interrupts
0x31	49	DATA_FORMAT	R/W	00000000	Data format control
0x32	50	DATAX0	R	00000000	X-Axis Data 0
0x33	51	DATAX1	R	00000000	X-Axis Data 1
0x34	52	DATAY0	R	00000000	Y-Axis Data 0
0x35	53	DATAY1	R	00000000	Y-Axis Data 1
0x36	54	DATAZ0	R	00000000	Z-Axis Data 0
0x37	55	DATAZ1	R	00000000	Z-Axis Data 1
0x38	56	FIFO_CTL	R/W	00000000	FIFO control
0x39	57	FIFO_STATUS	R	00000000	FIFO status

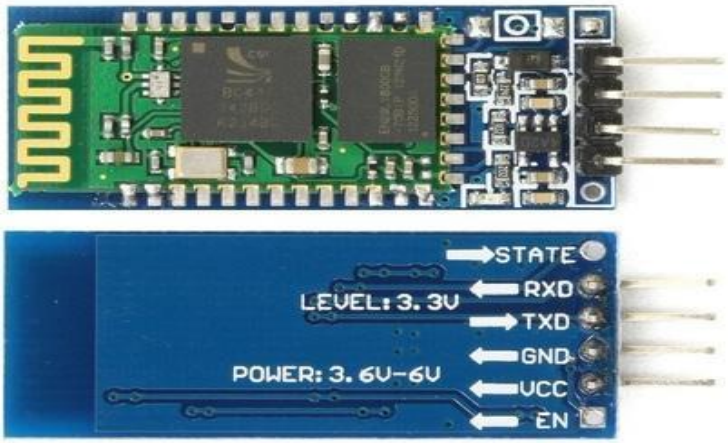
(4.5)BLUETOOTH MODULE

Basic Details:-

HC-06 is a Bluetooth module designed for establishing short range wireless data communication (<100 meters). It is very easy to interface and communicate. It can be interfaced with almost all microcontrollers or processors as it uses UART interface.

This module has the ability to transmit files at speed up to 2.1Mbps and works on Bluetooth 2.0 communication protocol. Unlike the HC-05 Bluetooth module, this module can only act as a slave device.

- Operating Voltage: 3.3V - 6V
- Operating Frequency range: 2.402 GHz - 2.480 GHz



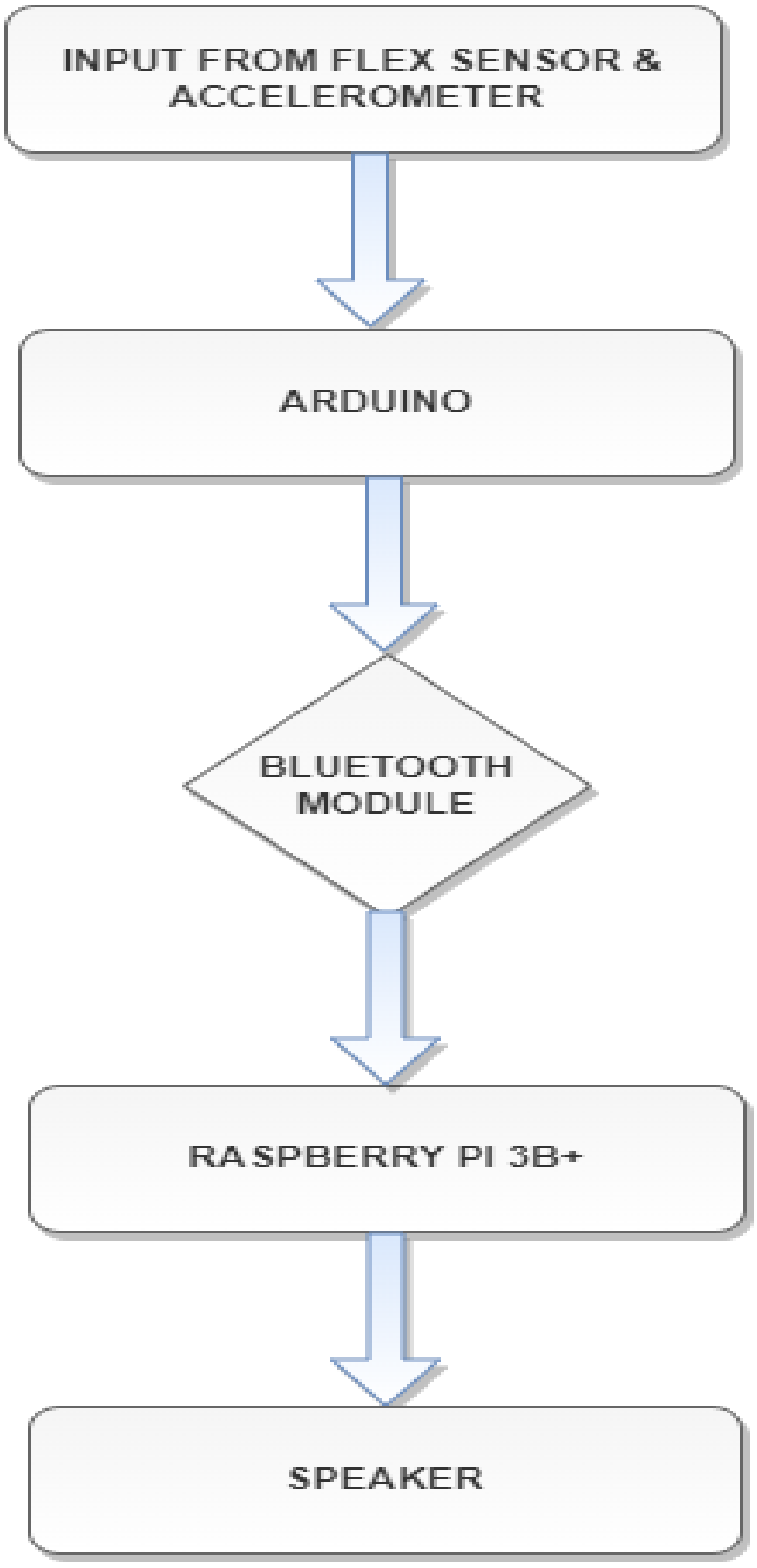
Configuration:-

- RXD: Serial Data Receive Pin. Used for serial input. 3.3V logic
- TXD: Serial Data Transmit Pin. Used for serial output. 3.3V logic
- GND: Ground
- VCC: +5V

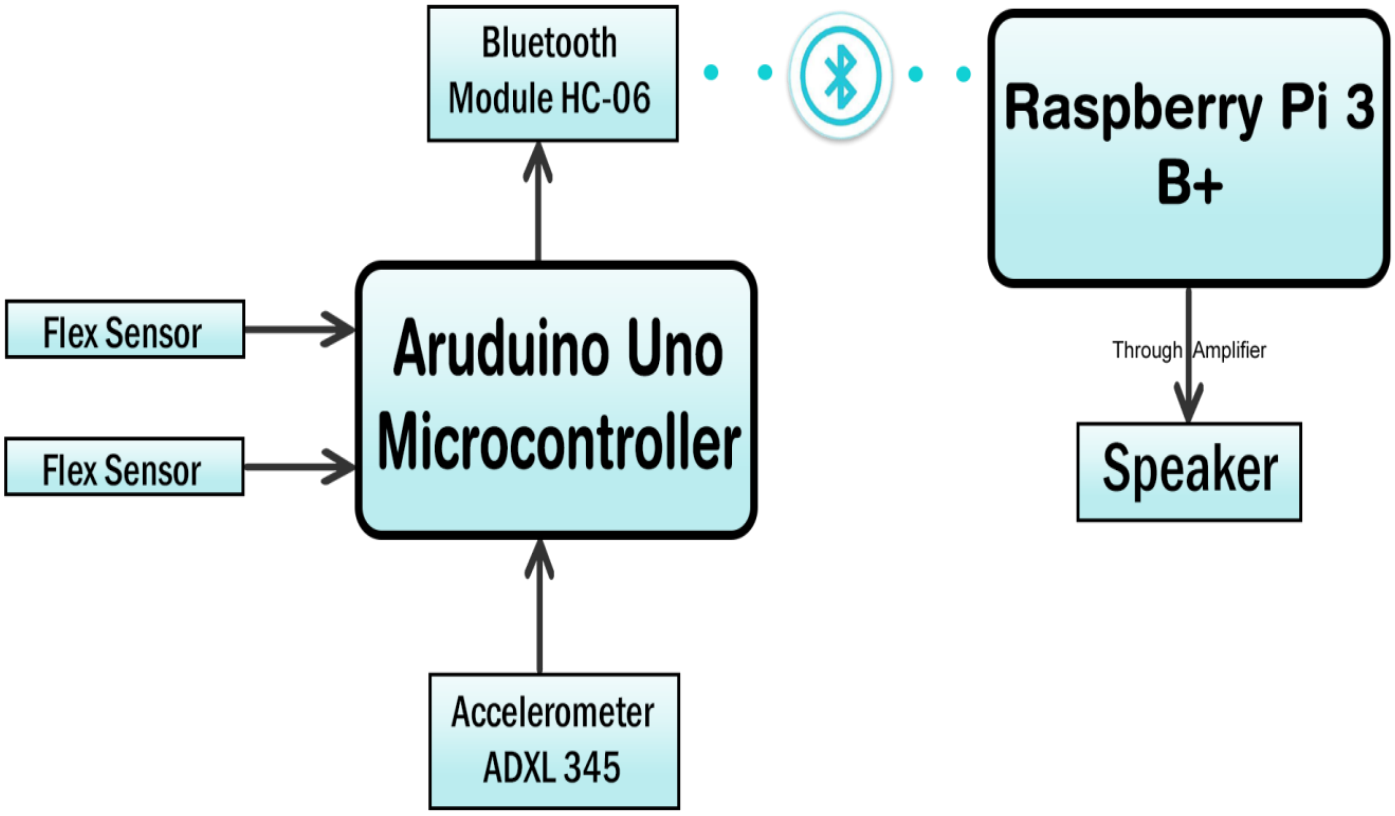
(4.6)SPEAKER CONNECTION

We use a 3-ohm, 5-watt speaker with 3.5 mm jack with it. It has a 3.5 mm female port to connect any type wired speaker or headphone. And we also can connect a bluetooth speaker and USB type speaker. As it has inbuilt bluetooth module and 4 USB female port.

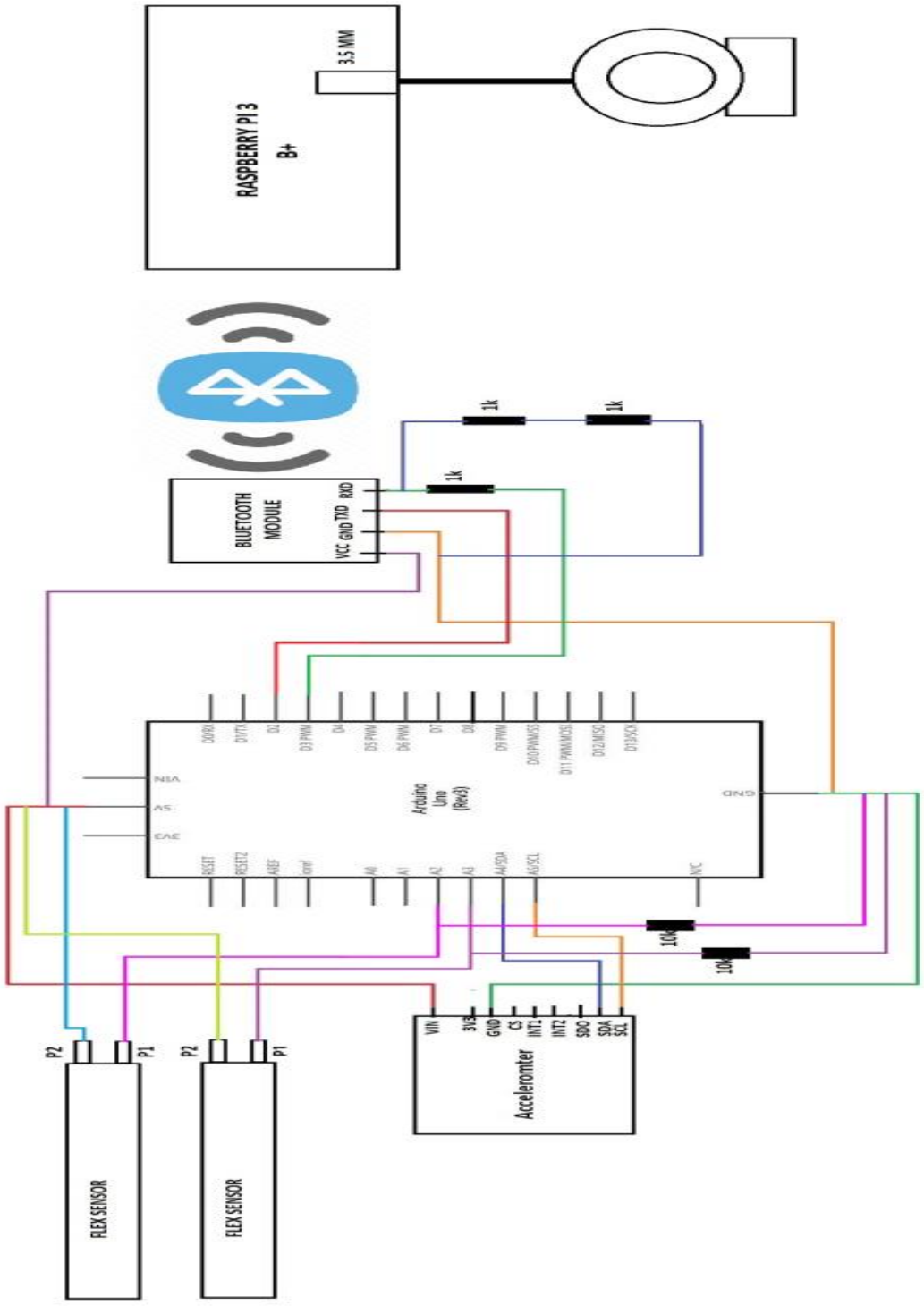
(4.7)FLOWCHART



4.8)BLOCK DIAGRAM



5)CIRCUIT DIAGRAM:-



6) CODES-

➢ *Arduino Code*

File Edit Sketch Tools Help

Open

Speaking_System_Arduino_code

```
#include "Arduino.h"
#include <Wire.h>
#include <Adafruit_Sensor.h>
#include <Adafruit_ADXL345_U.h>
#include <SoftwareSerial.h>

SoftwareSerial mySerial(2, 3);

Adafruit_ADXL345_Unified accel = Adafruit_ADXL345_Unified(12345);

void setup(void){
  Serial.begin(9600);
  mySerial.begin(9600);
    if(!accel.begin())  {
    Serial.println("Ooops, no ADXL345 detected ... Check your wiring!");
    while(1);
  }
  accel.setRange(ADXL345_RANGE_16_G);
}

void loop(void)
{
  sensors_event_t event;
  accel.getEvent(&event);
```

File Edit Sketch Tools Help

Speaking_System_Arduino_code

```
void loop(void)
{
  sensors_event_t event;
  accel.getEvent(&event);
  int x =event.acceleration.x;
  int y=event.acceleration.y;
  int z=event.acceleration.z;

  Serial.print("X: "); Serial.print(x); Serial.print(",");
  Serial.print("Y: "); Serial.print(y); Serial.print(",");
  Serial.print("Z: "); Serial.println(z);

  int fore_finger = analogRead(2);
  int middle_finger = analogRead(3);

  int ff = map(fore_finger, 880, 780, 0, 100);
  int mf = map(middle_finger, 880, 780, 0, 100);

  Serial.print("FF: "); Serial.print(ff); Serial.print(",");
  Serial.print("MF: "); Serial.println(mf);

 /*if(ff <3 && mf <22){
  Serial.println("I need water");
 }*/
```

File Edit Sketch Tools Help

New

Speaking_System_Arduino_code

```
/*if(ff <3 && mf <22){
 Serial.println("I need water");
}*/
   if(y>-5 && (x==-8 || x==-7) && z>=0 && ff>134 && mf>345){
    Serial.println("I need help");
    mySerial.print("1");
    delay(2000);
  }
  if((x==-8 || x==-9) && y<1 && z<-2 && y!=z && ff<150 && mf>349){
    Serial.println("I need food");
    mySerial.print("2");
    delay(2000);
  }
  if(x<-3 && y<x && y<z && ff>130 && mf>349){
    Serial.println("Call me");
    mySerial.print("3");
    delay(2000);
  }
  if(y>x && y==z && ff>120 && mf>330){
    Serial.println("I need water");
    mySerial.print("4");
    delay(2000);
  }
  if((x==1 || x==2) && y<x && y<z && ff>128 && mf>349){
```

File Edit Sketch Tools Help

Speaking_System_Arduino_code

```
  }
  if((x==-8 || x==-9) && y<1 && z<-2 && y!=z && ff<150 && mf>349){
    Serial.println("I need food");
    mySerial.print("2");
    delay(2000);
  }
  if(x<-3 && y<x && y<z && ff>130 && mf>349){
    Serial.println("Call me");
    mySerial.print("3");
    delay(2000);
  }
  if(y>x && y==z && ff>120 && mf>330){
    Serial.println("I need water");
    mySerial.print("4");
    delay(2000);
  }
  if((x==1 || x==2) && y<x && y<z && ff>128 && mf>349){
    Serial.println("I need medicine");
    mySerial.print("5");
    delay(2000);
  }

  delay(1000);
}
```

➢ *Python Code*

project.py

G: > Project > project.py

```python
from bluetooth import *
import binascii
from num2words import num2words
from subprocess import call


# eSpeak is a compact open source software speech synthesizer for English and other languages
cmd_beg= 'espeak '
cmd_end= ' 2>/dev/null'

server_sock = BluetoothSocket( RFCOMM )
bd_addr = "00:18:E4:40:00:06"
port = 1

server_sock.connect((bd_addr,port))
data = ""

while True:
    # socket.recv(recv_size)
    data = server_sock.recv(1024)
    if len(data) != 0 :
        _str = str(data)
        _str = _str.replace("b","")
        _str = _str.replace("'","")

        print (_str)
        data = ""

        if(_str == '1'):
            text = "i need water"
```

project.py 3

G: > Project > project.py > ...

```python
        _str = _str.replace("'","")

        print (_str)
        data = ""

        if(_str == '1'):
            text = "I need help"
            # Replacing ' ' with '_' to identify words in the text entered
            text = text.replace(' ', '_')
            call([cmd_beg+text+cmd_end], shell=True)
        if(_str == '2'):
            text = "I need food"
            text = text.replace(' ', '_')
            call([cmd_beg+text+cmd_end], shell=True)
        if(_str == '3'):
            text = "Call me"
            text = text.replace(' ', '_')
            call([cmd_beg+text+cmd_end], shell=True)
        if(_str == '4'):
            text = "I need water"
            text = text.replace(' ', '_')
            call([cmd_beg+text+cmd_end], shell=True)
        if(_str == '5'):
            text = "I need medicine"
            text = text.replace(' ', '_')
            call([cmd_beg+text+cmd_end], shell=True)

server_sock.close()
print ("all done")
```

7) FUTURE SCOPE-

Communication between mute and a normal person have always been a difficult task. Manufacturers around the world have formulated various sign language systems but they are not adaptable and cost effective for all the people. By using hand motions and gestures we put forth a smart speaking system which will help deaf people to communicate with normal people.

8) CONCLUSION-

In this project work, the sign language will be more helpful for the ease of communication between the mute people and normal people. The project mainly aims at reducing the gap of communication between the mute people and normal people. Here the methodology intercepts the mute signs into speech. In this system it overcomes the difficulties faced by mute people and helps them in improving their manner. The projected system is very easy to carry to any places when compared to existing systems. So, it will be very helpful.

9) REFERENCES-

I. **https://projects.raspberrypi.org/en/projects/raspberry-pi-getting-started/3**
II. **https://create.arduino.cc/projecthub/RucksikaaR/interfacing-the-hc-06-bluetooth-module-with-arduino-f9c315**
III. **https://howtomechatronics.com/tutorials/arduino/how-to-track-orientation-with-arduino-and-adxl345-accelerometer/**
IV. **https://lastminuteengineers.com/flex-sensor-arduino-tutorial/**
V. **https://www.rs-online.com/designspark/what-is-arduino-uno-a-getting-started-guide**