1. **Introduction**
   1. **Background**

Since their existence, humans have been physically and mentally challenged with various impediments that hinder their advancement on all economic, political, and social levels. Statistics around the world show a relatively high rate of people with speaking difficulties like **Stuttering,** **Deaf-mute**. Rates extend to an average of 9% of the whole population, whether in the Middle East, Europe, Africa, America, or other continents of the world.

It would be unjust for such people to be excluded from society just because they lack elementary communication means. They are mentally capable individuals who deserve and are even demanded to play an effective role in society. Society cannot dismiss whatever potential they have, simply because it needs it; for it is known that society advances only with the collective effort of all of its members. To this end, technology is utilized to aid humans in their collective effort to overcome such impediments

* 1. **Problem Statement**

Literate and employed deaf and dumb population face the problem in communicating with normal person at work, home etc. It is a result of the physical disability of hearing for deaf people and disability of speaking for dumb people that yields to lack of communication between a normal person and a deaf or a dumb person. It becomes the exact problem between two persons who knows different language but none of them are able to communicate with each other and so they require a translator which may not be always convenient to arrange and this kind of problem occurs in between a normal person and a deaf or a dumb person.

1. **Overview and Planning**

**2.1 Proposed System Overview**

The aim of the project “**SMART GLOVES**” is to provide a practical way of translating sign language into speech, offering people with vocal disabilities a means of communication with people incapable of understanding sign language. Firstly, we surveyed the existing communication aid systems and then found a cheap and easy way for interpreting the gestures to help deaf and dumb people communicate with ease. A motion capture system is used for sign language conversion. It captures the signs and displays as text on the screen.

**2.2 Challenges**

We will interpret the 26 alphabets of English language with specific sign and also use this system to convert the gestures to their respective meanings like food, ok, fine and display the meaning. Hardware components which we have used for this purpose are flex sensors, accelerometer, LCD and Arduino (microcontroller).

**2.3 Budget and Cost Estimation**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S.NO | COMPONENT | QUANTITY | COST PER PIECE(in Rs) | TOTAL COST (in Rs) |
| 1 | Arduino | 1 | 500 | 500 |
| 2 | Flex sensor | 5 | 500 | 2500 |
| 3 | Accelerometer | 1 | 180 | 180 |
| 4 | LCD | 1 | 100 | 100 |
| 5 | Connecting wires | 1 | 100 | 100 |
| 6 | Breadboard | 1 | 150 | 150 |
|  |  |  |  |  |
|  |  |  | **TOTAL** | **Rs.3530** |

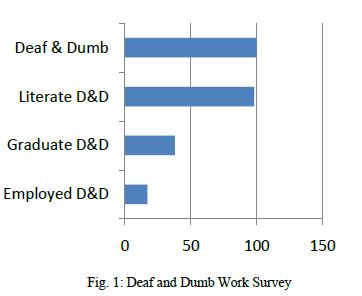
Total Cost = Cost of components + Indirect cost

= Rs 3530 + Rs 370 (approx.) = Rs 3900

1. **Literature Survey & Summary**

Communication is the only medium by which we can share our thoughts or convey the message but for a person with disability (deaf and dumb) faces difficulty in communication with normal person. Because of this, a person who lacks in hearing and speaking ability is not able to stand in race with normal person. Communication for a person who cannot hear is visual, not auditory. Generally dumb people use sign language for communication but they find difficulty in communicating with others who don’t understand sign language.

India constitutes 2.4 million of Deaf and Dumb population, which holds the world‘s 20% of the Deaf and Dumb Population. This person lacks the amenities which a normal person should own. The big reason behind this is lack of communication as deaf people are unable to listen and dumb people are unable to speak. Fig. 1 shows a survey analysis.



The sign language is an important and only method of communication for deaf-dumb persons. As sign language is a formal language employing a system of hand gesture for communication (by the deaf).

**Realistic Constraints:**

Economic:

* Cost effective system which can give voice to voiceless person with the help of Smart Gloves.
* Total estimated cost of product is below Rs. 3500 /-

Health and Safety:

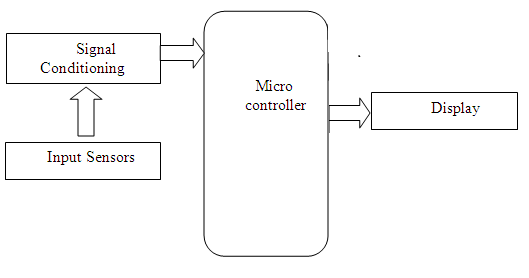
* The system works on 5v battery and there won’t be any electric shocks.
* Device doesn’t get heated up on continuous use.

Social and Ethical:

* The device is a gloves and which is a commonly used thing by humans when they travel.
* LCD Display is one of the way device can communicate with others.

1. **System Design**

**4.1 Block Diagram**

****

**4.2 Hardware Specifications**

|  |  |  |
| --- | --- | --- |
| Serial Number | Type of Component | Specification |
| 1 | Glove | Hand glove |
| 2 | Microcontroller | Arduino Uno |
| 3 | Flex Sensors | 4.5” |
| 4 | LCD | 16x2 |
| 5 | Accelerometer | ADXL335 |

**4.2.1 Arduino UNO:**

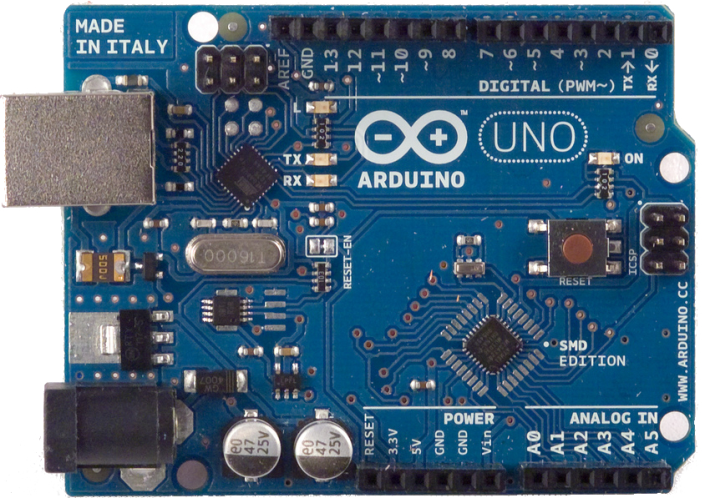


Fig. 1 Arduino Uno microcontroller Courtesy: Google images

The Uno is a microcontroller board based on the [ATmega328P.](http://www.atmel.com/Images/doc8161.pdf)It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.

**Power**

The Uno board can be powered via the USB connection or with an external power supply. The power source is selected automatically.

External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack.

The Uno has 6 analog inputs, labeled A0 through A5, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and the analogReference() function.

**Technical specs**

|  |  |
| --- | --- |
| Microcontroller | [ATmega328P](http://www.atmel.com/Images/doc8161.pdf) |
| Operating Voltage | 5V |
| Input Voltage (recommended) | 7-12V |
| Input Voltage (limit) | 6-20V |
| Digital I/O Pins | 14 (of which 6 provide PWM output) |
| PWM Digital I/O Pins | 6 |
| Analog Input Pins | 6 |
| DC Current per I/O Pin | 20 mA |
| DC Current for 3.3V Pin | 50 mA |
| Flash Memory | 32 KB (ATmega328P) of which 0.5 KB used by bootloader |
| SRAM | 2 KB (ATmega328P) |
| EEPROM | 1 KB (ATmega328P) |
| Clock Speed | 16 MHz |
| Length | 68.6 mm |
| Width | 53.4 mm |
| Weight | 25 g |



Fig. 2 Pin diagram of Atmega328 Courtesy: Google images

**Key Parameters**

|  |  |
| --- | --- |
| PARAMETER | VALUE |
| Flash(Kb) | 32Kb |
| Pin Count | 32 |
| Max. Operating Frequency(MHz) | 20MHz |
| CPU | 8-bit AVR |
| Max. I/O Pins | 23 |
| External Interrupts | 24 |

**4.2.2 FLEX SENSORS:**

****

Fig.3 Flex sensor Courtesy: Google images

**Flexion sensors** 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
* fibre-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).

**Mechanical Specifications:**

* **Life cycle:**> 1 million bends
* **Height:**< 0.44mm
* **Temperature range:** -35°Cto +80 °C

**Electrical Specifications:**

* **Flat resistance:** 25K Ohms
* **Resistance tolerance:** 30%
* **Bend resistance:** 45K – 125K Ohms
* **Power rating:** 1 watt (peak)

**Schematics**

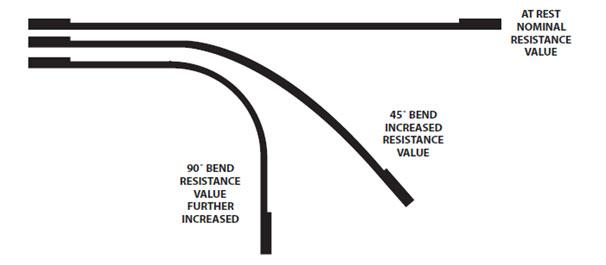
****

Fig. 4 working of flex sensor Courtesy: Google images

**4.2.3 ACCELEROMETER:**

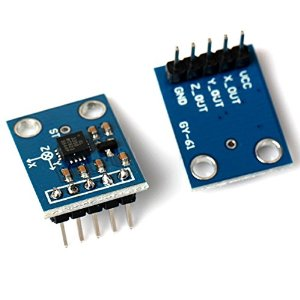
****

Fig. 5 Accelerometer Courtesy: Google images

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/s2) or in G-forces (g). A single G-force for us here on planet Earth is equivalent to 9.8 m/s2, 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.

**FEATURES**

* 3-axis sensing
* Small, low profile package
* 4 mm × 4 mm × 1.45 mm LFCSP
* Low power : 350 μA (typical)
* Single-supply operation: 1.8 V to 3.6 V
* 10,000 g shock survival
* Excellent temperature stability

**Pin Function Descriptions**

|  |  |  |
| --- | --- | --- |
| PIN NO. | PIN NAME | DESCRIPTION |
| 1 | VCC | Supply Voltage.1.8-5v |
| 2 | X-OUT | X Channel Output |
| 3 | Y-OUT | Y Channel Output |
| 4 | Z-OUT | Z Channel Output |
| 5 | GND | Supply Ground |

**Specifications**

* **Operating voltage range:** 1.8 - 5 V
* **Supply current:** 350uA
* **Operating temperature:** -40° °C to +85°C
* **Interfaces:** Analog
* **Dimensions:** 20.3mm×15.7mm×11.6mm
* **No. of axis:** 3

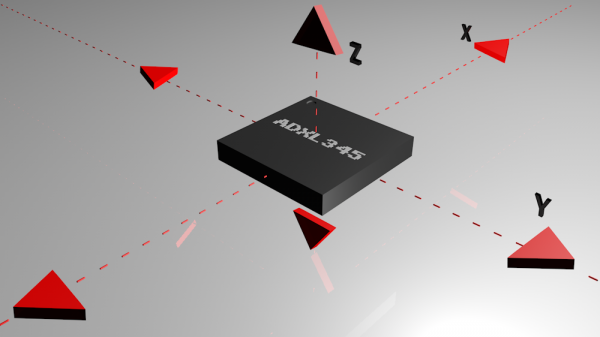
**[](https://cdn.sparkfun.com/assets/6/7/e/5/a/516c6b6ece395f0f49000000.jpeg)**

Fig. 6Axis measurement of accelerometer Courtesy: Google images

## How an Accelerometer Works

Accelerometers are electromechanical devices that sense either static or dynamic forces of acceleration. Static forces include gravity, while dynamic forces can include vibrations and movement.

Accelerometers can measure acceleration on one, two, or three axes. 3-axis units are becoming more common as the cost of development for them decreases.

* + 1. **LIQUID CRYSTAL DISPLAY (LCD)[4]:**

****

Fig. 7 LCD [16\*2] Courtesy: Google images

Alphanumeric displays are used in a wide range of applications, including palmtop computers, word processors, photocopiers, point of sale terminals, medical instruments, cellular phones, etc. The 16 x 2 intelligent alphanumeric dot matrix display is capable of displaying 224 different characters and symbols.

**Specifications**

* **Power voltage:** 0 – 7 V
* **Input voltage:** VSS/VDD
* **Operating temperature:** 0°C to +50°C
* **Storage temperature:** -20°C to +60°C

**Pin Diagram**



Fig. 8 LCD pin diagram Courtesy: Google images

**[](http://www.circuitsgallery.com/wp-content/uploads/2014/10/Arduino-hello-world.png)**

Fig .9 LCD interfacing Courtesy: project

**Pin Specifications**

|  |  |  |
| --- | --- | --- |
| PIN NO | NAME | FUNCTION |
| 1 | VSS | Ground pin |
| 2 | VCC | Power supply pin of 5V |
| 3 | VEE | Used for adjusting the contrast commonly attached to the potentiometer. |
| 4 | RS | RS is the register select pin used to write display data to the LCD (characters), this pin has to be high when writing the data to the LCD. During the initializing sequence and other commands this pin should low. |
| 5 | R/W | Reading and writing data to the LCD for reading the data R/W pin should be high (R/W=1) to write the data to LCD R/W pin should be low (R/W=0) |
| 6 | E | Enable pin is for starting or enabling the module. A high to low pulse of about 450ns pulse is given to this pin. |
| 7 | DB0 |  |
| 8 | DB1 |  |
| 9 | DB2 |  |
| 10 | DB3 |  |
| 11 | DB4 | DB0-DB7 Data pins for giving data(normal data like numbers characters or command data) which is meant to be displayed |
| 12 | DB5 |  |
| 13 | DB6 |  |
| 14 | DB7 |  |
| 15 | LED+ | Back light of the LCD which should be connected to Vcc |
| 16 | LED- | Back light of LCD which should be connected to ground. |

1. **System Implementation**

**5.1 LCD interfacing with Arduino UNO:**

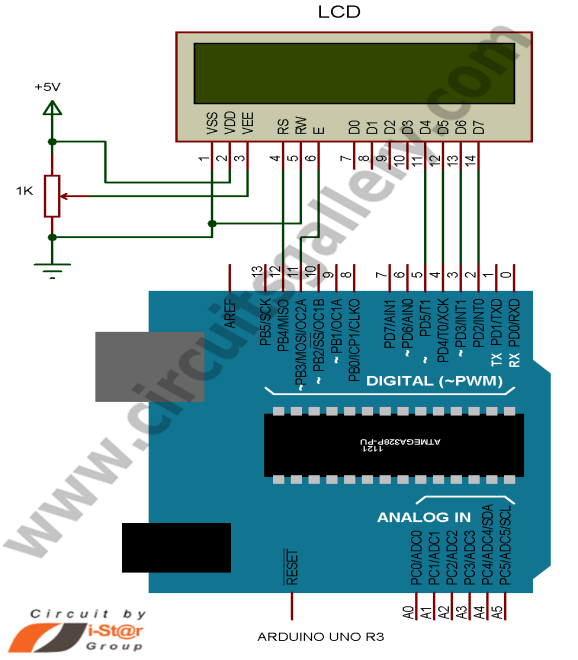
[](http://www.circuitsgallery.com/wp-content/uploads/2014/10/Interfacing-LCD-with-Arduino-.png)

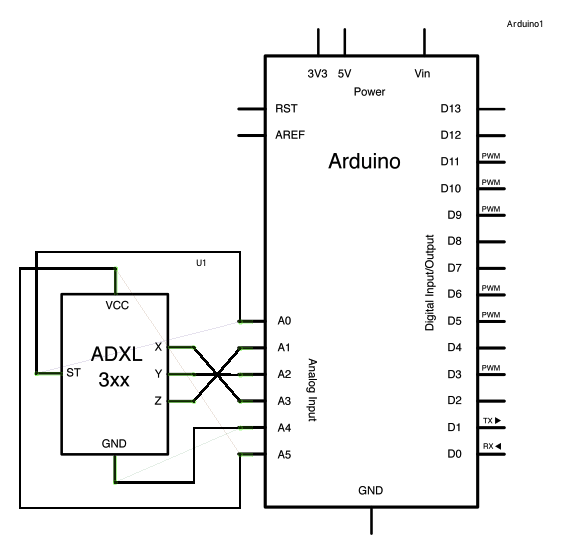
Fig. 10 Interfacing of LCD with Arduino Courtesy: Google images

Demonstrates the use a 16x2 LCD display. The LiquidCrystal library works with all LCD displays that are compatible with the Hitachi HD44780 driver. There are many of them out there, and you can usually tell them by the 16-pin interface.

This sketch prints "Hello World!" to the LCD and shows the time.

|  |
| --- |
| *The circuit:*  *\* LCD RS pin to digital pin 12*  *\* LCD Enable pin to digital pin 11*  *\* LCD D4 pin to digital pin 5*  *\* LCD D5 pin to digital pin 4*  *\* LCD D6 pin to digital pin 3*  *\* LCD D7 pin to digital pin 2*  *\* LCD R/W pin to ground*  *\* LCD VSS pin to ground*  *\* LCD VCC pin to 5V*  *\* 10K resistor:*  *\* ends to +5V and ground*  *\* wiper to LCD VO pin (pin 3)*  *// include the library code:*  *#include <LiquidCrystal.h>*  *// initialize the library with the numbers of the interface pins*  *LiquidCrystal lcd(12, 11, 5, 4, 3, 2);*  *void setup() {*  *// set up the LCD's number of columns and rows:*  *lcd.begin(16, 2);*  *// Print a message to the LCD.*  *lcd.print("hello, world!");*  *}*  *void loop() {*  *// set the cursor to column 0, line 1*  *// (note: line 1 is the second row, since counting begins with 0):*  *lcd.setCursor(0, 1);*  *// print the number of seconds since reset:*  *lcd.print(millis() / 1000);*  *}* |

**5.2 Accelerometer interfacing with Arduino UNO:**



|  |
| --- |
| *The circuit:  analog 0: accelerometer self test  analog 1: z-axis  analog 2: y-axis  analog 3: x-axis  analog 4: ground  analog 5: vcc*  *// these constants describe the pins. They won't change:* const int groundpin = 18;             *// analog input pin 4 -- ground* const int powerpin = 19;              *// analog input pin 5 -- voltage* const int xpin = A3;                  *// x-axis of the accelerometer* const int ypin = A2;                  *// y-axis* const int zpin = A1;                  *// z-axis (only on 3-axis models)*  void setup() {   *// initialize the serial communications:*   Serial.begin(9600);    *// Provide ground and power by using the analog inputs as normal*   *// digital pins.  This makes it possible to directly connect the*   *// breakout board to the Arduino.  If you use the normal 5V and*   *// GND pins on the Arduino, you can remove these lines.*   pinMode(groundpin, OUTPUT);   pinMode(powerpin, OUTPUT);   digitalWrite(groundpin, LOW);   digitalWrite(powerpin, HIGH); }  void loop() {   *// print the sensor values:*   Serial.print(analogRead(xpin));   *// print a tab between values:*   Serial.print("\t");   Serial.print(analogRead(ypin));   *// print a tab between values:*   Serial.print("\t");   Serial.print(analogRead(zpin));   Serial.println();   *// delay before next reading:*   delay(100); } |

* 1. **Flex Sensors interfacing with Arduino UNO:**

A flex sensor is a plastic strip with a conductive coating. When the strip is straight, the coating will be a certain resistance. When the strip is bent, the particles in the coating get further apart, increasing the resistance. You can use this sensor to sense finger movement in gloves.

Hardware connections:

The flex sensor has two pins, and since it's a resistor, the pins are interchangeable.

Connect one of the pins to ANALOG IN pin 0 on the Arduino.

int flexSensorPin = A0; //analog pin 0

void setup(){

Serial.begin(9600);

}

void loop(){

int flexSensorReading = analogRead(flexSensorPin);

Serial.println(flexSensorReading);

//In my tests I was getting a reading on the Arduino between 512, and 614.

//Using map(), you can convert that to a larger range like 0-100.

int flex0to100 = map(flexSensorReading, 512, 614, 0, 100);

Serial.println(flex0to100);

delay(250); //just here to slow down the output for easier reading

}