VIT UNIVERSITY
(Estd. u/s 3 of UGC Act 1956)
VELLORE ▪ CHENNAI
www.vit.ac.in

School of Electronics Engineering (SENSE)

"J" COMPONENT REVIEW-II REPORT

COURSE CODE / NAME	ECE3003 – MICROCONTROLLERS & ITS APPLICATIONS
PROGRAM / YEAR / SEM	B.Tech (ECE/ECM)/II Year / Fall 2018-19
DATE OF REVIEW	27th March 2019
J TITLE	MOTION BASED MESSAGE CONVEYER FOR PARALYTIC PATIENTS

TEAM MEMBERS DETAILS	**REGISTER NO.**	**NAME**
	17BEC1151	ANKUR MUKHERJEE
	17BEC1164	AMAN SHAJI.
	17BEC1225	SURYA V.
	17BLC1126	KARAN.K.SHAH

EVALUATION ITEMS	**Yes (√) / No (x)**
The project has achieved the objective set for this point?	
Level of Knowledge Gained While Completing the Project was satisfactory?	
Are the students having clear idea on their proposed and have they acquired to carry forward it?	
Are the contribution made by the individuals towards attaining objective of the project was satisfactory?	
Are the submitted report and presentation made by each team member was satisfactory?	

COURSE INCHARGE NAME	**Prof. V. PRAKASH**	**MARKS**	
REVIEWER'S NAME & SIGN			

Objective of the Project:

- **To help the paralytic or disabled people to communicate with their physios or an attendant while they need them .**

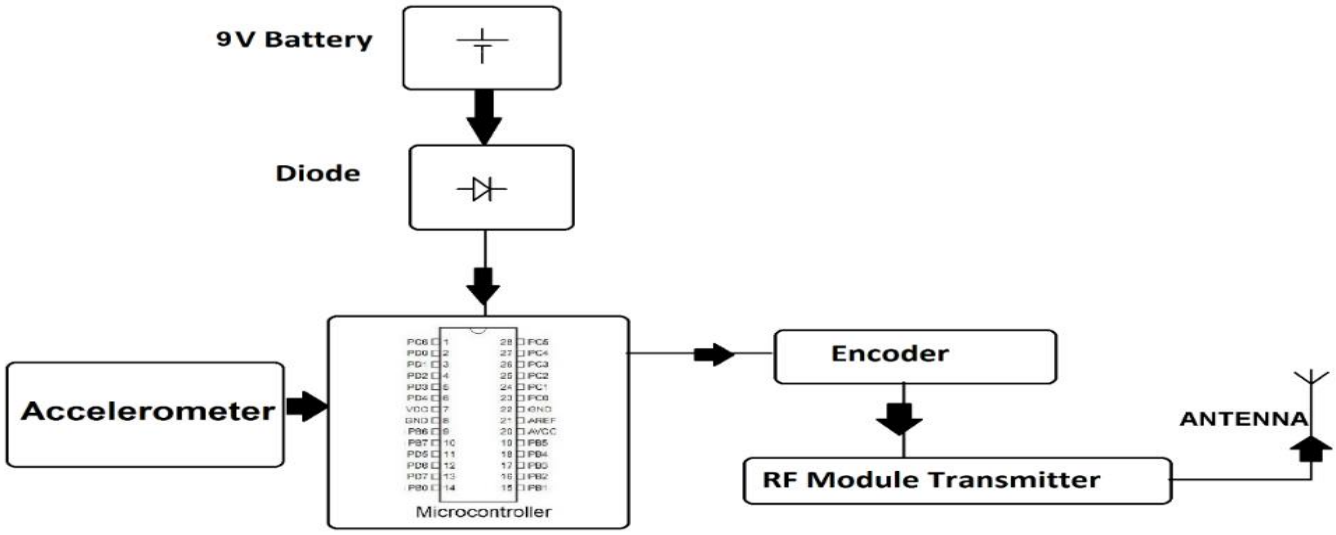
Components Required:

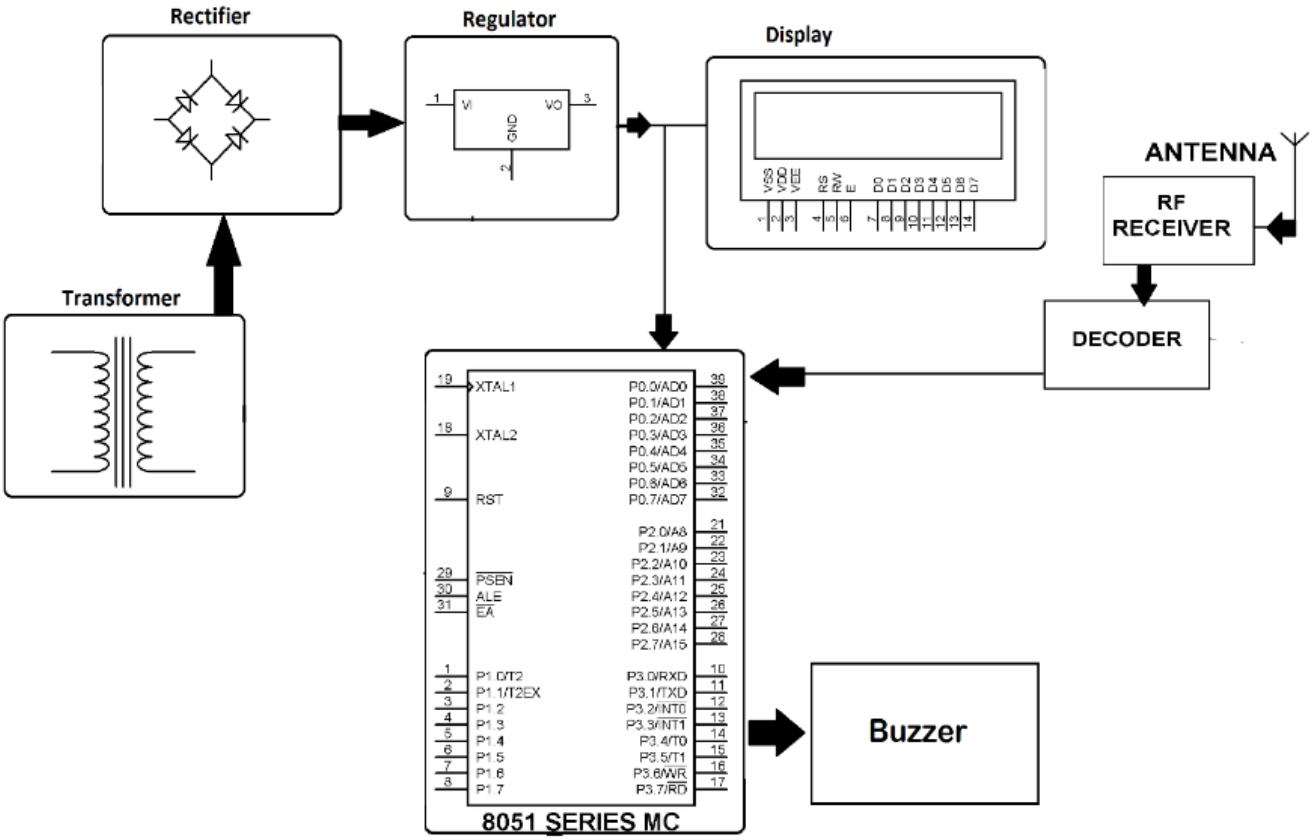
S. No	Component Name	Specification	Quantity	Cost (in Rs.)
1	8051 MICROCONTROLLERS	89C51	1	45.00
2	RADIO FREQUENCY MODULE	433MHz	1	200.00
3	LCD	16X2	1	150.00
4	ACCELEROMETER	ADXL355	1	330.00
5	BUZZER MINI	-	1	12.00
6	IC DECODER	HT-12D	1	30.00
7	IC ENCODER	HT-12E	1	30.00
8	CRYSTAL OSCILLATOR	XTAL	1	8.00
9	CAPACITORS	-	1	2.00
10	ARDUINO NANO	-	1	450.00

Overall cost of the Project: 1257 (in Rupees)

Block Diagram:

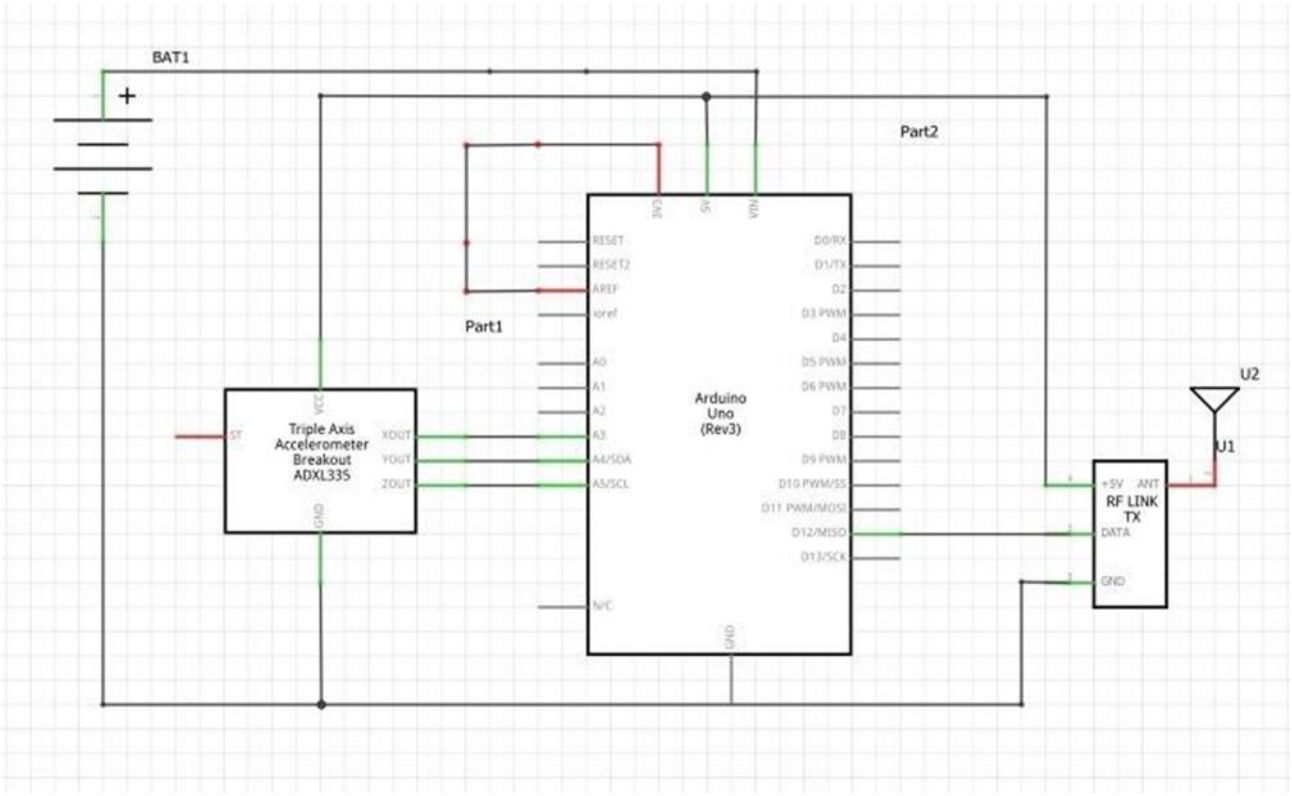
Transmitter



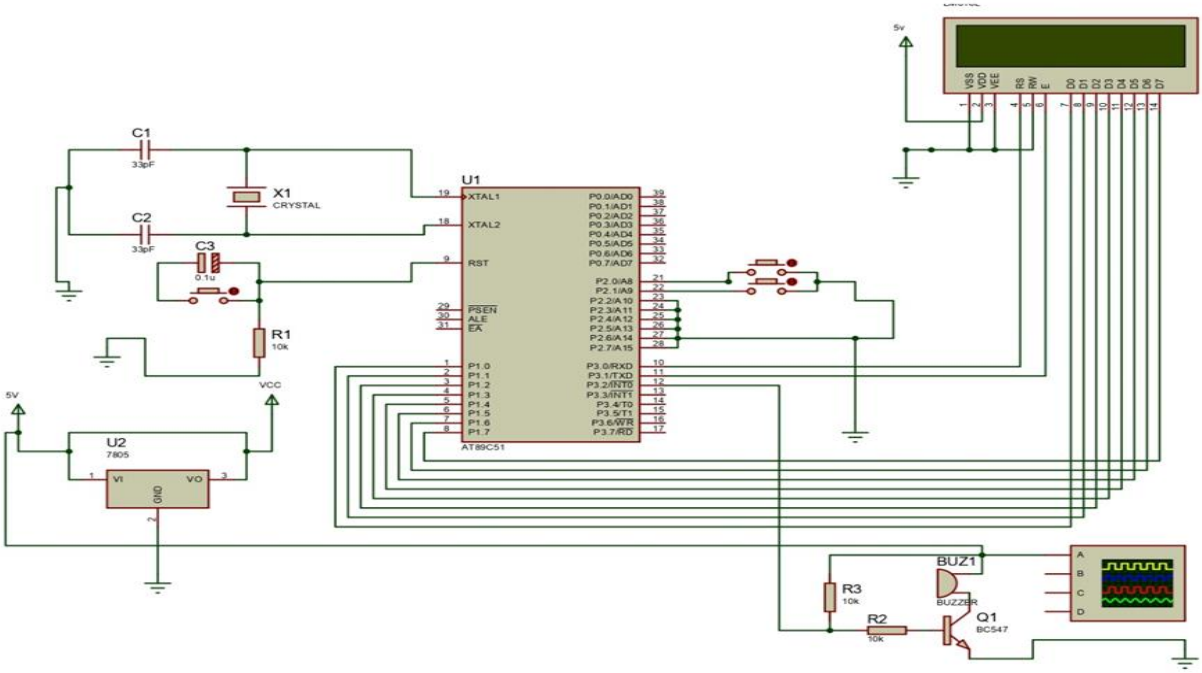


Schematic Diagram:

Transmitter



Receiver

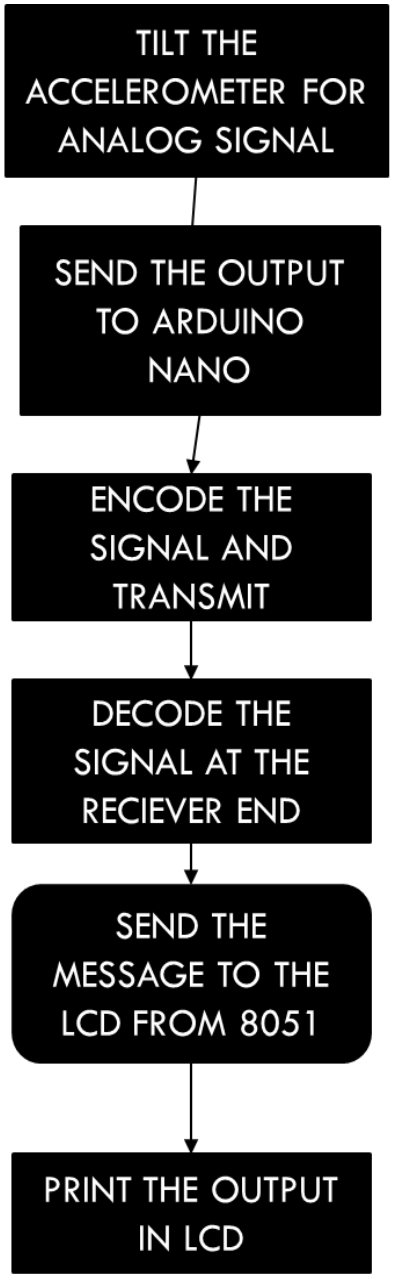


Project Description:

We got inspired from the late Stephan Hawking on how he communicated to everyone though he was paralyzed. Our project would work more efficiently for those paralytic people who cannot communicate using their voice and can at least move their limbs to an extent. We place a sensor on one of their moving limbs, and for each movement, a message would be displayed on the LCD and different sounds would be transmitted from the buzzer for different movements.

First the sensor reads the signal and since the signal is analog, we convert this signal to digital using Arduino Nano. This Arduino Nano acts like an Analog to Digital convertor and this is connected to a microcontroller viz a decoder. This decoder decodes the digital output and sends it to the LCD and buzzer via the microcontroller. The LCD then displays the message for a movement and the buzzer transmits a sound for that movement.

Flow chart:



Program:

:Arduino code:

```
const int xpin = A0;

const int ypin = A1;

const int zpin = A2;


void setup() {


  pinMode(5,OUTPUT);

  pinMode(4,OUTPUT);

  pinMode(3,OUTPUT);

  pinMode(2,OUTPUT);



  Serial.begin(9600);

}


void loop() {

 int x = analogRead(xpin);

 delay(200);


int y = analogRead(ypin);

 delay(200);



int z = analogRead(zpin);

 delay(200);
```

```
Serial.print(x);
Serial.print("\t");


Serial.print(y);
Serial.print("\n");


digitalWrite(5,LOW);
digitalWrite(4,LOW);
digitalWrite(3,LOW);
digitalWrite(2,LOW);

 if ((x<330)&&(x>300))
 {
    digitalWrite(5,HIGH);

    digitalWrite(4,LOW);
    digitalWrite(3,LOW);
    digitalWrite(2,LOW);
 }

  else if ((x<430)&&(x>400))
 {
   digitalWrite(4,HIGH);
```

```
   digitalWrite(5,LOW);
   digitalWrite(3,LOW);
   digitalWrite(2,LOW);
  }
  else if ((y<330)&&(y>290))
  {
   digitalWrite(3,HIGH);

   digitalWrite(5,LOW);
   digitalWrite(2,LOW);
   digitalWrite(4,LOW);
  }
  else if ((y<440)&&(y>410))
  {
   digitalWrite(2,HIGH);

   digitalWrite(5,LOW);
   digitalWrite(3,LOW);
   digitalWrite(4,LOW);
  }
}
```

:Microcontroller code:

```
#include<reg51.h>
/* DEFINITIONS OF PORTS
LCD - PORT1(datalines),
                        PORT3 .0,1(rs, enable)
BUZZER - PORT3.2
FROM DECODER - PORT2
```

```
*/
#define lcd P1
#define input P2 //output from decoder.
sbit buz = P3^2;                //buzzer
int c, count,count2;
sbit rs=P3^0;                   //led rs and enable pin
sbit e=P3^1;
sbit bu=P3^2;                           //buzzer
sbit led1=P3^3;                 //red
sbit led2=P3^4;                 //blue

void delay (int);
void cmd (char);
void display (char);
void custom (void);
void string (char *);
void init (void);
void MSDelay (unsigned int);

//BUZZER
//creates frequencies for different c, d
void call_delay (int c,int d){
//creates a buzzer for a 12 second time period
   TMOD = 0x01;
        TH0 = c;                                //0Xf8 ,0Xfa ,0XFB  ,0XFC
        TL0 = d;                                //0x80 ,0X00,0X00   ,0X00
        TR0 = 1;
        while (TF0 == 0);
```

```
        TF0 = 0;
        TR0 = 0;
}
void buzz (int c, int d){
        count=200;
         while(count! =0)
                        {
                                bu = ~bu; call_delay(c, d);
                                count=count-1;
                        }
}


//LCD DISPLAY
//delay for LCD
void delay (int d) {
        unsigned char i;
        for(;d>0;d--)
        {
                for(i=250;i>0;i--);
                for(i=248;i>0;i--);
        }
}
//for Using commands of lcd

void cmd (char c){
        lcd=c;
        rs=0;
```

```
        e=1;
        delay(5);
        e=0;
}
//display characters
void display (char c){
        lcd=c;
        rs=1;
        e=1;
        delay(5);
        e=0;
}
//reads string into separate characters and calls display function
void string (char *p){
        while(*p)
        {
                display(*p++);
        }
}
//initializes lcd
void init (void){
        cmd(0x38);
        cmd(0x0c);
        cmd(0x01);
        cmd(0x83);//line 1 position 3
}

void ambulance(void){
```

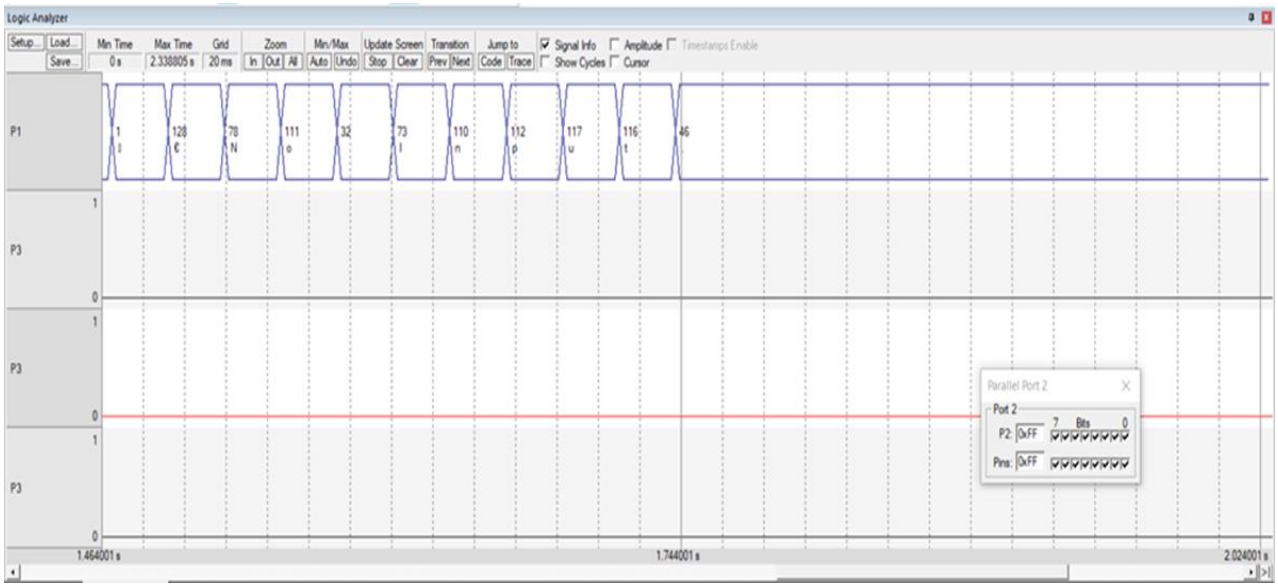
```
	count2=10;
	while(count2!=0)
			{
				led1=1;
				buzz(0xfe,0xb2);				//BUZZER freq1 3000 Hz LED1
				led1=0;
				led2=1;
				buzz(0xfe,0x5f);				//BUZZER freq2 2400Hz LED2
				led2=0;
				count2=count2-1;
			}
}
void main()
{
	P1 = 0x00;
	P3= 0x00;
	init();
		string("WELCOME.");
		delay(200);

		cmd(0x01);
		cmd(0x80);

	while(1)
	{
		if(input==0x01)	//emergency
		{
```

```
                cmd(0x01);                              //clears screen
                cmd(0x83);                              //Sets to position 3 of line 1
of display

                string("EMERGENCY !!");         //display
                ambulance();                    //create an ambulance signal
                cmd(0x01);                              //clear screen

        }
        else if(input==0x02)    //food
        {
                cmd(0x01);
                cmd(0x80);

                buz = 1;//BUZZER
                string("I NEED FOOD.");
                delay(200);

                cmd(0x01);
        }
        else if(input==0x04)    //water
        {
                cmd(0x01);
                cmd(0x80);
                string("I NEED WATER.");

                buz = 1;//BUZZER
                delay(200);
```

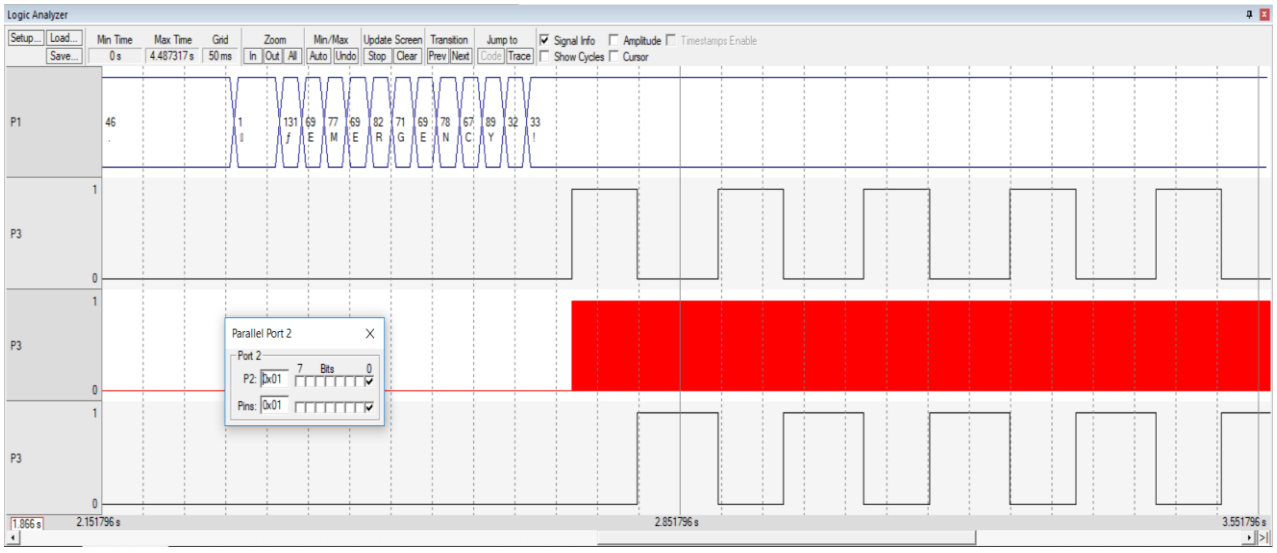
```
                cmd(0x01);
            }
            else if(input==0x08)      //restroom
            {
                cmd(0x01);
                cmd(0x80);
                string("Need to use the");
                cmd(0xc0); // line 2 first position
                string("RESTROOM.");
                buz = 1;//BUZZER

                delay(200);

                cmd(0x01);
            }
            else
            {
                cmd(0x01);
                cmd(0x80);
                string("No Input.");
                delay(100);
                buz = 0;
                cmd(0x01);
            }

        }
}
```

Simulation Output:

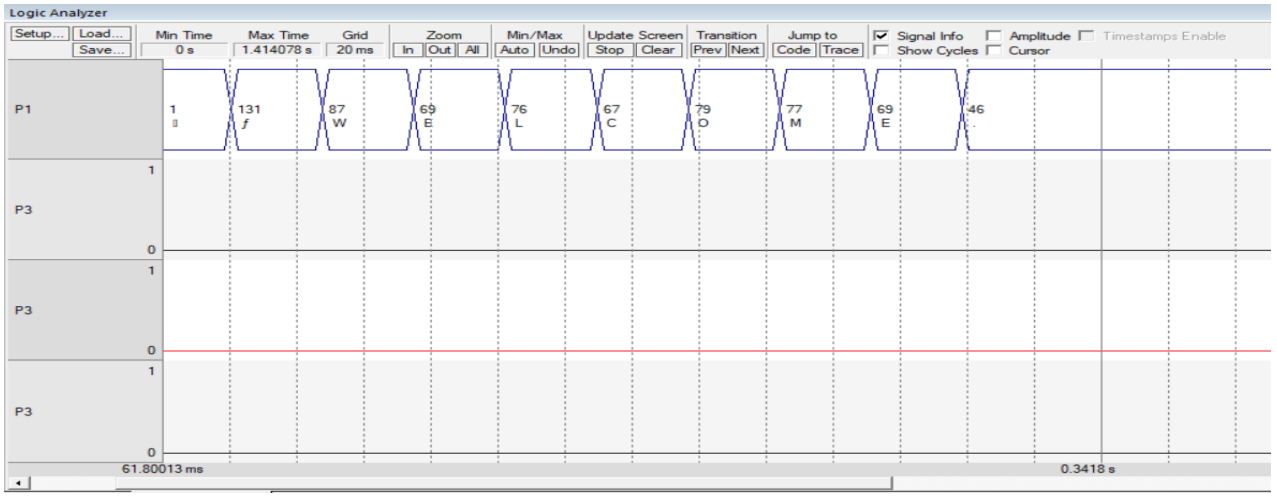
No input is applied



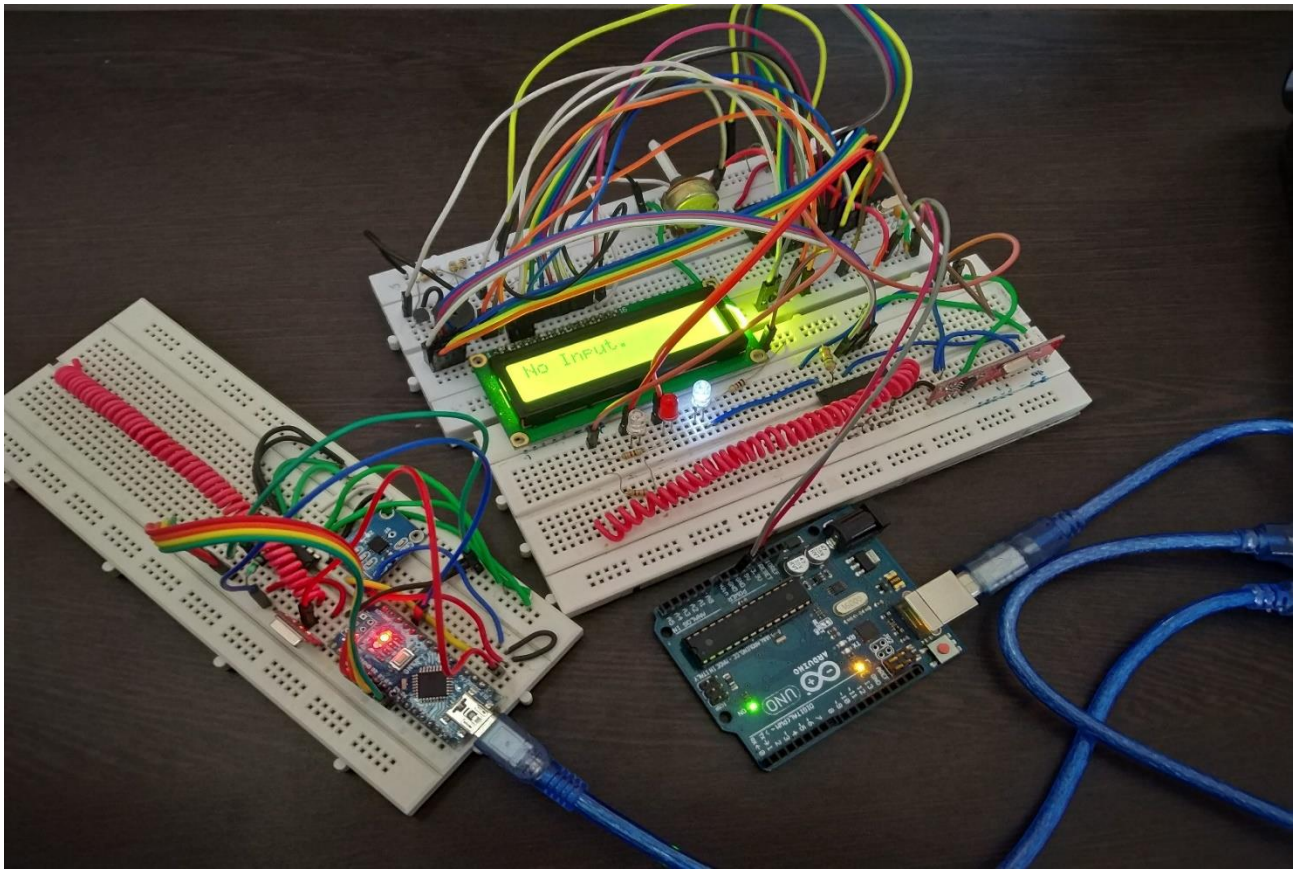
Emergency

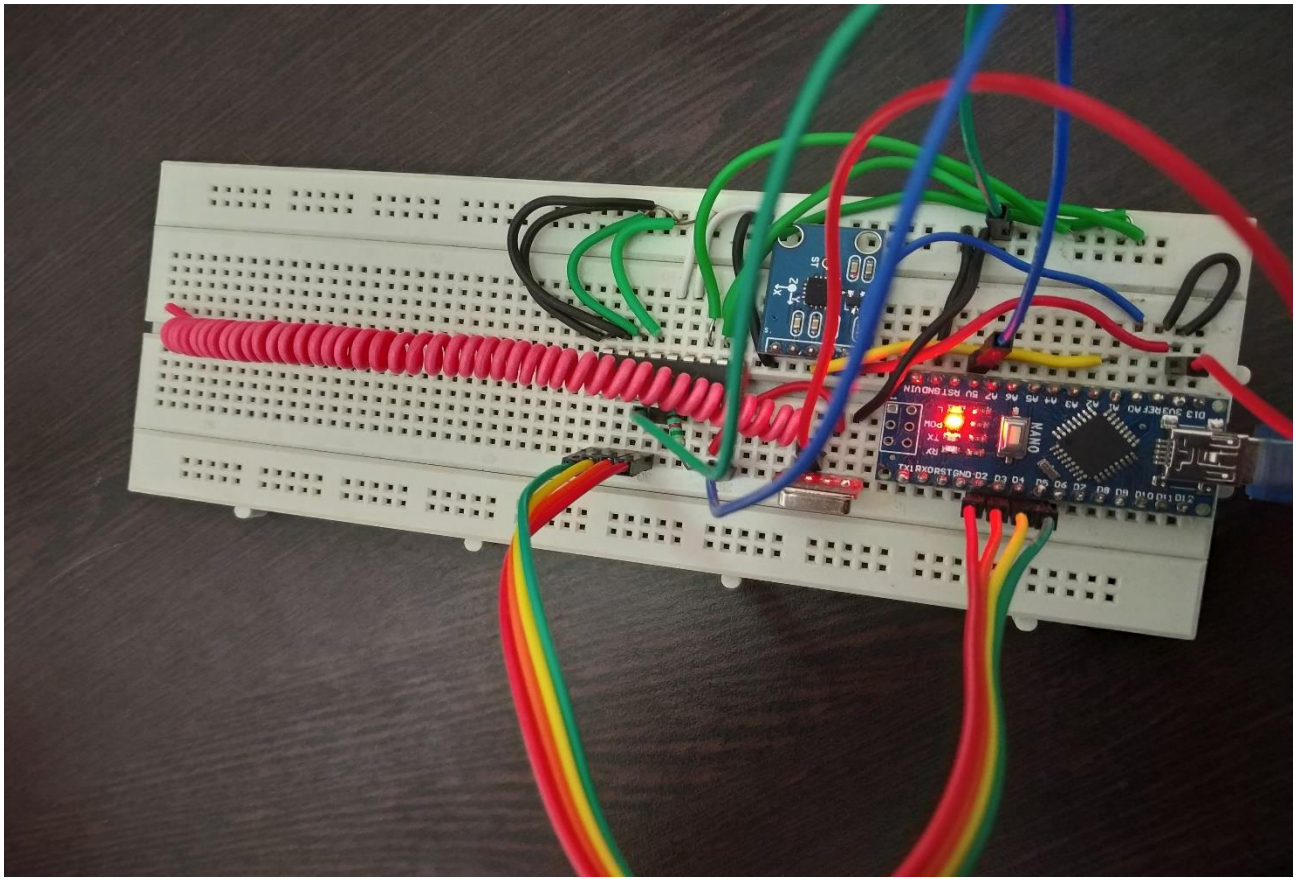


Input is applied



Implementation Output:





Inference:

Thus, we have implemented a prototype of the motion-based message conveyer for patients with difficulties in speech and communication which will assist them is communicating with the nurses and doctors in the time of need.

Concepts Learned:

The working of 8051 microcontroller.

Interfacing of LCD with a microcontroller.

Arduino programming.

Simulating in Proteus and Keil.

Applications:

Our model would be applicable for those who are suffering from speech disorders like:

Apraxia of speech

Cluttering

Developmental verbal dyspraxia

Speech sound disorders

Difficulties faced:

Simulation of the code in Proteus.

Interfacing LCD with 8051.

Making the schematic of the project.

Preparing the code for the project.

Burning the code to 8051.

Timeline: