

SMART WEARABLE SPECTACLES WITH NECK POSTURE MONITORING

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Technology degree in
Biomedical Engineering

By:

DHARANIVASAN.V (39240014)

ARAVINTH.R (39240008)



**DEPARTMENT OF BIOMEDICAL
ENGINEERING SCHOOL OF BIO AND
CHEMICAL ENGINEERING**

**SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY**

**(DEEMED TO BE UNIVERSITY)
Accredited with Grade “A” by NAAC**

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119

APRIL 2022



SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited with Grade “A” by NAAC



JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119
www.sathyabama.ac.in

DEPARTMENT OF BIOMEDICAL ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Interdisciplinary Project Report is the bonafide work of **DHARANIVASAN.V (Reg. No.39240014), ARAVINTH.R (Reg. No. 39240008)** who carried out the interdisciplinary project entitled **“SMART WEARABLE SPECTACLES WITH NECK POSTURE MONITORING”** under our supervision from **JAN 2021 to APRIL 2022.**

Dr. S. KRISHNA KUMAR, M.Sc., Ph.D.,
Guide and Supervisor

Dr. T.SUDHAKAR, M.Sc., Ph.D.,
Head of the Department

Submitted for Viva voce Examination held on 30th April 2022.

Internal Examiner

External Examiner

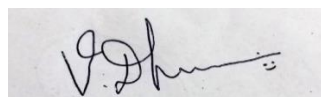
DECLARATION

We, **DHARANIVASAN.V (Reg. No. 39240014), ARAVINTH.R (Reg. No. 39240008)** hereby declare that the interdisciplinary Project Report entitled **“SMART WEARABLE SPECTACLES WITH NECK POSTURE MONITORING”** done by us under the guidance of **Dr. S. Krishna Kumar, Department of Biomedical Engineering** is submitted in partial fulfillment of the requirements for the award of **Bachelor of Technology** degree in **Biomedical Engineering**.

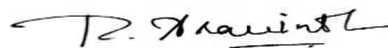
DATE: 30-4-2022

PLACE: Chennai

1.



2.



Scanned by TapScanner

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

We are pleased to acknowledge my sincere thanks to Board of Management of **SATHYABAMA** for their kind encouragement in doing this interdisciplinary project and for completing it successfully. We are grateful to them.

We convey our thanks to **Dr. T. Sudhakar, M.Sc., Ph.D.**, Head of the Department, **Department of Biomedical Engineering** for providing us necessary support and details at the right time during the progressive reviews.

We would like to express our sincere and deep sense of gratitude to our interdisciplinary Project Guide **Dr. S. Krishnakumar, M.Sc., Ph.D., Department of Biomedical Engineering** for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of our project work.

We wish to express our thanks to all **Teaching and Non-teaching** staff members of the **Department of Biomedical Engineering** who were helpful in many ways for the completion of the interdisciplinary project.

ABSTRACT:

In this modern fasttrack world usage of electronic gadgets is like part time of life. Technology is like a coin which has both positive and negative sides. While spending more time in front of the digital screen which leads to major health defects. To create a smart glass that detects eye and neck strain when using a laptop or mobile phone for a prolonged period of time. The alarm sound reminds to the user to change positions if the neck position is unchanged, if the person blinks less than the normal or if the person remains seated for an extended period of time. This will aid in posture correction while working for a prolonged duration. This will avoid many posture-related anomalies and digital eye strain.

TABLE OF CONTENTS

CHAPTER NO	TITLE	PAGE NO
	ABSTRACT	v
	LIST OF FIGURES	viii
1	INTRODUCTION	1
2	LITERATURE SURVEY	2
3	AIM & SCOPE	4
	3.1 AIM	4
	3.2 SCOPE	4
4	MATERIALS & METHOD	5
	4.1 METHODOLOGY	5
	4.2 HARDWARE USED	6
	4.2.1 ESP 32 Microcontroller	6
	4.2.2 Ultrasonic sensor	10
	4.2.3 MPU 6050 Gyroscope	10
	4.2.4 IR Blink sensor	11
	4.2.5 Buzzer	12
	4.2.6 Li ion Rechargeable Battery	12
	4.3 SOFTWARE USED	14
	4.3.1 Arduino IDE	14
5	RESULTS & DISCUSSION	16
	5.1 RESULT	16
	5.2 DISCUSSION	17

6	SUMMARY & CONCLUSION	18
	6.1 SUMMARY	18
	6.2 CONCLUSION	18
	REFERENCES	19
	APPENDIXES	20

LIST OF FIGURES

FIGURE NO	NAME OF THE FIGURE	PAGE NO
4.1	Block Diagram with Explanation	5
4.2	ESP 32 MICROCONTROLLER	6
4.3	Functional block diagram of ESP 32	7
4.4	Ultrasonic sensor	10
4.5	MPU 6050 Gyroscope Sensor	11
4.6	IR Blink sensor	11
4.7	Buzzer	12
4.8	Li ion Rechargeable battery	13
4.9	Arduino IDE	14
5.0	Hardware setup	16

CHAPTER – 1

INTRODUCTION

In this modern Tech world Digital device usage has increased substantially in recent years across all age groups, so that extensive daily use for both social and professional purposes is now normal. Technology is like a coin which has both positive and negative sides. Prolonged use of e-gadgets which leads to major health defects. Taking this into account, we cannot dismiss the fact that our prolonged exposure to screens and incorrect use of modern devices has health Implications. Reduced physical activity, poor sleeping patterns, and digital eye strain are just a few of the negative consequences of smartphone use. We spend most of our time in front of digital screen. Over 90% of people aged between 16 - 35 uses digital devices every day. On average, we spend 8 hours in front of the digital screen. During these pandemic even more usage of mobiles, PC, laptops increased enormously. As we know school, college students and IT workers doing there work in front of the digital screen. Computer vision syndrome (CVS) is a very common phenomenon amongst computer users. A total of 90% of computer users, who spend more than 3 hours a day in front of the computer screen, suffer from CVS. So the only organ being exposed more on it is "EYE ". So in order to rectify these health defects our prototype help in monitoring and produce the output as a buzzer sound.

CHAPTER - 2

LITERATURE SURVEY

Vinaya Kulkarni, Chetana Thombre ,(2018) An Eye Blink and Head Movement Detection for Computer Vision Syndrome - The usage of computers in our day-to-day activities has increased enormously leading to both positive and negative effects in our lives. The negative effects are related to health problems such as Computer Vision Syndrome (CVS) etc. Prolonged use of computers would lead to a significant reduction of spontaneous eye blink rate due to the high visual demand of the screen and concentration on the work. The proposed system develops a prototype using blink as a solution to prevent CVS. Various experiments are done and their algorithms are compared and concluded that the proposed algorithm yields 99.95% accuracy.

Jaehyun Lee, Youngjoon ,(2013)The Wearable Sensor System to Monitor the Head & Neck Posture in Daily Life - The neck pain is fairly common occurrence. Forward head posture and text neck are poor postures which may be related with neck pain but the evidence is not enough. We developed the wearable sensor which can assess the head & neck posture in daily life. Microprocessor, Bluetooth low energy, and 3-axis accelerometer, rechargeable battery and vibrator for reminding are used to implement the wearable sensor.

Simone Benedetto,, (2022) The aim of the present research was to study the effects of the display technology on visual fatigue. To this end, participants performed a longitudinal study in which two last generation e-readers (LCD, E-ink) and paper book were tested in three different prolonged reading sessions separated by - on average - ten days. Results from both objective (Blinks per second) and subjective (Visual Fatigue Scale) measures suggested that reading on the LCD (Kindle Fire HD) triggers higher visual fatigue with respect to both the E-ink (Kindle Paperwhite) and the paper book.

Kampanat WangsanPhit, ,(2019) Students in Virtual Classrooms during the COVID-19 Pandemic: Prevalence and Associated Factors- During the COVID-19 pandemic, computer vision syndrome (CVS) related to online classrooms were unavoidable. This cross-sectional study aimed to explore the prevalence, characteristics and associated factors of CVS. A total of 527 students who were currently studying in a virtual classroom. The most intense symptoms were the feeling of worsening eyesight (15.9%). The factors associated with CVS were female ($p < 0.001$), age ($p = 0.010$), atopic diseases ($p = 0.020$), prior ocular symptoms ($p < 0.001$), astigmatism ($p = 0.033$), distance from display < 20 cm ($p = 0.023$), presence of glare or reflection on screen ($p < 0.001$), low screen brightness ($p = 0.045$), sleep duration ($p = 0.030$), inadequate break time between classes ($p < 0.001$) and increased screen time usage during online study ($p < 0.001$). Recommendations to prevent CVS based on the adjustable factors might reduce the burden of online study.

Sundus Alfaitouri , (2019) he Effect of Posture and Duration of Smartphone Usage on Neck Flexion Angle - The main objective of this study was to measure and examine changes in the neck flexion angle while using a Smartphone according to posture and duration of usage. Photographic analysis procedures were carried out to measure the neck flexion angle. The participants in this study were twenty young adults (half males); they stood against a scaled board and used the same Smartphone for web-browsing for twenty minutes. Photos were taken at zero, five, ten, fifteen and twenty minutes. This procedure was repeated for each participant at three postures (standing, sitting without arms rest, sitting with arms rested on a table). The results revealed statistically significant effects of both posture and duration of usage; neck flexion angle increased significantly over time in all postures and was significantly smaller in the standing posture than in the sitting postures.

CHAPTER - 3

AIM & SCOPE

3.1 AIM:

- To create a smart glass that detects eye and neck strain while using Laptop or mobile phone for an extended period of time.
- To remind the person to change positions if the neck position is unchanged, if the person blinks less than the normal or if the person keeps his/her mobile nearer for an extended period of time.
- To aid in posture correction while working for a prolonged duration which will avoid many posture-related anomalies and improves productivity.
- And to reduce digital eye strain & to avoid computer vision syndrome.

3.2 SCOPE:

- Maintaining the good neck posture and eye vision is essential to live a healthy life.

CHAPTER- 4

MATERIALS & METHODS

4.1 METHODOLOGY

- This prototype helps in monitoring the eye related defects and neck posture correction for the user.
- Here we implement various ultrasonic sensor, Accelerometer sensor , IR blink sensor are incorporated along with ESP- 32 Microcontroller has been used in it.
- Therefore, we introduce microcontroller based smart wearable spectacles in order to detect these eye and neck related health defects.

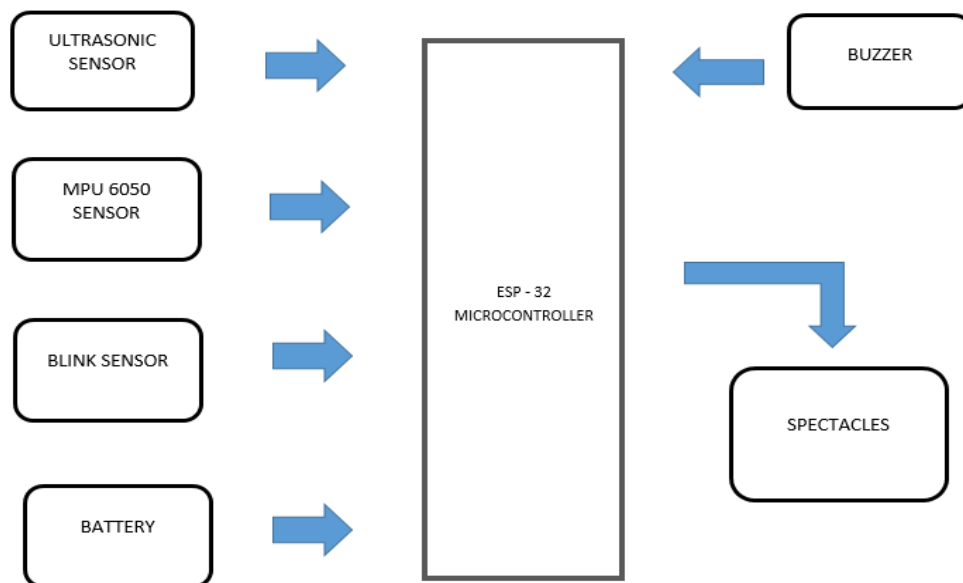


Fig 4.1 BLOCK DIAGRAM WITH EXPLANATION

Our project helps in monitoring the improper eye blink , digital eye strain, improper neck posture using blink sensor , GPU 6050 gyroscope sensor , ultrasonic sensor are inbuilt in the spectacles and if the user cross the threshold

level of the given parameters the buzzer will alert the user to change his /her position. These project aid in neck posture correction and eye related defects.

4.2 HARDWARE USED

4.2.1 ESP 32 MICROCONTROLLER

The ESP32 is a very versatile System On a Chip (SoC) that can be used as a general purpose microcontroller with quite an extensive set of peripherals including WiFi and Bluetooth wireless capabilities. Another module commonly referred to as an ESP32 is what is more appropriately called an ESP32 Development Module. This is basically an ESP32 module mounted on a board with additional support circuitry such as a voltage regulator and a serial to USB IC. It allows direct connection to a desktop PC that can then be used to compile, download, and run programs directly on this module.



Fig 4.2 ESP 32 MICROCONTROLLER

Processors:

Main processor: Tensilica Xtensa 32-bit LX6 microprocessor

Cores: All versions of the ESP32 series are dual-core except for ESP32-S0WD, which is single-core.

Clock frequency: up to 240 MHz

Performance: up to 600 DMIPS

Ultra low power co-processor: allows you to do ADC conversions, computation, and level thresholds while in deep sleep.

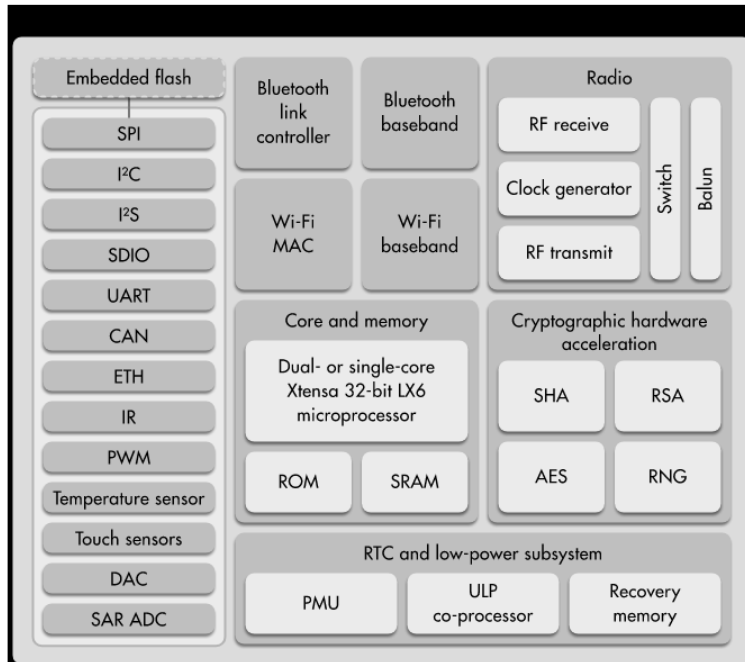


Fig 4.3 FUNCTIONAL BLOCK DIAGRAM OF ESP - 32

Wireless connectivity:

Wi-Fi: 802.11 b/g/n/e/i (802.11n @ 2.4 GHz up to 150 Mbit/s)

Bluetooth: v4.2 BR/EDR and Bluetooth Low Energy (BLE)

Memory:

ROM: 448 KB – For booting and core functions

SRAM: 520 KB – For data and instruction

RTC fast SRAM: 8 KB – For data storage and main CPU during RTC Boot from the deep-sleep mode

RTC slow SRAM: 8 KB – For co-processor accessing during deep-sleep mode

eFuse: 1 KBit – Of which 256 bits are used for the system (MAC address and chip configuration) and the remaining 768 bits are reserved for customer applications, including Flash-Encryption and Chip-ID

Embedded flash:

- 0 MB (ESP32-D0WDQ6, ESP32-D0WD, and ESP32-S0WD chips)
- 2 MB (ESP32-D2WD chip)
- 4 MB (ESP32-PICO-D4 SiP module)

Flash memory is connected internally via IO16, IO17, SD_CMD, SD_CLK, SD_DATA_0 and SD_DATA_1 on ESP32-D2WD and ESP32-PICO-D4.

External flash & SRAM: ESP32 supports up to four 16 MB external QSPI flashes and SRAMs with hardware encryption based on AES to protect developers' programs and data. ESP32 can access the external QSPI flash and SRAM through high-speed caches.

- Up to 16 MB of external flash are memory-mapped onto the CPU code space, supporting 8-bit, 16-bit and 32-bit access. Code execution from flash is supported.
- Up to 8 MB of external flash/SRAM memory are mapped onto the CPU data space, supporting 8-bit, 16-bit and 32-bit access. Data-read is supported on the flash and SRAM. Data-write is supported on the SRAM.

Peripheral input/output: The ESP32 offers a rich peripheral interface with DMA that includes:

- Capacitive touch
- ADCs (analog-to-digital converter)
- DACs (digital-to-analog converter)
- I²C (Inter-Integrated Circuit)
- UART (universal asynchronous receiver/transmitter)
- CAN 2.0 (Controller Area Network)
- SPI (Serial Peripheral Interface)

- I²S (Integrated Inter-IC Sound)
- RMII (Reduced Media-Independent Interface)
- PWM (pulse width modulation), and more.

Security:

- IEEE 802.11 standard security features all supported, including WPA, WPA/WPA2 and WAPI
- Secure boot
- Flash encryption
- 1024-bit OTP, up to 768-bit for customers
- Cryptographic hardware acceleration: AES, SHA-2, RSA, elliptic curve cryptography (ECC), random number generator (RNG).

4.2.2 ULTRASONIC SENSOR

Ultrasonic sensor is an electronic device that measures the distance of a target object by emitting ultrasonic sound waves, and converts the reflected sound into an electrical signal. Ultrasonic waves travel faster than the speed of audible sound (i.e. the sound that humans can hear). Ultrasonic sensors have two main components: the transmitter (which emits the sound using piezoelectric crystals) and the receiver (which encounters the sound after it has travelled to and from the target).

In order to calculate the distance between the sensor and the object, the sensor measures the time it takes between the emission of the sound by the transmitter to its contact with the receiver. The formula for this calculation is $D = \frac{1}{2} T \times C$ (where D is the distance, T is the time, and C is the speed of sound ~ 343 meters/second).

4.2.4 IR BLINK SENSOR

This Eye Blink sensor is IR based . The Variation Across the eye will vary as per eye blink . If the eye is closed means the output is high otherwise output is low. This to know the eye is closing or opening position. This output is give to logic circuit to indicate the alarm.



Fig 4.6 *IR BLINK SENSOR*

4.2.5 BUZZER

An audio signaling device like a beeper or buzzer may be electromechanical or piezoelectric or mechanical type. The main function of this is to convert the signal from audio to sound. Generally, it is powered through DC voltage and used in timers, alarm devices, printers, alarms, computers, etc. Based on the various designs, it can generate different sounds like alarm, music, bell & siren.



Fig 4.7 BUZZER

4.2.6 RECHARGEABLE BATTERY

A lithium-ion battery or Li-ion battery is a type of rechargeable battery composed of cells in which lithium ions move from the negative electrode through an electrolyte to the positive electrode during discharge and back when charging. Li-ion cells use an intercalated lithium compound as the material at the positive electrode and typically graphite at the negative electrode. Li-ion batteries have a high energy density, no memory effect (other than LFP cells) and low self-discharge. Cells can be manufactured to either prioritize energy or power density. They can however be a safety hazard since they contain flammable electrolytes and if damaged or incorrectly charged can lead to explosions and fires.



Fig 4.8 Li ion rechargeable battery

4.3 SOFTWARE USED

4.3.1 ARDUINO IDE

Arduino Integrated Development Environment - or Arduino Software (IDE) - contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions and a series of menus. It connects to the Arduino hardware to upload programs and communicate with them. The IDE application is suitable for different operating systems such as Windows, Mac OS X, and Linux. It supports the programming languages C and C++. Here, IDE stands for Integrated Development Environment.

The program or code written in the Arduino IDE is often called as sketching. We need to connect the Genuino and Arduino board with the IDE to upload the sketch written in the Arduino IDE software. The sketch is saved with the extension '.ino.'

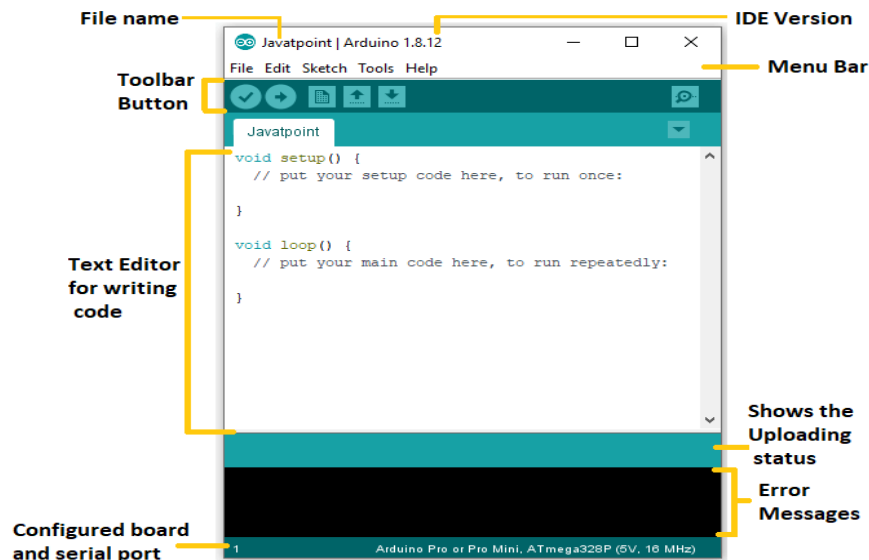


Fig 4.9 ARDUINO IDE

- Inexpensive - Arduino boards are relatively inexpensive compared to other microcontroller platforms. The least expensive version of the Arduino module can be assembled by hand, and even the pre-assembled Arduino modules cost less than \$50
- Cross-platform - The Arduino Software (IDE) runs on Windows, Macintosh OSX, and Linux operating systems. Most microcontroller systems are limited to Windows.
- Simple, clear programming environment - The Arduino Software (IDE) is easy-to-use for beginners, yet flexible enough for advanced users to take advantage of as well. For teachers, it's conveniently based on the Processing programming environment, so students learning to program in that environment will be familiar with how the Arduino IDE works.
- Open source and extensible software - The Arduino software is published as open source tools, available for extension by experienced programmers. The language can be expanded through C++ libraries, and people wanting to understand the technical details can make the leap from Arduino to the AVR C programming language on which it's based. Similarly, you can add AVR-C code directly into your Arduino programs if you want to.
- Open source and extensible hardware - The plans of the Arduino boards are published under a Creative Commons license, so experienced circuit designers can make their own version of the module, extending it and improving it. Even relatively inexperienced users can build the breadboard version of the module in order to understand how it works and save money.

CHAPTER-5

RESULTS AND DISCUSSION

5.1 RESULTS

In these project, we use various sensors to monitor the eye related defects such as, improper eye blink, digital eye strain and improper neck posture. We use ESP 32 microcontroller to implement the working of our prototype. The measured parameters are monitored and if exceed the threshold level the buzzer reminds the user to change the position.

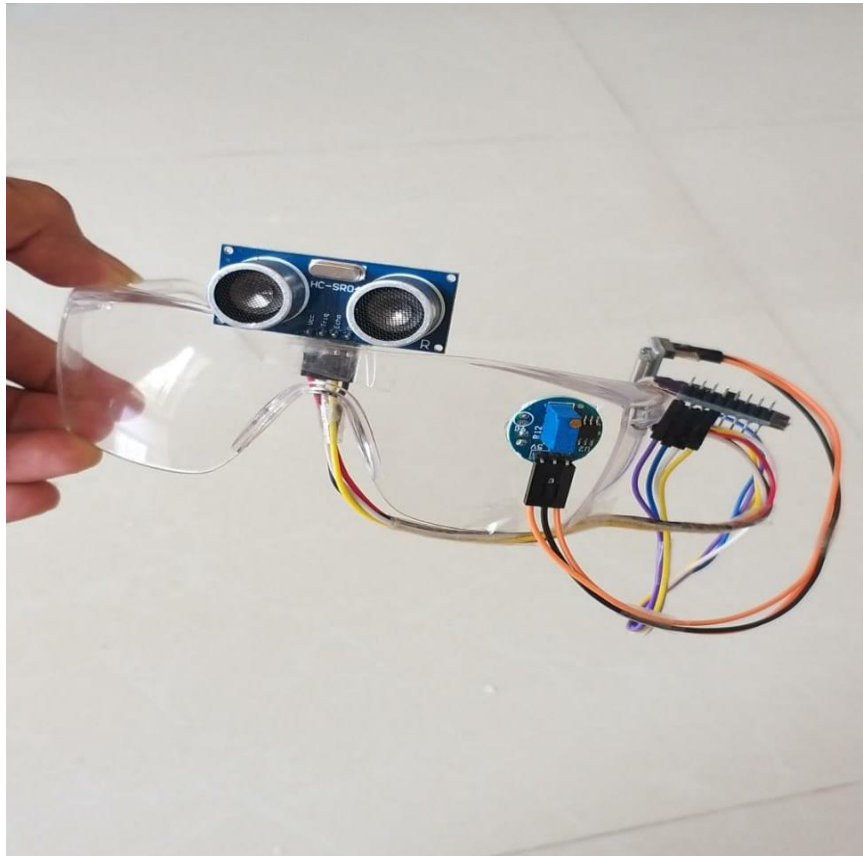


Fig 5.0 Hardware setup

5.2 DISCUSSION

This section provides about a review of applications of various sensors and systems used in this project. After successful implementation and testing of coding the following output was achieved. The above figure shows the prototype of the spectacles. Further it can be modified with some additional features such as measuring EOG of the eye movement and sensing the body temperature and its buzzer can be modified as notification in mobiles.

CHAPTER – 6

SUMMARY AND CONCLUSION

6.1SUMMARY

- As we take eye as concern we brought up these idea smart wearable health spectacles.
- We are using various sensors in order to monitor the eye related health defects and to correct neck posture.
- The output is received as buzzer sound if any abnormalities detected.
- So the user changes the position to get rid from these abnormalities.
- It is economical affordable and safe for the user.
- Overall process is done automatically.

6.2CONCLUSION

Our prototype aid to prevent eye related health defects and used in neck correction. It will be flexible and cheaper among the consumer and with good accuracy results shown. In these current generation surely it will be useful for the IT employees, online class students, playing mobile games. Furthermore, these project can be implemented as a real time user application in desktop, laptops, mobiles.

REFERENCES

1. Jaehyun and Younghoon Kim The Wearable Sensor System to Monitor the Head & Neck Posture in Daily Life Interdisciplinary Program of Medical & Biological Engineering, Graduate School, University of Ulsan 680-749, Korea 1 College of Engineering, School of Electrical Engineering, Biomedical Engineering, University of Ulsan (Manuscript received 06 June 2016; revised 01 July 2016; accepted 02 July 2. Journal of Biomedical Engineering Research 37: 112-118 (2016).
2. Kampanat Wangsan Self-Reported Computer Vision Syndrome among Thai University Students in Virtual Classrooms during the COVID-19 Pandemic: Prevalence and Associated Factors march 2022 International Journal of Environmental Research and Public Health. See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/359507294>.
3. Sundus Alfaitouri and Ahamed Altaboli The Effect of Posture and Duration of Smartphone Usage on Neck Flexion Angle Industrial and Manufacturing Systems Engineering Department University of Benghazi, Benghazi, Libya. Proceedings of the Human Factors and Ergonomics Society 2019 Annual Meeting.
4. Vinaya Kulkarni An Eye Blink and Head Movement Detection for Computer Vision Syndrome , February 2018, International Journal of Science and Engineering Applications Volume 7–Issue 02, 20-22, 2018, ISSN:-2319–7560.

APPENDIXES

// Basic demo for accelerometer readings from Adafruit MPU6050

```
#include <Adafruit_MPU6050.h>
#include <Adafruit_Sensor.h>
#include <Wire.h>

Adafruit_MPU6050 mpu;
long int time0;
int count = 0;
int xdata = 0;
int led=2;
void setup(void) {
  Serial.begin(115200);
  pinMode(led,OUTPUT);
  while (!Serial)
    delay(10); // will pause Zero, Leonardo, etc until serial console opens

  //Serial.println("Adafruit MPU6050 test!");

  // Try to initialize!
  if (!mpu.begin()) {
    //Serial.println("Failed to find MPU6050 chip");
    while (1) {
      delay(10);
    }
  }
  //Serial.println("MPU6050 Found!");

  mpu.setAccelerometerRange(MPU6050_RANGE_8_G);
  //Serial.print("Accelerometer range set to: ");
  switch (mpu.getAccelerometerRange()) {
    case MPU6050_RANGE_2_G:
```

```

    //Serial.println("+2G");
    break;
case MPU6050_RANGE_4_G:
    //Serial.println("+4G");
    break;
case MPU6050_RANGE_8_G:
    //Serial.println("+8G");
    break;
case MPU6050_RANGE_16_G:
    //Serial.println("+16G");
    break;
}
mpu.setGyroRange(MPU6050_RANGE_500_DEG);
//Serial.print("Gyro range set to: ");
switch (mpu.getGyroRange()) {
case MPU6050_RANGE_250_DEG:
    //Serial.println("+ 250 deg/s");
    break;
case MPU6050_RANGE_500_DEG:
    //Serial.println("+ 500 deg/s");
    break;
case MPU6050_RANGE_1000_DEG:
    //Serial.println("+ 1000 deg/s");
    break;
case MPU6050_RANGE_2000_DEG:
    //Serial.println("+ 2000 deg/s");
    break;
}

mpu.setFilterBandwidth(MPU6050_BAND_21_HZ);
//Serial.print("Filter bandwidth set to: ");
switch (mpu.getFilterBandwidth()) {
case MPU6050_BAND_260_HZ:
    //Serial.println("260 Hz");

```

```

        break;
    case MPU6050_BAND_184_HZ:
        //Serial.println("184 Hz");
        break;
    case MPU6050_BAND_94_HZ:
        //Serial.println("94 Hz");
        break;
    case MPU6050_BAND_44_HZ:
        //Serial.println("44 Hz");
        break;
    case MPU6050_BAND_21_HZ:
        //Serial.println("21 Hz");
        break;
    case MPU6050_BAND_10_HZ:
        //Serial.println("10 Hz");
        break;
    case MPU6050_BAND_5_HZ:
        //Serial.println("5 Hz");
        break;
}

//Serial.println("");
delay(100);
}

void loop() {

    /* Get new sensor events with the readings */
    sensors_event_t a, g, temp;
    mpu.getEvent(&a, &g, &temp);

    /* Print out the values */
    //Serial.print("Acceleration X: ");
    /* Serial.print(a.acceleration.x);

```

```

Serial.print(",");
Serial.print(a.acceleration.y);
Serial.print(",");
Serial.println(a.acceleration.z);*/

// long int time0;
// int count = 0;
xdata = (a.acceleration.x) + 10;
Serial.println(count);
if(xdata <= 9)
{
    count++;
}
else
{
    count=0;
    digitalWrite(led,LOW);
}
if(count > 1000)
{
    digitalWrite(led,HIGH);
    Serial.println("tAKEN LONG tIME");
}
//Serial.println(" m/s^2");

//Serial.print("Rotation X: ");
//Serial.print(g.gyro.x);
//Serial.print(", Y: ");
//Serial.print(g.gyro.y);
//Serial.print(", Z: ");
//Serial.print(g.gyro.z);
//Serial.println(" rad/s");

//Serial.print("Temperature: ");

```

```

//Serial.print(temp.temperature);
//Serial.println(" degC");

//Serial.println("");
delay(5);
}

```

2. ultrasonic code

```

int trigPin = 5;
int echoPin = 18;
int duration;
int distance;
int count = 0;
int xdata = 0;
int led=2;
void setup() {
  Serial.begin(115200);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(led,OUTPUT);
  while (!Serial)
    delay(10);
}
void loop() {
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  distance= duration*0.034/2;
  //Serial.println(distance);
}

```

```
// long int time0;  
// int count = 0;  
xdata = distance;  
Serial.println(count);  
if(xdata <= 30)
```

3. IR BLINK SENSOR CODE

```
int ir=13;  
int led=14;  
int count =0;  
int a=0;  
int b= 0;  
void setup()  
{  
  pinMode(ir,INPUT);  
  pinMode(led,OUTPUT);  
  Serial.begin(115200);  
}  
void loop()  
{  
  int irvalue=digitalRead(ir);  
  if(irvalue==LOW)  
  {  
    Serial.println("LOW");  
  
  }  
  else if (irvalue==HIGH)  
  {  
    Serial.println("High");  
  
    a++;  
  }  
}
```

```
b++;  
if (b==15)  
{  
  if(a<4)  
  {  
    Serial.println("not enough");  
    digitalWrite(led,HIGH);  
  }  
  else if (a>=4)  
  {  
    Serial.println("enough");  
    digitalWrite(led,LOW);  
  }  
  b=0;  
  a=0;  
}  
  
delay(1000);  
}
```