**COLLEGE OF ENGINEERING**

**DEPARTMENT OF COMPUTER ENGINEERING**

**BIOMEDICALENGINEERING 3**

**COE 381 (MICROPROCESSORS)**

**GROUPS 9 AND 10**

**GROUP MEMBERS**

Emily Korkor Tetteh- 1807822

Gyan Festus - 1804722

Aboagye-Atta Nana Kwadwo Oduro - 1799522

Moses Asante - 1802022

Aseda Boatemaa Tweneboah-Koduah - 1808622

Agyei David Beckham - 1800422

Koduah Roselyn Akua Ohenewaah - 1805222

Opoku Asare Millicent - 1806522

Michelle Veronique Quainoo- 1807122

Josephkerry Edinam Kwadzokpo - 1805422

TABLE OF CONTENT

Catalog

[DESIGN OF A POSTURE CORRECTION ALERT SYSTEM 3](#_Toc5177)

[PROBLEM STATEMENT 3](#_Toc20640)

ABSTRACT..............................................................................................3

[OBJECTIVES 4](#_Toc3480)

[METHODOLOGY 4](#_Toc2329)

[Wiring the MPU6050 sensor to Arduino UNO: 4](#_Toc9359)

[Wiring the Buzzer to Arduino UNO 5](#_Toc30896)

[Wiring the LED to Arduino UNO: 5](#_Toc22384)

[Code Logic Explanation: 6](#_Toc32181)

[1. Libraries Used: 6](#_Toc1931)

[2. Setup: 7](#_Toc26503)

[3. Main Loop: 7](#_Toc13515)

[4. Output: 7](#_Toc5193)

[Libraries and Why They Are Necessary: 8](#_Toc13086)

[Simulation & Testing: 8](#_Toc9506)

[Simulation results: 8](#_Toc4155)

[CONCLUSION 9](#_Toc17775)

**DESIGN OF A POSTURE CORRECTION ALERT SYSTEM**

**ABSTRACT**

Millions of people across the globe are suffering from back pain. The main reason behind is bad posture of back. We always tend to ignore these little symptoms of back pain, which would lead to serious deformities in the future. In the present, computers,laptops, electronic gadgets have been an important routine. So as prevention, we require a bio-electronic device which could give the inference on his posture and help maintain a better posture, hence maintaining a good spine health. The main focus of this paper is about the construction, working, testing of the posture alert system device developed using Arduino Uno and MPU6050 sensor. This device being developed is low cost, reliable and user friendly for most ages.

**PROBLEM DEFINITION**

Poor posture is a common problem that has great consequences for health, emotional well-being, and productivity. According to research, 80% of people will develop back discomfort at some point in their life, usually because of persistent bad posture, especially among those who spend long hours at desks or use electronic devices excessively. Chronic slouching causes musculoskeletal issues, decreased lung capacity, weariness and even lower self-esteem. Current solutions such as wearable posture correctors, smartphone apps, and ergonomic furniture have notable limitations. Wearables frequently suffer for inconsistent warnings, battery drain, and expensive cost, whilst apps struggle to maintain user attention. Ergonomic seats offer passive support but don’t actively correct posture. These shortcomings point out the need for low-cost automated, real -time posture correction systems that actively analyzes posture and provides users with instantaneous feedback.

**OBJECTIVES**

* Design a real time posture monitoring system using Arduino.
* Implement an MPU6050 sensor to detect incorrect posture by measuring the body’s tilt angle.
* Develop an alert mechanism (Buzzer + LED) to notify users of poor posture.
* Include an OLED display for easy interpretation of flexion data.
* Ensure the system is low-cost, energy-efficient, and easy to use.

**METHODOLOGY**

**Posture Correction Alert System Overview**

How it works?

The system continuously monitors the user’s posture using an MPU6050 motion sensor (accelerometer and gyroscope). The Arduino Uno is the microcontroller that processes sensor data and determines if the user’s posture deviates from the predefined threshold (within the range of 5 to 15 degrees). If poor posture is detected that is, angles outside the ran, the buzzer and LED are activated to remind the user to sit upright (OLED display displays “BAD POSTURE!”). Once the user corrects his/her posture, the alerts stop and monitoring continues.

**Wiring Explanation for the Posture Correction Alert System:**

The system consists of four main components:

* MPU6050 Accelerometer – The accelerometer measure the rate of change in velocity and can be used for the detection of orientation, motion and vibration.
* Buzzer and LED
* OLED Display
* Arduino UNO
* Wiring the MPU6050 sensor and OLED display (SSD1306).

to Arduino UNO:

The MPU6050 communicates with the Arduino UNO using I2C protocol, which requires is required when these pins are used:

* SDA (Serial Data Line)
* SCL (Serial Clock Line)

|  |  |  |
| --- | --- | --- |
| MPU6050 Pin/OLED Pin | Arduino UNO Pin | Function |
| VCC | 5V | Power supply |
| GND | GND | Ground |
| SDA | A4 | Transfers data |
| SCL | A5 | Synchronizes communication |

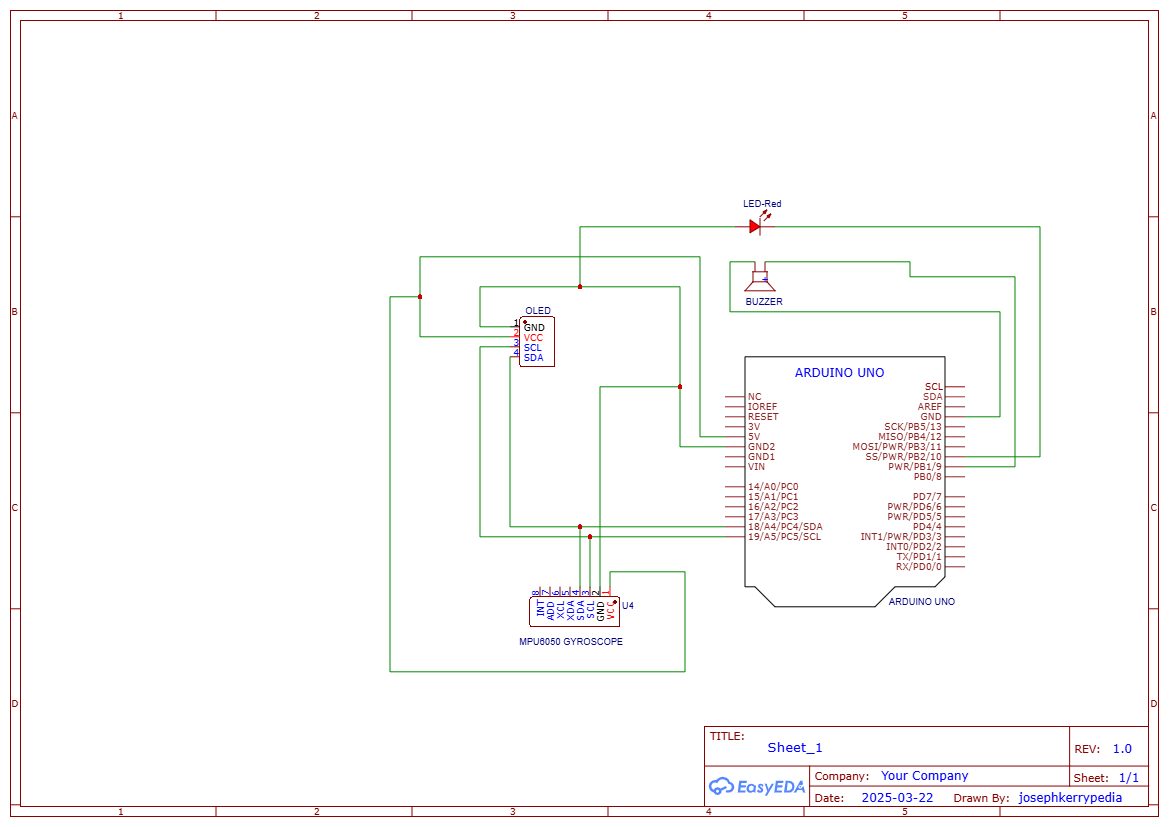
[The Arduino UNO (master) sends clock signals through the SCL. MPU6050 which is the slave, which then sends posture data to the Arduino via SDA.]

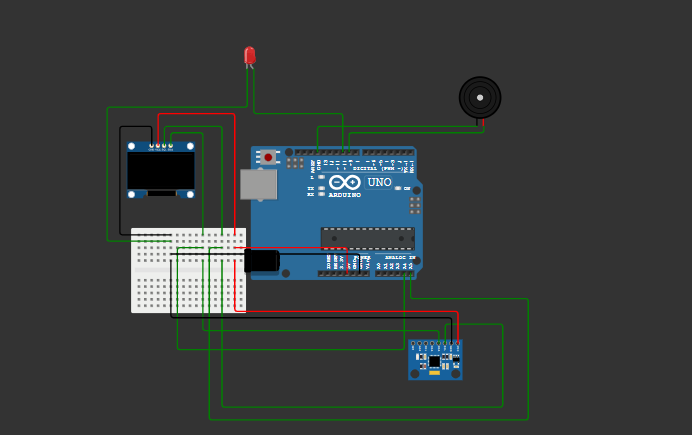
Wiring the Buzzer to Arduino UNO

The buzzer is connected to Digital Pin 9 for control (the positive leg is connected to D9 and negative leg to GND).

Wiring the LED to Arduino UNO:

The anode of the LED is connected to Digital Pin 10 and the cathode to GND.

**Schematics**



**Code Logic Explanation:**

1. **Libraries Used:**

* Wire.h: This is the I2C library, which allows communication between the Arduino and other I2C devices like the MPU6050 sensor and the OLED display.
* MPU6050.h: This library is specifically for communicating with the MPU6050 accelerometer and gyroscope sensor. It helps in fetching motion data (e.g., acceleration and rotation) from the sensor.
* Adafruit\_GFX.h: This is a graphics library needed for drawing text and shapes on the OLED screen.
* Adafruit\_SSD1306.h: This is the library for controlling SSD1306-based OLED displays. It helps in displaying text and graphics on the screen.

2. **Setup:**

* The MPU6050 sensor is initialized using the mpu.initialize() function.
* I2C communication is started using Wire.begin().
* The OLED display is initialized using display.begin(). If it fails to initialize, the program halts.
* Pin modes for the buzzer (Pin 9) and LED (Pin 10) are set to output so that they can be used for feedback.
* A connection test for the MPU6050 sensor is performed. If the sensor is not connected properly, the program stops.

3. **Main Loop:**

* The program continuously reads the accelerometer data using mpu.getMotion6(), which fetches the X, Y, and Z plane acceleration values.
* The tilt angle is calculated using the accelerometer data (ay and az values) with the formula angleX = atan2(ay, az) \* 180 / PI. This gives the angle of the tilt in degrees.
* The tilt angle is displayed on the OLED screen using display.print(). The angle is printed along with the text “Tilt:”.
* The program checks whether the tilt angle falls within a specific range (5 to 15 degrees). If the tilt angle is outside this range (i.e., the posture is "bad"), it turns on the buzzer and LED and displays "Bad Posture!" on the OLED screen. If the angle is within the range, the posture is considered "good", and the buzzer and LED are turned off, displaying "Good Posture".
* A delay of 1 second (delay (1000)) is added before the next reading.

4. Output:

* The system provides feedback via the buzzer (sound), LED (visual), and OLED screen (text).
* The buzzer emits a tone when the posture is bad and stops when the posture is good.
* The LED lights up when the posture is bad and turns off when the posture is good.
* The OLED display shows the tilt angle and a message indicating the user's posture.

**Libraries and Why They Are Necessary:**

* Wire.h: Essential for enabling I2C communication between the Arduino and external components like the MPU6050 sensor and OLED display.
* MPU6050.h: Required to communicate with the MPU6050 sensor and retrieve motion data (acceleration and angular velocity) needed to calculate the tilt angle.
* Adafruit\_GFX.h: A core library needed for graphical operations like displaying text or images on the OLED screen.
* Adafruit\_SSD1306.h: This library is specifically designed to control the SSD1306 OLED display, allowing you to render text, shapes, and images, and to manage the display’s settings.

**Simulation & Testing:**

The simulation setup for the Posture Correction Alert System was conducted using Wokwi to test functionality before hardware implementation. The circuit included an Arduino Uno, an MPU6050 accelerometer, an OLED display (SSD1306), a buzzer, and an LED. The MPU6050 and OLED were connected via I2C (SDA to A4, SCL to A5), while the buzzer and LED were connected to digital pins 9 and 10, respectively. The system continuously read tilt data, calculated the tilt angle, and displayed it on the OLED screen. If the tilt angle exceeded 15° or was below 5°, the buzzer and LED were activated, and a "Bad Posture!" warning was displayed. If within range, the system displayed "Good Posture" with no alerts. The simulation validated the system’s ability to detect posture changes and provide real-time feedback.

**Simulation results:**

1. Good Posture (Tilt Angle: 5° - 15°)

* The OLED displayed: "Tilt: 10.35°" and "Good Posture".
* The LED and Buzzer were OFF, indicating no alert was needed.

2. Slightly Bad Posture (Tilt Angle: >15° or <5°)

* The OLED displayed: "Tilt: 28.43°" and "Bad Posture!"
* The LED turned ON and OFF continuously, and the Buzzer emitted a buzz to warn the user.

4. Real-time Response

* The tilt angle is updated after every second.
* The system effectively detected posture changes, providing instant feedback

**CONCLUSION**

This project successfully developed a posture correction system using an MPU6050 sensor and an Arduino. The system provides real-time feedback to users by detecting tilt angles and triggering alerts for poor posture. The system consists of an OLED display, buzzer, and LED for immediate and clear feedback. Future enhancements could include

* Mobile integration
* Advanced filtering algorithms to reduce the error margin
* Multi axial monitoring, that is the inclusion of the y axis for more precise monitoring.
* Multi sensor monitoring which involves including gyroscope data to get a clearer and more precise information on the user’s spatial position).

Overall, this project presents an effective, low-cost solution for posture monitoring and correction which can be used in rehabilitation or simple domestic use. Some more work on the code could very allow it to not only detect sitting posture but also standing posture which will exponentially increase the scope of its application.

**10. References**

* Arduino Official Documentation
* MPU6050 Sensor Datasheet
* Adafruit OLED Display Libraries Documentation
* Cornell University Ergonomics Web. (n.d.). Ergonomics of sitting. Ergo.human.cornell.edu. https://ergo.human.cornell.edu/DEA3250Flipbook/DEA3250notes/sitting.html
* Cambridge University Hospitals . (2024, July 10). *Sitting Ergonomics*. NHS choices. https://www.cuh.nhs.uk/patient-information/seating-and-ergonomics/
* Haller, M., Richter, C., Brandl, P., Gross, S., Schossleitner, G., Schrempf, A., ... & Inami, M. (2011). Finding the right way for interrupting people improving their sitting posture. In *Human-Computer Interaction–INTERACT 2011: 13th IFIP TC 13 International Conference, Lisbon, Portugal, September 5-9, 2011, Proceedings, Part II 13* (pp. 1-17). Springer Berlin Heidelberg.