Applied Science and Innovative Research
ISSN 2474-4972 (Print) ISSN 2474-4980 (Online)
Vol. 10, No. 4, 2026
www.scholink.org/ojs/index.php/asir

Original Paper

Design and Development of a Wearable Posture Monitoring Prototype Using a Single IMU

Liwei Liu[1]

[1] Northern Arizona University, Flagstaff, AZ, 86001, United States

Received: October 9, 2026 Accepted: November 3, 2026 Online Published: November 26, 2026

doi:10.22158/asir.v10n4p37 URL: http://dx.doi.org/10.22158/asir.v10n4p37

Abstract

Long term use of computers can cause changes in head and neck posture. The project developed a wearable device for monitoring continuous sensor tilt deviation and providing posture reminders. The system uses Arduino UNO R4 and MPU6050, and is connected near C7 using tape and 3D printed casing. Acceleration measurement uses a low-pass filter to estimate the sensor tilt angle. Establish personal reference angles for three calibration records. The program classifies absolute deviations using thresholds of 5°, 10°, and 15°. A deviation of at least 5° for 10 seconds will result in a serial warning and LED feedback. A supporting iOS application has also been developed. For the 5 reference angles, the average absolute error generated by the simulated average estimation is 0.198°, and the maximum absolute error is 0.317°. The sensor measures the local inclination, which is affected by the movement of the torso and changes in attachments. The absolute deviation rule does not distinguish between forward and backward movements. Further testing is needed to evaluate measurement errors, calibration repeatability, warning time, and hardware to application communication.

Keywords

Wearable posture monitoring, MPU6050, Arduino UNO R4, Sensor inclination, Personal calibration, Posture feedback

1. Introduction

1.1 Background

In contemporary society, computers have become a commonly used tool for people's daily learning and life. We usually need to maintain a sitting posture for a long time while sitting in front of a computer desk reading screens and handling tasks. However, our sitting posture often has some issues, and the position of our head and neck can be affected by the work environment. Straker et al. conducted an experiment on 36 young adults and found that a lower display position increased the flexion angle of

the head and neck [1]. This indicates that the posture when using a computer is not only related to personal habits, but also to various factors such as monitor height and desktop layout.

This project focuses on forward head posture (FHP), defined as anterior displacement of the head relative to the torso. A systematic review reported an association between FHP and neck pain in adults, although this does not establish a causal relationship [2]. In this prototype, the MPU6050 estimates the tilt of its mounting location. Angular deviation describes the magnitude of the change in this sensor tilt from a personal reference posture. The relationship between this signal and FHP must be evaluated, because movements such as torso inclination can also change the sensor reading.

To describe the degree of forward tilt of the head, the craniovertebral angle (CVA) was used in the study. This angle is usually composed of the seventh cervical spinous process, C7, the line connecting it to the tragus, and the horizontal line passing through C7. A smaller CVA usually indicates a greater degree of forward movement of the head. Lau et al. studied the reliability and effectiveness of electronic head measurement instruments for measuring CVA, providing a basis for quantitative evaluation of head posture [3]. Compared to simple observation, using clear data indicators can help research clearer measurements.

1.2 History

The project uses anatomical landmarks and angle measurements to evaluate head posture. For example, Lau et al.'s study used an electronic instrument to measure craniovertebral angle [3]. These measurements provide a reference for posture assessment. But in daily life, it is necessary to adapt to a more practical way to collect and process data.

Different sensors provide different measurement methods. IMU measures acceleration and angular velocity, flexible strain sensors respond to deformation, and pressure sensors measure contact force. These signals describe different aspects of motion, so the selection of sensors should depend on the actual usage situation [4].

Different feedback during work was also investigated. Ailneni et al. studied 19 participants performing typing tasks with and without wearable posture sensors while sitting and standing. Participants can adjust their workstations. This study reported the lower neck flexion angle and gravitational moment of the sensor [5]. These findings are related to the workstation tasks being tested and cannot determine long-term therapeutic effects.

Park and Jung studied a single IMU system with personal reference pose and visual feedback [6]. Workplace posture systems also use mobile interfaces to provide feedback [7]. These studies provide a foundation for the placement of sensors, personal calibration, and verification of reminder times in this project.

1.3 Motivation

The motivation for this project is to research a simple wearable method to monitor posture changes during computer use. A single IMU is a very practical starting point, but its output must be interpreted based on the installation position and the measured motion.

Previous studies on cervical posture monitoring have provided a basis for detecting sensor positions near C7 [6]. When a fixed sensor rotates, the gravity component measured along its axis changes, allowing for the estimation of local tilt. Whether this tilt can truly reflect the condition of the head still needs to be tested.

The difference in repeated placement and individual posture can also change the initial angle of the sensor. Therefore, the project needs to refer to individual baselines. Also pay attention to the duration of the bad behavior in order to delay reminders during brief deviations. These design choices lead to three research questions:

1. Can sensors track controlled changes in tilt, and what is the relationship between their output and the target posture?
2. How repeatable is the individual baseline after fixed position and sensor modification?
3. Does the reminder respond to ongoing deviations and recoveries according to predetermined time rules?

1.4 Objective and Scope

The objective of this study is to develop a single IMU prototype. Evaluate whether it can correctly monitor posture changes related to FHP when using a computer while sitting. This project utilizes an Arduino UNO R4, an MPU6050 sensor, a casing, and accompanying iOS applications.

The specific objectives of the project are:

1. Compare sensing methods and select appropriate configurations for the prototype.
2. Implement acceleration-based tilt estimation, personal baseline calibration, angle deviation classification, and duration based reminders.
3. Evaluate inclination measurement, calibration repeatability, and warning timing through control testing.
4. Record the complete iOS application and evaluate hardware data transfer and synchronization feedback.

The project developed embedded programs, shell design, and accompanying iOS software. The current algorithm uses the absolute angle difference from the individual baseline, so it responds to deviations in any direction. The focus of the research is on engineering implementation and experimental evaluation. Clinical diagnosis and long-term treatment outcomes are not within its scope.

1.5 Overview

This article is divided into five parts. The first part introduces the research background, motivation, objectives, and scope. Section 2 reviewed sensing methods, attitude indicators, calibration methods, and feedback systems. Section 3 introduces the prototype, signal processing, and experimental process. Section 4 presents the measurement, calibration, and feedback results, as well as the completed iOS software, and discusses their limitations. Section 5 summarizes the survey results and identifies future work.

2. Literature Review

The posture monitoring system usually includes several steps such as data collection, posture judgment, and user feedback. Hardware determines what information the system can obtain, while software is responsible for processing the information and determining whether to alert the user. The sensors and judgment methods used in different research institutes vary. The project needs to be determined based on specific monitoring objectives. This section reviews the hardware and software research used in current studies, providing a basis for the design of this project.

2.1 Hardware

This section compares single IMU systems, multiple inertial sensors, flexible strain sensors, and chair based sensors. The focus of the comparison is on what each system measures, how the sensors are connected, and its applicability to monitoring the posture of the target. Sensor connections and housing design should also be considered as they can affect measurement conditions.

2.2 Attitude Monitoring Based on a Single IMU

IMU is a common sensor used in wearable devices. An accelerometer can record information related to the direction of gravity and motion acceleration, while a gyroscope provides angular velocity information. After processing, these signals can be used to calculate angles. Compared to directly observing human body images, IMU can record the motion and direction information of the installation position. Therefore, it is necessary to clarify the relationship between sensor output and target posture [4].

Park and Jung developed a head forward tilt monitoring system using a single MPU6050 and Raspberry Pi. They installed the devices on the forehead, chest, back of the neck, and near the cervical spine, and selected the vicinity of the cervical spine as the monitoring location under testing conditions [6]. This study suggests that a single IMU can serve as a sensor in wearable devices, but its installation location needs to be determined through experimentation and cannot be chosen solely based on comfort and convenience.

For this project, a single IMU provides a simpler prototype structure, making it easier to check data and calculate angles. However, the information obtained by a single IMU is relatively limited compared to devices with multiple sensors.

2.2.1 Attitude Monitoring Based on Multiple Inertial Sensors

Multiple inertial sensors can measure the orientation of different body parts, providing more information for analyzing changes between different body parts. There are three sensor modules in the smart clothing developed by Wong. These three modules are installed near the torso, middle torso, and pelvis. The system utilizes the inclination angle changes between adjacent modules to describe the posture changes of the chest and waist [8].

Wong's research did not rely solely on the tilt of one location but rather compared the relationships between different locations. For example, when the entire torso tilts forward, different positions may produce different angle changes. The author also pointed out that adding sensors can obtain more

posture information. But at the same time, it will also reduce the practicality of the system. We need to balance measurement capability with actual usage [8].

Measurements from multiple body positions can provide information about partial relative motion [8]. This may help distinguish between local neck movement and overall trunk tilt, but it also requires additional attachment points and handling. For the current single IMU prototype, the ability to distinguish these movements needs to be evaluated through real experiment.

2.2.2 Flexible Strain Sensor

Flexible strain sensors can generate electrical signals that reflect deformation through the stretching or bending of materials. This sensor is usually combined with flexible materials. Huang et al.'s study compared flexible strain sensors with inertial sensors, pressure sensors, and other sensors. It shows that different sensors are suitable for different human movements [4].

There are also many good designs for flexible sensors to monitor back bending. But it records the deformation of the sensor itself, and cannot directly use the output value as the neck angle. For this project, if this sensor is used, it is necessary to further check whether the tightness, fixed position, and initial stretching state of the clothing will change the measurement results.

Flexible strain sensors are more suitable for comparison as another design solution. Whether it is suitable for head forward tilt monitoring in this project needs to be judged based on actual usage.

2.2.3 Seat Based Posture Monitoring

There is also a type of research that installs sensors on seats and uses seat cushion pressure or the distance between the human body and the backrest to determine sitting posture. Jeong and Park designed an intelligent seat equipped with six pressure sensors and six infrared reflective distance sensors. The pressure sensor is installed inside the seat cushion, and the distance sensor is installed on the seat back. The system uses these signals for pose classification [9].

This solution does not require the device to be fixed on the human body. However, for this project, more attention is paid to monitoring the forward tilt of the head. If the user only changes the tilt angle of the head and the change in the torso is small, these signals may be too small to monitor.

Therefore, smart seat systems and wearable devices cannot be replaced. The former is more suitable for studying the relationship between the human body and the seat, while the latter focuses more on fixed target movements. Choosing what kind of model requires first determining the body parts and changes that need to be measured, and then deciding what system to use.

2.2.4 Wearing Structure and Fixing Method

In addition to choosing the type of sensor, the connection method between the sensor and the body can also affect the measurement results. Huang et al.'s study suggests that the wearing structure needs to consider comfort, measurement accuracy, and installation position, and the lack of a stable measurement environment may affect the measurement results [4]. Wong's research uses close fitting clothing to fix multiple sensors, enabling them to correspond to different body parts [8].

These studies indicate that the shell and fixed structure of this project should be considered in

conjunction with the measurement method. It is necessary to check whether the shell is stable, whether the wiring will affect the sensor, and whether it can maintain a similar direction after each re wearing. These factors may all affect the measurement results. Therefore, the role of the casing is not only to accommodate electronic components, but also to help the device maintain stability and consistency.

Figure 1 shows the three sensor arrangements discussed earlier. The single IMU scheme records local data, the multi-sensor scheme further compares different body parts, and the intelligent seat type scheme collects seat cushion pressure and backrest distance. This diagram is used to illustrate the measurement approaches of different studies.

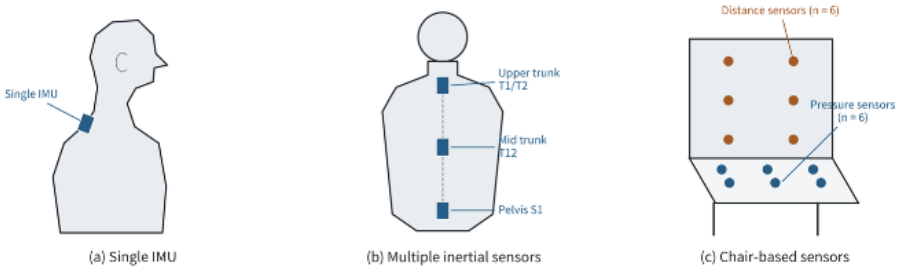


Figure 1. Original Schematic of Sensor Arrangements Based on (a) Park and Jung [6], (b) Wong and Wong [8], and (c) Jeong and Park [9].

2.3 Software

This section reviews the steps used to explain sensor measurements and provide posture feedback. Firstly, differentiate the attitude indicators and sensor orientation indicators, and then discuss signal processing, personal calibration, classification, and reminder time. The user interface is considered a way of presenting result information.

2.3.1 Posture Indicators and Angle Definitions

Before the system makes posture judgments, it is necessary to first determine the meaning of each input data. CVA is a commonly used data indicator for studying head forward tilt, defined as the angle between the line connecting the tragus and C7 relative to the horizontal line. Lau et al. studied the accuracy of measuring CVA using electronic devices, which provides a reference for the measurement of forward leaning posture in this project [3].

The IMU tilt angle describes the direction relative to the reference direction, which is different from the definition of CVA. Park and Jung's research defined the corresponding relationship between the two [6]. Therefore, when referring to reference materials, it is necessary to consider the reference direction of installation position, sensor direction, and angle.

Figure 2 shows the orientation angle of the decomposed pose indicator and sensor. The left image shows CVA, which is defined using C7, tragus, and horizontal reference [3,6]. The right panel displays the sensor angle relative to the vertical reference. The acceleration based calculation used in this system is described in section 3.2.

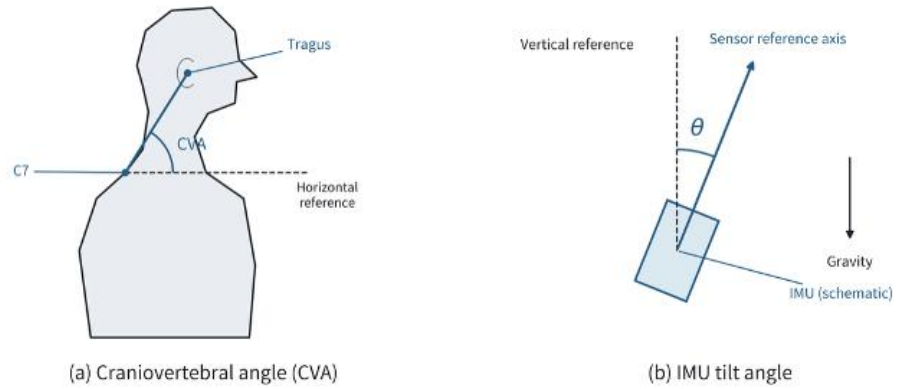


Figure 2. Original Schematic of (a) CVA, Following the Anatomical Definition in [3,6], and (b) an Illustrative IMU Orientation Angle Relative to a Vertical Reference.

2.3.2 Signal Processing and Direction Estimation

For this project, after obtaining the raw data, it is necessary to choose an appropriate method to process the data. Valenti et al. proposed a quaternion complementary filtering method that combines information from gyroscopes and accelerometers to estimate direction [10]. This study mainly focuses on micro aerial vehicles and cannot be directly applied to the forward leaning posture of the head, but the data processing methods can be referred to.

In the process of direction estimation, the gyroscope records the result of rotation, but the integration error may accumulate over time. Acceleration can provide a reference for gravity but may be influenced by other accelerations [10]. So the two signals have their own purposes and need to be processed according to the actual situation.

The project should first check whether the output results under static and slow tilting conditions are correct and then determine whether adjustments are needed. During inspection, attention should be paid to both output fluctuations and response speed. The method of handling should be determined based on the actual test results.

2.3.3 Personal Calibration and Duration Assessment

Calibration can establish a personal reference posture for each user. Park and Jung's study set baselines for each individual and monitored the time of deviation from the baseline to remind them. His research sets reminders for durations exceeding 10 seconds [6]. This method provides a reference for judgment in this project.

Wong's project also used a similar method for monitoring the torso, using reference postures, time ranges, calibration, etc. However, since they were monitoring the torso, the parameters set were not the same [8]. The setting of parameters is related to the test object, and the values at different positions cannot be directly used as a universal standard for all devices.

For this project, it is necessary to clarify the standard posture and collection time during calibration. The judgment of posture requires assessing the degree and duration of deviation, such as briefly

looking down at the keyboard. The final threshold and duration need to be set based on the actual situation to check for excessive error alerts.

2.3.4 Posture Classification Method

The judgment of posture can be based on clear judgment rules or machine learning methods. Fixed judgment rules can be set with angles, reference values, and durations to facilitate reminders. Jeong and Park's study used k-nearest neighbor algorithm to classify sitting posture based on pressure signals collected by sensors on the seat and distance signals collected by sensors on the seat back [9].

Different judgment methods require different methods to verify. For clear judgment rules, it is necessary to determine threshold and time rules that can correctly distinguish the target posture from normal posture changes. For machine learning methods, labeled data is required, and independent testing data is used to check whether the model can be tested correctly.

At present, the main focus of this project is on measuring the forward tilt of the head. Clear judgment rules will be adopted first, which will help the project analyze the relationship between signals and posture. Other more accurate methods can be considered in the future to make the project more precise.

2.3.5 User Interface and Feedback

User feedback is also an important part of posture monitoring systems. Figueira et al.'s research shows that posture feedback in the workplace can be alerted through the device itself or a smartphone [7]. These methods provide reference for the application interface design of this project.

The study by Ailneni et al. showed that some head and neck postures and mechanical indicators of users changed when typing on wearable devices [5]. This indicates that the feedback function needs to be combined with the user's real-time action type and cannot simply issue reminders based on numerical values.

For the IOS software of this project, it can be designed around calibration operations, status, and reminder information. Reminder information can include the status, whether posture needs to be adjusted, etc. The final design needs to be determined based on actual test results.

Table 1 shows the hardware configuration, monitoring objects, and processing methods mentioned earlier.

Table 1. Hardware Configuration, Monitoring Objects, and Processing Methods

Study	Devices or methods	Monitoring target	Relevance to this project
Wong and Wong [8]	Three inertial sensor modules integrated into a smart garment	Thoracic and lumbar posture changes	Measurement at multiple locations and sensor attachment
Jeong and Park [9]	Pressure and distance sensors with a k-nearest neighbor algorithm	Seated posture classification	Comparison of chair-based and body-worn sensing
Park and Jung [6]	Single IMU with an individual baseline and duration criterion	Forward head posture	Sensor placement, calibration, and alert criteria

Valenti et al. [10]	Quaternion-based complementary filtering	Orientation estimation using inertial sensors	Fusion of accelerometer and gyroscope data

2.3.6 Design Implications

Research has shown that attitude monitoring relies on both the measured signal and attachment conditions. A single IMU provides a simple sensing arrangement, while multiple sensors can provide more motion information. Therefore, the appropriate configuration depends on the motion that needs to be distinguished [4,8].

This project uses a single MPU6050 with personal baseline and duration based reminders. The evaluation will examine the measurement of inclination, baseline repeatability and warning response, as well as the relationship between local sensor inclination and target attitude. Section 3 describes the implementation and testing process.

3. Methodology

The monitoring of the forward tilt angle of the head during the use of computer scenarios in this study. The prototype design consists of inertial sensors, control board, and software. The development process first completes sensor data collection and tilt angle calculation, and then considers wearing structure, personal calibration, and posture judgment. This allows for separate checks on the relationship between hardware output, calculation methods, and target posture.

3.1 Hardware Design

This section describes prototype components, electrical connections, and sensor housings. It also explains the recommended installation location and additional conditions required for sensor output.

3.1.1 System Composition

The prototype uses the GY-521 module of Arduino UNO R4 and MPU6050. Sensor data is obtained through I ² C. Arduino calculates tilt angle through acceleration measurement. Connect to the computer via USB for program upload and serial monitoring.

The supporting iOS software has also been developed. The interface and connection with the sensor system were evaluated separately. Figure 3 distinguishes the implemented I ² C and USB serial paths from the application data connection, which needs to be verified through comprehensive operations.

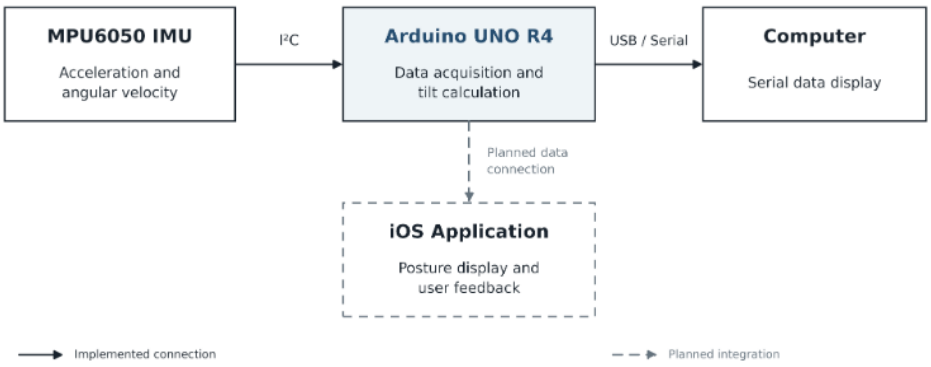


Figure 3. System Architecture

3.1.2 Hardware Connection

MPU6050 communicates with Arduino UNO R4 through I ² C. SDA carries data signals, while SCL carries clock signals. On UNO R4, A4 corresponds to SDA and A5 corresponds to SCL [11]. The power and logic connections follow the requirements of the actual GY-521 module used. The actual connection is shown in Figure 4.

After wiring, check whether the sensor readings will react when changing slowly. Part 4 reports on development observations and support logs.

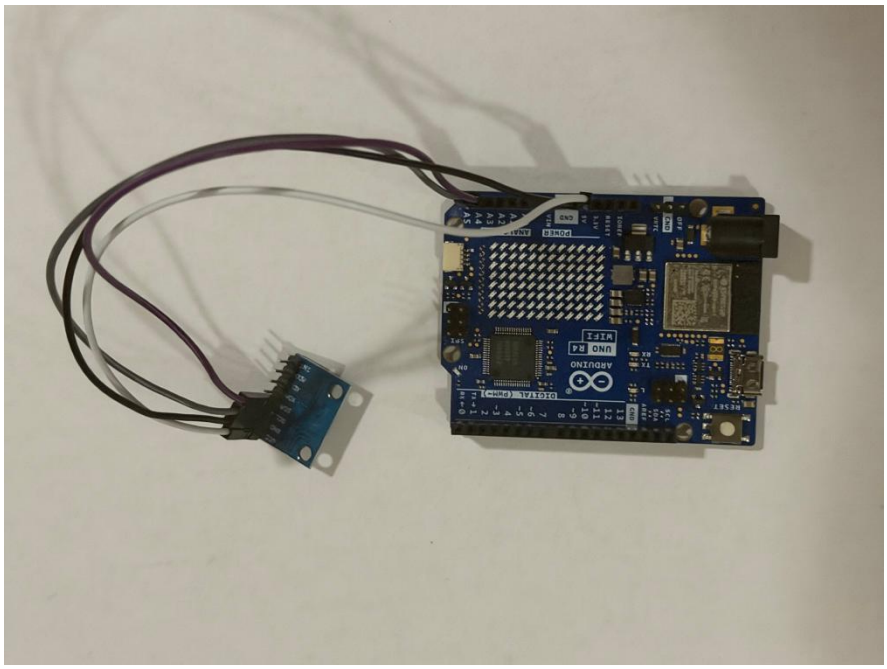


Figure 4. Actual Connection of the Prototype

3.1.3 Enclosure and Sensor Placement

The casing is designed to secure an MPU6050 module. And open in advance to provide cable

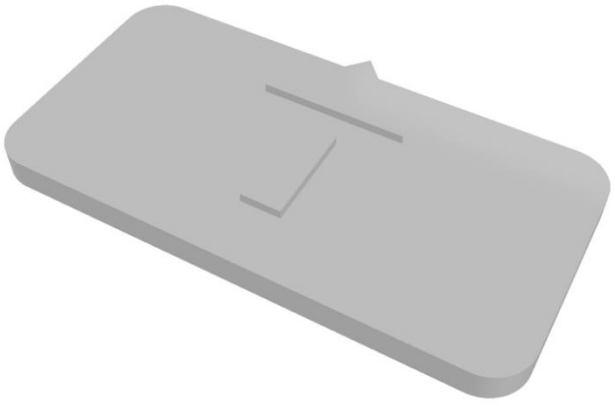
penetration.

Based on previous research on cervical posture monitoring [6], the vicinity of C7 was selected as the installation location. When a fixed sensor rotates with the surface of a local object, the gravity component measured along its axis changes. This allows for the calculation of changes in the tilt of individual baseline sensors. This change may be accompanied by neck and upper trunk movements in FHP patients. Head translation, torso tilt, etc. may also occur at the installation position, which can affect the sensor readings. Therefore, there needs to be a genuine evaluation of the relationship between the measurement signal and the target attitude.

The shell is fixed with tape at the bottom of the neck and placed near C7 (the lowest cervical spine). For repeated placement, use the same bone marker at the bottom of the neck as a placement reference. The chassis design is shown in Figure 5.



(a).



(b).

Figure 5. CAD Models of the Enclosure: (a) Enclosure Body and (b) Cover.

3.1.4 Preliminary Hardware Inspection

The preliminary inspection mainly checks the data recording, output mode, and response to slow directional changes of the sensors. These checks can basically confirm the correct operation of the equipment. Further measurement of known angles is required to quantify tilt errors, and body installation testing is needed to evaluate the relationship with the target posture.

3.2 Software Design

3.2.1 Program Structure and Running Process

This study developed a control program in Arduino IDE and used Adafruit_SPU6050, Adafruit_Sensor, and Wire library to complete the sensor setup. The program is divided into five modules according to its functions: main program, IMU data processing, reference posture calibration, posture classification, and feedback, making the data acquisition and judgment process clearer.

The main functions of each module are shown in the table below.

Table 2. Organization of the Posture-monitoring Program

Program file	Main functions
FHP_posture_system.ino	Global parameters, initialization, and the main loop
IMU.ino	Sensor initialization, acceleration filtering, and tilt calculation
Calibration.ino	Three-trial reference-posture calibration and consistency checking
PostureDetection.ino	Angular-deviation calculation and posture-level classification
Feedback.ino	Duration tracking, serial feedback, and LED control

After the program starts, set the serial port baud rate to 115200, configure the onboard LED as output, and set the initial state to off. The MPU6050 is initialized before personal calibration begins. If initialization fails, the serial port outputs an error message, and the program stays in a waiting loop; After successful initialization, wait for 1 second and start reference posture calibration.

After calibration is completed, the program waits for 3 seconds to enter the monitoring phase. The main loop sequentially completes tilt angle reading, reference value comparison, state classification, and feedback update, and performs a 20-millisecond delay at the end of each loop. The program running process is shown in Figure 6.

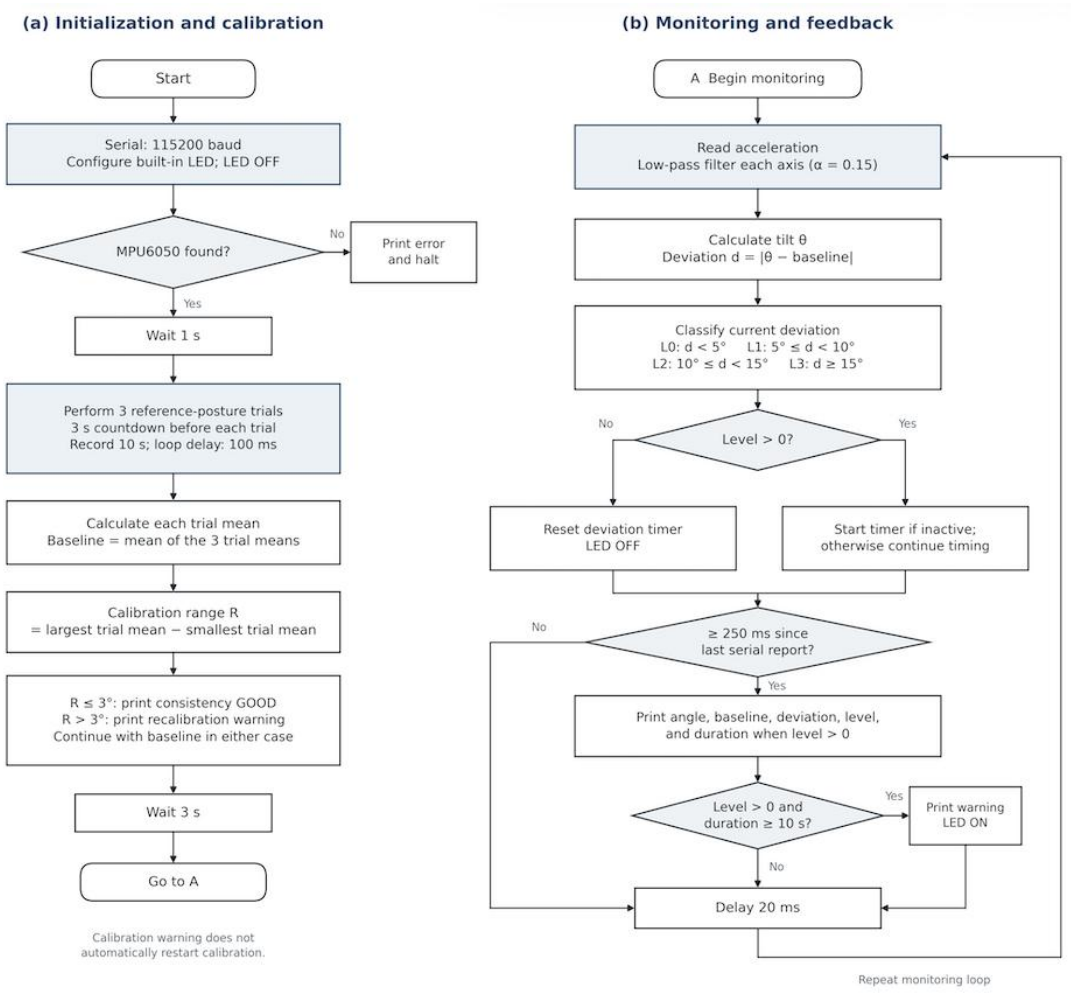


Figure 6. Complete Program Flowchart

3.2.2 Data Acquisition, Filtering, and Tilt Angle Calculation

The MPU6050 provides acceleration measurements along its x, y and z axes. The current tilt estimate is calculated from these acceleration components. Angular velocity is not combined with acceleration in this calculation.

$$\tilde{a}_{j,k} = (1-\alpha)\tilde{a}_{j,k-1} + \alpha\tilde{a}_{j,k}, j \in \{x, y, z\}$$

In the formula, $a_{j,k}$ is the currently read acceleration component, $\tilde{a}_{j,k}$ is the current filtering value, $\tilde{a}_{j,k-1}$ is the previous filtering value. This program sets the filtering coefficients to:

$$\alpha = 0.15$$

So each update retains 85% of the previous filtered value and adds 15% of the current measurement value. When reading for the first time, the program directly initializes the filtering variables with the first set of acceleration data to avoid gradually changing from zero.

After filtering is completed, the program calculates the tilt angle according to the following formula:

$$\theta_k = atan2(\tilde{a}_{x,k}, \sqrt{\tilde{a}_{y,k}{}^2 + \tilde{a}_{z,k}{}^2})\frac{180}{\pi}$$

The estimated angle is expressed in degrees, for acceleration vectors ranging from -90° to +90°. Its symbol follows the filtered x-axis acceleration and does not directly recognize forward or backward body movements. This estimate describes one aspect of the sensor direction, rather than the complete three-dimensional direction. Out of plane motion and additional motion acceleration can affect readings. Under ideal static conditions, rotation around the x-axis of the sensor does not alter this estimate, as the combined magnitude of y-axis and z-axis accelerations remains constant. Changing the rotation of the x-axis acceleration can alter the estimated angle.

The angle here belongs to the sensor tilt angle, which is different from the definition of CVA. The subsequent posture judgment should use changes relative to the personal reference value, rather than directly applying the absolute angle threshold of CVA.

The main program parameters are listed in Table 3.

Table 3. Software Parameters Used in the Current Implementation

Parameter	Setting
Serial baud rate	115200 baud
Low-pass filter coefficient	0.15
Number of calibration trials	3
Recording duration per calibration trial	10 s
Preparation countdown before each trial	3 s
Delay within each calibration sampling loop	100 ms
Delay at the end of the monitoring loop	20 ms
Serial reporting interval during monitoring	At least 250 ms
Angular-deviation thresholds	5°, 10°, and 15°
Continuous-deviation duration for warning	10 s
Calibration consistency criterion	Trial-mean range ≤ 3°

The calibration and monitoring circuits include delays of 100 ms and 20 ms, respectively. The actual usage process also depends on the collection, calculation, and serial output time. Different update intervals will also generate different filter responses over time. Therefore, measuring sampling intervals, output fluctuations, and response lag requires timestamp recording, fixed data, and control of tilt step size.

3.2.3 Three Reference Posture Calibrations

At the beginning of the program, a personal reference tilt angle is established through three independent tests. Before each measurement, the user will be prompted to maintain a reference sitting posture. After a preparation time of three seconds, the program repeatedly reads the filtered tilt angle within 10 seconds and calculates the average value of the record.

The average inclination angle for the r-th calibration is:

$$\bar{\theta_r} = \frac{1}{N_r}\sum_{i=1}^{N_r}\theta_{r,i}, r = 1,2,3$$

In the formula, N_r is the actual number of samples obtained in this record $\theta_{r,i}$ is the corresponding inclination angle. The program calculates the average value using actual counts.

After completing three records, take the arithmetic mean of the three average values of the reference inclination angle:

$$\theta_0 = \frac{\bar{\theta}_1 + \bar{\theta}_2 + \bar{\theta}_3}{3}$$

To check the repeatability of the reference posture, the program examined the range between three averages:

$$R_{cal} = \max(\bar{\theta}_1, \bar{\theta}_2, \bar{\theta}_3) - \min(\bar{\theta}_1, \bar{\theta}_2, \bar{\theta}_3)$$

In the program, when $R_{cal}$is less than or equal to 3 degrees, the program outputs a prompt indicating good calibration consistency; When it exceeds 3 degrees, it prompts to check the sensor position and reference posture and recommends recalibration. This indicator reflects the consistency of the three average inclination angles and does not represent accuracy.

Consistency warning but will not end baseline calculation, monitoring will continue even if it exceeds 3°. So, the user needs to reset the Arduino and perform baseline measurements again. During the normal measurement process, calibration will not be performed again. Keep the filter state between the three records in the calibration session.

3.2.4 Angle Deviation and State Classification

After calibration is completed, the program will compare the real-time tilt angle with the personal reference angle. The deviation calculation used in the project is:

$$d_k = |\theta_k - \theta_0|$$

In the formula d_k represents the magnitude of the deviation of the current inclination angle from the reference value. Its direction is not recorded, and the offset generated in different directions has the same effect. So the current output is interpreted as sensor tilt deviation. Specific forward detection requires a separate determination of installation direction, comparing forward and backward movements and torso tilt.

The program uses three thresholds to classify deviations into four levels:

$$L(d) = \begin{cases} 0, & 0 \le d < 5°, \\ 1, & 5° \le d < 10°, \\ 2, & 10° \le d < 15°, \\ 3, & d \ge 15°, \end{cases}$$

The corresponding states are shown in Table 4.

Table 4. Posture-deviation Levels Defined in the Program

Level	Absolute angular deviation	Interpretation
0	$0 \leq d < 5°$	Within the reference tolerance
1	$5° \leq d < 10°$	Small angular deviation
2	$10° \leq d < 15°$	Moderate angular deviation
3	$d \geq 15°$	Large angular deviation

These thresholds are engineering settings used to describe the angular deviation between the true pose and the baseline. It cannot represent the clinical grading of FHP severity.

3.2.5 Duration Assessment and Feedback

Start timing when the angular deviation reaches 5° or greater for the first time. The duration passed is calculated based on the difference between the current timestamp and the start timestamp. The calculation of continuous deviation time is:

$$T_{dev} = \frac{t_{now} - t_{start}}{1000}$$

If the deviation is always not less than 5°, the timing will continue. The changes between levels 1, 2, and 3 will not start timing again. When the deviation returns to below 5°, the timing status will be cleared, and the onboard LED will be turned off.

During the measurement period, the serial report is generated at least once every 250 milliseconds. Each report includes the current tilt angle, baseline angle, angle deviation, and status level. When the deviation is at least 5°, the continuous deviation duration is also reported. Update status classification in each monitoring cycle. The 10 second condition only applies to warning activation.

During the first report update, if the deviation remains at 5° or above and lasts for 10 seconds, a warning will be generated. Serial output displays warnings and requests to adjust posture, while the built-in LED is turned on. In subsequent report updates, this warning will be repeated when the deviation remains at 5° or above. The actual activation time depends on the report plan and processing time, so it must be evaluated using the recorded timestamp.

Currently, different levels use the same 10 second warning condition and LED feedback method, without setting different flashing modes or vibration intensities based on severity. The current version also does not have a separate recovery threshold set, so state switching and timing reset may occur when fluctuating around 5°.



(a). (b).

Figure 7. Serial-monitor Output Showing (a) Three-trial Calibration and Monitoring Near the Reference Angle and (b) Sustained Sensor-tilt Deviation with Warning Messages after the Displayed Duration Exceeds 10 s.

3.2.6 Software Verification and Companion Application

Software inspection needs to include initialization, data processing, calibration, classification, and feedback. Initialization is used to check whether the sensor is successfully connected and whether it can output correctly. Angle inspection requires observing the output under static and slowly tilted conditions, as well as the reading changes after restoring the sensor to its original position.

The calibration section requires checking the average value of three times, the final reference value, and the range. Check if the program provides corresponding prompts based on the 3° condition. It is also necessary to check whether the outputs near 5°, 10°, and 15° are correct.

It is also necessary to check the continuous deviation of timing less than 10 seconds and reaching 10 seconds, and check whether the timing is cleared and the LED is turned off when returning to the allowable range. The project also needs to check if the timing of level changes is continuous. The

verification results, including calibration records, warning timing and application demonstrations, are presented in Section 4.

The IOS application supporting the project has been developed as the user interface for the project. Applications are used to evaluate demonstration functionality, user actions, and display responsiveness. The source of the displayed values is identified as default values, demonstration data, or data received from sensors. Compare Arduino output with values and warnings in the application to evaluate software performance. Only a complete and authentic data path can establish synchronous feedback.

3.3 Experimental Evaluation

Evaluate the main design of inclination measurement, calibration repeatability, and feedback response. Controlled testing will differentiate the behavior of sensors and algorithms from their correlation with FHP. Will record testing conditions, reference instruments, and actual observation results.

3.3.1 Inclination Measurement and Posture Response

The sensor is fixed on a bracket that can be adjusted separately. Set to -20°, -10°, 0°, 10°, and 20° within the measurement plane. Record 10 seconds of output for each setting. This sequence will be repeated three times. Compare the average sensor angle with the measured reference angle and record the deviation and fluctuation each time.

The actual collection interval will be recorded during the calibration and monitoring phase. A controllable tilt step will be used to evaluate the filtered response. The response time will be measured from the recorded step to the point where the output reaches 90% of its final change. The acquisition interval will be distinguished from the continuous reporting interval.

A separate test will compare neutral posture, head forward movement, backward movement, and torso tilt with sensors connected near C7. The sensor recording and side CVA measurement are used to check the relationship between tilt and head posture. The current algorithm uses absolute deviation but will retain signature changes from the baseline for analysis.

3.3.2 Calibration Repeatability

When the sensor is fixed, users will complete three 10 second calibration records before starting to record real-time data. Record the average angle during the three processes, the personal baseline used, the calibration range, and consistency prompts. After dismantling, reinstalling will repeat the complete calibration process again. The results from one user will not be considered as evidence of consistency across different users.

3.3.3 Warning Timing and Recovery

After calibration, the deviation will remain above 5° for nominal durations of 5 seconds, 10 seconds, and 15 seconds. Each situation will be repeated three times, with a recovery rate of less than 5° between each experiment. The actual deviation, duration, warning time, and LED status will be recorded. The 10 second condition will be considered as a boundary test, as the activation of warnings also depends on the reporting schedule.

Additional checks will examine the recovery below 5°, continuous timing during changes between

non-zero levels, and fluctuations near the 5° boundary. The level of the report will also be checked around the thresholds of 5°, 10°, and 15°. The observed reactions will be compared with program rules, including any premature warnings, missed warnings, or unexpected resets.

4. Results

This section evaluates the prototype in terms of inclination measurement, calibration repeatability and warning response. The completed companion iOS application is also presented as a software development outcome. Experimental observations are considered separately from the functions implemented in the program.

4.1 Inclination Measurement and Posture Response

Figure 8 shows the comparison between the actual sensor angle and the reference angle. Use 5 reference angles: -20°, -10°, 0°, 10°, and 20°. Conduct three tests at each angle. The plotted points represent their mean values, and the error bars represent the sample standard deviation between the mean values of three experiments.

The average estimated values are -19.73°, -9.83°, 0.06°, 10.17°, and 20.32°, respectively. Their average absolute difference from the reference angle is 0.198°. The error bar also describes the variation between the average values of the experiment, rather than the variation between sensor noise within a fixed record.

The results show that the actual measurement results are basically consistent with the reference angle. But this can only support sensor tilt measurement. It cannot be determined whether the device can distinguish between forward head movement, backward head movement, or trunk tilt.

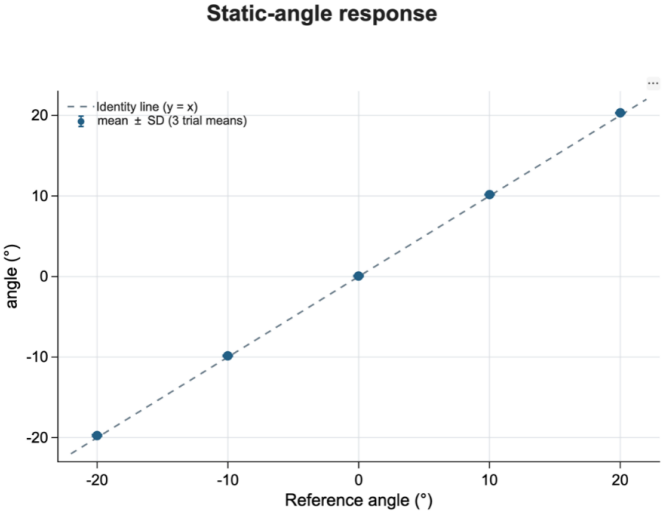


Figure 8. Comparison between Reference Angles and Sensor-angle

4.2 Calibration Calculation

Table 5 shows an example of calibration calculations before and after modifying the casing. Each test

value represents the average angle of a calibration record. The baseline used is the average of three experimental means, and the range is the difference between the maximum mean and the minimum mean. For the initial position, the example baseline is 18.40° and the range is 0.40°. After modification, the baseline is 20.37° and the range is 0.30°. Both ranges comply with the programming's 3° consistency criteria. With these inputs, the software will generate corresponding consistency messages.

This example demonstrates how a single baseline can explain the initial angular deviation after modification. But it cannot prove the repeatability or calibration accuracy of actual placement. In the current implementation, a warning will be generated for ranges greater than 3°, but the calculated baseline will still be retained and monitored.

Table 5. Calibration Example before and after Refitting the Enclosure

Placement condition	Trial 1 mean (°)	Trial 2 mean (°)	Trial 2 mean (°)	Baseline (°)	Range (°)
Initial placement	18.20	18.60	18.40	18.40	0.40
After refitting	20.20	20.50	20.40	20.37	0.30

4.3 Reminder Logic and Verification Status

Real device reminders indicate the magnitude and duration of sensor tilt deviation. Start timing when the absolute deviation reaches 5 °. Changes between levels 1, 2, and 3 will not restart the timer. When the deviation returns below 5 °, the timer is cleared.

According to the program logic, a continuous 5-second deviation will not trigger a warning. If the deviation remains at 5 ° or above for 10 seconds or more, an alarm will be triggered and will be issued on the next serial reporting opportunity. During a continuous 15 second deviation period, subsequent reports will repeat warnings and the LED will remain on until restored.

Table 6 shows the actual records. This includes the duration of the deviation, the actual response status, whether the serial port reminds, LED status, etc. And it also records whether the time record after recovery is reset to zero.

Table 6. Reminder and Recovery Behavior Based on the Embedded Program

Deviation duration (s)	First warning time	Serial warning	LED state during deviation	Timer reset after recovery
5	No warning	No	Off	Yes
10	Depends on reporting and recovery timing[1]	Conditional[1]	On if warning is triggered; otherwise off	Yes
15	First eligible serial report at or after 10 s	Yes, repeated in subsequent reports	On after warning activation	Yes

4.4 Companion iOS Application

The project also includes the development of a supporting iOS application. Figure 9 shows the application program interface. This work provides a user oriented part of the prototype, as well as embedded programs and shells. The development of this application is separate from hardware to mobile communication. The picture shows the main functions of the software.

The software still needs to undergo a comprehensive evaluation, comparing Arduino records with the values displayed in the application, and checking whether updates and reminders remain consistent throughout the entire data path.

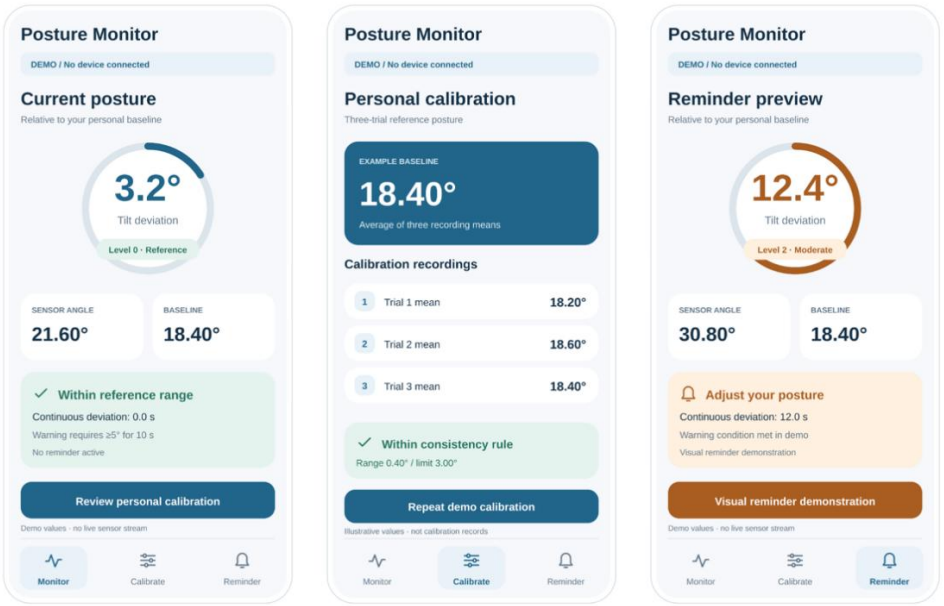


Figure 9. Interface Prototype Showing Monitoring (Left), Personal Calibration (Center), and a Reminder Preview (Right).

4.5 Discussion and Limitations

The project has already produced a hardware prototype, an embedded processing program, a shell design, and a supporting iOS application. These development achievements provide a foundation for further evaluation. Calibration and duration testing were also conducted.

One major limitation is the meaning of the measured signal. Estimate the tilt of the installation position of a single MPU6050. Changes in neck and trunk movements can affect this value. The absolute deviation rule also handles changes in two directions in the same way. Therefore, the current output is described as a sustained sensor tilt deviation, rather than a confirmed head forward posture.

Personal calibration provides a reference for subsequent comparisons, but its effectiveness still depends on the user's reference posture and repeated placement. The thresholds of 5°, 10°, and 15°, as well as a 10 second warning duration, are set for the project. Whether they are suitable for daily use has not been determined yet.

Further work should include accuracy and comfort testing for actual use. It is also necessary to compare the control of forward movement, backward movement, and trunk tilt. Finally, before declaring real-time monitoring or synchronous reminders, it is necessary to evaluate the communication between hardware and applications.

5. Conclusion

This project developed a wearable sensor tilt monitoring prototype using Arduino UNO R4 and MPU6050. The completed work includes hardware assembly, shell design placed near C7, embedded processing programs, and supporting iOS applications. Embedded programs handle acceleration measurements, estimate sensor tilt, and establish personal baselines from three calibration records. Then, it classifies the absolute angle deviation and uses its duration to determine the serial warning and LED feedback.

The current prototype provides a foundation for further testing. The real test data is presented in Section 4. In addition, the sensor measures the inclination of its installation position, rather than the position of the head relative to the torso. Both forward and backward changes can produce the same absolute deviation, and changes in torso movement or attachment can also affect the output. Therefore, the current system should be described as monitoring continuous sensor tilt deviation. The iOS application has been developed, but real-time sensor data transmission and synchronized reminders still need to be verified.

The next stage is to evaluate the prototype using recorded measurement values. And evaluate the actual usage status and comfort. Repeated fitting and calibration should be performed to check the consistency of placement. Finally, it is necessary to compare forward movement, backward movement, and torso tilt, as well as hardware to application testing, to determine the level of system support for its intended use.

References

[1] Straker, L., Burgess-Limerick, R., Pollock, C., Murray, K., Netto, K., Coleman, J., & Skoss, R. (2008). The impact of computer display height and desk design on 3D posture during information technology work by young adults. *Journal of Electromyography and Kinesiology*, *18*(2), 336-349. https://doi.org/10.1016/j.jelekin.2006.10.007

[2] Mahmoud, N. F., Hassan, K. A., Abdelmajeed, S. F., Moustafa, I. M., & Silva, A. G. (2019). The relationship between forward head posture and neck pain: A systematic review and meta-analysis. *Current Reviews in Musculoskeletal Medicine*, *12*(4), 562-577. https://doi.org/10.1007/s12178-019-09594-y

[3] Lau, H. M. C., Chiu, T. T. W., & Lam, T. H. (2009). Clinical measurement of craniovertebral angle by electronic head posture instrument: A test of reliability and validity. *Manual Therapy*, *14*(4), 363-368. https://doi.org/10.1016/j.math.2008.05.004

[4] Huang, X., Xue, Y., Ren, S., & Wang, F. (2023). Sensor-based wearable systems for monitoring human motion and posture: A review. *Sensors*, *23*(22), 9047. https://doi.org/10.3390/s23229047

[5] Ailneni, R. C., Syamala, K. R., Kim, I.-S., & Hwang, J. (2019). Influence of the wearable posture correction sensor on head and neck posture: Sitting and standing workstations. *Work*, *62*(1), 27-35. https://doi.org/10.3233/WOR-182839

[6] Park, G., & Jung, I. Y. (2024). Real-time forward head posture detection and correction system utilizing an inertial measurement unit sensor. *Applied Sciences*, *14*(19), 9075. https://doi.org/10.3390/app14199075

[7] Figueira, V., Silva, S., Costa, I., Campos, B., Salgado, J., Pinho, L., Freitas, M., Carvalho, P., Marques, J., & Pinho, F. (2024). Wearables for monitoring and postural feedback in the work context: A scoping review. *Sensors*, *24*(4), 1341. https://doi.org/10.3390/s24041341

[8] Wong, W. Y., & Wong, M. S. (2008). Smart garment for trunk posture monitoring: A preliminary study. *Scoliosis*, *3*, 7. https://doi.org/10.1186/1748-7161-3-7

[9] Jeong, H., & Park, W. (2021). Developing and evaluating a mixed sensor smart chair system for real-time posture classification: Combining pressure and distance sensors. *IEEE Journal of Biomedical and Health Informatics*, *25*(5), 1805-1813. https://doi.org/10.1109/JBHI.2020.3030096

[10] Valenti, R. G., Dryanovski, I., & Xiao, J. (2015). Keeping a good attitude: A quaternion-based orientation filter for IMUs and MARGs. *Sensors*, *15*(8), 19302-19330. https://doi.org/10.3390/s150819302

[11] Arduino. Wire Library— I²C communication and board pin assignments. Official Arduino reference.