ICE-3203
MICROPROCESSOR AND MICROCONTROLLER

DAGLI, LAUREN CHRISTIAN A. **I** **GUEVARRA**, KENJI C. **I** **MAYARI**, KURT VOLTAIRE P.

LINGAP: An IoT-Integrated Device for Elderly Stress Detection

A. DESIGN CONSTRAINTS

In this community-based project, proponents have to consider several design constraints:

- **User Comfort**. The device to be manufactured must be lightweight and comfortable for the elderly to wear for extended periods.
- **Accuracy**. It should display precise temperature and heat stress measurements.
- **Robustness**. The device must be durable enough to be used repeatedly.
- **Battery Life**. It must have a long-lasting power source.
- **User-Friendly**. The device must be easy to use.
- **Cost**. It must be affordable for everyone to ensure wider consumption.
- **Safety Standards**. The project must meet the safety standards and medical regulations related to its use.

B. DESIGN CONSIDERATIONS

Certain requirements are to be considered in creating the project that meets its purpose.

- **Accessibility.** Ensures inclusivity, making sure the design is usable for all.
- **Customer Preference.** Aligns the project with user needs, increasing satisfaction and usability.
- **Design Principles.** Provide a structural guide, ensuring aesthetics and functionality blend seamlessly.
- **Efficiency.** Enhance performance, minimizing waste and optimizing resources.
- **Extensibility.** Allow the project to adapt and grow over time, making future upgrades smoother.
- **Functionality.** Guarantee the project serves its intended purpose effectively.
- **Health and Safety.** Ensure that the final product does not pose risks to users or the environment
- **Lightness.** Contribute to practicality, making the design manageable without compromising strength or durability.

C. GENERAL OBJECTIVE AND SPECIFIC OBJECTIVES

In line with this project, general and specific objectives must be established and adhered to:

GENERAL OBJECTIVES

1. To create and implement a community-driven project to provide meaningful assistance to individuals, specifically the elderly.
2. To ensure the project aligns with the principles and objectives of the Sustainable Development Goals.
3. To actively contribute to the advancement of innovative solutions for future challenges.

SPECIFIC OBJECTIVES

1. To monitor health-related matters, supplementing the need for timely health care and guidance.
2. To create a compact system that allows the user to have real-time monitoring of their health status
3. To lessen human and machine interference, it is integrated compactly into a small microprocessor.

D. METHODOLOGY

METHODS AND PROCEDURES

1. *System Design and Planning*

   The proposed system was conceptualized and developed by the proponents to provide an efficient and accessible method of monitoring human stress levels through Galvanic Skin Response (GSR) signals. The system is built around a modular architecture that combines sensing, processing, alerting, and cloud communication components. At its core, the ESP32 microcontroller reads data from a GSR sensor, which detects variations in skin conductivity associated with emotional or physiological stress. These analog signals are then converted into voltage values and categorized into three stress levels—Low, Medium, and High—based on predefined thresholds.

   The classification process is handled locally by the ESP32, which also manages user feedback through a buzzer, vibration motor, and a set of colored LEDs representing the different stress states. To ensure real-time visibility, the system displays GSR values and corresponding stress levels on a 128x64 OLED screen. Furthermore, the microcontroller connects to the Blynk IoT platform over Wi-Fi to transmit sensor data, log events, and generate remote notifications when high stress levels are detected. This integration allows users and stakeholders to monitor stress data in real time from a remote location. The design emphasizes simplicity, reliability, and real-time feedback, making it suitable for applications in personal wellness monitoring, mental health tracking, or stress-related research.

2. *Hardware Assembly*

The proponents assembled the system using the following hardware components:

- **ESP32 Microcontroller**: Serves as the central control unit, managing all input/output operations and handling Wi-Fi connectivity for IoT integration.
- **GSR Sensor**: Connected to the analog pin (GPIO 34) of the ESP32, it detects changes in skin conductance.
- **OLED Display (SSD1306, 128x64 pixels)**: Connected via the I2C interface, it displays real-time GSR values and stress level indicators.
- **Buzzer (GPIO 26)**: Emits sound alerts during periods of high stress.
- **Vibration Motor (GPIO 27)**: Provides haptic feedback for both medium and high stress levels.
- **LED Indicators**:
  - **Green LED (GPIO 14)**: Indicates low stress.
  - **Yellow LED (GPIO 12)**: Indicates medium stress.
  - **Red LED (GPIO 13)**: Indicates high stress.
- **Power Supply**: The system is powered through USB or a battery source suitable for the ESP32 board.

All components were mounted and connected on a breadboard or prototyping board. Proper care was taken to avoid signal interference, and the wiring layout was organized to ensure both safety and readability during testing.

3. *Microcontroller code*

```
#define BLYNK_TEMPLATE_ID "TMPL68ah_C5e1"
#define BLYNK_TEMPLATE_NAME "GSR"
#define BLYNK_AUTH_TOKEN "eFbgkSOW52XoP7s3n_rQdXcqaolFG1pP"

#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#include <WiFi.h>
#include <BlynkSimpleEsp32.h>

// ==== WIFI CONFIG ====
char ssid[] = "Builder's Hive";
char pass[] = "Batstateubhive_0";
```

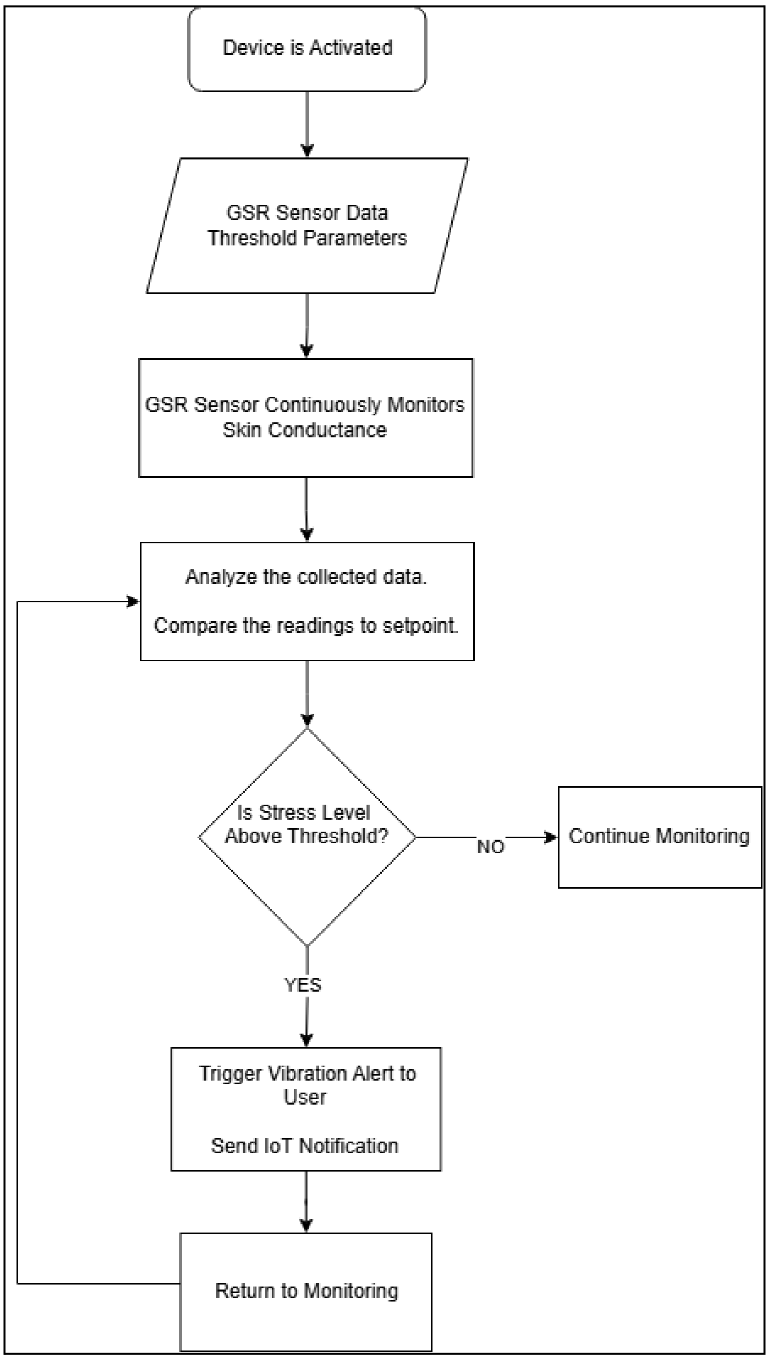
ICE-3203
MICROPROCESSOR AND MICROCONTROLLER

```
// ==== OLED CONFIG ====
#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64
#define OLED_RESET -1
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);

// ==== PIN CONFIG ====
#define GSR_PIN        34
#define BUZZER_PIN     26
#define VIBRATION_PIN  27
#define LED_GREEN_PIN  14
#define LED_YELLOW_PIN 12
#define LED_RED_PIN    13

// ==== STRESS THRESHOLDS ====
#define GSR_THRESHOLD_LOW     1000
#define GSR_THRESHOLD_MEDIUM  2500
#define GSR_THRESHOLD_HIGH    3500

BlynkTimer timer;
int lastStressLevel = 0;  // 0 = Low, 1 = Medium, 2 = High

void setup() {
  Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass);


  // Set pin modes
  pinMode(GSR_PIN, INPUT);
  pinMode(BUZZER_PIN, OUTPUT);
  pinMode(VIBRATION_PIN, OUTPUT);
  pinMode(LED_GREEN_PIN, OUTPUT);
  pinMode(LED_YELLOW_PIN, OUTPUT);
  pinMode(LED_RED_PIN, OUTPUT);
  // Initialize OLED
  if (!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
    while (true); // Halt
  }

  display.clearDisplay();
  display.setTextSize(1);
  display.setTextColor(SSD1306_WHITE);
  display.setCursor(0, 0);
  display.println("GSR Monitoring...");
  display.display();
  delay(2000);




  // Timer for GSR update every second
  timer.setInterval(1000L, sendGSR);
}

void sendGSR() {
```

```
int gsrValue = analogRead(GSR_PIN);
float voltage = (gsrValue / 4095.0) * 3.3;

String stressLabel;
int stressLevel;

// Stress classification
if (gsrValue < GSR_THRESHOLD_LOW) {
  stressLevel = 0;
  stressLabel = "Low";
} else if (gsrValue < GSR_THRESHOLD_MEDIUM) {
  stressLevel = 1;
  stressLabel = "Medium";
} else {
  stressLevel = 2;
  stressLabel = "High";
}

// Send to Blynk
Blynk.virtualWrite(V0, gsrValue);      // SuperChart
Blynk.virtualWrite(V1, stressLabel);   // Label
Blynk.virtualWrite(V2, stressLevel);   // Optional for graphs/indicators

// Alert if stress rises to High
if (stressLevel == 2 && lastStressLevel < 2) {
  Blynk.logEvent("stress_alert", "⚠️ High Stress Detected!");
}

lastStressLevel = stressLevel;

// OLED Display
display.clearDisplay();
display.setTextSize(1);
display.setTextColor(WHITE);
display.setCursor(0, 0);
display.print("GSR Value: ");
display.println(gsrValue);

display.setCursor(0, 16);
display.print("Voltage: ");
display.print(voltage, 2);
display.println(" V");

display.setCursor(0, 32);
display.print("Stress: ");
display.println(stressLabel);

display.display();

// Outputs
if (stressLevel == 2) {
  digitalWrite(BUZZER_PIN, HIGH);
  digitalWrite(VIBRATION_PIN, HIGH);
```

```
    digitalWrite(LED_GREEN_PIN, LOW);
    digitalWrite(LED_YELLOW_PIN, LOW);
    digitalWrite(LED_RED_PIN, HIGH);
  } else if (stressLevel == 1) {
    digitalWrite(BUZZER_PIN, LOW);
    digitalWrite(VIBRATION_PIN, LOW);
    digitalWrite(LED_GREEN_PIN, LOW);
    digitalWrite(LED_YELLOW_PIN, HIGH);
    digitalWrite(LED_RED_PIN, LOW);
  } else {
    digitalWrite(BUZZER_PIN, LOW);
    digitalWrite(VIBRATION_PIN, LOW);
    digitalWrite(LED_GREEN_PIN, HIGH);
    digitalWrite(LED_YELLOW_PIN, LOW);
    digitalWrite(LED_RED_PIN, LOW);
  }

  // Serial Debug
  Serial.print("GSR: ");
  Serial.print(gsrValue);
  Serial.print(" | Voltage: ");
  Serial.print(voltage, 2);
  Serial.print("V | Stress: ");
  Serial.println(stressLabel);
}


void loop() {
  Blynk.run();
  timer.run();
}
```

FLOWCHART



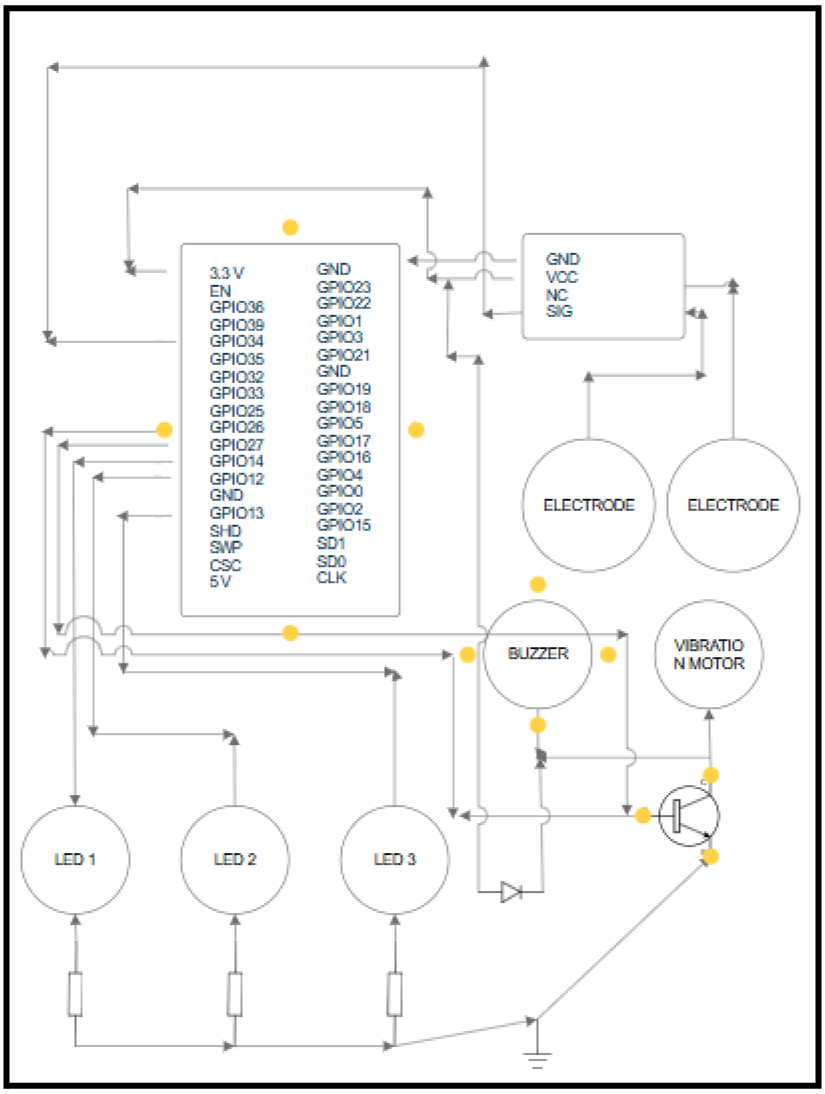
This section illustrates the "**LINGAP:** An IoT-Integrated Device for Elderly Stress Detection" process using a flowchart, highlighting the process variables that will be measured, controlled, and monitored.

The process begins with the activation of the device. Once activated, it measures body stress level through sweat conductance with the help of the gsr sensor. The proponents also set the threshold parameters for precise readings. The collected data is then analyzed and compared against the predefined setpoint. Based on this analysis, the system determines whether the maximum stress level has reached above threshold. If the collected data exceeds the maximum threshold, the device triggers a vibration alert, notifies the user and displays advisories or suggestions. And if the conditions remain within safe limits, the system continues monitoring without interruption.

ICE-3203
MICROPROCESSOR AND MICROCONTROLLER

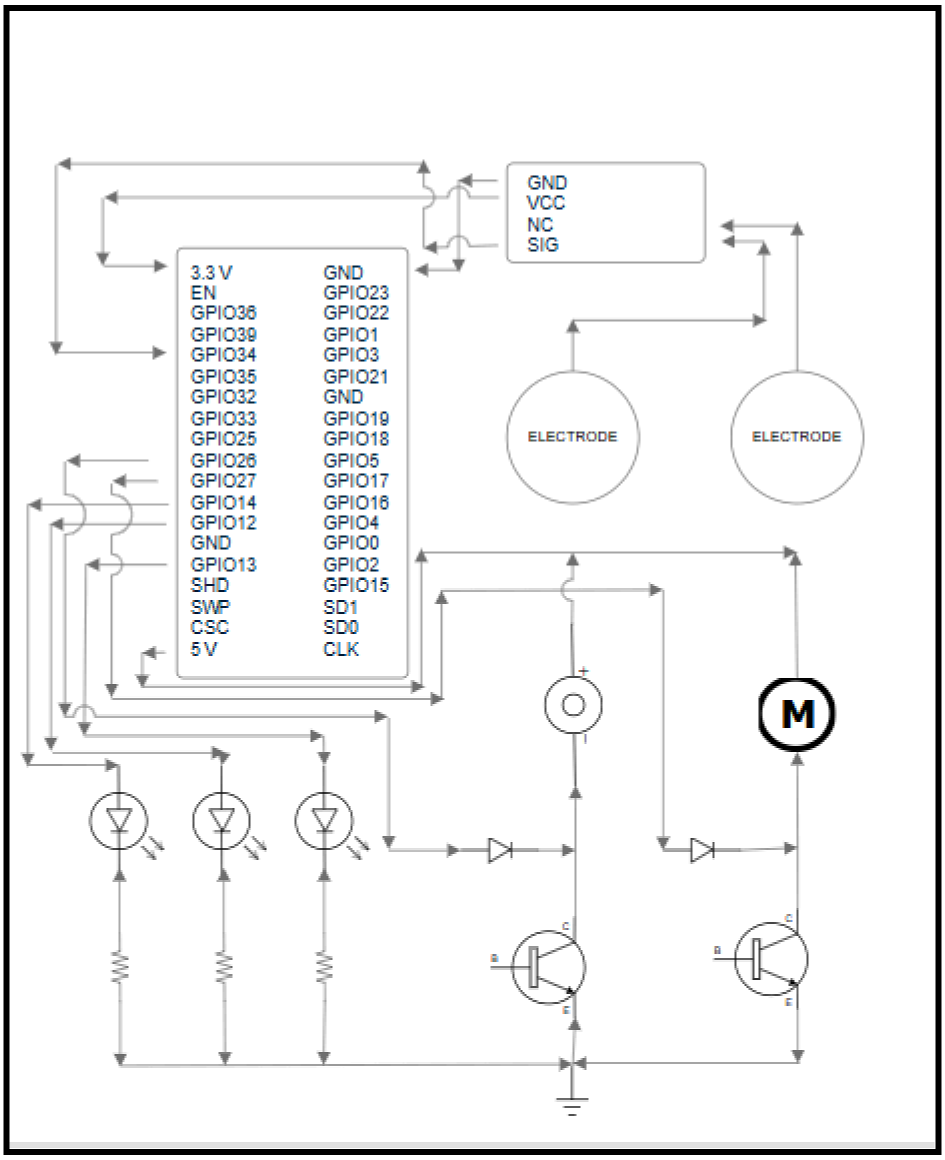
BLOCK DIAGRAM

INITIAL SCHEMATIC DIAGRAM (Without Op-amp, Vibration Motor, and Temperature sensor)



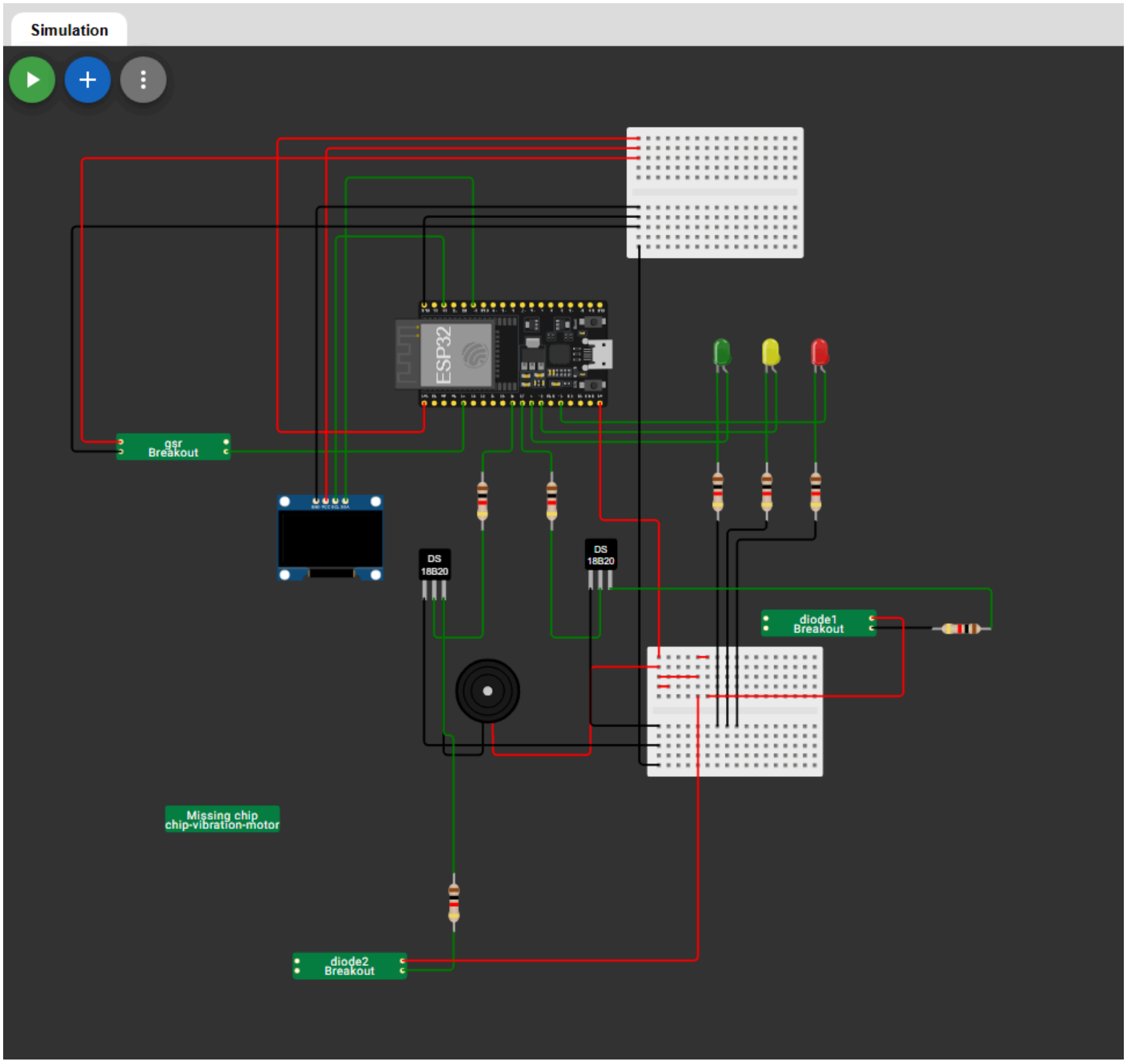
ICE-3203
MICROPROCESSOR AND MICROCONTROLLER

SCHEMATIC DIAGRAM



ICE-3203
MICROPROCESSOR AND MICROCONTROLLER

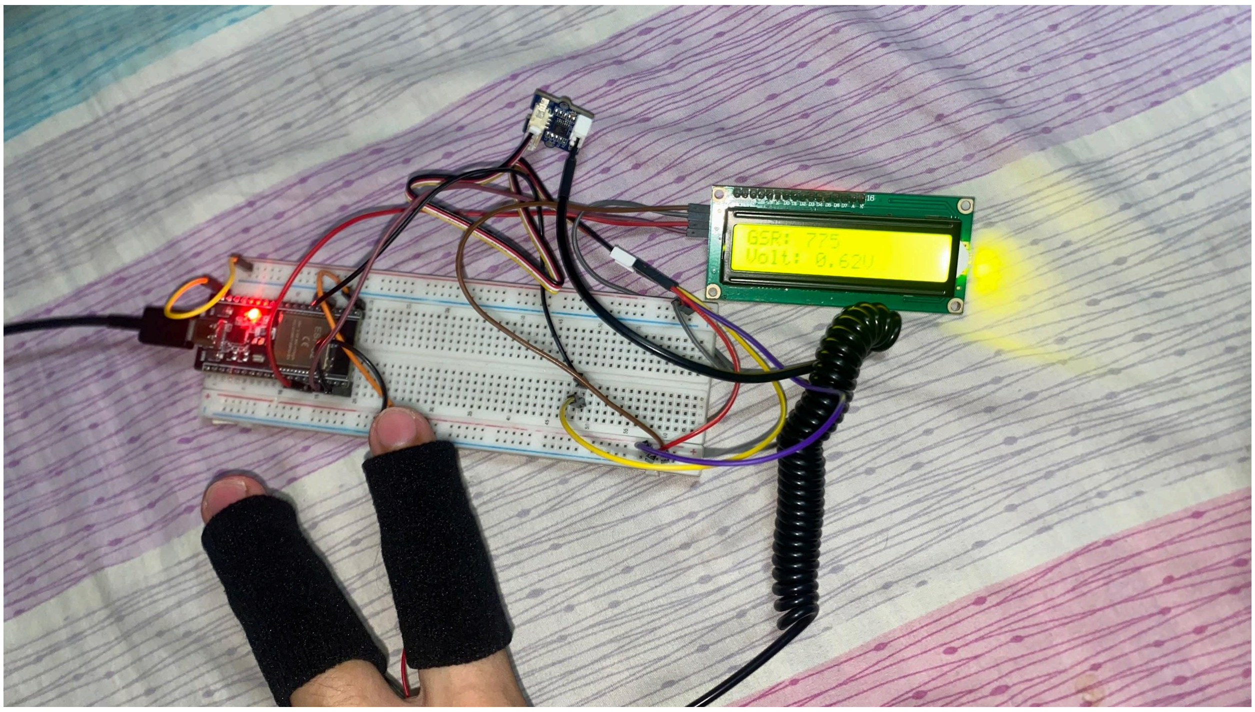
WOKWI SIMULATION



WOKWI PROJECT LINK: https://wokwi.com/projects/431751400831704065

ICE-3203
MICROPROCESSOR AND MICROCONTROLLER

TESTING



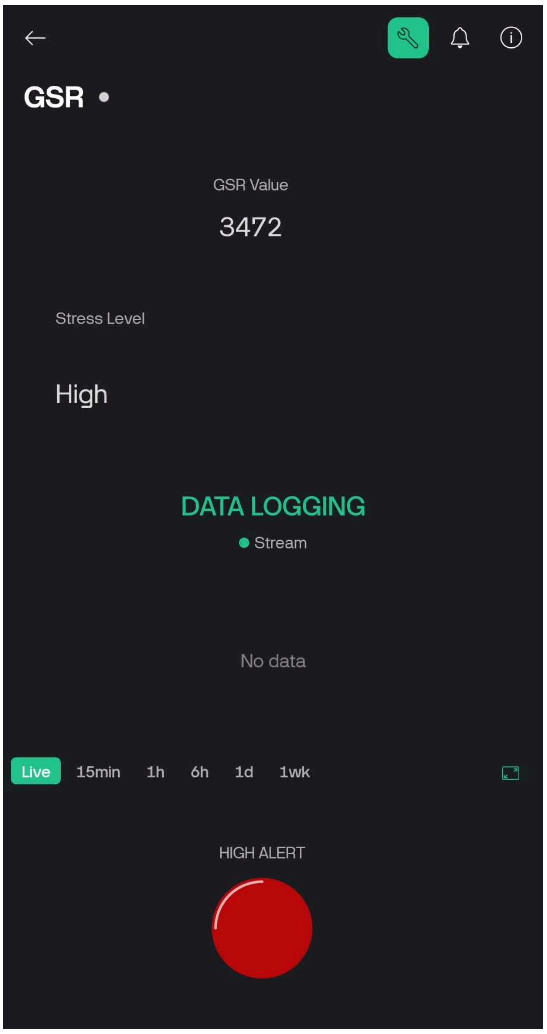
Hardwire Testing

Components:

- ESP32
- GSR SENSOR
- BREADBOARD
- JUMPER WIRES
- LCD(16x2)

In this testing, the proponents show how the GSR Sensor works. GSR detects changes in the electrical conductance of the skin, which is related to sweat gland activity and controlled by the autonomic nervous system.

BLYNK Interface



This section illustrates the Blynk IoT interface enables real-time stress monitoring using **Galvanic Skin Response (GSR) readings**, providing instant feedback and **high-alert notifications** for elevated stress levels. It features a **data logging function** that tracks fluctuations over various time frames, helping users identify triggers, analyze trends, and make informed stress management decisions through an intuitive dashboard.

CONCLUSIONS AND RECOMMENDATIONS

Based on the development and testing conducted by the proponents, the GSR-based stress monitoring system successfully achieved its primary objective of detecting and classifying stress levels in real time. By utilizing a Galvanic Skin Response sensor in conjunction with an ESP32 microcontroller, the system was able to effectively read physiological data, convert it into meaningful voltage readings, and categorize the results into Low, Medium, or High stress levels. The integration of an OLED display, visual and haptic alerts, and the Blynk IoT platform enhanced the system's functionality by providing immediate local feedback and enabling remote monitoring capabilities.

The study demonstrated that physiological signals such as skin conductance can be effectively used as indicators of stress, and with the proper hardware and software configuration, these signals can be analyzed and visualized in real time. The successful transmission of data to the Blynk platform and the triggering of cloud-based notifications proved the system's capability to serve in real-world stress monitoring applications. Overall, the project showcases a cost-effective and accessible solution for stress tracking that can be useful in academic, healthcare, and personal wellness contexts.

In light of the results and limitations encountered during the development process, the proponents offer the following recommendations for future improvements and research:

1. **Sensor Calibration**: Future iterations should include dynamic calibration methods to account for individual differences in skin conductivity and environmental factors that may affect sensor accuracy.
2. **Data Logging and Analysis**: Incorporating onboard or cloud-based data logging would allow for long-term tracking and analysis of stress trends, which could be useful for researchers and healthcare professionals.
3. **Mobile Application Integration**: A custom mobile application with enhanced visualization tools and user interaction features could improve the system's accessibility and user experience.
4. **Multi-Sensor Fusion**: Integrating additional biosensors such as heart rate, body temperature, or ECG could provide a more comprehensive analysis of stress and improve the accuracy of classification.
5. **Enclosure and Portability**: Designing a compact, wearable enclosure for the device would make it more practical for everyday use and field testing.
6. **Machine Learning Implementation**: Future research may consider the use of machine learning algorithms for more adaptive and personalized stress detection based on real-time and historical data.

By addressing these areas, the system can be further enhanced to meet the standards of advanced physiological monitoring tools, thus broadening its applicability in various fields such as mental health, education, workplace wellness, and sports science.