NEED TO DO[[1]](#footnote-1)

Tool for measuring welfare parameters(May 2025)

Mika Gielkens, *Student, UHasselt & KU Leuven*Dr. Eva Geurts (Promotor), Prof. Dr. Gustavo Rovelo Ruiz (Co-promotor)

*Abstract*—

*Index Terms*—

# INTRODUCTION (NEED TO IMPROVE)

I

n today's workplace, employee well-being is becoming a significant part of productivity and job satisfaction. Stress is one of the leading causes of work discomfort and can have negative impacts on physical and mental health. Therefore, the ability to measure and monitor stress-related physiological parameters effectively plays a critical role in creating a healthier workplace.

This thesis focuses on the development of a portable and miniaturized stress measurement device using physiological sensors interfaced with an Arduino Nano ESP32. With the inclusion of a MAX30102 optical heart rate sensor and a Grove - GSR Sensor for skin conductivity measurement, the device can capture real-time biometric readings of stress levels. Such physiological signals provide valuable feedback on an individual's stress response and allow for constant monitoring of well-being during work activity.

The aim of this project is to create and implement a system that not only collects data but also presents it in a useful format through a dedicated application. The application will enable users to real-time heart rate and and skin conductance data. The hardware and software in this project are designed to provide a simple tool for measuring workplace well-being.

To achieve this, the research first explored relevant sensor technologies used to measure stress. This is followed by developing a system architecture that would ensure seamless communication between the sensors and Arduino and data processing and visualization in the app.

By providing a low-cost and user-friendly solution to stress monitoring, the project aims to be a contributor to the new wave of workplace well-being technology and a foundation for more data-based solutions for managing stress.

# Related work (NEED TO DO)

## TEMPLATE

# Methods and materials (STARTED BUT NOT FINISHED)

## This section describes the hardware configuration and software architecture of the wearable stress measurement system. The system is built around an ESP32 microcontroller and incorporates multiple physiological sensors to capture heart rate, blood oxygen saturation (SpO₂), galvanic skin response (GSR), and respiratory activity. The sensor data is processed locally and transmitted via Bluetooth Low Energy (BLE) to a mobile application for visualization and analysis.

## Arduino Nano ESP32

The stress measuring device is powered by an Arduino Nano ESP32 which is based on the ESP32-S3 System on Chip. The main reason for using this microcontroller instead of other Arduino microcontrollers is the presence of Bluetooth and Wi-Fi. In this project, Bluetooth is used to wirelessly transmit the sensor data to a smartphone that runs software to present and visualize the stress level and the measurements. Other criteria for choosing this microcontroller include:

* Compact size
* Low power consumption
* Dual-core processor for real-time processing of sensor readings
* I2C interface, used by the MAX30102 and MAX30205 sensors for communication

Its compact size and energy efficiency make it ideal to use in a portable device.

## MAX30102

The MAX30102 is an optical biosensor that measures the subject’s heart rate and peripheral blood oxygen saturation (SpO₂). It uses the I2C interface for communication with the microcontroller. Due to its compact design and low-power requirement, the MAX30102 sensor is optimal to use in wearable/mobile devices.

The sensor includes LEDs, photodetectors and low-noise electronics capable of ambient light rejection. Although it contains ambient light rejection, it is still advised to limit the ambient light interference to ensure measurement accuracy. The sensor uses both a red and an infrared (IR) LED to extract heart rate and SpO₂ via signal processing.

The sensor is initialized with a sampling rate of 100 Hz and is read in sequences of 100 samples to calculate the heart rate and SpO2 using an open source algorithm.

## Grove GSR Sensor

The Galvanic Skin Response (GSR) sensor measures the skin conductance between two electrodes, which varies with sweat activity. Since skin conductance is linked to autonomic nervous system activity, it serves as a reliable indicator of stress.

It works by placing two electrodes on the skin (usually fingers) and applying a small amount of voltage. The resulting conductance is measured, with higher values indicating increased sweat gland activity. This can be the result of elevated stress levels.

The sensor outputs an analog voltage which is read using the pin A2 on the ESP32. An average over 40 samples is calculated to lessen the impact of fluctuations.

## Plux Piezo-Electric Respiration (PZT) Sensor

The Plux PZT Sensor is a sensor that is capable of gathering basic respiration data. It includes a chest-belt with a built-in piezoelectric sensing element (piezoelectric film) that gathers data by measuring respiratory motion. The sensor detects mechanical deformation of the sensor caused by changes in thoracic or abdominal volume when breathing.

The signal is read using the analog pin A0. A peak detection algorithm is used over a 60-second window to count the inhalations made within a minute. This results in a value for breaths per minute (BPM).

(<https://www.pluxbiosignals.com/products/respiration-pzt?srsltid=AfmBOoqX-ZqraYTy9DBPU243tGuNZ-K0sgG5AEa1SrWpJkrOwiXdz8Sm>)

|  |  |  |  |
| --- | --- | --- | --- |
| **Component** | **Function** | **Connection** | **Pin(s) used** |
| MAX30102 | Heart rate and SpO₂ via PPG | I²C | SDA/SCL (default Wire) |
| Grove - GSR Sensor | Galvanic skin response (skin conductance) | Analog | A2 |
| Plux PZT Sensor | Respiratory effort (chest movement) | Analog | A0 |

*Table 1: Summary of sensor functions, interfaces, and microcontroller pins used*

## Software Architecture

The firmware is written in C++ using the Arduino framework. The software handles sensor initialization, data acquisition, BLE communication, and command handling.

For BLE communication a custom BLE GATT service is implemented using the following UUIDs:

* **Service UUID**: 180C
* **Data Characteristic (UUID: 2A6E)**: Used to transmit sensor data (HR, SpO₂, GSR, and respiration).
* **Command Characteristic (UUID: 2A6F)**: Receives control commands from the mobile application.

The ESP32 is programmed to be able to support following commands:

* **START:** Begin continuous measurement from all sensors.
* **START HEART, START SPO2, START GSR, START BREATHING:** Begin measurement from a specific sensor.
* **STOP:** End all measurements and reset system flags.

Each sensor is handled by a dedicated function that enables sensor data acquisition. In each measurement loop, following steps are executed:

* **MAX30102**: Samples 100 values each from red and IR channels, processed by maxim\_heart\_rate\_and\_oxygen\_saturation().
* **GSR**: Samples the analog input 40 times with a short delay between reads and calculates the average value.
* **PZT (Breathing)**: Over a 60-second window, the analog signal is sampled at 20 Hz. A basic threshold-based peak detection method is used to count breaths.

To avoid conflicts, the sensor readings are performed sequentially, particularly for analog inputs. A delay of 100 ms is introduced in the main loop to ensure system responsiveness and allow BLE event handling.

After each data acquisition, the data from each sensor is combined into a single string (e.g., "HR:75 SpO2:97 GSR:523 Breathing:15") which is sent to the BLE client via a notification.

## Application

This application presents an integrated platform for the real-time and retrospective monitoring of four key physiological parameters: heart rate, galvanic skin response (GSR), blood oxygen saturation (SpO₂), and respiration rate. The design emphasizes usability and interpretability, employing tailored data visualizations for each parameter to facilitate user comprehension regardless of technical expertise.

An initial functionality introduced was a user account system, enabling organizations to deploy the monitoring tool without procuring an individual device for each employee. By allowing multiple users to authenticate on a shared device, this design minimizes capital expenditure and operational overhead. Consequently, organizations can allocate a single hardware unit to a cohort of users, who may access it whenever they need to assess their physiological parameters. This shared‐device paradigm not only reduces per‐user costs but also enhances the overall accessibility of the system, thereby facilitating broader adoption within resource‐constrained settings.

During the initial configuration process, first-time users are prompted to enter their age, which is used solely to estimate their maximum heart rate. This value is calculated using established physiological formulas commonly employed in health research (e.g., 220 minus age), and forms the basis for defining heart rate zones displayed in the application's radial gauge visualization. These zones are color-coded to distinguish between resting, moderate, and elevated exertion levels, enabling intuitive interpretation of cardiovascular status. Additionally, the setup process includes a brief baseline measurement phase, during which the user's resting respiration rate, average heart rate, and average GSR are recorded. These baseline values are critical for personalizing the application’s stress level detection, as both resting respiration and GSR serve as inputs to the stress estimation algorithm, allowing the system to adapt to individual physiological norms.

Heart rate, respiration rate and GSR are visualized through radial gauges, offering immediate, intuitive feedback on current values in relation to established healthy ranges. The heart rate gauge incorporates color gradients to present normal, cautionary, and critical thresholds, thereby enabling users to assess their physiological status in an instance. SpO₂ levels are conveyed via a progress bar with dynamic color coding, providing instant recognition of safe versus concerning levels. These visualization choices are informed by established principles in usability research, particularly the efficacy of graphical representations in enhancing comprehension and anomaly detection in health monitoring contexts.

A central feature of the application is the stress indicator, which integrates data from GSR and respiration rate to estimate the user’s current stress level. Research shows that GSR and respiration rate are valuable for stress detection, but does not establish specific threshold values for stress states. The focus of this research is the usability and not the algorithms or best values. For this reason, a “dummy” value for the critical GSR and respiratory rate is used. This approach enables the evaluation of interface usability independent of clinical validation. The resulting composite measure is communicated using a straightforward color-coded scheme—green, orange, or red—supplemented by a dialog box offering interpretive guidance. This approach ensures that feedback remains both transparent and actionable, supporting informed user response.

To facilitate longitudinal health tracking, the application incorporates a history section for each monitored parameter. This page is easily accessed by a button below each parameter. Users may toggle between graphical and table-based data presentations and select predefined time-based filters (e.g., “Today,” “Last 7 Days,” “Last 30 Days”) to adjust the scope of displayed data. These filters are implemented buttons positioned at the top of the interface, a design decision intended to preserve vertical screen space and maintain interface clarity on mobile devices. This configuration supports rapid, low-effort navigation and minimizes cognitive load, thereby enhancing overall usability. The application adopts a data visualization approach similar to Samsung Health for representing weekly and monthly trends, specifically by using bar graphs that display the minimum and maximum values for each day. This method provides a clear and compact summary of daily fluctuations without overwhelming the user with excessive detail. By showing a daily range rather than individual data points, the graph highlights variability and outliers while maintaining visual simplicity—especially important when multiple days are displayed on limited screen space. This technique facilitates quick comparisons across days, helping users identify patterns such as consistently elevated stress or irregular physiological responses. It is particularly effective for mobile interfaces, where space constraints demand concise yet meaningful visualizations. Moreover, using bars rather than line charts in this context avoids visual clutter and promotes interpretability, aligning with established best practices in mobile health data design. Inspired by Samsung Health’s proven interface, this approach balances information density with user comprehension, supporting both daily check-ins and longer-term trend analysis.

To further enhance user comprehension and create trust in the system, the application includes a dedicated information panel for each health metric. This panel can be accessed via an info button on each parameter’s screen and provides clear, concise explanations of what each physiological metric represents (e.g., heart rate, GSR, SpO₂, respiration rate), why it is important for health, and how the measurement is obtained by the device.

The information panel serves two purposes. First, it empowers users to understand the significance of their readings. This transparency clarifies health data, making the app more accessible and reducing confusion around unfamiliar terms. Second, by explaining the sensor methods and the physiological basis for each metric, the panel helps users understand the reliability and limitations of the measurements. Together, these features help prevent the “blackboxing” that remains a common issue in current health monitoring applications.

The initial user onboarding process is designed to personalize the application experience. During first-time setup, users are prompted to specify individual baseline values for heart rate, GSR and respiratory rate. This calibration step ensures that subsequent alerts and visual feedback are tailored to the user’s unique physiological profile, thereby enhancing the accuracy and personal relevance of the system’s insights.

Through the combination of adaptive visualizations, personalized feedback mechanisms, and user-centered interface design, this application aims to make physiological health monitoring both accessible and meaningful. The integration of intuitive interaction patterns with evidence-based visualization strategies supports a broad spectrum of user needs, from casual self-assessment to more systematic health tracking.

## Usability principles

<https://www.nngroup.com/articles/ten-usability-heuristics/>

User-Centricity: The design is centered around the needs and abilities of your target users, employing familiar language, clear icons, and intuitive navigation so users can interact with the app regardless of their technical background.

Consistency and Standards: Visual elements such as gauges, progress bars, and color codes are used consistently throughout the app. This adherence to common design patterns and conventions helps users quickly learn how to use different features and reduces confusion.

Visibility of System Status: Real-time feedback is provided through dynamic gauges and progress bars, ensuring users are always aware of the current state of their health parameters and the system itself.

Simplicity and Minimalism: The interface is intentionally uncluttered, showing only essential information and controls. This reduces cognitive load and helps users focus on key tasks, such as measuring or interpreting their health data.

Match Between System and Real World: The app uses real-world metaphors (e.g., gauges resembling analog dials) and everyday language, making it easier for users to understand the meaning of their data and actions.

Error Prevention and Recovery: The design includes clear feedback for actions like device pairing, and uses constraints and confirmation dialogs to minimize user errors and guide them in resolving issues if they occur.

Flexibility and Efficiency: Features like customizable thresholds and filters allow users to personalize their experience and quickly access relevant data, supporting both novice and experienced users.

Ease of Learning: The onboarding process guides first-time users step-by-step, ensuring that even those unfamiliar with health monitoring technology can set up and use the app effectively

# User study (NEED TO DO)

TEMPLATE

# Result

TEMPLATE

# Discussion (NEED TO DO)

TEMPLATE

# Conclusion (NEED TO DO)

TEMPLATE

# Acknowledgments

TEMPLATE

References

1. TEMPLATE

1. [↑](#footnote-ref-1)