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 explores the development of a small and portable device for monitoring a person's well-being while performing work activity. There are commercial alternatives like smartwatches, but they are expensive, not widely accessible, or not transparent in their operation. An inexpensive and versatile solution is therefore necessary. This project entails the construction of a well-being toolkit based on Arduino technology, with emphasis on determining the physiological parameters best suited to monitor stress or general well-being, including heart rate and skin conductance.

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 monitor their well-being real-time by measuring relevant physiological parameters such as heart rate and skin conductance. The hardware and software in this project are designed to provide a user friendly 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 ADD)

## TEMPLATE

# Methods and materials

## 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. A consolidated overview of all sensors used in this system and their integration with the Arduino Nano ESP32 is presented in table 1.

## MAX30102: heart rate and oxygen saturation sensor

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: skin conductance 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: respiration rate

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 in a time frame of 500ms with a short delay in between reads, each from red and IR channels, processed by maxim\_heart\_rate\_and\_oxygen\_saturation().
* **GSR**: Samples the analog input 40 times over 200ms 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 software offers a unified platform for the simultaneous monitoring of four key physiological parameters: heart rate, galvanic skin response (GSR), blood oxygen saturation (SpO₂), and respiration rate, in real-time and retrospectively. The program is designed with an emphasis on usability and interpretability, employing customized data visualizations for each parameter to be easily understood by users independent of technical background.

A key functionality that was introduced is a user account system, which allows organizations to deploy the monitoring tool without needing to purchase an individual device for each employee. The design allows multiple users to log in on a shared device, thus saving both capital expenditures and operating costs. Consequently, organizations can assign one hardware unit to be used by a group of users, who can use it at their convenience to assess their physiological measurements. The shared-device model not only reduces the cost per user but also enhances the overall accessibility of the system, thereby enhancing broader adoption in resource-scarce settings.

As part of the initial setup process, new users are prompted to enter their age, which is used for the purpose of estimating maximum heart rate. It is determined using standard physiological formulas that are commonly applied in health research (e.g., 208-0.7\*age), and serves as the basis for the calculation of heart rate zones represented in the application's radial gauge visualization. These areas are color-coded to distinguish between resting, moderate, and increased levels of exertion, permitting intuitive evaluation of cardiovascular status.

Further, the configuration process includes a brief baseline measurement period during which the user's resting respiratory rate, mean heart rate, and mean GSR are recorded. These baselines are necessary to individualize the application's detection of stress level, as both resting respiration and GSR are inputs to the stress estimation algorithm, which allows the system to adapt to the physiological norms of each user.

Heart rate, respiration rate and GSR are shown using radial gauges, offering immediate, intuitive indication of current value versus set healthy ranges. A color gradient across normal, warning, and critical levels is implemented for the heart rate gauge, enabling users to assess their physiological state at a glance. SpO₂ levels are shown by a progress bar with dynamic color coding, enabling instant recognition of safe from problem levels

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 enable longitudinal health tracking, the app features a history page for each parameter tracked. The screen is easily accessible by a button below each parameter. The user can toggle between graphical and tabular data presentation formats and select pre-established time-based filters ("Today," "Last 7 Days," "Last 30 Days") to modify the data range displayed. They are introduced as buttons positioned at the top of the interface, a design decision intended to preserve vertical screen space and interface legibility on handheld devices. The layout supports fast, low-effort navigation and minimizes cognitive load, thereby enhancing overall usability. The application follows the same data visualization strategy as Samsung Health for displaying monthly and weekly trends, namely by utilizing bar graphs to represent the minimum and maximum value for each day. This is a method of providing a concise and easy-to-understand overview of daily variations without bombarding the user with excessive information. The graph emphasizes variability and outliers by plotting a day-to-day range instead of specific data points, thereby preserving visual clarity, which is necessary in case numerous days are plotted within small screen space. The approach enables users to compare easily between days and to spot trends. It is especially useful for mobile devices, where spatial constraints imply short but significant visual representations are required. Moreover, the utilization of bars over line graphs here avoids visual clutter and promotes interpretability, in accordance with proven best practices in mobile health data design. Taking a cue from the effective interface of Samsung Health, this approach balances information density with user comprehension, supporting both day-to-day check-ins as well as 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 allows users to understand the significance of their readings. Such transparency explains health information, making the app more accessible and reducing confusion over unknown terms. Second, by explaining the sensor methods and the physiological basis for each measurement, the panel helps users understand the accuracy and limitations of the measurements. Together, these features help to reduce the "blackboxing" effect that remains a common issue among health monitoring apps today.

By coupling adaptive visualizations, personalized feedback mechanisms, and an interface design centered on the user, this application endeavors to render physiological health monitoring both possible and understandable. The pairing of natural interaction mechanisms with visualization techniques aims to facilitate a wide range of user requirements, from casual self-evaluation to more structured health monitoring.

## Usability principles used

<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)