STRESS IDENTIFICATION

Project Report

Introduction to Internet of Things

Spring 2024 BSCS 8A

Group Members:

Muneeb Ahmad	20L-1173
Faiqa Adnan	20L-1116
Aisha Muhammad Nawaz	20L-0921

National University Of Computer and Emerging Sciences
Department of Computer Science
Lahore, Pakistan

Table of Contents

1. Introduction

Stress has become a pervasive issue in our fast-paced society, affecting people of all ages and occupations. The ramifications of stress extend far beyond its immediate effects. While someone may appear physically fine, the toll on their mental health can be significant. Stress has been linked to a myriad of health conditions, including cardiovascular diseases, diabetes, and more. Moreover, the importance of mental health cannot be overstated; stress can exacerbate existing conditions and contribute to a decline in overall well-being. Despite being aware of the pressures we face, many overlook the importance of monitoring their own health, leading to potentially serious consequences, including fatalities.

To address this urgent need, we propose a solution leveraging wireless technology for stress detection and health monitoring. By utilizing sensors to track vital signs like heart rate and temperature, our system aims to provide real-time insights to individuals and healthcare professionals alike.

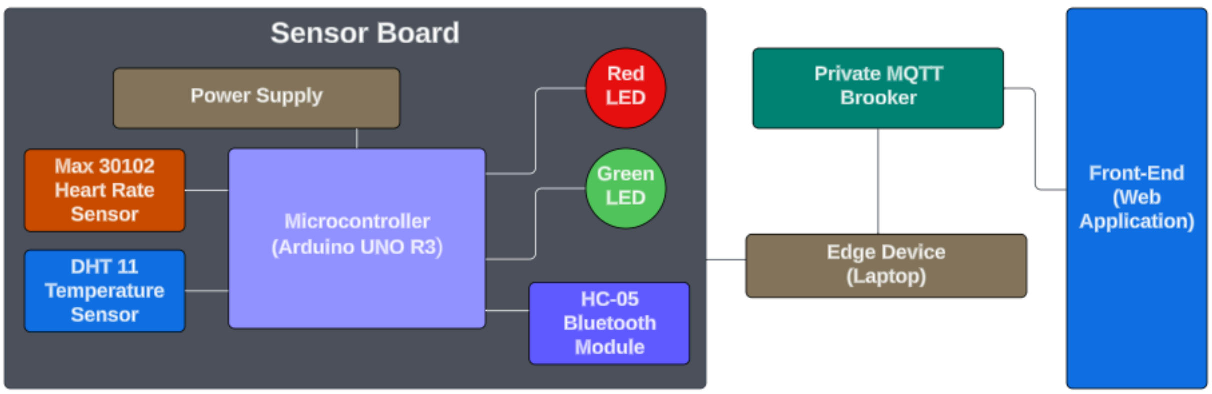
This report outlines our project's details, emphasizing its simplicity, efficiency, and potential to make a tangible impact on health management. Together, let's pave the way for a future where stress detection is accessible and actionable for everyone.

2. Project Aims

1. **Enhance Individual Well-being:** Develop a stress detection system that empowers users to monitor and manage their stress levels in real-time, utilizing seamlessly integrated sensors such as the Heart Rate Sensor and Temperature Sensor to promote proactive self-care and foster improved mental and physical well-being.
2. **Facilitate Medical Intervention:** Enable medical staff to remotely access comprehensive stress data through our system's wireless communication capabilities. By integrating user-friendly interfaces and reliable sensor technologies, our project ensures efficient transmission and interpretation of stress metrics, facilitating early intervention and personalized care strategies for individuals, thereby enhancing healthcare outcomes and patient satisfaction.
3. **Sensor Integration and Calibration:** Ensure seamless integration and calibration of the Heart Rate Sensor and Temperature Sensor to accurately measure heart rate and temperature, respectively.
4. **Signal Processing and Filtering:** Develop algorithms for signal processing and filtering to remove noise and extract relevant data from sensor readings, enhancing the accuracy of stress identification.
5. **Microcontroller Compatibility:** Ensure compatibility and efficient communication between the sensors and the microcontroller (Arduino Uno R3), facilitating data processing and analysis.
6. **Actuator Integration:** Integrate actuators such as Red and Green LED for visual feedback, providing users with real-time notifications or alerts regarding their stress levels.
7. **Wireless Communication:** Implement a reliable wireless communication module (e.g., HC-05 Bluetooth Module) to transmit sensor data wirelessly to edge devices for further analysis.

8. **Edge Device Setup:** Set up edge devices to receive sensor data and process it using MQTT client software, enabling seamless communication with the MQTT broker.
9. **MQTT Broker Configuration:** Configure the MQTT broker to facilitate communication between edge devices and the front-end user interface, ensuring efficient data transmission and management.
10. **Front-end Development:** Develop a user-friendly Web Application for monitoring and controlling the IoT system, allowing users to visualize stress data and receive insights in real-time.

3. Block Diagram of Sensor Board



4. Hardware Used

1. **Arduino UNO R3 Development Board ATmega328P CH340 CH340G**
   The Arduino UNO R3 Development Board, powered by the ATmega328P microcontroller, served as the central control unit for this project, interfacing with sensors, actuators, and peripherals to execute the programmed tasks.
2. **Max 30102 Heart Rate Sensor**
   The MAX30102 Heart Rate Sensor is employed to detect variations in blood volume through photoplethysmography, enabling the calculation of heart rate based on the pulsatile signal generated by the user's cardiovascular system.
3. **DHT 11 Temperature Sensor**
   In this project, the DHT11 temperature sensor is employed to gather real-time temperature data, mirroring the methodology outlined in research paper [2].
4. **Red and Green LED**
   The red LED signifies "Stress" while the green LED indicates "No Stress" within the project, providing visual feedback based on analyzed data.
5. **HC-05 Bluetooth Module**
   The Bluetooth module facilitated wireless data reception by the laptop from external sources, enabling subsequent transmission to the MQTT broker.

6. **Edge Device (Laptop)**
   The laptop, acting as the edge device, receives data via the Bluetooth module, then forwards it to the MQTT broker (broker.emqx.io). The front end, edge device, laptop is connected to the same MQTT broker, subscribes to the relevant topic, and displays the received data on the web application.
7. **Two 220 Ohm Resistors**
   Two 220 Ohm resistors are employed alongside the red and green LEDs to ensure proper current limitation and prevent overloading, thereby safeguarding the LEDs from potential damage.
8. **Breadboard**
   The breadboard was used to quickly try out different electronic connections for the project, making it easier to test and improve our circuit designs.

5. MQTT Topics

- StressDet092111161173

6. Working Demo of Hardware and Software Components

The working demo video is available at the following link:
https://drive.google.com/file/d/1P5Y-DU4iy5JbvUF4kLqOrjfoepVVix1e/view?usp=sharing

7. Methodology

We followed a methodology akin to that described in reference [1]. The Stress Detection system mainly consists of the following components: Microcontroller, Heart Rate Sensor, Temperature Sensor, Two LEDs and a Bluetooth Module. The Heart Rate and Temperature Temperature sensor collects data of beats per minute (bpm) and temperature. The sensors measure the heart rate and temperature, the arduino processes this information and the bluetooth module transmits it wirelessly to the edge device, laptop, from where the data is published to MQTT Broker. The front end edge device, laptop, then receives data from MQTT Broker and displays it on the web application. For the sensors to record the data, the user is required to place their finger on the sensor for a few seconds so that the readings for temperature and bpm are recorded properly and continuously to detect stress. The MAX30102 heart rate sensor detects infrared light reflected from the skin's surface, which is then captured by a photodiode. This sensor measures changes in blood volume based on the detected infrared signals. It calculates the heart rate and updates the beat average. Furthermore, DHT11 temperature sensors read temperature at regular intervals. Afterwards, the recorded values undergo a stress assessment process, comparing them against predetermined thresholds for heart rate and temperature. The criterion employed to identify stress is as follows: If the temperature is below 33 degrees Celsius and the heart rate exceeds 100 beats per minute, the system flags the condition as "STRESS"; otherwise, it categorizes it as "NO STRESS". In case stress is detected, the system activates a red LED, while a green LED is activated when no stress is detected.

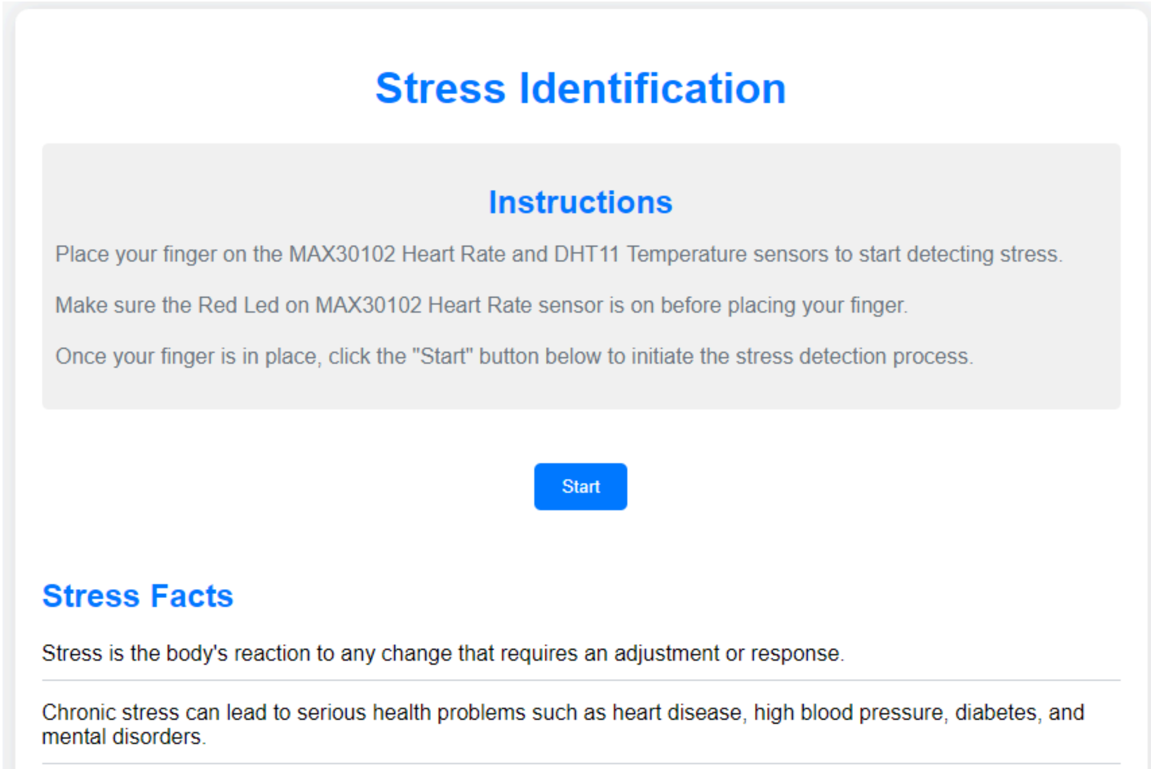
The data is transmitted over Bluetooth to the edge device. A python program is written to receive data from the Bluetooth module, parse the received data and publish it to an MQTT broker. A MQTT client instance is created with a client ID and set to start a clean session. The client connects to the MQTT broker and the MQTT loop is started. The received data is parsed and the values are extracted and the sensor message is printed. A check has also been implemented to make sure that only valid readings are published to the broker. The message is published to the MQTT broker on the topic “StressDet092111161173”.

For the front-end, a Python code utilizes Flask, a micro web framework for Python, along with the MQTT client library for communication with an MQTT broker. It establishes a connection with an MQTT broker hosted at "broker.emqx.io" on port 1883 and subscribes to a specific topic for receiving stress detection data. Upon receiving MQTT messages, the code parses the data and updates a list called "readings" with the received information. The Flask application renders an HTML template ('index.html') and passes the readings list to it for display. The application runs on port 3333 with debugging enabled. Additionally, it starts the MQTT client loop to continuously handle incoming messages.

The HTML code defines a web interface for stress identification. Users are instructed to place their finger on specific sensors before initiating stress detection by clicking the "Start" button. The system then fetches data from an MQTT broker, populates the readings in a list format, and displays them dynamically on the webpage. Stress levels are indicated by color-coded entries, and additional information on stress management is provided. The system is designed to be user-friendly, informative, and based on real-time physiological signals, enhancing awareness and understanding of stress-related issues.

8. User Guide for Front-End

This user guide offers concise instructions to effectively utilize the stress detection system, guiding users through setup, data acquisition, and interpretation for optimal stress monitoring.



1. Getting Started:

- **Hardware Setup:** Ensure that you have the MAX30102 Heart Rate sensor and DHT11 Temperature sensor properly connected to your device. Make sure that the Red LED on the MAX30102 Heart Rate sensor is on before proceeding.

2. Instructions:

- **Place Your Finger:** Position your finger on the MAX30102 Heart Rate and DHT11 Temperature sensors. This is essential for accurate stress detection.
- **Start Button:** Once your finger is in place, click the "Start" button below to initiate the stress detection process.

3. Stress Detection Process:

- **Loading Screen:** Upon clicking the "Start" button, a loading screen will appear indicating that the system is detecting your stress levels.
- **Readings Display:** After a brief moment, the stress detection results will be displayed on the screen. You'll see a list of readings including Heart Rate, Temperature, and Stress Status.

4. Interpreting the Results:

- **Stress Status:** Each reading will be categorized as either indicating stress or no stress. Stress readings will be highlighted in red, while no stress readings will be highlighted in green.

5. Restarting the Process:

- **Restart Button:** If you wish to restart the stress detection process, simply click the "Restart" button. This will clear previous readings and allow you to begin anew.

6. Stress Facts:

- **Additional Information:** Below the stress detection results, you'll find a section titled "Stress Facts." This section provides valuable information about stress, its effects on health, and methods for managing it effectively.

7. Source Information:

- **Research Reference:** At the bottom of the stress detection results, you'll find information about the research paper that served as the basis for the stress detection method implemented in this system.

8. Note:

- **Accuracy:** While this system provides valuable insights into your stress levels, it is important to note that it may not be as accurate as professional medical equipment. Use the information provided as a guide and consult with a healthcare professional if needed.

9. Bibliography

[1] Hendryani, Atika & Gunawan, Dadag & Rizkinia, Mia & Hidayati, Rinda & Hermawan, Frisa. (2023). Real-time stress detection and monitoring system using IoT-based physiological signals. Bulletin of Electrical Engineering and Informatics. 12. 2807-2815. 10.11591/eei.v12i5.5132.

[2] D. Thomas, V. L. Vineeth, P. G. Siddharth, and M. Shanmugasundaram, "IoT based mobile health hub," in Proc. IOP Conf. Ser.: Mater. Sci. Eng., vol. 263, no. 5, 2017, paper no. 052048. DOI: 10.1088/1757-899X/263/5/052048.

9. Group Member Contributions

1) **Hardware Setup & Connectivity, Edge Device Codes, Documentation** - L200921 (Aisha M Nawaz)
2) **Stress Detection Arduino Code, Documentation -** L201116 (Faiqa Adnan)
3) **Front-End Code, Documentation -** L201173 (Muneeb Ahmad)