ICT Innovation - PD&D
Technical Report

Student: Pietro Ventrucci

Team number: 1 - EISI

Contents

1 Introduction

EISI is a company committed to improving first aid interventions, particularly for patients with severe illnesses or conditions that require constant medication and treatment. Such patients may be unable to communicate essential medical information in emergencies, for example if they are unconscious or in shock. To address this issue, EISI is developing an application and service that can store patients' medical data, which will be accessible to healthcare workers on site. Their proposed solution involves retrieving this information either via Bluetooth, if the patient's phone is nearby, or by taking a photo of the patient's face to identify them through facial recognition linked to their medical profile; both actions could be performed with a smartphone.

In response to this challenge, our team explored how to enhance further or revolutionize the proposed solution. We designed a smartwatch that supports and extends the smartphone-based system, enabling first responders to capture a photo directly from the wrist device and display critical medical information without needing to handle any phone. By integrating AI capabilities, the smartwatch could in the future also provide decision support functions to optimize emergency response.

This report presents the development of an initial prototype implementing essential, though not exhaustive, features of the proposed smartwatch. The purpose is to demonstrate its feasibility, practicality, and potential impact in improving emergency medical interventions.

2 Problem Definition & Motivation

As is easy to guess, in emergency care, every second matters. One of the biggest problems faced by first responders is the delay in detecting, identifying, and managing unconscious patients at the scene. Currently, responders handle multiple devices, such as tablets, radios, and phones, and they often need to manually enter patient data into apps like MedicalNote [**?**]. This workflow increases the risk of delayed or ineffective interventions. Furthermore, these devices can easily be dropped, forgotten, or become difficult to operate under pressure.

To address these critical issues, we collected feedback from the emergency care environment and identified a key opportunity: developing the E-Watch. This device is a purpose-built smartwatch for emergency responders. Worn on the wrist, it combines multiple functions into a single device, keeping responders' hands free and providing real-time data linked with the EISI application.

Importantly, the smartwatch is not intended to completely replace the smartphone-based solution proposed by EISI, but rather to act as a complementary accessory. It expands the system's capabilities by adding options and modularity, allowing for faster and more intuitive patient identification and information retrieval in critical situations.

3 Design Process and Key Decisions

3.1 Requirements Analysis

The development of the E-Watch was guided by both functional and non-functional requirements, although many of these were addressed only partially due to the prototype nature of the current version. The following tables summarize the key requirements and indicate their current implementation status.

Table 1: Functional Requirements

Requirement	**Description**	**Status**
Facial image capture	The device must take a photo of the patient's face.	Implemented (prototype)
Bluetooth transmission to smartphone	The photo must be sent to a smartphone via Bluetooth.	Implemented (prototype)
Server communication for facial recognition	The smartphone should send the photo to a remote server for recognition.	Simulated (prototype app & server)
Display of patient data on the watch	After recognition, medical data should be sent back and shown on the watch.	Implemented (prototype)

Table 2: Non-Functional Requirements

Requirement	**Description**	**Status**
Compact size and light weight	Target: $<$ 100g, smartwatch like form factor.	Partially met (prototype constraints)
Ergonomics	Comfortable and stable on the wrist.	Met
Energy efficiency	Must last a full shift without recharge.	Not yet met (no optimizations)
IP68 protection	Resistant to water and dust.	Not implemented (not essential for prototype)
Time efficiency	Total process time (photo $\rightarrow$ data) $<$ 10 seconds.	Approximated in testing
Safe temperature operation	Device should not overheat during normal use.	Met

3.2 Hardware Design Decisions

The hardware design of the E-Watch aimed to balance compactness, functionality, and prototyping simplicity.

Microcontroller Selection We chose the **XIAO ESP32S3 Sense** microcontroller for its unique combination of features integrated into a small package. Its key features include an OV2640 camera sensor (suitable for basic facial capture), a digital microphone, Bluetooth 5.0 and Wi-Fi connectivity, and a 3.7V battery charging controller. This choice allowed us to significantly reduce the footprint of the prototype, integrating multiple essential functions onto a single board.

Figure 1: XIAO ESP32S3 Sense Microcontroller

Power Supply The entire watch is powered by a 3.7V 290mAh LiPo battery, which theoretically could provide up to 48 hours of autonomy when operating in light-sleep mode (3.8V / 5.47mA).

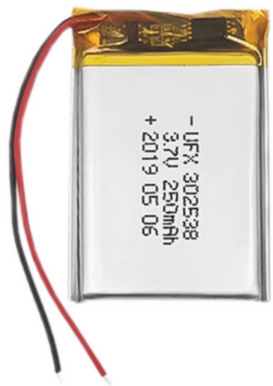

Figure 2: 3.7V 290mAh LiPo Battery

Display For the prototype display, we selected a **1.3-inch OLED screen** (SSD1306, 128x64 pixels). Its advantages include low power consumption, high contrast for easy readability in both indoor and outdoor conditions, and simple integration with the ESP32S3 via I2C. Although limited in resolution, it is sufficient for displaying critical patient data, confirmation messages, and basic status indicators. Future versions will include a colorful, high resolution display.

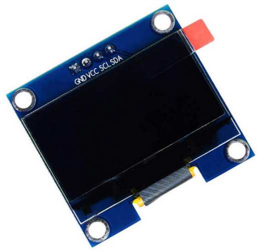

Figure 3: 1.3-inch OLED SSD1306 Display

Other Components The final prototype included:

Table 3: List of Hardware Components

Component	Description
XIAO ESP32S3 Sense	MCU + Camera + Microphone
3.7V 290mAh LiPo Battery	Power supply
White LED	Status indicator
SSD1306 128x64 OLED Display	User interface display
3x Push buttons	Simple input controls; selected instead of a rotary encoder for prototype simplicity

Circuit Diagram The schematic diagram illustrates all connections between the ESP32S3, display, buttons, LED, and battery controller.

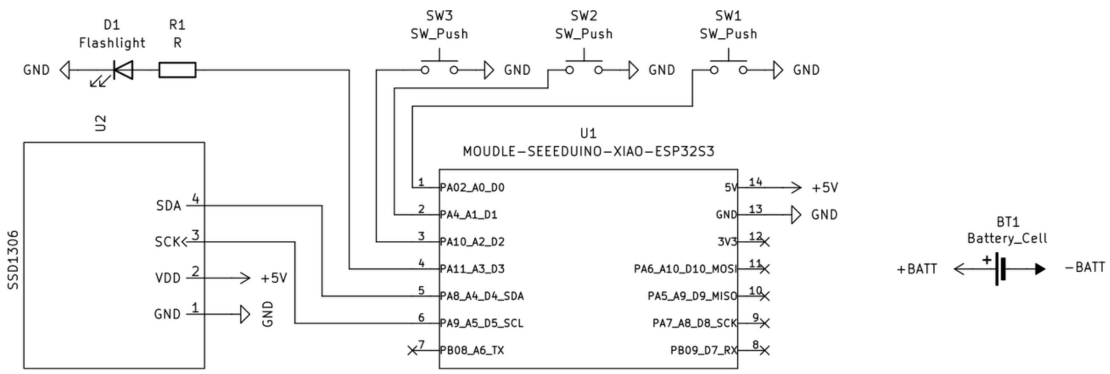

Figure 4: E-Watch Circuit Schematic

3D-Printed Case Design The watch case was designed and printed using PLA filament rather than resin, to ensure faster prototyping cycles, simpler and cheaper manufacturing, and easier modification and assembly. The lower part of the case is secured with screws, allowing quick disassembly to access and modify components during development and testing.

The case design needed to fit the available wristband while comfortably housing all components, ensuring ergonomic wearability. The camera is positioned on the outer wrist side, angled to avoid obstruction by the user's hand during photo capture.

Figure 5: 3D-Printed Watch Case Design.

3.3 Software Design Decisions

The software architecture was designed to enable efficient communication between the smartwatch, smartphone, and server, while remaining modular for future improvements.

Bluetooth Communication The smartwatch communicates with the smartphone via **Bluetooth Low Energy (BLE) 5.0**. The communication flow is as follows:

1. The smartwatch captures a photo using its integrated camera.
2. The photo is sent to the smartphone via BLE.
3. The smartphone forwards the photo to the server through an HTTP POST request.
4. Once the server returns the patient's medical data in the HTTP response, the smartphone transmits this data back to the smartwatch via BLE.

Server Architecture A prototype server was developed in **Python using Flask**, exposing several APIs to handle patient data management and facial recognition simulation.

For the facial recognition functionality, we used the **Python `face_recognition` module**. This choice was motivated by several factors:

- It provides a high-level and easy-to-use interface for facial recognition tasks, enabling rapid prototyping without requiring in-depth machine learning model training or dataset preparation.
- It is based on well-established models such as dlib's facial recognition implementation, ensuring good accuracy in standard conditions.
- It allowed us to focus development time on system integration and hardware-software communication rather than training custom models, which was out of scope for the first proof-of-concept prototype.
- Its compatibility with Python simplified backend development and integration within our Flask server architecture.

The server provides the following endpoints:

Table 4: Server API Endpoints

Endpoint	**Description**
/	Server status check
/dati	Retrieve patient data (only after recognition)
/log	Retrieve recognition logs (anonymous statistics)
/patient-count	Get total number of registered patients
/patients	Deprecated for security reasons
/recognize	Recognize a patient from a photo
/register	Register a new patient with a photo
/session-status/<patient_id>	Check session status for a specific patient
/stats	Aggregate system statistics

Android Application To simulate the smartphone component of the EISI system, an Android application was developed using **Python with Kivy and Buildozer**. The choice to build a native-like app rather than a simpler web interface was made to closely emulate the intended final service proposed by EISI.

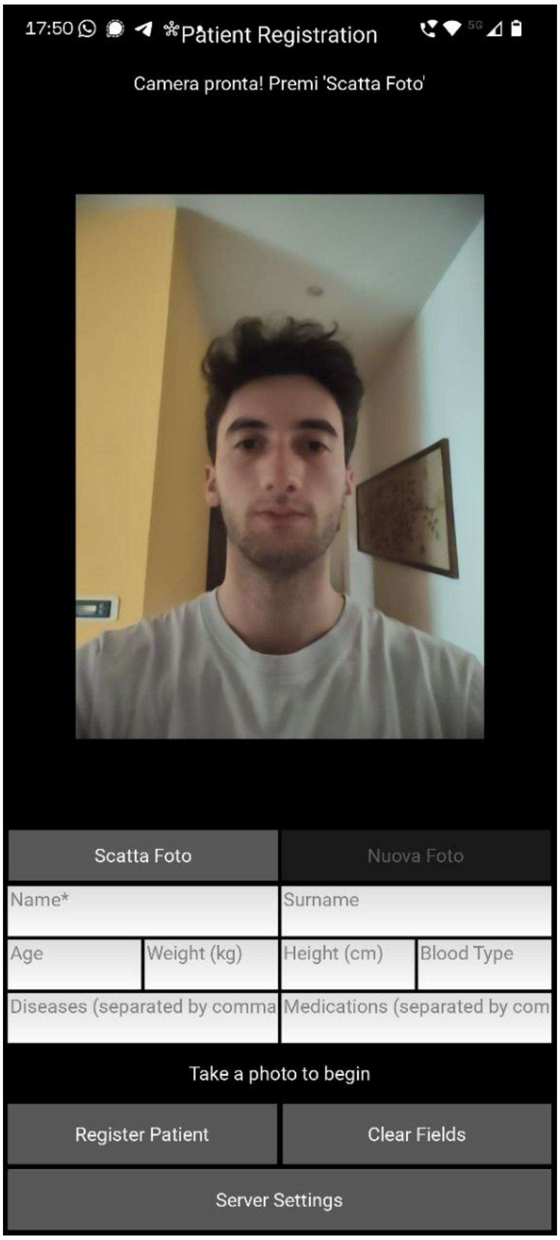

Figure 6: Registration Screen.

The application provides the following functionalities:

- User registration with personal and medical data.
- Simulation of backend interaction with the server.
- Facial recognition simulation directly from the smartphone.

Real-Time Operating System The smartwatch firmware uses **FreeRTOS** as its operating system. This choice was motivated by its advantages:

- Efficient task scheduling and prioritisation, ensuring responsiveness even with multiple processes.
- Low memory footprint, suitable for embedded devices like the ESP32S3.
- Broad compatibility with ESP-IDF, the official development framework for ESP microcontrollers.

3.4 Prototyping & Testing

The development process followed an iterative prototyping approach, aiming to incrementally validate the feasibility of each hardware and software component before full system integration. Each stage involved testing individual modules, integrating them, and refining based on the results.

Figure 7: Fit test of internal components to ensure proper placement and integration within the 3D-printed case.

Prototyping Phases The main prototyping phases included:

1. Initial hardware setup and microcontroller testing.
2. Integration of the camera module and image capture functionalities.
3. Development of the Bluetooth communication stack between the smartwatch and smartphone.

4. Implementation of the Android application for receiving photos and interacting with the server.

5. Development of the Flask server APIs and facial recognition prototype.

6. Assembly and ergonomic testing of the 3D-printed case.

Main Tests Performed The following key tests were conducted to evaluate the prototype:

- **Photo capture and transmission test.** Verified that the smartwatch could successfully capture and send images via BLE to the smartphone. During testing, some issues were identified with data transmission due to BLE's default sequential transfer limit of 512 bytes per packet. Although BLE 5.0 supports higher data rates and parallel transmissions, the current prototype implementation has not yet exploited these features, resulting in longer transfer times for full images.

- **Bluetooth connection stability test.** Evaluated the reliability and stability of the BLE connection between the smartwatch and smartphone in different operating environments, including indoor and outdoor settings.

- **Data communication with server.** Tested HTTP POST requests from the smartphone to the server and the reception of patient data in response. These tests confirmed the feasibility of integrating cloud-based facial recognition services in future versions.

- **Battery duration test.** Preliminary tests indicated that while theoretical battery life was up to 48 hours in light-sleep mode, actual usage times were significantly lower under continuous operation. Further optimisation will be required to meet the energy efficiency requirements for deployment.

- **Usability assessment.** Evaluated the ergonomics, ease of use, and practicality of wearing the prototype during typical movements and simulated emergency tasks. Despite the bulkiness of some components due to prototyping constraints, the watch remained wearable and operational.

- **Facial recognition functionality test.** While facial recognition was implemented using the `face_recognition` Python module, its testing was not considered a primary evaluation criterion, as the focus of this project was to demonstrate hardware feasibility and system integration rather than optimise recognition performance.

Lessons Learned Several insights emerged from the prototyping and testing phase:

- BLE data transfer optimisation is crucial for efficient operation; exploring BLE 5.0 parallel data transfer capabilities will be a priority in future iterations.

- Hardware design should account for battery capacity and consumption trade-offs, especially for mission-critical applications requiring long operational times.
- Early ergonomic testing significantly aids in identifying design improvements to enhance usability and user acceptance.
- Rapid prototyping using Python for both server and Android application allowed us to validate system architecture choices quickly, even though production-grade implementations will require more robust and secure frameworks.

Overall, the tests demonstrated the technical feasibility and promising potential of the E-Watch concept, laying the foundation for future development, optimisation, and deployment in real-world emergency scenarios.

4 Final Product Architecture

This section presents the final envisioned architecture of the E-Watch system, including its hardware specifications and a high-level overview of its software modules.

4.1 System Overview Diagram

The following diagram illustrates the main hardware components and input/output connections of the smartwatch prototype.

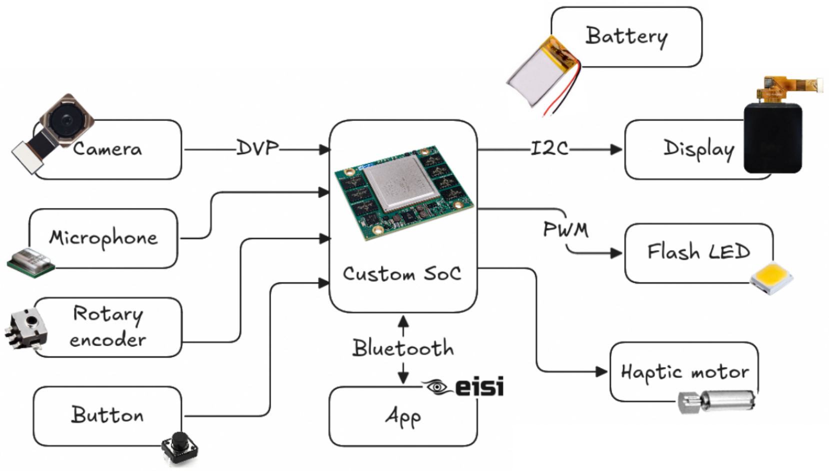

Figure 8: Hardware Inputs and Outputs, including planned haptic motor (future addition).

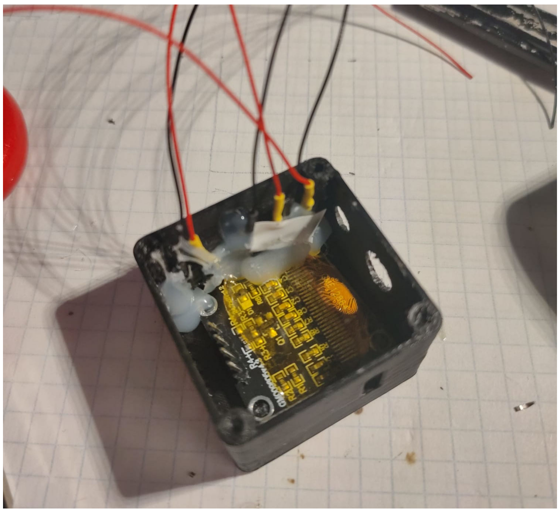

Figure 9: Assembly process: soldering and connecting all components.

4.2 Hardware Specifications

Microcontroller (ESP32S3 Sense)

- **Model:** Seeed Studio XIAO ESP32S3 Sense
- **CPU:** Dual-core Xtensa LX7 up to 240MHz
- **RAM:** 512KB SRAM, 8MB PSRAM
- **Connectivity:** Bluetooth 5.0, Wi-Fi 802.11 b/g/n
- **Power consumption:**
  - Active mode: approx. 160-240mA depending on workload
  - Light-sleep mode: approx. 5-10mA
  - Deep-sleep mode: approx. 0.8mA

Camera

- **Model:** OV2640
- **Resolution:** Up to 1600x1200 UXGA; used at lower resolutions (e.g. 320x240) to optimise BLE transmission.

Battery

- **Type:** 3.7V LiPo battery
- **Capacity:** 290mAh

- **Estimated autonomy:** up to 48h in light-sleep mode, approx. 4-6h under continuous active use.

Figure 10: Fully assembled and soldered internal circuit.

Haptic Feedback (Future Addition) A small vibration motor is planned for future versions to provide haptic feedback to the user for alarms, confirmations, or silent notifications.

4.3 Software Architecture

The following diagram illustrates the high-level software module architecture, showing how each module interacts within the system and with external components (smartphone and server).

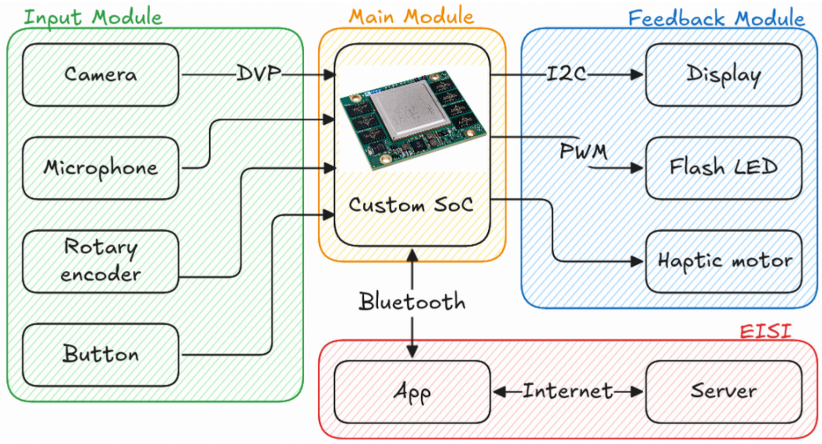

Figure 11: Software Modules: Input Module, Main Module (SoC), Output Module, EISI Module.

Module Overview

- **Input Module** – handles camera capture and potential future sensor readings.
- **Main Module (SoC)** – coordinates tasks, manages BLE communication, and runs under FreeRTOS for efficient scheduling.
- **Output Module** – manages OLED display, LED indicators, and (future) haptic motor control.
- **EISI Module** – represents external smartphone and server components for data processing, facial recognition, and retrieval of patient medical information.

Code Repository All code developed for this project, including the smartwatch firmware, Android application prototype, and server implementation, is available on GitHub at the following link:

`https://github.com/Orteip/EWATCH`

This repository includes detailed documentation, build instructions, and example configurations to facilitate reproducibility and future development by the team or external collaborators.

5 Critical Discussion

This section summarises the main strengths, limitations, and potential future improvements of the E-Watch project.

5.1 Strengths

The project demonstrated several notable strengths:

- **Innovation.** Introducing a smartwatch as an alternative or complementary device to smartphone-based solutions offers a novel approach to emergency patient identification.
- **Portability and speed.** The wrist-worn form factor ensures high portability, immediate accessibility, and faster response times in emergency interventions.
- **Modular system architecture.** The integration of ESP-based hardware, a smartphone app, and a server backend provides a modular design, enabling independent upgrades of each component in future iterations.

5.2 Limitations

However, the current prototype also presents some limitations:

- **High energy consumption.** Battery life is limited, especially when using the camera and BLE continuously.
- **Hardware constraints.** The ESP32S3 microcontroller has insufficient computational power for local heavy processing tasks such as on-device facial recognition.
- **Connectivity dependency.** The system requires a stable connection with the smartphone and server, which may not always be available in all emergency scenarios.

5.3 Future Work

Future developments will aim to address these limitations and extend the system's capabilities:

- **On-device facial recognition.** Integrating local facial recognition algorithms to reduce dependency on remote servers, requiring more powerful edge AI hardware.
- **Further miniaturisation.** Reducing the device size to improve comfort and wearability for long-term use.
- **Native Android/iOS application.** Developing a full-featured native mobile app for improved user experience and seamless integration with medical databases.
- **Industrial feasibility and certifications.** Conducting feasibility studies for industrial production and obtaining medical device certifications, as well as IP68 waterproof and dustproof ratings.

- **Energy optimisation.** Implementing advanced power management strategies to increase operational battery life.
- **Camera quality improvements.** Using higher quality cameras with better resolution and low-light performance to enhance facial recognition reliability.
- **Improved connectivity reliability.** Enhancing the robustness and stability of the Bluetooth connection between the smartwatch and smartphone, particularly in noisy environments.

6 Conclusion

This project explored the design, development, and prototyping of E-Watch, an innovative smartwatch aimed at supporting emergency responders in the rapid identification and management of patients.
The proposed system integrates a wearable device equipped with a camera and BLE communication, a smartphone application, and a server-based facial recognition backend. While the current prototype is at an early stage, it demonstrates the feasibility and potential impact of wearable technologies to improve response times, reduce human errors, and provide critical medical data seamlessly in emergency contexts.
From a healthcare perspective, such technology could significantly enhance patient safety, particularly for vulnerable individuals with severe pathologies who may be unconscious or unable to communicate during emergencies.
Personally, working on this project has strengthened my skills in embedded systems design, hardware-software integration, and rapid prototyping. I also gained valuable experience in teamwork, iterative design processes, and critically evaluating technological solutions within real-world constraints and needs. Overall, this project confirmed the importance of user-centred design and interdisciplinary collaboration to create impactful and effective solutions in the healthcare sector.