HW4: Smart Watch using ESP32
Note

Goda Gutparakytė

November 2025

1 Design

This is a prototype of a Smart Watch made with ESP32 WiFi enabled microcontroller. It measures heart rate, steps, fetches location and weather using an API, and gets local time and date using WiFi. It also has a button that switches the theme - dark, light and pink. It does have a battery and charging module for portable power.

2 Test/Accuracy results

For heart rate, I calibrated the reading using my own heart rate and a smart watch I own. The results are pretty accurate. For step detection, the algorithm is definitely not perfect, but it is able to detect a move and count it as a step. Other than that, there was not a lot of other testing to do.

3 Timing budget

The system uses simple timed loops to ensure that all sensors and display elements are updated at appropriate intervals.

- **Button press:** Checked via interrupt, triggers immediately on press.

- **Heart rate sensor (MAX30102):** Sampled continuously using the sensor's buffer and proccessed when new data is available

- **Accelerometer/gyroscope (MPU6050):** Sampled every 50 ms for step counting.

- **Display update:** Entire screen is refreshed once every 1000 ms.

- **Time update (NTP):** Updated once during startup, then kept using the ESP32 internal clock.

- **Weather data (Open-Meteo API):** Fetched every 15 minutes.

4 Issues

My initial idea was to do this project with ESP32 C3 SuperMini and make it portable. However, this microcontroller is not well documented yet and I had a ton of issues trying to set it up. Firstly, when trying to make it work with GC9A01 display, I could not get it work. The first problem I encountered was having to change the pin configuration in User_Setup folder. When I changed that, it still did not work. After a lot of research, in some blog it said to downgrade the version because the newest one does not work correclty with this display. After doing that, the display started working. Then it was time to connect the heart rate sensor. But that brought even more issues as I had to downgrade the pulse sensor version as well and it still did not work as expected. Other problem was trying to connect both the pulse sensor and the accelerometer

and gyrscope on the same pins (both use I2C protocols). Lastly, the ESP32 C3 SuperMini anthenna is so weak, that only 10% of the times I tried to connect to the WiFi it was successful. Since it was not working as I expected, I had to switch to ESP32 WiFi enabled microcontroller. Even though it is way bigger and not suitable for a portbale smart watch, I was able to make all of the functionalities work. Some issues I have encountered after switching the microcontrollers was using MPU6050 to calculate and detect steps. It is definitely still not ideal (and probably goes wild while travelling in a vehicle), for a prototype version it works quite well. Calibrating and calculating the heart rate was also a challenge, but after some time it shows pretty accurate hear rate. Overall, it was definitely a step out of my comfort zone. Even though the final version is not what I had initially in my mind, I am happy to at least have tried to work with ESP32 C3 SuperMini, and also experiencing working with SPI and I2C protocols.