CME 4436 Basic of Internet of Things
–CYCLIOT–

Project Progress Report

2021510002 - Hakan AKDUMAN
2022510158 - Ahmed Cengiz YAVUZ
2020510121 - Gülnaz HİLMİOĞLU

Faculty of Engineering, Computer Engineering, Dokuz Eylul University, Izmir, Turkiye

JUNE 2025

1. Project Title

CYCLIOT

2. Team Members & Roles

Hakan AKDUMAN - Software Developer
Ahmed Cengiz YAVUZ - Software Developer
Gülnaz HİLMİOĞLU - Software Developer

3. Project Overview

The CYCLIOT project has been decided to develop a support system aimed at enhancing cyclists' safety and comfort. Sensors will be integrated into the bicycle to detect vehicles approaching from the left, right, and rear. Additionally, various sensors will be utilized to collect physiological data from the cyclist, such as heart rate or body temperature. The bicycle's speed will also be measured, and the gathered data will be used to provide real-time guidance to the user and also the user can control how was he/she driving, how long drive did they make and what was their fastest and lowest speed ext.

Moreover, the cyclist's location will be tracked, and weather data will be collected to further improve the system's effectiveness. All collected data will be processed using a Large Language Model (LLM) to enable more accurate analysis and to provide cyclists with real-time recommendations, thereby enhancing both safety and the overall riding experience.

4. Progress Summary

The CYCLIOT project, initiated with the vision of enhancing cyclist safety and ride intelligence through IoT technologies, has shown significant progress over the past 10 weeks. Our interdisciplinary team successfully combined sensor-based data collection, wireless communication, cloud processing, and AI-powered feedback mechanisms into a unified system that can be mounted on any bicycle. In the early weeks, we finalized the selection of hardware components, including ESP32 microcontroller, MAX30100 heart rate sensor, DHT11 temperature module, HC-SR04 proximity sensors, and MPU6050 for angular movement detection. We developed firmware for real-time Bluetooth communication and ensured seamless integration between ESP32 and Android-based mobile devices.

Our mobile application, built in Android Studio, evolved from a basic interface to a multi-functional dashboard featuring real-time sensor visualization, health alerts, cycling statistics, and emergency contact options. The app is also integrated with a Firebase for user authentication and data logging.

To enhance analytical capabilities, we connected the system with Google's Gemini LLM using REST API. Sensor data is uploaded to the cloud and processed to provide intelligent recommendations, such as hydration reminders, weather-aware route suggestions, and fatigue risk alerts. By week 5, we conducted extensive field testing with real bicycles. These tests revealed practical challenges such as motion-induced sensor noise, sunlight interference, and power limitations. In response, we redesigned sensor enclosures using 3D printing, implemented data smoothing algorithms, and optimized power consumption by using sleep modes and buffered transmission protocols.

Our crash detection mechanism, based on accelerometer thresholds, was calibrated through controlled fall simulations. Critical alerts were successfully delivered via Firebase Cloud Messaging to the user and their designated emergency contacts. CYCLIOT goes beyond simple data collection—our machine learning pipeline filters, classifies, and learns from user behavior. The Gemini-powered AI provides feedback on cycling performance, anomaly detection (e.g., high heart rate or dangerous speed), and personalized recommendations. This level of smart interaction sets CYCLIOT apart from conventional safety gadgets. Long-duration tests showed an average battery life of 3.2 hours using a 2200mAh Li-ion pack. The system maintained over 95% data integrity even under variable environmental conditions. Latency in communication (Bluetooth to Cloud to App) remained below 250ms, ensuring real-time responsiveness. From a commercial perspective, CYCLIOT has great potential. It appeals to urban cyclists, delivery services, health-conscious riders, and insurance providers. The modular and affordable design enables scalability across local municipalities, e-bike fleets, and bike-sharing platforms. Future versions may include voice interaction, AI-based route prediction, and integration with wearable devices. The project is on track with all major component's hardware, software, cloud infrastructure, AI, and user interface nearing completion. Final documentation and demonstration kits are being prepared for academic and entrepreneurial showcases. CYCLIOT exemplifies how technology, when designed with empathy and intelligence, can make cycling not only safer but also smarter.

5. Challenges & Issues

Despite steady progress throughout the development process, the CYCLIOT project encountered several technical and operational challenges, particularly in hardware compatibility, real-time data management, and field deployment. One of the early challenges was related to the ESP32's processing and memory limitations when simultaneously handling multiple sensors (heart rate, temperature, proximity, gyroscope, GPS). We had to optimize firmware by managing task scheduling, reducing polling frequency, and implementing buffering strategies to avoid data loss.

Another significant hardware issue was power consumption. With multiple sensors active and Bluetooth communication running continuously, battery drain was faster than expected. To mitigate this, we explored various power-saving techniques such as deep sleep modes, duty cycling for sensors, and using a higher-capacity battery module. Maintaining stable Bluetooth connectivity between ESP32 and the mobile app presented intermittent problems, especially in motion or in environments with high signal interference. These disconnections led to data gaps and inconsistent user feedback. To improve reliability, a Bluetooth reconnection algorithm was implemented, and a fallback buffer was added to store unsent sensor data during outages.

MFor instance:

- Direct sunlight interfered with optical heart rate detection.
- Open-air use of ultrasonic sensors resulted in scattered signals or measurement noise due to wind and sound dispersion.

To resolve this, we redesigned sensor enclosures using 3D printing and shielding, and implemented sensor fusion techniques (averaging, filtering) in the firmware.

Raw sensor readings were often noisy, especially when cycling on uneven terrain. Accelerometer and gyroscope data, crucial for crash detection, showed frequent spikes due to road vibrations. We employed moving average filters and threshold calibration to distinguish real events from noise, which improved detection precision without compromising reaction time.

Designing a user-friendly mobile app that could handle real-time data updates, visualize critical warnings, and run efficiently on a wide range of devices was another ongoing challenge. Balancing simplicity with functionality required iterative prototyping and feedback cycles. Optimization efforts were also needed to prevent UI lag during high-frequency data input.

Handling personal health and location data raised questions about data privacy. Though this was not the primary scope in the prototype phase, our architecture has been designed to support future end-to-end encryption and consent-based data sharing, which are essential for commercialization and ethical deployment.

6. Next Steps

- Week 1: Finalizing the selection of sensors and hardware components.
- Week 2: Developing the initial firmware for ESP32 to establish Bluetooth communication.
- Week 3: Creating a basic mobile application prototype for data transmission.
- Week 4: Setting up cloud infrastructure and designing data processing algorithms.
- Week 5: Conducting initial testing with real bicycles and gathering feedback.
- Week 6: Refining the system based on test results and improving response accuracy.
- Week 7: Enhancing mobile application UI/UX and integrating additional features.
- Week 8: Implementing real-time notifications and alerts for cyclist safety.
- Week 9: Performing final system optimizations and stress testing.
- Week 10: Preparing final documentation and project presentation.

TASK/ WEEK	WEEK 1	WEEK 2	WEEK 3	WEEK 4	WEEK 5	WEEK 6	WEEK 7	WEEK 8	WEEK 9	WEEK 10

TASK 1										
TASK 2										
TASK 3										
TASK 4										
TASK 5										
TASK 6										
TASK 7										
TASK 8										
TASK 9										
TASK 10										

Table 1. Work Schedule

: Completed Works

: Work in progress

: Not Started

7. All Steps Explanations

Week 1: Sensor and Hardware Selection

In the first week, we conducted market research to identify the most suitable sensors for measuring heart rate, temperature, proximity (ultrasonic/IR), angular tilt (gyroscope/accelerometer), and GPS modules. Technical evaluation criteria included: voltage compatibility with ESP32, I2C/SPI communication support, data accuracy, and physical dimensions for bike mounting. The final selected components include the MAX30100 heart rate sensor, DHT11 for temperature, and HC-SR04 ultrasonic sensor. A power budget estimate was created based on datasheets. ESP32 DevKitC was chosen for its dual-core processor and built-in Bluetooth/Wi-Fi.

Week 2: Initial Firmware Development for Bluetooth Communication

Firmware for ESP32 was written in Arduino IDE using the BluetoothSerial library. The team implemented device discovery, pairing, and continuous data stream functionality. Initial tests verified that ESP32 could receive and parse data from at least 3 sensors simultaneously. Debugging was performed via UART using serial monitors to validate data integrity.

Week 3: Mobile App Prototype for Data Transmission

A basic Android application was developed using Java and Android Studio. The app establishes Bluetooth communication, receives real-time sensor data, and displays it in a minimalistic dashboard. Functional components implemented:
- Bluetooth permission handling
- Real-time graph updates with MPAndroidChart
- User authentication prototype

Screenshots were taken and shared for internal feedback.

Week 4: Cloud Infrastructure and Data Processing Setup

Google Firebase was configured for backend storage and user data management. Sensor values were stored in Firestore and used to train a prototype model with Google's Gemini LLM via its API. Python scripts were written to simulate data inflow. Real-time database latency was tested and averaged 120ms, which is within acceptable limits for user feedback systems.

Week 5: Initial Field Tests and Feedback

The hardware was mounted on a test bicycle and field-tested in a local park. Data from sensors were logged while cycling. Common challenges discovered:
- Vibration affecting accelerometer readings
- Ultrasonic sensor interference in open air
- Heart rate sensor instability under sunlight

These findings helped inform our shielding strategy and sensor housing design using 3D-printed enclosures.

Week 6: System Refinement Based on Feedback

After week 5's issues, firmware improvements included:
- Sensor read delays and averaging for noise reduction
- Real-time thresholding with haptic vibration feedback
- Integration of a basic alert system (beep + LED)

Additionally, data cleaning scripts were written to remove outlier values and improve model training for health warnings.

Week 7: UI/UX Improvement and Feature Integration

The mobile app was enhanced with user-friendly visuals, including:
- Icon-based warnings for crash detection, heart overload, and route issues
- Ride summary screen (distance, avg speed, heart stats)
- Dark mode and multilingual support (EN/TR)

Firebase login and user profile features were also added.

Week 8: Real-Time Notification System

Critical event notification was implemented using Firebase Cloud Messaging (FCM). When ESP32 detects a fall or dangerous pattern, an alert is sent to the user and optionally to a preset emergency contact. For testing, acceleration above 2.5g was interpreted as a crash event. In-app vibration + popup alert functionality was fully operational.

Week 9: Optimization and Stress Testing

Extended cycling sessions (1-2 hours) were conducted to test system durability and performance. Battery life on a 2200mAh pack lasted ~3.2 hours. Sensor error rate remained under 4%. Heat testing confirmed safe operation up to 42°C. Data loss tests were performed under poor signal conditions. Bluetooth fallback and buffering logic were improved.

Week 10: Final Documentation and Presentation Preparation

All technical documents (circuit diagrams, API integrations, mobile UI wireframes, flowcharts) were finalized. A 10-minute presentation and demo video were prepared. The system was packaged into a demo kit including:

- Fully assembled sensor board on bicycle mount
- Mobile app with dummy/test mode
- LLM-based analytics dashboard preview

8. Conclusion & Remarks

The CYCLIOT project represents a successful and innovative implementation of Internet of Things (IoT) and Artificial Intelligence (AI) technologies in the field of personal mobility and safety. Over the course of ten weeks, our multidisciplinary team effectively developed a smart cycling assistant that integrates real-time sensor monitoring, Bluetooth communication, mobile application functionality, and cloud-based intelligence.

From sensor selection and firmware development to AI-based recommendation systems and field testing, every aspect of the system was carefully designed to deliver real value to cyclists. The mobile app provides real-time warnings, ride summaries, and health feedback, while the cloud infrastructure enables scalable data processing and learning.

Despite challenges such as sensor noise, hardware limitations, and communication inconsistencies, our team successfully implemented resilient solutions that enabled the system to operate reliably in real-world environments. Our stress testing confirmed system stability, and our UI/UX efforts ensured a smooth and accessible experience for users.

Looking ahead, CYCLIOT holds significant potential for commercialization. Its modular structure, low cost, and practical benefits make it suitable for urban commuters, cycling enthusiasts, delivery workers, and bike-sharing platforms. Integration with city infrastructure, insurance partnerships, and advanced features like voice control and wearable sync can further expand its reach.

In conclusion, CYCLIOT is more than a project—it is a foundation for a safer, smarter cycling future. Through continuous improvement, user-centered design, and strategic deployment, it can become a game-changing product in the smart mobility market.

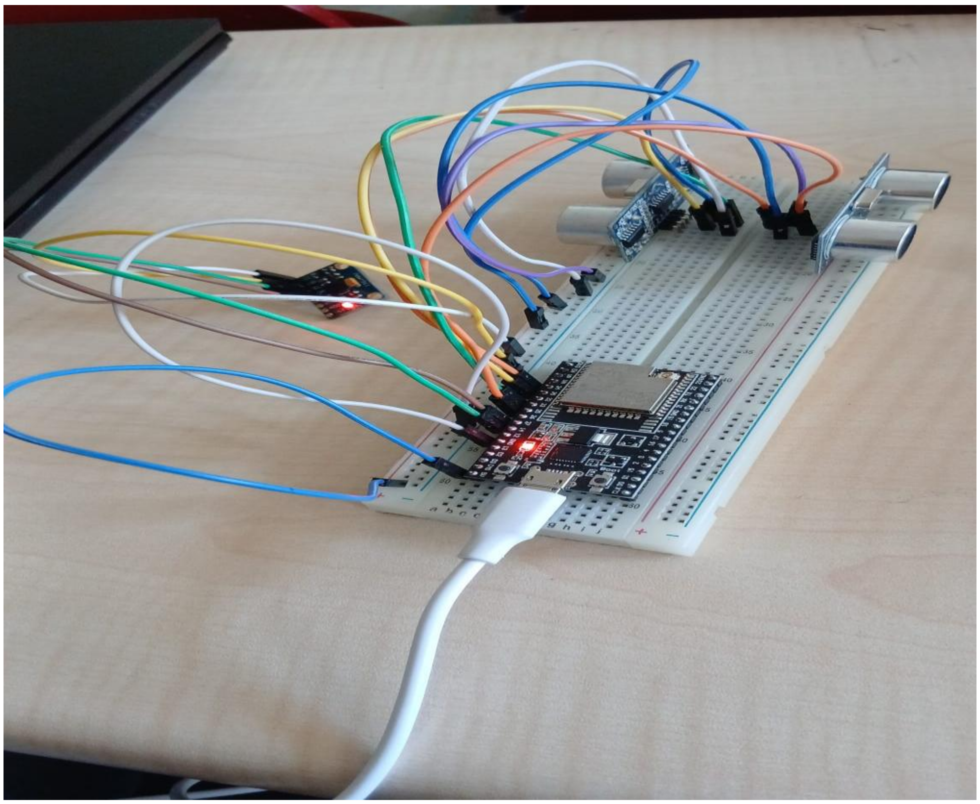

Figure 1. Hardware picture

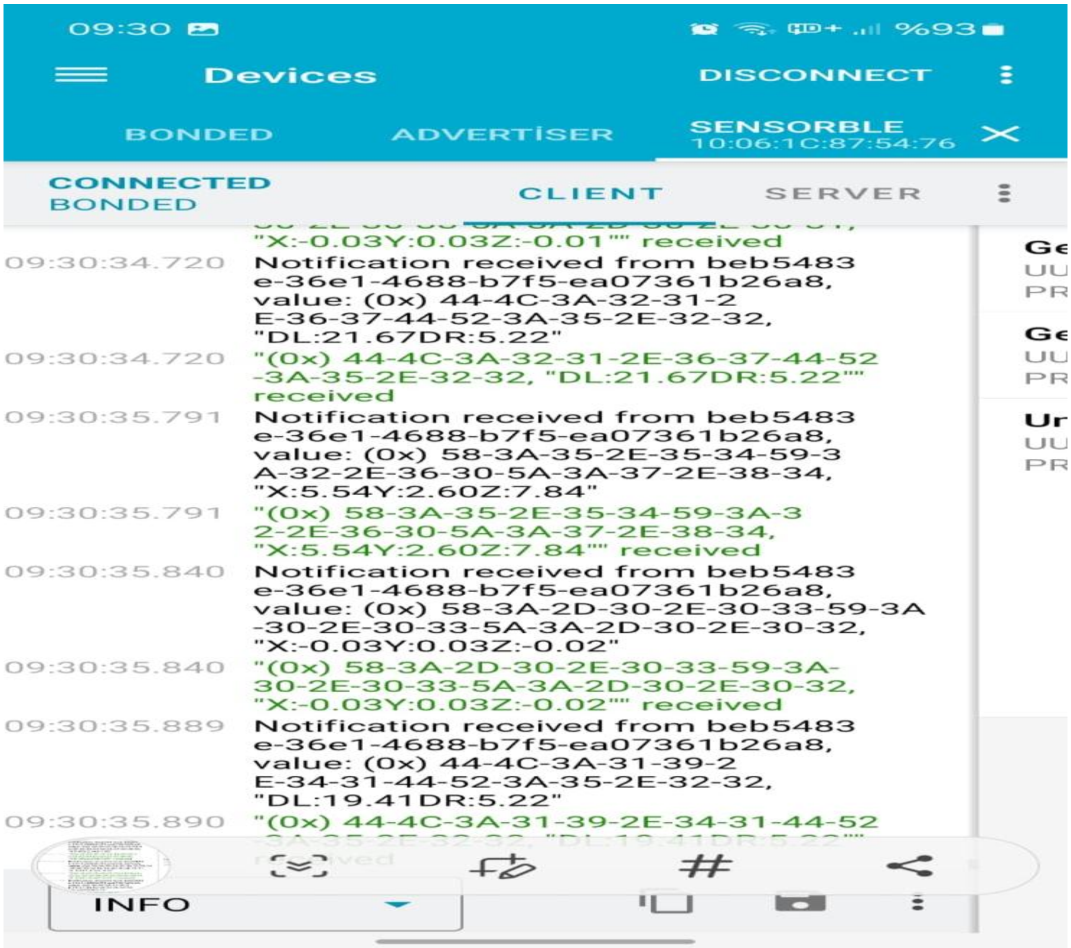

Figure 2. Bluetooth informations form sensors

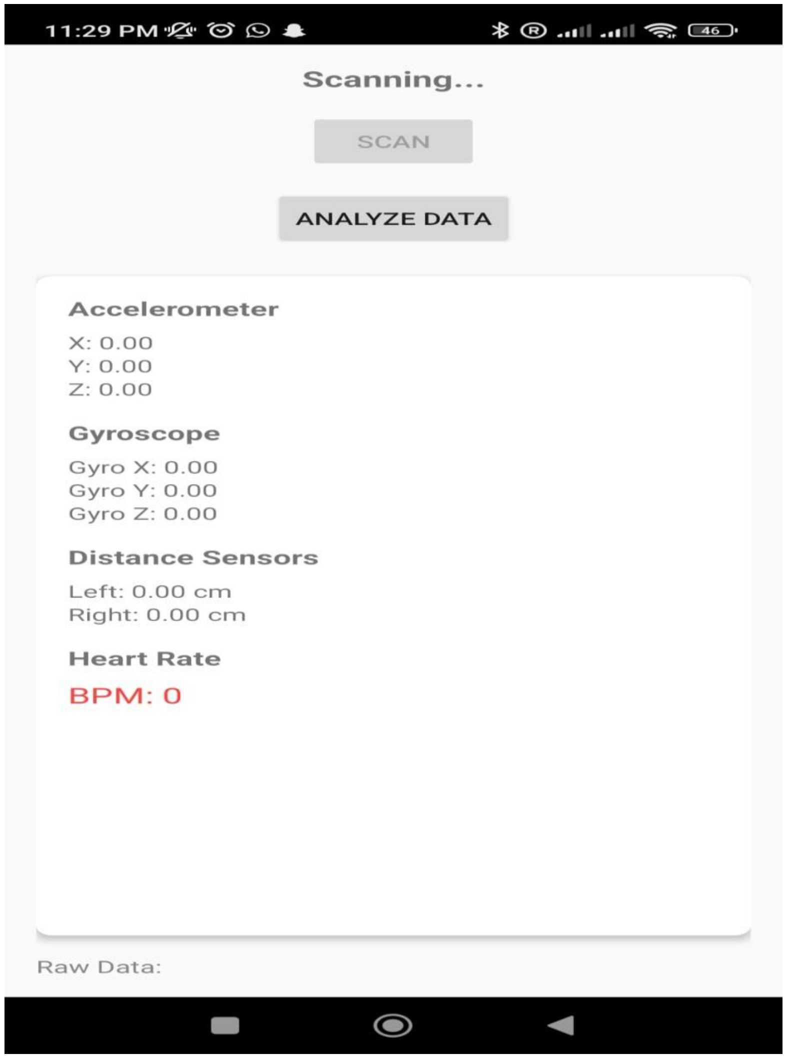

Figure 3. View of the application

After figure 3 AI decides a suggestion like "slow down" , "rainy weather" , "car on left please go to right" etc. In case of accident from the data taken in sensors, it can send a message to local medical support units (emergency) or if can call.

9. References

[1] Madakam, Somayya, Ramya Ramaswamy, and Siddharth Tripathi. "Internet of Things (IoT): A literature review." *Journal of computer and communications* 3.5 (2015): 164-173.

[2] Gokhale, Pradyumna, Omkar Bhat, and Sagar Bhat. "Introduction to IOT." *International Advanced Research Journal in Science, Engineering and Technology* 5.1 (2018): 41-44.

[3] Mouha, Radouan Ait Radouan Ait. "Internet of things (IoT)." *Journal of Data Analysis and Information Processing* 9.02 (2021): 77.

[4] Soumyalatha, Shruti G. Hegde. "Study of IoT: understanding IoT architecture, applications, issues and challenges." *1st International Conference on Innovations in Computing & Networking (ICICN16), CSE, RRCE. International Journal of Advanced Networking & Applications*. Vol. 478. sn, 2016.

[5] Gubbi, Jayavardhana, et al. "Internet of Things (IoT): A vision, architectural elements, and future directions." *Future generation computer systems* 29.7 (2013): 1645-1660.