EEL4599 Final Report

Autonomous Vehicle IoT

GitHub Repository

Gabriel Frank, Ryan Botwinick, Jacob Frankel
Electrical and Computer Engineering Department
University of Florida
Gainesville, FL, USA
gfrank1@ufl.edu, rbotwinick@ufl.edu, jacob.frankel@ufl.edu

I. Abstract

This project presents a distributed IoT sensor network for real-time obstacle detection and orientation tracking in autonomous vehicles. The system consists of two source nodes: an ESP32 with an ultrasonic sensor, and a Raspberry Pi Pico with an IMU. These nodes transmit sensor data via XBee radios to a laptop-based sink node, which forwards the processed data to ThingSpeak for cloud analysis. The contributors include Gabriel Frank (ESP32 node development), Ryan Botwinick (IMU integration on Pi Pico), and Jacob Frankel (XBee data reception and ThingSpeak integration). This report outlines the system design, deployment environment, data flow, and experimental results.

II. Introduction

Autonomous vehicles require robust environmental awareness to navigate safely. Our hypothetical scenario involves deploying a vehicle with sensor nodes to detect obstacles and track orientation during movement. These measurements help avoid collisions and maintain course stability. The system is designed to collect, transmit, and analyze sensor data in real time via a wireless XBee mesh network. Our project aligns with this use case by providing an end-to-end IoT architecture capable of supporting real-time situational awareness and remote data visualization through cloud platforms like ThingSpeak.

III. System Architecture

A. Hardware

- Node 1 (ESP32 + Ultrasonic Sensor)
  - Sensor: HC-SR04
  - Range: 2–400 cm
  - Resolution: ~3 mm
  - Communication: GPIO to ESP32, UART to XBee
- Node 2 (Raspberry Pi Pico + IMU)
  - Sensor: MPU6050
  - Range: ±2g–±16g (accelerometer), ±250°/s–±2000°/s (gyroscope)
  - Resolution: 16-bit
  - Communication: I2C to Pico, UART to XBee
- Sink Node (Laptop)
  - Receives XBee serial data
  - Parses and uploads data to ThingSpeak using Python

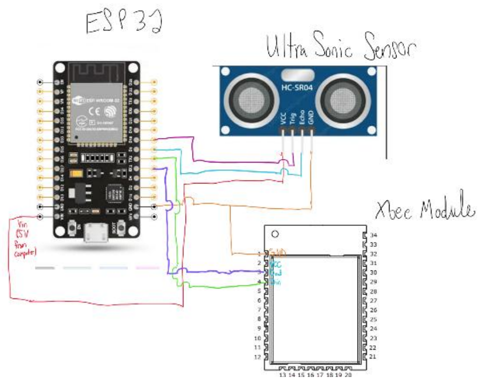

Fig. 1. System wiring diagram (ESP)

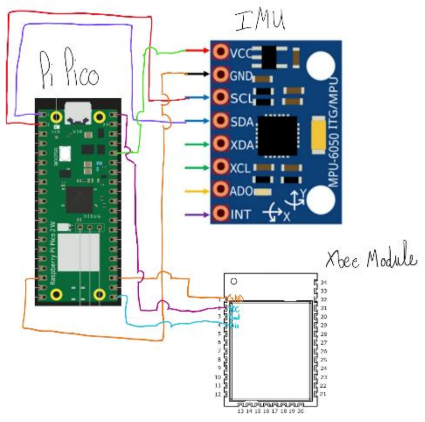

Fig. 2. System wiring diagram (Pico)

B. *Software*

- Micropython/C for Pico and ESP32 firmware, respectively
- Python on PC for data aggregation and cloud upload

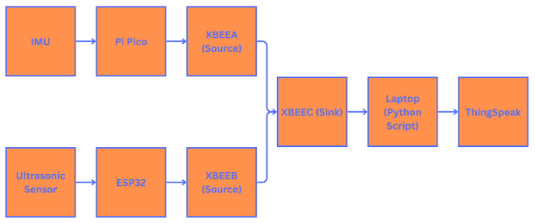

Fig. 3. Data flow diagram

IV. Experimental Procedure

The sensor network was deployed indoors in a controlled test environment to simulate vehicle motion. The ESP32 node, equipped with an ultrasonic sensor, was positioned facing an object at varying distances to emulate approaching obstacles. Meanwhile, the Pi Pico node, equipped with an IMU, was mounted on a mobile platform to capture orientation changes during movement. Variations in sensor readings were induced by moving the platform forward, backward, and along curved paths, as well as repositioning the object in front of the ultrasonic sensor.

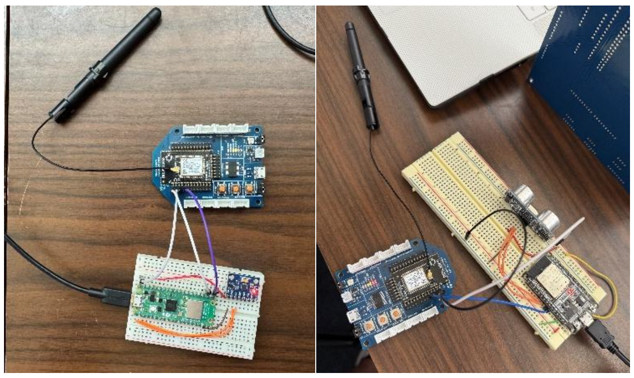

Fig. 4. Deployment Enviornment

V. Experimental Results

ThingSpeak plots show a clear correlation between object proximity and ultrasonic sensor output. IMU data reflected angular displacement and acceleration during turns. The results aligned with expectations:

- Ultrasonic readings decreased when approaching a wall.
- Accelerometer data spiked with harsh linear movements.
- Gyroscope data spiked during rapid turns.

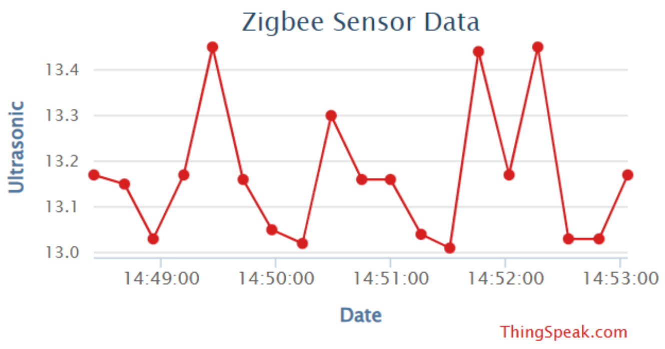

Fig. 5. Ultrasonic graph fron ThingSpeak

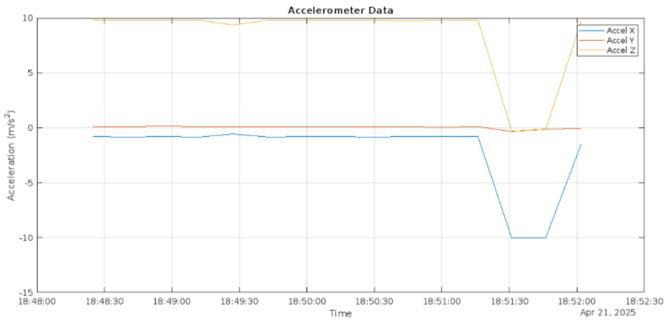

Fig. 6. Accelerometer graph from ThingSpeak

VI. Conclusion

This project demonstrated a complete wireless sensor network for autonomous navigation support using cost-effective microcontrollers and sensors. Key takeaways included managing asynchronous serial transmission with XBee, parsing multi-node sensor streams, and integrating with cloud platforms. Future improvements could involve time-synchronizing sensor streams, improving node energy efficiency, and adding GPS for position tracking.