Day 2 Agile Progress

Subsystem & Signal Integration

- Refactored and optimized the **signal handling mechanism** within the sensor.c and subsystem.c modules.
- Integrated signal-based control flow in ui.c to respond to asynchronous events like ignition off and emergency stop.
- Ensured safe state transitions using SIGUSR1 and mutex-protected shared memory updates.

Dashboard Enhancements

- Improved the **real-time dashboard** to reflect dynamic sensor values and control states.
- Added visual indicators for:
    - Engine temperature and speed
    - Gear position and fuel level
    - Emergency alerts (crash, obstacle detection)
- Ensured UI responsiveness under concurrent sensor updates.

Remote Station Communication

- Developed a **client-server architecture** for remote monitoring:
    - **Server**: Receives and logs ECU data from shared memory.
    - **Client**: Connects to server, displays live vehicle status.
- Used sockets for reliable data transmission between remote station and ECU system.

Controller Logic Development

- Implemented logic for:
    - **Emergency stop** based on crash or obstacle detection.
    - **Airbag deployment** prioritization.
    - **Fuel status** classification (Red, Yellow, White).
    - **Light control** based on gear and low-light conditions.
- Modularized control decisions for scalability and testing.

Build & Automation

- Created a **Makefile** to automate compilation of all modules (sensor.c, subsystem.c, ui.c, controller.c, etc.).
- Developed **shell scripts** for:
    - Launching ECU and subsystem processes in separate terminals.
    - Cleaning up shared memory and IPC resources.

- Simulating signal events for testing.