In the automotive industry, ECUs are communicating together to provide safety and driver's comfort.

ECUs are microcontrollers connected with input and output devices, input devices to sense the surrounding environments, and output devices to perform actions according to readings that came from input devices.

Project Requirements

1. Provide Fully Static Design
2. Provide a Fully Dynamic design

For two ECUs communicating together to control car lights according to door state, light switch state, and car speed state.

Project Instructions

The project can be distributed over 2 main tasks:

Provide Fully Static Design

1. Read project requirements
2. Static design analysis

Provide Fully Dynamic design

1. Dynamic design analysis

Read project requirements	Hardware requirements: 1. Two microcontrollers connected via CAN bus 2. One Door sensor (D) 3. One Light switch (L) 4. One Speed sensor (S) 5. ECU 1 connected to D, S, and L, all input devices 6. Two lights, right (RL) and left (LL) 7. One buzzer (B) 8. ECU 2 connected to RL, LL, and B, all output devices Software requirements: 1. ECU 1 will send status messages periodically to ECU 2 through the CAN protocol 2. Status messages will be sent using Basic Communication Module (BCM) 3. Door state message will be sent every 10ms to ECU 2 4. Light switch state message will be sent every 20ms to ECU 2 5. Speed state message will be sent every 5ms to ECU 2 6. Each ECU will have an OS and application SW components 7. If the door is opened while the car is moving → Buzzer ON, Lights OFF 8. If the door is opened while the car is stopped → Buzzer OFF, Lights ON 9. If the door is closed while the lights were ON → Lights are OFF after 3 seconds 10. If the car is moving and the light switch is pressed → Buzzer OFF, Lights ON 11. If the car is stopped and the light switch is pressed →

	Buzzer ON, Lights ON You should draw and deliver the system schematic (Block Diagram) according to your requirements understanding, a screenshot is required

2- Static design analysis	For ECU 1: 1. Make the layered architecture 2. Specify ECU components and modules 3. Provide full detailed APIs for each module as well as a detailed description for the used typedefs 4. Prepare your folder structure according to the previous points For ECU 2: 1. Make the layered architecture 2. Specify ECU components and modules 3. Provide full detailed APIs for each module as well as a detailed description for the used typedefs 4. Prepare your folder structure according to the previous points You should deliver a pdf file containing all your work and a video recording where you will discuss your work (maximum 3min long)

Provide Fully Dynamic design

CRITERIA	MEETS SPECIFICATIONS

CRITERIA	MEETS SPECIFICATIONS
Dynamic design analysis	For ECU 1: 1. Draw a state machine diagram for each ECU component 2. Draw a state machine diagram for the ECU operation 3. Draw the sequence diagram for the ECU 4. Calculate CPU load for the ECU For ECU 2: 1. Draw a state machine diagram for each ECU component 2. Draw a state machine diagram for the ECU operation 3. Draw the sequence diagram for the ECU 4. Calculate CPU load for the ECU Calculate bus load in your system: With what percentage of system bus was busy per 1 second You should deliver a pdf file containing all your work and a video recording where you will discuss your work (maximum 5min long)

suggestions to Make Your Project Stand Out!

Tasks Pseudocode

Meets Specifications

For ECU 1:

1. Write Pseudocode for each ECU component

For ECU 2:

1. Write Pseudocode for each ECU component

You should deliver all ECUs components .c and .h files