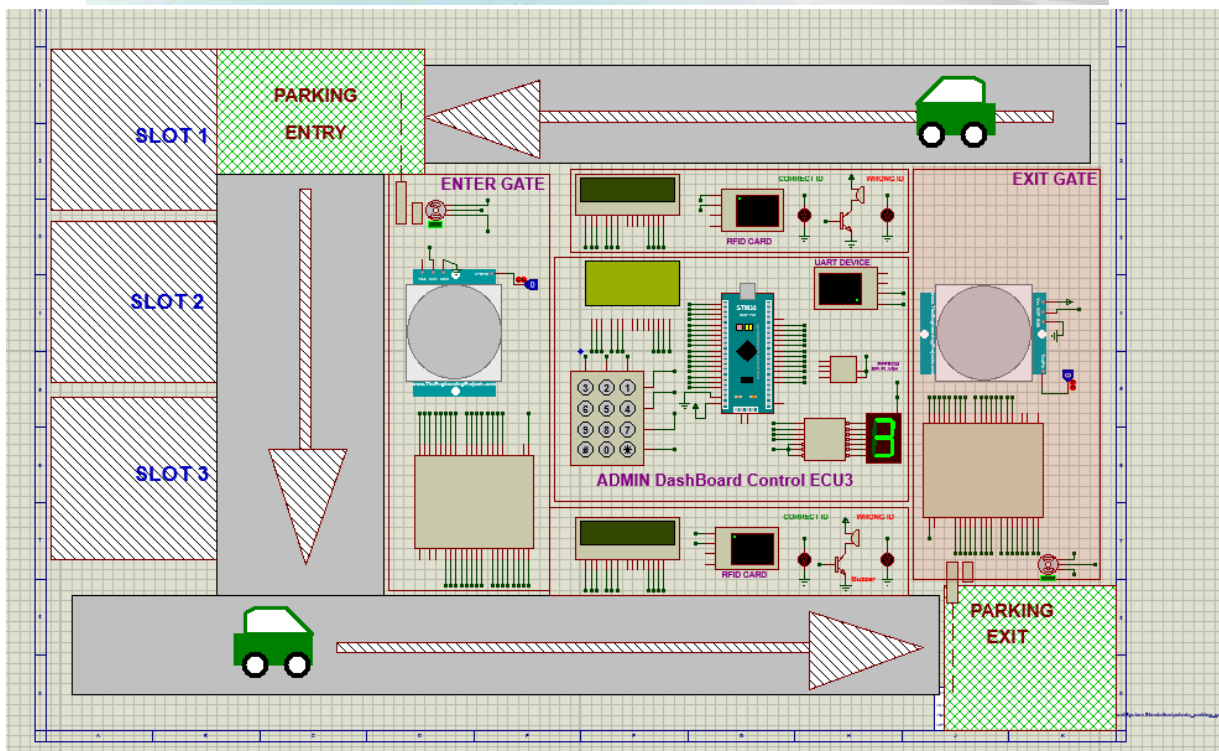


# Second \_Term\_ Final\_ Project

## (Smart\_ Vehicle\_ Parking\_ Project)



Eng: Hady Samir Abdelfattah

## Table of Contents

LIST OF FIGURES.....	3
Description .....	4
System Specifications.....	5
ECU1 .....	5
ECU2 .....	5
ECU3 .....	5
System Assumptions:.....	5
System Architecture.....	6
1- Case study .....	6
2- Method.....	6
3- Requirement.....	6
4- Space exploration/partitioning .....	7
5- System Analysis .....	8
i- Use Case Diagram .....	8
ii- Simple Activity Diagram.....	10
iii- Sequence Diagram (UML) .....	12
6- System Design .....	15

# LIST OF FIGURES

FIGURE 1:SYSTEM ARCHITECTURE	5
FIGURE 2:SYSTEM REQUIREMENT	5
FIGURE 3:SYSTEM PARTITIONING	6
FIGURE 4:ECU2 USE CASE DIAGRAM	8
FIGURE 5: ECU1 & ECU3 USE CASE DIAGRAM	9
FIGURE 6:ECU2 ACTIVITY DIAGRAM	10
FIGURE 7:ECU1 ACTIVITY DIAGRAM	10
FIGURE 8:ECU3 ACTIVITY DIAGRAM	11
FIGURE 9:ECU1 UML DIAGRAM	12
FIGURE 10:ECU2 UML DIAGRAM	13
FIGURE 11:ECU3 UML DIAGRAM	14
FIGURE 12:SYSTEM DESIGN	15

# Private Parking Garage

## Description

This Project aims to make a design for a private parking garage area for people in a specific area or garage for a company. This system is split into three ECUs:

### **First ECU for the entrance gate**

This ECU is responsible for the gate that is based on the servo motor, the RFID reader based on UART for the user interface, Buzzer for the beep sound when the driver enters an unauthorized ID, and Some LEDs like green and red for authorized and unauthorized ID.

### **Second ECU for admin dashboard**

This ECU is responsible for the admin privileges to add, delete, and edit driver data. The system may have more than one admin, each one of them has its username and password.

The ECU has an LCD and keypad as an admin interface, a UART device for entering admin data, and a seven-segment to display the number of available slots in the garage.

### **Third ECU for the exit gate**

This ECU is responsible for the gate that is based on the servo motor, the RFID reader based on UART for the user interface, Buzzer for the beep sound when the driver enters an unauthorized ID, and Some LEDs like green and red for authorized and unauthorized ID.

LCD will display some messages for the driver to determine what will do.

The Whole system is connected together. when a driver enters a valid ID the ECU1 sends data through SPI to ECU2, and the ECU2 starts checking if the ID is valid or not and checking if it is inside the garage and wants to exit from the entrance gate then the ECU2 return the result of checking to ECU1 to display the result of computing on the LCD is valid ID or invalid ID.

When the driver wants to exit the garage space and enter the ID the ECU3 sends the driver data to ECU2, and the ECU2 starts checking if the ID is valid or not and checking if it is outside the garage and wants to enter from the exit gate then the ECU2 return the result of checking to ECU1 to display the result of computing on the LCD is valid ID or invalid ID.

# System Specifications

## ECU1

- 1- Control The servo motor of the entrance gate.
- 2- Display the states on LCD

## ECU2

- 1- Holds Predefined admins data.
- 2- Validate the driver data.
- 3- Display admin dashboard.
- 4- Display number of available slots in garage.

## ECU3

- 1- Control The servo motor of the exit gate.
- 2- Display the states on LCD

## System Assumptions:

- 1- The Distance between the ECU1 and ECU2 is shorter than 50 cm.
- 2- The Distance between the ECU3 and ECU2 is shorter than 50 cm.
- 3- Sensors never fail.
- 4- Communication wires are never damaged.

# System Architecture

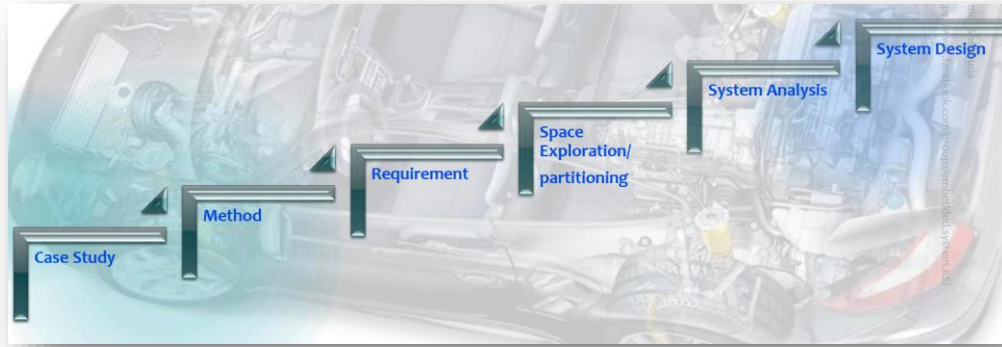


Figure 1: System Architecture

## 1- Case study

software that controls the private parking garage.

## 2- Method

Adaptive Technique: Agile Scrum Methodology

## 3- Requirement

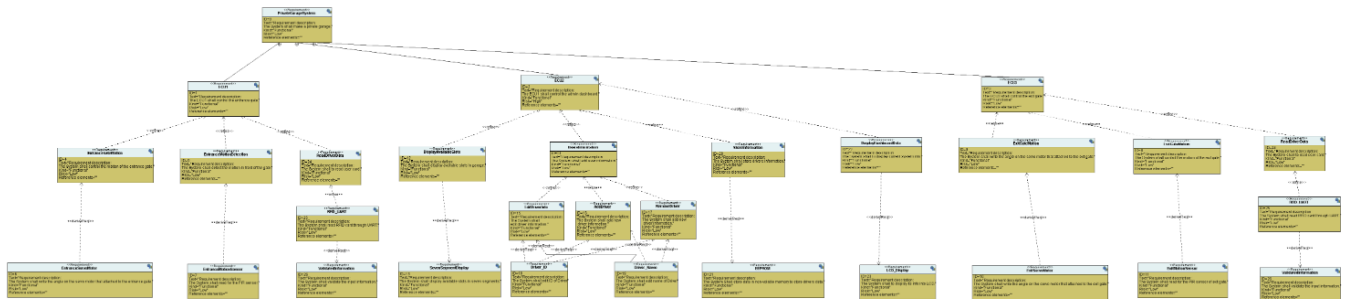


Figure 2: System Requirement

## 4- Space exploration/partitioning

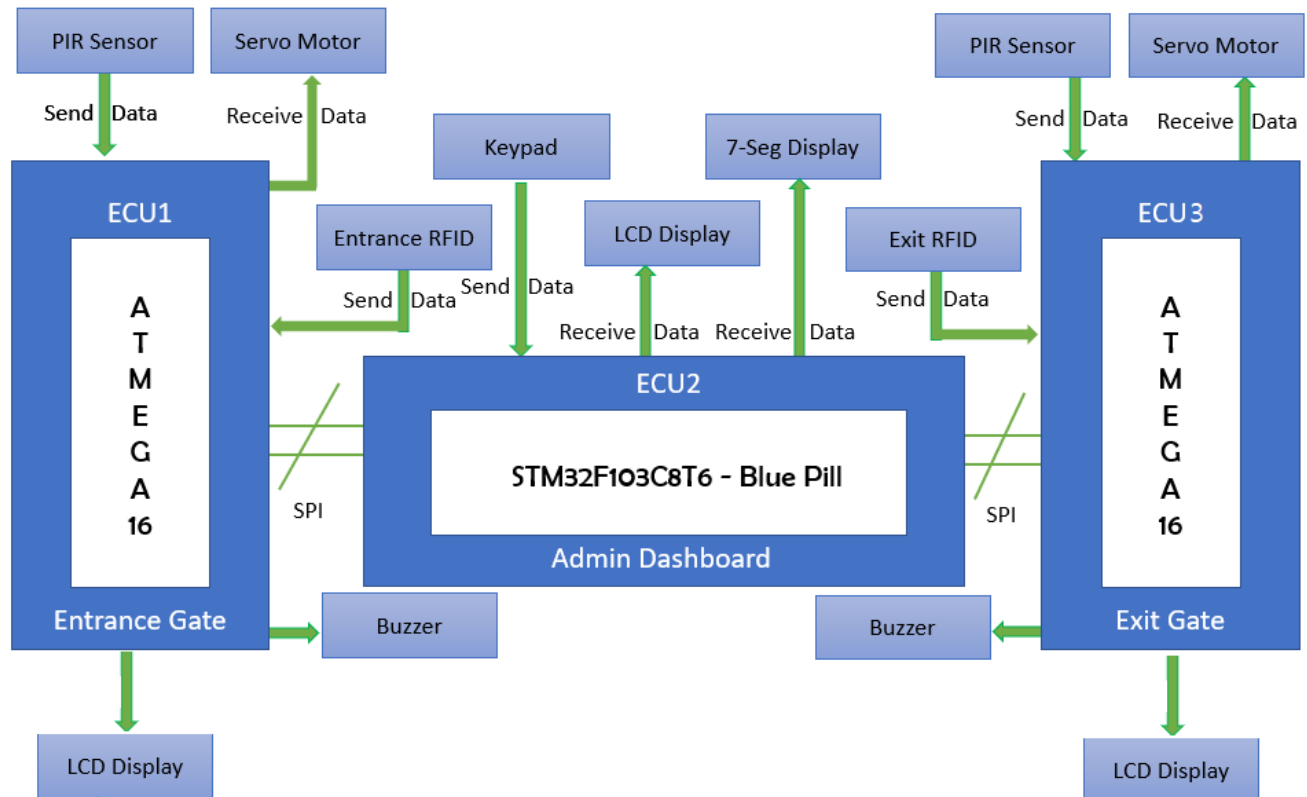


Figure 3: System Partitioning

microprocessor its specification

- 1- ARM 32-bit Cortex™-M3 CPU Core
  - i) 72 MHz maximum frequency
  - ii) Single-cycle multiplication and hardware division.
- 2- Memories
  - i) 32 Kbytes of Flash memory
  - ii) 10 Kbytes of SRAM
- 3- Clock, reset and supply management
  - i) 2.0 to 3.6 V application supply and I/Os.
  - ii) 4-to-16 MHz crystal oscillator.
  - iii) 32 kHz oscillator for RTC with calibration

And used **ATmega16** For ECU1 and ECU3

Program Memory Type	Flash
Program Memory Size (KB)	16
CPU Speed (MIPS/DMIPS)	16
Data EEPROM (bytes)	512

## 5- System Analysis

### i- Use Case Diagram

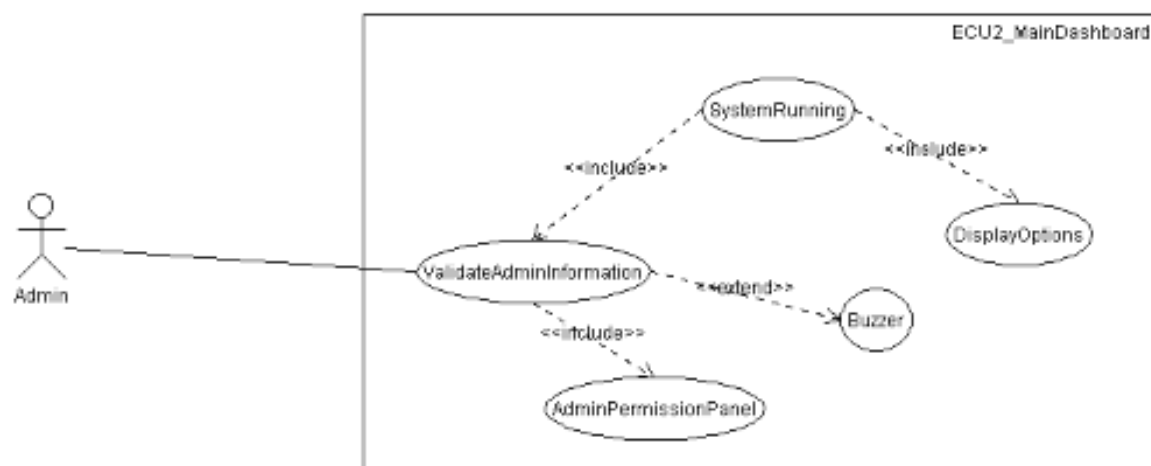


Figure 4:ECU2 Use Case Diagram



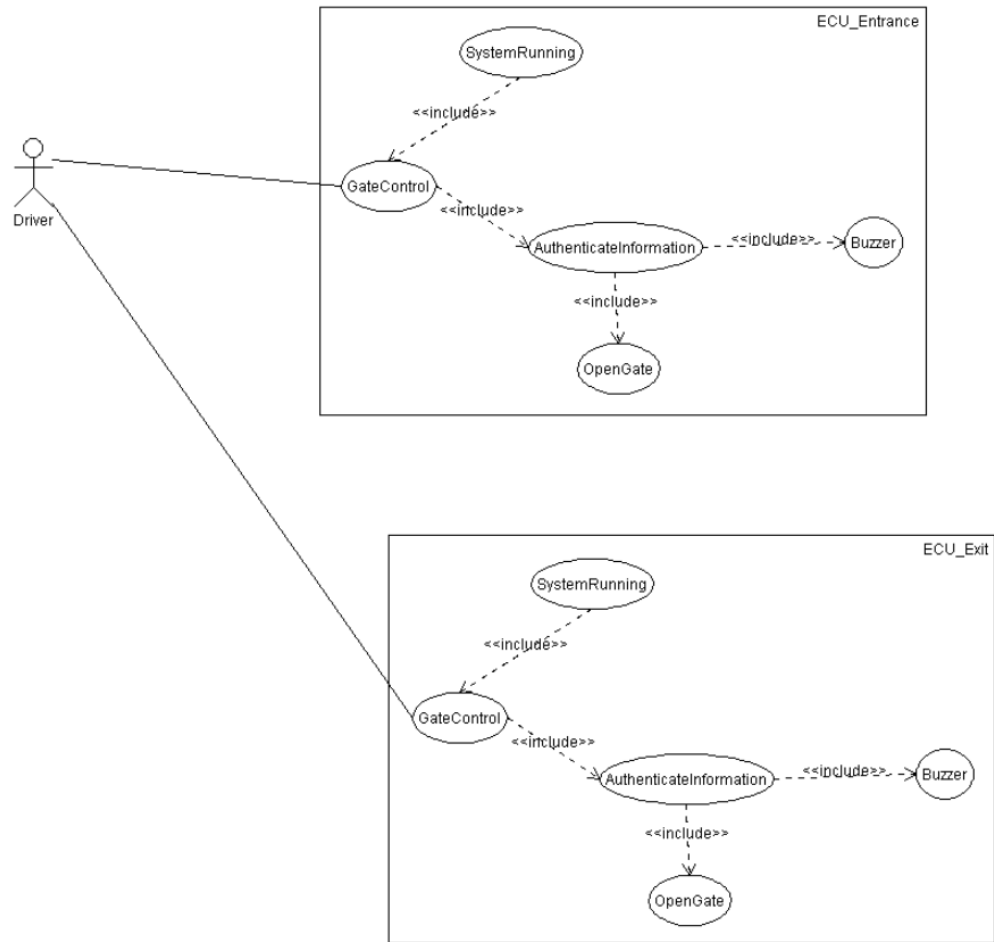


Figure 5: ECU1 & ECU3 Use Case Diagram

## ii- Simple Activity Diagram

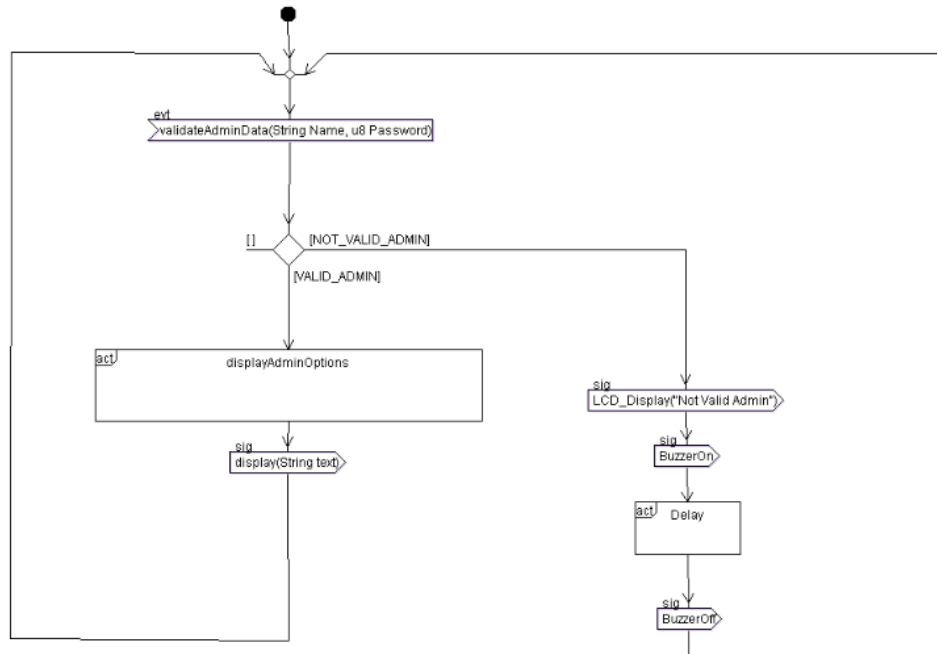


Figure 6:ECU2 Activity Diagram

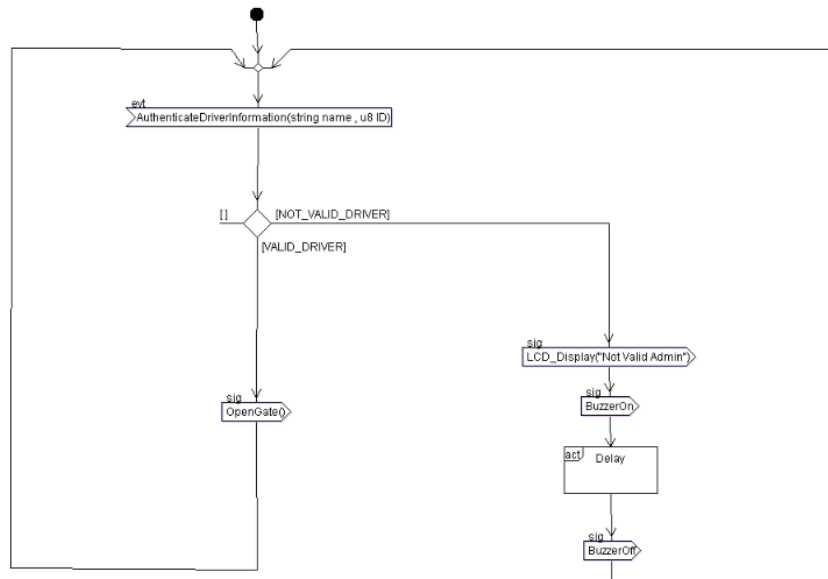


Figure 7:ECU1 Activity Diagram

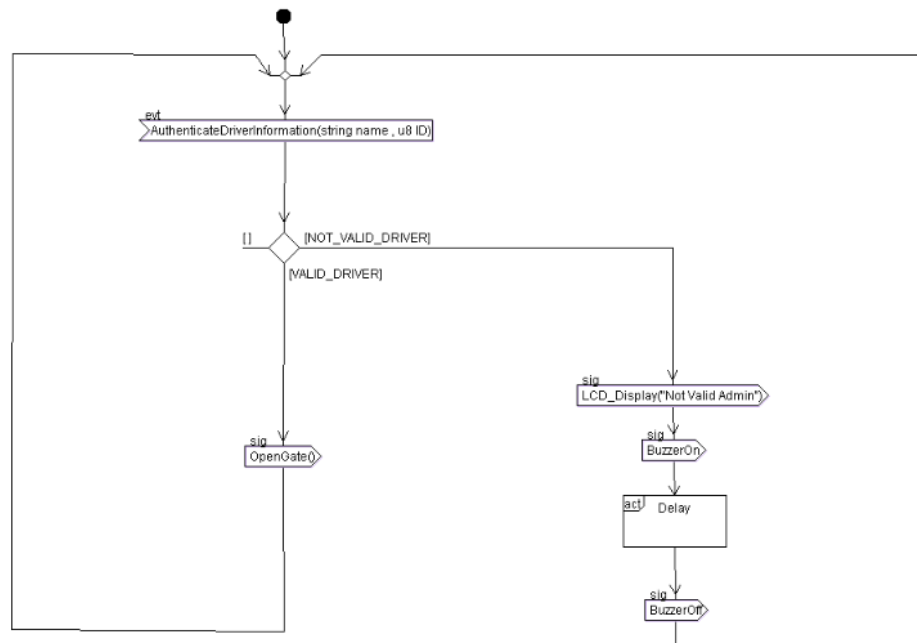


Figure 8:ECU3 Activity Diagram

### iii- Sequence Diagram (UML)

#### - ECU1 UML

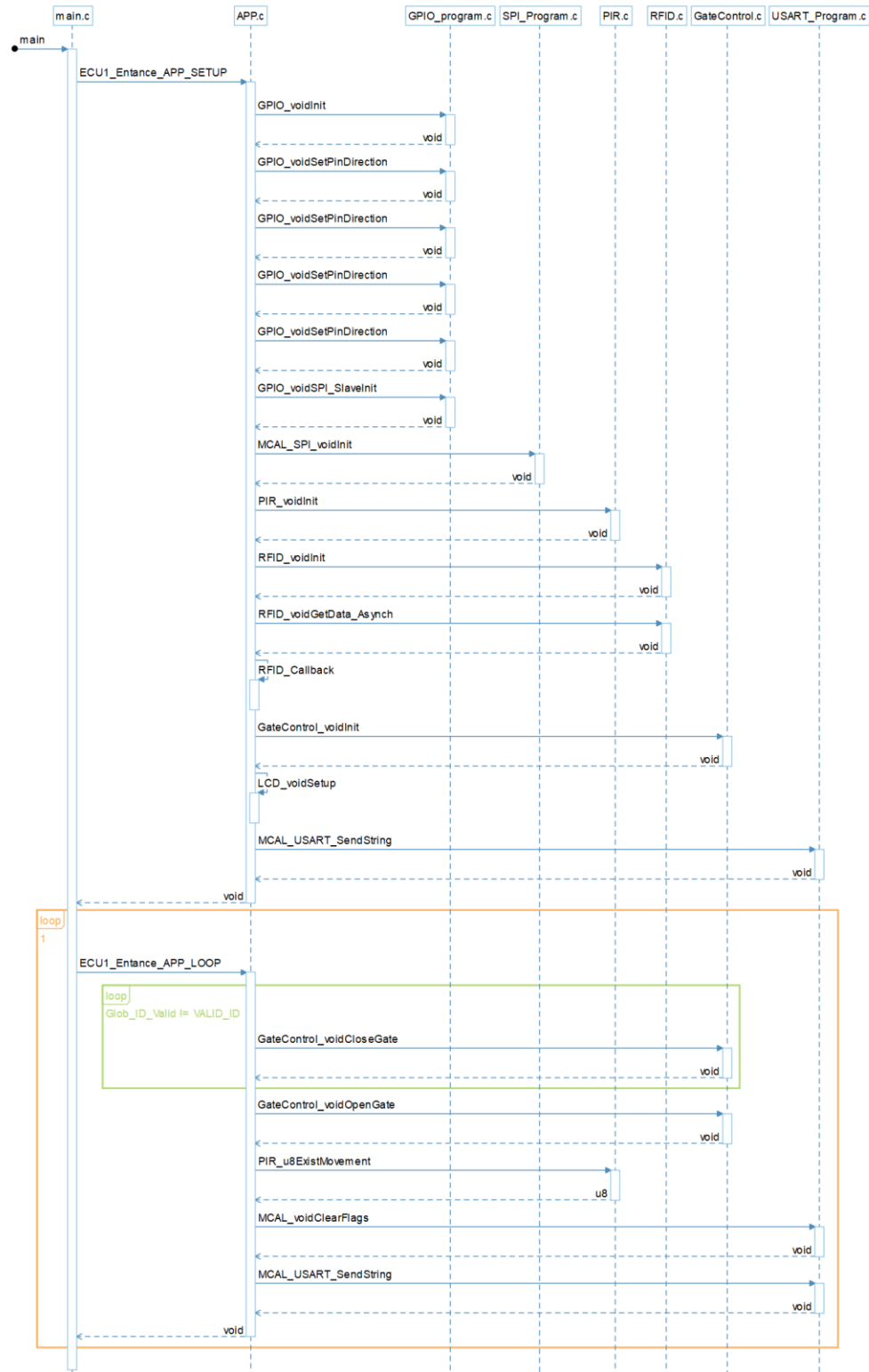


Figure 9:ECU1 UML Diagram

- ECU2 UML

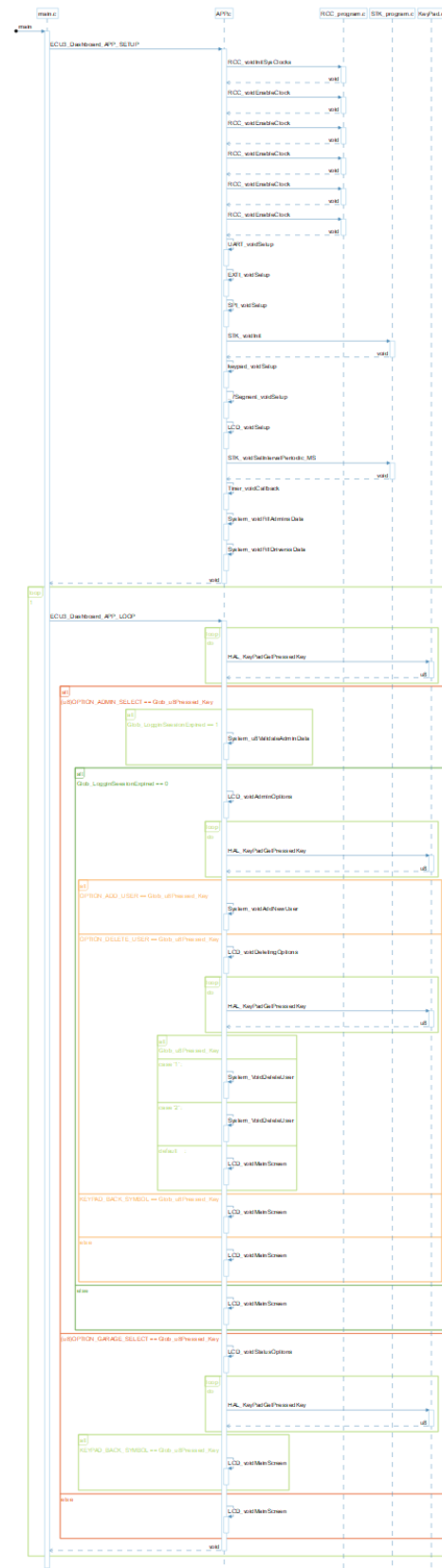


Figure 10:ECU2 UML Diagram

## - ECU3 UML

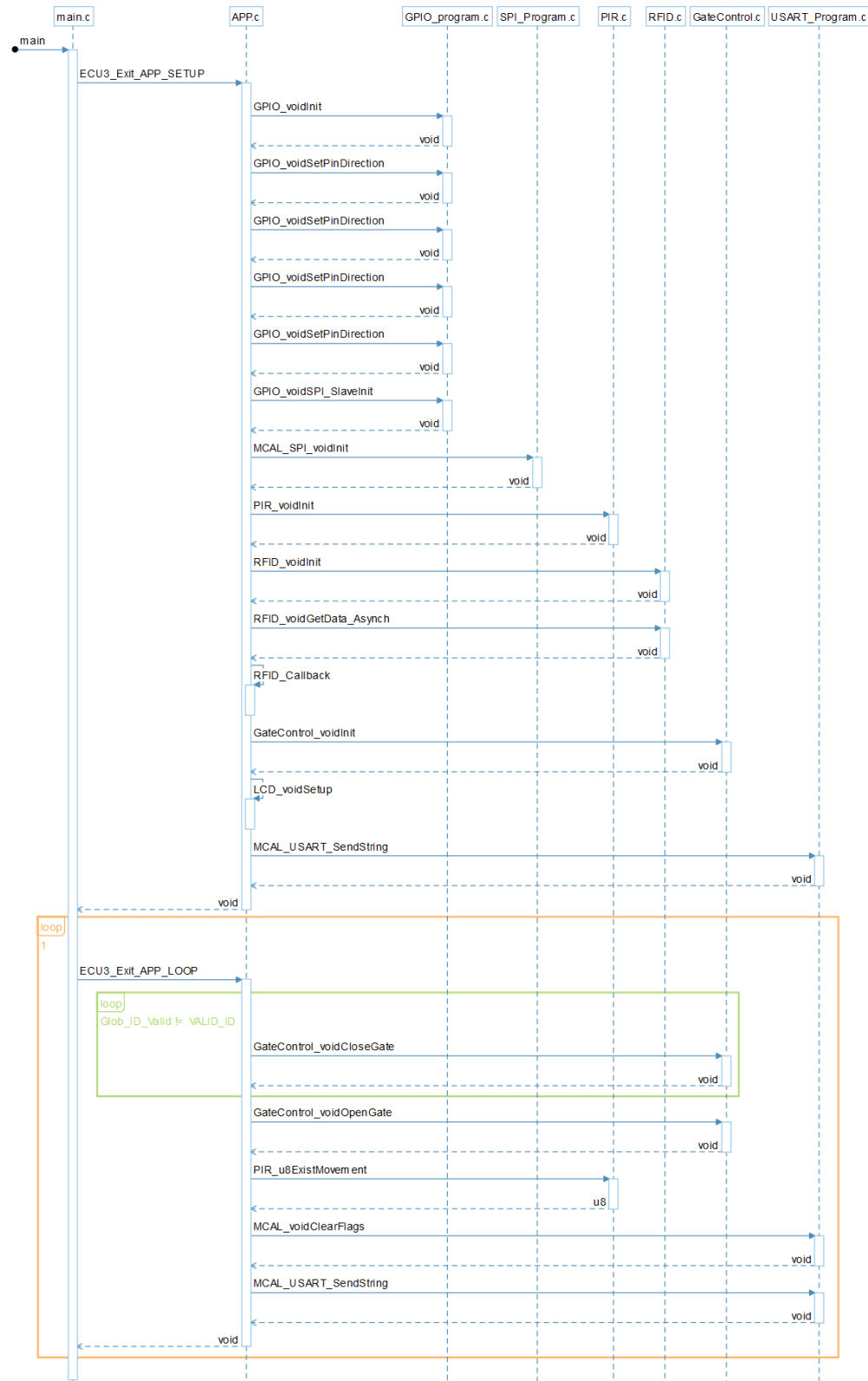


Figure 11:ECU3 UML Diagram

## 6- System Design

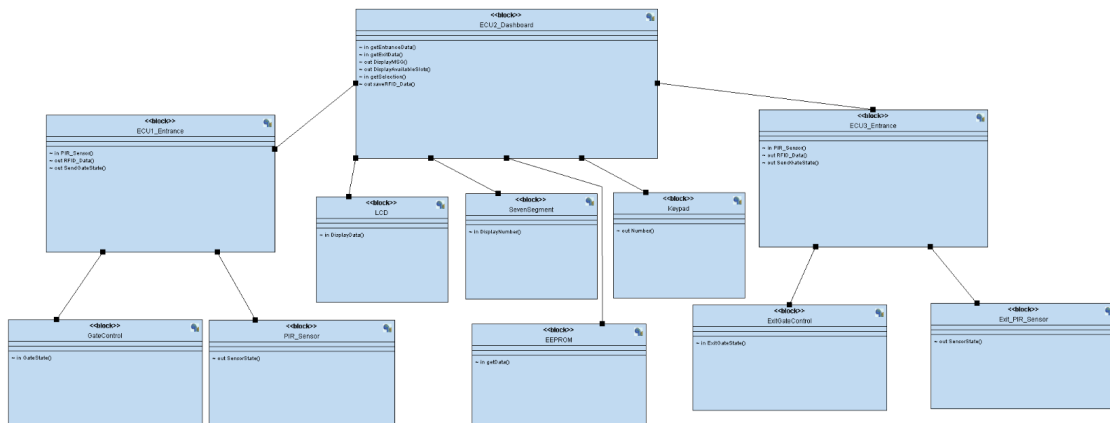


Figure 12: System Design