SMART Vehicle Parking Project

Mastering Embedded System Online Diploma

Eng. Hady Samir Abdelfattah

Profile Link

Project GitHub Repo Link:

Table of Contents

Private Parking Garage

Description

• The Project aims to make a design for a private parking garage for a company, this system is split into three ECUs (ECU1&3 ATmega32 - ECU2 STM32F103C6).

• ECU1&3: Control The servo motor of the entrance and exit gates& display the states on LCD.

• ECU1&3 drivers {MCAL(UART-SPI-PWM-GPIO-EXTI) – HAL(RFID-PIR-LCD-Servomotor)}.

• ECU2: Holds Predefined admins data, validate the driver data, Display admin dashboard and number of available slots in garage.

• ECU2 drivers {MCAL(USART-SPI-RCC-GPIO-EXTI) – HAL(7Segment-Keypad-LCD-Buzzer)}.

• Using TTool to design System Architecture {Requirements Diagram - System Analysis - System Design}.

• Using Jira software to apply Agile Scrum methodology to a project.

• Test the project by writing test cases

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- Controller maintenance is not modeled.
4- Sensors never fail.
5- Communication wires are never damaged.

System Architecture

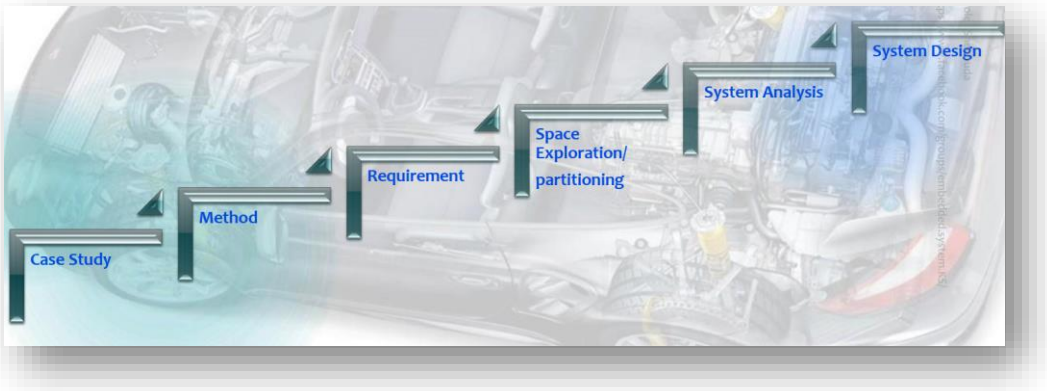

Figure 1:System Architecture

1- Case study

software that controls the private parking garage.

2- Method

Adaptive Technique: Agile Scrum Methodology

3- Requirement

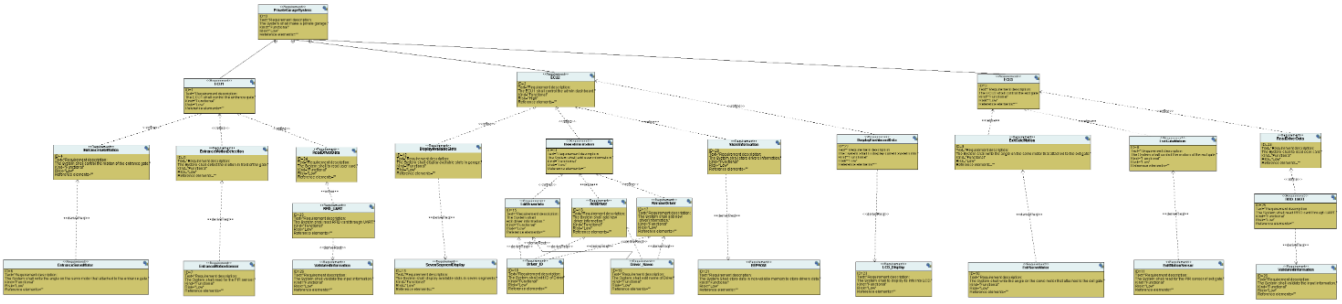

Figure 2:System Requirement

4- Space exploration/partitioning

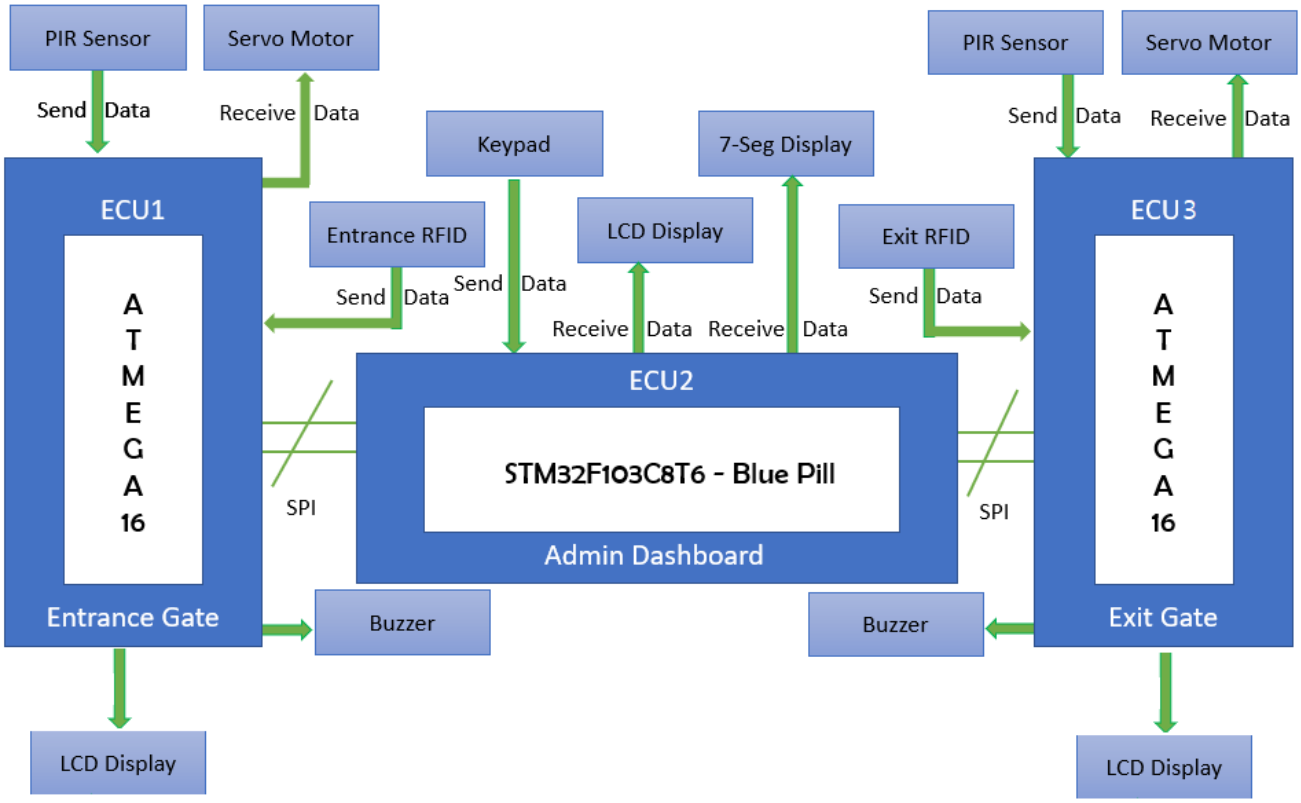

Figure 3:System Partitioning

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

5- System Analysis

i- Use Case Diagram

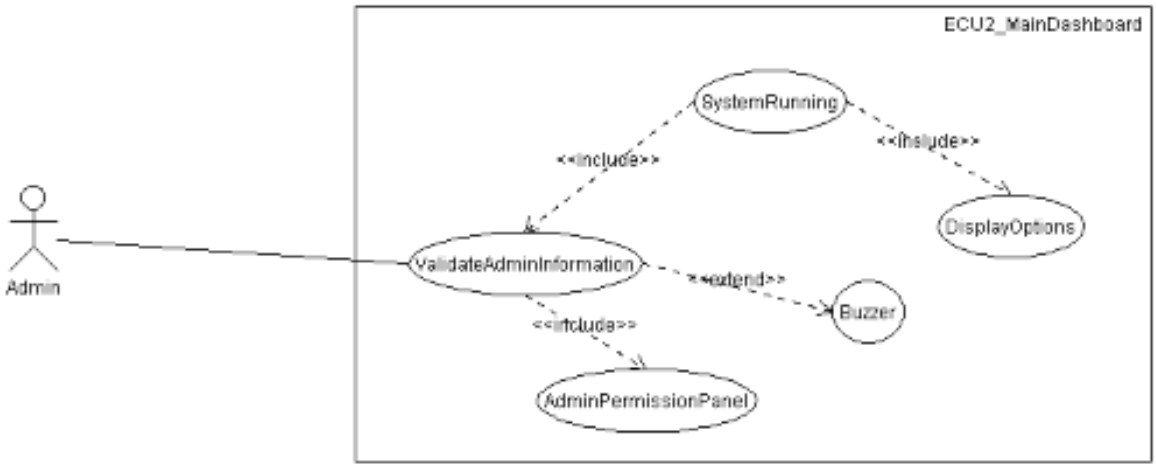

Figure 4:ECU2 Use Case Diagram

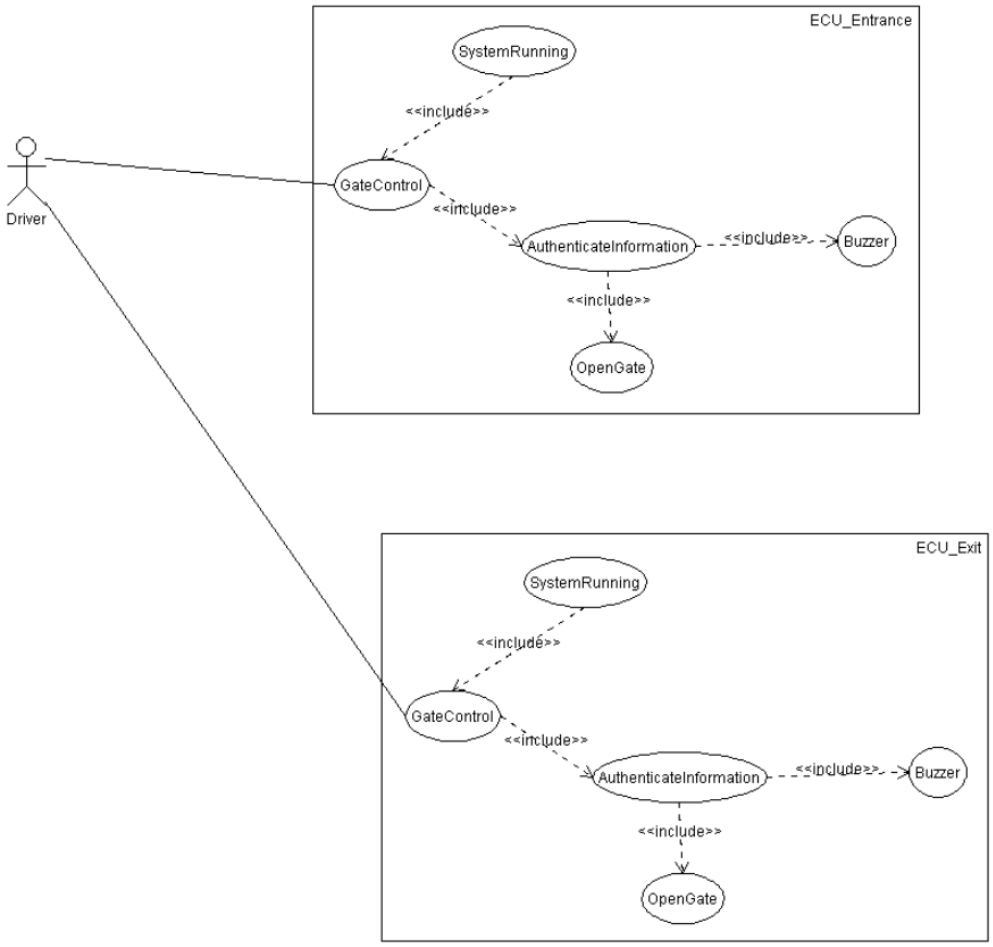

Figure 5: ECU1 & ECU3 Use Case Diagram

ii- Simple Activity Diagram

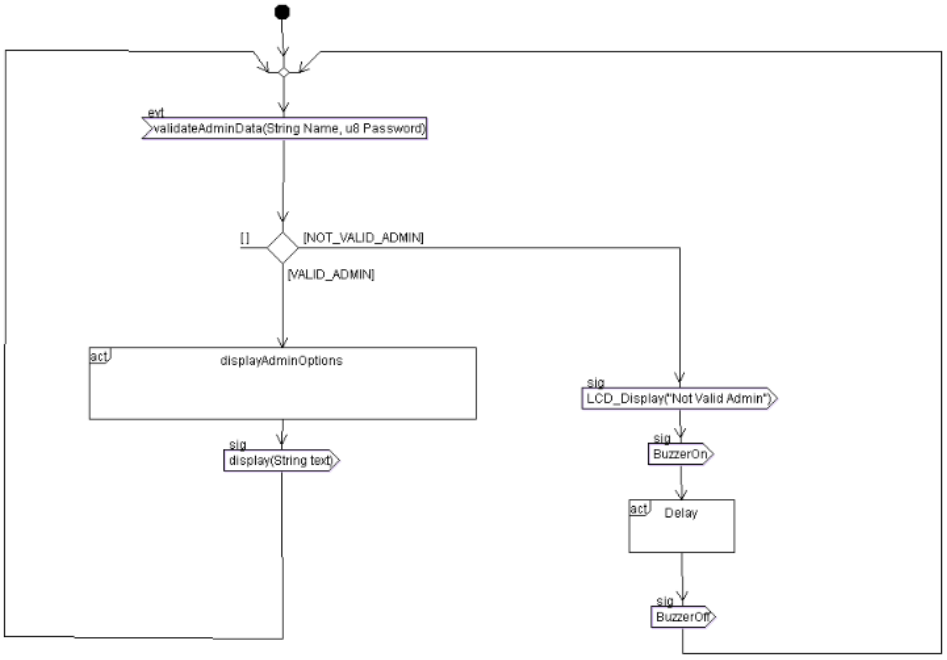

Figure 6:ECU2 Activity Diagram

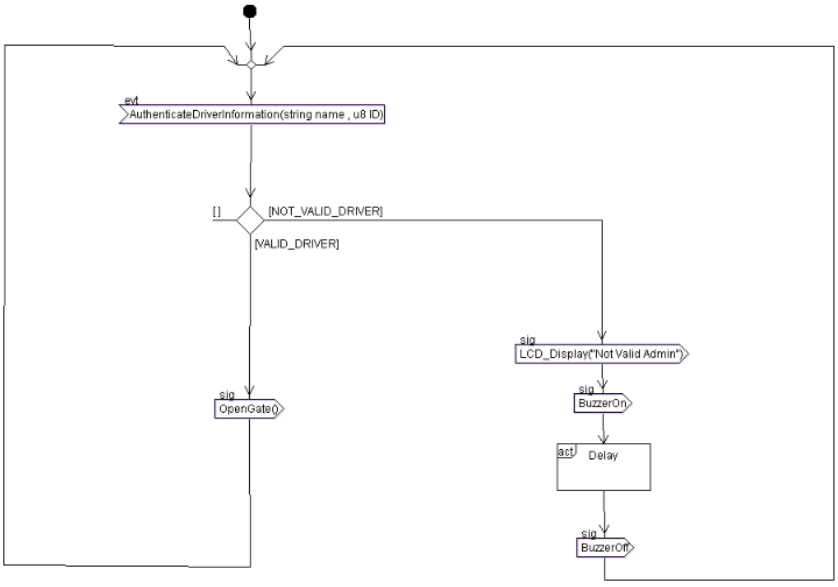

Figure 7:ECU1 Activity Diagram

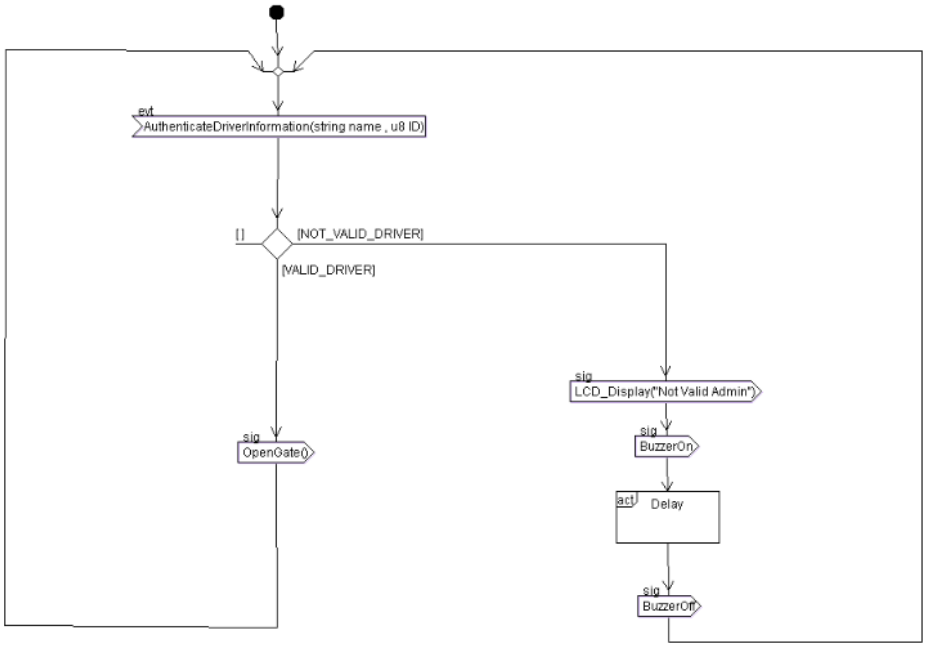

Figure 8:ECU3 Activity Diagram

iii- Sequence Diagram (UML)

- ECU1 UML

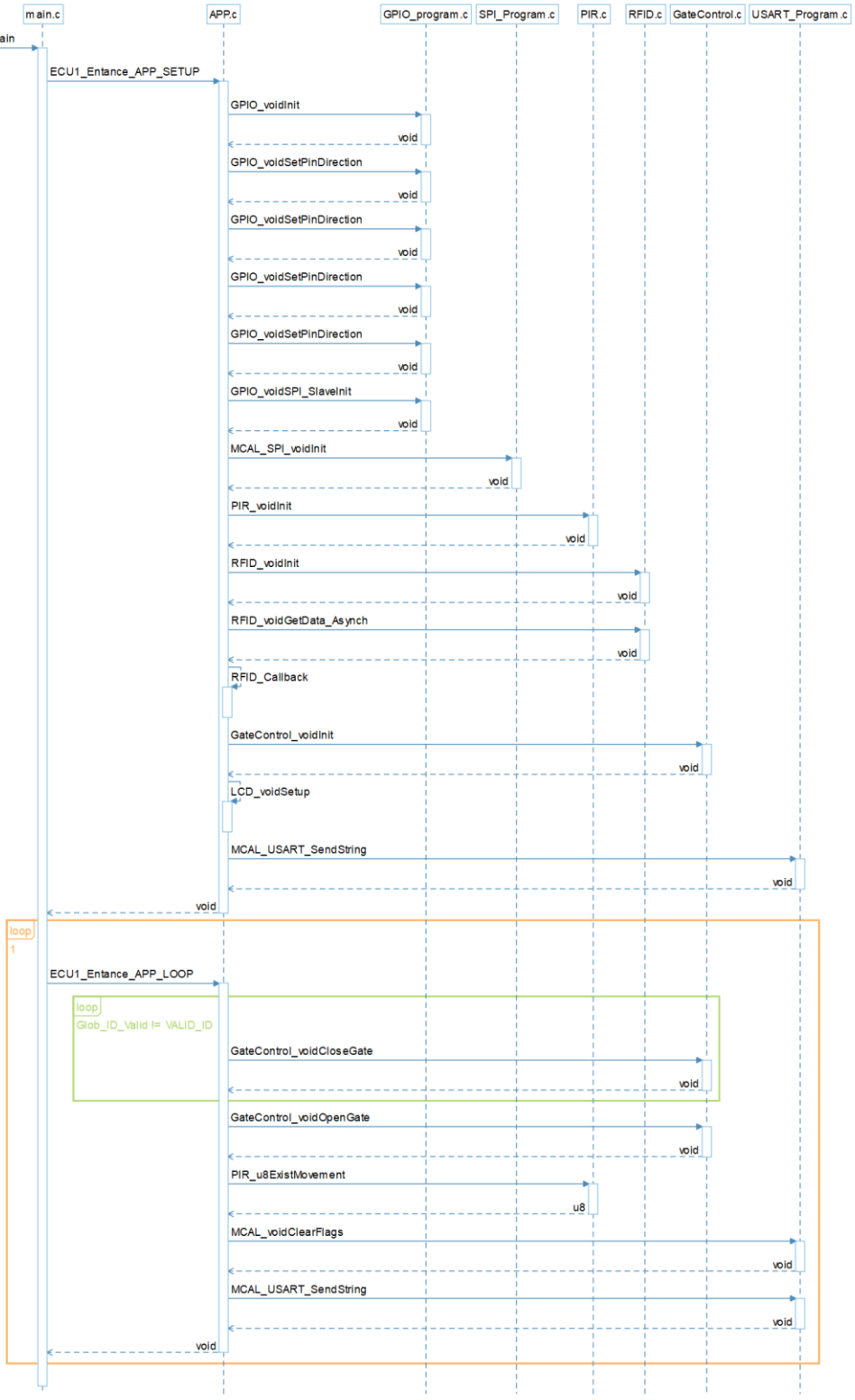

Figure 9:ECU1 UML Diagram

- ECU2 UML

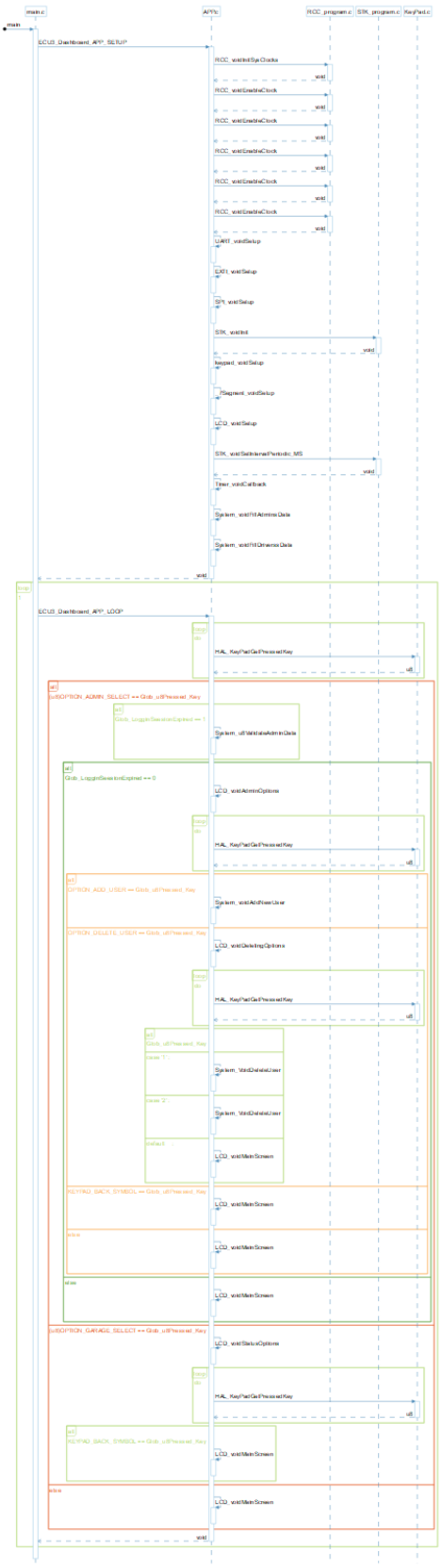

Figure 1:ECU2 UML Diagram

- ECU3 UML

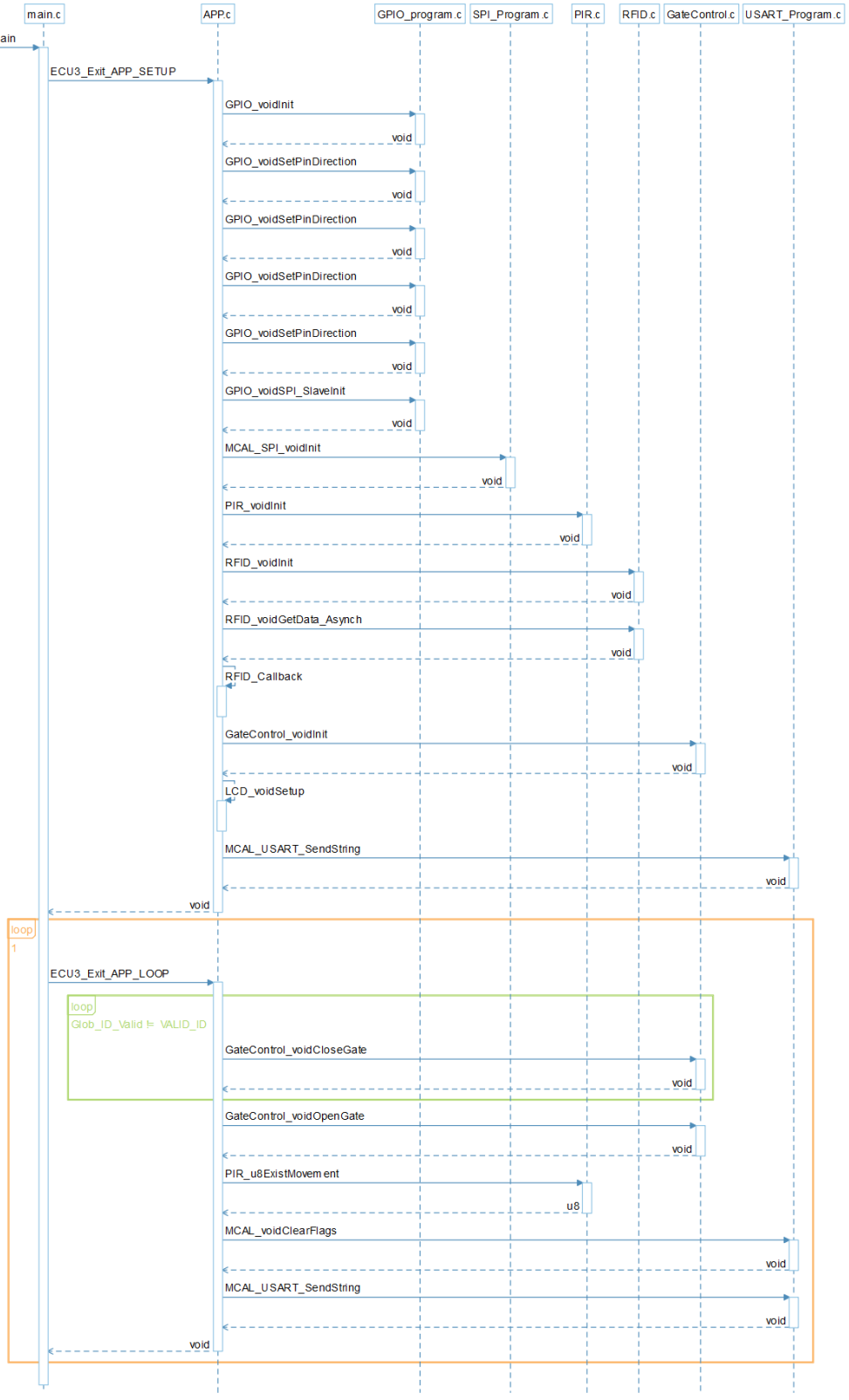

Figure 2:ECU3 UML Diagram

6- System Design

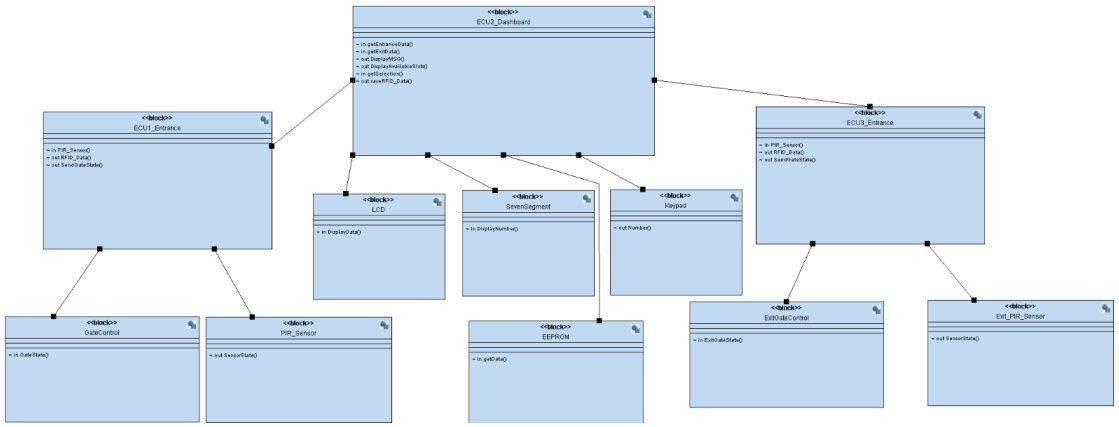

Figure 3:System Design

Hardware Simulation

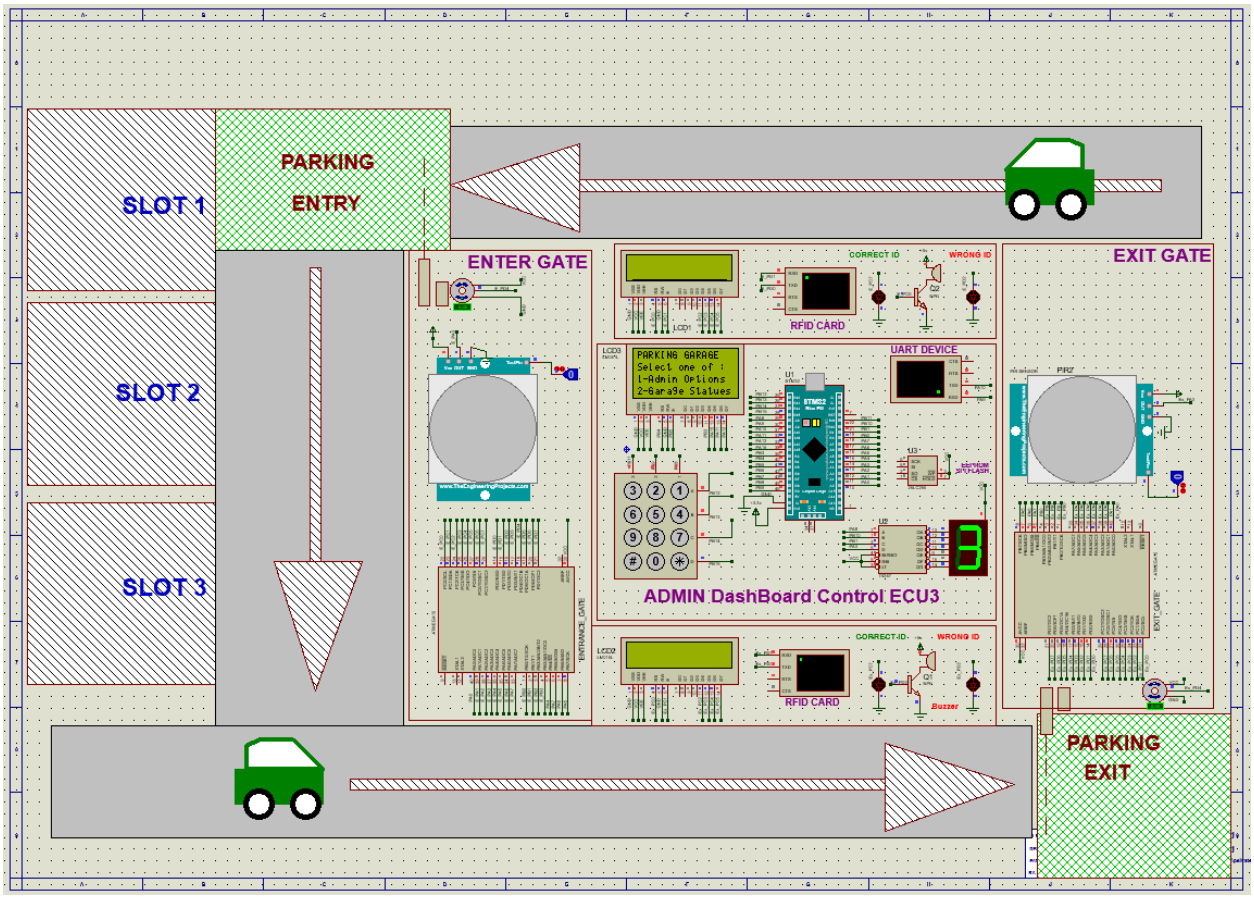

Figure 13:simulation test

Test Cases

- ECU1

Project Name:	**peivate_parking_garage**
Created by:	**Hady Samir Abdelfattah**
Reviewed By:	**Hady Samir Abdelfattah**
Date of creation:	**8/9/2023**

TEST SCENARIO	TEST CASE ID	TEST Title	PRE-CONDITION	TEST Type	TEST DATA	EXPECTED RESULT	ACTUAL RESULT	STATUS (PASS/ FAIL)
ECU1								
Validate functionality of RFID card reader (Entrnce Gate)	TC_RFID_001	Validate that RFID reader works well with a vaild data.	1-Atmel Studio 2- Proteus Simulation 3- RFID Driver 4- UART Driver	Uinit Test	username= "Hady" ID = "1234567"	"Driver Name: Hady "Driver ID: 1234567"	"Driver Name: Hady "Driver ID: 1234567"	Pass
	TC_RFID_002	Validate that behaviour of RFID Reader when enter username larger than	1-Atmel Studio 2- Proteus Simulation 3- RFID Driver 4- UART Driver	Uinit Test	username= "HadySamirAbdelfattah" ID = "1234567"	System will ignore any characters after the specified username length	"Driver Name: HadySamirA "Driver ID: 1234567"	Pass
	TC_RFID_003	Validate that behaviour of RFID Reader when enter ID larger than expected	1-Atmel Studio 2- Proteus Simulation 3- RFID Driver 4- UART Driver	Uinit Test	username= "Hady" ID = "12345678910"	System will ignore any characters after the specified ID length	"Driver Name: Hady "Driver ID: 1234567"	Pass
	TC_RFID_004	Validate that behaviour of RFID Reader when enter ID smaller than expected	1-Atmel Studio 2- Proteus Simulation 3- RFID Driver 4- UART Driver	Uinit Test	username= "Hady" ID = "123"	The system will wait until the length of the ID be in a pre-specified length	Nothing	Pass
Validate functionality of SPI Communication (ECU1)	TC_SPI_005	Validate that the can exchange data by SPI	1-STM32CUBE IDE 2- Proteus Simulation 3- RCC Driver 4- GPIO Driver	Uinit Test	Exchange '+' and 'A' between master and slave	a' at master '+' at slave	a' at master '+' at slave	Pass

Figure 14:ECU1 TEST CASES

Validate functionality of PIR Sensor (Entrnce Gate	TC_PIR_006	Validate that the PIR works well when finds motion	1-Atmel Studio 2- Proteus Simulation 3- GPIO Driver 4- PIR Driver	Uinit Test	Exist Motion	Turn on led	Led is on	Pass
	TC_PIR_007	Validate that the PIR works well when there is no motion	1-Atmel Studio 2- Proteus Simulation 3- GPIO Driver 4- PIR Driver	Uinit Test	No Motion	Turn off led	Led is off	Pass
Validate functionality of sending RFID data through SPI (Entrnce Gate) Communication	TC_Gate_008	Validate that the gate will open when ID of Driver is valid.	1-Atmel Studio 2- Proteus Simulation 3- RFID Driver 4- UART Driver	Functional Test	Valid username= "Hady" Valid ID = "1234567" PIR Reads Exist Motion	Gate will open and never closed	Gate will open and never closed	Pass
	TC_Gate_009	Validate that the gate will open when ID of Driver is valid and close after that	1-Atmel Studio 2- Proteus Simulation 3- RFID Driver 4- UART Driver	Functional Test	Valid username= "Hady" Valid ID = "1234567" PIR Reads No Motion	Gate will open till car fully entered the garage	Gate will open till car fully entered the garage	Pass
	TC_Gate_010	Validate that the gate will not open when ID of Driver is invalid.	1-Atmel Studio 2- Proteus Simulation 3- RFID Driver 4- UART Driver	Functional Test	InValid username= "Mody" Valid ID = "1234567" PIR Reads No Motion	The Gate Will Never Open	The Gate is closed	Pass
	TC_Gate_011	Validate that the gate will not open when ID of Driver is invalid.	1-Atmel Studio 2- Proteus Simulation 3- RFID Driver 4- UART Driver	Functional Test	InValid username= "Hady" InValid ID = "0120233" PIR Reads No Motion	The Gate Will Never Open	The Gate is closed	Pass
Validate functionality of LCD and Buzzer	TC_LCD_Buzzer_012	Validate that the Valid ID Message	1-Atmel Studio 2- Proteus Simulation 3- RFID Driver 4- UART Driver	Functional Test	Valid username= "Hady" Valid ID = "1234567"	Yor ID is Valid	Yor ID is Valid	Pass
	TC_LCD_Buzzer_013	Validate that the InValid ID Message	1-Atmel Studio 2- Proteus Simulation 3- RFID Driver 4- UART Driver	Functional Test	InValid username= "Mody" Valid ID = "1234567"	InValid ID and Buzzer Works	InValid ID and Buzzer Works	Pass

Figure 15:ECU1 TEST CASES

- ECU2

ECU2								
validate functionality of GPIO and RCC	TC_GPIO_001	Validate that the GPIO Works as output	1-STM32CUBE IDE 2- Proteus Simulation 3- RCC Driver 4- GPIO Driver	Uinit Test	Connect led with MCU and Ground	LED will be on when MCU dirve high	LED turned on when MCU dirve high	Pass
	TC_GPIO_002	Validate that the GPIO Works as input	1-STM32CUBE IDE 2- Proteus Simulation 3- RCC Driver 4- GPIO Driver	Uinit Test	Connect led with MCU and Vcc	LED will be on when push button has be pressed	LED turned on when push button was pressed	Pass
validate functionality of Buzzer	TC_Buzzer_003	Validate that the Buzzer Works when apply high signal	1-STM32CUBE IDE 2- Proteus Simulation 3- RCC Driver 4- GPIO Driver	Uinit Test	Apply High signal	Buzzer turn on	Buzzer turn on	Pass
	TC_Buzzer_004	Validate that the Buzzer Works when apply low signal	1-STM32CUBE IDE 2- Proteus Simulation 3- RCC Driver 4- GPIO Driver	Uinit Test	Apply Low signal	Buzzer turn off	Buzzer turn off	Pass
validate functionality of 7-Segment	TC_7-Segment_005	Validate that the 7-Segment Works while sending number	1-STM32CUBE IDE 2- Proteus Simulation 3- RCC Driver 4- GPIO Driver	Uinit Test	sending number = 2	display 2	display 2	Pass
validate functionality of Keypad	TC_Keypad_006	Validate that the Keypad Works well	1-STM32CUBE IDE 2- Proteus Simulation 3- RCC Driver 4- GPIO Driver	Uinit Test	sending number 2 using KeyPad	MCU Reads 2	MCU Reads 3	Pass
validate functionality of LCD	TC_LCD_007	Validate that the Keypad Works well with strings,numbers and special char	1-STM32CUBE IDE 2- Proteus Simulation 3- RCC Driver 4- GPIO Driver	Uinit Test	sending "Hady123@*"	LCD display "Hady123@*"	LCD display "Hady123@*"	Pass
Validate functionality of SPI Communication (ECU2)	TC_SPI_008	Validate that the can exchange data by SPI	1-STM32CUBE IDE 2- Proteus Simulation 3- RCC Driver 4- GPIO Driver	Uinit Test	Exchange '+' and 'A' between master and slave	a' at master '+' at slave	a' at master '+' at slave	Pass

Figure 16:ECU2 TEST CASES

validate functionality of Admin	TC_ADMIN_009	Validate that Admin Can Login	1-STM32CUBE IDE 2- Proteus Simulation 3- RCC Driver 4- GPIO Driver	Functional Test	Admin username= "Hady" Pass = "1234567"	Successful login	Successful login	Pass
	TC_ADMIN_010	Validate that Admin Can't Login if username is invalid	1-STM32CUBE IDE 2- Proteus Simulation 3- RCC Driver 4- GPIO Driver	Functional Test	Admin username= "Mooo" Pass = "1234567"	Unsuccessful login	Unsuccessful login	Pass
	TC_ADMIN_011	Validate that Admin Can't Login if Password is invalid	1-STM32CUBE IDE 2- Proteus Simulation 3- RCC Driver 4- GPIO Driver	Functional Test	Admin username= "Hady" Pass = "0000000"	Unsuccessful login	Unsuccessful login	Pass
	TC_ADMIN_012	Validate that Admin Can Add New Driver	1-STM32CUBE IDE 2- Proteus Simulation 3- RCC Driver 4- GPIO Driver	Functional Test	Admin username= "Hady" Pass = "1234567" Driver Data "Mo" "1231231"	Done	Done	Pass
	TC_ADMIN_013	Validate that Admin Can Delete a Driver	1-STM32CUBE IDE 2- Proteus Simulation 3- RCC Driver 4- GPIO Driver	Functional Test	Admin username= "Hady" Pass = "1234567" Driver Data "Heba" "1234567"	Successful	Successful	Pass
	TC_ADMIN_014	Validate that Admin Can Delete all Driver	1-STM32CUBE IDE 2- Proteus Simulation 3- RCC Driver 4- GPIO Driver	Functional Test	Admin username= "Hady" Pass = "1234567"	Successful	Successful	Pass

Figure 17:ECU2 TEST CASES

- ECU3

ECU3								
Validate functionality of RFID card reader (Entrnce Gate)	TC_RFID_001	Validate that RFID reader works well with a vaild data.	1-Atmel Studio 2- Proteus Simulation 3- RFID Driver 4- UART Driver	Uinit Test	username= "Hady" ID = "1234567"	"Driver Name: Hady "Driver ID: 1234567"	"Driver Name: Hady "Driver ID: 1234567"	Pass
	TC_RFID_002	Validate that behaviour of RFID Reader when enter username larger than	1-Atmel Studio 2- Proteus Simulation 3- RFID Driver 4- UART Driver	Uinit Test	username= "HadySamirAbdelfattah" ID = "1234567"	System will ignore any characters after the specified username length	"Driver Name: HadySamirA "Driver ID: 1234567"	Pass
	TC_RFID_003	Validate that behaviour of RFID Reader when enter ID larger than expected	1-Atmel Studio 2- Proteus Simulation 3- RFID Driver 4- UART Driver	Uinit Test	username= "Hady" ID = "12345678910"	System will ignore any characters after the specified ID length	"Driver Name: Hady "Driver ID: 1234567"	Pass
	TC_RFID_004	Validate that behaviour of RFID Reader when enter ID smaller than expected	1-Atmel Studio 2- Proteus Simulation 3- RFID Driver 4- UART Driver	Uinit Test	username= "Hady" ID = "123"	The system will wait until the length of the ID be in a pre-specified length	Nothing	Pass
Validate functionality of SPI Communication (ECU1)	TC_SPI_005	Validate that the can exchange data by SPI	1-STM32CUBE IDE 2- Proteus Simulation 3- RCC Driver 4- GPIO Driver	Uinit Test	Exchange '+' and 'A' between master and slave	a' at master '+' at slave	a' at master '+' at slave	Pass
Validate functionality of PIR Sensor (Entrnce Gate	TC_PIR_006	Validate that the PIR works well when finds motion	1-Atmel Studio 2- Proteus Simulation 3- GPIO Driver 4- PIR Driver	Uinit Test	Exist Motion	Turn on led	Led is on	Pass
	TC_PIR_007	Validate that the PIR works well when there is no motion	1-Atmel Studio 2- Proteus Simulation 3- GPIO Driver 4- PIR Driver	Uinit Test	No Motion	Turn off led	Led is off	Pass

Figure 18 ECU3 TEST CASES

Validate functionality of sending RFID data through SPI (Entrnce Gate) Communication	TC_Gate_008	Validate that the gate will open when ID of Driver is valid.	1-Atmel Studio 2- Proteus Simulation 3- RFID Driver 4- UART Driver	Functional Test	Valid username= "Hady" Valid ID = "1234567" PIR Reads Exist Motion	Gate will open and never closed	Gate will open and never closed	Pass
	TC_Gate_009	Validate that the gate will open when ID of Driver is valid and close after that	1-Atmel Studio 2- Proteus Simulation 3- RFID Driver 4- UART Driver	Functional Test	Valid username= "Hady" Valid ID = "1234567" PIR Reads No Motion	Gate will open till car fully entered the garage	Gate will open till car fully entered the garage	Pass
	TC_Gate_010	Validate that the gate will not open when ID of Driver is invalid.	1-Atmel Studio 2- Proteus Simulation 3- RFID Driver 4- UART Driver	Functional Test	InValid username= "Mody" Valid ID = "1234567" PIR Reads No Motion	The Gate Will Never Open	The Gate is closed	Pass
	TC_Gate_011	Validate that the gate will not open when ID of Driver is invalid.	1-Atmel Studio 2- Proteus Simulation 3- RFID Driver 4- UART Driver	Functional Test	InValid username= "Hady" InValid ID = "0120233" PIR Reads No Motion	The Gate Will Never Open	The Gate is closed	Pass
Validate functionality of LCD and Buzzer	TC_LCD_Buzzer_012	Validate that the Valid ID Message	1-Atmel Studio 2- Proteus Simulation 3- RFID Driver 4- UART Driver	Functional Test	Valid username= "Hady" Valid ID = "1234567"	Yor ID is Valid	Yor ID is Valid	Pass
	TC_LCD_Buzzer_013	Validate that the InValid ID Message	1-Atmel Studio 2- Proteus Simulation 3- RFID Driver 4- UART Driver	Functional Test	InValid username= "Mody" Valid ID = "1234567"	InValid ID and Buzzer Works	InValid ID and Buzzer Works	Pass

Figure 19 ECU2 TEST CASES