Mastering Embedded System Online Diploma

<https://www.learn-in-depth.com/>

Second Term (Final Project)

Eng. Mohamed Abd El-Naby Mohamed

My Profile:

<https://www.learn-in-depth.com/online-diploma/mahameda.naby@gmail.com>

Table of Contents

[LIST OF FIGURES 4](#_Toc122465935)

[Description 5](#_Toc122465936)

[System Specifications 6](#_Toc122465937)

[ECU1 6](#_Toc122465938)

[ECU2 6](#_Toc122465939)

[ECU3 6](#_Toc122465940)

[System Assumptions: 6](#_Toc122465941)

[Project Overview: 7](#_Toc122465942)

[i- All System 7](#_Toc122465943)

[ii- Calls 7](#_Toc122465944)

[System Architecture 9](#_Toc122465945)

[1- Case study 9](#_Toc122465946)

[2- Method 9](#_Toc122465947)

[3- Requirement 9](#_Toc122465948)

[4- Space exploration/partitioning 10](#_Toc122465949)

[5- System Analysis 11](#_Toc122465950)

[i- Use Case Diagram 11](#_Toc122465951)

[ii- Simple Activity Diagram 13](#_Toc122465952)

[iii- Sequence Diagram (UML) 15](#_Toc122465953)

[6- System Design 18](#_Toc122465954)

[Boot Sequence of STM32 18](#_Toc122465955)

[Map File 19](#_Toc122465956)

[Memory dump 20](#_Toc122465957)

[Some of Symbols 21](#_Toc122465958)

[ELF image details 22](#_Toc122465959)

[Hardware Simulation 23](#_Toc122465960)

[Test Cases 23](#_Toc122465961)

[- ECU1 23](#_Toc122465962)

[- ECU2 25](#_Toc122465963)

[- ECU1 27](#_Toc122465964)

# LIST OF FIGURES

[Figure 1:Project Overview 7](#_Toc122465965)

[Figure 2:ECU1 Calls 7](#_Toc122465966)

[Figure 3:ECU2 Calls 8](#_Toc122465967)

[Figure 4:ECU3 Calls 8](#_Toc122465968)

[Figure 5:System Architecture 9](#_Toc122465969)

[Figure 6:System Requirement 9](file:///C:\Users\Abnaby\Desktop\pressureControl.docx#_Toc122465970)

[Figure 7:System Partitioning 10](file:///C:\Users\Abnaby\Desktop\pressureControl.docx#_Toc122465971)

[Figure 8:ECU2 Use Case Diagram 11](#_Toc122465972)

[Figure 9: ECU1 & ECU3 Use Case Diagram 12](#_Toc122465973)

[Figure 10:ECU2 Activity Diagram 13](#_Toc122465974)

[Figure 11:ECU1 Activity Diagram 13](#_Toc122465975)

[Figure 12:ECU3 Activity Diagram 14](#_Toc122465976)

[Figure 13:ECU1 UML Diagram 15](#_Toc122465977)

[Figure 14:ECU2 UML Diagram 16](#_Toc122465978)

[Figure 15:ECU3 UML Diagram 17](#_Toc122465979)

[Figure 16:System Design 18](#_Toc122465980)

[Figure 17:Sequence 19](#_Toc122465981)

[Figure 18:Vector table position in map file 19](#_Toc122465982)

[Figure 19:Memory Dump with debug section 20](#_Toc122465983)

[Figure 20:Symbols of ELF image 21](#_Toc122465984)

[Figure 21:ELF image details 22](#_Toc122465985)

[Figure 22:simulation test 23](#_Toc122465986)

[Figure 23:ECU1 TEST CASES 23](#_Toc122465987)

[Figure 24:ECU1 TEST CASES 24](#_Toc122465988)

[Figure 25:ECU1 TEST CASES 24](#_Toc122465989)

[Figure 26:ECU1 TEST CASES 24](#_Toc122465990)

[Figure 27:ECU2 TEST CASES 25](#_Toc122465991)

[Figure 28:ECU2 TEST CASES 25](#_Toc122465992)

[Figure 29:ECU2 TEST CASES 26](#_Toc122465993)

[Figure 30:ECU2 TEST CASES 26](#_Toc122465994)

[Figure 31:ECU3 TEST CASES 27](#_Toc122465995)

[Figure 32:ECU3 TEST CASES 27](#_Toc122465996)

[Figure 33:ECU3 TEST CASES 27](#_Toc122465997)

[Figure 34:ECU3 TEST CASES 28](#_Toc122465998)

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

# Project Overview:

## All System

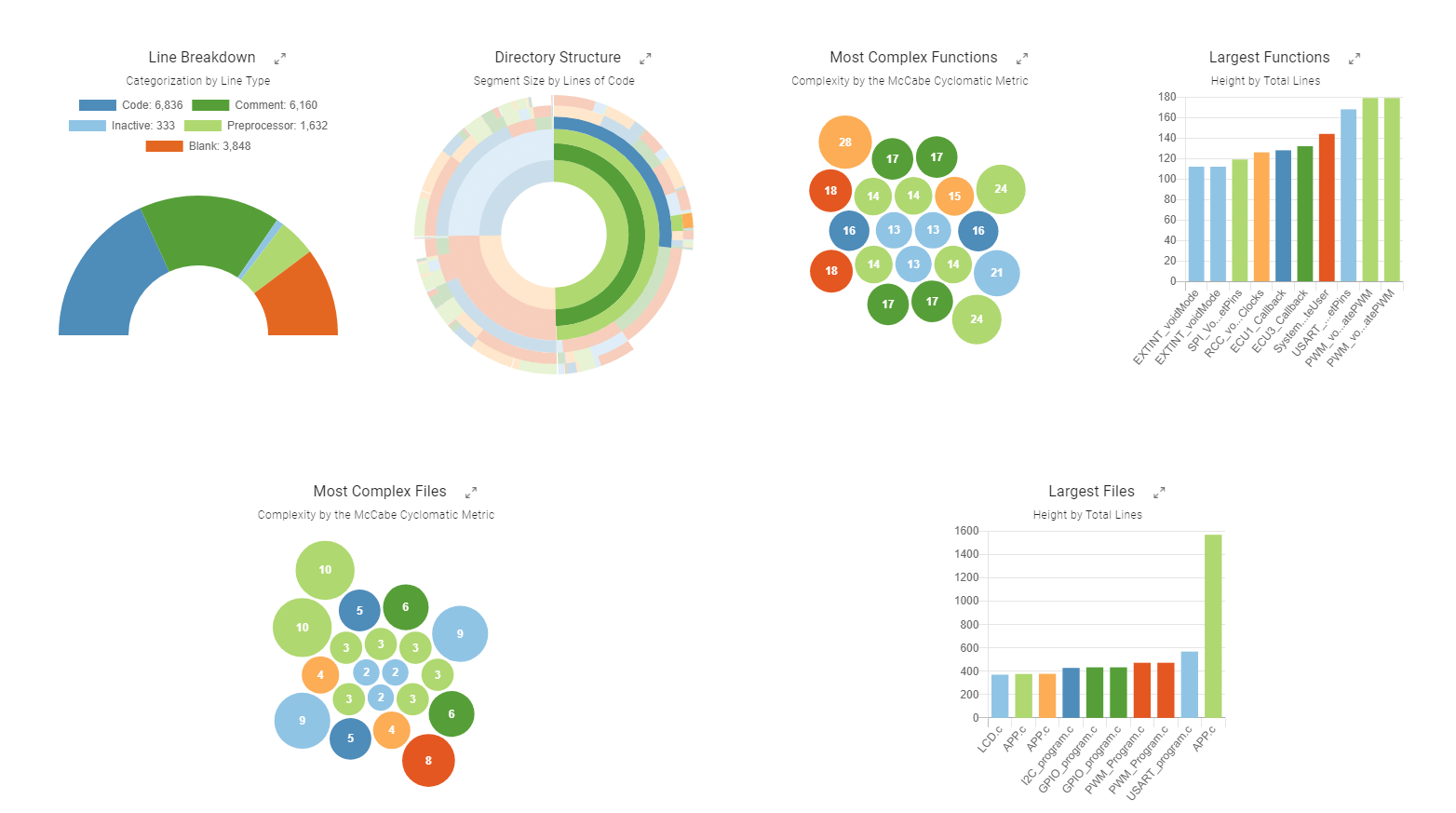


Figure 1:Project Overview

## Calls

* ECU1

Diagram, schematic

Description automatically generated

Figure 2:ECU1 Calls

* ECU2

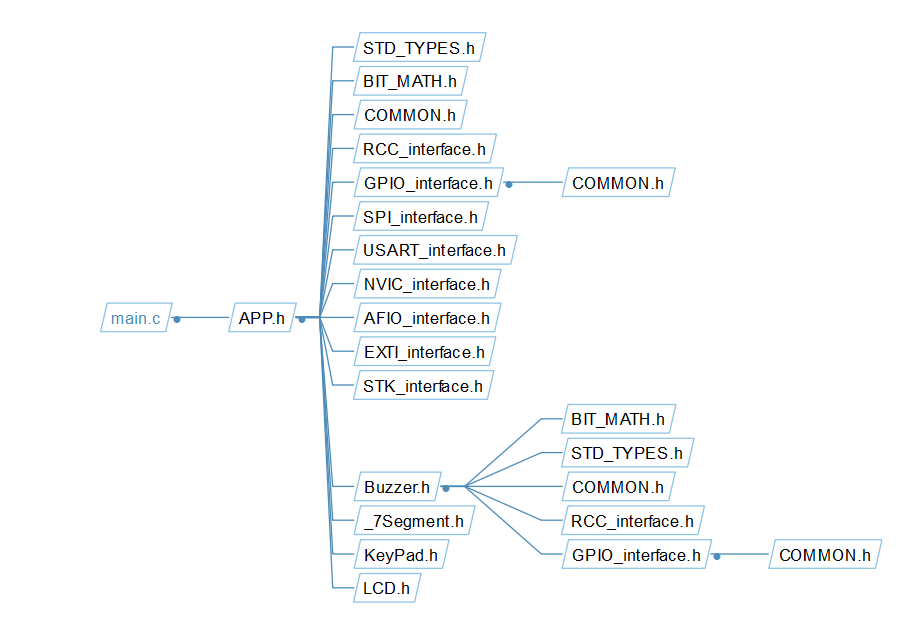


Figure 3:ECU2 Calls

* ECU3

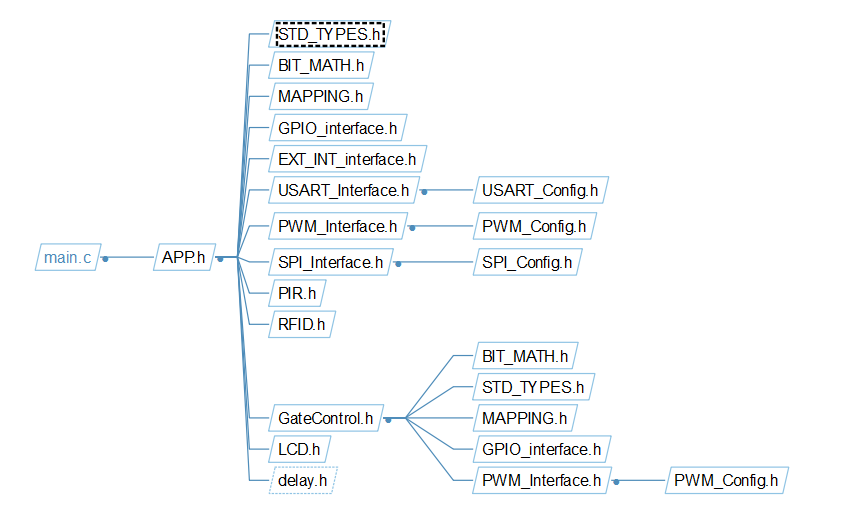


Figure 4:ECU3 Calls

# System Architecture

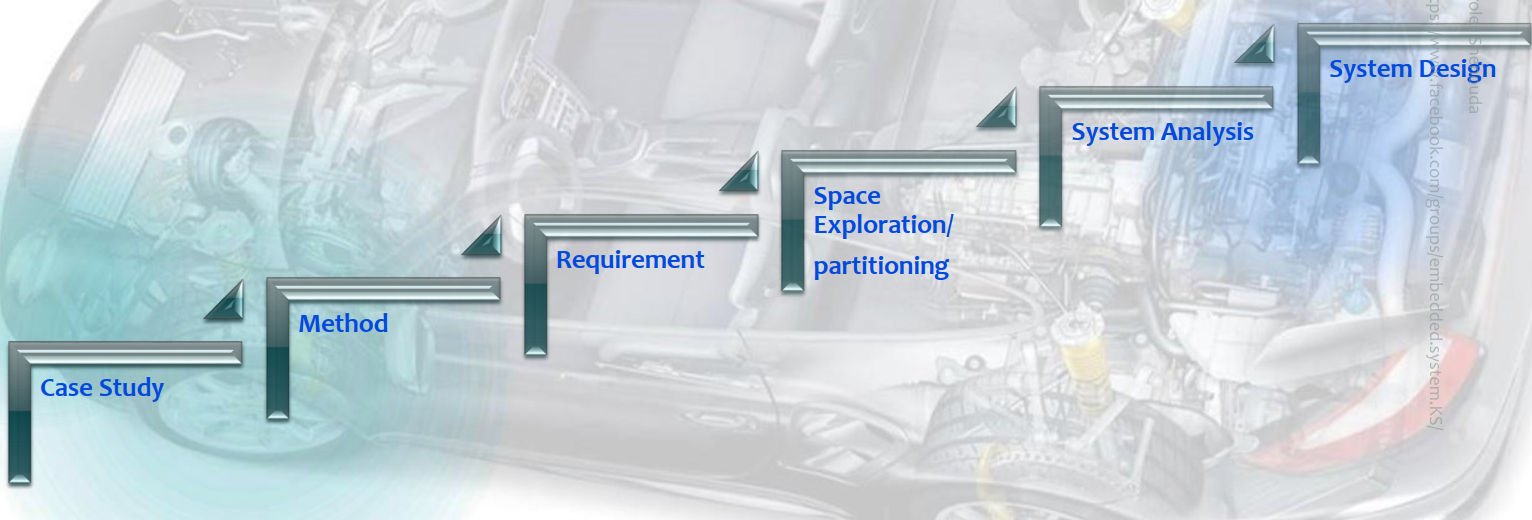


Figure 5:System Architecture

## Case study

software that controls the private parking garage.

## Method

Adaptive Technique: Agile Scrum Methodology

## Requirement

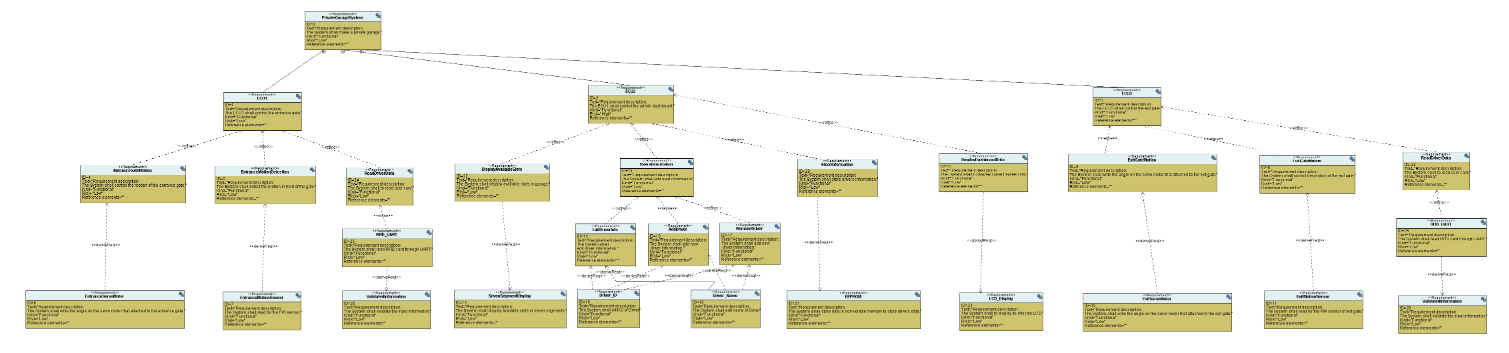


Figure 6:System Requirement

## Space exploration/partitioning

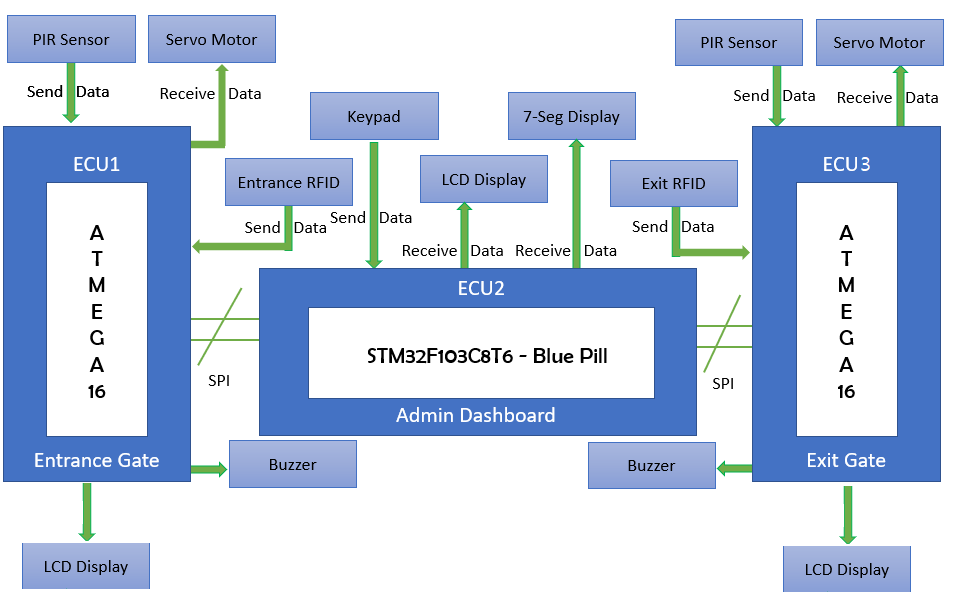


Figure 7:System Partitioning

I used STM32F103C8T6 - Blue Pill board for ECU2 which based on ARM Cortex m3 microprocessor its specification

1. ARM 32-bit Cortex™-M3 CPU Core
2. 72 MHz maximum frequency
3. Single-cycle multiplication and hardware division.
4. Memories
5. 32 Kbytes of Flash memory
6. 10 Kbytes of SRAM
7. Clock, reset and supply management
8. 2.0 to 3.6 V application supply and I/Os.
9. 4-to-16 MHz crystal oscillator.
10. 32 kHz oscillator for RTC with calibration

And used ATmega16 For ECU1 and ECU3

Graphical user interface

Description automatically generated

## System Analysis

### Use Case Diagram

Diagram

Description automatically generated

Figure 8:ECU2 Use Case Diagram

Diagram

Description automatically generated

Figure 9: ECU1 & ECU3 Use Case Diagram

### Simple Activity Diagram

Diagram

Description automatically generated

Figure 10:ECU2 Activity Diagram

Diagram

Description automatically generated

Figure 11:ECU1 Activity Diagram

Diagram

Description automatically generated

Figure 12:ECU3 Activity Diagram

### Sequence Diagram (UML)

* ECU1 UML

Graphical user interface

Description automatically generated with medium confidence

Figure 13:ECU1 UML Diagram

* ECU2 UML

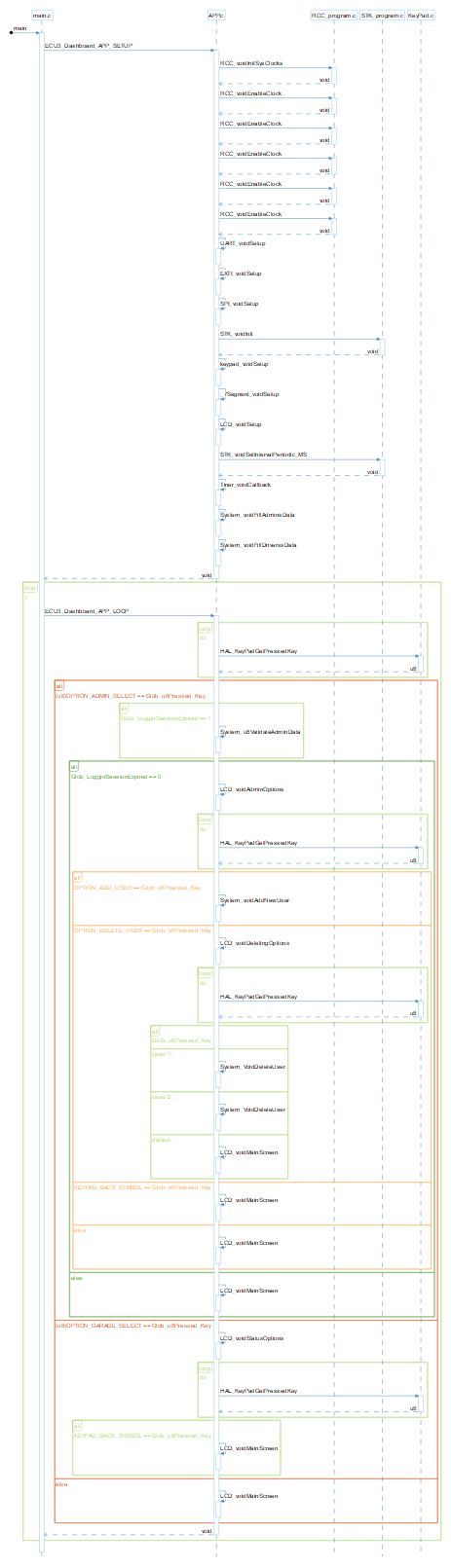


Figure 14:ECU2 UML Diagram

* ECU3 UML

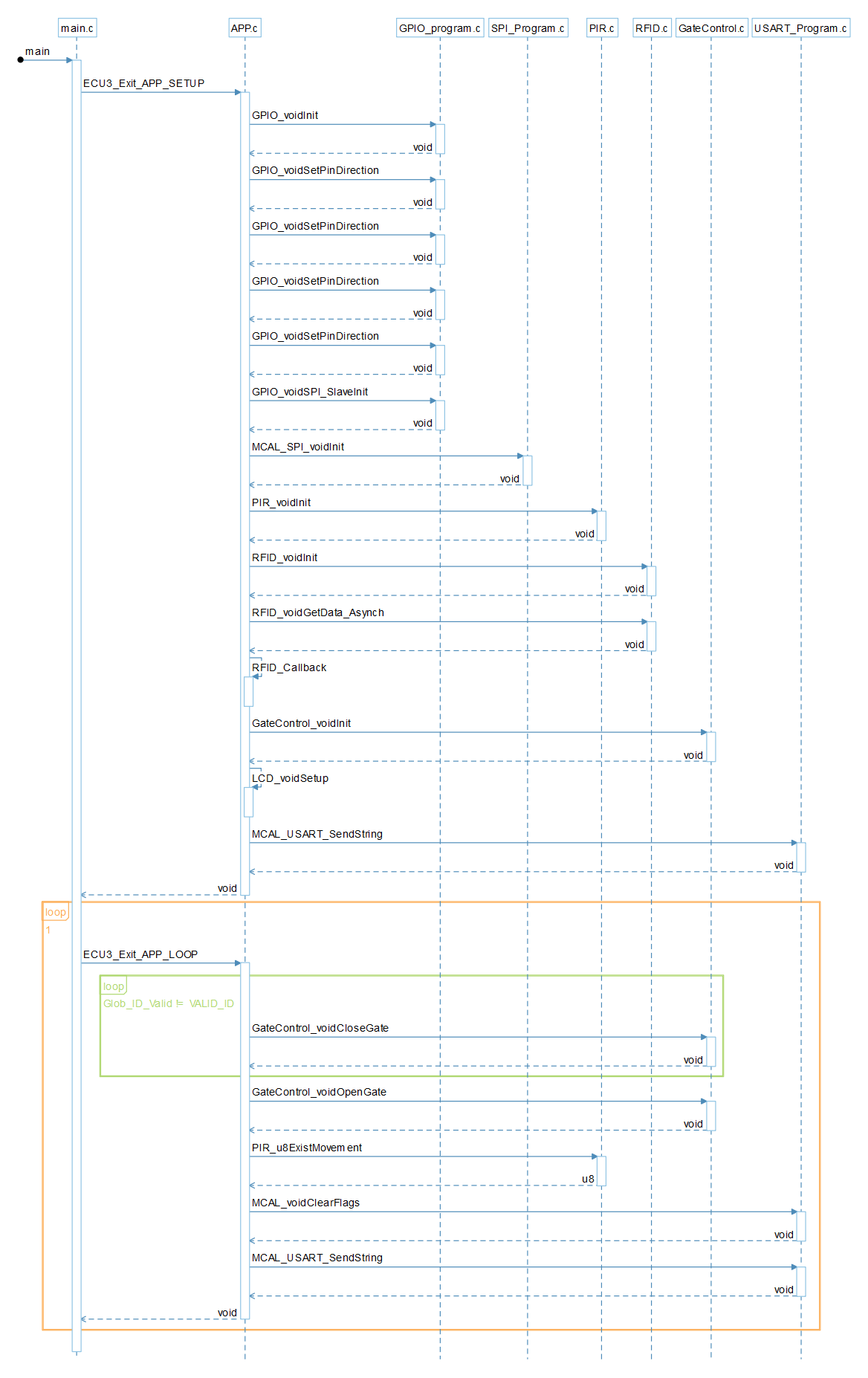


Figure 15:ECU3 UML Diagram

## System Design

Graphical user interface, application

Description automatically generated

Figure 16:System Design

# Boot Sequence of STM32

Entry Point (Reset Section)

Reset section

FLASH MEM

BareMetal SW

A picture containing map

Description automatically generated

Figure 17:Sequence

Our entry point is reset handler that move .data from FLASH to SRAM and reserve .bss section in SRAM.

# Map File

* Memory map

Graphical user interface, text

Description automatically generated

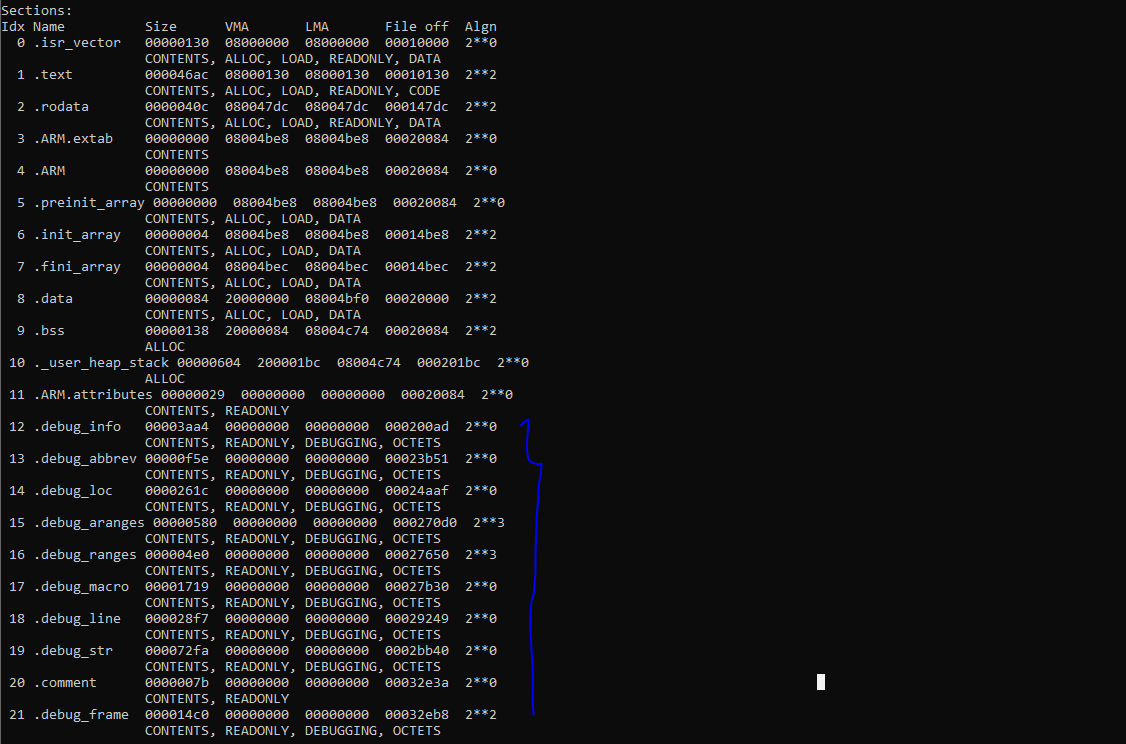


Start address of flash memory



Figure 18:Vector table position in map file

# Memory dump



Debug info

Figure 19:Memory Dump with debug section

# Some of Symbols

Text

Description automatically generated

Figure 20:Symbols of ELF image

All symbols successfully resolved

For each symbol check [link](https://sourceware.org/binutils/docs/binutils/nm.html)

# ELF image details

Text

Description automatically generated

Figure 21:ELF image details

# Hardware Simulation

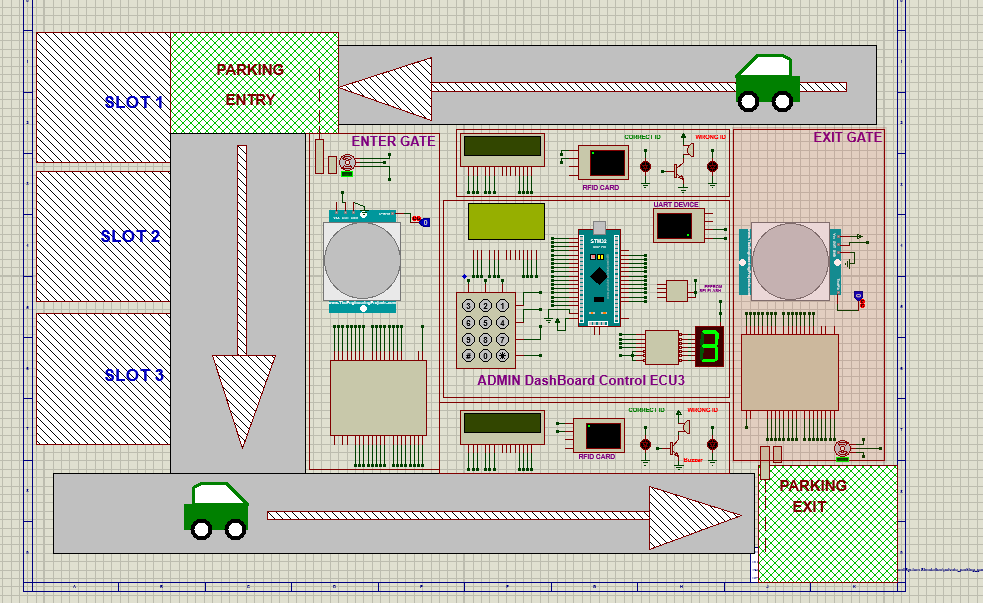


Figure 22:simulation test

# Test Cases

### ECU1

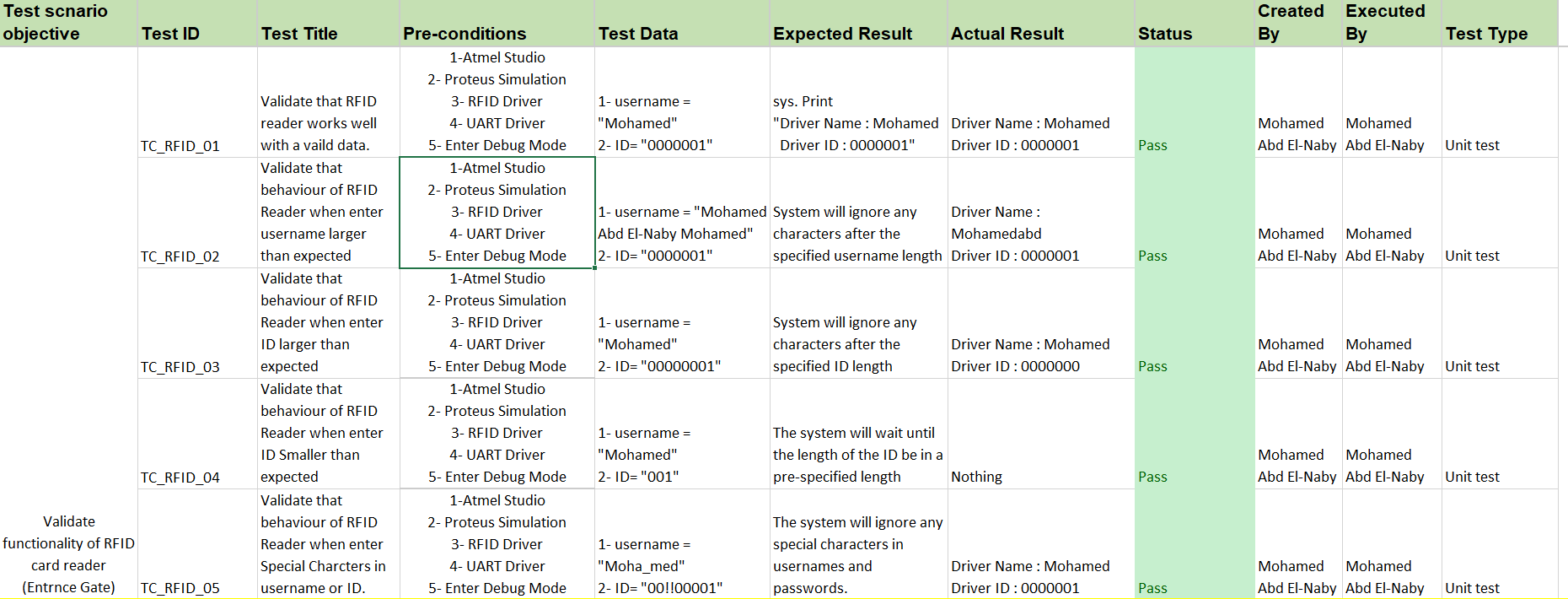


Figure 23:ECU1 TEST CASES

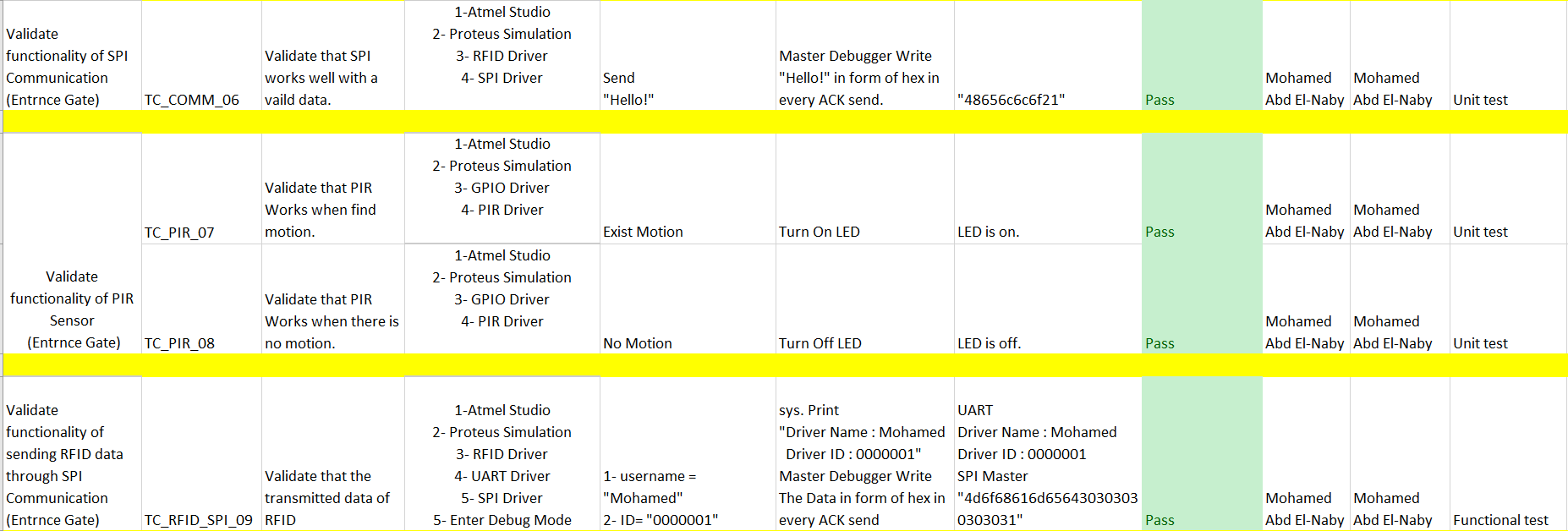


Figure 24:ECU1 TEST CASES



Figure 25:ECU1 TEST CASES

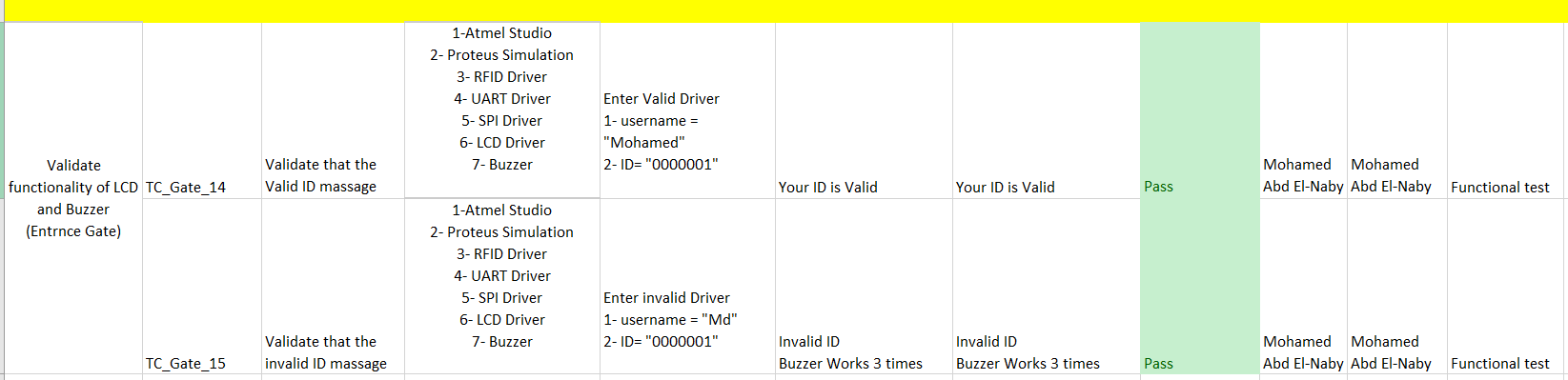


Figure 26:ECU1 TEST CASES

### ECU2

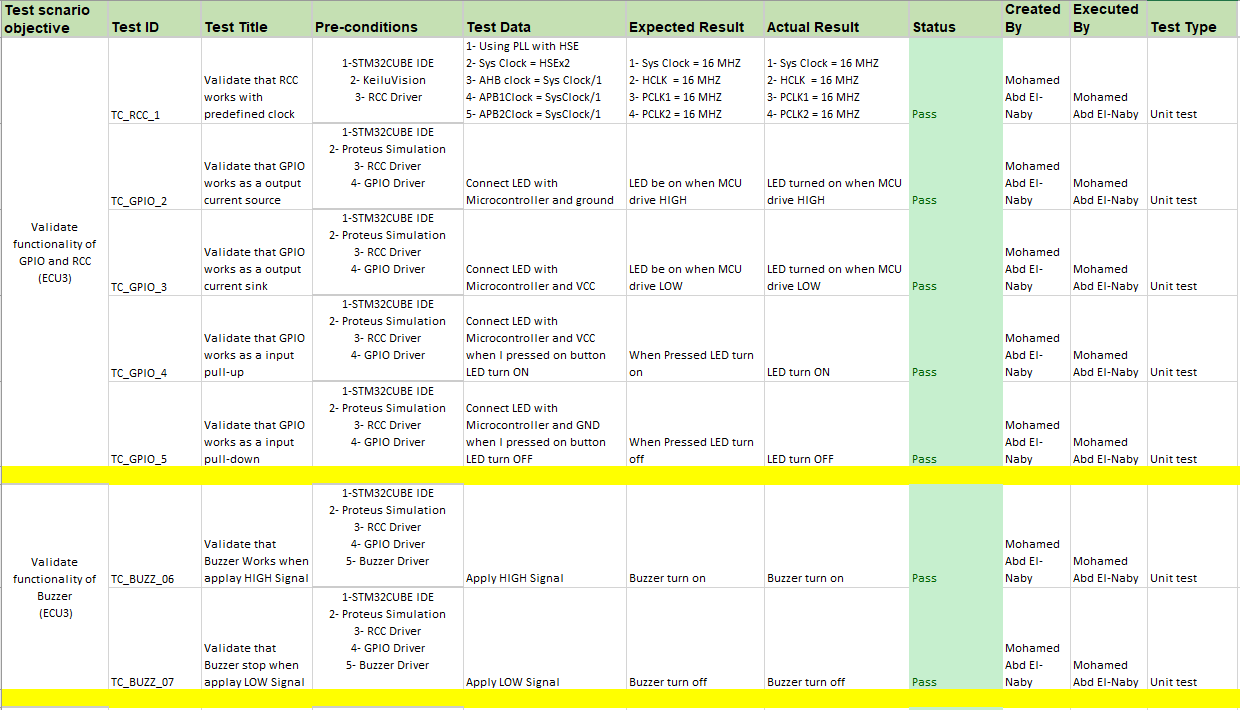


Figure 27:ECU2 TEST CASES

A picture containing table

Description automatically generated

Figure 28:ECU2 TEST CASES

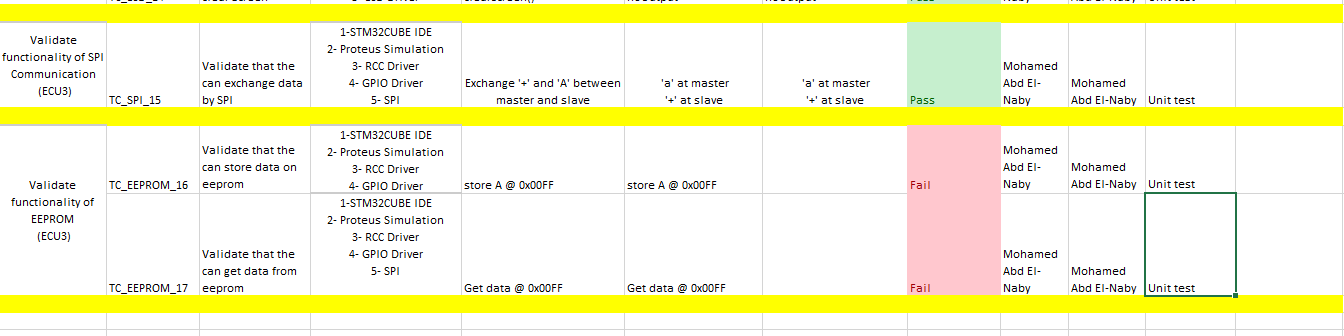


Figure 29:ECU2 TEST CASES

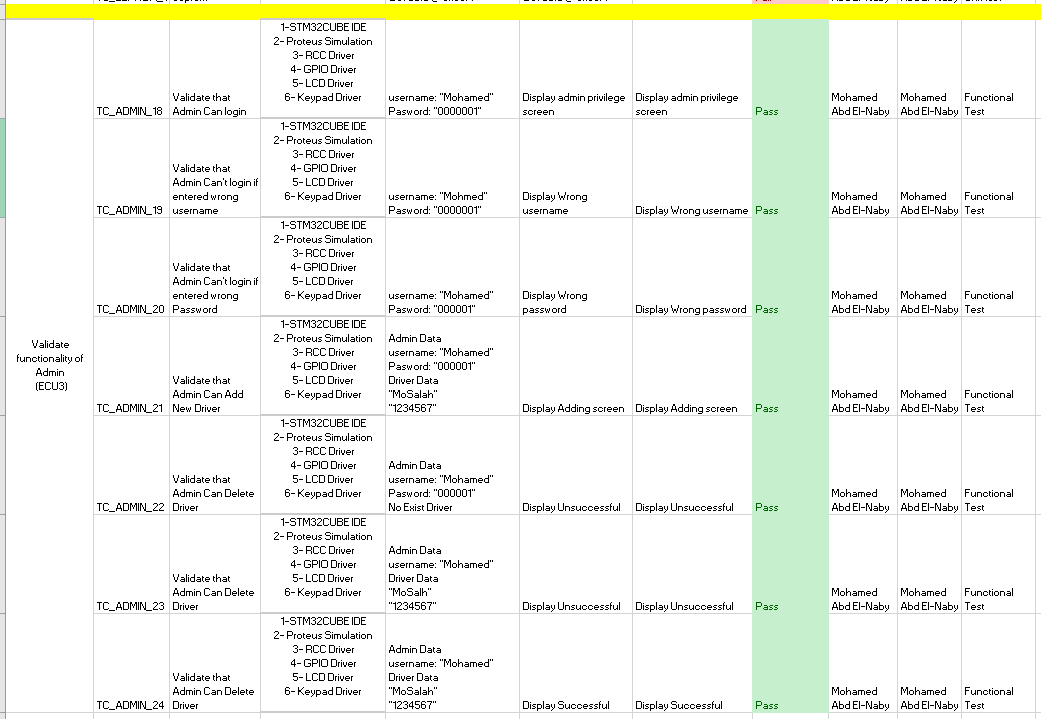


Figure 30:ECU2 TEST CASES

### ECU1

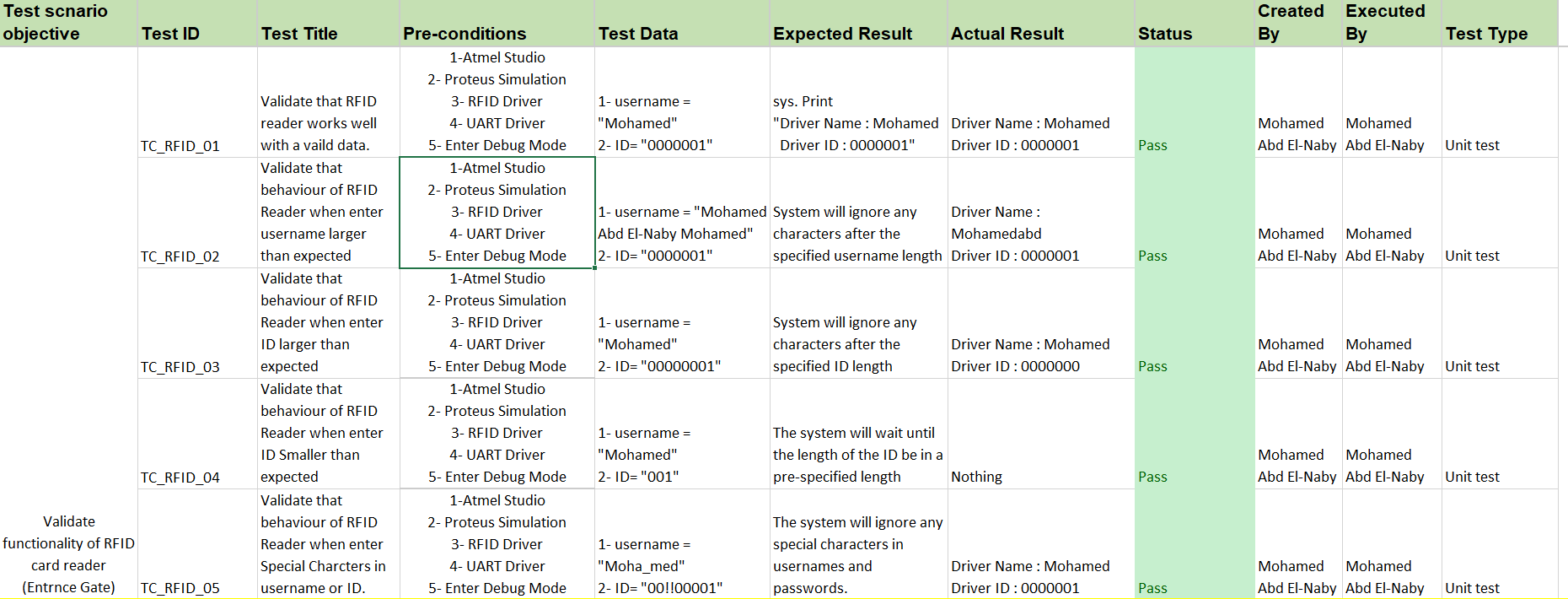


Figure 31:ECU3 TEST CASES

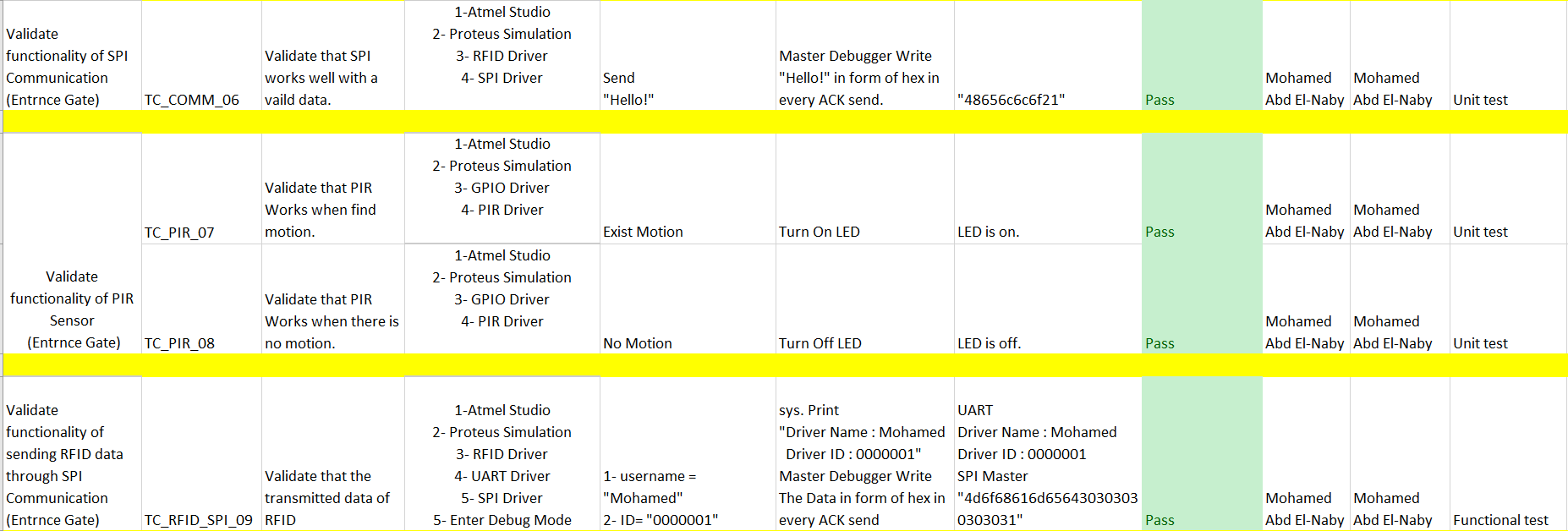


Figure 32:ECU3 TEST CASES



Figure 33:ECU3 TEST CASES

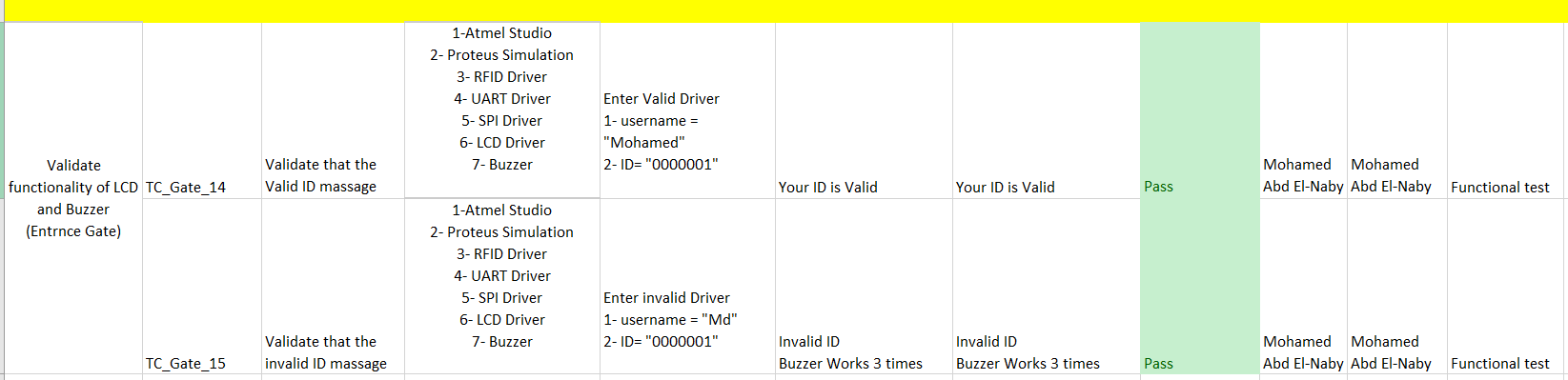


Figure 34:ECU3 TEST CASES