Table of Contents

[1 System Block Diagram 1](#_Toc114774933)

[2 Static Design 2](#_Toc114774934)

[2.1Layered Architecture Design 2](#_Toc114774935)

[1.1ECU 1 3](#_Toc114774936)

[1.1.1 Component and Modules 3](#_Toc114774937)

[1.1.2 APIs Description 3](#_Toc114774938)

[1.1.3 Folder Structure 8](#_Toc114774939)

[1.2ECU 2 10](#_Toc114774940)

[1.2.1 Component and Modules 10](#_Toc114774941)

[1.2.2 APIs Description 10](#_Toc114774942)

[1.2.3 Folder Structure 14](#_Toc114774943)

[2 Dynamic Design 15](#_Toc114774944)

[2.1ECU 1 15](#_Toc114774945)

[2.1.1 ECU components state machines 15](#_Toc114774946)

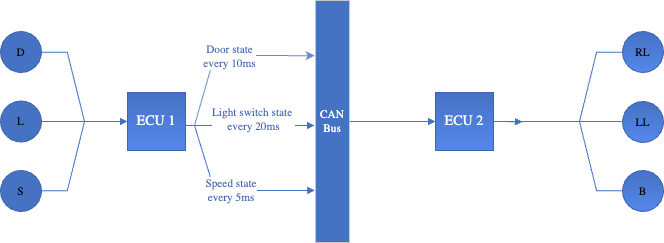
[2.1.2 ECU state machine 15](#_Toc114774947)

[2.2ECU 2 15](#_Toc114774948)

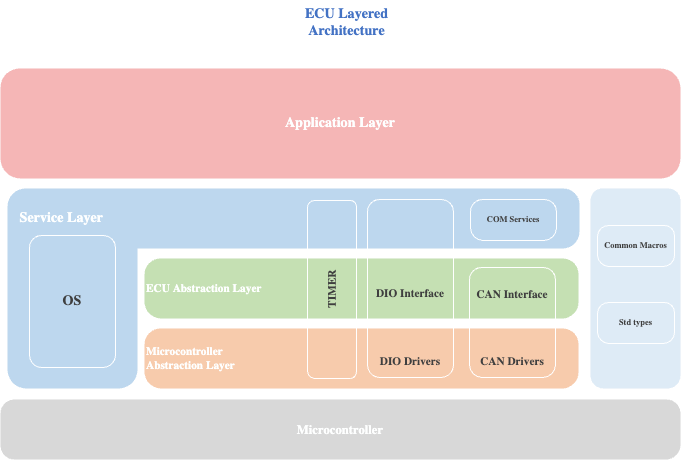
[2.2.1 ECU components state machines 15](#_Toc114774949)

[2.2.2 ECU state machine 15](#_Toc114774950)

1. System Block Diagram



1. Static Design
   1. Layered Architecture Design



* 1. ECU 1
     1. Component and Modules

|  |  |  |  |
| --- | --- | --- | --- |
| MCAL | | ECU Abstraction Layer | |
| **DIO APIs** | * Initialize * read * write | **Door Sensor APIs** | * Initialize * Read |
| **Timer APIs** | * Initialize * Start * Stop | **Light Switch APIs** | * Initialize * Read |
| **CAN APIs** | * Initialize * Send * Receive | **Speed Sensor APIs** | * Initialize * Read |
| **ADC APIs** | * Initialize * Read |

* + 1. APIs Description

**DIO APIs:**

|  |  |  |  |
| --- | --- | --- | --- |
| Function name | DIO\_Init | | |
| Arguments | Inputs | pinConfig | DIO\_pinConfig \* |
|  | array of pins configuration | |
| Output | N/A | |
| Input/Output | N/A |  |
| Return | E\_OK | 0 | |
| E\_NOK | 1 | |
| Description | Initialize ports according to the configurations in DIO\_Config.c | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Function name | DIO\_Read | | |
| Arguments | Inputs | channel | Gpio\_ChannelType |
| the channel to be read | |
| Output | uint8 | channel value |
| Input/Output | N/A | |
| Return | uint8 | pin value | |
| Description | Reads the value of a certain pin in a port (channel) | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Function name | DIO\_Write | | |
| Arguments | Inputs | channel | Gpio\_ChannelType |
| the channel to be read | |
| Output | N/A | |
| Input/Output |  | |
| Return | uint8 | pin value | |
| Description | Reads the value of a certain pin in a port (channel) | | |

|  |  |  |
| --- | --- | --- |
| Name | DIO\_pinConfig | |
| Type | Struct | |
| Elements | channel | channel port and pin |
| pinMode | the mode of the pin |
| pinValue | the initial value of the pin |
| pinDirection | the direction of the pin |
| Description | pin configuration used for initializing a pin | |

|  |  |  |
| --- | --- | --- |
| Name | Gpio\_ChannelType | |
| Type | Struct | |
| Elements | port | channel port |
| pin | channel pin |
| Description | pin configuration used for initializing a pin | |

**ADC APIs:**

|  |  |  |  |
| --- | --- | --- | --- |
| Function name | ADC\_Init | | |
| Arguments | Inputs | pinConfig | DIO\_pinConfig \* |
|  | array of pins configuration | |
| Output | N/A | |
| Input/Output | N/A |  |
| Return | E\_OK | 0 | |
| E\_NOK | 1 | |
| Description | Initialize ports according to the configurations in DIO\_Config.c | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Function name | ADC\_Read | | |
| Arguments | Inputs | channel | Gpio\_ChannelType |
| the channel to be read | |
| Output | uint8 | channel value |
| Input/Output | N/A | |
| Return | uint32 | pin value | |
| Description | Reads the value of a certain pin in a port (channel) | | |

**Timer APIs:**

|  |  |  |  |
| --- | --- | --- | --- |
| Function name | Timer\_Init | | |
| Arguments | Inputs | timerConfig | Timer\_configType |
|  | Configuration structure | |
| Output | N/A | |
| Input/Output | N/A |  |
| Return | E\_OK | 0 | |
| E\_NOK | 1 | |
| Description | Initialize a timer | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Function name | Timer\_start | | |
| Arguments | Inputs | channel | uint\_8 |
| Timer channel to be stated | |
| time | Uint\_32 |
| The time for the timer | |
| Output | N/A |  |
| Input/Output | N/A | |
| Return | E\_OK | 0 | |
| E\_NOK | 1 | |
| Description | Starts a given timer with a given time | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Function name | Timer\_stop | | |
| Arguments | Inputs | channel | uint\_8 |
| Timer channel to be stopped | |
| Output | N/A |  |
| Input/Output | N/A | |
| Return | E\_OK | 0 | |
|  | E\_NOK | 1 | |
| Description | Stops a given timer | | |

**CAN APIs:**

|  |  |  |  |
| --- | --- | --- | --- |
| Function name | CAN\_Init | | |
| Arguments | Inputs | config | CAN\_configType |
|  | Configuration structure | |
| Output | N/A | |
| Input/Output | N/A |  |
| Return | E\_OK | 0 | |
| E\_NOK | 1 | |
| Description | Initialize the CAN communication | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Function name | CAN\_sendData | | |
| Arguments | Inputs | data | uint8 |
| The data to be sent | |
| Output | N/A |  |
| Input/Output | N/A | |
| Return | E\_OK | 0 | |
| E\_NOK | 1 | |
| Description | Sends data over CAN bus | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Function name | CAN\_receiveData | | |
| Arguments | Inputs | id | uint8 |
| Data ID | |
| Output | N/A | |
| Input/Output | N/A | |
| Return | uint8 | Data received | |
| Description | Recieves data from the CAN bus | | |

**Door Sensor APIs:**

|  |  |  |  |
| --- | --- | --- | --- |
| Function name | DoorSensor\_Init | | |
| Arguments | Inputs | config | Door\_configType |
|  | Configuration structure | |
| Output | N/A | |
| Input/Output | N/A |  |
| Return | E\_OK | 0 | |
| E\_NOK | 1 | |
| Description | Initialize the door sensor | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Function name | DoorSensor\_ReadValue | | |
| Arguments | Inputs | channel | Uint8 |
| The door sensor to be read value from | |
| Output | N/A | |
| Input/Output | N/A | |
| Return | uint8 | Sensor value | |
| Description | Reads the value of the door sensor | | |

**Light Switch APIs:**

|  |  |  |  |
| --- | --- | --- | --- |
| Function name | LightSW\_Init | | |
| Arguments | Inputs | config | LightSW\_configType |
|  | Configuration structure | |
| Output | N/A | |
| Input/Output | N/A |  |
| Return | E\_OK | 0 | |
| E\_NOK | 1 | |
| Description | Initialize the light switch | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Function name | LightSW\_ReadValue | | |
| Arguments | Inputs | channel | Uint8 |
| The light switch to be read value from | |
| Output | N/A | |
| Input/Output | N/A | |
| Return | Uint8 | Switch state | |
| Description | Reads the state of the light switch | | |

**Speed Sensor APIs:**

|  |  |  |  |
| --- | --- | --- | --- |
| Function name | SpeedSensor\_Init | | |
| Arguments | Inputs | config | Speed\_configType |
|  | Configuration structure | |
| Output | N/A | |
| Input/Output | N/A |  |
| Return | E\_OK | 0 | |
| E\_NOK | 1 | |
| Description | Initialize the speed sensor | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Function name | SpeedSensor\_ReadValue | | |
| Arguments | Inputs | channel | Uint8 |
| The speed sensor to be read value from | |
| Output | N/A | |
| Input/Output | N/A | |
| Return | Uint32 | Sensor value | |
| Description | Reads the value of the speed sensor | | |

* + 1. Folder Structure

/ECU1/MCAL/

/ECU1/MCAL/Inc/

/ECU1/HAL/

/ECU1/HAL/Inc/

/ECU1/APP/

/ECU1/APP/Inc

/ECU1/Config

/ECU1/Service

/ECU1/Common

* 1. ECU 2
     1. Component and Modules

|  |  |  |  |
| --- | --- | --- | --- |
| MCAL | | ECU Abstraction Layer | |
| **DIO APIs** | * Initialize * read * write | **Lights APIs** | * Initialize * on * off |
| **Timer APIs** | * Initialize * Start * Stop | **Buzzer APIs** | * Initialize * on * off |
| **CAN APIs** | * Initialize * Send * Receive |  |  |

* + 1. APIs Description

**DIO APIs:**

|  |  |  |  |
| --- | --- | --- | --- |
| Function name | DIO\_Init | | |
| Arguments | Inputs | pinConfig | DIO\_pinConfig \* |
|  | array of pins configuration | |
| Output | N/A | |
| Input/Output | N/A |  |
| Return | E\_OK | 0 | |
| E\_NOK | 1 | |
| Description | Initialize ports according to the configurations in DIO\_Config.c | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Function name | DIO\_Read | | |
| Arguments | Inputs | channel | Gpio\_ChannelType |
| the channel to be read | |
| Output | uint8 | channel value |
| Input/Output | N/A | |
| Return | uint8 | pin value | |
| Description | Reads the value of a certain pin in a port (channel) | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Function name | DIO\_Write | | |
| Arguments | Inputs | channel | Gpio\_ChannelType |
| the channel to be read | |
| Output | N/A | |
| Input/Output |  | |
| Return | uint8 | pin value | |
| Description | Reads the value of a certain pin in a port (channel) | | |

|  |  |  |
| --- | --- | --- |
| Name | DIO\_pinConfig | |
| Type | Struct | |
| Elements | channel | channel port and pin |
| pinMode | the mode of the pin |
| pinValue | the initial value of the pin |
| pinDirection | the direction of the pin |
| Description | pin configuration used for initializing a pin | |

|  |  |  |
| --- | --- | --- |
| Name | Gpio\_ChannelType | |
| Type | Struct | |
| Elements | port | channel port |
| pin | channel pin |
| Description | pin configuration used for initializing a pin | |

**Timer APIs:**

|  |  |  |  |
| --- | --- | --- | --- |
| Function name | Timer\_Init | | |
| Arguments | Inputs | timerConfig | Timer\_configType |
|  | Configuration structure | |
| Output | N/A | |
| Input/Output | N/A |  |
| Return | E\_OK | 0 | |
| E\_NOK | 1 | |
| Description | Initialize a timer | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Function name | Timer\_start | | |
| Arguments | Inputs | channel | uint\_8 |
| Timer channel to be stated | |
| time | Uint\_32 |
| The time for the timer | |
| Output | N/A |  |
| Input/Output | N/A | |
| Return | E\_OK | 0 | |
| E\_NOK | 1 | |
| Description | Starts a given timer with a given time | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Function name | Timer\_stop | | |
| Arguments | Inputs | channel | uint\_8 |
| Timer channel to be stopped | |
| Output | N/A |  |
| Input/Output | N/A | |
| Return | E\_OK | 0 | |
| E\_NOK | 1 | |
| Description | Stops a given timer | | |

**CAN APIs:**

|  |  |  |  |
| --- | --- | --- | --- |
| Function name | CAN\_Init | | |
| Arguments | Inputs | config | CAN\_configType |
|  | Configuration structure | |
| Output | N/A | |
| Input/Output | N/A |  |
| Return | E\_OK | 0 | |
| E\_NOK | 1 | |
| Description | Initialize the CAN communication | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Function name | CAN\_sendData | | |
| Arguments | Inputs | data | uint8 |
| The data to be sent | |
| Output | N/A |  |
| Input/Output | N/A | |
| Return | E\_OK | 0 | |
| E\_NOK | 1 | |
| Description | Sends data over CAN bus | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Function name | CAN\_receiveData | | |
| Arguments | Inputs | id | uint8 |
| Data ID | |
| Output | N/A | |
| Input/Output | N/A | |
| Return | uint8 | Data received | |
| Description | Receives data from the CAN bus | | |

**Light APIs:**

|  |  |  |  |
| --- | --- | --- | --- |
| Function name | Light\_Init | | |
| Arguments | Inputs | config | Light\_configType |
|  | Configuration structure | |
| Output | N/A | |
| Input/Output | N/A |  |
| Return | E\_OK | 0 | |
| E\_NOK | 1 | |
| Description | Initialize the lights | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Function name | Light\_On | | |
| Arguments | Inputs | channel | Uint8 |
| The light to be turned on | |
| Output | N/A |  |
| Input/Output | N/A | |
| Return | E\_OK | 0 | |
| E\_NOK | 1 | |
| Description | Turns on the light given | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Function name | Light\_Off | | |
| Arguments | Inputs | channel | Uint8 |
| The light to be turned on | |
| Output | N/A |  |
| Input/Output | N/A | |
| Return | E\_OK | 0 | |
| E\_NOK | 1 | |
| Description | Turns off the light given | | |

* + 1. Folder Structure

/ECU2/MCAL/

/ECU2/MCAL/Inc/

/ECU2/HAL/

/ECU2/HAL/Inc/

/ECU2/APP/

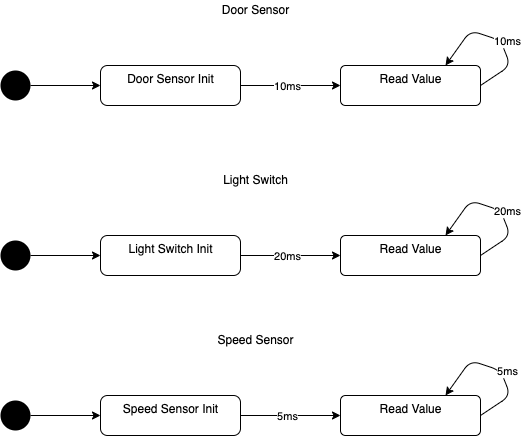
/ECU2/APP/Inc

/ECU2/Config

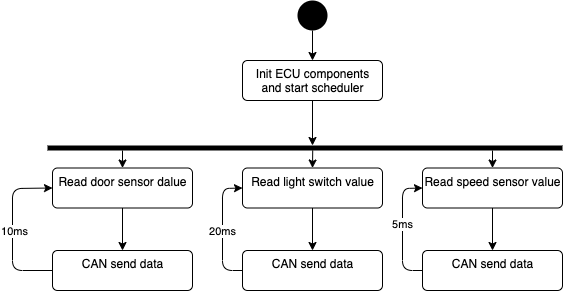
/ECU2/Service

/ECU2/Common

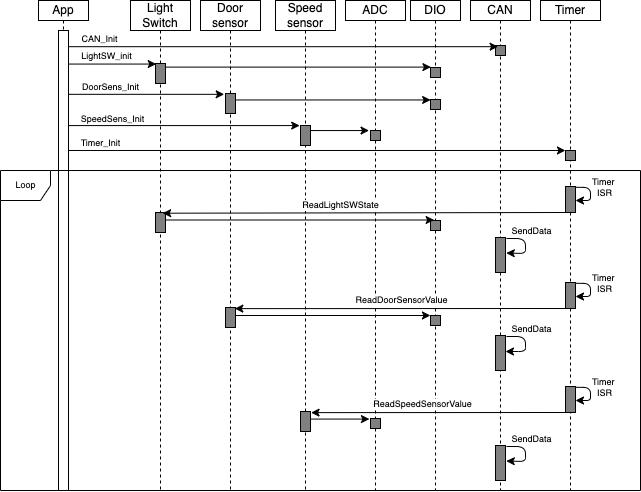
1. Dynamic Design
   1. ECU 1
      1. ECU components state machines

****

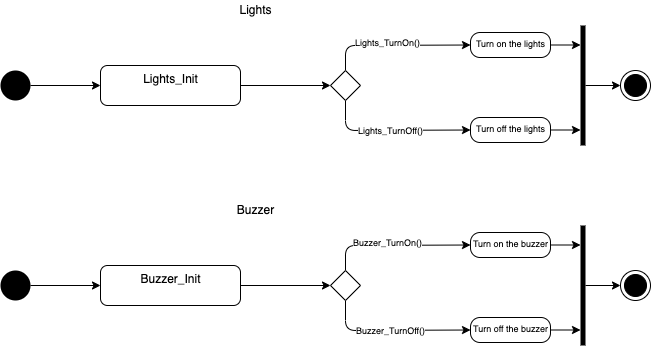
* + 1. ECU state machine



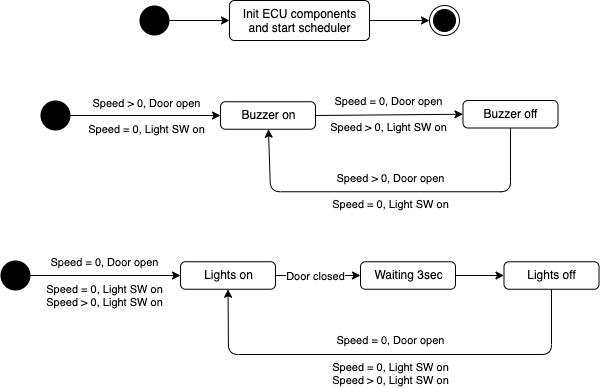
* + 1. ECU sequence diagram



* 1. ECU 2
     1. ECU components state machines



* + 1. ECU state machine



* + 1. ECU sequence diagram

