

Static Design

FEBRUARY 9

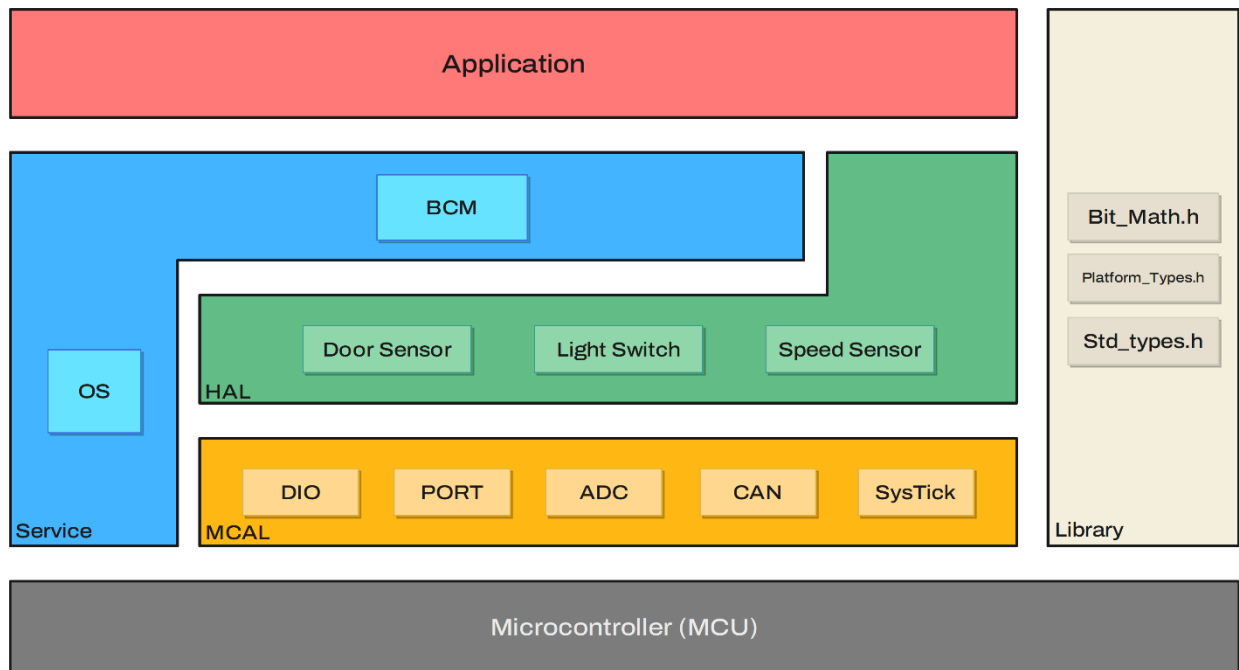
Automotive Door Control System Design
Khaled Ibrahim Abdulaziz



1.ECU1

1.1 Layered Architecture

Making the layered architecture and specifying ECU1 components and modules.



1.2 APIs and Typedefs

1.2.1 PORT

1.2.1.1 APIs

Name	Port_Init
Syntax	void Port_Init(Port_ConfigType *PortConfigPtr)
Description	Initializes the Pin with the configured functionality
Sync/Async	Synchronous
Reentrancy	Non-Reentrant
Parameters(in)	PortConfigPtr
Parameters(out)	None
Return value	Void

1.2.1.2 Typedefs

Name	Port_ConfigType
Type	struct

Description	A struct holds the configurations of the pin including its port, number, mode, level, etc..
--------------------	---

1.2.2 DIO

1.2.2.1 APIs

Name	Dio_ReadChannel
Syntax	Dio_LevelType Dio_ReadChannel(Dio_PortType PortId, Dio_ChannelType ChannelId)
Description	Reads the DIO channel
Sync/Async	Synchronous
Reentrancy	Non-Reentrant
Parameters(in)	PortId, ChannelId
Parameters(out)	None
Return value	Dio_LevelType

Name	Dio_WriteChannel
Syntax	void Dio_WriteChannel(Dio_PortType PortId, Dio_ChannelType ChannelId, Dio_LevelType level)
Description	Writes to the DIO channel
Sync/Async	Synchronous
Reentrancy	Non-Reentrant
Parameters(in)	PortId, ChannelId, level
Parameters(out)	None
Return value	void

1.2.2.2 Typedefs

Name	Dio_LevelType
Type	enum
Description	An enum holds the level of the pin which is Low (0) or High (1).

Name	Dio_PortType
Type	enum
Description	An enum holds the ports of MCU such as Port_A, Port_B, etc..

Name	Dio_ChannelType
Type	enum
Description	An enum holds the pins of the port such as Pin_0, Pin_1, etc...

1.2.3 ADC

1.2.3.1 APIs

Name	ADC_Init
Syntax	void ADC_Init(ADC_ConfigType *ADCConfigPtr)
Description	Initializes ADC module with the desired configurations
Sync/Async	Synchronous
Reentrancy	Non-Reentrant
Parameters(in)	ADCConfigPtr
Parameters(out)	None
Return value	void

Name	ADC_Read
Syntax	void ADC_ReadChannel(ADC_ChannelType ChannelId)
Description	Reads the current value of ADC channel
Sync/Async	Synchronous
Reentrancy	Non-Reentrant
Parameters(in)	ChannelId
Parameters(out)	None
Return value	void

1.2.3.2 Typedefs

Name	ADCConfigPtr
Type	struct
Description	A struct that holds all ADC configurations.

Name	ADC_ChannelType
Type	enum
Description	An enum represents the ADC channels like ADC_0, ADC_1, etc..

1.2.4 CAN

1.2.4.1 APIs

Name	CAN_Init
Syntax	void CAN_Init(CAN_ConfigType *CANConfigPtr)
Description	Initializes CAN module with the desired configurations
Sync/Async	Synchronous
Reentrancy	Non-Reentrant
Parameters(in)	CANConfigPtr
Parameters(out)	None

Return value	void
---------------------	------

Name	CAN_Transmit
Syntax	void CAN_Transmit(uint32_t data)
Description	Sends data through CAN bus
Sync/Async	Synchronous
Reentrancy	Non-Reentrant
Parameters(in)	data
Parameters(out)	None
Return value	void

1.2.4.2 Typedefs

Name	CAN_ConfigType
Type	struct
Description	A struct that holds all CAN configurations.

1.2.5 SysTick

1.2.5.1 APIs

Name	SysTick_Init
Syntax	void SysTick_Init(SysTick_ConfigType SysTickConfigPtr)
Description	Initializes SysTick timer with the desired configurations
Sync/Async	Synchronous
Reentrancy	Non-Reentrant
Parameters(in)	SysTickConfigPtr
Parameters(out)	None
Return value	void

1.2.5.2 Typedefs

Name	SysTick_ConfigType
Type	struct
Description	A struct that holds all SysTick configurations.

1.2.6 Door Sensor

1.2.6.1 APIs

Name	DoorSensor_Init
Syntax	void DoorSensor_Init(DS_ConfigType DSConfig)
Description	Initializes Door Sensor module with the desired configurations
Sync/Async	Synchronous

Reentrancy	Non-Reentrant
Parameters(in)	DSConfig
Parameters(out)	None
Return value	void

Name	DoorSensor_GetState
Syntax	DS_StateType DoorSensor_GetState(void)
Description	Gets the current state of Door Sensor
Sync/Async	Synchronous
Reentrancy	Reentrant
Parameters(in)	void
Parameters(out)	None
Return value	DS_StateType

1.2.6.2 Typedefs

Name	DSConfig
Type	struct
Description	A struct that holds all Door Sensor configurations.

Name	DS_StateType
Type	enum
Description	An enum represents the Door Sensor state either Open (1) or Closed (0).

1.2.7 Light Switch

1.2.7.1 APIs

Name	LightSwitch_Init
Syntax	Void LightSwitch_Init(LS_ConfigType LSConfig)
Description	Initializes Light Switch module with the desired configurations
Sync/Async	Synchronous
Reentrancy	Reentrant
Parameters(in)	DSConfig
Parameters(out)	None
Return value	void

Name	LightSwitch_GetState
Syntax	LS_StateType LightSwitch_GetState(void)
Description	Gets the current state of Light Switch
Sync/Async	Synchronous

Reentrancy	Reentrant
Parameters(in)	void
Parameters(out)	None
Return value	LS_StateType

1.2.7.2 Typedefs

Name	LSConfig
Type	struct
Description	A struct that holds all Light Switch configurations.

Name	LS_StateType
Type	enum
Description	An enum represents the Light Switch state either ON (1) or OFF (0).

1.2.8 Speed Sensor

1.2.8.1 APIs

Name	SpeedSensor_Init
Syntax	void SpeedSensor_Init(SS_ConfigType SSConfig)
Description	Initializes Speed Sensor module with the desired configurations
Sync/Async	Synchronous
Reentrancy	Reentrant
Parameters(in)	SSConfig
Parameters(out)	None
Return value	void

Name	SpeedSensor_GetState
Syntax	uint16_t SpeedSensor_GetValue(void)
Description	Gets the current value of Speed Sensor Module
Sync/Async	Synchronous
Reentrancy	Non-Reentrant
Parameters(in)	void
Parameters(out)	None
Return value	uint16_t

1.2.8.2 Typedefs

Name	SSConfig
Type	struct
Description	A struct that holds all Speed Sensor configurations.

1.2.9 BCM

1.2.9.1 APIs

Name	BCM_Manager
Syntax	void BCM_Manager (Sensor_IdType SensorId)
Description	Manages the transmitted data through CAN Bus
Sync/Async	Synchronous
Reentrancy	Non-Reentrant
Parameters(in)	SensorId
Parameters(out)	None
Return value	void

1.2.9.2 Typedefs

Name	Sensor_IdType
Type	enum
Description	An enum holds the sensor Id.

1.3 Folder Structure

```
ECU1
├── Application
│   └── main.c
├── HAL
│   ├── Include
│   │   ├── Door_Sensor.h
│   │   ├── Light_Switch.h
│   │   └── Speed_Sensor.h
│   ├── Door_Sensor.c
│   ├── Light_Switch.c
│   └── Speed_Sensor.c
├── Library
│   ├── Bit_Math.h
│   ├── Platform_Types.h
│   └── Std_Types.h
├── MCAL
│   ├── ADC
│   │   ├── Config
│   │   │   ├── ADC_Cfg.h
│   │   │   └── ADC_Lcfg.c
│   │   ├── Include
│   │   │   ├── ADC.h
│   │   │   └── ADC_types.h
│   │   └── ADC.c
│   ├── CAN
│   └── Config
```



```

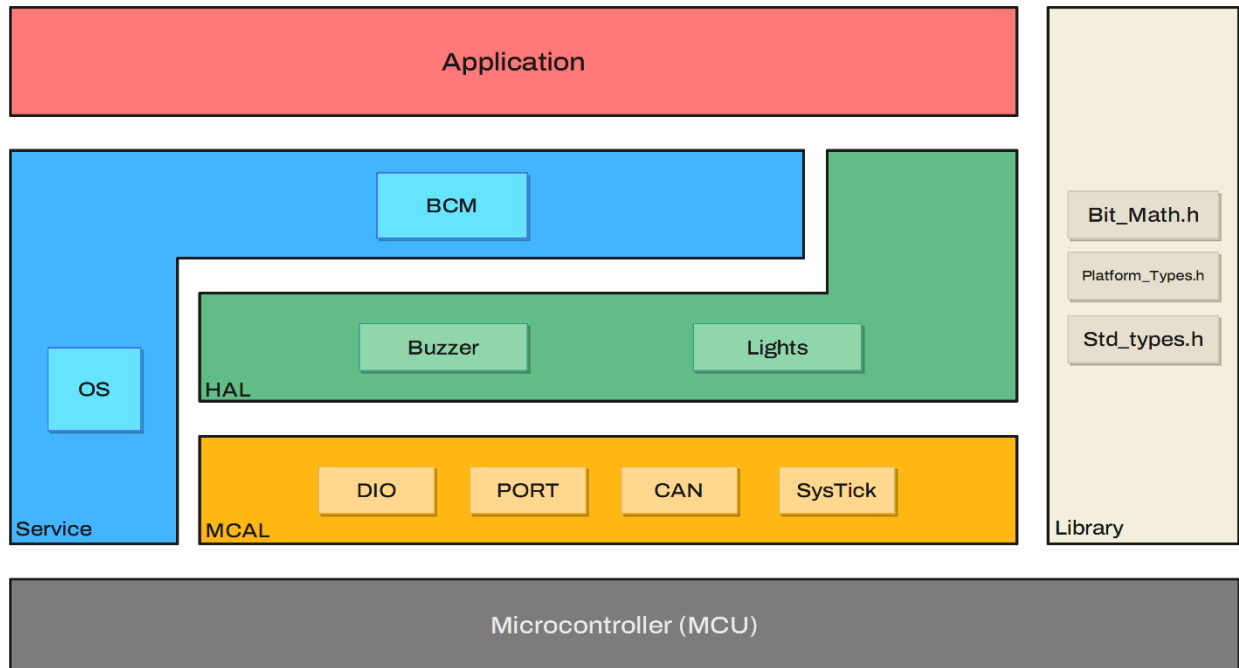
| | | └─ CAN_Cfg.h
| | | └─ CAN_Lcfg.c
| | └─ Include
| |   └─ CAN.h
| |   └─ CAN.c
| └─ DIO
|   └─ Include
|     └─ DIO.h
|     └─ DIO_Types.h
|     └─ DIO.c
| └─ PORT
|   └─ Config
|     └─ Port_Cfg.h
|     └─ Port_Lcfg.c
|   └─ Include
|     └─ Port.h
|     └─ Port_Types.h
|     └─ Port.c
| └─ SysTick
|   └─ Config
|     └─ SysTick_Cfg.h
|     └─ SysTick_Lcfg.c
|   └─ Include
|     └─ SysTick.h
|     └─ SysTick_Types.h
|     └─ SysTick.c
└─ Service
  └─ Include
    └─ BCM.h
    └─ OS.h
  └─ BCM.c
  └─ OS.c

```

2.ECU2

2.1 Layered Architecture

Making the layered architecture and specifying ECU2 components and modules.



2.2 APIs and Typedefs

2.2.1 PORT

2.2.1.1 APIs

Name	Port_Init
Syntax	void Port_Init(Port_ConfigType *PortConfigPtr)
Description	Initializes the Pin with the configured functionality
Sync/Async	Synchronous
Reentrancy	Non-Reentrant
Parameters(in)	PortConfigPtr
Parameters(out)	None
Return value	Void

2.2.1.2 Typedefs

Name	Port_ConfigType
Type	struct

Description	A struct holds the configurations of the pin including its port, number, mode, level, etc..
--------------------	---

2.2.2 DIO

2.2.2.1 APIs

Name	Dio_ReadChannel
Syntax	Dio_LevelType Dio_ReadChannel(Dio_PortType PortId, Dio_ChannelType ChannelId)
Description	Reads the DIO channel
Sync/Async	Synchronous
Reentrancy	Non-Reentrant
Parameters(in)	PortId, ChannelId
Parameters(out)	None
Return value	Dio_LevelType

Name	Dio_WriteChannel
Syntax	void Dio_WriteChannel(Dio_PortType PortId, Dio_ChannelType ChannelId, Dio_LevelType level)
Description	Writes to the DIO channel
Sync/Async	Synchronous
Reentrancy	Non-Reentrant
Parameters(in)	PortId, ChannelId, level
Parameters(out)	None
Return value	void

2.2.2.2 Typedefs

Name	Dio_LevelType
Type	enum
Description	An enum holds the level of the pin which is Low (0) or High (1).

Name	Dio_PortType
Type	enum
Description	An enum holds the ports of MCU such as Port_A, Port_B, etc..

Name	Dio_ChannelType
Type	enum
Description	An enum holds the pins of the port such as Pin_0, Pin_1, etc...

2.2.3 CAN

2.2.3.1 APIs

Name	CAN_Init
Syntax	void CAN_Init(CAN_ConfigType *CANConfigPtr)
Description	Initializes CAN module with the desired configurations
Sync/Async	Synchronous
Reentrancy	Non-Reentrant
Parameters(in)	CANConfigPtr
Parameters(out)	None
Return value	void

Name	CAN_Receive
Syntax	uint32_t CAN_Receive (void)
Description	Receives data through CAN bus
Sync/Async	Synchronous
Reentrancy	Non-Reentrant
Parameters(in)	void
Parameters(out)	None
Return value	uint32_t

2.2.3.2 Typedefs

Name	CAN_ConfigType
Type	struct
Description	A struct that holds all CAN configurations.

2.2.4 SysTick

2.2.4.1 APIs

Name	SysTick_Init
Syntax	void SysTick_Init(SysTick_ConfigType SysTickConfigPtr)
Description	Initializes SysTick timer with the desired configurations
Sync/Async	Synchronous
Reentrancy	Non-Reentrant
Parameters(in)	SysTickConfigPtr
Parameters(out)	None
Return value	void

2.2.4.2 Typedefs

Name	SysTick_ConfigType
Type	struct
Description	A struct that holds all SysTick configurations.

2.2.5 Buzzer

2.2.5.1 APIs

Name	Buzzer_Init
Syntax	void Buzzer_Init(Buzzer_ConfigType BuzzerConfig)
Description	Initializes Buzzer module with the desired configurations
Sync/Async	Synchronous
Reentrancy	Non-Reentrant
Parameters(in)	BuzzerConfig
Parameters(out)	None
Return value	void

Name	Buzzer_SetState
Syntax	void Buzzer_SetState(Buzzer_StateType Buzzer_State)
Description	Sets the state of Buzzer
Sync/Async	Synchronous
Reentrancy	Reentrant
Parameters(in)	Buzzer_State
Parameters(out)	None
Return value	void

2.2.5.2 Typedefs

Name	BuzzerConfig
Type	struct
Description	A struct that holds all Buzzer configurations.

Name	Buzzer_StateType
Type	enum
Description	An enum represents the Buzzer state either ON (1) or OFF (0).

2.2.6 Light

2.2.6.1 APIs

Name	Lights_Init
Syntax	Void Lights_Init(Lights_ConfigType LightsConfig)
Description	Initializes Lights module with the desired configurations
Sync/Async	Synchronous

Reentrancy	Reentrant
Parameters(in)	LightsConfig
Parameters(out)	None
Return value	void

Name	Lights_SetState
Syntax	void Lights_SetState(Lights_SideType Lights_Side, Lights_StateType Lights_State)
Description	Sets the state of Lights by its side
Sync/Async	Synchronous
Reentrancy	Reentrant
Parameters(in)	Lights_Side, Lights_State
Parameters(out)	None
Return value	void

2.2.6.2 Typedefs

Name	LightsConfig
Type	struct
Description	A struct that holds all Light configurations.

Name	Lights_SideType
Type	enum
Description	An enum represents the Lights state either Left (0) or Right (1).

Name	Lights_StateType
Type	enum
Description	An enum represents the Lights state either ON (1) or OFF (0).

2.2.7 BCM

2.2.7.1 APIs

Name	BCM_Manager
Syntax	void BCM_Manager (Sensor_IdType *SensorId, uint32_t *data)
Description	Manages the transmitted data through CAN Bus
Sync/Async	Synchronous
Reentrancy	Non-Reentrant
Parameters(in)	SensorId, data
Parameters(out)	None
Return value	void

2.2.7.2 Typedefs

Name	Sensor_IdType
Type	enum
Description	An enum holds the sensor Id.

2.3 Folder Structure

```
ECU2
├── Application
│   └── main.c
├── HAL
│   ├── Include
│   │   ├── Buzzer.h
│   │   └── Light.h
│   ├── Buzzer.c
│   └── Lights.c
├── Library
│   ├── Bit_Math.h
│   ├── Platform_Types.h
│   └── Std_Types.h
├── MCAL
│   ├── CAN
│   │   ├── Config
│   │   │   ├── CAN_Cfg.h
│   │   │   └── CAN_Lcfg.c
│   │   ├── Include
│   │   │   └── CAN.h
│   │   └── CAN.c
│   ├── DIO
│   │   ├── Include
│   │   │   ├── DIO.h
│   │   │   └── DIO_Types.h
│   │   └── DIO.c
│   ├── PORT
│   │   ├── Config
│   │   │   ├── Port_Cfg.h
│   │   │   └── Port_Lcfg.c
│   │   ├── Include
│   │   │   ├── Port.h
│   │   │   └── Port_Types.h
│   │   └── Port.c
│   └── SysTick
│       ├── Config
│       │   ├── SysTick_Cfg.h
│       │   └── SysTick_Lcfg.c
│       ├── Include
│       │   └── SysTick.h
```

```
| | | L SysTick_Types.h
| | L SysTick.c
L Service
| T Include
| | T BCM.h
| | L OS.h
| T BCM.c
| L OS.c
```