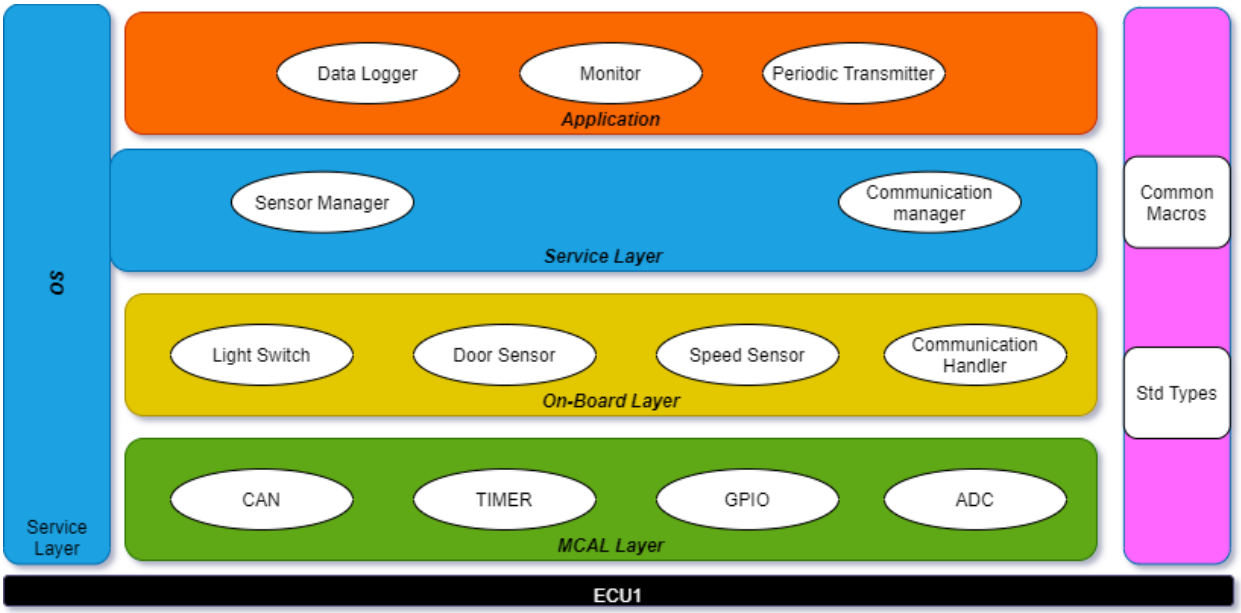
Static Design Analysis

ECU 1: -

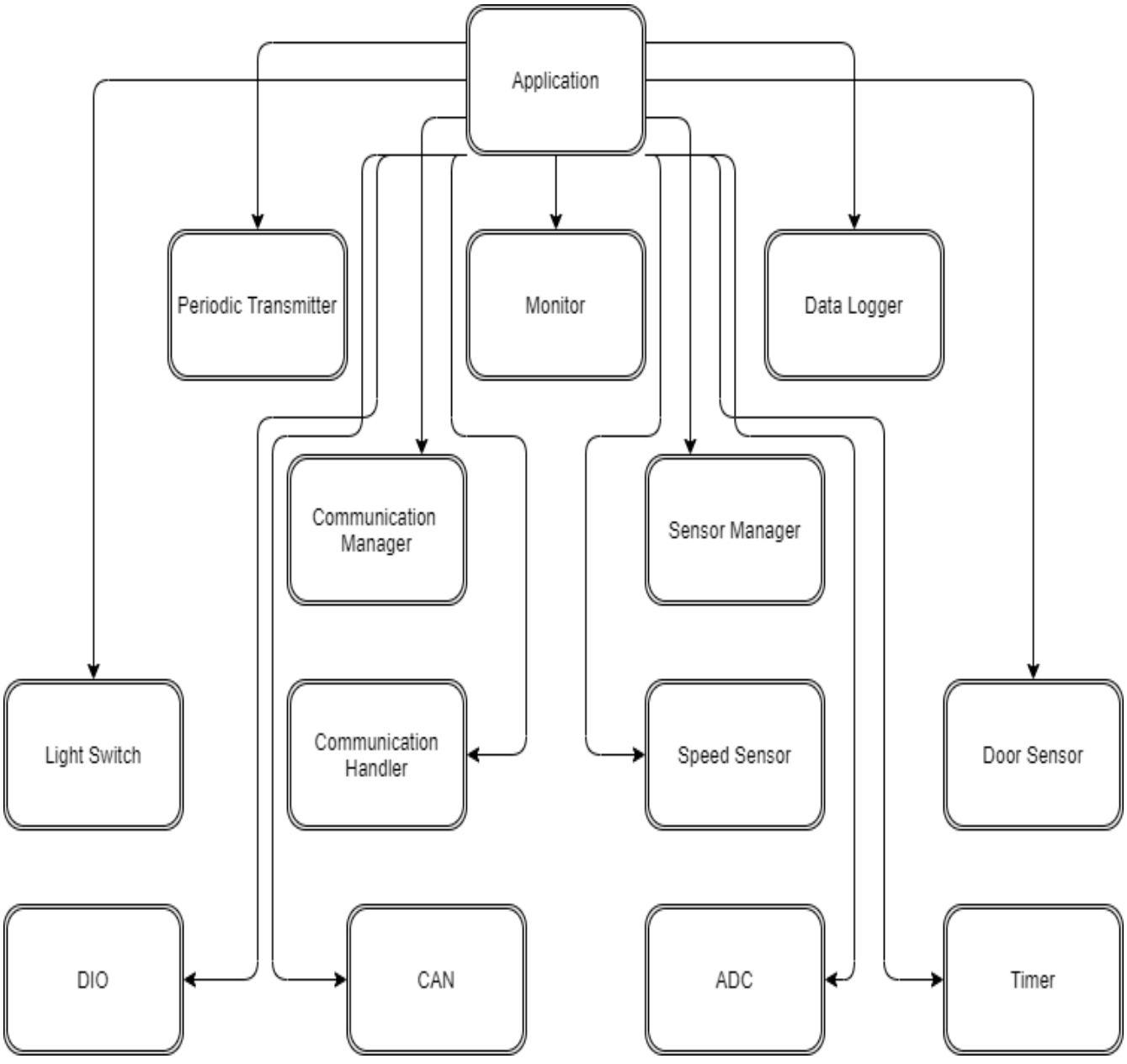
1. Layered Architecture



Notes:

- Door sensor receives digital value, high if closed and low if it's open.
- Speed sensor receives an analog value of car speed and then converts it to digital high or low.
- The sensor manager controls the light switcher to receive digital value.
- We have three tasks (door sensor, speed sensor, and light switch); each does the following monitor (Sensor Manager), log readings and periodic transmission via CAN bus (Communication Manager).

2. Modules


Application
Periodic Transmitter
Monitor
Data Logger
Communication Manager
Sensor Manager
Light Switch
Communication Handler
Speed Sensor
Door Sensor
DIO
CAN
ADC
Timer

3. ECU 1 APIs

MODULE	APIS
APPLICATION TASKS	doorSensorTask speedSensorTask lightSwitchTask

Syntax	void doorSensorTask (void)
Sync/Async	Synchronous
Reentrancy	Non-Reentrant
Parameters	None
Return	None
Description	Door Sensor Task

Syntax	void speedSensorTask (void)
Sync/Async	Synchronous
Reentrancy	Non-Reentrant
Parameters	None
Return	None
Description	Speed Sensor Task

Syntax	void lightSwitchTask (void)
Sync/Async	Synchronous
Reentrancy	Non-Reentrant
Parameters	None
Return	None
Description	Light Switch Task

MODULE	APIS
DATA LOGGER	dataLog_Save

Syntax	void dataLog_Save (uint64 data)
Sync/Async	Synchronous
Reentrancy	Non-Reentrant
Parameters	Data
Return	None
Description	Save the logged data.

MODULE	APIS
(BCM) COMMUNICATION MANAGER	BCM_Manager

Syntax	void BCM_Manager (uint64 Data, uint8 Bus_ID)
Sync/Async	Synchronous
Reentrancy	Non-Reentrant
Parameters	Data, Bus id
Return	None
Description	Coordinates bus communication requests

MODULE	APIS
SENSOR MANAGER	sensorManager

Syntax	Uint8 sensorManager (uint8 Sensor_ID)
Sync/Async	Synchronous
Reentrancy	Non-Reentrant
Parameters	Sensor_ID
Return	Sensor status
Description	Select the sensor to read the status

MODULE	APIS
COMMUNICATION HANDLER	BCM_Handler

Syntax	void BCM_Handler (uint64 data, uint8 Bus_ID)
Sync/Async	Synchronous
Reentrancy	Non-Reentrant
Parameters	Data, Bus_ID
Return	None
Description	Coordinates the bus communication requests (HW layer)

MODULE	APIS
SENSOR HANDLER	Sensor_Handler

Syntax	uint8 Sensor_Handler (uint8 Sensor_ID)
Sync/Async	Synchronous
Reentrancy	Non-Reentrant
Parameters	Sensor_ID
Return	Data
Description	Select the sensor to read the status(HW Layer)

MODULE	APIS
DOOR SENSOR	doorSensor_Init doorSensor_Status

Syntax	ERROR_STATE doorSensor_Init (void)
Sync/Async	Synchronous
Reentrancy	Non-Reentrant
Parameters	None
Return	ERROR_STATE
Description	Initialize Door Sensor

Syntax	uint8 doorSensor_Status (void)
Sync/Async	Synchronous
Reentrancy	Non-Reentrant
Parameters	None
Return	Sensor_Status
Description	Read the status of the door sensor to check whether the door is open or not

MODULE	APIS
SPEED SENSOR	speedSensor_Init speedSensor_Status

Syntax	ERROR_STATE speedSensor_Init (void)
Sync/Async	Synchronous
Reentrancy	Non-Reentrant
Parameters	None
Return	ERROR_STATE
Description	Initialize Speed Sensor

Syntax	uint8 speedSensor_Status (void)
Sync/Async	Synchronous
Reentrancy	Non-Reentrant
Parameters	None
Return	Sensor_Status
Description	Read the status of the speed sensor to check whether the car is moving or not

MODULE	APIS
LIGHT SWITCH	lightSwitch_Init lightSwitch_Status

Syntax	ERROR_STATE lightSwitch_Init (void)
Sync/Async	Synchronous
Reentrancy	Non-Reentrant
Parameters	None
Return	ERROR_STATE
Description	Initialize configurations of Light Switch

Syntax	uint8 lightSwitch_Status (void)
Sync/Async	Synchronous
Reentrancy	Non-Reentrant
Parameters	None
Return	Switch_Status
Description	Read the status of light switch to check whether it’s on or off

MODULE	APIS
DIO	DIO_Init DIO_WriteChannel DIO_ReadChannel

Syntax	ERROR_STATE DIO_Init (void)
Sync/Async	Synchronous
Reentrancy	Non-Reentrant
Parameters	None
Return	ERROR_STATE
Description	Initialize pins as Dio with required configuation

Syntax	void DIO_WriteChannel(DIO_PinType Pin_ID, DIO_PinLevel Value)
Sync/Async	Synchronous
Reentrancy	Non-Reentrant
Parameters	Pin_ID, Value
Return	None
Description	Write Value on selected pin.

Syntax	DIO_PinLevel DIO_ReadChannel(DIO_PinType Pin_ID)
Sync/Async	Synchronous
Reentrancy	Non-Reentrant
Parameters	Pin_ID
Return	Value
Description	Read Value from the selected pin.

MODULE	**APIS**
ADC	ADC_Init ADC_ReadChannel

Syntax	ERROR_STATE ADC_Init (uint8 Pin_ID)
Sync/Async	Synchronous
Reentrancy	Non-Reentrant
Parameters	Pin_ID
Return	ERROR_STATE
Description	Initialize selected pins as ADC input to read sensors

Syntax	uint8 ADC_ReadChannel(uint8 Pin_ID)
Sync/Async	Synchronous
Reentrancy	Non-Reentrant
Parameters	Pin_ID
Return	Value
Description	Read Value from the selected pin.

MODULE	APIS
TIMER	Timer_Init Timer_Start Timer_Stop

Syntax	ERROR_STATE Timer_Init (void)
Sync/Async	Synchronous
Reentrancy	Non-Reentrant
Parameters	None
Return	ERROR_STATE
Description	Initialize pins connected to Timer internally, initialize mode and other configurations

Syntax	void Timer_Start (TimerTickType ticks)
Sync/Async	Synchronous
Reentrancy	Non-Reentrant
Parameters	Ticks
Return	None
Description	Make Timer counter count till the tick achieve required period for sending status message

Syntax	void Timer_Stop (void)
Sync/Async	Synchronous
Reentrancy	Non-Reentrant
Parameters	None
Return	None
Description	Stop timer

MODULE	APIS
CAN	CAN_Init CAN_Transmit

Syntax	ERROR_STATE CAN_Init (void)
Sync/Async	Synchronous
Reentrancy	Non-Reentrant
Parameters	None
Return	ERROR_STATE
Description	Initialize CAN communication with required configuration

Syntax	void CAN_Transmit(DIO_PinType Pin_ID, uint64 Data)
Sync/Async	Synchronous
Reentrancy	Non-Reentrant
Parameters	Pin_ID, Data
Return	None
Description	Transmit data via CAN

MODULE	APIS
PERIODIC TRANSMITTER	periodicTransmitter_Init PeriodicTransmitter_Send

Syntax	ERROR_STATE periodicTransmitter_Init (void)
Sync/Async	Synchronous
Reentrancy	Non-Reentrant
Parameters	None
Return	ERROR_STATE
Description	Initialise periodic transmission

Syntax	void periodicTransmitter_Send(DIO_PinType Pin_ID, uint64 Data)
Sync/Async	Synchronous
Reentrancy	Non-Reentrant
Parameters	Pin_ID, Data
Return	None
Description	Send periodic status to ECU 1 via CAN.

4. Typedef

typedef unsigned char uint8

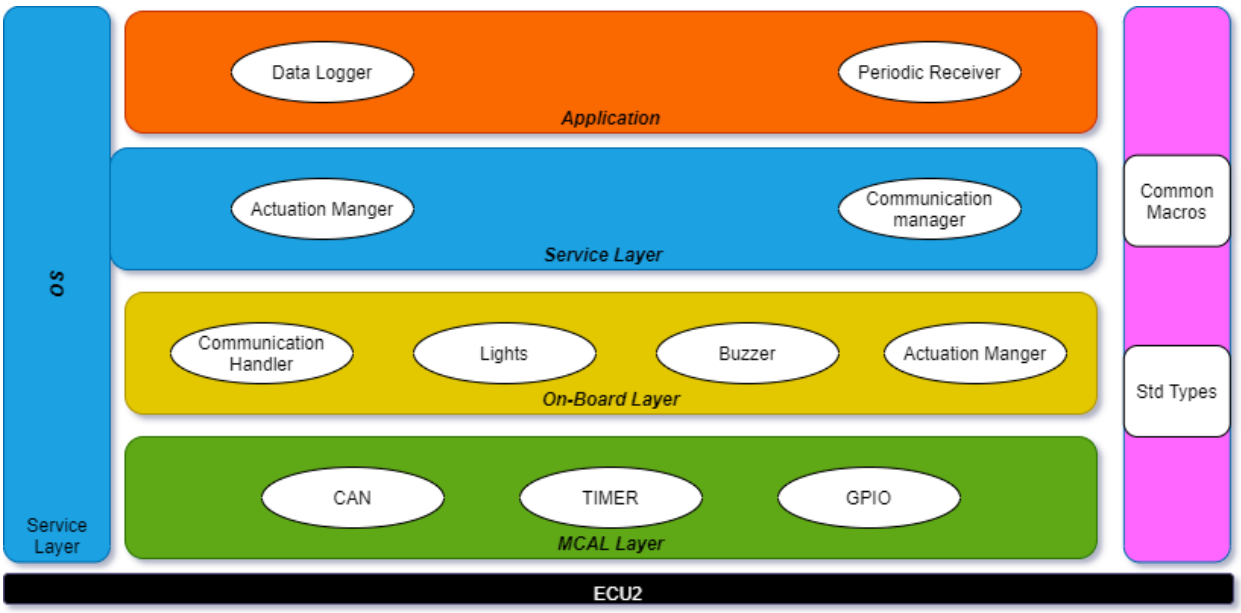
typedef unsigned long long uint64 CAN Frame

typedef uint8 DIO_Pin_Level HIGH = 1, LOW = 0

typedef uint8 DIO_Pin_Type OUTPUT = 1, LOW = 0

ECU 2: -

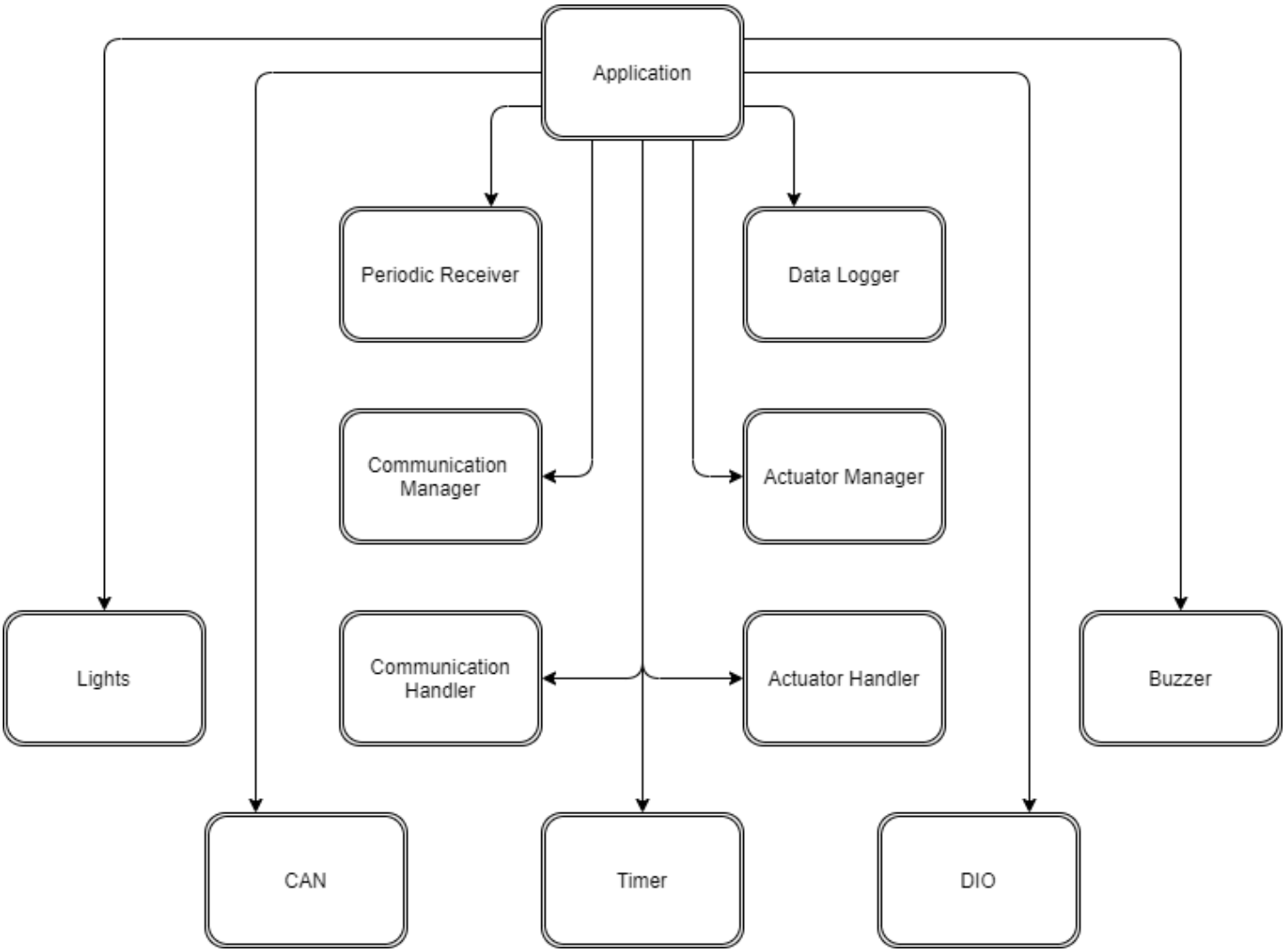
1. Layered Architecture



Notes

- We have one task that receives the status message via CAN bus and checks their values versus the initial value and does some changes based on the conditions

2. Modules


Application
Periodic Receiver
Data Logger
Communication Manager
Actuator Manager
Lights
Communication Handler
Actuator Handler
Buzzer
CAN
Timer
DIO

3. ECU 2 APIs

MODULE	APIS
DATA LOGGER	dataLog_Save

Syntax	void dataLog_Save (uint64 data)
Sync/Async	Synchronous
Reentrancy	Non-Reentrant
Parameters	Data
Return	None
Description	Save the logged data.

MODULE	APIS
(BCM) COMMUNICATION MANAGER	BCM_Manager

Syntax	void BCM_Manager (uint64 Data, uint8 Bus_ID)
Sync/Async	Synchronous
Reentrancy	Non-Reentrant
Parameters	Data, Bus id
Return	None
Description	Coordinates bus communication requests

MODULE	APIS
ACTUATOR MANAGER	actuatorManager

Syntax	void actutatorManager (uint8 Actuator_ID, uint8 Value)
Sync/Async	Synchronous
Reentrancy	Non-Reentrant
Parameters	Actuator_ID, Value
Return	None
Description	Select the actuator and change its value

MODULE	APIS
COMMUNICATION HANDLER	BCM_Handler

Syntax	void BCM_Handler (uint64 data, uint8 Bus_ID)
Sync/Async	Synchronous
Reentrancy	Non-Reentrant
Parameters	Data, Bus_ID
Return	None
Description	Coordinates bus communication requests(HW layer)

MODULE	APIS
ACTUATOR HANDLER	Actuator_Handler

Syntax	void Actuator_Handler (uint8 Actuator_ID, uint8 value)
Sync/Async	Synchronous
Reentrancy	Non-Reentrant
Parameters	Actutator_ID, Value
Return	None
Description	Read the selected actuator (HW Layer)

MODULE	APIS
BUZZER	Buzzer_Init Buzzer_Start Buzzer_Stop

Syntax	ERROR_STATE Buzzer_Init (void)
Sync/Async	Synchronous
Reentrancy	Non-Reentrant
Parameters	None
Return	ERROR_STATE
Description	Initialize the used pins for Digital Output

Syntax	void Buzzer_Start (void)
Sync/Async	Synchronous
Reentrancy	Non-Reentrant
Parameters	Ticks
Return	None
Description	Turn on the buzzer

Syntax	void Buzzer_Stop (void)
Sync/Async	Synchronous
Reentrancy	Non-Reentrant
Parameters	None
Return	None
Description	Stop Buzzer

MODULE	**APIS**
LIGHTS	Lights_Init Lights_ON Lights_OFF

Syntax	ERROR_STATE Lights_Init (void)
Sync/Async	Synchronous
Reentrancy	Non-Reentrant
Parameters	None
Return	ERROR_STATE
Description	Initialize the used pins for Digital Output

Syntax	void Lights_ON (void)
Sync/Async	Synchronous
Reentrancy	Non-Reentrant
Parameters	Ticks
Return	None
Description	Turn on the Lights

Syntax	void Lights_OFF (void)
Sync/Async	Synchronous
Reentrancy	Non-Reentrant
Parameters	None
Return	None
Description	Turn off Lights

MODULE	APIS
DIO	DIO_Init DIO_WriteChannel DIO_ReadChannel

Syntax	ERROR_STATE DIO_Init (void)
Sync/Async	Synchronous
Reentrancy	Non-Reentrant
Parameters	None
Return	ERROR_STATE
Description	Initialize DIO pins with the required configurations

Syntax	void DIO_WriteChannel(DIO_PinType Pin_ID, DIO_PinLevel Value)
Sync/Async	Synchronous
Reentrancy	Non-Reentrant
Parameters	Pin_ID, Value
Return	None
Description	Write Value on selected pin.

Syntax	DIO_PinLevel DIO_ReadChannel(DIO_PinType Pin_ID)
Sync/Async	Synchronous
Reentrancy	Non-Reentrant
Parameters	Pin_ID
Return	Value
Description	Read Value from the selected pin.

MODULE	APIS
TIMER	Timer_Init Timer_Start Timer_Stop

Syntax	ERROR_STATE Timer_Init (void)
Sync/Async	Synchronous
Reentrancy	Non-Reentrant
Parameters	None
Return	ERROR_STATE
Description	Initialize pins connected to Timer internally, initialize mode and other configurations

Syntax	void Timer_Start (TimerTickType ticks)
Sync/Async	Synchronous
Reentrancy	Non-Reentrant
Parameters	Ticks
Return	None
Description	Make Timer counter count till the tick achieve required period for sending status message

Syntax	void Timer_Stop (void)
Sync/Async	Synchronous
Reentrancy	Non-Reentrant
Parameters	None
Return	None
Description	Stop timer

MODULE	APIS
CAN	CAN_Init CAN_Receive

Syntax	ERROR_STATE CAN_Init (void)
Sync/Async	Synchronous
Reentrancy	Non-Reentrant
Parameters	None
Return	ERROR_STATE
Description	Initialize CAN with the required configurations

Syntax	uint64 CAN_Transmit(DIO_PinType Pin_ID)
Sync/Async	Synchronous
Reentrancy	Non-Reentrant
Parameters	Pin_ID, Data
Return	Data
Description	Receive data via CAN

MODULE	APIS
PERIODIC TRANSMITTER	periodicTransmitter_Init PeriodicTransmitter_Receive

Syntax	ERROR_STATE periodicTransmitter_Init (void)
Sync/Async	Synchronous
Reentrancy	Non-Reentrant
Parameters	None
Return	ERROR_STATE
Description	Initialize periodic transmission with the required configurations

Syntax	void periodicTransmitter_Receive(DIO_PinType Pin_ID, uint64* Data)
Sync/Async	Synchronous
Reentrancy	Non-Reentrant
Parameters	Pin_ID, Data pointer
Return	None
Description	Receive periodic status from ECU 2 via CAN

4. Typedef.

typedef unsigned char uint8	
typedef unsigned long long uint64	CAN Frame
typedef uint8 DIO_Pin_Type	OUTPUT = 1, INPUT = 0
typedef uint8 DIO_Pin_Level	HIGH = 1, LOW = 0
typedef uint8 CAN_Pin_Type	CANH = 1, CANL = 0