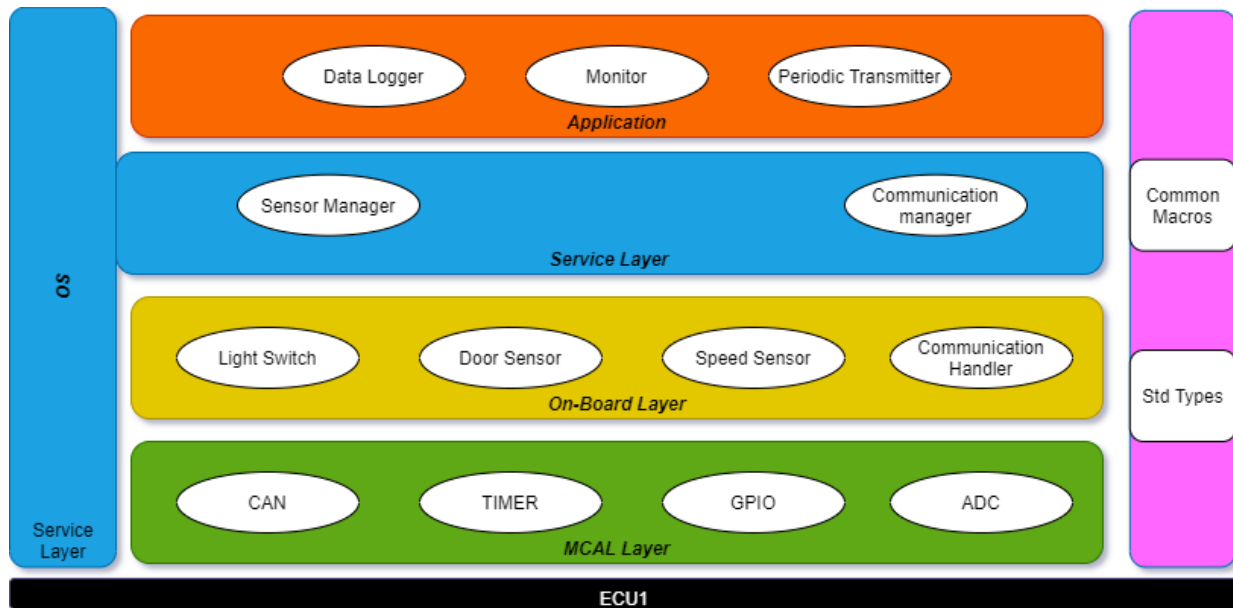# Static Design Analysis

ECU 1: -

1. <u>Layered Architecture</u>



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. <u>Modules</u>
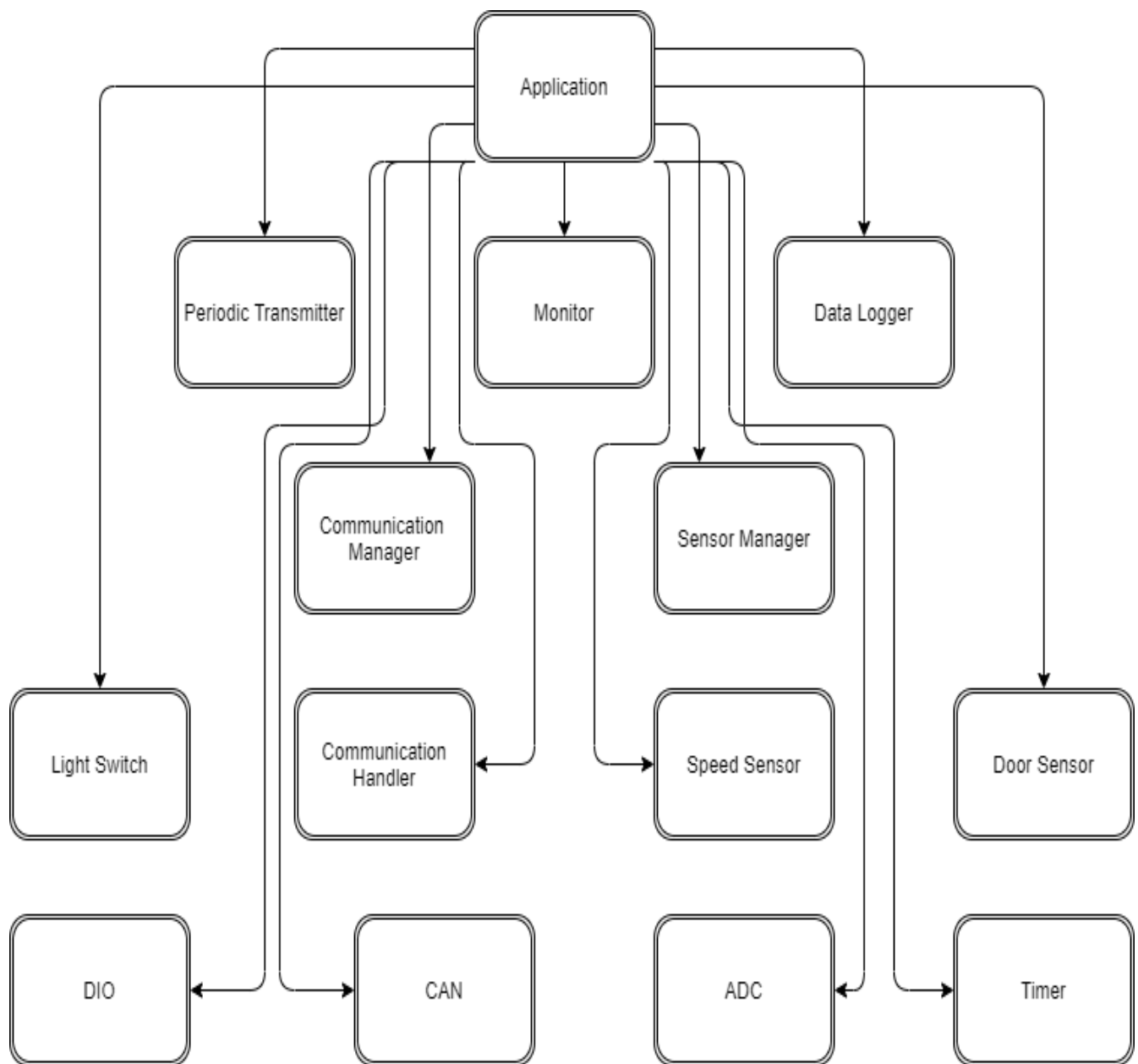
3. ECU 1 APIs

| MODULE | APIS |
|---|---|
| **APPLICATION TASKS** | doorSensorTask<br><br>speedSensorTask<br><br>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 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 | Read the selected sensor. |

| 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 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 | Read the selected sensor (HW Layer) |

| MODULE | APIS |
|---|---|
| **DOOR SENSOR** | doorSensor_Init<br><br>doorSensor_Status |

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

| Syntax | uint8 doorSensor_Status (void) |
|---|---|
| Sync/Async | Synchronous |
| Reentrancy | Non-Reentrant |
| Parameters | None |
| Return | Sensor_Status |
| Description | Read Door Sensor Status |

| MODULE | APIS |
|---|---|
| **SPEED SENSOR** | speedSensor_Init<br><br>speedSensor_Status |

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

| Syntax | uint8 speedSensor_Status (void) |
|---|---|
| Sync/Async | Synchronous |
| Reentrancy | Non-Reentrant |
| Parameters | None |
| Return | Sensor_Status |
| Description | Read Speed Sensor Status |

| MODULE | APIS |
|---|---|
| **LIGHT SWITCH** | lightSwitch_Init<br><br>lightSwitch_Status |

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

| Syntax | uint8 lightSwitch_Status (void) |
|---|---|
| Sync/Async | Synchronous |
| Reentrancy | Non-Reentrant |
| Parameters | None |
| Return | Switch_Status |
| Description | Read Light Switch Status |

| MODULE | APIS |
|---|---|
| **DIO** | DIO_Init<br><br>DIO_WriteChannel<br><br>DIO_ReadChannel |

| Syntax | ERROR_STATE DIO_Init (void) |
|---|---|
| Sync/Async | Synchronous |
| Reentrancy | Non-Reentrant |
| Parameters | None |
| Return | ERROR_STATE |
| Description | Initialise DIO pins |

| 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<br><br>ADC_ReadChannel |

| Syntax | ERROR_STATE ADC_Init (uint8 Pin_ID) |
|---|---|
| Sync/Async | Synchronous |
| Reentrancy | Non-Reentrant |
| Parameters | Pin_ID |
| Return | ERROR_STATE |
| Description | Initialise ADC to selected pins |

| 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<br><br>Timer_Start<br><br>Timer_Stop |

| Syntax | ERROR_STATE Timer_Init (void) |
|---|---|
| Sync/Async | Synchronous |
| Reentrancy | Non-Reentrant |
| Parameters | None |
| Return | ERROR_STATE |
| Description | Initialise internal timer |

| Syntax | void Timer_Start (TimerTickType ticks) |
|---|---|
| Sync/Async | Synchronous |
| Reentrancy | Non-Reentrant |
| Parameters | Ticks |
| Return | None |
| Description | Set desired ticks in the timer |

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

| MODULE | APIS |
|:---:|:---:|
| **CAN** | CAN_Init<br><br>CAN_Transmit |

| | |
|:---:|:---:|
| Syntax | ERROR_STATE CAN_Init (void) |
| Sync/Async | Synchronous |
| Reentrancy | Non-Reentrant |
| Parameters | None |
| Return | ERROR_STATE |
| Description | Initialise CAN communication |

| | |
|:---:|:---:|
| 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 |

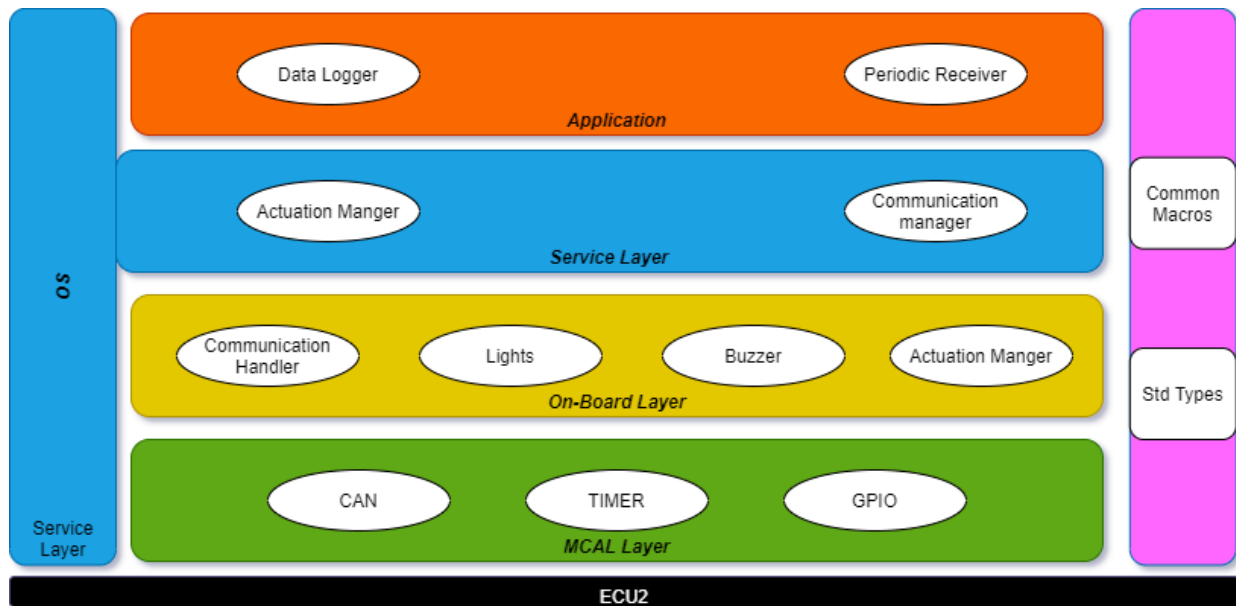| MODULE | APIS |
|:---:|:---:|
| **PERIODIC TRANSMITTER** | periodicTransmitter_Init<br><br>PeriodicTransmitter_Send |

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

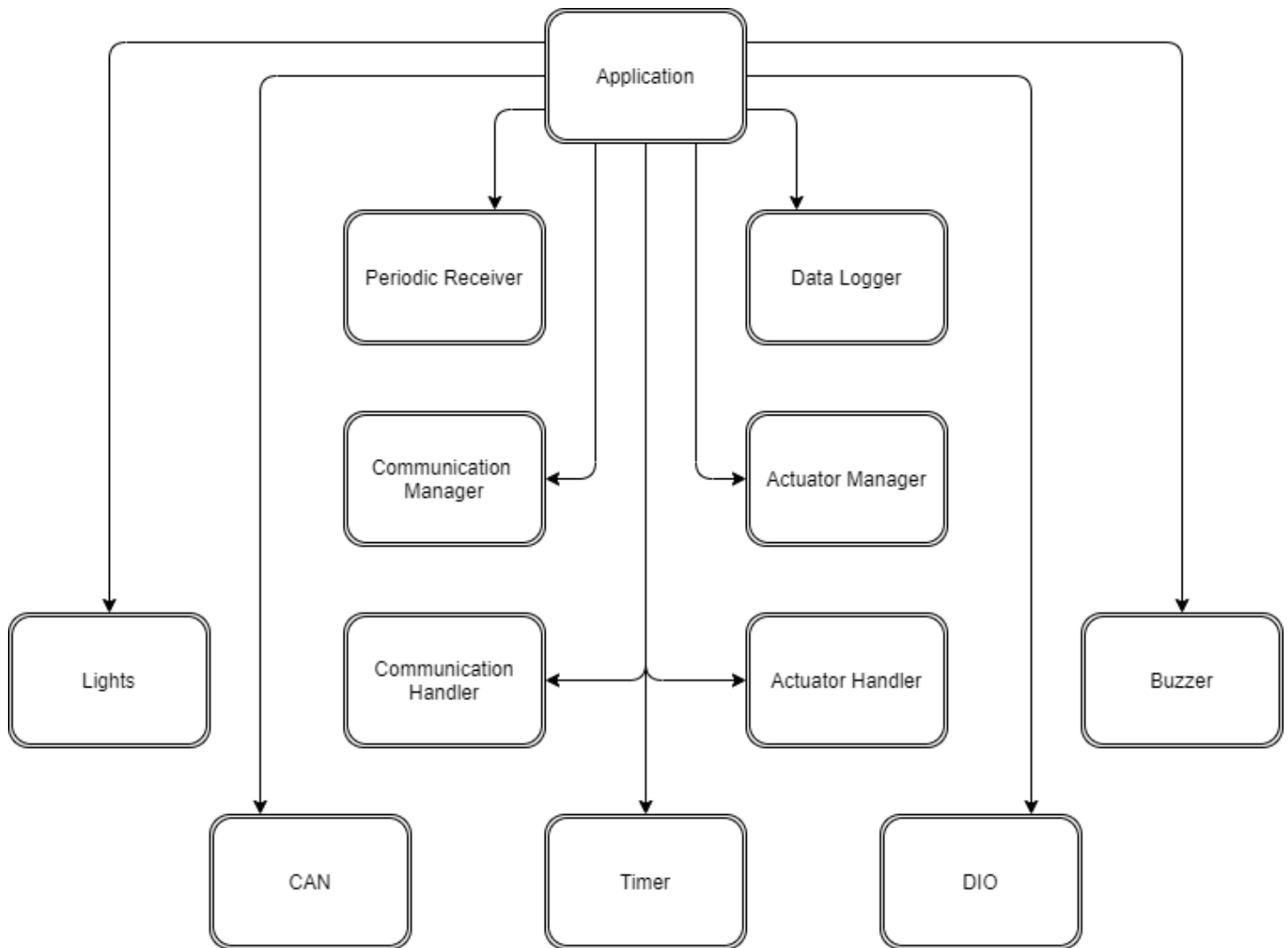| | |
|---|---|
| Syntax | void periodicTransmitter_Send(DIO_PinType Pin_ID, uint64 Data) |
| Sync/Async | Synchronous |
| Reentrancy | Non-Reentrant |
| Parameters | Pin_ID, Data |
| Return | None |
| Description | Send periodic status to ECU 1 via CAN. |

# 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

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 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 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<br><br>Buzzer_Start<br><br>Buzzer_Stop |

| | |
|---|---|
| Syntax | ERROR_STATE Buzzer_Init (void) |
| Sync/Async | Synchronous |
| Reentrancy | Non-Reentrant |
| Parameters | None |
| Return | ERROR_STATE |
| Description | Initialise Buzzer |

| 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<br><br>Lights_ON<br><br>Lights_OFF |

| Syntax | ERROR_STATE Lights_Init (void) |
|---|---|
| Sync/Async | Synchronous |
| Reentrancy | Non-Reentrant |
| Parameters | None |
| Return | ERROR_STATE |
| Description | Initialise the Lights |

| 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<br><br>DIO_WriteChannel<br><br>DIO_ReadChannel |

| Syntax | ERROR_STATE DIO_Init (void) |
|---|---|
| Sync/Async | Synchronous |
| Reentrancy | Non-Reentrant |
| Parameters | None |
| Return | ERROR_STATE |
| Description | Initialise DIO pins |

| 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<br><br>Timer_Start<br><br>Timer_Stop |

| Syntax | ERROR_STATE Timer_Init (void) |
|--------|-------------------------------|
| Sync/Async | Synchronous |
| Reentrancy | Non-Reentrant |
| Parameters | None |
| Return | ERROR_STATE |
| Description | Initialise internal timer |

| Syntax | void Timer_Start (TimerTickType ticks) |
|--------|----------------------------------------|
| Sync/Async | Synchronous |
| Reentrancy | Non-Reentrant |
| Parameters | Ticks |
| Return | None |
| Description | Set desired ticks in the timer |

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

| MODULE | APIS |
|--------|------|
| **CAN** | CAN_Init<br><br>CAN_Receive |

| Syntax | ERROR_STATE CAN_Init (void) |
|--------|------|
| Sync/Async | Synchronous |
| Reentrancy | Non-Reentrant |
| Parameters | None |
| Return | ERROR_STATE |
| Description | Initialise CAN communication |

| 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<br><br>PeriodicTransmitter_Receive |

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

| | |
|---|---|
| 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 to ECU 2 via CAN |