

Automotive Door Control System Design

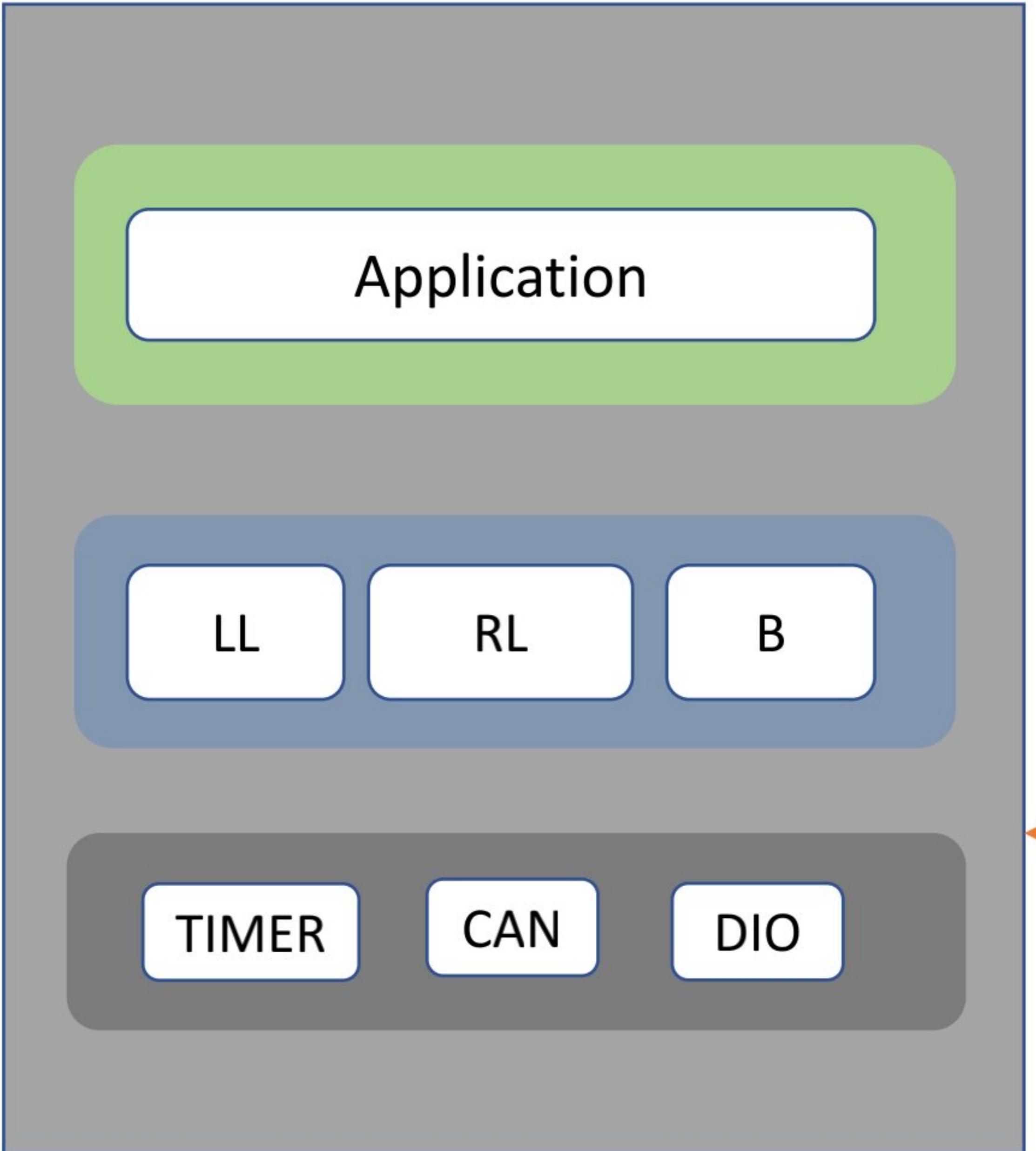
(Dynamic Design)

Name: Abdullah Mohamed

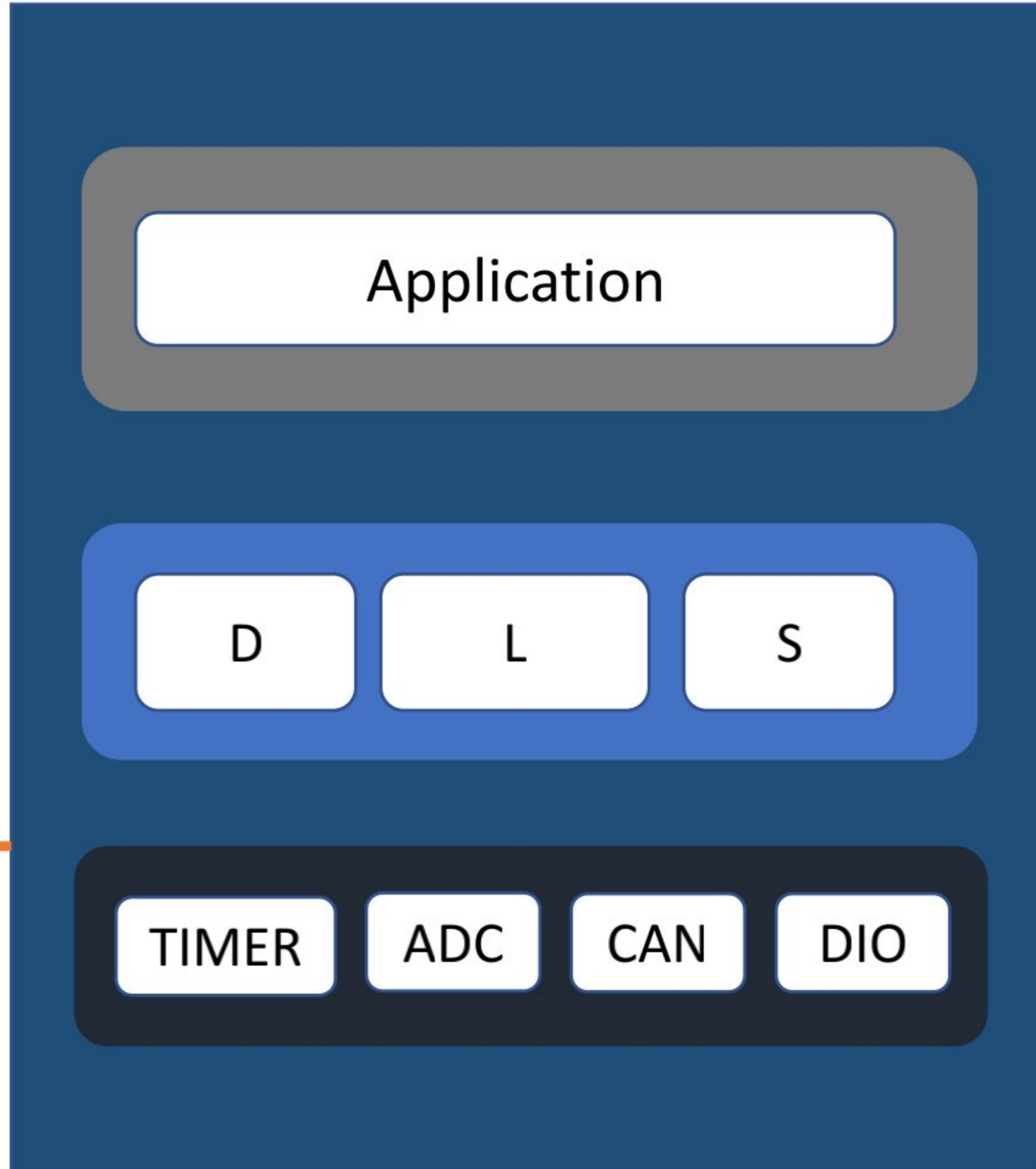
Email: mhamah50513@gmail.com

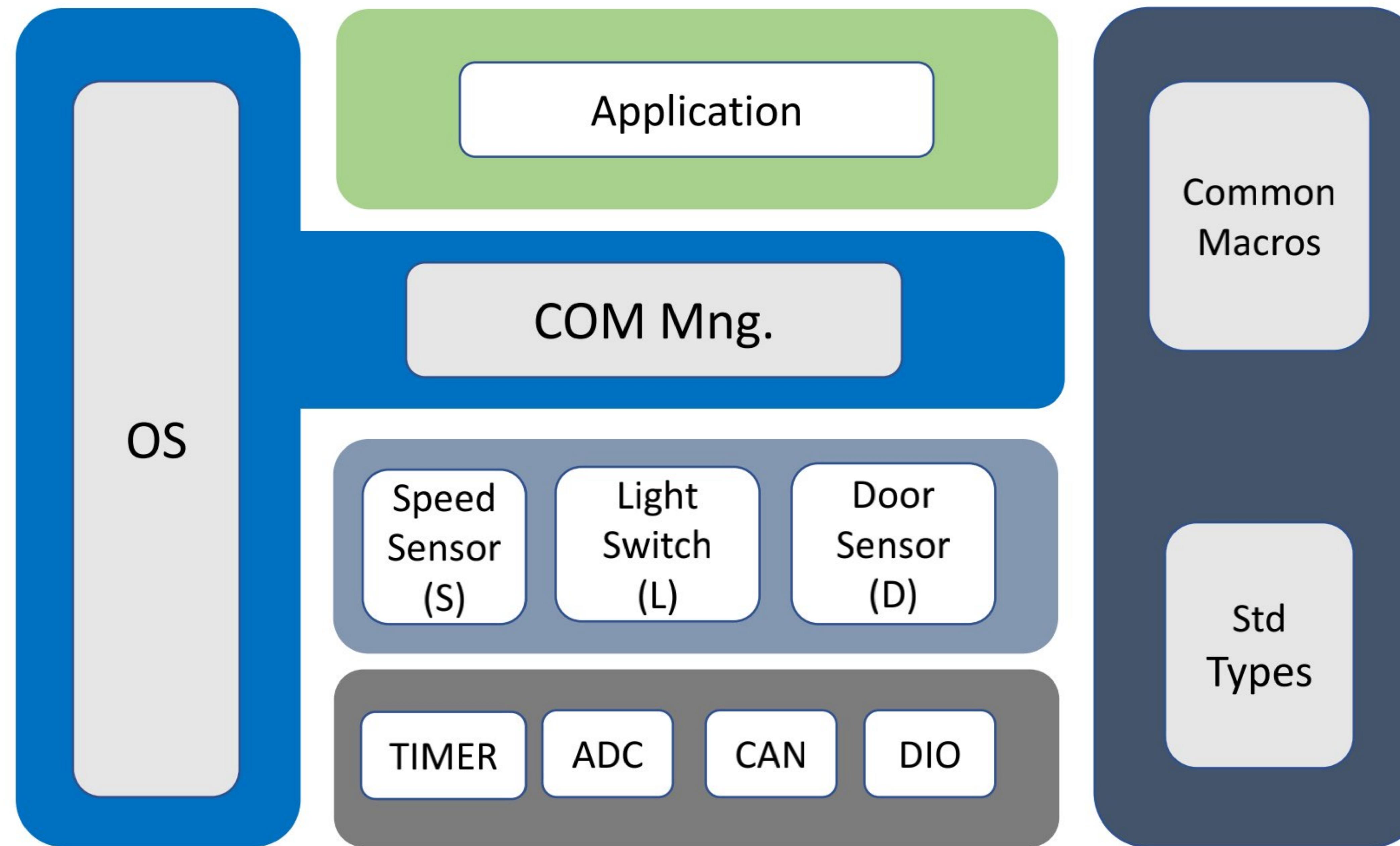
Static Design

ECU2



ECU1





ECU 1

DIO APIs:

| | | |
|----------------------------|---|---|
| Function Name | DIO_Init() | |
| API Type | Init | |
| Parameters (INPUT) | DIO_Port DIO_Channel DIO_PinLevel | |
| Parameters (OUTPUT) | None | |
| Return | E_OK | 0 |
| | E_NOK | 1 |
| Description | initialization the Dio module | |

| | | |
|----------------------------|----------------------------------|---|
| Function Name | DIO_Read() | |
| API Type | Getter | |
| Parameters (INPUT) | DIO_Port | |
| | DIO_Channel | |
| Parameters (OUTPUT) | DIO_PinLevel | |
| Return | E_OK | 0 |
| | E_NOK | 1 |
| Description | Reading the value of the channel | |

| | | |
|----------------------------|----------------------------------|---|
| Function Name | DIO_Write() | |
| API Type | Setter | |
| Parameters (INPUT) | DIO_Channel DIO_PinLevel | |
| Parameters (OUTPUT) | None | |
| Return | E_OK | 0 |
| | E_NOK | 1 |
| Description | Write on the channel low or high | |

| | |
|-------------|-----------------------------|
| Name | DIO_Port |
| Type | typedef enum |
| Range | {Port A to PortF } |
| Description | The decimal number for Port |

| | |
|-------------|----------------------------|
| Name | DIO_Channel |
| Type | typedef enum |
| Range | { PIN0 to PIN7} |
| Description | The decimal number for Pin |

| | | |
|-------------|--|--------------------------|
| Name | DIO_PinLevel | |
| Type | typedef enum | |
| Range | 0 | Low or Input Direction |
| | 1 | High or Output Direction |
| Description | The direction of the channel or the level on it. | |

Timer APIs:

| | | |
|----------------------------|---------------------------------|--|
| Function Name | TIMER_Inti() | |
| API Type | Init | |
| Parameters (INPUTS) | * ConfigPtr TIMER_ConfigType | |
| Parameters (OUTPUT) | None | |
| Return | E_OK 0 | |
| | E_NOK 1 | |
| Description | initialization the timer module | |

| | | |
|----------------------------|-------------------------|-------------------|
| Function Name | TIMER_Start() | |
| API Type | - | |
| Parameters (INPUTS) | Channel | TIMER_ChannelType |
| | Value | TIMER_ValueType |
| Parameters (OUTPUT) | None | |
| Return | E_OK | 0 |
| | E_NOK | 1 |
| Description | Start the timer channel | |

| | | |
|----------------------------|------------------------|-------------------|
| Function Name | TIMER_Stop() | |
| API Type | - | |
| Parameters (INPUTS) | Channel | TIMER_ChannelType |
| Parameters (OUTPUT) | None | |
| Return | E_OK | 0 |
| | E_NOK | 1 |
| Description | Stop the timer channel | |

| | |
|-------------|--------------------------|
| Name | TIMER_ChannelType |
| Type | Uint8_t |
| Description | The channel of the timer |

| | |
|-------------|--|
| Name | TIMER_ValueType |
| Type | Uint8_t |
| Description | Type for reading and setting the timer value number of ticks |

| | |
|-------------|--|
| Name | TIMER_ConfigType |
| Type | Structure |
| Description | This structure is including the configuration set required for initializing the timer module |

ADC APIs:

| | | |
|----------------------------|-------------------------------|--|
| Function Name | ADC_Init() | |
| API Type | Init | |
| Parameters (INPUTS) | * ConfigPtr ADC_ConfigType | |
| Parameters (OUTPUT) | None | |
| Return | E_OK 0 | |
| | E_NOK 1 | |
| Description | initialization the ADC module | |

| | | |
|----------------------------|--|-----------------|
| Function Name | ADC_Read () | |
| API Type | Init | |
| Parameters (INPUTS) | Channel | ADC_ChannelType |
| Parameters (OUTPUT) | None | |
| Return | E_OK | 0 |
| | E_NOK | 1 |
| Description | This API to read the value in ADC registers and return it. | |

| | |
|-------------|---|
| Name | ADC_ChannelType |
| Type | Uint8_t |
| Description | This the data of struct including config of ADC |

| | |
|-------------|-----------------------|
| Name | ADC_ConfigType |
| Type | structure |
| Description | |

CAN APIs:

| | | |
|----------------------------|-------------------------------|--------|
| Function Name | CAN_Init() | |
| API Type | Init | |
| Parameters (INPUTS) | * ConfigPtr CAN_ConfigType | |
| Parameters (OUTPUT) | None | |
| Return | E_OK E_NOK | 0 1 |
| Description | Initializes the CAN Module | |

| | | |
|----------------------------|--------------------------------|----------|
| Function Name | CAN_Baudrate() | |
| API Type | | |
| Parameters (INPUTS) | Controller | Uint8_t |
| | Baudrate | Uint16_t |
| Parameters (OUTPUT) | None | |
| Return | E_OK | 0 |
| | E_NOK | 1 |
| Description | Set the baudrate to CAN Module | |

| | | |
|----------------------------|---------------------------------|----------|
| Function Name | CAN_SendData() | |
| API Type | - | |
| Parameters (INPUTS) | Data | Uint32_t |
| Parameters (OUTPUT) | None | |
| Return | E_OK | 0 |
| | E_NOK | 1 |
| Description | Send the data by the CAN Module | |

| | | |
|----------------------------|------------------------------|---|
| Function Name | CAN_ReceiveData() | |
| API Type | Getter | |
| Parameters (INPUTS) | void | |
| Parameters (OUTPUT) | None | |
| Return | E_OK | 0 |
| | E_NOK | 1 |
| Description | Receive data from CAN Module | |



| | |
|-------------|---|
| Name | CAN_ConfigType |
| Type | structure |
| Range | |
| Description | This Structure includes the configuration set required for initializing the CAN |

Door Sensor APIs:

| | | |
|----------------------------|------------------------------------|---|
| Function Name | DoorSen_Init() | |
| API Type | Init | |
| Parameters (INPUTS) | None | |
| Parameters (OUTPUT) | None | |
| Return | E_OK | 0 |
| | E_NOK | 1 |
| Description | Initializes the door sensor module | |



| | | |
|----------------------------|-------------------------------------|---|
| Function Name | DoorSen_ReadValue() | |
| API Type | Getter | |
| Parameters (INPUTS) | None | |
| Parameters (OUTPUT) | None | |
| Return | E_OK | 0 |
| | E_NOK | 1 |
| Description | Get the state of door sensor module | |



Light Switch APIs:

| | | |
|----------------------------|-------------------------------------|---|
| Function Name | LightSW_Init() | |
| API Type | Init | |
| Parameters (INPUTS) | None | |
| Parameters (OUTPUT) | None | |
| Return | E_OK | 0 |
| | E_NOK | 1 |
| Description | Initializes the Light Switch module | |



| | | |
|----------------------------|--------------------------------------|---|
| Function Name | LightSW_ReadValue() | |
| API Type | Init | |
| Parameters (INPUTS) | None | |
| Parameters (OUTPUT) | None | |
| Return | E_OK | 0 |
| | E_NOK | 1 |
| Description | Get the state of Light Switch module | |

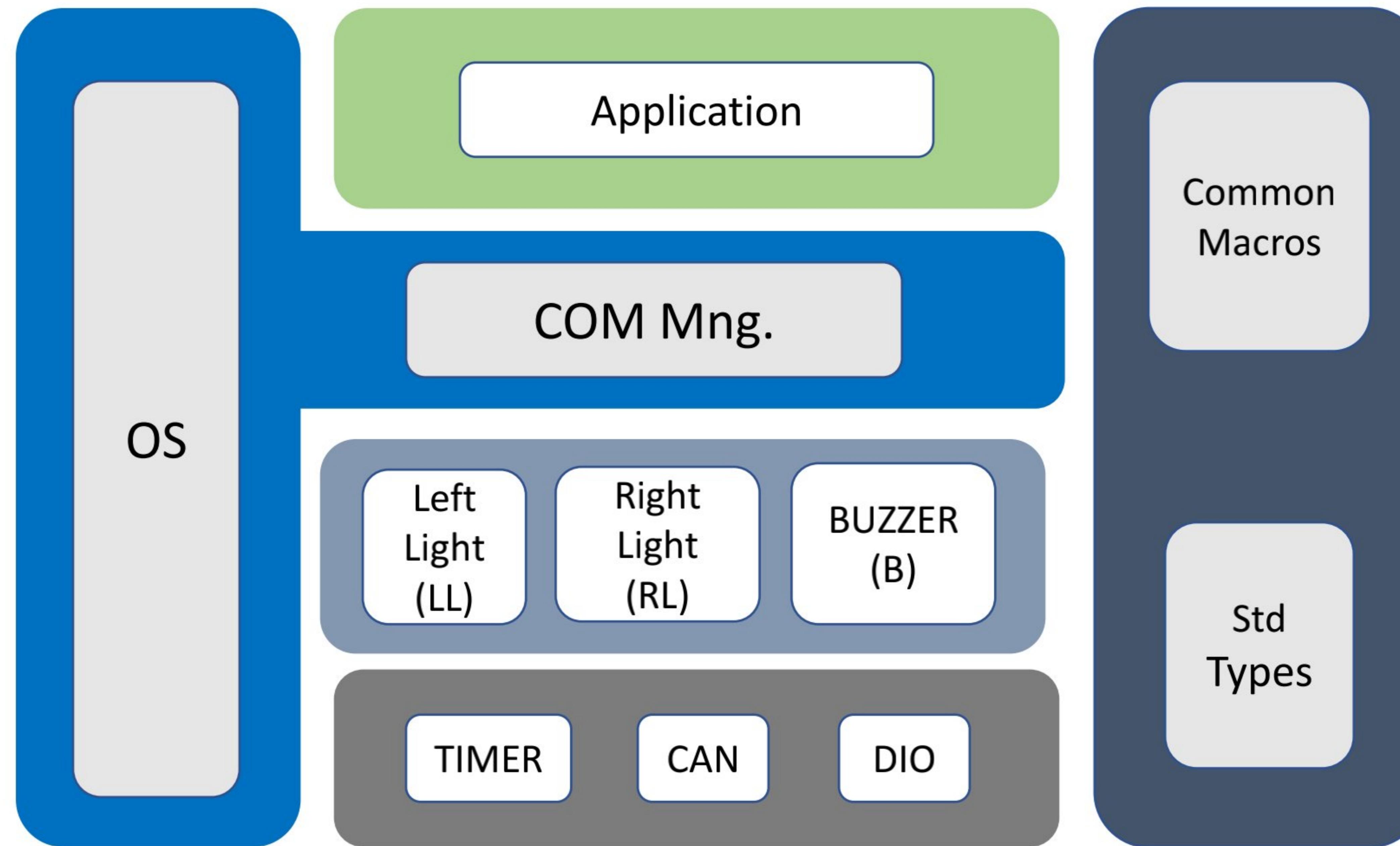


Speed Sensor APIs:

| | | |
|----------------------------|------------------------------|---|
| Function Name | SpeedSen_Init() | |
| API Type | Init | |
| Parameters (INPUTS) | None | |
| Parameters (OUTPUT) | None | |
| Return | E_OK | 0 |
| | E_NOK | 1 |
| Description | Initializes the timer module | |

| | | |
|----------------------------|--------------------------------------|---|
| Function Name | SpeedSen_ReadValue() | |
| API Type | Init | |
| Parameters (INPUTS) | None | |
| Parameters (OUTPUT) | None | |
| Return | E_OK | 0 |
| | E_NOK | 1 |
| Description | Get the state of Speed Sensor module | |





ECU 2

DIO APIs:

| | | |
|----------------------------|---|---|
| Function Name | DIO_Init() | |
| API Type | Init | |
| Parameters (INPUT) | DIO_Port DIO_Channel DIO_PinLevel | |
| Parameters (OUTPUT) | None | |
| Return | E_OK | 0 |
| | E_NOK | 1 |
| Description | initialization the Dio module | |

| | | |
|----------------------------|----------------------------------|---|
| Function Name | DIO_Read() | |
| API Type | Getter | |
| Parameters (INPUT) | DIO_Port | |
| | DIO_Channel | |
| Parameters (OUTPUT) | DIO_PinLevel | |
| Return | E_OK | 0 |
| | E_NOK | 1 |
| Description | Reading the value of the channel | |

| | | |
|----------------------------|----------------------------------|---|
| Function Name | DIO_Write() | |
| API Type | Setter | |
| Parameters (INPUT) | DIO_Channel DIO_PinLevel | |
| Parameters (OUTPUT) | None | |
| Return | E_OK | 0 |
| | E_NOK | 1 |
| Description | Write on the channel low or high | |

| | |
|-------------|-----------------------------|
| Name | DIO_Port |
| Type | typedef enum |
| Range | {Port A to PortF } |
| Description | The decimal number for Port |

| | |
|-------------|----------------------------|
| Name | DIO_Channel |
| Type | typedef enum |
| Range | { PIN0 to PIN7} |
| Description | The decimal number for Pin |

| | | |
|-------------|--|--------------------------|
| Name | DIO_PinLevel | |
| Type | typedef enum | |
| Range | 0 | Low or Input Direction |
| | 1 | High or Output Direction |
| Description | The direction of the channel or the level on it. | |

Timer APIs:

| | | |
|----------------------------|---------------------------------|------------------|
| Function Name | TIMER_Inti() | |
| API Type | Init | |
| Parameters (INPUTS) | * ConfigPtr | TIMER_ConfigType |
| Parameters (OUTPUT) | None | |
| Return | E_OK | 0 |
| | E_NOK | 1 |
| Description | initialization the timer module | |

| | | |
|----------------------------|-------------------------|-------------------|
| Function Name | TIMER_Start() | |
| API Type | - | |
| Parameters (INPUTS) | Channel | TIMER_ChannelType |
| | Value | TIMER_ValueType |
| Parameters (OUTPUT) | None | |
| Return | E_OK | 0 |
| | E_NOK | 1 |
| Description | Start the timer channel | |

| | | |
|----------------------------|------------------------|-------------------|
| Function Name | TIMER_Stop() | |
| API Type | - | |
| Parameters (INPUTS) | Channel | TIMER_ChannelType |
| Parameters (OUTPUT) | None | |
| Return | E_OK | 0 |
| | E_NOK | 1 |
| Description | Stop the timer channel | |

| | |
|-------------|--------------------------|
| Name | TIMER_ChannelType |
| Type | Uint8_t |
| Description | The channel of the timer |

| | |
|-------------|--|
| Name | TIMER_ValueType |
| Type | Uint8_t |
| Description | Type for reading and setting the timer value number of ticks |

| | |
|-------------|--|
| Name | TIMER_ConfigType |
| Type | Structure |
| Description | This structure is including the configuration set required for initializing the timer module |

CAN APIs:

| | | |
|----------------------------|-------------------------------|--------|
| Function Name | CAN_Init() | |
| API Type | Init | |
| Parameters (INPUTS) | * ConfigPtr CAN_ConfigType | |
| Parameters (OUTPUT) | None | |
| Return | E_OK E_NOK | 0 1 |
| Description | Initializes the CAN Module | |

| | | |
|----------------------------|--------------------------------|----------|
| Function Name | CAN_Baudrate() | |
| API Type | | |
| Parameters (INPUTS) | Controller | Uint8_t |
| | Baudrate | Uint16_t |
| Parameters (OUTPUT) | None | |
| Return | E_OK | 0 |
| | E_NOK | 1 |
| Description | Set the baudrate to CAN Module | |

| | | |
|----------------------------|---------------------------------|----------|
| Function Name | CAN_SendData() | |
| API Type | - | |
| Parameters (INPUTS) | Data | Uint32_t |
| Parameters (OUTPUT) | None | |
| Return | E_OK | 0 |
| | E_NOK | 1 |
| Description | Send the data by the CAN Module | |

| | | |
|----------------------------|------------------------------|---|
| Function Name | CAN_ReceiveData() | |
| API Type | Getter | |
| Parameters (INPUTS) | void | |
| Parameters (OUTPUT) | None | |
| Return | E_OK | 0 |
| | E_NOK | 1 |
| Description | Receive data from CAN Module | |

| | |
|-------------|---|
| Name | CAN_ConfigType |
| Type | structure |
| Range | |
| Description | This Structure includes the configuration set required for initializing the CAN |

Light Right(LR) APIs:

| | | |
|----------------------------|-----------------------------|---|
| Function Name | LR_Init() | |
| API Type | - | |
| Parameters (INPUTS) | DIO_Port , DIO_Pin | |
| Parameters (OUTPUT) | None | |
| Return | E_OK | 0 |
| | E_NOK | 1 |
| Description | Initializes the Light Right | |

| | | |
|----------------------------|---------------------|---|
| Function Name | LR_ON() | |
| API Type | - | |
| Parameters (INPUTS) | DIO_Port , DIO_Pin | |
| Parameters (OUTPUT) | None | |
| Return | E_OK | 0 |
| | E_NOK | 1 |
| Description | make Light right on | |

| | | |
|----------------------------|----------------------|---|
| Function Name | LR_OFF() | |
| API Type | - | |
| Parameters (INPUTS) | DIO_Port , DIO_Pin | |
| Parameters (OUTPUT) | None | |
| Return | E_OK | 0 |
| | E_NOK | 1 |
| Description | Make Light right off | |

Light Left (LL) APIs:

| | | |
|----------------------------|----------------------------|---|
| Function Name | LL_Init() | |
| API Type | - | |
| Parameters (INPUTS) | DIO_Port , DIO_Pin | |
| Parameters (OUTPUT) | None | |
| Return | E_OK | 0 |
| | E_NOK | 1 |
| Description | Initializes the Light lift | |

| | | |
|----------------------------|--------------------|---|
| Function Name | LL_ON() | |
| API Type | - | |
| Parameters (INPUTS) | DIO_Port , DIO_Pin | |
| Parameters (OUTPUT) | None | |
| Return | E_OK | 0 |
| | E_NOK | 1 |
| Description | Make Light lift on | |

| | | |
|----------------------------|---------------------|---|
| Function Name | LL_OFF() | |
| API Type | - | |
| Parameters (INPUTS) | DIO_Port , DIO_Pin | |
| Parameters (OUTPUT) | None | |
| Return | E_OK | 0 |
| | E_NOK | 1 |
| Description | Make Light lift off | |

Buzzer (B) APIs:

| | | |
|----------------------------|---|---|
| Function Name | Buzzer_Init() | |
| API Type | Init | |
| Parameters (INPUTS) | DIO_Port , DIO_Pin | |
| Parameters (OUTPUT) | None | |
| Return | E_OK | 0 |
| | E_NOK | 1 |
| Description | Initializes the Buzzer module (make the pin output) | |

| | | |
|----------------------------|--------------------|---|
| Function Name | Buzzer_ON() | |
| API Type | - | |
| Parameters (INPUTS) | DIO_Port , DIO_Pin | |
| Parameters (OUTPUT) | None | |
| Return | E_OK | 0 |
| | E_NOK | 1 |
| Description | Turn on the buzzer | |

| | | |
|----------------------------|---------------------|---|
| Function Name | Buzzer_OFF() | |
| API Type | - | |
| Parameters (INPUTS) | DIO_Port , DIO_Pin | |
| Parameters (OUTPUT) | None | |
| Return | E_OK | 0 |
| | E_NOK | 1 |
| Description | Turn off the buzzer | |