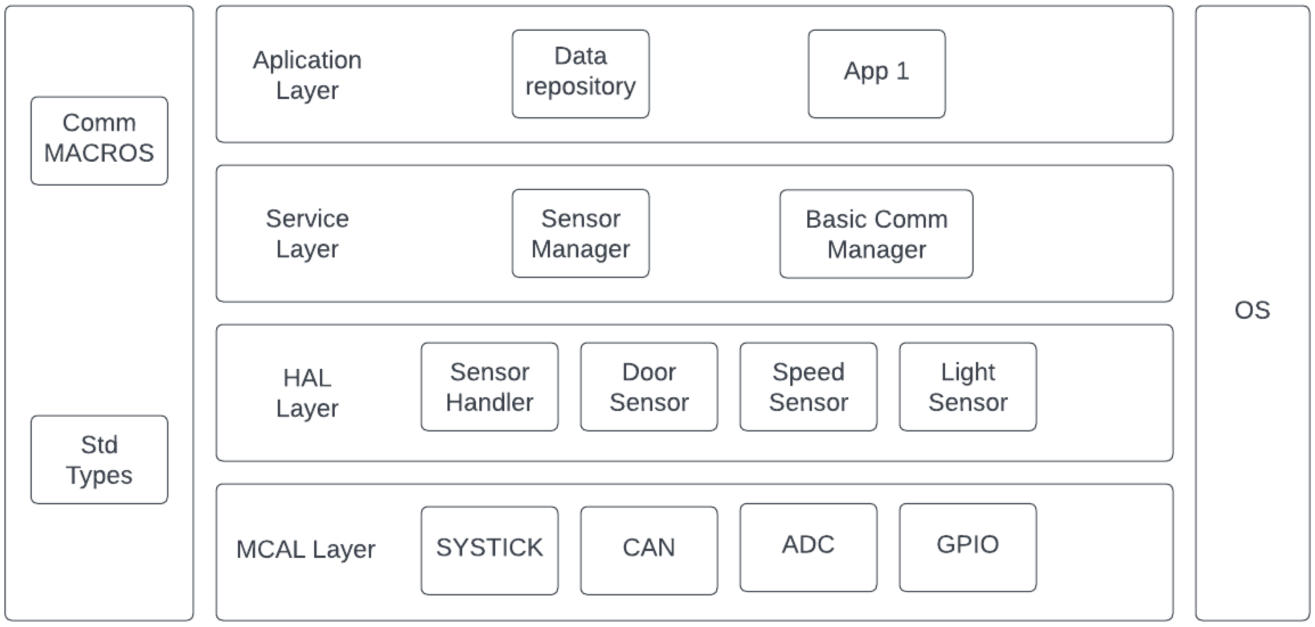
Automotive door control system design Static Design Analysis

Name: Youssef Mamdouh Abdelaty
Mail: Youssef.elkased@gmail.com

ECU 1
Layered Architecutre



1- APIs

- GPIO module:

Description	Initialize the GPIO with the structure configrations
Reentrancy	Non-reentrant
Synchronization	Synchronous
Type	init
Input	Pointer to sturcture
Output	void
Return	void

Description	Write the required GPIO port,Pin with the required value
Reentrancy	Non-reentrant
Synchronization	Synchronous
Type	Setter
Input	Port number - Pin number – pin value
Output	void
Return	void

Description	Read the required Gpio port , pin.
Reentrancy	Non-reentrant
Synchronization	Synchronous
Type	Getter
Input	Port number - Pin number
Output	void
Return	Uint8

- ADC Module:

Description	Initialize the ADC with the structure configrations
Reentrancy	Non-reentrant
Synchronization	Synchronous
Type	init
Input	Pointer to structure
Output	void
Return	void

Description	Read the required channel ID
Reentrancy	Non-reentrant
Synchronization	Synchronous
Type	getter
Input	channel ID
Output	void
Return	uint32

- CAN Module:

void CAN_init (struct * Config_ptr);	
Description	Initialize CAN bus with the structure configrations
Reentrancy	Non-reentrant
Synchronization	Synchronous
Type	init
Input	Pointer to structure
Output	void
Return	void

void CAN_transmit (uint8 CanPin_ID, uint64 Message);	
Description	Send a required message via required pin ID
Reentrancy	Non-reentrant
Synchronization	Synchronous
Type	transmit
Input	Can Pin number - Message
Output	void
Return	void

- Speed Sensor Module:

void SpeedSensor_init (struct * Config_ptr);	
Description	Initialize the speed sensor pin via ADC
Reentrancy	Non-reentrant
Synchronization	Synchronous
Type	init
Input	Pointer to stucture
Output	void
Return	void

Uint16 SpeedSensor_getSpeed (void);	
Description	Get the speed from the speed sensor via ADC
Reentrancy	Non-reentrant
Synchronization	Synchronous
Type	getter
Input	Pointer to stucture
Output	void
Return	Uint16

- Door Sensor Module:

uint8 DoorSensor_getStatus (void)	
Description	Read the door sensor status via GPIO
Reentrancy	Non-reentrant
Synchronization	Synchronous
Type	getter
Input	Pointer to structure
Output	void
Return	uint8

void DoorSensor_init (struct * Config_ptr);	
Description	Initialize the door sensor pin via GPIO
Reentrancy	Non-reentrant
Synchronization	Synchronous
Type	init
Input	Pointer to structure
Output	void
Return	void

- Light Switch Module:

void LightSwitch_init (struct * Config_ptr);	
Description	Initialize the light switch module with the structure
Reentrancy	Non-reentrant
Synchronization	Synchronous
Type	init
Input	Pointer to structure
Output	void
Return	void

uint8 LightSwitch_getStatus (void);	
Description	Read the light swich status
Reentrancy	Non-reentrant
Synchronization	Synchronous
Type	getter
Input	void
Output	void
Return	Uint8

- Sensor handler Module:

uint32 Sensor_handler (uint8 Sensor_ID);	
Description	chooses which sensor to read from hardware layer
Reentrancy	Non-reentrant
Synchronization	Synchronous
Type	setter
Input	Sensor ID
Output	void
Return	Uint32

- Communication handler module:

void BCM_handler (uint64 handler_Message, uint8 bus);	
Description	Choose which bus to send the required message
Reentrancy	Non-reentrant
Synchronization	Synchronous
Type	setter
Input	Message – bus
Output	void
Return	void

- Sensor manager Module:

uint32 Sensor_manager (uint8 sensor_Id);	
Description	Allow the application layer to choose the required sensor
Reentrancy	Non-reentrant
Synchronization	Synchronous
Type	setter
Input	Sensor ID
Output	void
Return	Uint32

- Basic Communication manager Module:

Void BCM_mananger (uint64 Manager_Message, uint8 bus);	
Description	Allow the application layer to choose the required bus
Reentrancy	Non-reentrant
Synchronization	Synchronous
Type	setter
Input	Message – bus
Output	void
Return	void

- Application1 Module :

void SendSpeed (void);	
Description	Send the speed sensor state to ECU2 via CAN bus
Reentrancy	Non-reentrant
Synchronization	Synchronous
Type	send
Input	void
Output	void
Return	void

void SendDoorState (void);	
Description	Send the door sensor state to ECU2 via CAN bus
Reentrancy	Non-reentrant
Synchronization	Synchronous
Type	send
Input	void
Output	void
Return	void

void SendLightSwitchState (void);	
Description	Send the light switch state to ECU2 via CAN bus
Reentrancy	Non-reentrant
Synchronization	Synchronous
Type	send
Input	void
Output	void
Return	void

3- Folder Structure:

MCAL	HAL	Service
Systick.c	Sensor_Handler.c	OS.c
ADC.c	Comm_Handler.c	Basic_Comm_mngr.c
CAN.c	Light_Switch.c	Sensors_mngr.c
GPIO.c	Door_sensor.c	
	Speed_sensor.c	
App	Config	
Data_repo.c	Systick_PBConfig.c	
App.c	ADC_PBConfig.c	
	CAN_PBConfig.c	
	GPIO_PBConfig.c	
	Switch_PBConfig.c	
	Door_PBConfig.c	
	Speed_PBConfig.c	

Common (Header Files) Folder:

Systick.h	ADC.h	CAN.h	GPIO.h
Sensor_handler.h	Comm_handler.h	Switch.h	Door.h
Speed.h	OS.h	App.h	Data_repo.h
Systick_Config.h	ADC_Config.h	CAN_Config.h	GPIO_Config.h
Switch_Config.h	Door_Config.h	Speed_Config.h	Sensor_mngr.h
Common_Macros.h	Std_lib.h	MCU_Regs.h	Comm_mngr.h

4- Drivers Structure:

1- GPIO Driver:

-GPIO.c

-GPIO.h

-GPIO_PBConfig.c

-GPIO_Config.h

2- ADC Driver:

-ADC.c

-ADC.h

-ADC_PBConfig.c

-ADC_Config.h

3- CAN Driver:

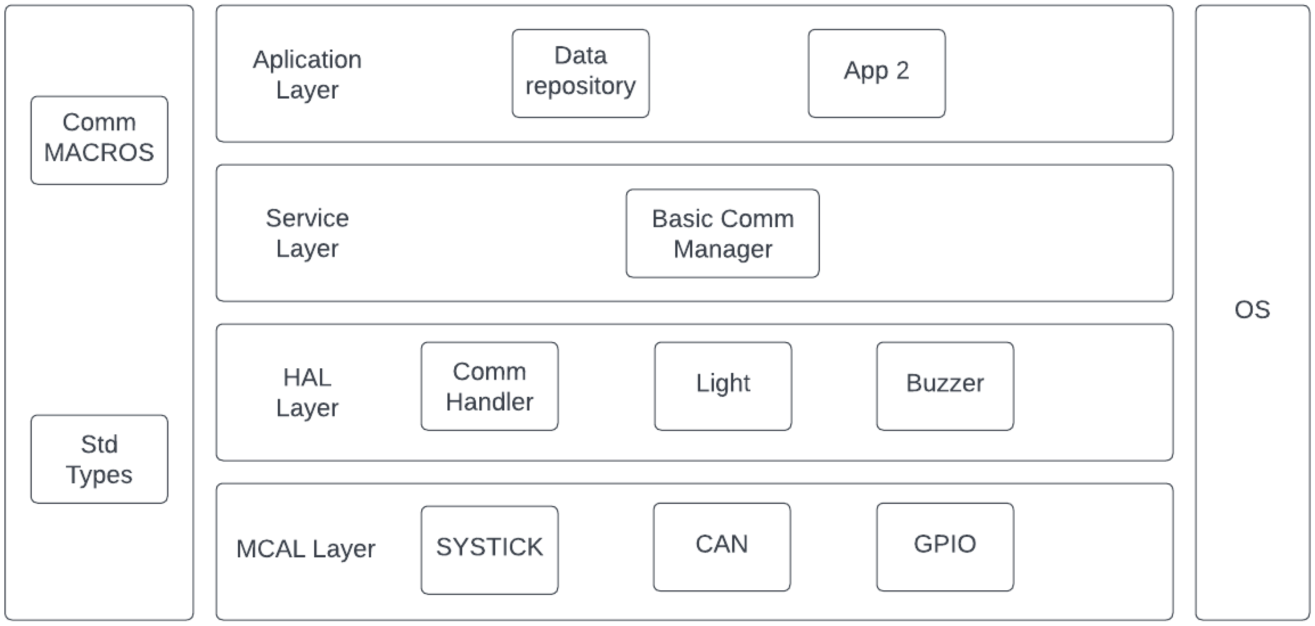
-CAN.c

-CAN.h

-CAN_PBConfig.c

-CAN_Config.h

ECU 2
Layered Architecutre



1- APIs

➢ **GPIO module:**

Description	Initialize the GPIO with the structure configrations
Reentrancy	Non-reentrant
Synchronization	Synchronous
Type	init
Input	Pointer to sturcture
Output	void
Return	void

Description	Write the required GPIO port,Pin with the required value
Reentrancy	Non-reentrant
Synchronization	Synchronous
Type	Setter
Input	Port number - Pin number – pin value
Output	void
Return	void

Description	Read the required Gpio port , pin.
Reentrancy	Non-reentrant
Synchronization	Synchronous
Type	Getter
Input	Port number - Pin number
Output	void
Return	Uint8

- **CAN Module :**

void CAN_init (struct * Config_ptr);	
Description	Initialize CAN bus with the structure configrations
Reentrancy	Non-reentrant
Synchronization	Synchronous
Type	init
Input	Pointer to structure
Output	void
Return	void

Uint64 CAN_Receive (uint8 CAN_Pin_Id);	
Description	Receive the CAN message from the required Pin ID
Reentrancy	Non-reentrant
Synchronization	Synchronous
Type	receive
Input	void
Output	void
Return	Uint64

- **Buzzer Module:**

void BUZZER_on (void);	
Description	Set the buzzer on
Reentrancy	Non-reentrant
Synchronization	Synchronous
Type	setter
Input	void
Output	void
Return	void

void BUZZER_off (void);	
Description	Set the buzzer off
Reentrancy	Non-reentrant
Synchronization	Synchronous
Type	setter
Input	void
Output	void
Return	void

- **Communication handler module:**

void BCM_handler (uint64 handler_Message, uint8 bus);	
Description	Choose which bus to receive message
Reentrancy	Non-reentrant
Synchronization	Synchronous
Type	setter
Input	Message – bus
Output	void
Return	void

- **Basic Communication manager Module:**

uint64 BCM_mananger (uint8 bus);	
Description	Allow the application layer to choose which bus to read the message from
Reentrancy	Non-reentrant
Synchronization	Synchronous
Type	setter
Input	bus
Output	void
Return	Uint64

- **Light Module:**

void LightSwitch_init (struct * Config_ptr);	
Description	Initialize the light switch module with the structure
Reentrancy	Non-reentrant
Synchronization	Synchronous
Type	init
Input	Pointer to structure
Output	void
Return	void

void light_OFF (void);	
Description	Set the light off
Reentrancy	Non-reentrant
Synchronization	Synchronous
Type	setter
Input	void
Output	void
Return	void

- **Data repository Module:**

void Data_repository (uint64 data);	
Description	Save the required data
Reentrancy	Non-reentrant
Synchronization	Synchronous
Type	setter
Input	Data to be saved
Output	void
Return	Data to be saved

- **Application2 Module:**

void Receive_Message (void)	
Description	Receive the message from ECU1 periodically to take actions
Reentrancy	Non-reentrant
Synchronization	Synchronous
Type	setter
Input	void
Output	void
Return	void

2- Folder Structure

MCAL	**HAL**	**Service**
Systick.c	**Light.c**	**OS.c**
GPIO.c	**Comm_Handler.c**	**Basic_Comm_mngr.c**
CAN.c	**Buzzer.c**	
Application	**Config**	
Data_repo.c	**Systick_PBConfig.c**	
App2.c	**Light_PBConfig.c**	
	CAN_PBConfig.c	
	GPIO_PBConfig.c	
	Buzzer_PBConfig.c	

Common (Header files) Folder:

Systick.h	Light.h	CAN.h	GPIO.h
Buzzer.h	Comm_handler.h	OS.h	Comm_mngr.h
Data_repo.h	App2.h	Systick_Config.h	Light_Config.h
CAN_Config.h	GPIO_Config.h	Buzzer_Config.h	MCU_Regs.h
Common_Macros.h	Std_lib.h		

5- Drivers Structure:

1- Systick Driver:

-Systick.c

-Systick.h

-Systick_PBConfig.c

-Systick_Config.h

2- CAN Driver:

-CAN.c

-CAN.h

-CAN_PBConfig.c

-CAN_Config.h

3- GPIO Driver:

- **GPIO.C**
- **GPIO.H**
- **GPIO_PBConfig.c**
- **GPIO_Config.h**

Type defs:

```
- unsigned char uint8     - unsigned long long uint64

- unsigned long uint32    - unsigned short uint16
```