

Embedded Systems Advanced Nano Degree

Automotive door control system design

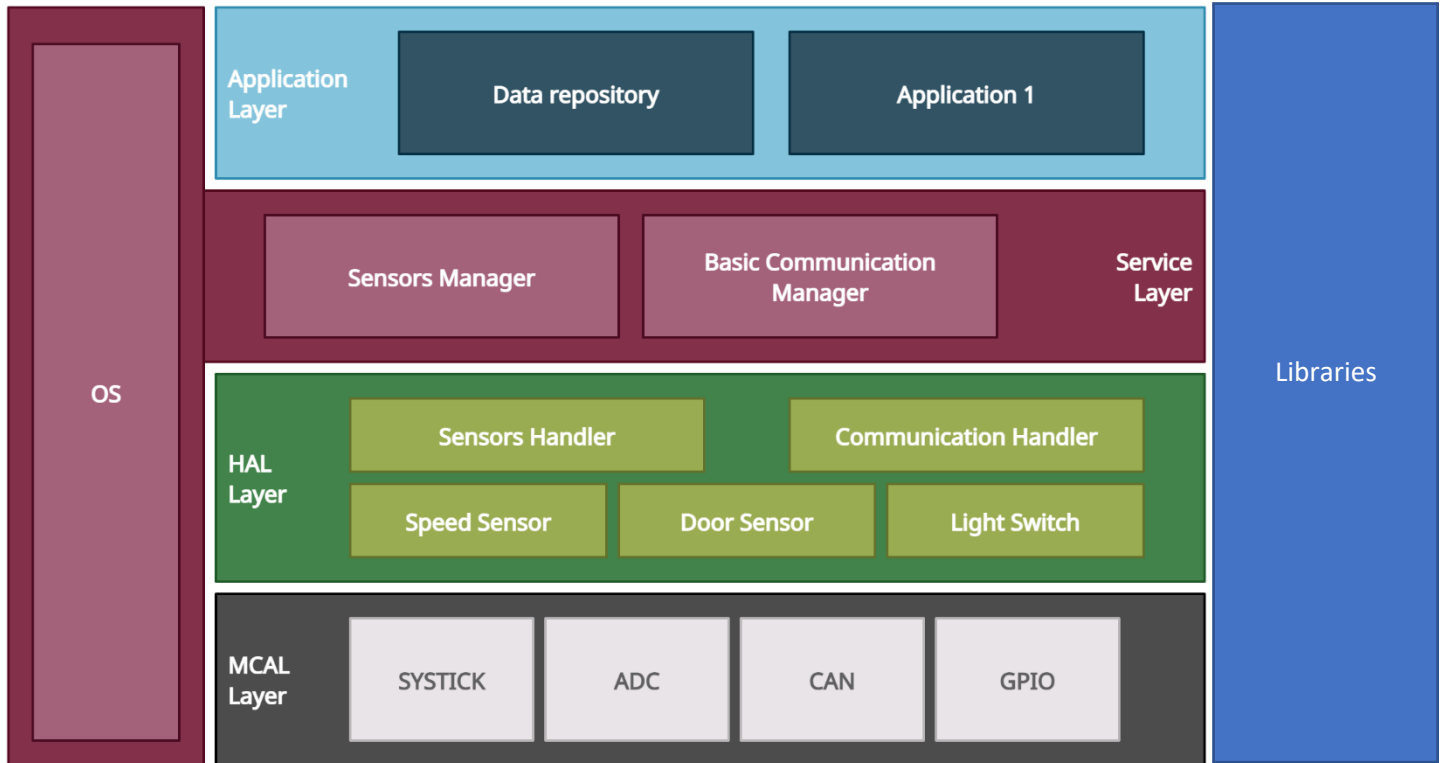
Static Design Analysis

Youssef Hussien Mahmoud

ysfhussien@gmail.com

ECU_1

1- Layered Architecutre



2- APIs

GPIO module:

API	void GPIO_init (struct * config_ptr);		
Description	Initialize the GPIO with the structure configurations		
Sync/Async	Synchronous	Reentrancy	Non-reentrant
Parameters	Pointer to structure	Return	void

API	void GPIO_write_Pin (uint32 Port_No, uint32 Pin_No, uint8 value);		
Description	Write the required GPIO port, Pin with the required value		
Sync/Async	Synchronous	Reentrancy	Non-reentrant
Parameters	Port number – Pin number – pin value	Return	None

API	uint8 GPIO_read_Pin (uint32 Port_No, uint32 Pin_No);		
Description	Read the required Gpio port , pin.		
Sync/Async	Synchronous	Reentrancy	Non-reentrant
Parameters	Port number – Pin number	Return	uint8

ADC Module:

API	void ADC_init (struct * Config_ptr);		
Description	Initialize the ADC with the structure configurations		
Sync/Async	Synchronous	Reentrancy	Non-reentrant
Parameters	Pointer to structure	Return	None

API	uint32 ADC_readChannel (uint8 Channel_id);		
Description	Read the required channel ID		
Sync/Async	Synchronous	Reentrancy	Non-reentrant
Parameters	channel ID	Return	uint32

CAN Module:

API	void CAN_init (struct * Config_ptr);		
Description	Initialize CAN bus with the structure configurations		
Sync/Async	Synchronous	Reentrancy	Non-reentrant
Parameters	Pointer to structure	Return	void

API	void CAN_transmit (uint8 CanPin_ID, uint64 Message);		
Description	Send a required message via required pin ID		
Sync/Async	Synchronous	Reentrancy	Non-reentrant
Parameters	Can Pin number – Message	Return	None

Speed Sensor Module:

API	void SpeedSensor_init (struct * Config_ptr);		
Description	Initialize the speed sensor pin via ADC		
Sync/Async	Synchronous	Reentrancy	Non-reentrant
Parameters	Pointer to structure	Return	void

API	Uint16 SpeedSensor_getSpeed (void);		
Description	Get the speed from the speed sensor via ADC		
Sync/Async	Synchronous	Reentrancy	Non-reentrant
Parameters	None	Return	Uint16

Door Sensor Module:

API	void DoorSensor_init (struct * Config_ptr);		
Description	Initialize the door sensor pin via GPIO		
Sync/Async	Synchronous	Reentrancy	Non-reentrant
Parameters	Pointer to structure	Return	void

API	uint8 DoorSensor_getStatus (void);		
Description	Read the door sensor status via GPIO		
Sync/Async	Synchronous	Reentrancy	Non-reentrant
Parameters	None	Return	uint8

Light Switch Module:

API	void LightSwitch_init (struct * Config_ptr);		
Description	Initialize the light switch module with the structure configurations		
Sync/Async	Synchronous	Reentrancy	Non-reentrant
Parameters	Pointer to structure	Return	void

API	uint8 LightSwitch_getStatus (void);		
Description	Read the light switch status		
Sync/Async	Synchronous	Reentrancy	Non-reentrant
Parameters	None	Return	Uint8

Sensor handler Module:

API	uint32 Sensor_handler (uint8 Sensor_ID);		
Description	chooses which sensor to read from hardware layer		
Sync/Async	Synchronous	Reentrancy	Non-reentrant
Parameters	Sensor ID	Return	Uint32

Communication handler module:

API	void BCM_handler (uint64 handler_Message, uint8 bus);		
Description	Choose which bus to send the required message		
Sync/Async	Synchronous	Reentrancy	Non-reentrant
Parameters	Message – bus	Return	None

Sensor manager Module:

API	uint32 Sensor_manager (uint8 sensor_Id);		
Description	Allow the application layer to choose the required sensor		
Sync/Async	Synchronous	Reentrancy	Non-reentrant
Parameters	Sensor ID	Return	Uint32

Basic Communication manager Module:

API	Void BCM_mananger (uint64 Manager_Message, uint8 bus);		
Description	Allow the application layer to choose the required bus		
Sync/Async	Synchronous	Reentrancy	Non-reentrant
Parameters	Message – bus	Return	None

Data Repository Module:

API	void Data_repository (uint64 data);		
Description	Save the required data		
Sync/Async	Synchronous	Reentrancy	Non-reentrant
Parameters	Data to be saved	Return	None

Application1 Module :

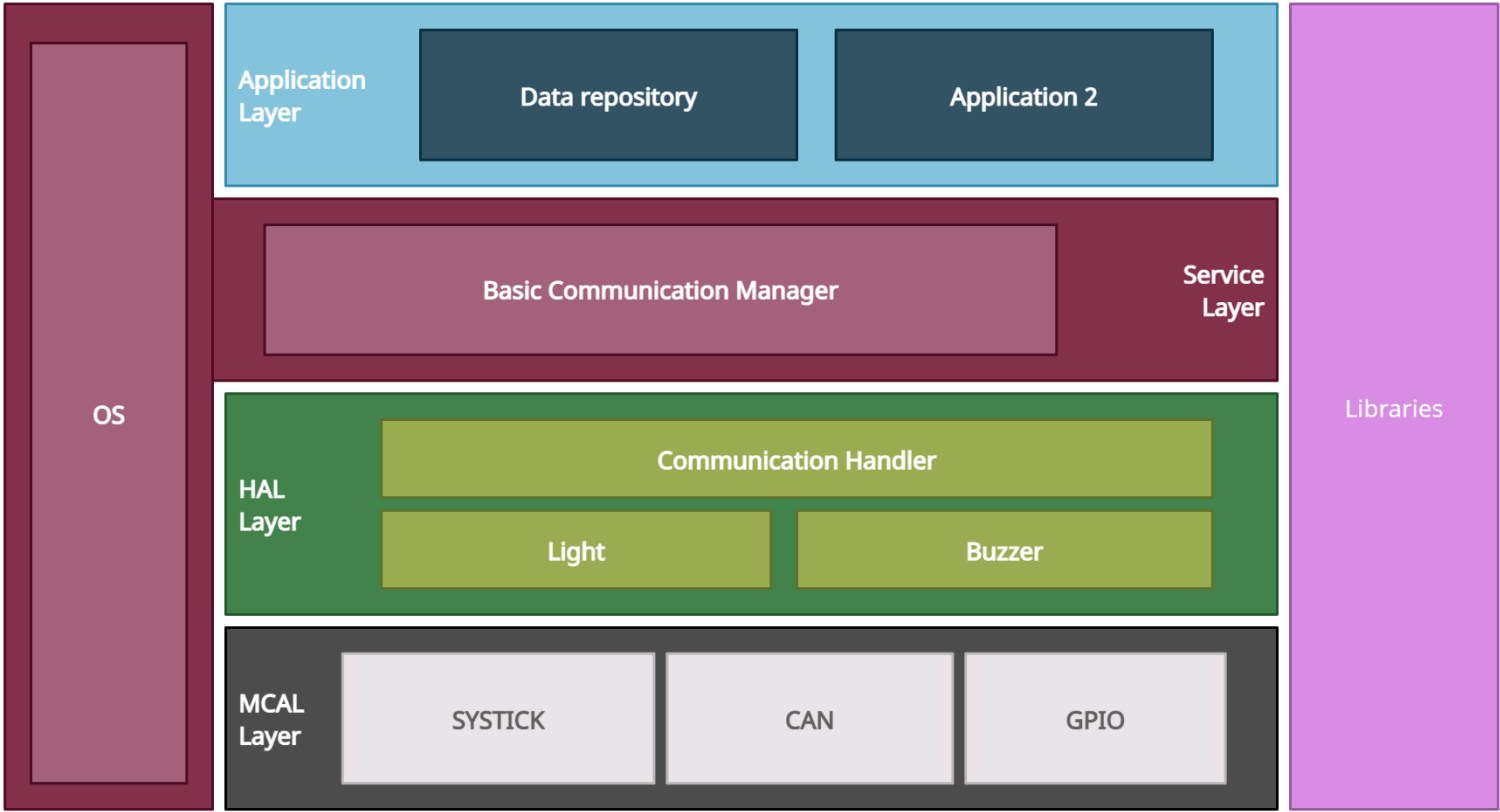
API	void SendDoorState (void);		
Description	Send the door sensor state to ECU2 via CAN bus		
Sync/Async	Synchronous	Reentrancy	Non-reentrant
Parameters	None	Return	None

API	void SendSpeed (void);		
Description	Send the speed sensor state to ECU2 via CAN bus		
Sync/Async	Synchronous	Reentrancy	Non-reentrant
Parameters	None	Return	None

API	void SendLightSwitchState (void);		
Description	Send the light switch state to ECU2 via CAN bus		
Sync/Async	Synchronous	Reentrancy	Non-reentrant
Parameters	None	Return	None

ECU_2

1- Layered Architecture



2- APIs

GPIO module:

API	void GPIO_init (struct * config_ptr);		
Description	Initialize the GPIO with the structure configurations		
Sync/Async	Synchronous	Reentrancy	Non-reentrant
Parameters	Pointer to structure	Return	void

API	void GPIO_write_Pin (uint32 Port_No, uint32 Pin_No, uint8 value);		
Description	Write the required GPIO port, Pin with the required value		
Sync/Async	Synchronous	Reentrancy	Non-reentrant
Parameters	Port number – Pin number – pin value	Return	None

API	uint8 GPIO_read_Pin (uint32 Port_No, uint32 Pin_No);		
Description	Read the required Gpio port , pin.		
Sync/Async	Synchronous	Reentrancy	Non-reentrant
Parameters	Port number – Pin number	Return	uint8

CAN Module :

API	void CAN_init (struct * Config_ptr);		
Description	Initialize CAN bus with the structure configurations		
Sync/Async	Synchronous	Reentrancy	Non-reentrant
Parameters	Pointer to structure	Return	void

API	uint64 CAN_Receive (uint8 CAN_Pin_Id);		
Description	Receive the CAN message from the required Pin ID		
Sync/Async	Synchronous	Reentrancy	Non-reentrant
Parameters	Can Pin number	Return	uint64

Buzzer Module:

API	void BUZZER_init (struct * Config_ptr);		
Description	Initialize the buzzer with the structure configurations via GPIO		
Sync/Async	Synchronous	Reentrancy	Non-reentrant
Parameters	Pointer to structure	Return	void

API	void BUZZER_on (void);		
Description	Set the buzzer on		
Sync/Async	Synchronous	Reentrancy	Non-reentrant
Parameters	None	Return	None

API	void BUZZER_off (void);		
Description	Set the buzzer off		
Sync/Async	Synchronous	Reentrancy	Non-reentrant
Parameters	None	Return	None

Communication handler module:

API	uint64 BCM_Handler (uint8 bus);		
Description	Choose which bus to read the message from MCAL layer		
Sync/Async	Synchronous	Reentrancy	Non-reentrant
Parameters	the bus	Return	Uint64

Basic Communication manager Module:

API	uint64 BCM_mananger (uint8 bus);		
Description	Allow the application layer to choose which bus to read the message from		
Sync/Async	Synchronous	Reentrancy	Non-reentrant
Parameters	bus	Return	Uint64

Light Module:

API	void Light_init (struct * Config_ptr);		
Description	Initialize the light via GPIO driver		
Sync/Async	Synchronous	Reentrancy	Non-reentrant
Parameters	Pointer to structure	Return	void

API	void light_ON (void);		
Description	Set the light on		
Sync/Async	Synchronous	Reentrancy	Non-reentrant
Parameters	None	Return	None

API	void light_OFF (void);		
Description	Set the light off		
Sync/Async	Synchronous	Reentrancy	Non-reentrant
Parameters	None	Return	None

Data repository Module:

API	void Data_repository (uint64 data);		
Description	Save the required data		
Sync/Async	Synchronous	Sync/Async	Synchronous
Parameters	Data to be saved	Parameters	Data to be saved

Application2 Module:

API	void Receive_Message (void);		
Description	Receive the message from ECU1 periodically to take actions		
Sync/Async	Synchronous	Reentrancy	Non-reentrant
Parameters	None	Return	None

Type definitions:

- typedef unsigned long uint32
- typedef unsigned short uint16
- typedef unsigned char uint8
- typedef unsigned long long uint64