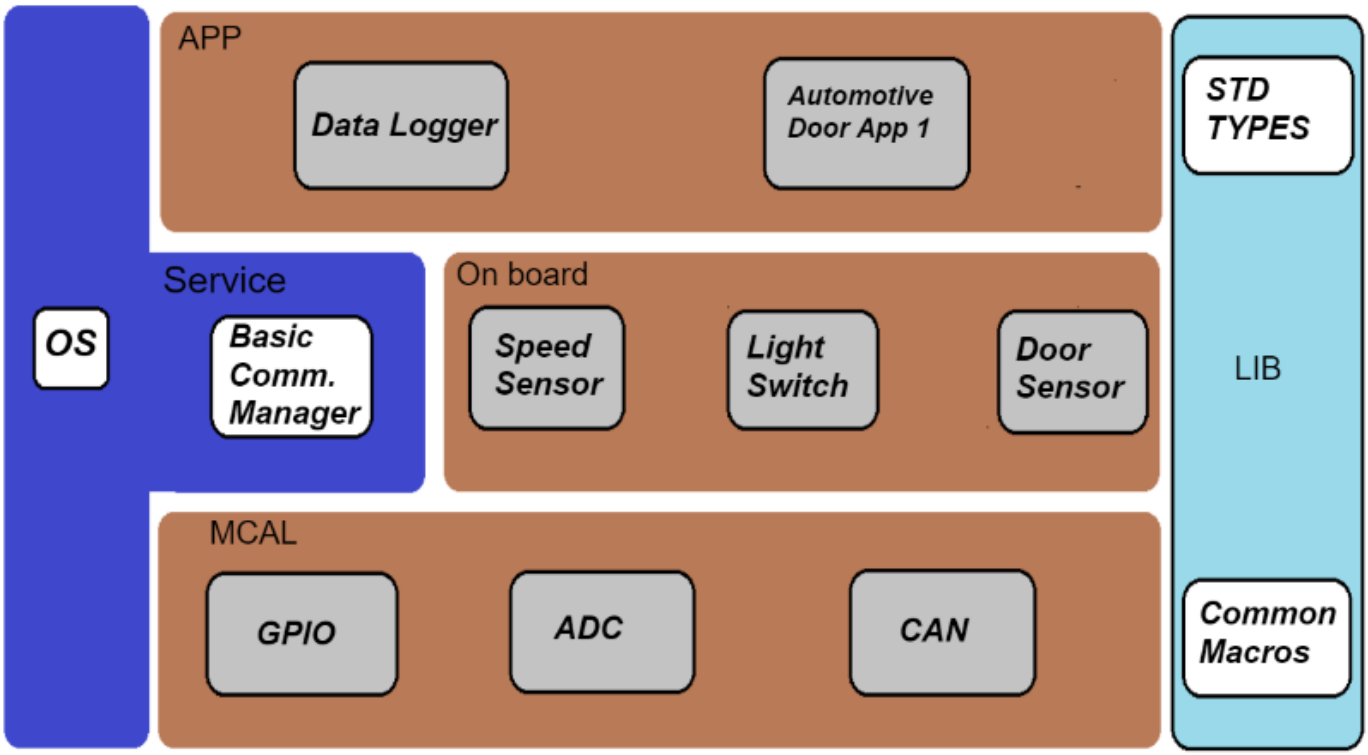
Automotive door control system design

Prepared by: Tarek Wael

[I]Static Design Analysis:

ECU 1

Layered Architecture



APIs

GPIO Module:

API	***void GPIO_Init(void)***		
Description	Initializes the GPIO.		
Sync/Async	Synchronous	***Reentrancy***	Non reentrant
Parameters	None	***Return***	None

API	***void GPIO_write(uint8 portID, uint8 pinID, uint8 Value)***		
Description	Outputs a value on a pin		
Sync/Async	Synchronous	***Reentrancy***	Non reentrant
Parameters	Port ID, Pin ID, Output value	***Return***	None

- PortID : uint8 variable, determines which port to interface with ranging from 0->4
- PinID: uint8 variable, determines which pin to interface with ranging from 0->16
- Value: uint8 variable, determines the value of the pin either 1 or 0

API	***uint8 GPIO_read(uint8 portID, uint8 pinID)***		
Description	Reads the value from a pin		
Sync/Async	Synchronous	***Reentrancy***	Non reentrant
Parameters	Port ID, Pin ID	***Return***	Current value of the pin

- PortID : uint8 variable, determines which port to interface with ranging from 0->4
- PinID: uint8 variable, determines which pin to interface with ranging from 0->16

ADC Module:

API	***void ADC_Init(void)***		
Description	Initializes the ADC.		
Sync/Async	Synchronous	***Reentrancy***	Non reentrant
Parameters	None	***Return***	None

API	***uint32 ADC_readChannel(uint8 channelID)***		
Description	Reads the analog value		
Sync/Async	Synchronous	***Reentrancy***	Non reentrant
Parameters	Channel ID	***Return***	Analog Value

- channelID: uint8 variable, determines which ADC channel to take conversion from, it ranges from 0->12

CAN Module:

API	***void CAN_Init(void)***		
Description	Initializes CAN.		
Sync/Async	Synchronous	***Reentrancy***	Non reentrant
Parameters	None	***Return***	None

API	***void CAN_voidStart(void)***		
Description	Starts CAN protocol		
Sync/Async	Synchronous	***Reentrancy***	Non reentrant
Parameters	None	***Return***	None

API	***void CAN_AddTxMsg(CAN_TxHeaderTypeDef *pTxHeader, uint8 Local_u8Data [])***		
Description	Send message through CAN		
Sync/Async	Synchronous	***Reentrancy***	Non reentrant
Parameters	PTxHeader , Data array	***Return***	None

- pTxHeader : a pointer to the head of the transmitted buffer.
- Local_u8Data []: An array of type uint8 that holds the data needed to be transmitted

Speed Sensor Module:

API	***Uint32 SpeedSensorGetSpeed(void)***		
Description	Returns vehicle speed		
Sync/Async	Synchronous	***Reentrancy***	Non reentrant
Parameters	None	***Return***	Speed value

Door Sensor Module:

API	***Uint8 DoorGetStatus(void)***		
Description	Returns door status		
Sync/Async	Synchronous	***Reentrancy***	Non reentrant
Parameters	None	***Return***	Door Status

Light Switch Module:

API	***Uint8 LSwitchGetStatus(void)***		
Description	Returns light switch status		
Sync/Async	Synchronous	***Reentrancy***	Non reentrant
Parameters	None	***Return***	Switch Status

Data Logger Module:

API	*void DataLoggerSave(uint32 data)*		
Description	Save the data sent to it.		
Sync/Async	Synchronous	***Reentrancy***	Non reentrant
Parameters	Data	***Return***	None

App Module:

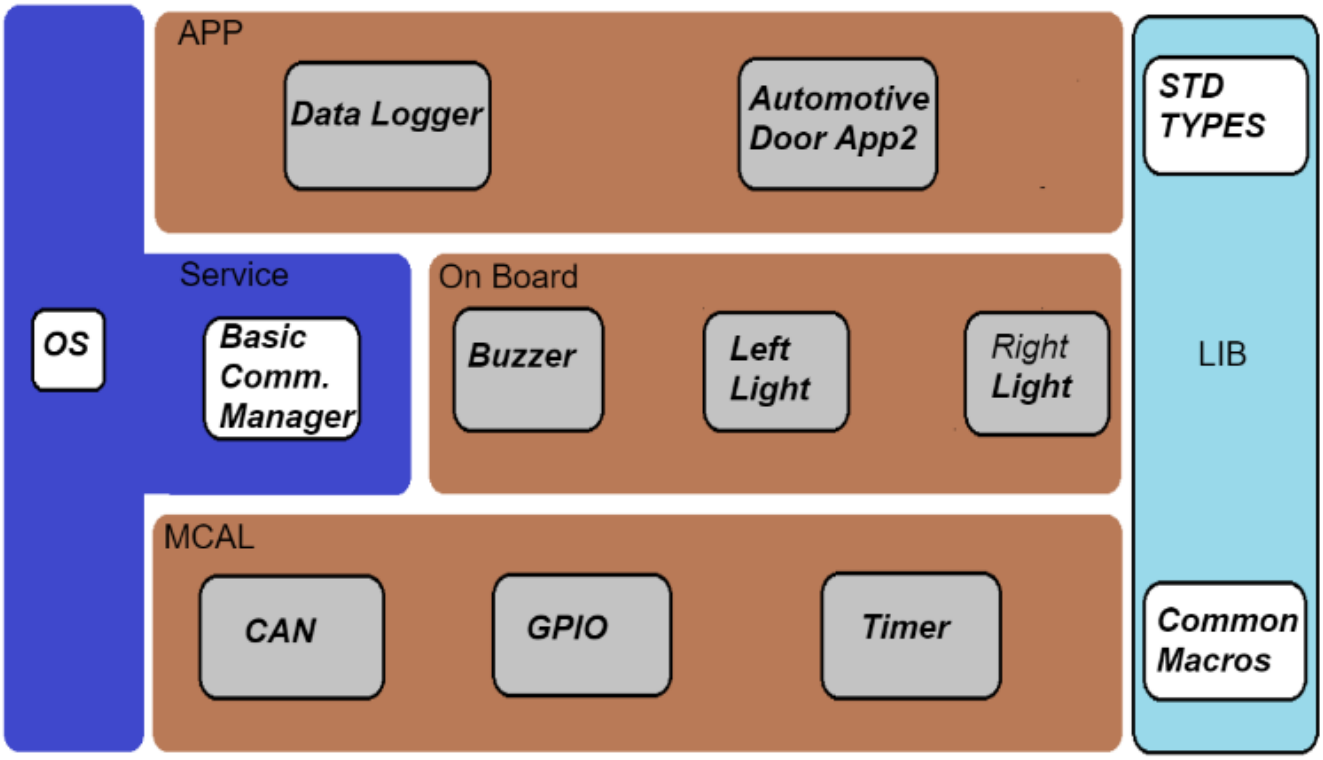
API	*void SendVehicleSpeed_Task(void)*		
Description	Sends vehicle's filtered speed read from speed sensor to ECU2 via CAN bus		
Sync/Async	Synchronous	***Reentrancy***	Non reentrant
Parameters	None	***Return***	None

API	*void SendDoorState_Task(void)*		
Description	Sends doors' state read from door sensors to ECU2 via CAN bus		
Sync/Async	Synchronous	***Reentrancy***	Non reentrant
Parameters	None	***Return***	None

API	*void SendSwitchState_Task(void)*		
Description	Sends light switch state to ECU2 via CAN bus		
Sync/Async	Synchronous	***Reentrancy***	Non reentrant
Parameters	None	***Return***	None

ECU 2

Layered Architecture



APIs

GPIO Module:

API	***void GPIO_Init(void)***		
Description	Initializes the GPIO.		
Sync/Async	Synchronous	***Reentrancy***	Non reentrant
Parameters	None	***Return***	None

API	***void GPIO_write(uint8 portID, uint8 pinID, uint8 Value)***		
Description	Outputs a value on a pin		
Sync/Async	Synchronous	***Reentrancy***	Non reentrant
Parameters	Port ID, Pin ID, Output value	***Return***	*None*

- PortID : uint8 variable, determines which port to interface with ranging from 0->4
- PinID: uint8 variable, determines which pin to interface with ranging from 0->16
- Value: uint8 variable, determines the value of the pin either 1 or 0

API	***uint8 GPIO_read(uint8 portID, uint8 pinID)***		
Description	Reads the value from a pin		
Sync/Async	Synchronous	***Reentrancy***	Non reentrant
Parameters	Port ID, Pin ID	***Return***	Current value of the pin

- PortID : uint8 variable, determines which port to interface with ranging from 0->4
- PinID: uint8 variable, determines which pin to interface with ranging from 0->16

CAN Module:

API	***void CAN_Init(void)***		
Description	Initializes CAN.		
Sync/Async	Synchronous	***Reentrancy***	Non reentrant
Parameters	None	***Return***	None

API	***void CAN_voidStart(void)***		
Description	Starts CAN protocol		
Sync/Async	Synchronous	***Reentrancy***	Non reentrant
Parameters	None	***Return***	None

API	void CAN_voidGetRxMsg(CAN_RxHeaderTypeDef *pRxHeader, uint8 Local_u8Data[])		
Description	Receive message through CAN		
Sync/Async	Synchronous	***Reentrancy***	Non reentrant
Parameters	Rx Buffer, PTxHeader , Data array	***Return***	None

- pRxHeader : a pointer to the head of the transmitted buffer.
- Local_u8Data []: An array of type uint8 that holds the data received.

Timer Module:

API	***void TIM_Init(uint8 TimNum)***		
Description	Initialize the Timer.		
Sync/Async	Synchronous	***Reentrancy***	Non reentrant
Parameters	Tim_x	***Return***	None

- TimNum : uint8 variable that determines which timer to enable, it ranges from 0->6

API	***void TIM_ SetInterval(uint32 time_ms, void (*Copy_ptr) (void))***		
Description	Reads the analog value.		
Sync/Async	Asynchronous	***Reentrancy***	Non reentrant
Parameters	Time in ms , pointer to callback function	***Return***	None

- Time_ms : uint32 variable which is considered the time to count in milliseconds.
- Copy_ptr: A pointer to function which is the notification function that performs an action when the ISR triggers.

API	***uint32 TIM_GetElapsedTime(void)***		
Description	Get elapsed time.		
Sync/Async	Synchronous	***Reentrancy***	Non reentrant
Parameters	None	***Return***	Elapsed time

Buzzer Module:

API	*void BuzzerON(void)*		
Description	Activates buzzer		
Sync/Async	Synchronous	***Reentrancy***	Non reentrant
Parameters	None	***Return***	None

API	*void BuzzerOFF(void)*		
Description	Deactivates buzzer		
Sync/Async	Synchronous	***Reentrancy***	Non reentrant
Parameters	None	***Return***	None

Lights Module:

API	*void LightsON(void)*		
Description	Switch on lights		
Sync/Async	Synchronous	***Reentrancy***	Non reentrant
Parameters	None	***Return***	None

API	*void LightsOFF(void)*		
Description	Switch off lights		
Sync/Async	Synchronous	***Reentrancy***	Non reentrant
Parameters	None	***Return***	None

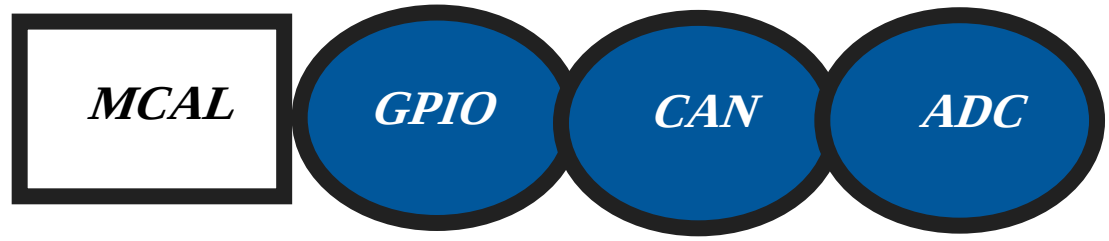
Data Logger Module:

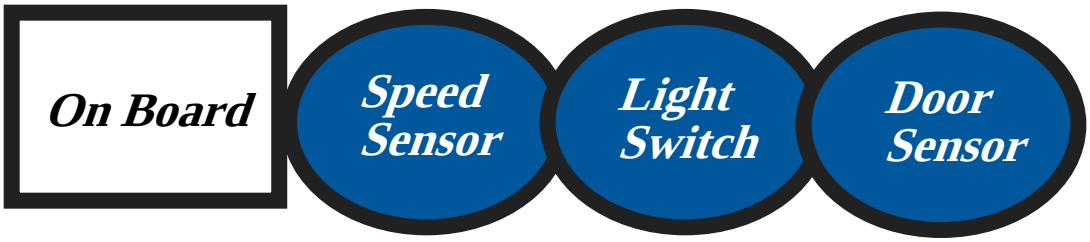
API	*void DataLoggerSave(uint32 data)*		
Description	Save the data sent to it.		
Sync/Async	Synchronous	***Reentrancy***	Non reentrant
Parameters	Data	***Return***	None

App Module:

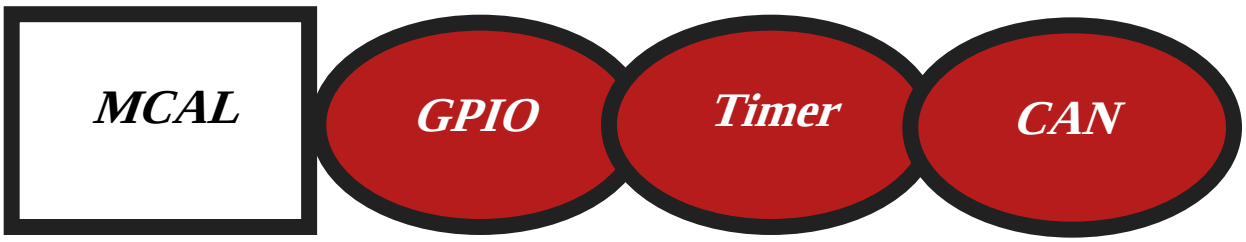
API	*void Action_Task(void)*		
Description	Make an action according to the 3 input sensors' readings		
Sync/Async	Synchronous	***Reentrancy***	Non reentrant
Parameters	None	***Return***	None

Folder Structure
ECU 1



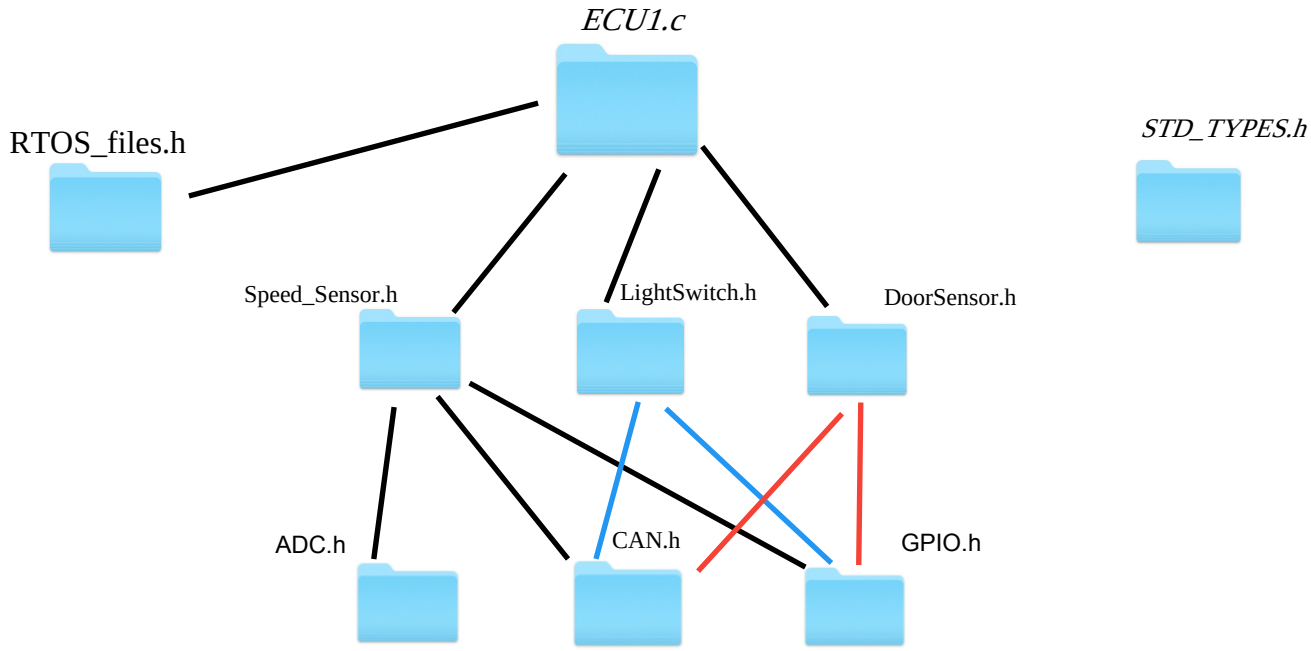


Folder Structure
ECU 2



File Inclusion

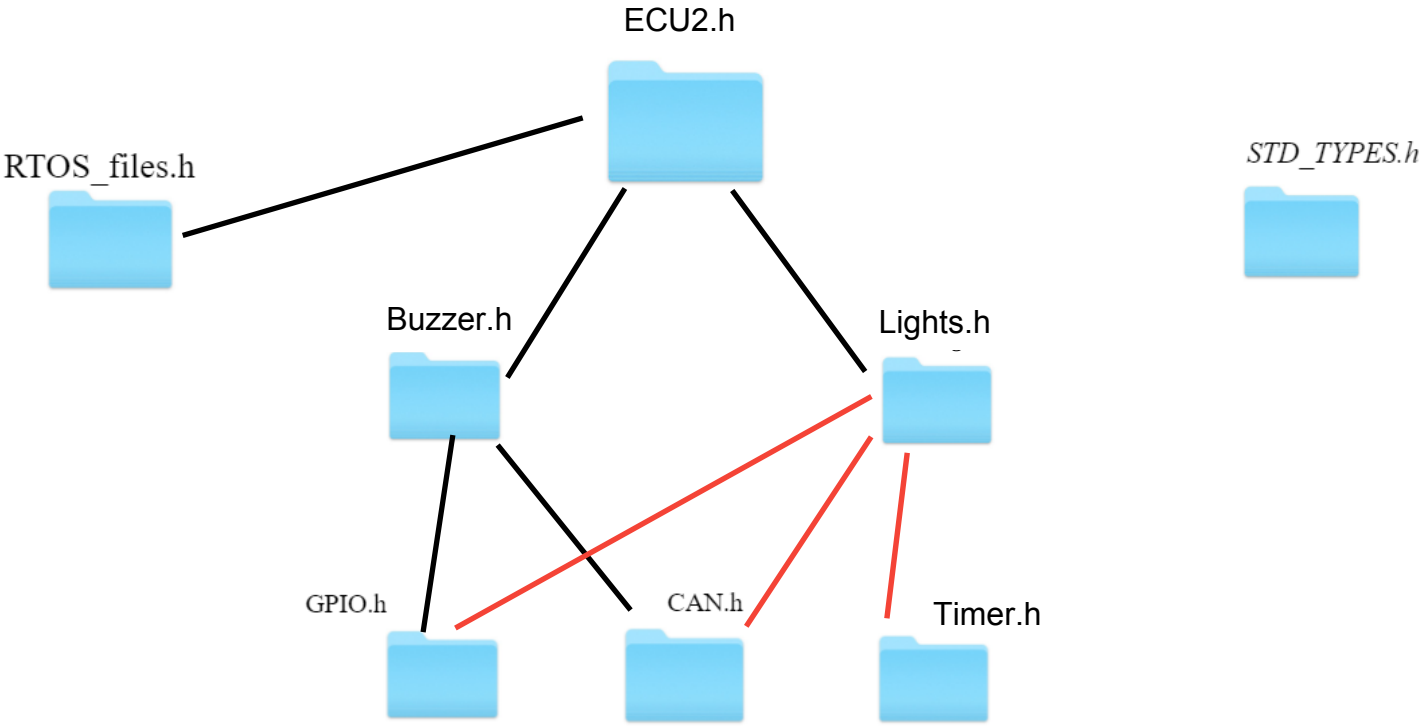
ECU1:



(STD_TYPES.h is included in all program files)

File Inclusion

ECU2:



(STD_TYPES.h is included in all program files)

Firmware File Structure:

- Periph.c
- Periph.h
- Periph_Private.h
- Periph_Config.h