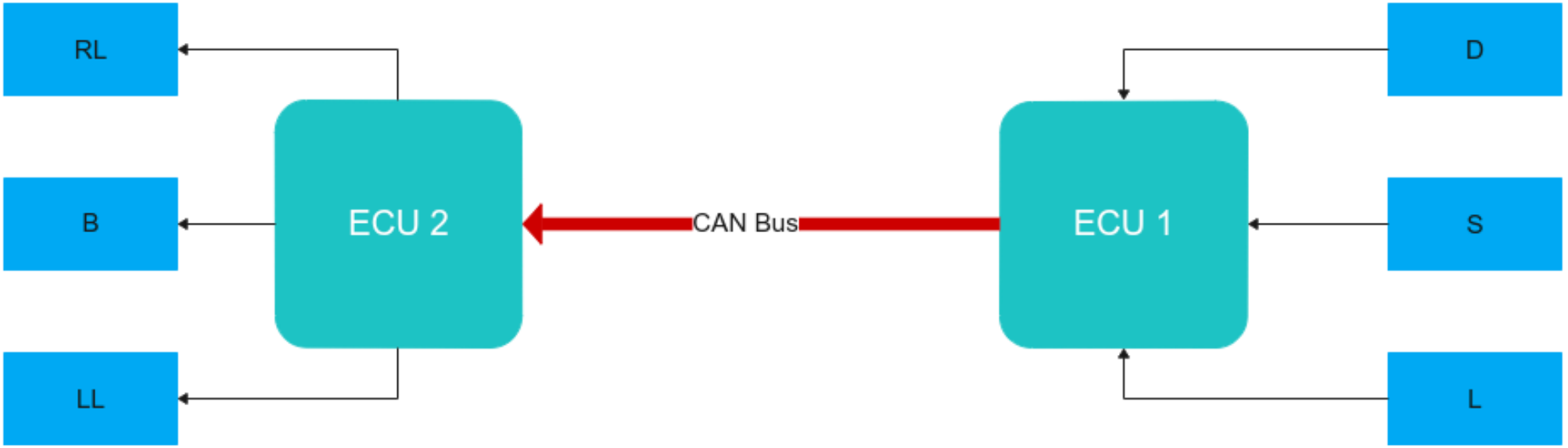
Project (3)

Automotive door control system design

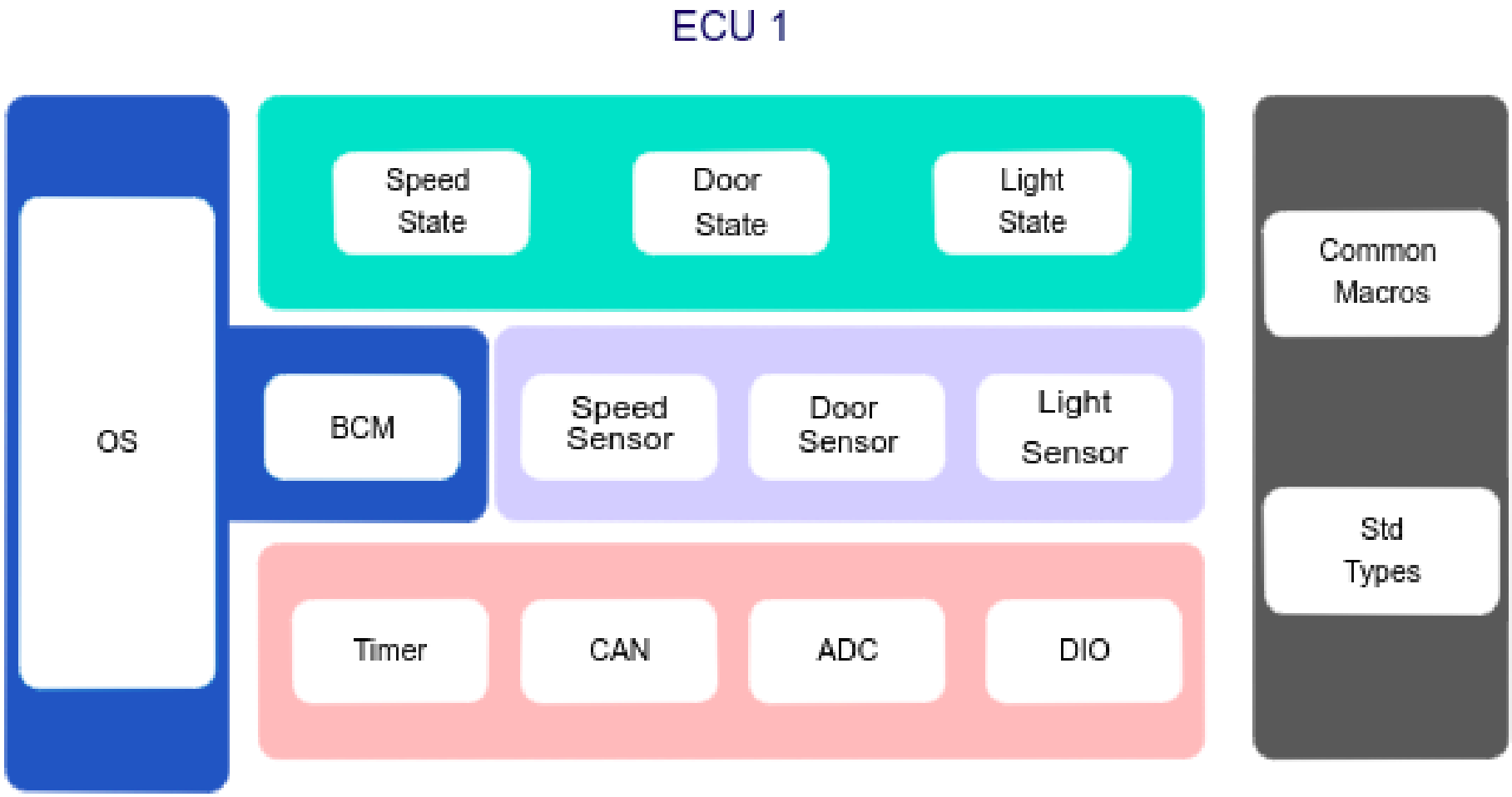
Name	Abdurrahman Mohame Elhefnawy
Email	Abdurrahman.elhefnwy@gmail.com

Block Diagram



ECU 1

layered architecture



DIO Module

This module is for reading or writing from/to DIO pins.

DIO Module's APIs:

DIO_init():

Syntax	Void DIO_init(void)
Description	This function used to initialize DIO module.
API Type	Init
Parameters (in)	None
Parameters (out)	None
Return value	None

DIO Module's APIs:

DIO_Write():

Syntax	Void DIO_Write(DIO_PinType,Dio_PinStateType)
Description	Function used to Write a state on a pin.
API Type	Setter
Parameters (in)	PinID , PinState
Parameters (out)	None
Return value	None

DIO Module's APIs:

DIO_read():

Syntax	Dio_PinStateType DIO_read(DIO_PinType)
Description	Function used to the state of a pin.
API Type	Getter
Parameters (in)	PinID
Parameters (out)	None
Return value	PinState

DIO Module's typedefs:

Type Name	Dio_PinStateType
Description	Type used to describe pin state
Type	enum
Range	HIGH-LOW

Type Name	Dio_PinType
Description	Type used to describe Pin Number
Type	enum
Range	PIN0 to PIN7

DIO Module's typedefs:

Type Name	Dio_PinDirectionType
Description	Type used to describe Pin Direction
Type	enum
Range	INPUT- OUTPUT

Timer Module

This module is for tick calculations and scheduling.

Timer Module's APIs:

Timer_init():

Syntax	void Gpt_Init ( )
Description	Used to initialize Timers Module.
API Type	Init
Parameters (in)	None
Parameters (out)	None
Return value	None

Timer Module's APIs:

Timer_ StartTimer():

Syntax	void Gpt_StartTimer (Gpt_ChannelType Channel, Gpt_Value)
Description	Starts selected timer cahnnel with defined target time.
API Type	-
Parameters (in)	Channel: timer channel , Value: target counter value
Parameters (out)	None
Return value	None

Timer Module's APIs:

Timer_ Gpt_GetTimeElapsed ():

Syntax	Gpt_ValueType Gpt_GetTimeElapsed (Gpt_ChannelType Channel)
Description	Gets the Elapsed time of a specific channel
API Type	Getter
Parameters (in)	Channel: timer channel
Parameters (out)	None
Return value	Time Elapsed

Timer Module's typedefs:

Type Name	GPT_ValueType
Description	Type for holding timer ticks
Type	Uint8

Type Name	Gpt_ChannelType
Description	Type used to describe Channel Number
Type	enum

ADC Module

This module is for Converting Analog readings into digital.

ADC Module's APIs:

ADC_Init()

Syntax	void ADC_Init(void)
Description	Used to initialize ADC Module.
API Type	Init
Parameters (in)	None
Parameters (out)	None
Return value	None

:

ADC Module's APIs:

ADC_read():

Syntax	void ADC_ read (ADC_ChannelType)
Description	Reads an analog voltage from a specific channel
API Type	-
Parameters (in)	Channel: timer channel , Value: target counter value
Parameters (out)	None
Return value	None

ADC Module's typedefs:

Type Name	ADC_ChannelType
Description	Type used to describe Channel Number
Type	enum

CAN Module

This module is for Communication between the 2 ECUs.

CAN Module's APIs:

CAN_init():

Syntax	void CAN_Init ( )
Description	Used to initialize CAN Module.
API Type	Init
Parameters (in)	None
Parameters (out)	None
Return value	None

CAN_Module's APIs:

CAN_ SendData():

Syntax	void CAN_ SendByte (Uint8 Data)
Description	Sends one byte.
API Type	-
Parameters (in)	Data: One Byte
Parameters (out)	None
Return value	None

CAN_Module's APIs:

CAN_ SendString():

Syntax	void CAN_ SendString (Uint8* Str)
Description	Sends stream of byte.
API Type	-
Parameters (in)	Str: Stream of Bytes
Parameters (out)	None
Return value	None

CAN_Module's APIs:

CAN_ RecicveByte():

Syntax	Uint8 CAN_ RecieveByte (void)
Description	Receives one byte.
API Type	-
Parameters (in)	None
Parameters (out)	None
Return value	Received Byte

Light_Sensor Module

This module is for Interfacing with the Light sensor.

Light_Sensor Module's APIs:

Light_Sensor _init():

Syntax	Light_Sensor _init():
Description	Used to initialize Light_Sensor Module.
API Type	Init
Parameters (in)	None
Parameters (out)	None
Return value	None

Light_Sensor Module's APIs:

Light_Sensor _ReadValue():

Syntax	Uint_8 Light_Sensor _ReadValue(void):
Description	Used to read light sensor data.
API Type	Getter
Parameters (in)	None
Parameters (out)	None
Return value	Sensor Value

Door_Sensor Module

This module is for Interfacing with the Door sensor.

Door_Sensor Module's APIs:

Light_Sensor _init():

Syntax	Door_Sensor _init():
Description	Used to initialize Door_Sensor Module.
API Type	Init
Parameters (in)	None
Parameters (out)	None
Return value	None

Door_Sensor Module's APIs:

Door_Sensor _ReadValue():

Syntax	Uint_8 Door_Sensor _ReadValue(void):
Description	Used to read Door sensor data.
API Type	Getter
Parameters (in)	None
Parameters (out)	None
Return value	Sensor Value

Speed_Sensor Module

This module is for Interfacing with the Speed sensor.

Speed _Sensor Module's APIs:

Light_Sensor _init():

Syntax	Speed_Sensor _init():
Description	Used to initialize Speed_Sensor Module.
API Type	Init
Parameters (in)	None
Parameters (out)	None
Return value	None

Speed_Sensor Module's APIs:

Speed_Sensor _ReadValue():

Syntax	Uint_8 Speed_Sensor _ReadValue(void):
Description	Used to read Speed sensor data.
API Type	Getter
Parameters (in)	None
Parameters (out)	None
Return value	Sensor Value

BCM Module

This module is for managing Communication process.

BCM Module's APIs:

BCM _init():

Syntax	BCM_init():
Description	Used to initialize BCM Module.
API Type	Init
Parameters (in)	None
Parameters (out)	None
Return value	None

BCM Module's APIs:

BCM_SendString():

Syntax	Void BCM_SendString(BusType bus,uint8* Str):
Description	Used to send string.
API Type	-
Parameters (in)	Bus Type and string to be sent
Parameters (out)	None
Return value	None

BCM Module's APIs:

BCM_Recieve():

Syntax	Uint8 BCM_ReceiveBusType bus):
Description	Used to receive byte.
API Type	Getter
Parameters (in)	Bus Type
Parameters (out)	None
Return value	Received byte.

BCM Module's typedefs:

Type Name	BCM_BusType
Description	Type used to describe Bus Id
Type	enum

Light_State Module

This module is for reading from light module and send its state.

Light_State Module's APIs:

Light_State_monitor ():

Syntax	Void Light_State_monitor():
Description	Used for for reading from light module and send its state.
API Type	-
Parameters (in)	None
Parameters (out)	None
Return value	None

Door_State Module

This module is for reading from Door modue and send its state.

Door_State Module's APIs:

Door_State_monitor ():

Syntax	Void Door_State_monitor():
Description	Used for for reading from Door module and send its state.
API Type	-
Parameters (in)	None
Parameters (out)	None
Return value	None

Speed_State Module

This module is for reading from Speed module and send its state.

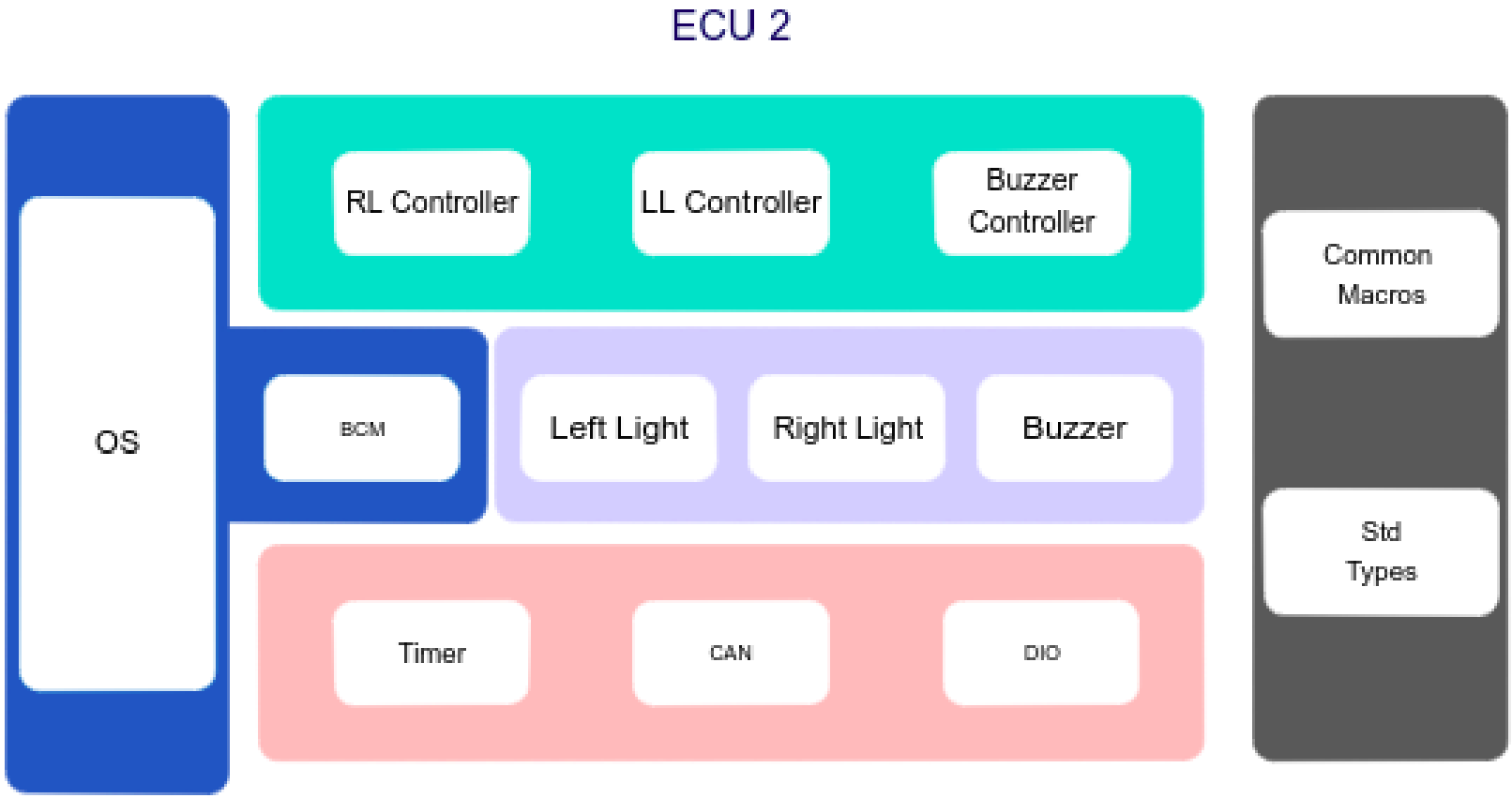
Speed_State Module's APIs:

Speed_State_monitor ():

Syntax	Void Speed_State_monitor():
Description	Used for for reading from Speed modue and send its state.
API Type	-
Parameters (in)	None
Parameters (out)	None
Return value	None

ECU 2

layered architecture



DIO Module

This module is for reading or writing from/to DIO pins.

DIO Module's APIs:

DIO_init():

Syntax	Void DIO_init(void)
Description	This function used to initialize DIO module.
API Type	Init
Parameters (in)	None
Parameters (out)	None
Return value	None

DIO Module's APIs:

DIO_Write():

Syntax	Void DIO_Write(DIO_PinType,Dio_PinStateType)
Description	Function used to Write a state on a pin.
API Type	Setter
Parameters (in)	PinID , PinState
Parameters (out)	None
Return value	None

DIO Module's APIs:

DIO_read():

Syntax	Dio_PinStateType DIO_read(DIO_PinType)
Description	Function used to the state of a pin.
API Type	Getter
Parameters (in)	PinID
Parameters (out)	None
Return value	PinState

DIO Module's typedefs:

Type Name	Dio_PinStateType
Description	Type used to describe pin state
Type	enum
Range	HIGH-LOW

Type Name	Dio_PinType
Description	Type used to describe Pin Number
Type	enum
Range	PIN0 to PIN7

DIO Module's typedefs:

Type Name	Dio_PinDirectionType
Description	Type used to describe Pin Direction
Type	enum
Range	INPUT- OUTPUT

Timer Module

This module is for tick calculations and scheduling.

Timer Module's APIs:

Timer_init():

Syntax	void Gpt_Init ( )
Description	Used to initialize Timers Module.
API Type	Init
Parameters (in)	None
Parameters (out)	None
Return value	None

Timer Module's APIs:

Timer_ StartTimer():

Syntax	void Gpt_StartTimer (Gpt_ChannelType Channel, Gpt_Value)
Description	Starts selected timer cahnnel with defined target time.
API Type	-
Parameters (in)	Channel: timer channel , Value: target counter value
Parameters (out)	None
Return value	None

Timer Module's APIs:

Timer_ Gpt_GetTimeElapsed ():

Syntax	Gpt_ValueType Gpt_GetTimeElapsed (Gpt_ChannelType Channel)
Description	Gets the Elapsed time of a specific channel
API Type	Getter
Parameters (in)	Channel: timer channel
Parameters (out)	None
Return value	Time Elapsed

Timer Module's typedefs:

Type Name	GPT_ValueType
Description	Type for holding timer ticks
Type	Uint8

Type Name	Gpt_ChannelType
Description	Type used to describe Channel Number
Type	enum

CAN Module

This module is for Communication between the 2 ECUs.

CAN Module's APIs:

CAN_init():

Syntax	void CAN_Init ( )
Description	Used to initialize CAN Module.
API Type	Init
Parameters (in)	None
Parameters (out)	None
Return value	None

CAN_Module's APIs:

CAN_ SendData():

Syntax	void CAN_ SendByte (Uint8 Data)
Description	Sends one byte.
API Type	-
Parameters (in)	Data: One Byte
Parameters (out)	None
Return value	None

CAN_Module's APIs:

CAN_ SendString():

Syntax	void CAN_ SendString (Uint8* Str)
Description	Sends stream of byte.
API Type	-
Parameters (in)	Str: Stream of Bytes
Parameters (out)	None
Return value	None

CAN_Module's APIs:

CAN_ RecicveByte():

Syntax	Uint8 CAN_ RecieveByte (void)
Description	Receives one byte.
API Type	-
Parameters (in)	None
Parameters (out)	None
Return value	Received Byte

Buzzer Module

This module is for Interfacing with the Buzzer.

Buzzer Module's APIs:

Buzzer_init():

Syntax	Void Buzzer_init(void):
Description	Used to initialize Buzzer Module.
API Type	Init
Parameters (in)	None
Parameters (out)	None
Return value	None

Buzzer Module's APIs:

Buzzer_On():

Syntax	Void Buzzer_On(void):
Description	Used to turn the buzzer on .
API Type	-
Parameters (in)	None
Parameters (out)	None
Return value	None

Buzzer Module's APIs:

Buzzer_Off():

Syntax	Void Buzzer_Off(void):
Description	Used to turn the buzzer Off.
API Type	-
Parameters (in)	None
Parameters (out)	None
Return value	None

Right_Light Module

This module is for Interfacing with the Right light.

Right_Light Module's APIs:

Right_Light_init():

Syntax	Void Right_Light _init(void):
Description	Used to initialize Right_Light Module.
API Type	Init
Parameters (in)	None
Parameters (out)	None
Return value	None

Right_Light Module's APIs:

Right_Light _On():

Syntax	Void Right_Light _On(void):
Description	Used to turn the Right_Light on .
API Type	-
Parameters (in)	None
Parameters (out)	None
Return value	None

Right_Light Module's APIs:

Right_Light _Off():

Syntax	Void Right_Light _Off(void):
Description	Used to turn the Right_Light Off.
API Type	-
Parameters (in)	None
Parameters (out)	None
Return value	None

Left_Light Module

This module is for Interfacing with the Left light.

Left _Light Module's APIs:

Left _Light_init():

Syntax	Void Left _Light _init(void):
Description	Used to initialize Left _Light Module.
API Type	Init
Parameters (in)	None
Parameters (out)	None
Return value	None

Left _Light Module's APIs:

Left _Light _On():

Syntax	Void Left _Light _On(void):
Description	Used to turn the Left _Light on .
API Type	-
Parameters (in)	None
Parameters (out)	None
Return value	None

Left _Light Module's APIs:

Left _Light _Off():

Syntax	Void Left _Light _Off(void):
Description	Used to turn the Left _Light Off.
API Type	-
Parameters (in)	None
Parameters (out)	None
Return value	None

BCM Module

This module is for managing Communication process.

BCM Module's APIs:

BCM _init():

Syntax	BCM_init():
Description	Used to initialize BCM Module.
API Type	Init
Parameters (in)	None
Parameters (out)	None
Return value	None

BCM Module's APIs:

BCM_SendString():

Syntax	Void BCM_SendString(BusType bus,uint8* Str):
Description	Used to send string.
API Type	-
Parameters (in)	Bus Type and string to be sent
Parameters (out)	None
Return value	None

BCM Module's APIs:

BCM_Recieve():

Syntax	Uint8 BCM_ReceiveBusType bus):
Description	Used to receive byte.
API Type	Getter
Parameters (in)	Bus Type
Parameters (out)	None
Return value	Received byte.

BCM Module's typedefs:

Type Name	BCM_BusType
Description	Type used to describe Bus Id
Type	enum

Buzzer_Controller Module

This module is for Controlling Buzzer module.

Buzzer_Controller Module's APIs:

Buzzer_Control():

Syntax	Void Buzzer_Controll(Buzzer_stateType state):
Description	Used for for Controlling Buzzer module.
API Type	-
Parameters (in)	State ON or OFF
Parameters (out)	None
Return value	None

Buzzer Controller Module's typedefs:

Type Name	Buzzer_stateType
Description	Type used to describe Buzzer State
Type	enum (HIGH-LOW)

LL_Controller Module

This module is for Controlling LLmodule.

LL_Controller Module's APIs:

LL_Control():

Syntax	Void LL_Controll(LL_stateType state):
Description	Used for for Controlling LL module.
API Type	-
Parameters (in)	State ON or OFF
Parameters (out)	None
Return value	None

LL Controller Module's typedefs:

Type Name	LL_stateType
Description	Type used to describe LL State
Type	enum (HIGH-LOW)

RL_Controller Module

This module is for Controlling RL module.

RL_Controller Module's APIs:

RL_Control():

Syntax	Void RL_Controll(RL_stateType state):
Description	Used for for Controlling RL module.
API Type	-
Parameters (in)	State ON or OFF
Parameters (out)	None
Return value	None

RL Controller Module's typedefs:

Type Name	RL_stateType
Description	Type used to describe RL State
Type	enum (HIGH-LOW)