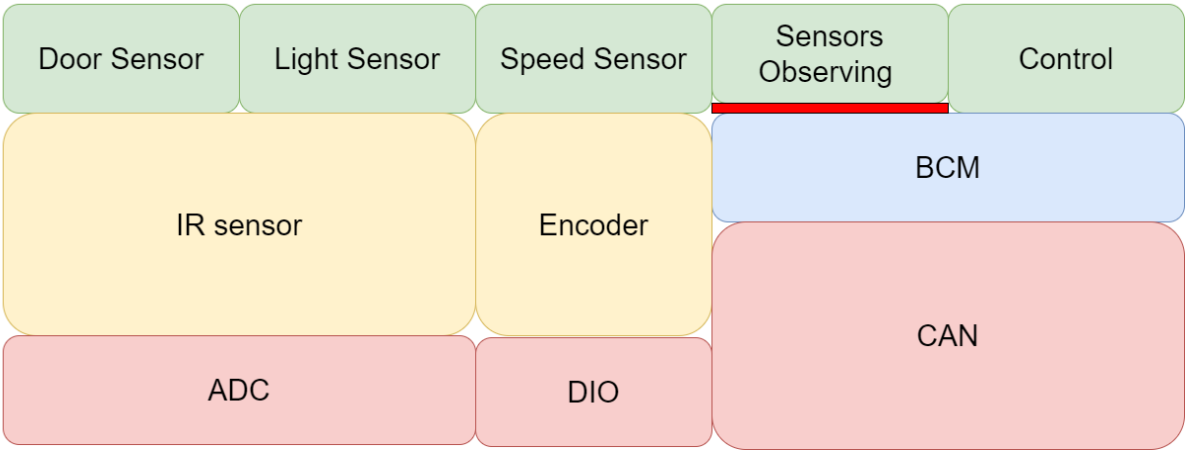
Full static and dynamic design for two ECUs system

By/ Ahmed Maged

First – Sensors ECU

1 – Static design:

A- Layered architecture



B- System Parameters

Name	gSysPar		
Type	struct		
elements	DoorState	DoorState_t	Contain global current Door state
	LightState	LightState _t	Contain global current light state
	SpeedState	MovingState_t	Contain global current moving state
Description	This a global structure containing the current states of the sensors		

C- APIs documentation.

1 – Door Sensor Stack:

Function Name	Error_t Door_init(void);		
Arguments	inputs	N/A	
		Description:	
	outputs	N/A	
		description:	
	input/output	N/A	
		description:	
Return	E_OK	0	
	E_NOK	1	
Description	call this api to initialize the Door Sensor module as in Door_config.h		

Function Name	Error_t Door_Update(void);		
Arguments	inputs	N/A	
		Description:	
	outputs	N/A	
		description:	
	input/output	N/A	
		description:	
Return	E_OK	0	
	E_NOK	1	
Description	call this api periodically as it is the main function of the module and contain the module state machine		

Function Name	Error_t Door_GetCurrentState(DoorState_t* pReturnDoorState);		
Arguments	inputs	N/A	
		Description:	
	outputs	pReturnDoorState	enumeration
		description:	
	input/output	N/A	
		description:	
Return	E_OK	0	
	E_NOK	1	
Description	call this api to get the current state of the module		

Name	pReturnDoorState		
Type	enumeration		
Range	CLOSED	0	Indicate its currently closed
	OPENED	1	Indicate its currently opened
Description	These values are to determine the current door state		

Function Name	Error_t IR_init(void);		
Arguments	inputs	N/A	
		Description:	
	outputs	N/A	
		description:	
	input/output	N/A	
		description:	
Return	E_OK	0	
	E_NOK	1	
Description	call this api to initialize the Door Sensor module as in IR_config.h		

Function Name	Error_t IR_Update(Ch_ID_t ChannelID_cpy);		
Arguments	inputs	ChannelID_cpy	enumeration
		Description: the channel id containing the sensor	
	outputs	N/A	
		description:	
	input/output	N/A	
		description:	
Return	E_OK	0	
	E_NOK	1	
Description	call this api periodically as it is the main function of the module and contain the module state machine		

Name	ChannelID_cpy		
Type	enumeration		
Range	CHANNEL0	0	To read sensor on Channel 0
	CHANNEL1	1	To read sensor on Channel 1
Description	These values are the channel IDs contains the sensors		

Function Name	IR_GetCurrentValue(Ch_ID_t ChannelID_cpy, u16* pReturnValue);		
Arguments	inputs	ChannelID_cpy	enumeration
		Description: the channel id containing the sensor	
	outputs	pReturnValue	u16
		description: the ADC conversion result	
	input/output	N/A	
		description:	
Return	E_OK	0	
	E_NOK	1	
Description	call this api to get the current state of the module		

Name	ChannelID_cpy		
Type	enumeration		
Range	CHANNEL0	0	To read sensor on Channel 0
	CHANNEL1	1	To read sensor on Channel 1
Description	These values are the channel IDs contains the sensors		

Name	pReturnValue		
Type	U16		
Range	0	0	The least result
	1023	1	The biggest result (10 Bit resolution ADC)
Description	These values are the channel IDs contains the sensors		

2 – Light Sensor Stack:

Function Name	Error_t Light_init(void);		
Arguments	inputs	N/A	
		Description:	
	outputs	N/A	
		description:	
	input/output	N/A	
		description:	
Return	E_OK	0	
	E_NOK	1	
Description	call this api to initialize the Light Sensor module as in Light_config.h		

Function Name	Error_t Light_Update(void);		
Arguments	inputs	N/A	
		Description:	
	outputs	N/A	
		description:	
	input/output	N/A	
		description:	
Return	E_OK	0	
	E_NOK	1	
Description	call this api periodically as it is the main function of the module and contain the module state machine		

Function Name	Error_t Light_GetCurrentState(LightState_t* pReturnLightState);		
Arguments	inputs	N/A	
		Description:	
	outputs	pReturnLightState	enumeration
		description:	
	input/output	N/A	
		description:	
Return	E_OK	0	
	E_NOK	1	
Description	call this api to get the current state of the module		

Name	pReturnLightState		
Type	enumeration		
Range	OFF	0	Indicate its currently Off
	ON	1	Indicate its currently On
Description	These values are to determine the current light state		

*IR previously mentioned in Door Stack.

3 – Speed Sensor Stack:

Function Name	Error_t Speed_init(void);		
Arguments	inputs	N/A	
		Description:	
	outputs	N/A	
		description:	
	input/output	N/A	
		description:	
Return	E_OK	0	
	E_NOK	1	
Description	call this api to initialize the Door Sensor module as in Door_config.h		

Function Name	Error_t Speed_Update(void);		
Arguments	inputs	N/A	
		Description:	
	outputs	N/A	
		description:	
	input/output	N/A	
		description:	
Return	E_OK	0	
	E_NOK	1	
Description	call this api periodically as it is the main function of the module and contain the module state machine		

Function Name	Error_t Speed_GetCurrentState(MovingState_t* pReturnMovingState);		
Arguments	inputs	N/A	
		Description:	
	outputs	pReturnMovingState	enumeration
		description:	
	input/output	N/A	
		description:	
Return	E_OK	0	
	E_NOK	1	
Description	call this api to get the current state of the module		

Name	pReturnMovingState		
Type	enumeration		
Range	Stopped	0	Indicate its currently stopped
	Moving	1	Indicate its currently moving
Description	These values are to determine the current Moving state		

Function Name	Error_t Encoder_init(void);		
Arguments	inputs	N/A	
		Description:	
	outputs	N/A	
		description:	
	input/output	N/A	
		description:	
Return	E_OK	0	
	E_NOK	1	
Description	call this api to initialize the Door Sensor module as in Encoder_config.h		

Function Name	Error_t Encoder_Update(Ch_ID_t ChannelID_cpy);		
Arguments	inputs	ChannelID_cpy	enumeration
		Description:	
	outputs	N/A	
		description:	
	input/output	N/A	
		description:	
Return	E_OK	0	
	E_NOK	1	
Description	call this api periodically as it is the main function of the module and contain the module state machine		

Name	ChannelID_cpy		
Type	enumeration		
Range	CHANNEL0	0	The starting Channels
	CHANNEL7	1	The Max num of channels
Description	These values are the channel IDs (Pins)		

Function Name	Encoder_GetCurrentRPS(Ch_ID_t ChannelID_cpy, u16*pReturnRPS)		
Arguments	inputs	ChannelID_cpy	enumeration
		Description:	
	outputs	pReturnRPS	U16
		description:	
	input/output	N/A	
		description:	
Return	E_OK	0	
	E_NOK	1	
Description	call this api to get the current state of the module		

Name	ChannelID_cpy		
Type	enumeration		
Range	CHANNEL0	0	The starting Channels
	CHANNEL7	1	The Max num of channels
Description	These values are the channel IDs (Pins)		

Name	pReturnRPS	
Type	U16	
Range	0	RPS is zero
	65,536	Max RPS num
Description	These values the Rotate Per Second number	

4 – Sensors Observer:

Function Name	Error_t Observer_init(void);		
Arguments	inputs	N/A	
		Description:	
	outputs	N/A	
		description:	
	input/output	N/A	
		description:	
Return	E_OK	0	
	E_NOK	1	
Description	call this api to initialize the Door Sensor module as in SenObserver_config.h		

Function Name	Error_t Observer_Notify(void);		
Arguments	inputs	N/A	
		Description:	
	outputs	N/A	
		description:	
	input/output	N/A	
		description:	
Return	E_OK	0	
	E_NOK	1	
Description	call this api periodically as it is the main function of the module and contain the module state machine		

Function Name	Observer_Subscribe(Error_t (*pGetter)(void* pReturnData), Error_t (*pNotification)(void))		
Arguments	inputs	pGetter	*Function
		Description: the sensor data getter.	
		pNotification	*Function
		Description: the called function to notify an update	
	outputs	N/A	
		description:	
	input/output	N/A	
		description:	
Return	E_OK	0	
	E_NOK	1	
Description	Call this api to set a new subscription, with the data getter and the function to call to notify update.		

Name	pGetter	
Type	*function	
Range	Door_GetCurrentState	To get door state
	Light_GetCurrentState	To get light state
	Speed_GetCurrentState	To get speed state
Description	These data getters	

Name	pNotification	
Type	*function	
Range	Control_DoorUpdate_Clbk	To call when Door data updated
	Control_LightUpdate_Clbk	To call when light data updated
	Control_SpeedUpdate_Clbk	To call when speed data updated
Description	These notifications APIs when data associated updated	

Function Name	Observer_UnSubscribe(Error_t (*pGetter)(void* pReturnData), Error_t (*pNotification)(void))		
Arguments	inputs	pGetter	*Function
		Description: the sensor data getter.	
		pNotification	*Function
		Description: the called function to notify an update	
	outputs	N/A	
		description:	
	input/output	N/A	
		description:	
Return	E_OK	0	
	E_NOK	1	
Description	Call this api to remove subscription, with the data getter and the function to call to notify update.		

Name	pGetter	
Type	*function	
Range	Door_GetCurrentState	To get door state
	Light_GetCurrentState	To get light state
	Speed_GetCurrentState	To get speed state
Description	These data getters	

Name	pNotification	
Type	*function	
Range	Control_DoorUpdate_Clbk	To call when Door data updated
	Control_LightUpdate_Clbk	To call when light data updated
	Control_SpeedUpdate_Clbk	To call when speed data updated
Description	These notifications APIs when data associated updated	

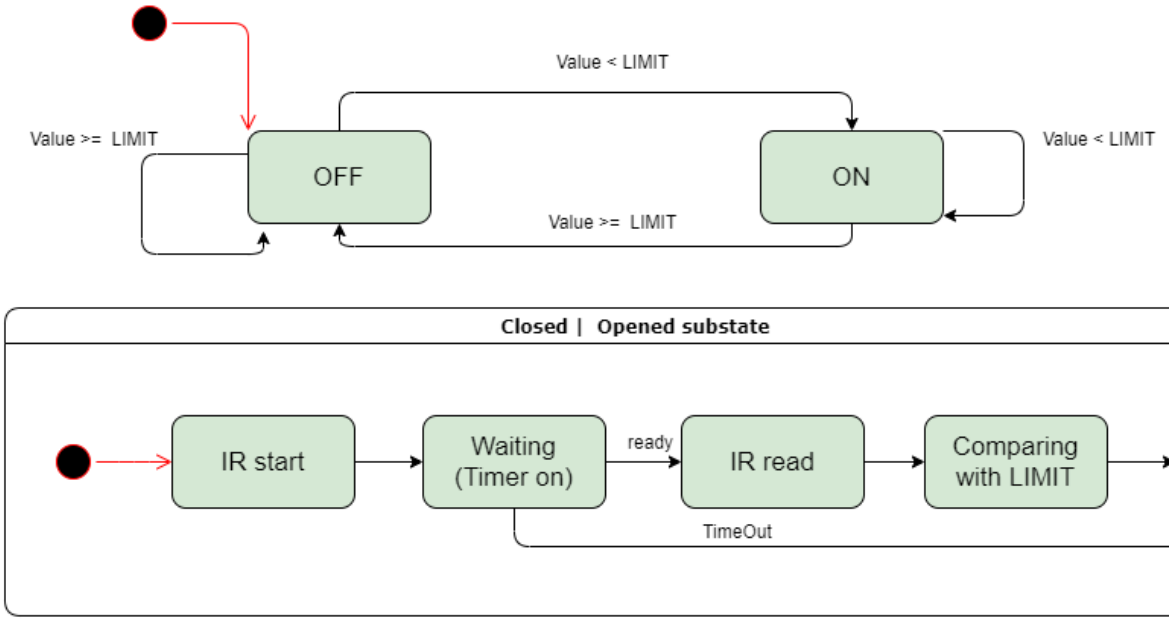
5 – Control module:

Function Name	Error_t Control_init(void);		
Arguments	inputs	N/A	
		Description:	
	outputs	N/A	
		description:	
	input/output	N/A	
		description:	
Return	E_OK	0	
	E_NOK	1	
Description	call this api to initialize the Door Sensor module as in control_config.h		

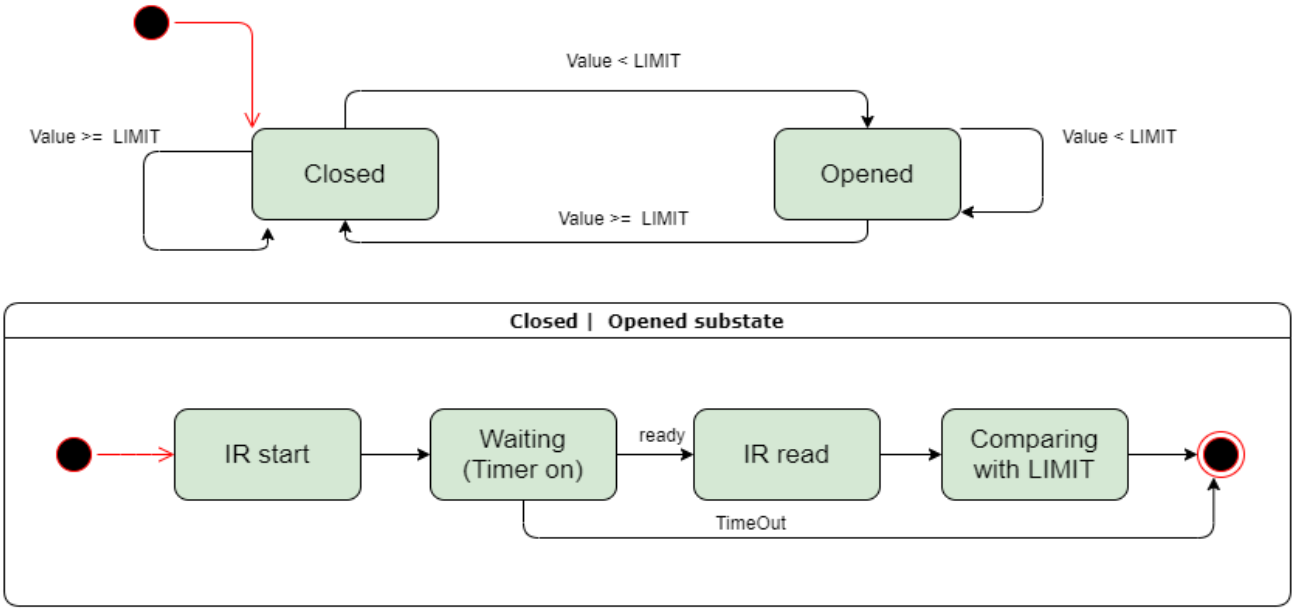
Function Name	Error_t Control_Update(void);		
Arguments	inputs	N/A	
		Description:	
	outputs	N/A	
		description:	
	input/output	N/A	
		description:	
Return	E_OK	0	
	E_NOK	1	
Description	call this api periodically as it is the main function of the module and contain the module state machine		

2 – Dynamic Design:

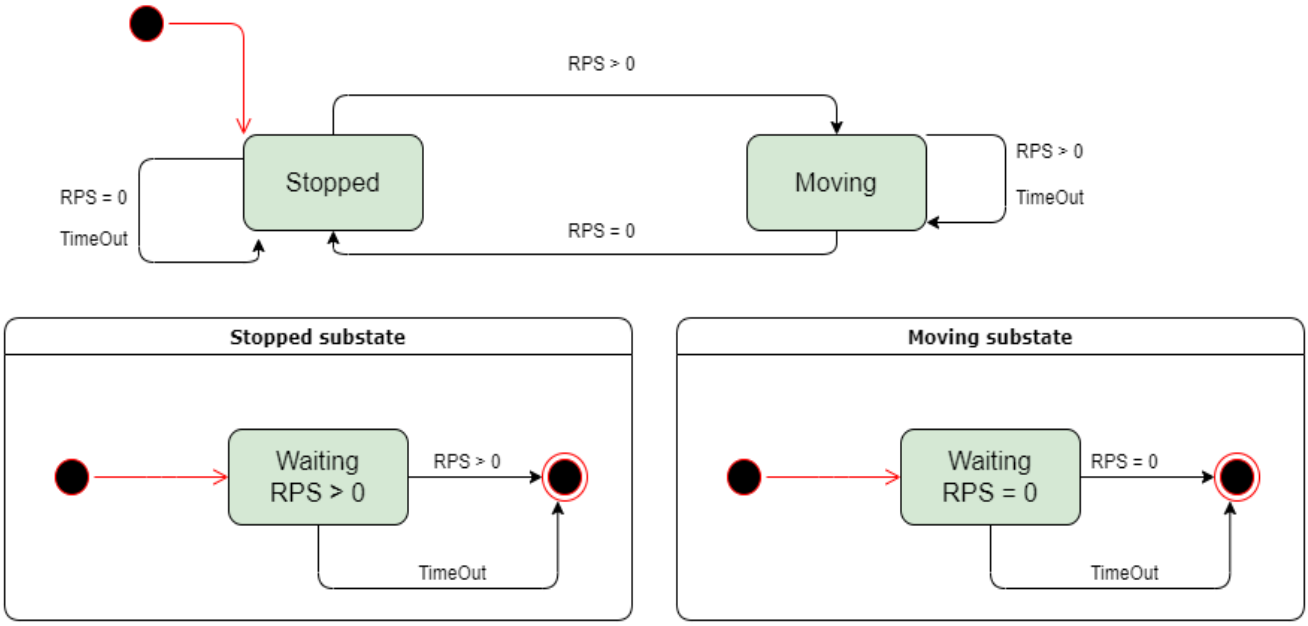
A- Light Module state machine



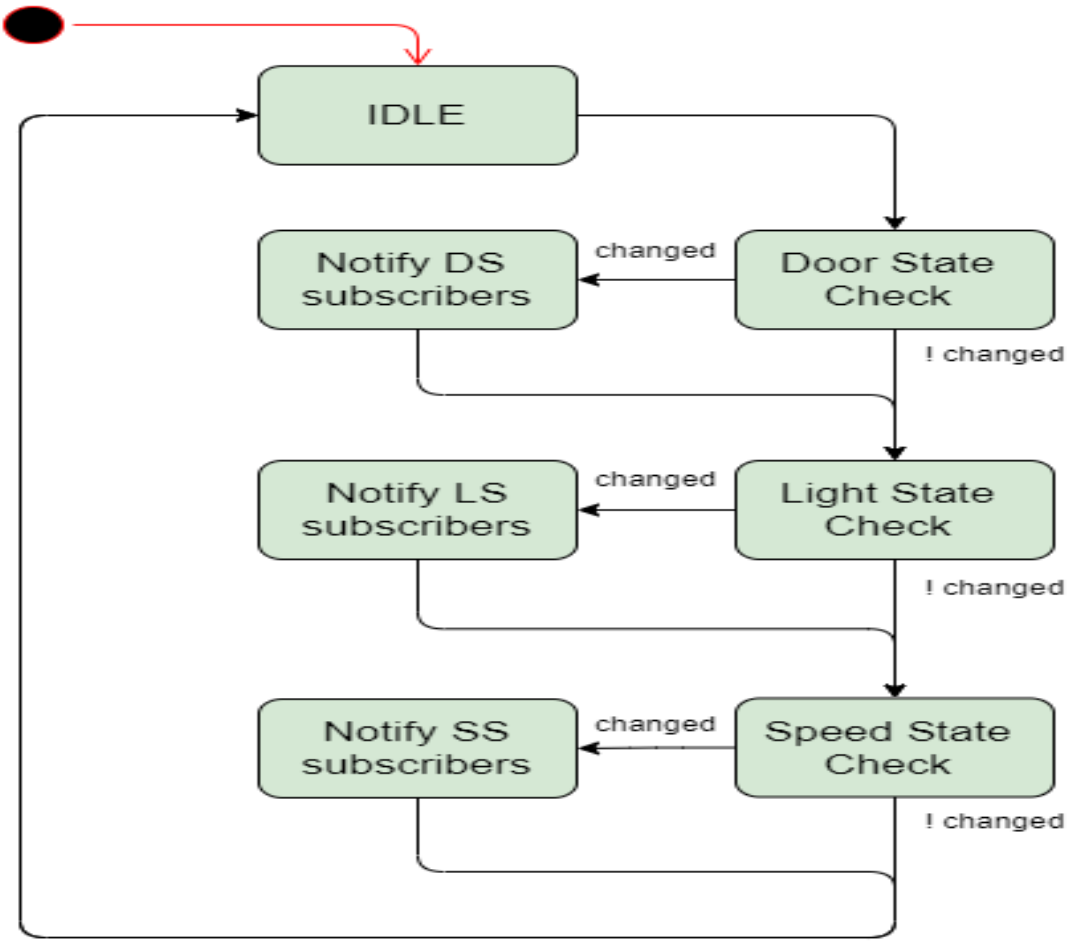
B- Door module state machine



C- Speed module state machine

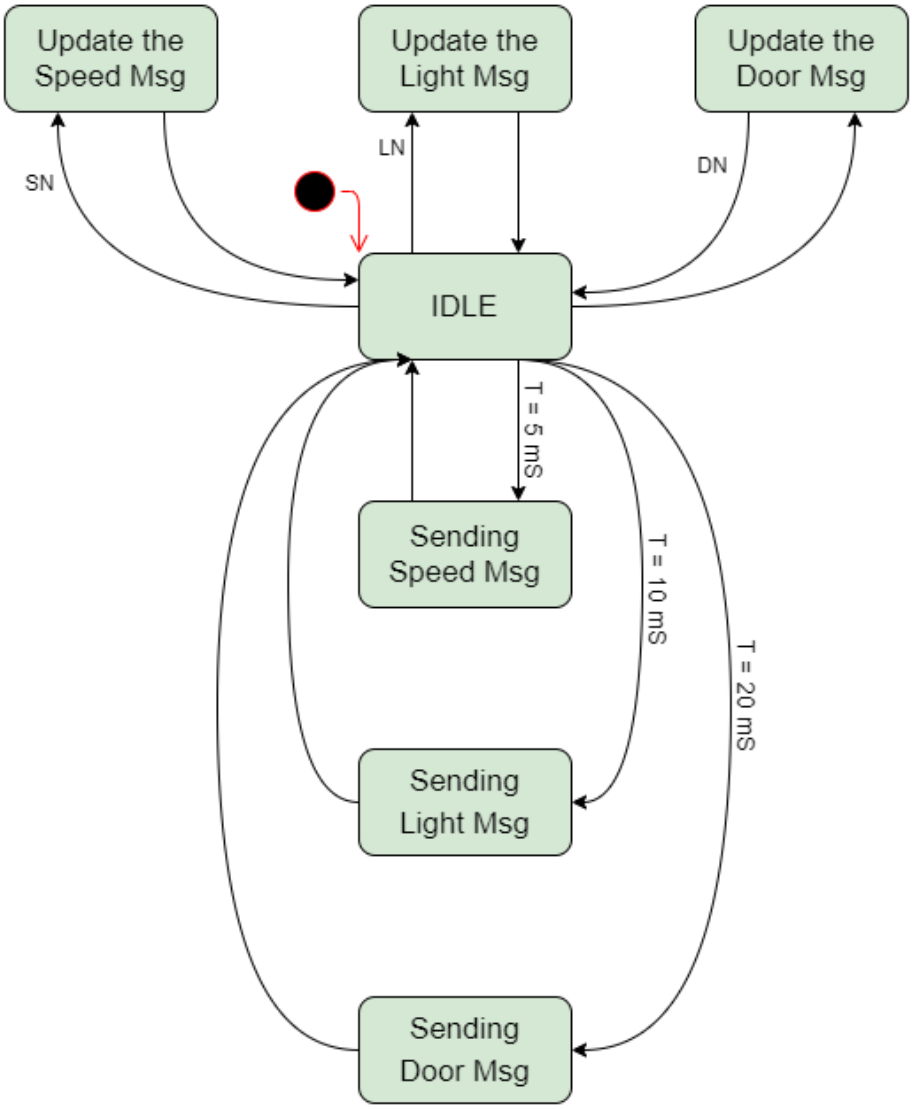


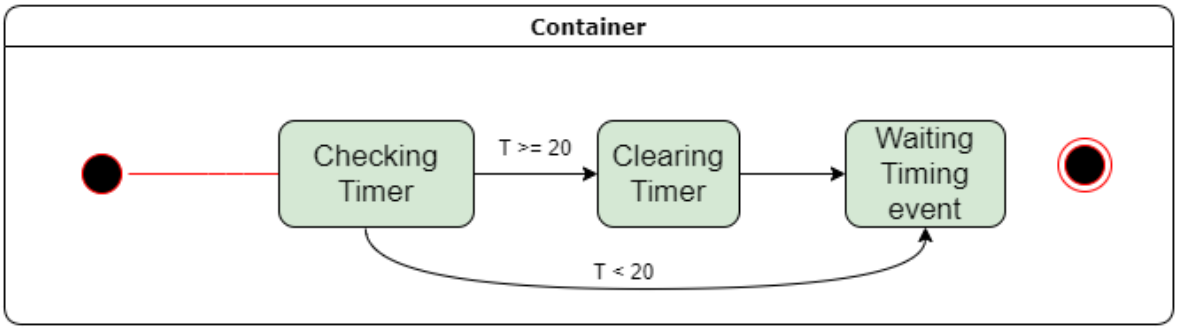
D- Sensor observing module state machine



E- Control module

SN	Speed Update notification
LN	Light Update notification
DN	Door Update notification

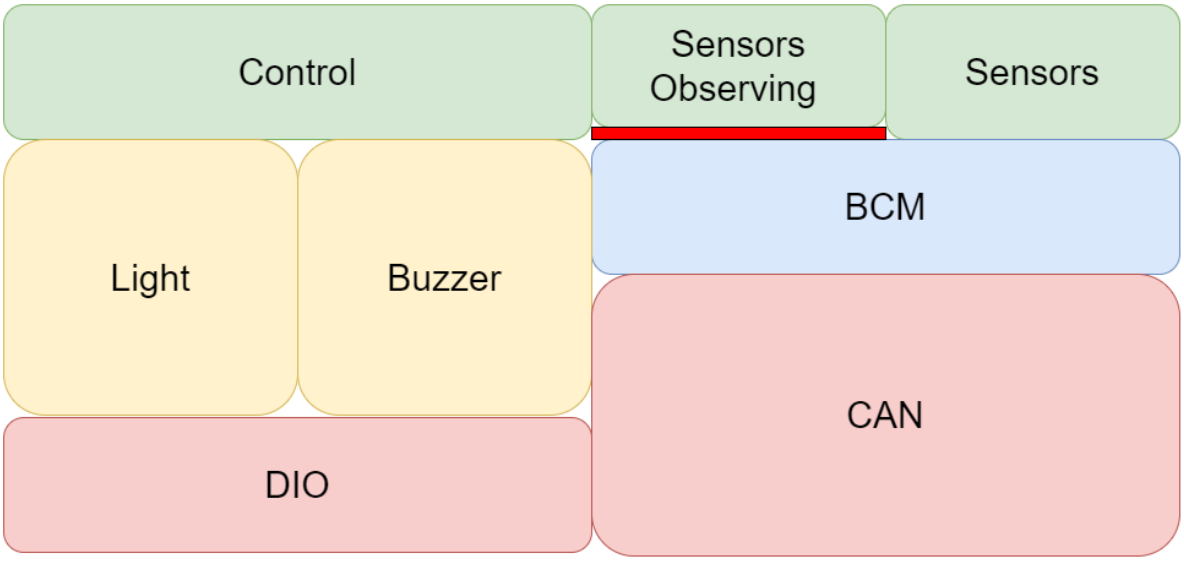




Second – Actuators ECU

1 – Static design:

A- Layered architecture



B- System Parameters

Name	gSysPar		
Type	struct		
elements	DoorSensorState	DoorState_t	Contain global current Door state
	LightSensorState	LightState _t	Contain global current light state
	SpeedSensorState	MovingState_t	Contain global current moving state
	BuzzerState	BuzzerState_t	Contain global current Buzzer state
	LightState	LightState _t	Contain global current light state
Description	This a global structure containing the current states of the sensors		

C- APIs documentation.

1 – Sensors Module:

Function Name	Error_t Sensors_init(void);		
Arguments	inputs	N/A	
		Description:	
	outputs	N/A	
		description:	
	input/output	N/A	
		description:	
Return	E_OK	0	
	E_NOK	1	
Description	call this api to initialize the Door Sensor module as in Sensors_config.h		

Function Name	Error_t Sensors_Update(void);		
Arguments	inputs	N/A	
		Description:	
	outputs	N/A	
		description:	
	input/output	N/A	
		description:	
Return	E_OK	0	
	E_NOK	1	
Description	call this api periodically as it is the main function of the module and contain the module state machine		

Function Name	Sensors_GetDoorState(DoorState_t* pReturnDoorState);		
Arguments	inputs	N/A	
		Description:	
	outputs	pReturnDoorState	enumeration
		description:	
	input/output	N/A	
		description:	
Return	E_OK	0	
	E_NOK	1	
Description	call this api to get the current door state		

Name	pReturnDoorState		
Type	enumeration		
Range	CLOSED	0	Indicate its currently closed
	OPENED	1	Indicate its currently opened
Description	These values are to determine the current door state		

Function Name	Sensors_GetLightState(LightState_t* pReturnLightState);		
Arguments	inputs	N/A	
		Description:	
	outputs	pReturnLightState	enumeration
		description:	
	input/output	N/A	
		description:	
Return	E_OK	0	
	E_NOK	1	
Description	call this api to get the current light state		

Name	pReturnLightState		
Type	enumeration		
Range	OFF	0	Indicate its currently closed
	ON	1	Indicate its currently opened
Description	These values are to determine the current light state		

Function Name	Sensors_GetMovingState(MovingState_t* pReturnMovingState)		
Arguments	inputs	N/A	
		Description:	
	outputs	pReturnMovingState	enumeration
		description:	
	input/output	N/A	
		description:	
Return	E_OK	0	
	E_NOK	1	
Description	call this api to get the current light state		

Name	pReturnMovingState		
Type	enumeration		
Range	STOPPED	0	Indicate its currently closed
	MOVING	1	Indicate its currently opened
Description	These values are to determine the current moving state		

2 – Sensors Observer:

Function Name	Error_t Observer_init(void);		
Arguments	inputs	N/A	
		Description:	
	outputs	N/A	
		description:	
	input/output	N/A	
		description:	
Return	E_OK	0	
	E_NOK	1	
Description	call this api to initialize the Door Sensor module as in SenObserver_config.h		

Function Name	Error_t Observer_Notify(void);		
Arguments	inputs	N/A	
		Description:	
	outputs	N/A	
		description:	
	input/output	N/A	
		description:	
Return	E_OK	0	
	E_NOK	1	
Description	call this api periodically as it is the main function of the module and contain the module state machine		

Function Name	Observer_Subscribe(Error_t (*pGetter)(void* pReturnData), Error_t (*pNotification)(void))		
Arguments	inputs	pGetter	*Function
		Description: the sensor data getter.	
		pNotification	*Function
		Description: the called function to notify an update	
	outputs	N/A	
		description:	
	input/output	N/A	
		description:	
Return	E_OK	0	
	E_NOK	1	
Description	Call this api to set a new subscription, with the data getter and the function to call to notify update.		

Name	pGetter	
Type	*function	
Range	Door_GetCurrentState	To get door state
	Light_GetCurrentState	To get light state
	Speed_GetCurrentState	To get speed state
Description	These data getters	

Name	pNotification	
Type	*function	
Range	Control_DoorUpdate_Clbk	To call when Door data updated
	Control_LightUpdate_Clbk	To call when light data updated
	Control_SpeedUpdate_Clbk	To call when speed data updated
Description	These notifications APIs when data associated updated	

Function Name	Observer_UnSubscribe(Error_t (*pGetter)(void* pReturnData), Error_t (*pNotification)(void))		
Arguments	inputs	pGetter	*Function
		Description: the sensor data getter.	
		pNotification	*Function
		Description: the called function to notify an update	
	outputs	N/A	
		description:	
	input/output	N/A	
		description:	
Return	E_OK	0	
	E_NOK	1	
Description	Call this api to remove subscription, with the data getter and the function to call to notify update.		

Name	pGetter	
Type	*function	
Range	Door_GetCurrentState	To get door state
	Light_GetCurrentState	To get light state
	Speed_GetCurrentState	To get speed state
Description	These data getters	

Name	pNotification	
Type	*function	
Range	Control_DoorUpdate_Clbk	To call when Door data updated
	Control_LightUpdate_Clbk	To call when light data updated
	Control_SpeedUpdate_Clbk	To call when speed data updated
Description	These notifications APIs when data associated updated	

3 – Control module:

Function Name	Error_t Control_init(void);		
Arguments	inputs	N/A	
		Description:	
	outputs	N/A	
		description:	
	input/output	N/A	
		description:	
Return	E_OK	0	
	E_NOK	1	
Description	call this api to initialize the Door Sensor module as in control_config.h		

Function Name	Error_t Control_Update(void);		
Arguments	inputs	N/A	
		Description:	
	outputs	N/A	
		description:	
	input/output	N/A	
		description:	
Return	E_OK	0	
	E_NOK	1	
Description	call this api periodically as it is the main function of the module and contain the module state machine		

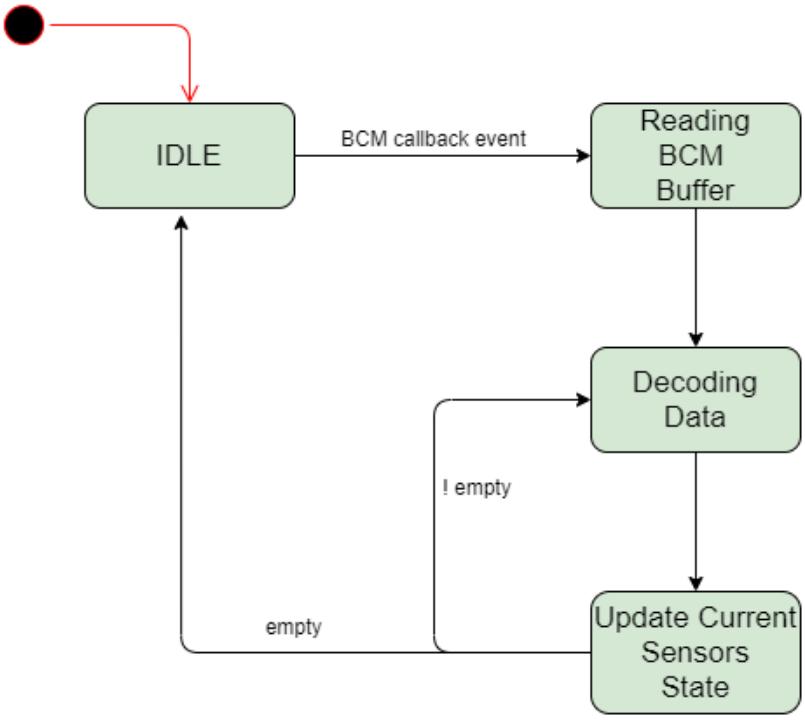
Function Name	Error_t Control_DoorUpdate_Clbk(void);		
Arguments	inputs	N/A	
		Description:	
	outputs	N/A	
		description:	
	input/output	N/A	
		description:	
Return	E_OK	0	
	E_NOK	1	
Description	The Sensors observer mngr supposed to call this api to indicates an update happened to Door state		

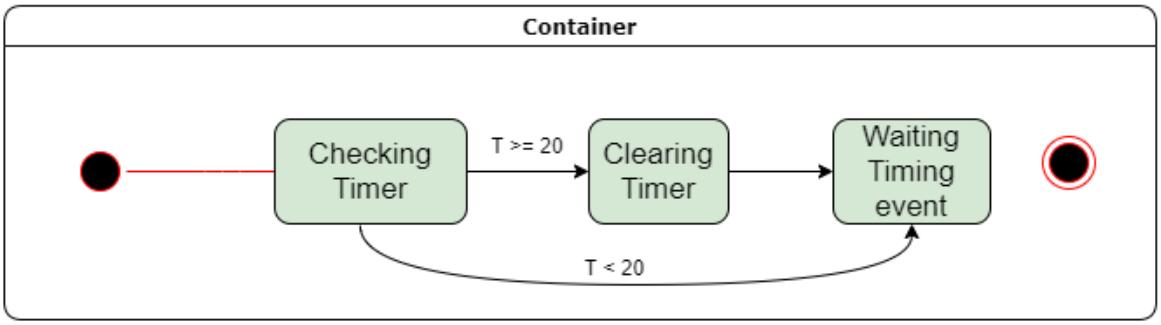
Function Name	Error_t Control_LightUpdate_Clbk(void);		
Arguments	inputs	N/A	
		Description:	
	outputs	N/A	
		description:	
	input/output	N/A	
		description:	
Return	E_OK	0	
	E_NOK	1	
Description	The Sensors observer mngr supposed to call this api to indicates an update happened to light state		

Function Name	Error_t Control_SpeedUpdate_Clbk(void);		
Arguments	inputs	N/A	
		Description:	
	outputs	N/A	
		description:	
	input/output	N/A	
		description:	
Return	E_OK	0	
	E_NOK	1	
Description	The Sensors observer mngr supposed to call this api to indicates an update happened to moving state		

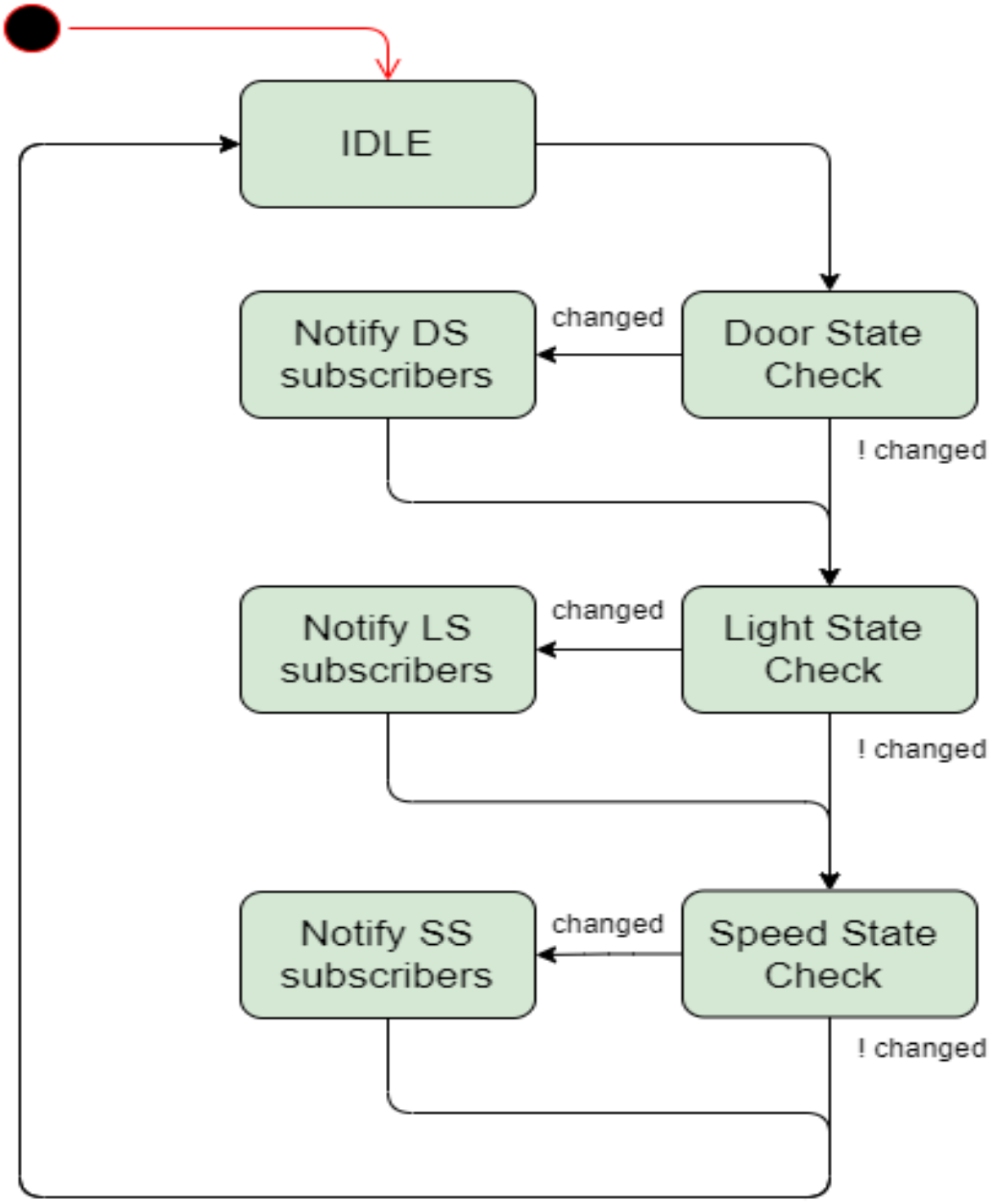
2 – Dynamic Design:

A- Sensor Module

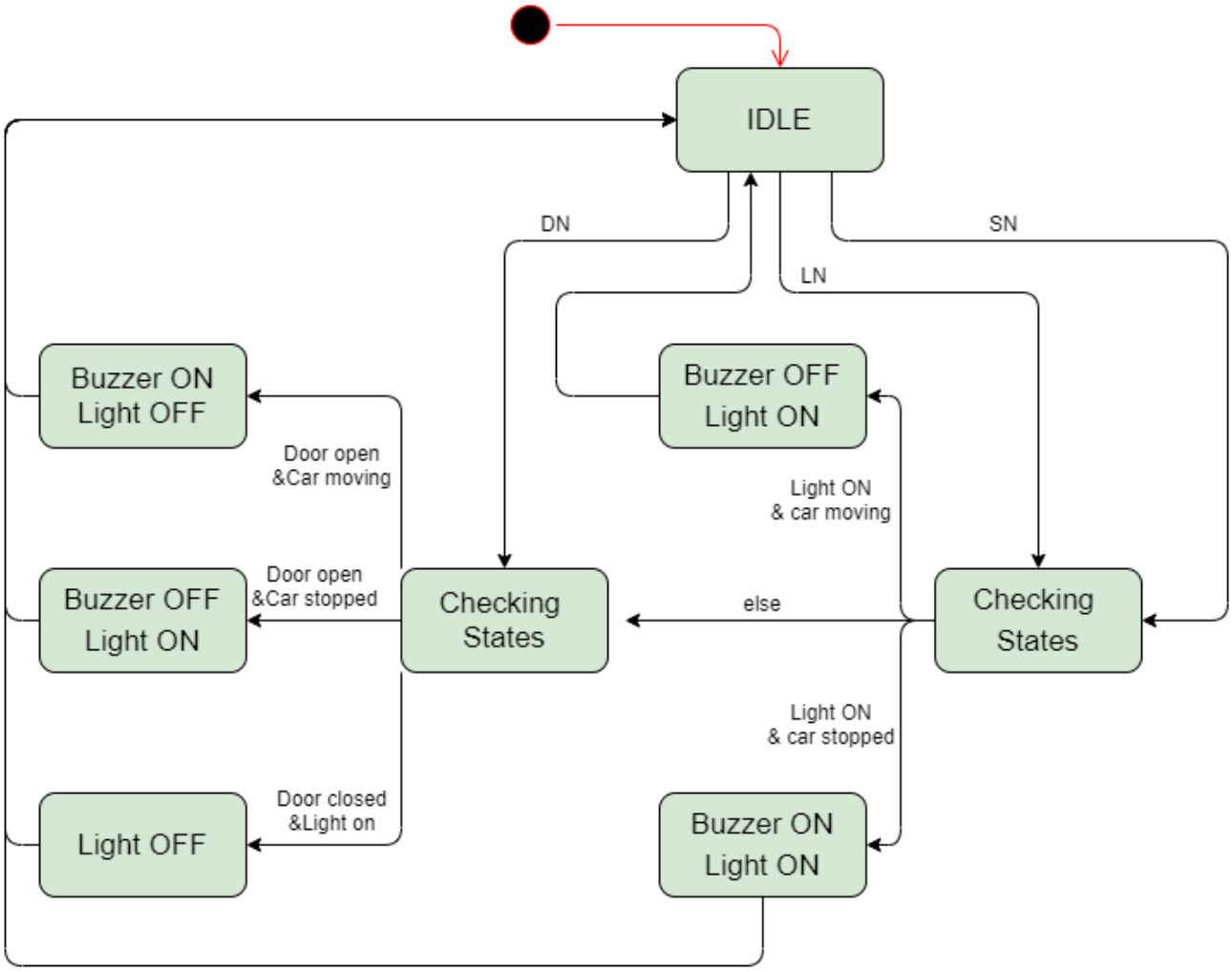


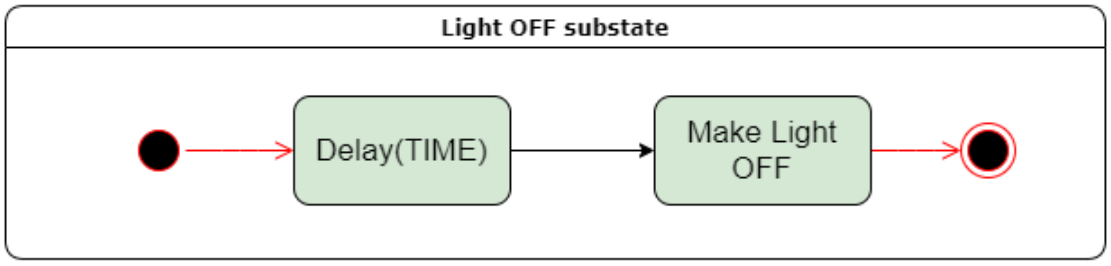


B- Sensor Observer module



C- Control module





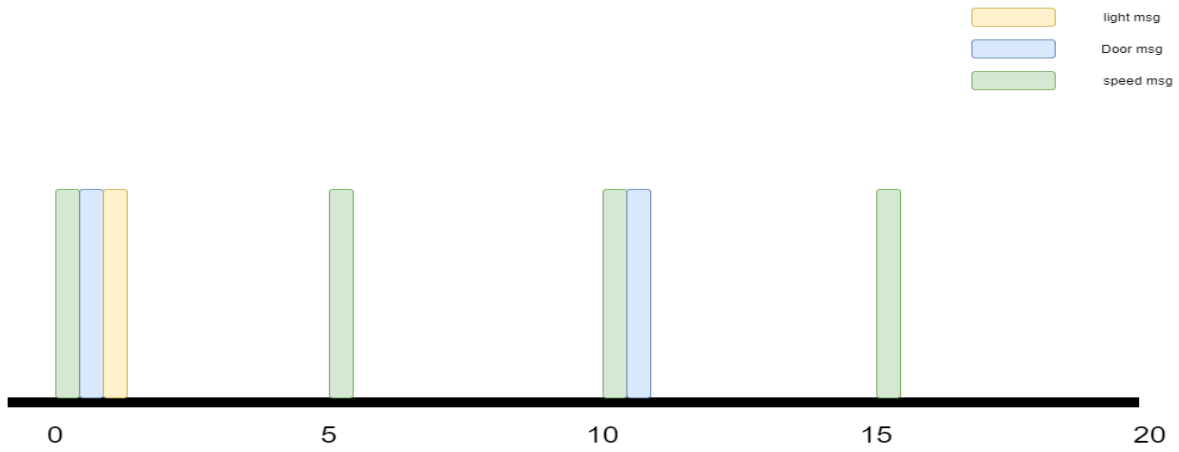
Bus Load:

Assume 500Kbit/s bit rate:

Given One can message contain nearly 125 bit.

Then one message time = 125/ (500,000) = 0.25 mS.

During a major cycle (20mS):



Number of messages = 7

Time of all msg = 7 x 0.25mS = 1.75mS

Bus load = (1.75 / 20) x 100 = 8.75%