AUTOSAR SIMULATION

Final Project

By/ Mohamed Hafez Mohamed

[DATE]

[COMPANY NAME]

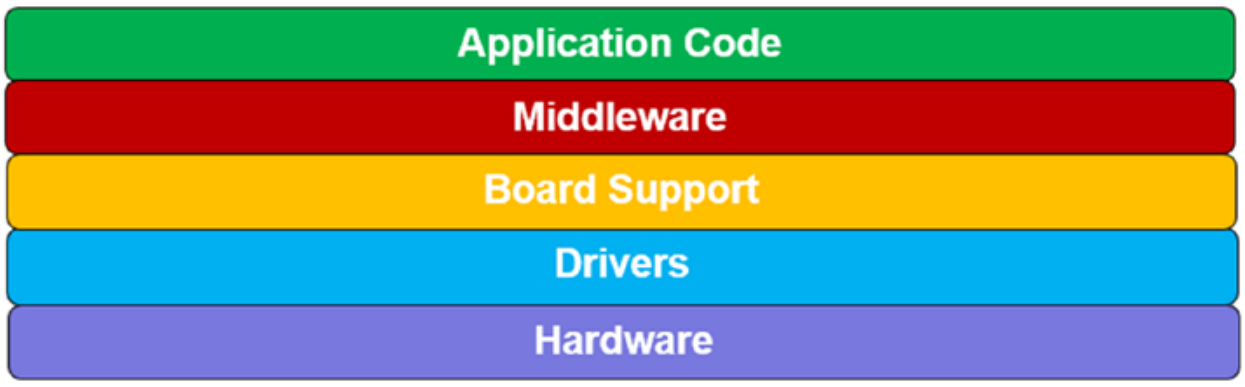
[Company address]

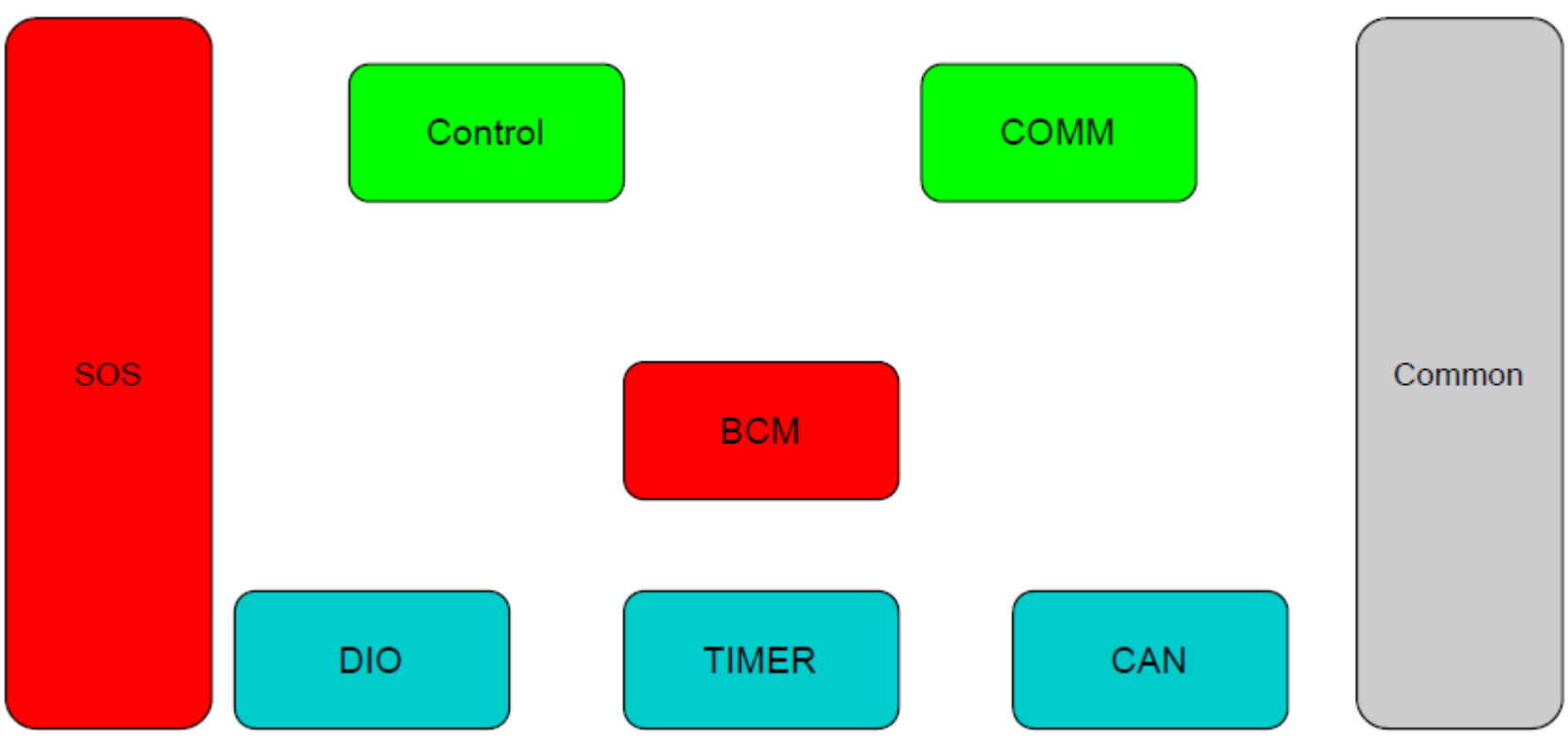
ECU B

1- High Level Design:

1- Static Design:

1- Layered Architecture:





2- API Documentation:

1- DIO Module:

Function name	DIO_InitPin		
Arguments	Input	DioPin	Enumeration
		It’s pin number	
		DioPinMode	Enumeration
		Define pin mode	
	Output	None	
	Input/Output	None	
Return	E_OK	0	
	E_NOK	1	
Description	Responsible for initializing pin mode		

Name:	Dio_Pin		
Type:	Enumeration		
Range	A0 : D0	32	Pin Number
Description:	Pin Number		

Name:	DioPinMode		
Type:	Enumeration		
Range	DIO_INPUT	0	Input pin
	DIO_PUSH_PULL	1	Output pin
Description:	Define Pin Mode		

Function name	DIO_Write		
Arguments	Input	DioPin	Enumeration
		It’s pin number	
		Value	Enumeration
		Define pin value	
	Output	None	
	Input/Output	None	
Return	E_OK	0	
	E_NOK	1	
Description	Responsible for Writing value on physical pin		

Name:	Dio_Pin		
Type:	Enumeration		
Range	A0 : D0	32	Pin Number
Description:	Pin Number		

Name:	Value		
Type:	Enumeration		
Range	LOW	0	Low level volt
	HIGH	1	High level volt
Description:	Define Pin Value		

Function name	Dio_ReadPin		
Arguments	Input	Dio_Pin	Enum
		It’s pin number	
	Output	Pin_Level	U8 *
	Input/Output	None	
Return	E_OK	0	
	E_NOK	1	
Description	Get the value of pin and store it in the Pin_Level pointer		

Name:	Dio_Pin		
Type:	Enumeration		
Range	A0 : D0	32	Pin Number
Description:	Pin Number		

2- Timer Module:

Function name	Timer_Init		
Arguments	Input	None	None
	Output	None	None
	Input/Output	None	
Return	E_OK	0	
	E_NOK	1	
Description	Initialize timer peripheral based on array in configuration file		

Function name	Timer_Start		
Arguments	Input	Channel	Enumeration
		It's Channel number	
		Value	U16
		Define initial value	
	Output	None	
	Input/Output	None	
Return	E_OK	0	
	E_NOK	1	
Description	Responsible for starting timer		

Name:	Timer_Channels		
Type:	Enumeration		
Range	Channel0	0	
	Channel1	1	
	Channel2	2	
Description:	Define Number of channels		

3- Can Module:

Function name	Can_Init		
Arguments	Input	None	None
	Output	None	None
	Input/Output	None	
Return	E_OK	0	
	E_NOK	1	
Description	Initialize can peripheral based on array in configuration file		

Function name	Can_Read		
Arguments	Input	None	None
	Output	Received_value	U8 *
	Input/Output	None	
Return	E_OK	0	
	E_NOK	1	
Description	Read the received value.		

Function name	Can_Write		
Arguments	Input	Data	U32
	Output	None	None
	Input/Output	None	
Return	E_OK	0	
	E_NOK	1	
Description	Transmit Data.		

4- Bcm Module:

Function name	Bcm_Init		
Arguments	Input	None	None
	Output	None	None
	Input/Output	None	
Return	E_OK	0	
	E_NOK	1	
Description	Initialize buffer.		

Function name	Bcm_MainFunction		
Arguments	Input	Data_buffer	U32 *
	Output	None	None
	Input/Output	None	
Return	E_OK	0	
	E_NOK	1	
Description	Send Data.		

5- Control Module:

Function name	Control_Init		
Arguments	Input	None	None
	Output	None	None
	Input/Output	None	
Return	E_OK	0	
	E_NOK	1	
Description	Initialize Actuators.		

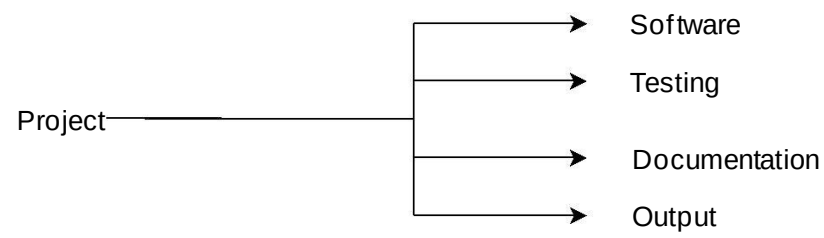
Function name	Control_MainFunction		
Arguments	Input	Buffer	U8 *
	Output	None	None
	Input/Output	None	
Return	E_OK	0	
	E_NOK	1	
Description	Control Actuators.		

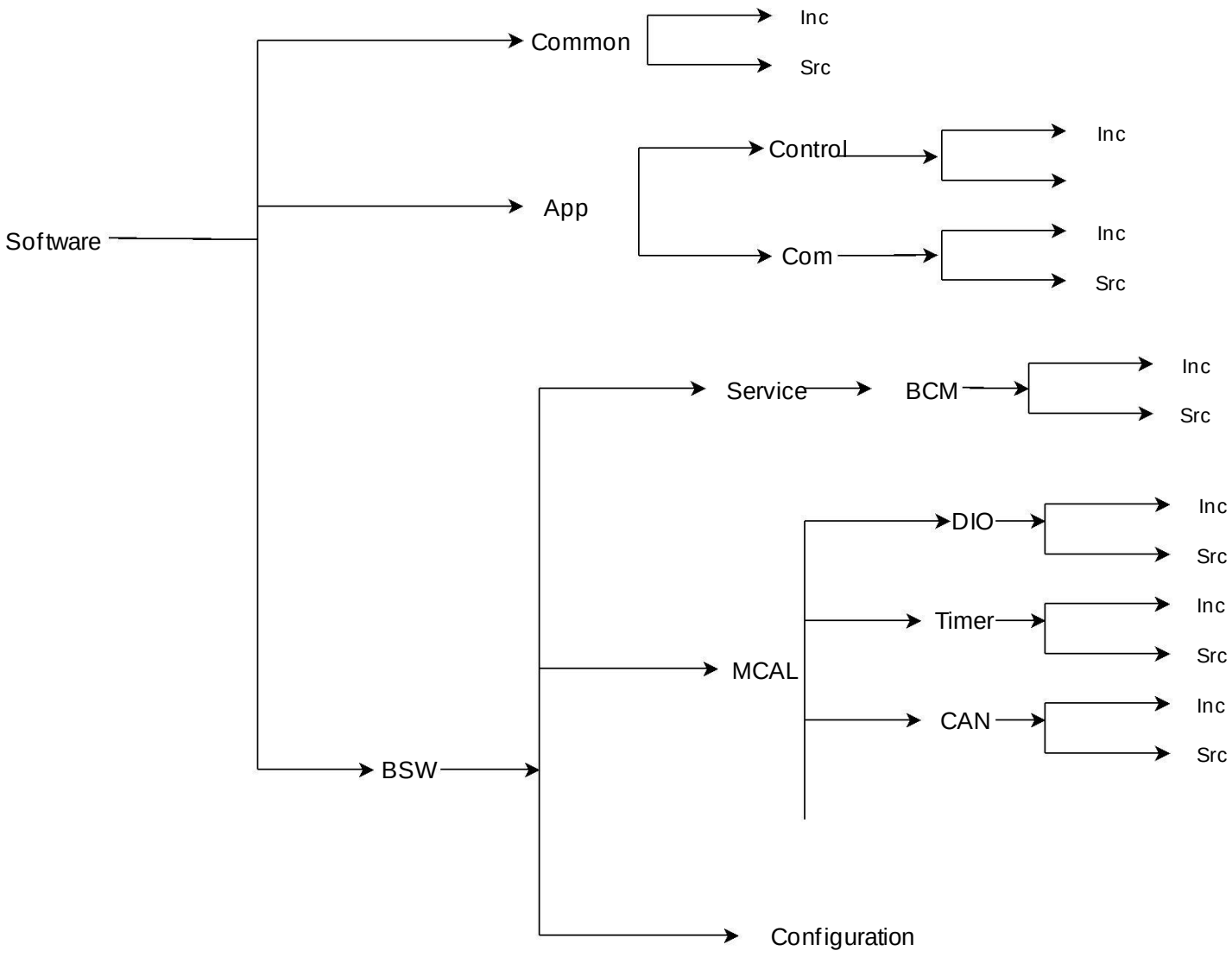
6- Com Module:

Function name	Com_MainFunction		
Arguments	Input	Data_buffer	U32 *
	Output	None	None
	Input/Output	None	
Return	E_OK	0	
	E_NOK	1	
Description	Send Data to the ECU B.		

2- Low Level Design:

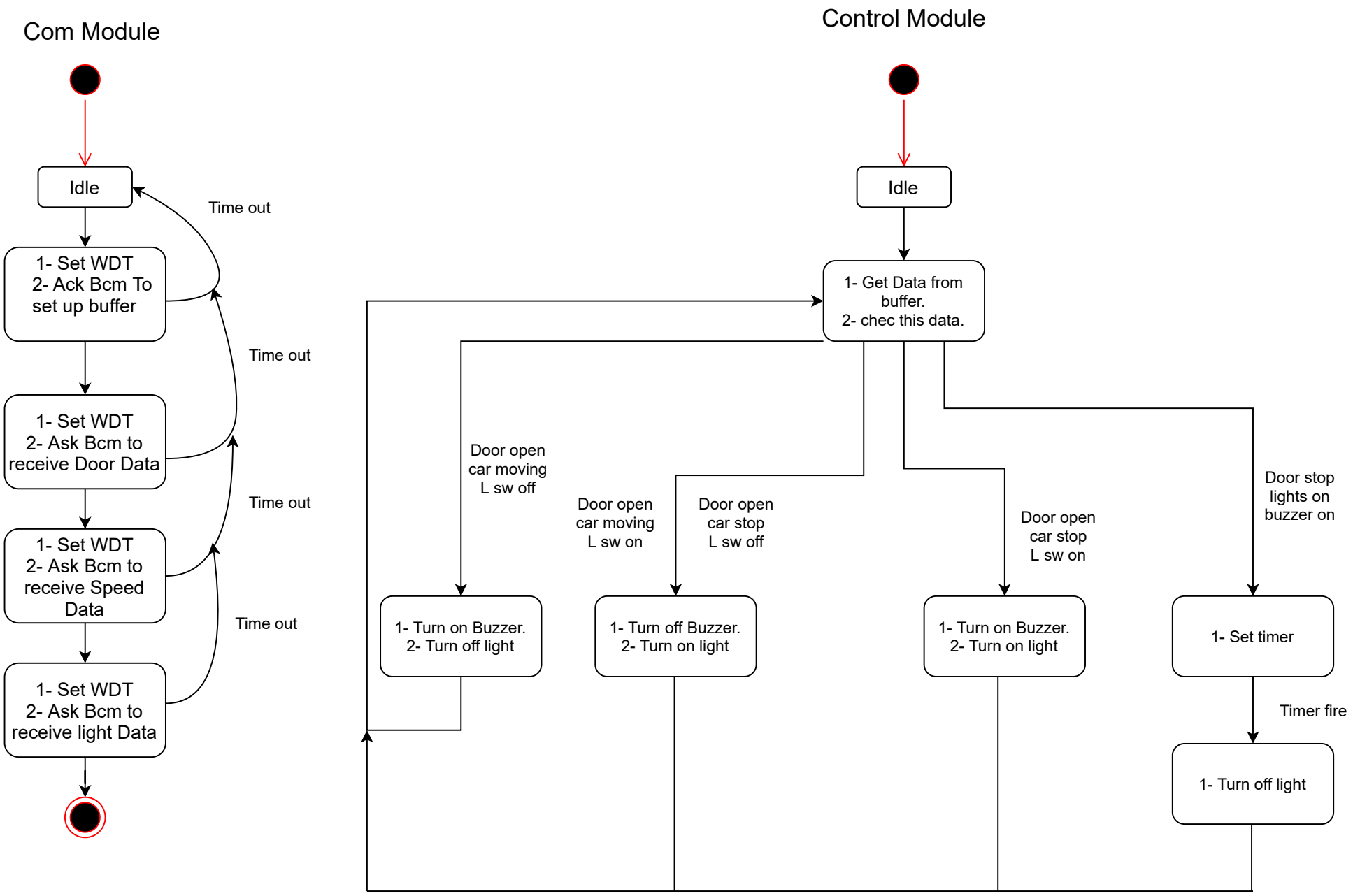
1- Physical Design (Folder Structure):





2- Logical Design:

1- State Machines



2- Sequance Diagram