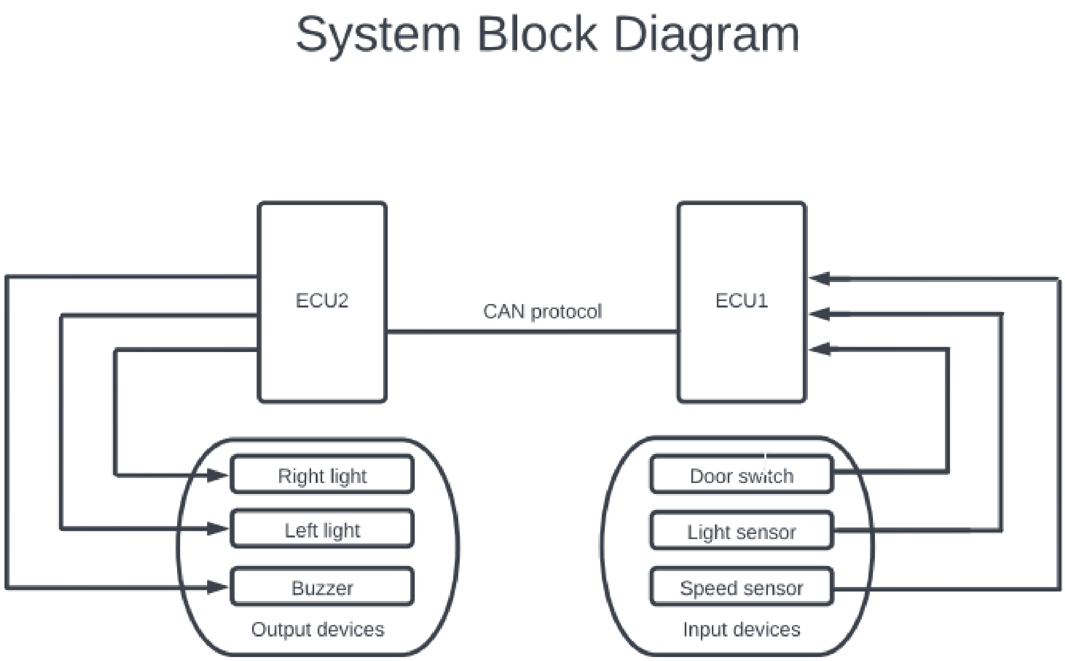
Automotive door control system design

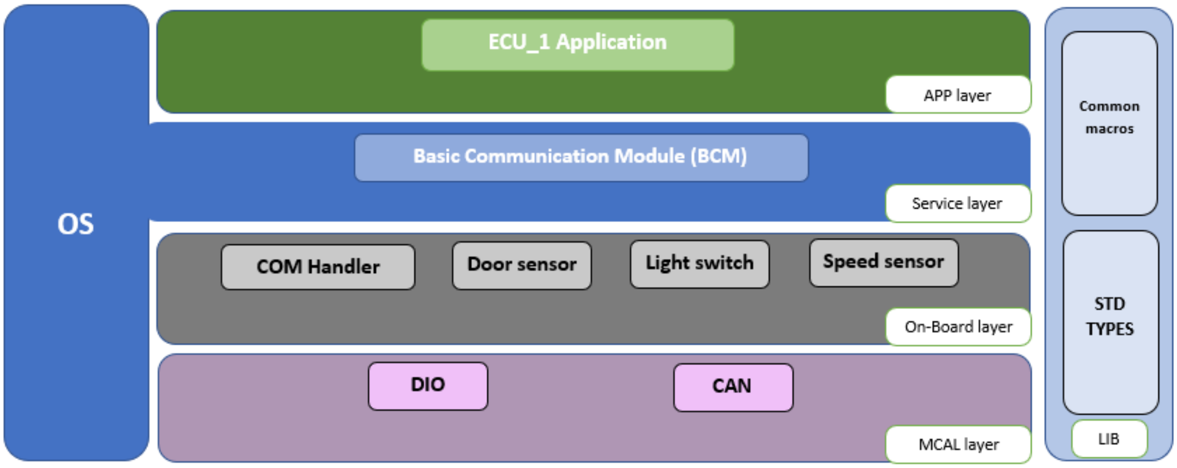
Static design

System block diagram:



ECU 1:

Layered architecture:



ECU 1 components and modules:

1- Digital Input/Output (DIO)

2- CAN protocol

3- Basic Communication Module (BCM)

4- External hardware: Door switch, Light switch, Speed sensor

Detailed APIs for each module as well as a detailed description for the used typedefs:

Functions: for ECU 1

Module	DIO
Function name	MDIO_Void_Init(void)
Arguments	void
Return	void
Description	Initialize DIO module

Module	DIO
Function name	MDIO_void_SetPinDirection
Arguments	enum DIOPORT_t PortID u8 PinNumber enum DIODIRECTION_t Direction
Return	void
Description	Set a pin’s direction to input or output

Module	DIO
Function name	MDIO_void_SetPinValue
Arguments	enum DIOPORT_t PortID, u8 PinNumber, enum DIOSTATE_t Value
Return	void
Description	Set a pin’s value to high or low

Module	DIO
Function name	MDIO_void_GetPinValue
Arguments	enum DIOPORT_t PortID, u8 PinNumber
Return	u8 (value)
Description	Get a pin’s value

Module	CAN
Function name	MCAN_void_INIT
Arguments	void
Return	void
Description	Initialize CAN module

Module	CAN
Function name	MCAN_void_SendByte
Arguments	u8 byte

Return	void
Description	Send a byte using CAN

Module	DoorSensor
Function name	HDoorSensor_Setup
Arguments	void
Return	void
Description	Setup the pins used by the door switch

Module	DoorSensor
Function name	HDoorSensor_Read
Arguments	void
Return	u8 (High or Low)
Description	Read switch state

Module	LightSwitch
Function name	HLightSW_Setup
Arguments	void
Return	void
Description	Setup the pins used by the light switch

Module	LightSwitch
Function name	HLightSW _Read
Arguments	void
Return	u8 (High or Low)
Description	Read switch state

Module	SpeedSensor
Function name	HSpeedSensor_Setup
Arguments	void
Return	void
Description	Setup the pins used by the speed sensor

Module	SpeedSensor
Function name	HSpeedSensor _Read
Arguments	void
Return	u8 data (High or Low)
Description	Setup the pins used by the door switch

Module	BCM
Function name	SBCM_void_Send
Arguments	u8 byte
Return	void
Description	Send a byte using BCM module

Module	COMhandler
Function name	HCOMhandler_Send
Arguments	U8 byte
Return	void

Description	Send a byte using communication handler and choose CAN

Module	RTOS
Function name	prvSetupHardware
Arguments	void
Return	void
Description	Setup hardware used by RTOS

Module	RTOS
Function name	vTaskStartScheduler
Arguments	void
Return	void
Description	Start scheduling algorithm

Enums used:

Module	DIO
Enum name	DIOPORT_t
Range	DIO_PORTB,DIO_PORTC,DIO_PORTD
Description	Selects port

Module	DIO
Enum name	DIODIRECTION_t
Range	DIO_INPUT,DIO_OUTPUT,DIO_INPUT_PULLUP
Description	Selects direction

Module	DIO
Enum name	DIOSTATE_t
Range	DIO_LOW,DIO_HIGH
Description	Selects value

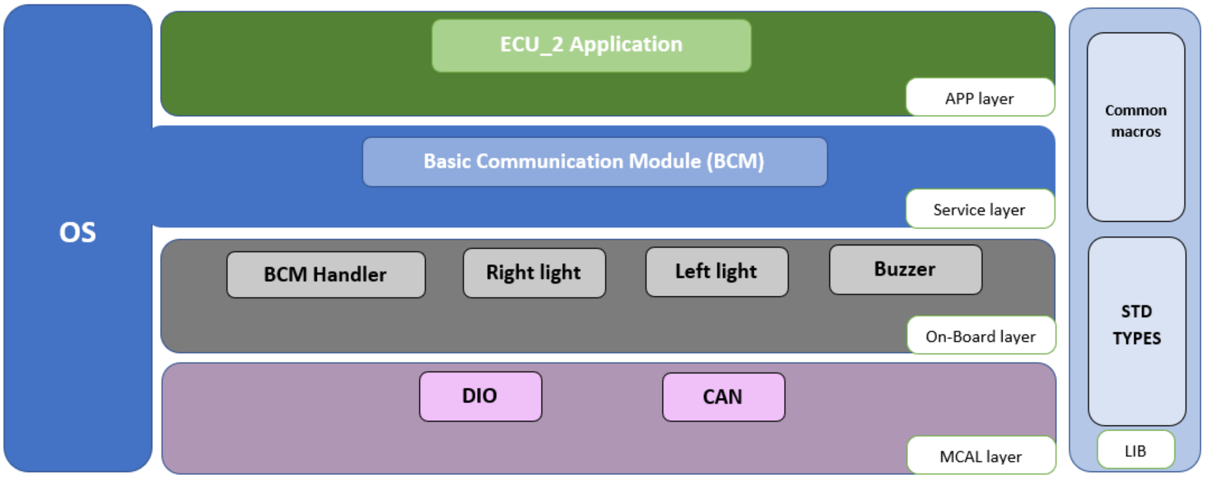
Typedefs and macros:

Module	STDTYPES
typename	unsigned char
New_type	u8

Module	STDTYPES
typename	unsigned short int
New_type	u16

ECU 2:

Layered architecture:



ECU 2 components and modules:

1- Digital Input/Output (DIO)

2- CAN protocol

3- Basic Communication Module (BCM)

4- External hardware: Left light, Right light, Buzzer

Detailed APIs for each module as well as a detailed description for the used typedefs:

Functions: for ECU 2

Module	DIO
Function name	MDIO_Void_Init(void)
Arguments	void
Return	void
Description	Initialize DIO module

Module	DIO
Function name	MDIO_void_SetPinDirection
Arguments	enum DIOPORT_t PortID u8 PinNumber enum DIODIRECTION_t Direction
Return	void
Description	Set a pin's direction to input or output

Module	DIO
Function name	MDIO_void_SetPinValue
Arguments	enum DIOPORT_t PortID, u8 PinNumber, enum DIOSTATE_t Value
Return	void
Description	Set a pin's value to high or low

Module	DIO
Function name	MDIO_void_GetPinValue
Arguments	enum DIOPORT_t PortID, u8 PinNumber
Return	u8 (value)
Description	Get a pin's value

Module	CAN
Function name	MCAN_void_INIT
Arguments	void
Return	void
Description	Initialize CAN module

Module	CAN
Function name	MCAN_u8_ReadByte
Arguments	void
Return	u8 byte
Description	Read a byte using CAN

Module	RightLight
Function name	HRightLight_Setup
Arguments	void
Return	void
Description	Setup the pins used by the right light

Module	RightLight
Function name	HRightLight_Set
Arguments	u8 (high or low)
Return	void
Description	set right light state

Module	LeftLight
Function name	HLeftLight_Setup
Arguments	void
Return	void
Description	Setup the pins used by the left light

Module	LeftLight
Function name	HLeftLight_Set
Arguments	u8 (high or low)
Return	void
Description	set left light state

Module	Buzzer
Function name	HBuzzeer_Setup
Arguments	void
Return	void
Description	Setup the pins used by the buzzer

Module	Buzzer
Function name	HBuzzeer_Set
Arguments	u8 (high or low)
Return	void
Description	set buzzer state

Module	BCM
Function name	SBCM_u8_Read
Arguments	void
Return	u8 byte
Description	Read a byte using BCM module

Module	COMhandler
Function name	HCOMhandler_Read
Arguments	void
Return	u8 byte
Description	Read a byte using communication handler and choose CAN

Module	RTOS
Function name	prvSetupHardware
Arguments	void
Return	void

Description	Setup hardware used by RTOS

Module	RTOS
Function name	vTaskStartScheduler
Arguments	void
Return	void
Description	Start scheduling algorithm

Module	Main App
Function name	Setter
Arguments	void
Return	void
Description	Handle the 5 states of the system operation

Enums used:

Module	DIO
Enum name	DIOPORT_t
Range	DIO_PORTB,DIO_PORTC,DIO_PORTD
Description	Selects port

Module	DIO
Enum name	DIODIRECTION_t
Range	DIO_INPUT,DIO_OUTPUT,DIO_INPUT_PULLUP
Description	Selects direction

Module	DIO
Enum name	DIOSTATE_t
Range	DIO_LOW,DIO_HIGH
Description	Selects value

Typedefs and macros:

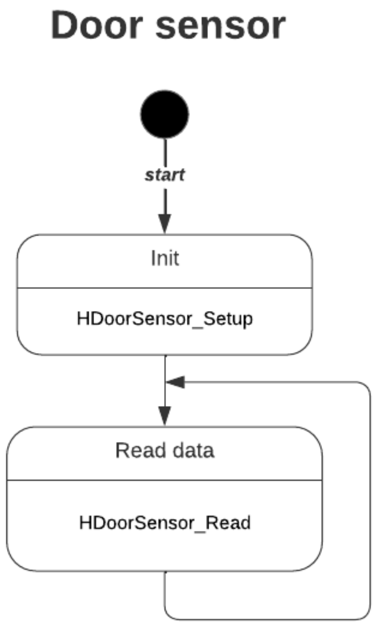
Module	STDTYPES
typename	unsigned char
New_type	u8

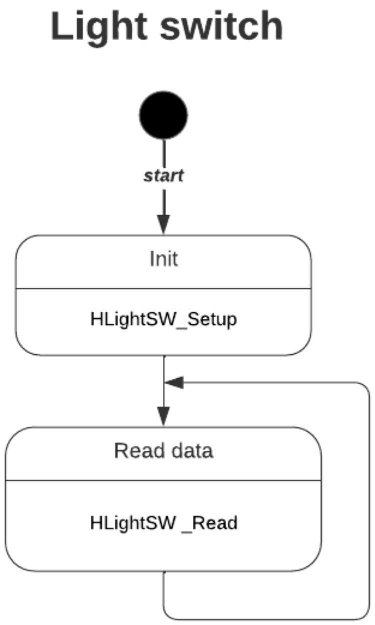
Module	STDTYPES
typename	unsigned short int
New_type	u16

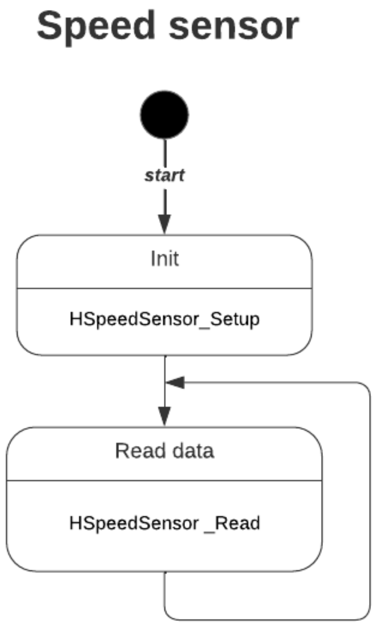
Dynamic design

ECU 1:

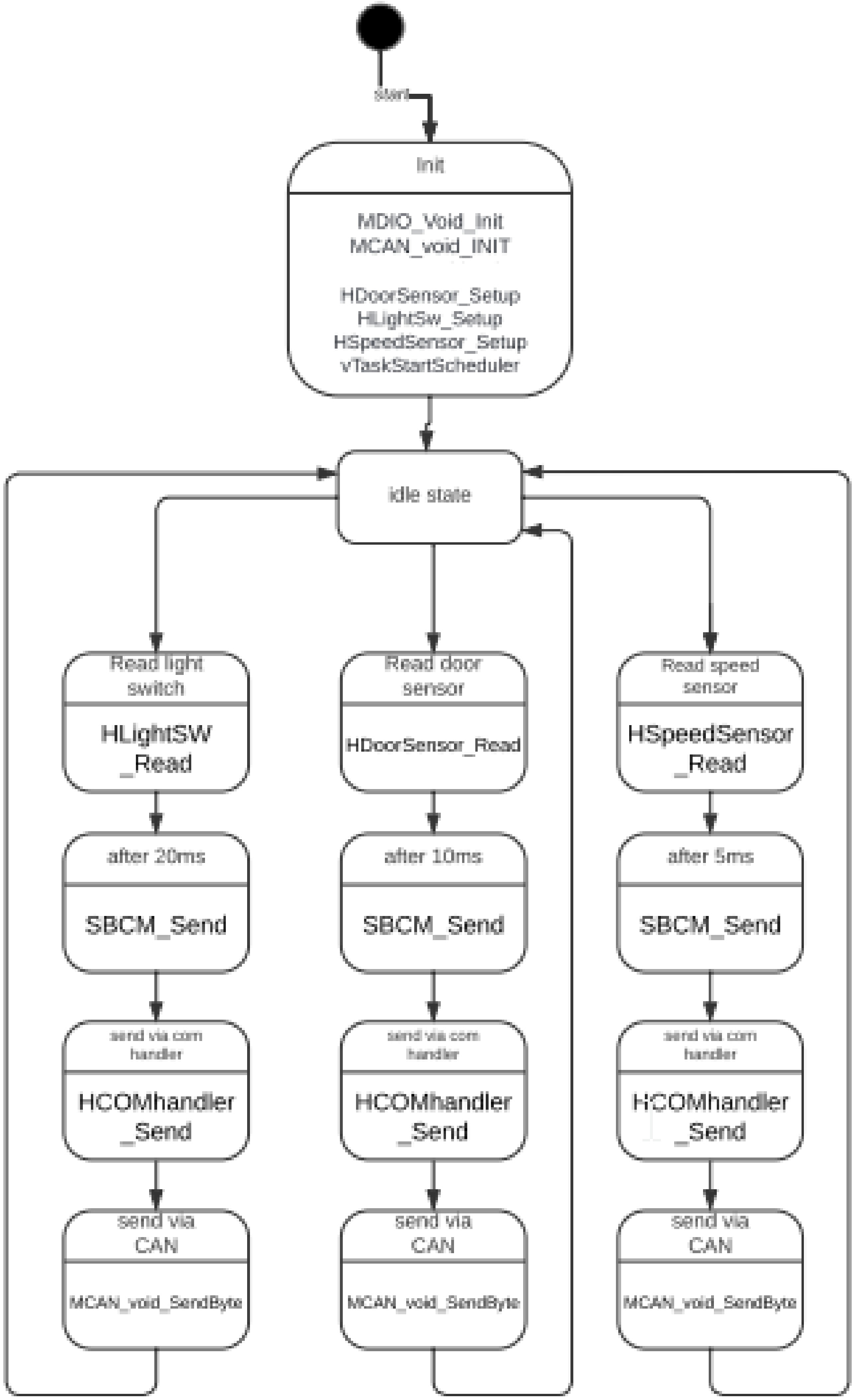
State machine diagram for each ECU component:



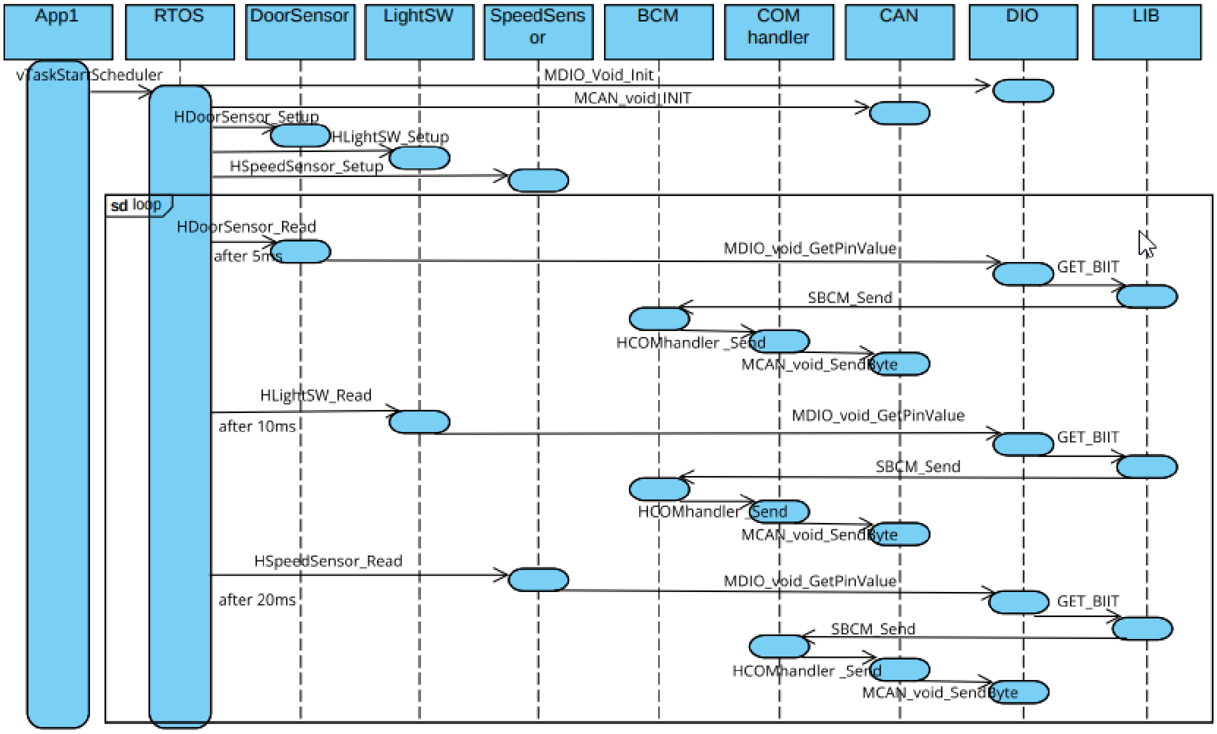




State machine diagram for the ECU operation:



The sequence diagram for the ECU:

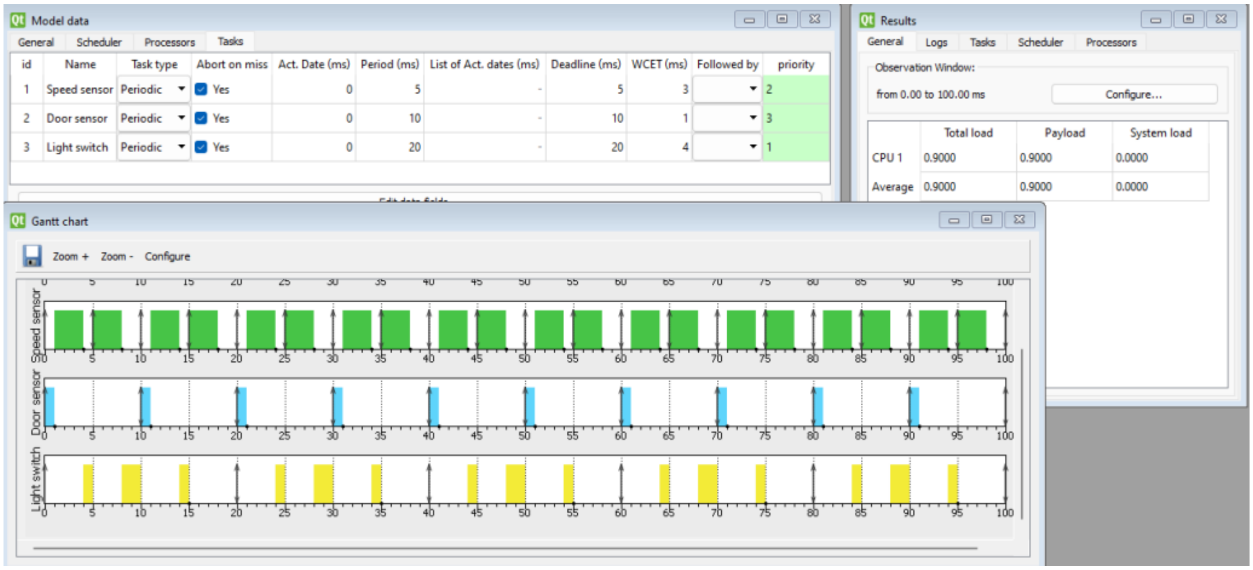


CPU load for the ECU:

Calculating CPU load for ECU1:

Task 1 (Speed sensor): P:5, E:3, D:5
Task 2 (Door sensor): P:10, E:1, D:10
Task 3 (Light switch): P:20, E:4, D:20

Applying the above data on simso: CPU load for ECU 1 is 90%



Model data

id	Name	Task type	Abort on miss	Act. Date (ms)	Period (ms)	List of Act. dates (ms)	Deadline (ms)	WCET (ms)	Followed by	priority
1	Speed sensor	Periodic	☑ Yes	0	5	-	5	3		2
2	Door sensor	Periodic	☑ Yes	0	10	-	10	1		3
3	Light switch	Periodic	☑ Yes	0	20	-	20	4		1

Results

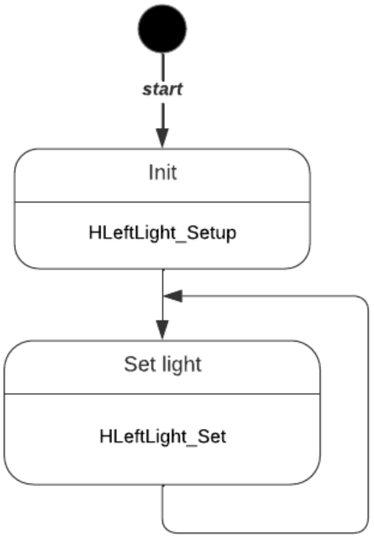
Observation Window:
from 0.00 to 100.00 ms

	Total load	Payload	System load
CPU 1	0.9000	0.9000	0.0000
Average	0.9000	0.9000	0.0000

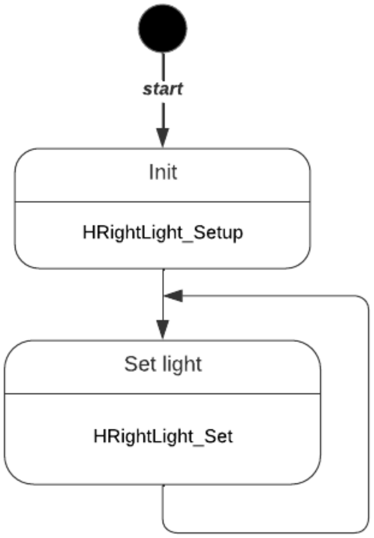
ECU 2:

State machine diagram for each ECU component:

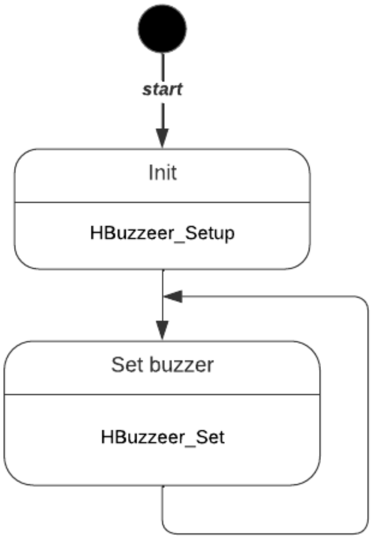
Left light

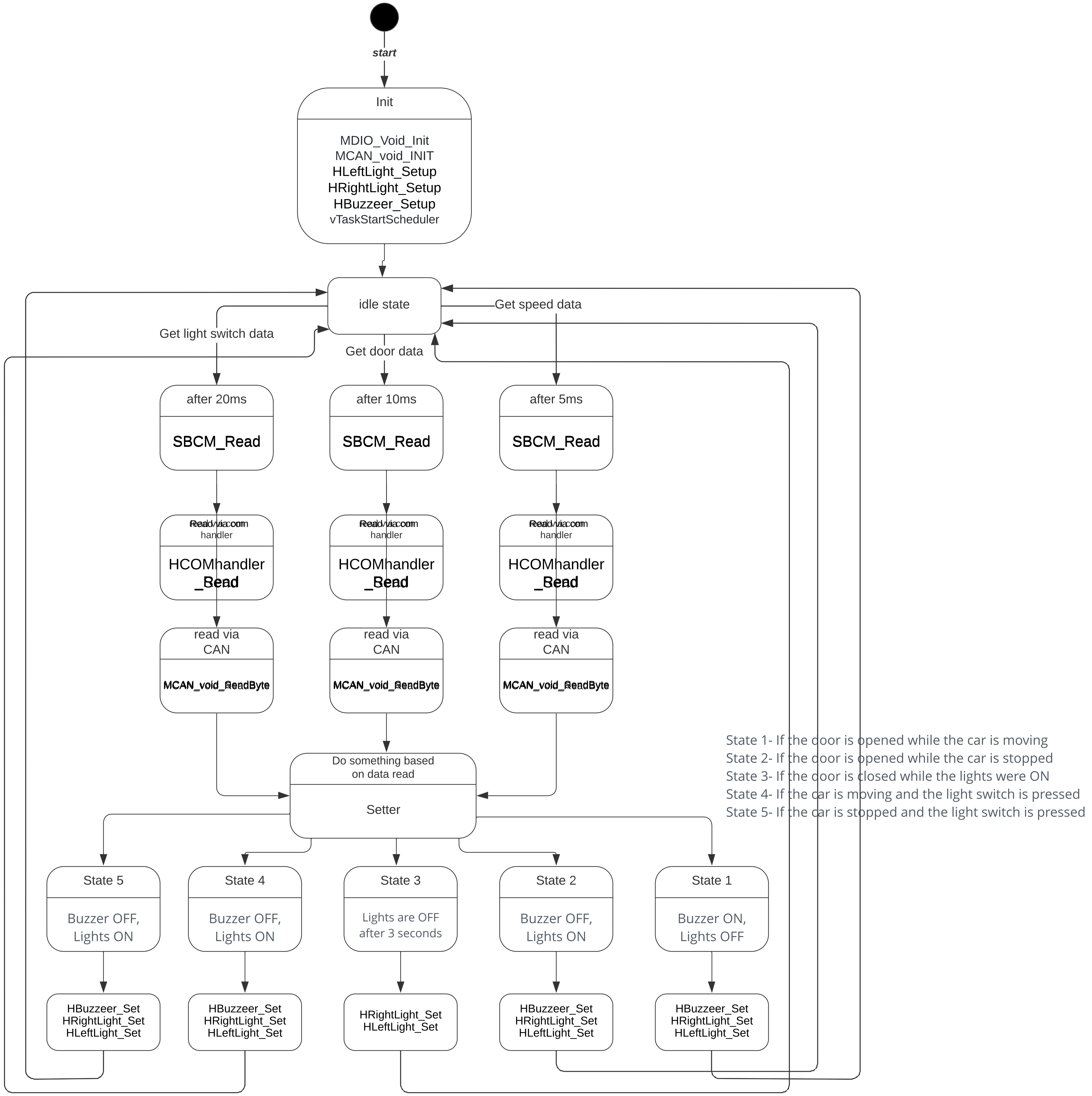


Right light

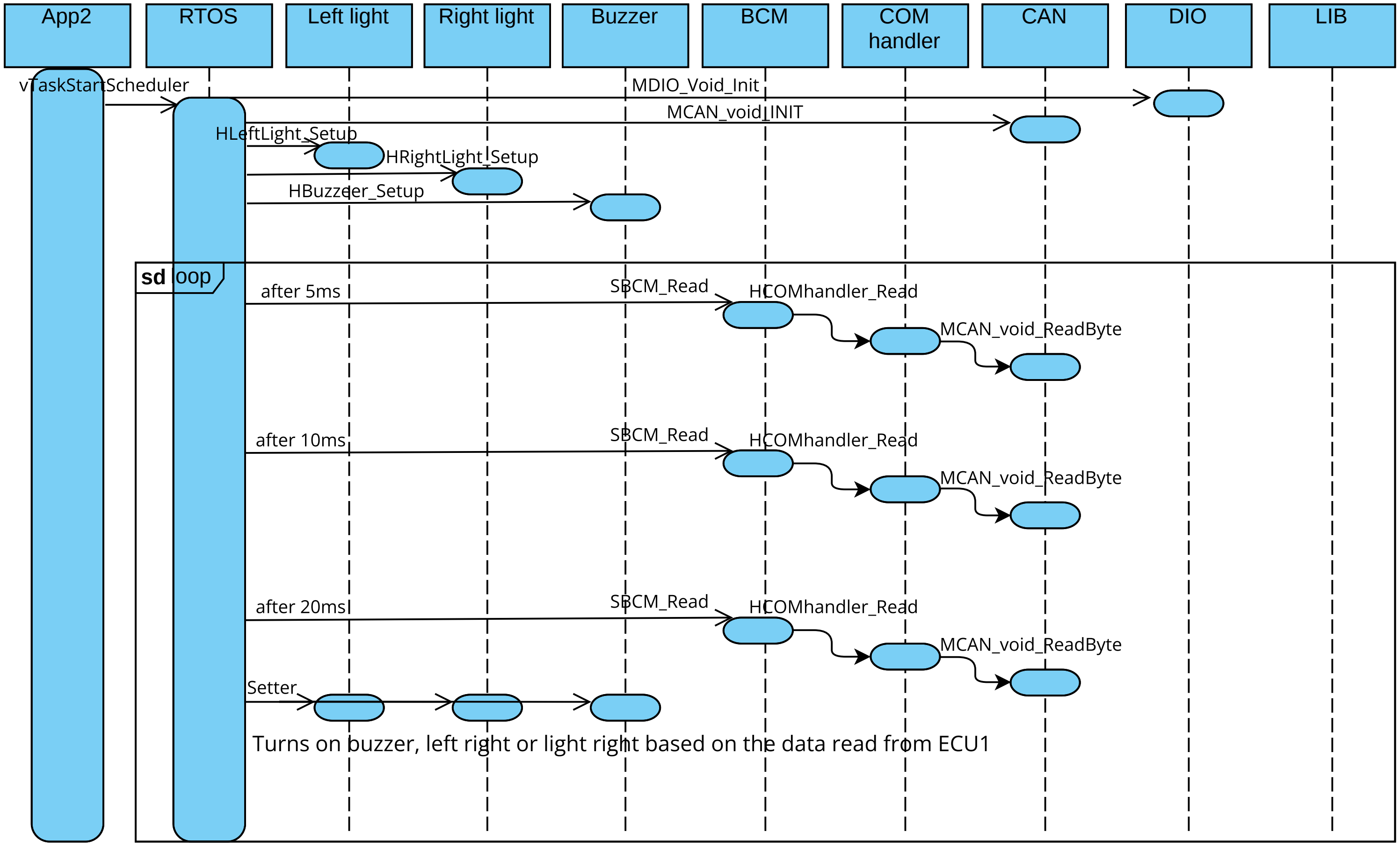


Buzzer





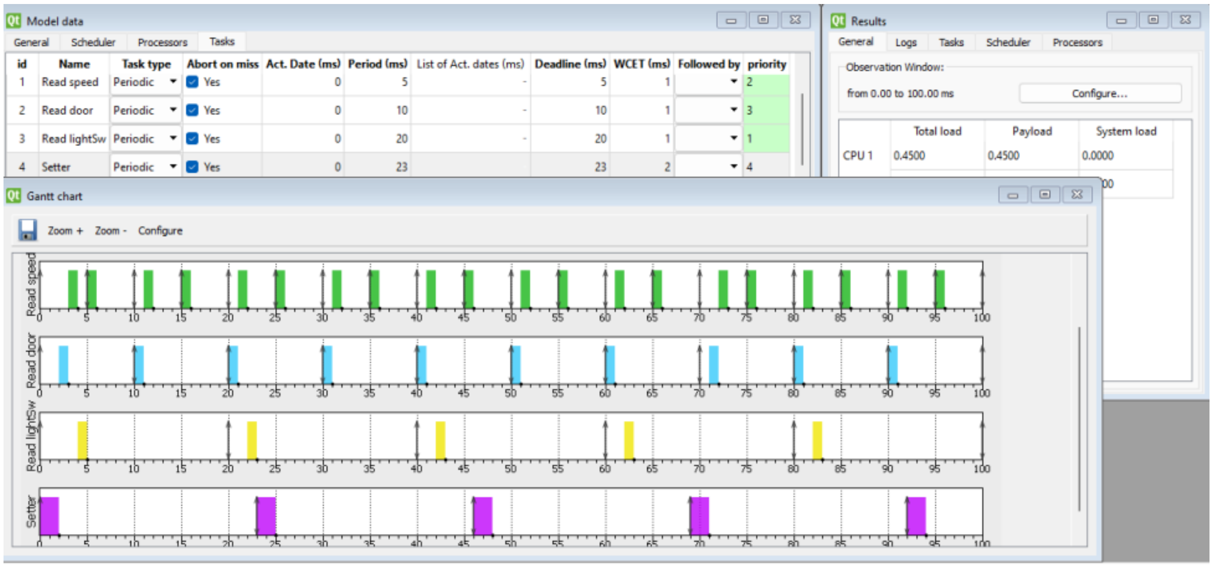
State 1- If the door is opened while the car is moving
State 2- If the door is opened while the car is stopped
State 3- If the door is closed while the lights were ON
State 4- If the car is moving and the light switch is pressed
State 5- If the car is stopped and the light switch is pressed


App2
RTOS
Left light
Right light
Buzzer
BCM
COM handler
CAN
DIO
LIB
vTaskStartScheduler
MDIO_Void_Init
MCAN_void_INIT
HLeftLight_Setup
HRightLight_Setup
HBuzzer_Setup
sd loop
after 5ms
SBCM_Read
HCOMhandler_Read
MCAN_void_ReadByte
after 10ms
SBCM_Read
HCOMhandler_Read
MCAN_void_ReadByte
after 20ms
SBCM_Read
HCOMhandler_Read
MCAN_void_ReadByte
Setter
Turns on buzzer, left right or light right based on the data read from ECU1

CPU load for the ECU2:

Calculating CPU load for ECU2:

Task 1 (Receive speed sensor data): P:5, E:1, D:5
Task 2 (Receive door sensor data): P:10, E:1, D:10
Task 3 (Receive light switch data): P:20, E:1, D:20
Task 4 (Setter): P:23, E:2, D:23
Applying the above data on simso: CPU load for ECU 2 is 45%



id	Name	Task type	Abort on miss	Act. Date (ms)	Period (ms)	List of Act. dates (ms)	Deadline (ms)	WCET (ms)	Followed by	priority
1	Read speed	Periodic	☑ Yes	0	5	-	5	1		2
2	Read door	Periodic	☑ Yes	0	10	-	10	1		3
3	Read lightSw	Periodic	☑ Yes	0	20	-	20	1		1
4	Setter	Periodic	☑ Yes	0	23		23	2		4

Observation Window:
from 0.00 to 100.00 ms

	Total load	Payload	System load
CPU 1	0.4500	0.4500	0.0000