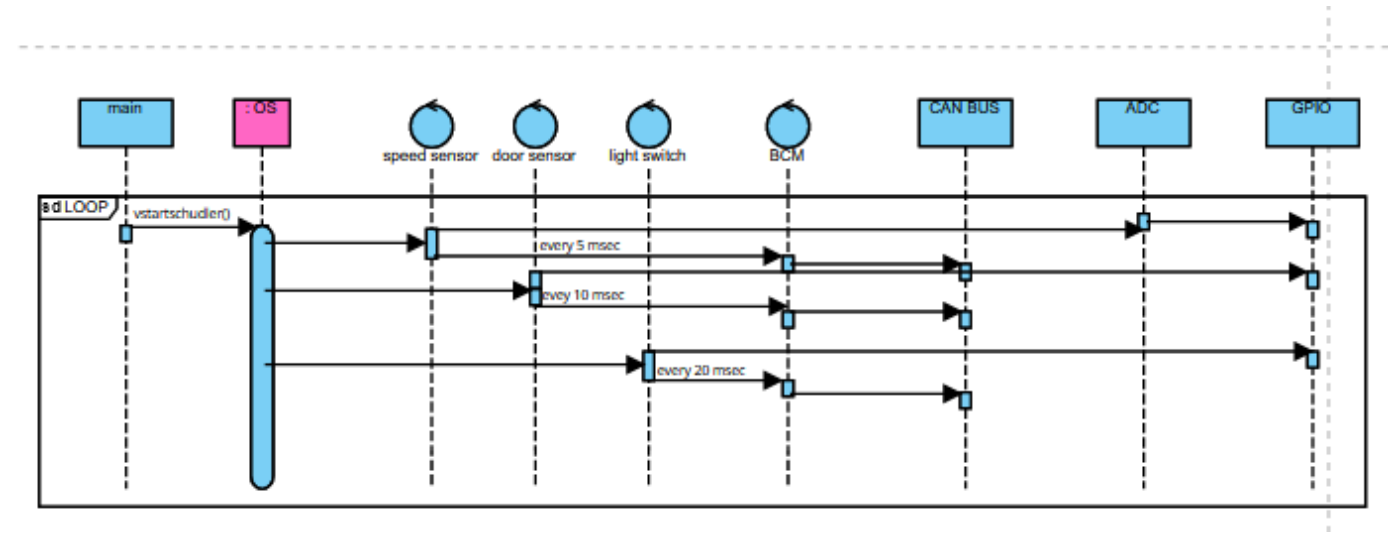
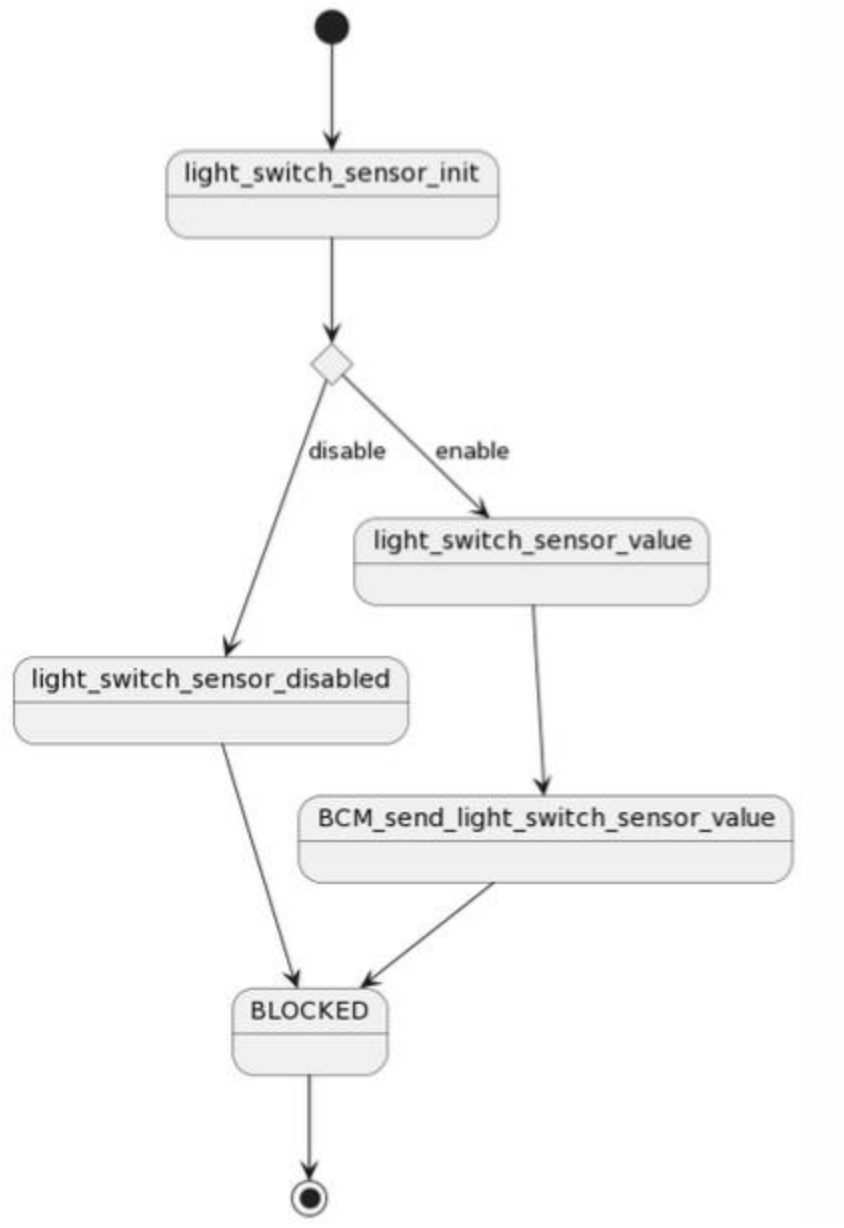
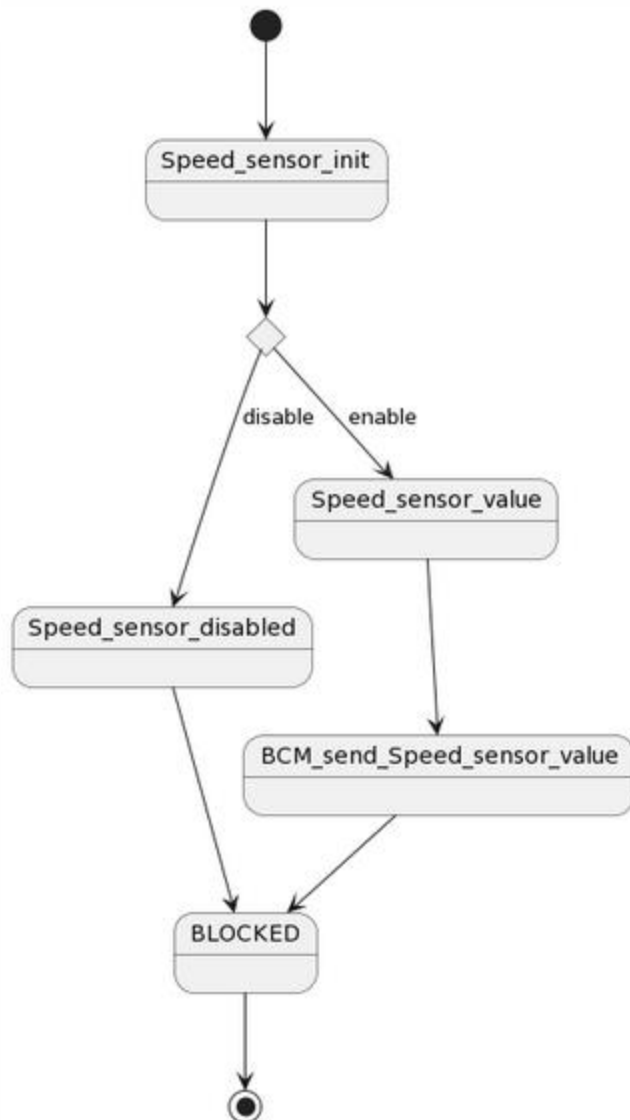
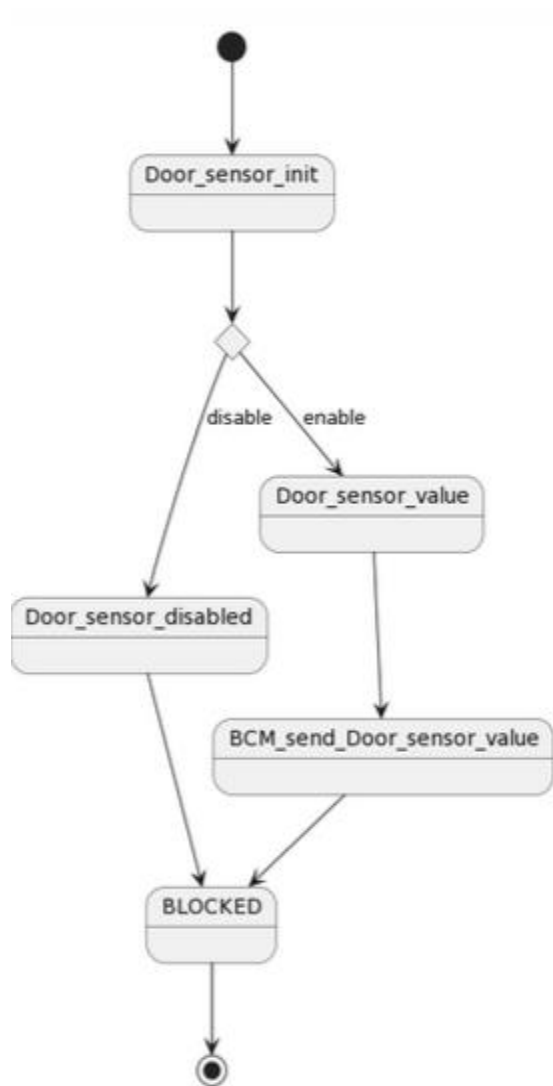


Sequence diagram ECU1:



STATE MACHINE DIGRAM ECU1:





Code example:

```

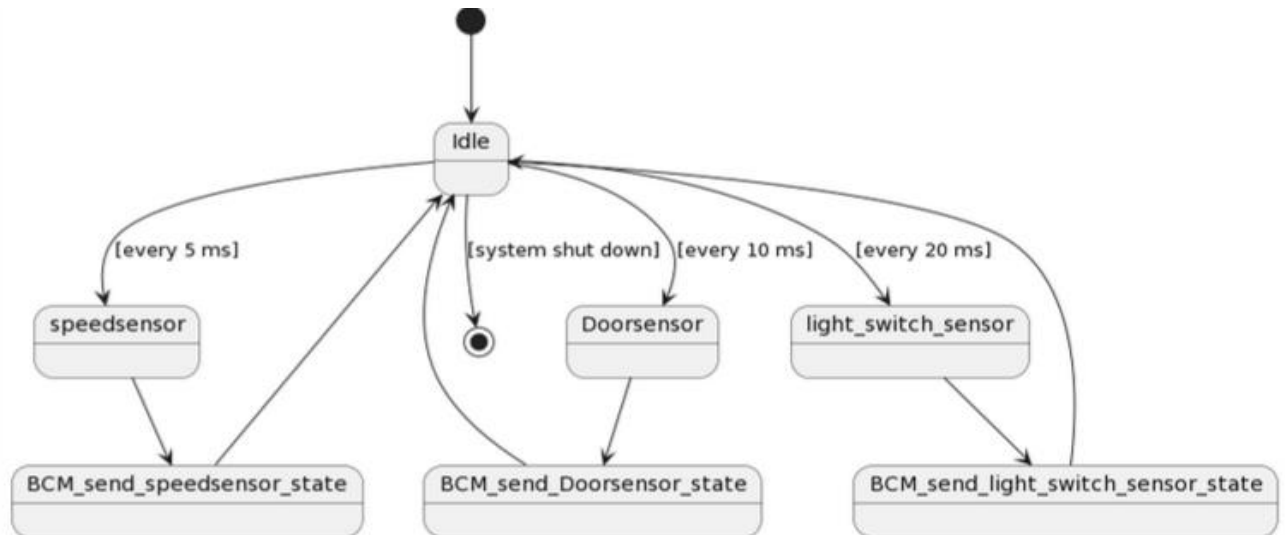
@startuml
[*]-->lights_init
state c <<choice>>
lights_init-->BCM_get_light_switch_sensor_value
BCM_get_light_switch_sensor_value-->c
c-->light_switch_sensor_disabled: disable
c-->light_switch_sensor_value: enable
light_switch_sensor_value-->BCM_send_light_switch_sensor_value
  
```

BCM_send_light_switch_sensor_value-->BLOCKED

light_switch_sensor_disabled-->BLOCKED

BLOCKED-->[*]

@enduml



Code:

@startuml

[*]-->Idle

Idle --> speedsensor: [every 5 ms]

Idle --> Doorsensor: [every 10 ms]

Idle --> light_switch_sensor: [every 20 ms]

speedsensor--> BCM_send_speedsensor_state

Doorsensor--> BCM_send_Doorsensor_state

light_switch_sensor--> BCM_send_light_switch_sensor_state

BCM_send_speedsensor_state-->Idle

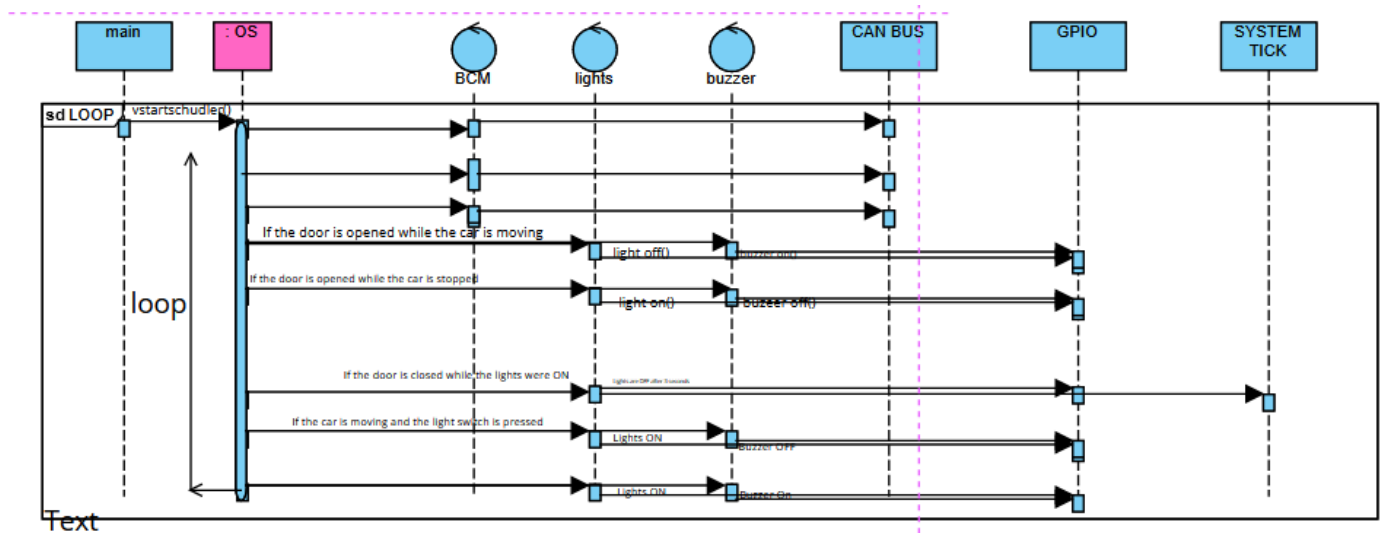
BCM_send_Doorsensor_state-->Idle

BCM_send_light_switch_sensor_state-->Idle

Idle-->[*]: [system shut down]

@enduml

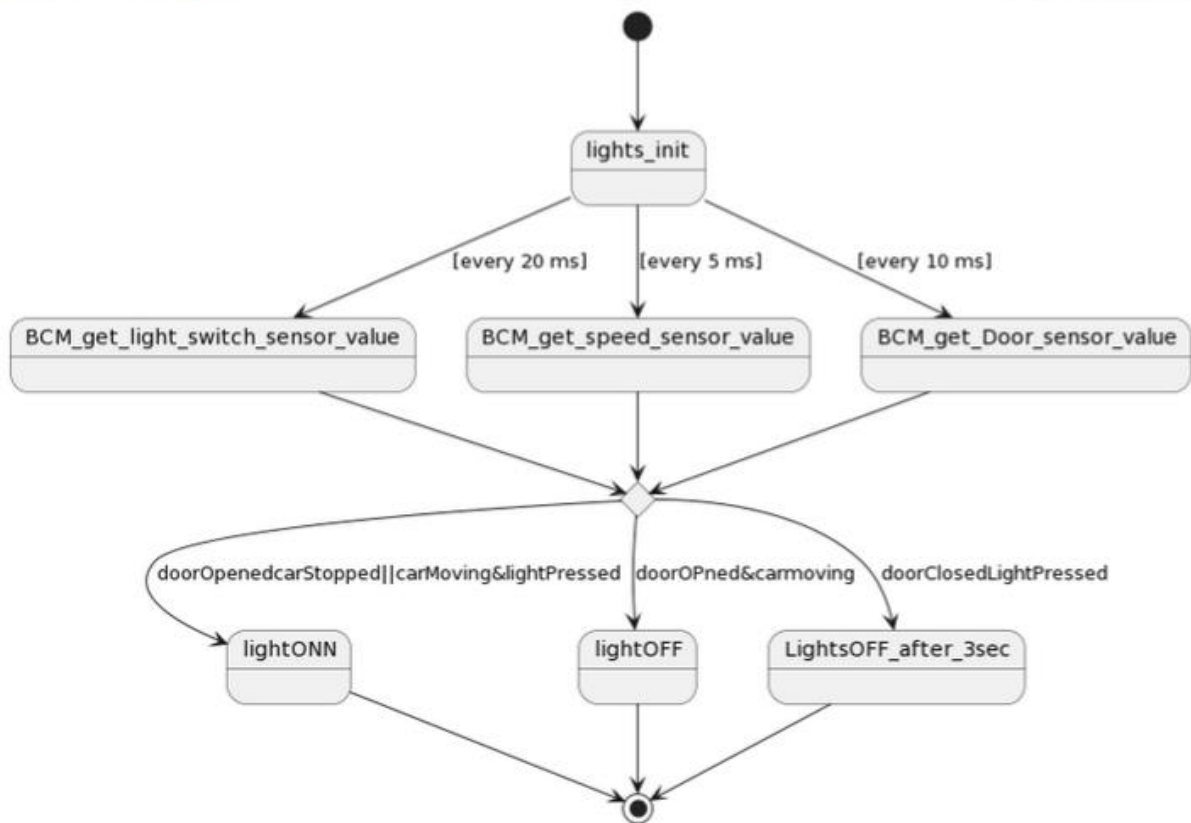
ECU2:

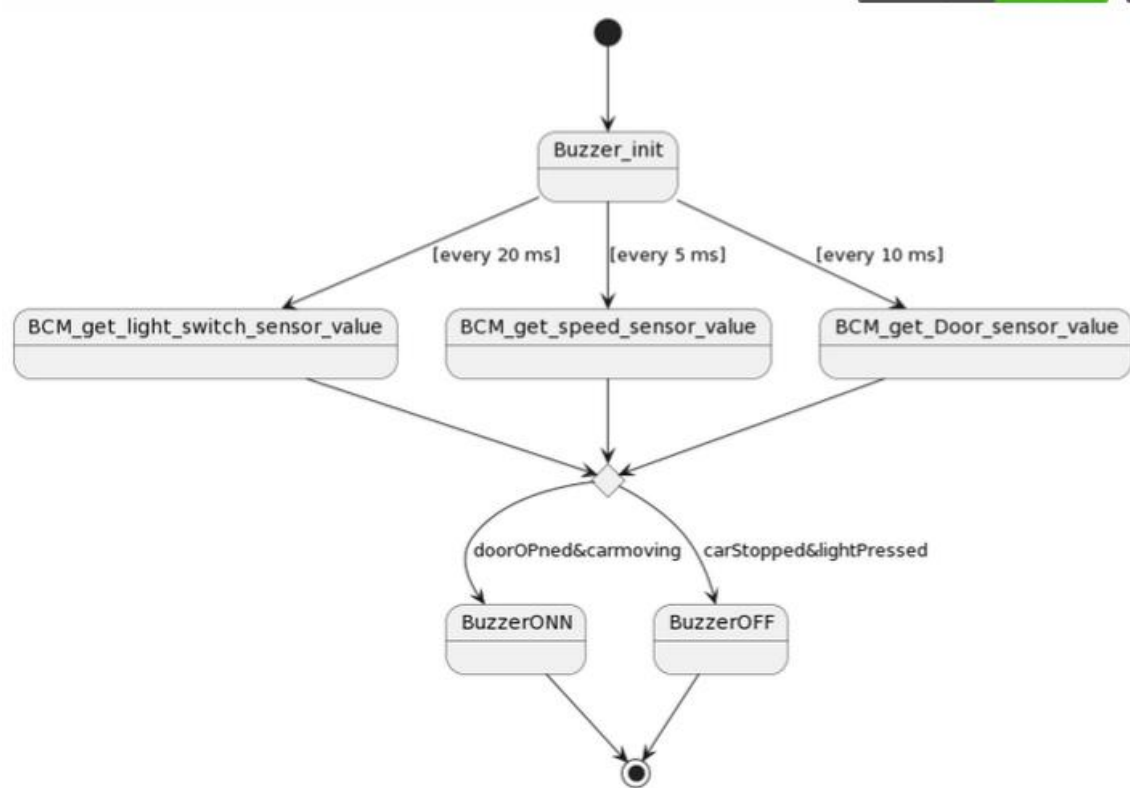


State machine ECU2:

[PNG](#) [SVG](#) [ASCII](#) [Art](#)

online diagrams 183,816,867 current





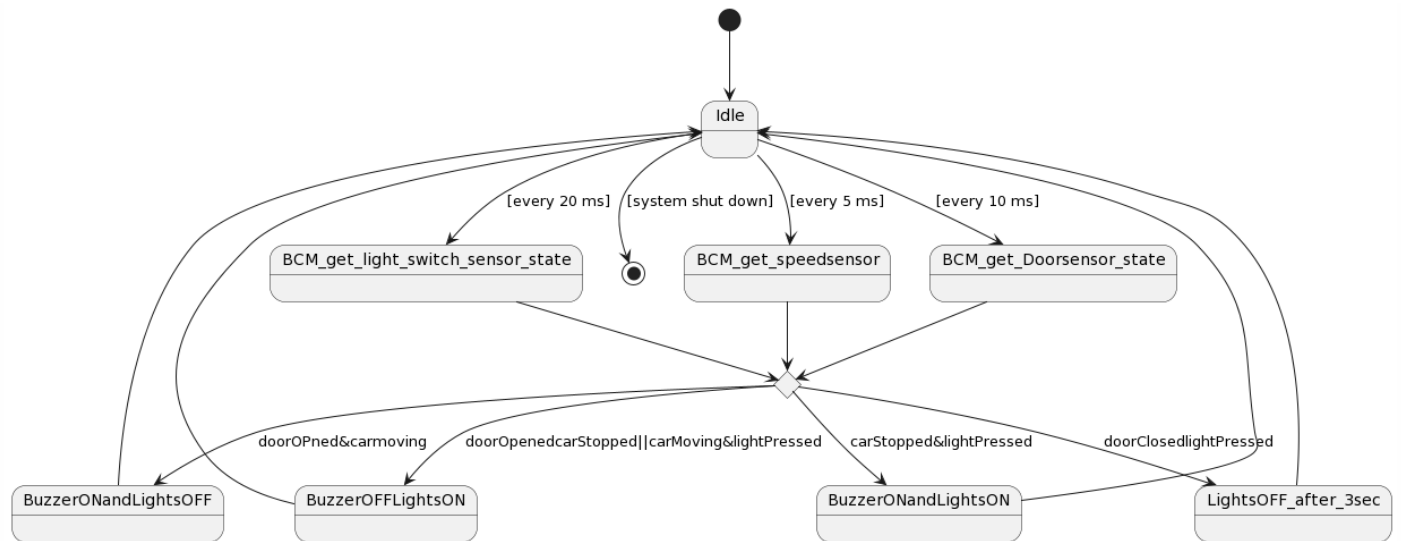
Code example:

```

@startuml
[*]-->lights_init
state c <<choice>>
lights_init-->BCM_get_light_switch_sensor_value
lights_init-->BCM_get_speed_sensor_value
lights_init-->BCM_get_Door_sensor_value
BCM_get_light_switch_sensor_value-->c
BCM_get_speed_sensor_value-->c
BCM_get_Door_sensor_value-->c
c-->lightONN: doorOpenedcarStopped | | carMoving&lightPressed
c-->lightOFF: doorOPned&carmoving
c --> LightsOFF_after_3sec: doorClosedLightPressed
lightONN-->[*]
  
```

lightOFF-->[*]

LightsOFF_after_3sec-->[*] @enduml



@startuml

[*]-->Idle

Idle --> BCM_get_speedsensor:[every 5 ms]

Idle -->BCM_get_Doorsensor_state: [every 10 ms]

Idle -->BCM_get_light_switch_sensor_state: [every 20 ms]

state c <<choice>>

BCM_get_speedsensor--> c

BCM_get_Doorsensor_state--> c

BCM_get_light_switch_sensor_state--> c

c --> BuzzerONandLightsOFF:doorOPned&carmoving

c --> BuzzerOFFLightsON:doorOpenedcarStopped||carMoving&lightPressed

c --> BuzzerONandLightsON: carStopped&lightPressed

c --> LightsOFF_after_3sec: doorClosedlightPressed

BuzzerONandLightsOFF-->Idle

BuzzerOFFLightsON-->Idle

BuzzerONandLightsON-->Idle

LightsOFF_after_3sec-->Idle

Idle-->[*]: [system shut down]@enduml

Using simso

Choosing EDF scheduler

Assuming dead line =periodicity

Assuming execution time of each task equal 1 ms

id	Name	Task type	Abort on miss	Act. Date (ms)	Period (ms)	List of Act. dates (ms)	Deadline (ms)	WCET (ms)
1	TASK T1	Periodic	<input checked="" type="checkbox"/> Yes	0	10	-	10	1
2	TASK T2	Periodic	<input checked="" type="checkbox"/> Yes	0	5	-	5	1
3	TASK T3	Periodic	<input checked="" type="checkbox"/> Yes	0	20	-	20	1

	Total load	Payload	System load
CPU 1	0.3500	0.3500	0.0000
Average	0.3500	0.3500	0.0000

Bus load:

Assume buad rate = 1mbps

Clock rate = 16 MHz

Number of bits

1 bit SOF

11 bit messages identifier

1 bit RTR

6 bit control field

8 bit data field

15 bit CRC sequence

1 bit ACK

2 BIT delima

7 bit EOP

3 bit IFS

Number of bit in a single msg is 55 bit

Bit time = $1/1000,000 = 1\mu s$

Time to transfer one frame = $55 \times 1 = 55\mu s$

Making hyper period = 40ms

Task1 one frame transfer every 5 ms so frame will transfer 8 times

Task2 frame transfer every 10 ms so frame will transfer 4 time

Task 3 frame transfer every 20 ms so frame will transfer 2 time

Total Execution time = $(8 \times 55\mu s + 4 \times 55\mu s + 2 \times 55\mu s) = 770\mu s$

bus load = Total Execution time / hyper period

bus load = $770 \times 10^{-6} / 40 \times 10^{-3} = 0.19\%$