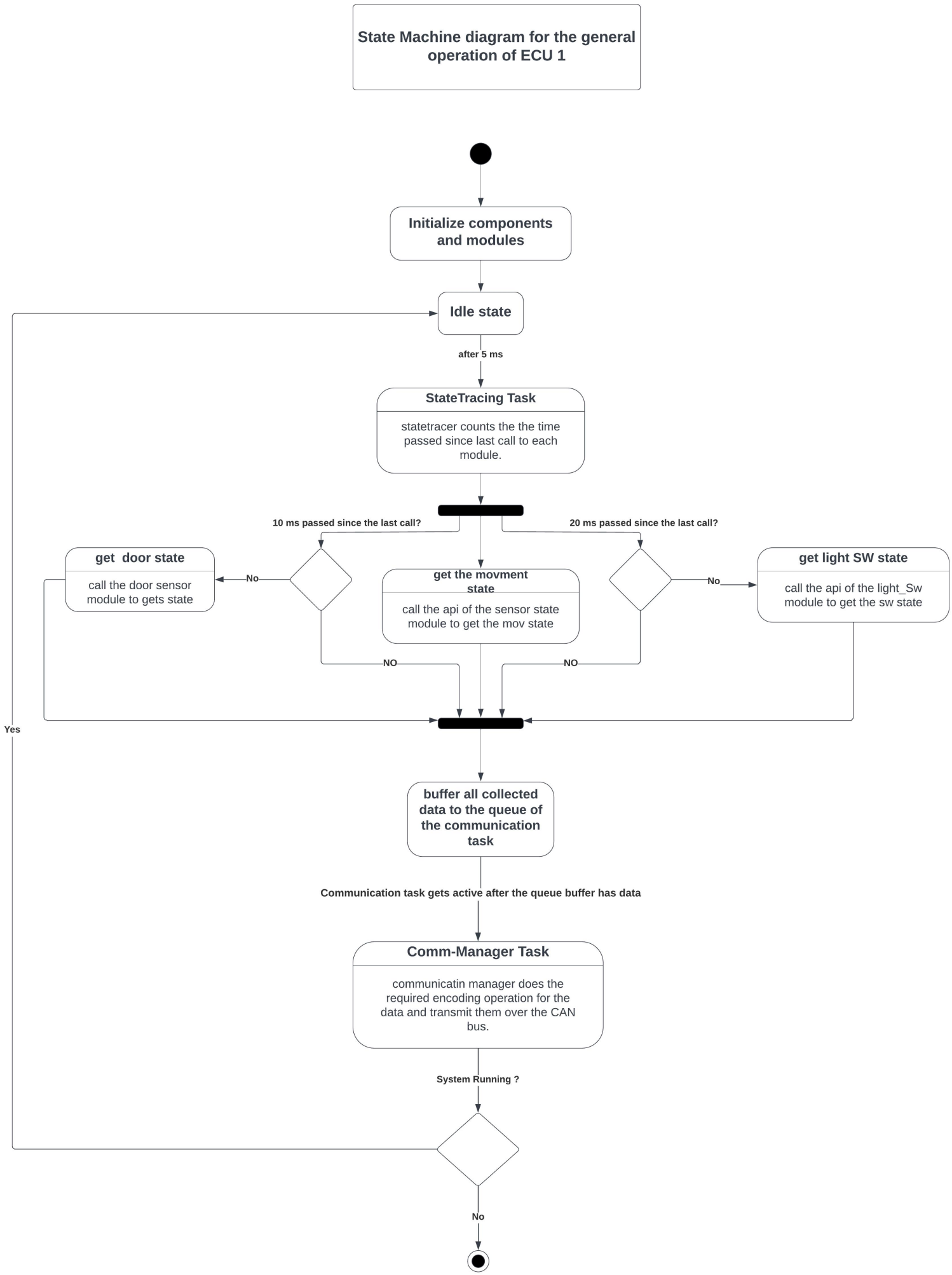
Automotive door control system

Dynamic Design

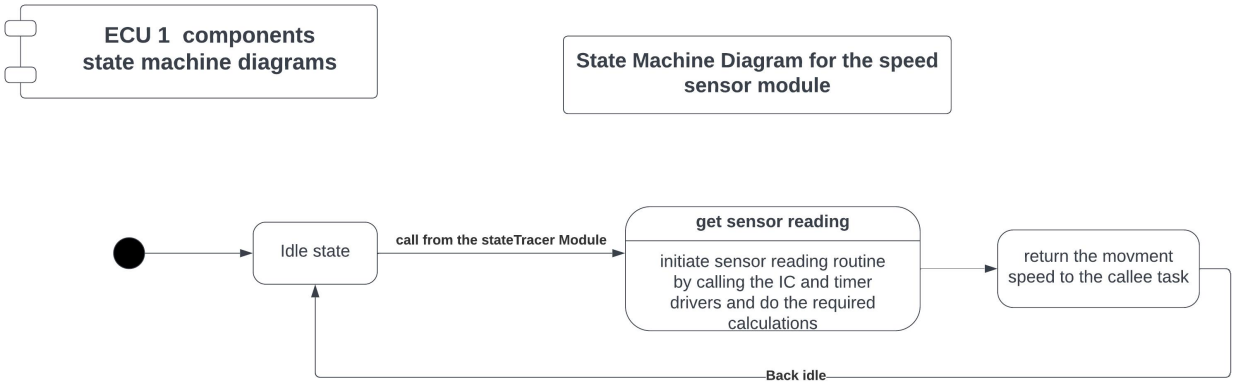
In this paper we illustrate the dynamic design of the automotive door control system, that includes the state machine diagrams for each component in the the system as well as the state machine diagrams for the full operation of both of ECUs, Sequence diagrams and CPU loads calculations.

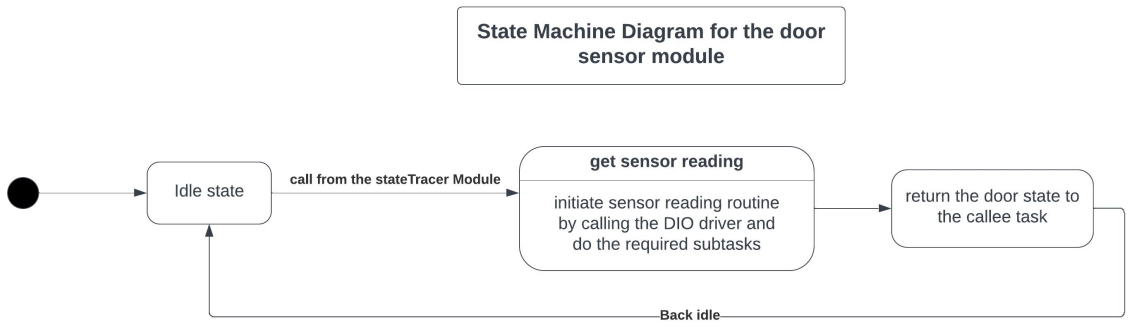
ECU 1

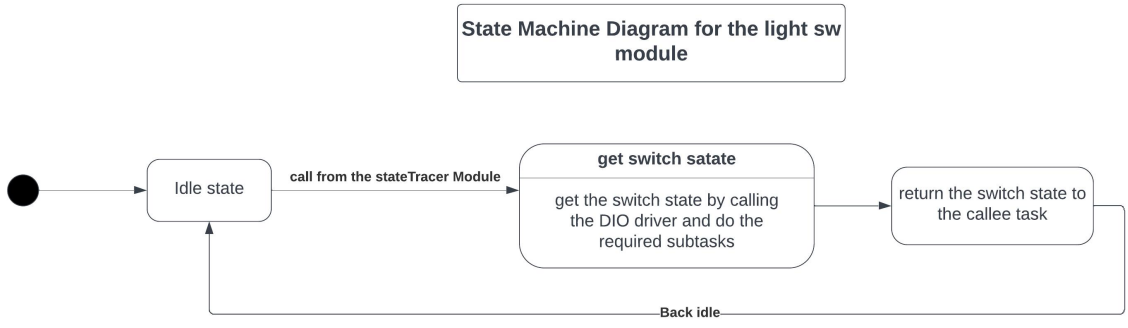
1. State machine diagram for full operation of ECU 1

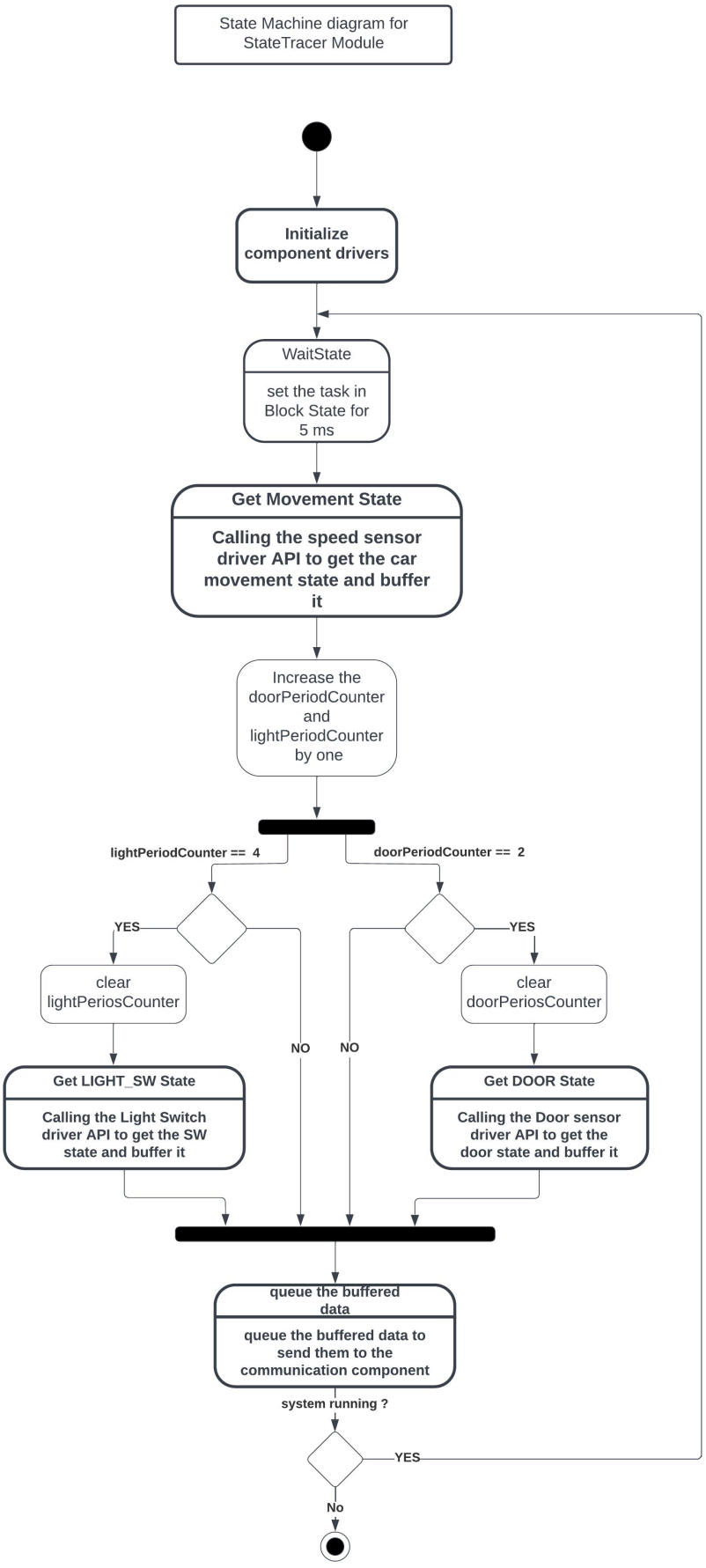

State Machine diagram for the general operation of ECU 1
Initialize components and modules
Idle state
after 5 ms
StateTracing Task
statetracer counts the the time passed since last call to each module.
10 ms passed since the last call?
20 ms passed since the last call?
get door state
call the door sensor module to gets state
No
get the movment state
call the api of the sensor state module to get the mov state
No
get light SW state
call the api of the light_Sw module to get the sw state
NO
NO
Yes
buffer all collected data to the queue of the communication task
Communication task gets active after the queue buffer has data
Comm-Manager Task
communicatin manager does the required encoding operation for the data and transmit them over the CAN bus.
System Running ?
No

2. State machine diagrams for each component in the system.

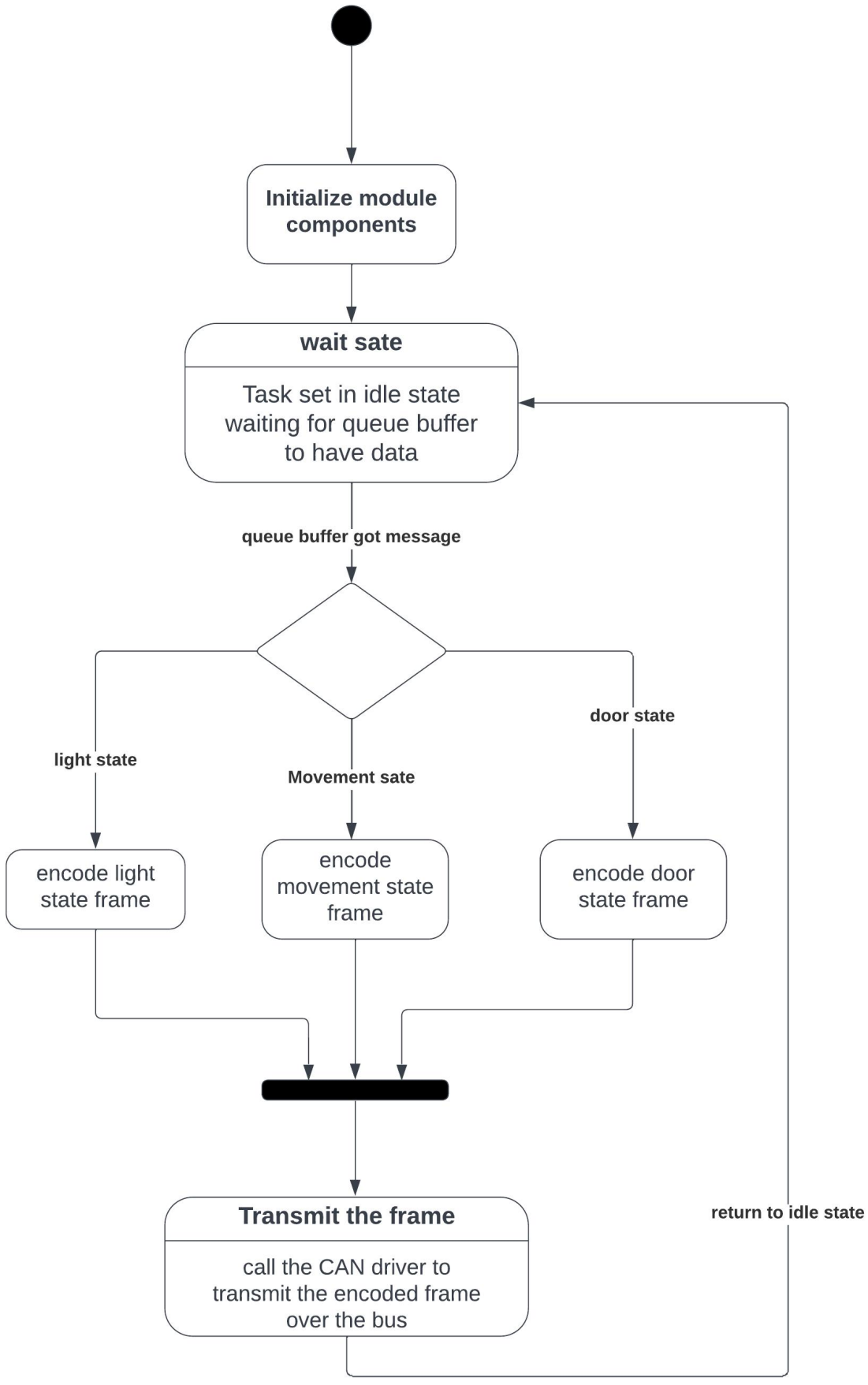




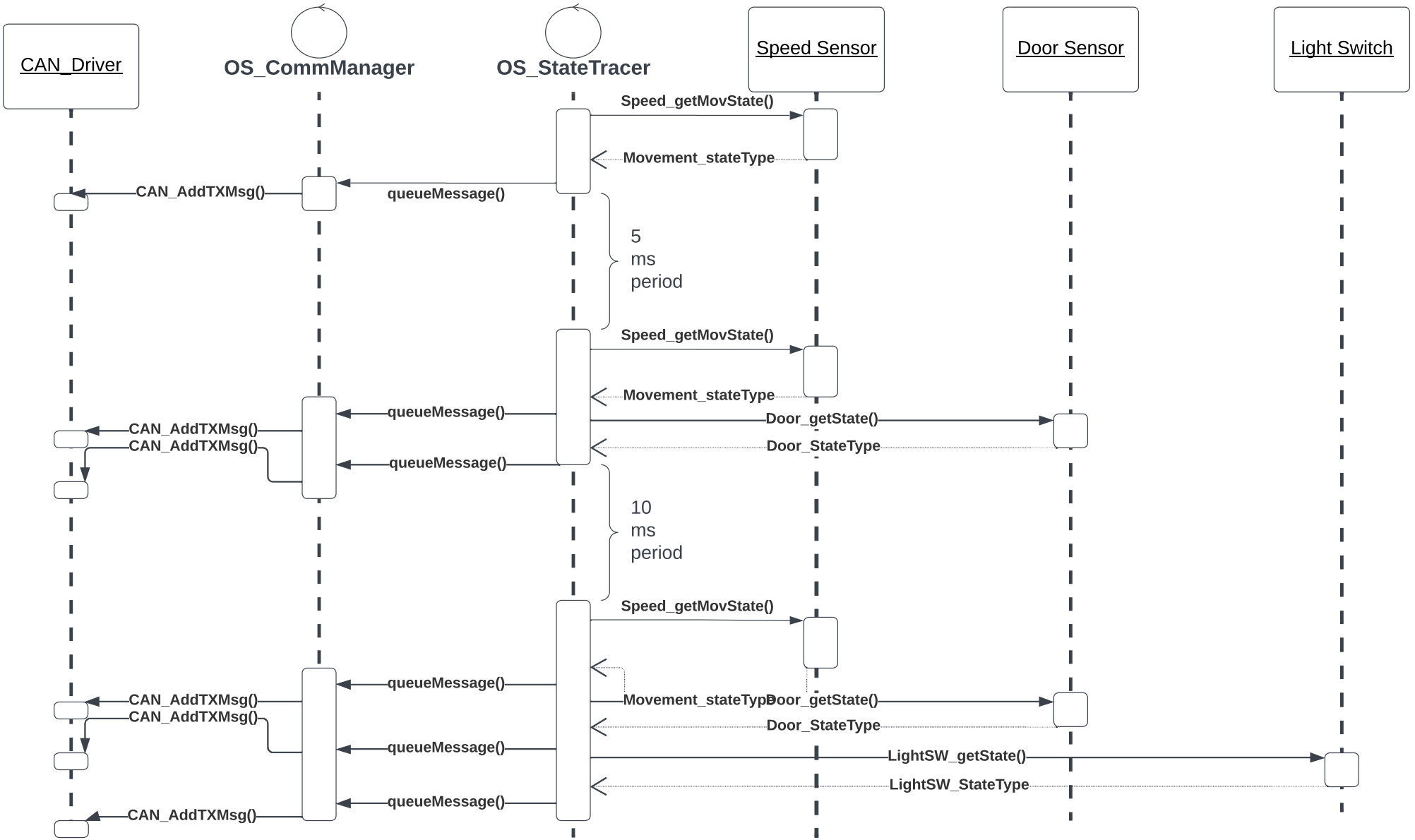



State Machine diagram for StateTracer Module
Initialize component drivers
WaitState
set the task in Block State for 5 ms
Get Movement State
Calling the speed sensor driver API to get the car movement state and buffer it
Increase the doorPeriodCounter and lightPeriodCounter by one
lightPeriodCounter == 4
doorPeriodCounter == 2
YES
YES
clear lightPeriosCounter
clear doorPeriosCounter
NO
NO
Get LIGHT_SW State
Calling the Light Switch driver API to get the SW state and buffer it
Get DOOR State
Calling the Door sensor driver API to get the door state and buffer it
queue the buffered data
queue the buffered data to send them to the communication component
system running ?
YES
No

State machine diagram for Communication Module

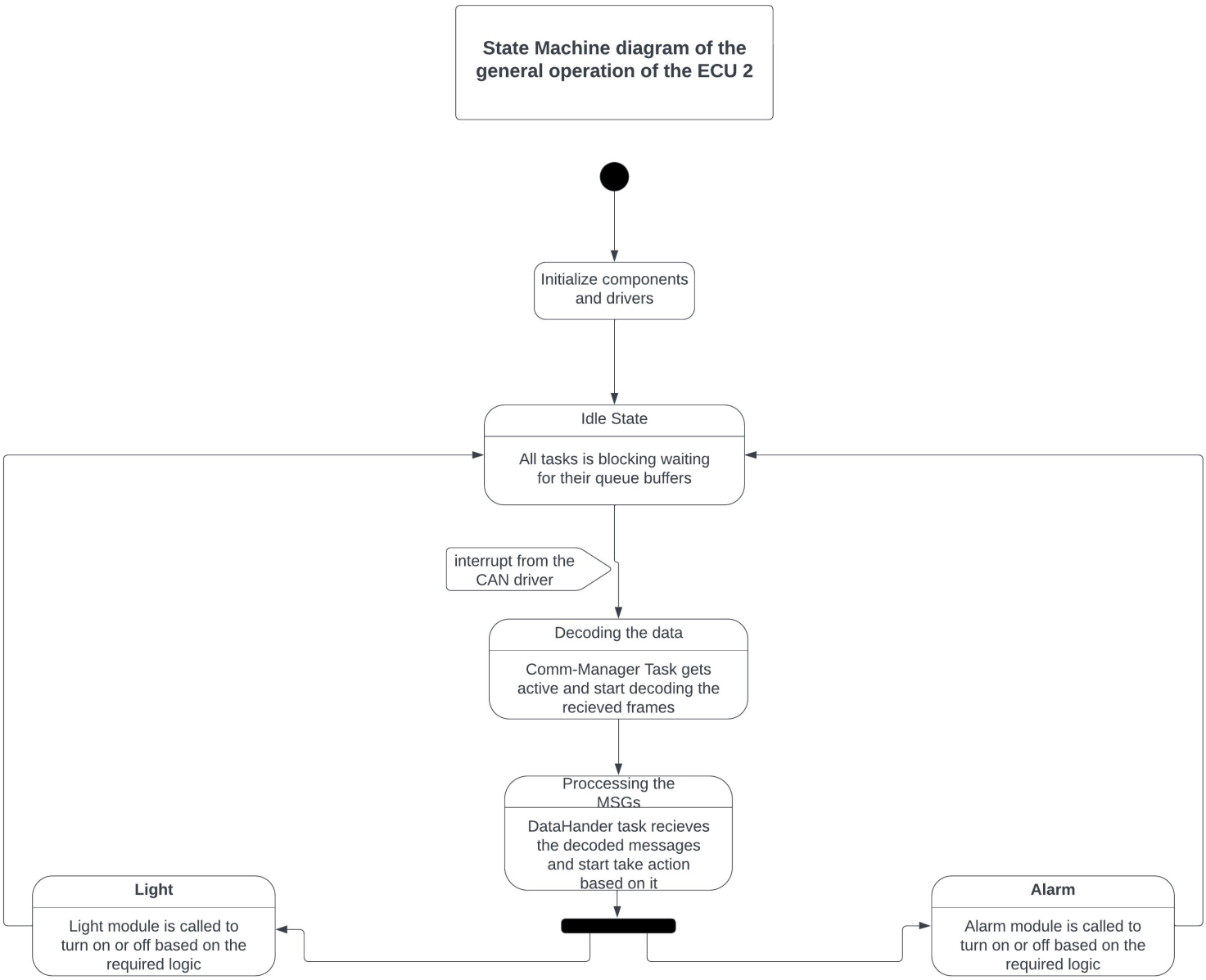


Sequence Diagram for ECU 1

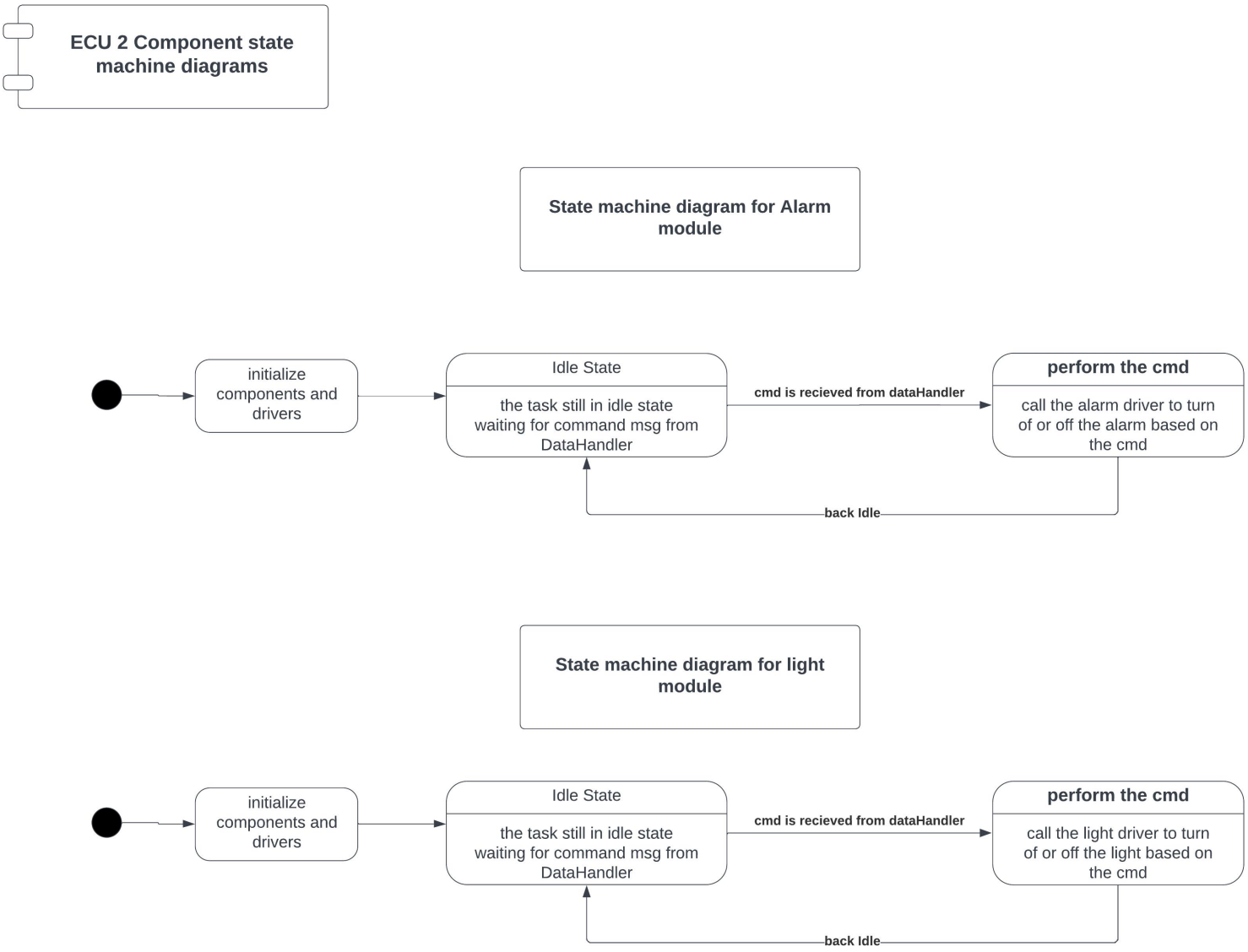


ECU 2

1. State machine diagram for full operation of ECU 2

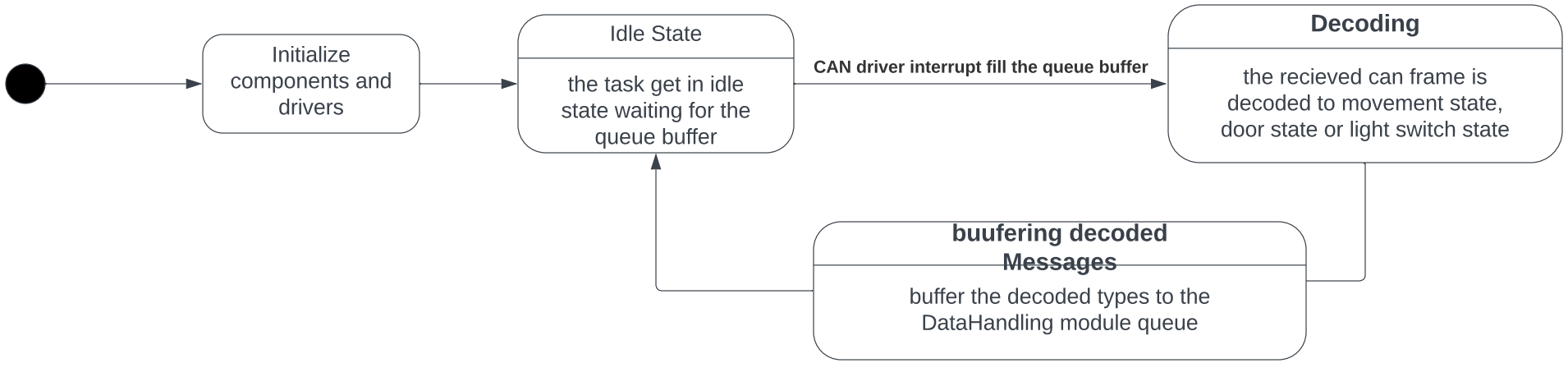


2. State machine diagrams for each component in the system.

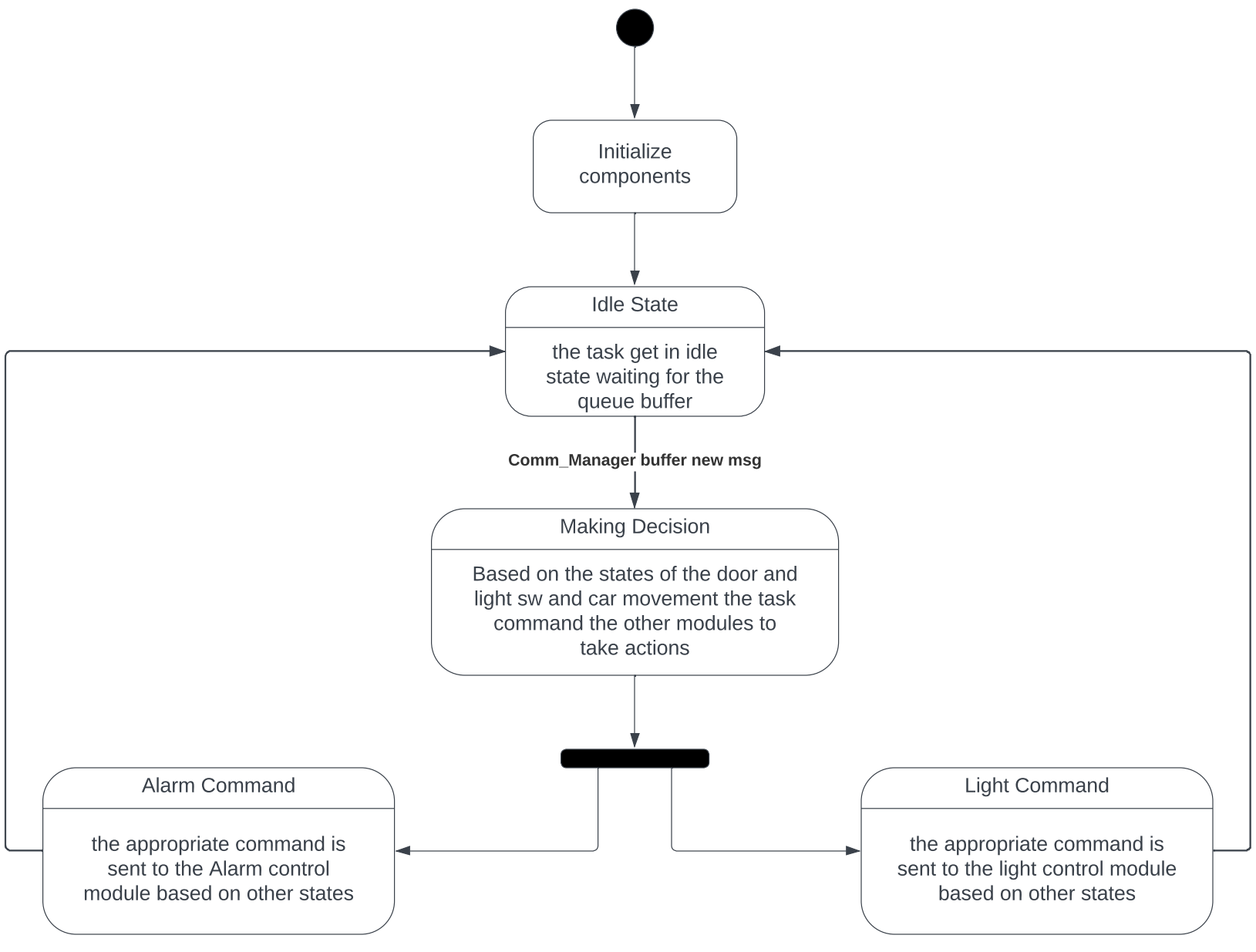


ECU 2 Component state machine diagrams

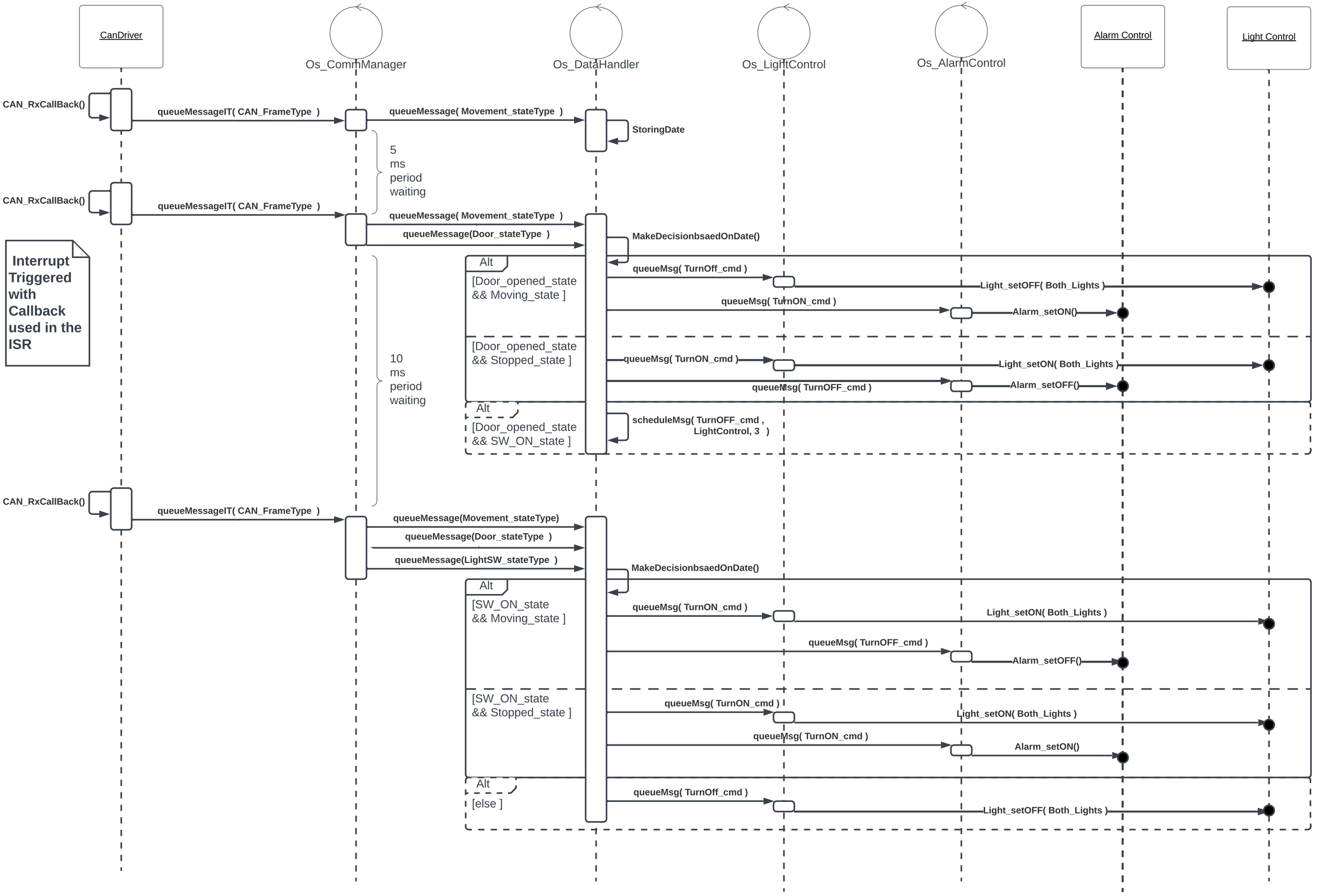
State Machine diagram for the Communcation Manager



State Machine diagram for the Data Handler



Sequence Diagram of ECU 2



■ CPU Load Calculation.

1. For ECU 1

Every 20 ms all tasks are scheduled in the system.
The hyperperiod H for ECU 1 is 20 ms.

If we assume all tasks are equal in execution time and that time is 1 ms.

CPU Load in for ECU 1 (U) = $\frac{E_1 + E_2 + E_3}{H} = \frac{1\times1 + 1\times2 + 1\times4}{20} \times 100$

= 35%

2. For ECU 2

MCU 2 is interrupted every 5 ms by the CAN driver,
so The Communication task is scheduled every 5 ms too,
Hence the the DataHandling task is activated every 5 ms.

The hyperperiod for the system is 5 ms.

If we assume all tasks are equal in execution time and that time is 1 ms.

CPU Load in for ECU 2 (U) = $\frac{E_1 + E_2}{H} = \frac{1+1}{5} \times 100$

= 40%