Nanodegree

Automotive door control design

Ahmed Elebaby

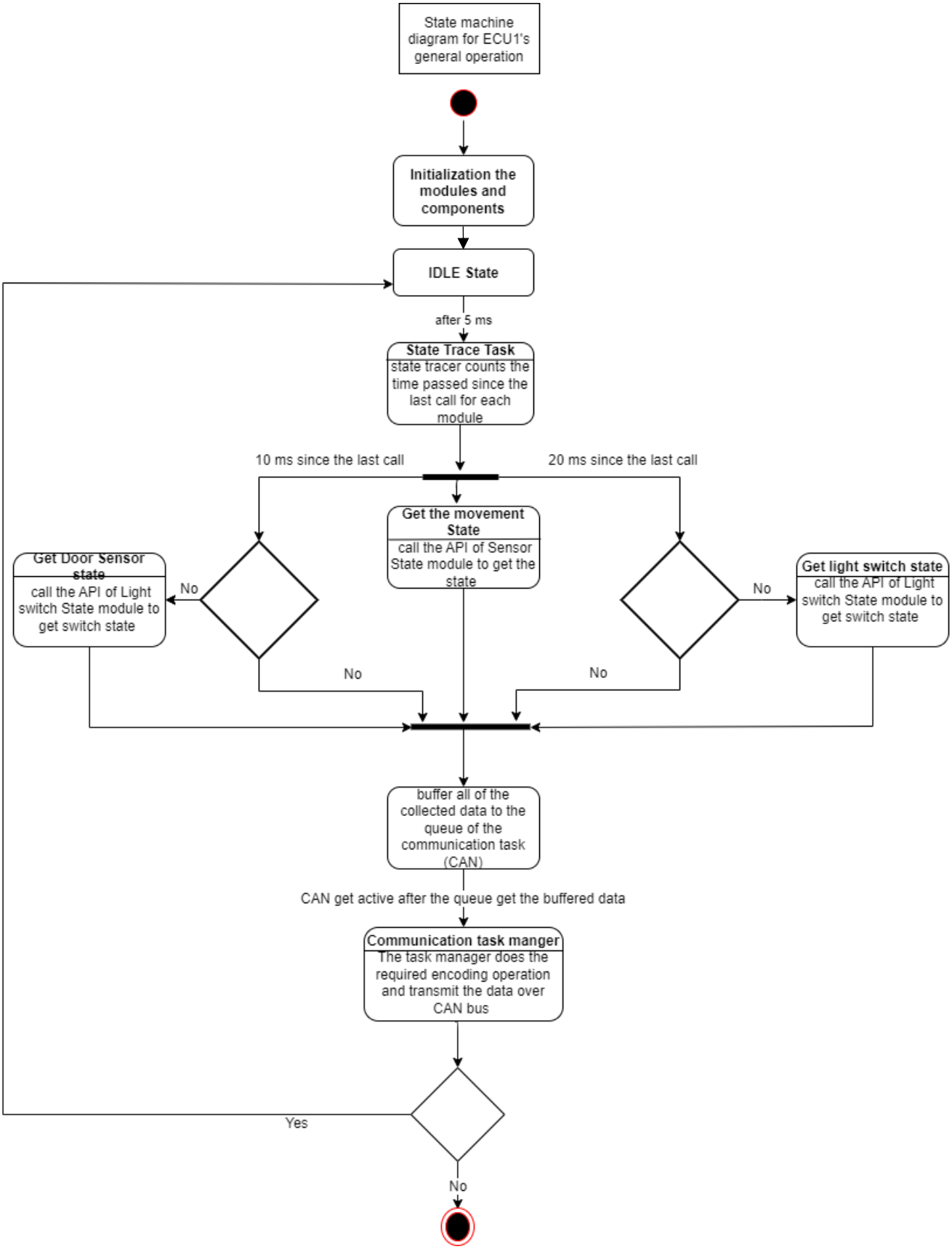
Dynamic design

Introduction:

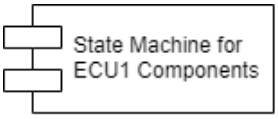
In this document, we present the dynamic design of the automotive door control system. This design features state machine diagrams for each component in the system, as well as complete state machine diagrams for both ECUs. Additionally, we include sequence diagrams and calculations for CPU loads.

ECU 1:

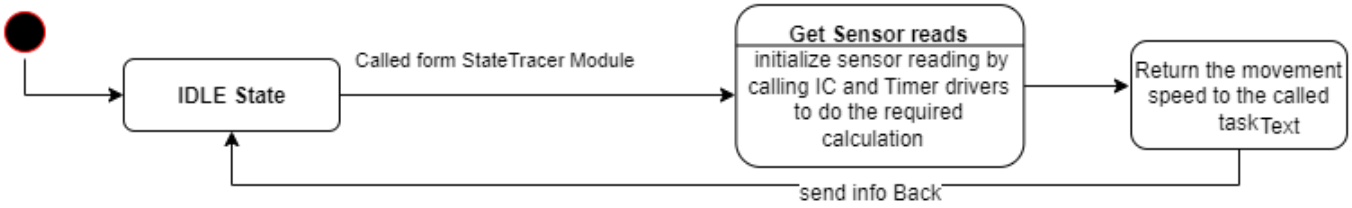
→ State machine diagram for full operation of ECU 1



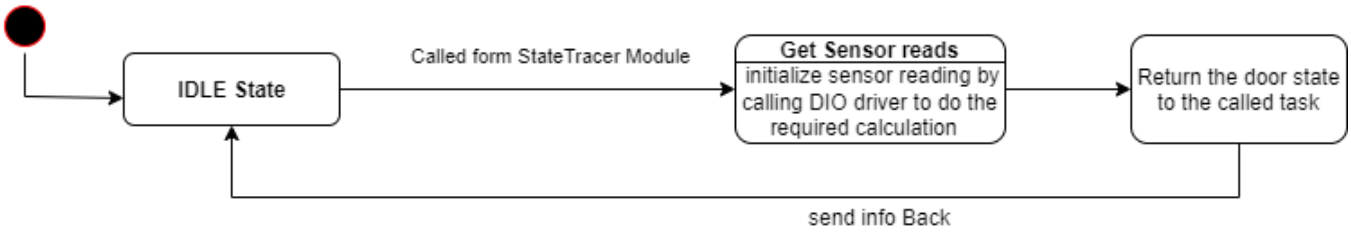
→ State machine diagrams for each component in the system

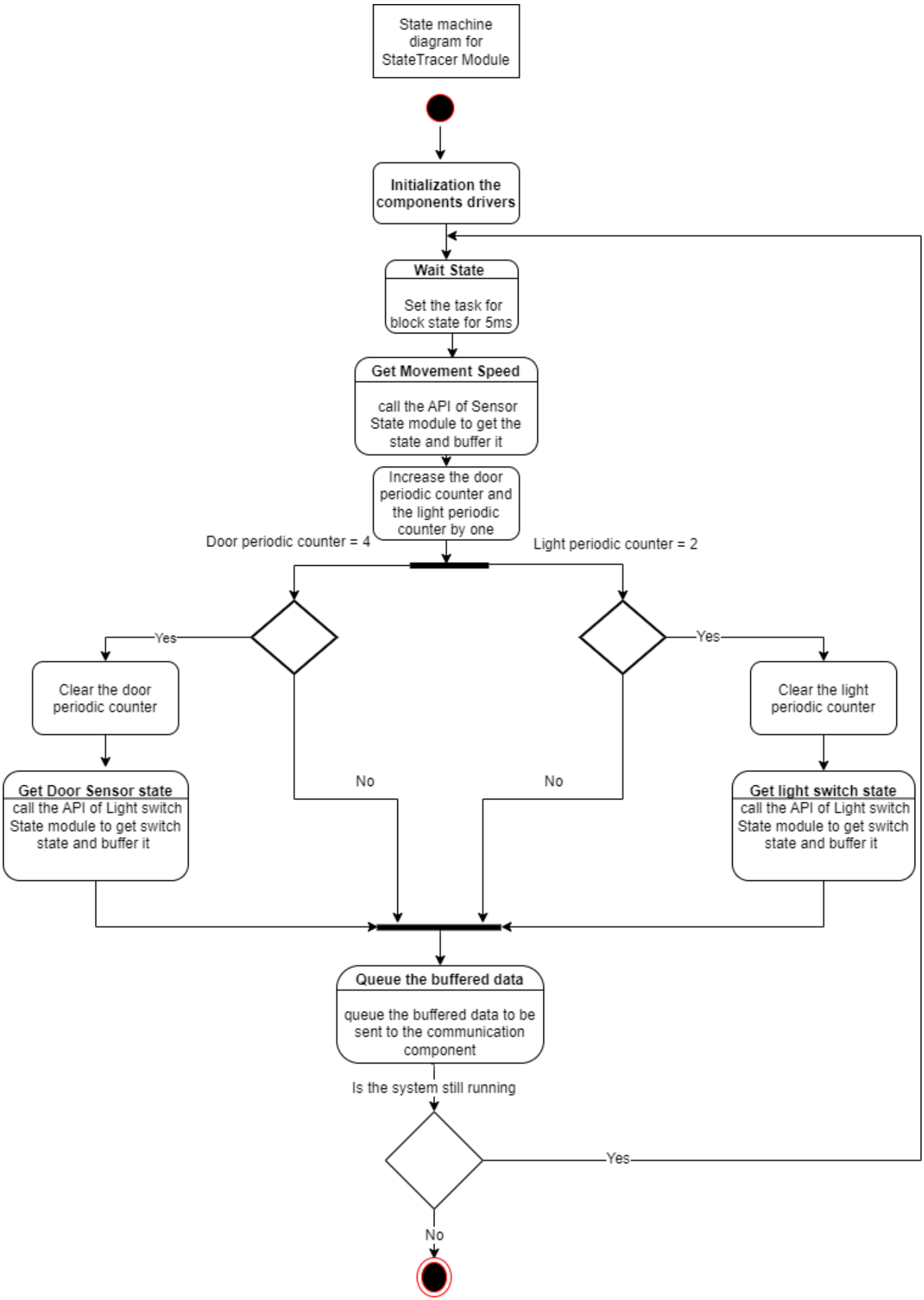


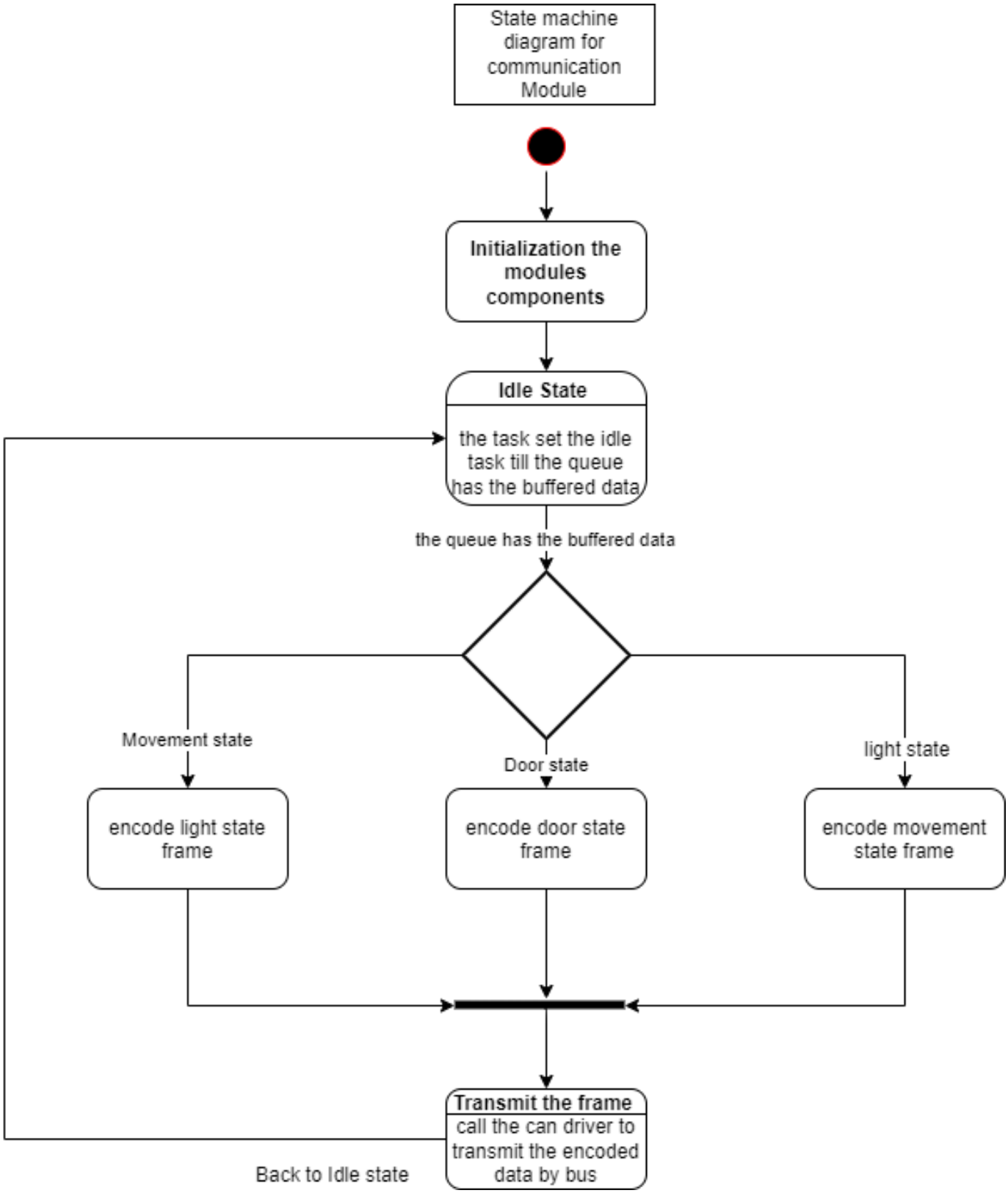
State machine diagram for Speed sensor

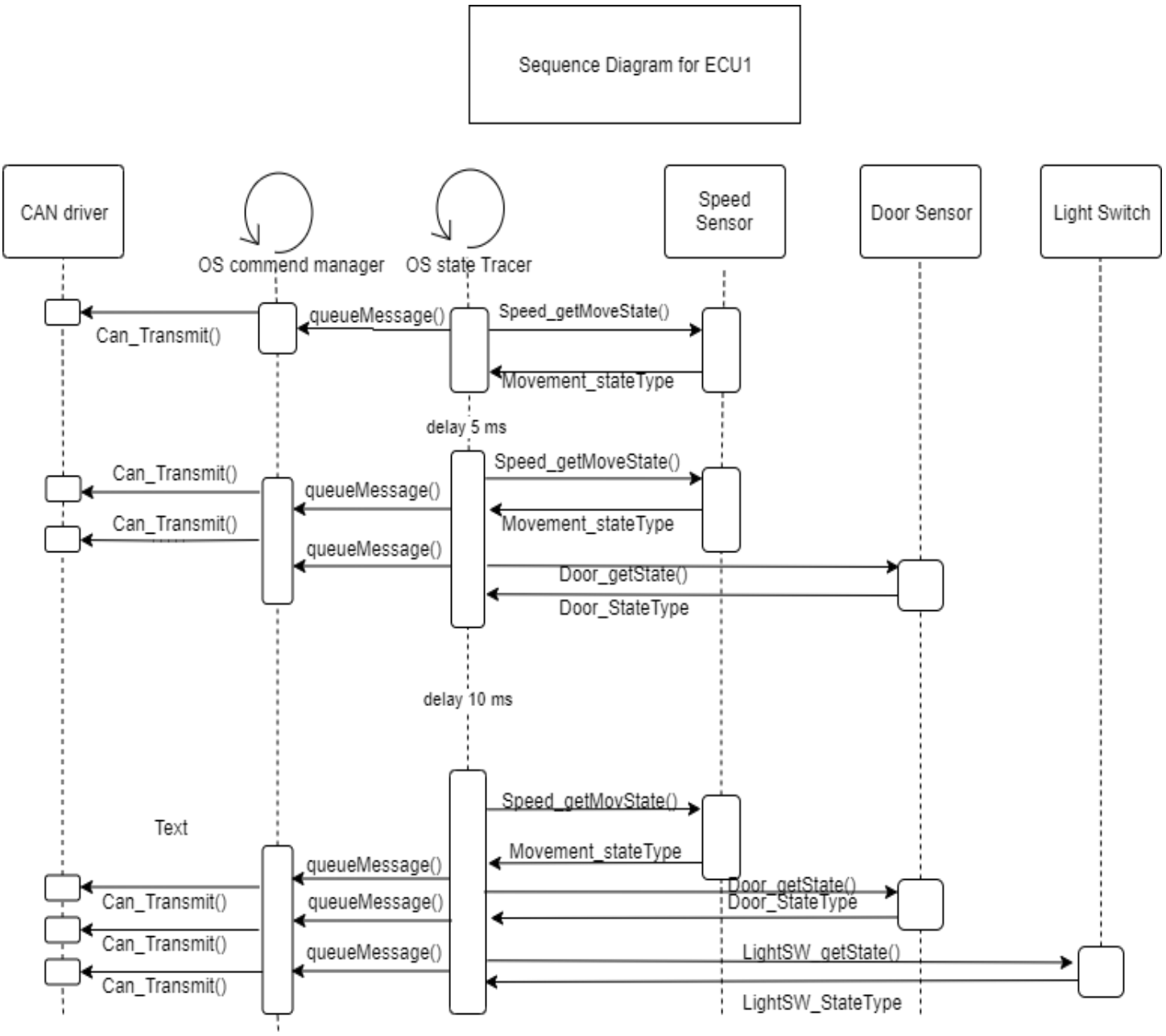


State machine diagram for door sensor



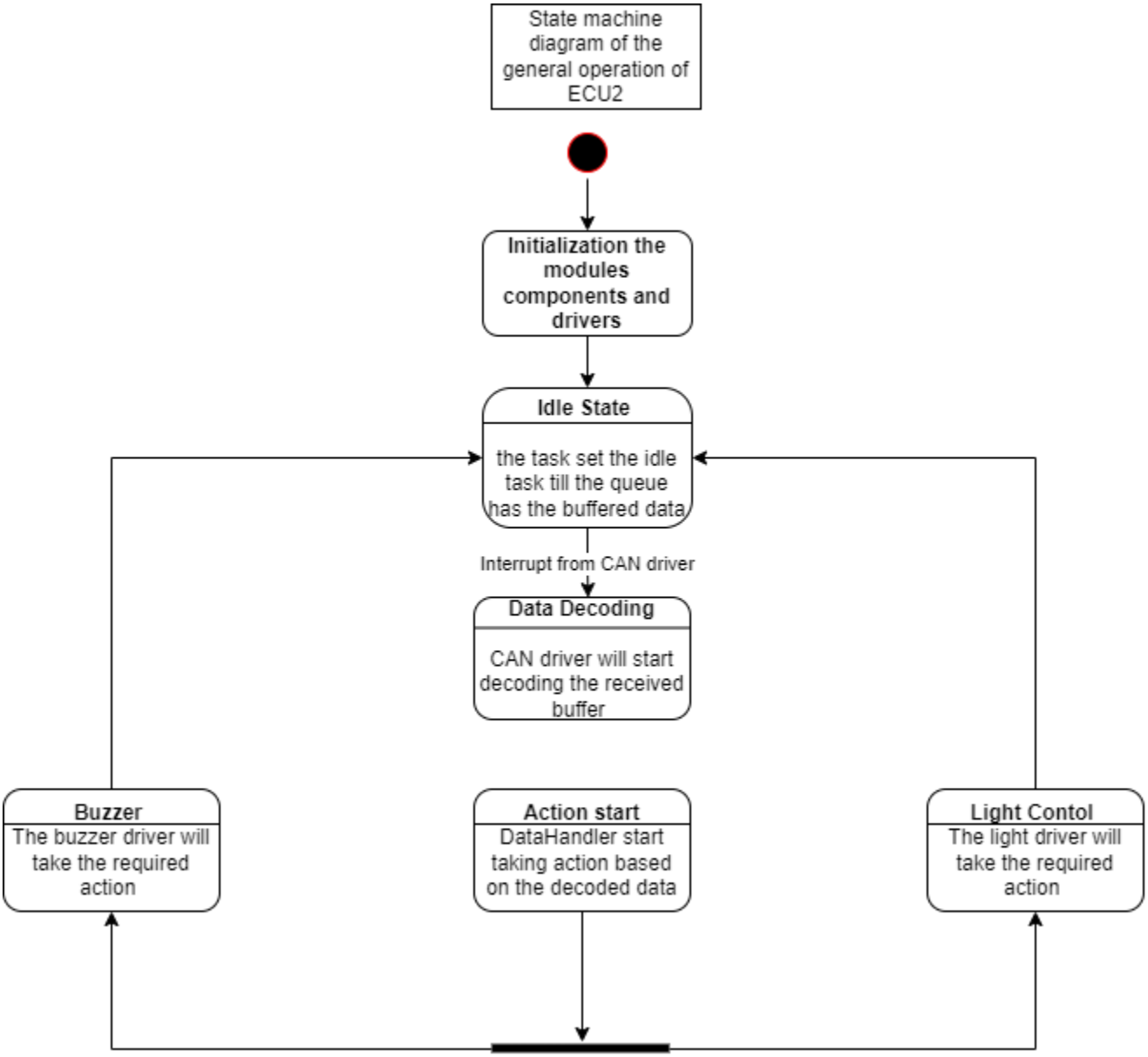

State machine diagram for StateTracer Module
Initialization the components drivers
Wait State
Set the task for block state for 5ms
Get Movement Speed
call the API of Sensor State module to get the state and buffer it
Increase the door periodic counter and the light periodic counter by one
Door periodic counter = 4
Light periodic counter = 2
Yes
Clear the door periodic counter
Get Door Sensor state
call the API of Light switch State module to get switch state and buffer it
No
No
Yes
Clear the light periodic counter
Get light switch state
call the API of Light switch State module to get switch state and buffer it
Queue the buffered data
queue the buffered data to be sent to the communication component
Is the system still running
Yes
No


State machine diagram for communication Module
Initialization the modules components
Idle State
the task set the idle task till the queue has the buffered data
the queue has the buffered data
Movement state
Door state
light state
encode light state frame
encode door state frame
encode movement state frame
Transmit the frame
call the can driver to transmit the encoded data by bus
Back to Idle state

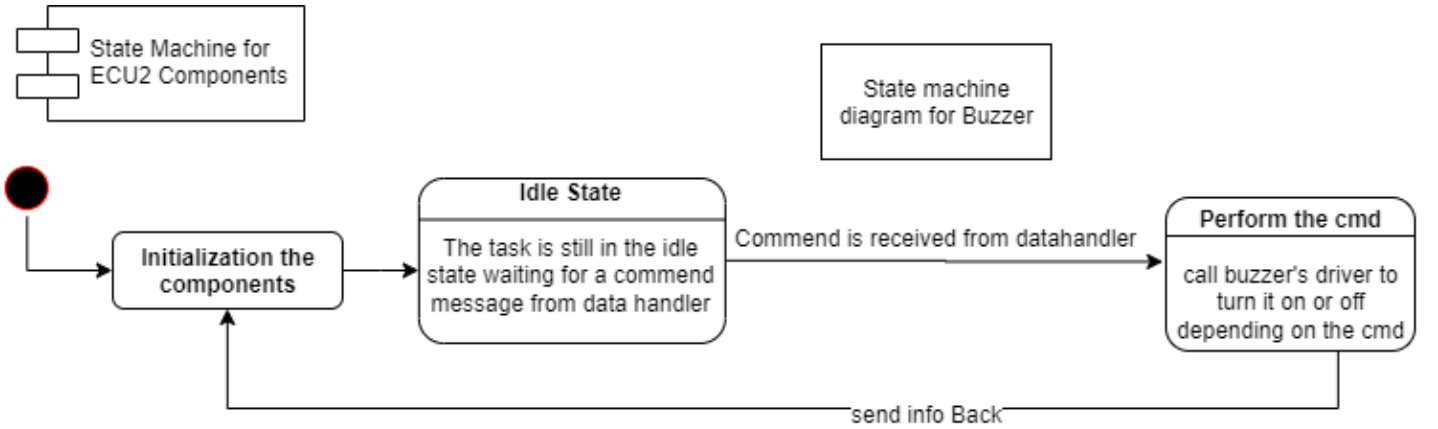

Sequence Diagram for ECU1
CAN driver
OS commend manager
OS state Tracer
Speed Sensor
Door Sensor
Light Switch
Can_Transmit()
queueMessage()
Speed_getMoveState()
Movement_stateType
delay 5 ms
Can_Transmit()
Can_Transmit()
queueMessage()
queueMessage()
Speed_getMoveState()
Movement_stateType
Door_getState()
Door_StateType
delay 10 ms
Text
Speed_getMovState()
Movement_stateType
queueMessage()
queueMessage()
queueMessage()
Door_getState()
Door_StateType
Can_Transmit()
Can_Transmit()
Can_Transmit()
LightSW_getState()
LightSW_StateType

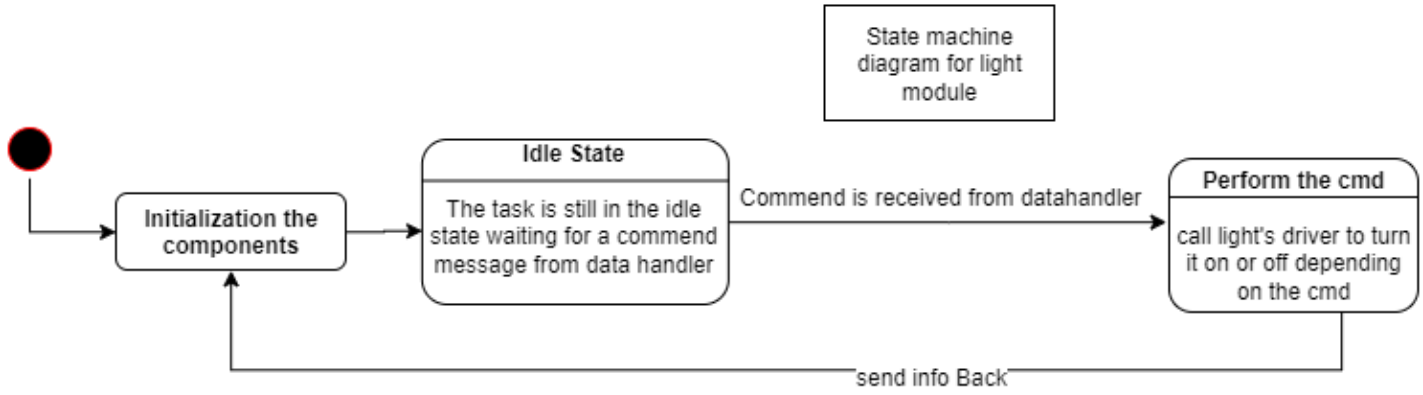
ECU 2:

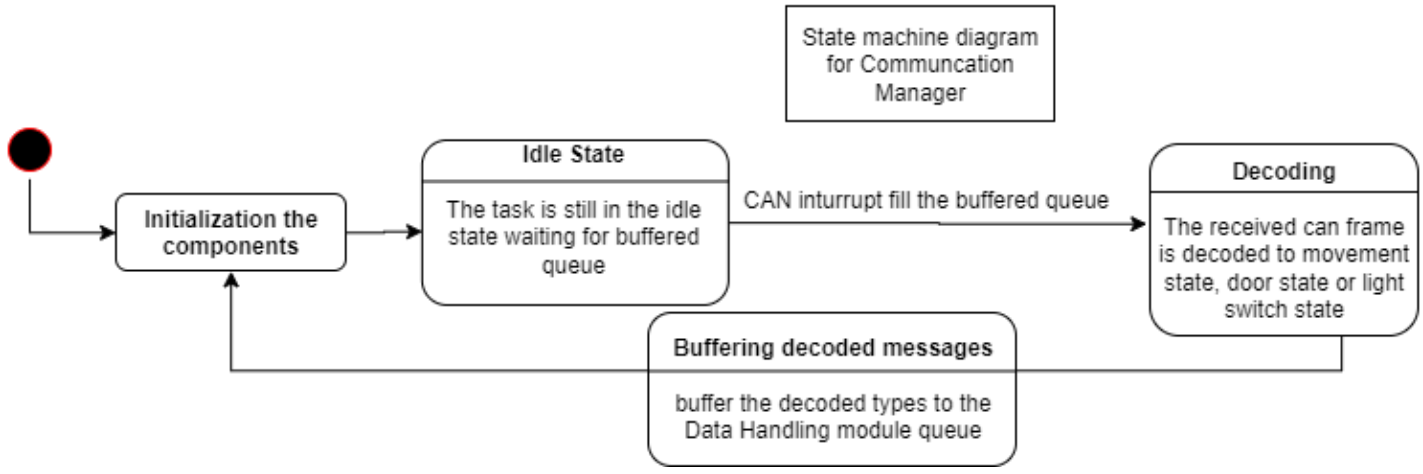
→ State machine diagram for full operation of ECU 2

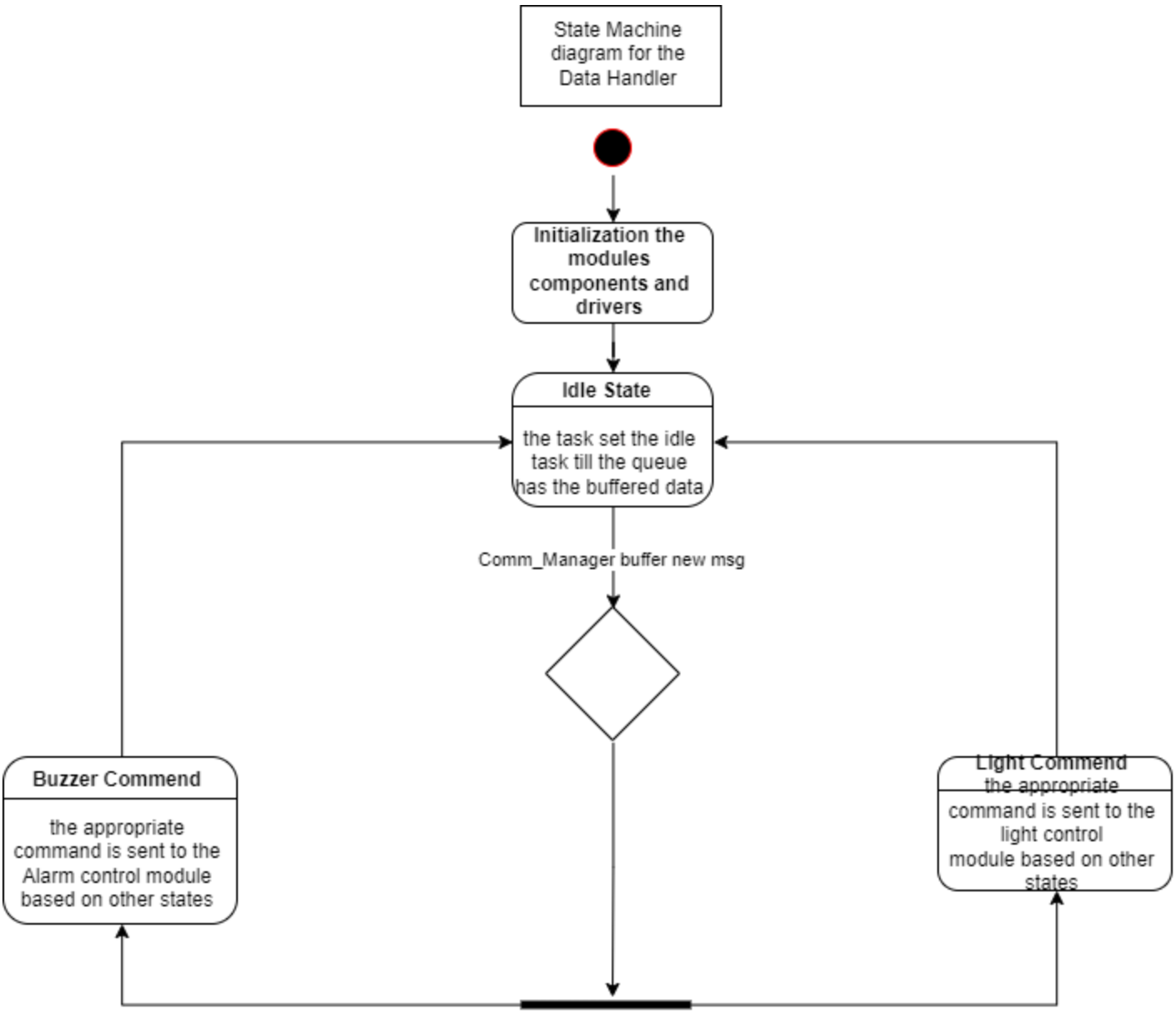


→ State machine diagrams for each component in the system

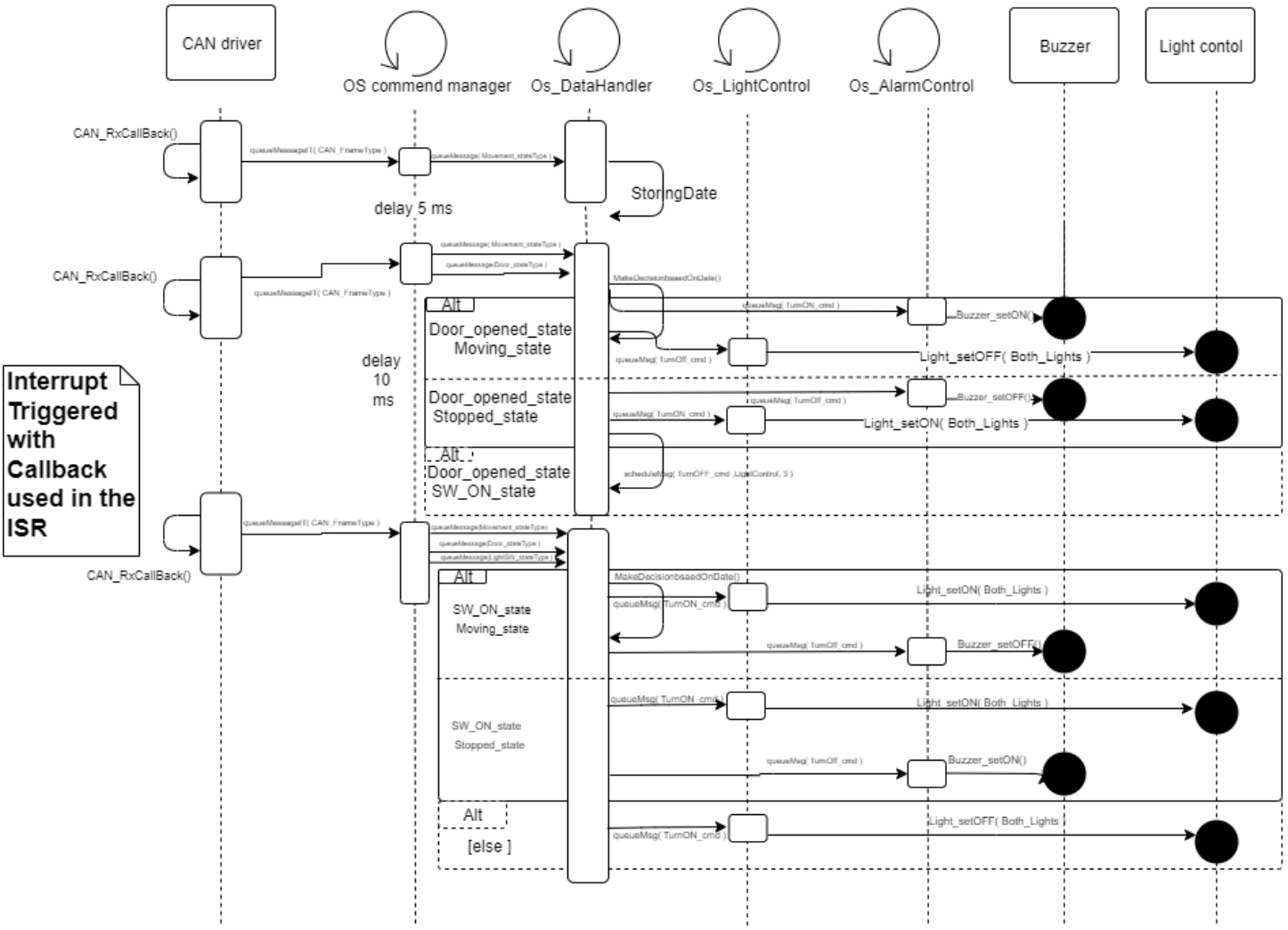







State Machine diagram for the Data Handler
Initialization the modules components and drivers
Idle State
the task set the idle task till the queue has the buffered data
Comm_Manager buffer new msg
Buzzer Commend
the appropriate command is sent to the Alarm control module based on other states
Light Commend
the appropriate command is sent to the light control module based on other states

Sequence Diagram for ECU1



CPU load for the ECU

ECU 1

Every 20 milliseconds, all tasks within the system are scheduled. The hyperperiod for ECU 1 is 20 milliseconds, and if it's assumed that all tasks take an equal amount of time to execute, with a duration of 1 millisecond.

CPU Load ECU 1 =(E1+E2+E3)/Hyperperiod= (1*1+1*2+1*4)/20=35%

ECU 2

MCU 2 is triggered every 5 milliseconds by the CAN driver, which results in the Communication task also being scheduled at this interval. As a result, the DataHandling task is activated every 5 milliseconds as well. The overall cycle time for the system is set at 5 milliseconds. Assuming all tasks have equal execution time and that time is 1millisecond.

CPU Load ECU 2 = (E1+E2)/H=(1*1+1*2)/5=40%