TI-AREM Assignment4. Active Object Pattern and State Machine with AND-States.

Description:
This assignment is a refinement of Assignment3 where students refine their implementation to incorporate a concurrent state machine and use the active object pattern**.**

Goals:
When you have completed this exercise, you will have:
- Experience with implementing AND-States in the State Pattern, implementing an active object pattern where objects can be created for each request to the the active object.
- Experience with implementing command pattern and interconnect it to the state pattern.
- Design of a generic state machine for embedded systems by connecting State and active object patterns.
- Concurrency manipulation using the active object pattern and threads.

Estimated Time: 1 week

Exercise:

1.Reorganize your design and code from the Assignment3 to incorporate the following:
1.1. *RealTimeloop* is a AND-state machine including two concurrent state machines as depicted below.

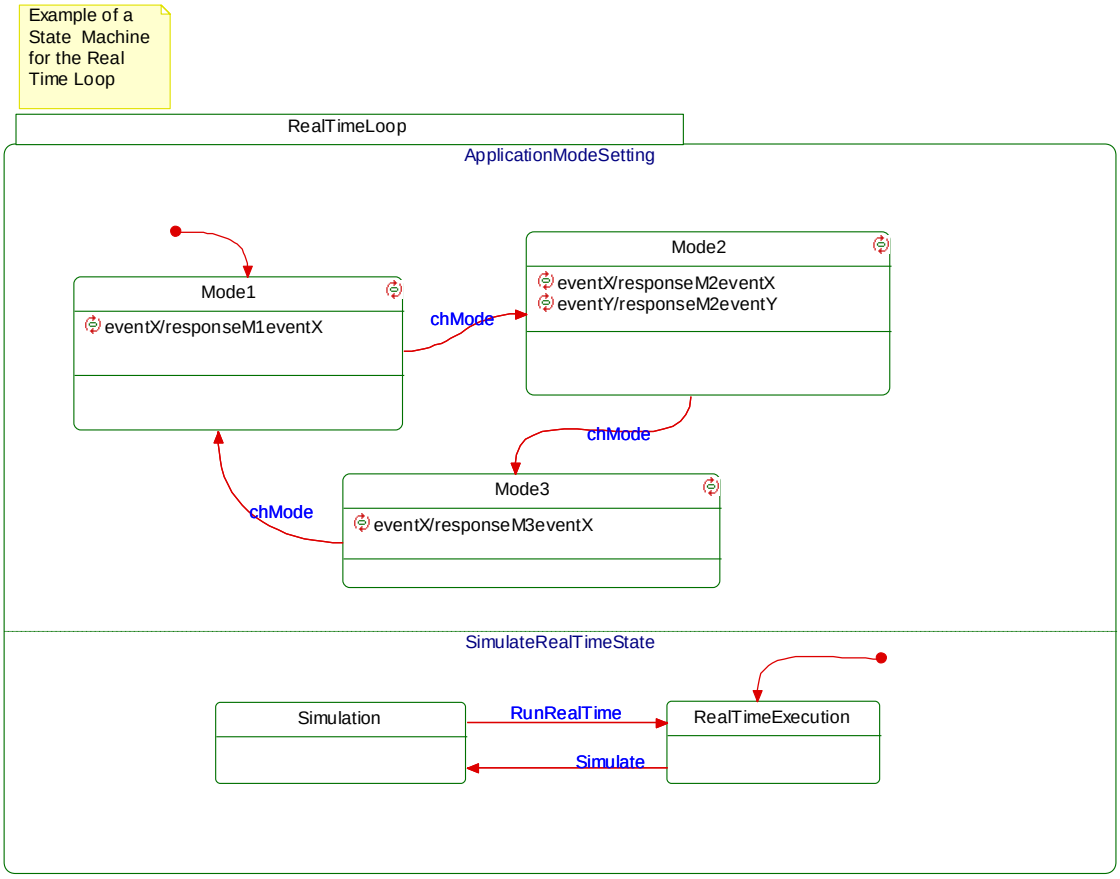
1.2 Consider *RealTimeLoop* to be an active object where for each *Simulate* event a new *Simulation* object is created (to simulate a fresh execution). Each created object, running in the context thread, will increment a global variable *SimCount*. No need to protect the access to the variable *SimCount*.

1.3 Consider each event going to *RealTimeLoop* to be a command.

1.4 Design and discuss the implementation of this enhancement in your current design of your state pattern.

1.5 Test your current solution on the Zybo board.

1.6 Report about the challenges and surprises.


Example of a State Machine for the Real Time Loop
RealTimeLoop
ApplicationModeSetting
Mode1
eventX/responseM1eventX
Mode2
eventX/responseM2eventX
eventY/responseM2eventY
Mode3
eventX/responseM3eventX
chMode
chMode
chMode
SimulateRealTimeState
Simulation
RealTimeExecution
RunRealTime
Simulate