Embedded Real-time Systems
Exercise 1. Implementation of a Generic State Machine for an Embedded System

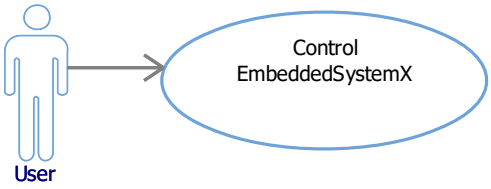
Description:
In this exercise you will implement a State Machine with the *GoF State Pattern* and the *GoF Singleton Pattern*

Goals:
When you have completed this exercise, you will
· have experience with implementing the GoF State pattern for a Hierarchical State Machine
· have developed an implementation for a Generic State Machine for an Embedded System
· have getting started with using Rhapsody or an alternative UML tool to document your design

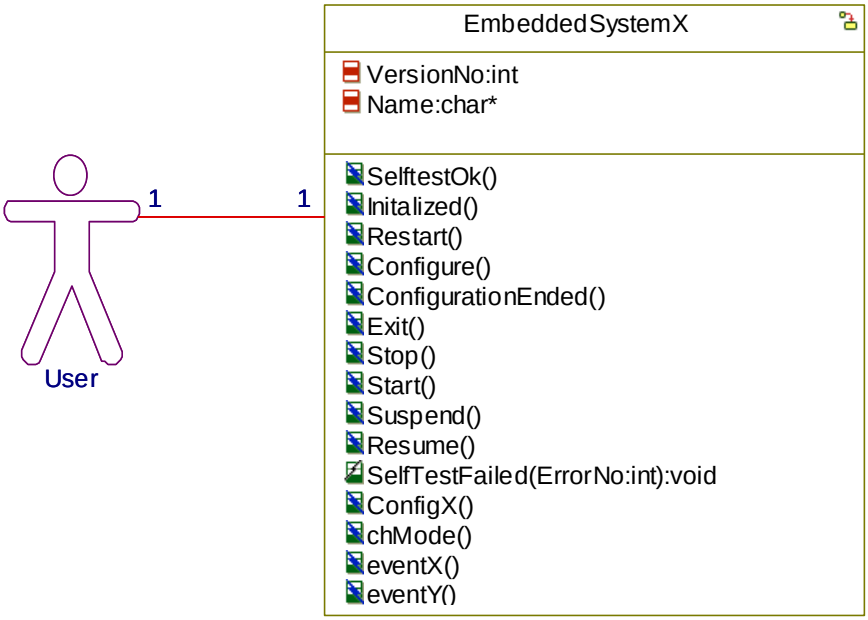
---

Exercise 1:

Use Case Diagram:

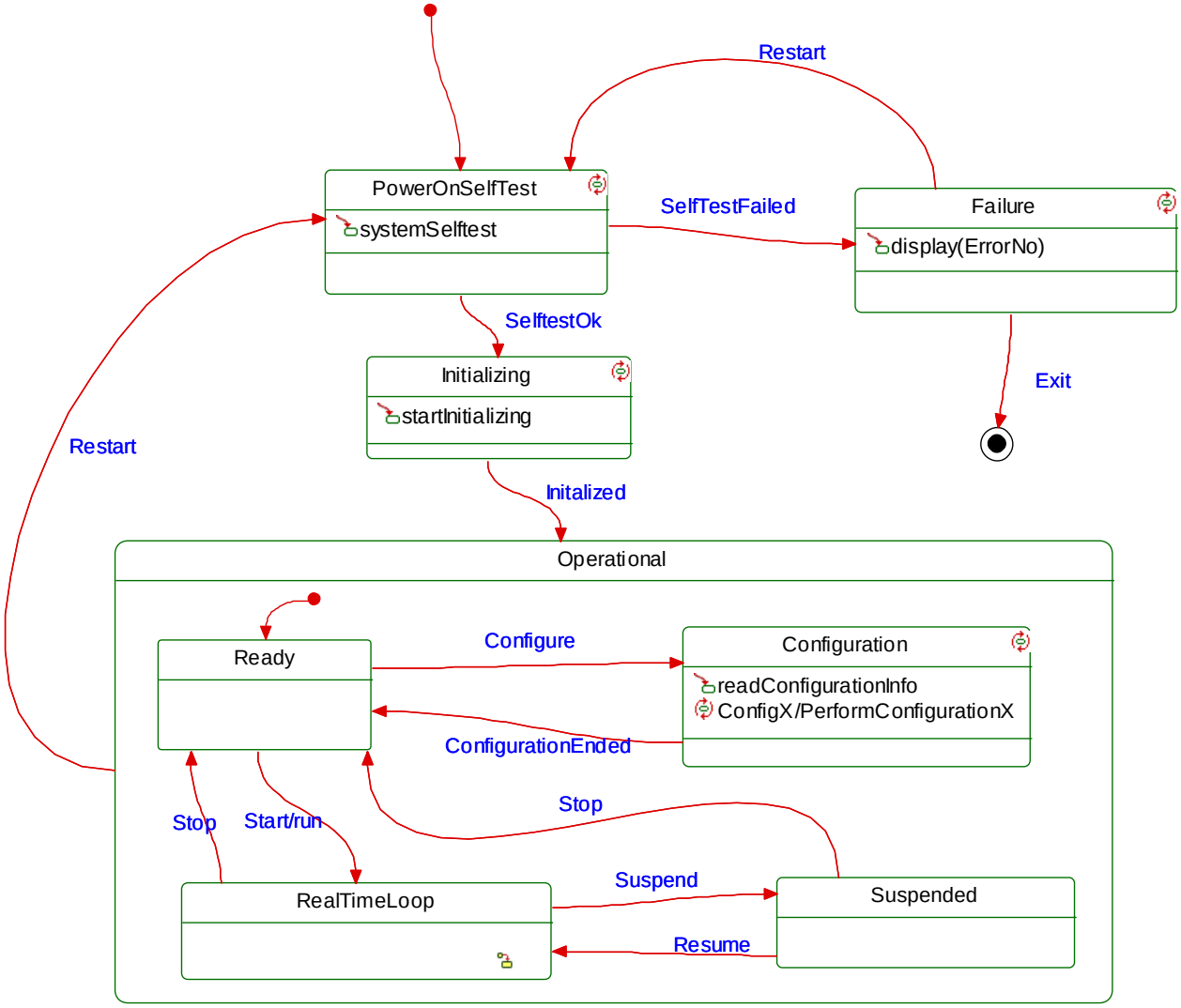


Class Diagram with event operations:



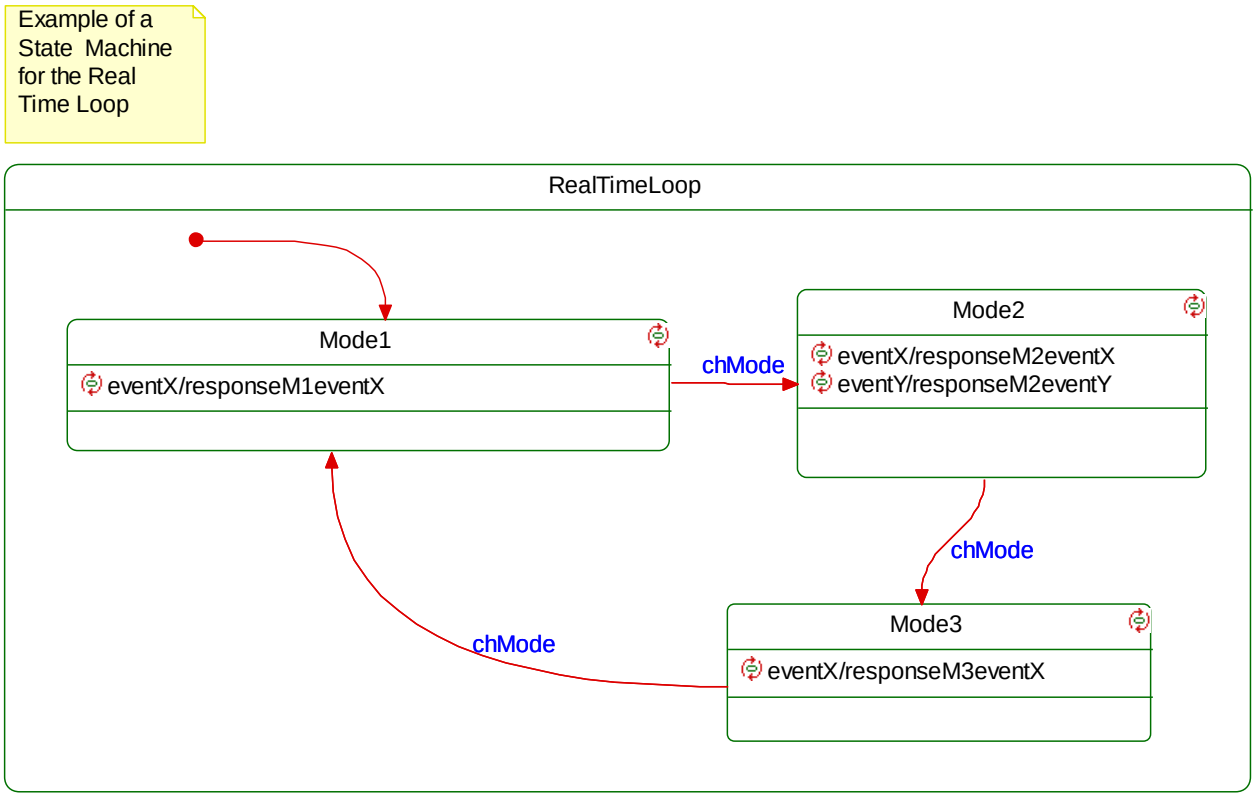
Class EmbeddedSystemX has the state machine shown on the following pages.

State Diagram of EmbeddedSystemX:



The implementation shall include the following sub state diagram for the RealTimeLoop Class.

Sub State Diagram of State RealTimeLoop:



1.1. Design a solution for implementing this State Diagram with the GoF State Pattern, where each state is implemented with the Singleton Pattern.
Result: A class diagram.

1.2. Design, implement and test the design with a PC application implemented in C++.

1.2.1 Insert the class diagram for the solution in Rhapsody or an alternative UML tool

1.2.2 Implement the Context Class (EmbeddedSystemX) with the shown event operations and add the necessary operations for implementing the State Pattern. Add a test operation for displaying the actual state.

1.2.3 Implement the classes for the State Pattern

1.2.4 Test the solution with a main program, where the user activates the public event operations and the actual state is displayed after each invocation.