EMBSYS 110 Assignment 2

Washing Machine Enhancement

1 Goals

The goal of this assignment is to enhance the washing machine active object (AOWashingMachine). It contains two parts:

1. Part I – Add a "Delay" state to allow water to be drained before the door is unlocked when the pause button is pressed. The design is illustrated in the statechart named **AOWashingMachine-exercise.pdf** in the project source tree (under src/AOWashingMachine). The implementation was walked through in the guest lecture.
2. Part II – Show LED patterns according to the following requirements:

   a) Disable all LED patterns when the door is open.

   b) Show LED pattern 0 (first pattern defined in Assignment 1) when the door is closed but unlocked.

   c) Show LED pattern 1 (second pattern defined in Assignment 1) when the door is closed and locked.

2 Setup

You can skip these setup steps if you have already completed them during the guest lecture.

1. Download the project compressed file (platform-stm32f401-nucleo_assignment_2_washing_machine.tgz) from:

   https://canvas.uw.edu/courses/1188665/files/folder/Assignments/Assignment 2 Washing Machine

   Place the downloaded tgz file in **~/Projects/stm32**.
2. Make sure you backup your existing project folder, e.g.

   mv platform-stm32f401-nucleo platform-stm32f401-nucleo.bak

   You may need to copy your LED patterns created for Assignment 1 from the old project folder.
3. Decompress the tgz file with:

   tar xvzf platform-stm32f401-nucleo_assignment_2_washing_machine.tgz

The project folder will be expanded to **~/Projects/stm32/platform-stm32f401-nucleo**.

4. Launch Eclipse. Hit F5 to refresh the project content.

   Or you can right-click on the project in Project Explorer and then click "Refresh".

5. Clean and rebuild the project. Download it to the board and make sure it runs.

   In minicom, you should see a new command named "wash". Try out the command "wash c" or "wash o". You should see log messages printed out on the console.

3 Tasks

1. Implement the "Delay" state according to **Part I** of the **Goals** section above.

2. Implement the display of the LED patterns in AOWashingMachine according to **Part II** of the **Goals** sections above.

   You should replace the default LED patterns in **src/UserLed/LedPattern.cpp** with the ones that you created in Assignment 1.

3. The statechart was created with the free tools named **UMLet**. You can download the latest version from:

   http://umlet.com/changes.htm

   The stand-alone version is more stable. It runs on Java, so it should work in your VM which has Java installed. The use of this tool is NOT required but you are encouraged to add the control of the LED patterns in **AOWashingMachine-exercise.uxf**.

4. Test your code with the console command "wash ..." and ensure the expected log messages and LED patterns are observed.

4 Submission

The due date is 4/30 Monday 11:59pm. Please submit:

1. A zip file containing the source code in **src/UserLed** and **src/AOWashingMachine**. If you have modified other folders you can include them as well.

2. A log file capturing the output of the UART console. You can turn on capturing in minicom by:

   sudo minicom ***-C log.txt***

   Your log should contain the output of the following sequence of events:

   a) After reset, close the door of the washing machine.

   b) Press the "start/pause" button to start washing.

c) Before the washing cycle completes, press the "start/pause" button to pause the washing cycle. The washing machine should go through the "Delay" state to drain the water before unlocking the door.

d) After the door has been unlocked, open the door.

Note the proper LED pattern corresponding to each state should be displayed.

3. Upload to the usual Canvas assignment upload location.