Assignment 2a: Intro Stateflow

1DT059: Model-based Design of Embedded Software
Uppsala University

September 17, 2020

In this assignment you produce Stateflow (and Simulink) models. Each exercise should be implemented in a single .slx-file that is named after the exercise (the solution to assignment 2, exercise 2 should be in file `a2e2.slx`). Also include a file `a2.pdf` with a short high-level description of the ideas of your solutions (you do not need to include obvious things). Include your name at the top of all submitted files.

Assignments are to be solved by students individually. You can discuss ideas and concepts with fellow students, but it is absolutely forbidden to share or copy (even parts of) solutions, lines of code, or similar.

Problem 1 Vending Machine*:*
(**20p**) Consider a simple vending machine for buying snacks at the ICT Campus of Uppsala University. The machine accepts coins (i.e., real old-fashioned coins), accepts requests for snacks, and delivers snacks (if enough coins have been inserted). To make the problem more specific:

- The machine accepts coins for for 1kr, 5kr, and 10kr.
- There are two types of Snacks: **A** (costs 8kr), and **B** (costs 6kr).
- The workflow is that the machine
  - first welcomes the customer and asks for coins.
  - Thereafter, the user can insert coins. When the user wants to order, he/she must first exit the "coin-inserting" mode by pressing an "order" button.
  - After pressing the "order" button, the user can then press button for **A** or for **B**. If the current balance (sum of inserted coins) is enough the snack is delivered, otherwise some "sorry" message is displayed.
  - After ordering snacks, the user can choose between either of
    - "Finish", after which any excess change is payed back, and the machine goes back to its idle state

* "Insert more money", which allows to enter more coins and thereafter order snacks
* "order more snacks", which allows to order another snacks.

Make a well structured Stateflow model of the vending machine. One should be able to interact with the machine by inserting coins, pushing buttons, and seeing the output.

Problem 2 Microwave Oven*:*
(**20p**) Consider a simple version of a controller for a microwave oven. The microwave is used to cook food items for specified periods of time at specified levels of power. A user can enter desired cooking time and cooking power, and start or stop the oven. To make the problem more specific.

- The oven allows the user to select cooking time. For simplicity, the cooking time can be given in seconds (up to, say, 1000s). The remaining cooking time is displayed on the oven. The initial (default) value is 0.
- The oven also allows the user to select cooking power, as any integer between 0 and 800. The current cooking power is displayed on the oven. The initial (default) value is 0.
- The oven has some buttons:

  Start for starting or resuming cooking. During cooking, the display for remaining time is decreased with the passage of time.

  Stop for stopping the cooking. If the oven is cooking, this suspends cooking and stops the decrease of time on the time display. IF the oven is not cooking, the **Stop** button acts a reset, which resets the remaining time to 0.
- The oven has a door which can be opened or closed by the user. Opening the door suspends cooking and stops the decrease of time on the time display. Closing the door does not restart the cooking: for that the **Start** button must also be pressed.
- When the time display reaches 0, the oven sounds an alarm, and stops the cooking.
- During cooking, the user can freely change the remaining time and cooking power. If the user changes remaining time to 0, the alarm sounds, and cooking stops (if it was active).
- If the user has been inactive for more than 300 seconds (e.g., set the cooking time, but thereafter done nothing), the oven resets to its initial values.

Make a well structured Stateflow model of a microwave oven as above. Realize the interface by suitable interface blocks, that will allow you to interact with the model during simulation.

Submission

Solutions (all files) to this assignment are to be submitted via the Student Portal by
Thursday, September 24, 2020