Vending Machine - Coffee

ESDM Project

Figure 1: Coffee Vending Machine

General Description

1. Create and test a Simulink model containing a state machine implementing the logic module of a vending machine.
2. Write a report on the project, containing:
   a. An overall description of the design (how it works, states, transitions etc,).
   b. Some tests of the functionality (2-3 tests, depending on complexity, covering normal usage and some error scenario)

      For each test, indicate:

      - The test scenario: what are the inputs, what are the desired outputs
      - The test results: include screenshots from the tests, to prove the tests work

Requirements

1. The vending machine has 5 products categories available: Coffee, Espresso, Hot Water, Cappuccino, Moccaccino.
2. The machine has all these products pre-made. All it does is to pour them in the right quantities.
3. The machine starts with 1 liter of each beverage.
4. List of inputs and outputs of the model:

   Inputs:

   - ProductSelection: integer (0 to 5)
     - when 0, no product is selected
     - when non-zero, it is the code of product selected by the user (1 to 5)
   - MoneyInput: integer
     - when 0, no money is inserted
     - when non-zero, it is the current value of the money inserted by the customer
   - Cancel: boolean
     - when True, cancels an ongoing operation. All money input until this moment shall be returned to the customer.
   - ResetStock
     - when True, the stock for all products is set to 1 liter (e.g. the machine was refilled).

   Outputs:

   - DispenseCup: boolean
     - the transition from False to True activates the dispensing of one plastic cup
   - PourProduct: integer (0 to 5), controls the dispensing of products
     - when 0, nothing happens
     - when non-zero, the product with that code is poured in the cup
     - pouring lasts for 5 seconds, for all products except Espresso, for which it lasts only 1 second
     - after the time is elapsed, the machine must stop pouring by setting PourProduct output back to 0
     - quantity is always 200 milliliters, except for Espresso, where it is only 50 milliliters
   - DispenseStirrer: boolean
     - the transition from False to True activates the dispensing of one stirrer

- MoneyReturn: integer, controls the money returned to the customer
  - when 0, nothing happens
  - when non-zero, the specified amount of money will be returned to the customer
- Status: integer, a status message indicating the current state
  - 0 = Idle, awaiting operation
  - 1 = Operation in progress
  - 2 = Success
  - 3 = Incorrect product code
  - 4 = Product out of stock

5. The vending machine operates in 4 basic steps:
   - first you enter the product code of the product
   - then you enter the money
   - then the product is dispensed: first the cup, then the beverage, then the stirrer
   - then the rest of the money is returned
6. The vending machine starts with 1 liter of each beverage.
7. The machine keeps track of how much beverage it still has available, and refuses to produce a beverage when it doesn't have the necessary quantity available.
8. The price of every type of product is fixed and known (you pick some value, e.g 5).
9. The machine shall detect if the user requests an invalid product code, and signal this at the Status output.
10. The machine shall detect if the user requests a product which is currently out of stock, and signal at the Status output.
11. The machine shall calculate the rest of the money and provide back the change (Note: assume the machine has an infinite supply of coins/notes).
12. After dispensing a product, the machine will wait 5 seconds before accepting any new operation.
13. The quantity of products available can be reset back to the value of 1 liters when the input `ResetStock` is activated.
14. The machine shall always provide a status code output.
15. The `ResetStock` input button shall be debounced both ways, with a time duration of 0.4 seconds.
16. Use parameters from Matlab for all values you consider necessary (e.g. duration of delays, prices etc.). Our customer may want to adjust the parameters at any time.
17. Test your state machine (use one/multiple separate test models if necessary)