SCHOOL: Electronic and Electrical Engineering

Loughborough University

**Devon Kerai (....8203)**

**COVER SHEET FOR ASSESSED COURSEWORK**

**Students must complete this coversheet to accompany each piece of assessed coursework.**

**PROGRAMME: Electronic and Electrical Engineering (ELUB10)** **YEAR/PART: C**

**MODULE: Real Time Software Engineering (14ELC018)**

**ASSESSMENT TITLE: UML design**

**STAFF MEMBER RESPONSIBLE: David Mulvaney**

**Student Declaration (please sign and date)**

I certify that the attached work is my own work and that anything taken from or based upon the work of others has its source clearly and explicitly cited. I certify that each and every quotation, diagram or other piece of exposition which is copied from or based upon the work of others has its source clearly acknowledged in the text. All field work and/or laboratory work has been carried out by me (or my group) with no more assistance from the members of the department/school than has been specified. Any additional assistance which has been received is indicated and referenced in my work.

**I have abided by the department/school policy on plagiarism and the Coursework Code of Practice as specified in the Student Handbook (see Learn).**

Submitted By:

**Group: Group 5**

**Devon Kerai (....8203)** Date: 28/11/2014 Signature: Dkei

**Simon Alexander Tate (....4127)** Signature (if applicable): SATate

**Guoyue Wang (....7301)** Signature (if applicable): 王国跃

**Specific Items:** Written documents(1)

**Note:**
**Late-Hand-In: Approval for an extension to the deadline must be sought from the Responsible Examiner prior to the deadline. For any submission handed in after an agreed extension you must ensure that you have completed an Impaired Performance Form for your mark to be considered by the Module Board.**

# 14ELC018 Real-time Software Engineering Coursework

## Task 1 - Top-level design of washing machine system

By Devon Kerai, Simon Tate and Guoyue Wang

## Specifications

Program Selection

a) The washing programs can be selected by three buttons.
b) The selected program will be indicated by three program LEDs as a binary encoded number.

Washing Process

a) The processes are designed to be in the following order and last 2.5 hours.
    i. Filling in water for 5 minutes.
    ii. Heating water for 5 minutes.
    iii. Washing for 90 minutes.
    iv. Spinning for 20 minutes.
    v. Drying for 30 minutes.
    vi. Complete.
    vii. System Reset.

b) Press the Accept button to start the washing program chosen.
c) Pressing the Cancel button will pause the system and can be continued by pressing the Accept button.
d) Pressing the Cancel button twice will reset the system.

Reservoir Door

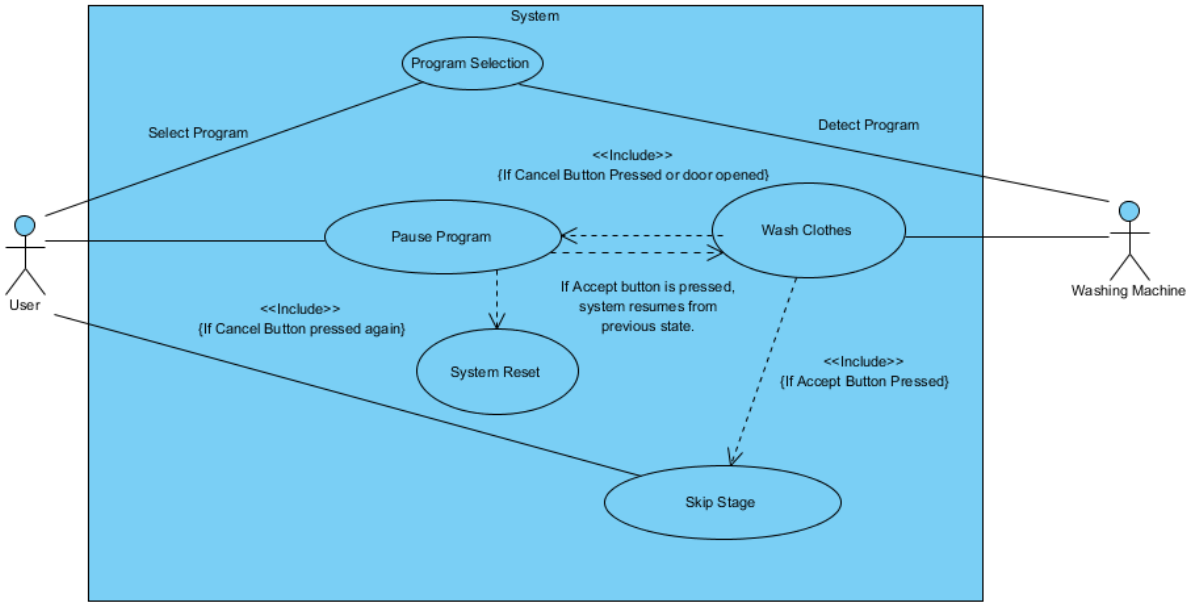
a) The door switch is indicated by a LED.
b) The program will not start unless the reservoir door is closed.
c) If the door is opened during the washing process, the system will suspend the wash and it can be resumed when the door is closed again.
d) If the Cancel button is pressed during suspension, the system will be reset.

Advanced Functions

a) A buzzer will generate a warning sound for three seconds when the system is interrupted (e.g. Reservoir door is opened during the washing process).
b) Pressing the Accept button during washing process will cause the program to advance to the next stage.
c) A 7-segment display is used to indicate the current washing stage.

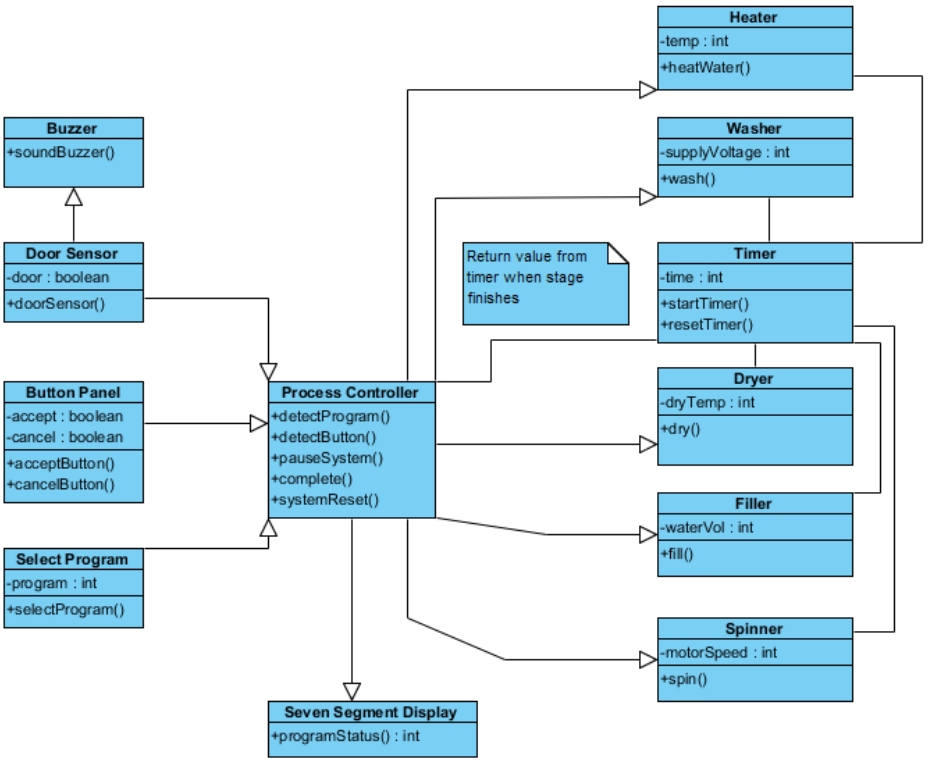
## Design Diagrams

Use Case Diagram (Structural)




To initiate a wash, the user has to select a washing program. The washing machine then detects the chosen washing program and executes it. While the washing program is executing, you can press either the Accept or Cancel button. Pressing the Accept button advances the program to the next stage whereas pressing the Cancel button pauses the system. While the system is paused, if the Accept button is pressed, the washing program resumes to its previous state and if the Cancel button is pressed twice, the system resets.

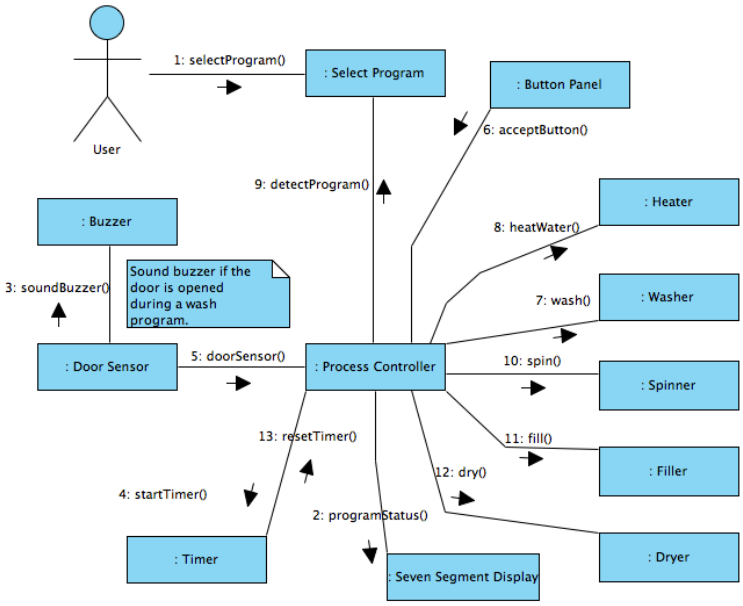
Class Diagram (Structural)




The class diagram includes a process controller that is the main function of the program and controls how the overall system operates. The process controller accepts inputs from the Door Sensor, Accept Button, Cancel Button and the Select Program classes. The Door Sensor and Button Panel pass Boolean values to the process controller while the Select Program passes an integer value.

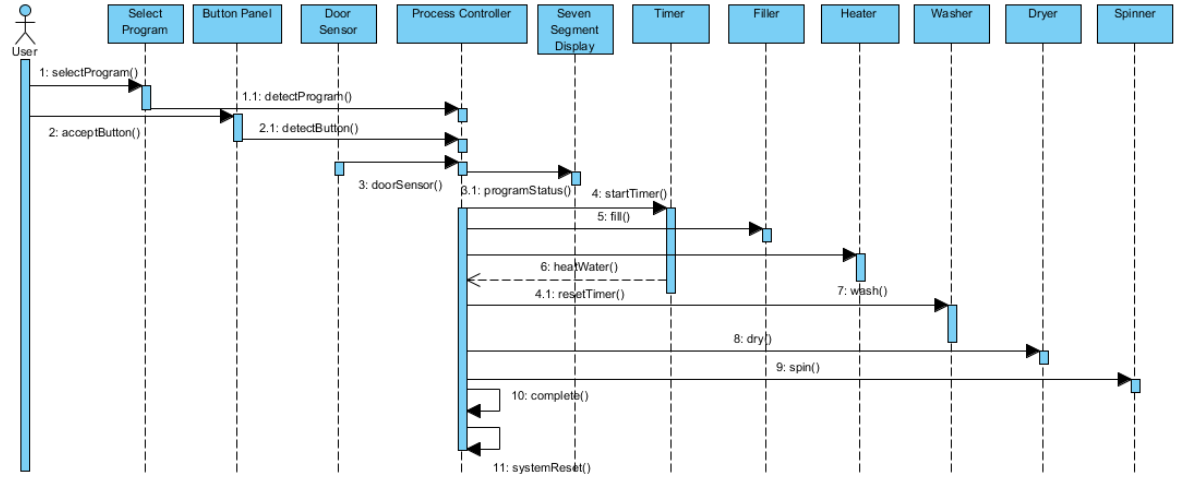
The process controller then uses these values to output to the Heater, Washer, Dryer, Filler, Spinner, Timer and Seven Segment Display classes. The values passed to these (all integers) are dependent on the program selected by the user. The timer is associated with all the other classes through the process controller. It passes the timer value and returns it to the process controller when the desired integer has been reached so that the system can determine when the current stage has ended.

## Collaboration Diagram (Behavioural)

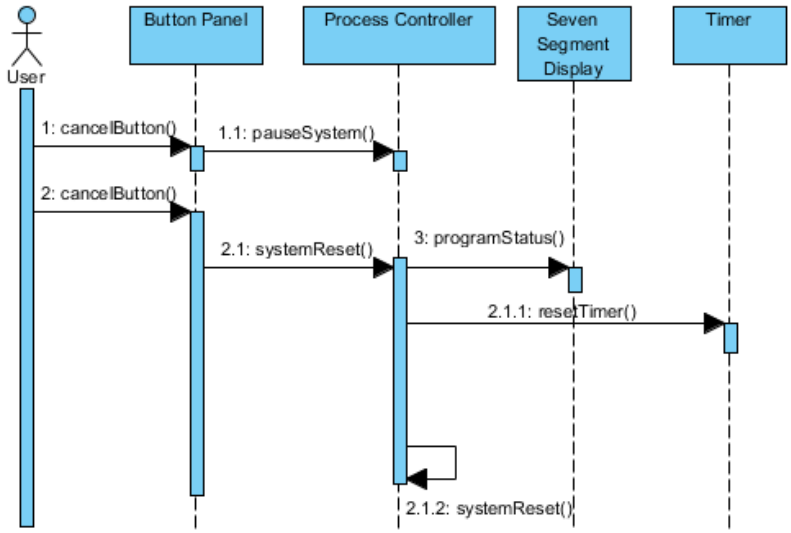



The collaboration diagram shows the functions that are called when selecting a washing program. The Process Controller acts as the main class that calls the majority of the functions. The “doorSensor” function includes a hardware interrupt which sounds a buzzer when the door is opened during a washing program.

## Sequence Diagram (Behavioural)

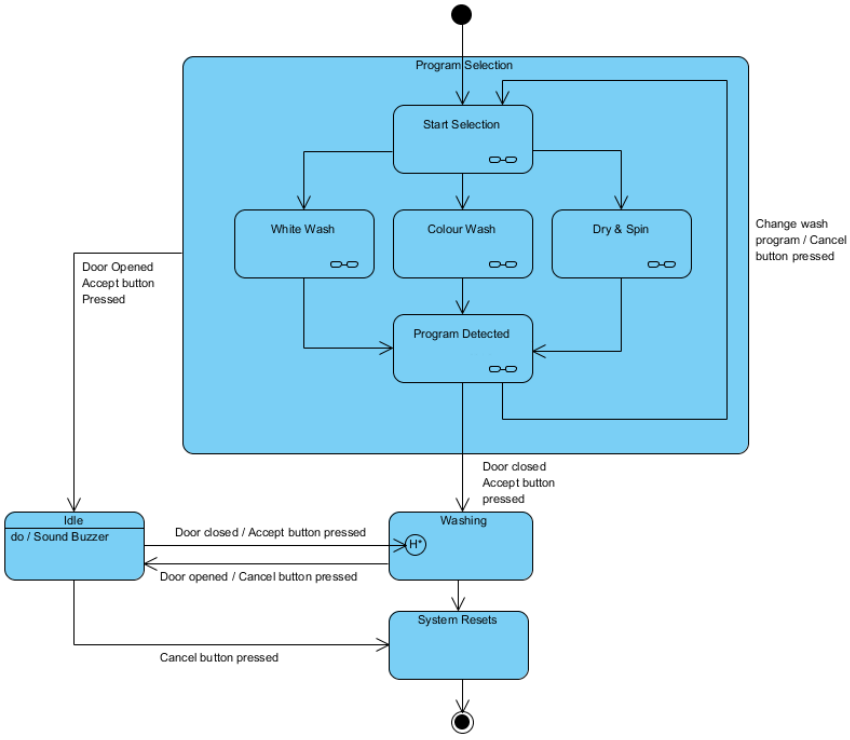



Sequence diagram 1 shows the order of the function calls when a washing program is selected. Some of the last calls to make is to verify that the washing program is complete and then to reset the system.

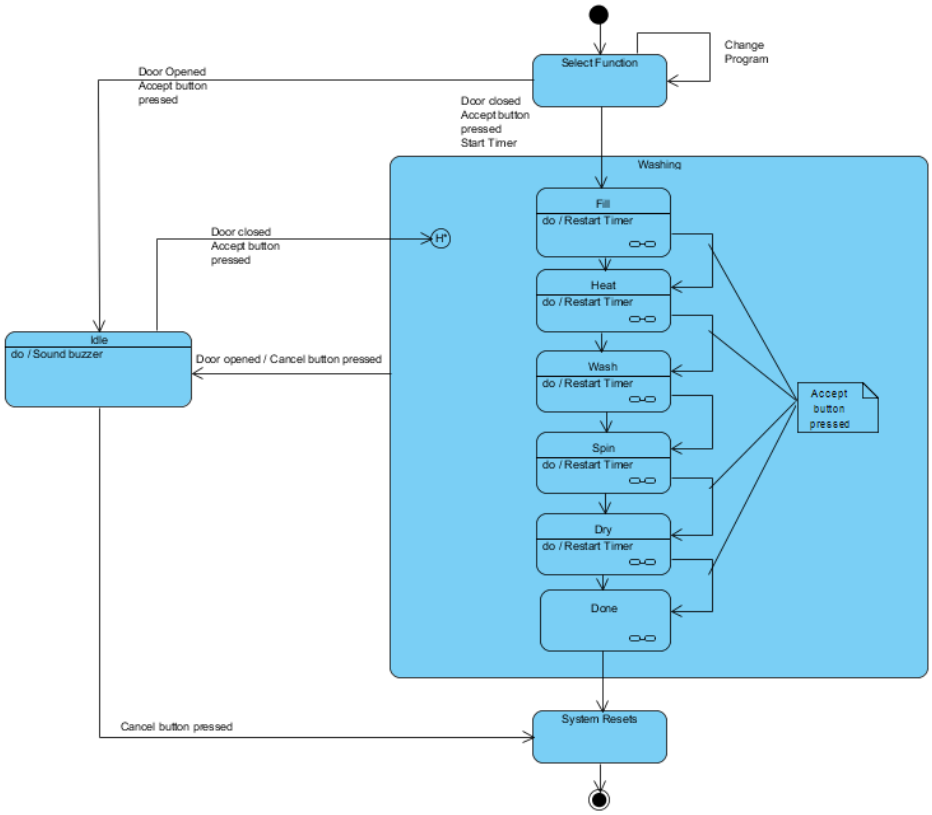



Sequence diagram 2 demonstrates the Cancel button functionality. Once pressed twice, the entire system resets and the seven-segment display is cleared.

State Diagram (Behavioural)




As the system starts, it enters the Program Selection state which is distributed into several sub-states for selecting and detecting the programs. In order to proceed to the next state, the system is conditioned by the Accept button (that has to be pressed) and the door which has to be closed.




After selecting a washing program, the Washing state is also distributed into several sub-states. The sub-states are used to describe the washing stages, a timer is used to determine the time for each stage and it resets after each stage completes. The Washing state will also be interrupted if either the reservoir door opens or the Cancel button is pressed. The History sign means that the system can be restored and resumed from the interruption. Finally, if the Cancel Button is pressed again during the interruption, the system will be reset to its initial state.

## Conclusion

The above UML diagrams shows our thoughts while designing our washing machine from a top-level design and will be used to determine the underlining functionality to include in our software to control the operations of the Bytronic washing machine simulator.

This design will be used to interface with our hardware through low-level device drivers which will be written at a later stage. By changing the low-level device drivers, our design can be used on multiple washing machines.