Department of Electrical Engineering and Computer Science
EECS 395/495 – Embedded System Design and Synthesis
Spring 2017

Efficient Morning Routine System

Milestone 2

Authors: Alexey Revinski, Karan Shah
Prepared by: Karan Shah
Prepared for: Prof. Lawrence Henschen, Prof. Chung Lee
Date: Friday, May 19, 2017

Table of Contents

List of Figures

List of Tables

Executive Summary

This document provides a detailed overview the computational aspect of an embedded morning routine system design, which would offer some of its users a hands-free way of starting their mornings, and to some – the much-needed wake up discipline.

The main system consists of timer-based sound and light alarm modules that work in conjunction with an automated window-curtain opening system to offer the user ways of waking themselves up that were previously unexplored. Users would have a choice of natural or artificial light stimuli, as well as sound. Gentle music or harsh alarm – the user has complete control over their waking.

Besides these three components, the system will also offer multiple add-on features to decrease the amount of time the user spends to get ready in the morning and to make the morning hours more enjoyable for those who do not look forward to early dawn.

These features will include a shower temperature conditioning module that will keep the shower water at a temperature that is comfortable for the user; a coffee-making module that will be able to turn on a pre-filled coffee-machine at a specified time of day, a daily organizer module that will show the user his or her list of tasks for the day; and a newscasting module that will have the capability to be turned on a time-based schedule to tune the user into the morning news.

In the previous document, we specified some of the user cases and different scenarios that can take place using the system. This document describes the modelling aspect of the system in detail.

Design Overview

The proposed morning routine system has many subsystems that are working at the same time. Hence, it can be represented by a hierarchical state machine. The hierarchical state machine is shown in Figure 1 along with definitions of each state and variable. Apart from initialization and fault states, all of the actual functions of the system are in one “Normal Function” AND state. This AND super-state has four subsystems working in parallel – the “Front End”, which includes the Light, Sound, Window Curtain, Coffee-Making, and Shower Temperature modules (the latter two were chosen as a representation of the extra features in the system); the “Timing System”, which includes three separate timers; the “Sensor System”, which includes all of the sensing equipment used by the system; and, finally, the “Man-Machine Interface”, which includes a digital display and the main processor of the system.

The Front-End subsystem will be directly interacting with the user – waking him up, making his coffee, and maintaining his shower temperature at a comfortable level. The modules within the Front-End subsystem will have simple “on/off” functionality; some modules will have analog capabilities – brightness of light, intensity of sound, etc. But, those variables will be controlled by the main processor, as part of the MMI.

The Timing System will consist of two timers – one responsible for the actual morning alarm, and, another – for turning off the Front-End modules after a certain period of no action from the user.

The sensing unit will have (so far) two sensors – a motion detecting module, to confirm that the user woke up and determine his location in the house, and a force sensor embedded into the user’s mattress, to determine his level of alertness and activity when in bed.

The Man-Machine Interface will, as mentioned, consist of the digital display and the main processor. The user will use the digital display for programming the system and getting information like shower temperature, time of day, news, daily organizer, etc. The main processor will control the entire system – it will accept programming commands from the user, interact with timers, and turn on and off certain modules based on feedback from all subsystems.

The following sections will describe each of the modules and their states in detail.

Light Module Subsystem

The light module subsystem is used for control of light's in the room. The finite state machine for light system is shown in Figure 2. The system consists of two states namely, *on* and *off*. The system will go into on state when the user manually switches on the light or the system receives a command from the processing unit or the sensing unit acknowledges the user has entered the room or there is an interrupt from the user for reconfiguring the module. The system will switch off only when all the conditions are not present. There is also a *history* state for going back to the previous configuration when the user makes certain changes to the system.

Window and Curtain Subsystem

The window and curtain module subsystem is used for control of curtain and window in the room. The finite state machine for window and curtain system is shown in Figure 4. The system consists of two states namely, *on* and *off*. The system will go into on state when the system receives a command from the processing unit or there is an interrupt from the user for reconfiguring the module. The system will switch off only when all the conditions are not present. There is also a *history* state for going back to the previous configuration when the user makes certain changes to the system.

Sound Subsystem

The sound module subsystem is used sounding the alarm when the timer overflows. The finite state machine for window and curtain system is shown in Figure 3. The system will be on when the processor sends a request for sounding the alarm. The system will switch off when one of the timer's value is equal to the sound system timer.

Coffee Subsystem

The coffee subsystem is used for making the coffee pot on and after a pre-defined time, the system will switch off. The finite state machine for window and curtain system is shown in Figure 12. The system becomes on after the processing unit sends a message to the system.

Shower Subsystem

The shower subsystem is used for attaining particular temperature of the shower defined by the user. The finite state machine for window and curtain system is shown in Figure 5. The system becomes on after the processing unit sends a message to the system for switching on the module. It will switch off after 5min as the water temperature is a particular level.

Timer Subsystem

The timer subsystem is used for controlling the timers for the alarm system. The system consists of two timers – timer1 and timer2. The finite state machine for two timer system's is shown in Figure 6 and Figure 7 respectively. Timer 1 is used for sounding the alarm. When timer1 overflows, the sound module sound the alarm. After timer 1 overflows, timer 2 wakes up and runs till the user acknowledges the alarm. It has two states namely on and wait states.

Sensor Subsystem

The sensor subsystem is used for sensing whether the user is in the room or sleeping on the bed. The system consists of two sensors – sensor1 and sensor2. The finite state machine for two timer system's is shown in Figure 10 and Figure 11 respectively. Sensor 1 is used for sensing if the person is in the room or not. It switches on when tim1 is completed or user enters the room. Sensor 2 is used for determining if the user is in the bed or not after tim2 is on.

Man-Machine Interface

The Man-machine interface consists of a processor and a display. The FSM for the processor and the display is shown in Figure 9 and Figure 8 respectively. The display will wake up when it receives an interrupt from the user or the processor send a command for displaying some information. The processor is switched on as soon as the overall system is initialized. The processor will only stop when a fault occurs or user switches off the system manually.

Appendix A: Overall System Model

State	Type	Description
INIT	Basic State	Initialization State
ON	AND super state	System is working properly
F	Basic State	Fault state
FE	AND super state	Front End subsystem
TIM	AND super state	Timing subsystem
MMI	AND super state	Man-Machine Interface subsystem
SEN	AND super state	Sensing subsystem
L	OR super state	Light module subsystem
S	OR super state	Sound module subsystem
W	OR super state	Window-Curtain subsystem
C	OR super state	Coffee-making subsystem
T	OR super state	Shower temperature subsystem
T1	OR super state	Timer 1 subsystem
T2	OR super state	Timer 2 subsystem
S1	OR super state	Sensor 1 subsystem
S2	OR super state	Sensor 2 subsystem
M	OR super state	Microprocessor
D	OR super state	Display

Table 1: States used and their description

Variable	Description
a	=1 delay before initialization
b	=1 all subsystems are on
c	=1 all front-end modules are on
d	=1 all timers are initialized
e	=1 all sensors are initialized
f	=1 interface between user and processor initialized
g	=1 fault taken place
h	=1 fault rectified

Table 2: Variables used and their description

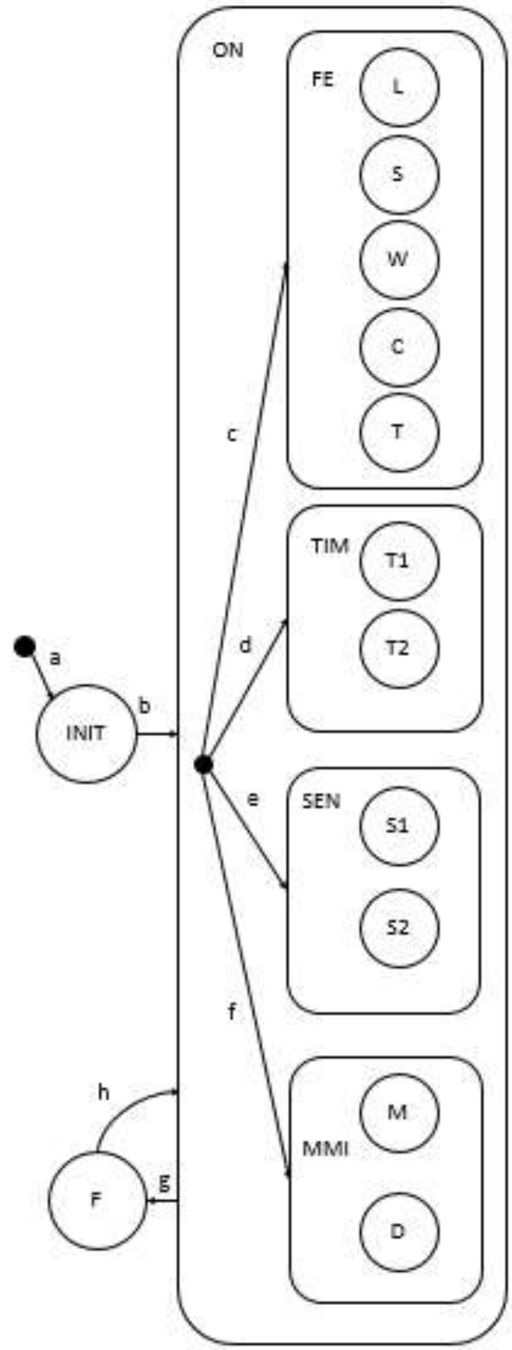

Figure 1: Overall System Model (Alex)

Appendix B: Light Subsystem Model

Light Module

Input:
x = Input from user
y = Message from processing unit
z = user in in the room

Output:
light_module = system is either on or off
light_history = previous configuration of light module

States:
On = system is in on state
Off = system is in off state
H = history

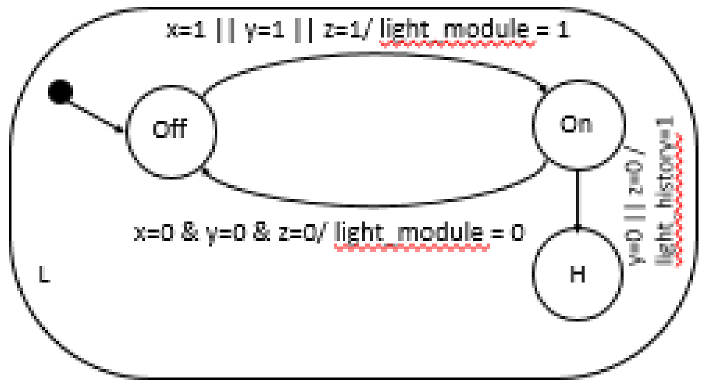

Figure 2: Light Subsystem Model (Karan)

Appendix C: Sound Subsystem Model

Sound Module

Input:
y = Message from processing unit
sound_on= user selects a preview for a particular sound

Output:
sound_module = system is either on or off

Timers:
T_sound = timer for sound system
Tim1 = timer for alarm
Tim2 = timer for snooze

States:
On = system is in on state
Off = system is in off state

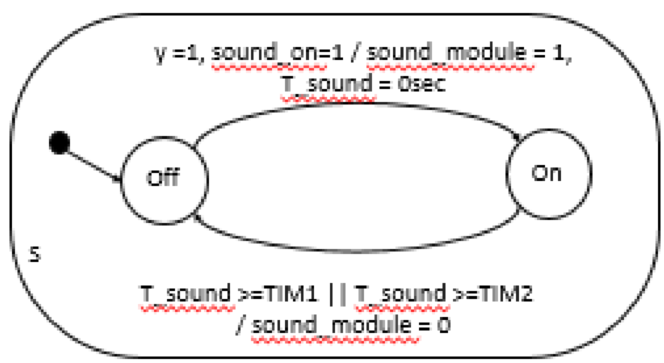

Figure 3: Sound Subsystem Model (Karan)

Appendix D: Window Subsystem Model

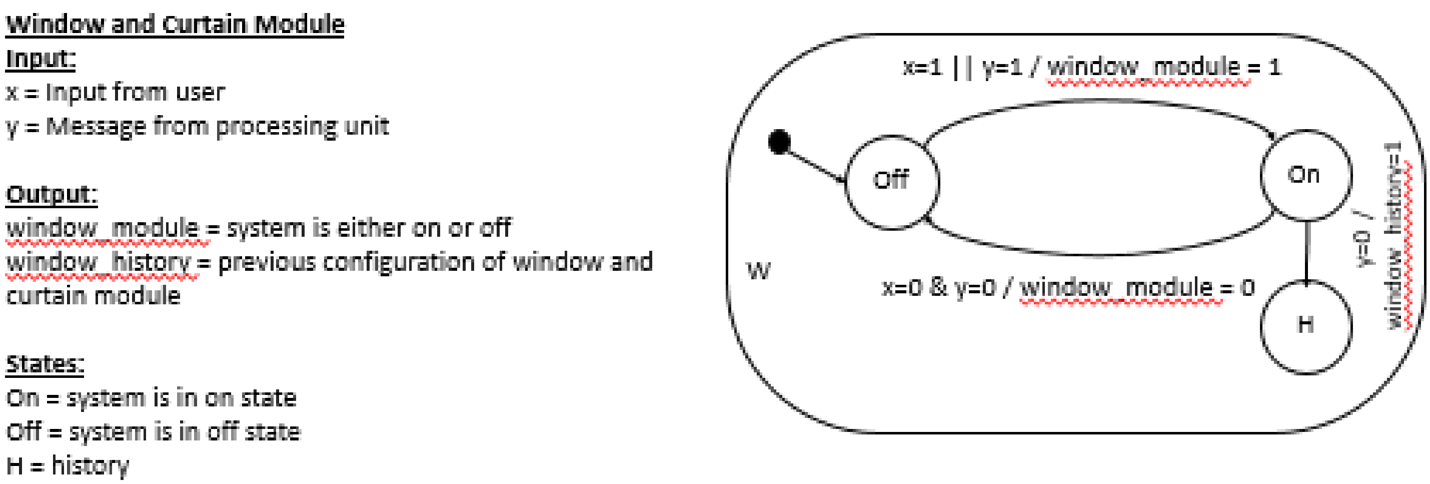

Figure 4: Window Subsystem Model (Karan)

Appendix E: Shower Subsystem Model

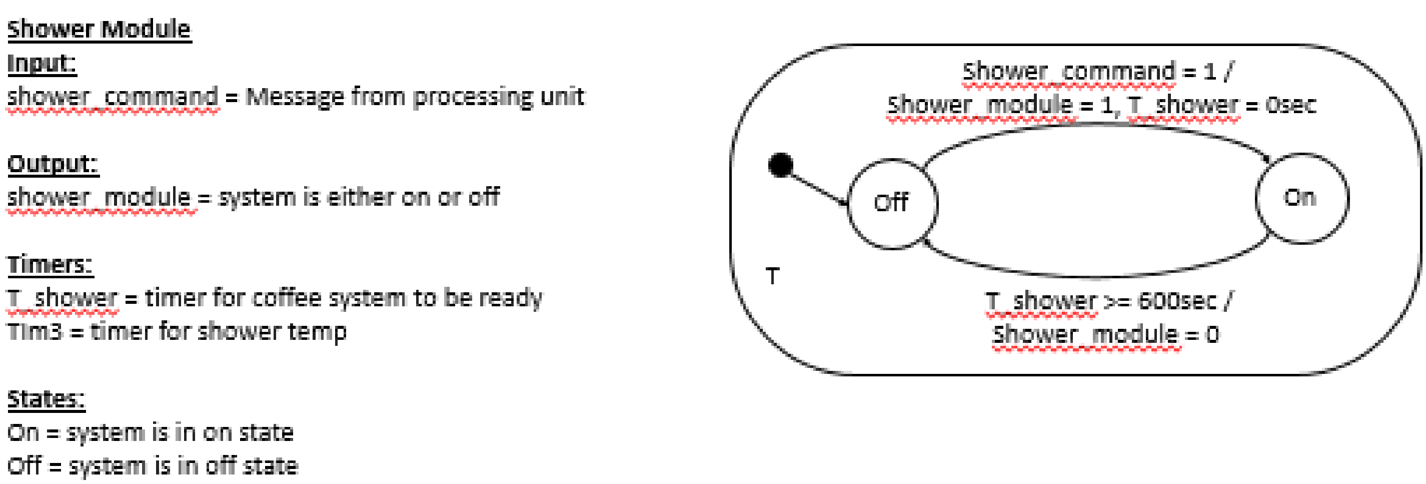

Figure 5: Shower Subsystem Model (Karan)

Appendix F: Timer 1 Subsystem Model

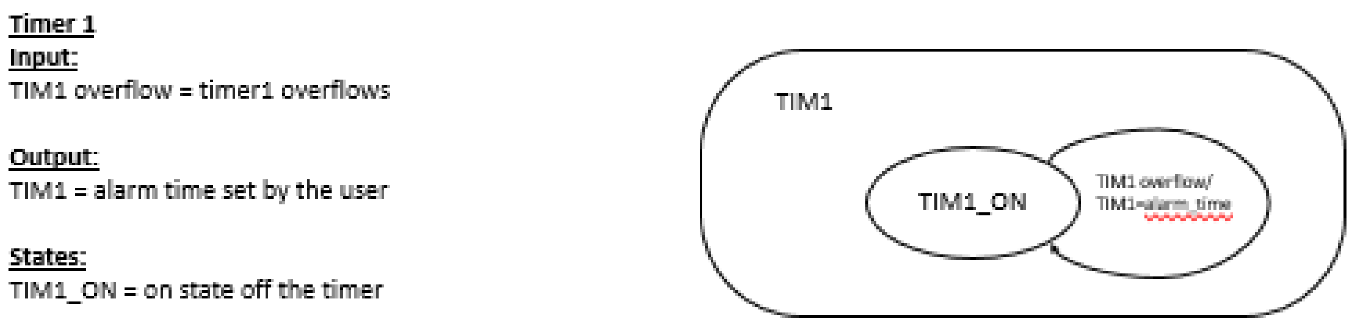

Figure 6: Timer 1 Subsystem Model (Alex)

Appendix G: Timer 2 Subsystem Model

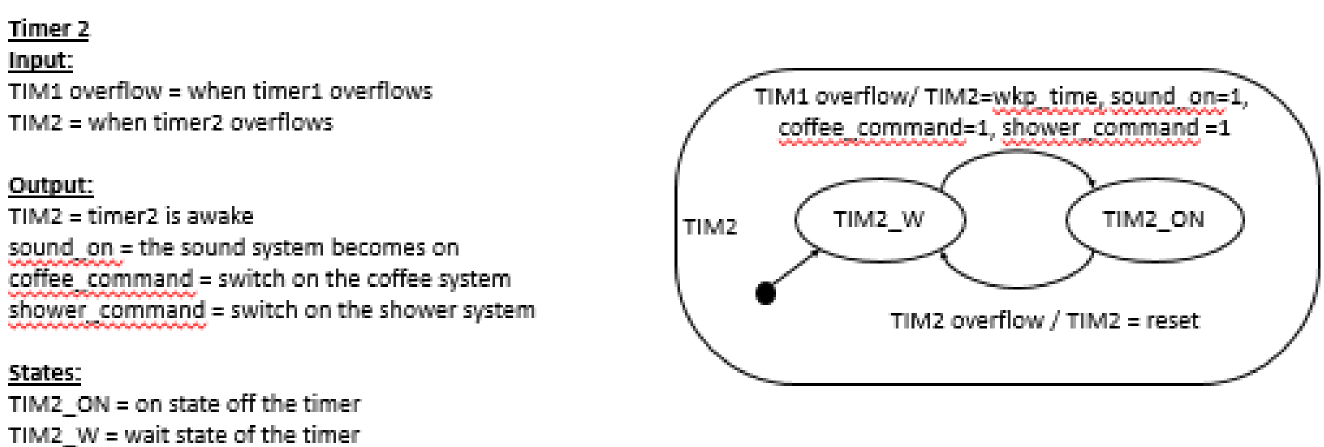

Figure 7: Timer 2 Subsystem Model (Alex)

Appendix H: Display Subsystem Model

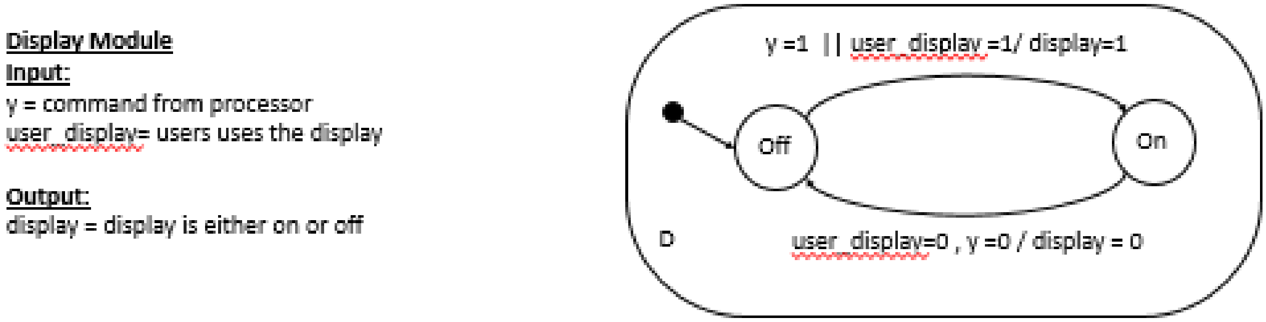

Figure 8: Display Subsystem Model (Karan)

Appendix I: Processor Subsystem Model

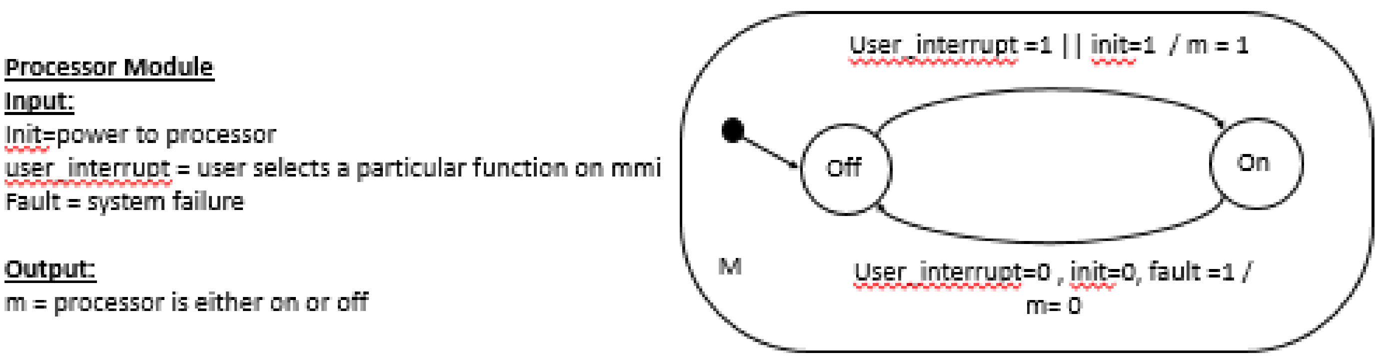

Figure 9: Processor Subsystem Model (Karan)

Appendix J: Sensor1 Subsystem Model

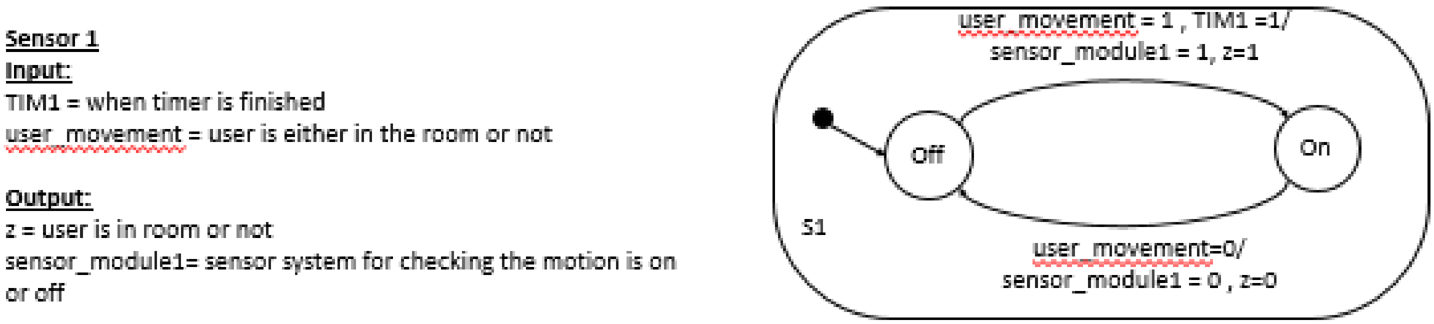

Figure 10: Sensor 1 Subsystem Model (Karan)

Appendix K: Sensor2 Subsystem Model

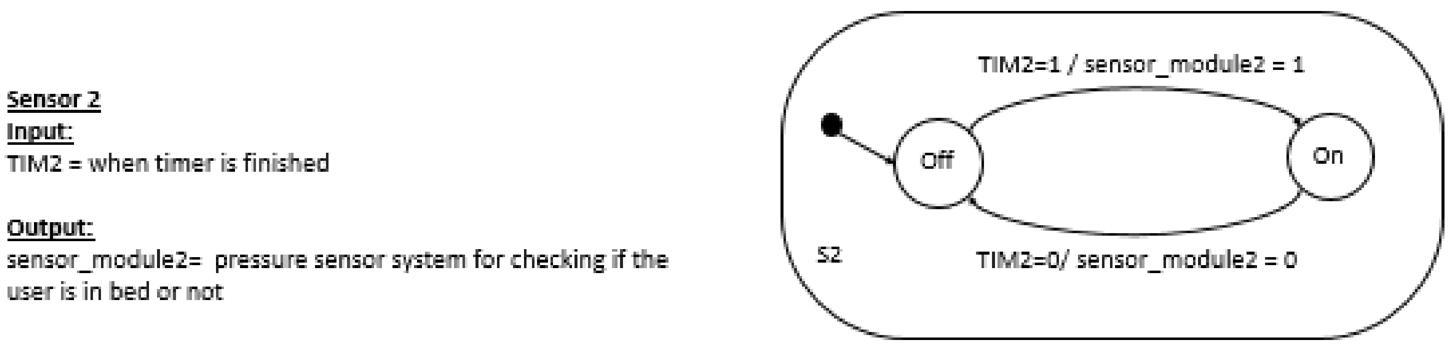

Figure 11: Sensor 2 Subsystem Model (Karan)

Appendix L: Coffee Subsystem Model

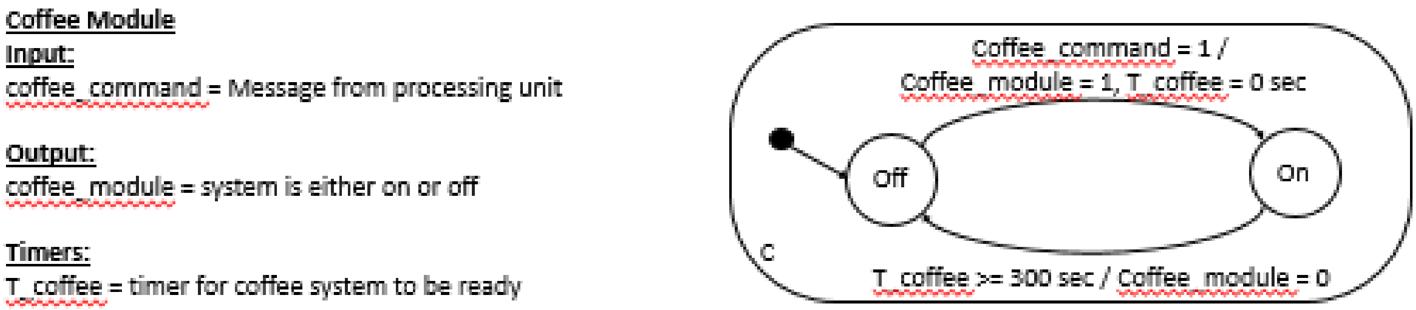

Figure 12: Coffee Subsystem Model (Karan)