**Assignment #1 (RTOS for Embedded Systems**

Sketch UML State Diagrams for the following problem:

You are assigned to develop the controller system for the air fryer product known as the *i*Fryer. The product comes with a 4 button interface and a 7-segment display panel. The system allows user to set the temperature for air frying from 100°C to 270°C, in 5°C increment/decrement, and setting the period of air frying in seconds, in second increment/decrement.

Two buttons, known as Up button and Down button, are used to increase and decrease the frying temperature or frying period. There is a button, known as Mode button, for the user to select either temperature setting or period setting. When the system starts, it is in temperature setting mode. A press (push and release) on the Mode button changes it to period setting mode. Another press will bring it back to original mode.

The last button, known as Start/Pause/Reset button, is use to start and stop the frying process. When the system starts, the fryer is in stop state. A press on it will start the frying. A short press after that will pause the process. A long press will reset the system. A long pressed is defined as press and hold for longer than 1 second. In reset state, the period is 0 second and the temperature is set to 100°C. Similarly, when the system is powered on, the system goes into reset state.

When the user has set the temperature and the period to fry and the Start button is pressed, the *i*Fryer will air fry until the period ends. It beeps for 5 seconds to indicate the frying has ended.

While frying the 7-segment panel displays the temperature for 2 seconds, then switch to display the period left for another 2 seconds. It will continue to do that till the frying ends.

The following is the API you can use to control/sense the systems:

typedef enum {START\_PAUSE, MODE, UP, DOWN} Button;

void startFrying()

void pauseFrying()

void resetSystem()

void setTemperature(int newTemp)

void setPeriod(int newPeriod)

void isButtonPressed(Button button);

void decreasePeriod()

void startBeeping();

void stopBeeping();

int getTime(); // in seconds