Supplementary Assignment

Model Based Embedded system design.

The model is based on a Digital watch that has been implemented In Simulink/State flow. As mentioned in the specification document there are three modes in the state chart namely Normal mode , Change time mode and alarm mode.

There is a parent chart called Digital clock where these three modes are implemented.

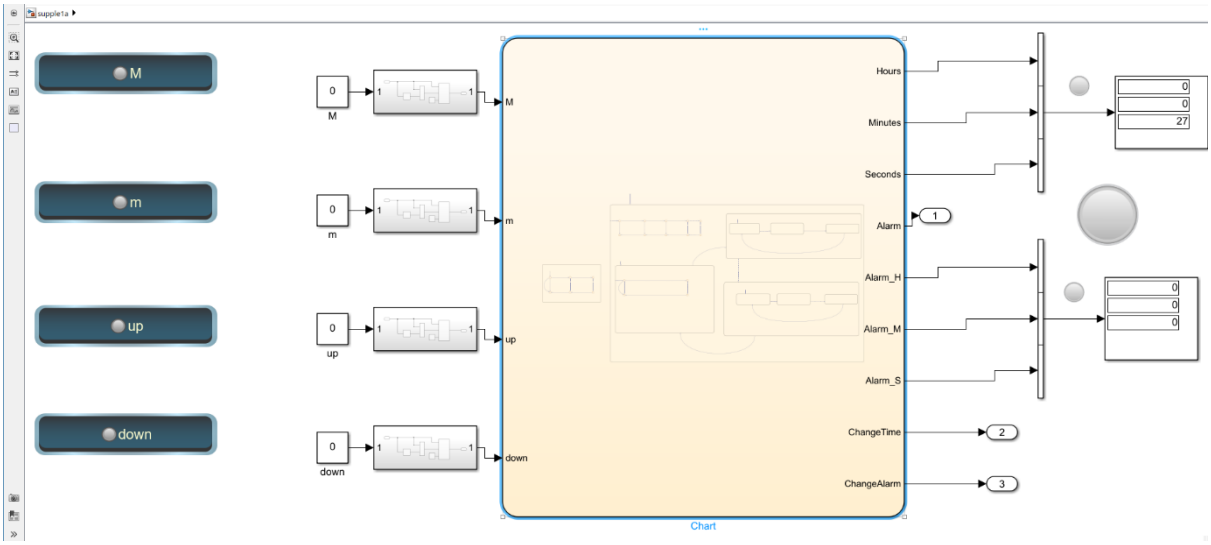
Implemented a function for button presses up and down.

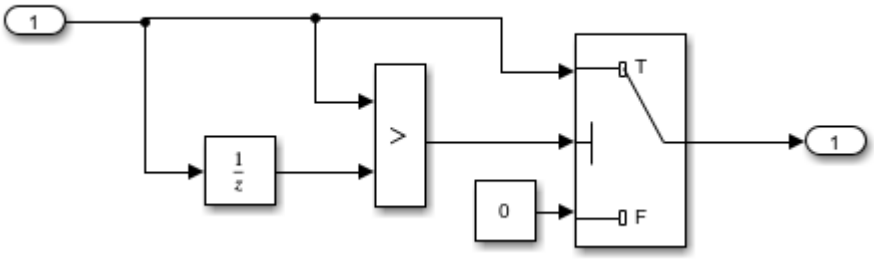
When M button is pressed the mode will cycle between the above mentioned three modes. When m button is pressed the clock cycles between hours, minutes and seconds in both time change mode and alarm mode.

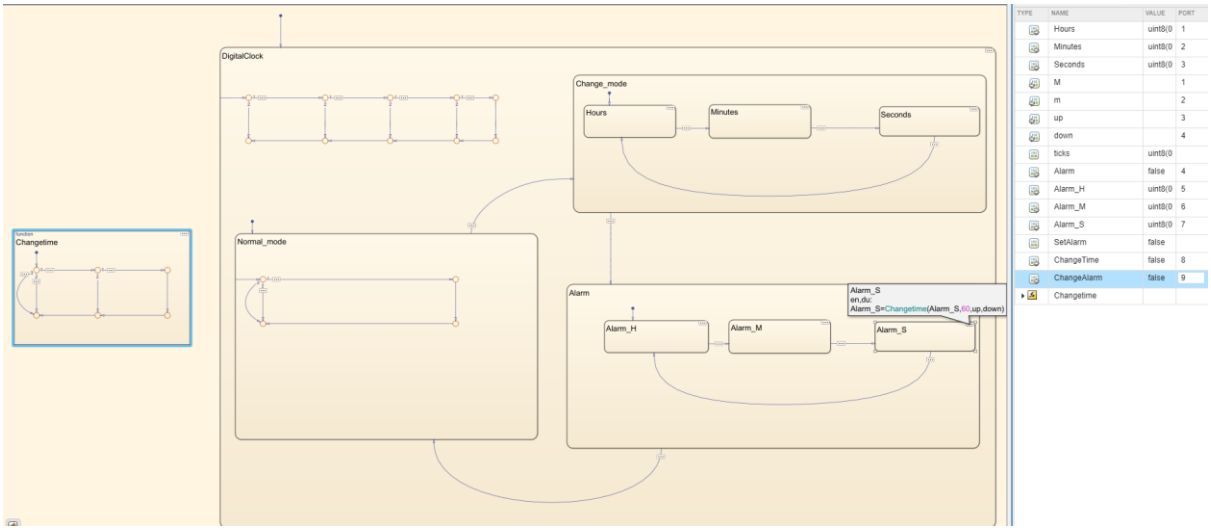
Implemented a flow chart when time is recorded as ticks. For every 10 ticks 1 second is increased. This helped in implementation of clock. Even when the mode is changed the clock will continue running. The up and down buttons are used to increase and decrease the minutes seconds and hours.

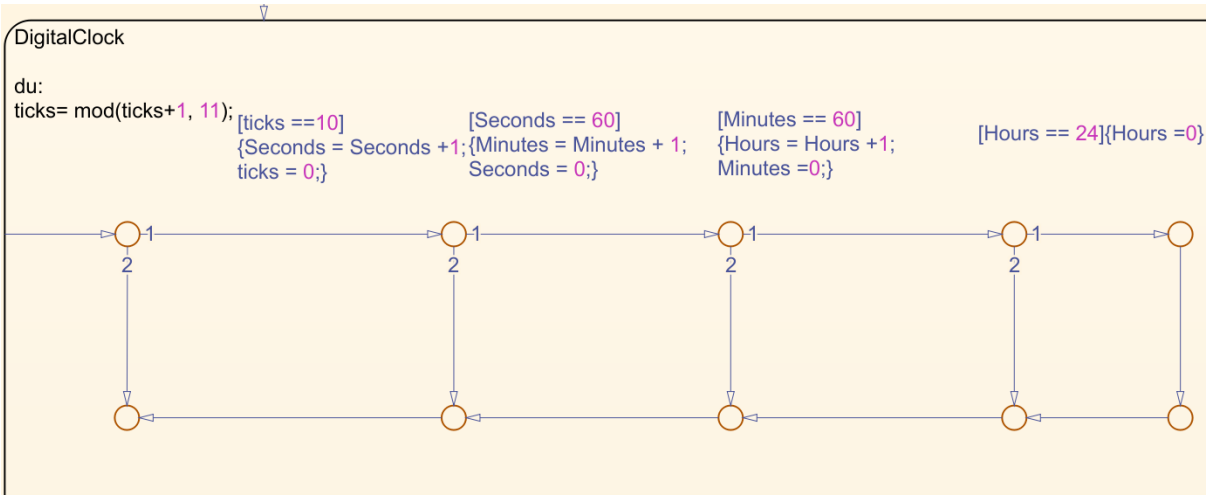
When alarm rings there is a led light which indicates alarm is on and will turn off if you press button m.

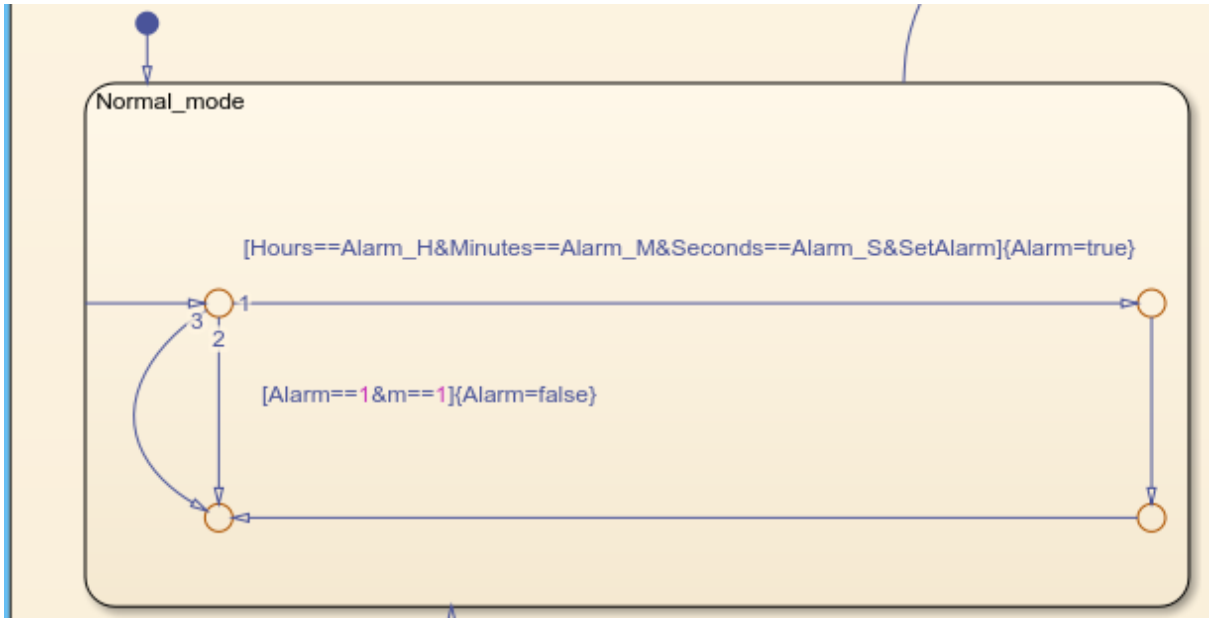
To give a valid input when the button is pressed, I used a circuit which compares the given value with previous value and if the given value is greater than the previous value then it identifies it as a button press and changes the modes and identified up and down buttons.

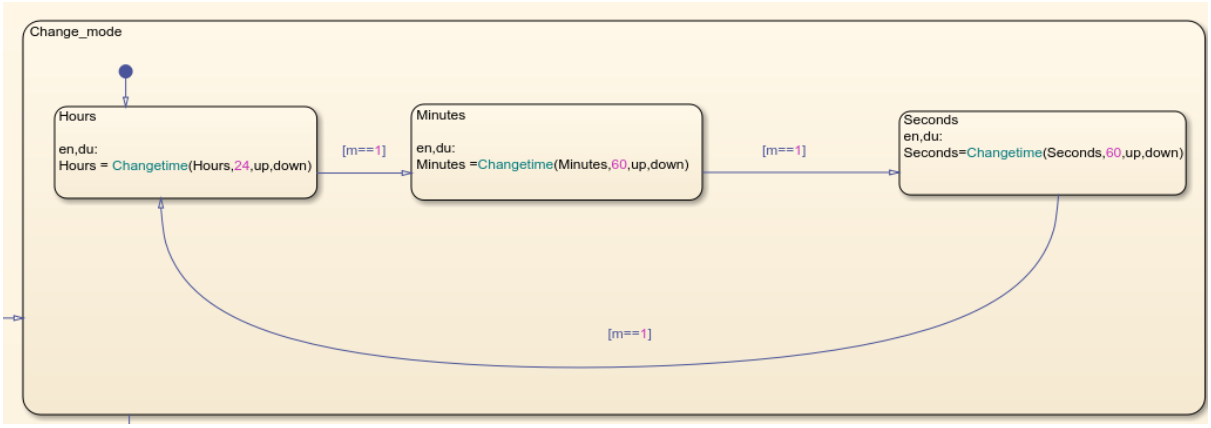


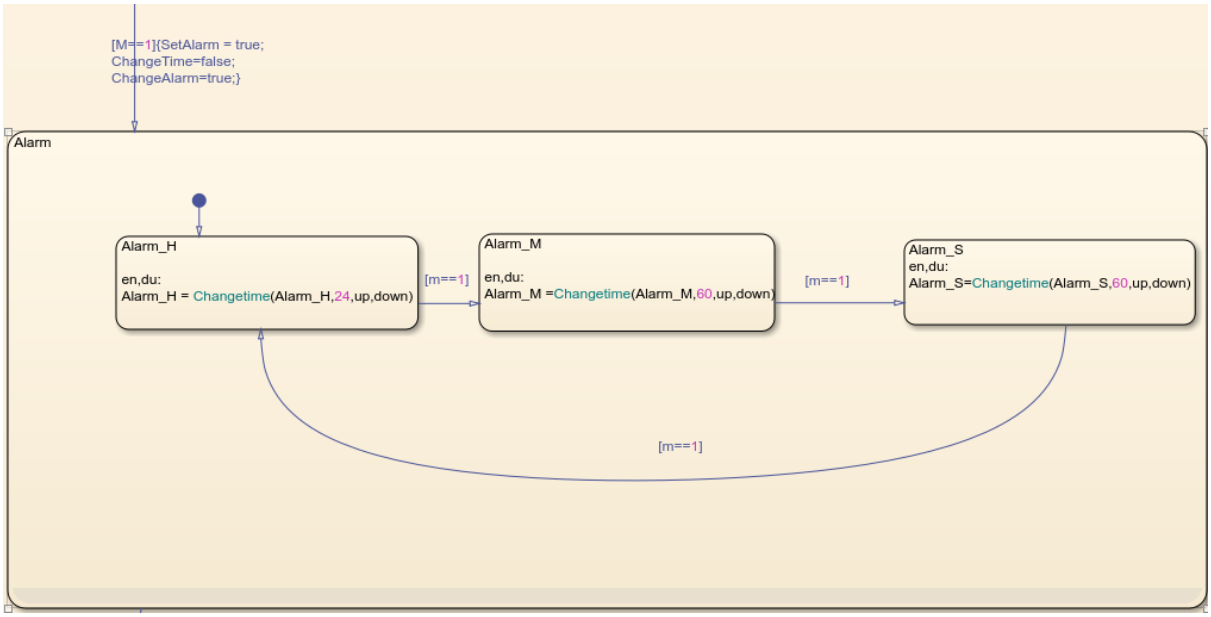

1
1
z
>
T
F
0
1

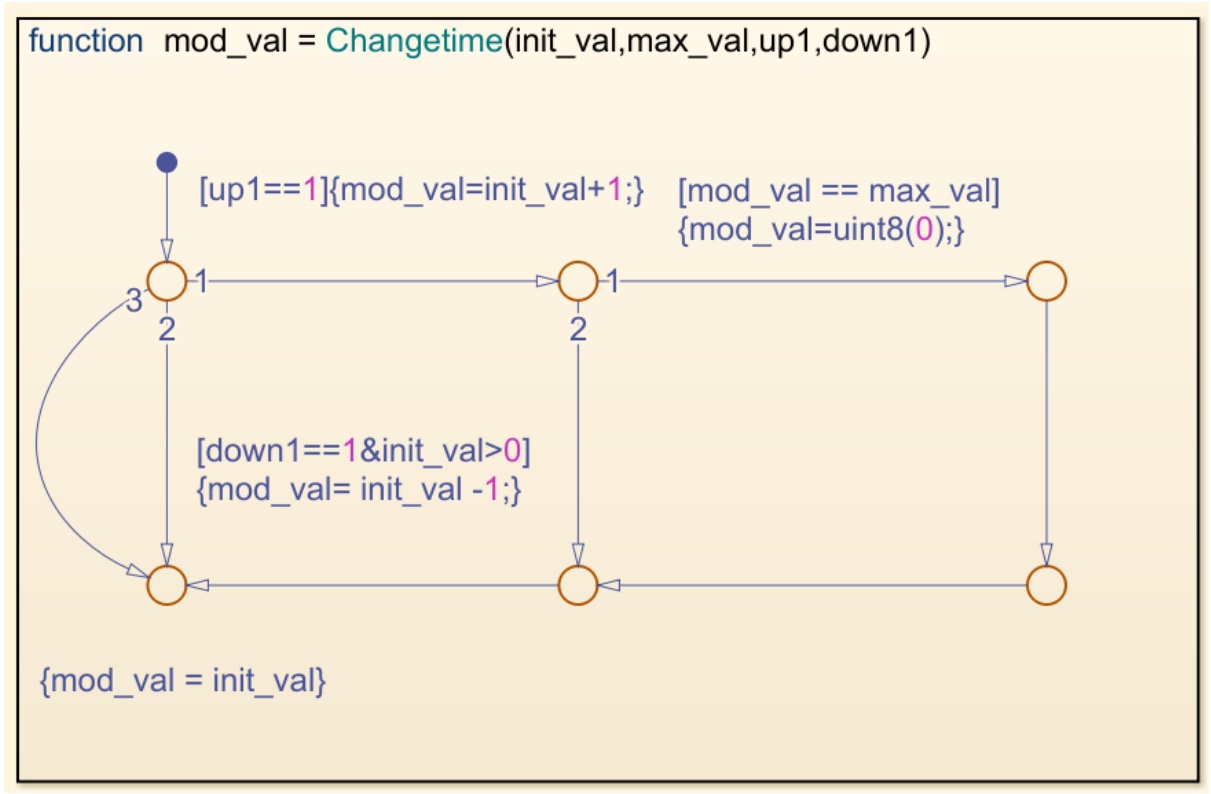

DigitalClock
Change_mode
Hours
Minutes
Seconds
Normal_mode
Alarm
Alarm_H
Alarm_M
Alarm_S
Changetime
TYPE
NAME
VALUE
PORT
Hours
Minutes
Seconds
M
m
up
down
ticks
Alarm
Alarm_H
Alarm_M
Alarm_S
SetAlarm
ChangeTime
ChangeAlarm
Changetime


DigitalClock
du:
ticks= mod(ticks+1, 11);
[ticks ==10]
{Seconds = Seconds +1;
ticks = 0;}
[Seconds == 60]
{Minutes = Minutes + 1;
Seconds = 0;}
[Minutes == 60]
{Hours = Hours +1;
Minutes =0;}
[Hours == 24]{Hours =0}
1
2


Normal_mode
[Hours==Alarm_H&Minutes==Alarm_M&Seconds==Alarm_S&SetAlarm]{Alarm=true}
1
2
3
[Alarm==1&m==1]{Alarm=false}


Change_mode
Hours
en,du:
Hours = Changetime(Hours,24,up,down)
[m==1]
Minutes
en,du:
Minutes =Changetime(Minutes,60,up,down)
[m==1]
Seconds
en,du:
Seconds=Changetime(Seconds,60,up,down)
[m==1]


[M==1]{SetAlarm = true;
ChangeTime=false;
ChangeAlarm=true;}
Alarm
Alarm_H
en,du:
Alarm_H = Changetime(Alarm_H,24,up,down)
[m==1]
Alarm_M
en,du:
Alarm_M =Changetime(Alarm_M,60,up,down)
[m==1]
Alarm_S
en,du:
Alarm_S=Changetime(Alarm_S,60,up,down)
[m==1]


function mod_val = Changetime(init_val,max_val,up1,down1)
[up1==1]{mod_val=init_val+1;}
[mod_val == max_val]
{mod_val=uint8(0);}
1
1
3
2
2
[down1==1&init_val>0]
{mod_val= init_val -1;}
{mod_val = init_val}