Mastering Embedded System Online Diploma

High-Pressure-Detector

Name: Loay Tamer Mohamed Shawky

GitHub: Loay tamer (GitHub)

1- Case Study:

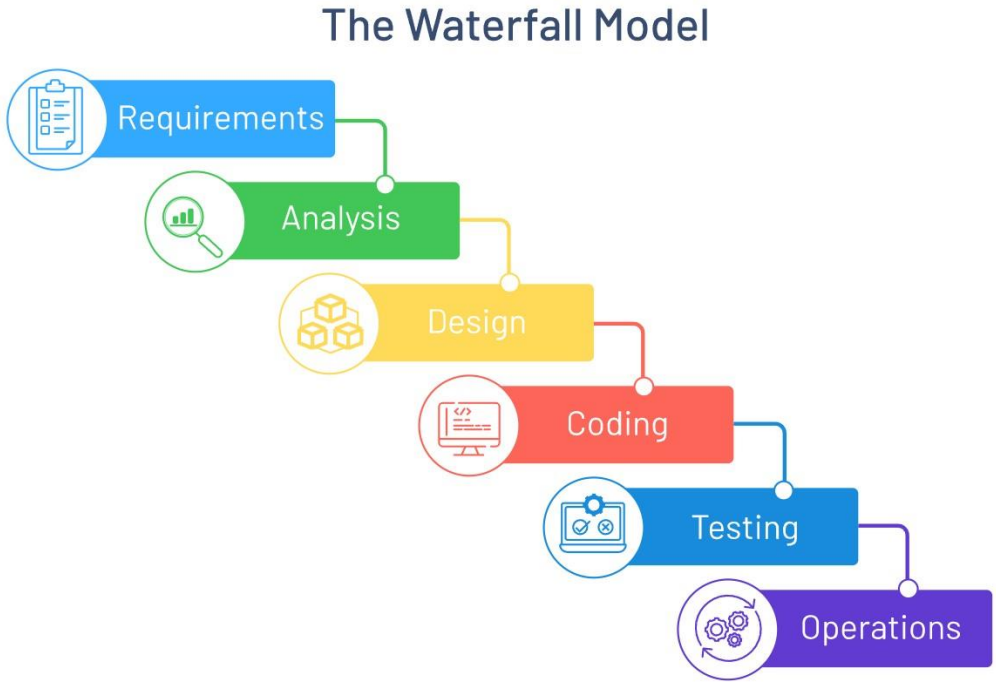
- A pressure controller informs the crew of a cabin with an alarm when the pressure exceeds 20 bars in the cabin.
- The alarm duration equals 60 seconds.

Assumptions:

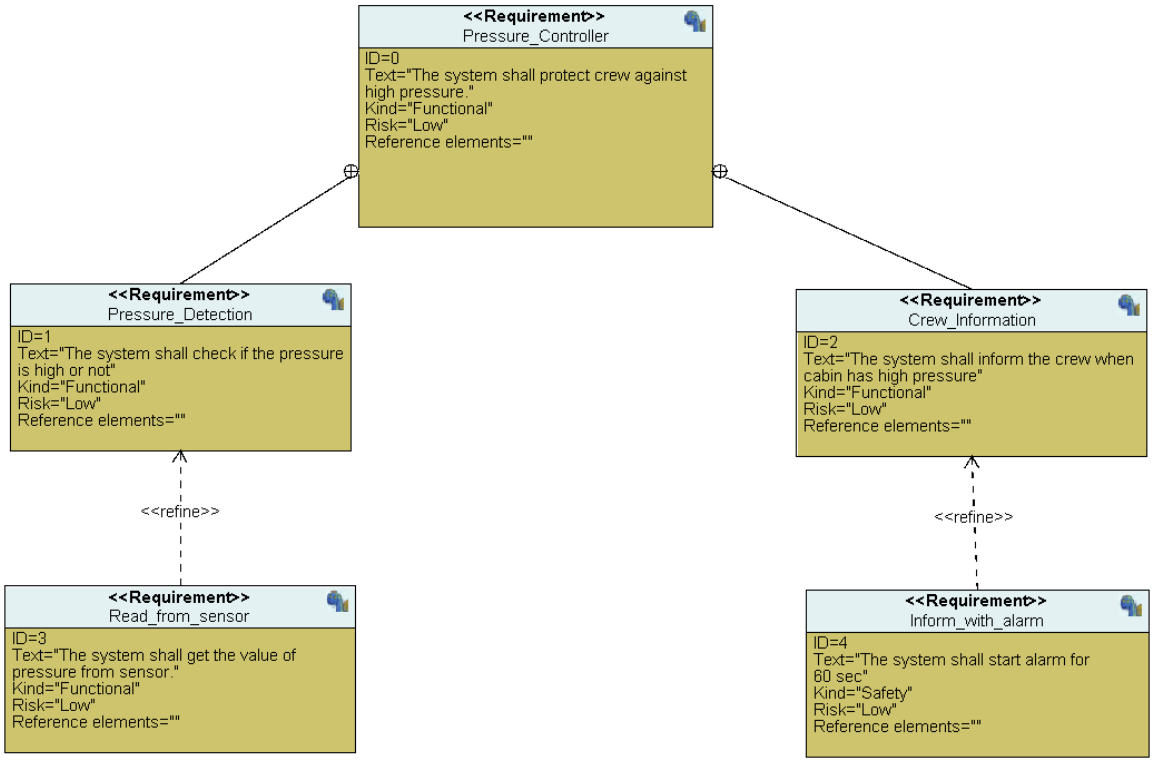
- The controller set up and shutdown procedures are not modeled.
- The controller maintenance is not modeled.
- The pressure sensor never fails.
- The alarm never fails.
- The controller never faces power cut.

2- Method:

- The system was designed using the Waterfall model.

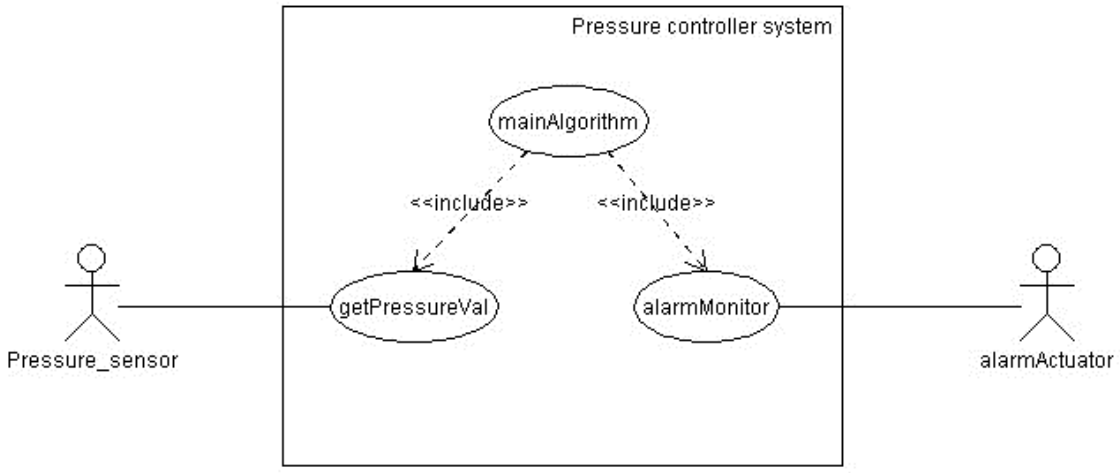


3- Requirements:

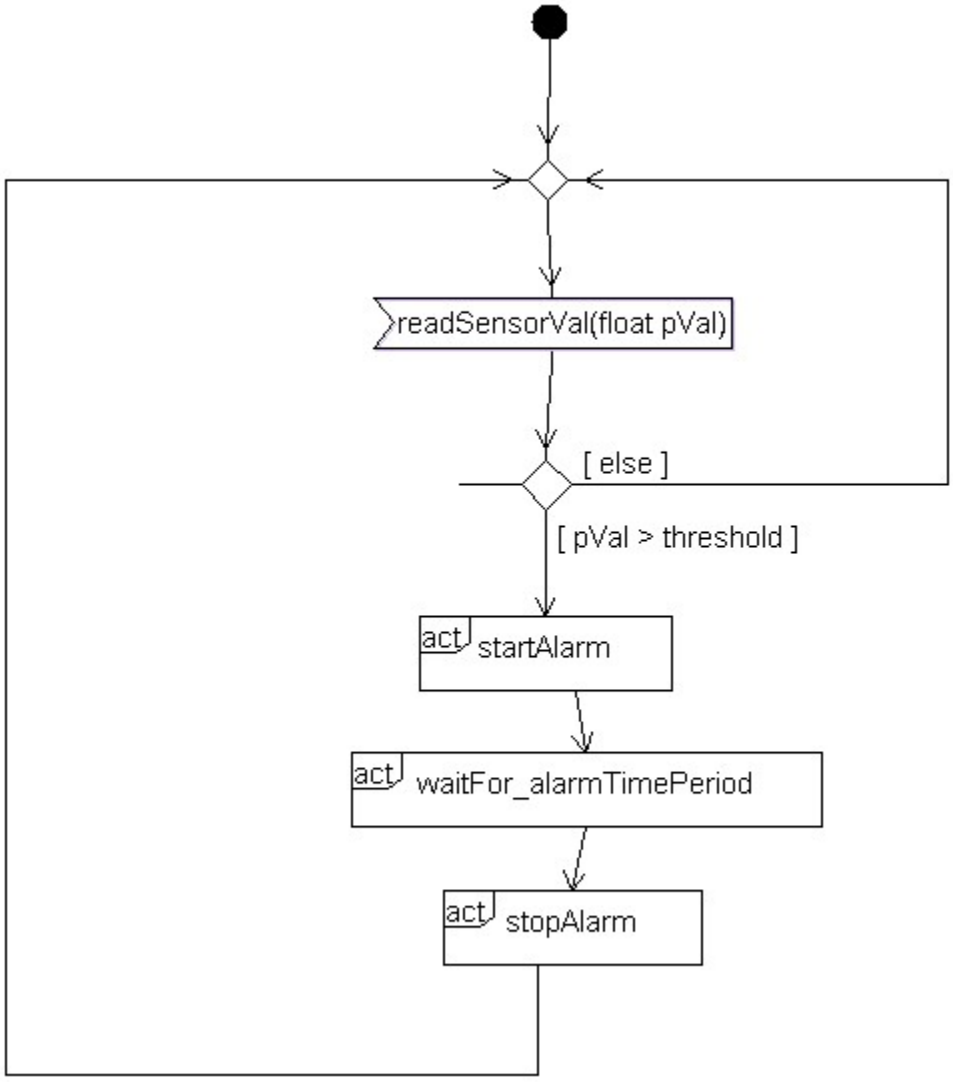


4- System Analysis:

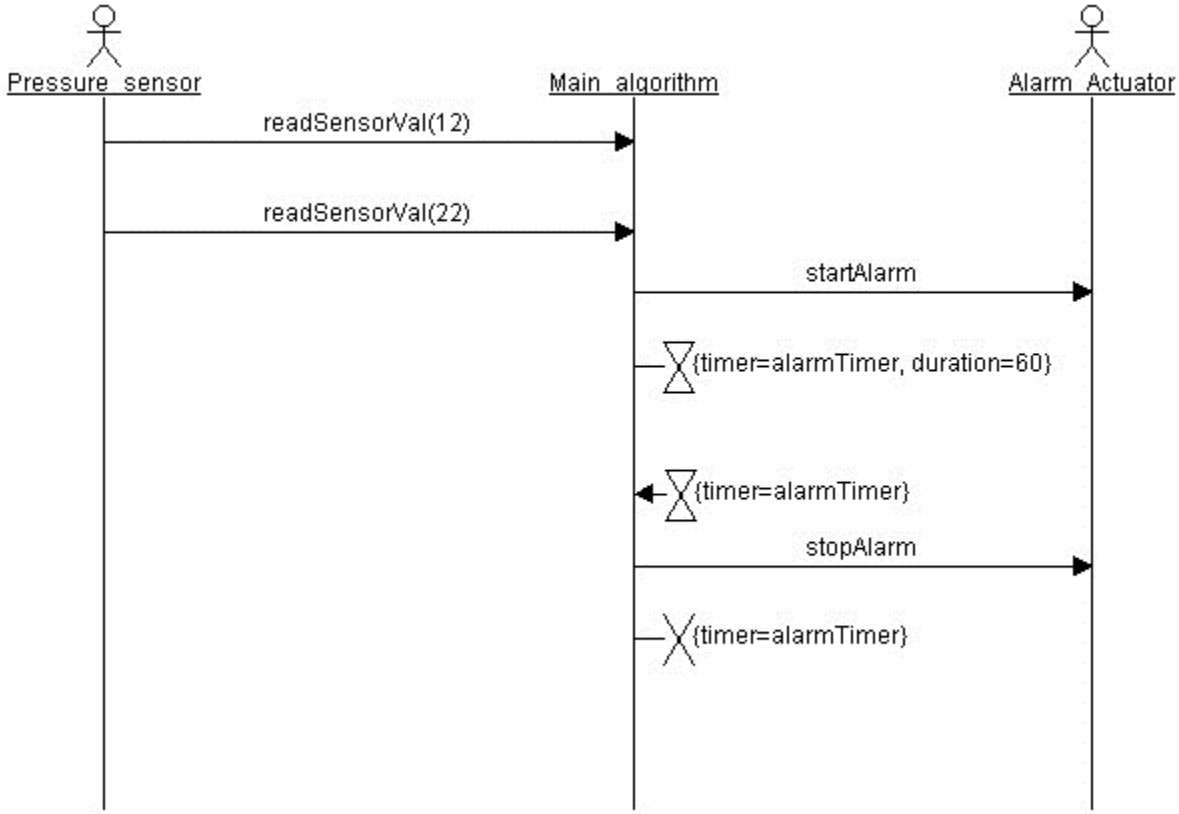
- Use case diagram:



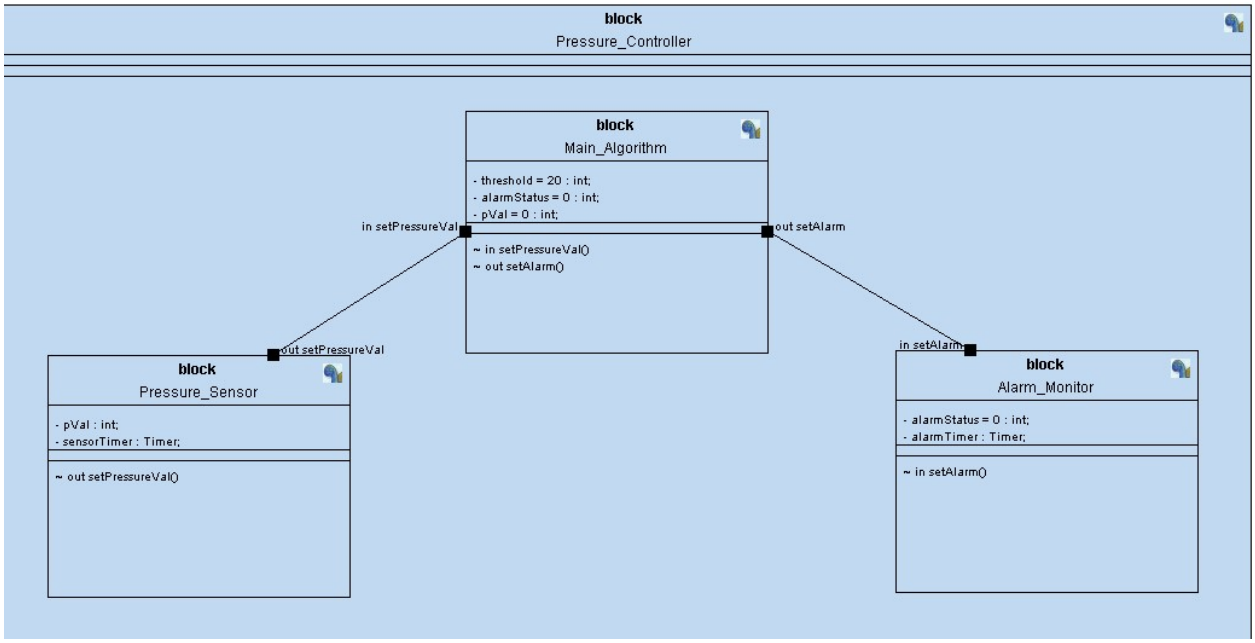
- Activity diagram:



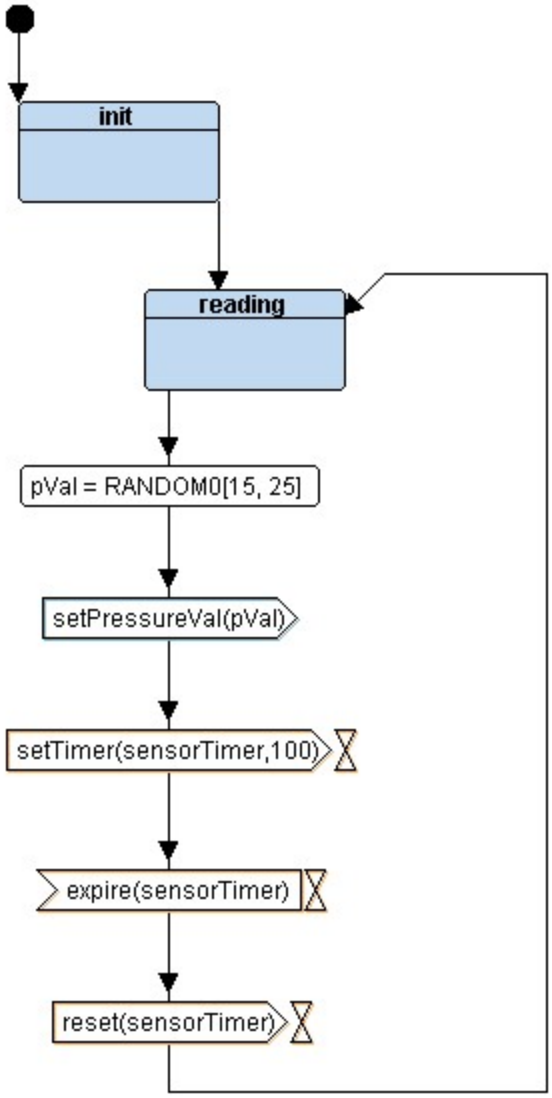
- Sequence diagram:



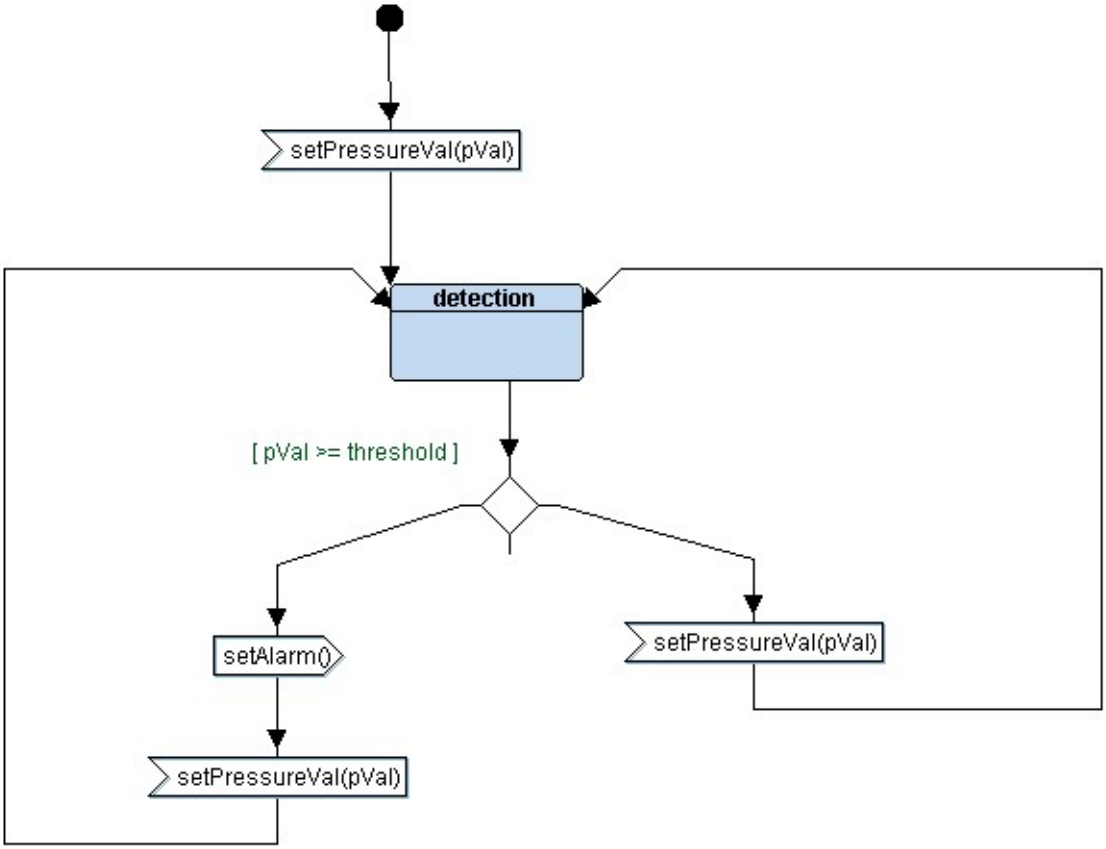
5- System Design:



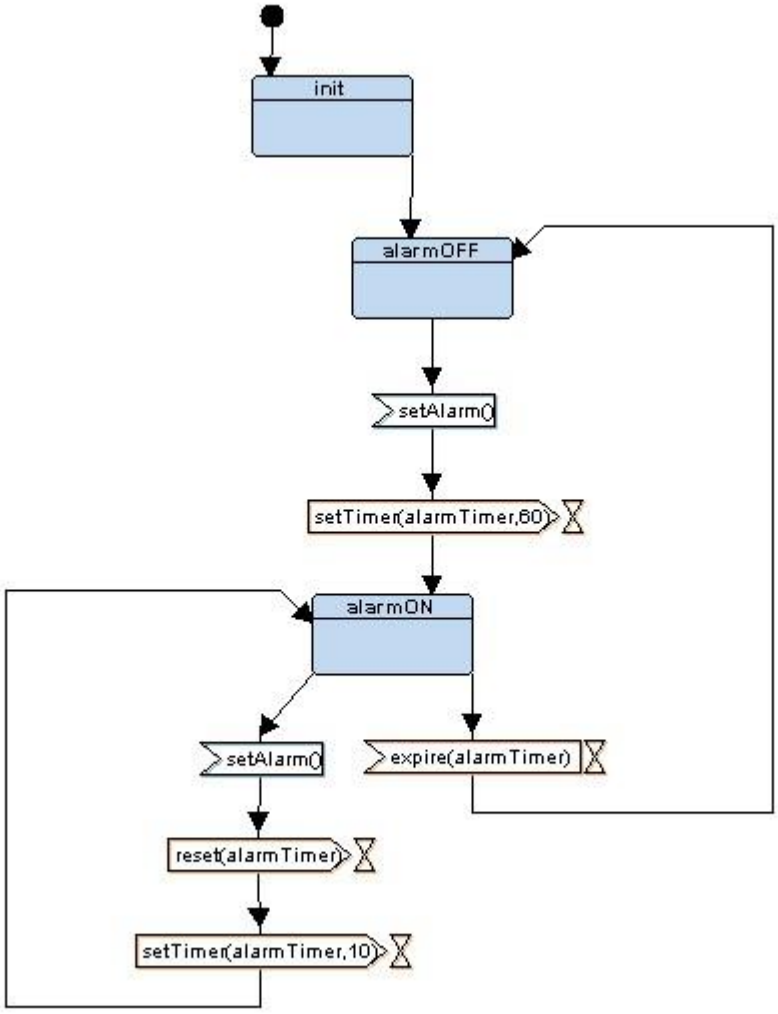
- Pressure sensor state machine:



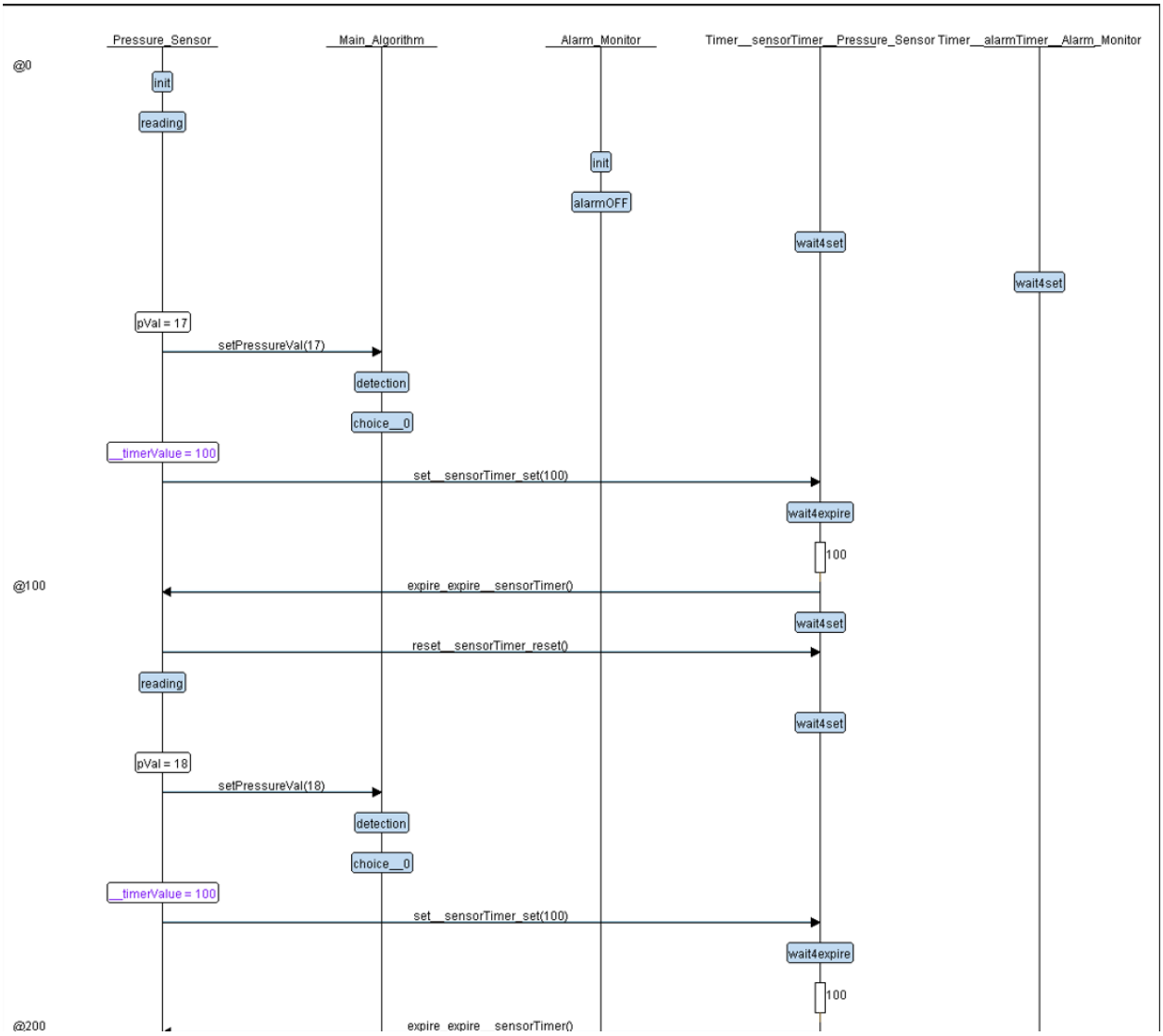
- Main Algorithm state machine:

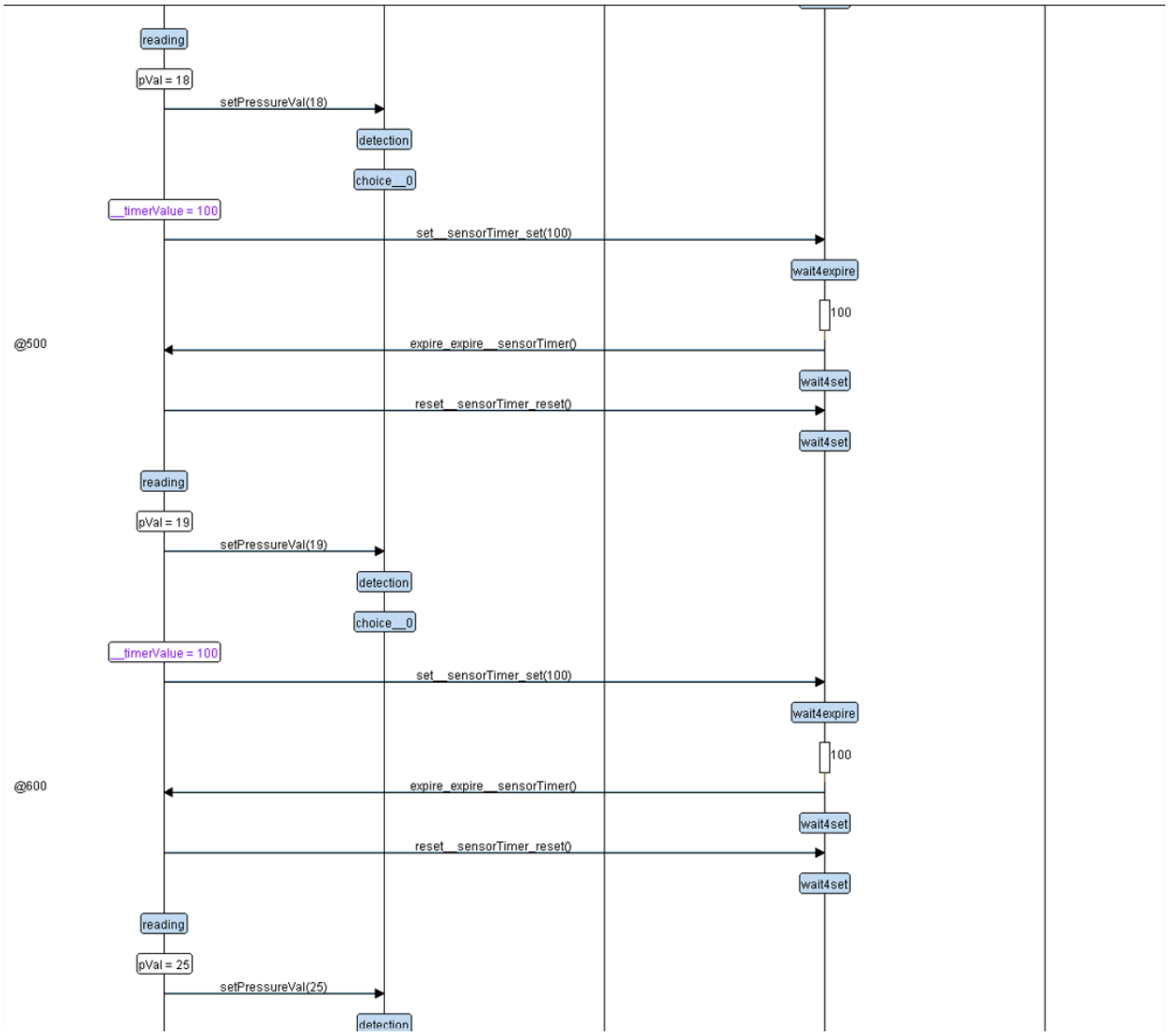


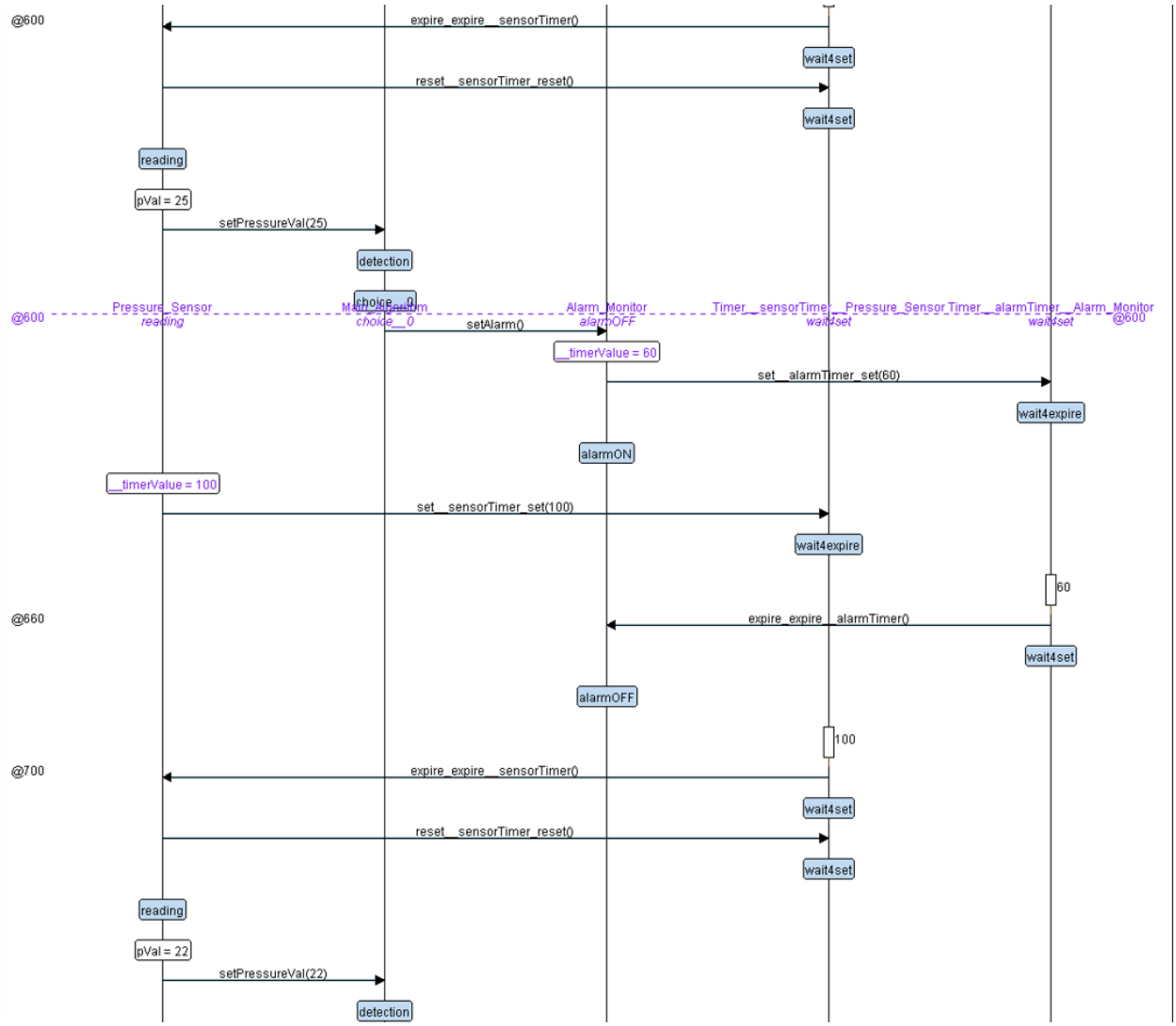
- Alarm Monitor state machine:



Verification:

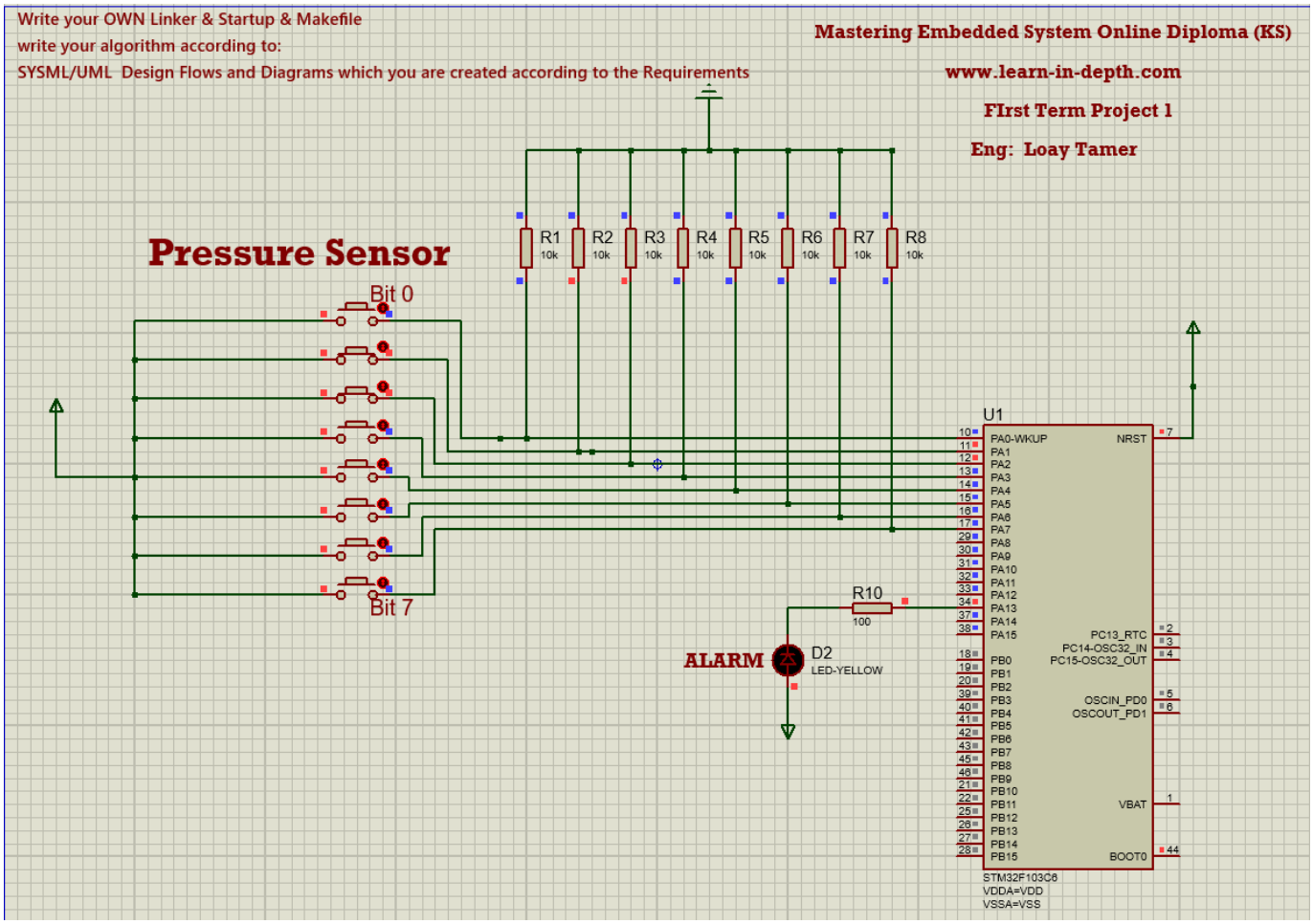



reading
pVal = 18
setPressureVal(18)
detection
choice__0
__timerValue = 100
set__sensorTimer_set(100)
wait4expire
100
@500
expire_expire__sensorTimer()
wait4set
reset__sensorTimer_reset()
wait4set
reading
pVal = 19
setPressureVal(19)
detection
choice__0
__timerValue = 100
set__sensorTimer_set(100)
wait4expire
100
@600
expire_expire__sensorTimer()
wait4set
reset__sensorTimer_reset()
wait4set
reading
pVal = 25
setPressureVal(25)
detection

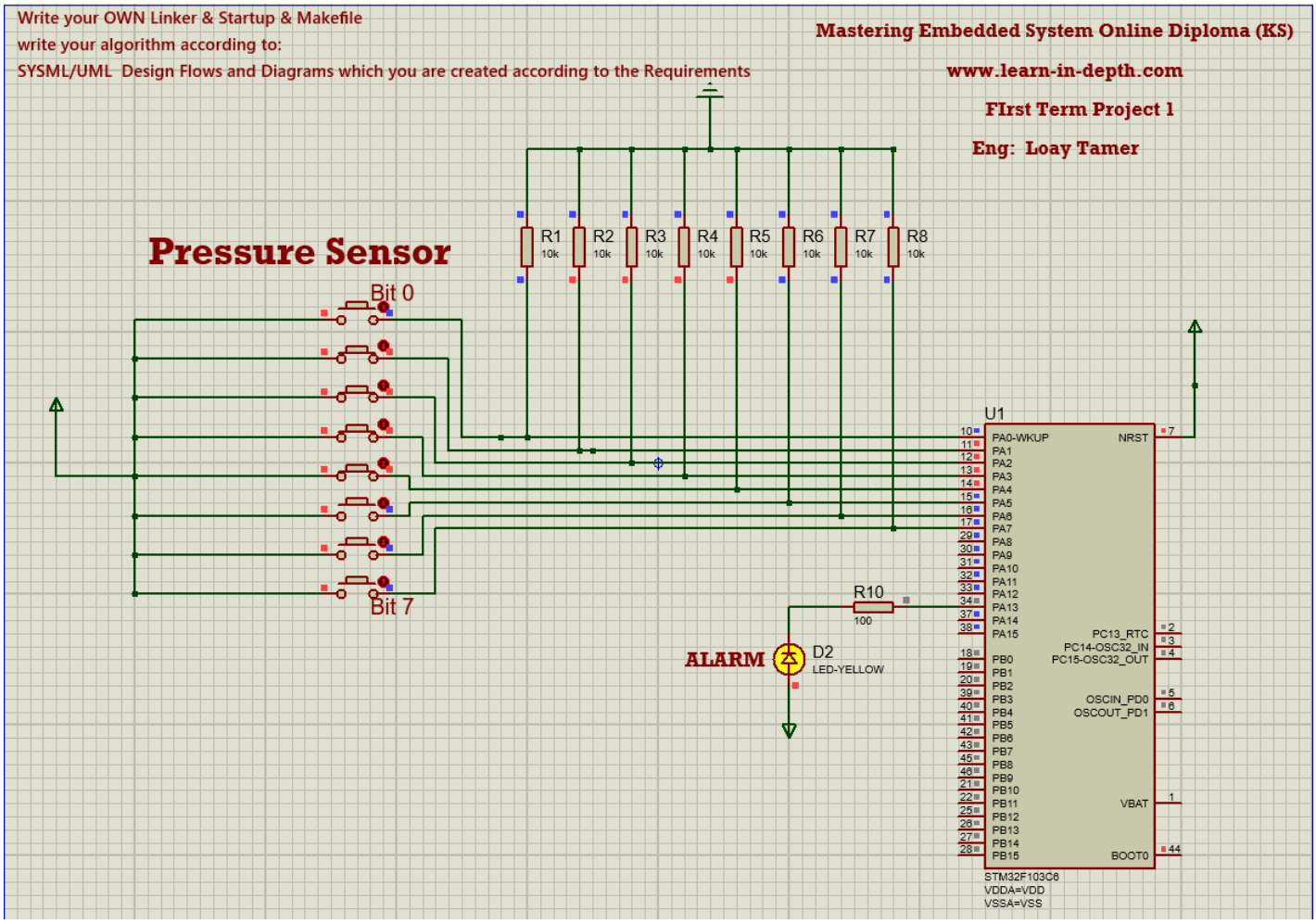

@600
expire_expire__sensorTimer()
wait4set
reset__sensorTimer_reset()
wait4set
reading
pVal = 25
setPressureVal(25)
detection
choice__0
@600
Pressure_Sensor
reading
Main_Program
choice__0
Alarm_Monitor
alarmOFF
Timer__sensorTimer
wait4set
Pressure_Sensor Timer__alarmTimer
wait4set
Alarm_Monitor
@600
setAlarm()
__timerValue = 60
set__alarmTimer_set(60)
wait4expire
alarmON
__timerValue = 100
set__sensorTimer_set(100)
wait4expire
60
@660
expire_expire__alarmTimer()
wait4set
alarmOFF
100
@700
expire_expire__sensorTimer()
wait4set
reset__sensorTimer_reset()
wait4set
reading
pVal = 22
setPressureVal(22)
detection

Proteus simulation:

- Case 1: pressure value = 6 < threshold



- Case 2: pressure value = 30 < threshold



Symbols:

```
$ make symbols
arm-none-eabi-nm.exe *.o

alarm_monitor.o:
000000a0 T alarm_monitor_init
00000004 C alarm_monitor_state
00000001 C alarm_monitor_state_id
00000000 B alarm_state
00000080 T setAlarm
0000002c T ST_alarm_OFF
00000000 T ST_alarm_ON

driver.o:
00000000 T Delay
00000024 T getPressureVal
0000008c T GPIO_INITIALIZATION
0000003c T Set_Alarm_actuator
```

```
main.o:
         U alarm_monitor_state
00000001 C alarm_monitor_state_id
         U Delay
         U getPressureVal
         U GPIO_INITIALIZATION
00000040 T main
         U pController_state
00000001 C pressure_controller_state_id
         U pSensor_state
00000001 C pSensor_state_id
         U Set_Alarm_actuator
         U setPressureVal
00000000 T setup
         U ST_alarm_OFF
         U ST_pressure_detection
         U ST_pSensor_reading

pressure_controller.o:
00000000 B pc_pVal
00000004 C pController_state
00000001 C pressure_controller_state_id
         U return_pVal
         U setAlarm
00000000 T ST_pressure_detection
00000000 D threshold
```

```
pressure_sensor.o:
00000080 T pressure_sensor_init
00000004 C pSensor_state
00000001 C pSensor_state_id
00000000 B pVal
00000068 T return_pVal
00000004 B sensor_timer
00000048 T setPressureVal
00000000 T ST_pSensor_reading

startup.o:
         U _E_bss
         U _E_data
         U _E_text
         U _S_bss
         U _S_data
         U _stack_top
000000b0 W Bus_Fault
000000b0 T Default_Handler
000000b0 W H_fault_Handler
         U main
000000b0 W MM_fault_Handler
000000b0 W NMI_Handler
00000000 T Reset_Handler
000000b0 W Usage_Fault_Handler
00000000 D vectors
```