Mastering Embedded System Online Diploma

www.learn-in-depth.com First Term (Final Project 1)

Eng. Hossam El-Dien Adel

My Profile: https://www.learn-in-depth.com/hossam.adel2024@gmail.com

Case Study:

A client expects you to deliver the software of the following system

Specification (from the client):

-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.

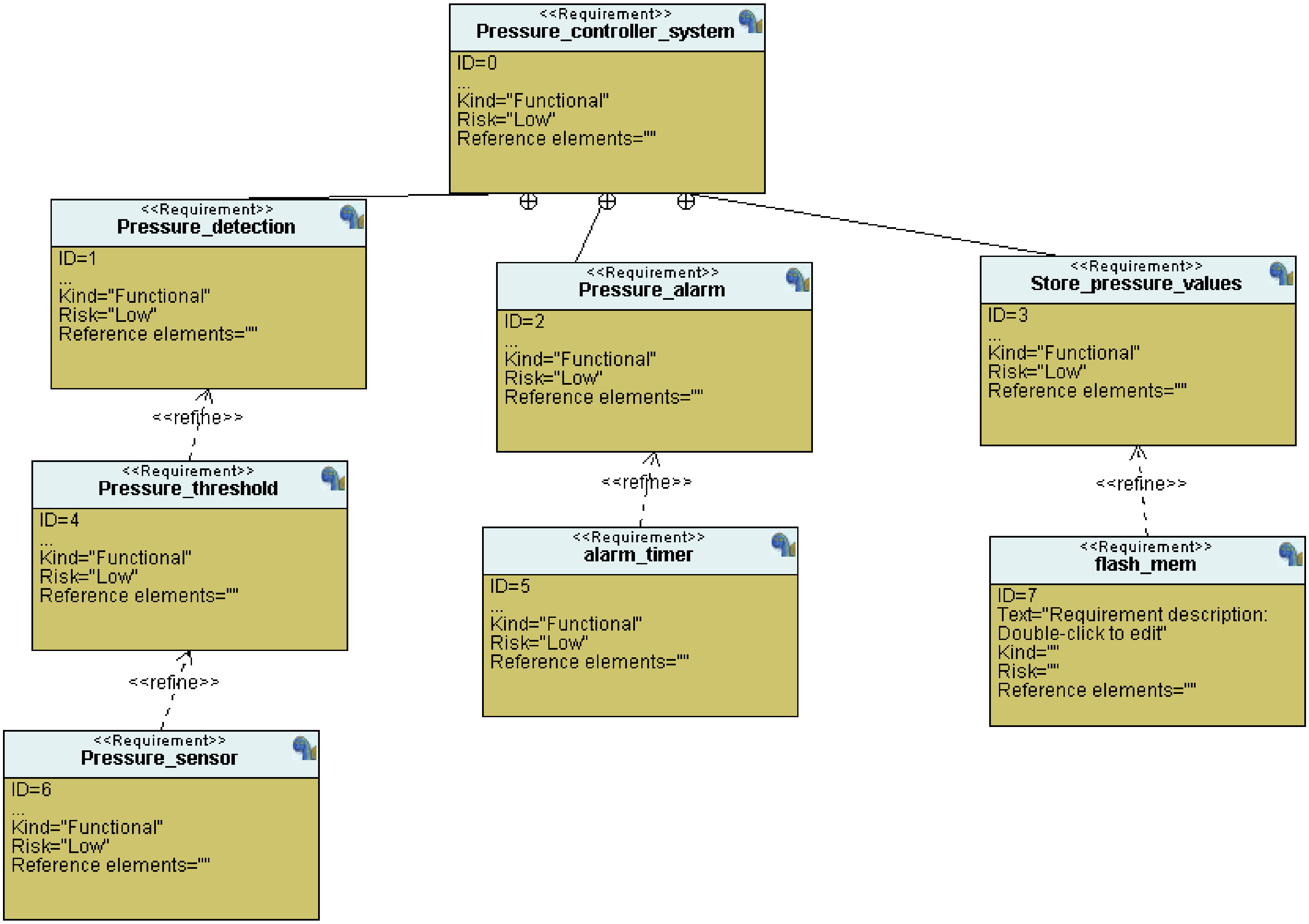
The controller maintenance is not modeled.

The pressure sensor never fails.

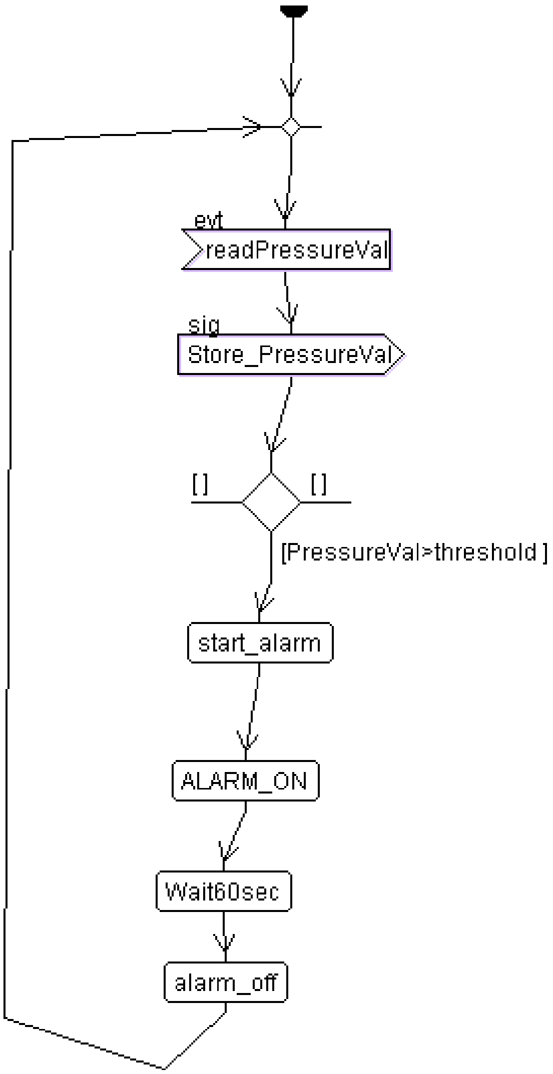
The alarm never fails.

The controller never faces power cut.

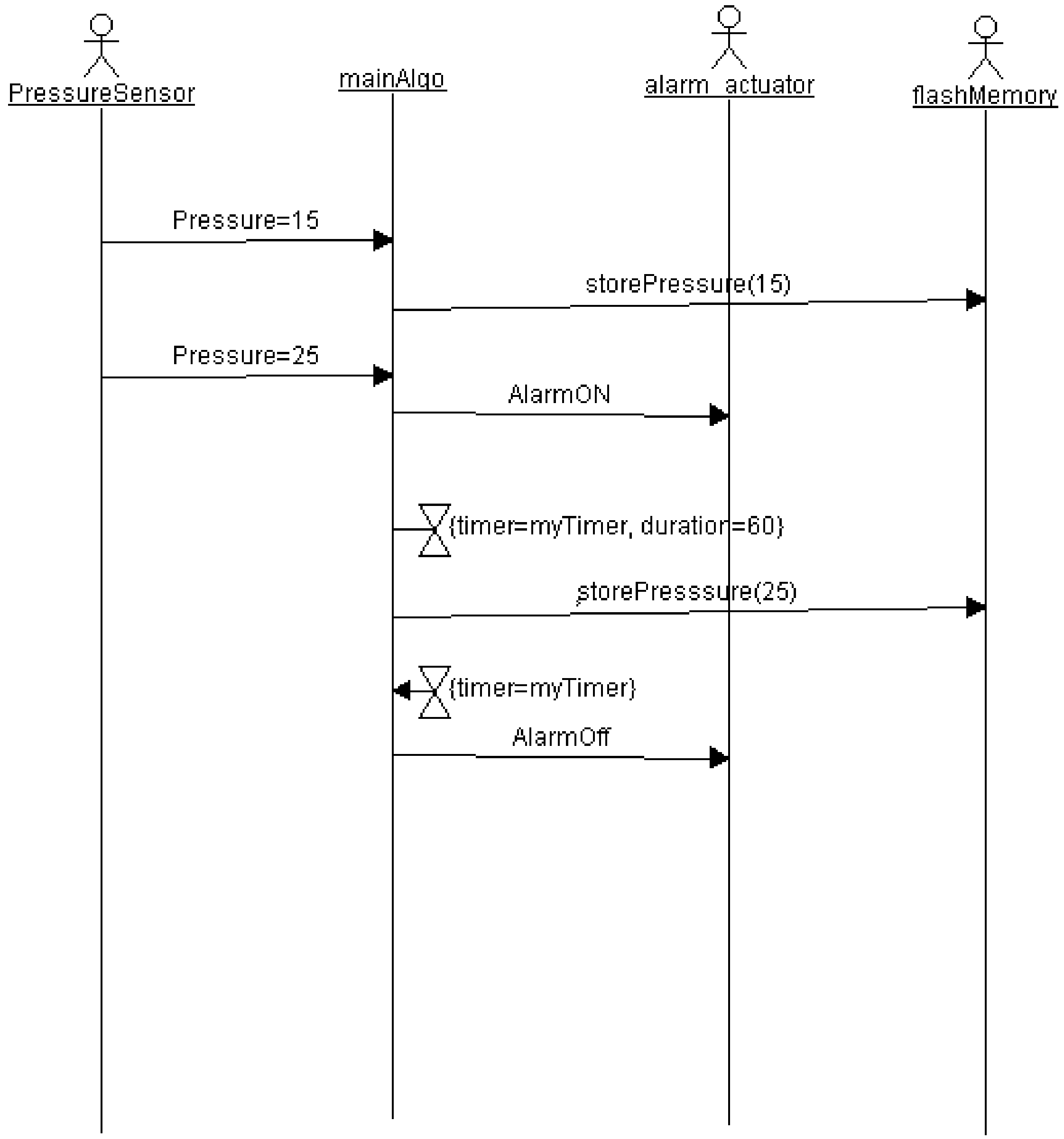
Requirements Diagram:



Activity Diagram:

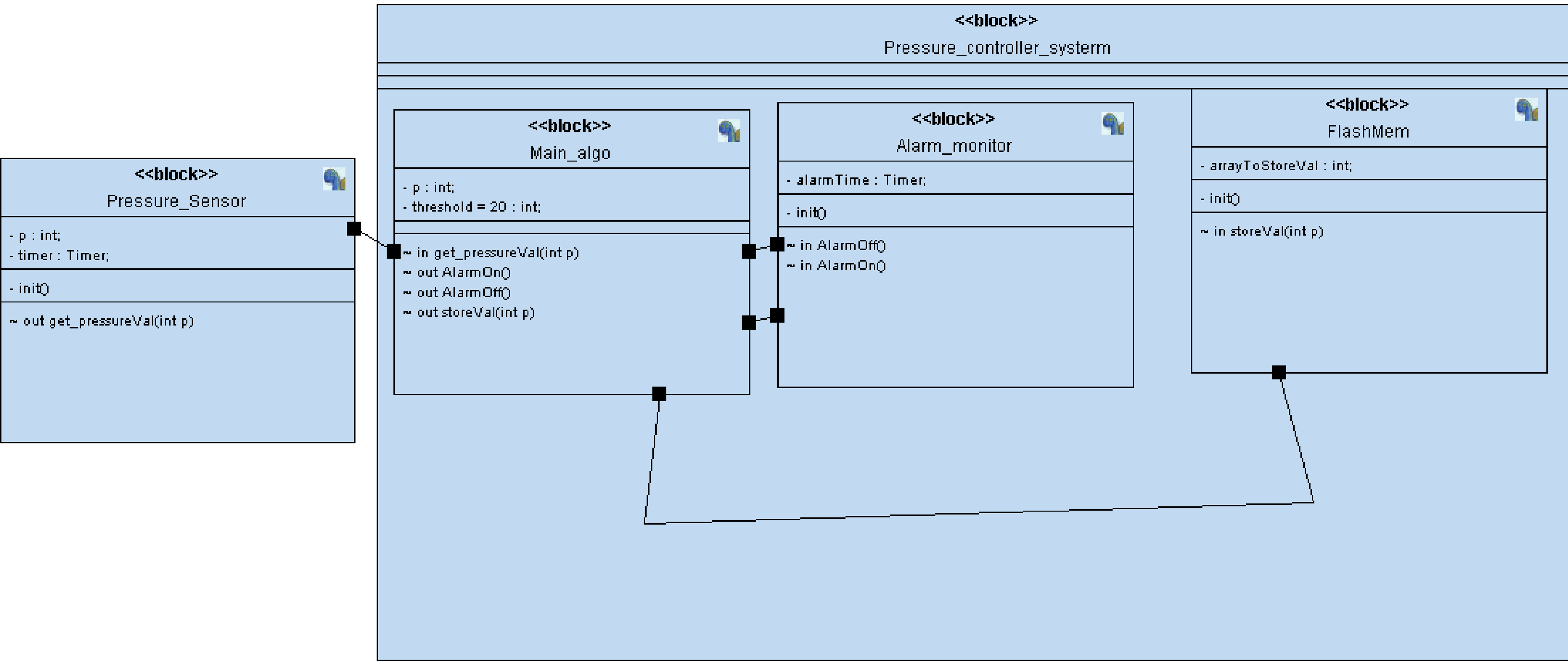


Sequence Diagram:

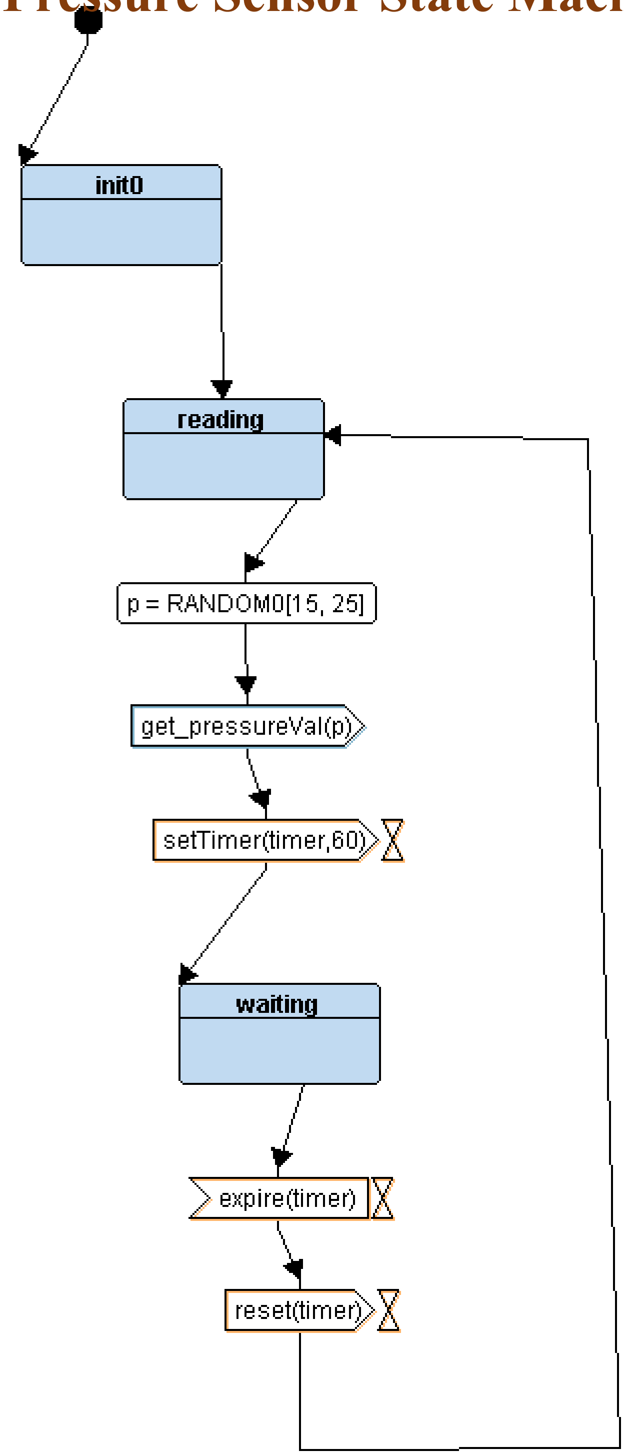


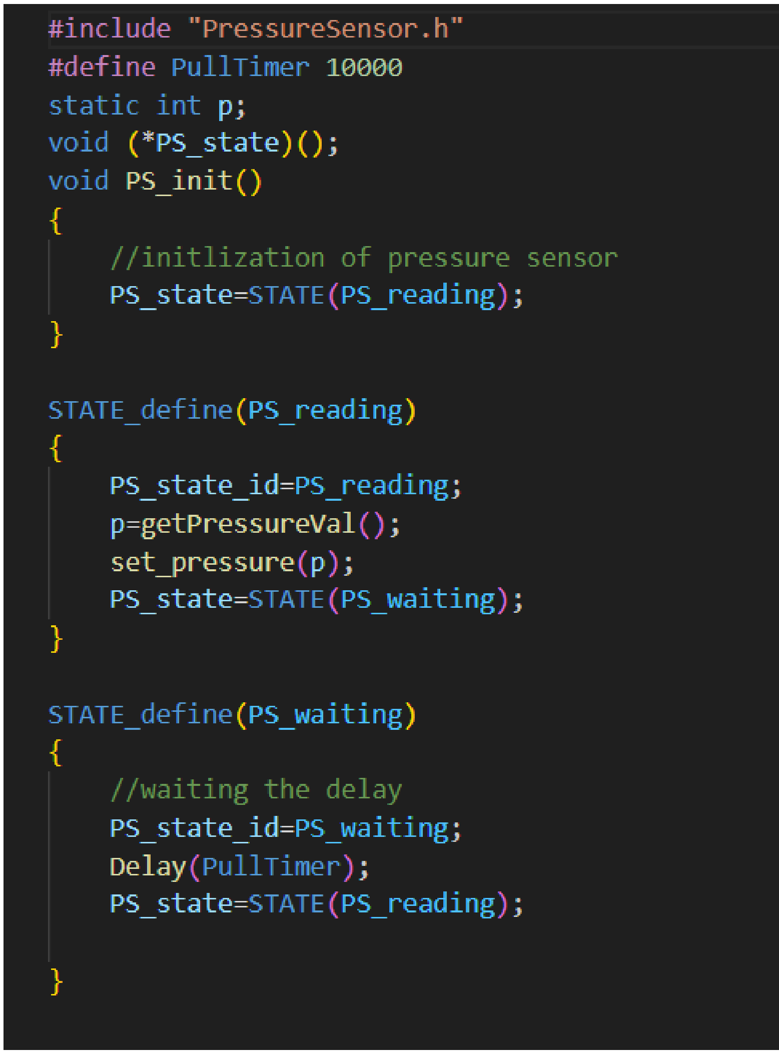
System Design:

Pressure Controller design:



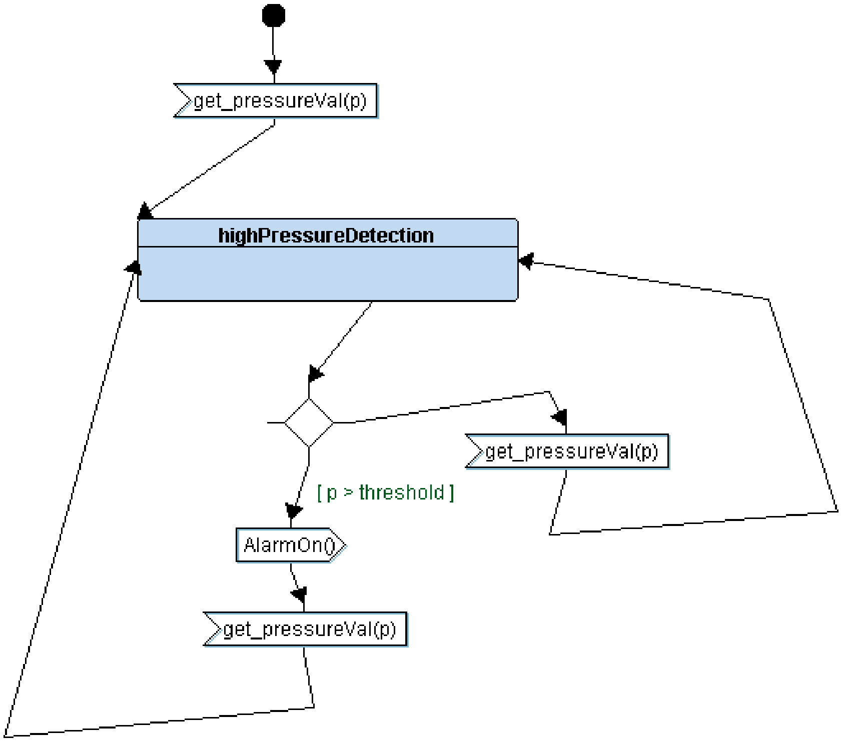
Pressure Sensor State Machine:

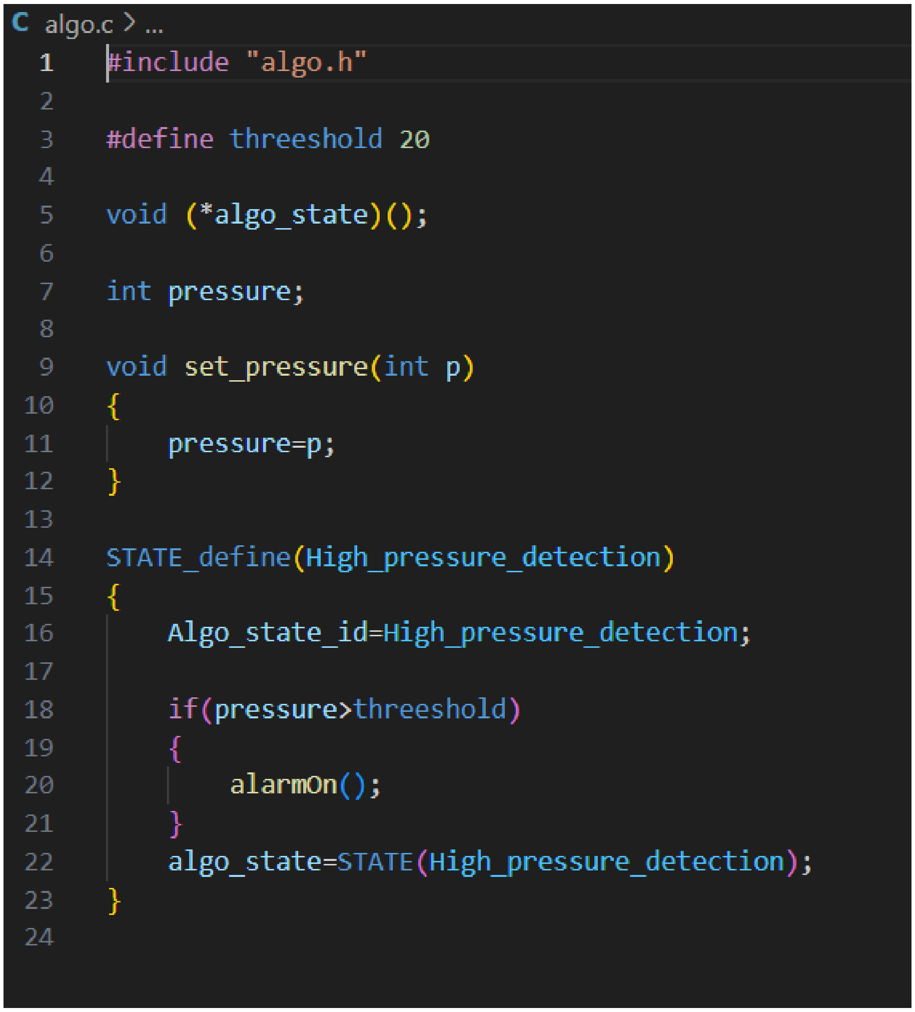




```
#include "PressureSensor.h"
#define PullTimer 10000
static int p;
void (*PS_state)();
void PS_init()
{
    //initlization of pressure sensor
    PS_state=STATE(PS_reading);
}

STATE_define(PS_reading)
{
    PS_state_id=PS_reading;
    p=getPressureVal();
    set_pressure(p);
    PS_state=STATE(PS_waiting);
}

STATE_define(PS_waiting)
{
    //waiting the delay
    PS_state_id=PS_waiting;
    Delay(PullTimer);
    PS_state=STATE(PS_reading);

}
```

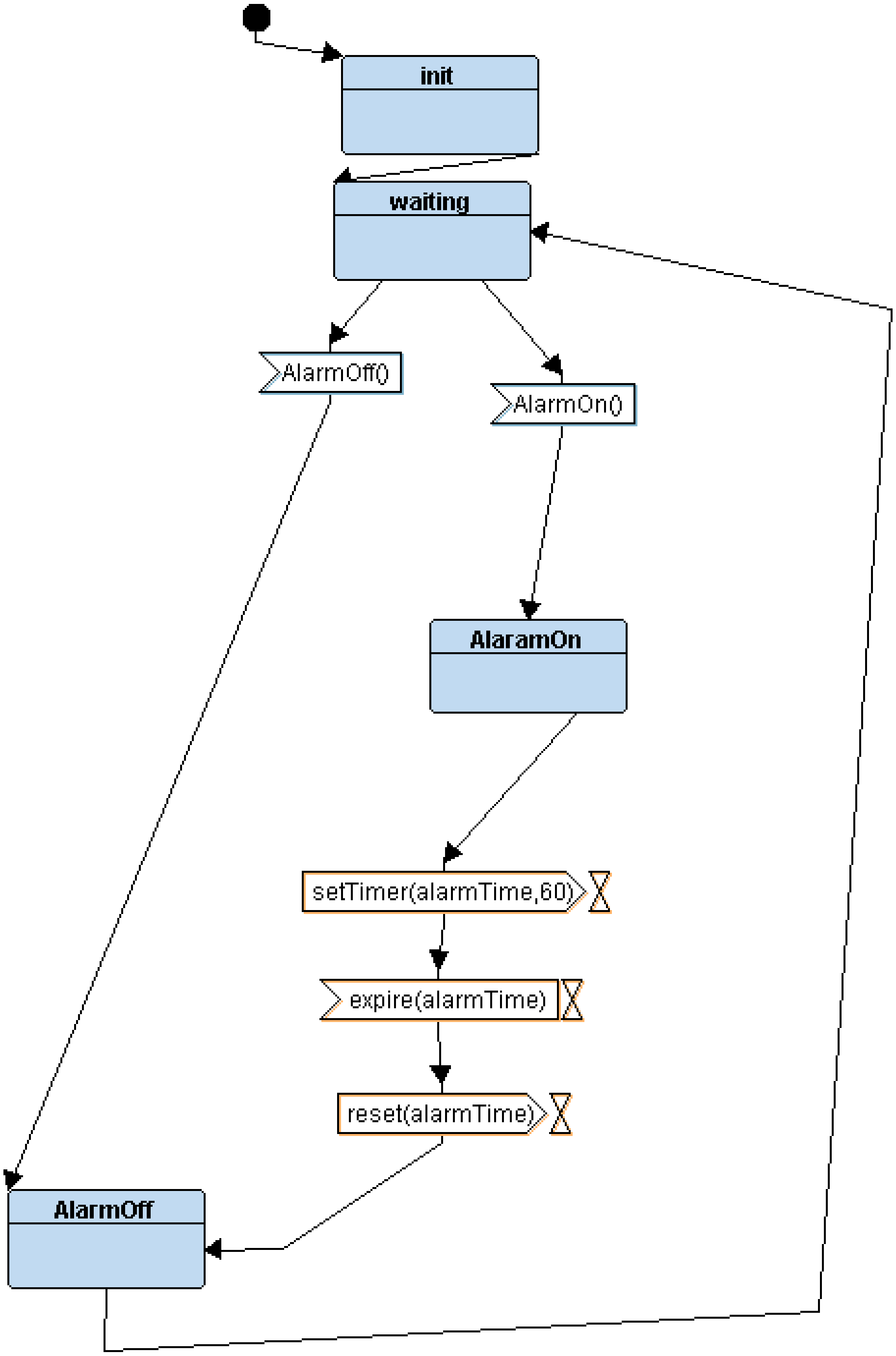
MainAlgo State Machine:

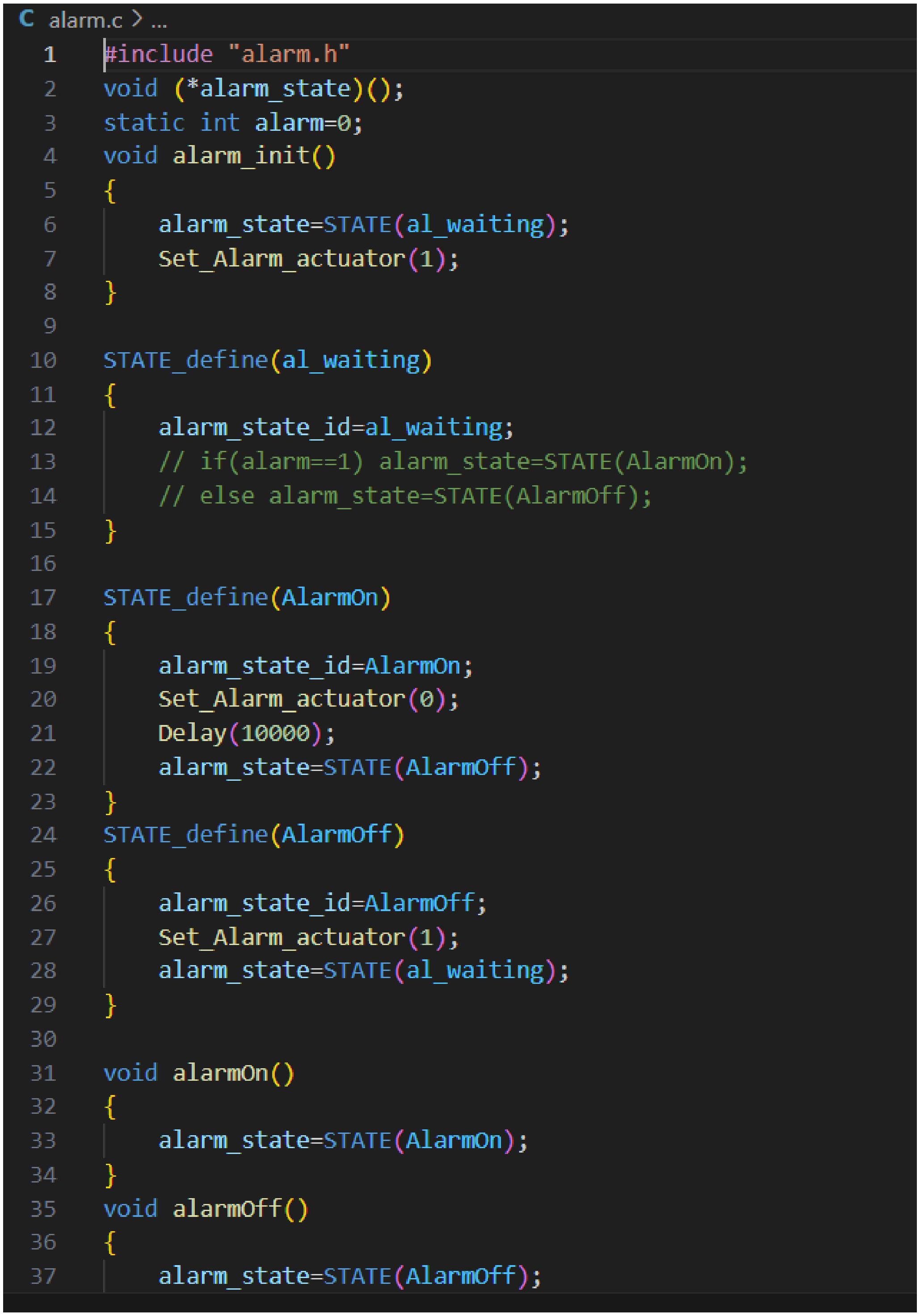




```c
#include "algo.h"

#define threeshold 20

void (*algo_state)();

int pressure;

void set_pressure(int p)
{
    pressure=p;
}

STATE_define(High_pressure_detection)
{
    Algo_state_id=High_pressure_detection;

    if(pressure>threeshold)
    {
        alarmOn();
    }
    algo_state=STATE(High_pressure_detection);
}

```

Alarm Monitor State Machine:

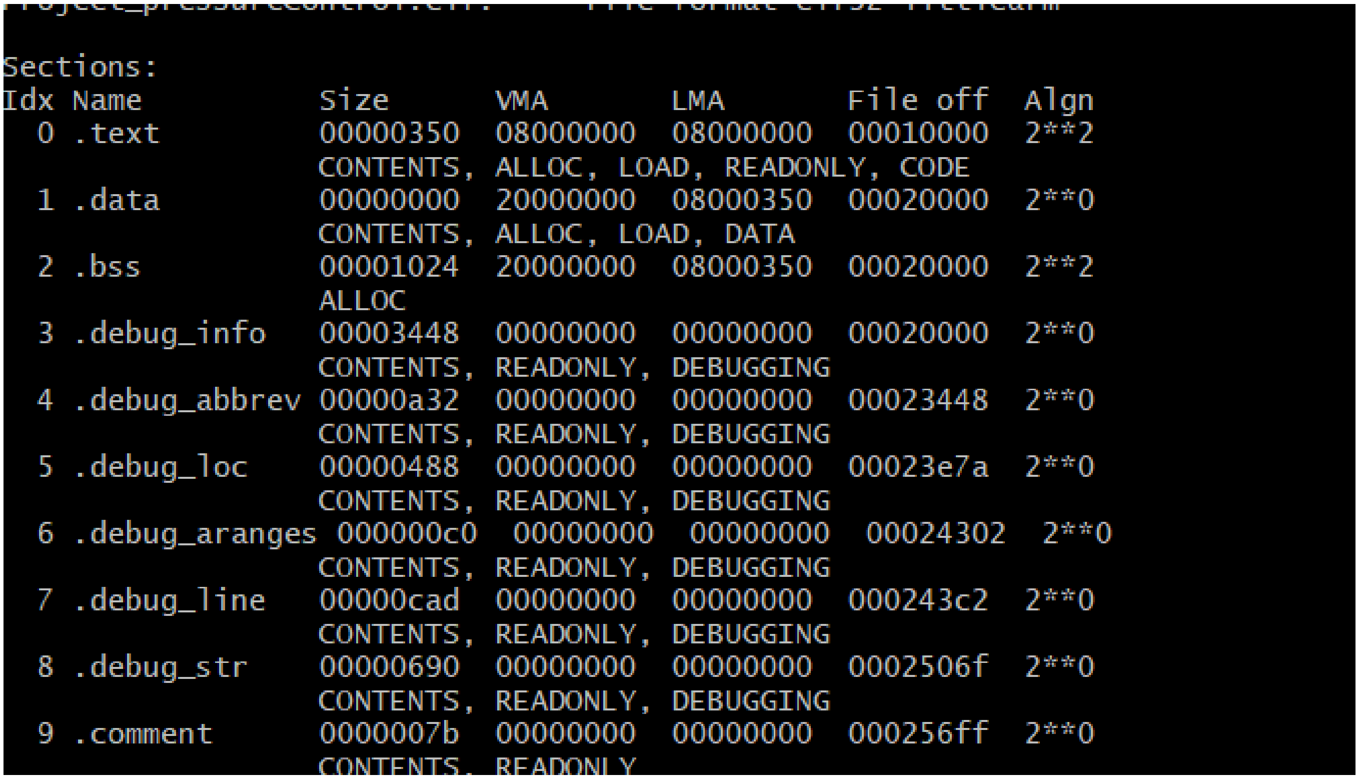




```c
#include "alarm.h"
void (*alarm_state)();
static int alarm=0;
void alarm_init()
{
    alarm_state=STATE(al_waiting);
    Set_Alarm_actuator(1);
}

STATE_define(al_waiting)
{
    alarm_state_id=al_waiting;
    // if(alarm==1) alarm_state=STATE(AlarmOn);
    // else alarm_state=STATE(AlarmOff);
}

STATE_define(AlarmOn)
{
    alarm_state_id=AlarmOn;
    Set_Alarm_actuator(0);
    Delay(10000);
    alarm_state=STATE(AlarmOff);
}
STATE_define(AlarmOff)
{
    alarm_state_id=AlarmOff;
    Set_Alarm_actuator(1);
    alarm_state=STATE(al_waiting);
}

void alarmOn()
{
    alarm_state=STATE(AlarmOn);
}
void alarmOff()
{
    alarm_state=STATE(AlarmOff);
```

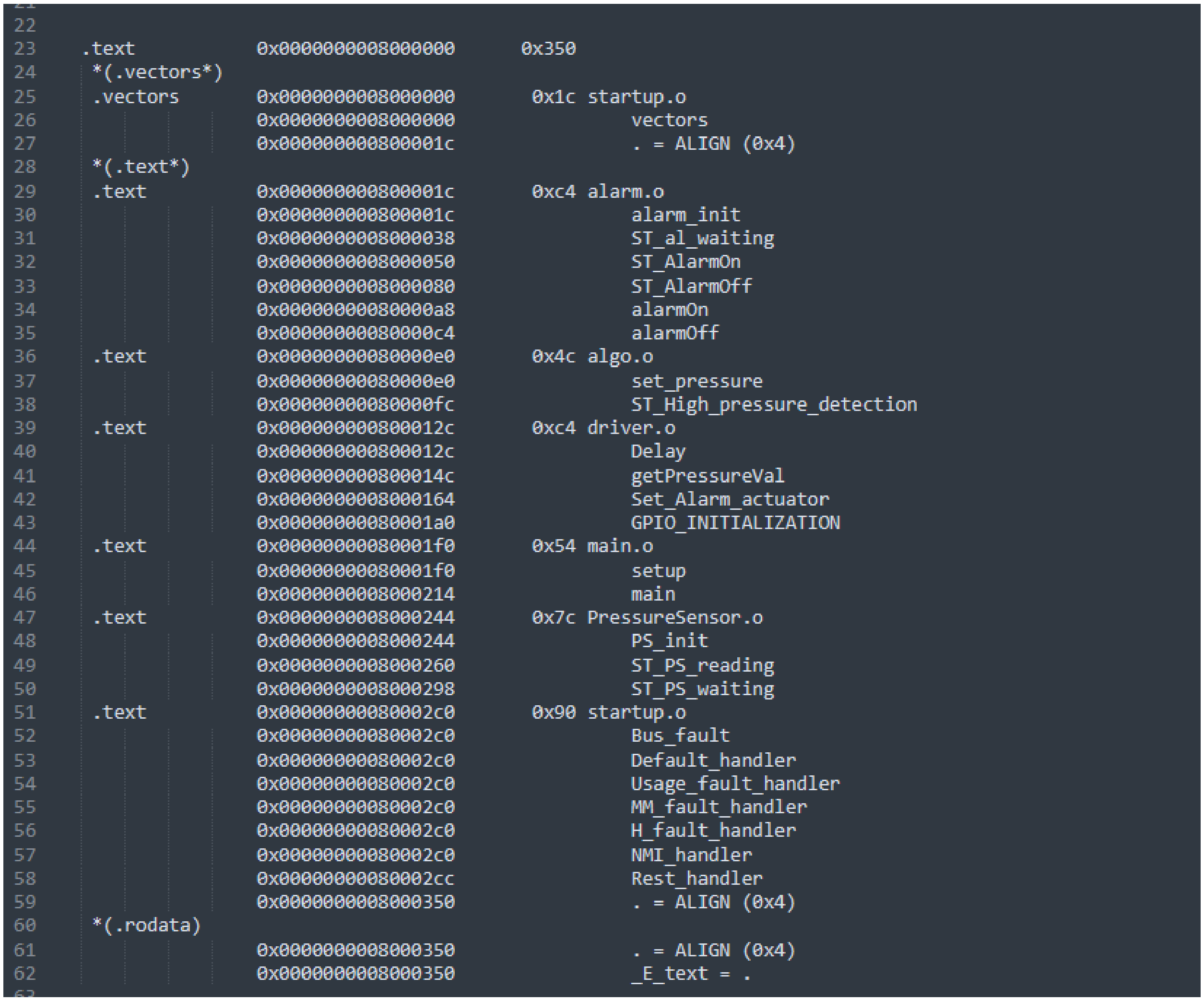
Some SW analysis:

Section table:



```
Project_pressurecontrol.elf:     file format elf32-littlearm

Sections:
Idx Name          Size      VMA       LMA       File off  Algn
  0 .text         00000350  08000000  08000000  00010000  2**2
                  CONTENTS, ALLOC, LOAD, READONLY, CODE
  1 .data         00000000  20000000  08000350  00020000  2**0
                  CONTENTS, ALLOC, LOAD, DATA
  2 .bss          00001024  20000000  08000350  00020000  2**2
                  ALLOC
  3 .debug_info   00003448  00000000  00000000  00020000  2**0
                  CONTENTS, READONLY, DEBUGGING
  4 .debug_abbrev 00000a32  00000000  00000000  00023448  2**0
                  CONTENTS, READONLY, DEBUGGING
  5 .debug_loc    00000488  00000000  00000000  00023e7a  2**0
                  CONTENTS, READONLY, DEBUGGING
  6 .debug_aranges 000000c0  00000000  00000000  00024302  2**0
                  CONTENTS, READONLY, DEBUGGING
  7 .debug_line   00000cad  00000000  00000000  000243c2  2**0
                  CONTENTS, READONLY, DEBUGGING
  8 .debug_str    00000690  00000000  00000000  0002506f  2**0
                  CONTENTS, READONLY, DEBUGGING
  9 .comment      0000007b  00000000  00000000  000256ff  2**0
                  CONTENTS, READONLY
```

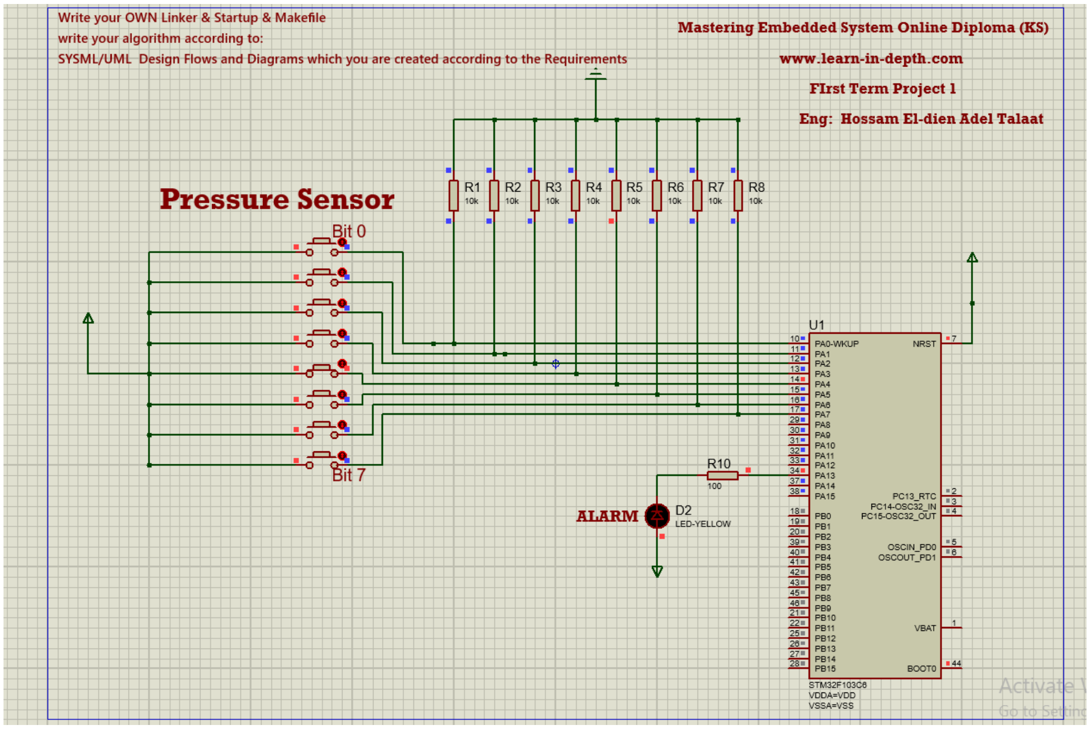
Map file:



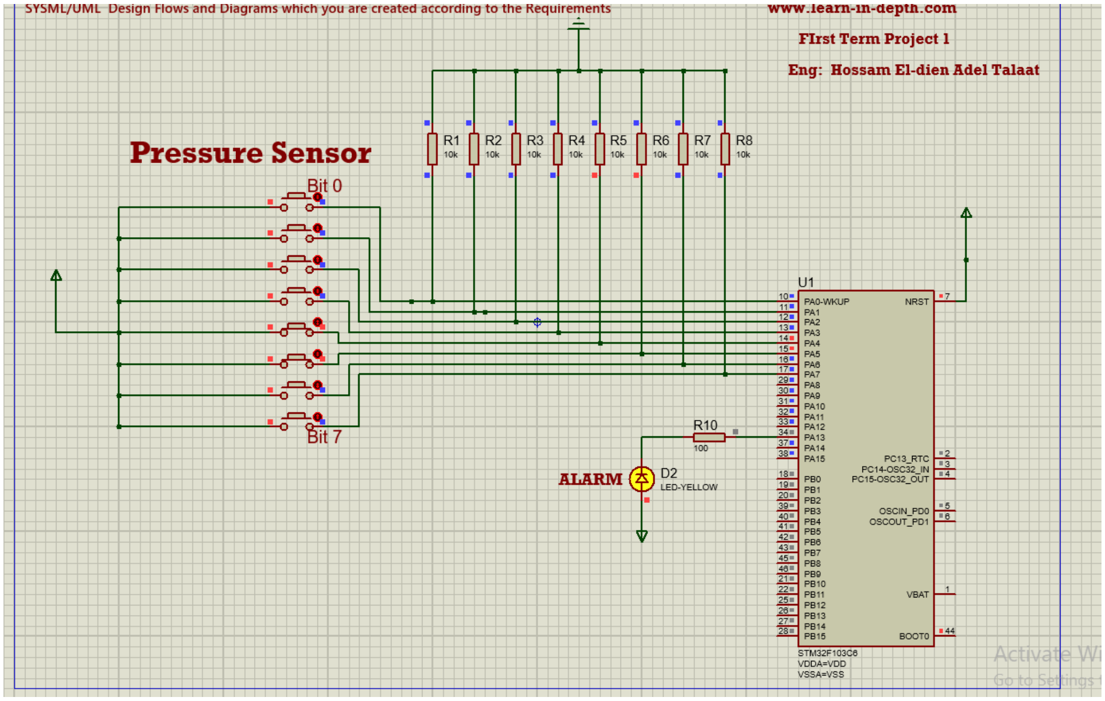
```
 .text           0x0000000008000000      0x350
  *(.vectors*)
  .vectors       0x0000000008000000       0x1c startup.o
                 0x0000000008000000                vectors
                 0x000000000800001c                . = ALIGN (0x4)
  *(.text*)
  .text          0x000000000800001c       0xc4 alarm.o
                 0x000000000800001c                alarm_init
                 0x0000000008000038                ST_al_waiting
                 0x0000000008000050                ST_AlarmOn
                 0x0000000008000080                ST_AlarmOff
                 0x00000000080000a8                alarmOn
                 0x00000000080000c4                alarmOff
  .text          0x00000000080000e0       0x4c algo.o
                 0x00000000080000e0                set_pressure
                 0x00000000080000fc                ST_High_pressure_detection
  .text          0x000000000800012c       0xc4 driver.o
                 0x000000000800012c                Delay
                 0x000000000800014c                getPressureVal
                 0x0000000008000164                Set_Alarm_actuator
                 0x00000000080001a0                GPIO_INITIALIZATION
  .text          0x00000000080001f0       0x54 main.o
                 0x00000000080001f0                setup
                 0x0000000008000214                main
  .text          0x0000000008000244       0x7c PressureSensor.o
                 0x0000000008000244                PS_init
                 0x0000000008000260                ST_PS_reading
                 0x0000000008000298                ST_PS_waiting
  .text          0x00000000080002c0       0x90 startup.o
                 0x00000000080002c0                Bus_fault
                 0x00000000080002c0                Default_handler
                 0x00000000080002c0                Usage_fault_handler
                 0x00000000080002c0                MM_fault_handler
                 0x00000000080002c0                H_fault_handler
                 0x00000000080002c0                NMI_handler
                 0x00000000080002cc                Rest_handler
                 0x0000000008000350                . = ALIGN (0x4)
  *(.rodata)
                 0x0000000008000350                . = ALIGN (0x4)
                 0x0000000008000350                _E_text = .
```

Simulation:

Pressure (0) < threshold (20)



Pressure (48) > threshold (20)


SYSML/UML Design Flows and Diagrams which you are created according to the Requirements
www.learn-in-depth.com
FIrst Term Project 1
Eng: Hossam El-dien Adel Talaat
Pressure Sensor
Bit 0
Bit 7
R1 10k
R2 10k
R3 10k
R4 10k
R5 10k
R6 10k
R7 10k
R8 10k
U1
PA0-WKUP
PA1
PA2
PA3
PA4
PA5
PA6
PA7
PA8
PA9
PA10
PA11
PA12
PA13
PA14
PA15
PB0
PB1
PB2
PB3
PB4
PB5
PB6
PB7
PB8
PB9
PB10
PB11
PB12
PB13
PB14
PB15
NRST
PC13_RTC
PC14-OSC32_IN
PC15-OSC32_OUT
OSCIN_PD0
OSCOUT_PD1
VBAT
BOOT0
R10
100
ALARM
D2
LED-YELLOW
STM32F103C8
VDDA=VDD
VSSA=VSS