Learn-In-Depth

Be Professional in Embedded System

First term Project1 Report

Pressure Controller Project

Presented by: Eng. Mohamed Ahmed Salama

Submitted to: Eng. Keroles Shenoda

GitHub Project

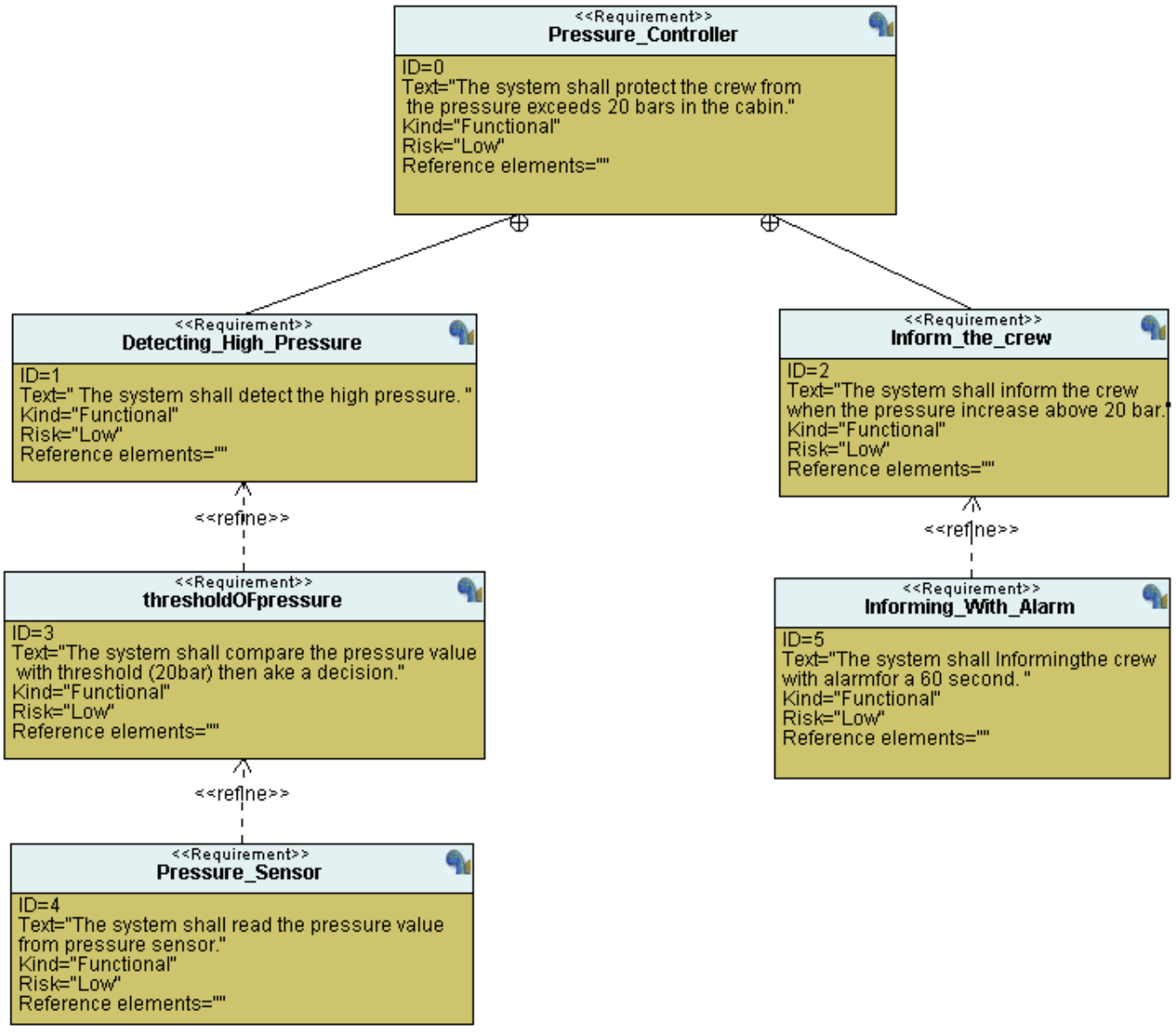
My Diploma profile

Problem description

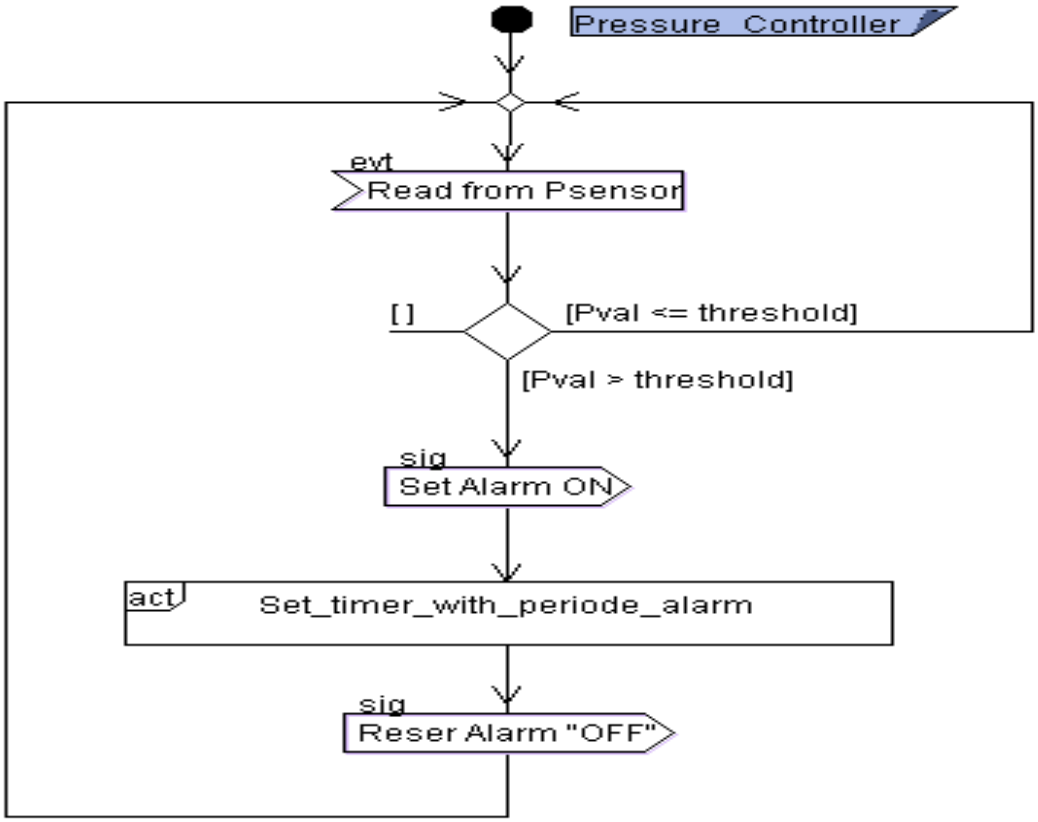
A client expects you to deliver the software of the following system specifications (from the client):

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

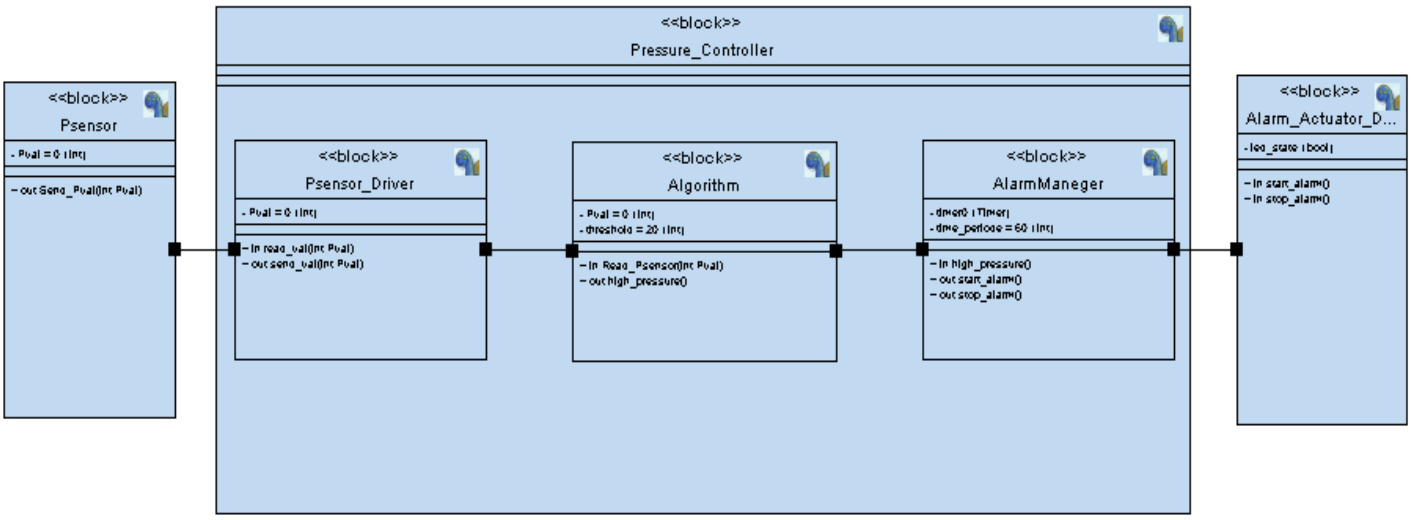
Requirements Diagram



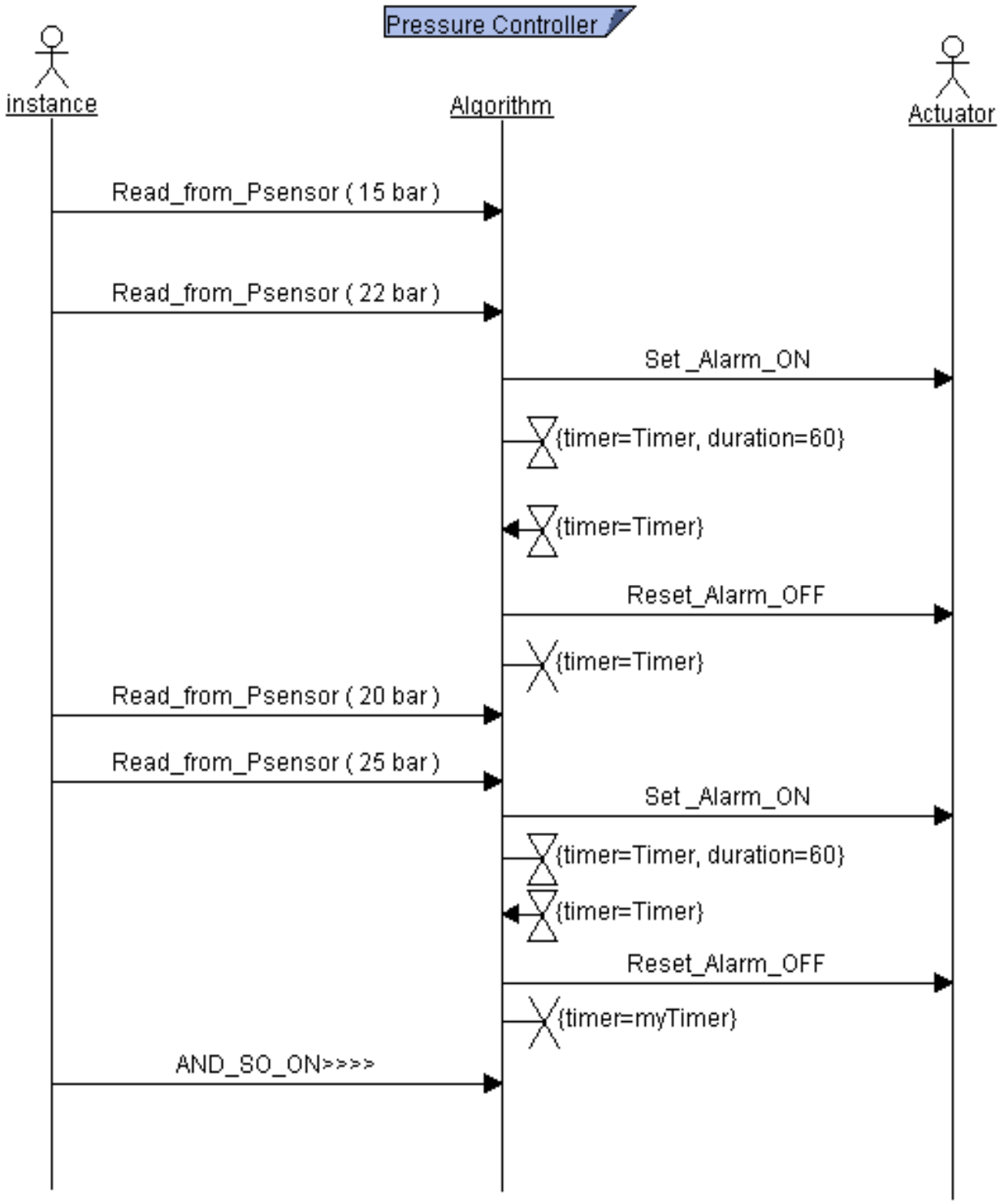
Activity Diagram



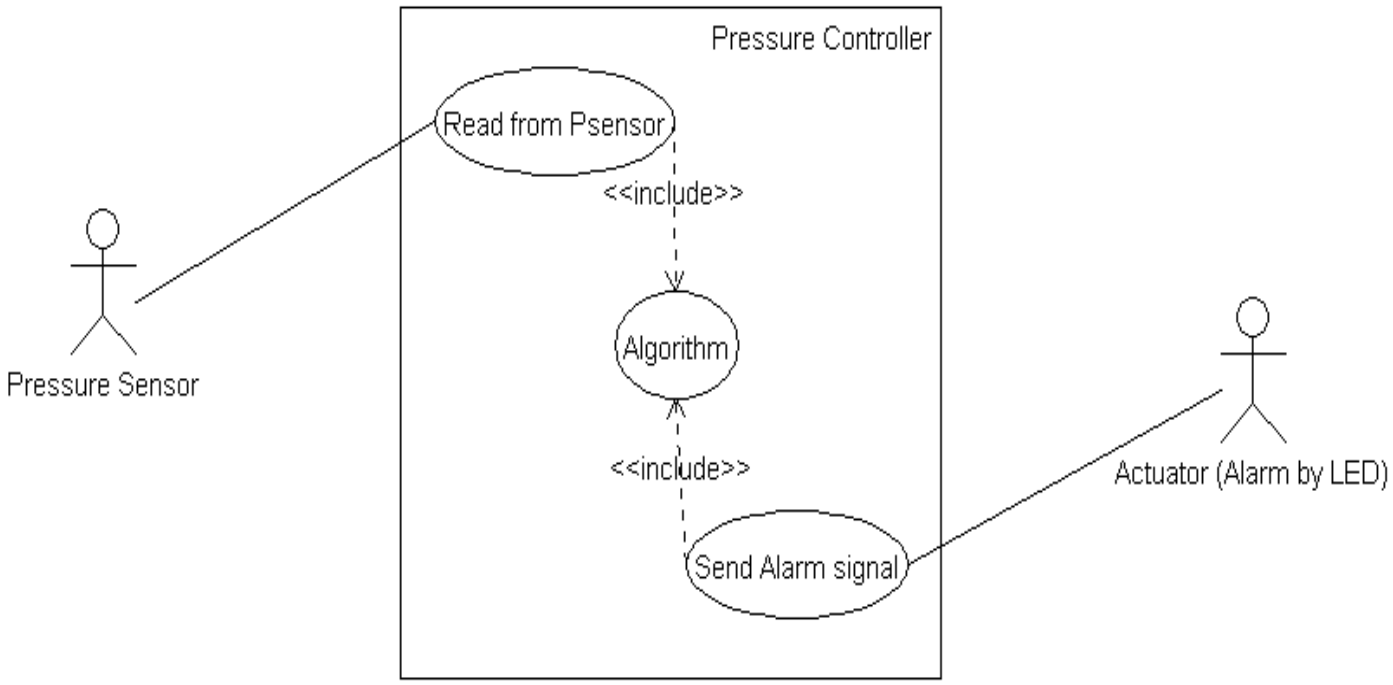
Block Diagram



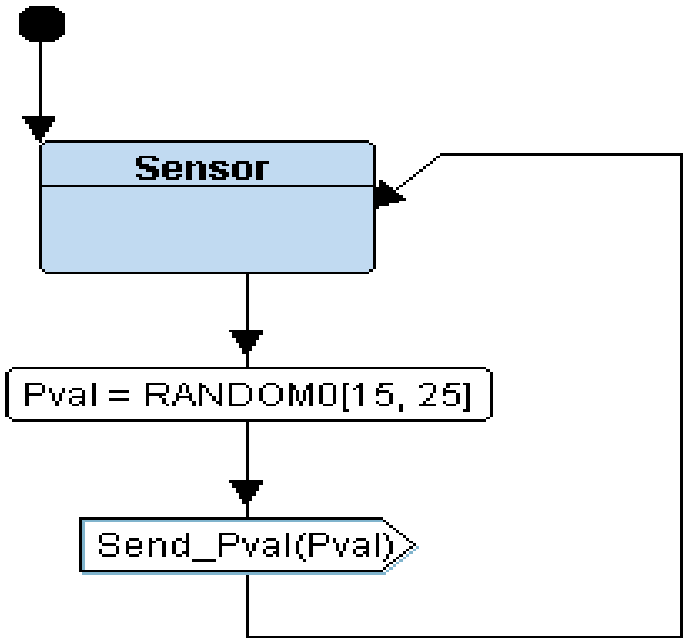
Sequence Diagram

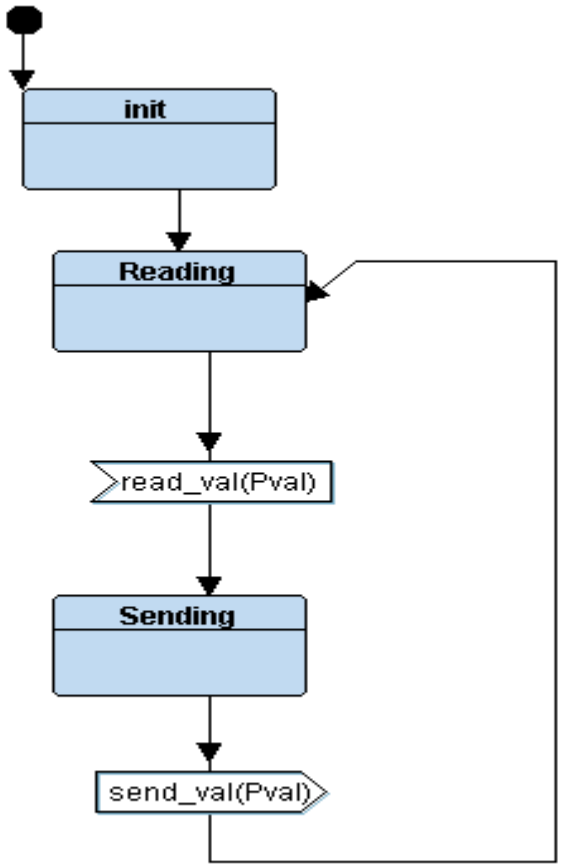


Use Case Diagram

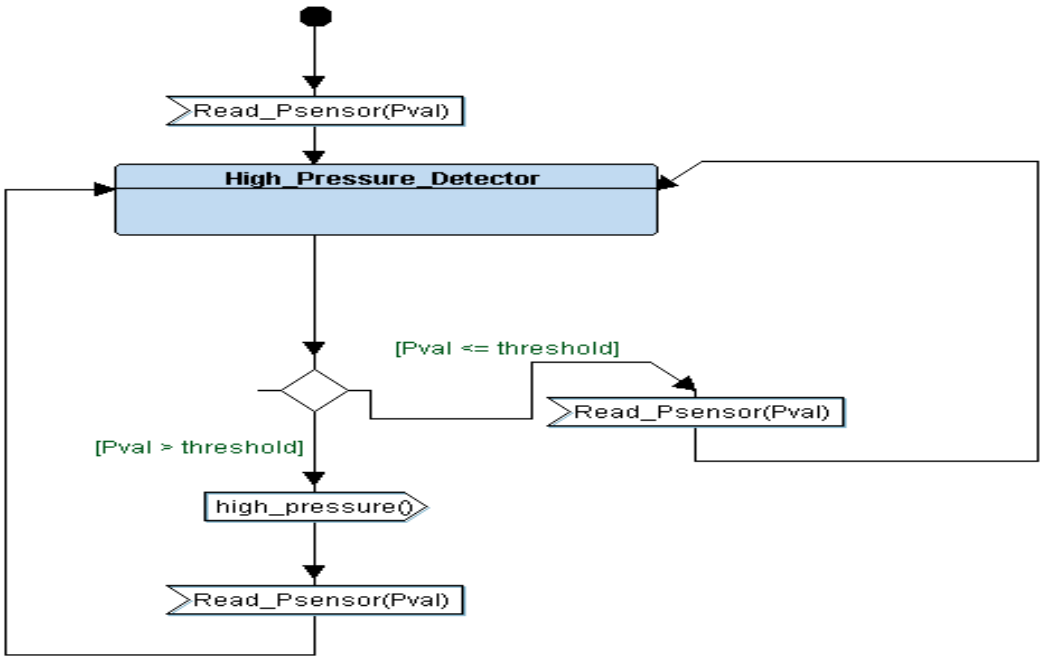


Pressure Sensor Block Diagram

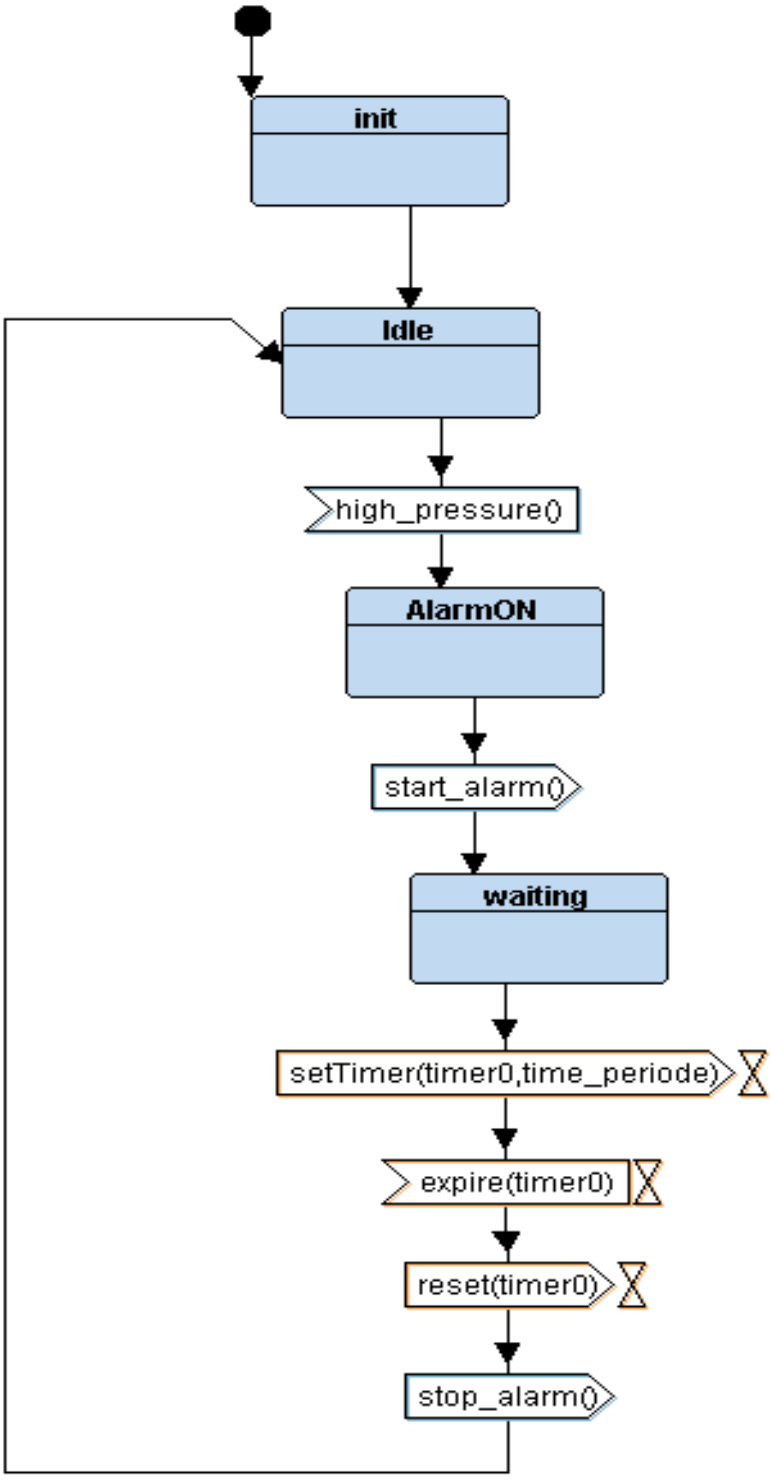




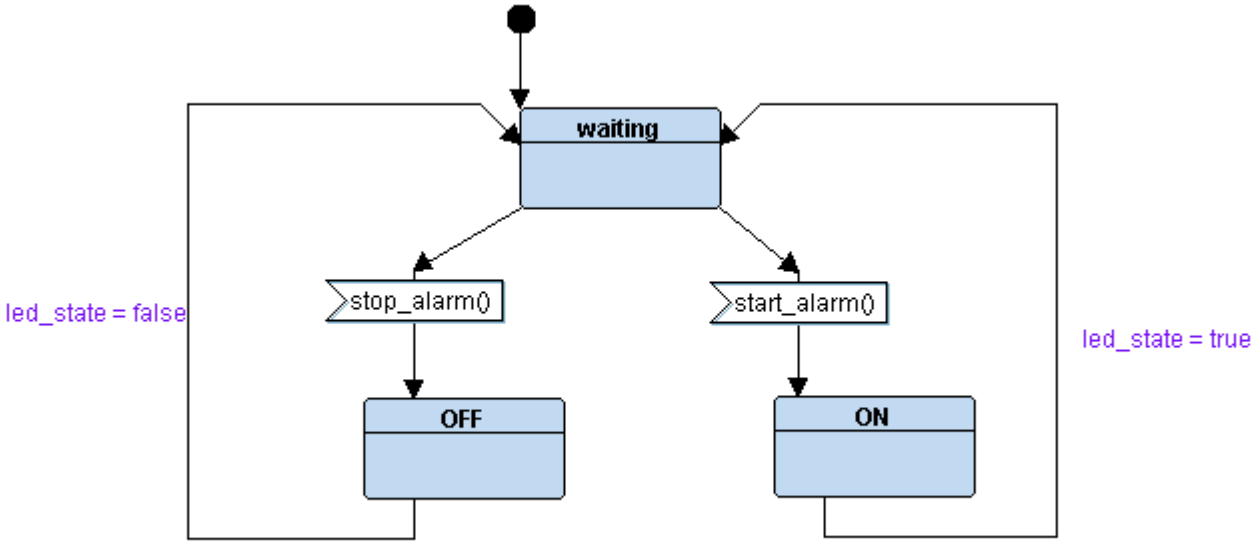
Algorithm Block Diagram



Alarm Monitor Block Diagram



Alarm Actuator Block Diagram



Code implementation

F:\Embedded_Systems\3-Course_Learn_in_depth\Units\First Term\Projects\1\Project\Src\main.c - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

main.c

```c
/*
 * main.c
 *
 *      Author: Eng. Mohamed Ahmed
 */

#include "driver.h"
#include "Alarm_Monitor.h"
#include "Alarm_Actuator.h"
#include "Algorithm.h"
#include "Sensor.h"
#include "State.h"

void setup()
{
    //init all drivers
    GPIO_INITIALIZATION();
    //init block
    PrSensor_init();
    alg_state = STATE(HighPreDetected);
    AM_state = STATE(AlarmOff);
    Alarm_init();
}
void main()
{
    setup();
    while(1)
    {
        PrS_state();
        alg_state();
        AM_state();
        Led_state();
```

C source file | length : 501 lines : 34 | Ln : 1 Col : 1 Pos : 1 | Windows (CR LF) | UTF-8 | INS

F:\Embedded_Systems\3-Course_Learn_in_depth\Units\First Term\Projects\1\Project\Src\Algorithm.c - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

main.c | Algorithm.c | Alarm_Monitor.c | Alarm_Actuator.c | driver.c | Sensor.c

```c
/*
 * Algorithem.c
 *
 *      Author: Eng . Mohamed Ahmed
 */

#include "Algorithm.h"

//variabls
uint8 pVal = 0;
uint8 threshold = 20;

// state pointer to function
void (*alg_state)();

// connection abstraction
void setPressureVal(uint8 pressure)
{
    pVal = pressure;
    alg_state = STATE(HighPreDetected);
}

STATE_define(HighPreDetected)
{
    alg_state_id = HighPreDetected;
    if(pVal > threshold)
    {
        HighPressure();
        alg_state = STATE(HighPreDetected);
    }
    else
        alg_state = STATE(HighPreDetected);
```

C source file | length : 537 lines : 33 | Ln : 1 Col : 1 Pos : 1 | Windows (CR LF) | UTF-8 | INS

F:\Embedded_Systems\3-Course_Learn_in_depth\Units\First Term\Projects\1\Project\Src\Alarm_Monitor.c - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

main.c | Algorithm.c | Alarm_Monitor.c | Alarm_Actuator.c | driver.c | Sensor.c

```c
/*
 * Alarm_Monitor.c
 *
 *      Author: Eng. Mohamed Ahmed
 */


#include "Alarm_Monitor.h"

void (*AM_state)();

void HighPressure()
{
    AM_state = STATE(AlarmOn);
}

STATE_define(AlarmOff)
{
    AM_state_id = AlarmOff;
    StopAlarm();
}

STATE_define(AlarmOn)
{
    AM_state_id = AlarmOn;
    StartAlarm();
    AM_state = STATE(AlarmOff);
}
```

C source file | length : 357 lines : 29 | Ln : 1 Col : 1 Pos : 1 | Windows (CR LF) | UTF-8 | INS

F:\Embedded_Systems\3-Course_Learn_in_depth\Units\First Term\Projects\1\Project\Src\Alarm_Actuator.c - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

main.c | Algorithm.c | Alarm_Monitor.c | Alarm_Actuator.c | driver.c | Sensor.c

```c
/*
 * Alarm_Actuator.c
 *
 *      Author: Eng. Mohamed Ahmed
 */

#include "Alarm_Actuator.h"
#include "driver.h"

void (*Led_state)();

void Alarm_init()
{
    // init the alarm
    Led_state = STATE(Waiting);
}
void StartAlarm()
{
    Led_state = STATE(LedOn);
}

void StopAlarm()
{
    Led_state = STATE(LedOff);
}

STATE_define(LedOn)
{
    Led_state_id = LedOn;
    Set_Alarm_actuator(0);
    Delay(1500000);
    Set_Alarm_actuator(1);
```

C source file | length : 707 lines : 49 | Ln : 1 Col : 1 Pos : 1 | Windows (CR LF) | UTF-8 | INS

F:\Embedded_Systems\3-Course_Learn_in_depth\Units\First Term\Projects\1\Project\Src\driver.c - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

main.c | Algorithm.c | Alarm_Monitor.c | Alarm_Actuator.c | driver.c | Sensor.c

```c
// Eng . Mohamed Ahmed

#include "driver.h"
#include <stdint.h>
#include <stdio.h>
void Delay(int nCount)
{
    for(; nCount != 0; nCount--);
}

int getPressureVal(){
    return (GPIOA_IDR & 0xFF);
}

void Set_Alarm_actuator(int i){
    if (i == 1){
        SET_BIT(GPIOA_ODR,13);
    }
    else if (i == 0){
        RESET_BIT(GPIOA_ODR,13);
    }
}

void GPIO_INITIALIZATION (){
    SET_BIT(APB2ENR, 2);
    GPIOA_CRL &= 0xFF0FFFFF;
    GPIOA_CRL |= 0x00000000;
    GPIOA_CRH &= 0xFF0FFFFF;
    GPIOA_CRH |= 0x22222222;
}
```

C source file | length : 508 lines : 31 | Ln : 1 Col : 1 Pos : 1 | Windows (CR LF) | UTF-8 | INS

F:\Embedded_Systems\3-Course_Learn_in_depth\Units\First Term\Projects\1\Project\Src\Sensor.c - Notepad++

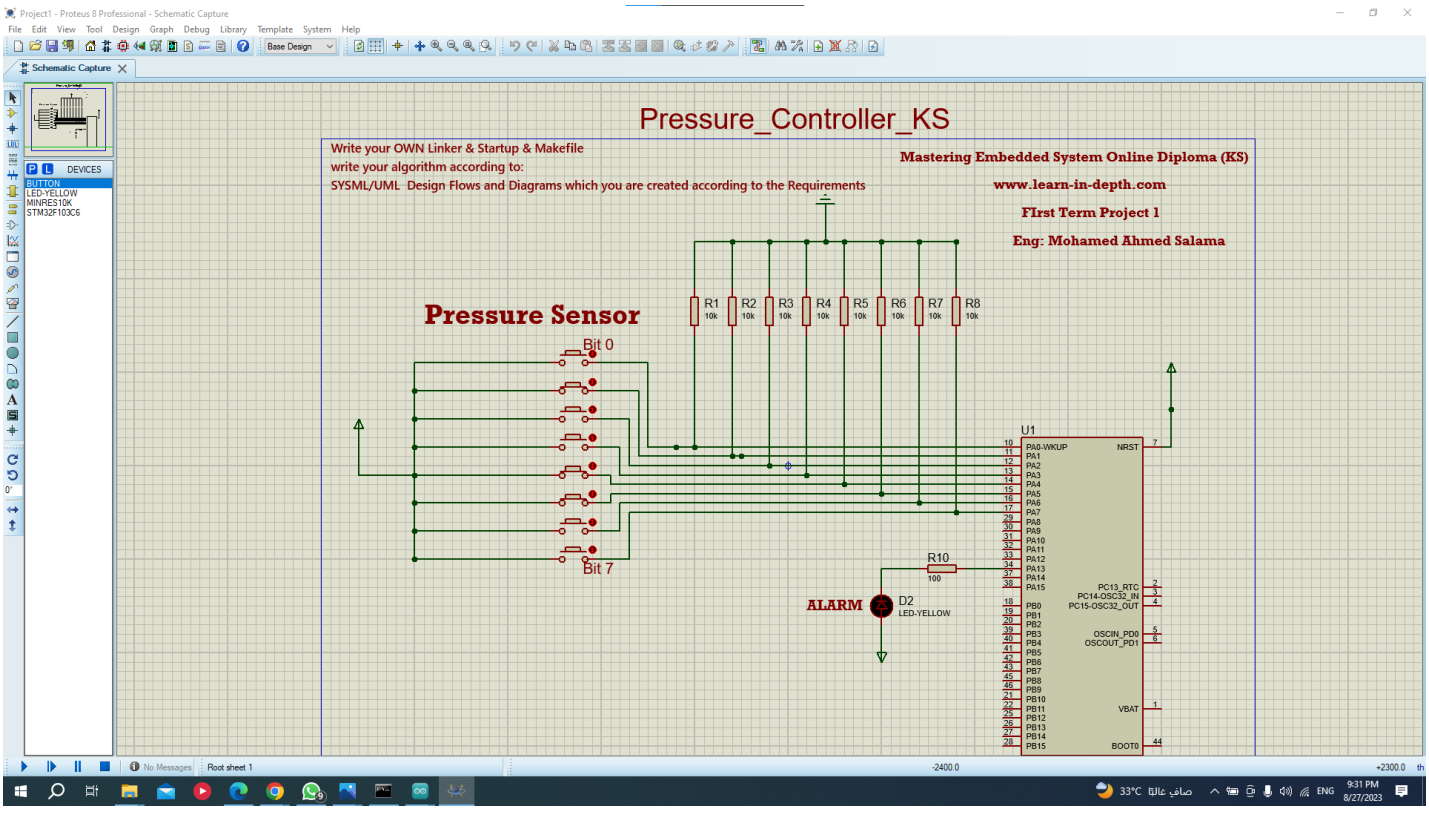
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

main.c | Algorithm.c | Alarm_Monitor.c | Alarm_Actuator.c | driver.c | Sensor.c

```c
/*
 * Sensor.c
 *
 *      Author: Eng. Mohamed Ahmed
 */

#include "Sensor.h"
#include "driver.h"

// variables
uint8 pressure = 0;

// State pointer to function
void (*PrS_state)();

// Flow of the program

void PrSensor_init()
{
    // initialize of the pressure sensor will be called from driver.h
    PrS_state = STATE(PrS_Reading);
}

STATE_define(PrS_Reading)
{
    PrS_state_id = PrS_Reading;
    pressure = getPressureVal();
    setPressureVal(pressure);
    PrS_state = STATE(PrS_Reading);
}
```

C source file | length : 514 lines : 32 | Ln : 1 Col : 1 Pos : 1 | Windows (CR LF) | UTF-8 | INS

Proteus Simulation



When pressure = 50 > threshold so alarm is on for 60 seconds then off