

Ain Shams University

Faculty of Engineering

Mechatronics Engineering Department



#LEARN_IN DEPTH

Mastering Embedded System Online Diploma

www.learn-in-depth.com

First Term (Final Project 1)

“Pressure Detection System”

Eng. Omar Ahmed Dawood

My Profile: < oa0578222@gmail.com (learn-in-depth.com) >

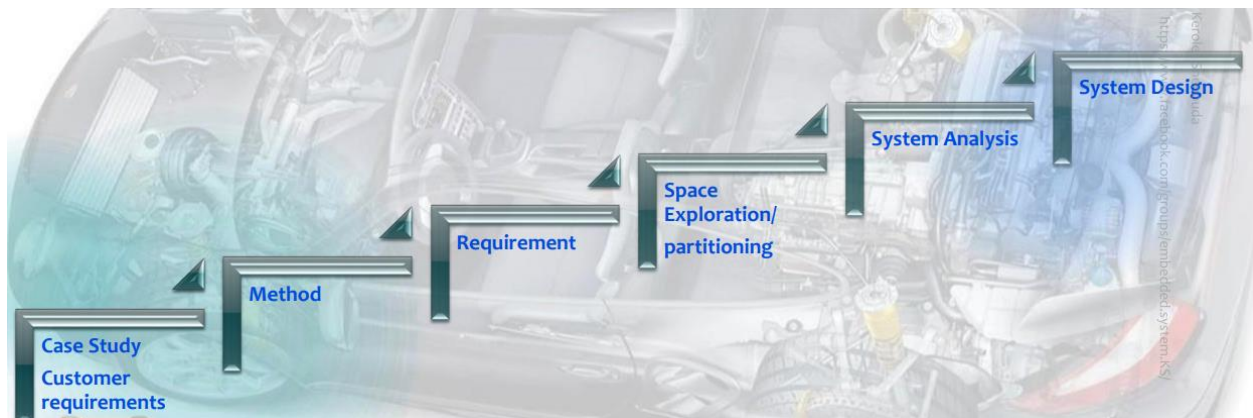
Contents

1	Introduction	3
1.1	Design Sequence	3
2	Requirements of Client (Case Study)	3
3	Method	4
4	Requirements Diagram	5
5	Space Exploration	5
6	System Analysis	6
6.1	Use Case Diagram	6
6.2	Activity Diagram.....	6
6.3	Sequence Diagram	7
7	System Design	8
7.1	All System.....	8
7.2	Pressure Sensor Block	8
7.3	Main Alg Block.....	9
7.4	LED Indicator Block	9
7.5	Alarm Monitoring Block	10
8	System Design Simulation	11
8.1	Pressure Less than 20.....	11
8.2	Pressure More than 20	11
9	Implementation	12
9.1	Pressure Sensor.....	12
9.2	Main Alg	12
9.3	Alarm Monitoring.....	13
9.4	LED Indicator	13
9.5	Linker script file.....	14
9.6	Startup file.....	15
10	Simulation on proteus	16
10.1	Hardware	16
10.2	Pressure less than 20 bar	16
10.3	Pressure more than 20 bar	17

1 Introduction

- in this project we will make Pressure Detection sensor that detect high pressure in the cabin of plane.
- Occur Alarm when Pressure exceeds on 20 bar.
- Alarm will continue for 60 seconds.
- We will Apply Design sequence to make this project.
- We will use stm32f103 microcontroller.

1.1 Design Sequence



Design Sequence

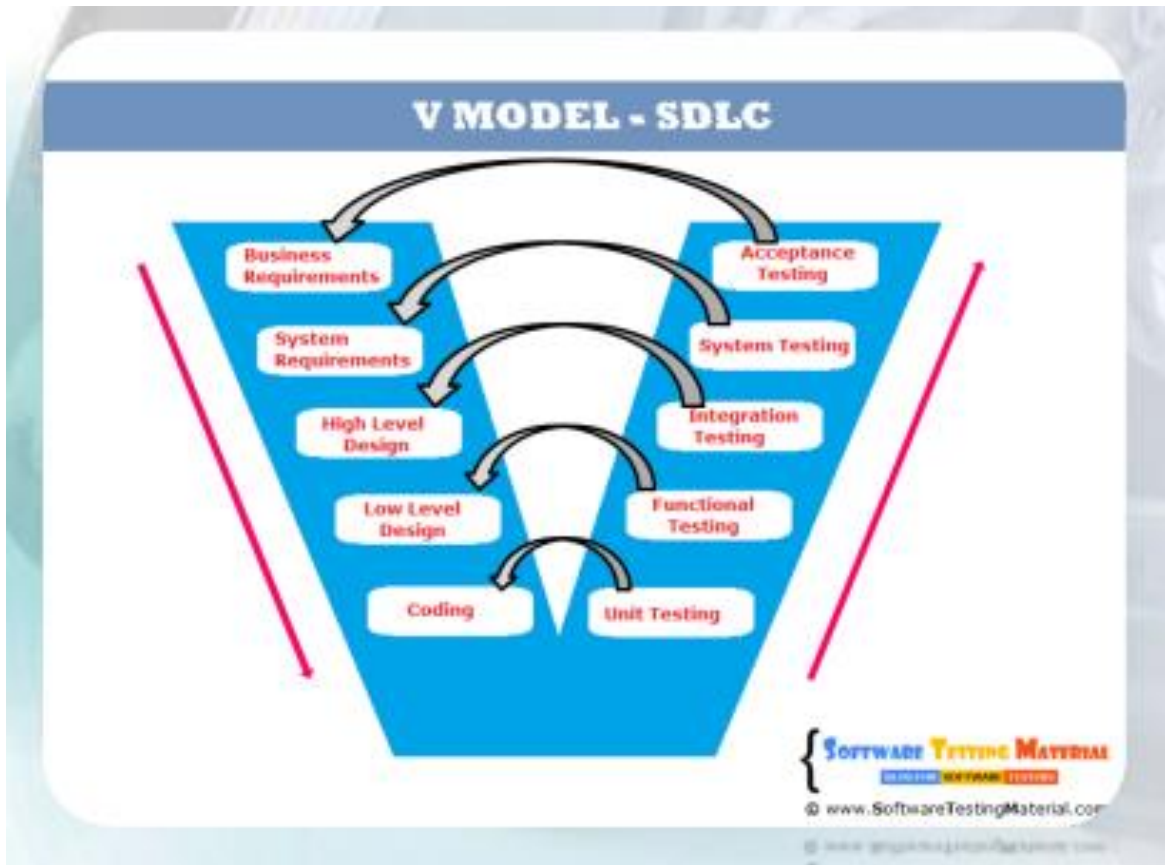
2 Requirements of Client (Case Study)

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

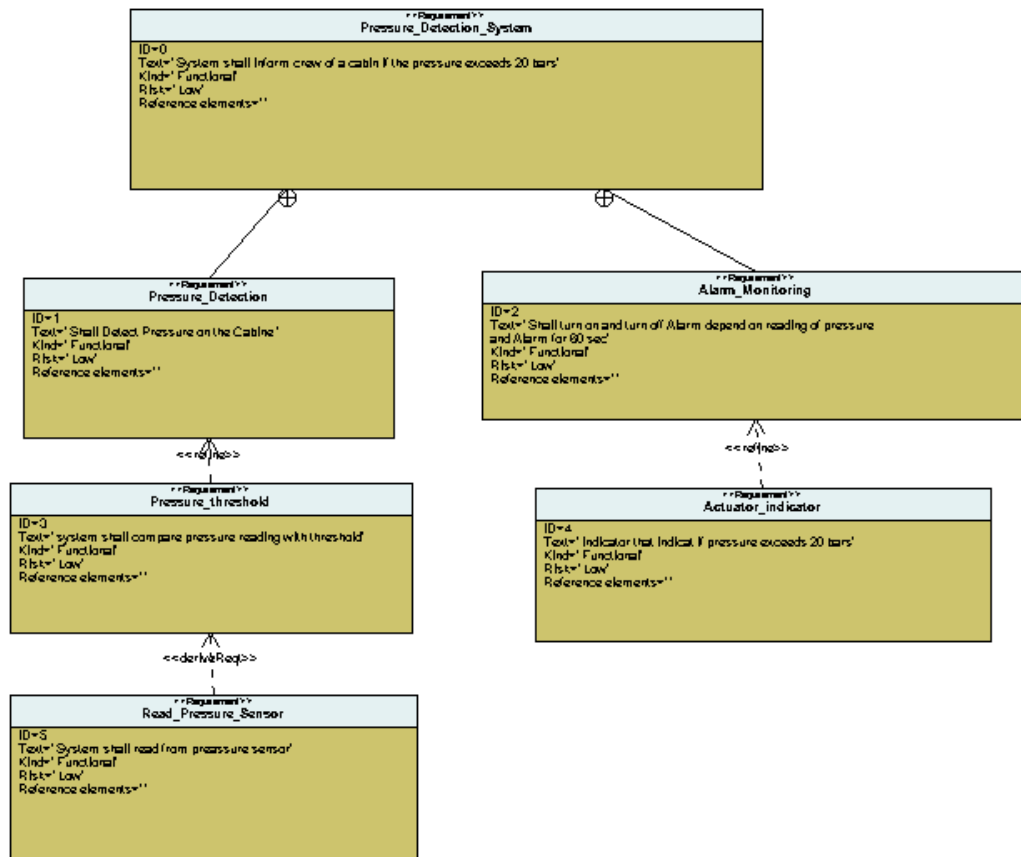
- Specification (from the client)
- A pressure controller informs the crew of a cabin with an alarm when the pressure exceeds 20 bars in the cabin
- The alarm duration equals 60 seconds.

3 Method

- We will Apply V Model.

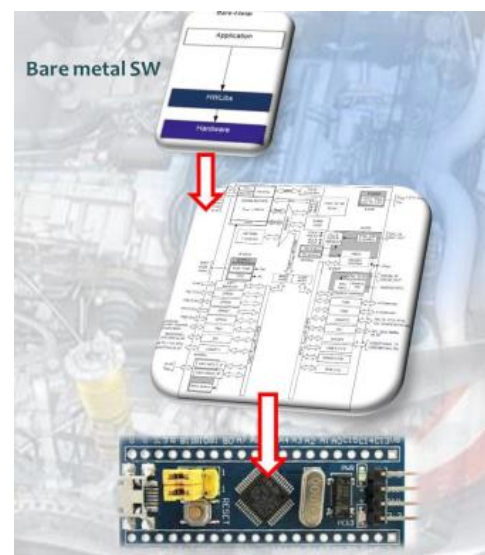


4 Requirements Diagram



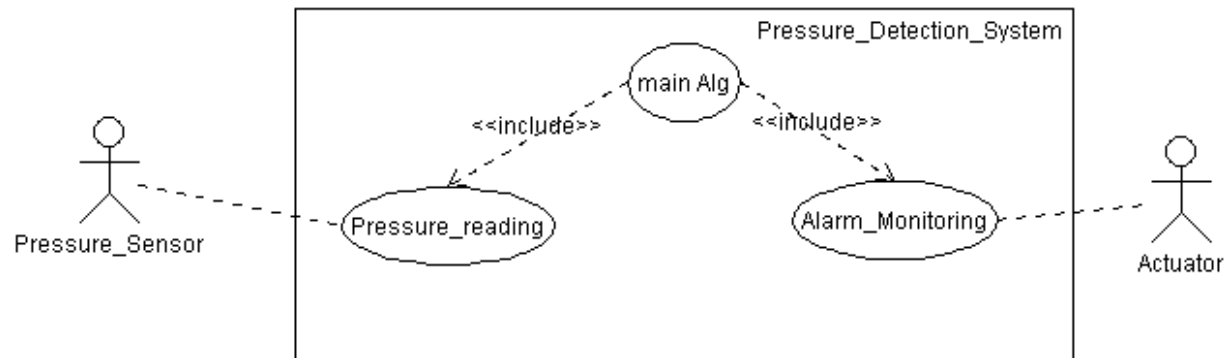
5 Space Exploration

- We will use only one ECU with one CPU Cortex M3 (STM32f103) Microcontroller.

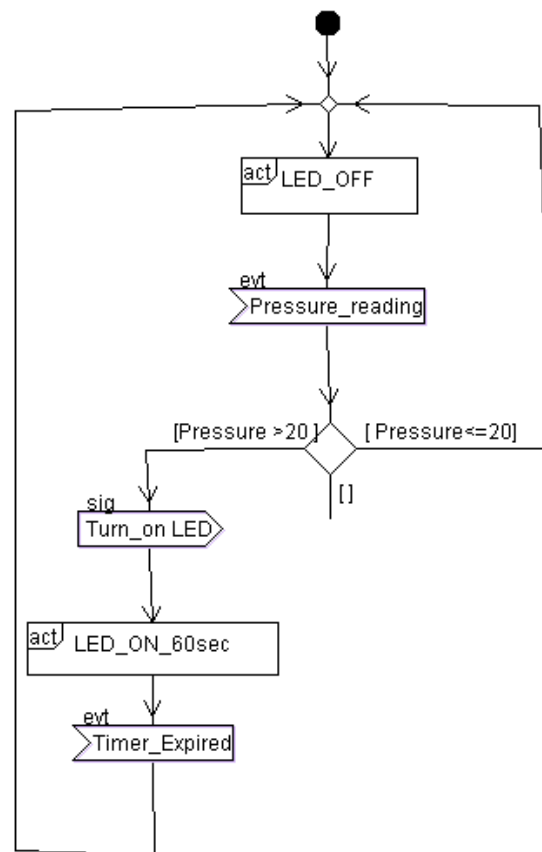


6 System Analysis

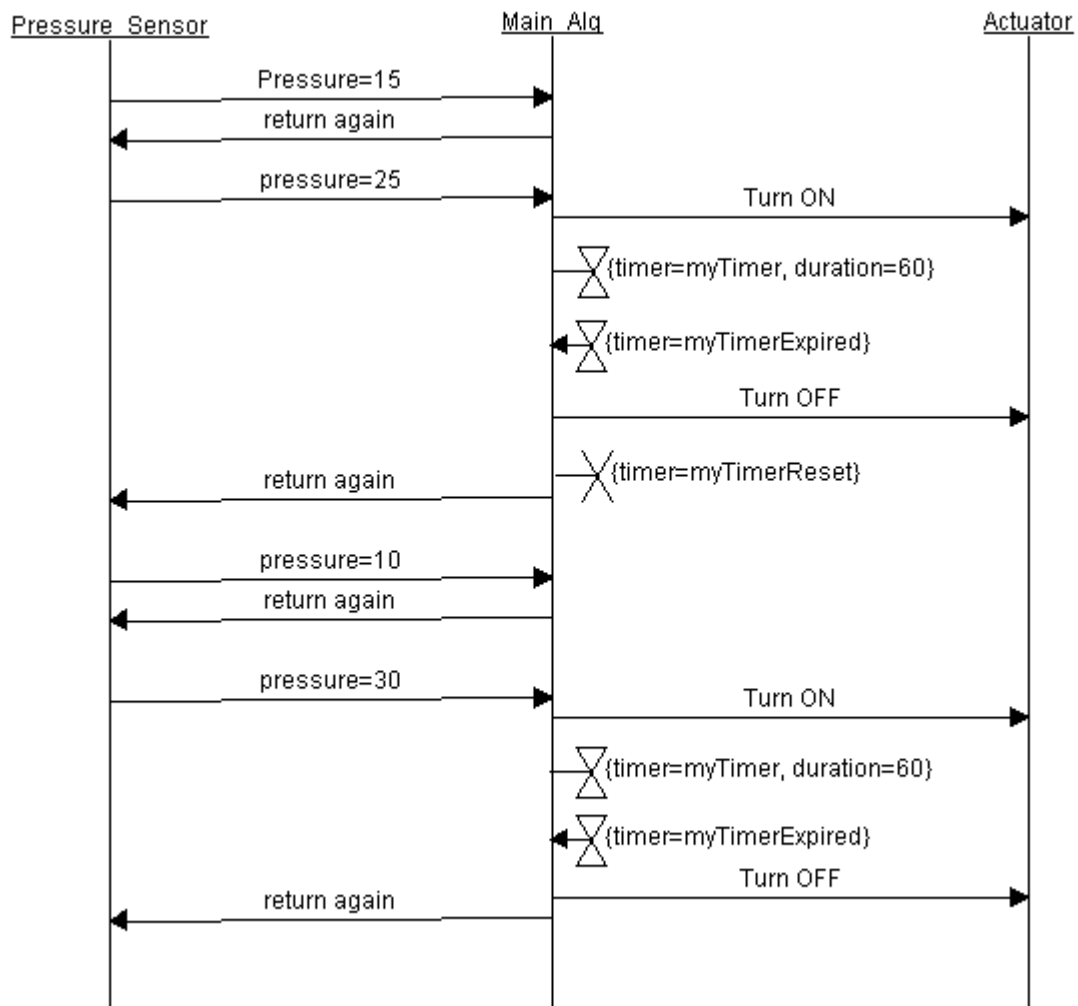
6.1 Use Case Diagram



6.2 Activity Diagram

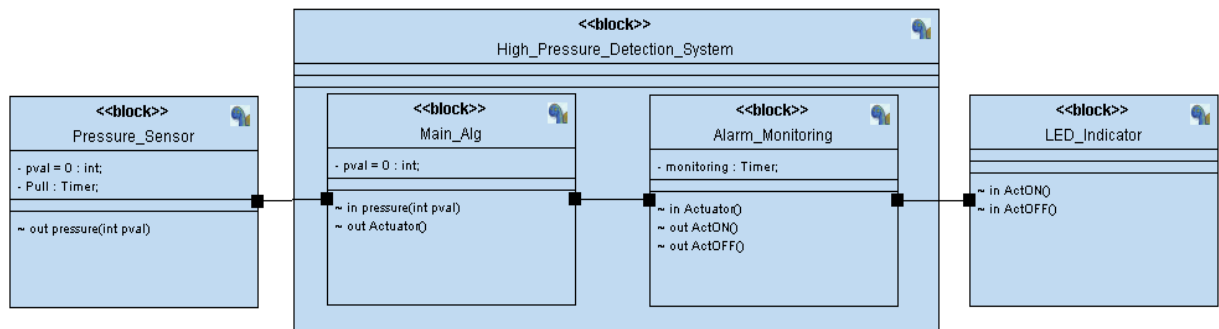


6.3 Sequence Diagram

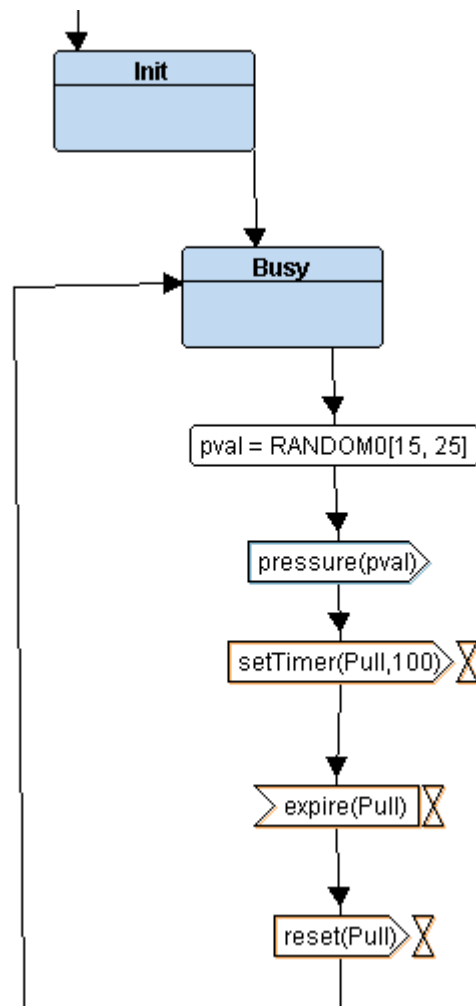


7 System Design

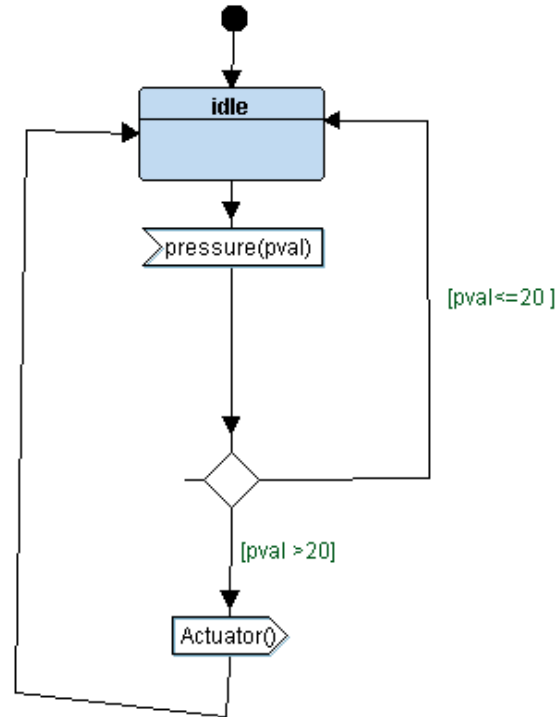
7.1 All System



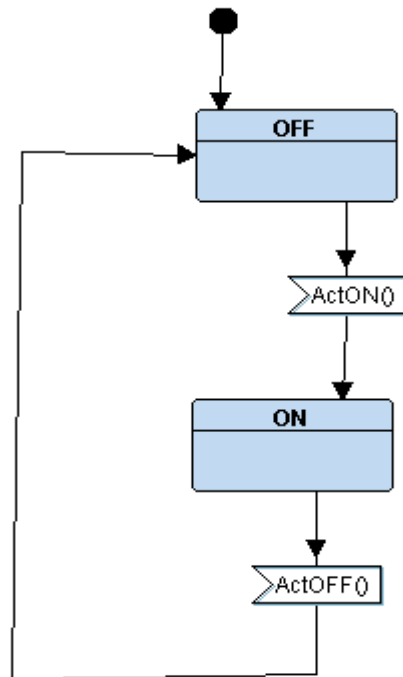
7.2 Pressure Sensor Block



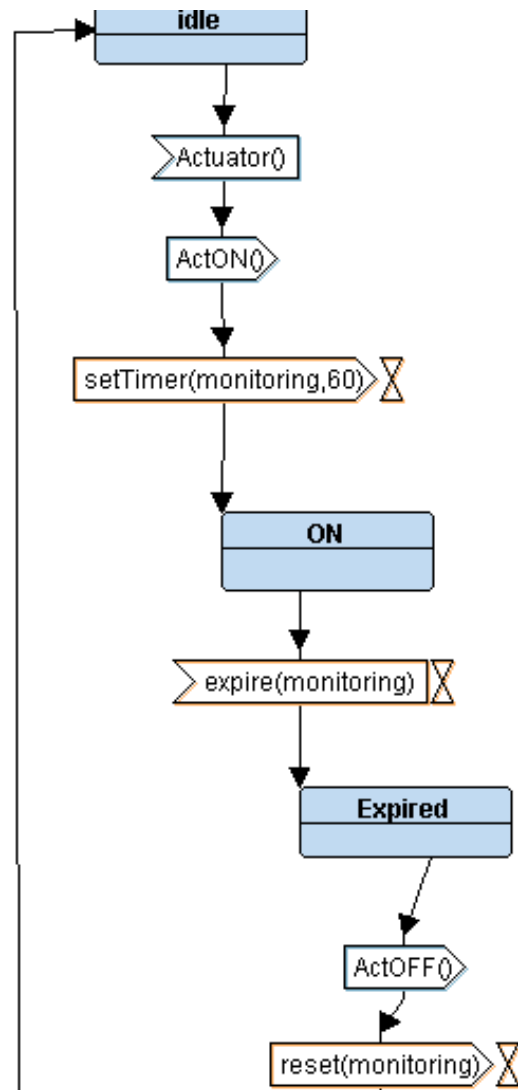
7.3 Main Alg Block



7.4 LED Indicator Block

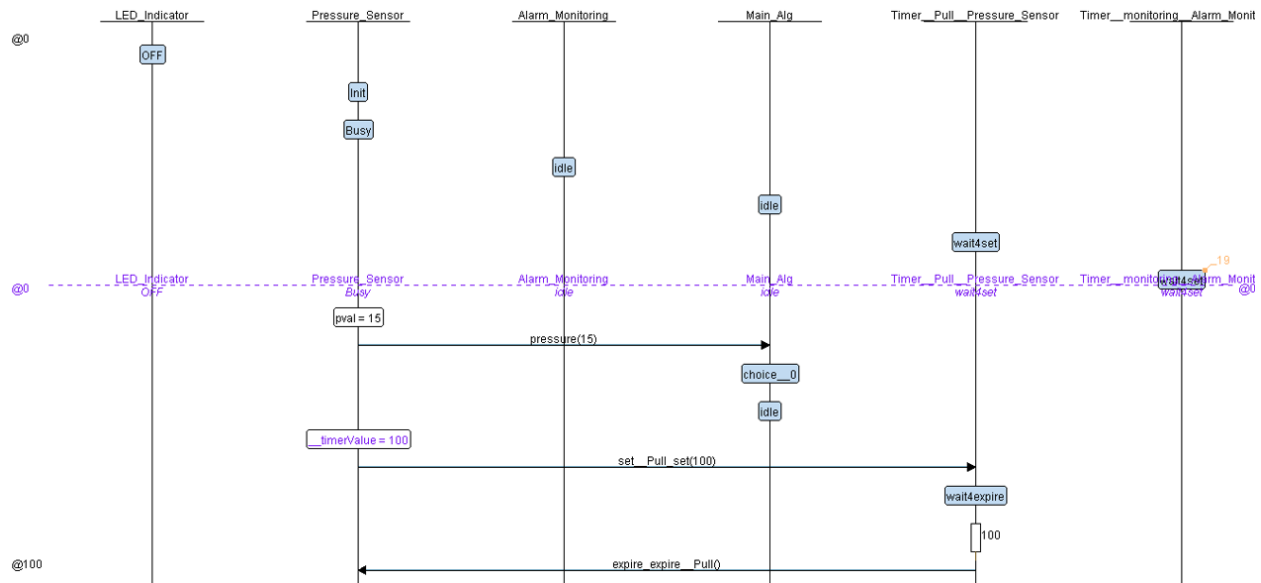


7.5 Alarm Monitoring Block

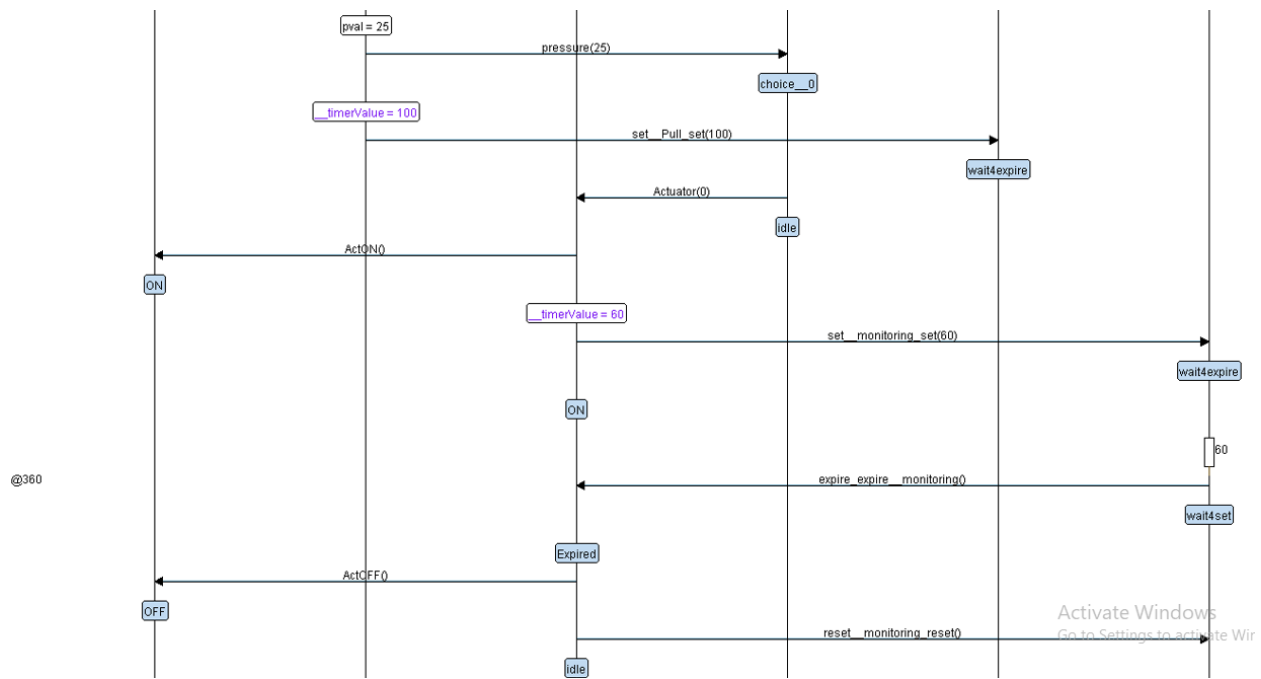


8 System Design Simulation

8.1 Pressure Less than 20



8.2 Pressure More than 20



9 Implementation

9.1 Pressure Sensor

```
#include "driver.h"
#include "pressure_sensor.h"

int getPressureReading ()
{
    return getPressureVal();
}
```

.c file

```
#ifndef _PRESSURE_SENSOR_H

int getPressureReading ();

#endif
```

.h file

9.2 Main Alg

```
#include "Alarm_Monitoring.h"
#include "Main_Algorithm.h"
#include "pressure_sensor.h"

void Alg ()
{
    if (getPressureReading()>20)
    {
        Alarm(); }
    else
    {
        Normal (); }
}
```

.c file

```
#ifndef _MAIN_ALG_H

void Alg ();

#endif
```

.h file

9.3 Alarm Monitoring

```
#include "driver.h"
#include "LED_Indicator.h"
#include "Alarm_Monitoring.h"
```

```
void Alarm ()
{
    LED_ON ();
    Delay(_60_Sec);
    LED_OFF ();
    Delay(_60_Sec);
}
```

```
void Normal ()
{
    LED_OFF ();
}
```

.c file

```
#ifndef _ALARM_MONITERING

#define _60_Sec      1000000000000

void Alarm ();

void Normal ();

#endif
```

.h file

9.4 LED Indicator

```
#include "driver.h"
#include "LED_Indicator.h"

void LED_ON ()
{
    Set_Alarm_actuator(ON);
}

void LED_OFF ()
{
    Set_Alarm_actuator(OFF);
}
```

.c file

```
#ifndef _LED_INDICATOR_H

#define ON      0
#define OFF    1

void LED_ON ();

void LED_OFF ();

#endif
```

.h file

9.5 Linker script file

```
/*
linker_script_cortex_m3.ld
Author: Omar Ahmed
*/

MEMORY
{
    flash(RX) : ORIGIN = 0x08000000, LENGTH = 128k
    sram(RWX) : ORIGIN = 0x20000000, LENGTH = 20k
}
SECTIONS
{
    .text :
    {
        *(.vectors*)
        *(.text*)
        *(.rodata)
        _E_text = . ;
    }>flash

    .data :
    {
        _S_data = . ;
        *(.data)
        . = ALIGN(4) ;
        _E_data = . ;
    }>sram AT>flash

    .bss :
    {
        _S_bss = . ;
        *(.bss)
        _E_bss = . ;
        . = ALIGN (4) ;
        . = . + 0x1000 ;
        _stack_top = . ;
    }>sram
}
```

9.6 Startup file

```
/*
startup.c
Eng. Omar Ahmed
*/

extern int main ();
void reset_handler ();

void Default_handler ()
{
    reset_handler();
}

void NMI_handler () __attribute__ ((weak,alias("Default_handler")));

void H_FAULT_handler ()__attribute__ ((weak,alias("Default_handler")));

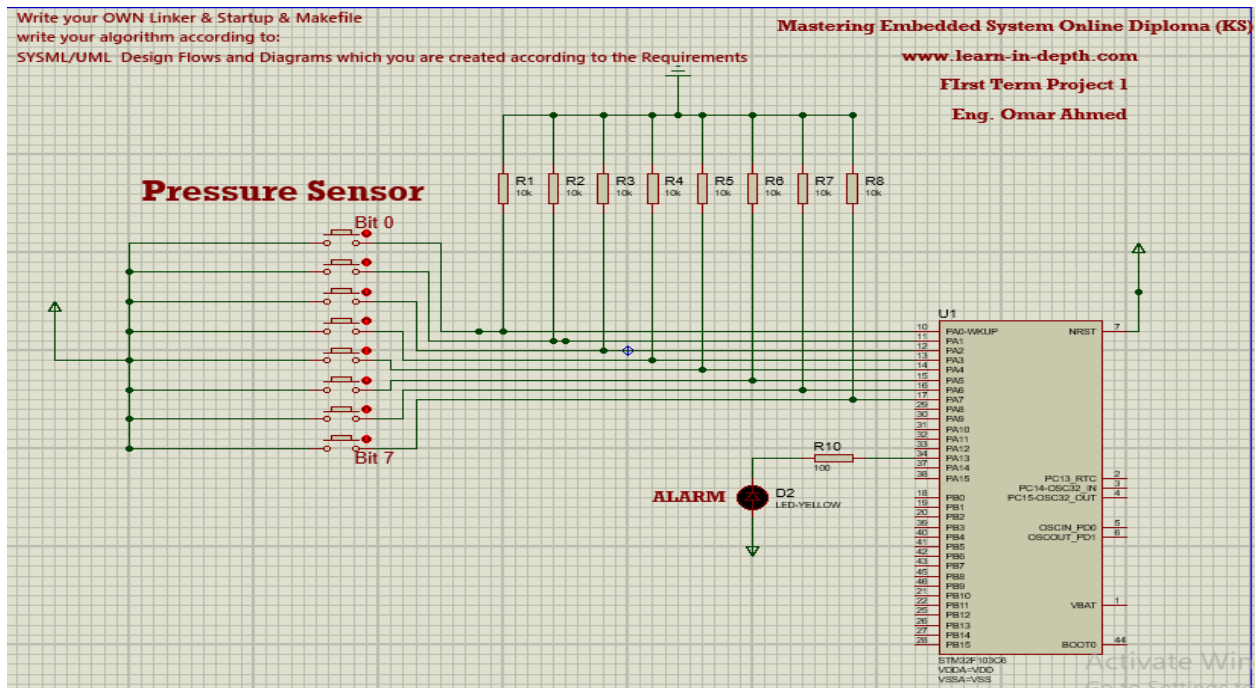
#include <stdint.h>

extern uint32_t _stack_top;

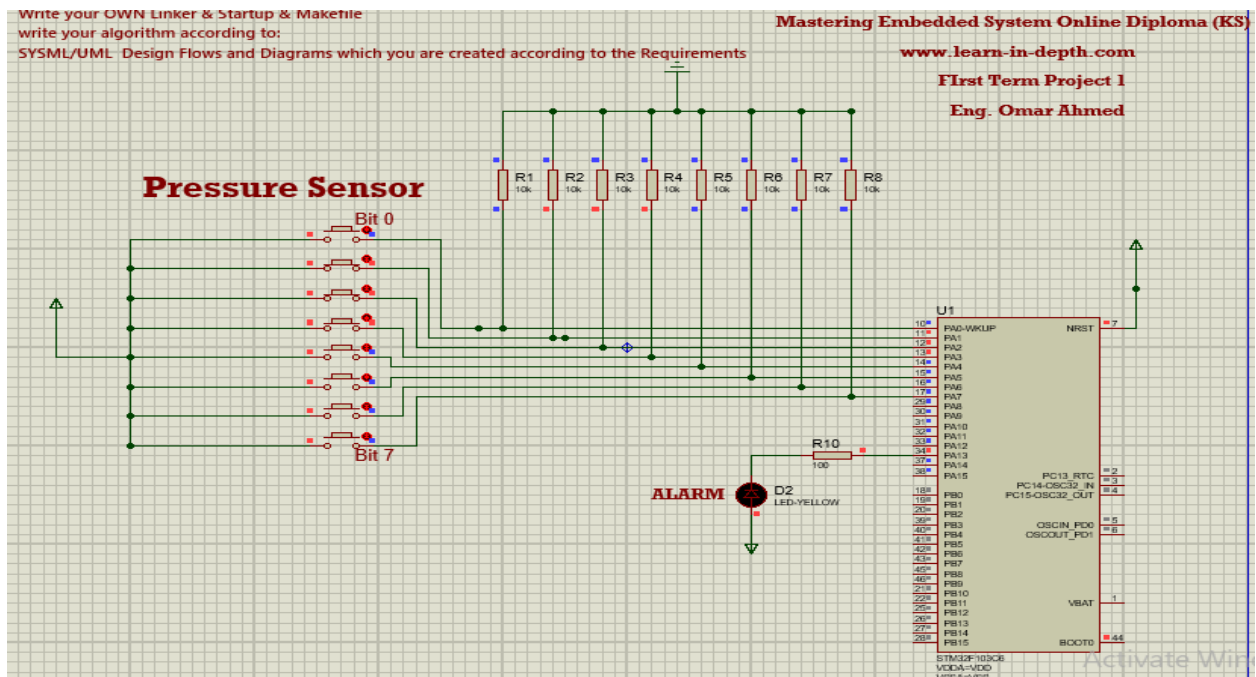
uint32_t vectors[] __attribute__((section(".vectors"))) =
{
    (uint32_t) &_stack_top , (uint32_t)&reset_handler,(uint32_t)&NMI_handler,(uint32_t)&H_FAULT_handler
};
extern uint32_t _E_text;
extern uint32_t _S_data;
extern uint32_t _E_data;
extern uint32_t _S_bss;
extern uint32_t _E_bss;
```

```
void reset_handler ()
{
    int i;
    uint32_t Data_size = (uint8_t*)&_E_data - (uint8_t*)&_S_data ;
    uint8_t *psrc = (uint8_t*)&_E_text;
    uint8_t *pdist = (uint8_t*)&_S_data;
    for (i=0;i<Data_size;i++)
    {
        *pdist = *psrc;
        pdist ++; psrc++;
    }
    uint32_t bss_size = (uint8_t*)&_E_bss - (uint8_t*)&_S_bss ;
    pdist = (uint8_t*)&_S_bss;
    for (i=0;i<bss_size;i++)
    {
        *pdist = 0;
        pdist++;
    }
    main();
}
```

10.1 Hardware



10.2 Pressure less than 20 bar



10.3 Pressure more than 20 bar

