PROJECT1: PRESSURE CONTROLLER

Eng. Ali Mohamed Taima

LEARN IN DEPTH

MASTERING EMBEDDED SYSTEM ONLINE DIPLOMA

Eng. Keroles

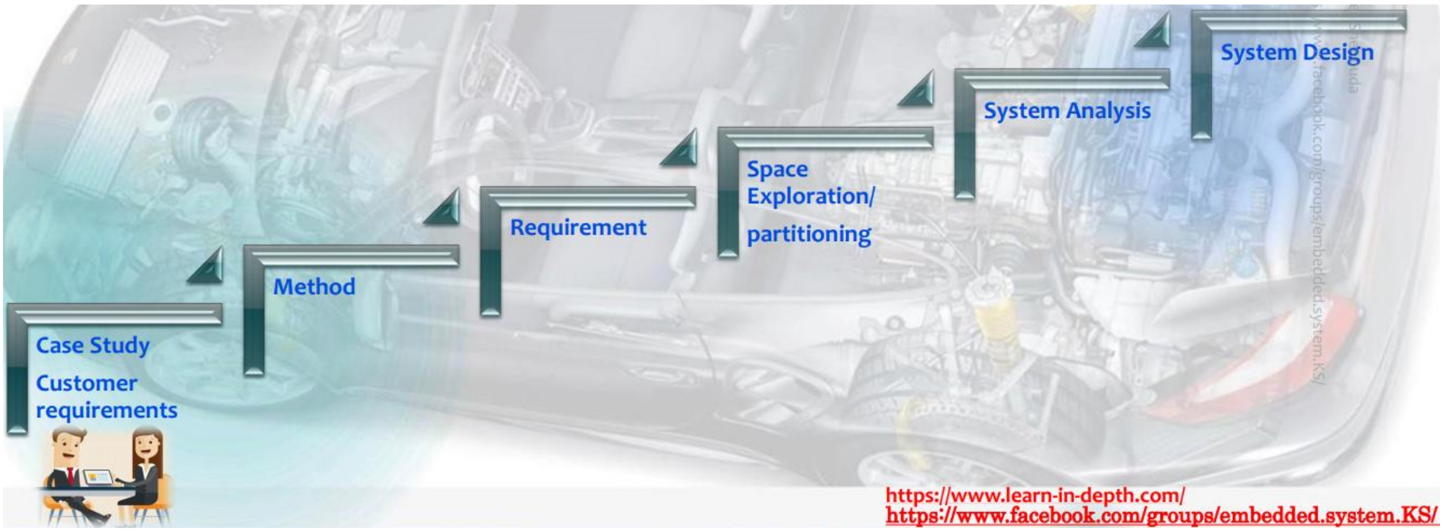
Embedded System Architecting/Design Sequence

1. Case Study

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

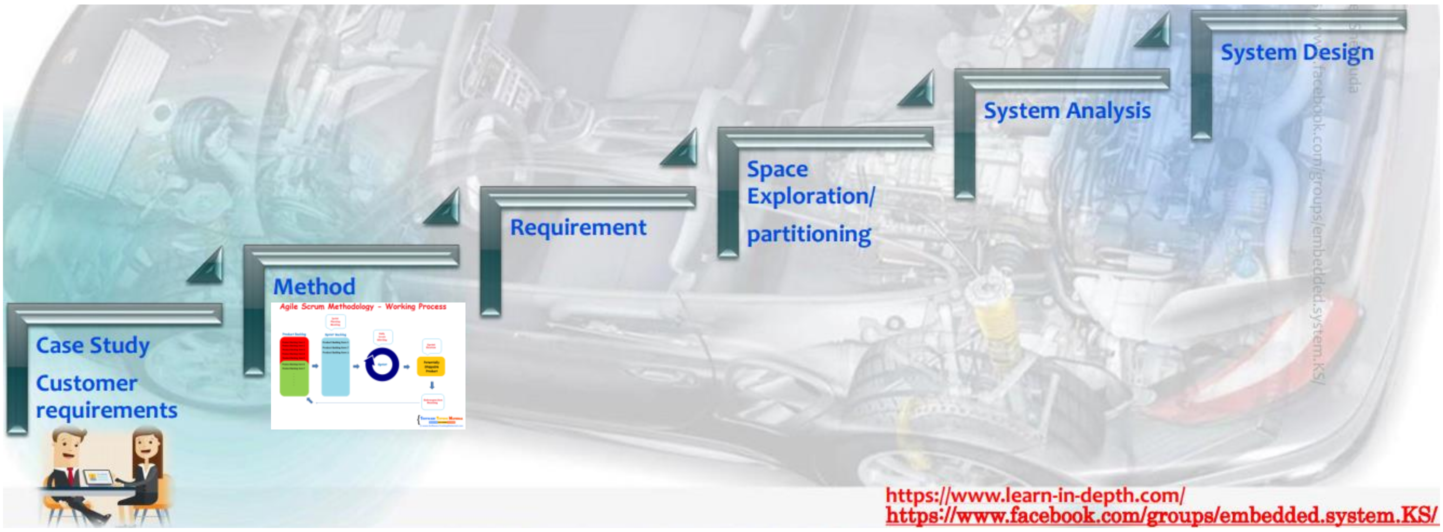
- ▶ Specification (from the client)
- ▶ **A pressure controller** informs the crew of a cabin with an alarm **when the pressure exceeds 20 bars in the cabin**
- ▶ **The alarm duration** equals 60 seconds.

- ▶ Pressure Controller: 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


Case Study
Customer requirements
Method
Requirement
Space Exploration/ partitioning
System Analysis
System Design
https://www.learn-in-depth.com/
https://www.facebook.com/groups/embedded.system.KS/

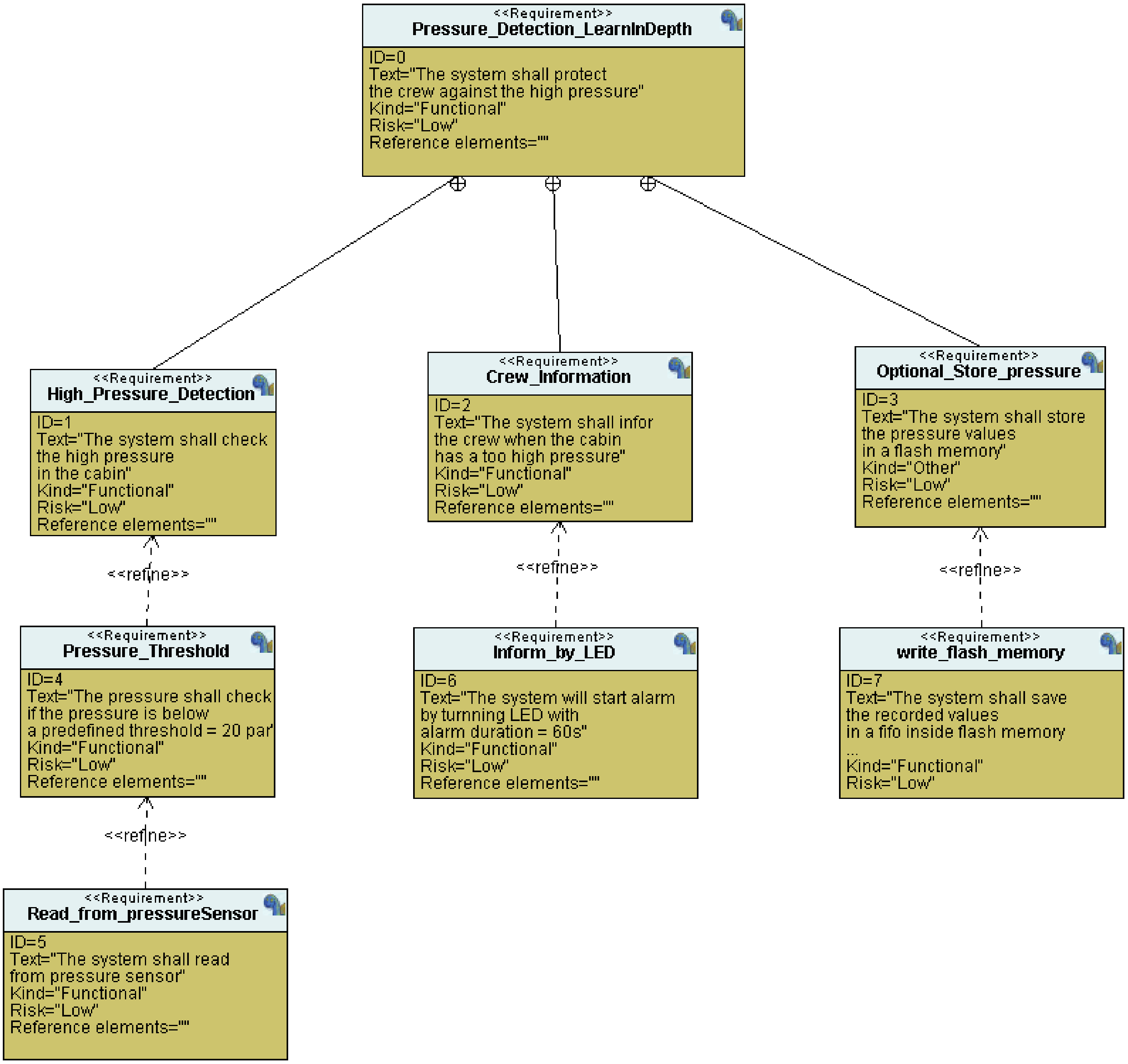
2. Method

I choose the Agile Scrum methodology to work with my project


Case Study
Customer requirements
Method
Agile Scrum Methodology - Working Process
Requirement
Space Exploration/ partitioning
System Analysis
System Design
https://www.learn-in-depth.com/
https://www.facebook.com/groups/embedded.system.KS/

3. Requirements

Here are the requirements in the project



<<Requirement>>
Pressure_Detection_LearnInDepth
ID=0
Text="The system shall protect the crew against the high pressure"
Kind="Functional"
Risk="Low"
Reference elements=""

<<Requirement>>
High_Pressure_Detection
ID=1
Text="The system shall check the high pressure in the cabin"
Kind="Functional"
Risk="Low"
Reference elements=""

<<Requirement>>
Crew_Information
ID=2
Text="The system shall infor the crew when the cabin has a too high pressure"
Kind="Functional"
Risk="Low"
Reference elements=""

<<Requirement>>
Optional_Store_pressure
ID=3
Text="The system shall store the pressure values in a flash memory"
Kind="Other"
Risk="Low"
Reference elements=""

<<refine>>

<<Requirement>>
Pressure_Threshold
ID=4
Text="The pressure shall check if the pressure is below a predefined threshold = 20 par"
Kind="Functional"
Risk="Low"
Reference elements=""

<<Requirement>>
Inform_by_LED
ID=6
Text="The system will start alarm by turning LED with alarm duration = 60s"
Kind="Functional"
Risk="Low"
Reference elements=""

<<Requirement>>
write_flash_memory
ID=7
Text="The system shall save the recorded values in a fifo inside flash memory
..."
Kind="Functional"
Risk="Low"

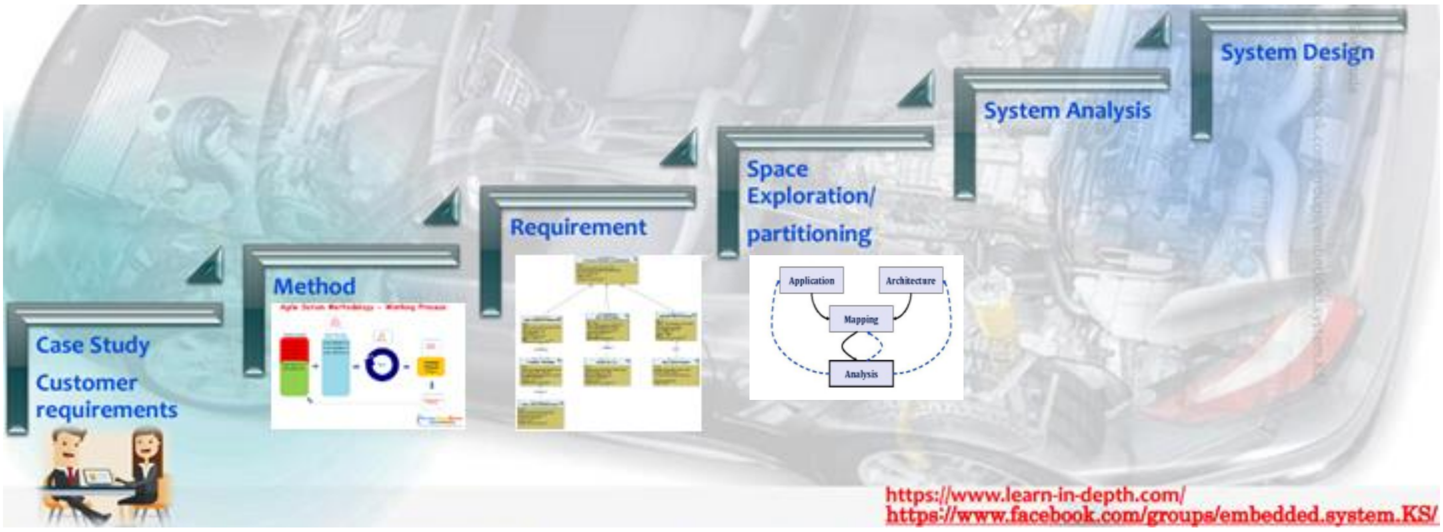
<<refine>>

<<Requirement>>
Read_from_pressureSensor
ID=5
Text="The system shall read from pressure sensor"
Kind="Functional"
Risk="Low"
Reference elements=""

Case Study
Customer requirements
Method
Agile Scrum Methodology - Working Process
Requirement
Space Exploration/ partitioning
System Analysis
System Design
https://www.learn-in-depth.com/
https://www.facebook.com/groups/embedded.system.KS/

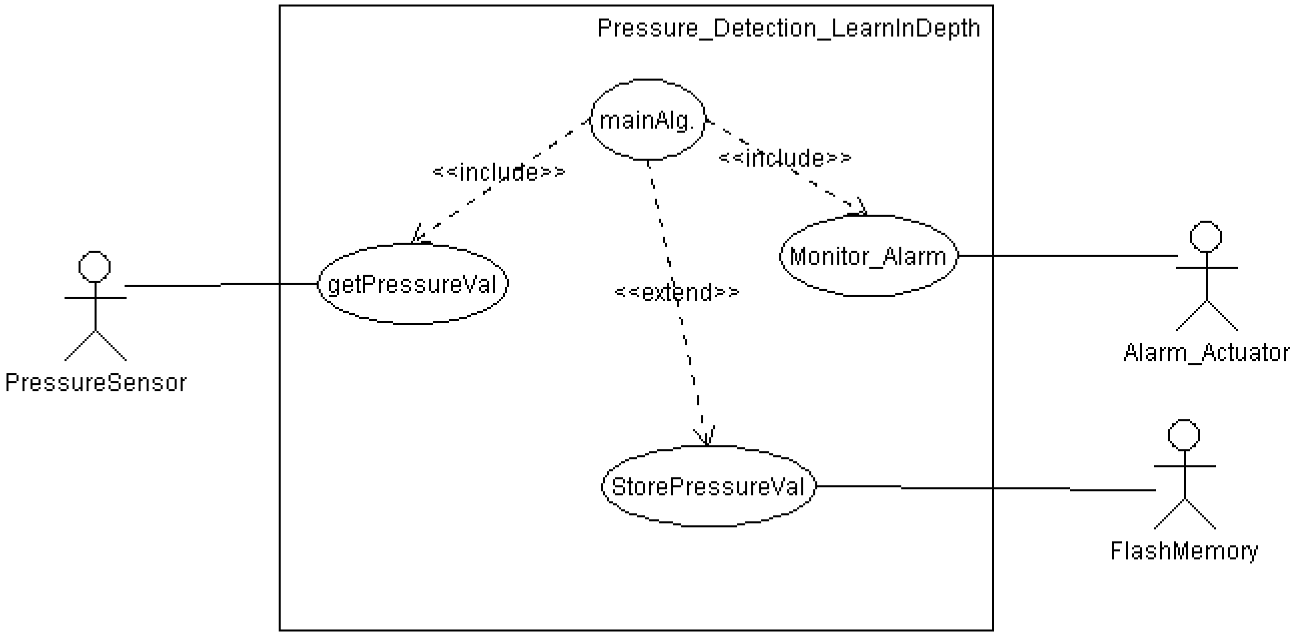
4. Space Exploration/partitioning

After some search I found that the best MC that fit my project is STMF103

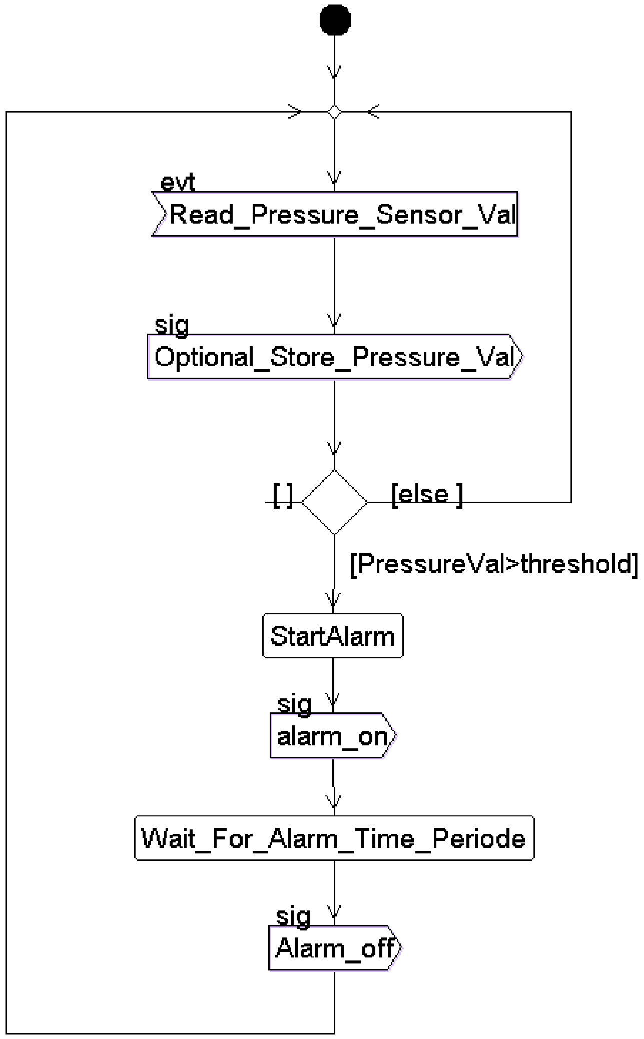


5. System Analysis

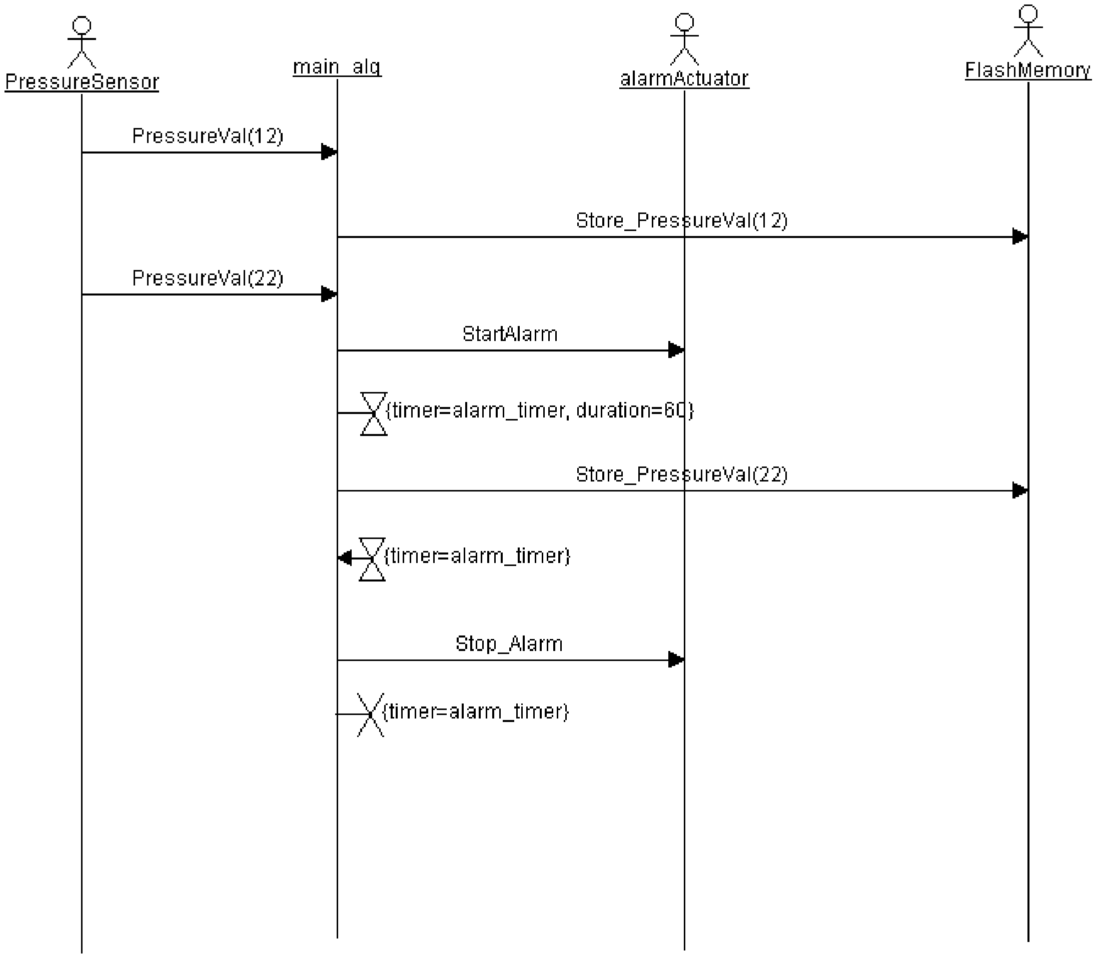
a. UML:Use Case Diagram



b. UML: Activity Diagram

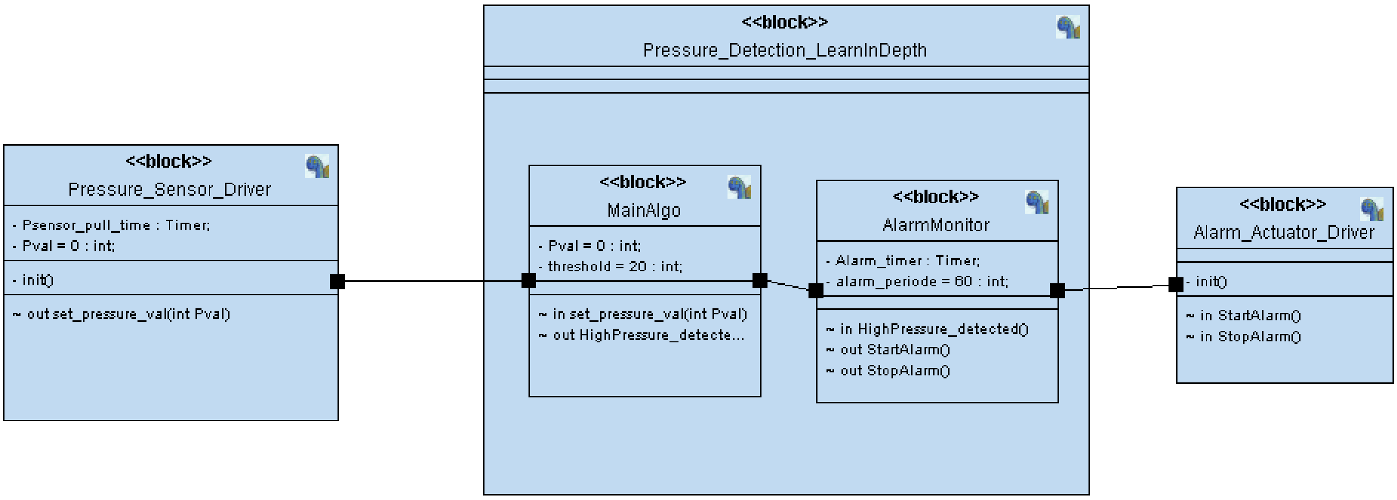


c. UML: Sequence Diagram

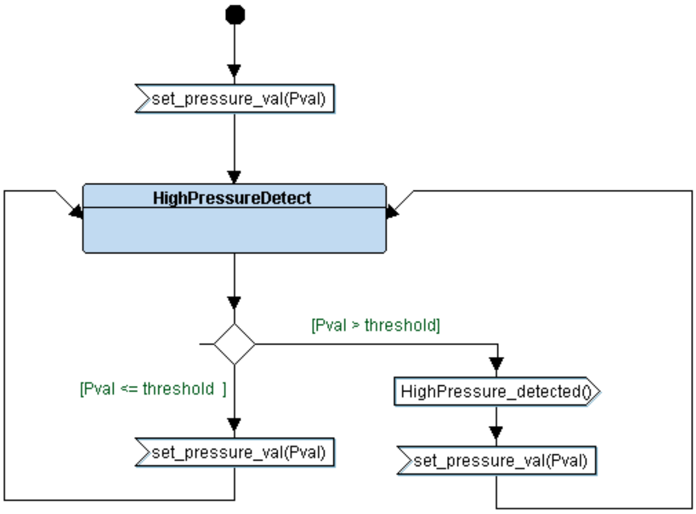


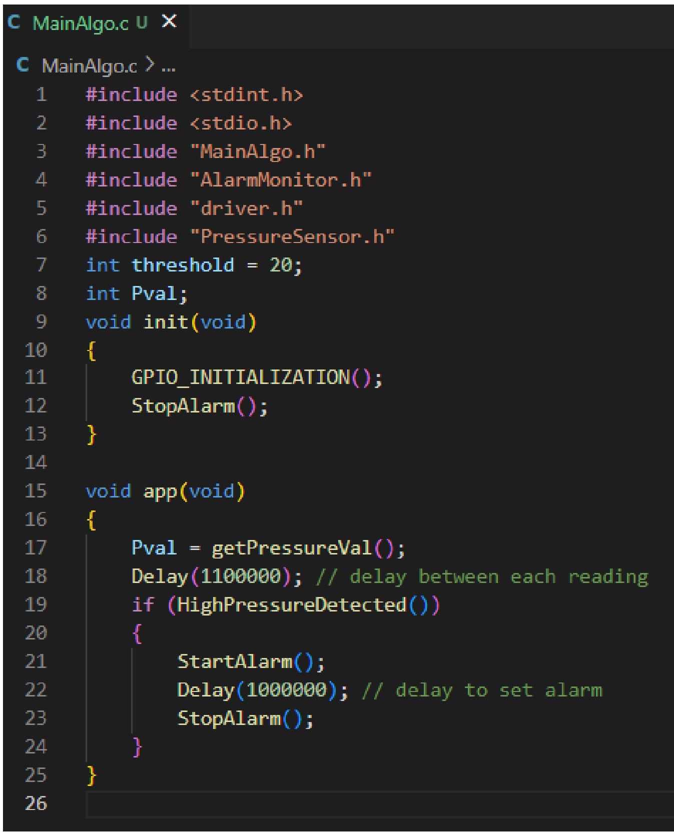
6. System Design

a. Block Diagram of the system



b. Main Algorithm



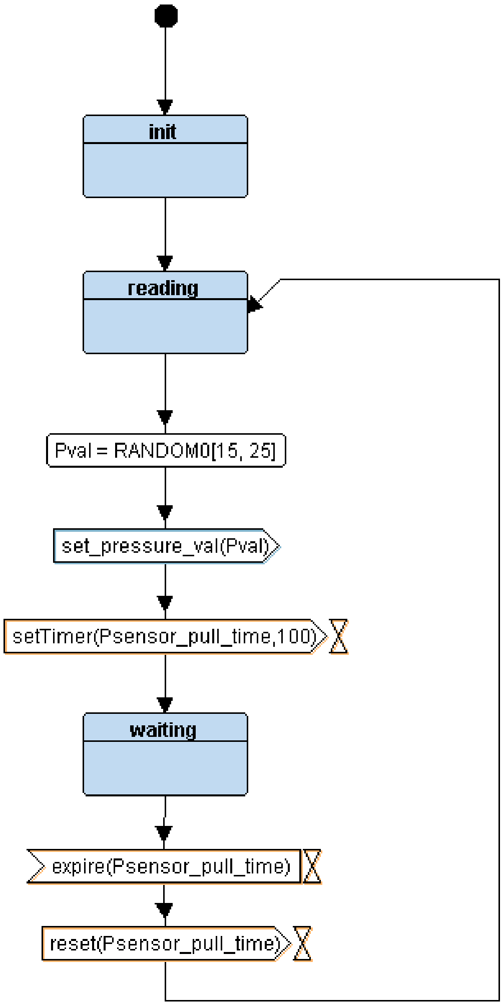


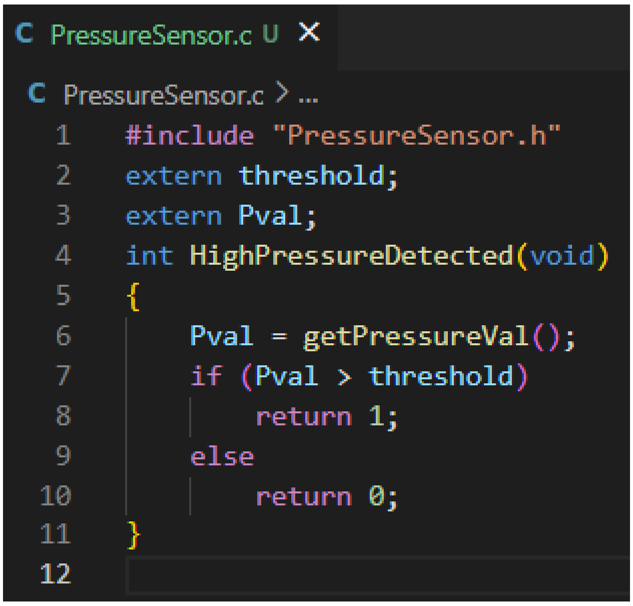
```c
#include <stdint.h>
#include <stdio.h>
#include "MainAlgo.h"
#include "AlarmMonitor.h"
#include "driver.h"
#include "PressureSensor.h"
int threshold = 20;
int Pval;
void init(void)
{
    GPIO_INITIALIZATION();
    StopAlarm();
}

void app(void)
{
    Pval = getPressureVal();
    Delay(1100000); // delay between each reading
    if (HighPressureDetected())
    {
        StartAlarm();
        Delay(1000000); // delay to set alarm
        StopAlarm();
    }
}

```

c. Pressure Sensor Driver

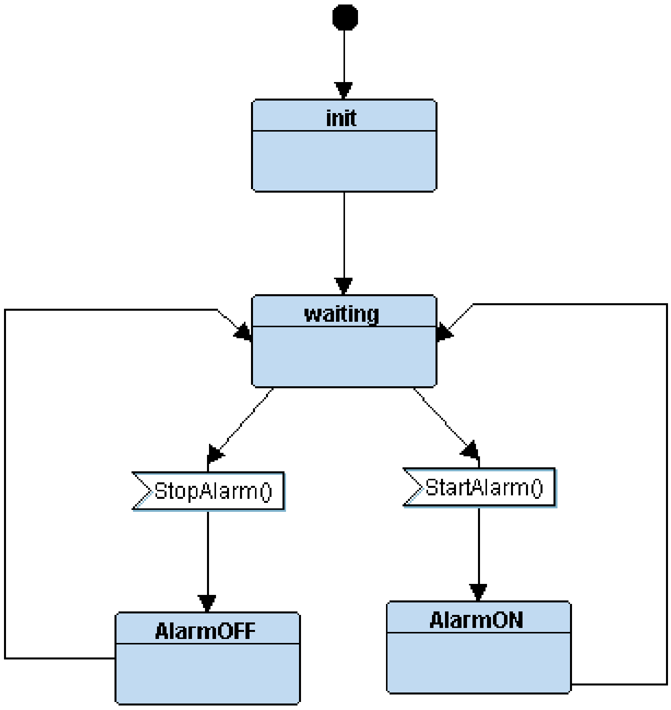


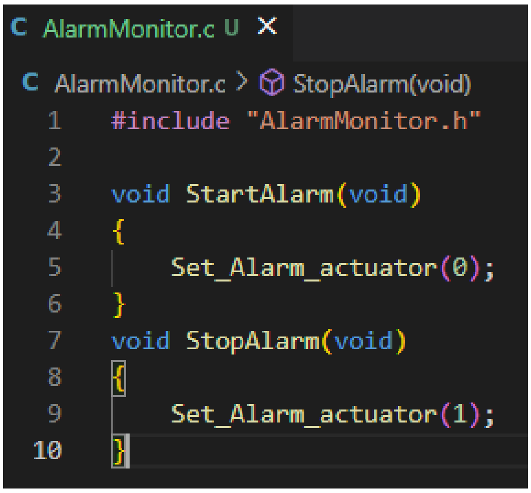


```c
#include "PressureSensor.h"
extern threshold;
extern Pval;
int HighPressureDetected(void)
{
    Pval = getPressureVal();
    if (Pval > threshold)
        return 1;
    else
        return 0;
}

```

d. Alarm Monitor

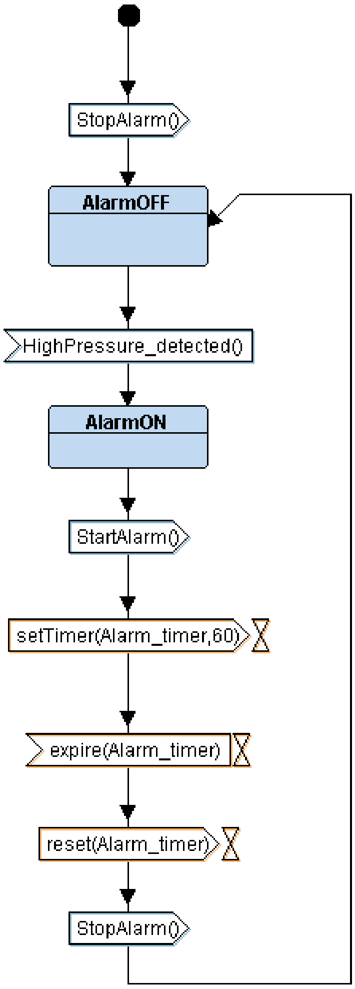


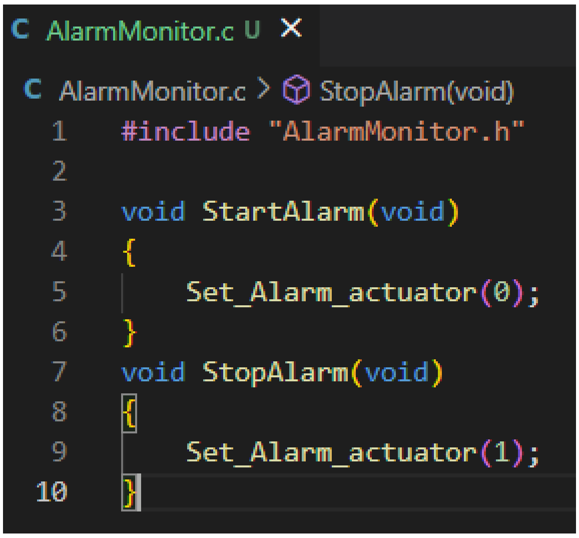


```c
#include "AlarmMonitor.h"

void StartAlarm(void)
{
    Set_Alarm_actuator(0);
}
void StopAlarm(void)
{
    Set_Alarm_actuator(1);
}
```

e. Alarm Actuator Driver

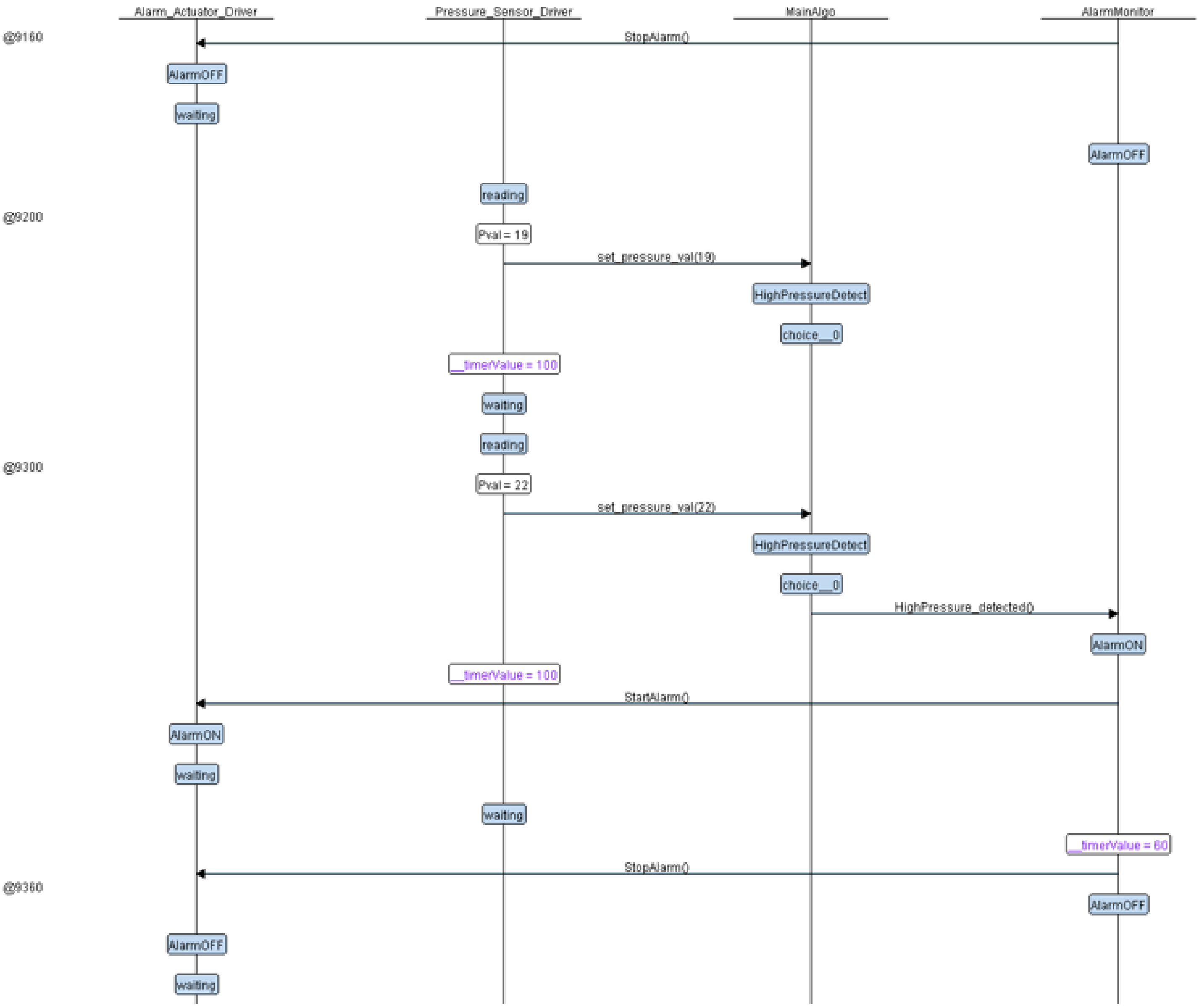


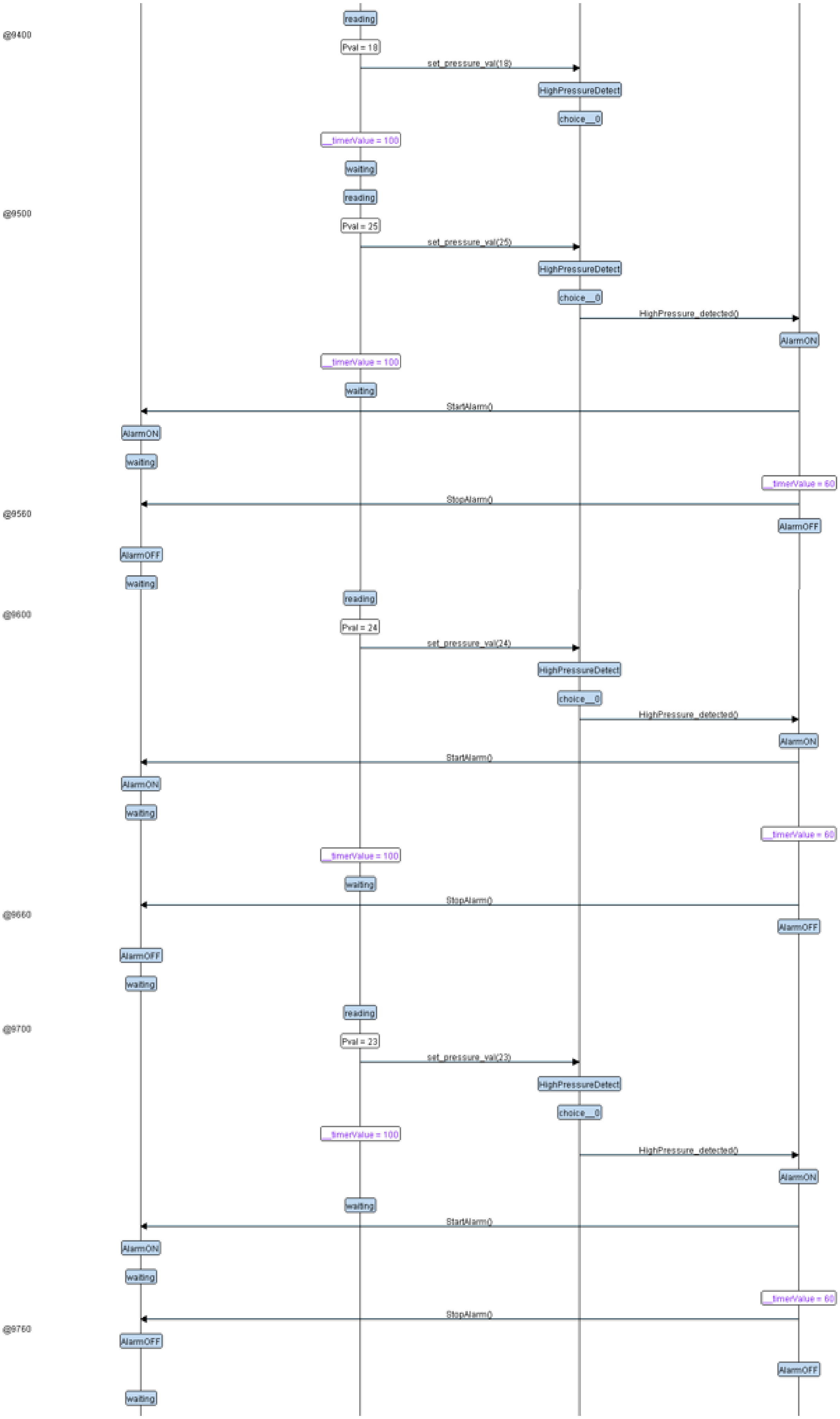


```c
#include "AlarmMonitor.h"

void StartAlarm(void)
{
    Set_Alarm_actuator(0);
}
void StopAlarm(void)
{
    Set_Alarm_actuator(1);
}
```

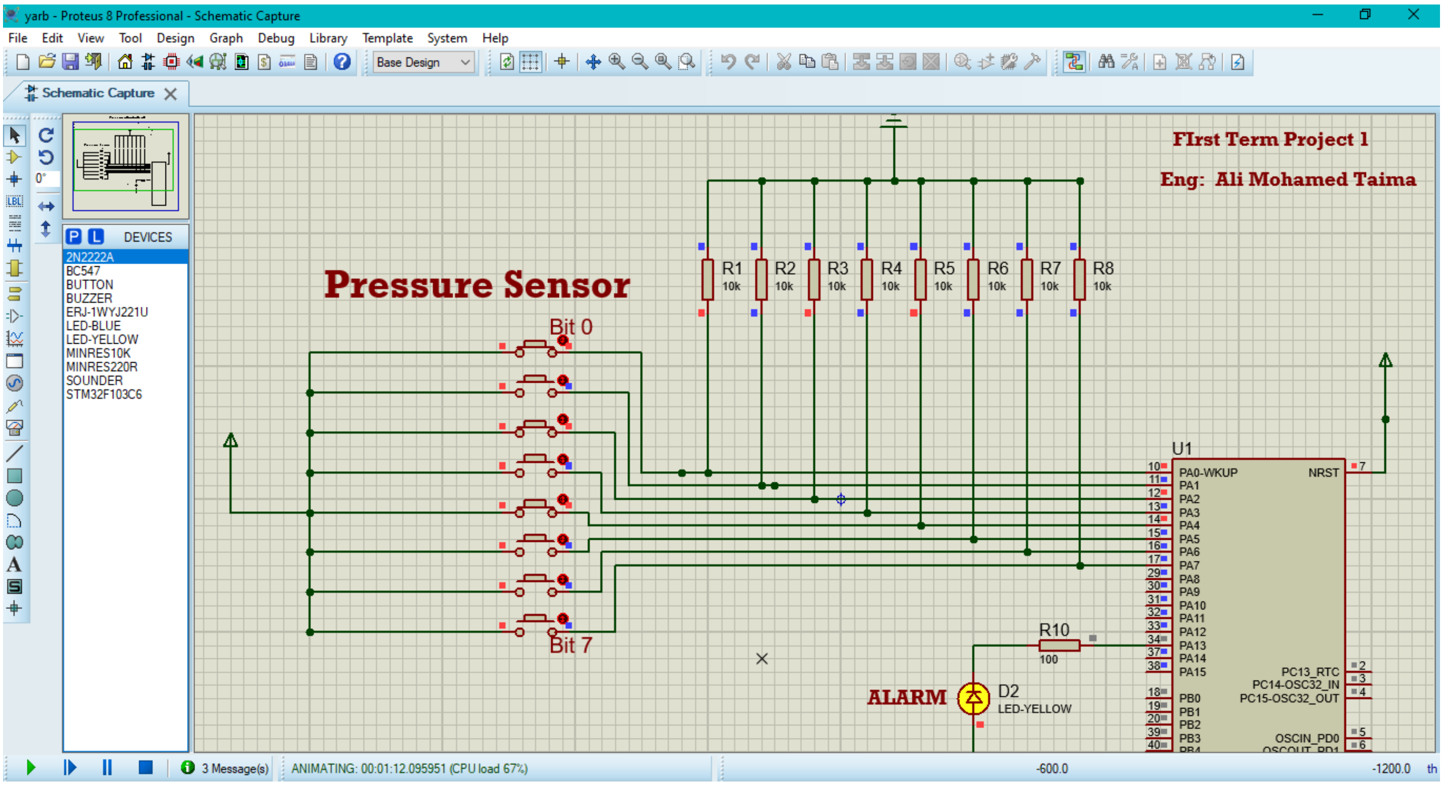
Case Study
Customer requirements
Method
Requirement
Space Exploration/ partitioning
Application
Architecture
Mapping
Analysis
System Analysis
1. Use Case Diagram
2. Activity Diagram
3. Sequence Diagram
System Design
https://www.learn-in-depth.com/
https://www.facebook.com/groups/embedded.system.KS/


Alarm_Actuator_Driver
Pressure_Sensor_Driver
MainAlgo
AlarmMonitor
@9160
StopAlarm()
AlarmOFF
waiting
AlarmOFF
reading
@9200
Pval = 19
set_pressure_val(19)
HighPressureDetect
choice__0
__timerValue = 100
waiting
reading
@9300
Pval = 22
set_pressure_val(22)
HighPressureDetect
choice__0
HighPressure_detected()
AlarmON
__timerValue = 100
StartAlarm()
AlarmON
waiting
waiting
__timerValue = 60
StopAlarm()
@9360
AlarmOFF
AlarmOFF
waiting

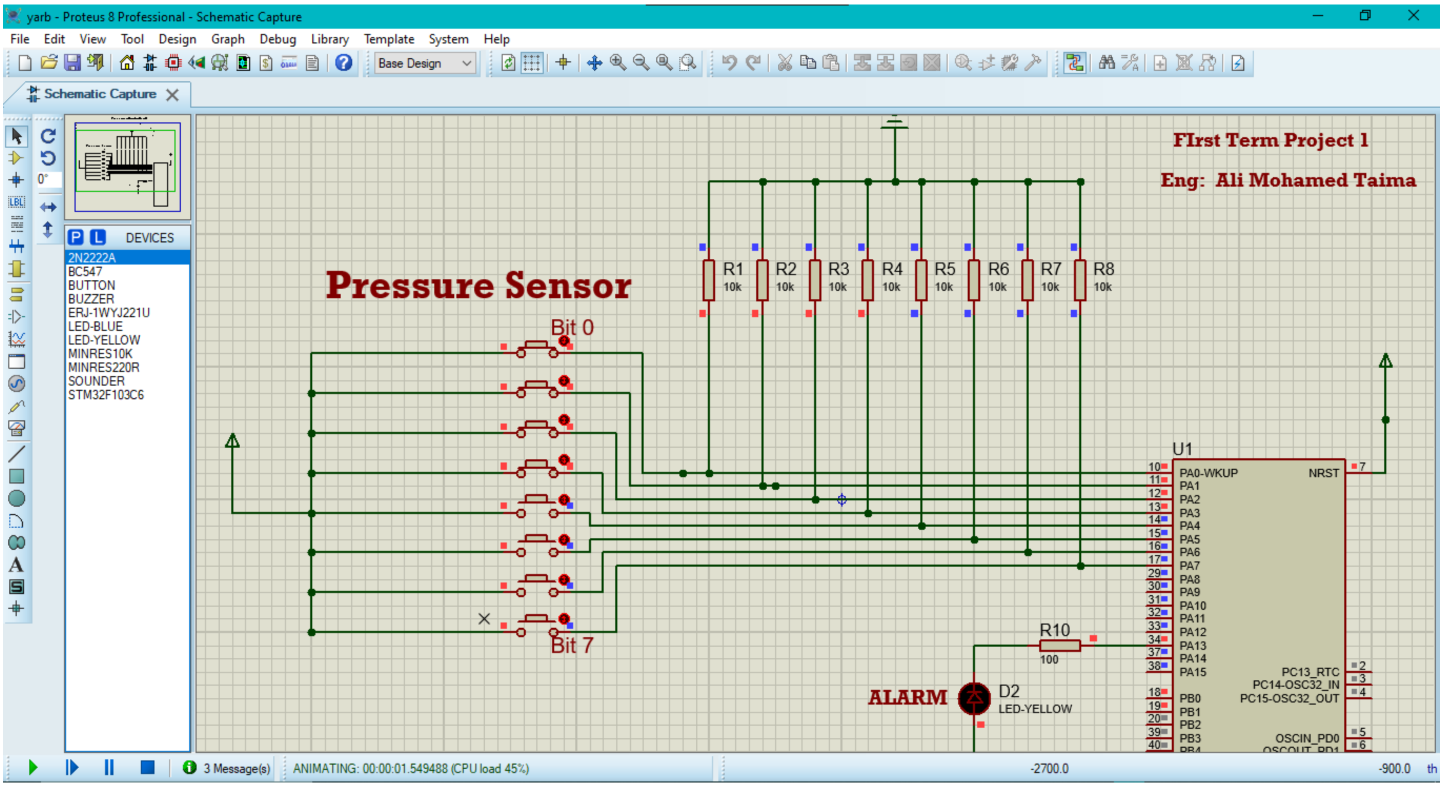

@9400
reading
Pval = 18
set_pressure_val(18)
HighPressureDetect
choice__0
__timerValue = 100
waiting
reading
@9500
Pval = 25
set_pressure_val(25)
HighPressureDetect
choice__0
HighPressure_detected()
AlarmON
__timerValue = 100
waiting
StartAlarm()
AlarmON
waiting
__timerValue = 60
StopAlarm()
@9560
AlarmOFF
AlarmOFF
waiting
reading
@9600
Pval = 24
set_pressure_val(24)
HighPressureDetect
choice__0
HighPressure_detected()
AlarmON
StartAlarm()
AlarmON
waiting
__timerValue = 60
__timerValue = 100
waiting
StopAlarm()
@9660
AlarmOFF
AlarmOFF
waiting
reading
@9700
Pval = 23
set_pressure_val(23)
HighPressureDetect
choice__0
__timerValue = 100
HighPressure_detected()
AlarmON
waiting
StartAlarm()
AlarmON
waiting
__timerValue = 60
StopAlarm()
@9760
AlarmOFF
AlarmOFF
waiting

Simulation test

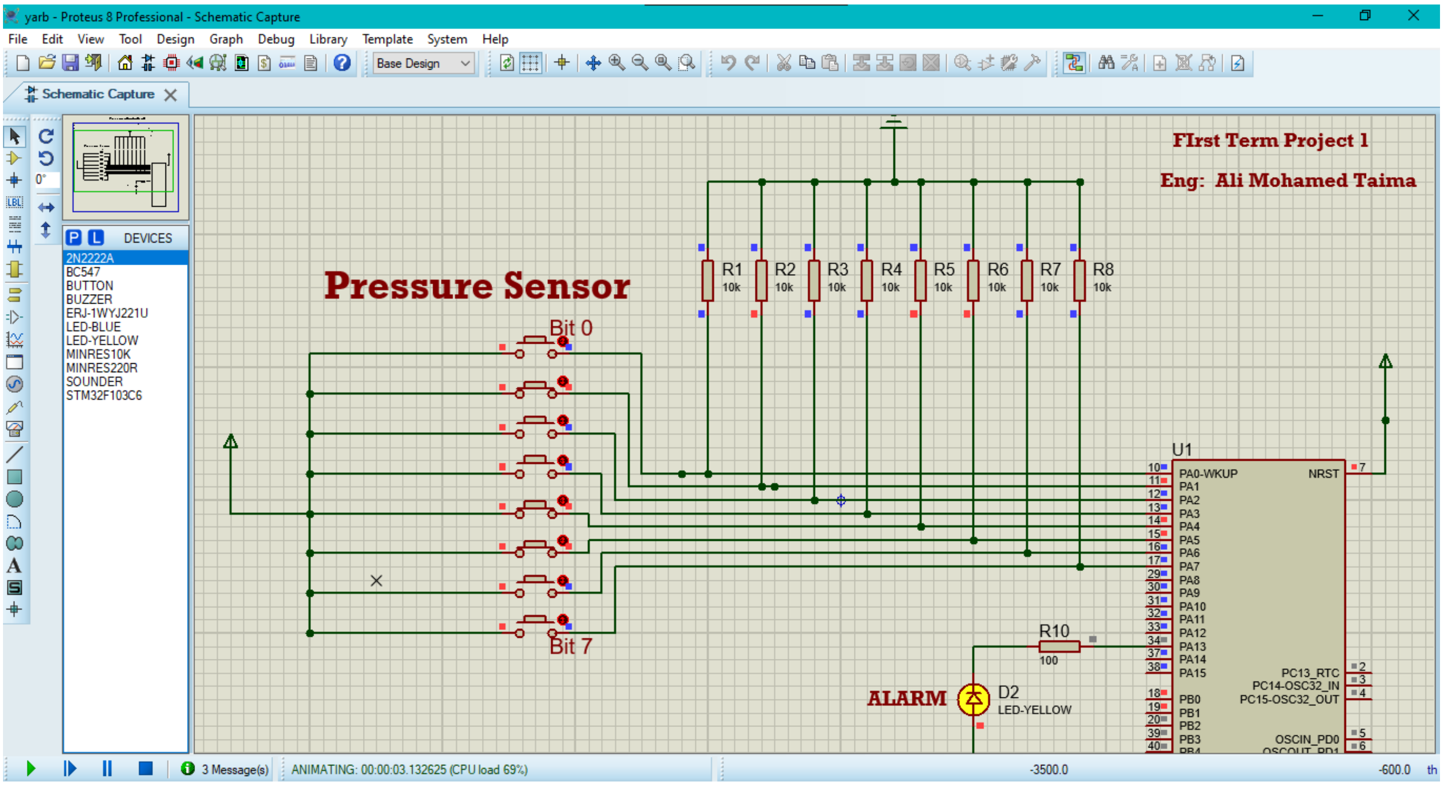
For binary 0001 0101 which is 21 and make High pressure detected the so the alarm get on for 60 seconds



When I try binary 0000 1111 which is 15 (below the threshold 20) so nothing happen and the alarm still off

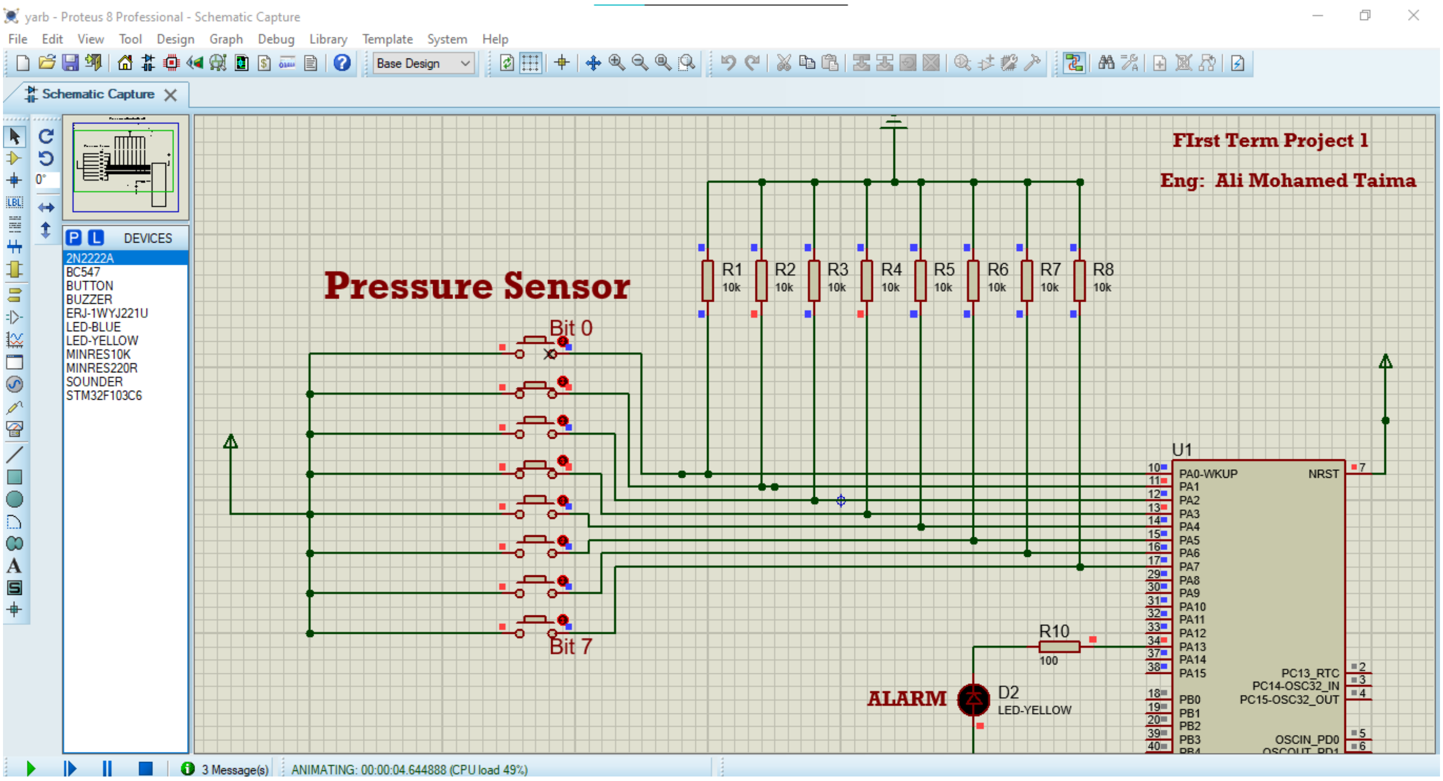


For binary 0011 0010 which is 50 and make High pressure detected the so the alarm get on for 60 seconds



3 Message(s) ANIMATING: 00:00:03.132625 (CPU load 69%)

When I try binary 0000 1010 which is 10 (below the threshold 20) so nothing happen and the alarm still off



3 Message(s) ANIMATING: 00:00:04.644888 (CPU load 49%)

SW analysis

.map file

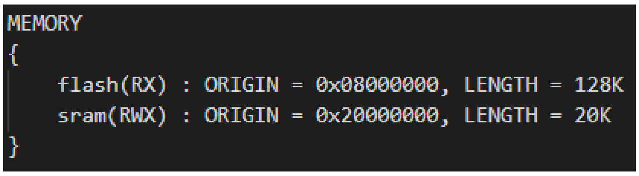
Allocating common symbols

Common symbol	size	file
Pval	0x4	MainAlgo.o

Memory Configuration

Name	Origin	Length	Attributes
flash	0x08000000	0x00020000	xr
sram	0x20000000	0x00005000	xrw
default	0x00000000	0xffffffff	

As I write in the linker file the memory initialized



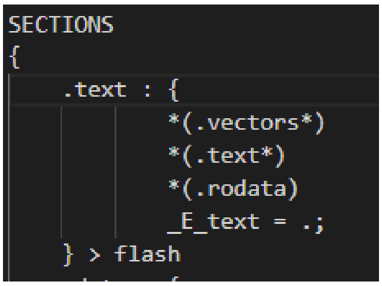
```
MEMORY
{
    flash(RX) : ORIGIN = 0x08000000, LENGTH = 128K
    sram(RWX) : ORIGIN = 0x20000000, LENGTH = 20K
}
```

Linker script and memory map

I compine the different sections with each other as in my linker

```
.text          0x08000000     0x2a0
 *(.vectors*)
 .vectors       0x08000000      0x1c startup.o
                0x08000000               vectors
 *(.text*)
 .text          0x0800001c      0xbc startup.o
                0x0800001c               NMI_Handler
                0x0800001c               Bus_Fault_Handler
                0x0800001c               Usage_Fault_Handler
                0x0800001c               Default_Handler
```

.text section



```
SECTIONS
{
    .text : {
            *(.vectors*)
            *(.text*)
            *(.rodata)
            _E_text = .;
    } > flash
```

```
             0x0800001c          H_fault_Handler
             0x0800001c          NM_Fault_Handler
             0x08000028          Reset_Handler
.text        0x080000d8     0x10 main.o
             0x080000d8          main
.text        0x080000e8     0x3c PressureSensor.o
             0x080000e8          HighPressureDetected
.text        0x08000124     0x50 MainAlgo.o
             0x08000124          init
             0x08000134          app
.text        0x08000174    0x10c driver.o
             0x08000174          Delay
             0x08000198          getPressureVal
             0x080001b0          Set_Alarm_actuator
             0x08000200          GPIO_INITIALIZATION
.text        0x08000280     0x20 AlarmMonitor.o
             0x08000280          StartAlarm
             0x08000290          StopAlarm
*(.rodata)
             0x080002a0          _E_text = .
```

Cont.
.text section

```
.glue_7       0x080002a0      0x0
.glue_7       0x00000000      0x0 linker stubs

.glue_7t      0x080002a0      0x0
.glue_7t      0x00000000      0x0 linker stubs

.vfp11_veneer 0x080002a0      0x0
.vfp11_veneer 0x00000000      0x0 linker stubs
```

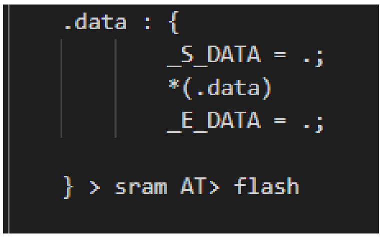
```
.v4_bx        0x080002a0      0x0
 .v4_bx       0x00000000      0x0 linker stubs

.iplt       0x080002a0      0x0
 .iplt      0x00000000      0x0 startup.o

.rel.dyn      0x080002a0      0x0
 .rel.iplt   0x00000000      0x0 startup.o

.data        0x20000000      0x4 load address 0x080002a0
           0x20000000            _S_DATA = .
 *(.data)
 .data       0x20000000      0x0 startup.o
 .data       0x20000000      0x0 main.o
 .data       0x20000000      0x0 PressureSensor.o
 .data       0x20000000      0x4 MainAlgo.o
           0x20000000            threshold
 .data       0x20000004      0x0 driver.o
 .data       0x20000004      0x0 AlarmMonitor.o
           0x20000004            _E_DATA = .
```

.data section

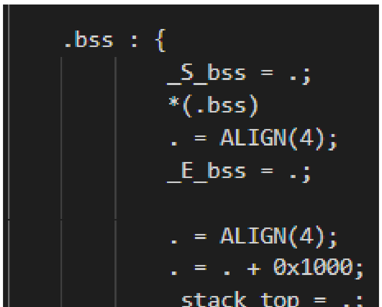


```
.data : {
        _S_DATA = .;
        *(.data)
        _E_DATA = .;

} > sram AT> flash
```

```
.igot.plt     0x20000004      0x0 load address 0x080002a4
 .igot.plt    0x00000000      0x0 startup.o

.bss        0x20000004    0x1004 load address 0x080002a4
           0x20000004            _S_bss = .
 *(.bss)
 .bss       0x20000004      0x0 startup.o
```



```
.bss : {
        _S_bss = .;
        *(.bss)
        . = ALIGN(4);
        _E_bss = .;

        . = ALIGN(4);
        . = . + 0x1000;
        _stack_top = .;
```

```
.bss        0x20000004      0x0 main.o
.bss        0x20000004      0x0 PressureSensor.o
.bss        0x20000004      0x0 MainAlgo.o
.bss        0x20000004      0x0 driver.o
.bss        0x20000004      0x0 AlarmMonitor.o
            0x20000004            . = ALIGN (0x4)
            0x20000004            _E_bss = .
            0x20000004            . = ALIGN (0x4)
            0x20001004            . = (. + 0x1000)
*fill*      0x20000004    0x1000
            0x20001004            _stack_top = .
COMMON        0x20001004      0x4 MainAlgo.o
            0x20001004            Pval
```

.bss section

Stack top

```
LOAD startup.o
LOAD main.o
LOAD PressureSensor.o
LOAD MainAlgo.o
LOAD driver.o
LOAD AlarmMonitor.o
OUTPUT(Project1_pressure_controller.elf elf32-littlearm)

.debug_info    0x00000000     0x496
 .debug_info   0x00000000     0x168 startup.o
 .debug_info   0x00000168      0x46 main.o
 .debug_info   0x000001ae      0x75 PressureSensor.o
 .debug_info   0x00000223      0xca MainAlgo.o
 .debug_info   0x000002ed     0x103 driver.o
 .debug_info   0x000003f0      0xa6 AlarmMonitor.o

.debug_abbrev  0x00000000     0x2a5
 .debug_abbrev 0x00000000      0xc2 startup.o
 .debug_abbrev 0x000000c2      0x37 main.o
 .debug_abbrev 0x000000f9      0x62 PressureSensor.o
 .debug_abbrev 0x0000015b      0x6b MainAlgo.o
 .debug_abbrev 0x000001c6      0x9d driver.o
 .debug_abbrev 0x00000263      0x42 AlarmMonitor.o
```

```
.debug_loc     0x00000000     0x234
 .debug_loc    0x00000000      0x64 startup.o
 .debug_loc    0x00000064      0x2c main.o
 .debug_loc    0x00000090      0x2c PressureSensor.o
 .debug_loc    0x000000bc      0x58 MainAlgo.o
 .debug_loc    0x00000114      0xc8 driver.o
 .debug_loc    0x000001dc      0x58 AlarmMonitor.o

.debug_aranges  0x00000000      0xc0
 .debug_aranges
               0x00000000      0x20 startup.o
 .debug_aranges
               0x00000020      0x20 main.o
 .debug_aranges
               0x00000040      0x20 PressureSensor.o
 .debug_aranges
               0x00000060      0x20 MainAlgo.o
 .debug_aranges
               0x00000080      0x20 driver.o
 .debug_aranges
               0x000000a0      0x20 AlarmMonitor.o

.debug_line    0x00000000     0x24e
 .debug_line   0x00000000      0xad startup.o
 .debug_line   0x000000ad      0x3f main.o
 .debug_line   0x000000ec      0x43 PressureSensor.o
 .debug_line   0x0000012f      0x45 MainAlgo.o
 .debug_line   0x00000174      0x99 driver.o
 .debug_line   0x0000020d      0x41 AlarmMonitor.o

.debug_str     0x00000000     0x1fc
 .debug_str    0x00000000     0x13b startup.o
                              0x16f (size before relaxing)
 .debug_str    0x0000013b       0xc main.o
                               0x65 (size before relaxing)
 .debug_str    0x00000147      0x44 PressureSensor.o
                               0x9d (size before relaxing)
 .debug_str    0x0000018b      0x10 MainAlgo.o
                              0x10a (size before relaxing)
 .debug_str    0x0000019b      0x3d driver.o
                              0x140 (size before relaxing)
 .debug_str    0x000001d8      0x24 AlarmMonitor.o
                               0x10f (size before relaxing)

.comment       0x00000000      0x11
 .comment      0x00000000      0x11 startup.o
                               0x12 (size before relaxing)
 .comment      0x00000000      0x12 main.o
 .comment      0x00000000      0x12 PressureSensor.o
 .comment      0x00000000      0x12 MainAlgo.o
 .comment      0x00000000      0x12 driver.o
 .comment      0x00000000      0x12 AlarmMonitor.o

.ARM.attributes
               0x00000000      0x33
 .ARM.attributes
               0x00000000      0x33 startup.o
 .ARM.attributes
               0x00000033      0x33 main.o
 .ARM.attributes
               0x00000066      0x33 PressureSensor.o
 .ARM.attributes
               0x00000099      0x33 MainAlgo.o
 .ARM.attributes
               0x000000cc      0x33 driver.o
 .ARM.attributes
               0x000000ff      0x33 AlarmMonitor.o

.debug_frame   0x00000000      0x1a8
 .debug_frame  0x00000000      0x48 startup.o
 .debug_frame  0x00000048      0x2c main.o
 .debug_frame  0x00000074      0x2c PressureSensor.o
 .debug_frame  0x000000a0      0x48 MainAlgo.o
 .debug_frame  0x000000e8      0x78 driver.o
 .debug_frame  0x00000160      0x48 AlarmMonitor.o
```