

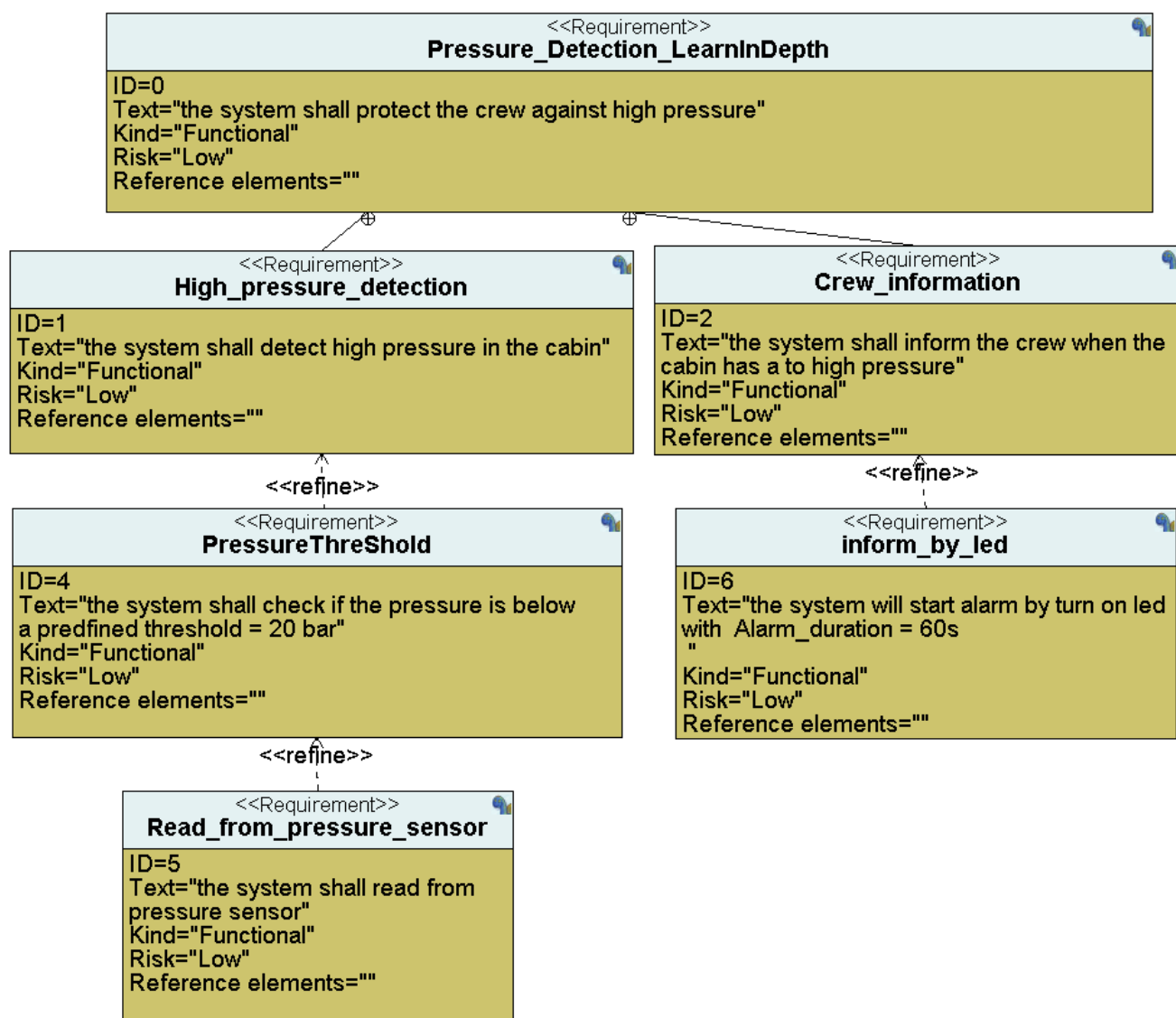
Pressure Controller

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

1-Requirements Diagram



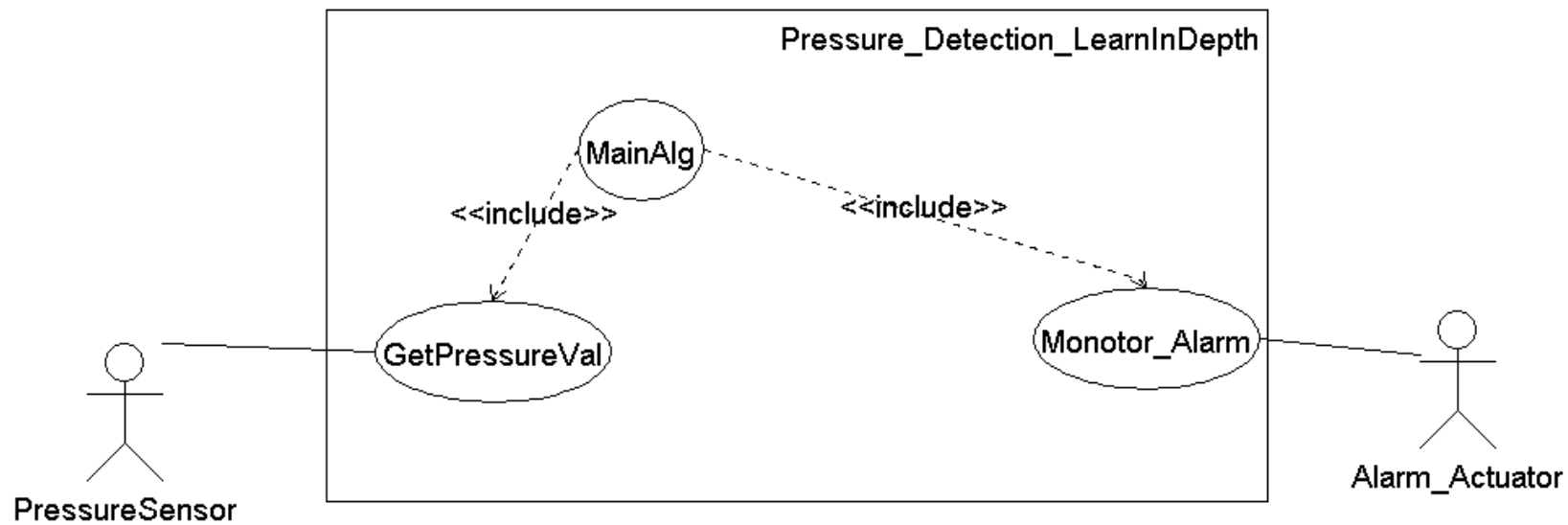
2-System Analysis

1. Use Case Diagram

this diagram describes main functionality in system ,and how the system interacts with outside system which boundary of system..

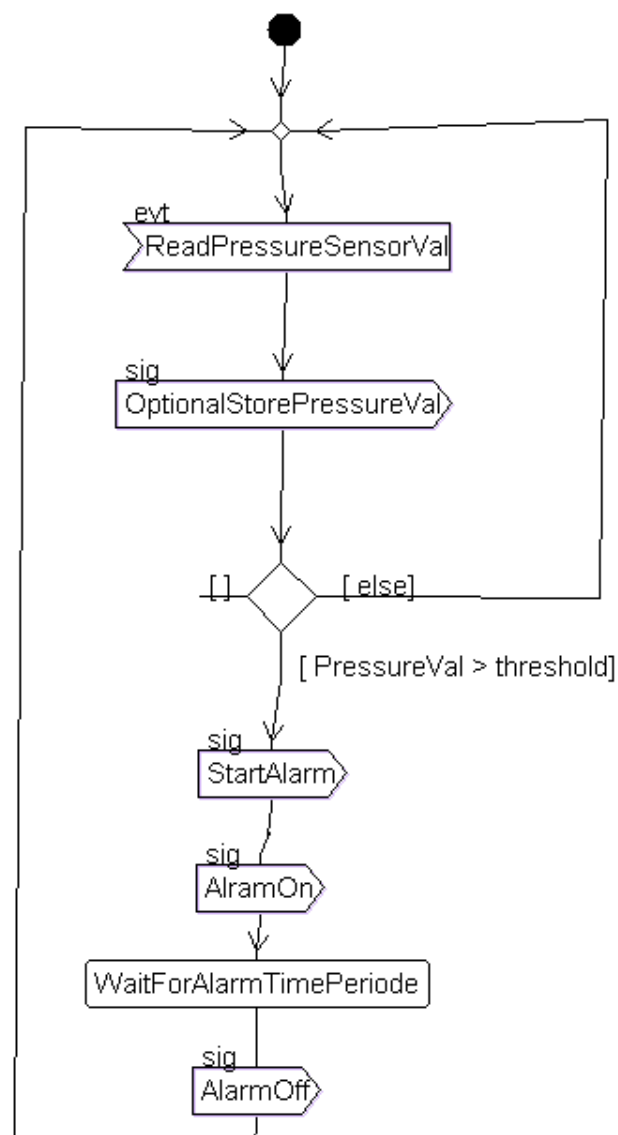
➤ **Pressure Sensor** to get pressure value ,

- **Alarm monitor** to occur the action of high pressure on it.



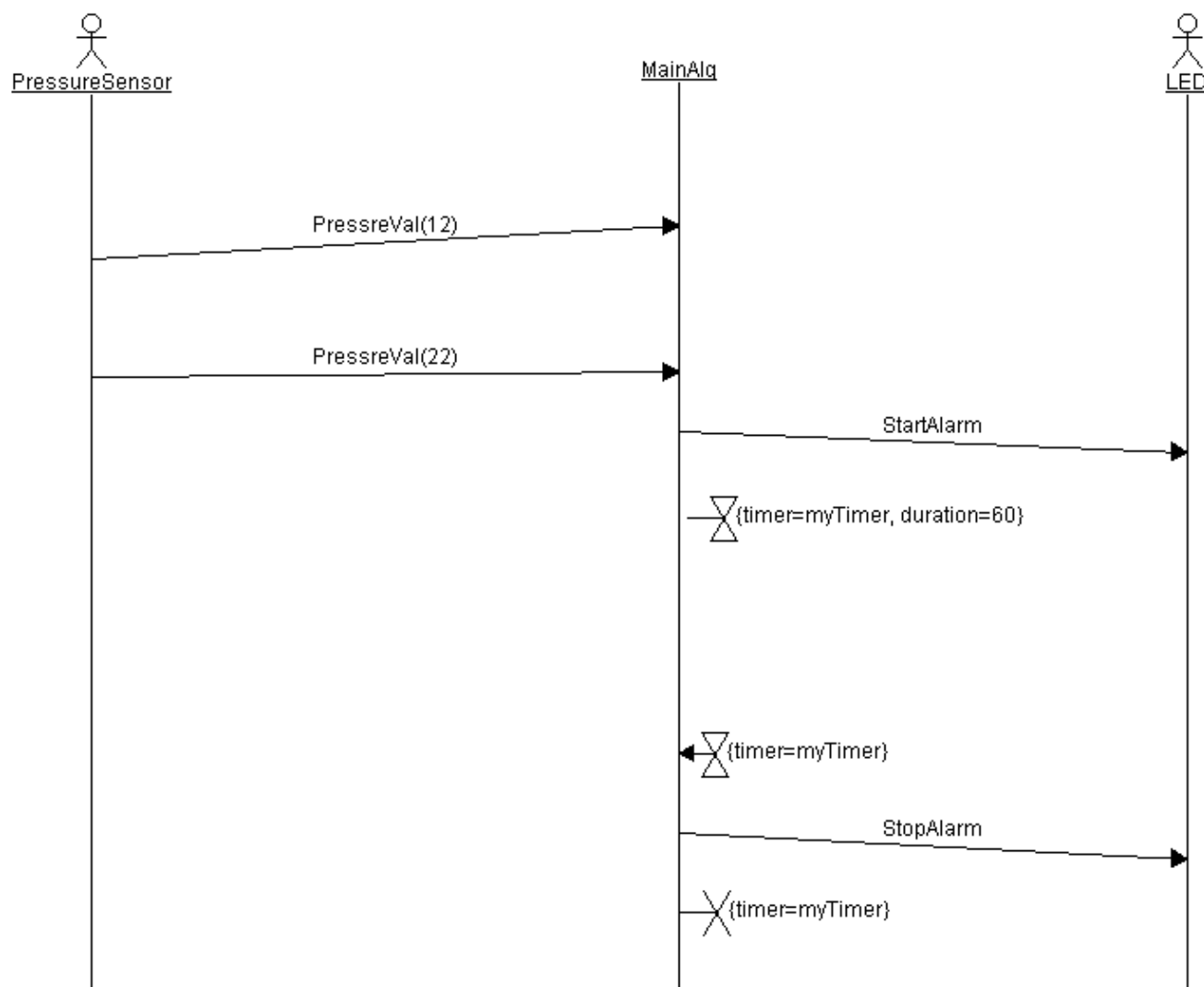
2. Activity Diagram

Activity diagrams describe the workflow behavior of a system



2. Sequence Diagram

- **case 1:** Pressure sensor get '12bar'
it be sent to main Algorithm this value is less than '20 bar'
not action
- **case 2:** Pressure sensor get '22bar'
it be sent to main Algorithm this value is bigger than '20 bar'
, then led on ,waiting 2s and led of and so on while the value
stile bigger than '20bar'.

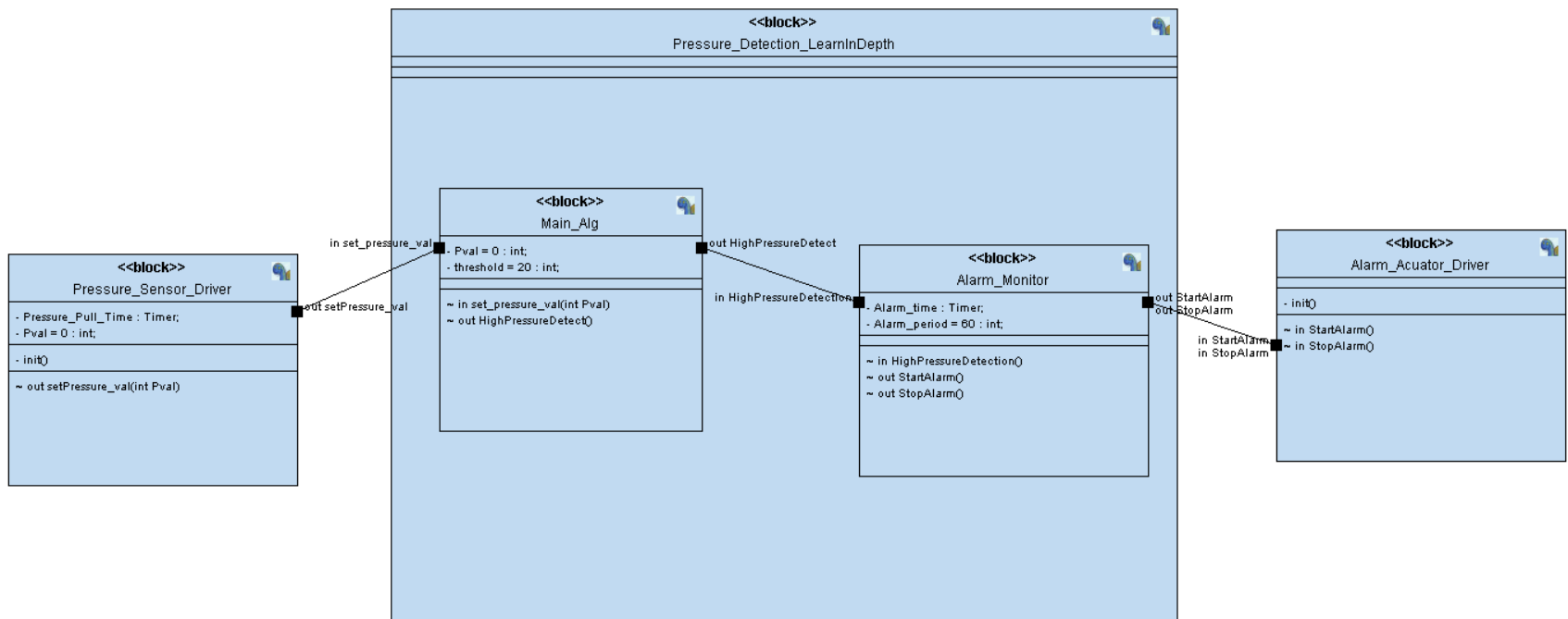


4- Block Diagram

this diagram describes the module in system which

- **Pressure Sensor Driver**
- **Main Algorithm**

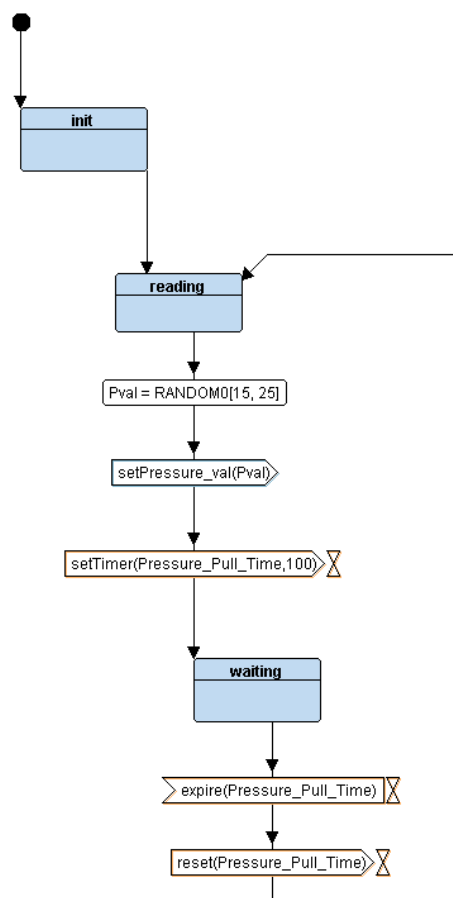
- Alarm Monitor
- Alarm Actuator Driver



5-State Machine

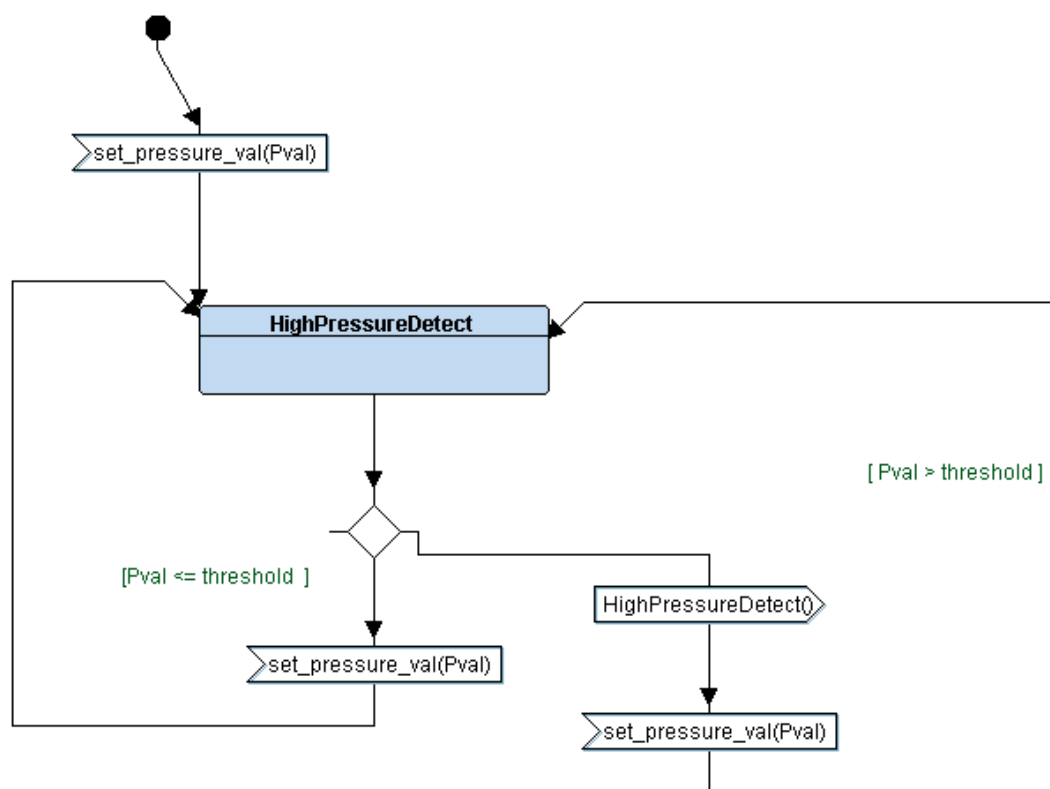
1. Pressure Sensor Driver

start initialization and get values of Pressure with period 100s and sent it to main Algorithm to process it

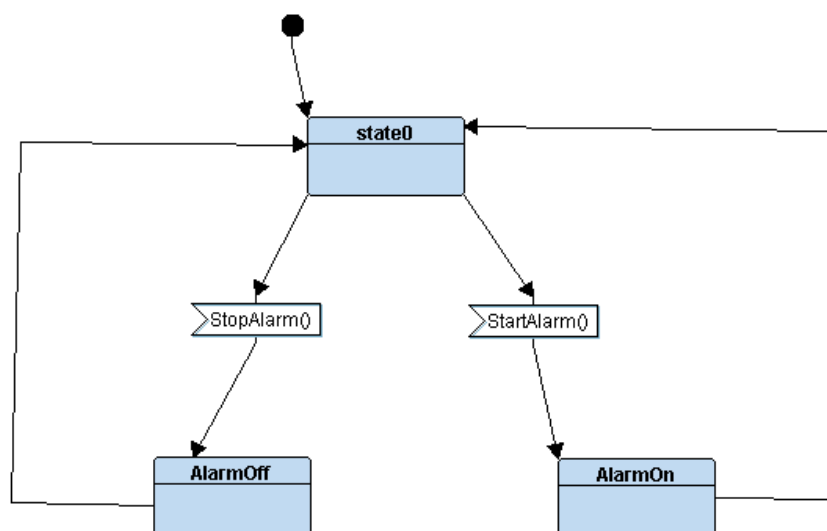


2. Main Algorithm

it receives the pressure value from pressure sensor and test it if it less than 20bar it sent to *high pressure Detected* function and if the value bigger than 20 bar it repeats revise values.

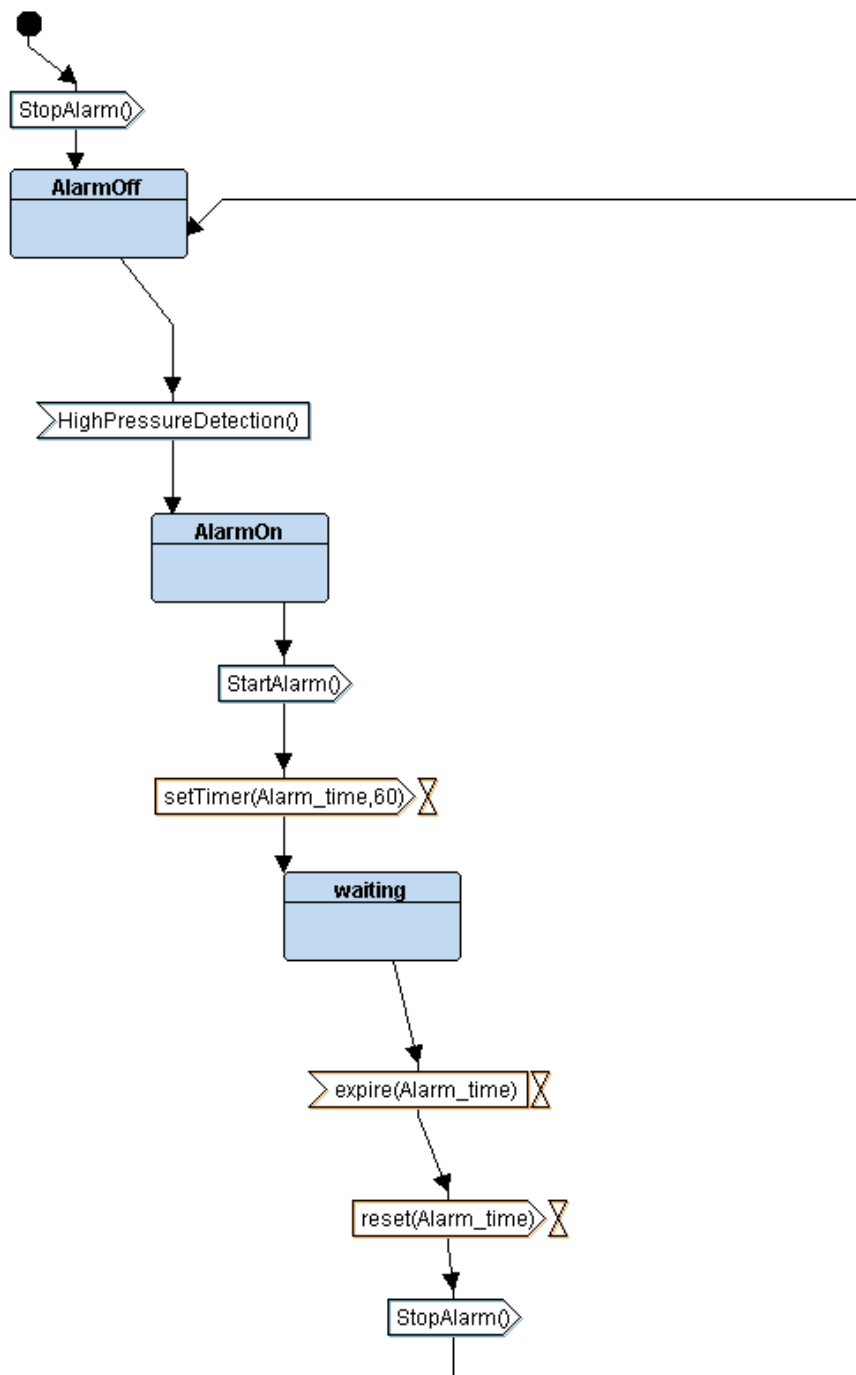


3. State Machine Alarm Actuator Driver

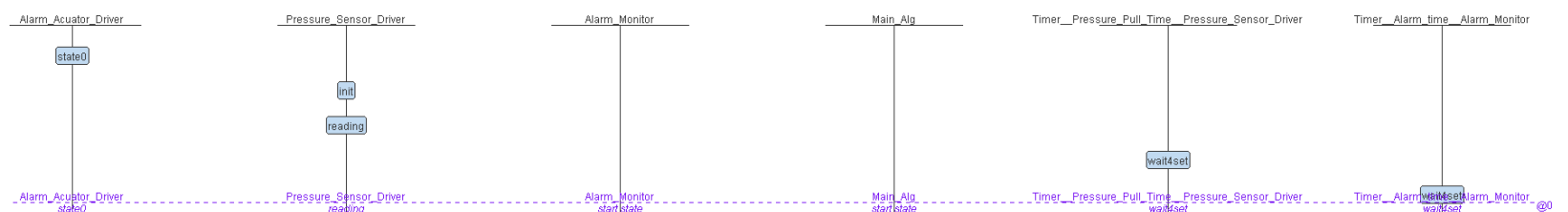


4. State Machine of Alarm Monitor

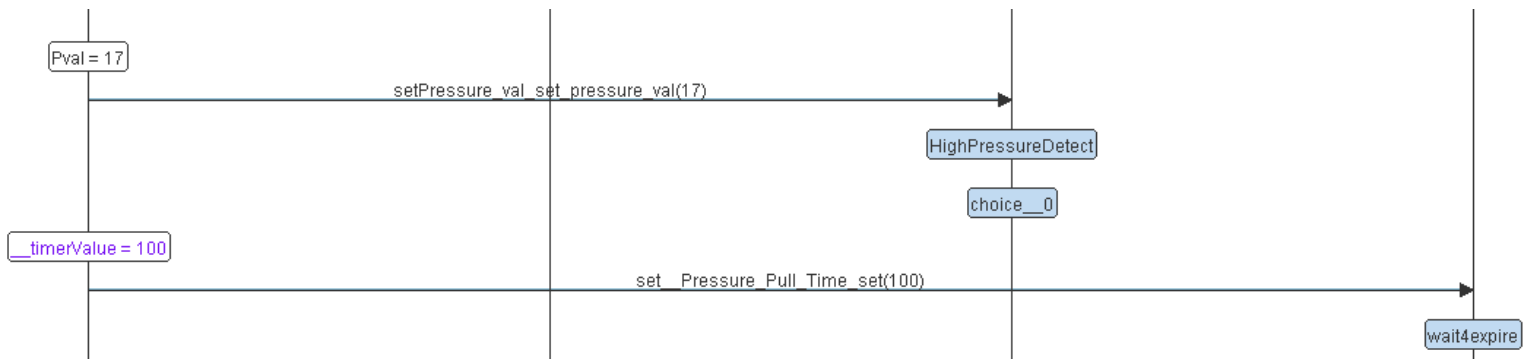
it starts the LED is off if it detects high pressure value from main Algorithm it makes LED is on and waiting 60s and make LED off and repeat checking again.



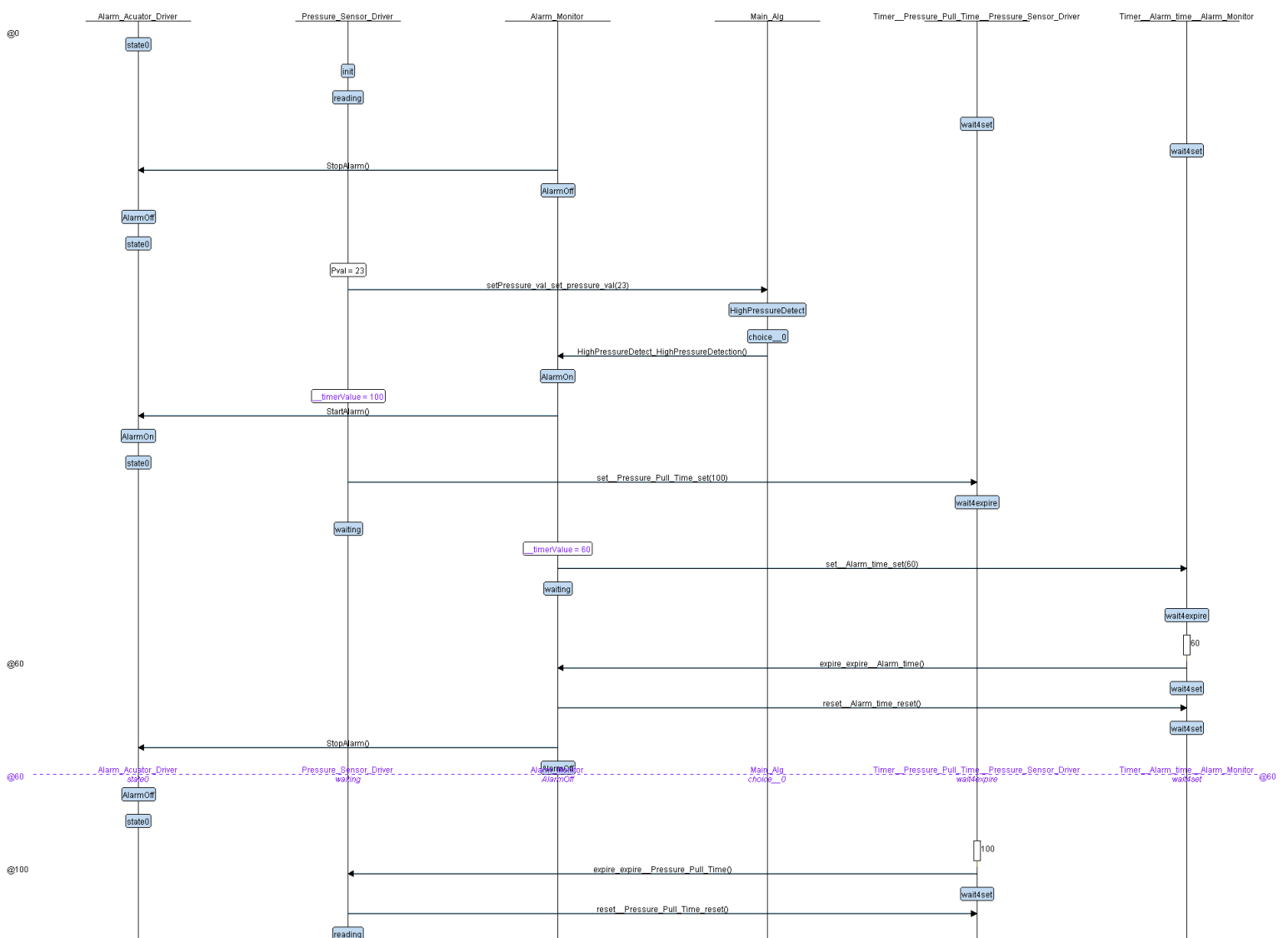
Simulation to show sequence of system



case 1: the value is less (17) than 20 bar



case 2 : the pressure value bigger(23) than 20 bar



Then I'm developed the project and this is result

1-Code sections

1. Startup Sections.o

```
PS D:\study\GNU-Tools-ARM-Embedded\7 2017-q4-major\bin> .\arm-none-eabi-objdump.exe -h startup.o
startup.o:      file format elf32-littlearm

Sections:
Idx Name          Size      VMA           LMA           File off  Algn
  0 .text          00000008  00000000  00000000  00000034  2**1
    CONTENTS, ALLOC, LOAD, RELOC, READONLY, CODE
  1 .data           00000000  00000000  00000000  0000003c  2**0
    CONTENTS, ALLOC, LOAD, DATA
  2 .bss            00000000  00000000  00000000  0000003c  2**0
    ALLOC
  3 .vectors        00000050  00000000  00000000  0000003c  2**0
    CONTENTS, RELOC, READONLY
  4 .ARM.attributes 00000021  00000000  00000000  0000008c  2**0
    CONTENTS, READONLY
```

2.Sections Of Driver.o

```
PS D:\study\GNU-Tools-ARM-Embedded\7 2017-q4-major\bin> .\arm-none-eabi-objdump.exe -h driver.o
driver.o:      file format elf32-littlearm

Sections:
Idx Name          Size      VMA           LMA           File off  Algn
  0 .text          000000c4  00000000  00000000  00000034  2**2
    CONTENTS, ALLOC, LOAD, READONLY, CODE
  1 .data           00000000  00000000  00000000  000000f8  2**0
    CONTENTS, ALLOC, LOAD, DATA
  2 .bss            00000000  00000000  00000000  000000f8  2**0
    ALLOC
  3 .comment        0000007f  00000000  00000000  000000f8  2**0
    CONTENTS, READONLY
  4 .ARM.attributes 00000033  00000000  00000000  00000177  2**0
```

3.Sections Of Main.o

```
PS D:\study\GNU-Tools-ARM-Embedded\7 2017-q4-major\bin> .\arm-none-eabi-objdump.exe -h main.o
main.o:      file format elf32-littlearm

Sections:
Idx Name          Size      VMA           LMA           File off  Algn
  0 .text          00000030  00000000  00000000  00000034  2**2
    CONTENTS, ALLOC, LOAD, RELOC, READONLY, CODE
  1 .data           00000000  00000000  00000000  00000064  2**0
    CONTENTS, ALLOC, LOAD, DATA
  2 .bss            00000000  00000000  00000000  00000064  2**0
    ALLOC
  3 .comment        0000007f  00000000  00000000  00000064  2**0
    CONTENTS, READONLY
  4 .ARM.attributes 00000033  00000000  00000000  000000e3  2**0
    CONTENTS, READONLY
PS D:\study\GNU-Tools-ARM-Embedded\7 2017-q4-major\bin>
```


4.Sections Of Alarm_control.o

```
PS D:\study\GNU-Tools-ARM-Embedded\7 2017-q4-major\bin> .\arm-none-eabi-objdump.exe -h Alarm_control.o
Alarm_control.o:      file format elf32-littlearm

Sections:
Idx Name              Size      VMA           LMA           File off  Algn
  0 .text              00000024  00000000  00000000  00000034  2**2
  1 .data              00000000  00000000  00000000  00000058  2**0
  2 .bss              00000000  00000000  00000000  00000058  2**0
  3 .comment           0000007f  00000000  00000000  00000058  2**0
  4 .ARM.attributes    00000033  00000000  00000000  000000d7  2**0
```

2-Symbols of code

1-Symbols of startup.o

```
PS D:\study\GNU-Tools-ARM-Embedded\7 2017-q4-major\bin> .\arm-none-eabi-nm.exe .\startup.o
00000000 t _reset
          U main
00000006 t Vector_handler
```

2-Symbols of driver.o

```
PS D:\study\GNU-Tools-ARM-Embedded\7 2017-q4-major\bin> .\arm-none-eabi-nm.exe .\driver.o
00000000 T Delay
00000020 T getPressureVal
00000074 T GPIO_INITIALIZATION
00000038 T Set_Alarm_actuator
```

3-Symbols of Alarm control

```
PS D:\study\GNU-Tools-ARM-Embedded\7 2017-q4-major\bin> .\arm-none-eabi-nm.exe .\Alarm_control.o
          U Delay
00000000 T High_Pressure_Detected
```

4-Symblos of main.o

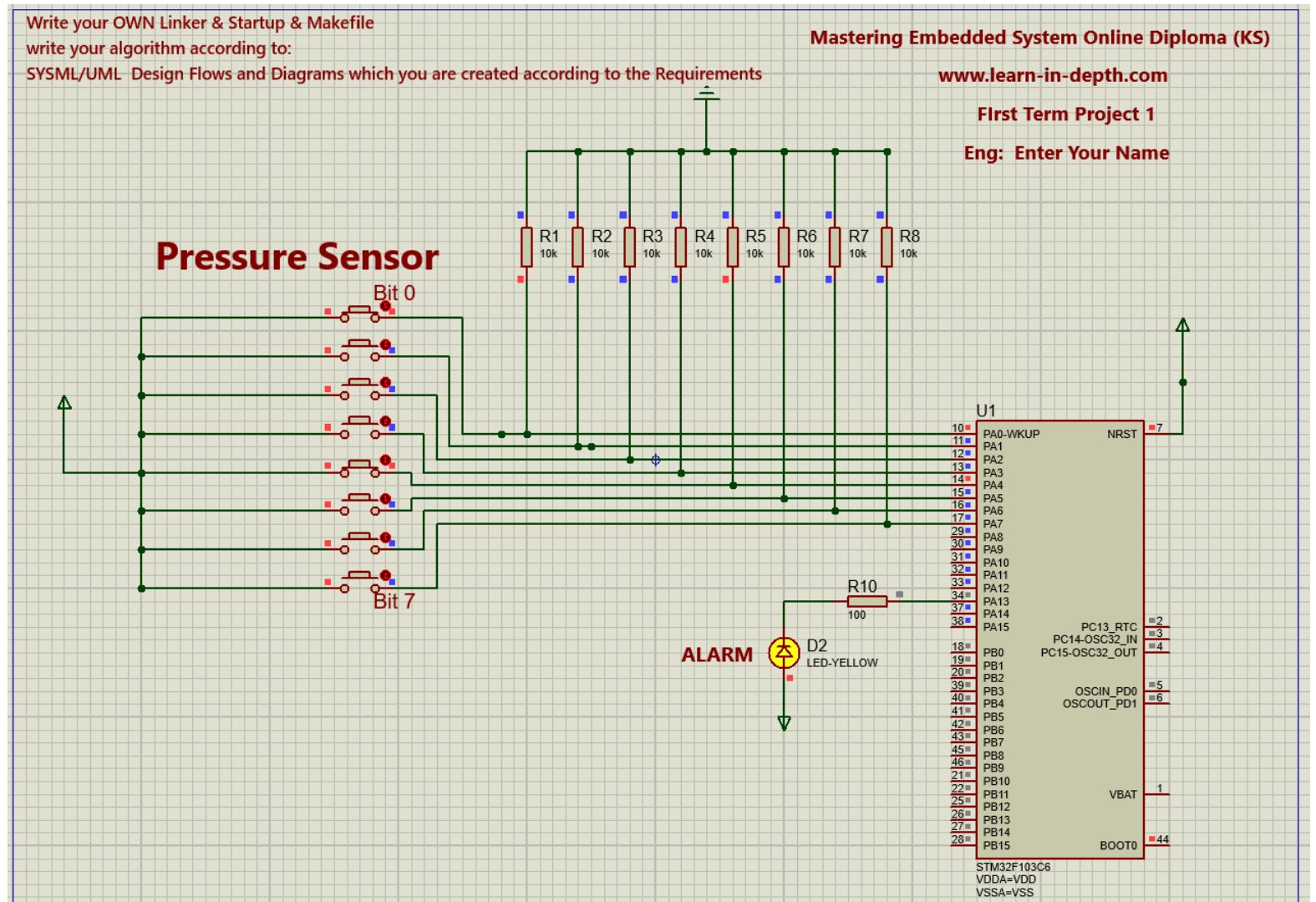
```
PS D:\study\GNU-Tools-ARM-Embedded\7 2017-q4-major\bin> .\arm-none-eabi-nm.exe main.o
          U Delay
          U getPressureVal
          U GPIO_INITIALIZATION
          U High_Pressure_Detected
00000000 T main
```

5-Symbols of Pressure Controller.elf

```
PS D:\study\GNU-Tools-ARM-Embedded\7 2017-q4-major\bin> .\arm-none-eabi-nm.exe .\Pressure-Controller-cortex-m3.elf
08000168 t _reset
08000074 T Delay
08000094 T getPressureVal
080000e8 T GPIO_INITIALIZATION
08000050 T High_Pressure_Detected
08000138 T main
080000ac T Set_Alarm_actuator
0800016e t Vector_handler
```

Simulation on proteus

Case1 : when pressure value is 17 bar (less than 20bar)



Case2 : when pressure value is 23 bar (bigger than 20bar)

Write your OWN Linker & Startup & Makefile

write your algorithm according to:

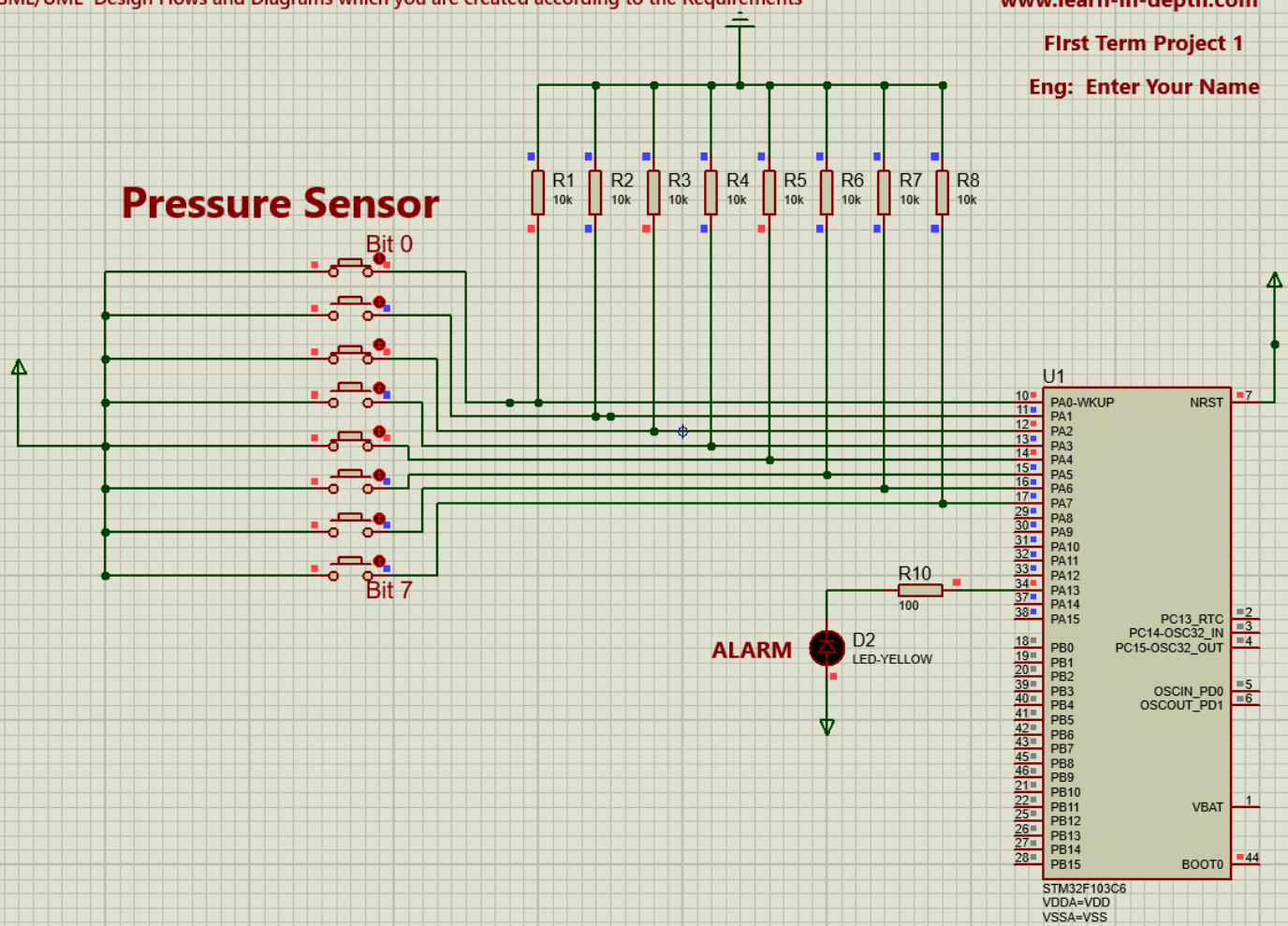
SYSML/UML Design Flows and Diagrams which you are created according to the Requirements

Mastering Embedded System Online Diploma (KS)

www.learn-in-depth.com

First Term Project 1

Eng: Enter Your Name



Write your OWN Linker & Startup & Makefile

write your algorithm according to:

SYSML/UML Design Flows and Diagrams which you are created according to the Requirements

Mastering Embedded System Online Diploma (KS)

www.learn-in-depth.com

First Term Project 1

Eng: Enter Your Name

