Pressure System

Customer requirement

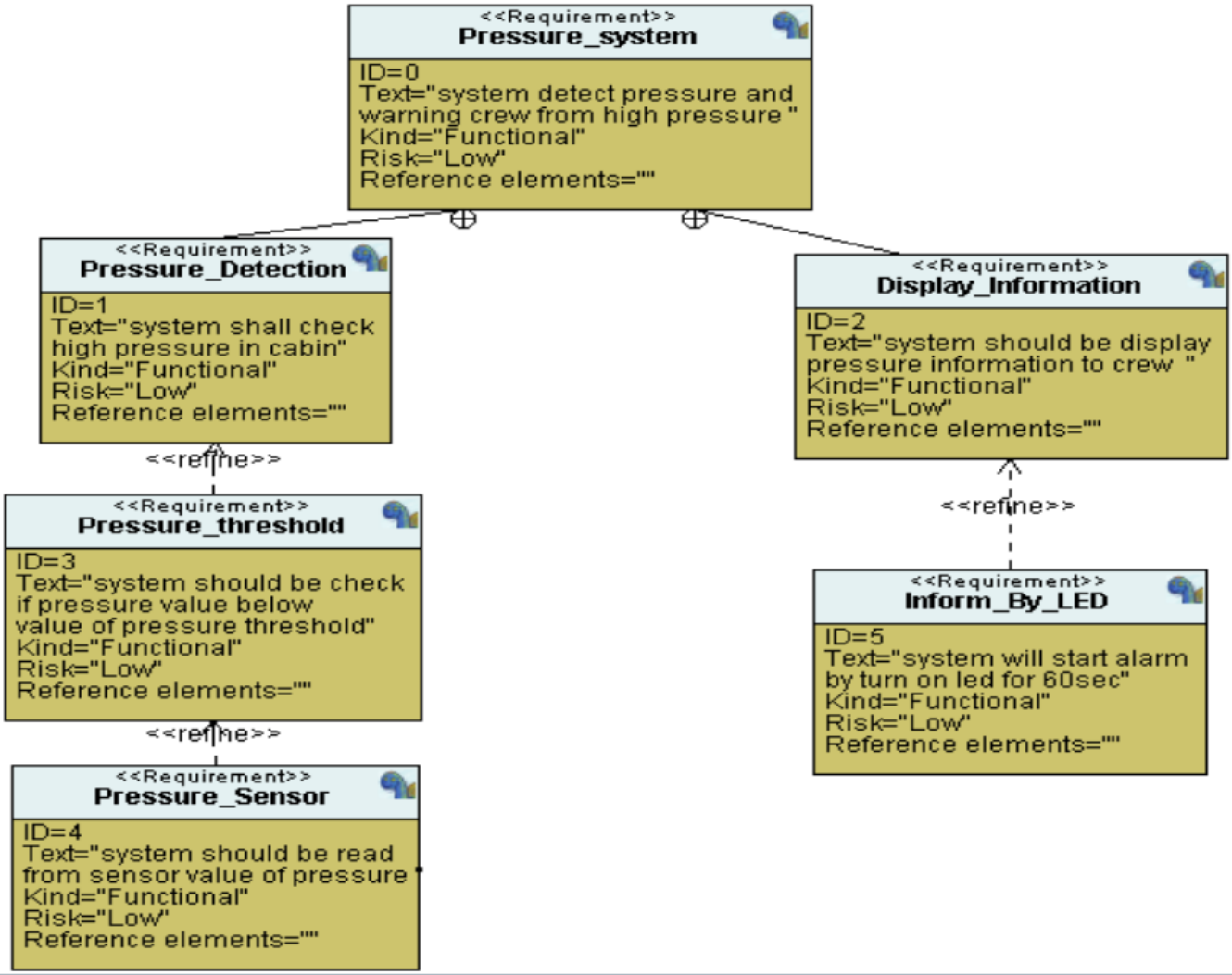
Specification:

1. System read pressure value from sensor in cabin.
2. System inform crew if pressure value above 20 bars by turn on led for 60 secs.

Assumptions:

1. The pressure sensor never fails.
2. The led alarm never fails.

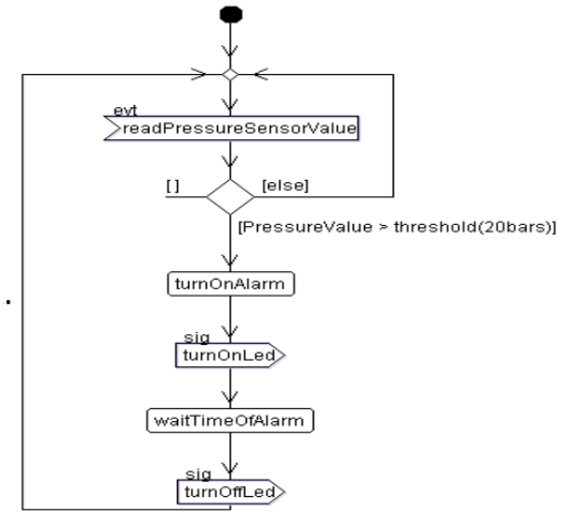
Customer requirement diagram:



Pressure system need two component, pressure detection and display information.

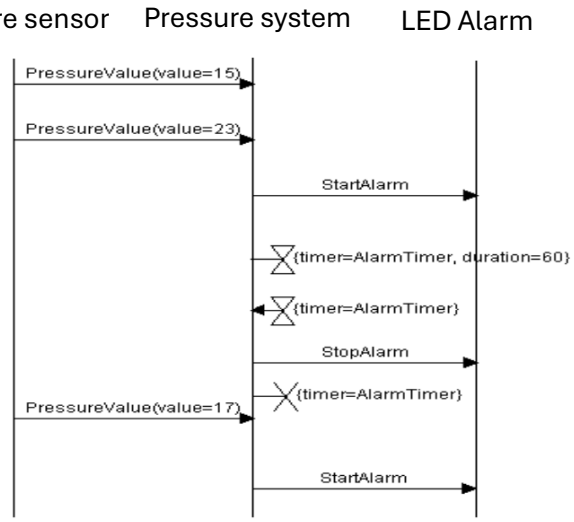
UML: Activity Diagram

1. start program
2. Read pressure value from sensor
3. If pressure value smaller than threshold return to waiting new value and check.
4. If pressure value larger than threshold will start alarm by turn on led for 60 sec then return to waiting new value and check.

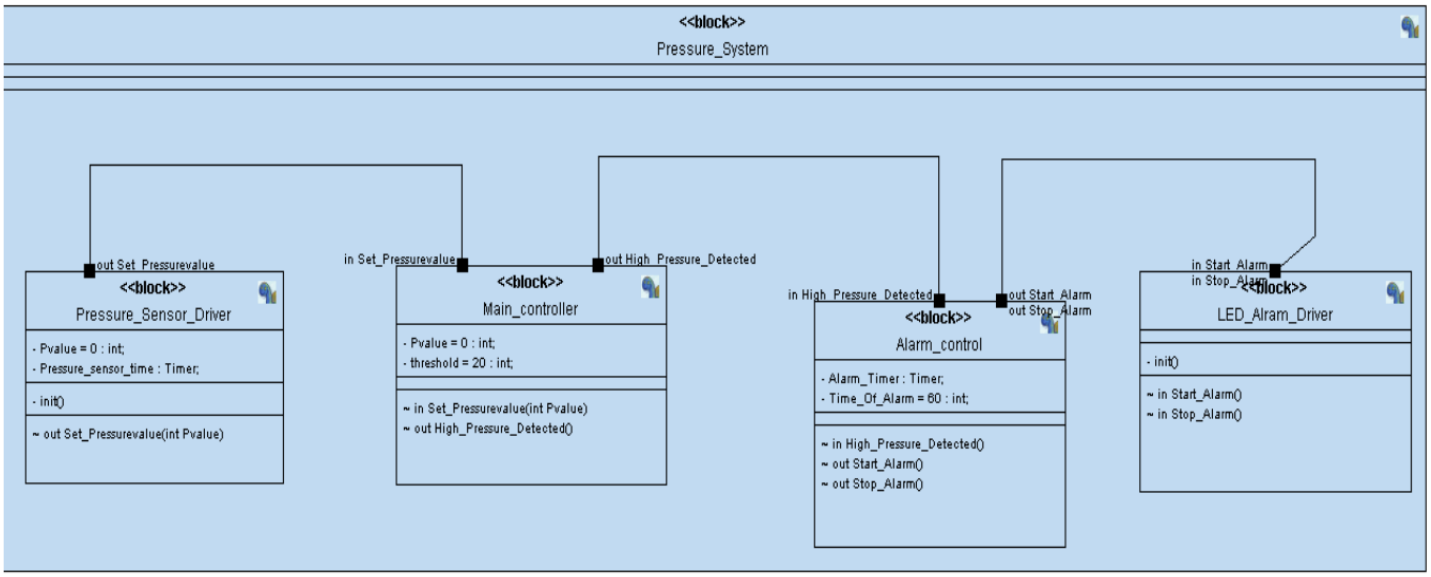


UML: Sequence Diagram

1. Assume pressure sensor send value to system smaller than threshold system won't happen anything.
2. Assume pressure sensor send value to system larger than threshold system will send to led alarm to start alarm for 60 sec then stop alarm.

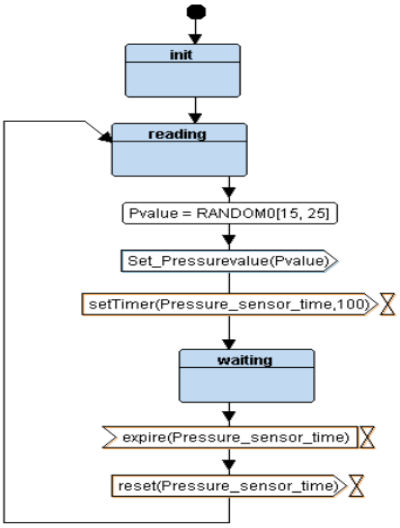


Design: Block Diagram



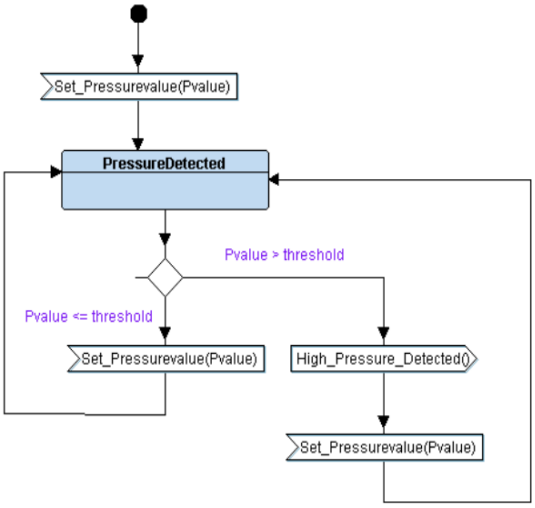
Block: Pressure Sensor Driver

1. Start will be initialized pressure sensor driver.
2. Go to reading state to get value of pressure.
3. Send pressure value to main controller.
4. wait for 100 sec to go to reading state again.



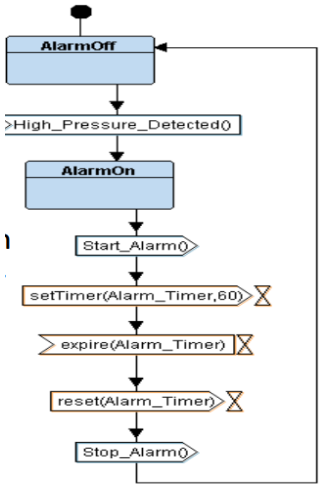
Block: Main Controller

1. Main controllers wait to receive pressure value from pressure sensor
2. If pressure value smaller than threshold will be waiting to a new pressure value.
3. If pressure value smaller than threshold will be send signal to alarm control to be start alarm then wait to new pressure value.



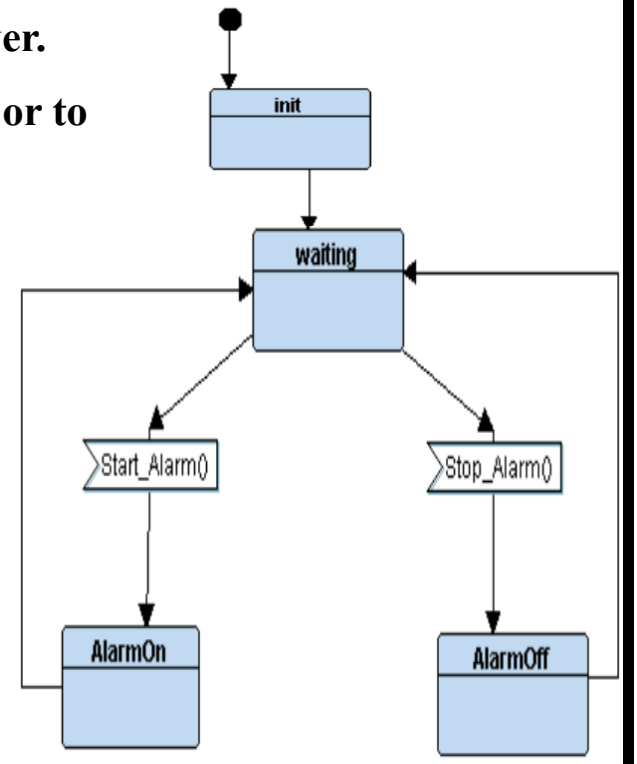
Block: Alarm controller

1. First alarm controller will be in state alarm off waiting to receive high pressure detected to go to state alarm on.
2. Alarm controller send to led_alarm signal to turn on then set timer for 60 sec then send signal to led_alarm to turn off then return to state alarm off.



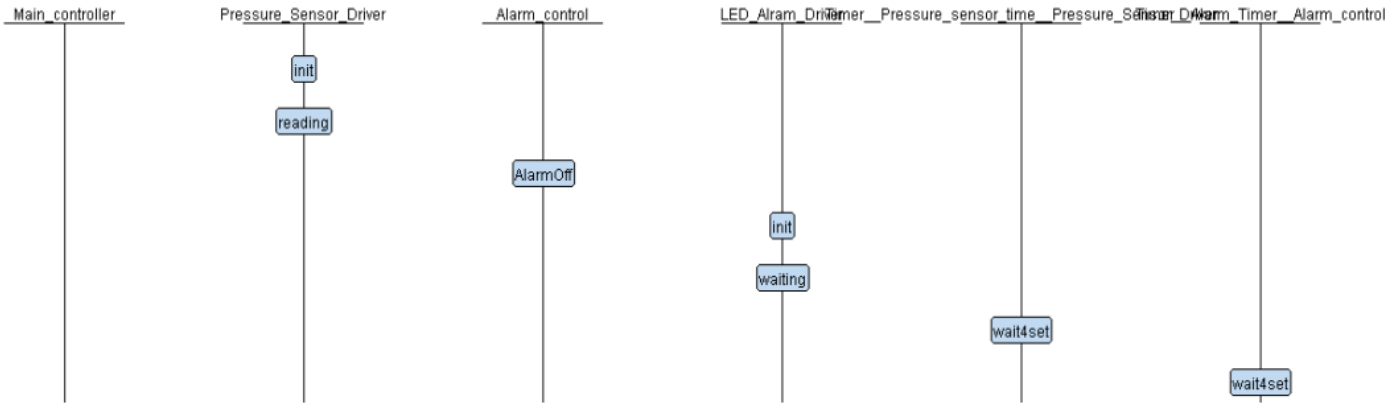
Block: LED Alarm Driver

1. Start will be initialized Led alarm driver.
2. Driver waiting to signal to start alarm or to stop alarm.
3. If signal start alarm will be go to state alarm on.
4. If signal stop alarm will be go to state alarm off.

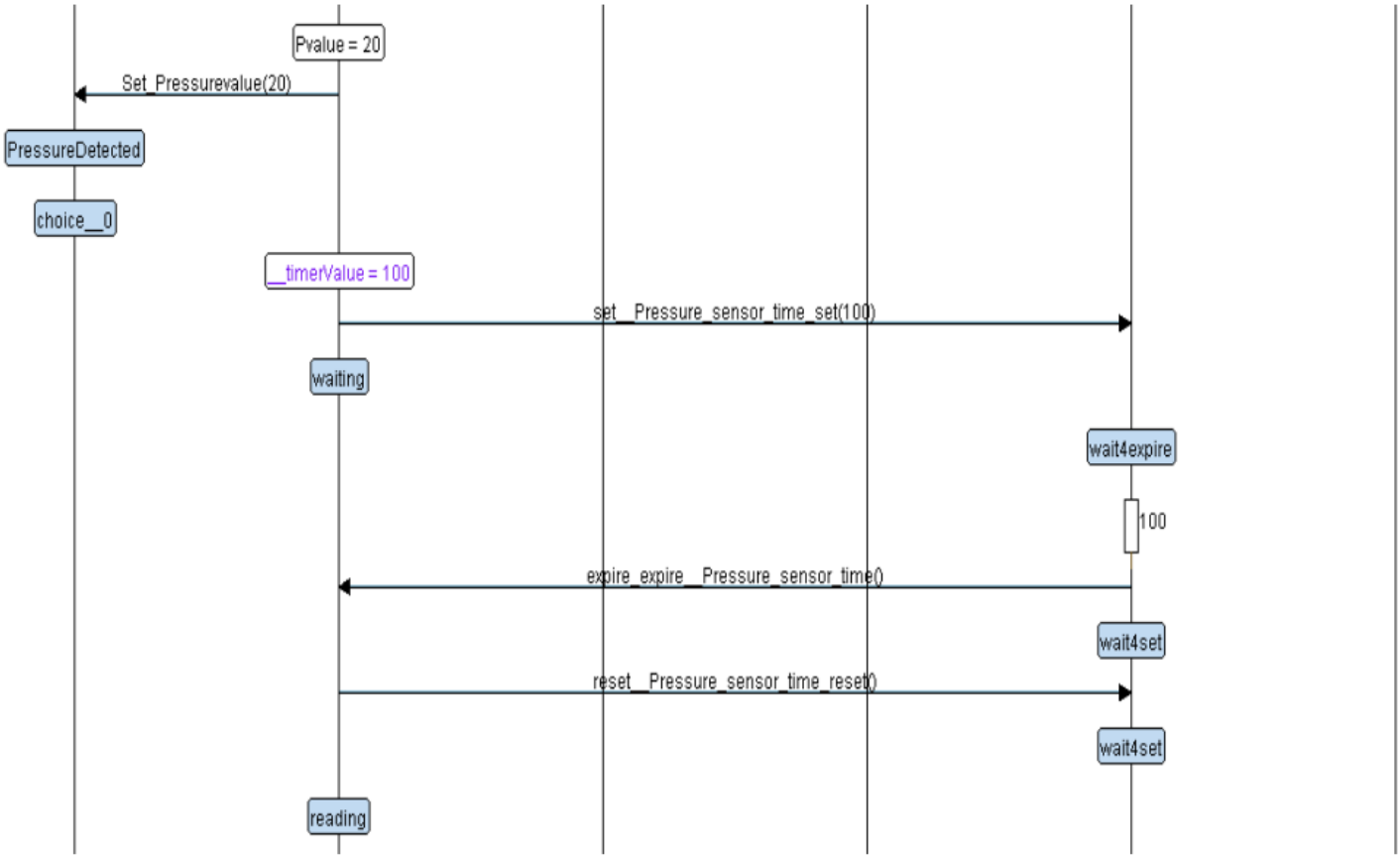


simulate project to show sequence

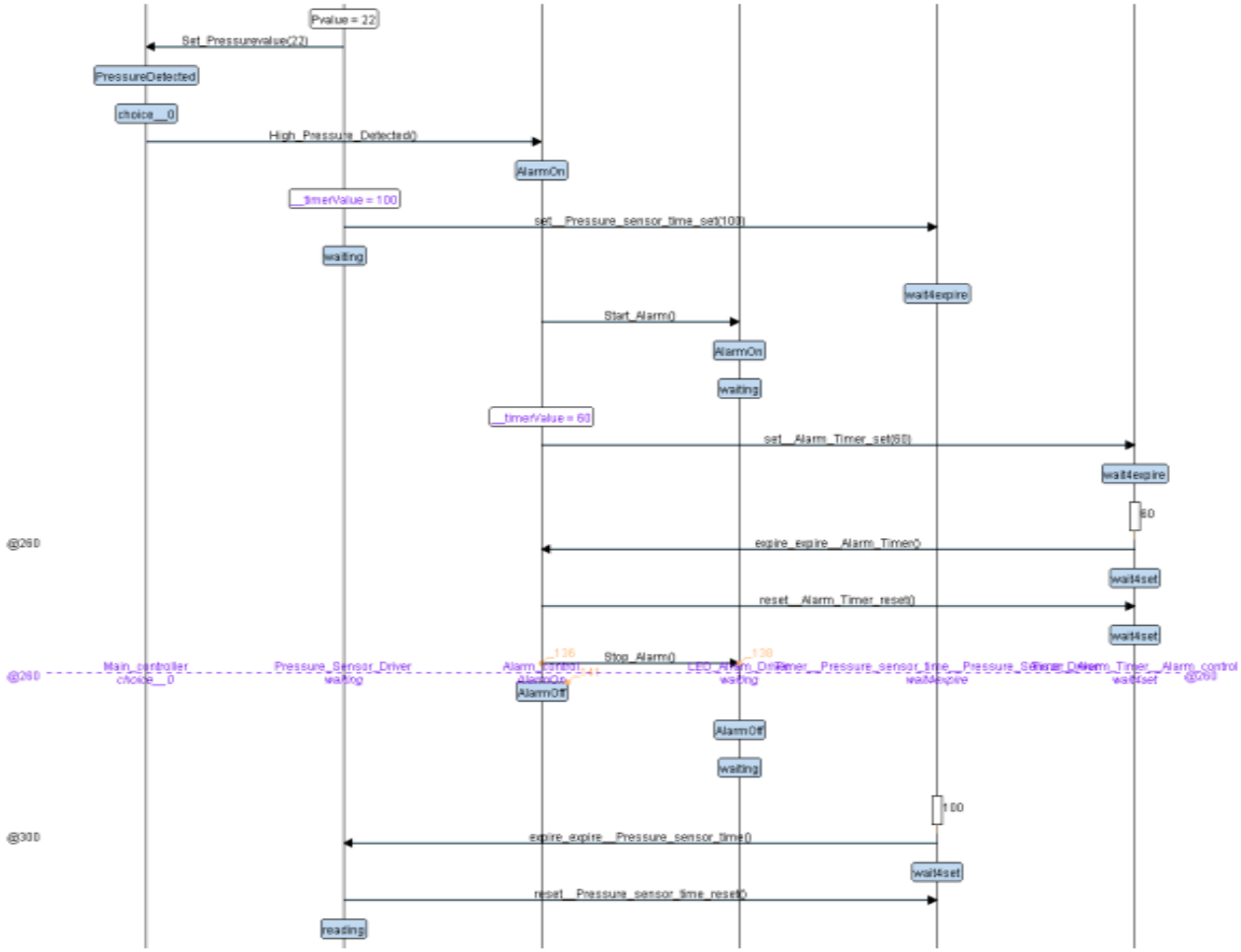
1. when start program.



2.when sensor read value smaller than threshold (20 bar).



3.when sensor read value bigger than threshold (20 bar).



After implement code will show symbols and sections for each file:

1. **Symbols for driver of pressure sensor and Led alarm actuator:**

```
20100@MRK MINGW64 /e/Emmbeded_diploma/Learn_In_Depth/Project1 (main)
$ arm-none-eabi-nm.exe driver.o
00000000 T Delay
00000020 T getPressureVal
00000074 T GPIO_INITIALIZATION
00000038 T Set_Alarm_actuator
```

.

2. **Symbols for Alarm control:**

```
20100@MRK MINGW64 /e/Emmbeded_diploma/Learn_In_Depth/Project1 (main)
$ arm-none-eabi-nm.exe control_of_alarm.o
         U Delay
00000000 T High_pressure_detected
         U Set_Alarm_actuator
```

3. **Symbols for main controller:**

```
20100@MRK MINGW64 /e/Emmbeded_diploma/Learn_In_Depth/Project1 (main)
$ arm-none-eabi-nm.exe control_of_alarm.o
         U Delay
00000000 T High_pressure_detected
         U Set_Alarm_actuator
```

4- Symbols for pressure detector

```
20100@MRK MINGW64 /e/Emmbeded_diploma/Learn_In_Depth/Project1 (main)
$ arm-none-eabi-nm.exe project1_pressure_detector.elf
20000000 B _E_bss
20000000 D _E_data
080001c0 T _E_text
20000000 B _S_bss
20000000 D _S_data
20001000 B _stack_top
080001b4 W Bus_Fault
080001b4 T Default_handler
08000040 T Delay
08000060 T getPressureVal
080000b4 T GPIO_INITIALIZATION
080001b4 W H_fault_Handler
0800001c T High_pressure_detected
08000104 T main
080001b4 W MM_Fault_Handler
080001b4 W NMI_Handler
08000130 T Reset_Handler
08000078 T Set_Alarm_actuator
080001b4 W Usage_Fault_Handler
08000000 T vectors
```

5. Sections for driver of pressure sensor and Led alarm actuator:

```
20100@MRK MINGW64 /e/Emmbeded_diploma/Learn_In_Depth/Project1 (main)
$ 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 .debug_info   00000a05  00000000  00000000  000000f8  2**0
                  CONTENTS, RELOC, READONLY, DEBUGGING
  4 .debug_abbrev 000001de  00000000  00000000  00000afd  2**0
                  CONTENTS, READONLY, DEBUGGING
  5 .debug_loc    00000140  00000000  00000000  00000cdb  2**0
                  CONTENTS, READONLY, DEBUGGING
  6 .debug_aranges 00000020  00000000  00000000  00000e1b  2**0
                  CONTENTS, RELOC, READONLY, DEBUGGING
  7 .debug_line   000001b9  00000000  00000000  00000e3b  2**0
                  CONTENTS, RELOC, READONLY, DEBUGGING
  8 .debug_str    00000568  00000000  00000000  00000ff4  2**0
                  CONTENTS, READONLY, DEBUGGING
  9 .comment      0000007f  00000000  00000000  0000155c  2**0
                  CONTENTS, READONLY
 10 .debug_frame  000000a0  00000000  00000000  000015dc  2**2
                  CONTENTS, RELOC, READONLY, DEBUGGING
 11 .ARM.attributes 00000033  00000000  00000000  0000167c  2**0
                  CONTENTS, READONLY
```

6. Sections for Alarm control:

```
20100@MRK MINGW64 /e/Emmbeded_diploma/Learn_In_Depth/Project1 (main)
$ 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 .debug_info   00000a05  00000000  00000000  000000f8  2**0
                  CONTENTS, RELOC, READONLY, DEBUGGING
  4 .debug_abbrev 000001de  00000000  00000000  00000afd  2**0
                  CONTENTS, READONLY, DEBUGGING
  5 .debug_loc    00000140  00000000  00000000  00000cdb  2**0
                  CONTENTS, READONLY, DEBUGGING
  6 .debug_aranges 00000020  00000000  00000000  00000e1b  2**0
                  CONTENTS, RELOC, READONLY, DEBUGGING
  7 .debug_line   000001b9  00000000  00000000  00000e3b  2**0
                  CONTENTS, RELOC, READONLY, DEBUGGING
  8 .debug_str    00000568  00000000  00000000  00000ff4  2**0
                  CONTENTS, READONLY, DEBUGGING
  9 .comment      0000007f  00000000  00000000  0000155c  2**0
                  CONTENTS, READONLY
 10 .debug_frame  000000a0  00000000  00000000  000015dc  2**2
                  CONTENTS, RELOC, READONLY, DEBUGGING
 11 .ARM.attributes 00000033  00000000  00000000  0000167c  2**0
                  CONTENTS, READONLY
```

7. Sections for main controller:

```
20100@MRK MINGW64 /e/Emmbeded_diploma/Learn_In_Depth/Project1 (main)
$ 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         0000002c  00000000  00000000  00000034  2**1
                  CONTENTS, ALLOC, LOAD, RELOC, READONLY, CODE
  1 .data         00000000  00000000  00000000  00000060  2**0
                  CONTENTS, ALLOC, LOAD, DATA
  2 .bss          00000000  00000000  00000000  00000060  2**0
                  ALLOC
  3 .debug_info   000009c1  00000000  00000000  00000060  2**0
                  CONTENTS, RELOC, READONLY, DEBUGGING
  4 .debug_abbrev 00000187  00000000  00000000  00000a21  2**0
                  CONTENTS, READONLY, DEBUGGING
  5 .debug_loc    00000038  00000000  00000000  00000ba8  2**0
                  CONTENTS, READONLY, DEBUGGING
  6 .debug_aranges 00000020  00000000  00000000  00000be0  2**0
                  CONTENTS, RELOC, READONLY, DEBUGGING
  7 .debug_line   0000019f  00000000  00000000  00000c00  2**0
                  CONTENTS, RELOC, READONLY, DEBUGGING
  8 .debug_str    00000546  00000000  00000000  00000d9f  2**0
                  CONTENTS, READONLY, DEBUGGING
  9 .comment      0000007f  00000000  00000000  000012e5  2**0
                  CONTENTS, READONLY
 10 .debug_frame  00000030  00000000  00000000  00001364  2**2
                  CONTENTS, RELOC, READONLY, DEBUGGING
 11 .ARM.attributes 00000033  00000000  00000000  00001394  2**0
                  CONTENTS, READONLY
```

8. Symbols for Pressure system:

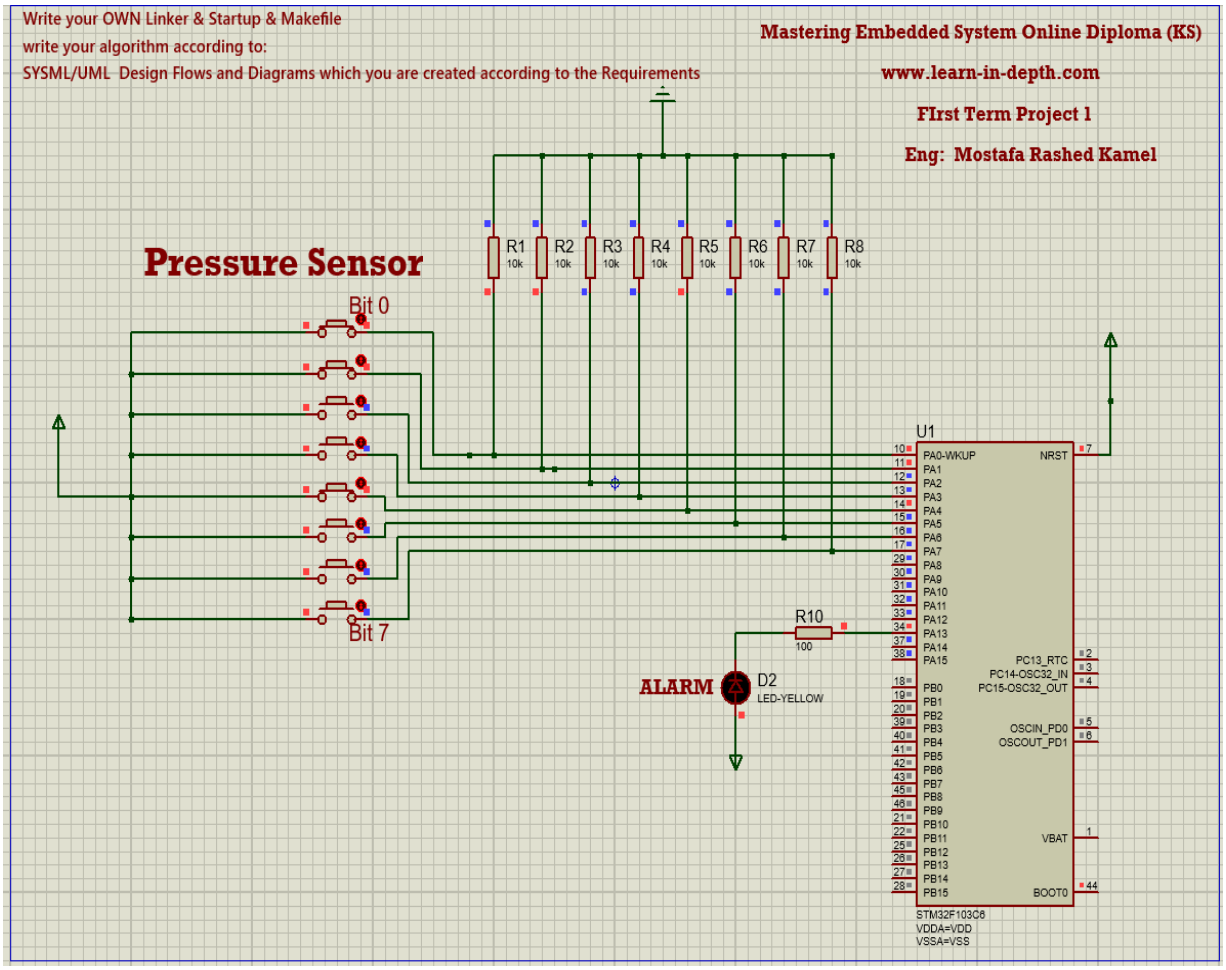
```
20100@MRK MINGW64 /e/Emmbeded_diploma/Learn_In_Depth/Project1 (main)
$ arm-none-eabi-objdump.exe -h project1_pressure_detector.elf

project1_pressure_detector.elf:     file format elf32-littlearm

Sections:
Idx Name          Size      VMA       LMA       File off  Algn
  0 .text         000001c0  08000000  08000000  00010000  2**2
                  CONTENTS, ALLOC, LOAD, READONLY, CODE
  1 .data         00000000  20000000  080001c0  00020000  2**0
                  CONTENTS, ALLOC, LOAD, DATA
  2 .bss          00001000  20000000  080001c0  00020000  2**0
                  ALLOC
  3 .debug_info   00001ebb  00000000  00000000  00020000  2**0
                  CONTENTS, READONLY, DEBUGGING
  4 .debug_abbrev 0000059b  00000000  00000000  00021ebb  2**0
                  CONTENTS, READONLY, DEBUGGING
  5 .debug_loc    00000220  00000000  00000000  00022456  2**0
                  CONTENTS, READONLY, DEBUGGING
  6 .debug_aranges 00000080  00000000  00000000  00022676  2**0
                  CONTENTS, READONLY, DEBUGGING
  7 .debug_line   000005c0  00000000  00000000  000226f6  2**0
                  CONTENTS, READONLY, DEBUGGING
  8 .debug_str    000005cf  00000000  00000000  00022cb6  2**0
                  CONTENTS, READONLY, DEBUGGING
  9 .comment      0000007e  00000000  00000000  00023285  2**0
                  CONTENTS, READONLY
 10 .ARM.attributes 00000033  00000000  00000000  00023303  2**0
                  CONTENTS, READONLY
 11 .debug_frame  0000014c  00000000  00000000  00023338  2**2
                  CONTENTS, READONLY, DEBUGGING
```

When run executable on simulation

1. When pressure value (19) smaller than threshold (20):



2. When pressure value (23) larger than threshold (20):