

Pressure Detector system

Customer requirement:

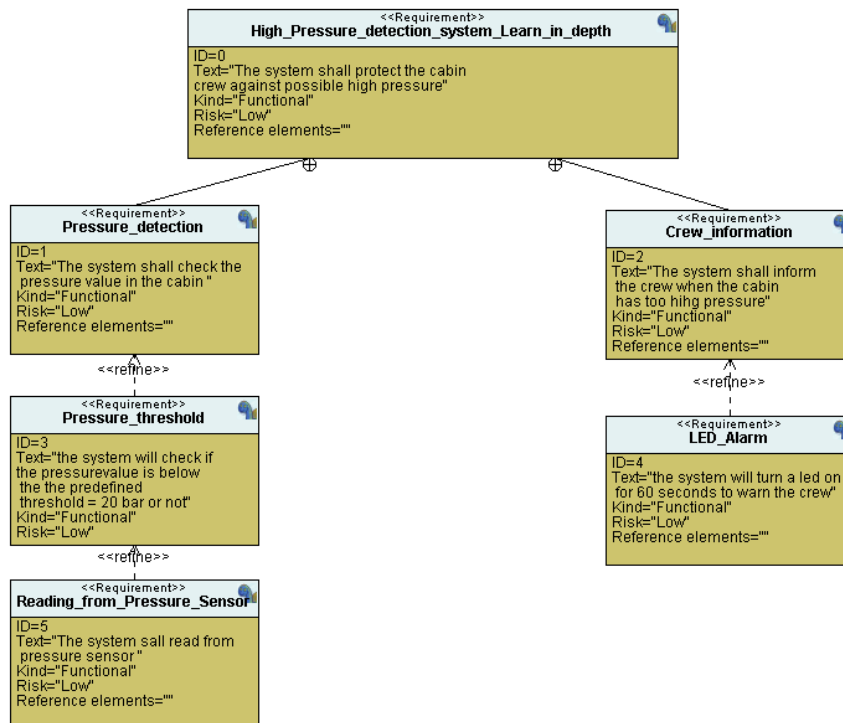
Specifications:

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

Assumptions:

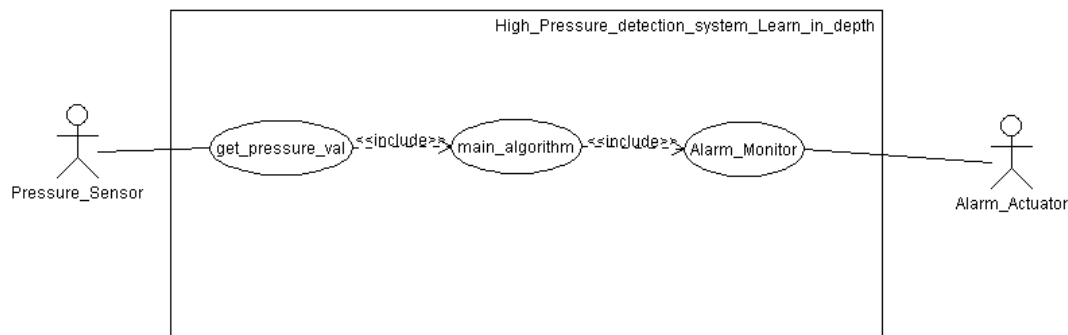
1. The pressure sensor never fails.
2. The led alarm never fails.

Customer requirement diagram:



Pressure system need two component, pressure detector and display information device.

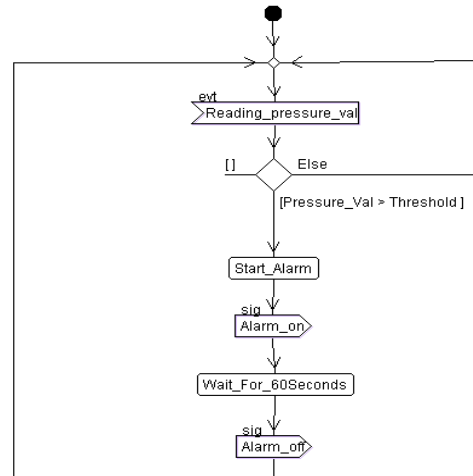
Use Case Diagram:



1. get_pressure_Val(Use case) get value of pressure from pressure_sensor(Actor).
2. Main_algorithm(Use case) compare the pressure value with the Threshold.
3. Alarm_Monitor(Use case) Turn on Alarm_Actuator(Actor) when the value of the cabin pressure larger than 20 bars.

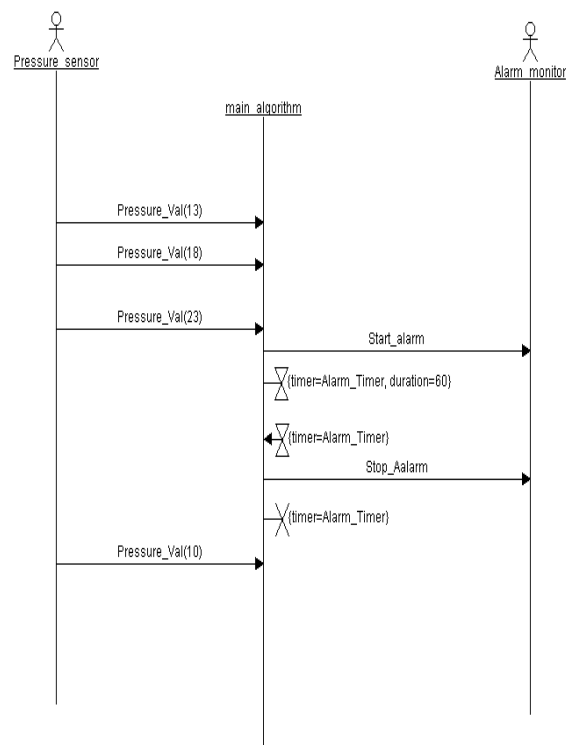
Activity Diagram:

1. Start program.
2. Read pressure value from sensor.
3. If pressure smaller than the threshold return to waiting for new value.
4. If pressure value larger than threshold will start alarm by turning a led on for 60 second then return to waiting for new value.

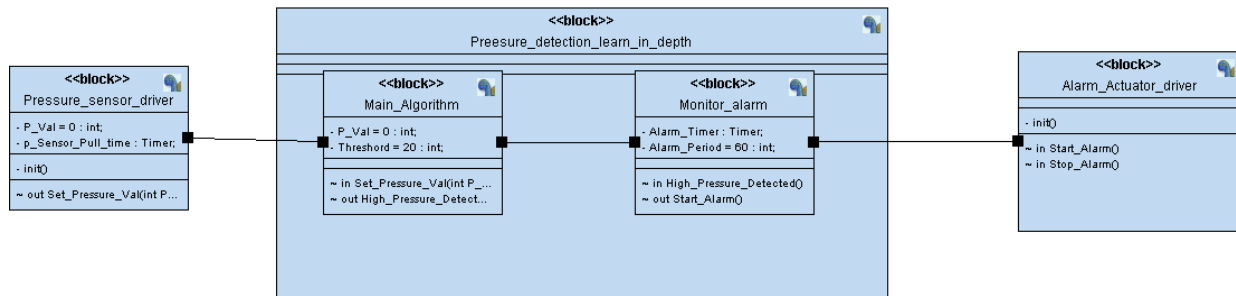


Sequence Diagram:

1. Assume pressure sensor send value to main algorithm smaller than threshold main algorithm won't take any action.
2. Assume pressure sensor send value to main algorithm larger than threshold main algorithm will send start alarm for 60 sec then stop the alarm.

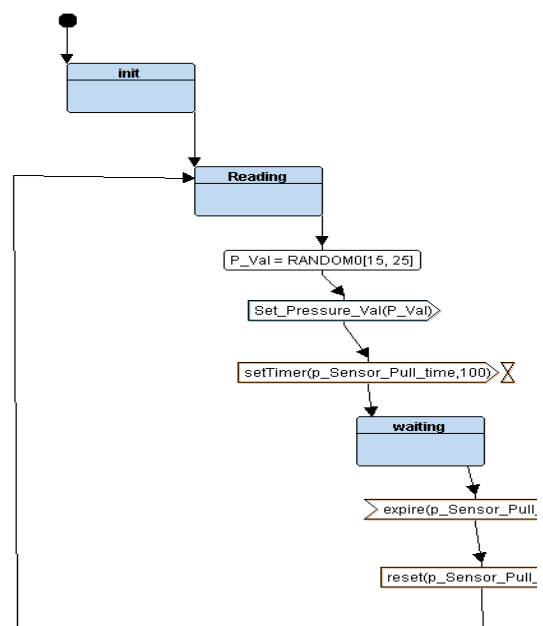


Design: Block Diagram:



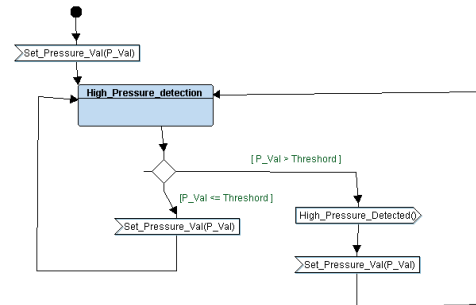
Block: Pressure Sensor Driver:

1. Start will initialize 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 reading state again.



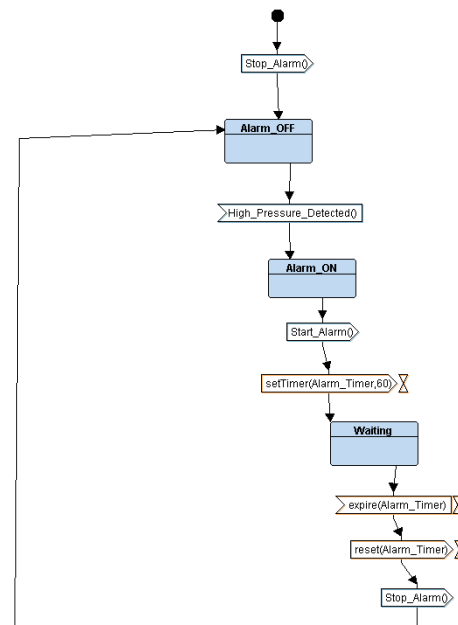
Block: Main Algorithm:

1. Main controller waits 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 signal to alarm control to be start alarm then wait to new pressure value.



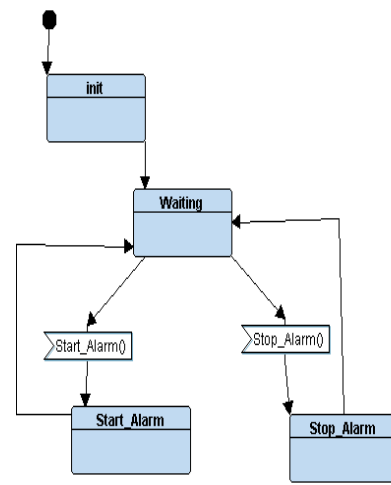
Block: Monitor Alarm:

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 Alarm_ON signal to turn Led on then set timer for 60 sec then send Stop_Alarm signal to Alarm Actuator.

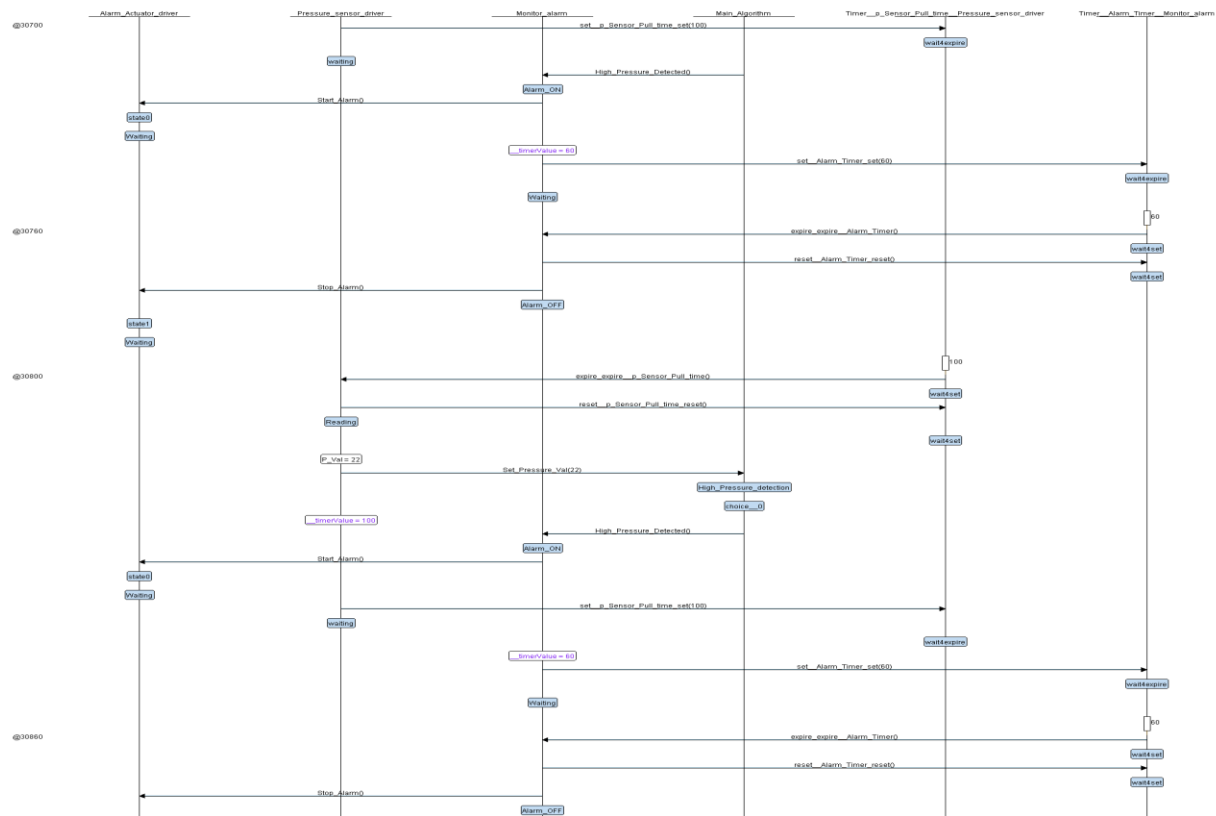


Block: Alarm Actuator Driver:

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

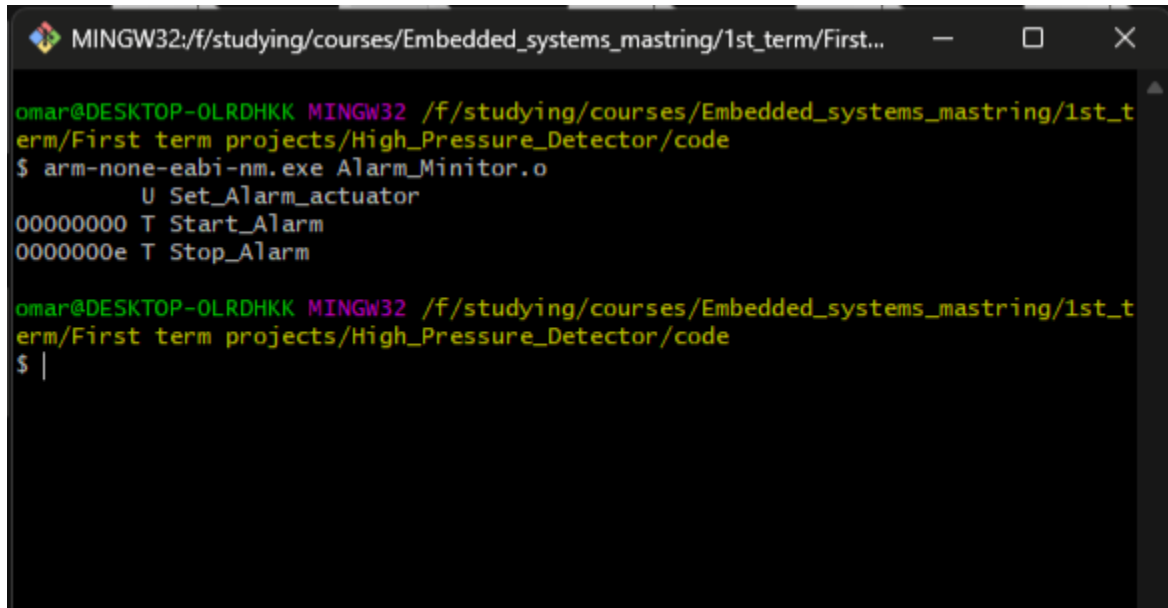


Analysis and Timing Diagram:



Symbols for each block:

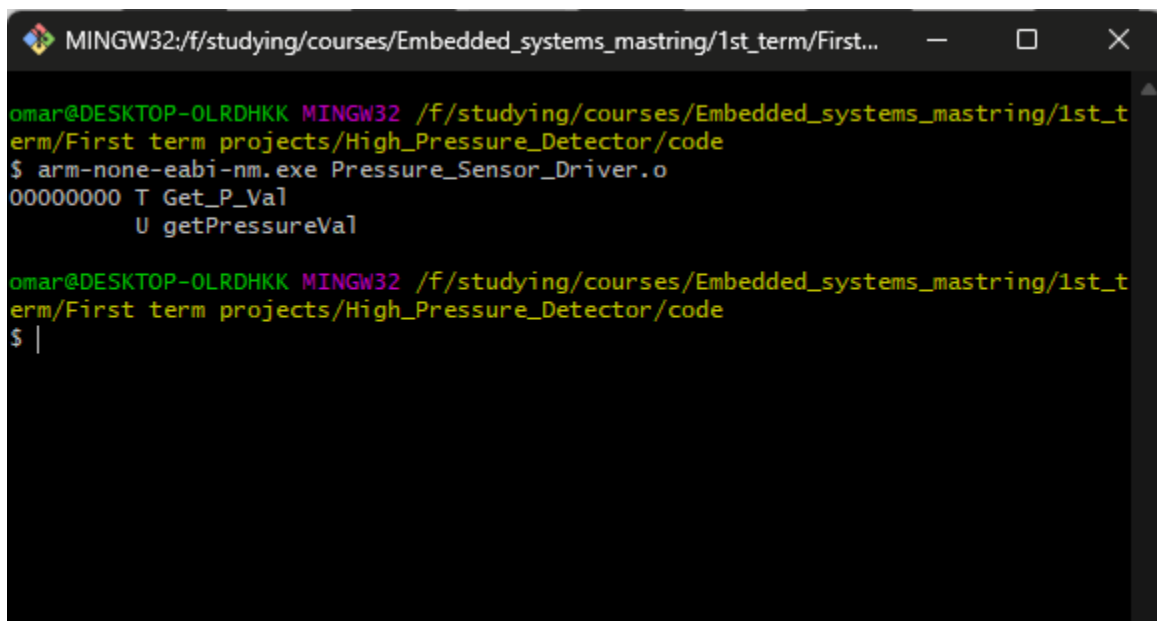
- Symbols for Alarm_Minitor.o



```
MINGW32:/f/studying/courses/Embedded_systems_mastring/1st_term/First...
omar@DESKTOP-OLRDHKK MINGW32 /f/studying/courses/Embedded_systems_mastring/1st_t
erm/First term projects/High_Pressure_Detector/code
$ arm-none-eabi-nm.exe Alarm_Minitor.o
                 U Set_Alarm_actuator
00000000 T Start_Alarm
0000000e T Stop_Alarm

omar@DESKTOP-OLRDHKK MINGW32 /f/studying/courses/Embedded_systems_mastring/1st_t
erm/First term projects/High_Pressure_Detector/code
$ |
```

- Symbols for Pressure_Sensor.o



```
MINGW32:/f/studying/courses/Embedded_systems_mastring/1st_term/First...
omar@DESKTOP-OLRDHKK MINGW32 /f/studying/courses/Embedded_systems_mastring/1st_t
erm/First term projects/High_Pressure_Detector/code
$ arm-none-eabi-nm.exe Pressure_Sensor_Driver.o
00000000 T Get_P_Val
                 U getPressureVal

omar@DESKTOP-OLRDHKK MINGW32 /f/studying/courses/Embedded_systems_mastring/1st_t
erm/First term projects/High_Pressure_Detector/code
$ |
```

- Symbols for main.o

```
MINGW32:/f/studying/courses/Embedded_systems_mastring/1st_term/First term projects/High_Pressure_Dete...  
omar@DESKTOP-OLRDHKK MINGW32 /f/studying/courses/Embedded_systems_mastring/1st_term/First term projects/Hi  
gh_Pressure_Detector/code  
$ arm-none-eabi-nm.exe main.o  
                 U App_init  
                 U Delay  
                 U High_Pressure_detection_system_APP  
00000000 T main  
  
omar@DESKTOP-OLRDHKK MINGW32 /f/studying/courses/Embedded_systems_mastring/1st_term/First term projects/Hi  
gh_Pressure_Detector/code  
$ |
```

- Symbols for High_Pressure_Detection.elf

```
MINGW32:/f/studying/courses/Embedded_systems_mastring/1st_term/First...  
omar@DESKTOP-OLRDHKK MINGW32 /f/studying/courses/Embedded_systems_mastring/1st_t  
erm/First term projects/High_Pressure_Detector/code  
$ arm-none-eabi-nm.exe High_Pressure_Detection_System.elf  
080001a6 t _reset  
080001b4 D Alarm_Period  
08000144 T App_init  
08000068 T Delay  
08000190 T Get_P_Val  
08000088 T getPressureVal  
080000dc T GPIO_INITIALIZATION  
08000150 T High_Pressure_detection_system_APP  
0800012c T main  
080000a0 T Set_Alarm_actuator  
0800004c T Start_Alarm  
0800005a T Stop_Alarm  
080001b0 D threshold  
080001ac t Vector_handler  
  
omar@DESKTOP-OLRDHKK MINGW32 /f/studying/courses/Embedded_systems_mastring/1st_t  
erm/First term projects/High_Pressure_Detector/code  
$
```


Sections for each Block:

- Section for Alarm_minitor.o

```
MINGW32:/f/studying/courses/Embedded_systems_mastring/1st_term/First term projects/High_Pressure_Dete...
omar@DESKTOP-OLRDHKK MINGW32 /f/studying/courses/Embedded_systems_mastring/1st_term/First term projects/Hi
gh_Pressure_Detector/code
$ arm-none-eabi-objdump.exe -h Alarm_Minitor.o

Alarm_Minitor.o:      file format elf32-littlearm

Sections:
Idx Name              Size      VMA       LMA       File off  Algn
 0 .text              0000001c  00000000  00000000  00000034  2**1
                   CONTENTS, ALLOC, LOAD, RELOC, READONLY, CODE
 1 .data              00000000  00000000  00000000  00000050  2**0
                   CONTENTS, ALLOC, LOAD, DATA
 2 .bss              00000000  00000000  00000000  00000050  2**0
                   ALLOC
```

- Section for Pressure_Sensor_Driver.o

```
MINGW32:/f/studying/courses/Embedded_systems_mastring/1st_term/First term projects/High_Pressure_Dete...
omar@DESKTOP-OLRDHKK MINGW32 /f/studying/courses/Embedded_systems_mastring/1st_term/First term projects/Hi
gh_Pressure_Detector/code
$ arm-none-eabi-objdump.exe -h Pressure_Sensor_Driver.o

Pressure_Sensor_Driver.o:      file format elf32-littlearm

Sections:
Idx Name              Size      VMA       LMA       File off  Algn
 0 .text              00000016  00000000  00000000  00000034  2**1
                   CONTENTS, ALLOC, LOAD, RELOC, READONLY, CODE
 1 .data              00000000  00000000  00000000  0000004a  2**0
                   CONTENTS, ALLOC, LOAD, DATA
 2 .bss              00000000  00000000  00000000  0000004a  2**0
                   ALLOC
```

- Section for main.o

```
MINGW32:/f/studying/courses/Embedded_systems_mastring/1st_term/First term projects/High_Pressure_Dete...
omar@DESKTOP-OLRDHKK MINGW32 /f/studying/courses/Embedded_systems_mastring/1st_term/First term projects/Hi
gh_Pressure_Detector/code
$ 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              00000018  00000000  00000000  00000034  2**2
                   CONTENTS, ALLOC, LOAD, RELOC, READONLY, CODE
 1 .data              00000000  00000000  00000000  0000004c  2**0
                   CONTENTS, ALLOC, LOAD, DATA
 2 .bss              00000000  00000000  00000000  0000004c  2**0
                   ALLOC
```

- Section for High_Pressure_Detection_System.elf

```
MINGW32:/f/studying/courses/Embedded_systems_mastring/1st_term/First term projects/High_Pressure_Dete...
omar@DESKTOP-OLRDHKK MINGW32 /f/studying/courses/Embedded_systems_mastring/1st_term/First term projects/Hi
gh_Pressure_Detector/code
$ arm-none-eabi-objdump.exe -h High_Pressure_Detection_System.elf

High_Pressure_Detection_System.elf:      file format elf32-littlearm

Sections:
Idx Name          Size      VMA       LMA       File off  Algn
  0 .text          000001ae  08000000  08000000  00010000  2**2
CONTENTS, ALLOC, LOAD, READONLY, CODE
  1 .data          00000008  080001b0  080001b0  000101b0  2**2
CONTENTS, ALLOC, LOAD, DATA
```

- Map file

```
makefile  makefile  m_file.map  m_file.map
File Edit View
Memory Configuration
Name      Origin      Length      Attributes
flash     0x08000000  0x00020000  xr
sram      0x20000000  0x00050000  xrw
*default* 0x08000000  0xffffffff

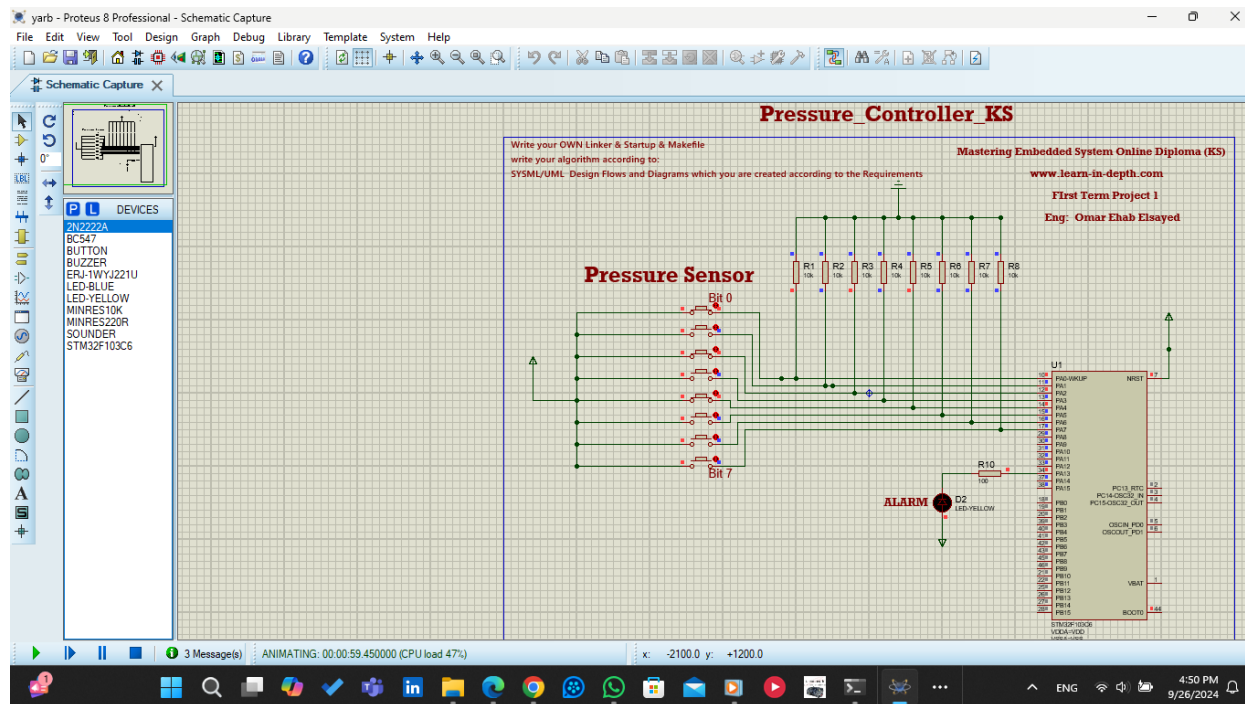
Linker script and memory map

.text      0x08000000  0x1ae
*(.vectors*)
.vectors   0x08000000  0x4c startup.o
*(.text*)
.text      0x0800004c  0x1c Alarm_Minitor.o
           0x0800004c  Start_Alarm
           0x0800005a  Stop_Alarm
.text      0x08000068  0xc4 driver.o
           0x08000068  Delay
           0x08000088  getPressureVal
           0x080000a0  Set_Alarm_actuator
           0x080000dc  GPIO_INITIALIZATION
.text      0x0800012c  0x16 main.o
           0x0800012c  main
*fill*     0x08000142  0x2
.text      0x08000144  0x4c Main_Algorithm.o
           0x08000144  App_Init
           0x08000150  High_Pressure_detection_system_APP
.text      0x08000190  0x16 Pressure_Sensor_Driver.o
           0x08000190  Get_P_Val
.text      0x080001a6  0x8 startup.o
.data      0x080001b0  0x8

Ln 1, Col 1  6,590 characters  100% Windows (CRLF) UTF-8
6:33 10/23
```

- Proteus simulation

- If the pressure Value is less than 20 bar, the Led is off.



- If the Pressure is bigger than 20 bar, the is on for 60 Second.

