Mastering Embedded System Online Diploma

First Term (Final project 1)

Pressure Control

ENG. Omar Hesham Labib

Website: https://www.learn-in-depth.com/online-diploma/omarhisham32%40gmail.com

Contents

Preface

Our Mission is to detect High Pressure in Cabin
There's a sensor to measure the pressure in the cabin and if the sensor detected a high pressure than normal it sends a signal to an existing Alarm then the Alarm makes a led to turn on for a while (not a constant Period).
In this Project many tools were used

1. Case Study

Our Client Needs to detect if there's a high pressure-(20 bar) in the cabin to protect the crew by alarming them by turning on a led for 60 seconds
the sensor also sends the measured pressure signal every 30 seconds to store it in an external memory this is an optional feature.

- Assumptions

1- Real time pressure detecting

2- Real time alarming

3- Sensors must not fail

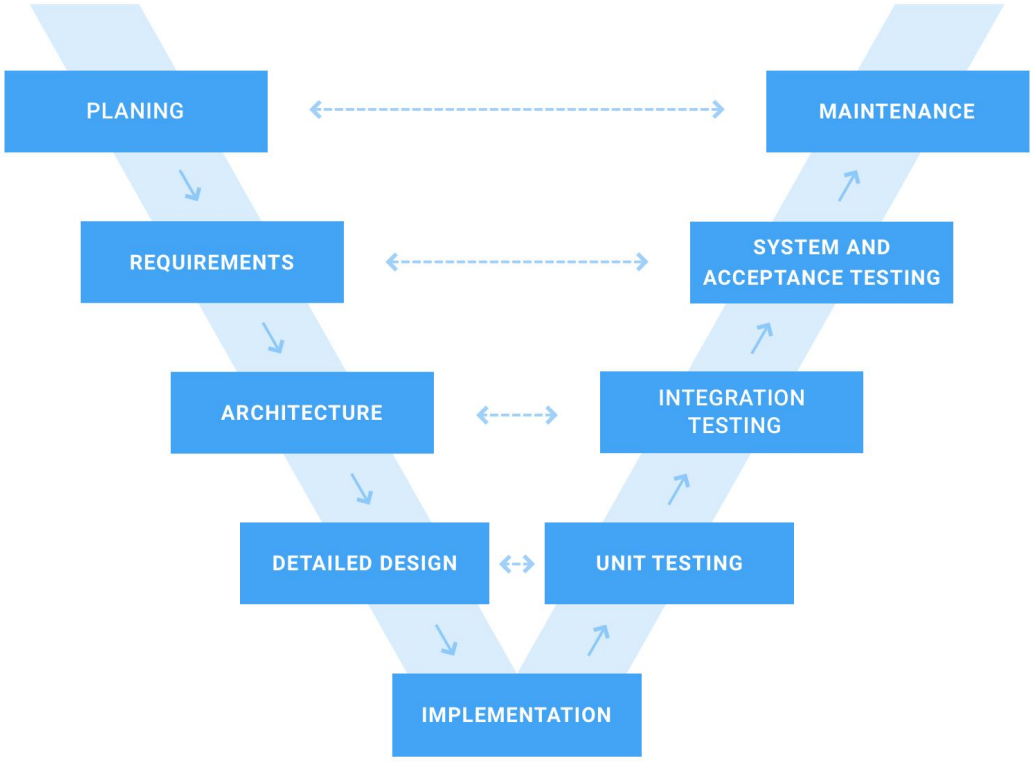
4- External power to avoid failing

2. Method

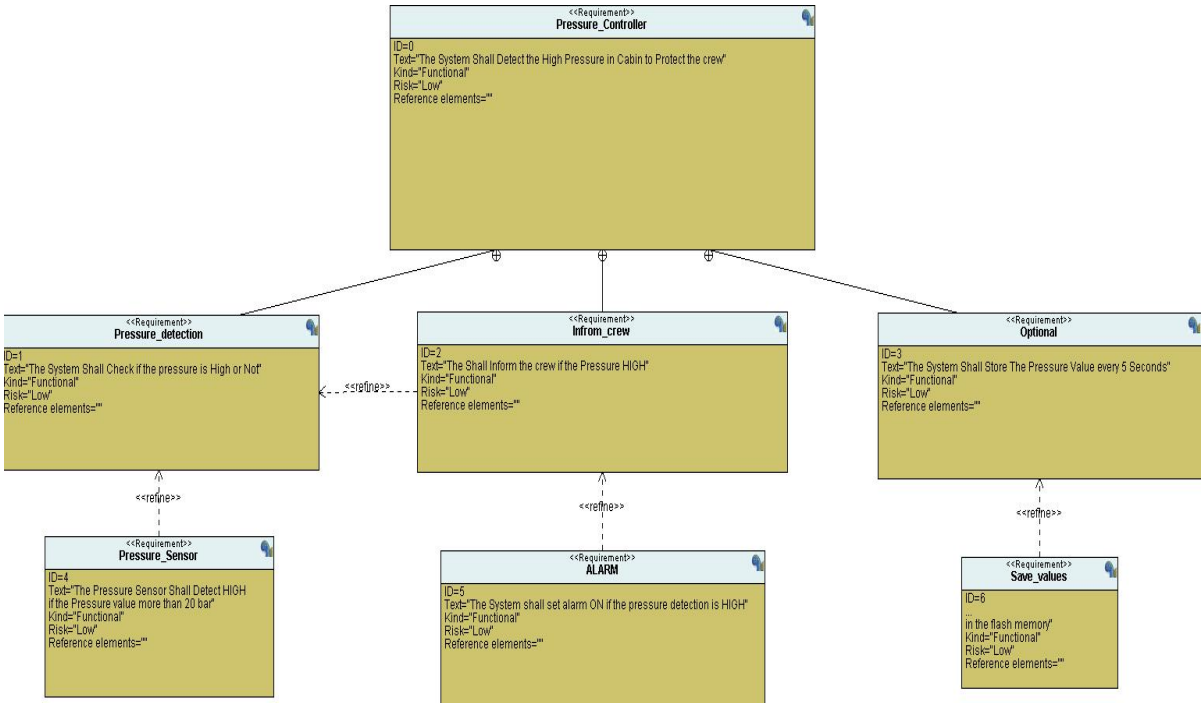
V-model is used in this Project

The V-model is a type of SDLC model in which processes are executed sequentially in a V-shape. It is also referred to as the Verification and Validation model. It is based on the assessment of a testing phase to each development stage. Each step's development is directly related to the testing phase. The next phase begins only after the previous phase has been completed.

V-Model

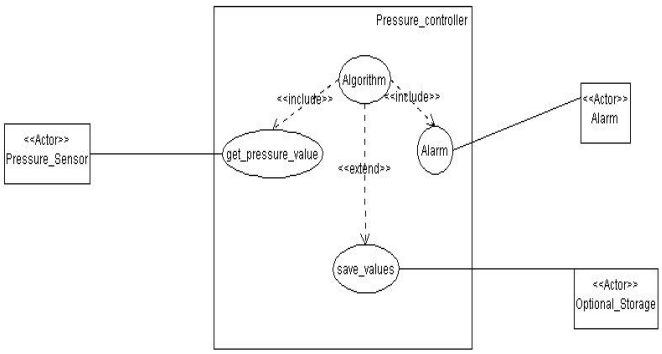


3. Requirements

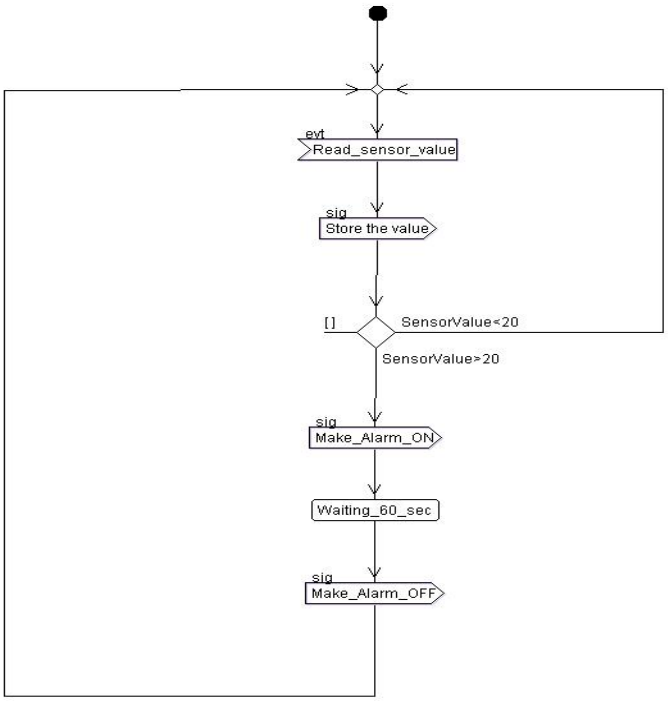


4. System Analysis

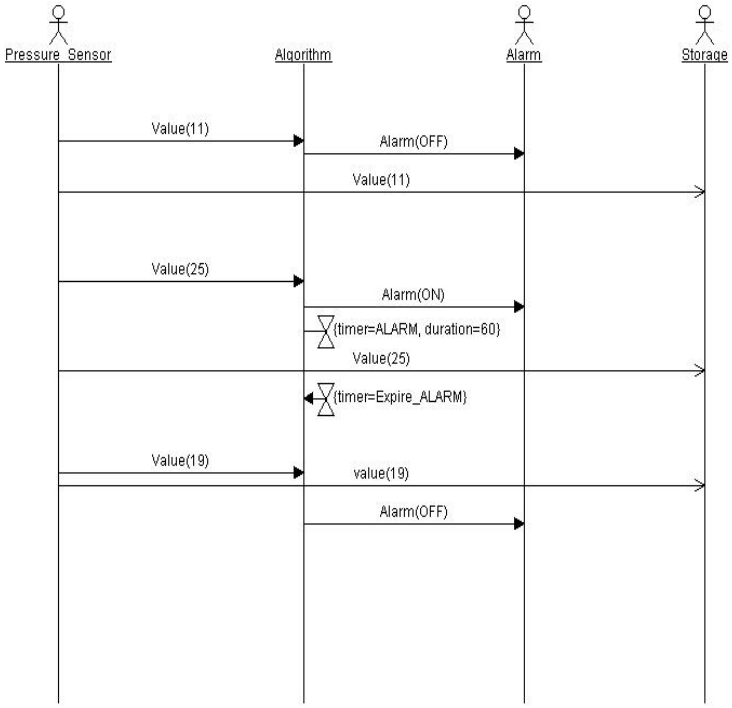
4.1 - use case Diagram



4.2- Activity Diagram

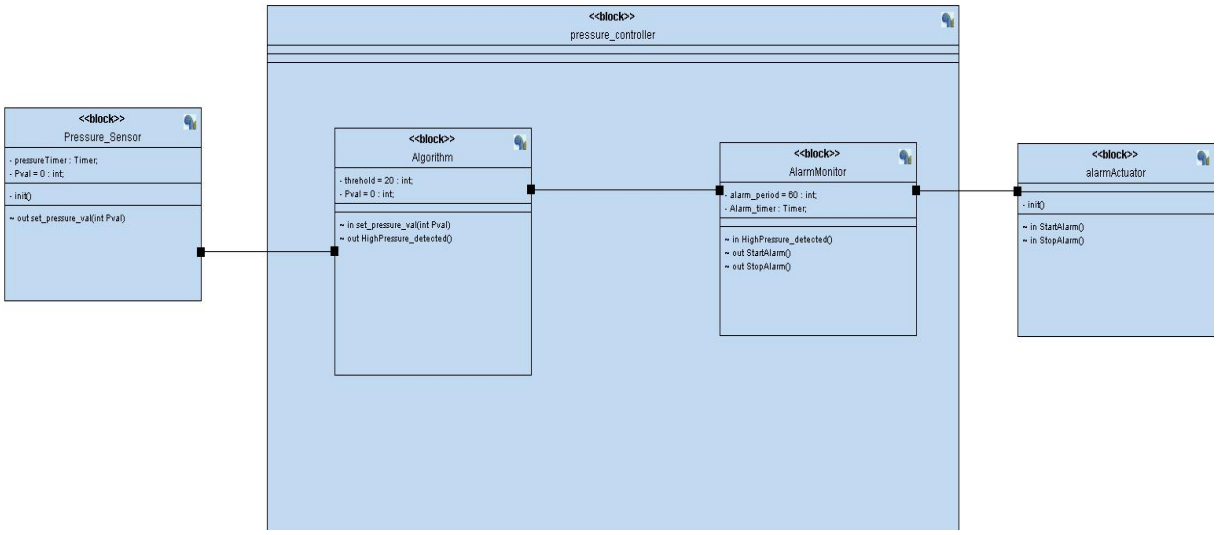


4.3- Sequence Diagram



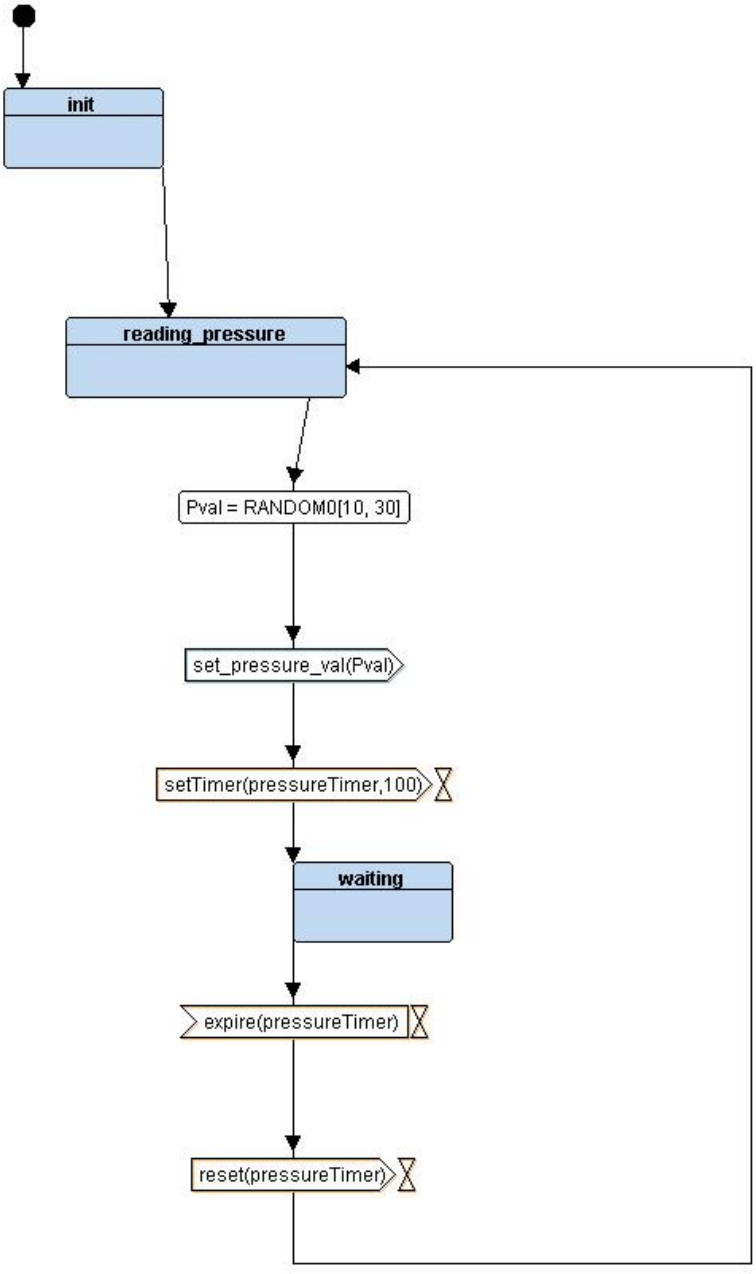
5- System Design

5.1 Block diagram

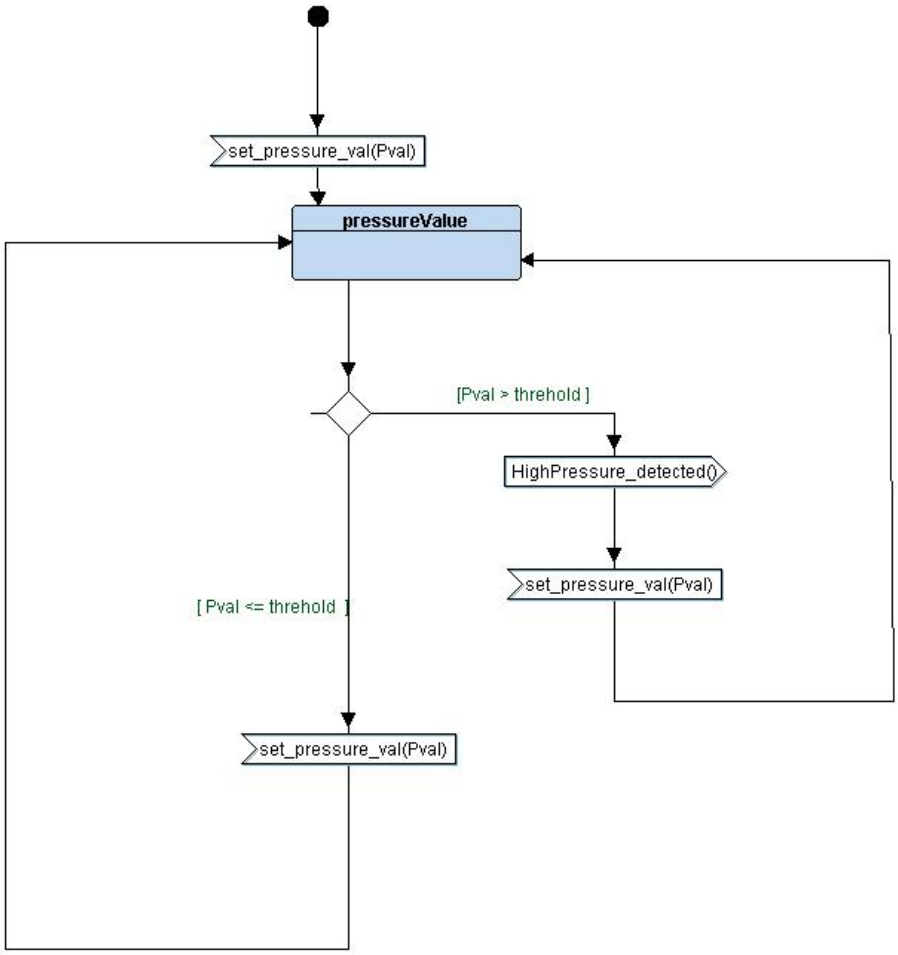


5.2 State Machine

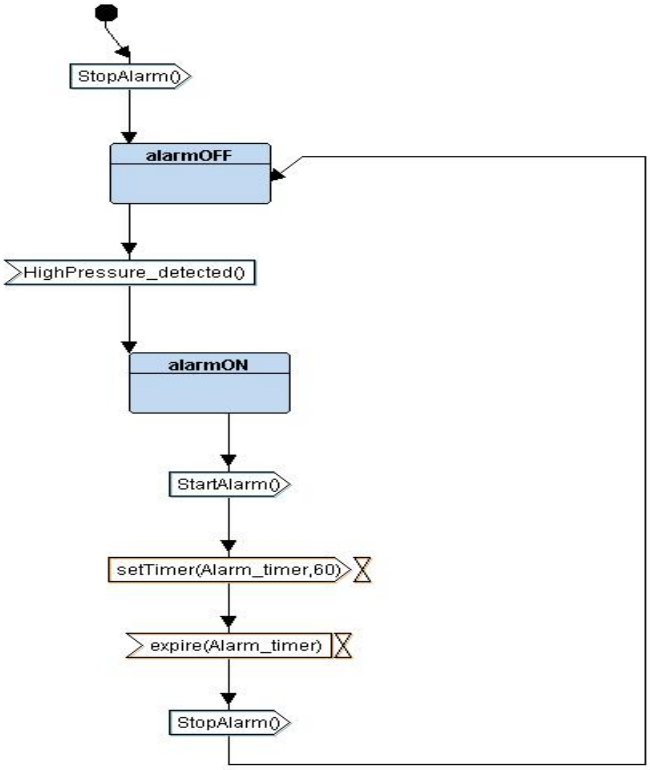
5.2.1 Pressure Sensor



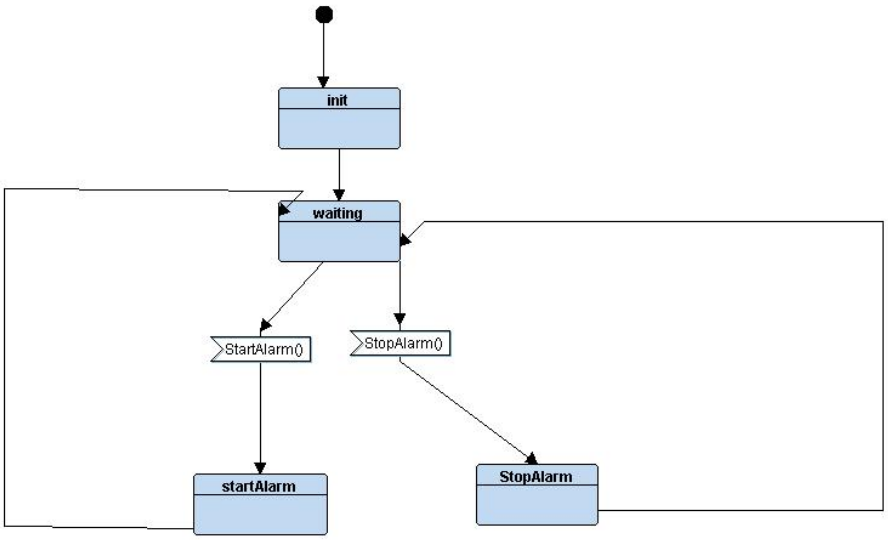
5.2.2 Main Algorithm



5.2.3 Alarm Monitor

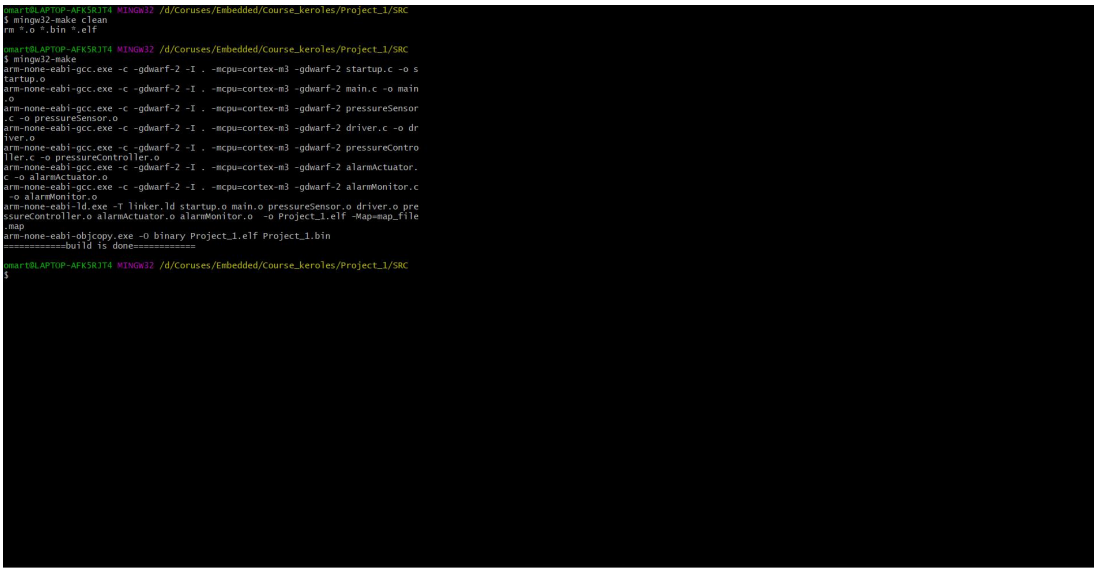


5.2.4 Alarm Actuator

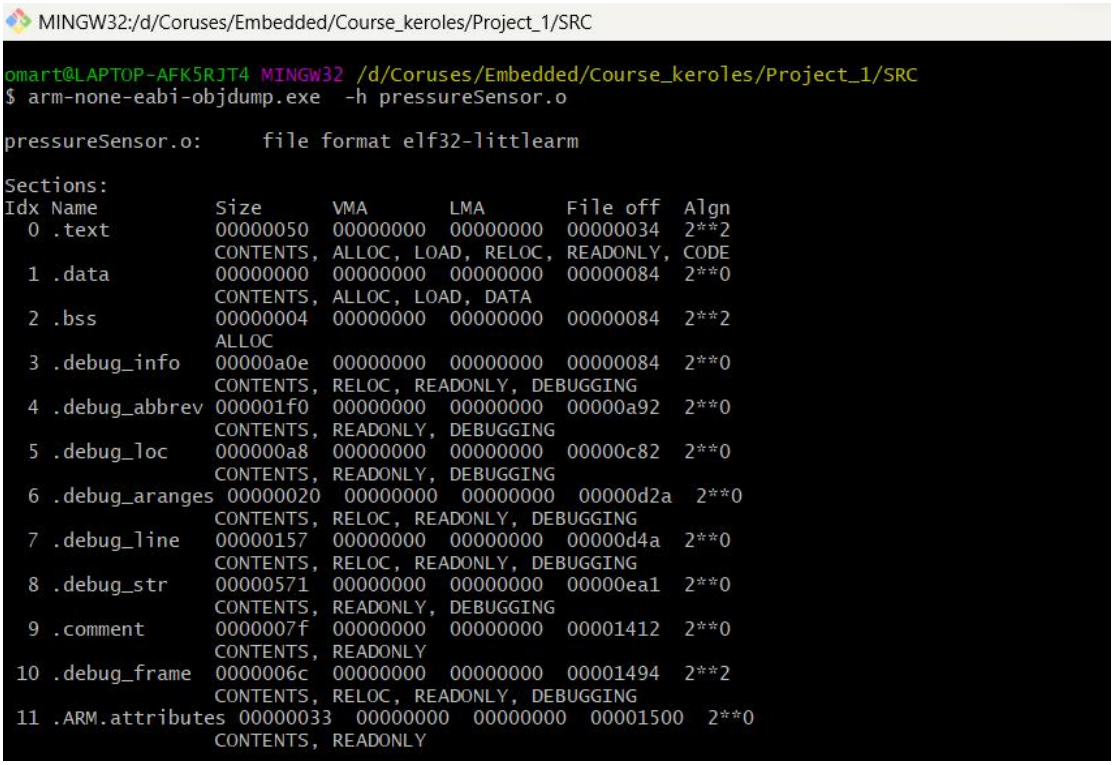


6- Software Analysis

6.1 arm compiler using our makefile



6.2 PressureSenor Output File



6.3 Startup Output File

MINGW32:/d/Coruses/Embedded/Course_keroles/Project_1/SRC

```
omart@LAPTOP-AFK5RJT4 MINGW32 /d/Coruses/Embedded/Course_keroles/Project_1/SRC
$ 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         00000090  00000000  00000000  00000034  2**2
                  CONTENTS, ALLOC, LOAD, RELOC, READONLY, CODE
  1 .data         00000000  00000000  00000000  000000c4  2**0
                  CONTENTS, ALLOC, LOAD, DATA
  2 .bss          00000000  00000000  00000000  000000c4  2**0
                  ALLOC
  3 .vectors      0000001c  00000000  00000000  000000c4  2**2
                  CONTENTS, ALLOC, LOAD, RELOC, DATA
  4 .debug_info   000001b4  00000000  00000000  000000e0  2**0
                  CONTENTS, RELOC, READONLY, DEBUGGING
  5 .debug_abbrev 000000d8  00000000  00000000  00000294  2**0
                  CONTENTS, READONLY, DEBUGGING
  6 .debug_loc    0000007c  00000000  00000000  0000036c  2**0
                  CONTENTS, READONLY, DEBUGGING
  7 .debug_aranges 00000020  00000000  00000000  000003e8  2**0
                  CONTENTS, RELOC, READONLY, DEBUGGING
  8 .debug_line   0000013c  00000000  00000000  00000408  2**0
                  CONTENTS, RELOC, READONLY, DEBUGGING
  9 .debug_str    000001d7  00000000  00000000  00000544  2**0
                  CONTENTS, READONLY, DEBUGGING
 10 .comment      0000007f  00000000  00000000  0000071b  2**0
                  CONTENTS, READONLY
 11 .debug_frame  00000050  00000000  00000000  0000079c  2**2
                  CONTENTS, RELOC, READONLY, DEBUGGING
 12 .ARM.attributes 00000033  00000000  00000000  000007ec  2**0
                  CONTENTS, READONLY
```

6.3 AlarmMonitor Output File

MINGW32:/d/Coruses/Embedded/Course_keroles/Project_1/SRC

```
omart@LAPTOP-AFK5RJT4 MINGW32 /d/Coruses/Embedded/Course_keroles/Project_1/SRC
$ arm-none-eabi-objdump.exe  -h alarmMonitor.o

alarmMonitor.o:     file format elf32-littlearm

Sections:
Idx Name          Size      VMA       LMA       File off  Algn
  0 .text         00000064  00000000  00000000  00000034  2**2
                  CONTENTS, ALLOC, LOAD, RELOC, READONLY, CODE
  1 .data         00000000  00000000  00000000  00000098  2**0
                  CONTENTS, ALLOC, LOAD, DATA
  2 .bss          00000000  00000000  00000000  00000098  2**0
                  ALLOC
  3 .debug_info   000009fd  00000000  00000000  00000098  2**0
                  CONTENTS, RELOC, READONLY, DEBUGGING
  4 .debug_abbrev 000001b6  00000000  00000000  00000a95  2**0
                  CONTENTS, READONLY, DEBUGGING
  5 .debug_loc    000000c8  00000000  00000000  00000c4b  2**0
                  CONTENTS, READONLY, DEBUGGING
  6 .debug_aranges 00000020  00000000  00000000  00000d13  2**0
                  CONTENTS, RELOC, READONLY, DEBUGGING
  7 .debug_line   00000157  00000000  00000000  00000d33  2**0
                  CONTENTS, RELOC, READONLY, DEBUGGING
  8 .debug_str    00000560  00000000  00000000  00000e8a  2**0
                  CONTENTS, READONLY, DEBUGGING
  9 .comment      0000007f  00000000  00000000  000013ea  2**0
                  CONTENTS, READONLY
 10 .debug_frame  00000084  00000000  00000000  0000146c  2**2
                  CONTENTS, RELOC, READONLY, DEBUGGING
 11 .ARM.attributes 00000033  00000000  00000000  000014f0  2**0
                  CONTENTS, READONLY
```

6.4 Project's Symbols

MINGW32:/d/Coruses/Embedded/Course_keroles/Project_1/SRC

```
omart@LAPTOP-AFK5RJT4 MINGW32 /d/Coruses/Embedded/Course_keroles/Project_1/SRC
$ arm-none-eabi-nm.exe  Project_1.elf
20000004 B _E_BSS
080002b0 D _E_DATA
080002b0 T _E_TEXT
20000000 B _S_BSS
080002b0 D _S_DATA
0800024c T alarminit
0800001c W Bus_fault_handler
0800001c T Default_handler
08000158 T Delay
08000178 T getPressureVal
080001cc T GPIO_INITIALIZATION
0800001c W H_fault_handler
0800014c T HighPressure_detected
080000d4 T main
0800001c W MM_fault_handler
0800001c W NMI_handler
08000116 T pressure_value
08000108 T pressureInit
20001008 B ptrAlarm
20001004 B ptrValue
20000000 B Pval
08000028 T reset_handler
08000190 T Set_Alarm_actuator
0800025a T setAlarm
080000ac T setup
20001004 B stack_top
08000274 T startAlarm
0800021c T StartAlarm
080002a0 T stopAlarm
0800023c T StopAlarm
0800001c W Usage_fault_handler
08000000 T vectors
```

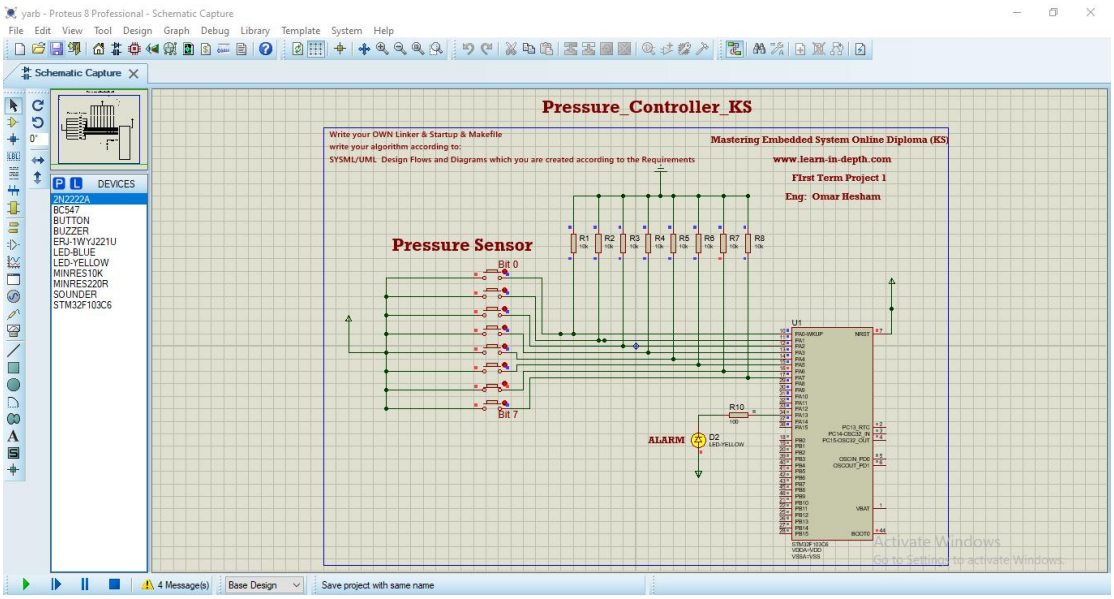
6.5 Disassembly of main.o

MINGW32:/d/Coruses/Embedded/Course_keroles/Project_1/SRC

```
omart@LAPTOP-AFK5RJT4 MINGW32 /d/Coruses/Embedded/Course_keroles/Project_1/SRC
$ arm-none-eabi-objdump.exe  -d main.o

main.o:     file format elf32-littlearm


Disassembly of section .text:

00000000 <setup>:
   0:   b480            push    {r7}
   2:   af00            add     r7, sp, #0
   4:   4b04            ldr     r3, [pc, #16]   ; (18 <setup+0x18>)
   6:   4a05            ldr     r2, [pc, #20]   ; (1c <setup+0x1c>)
   8:   601a            str     r2, [r3, #0]
   a:   4b05            ldr     r3, [pc, #20]   ; (20 <setup+0x20>)
   c:   4a05            ldr     r2, [pc, #20]   ; (24 <setup+0x24>)
   e:   601a            str     r2, [r3, #0]
  10:   bf00            nop
  12:   46bd            mov     sp, r7
  14:   bc80            pop     {r7}
  16:   4770            bx      lr
        ...

00000028 <main>:
  28:   b580            push    {r7, lr}
  2a:   af00            add     r7, sp, #0
  2c:   f7ff fffe       bl      0 <GPIO_INITIALIZATION>
  30:   f7ff fffe       bl      0 <alarminit>
  34:   f7ff fffe       bl      0 <setup>
  38:   f7ff fffe       bl      0 <pressure_value>
  3c:   4b05            ldr     r3, [pc, #20]   ; (54 <main+0x2c>)
  3e:   681b            ldr     r3, [r3, #0]
  40:   4798            blx     r3
  42:   4b05            ldr     r3, [pc, #20]   ; (58 <main+0x30>)
  44:   681b            ldr     r3, [r3, #0]
  46:   4798            blx     r3
  48:   f649 4040       movw    r0, #40000      ; 0x9c40
  4c:   f7ff fffe       bl      0 <Delay>
  50:   e7f2            b.n     38 <main+0x10>
  52:   bf00            nop
        ...
```

7- Proteus Simulation

When the pressure was more than 20 bar
Then the led turned on for 60 seconds



When the pressure was less than 20 bar
Then the led still off