Mastering Embedded System Online Diploma

www.learn-in-depth.com

First Term (Final project 1)

Engineer. Abdelrahman Aref Nadi

My profile:

https://www.learn-in-depth.com/online-diploma/abdo01445%40gmail.com

High pressure detection system

System design sequence:

1. Case study:

   The client expects the delivery of the software of the following system:

   - A pressure detection system that informs the crew of a cabin with an alarm when the pressure exceeds 20 bars in the cabin.
   - The alarm duration equals 60 seconds.

   Assumptions:

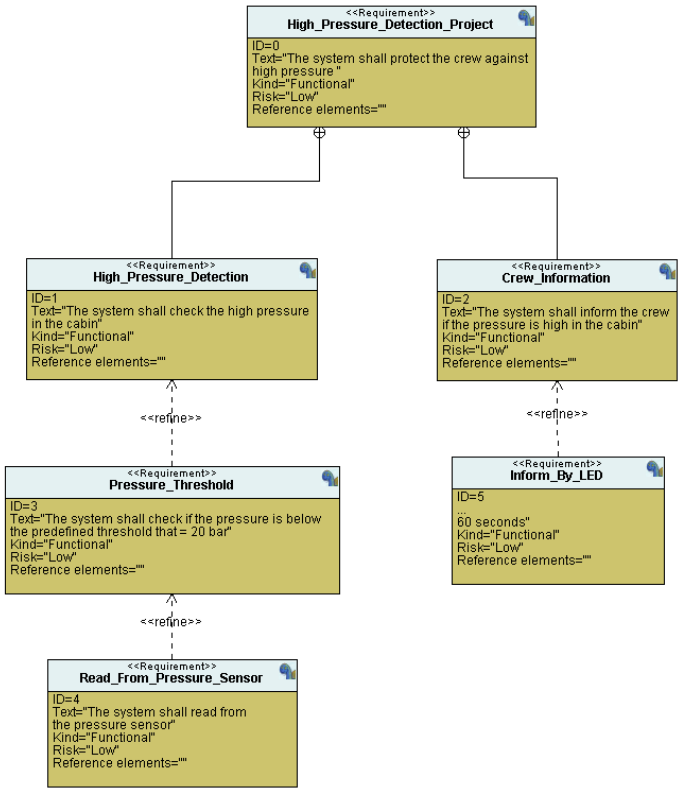
   - The controller setup and shut down procedures are not modeled.
   - The controller maintenance is not modeled.
   - The pressure sensor never fails.
   - The alarm never fails.
   - The controller never faces power cut.

2. Method:

   The chosen method in designing and implementing this system is the V-model method.

3. System requirements:

   The following diagram is the UML requirement diagram for this system.



4. Space exploration and hardware / software partitioning:

Hardware:

- Controller: STM32F103C6.
- Alarm: LED.
- Sensor: Pressure sensor.

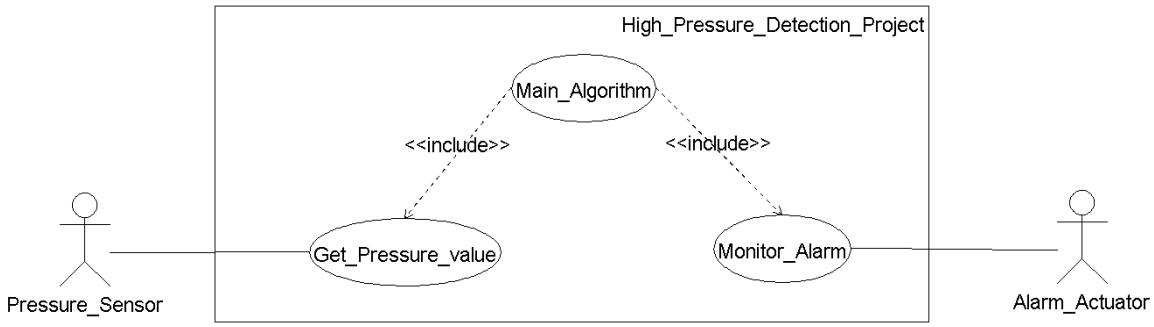
Software:

Here, we have four software modules:

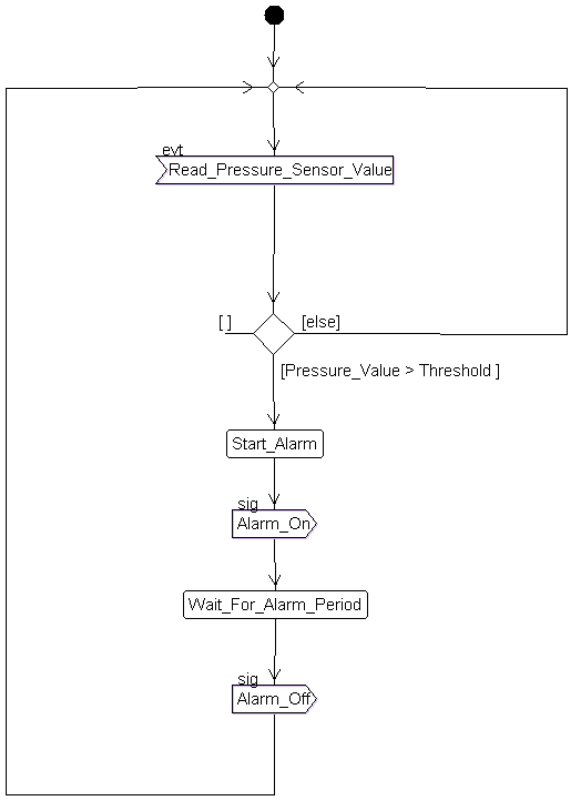
- Pressure sensor driver.
- Main algorithm.
- Alarm monitor.
- Alarm actuator driver.

5. System analysis:

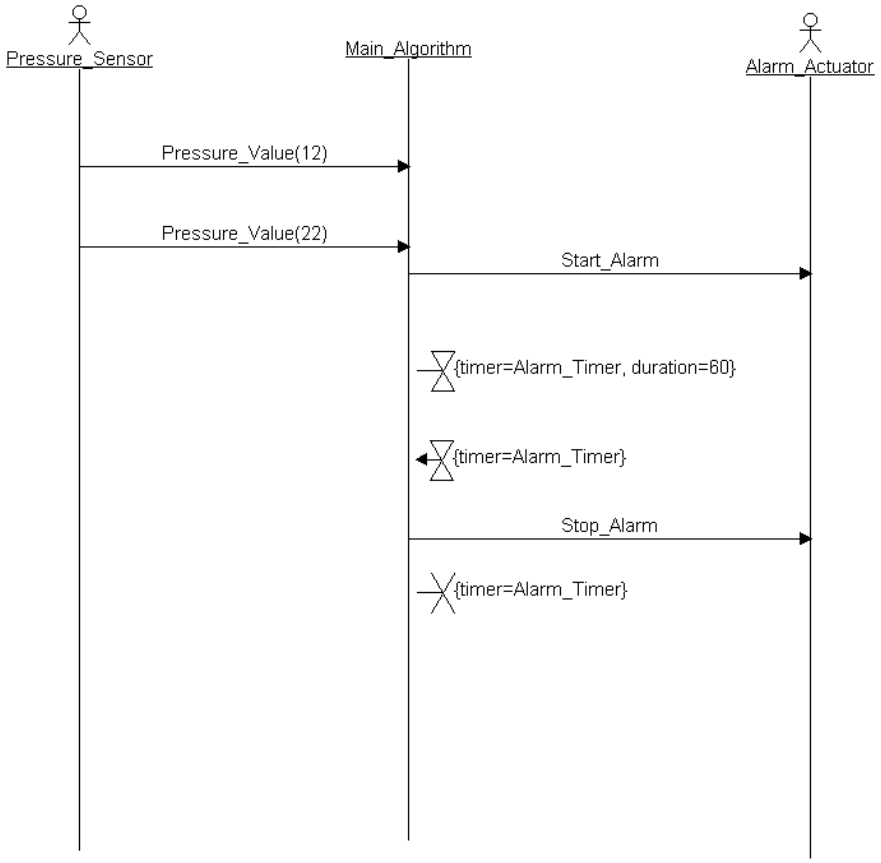
5.1. Use Case Diagram:



5.2. Activity Diagram:

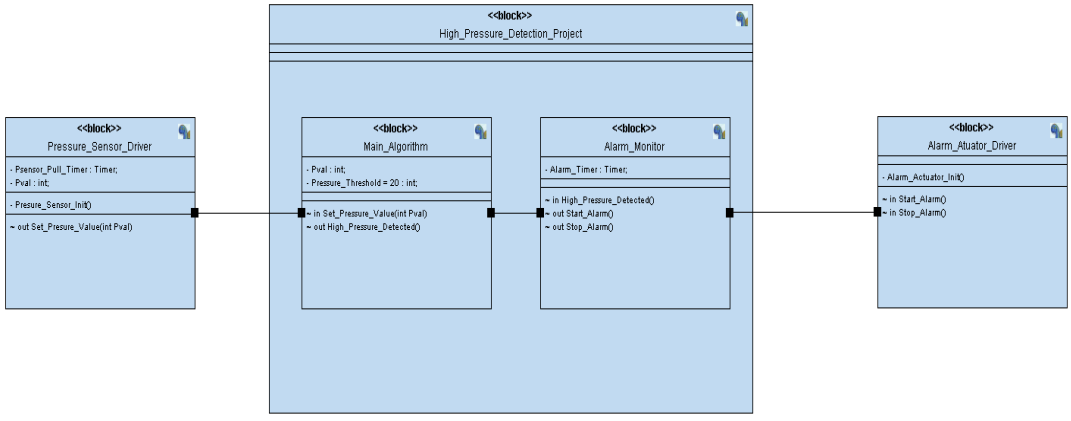


5.3. Sequence diagram:



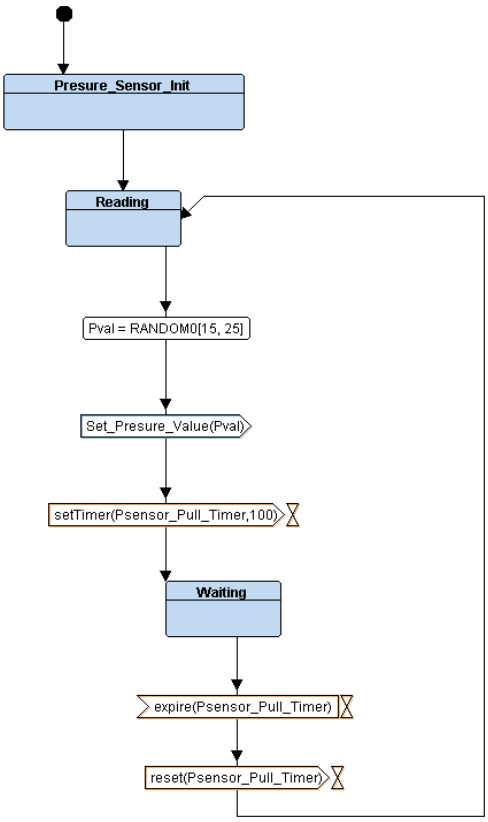
6. System design:

6.1. System Block Diagram

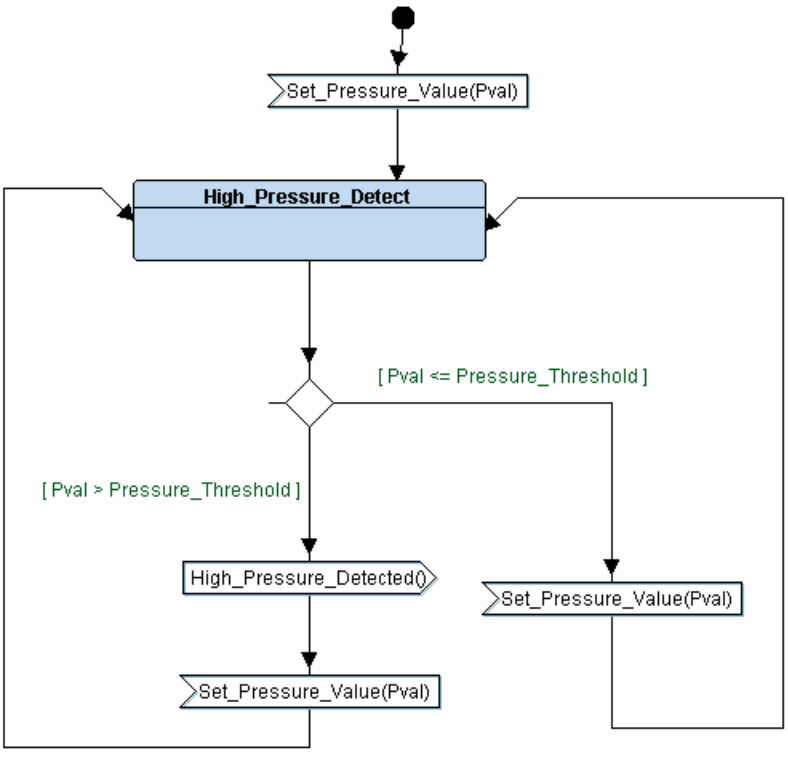


6.2. Software components state machines

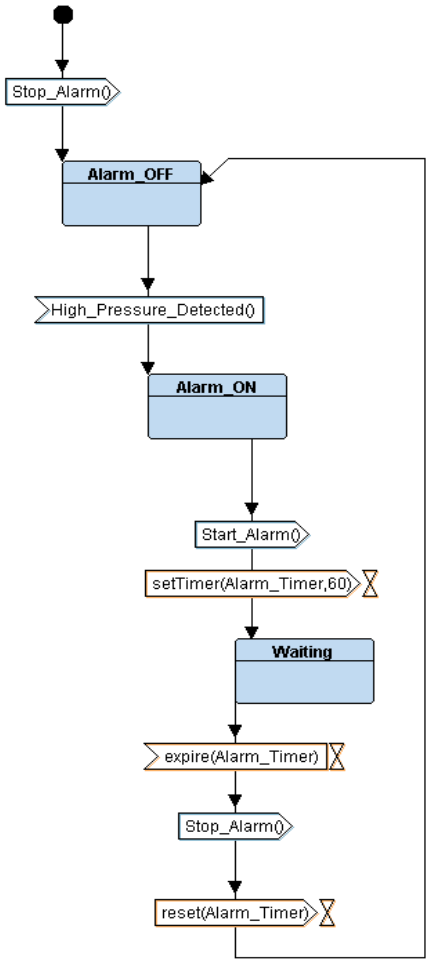
- **Pressure sensor driver state machine.**



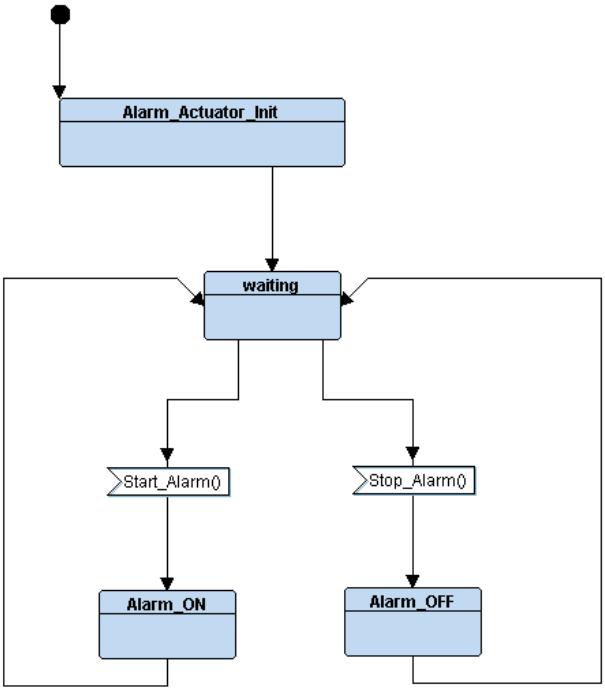
- **Main algorithm state machine.**



- **Alarm monitor state machine.**



- **Alarm actuator driver state machine.**

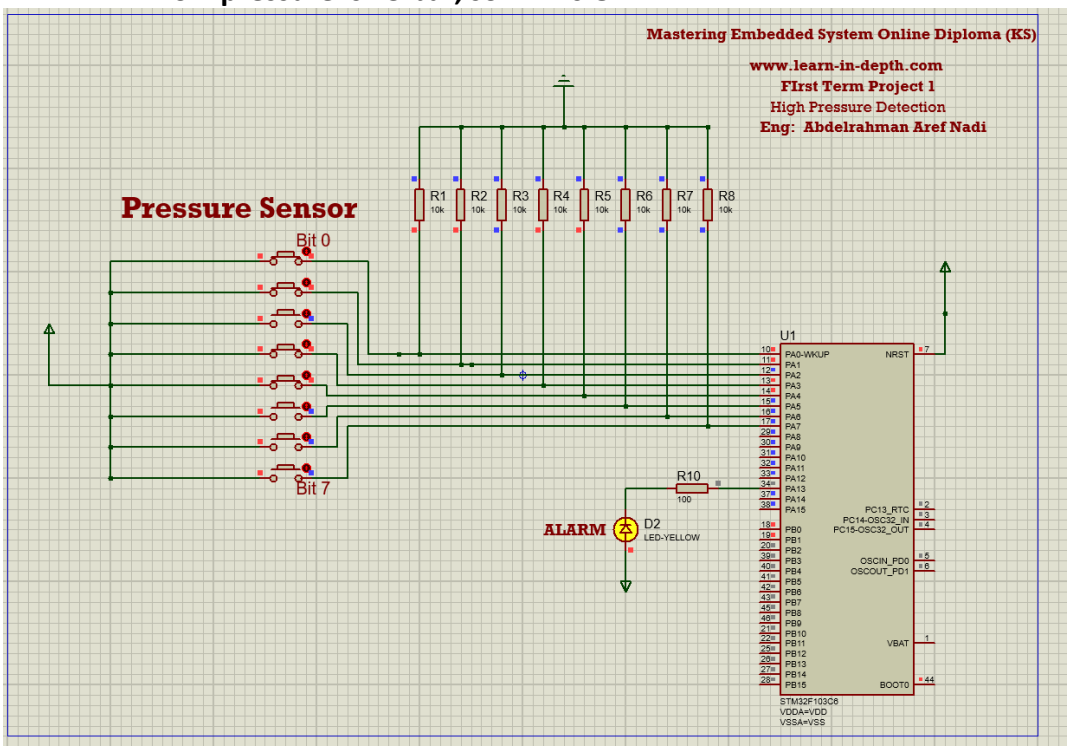


7. Appendix:

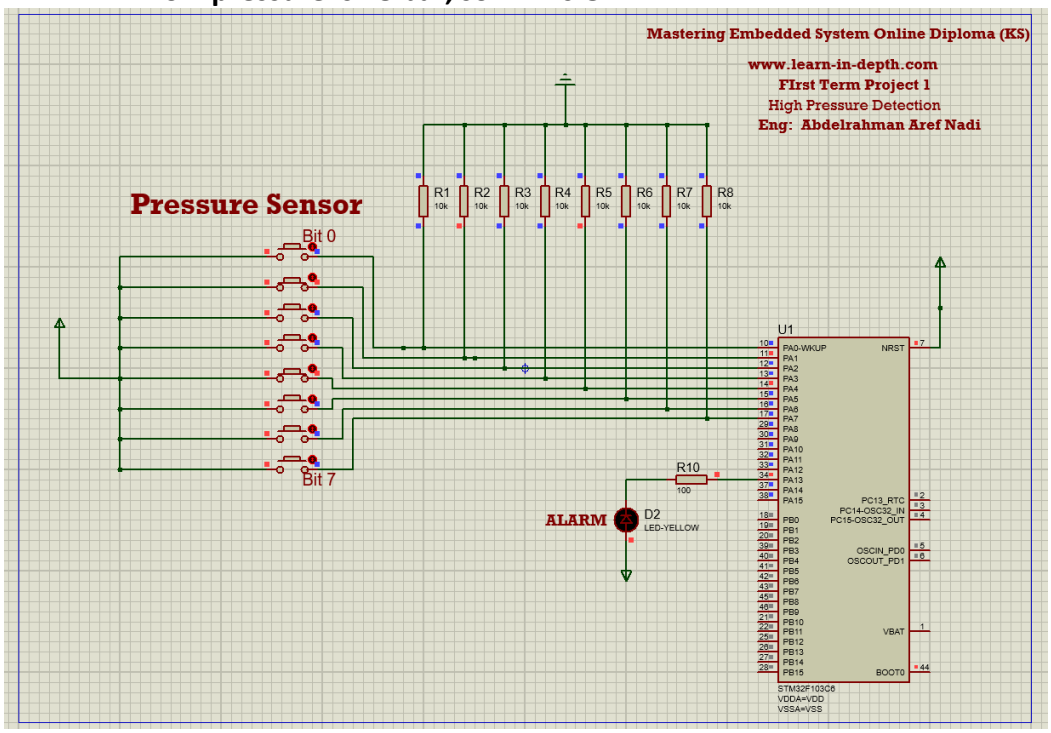
7.1. Simulation on proteus:

Here, we have two cases in simulation:

- LED is ON: pressure is greater than 20 bar.
  Now pressure is 25 bar, so LED is ON.



- LED is OFF: pressure is less than 20 bar.
  Now pressure is 18 bar, so LED is OFF.



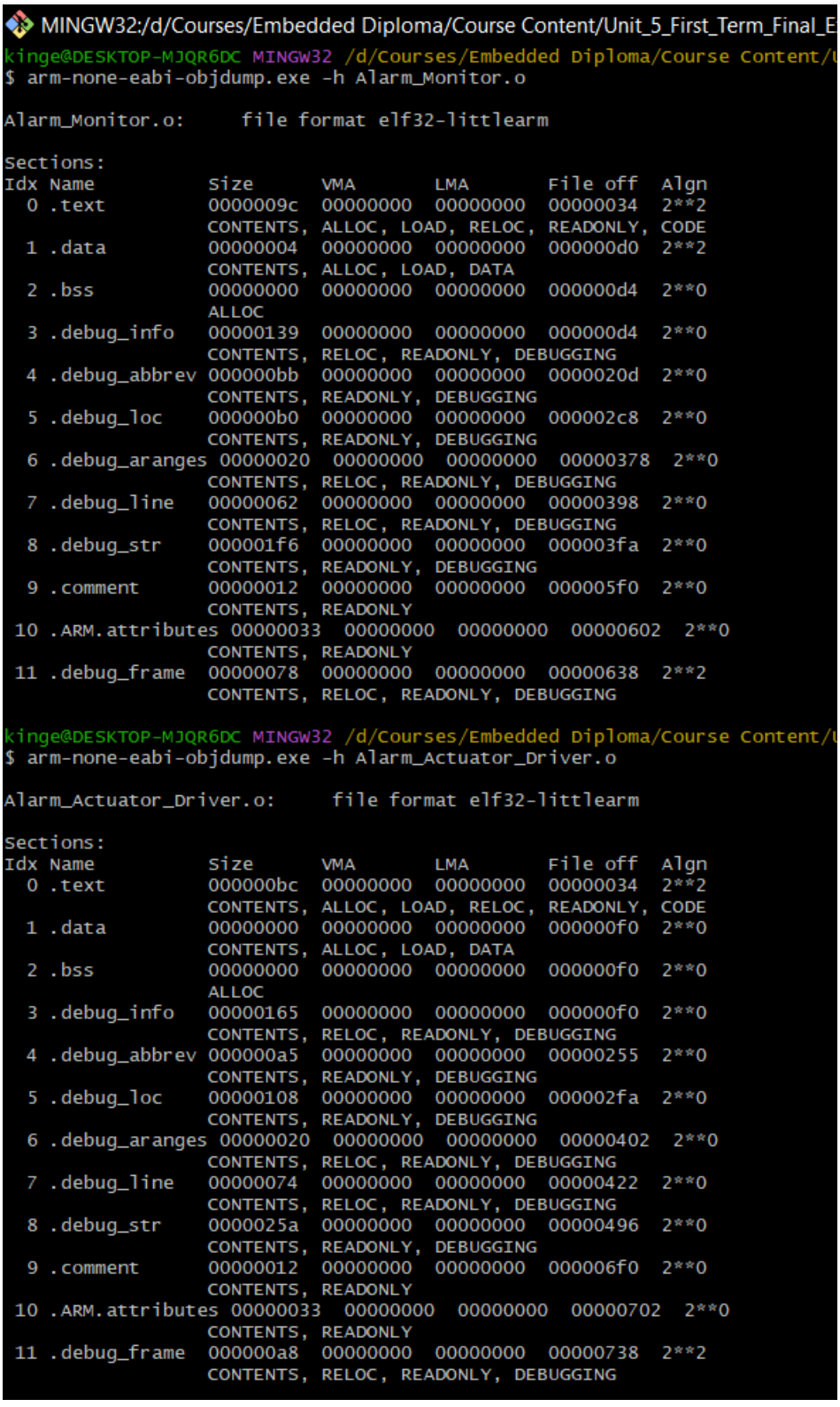
7.2. Software analysis:

- Software building using ARM cross toolchain.

```
MINGW32:/d/Courses/Embedded Diploma/Course Content/Unit_5_First_Term_Final_Exams_&_Projects/2_Project_1/FIRST_TERM_project1/Src

kinge@DESKTOP-MJQR6DC MINGW32 /d/Courses/Embedded Diploma/Course Content/Unit_5_First_Term_Final_Exams_&_Projects/2_Project_1/FIRST_TERM_project1/Src
$ make
arm-none-eabi-gcc.exe -mcpu=cortex-m3 -gdwarf-2 -mthumb -I . -c startup.c -o startup.o
arm-none-eabi-gcc.exe -mcpu=cortex-m3 -gdwarf-2 -mthumb -I . -c Main_Algorithm.c -o Main_Algorithm.o
arm-none-eabi-gcc.exe -mcpu=cortex-m3 -gdwarf-2 -mthumb -I . -c main.c -o main.o
arm-none-eabi-gcc.exe -mcpu=cortex-m3 -gdwarf-2 -mthumb -I . -c Pressure_Sensor_Driver.c -o Pressure_Sensor_Driver.o
arm-none-eabi-gcc.exe -mcpu=cortex-m3 -gdwarf-2 -mthumb -I . -c Alarm_Actuator_Driver.c -o Alarm_Actuator_Driver.o
arm-none-eabi-gcc.exe -mcpu=cortex-m3 -gdwarf-2 -mthumb -I . -c driver.c -o driver.o
arm-none-eabi-gcc.exe -mcpu=cortex-m3 -gdwarf-2 -mthumb -I . -c Alarm_Monitor.c -o Alarm_Monitor.o
arm-none-eabi-ld.exe -T linker_script.ld startup.o Main_Algorithm.o main.o Pressure_Sensor_Driver.o Alarm_Actuator_Driver.o driver.o Alarm_Monitor.o   -Map=Map_File.map -o HighPressureDetection.elf
arm-none-eabi-objcopy.exe -O binary HighPressureDetection.elf HighPressureDetection.bin
Build is Done ...

kinge@DESKTOP-MJQR6DC MINGW32 /d/Courses/Embedded Diploma/Course Content/Unit_5_First_Term_Final_Exams_&_Projects/2_Project_1/FIRST_TERM_project1/Src
$ ls
Alarm_Actuator_Driver.c  Alarm_Monitor.c  HighPressureDetection.bin  Main_Algorithm.h  Map_File.map            Pressure_Sensor_Driver.h  driver.h         main.c     startup.o
Alarm_Actuator_Driver.h  Alarm_Monitor.h  HighPressureDetection.elf  Main_Algorithm.o  Platforms.h             Pressure_Sensor_Driver.o  driver.o         main.o     state.h
Alarm_Actuator_Driver.o  Alarm_Monitor.o  Main_Algorithm.c           Makefile          Pressure_Sensor_Driver.c  driver.c                  linker_script.ld  startup.c
```

- Analyzing sections in output object files and final elf image.

```
MINGW32:/d/Courses/Embedded Diploma/Course Content/Unit_5_First_Term_Fin

kinge@DESKTOP-MJQR6DC MINGW32 /d/Courses/Embedded Diploma/Course Conte
$ 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         0000007c  00000000  00000000  00000034  2**2
                  CONTENTS, ALLOC, LOAD, RELOC, READONLY, CODE
  1 .data         00000000  00000000  00000000  000000b0  2**0
                  CONTENTS, ALLOC, LOAD, DATA
  2 .bss          00000000  00000000  00000000  000000b0  2**0
                  ALLOC
  3 .debug_info   0000012d  00000000  00000000  000000b0  2**0
                  CONTENTS, RELOC, READONLY, DEBUGGING
  4 .debug_abbrev 000000c7  00000000  00000000  000001dd  2**0
                  CONTENTS, READONLY, DEBUGGING
  5 .debug_loc    00000090  00000000  00000000  000002a4  2**0
                  CONTENTS, READONLY, DEBUGGING
  6 .debug_aranges 00000020  00000000  00000000  00000334  2**0
                  CONTENTS, RELOC, READONLY, DEBUGGING
  7 .debug_line   0000006e  00000000  00000000  00000354  2**0
                  CONTENTS, RELOC, READONLY, DEBUGGING
  8 .debug_str    0000020d  00000000  00000000  000003c2  2**0
                  CONTENTS, READONLY, DEBUGGING
  9 .comment      00000012  00000000  00000000  000005cf  2**0
                  CONTENTS, READONLY
 10 .ARM.attributes 00000033  00000000  00000000  000005e1  2**0
                  CONTENTS, READONLY
 11 .debug_frame  00000060  00000000  00000000  00000614  2**2
                  CONTENTS, RELOC, READONLY, DEBUGGING

kinge@DESKTOP-MJQR6DC MINGW32 /d/Courses/Embedded Diploma/Course Conte
$ arm-none-eabi-objdump.exe -h Main_Algorithm.o

Main_Algorithm.o:     file format elf32-littlearm

Sections:
Idx Name          Size      VMA       LMA       File off  Algn
  0 .text         00000060  00000000  00000000  00000034  2**2
                  CONTENTS, ALLOC, LOAD, RELOC, READONLY, CODE
  1 .data         00000004  00000000  00000000  00000094  2**2
                  CONTENTS, ALLOC, LOAD, DATA
  2 .bss          00000004  00000000  00000000  00000098  2**2
                  ALLOC
  3 .debug_info   00000115  00000000  00000000  00000098  2**0
                  CONTENTS, RELOC, READONLY, DEBUGGING
  4 .debug_abbrev 000000a5  00000000  00000000  000001ad  2**0
                  CONTENTS, READONLY, DEBUGGING
  5 .debug_loc    00000064  00000000  00000000  00000252  2**0
                  CONTENTS, READONLY, DEBUGGING
  6 .debug_aranges 00000020  00000000  00000000  000002b6  2**0
                  CONTENTS, RELOC, READONLY, DEBUGGING
  7 .debug_line   00000062  00000000  00000000  000002d6  2**0
                  CONTENTS, RELOC, READONLY, DEBUGGING
  8 .debug_str    000001be  00000000  00000000  00000338  2**0
                  CONTENTS, READONLY, DEBUGGING
  9 .comment      00000012  00000000  00000000  000004f6  2**0
                  CONTENTS, READONLY
 10 .ARM.attributes 00000033  00000000  00000000  00000508  2**0
                  CONTENTS, READONLY
 11 .debug_frame  00000048  00000000  00000000  0000053c  2**2
                  CONTENTS, RELOC, READONLY, DEBUGGING
```


MINGW32:/d/Courses/Embedded Diploma/Course Content/Unit_5_First_Term_Final_E
kinge@DESKTOP-MJQR6DC MINGW32 /d/Courses/Embedded Diploma/Course Content/U
$ arm-none-eabi-objdump.exe -h Alarm_Monitor.o
Alarm_Monitor.o: file format elf32-littlearm
Sections:
Idx Name Size VMA LMA File off Algn
0 .text 0000009c 00000000 00000000 00000034 2**2
CONTENTS, ALLOC, LOAD, RELOC, READONLY, CODE
1 .data 00000004 00000000 00000000 000000d0 2**2
CONTENTS, ALLOC, LOAD, DATA
2 .bss 00000000 00000000 00000000 000000d4 2**0
ALLOC
3 .debug_info 00000139 00000000 00000000 000000d4 2**0
CONTENTS, RELOC, READONLY, DEBUGGING
4 .debug_abbrev 000000bb 00000000 00000000 0000020d 2**0
CONTENTS, READONLY, DEBUGGING
5 .debug_loc 000000b0 00000000 00000000 000002c8 2**0
CONTENTS, READONLY, DEBUGGING
6 .debug_aranges 00000020 00000000 00000000 00000378 2**0
CONTENTS, RELOC, READONLY, DEBUGGING
7 .debug_line 00000062 00000000 00000000 00000398 2**0
CONTENTS, RELOC, READONLY, DEBUGGING
8 .debug_str 000001f6 00000000 00000000 000003fa 2**0
CONTENTS, READONLY, DEBUGGING
9 .comment 00000012 00000000 00000000 000005f0 2**0
CONTENTS, READONLY
10 .ARM.attributes 00000033 00000000 00000000 00000602 2**0
CONTENTS, READONLY
11 .debug_frame 00000078 00000000 00000000 00000638 2**2
CONTENTS, RELOC, READONLY, DEBUGGING
kinge@DESKTOP-MJQR6DC MINGW32 /d/Courses/Embedded Diploma/Course Content/U
$ arm-none-eabi-objdump.exe -h Alarm_Actuator_Driver.o
Alarm_Actuator_Driver.o: file format elf32-littlearm
Sections:
Idx Name Size VMA LMA File off Algn
0 .text 000000bc 00000000 00000000 00000034 2**2
CONTENTS, ALLOC, LOAD, RELOC, READONLY, CODE
1 .data 00000000 00000000 00000000 000000f0 2**0
CONTENTS, ALLOC, LOAD, DATA
2 .bss 00000000 00000000 00000000 000000f0 2**0
ALLOC
3 .debug_info 00000165 00000000 00000000 000000f0 2**0
CONTENTS, RELOC, READONLY, DEBUGGING
4 .debug_abbrev 000000a5 00000000 00000000 00000255 2**0
CONTENTS, READONLY, DEBUGGING
5 .debug_loc 00000108 00000000 00000000 000002fa 2**0
CONTENTS, READONLY, DEBUGGING
6 .debug_aranges 00000020 00000000 00000000 00000402 2**0
CONTENTS, RELOC, READONLY, DEBUGGING
7 .debug_line 00000074 00000000 00000000 00000422 2**0
CONTENTS, RELOC, READONLY, DEBUGGING
8 .debug_str 0000025a 00000000 00000000 00000496 2**0
CONTENTS, READONLY, DEBUGGING
9 .comment 00000012 00000000 00000000 000006f0 2**0
CONTENTS, READONLY
10 .ARM.attributes 00000033 00000000 00000000 00000702 2**0
CONTENTS, READONLY
11 .debug_frame 000000a8 00000000 00000000 00000738 2**2
CONTENTS, RELOC, READONLY, DEBUGGING

```
MINGW32:/d/Courses/Embedded Diploma/Course Content/Unit_5_First_Term_Fin

kinge@DESKTOP-MJQR6DC MINGW32 /d/Courses/Embedded Diploma/Course Conte
$ arm-none-eabi-objdump.exe -h  HighPressureDetection.elf

HighPressureDetection.elf:     file format elf32-littlearm

Sections:
Idx Name          Size      VMA       LMA       File off  Algn
  0 .text         000004b8  08000000  08000000  00008000  2**2
                  CONTENTS, ALLOC, LOAD, READONLY, CODE
  1 .data         00000008  20000000  080004b8  00010000  2**2
                  CONTENTS, ALLOC, LOAD, DATA
  2 .bss          00001018  20000008  080004c0  00010008  2**2
                  ALLOC
  3 .debug_info   000008f5  00000000  00000000  00010008  2**0
                  CONTENTS, READONLY, DEBUGGING
  4 .debug_abbrev 000004ce  00000000  00000000  000108fd  2**0
                  CONTENTS, READONLY, DEBUGGING
  5 .debug_loc    00000430  00000000  00000000  00010dcb  2**0
                  CONTENTS, READONLY, DEBUGGING
  6 .debug_aranges 000000e0  00000000  00000000  000111fb  2**0
                  CONTENTS, READONLY, DEBUGGING
  7 .debug_line   00000372  00000000  00000000  000112db  2**0
                  CONTENTS, READONLY, DEBUGGING
  8 .debug_str    000004c8  00000000  00000000  0001164d  2**0
                  CONTENTS, READONLY, DEBUGGING
  9 .comment      00000011  00000000  00000000  00011b15  2**0
                  CONTENTS, READONLY
 10 .ARM.attributes 00000033  00000000  00000000  00011b26  2**0
                  CONTENTS, READONLY
 11 .debug_frame  000002d4  00000000  00000000  00011b5c  2**2
                  CONTENTS, READONLY, DEBUGGING
```

- Analyzing symbol tables in output object files and final elf image.
  - **A. Before Linking & Resolving (All .o files)**

```
MINGW32:/d/Courses/Embedded Diploma/Course Content/Unit_5_First_Term_F

kinge@DESKTOP-MJQR6DC MINGW32 /d/Courses/Embedded Diplo
$ arm-none-eabi-nm.exe Pressure_Sensor_Driver.o
         U Delay
         U getPressureVal
00000000 T Pressure_Sensor_Driver_init
00000004 C PRESSURE_SENSOR_DRIVER_state
00000001 C PRESSURE_SENSOR_DRIVER_state_id
         U Set_Pressure_value
0000000c T ST_PRESSURE_SENSOR_DRIVER_reading
00000050 T ST_PRESSURE_SENSOR_DRIVER_waiting

kinge@DESKTOP-MJQR6DC MINGW32 /d/Courses/Embedded Diplo
$ arm-none-eabi-nm.exe Main_Algorithm.o
         U High_Pressure_Detected
00000000 B Main_Algorithm_pVal
00000004 C MAIN_ALGORITHM_state
00000001 C MAIN_ALGORITHM_state_id
00000000 T Set_Pressure_value
00000030 T ST_MAIN_ALGORITHM_high_pressure_detect
00000000 D Threshold

kinge@DESKTOP-MJQR6DC MINGW32 /d/Courses/Embedded Diplo
$ arm-none-eabi-nm.exe Alarm_Monitor.o
00000004 C ALARM_MONITOR_state
00000001 C ALARM_MONITOR_state_id
00000000 D Alarm_Timer
         U Delay
00000000 T High_Pressure_Detected
0000001c T ST_ALARM_MONITOR_alarm_off
00000034 T ST_ALARM_MONITOR_alarm_on
00000070 T ST_ALARM_MONITOR_waiting
         U Start_Alarm
         U Stop_Alarm

kinge@DESKTOP-MJQR6DC MINGW32 /d/Courses/Embedded Diplo
$ arm-none-eabi-nm.exe Alarm_Actuator_Driver.o
00000000 T Alarm_Actuator_Driver_init
00000004 C ALARM_ACTUATOR_DRIVER_state
00000001 C ALARM_ACTUATOR_DRIVER_state_id
         U Set_Alarm_actuator
00000044 T ST_ALARM_ACTUATOR_DRIVER_alarm_off
00000074 T ST_ALARM_ACTUATOR_DRIVER_alarm_on
000000a4 T ST_ALARM_ACTUATOR_DRIVER_waiting
0000000c T Start_Alarm
00000028 T Stop_Alarm
```

B. After Linking & Resolving (.elf file)

```
kinge@DESKTOP-MJQR6DC MINGW32 /d/Courses/Embedded D
$ arm-none-eabi-nm.exe  HighPressureDetection.elf
2000000c B _E_BSS_
20000008 D _E_DATA_
080004b8 T _E_TEXT_
20000008 B _S_BSS_
20000000 D _S_DATA_
08000254 T Alarm_Actuator_Driver_init
20001018 B ALARM_ACTUATOR_DRIVER_state
20001013 B ALARM_ACTUATOR_DRIVER_state_id
2000101c B ALARM_MONITOR_state
20001011 B ALARM_MONITOR_state_id
20000004 D Alarm_Timer
0800001c W Bus_Fault
0800001c T Default_Handler
08000310 T Delay
08000334 T getPressureVal
0800039c T GPIO_INITIALIZATION
0800001c W H_Fault_Handler
0800041c T High_Pressure_Detected
0800019c T main
20000008 B Main_Algorithm_pVal
2000100c B MAIN_ALGORITHM_state
20001010 B MAIN_ALGORITHM_state_id
0800001c W MM_Fault_Handler
0800001c W NMI_Handler
080001d8 T Pressure_Sensor_Driver_init
20001014 B PRESSURE_SENSOR_DRIVER_state
20001012 B PRESSURE_SENSOR_DRIVER_state_id
08000028 T Reset_Handler
0800034c T Set_Alarm_actuator
080000e0 T Set_Pressure_value
08000140 T setup
08000298 T ST_ALARM_ACTUATOR_DRIVER_alarm_off
080002c8 T ST_ALARM_ACTUATOR_DRIVER_alarm_on
080002f8 T ST_ALARM_ACTUATOR_DRIVER_waiting
08000438 T ST_ALARM_MONITOR_alarm_off
08000450 T ST_ALARM_MONITOR_alarm_on
0800048c T ST_ALARM_MONITOR_waiting
08000110 T ST_MAIN_ALGORITHM_high_pressure_detect
080001e4 T ST_PRESSURE_SENSOR_DRIVER_reading
08000228 T ST_PRESSURE_SENSOR_DRIVER_waiting
2000100c B Stack_Top
08000260 T Start_Alarm
0800027c T Stop_Alarm
20000000 D Threshold
0800001c W Usage_Fault_Handler
08000000 T vectors
```

- Analyzing the final elf image.

```
kinge@DESKTOP-MJQR6DC MINGW32 /d/Courses/Embedded Diploma/Course Content/Unit_5_Fi
$ arm-none-eabi-readelf.exe -S  HighPressureDetection.elf
There are 16 section headers, starting at offset 0x11ed0:

Section Headers:
  [Nr] Name              Type            Addr     Off    Size   ES Flg Lk Inf Al
  [ 0]                   NULL            00000000 000000 000000 00      0   0  0
  [ 1] .text             PROGBITS        08000000 008000 0004b8 00  AX  0   0  4
  [ 2] .data             PROGBITS        20000000 010000 000008 00  WA  0   0  4
  [ 3] .bss              NOBITS          20000008 010008 001018 00  WA  0   0  4
  [ 4] .debug_info       PROGBITS        00000000 010008 0008f5 00      0   0  1
  [ 5] .debug_abbrev     PROGBITS        00000000 0108fd 0004ce 00      0   0  1
  [ 6] .debug_loc        PROGBITS        00000000 010dcb 000430 00      0   0  1
  [ 7] .debug_aranges    PROGBITS        00000000 0111fb 0000e0 00      0   0  1
  [ 8] .debug_line       PROGBITS        00000000 0112db 000372 00      0   0  1
  [ 9] .debug_str        PROGBITS        00000000 01164d 0004c8 01  MS  0   0  1
  [10] .comment          PROGBITS        00000000 011b15 000011 01  MS  0   0  1
  [11] .ARM.attributes   ARM_ATTRIBUTES  00000000 011b26 000033 00      0   0  1
  [12] .debug_frame      PROGBITS        00000000 011b5c 0002d4 00      0   0  4
  [13] .shstrtab         STRTAB          00000000 011e30 00009d 00      0   0  1
  [14] .symtab           SYMTAB          00000000 012150 000550 10     15  39  4
  [15] .strtab           STRTAB          00000000 0126a0 0003f1 00      0   0  1
Key to Flags:
  W (write), A (alloc), X (execute), M (merge), S (strings)
  I (info), L (link order), G (group), T (TLS), E (exclude), x (unknown)
  O (extra OS processing required) o (OS specific), p (processor specific)

kinge@DESKTOP-MJQR6DC MINGW32 /d/Courses/Embedded Diploma/Course Content/Unit_5_
$ arm-none-eabi-readelf.exe -a  HighPressureDetection.elf
ELF Header:
  Magic:   7f 45 4c 46 01 01 01 00 00 00 00 00 00 00 00 00
  Class:                             ELF32
  Data:                              2's complement, little endian
  Version:                           1 (current)
  OS/ABI:                            UNIX - System V
  ABI Version:                       0
  Type:                              EXEC (Executable file)
  Machine:                           ARM
  Version:                           0x1
  Entry point address:               0x8000000
  Start of program headers:          52 (bytes into file)
  Start of section headers:          73424 (bytes into file)
  Flags:                             0x5000002, has entry point, Version5 EABI
  Size of this header:               52 (bytes)
  Size of program headers:           32 (bytes)
  Number of program headers:         2
  Size of section headers:           40 (bytes)
  Number of section headers:         16
  Section header string table index: 13
```

7.3. You can also watch a video of simulation. And download all simulation files. My google drive:

https://drive.google.com/drive/folders/1Y-65JLfBae1A97ccEnGubEwZs7nAutDI?usp=share_link

7.4. You can also see all (.c) , (.h) , (Makefile) , (Linker Script) , (Startup file) and (.map file) in my github repository :

https://github.com/aaref5720/Master_Embedded_Systems/tree/main/Unit_5_First_Term_Projects/Project_1_Pressure_Detection/Source_Code