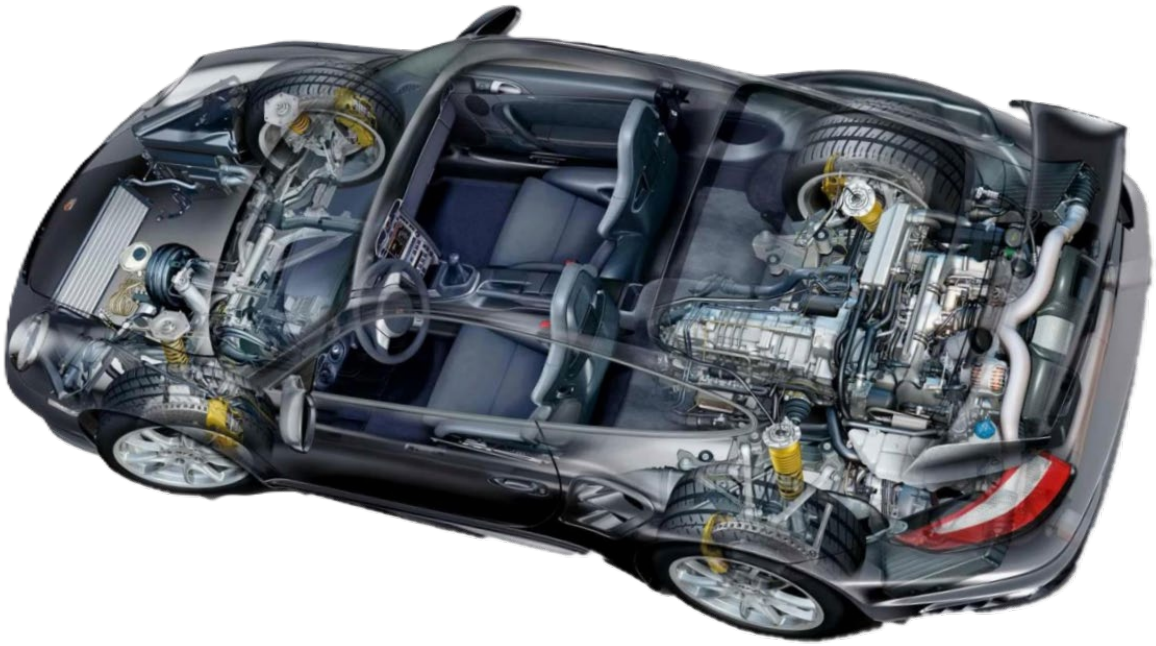
Mastering Embedded System Online Diploma

http://www.learn-in-depth-store.com



PRESSURE CONTROLLING SYSTEM

Abstract

A pressure controller system aimed to control and monitor a pressure within environment ,the client want to informs the crew of a cabin with an alarm when the pressure exceeds 20 bars in the cabin

Eng. Ayat Mohamed Mohamed

ayatmohamed2372002@gmail.com

GitHub Repo: https://github.com/Ayat237/Embedded_system_online_diploma.git

Table of Contents

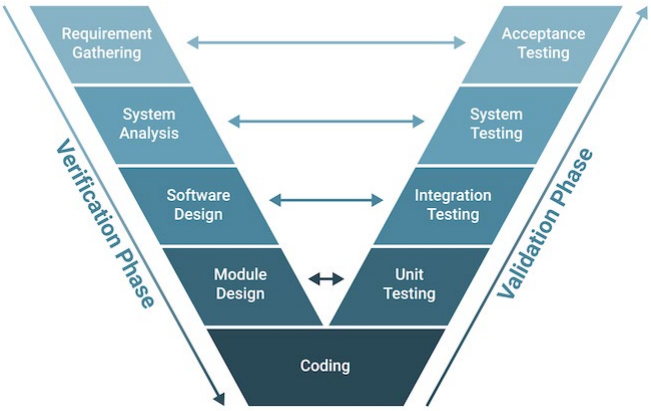
1. Case Study

- A ”client” expects you to deliver the software of the following system:
- 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.
  - After 60 seconds the alarm will stop.
- Pressure Controller Assumptions
  - The controller set up and shutdown 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

---

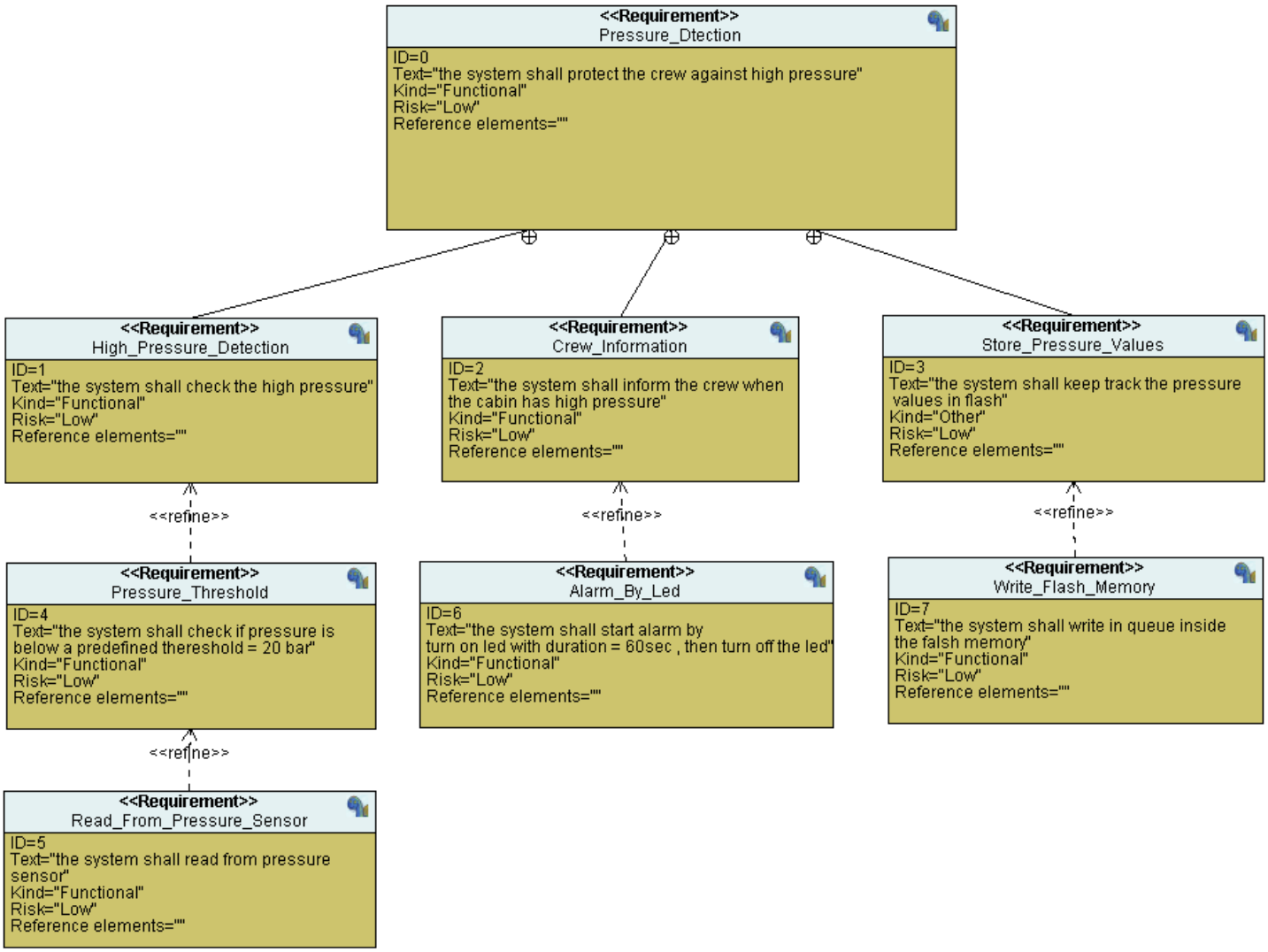
2. Method

- V-model-based development is used in this project



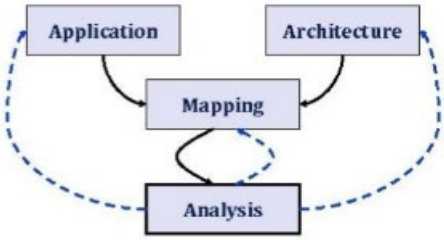
3. Requirements

- Based on case study and assumptions we will define all requirements in requirement diagram
- Where a requirements diagram is a visual representation of the requirements for a system, along with the relationships between those requirements and other elements in the system model.

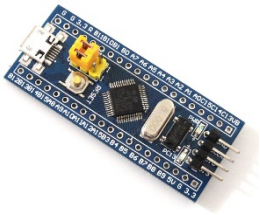


4. Space Exploration/Partitioning

- It is a way to fined the optimal solution
- provides a starting point for implementing the cabin pressure controller system



- Microcontroller Selection:
  - STM32F103 –> ARM Cortex-M3
  - Is a popular choice for embedded systems development
  - It meet all requirements needed for this project due to it has high-performance, real-time processing in cost-constrained applications and can handle complex tasks



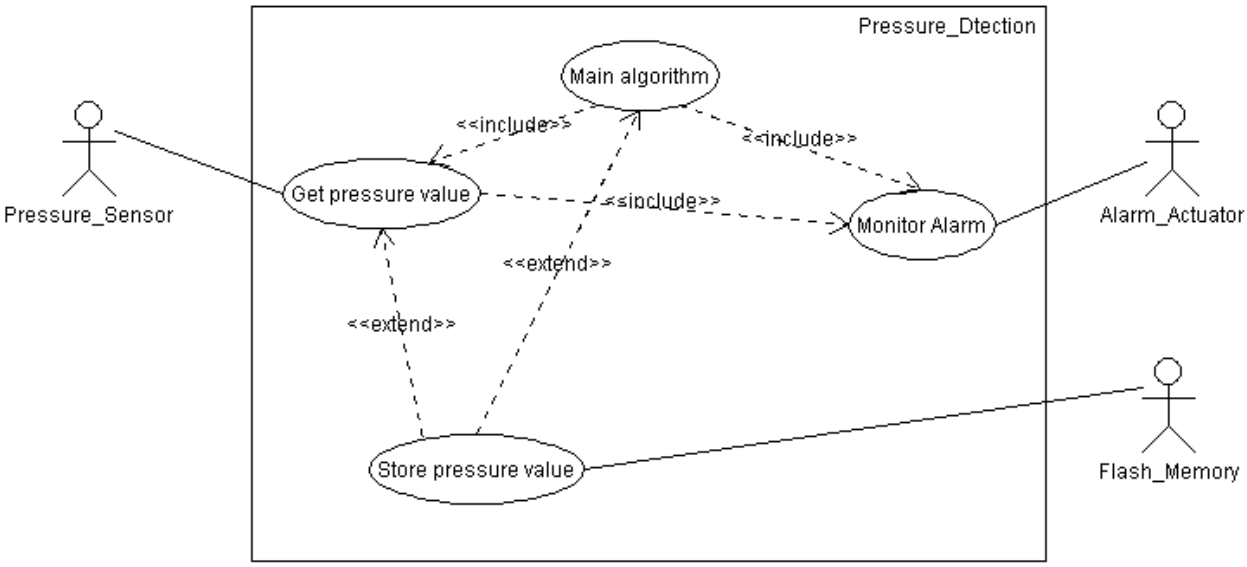
---

5. System Analysis

- In this stage we will understand the main functionalities of the system to be designed
- Analysis methods
  - System boundary and main functions → Use Case Diagram
  - Relations between main functions → Activity Diagram
  - Communications between main system entities and actors → Sequence Diagram

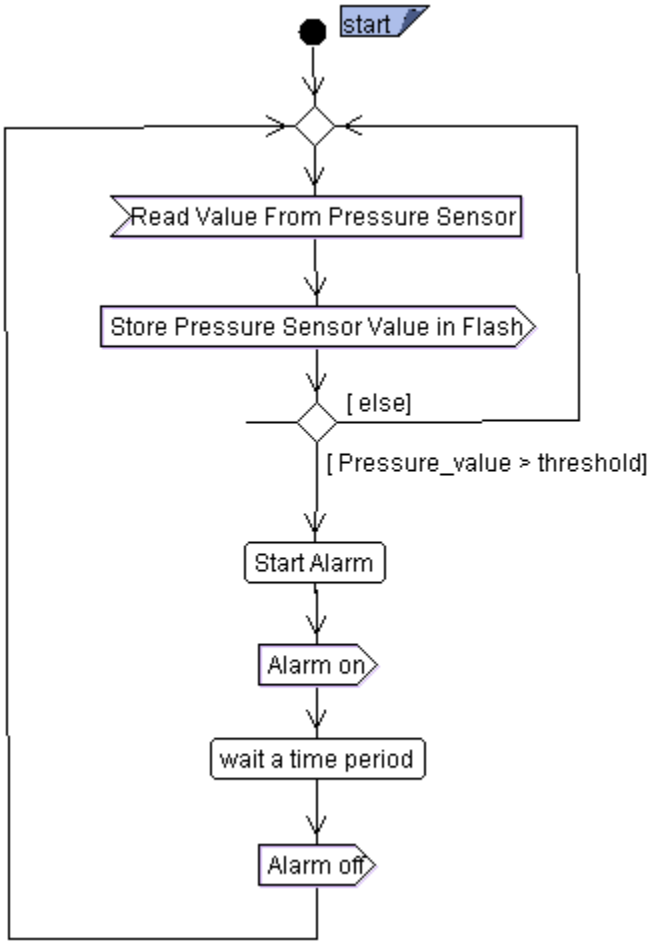
I. Use Case Diagram

- Shows what the system does and who uses it



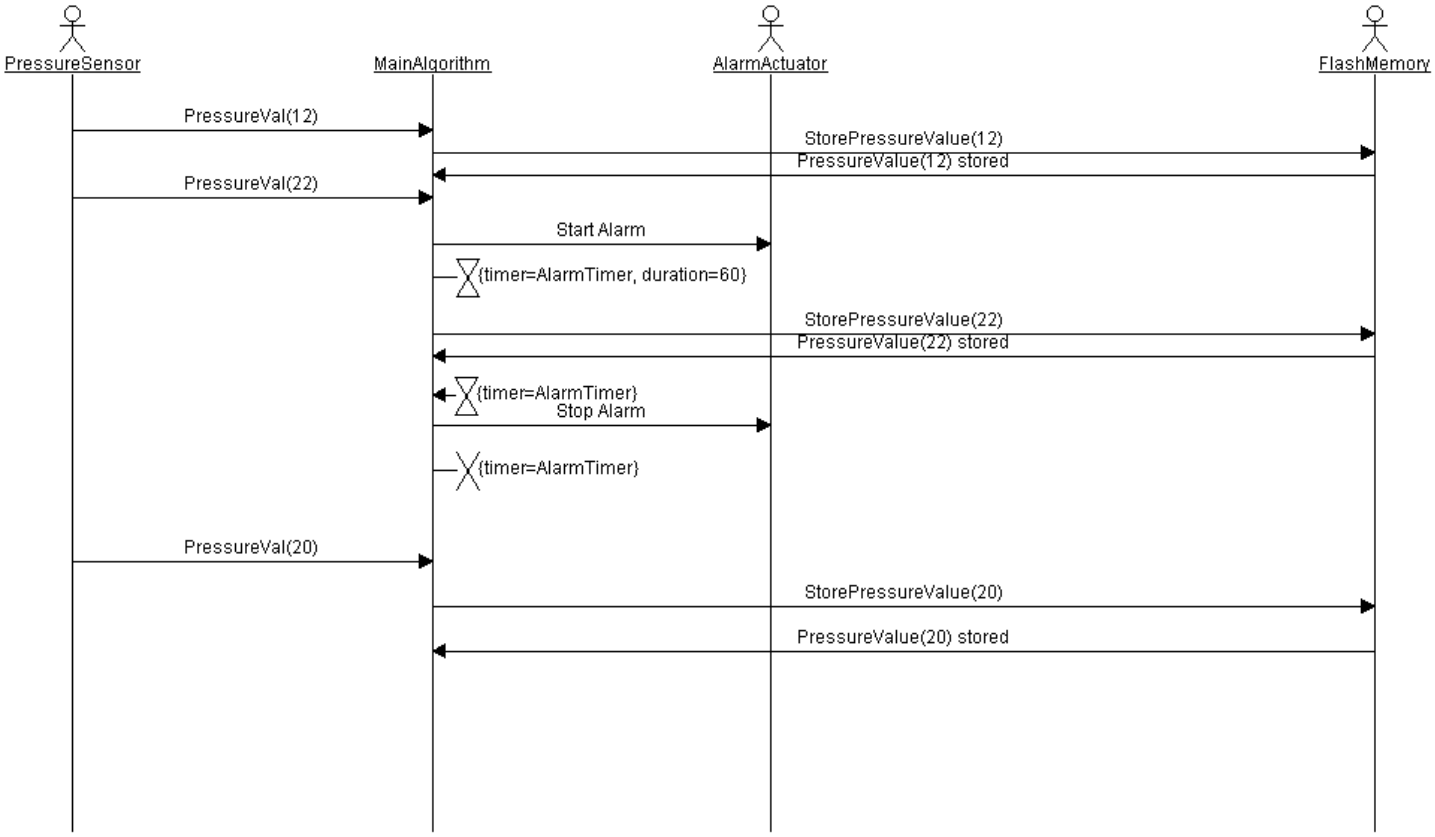
II. Activity Diagram

- Activity diagrams describe the workflow behavior of a system



III. Sequence Diagram

- An interaction diagram that details how operations are carried out.
- What messages are sent and when.
- Sequence diagrams are organized according to time.

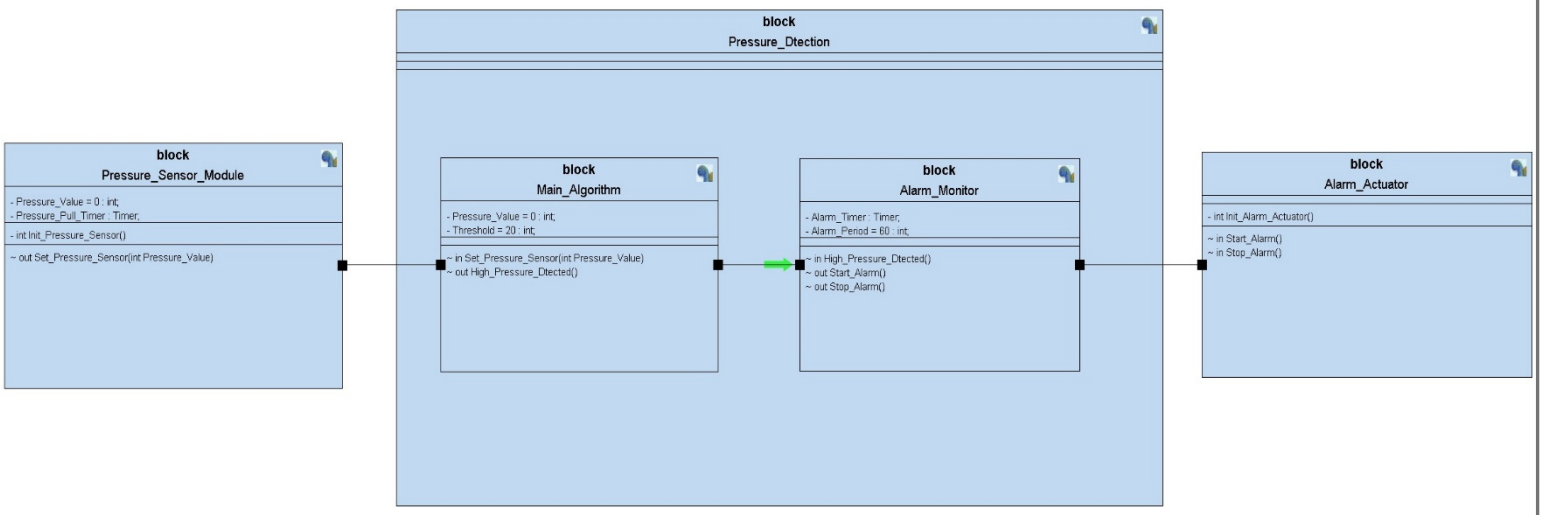


---

6. System Design

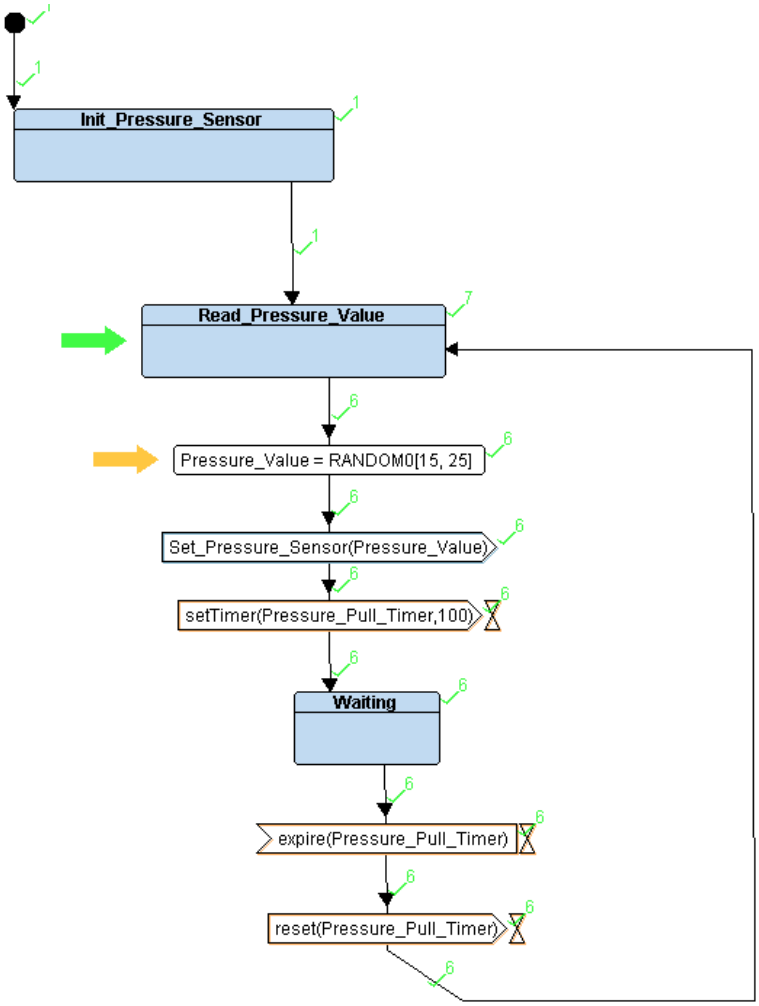
- Design is what a client wants by complies with the client requirements.
- Design Methods
  - System architecture → Block Definition Diagram and Internal Block Diagram
  - Behavior of the system → State Machine Diagram

I. Block Diagram

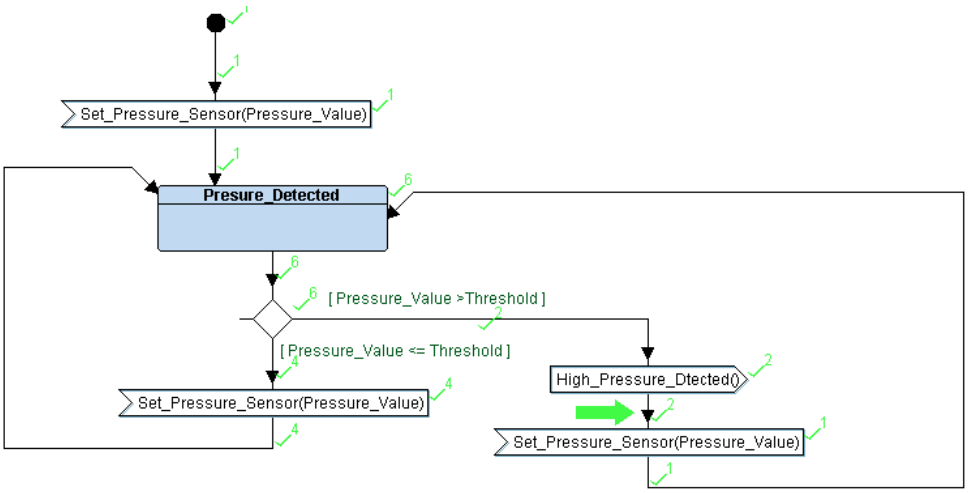


II. State Machine Diagram

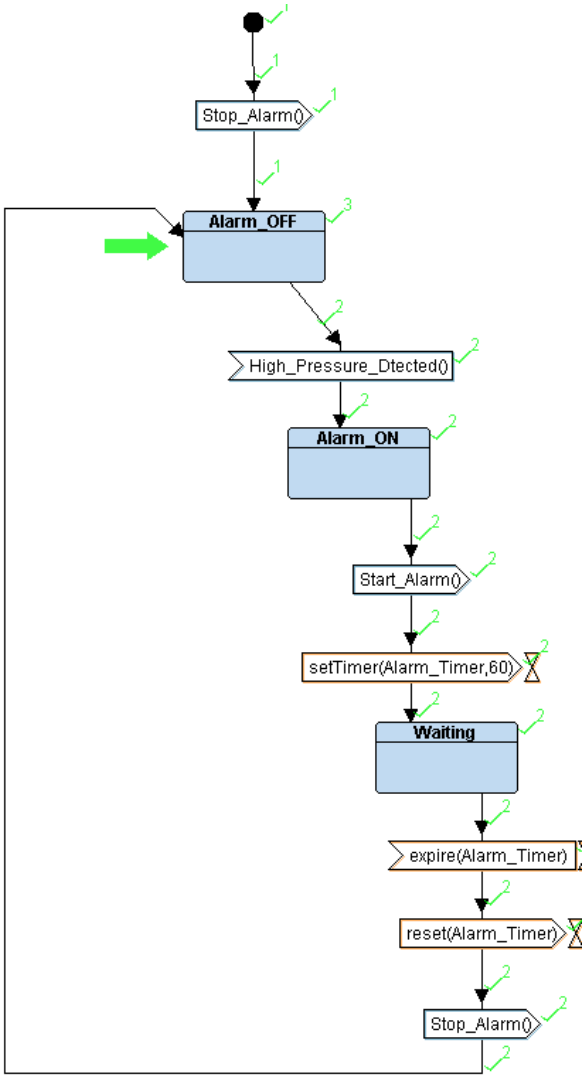
1) Pressure Sensor Module



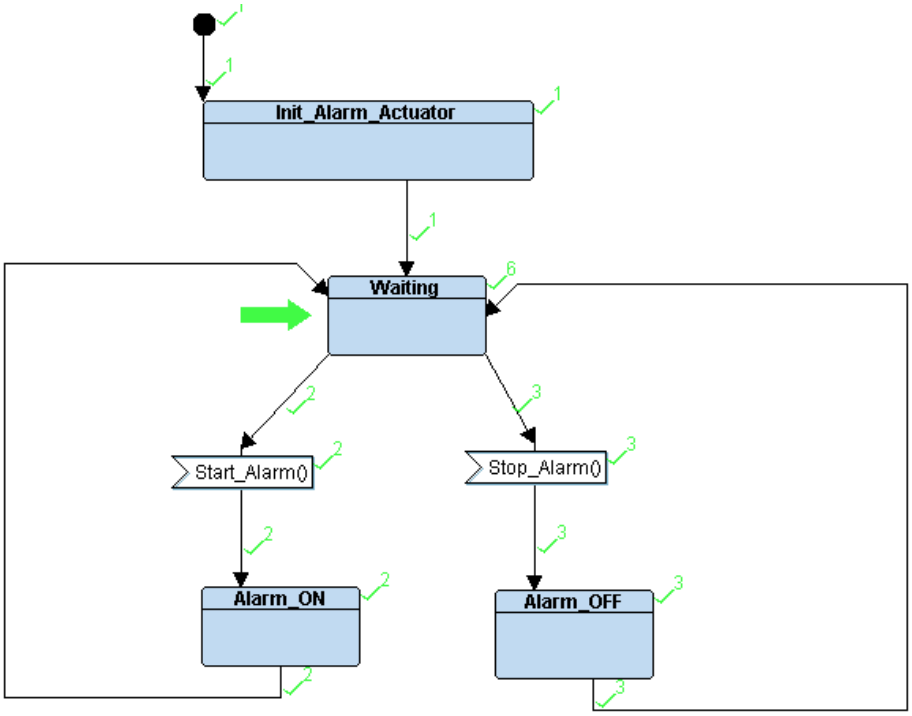
2) Main Algorithm



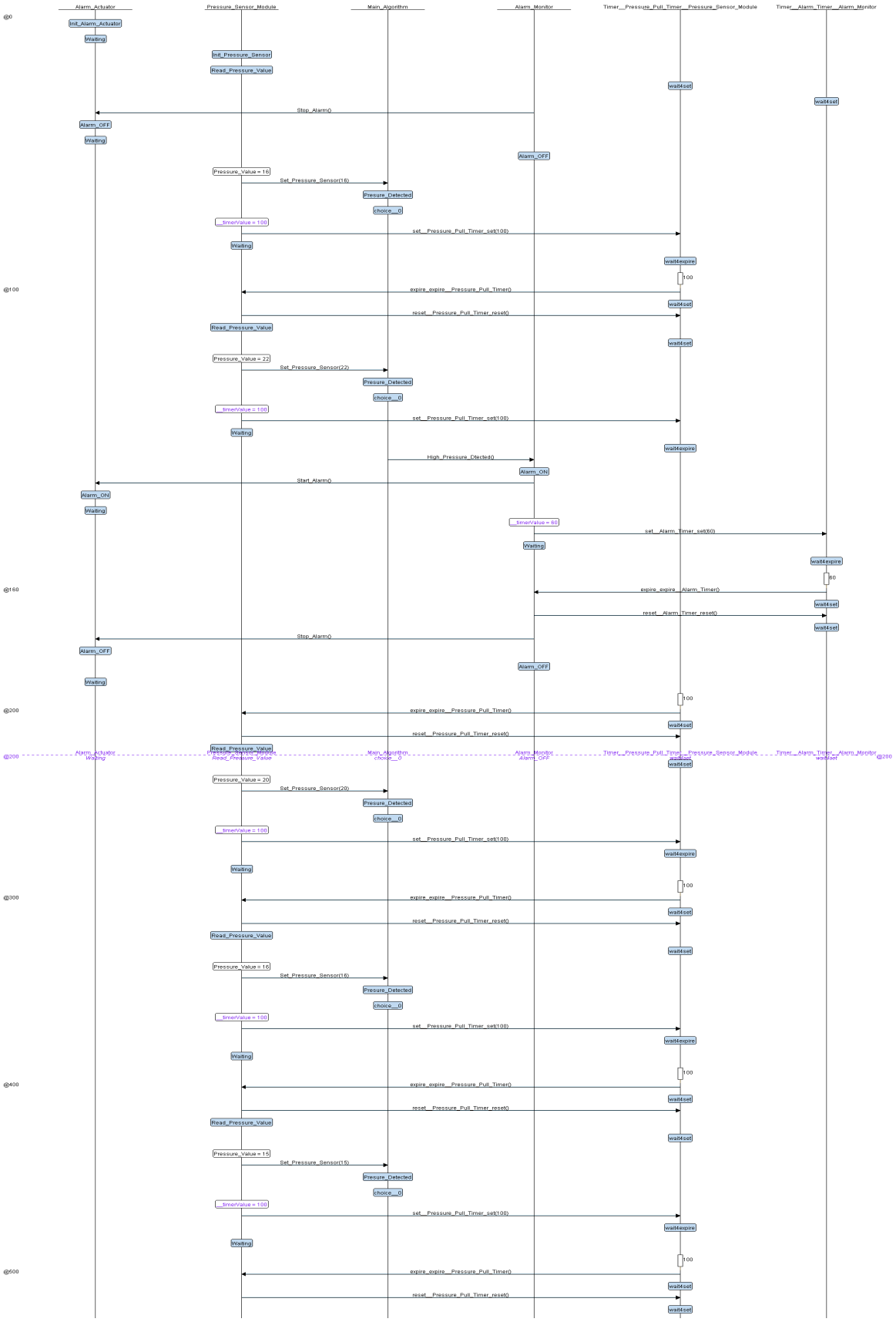
3) Alarm Monitor



4) Alarm Actuator



5) Simulation Of All State Diagrams



7. Files

I. C code running(log.txt)

```
PS_Read_Value : Pressure = 16 ----> PS_Waiting
.... Low Pressure Detected ....
.... Alarm Waiting ....
.... Alarm oFF ....
...PS Sensor Waiting....
.... Low Pressure Detected ....
.... Alarm stopped ....
.... Alarm oFF ....
PS_Read_Value : Pressure = 22 ----> PS_Waiting
.... High Pressure Detected ....
.... Alarm started ....
.... Alarm ON ....
...PS Sensor Waiting....
.... High Pressure Detected ....
.... Alarm started ....
.... Alarm ON ....
PS_Read_Value : Pressure = 20 ----> PS_Waiting
.... Low Pressure Detected ....
.... Alarm Waiting ....
.... Alarm oFF ....
...PS Sensor Waiting....
.... Low Pressure Detected ....
.... Alarm stopped ....
.... Alarm oFF ....
```

II. Symbol Table

```
20000014 B _E_bss
2000000c D _E_DATA
08000414 T _E_text
2000000c B _S_bss
20000000 D _S_DATA
20001014 B _stack_top
0800001c T AA_Init
20001014 B AA_State
20001018 B AA_State_et
20000000 D Alarm_Period
080000cc T AM_Init
20001020 B AM_State
2000101c B AM_State_et
08000384 W Bus_Fault
08000384 T Default_Handler
08000170 T Delay
08000190 T getPressureVal
080001e4 T GPIO_INITIALIZATION
08000384 W H_Fault_Handler
08000154 T High_Pressure_Dtected
2000000c B MA_Pressure_Value
20001024 B MA_State
20001028 B MA_State_et
08000234 T main
08000384 W MM_Fault_Handler
08000384 W NMI_Handler
20000008 D Pressure_Pull_Timer
08000310 T PS_Init
20000010 B PS_Pressure_Value
2000102c B PS_State
```

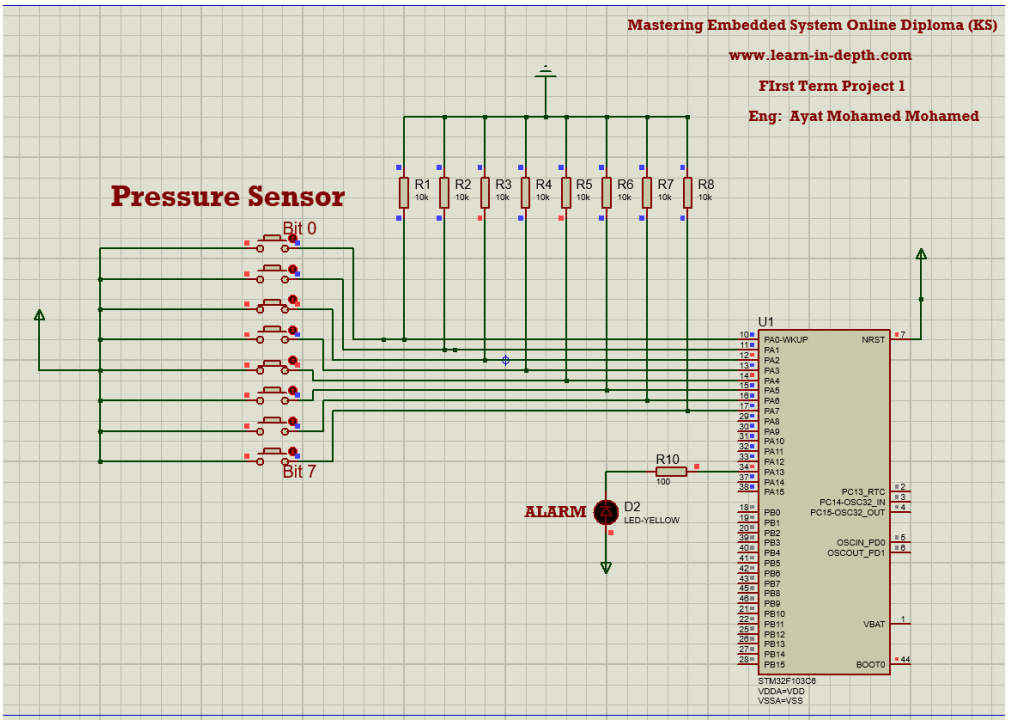
```
20001030 B PS_State_et
08000390 T Reset_Handler
080001a8 T Set_Alarm_actuator
080002f4 T Set_Pressure_Sensor
08000268 T setup
08000070 T ST_AA_Alarm_OFF
0800004c T ST_AA_Alarm_ON
08000028 T ST_AA_Waiting
080000d8 T ST_AM_Alarm_OFF
080000fc T ST_AM_Alarm_ON
08000120 T ST_AM_Waiting
080002b4 T ST_MA_Pressure_Detected
0800031c T ST_PS_Read_Value
08000354 T ST_PS_Waiting
08000094 T Start_Alarm
080000b0 T Stop_Alarm
20000004 D Threshold
08000384 W Usage_Fault_Handler
08000000 T vectors
```

III. Section.txt

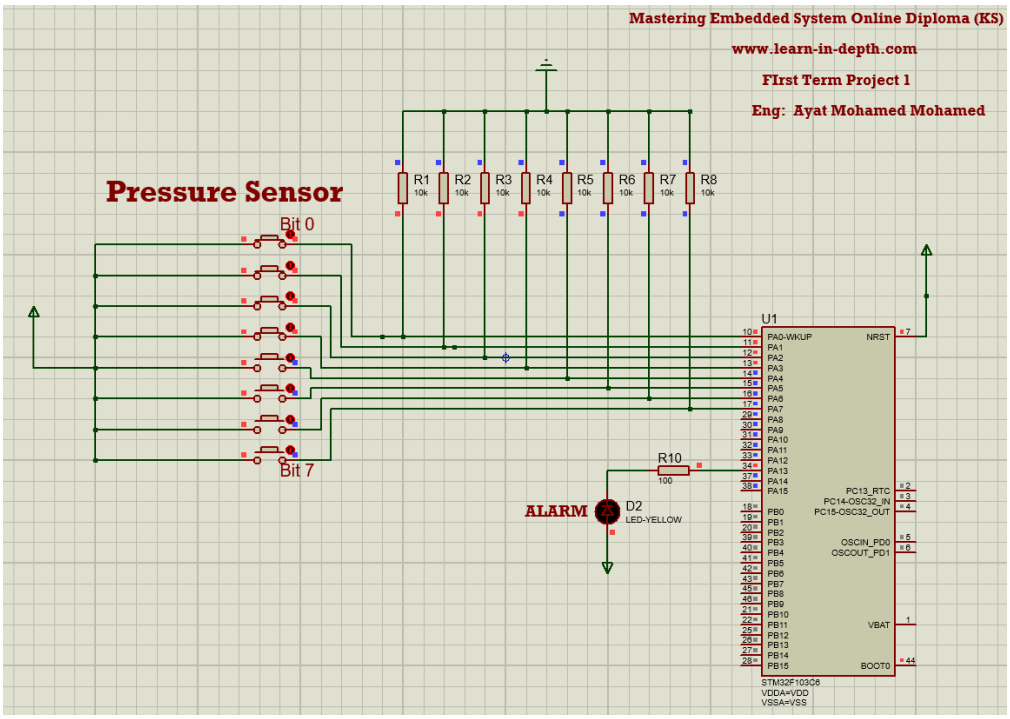
```
main.o:     file format elf32-littlearm

Sections:
Idx Name          Size      VMA       LMA       File off  Algn
  0 .text         00000080  00000000  00000000  00000034  2**2
                  CONTENTS, ALLOC, LOAD, RELOC, READONLY, CODE
  1 .data         00000000  00000000  00000000  000000b4  2**0
                  CONTENTS, ALLOC, LOAD, DATA
  2 .bss          00000000  00000000  00000000  000000b4  2**0
                  ALLOC
  3 .debug_info   000009d8  00000000  00000000  000000b4  2**0
                  CONTENTS, RELOC, READONLY, DEBUGGING
  4 .debug_abbrev 000001a1  00000000  00000000  00000a8c  2**0
                  CONTENTS, READONLY, DEBUGGING
  5 .debug_loc    00000058  00000000  00000000  00000c2d  2**0
                  CONTENTS, READONLY, DEBUGGING
  6 .debug_aranges 00000020  00000000  00000000  00000c85  2**0
                  CONTENTS, RELOC, READONLY, DEBUGGING
  7 .debug_line   0000014b  00000000  00000000  00000ca5  2**0
                  CONTENTS, RELOC, READONLY, DEBUGGING
  8 .debug_str    0000056e  00000000  00000000  00000df0  2**0
                  CONTENTS, READONLY, DEBUGGING
  9 .comment      0000007f  00000000  00000000  0000135e  2**0
                  CONTENTS, READONLY
 10 .debug_frame  00000048  00000000  00000000  000013e0  2**2
                  CONTENTS, RELOC, READONLY, DEBUGGING
 11 .ARM.attributes 00000033  00000000  00000000  00001428  2**0
                  CONTENTS, READONLY
```

8. Proteus Simulation

1. Pressure value = 20



2. Pressure value = 15



3. Pressure value = 25