

Mastering Embedded System Online Diploma

www.learn-in-depth.com

First Term (Final Project 1)

Eng. Mahmoud Ashraf

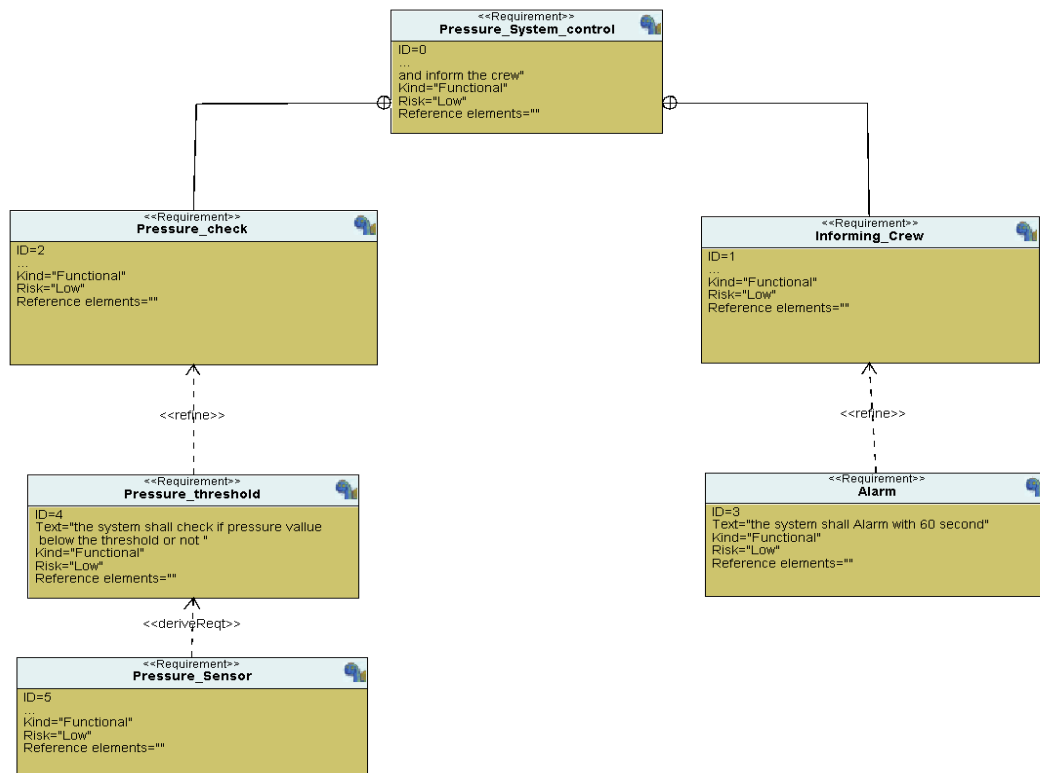
Case Study:

- The monitoring way to detect the pressure will be alarm with LEDs.
- Pressure limit is 20 bar. if exceed the limits alarm goes on while 60 second.

Assumption:

- Controller set up and shutdown procedures are not modeled
- Controller maintenance is not modeled
- Pressure sensor used never fails
- Alarm never fails
- No power cut

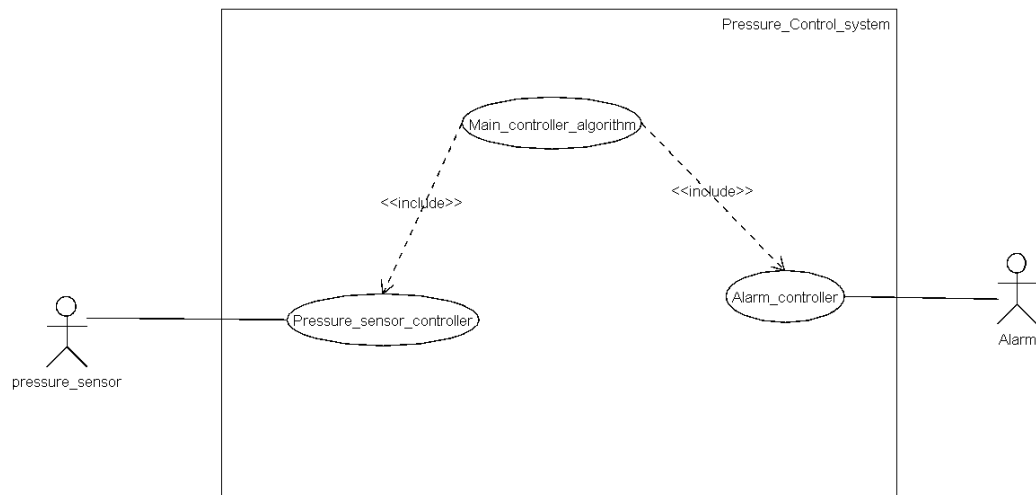
System Requirement Diagram:



Space Exploration:

This project needs one ECU, which will be **STM32**.

System Analysis:

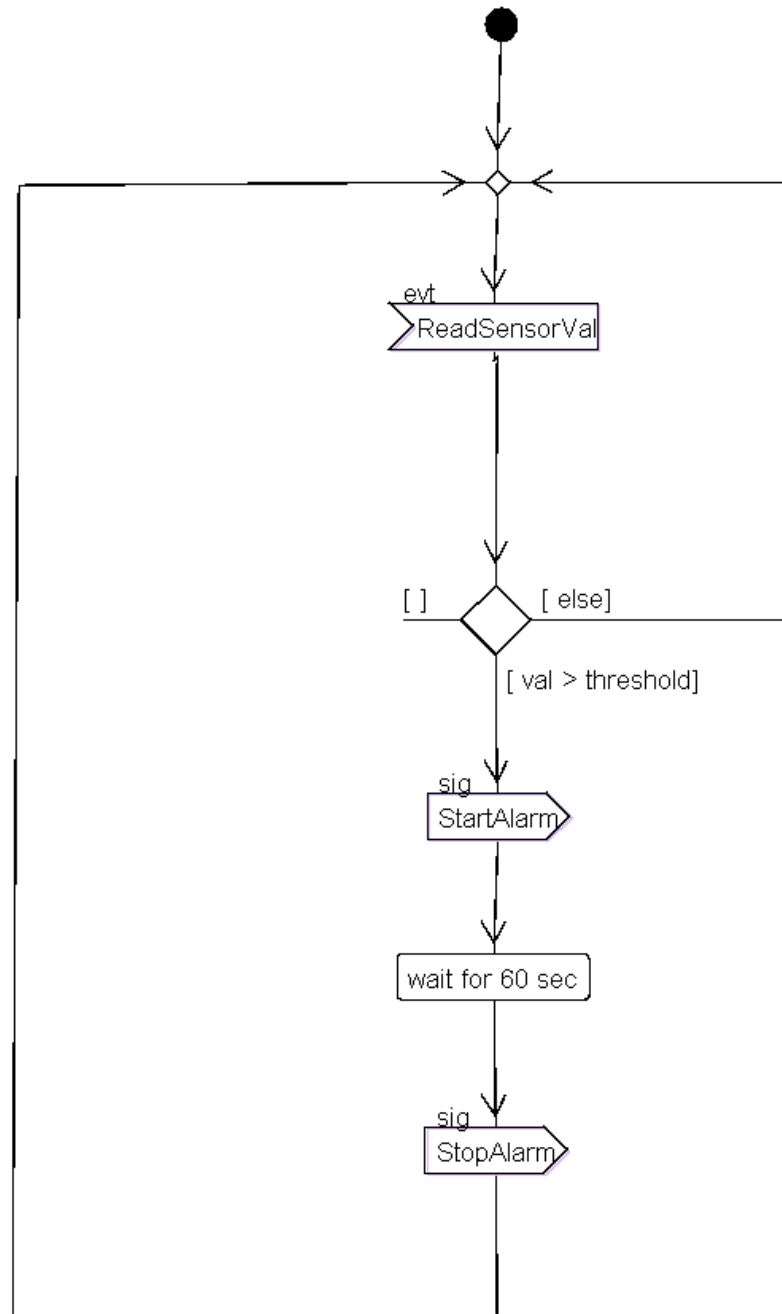


Main controller algorithm: compare the value of sensor with threshold (20 bar) and take the action.

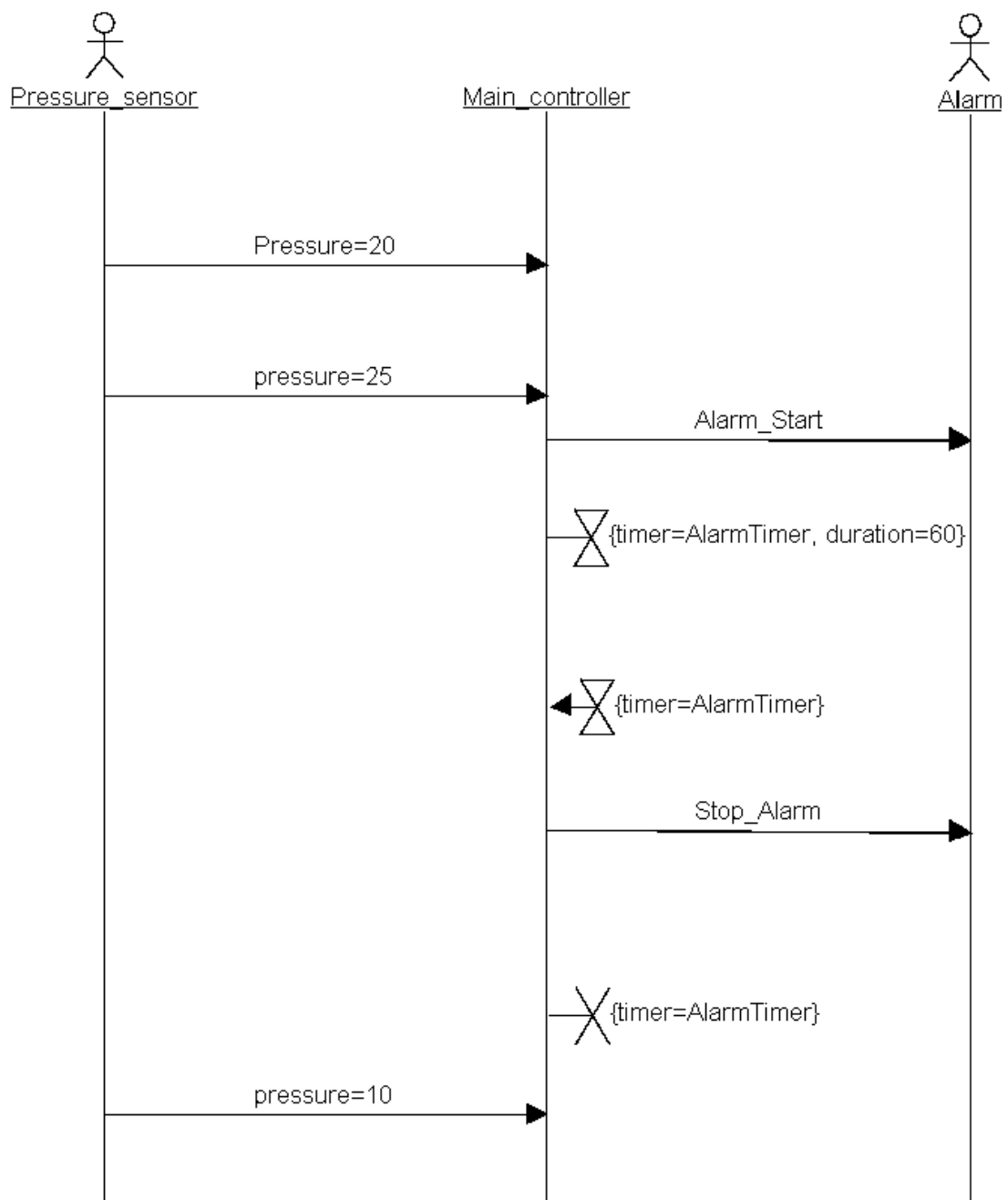
Pressure sensor: get reading of the sensor.

Alarm controller: control the alarm for 60 second when high pressure detected.

Activity Diagram:

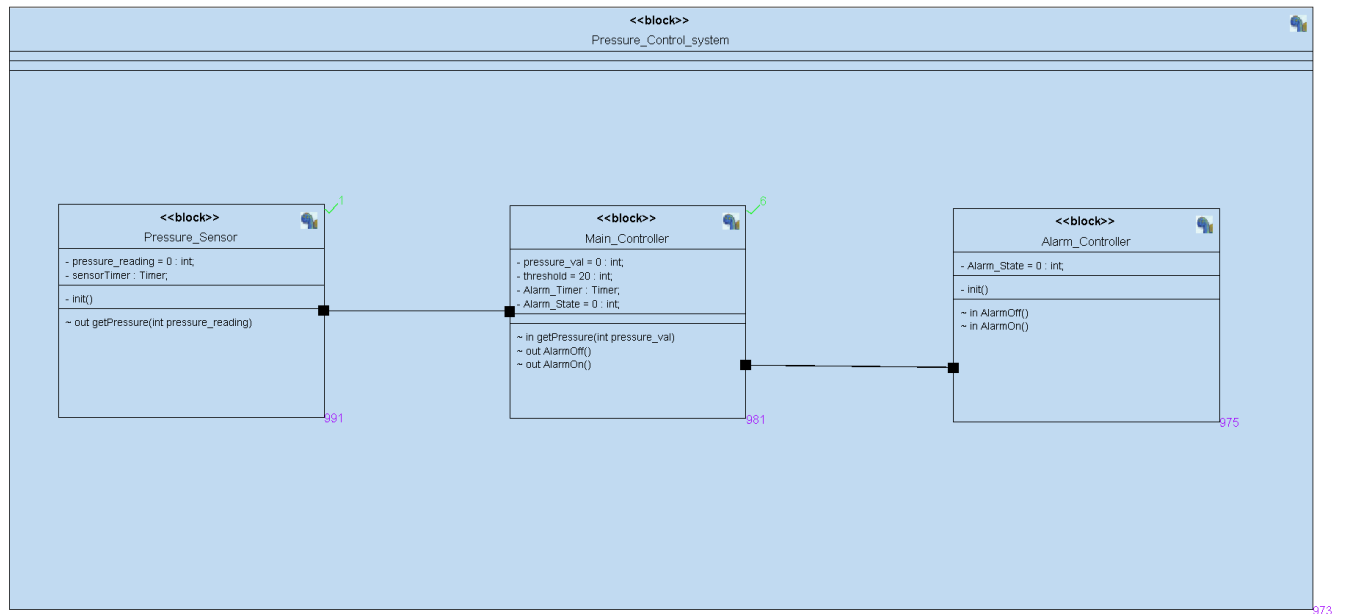


Sequence Diagram:

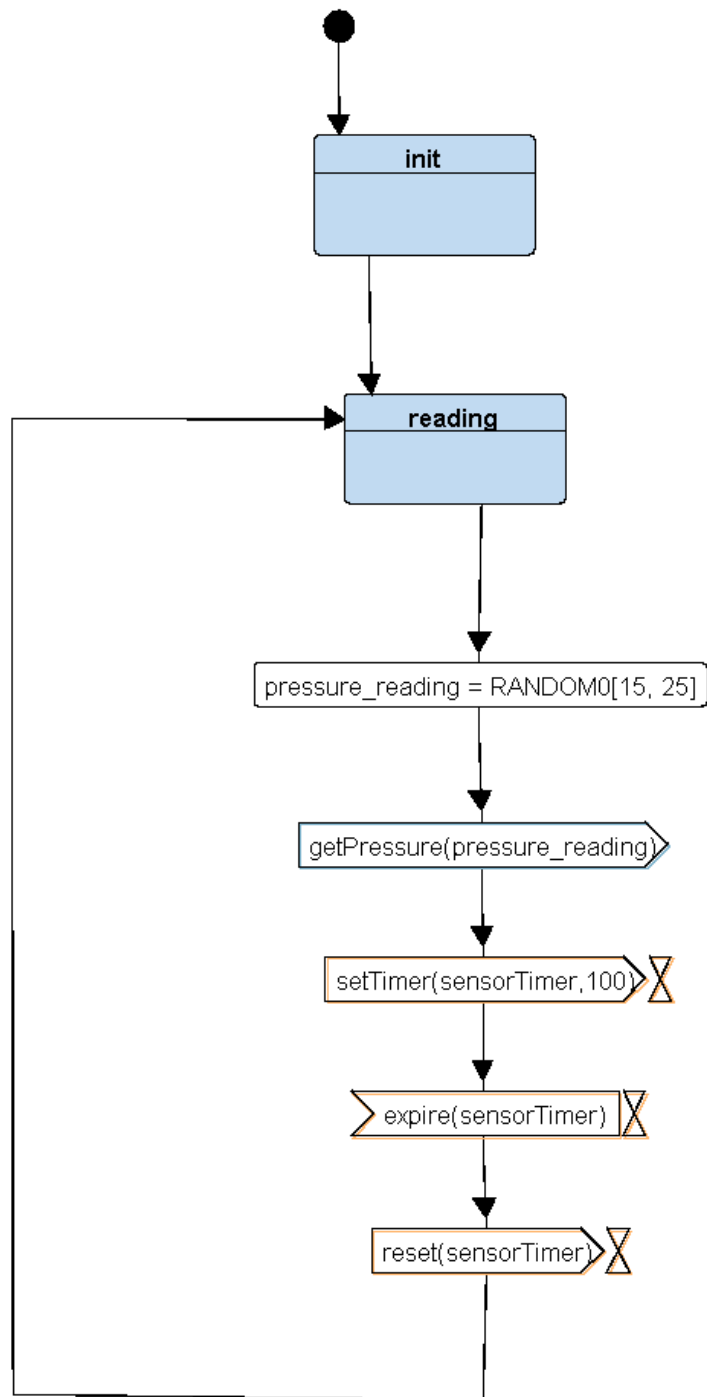


System Design:

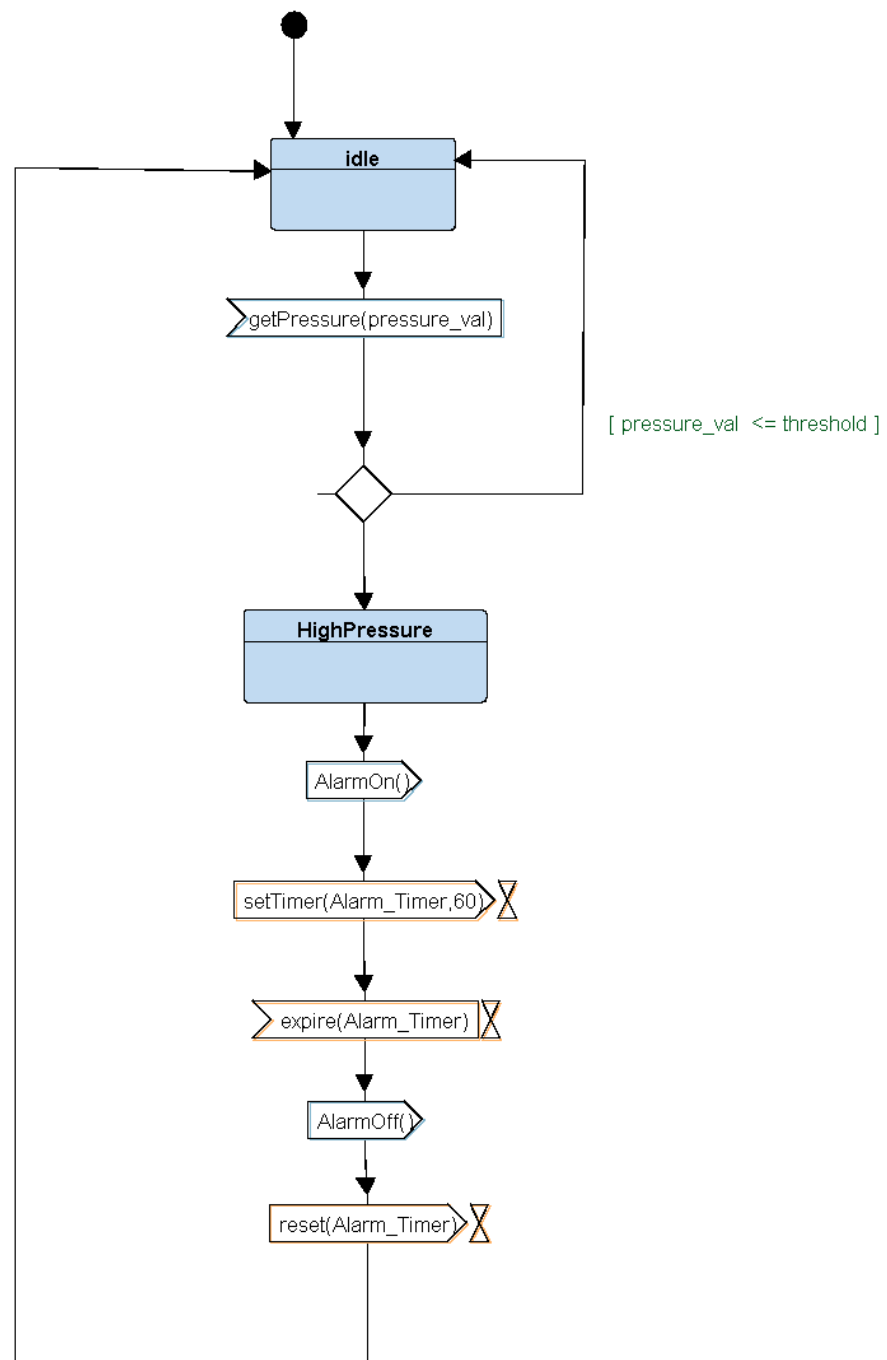
Block Diagram:



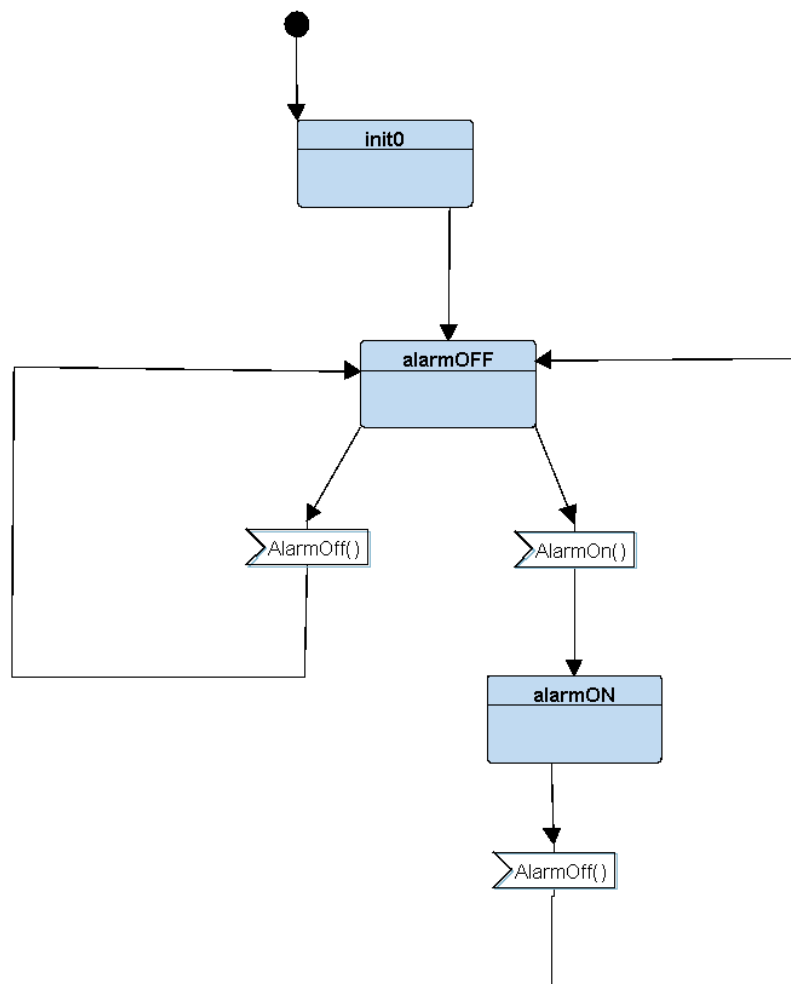
State Machine Pressure Sensor:



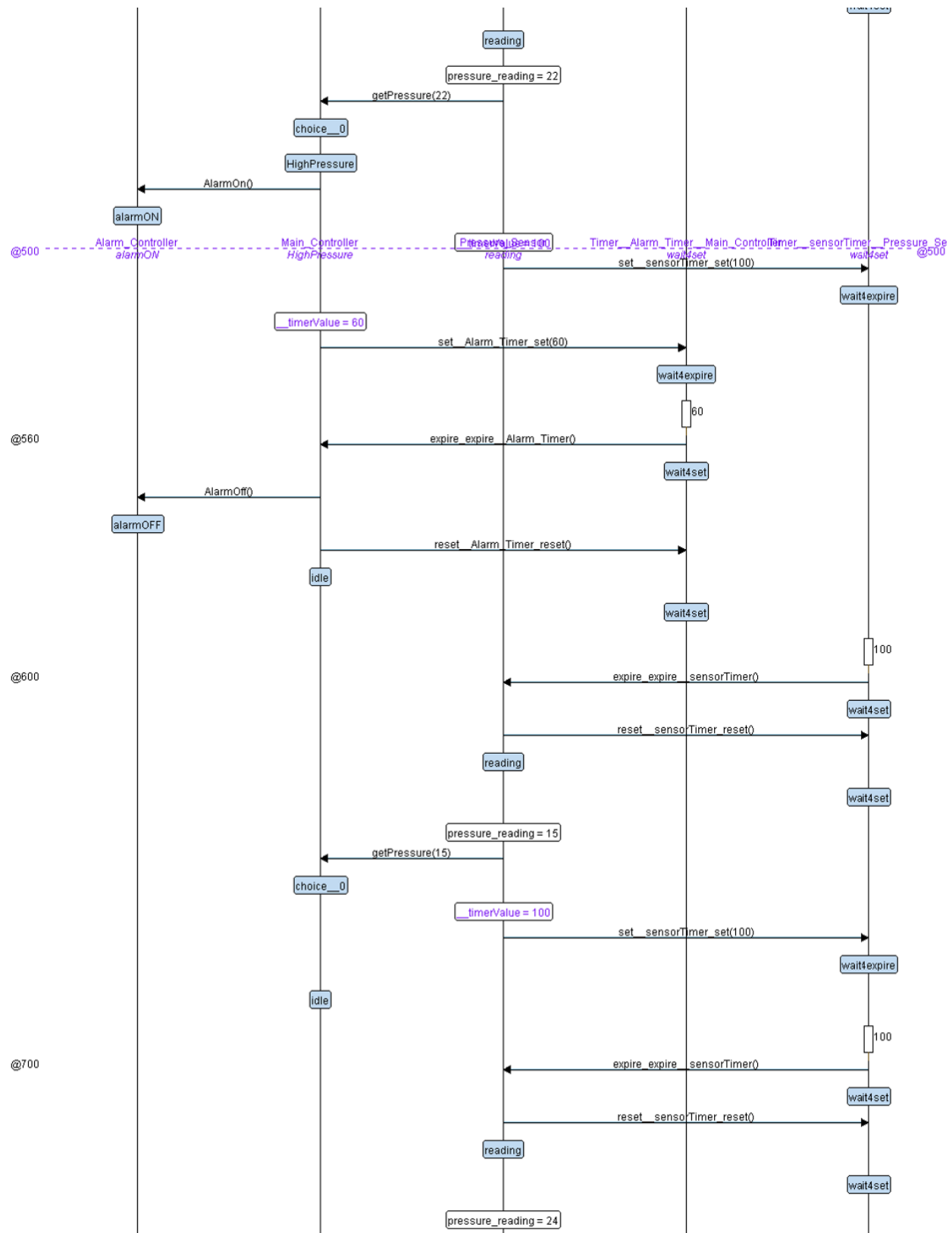
State Machine Main Controller:



State Machine Alarm Controller:



Simulation:



System Implementation:

CM3 Source Code - U1

```

1  //-----
2  // @author (Mahmoud Ashraf)
3  // @email (M.ashraf_g02@gmail.com)
4  // @create date 2022-09-17 21:34:07
5  // @modify date 2022-09-17 21:34:07
6  // @desc [description]
7  //-----
8  #include <stdint.h>
9  #include "main.h"
10
11 void setup_system(void)
12 {
13     GPIO_INITIALIZATION();
14     Set_Alarm_actuator(Alarm_OFF);
15     pSensor_state = STATE(reading);
16     pAlarmState = STATE(Alarm_OFF);
17     pcontroller_state = STATE(idle);
18 }
19
20 int main()
21 {
22     setup_system();
23     while (1)
24     {
25         pSensor_state = STATE(reading);
26         pcontroller_state();
27         pAlarmState();
28     }
29     return 0;
30 }

```

CM3 PORT0 at...

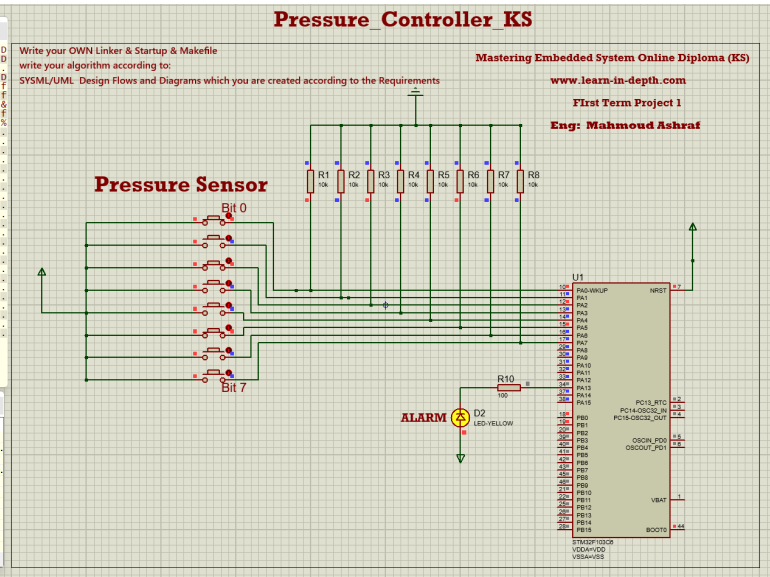
```

40010800 68 D
40010801 68 D
40010802 4
40010803 68 D
40010804 102 F
40010805 102 F
40010806 18 4
40010807 102 F
40010808 37
40010809 0
4001080A 0
4001080B 0
4001080C 0
4001080D 0
4001080E 0
4001080F 0
40010810 0
40010811 0
40010812 0
40010813 0
40010814 0
40010815 0
40010816 0
40010817 0
40010818 0
40010819 0
4001081A 0
4001081B 0
4001081C 0
4001081D 0
4001081E 0
4001081F 0

```

CM3 Variables - U1

Name	Address	Type	Value
Alarm_State_ID	20001004	enum <unnamed>	alarm_ON (0)
Controller_State_ID	2000100C	enum <unnamed>	highPressure.
Alarm_State_ID	20001004	enum <unnamed>	alarm_ON (0)
Controller_State_ID	2000100C	enum <unnamed>	highPressure.
pSensorState_ID	20001014	enum <unnamed>	reading (0)
pSensorState_ID	20001014	enum <unnamed>	reading (0)
pressure_val	20000000	byte	%
Vectors	08000000	dword[6]	dword[6]
Alarm_State_ID	20001004	enum <unnamed>	alarm_ON (0)



CM3 Source Code - U1

```

1  //-----
2  // @author (Mahmoud Ashraf)
3  // @email (M.ashraf_g02@gmail.com)
4  // @create date 2022-09-17 21:34:07
5  // @modify date 2022-09-17 21:34:07
6  // @desc [description]
7  //-----
8  #include <stdint.h>
9  #include "main.h"
10
11 void setup_system(void)
12 {
13     GPIO_INITIALIZATION();
14     Set_Alarm_actuator(Alarm_OFF);
15     pSensor_state = STATE(reading);
16     pAlarmState = STATE(Alarm_OFF);
17     pcontroller_state = STATE(idle);
18 }
19
20 int main()
21 {
22     setup_system();
23     while (1)
24     {
25         pSensor_state = STATE(reading);
26         pcontroller_state();
27         pAlarmState();
28     }
29     return 0;
30 }

```

CM3 PORT0 at...

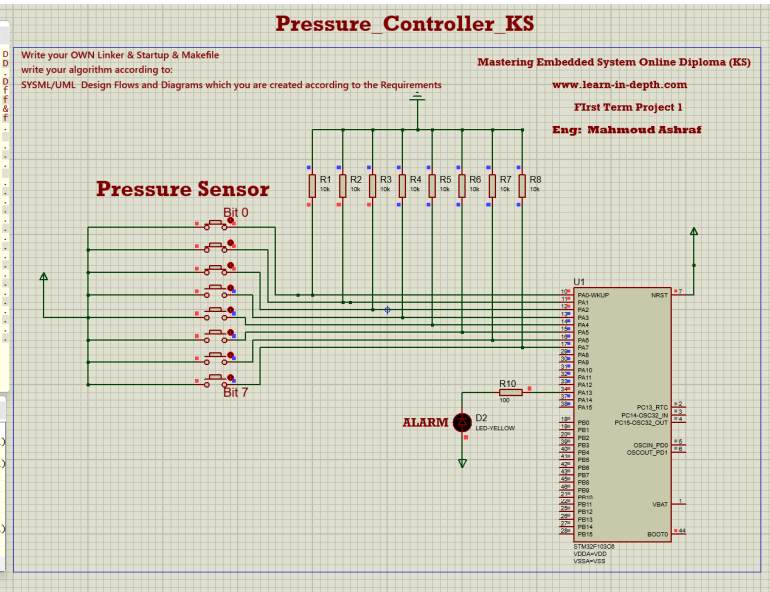
```

40010800 68 D
40010801 68 D
40010802 4
40010803 68 D
40010804 102 F
40010805 102 F
40010806 18 4
40010807 102 F
40010808 37
40010809 32
4001080A 0
4001080B 0
4001080C 0
4001080D 32
4001080E 0
4001080F 0
40010810 0
40010811 0
40010812 0
40010813 0
40010814 0
40010815 0
40010816 0
40010817 0
40010818 0
40010819 0
4001081A 0
4001081B 0
4001081C 0
4001081D 0
4001081E 0
4001081F 0

```

CM3 Variables - U1

Name	Address	Type	Value
Alarm_State_ID	20001004	enum <unnamed>	alarm_OFF (1)
Controller_State_ID	2000100C	enum <unnamed>	idle (0)
Alarm_State_ID	20001004	enum <unnamed>	alarm_OFF (1)
Controller_State_ID	2000100C	enum <unnamed>	idle (0)
pSensorState_ID	20001014	enum <unnamed>	reading (0)
pSensorState_ID	20001014	enum <unnamed>	reading (0)
pressure_val	20000000	byte	0x07
Vectors	08000000	dword[6]	dword[6]
Alarm_State_ID	20001004	enum <unnamed>	alarm_OFF (1)



Symbols:

```
$ arm-none-eabi-nm.exe Pressure_Control_system.elf
20000000 B _E_BSS
20000000 T _E_DATA
080003a0 T _E_TEXT
20000000 B _S_BSS
20000000 T _S_DATA
20002000 B _STACK_TOP
20002000 B Alarm_State_ID
080002e0 W Bus_fault_Handler
20002008 B Controller_State_ID
080002e0 T Default_Handler
08000114 T Delay
08000138 T getPressureVal
080001a0 T GPIO_INITIALIZATION
080002e0 W H_fault_Handler
08000268 T main
080002e0 W MM_fault_Handler
20002004 B pAlarmState
2000200c B pcontroller_state
20002014 B ppSensor_state
0800009c T pressure_action
20002010 B pressure_val
20002011 B pSensorState_ID
080002ec T Reset_Handler
08000150 T Set_Alarm_actuator
08000018 T SetAlarm
08000220 T setup_system
08000080 T ST_alarm_OFF
08000058 T ST_alarm_ON
080000f8 T ST_highPressure
080000dc T ST_idle
08000298 T ST_reading
080002e0 W Usage_fault_Handler
08000000 D vectors
```

Obj Dump:

```
$ arm-none-eabi-objdump.exe -h Pressure_Control_system.elf

Pressure_Control_system.elf:      file format elf32-littlearm

Sections:
Idx Name          Size      VMA       LMA       File off  Algn
  0 .vectors        00000018  08000000  08000000  00008000  2**2
    CONTENTS, ALLOC, LOAD, DATA
  1 .text           00000388  08000018  08000018  00008018  2**2
    CONTENTS, ALLOC, LOAD, READONLY, CODE
  2 .bss            00002018  20000000  080003a0  00010000  2**2
    ALLOC
  3 .debug_info      000006c7  00000000  00000000  000083a0  2**0
    CONTENTS, READONLY, DEBUGGING
  4 .debug_abbrev    000003d5  00000000  00000000  00008a67  2**0
    CONTENTS, READONLY, DEBUGGING
  5 .debug_loc       000002d0  00000000  00000000  00008e3c  2**0
    CONTENTS, READONLY, DEBUGGING
  6 .debug_aranges   000000c0  00000000  00000000  0000910c  2**0
    CONTENTS, READONLY, DEBUGGING
  7 .debug_line      00000398  00000000  00000000  000091cc  2**0
    CONTENTS, READONLY, DEBUGGING
  8 .debug_str       000002de  00000000  00000000  00009564  2**0
    CONTENTS, READONLY, DEBUGGING
  9 .comment         00000011  00000000  00000000  00009842  2**0
    CONTENTS, READONLY
10 .ARM.attributes  00000033  00000000  00000000  00009853  2**0
    CONTENTS, READONLY
11 .debug_frame      00000200  00000000  00000000  00009888  2**2
    CONTENTS, READONLY, DEBUGGING
```

Source Code:

- Main.c /Main.h
- Alarm.c/Alarm.h
- Sensor.c/Sensor.h
- Make File
- Startup.c
- LinkerScript.ld
- Map File

Uploaded to GitHub Repo:

https://github.com/Mahmoudg02/Learn_in_Depth.git