Mastering Embedded System Online

Diploma

Pressure Detection Controller

Name: Alaa Raed

GitHub: Alaa Raed (GitHub)

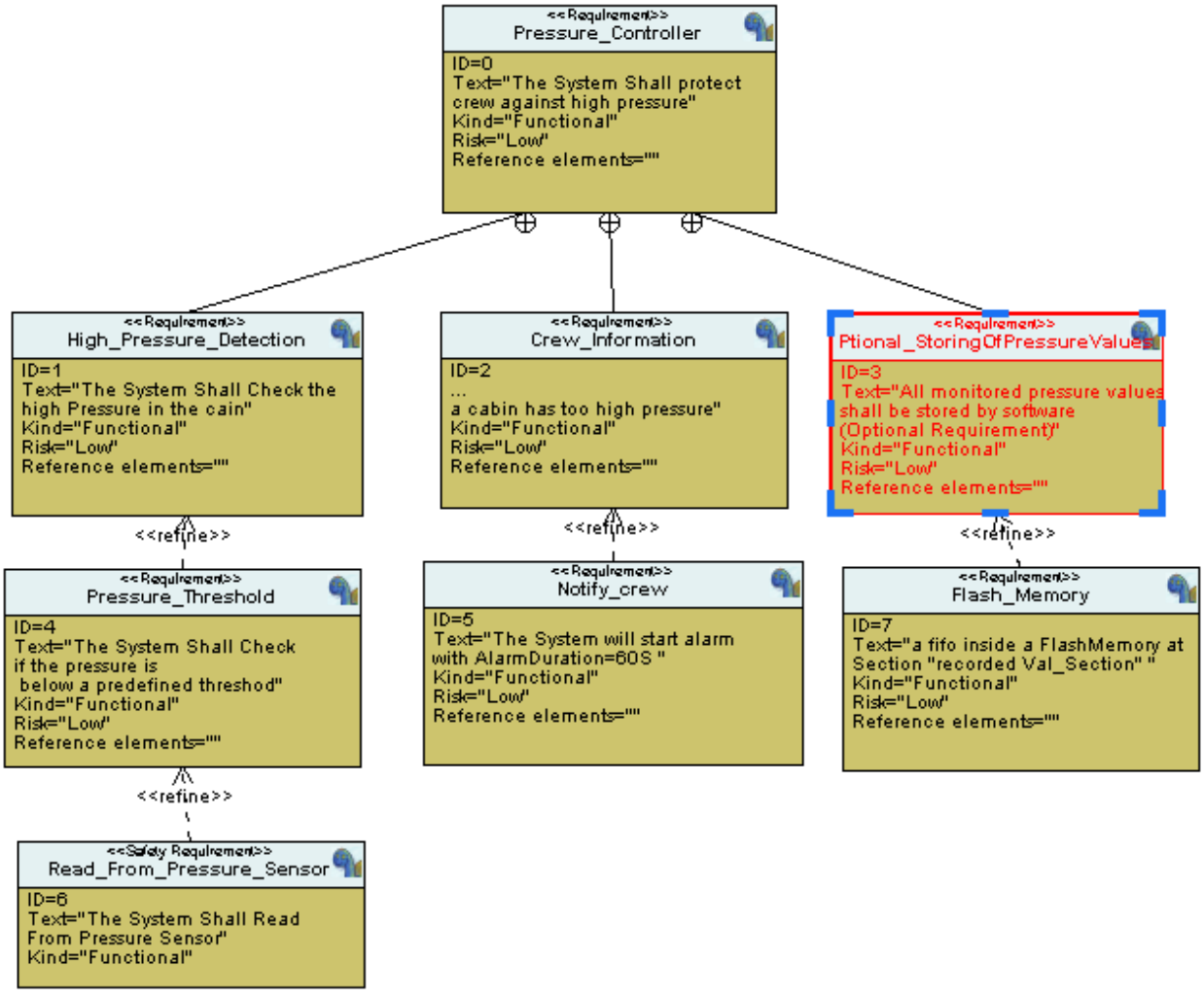
Profile: www.learn-in-depth-store.com

Table Of Content:

1-Client Requirements:

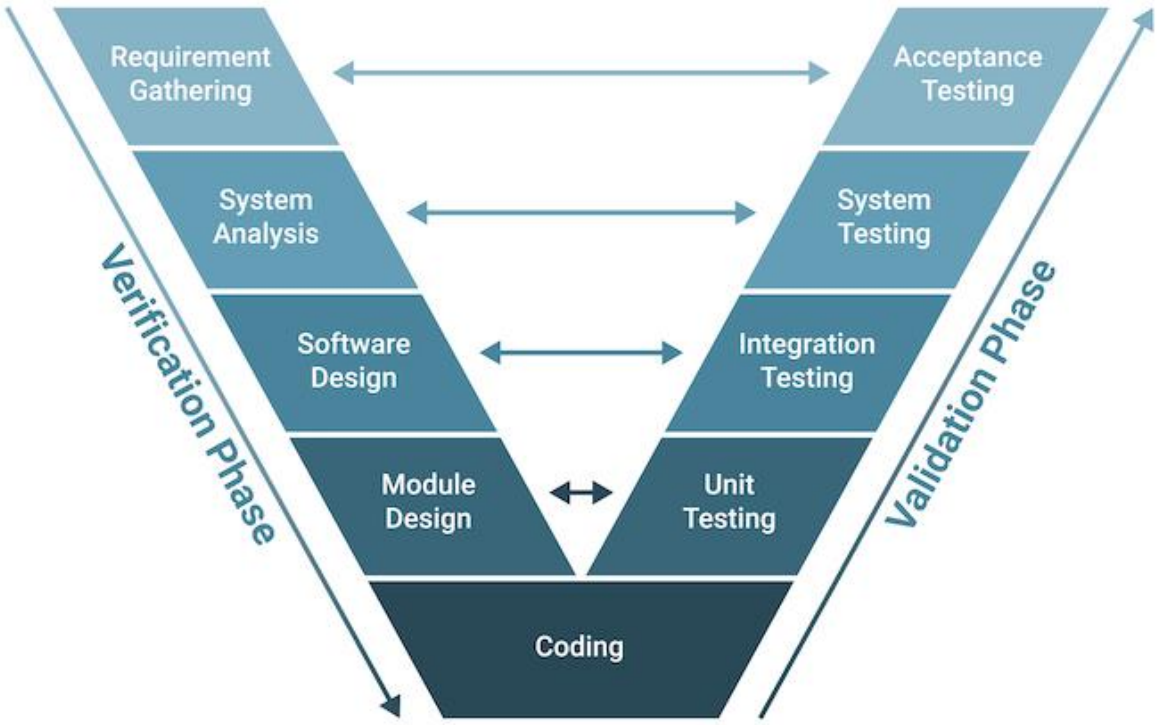
- **A pressure controller informs the crew of a cabin with an alarm when the pressure exceeds 20 bars in the cabin.**
- **The alarm duration equals 60 seconds.**

1.1 Requirements:



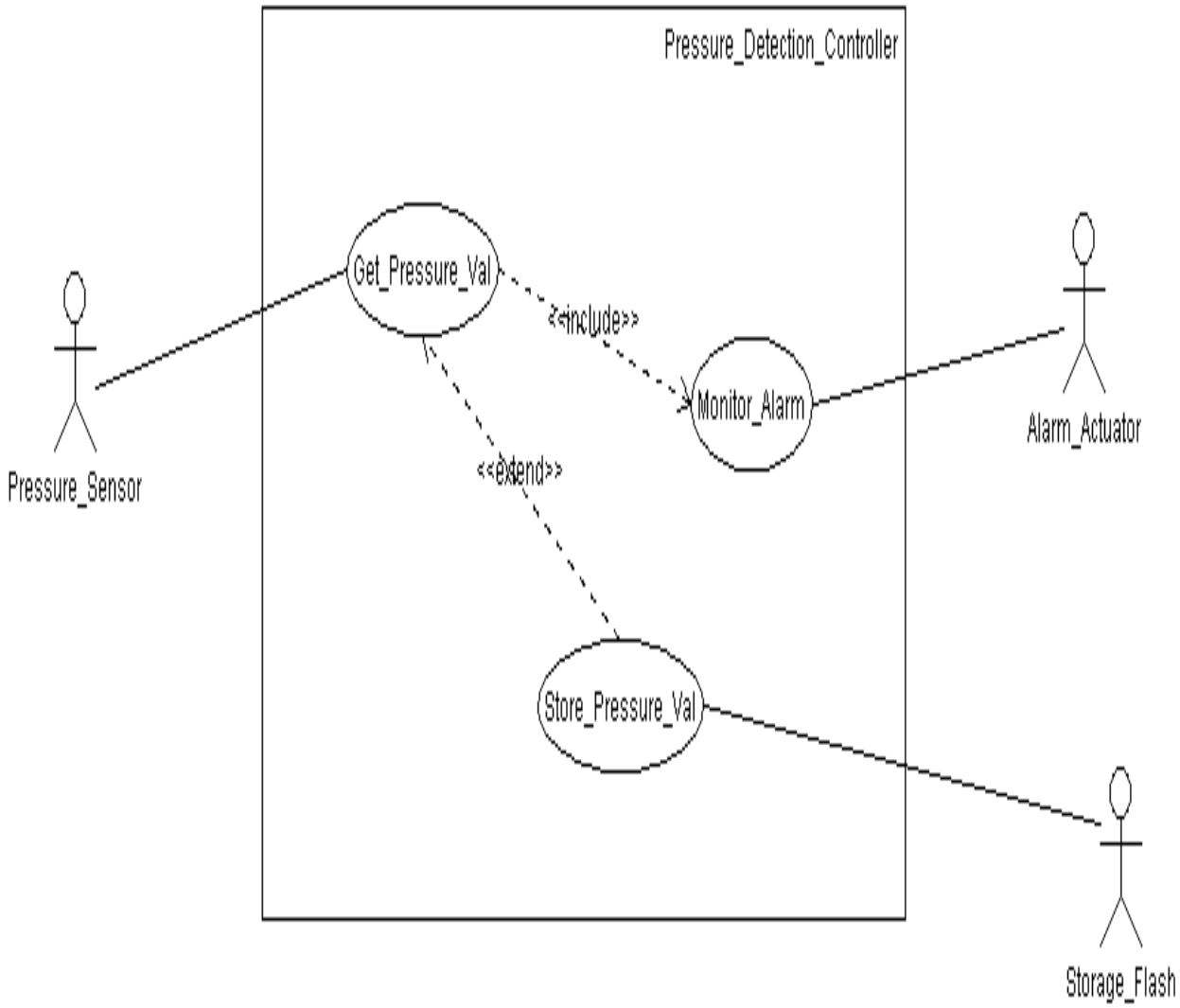
2-Methdology:

The V-model is named after its shape, which resembles the letter “V.” In the V-model, we divide the software development life cycle into phases and each phase is associated with a corresponding testing phase. The left-hand side of the V represents the verification phase while the right-hand side represents the validation phase.

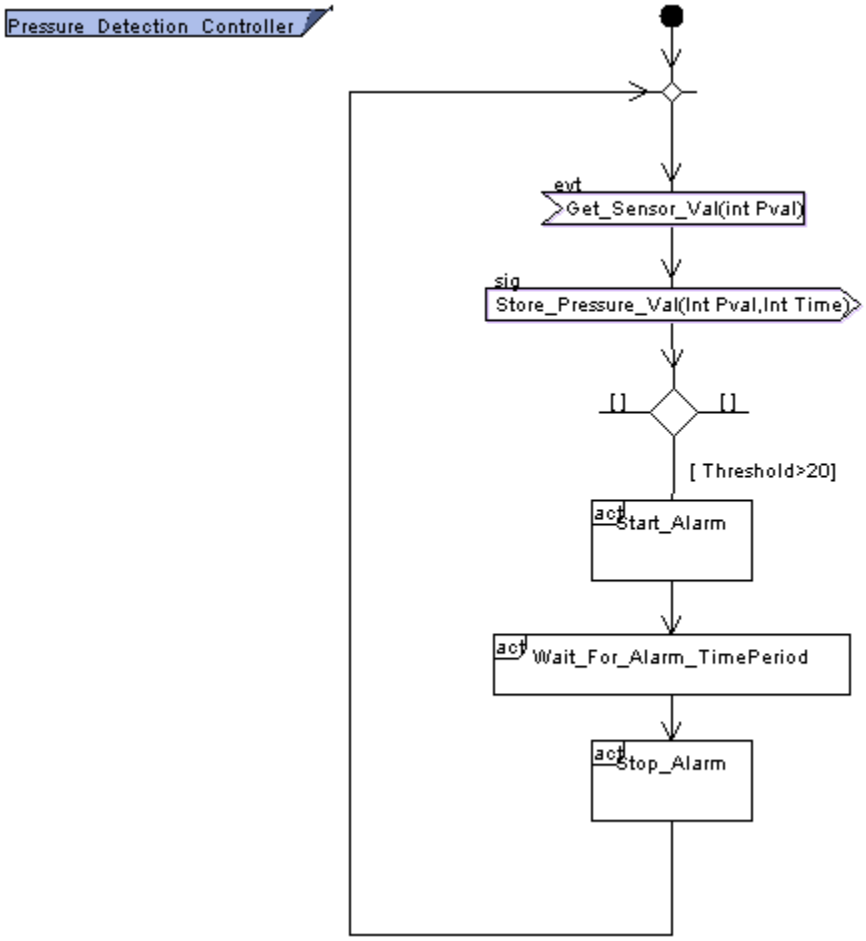


3-System Analaysis:

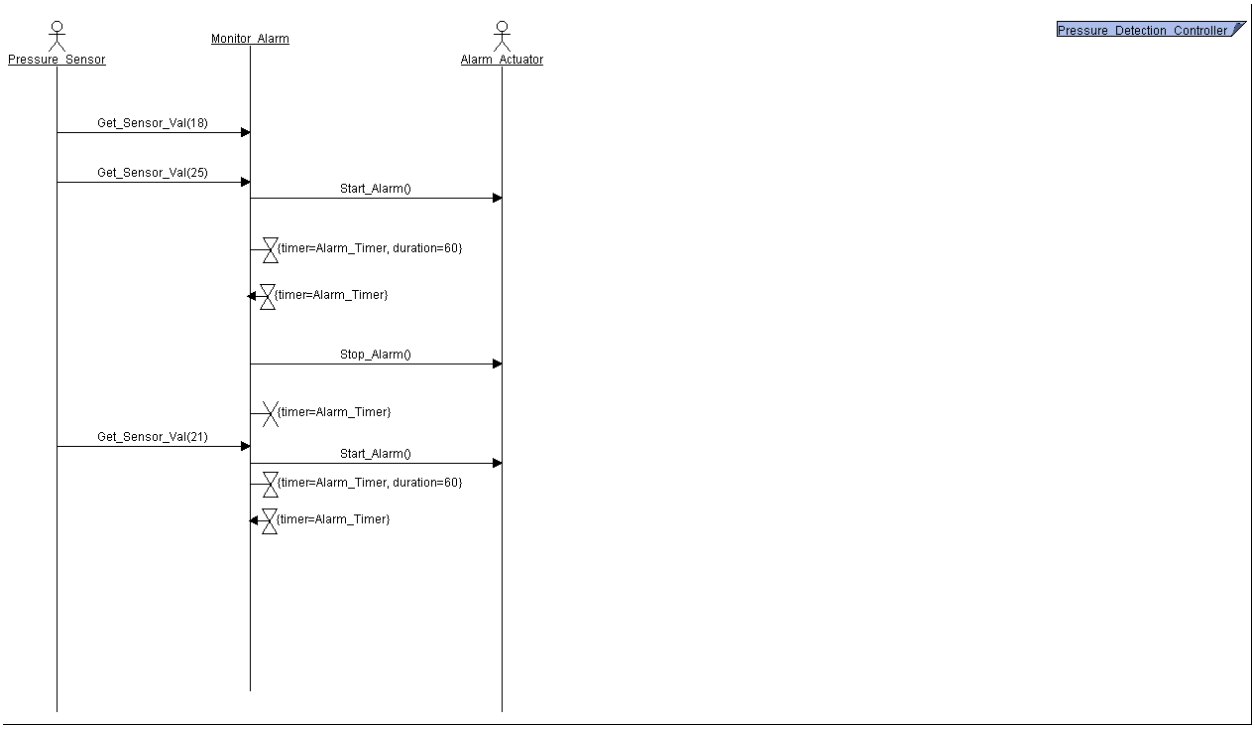
- **Use Case Diagram**



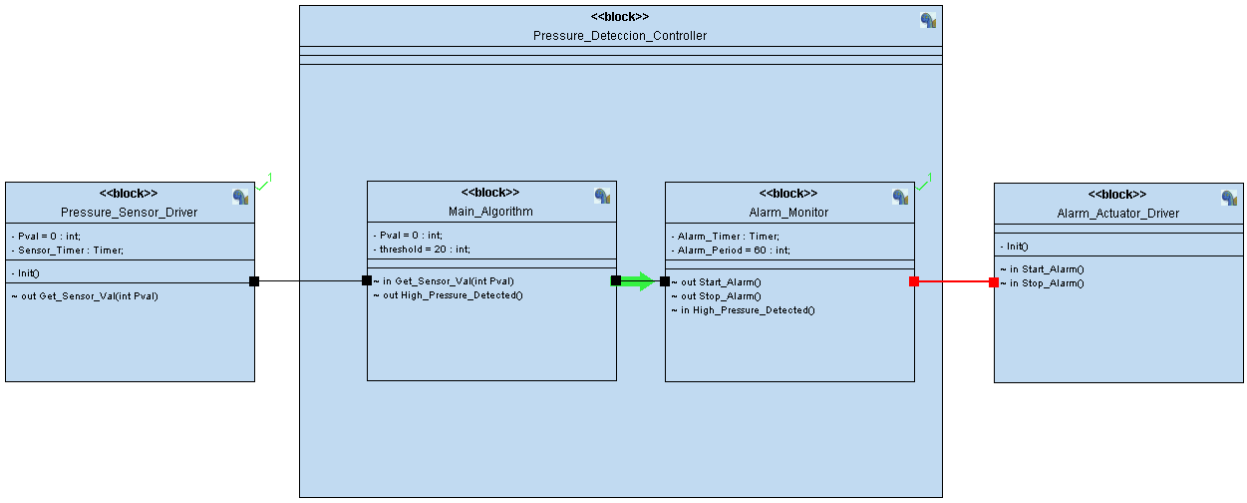
- **Activity Diagram:**



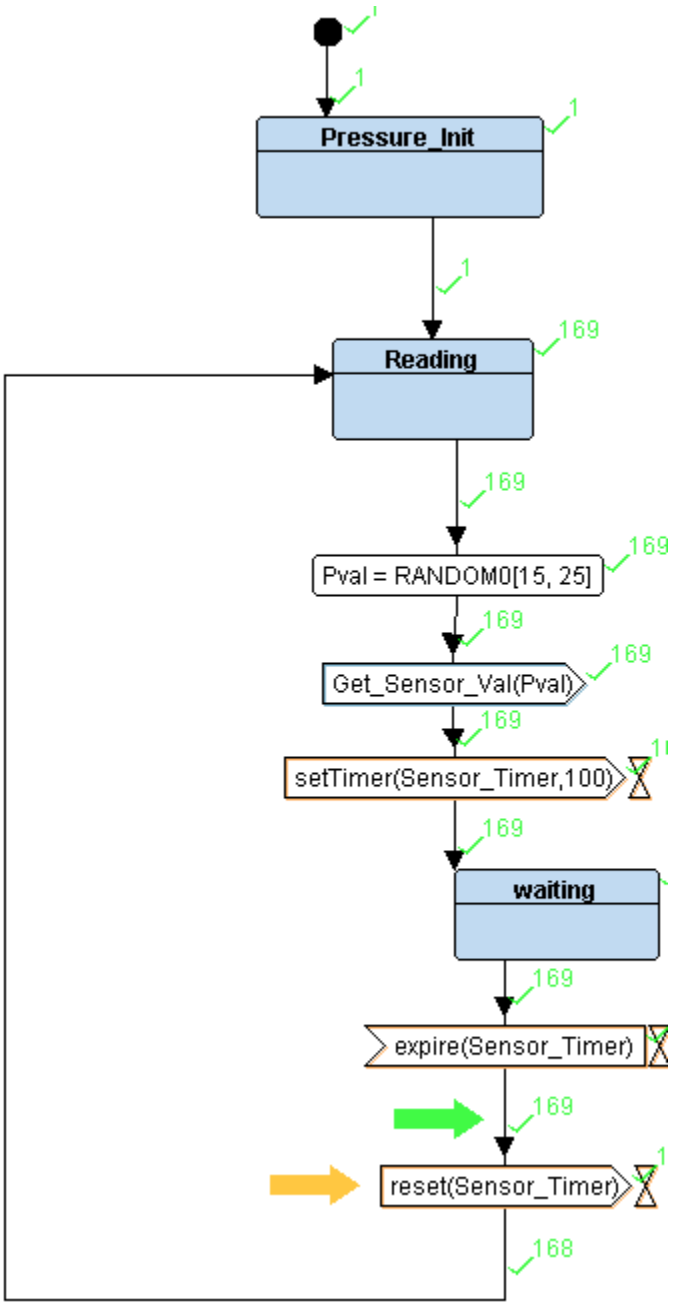
- **Sequence Diagram:**



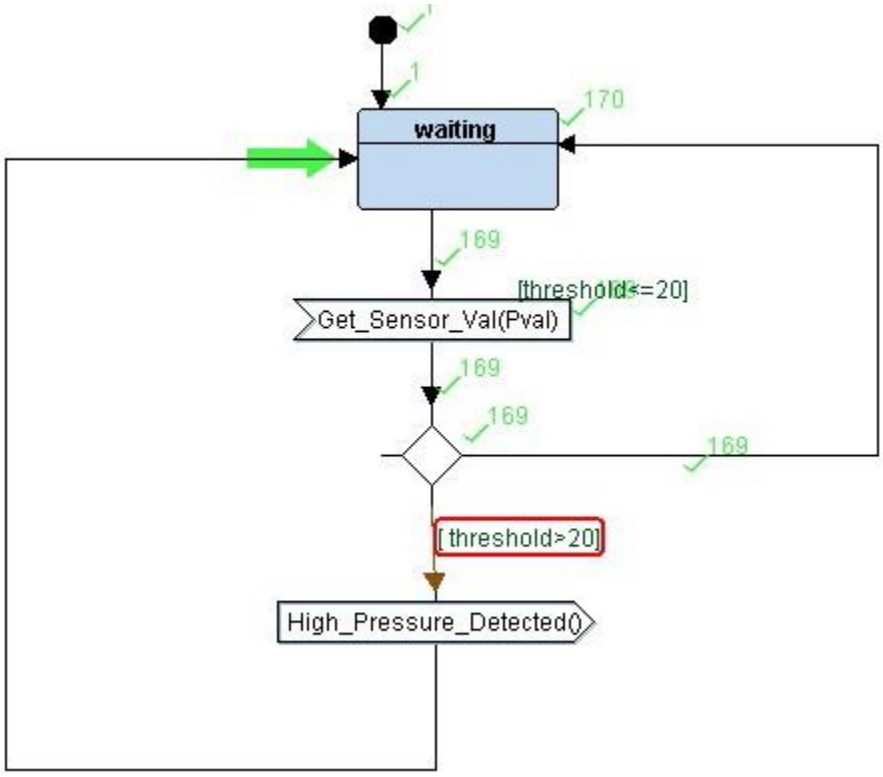
5-System Design Diagram (Block Diagram):



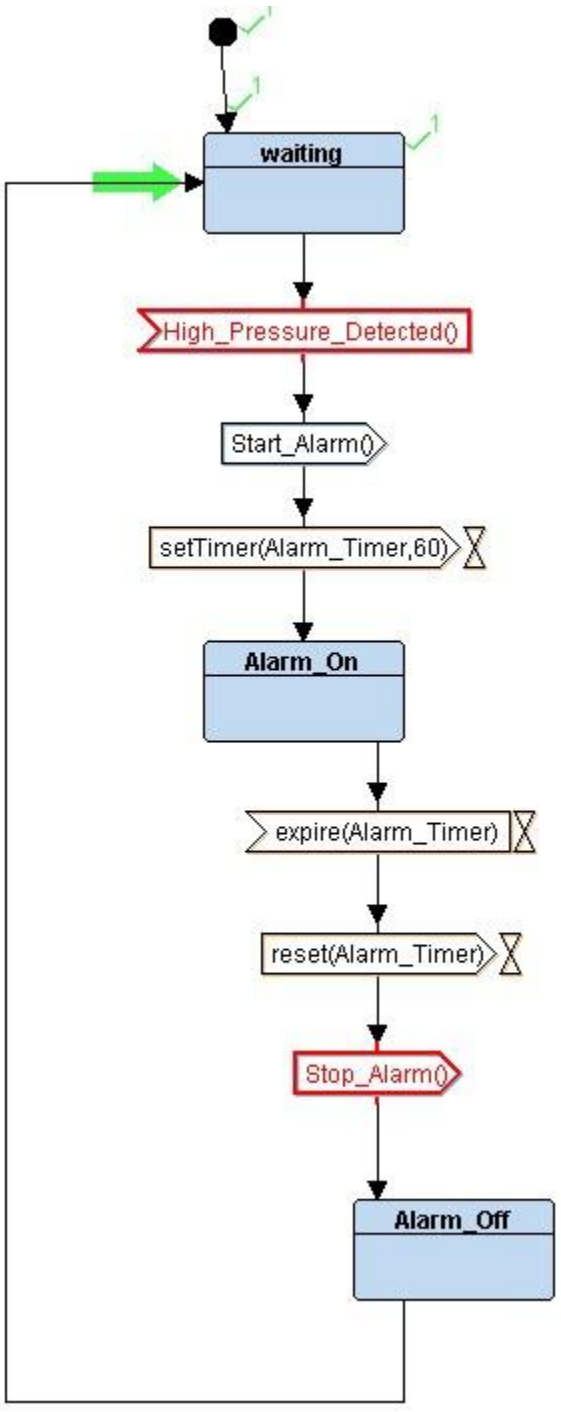
- **Pressure Sensor State Machine:**



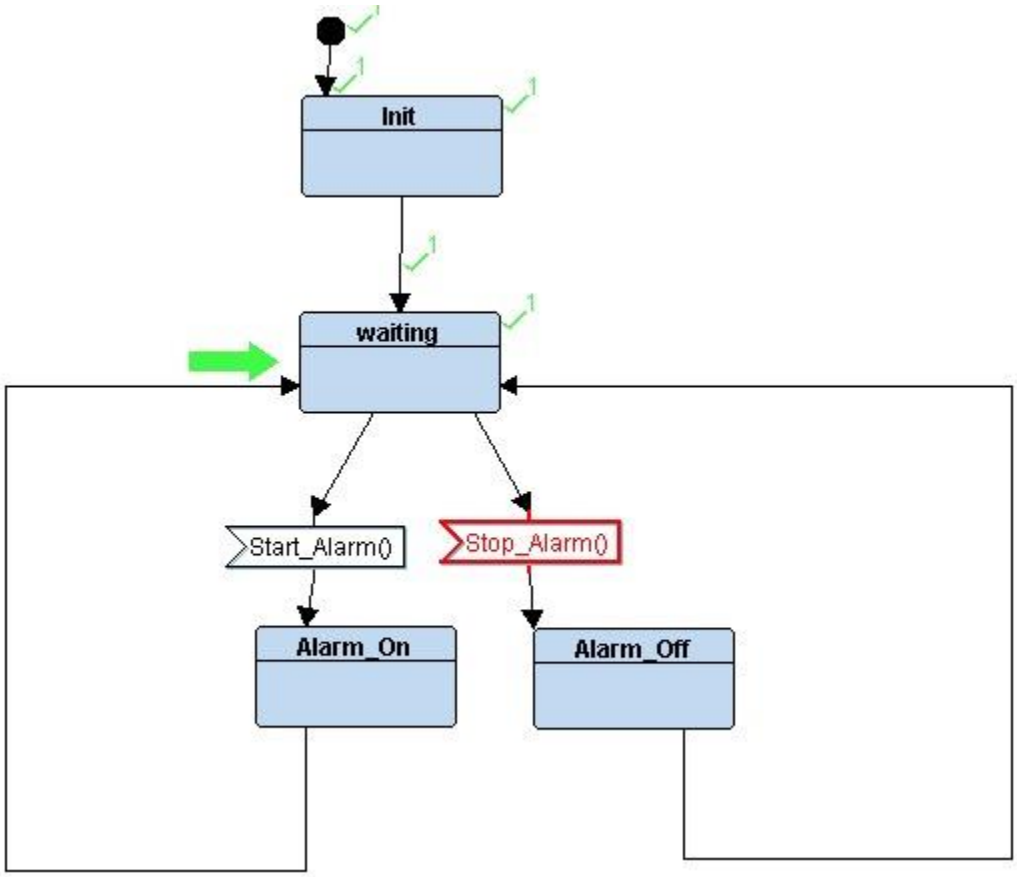
Main Algorithm State Machine :



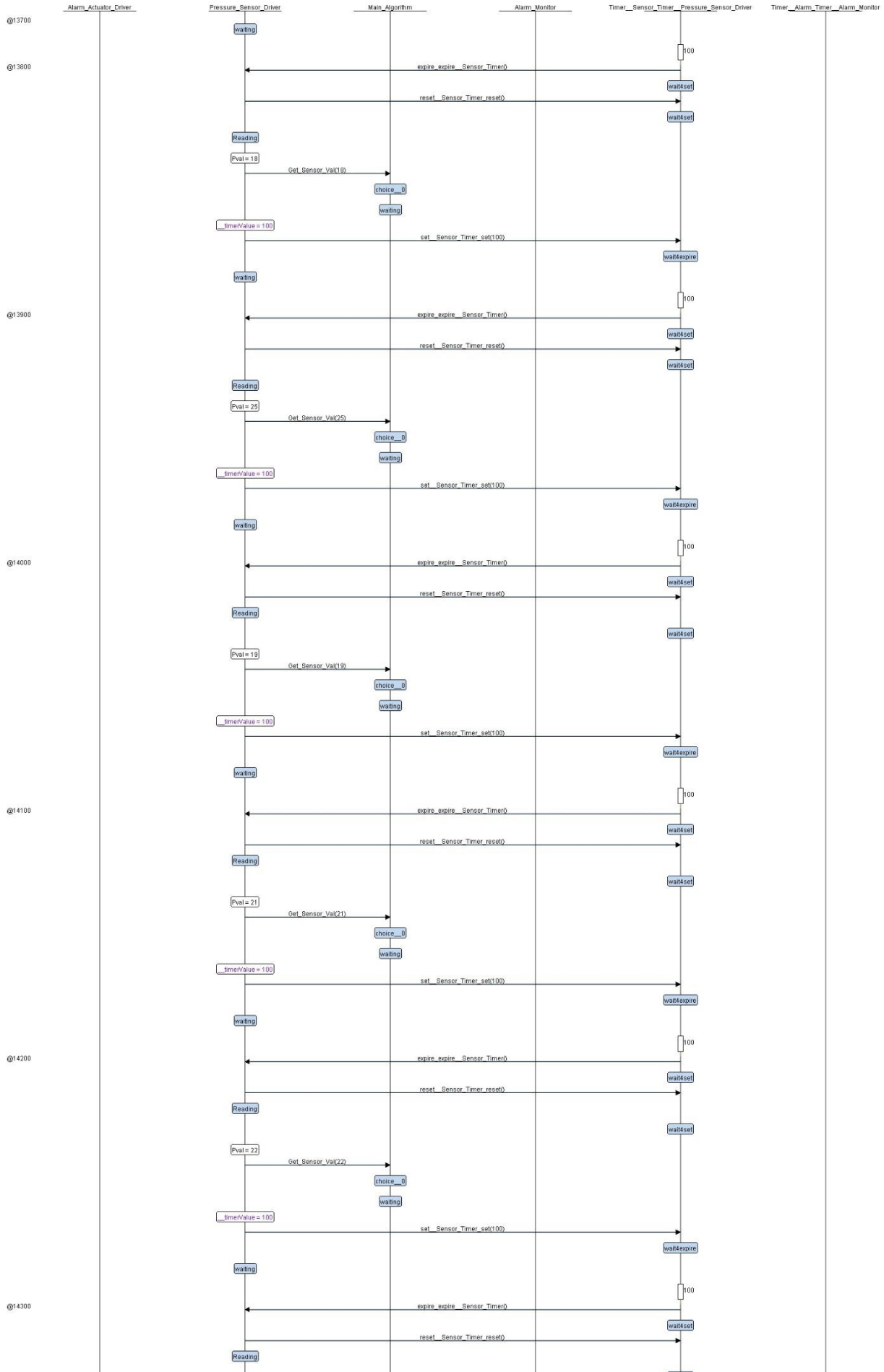
- **Alarm Monitor State Machine:**



Alarm Actuator State Machine:

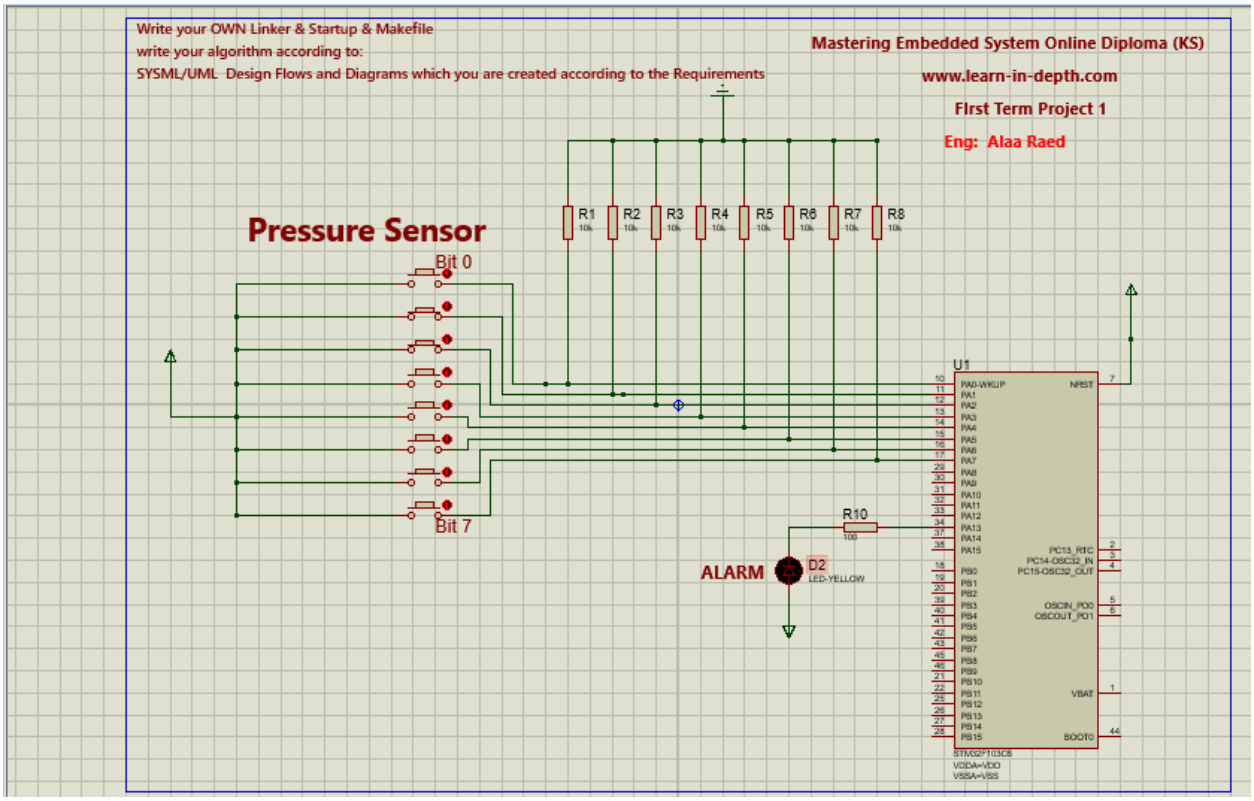


Verification :

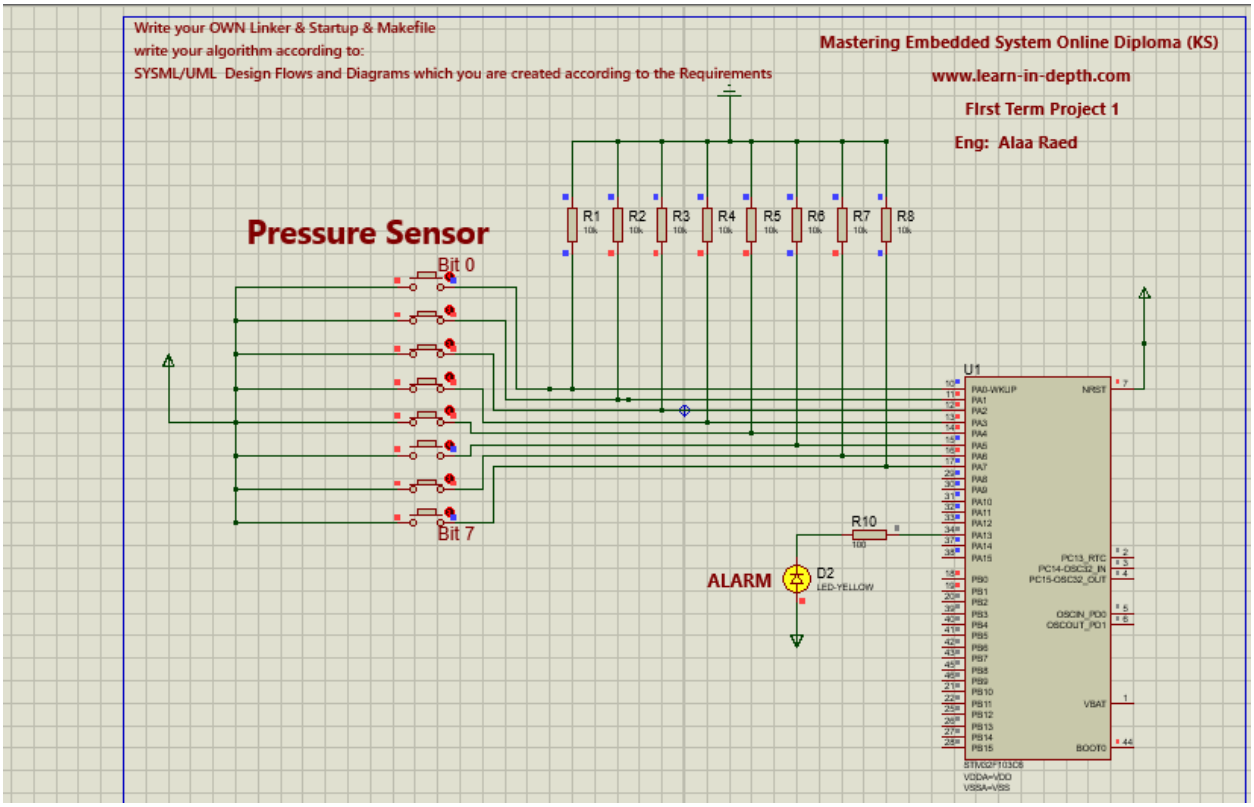

Alarm_Actuator_Driver
Pressure_Sensor_Driver
Main_Algorithm
Alarm_Monitor
Timer__Sensor_Timer__Pressure_Sensor_Driver
Timer__Alarm_Timer__Alarm_Monitor
@13700
waiting
100
@13800
expire_expire__Sensor_Timer()
wait4set
reset__Sensor_Timer_reset()
wait4set
Reading
Pval = 18
Get_Sensor_Val(18)
choice__0
waiting
__timerValue = 100
set__Sensor_Timer_set(100)
wait4expire
waiting
100
@13900
expire_expire__Sensor_Timer()
wait4set
reset__Sensor_Timer_reset()
wait4set
Reading
Pval = 25
Get_Sensor_Val(25)
choice__0
waiting
__timerValue = 100
set__Sensor_Timer_set(100)
wait4expire
waiting
100
@14000
expire_expire__Sensor_Timer()
wait4set
reset__Sensor_Timer_reset()
Reading
wait4set
Pval = 19
Get_Sensor_Val(19)
choice__0
waiting
__timerValue = 100
set__Sensor_Timer_set(100)
wait4expire
waiting
100
@14100
expire_expire__Sensor_Timer()
wait4set
reset__Sensor_Timer_reset()
Reading
wait4set
Pval = 21
Get_Sensor_Val(21)
choice__0
waiting
__timerValue = 100
set__Sensor_Timer_set(100)
wait4expire
waiting
100
@14200
expire_expire__Sensor_Timer()
wait4set
reset__Sensor_Timer_reset()
Reading
wait4set
Pval = 22
Get_Sensor_Val(22)
choice__0
waiting
__timerValue = 100
set__Sensor_Timer_set(100)
wait4expire
waiting
100
@14300
expire_expire__Sensor_Timer()
wait4set
reset__Sensor_Timer_reset()
Reading

Proteus Simulation:

Case 1:Pressure_val(6)<threshold(20)



Case 2: Pressure_val(30)>hreshold(20)


Write your OWN Linker & Startup & Makefile
write your algorithm according to:
SYSML/UML Design Flows and Diagrams which you are created according to the Requirements
Mastering Embedded System Online Diploma (KS)
www.learn-in-depth.com
FIrst Term Project 1
Eng: Alaa Raed
Pressure Sensor
Bit 0
Bit 7
R1
R2
R3
R4
R5
R6
R7
R8
R10
ALARM
D2
LED-YELLOW
U1

Symbols:

```
$ arm-none-eabi-nm.exe Project_1.elf
20000410 B _E_bss
20000004 D _E_DATA
0800036c T _E_text
20001410 B _heap_End
20000004 B _S_bss
20000000 D _S_DATA
20002410 B _stack_top
20002414 B Alarm_Actuator
08000010 T Alarm_Actuator_Init
20002410 B Alarm_Actuator_State_id
20000004 B alarm_flag
20002418 B Alarm_Monitor_State
2000241c B AlarmMonitor_State_id
080002dc T Deafault_Handler
080001b0 T Delay
08000000 T g_p_fn_vectors
080001d0 T getPressureVal
08000224 T GPIO_INITIALIZATION
080002dc W H_Fault_Handler
080000bc T High_Pressure_Detected
20002424 B High_Pressure_State_id
08000274 T Init
080002b4 T main
20002420 B Main_Algorithm
080002dc W NMI_Handler
08000170 T Pressure_Sensor_Init
20002428 B Pressure_Sensor_State
20000008 b Pressure_val
2000242c B PS_State_id
2000000c b Pval
080002e8 T Reset_Handler
080001e8 T Set_Alarm_actuator
0800011c T Set_Presssure_Val
080000d8 T st_alarm_off
080000f0 T st_alarm_on
08000138 T st_High_Pressure
0800006c T st_OFF
08000094 T st_ON
0800017c T st_reading
08000054 T st_Waiting
20000010 b Stack_top
0800001c T Start_Alarm
08000038 T Stop_Alarm
20000000 d threshold
```

MapFile:

pressure_sensor.c | Pressure_Sensor.h | Map_file.map

```
1
2   Allocating common symbols
3   Common symbol       size              file
4
5   Alarm_Monitor_State
6                       0x4               Alarm_Monitor.o
7   AlarmMonitor_State_id
8                       0x1               Alarm_Monitor.o
9   Pressure_Sensor_State
10                      0x4               Pressure_Sensor.o
11  Alarm_Actuator_State_id
12                      0x1               Alarm_Actuator.o
13  PS_State_id         0x1               Pressure_Sensor.o
14  Main_Algorithm      0x4               Main_Algorithm.o
15  Alarm_Actuator      0x4               Alarm_Actuator.o
16  High_Pressure_State_id
17                      0x1               Main_Algorithm.o
18
19  Memory Configuration
20
21  Name              Origin             Length             Attributes
22  flash             0x08000000         0x00010000         xr
23  sram              0x20000000         0x00005000         xrw
24  *default*         0x00000000         0xffffffff
25
26  Linker script and memory map
27
28
29  .text           0x08000000      0x36c
30   *(.vectors*)
31   .vectors       0x08000000       0x10 startup.o
32                  0x08000000                g_p_fn_vectors
33   *(.text*)
34   .text          0x08000010       0xac Alarm_Actuator.o
35                  0x08000010                Alarm_Actuator_Init
36                  0x0800001c                Start_Alarm
37                  0x08000038                Stop_Alarm
38                  0x08000054                st Waiting
```

pressure_sensor.c | Pressure_Sensor.h | Map_file.map

```
                0x0800006c                st_OFF
                0x08000094                st_ON
 .text          0x080000bc       0x60 Alarm_Monitor.o
                0x080000bc                High_Pressure_Detected
                0x080000d8                st_alarm_off
                0x080000f0                st_alarm_on
 .text          0x0800011c       0x54 Main_Algorithm.o
                0x0800011c                Set_Presssure_Val
                0x08000138                st_High_Pressure
 .text          0x08000170       0x40 Pressure_Sensor.o
                0x08000170                Pressure_Sensor_Init
                0x0800017c                st_reading
 .text          0x080001b0       0xc4 driver.o
                0x080001b0                Delay
                0x080001d0                getPressureVal
                0x080001e8                Set_Alarm_actuator
                0x08000224                GPIO_INITIALIZATION
 .text          0x08000274       0x68 main.o
                0x08000274                Init
                0x080002b4                main
 .text          0x080002dc       0x90 startup.o
                0x080002dc                H_Fault_Handler
                0x080002dc                Deafault_Handler
                0x080002dc                NMI_Handler
                0x080002e8                Reset_Handler
 *(.rodata)
                0x0800036c                _E_text = .

.glue_7         0x0800036c        0x0
 .glue_7        0x0800036c        0x0 linker stubs

.glue_7t        0x0800036c        0x0
 .glue_7t       0x0800036c        0x0 linker stubs

.vfp11_veneer   0x0800036c        0x0
 .vfp11_veneer  0x0800036c        0x0 linker stubs

.v4_bx          0x0800036c        0x0
```

pressure_sensor.c | Pressure_Sensor.h | Map_file.map

```
 .v4_bx          0x0800036c        0x0 linker stubs

.iplt            0x0800036c        0x0
 .iplt           0x0800036c        0x0 Alarm_Actuator.o

.rel.dyn         0x0800036c        0x0
 .rel.iplt       0x0800036c        0x0 Alarm_Actuator.o

.data            0x20000000        0x4 load address 0x0800036c
                 0x20000000                _S_DATA = .
 *(.data)
 .data           0x20000000        0x0 Alarm_Actuator.o
 .data           0x20000000        0x0 Alarm_Monitor.o
 .data           0x20000000        0x4 Main_Algorithm.o
 .data           0x20000004        0x0 Pressure_Sensor.o
 .data           0x20000004        0x0 driver.o
 .data           0x20000004        0x0 main.o
 .data           0x20000004        0x0 startup.o
 *(.data.*)
                 0x20000004                . = ALIGN (0x4)
                 0x20000004                _E_DATA = .

.igot.plt        0x20000004        0x0 load address 0x08000370
 .igot.plt       0x20000004        0x0 Alarm_Actuator.o

.bss             0x20000004     0x2429 load address 0x08000370
                 0x20000004                _S_bss = .
 *(.bss*)
 .bss            0x20000004        0x4 Alarm_Actuator.o
                 0x20000004                alarm_flag
 .bss            0x20000008        0x0 Alarm_Monitor.o
 .bss            0x20000008        0x4 Main_Algorithm.o
 .bss            0x2000000c        0x4 Pressure_Sensor.o
 .bss            0x20000010        0x0 driver.o
 .bss            0x20000010        0x0 main.o
 .bss            0x20000010      0x400 startup.o
                 0x20000410                _E_bss = .
                 0x20000410                . = ALIGN (0x4)
```

pressure_sensor.c | Pressure_Sensor.h | Map_file.map

```
                0x20001410                . = (. + 0x1000)
 *fill*         0x20000410     0x1000
                0x20001410                _heap_End = .
                0x20001410                . = ALIGN (0x4)
                0x20002410                . = (. + 0x1000)
 *fill*         0x20001410     0x1000
                0x20002410                _stack_top = .
 COMMON         0x20002410        0x8 Alarm_Actuator.o
                0x20002410                Alarm_Actuator_State_id
                0x20002414                Alarm_Actuator
 COMMON         0x20002418        0x5 Alarm_Monitor.o
                0x20002418                Alarm_Monitor_State
                0x2000241c                AlarmMonitor_State_id
 *fill*         0x2000241d        0x3
 COMMON         0x20002420        0x5 Main_Algorithm.o
                0x20002420                Main_Algorithm
                0x20002424                High_Pressure_State_id
 *fill*         0x20002425        0x3
 COMMON         0x20002428        0x5 Pressure_Sensor.o
                0x20002428                Pressure_Sensor_State
                0x2000242c                PS_State_id
LOAD Alarm_Actuator.o
LOAD Alarm_Monitor.o
LOAD Main_Algorithm.o
LOAD Pressure_Sensor.o
LOAD driver.o
LOAD main.o
LOAD startup.o
OUTPUT(Project_1.elf elf32-littlearm)

.comment        0x00000000       0x7e
 .comment       0x00000000       0x7e Alarm_Actuator.o
                                 0x7f (size before relaxing)
 .comment       0x0000007e       0x7f Alarm_Monitor.o
 .comment       0x0000007e       0x7f Main_Algorithm.o
 .comment       0x0000007e       0x7f Pressure_Sensor.o
 .comment       0x0000007e       0x7f driver.o
 .comment       0x0000007e       0x7f main.o
```

```
 .comment        0x0000007e       0x7f startup.o

.ARM.attributes
                 0x00000000       0x33
 .ARM.attributes
                 0x00000000       0x33 Alarm_Actuator.o
 .ARM.attributes
                 0x00000033       0x33 Alarm_Monitor.o
 .ARM.attributes
                 0x00000066       0x33 Main_Algorithm.o
 .ARM.attributes
                 0x00000099       0x33 Pressure_Sensor.o
 .ARM.attributes
                 0x000000cc       0x33 driver.o
 .ARM.attributes
                 0x000000ff       0x33 main.o
 .ARM.attributes
                 0x00000132       0x33 startup.o
```