Mastering Embedded System Online Diploma

www.learn-in-depth.com

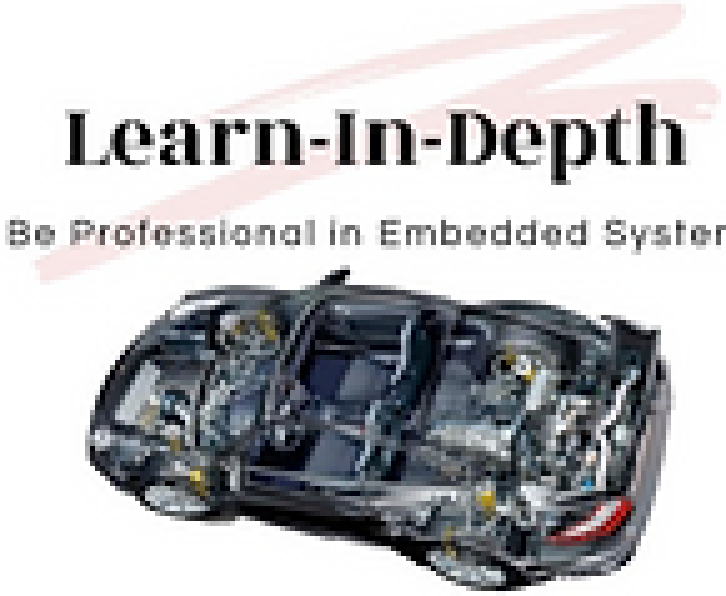
First Term (Final Project 1)

Pressure Controller System

Eng. Beshoy Emad

My Profile:

https://www.learn-in-depth-store.com/certificate/eng.beshoyemad64%40gmail.com



Pressure Controller System

System Architecting

Step [1]: Case study

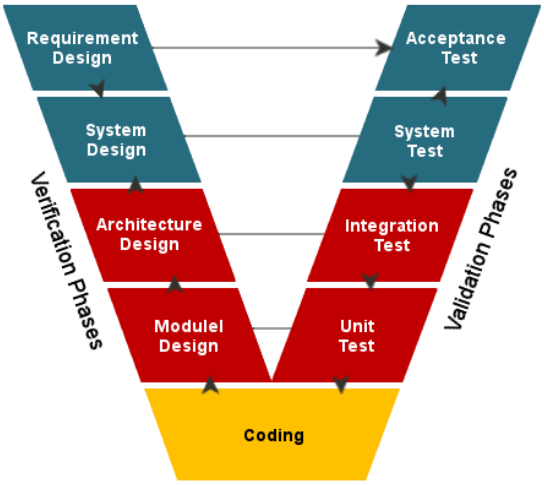
- A pressure controller informs the crew of a cabin with an alarm when the pressure exceeds 20 bars in the cabin.
- The alarm duration equals 60 seconds.
- Optional: the pressure values are stored in flash memory, we can add in the next version.

Assumptions:

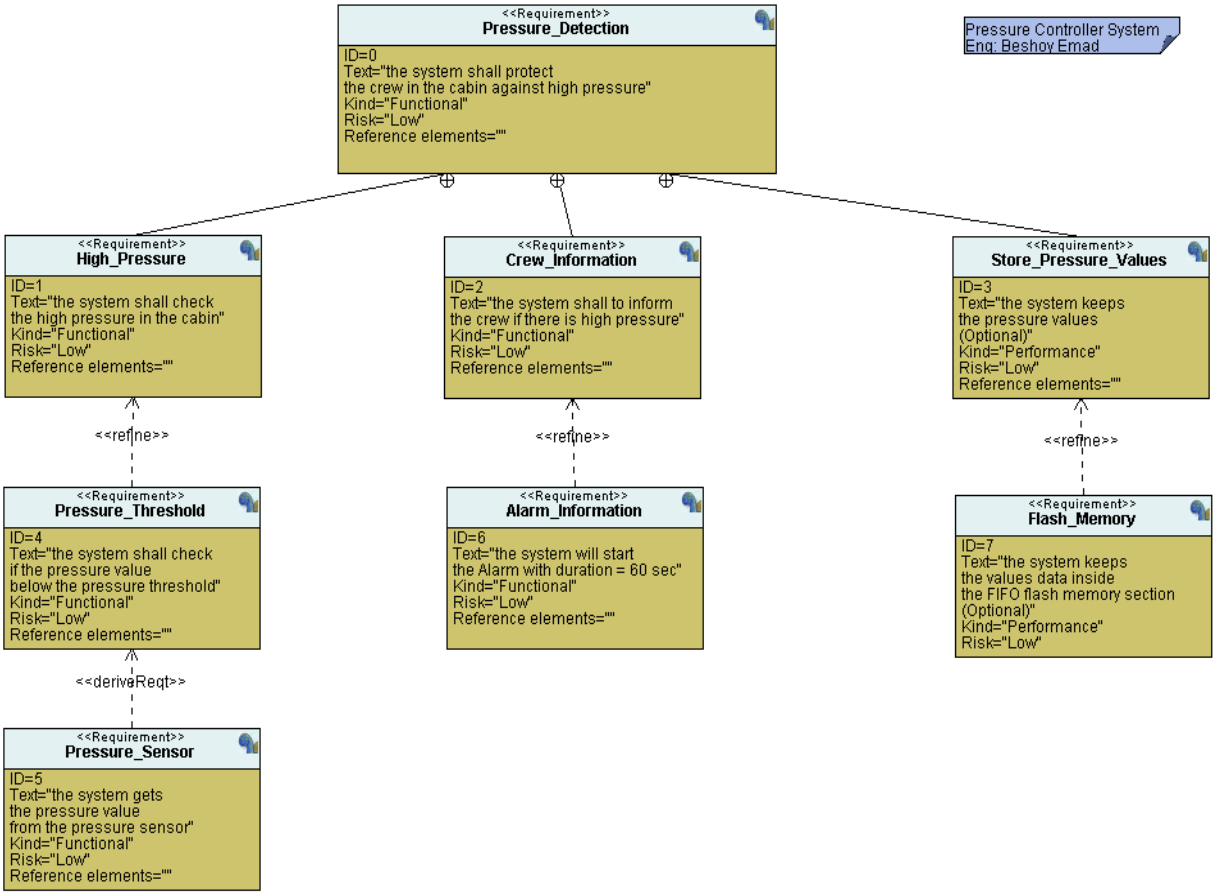
- The controller set up and shutdown procedures are not modeled.
- The controller maintenance is not modeled.
- The pressure sensor never fails.
- The alarm never fails.

Step [2]: Method

- V model



Step [3]: Requirements



Pressure detection includes:

1- High pressure
   - pressure threshold <<refine>>
   - pressure sensor <<deriveReqt>>

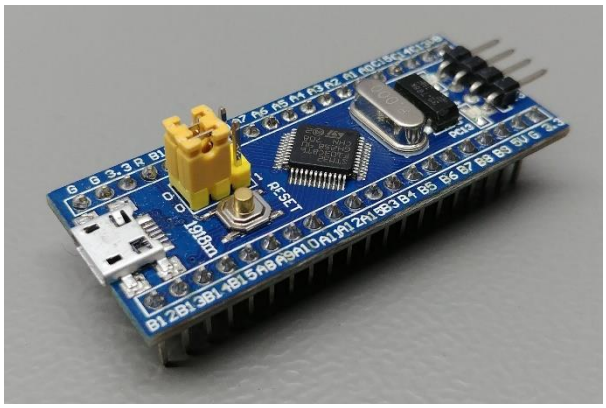
2- Crew information
   - alarm information<<refine>>

3- Store pressure value (optional)
   - flash memory<<refine>>

Step [4]: Space Exploration

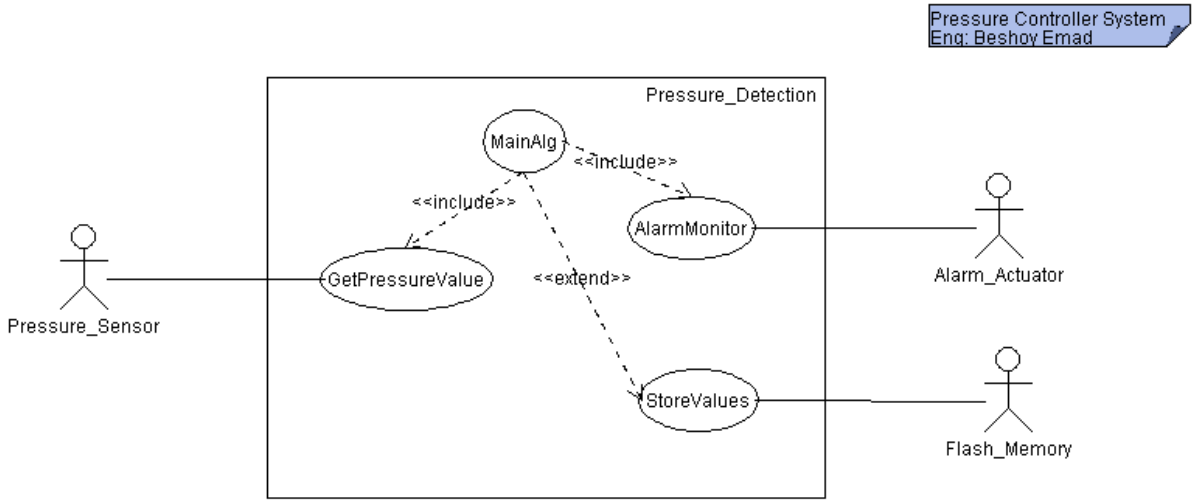
- I used in this project microcontroller stm3.

- 32-bit Arm processor core cortex m3.

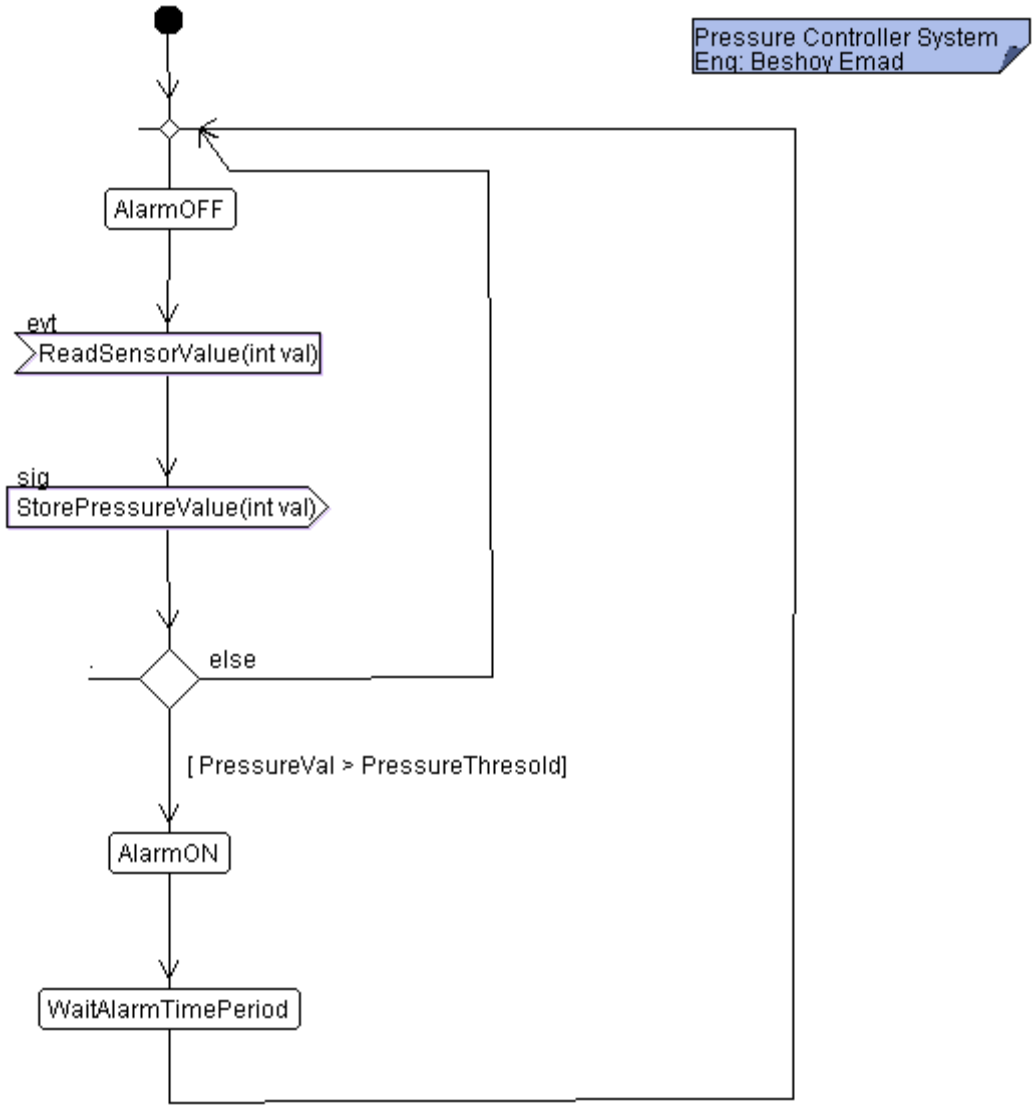


Step [5]: System Analysis

[1]: use case diagram



[2]: Activity Diagram



- The system starts alarm off.

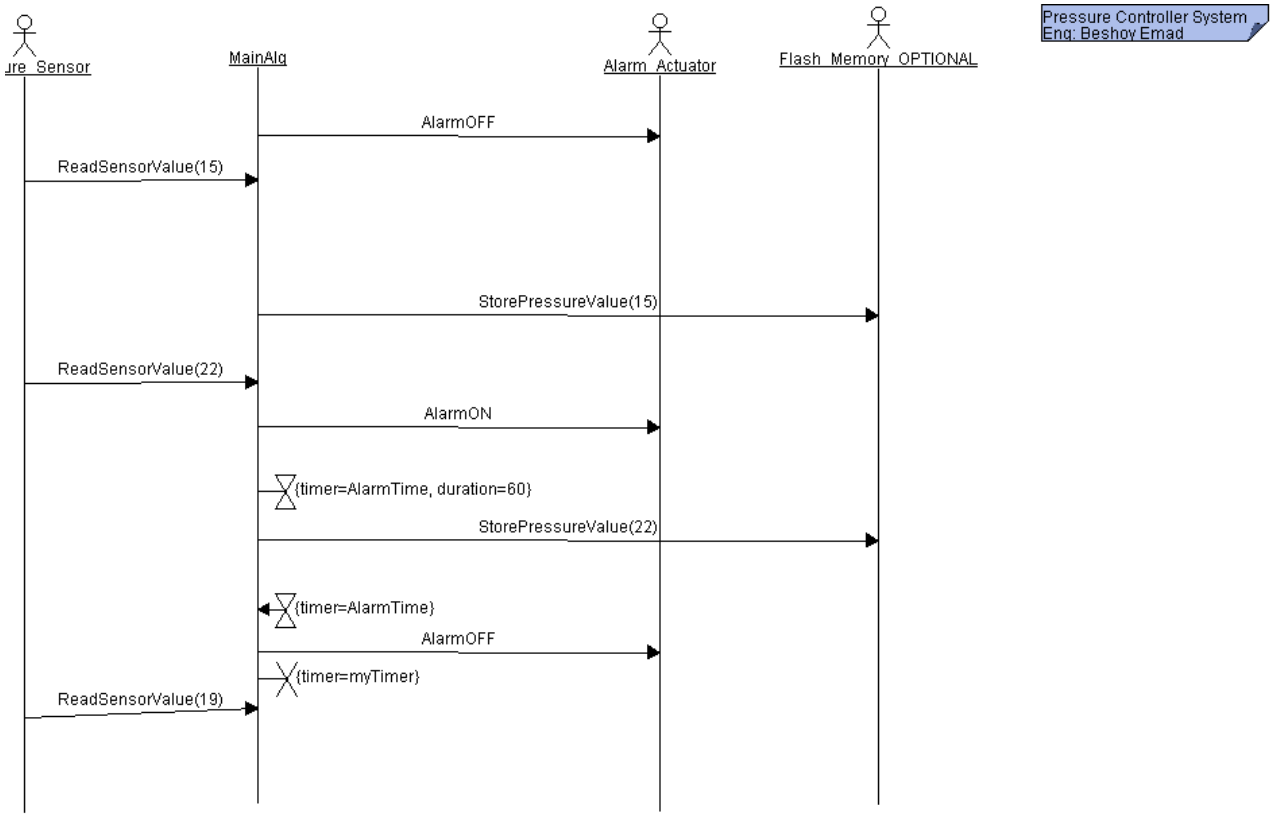
- Read the sensor value.

- Store this value in flash memory (optional).

- If the pressure value > pressure threshold the alarm starts on.

- Waiting alarm duration then starts alarm off.

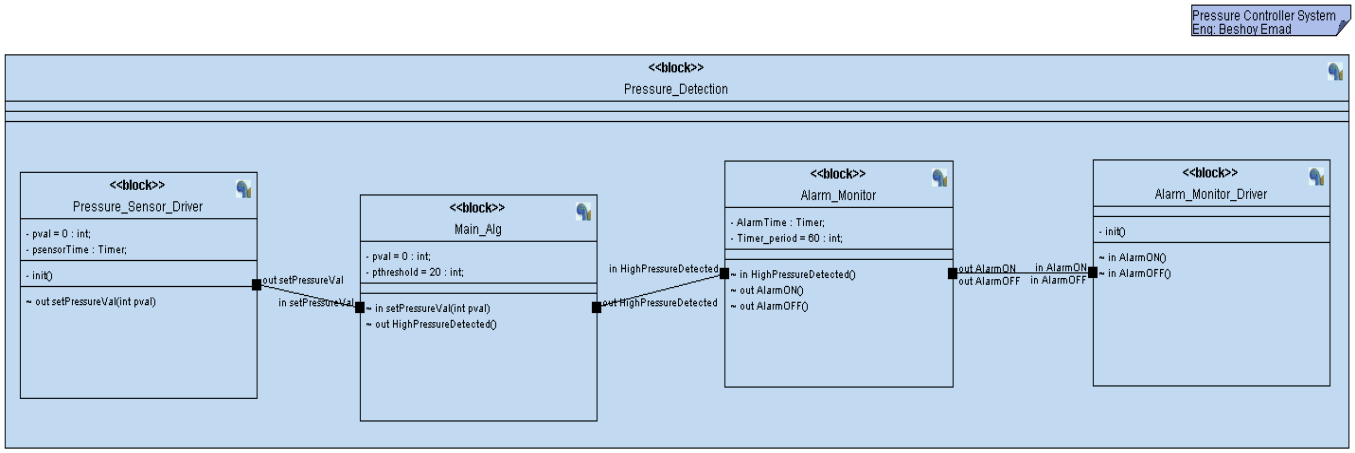
[3]: Sequence Diagram



State (1): Pressure sensor sending the pressure value= 15 to MainAlg. The action alarm actuator is off , and then the value is stored in flash memory.

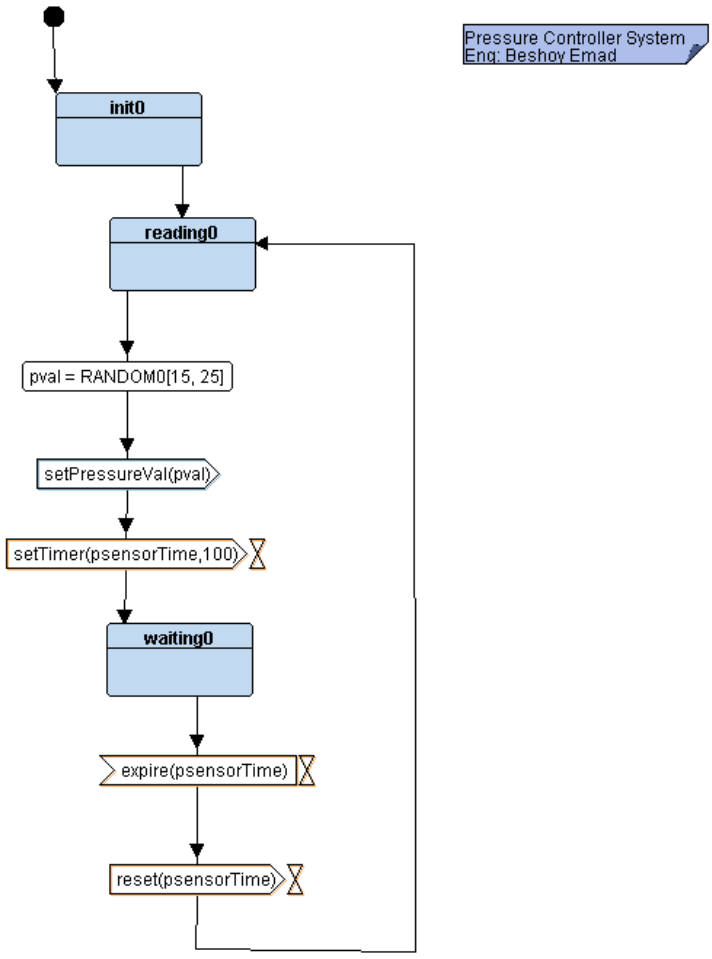
State (2): Pressure sensor sending the pressure value= 22 to MainAlg. The action alarm actuator is on, starts alarm duration then returns off state when the time expires. Then, the value is stored in flash memory.

Step [6]: System Diagram

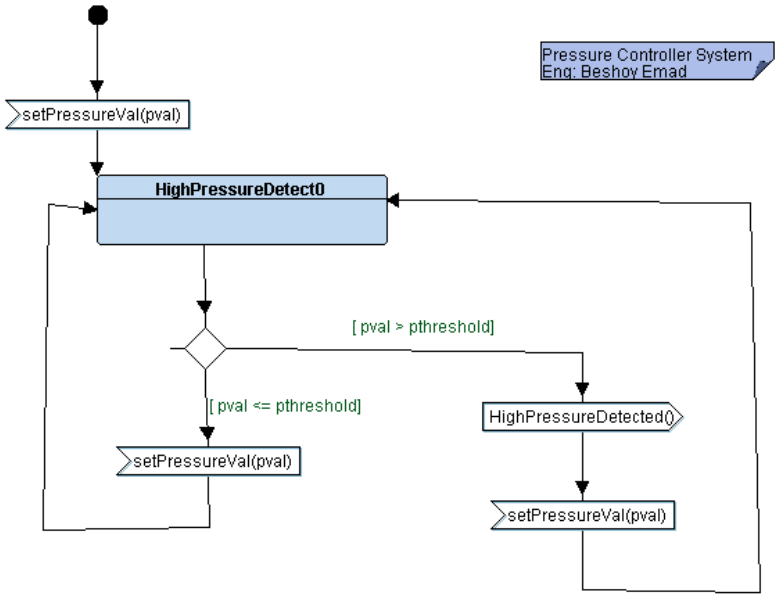


The project includes 4 modules:

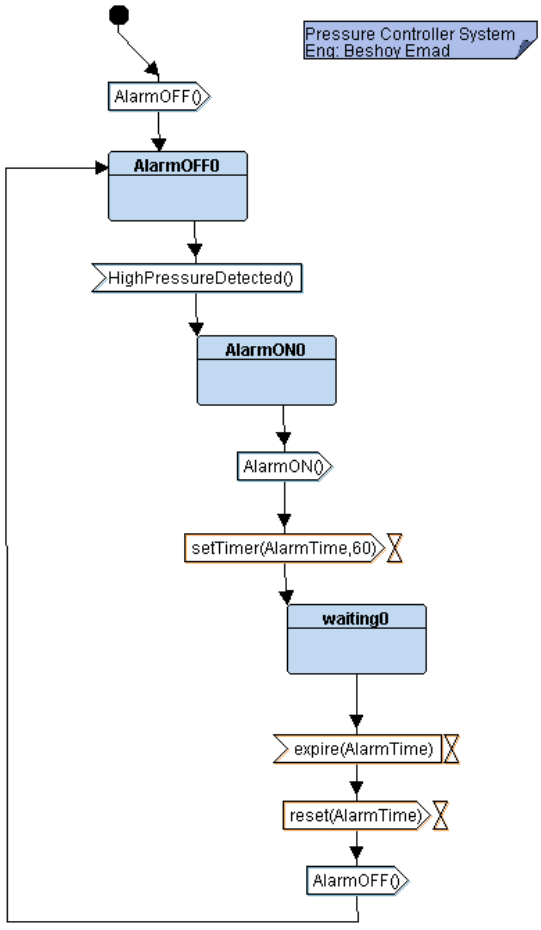
Module (1): Pressure_Sensor_Driver UML



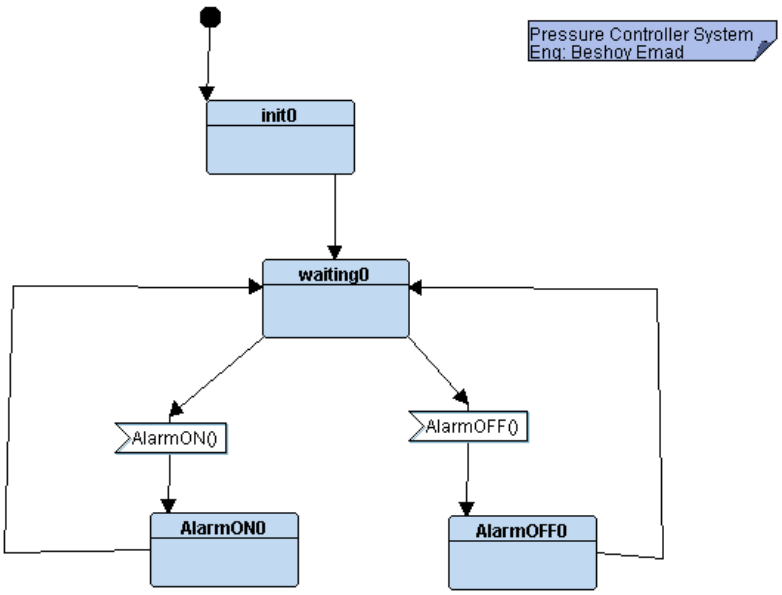
Module (2): Main_Alg UML



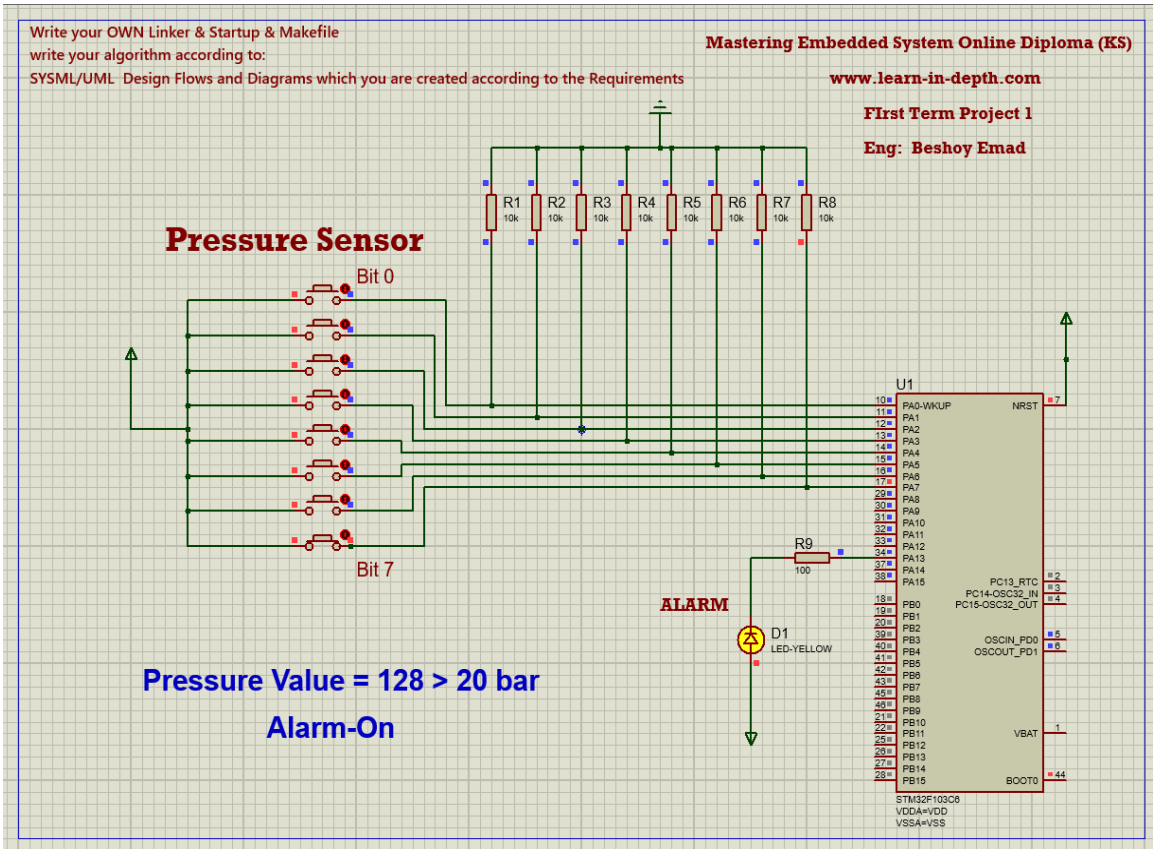
Module (3): Alarm_Monitor UML

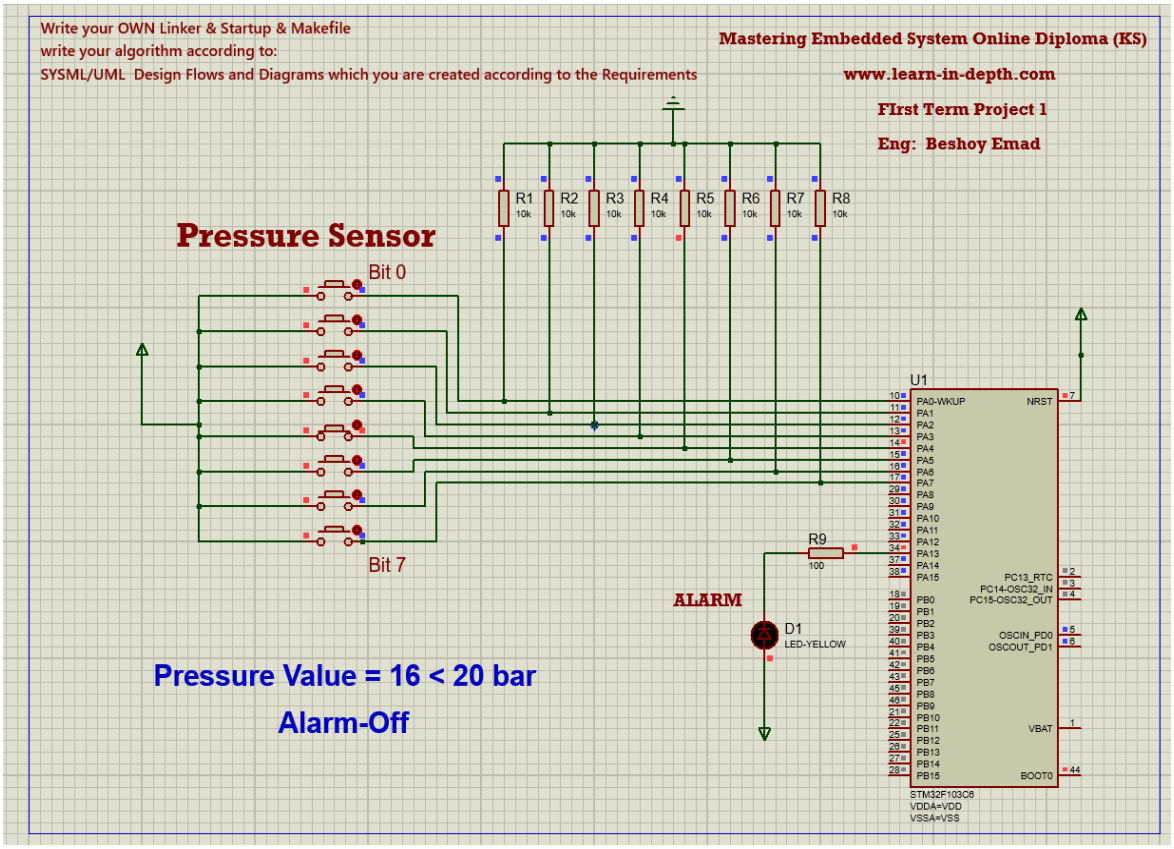


Module (4): Alarm_Monitor_Driver UML



Proteus





Section headers

```
Beshoy Emad@LAPTOP-GE4482BV MINGW64 ~/Downloads/HTI/ES-Kerolos Shenoda/
$ 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         00000070  00000000  00000000  00000034  2**2
                  CONTENTS, ALLOC, LOAD, RELOC, READONLY, CODE
  1 .data         00000000  00000000  00000000  000000a4  2**0
                  CONTENTS, ALLOC, LOAD, DATA
  2 .bss          00000004  00000000  00000000  000000a4  2**2
                  ALLOC
  3 .comment      0000007f  00000000  00000000  000000a4  2**0
                  CONTENTS, READONLY
  4 .ARM.attributes 00000033  00000000  00000000  00000123  2**0
                  CONTENTS, READONLY
```

Pressure_Sensor_Driver.o

```
Beshoy Emad@LAPTOP-GE4482BV MINGW64 ~/Downloads/HTI/ES-Kerolos Shenoda/First Term/Pressure_Controller_System
$ arm-none-eabi-objdump.exe -h algorithm.o

algorithm.o:     file format elf32-littlearm

Sections:
Idx Name          Size      VMA       LMA       File off  Algn
  0 .text         00000070  00000000  00000000  00000034  2**2
                  CONTENTS, ALLOC, LOAD, RELOC, READONLY, CODE
  1 .data         00000004  00000000  00000000  000000a4  2**2
                  CONTENTS, ALLOC, LOAD, DATA
  2 .bss          00000004  00000000  00000000  000000a8  2**2
                  ALLOC
  3 .comment      0000007f  00000000  00000000  000000a8  2**0
                  CONTENTS, READONLY
  4 .ARM.attributes 00000033  00000000  00000000  00000127  2**0
                  CONTENTS, READONLY
```

algorithm.o

```
Beshoy Emad@LAPTOP-GE4482BV MINGW64 ~/Downloads/HTI/ES-Kerolos Shenoda/First Term/Pressure_Controller_System
$ 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         00000098  00000000  00000000  00000034  2**2
                  CONTENTS, ALLOC, LOAD, RELOC, READONLY, CODE
  1 .data         00000004  00000000  00000000  000000cc  2**2
                  CONTENTS, ALLOC, LOAD, DATA
  2 .bss          00000000  00000000  00000000  000000d0  2**0
                  ALLOC
  3 .comment      0000007f  00000000  00000000  000000d0  2**0
                  CONTENTS, READONLY
  4 .ARM.attributes 00000033  00000000  00000000  0000014f  2**0
                  CONTENTS, READONLY
```

alarm_monitor.o

```
Beshoy Emad@LAPTOP-GE4482BV MINGW64 ~/Downloads/HTI/ES-Kerolos Shenoda/First Term/Pressure_Controller_System
$ arm-none-eabi-objdump.exe -h alarm_monitor_driver.o

alarm_monitor_driver.o:     file format elf32-littlearm

Sections:
Idx Name          Size      VMA       LMA       File off  Algn
  0 .text         000000b8  00000000  00000000  00000034  2**2
                  CONTENTS, ALLOC, LOAD, RELOC, READONLY, CODE
  1 .data         00000000  00000000  00000000  000000ec  2**0
                  CONTENTS, ALLOC, LOAD, DATA
  2 .bss          00000000  00000000  00000000  000000ec  2**0
                  ALLOC
  3 .comment      0000007f  00000000  00000000  000000ec  2**0
                  CONTENTS, READONLY
  4 .ARM.attributes 00000033  00000000  00000000  0000016b  2**0
                  CONTENTS, READONLY
```

alarm_monitor_driver.o

```
Beshoy Emad@LAPTOP-GE4482BV MINGW64 ~/Downloads/HTI/ES-Kerolos Shenoda/First Term/Pressure_Controller_System
$ arm-none-eabi-objdump.exe -h main.o

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 .comment      0000007f  00000000  00000000  000000b4  2**0
                  CONTENTS, READONLY
  4 .ARM.attributes 00000033  00000000  00000000  00000133  2**0
                  CONTENTS, READONLY
```

main.o

```
Beshoy Emad@LAPTOP-GE4482BV MINGW64 ~/Downloads/HTI/ES-Kerolos Shenoda/First Term/Pressure_Controller_System
$ arm-none-eabi-objdump.exe -h Pressure_Controller_System.elf

Pressure_Controller_System.elf:     file format elf32-littlearm

Sections:
Idx Name          Size      VMA       LMA       File off  Algn
  0 .text         00000420  08000000  08000000  00010000  2**2
                  CONTENTS, ALLOC, LOAD, READONLY, CODE
  1 .data         00000008  20000000  08000420  00020000  2**2
                  CONTENTS, ALLOC, LOAD, DATA
  2 .bss          00001020  20000008  08000428  00020008  2**2
                  ALLOC
  3 .comment      0000007e  00000000  00000000  00020008  2**0
                  CONTENTS, READONLY
  4 .ARM.attributes 00000033  00000000  00000000  00020086  2**0
                  CONTENTS, READONLY
```

pressure_controller_system.elf

Symbol Table

```
Beshoy Emad@LAPTOP-GE4482BV MINGW64 ~/Downloads/HTI/ES-Kerolos Shenoda/First Ter
m/Pressure_Controller_System/Obj
$ arm-none-eabi-objdump.exe -t pressure_sensor_driver.o

pressure_sensor_driver.o:     file format elf32-littlearm

SYMBOL TABLE:
00000000 l    df *ABS*  00000000 pressure_sensor_driver.c
00000000 l    d  .text  00000000 .text
00000000 l    d  .data  00000000 .data
00000000 l    d  .bss   00000000 .bss
00000000 l    d  .debug_info    00000000 .debug_info
00000000 l    d  .debug_abbrev  00000000 .debug_abbrev
00000000 l    d  .debug_loc     00000000 .debug_loc
00000000 l    d  .debug_aranges 00000000 .debug_aranges
00000000 l    d  .debug_line    00000000 .debug_line
00000000 l    d  .debug_str     00000000 .debug_str
00000000 l    d  .debug_frame   00000000 .debug_frame
00000000 l    d  .comment       00000000 .comment
00000000 l    d  .ARM.attributes        00000000 .ARM.attributes
00000001       O *COM*  00000001 PS_state_id
00000000 g     O .bss   00000004 PS_pval
00000004       O *COM*  00000004 PS_state
00000000 g     F .text  0000000c PS_init
0000000c g     F .text  00000038 ST_PS_reading
00000000         *UND*  00000000 getPressureVal
00000000         *UND*  00000000 setPressureVal
00000044 g     F .text  0000002c ST_PS_waiting
00000000         *UND*  00000000 Delay
```

Pressure_Sensor_Driver.o

```
Beshoy Emad@LAPTOP-GE4482BV MINGW64 ~/Downloads/HTI/ES-Kerolos Shenoda/First Term/Pressure_Controller_System
$ arm-none-eabi-objdump.exe -t algorithm.o

algorithm.o:     file format elf32-littlearm

SYMBOL TABLE:
00000000 l    df *ABS*  00000000 algorithm.c
00000000 l    d  .text  00000000 .text
00000000 l    d  .data  00000000 .data
00000000 l    d  .bss   00000000 .bss
00000000 l    d  .comment       00000000 .comment
00000000 l    d  .ARM.attributes        00000000 .ARM.attributes
00000001       O *COM*  00000001 Alg_state_id
00000000 g     O .bss   00000004 Alg_pval
00000000 g     O .data  00000004 Alg_pthreshold
00000004       O *COM*  00000004 Alg_state
00000000 g     F .text  0000002c setPressureVal
0000002c g     F .text  00000044 ST_Alg_HighPressureDetect
00000000         *UND*  00000000 alg_HighPressureDetect
```

algorithm.o

```
Beshoy Emad@LAPTOP-GE4482BV MINGW64 ~/Downloads/HTI/ES-Kerolos Shenoda/First Term/Pressure_Controller_System
$ arm-none-eabi-objdump.exe -t alarm_monitor.o

alarm_monitor.o:     file format elf32-littlearm

SYMBOL TABLE:
00000000 l    df *ABS*  00000000 alarm_monitor.c
00000000 l    d  .text  00000000 .text
00000000 l    d  .data  00000000 .data
00000000 l    d  .bss   00000000 .bss
00000000 l    d  .comment       00000000 .comment
00000000 l    d  .ARM.attributes        00000000 .ARM.attributes
00000001       O *COM*  00000001 AL_state_id
00000000 g     O .data  00000004 Timer_period
00000004       O *COM*  00000004 AL_state
00000000 g     F .text  0000001c alg_HighPressureDetect
00000040 g     F .text  00000024 ST_AL_AlarmON
0000001c g     F .text  00000024 ST_AL_AlarmOFF
00000000         *UND*  00000000 AlarmOFF
00000000         *UND*  00000000 AlarmON
00000064 g     F .text  00000034 ST_AL_waiting
00000000         *UND*  00000000 Delay
```

alarm_monitor.o

```
Beshoy Emad@LAPTOP-GE4482BV MINGW64 ~/Downloads/HTI/ES-Kerolos Shenoda/First Term/Pressure_Controller_System
$ arm-none-eabi-objdump.exe -t alarm_monitor_driver.o

alarm_monitor_driver.o:     file format elf32-littlearm

SYMBOL TABLE:
00000000 l    df *ABS*  00000000 alarm_monitor_driver.c
00000000 l    d  .text  00000000 .text
00000000 l    d  .data  00000000 .data
00000000 l    d  .bss   00000000 .bss
00000000 l    d  .comment       00000000 .comment
00000000 l    d  .ARM.attributes        00000000 .ARM.attributes
00000001       O *COM*  00000001 ALDR_state_id
00000004       O *COM*  00000004 ALDR_state
00000000 g     F .text  0000000c ALDR_init
0000000c g     F .text  0000001c AlarmON
00000068 g     F .text  00000028 ST_ALDR_AlarmON
00000028 g     F .text  0000001c AlarmOFF
00000090 g     F .text  00000028 ST_ALDR_AlarmOFF
00000044 g     F .text  00000024 ST_ALDR_waiting
00000000         *UND*  00000000 Set_Alarm_actuator
```

alarm_monitor_driver.o

```
Beshoy Emad@LAPTOP-GE4482BV MINGW64 ~/Downloads/HTI/ES-Kerolos Shenoda/First Term/Pressure_Controller_System
$ arm-none-eabi-objdump.exe -t main.o

main.o:     file format elf32-littlearm

SYMBOL TABLE:
00000000 l    df *ABS*  00000000 main.c
00000000 l    d  .text  00000000 .text
00000000 l    d  .data  00000000 .data
00000000 l    d  .bss   00000000 .bss
00000000 l    d  .comment       00000000 .comment
00000000 l    d  .ARM.attributes        00000000 .ARM.attributes
00000001       O *COM*  00000001 PS_state_id
00000001       O *COM*  00000001 Alg_state_id
00000001       O *COM*  00000001 AL_state_id
00000001       O *COM*  00000001 ALDR_state_id
00000000 g     F .text  00000048 setup
00000000         *UND*  00000000 PS_init
00000000         *UND*  00000000 ALDR_init
00000000         *UND*  00000000 PS_state
00000000         *UND*  00000000 ST_PS_reading
00000000         *UND*  00000000 Alg_state
00000000         *UND*  00000000 setPressureVal
00000000         *UND*  00000000 AL_state
00000000         *UND*  00000000 ST_AL_AlarmOFF
00000000         *UND*  00000000 ALDR_state
00000000         *UND*  00000000 ST_ALDR_waiting
00000048 g     F .text  00000038 main
00000000         *UND*  00000000 GPIO_INITIALIZATION
```

main.o

```
Beshoy Emad@LAPTOP-GE4482BV MINGW64 ~/Downloads/HTI/ES-Kerolos Shenoda/First Term/Pressure_Controller_System
$ arm-none-eabi-objdump.exe -t Pressure_Controller_System.elf

Pressure_Controller_System.elf:     file format elf32-littlearm

SYMBOL TABLE:
08000000 l    d  .text  00000000 .text
20000000 l    d  .data  00000000 .data
20000008 l    d  .bss   00000000 .bss
00000000 l    d  .comment       00000000 .comment
00000000 l    d  .ARM.attributes        00000000 .ARM.attributes
00000000 l    df *ABS*  00000000 startup.c
00000000 l    df *ABS*  00000000 alarm_monitor_driver.c
00000000 l    df *ABS*  00000000 algorithm.c
00000000 l    df *ABS*  00000000 main.c
00000000 l    df *ABS*  00000000 pressure_sensor_driver.c
00000000 l    df *ABS*  00000000 driver.c
00000000 l    df *ABS*  00000000 alarm_monitor.c
08000388 g     F .text  0000001c alg_HighPressureDetect
080000b8 g     F .text  0000001c AlarmON
08000028 g     F .text  00000084 reset_handler
2000101d g     O .bss   00000001 AL_state_id
20001018 g     O .bss   00000004 Alg_state
08000338 g     F .text  00000050 GPIO_INITIALIZATION
0800001c  w    F .text  0000000c NMI_handler
20000008 g       .data  00000000 _E_data
0800013c g     F .text  00000028 ST_ALDR_AlarmOFF
20000000 g       .data  00000000 _S_data
08000164 g     F .text  0000002c setPressureVal
20000010 g       .bss   00000000 _E_bss
0800001c  w    F .text  0000000c USAGE_FAULT_handler
08000260 g     F .text  00000038 ST_PS_reading
0800001c  w    F .text  0000000c NM_FAULT_handler
0800001c  w    F .text  0000000c BUS_FAULT_handler
08000190 g     F .text  00000044 ST_Alg_HighPressureDetect
0800001c g     F .text  0000000c default_handler
20001024 g     O .bss   00000004 AL_state
08000254 g     F .text  0000000c PS_init
080002e4 g     F .text  00000018 getPressureVal
20001020 g     O .bss   00000004 PS_state
20000000 g     O .data  00000004 Alg_pthreshold
20000008 g       .bss   00000000 _S_bss
20000004 g     O .data  00000004 Timer_period
080003a8 g     F .text  00000024 ST_AL_AlarmOFF
20001010 g       .bss   00000000 stack_top
080002fc g     F .text  0000003c Set_Alarm_actuator
0800021c g     F .text  00000038 main
0800001c  w    F .text  0000000c HARD_FAULT_handler
08000114 g     F .text  00000028 ST_ALDR_AlarmON
2000101c g     O .bss   00000001 Alg_state_id
080003ec g     F .text  00000034 ST_AL_waiting
2000000c g     O .bss   00000004 PS_pval
2000101e g     O .bss   00000001 PS_state_id
080002c4 g     F .text  00000020 Delay
080001d4 g     F .text  00000048 setup
08000298 g     F .text  0000002c ST_PS_waiting
080000f0 g     F .text  00000024 ST_ALDR_waiting
080003c8 g     F .text  00000024 ST_AL_AlarmON
20001010 g     O .bss   00000001 ALDR_state_id
080000d4 g     F .text  0000001c AlarmOFF
08000420 g       .text  00000000 _E_text
08000000 g     O .text  0000001c vectors
20000008 g     O .bss   00000004 Alg_pval
20001014 g     O .bss   00000004 ALDR_state
080000ac g     F .text  0000000c ALDR_init
```

pressure_controller_system.elf