First project

Mastering Embedded System Online Diploma

High-Pressure-Detector

Name: Abdullah Mohamed Ibrahim hassan

My Profile: abdallahmohamed4197@gmail.com (learn-in-depth.com)

My GitHub: AbdallahMohamedd (Abdallah mohamed) (github.com)

LinkedIn: Abdallah Mohamed | LinkedIn

Facebook: Facebook

Case study:

After meeting with the customer, we have outlined the core concept of the project, which includes the following key requirements:

- The system must be capable of measuring pressure within a crew cabin.
- An alarm should be triggered when the pressure exceeds 20 bar.
- The alarm should remain active for a duration of 60 seconds.
- Optionally, the system should have the capability to store measured pressure values in a flash memory, each accompanied by a timestamp.

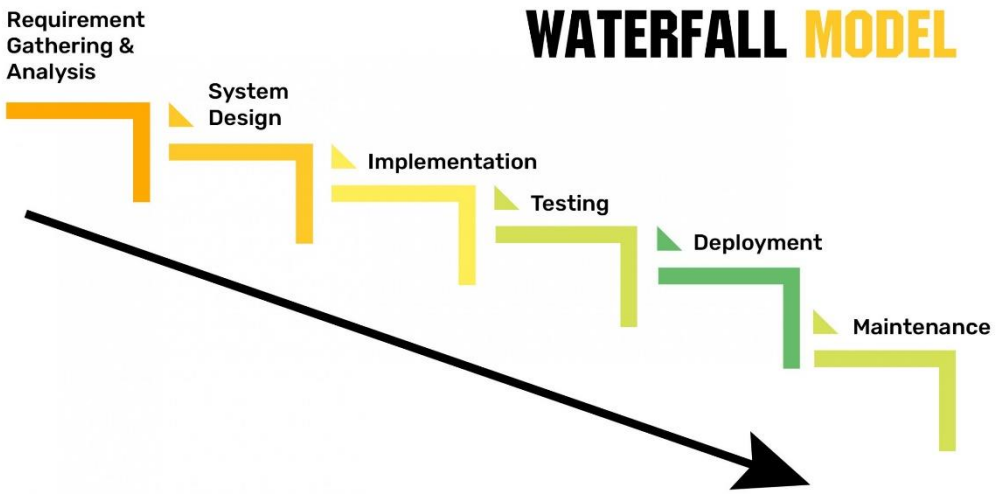
Several assumptions have been considered during project planning:

- The sensor is assumed to have no risk of failure.
- The alarm system is also assumed to have no risk of failure.
- The microcontroller unit (MCU) is expected to always have a power source.
- Maintenance and upkeep of the MCU are not within the scope of this project.

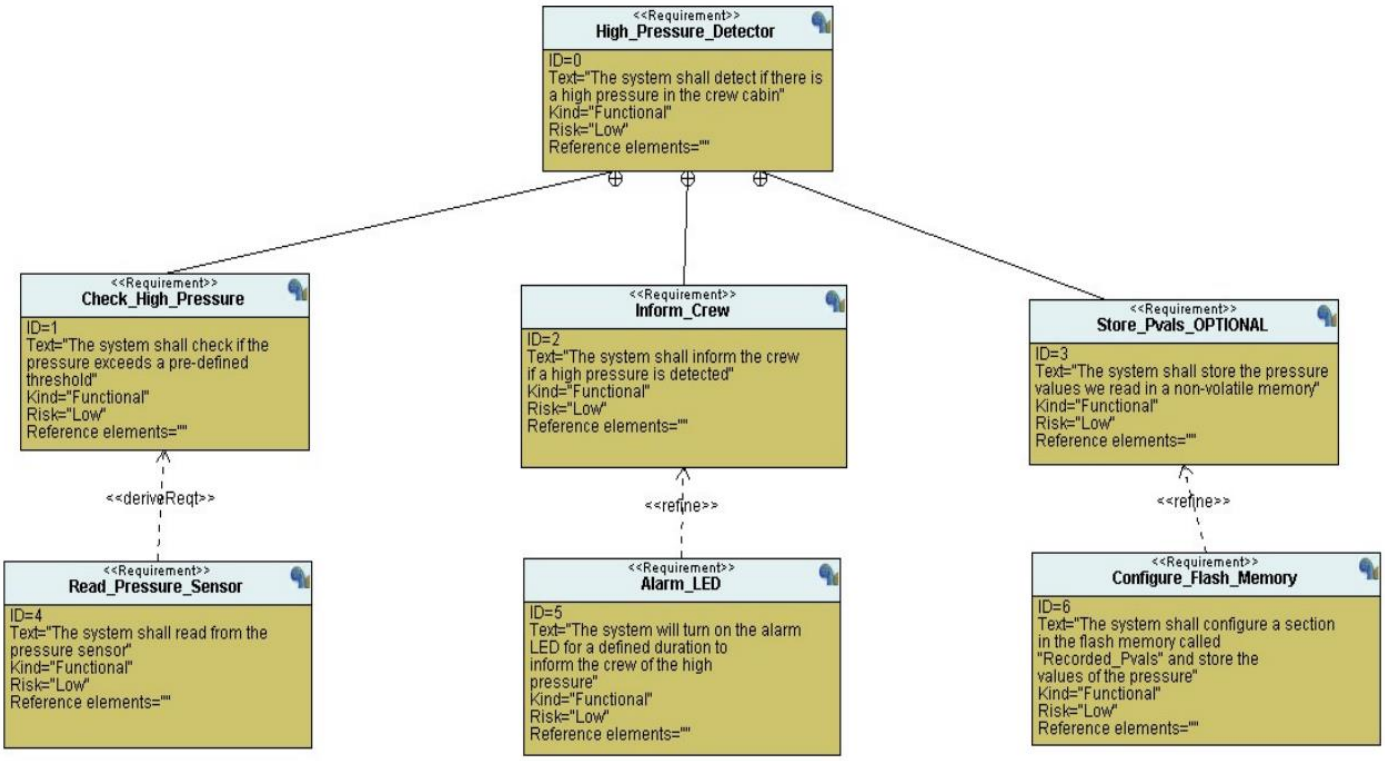
Regarding versioning, it's worth noting that the feature for saving captured pressure values will **NOT** be included in the initial software release.

Method:

The system was designed using the Waterfall model, following the UML standards.



Requirements:

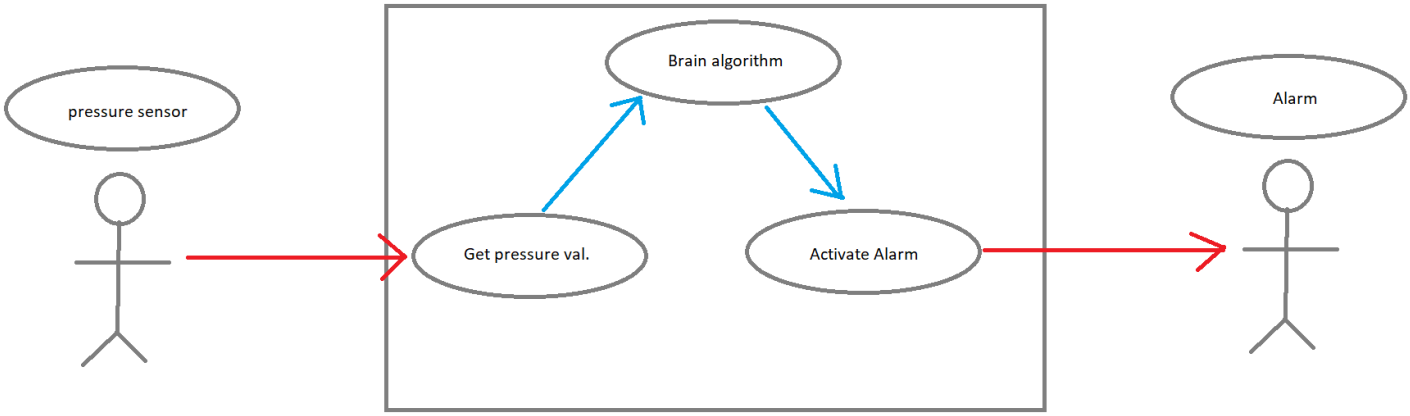


space exploration:

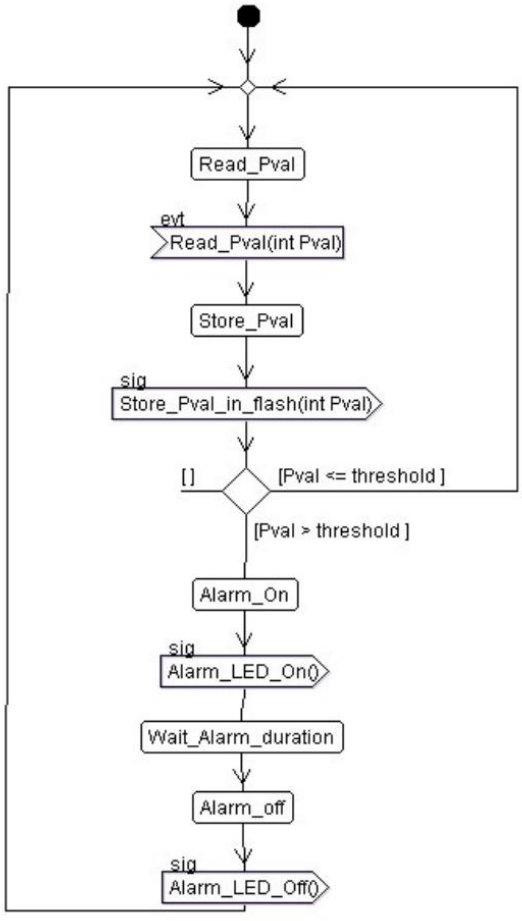
A single SOC Stm32 microcontroller with a **cortex-M3** processor will be used to implement this project.

System analysis:

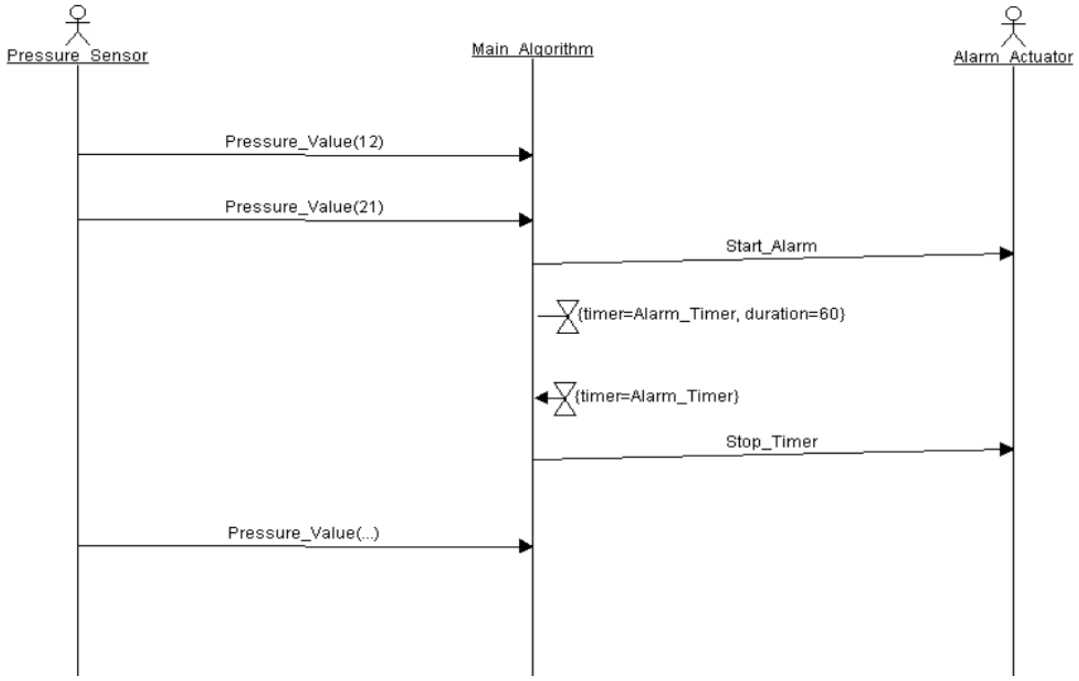
a. Use Case Diagram



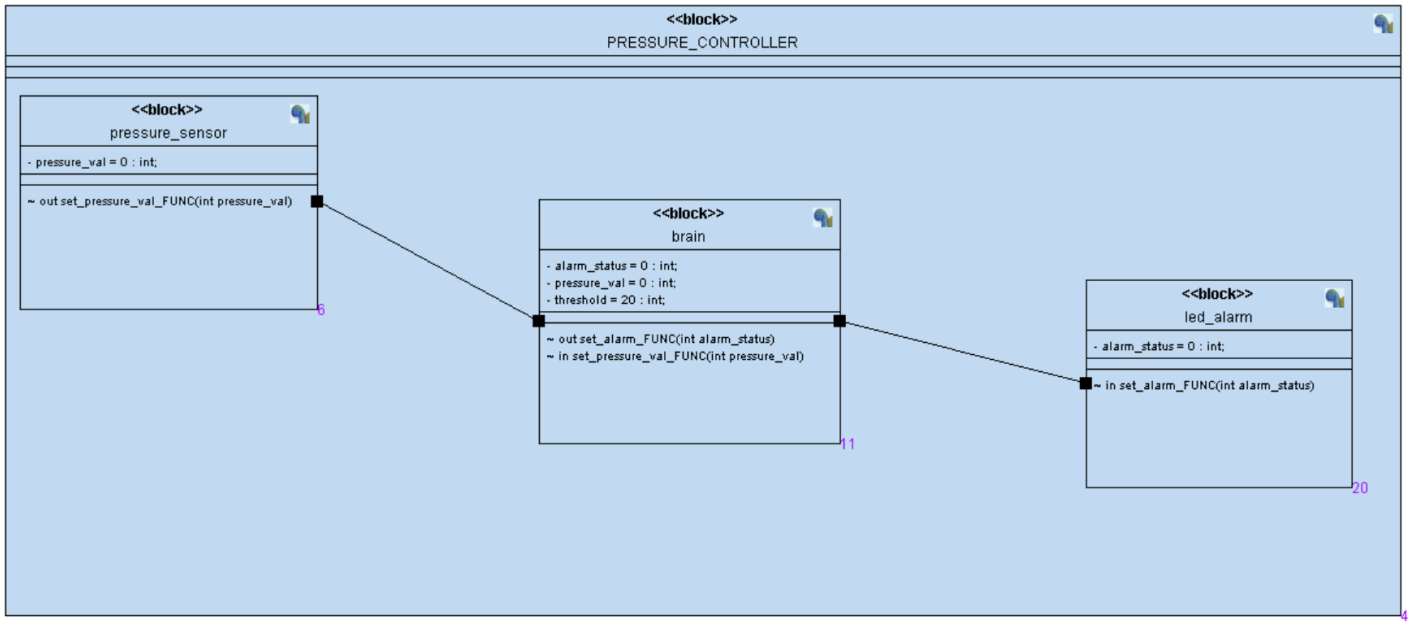
b. Activity diagram



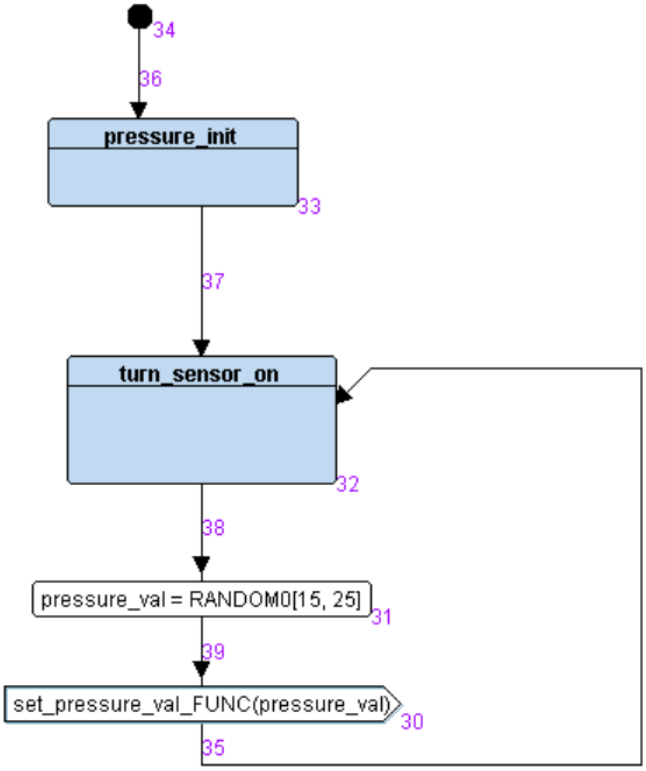
C. sequence diagram



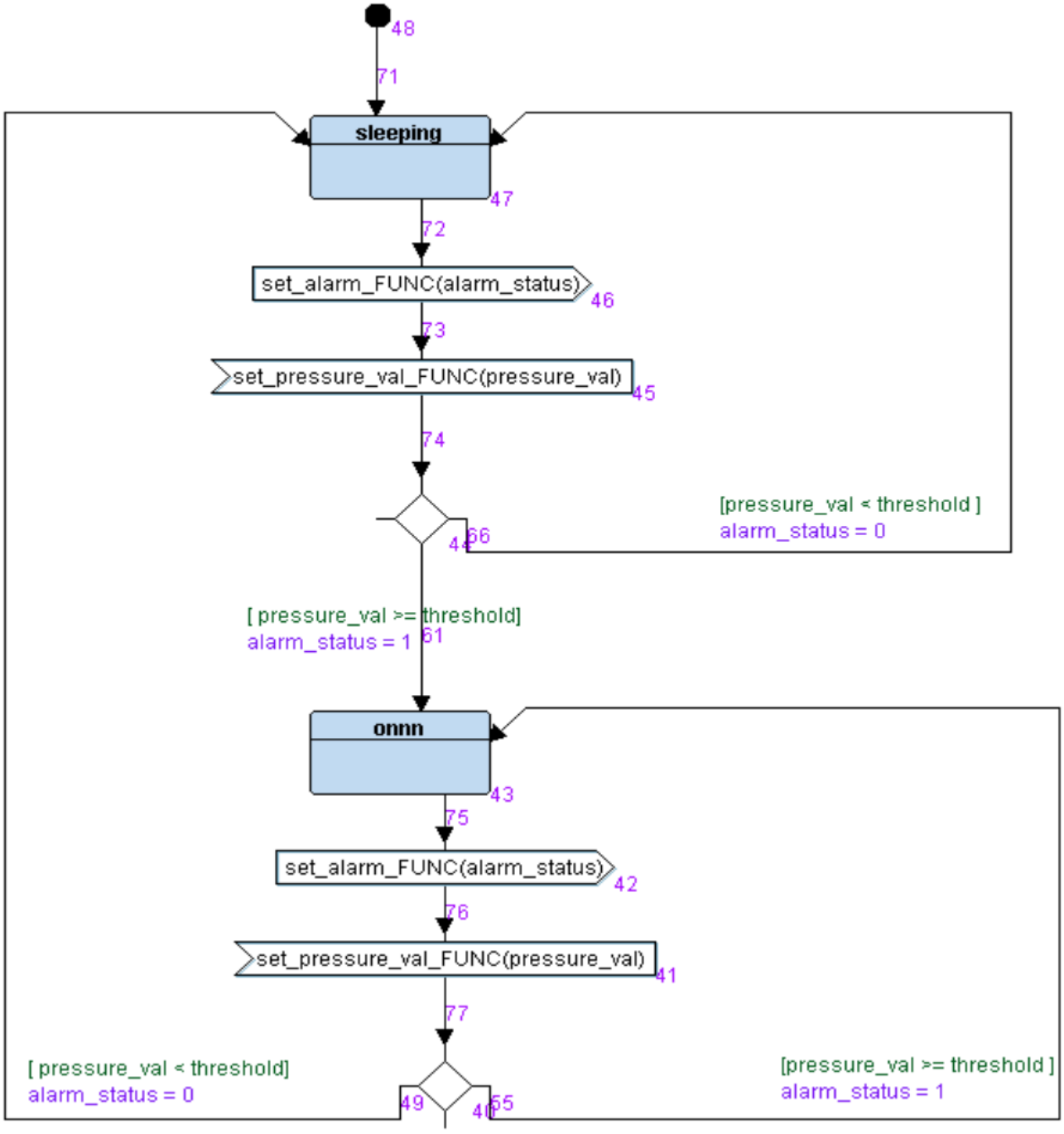
block diagram:



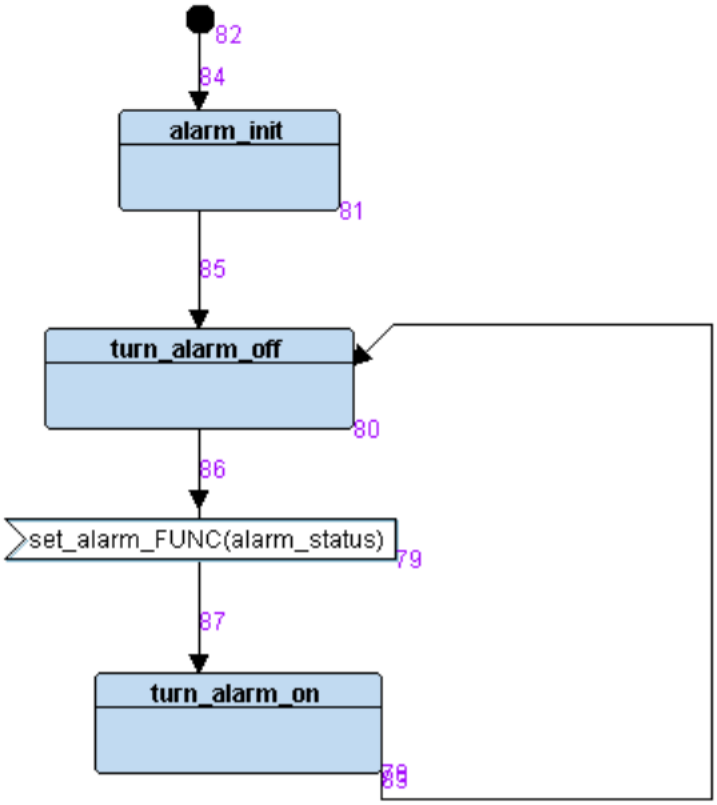
Pressure sensor driver



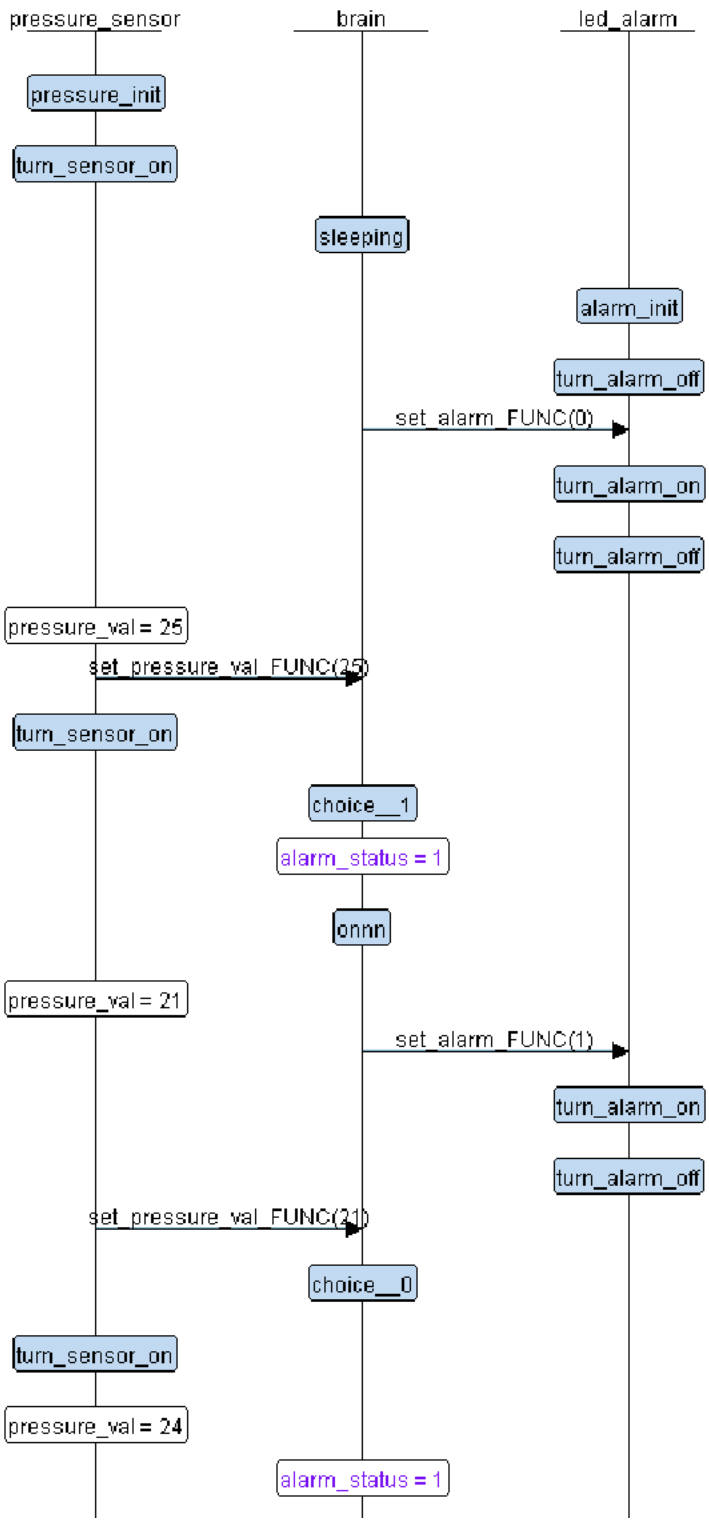
Brain

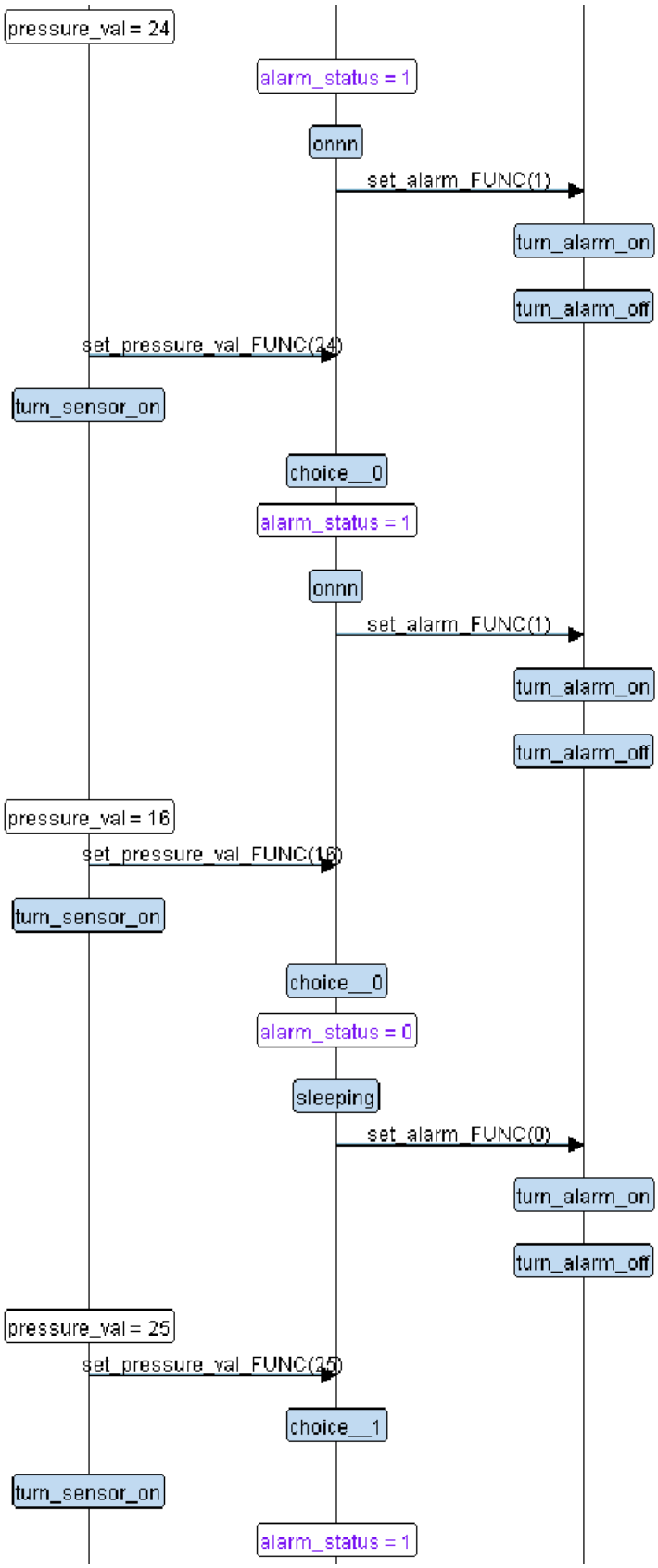

48
71
sleeping
47
72
set_alarm_FUNC(alarm_status)
46
73
set_pressure_val_FUNC(pressure_val)
45
74
44
66
[pressure_val < threshold]
alarm_status = 0
[pressure_val >= threshold]
alarm_status = 1
61
onnn
43
75
set_alarm_FUNC(alarm_status)
42
76
set_pressure_val_FUNC(pressure_val)
41
77
[pressure_val < threshold]
alarm_status = 0
49
40
55
[pressure_val >= threshold]
alarm_status = 1

Led alarm.


82
84
alarm_init
81
85
turn_alarm_off
80
86
set_alarm_FUNC(alarm_status)
79
87
turn_alarm_on

Logic verification

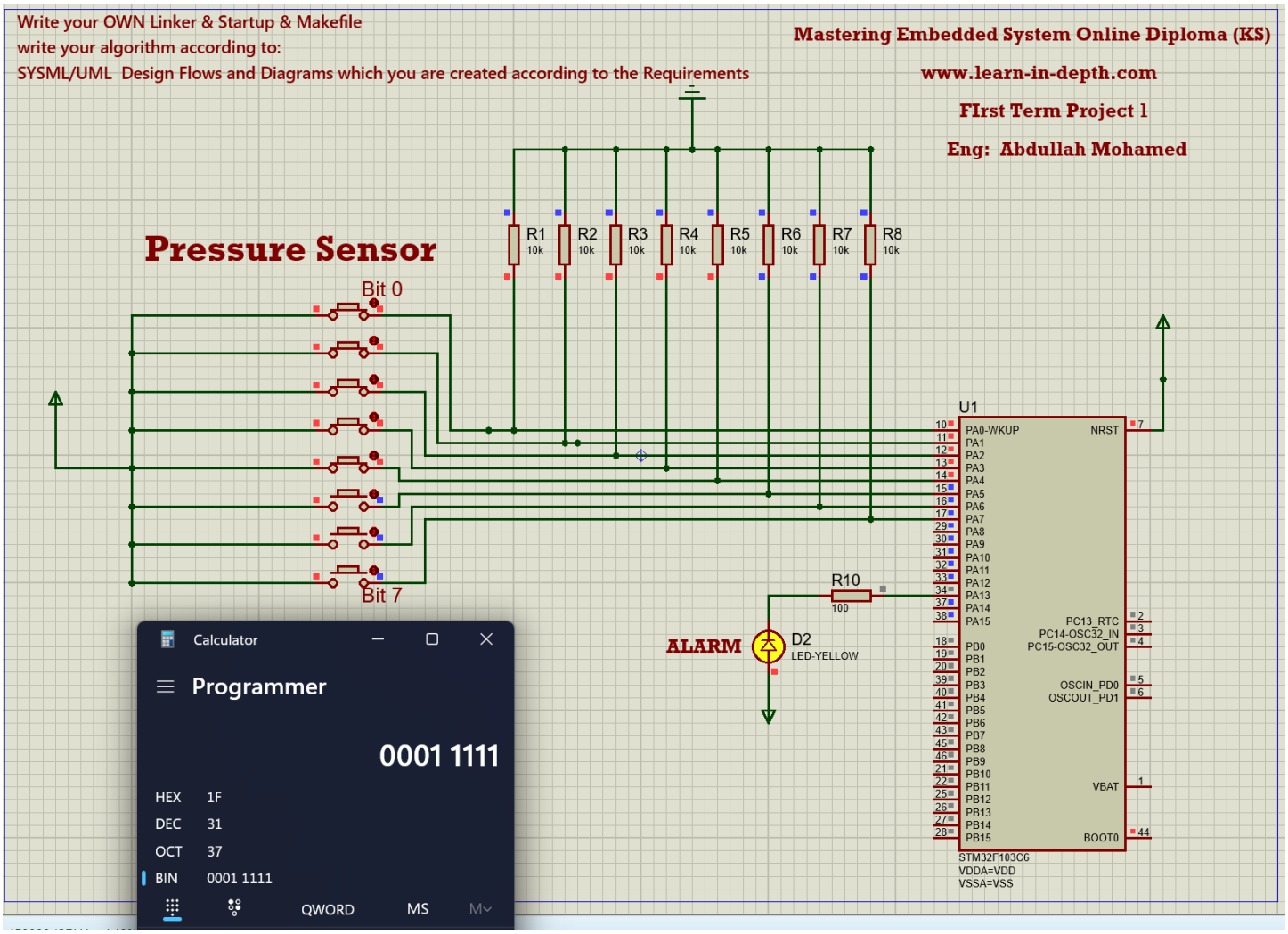



pressure_val = 24
alarm_status = 1
onnn
set_alarm_FUNC(1)
turn_alarm_on
turn_alarm_off
set_pressure_val_FUNC(24)
turn_sensor_on
choice__0
alarm_status = 1
onnn
set_alarm_FUNC(1)
turn_alarm_on
turn_alarm_off
pressure_val = 16
set_pressure_val_FUNC(16)
turn_sensor_on
choice__0
alarm_status = 0
sleeping
set_alarm_FUNC(0)
turn_alarm_on
turn_alarm_off
pressure_val = 25
set_pressure_val_FUNC(25)
choice__1
turn_sensor_on
alarm_status = 1

Proteus simulation

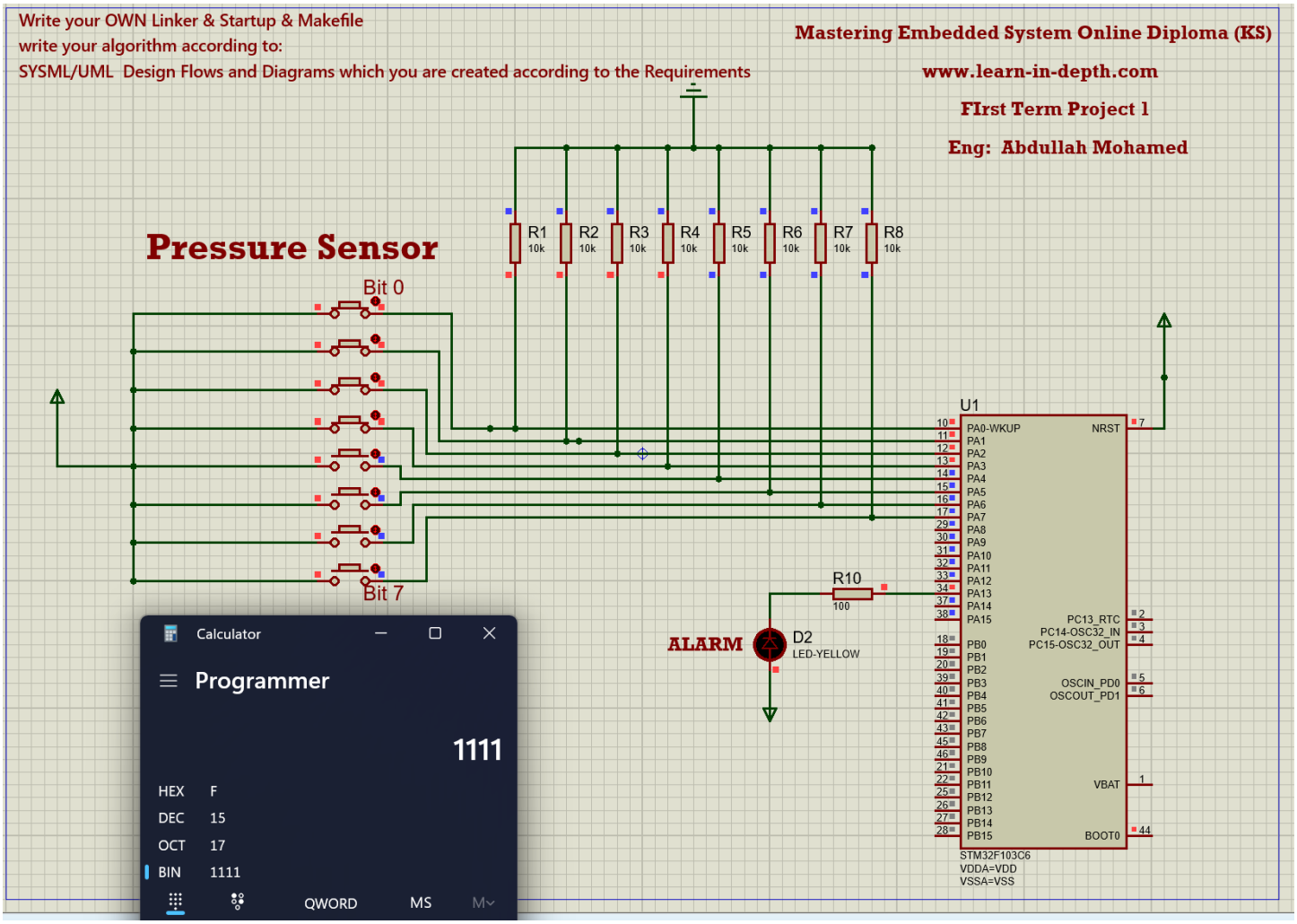
Case 1:

Reading 37 bar.... is greater than “threshold (20 bar)”



Case 2:

Reading 15 bar.... is smaller than “threshold (20 bar)”



Symbols:

```
Abdullah Mohamed@DESKTOP-SV6FTHD MINGW64
$ arm-none-eabi-nm.exe pressure_sensor.o
         U getPressureVal
00000000 T pressure_init_FUNC
00000001 C preusser_current_state
00000004 C PS_p_to_func
00000000 B PS_pressure_val
         U set_pressure_connection_FUNC
0000001c T turn_sensor_on_FUNC

Abdullah Mohamed@DESKTOP-SV6FTHD MINGW64
$ arm-none-eabi-nm.exe led_alarm.o
00000001 C alarm_current_state
00000040 T alarm_init_FUNC
00000004 C alarm_p_to_func
         U Delay
00000000 B led_alarm
         U Set_Alarm_actuator
00000000 T Set_Alarm_connection_FUNC
0000005c T turn_alarm_off_FUNC
00000078 T turn_alarm_on_FUNC

Abdullah Mohamed@DESKTOP-SV6FTHD MINGW64
$ arm-none-eabi-nm.exe brain.o
00000004 B brain_alarm_status
00000001 C brain_current_state
00000004 C brain_p_to_func
00000000 B brain_pressure_val
00000000 D brain_threshold
00000078 T onnn_FUNC
         U Set_Alarm_connection_FUNC
00000000 T set_pressure_connection_FUNC
0000005c T sleeping_FUNC
```

```
Abdullah Mohamed@DESKTOP-SV6FTHD MINGW64
$ arm-none-eabi-nm.exe driver.o
00000000 T Delay
00000024 T getPressureVal
0000008c T GPIO_INITIALIZATION
0000003c T Set_Alarm_actuator

Abdullah Mohamed@DESKTOP-SV6FTHD MINGW64
$ arm-none-eabi-nm.exe main.o
00000001 C alarm_current_state
         U alarm_init_FUNC
         U alarm_p_to_func
00000001 C brain_current_state
         U brain_p_to_func
         U GPIO_INITIALIZATION
00000020 T main
         U pressure_init_FUNC
00000001 C preusser_current_state
         U PS_p_to_func
00000000 T setup
         U sleeping_FUNC

Abdullah Mohamed@DESKTOP-SV6FTHD MINGW64
$ arm-none-eabi-nm.exe startup.o
         U _E_BSS
         U _E_DATA
         U _E_TEXT
         U _S_BSS
         U _S_DATA
00000000 W Bus_fault
00000000 T Default_Handler
00000000 W H_fault_Handler
         U main
00000000 W MM_fault_Handler
00000000 W NMI_Handler
0000000c T Rest_Handler
         U stack_top
00000000 W Usage_fault_Handler
00000000 D vectors
```