Pressure Controller

Mastering Embedded System Online Diploma

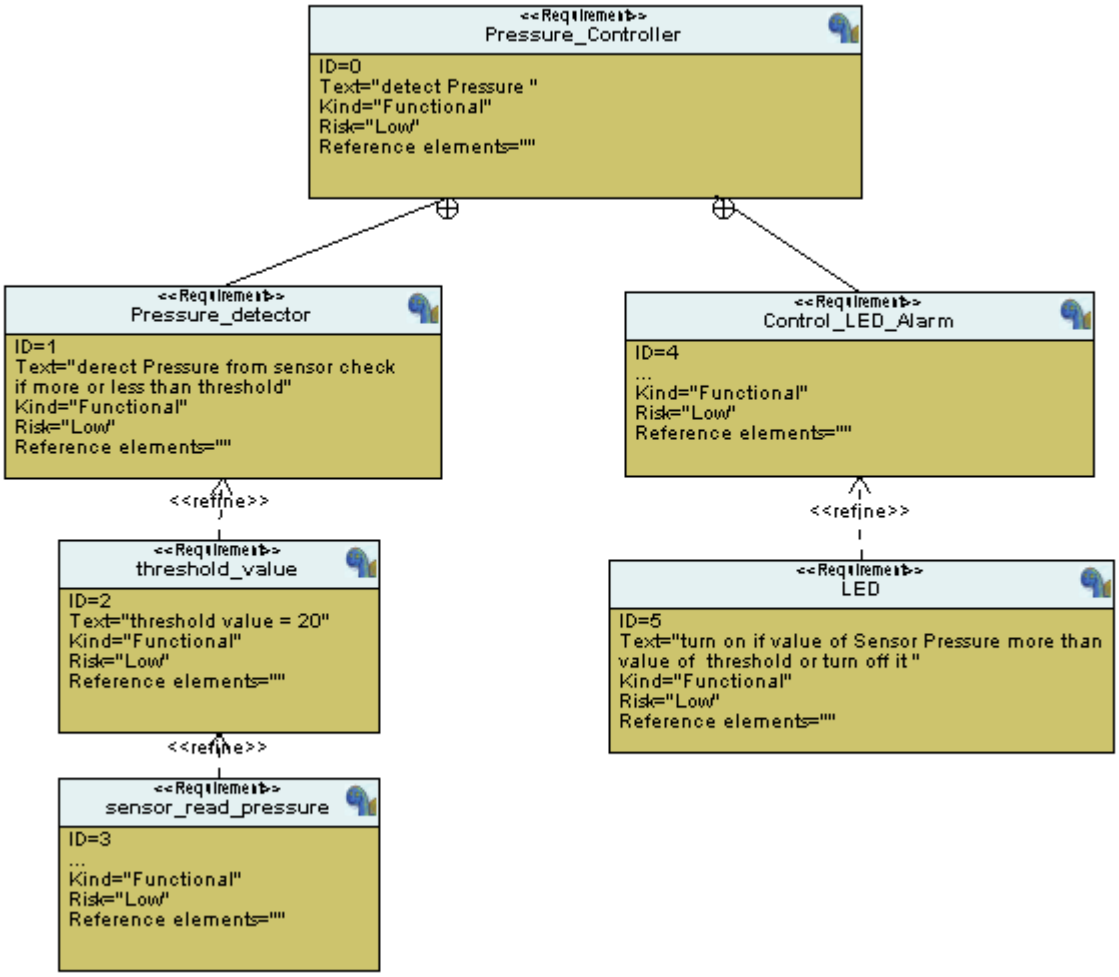
www.learn-in-depth.com

First Term (Final Project 1)

Eng. Omar Adel Shalaan

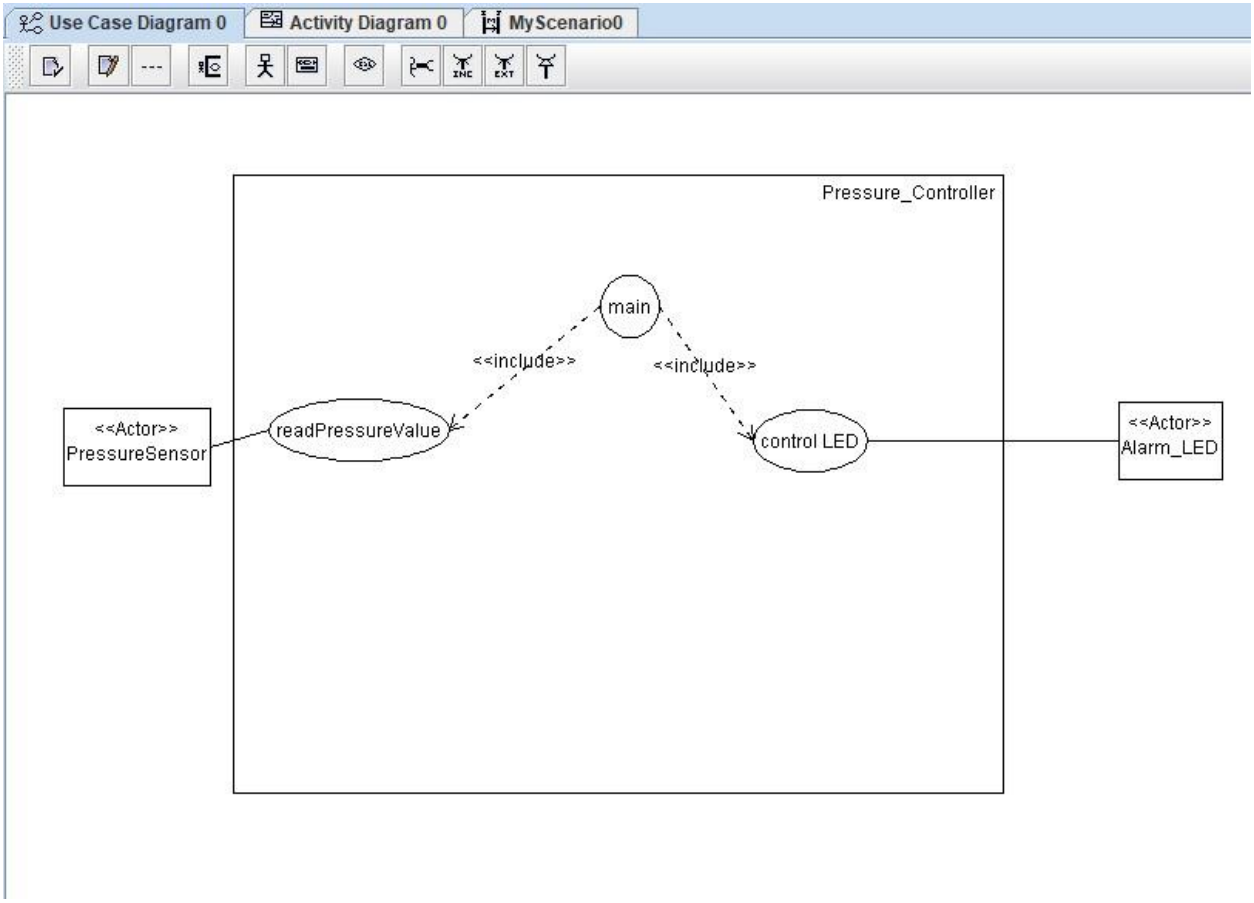
My Profile:

Requirements Diagram

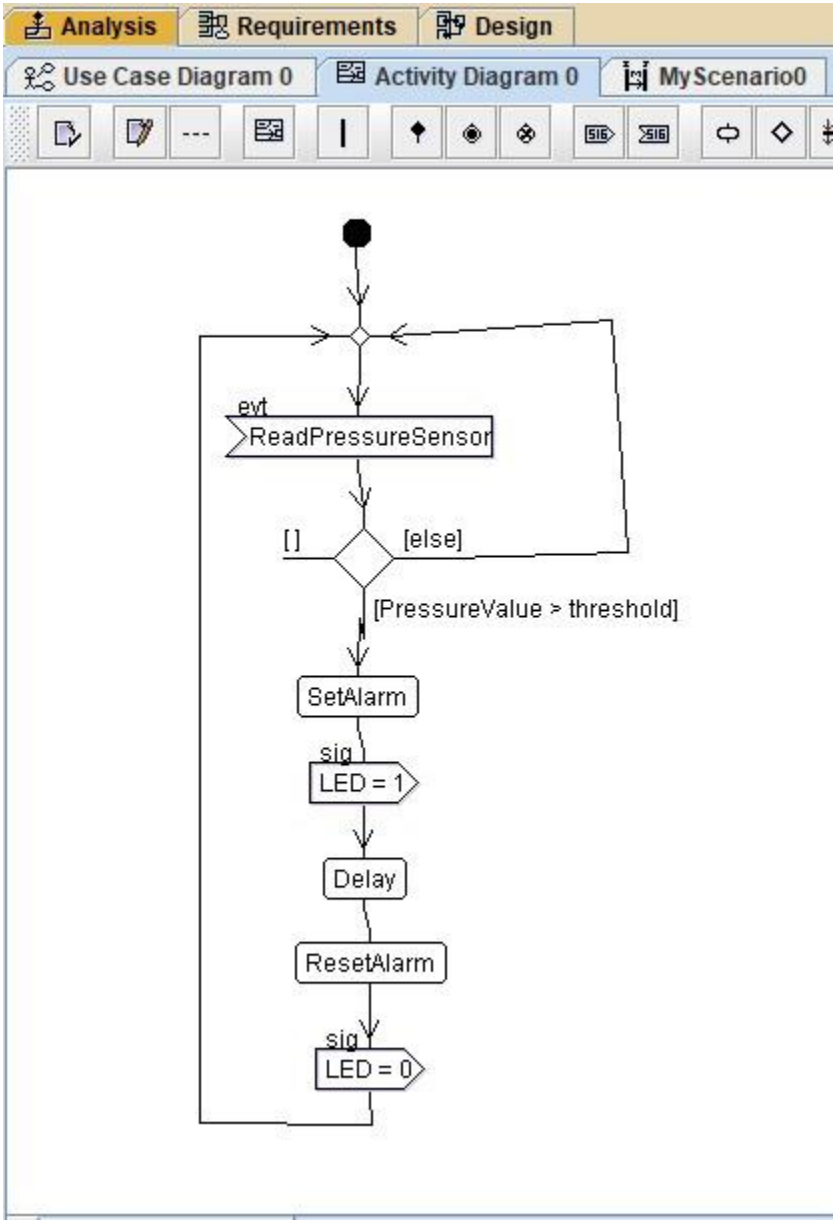


System Analysis

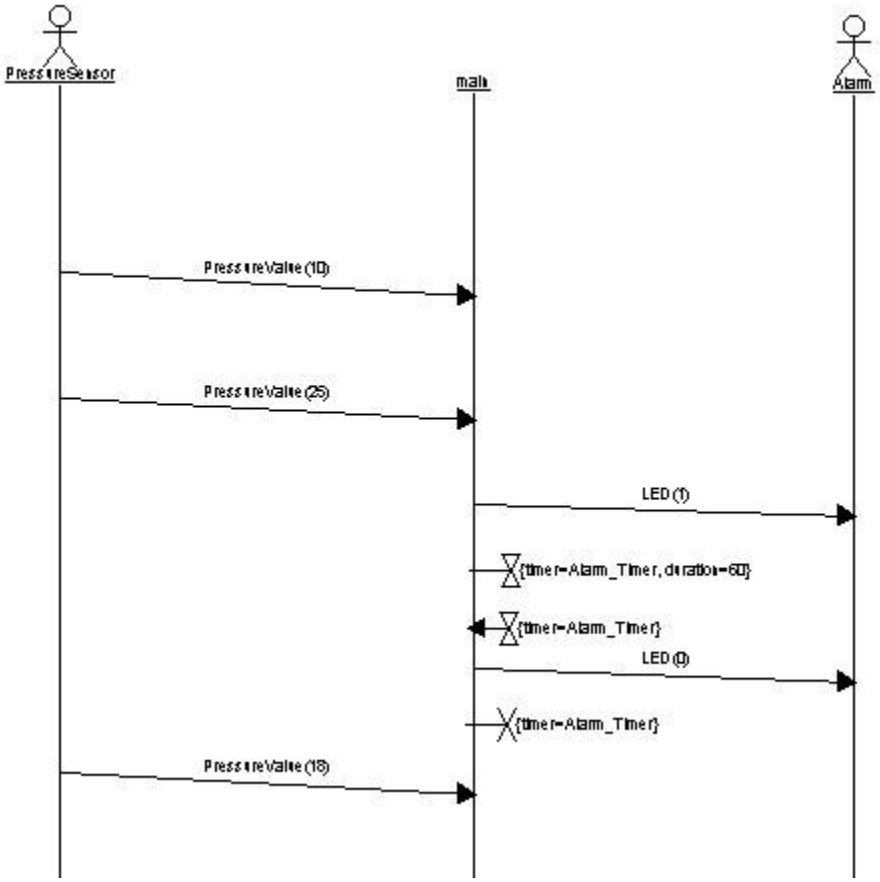
Case Diagram:



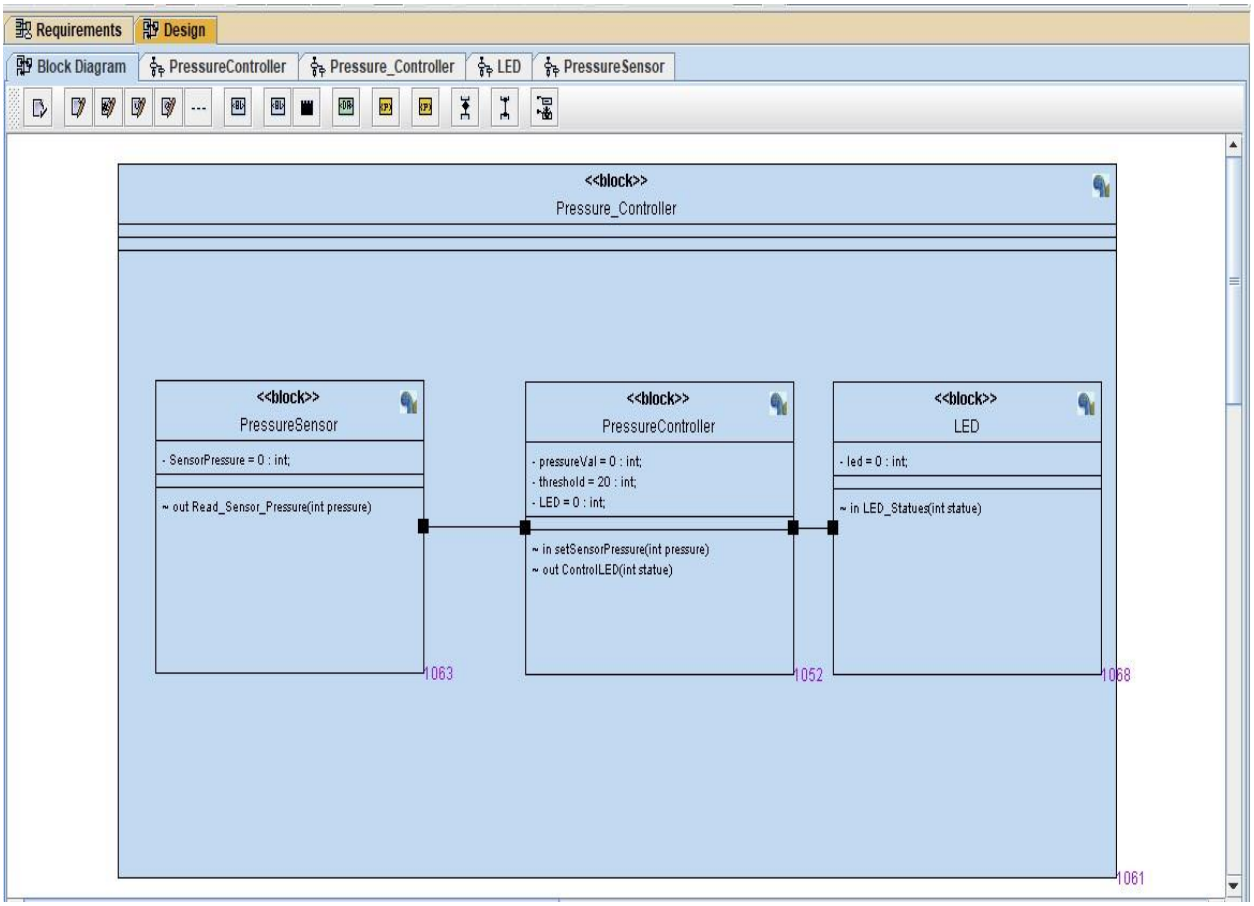
Activity Diagram:

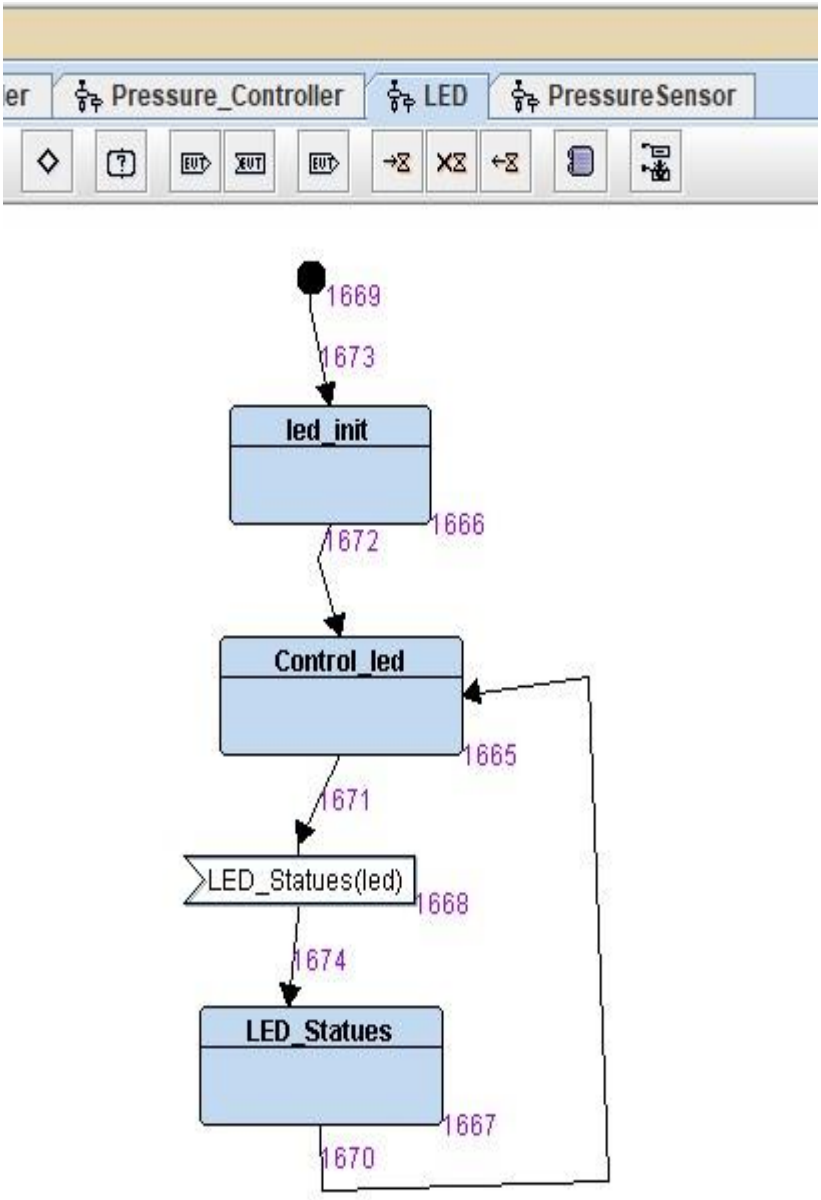


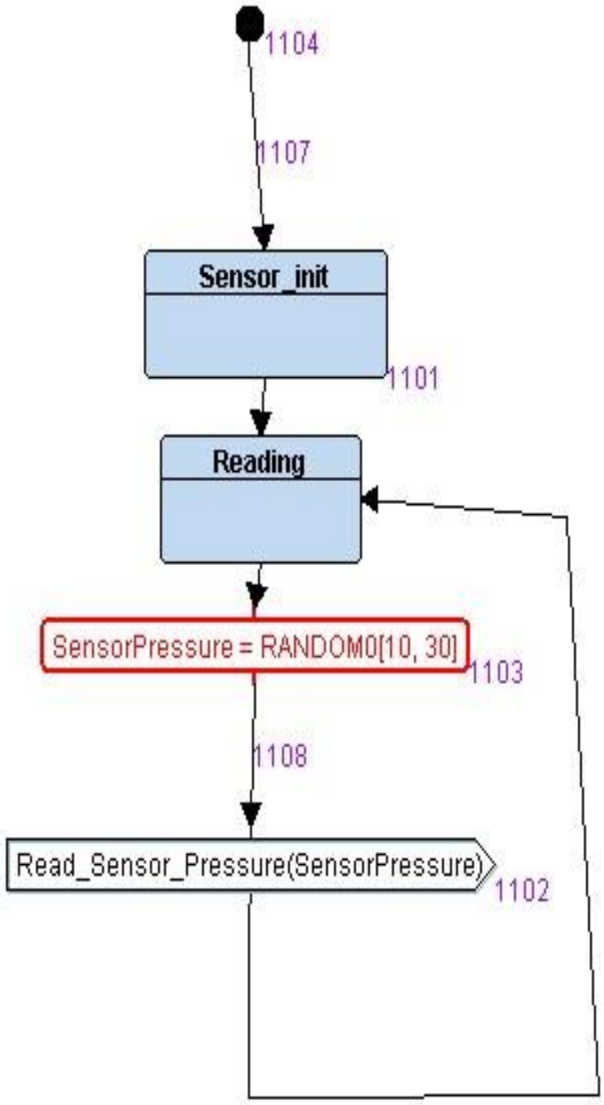
Sequence Diagram:

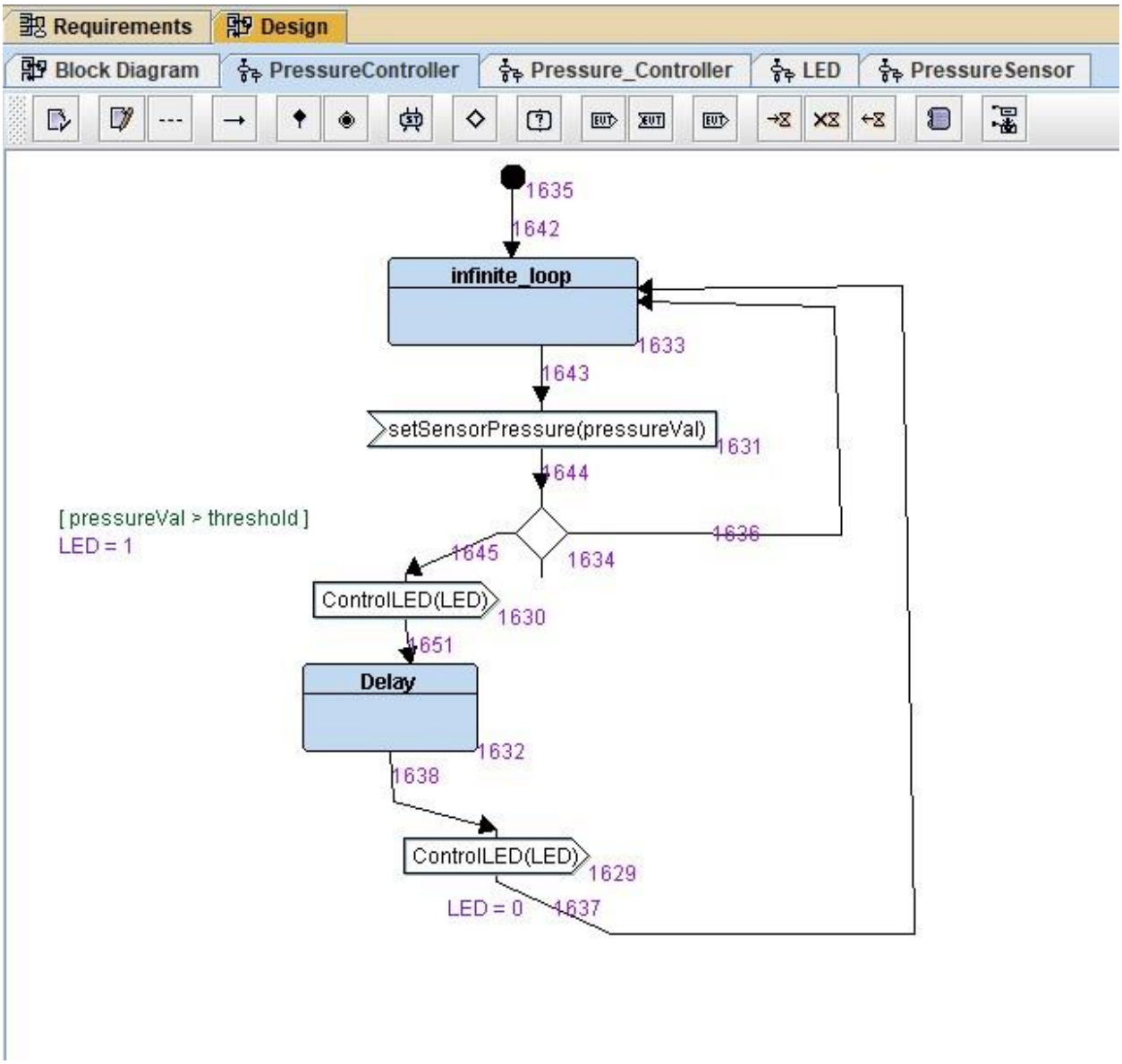


System Design

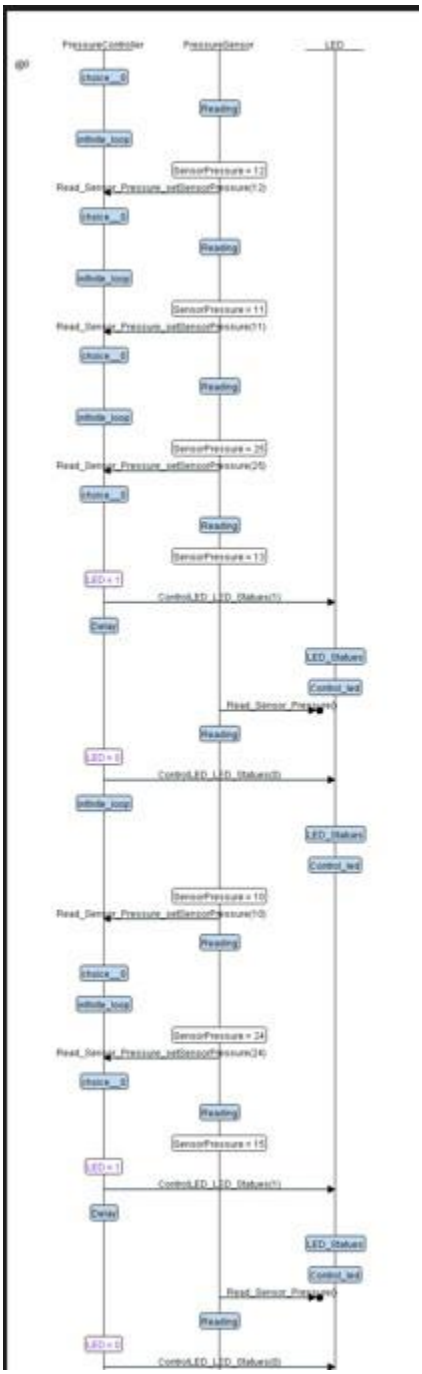



er
Pressure_Controller
LED
PressureSensor
1669
1673
led_init
1666
1672
Control_led
1665
1671
LED_Statues(led)
1668
1674
LED_Statues
1667
1670


1104
1107
Sensor_init
1101
Reading
SensorPressure = RANDOM0[10, 30]
1103
1108
Read_Sensor_Pressure(SensorPressure)
1102


Requirements
Design
Block Diagram
PressureController
Pressure_Controller
LED
PressureSensor
1635
1642
infinite_loop
1633
1643
setSensorPressure(pressureVal)
1631
1644
[pressureVal > threshold]
LED = 1
1636
1645
1634
ControlLED(LED)
1630
1651
Delay
1632
1638
ControlLED(LED)
1629
LED = 0
1637

Sequence Diagram



Sections

```
(===========objdump===========)
arm-none-eabi-objdump  -h Pressure_Controller.elf

Pressure_Controller.elf:     file format elf32-littlearm

Sections:
Idx Name          Size      VMA       LMA       File off  Algn
  0 .text         00000248  08000000  08000000  00008000  2**2
                  CONTENTS, ALLOC, LOAD, READONLY, CODE
  1 .bss          00001004  20000000  08000248  00010000  2**0
                  ALLOC
  2 .debug_info   000003b3  00000000  00000000  00008248  2**0
                  CONTENTS, READONLY, DEBUGGING
  3 .debug_abbrev 000001f0  00000000  00000000  000085fb  2**0
                  CONTENTS, READONLY, DEBUGGING
  4 .debug_loc    000001b0  00000000  00000000  000087eb  2**0
                  CONTENTS, READONLY, DEBUGGING
  5 .debug_aranges 00000080  00000000  00000000  0000899b  2**0
                  CONTENTS, READONLY, DEBUGGING
  6 .debug_line   000001a8  00000000  00000000  00008a1b  2**0
                  CONTENTS, READONLY, DEBUGGING
  7 .debug_str    000001a1  00000000  00000000  00008bc3  2**0
                  CONTENTS, READONLY, DEBUGGING
  8 .comment      00000011  00000000  00000000  00008d64  2**0
                  CONTENTS, READONLY
  9 .ARM.attributes 00000033  00000000  00000000  00008d75  2**0
                  CONTENTS, READONLY
 10 .debug_frame  00000138  00000000  00000000  00008da8  2**2
                  CONTENTS, READONLY, DEBUGGING
(============================)

D:\Learn In Depth\Projects\Pressure_Controller\Code>
```

Symbols

```
D:\Learn In Depth\Projects\Pressure_Controller\Code>
D:\Learn In Depth\Projects\Pressure_Controller\Code>
D:\Learn In Depth\Projects\Pressure_Controller\Code>make Symbols
(===========Symbol===========)
arm-none-eabi-nm Pressure_Controller.elf
20000004 B _E_Bss
20000000 T _E_Data
08000248 T _E_Text
20000000 B _S_Bss
20000000 T _S_Data
20001004 B _stack_top
0800001c T APP_init
08000030 T APP_start
08000184 W Bus_handler
08000184 T Default_Handler
08000070 T Delay
08000094 T getPressureVal
080000fc T GPIO_INITIALIZATION
08000184 W H_fault_Handler
08000174 T main
08000184 W MM_Fault_Handler
08000184 W NMI_Handler
20000000 B pressureVal_g
08000190 T Reset_handler
080000ac T Set_Alarm_actuator
08000184 W Usage_Fault_Handler
08000000 T vectors
(============================)
```

readelf

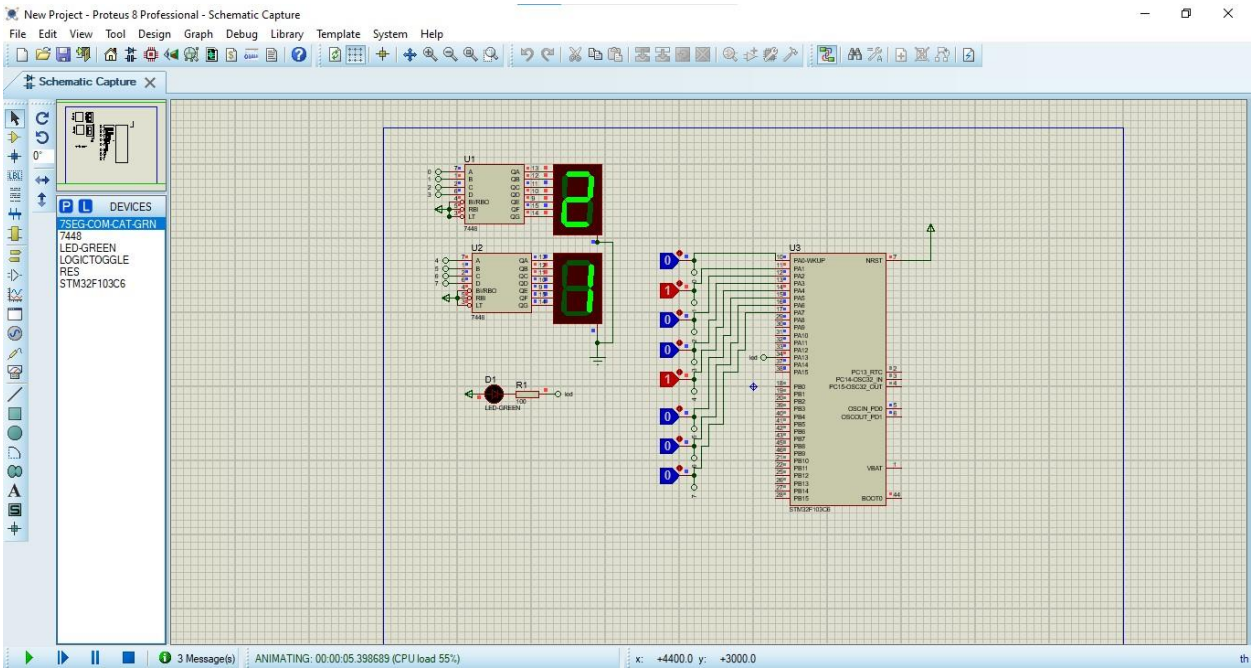
```
D:\Learn In Depth\Projects\Pressure_Controller\Code>make readelf
(==========readelf==========)
arm-none-eabi-readelf -h Pressure_Controller.elf
ELF Header:
  Magic:   7f 45 4c 46 01 01 01 00 00 00 00 00 00 00 00 00
  Class:                             ELF32
  Data:                              2's complement, little endian
  Version:                           1 (current)
  OS/ABI:                            UNIX - System V
  ABI Version:                       0
  Type:                              EXEC (Executable file)
  Machine:                           ARM
  Version:                           0x1
  Entry point address:               0x8000000
  Start of program headers:          52 (bytes into file)
  Start of section headers:          36728 (bytes into file)
  Flags:                             0x5000002, has entry point, Version5 
  Size of this header:               52 (bytes)
  Size of program headers:           32 (bytes)
  Number of program headers:         2
  Size of section headers:           40 (bytes)
  Number of section headers:         15
  Section header string table index: 12
(============================)

D:\Learn In Depth\Projects\Pressure_Controller\Code>
```

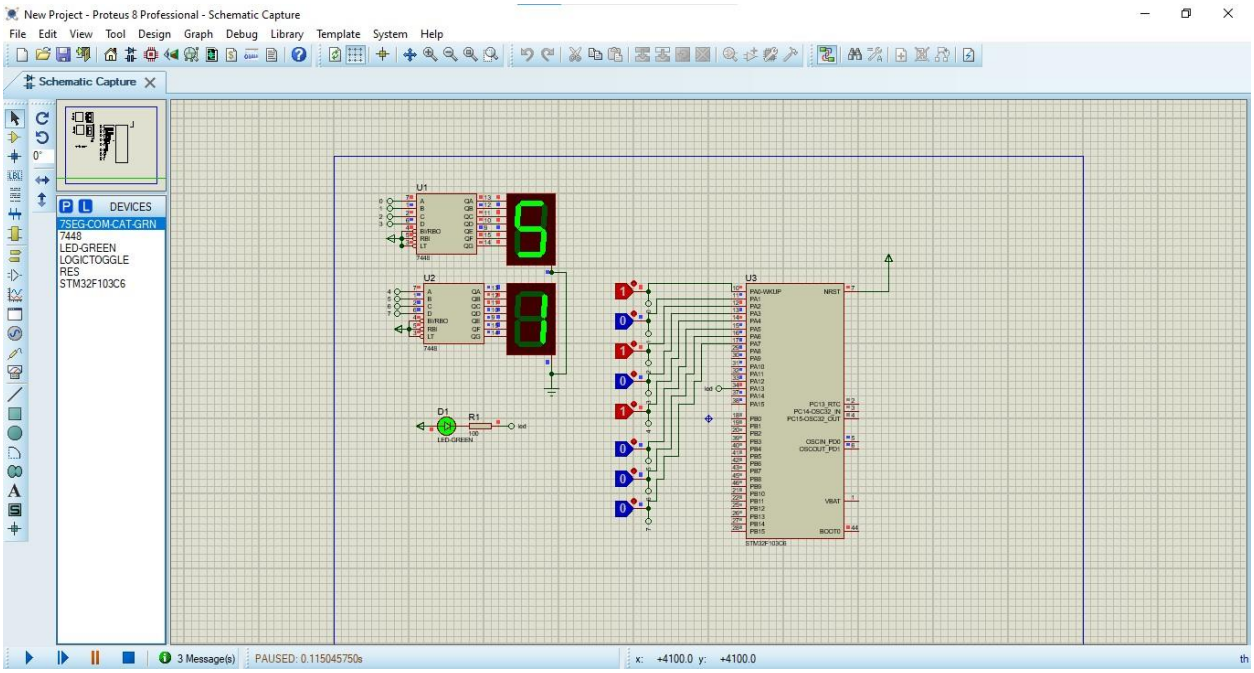
Map file

```
10
11
12  .text           0x08000000      0x248
13   *(.vectors*)
14   .vectors        0x08000000       0x1c Startup.o
15                   0x08000000                vectors
16   *(.text*)
17   .text           0x0800001c       0x54 app.o
18                   0x0800001c                APP_init
19                   0x08000030                APP_start
20   .text           0x08000070      0x104 driver.o
21                   0x08000070                Delay
22                   0x08000094                getPressureVal
23                   0x080000ac                Set_Alarm_actuator
24                   0x080000fc                GPIO_INITIALIZATION
25   .text           0x08000174       0x10 main.o
26                   0x08000174                main
27   .text           0x08000184       0xc4 Startup.o
28                   0x08000184                NMI_Handler
29                   0x08000184                MM_Fault_Handler
30                   0x08000184                Usage_Fault_Handler
31                   0x08000184                Default_Handler
32                   0x08000184                H_fault_Handler
33                   0x08000184                Bus_handler
34                   0x08000190                Reset_handler
35                   0x08000248                . = ALIGN (0x4)
36                   0x08000248                _E_Text = .
37
```

```
55
56  .data           0x20000000        0x0 load address 0x08000248
57                  0x20000000                _S_Data = .
58   *(.data)
59   .data          0x20000000        0x0 app.o
60   .data          0x20000000        0x0 driver.o
61   .data          0x20000000        0x0 main.o
62   .data          0x20000000        0x0 Startup.o
63   *(.rodata)
64                  0x20000000                . = ALIGN (0x4)
65                  0x20000000                _E_Data = .
66
67  .igot.plt       0x20000000        0x0 load address 0x08000248
68   .igot.plt      0x00000000        0x0 app.o
69
70  .bss            0x20000000     0x1004 load address 0x08000248
71                  0x20000000                _S_Bss = .
72   *(.bss)
73   .bss           0x20000000        0x1 app.o
74                  0x20000000                pressureVal_g
75   .bss           0x20000001        0x0 driver.o
76   .bss           0x20000001        0x0 main.o
77   .bss           0x20000001        0x0 Startup.o
78                  0x20000004                . = ALIGN (0x4)
79   *fill*         0x20000001        0x3
80                  0x20000004                _E_Bss = .
81                  0x20001004                . = (. + 0x1000)
82   *fill*         0x20000004     0x1000
83                  0x20001004                _stack_top = .
84  LOAD app.o
85  LOAD driver.o
```

Proteus Run


New Project - Proteus 8 Professional - Schematic Capture
File Edit View Tool Design Graph Debug Library Template System Help
Schematic Capture
DEVICES
7SEG-COM-CAT-GRN
7448
LED-GREEN
LOGICTOGGLE
RES
STM32F103C6
U1
U2
U3
D1
R1
3 Message(s)
ANIMATING: 00:00:05.398689 (CPU load 55%)
x: +4400.0 y: +3000.0


New Project - Proteus 8 Professional - Schematic Capture
File Edit View Tool Design Graph Debug Library Template System Help
Schematic Capture
DEVICES
7SEG-COM-CAT-GRN
7448
LED-GREEN
LOGICTOGGLE
RES
STM32F103C6
U1
U2
U3
D1
R1
3 Message(s)
PAUSED: 0.115045750s
x: +4100.0 y: +4100.0

Proteus Debug