Mastering Embedded Systems Online Diploma

www.learn-in-depth.com

First Term (Final Project 1)

Eng. Ayman Mohamed Elashry

My Profile:

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

The Project contains:

- Case study.
- Method.
- Requirements.
- System Analysis.
- System Design.

First, I will start with the case study.

Case Study

Project Notes for High-Pressure Alarm System Integration

The client has requested the development of a product to assist pilots in the cabin by providing high-pressure alerts through an opening alarm. As part of this project, I am tasked with designing a programmable circuit that will be integrated into the airplane system. However, there are some critical points to consider:

- Controller Activation Signal:

The controller must have an electric signal for activation. Ensure that you can deactivate this model, when necessary, as it is not currently modeled.

- Controller Maintenance:

Maintenance procedures for the controller are not modeled. Before deploying the system model, thoroughly check the pressure sensor's sensitivity and verify the alarm's status.

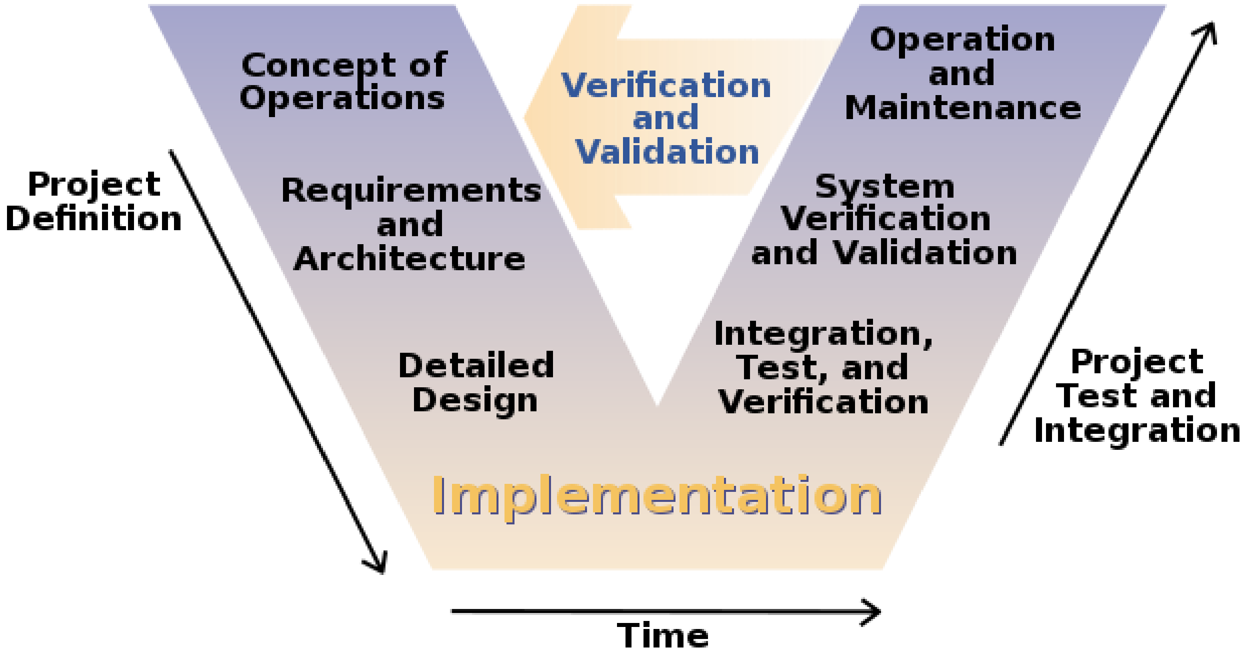
- Power Interruptions:

Avoid power cuts to the system, as they may lead to unexpected behavior.

Please review these notes carefully, as I cannot assume responsibility for any damage incurred during implementation.

Method

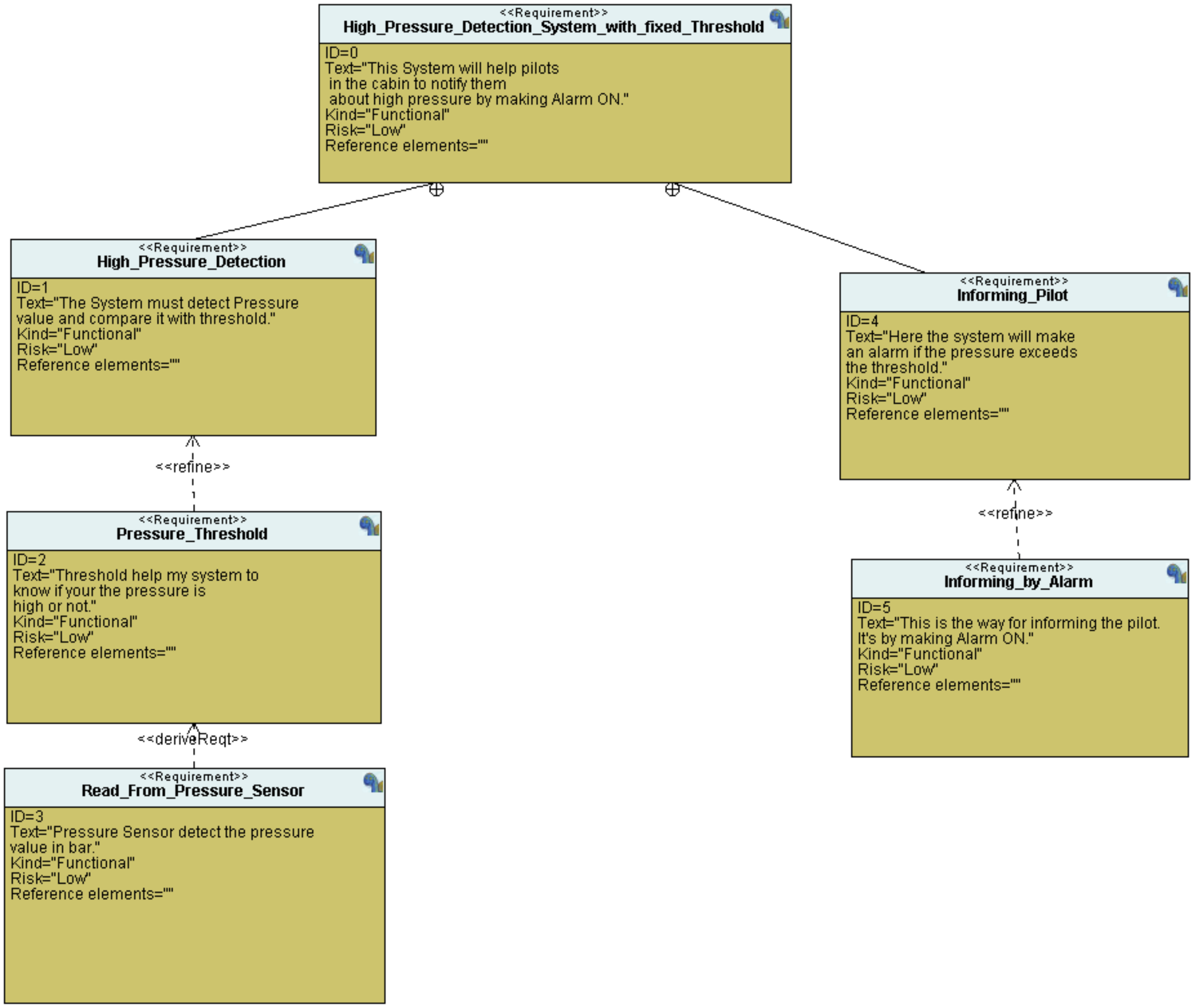
I will choose V Model Software Development


Concept of Operations
Verification and Validation
Operation and Maintenance
Project Definition
Requirements and Architecture
System Verification and Validation
Detailed Design
Integration, Test, and Verification
Project Test and Integration
Implementation
Time

Picture Reference

System Requirements

Here I will make a system requirements diagram.



<<Requirement>>
High_Pressure_Detection_System_with_fixed_Threshold

ID=0
Text="This System will help pilots
in the cabin to notify them
about high pressure by making Alarm ON."
Kind="Functional"
Risk="Low"
Reference elements=""

<<Requirement>>
High_Pressure_Detection

ID=1
Text="The System must detect Pressure
value and compare it with threshold."
Kind="Functional"
Risk="Low"
Reference elements=""

<<Requirement>>
Informing_Pilot

ID=4
Text="Here the system will make
an alarm if the pressure exceeds
the threshold."
Kind="Functional"
Risk="Low"
Reference elements=""

<<Requirement>>
Pressure_Threshold

ID=2
Text="Threshold help my system to
know if your the pressure is
high or not."
Kind="Functional"
Risk="Low"
Reference elements=""

<<Requirement>>
Informing_by_Alarm

ID=5
Text="This is the way for informing the pilot.
It's by making Alarm ON."
Kind="Functional"
Risk="Low"
Reference elements=""

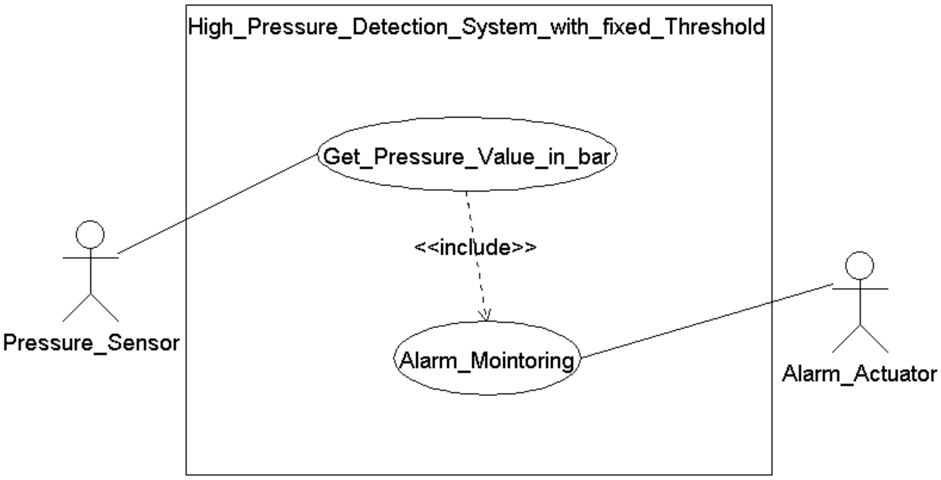
<<Requirement>>
Read_From_Pressure_Sensor

ID=3
Text="Pressure Sensor detect the pressure
value in bar."
Kind="Functional"
Risk="Low"
Reference elements=""

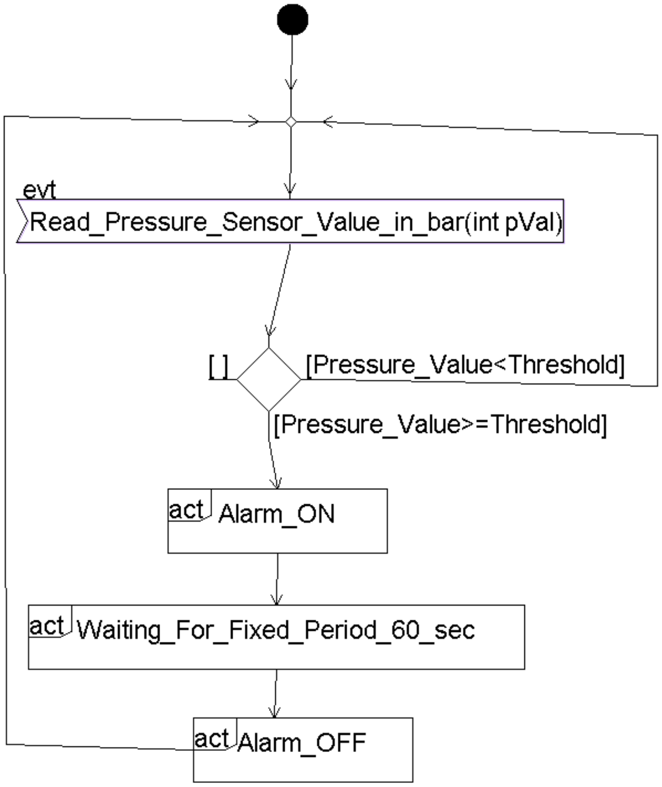
This is my Requirement diagram for the required system.

System Analysis

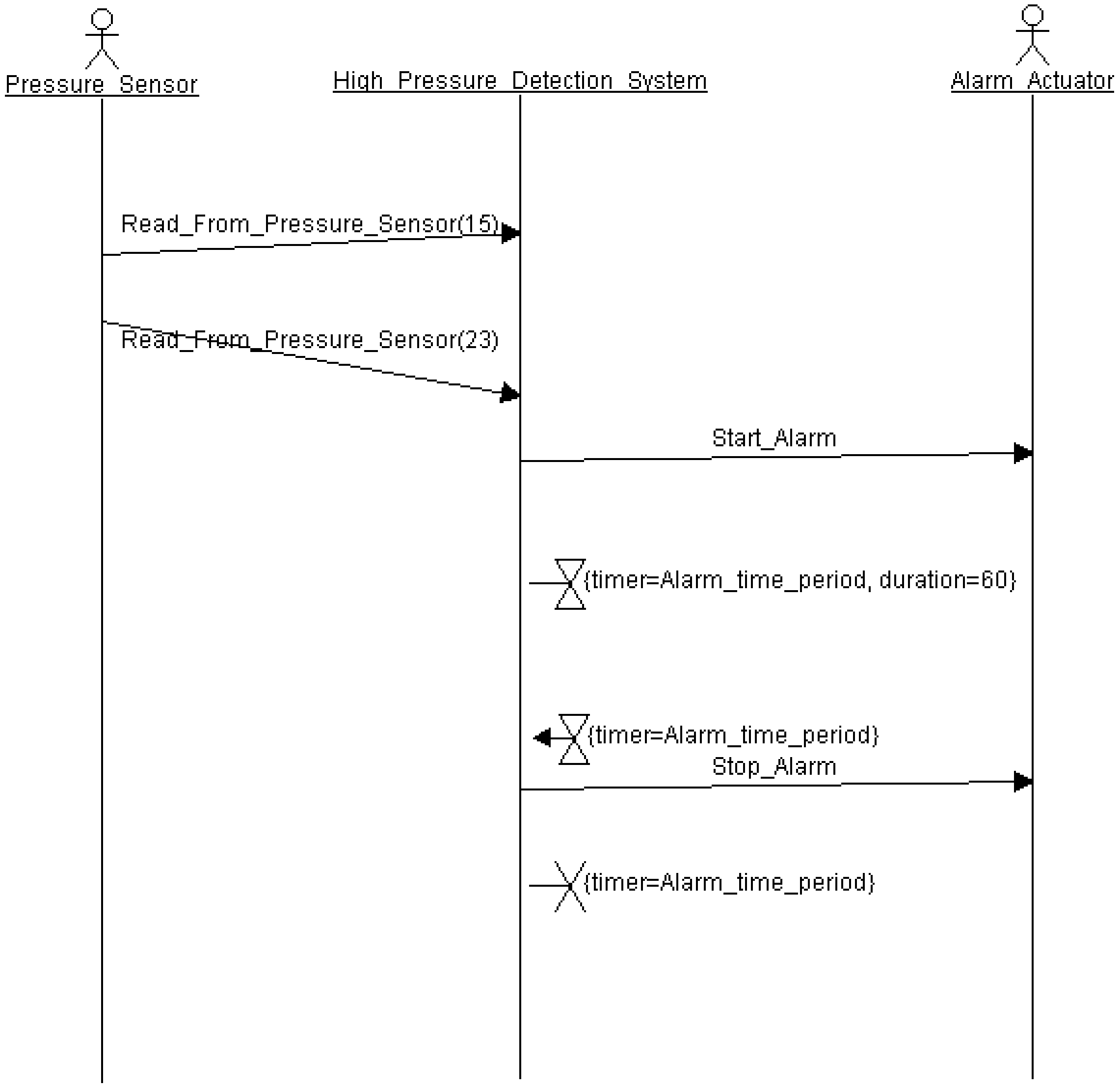
Use Case Diagram:


High_Pressure_Detection_System_with_fixed_Threshold
Get_Pressure_Value_in_bar
<<include>>
Alarm_Mointoring
Pressure_Sensor
Alarm_Actuator

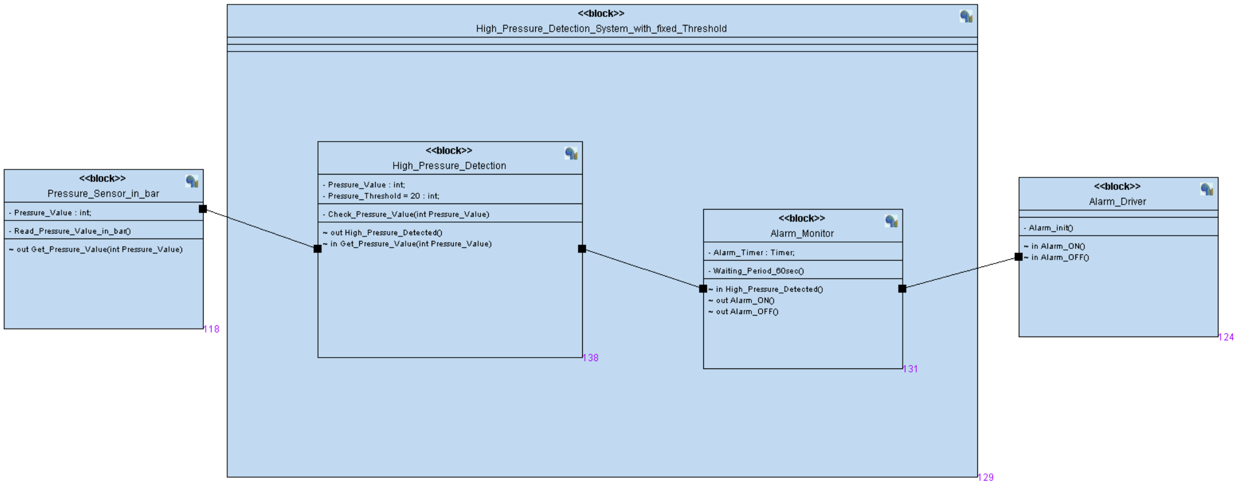
Activity Diagram:

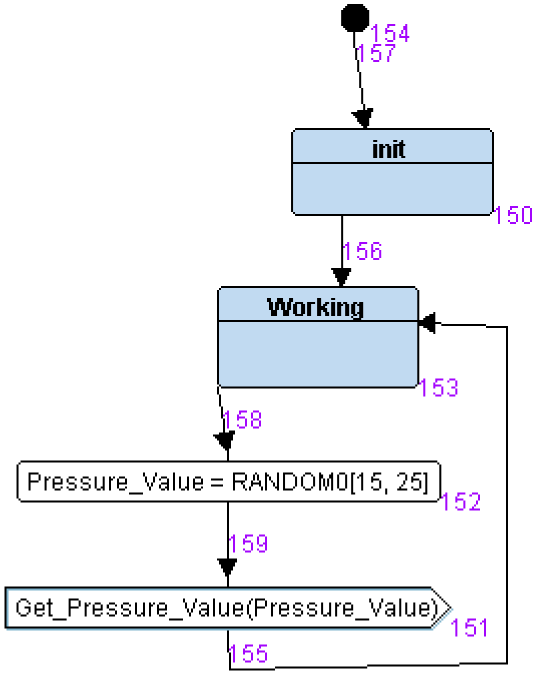

evt
Read_Pressure_Sensor_Value_in_bar(int pVal)
[ ]
[Pressure_Value<Threshold]
[Pressure_Value>=Threshold]
act Alarm_ON
act Waiting_For_Fixed_Period_60_sec
act Alarm_OFF

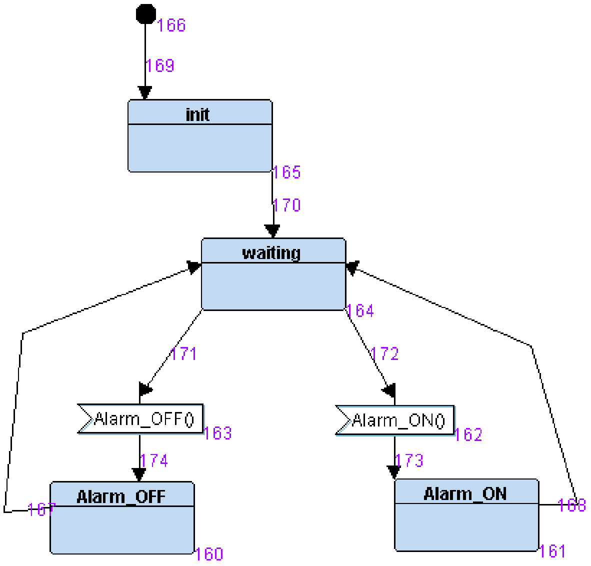
Sequence Diagram:

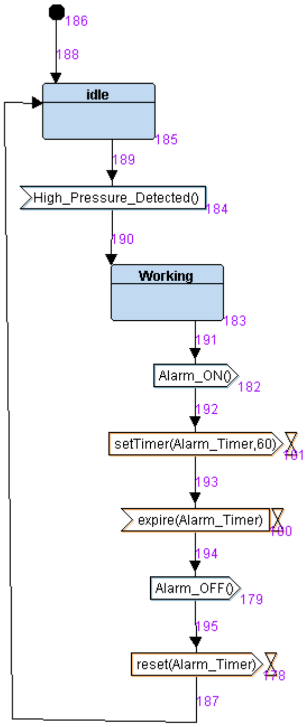

Pressure_Sensor
High_Pressure_Detection_System
Alarm_Actuator
Read_From_Pressure_Sensor(15)
Read_From_Pressure_Sensor(23)
Start_Alarm
{timer=Alarm_time_period, duration=60}
{timer=Alarm_time_period}
Stop_Alarm
{timer=Alarm_time_period}

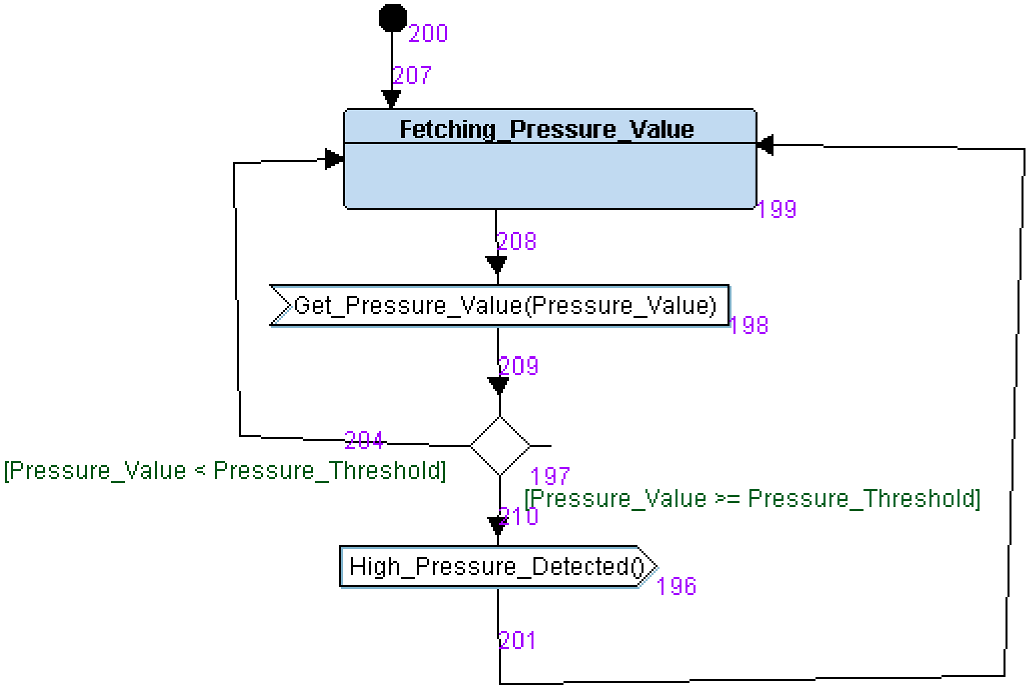
System Design:

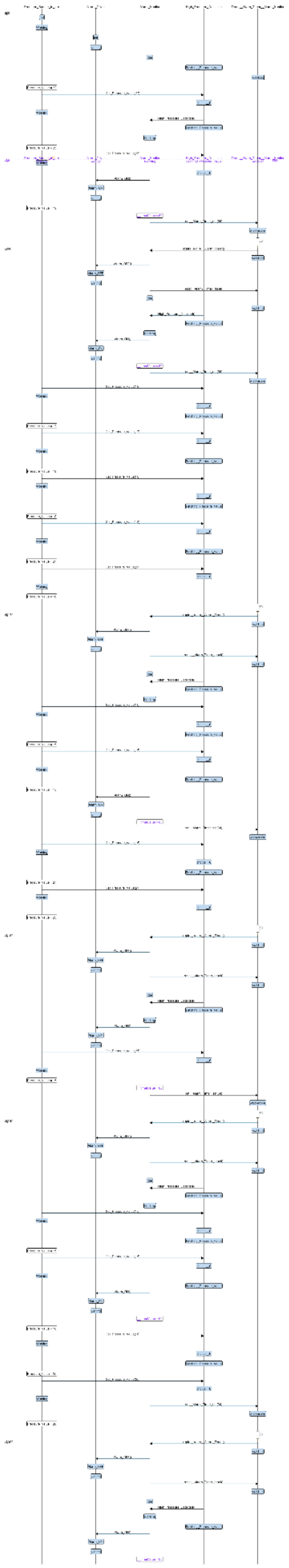

<<block>>
High_Pressure_Detection_System_with_fixed_Threshold
<<block>>
Pressure_Sensor_in_bar
- Pressure_Value : int;
- Read_Pressure_Value_in_bar()
~ out Get_Pressure_Value(int Pressure_Value)
118
<<block>>
High_Pressure_Detection
- Pressure_Value : int;
- Pressure_Threshold = 20 : int;
- Check_Pressure_Value(int Pressure_Value)
~ out High_Pressure_Detected()
~ in Get_Pressure_Value(int Pressure_Value)
138
<<block>>
Alarm_Monitor
- Alarm_Timer : Timer;
- Waiting_Period_60sec()
~ in High_Pressure_Detected()
~ out Alarm_ON()
~ out Alarm_OFF()
131
<<block>>
Alarm_Driver
- Alarm_init()
~ in Alarm_ON()
~ in Alarm_OFF()
124
129


154
157
init
150
156
Working
153
158
Pressure_Value = RANDOM0[15, 25]
152
159
Get_Pressure_Value(Pressure_Value)
151
155


166
169
init
165
170
waiting
164
171
172
Alarm_OFF()
163
Alarm_ON()
162
174
173
Alarm_OFF
167
160
Alarm_ON
168
161

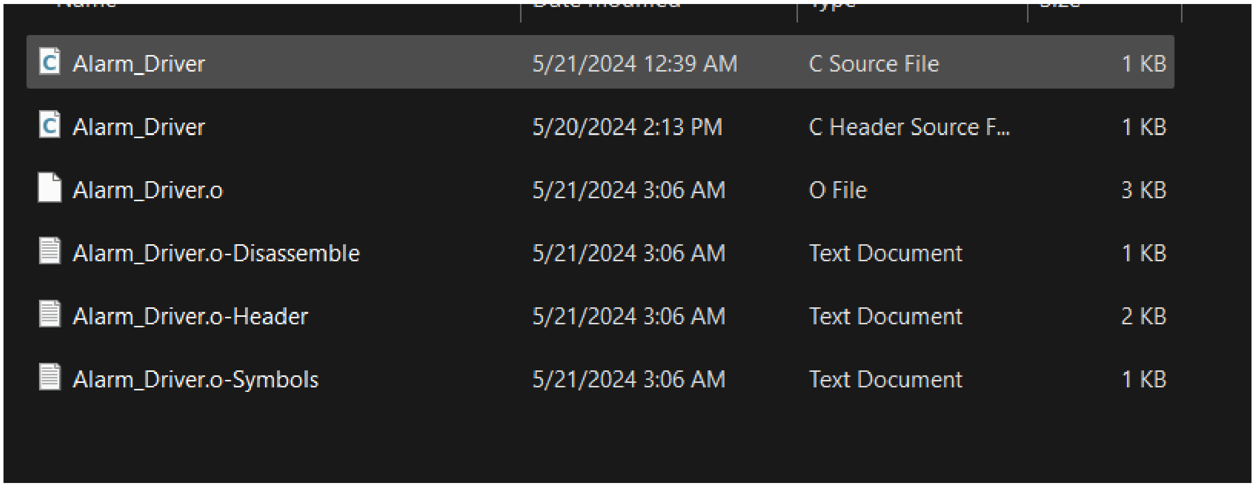

186
188
idle
185
189
High_Pressure_Detected()
184
190
Working
183
191
Alarm_ON()
182
192
setTimer(Alarm_Timer,60)
181
193
expire(Alarm_Timer)
180
194
Alarm_OFF()
179
195
reset(Alarm_Timer)
178
187


200
207
Fetching_Pressure_Value
199
208
Get_Pressure_Value(Pressure_Value)
198
209
204
[Pressure_Value < Pressure_Threshold]
197
[Pressure_Value >= Pressure_Threshold]
210
High_Pressure_Detected()
196
201

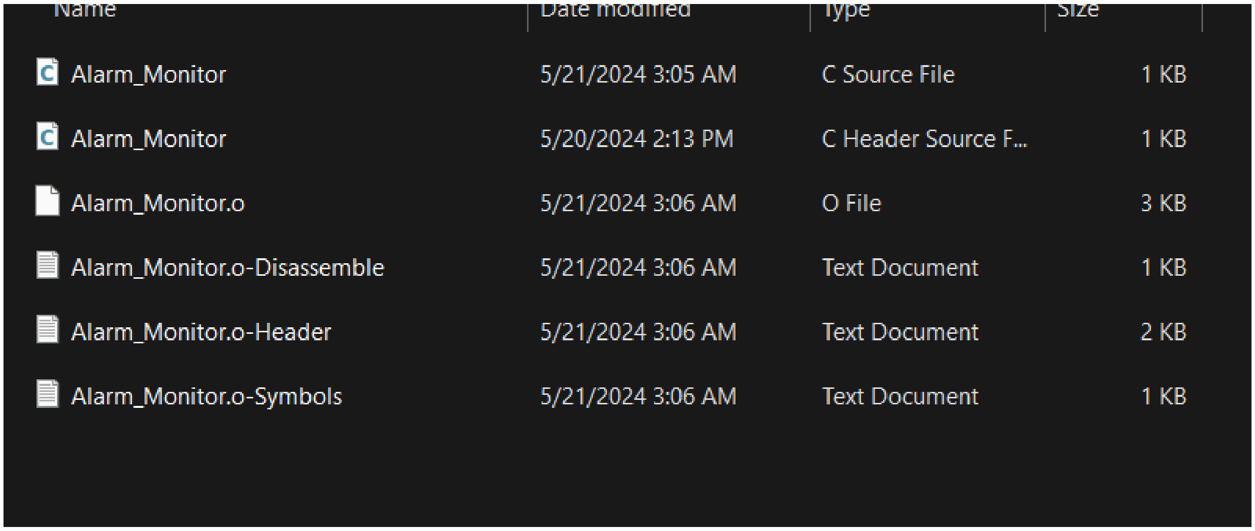


Interactive Simulation

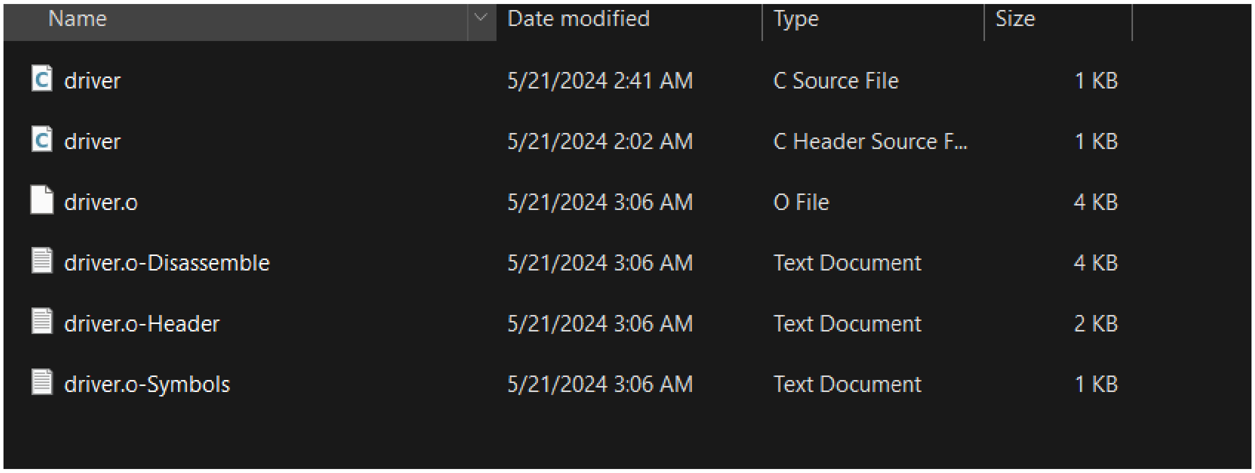
Implementation with Embedded C Programming



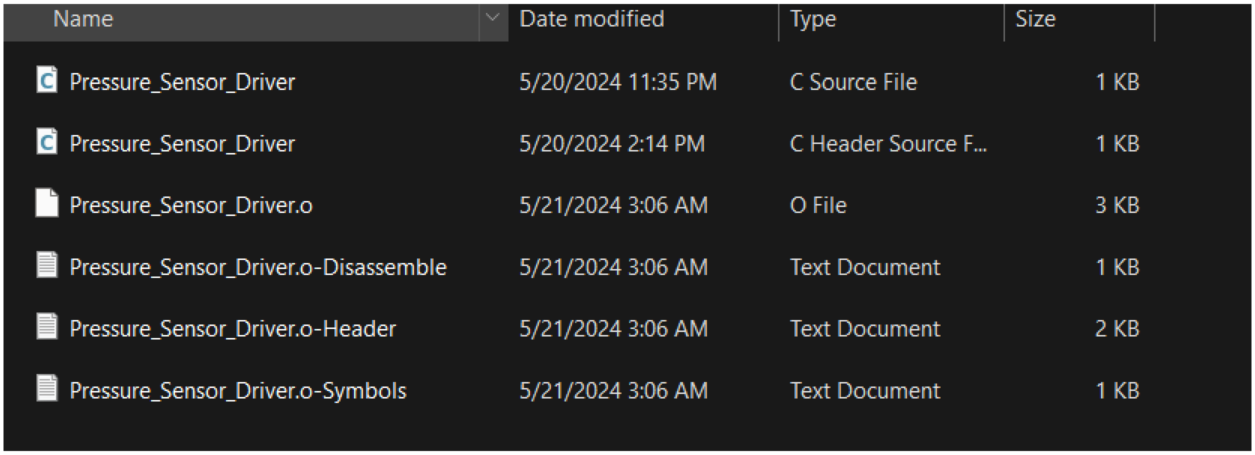
Alarm_Driver	5/21/2024 12:39 AM	C Source File	1 KB
Alarm_Driver	5/20/2024 2:13 PM	C Header Source F...	1 KB
Alarm_Driver.o	5/21/2024 3:06 AM	O File	3 KB
Alarm_Driver.o-Disassemble	5/21/2024 3:06 AM	Text Document	1 KB
Alarm_Driver.o-Header	5/21/2024 3:06 AM	Text Document	2 KB
Alarm_Driver.o-Symbols	5/21/2024 3:06 AM	Text Document	1 KB



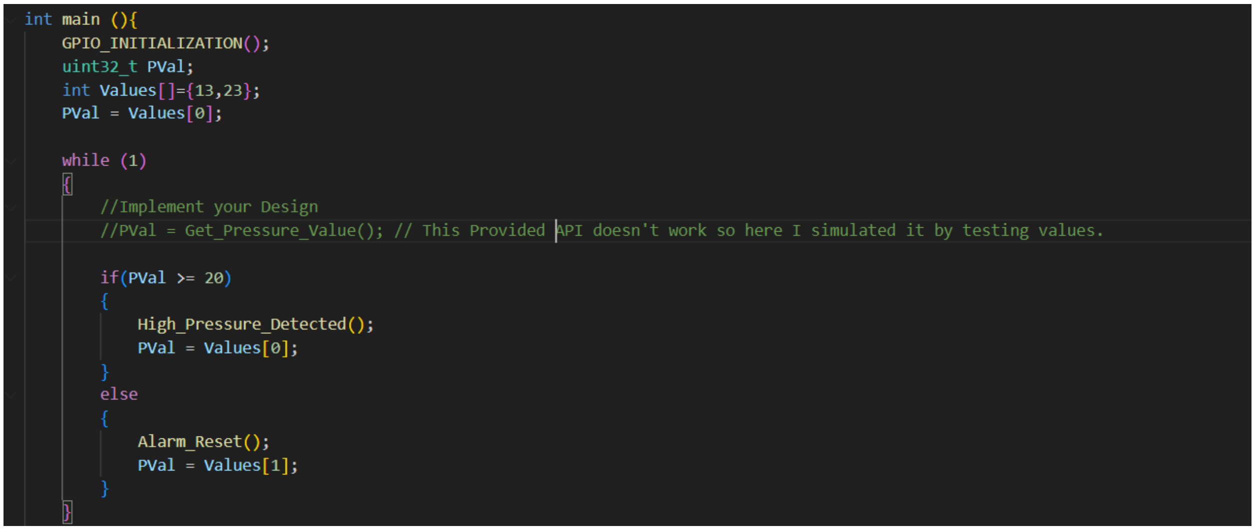
Name	Date modified	Type	Size
Alarm_Monitor	5/21/2024 3:05 AM	C Source File	1 KB
Alarm_Monitor	5/20/2024 2:13 PM	C Header Source F...	1 KB
Alarm_Monitor.o	5/21/2024 3:06 AM	O File	3 KB
Alarm_Monitor.o-Disassemble	5/21/2024 3:06 AM	Text Document	1 KB
Alarm_Monitor.o-Header	5/21/2024 3:06 AM	Text Document	2 KB
Alarm_Monitor.o-Symbols	5/21/2024 3:06 AM	Text Document	1 KB



Name	Date modified	Type	Size
driver	5/21/2024 2:41 AM	C Source File	1 KB
driver	5/21/2024 2:02 AM	C Header Source F...	1 KB
driver.o	5/21/2024 3:06 AM	O File	4 KB
driver.o-Disassemble	5/21/2024 3:06 AM	Text Document	4 KB
driver.o-Header	5/21/2024 3:06 AM	Text Document	2 KB
driver.o-Symbols	5/21/2024 3:06 AM	Text Document	1 KB



Name	Date modified	Type	Size
Pressure_Sensor_Driver	5/20/2024 11:35 PM	C Source File	1 KB
Pressure_Sensor_Driver	5/20/2024 2:14 PM	C Header Source F...	1 KB
Pressure_Sensor_Driver.o	5/21/2024 3:06 AM	O File	3 KB
Pressure_Sensor_Driver.o-Disassemble	5/21/2024 3:06 AM	Text Document	1 KB
Pressure_Sensor_Driver.o-Header	5/21/2024 3:06 AM	Text Document	2 KB
Pressure_Sensor_Driver.o-Symbols	5/21/2024 3:06 AM	Text Document	1 KB



```
int main (){
    GPIO_INITIALIZATION();
    uint32_t PVal;
    int Values[]={13,23};
    PVal = Values[0];

    while (1)
    {
        //Implement your Design
        //PVal = Get_Pressure_Value(); // This Provided API doesn't work so here I simulated it by testing values.

        if(PVal >= 20)
        {
            High_Pressure_Detected();
            PVal = Values[0];
        }
        else
        {
            Alarm_Reset();
            PVal = Values[1];
        }
    }
```

Here I have an important note:

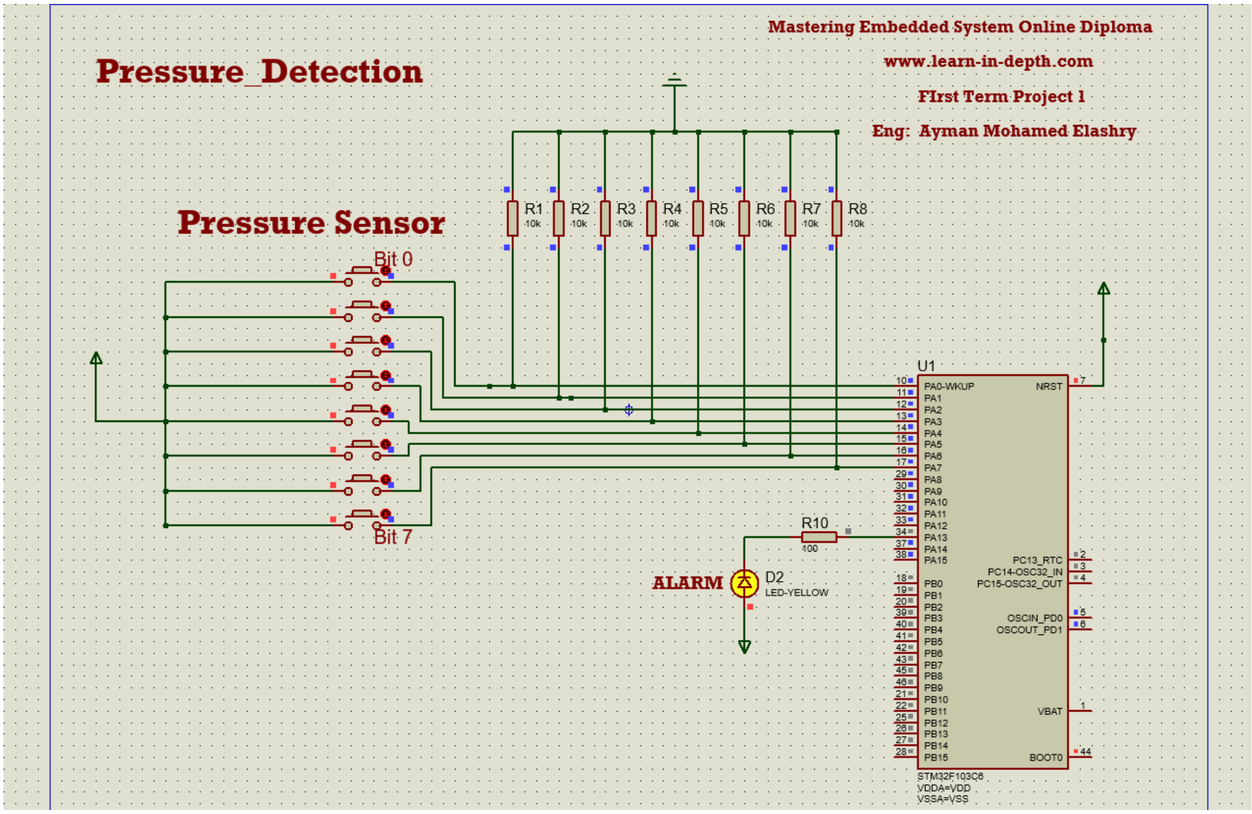
This API :

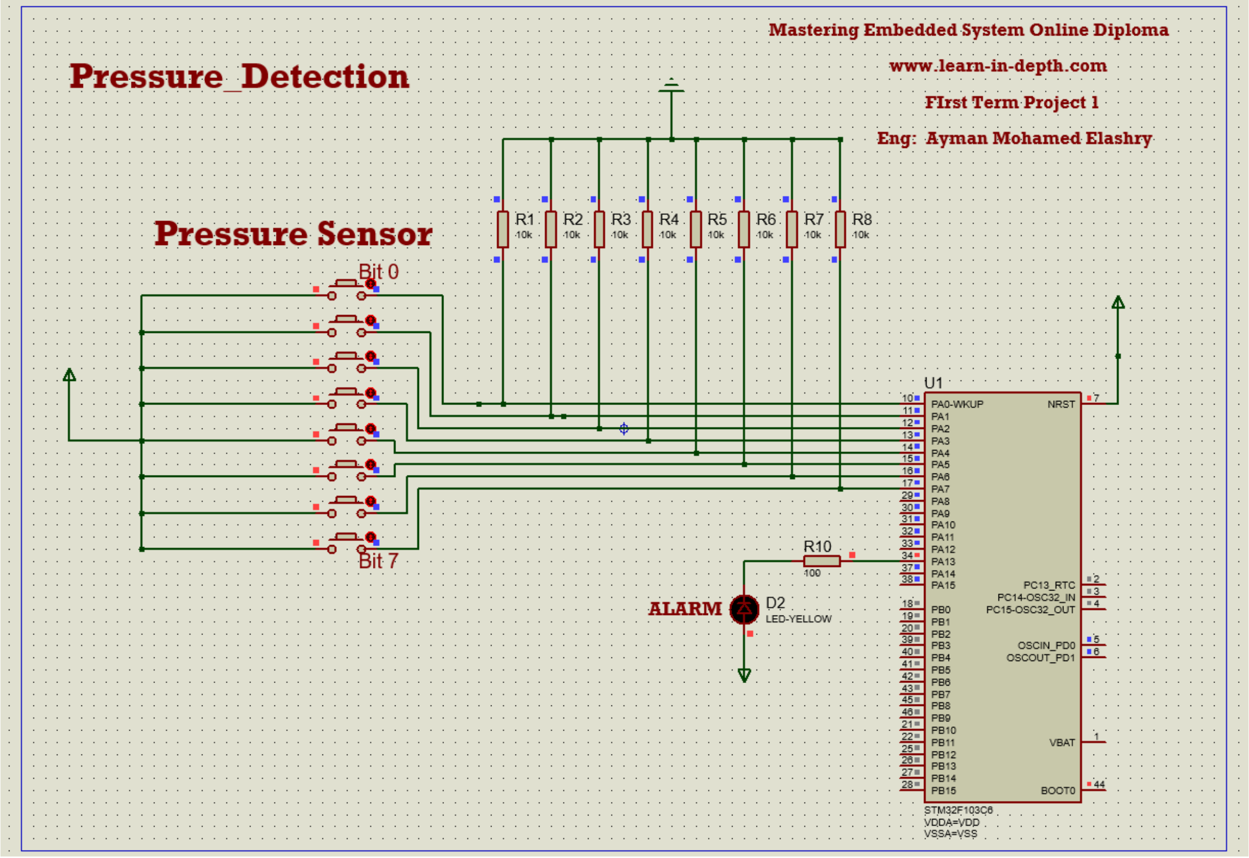
```
uint32_t getPressureVal();
```

doesn’t work.

So, I tested my logic with values I made in the C code while writing the code.

Please review this API.


Pressure_Detection
Mastering Embedded System Online Diploma
www.learn-in-depth.com
FIrst Term Project 1
Eng: Ayman Mohamed Elashry
Pressure Sensor
Bit 0
Bit 7
R1 10k
R2 10k
R3 10k
R4 10k
R5 10k
R6 10k
R7 10k
R8 10k
R10 100
ALARM
D2
LED-YELLOW
U1
PA0-WKUP
PA1
PA2
PA3
PA4
PA5
PA6
PA7
PA8
PA9
PA10
PA11
PA12
PA13
PA14
PA15
PB0
PB1
PB2
PB3
PB4
PB5
PB6
PB7
PB8
PB9
PB10
PB11
PB12
PB13
PB14
PB15
NRST
PC13_RTC
PC14-OSC32_IN
PC15-OSC32_OUT
OSCIN_PD0
OSCOUT_PD1
VBAT
BOOT0
STM32F103C6
VDDA=VDD
VSSA=VSS


Pressure_Detection
Mastering Embedded System Online Diploma
www.learn-in-depth.com
FIrst Term Project 1
Eng: Ayman Mohamed Elashry
Pressure Sensor
Bit 0
Bit 7
R1 10k
R2 10k
R3 10k
R4 10k
R5 10k
R6 10k
R7 10k
R8 10k
R10 100
ALARM
D2
LED-YELLOW
U1
PA0-WKUP
PA1
PA2
PA3
PA4
PA5
PA6
PA7
PA8
PA9
PA10
PA11
PA12
PA13
PA14
PA15
PB0
PB1
PB2
PB3
PB4
PB5
PB6
PB7
PB8
PB9
PB10
PB11
PB12
PB13
PB14
PB15
NRST
PC13_RTC
PC14-OSC32_IN
PC15-OSC32_OUT
OSCIN_PD0
OSCOUT_PD1
VBAT
BOOT0
STM32F103C6
VDDA=VDD
VSSA=VSS