Nov 2021

Mastering embedded system online diploma by Eng. Keroles Shenouda  
[www.learn-in-depth.com](http://www.learn-in-depth.com)

First Term (Final Project 1)

Eng. Mazen Talaat

My profile: <https://bit.ly/3DUmnyz>

Table of contents

[Case Study 2](#_Toc88862190)

[Assumptions 2](#_Toc88862191)

[Method 2](#_Toc88862192)

[Requirements Diagram 3](#_Toc88862193)

[System Analysis 4](#_Toc88862194)

[Use case diagram 4](#_Toc88862195)

[Activity diagram 4](#_Toc88862196)

[Sequence diagram 5](#_Toc88862197)

[Block Diagram 6](#_Toc88862198)

[Pressure sensor 6](#_Toc88862199)

[Alarm controller 6](#_Toc88862200)

[Pressure controller 6](#_Toc88862201)

[Alarm state machine 7](#_Toc88862202)

[Pressure sensor state machine 7](#_Toc88862203)

[Pressure controller state machine 8](#_Toc88862204)

[Final Simulation 9](#_Toc88862205)

[Code 10](#_Toc88862206)

[driver.h/c 10](#_Toc88862207)

[Pressure\_Senor.h/c 11](#_Toc88862208)

[Alarm\_Controller.h/c 12](#_Toc88862209)

[Pressure\_Controller.h/c 13](#_Toc88862210)

[main.c 14](#_Toc88862211)

[startup.c 15](#_Toc88862212)

[linker\_script.ld 15](#_Toc88862213)

[map\_file.map 16](#_Toc88862214)

[MakeFile 17](#_Toc88862215)

[Proteus Simulation 18](#_Toc88862216)

[Symbols table 21](#_Toc88862217)

[Sections 22](#_Toc88862218)

# Case Study

A “client” expects to deliver the software of the following system:

* 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.

# 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
* The controller never faces power cut

# Method

I will use the waterfall method which consists of

1. Requirements.
2. Analysis.
3. Design.
4. Coding.
5. Testing.
6. Deployment.

# Idea

1. Get the pressure sensor data.
2. If the pressure is greater than or equal to the threshold = 20 bars.
   1. Start the alarm for 60 seconds.
3. If the pressure is less than the threshold = 20 bars.
   1. Continue.
4. Repeat.

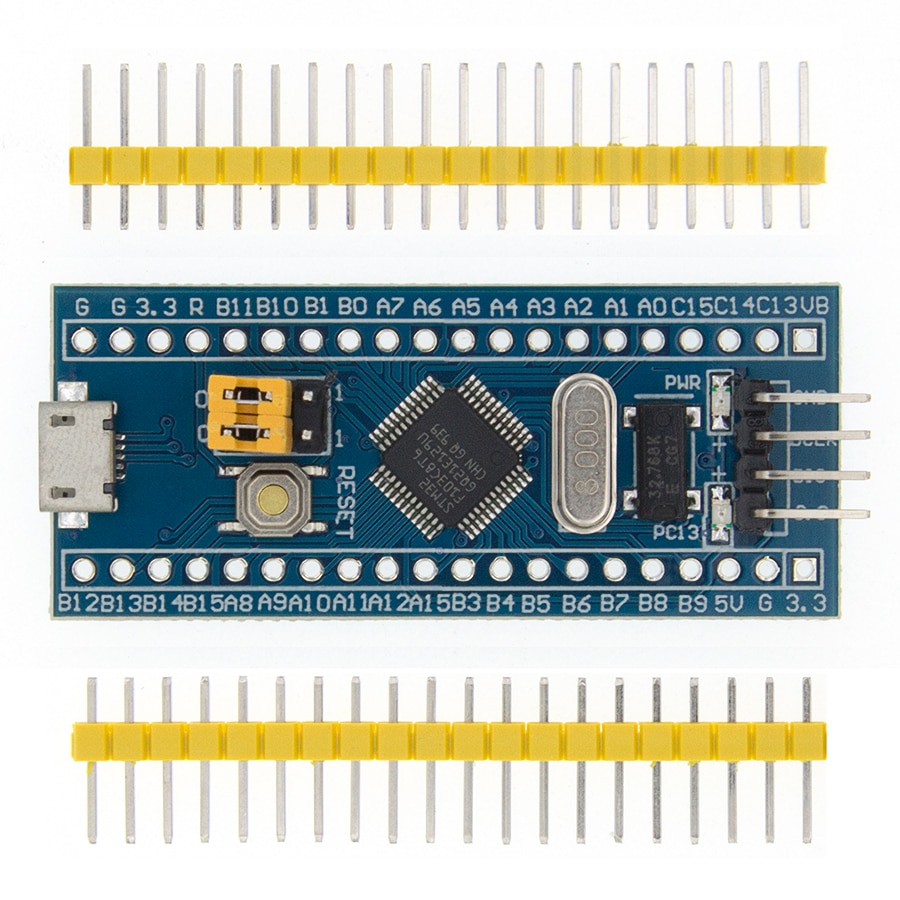
# Requirements Diagram

Diagram

Description automatically generated

# Space Exploration

We will use ARM STM32 board as it meets our requirements and available in stores.



# System Analysis

## Use case diagram

Diagram

Description automatically generated

## Activity diagram

Diagram

Description automatically generated

## Sequence diagram

Diagram

Description automatically generated with low confidence

# Block Diagram

Chart

Description automatically generated

We will have 3 blocks:

1. Pressure sensor.
2. Pressure controller.
3. Alarm controller.

## Pressure sensor

This block will have the pressure variable and will set it by the function getPressureVal.

## Alarm controller

This block will have the control over the alarm actuator using the function Set\_Alarm\_actuator.

## Pressure controller

This block will have the control algorithm by retrieving, comparing and sending.

# Alarm state machine

Diagram

Description automatically generated

The alarm actuator will be initialized and begin at idle state and will start or stop depending on the “en” variable.

# Pressure sensor state machine

Diagram

Description automatically generated

The pressure sensor will be initialized and begin at busy state and keep sending the pressure value every 10 sec.

# Pressure controller state machine

Diagram

Description automatically generated

The system will begin at the waiting state then sets the alarm off and get the pressure from the sensor then checks whether the pressure is greater than the threshold or not.  
If it is greater, it will switch to the alarm state and set the alarm on for 60 seconds then repeat the process.

# Final Simulation

Diagram, schematic

Description automatically generated

All blocks started with initializing, then busy for sensor, waiting for the controller and idle for the alarm.  
Then, the controller sets the alarm off and get the pressure from the sensor which is equal to 18. It waits for 10 seconds to get the new pressure from sensor.  
When the pressure is equal to 21, it switches to the alarm state and set the alarm on for 60 seconds, after that it will read from the sensor again.

# Code

## driver.h/c

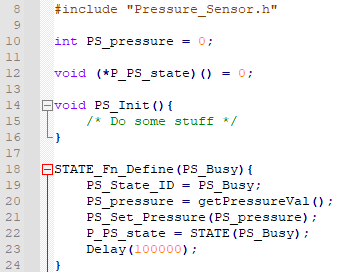
Table

Description automatically generated with medium confidence

Text

Description automatically generated

## Pressure\_Senor.h/c

Diagram

Description automatically generated

Text

Description automatically generated

## Alarm\_Controller.h/c

Text

Description automatically generatedDiagram

Description automatically generated

Text

Description automatically generated

## Pressure\_Controller.h/c

Graphical user interface, text, application

Description automatically generatedText, letter

Description automatically generatedDiagram

Description automatically generated

## main.c

Text

Description automatically generated

## startup.c

Text

Description automatically generated

## linker\_script.ld

A picture containing diagram

Description automatically generated

## map\_file.map

Table

Description automatically generatedTable

Description automatically generated

## MakeFile

Text

Description automatically generated with medium confidence

# Proteus Simulation

Diagram, schematic

Description automatically generated



Pressure = 0, LED Off

Diagram, schematic

Description automatically generated

Table

Description automatically generated

Pressure = 32, LED On

Schematic

Description automatically generated with medium confidence



Pressure = 0, LED Off

After some time (nCount)

# Symbols table

Text

Description automatically generated

# Sections

main.o

Graphical user interface, text

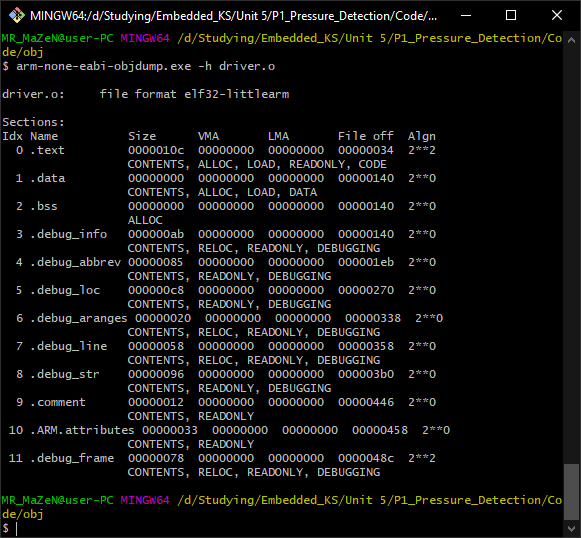
Description automatically generated

Alarm\_Controller.o

Graphical user interface, text

Description automatically generated

driver.o



Pressure\_Controller.o

Graphical user interface, text, application

Description automatically generated

Pressure\_Sensor.o

Graphical user interface, text

Description automatically generated

startup.o

Graphical user interface, text, application

Description automatically generated

Project.elf

Graphical user interface, text

Description automatically generated