Table of Contents:

1-introduction

2- system architecting /design sequence

2.1 case study

2.2 method

2.3 system requirement

2.4 space exploration/partitioning

2.5 System analysis

2.5.1 use case diagram

2.5.2 Activity diagram

2.5.3 sequence diagram

2.6 system design

2.7 system verification

3- source code

3.0- make file

3.1 – state.h

3.2- pressure\_sensor.c/.h

3.3 – main\_algo.c/.h

3.4- alarm\_moinitor.c/.h

3.5 – alarm\_driver.c/.h

3.6 – driver.c/.h

3.7-main.c

3.8 – startup code with assembly

3.9 -linker script

4- Symbols

4.1- pressure sensor

4.2 – main algo

4.3- alarm\_moinitor

4.4 – alarm driver

4.5 – driver

4.6- main

4.7 – startup

4.8- elf image symbols

5- Sections Headers

5.1- pressure sensor

5.2 – main algo

5.3- alarm\_moinitor

5.4 – alarm driver

5.5 – driver

5.6- main

5.7 – startup

5.8- elf image symbols 6- run application on protos

7 -run on protos

1-introuction

This project is about how to detect high pressure in cabin and inform the crew of cabin with an alarm when the pressure exceeds 20 bars and the alarm duration equals 60 seconds.

2- system architecting /design sequence

2.1 case study (pressure controlling system)

-” **client**” expects you to deliver the software of the following system:

- Specification (from the client)

**- 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.

- keeps track of the measured values.

- Pressure Controller: Assumptions

The controller set up and shutdown procedures are not modeled

The controller maintenance is not modeled

The pressure sensor never fails

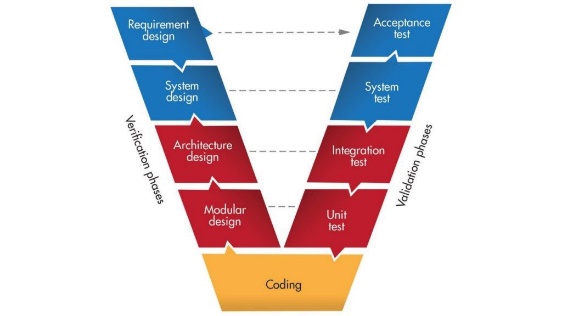
The alarm never fails

The controller never faces power cut

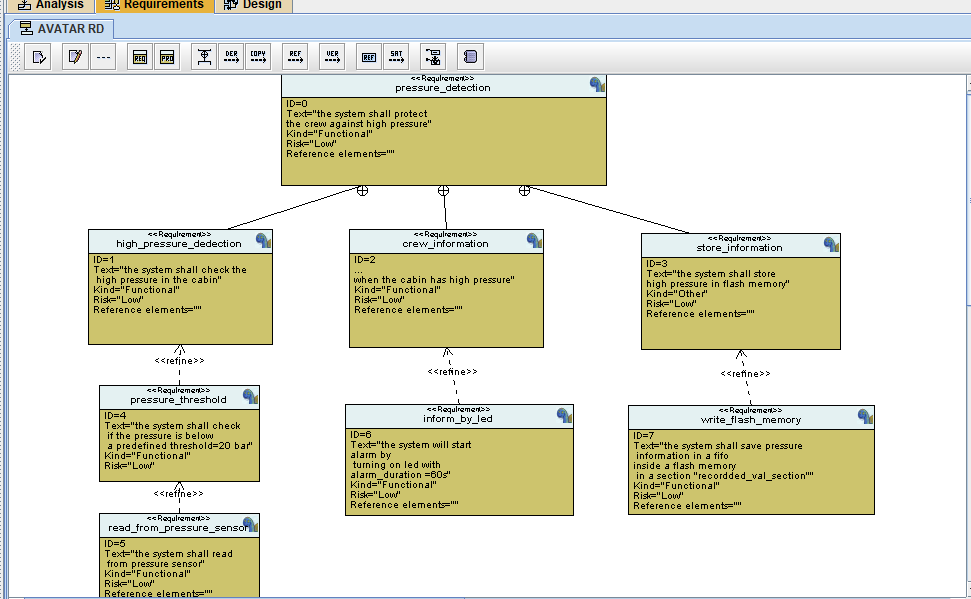
**- Versioning**• The” keep track of measured value” option is not modeled in the first version of the design

2.2 method (SDLC &STLC)

starting with v-model for example



2.3 system requirement

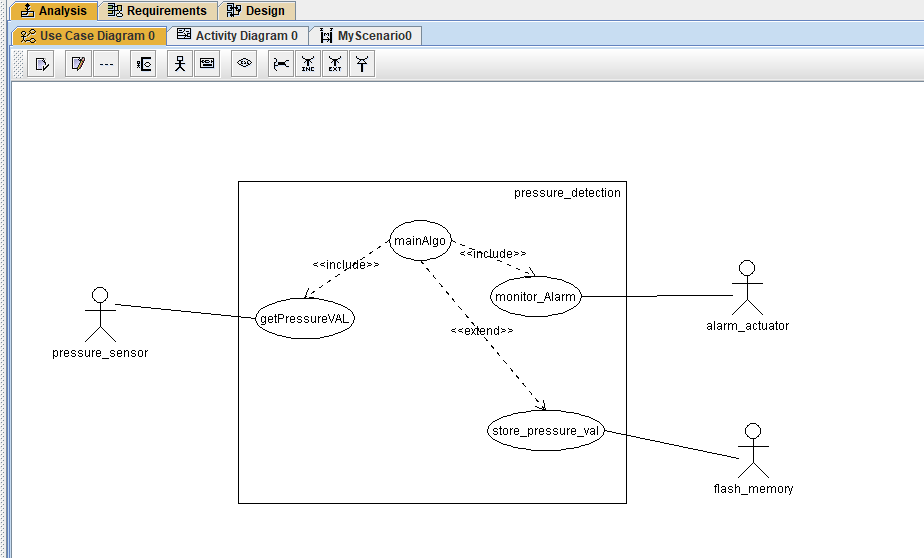


2.4 space exploration/partitioning

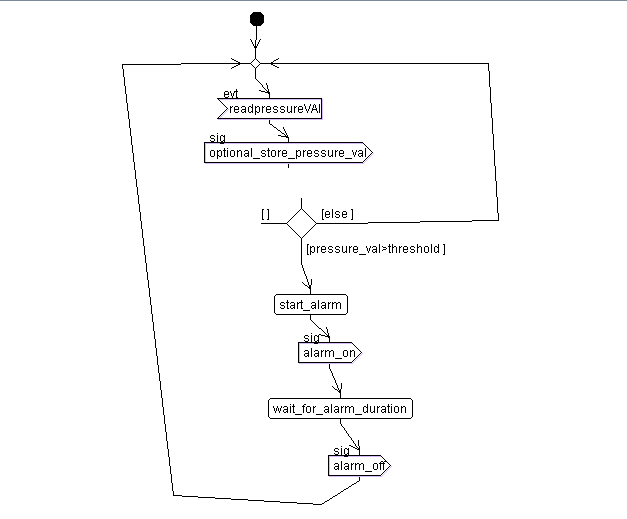
How to chose MCs and number of MCs.

2.5 System analysis

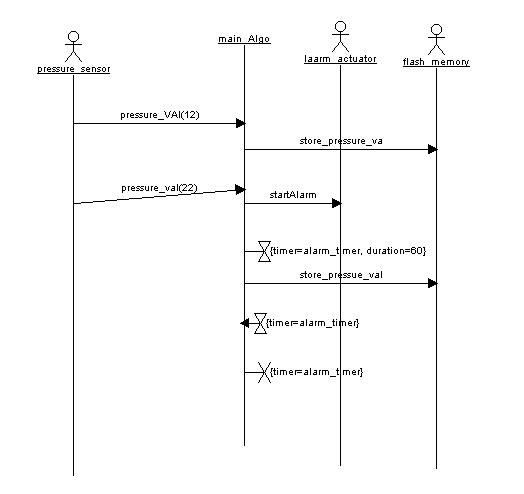
2.5.1 use case diagram



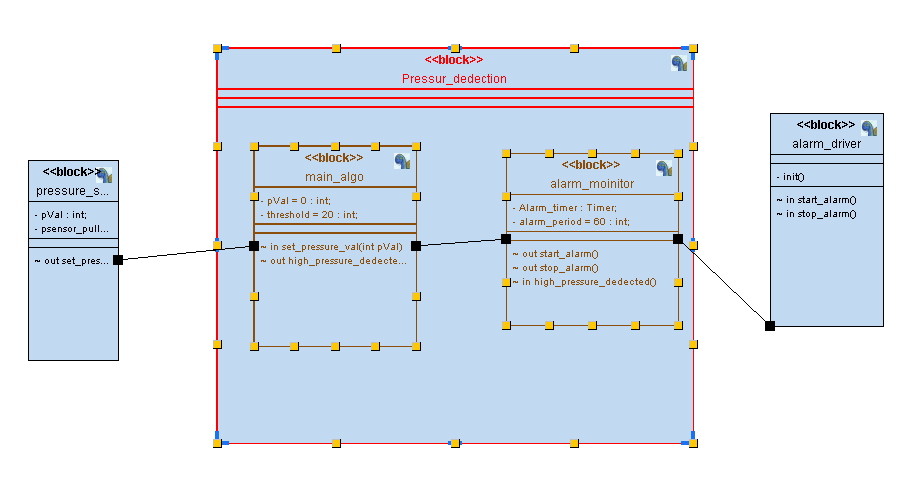
2.5.2 Activity diagram



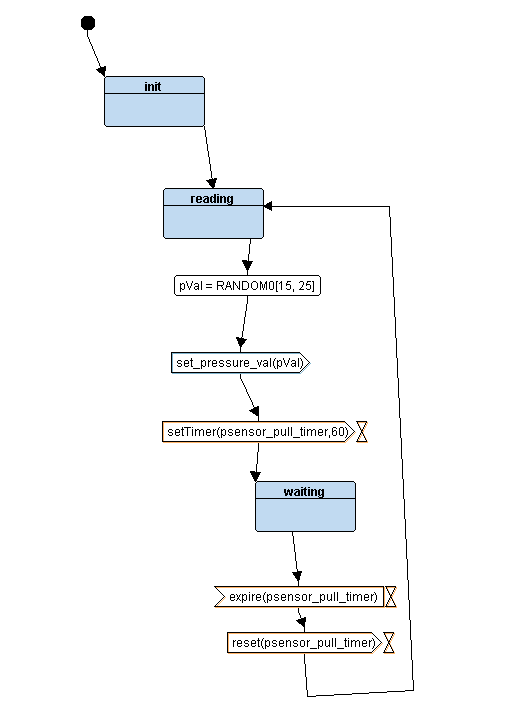
2.5.3 sequence diagram



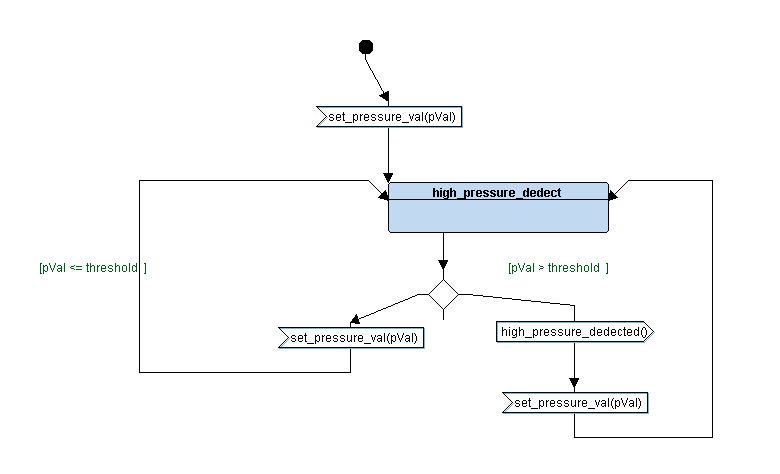
2.6 system design



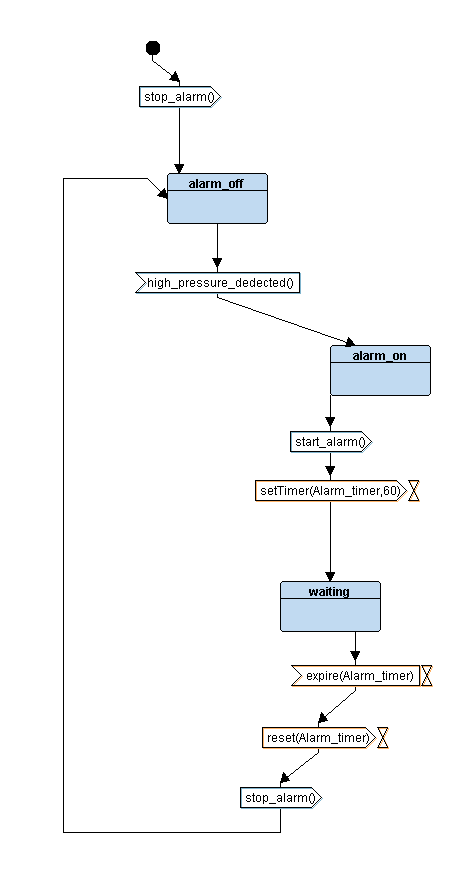
Pressure detection state diagram



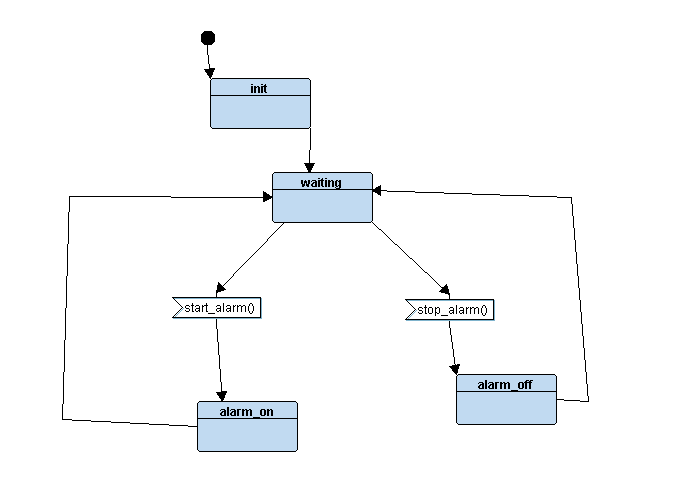
Main algorithm state diagram



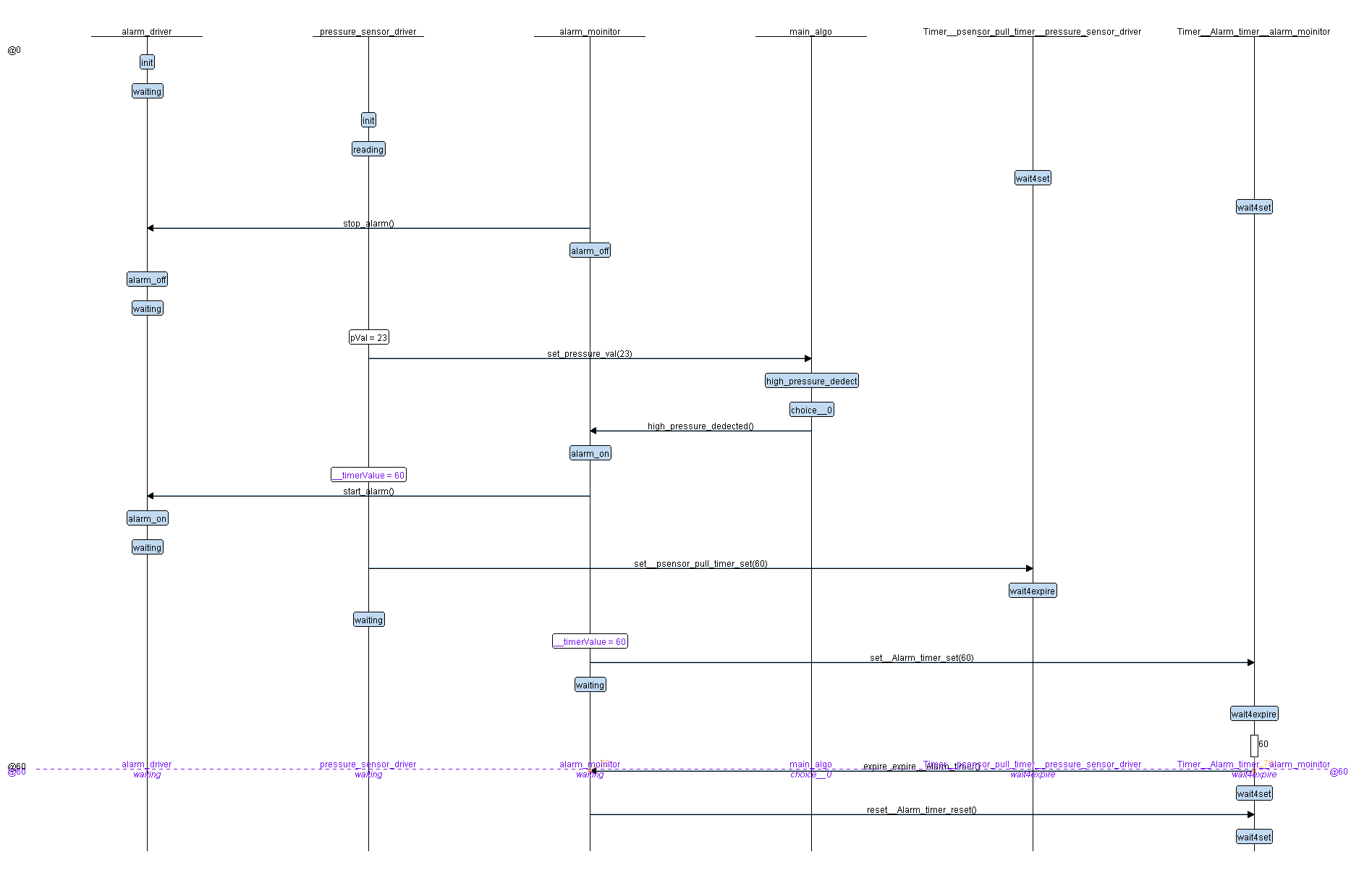
Alarm monitoring state diagram



Alarm driver state diagram

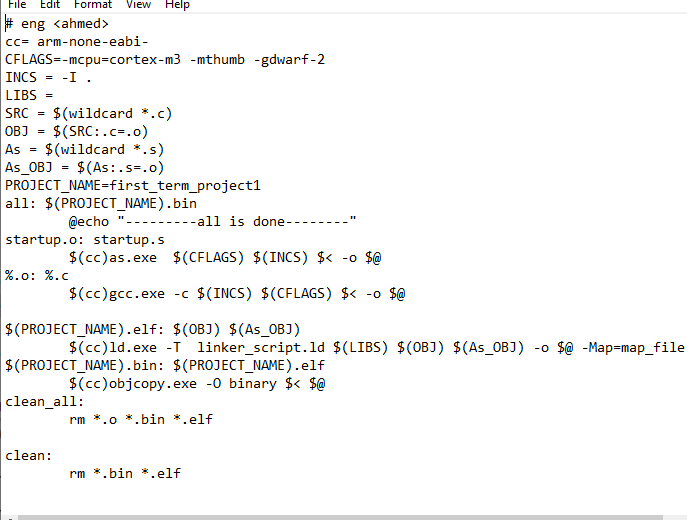


2.7 system vervication



3- source code

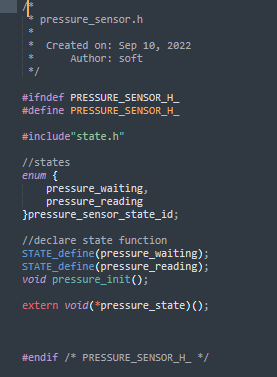
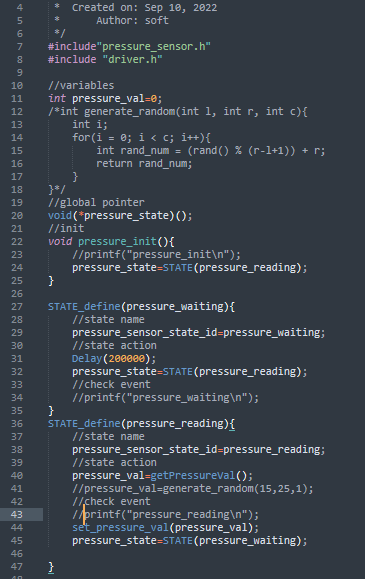
3.0- make file



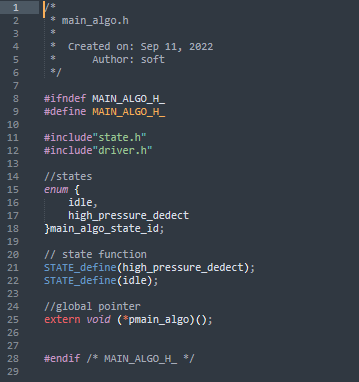
3.1 – state.h



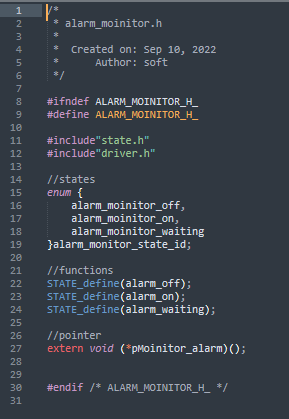
3.2- pressure\_sensor.c/.h



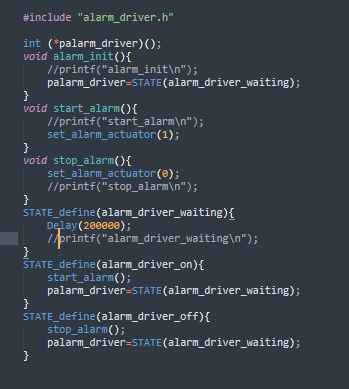
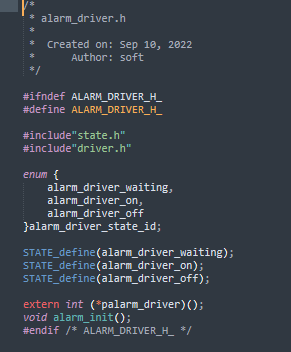
3.3 – main\_algo.c/.h



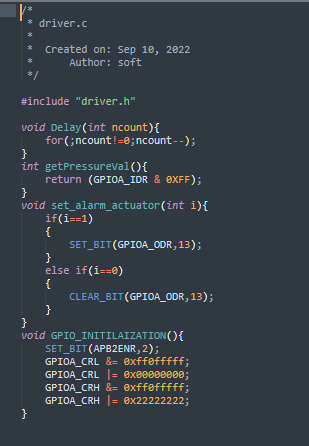
3.4- alarm\_moinitor.c/.h



3.5 – alarm\_driver.c/.h



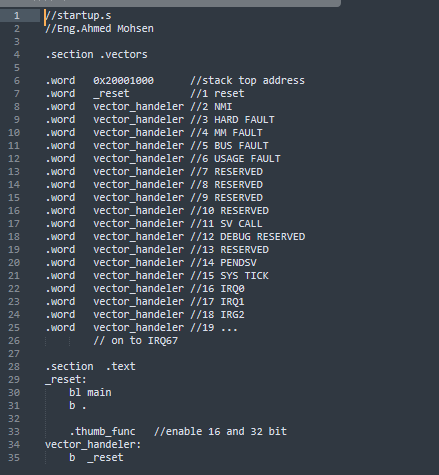
3.6 – driver.c/.h



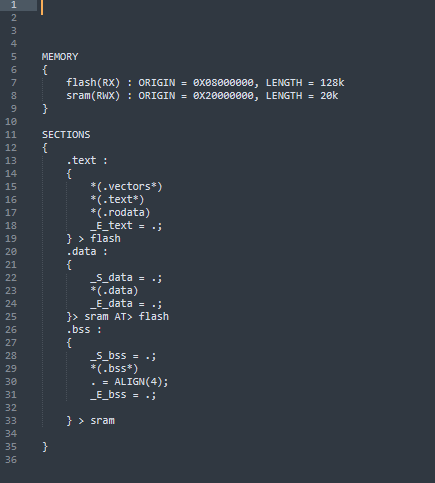
3.7-main.c



3.8 – starup code with assembly

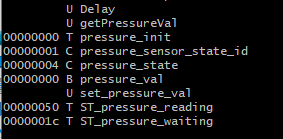


3.9 -linker script

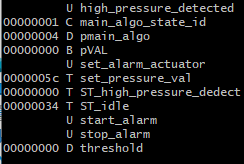


4- Symbols

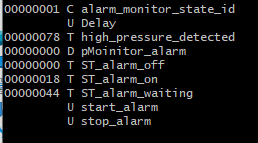
4.1- pressure\_sensor.o



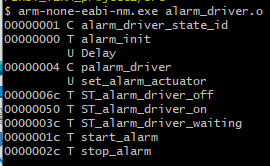
4.2 – main\_algo.o



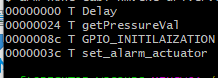
4.3- alarm\_moinitor.o



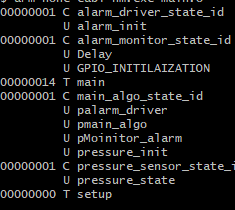
4.4 – alarm\_driver.o



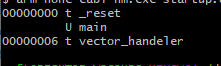
4.5 – driver.o



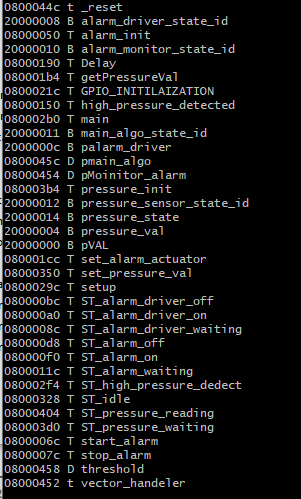
4.6-main.o



4.7 – starup .o

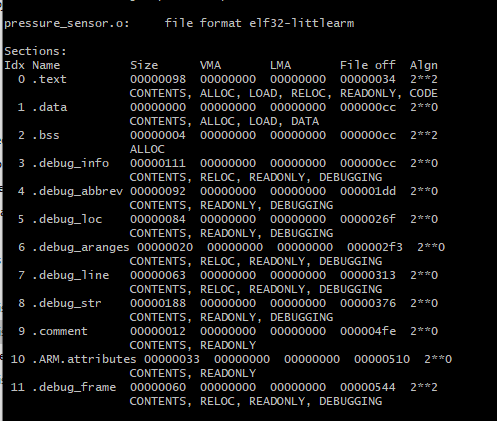


4.8- elf image symbols

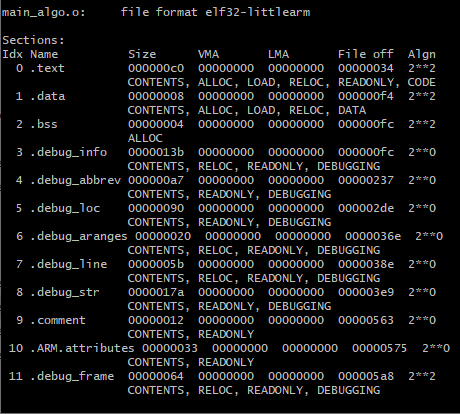


5- Sections Headers

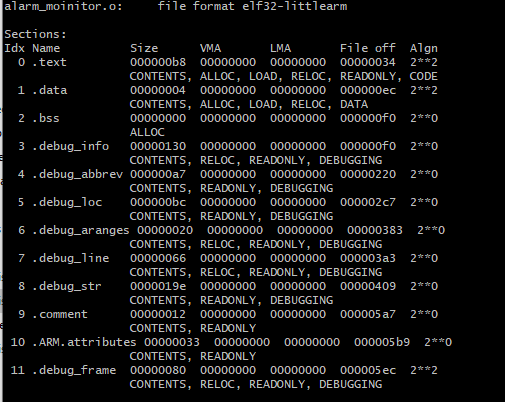
5.1- pressure sensor



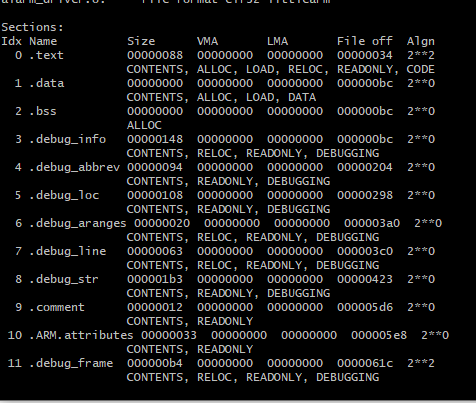
5.2 – main algo



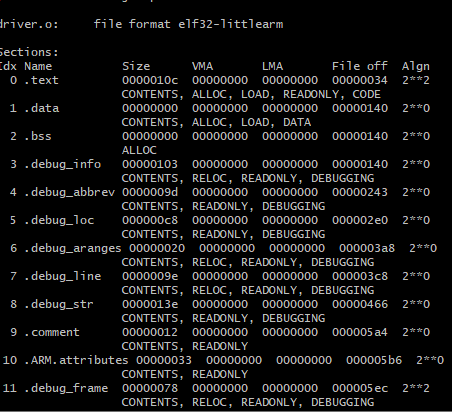
5.3- alarm\_moinitor



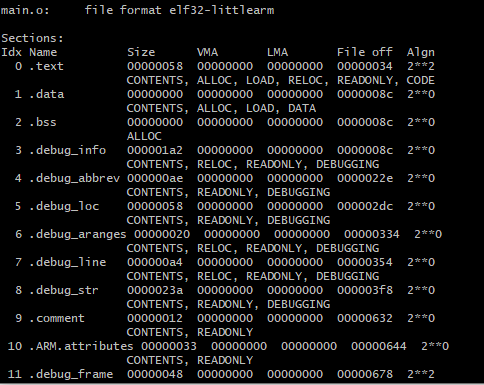
5.4 – alarm driver



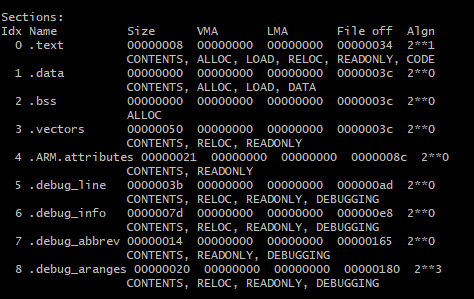
5.5 – driver



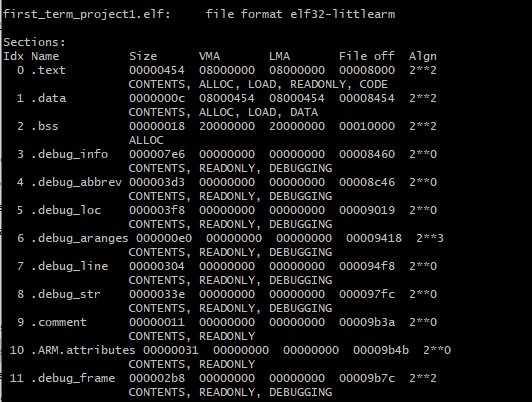
5.6- main



5.7 – startup



5.8- elf image symbols 6- run application on protus



7 -run on protos

