Mastering Embedded System Online Diploma

http://learn-in-depth.com/

First Term (Final Project 1)

Eng: Ayman Gamal Abd_Elbaset

My Profile:

https://www.learn-in-depth.com/online-diploma/ayman2gamal98%40gmail.com

Contents

Pressure Controlling System

1. Introduction

This project will be good for cabin crew, it will detect the pressure inside cabin and in case of high pressure it will told the crew by alarm, it will detect pressure bigger than 20 bar and alarm will go on for 60 second

This file will know you everything for this project using SysML language.

2. System architecting /Design sequence

2.1 Case study

A client expects you 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.

-keeps track of the measured values

Assumption:

1.the alarm never fails.

2.the pressure sensor never fails.

3.the controller maintenance is not modeled.

4. the controller never faces power cut.

2.2 V model

We will use this method in implement software design

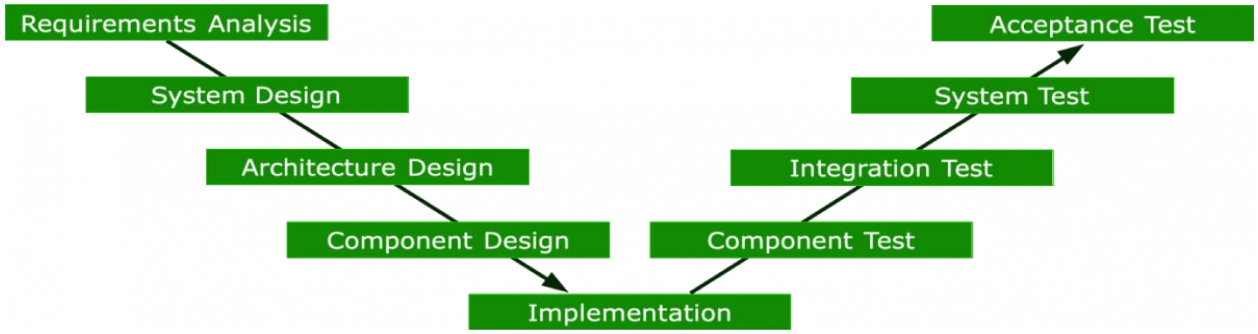


Figure 1 V method

2.3 Requirement

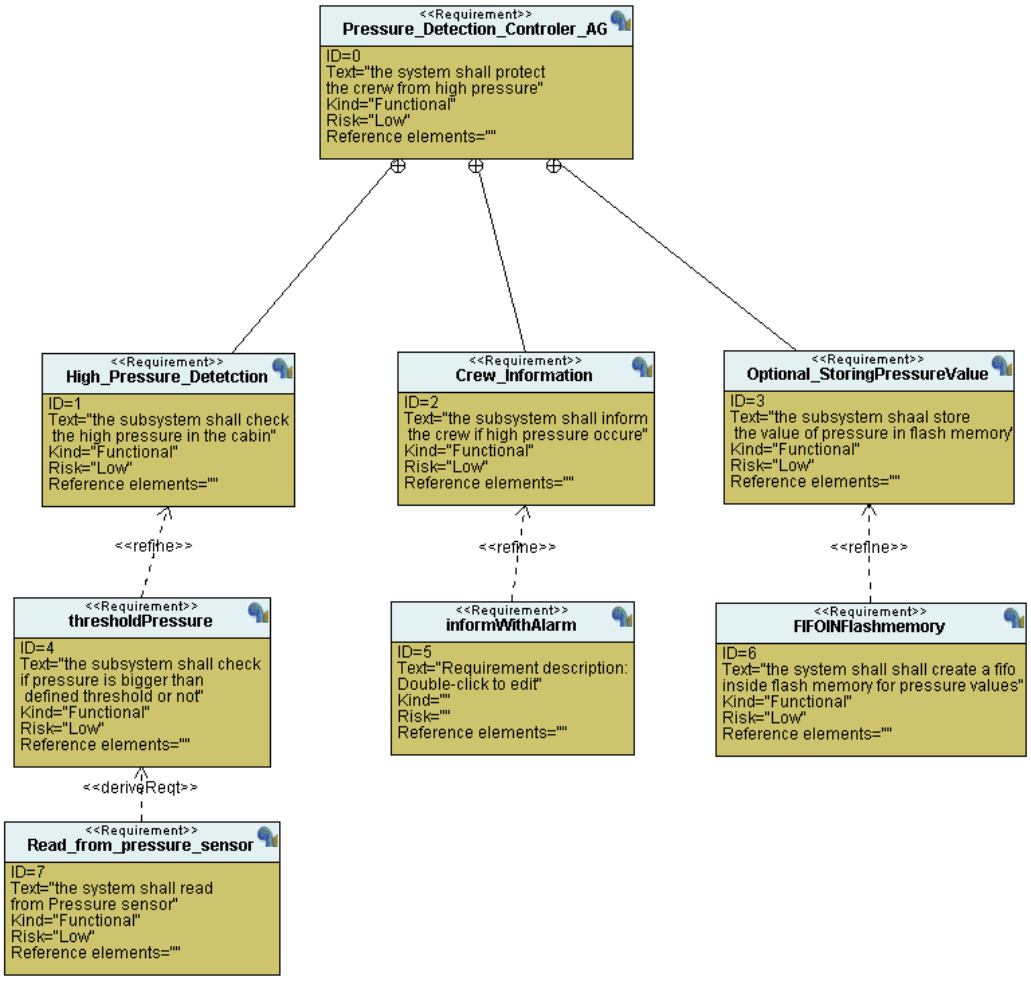


Figure 2 Requirement

2.4 Hardware /Software Partitioning

life.augmented

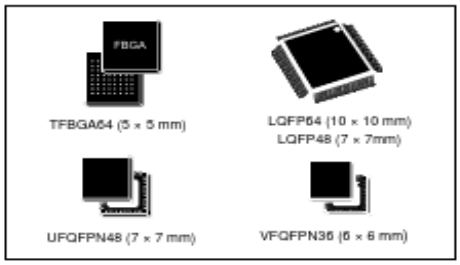
STM32F103x4
STM32F103x6

Low-density performance line, ARM-based 32-bit MCU with 16 or 32 KB Flash, USB, CAN, 6 timers, 2 ADCs, 6 com. interfaces

Datasheet – production data

Features

- ARM 32-bit Cortex™-M3 CPU Core
  - 72 MHz maximum frequency, 1.25 DMIPS/MHz (Dhrystone 2.1) performance at 0 wait state memory access
  - Single-cycle multiplication and hardware division
- Memories
  - 16 or 32 Kbytes of Flash memory
  - 6 or 10 Kbytes of SRAM
- Clock, reset and supply management
  - 2.0 to 3.6 V application supply and I/Os
  - POR, PDR, and programmable voltage detector (PVD)
  - 4-to-16 MHz crystal oscillator
  - Internal 8 MHz factory-trimmed RC
  - Internal 40 kHz RC
  - PLL for CPU clock
  - 32 kHz oscillator for RTC with calibration
- Low power
  - Sleep, Stop and Standby modes
  - V_{BAT} supply for RTC and backup registers
- 2 x 12-bit, 1 µs A/D converters (up to 16 channels)
  - Conversion range: 0 to 3.6 V
  - Dual-sample and hold capability
  - Temperature sensor
- DMA
  - 7-channel DMA controller
  - Peripherals supported: timers, ADC, SPIs, I²Cs and USARTs
- Up to 51 fast I/O ports
  - 26/37/51 I/Os, all mappable on 16 external interrupt vectors and almost all 5 V-tolerant



- Debug mode
  - Serial wire debug (SWD) & JTAG interfaces
- 6 timers
  - Two 16-bit timers, each with up to 4 IC/OC/PWM or pulse counter and quadrature (incremental) encoder input
  - 16-bit, motor control PWM timer with dead-time generation and emergency stop
  - 2 watchdog timers (Independent and Window)
  - SysTick timer 24-bit downcounter
- 6 communication interfaces
  - 1 x I²C interface (SMBus/PMBus)
  - 2 × USARTs (ISO 7816 interface, LIN, IrDA capability, modem control)
  - 1 × SPI (18 Mbit/s)
  - CAN interface (2.0B Active)
  - USB 2.0 full-speed interface
- CRC calculation unit, 96-bit unique ID
- Packages are ECOPACK®

Table 1. Device summary

Reference	Part number
STM32F103x4	STM32F103C4, STM32F103R4, STM32F103T4
STM32F103x6	STM32F103C6, STM32F103R6, STM32F103T6

This is information on a product in full production. www.st.com

Figure 3 Data sheet stm32

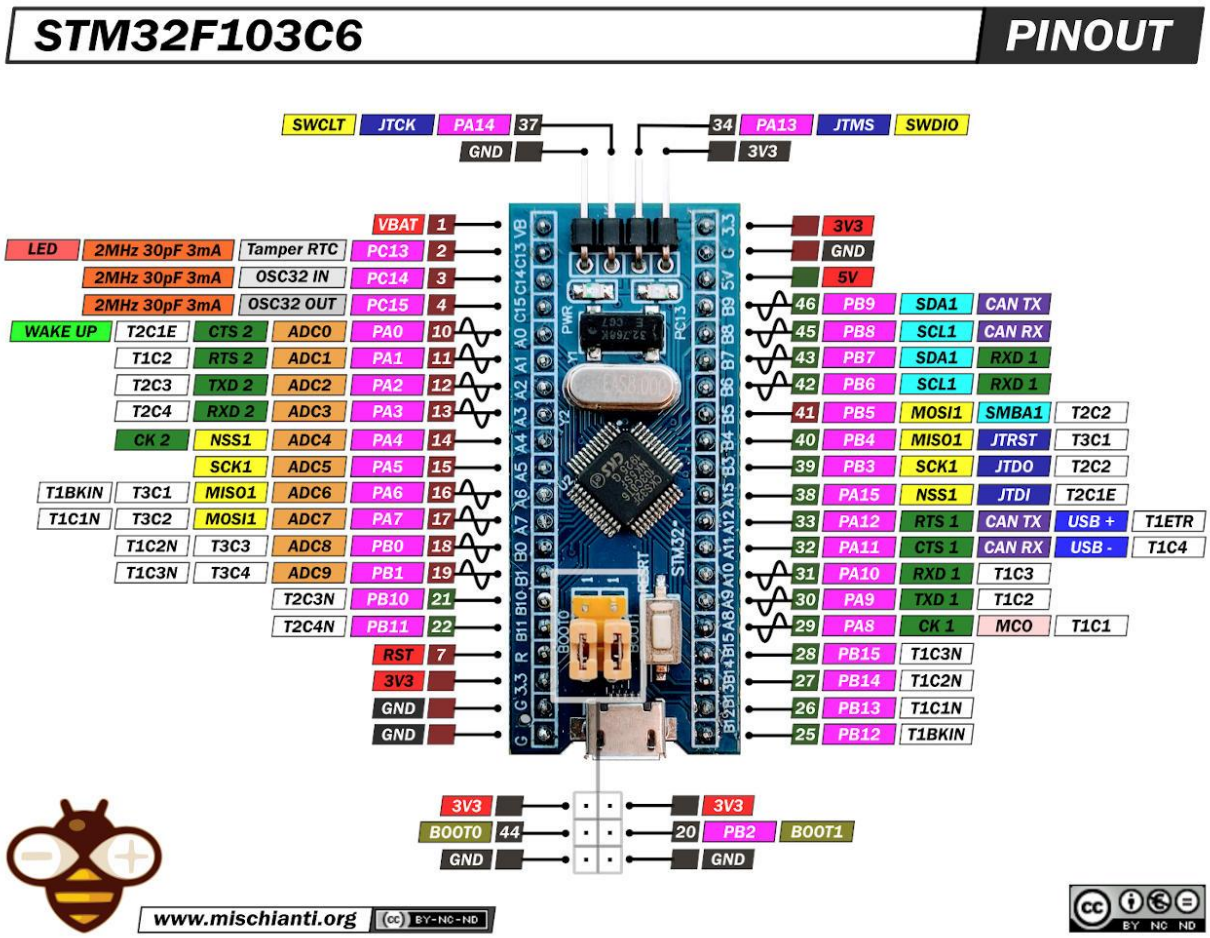


Figure 4 stm32 kit

2.5 System Analysis

2.5.1 Use Case Diagram

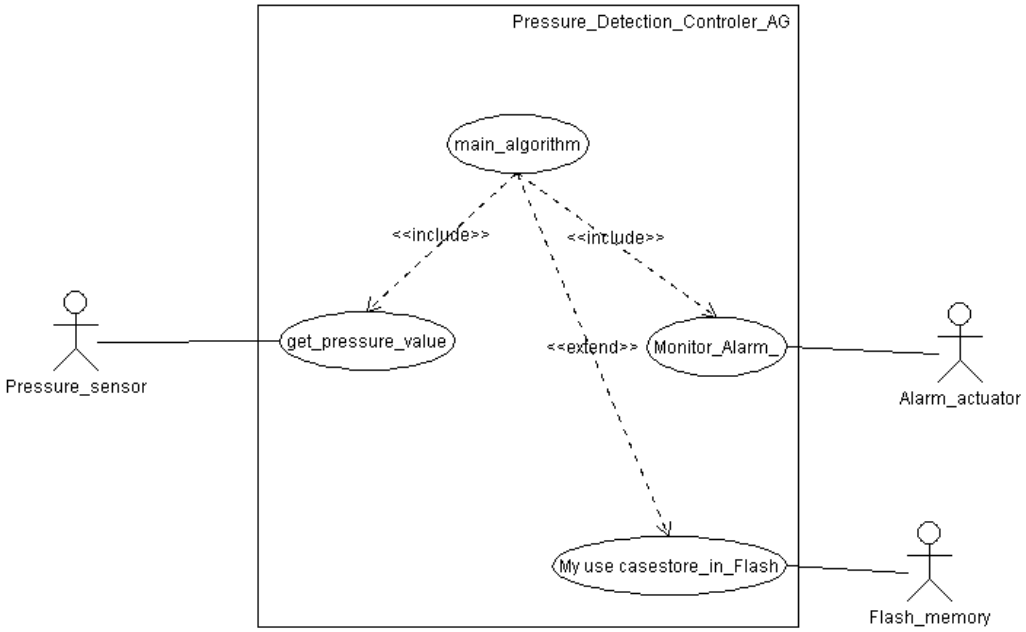
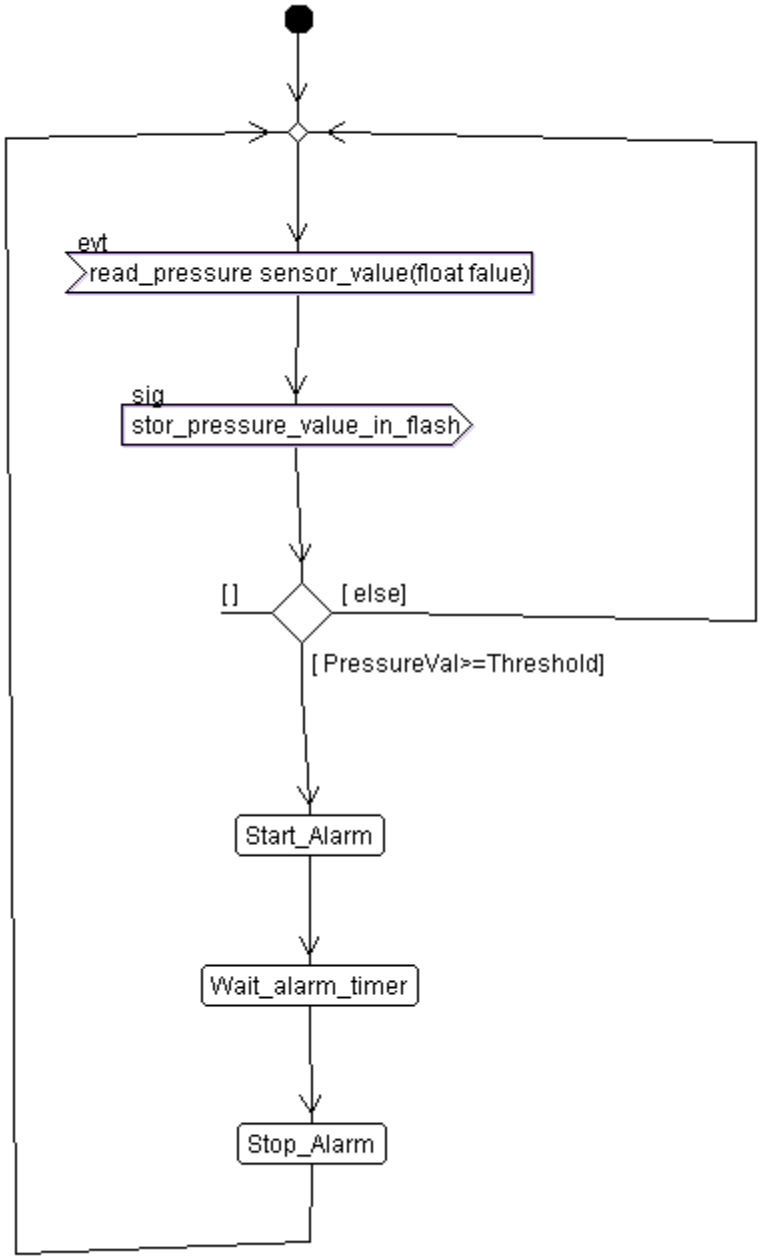


Figure 5 Use Case diagram

2.5.2 Activity Diagram



2.5.3 Sequence diagram

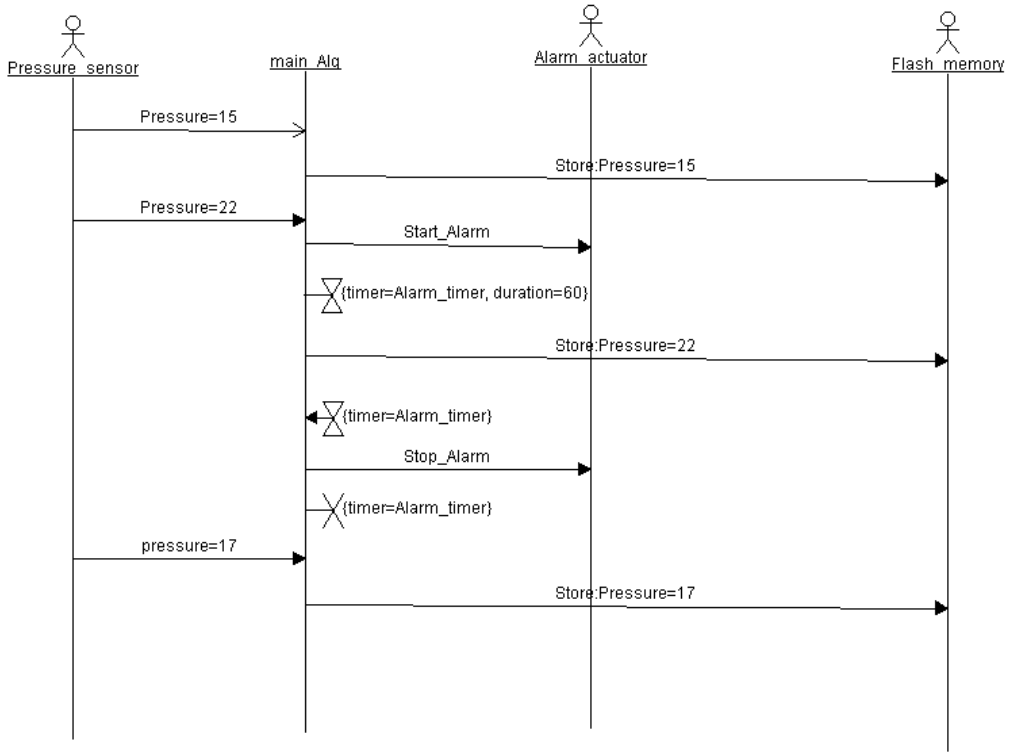


Figure 6 scenario

2.6 System Design

2.6.1 Block Diagram

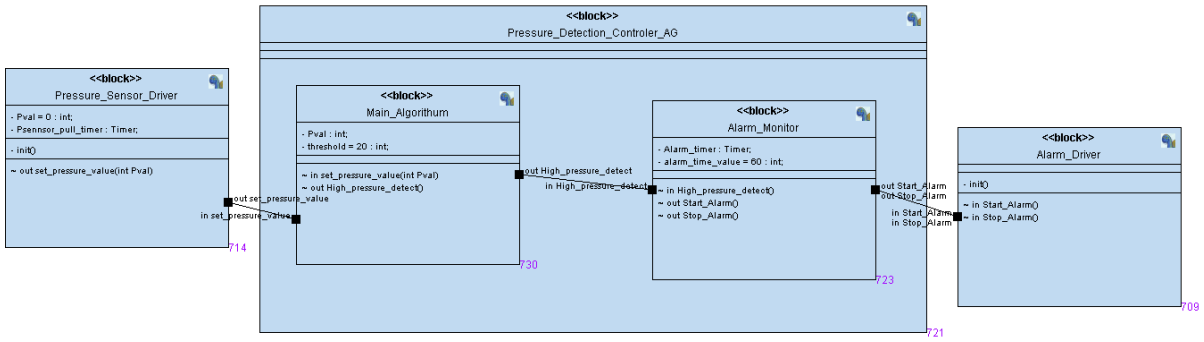


Figure 7 Block diagram

2.6.2 State Machine: Pressure Sensor Driver

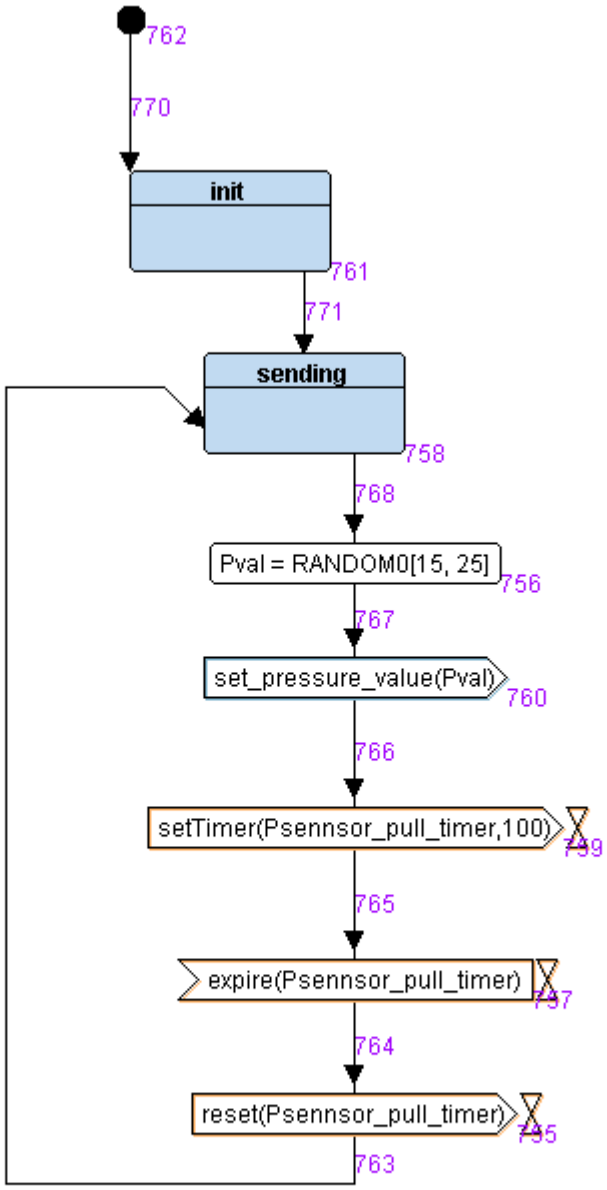


Figure 8 Pressure Sensor

2.6.3 State Machine: Main Algorithm

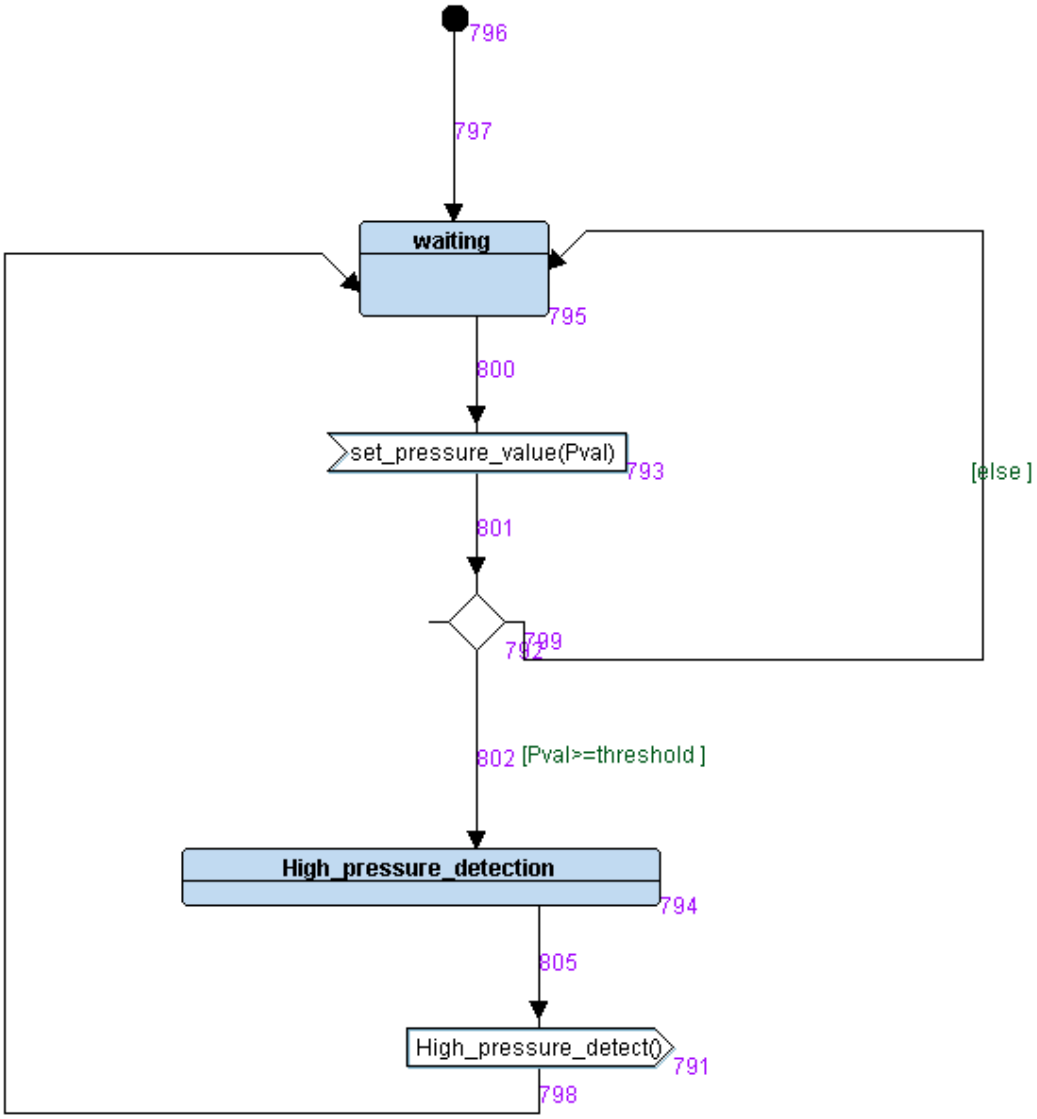


Figure 9 main algorithm

2.6.4 State Machine: Alarm Monitor

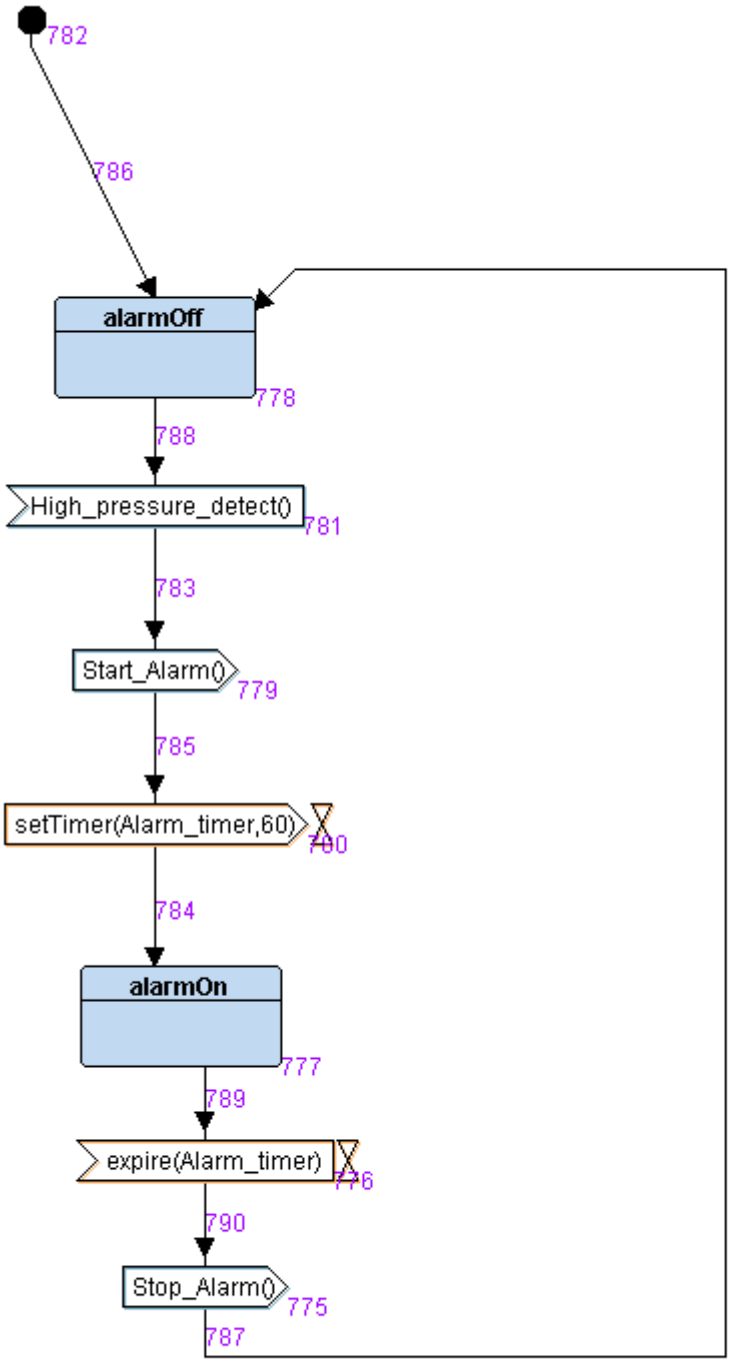


Figure 10 Alarm monitor

2.6.5 State Machine: Alarm Actuator Driver

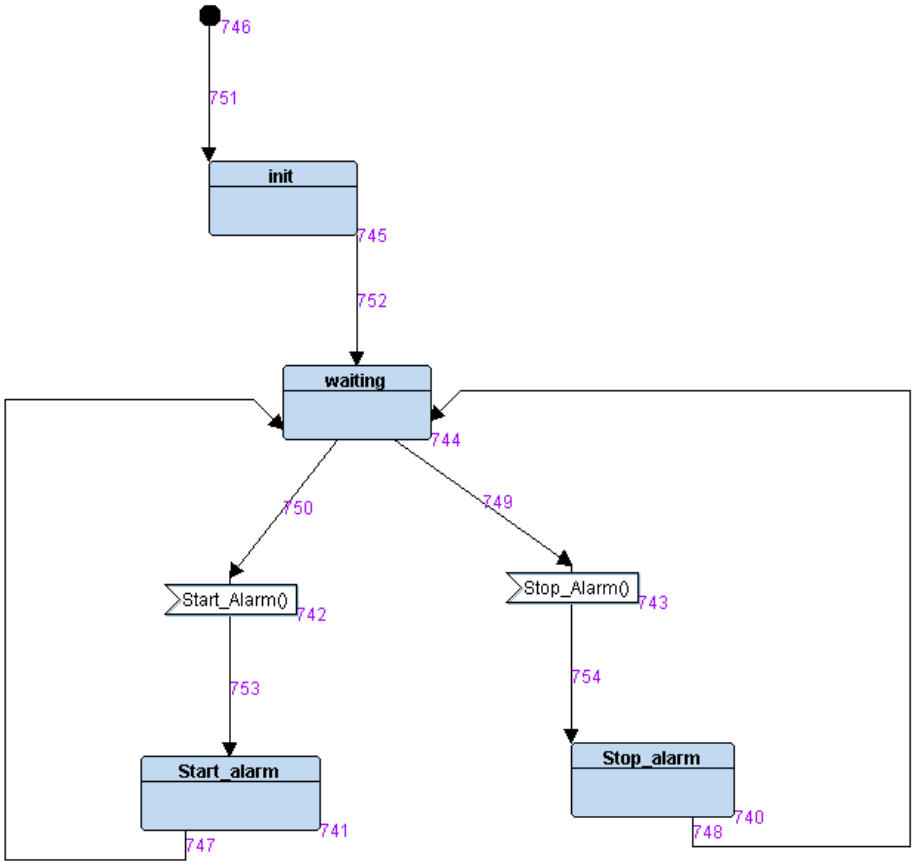


Figure 11 Alarm Driver

2.6.6 Simulation for state machine

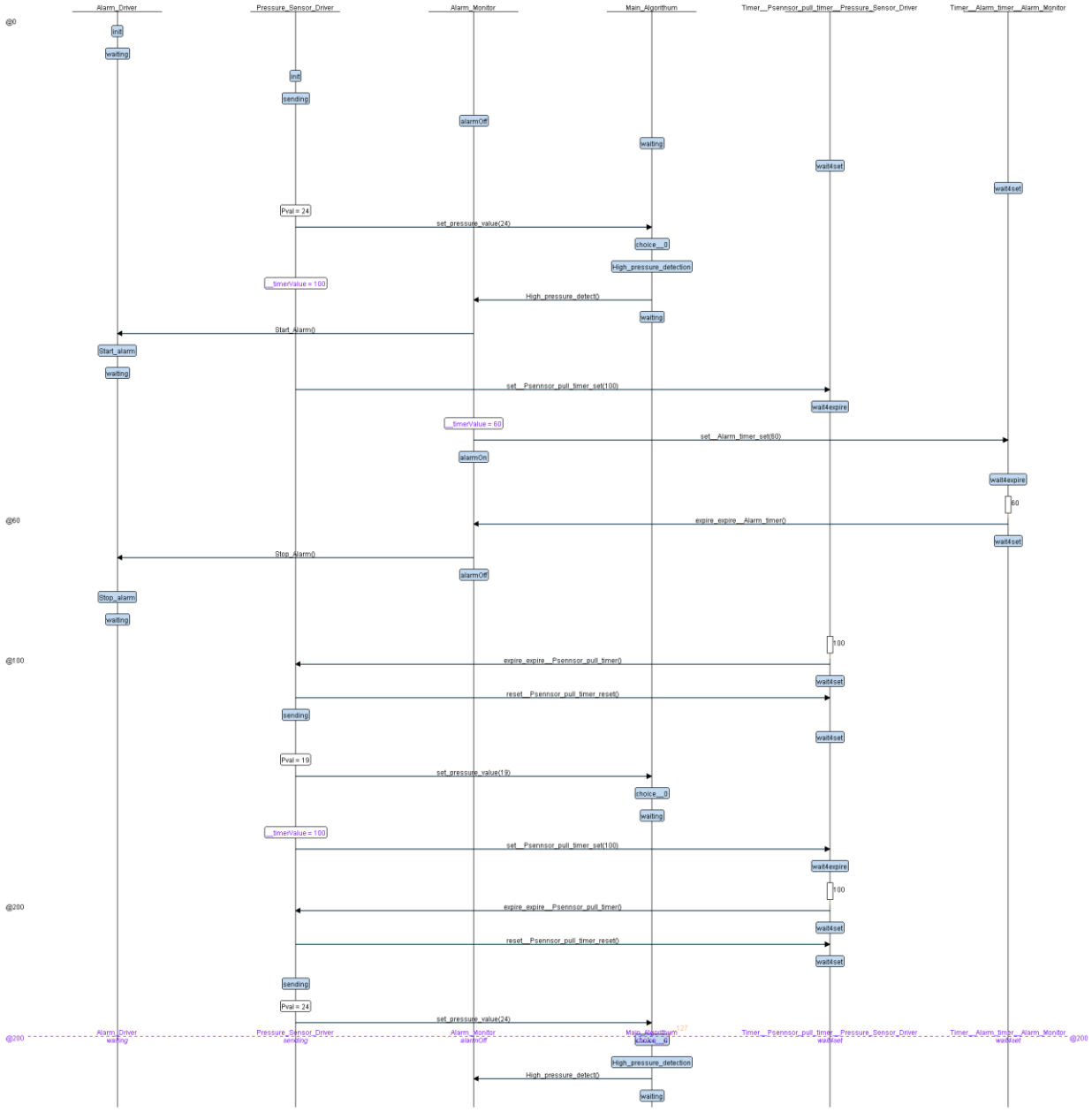


Figure 12 simulation for state machine

3. Software

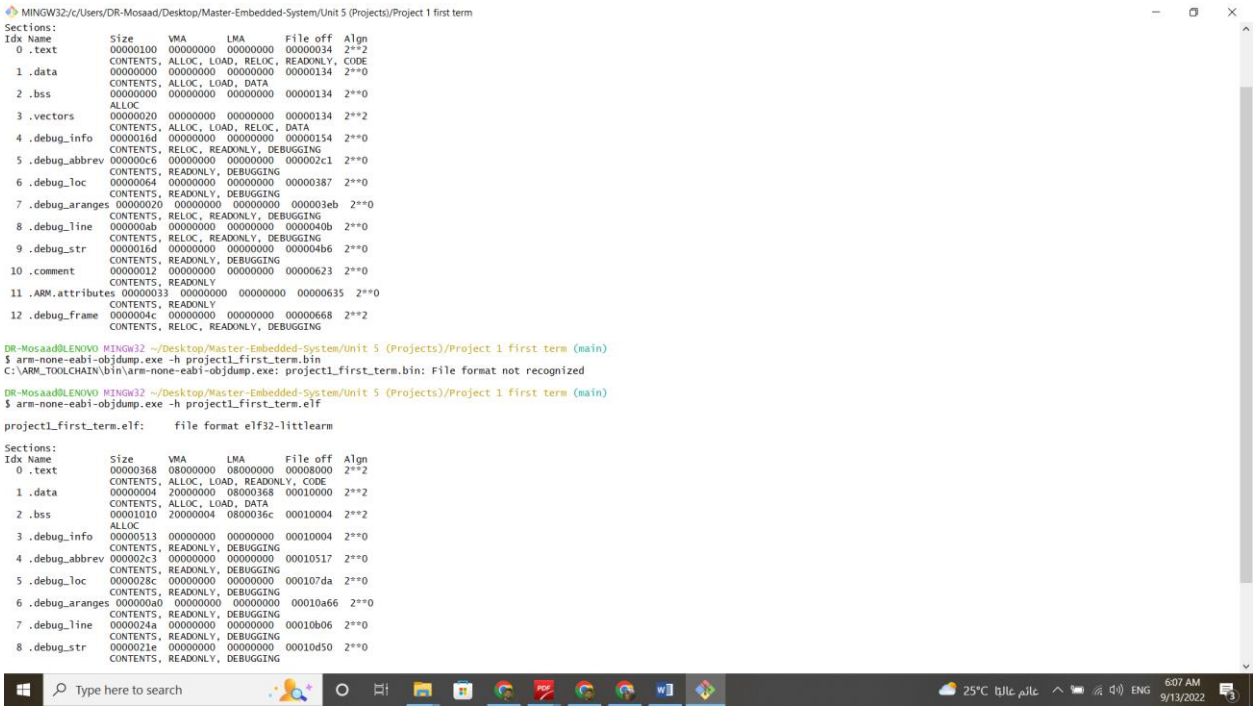
3.1 codes

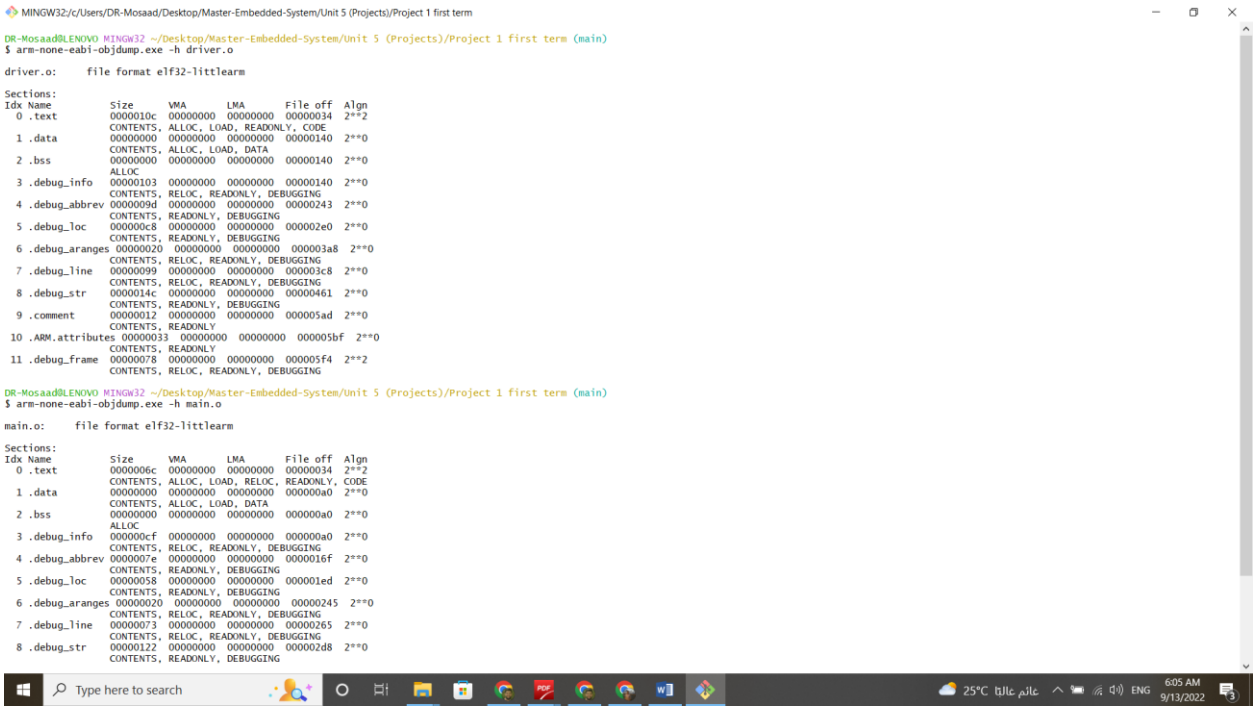
We write codes from scratch:

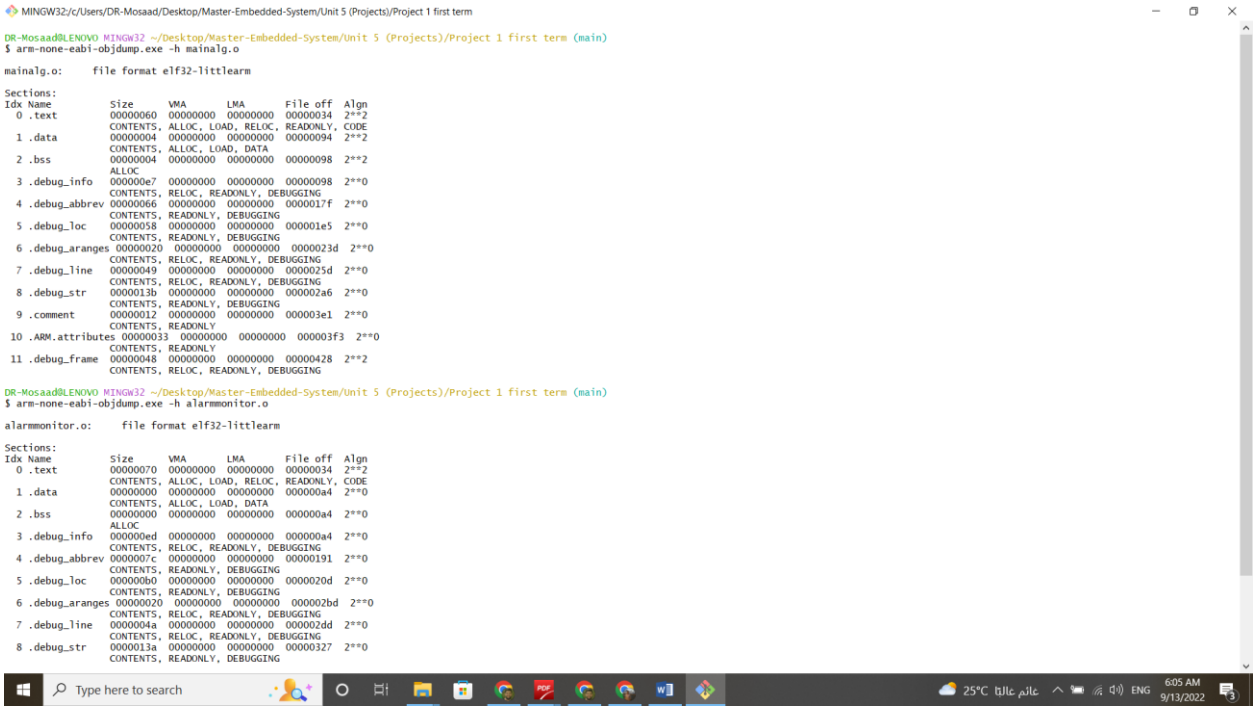
- Makefile
- Linker_script.ld
- Startup.c
- Main.c
- Driver.h/.c
- Mainalg.h/.c
- Alarmmonitor.h/.c
- Get map file for final executable file
  To see them click_here

3.2 Debugger sections

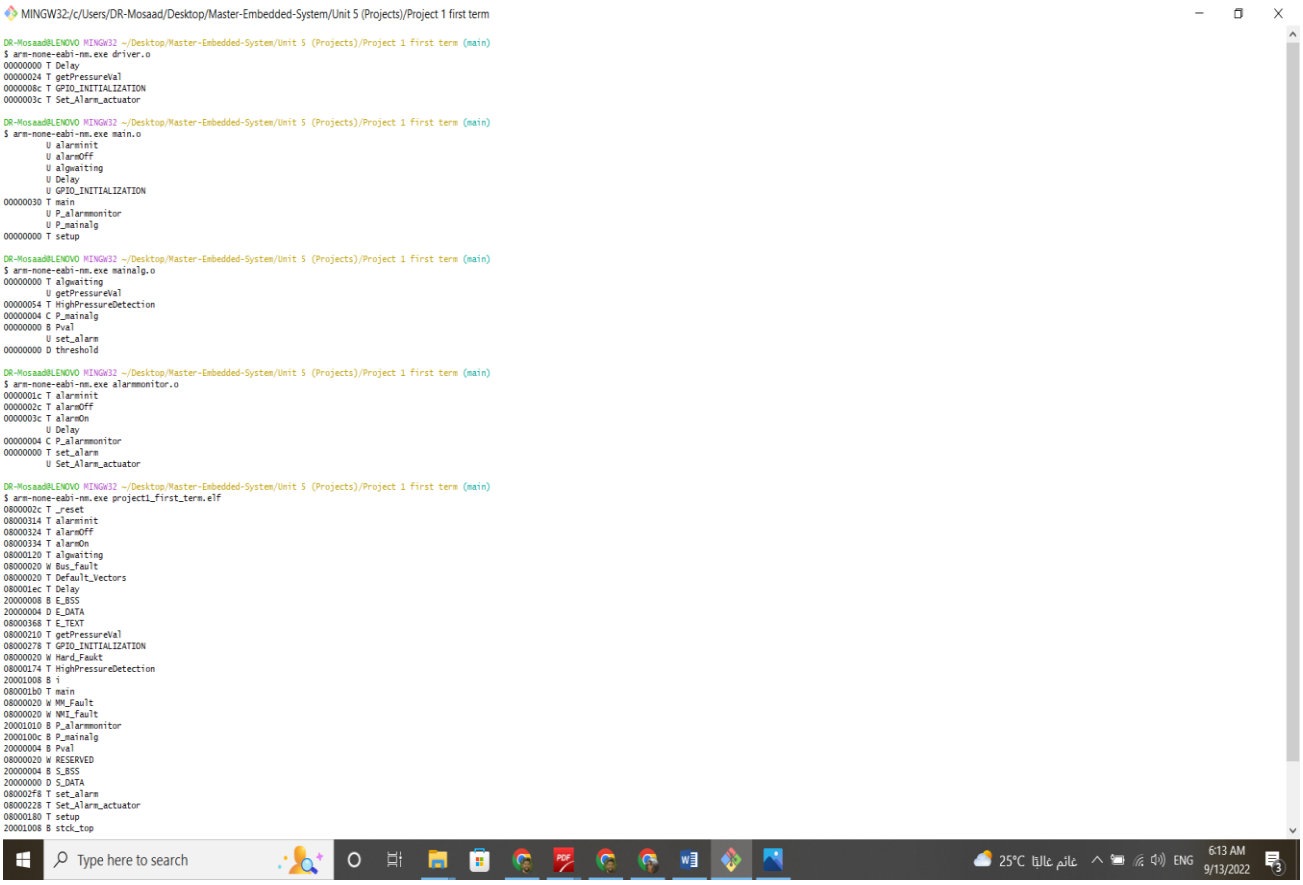
3.2.1 sections



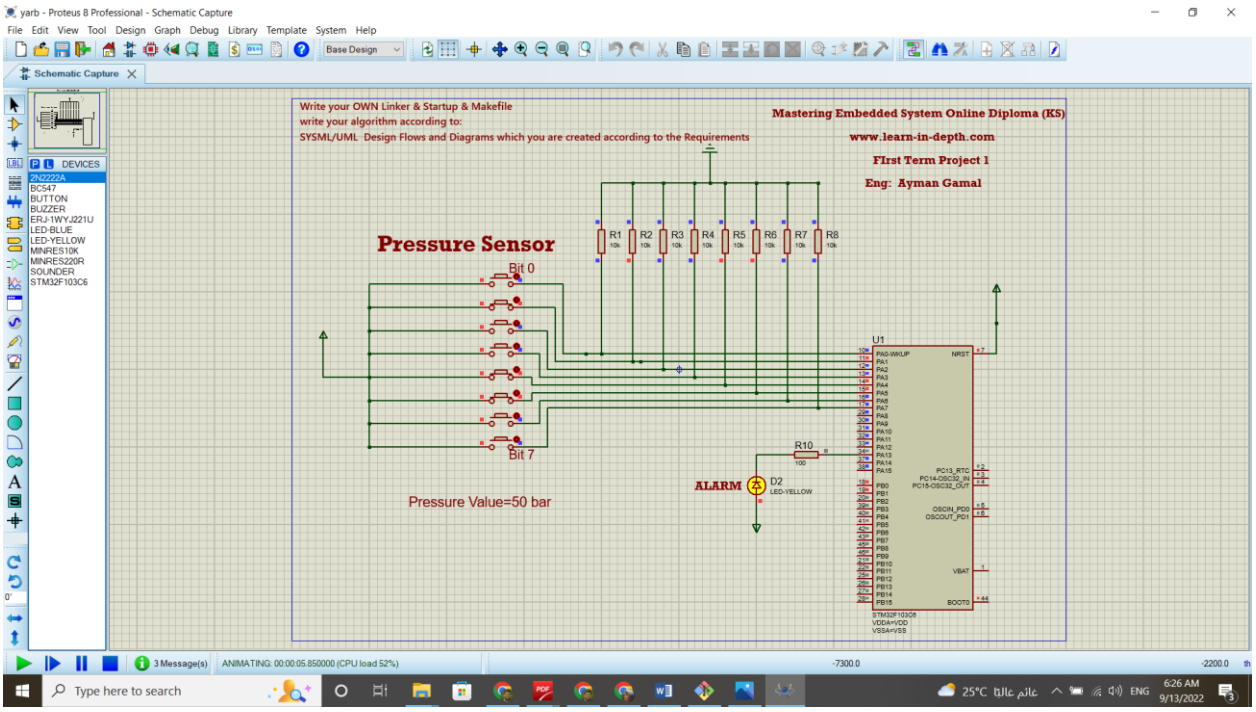

MINGW32:/c/Users/DR-Mosaad/Desktop/Master-Embedded-System/Unit 5 (Projects)/Project 1 first term
DR-Mosaad@LENOVO MINGW32 ~/Desktop/Master-Embedded-System/Unit 5 (Projects)/Project 1 first term (main)
$ arm-none-eabi-objdump.exe -h driver.o
driver.o: file format elf32-littlearm
Sections:
Idx Name Size VMA LMA File off Algn
0 .text 0000010c 00000000 00000000 00000034 2**2
CONTENTS, ALLOC, LOAD, READONLY, CODE
1 .data 00000000 00000000 00000000 00000140 2**0
CONTENTS, ALLOC, LOAD, DATA
2 .bss 00000000 00000000 00000000 00000140 2**0
ALLOC
3 .debug_info 00000103 00000000 00000000 00000140 2**0
CONTENTS, RELOC, READONLY, DEBUGGING
4 .debug_abbrev 0000009d 00000000 00000000 00000243 2**0
CONTENTS, READONLY, DEBUGGING
5 .debug_loc 000000c8 00000000 00000000 000002e0 2**0
CONTENTS, READONLY, DEBUGGING
6 .debug_aranges 00000020 00000000 00000000 000003a8 2**0
CONTENTS, RELOC, READONLY, DEBUGGING
7 .debug_line 00000099 00000000 00000000 000003c8 2**0
CONTENTS, RELOC, READONLY, DEBUGGING
8 .debug_str 0000014c 00000000 00000000 00000461 2**0
CONTENTS, READONLY, DEBUGGING
9 .comment 00000012 00000000 00000000 000005ad 2**0
CONTENTS, READONLY
10 .ARM.attributes 00000033 00000000 00000000 000005bf 2**0
CONTENTS, READONLY
11 .debug_frame 00000078 00000000 00000000 000005f4 2**2
CONTENTS, RELOC, READONLY, DEBUGGING
DR-Mosaad@LENOVO MINGW32 ~/Desktop/Master-Embedded-System/Unit 5 (Projects)/Project 1 first term (main)
$ arm-none-eabi-objdump.exe -h main.o
main.o: file format elf32-littlearm
Sections:
Idx Name Size VMA LMA File off Algn
0 .text 0000006c 00000000 00000000 00000034 2**2
CONTENTS, ALLOC, LOAD, RELOC, READONLY, CODE
1 .data 00000000 00000000 00000000 000000a0 2**0
CONTENTS, ALLOC, LOAD, DATA
2 .bss 00000000 00000000 00000000 000000a0 2**0
ALLOC
3 .debug_info 000000cf 00000000 00000000 000000a0 2**0
CONTENTS, RELOC, READONLY, DEBUGGING
4 .debug_abbrev 0000007e 00000000 00000000 0000016f 2**0
CONTENTS, READONLY, DEBUGGING
5 .debug_loc 00000058 00000000 00000000 000001ed 2**0
CONTENTS, READONLY, DEBUGGING
6 .debug_aranges 00000020 00000000 00000000 00000245 2**0
CONTENTS, RELOC, READONLY, DEBUGGING
7 .debug_line 00000073 00000000 00000000 00000265 2**0
CONTENTS, RELOC, READONLY, DEBUGGING
8 .debug_str 00000122 00000000 00000000 000002d8 2**0
CONTENTS, READONLY, DEBUGGING
Type here to search
25°C
6:05 AM
9/13/2022

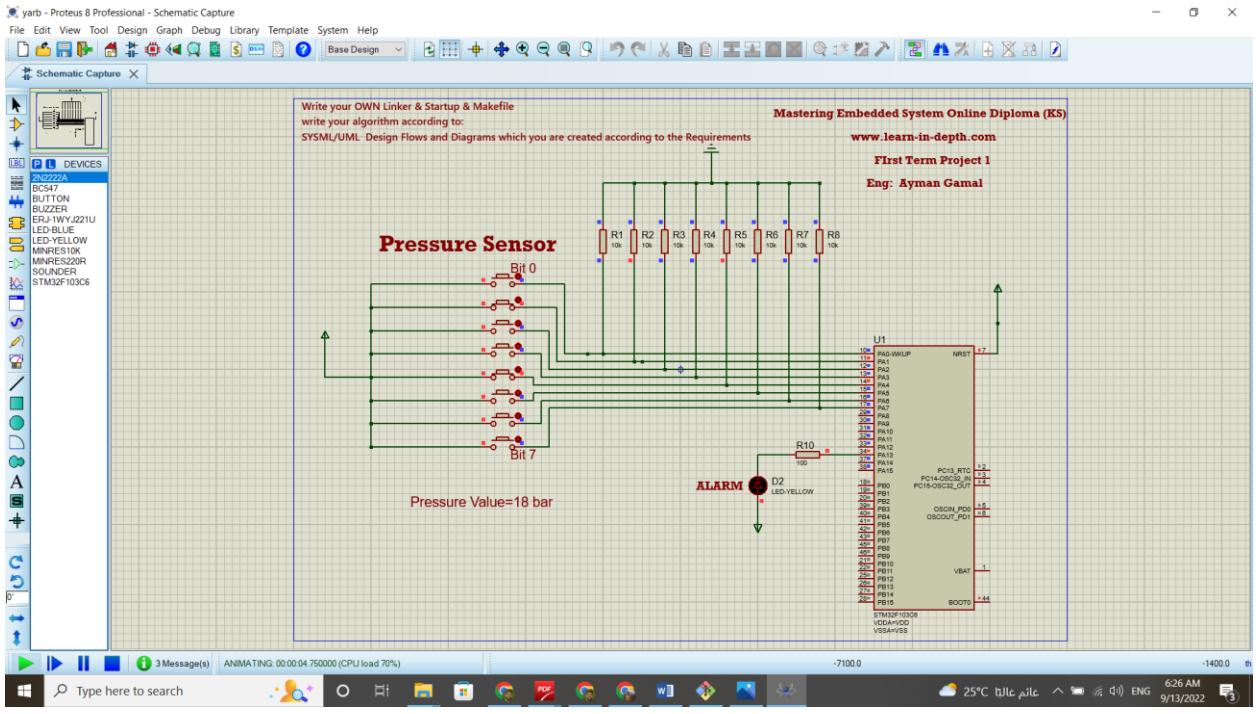

MINGW32:/c/Users/DR-Mosaad/Desktop/Master-Embedded-System/Unit 5 (Projects)/Project 1 first term
DR-Mosaad@LENOVO MINGW32 ~/Desktop/Master-Embedded-System/Unit 5 (Projects)/Project 1 first term (main)
$ arm-none-eabi-objdump.exe -h mainalg.o
mainalg.o: file format elf32-littlearm
Sections:
Idx Name Size VMA LMA File off Algn
0 .text 00000060 00000000 00000000 00000034 2**2
CONTENTS, ALLOC, LOAD, RELOC, READONLY, CODE
1 .data 00000004 00000000 00000000 00000094 2**2
CONTENTS, ALLOC, LOAD, DATA
2 .bss 00000004 00000000 00000000 00000098 2**2
ALLOC
3 .debug_info 000000e7 00000000 00000000 00000098 2**0
CONTENTS, RELOC, READONLY, DEBUGGING
4 .debug_abbrev 00000066 00000000 00000000 0000017f 2**0
CONTENTS, READONLY, DEBUGGING
5 .debug_loc 00000058 00000000 00000000 000001e5 2**0
CONTENTS, READONLY, DEBUGGING
6 .debug_aranges 00000020 00000000 00000000 0000023d 2**0
CONTENTS, RELOC, READONLY, DEBUGGING
7 .debug_line 00000049 00000000 00000000 0000025d 2**0
CONTENTS, RELOC, READONLY, DEBUGGING
8 .debug_str 0000013b 00000000 00000000 000002a6 2**0
CONTENTS, READONLY, DEBUGGING
9 .comment 00000012 00000000 00000000 000003e1 2**0
CONTENTS, READONLY
10 .ARM.attributes 00000033 00000000 00000000 000003f3 2**0
CONTENTS, READONLY
11 .debug_frame 00000048 00000000 00000000 00000428 2**2
CONTENTS, RELOC, READONLY, DEBUGGING
DR-Mosaad@LENOVO MINGW32 ~/Desktop/Master-Embedded-System/Unit 5 (Projects)/Project 1 first term (main)
$ arm-none-eabi-objdump.exe -h alarmmonitor.o
alarmmonitor.o: file format elf32-littlearm
Sections:
Idx Name Size VMA LMA File off Algn
0 .text 00000070 00000000 00000000 00000034 2**2
CONTENTS, ALLOC, LOAD, RELOC, READONLY, CODE
1 .data 00000000 00000000 00000000 000000a4 2**0
CONTENTS, ALLOC, LOAD, DATA
2 .bss 00000000 00000000 00000000 000000a4 2**0
ALLOC
3 .debug_info 000000ed 00000000 00000000 000000a4 2**0
CONTENTS, RELOC, READONLY, DEBUGGING
4 .debug_abbrev 0000007c 00000000 00000000 00000191 2**0
CONTENTS, READONLY, DEBUGGING
5 .debug_loc 000000b0 00000000 00000000 0000020d 2**0
CONTENTS, READONLY, DEBUGGING
6 .debug_aranges 00000020 00000000 00000000 000002bd 2**0
CONTENTS, RELOC, READONLY, DEBUGGING
7 .debug_line 0000004a 00000000 00000000 000002dd 2**0
CONTENTS, RELOC, READONLY, DEBUGGING
8 .debug_str 0000013a 00000000 00000000 00000327 2**0
CONTENTS, READONLY, DEBUGGING
Type here to search
25°C
6:05 AM
9/13/2022

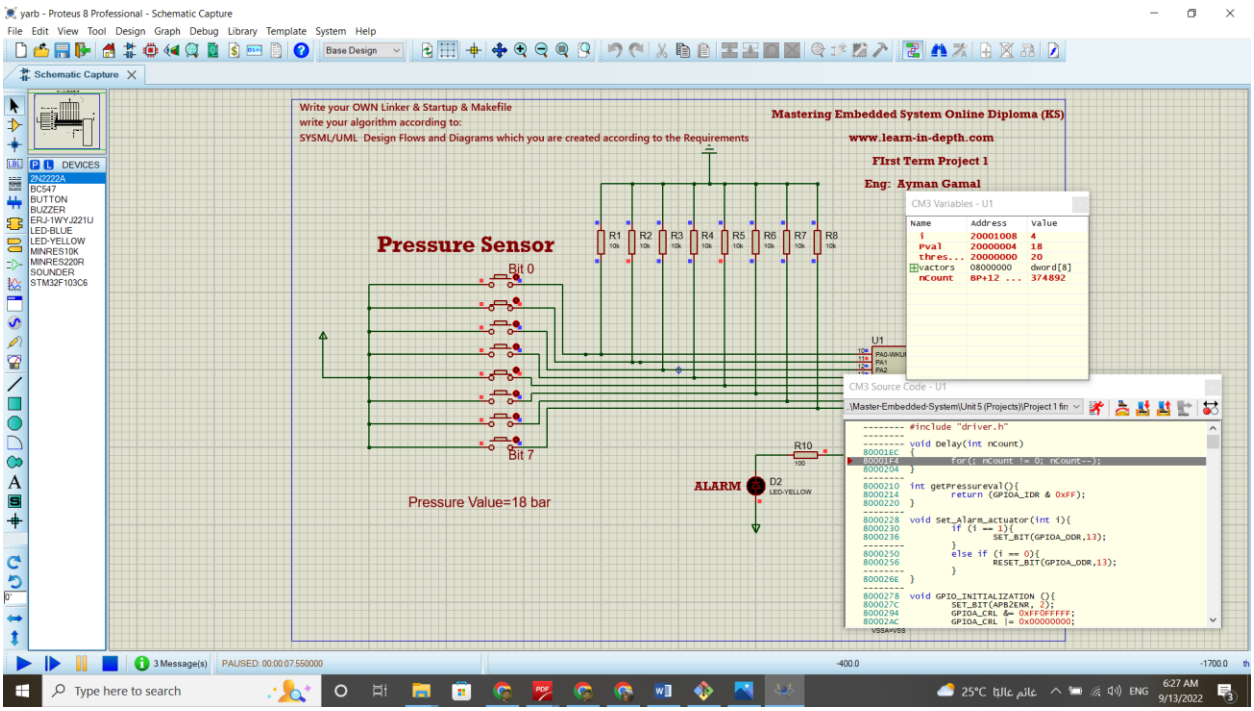
3.2.2 symbols


MINGW32:/c/Users/DR-Mosaad/Desktop/Master-Embedded-System/Unit 5 (Projects)/Project 1 first term
DR-Mosaad@LENOVO MINGW32 ~/Desktop/Master-Embedded-System/Unit 5 (Projects)/Project 1 first term (main)
$ arm-none-eabi-nm.exe driver.o
00000000 T Delay
00000024 T getPressureVal
0000008c T GPIO_INITIALIZATION
0000003c T Set_Alarm_actuator
DR-Mosaad@LENOVO MINGW32 ~/Desktop/Master-Embedded-System/Unit 5 (Projects)/Project 1 first term (main)
$ arm-none-eabi-nm.exe main.o
U alarminit
U alarmOff
U algwaiting
U Delay
U GPIO_INITIALIZATION
00000030 T main
U P_alarmmonitor
U P_mainalg
00000000 T setup
DR-Mosaad@LENOVO MINGW32 ~/Desktop/Master-Embedded-System/Unit 5 (Projects)/Project 1 first term (main)
$ arm-none-eabi-nm.exe mainalg.o
00000000 T algwaiting
U getPressureVal
00000054 T HighPressureDetection
00000004 C P_mainalg
00000000 B Pval
U set_alarm
00000000 D threshold
DR-Mosaad@LENOVO MINGW32 ~/Desktop/Master-Embedded-System/Unit 5 (Projects)/Project 1 first term (main)
$ arm-none-eabi-nm.exe alarmmonitor.o
0000001c T alarminit
0000002c T alarmOff
0000003c T alarmOn
U Delay
00000004 C P_alarmmonitor
00000000 T set_alarm
U Set_Alarm_actuator
DR-Mosaad@LENOVO MINGW32 ~/Desktop/Master-Embedded-System/Unit 5 (Projects)/Project 1 first term (main)
$ arm-none-eabi-nm.exe project1_first_term.elf
0800002c T _reset
08000314 T alarminit
08000324 T alarmOff
08000334 T alarmOn
08000120 T algwaiting
08000020 W Bus_fault
08000020 T Default_Vectors
080001ec T Delay
20000008 B E_BSS
20000004 D E_DATA
08000368 T E_TEXT
08000210 T getPressureVal
08000278 T GPIO_INITIALIZATION
08000020 W Hard_Faukt
08000174 T HighPressureDetection
20001008 B i
080001b0 T main
08000020 W MM_Fault
08000020 W NMI_fault
20001010 B P_alarmmonitor
2000100c B P_mainalg
20000004 B Pval
08000020 W RESERVED
20000004 B S_BSS
20000000 D S_DATA
080002f8 T set_alarm
08000228 T Set_Alarm_actuator
08000180 T setup
20001008 B stck_top
Type here to search
25°C
ENG
6:13 AM
9/13/2022

3.3 Simulation proteus







4. [Youtube_simulation_by_me]