PRESSURE DETECTION PROJECT

1st Project - First Term

BY:

Eng. Abram Samuel

Project GitHub Link : PRESSURE DETECTION - GitHub

GitHub Repository: MASTERING-EMBEDDED-SYSTEMS

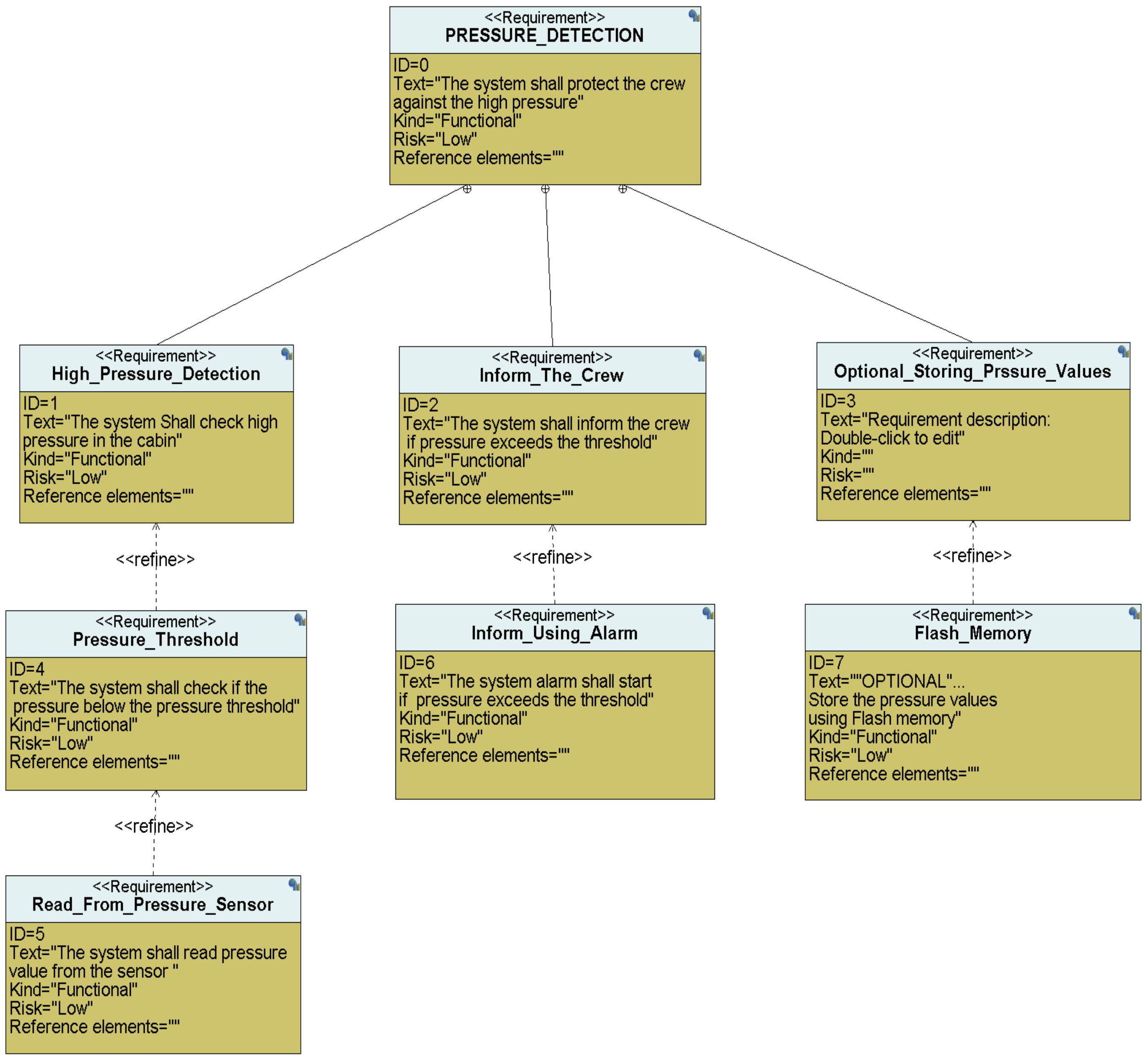
LinkedIn: Abram Samuel

➢ About The Project:

- High Pressure Detection project to inform the cabin crew If the pressure exceeded the threshold informing the crew using Alarm.

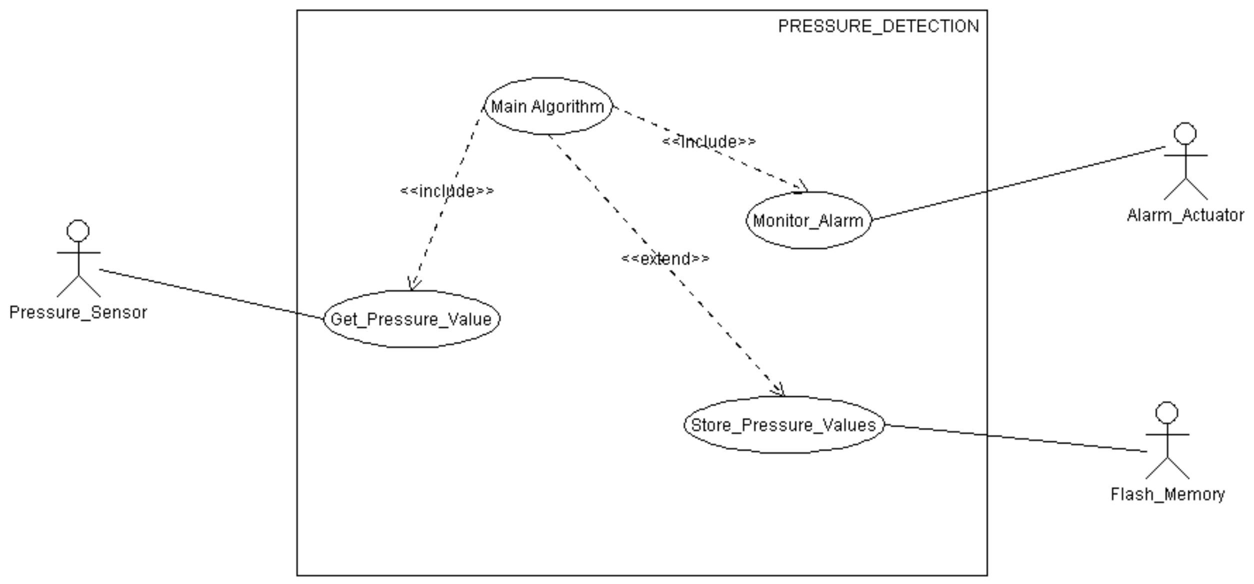
➢ System Architecture & Design Sequence

- **Requirements**

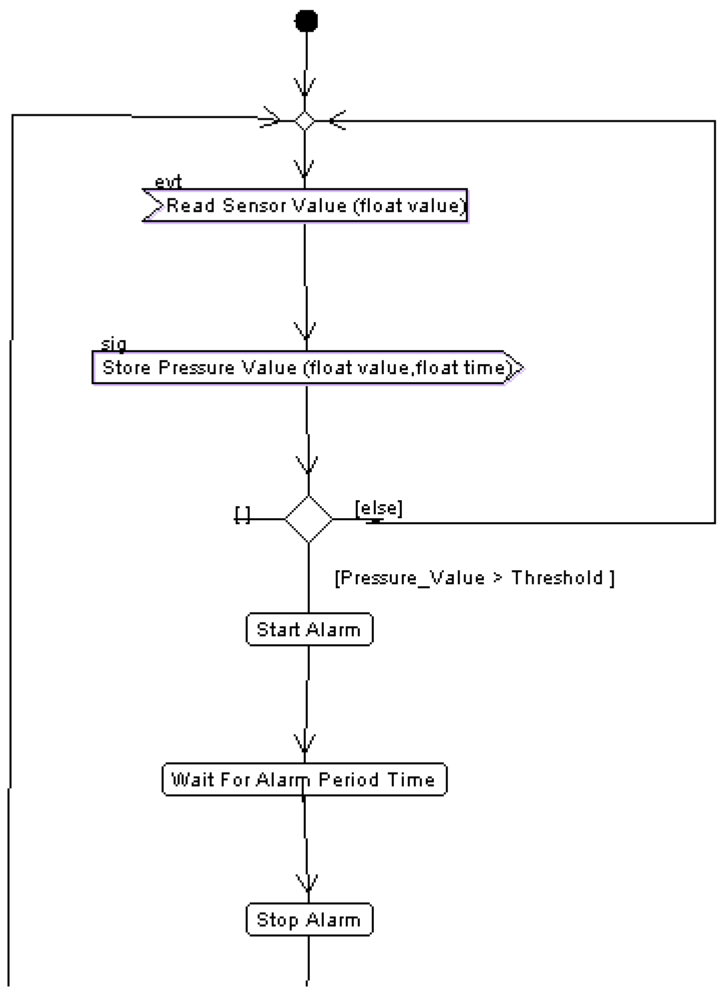


- **System Analysis**

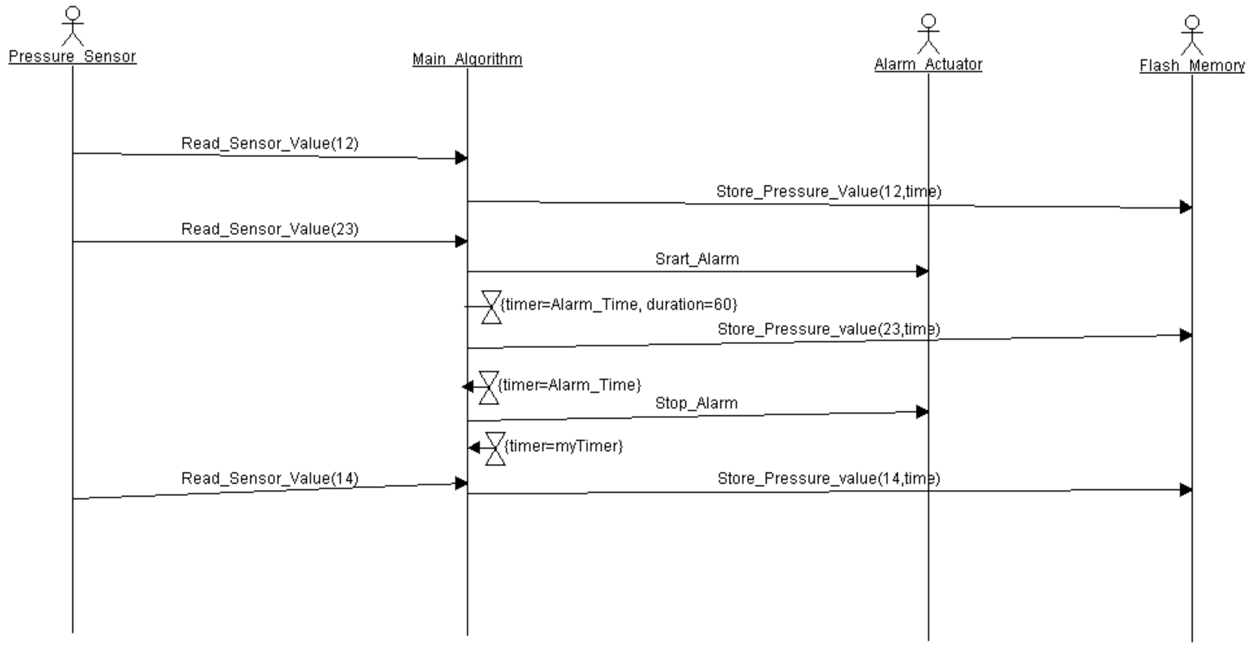
1. Use Case Diagram



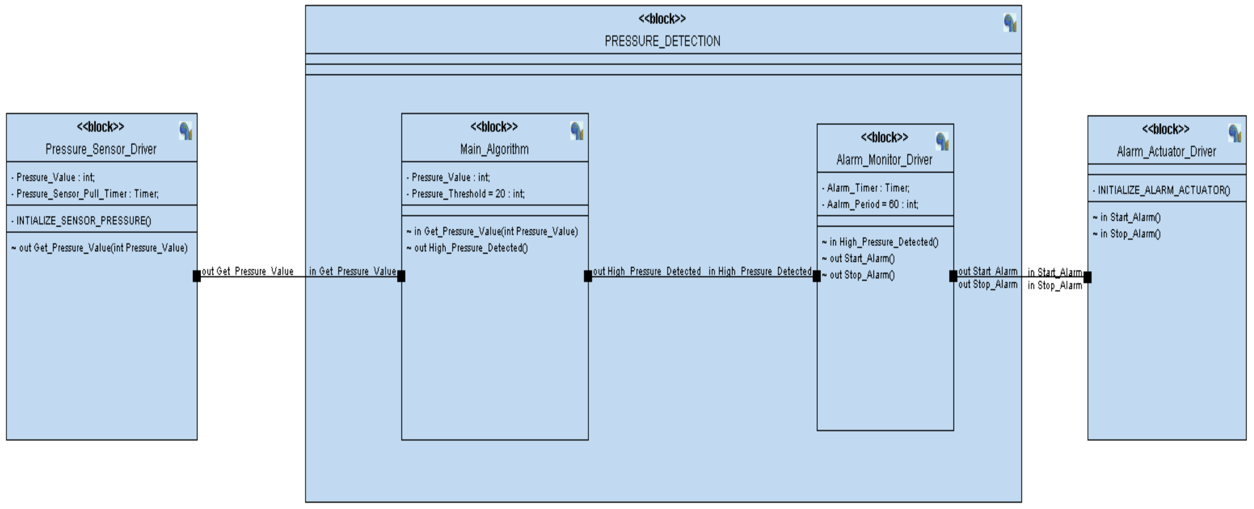
2. Activity Diagram


evt
Read Sensor Value (float value)
sig
Store Pressure Value (float value,float time)
[]
[else]
[Pressure_Value > Threshold]
Start Alarm
Wait For Alarm Period Time
Stop Alarm

3. Sequence Diagram

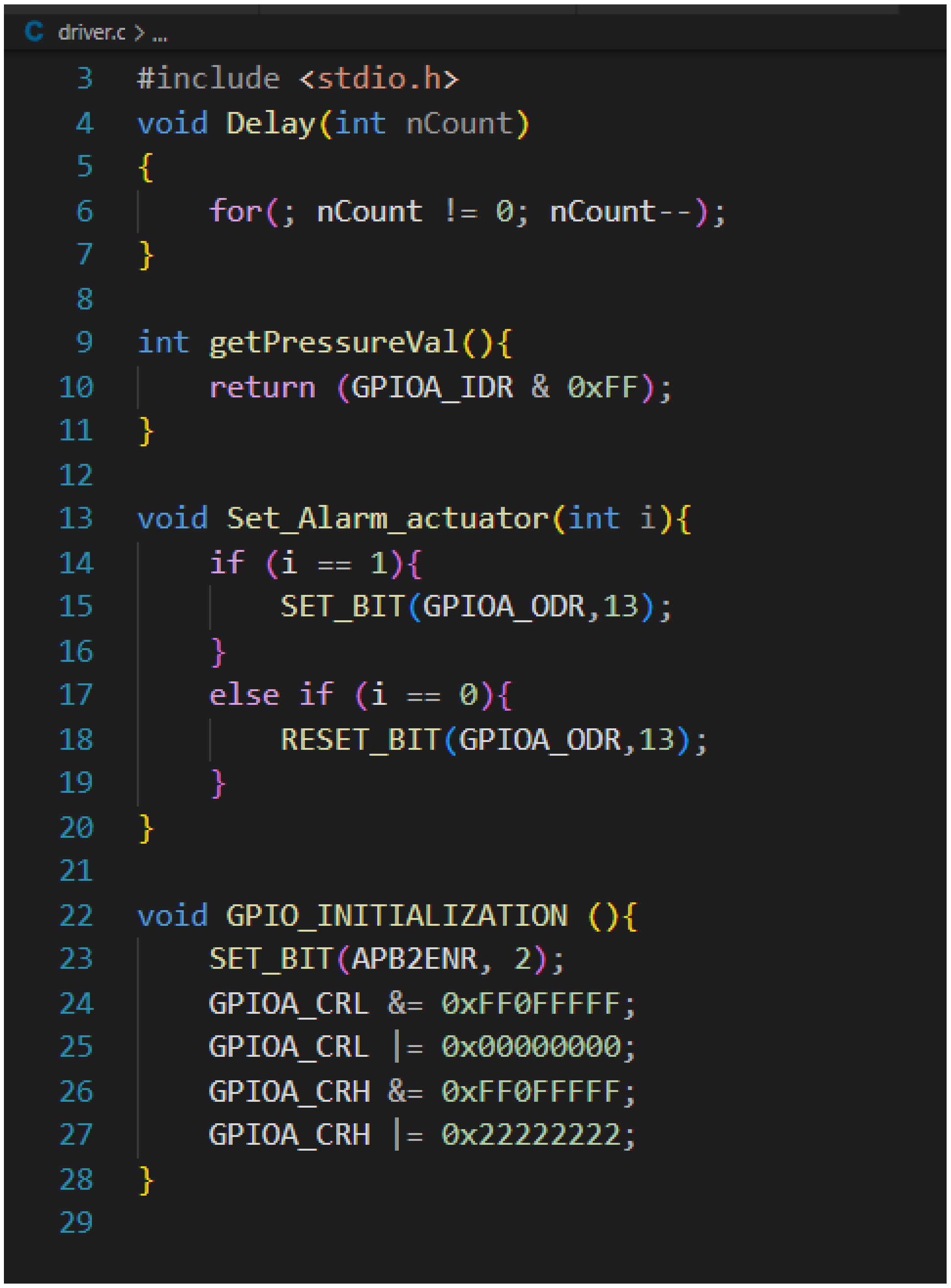


- **System Design**

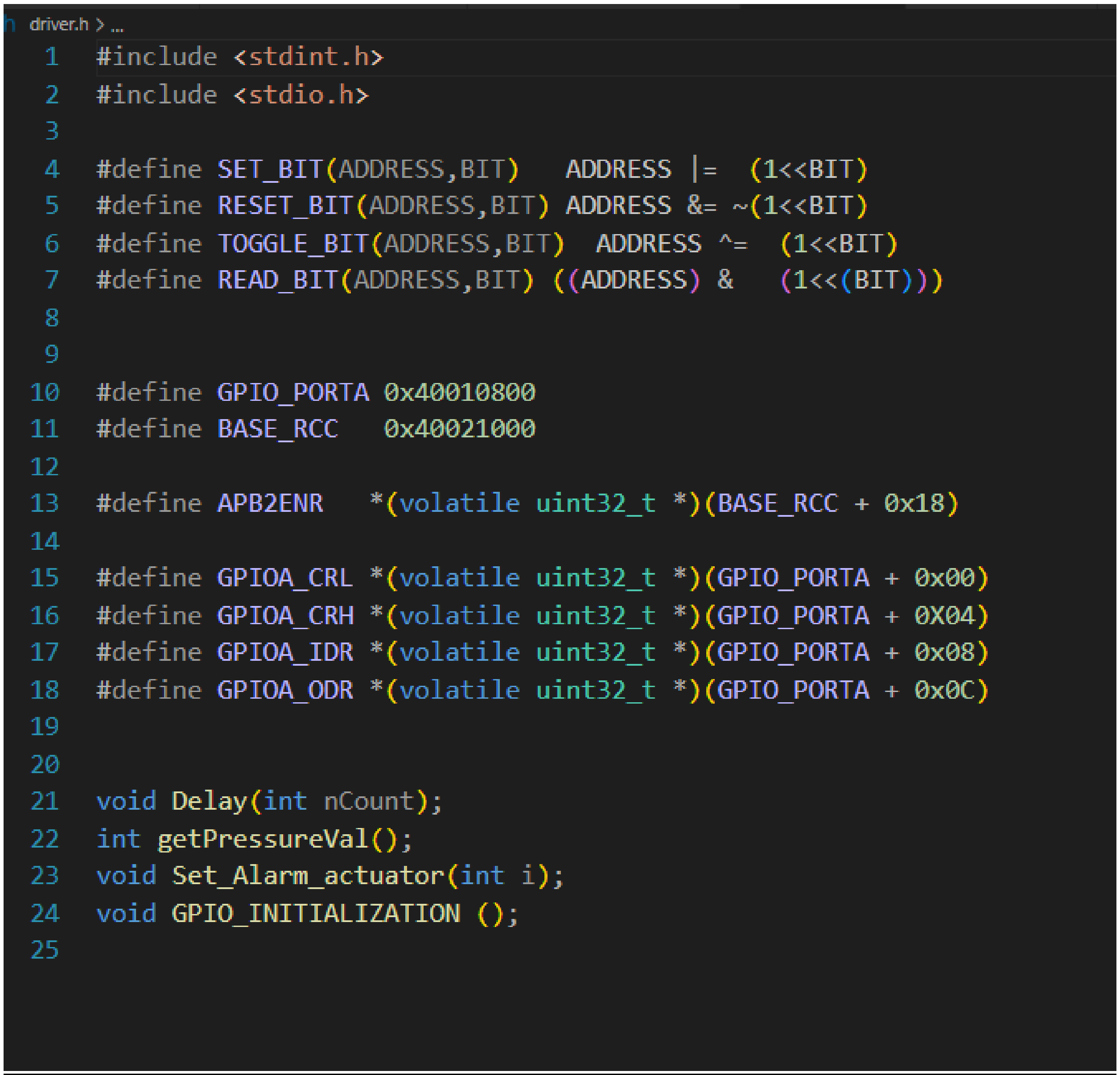


➢ .c & .h & .o files

1. Drive .c & .h & .o files

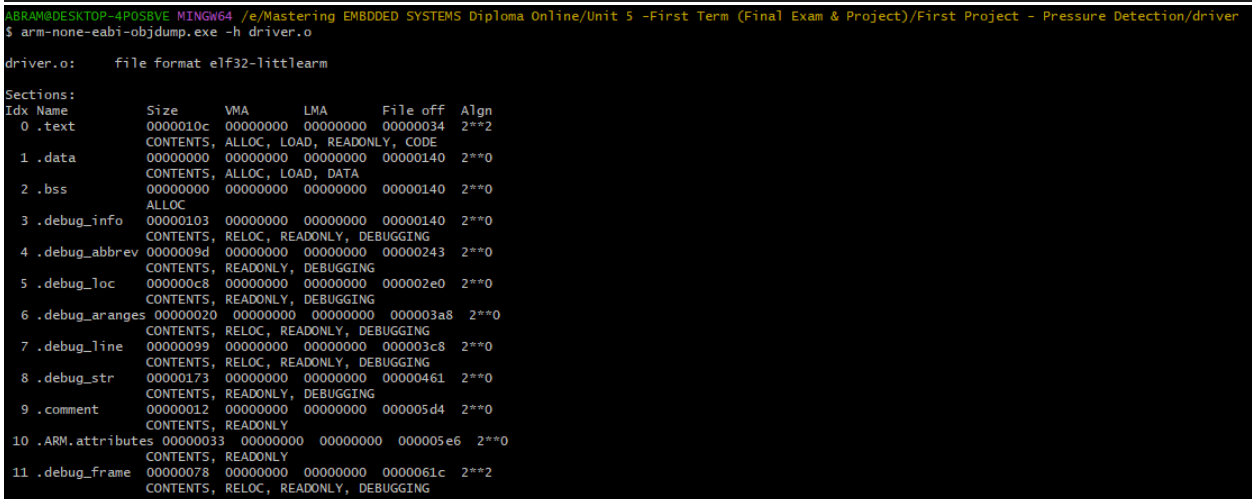


```c
#include <stdio.h>
void Delay(int nCount)
{
    for(; nCount != 0; nCount--);
}

int getPressureVal(){
    return (GPIOA_IDR & 0xFF);
}

void Set_Alarm_actuator(int i){
    if (i == 1){
        SET_BIT(GPIOA_ODR,13);
    }
    else if (i == 0){
        RESET_BIT(GPIOA_ODR,13);
    }
}

void GPIO_INITIALIZATION (){
    SET_BIT(APB2ENR, 2);
    GPIOA_CRL &= 0xFF0FFFFF;
    GPIOA_CRL |= 0x00000000;
    GPIOA_CRH &= 0xFF0FFFFF;
    GPIOA_CRH |= 0x22222222;
}

```



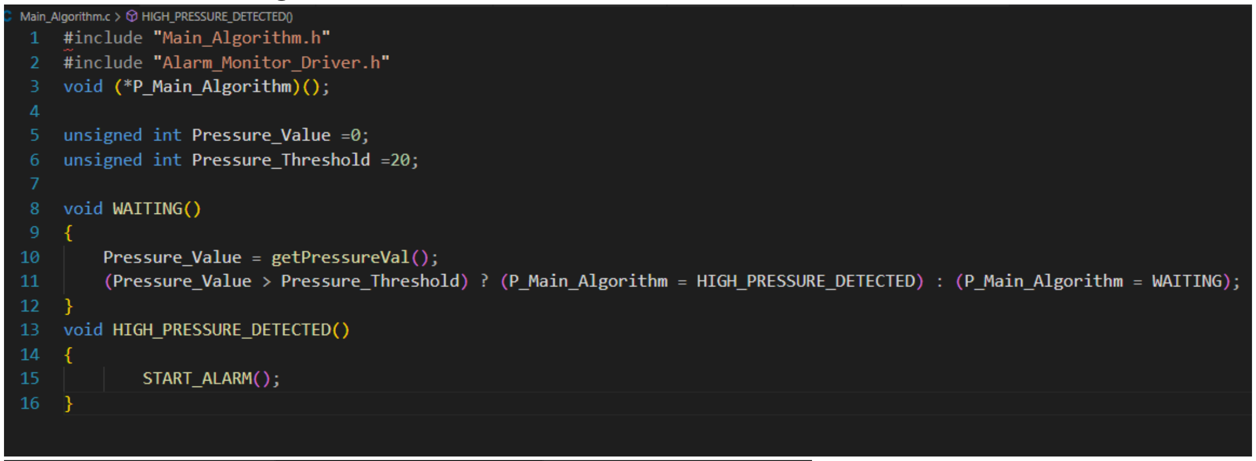
driver.h > ...

```c
#include <stdint.h>
#include <stdio.h>

#define SET_BIT(ADDRESS,BIT)    ADDRESS |=  (1<<BIT)
#define RESET_BIT(ADDRESS,BIT) ADDRESS &= ~(1<<BIT)
#define TOGGLE_BIT(ADDRESS,BIT)  ADDRESS ^=  (1<<BIT)
#define READ_BIT(ADDRESS,BIT) ((ADDRESS) &   (1<<(BIT)))


#define GPIO_PORTA 0x40010800
#define BASE_RCC    0x40021000

#define APB2ENR   *(volatile uint32_t *)(BASE_RCC + 0x18)

#define GPIOA_CRL *(volatile uint32_t *)(GPIO_PORTA + 0x00)
#define GPIOA_CRH *(volatile uint32_t *)(GPIO_PORTA + 0X04)
#define GPIOA_IDR *(volatile uint32_t *)(GPIO_PORTA + 0x08)
#define GPIOA_ODR *(volatile uint32_t *)(GPIO_PORTA + 0x0C)


void Delay(int nCount);
int getPressureVal();
void Set_Alarm_actuator(int i);
void GPIO_INITIALIZATION ();

```

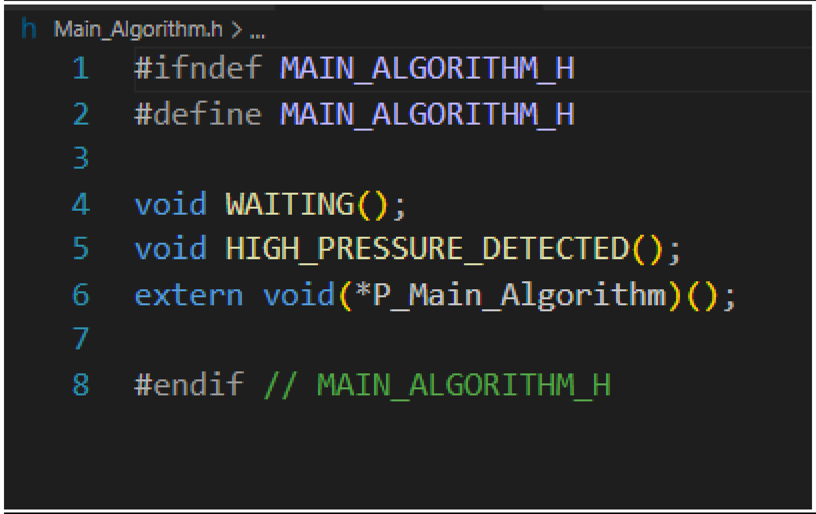


```
ABRAM@DESKTOP-4POSBVE MINGW64 /e/Mastering EMBDDED SYSTEMS Diploma Online/Unit 5 -First Term (Final Exam & Project)/First Project - Pressure Detection/driver
$ arm-none-eabi-objdump.exe -h driver.o

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    00000173  00000000  00000000  00000461  2**0
                  CONTENTS, READONLY, DEBUGGING
  9 .comment      00000012  00000000  00000000  000005d4  2**0
                  CONTENTS, READONLY
 10 .ARM.attributes 00000033  00000000  00000000  000005e6  2**0
                  CONTENTS, READONLY
 11 .debug_frame  00000078  00000000  00000000  0000061c  2**2
                  CONTENTS, RELOC, READONLY, DEBUGGING
```

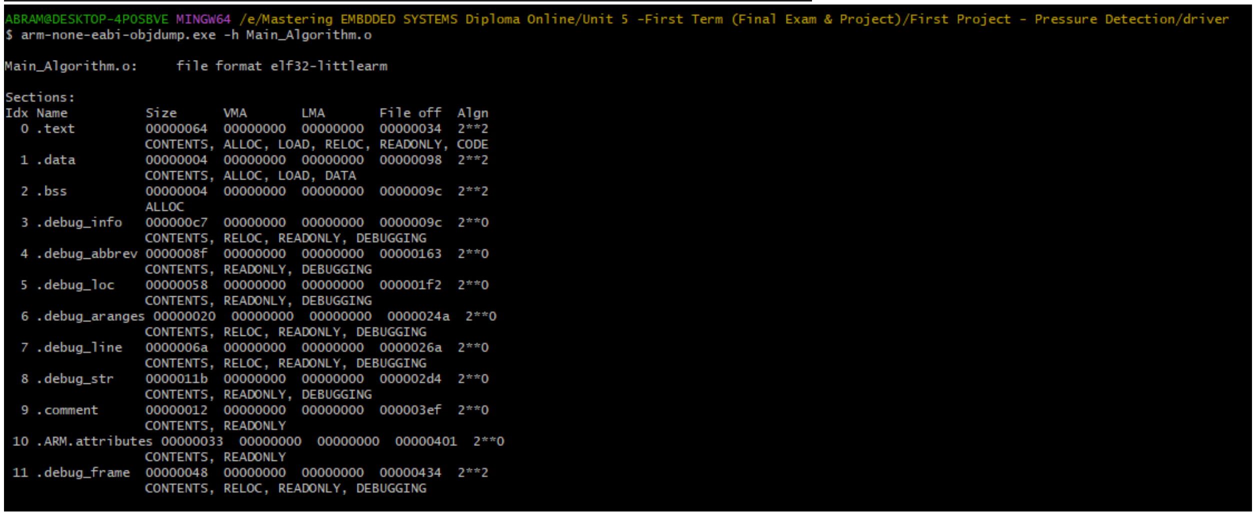
2. Main Algorithm .c & .h & .o files



```c
#include "Main_Algorithm.h"
#include "Alarm_Monitor_Driver.h"
void (*P_Main_Algorithm)();

unsigned int Pressure_Value =0;
unsigned int Pressure_Threshold =20;

void WAITING()
{
    Pressure_Value = getPressureVal();
    (Pressure_Value > Pressure_Threshold) ? (P_Main_Algorithm = HIGH_PRESSURE_DETECTED) : (P_Main_Algorithm = WAITING);
}
void HIGH_PRESSURE_DETECTED()
{
        START_ALARM();
}
```

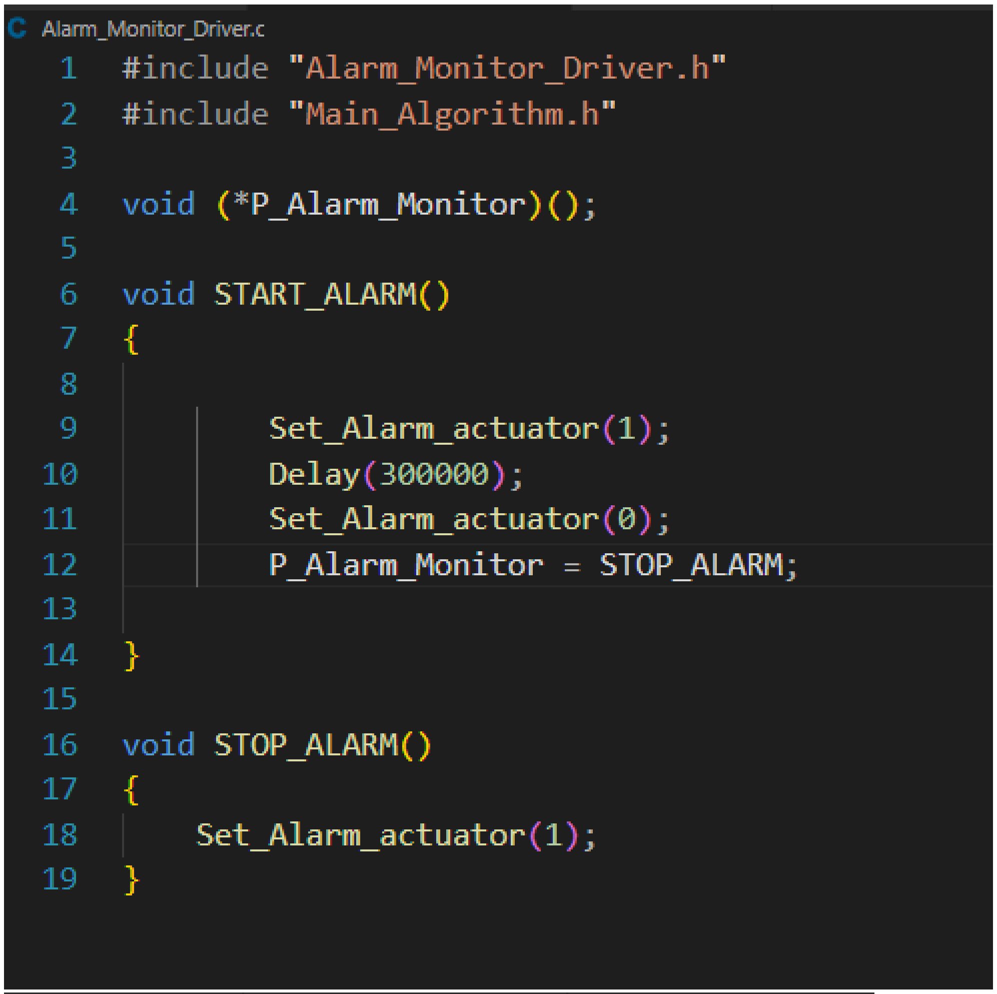


```c
#ifndef MAIN_ALGORITHM_H
#define MAIN_ALGORITHM_H

void WAITING();
void HIGH_PRESSURE_DETECTED();
extern void(*P_Main_Algorithm)();

#endif // MAIN_ALGORITHM_H
```



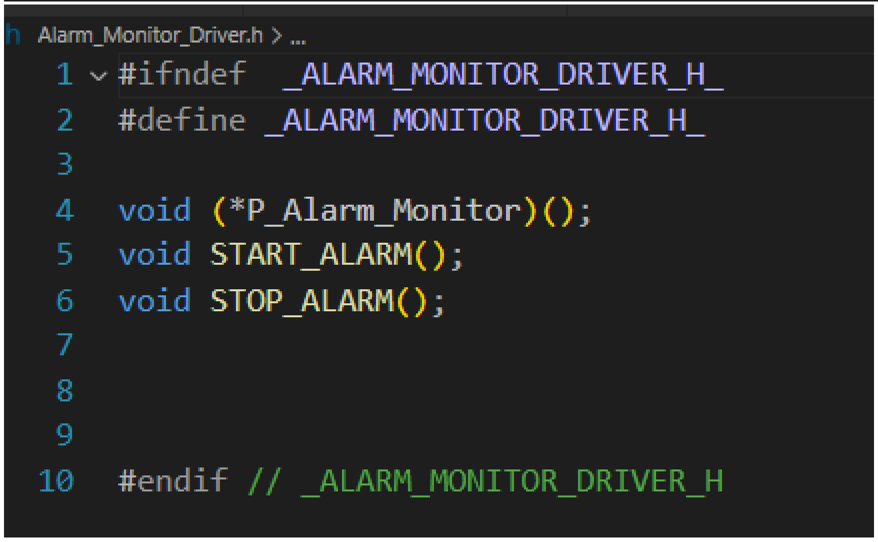
```
ABRAM@DESKTOP-4POSBVE MINGW64 /e/Mastering EMBDDED SYSTEMS Diploma Online/Unit 5 -First Term (Final Exam & Project)/First Project - Pressure Detection/driver
$ arm-none-eabi-objdump.exe -h Main_Algorithm.o

Main_Algorithm.o:     file format elf32-littlearm

Sections:
Idx Name          Size      VMA       LMA       File off  Algn
  0 .text         00000064  00000000  00000000  00000034  2**2
                  CONTENTS, ALLOC, LOAD, RELOC, READONLY, CODE
  1 .data         00000004  00000000  00000000  00000098  2**2
                  CONTENTS, ALLOC, LOAD, DATA
  2 .bss          00000004  00000000  00000000  0000009c  2**2
                  ALLOC
  3 .debug_info   000000c7  00000000  00000000  0000009c  2**0
                  CONTENTS, RELOC, READONLY, DEBUGGING
  4 .debug_abbrev 0000008f  00000000  00000000  00000163  2**0
                  CONTENTS, READONLY, DEBUGGING
  5 .debug_loc    00000058  00000000  00000000  000001f2  2**0
                  CONTENTS, READONLY, DEBUGGING
  6 .debug_aranges 00000020  00000000  00000000  0000024a  2**0
                  CONTENTS, RELOC, READONLY, DEBUGGING
  7 .debug_line   0000006a  00000000  00000000  0000026a  2**0
                  CONTENTS, RELOC, READONLY, DEBUGGING
  8 .debug_str    0000011b  00000000  00000000  000002d4  2**0
                  CONTENTS, READONLY, DEBUGGING
  9 .comment      00000012  00000000  00000000  000003ef  2**0
                  CONTENTS, READONLY
 10 .ARM.attributes 00000033  00000000  00000000  00000401  2**0
                  CONTENTS, READONLY
 11 .debug_frame  00000048  00000000  00000000  00000434  2**2
                  CONTENTS, RELOC, READONLY, DEBUGGING
```

3. Alarm Monitor Driver .c & .h & .o files



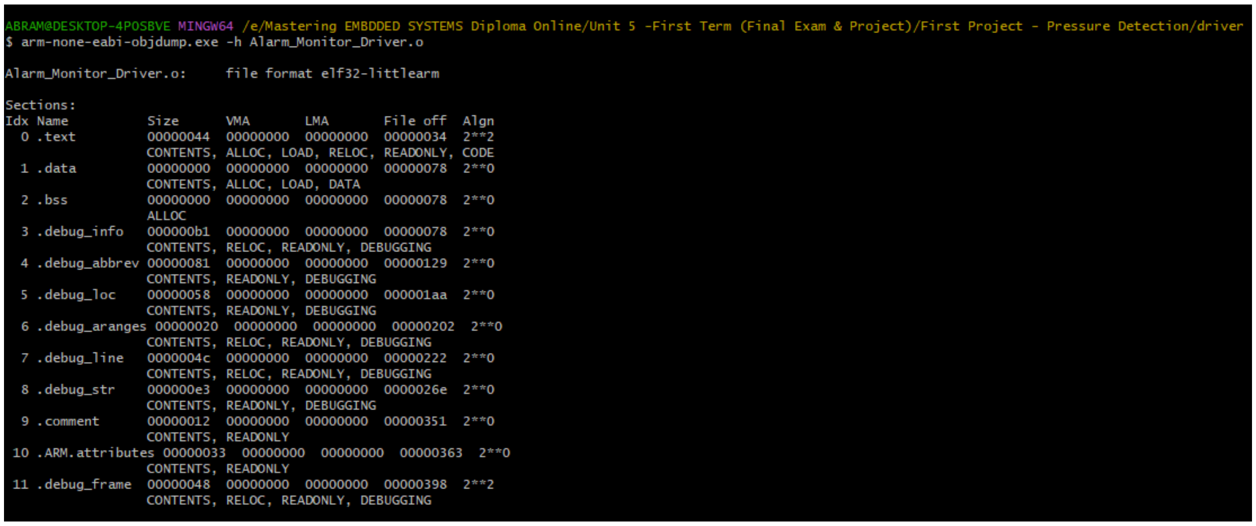
Alarm_Monitor_Driver.c

```c
#include "Alarm_Monitor_Driver.h"
#include "Main_Algorithm.h"

void (*P_Alarm_Monitor)();

void START_ALARM()
{

        Set_Alarm_actuator(1);
        Delay(300000);
        Set_Alarm_actuator(0);
        P_Alarm_Monitor = STOP_ALARM;

}

void STOP_ALARM()
{
    Set_Alarm_actuator(1);
}
```



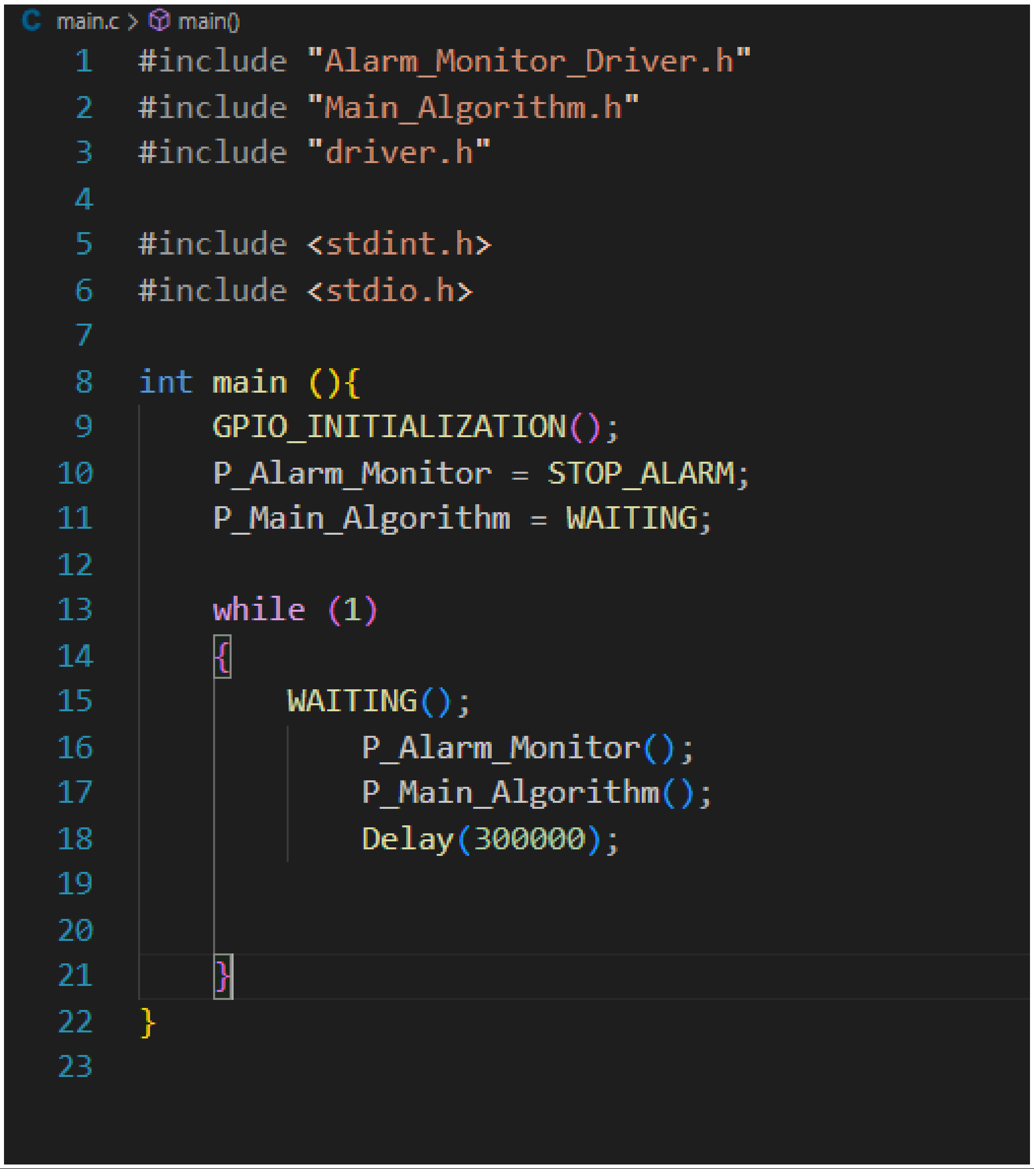
Alarm_Monitor_Driver.h > ...

```c
#ifndef  _ALARM_MONITOR_DRIVER_H_
#define _ALARM_MONITOR_DRIVER_H_

void (*P_Alarm_Monitor)();
void START_ALARM();
void STOP_ALARM();



#endif // _ALARM_MONITOR_DRIVER_H
```



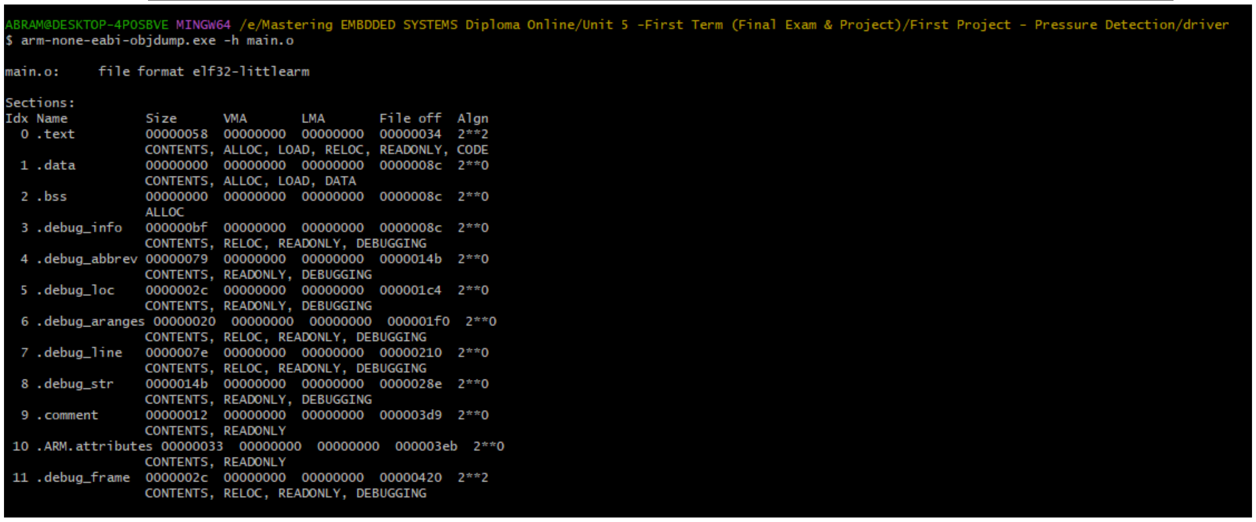
```
ABRAM@DESKTOP-4P0SBVE MINGW64 /e/Mastering EMBDDED SYSTEMS Diploma Online/Unit 5 -First Term (Final Exam & Project)/First Project - Pressure Detection/driver
$ arm-none-eabi-objdump.exe -h Alarm_Monitor_Driver.o

Alarm_Monitor_Driver.o:     file format elf32-littlearm

Sections:
Idx Name          Size      VMA       LMA       File off  Algn
  0 .text         00000044  00000000  00000000  00000034  2**2
                  CONTENTS, ALLOC, LOAD, RELOC, READONLY, CODE
  1 .data         00000000  00000000  00000000  00000078  2**0
                  CONTENTS, ALLOC, LOAD, DATA
  2 .bss          00000000  00000000  00000000  00000078  2**0
                  ALLOC
  3 .debug_info   000000b1  00000000  00000000  00000078  2**0
                  CONTENTS, RELOC, READONLY, DEBUGGING
  4 .debug_abbrev 00000081  00000000  00000000  00000129  2**0
                  CONTENTS, READONLY, DEBUGGING
  5 .debug_loc    00000058  00000000  00000000  000001aa  2**0
                  CONTENTS, READONLY, DEBUGGING
  6 .debug_aranges 00000020  00000000  00000000  00000202  2**0
                  CONTENTS, RELOC, READONLY, DEBUGGING
  7 .debug_line   0000004c  00000000  00000000  00000222  2**0
                  CONTENTS, RELOC, READONLY, DEBUGGING
  8 .debug_str    000000e3  00000000  00000000  0000026e  2**0
                  CONTENTS, READONLY, DEBUGGING
  9 .comment      00000012  00000000  00000000  00000351  2**0
                  CONTENTS, READONLY
 10 .ARM.attributes 00000033  00000000  00000000  00000363  2**0
                  CONTENTS, READONLY
 11 .debug_frame  00000048  00000000  00000000  00000398  2**2
                  CONTENTS, RELOC, READONLY, DEBUGGING
```

4. Main .c & .o files



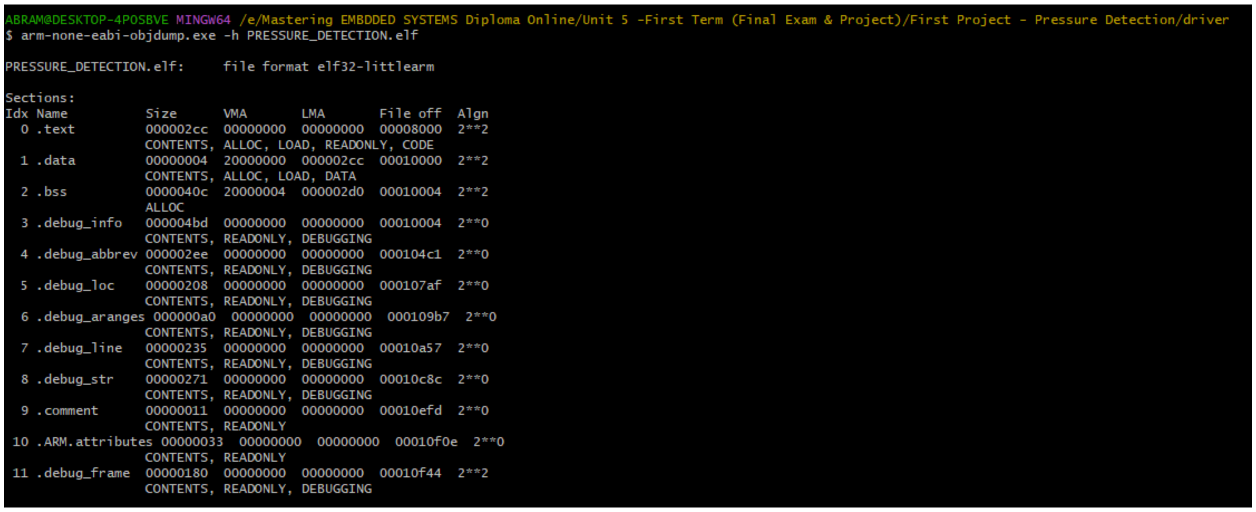
main.c > main()

```c
#include "Alarm_Monitor_Driver.h"
#include "Main_Algorithm.h"
#include "driver.h"

#include <stdint.h>
#include <stdio.h>

int main (){
    GPIO_INITIALIZATION();
    P_Alarm_Monitor = STOP_ALARM;
    P_Main_Algorithm = WAITING;

    while (1)
    {
        WAITING();
            P_Alarm_Monitor();
            P_Main_Algorithm();
            Delay(300000);


    }
}

```



```
ABRAM@DESKTOP-4POSBVE MINGW64 /e/Mastering EMBDDED SYSTEMS Diploma Online/Unit 5 -First Term (Final Exam & Project)/First Project - Pressure Detection/driver
$ 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         00000058  00000000  00000000  00000034  2**2
                  CONTENTS, ALLOC, LOAD, RELOC, READONLY, CODE
  1 .data         00000000  00000000  00000000  0000008c  2**0
                  CONTENTS, ALLOC, LOAD, DATA
  2 .bss          00000000  00000000  00000000  0000008c  2**0
                  ALLOC
  3 .debug_info   000000bf  00000000  00000000  0000008c  2**0
                  CONTENTS, RELOC, READONLY, DEBUGGING
  4 .debug_abbrev 00000079  00000000  00000000  0000014b  2**0
                  CONTENTS, READONLY, DEBUGGING
  5 .debug_loc    0000002c  00000000  00000000  000001c4  2**0
                  CONTENTS, READONLY, DEBUGGING
  6 .debug_aranges 00000020  00000000  00000000  000001f0  2**0
                  CONTENTS, RELOC, READONLY, DEBUGGING
  7 .debug_line   0000007e  00000000  00000000  00000210  2**0
                  CONTENTS, RELOC, READONLY, DEBUGGING
  8 .debug_str    0000014b  00000000  00000000  0000028e  2**0
                  CONTENTS, READONLY, DEBUGGING
  9 .comment      00000012  00000000  00000000  000003d9  2**0
                  CONTENTS, READONLY
 10 .ARM.attributes 00000033  00000000  00000000  000003eb  2**0
                  CONTENTS, READONLY
 11 .debug_frame  0000002c  00000000  00000000  00000420  2**2
                  CONTENTS, RELOC, READONLY, DEBUGGING
```

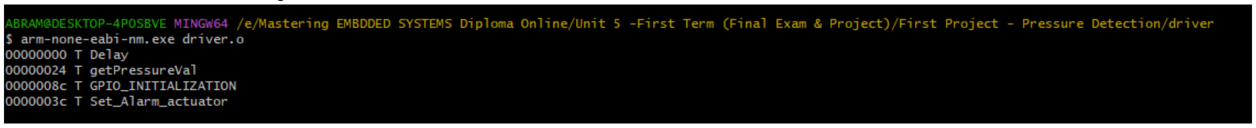
5. PRESSURE DETECTION.elf sections



```
ABRAM@DESKTOP-4POSBVE MINGW64 /e/Mastering EMBDDED SYSTEMS Diploma Online/Unit 5 -First Term (Final Exam & Project)/First Project - Pressure Detection/driver
$ arm-none-eabi-objdump.exe -h PRESSURE_DETECTION.elf

PRESSURE_DETECTION.elf:     file format elf32-littlearm

Sections:
Idx Name          Size      VMA       LMA       File off  Algn
  0 .text         000002cc  00000000  00000000  00008000  2**2
                  CONTENTS, ALLOC, LOAD, READONLY, CODE
  1 .data         00000004  20000000  000002cc  00010000  2**2
                  CONTENTS, ALLOC, LOAD, DATA
  2 .bss          0000040c  20000004  000002d0  00010004  2**2
                  ALLOC
  3 .debug_info   000004bd  00000000  00000000  00010004  2**0
                  CONTENTS, READONLY, DEBUGGING
  4 .debug_abbrev 000002ee  00000000  00000000  000104c1  2**0
                  CONTENTS, READONLY, DEBUGGING
  5 .debug_loc    00000208  00000000  00000000  000107af  2**0
                  CONTENTS, READONLY, DEBUGGING
  6 .debug_aranges 000000a0  00000000  00000000  000109b7  2**0
                  CONTENTS, READONLY, DEBUGGING
  7 .debug_line   00000235  00000000  00000000  00010a57  2**0
                  CONTENTS, READONLY, DEBUGGING
  8 .debug_str    00000271  00000000  00000000  00010c8c  2**0
                  CONTENTS, READONLY, DEBUGGING
  9 .comment      00000011  00000000  00000000  00010efd  2**0
                  CONTENTS, READONLY
 10 .ARM.attributes 00000033  00000000  00000000  00010f0e  2**0
                  CONTENTS, READONLY
 11 .debug_frame  00000180  00000000  00000000  00010f44  2**2
                  CONTENTS, READONLY, DEBUGGING
```

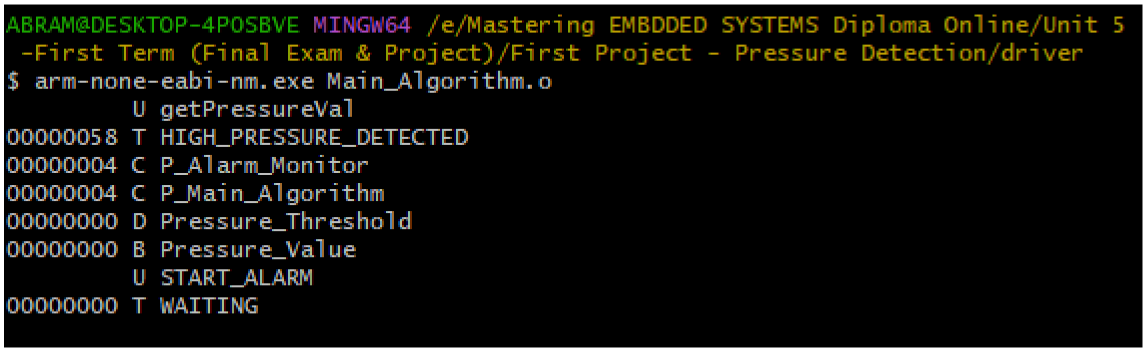
➢ Symbols of the files

1. Driver symbols



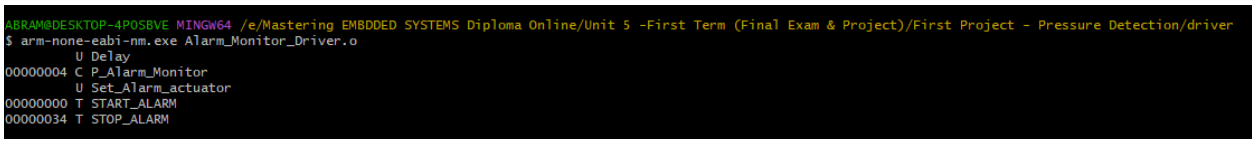
```
ABRAM@DESKTOP-4POSBVE MINGW64 /e/Mastering EMBDDED SYSTEMS Diploma Online/Unit 5 -First Term (Final Exam & Project)/First Project - Pressure Detection/driver
$ arm-none-eabi-nm.exe driver.o
00000000 T Delay
00000024 T getPressureVal
0000008c T GPIO_INITIALIZATION
0000003c T Set_Alarm_actuator
```

2. Main Algorithm symbols



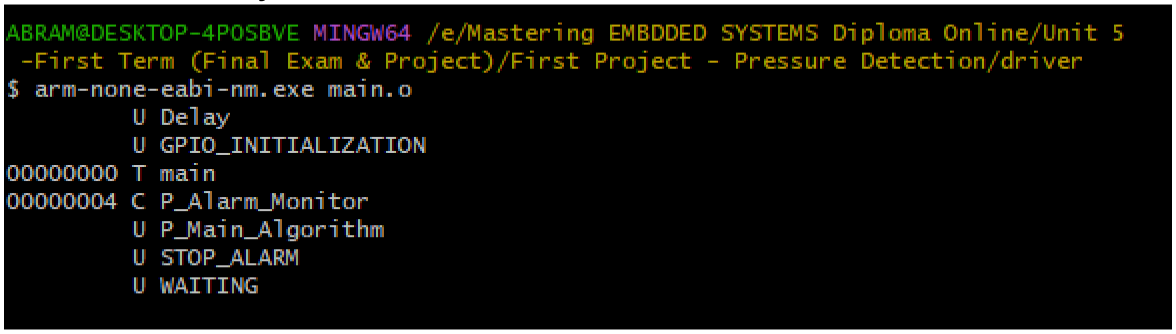
```
ABRAM@DESKTOP-4POSBVE MINGW64 /e/Mastering EMBDDED SYSTEMS Diploma Online/Unit 5
 -First Term (Final Exam & Project)/First Project - Pressure Detection/driver
$ arm-none-eabi-nm.exe Main_Algorithm.o
         U getPressureVal
00000058 T HIGH_PRESSURE_DETECTED
00000004 C P_Alarm_Monitor
00000004 C P_Main_Algorithm
00000000 D Pressure_Threshold
00000000 B Pressure_Value
         U START_ALARM
00000000 T WAITING
```

3. Alarm Monitor Driver symbols



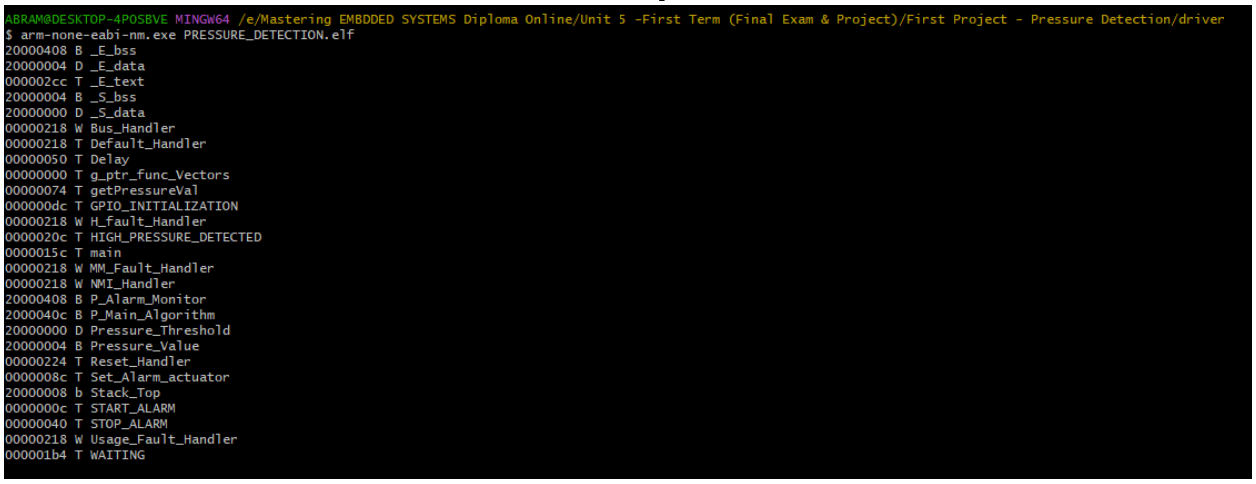
```
ABRAM@DESKTOP-4POSBVE MINGW64 /e/Mastering EMBDDED SYSTEMS Diploma Online/Unit 5 -First Term (Final Exam & Project)/First Project - Pressure Detection/driver
$ arm-none-eabi-nm.exe Alarm_Monitor_Driver.o
         U Delay
00000004 C P_Alarm_Monitor
         U Set_Alarm_actuator
00000000 T START_ALARM
00000034 T STOP_ALARM
```

5. Main symbols



```
ABRAM@DESKTOP-4POSBVE MINGW64 /e/Mastering EMBDDED SYSTEMS Diploma Online/Unit 5
 -First Term (Final Exam & Project)/First Project - Pressure Detection/driver
$ arm-none-eabi-nm.exe main.o
         U Delay
         U GPIO_INITIALIZATION
00000000 T main
00000004 C P_Alarm_Monitor
         U P_Main_Algorithm
         U STOP_ALARM
         U WAITING
```

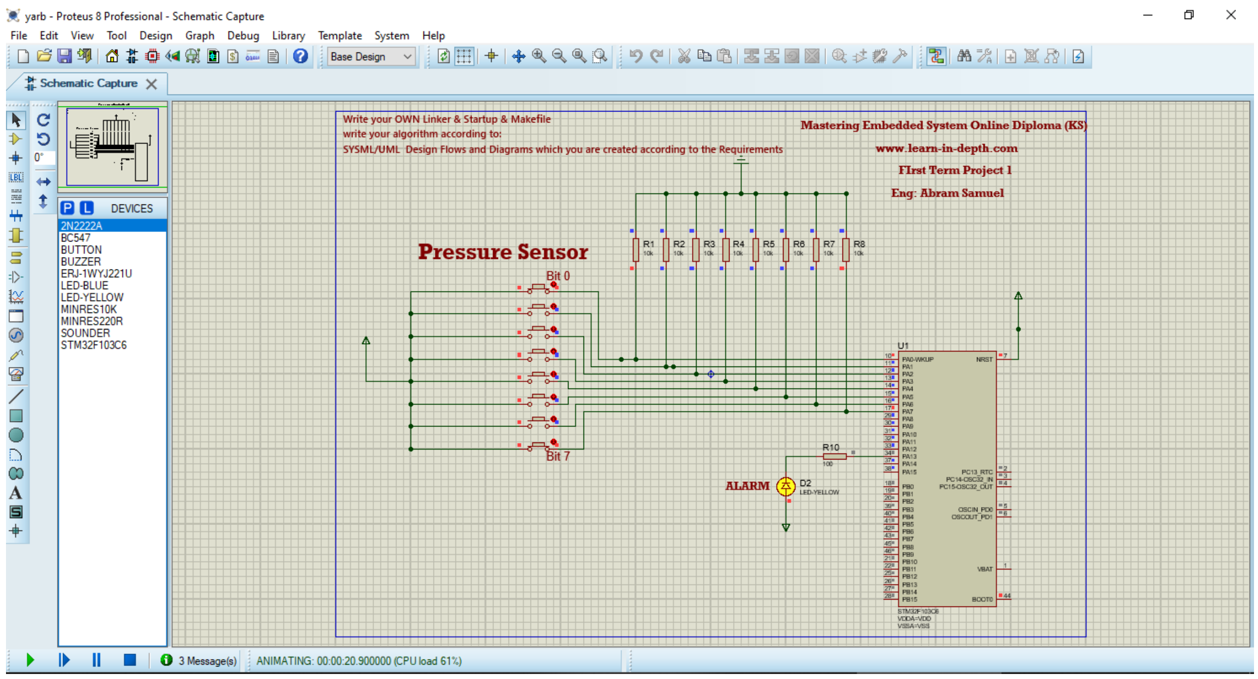
6. PRESSURE DETECTION.elf symbols



```
ABRAM@DESKTOP-4POSBVE MINGW64 /e/Mastering EMBDDED SYSTEMS Diploma Online/Unit 5 -First Term (Final Exam & Project)/First Project - Pressure Detection/driver
$ arm-none-eabi-nm.exe PRESSURE_DETECTION.elf
20000408 B _E_bss
20000004 D _E_data
000002cc T _E_text
20000004 B _S_bss
20000000 D _S_data
00000218 W Bus_Handler
00000218 T Default_Handler
00000050 T Delay
00000000 T g_ptr_func_Vectors
00000074 T getPressureVal
000000dc T GPIO_INITIALIZATION
00000218 W H_fault_Handler
0000020c T HIGH_PRESSURE_DETECTED
0000015c T main
00000218 W MM_Fault_Handler
00000218 W NMI_Handler
20000408 B P_Alarm_Monitor
2000040c B P_Main_Algorithm
20000000 D Pressure_Threshold
20000004 B Pressure_Value
00000224 T Reset_Handler
0000008c T Set_Alarm_actuator
20000008 b Stack_Top
0000000c T START_ALARM
00000040 T STOP_ALARM
00000218 W Usage_Fault_Handler
000001b4 T WAITING
```

➢ Simulation Results with Description

- Alarm ON
  Alarm start blinking when pressure exceeds the
  threshold (20 Bar) and waiting for another reading from
  pressure sensor



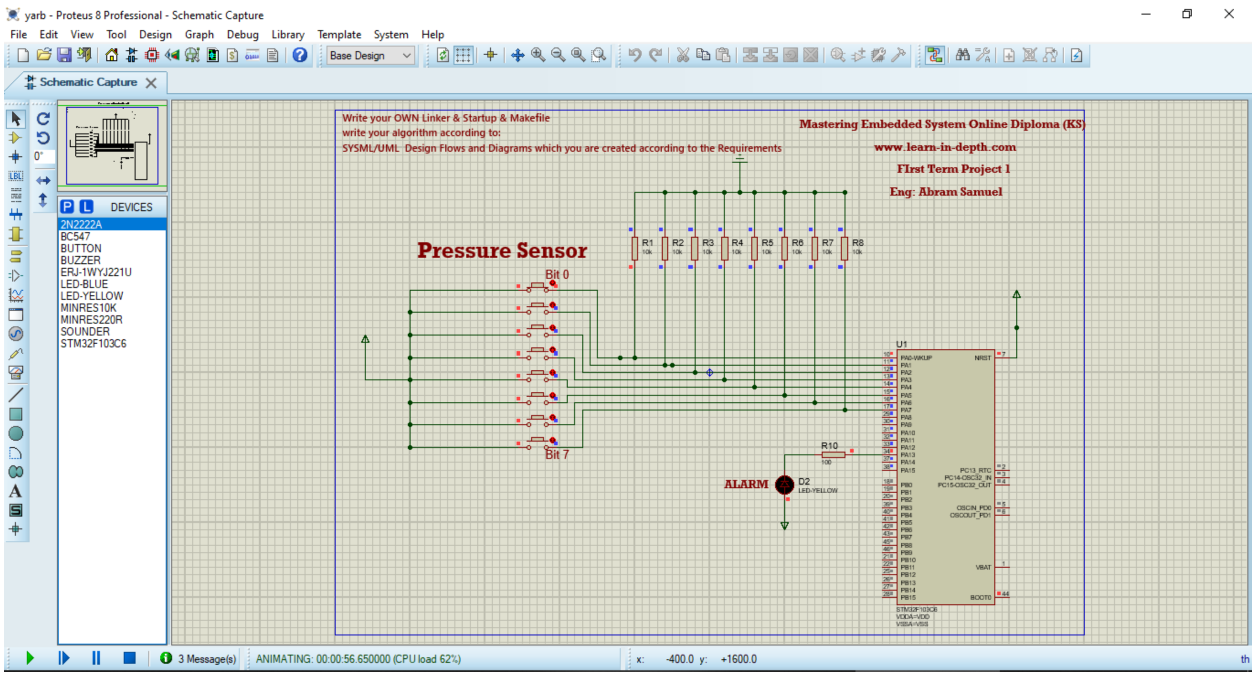
yarb - Proteus 8 Professional - Schematic Capture

Write your OWN Linker & Startup & Makefile
write your algorithm according to:
SYSML/UML Design Flows and Diagrams which you are created according to the Requirements

Mastering Embedded System Online Diploma (KS)
www.learn-in-depth.com
FIrst Term Project 1
Eng: Abram Samuel

3 Message(s)
ANIMATING: 00:00:20.900000 (CPU load 61%)

- Alarm OFF
  The alarm is OFF and when the pressure is below the threshold (20 Bar). It will start blinking when the pressure sensor reading is over the threshold (20 Bar).



Write your OWN Linker & Startup & Makefile
write your algorithm according to:
SYSML/UML Design Flows and Diagrams which you are created according to the Requirements

Mastering Embedded System Online Diploma (KS)
www.learn-in-depth.com
FIrst Term Project 1
Eng: Abram Samuel

3 Message(s)
ANIMATING: 00:00:56.650000 (CPU load 62%)
x: -400.0 y: +1600.0