First Term (Final Project No.1)

Project Name : Pressure_Controller_TM

Eng. Taha Mohamed Taha Ragab Hamed

Under Supervision : Eng. Keroles Shenouda

Drive: https://drive.google.com/drive/folders/1G5EyYpLFo2WVEJhHGYiJXmCJJSIefbIH?usp=sharing

GitHub: https://github.com/Taha249/Master_Embedded_System_Diploma.git

Page Email: taha.mo7amed24@gmail.com

2023/2024

Contents

1- Project Description

Create a Pressure Controller

A client expects to 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.

The system should have:

1- Requirement Diagram

2- System Analysis

- Use Case Diagram

- Activity Diagram

- Sequence Diagram

3- System Design(Modules with its own state machines)

4- You have write Embedded C codes consists of modules

- .c/.h files of each module

- MakeFile

- Startup.c

- Linker.ld

5- In the driver.c file we provide the following APIS

- void Delay(int nCount);

- int detPressureVal();

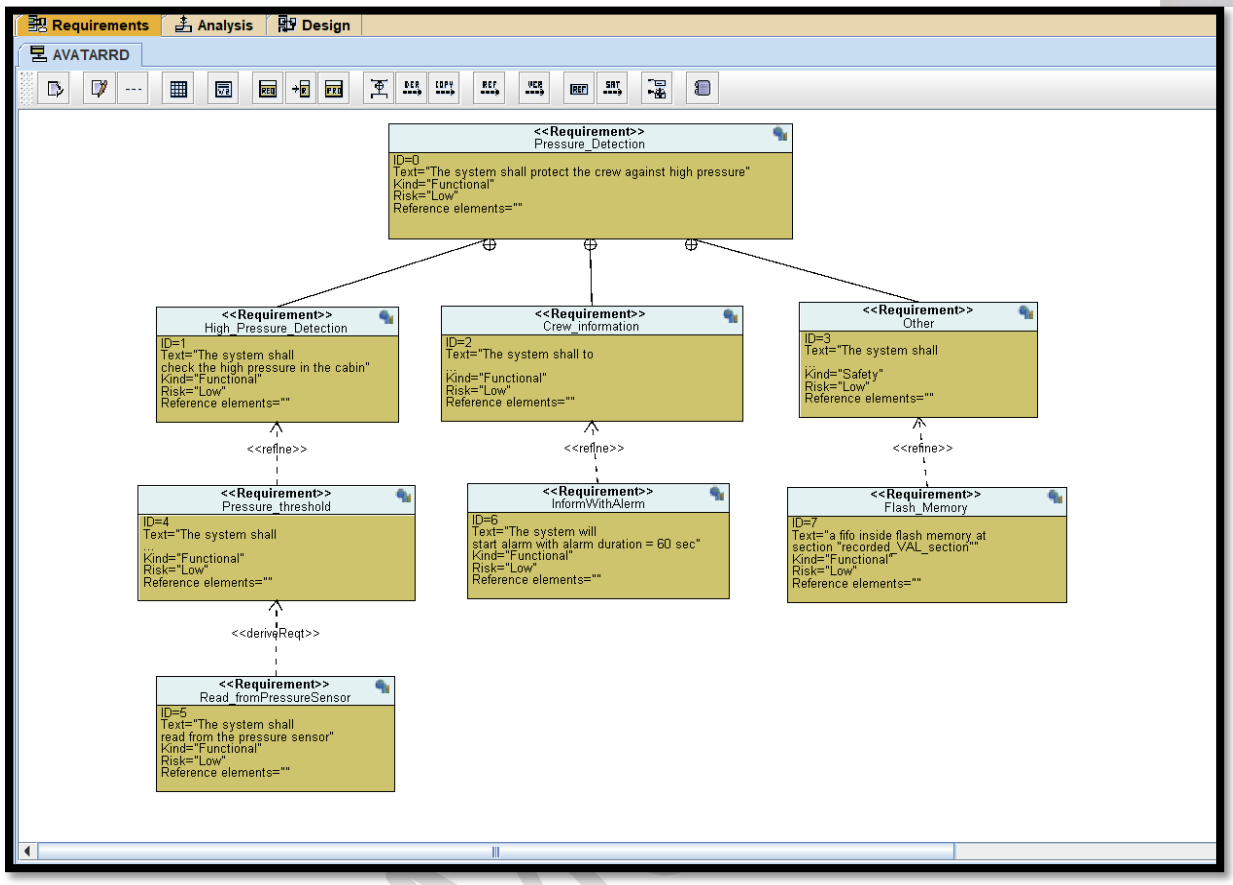
- void Set_Alarm_actuator(int i);

- void GPIO_INITIALIZATION();

#LEARN_IN DEPTH

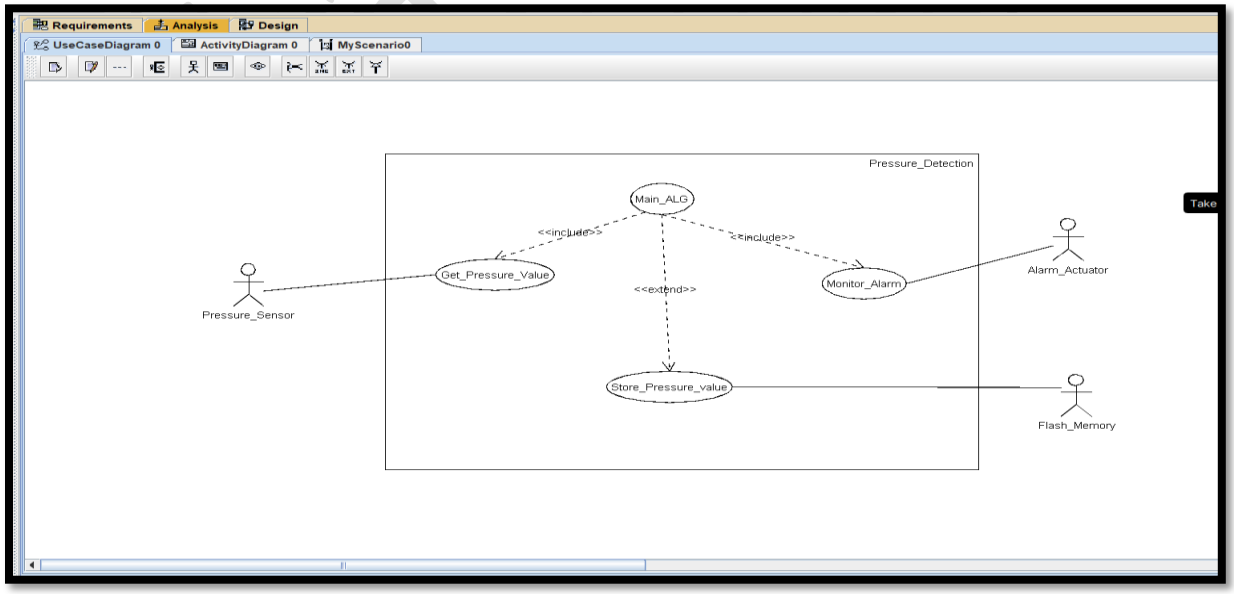
#Be_professional_in embedded_system

1- Requirement Diagram

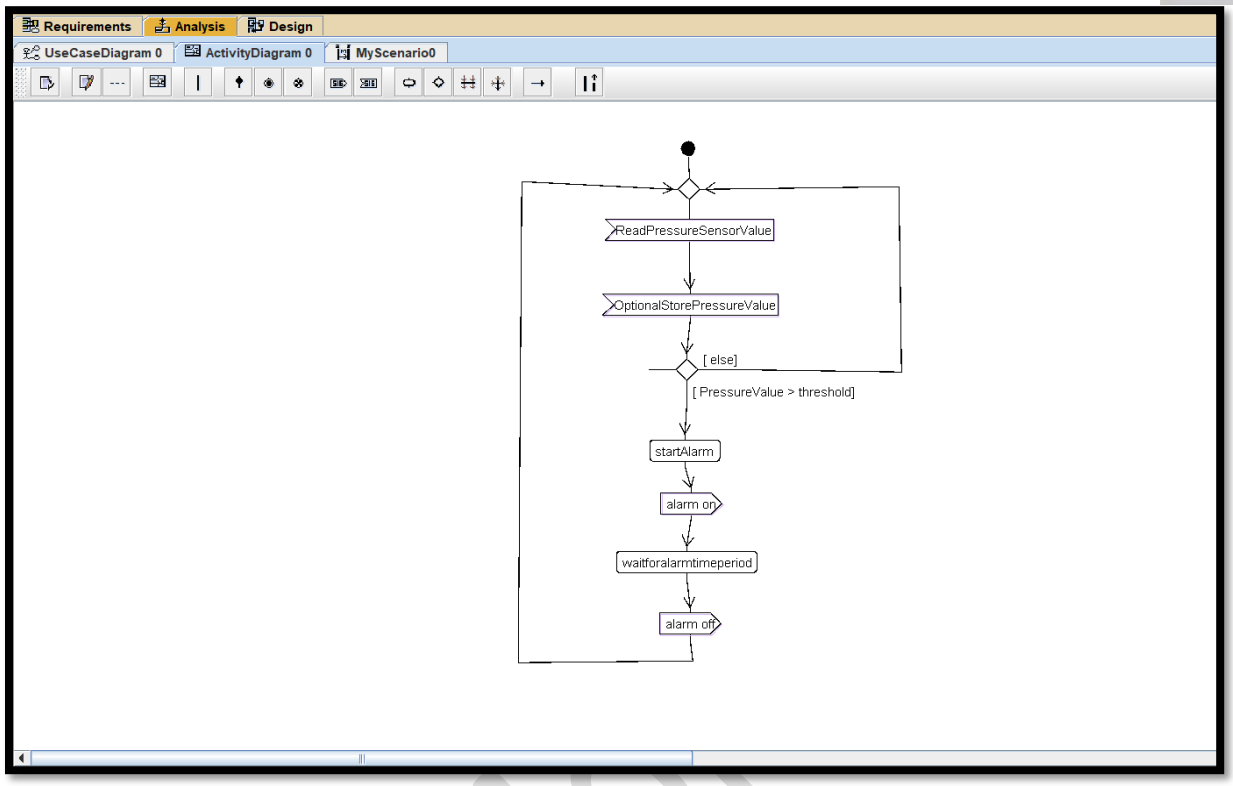


2- System Analysis

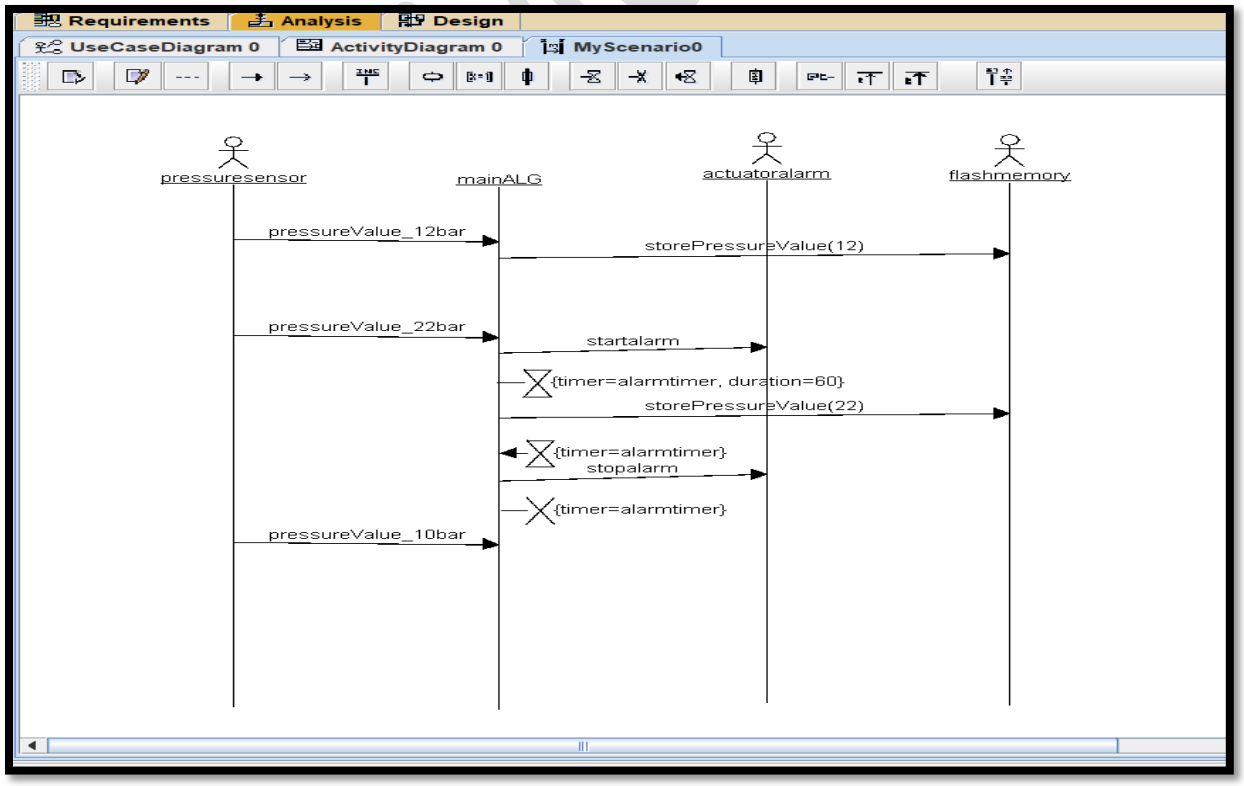
2.1 Use Case Diagram



2.2 Activity Diagram



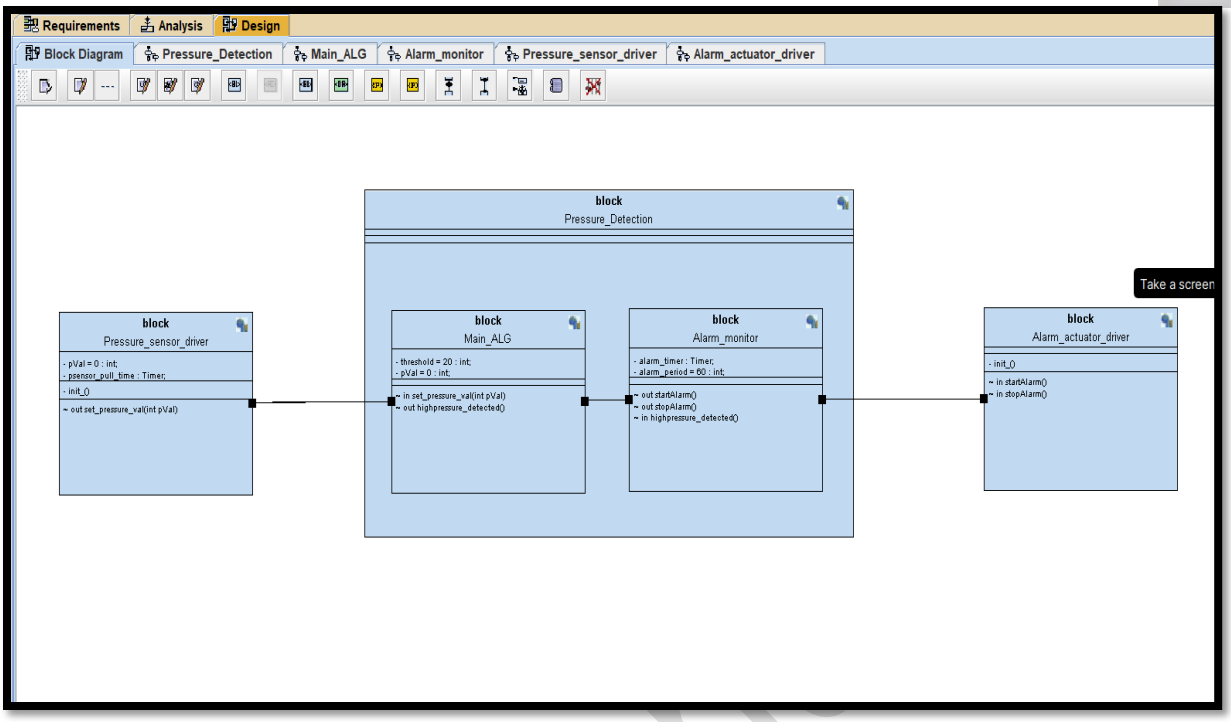
2.3 Sequence Diagram



#LEARN_IN DEPTH

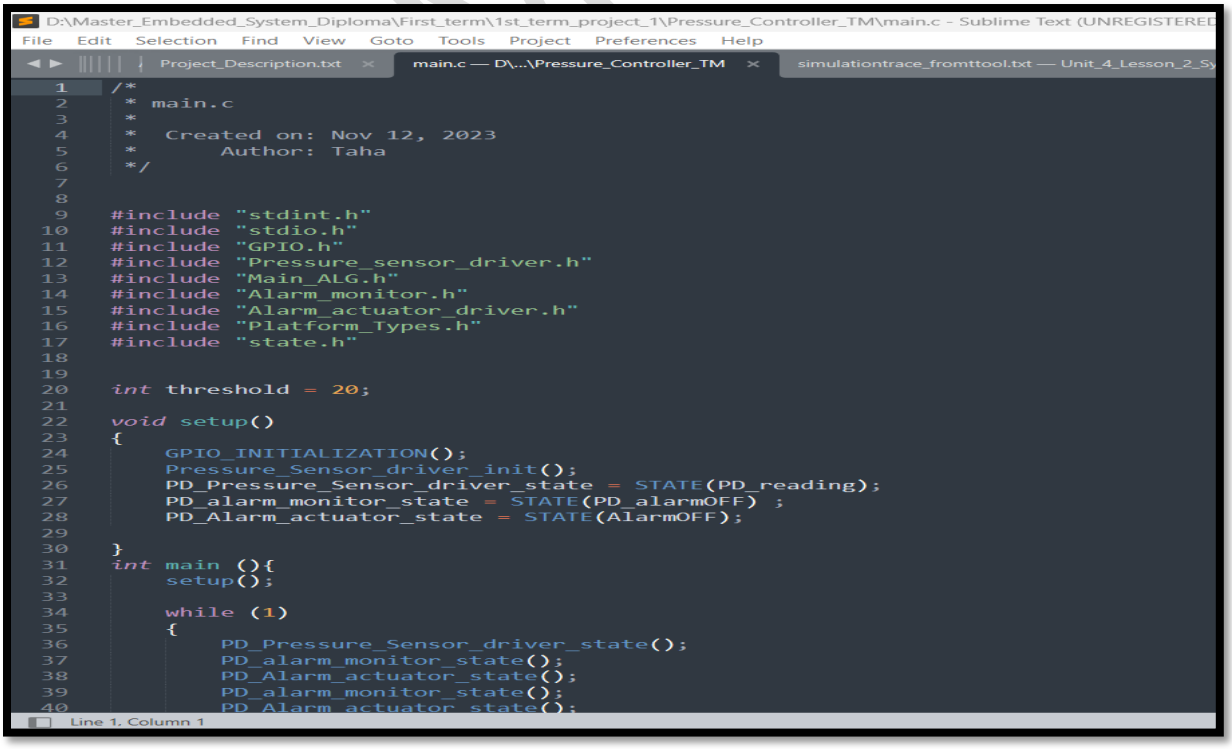
#Be_professional_in embedded_system

3- System Design (Modules with its own state machines)



4- Embedded Codes

4.1 Main.c



```
/*
 * main.c
 *
 *  Created on: Nov 12, 2023
 *      Author: Taha
 */


#include "stdint.h"
#include "stdio.h"
#include "GPIO.h"
#include "Pressure_sensor_driver.h"
#include "Main_ALG.h"
#include "Alarm_monitor.h"
#include "Alarm_actuator_driver.h"
#include "Platform_Types.h"
#include "state.h"


int threshold = 20;

void setup()
{
    GPIO_INITIALIZATION();
    Pressure_Sensor_driver_init();
    PD_Pressure_Sensor_driver_state = STATE(PD_reading);
    PD_alarm_monitor_state = STATE(PD_alarmOFF) ;
    PD_Alarm_actuator_state = STATE(AlarmOFF);

}
int main (){
    setup();

    while (1)
    {
        PD_Pressure_Sensor_driver_state();
        PD_alarm_monitor_state();
        PD_Alarm_actuator_state();
        PD_alarm_monitor_state();
        PD_Alarm_actuator_state();
```

4.2 MakeFile

Makefile | Makefile | Pressure_Controller_TM.axf | Pressure_Controller_TM.bin

```
#@copyright : Taha

CC=arm-none-eabi-
CFLAGS= -mcpu=cortex-m4 -mthumb -gdwarf-2 -g
INCS=-I . -std=c99
LIBS=
SRC= $(wildcard *.c)
OBJ= $(SRC:.c=.o)
As= $(wildcard *.s)
AsOBJ= $(As:.s=.o)
Project_name=Pressure_Controller_TM

all: $(Project_name).bin

    @echo "------Build is Done---------"


%.o:    %.c
    $(CC)gcc.exe -c $(CFLAGS) $(INCS) $< -o $@


$(Project_name).elf:  $(OBJ) $(AsOBJ)
    $(CC)ld.exe -T linker_script.ld $(LIBS)  $(OBJ) $(AsOBJ) -specs=nosys.specs -o $@ -Map=Map_file.map
    cp $(Project_name).elf $(Project_name).axf

$(Project_name).bin: $(Project_name).elf
    $(CC)objcopy.exe -O binary $< $@

clean-all:
    rm *.o *.elf *.bin

clean:
    rm *.elf *.bin
```

4.3 Startup.c file

main.c — D\ | Alarm_monitor.h | Alarm_monitor.c | Project_Description.txt | — D\...\Pressure_Controller_TM

```
//startup.c
//Taha Mohamed

#include <stdint.h>

void Reset_Handler ();
extern int main(void);

void Default_Handler()
{
    Reset_Handler();
}

void NMI_Handler () __attribute__((weak, alias ("Default_Handler")));
void H_fault_Handler () __attribute__((weak, alias ("Default_Handler")));

//booking 1024 B located by .bss through uninitialized array of int 256 element
static unsigned long _stack_top[256];

void (* const  g_p_fn_Vectors[])() __attribute__((section(".vectors"))) = //arra
{
    (void (*)()) ((unsigned long)_stack_top + sizeof(_stack_top)), // casting i
    &Reset_Handler,
    &NMI_Handler,
    &H_fault_Handler,

};

extern unsigned int _E_text ;
extern unsigned int _S_DATA ;
extern unsigned int _E_DATA ;
extern unsigned int _S_bss ;
extern unsigned int _E_bss ;
```

#LEARN_IN DEPTH

#Be_professional_in embedded_system

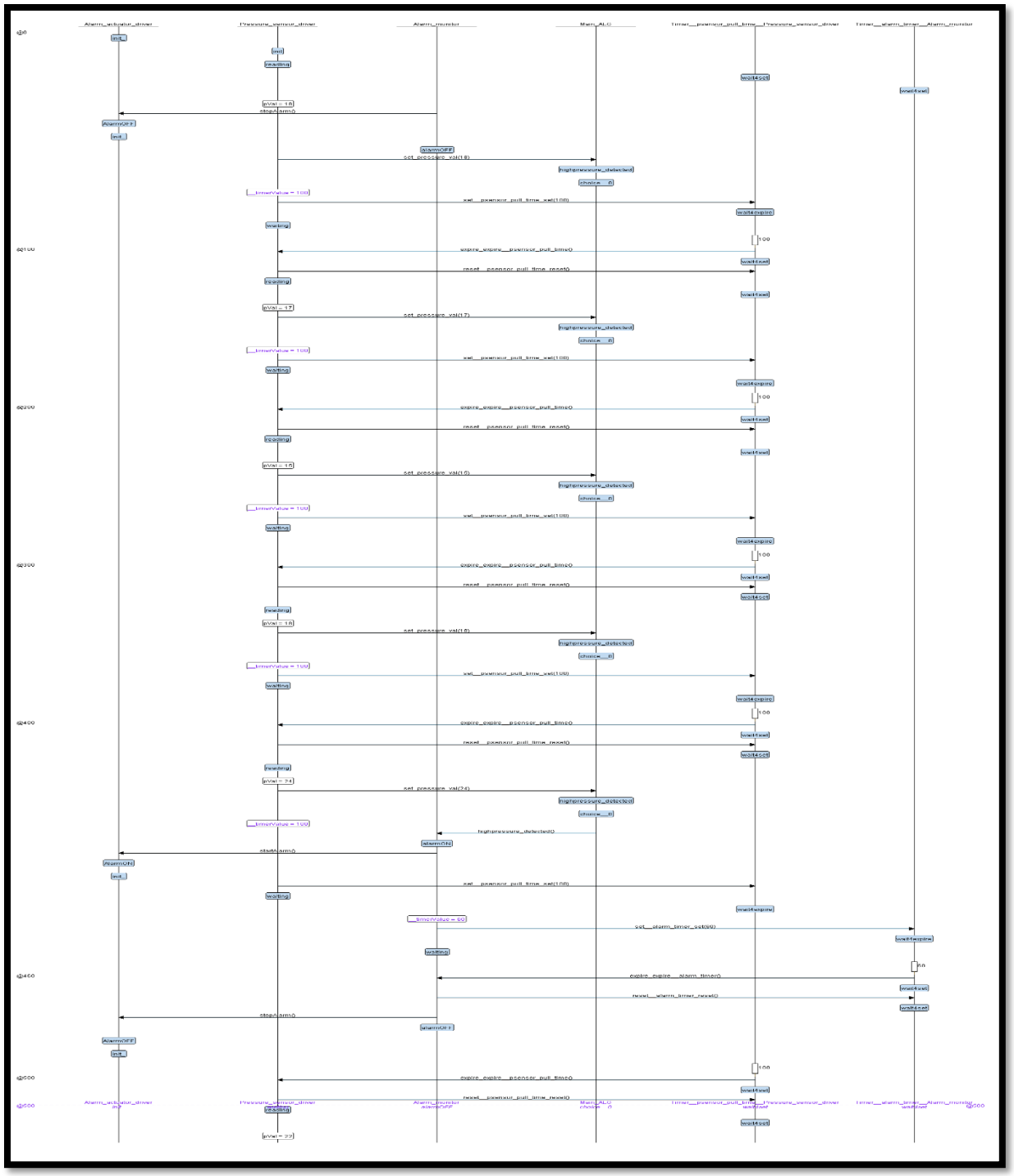
4.4 Linker Script.ld

```
/* linker_script CortexM3
Taha Mohamed
*/

MEMORY
{
flash(RX) : ORIGIN = 0x00000000, LENGTH = 128K
sram(RWX) : ORIGIN = 0x20000000, LENGTH = 20K
}

SECTIONS
{
    .text : {
            *(.vectors*)
            *(.text*)
            *(.rodata*)
            _E_text = . ;
    }> flash
    .data : {
            _S_DATA = . ;
            *(.data)
            _E_DATA = . ;
    }> sram AT> flash
    .bss  : {
            _S_bss = . ;
            *(.bss*)
            . = ALIGN(4);
            _E_bss = . ;
    }> sram
}
```

4.5 GPIO.c

```
/*
 * driver.c
 *
 *  Created on: Nov 12, 2023
 *      Author: Taha
 */


#include "GPIO.h"
#include "stdint.h"
#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;
```

5- Simulationtrace_fromttool



6- Simulationtrace_fromttool.csv

simulationtrace_fromttool.csv - Excel (Product Activation Failed)

File Home Insert Page Layout Formulas Data Review View Help PDFelement Tell me what you want to do

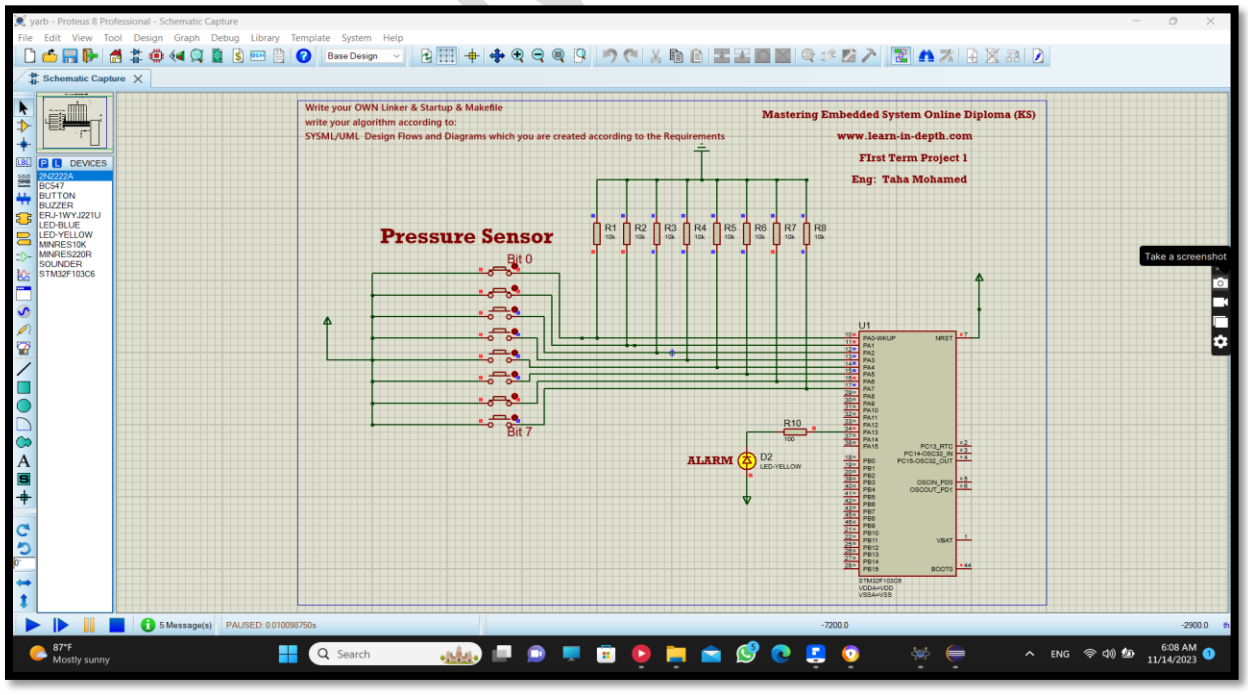
A1 ID

ID	block	elementID	element U	linked trar	initial cloc	final clock	duration	attributes	actions
0	Alarm_actuator_driver	225	b33b88e5	null	0	0	0	null	null
1	Pressure_sensor_driver	241	458b90ed	null	0	0	0	0 0	null
2	Alarm_monitor	264	af34155b-	null	0	0	0	60 0	null
3	Main_ALG	282	06a306e7	null	0	0	0	20 0	null
4	Timer__psensor_pull_time__Pressure_sensor_driver	306	bb641bd7	null	0	0	0	0	null
5	Timer__alarm_timer__Alarm_monitor	345	a31b73b3	null	0	0	0	0	null
6	Alarm_actuator_driver	226	d570296c	null	0	0	0	null	null
7	Alarm_actuator_driver	224	4aaa5e97	null	0	0	0	null	null
8	Pressure_sensor_driver	250	80a6b340	null	0	0	0	0 0	null
9	Pressure_sensor_driver	240	1bb7f85b-	null	0	0	0	0 0	null
10	Pressure_sensor_driver	249	4b0bcabe	null	0	0	0	0 0	null
11	Pressure_sensor_driver	239	eb769464	null	0	0	0	0 0	null
12	Alarm_monitor	272	9cf47914-	null	0	0	0	60 0	null
13	Main_ALG	294	77de7345	null	0	0	0	20 0	null
14	Timer__psensor_pull_time__Pressure_sensor_driver	314	bb641bd7	null	0	0	0	0	null
15	Timer__psensor_pull_time__Pressure_sensor_driver	307	bb641bd7	null	0	0	0	0	null
16	Timer__alarm_timer__Alarm_monitor	353	a31b73b3	null	0	0	0	0	null
17	Timer__alarm_timer__Alarm_monitor	346	a31b73b3	null	0	0	0	0	null
18	Pressure_sensor_driver	248	9f80b210-	null	0	0	0	0 0	null
19	Pressure_sensor_driver	238	e4c1c006-	null	0	0	0	18 0	action#0: pVal = 18
20	Pressure_sensor_driver	247	36595ffb-	null	0	0	0	18 0	null
21	Alarm_monitor	260	d3e255fd-	23	0	0	0	60 0	null
22	Alarm_actuator_driver	231	9d5b2d48	null	0	0	0	null	null
23	Alarm_actuator_driver	222	c4ce1e9e-	21	0	0	0	null	null
24	Alarm_actuator_driver	229	79e28b23	null	0	0	0	null	null
25	Alarm_actuator_driver	220	370d6f88-	null	0	0	0	null	null
26	Alarm_monitor	271	d02cad3a	null	0	0	0	60 0	null
27	Alarm_actuator_driver	227	ec8d86d0	null	0	0	0	null	null
28	Alarm_actuator_driver	224	4aaa5e97	null	0	0	0	null	null
29	Alarm_monitor	263	09a4e5a3	null	0	0	0	60 0	null
30	Pressure_sensor_driver	237	67d2c573	31	0	0	0	18 0	null
31	Main_ALG	281	5f84b859-	30	0	0	0	20 18	action#0: 18

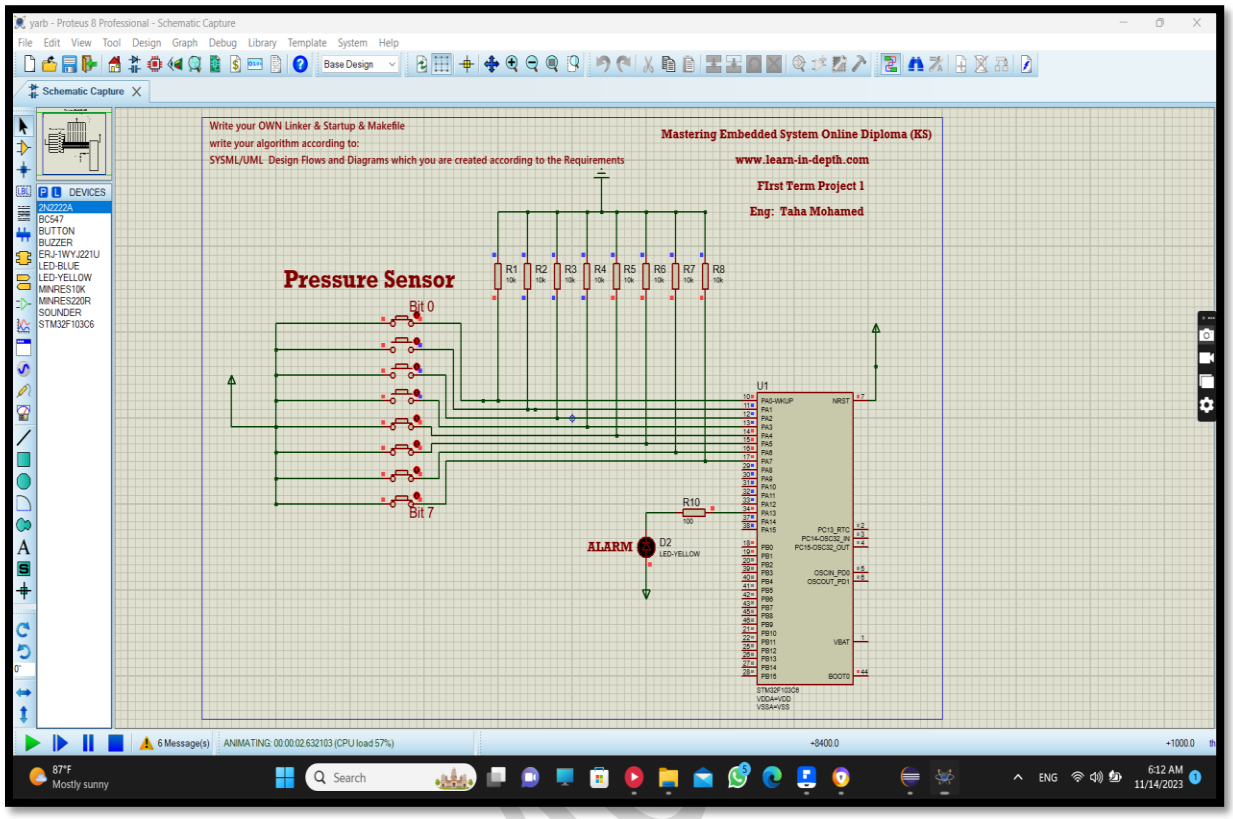
simulationtrace_fromttool

Ready Accessibility: Unavailable

7- Alarm ON(when pressure exceeds 20bar)



8- Alarm OFF



9- Interactive Simulation