

Learn IN Depth

Be Professional In Embedded System



Mastering Embedded System Online Diploma

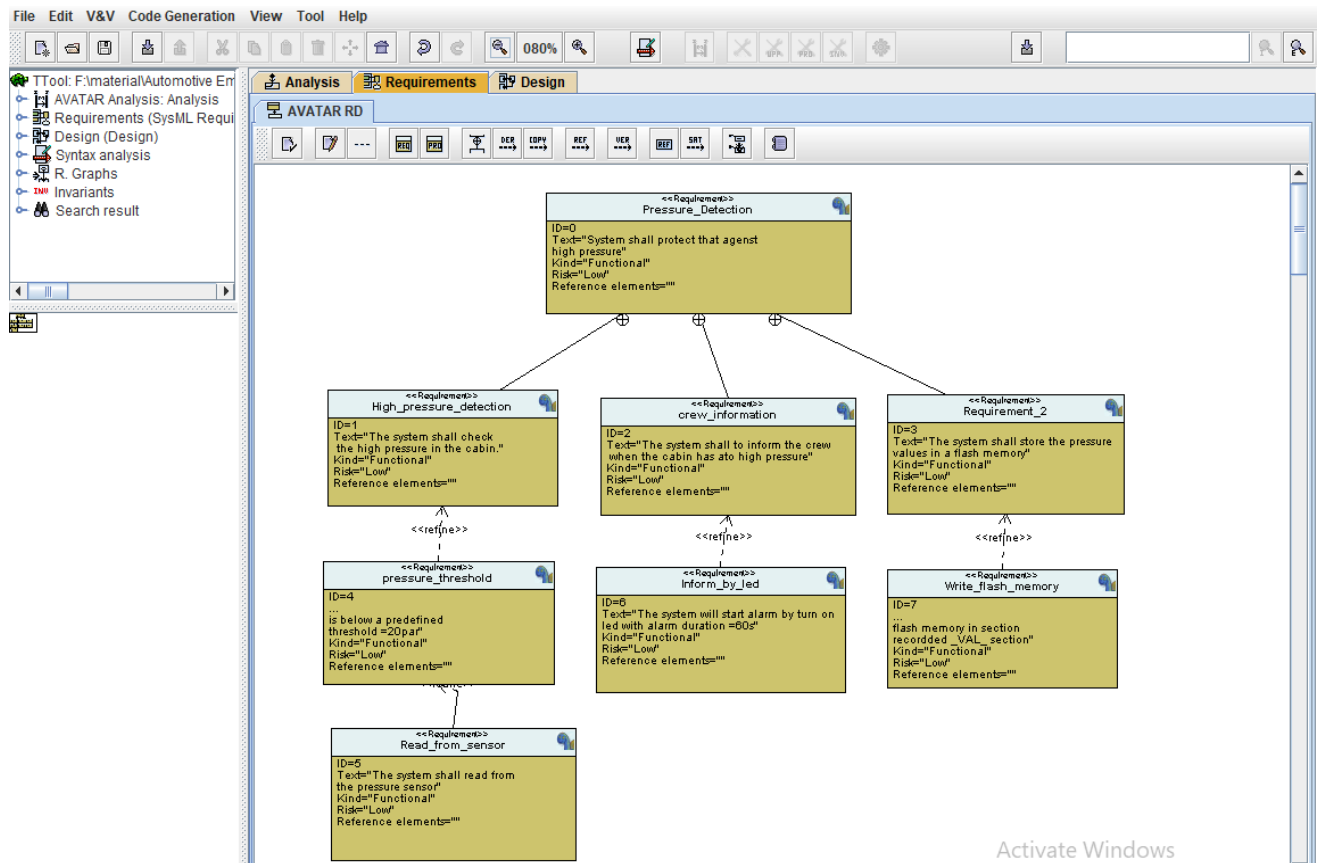
www.learn-in-depth.com

First Term (Final Project 1)

Eng: Hossam Magdy Afifi

My Profile: <https://www.learn-in-depth.com/online-diploma/hossammagdy308%40gmail.com>

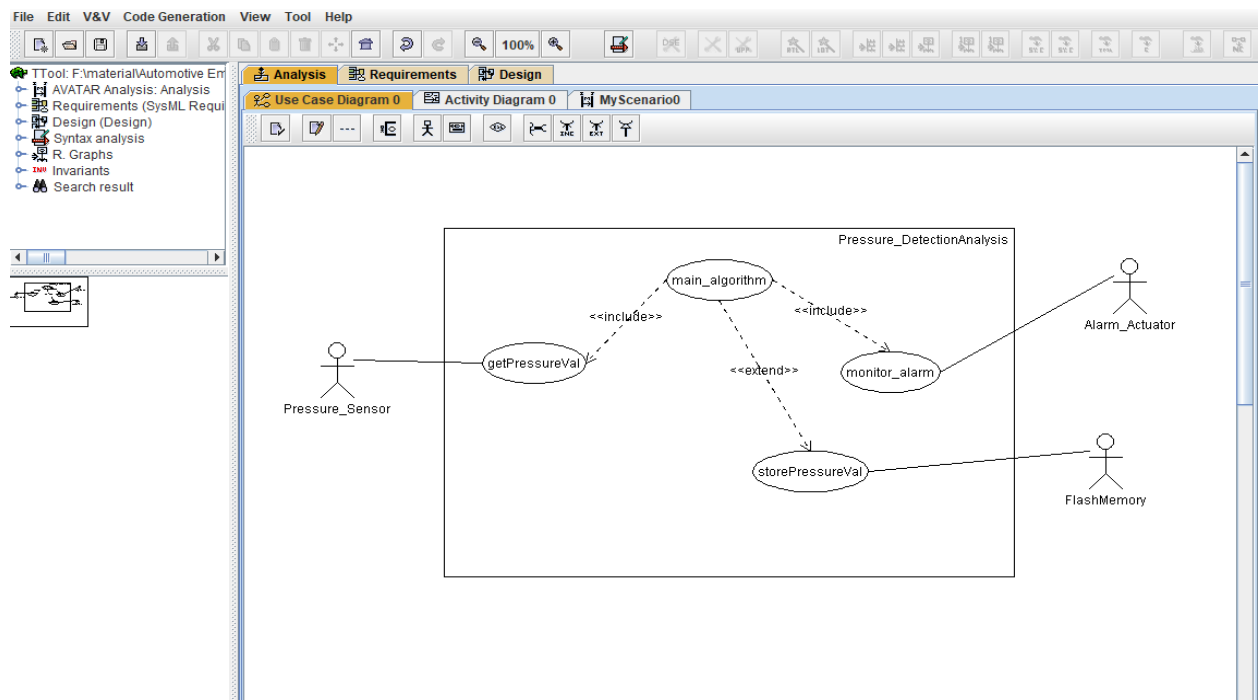
➤ Requirement Diagram:



➤ System Analysis:

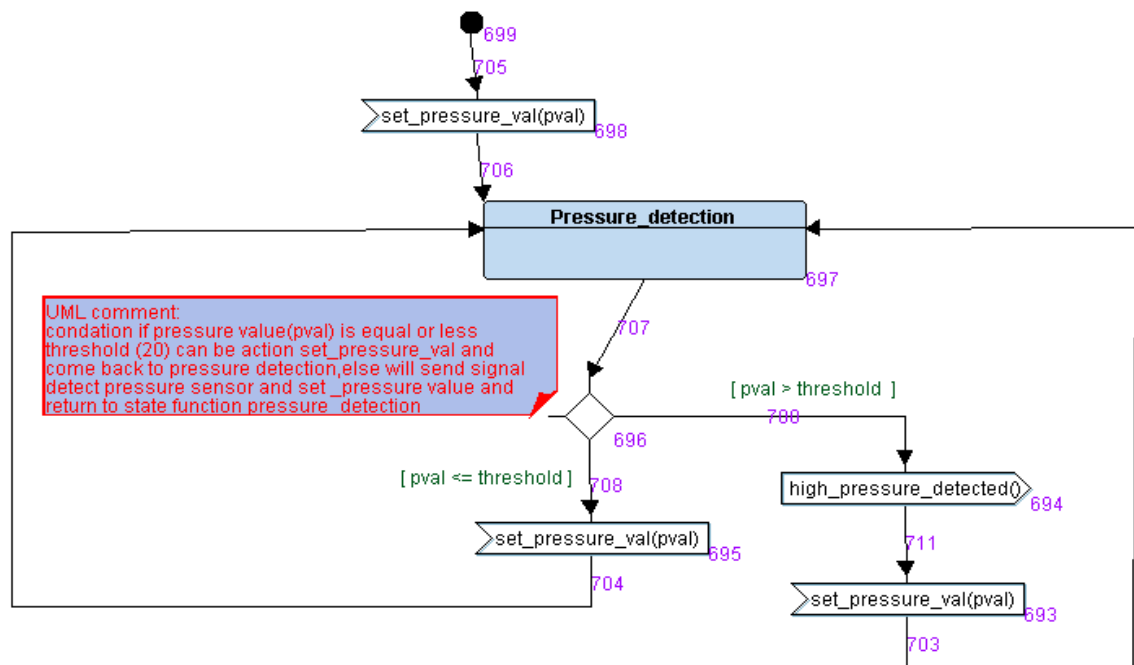
1. Use Case Diagram
2. Activity Diagram
3. Sequence Diagram

1) Use Case Diagram

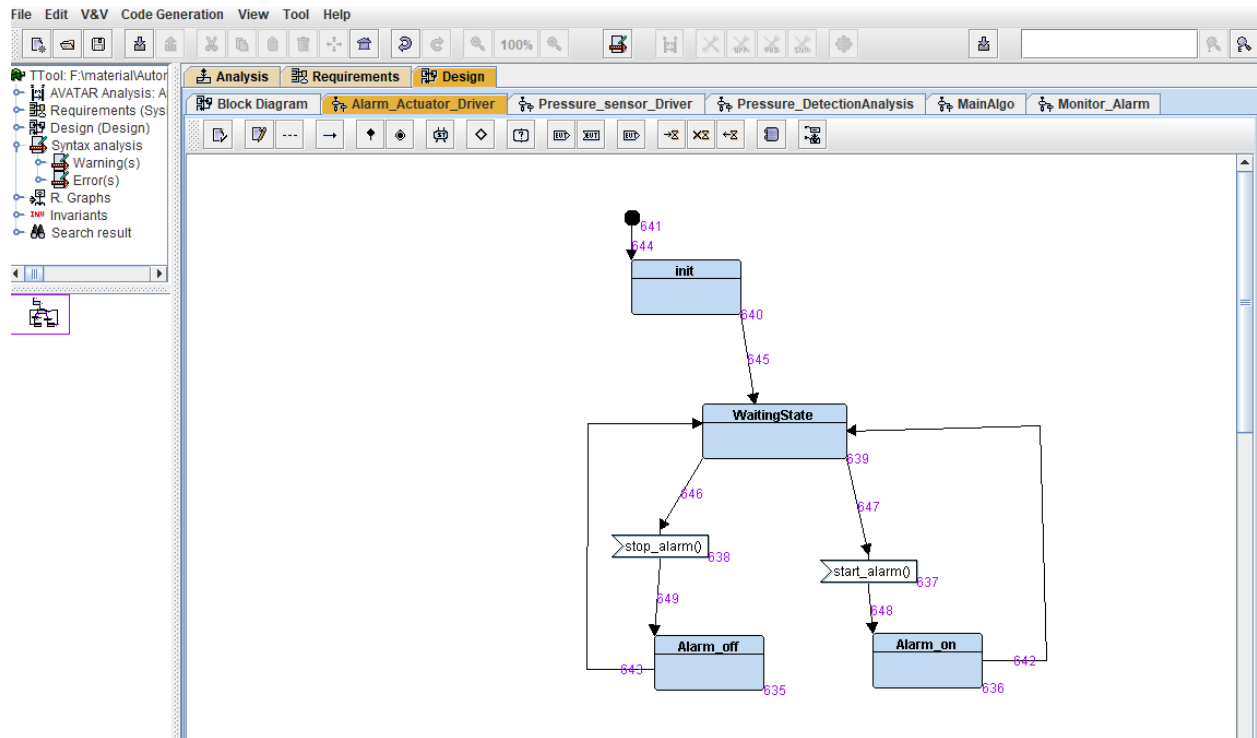


2) Activity Diagram

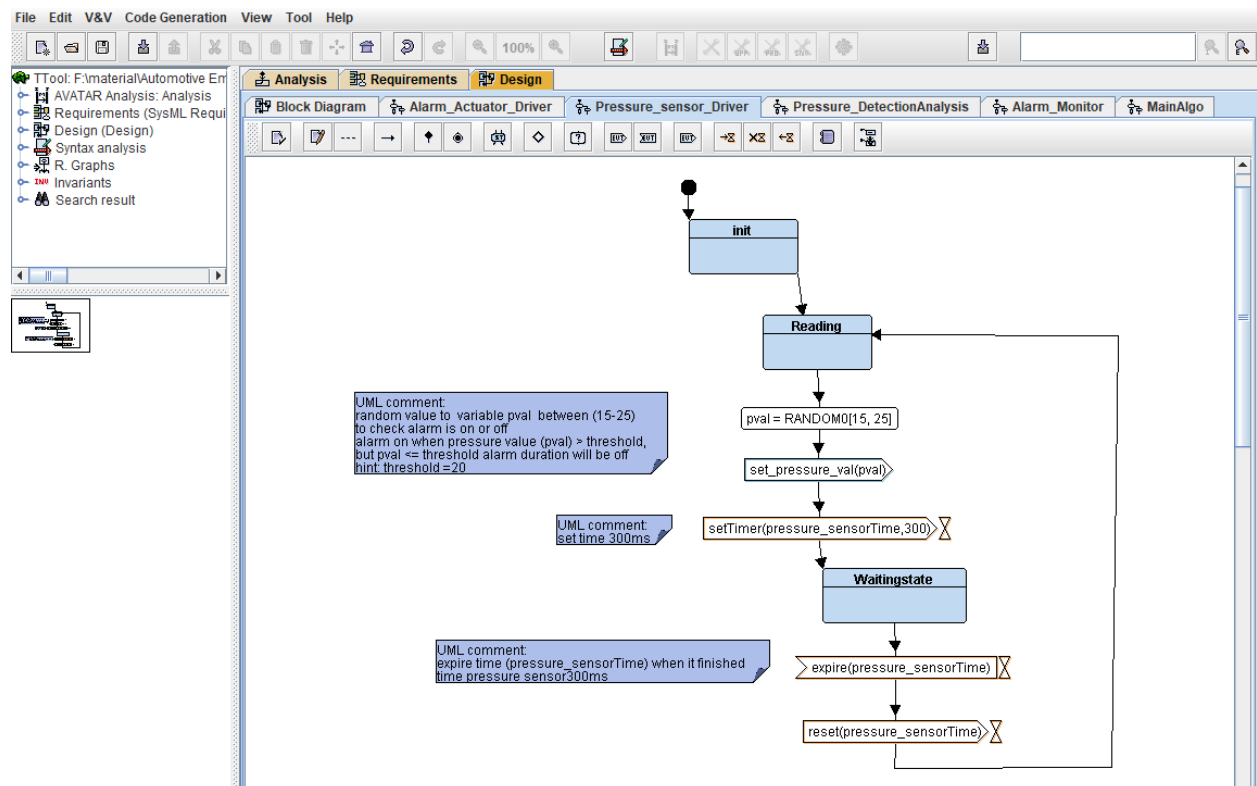
1. Main Algorithm :



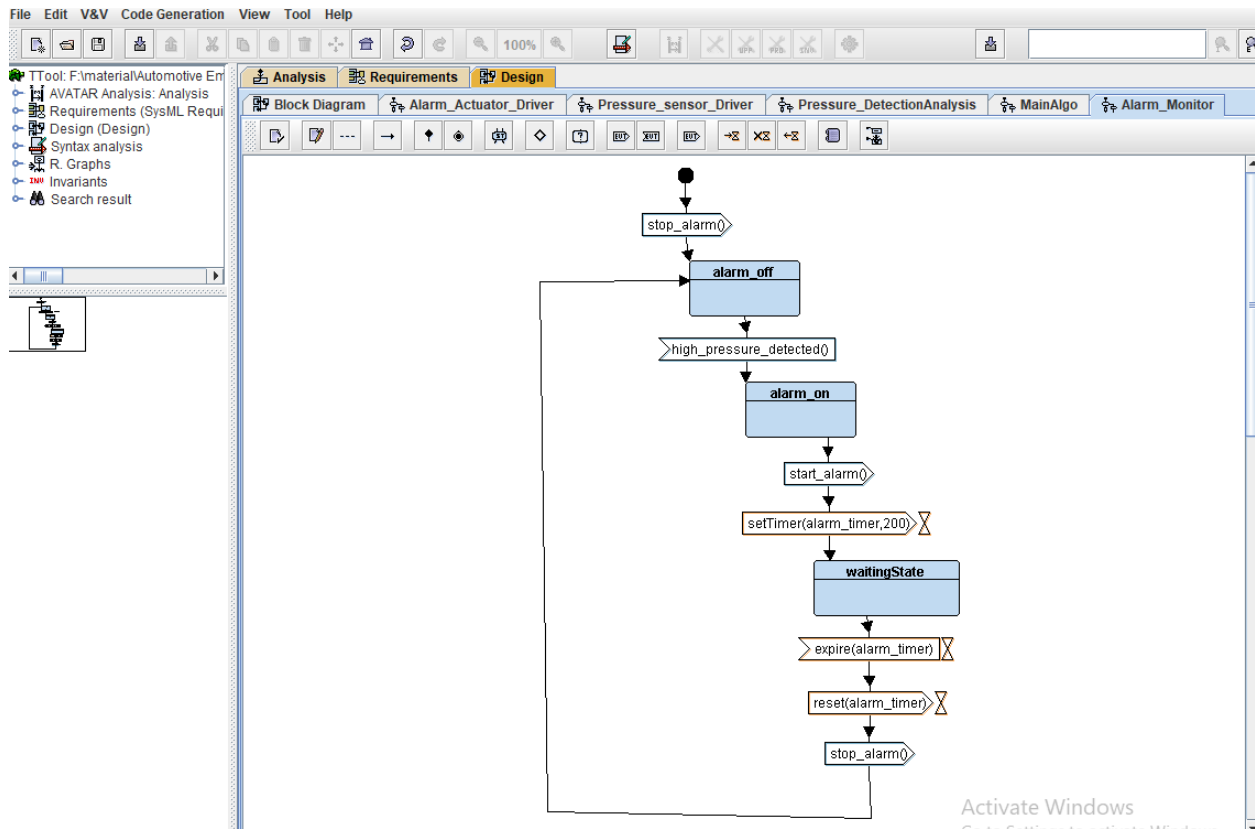
2. Alarm Actuator Driver:



3. Pressure Sensor Driver:

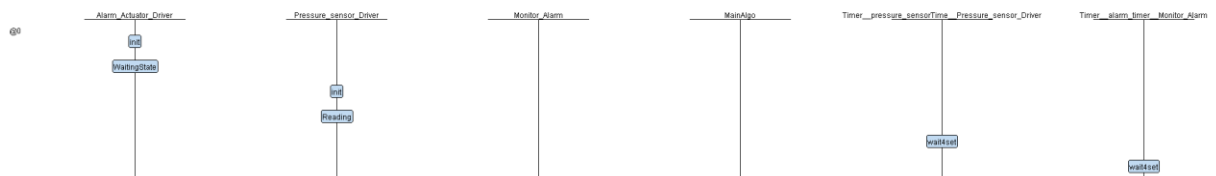


4. Alarm Monitor:



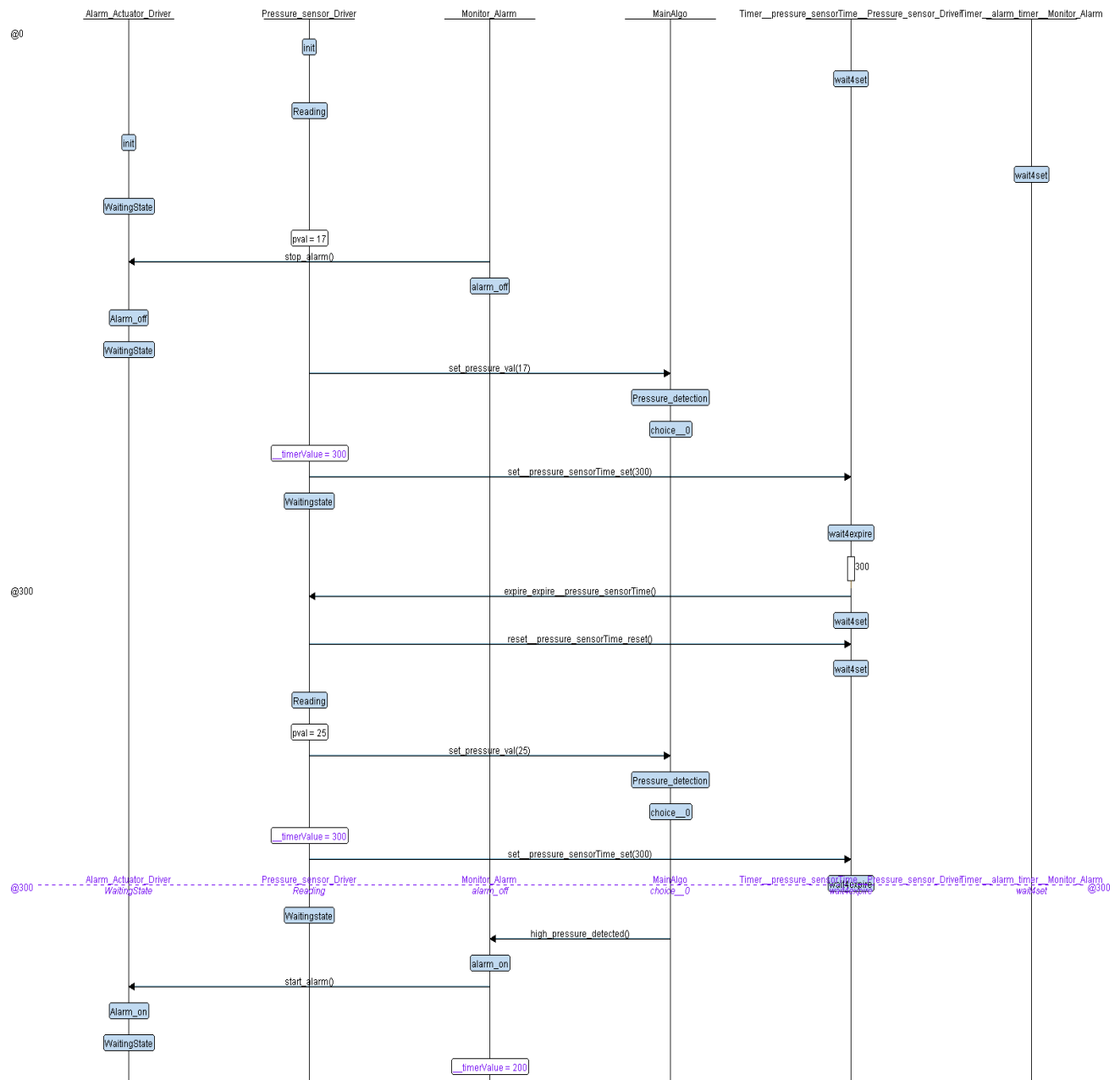
3) Sequence Diagram :

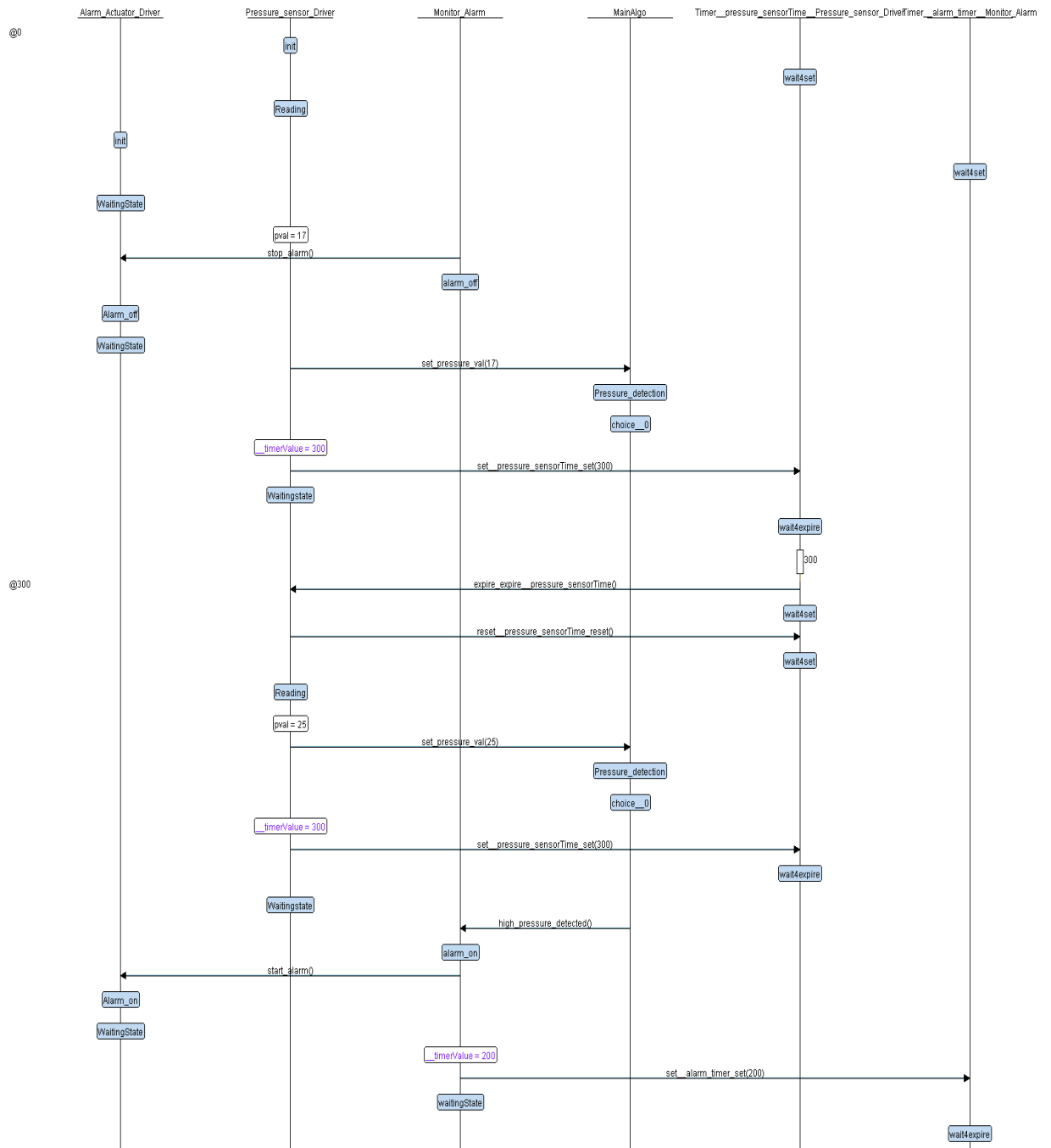
In the First, Pressure sensor reading value and alarm monitor was stated in stop alarm so alarm actuator is turn off.



When pressure sensor reads less than threshold value, alarm actuator turned off and monitor alarm stop.

Next step, pressure sensor driver reads 25bar man algorithm check this value is less or more than threshold value, if pval is more than threshold start alarm monitor and alarm actuator turn on, alarm monitor turn on for delay time 300sec and turn off ,otherwise alarm monitor and alarm actuator stopped.





The screenshot displays the TTool software interface, which is used for the development and analysis of software models. The interface is divided into several panes:

- Top Pane:** Contains the menu bar (File, Edit, V&V, Code Generation, View, Tool, Help) and a toolbar with various icons for file operations, editing, and navigation. The status bar at the bottom indicates the current zoom level is 051%.
- Left Pane:** Shows the project tree on the left, listing the current project structure:
 - TTTool: F:\material\Avatar
 - AVATAR Analysis: A
 - Design (Design)
 - Syntax analysis
 - Warning(s)
 - Error(s)
 - R. Graphs
 - Invariants
 - Search result
- Top Bar:** Contains the main tabs for the current project:
 - Analysis
 - Requirements
 - Design
- Sub-Tab Bar:** Below the main tabs, there are sub-tabs for the current view:
 - Block Diagram
 - Alarm_Actuator_Driver
 - Pressure_sensor_Driver
 - Pressure_DetectionAnalysis
 - MainAlgo
 - Monitor_Alarm
- Diagram Area:** The central workspace displays a UML Block Diagram. The diagram shows the following components and their relationships:
 - Pressure_sensor_Driver**: A component with a provided interface (providedIn) and a required interface (requires). It is connected to the **Pressure_Detection** component.
 - Alarm_Actuator_Driver**: A component with a provided interface (providedIn) and a required interface (requires). It is connected to the **Pressure_Detection** component.
 - Pressure_Detection**: A component that provides the **Pressure_sensor_Driver** and requires the **Alarm_Actuator_Driver**.
 - Pressure_DetectionAnalysis**: A component that provides the **MainAlgo** and requires the **Monitor_Alarm**.
 - MainAlgo**: A component that provides the **Pressure_DetectionAnalysis** and requires the **Pressure_sensor_Driver**.
 - Monitor_Alarm**: A component that provides the **Pressure_DetectionAnalysis** and requires the **Alarm_Actuator_Driver**.

The screenshot displays the Proteus software interface with a schematic diagram titled "Pressure Controller HM". The diagram shows a pressure sensor connected to an 8-bit ADC (R1-R8) and a microcontroller (U1). The microcontroller is configured with various pins and components, including an LED alarm (D2) and a buzzer (B1). The status bar at the bottom indicates "Pressure = 128 > 20" and "Alarm will be enabled for 1000s".

Pressure Controller HM

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: Hossam Magdy Afifi

Pressure Sensor

Bit 0
Bit 7

Pressure = 128 > 20

Alarm will be enabled for 1000s

U1
PA0-WKUP
PA1
PA2
PA3
PA4
PA5
PA6
PA7
PA8
PA9
PA10
PA11
PA12
PA13
PA14
PA15
PC13_RTC
PC14-OSC32_K
PC15-OSC32_OUT
OSCIN_PDO
OSCOUT_PDO
VREF
BOOT0
STM32F103C6

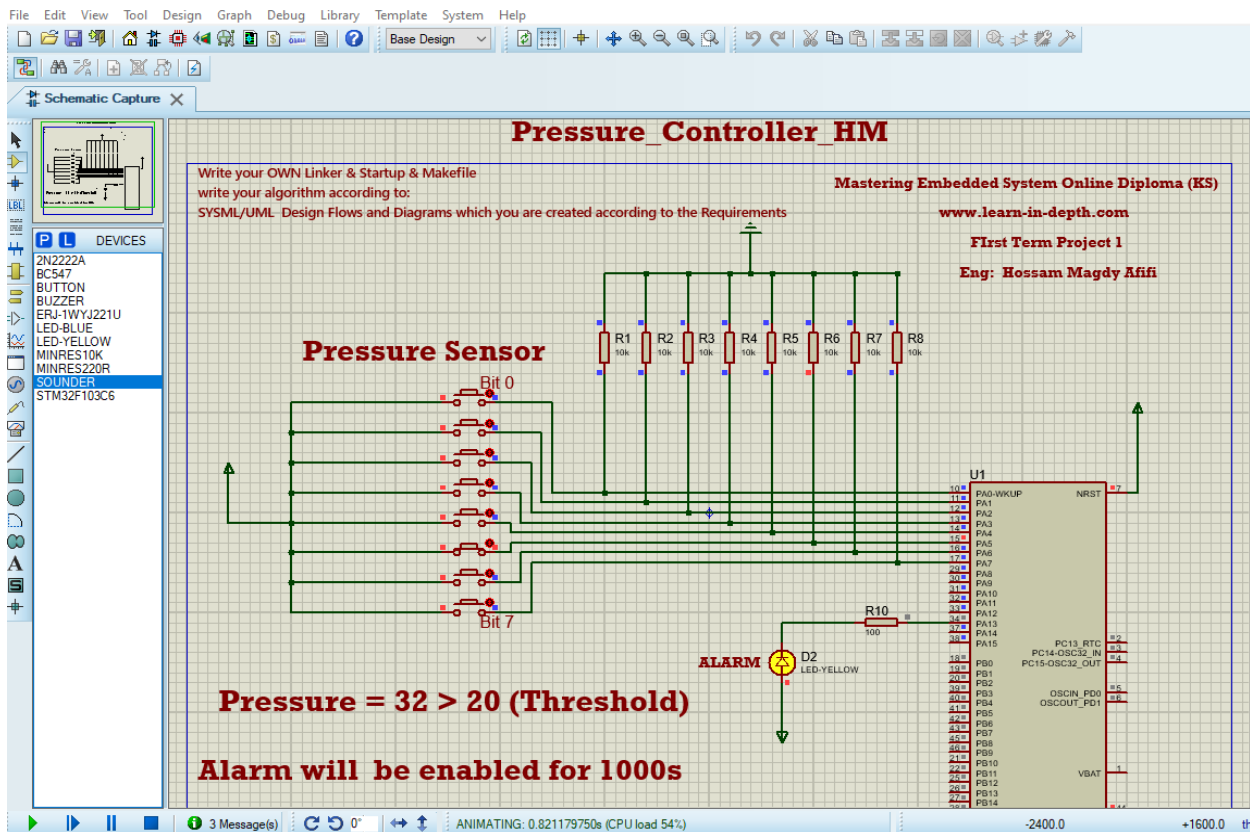
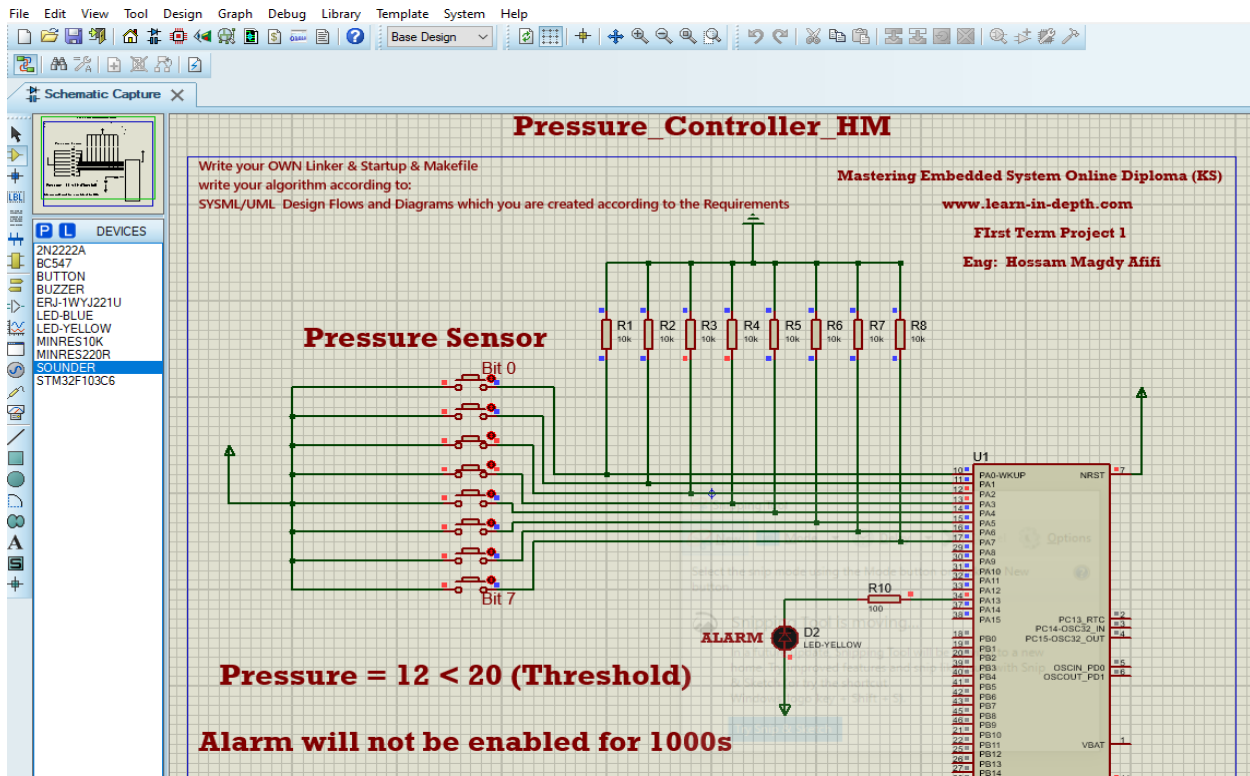
R10
100

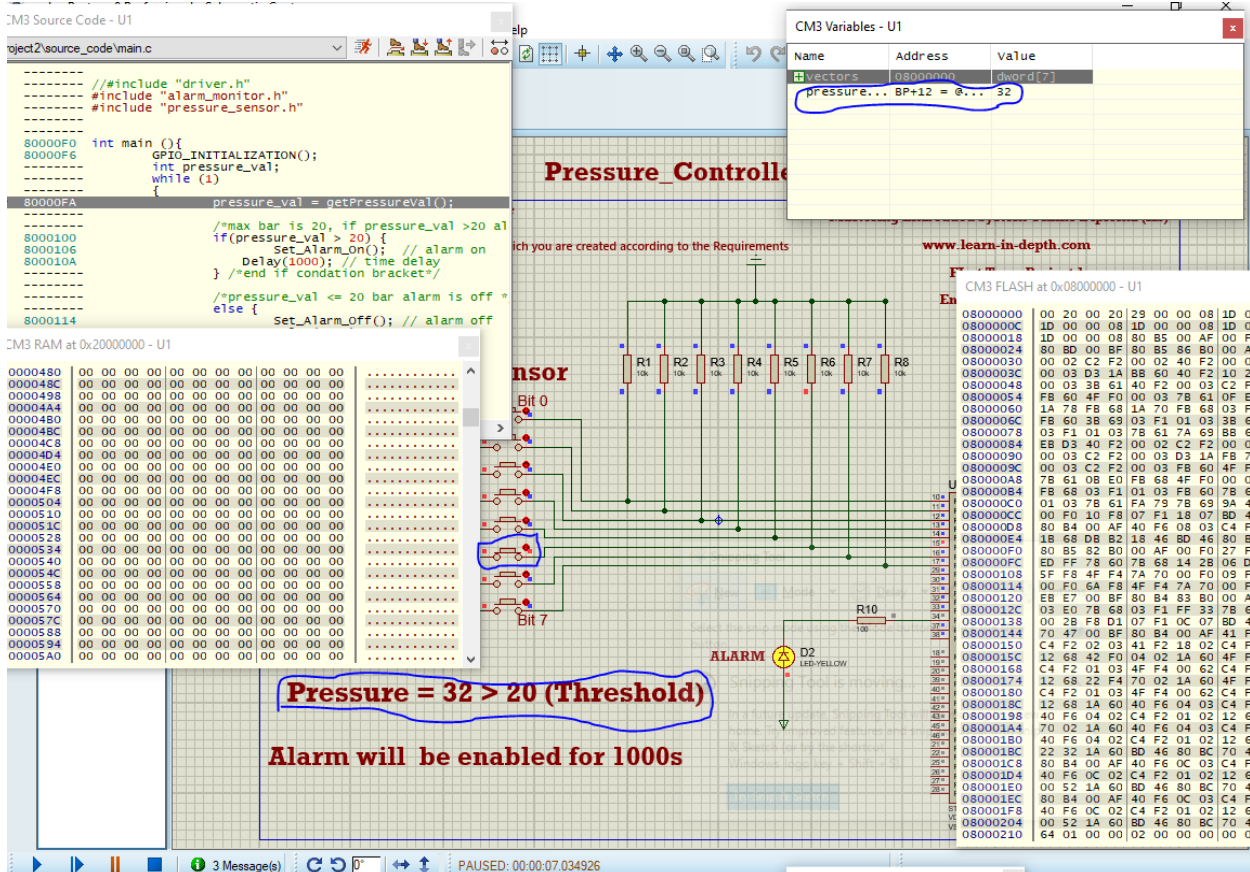
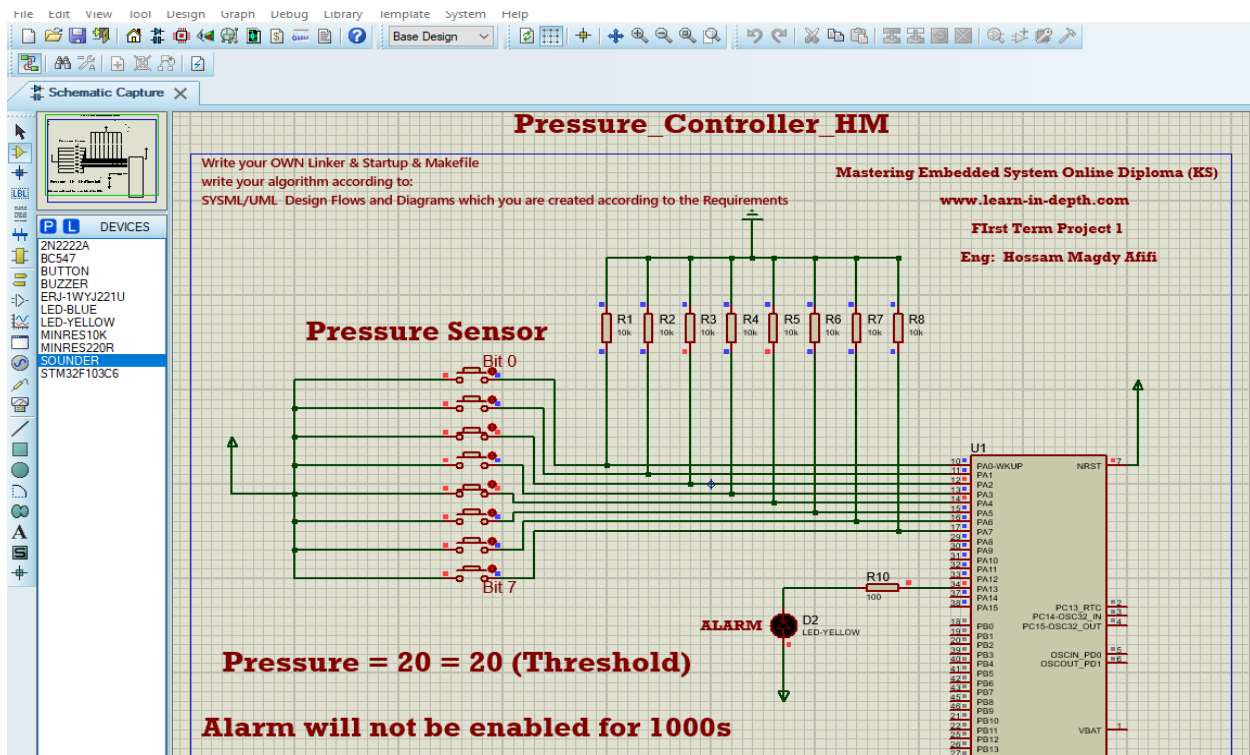
D2
LED-YELLOW


3 Message(s)

ANIMATING: 00:00:08.817107 (CPU load 50%)

-300.0 -600.0







```

project2\source_code\pressure_sensor.c

-----
#include "driver.h"
#include "pressure_sensor.h"
-----

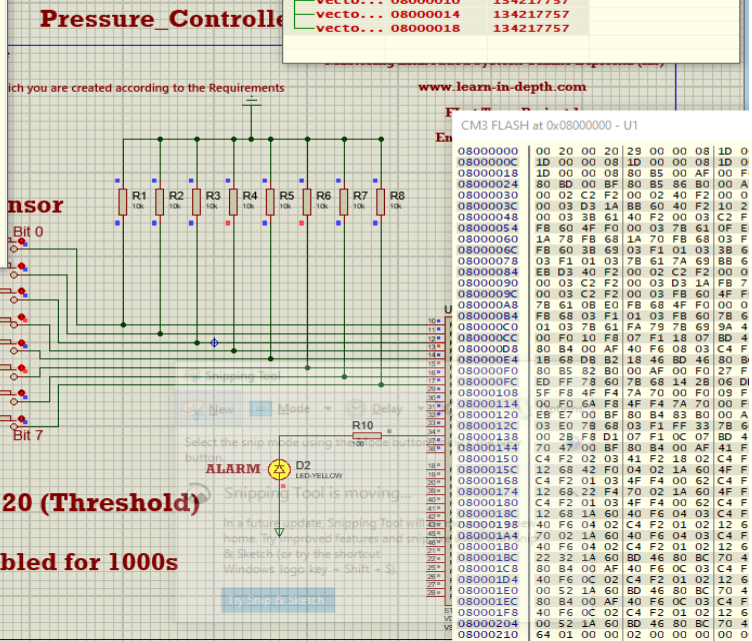
8000000 int getPressureVal() {
800000C     return (GPIOA_IDR & 0xFF);
8000008 }

```

Name	Address	Value
vectors	08000000	dword[7]
vecto...	08000000	536879104
vecto...	08000004	134217769
vecto...	08000008	134217757
vecto...	0800000C	134217757
vecto...	08000010	134217757
vecto...	08000014	134217757
vecto...	08000018	134217757

[illegible]

Alarm will be enabled for 1000s



```

project2_source\main.c
-----
// #include "driver.h"
// #include "alarm_monitor.h"
// #include "pressure_sensor.h"

80000F0 int main() {
80000F6     GPIO_INITIALIZATION();
           int pressure_val;
           while (1)
           {
               pressure_val = getPressureVal();

               /*max bar is 20; if pressure_val >20 al
8000106 if(pressure_val > 20) { // alarm on
8000108     Set_Alarm_On(); // alarm on
800010A     Delay(1000); // time delay
           } /*end if condation bracket*/

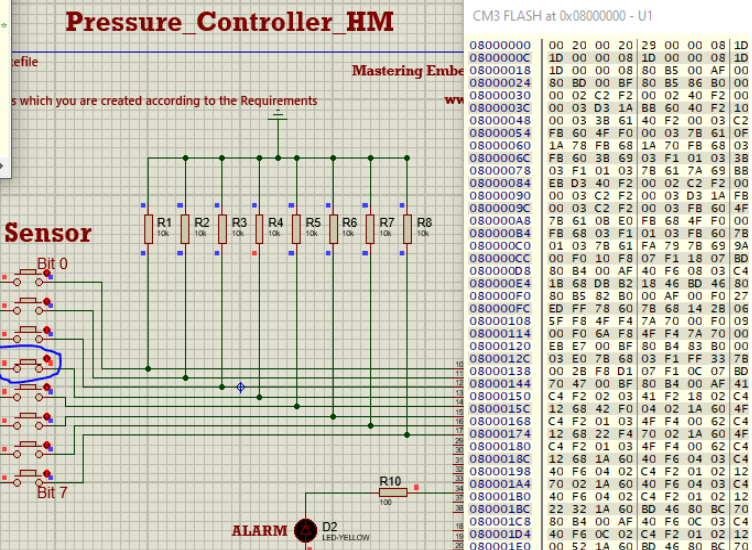
           /*pressure_val <= 20 bar alarm is off =
8000114 else {
           Set_Alarm_Off(); // alarm off
8000118     Delay(1000);
           } /*end else bracket*/

           //Implement your Design

8000112     }
           return 0;
}

```

Name	Address	Value
vectors	00000000	dword[7]
pressu...	BP+12	= ... 8

[illegible]


```

----- //include "driver.h"
----- #include "alarm_monitor.h"
----- #include "pressure_sensor.h"

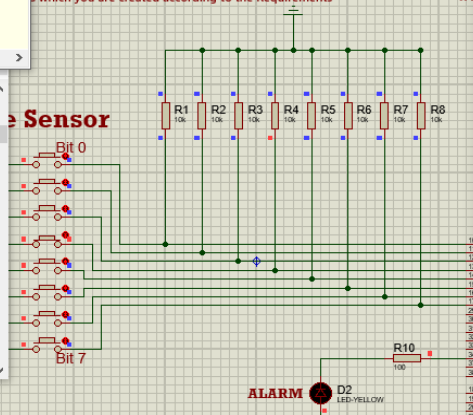
80000F0 int main () {
80000F6     GPIO_INITIALIZATION();
-----     int pressure_val;
-----     while (1)
80000FA     {
-----         pressure_val = getPressureVal();
-----         //max bar is 20, if pressure_val >20 al
8000106         if (pressure_val > 20)
-----             Set_Alarm_On(); // alarm on
8000114             Delay(1000); // time delay
-----             //end if condation bracket*/
-----
-----         //pressure_val <= 20 bar alarm is off =
8000114         else {
-----             Set_Alarm_Off(); // alarm off
8000118             Delay(1000);
-----             //end else bracket*/
-----
-----         //Implement your Design
8000112     }
----- }
----- return 0;

```

[illegible]

Pressure_Controller_HM

s which you are created according to the Requirements



CM3 FLASH at 0x08000000 - U1

08000000	00	20	00	20	29	00	00	08	1D	0
0800000C	1D	00	00	08	1D	00	00	08	1D	0
08000018	1D	00	00	08	80	85	00	AF	00	F
0800001C	00	00	00	00	00	00	00	00	00	0
08000030	00	02	C2	F2	00	02	40	F2	00	0
0800003C	00	03	D3	1A	8B	60	40	F2	10	2
08000048	00	03	3B	61	40	00	00	00	00	0
08000054	00	FB	68	68	03	F0	01	03	3B	F
08000060	1A	78	F8	68	1A	78	F8	68	03	F
0800006C	FB	68	68	68	03	F0	01	03	3B	F
08000078	03	00	00	00	00	00	00	00	00	0
08000084	EB	D3	40	F2	00	02	C2	F2	00	0
08000090	00	03	C2	F2	00	03	D3	1A	FB	7
0800009C	00	03	C2	F2	00	03	FB	68	40	F
080000A8	7B	F8	68	03	F0	01	03	3B	61	4
080000B4	FB	68	03	F1	01	03	FB	68	7B	F
080000C0	01	03	78	61	FA	78	78	69	9A	4
080000CC	00	00	00	00	00	00	00	00	00	0
080000D8	80	B4	00	AF	40	F6	08	03	C4	F
080000E4	1B	68	DB	B2	18	4B	8D	46	80	B
080000F0	80	85	82	80	00	AF	00	00	F0	F
080000FC	00	00	00	00	00	00	00	00	00	0
08000108	5F	FA	48	F4	7A	70	00	00	F0	F
08000114	00	00	6A	F8	4F	7A	7A	70	00	F
08000120	EB	F8	00	84	B4	83	80	00	00	0
0800012C	03	00	78	D8	68	00	00	00	00	0
08000138	00	28	F8	00	F7	F1	0C	07	8D	4
08000144	70	47	00	8F	B0	04	02	AF	41	F
08000150	00	00	00	00	00	00	00	00	00	0
0800015C	12	68	42	00	84	02	1A	60	4F	F
08000168	4	F2	01	03	4F	F4	00	62	64	F
08000174	12	68	22	4A	70	02	1A	60	4F	F
08000180	00	00	00	00	00	00	00	00	00	0
0800018C	12	68	1A	02	40	F6	04	03	C4	F
08000198	40	F6	04	02	42	F2	01	02	12	6
080001A4	00	00	00	00	00	00	00	00	00	0
080001B0	40	F6	04	02	42	F2	01	02	12	6
080001BC	22	32	1A	00	84	46	80	BC	70	F
080001C8	80	B4	00	AF	40	F6	0C	03	C4	F
080001D4	00	00	00	00	00	00	00	00	00	0
080001E0	00	52	1A	00	84	46	80	BC	70	F

```

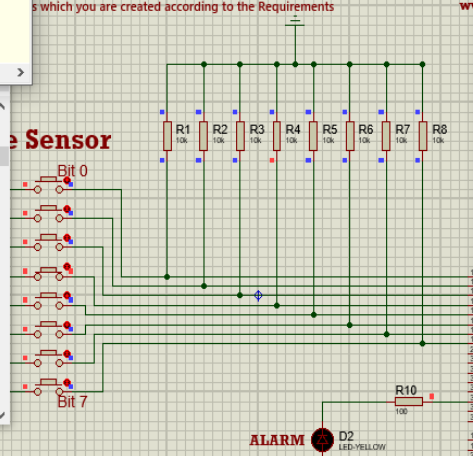
----- //include "driver.h"
----- #include "alarm_monitor.h"
----- #include "pressure_sensor.h"
-----
80000F0  int main () {
80000F6      GPIO_INITIALIZATION();
-----      int pressure_val;
-----      while (1)
80000FA          {
-----              pressure_val = getPressureVal();
-----
-----              /*max bar is 20, if pressure_val >20 al
8000100              if (pressure_val > 20) {
8000106                  SetAlarm_ON; // alarm on
800010A                  delay(1000); // time delay
-----              } /*end if condation bracket*/
-----
-----              /*pressure_val <= 20 bar alarm is off
8000114              else {
-----                  SetAlarm_Off(); // alarm off
8000118                  Delay(1000);
-----              } /*end else bracket*/
-----
-----              //Implement your Design
8000112          }
-----      return 0;
-----

```

[illegible]

Pressure Controller HM

tefile



CM3 FLASH at 0x08000000 - U1

08000000	00	20	00	20	29	00	00	08	1D
0800000C	1D	00	00	08	1D	00	00	08	1D
08000018	1D	00	00	08	85	05	00	AF	00
08000024	80	8D	00	8F	80	85	86	80	00
08000030	00	02	C2	F2	00	02	40	F2	00
0800003C	00	03	D3	1A	86	60	40	F2	10
08000048	00	03	03	00	00	00	00	00	00
08000054	F8	60	40	AF	00	03	78	61	0F
08000060	1A	78	F8	68	1A	70	F8	68	0F
0800006C	F8	60	38	69	03	F1	01	03	88
08000078	03	F1	01	03	78	61	7A	69	BB
08000084	E8	60	40	F2	00	02	C2	F2	00
08000090	00	00	00	00	00	00	00	00	00
0800009C	00	03	C2	F2	00	03	F8	60	4F
080000A8	78	61	08	0F	F8	68	4F	F0	00
080000B4	F8	68	03	F1	01	03	F8	60	78
080000C0	01	03	78	61	FA	79	78	69	9A
080000CC	00	00	00	00	00	00	00	00	00
080000D8	80	84	00	AF	40	F6	08	03	C4
080000E4	18	68	D8	82	18	46	D8	46	80
080000F0	80	85	82	80	00	AF	00	F0	2F
080000FC	00	FF	F8	F8	78	68	14	28	06
08000108	5F	F8	F8	F8	78	68	14	28	06
08000114	00	06	FA	F8	4F	FA	F0	70	00
08000120	E8	07	00	8F	80	84	83	80	00
0800012C	03	07	F8	68	03	F1	FF	33	78
08000138	00	28	F8	0F	07	F1	0C	07	BD
08000144	70	47	0D	8F	80	84	00	AF	14
08000150	03	C4	03	C4	03	C4	03	C4	03
0800015C	12	68	42	0F	04	02	1A	60	4F
08000168	C4	F2	01	03	4F	00	00	62	C4
08000174	12	68	22	10	70	02	1A	60	4F
08000180	C4	F2	01	03	4F	00	00	62	C4
0800018C	12	68	1A	60	4F	00	00	62	C4
08000198	00	00	00	00	00	00	00	00	00
080001A4	70	02	1A	60	40	F6	04	03	C4
080001B0	40	F6	04	02	C4	F2	01	02	12
080001BC	22	1A	60	40	60	8C	70	00	00
080001C8	80	84	00	AF	40	F6	0C	03	C4
080001D4	00	00	00	00	00	00	00	00	00
080001E0	00	52	1A	60	80	46	00	8C	70

File Edit View Tool Design Graph Debug Library Template System Help

CM3 Source Code - U1

```

-----
//include "driver.h"
#include "alarm_monitor.h"
#include "pressure_sensor.h"

8000F0 int main(){
8000F6     GPIO_INITIALIZATION();
-----
int pressure_val;
-----
while (1)
{
8000FA     pressure_val = getPressureVal();
-----
/*max bar is 20, if pressure_val >20 al
8000100 if(pressure_val > 20){ // alarm on
8000106     Set_Alarm_On(); // alarm on
800010A     Delay(1000); // time delay
-----
} /*end if condation bracket*/
-----
/*pressure_val <= 20 alarm is off =
8000114     else { Set_Alarm_Off(); // alarm off
8000118     Delay(1000);
-----
} /*end else bracket*/
-----
//Implement your Design
8000112 return 0;
-----
}
-----
}

```

CM3 Registers - U1

PC	08000100	Privileged
Pr1	00.200	Mode Thread
R0	00000008	R7 20001FD0
R1	00000000	R8 00000000
R2	00002200	R9 00000000
R3	00000008	R10 00000000
R4	00000000	R11 00000000
R5	00000000	R12 00000000
R6	00000000	LR 080000FF
MSP*	20001FD0	PSP 00000000
IRQ*	0	
APSR	CNVQZ 10001	EPSP 01000000
		1c11t 000.00000

CM3 FLASH at 0x08000000 - U1

Address	Value
08000000	00 20 00 20 29 00 00 08 1D 0C
0800000C	1D 00 00 08 1D 00 00 08 1D 0C
08000018	1D 00 00 08 80 85 00 AF 00 FC
08000024	80 8D 00 BF 80 85 86 80 00 AF
08000030	00 02 C2 F2 00 02 40 F2 00 03
0800003C	00 03 D3 1A 88 60 40 F2 10 23
08000048	00 03 38 61 40 F2 00 03 C2 F2
08000054	F8 60 4F F0 00 03 78 61 0F EC
08000060	1A 78 F8 68 1A 70 F8 68 03 F1
0800006C	F8 60 38 69 03 F1 01 03 38 61
08000078	03 F1 01 03 78 61 7A 69 BB 65
08000084	EB D3 40 F2 00 02 C2 F2 00 02
08000090	00 03 C2 F2 00 03 D3 1A F8 71
0800009C	00 03 C2 F2 00 03 F8 60 4F FC
080000A8	78 61 08 E0 F8 68 4F F0 00 02
080000B4	F8 68 03 F1 01 03 F8 60 78 65
080000C0	01 03 78 61 FA 79 78 69 9A 42
080000CC	00 F0 10 F8 07 F1 18 07 BD 46
080000D8	80 84 00 AF 40 F6 08 03 C4 F2
080000E4	18 68 DB B2 18 46 80 46 80 BC
080000F0	80 85 82 80 00 AF 00 F0 27 F8
080000FC	ED FF 78 60 78 68 14 28 06 DD
08000108	5F F8 4F F4 7A 70 00 F0 09 F8
08000114	00 F0 6A F8 4F F4 7A 70 00 FC
08000120	EB E7 00 BF 80 84 83 80 00 AF

Controller_HM

Mastering Embedded System Online Diploma (KS)

www.learn-in-depth.com

First Term Project 1

Eng: Hossam Magdy Afifi

CM3 FLASH at 0x08000000 - U1

➤ Make File:

```

amagd@DESKTOP-C174UHM MINGW32 /f/material/Automotive Embedded System Course/kerolis shon
oda/course embededd system online/first term project/Project2/source_code
$ mingw32-make all
arm-none-eabi-gcc.exe -c -I . -mcpu=cortex-m3 -mthumb -gdwarf-2 startup.c -o startup.o
#####Finished compiler#####
arm-none-eabi-gcc.exe -c -I . -mcpu=cortex-m3 -mthumb -gdwarf-2 pressure_sensor.c -o pr
essure_sensor.o
#####Finished compiler#####
arm-none-eabi-gcc.exe -c -I . -mcpu=cortex-m3 -mthumb -gdwarf-2 main.c -o main.o
#####Finished compiler#####
arm-none-eabi-gcc.exe -c -I . -mcpu=cortex-m3 -mthumb -gdwarf-2 driver.c -o driver.o
#####Finished compiler#####
arm-none-eabi-gcc.exe -c -I . -mcpu=cortex-m3 -mthumb -gdwarf-2 alarm_monitor.c -o alar
m_monitor.o
#####Finished compiler#####
arm-none-eabi-ld.exe -T linker_script.ld -nostartfiles startup.o pressure_sensor.o m
ain.o driver.o alarm_monitor.o -o Pressure_Detection.elf -Map=output.map
Finished linker_script
arm-none-eabi-objcopy.exe -O binary Pressure_Detection.elf Pressure_Detection.hex
#####Finished hex file#####
#####DONE ALL#####

```

➤ SW analysis:

```
MINGW32:/f/material/Automotive Embedded System Course/kerolis shonoda/course embededd system online/first term project/Project2/source_code
oda/course embededd system online/first term project/Project2/source_code
$ arm-none-eabi-nm.exe driver.o
00000000 T Delay
00000024 T GPIO_INITIALIZATION

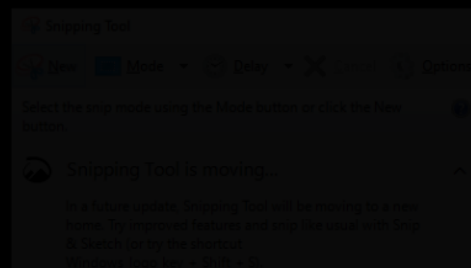
amagd@DESKTOP-C174UHM MINGW32 /f/material/Automotive Embedded System Course/kerolis shonoda/course embededd system online/first term project/Project2/source_code
$ arm-none-eabi-nm.exe alarm_monitor.o
00000024 T Set_Alarm_Off
00000000 T Set_Alarm_On

amagd@DESKTOP-C174UHM MINGW32 /f/material/Automotive Embedded System Course/kerolis shonoda/course embededd system online/first term project/Project2/source_code
$ arm-none-eabi-nm.exe pressure_sensor.o
00000000 T getPressureVal

amagd@DESKTOP-C174UHM MINGW32 /f/material/Automotive Embedded System Course/kerolis shonoda/course embededd system online/first term project/Project2/source_code
$ arm-none-eabi-nm.exe startup.o
U _E_bss
U _E_DATA
U _E_text
U _S_bss
U _S_DATA
U _stack_top
00000000 W Bus_fault_Handler
00000000 T Default_Handler
00000000 W H_fault_Handler
U main
00000000 W MM_fault_Handler
00000000 W NMT_Handler
0000000c T Reset_Handler
00000000 W Usage_fault_Handler
00000000 D vectors
```

```
amagd@DESKTOP-C174UHM MINGW32 /f/material/Automotive Embedded System Course/kerolis shonoda/course embededd system online/first term project/Project2/source_code
$ arm-none-eabi-nm.exe main.o
U Delay
U getPressureVal
U GPIO_INITIALIZATION
00000000 T main
U Set_Alarm_Off
U Set_Alarm_On

amagd@DESKTOP-C174UHM MINGW32 /f/material/Automotive Embedded System Course/kerolis shonoda/course embededd system online/first term project/Project2/source_code
$ arm-none-eabi-nm.exe Pressure_Detection.elf
20000000 B _E_bss
20000000 T _E_DATA
08000214 T _E_text
20001000 B _heap_End
20000000 B _S_bss
20000000 T _S_DATA
20002000 B _stack_top
0800001c W Bus_fault_Handler
0800001c T Default_Handler
08000128 T Delay
080000d8 T getPressureVal
0800014c T GPIO_INITIALIZATION
0800001c W H_fault_Handler
080000f0 T main
0800001c W MM_fault_Handler
0800001c W NMT_Handler
08000028 T Reset_Handler
080001f0 T Set_Alarm_Off
080001cc T Set_Alarm_On
0800001c W Usage_fault_Handler
08000000 T vectors
```



```
MINGW32:/f/material/Automotive Embedded System Course/kerolis shonoda/course embededd system online/first term project/Project2/source_code
amagd@DESKTOP-C174UHM MINGW32 /f/material/Automotive Embedded System Course/kerolis shonoda/course embededd system online/first term project/Project2/source_code
$ arm-none-eabi-objdump.exe -h alarm_monitor.o

alarm_monitor.o:      file format elf32-littlearm

Sections:
Idx Name              Size      VMA       LMA       File off  Algn
 0 .text              00000048  00000000  00000000  00000034  2**2
CONTENTS, ALLOC, LOAD, READONLY, CODE
 1 .data              00000000  00000000  00000000  0000007c  2**0
CONTENTS, ALLOC, LOAD, DATA
 2 .bss               00000000  00000000  00000000  0000007c  2**0
ALLOC
 3 .debug_info        000000af  00000000  00000000  0000007c  2**0
CONTENTS, RELOC, READONLY, DEBUGGING
 4 .debug_abbrev      0000004d  00000000  00000000  0000012b  2**0
CONTENTS, READONLY, DEBUGGING
 5 .debug_loc         00000058  00000000  00000000  00000178  2**0
CONTENTS, READONLY, DEBUGGING
 6 .debug_aranges     00000020  00000000  00000000  000001d0  2**0
CONTENTS, RELOC, READONLY, DEBUGGING
 7 .debug_line        00000087  00000000  00000000  000001f0  2**0
CONTENTS, RELOC, READONLY, DEBUGGING
 8 .debug_str         00000156  00000000  00000000  00000277  2**0
CONTENTS, READONLY, DEBUGGING
 9 .comment           00000012  00000000  00000000  000003cd  2**0
CONTENTS, READONLY
10 .ARM.attributes    00000033  00000000  00000000  000003df  2**0
CONTENTS, READONLY
11 .debug_frame       00000040  00000000  00000000  00000414  2**2
CONTENTS, RELOC, READONLY, DEBUGGING
```

```
MINGW32:/f/material/Automotive Embedded System Course/kerolis shonoda/course embededd system online/first term project/Project2/source_code
amagd@DESKTOP-C174UHM MINGW32 /f/material/Automotive Embedded System Course/kerolis shonoda/course embededd system online/first term project/Project2/source_code
$ 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              000000a4  00000000  00000000  00000034  2**2
CONTENTS, ALLOC, LOAD, READONLY, CODE
 1 .data              00000000  00000000  00000000  000000d8  2**0
CONTENTS, ALLOC, LOAD, DATA
 2 .bss               00000000  00000000  00000000  000000d8  2**0
ALLOC
 3 .debug_info        000000c3  00000000  00000000  000000d8  2**0
CONTENTS, RELOC, READONLY, DEBUGGING
 4 .debug_abbrev      00000076  00000000  00000000  0000019b  2**0
CONTENTS, READONLY, DEBUGGING
 5 .debug_loc         00000064  00000000  00000000  00000211  2**0
CONTENTS, READONLY, DEBUGGING
 6 .debug_aranges     00000020  00000000  00000000  00000275  2**0
CONTENTS, RELOC, READONLY, DEBUGGING
 7 .debug_line        00000090  00000000  00000000  00000295  2**0
CONTENTS, RELOC, READONLY, DEBUGGING
 8 .debug_str         00000155  00000000  00000000  00000325  2**0
CONTENTS, READONLY, DEBUGGING
 9 .comment           00000012  00000000  00000000  0000047a  2**0
CONTENTS, READONLY
10 .ARM.attributes    00000033  00000000  00000000  0000048c  2**0
CONTENTS, READONLY
11 .debug_frame       00000044  00000000  00000000  000004c0  2**2
CONTENTS, RELOC, READONLY, DEBUGGING
```

```
amagd@DESKTOP-C174UHM MINGW32 /f/material/Automotive Embedded System Course/kerolis shonoda/course embededd system online/first term project/Project2/source_code
$ arm-none-eabi-objdump.exe -D Pressure_Detection.elf
```

```
Pressure_Detection.elf:      file format elf32-littlearm
```

```
Disassembly of section .text:
```

```
08000000 <vectors>:
80000000:      20002000      andcs   r2, r0, r0
80000004:      08000029      stmdbeq r0, {r0, r3, r5}
80000008:      0800001d      stmdbeq r0, {r0, r2, r3, r4}
8000000c:      0800001d      stmdbeq r0, {r0, r2, r3, r4}
80000010:      0800001d      stmdbeq r0, {r0, r2, r3, r4}
80000014:      0800001d      stmdbeq r0, {r0, r2, r3, r4}
80000018:      0800001d      stmdbeq r0, {r0, r2, r3, r4}

0800001c <Default_Handler>:
8000001c:      b580          push    {r7, lr}
8000001e:      af00          add     r7, sp, #0
80000020:      f000 f802     bl      80000028 <Reset_Handler>
80000024:      bd80          pop     {r7, pc}
80000026:      bf00          nop

08000028 <Reset_Handler>:
80000028:      b580          push    {r7, lr}
8000002a:      b086          sub     sp, #24
8000002c:      af00          add     r7, sp, #0
8000002e:      f240 0200     movw   r2, #0
80000032:      f2c2 0200     movt   r2, #8192      ; 0x2000
80000036:      f240 0300     movw   r3, #0
8000003a:      f2c2 0300     movt   r3, #8192      ; 0x2000
8000003e:      1ad3          subs   r3, r2, r3
80000040:      60bb          str     r3, [r7, #8]
80000042:      f240 2314     movw   r3, #532       ; 0x214
80000046:      f6c0 0300     movt   r3, #2048      ; 0x800
8000004a:      613b          str     r3, [r7, #16]
8000004c:      f240 0300     movw   r3, #0
80000050:      f2c2 0300     movt   r3, #8192      ; 0x2000
80000054:      60fb          str     r3, [r7, #12]
```

Activate Windows
Go to Settings to activate Windows.

```
080000d8 <getPressureVal>:
800000d8:      b480          push    {r7}
800000da:      af00          add     r7, sp, #0
800000dc:      f640 0308     movw   r3, #2056      ; 0x808
800000e0:      f2c4 0301     movt   r3, #16385     ; 0x4001
800000e4:      681b          ldr     r3, [r3, #0]
800000e6:      b2db          uxtb   r3, r3
800000e8:      4618          mov     r0, r3
800000ea:      46bd          mov     sp, r7
800000ec:      bc80          pop     {r7}
800000ee:      4770          bx      lr

080000f0 <main>:
800000f0:      b580          push    {r7, lr}
800000f2:      b082          sub     sp, #8
800000f4:      af00          add     r7, sp, #0
800000f6:      f000 f829     bl      8000014c <GPIO_INITIALIZATION>
800000fa:      f7ff ffed     bl      800000d8 <getPressureVal>
800000fe:      6078          str     r0, [r7, #4]
8000100:      687b          ldr     r3, [r7, #4]
8000102:      2b14          cmp     r3, #20
8000104:      dd0c          ble.n   80001120 <main+0x30>
8000106:      f000 f861     bl      800011cc <Set_Alarm_On>
800010a:      f04f 00c8     mov.w   r0, #200      ; 0xc8
800010e:      f000 f80b     bl      80001128 <Delay>
8000112:      f000 f86d     bl      800011f0 <Set_Alarm_Off>
8000116:      f04f 00c8     mov.w   r0, #200      ; 0xc8
800011a:      f000 f805     bl      80001128 <Delay>
800011e:      e7ec          b.n     800000fa <main+0xa>
8000120:      f000 f866     bl      800011f0 <Set_Alarm_Off>
8000124:      e7e9          b.n     800000fa <main+0xa>
8000126:      bf00          nop

08000128 <Delay>:
8000128:      b480          push    {r7}
800012a:      b083          sub     sp, #12
800012c:      af00          add     r7, sp, #0
800012e:      6078          str     r0, [r7, #4]
8000130:      e003          b.n     8000133a <Delay+0x12>
8000132:      687b          ldr     r3, [r7, #4]
8000134:      f103 33ff     add.w   r3, r3, #4294967295
```

Activate Windows
Go to Settings to activate Windows.


```

0800014c: <GPIO_INITIALIZATION>:
800014c: b480          push    {r7}
800014e: af00          add     r7, sp, #0
8000150: f241 0318     movw   r3, #4120          ; 0x1018
8000154: f2c4 0302     movt   r3, #16386       ; 0x4002
8000158: f241 0218     movw   r2, #4120          ; 0x1018
800015c: f2c4 0202     movt   r2, #16386       ; 0x4002
8000160: 6812          ldr     r2, [r2, #0]
8000162: f042 0204     orr.w  r2, r2, #4
8000166: 601a          str     r2, [r3, #0]
8000168: f44f 6300     mov.w  r3, #2048         ; 0x800
800016c: f2c4 0301     movt   r3, #16385       ; 0x4001
8000170: f44f 6200     mov.w  r2, #2048         ; 0x800
8000174: f2c4 0201     movt   r2, #16385       ; 0x4001
8000178: 6812          ldr     r2, [r2, #0]
800017a: f422 0270     bic.w  r2, r2, #15728640 ; 0xf00000
800017e: 601a          str     r2, [r3, #0]
8000180: f44f 6300     mov.w  r3, #2048         ; 0x800
8000184: f2c4 0301     movt   r3, #16385       ; 0x4001
8000188: f44f 6200     mov.w  r2, #2048         ; 0x800
800018c: f2c4 0201     movt   r2, #16385       ; 0x4001
8000190: 6812          ldr     r2, [r2, #0]
8000192: 601a          str     r2, [r3, #0]
8000194: f640 0304     movw   r3, #2052         ; 0x804
8000198: f2c4 0301     movt   r3, #16385       ; 0x4001
800019c: f640 0204     movw   r2, #2052         ; 0x804
80001a0: f2c4 0201     movt   r2, #16385       ; 0x4001
80001a4: 6812          ldr     r2, [r2, #0]
80001a6: f422 0270     bic.w  r2, r2, #15728640 ; 0xf00000
80001aa: 601a          str     r2, [r3, #0]
80001ac: f640 0304     movw   r3, #2052         ; 0x804
80001b0: f2c4 0301     movt   r3, #16385       ; 0x4001
80001b4: f640 0204     movw   r2, #2052         ; 0x804
80001b8: f2c4 0201     movt   r2, #16385       ; 0x4001
80001bc: 6812          ldr     r2, [r2, #0]
80001be: f042 3222     orr.w  r2, r2, #572662306 ; 0x22222222
80001c2: 601a          str     r2, [r3, #0]
80001c4: 46bd          mov     sp, r7
80001c6: bc80          pop     {r7}
80001c8: 4770          bx      lr
80001ca: bf00          nop

```

Activate Windows
Go to Settings to activate Windows.

Disassembly of section .bss:

```

20000000 <_E_bss>:
...

```

```

20001000 <_heap_End>:
...

```

Disassembly of section .debug_info:

```

00000000 <.debug_info>:
0: 00000164      andeq    r0, r0, r4, ror #2
4: 00000002      andeq    r0, r0, r2
8: 01040000      mrseq    r0, (UNDEF: 4)
c: 000000c7      andeq    r0, r0, r7, asr #1
10: 00000001      andeq    r0, r0, r1
14: 0000e500      andeq    lr, r0, r0, lsl #10
18: 00001c00      andeq    r1, r0, r0, lsl #24
1c: 0000d808      andeq    sp, r0, r8, lsl #16
20: 00000008      andeq    r0, r0, r8
24: 06010200      streq    r0, [r1], -r0, lsl #4
28: 0000003c      andeq    r0, r0, ip, lsr r0
2c: 3a080102      bcc      20043c <vectors-0x7dffc4>
30: 02000000      andeq    r0, r0, #0
34: 00db0502      sbcseq   r0, fp, r2, lsl #10
38: 02020000      andeq    r0, r2, #0
3c: 00004807      andeq    r4, r0, r7, lsl #16
40: 05040200      streq    r0, [r4, #-512] ; 0x200
44: 000000ae      andeq    r0, r0, lr, lsr #1
48: 00016903      andeq    r6, r1, r3, lsl #18
4c: 53500200      cmppl    r0, #0, 4
50: 02000000      andeq    r0, r0, #0
54: 00780704      rsbseq   r0, r8, r4, lsl #14
58: 08020000      stmdaeq  r2, {} ; <UNPREDICTABLE>
5c: 0000a905      andeq    s1, r0, r5, lsl #18
60: 07080200      streq    r0, [r8, -r0, lsl #4]
64: 00000073      andeq    r0, r0, r3, ror r0
68: 69050404      stmdbvs  r5, {r2, s1}
6c: 0200746e      andeq    r7, r0, #1845493760 ; 0x6e000000
70: 007d0704      rsbseq   r0, sp, r4, lsl #14

```

Activate Windows
Go to Settings to activate Windows.

```

amagd@DESKTOP-C174UHM MINGW32 /f/material/Automotive Embedded System Course/kerolis shon
oda/course embededd system online/first term project/Project2/source_code
$ 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	00000038	00000000	00000000	00000034	2**2
	CONTENTS, ALLOC, LOAD, RELOC, READONLY, CODE					
1	.data	00000000	00000000	00000000	0000006c	2**0
	CONTENTS, ALLOC, LOAD, DATA					
2	.bss	00000000	00000000	00000000	0000006c	2**0
	ALLOC					
3	.debug_info	00000085	00000000	00000000	0000006c	2**0
	CONTENTS, RELOC, READONLY, DEBUGGING					
4	.debug_abbrev	0000007a	00000000	00000000	000000f1	2**0
	CONTENTS, READONLY, DEBUGGING					
5	.debug_loc	00000038	00000000	00000000	0000016b	2**0
	CONTENTS, READONLY, DEBUGGING					
6	.debug_aranges	00000020	00000000	00000000	000001a3	2**0
	CONTENTS, RELOC, READONLY, DEBUGGING					
7	.debug_line	00000040	00000000	00000000	000001c3	2**0
	CONTENTS, RELOC, READONLY, DEBUGGING					
8	.debug_str	000000c3	00000000	00000000	00000203	2**0
	CONTENTS, READONLY, DEBUGGING					
9	.comment	00000012	00000000	00000000	000002c6	2**0
	CONTENTS, READONLY					
10	.ARM.attributes	00000033	00000000	00000000	000002d8	2**0
	CONTENTS, READONLY					
11	.debug_frame	00000030	00000000	00000000	0000030c	2**2
	CONTENTS, RELOC, READONLY, DEBUGGING					

```

amagd@DESKTOP-C174UHM MINGW32 /f/material/Automotive Embedded System Course/kerolis shon
oda/course embededd system online/first term project/Project2/source_code
$ arm-none-eabi-objdump.exe -h pressure_sensor.o

```

```

pressure_sensor.o:      file format elf32-littlearm

```

Sections:

Idx	Name	Size	VMA	LMA	File off	Algn
0	.text	00000018	00000000	00000000	00000034	2**2
	CONTENTS, ALLOC, LOAD, READONLY, CODE					
1	.data	00000000	00000000	00000000	0000004c	2**0
	CONTENTS, ALLOC, LOAD, DATA					
2	.bss	00000000	00000000	00000000	0000004c	2**0
	ALLOC					
3	.debug_info	0000009e	00000000	00000000	0000004c	2**0
	CONTENTS, RELOC, READONLY, DEBUGGING					
4	.debug_abbrev	0000004f	00000000	00000000	000000ea	2**0
	CONTENTS, READONLY, DEBUGGING					
5	.debug_loc	0000002c	00000000	00000000	00000139	2**0
	CONTENTS, READONLY, DEBUGGING					
6	.debug_aranges	00000020	00000000	00000000	00000165	2**0
	CONTENTS, RELOC, READONLY, DEBUGGING					
7	.debug_line	00000086	00000000	00000000	00000185	2**0
	CONTENTS, RELOC, READONLY, DEBUGGING					
8	.debug_str	0000014c	00000000	00000000	0000020b	2**0
	CONTENTS, READONLY, DEBUGGING					
9	.comment	00000012	00000000	00000000	00000357	2**0
	CONTENTS, READONLY					
10	.ARM.attributes	00000033	00000000	00000000	00000369	2**0
	CONTENTS, READONLY					
11	.debug_frame	00000028	00000000	00000000	0000039c	2**2
	CONTENTS, RELOC, READONLY, DEBUGGING					

```
amagd@DESKTOP-C174UHM MINGW32 /f/material/Automotive Embedded System Course/kerolis shon
oda/course embededd system online/first term project/Project2/source_code
$ arm-none-eabi-objdump.exe -h startup.o
```

```
startup.o:      file format elf32-littlearm
```

```
Sections:
```

Idx	Name	Size	VMA	LMA	File off	Algn
0	.text	000000bc	00000000	00000000	00000034	2**2
	CONTENTS, ALLOC, LOAD, RELOC, READONLY, CODE					
1	.data	00000000	00000000	00000000	000000f0	2**0
	CONTENTS, ALLOC, LOAD, DATA					
2	.bss	00000000	00000000	00000000	000000f0	2**0
	ALLOC					
3	.vector	0000001c	00000000	00000000	000000f0	2**2
	CONTENTS, ALLOC, LOAD, RELOC, DATA					
4	.debug_info	00000168	00000000	00000000	0000010c	2**0
	CONTENTS, RELOC, READONLY, DEBUGGING					
5	.debug_abbrev	000000c2	00000000	00000000	00000274	2**0
	CONTENTS, READONLY, DEBUGGING					
6	.debug_loc	00000064	00000000	00000000	00000336	2**0
	CONTENTS, READONLY, DEBUGGING					
7	.debug_aranges	00000020	00000000	00000000	0000039a	2**0
	CONTENTS, RELOC, READONLY, DEBUGGING					
8	.debug_line	000000ae	00000000	00000000	000003ba	2**0
	CONTENTS, RELOC, READONLY, DEBUGGING					
9	.debug_str	000001a6	00000000	00000000	00000468	2**0
	CONTENTS, READONLY, DEBUGGING					
10	.comment	00000012	00000000	00000000	0000060e	2**0
	CONTENTS, READONLY					
11	.ARM.attributes	00000033	00000000	00000000	00000620	2**0
	CONTENTS, READONLY					
12	.debug_frame	0000004c	00000000	00000000	00000654	2**2
	CONTENTS, RELOC, READONLY, DEBUGGING					

Activate Windows