HIGH PRESSURE DETECTION

Mastering Embedded System Online Diploma

www.learn-in-depth.com

First Term (First project)

Eng : Abdallah Ahmed Mohammed Ibrahim

My profile :

https://www.learn-in-depth.com/online-diploma/abdallahahmed17120%40gmail.com

CONTENT :

Case study

Requirement

System analysis

System design

Sections & symbols for object files

Startup .c

Linker script

Simulation on proteus

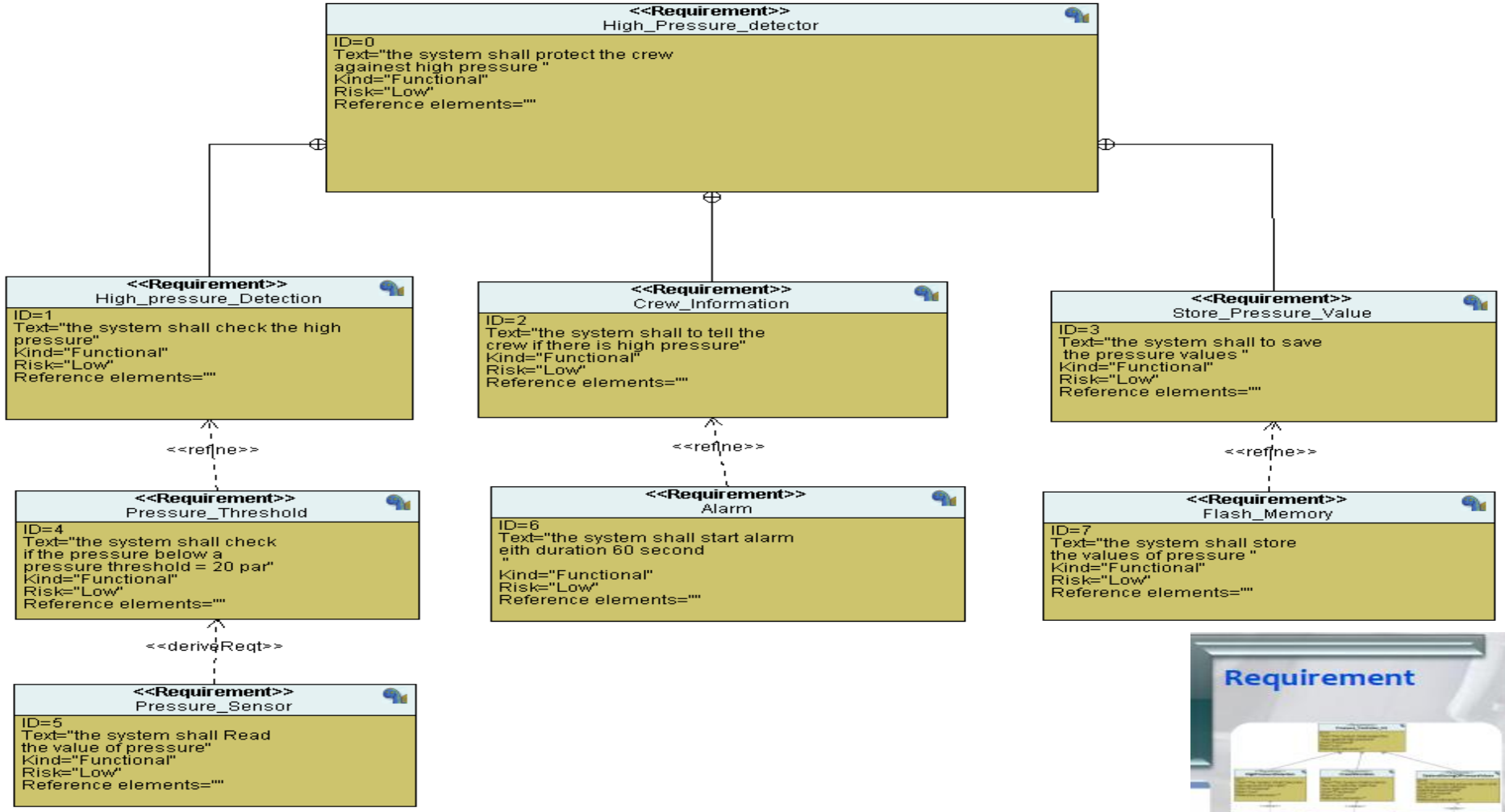
Case study : a pressure Detection system

* A pressure controller informs the crew of a cabin with an alarm when the pressure exceeds 20 par in the cabin .

* The alarm duration will be 60 seconds .

* Store the values of pressure (in another version).

Requirement node



System analysis

A. Use case diagram

B. Activity diagram

C. Sequence diagram

System analysis

A. Use case diagram

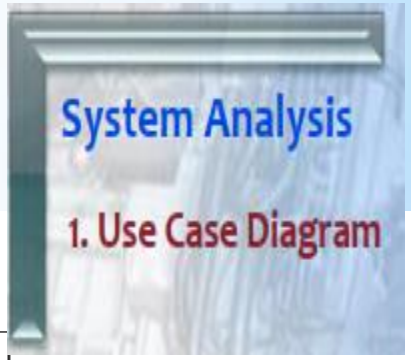
The diagram show that pressure sensor will send a signal to main algorithm .

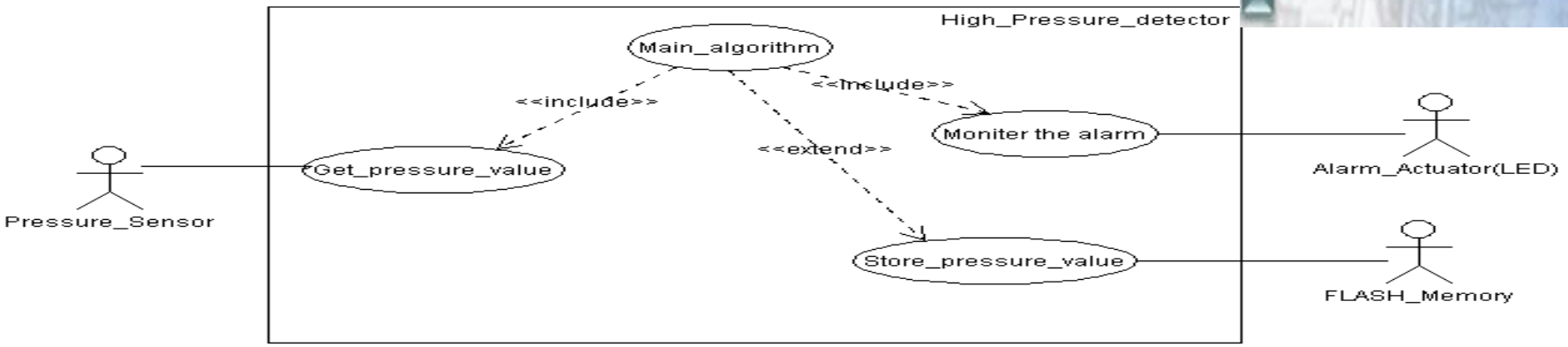
The main algorithm will check if the value of pressure sensor is more than threshold or not.

If the value is more than threshold , the main algorithm will send to monitor signal (high pressure detection).

The monitor will alarm Led to turn on and then turn off.

There is an extend option to store these values (in another version).





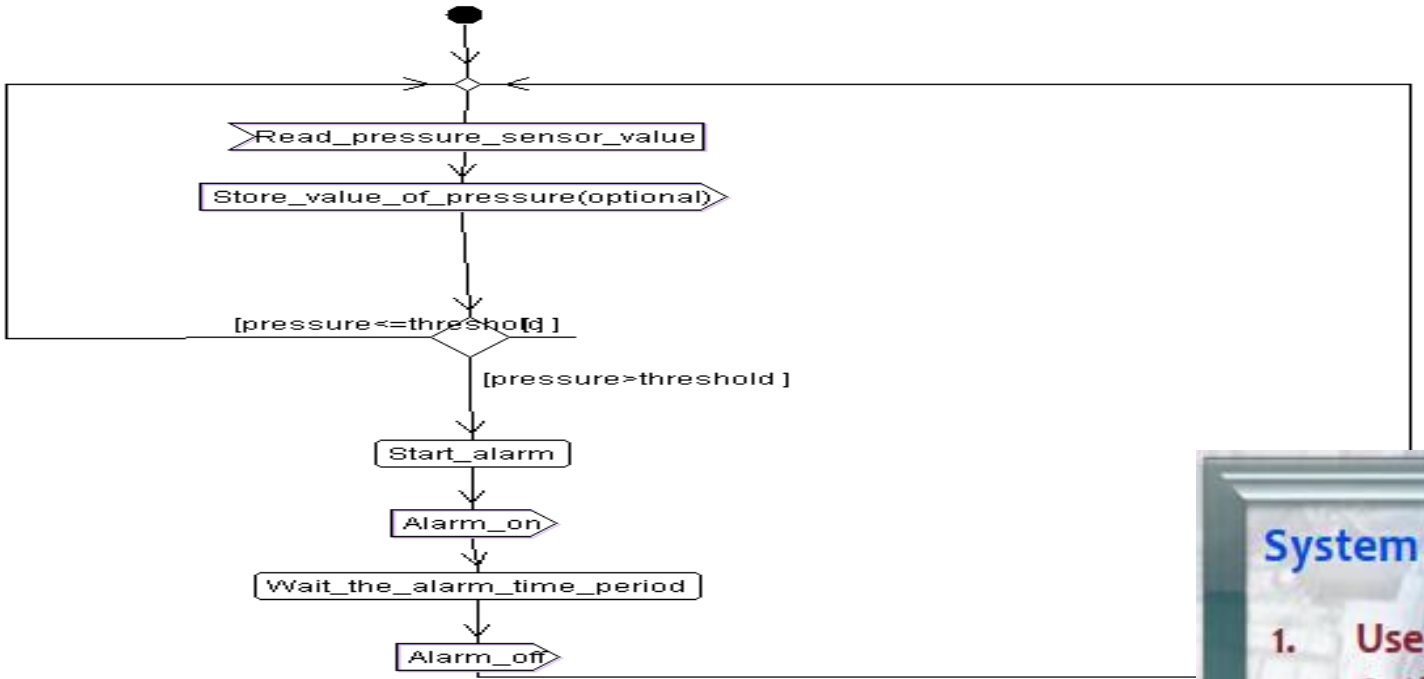
System analysis

B. Activity diagram

By reading the value of pressure sensor if it more than threshold :

This value will stored in FLASH and then send signal to start alarm.

The alarm duration will be for a period of time and then the alarm will be off.



System Analysis
1. Use Case Diagram
2. Activity Diagram

System analysis

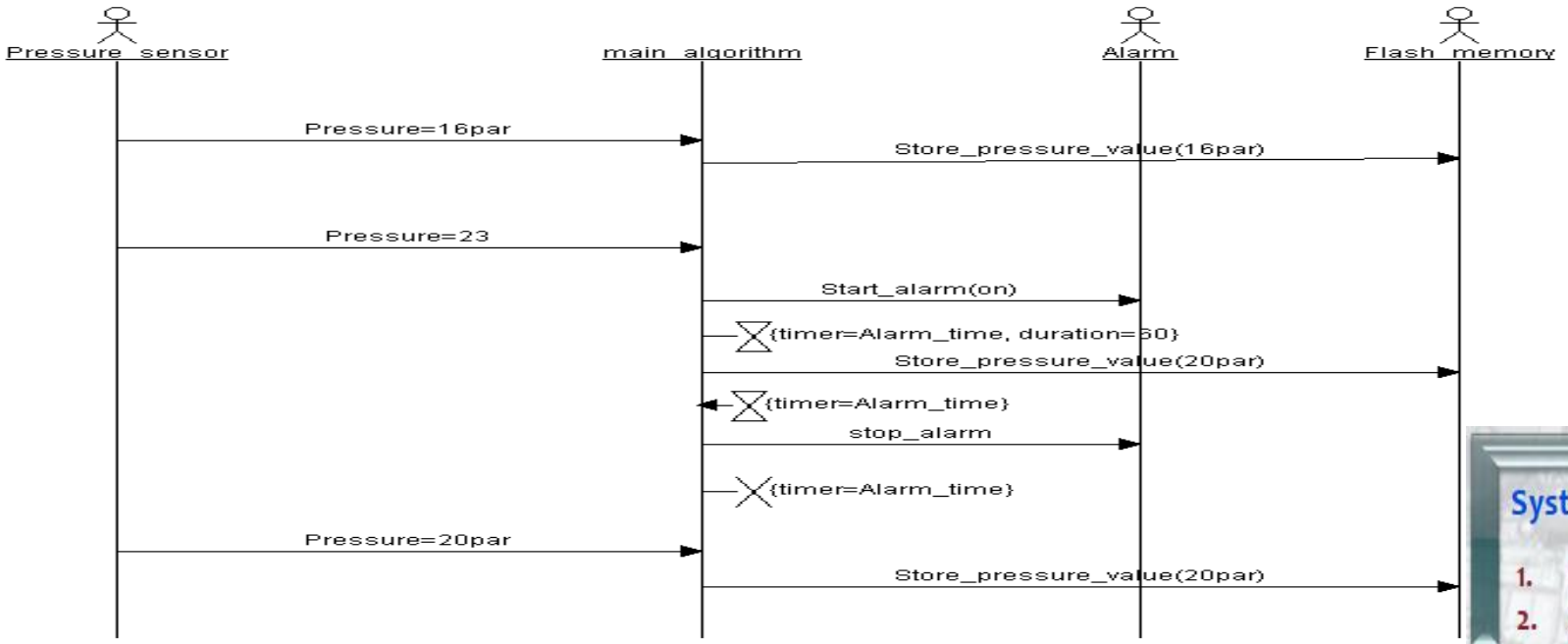
C. Sequence diagram

The diagram show the communication between systems.

When the pressure sensor send 16 par this value is below the pressure threshold .

So will store this value and wait for the second one.

Lock at the second value of 23 upper than threshold so will doing two function now the first one send signal to start alarm and the another one store this value .



System design

A. BLOCK DIAGRAM

B. State machine : Pressure Sensor driver

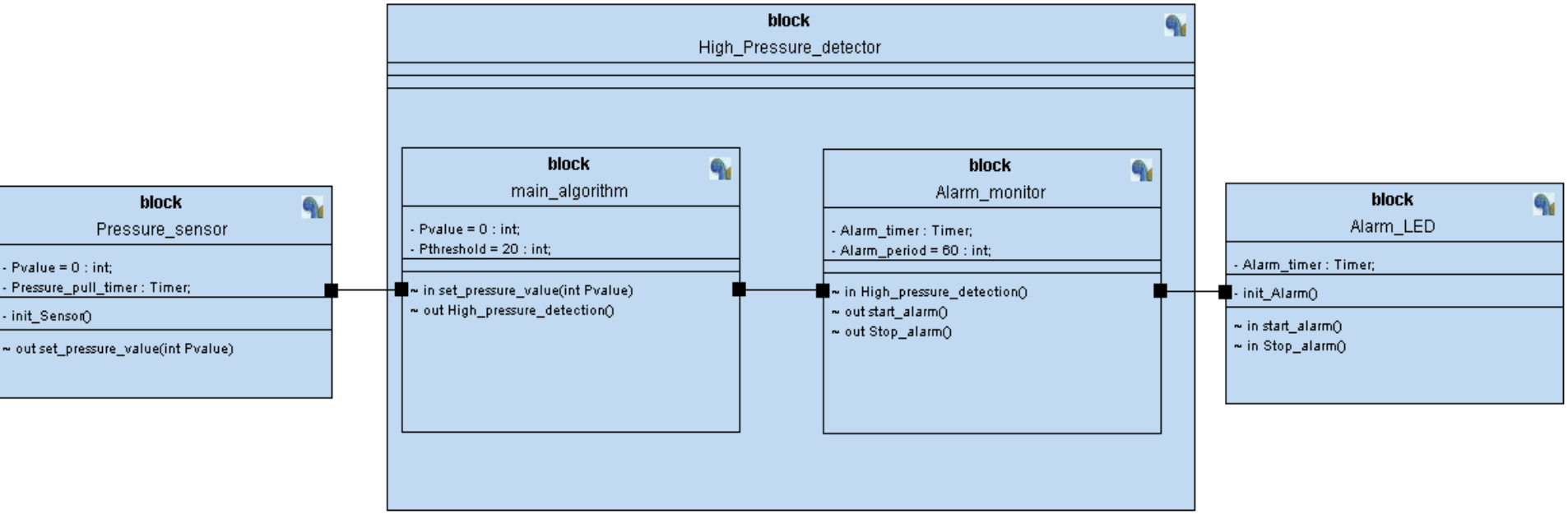
C. State machine : Main controller

D. State machine : Alarm monitor

E. State machine : Alarm actuator driver

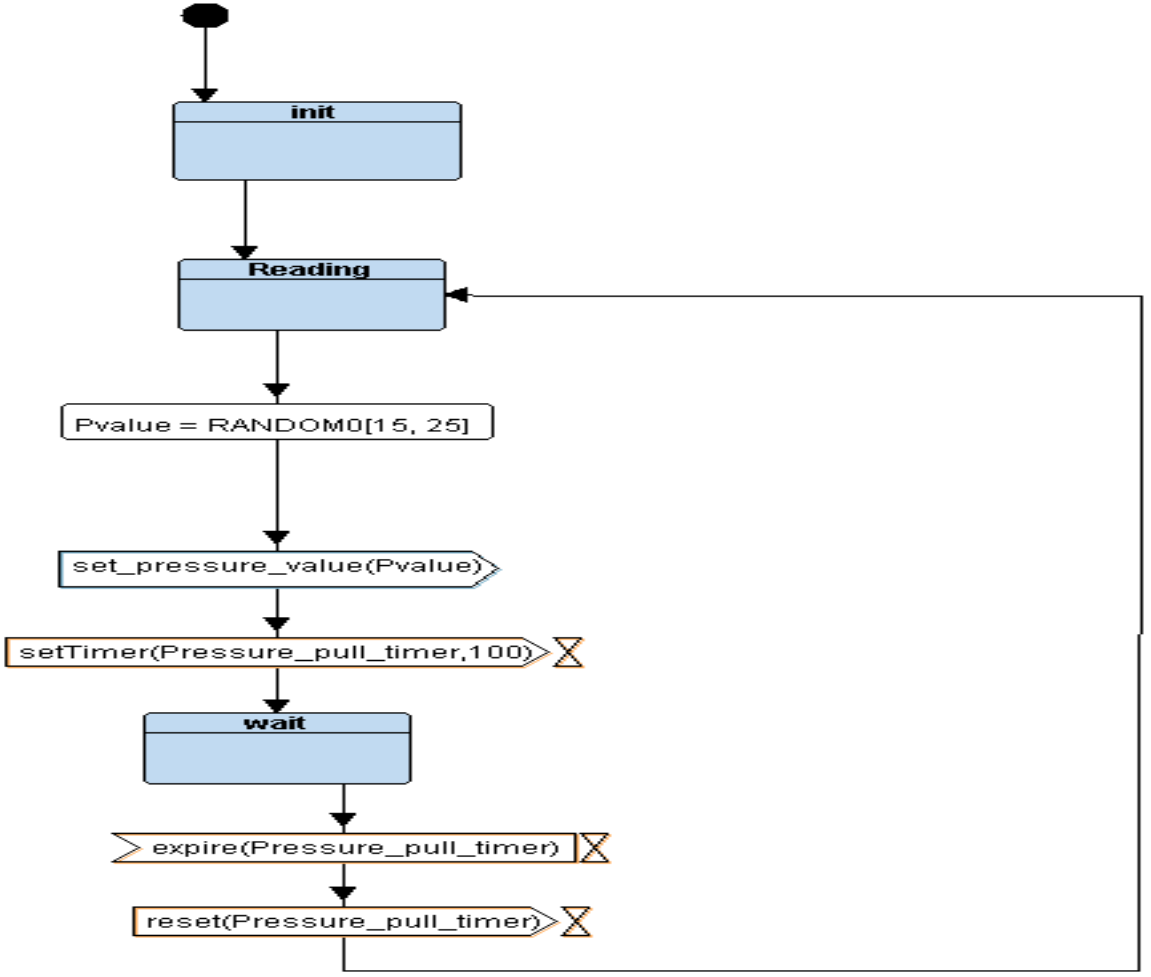
System design

A. BLOCK DIAGRAM



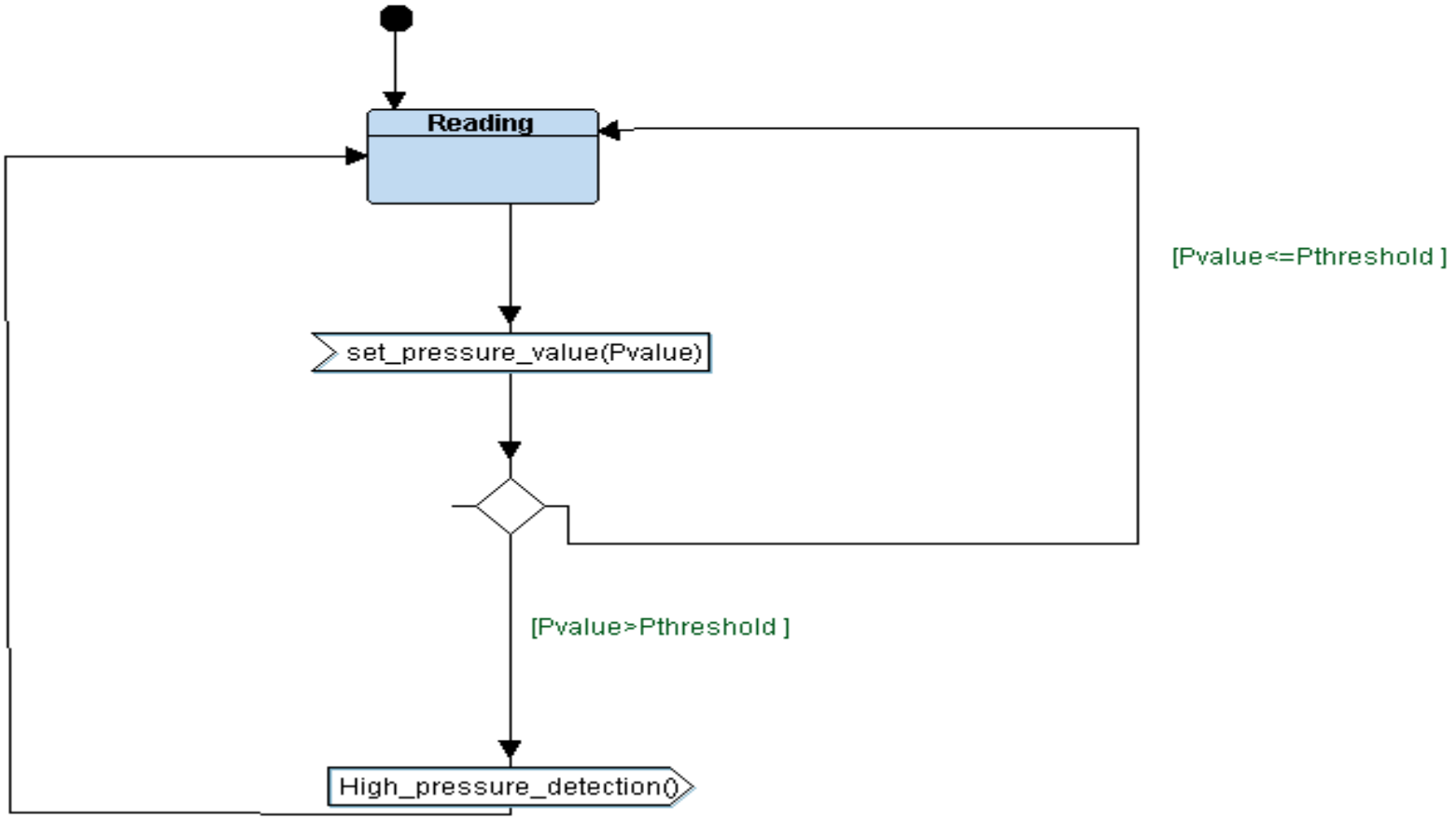
System design

B. State machine : Pressure Sensor drive



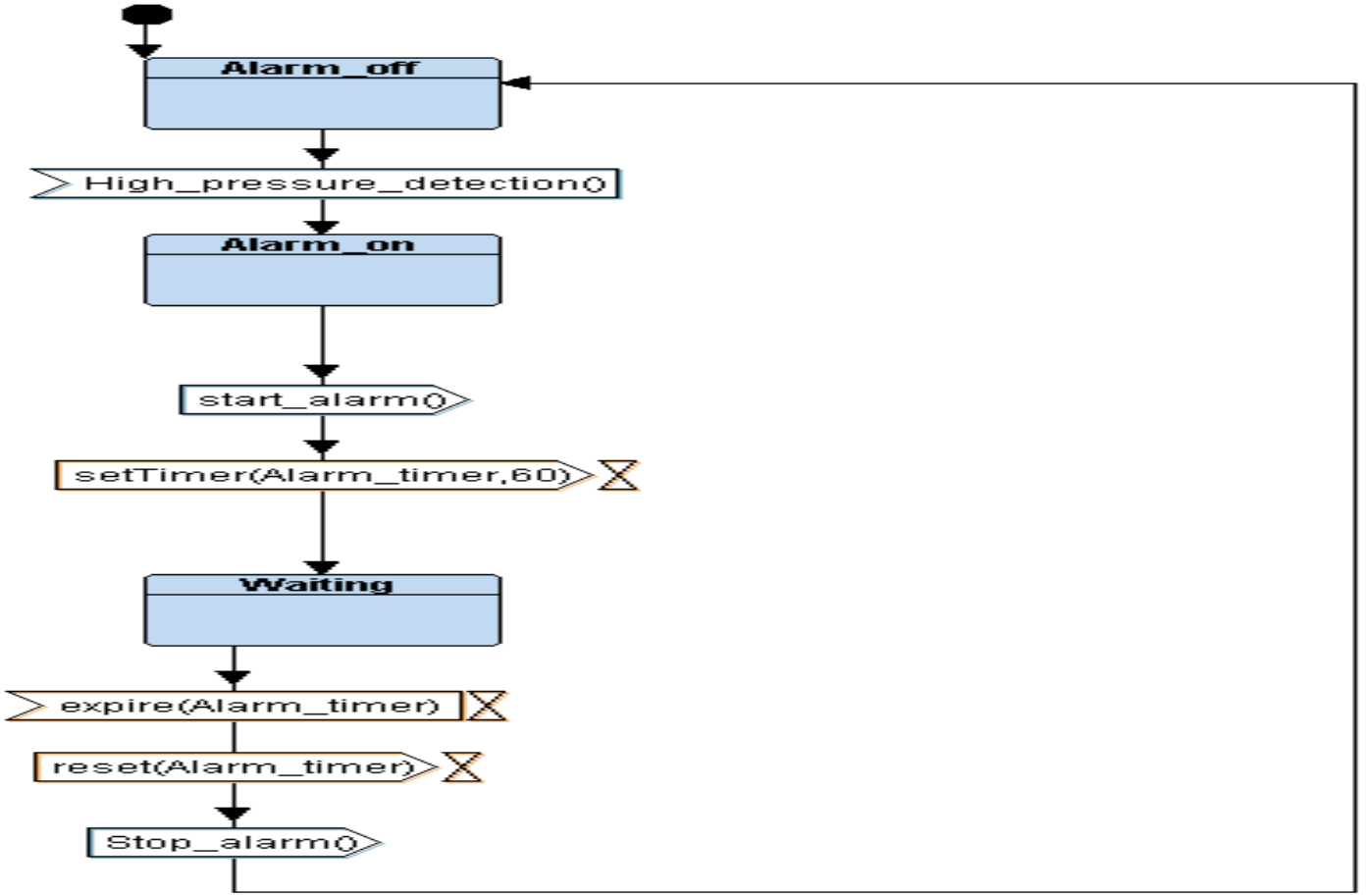
System design

C. State machine : Main controller



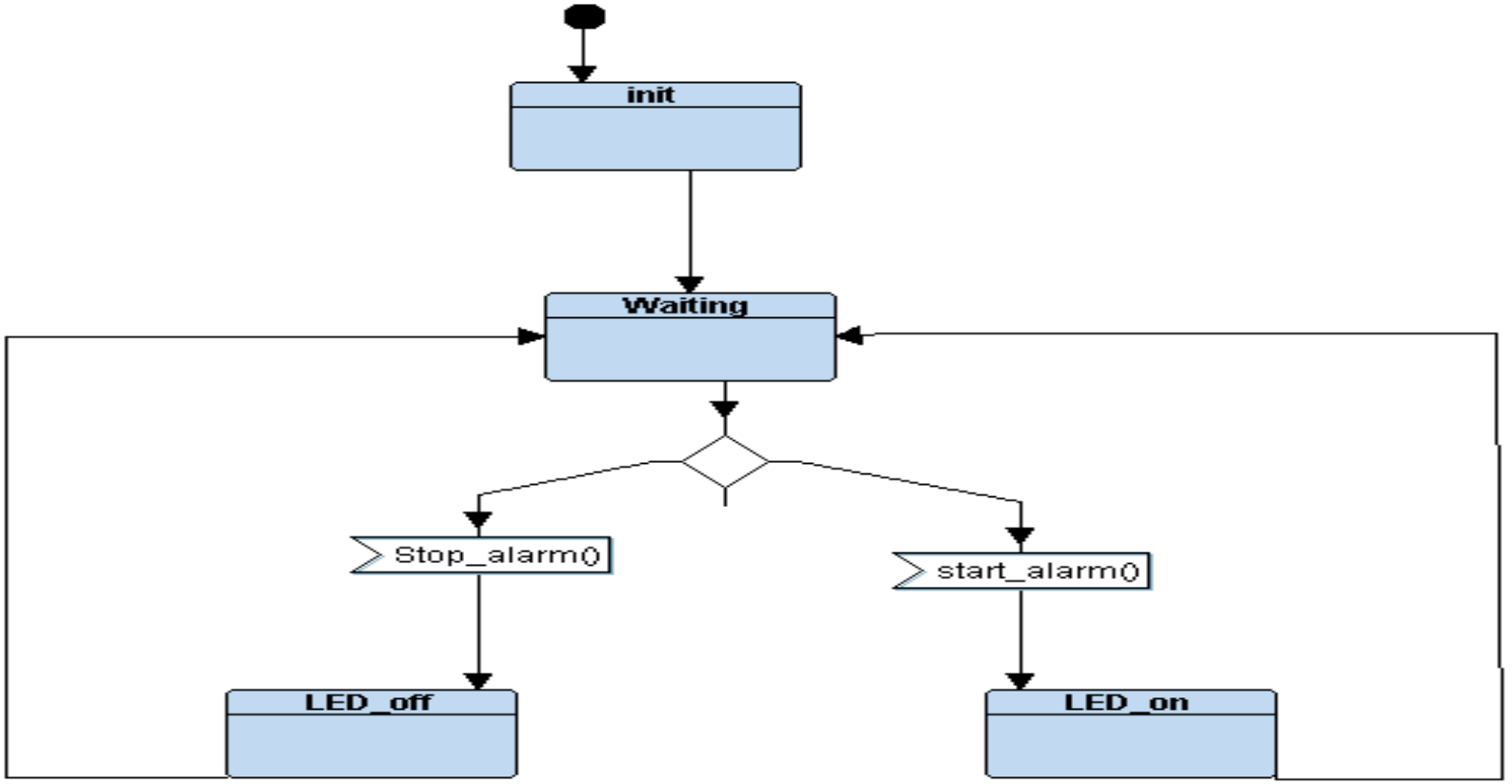
System design

D. State machine : Alarm monitor



System design

E. State machine : Alarm actuator driver



Sections & symbols for object files

A. Main file

B. Pressure Sensor driver file

C. Main controller file

D. Alarm monitor file

E. Alarm actuator driver file

F. Executable file

Sections & symbols for object files

A. Main file

```
Abotaleb@DESKTOP-RBI99BO MINGW32 /f/First_Term/Unit_5_Final_projects/First_pro
$ 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         0000007c  00000000  00000000  00000034  2**2
                  CONTENTS, ALLOC, LOAD, RELOC, READONLY, CODE
  1 .data         00000000  00000000  00000000  000000b0  2**0
                  CONTENTS, ALLOC, LOAD, DATA
  2 .bss          00000000  00000000  00000000  000000b0  2**0
                  ALLOC
  3 .debug_info   00000ab3  00000000  00000000  000000b0  2**0
                  CONTENTS, RELOC, READONLY, DEBUGGING
  4 .debug_abbrev 000001d6  00000000  00000000  00000b63  2**0
                  CONTENTS, READONLY, DEBUGGING
  5 .debug_loc    00000070  00000000  00000000  00000d39  2**0
                  CONTENTS, READONLY, DEBUGGING
  6 .debug_aranges 00000020  00000000  00000000  00000da9  2**0
                  CONTENTS, RELOC, READONLY, DEBUGGING
  7 .debug_line   00000166  00000000  00000000  00000dc9  2**0
                  CONTENTS, RELOC, READONLY, DEBUGGING
  8 .debug_str    0000060f  00000000  00000000  00000f2f  2**0
                  CONTENTS, READONLY, DEBUGGING
  9 .comment      0000007f  00000000  00000000  0000153e  2**0
                  CONTENTS, READONLY
 10 .debug_frame  0000004c  00000000  00000000  000015c0  2**2
                  CONTENTS, RELOC, READONLY, DEBUGGING
 11 .ARM.attributes 00000033  00000000  00000000  0000160c  2**0
                  CONTENTS, READONLY
```

```
$ arm-none-eabi-nm.exe main.o
         U Alarm_STATE
00000001 C Alarm_state_id
         U GPIO_INITIALIZATION
00000001 C M_state_id
00000044 T main
00000001 C Main_AL_state_id
         U MAL_STATE
         U Monitor_STATE
00000001 C PS_stete_id
         U PSensor_STATE
00000000 T setup
         U ST_Alarm_off
         U ST_LED_off
         U ST_M_Waiting
         U ST_Pressure_Sensor_Init
```

Sections & symbols for object files

B. Pressure Sensor driver file

```
Abotaleb@DESKTOP-RBI99BO MINGW32 /f/First_Term/Unit_3_Final_projects/First_project/Co
$ arm-none-eabi-objdump.exe -h Sensor_pressure.o

Sensor_pressure.o:     file format elf32-littlearm

Sections:
Idx Name          Size      VMA       LMA       File off  Algn
  0 .text         00000098  00000000  00000000  00000034  2**2
                  CONTENTS, ALLOC, LOAD, RELOC, READONLY, CODE
  1 .data         00000000  00000000  00000000  000000cc  2**0
                  CONTENTS, ALLOC, LOAD, DATA
  2 .bss          00000004  00000000  00000000  000000cc  2**2
                  ALLOC
  3 .debug_info   00000a67  00000000  00000000  000000cc  2**0
                  CONTENTS, RELOC, READONLY, DEBUGGING
  4 .debug_abbrev 000001f9  00000000  00000000  00000b33  2**0
                  CONTENTS, READONLY, DEBUGGING
  5 .debug_loc    000000e0  00000000  00000000  00000d2c  2**0
                  CONTENTS, READONLY, DEBUGGING
  6 .debug_aranges 00000020  00000000  00000000  00000e0c  2**0
                  CONTENTS, RELOC, READONLY, DEBUGGING
  7 .debug_line   0000018e  00000000  00000000  00000e2c  2**0
                  CONTENTS, RELOC, READONLY, DEBUGGING
  8 .debug_str    000005d6  00000000  00000000  00000fba  2**0
                  CONTENTS, READONLY, DEBUGGING
  9 .comment      0000007f  00000000  00000000  00001590  2**0
                  CONTENTS, READONLY
 10 .debug_frame  00000088  00000000  00000000  00001610  2**2
                  CONTENTS, RELOC, READONLY, DEBUGGING
 11 .ARM.attributes 00000033  00000000  00000000  00001698  2**0
                  CONTENTS, READONLY
```

```
Abotaleb@DESKTOP-RBI99BO MINGW32 /f/First_Term
$ arm-none-eabi-nm.exe Sensor_pressure.o
00000004 C counter_delay
         U Delay
00000080 T getPressureVal
00000000 B PS_Pressure
00000001 C PS_stete_id
00000004 C PSensor_STATE
         U Set_Sensor_pressure
00000000 T ST_Pressure_Sensor_Init
0000001c T ST_PS_Reading
00000058 T ST_PS_Waiting
```

Sections & symbols for object files

C. Main controller file

```
Abotaleb@DESKTOP-RBI99BO MINGW32 /f/First_Term/Unit_5_Final_projects/First_project
$ arm-none-eabi-objdump.exe -h Main_AL.o

Main_AL.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         00000004  00000000  00000000  000000f0  2**2
                  CONTENTS, ALLOC, LOAD, DATA
  2 .bss          00000000  00000000  00000000  000000f4  2**0
                  ALLOC
  3 .debug_info   00000a8b  00000000  00000000  000000f4  2**0
                  CONTENTS, RELOC, READONLY, DEBUGGING
  4 .debug_abbrev 00000208  00000000  00000000  00000b7f  2**0
                  CONTENTS, READONLY, DEBUGGING
  5 .debug_loc    00000110  00000000  00000000  00000d87  2**0
                  CONTENTS, READONLY, DEBUGGING
  6 .debug_aranges 00000020  00000000  00000000  00000e97  2**0
                  CONTENTS, RELOC, READONLY, DEBUGGING
  7 .debug_line   00000181  00000000  00000000  00000eb7  2**0
                  CONTENTS, RELOC, READONLY, DEBUGGING
  8 .debug_str    000005ed  00000000  00000000  00001038  2**0
                  CONTENTS, READONLY, DEBUGGING
  9 .comment      0000007f  00000000  00000000  00001625  2**0
                  CONTENTS, READONLY
 10 .debug_frame  00000094  00000000  00000000  000016a4  2**2
                  CONTENTS, RELOC, READONLY, DEBUGGING
 11 .ARM.attributes 00000033  00000000  00000000  00001738  2**0
                  CONTENTS, READONLY
```

```
Abotaleb@DESKTOP-RBI99BO MINGW32 /f/First_Term/Unit_
$ arm-none-eabi-nm.exe Main_AL.o
00000001 C Main_AL_state_id
00000004 C MAL_clock
00000004 C MAL_Pressure
00000000 D MAL_Pressure_thresold
00000004 C MAL_STATE
         U Set_monitor_High_Pressure_Detection
00000000 T Set_Sensor_pressure
0000002c T ST_M_Checking
00000074 T ST_M_Sending
00000098 T ST_M_Waiting
```

Sections & symbols for object files

D. Alarm monitor file

```
Abotaleb@DESKTOP-RBI99BO MINGW32 /f/First_Term/Unit_5_Final_projects/First_project/Cooc
$ arm-none-eabi-objdump.exe -h Monitor.o

Monitor.o:     file format elf32-littlearm

Sections:
Idx Name          Size      VMA       LMA       File off  Algn
  0 .text         00000080  00000000  00000000  00000034  2**2
                  CONTENTS, ALLOC, LOAD, RELOC, READONLY, CODE
  1 .data         00000000  00000000  00000000  000000b4  2**0
                  CONTENTS, ALLOC, LOAD, DATA
  2 .bss          00000000  00000000  00000000  000000b4  2**0
                  ALLOC
  3 .debug_info   00000a57  00000000  00000000  000000b4  2**0
                  CONTENTS, RELOC, READONLY, DEBUGGING
  4 .debug_abbrev 000001e1  00000000  00000000  00000b0b  2**0
                  CONTENTS, READONLY, DEBUGGING
  5 .debug_loc    000000c8  00000000  00000000  00000cec  2**0
                  CONTENTS, READONLY, DEBUGGING
  6 .debug_aranges 00000020  00000000  00000000  00000db4  2**0
                  CONTENTS, RELOC, READONLY, DEBUGGING
  7 .debug_line   0000017e  00000000  00000000  00000dd4  2**0
                  CONTENTS, RELOC, READONLY, DEBUGGING
  8 .debug_str    000005d7  00000000  00000000  00000f52  2**0
                  CONTENTS, READONLY, DEBUGGING
  9 .comment      0000007f  00000000  00000000  00001529  2**0
                  CONTENTS, READONLY
 10 .debug_frame  00000084  00000000  00000000  000015a8  2**2
                  CONTENTS, RELOC, READONLY, DEBUGGING
 11 .ARM.attributes 00000033  00000000  00000000  0000162c  2**0
                  CONTENTS, READONLY
```

```
Abotaleb@DESKTOP-RBI99BO MINGW32 /f/First_Term/Unit
$ arm-none-eabi-nm.exe Monitor.o
         U Delay
00000004 C M_Alarm_Value
00000001 C M_state_id
00000004 C Monitor_STATE
00000000 T Set_monitor_High_Pressure_Detection
00000040 T ST_Alarm_off
0000001c T ST_Alarm_on
00000058 T ST_Alarm_Waiting
         U Start_Alarm
         U Stop_Alarm
```

Sections & symbols for object files

E. Alarm actuator driver file

```
Abotaleb@DESKTOP-RBI99BO MINGW32 /f/First_Term/Unit_5_Final_projects/First_pro
$ arm-none-eabi-objdump.exe -h Alarm.o

Alarm.o:     file format elf32-littlearm

Sections:
Idx Name          Size      VMA       LMA       File off  Algn
  0 .text         000000d4  00000000  00000000  00000034  2**2
                  CONTENTS, ALLOC, LOAD, RELOC, READONLY, CODE
  1 .data         00000000  00000000  00000000  00000108  2**0
                  CONTENTS, ALLOC, LOAD, DATA
  2 .bss          00000000  00000000  00000000  00000108  2**0
                  ALLOC
  3 .debug_info   00000a81  00000000  00000000  00000108  2**0
                  CONTENTS, RELOC, READONLY, DEBUGGING
  4 .debug_abbrev 000001e1  00000000  00000000  00000b89  2**0
                  CONTENTS, READONLY, DEBUGGING
  5 .debug_loc    00000150  00000000  00000000  00000d6a  2**0
                  CONTENTS, READONLY, DEBUGGING
  6 .debug_aranges 00000020  00000000  00000000  00000eba  2**0
                  CONTENTS, RELOC, READONLY, DEBUGGING
  7 .debug_line   00000181  00000000  00000000  00000eda  2**0
                  CONTENTS, RELOC, READONLY, DEBUGGING
  8 .debug_str    000005c6  00000000  00000000  0000105b  2**0
                  CONTENTS, READONLY, DEBUGGING
  9 .comment      0000007f  00000000  00000000  00001621  2**0
                  CONTENTS, READONLY
 10 .debug_frame  000000c4  00000000  00000000  000016a0  2**2
                  CONTENTS, RELOC, READONLY, DEBUGGING
 11 .ARM.attributes 00000033  00000000  00000000  00001764  2**0
                  CONTENTS, READONLY
```

```
Abotaleb@DESKTOP-RBI99BO MINGW32 /f/
$ arm-none-eabi-nm.exe Alarm.o
00000004 C Alarm_STATE
00000001 C Alarm_state_id
         U Delay
00000004 C LED_Check
         U Set_Alarm_actuator
00000000 T ST_LED_Init
00000084 T ST_LED_off
00000054 T ST_LED_on
000000b4 T ST_LED_Waiting
0000000c T Start_Alarm
00000030 T Stop_Alarm
```

Sections & symbols for object files

F. Final file

```
Abotaleb@DESKTOP-RBI99BO MINGW32 /f/First_Term/Unit_5_Final_projects/First_project/
$ arm-none-eabi-objdump.exe -h High_Pressure_Detection.elf

High_Pressure_Detection.elf:     file format elf32-littlearm

Sections:
Idx Name          Size      VMA       LMA       File off  Algn
  0 .text         0000047c  08000000  08000000  00010000  2**2
                  CONTENTS, ALLOC, LOAD, READONLY, CODE
  1 .data         00000004  20000000  0800047c  00020000  2**2
                  CONTENTS, ALLOC, LOAD, DATA
  2 .bss          0000102c  20000004  08000480  00020004  2**2
                  ALLOC
  3 .debug_info   000048c2  00000000  00000000  00020004  2**0
                  CONTENTS, READONLY, DEBUGGING
  4 .debug_abbrev 00000d2f  00000000  00000000  000248c6  2**0
                  CONTENTS, READONLY, DEBUGGING
  5 .debug_loc    000005f0  00000000  00000000  000255f5  2**0
                  CONTENTS, READONLY, DEBUGGING
  6 .debug_aranges 000000e0  00000000  00000000  00025be5  2**0
                  CONTENTS, READONLY, DEBUGGING
  7 .debug_line   000009b5  00000000  00000000  00025cc5  2**0
                  CONTENTS, READONLY, DEBUGGING
  8 .debug_str    000007a9  00000000  00000000  0002667a  2**0
                  CONTENTS, READONLY, DEBUGGING
  9 .comment      0000007e  00000000  00000000  00026e23  2**0
                  CONTENTS, READONLY
 10 .ARM.attributes 00000033  00000000  00000000  00026ea1  2**0
                  CONTENTS, READONLY
 11 .debug_frame  00000380  00000000  00000000  00026ed4  2**2
                  CONTENTS, READONLY, DEBUGGING
```

Sections & symbols for object files

F. Final file con.

```
Abotaleb@DESKTOP-RBI99BO MINGW32 /f/First_Term/Unit_5_Final_projects/First_proje
$ arm-none-eabi-nm.exe High_Pressure_Detection.elf
20000008 B _E_bss
20000004 D _E_DATA
0800047c T _E_text
20000004 B _S_bss
20000000 D _S_DATA
20001008 B _stack_top
20001008 B Alarm_STATE
20001010 B Alarm_state_id
080003ec W Bus_fault
2000102c B counter_delay
080003ec T Default_Handler
080000f0 T Delay
080003d4 T getPressureVal
0800014c T GPIO_INITIALIZATION
080003ec W H_fault_Handler
2000100c B LED_Check
20001020 B M_Alarm_Value
20001013 B M_state_id
080001e0 T main
20001012 B Main_AL_state_id
20001018 B MAL_clock
2000101c B MAL_Pressure
20000000 D MAL_Pressure_thresold
20001014 B MAL_STATE
080003ec W MM_fault_Handler
20001024 B Monitor_STATE
080003ec W NMI_Handler
20000004 B PS_Pressure
20001011 B PS_stete_id
20001028 B PSensor_STATE
080003f8 T Reset_Handler
08000110 T Set_Alarm_actuator
080002d4 T Set_monitor_High_Pressure_Detection
08000218 T Set_Sensor_pressure
0800019c T setup
08000314 T ST_Alarm_off
080002f0 T ST_Alarm_on
0800032c T ST_Alarm_Waiting
0800001c T ST_LED_Init
080000a0 T ST_LED_off
08000070 T ST_LED_on
080000d0 T ST_LED_Waiting
08000244 T ST_M_Checking
0800028c T ST_M_Sending
080002b0 T ST_M_Waiting
08000354 T ST_Pressure_Sensor_Init
08000370 T ST_PS_Reading
080003ac T ST_PS_Waiting
08000028 T Start_Alarm
0800004c T Stop_Alarm
080003ec W Usage_fault_Handler
08000000 T Vectors
```

Startup .c

```
2   #include<stdio.h>
3
4
5   extern unsigned int _stack_top;
6   extern int main(void);
7   void Reset_Handler(void);
8   void Default_Handler()
9   {
10      Reset_Handler();
11  }
12
13  void NMI_Handler()__attribute__((weak,alias ("Default_Handler")));;
14  void H_fault_Handler()__attribute__((weak,alias ("Default_Handler")));;
15  void MM_fault_Handler()__attribute__((weak,alias ("Default_Handler")));;
16  void Bus_fault()__attribute__((weak,alias ("Default_Handler")));;
17  void Usage_fault_Handler()__attribute__((weak,alias ("Default_Handler")));;
18
19  unsigned int Vectors[] __attribute__((section(".Vectors")))={
20  (unsigned int)          &_stack_top,
21  (unsigned int)          &Reset_Handler,
22  (unsigned int)          &NMI_Handler,
23  (unsigned int)          &H_fault_Handler,
24  (unsigned int)          &MM_fault_Handler,
25  (unsigned int)          &Bus_fault,
26  (unsigned int)          &Usage_fault_Handler
27  };
28  extern unsigned int  _E_text;
29  extern unsigned int  _S_DATA;
30  extern unsigned int  _E_DATA;
31  extern unsigned int  _S_bss;
32  extern unsigned int  _E_bss;
33
```

Startup .c con.

```
void Reset_Handler(void)
{
    // copy data from ROM to RAM
    unsigned int  DATA_size = (unsigned char *)&_E_DATA - (unsigned char *)&_S_DATA;
    unsigned char  * p_src=(unsigned char *)&_E_text;
    unsigned char * p_dst=(unsigned char *)&_S_DATA;
    unsigned int  i;
    for( i=0;i<DATA_size;i++)
    {
        *((unsigned char *)p_dst++)=*((unsigned char *)p_src++);
    }

    // init the bss sectoin with zero

    unsigned int bss_size = (unsigned char *)&_E_bss - (unsigned char *)&_S_bss;
    p_dst=(unsigned char *)&_S_bss;

    for(i=0;i<bss_size;i++)
    {
        *((unsigned char *)p_dst++)=(unsigned char)0;
    }

    // jump to main

    main();
}
```

Linker script

```
2
3   MEMORY
4   {
5       flash(RX) : ORIGIN = 0x08000000, LENGTH = 128K
6       sram(RWX) : ORIGIN = 0x20000000, LENGTH = 20K
7   }
8
9   SECTIONS
10  {
11      .text :{
12          *(.Vectors*)
13          *(.text*)
14          *(.rodata*)
15          _E_text = . ;
16      }>flash
17      .data :{
18          _S_DATA = . ;
19          *(.data*)
20          . = ALIGN(4);
21          _E_DATA = . ;
22      }> sram AT> flash
23      .bss :{
24          _S_bss = . ;
25          *(.bss*)
26          . = ALIGN(4);
27          _E_bss = . ;
28          . = . + 0x1000;
29          _stack_top = . ;
30      }>sram
31
32  }
```

Simulation run code

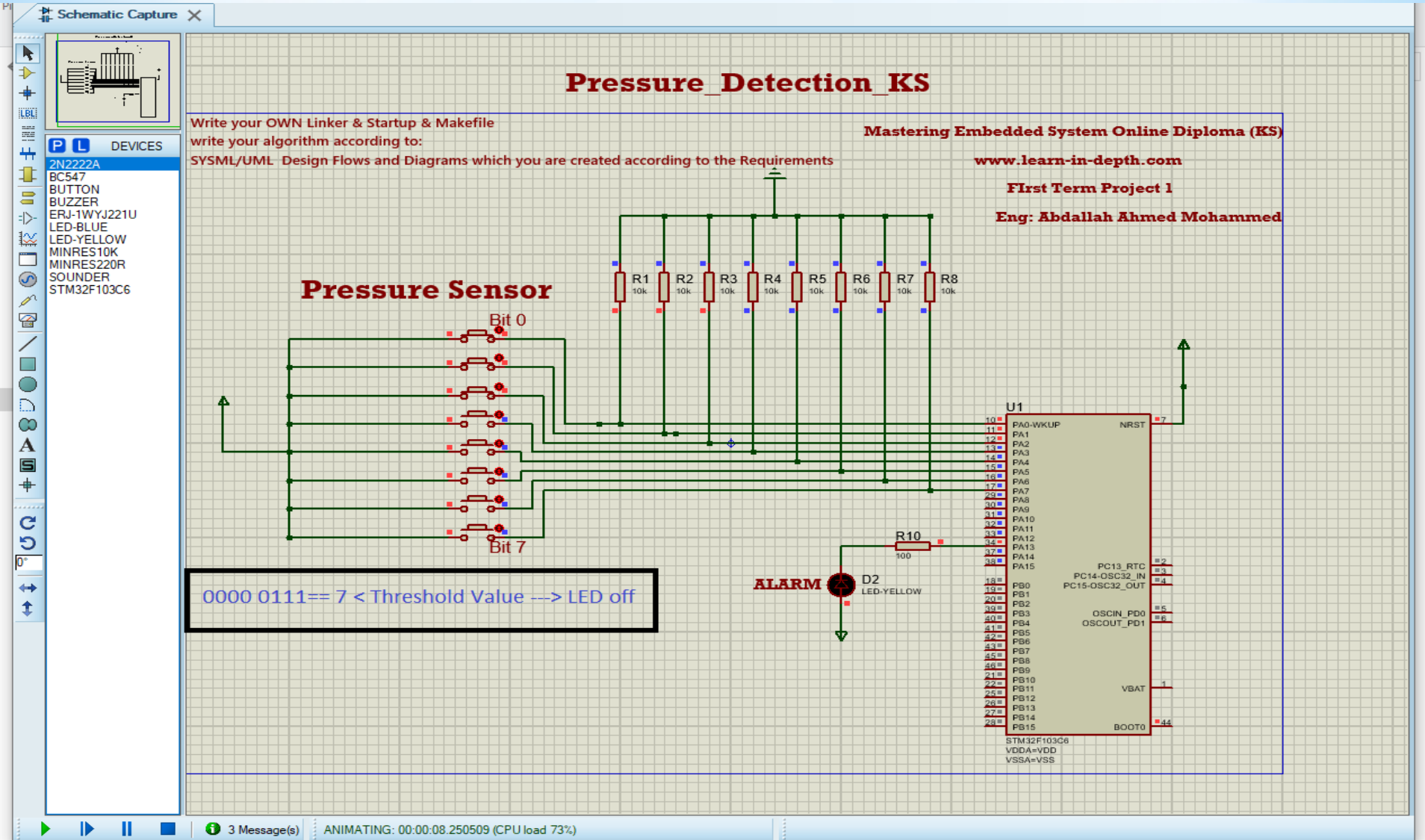
A. When the pressure sensor read < threshold

B. When the pressure sensor read = threshold

C. When the pressure sensor read > threshold

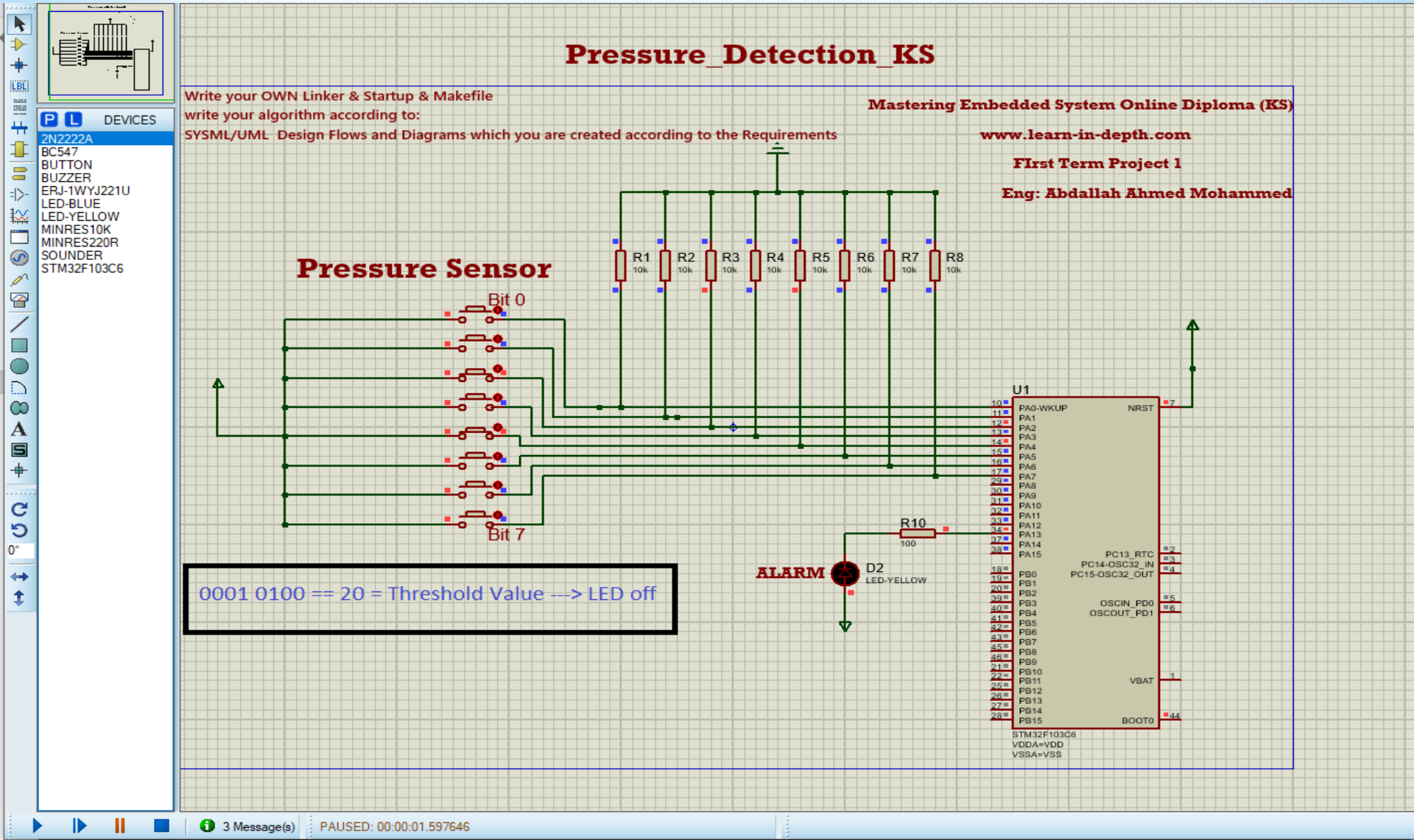
Simulation on proteus

A. When the pressure sensor read < threshold



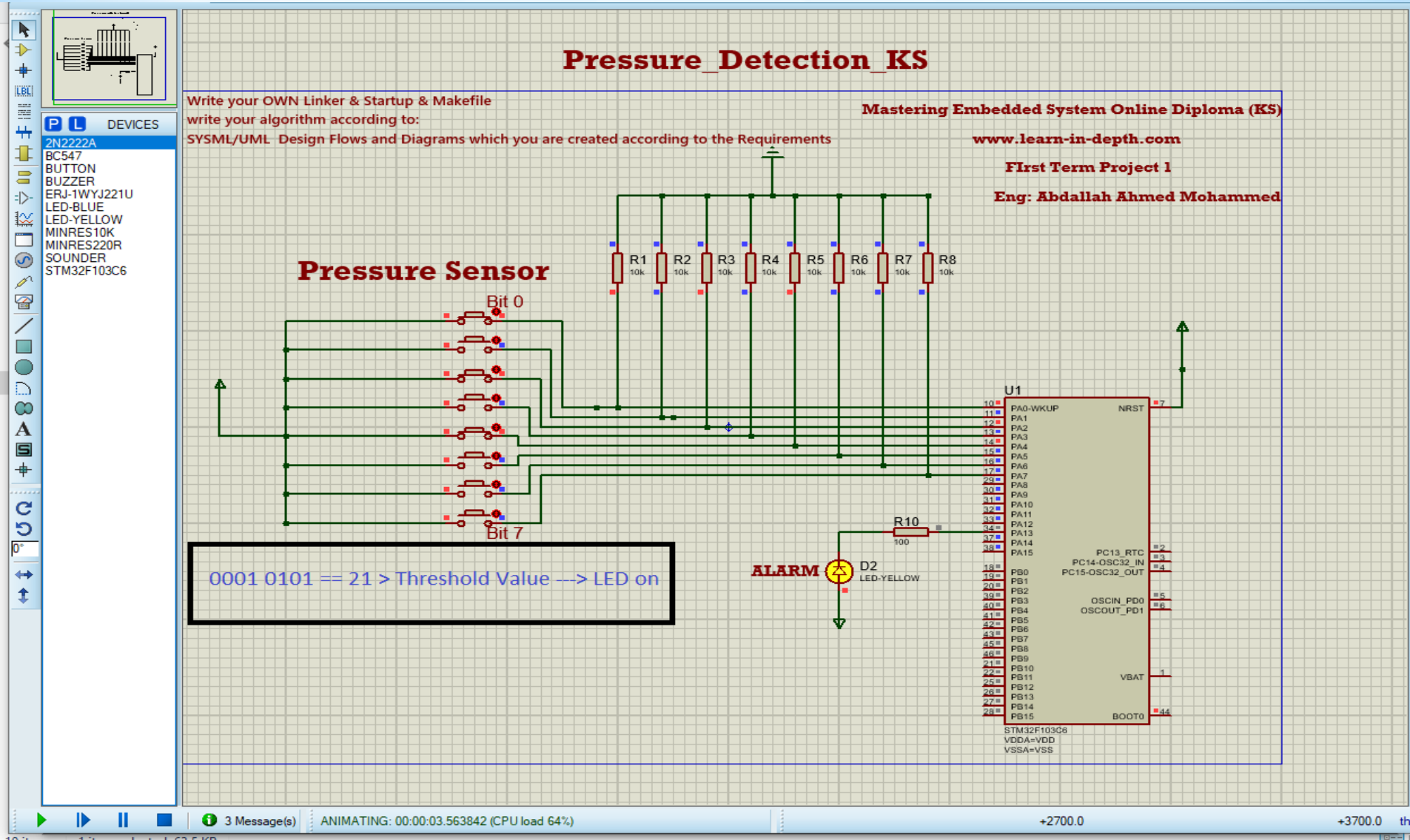
Simulation on proteus

B. When the pressure sensor read = threshold



Simulation on proteus

C. When the pressure sensor read > threshold



Embedded System online diploma

learn-in-depth

Be Professional In Embedded System

Eng. Keroles Shenouda

Eng. Abdallah Ahmed Mohammed

email: abdallahahmed17120@gmail.com

LinkedIn Profile

Abdallah progress