



---

# PRESSURE CONTROL SYSTEM

---



**PROJECT 1**

MADE BY:

YEHIA HASAN YEHIA

## Contents

1.0 CASE STUDY .....	3
2.0 METHOD.....	3
3.0 REQUIREMENTS.....	3
3.1 Agreed-upon Requirements .....	3
3.2 Requirement Diagram.....	4
4.0 SYSTEM ANALYSIS.....	5
4.1 Use Case Diagram.....	5
4.2 Activity Diagram.....	6
4.3 Sequence Diagram .....	7
5.0 SYSTEM DESIGN .....	8
5.1 Block Diagram .....	8
5.2 State Diagrams .....	9
5.2.1 Pressure Sensor .....	9
5.2.2 Pressure Controller.....	10
5.2.3 Alarm Controller .....	11
5.2.4 Alarm Actuator.....	12
6.0 INTERACTIVE SIMULATION.....	12
7.0 PROTOTYPE RUN .....	14
8.0 CODE SECTIONS.....	15
8.1 PController.o .....	15
8.2 PSensorDriver.o .....	15
8.3 AlarmController.o .....	16
8.4 driver.o .....	16
8.5 startup.o.....	17
8.6 main.o.....	17
8.7 The elf File .....	18
9.0 Symbols.....	19
9.1 PController.o .....	19

9.2 PSensorDriver.o .....	19
9.3 AlarmController.o .....	19
9.4 driver.o .....	20
9.5 startup.o.....	20
9.6 main.o.....	20
9.7 The elf File .....	21
10.0 SIMULATION .....	22
10.1 Initialization .....	22
10.2 Pressure = 10 bars .....	23
10.3 Pressure = 20 bars .....	24
10.4 Pressure = 28 bars .....	25

## 1.0 CASE STUDY

It is required to implement a pressure control system in a plane that should start an alarm for one minute if the pressure exceeds the threshold (20 bars)

## 2.0 METHOD

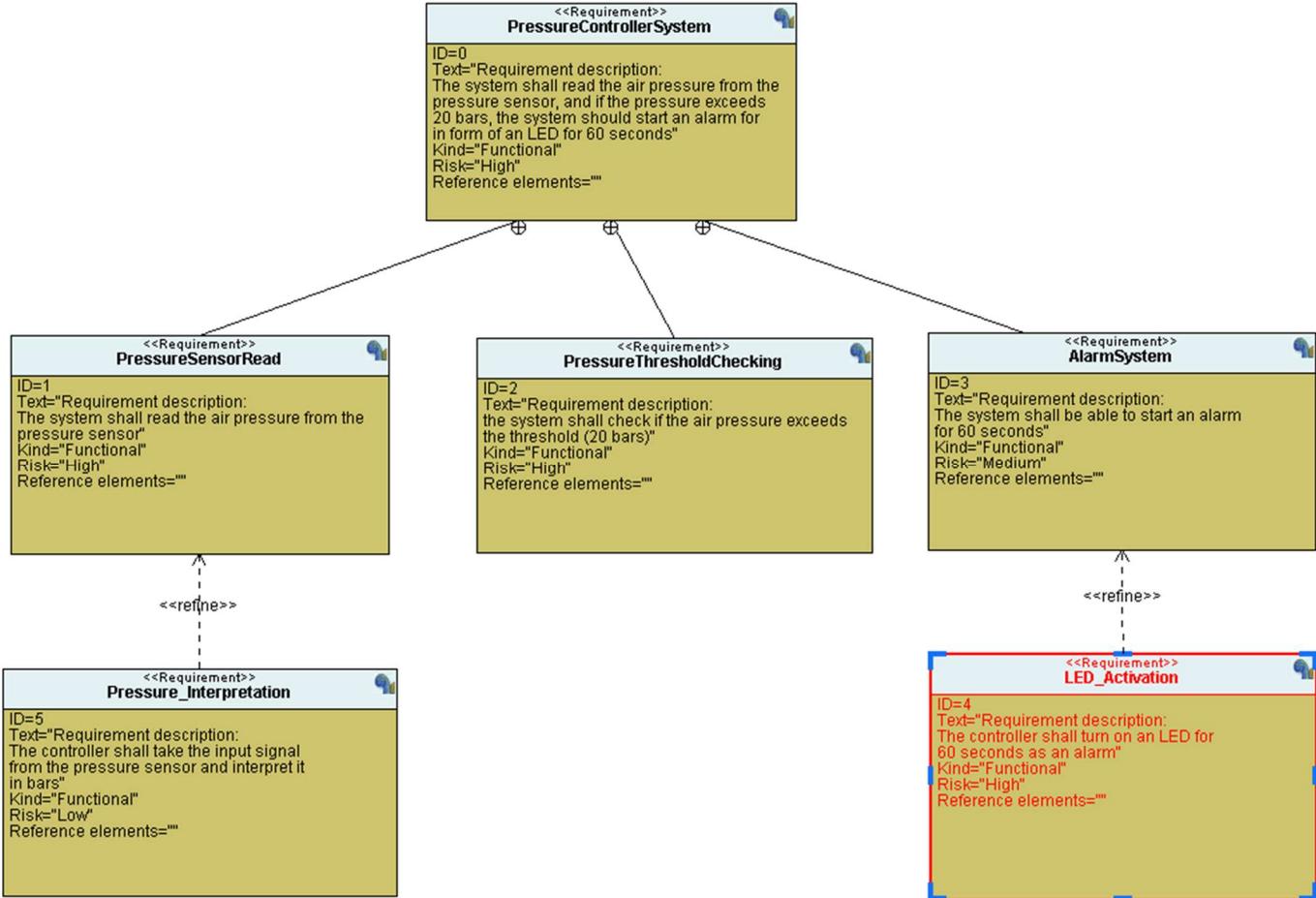
The waterfall model is the best method here, since the requirements are well defined.

## 3.0 REQUIREMENTS

### 3.1 Agreed-upon Requirements

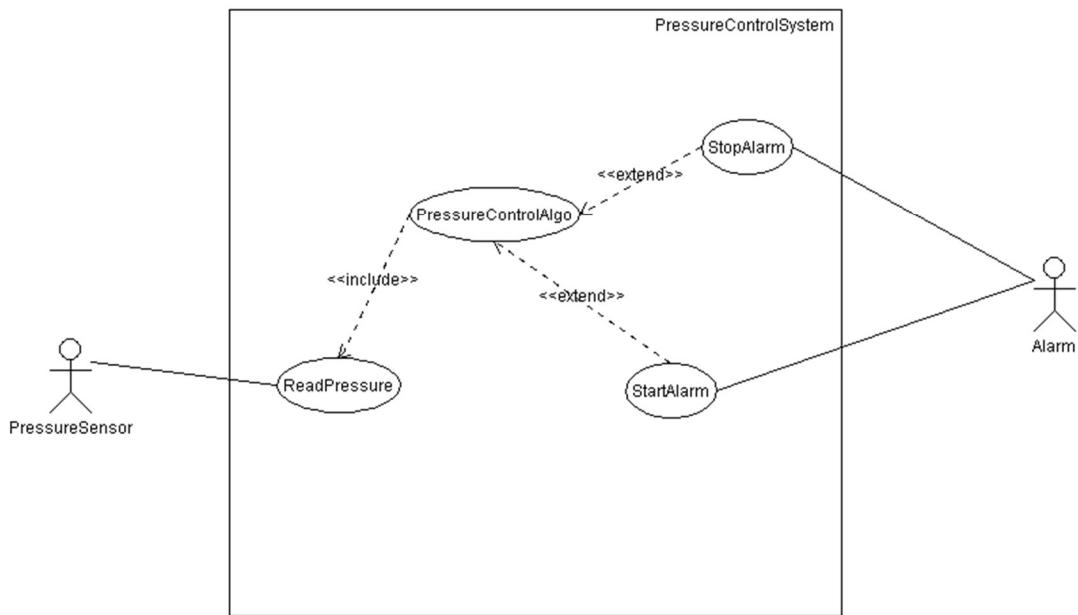
Requirement	State
Check if pressure reading exceeds threshold	To be Implemented
Start an Alarm System for 60 seconds if threshold exceeded	To be Implemented

## 3.2 Requirement Diagram

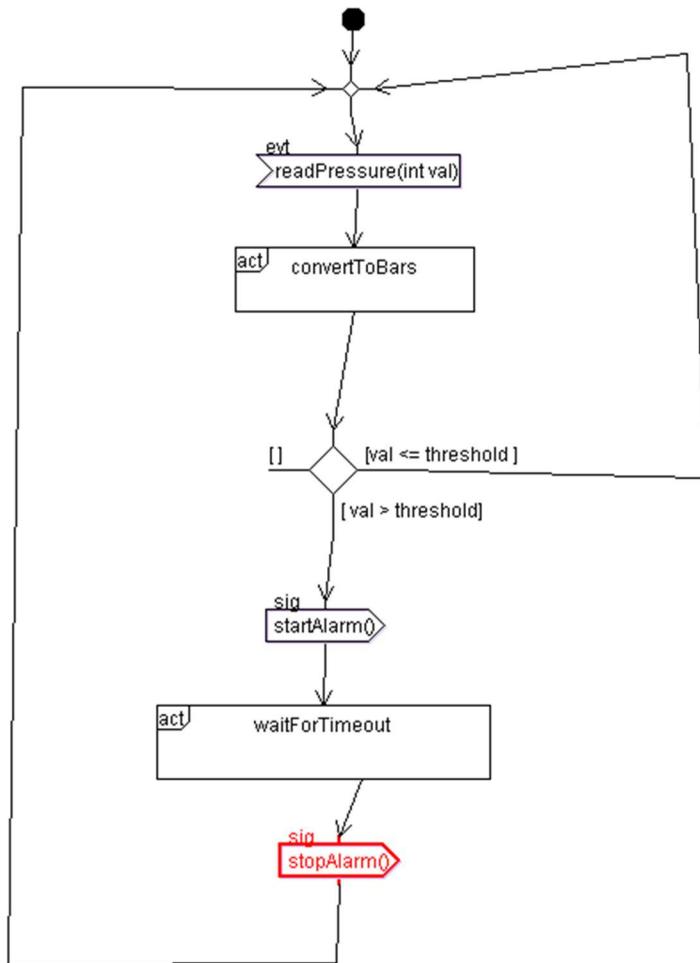


## 4.0 SYSTEM ANALYSIS

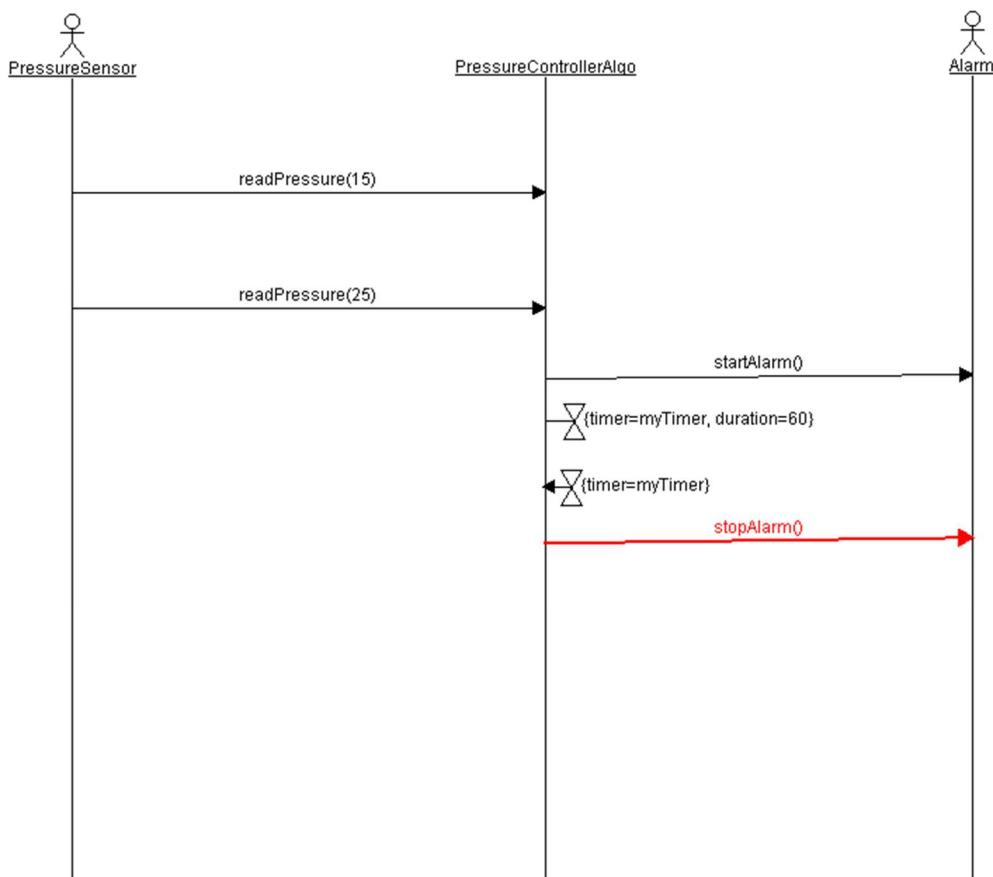
### 4.1 Use Case Diagram



## 4.2 Activity Diagram

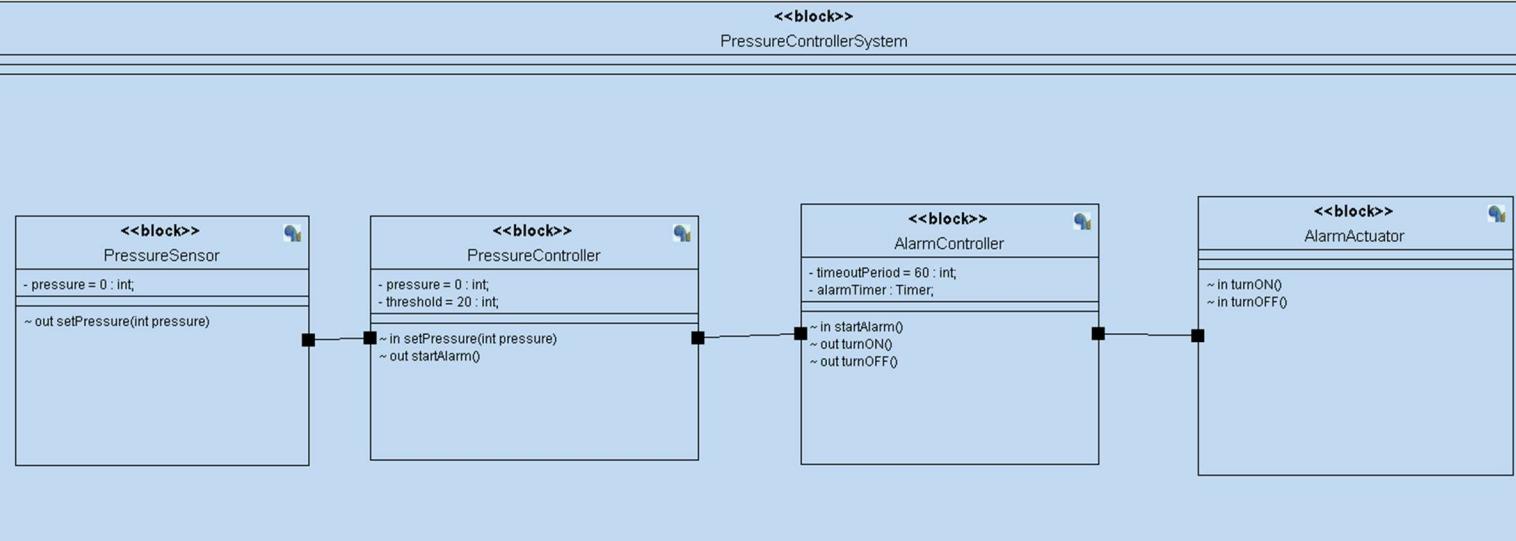


## 4.3 Sequence Diagram



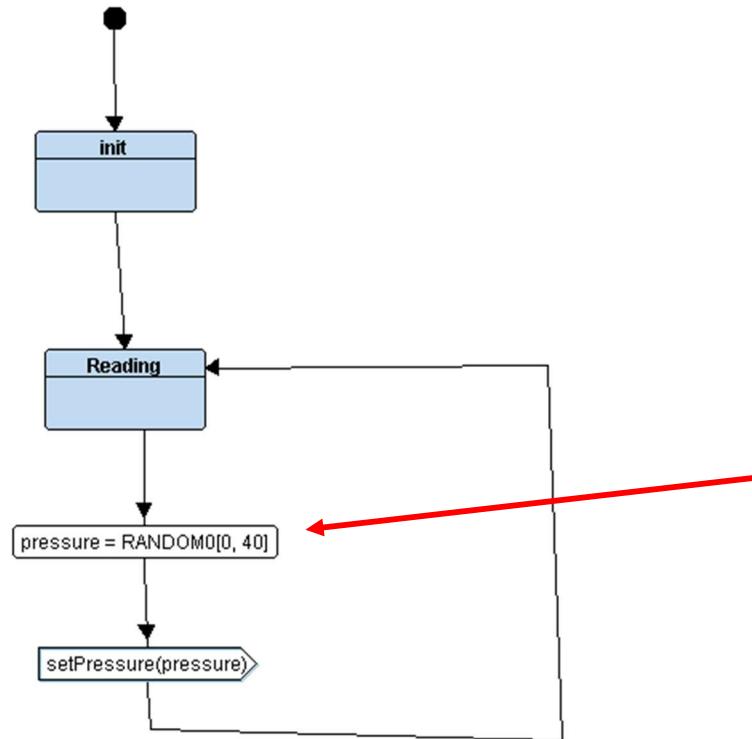
## 5.0 SYSTEM DESIGN

### 5.1 Block Diagram

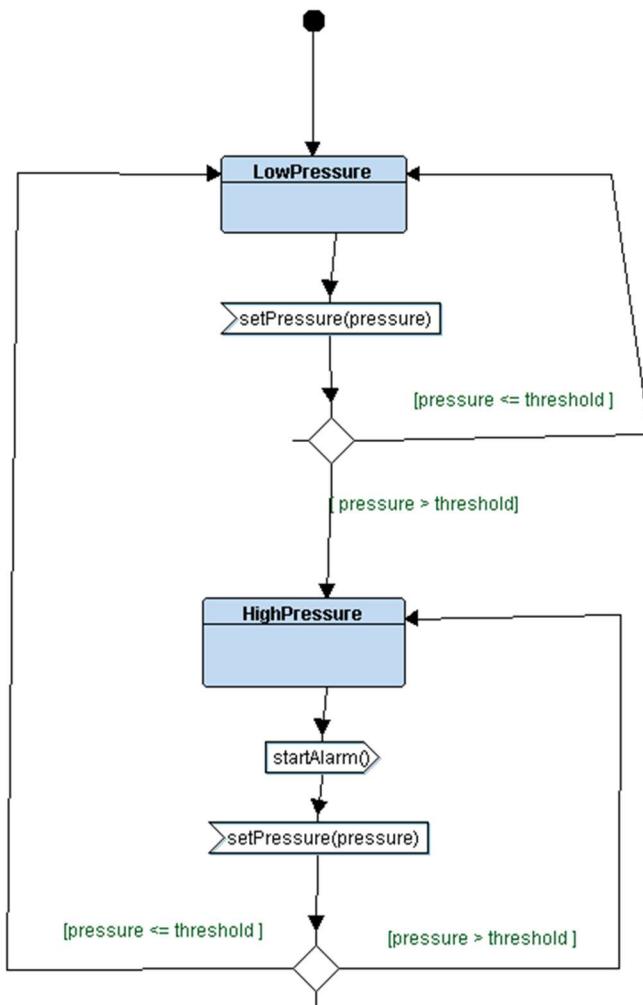


## 5.2 State Diagrams

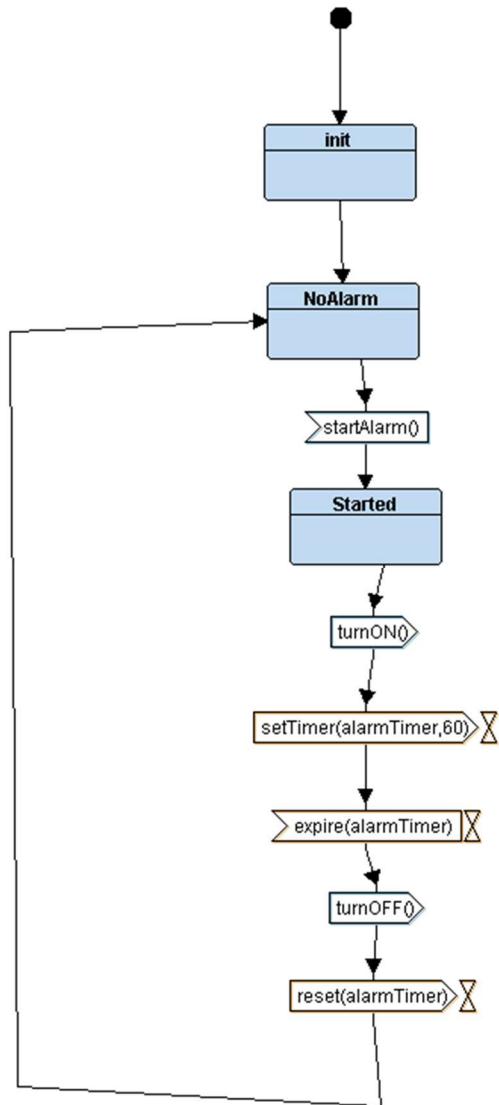
### 5.2.1 Pressure Sensor



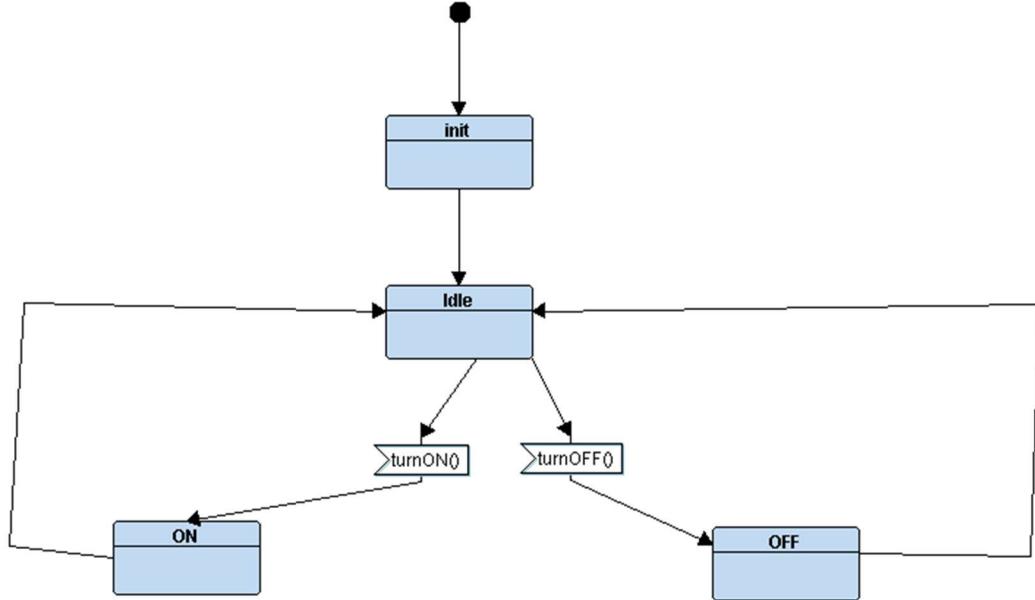
### 5.2.2 Pressure Controller



### 5.2.3 Alarm Controller

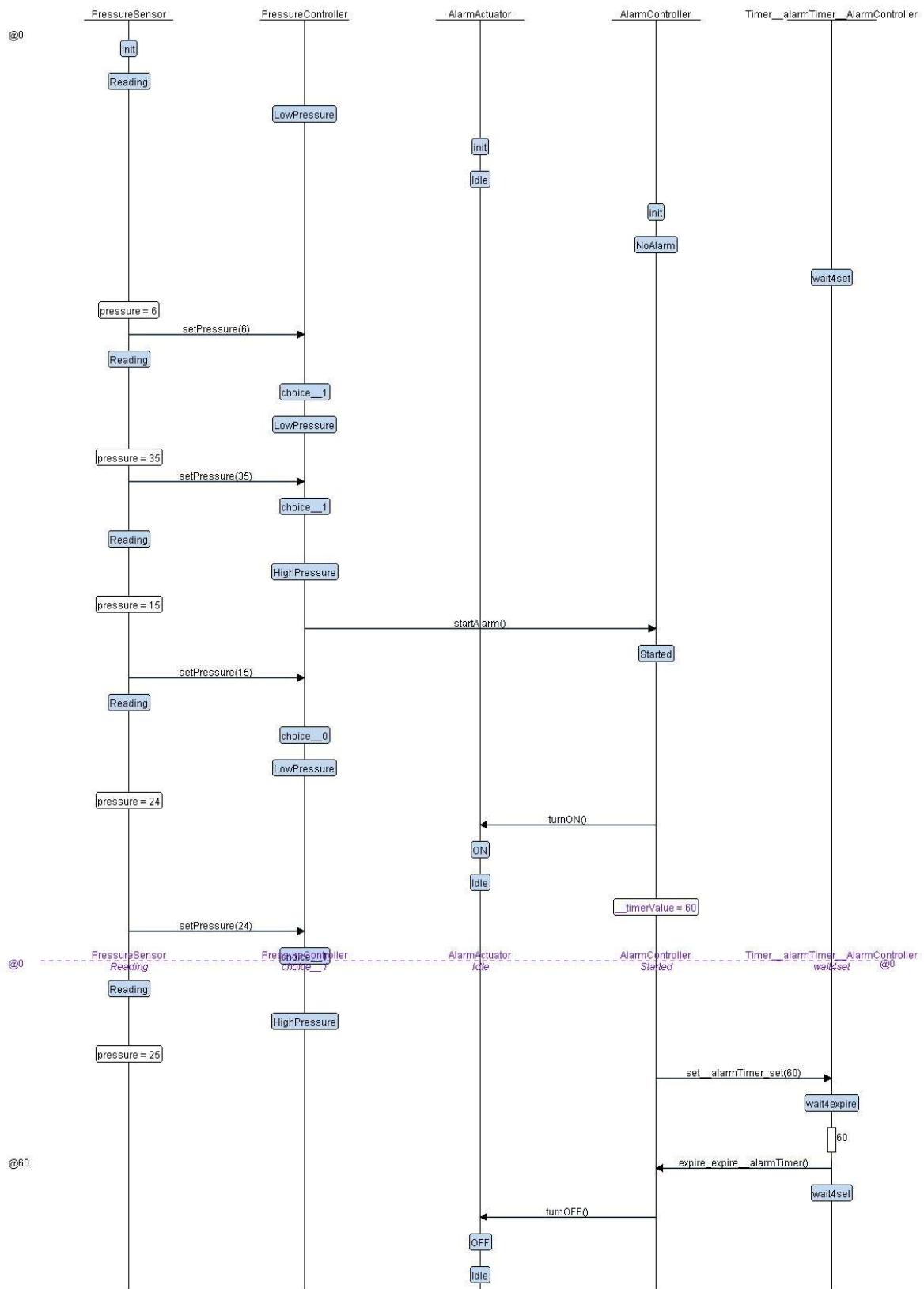


#### 5.2.4 Alarm Actuator



## 6.0 INTERACTIVE SIMULATION

Simulation image is too big to be on same page with the header. So, it is in the next page



## 7.0 PROTOTYPE RUN

```
$ Build/PCtrl.exe
ACTRL: Initialized
PController: Initialized
PSensor: initialized

PSensor: Pressure Reading = 35
PSensor ----> PCTRL : set pressure to 35
PCTRL: In High Pressure State, Pressure = 35
PCTRL ----> ACTRL : Alarm Started
ACTRL ----> ALARM STARTED

Press Enter to continue:

PSensor: Pressure Reading = 7
PSensor ----> PCTRL : set pressure to 7
PCTRL: In Low Pressure State, Pressure = 7
ACTRL ----> NO ALARM

Press Enter to continue:

PSensor: Pressure Reading = 17
PSensor ----> PCTRL : set pressure to 17
PCTRL: In Low Pressure State, Pressure = 17
ACTRL ----> NO ALARM

Press Enter to continue:

PSensor: Pressure Reading = 20
PSensor ----> PCTRL : set pressure to 20
PCTRL: In Low Pressure State, Pressure = 20
ACTRL ----> NO ALARM

Press Enter to continue:

PSensor: Pressure Reading = 33
PSensor ----> PCTRL : set pressure to 33
PCTRL: In High Pressure State, Pressure = 33
PCTRL ----> ACTRL : Alarm Started
ACTRL ----> ALARM STARTED
```

## 8.0 CODE SECTIONS

### 8.1 PController.o

```
Sections:
Idx Name      Size    VMA     LMA     File off  Align
 0 .text      00000088 00000000 00000000 00000034 2**2
              CONTENTS, ALLOC, LOAD, RELOC, READONLY, CODE
 1 .data      00000000 00000000 00000000 000000bc 2**0
              CONTENTS, ALLOC, LOAD, DATA
 2 .bss       00000004 00000000 00000000 000000bc 2**2
              ALLOC
 3 .debug_info 0000012c 00000000 00000000 000000bc 2**0
              CONTENTS, RELOC, READONLY, DEBUGGING
 4 .debug_abbrev 000000f6 00000000 00000000 000001e8 2**0
              CONTENTS, READONLY, DEBUGGING
 5 .debug_loc   00000110 00000000 00000000 000002de 2**0
              CONTENTS, READONLY, DEBUGGING
 6 .debug_aranges 00000020 00000000 00000000 000003ee 2**0
              CONTENTS, RELOC, READONLY, DEBUGGING
 7 .debug_line   00000081 00000000 00000000 0000040e 2**0
              CONTENTS, RELOC, READONLY, DEBUGGING
 8 .debug_str    0000020c 00000000 00000000 0000048f 2**0
              CONTENTS, READONLY, DEBUGGING
 9 .comment     0000007f 00000000 00000000 0000069b 2**0
              CONTENTS, READONLY
10 .debug_frame 00000094 00000000 00000000 0000071c 2**2
              CONTENTS, RELOC, READONLY, DEBUGGING
11 .ARM.attributes 00000033 00000000 00000000 000007b0 2**0
              CONTENTS, READONLY
```

### 8.2 PSensorDriver.o

```
Build/PSensorDriver.o:      file format elf32-littlearm

Sections:
Idx Name      Size    VMA     LMA     File off  Align
 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       00000004 00000000 00000000 0000008c 2**2
              ALLOC
 3 .debug_info 00000a08 00000000 00000000 0000008c 2**0
              CONTENTS, RELOC, READONLY, DEBUGGING
 4 .debug_abbrev 000001e8 00000000 00000000 00000a94 2**0
              CONTENTS, READONLY, DEBUGGING
 5 .debug_loc   00000070 00000000 00000000 00000c7c 2**0
              CONTENTS, READONLY, DEBUGGING
 6 .debug_aranges 00000020 00000000 00000000 00000cec 2**0
              CONTENTS, RELOC, READONLY, DEBUGGING
 7 .debug_line   00000159 00000000 00000000 00000d0c 2**0
              CONTENTS, RELOC, READONLY, DEBUGGING
 8 .debug_str    000005e0 00000000 00000000 00000e65 2**0
              CONTENTS, READONLY, DEBUGGING
 9 .comment     0000007f 00000000 00000000 00001445 2**0
              CONTENTS, READONLY
10 .debug_frame 0000004c 00000000 00000000 000014c4 2**2
              CONTENTS, RELOC, READONLY, DEBUGGING
11 .ARM.attributes 00000033 00000000 00000000 00001510 2**0
              CONTENTS, READONLY
```

## 8.3 AlarmController.o

```
Sections:
Idx Name      Size    VMA      LMA      File off  Algn
 0 .text      00000094  00000000  00000000  00000034  2**2
              CONTENTS, ALLOC, LOAD, RELOC, READONLY, CODE
 1 .data      00000000  00000000  00000000  000000c8  2**0
              CONTENTS, ALLOC, LOAD, DATA
 2 .bss       00000000  00000000  00000000  000000c8  2**0
              ALLOC
 3 .debug_info 000000a1e 00000000  00000000  000000c8  2**0
              CONTENTS, RELOC, READONLY, DEBUGGING
 4 .debug_abbrev 000001d9 00000000  00000000  00000ae6  2**0
              CONTENTS, READONLY, DEBUGGING
 5 .debug_loc   000000e0 00000000  00000000  00000cbf  2**0
              CONTENTS, READONLY, DEBUGGING
 6 .debug_aranges 00000020 00000000  00000000  00000d9f  2**0
              CONTENTS, RELOC, READONLY, DEBUGGING
 7 .debug_line   00000150 00000000  00000000  00000dbf  2**0
              CONTENTS, RELOC, READONLY, DEBUGGING
 8 .debug_str    0000061c 00000000  00000000  00000f0f  2**0
              CONTENTS, READONLY, DEBUGGING
 9 .comment     0000007f 00000000  00000000  0000152b  2**0
              CONTENTS, READONLY
10 .debug_frame 00000088 00000000  00000000  000015ac  2**2
              CONTENTS, RELOC, READONLY, DEBUGGING
11 .ARM.attributes 00000033 00000000  00000000  00001634  2**0
              CONTENTS, READONLY
```

## 8.4 driver.o

```
Sections:
Idx Name      Size    VMA      LMA      File off  Algn
 0 .text      000000c4  00000000  00000000  00000034  2**2
              CONTENTS, ALLOC, LOAD, READONLY, CODE
 1 .data      00000000  00000000  00000000  000000f8  2**0
              CONTENTS, ALLOC, LOAD, DATA
 2 .bss       00000000  00000000  00000000  000000f8  2**0
              ALLOC
 3 .debug_info 000000a05 00000000  00000000  000000f8  2**0
              CONTENTS, RELOC, READONLY, DEBUGGING
 4 .debug_abbrev 000001de 00000000  00000000  00000af0  2**0
              CONTENTS, READONLY, DEBUGGING
 5 .debug_loc   00000140 00000000  00000000  00000cd0  2**0
              CONTENTS, READONLY, DEBUGGING
 6 .debug_aranges 00000020 00000000  00000000  00000e1b  2**0
              CONTENTS, RELOC, READONLY, DEBUGGING
 7 .debug_line   000001c0 00000000  00000000  00000e3b  2**0
              CONTENTS, RELOC, READONLY, DEBUGGING
 8 .debug_str    000005bc 00000000  00000000  00000ffb  2**0
              CONTENTS, READONLY, DEBUGGING
 9 .comment     0000007f 00000000  00000000  000015b7  2**0
              CONTENTS, READONLY
10 .debug_frame 000000a0 00000000  00000000  00001638  2**2
              CONTENTS, RELOC, READONLY, DEBUGGING
11 .ARM.attributes 00000033 00000000  00000000  000016d8  2**0
              CONTENTS, READONLY
```

## 8.5 startup.o

```
Sections:
Idx Name      Size    VMA     LMA     File off  Align
 0 .text      00000090  00000000  00000000  00000034  2**2
              CONTENTS, ALLOC, LOAD, RELOC, READONLY, CODE
 1 .data      00000000  00000000  00000000  000000c4  2**0
              CONTENTS, ALLOC, LOAD, DATA
 2 .bss       00000000  00000000  00000000  000000c4  2**0
              ALLOC
 3 .vectors   0000001c  00000000  00000000  000000c4  2**2
              CONTENTS, ALLOC, LOAD, RELOC, DATA
 4 .debug_info 0000017b  00000000  00000000  000000e0  2**0
              CONTENTS, RELOC, READONLY, DEBUGGING
 5 .debug_abbrev 000000d8  00000000  00000000  0000025b  2**0
              CONTENTS, READONLY, DEBUGGING
 6 .debug_loc   00000094  00000000  00000000  00000333  2**0
              CONTENTS, READONLY, DEBUGGING
 7 .debug_aranges 00000020  00000000  00000000  000003c7  2**0
              CONTENTS, RELOC, READONLY, DEBUGGING
 8 .debug_line  0000007f  00000000  00000000  000003e7  2**0
              CONTENTS, RELOC, READONLY, DEBUGGING
 9 .debug_str   0000019f  00000000  00000000  00000466  2**0
              CONTENTS, READONLY, DEBUGGING
10 .comment    0000007f  00000000  00000000  00000605  2**0
              CONTENTS, READONLY
11 .debug_frame 00000054  00000000  00000000  00000684  2**2
              CONTENTS, RELOC, READONLY, DEBUGGING
12 .ARM.attributes 00000033  00000000  00000000  000006d8  2**0
              CONTENTS, READONLY
```

## 8.6 main.o

```
Build/main.o:      file format elf32-littlearm

Sections:
Idx Name      Size    VMA     LMA     File off  Align
 0 .text      00000040  00000000  00000000  00000034  2**2
              CONTENTS, ALLOC, LOAD, RELOC, READONLY, CODE
 1 .data      00000000  00000000  00000000  00000074  2**0
              CONTENTS, ALLOC, LOAD, DATA
 2 .bss       00000000  00000000  00000000  00000074  2**0
              ALLOC
 3 .debug_info 000009cb  00000000  00000000  00000074  2**0
              CONTENTS, RELOC, READONLY, DEBUGGING
 4 .debug_abbrev 000001a1  00000000  00000000  00000a3f  2**0
              CONTENTS, READONLY, DEBUGGING
 5 .debug_loc   00000058  00000000  00000000  00000be0  2**0
              CONTENTS, READONLY, DEBUGGING
 6 .debug_aranges 00000020  00000000  00000000  00000c38  2**0
              CONTENTS, RELOC, READONLY, DEBUGGING
 7 .debug_line  00000173  00000000  00000000  00000c58  2**0
              CONTENTS, RELOC, READONLY, DEBUGGING
 8 .debug_str   000005a2  00000000  00000000  00000dc0  2**0
              CONTENTS, READONLY, DEBUGGING
 9 .comment    0000007f  00000000  00000000  0000136d  2**0
              CONTENTS, READONLY
10 .debug_frame 00000048  00000000  00000000  000013ec  2**2
              CONTENTS, RELOC, READONLY, DEBUGGING
11 .ARM.attributes 00000033  00000000  00000000  00001434  2**0
              CONTENTS, READONLY
```

## 8.7 The elf File

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

Sections:
Idx Name      Size    VMA     LMA     File off  Align
 0 .text      00000324 08000000 08000000 00010000 2**2
                CONTENTS, ALLOC, LOAD, READONLY, CODE
 1 .data      00000000 20000000 08000324 00020000 2**0
                CONTENTS, ALLOC, LOAD, DATA
 2 .bss       00001020 20000000 08000324 00020000 2**2
                ALLOC
 3 .debug_info 00002a9d 00000000 00000000 00020000 2**0
                CONTENTS, READONLY, DEBUGGING
 4 .debug_abbrev 0000090e 00000000 00000000 00022a9d 2**0
                CONTENTS, READONLY, DEBUGGING
 5 .debug_loc   0000048c 00000000 00000000 000233ab 2**0
                CONTENTS, READONLY, DEBUGGING
 6 .debug_aranges 000000c0 00000000 00000000 00023837 2**0
                CONTENTS, READONLY, DEBUGGING
 7 .debug_line   000006dc 00000000 00000000 000238f7 2**0
                CONTENTS, READONLY, DEBUGGING
 8 .debug_str    0000080c 00000000 00000000 00023fd3 2**0
                CONTENTS, READONLY, DEBUGGING
 9 .comment      0000007e 00000000 00000000 000247df 2**0
                CONTENTS, READONLY
10 .ARM.attributes 00000033 00000000 00000000 0002485d 2**0
                CONTENTS, READONLY
11 .debug_frame  000002a4 00000000 00000000 00024890 2**2
                CONTENTS, READONLY, DEBUGGING
```

## 9.0 Symbols

### 9.1 PController.o

```
00000004 C PControllerCurrState
00000070 T PControllerHighPState
00000058 T PControllerLowPState
00000000 b PControllerPressure
0000003c T PControllerSetup
00000001 C PControllerStateID
00000000 T setPressure
U startAlarm
```

### 9.2 PSensorDriver.o

```
$ arm-none-eabi-nm Build/PSensorDriver.o
U getPressureVal
00000004 C PSensorCurrState
00000000 b PSensorPressure
00000001c T PSensorReadingState
00000000 T PSensorSetup
00000001 C PSensorStateID
U setPressure
```

### 9.3 AlarmController.o

```
$ arm-none-eabi-nm Build/AlarmController.o
00000004 C AControllerCurrState
00000038 T AControllerNoAlarmState
0000001c T AControllerSetup
0000005c T AControllerStartedState
00000001 C AControllerStateID
U Delay
U Set_Alarm_actuator
00000000 T startAlarm
```

## 9.4 driver.o

```
$ arm-none-eabi-nm Build/driver.o
00000000 T Delay
00000020 T getPressureVal
00000074 T GPIO_INITIALIZATION
00000038 T Set_Alarm_actuator
```

## 9.5 startup.o

```
$ arm-none-eabi-nm Build/startup.o
00000084 W busFaultHandler
00000084 T defaultHandler
    U E_BSS
    U E_DATA
    U E_TEXT
00000084 W hardFaultHandler
    U main
00000084 W MMFaultHandler
00000084 W NMIHandler
00000000 T resetHandler
    U S_BSS
    U S_DATA
    U stack_top
00000084 W usageFaultHandler
00000000 D vectors
```

## 9.6 main.o

```
$ arm-none-eabi-nm Build/main.o
    U AControllerCurrState
    U AControllerSetup
    U GPIO_INITIALIZATION
00000014 T main
    U PControllerCurrState
    U PControllerSetup
    U PSensorCurrState
    U PSensorSetup
00000000 T setup
```

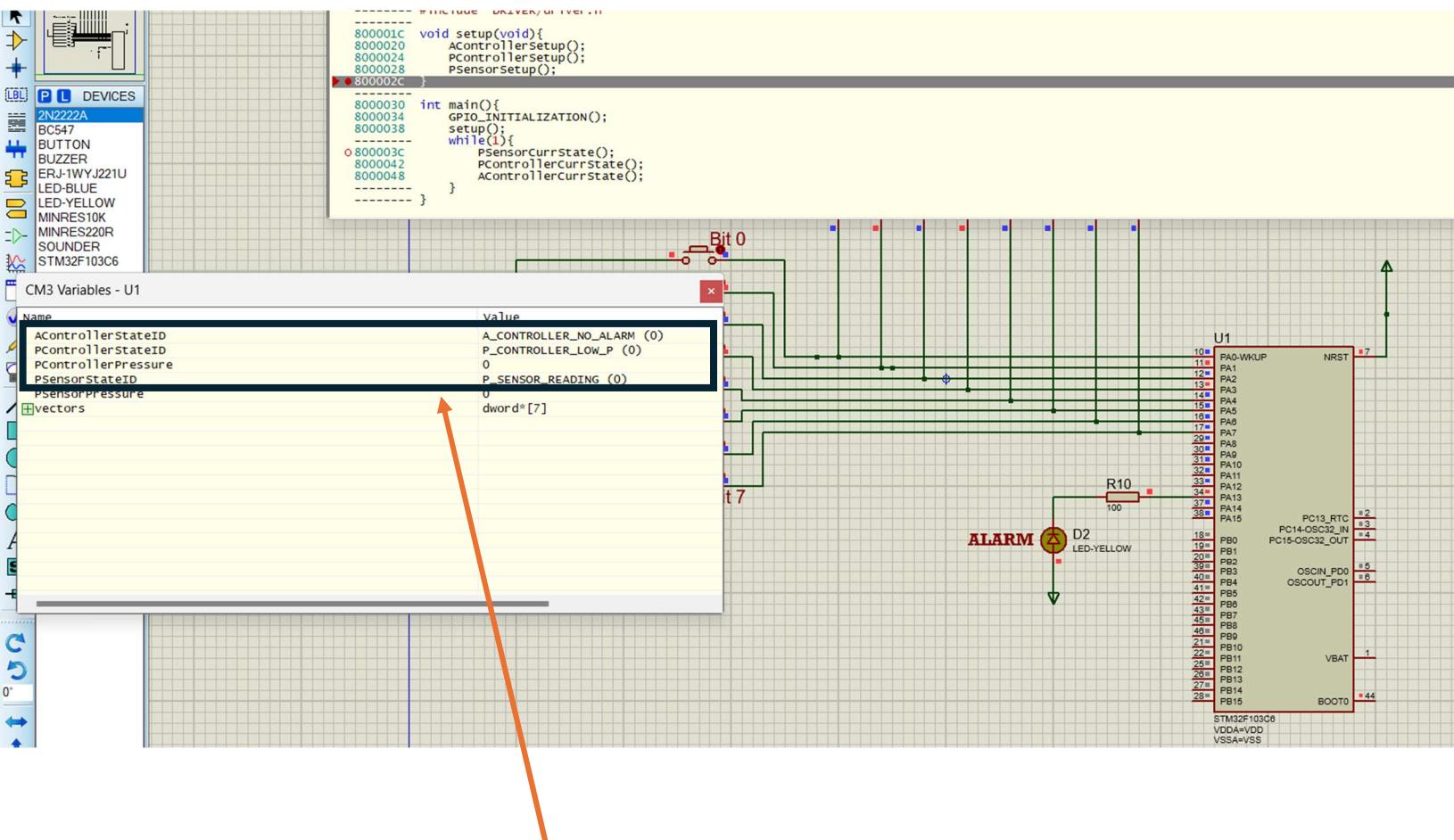
## 9.7 The elf File

```
$ arm-none-eabi-nm PressureCTRL.elf
2000000c B AControllerCurrState
08000124 T AControllerNoAlarmState
08000108 T AControllerSetup
08000148 T AControllerStartedState
20000008 B AControllerStateID
080000e0 W busFaultHandler
080000e0 T defaultHandler
08000260 T Delay
20000020 B E_BSS
20000000 D E_DATA
08000324 T E_TEXT
08000280 T getPressureVal
080002d4 T GPIO_INITIALIZATION
080000e0 W hardFaultHandler
08000030 T main
080000e0 W MMFaultHandler
080000e0 W NMIHandler
20000010 B PControllerCurrState
080001f0 T PControllerHighPState
080001d8 T PControllerLowPState
20000000 b PControllerPressure
080001bc T PControllerSetup
20000014 B PControllerStateID
2000001c B PSensorCurrState
20000004 b PSensorPressure
08000224 T PSensorReadingState
08000208 T PSensorSetup
20000018 B PSensorStateID
0800005c T resetHandler
20000000 B S_BSS
20000000 D S_DATA
08000298 T Set_Alarm_actuator
08000180 T setPressure
0800001c T setup
20001020 B stack_top
080000ec T startAlarm
080000e0 W usageFaultHandler
08000000 T vectors
```

\*That's a lot of symbols 😅

# 10.0 SIMULATION

## 10.1 Initialization



After Initialization

Simulation shows each module starts at  
correct first state

## 10.2 Pressure = 10 bars

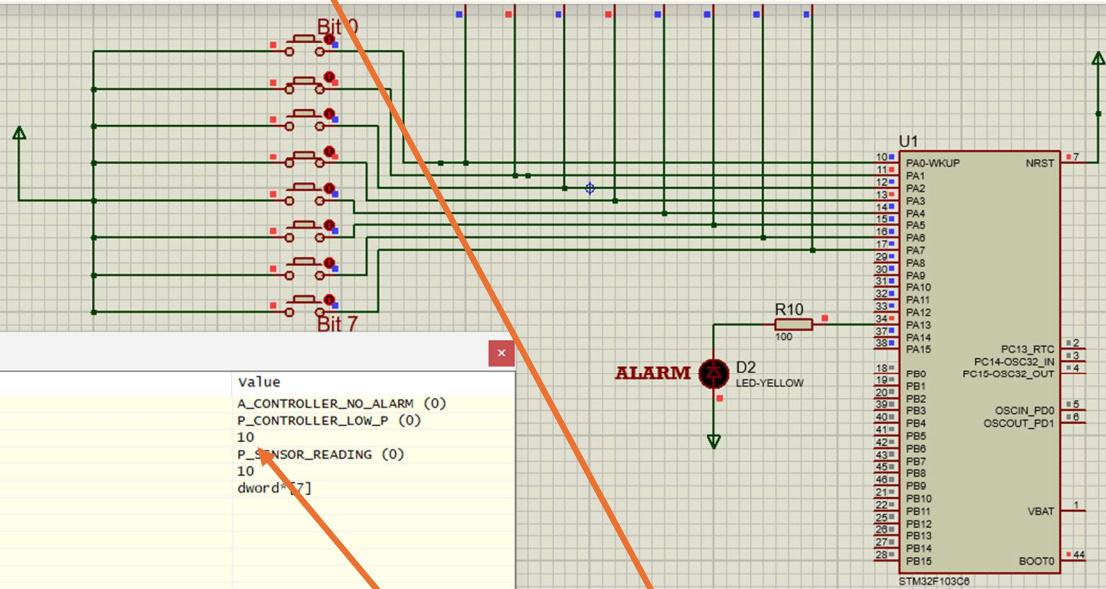
```

8000108 void AControllerSetup(void){
800010C   AControllercurrState = STATE(AControllerNoAlarm);
8000112   Set_Alarm_actuator(1);
8000118 }

8000124 STATE_DEFINE(AControllerNoAlarm){
8000128   AControllercurrStateID = A_CONTROLLER_NO_ALARM;
800012E   AControllercurrState = STATE(AControllerNoAlarm);
8000132 }

8000148 STATE_DEFINE(AControllerStarted){
800014C   AControllerStateID = A_CONTROLLER_STARTED;
8000152   Set_Alarm_actuator(0);
8000158   Delay(600000);
800015E   Set_Alarm_actuator(1);
8000164   AControllercurrState = STATE(AControllerNoAlarm);
800016A }

```



Here Pressure = 10 bars < 20

So Alarm Controller is in No Alarm State

\*Side Note:

In implementation I used a delay of 600,000 not 60 because this wasn't a real timer.

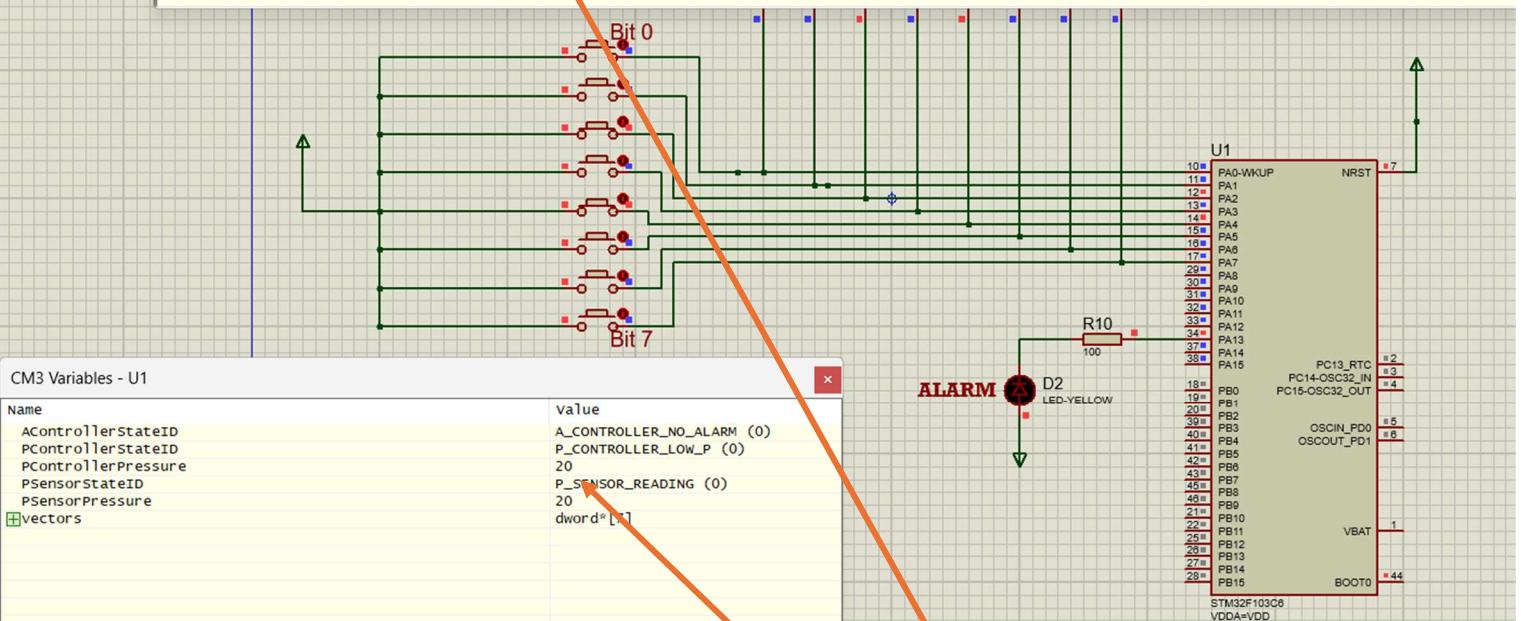
60 was too quick so I increased it to 600,000.

## 10.3 Pressure = 20 bars

```
8000108 void AControllerSetup(void){  
800010C AControllerCurrState = STATE(AControllerNoAlarm);  
8000112 Set_Alarm_actuator(0);  
8000118 }
```

```
8000124 STATE_DEFINE(AControllerNoAlarm){  
8000128 AControllerStateID = A_CONTROLLER_NO_ALARM;  
800012E AControllerCurrState = STATE(AControllerNoAlarm);  
800012F }
```

```
8000148 STATE_DEFINE(AControllerStarted){  
800014C AControllerStateID = A_CONTROLLER_STARTED;  
8000152 Set_Alarm_actuator(0);  
8000158 Delay(600000);  
800015E Set_Alarm_actuator(1);  
8000164 AControllerCurrState = STATE(AControllerNoAlarm);  
800016A }
```



Here Pressure = 20 bars

So Alarm Controller is in No Alarm State

## 10.4 Pressure = 28 bars

