

High Pressure Detector (HBD)

Creator: Eng.Ziad Ashraf Kamel Gamalelden

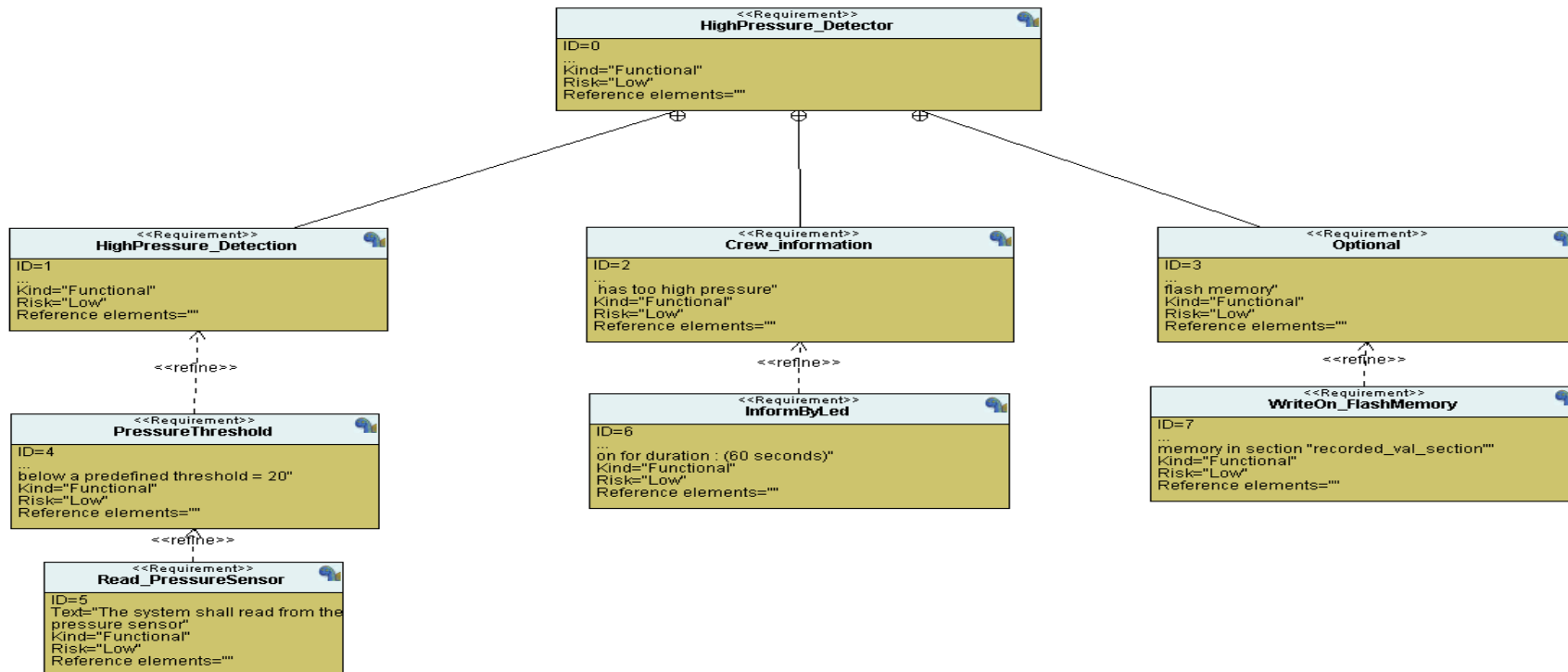
Mentor: Eng.Kerlos Shenouda

Course: Learn-in-Depth Embedded systems.

Requirements:

A system is required to detect high pressure -higher than 20Pa- and raise an alarm for 60 seconds -yellow led- on detection.

- Discussion with the client resulted in this requirement diagram which is confirmed by both sides
- **Notes:** the system assumes the perfection of pressure values and ignores the malfunctioning of any devices, any previously mentioned points are the client's responsibility

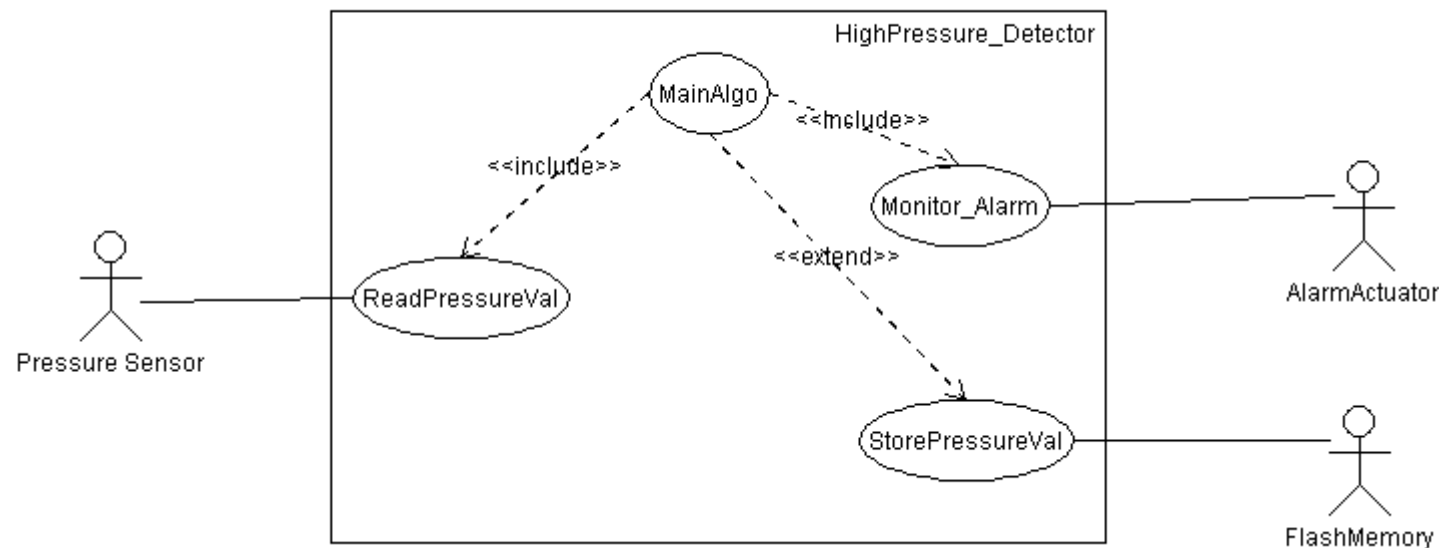


Analysis:

Analyzing the required sensors and the whole system, we concluded the following diagram

- **UML**(use case diagram):

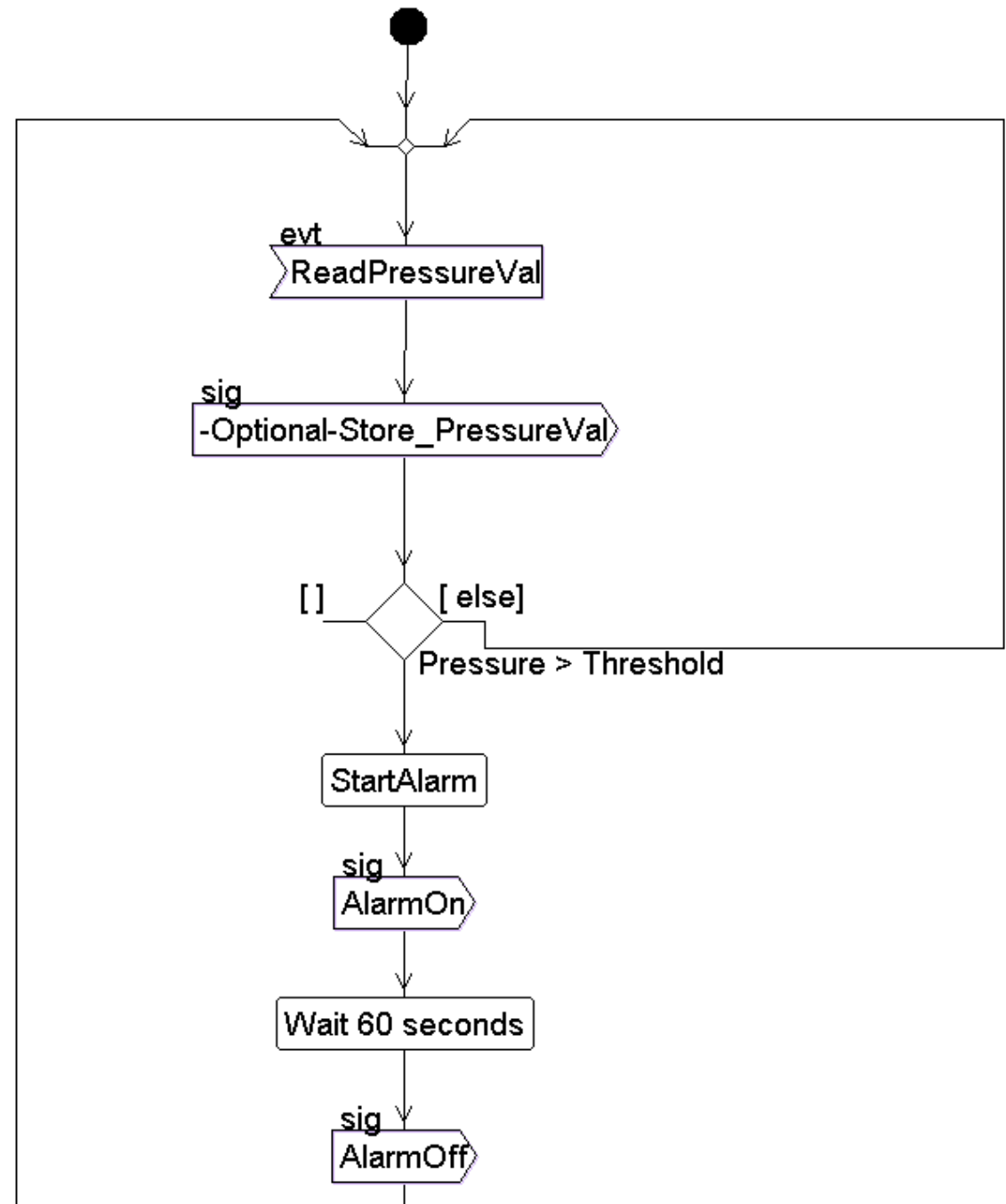
The system flow goes from left to right as pressure value is read from the sensor and sent to the main algorithm which in turn will raise an alarm in case that the pressure goes higher than the pressure threshold.



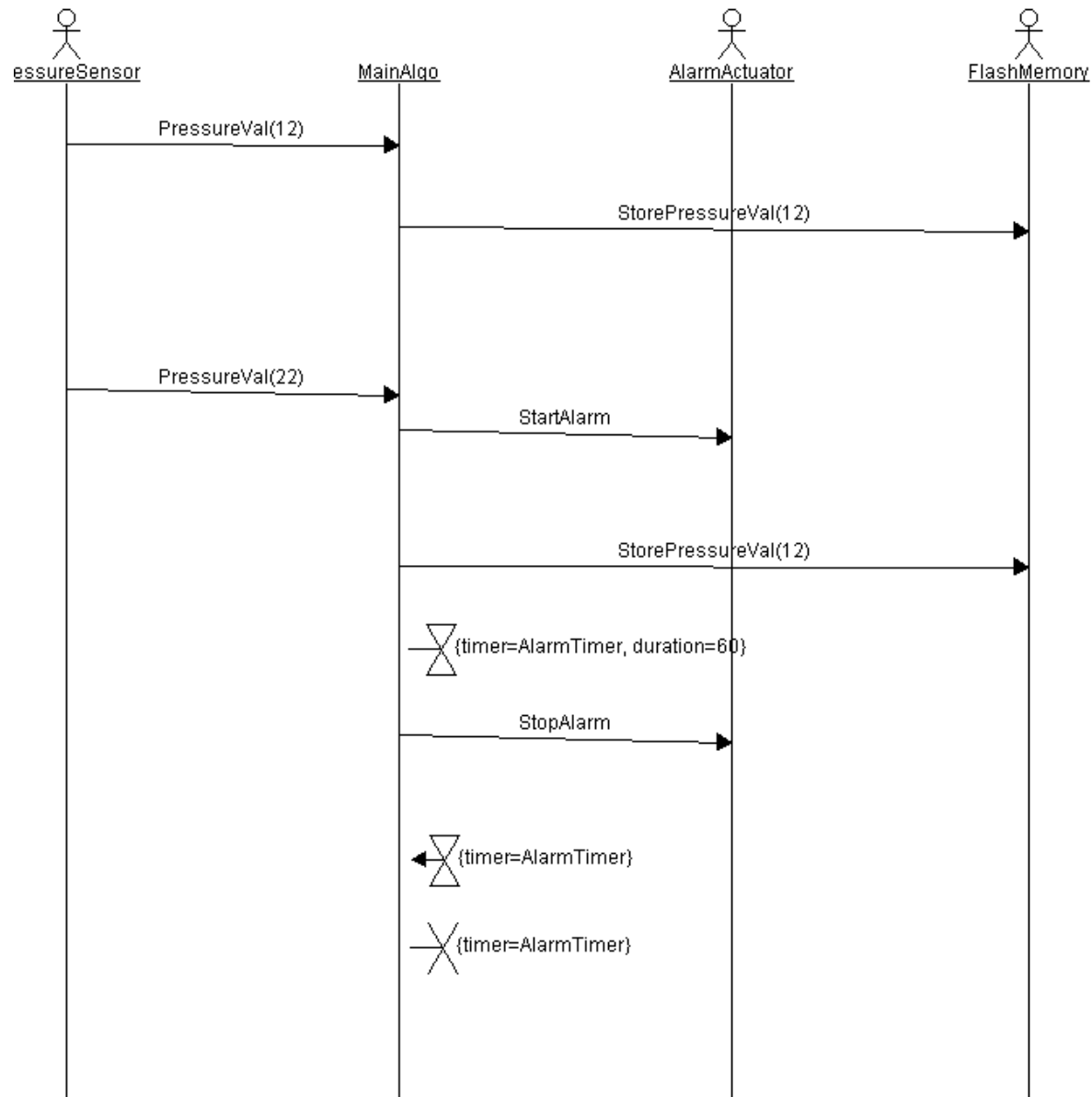
Algorithm:

A flowchart of how the system behaves:

- ❖ Read Pressure value
- ❖ Store it (optional)
- ❖ If (pressure > threshold)
- ❖ Start Alarm
- ❖ Send Alarm signal to actuator
- ❖ Wait 60 seconds
- ❖ Alarm off
- ❖ Repeat step 1
- ❖ Else
- ❖ Repeat step 1

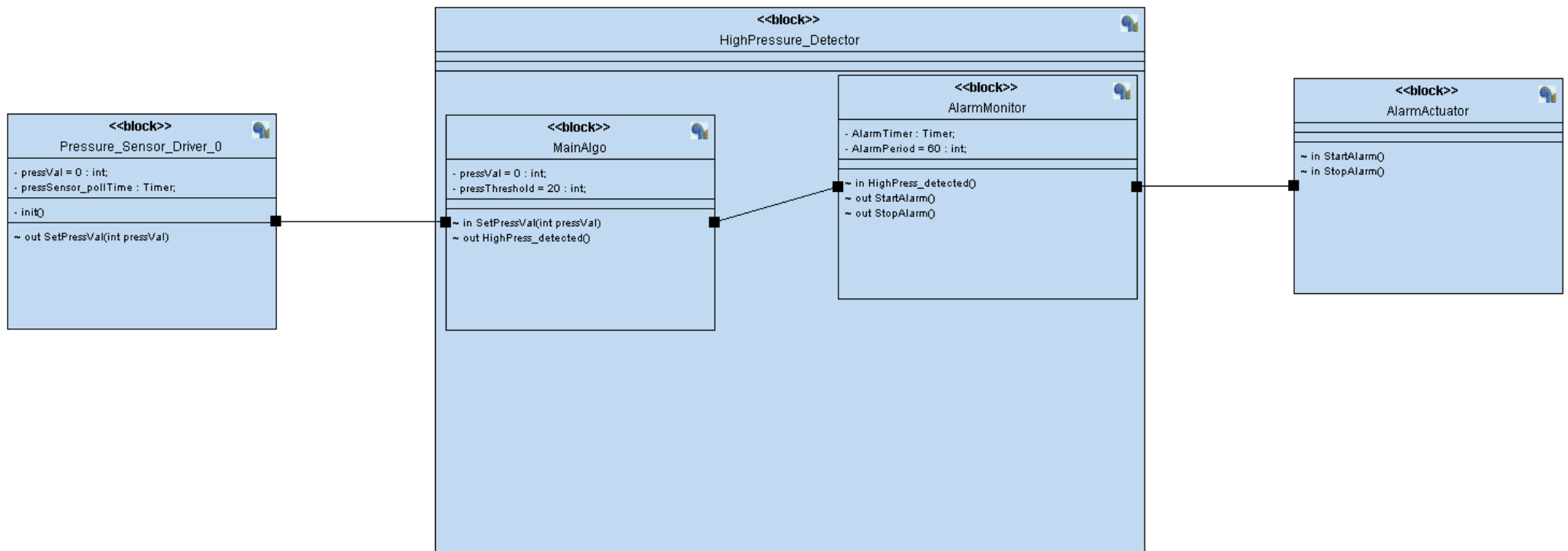


An algorithm timeline for more clarification.



Block diagram:

A block diagram that shows the connection between modules and the flow of the system.



Software simulation:

- Simulating the code on software IDE results in the following as a sequence of what is happening.
- As you see the pressure goes from 22pa above pressure threshold, therefore HP is detected.
- After the pressure goes below or equal to 20, the alarm keeps on for 60 seconds and then is turned off.

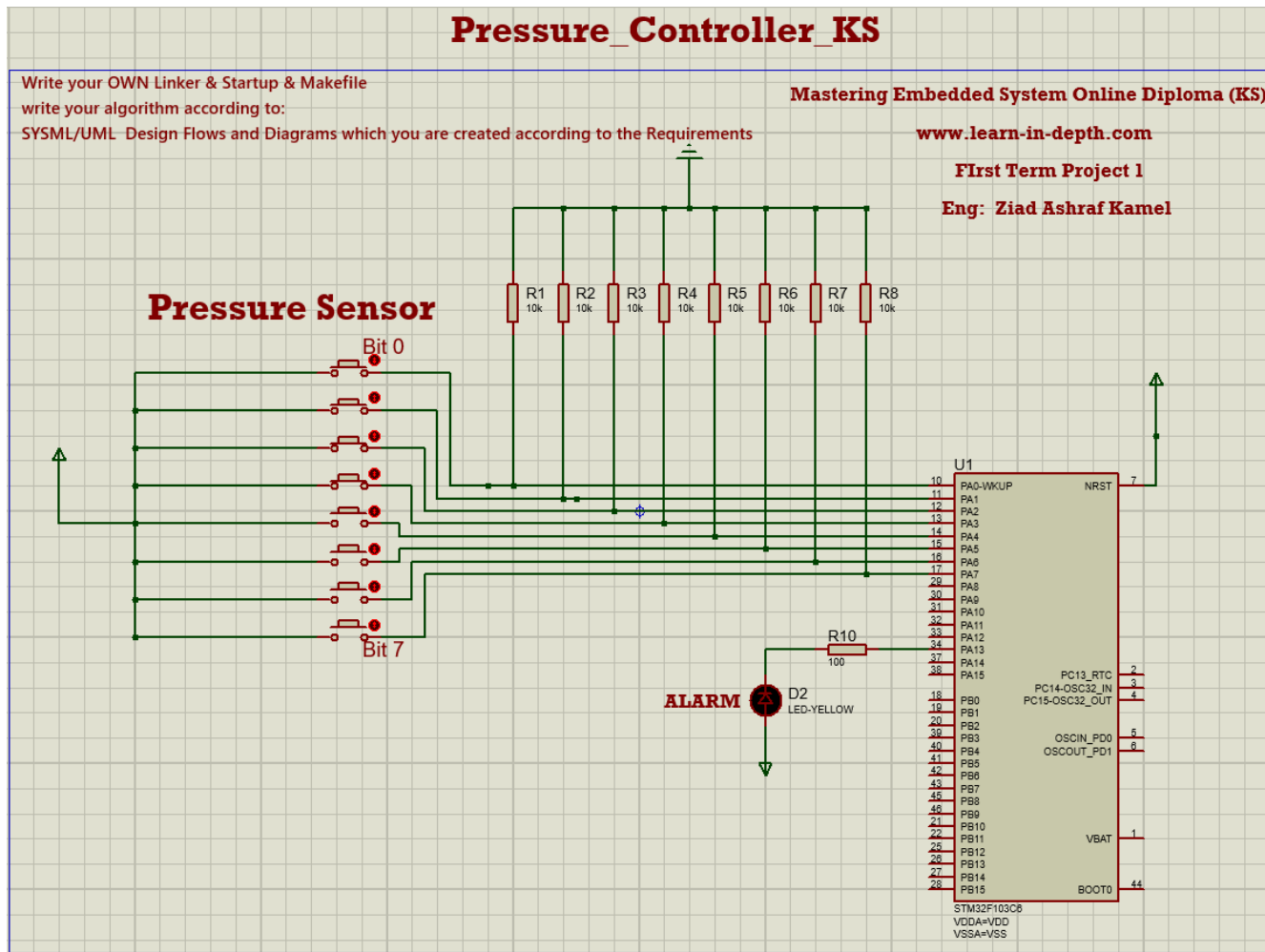
```
HP sensor initialized
HP detector initialized
HP alarm monitor initialized
HP_SENSOR_READING: Pressure Value = 22
Sensor >>> Detector: Pressure Value = 22
HP_DETECTOR_READING: Pressure Value = 22, -HP DETECTED-
Detector >>> AlarmMonitor: High Pressure detected!
HP_MON_ALARMON: Alerting actuator.
HP_ACT_ALARMON: Alarm is on.
HP_SENSOR_READING: Pressure Value = 20
Sensor >>> Detector: Pressure Value = 20
HP_DETECTOR_READING: Pressure Value = 20, -HP NOT DETECTED-
HP_ALARMWAITING: Alarm is still on|
HP_SENSOR_READING: Pressure Value = 18
Sensor >>> Detector: Pressure Value = 18
HP_DETECTOR_READING: Pressure Value = 18, -HP NOT DETECTED-
HP_ALARMWAITING: Alarm is still on
HP_SENSOR_READING: Pressure Value = 16
Sensor >>> Detector: Pressure Value = 16
HP_DETECTOR_READING: Pressure Value = 16, -HP NOT DETECTED-
HP_ALARMWAITING: Alarm is still on
HP_SENSOR_READING: Pressure Value = 14
Sensor >>> Detector: Pressure Value = 14
HP_DETECTOR_READING: Pressure Value = 14, -HP NOT DETECTED-
HP_ALARMOFF: Halting Actuator
HP_ACT_ALARMON: Alarm is off.
HP_SENSOR_READING: Pressure Value = 12
Sensor >>> Detector: Pressure Value = 12
HP_DETECTOR_READING: Pressure Value = 12, -HP NOT DETECTED-
HP_SENSOR_READING: Pressure Value = 10
Sensor >>> Detector: Pressure Value = 10
```

Note: You can change code from being software simulated to be ready for bare-metal application by changing only one line in code which disables global debugging.

Hardware Simulation:

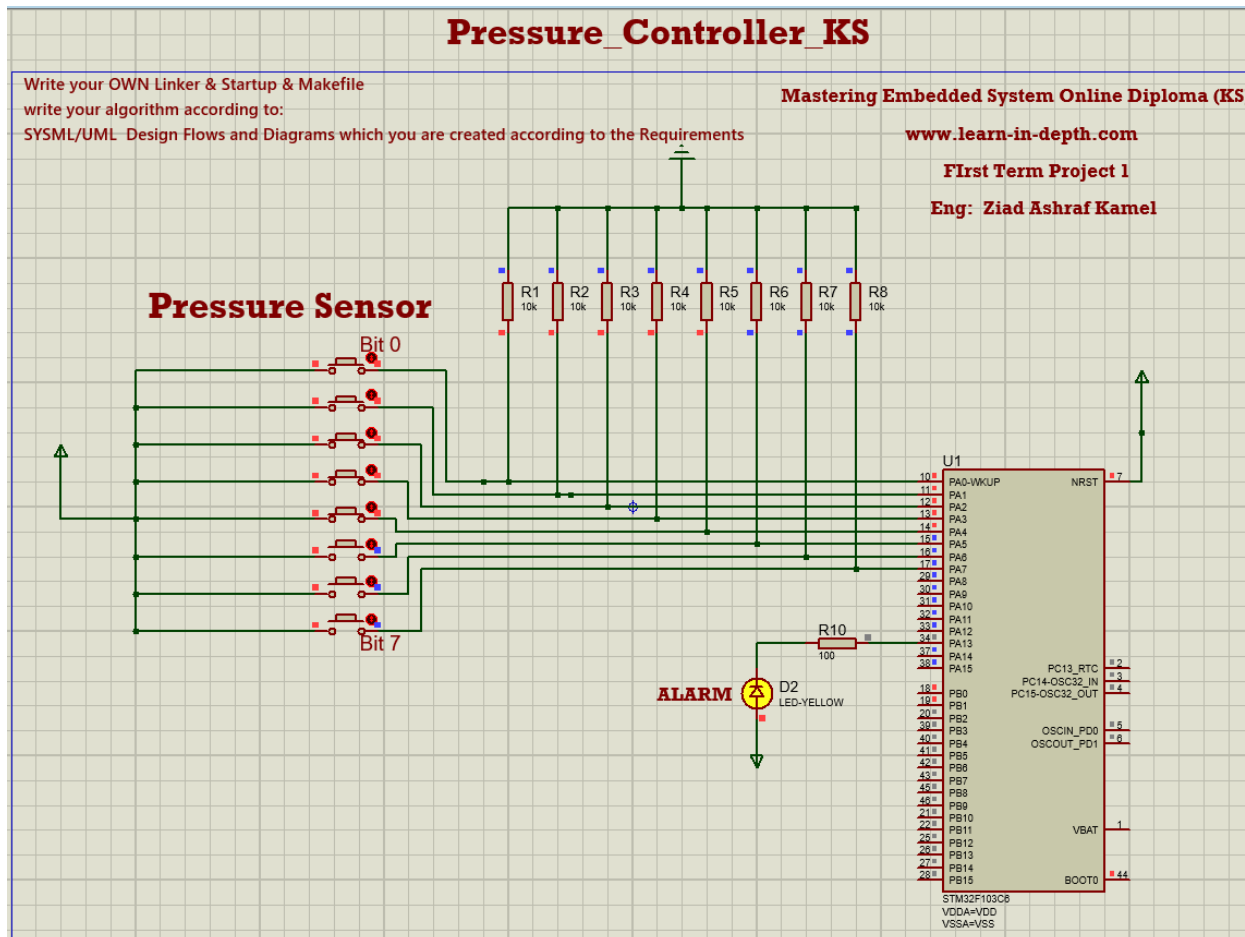
A simulation snapshot to show how the system works:

- The pressure sensor is modeled as a binary entry system.
- The led D2 is the alarm actuator.



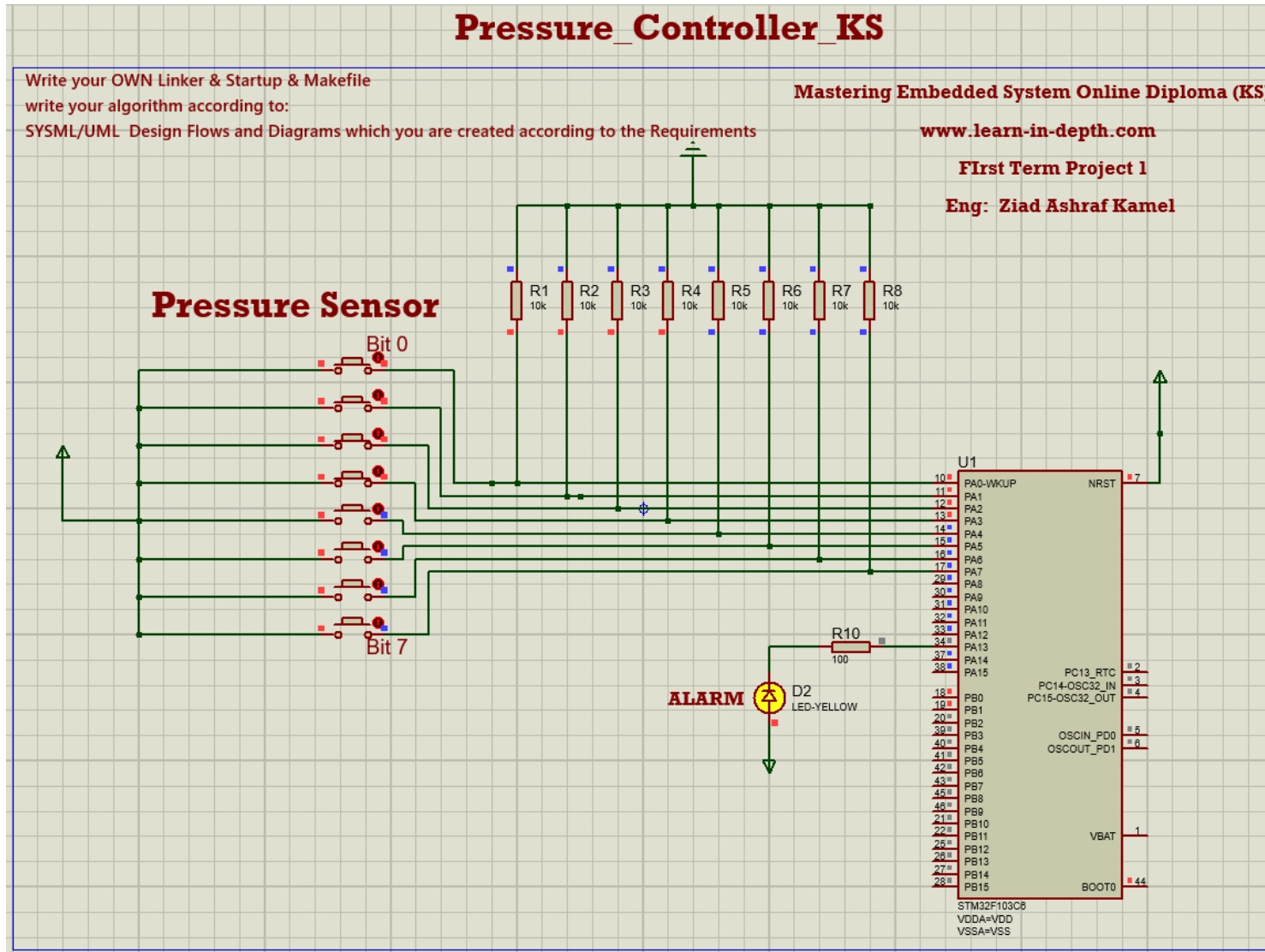
State1(High pressure):

- The pressure went over the pressure threshold.
- The pressure now is 31 (11111 in binary)
- The alarm is raised on instantly and will hold until the pressure goes down.



State2(Low pressure after being alarmed):

- The pressure now is 15 (01111 in binary)
- The alarm actuator keeps on for full 60 seconds.



State3(After waiting for 60 seconds):

- As you see the led is now off as the alarm is halted.

