Airplane cabinet pressure detection

Prepared by: Ahmed Khaled Khalaf

* Contents: -
  + Case study
  + Methodology
  + (HW/SW Partitioning)
  + Requirement Diagram
  + System analysis
  + System analysis: Use case diagram
  + System analysis: Activity diagram
  + System analysis: Sequence diagram
  + System design
* Pressure sensor state diagram
* Main Block state diagram
* Alarm manager state diagram
* Alarm Actuator state diagram
  + A brief look at source files
  + Finding the entry point using readelf utility
  + A brief look at the map file and symbols
  + Debugging and testing using keil
  + Software usage and hardware simulation
* Case study:

Pressure Detection system to inform the crew of the airplane in the cabinet with an alarm when the pressure exceeds 20 bars in that cabinet.

The alarm duration equals 60 seconds. Keep tracking of the measured values.

* Assumptions:
* System setup and shutdown procedure are not modelled.
* System maintenance is not modelled.
* The pressure sensor never fails.
* Alarm actuator never fails.
* Alarm Led never fails.
* System never faces power cut.
* Methodology:

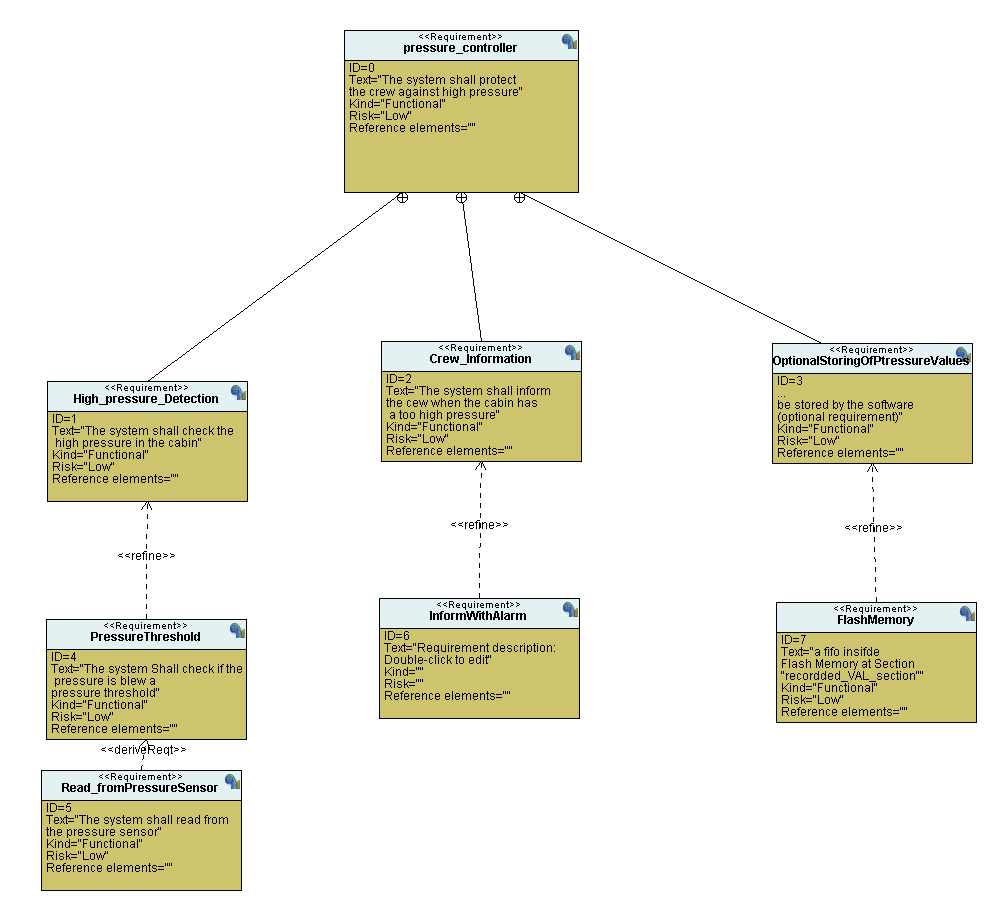
Since the system has multiple modules and seems to be at the large scale, the system will use a testing-based model like v-model. Every phase in this project will be tested and especially the implementation phase. Each software module will be implemented and unit-tested separately then integrated and integration testing will be performed.

* HW/SW Partitioning:

For the hardware, we have STM32 microcontroller with a cortex-m3

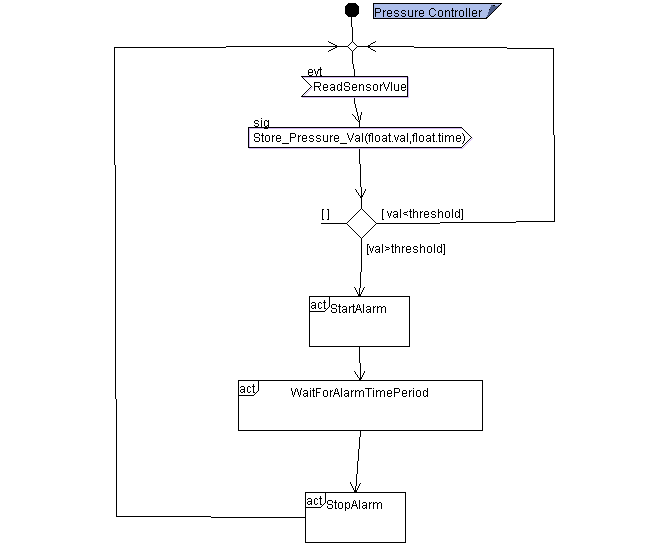
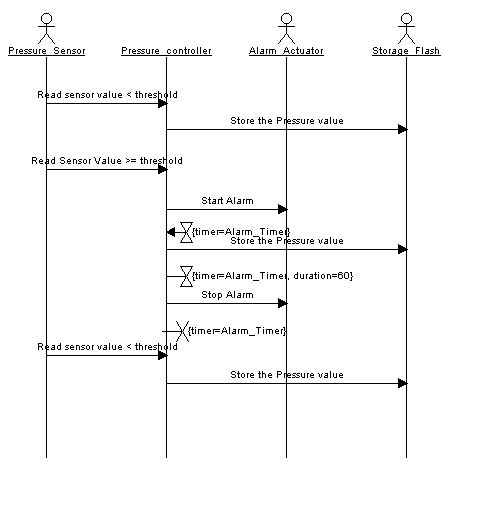
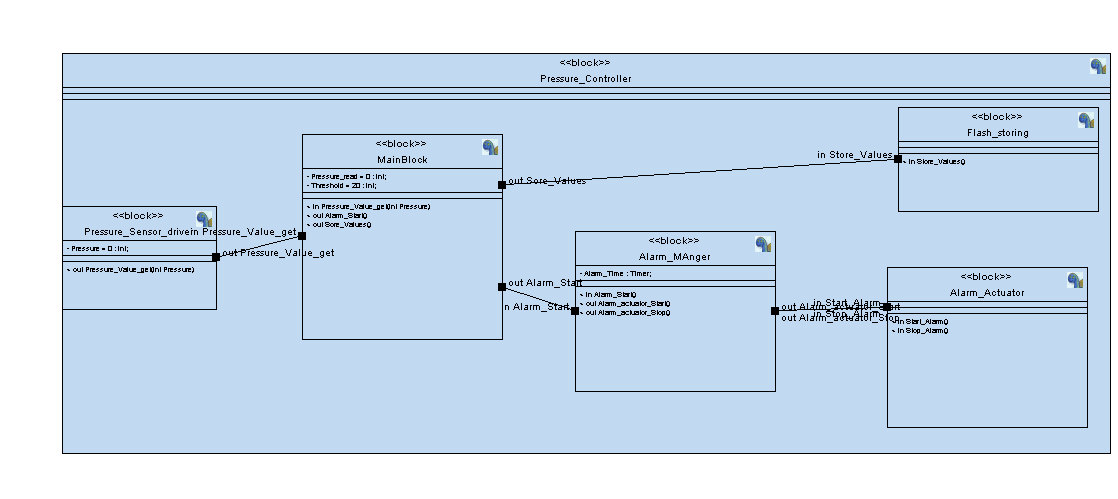
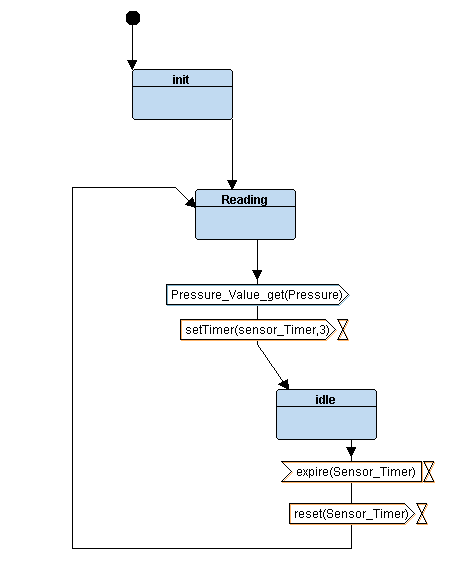
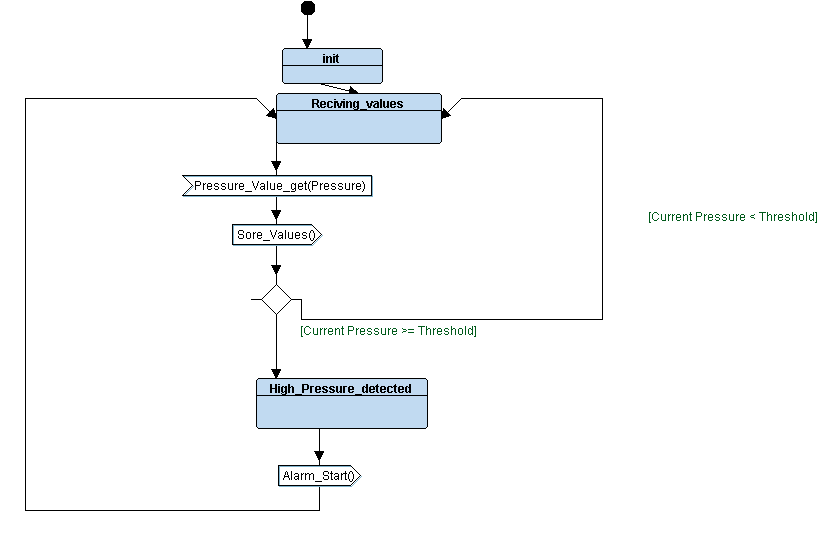
processor that will be enough for this application, pressure sensor (simulated as 8 bush buttons connected with Pull up resistances), and Alarm actuator (simulated as yellow led and pull up resistance).

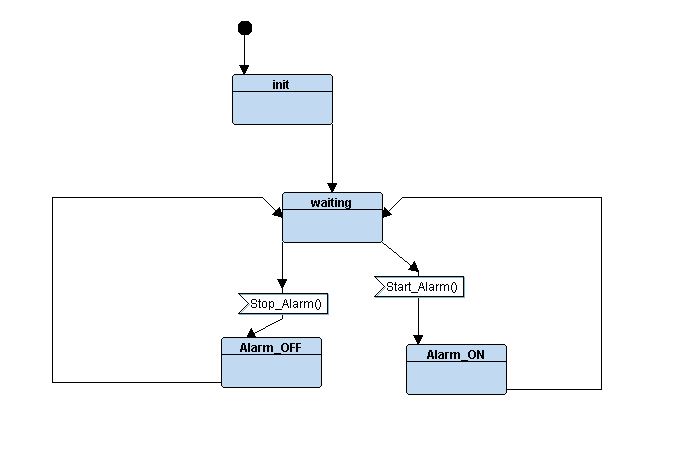
* Requirement Diagram:

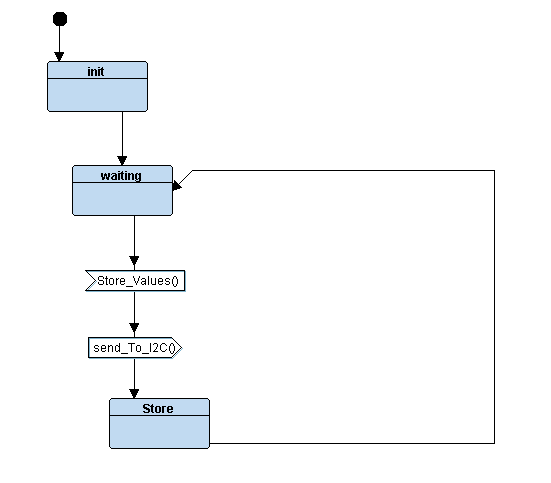


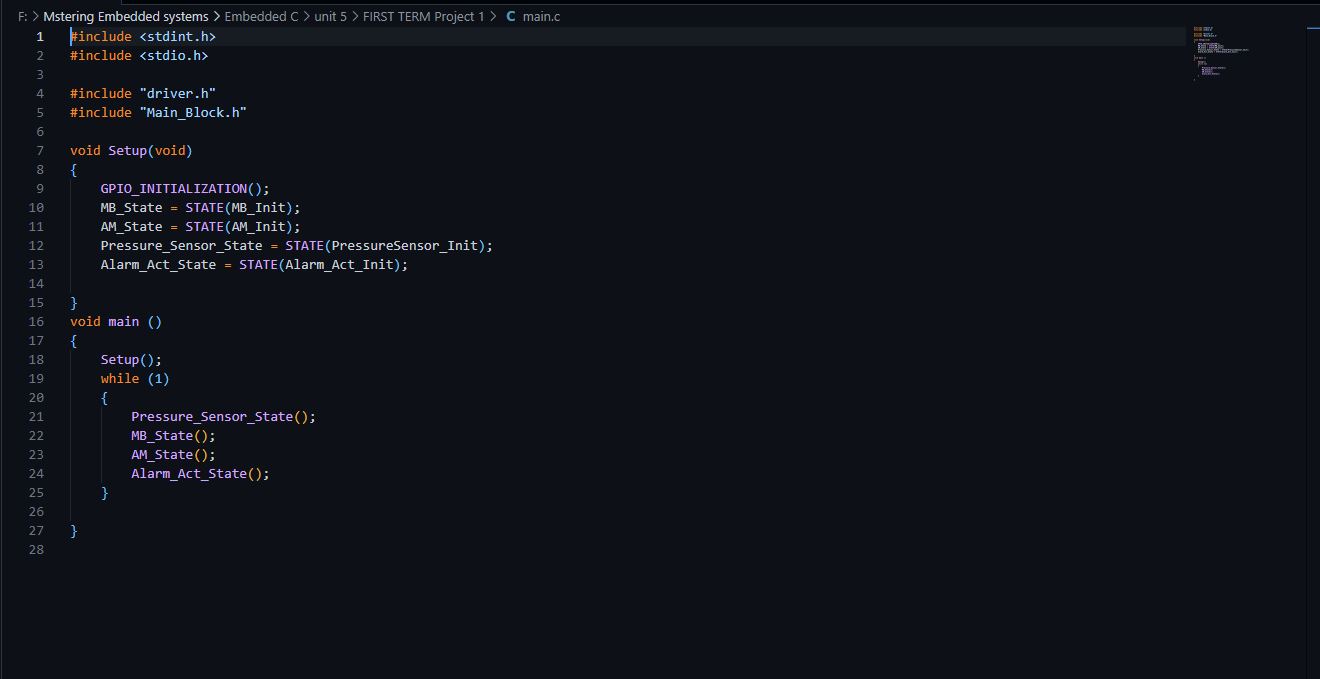
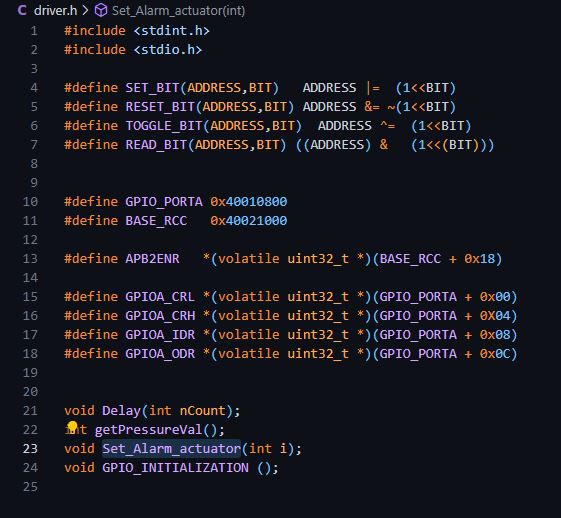
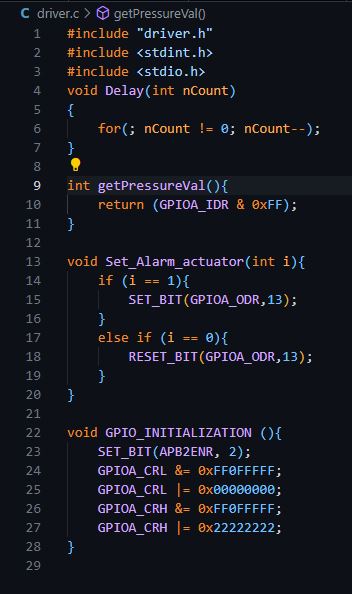
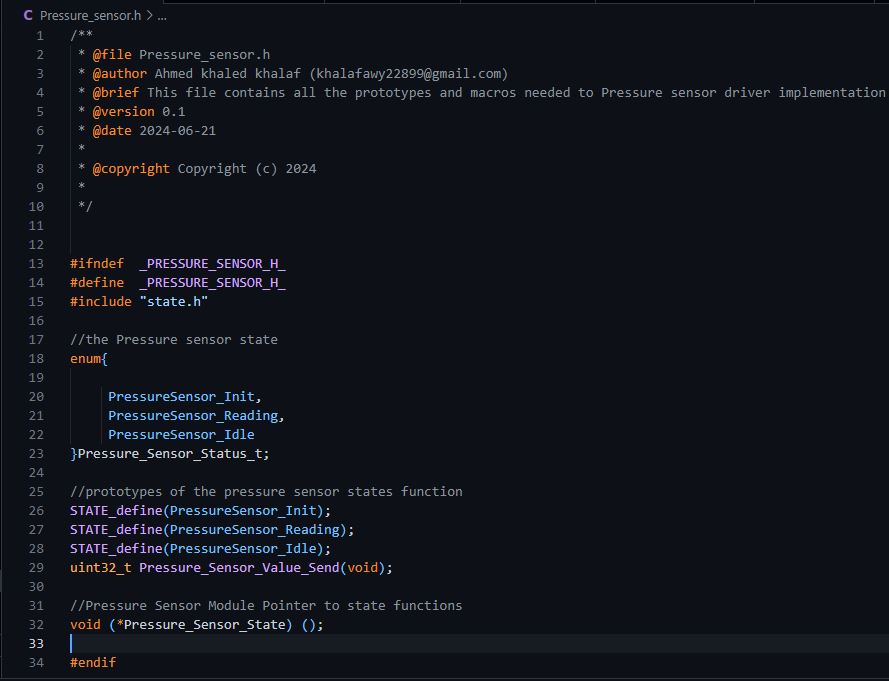
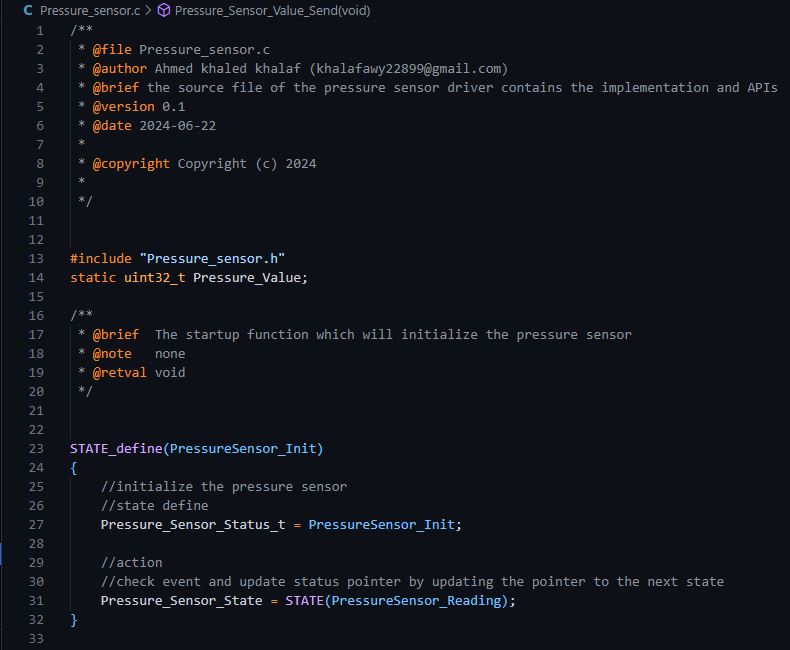
* System analysis:
* Use case diagram:

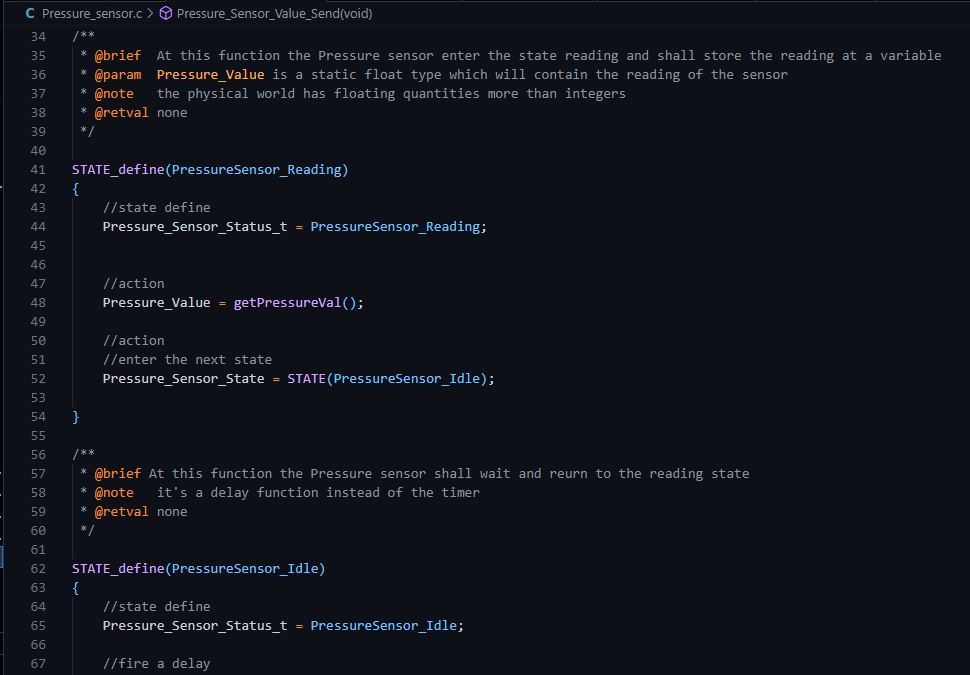
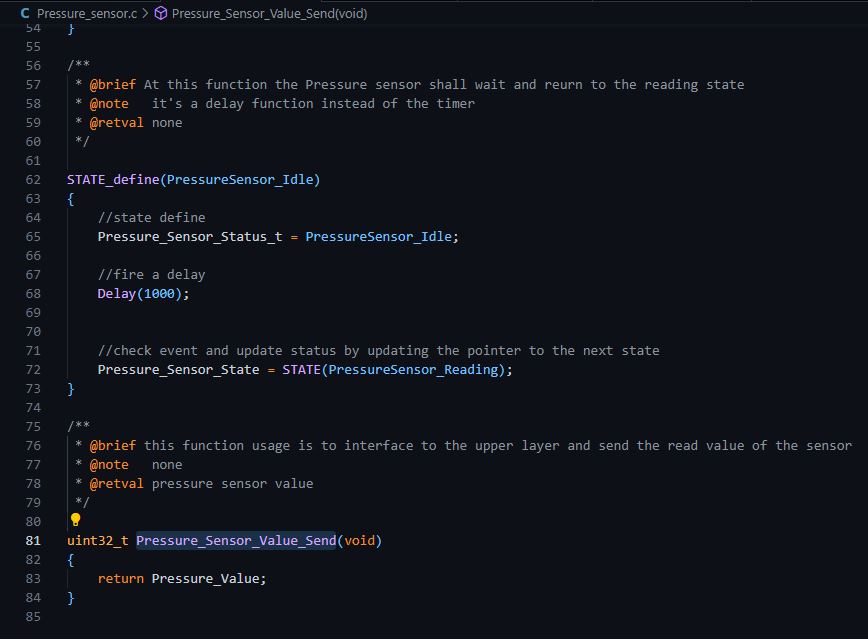


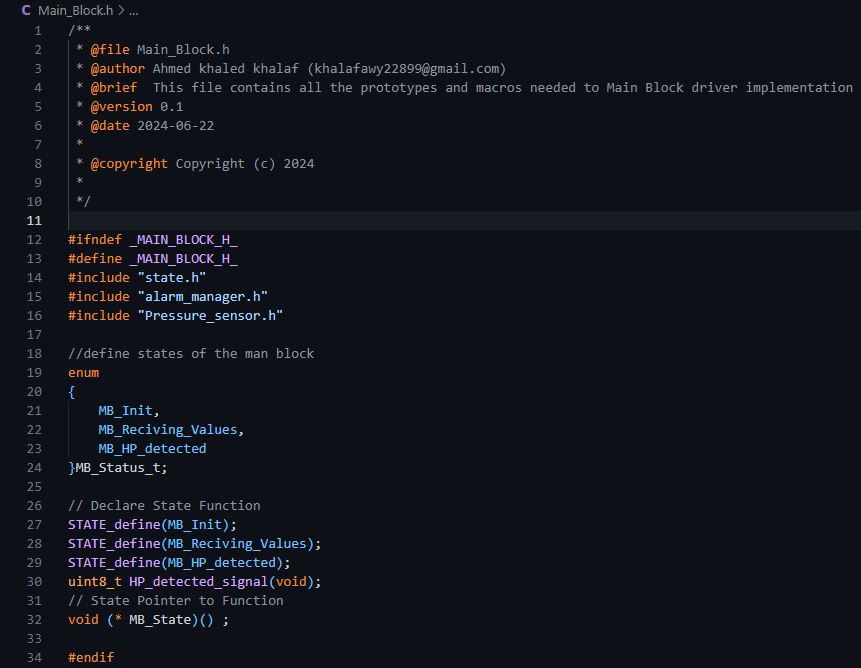
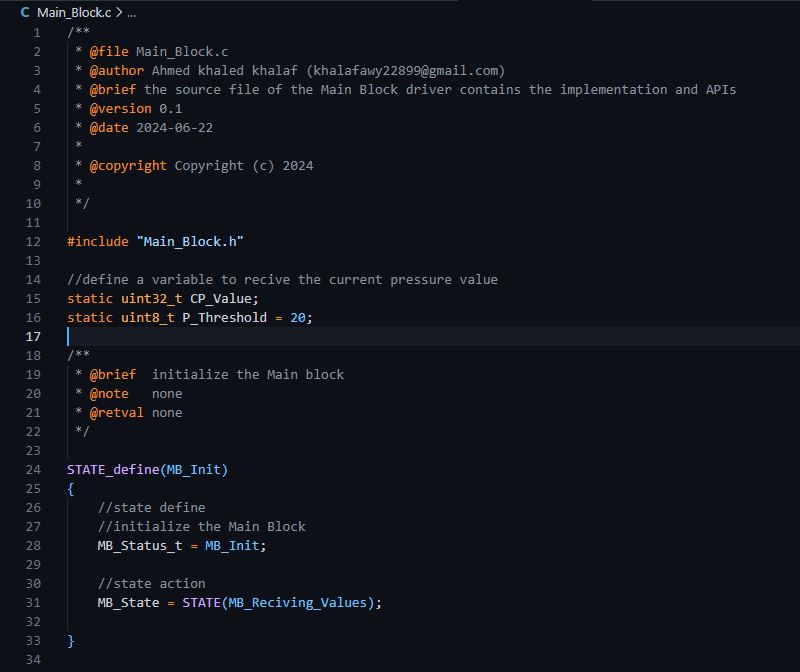
* Activity diagram:
* Sequence diagram:
* System design:
* Pressure state diagram:
* Main Block state diagram:
* A diagram of a flowchart

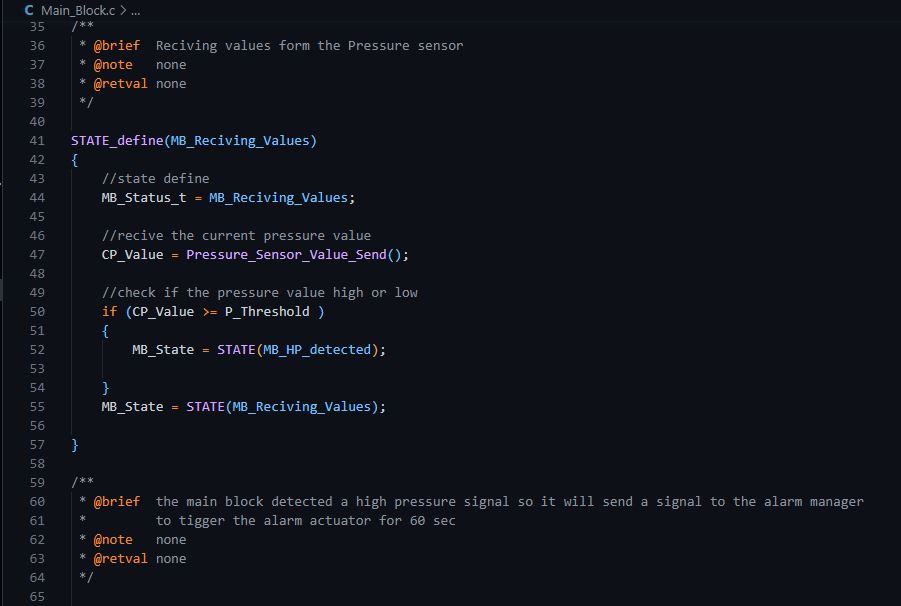
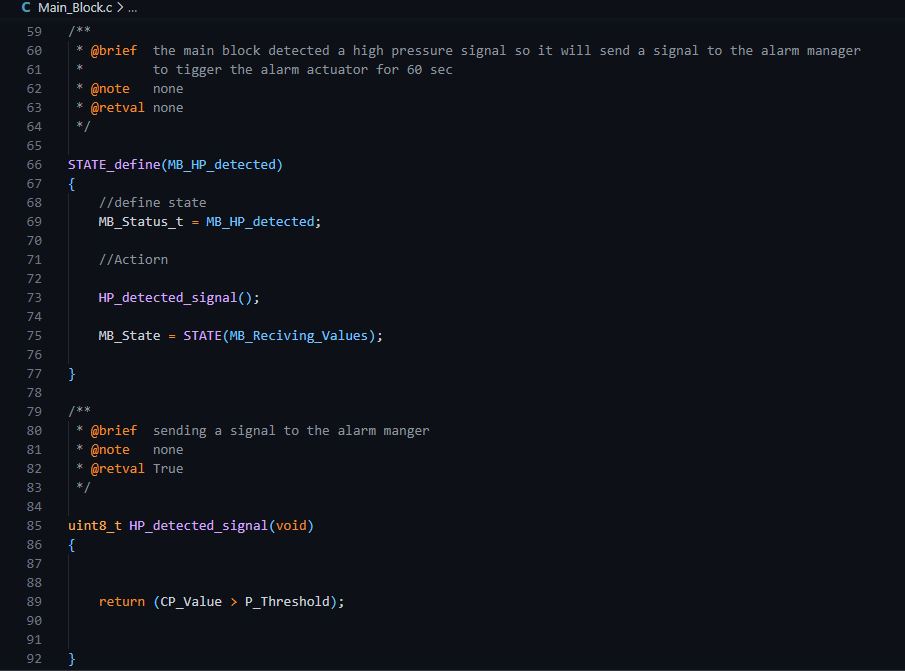
  Description automatically generatedAlarm Manager state diagram:
* Alarm Actuator state diagram:
* Flash memory store state diagram:

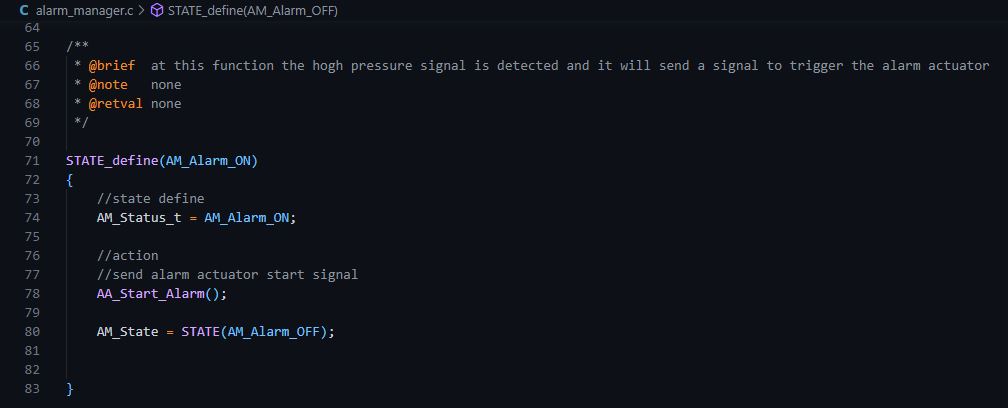
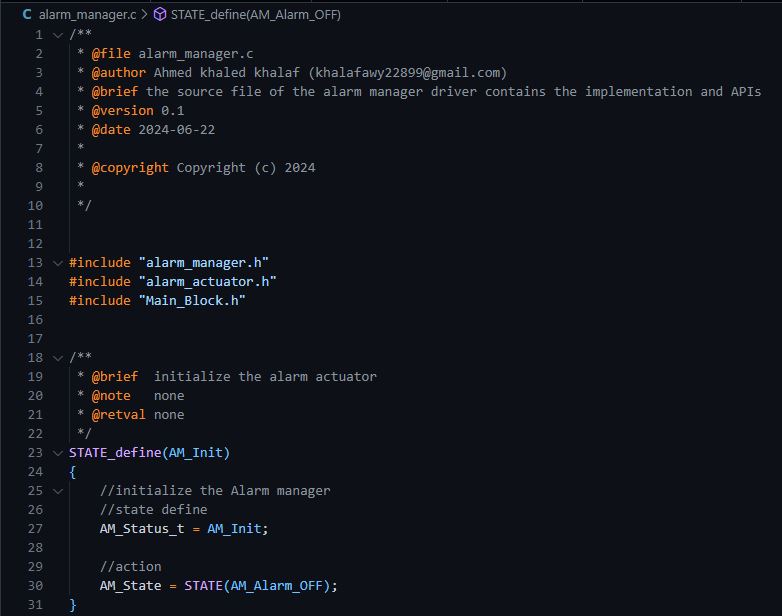


* Main.c
* State.h
* driver .h and .c
* Pressure sensor driver

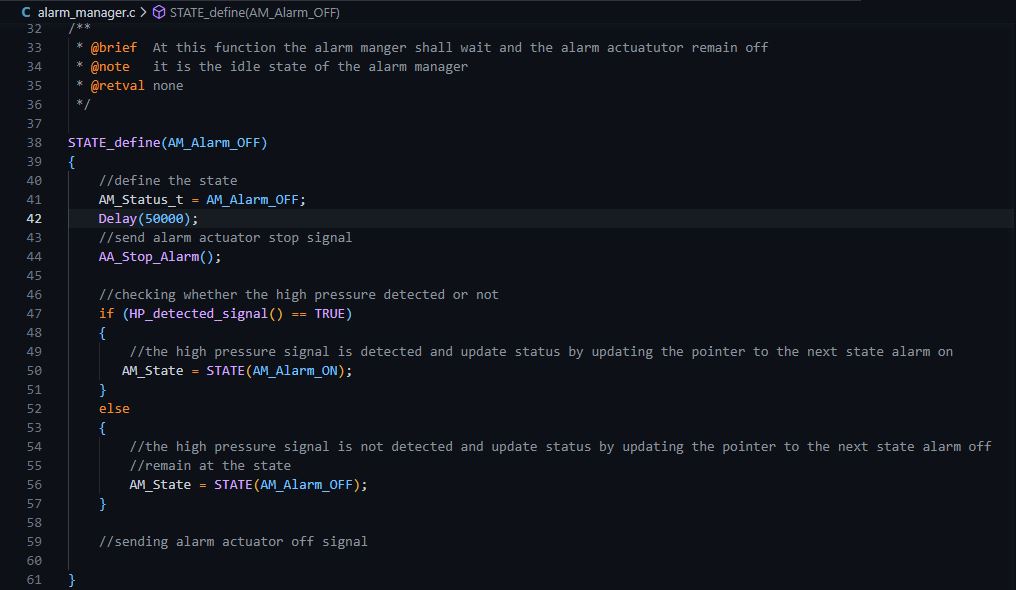


* Main block:



* Alarm Manager:

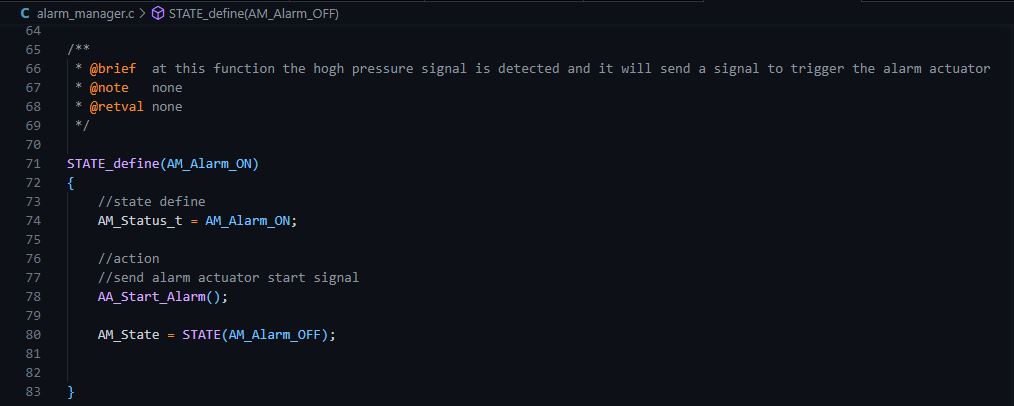
A screenshot of a computer

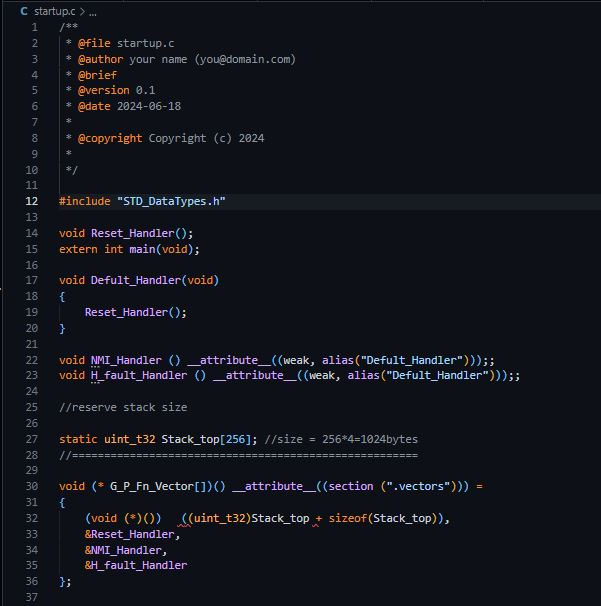
Description automatically generated

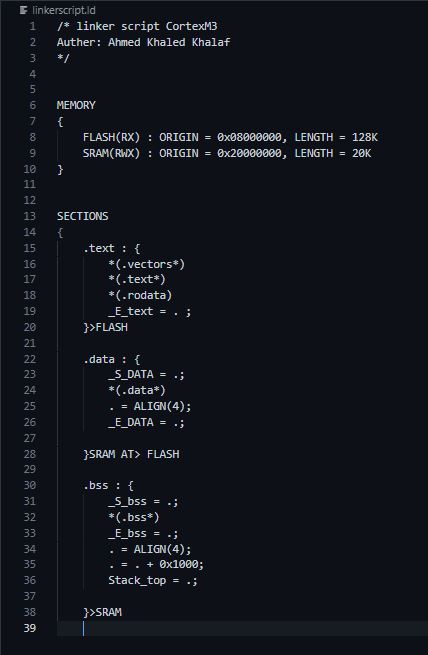
* A screenshot of a computer screen

  Description automatically generatedAlarm actuator:

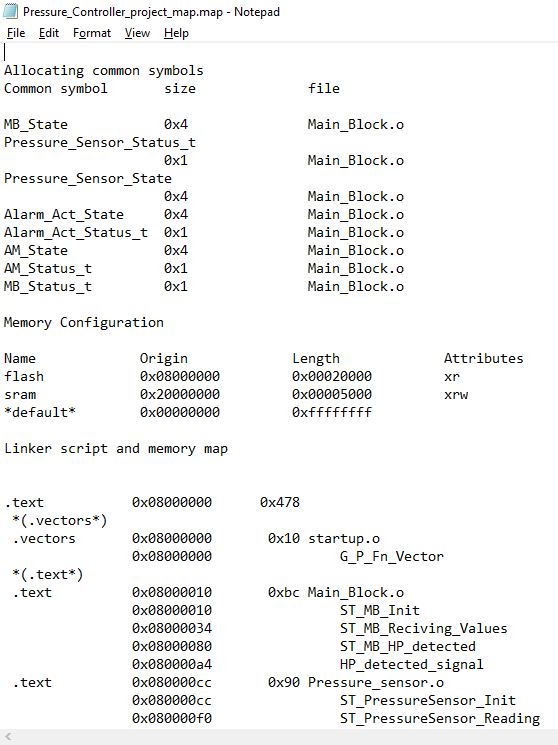




* Startup.c:
* Linker script:

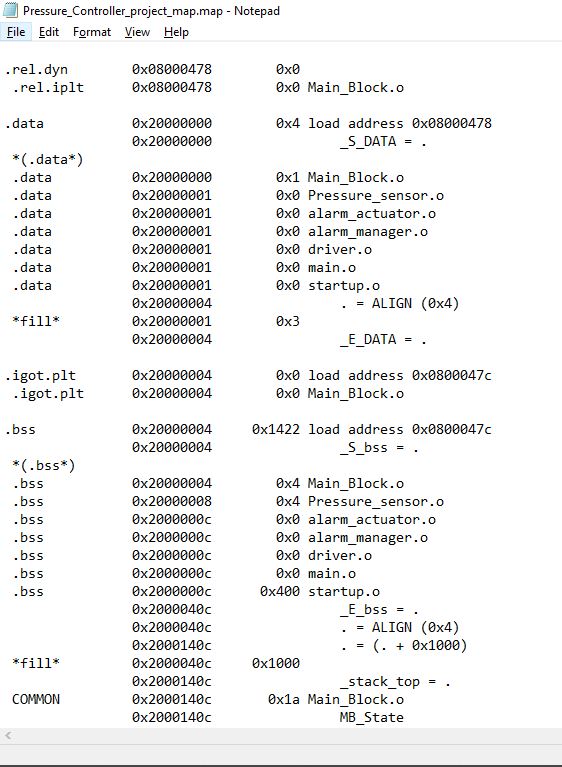


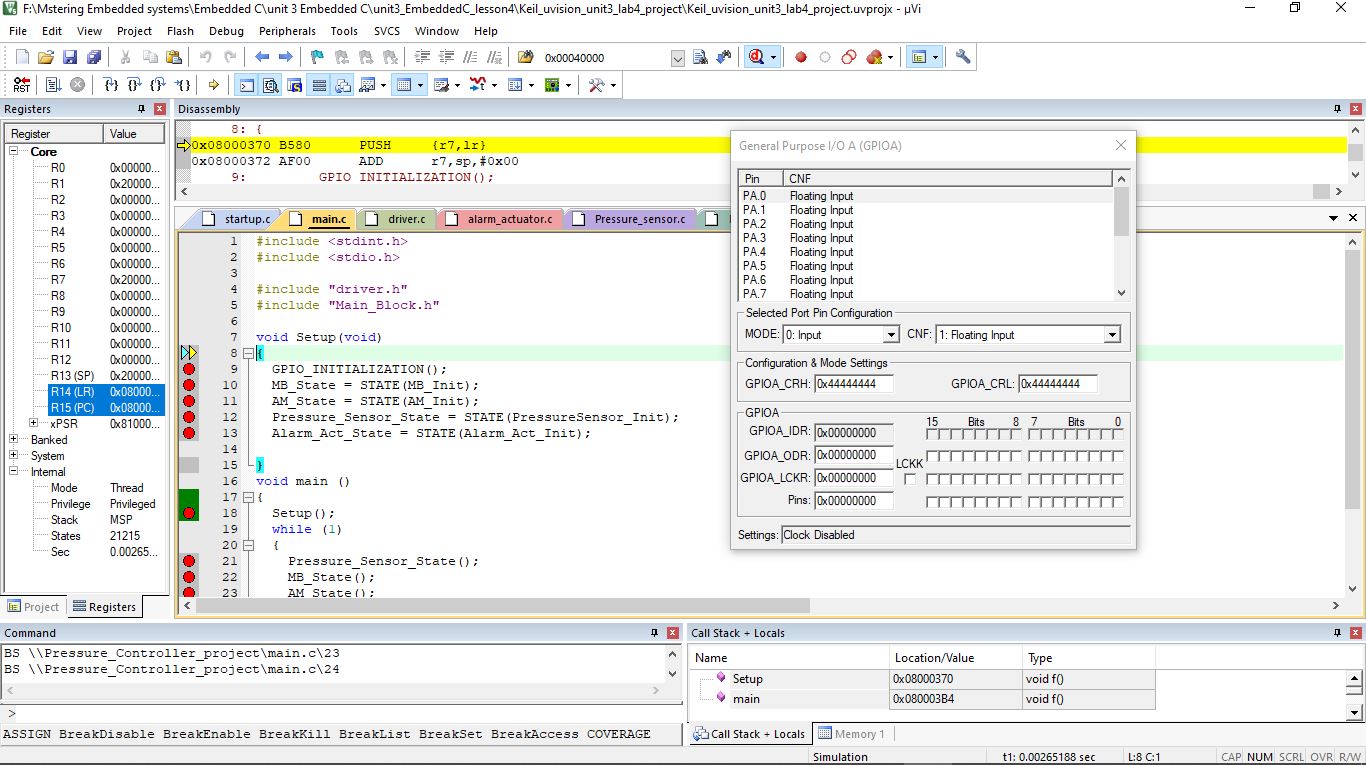
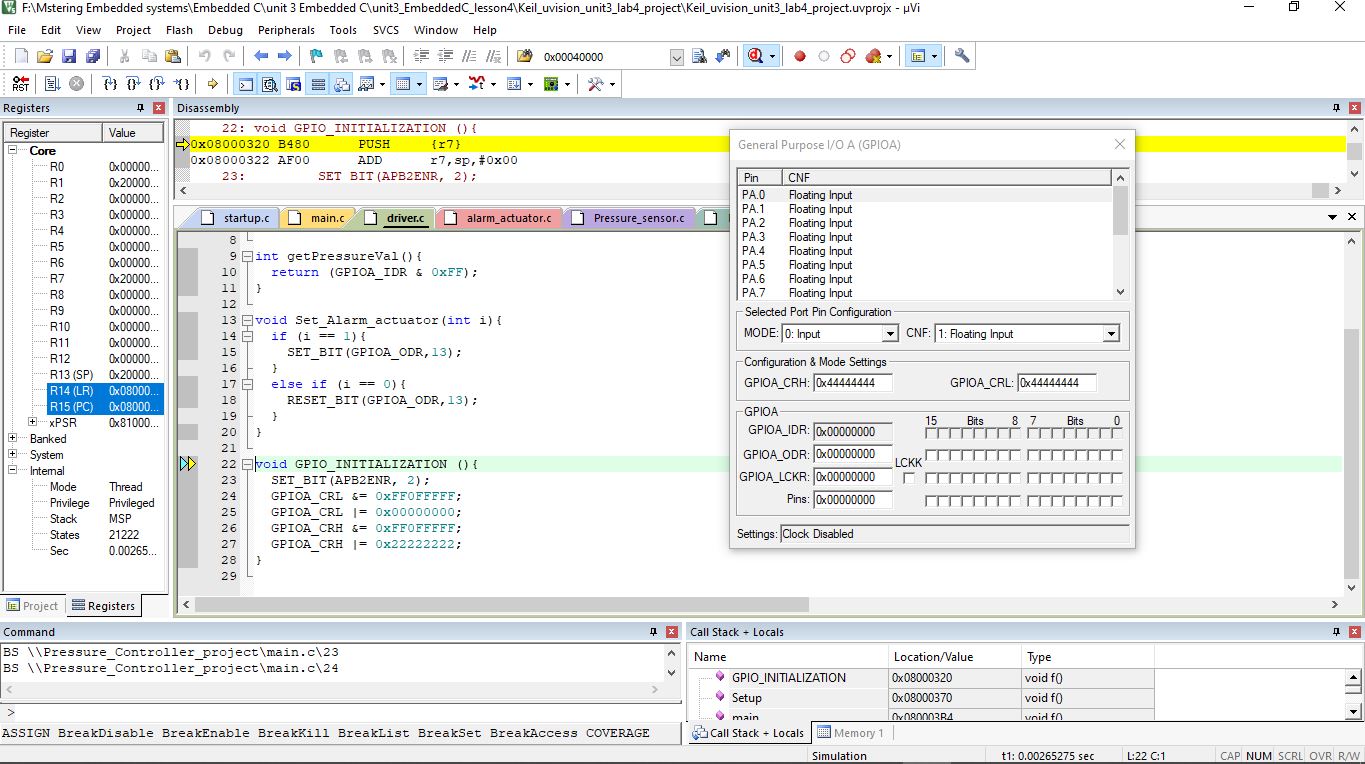
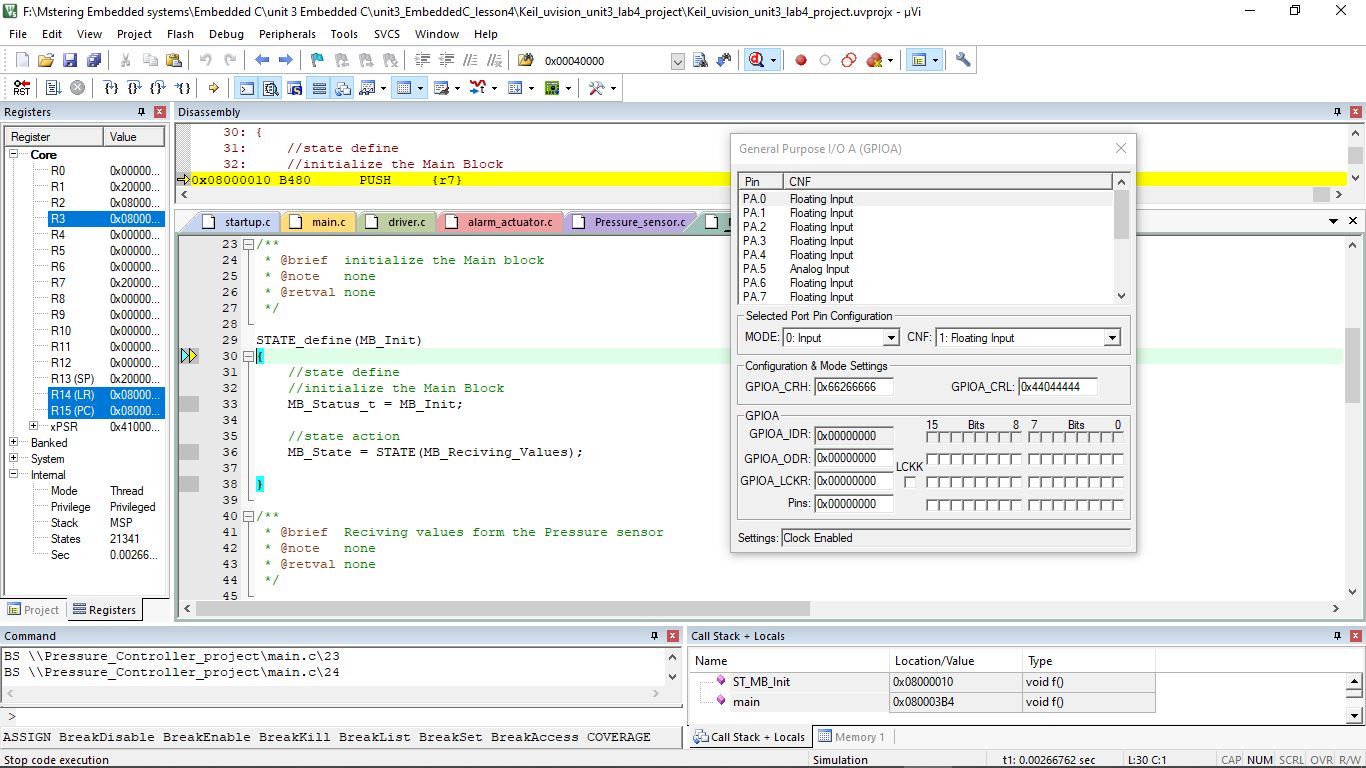
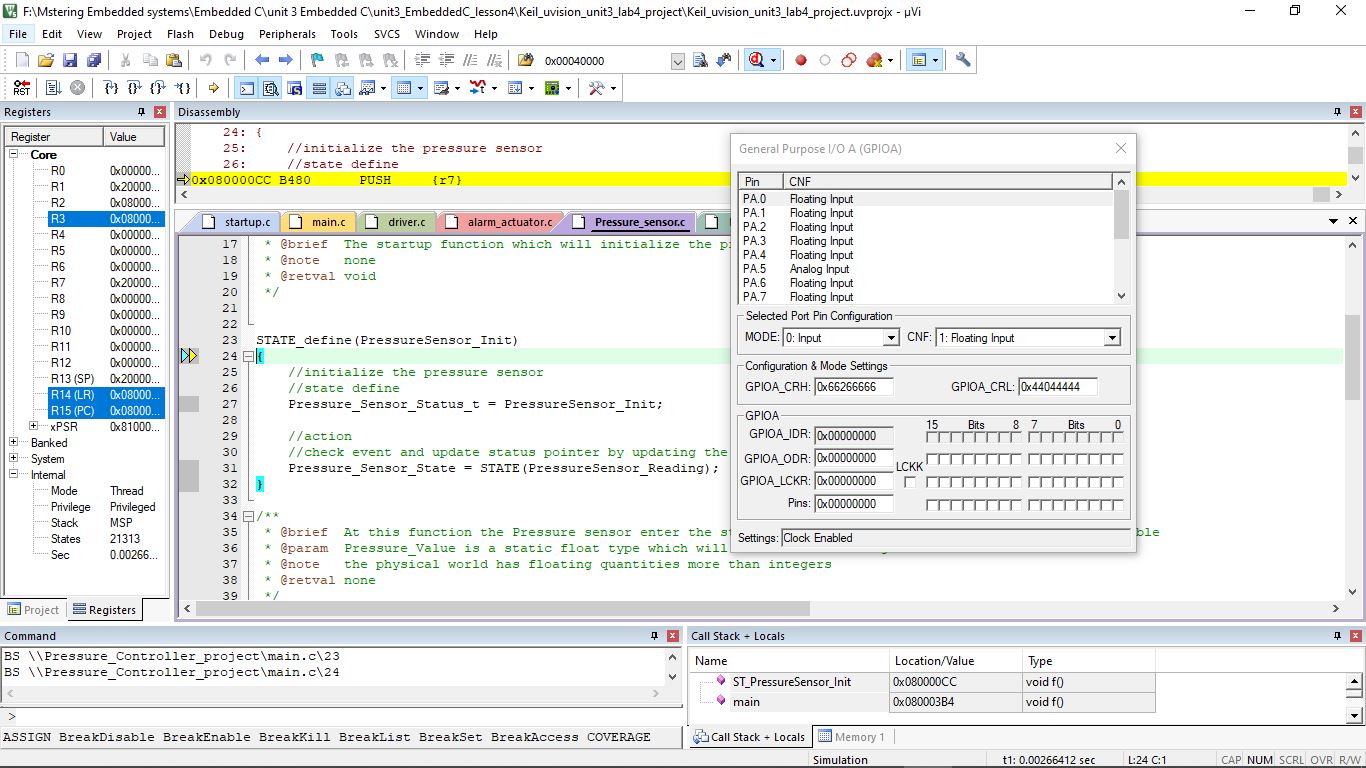
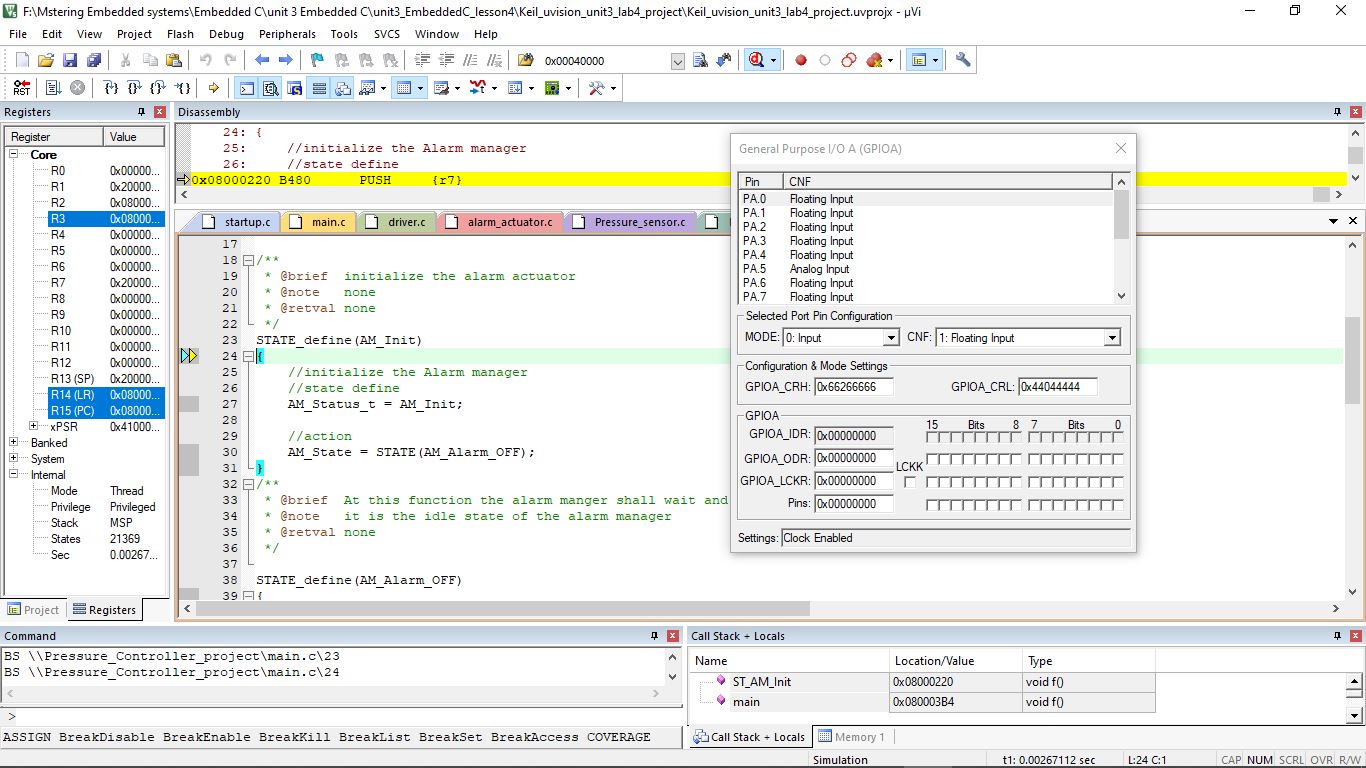
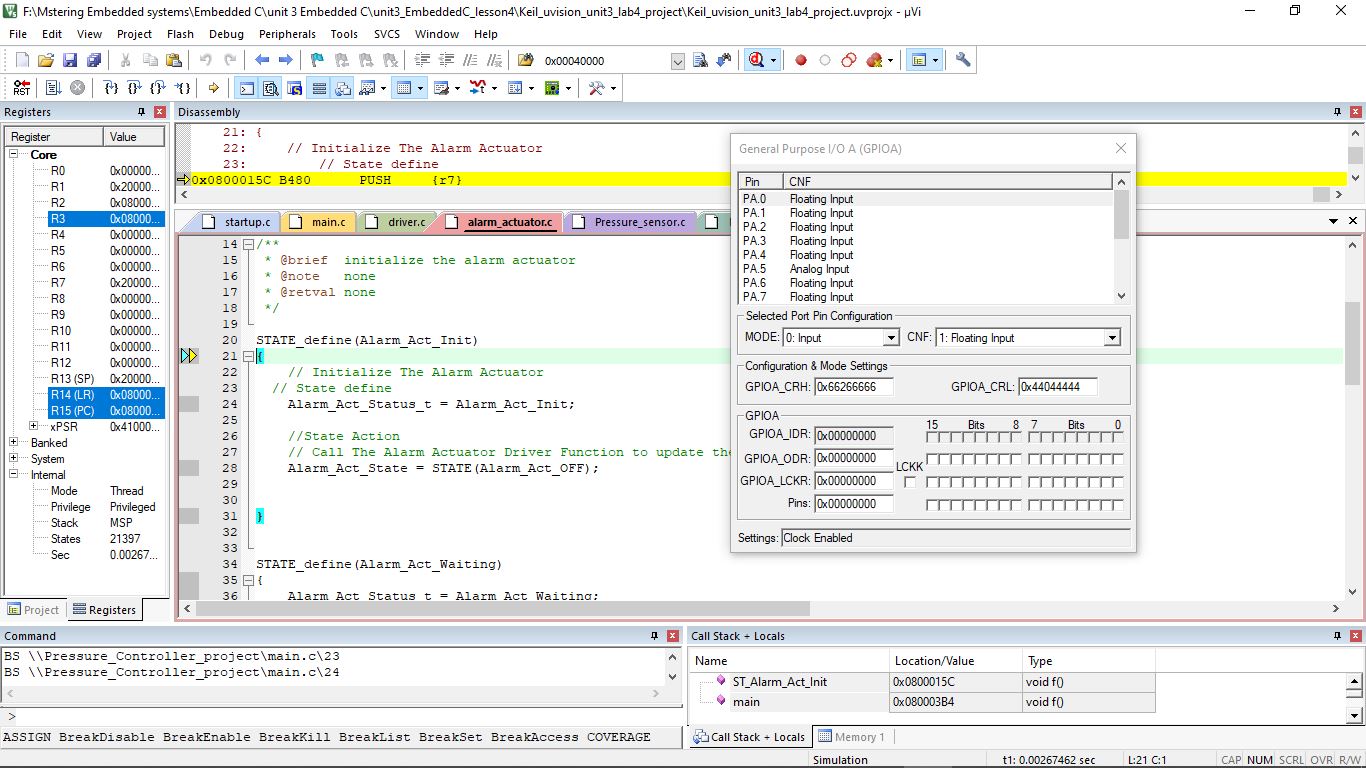
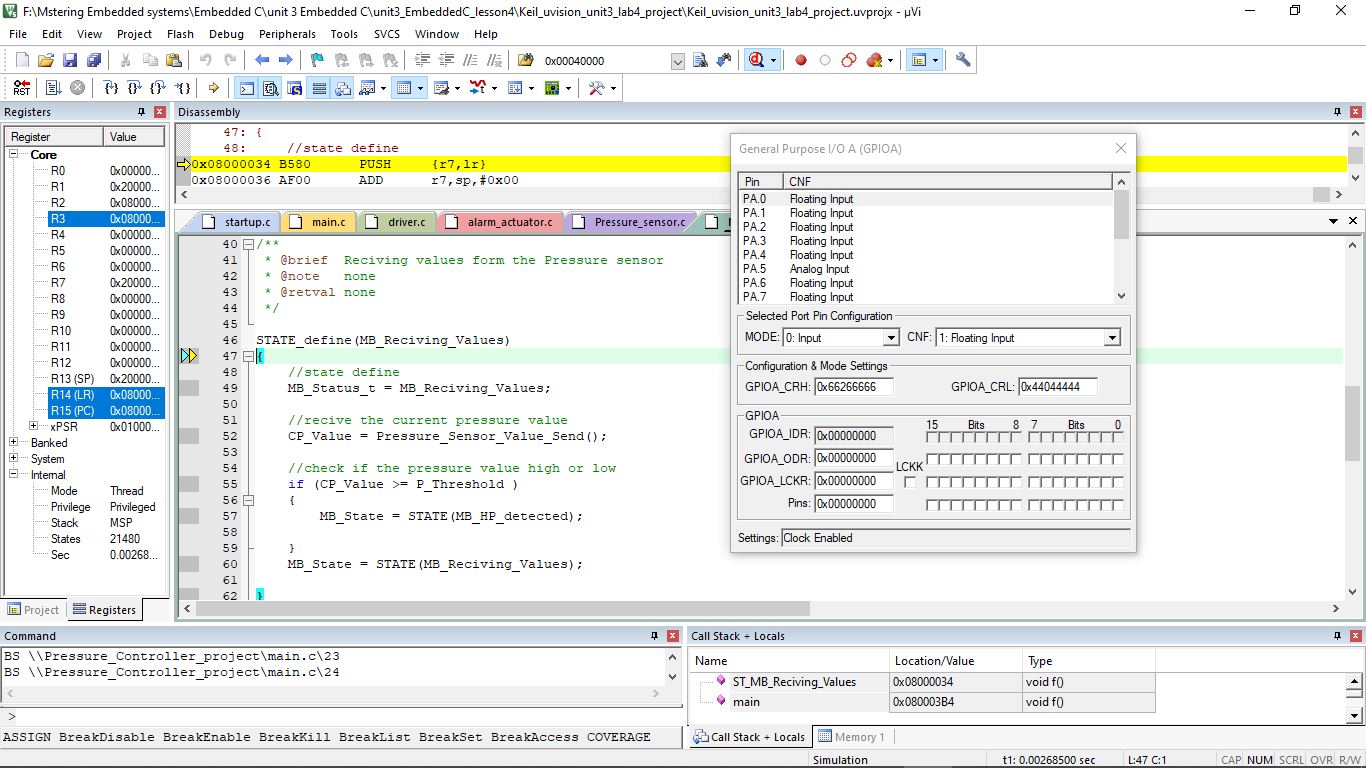
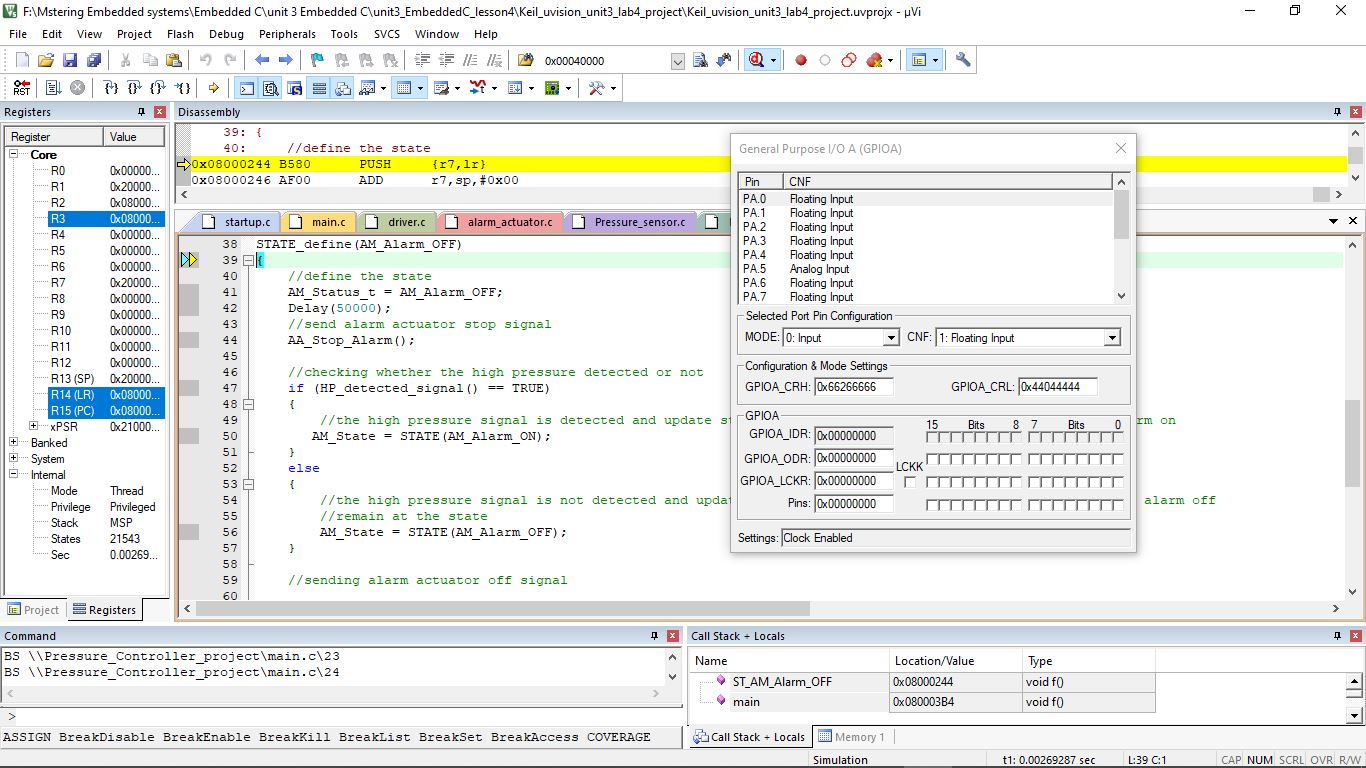
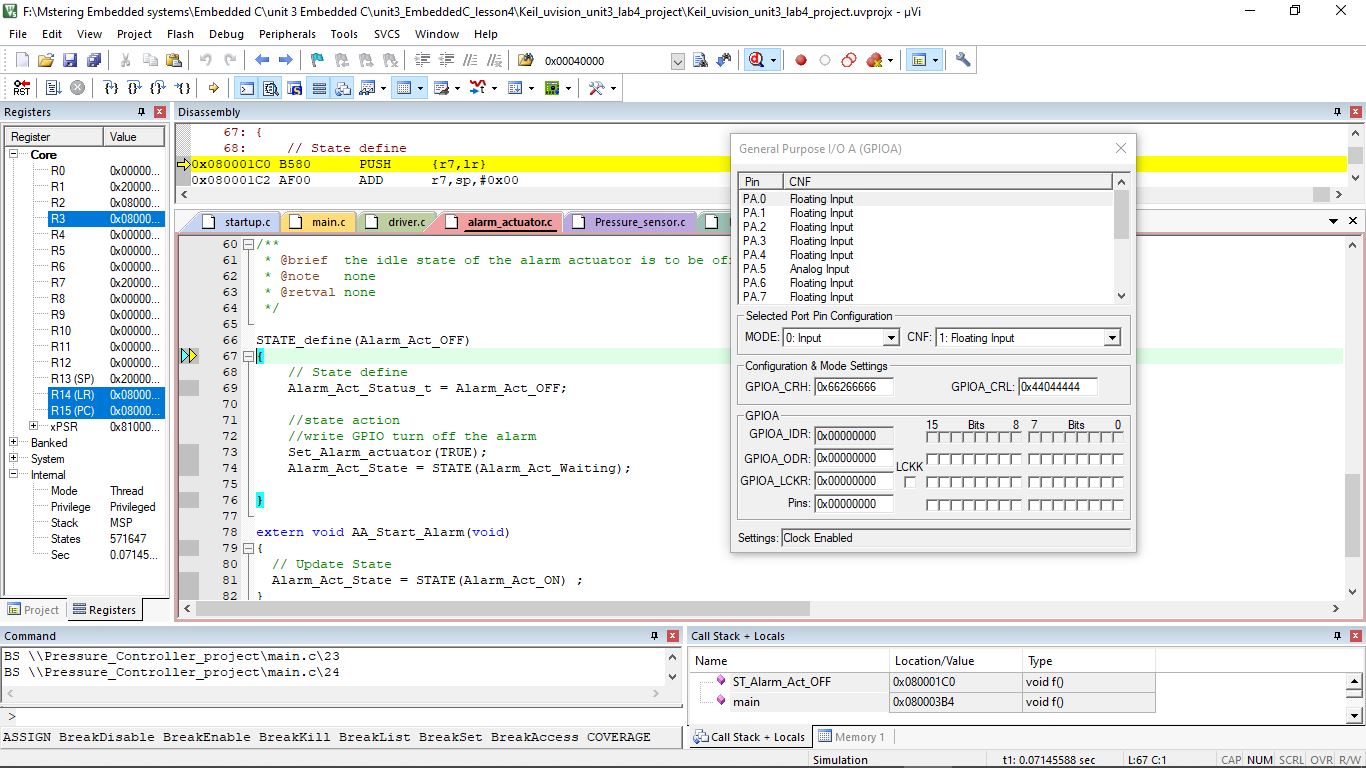
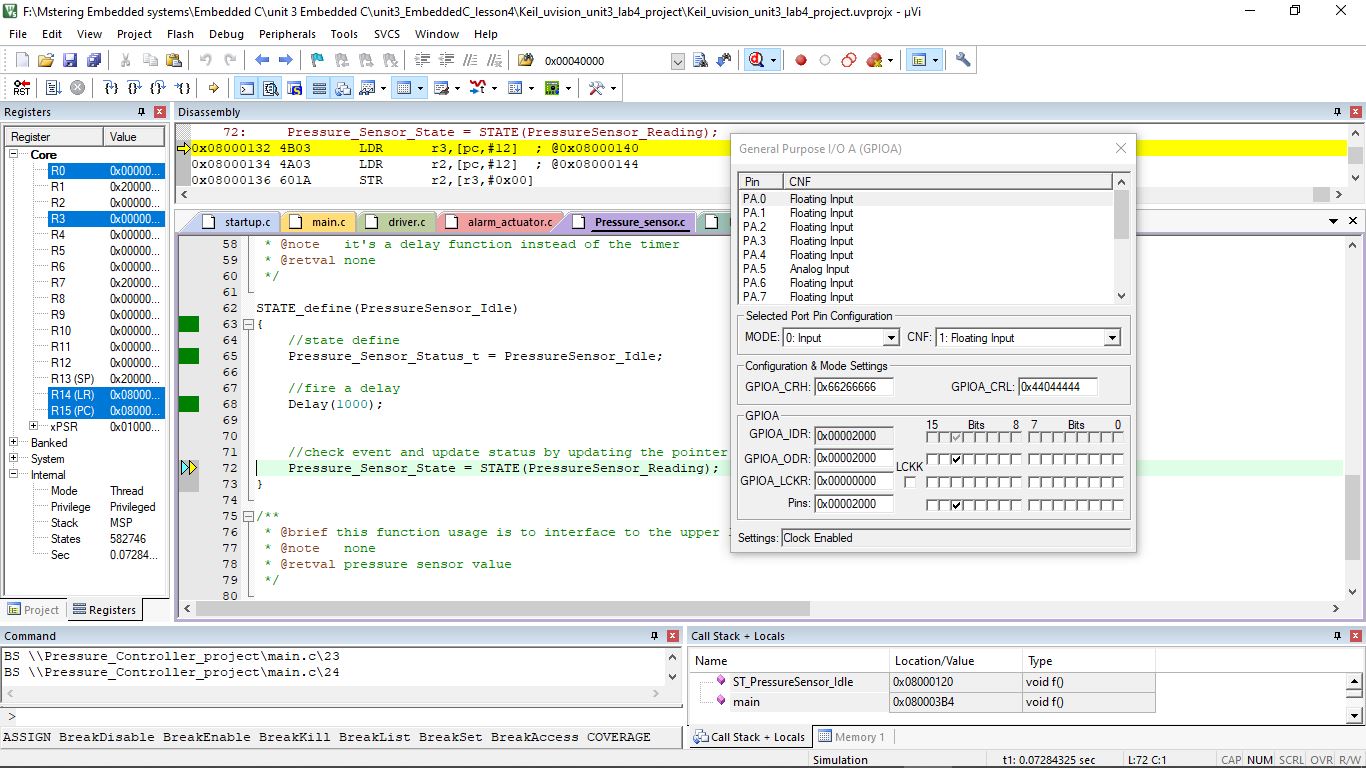
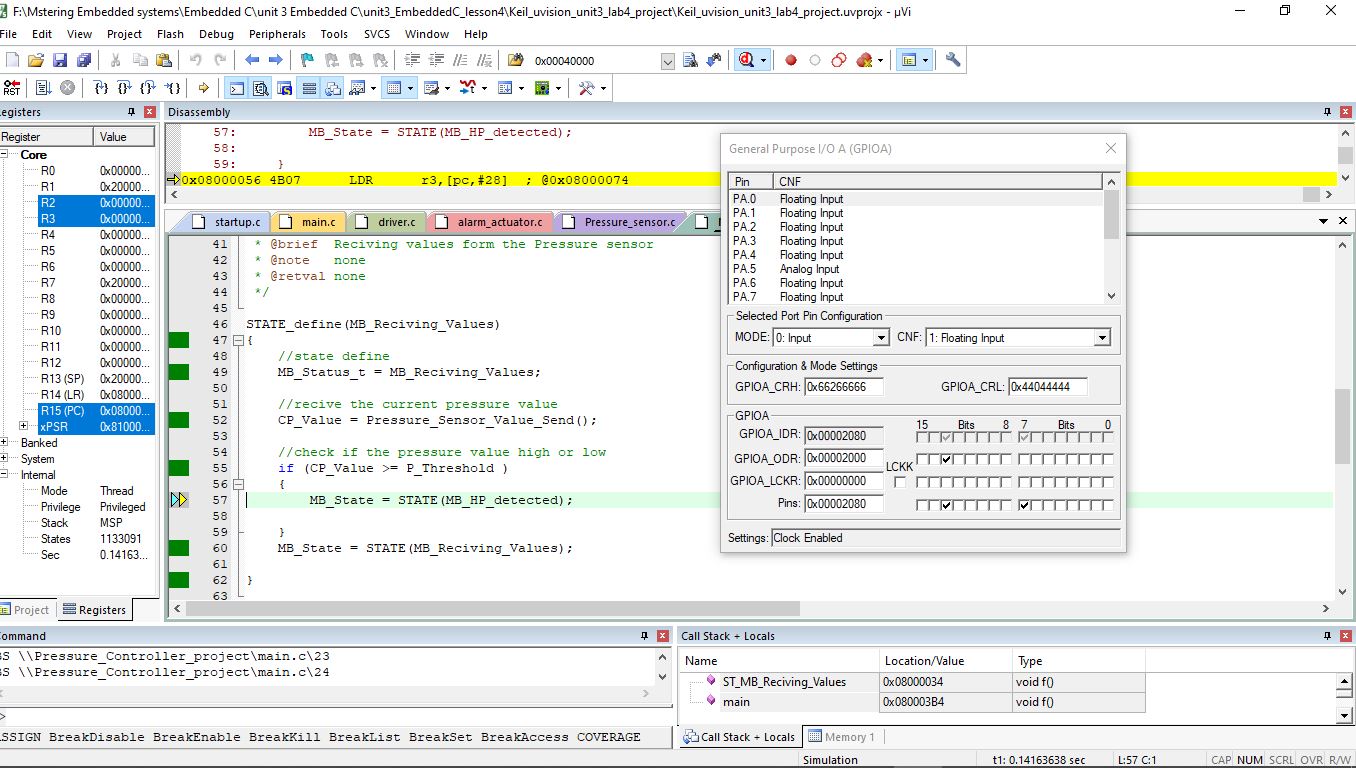
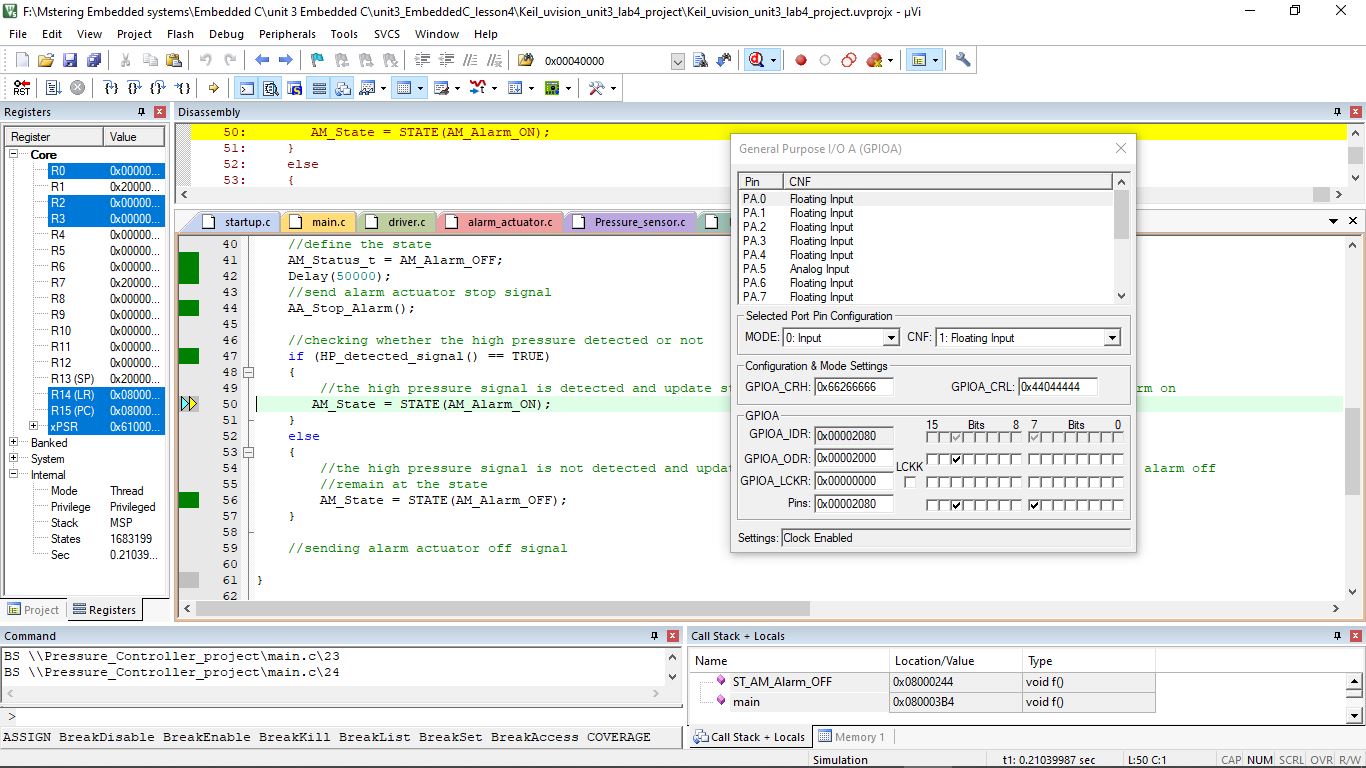
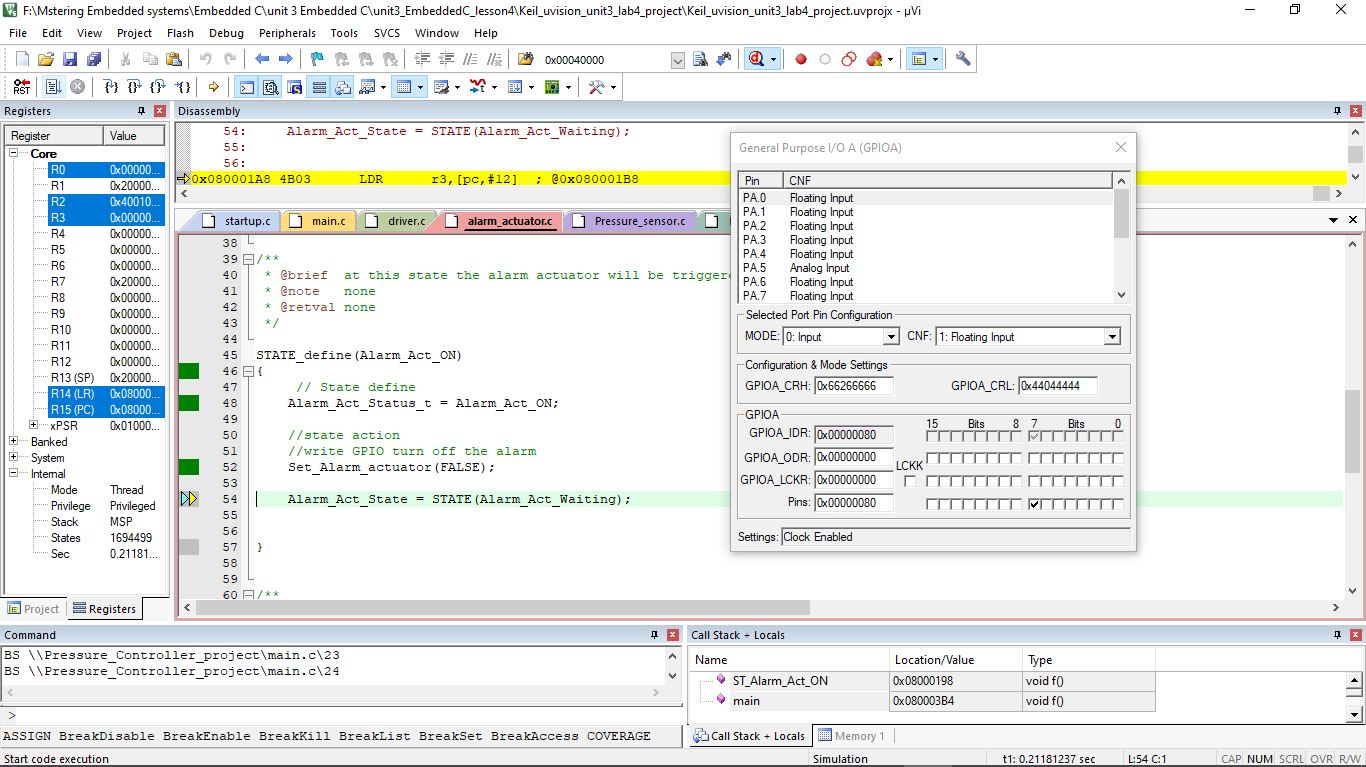
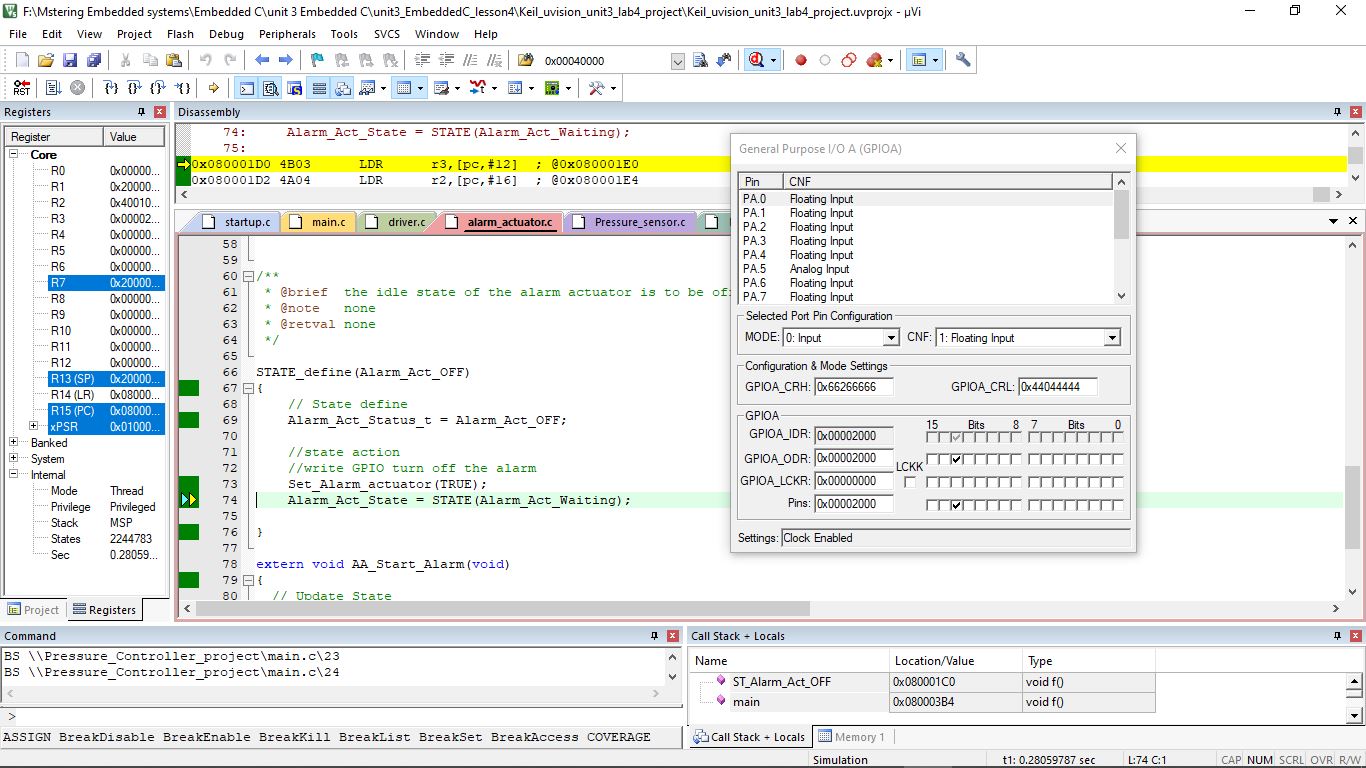
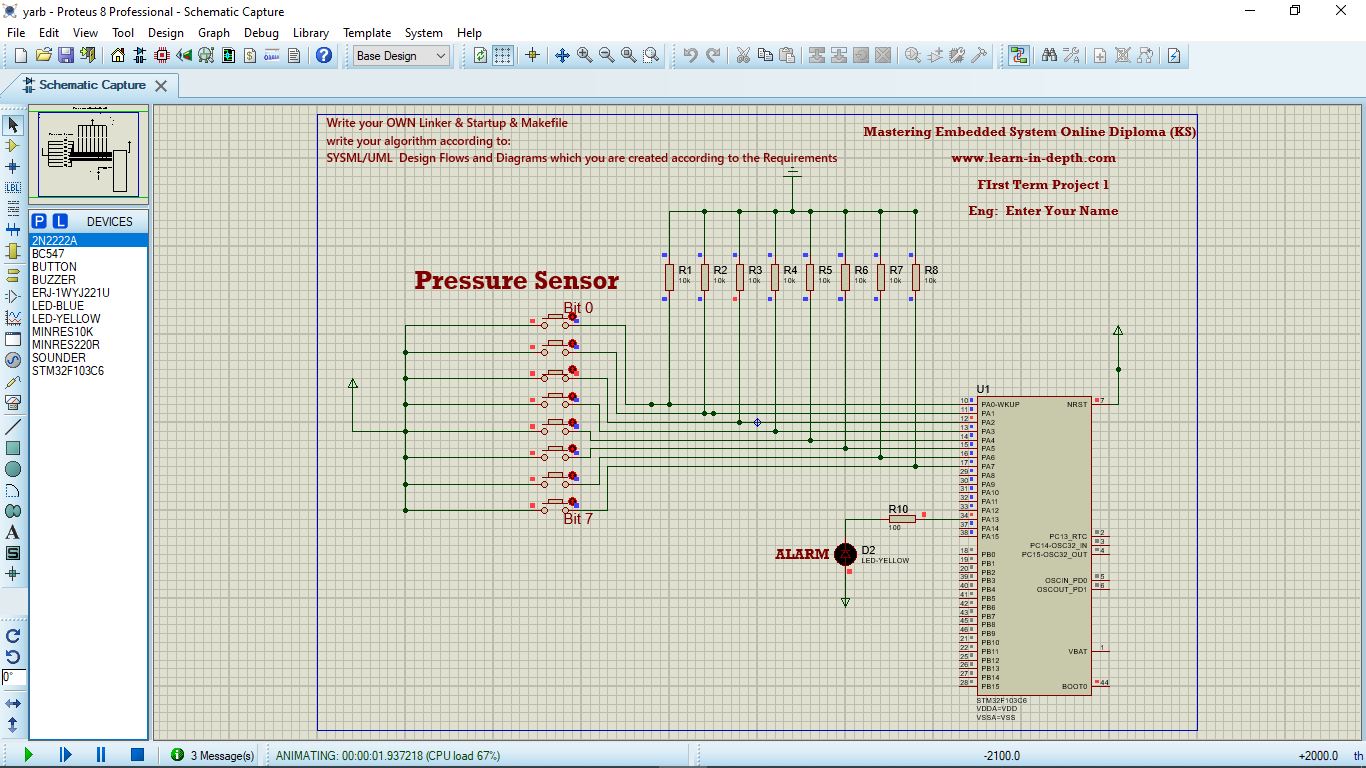
* Symbol table:
* Map file:



A close-up of a computer code

Description automatically generated



* Debugging and testing using keil:
* Setup function:
* GPIO PortA initialization:
* Main Block initialization:
* Pressure sensor initialization:
* Pressure sensor initialization:
* Alarm actuator initialization:
* pressure sensor reading state
* main block receiving pressure values state:
* Alarm manager alarm off state:
* alarm actuator off state:
* pressure sensor idle state:
* main block detects high pressure:
* alarm manager received high pressure signal from the main block and get into alarm on state:
* alarm actuator turn on the alarm led:
* alarm actuator turn off the led after the end of the delay:
* Simulation:

