# 

Mastering Embedded System Online Diploma

www.learn-in-depth.com

First Term (Final Project 1)

Eng. Habeba Ahmed Elbaghdady

Table of Contents:

1. Case study.
2. System Requirements.
3. System Analysis.

* Use case diagram.
* Activity diagram.
* Sequence diagram.

1. System Design.

* Block diagram.
* State machine.
* Pressure Sensor Driver.
* Main Algorithm.
* Alarm Monitor.
* Alarm Actuator Driver.

1. Implementation.
   * Make File.
   * Startup.
   * Linker Script.
   * Cross Toolchain.
   * C Code.

* States.
* Drivers.
* Pressure sensor driver.
* Main Algorithm.
* Alarm Monitor.
* Alarm Actuator Driver.
* Main
  + Symbol Table.
  + Section Table.
  + Map file.
  + Hardware.

Case Study:

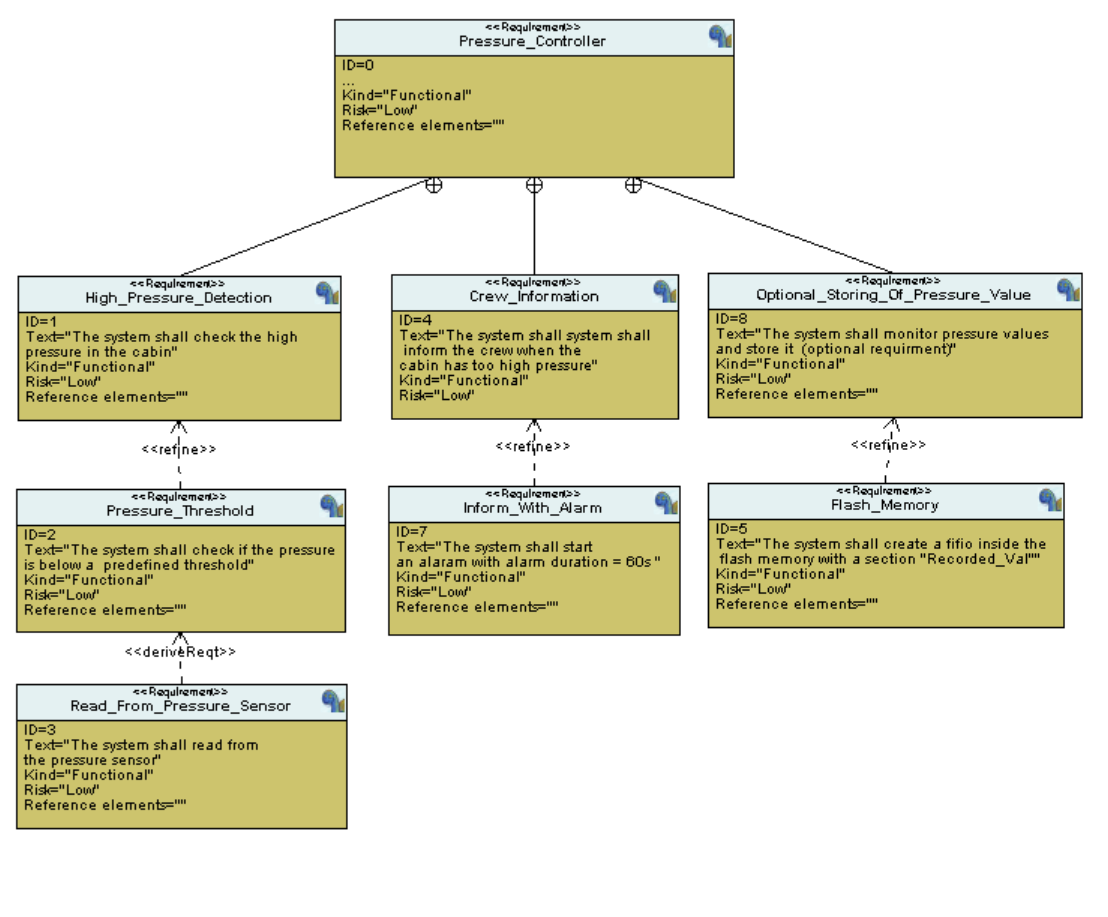
We have a pressure detection system in an airplane in which we have a sensor that measures the pressure and in case the pressure is more than 20 bar, an alarm is activated for a duration of 60 seconds to inform the crew of the airplane.

System Requirements:

We have some requirements in which the system shall do.

There is a part of the requirements which is storing the pressure values in a flash memory.

This requirement is not implemented in the first release of the system.



System Analysis:

The system analysis includes the use-case diagram , activity diagram, and sequence diagram to show the interactions between actors and our system.

1. Use-Case Diagram

It shows our actors which are pressure sensor, alarm actuator and storage flash and how they act with our system.

The system has a main algorithm that takes the pressure sensor values and checks if it exceeds our threshold or not, if it does, the main algorithm makes the alarm work to inform the crew.

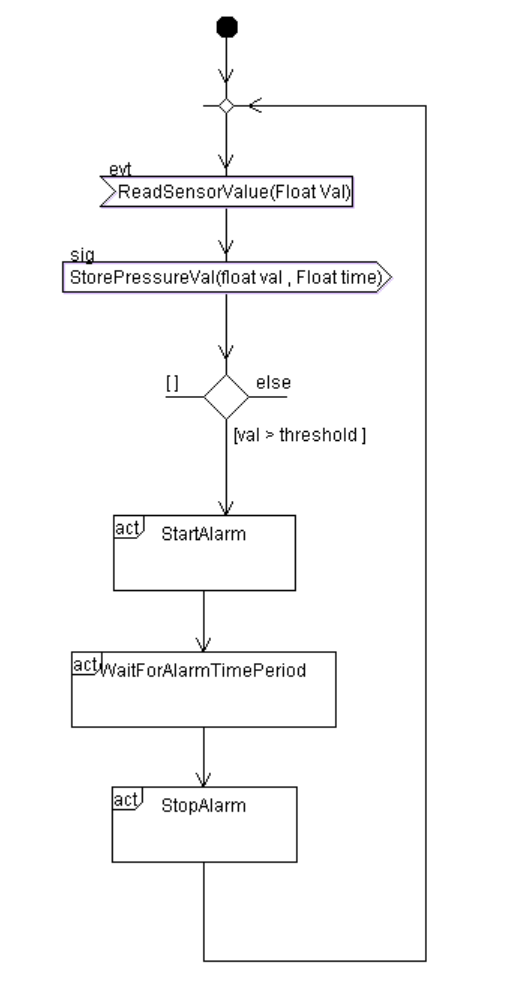
The storing of the pressure sensors values will be in the extended version of our system.

A diagram of a system

Description automatically generated

1. Activity Diagram:

The system waits for the event of sensor reading that gets the pressure value form the sensor, it stores the value of the sensor and checks if the value of the pressure sensor is more than 20, if it does, it starts an alarm with a duration of 60 seconds to inform the crew and stops the alarm.

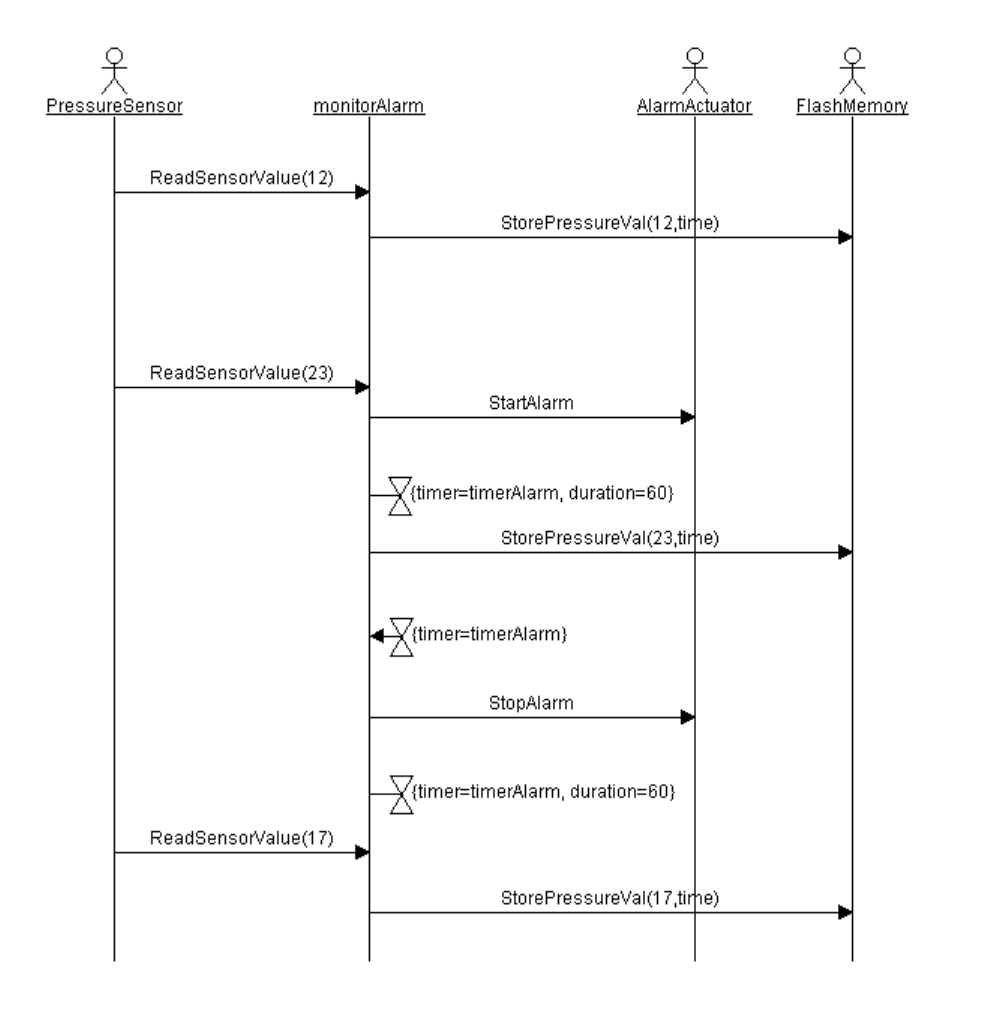


1. Sequence Diagram:

In this diagram, we have different interactions between the actors of our system.

Pressure sensor reads the first value which is 12, this value is less than the threshold, so the main algorithm only stores the value and return to read from the sensor.

When the value read from the sensor is more than the threshold it starts the alarm for a duration of 60 seconds, and it stores the value.



System Design:

1. Block Diagram:

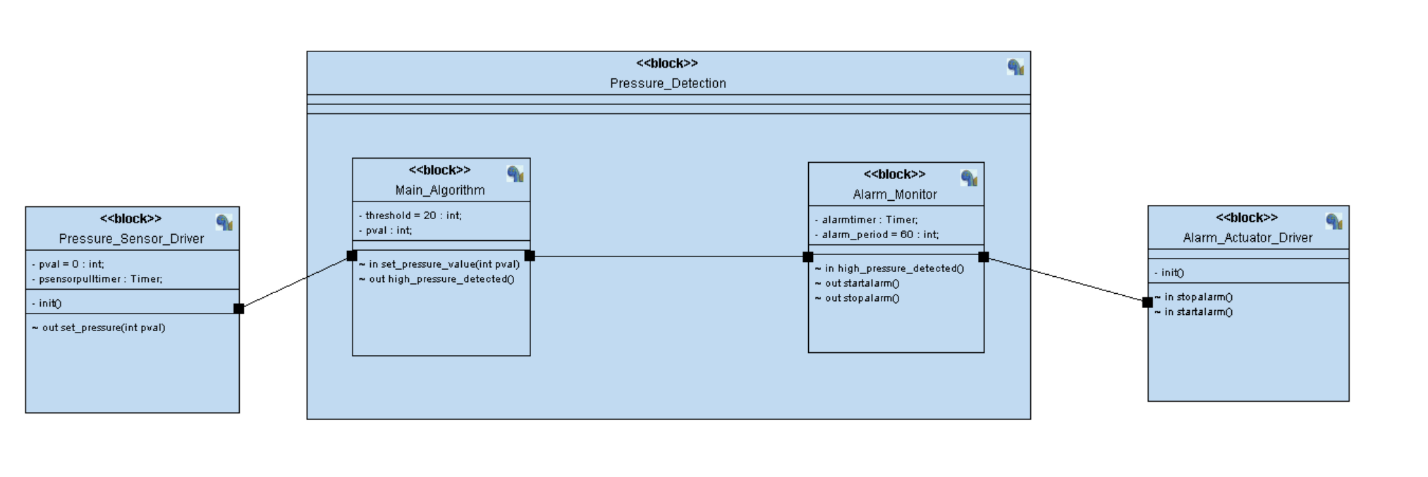
We have pressure sensor driver includes an init function and it measures the surrounding pressure and send the signal with its value to the main algorithm.

The main block is the pressure detection block which contains the main algorithm and the alarm monitor.

Main Algorithm takes the pressure value from the pressure sensor and compare it with the threshold if it is more than the threshold it sends a signal that there is high pressure detected to the alarm monitor.

Alarm monitor has the timer and its duration, it sends a signal of start alarm and stop alarm to the Alarm actuator driver.

Alarm actuator driver is used to start or stop the alarm according to the signal that came from the alarm monitor.



1. State Machine

* Pressure Sensor driver:

It is first initialized, and it starts reading a value from a random generator.

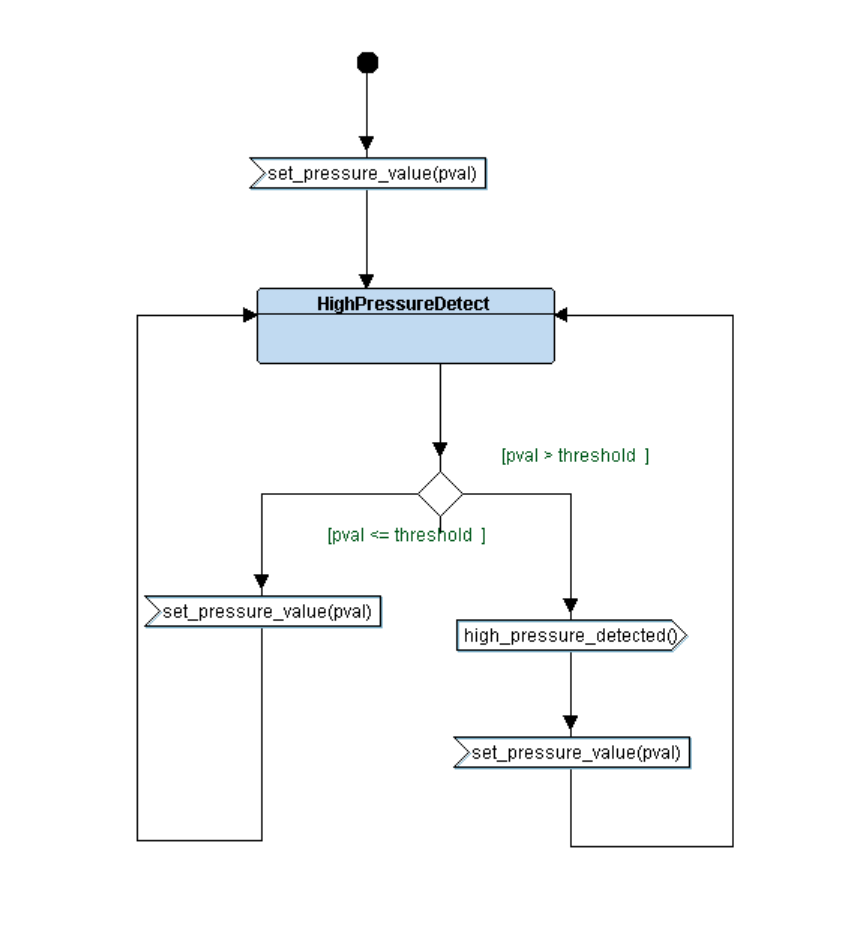
It sets the value of pressure and sends the signal to the main algorithm and

It starts a timer at which at the end of it rereads the pressure value.



* Main Algorithm

The main algorithm takes the value from the pressure sensor and compare the value with the threshold if it is greater than the threshold a signal of high detected pressure is sent to the alarm monitor, if it is less than it rereads the pressure again.

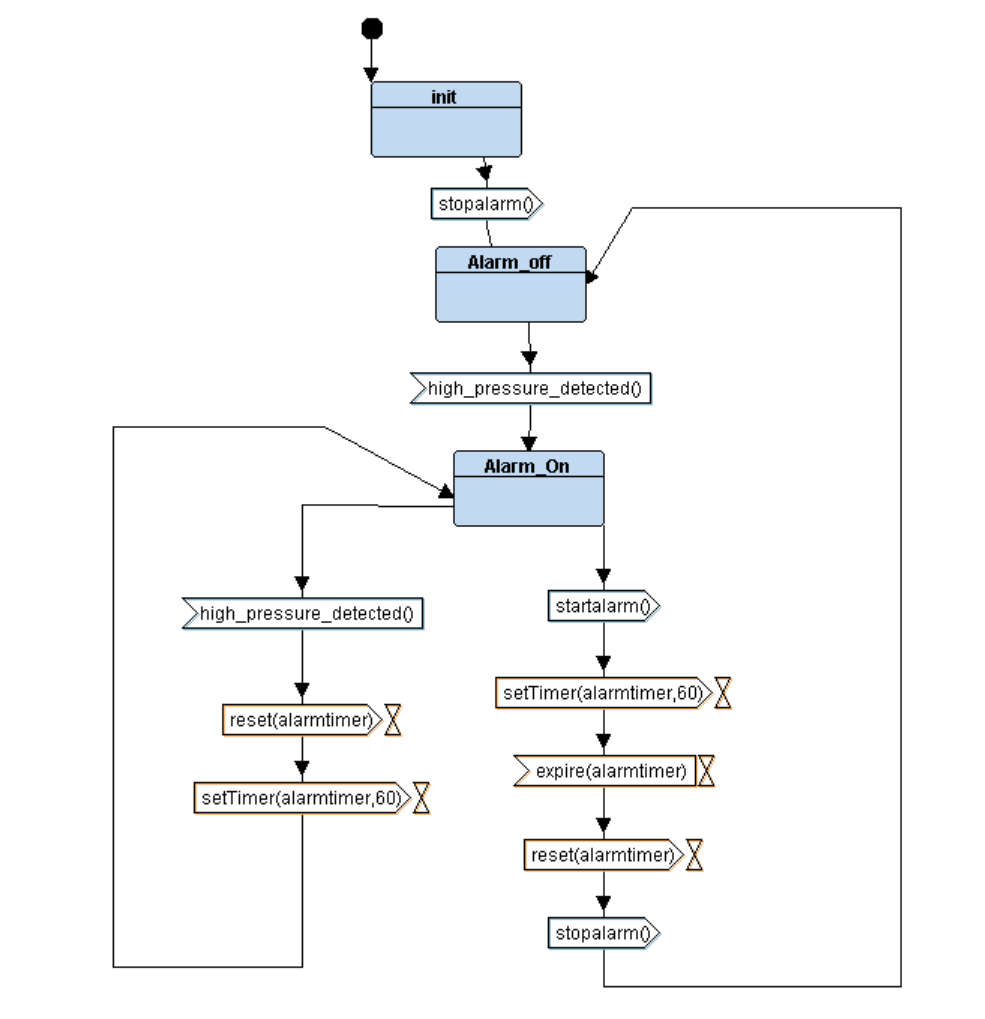


* Alarm Monitor

The Alarm Monitor initialized, and it stops the alarm.

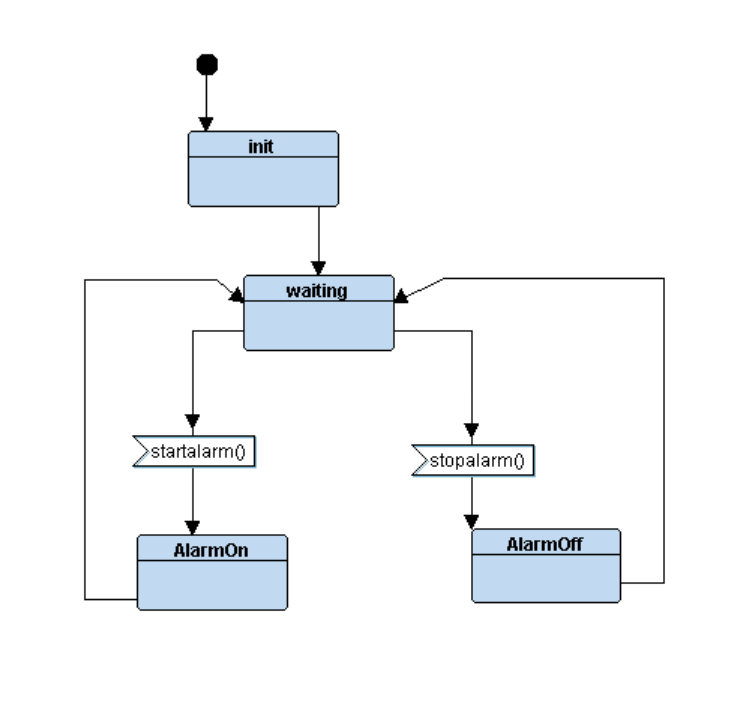
If the high detected pressure signal is sent the alarm is started with the duration of 60 seconds.

After the duration of the alarm is finished, it returns to alarm off, if another high-pressure signal is sent it reset the alarm timer and the alarm works for the specified duration.



* Alarm Actuator Driver:

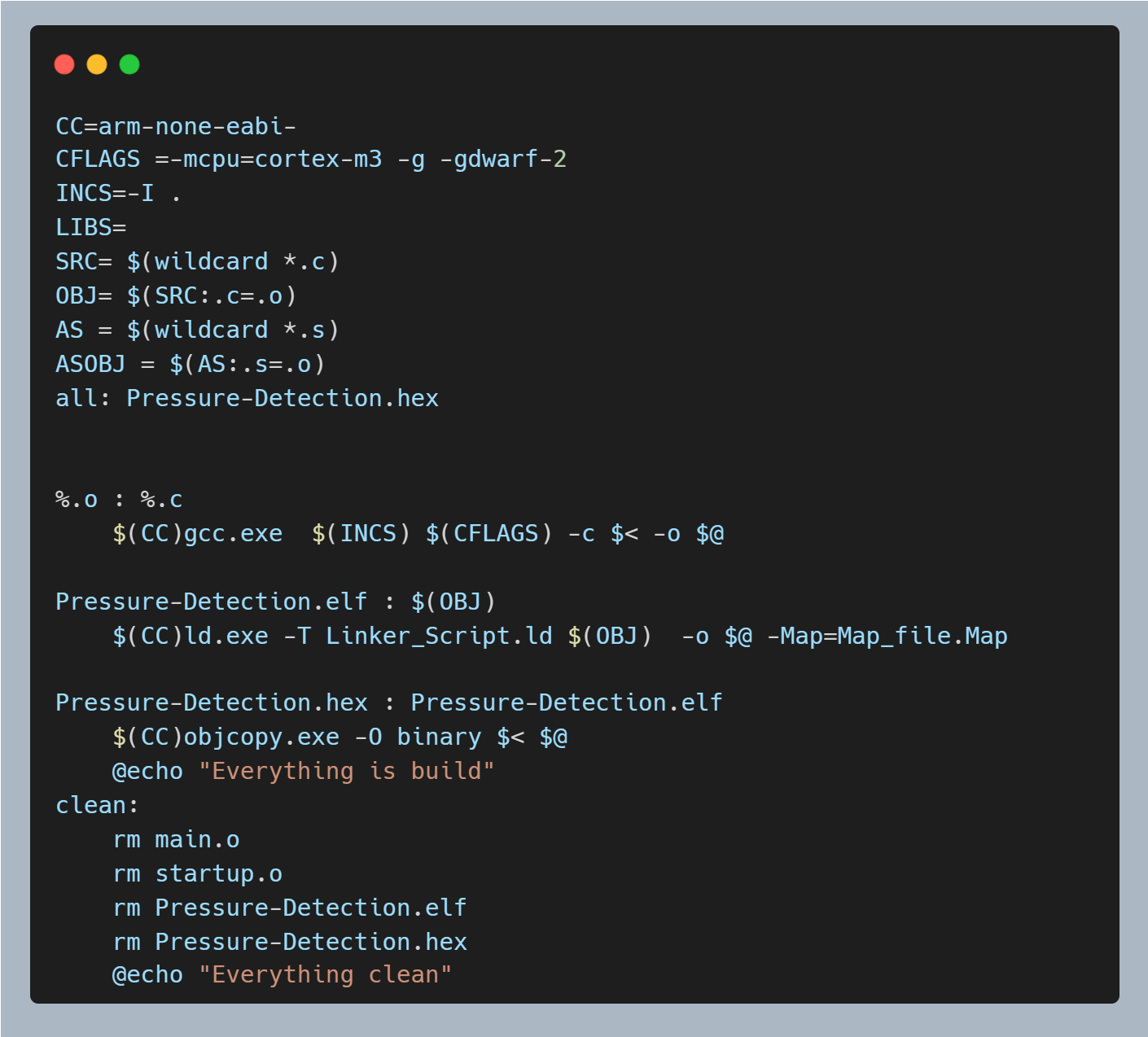
The Alarm Actuator is initialized and stays in the waiting state until a signal comes from the Alarm Monitor to start the alarm or stop it.



Implemenation

* 1. Make File

Make file is used to automate the build of our code files into an executable that can run on our board.



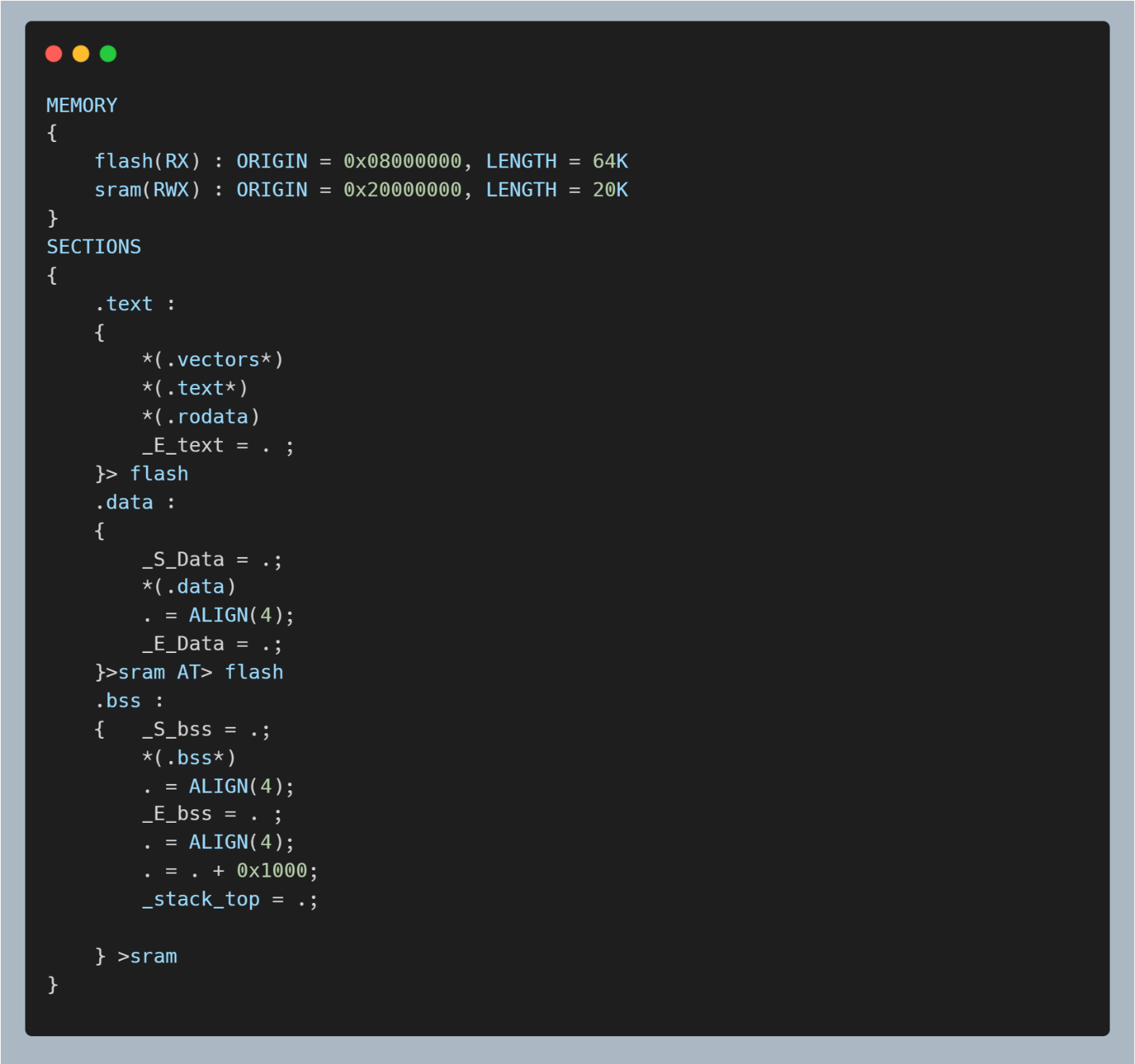
* 1. Startup

It is the file that run before the main when the board is in power on /Reset Mode.



* 1. Linker Script.

The linker script is created to link our code and store them in different sections and locates them in flash Memory or RAM.

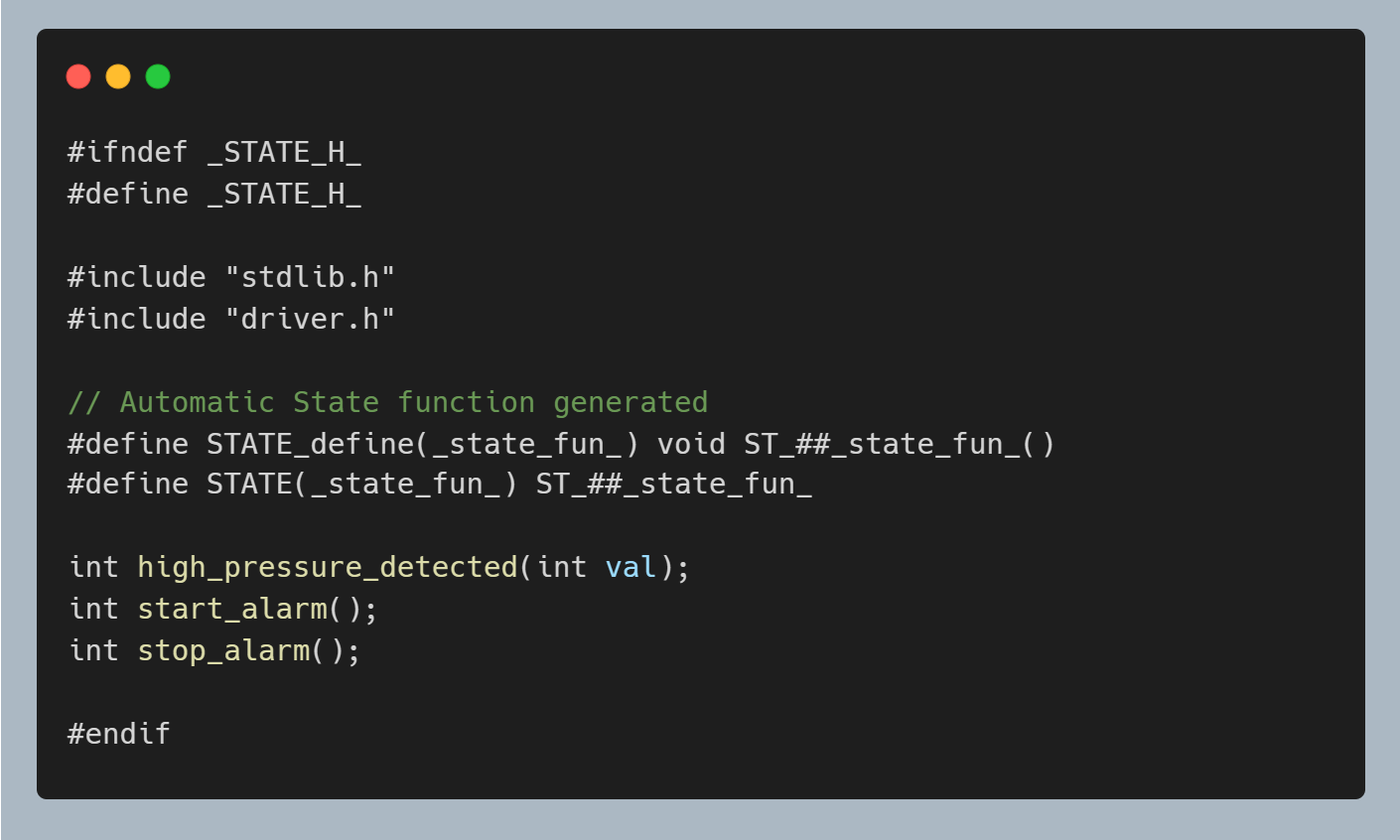


* 1. Cross Toolchain

We need arm toolchain downloaded on our host Machine to compile our code to the specifications of the target machine.

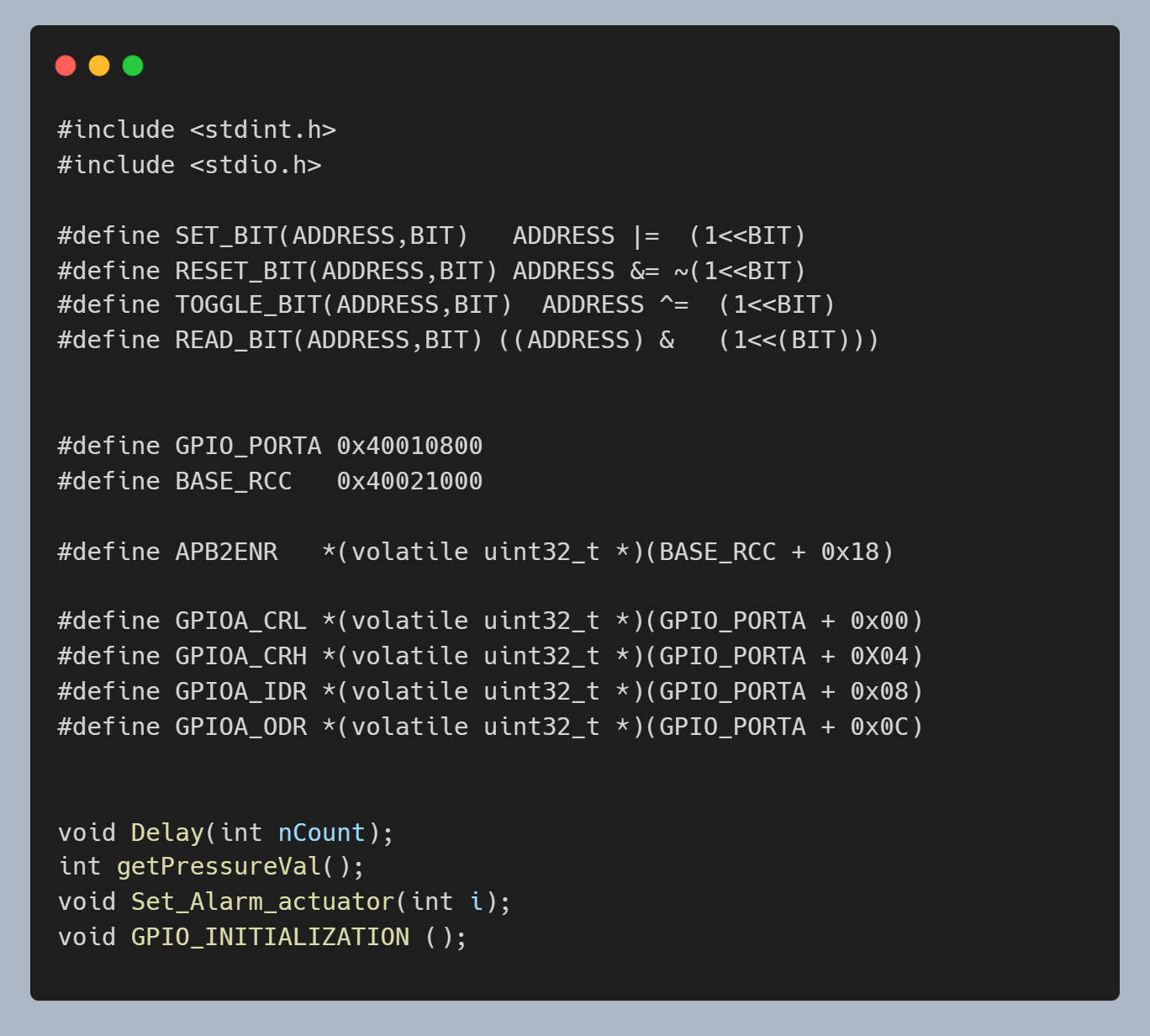
* 1. C Code.
* States

The State.h file is used to define any state machine and contains the signals sent from one block to another.



- Drivers

* Driver.h



* Driver.c

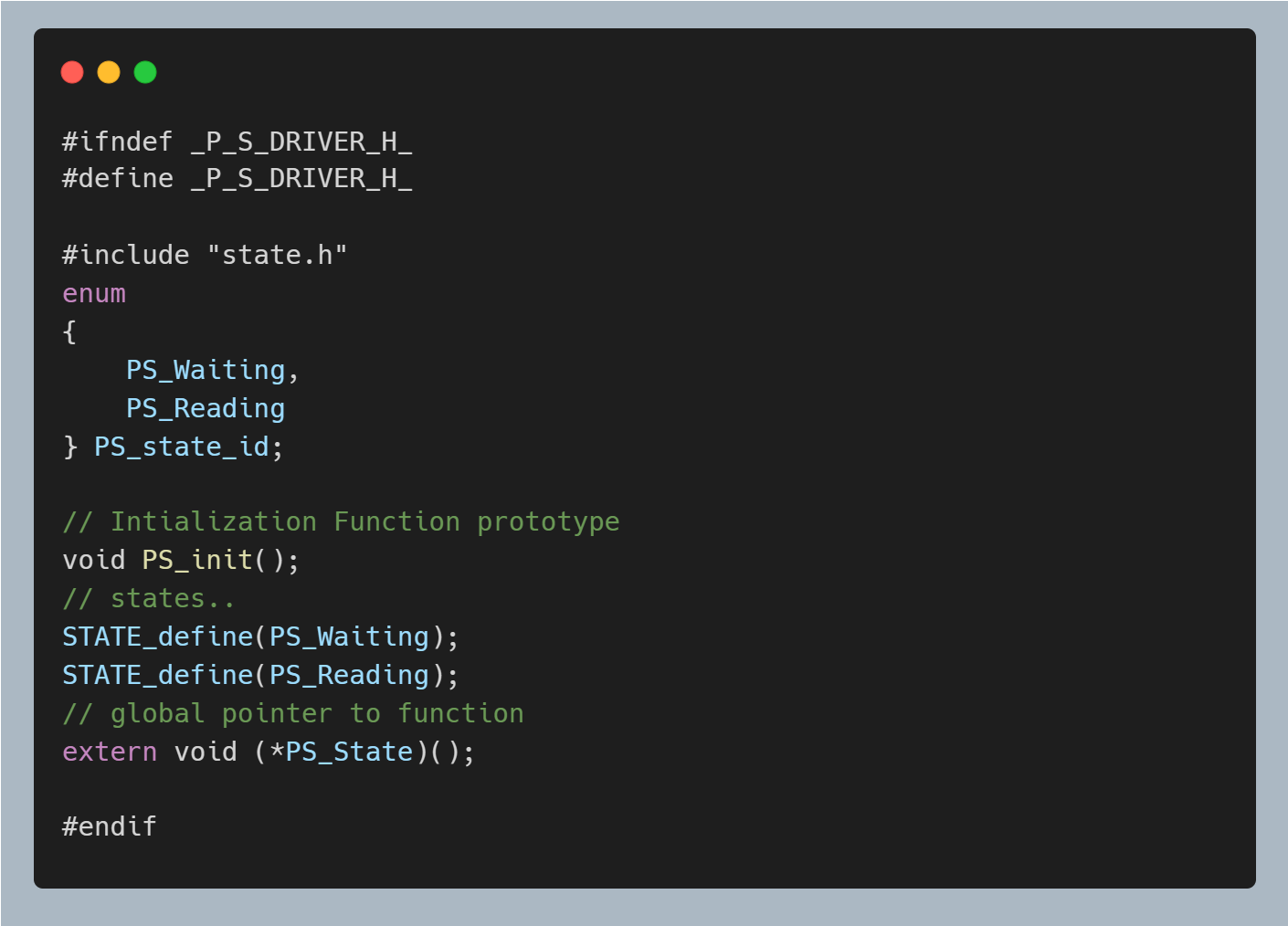
It has the definitions of

* 1. Delay function to be used as a delay for a few seconds.
  2. Set alarm actuator to set the led or reset it according to the pressure value.
  3. Gpio Initialization to initialize the gpio of the board.
  4. Get pressure value to get the value of the pressure according to the input.



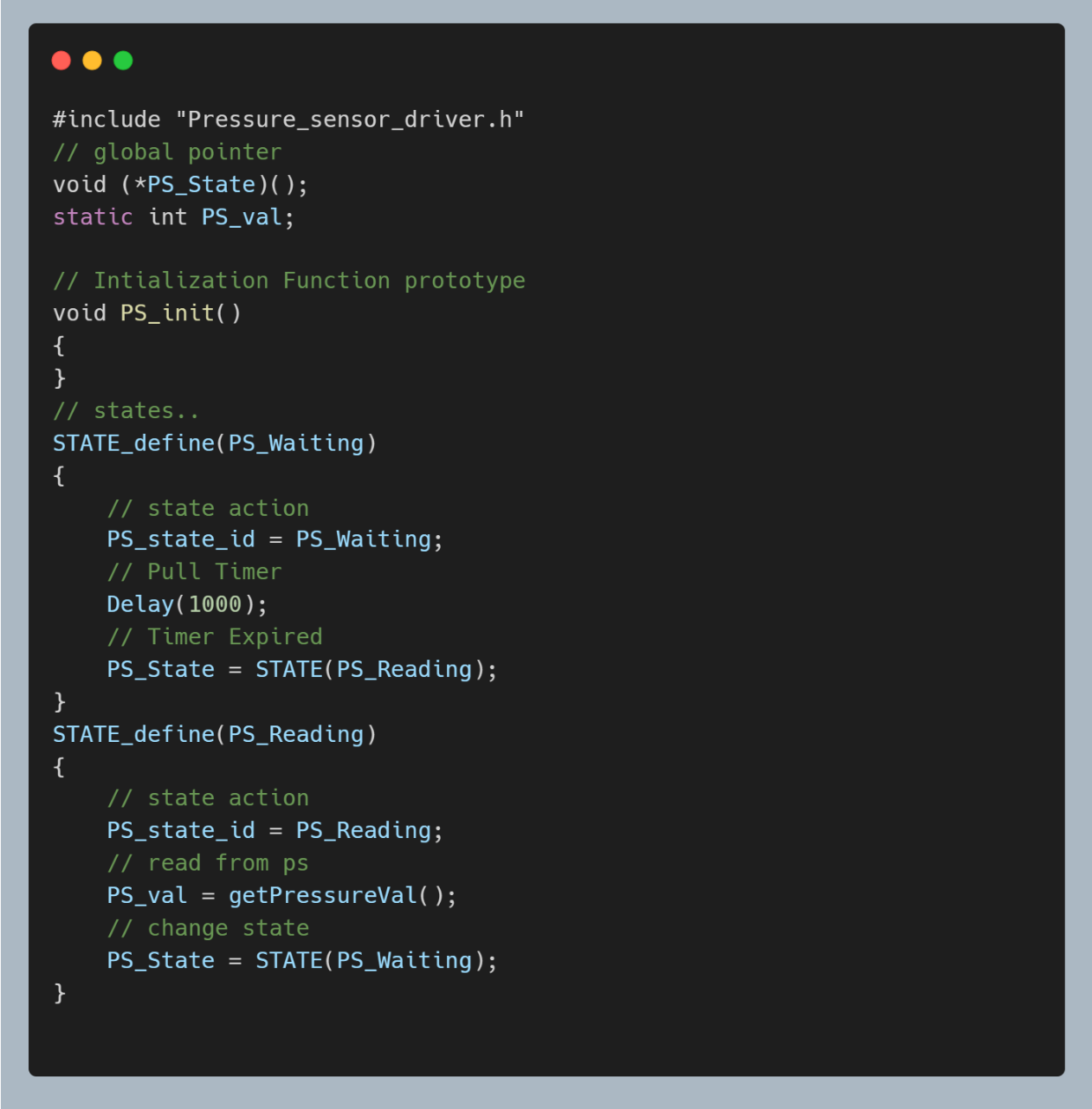
- Pressure sensor driver.

* Pressure\_sensor.h

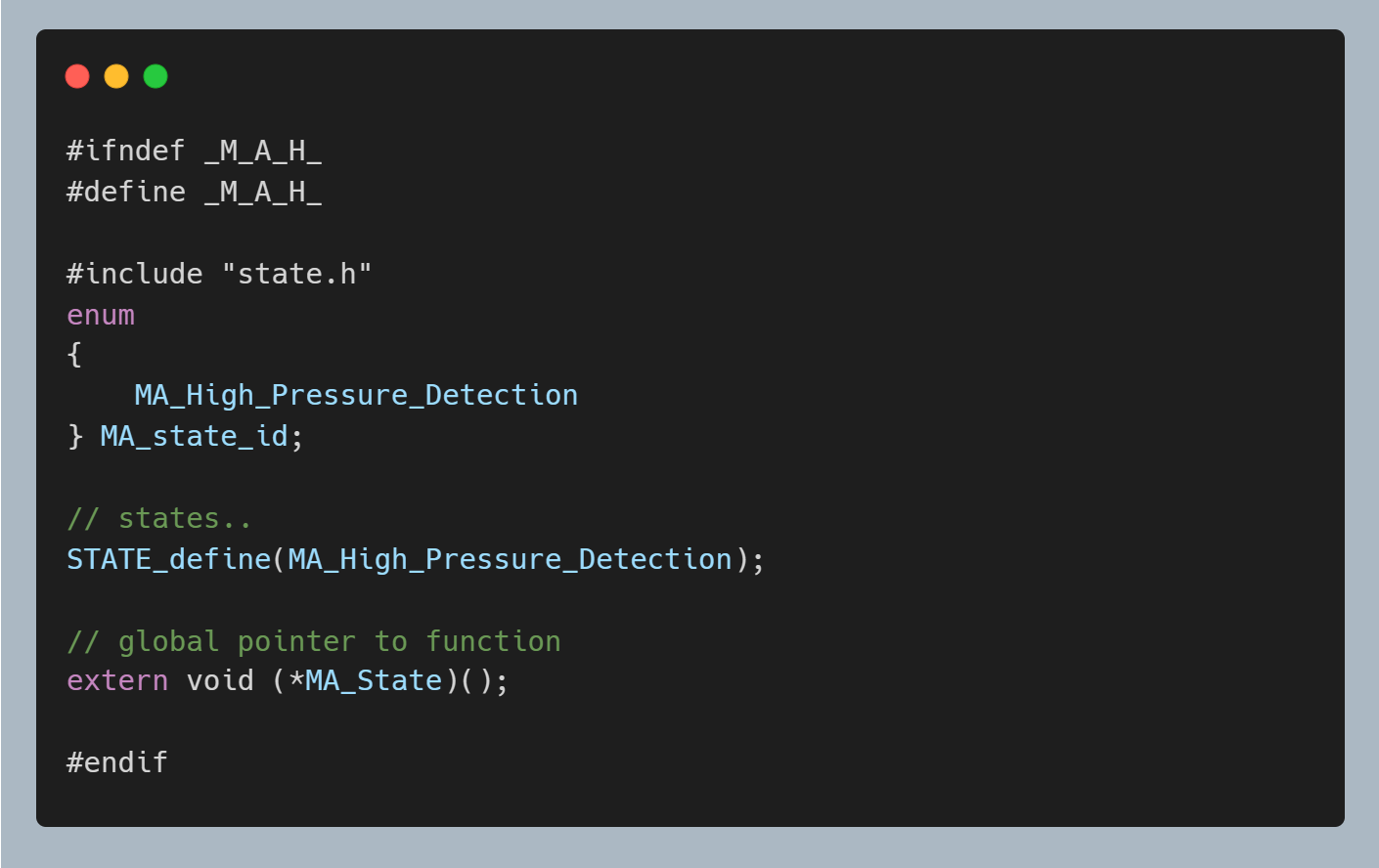


* Pressure\_sensor.c

It contains the state machines definitions.

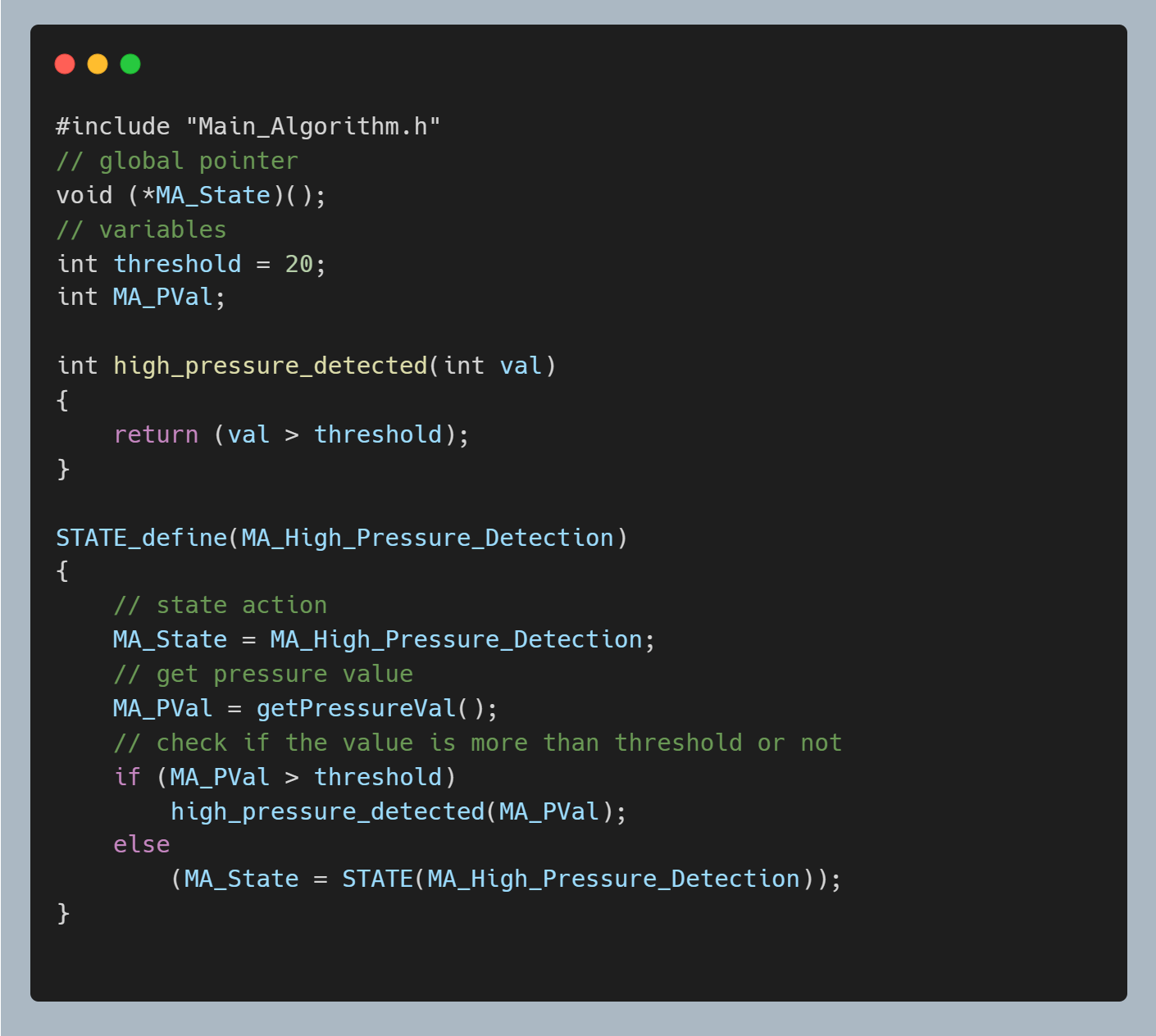


* Main Algorithm.
* Main\_Algorithm.h



* Main\_Algorithm.c

It includes the Implementation of high pressure detected function and the state that checks for high pressure.

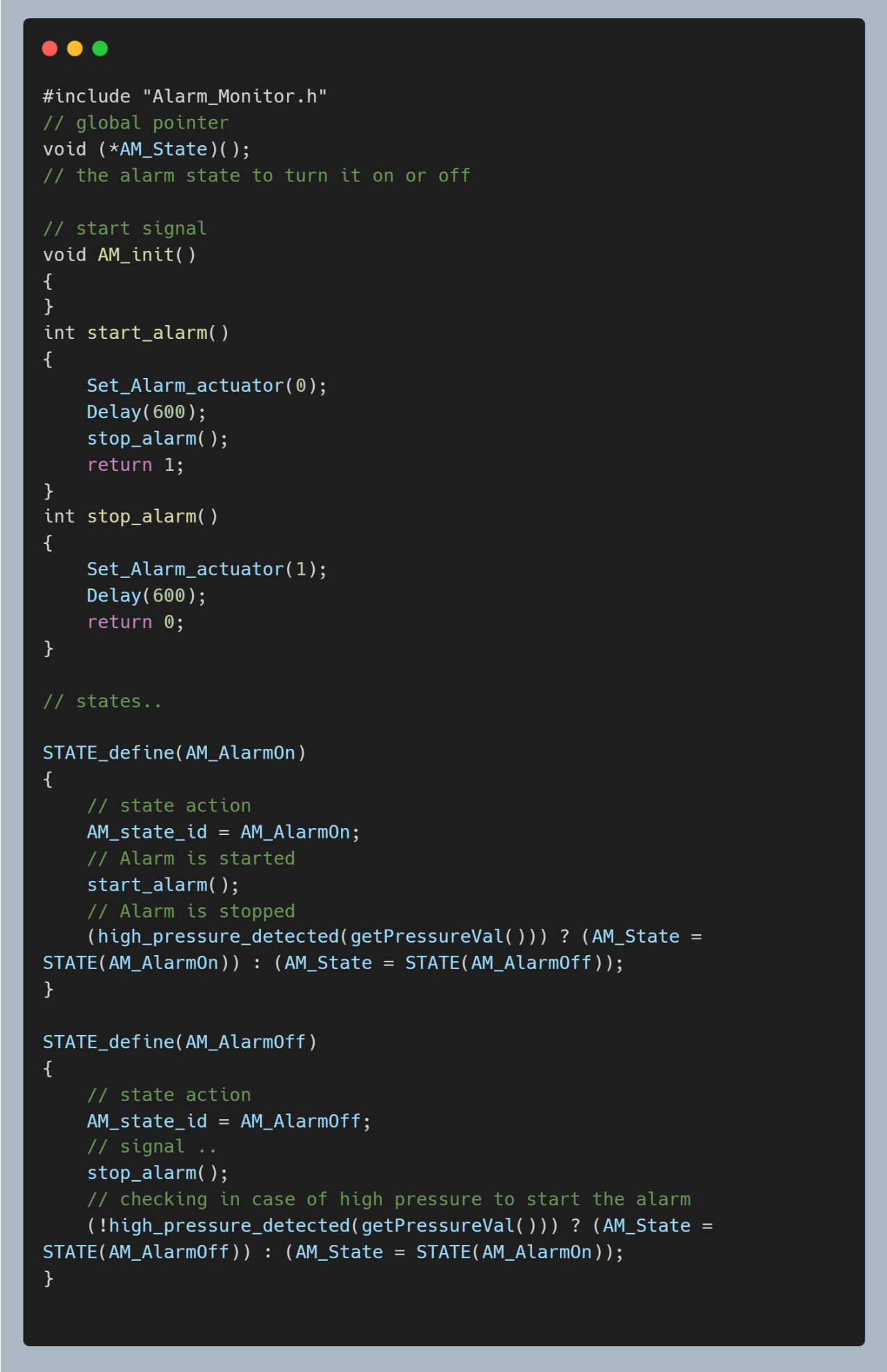


* Alarm Monitor.
* Alarm\_Monitor.h

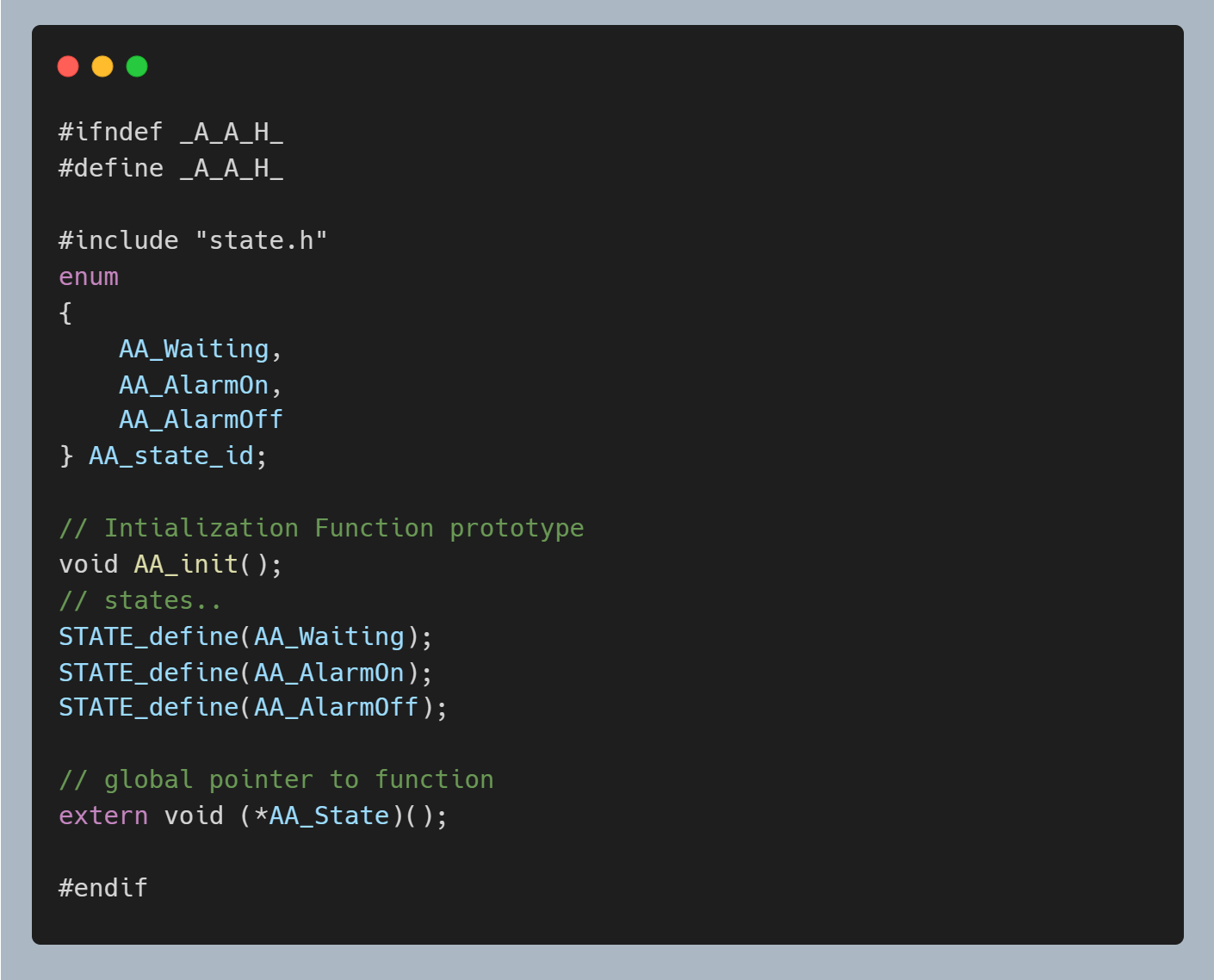


* Alarm\_Monitor.c

It implements the start alarm function where we set the led to work for a duration of 60 seconds and stop alarm function which stops the led again.

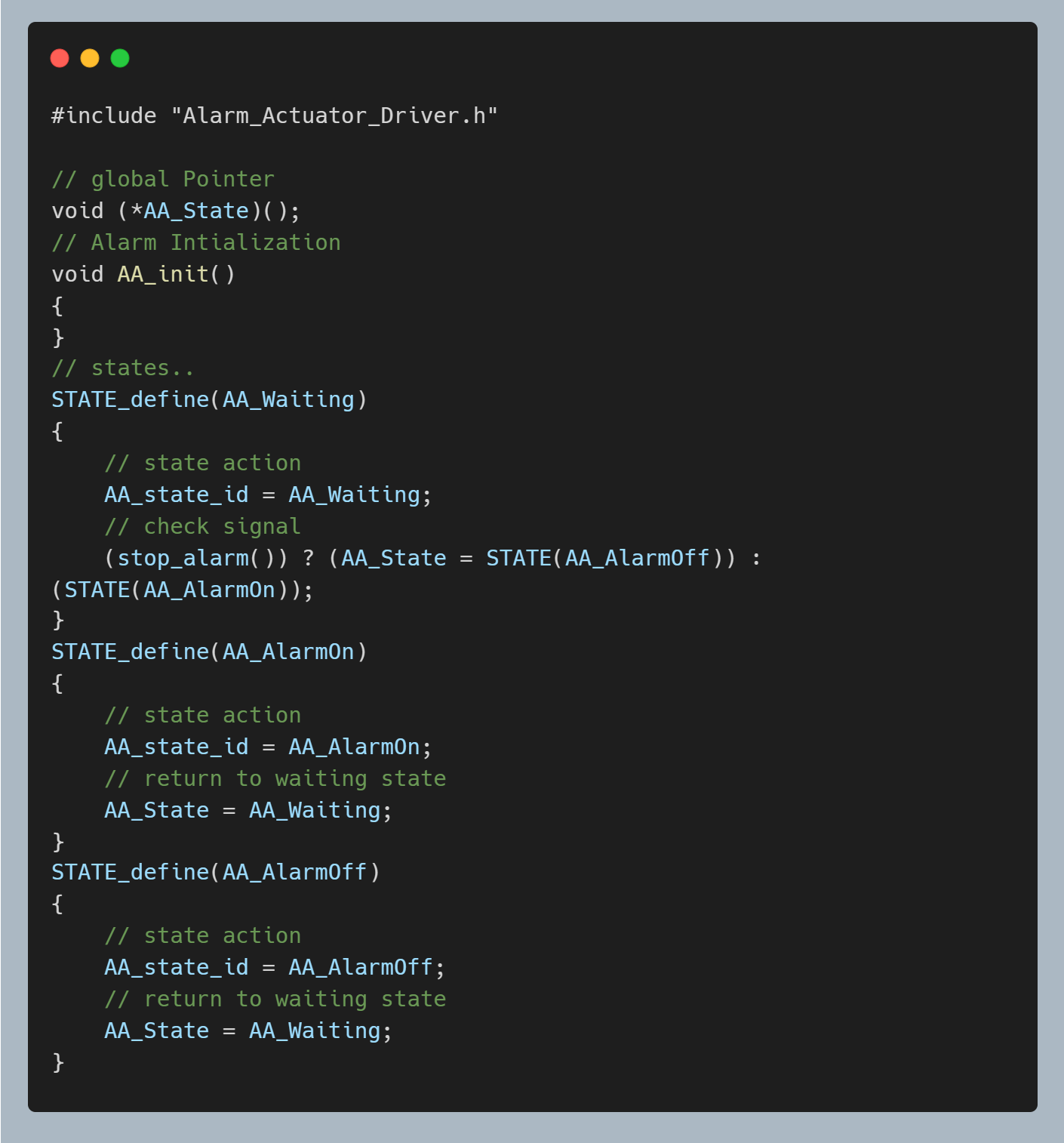


* Alarm Actuator Driver
* Alarm\_Actuator.h

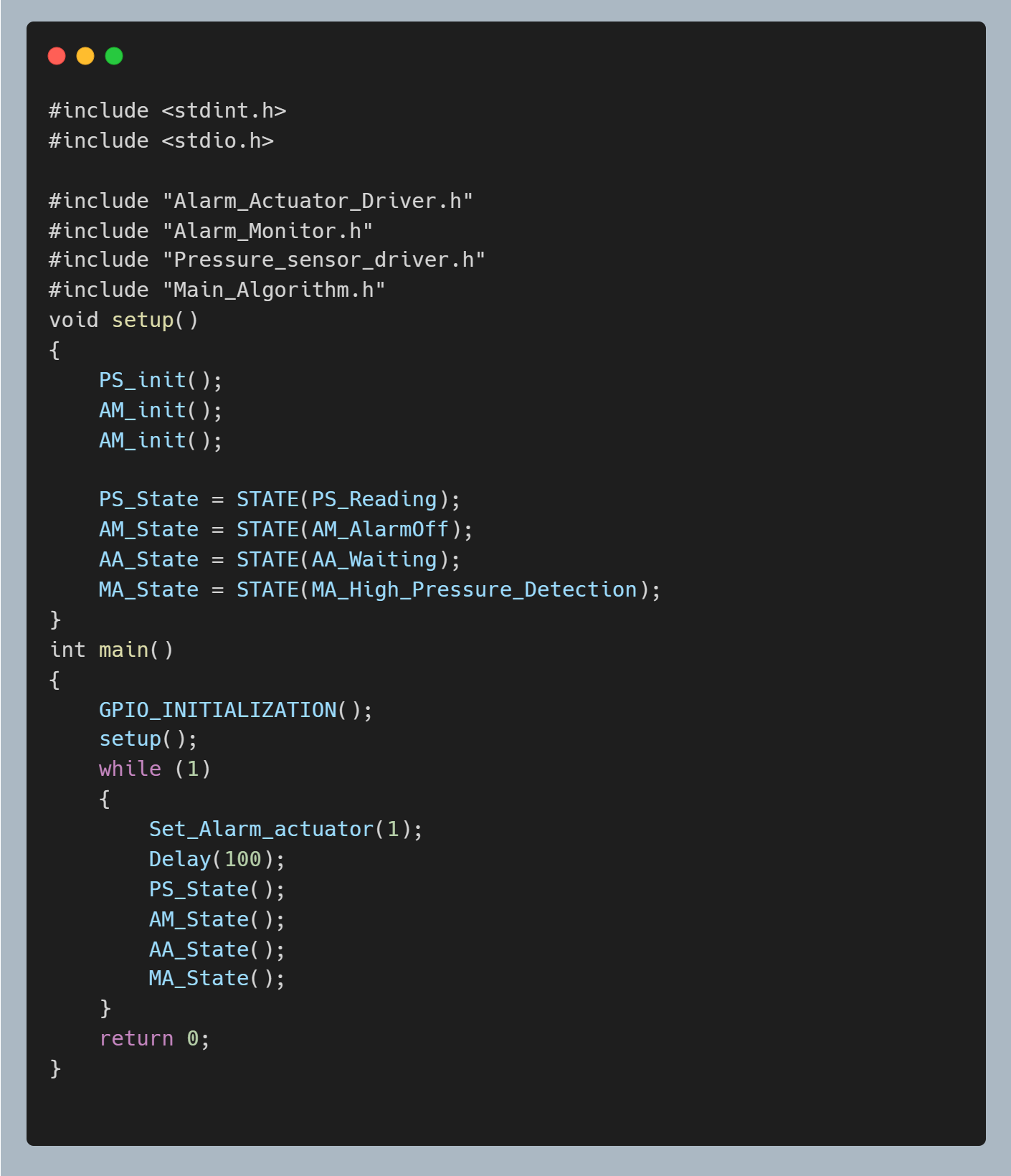


* Alarm\_Actuator.c

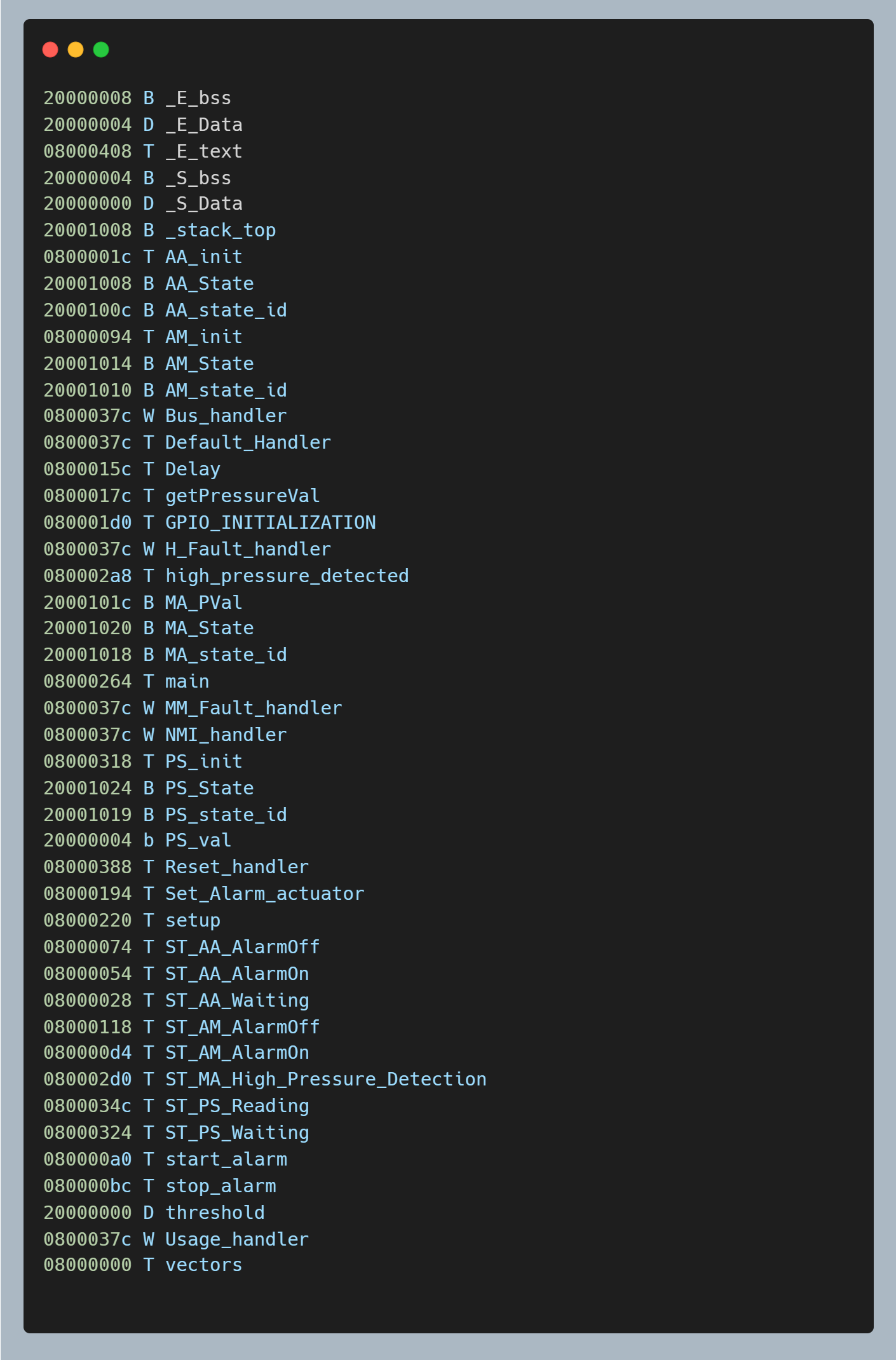
It implements the checking of the signal so it can stay in alarm off or go to alarm on.



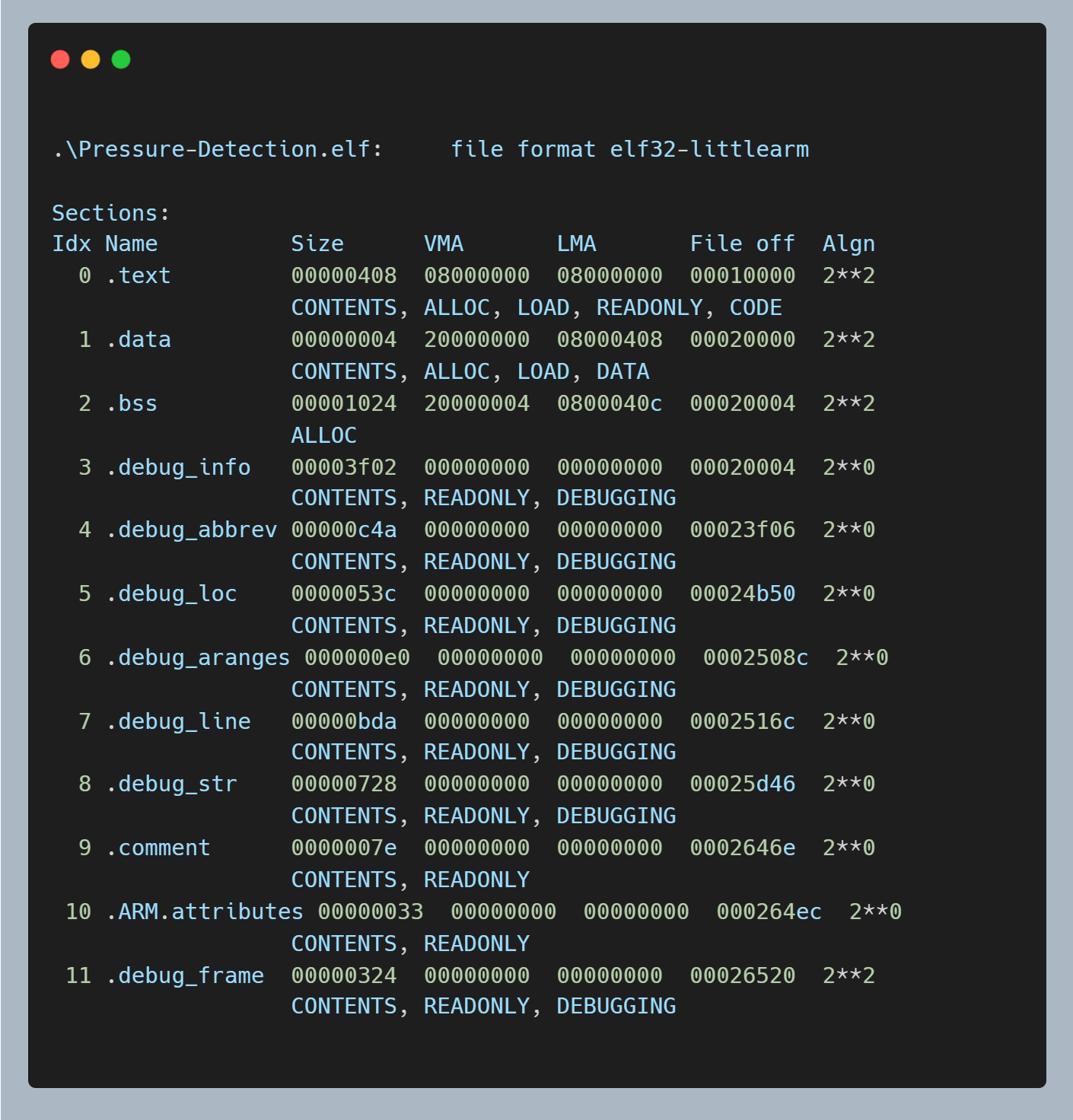
* Main



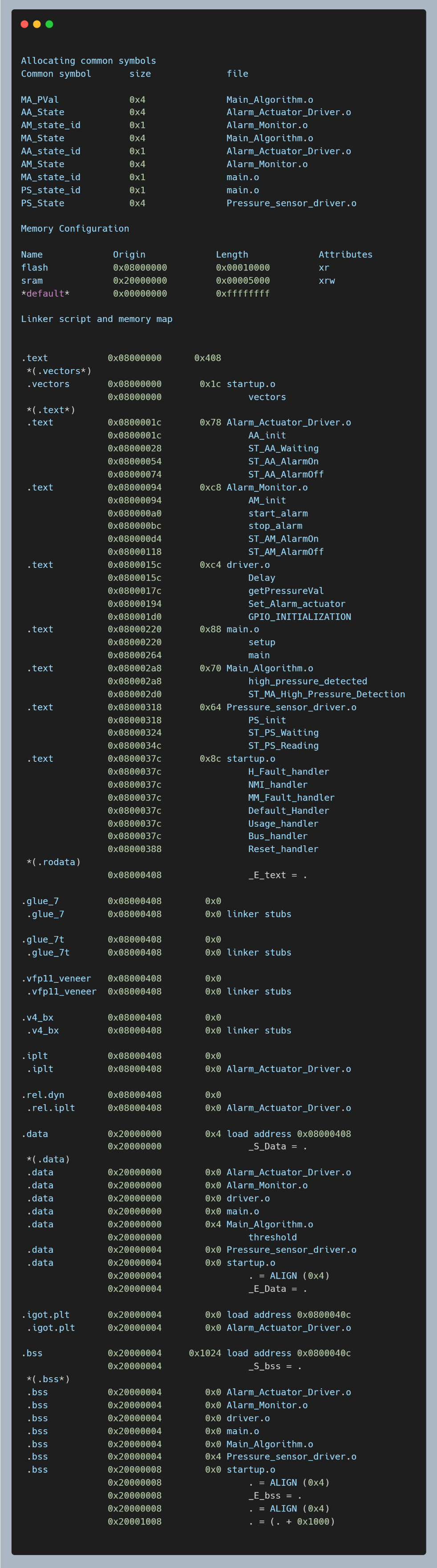
* 1. Symbol Table.



* 1. Section Table



* 1. Map file.



* 1. Hardware

A computer screen shot of a diagram

Description automatically generated

A computer screen shot of a computer

Description automatically generated

