SET UP A FIREWALL

# Link

<https://www.youtube.com/watch?v=qCuVBD2dmTA&list=PLnMKNibPkDnFzux3PHKUEi14ftDn9Cbm7&index=17>

# Description

In this paperwork, we will learn how to setup and configure a firewall on our STM32 board.

Contents

[Link 1](#_Toc89988274)

[Description 1](#_Toc89988275)

[Prerequisites 2](#_Toc89988276)

[STM32 Board 2](#_Toc89988277)

[ST-Link cable 2](#_Toc89988278)

[STM32CubeMX 2](#_Toc89988279)

[STM32CubeIDE 2](#_Toc89988280)

[Walkthrough 3](#_Toc89988281)

[Step 1 : Launch STM32CubeMX and generate the code 3](#_Toc89988282)

[Step 2 : Add firewall files 4](#_Toc89988283)

[Step 3 : Separate the memory. 5](#_Toc89988284)

[Step 4 : Write the main 6](#_Toc89988285)

[Step 5 : Conclusion 7](#_Toc89988286)

# Prerequisites

A picture containing chart

Description automatically generated

## STM32 Board

## ST-Link cable

## STM32CubeMX

## STM32CubeIDE

# Walkthrough

## Step 1 : Launch STM32CubeMX and generate the code

Launch STM32CubeMX and select the right board depending on the one you are using. In my case I use the L476RG Nucleo board. Then you can generate the code of your project.  
Don’t forget to select the correct IDE (in my case STM32CubeIDE).   
Graphical user interface, application

Description automatically generated

Step 2 : Add firewall files  
Go to your project, and go in Drivers > STM32L4xx\_HAL\_Driver > Inc.  
In this folder, you will add the 2 files :   
In Src : add STM32L4xx\_hal\_firewall.c  
In Inc : add STM32L4xx\_hal\_firewall.h  
You could find your drivers in your STM32 File.  
A picture containing table

Description automatically generated

Step 3 : Separate the memory.  
First thing, you have to separate the memory in the file named : <board>\_FLASH.id.

Text

Description automatically generated

Step 4 : Write the main  
You have define the GPIO\_NUMBER. Then create the Protected\_function that disable the firewall to call the Toggle\_led function that is using the low level commands directly (so they are restricted with the firewall).   
In the code section 2, we have the instantiation of the Firewall, and the function that enables it.  
In section 3 of the code, you will have the interruptions that are disabled, and then calls the protected function in a loop with delay.

  
Graphical user interface, text, application

Description automatically generated  
Text

Description automatically generated with medium confidence

Step 5 : Conclusion  
When we compile and run this code with the debugger, we can see that the led is blinking. Which is normal. To test if the firewall really works, you just have to comment the Protected\_function in the loop and use the Toggle\_led function.  
By doing this you will see that the led is not blinking anymore because the firewall is not disabled as in the Protected\_function.