ACTIVATE PCROP

# Link

<https://www.youtube.com/watch?v=qCuVBD2dmTA&list=PLnMKNibPkDnFzux3PHKUEi14ftDn9Cbm7&index=7>

# Description

In this paperwork, we will do all the steps to quickly separate activities in the memory and then to forbid to write or read in this specific zone in order to hide the code on the stm32 board we are using.

Contents

[Link 1](#_Toc89717128)

[Description 1](#_Toc89717129)

[Prerequisites 2](#_Toc89717130)

[STM32 Board 2](#_Toc89717131)

[ST-Link cable 2](#_Toc89717132)

[STM32CubeProgrammer 2](#_Toc89717133)

[STM32CubeMX 2](#_Toc89717134)

[STM32CubeIDE 2](#_Toc89717135)

[Walkthrough 3](#_Toc89717136)

[Step 1 : Run STM32CubeMX and generate the code 3](#_Toc89717137)

[Step 2 : Separate the specific code in another file. 4](#_Toc89717138)

[Step 3 : Link the file to the main 4](#_Toc89717139)

[Step 4 : Separate the memory 5](#_Toc89717140)

[Step 5 : Activate PCROP on STM32Programmer 6](#_Toc89717141)

[Step 6 : Check the PCROP memory section. 7](#_Toc89717142)

# Prerequisites

A picture containing chart

Description automatically generated

## STM32 Board

## ST-Link cable

## STM32CubeProgrammer

## STM32CubeMX

## STM32CubeIDE

# Walkthrough

## Step 1 : Run 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 WB55 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 : Separate the specific code in another file.

To do this you will need to create a new file in your src folder and write for example the code of a blinking light.

Graphical user interface, text, application

Description automatically generated

Step 3 : Link the file to the main  
When the code is written you just have to link it in our main.c file. To do so we will just write name of the previous file as a function. In our case led\_blinking().

Graphical user interface, text, application

Description automatically generated

## Step 4 : Separate the memory

To separate the memory I have to add a PCROP section in the memory (to do so I had to reduce the flash memory) .  
When created I just have to add in SECTIONS the .PCROPed to explain to my code that this section is reserved to my specific led\_blinking.c’s executable file.Graphical user interface, text, application

Description automatically generated  
Then you can just compile and debug.

## Step 5 : Activate PCROP on STM32Programmer

In this step we will go in the OB section of STM32Programmer and go to PCROP Protection. When you are in, you can select the start address and the end address of memory that you want to protect. Warning, you must have different values on it, so if it doesn’t change your value you can just increment it by one as for me.

Graphical user interface, text, application, email

Description automatically generated  
When PCROP is applied, don’t forget to press the reset button on your board.

## Step 6 : Check the PCROP memory section.

In this step you just have to go to the memory section of our PCROP, which is 0x08008000 and you will see that we can’t see anything in the memory, the same pattern is always written.

Graphical user interface, application, table

Description automatically generated