

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
Description	1
Prerequisites	2
STM32 Board	2
ST-Link cable	2
STM32CubeProgrammer	2
STM32CubeMX	2
STM32CubeIDE	2
Walkthrough	3
Step 1 : Run STM32CubeMX and generate the code	3
Step 2 : Separate the specific code in another file.	4
Step 3 : Link the file to the main	4
Step 4 : Separate the memory	5
Step 5 : Activate PCROP on STM32Programmer	6
Step 6 : Check the PCROP memory section.	7

Prerequisites

Security Features by STM32 Series 7

STM32 Series	Security Features																Arm Cortex®
	96-Bit Unique ID	FLASH WRP	FLASH PCROP	FLASH RDP	Unique entry point	Secure mem/HDP	MPU	Firewall	Trustzone	OTFDEC	Tamper	TRNG	CRYPT AES	HASH	PKA	CryptoLib	
STM32 F0	■	■		■							■					■	M0
STM32 F1	■	■					■				■					■	M3
STM32 F2	■	■		■			■				■	■	■	■		■	M3
STM32 F3	■	■		■			■				■					■	M4
STM32 F4	■	■	■	■			■				■	■	■	■		■	M4
STM32 F7	■	■	■	■			■				■	■	■	■		■	M7
STM32 L0	■	■		■			■	■			■		■			■	M0+
STM32 L1	■	■		■			■				■		■			■	M3
STM32 L4	■	■		■			■	■			■	■	■			■	M4
STM32 L5	■	■		■	■	■	■		■	■	■	■	■	■	■	■	M33
STM32 H7	■	■	■	■	■	■	■				■	■	■	■		■	M7/M4
STM32 G0	■	■		■		■	■				■	■	■			■	M0+
STM32 G4	■	■		■		■	■				■	■	■			■	M4
STM32 WB	■	■		■			■				■	■	■		■	■	M4/M0+

Available on all devices

Depends on device part number

STM32 Board

ST-Link cable

STM32CubeProgrammer

STM32CubeMX

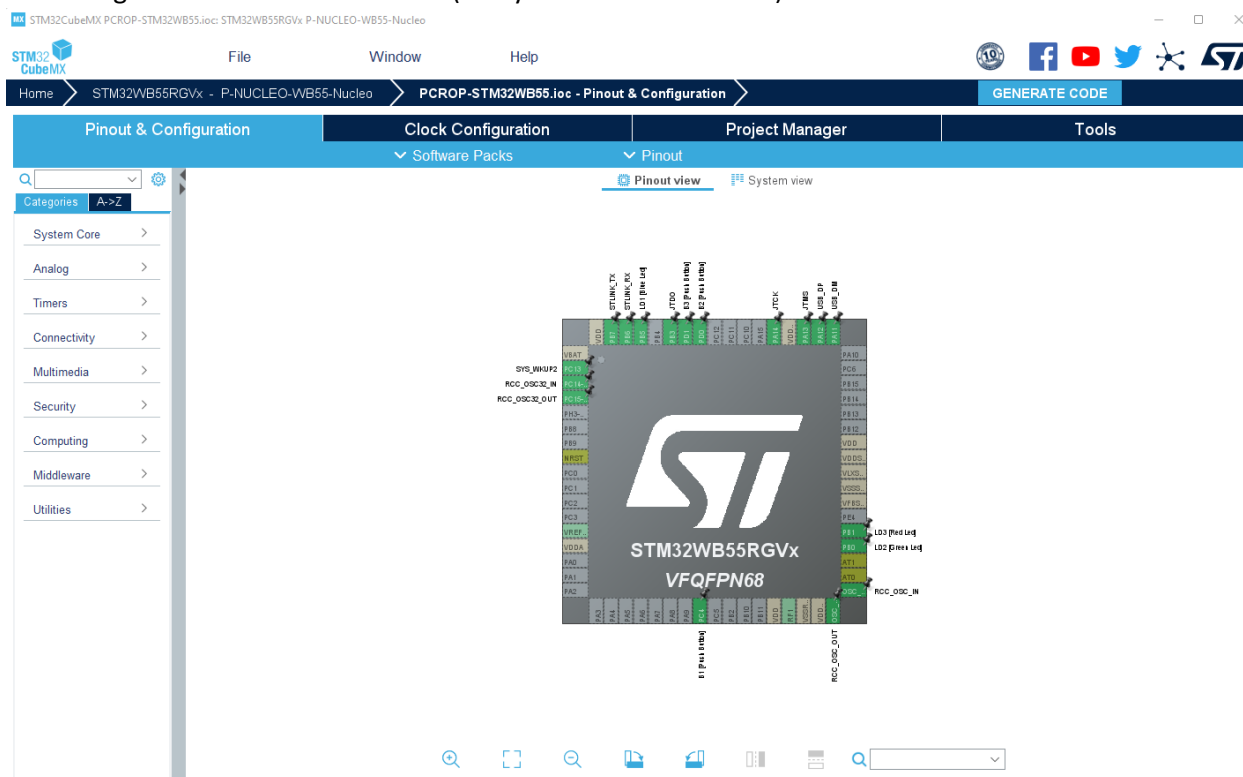
STM32CubeIDE

Walkthrough

Step 1 : Run STM32CubeMX and generate the code

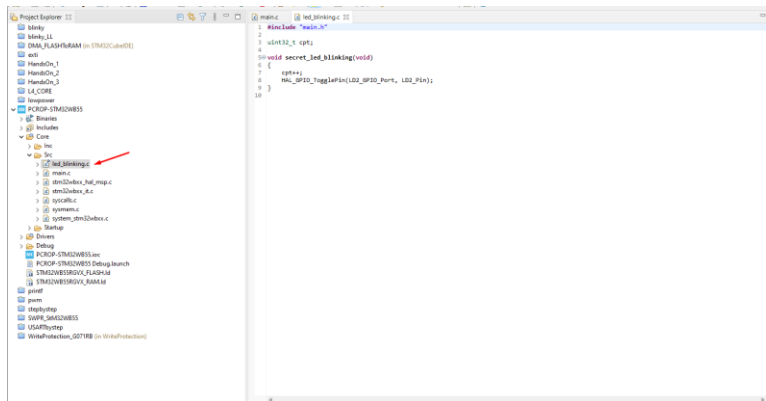
Launch STM32CubeMX and select the right board depending on the one you are using. In my case I use the WB55 Nucleo board. Then you can generate the code of your project.

Don't forget to select the correct IDE (in my case STM32CubeIDE).



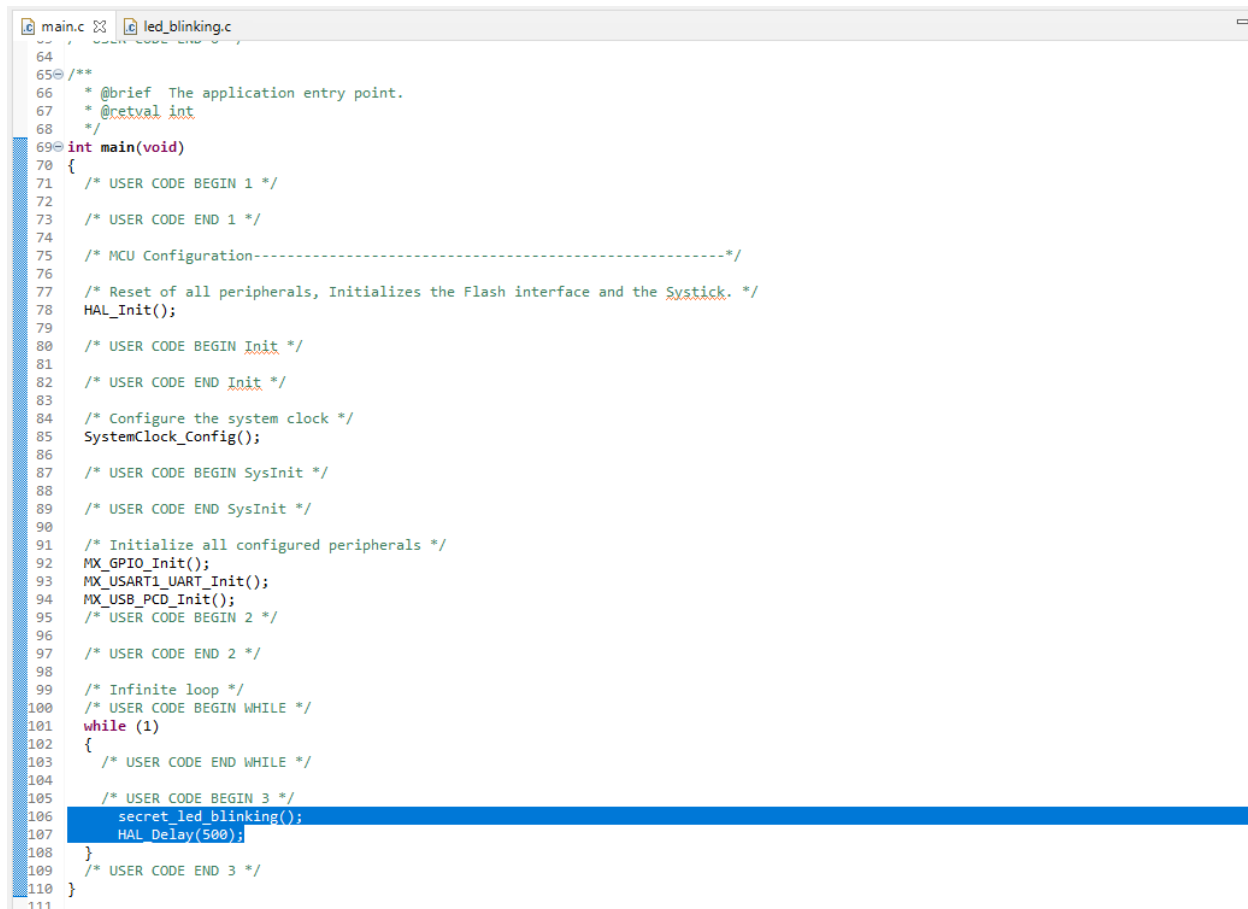
Step 2 : Separate the specific code in another file.

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.



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().



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.

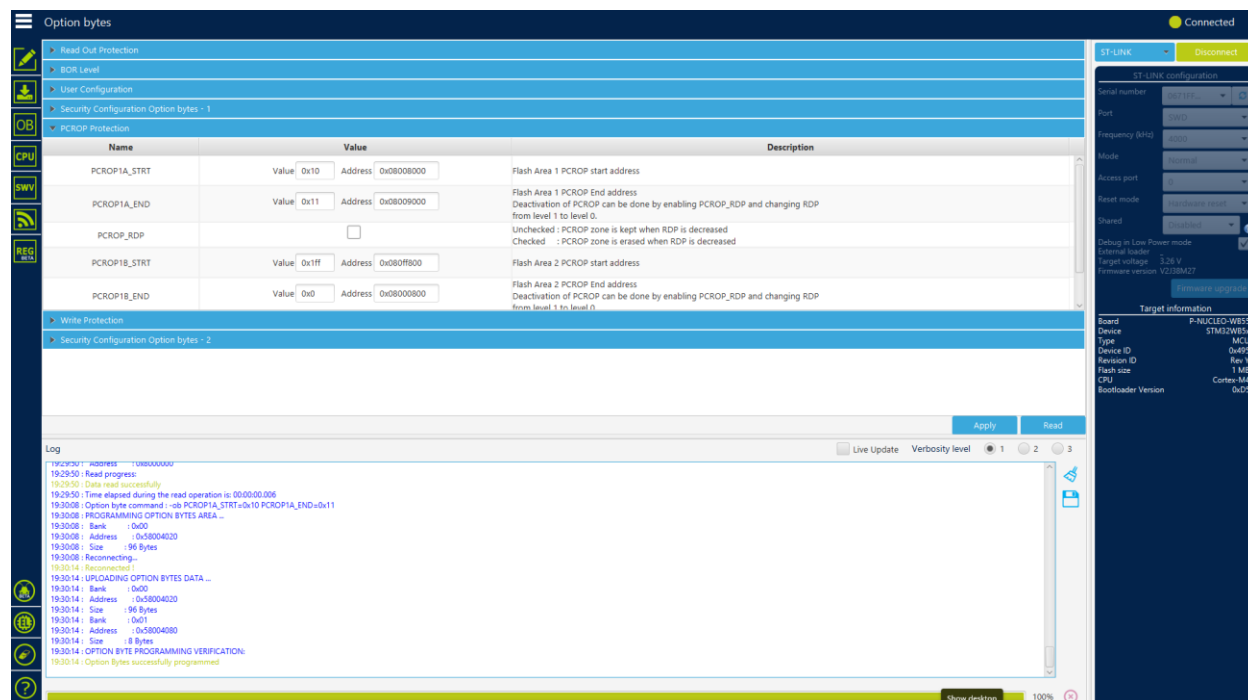
```
STM32WB55RGVX_FLASH.ld
34 */
35
36 /* Entry Point */
37 ENTRY(Reset_Handler)
38
39 /* Highest address of the user mode stack */
40 _estack = ORIGIN(RAM1) + LENGTH(RAM1); /* end of RAM1 */
41 /* Generate a link error if heap and stack don't fit into RAM */
42 _Min_Heap_Size = 0x200; /* required amount of heap */
43 _Min_Stack_Size = 0x400; /* required amount of stack */
44
45 /* Specify the memory areas */
46 MEMORY
47 {
48 FLASH (rx)           : ORIGIN = 0x08000000, LENGTH = 16K
49 PCROP (x)            : ORIGIN = 0x08000000, LENGTH = 16K
50 RAM1 (xrw)           : ORIGIN = 0x20000004, LENGTH = 0x2FFFC
51 RAM_SHARED (xrw)     : ORIGIN = 0x20030000, LENGTH = 10K
52 }
53
54 /* Define output sections */
55 SECTIONS
56 {
57 /* The startup code goes first into FLASH */
58 .isr_vector :
59 {
60 . = ALIGN(4);
61 KEEP(*(.isr_vector)) /* Startup code */
62 . = ALIGN(4);
63 } >FLASH
64
65 .PCROPed :
66 {
67 . = ALIGN(4);
68 *led_blinking.o (.text .text*)
69 . = ALIGN(4);
70 } > PCROP
71
```

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.



The screenshot shows the STM32Programmer software interface. The 'Option bytes' section is expanded, and 'PCROP Protection' is selected. The configuration table is as follows:

Name	Value	Address	Description
PCROP1A_STRT	Value: 0x10	Address: 0x08000000	Flash Area 1 PCROP start address
PCROP1A_END	Value: 0x11	Address: 0x08009000	Flash Area 1 PCROP End address Deactivation of PCROP can be done by enabling PCROP_RDP and changing RDP from level 1 to level 0.
PCROP_RDP	<input type="checkbox"/>		Unchecked : PCROP zone is kept when RDP is decreased Checked : PCROP zone is erased when RDP is decreased
PCROP1B_STRT	Value: 0x1FF	Address: 0x0800F800	Flash Area 2 PCROP start address
PCROP1B_END	Value: 0x0	Address: 0x08000800	Flash Area 2 PCROP End address Deactivation of PCROP can be done by enabling PCROP_RDP and changing RDP from level 1 to level 0.

Below the table, there are 'Apply' and 'Read' buttons. The 'Log' section at the bottom shows the following messages:

```

19:30:01 : Success : 1980000000
19:30:01 : Read progress
19:30:01 : Data read successfully
19:30:01 : Time elapsed during the read operation is 00:00:00.006
19:30:08 : Option byte command : -sb PCROP1A_STRT=0x10 PCROP1A_END=0x11
19:30:08 : PROGRAMMING OPTION BYTES AREA ...
19:30:08 : Bank : 0x00
19:30:08 : Address : 0x08004020
19:30:08 : Size : 96 Bytes
19:30:08 : Reconnecting...
19:30:14 : Reconnected !
19:30:14 : UPLOADING OPTION BYTES DATA ...
19:30:14 : Bank : 0x00
19:30:14 : Address : 0x08004020
19:30:14 : Size : 96 Bytes
19:30:14 : Bank : 0x01
19:30:14 : Address : 0x08004080
19:30:14 : Size : 8 Bytes
19:30:14 : OPTION BYTE PROGRAMMING VERIFICATION:
19:30:14 : Option Bytes successfully programmed
  
```

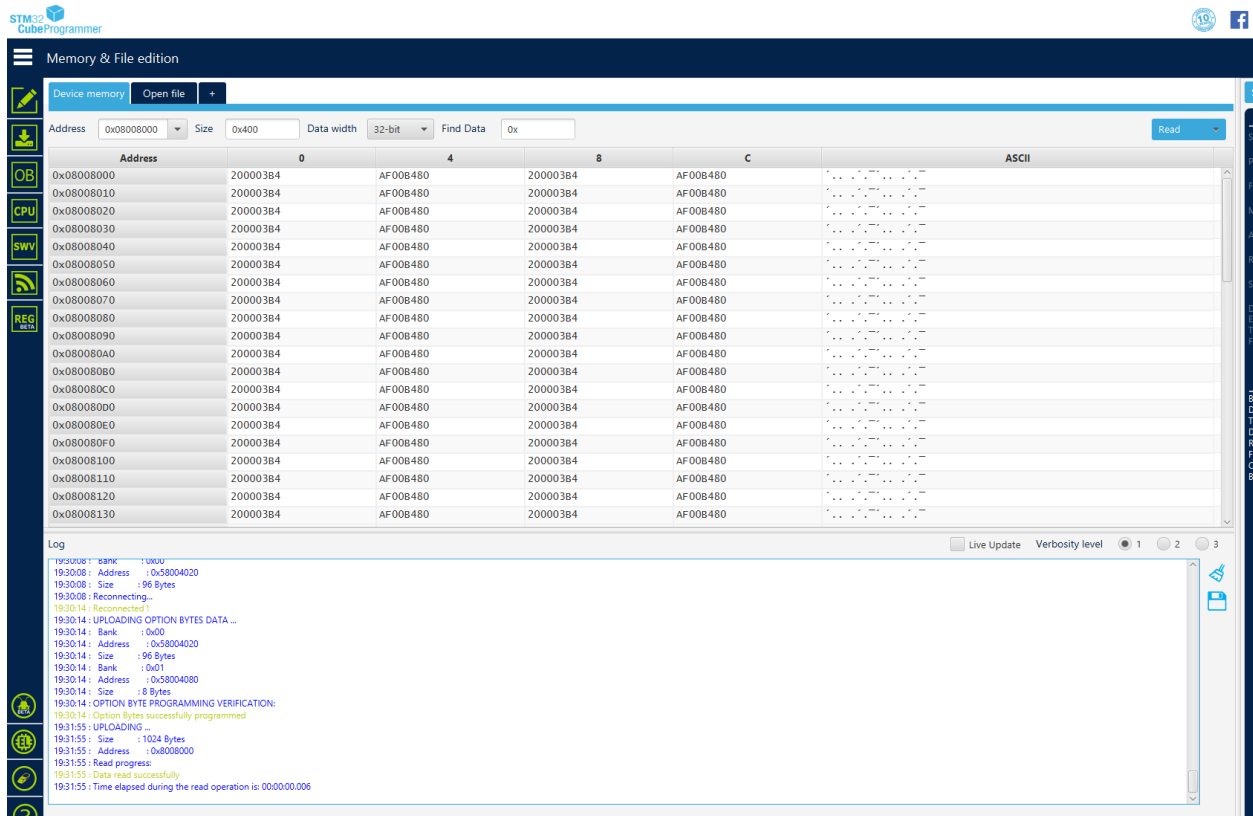
On the right side, the 'ST-LINK configuration' panel shows 'Connected' status. The 'Target information' panel shows:

- Board: P-NUCLEO-WB55
- Device: STM32WB55
- Type: MCU
- Device ID: 0x00000000
- Revision ID: Rev 1
- Flash size: 1 MB
- CPU: Cortex-M4
- Bootloader Version: 0x05

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.



Memory & File edition

Device memory | Open file | +

Address: 0x08008000 | Size: 0x400 | Data width: 32-bit | Find Data: 0x | Read

Address	0	4	8	C	ASCII
0x08008000	200003B4	AF00B480	200003B4	AF00B480
0x08008010	200003B4	AF00B480	200003B4	AF00B480
0x08008020	200003B4	AF00B480	200003B4	AF00B480
0x08008030	200003B4	AF00B480	200003B4	AF00B480
0x08008040	200003B4	AF00B480	200003B4	AF00B480
0x08008050	200003B4	AF00B480	200003B4	AF00B480
0x08008060	200003B4	AF00B480	200003B4	AF00B480
0x08008070	200003B4	AF00B480	200003B4	AF00B480
0x08008080	200003B4	AF00B480	200003B4	AF00B480
0x08008090	200003B4	AF00B480	200003B4	AF00B480
0x080080A0	200003B4	AF00B480	200003B4	AF00B480
0x080080B0	200003B4	AF00B480	200003B4	AF00B480
0x080080C0	200003B4	AF00B480	200003B4	AF00B480
0x080080D0	200003B4	AF00B480	200003B4	AF00B480
0x080080E0	200003B4	AF00B480	200003B4	AF00B480
0x080080F0	200003B4	AF00B480	200003B4	AF00B480
0x08008100	200003B4	AF00B480	200003B4	AF00B480
0x08008110	200003B4	AF00B480	200003B4	AF00B480
0x08008120	200003B4	AF00B480	200003B4	AF00B480
0x08008130	200003B4	AF00B480	200003B4	AF00B480

Log | Live Update | Verbosity level: 1 | 2 | 3

```

19:30:08 : Bank : 0x00
19:30:08 : Address : 0x58004020
19:30:08 : Size : 96 Bytes
19:30:08 : Reconnecting...
19:30:14 : Reconnected !
19:30:14 : UPLOADING OPTION BYTES DATA ...
19:30:14 : Bank : 0x00
19:30:14 : Address : 0x58004020
19:30:14 : Size : 96 Bytes
19:30:14 : Bank : 0x01
19:30:14 : Address : 0x58004080
19:30:14 : Size : 8 Bytes
19:30:14 : OPTION BYTE PROGRAMMING VERIFICATION:
19:30:14 : Option Bytes successfully programmed
19:31:55 : UPLOADING ...
19:31:55 : Size : 1024 Bytes
19:31:55 : Address : 0x08008000
19:31:55 : Read progress:
19:31:55 : Data read successfully
19:31:55 : Time elapsed during the read operation is 00:00:00.006
  
```