STM32 Crypto Library

This tutorial will introduce the STM32 Crypto Library. After this tutorial you will be familiar with how to add the library and header file and implement a basic code example.

- This Crypto Library contains a software implementation of the cryptographic algorithms and has hardware accelerators enhancement for some of them.
- Includes: encryption, hashing, message, authentication, and digital signing.

Literature:

- STM32L476xx Datasheet
- UM1724 User manual STM32 Nucleo-64 boards
- UM1884 Description of STM32L4/L4+ HAL and low-layer drivers
- UM1718 User manual STM32CubeMX for STM32 configuration and initialization C code generation
- RM0351 Reference Manual
- See X-CUBE-CryptoLib Documentation

Resources:

- X-CUBE-CRYPTOLIB

Stages

- Download X-CUBE-CRYPTOLIB
- Create New Project
- Add Header Files and Edit main.c
- Build and Debug

1. Download X-CUBE-CRYPTOLIB

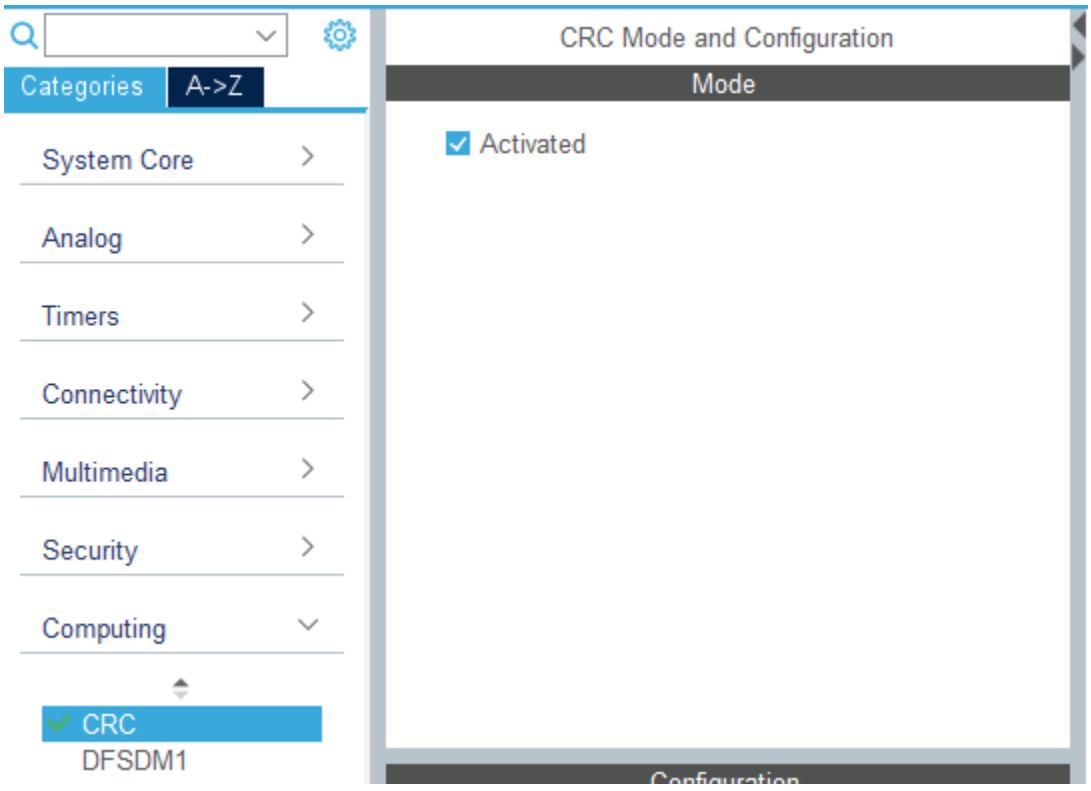
The library can be found on ST's website at: https://www.st.com/en/embedded-software/x-cube-cryptolib.html

It may take some time to be approved for download for security reasons. Regularly check your email.

2. Create New Project

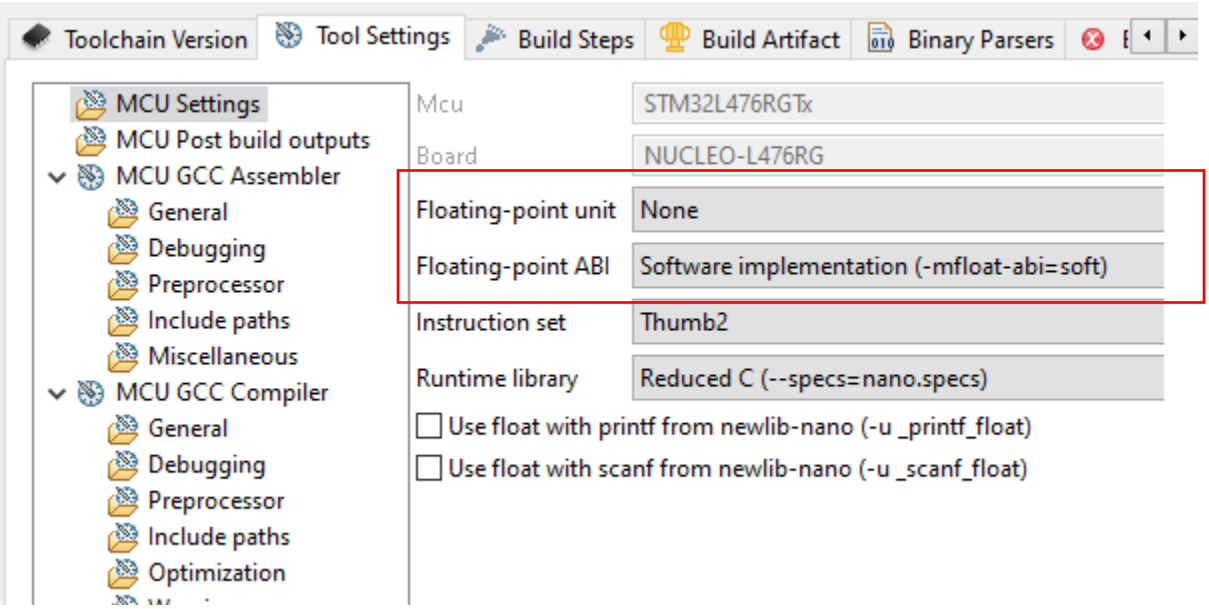
Go to File->New->STM32 Project. Navigate to board selector and select Nucleo-L476RG. Select yes when prompted to initialize peripherals in their default mode.

Once CubeMX is opened, expand the computing tab(on the left) and click on CRC. Check the box to activate it.

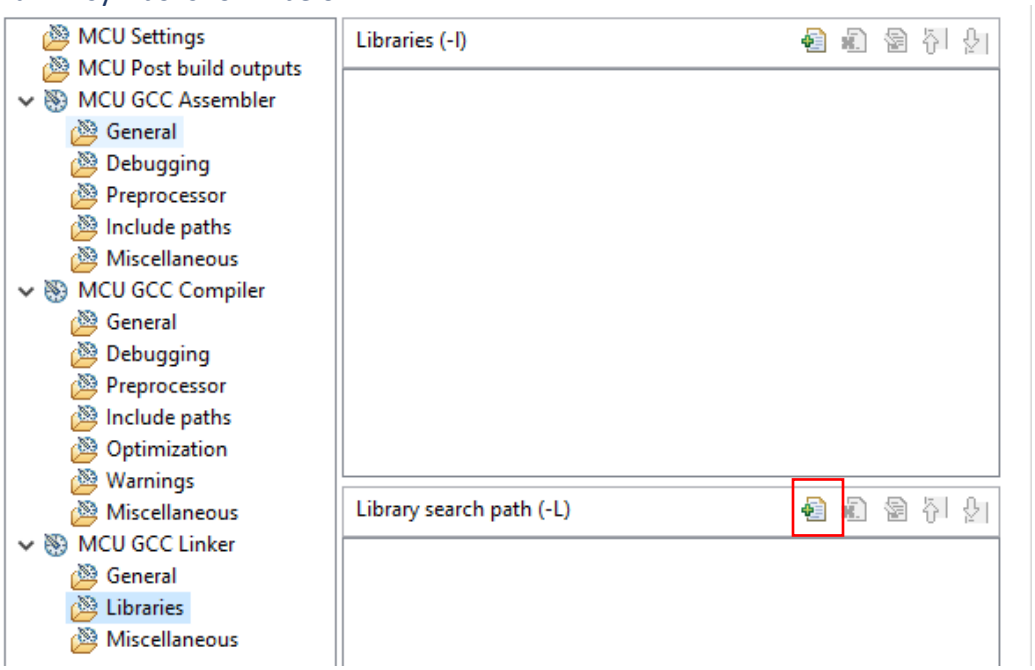


Next, save the change and select yes when asked to generate code. First, we want to add the library to the linker.

Right click the project from the Project Explorer window and select properties. Go to C/C++ Build->Settings->Tool Settings. In the drop-down menu labeled “Floating-pint unit” select “None.” In the drop-down menu labeled “Floating-point ABI” select “Software implementation” as shown below:



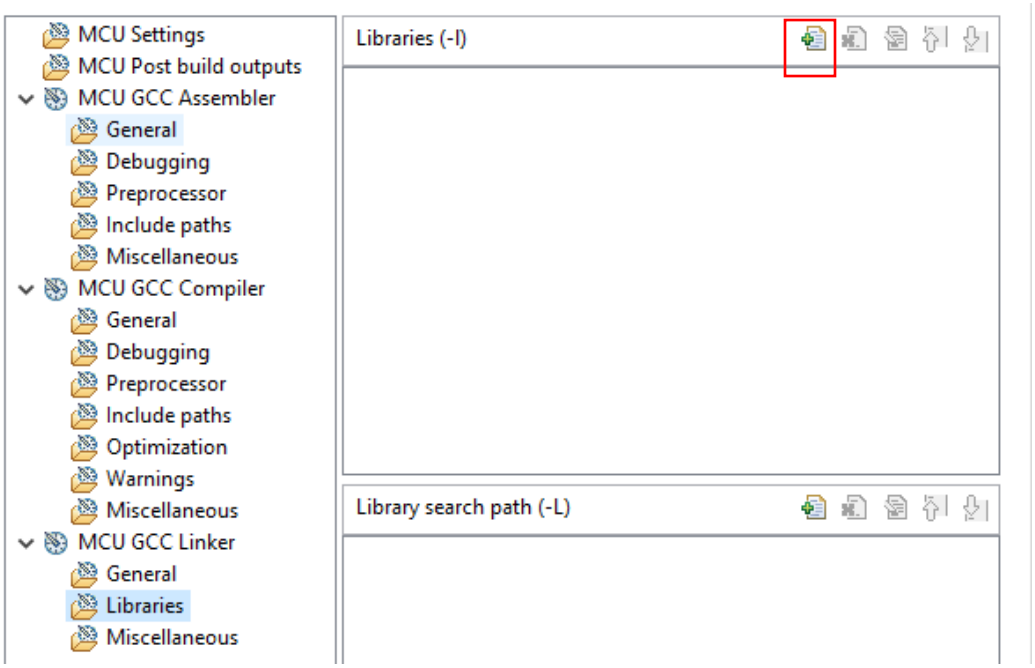
Next, still in the Tool Settings tab, expand “MCU GCC Linker” and click “Libraries.” Select the icon with a “+” symbol shown below.



Click “File System” and navigate to the directory shown where the X-CUBE-CRYPTOLIB was downloaded. An example path is shown below:

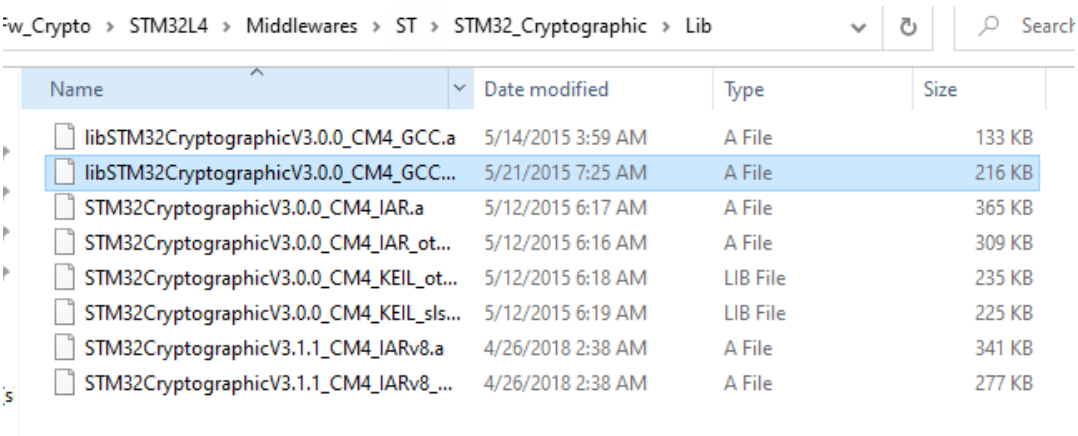
C:\en.x-cube-cryptolib\STM32CubeExpansion_Crypto_V3.1.0\Fw_Crypto\STM32L4\Middlewares\ST\STM32_Cryptographic\Lib

Once selected, click OK. Now we will add the library name. Click on the icon with a plus show below:



First, in you File Explorer application, open the location where the X-CUBE-CRYPTOLIB was downloaded and navigate to the directory shown below:

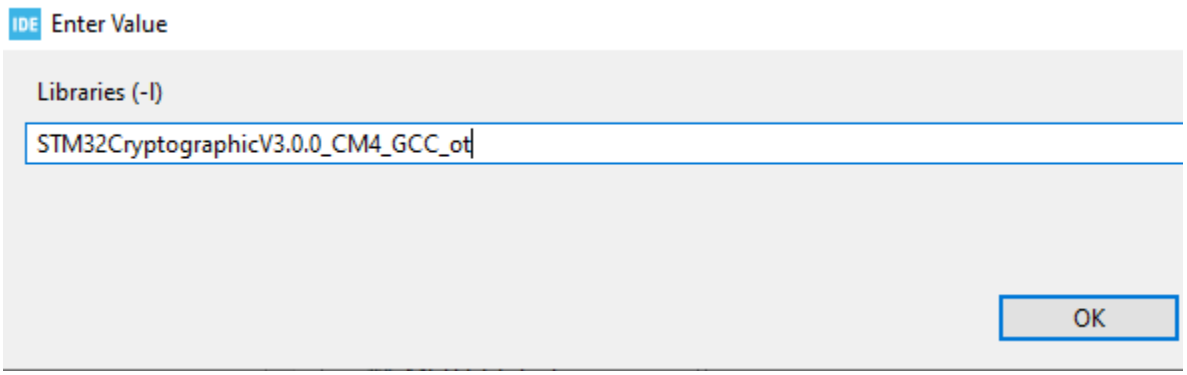
C:\en.x-cube-cryptolib\STM32CubeExpansion_Crypto_V3.1.0\Fw_Crypto\STM32L4\Middlewares\ST\STM32_Cryptographic\Lib



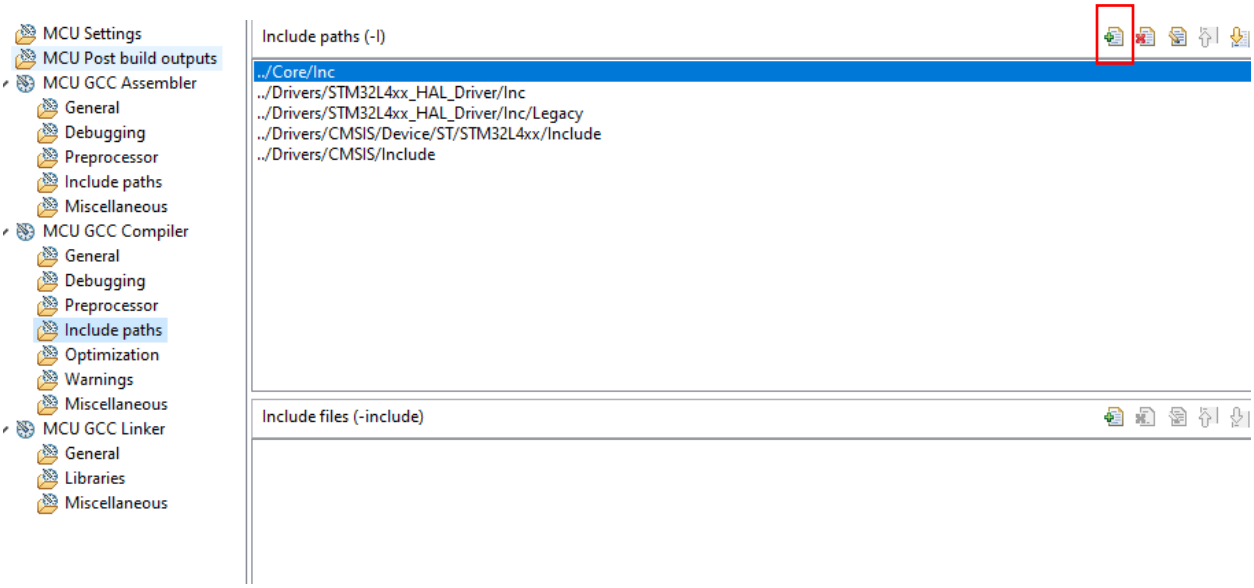
The file named “libSTM32CryptographicV3.0.0_CM4_GCC_ot.a” is what we will be selecting; however, we only want to select a portion of the file name. Copy the highlighted part shown below:

libSTM32CryptographicV3.0.0_CM4_GCC_ot.a

Paste it into CubeIDE.



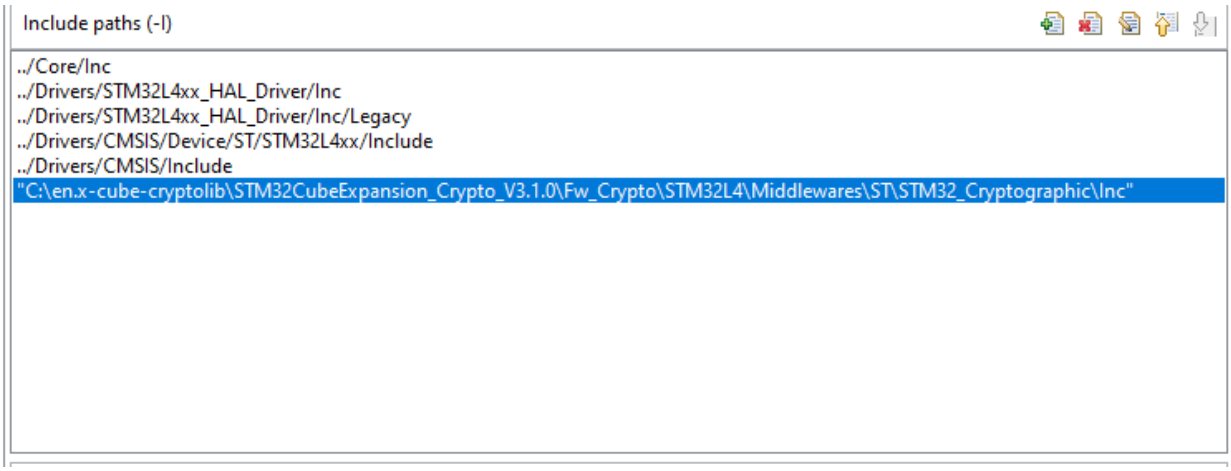
Next, expand the MCU GCC Compiler tab and click on “Include paths.” Click the icon with a “+” symbol.



Click File System and select the include directory shown below:

C:\en.x-cube-cryptolib\STM32CubeExpansion_Crypto_V3.1.0\Fw_Crypto\STM32L4\Middlewares\ST\STM32_Cryptographic\Inc

Include paths should look like this:



Now click, Apply and Close.

3. Add Header Files and Edit main.c

Navigate to main.c and type #include “crypto.h.” Right click crypto.h and click Open Declaration. Once crypto.h is open, right click “config.h” and click Open Declaration. Navigate to lines 139-141, we can see it is current defined for IAR, but we want to define it for GCC. Comment out #define IAR and uncomment #define GCC as shown below:

```
138   */
139    /*#define IAR      ((uint8_t)0x01)         !< Library is compiled with Iar  */
140   /* #define KEIL                  ((uint8_t)0x02)   */    /*!< Library is compiled with Keil *
141   #define GCC                    ((uint8_t)0x03)   */    /*!< Library is compiled with GCC  */
142
143 /**
```

To verify the steps completed so far, build the project and verify there are no errors. Now we can copy and paste the code from the Nucleo-L152RE example found in the crypto expansion pack.

Go to File->Import and select the directory shown below:

C:\en.x-cube-cryptolib\STM32CubeExpansion_Crypto_V3.1.0\Fw_Crypto\STM32L1\Projects\STM32L152RE-Nucleo\AES\AES128_CCM\TrueSTUDIO

Once the project has been imported open main.c and copy lines 26-98 and paste them into the Private Variables section in our main.c file. Next copy the 3 private function prototypes (lines 103-128) and paste them under “USER CODE BEGIN PFP.”

Next, copy the function implementations found under the main function (lines 239-421) and paste them under USER CODE BEGN 4. Now we will copy the main algorithm. Copy lines 170-231) and paste under the line where CRC CLK was enabled (line 168). Next, we will copy line 143 from the Nucleo-L1 example and paste it under USER CODE BEGIN 1.

4. Build and Debug

Build the project and click debug. Resume the execution, we can see the green LED was on meaning AES CMM operations were successful.