AES

# Link

<https://www.youtube.com/watch?v=qCuVBD2dmTA&list=PLnMKNibPkDnFzux3PHKUEi14ftDn9Cbm7&index=25>

# Description

In this paperwork, we will learn how to use symmetric encryption on our board.

Contents

[Link 1](#_Toc90314842)

[Description 1](#_Toc90314843)

[Prerequisites 2](#_Toc90314844)

[STM32 Board 2](#_Toc90314845)

[ST-Link cable 2](#_Toc90314846)

[STM32CubeMX 2](#_Toc90314847)

[STM32CubeIDE 2](#_Toc90314848)

[Walkthrough 3](#_Toc90314849)

[Step 1 : Launch STM32CubeMX and generate the code 3](#_Toc90314850)

[Step 2 : Write the main 4](#_Toc90314851)

[Step 3 : Check the encryption 5](#_Toc90314852)

# Prerequisites

A picture containing chart

Description automatically generated

## STM32 Board

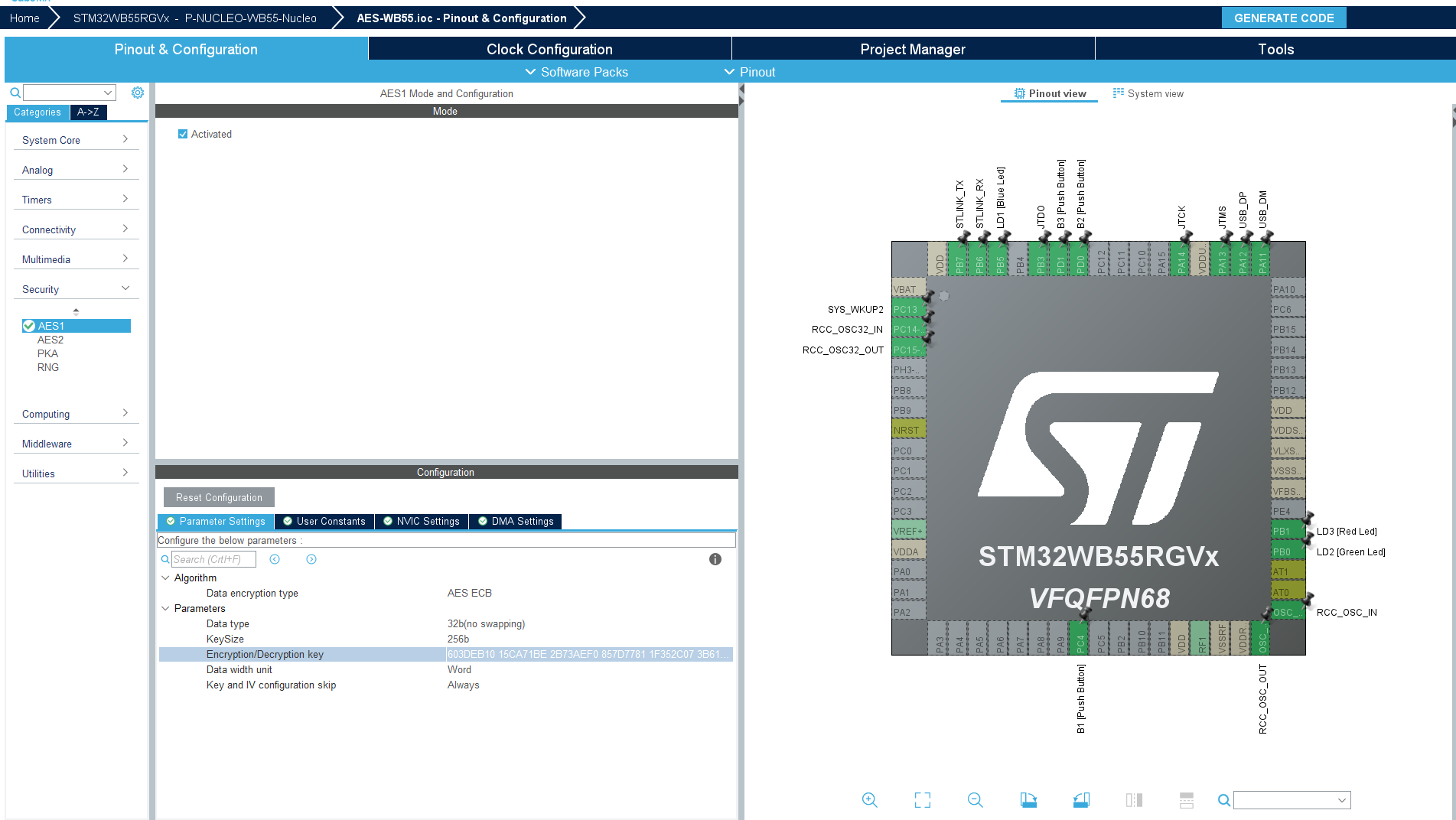
## ST-Link cable

## 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 WB55 Nucleo board. Then you can generate the code of your project.  
Then you have to activate AES1.  
In our example we will use a 256b key size with the following Encryption and Decryption key :   
603deb1015ca71be2b73aef0857d77811f352c073b6108d72d9810a30914dff4  
  


Step 2 : Write the main  
First thing we have to declare variables in the main.c.  
So we will set an input data as an array, and put two empty arrays for the data we will encrypt and decrypt.  
Text, letter

Description automatically generated

Then we will write in the main the following code. The Measure functions are made to calculate the number of ticks of the clock needed to realize those functions.  
Graphical user interface, text, application, email

Description automatically generated  
If the code is completed, just compile and launch the debug.

Step 3 : Check the encryption

Finally we just have to put our variables in the debug expressions and check them after the program has been launched.  
Graphical user interface, table

Description automatically generated  
We can see that the encryption and decryption work perfectly because the input data and the decrypted one are the same. As for the encrypted data, we can see that it has totally changed from the input\_data.