Use of USB in Virtual Com mode

Targeted competences: Use of USB in order to realize a virtual serial communication

Hardware: STM32F7 Nucleo board

Framework: STM32CubeIDE v1.1.0 from STMircoelectronics

The aim of this document is to show how to use an USB link as a virtual serial communication between the PC and th STM32 board.

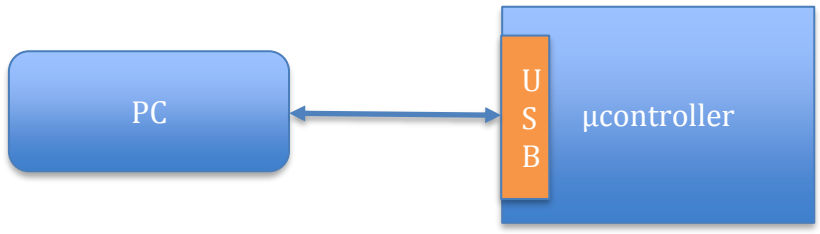

Figure 1: Global view of the system

We develop here a small code able to write a message to the PC (on the USB link) as soon as it received a message from the PC by this same link. If the PC sends no message then the microcontroller does not send any message too.

1. Microcontroller configuration

The first step is to configure the microcontroller. In our case, we use the NUCLEO-F767ZI platform based on a STM32F7 architecture. In order to configure this board we will use the CubeMx softwareThe first step after choosing the board is to name the project.

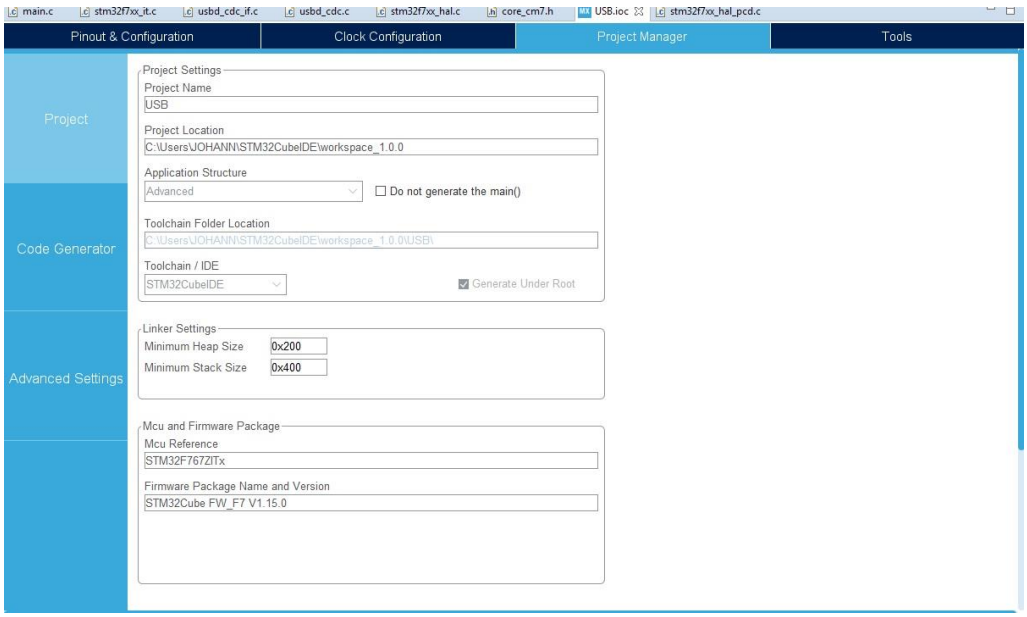

Figure 2:CubeIDE interface

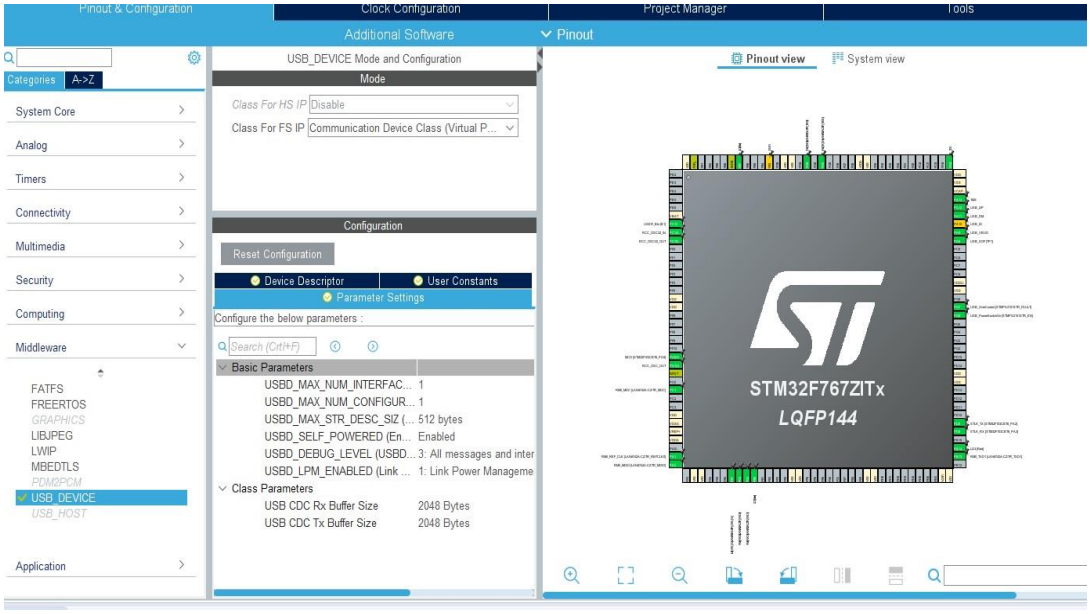

Figure 3:CubeMx interface to configure the STM32 Device

Here, I chose to use the USB bus from the Middleware tab and I configure the buffer size at 2048 Bytes. You can choose another configuration of course because more the buffer is large and more the memory usage is important. If in your application, you receive and/or transfer only messages of few bytes you can choose smaller buffer sizes.

Now we have to configure the clock for the whole board; to do this click on clock configuration item. With this configuration panel, we can choose the frequency of different µcontroller components as CPU frequency, AHB, APB1 buses and so on. I decide to use the maximum frequency of the SYSCLK that is 216MHz. In this case, the frequency of the APB1 and APB2 timer clocks are respectively 108MHz and 216MHz (see Figure 5)..

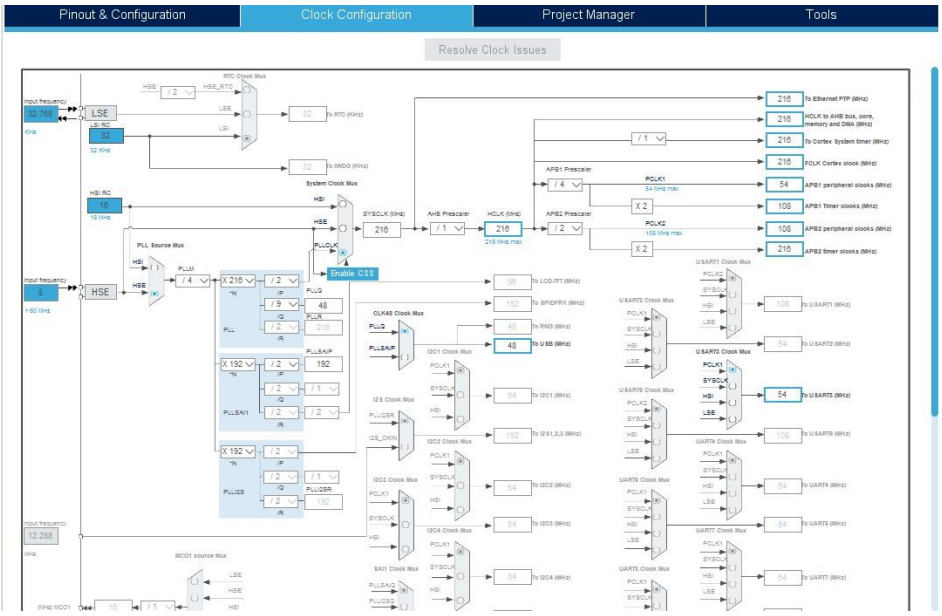

Figure 4: Clock configuration for the board

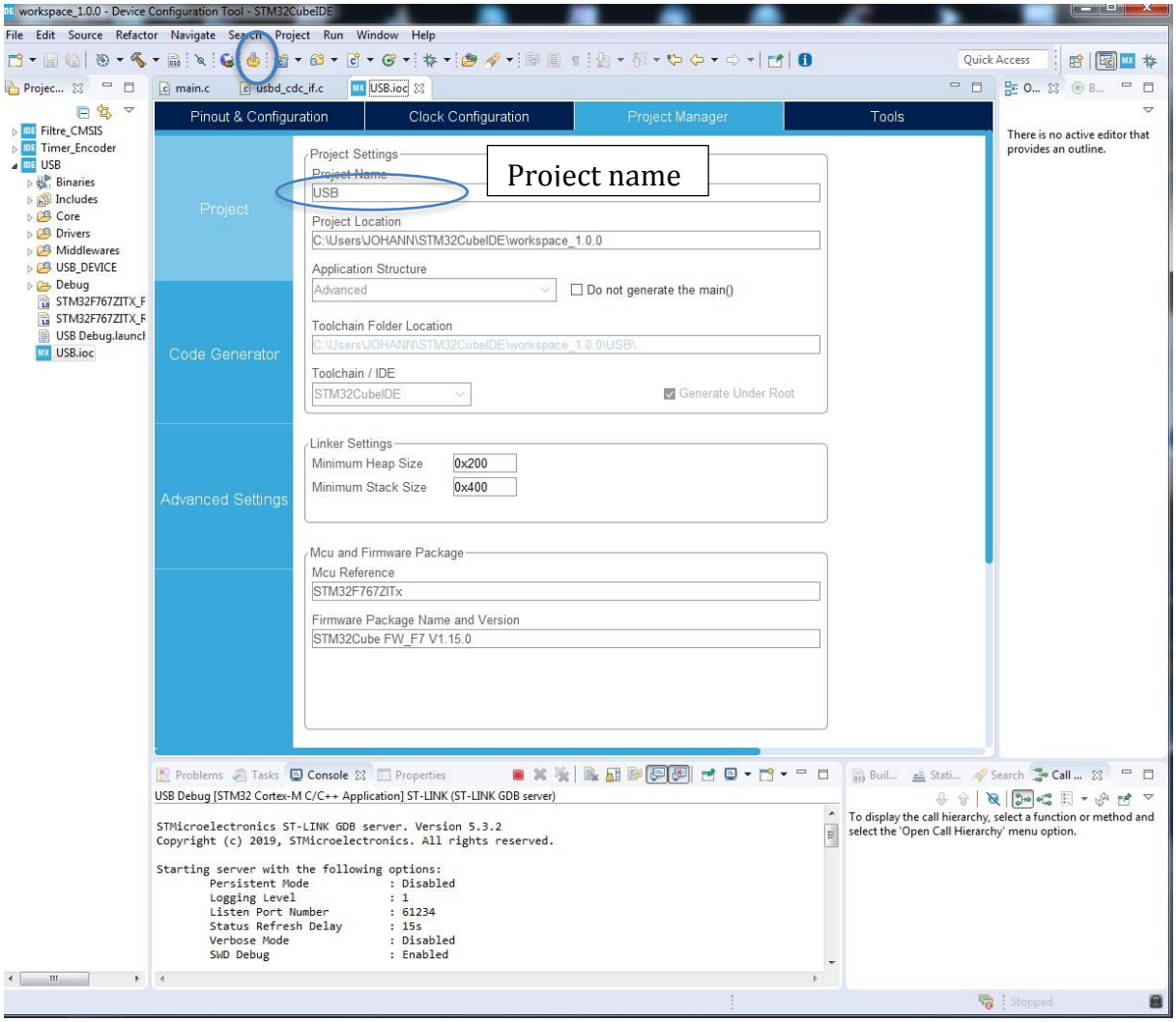

Figure 5: Project manager configuration

You must give a name for your project and modify if you need the stack and heap parameters. Be careful if you use dynamic memory allocation indeed you should try to estimate the maximum memory size you need in order to choose the best heap size. If the heap size is too small, you will have some bugs during the program execution.

After clicking on Generate Code, CubeMx generates the application code and creates the project as shown in the figure below.

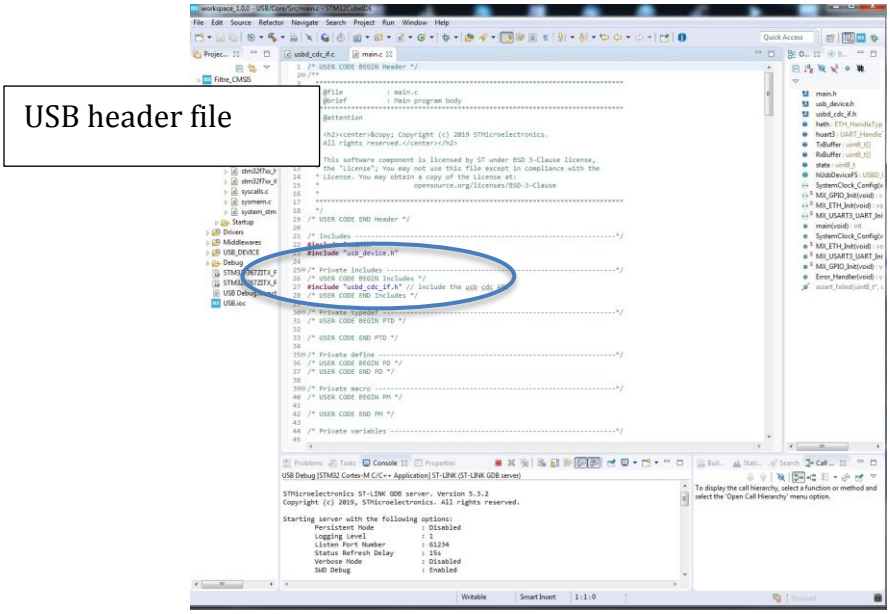

Figure 6: Code generation

In order to use the USB API, we have to define some buffers and variables. Here I declare 2 buffers: one for the reception and one for the transfer. I also define a variable state to be able to know if I received something from the PC by the USB link.

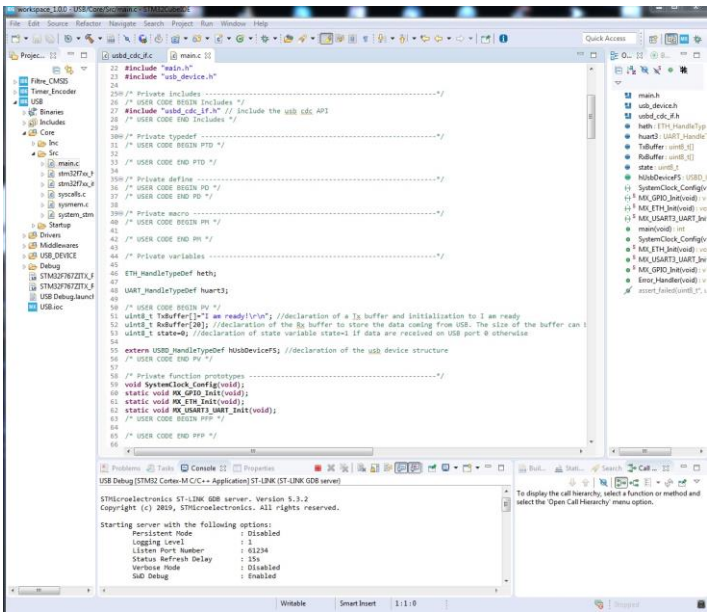

Figure 7: Variable declarations

In order to save the message transmitted from the PC, I stored this one into my RxBuffer (declared into the main.c and declared as an extern variable into the usbd_cdc_if.c file). To do that I added some code into the CDC_Receive_FS function as presented below.

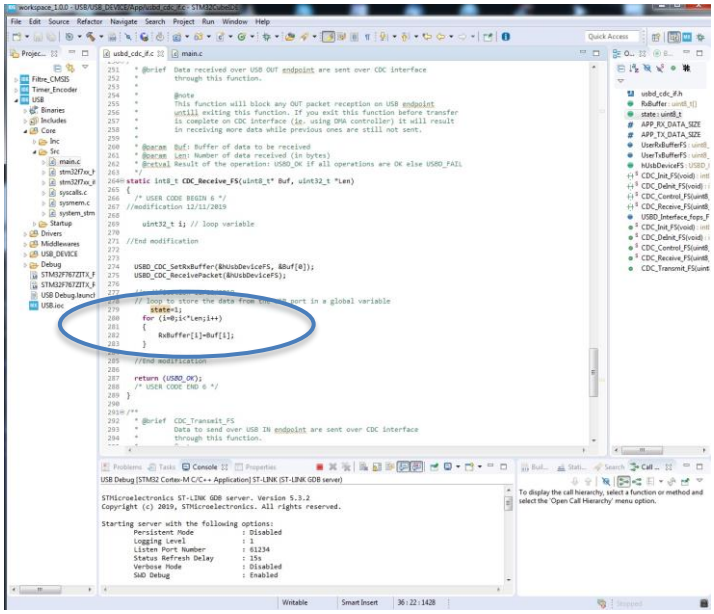

Figure 8: CDC_Receive_FS function modification

As a message is received from the USB link then the callback function CDC_Receive_FS is called. In this function, I added the following code: first I set the state variable to specify that a message has been received from the USB link, second I stored all the data of the Buf buffer (USB buffer) into my RxBuffer. The RxBuffer is a clone of the Buf buffer that I can use to realize different processing if I want. Here I use only the RxBuffer in order to resend the message to the PC (like an echo).

In the While(1) part of the main.c I added some code in order to send to the PC the TxBuffer (I am ready message) and the RxBuffer (this message is an echo of the message received from the PC). If no message is sent from the PC first then no message is sent from the microcontroller to the PC.

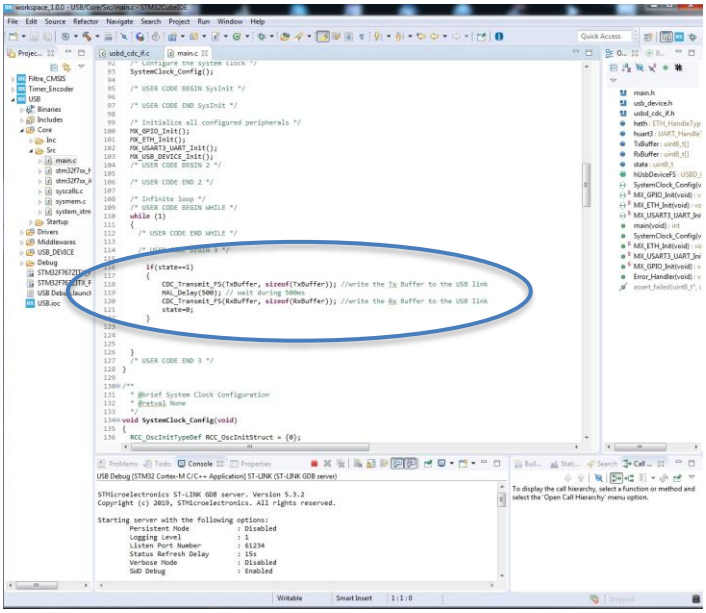

Figure 9: Code to resend to the PC the TxBuffer and the RxBuffer

At the end we have to compile the code in order to obtain the executable file for the STM32F7. When the compilation is done without error, we can transfer the code into the board.

A video showing the test of the code can be found here: https://youtu.be/Cg7sSYqJxYA