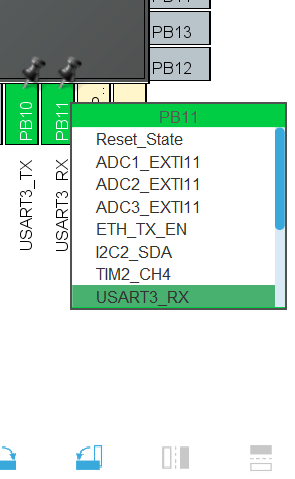
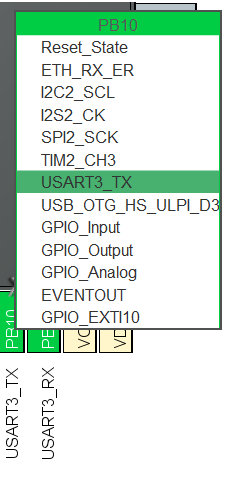
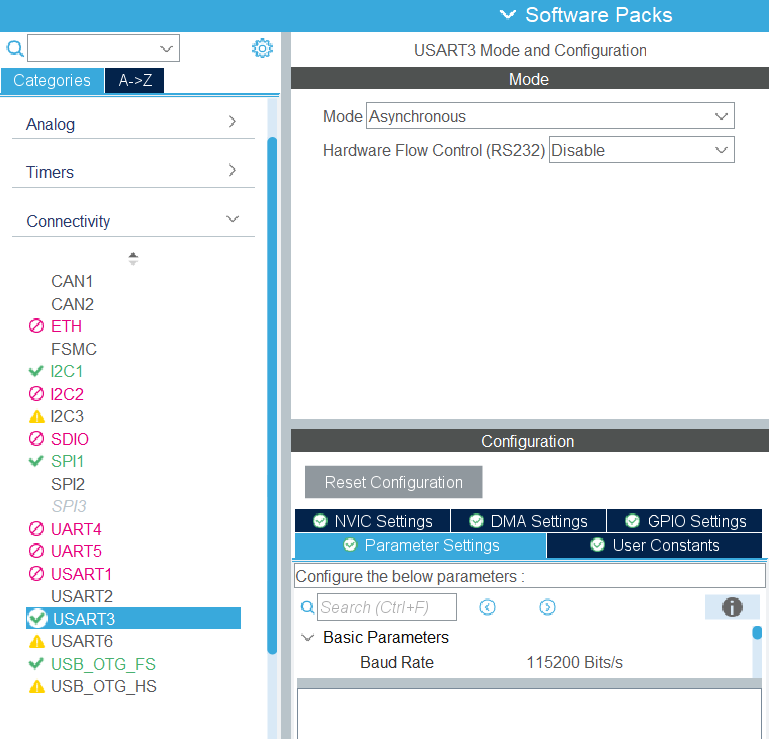
**UART Serial Communication**

**Send a message:**

1. You will need to have another board that support serial and have a TXD/RXD .
2. Open the .ioc of your project an configure pins PB10 and PB11



1. Go to Connectivity and USART3 and make that configure it , also set your Baud Rate to 11520



1. Ctrl + S to generate your code
2. Complete the main with this code:

char message[40] = {'\0'};

// Declare a character buffer of 40 positions, initialized with '\0' (null character) to mark the end of the string.

/\* Infinite loop \*/

while (1)

{

MX\_USB\_HOST\_Process(); // Process USB host functions. This may be used to communicate with connected USB devices.

strcpy(message,"Hello\n \r");

// Copy the string "Hello\n \r" into the buffer `message`. "\n" is a newline, and "\r" is a carriage return.

HAL\_UART\_Transmit(&huart3 , (uint8\_t \*)message , strlen(message) , 100);

// Transmit the content of the `message` buffer via UART. UART3 is used, and the message is cast to a uint8\_t\* (pointer to //data).

// The number of characters sent is determined by `strlen(message)`, and the timeout is set to 100ms.

HAL\_Delay(1000); // Introduce a 1000ms (1 second) delay between transmissions.

}

1. Install MobaXterm.

<https://www.youtube.com/watch?v=36UeCSEJHNo>

1. Build and run the code and you should see Hello in the terminal

**Overwriting printf()**

1. Create a new file in Src ( your\_proj -> Core -> Src )

And put that code in it :

#include "stm32f4xx\_hal.h" // Include the header file for the STM32 HAL (Hardware Abstraction Layer) library, specific to the STM32F4 series.

extern UART\_HandleTypeDef huart3; // Declare the UART handle for UART3 as external, meaning it’s defined elsewhere in the project.

int \_write(int file, char \*data, int len) // Define the \_write function, which is typically used to redirect printf() output to a custom destination (e.g., UART).

{

HAL\_UART\_Transmit(&huart3, (uint8\_t \*) data, len, HAL\_MAX\_DELAY);

// Transmit the data over UART3. The `data` pointer is cast to `uint8\_t\*` because HAL functions expect this type.

// The `len` parameter specifies the number of bytes to send, and `HAL\_MAX\_DELAY` ensures the function waits indefinitely for the transmission to complete.

return len; // Return the length of the data sent, which informs the calling function how many bytes were successfully transmitted.

}

1. Save it and you could print in the terminal using printf();