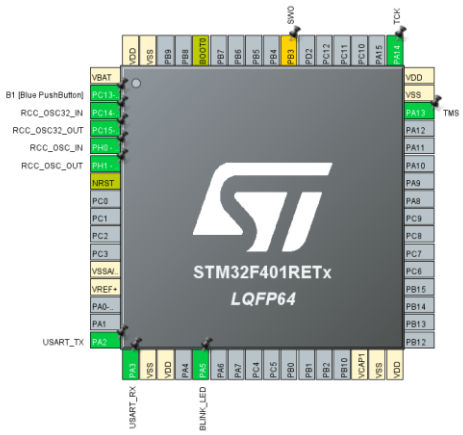
LTOS – Asynchronous USART Communication Example with NUCLEO-F401RE

Before start to explain this project, I strongly recommend to read document in LedBlinkPrj_NucleoF401RE folder to learn adding LTOS to your project. This project was created with STM32CubeIDE version: 1.7.0

1- Asynchronous USART Communication

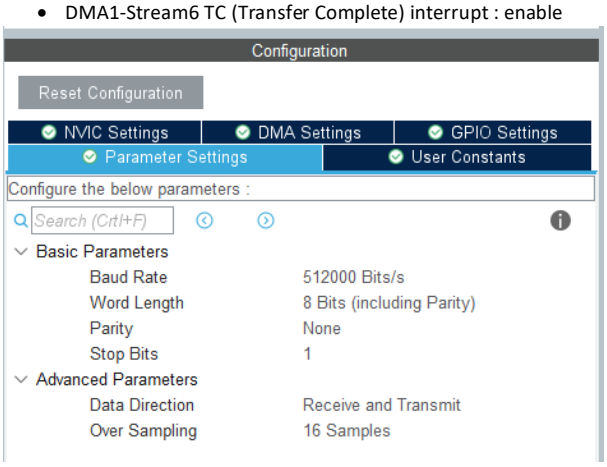
UART is very famous communication type in embedded systems. But, pooling UART for waiting bytes is so dangerous for real-time projects. So, it is very important to design UART communication to don't make busy processor by sending or receiving byte(s). Thanks to DMA, we don't need to worry about this problem anymore.

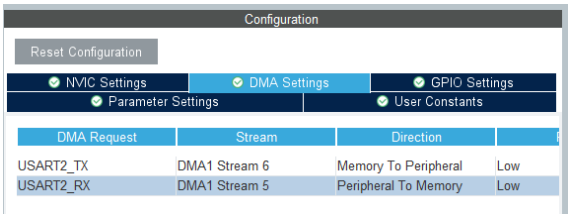


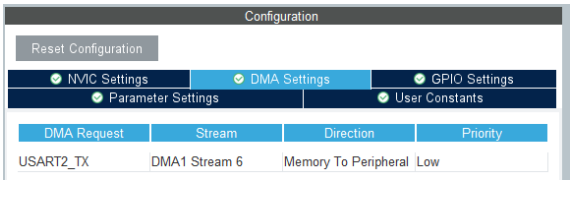
2- USART configuration

In Nucleo-F401RE development board, USART2 is used and its pins are PA.2 (USART-TX) and PA.3 (USART-RX). USART2 configuration:

- 512000 Baud Rate, 8 Data bits, 1 Stop bit, No Parity
- USART2-RXNE (RX Not Empty) interrupt : Enable
- DMA1-Stream6 set for USART2-Transmition
- DMA1-Stream6 TC (Transfer Complete) interrupt : enable







3- Asynchronous Communication

Asynchronous Communication is created to send/receive byte(s) through UART. DMA is used to transmit byte(s) and RX interrupt to get byte(s). There are 2 basic functions:

Name	Brief	Parameter(s)		Return
ASC_tx	Transmit byte(s)	**ptx**	Transmit buffer pointer	**ASC_XFER_SUCCESS** or error: **ASC_XFER_TX_LEN_ERR**, **ASC_XFER_TX_BUSY_ERR**
		len	Transmit buffer length	
ASC_rx	Receive byte(s)	**pcb**	pointer of callback function	**ASC_XFER_SUCCESS** or error: **ASC_XFER_RX_LEN_ERR**, **ASC_XFER_RX_CALLBACK_ERR**, **ASC_XFER_RX_BUSY_ERR**
		len	Receive length	
		tm	Time in LTOS time resolution.	

In addition, in ASC module, ascTASK added to call rx callback function set by **pcb** parameter in the ASC_rx. By this way, RX interrupt execution kept minimum. After finishing collection byte(s), **ascOS** attached with time set by **tm** parameter in the ASC_rx function.