POLITECNICO
DI MILANO

www.polimi.it

STM32 – USART

Prof. Federica Villa

USART

Universal Synchronous and Asynchronous Receiver-Transmitter

=

type of a serial interface device that can be programmed to communicate asynchronously or synchronously.

COM ports provide a convenient way for PCs and embedded systems to exchange information.

The traditional COM port on a PC is an RS-232 serial port. Recent PCs often skip RS-232 in favor of USB.

Nevertheless a USB device can appear as a virtual COM port that applications can access using COM-port APIs or libraries.

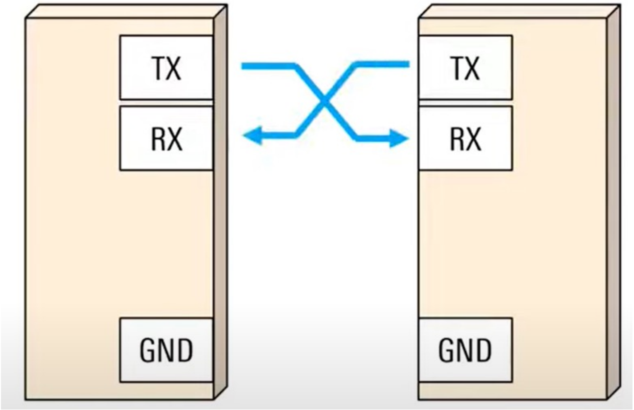
Many existing devices with asynchronous serial ports can use a USB UART to communicate with PCs as a virtual COM.

What is UART?

Protocol for **exchanging serial data between two devices**

Uses only **2 wires**: **TX → RX**, in each direction, plus a common ground

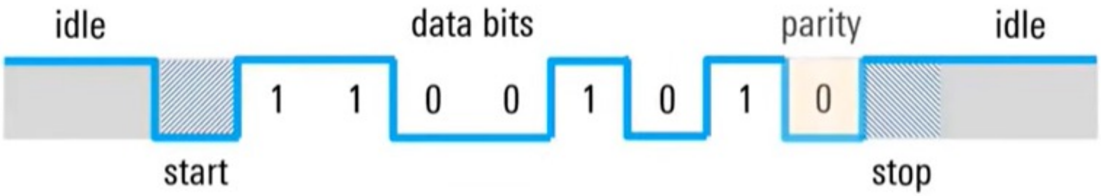
Transmitter and receiver **do not share a common clock**: they must **agree** on the speed (**Baud rate**) and **parameters** of the transmission



Data is transmitted as “frames”:

- **Start/stop** bits
- **Data** bits (5 to 9, usually 7 or 8, sent LSB first)
- **Parity** bit (optional) useful for error detection:

  Even parity: ‘1’ if number of ‘1’ is even / **Odd parity**: ‘1’ if number of ‘1’ is odd.



UART HAL functions

There are many HAL functions for UART.

Basics functions:

HAL_StatusTypeDef **HAL_UART_Receive**(UART_HandleTypeDef *huart, uint8_t *pData, uint16_t Size, uint32_t Timeout)

HAL_StatusTypeDef **HAL_UART_Transmit**(UART_HandleTypeDef *huart, uint8_t *pData, uint16_t Size, uint32_t Timeout)

Direct Memory Access functions:

HAL_StatusTypeDef **HAL_UART_Receive_DMA**(UART_HandleTypeDef *huart, uint8_t *pData, uint16_t Size)

HAL_StatusTypeDef **HAL_UART_Transmit_DMA**(UART_HandleTypeDef *huart, uint8_t *pData, uint16_t Size)

Project

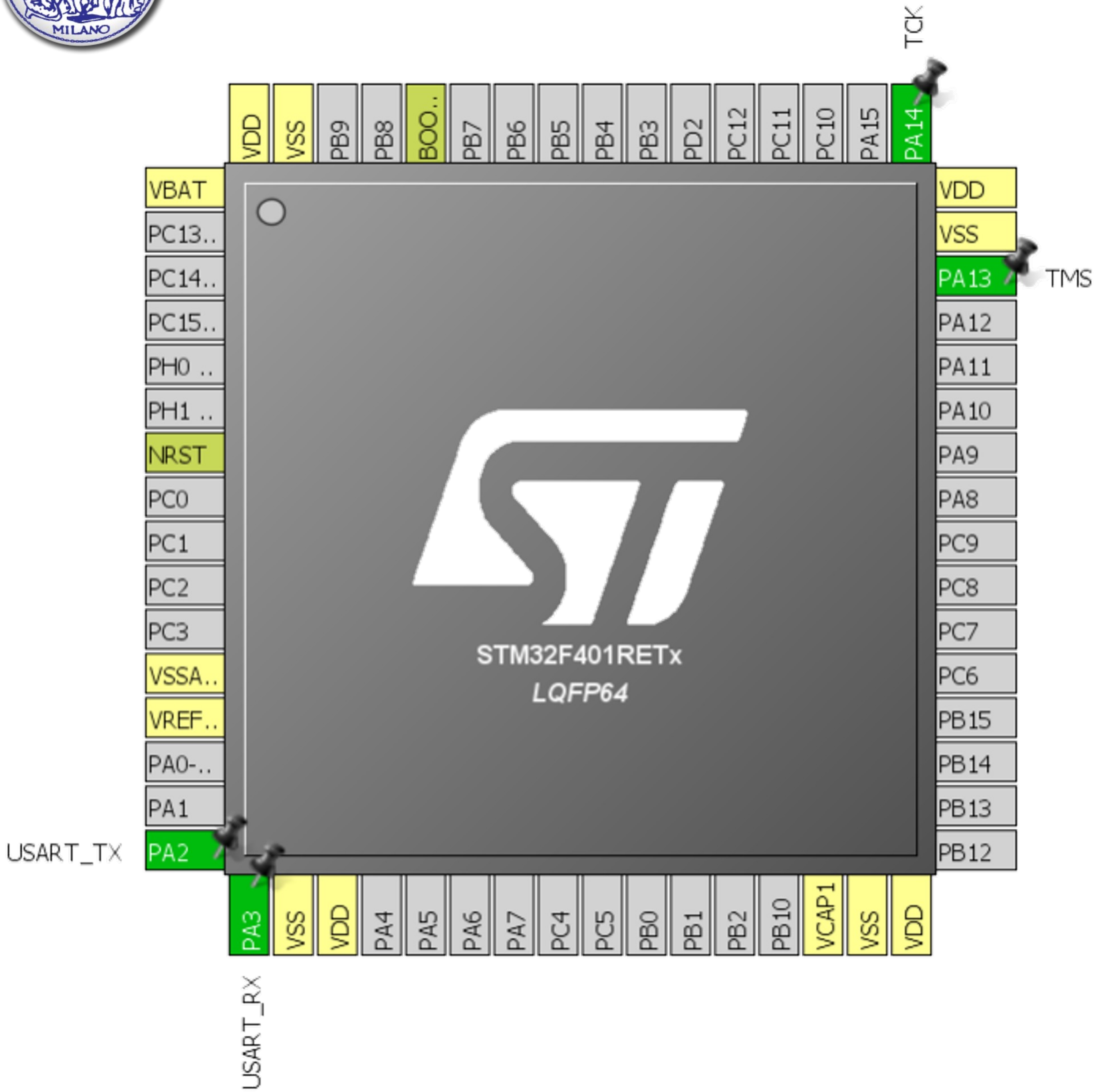
Objective of this project is
send information
from the microcontroller to the PC,
using the USART interface for the Virtual COM.

You will send a string containing your name and your year of birth followed by a new line every one second.

For receiving the string we will use a terminal emulator (PuTTY).

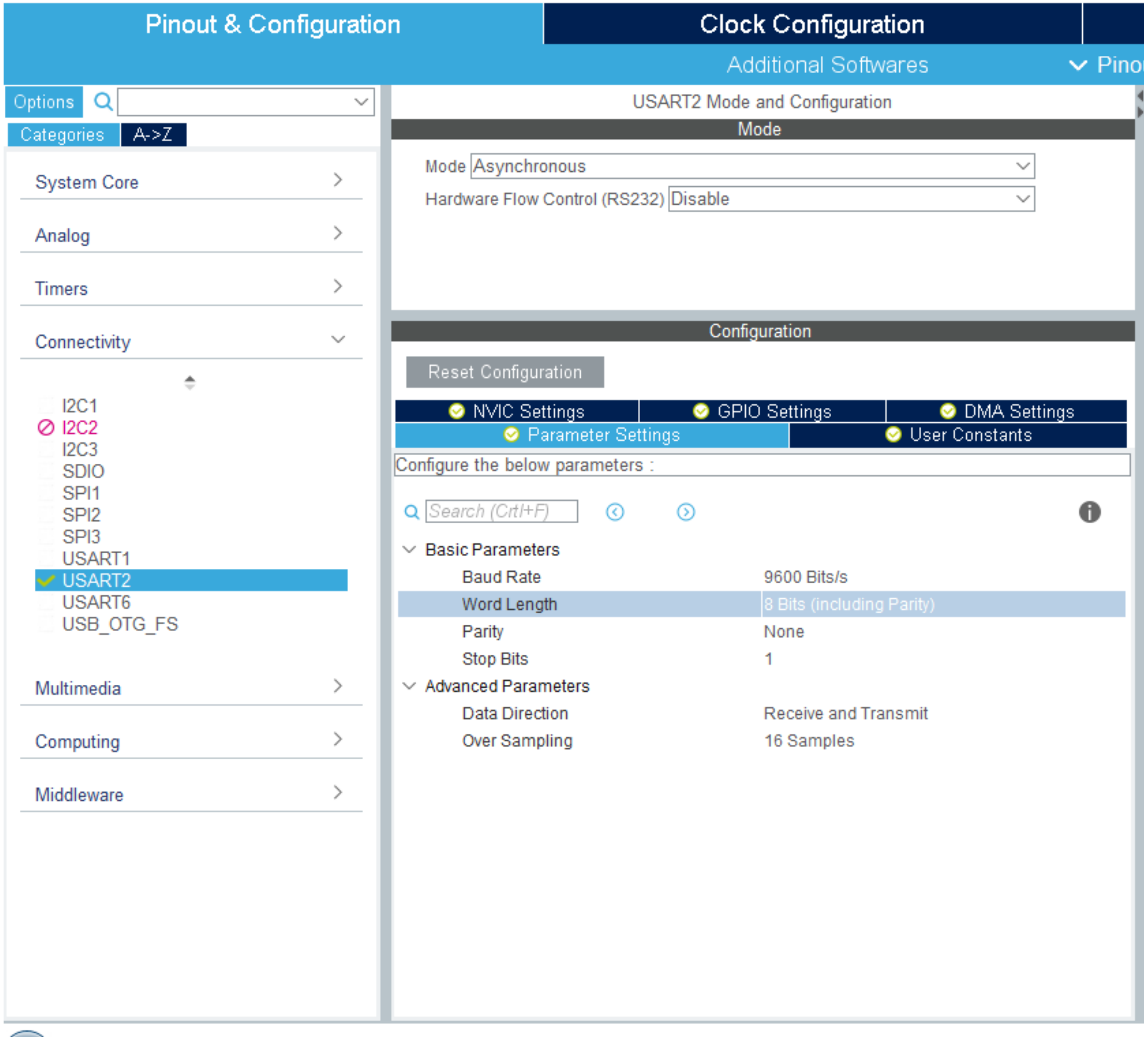
You can download PuTTY here: https://www.putty.org/

CUBE project Pinout



- Open a new project (use default)
- Check that the two pins for USART are enabled

USART configuration



USART2 Mode and Configuration

Mode

Mode: Asynchronous

Hardware Flow Control (RS232): Disable

Configuration

Configure the below parameters :

Basic Parameters

Baud Rate	9600 Bits/s
Word Length	8 Bits (including Parity)
Parity	None
Stop Bits	1

Advanced Parameters

Data Direction	Receive and Transmit
Over Sampling	16 Samples

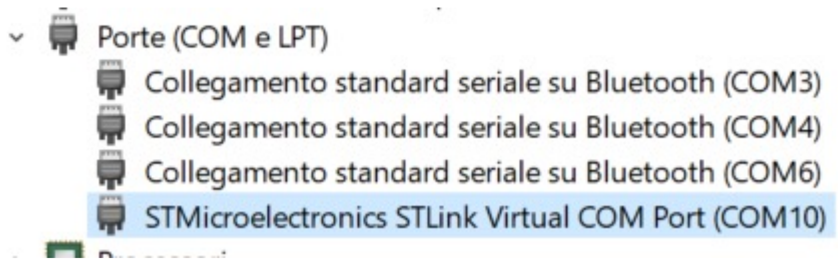
- Select a Baud Rate (e.g. 115200 Bits/s).

  The same BR should be set in PuTTY, since it is an asynchronous communication!

Identify the COM port on your PC

Open “Device Manager” (Gestione dispositivi) by right-clicking on the start menu, and selecting “Device Manager”

Scroll down until “Ports (COM and LPT)” and open the submenu. The COM port we need should be identified by the name “STMicroelectronics STLink Virtual COM Port (COM__)”



Porte (COM e LPT)

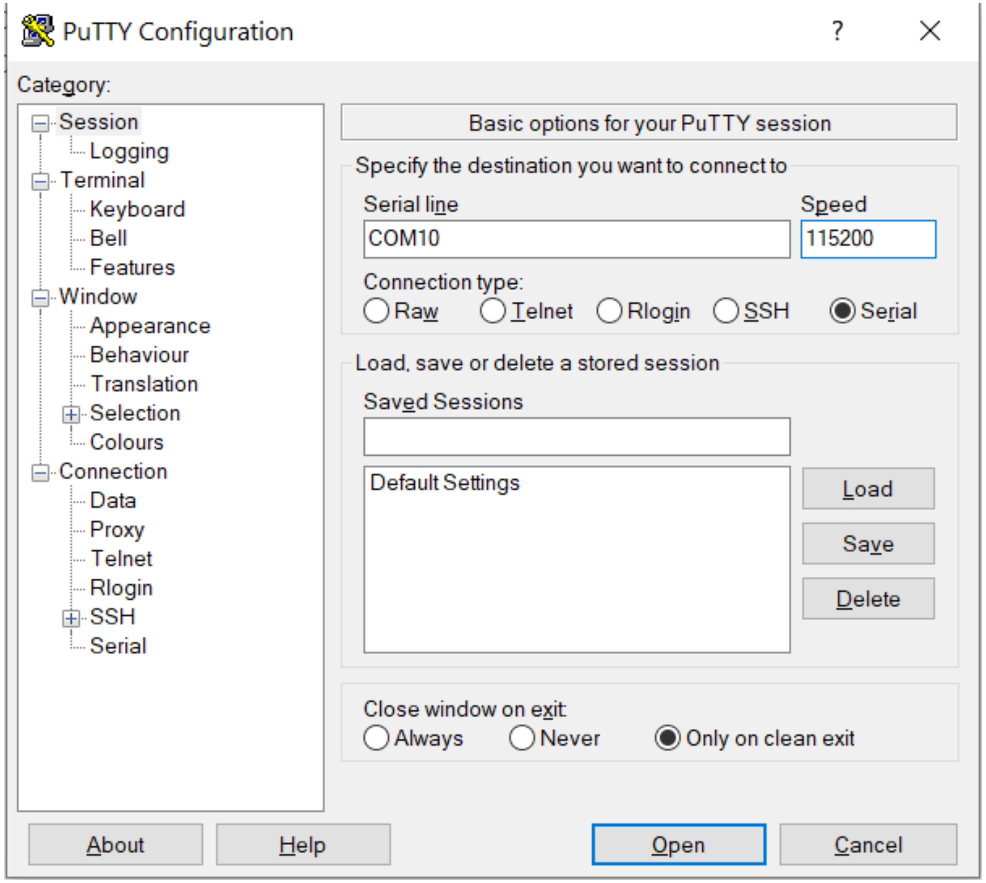
Collegamento standard seriale su Bluetooth (COM3)

Collegamento standard seriale su Bluetooth (COM4)

Collegamento standard seriale su Bluetooth (COM6)

STMicroelectronics STLink Virtual COM Port (COM10)

Debug



Basic options for your PuTTY session

Specify the destination you want to connect to

Serial line: COM10

Speed: 115200

Connection type: ○ Raw ○ Telnet ○ Rlogin ○ SSH ● Serial

Load, save or delete a stored session

Saved Sessions:

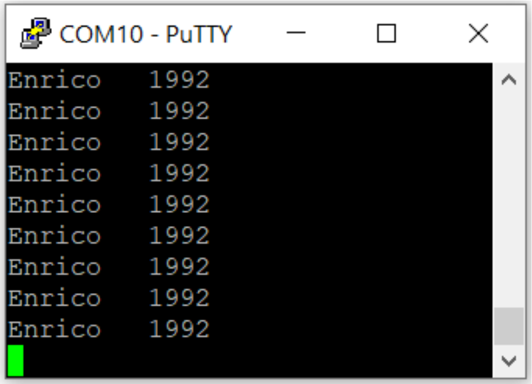
Default Settings

Close window on exit: ○ Always ○ Never ● Only on clean exit

- Start PuTTY to receive data, use it as “Serial” and type the COM reserved on your pc to the STM32. Select the same speed you used for the UART.

- Compile and debug you code.

- This is the output you should see:



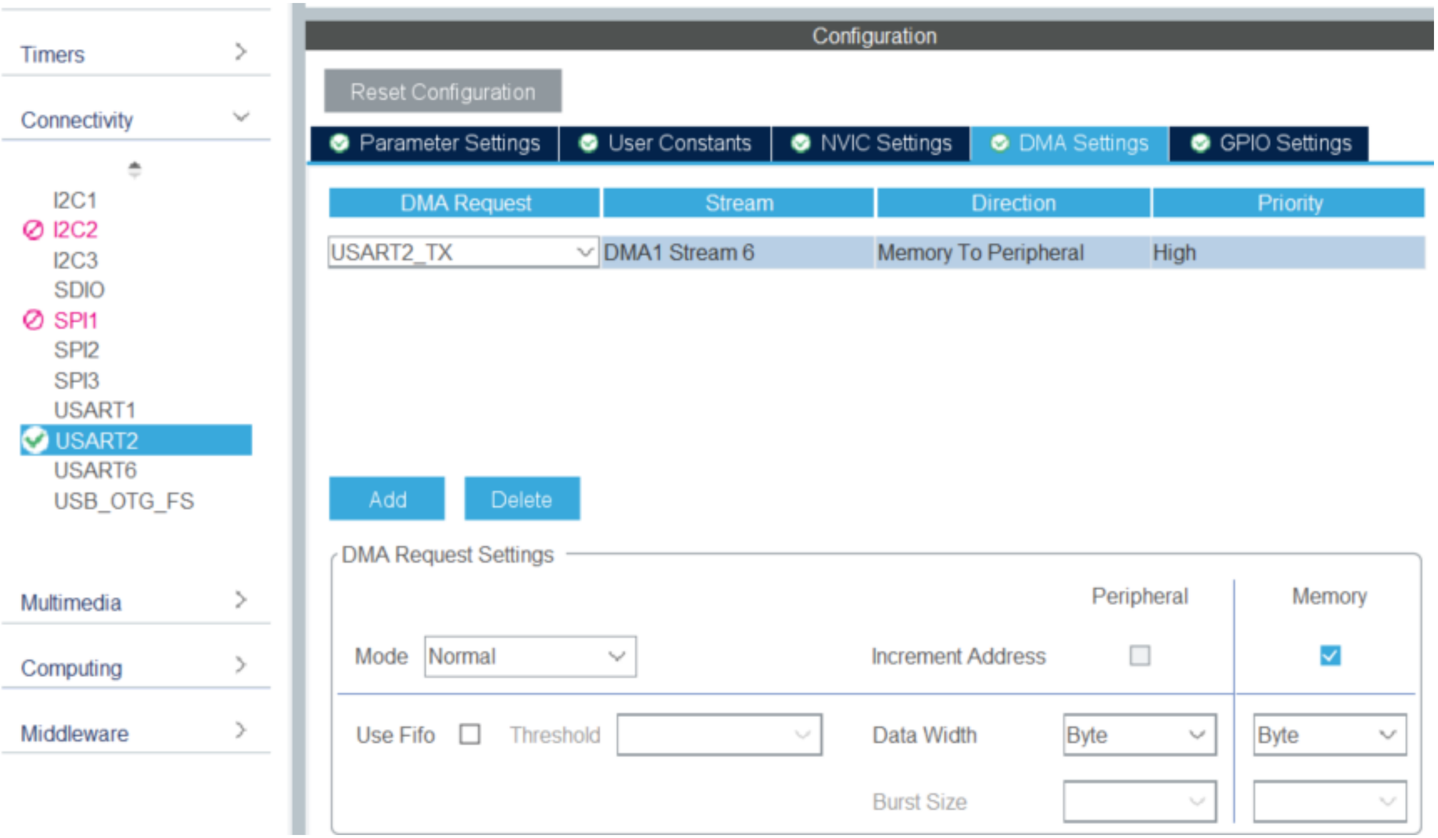
```
Enrico    1992
Enrico    1992
Enrico    1992
Enrico    1992
Enrico    1992
Enrico    1992
Enrico    1992
Enrico    1992
Enrico    1992
```

Project - DMA

Repeat the previous project, using DMA functions

Project – DMA: Setup

Enable DMA in the USART2 peripheral with USART2_TX requests, in normal mode



DMA Request	Stream	Direction	Priority
USART2_TX	DMA1 Stream 6	Memory To Peripheral	High

DMA Request Settings

			Peripheral	Memory
Mode	Normal	Increment Address	□	☑
Use Fifo □	Threshold	Data Width	Byte	Byte
		Burst Size		

Conclusion

We will reuse the same code in the next project in which we will send the ADC result through the Virtual COM and display it to PuTTY terminal.

The output of an ADC conversion is a uint32_t variable.