**Passos para integrar o micro-ROS à STM32CubeIDE**

**Referência:** https://github.com/micro-ROS/micro\_ros\_stm32cubemx\_utils

**Criando um projeto micro-ROS na STM32CubeIDE**

1. Instale o STM32Cube IDE, o Docker e o Git (Linux)
2. Crie um projeto padrão para a sua plataforma no STM32CubeIDE
3. Clone o repositório do [micro\_ros\_stm32cubemx\_utils](https://github.com/micro-ROS/micro_ros_stm32cubemx_utils) na pasta do projeto do STM32CubeIDE

*# Abra o terminal no diretório do projeto (para facilitar, pela interface gráfica do Linux navegue até o diretório do projeto, clique com o botão direito do mouse em uma área vazia e em “Abrir no terminal”*

sudo git clone https://github.com/micro-ROS/micro\_ros\_stm32cubemx\_utils.git

1. No STM32CubeIDE vá em Project -> Settings -> C/C++ Build -> Settings -> Build Steps Tab e em pre-build steps adicione:

docker pull microros/micro\_ros\_static\_library\_builder:foxy && docker run --rm -v "${workspace\_loc:/${ProjName}}":/project --env MICROROS\_LIBRARY\_FOLDER=micro\_ros\_stm32cubemx\_utils/microros\_static\_library\_ide microros/micro\_ros\_static\_library\_builder:foxy

Esse código executará o docker que organizará os arquivos fontes necessários para utilizar o micro-ROS no STM32CubeIDE.

**Atenção:** após executar esse Docker pela primeira vez (compilar o projeto pela primeira vez), esse comando pode ser excluído.

1. Adicione o caminho do micro-ROS nos “includes”. Vá em Project -> Settings -> C/C++ Build -> Settings -> Tool Settings Tab -> MCU GCC Compiler -> Include paths e adicione:

${PWD}/../micro\_ros\_stm32cubemx\_utils/microros\_static\_library\_ide/libmicroros/include

1. Adicione a biblioteca pré-compilada do micro-ROS. Vá em Project -> Settings -> C/C++ Build -> Settings -> MCU GCC Linker -> Libraries:

* Adicione em Libraries (-l)

microros

* Adicione em Library search path (-L)

${PWD}/../micro\_ros\_stm32cubemx\_utils/microros\_static\_library\_ide/libmicroros

1. Adicione os seguintes arquivos de código-fonte ao seu projeto (arrastando-os para a pasta de destino na IDE STM32CubeIDE):

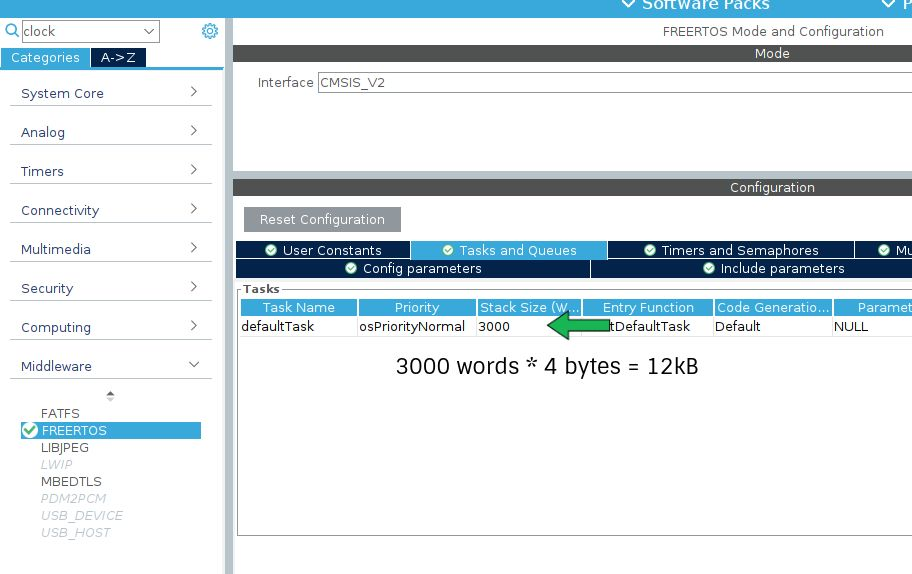
extra\_sources/microros\_time.c

extra\_sources/microros\_allocators.c

extra\_sources/custom\_memory\_manager.c

extra\_sources/microros\_transports/dma\_transport.c

1. Certifique-se de que, se estiver usando FreeRTOS, a tarefa micro-ROS tenha mais de 10 kB (3000 words) de *stack*:



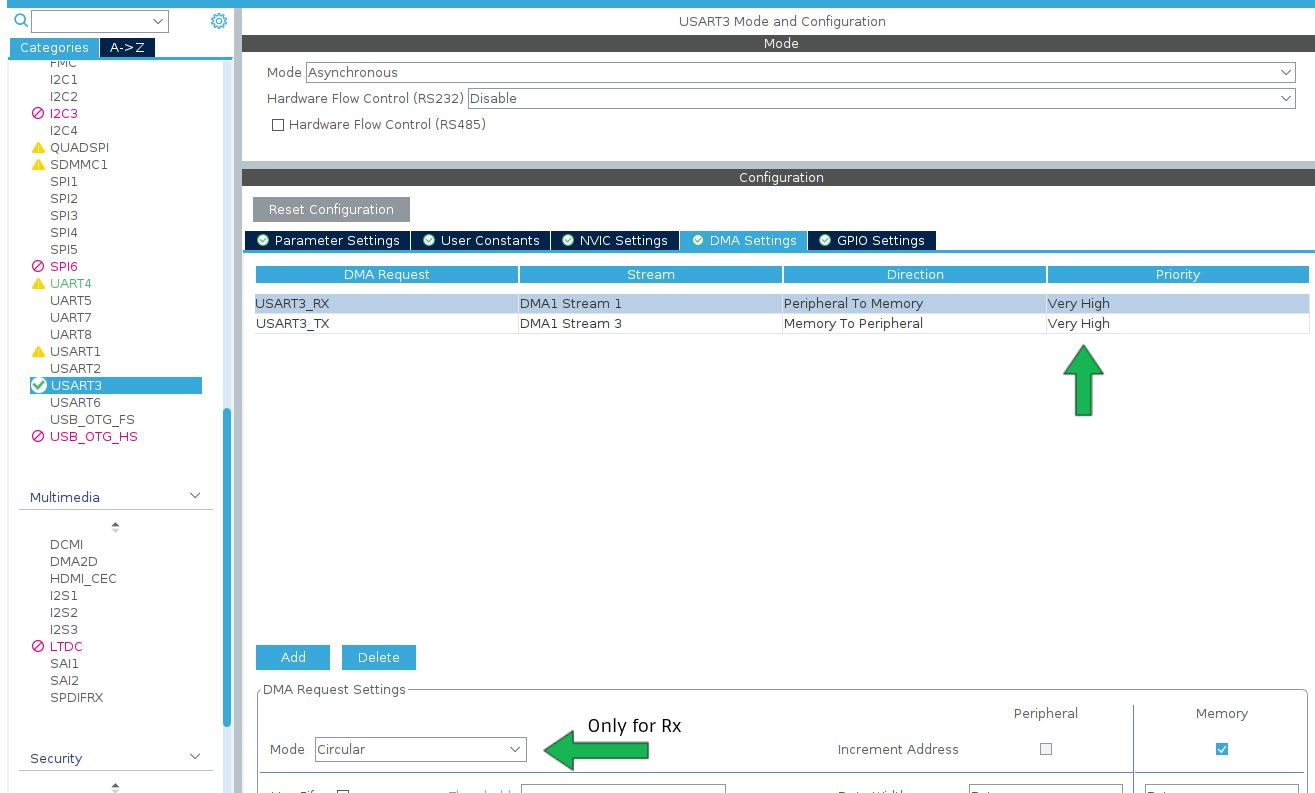
1. Configure a interface de transporte no projeto STM32CubeMX, verifique a seção “**Configurando a camada de transporte do micro-ROS na STM32CubeIDE”** para obter instruções sobre as camadas de transportes personalizadas fornecidas.
2. Crie e execute seu projeto

**Configurando a camada de transporte do micro-ROS na STM32CubeIDE**

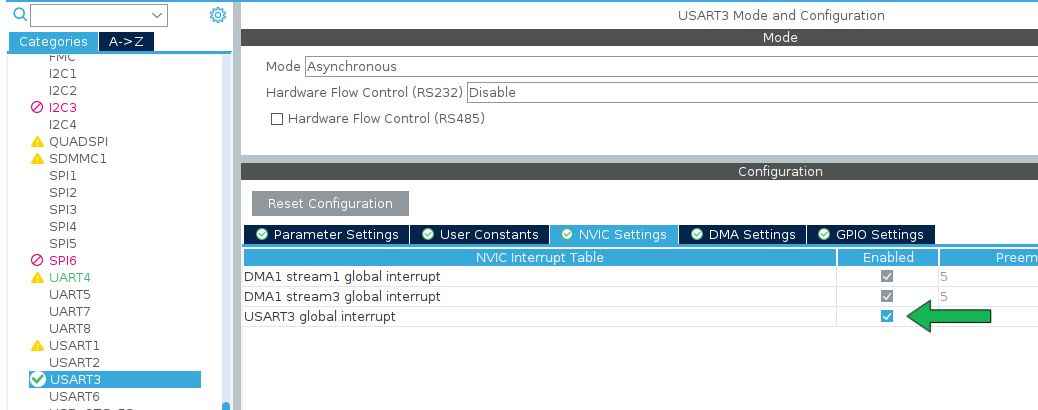
No [micro\_ros\_stm32cubemx\_utils](https://github.com/micro-ROS/micro_ros_stm32cubemx_utils) são disponibilizados dois arquivos para configuração da camada de transporte: “**UART com DMA”** e “**UART com interrupções”**

**UART com DMA**

1. Habilite a UART no STM32CubeMX
2. Para a UART selecionada, habilite o DMA para Tx e Rx em DMA Settings
3. Defina a priotity DMA como Very High para Tx e Rx
4. Defina o modo DMA como Circular para Rx:



1. Habilite a global interrupt da UART utilizada em NVIC Settings:



**UART com interrupções**

1. Habilite a UART no STM32CubeMX
2. Habilite a global interrupt da UART utilizada em NVIC Settings:

