

Esse é o quarto artigo de uma nova série escrita pelo engenheiro Ismael Lopes da Silva, exclusivamente para o site "[www.embarcados.com.br](http://www.embarcados.com.br)". Nessa série focarei no Microcontrolador da STMicroelectronics, o MCU STM32F103C8T6, que é um ARM Cortex-M3. Os pré-requisitos para uma boa compreensão dos artigos é ter o domínio da Linguagem C Embedded e conceitos de eletrônica.

## Usando o STM32CubeIDE como imprimiremos "Hello World" no Target?

Como vamos imprimir uma mensagem no dispositivo target (MCU), se não temos um Display conectado a placa/MCU? Isso será explicado detalhadamente nesse artigo, então, vamos editar o arquivo main.c para termos uma aplicação mínima, conforme mostrado a seguir. Apenas adicionamos um cabeçalho para funções padrões em "C", e também a função "printf".

```
/* *****  
 * @file      : main.c  
 * @author    : Auto-generated by STM32CubeIDE  
 * @brief     : Main program body  
 *****  
 * @attention  
 *  
 * <h2><center>&copy; Copyright (c) 2019 STMicroelectronics.  
 * All rights reserved.</center></h2>  
 *  
 * This software component is licensed by ST under BSD 3-Clause license,  
 * the "License"; You may not use this file except in compliance with the  
 * License. You may obtain a copy of the License at:  
 *      opensource.org/licenses/BSD-3-Clause  
 *  
 ***** */
```

```
#if !defined(__SOFT_FP__) && defined(__ARM_FP)  
#warning "FPU is not initialized, but the project is compiling for an FPU. Please initialize the FPU  
before use."  
#endif
```

```
#include <stdio.h>
```

```
int main(void)  
{  
    printf("Hello World\n");  
    for(;;);  
}
```

A solução vem do processador ARM Cortex-M3/M4/M7 ou mais recente. Nesses processadores podemos fazer a função "printf" trabalhar, usando o pino SWO da interface de depuração SWD. SWO significa "Serial Wire Output" e SWD significa "Serial Wire Debug". SWD é um protocolo a dois fios (SWIO e SWCLK) para acessar a interface ARM de depuração, e um fio para acessar o recurso de rastrear (Trace), usando a linha SWO.

Dentro do nosso processador ARM Cortex-M3, há um periférico chamado ITM (Instrumentation Trace Macrocell). ITM é uma fonte para rastrear, usando a função "printf". Podemos rastrear eventos da aplicação e também pode gerar diagnóstico de informação do sistema.

O conector SWD tem três pinos, na qual dois pinos são usados para depurar e um é usado para rastrear (trace). Rastrear (trace) significa obter informação/mensagem do processador. Usando a interface SWD podemos programar a memória Flash do MCU, podemos acessar regiões de memória, adicionar breakpoints, rodar e parar a CPU e também podemos usar a SWV (Serial Wire Viewer) para usar "printf" no processo de rastreamento (tracing).

Dentro do ITM há um buffer serial (FIFO), então, o que faremos é escrever uma mensagem (alguns bytes) usando a função "printf", e colocaremos dentro desse buffer. A saída desse buffer é conectada ao pino SWO, que está conectado ao circuito do ST Link V2, e que está conectado ao target (MCU), então, podemos capturar a mensagem usando nosso STM32CubeIDE. Nem todas as ferramentas de depuração permitem usar esse recurso, mas, o STM32CubeIDE tem disponível essa funcionalidade.

### Modificações de software para capturar "printf"

Na pasta "Src", no "Project Explorer", temos um arquivo denominado syscalls.c, que deve ser editado para que podemos usar a função "printf", portanto, após as diretivas dessa função adicione o seguinte trecho de programa. Basicamente nesse trecho temos outras diretivas e a função ITM\_SendChar, que habilitará o uso da função "printf".

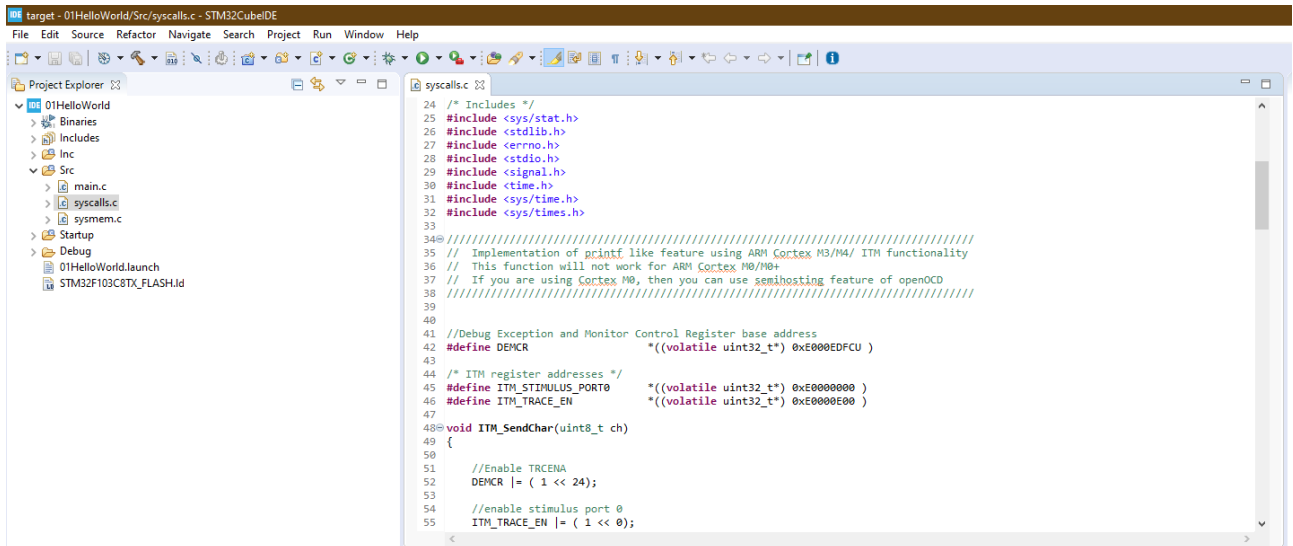
```
////////////////////////////////////
//      Implementation of printf like feature using ARM Cortex M3/M4/ ITM functionality
//      This function will not work for ARM Cortex M0/M0+
//      If you are using Cortex M0, then you can use semihosting feature of openOCD
////////////////////////////////////

//Debug Exception and Monitor Control Register base address
#define DEMCR                *((volatile uint32_t*) 0xE000EDFCU )

/* ITM register addresses */
#define ITM_STIMULUS_PORT0    *((volatile uint32_t*) 0xE0000000 )
#define ITM_TRACE_EN          *((volatile uint32_t*) 0xE0000E00 )

void ITM_SendChar(uint8_t ch)
{
    //Enable TRCENA
    DEMCR |= ( 1 << 24);
    //enable stimulus port 0
    ITM_TRACE_EN |= ( 1 << 0);
    // read FIFO status in bit [0]:
    while(!(ITM_STIMULUS_PORT0 & 1));
    //Write to ITM stimulus port0
    ITM_STIMULUS_PORT0 = ch;
}
```

O trecho acima encaixamos conforme ilustrado na figura 1.



**Figura 1** – Edição da função ITM\_SendChar do arquivo syscalls.c

Ainda no arquivo syscalls.c, temos que modificar a função “\_write”, conforme mostrado a seguir, e ilustrado na figura 2.

```

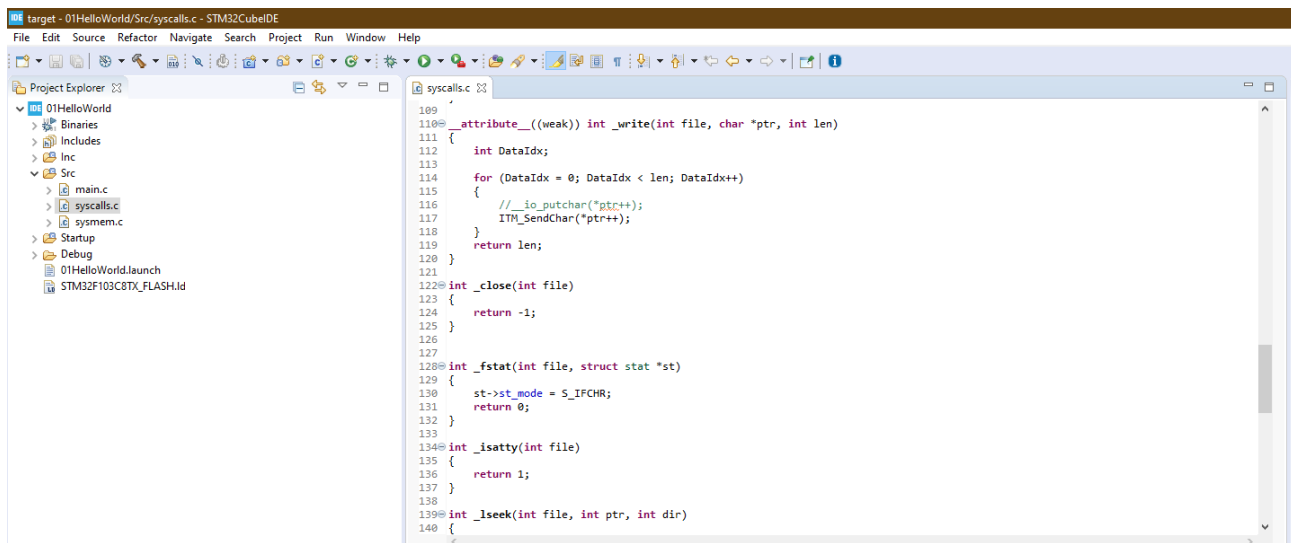
__attribute__((weak)) int _write(int file, char *ptr, int len)
{
    int DataIdx;

    for (DataIdx = 0; DataIdx < len; DataIdx++)
    {
        //_io_putchar(*ptr++);
        ITM_SendChar(*ptr++);
    }

    return len;
}

```

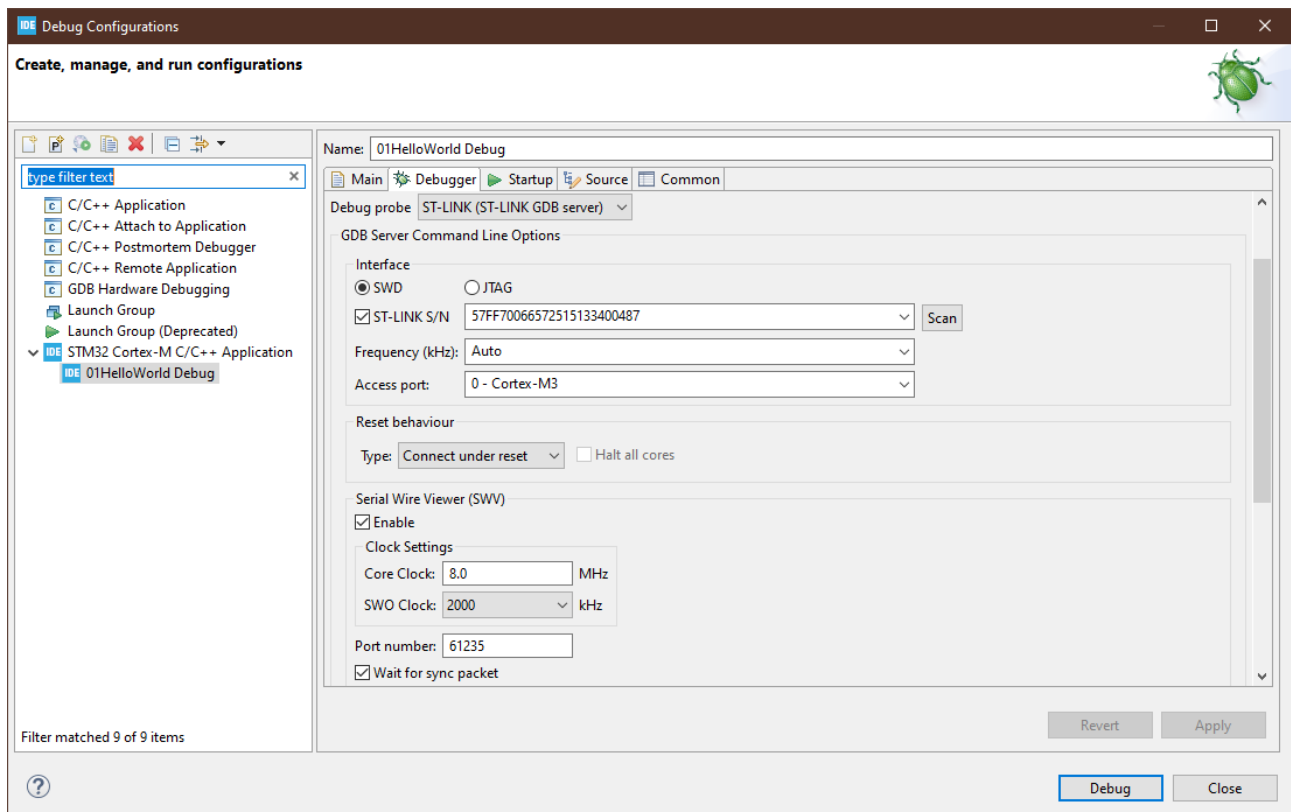
Após as modificações salve o arquivo syscalls.c.



**Figura 2** – Edição da função “\_write” do arquivo syscalls.c

Então, como funcionará? A biblioteca padrão implementada chamará a função “\_write” que foi modificada em syscalls.c. A função “\_write” é chamada, e a mensagem é recebida através de um ponteiro, portanto, apenas enviamos os dados para o FIFO do ITM. A função escreve para dentro do buffer (FIFO) e do FIFO os dados vem através da linha SWO, passando pelo circuito do ST-LINK V2 e capturada pelo STM32CubeIDE, sendo imprimida no console do SWV (Serial Wire Viewer).

Na janela “Projetc Explorer”, clique com o botão direito do mouse sobre o projeto “01HelloWorld” e selecione “Debug As” e depois “Debug Configuration”. Selecione a aba [Debbuger] e habilite a Serial Wire Viewer (SWV), conforme ilustrado na figura 3. O core clock deve ser 8,0MHz, porque ainda não configuramos o clock do sistema. Depois clique no botão [Apply] e no [Debug].



**Figura 3** – Configurar o Serial Wire Viewer (SWV)

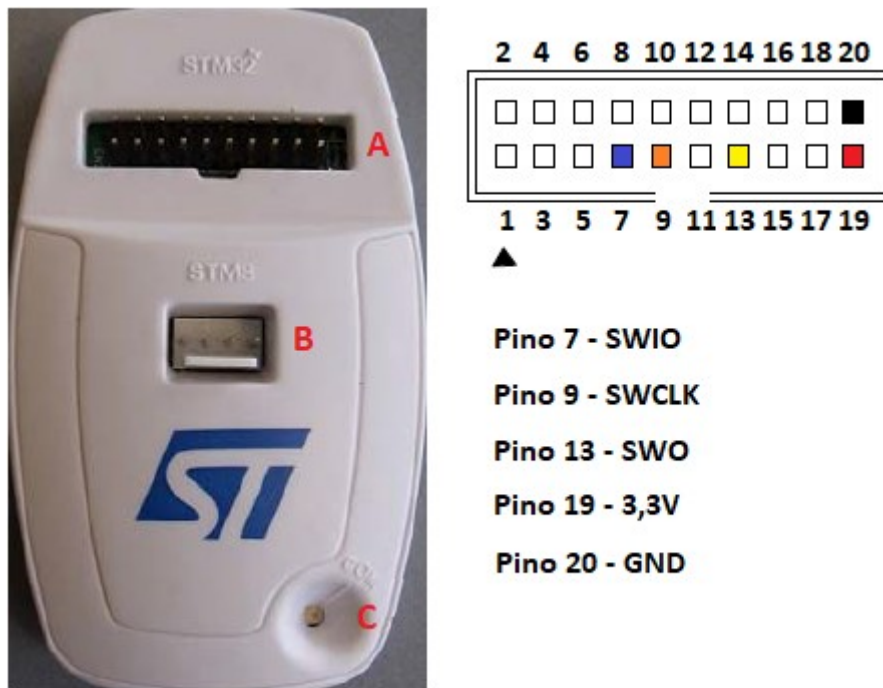
### Modificações de hardware para capturar “printf”

O ST-LINK V2 é um programador/depurador para os microcontroladores STM8 e STM32. As interfaces SWIM (Single Wire Interface Module) e JTAG/SWD (Serial Wire Debugging) facilitam a comunicação com qualquer microcontrolador STM8 ou STM32.



**Figura 4** – ST Link V2

A figura 5 ilustra os conectores e o LED que indica atividade de comunicação.



**Figura 5** – Conexões SWIO, SWCLK e SWO

A = Conector JTAG e SWD para MCU STM32

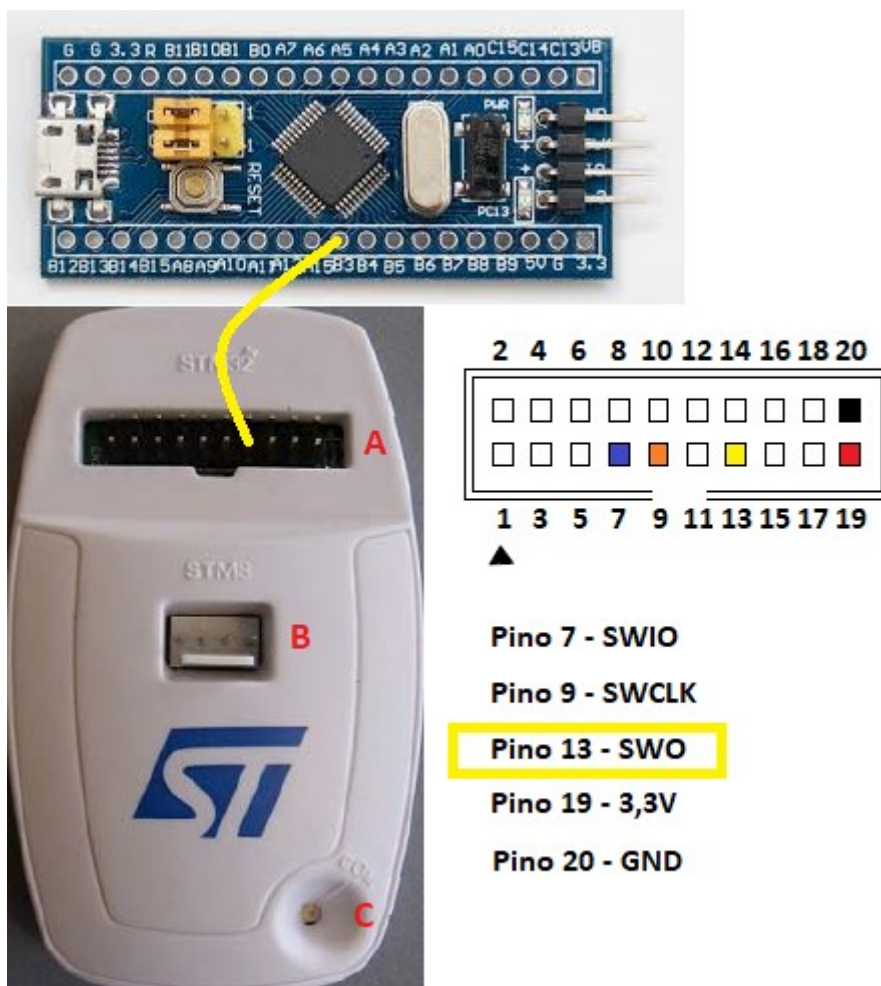
B = Conector SWIM para MCU STM8

C = LED de atividade de comunicação

Pino	Função ST LINK V2	Função SWD
1	MCU VDD (3,3V)	
2	MCU VDD (3,3V)	
3	JTAG TRST	
4	GND	
5	JTAG TDO	
6	GND	
7	JTAG TMS, SW IO	Depurar (Debugging)
8	GND	
9	JTAG TCK, SW CLK	Depurar (Debugging)
10	GND	
11	Não conectado	
12	GND	
13	JTAG TDI, SWO	Rastrear (Tracing)
14	GND	
15	NRST	
16	GND	
17	Não conectado	
18	GND	
19	VDD (3,3V)	
20	GND	

**Tabela 1** – Pinos do conector JTAG e SWD

O projeto da placa Blue Pill, que contém o nosso MCU STM32F103C8T6, não vem com a ligação da linha SWO entre o dispositivo target e o ST Link V2, portanto, temos que implementá-la para que a função "printf" seja capturada pelo STM32CubeIDE. Conectaremos o pino PB3 do MCU, que é o pino 39 do STM32F103C8T6, ao pino 13 do ST Link V2, conforme ilustrado na figura 6.



**Figura 6** – Conectando a linha SWO com o target (pino PB3)

### **Modificações de software e hardware prontas! Ajustes finais para capturar "printf"**

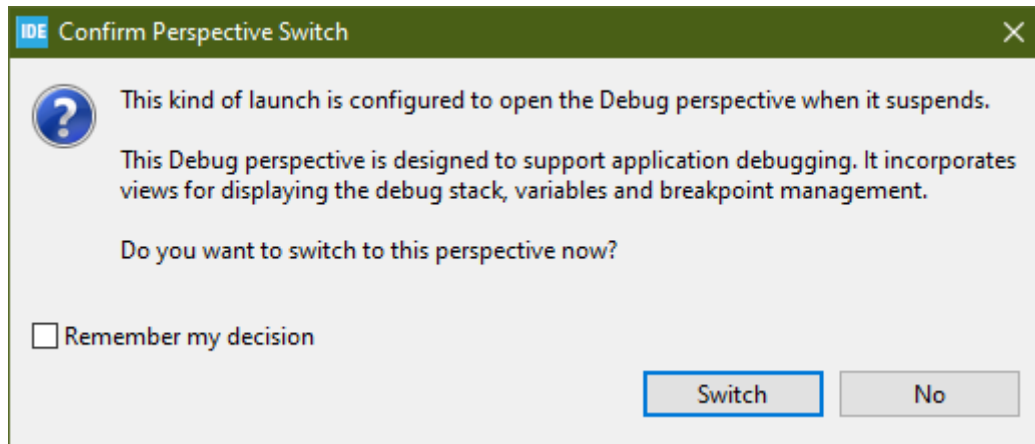
As mensagens que são capturadas através da função "printf" não são visualizadas na janela "Console". Temos que habilitar a visualização da janela "SWV ITM Console", e depois fazer algumas configurações finais para depurar e capturar as mensagens oriundas do "printf".

Para habilitar a janela "SWV ITM Console" a aplicação deve ser sendo depurada (debugging), porque se não estivermos nesse contexto, o STM32CubeIDE não nos dá as opções para configurá-la. Vamos novamente limpar, compilar e depurar a aplicação para finalizar os passos por completo.

Segue a sequência para deixarmos a aplicação em ordem:

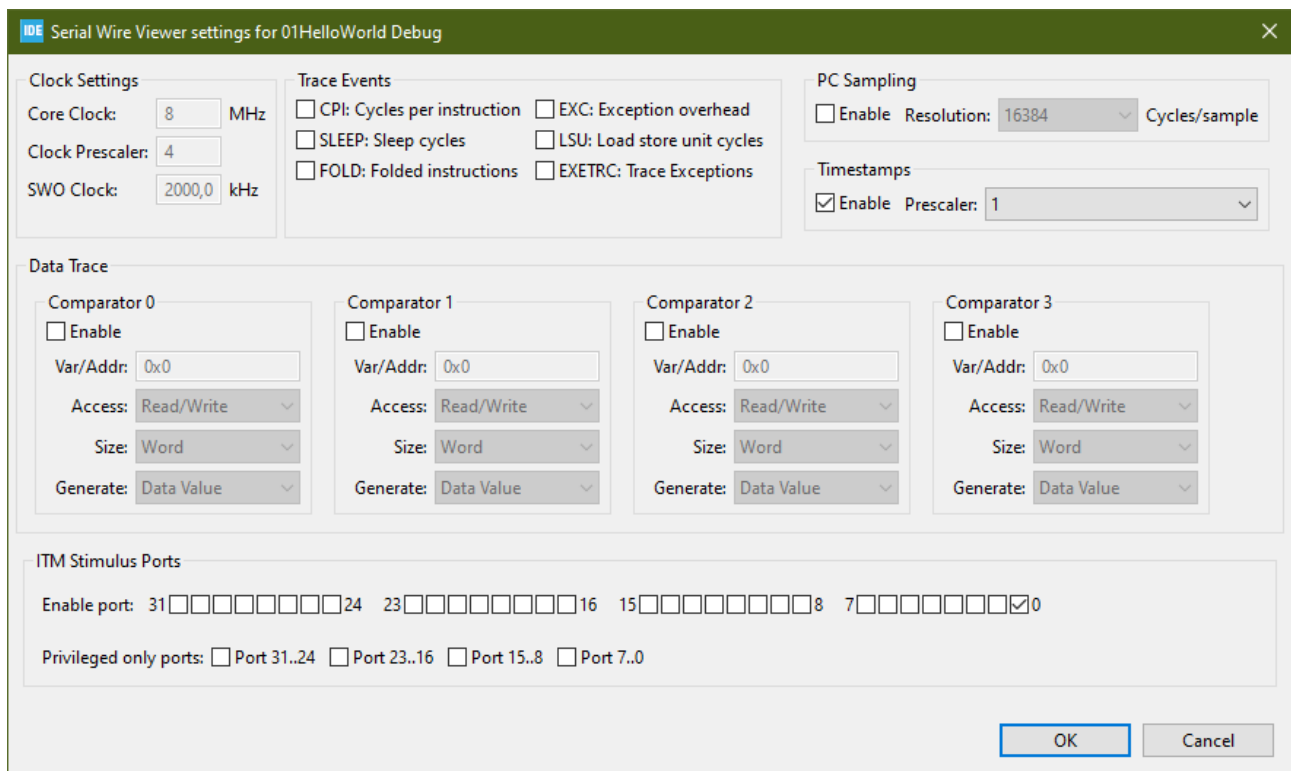
1. Na janela "Projetc Explorer", clique com o botão direito do mouse sobre o projeto "01HelloWorld" e selecione "Clean Project". Isso limpa a compilação anterior;
2. Na janela "Projetc Explorer", clique com o botão direito do mouse sobre o projeto "01HelloWorld" e selecione "Build Project". Isso compila a aplicação;

3. Na janela "Projetc Explorer", clique com o botão direito do mouse sobre o projeto "01HelloWorld" e selecione "Debug As" e depois "STM32 Cortex-M C/C++ Application". Isso faz com que o STM32CubeIDE chavear para a perspectiva de depuração, conforme ilustrado na figura 7. Lembrando que já configuramos a ferramenta de depuração, sendo assim, habilitamos o SWV, conforme ilustrado na figura 3. Clique no botão [Switch] para depurar.



**Figura 7** – Perspectiva de depuração (debugging)

4. Na janela de código será mostrada o arquivo main.c, já destacando um breakpoint na linha onde temos a função 'printf("Hello World\n");'. No menu "Window", parte superior da tela, selecione "Show View", depois "SWV" e "SWV ITM Data Console". Observe que na parte inferior da tela, foi aberto a janela "SWV ITM Data Console";
5. Na janela "SWV ITM Data Console" tem um botão chamado "configure trace". É o primeiro botão. Clique nesse botão para configurarmos a SWV.



**Figura 8** – Configurar "SWV Serial Wire Viewer"

Na janela "SWV Serial Wire Viewer Settings", na seção inferior chamada "ITM Stimulus Ports", apenas selecione a porta 0, habilitando-a, conforme ilustrado na figura 8. Clique no botão [OK] para concluir a configuração;

6. Na janela "SWV ITM Data Console" tem um botão chamado "start trace". É o segundo botão. Clique nesse botão para iniciar o rastreamento (tracing), portanto, capturar a mensagem de "printf".



**Figura 9** – Janela "SWV ITM Data Console" porta 0

7. Para capturamos a mensagem pressione a tecla [F8] ou clique no botão "Resume", parte superior da tela. O processo de depuração será resumido, então, podemos ver que a mensagem foi capturada. Na janela "SWV ITM Data Console" podemos ver "Hello World", conforme ilustrado na figura 10.



**Figura 10** – Janela "SWV ITM Data Console" porta 0

8. Para encerrar o processo de depuração pressione a tecla [CTRL+F2] ou clique no botão "terminate", parte superior da tela.