# Usando SWO para monitoração e Debug

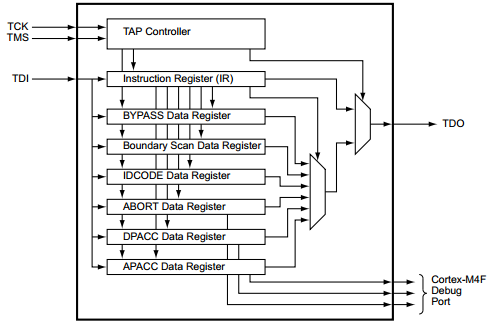
Introdução

SWD:

*Serial Wire Debug* (SWD) é uma interface alternative que utiliza apenas 2 pinos que utiliza o mesmo protocol JTAG. Ele também utiliza o retorno comum (terra/GND) da conexão existente. SWD utiliza o mesmo protocolo bi-direcional padrão das CPUs ARM, definido no documento (ref) ARM Debug Interface v5.[[20]](https://en.wikipedia.org/wiki/JTAG" \l "cite_note-20) This enables the debugger to become another [AMBA](https://en.wikipedia.org/wiki/Advanced_Microcontroller_Bus_Architecture) bus master for access to system memory and peripheral or debug registers. Data rate is up to 4 Mbytes/sec at 50 MHz. SWD also has built-in error detection. Nos dispositivos com JTAG e que dispõem de capacidade SWD, o TMS é usado como SWDIO e o TCL utilizado como SWCLK, permitindo uma funcionalidade dupla para os programadores.

Nesse trabalho vou abordar a função do sinal SWO dos processadores baseados em ARM que apresentam um conjunto de pinos para a função de configuração *in circuit*, que tentam ser o menos intrusivo possível durante a fase de depuração e desenvolvimento.

Serão apresentados as implementações em diferentes modelos de MCU e as diferentes configurações necessárias para poder utilizar esse sinal.



|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **EK-TM4C123GXL** | **MSP432** | **FRDM KL25** | **Arduino Due** | **BluePill** | **STM32F407** | **XNUCLEO** |
| Fabricante | Texas Instrument | Texas Instrument | NXP | Microchip | ST | ST | ST |
| MCU | TM4C123GH6PM | MSP432P401R | KL25Z128VLK4 | SAM3X8E | STM32F103C8 | STM32F407ZG | STM32F103RB |
| Encapsulamento | 64 LQFP | 100 PZ | 80 LQFP | 144 LQFP | 48 LQFP |  | 64 LQFP |
| Família | Cortex M4F | Cortex M4F | Cortex M0+ | Cortex M3 | Cortex M3 | Cortex M4 | Cortex M3 |
| Flash | 256 K | 256 K | 128 K | 512 K | 64 K | 1024 K | 128 K |
| SRAM | 32 K | 64 K | 16 K | 96 K | 20 K | 192K + 4K | 20 K |
| EEPROM | 2 K | 32 K | - | - | - | - | - |
| Max Clock | 80 MHz | 48 MHz | 48 MHz | 84 MHz | 72 MHz | 168 MHz | 72 MHz |
| Tensão | 3,3v | 3,3v | 3,3v | 3,3v | 3,3v | 3,3v | 3,3v |
| TCK/SWCLK | PC0 (52) | (95) |  |  |  |  |  |
| TMS/SWDIO | PC1 (51) | (94) |  |  |  |  |  |
| TDI | PC2 (50) | PJ.4 (92) |  |  |  |  |  |
| TDO/SWO | PC3 (49) | PJ.5 (93) |  |  |  |  |  |

### TM4C123GH6PM

No estado após o reset os pinos já estão programados para agir como JTAG/SWD, não sendo necessária nenhuma programação adicional, além disso eles estão protegidos para que não sejam alterados acidentalmente via software.

Digital Enable (DEN) PC[3:0] = 1111

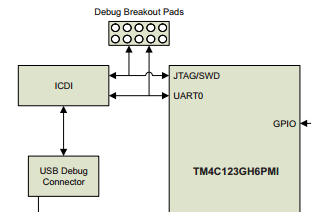
Pull-up Resistor On (PUR) PC[3:0] = 1111

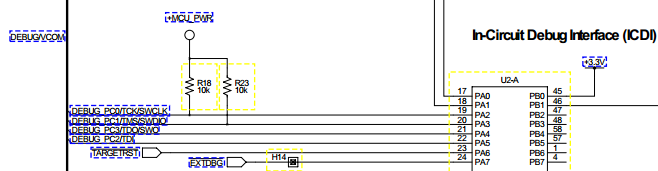
Pull-down Resistor disable (PDR) PC[3:0] = 0000

Enable Alternate function (AFIO) PC[3:0] = 1111

GPIOCTL PC[15:12] =0001, PC[11:8]= 0001, PC[7:4]=0001, PC[3:0]=0001

A placa de desenvolvimento Launchpad possui um ICDI para facilitar a programação e depuração da MCU, porém todos os pinos JTAG/SWD estão disponíveis via pontes de interconexão (breakout pads), que podem ser acesados para conectar um debugger externo. Porém nada prático, pois é necessário soldar um header na placa para poder conectar o debugger externo.





### MSP432P401R

A placa de desenvolvimento possui um ICDI/Debugger que permite a programação e depuração da MCU via USB. A interconexão entre os dois é feito através de Jumpers, o que permite acesso aos pinos JTAG/SWD da MCU por outro debugger.

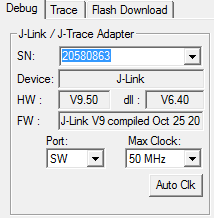


### STM32F103C8 (BLUEPILL)

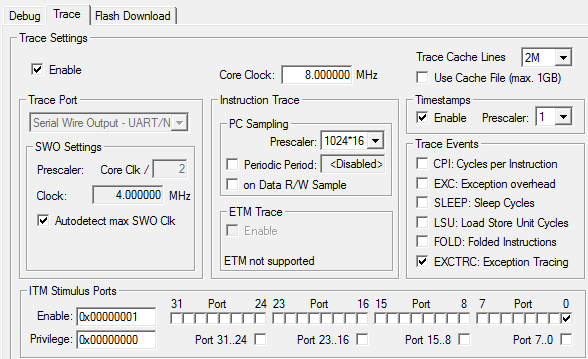
Conecte o pino PB3 (SWO) a entrada SWO do debugger (J-Link).

No Keil, na configuração do projeto, aba debug, pressione setting.

Certifique-se de selecionar a opção SW na caixa Port:



Selecione a aba Trace e configure de acordo:



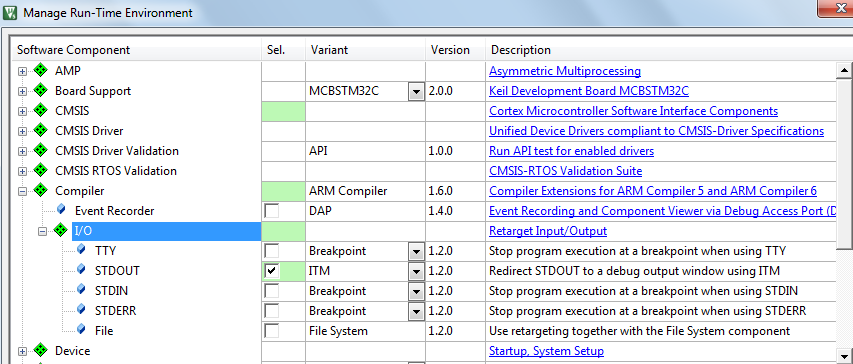
Habilite o trace (enable)

Marque o Autodetect max SWO Clk

Informe o Core Clock conforme a configuração do projeto.

Em ITM Stimulus Ports, certifique-se que o Port 0 esteja selecionada.

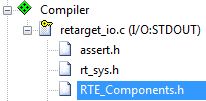
Em Manage Run-Time Environment:



Selecione Compiler, I/O, STDOUT.

Mude a opção para ITM, para redirecionar a saída para utilizar uma das portas stimulus – nesse caso a porta 0, que foi selecionado na configuração do Trace.

Essa configuração irá adicionar um módulo e um arquivo de configuração ( retarget\_io.c RTE\_Components.h) ao projeto:

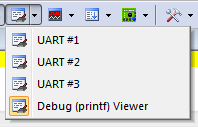


No programa (main.c) acrescente uma linha de impressão:

printf("Hello World!\n");

Compile o programa e entre no modo DEBUG.

Selecione para exibir a janela de printf viewer



Inicie o programa e observe a impressão da mensagem:

