

SWO 引腳的巧妙應用

在 Cortex-M3\M4 系列 MCU 中，內核的調試組件都有一個儀器化跟蹤單元 (ITM)。ITM 的一個主要的用途，就是支援調試資訊的輸出(例如，`printf` 格式輸出)。ITM 包含了 32 個刺激埠，允許不同的軟體把數據輸出到不同的埠，從而讓調試主機可以把它們的資訊分離開。每個埠都可以獨立的使能/除能，還可以允許或禁止用戶進程對它執行寫操作。那麼這些與 MCU 的 SWO 引腳有什麼關係呢？剛才我們有說 ITM 包含了 32 個埠，其中 SWO 引腳，就是 ITM 模塊的埠 0，我們可以直接用它來輸出一些調試資訊。那麼現在就看看如何使用它吧。

當 MCU 連接好 J-link 之後，我們在工程中的主函數起始處新增 `stdio.h` 檔案，如下所示。

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

接下來為了能直接調用 `printf` 函數格式化輸出，又需要用到我們之前提到的重新導向技術了，只是這次不是重新導向到序列埠，而是 ITM 模塊的 0 通道，即 SWO 引腳。如程序清單 1 所示。

程序清單 1 重新導向代碼

```
int fputc(int ch, FILE *f)
{
    ITM_SendChar(ch);
    return(ch);
}
```

其中 `ITM_SendChar` 函數是 `core_cm4.h` 內核檔案裡定義的內聯函數，我們可以直接調用。

完成重新導向之後，我們寫了個定時列印 HelloWorld 的程序。打開 Keil 工程的【Options for Target】->【debug】->【settings】->【Trace】，對 Trace 選項卡進行如圖 3 所示的組態。

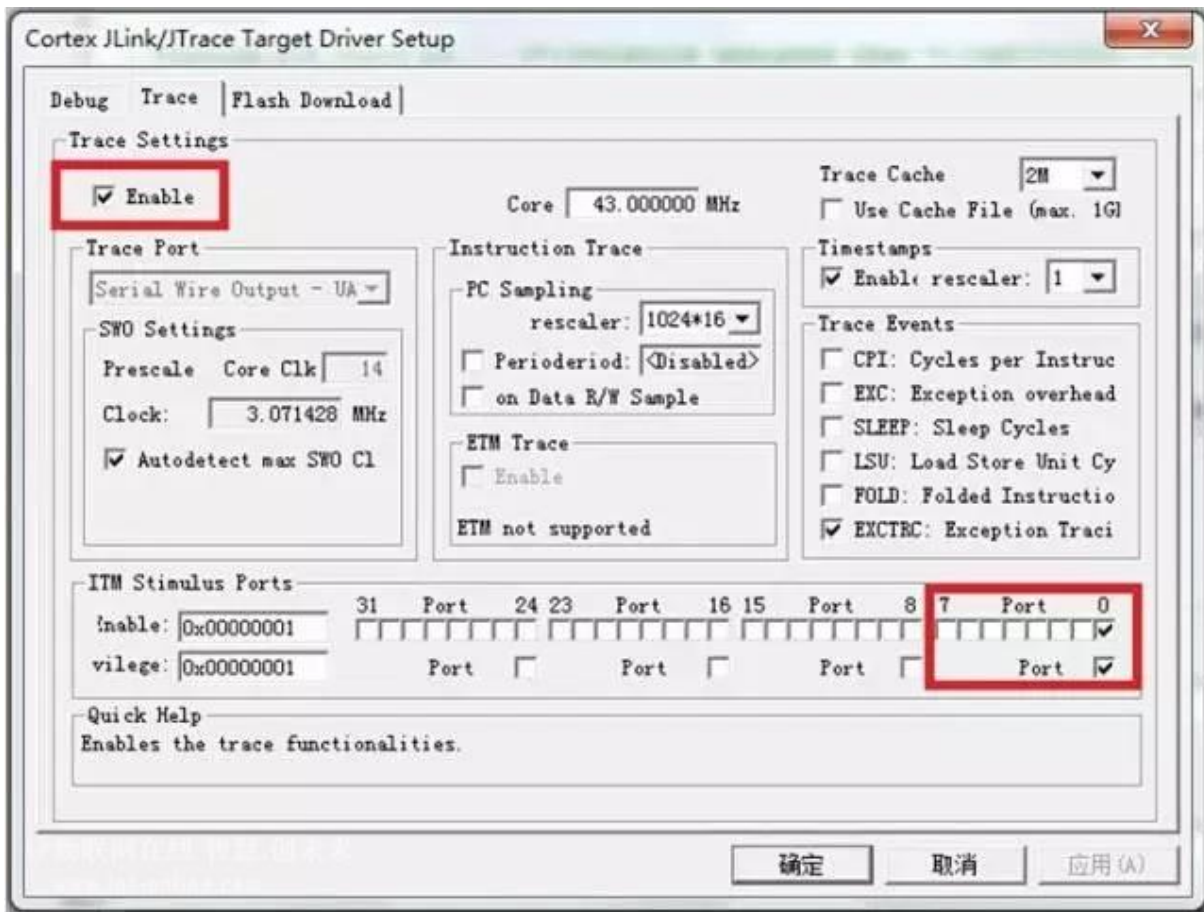


圖 3 Trace 選項卡組態

點擊確定之後，進入調試模式，打開【View】->【Serial Windows】->【Debug(printf) Viewer】，這時我們就可以看到 MCU 通過 J-link 向編譯器輸出的列印資訊了，如圖 4 所示。列印速度還很快哦。

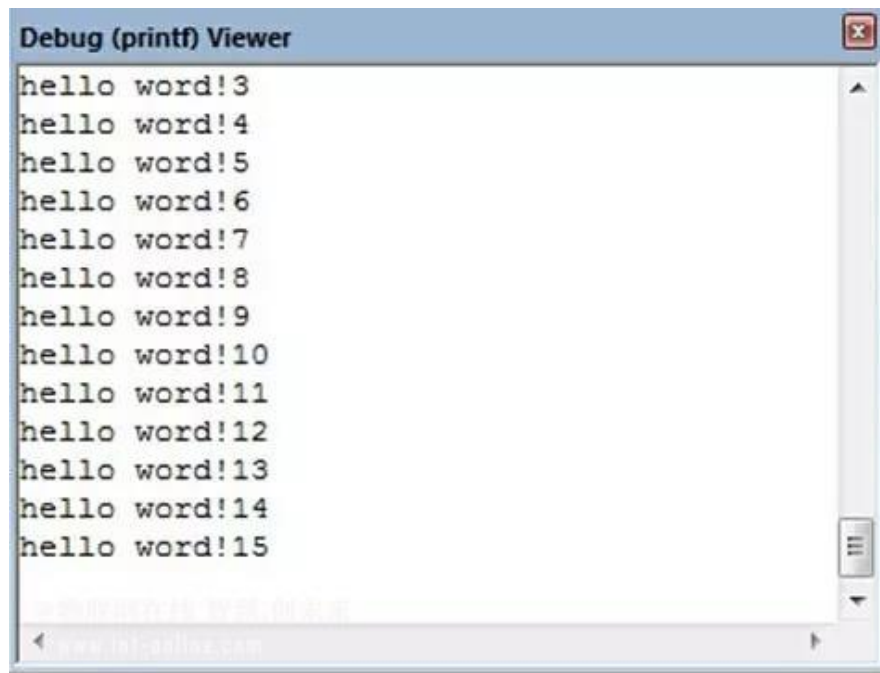


圖 4 Debug Viewer 窗口

到此我們算是學會了使用 **SWO** 引腳來調試輸出了，是不是很酷哦。細心的網友可能會發現為什麼 **Cortex-M0\M0+**內核的 **MCU** 沒有找到 **SWO** 引腳呢？這是因為它們採用的還是 **ARM-V6** 的老架構，內核的調試組件沒有 **ITM** 模塊。