透過SVC取得目前程式執行的PC值

NuMicro® 32 位系列微控制器範例代碼介紹

文件資訊

代碼簡述	本範例代碼透過 SVC 讀取，並打印 MCU 目前程式執行的 pc，stack， r0， r1， r12 等暫存器值。
BSP 版本	NUC230_240 Series BSP CMSIS V3.01.001
開發平台	NuTiny-EVB-NUC240 V1.2

The information described in this document is the exclusive intellectual property of Nuvoton Technology Corporation and shall not be reproduced without permission from Nuvoton.

Nuvoton is providing this document only for reference purposes of NuMicro microcontroller based system design. Nuvoton assumes no responsibility for errors or omissions.

All data and specifications are subject to change without notice.

For additional information or questions, please contact: Nuvoton Technology Corporation.

www.nuvoton.com

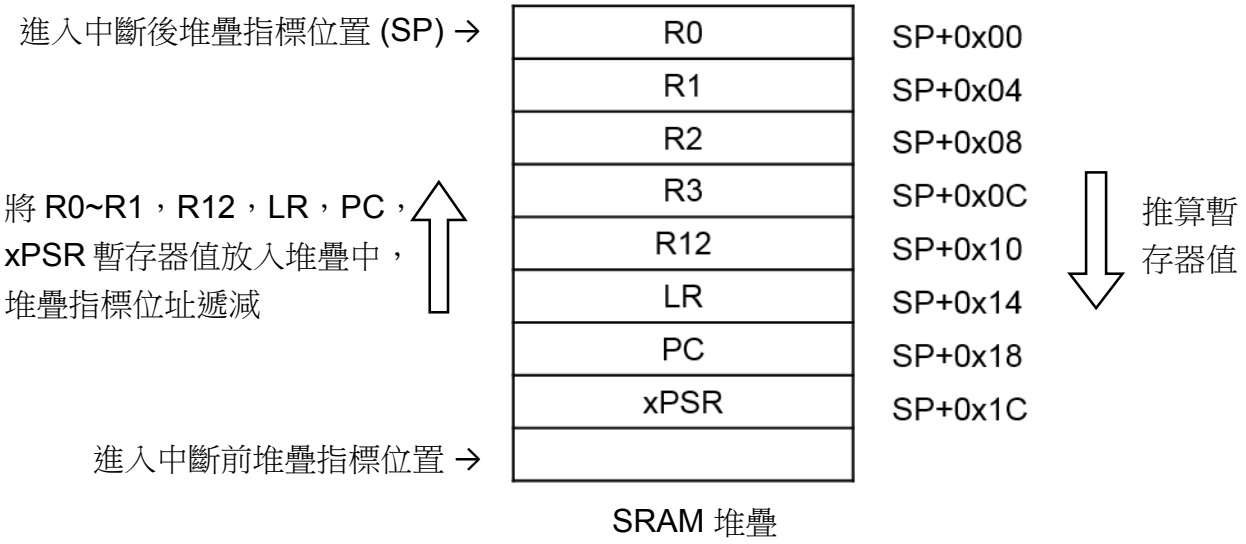
1 功能介紹

1.1 簡介

透過SVC讀取程式執行當下的PC值，執行SVC指令時，系統會產生SVC中斷，可用於OS系統服務程式的操作。產生SVC中斷時，系統會將目前暫存器的狀態放入堆疊中，本範例代碼利用這些資訊來獲取呼叫產生SVC中斷時的程式位址(PC)，並將該PC值打印出來，其他可獲取的資訊包括堆疊(stack)，R0~R3，R12，xPSR。

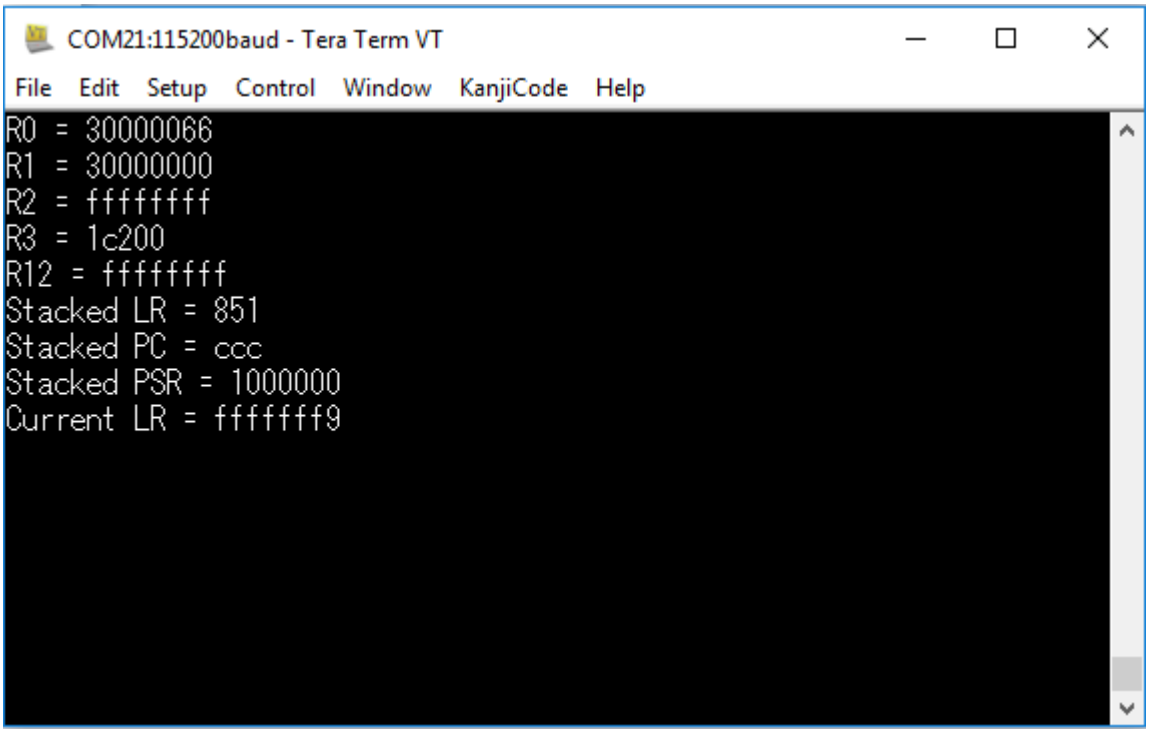
1.2 原理

中斷發生時，系統會自動將下列8個暫存器數值放入堆疊中：R0，R1，R2，R3，R12，LR，PC，xPSR。在SCV中斷裡，讀取目前堆疊指標的位置，即可推算得知這8個數值，包括執行SVC指令時的程式位址(PC)。



1.3 執行結果

使用NUC240開發板，PB.0(UART RX)及PB.1(UART TX)連接到連接至NuBridge，從UART0打印訊息。訊息顯示呼叫SVC當下的PC值和其他資訊。



2 代碼介紹

產生SVC中斷

執行SVC指令產生SVC中斷，利用Keil MDK提供的 __svc 關鍵字定義SVC函式，定義數字0的SVC函式，沒有傳入參數及傳回值。

```
    /* Define SVC function */
    void __svc(0x00) svc_PC(void);
```

使用時直接呼叫定義的SVC函式即可。

```
    /* Supervisor Call */
    svc_PC();
```

SVC中斷處理

讀取LR的第二個位元，檢查使用的堆疊是哪一個，PSP或MSP。呼叫打印函式，傳入參數：堆疊指標和返回地址。

```
__asm void SVC_Handler(void)
{
    MOVS    r0, #4
    MOV     r1, LR
    TST     r0, r1
    BEQ     stacking_used_MSP
    MRS     R0, PSP         ; first parameter - stacking was using PSP
    B       get_LR_and_branch
stacking_used_MSP
    MRS     R0, MSP         ; first parameter - stacking was using MSP
get_LR_and_branch
    MOV     R1, LR          ; second parameter is LR current value
    LDR     R2,=__cpp(dump_handler_c)
    BX      R2
    NOP
    BX      LR
}
```

打印PC值

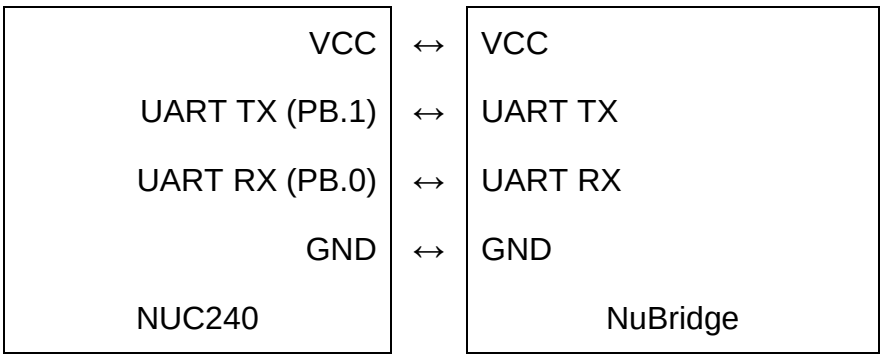
從堆疊裡讀取R0~R3，R12，LR，PC和xPSR數值，並打印出來。

```
void dump_handler_c(unsigned int * hardfault_args, unsigned lr_value)
{
    static volatile unsigned int stacked_r0;
    static volatile unsigned int stacked_r1;
    static volatile unsigned int stacked_r2;
    static volatile unsigned int stacked_r3;
    static volatile unsigned int stacked_r12;
    static volatile unsigned int stacked_lr;
    static volatile unsigned int stacked_pc;
    static volatile unsigned int stacked_psr;

    stacked_r0 = ((unsigned long) hardfault_args[0]);
    stacked_r1 = ((unsigned long) hardfault_args[1]);
    stacked_r2 = ((unsigned long) hardfault_args[2]);
    stacked_r3 = ((unsigned long) hardfault_args[3]);
    stacked_r12 = ((unsigned long) hardfault_args[4]);
    stacked_lr = ((unsigned long) hardfault_args[5]);
    stacked_pc = ((unsigned long) hardfault_args[6]);
    stacked_psr = ((unsigned long) hardfault_args[7]);

    printf ("R0 = %x\n", stacked_r0);
    printf ("R1 = %x\n", stacked_r1);
    printf ("R2 = %x\n", stacked_r2);
    printf ("R3 = %x\n", stacked_r3);
    printf ("R12 = %x\n", stacked_r12);
    printf ("Stacked LR = %x\n", stacked_lr);
    printf ("Stacked PC = %x\n", stacked_pc);
    printf ("Stacked PSR = %x\n", stacked_psr);
    printf ("Current LR = %x\n", lr_value);
}
```

3 軟體與硬體環境

- **軟體環境**
  - BSP 版本
    - NUC230_240 Series BSP CMSIS V3.01.001
  - IDE 版本
    - Keil uVersion 5.28
- **硬體環境**
  - 電路元件
    - NuTiny-EVB-NUC240 V1.2
    - NuBridge
  - 示意圖

NUC240 開發版連接 Nubridge，用來打印訊息。



4 目錄資訊

- EC_NUC240_READ_PC_BY_SVC_V1.00
  - Library — Sample code header and source files
    - CMSIS — Cortex® Microcontroller Software Interface Standard (CMSIS) by Arm® Corp.
    - Device — CMSIS compliant device header file
    - StdDriver — All peripheral driver header and source files
  - SampleCode
    - ExampleCode — Source file of example code

5 如何執行範例程式

1. 根據目錄資訊章節進入 ExampleCode 路徑中的 KEIL 資料夾，雙擊 NUC240_READ_PC_BY_SVC.uvproj。
2. 進入編譯模式介面
   a. 編譯
   b. 下載代碼至記憶體
3. 進入除錯模式介面
   a. 執行代碼

6 修訂紀錄

Date	Revision	Description
Jul. 31, 2019	1.00	1. 初始發布.

Important Notice

Nuvoton Products are neither intended nor warranted for usage in systems or equipment, any malfunction or failure of which may cause loss of human life, bodily injury or severe property damage. Such applications are deemed, "Insecure Usage".

Insecure usage includes, but is not limited to: equipment for surgical implementation, atomic energy control instruments, airplane or spaceship instruments, the control or operation of dynamic, brake or safety systems designed for vehicular use, traffic signal instruments, all types of safety devices, and other applications intended to support or sustain life.

All Insecure Usage shall be made at customer's risk, and in the event that third parties lay claims to Nuvoton as a result of customer's Insecure Usage, customer shall indemnify the damages and liabilities thus incurred by Nuvoton.

Please note that all data and specifications are subject to change without notice.
All the trademarks of products and companies mentioned in this datasheet belong to their respective owners.