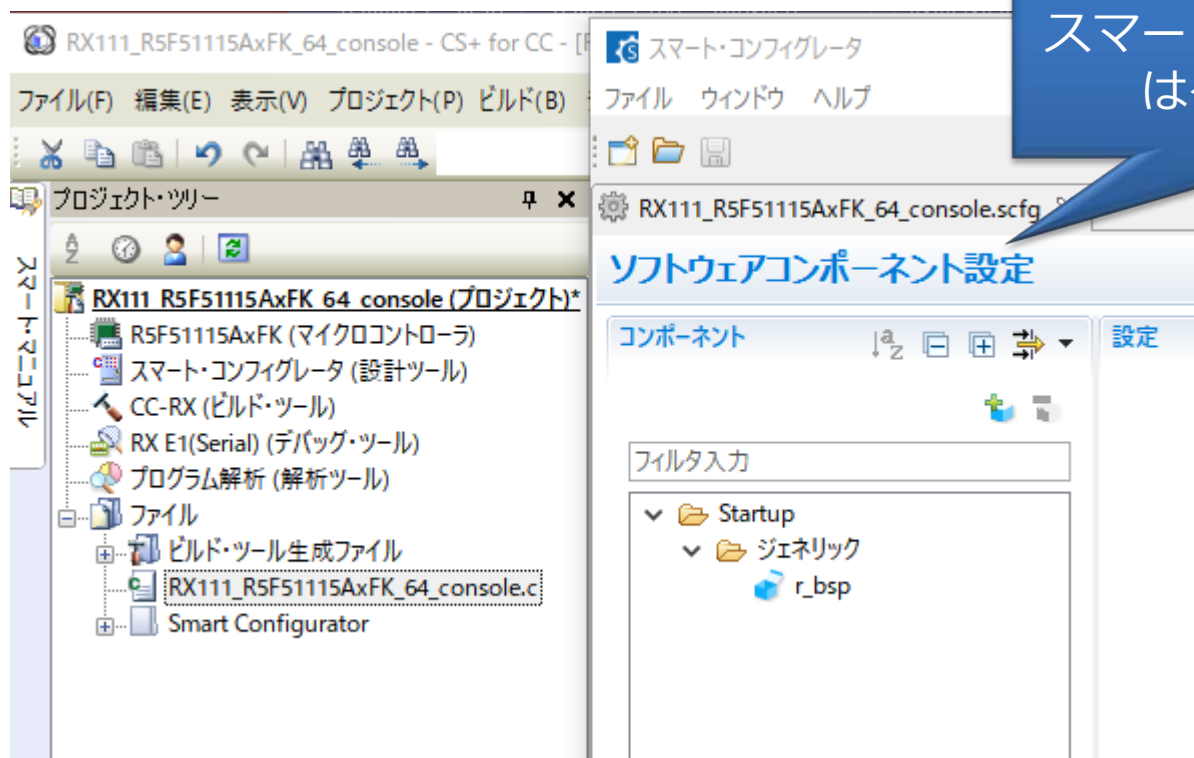
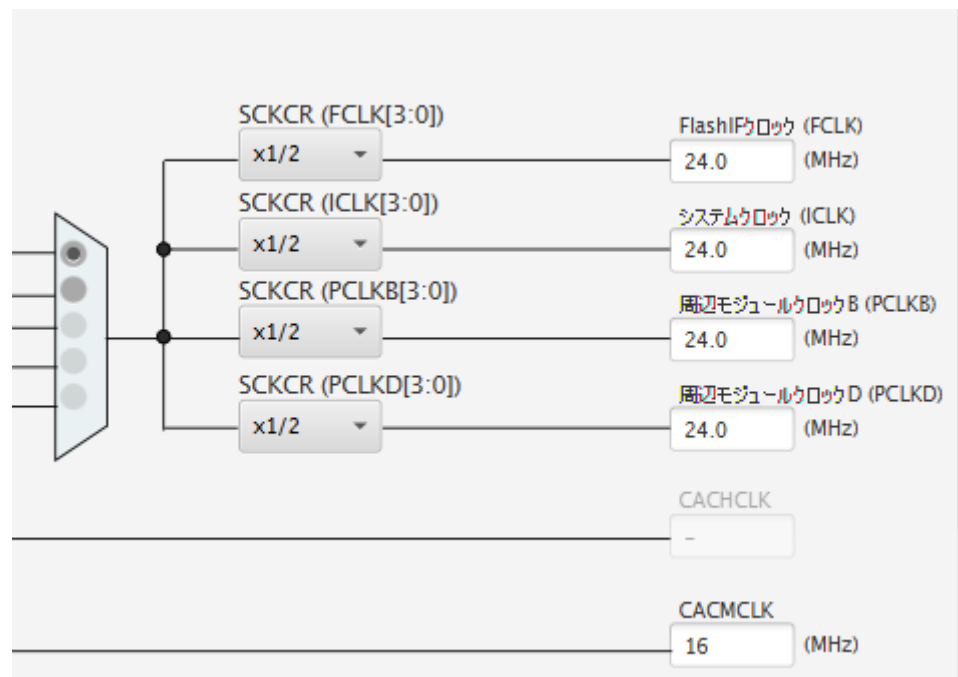


CS+、RX-SCでデバッグ コンソール出力する方法 エミュレータ編

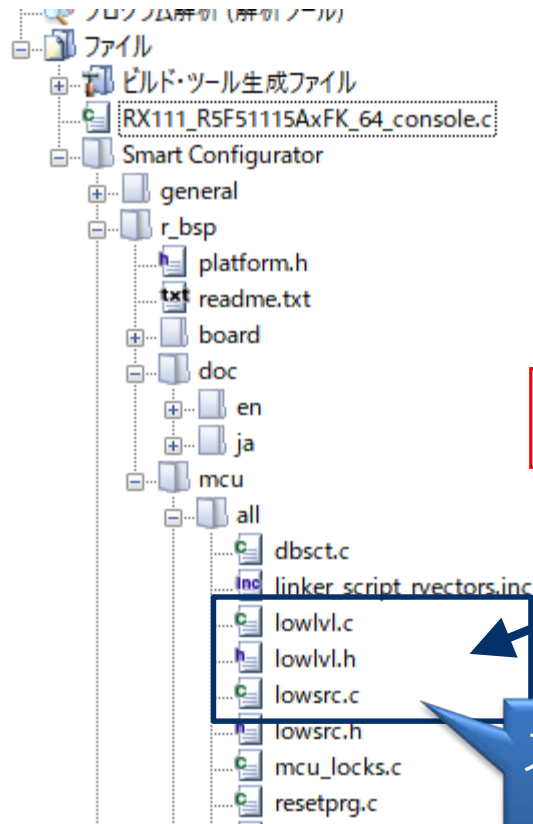
CS+ RXプロジェクト作成



スマート・コンフィグレータ
はクロックのみ設定



プロジェクトにコンソール出力用ファイルが3つある

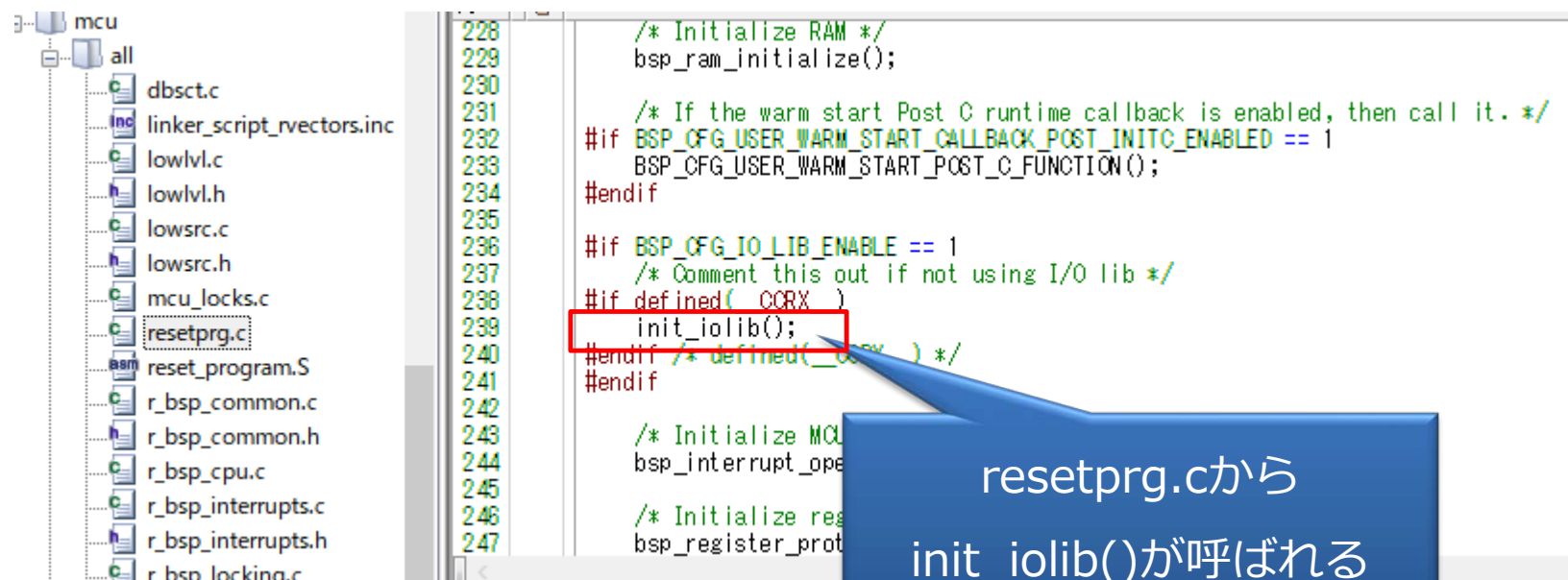


低水準インタフェースルーチンに関しては、下記を参照

<http://tool-support.renesas.com/autoupdate/support/onlinehelp/ja-JP/csp/V6.00.00/CS+.chm/Compiler-CCRX.chm/Output/ccrx08c0304y0000.html>

スマート・コンフィグレータを使用していない場合は、
手作業でコピーが必要

`_INIT_IOLIB()`が有効になっている



```
228 /* Initialize RAM */
229 bsp_ram_initialize();
230
231 /* If the warm start Post C runtime callback is enabled, then call it. */
232 #if BSP_CFG_USER_WARM_START_CALLBACK_POST_INITC_ENABLED == 1
233 BSP_CFG_USER_WARM_START_POST_C_FUNCTION();
234 #endif
235
236 #if BSP_CFG_IO_LIB_ENABLE == 1
237 /* Comment this out if not using I/O lib */
238 #if defined( CCRX )
239 init_iolib();
240 #endif /* defined( CCRX ) */
241 #endif
242
243 /* Initialize MCU
244 bsp_interrupt_op
245
246 /* Initialize reg
247 bsp_register_prot
```

デバッグコンソールへ出力する

mainのソース例

```
#include <string.h>
#include <stdio.h>
#include <stddef.h>
#include <machine.h>

char msg[128];

void main(void)
{
    int i;

    for (i=0; i<10; i++)
    {
        sprintf( msg, "count=%d ¥n", i );
        fprintf( stdout, msg );
    }
    while(1) { nop(); }
}
```

実行結果が表示される

The screenshot shows a development environment with a source code editor and a debug console. The source code is as follows:

```
/* NOTE:THIS IS A TYPICAL EXAMPLE.
/*
/******
#include "r_smc_entry.h"
#ifdef [Preprocessor block]
#endif

void main(void);
#ifdef [Preprocessor block]
#endif

char msg[128];

void main(void)
{
    int i;

    for (i=0; i<10; i++)
    {
        sprintf( msg, "count=%d ¥n", i );
        fprintf( stdout, msg );
    }
    while(1) { nop(); }
}
```

The debug console on the right shows the output:

```
count=0
count=1
count=2
count=3
count=4
count=5
count=6
count=7
count=8
count=9
```

A blue arrow points from the `for` loop in the code to the output in the console. A red arrow points to the 'デバッグ・コンソール(G)' menu item in the '表示(V)' menu.