

欢迎加入Cypress BLE技术讨论群130946269进行技术讨论，供应商和销售就不要进来了，发现了一律都T掉(不喜欢群里一堆广告，而不是技术问题)，更多的CY BLE教程可以到<http://www.alongble.com/> 中获取

一、前言

最近在使用PSOC BLE，为了调试方便把信息通过串口打印出来。所以想要用printf函数重定向至串口，由于PSOC Creator默认用的是GCC编译器，所以我们只需要重写__write()函数即可，因为GCC编译环境下printf函数最终调用的是__write()函数。但是为了兼容其他编译器，常规的写法如下所示：

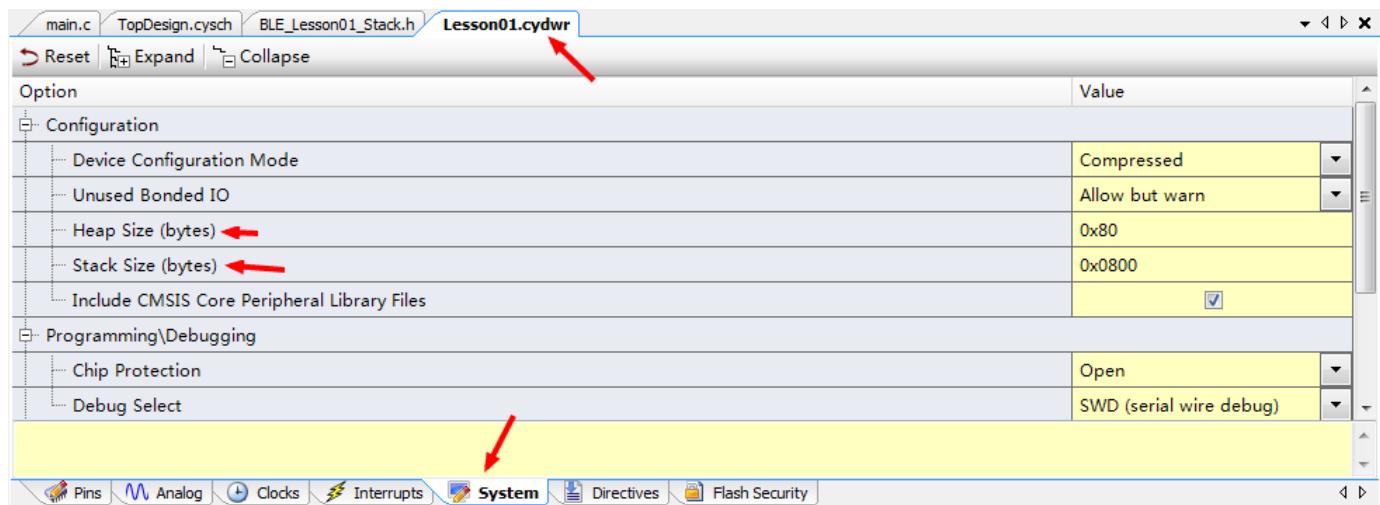
```
1. #if defined(__ARMCC_VERSION)
2. /* For MDK/RVDS compiler revise fputc function for printf functionality */
3. struct __FILE
4. {
5.     int handle;
6. };
7. enum
8. {
9.     STDIN_HANDLE,
10.    STDOUT_HANDLE,
11.    STDERR_HANDLE
12. };
13. FILE __stdin = {STDIN_HANDLE};
14. FILE __stdout = {STDOUT_HANDLE};
15. FILE __stderr = {STDERR_HANDLE};
16. int fputc(int ch, FILE *file)
17. {
18.     int ret = EOF;
19.     switch( file->handle )
20.     {
21.         case STDOUT_HANDLE:
22.             UART_DEB_UartPutChar(ch);
23.             ret = ch ;
24.             break ;
25.         case STDERR_HANDLE:
26.             ret = ch ;
27.             break ;
28.         default:
29.             file = file;
30.             break ;
31.     }
32.     return ret ;
33. }
34. #elif defined (__ICCARM__)      /* IAR */
35. /* For IAR compiler revise __write() function for printf functionality */
36. size_t __write(int handle, const unsigned char * buffer, size_t size)
37. {
38.     size_t nChars = 0;
39.     if (buffer == 0)
40.     {
41.         /*
42.          * This means that we should flush internal buffers. Since we
```

```

43.         * don't we just return. (Remember, "handle" == -1 means that all
44.         * handles should be flushed.)
45.         */
46.     return (0);
47. }
48. for /* Empty */; size != 0; --size)
49. {
50.     UART_DEB_UartPutChar(*buffer++);
51.     ++nChars;
52. }
53. return (nChars);
54. }
55. #else /* (__GNUC__)  GCC */
56. /* For GCC compiler revise _write() function for printf functionality */
57. int _write(int file, char *ptr, int len)
58. {
59.     int i;
60.     file = file;
61.     for (i = 0; i < len; i++)
62.     {
63.         UART_DEB_UartPutChar(*ptr++); //这里替代成UART的发送单字节的函数即可
64.     }
65.     return len;
66. }
67. #endif /* (__ARMCC_VERSION) */

```

重写了_write()函数，最后编译即可使用printf()函数通过串口打印出串口信息了。**切记，必须还要把Heap Size(Bytes)设大点的值，不然printf函数没法打印串口消息，但是必须要小于芯片的RAM大小。** 修改Heap Size(Bytes)如下图所示：



至此，现在就可以用Printf函数打印串口消息。