

关于 WinAVR-20060421 中的标准输入 / 输出

WinAVR20060421 对标准输入 / 输出对象连接设备的形势进行了改进, 导致本书中多个例子在 WinAVR20060421 中无法编译。下面先解释新版本中标准输入 / 输出的使用方法。

先看 avr-libc 使用手册提供的例子:

```
#include <stdio.h>
static int uart_putchar(char c, FILE *stream);
static FILE mystdout = FDEV_SETUP_STREAM(uart_putchar, NULL,
                                          _FDEV_SETUP_WRITE);

static int
uart_putchar(char c, FILE *stream)
{
    if (c == '\n')
        uart_putchar('\r', stream);
    loop_until_bit_is_set(UCSRA, UDRE);
    UDR = c;
    return 0;
}

int
main(void)
{
    init_uart();
    stdout = &mystdout;
    printf("Hello, world!\n");
    return 0;
}
```

在默认情况下, `fdevopen` 以及浮点数转换的 `printf` 和 `scanf` 都需要动态内存分配 (`malloc`), 这将连接动态内存分配模块, 导致代码膨胀, 如果系统中不使用浮点数转换, 通过使用上面的方法连接设备可以避免连接动态内存分配模块, 这对于资源有限的单片机系统是难得的。

替代 `fdevopen` 函数的是宏 `FDEV_SETUP_STREAM`, 它将初始化一个标准的 i/o 对象 (上例中为 `mystdout`), 由于这个对象是用户定义的变量, 固然无需动态分配内存。FILE 是标准 I/O 流使用的数据结构, `stdin`, `stdout`, `stderr` 均为指向 FILE 结构的指针。正如上面的 `main` 函数中那样标准流对象 `stdout` 指向用户定义的流对象数据结构后 `printf` 可以正常使用了。此时不能用 `fclose` 关闭, 如果需要关闭的话使用 `fdev_close`。

关于宏 `FDEV_SETUP_STREAM` 和相关参数说明请参考 `avr-libc` 用户手册。下面是书中相关示例的三种处理方法:

1. 使用 WinAVR20060421 之前版本编译。
2. 包含 `stdio.h` 文件之前定义宏 `__STDIO_FDEVOPEN_COMPAT_12`

如:

```
#define __STDIO_FDEVOPEN_COMPAT_12
#include <stdio.h>
```

.....

这一方法与前一种方法效果是一样的。意思是该宏的意思是兼容 `avr-libc1.2` 版本。

3. 修改代码使用新的方法连接设备。

第一步：在全局声明 `FILE` 变量并用宏 `FDEV_SETUP_STREAM` 初始化

```
static FILE mystdout = FDEV_SETUP_STREAM(uart_putchar, NULL,  
                                         _FDEV_SETUP_WRITE);
```

第二步：改变 `uart_putchar` 函数参数列表

```
int uart_putchar(char c, FILE *stream)
```

第三步：用 `stdout = &mystdout;` 替换 `fdevopen` 的调用。

第三种方法（也是建议在新程序中使用的）它生成的代码量会比前两种方法生成的会更小。

时间仓促就写到这里了，如有错误之处请到本书网站留言指正！谢谢。