

Q_Shell V1.2 使用说明

一. 综述

Q_Shell 是由酷享科技开发的一个串口调试组件，有了它你可以通过 PC 上的串口工具很方便的调试程序里的大多数函数（参数不超过四个）；查看和赋值程序中的部分变量（支持 u8,u16,u32 类型的全局变量）；读写指定地址处的寄存器值（32 位）。其特点如下：

1.使用方便

需要调用的函数或变量只需在本文件中注册，不需要手动添加任何声明。

2.功能齐全

函数参数支持十进制或十六进制数，字符串，指针。支持读写变量和寄存器；可以只用 TAB 键将未输入完整的函数补齐。

3.占用资源极少

仅占用 2K 字节的 ROM 空间，2 字节的全局变量空间，和 20 字节的栈空间。

4.拆卸容易

更改一个宏的值即可拆卸掉全部组件。

5.使用安全

支持使用前需要输入验证密码，可防止用户误操作。也可取消密码输入，只需要将 Q_Shell.h 中的宏 PASSWORD_DISABLE 置 1 禁止密码功能。

二. 移植

Q_Shell 组件可以运行在有操作系统的固件中，也可在无操作系统的裸机上直接进行移植。无操作系统移植的步骤如下：

1.将 Q_Shell.c 和 Q_Shell.h 两个文件加入到工程中。

2.在需要使用 Q_Shell 功能的 C 文件中包含 Q_Shell.h。

3. 将 Q_Shell.h 中的宏 USE_Q_SHELL_FUNCTION 置 1 开启 Q_Shell ，
USE_THREAD_MODE 置 0 禁止线程模式。

4.按以下要求写好串口驱动:

(1) 需要实现 `printf` 函数

(2) 在串口工具中输入命令字符串并按下回车后, 需要将从上位机接收到的字符串保存在一个字符串数组中, 然后将其传至 `Q_Sh_CmdHandler` 来执行命令。

有操作系统移植的步骤如下:

1.将 `Q_Shell.c` 和 `Q_Shell.h` 两个文件加入到工程中。

2.在需要使用 `Q_Shell` 功能的 C 文件中包含 `Q_Shell.h`。

3. 将 `Q_Shell.h` 中的宏 `USE_Q_SHELL_FUNCTION` 置 1 开启 `Q_Shell` ,
`USE_THREAD_MODE` 置 1 开启线程模式。

4.按以下要求写好串口驱动:

(1) 需要实现 `printf` 函数

(2) 在串口工具中输入命令字符串并按下回车后, 需要将从上位机接收到的字符串保存在一个字符串数组中, 然后将其传至 `Q_Sh_CmdHandler` 来执行命令。

5.将 `Q_Sh_GetInput` 函数放入循环执行的线程或任务中, 线程或任务的循环周期越快在终端上的反应越快, 推荐周期在 10Hz 左右。

三. 使用举例

1.调用注册的函数和变量

例如, 我们想通过 `Q_Shell` 调用在*.c 中定义的全局变量 `unsigned char var;`
以及函数 `unsigned char fun(char *str,unsigned int i,unsigned char j);`

我们可以通过以下两个步骤来完成。

第一步: 在 *.C 中注册需要调用的函数和变量

```
...
unsigned char Var;
QSH_VAR_REG (Var, "unsigned char Var", "u8");
...
unsigned char Fun(char *str, unsigned int i, char j)
{
...
}
QSH_FUN_REG (Fun, "unsigned char Fun(char *str,unsigned int i, char j)");
...
```

其中 QSH_VAR_REG 的第一个参数填变量名；第二个参数填描述该变量的字符串，可以书写任意的内容；第三个参数填变量类型描述字符串，根据所注册的变量类型选填“u8”“u16”“u32”中的一个。

其中 QSH_FUN_REG 的第一个参数填函数名；第二个参数填描述该函数的字符串，可以书写任意的内容。

第二步：在 PC 上的串口工具中输入以下命令并按回车

lv()	查看一共注册了哪些变量
get(Var)	查看变量 Var 的值
set(Var,16) 或 set(Var,0x10)	给变量 Var 赋值
lf()	查看一共注册了哪些函数
clear()	清除串口终端当前屏
reset()	软复位（需要自己实现，不同 MCU 可能不同）
fun("123",0x20001000,1)	调用已注册的函数 fun

2.读写指定地址处的寄存器

例如，我们想读写位于 0xE000E004 处寄存器

可在 PC 上的串口工具中输入以下命令

read(0xE000E004)	读出寄存器的值
write(0xE000E004,0xffffffff)	给寄存器赋值

四．版本记录

* Date	Author	Notes
* 2011-10-5	ChenYu	first version
* 2014-12-25	RTplay	增加 TAB 自动补齐功能，增加操作系统支持， 增加 clear，reset 函数
* 2015-8-3	RTplay	在中断模式下增加 “↑” “↓” 按键调用已使用命令功能， 命令保存 10 条