

# FREERTOS CLI DEMO ON I.MX RT

DAWEI



EXTERNAL USE



SECURE CONNECTIONS  
FOR A SMARTER WORLD

# 背景

在MCUX SDK中一般已经集成了一个Shell demo，但是有以下几点问题：

1. 直至SDK 2.8.2，该Shell demo还是BM工程，未集成Freertos示例。而且serial manager中设置了太多的宏，包括了太多功能，用户很难厘清该设置哪些宏，实现哪些功能才能实现Freertos Shell；
2. 基于RTOS的Shell在读取串口/UDP/TCP输入命令的时候，如果不实现Non-Blocking功能，将影响RTOS本身的调度功能；
3. MCUX SDK的shell不通用，Freertos实际上也提供CLI功能而且集成更简单通用；

# 如何集成

# 基础驱动

要实现基于阻塞的基础驱动，导入freertos\_lpuart例程即可实现基于LPUART Transfer接口的阻塞串口驱动。

另外，AN12679实现了RS485的基于DMA的串口发送接口控制。也可以移植到此，实现基于DMA的阻塞串口驱动。详见Private\_lpuart\_freertos.c/.h

套用默认的lpuart\_rtos\_handle\_t数据结构及互斥机制，同样可以实现串口阻塞功能。

# Freertos Plus

下载FreeRTOSv10.4.1，除了SDK中已经集成的Freertos kernel 文件外，还包含了plus的内容：  
包括了：CLI/UDP/TCP/Trace等功能  
并且提供了一些示例程序文件。

- FreeRTOS-Plus-CLI
- FreeRTOS-Plus-IO
- FreeRTOS-Plus-TCP
- FreeRTOS-Plus-Trace
- FreeRTOS-Plus-UDP
- Reliance-Edge
- WolfSSL
- WebDocs.url

- Common
- FreeRTOS\_Plus\_CLI\_with\_Trace\_Windows\_Simulator
- FreeRTOS\_Plus\_FAT\_SL\_and\_CLI\_Windows\_Simulator
- FreeRTOS\_Plus\_Reliance\_Edge\_and\_CLI\_Windows\_Simulator
- FreeRTOS\_Plus\_TCP\_Minimal\_Windows\_Simulator
- FreeRTOS\_Plus\_UDP\_and\_CLI\_LPC1830\_GCC
- FreeRTOS\_Plus\_UDP\_and\_CLI\_Windows\_Simulator
- FreeRTOS\_Plus\_WolfSSL\_Windows\_Simulator

# CLI功能集成

1. FreeRTOS\_CLI.c/.h文件：其中包含了CLI基本功能，包括help功能及命令检索、错误处理等功能。无需修改
2. CLI-commands.c：示例了一些基础功能的Shell命令，包括RTOS stats、RunTime, Echo Parameters命令等
3. UARTCommandConsole.c：串口接收处理任务。面向这个文件，需要实现接口函数xSerialPortInitMinimal、vSerialPutString、xSerialGetChar、xSerialPutChar。这些都利用基础驱动API实现。

串口接收处理任务都是依次读取单字节串口输入数据以\r\n结尾处理shell命令。

如果使用DMA+Idle的基础驱动，每次idle中断产生接收一个字符，需要上位机console在手动输入后立即发出字符，这样每个字符都会产生idle中断。

```
1: void vRegisterCLICommands( void )
2: {
3:     /* Register all the command line commands defined immediately above. */
4:     FreeRTOS_CLIRegisterCommand( &xTaskStats );
5:     FreeRTOS_CLIRegisterCommand( &xRunTimeStats );
6:     FreeRTOS_CLIRegisterCommand( &xThreeParameterEcho );
7:     FreeRTOS_CLIRegisterCommand( &xParameterEcho );
8:     //FreeRTOS_CLIRegisterCommand( &xStartTrace );
9: }
```

# CLI功能展示

有图console界面展示了如何输入命令获得shell互动的结果。

加载测试了下面几个shell命令，

help: 打印所有支持的命令及其说明

task-stats: 统计Freertos任务

run-time-stats: 统计Freertos任务占用时间量。

为了支持此功能，还需要使能一个定时器，作为系统时间的测量。这边使能GPT定时器作为时基

echo\_3\_parameters: 演示3参数shell 命令

echo\_parameters: 演示任意参数shell命令

```
FreeRTOS command server.
Type Help to view a list of registered commands.

>help
help:
Lists all the registered commands

task-stats:
Displays a table showing the state of each FreeRTOS task

run-time-stats:
Displays a table showing how much processing time each FreeRTOS task has used

echo_3_parameters <param1> <param2> <param3>:
Expects three parameters, echos each in turn

echo_parameters <...>:
Take variable number of parameters, echos each in turn

[Press ENTER to execute the previous command again]
>
task-stats
Task      State Priority Stack  #
*****
CLI              X      4    526   1
IDLE
[Press ENTER to execute the previous command again]
>run-time-stats
Task      Abs Time    % Time
*****
CLI              38      <1%
IDLE
[Press ENTER to execute the previous command again]
>echo_3_parameters 1 2 3
The three parameters were:
1: 1
2: 2
3: 3
[Press ENTER to execute the previous command again]
>
```

# Task Stat功能实现

FreeRTOS中实现了系统统计功能，这些功能能够集成在CLI菜单中，实现运行时间统计、task列表等。

使能这个功能需要打开下面的宏。

```
/* Run time and task stats gathering related definitions. */
```

```
#define configGENERATE_RUN_TIME_STATS 1//dawei:0
```

```
#define configUSE_TRACE_FACILITY 1
```

```
#define configUSE_STATS_FORMATTING_FUNCTIONS 1//dawei:0
```

//此配置为shell输出buffer的size，如果需要较多的输出内容，需要定义较大的size，但是不要超过g\_txBuffer的大小，

//g\_txBuffer为非cacheable的属性，最终将output buffer copy到g\_txBuffer再发送

```
#define configCOMMAND_INT_MAX_OUTPUT_SIZE 1024
```

并为下面的两个标准调用提供实现接口：这个接口使用GPT定时器实现。

```
//dawei add
```

```
#define portCONFIGURE_TIMER_FOR_RUN_TIME_STATS() AppConfigureTimerForRuntimeStats()
```

```
#define portGET_RUN_TIME_COUNTER_VALUE() AppGetRuntimeCounterValueFromISR()
```



# END



SECURE CONNECTIONS  
FOR A SMARTER WORLD