ECx00U&EGx00U 系列
QuecOpen SPI NAND Flash API 参考手册

LTE Standard 模块系列

版本：1.0

日期：2021-08-27

状态：受控文件

上海移远通信技术股份有限公司（以下简称“移远通信”）始终以为客户提供最及时、最全面的服务为宗旨。如需任何帮助，请随时联系我司上海总部，联系方式如下：

上海移远通信技术股份有限公司
上海市闵行区田林路 1016 号科技绿洲 3 期（B 区）5 号楼　　邮编：200233
电话：+86 21 5108 6236　　邮箱：info@quectel.com

或联系我司当地办事处，详情请登录：http://www.quectel.com/cn/support/sales.htm。

如需技术支持或反馈我司技术文档中的问题，请随时登陆网址：
http://www.quectel.com/cn/support/technical.htm 或发送邮件至：support@quectel.com。

前言

移远通信提供该文档内容以支持客户的产品设计。客户须按照文档中提供的规范、参数来设计产品。同时，您理解并同意，移远通信提供的参考设计仅作为示例。您同意在设计您目标产品时使用您独立的分析、评估和判断。在使用本文档所指导的任何硬软件或服务之前，请仔细阅读本声明。您在此承认并同意，尽管移远通信采取了商业范围内的合理努力来提供尽可能好的体验，但本文档和其所涉及服务是在“可用”基础上提供给您的。移远通信可在未事先通知的情况下，自行决定随时增加、修改或重述本文档。

使用和披露限制

许可协议

除非移远通信特别授权，否则我司所提供硬软件、材料和文档的接收方须对接收的内容保密，不得将其用于除本项目的实施与开展以外的任何其他目的。

版权声明

移远通信产品和本协议项下的第三方产品可能包含受移远通信或第三方材料、硬软件和文档版权保护的相关资料。除非事先得到书面同意，否则您不得获取、使用、向第三方披露我司所提供的文档和信息，或对此类受版权保护的资料进行复制、转载、抄袭、出版、展示、翻译、分发、合并、修改，或创造其衍生作品。移远通信或第三方对受版权保护的资料拥有专有权，不授予或转让任何专利、版权、商标或服务商标权的许可。为避免歧义，除了正常的非独家、免版税的产品使用许可，任何形式的购买都不可被视为授予许可。对于任何违反保密义务、未经授权使用或以其他非法形式恶意使用所述文档和信息的违法侵权行为，移远通信有权追究法律责任。

商标

除另行规定，本文档中的任何内容均不授予在广告、宣传或其他方面使用移远通信或第三方的任何商标、商号及名称，或其缩略语，或其仿冒品的权利。

第三方权利

您理解本文档可能涉及一个或多个属于第三方的硬软件和文档（“第三方材料”）。您对此类第三方材料的使用应受本文档的所有限制和义务约束。

移远通信针对第三方材料不做任何明示或暗示的保证或陈述，包括但不限于任何暗示或法定的适销性或特定用途的适用性、平静受益权、系统集成、信息准确性以及与许可技术或被许可人使用许可技术相关的不侵犯任何第三方知识产权的保证。本协议中的任何内容都不构成移远通信对任何移远通信产品或任何其他硬软件、设备、工具、信息或产品的开发、增强、修改、分销、营销、销售、提供销售或以其他方式维持生产的陈述或保证。此外，移远通信免除因交易过程、使用或贸易而产生的任何和所有保证。

免责声明

1) 移远通信不承担任何因未能遵守有关操作或设计规范而造成损害的责任。

2) 移远通信不承担因本文档中的任何因不准确、遗漏、或使用本文档中的信息而产生的任何责任。

3) 移远通信尽力确保开发中功能的完整性、准确性、及时性，但不排除上述功能错误或遗漏的可能。除非另有协议规定，否则移远通信对开发中功能的使用不做任何暗示或法定的保证。在适用法律允许的最大范围内，移远通信不对任何因使用开发中功能而遭受的损害承担责任，无论此类损害是否可以预见。

4) 移远通信对第三方网站及第三方资源的信息、内容、广告、商业报价、产品、服务和材料的可访问性、安全性、准确性、可用性、合法性和完整性不承担任何法律责任。

文档历史

修订记录

版本	日期	作者	变更表述
-	2021-03-02	Ryan YI	文档创建
1.0	2021-08-27	Ryan YI	受控版本

目录

文档历史 3
目录 4
表格索引 5
图片索引 6

1 引言 7
1.1. 适用模块 7

2 SPI NAND Flash API 介绍 8
2.1. 头文件 8
2.2. 函数概览 8
2.3. 函数详解 9
2.3.1. ql_spi_nand_init 9
2.3.1.1. ql_errcode_spi_nand_e 9
2.3.1.2. ql_spi_port_e 11
2.3.1.3. ql_spi_clk_e 11
2.3.1.4. ql_spi_input_sel_e 12
2.3.1.5. ql_spi_transfer_mode_e 13
2.3.1.6. ql_spi_cs_sel_e 13
2.3.2. ql_spi_nand_init_ext 14
2.3.2.1. ql_spi_nand_config_s 14
2.3.3. ql_spi_nand_read_devid 15
2.3.4. ql_spi_nand_read_page_spare 16
2.3.5. ql_spi_nand_write_page_spare 16
2.3.6. ql_spi_nand_read_status 17
2.3.6.1. ql_spi_nand_status_reg_e 17
2.3.7. ql_spi_nand_write_status 18
2.3.8. ql_spi_nand_erase_block 19
2.3.9. ql_spi_nand_reset 19

3 示例 20
3.1. 开发示例 20
3.2. 功能调试 20

4 附录 参考文档及术语缩写 22

表格索引

表 1：适用模块 7
表 2：函数概览 8
表 3：参考文档 22
表 4：术语缩写 22

图片索引

图 1：入口函数 ql_spi_nand_flash_demo_init() 20
图 2：SPI NAND Flash API 示例默认不启动 20
图 3：USB 端口显示..... 21
图 4：Log 信息 21

1 引言

移远通信 ECx00U 系列和 EGx00U 模块支持 QuecOpen®方案；QuecOpen®是基于 RTOS 的嵌入式开发平台，可简化 IoT 应用的软件设计和开发过程。有关 QuecOpen®的详细信息，请参考***文档 [1]***。

本文档主要介绍在 QuecOpen®方案下，如何在 ECx00U 系列和 EGx00U 模块上通过 SPI 总线操作 NAND Flash。

1.1. 适用模块

表 1：适用模块

模块系列	模块
ECx00U	EC200U 系列
	EC600U 系列
EGx00U	EG500U-CN
	EG700U-CN

2 SPI NAND Flash API 介绍

2.1. 头文件

SPI NAND Flash API 的头文件为 *ql_api_spi_nand_flash.h*，位于 SDK 包的 *components\ql-kernel\inc* 目录下。若无特别说明，本文档所述头文件均在该目录下。

2.2. 函数概览

表 2：函数概览

函数	说明
ql_spi_nand_init()	初始化配置 SPI（仅配置 SPI 总线和时钟频率）
ql_spi_nand_init_ext()	初始化配置 SPI
ql_spi_nand_read_devid()	读取 NAND Flash 设备 ID
ql_spi_nand_read_page_spare()	读取 NAND Flash 的页和备用区数据
ql_spi_nand_write_page_spare()	写入数据到 NAND Flash 的页和备用区
ql_spi_nand_read_status()	读取状态寄存器的值
ql_spi_nand_write_status()	写入状态寄存器的值
ql_spi_nand_erase_block()	擦除 NAND Flash 的块
ql_spi_nand_reset()	复位 NAND Flash

2.3. 函数详解

2.3.1. ql_spi_nand_init

该函数用于初始化配置 SPI（仅配置 SPI 总线和时钟频率）。调用该函数前，需通过 *ql_pin_set_func()* 设置相关 GPIO 引脚为 SPI 功能，详情请参考***文档 [2]***。

调用该函数将默认配置 *ql_spi_input_sel_e* 为 *QL_SPI_DI_1*，*ql_spi_transfer_mode_e* 为 *QL_SPI_DIRECT_POLLING*，*ql_spi_cs_sel_e* 为 *QL_SPI_CS0*。相关参数描述详情请参考***第 2.3.1.4 章***、***第 2.3.1.5 章***和***第 2.3.1.6 章***。

- **函数原型**

```
ql_errcode_spi_nand_e ql_spi_nand_init(ql_spi_port_e port, ql_spi_clk_e spiclk)
```

- **参数**

port：
[In] SPI 总线。详见***第 2.3.1.2 章***。

spiclk：
[In] SPI 时钟频率。

- **返回值**

详见***第 2.3.1.1 章***。

备注

若选择了错误的 SPI 总线进行初始化，将导致 NAND Flash 读、写、擦操作失败。因此，需对初始化返回值进行判定，若判定为初始化失败，应避免进行后续读、写、擦操作。

2.3.1.1. ql_errcode_spi_nand_e

SPI NAND Flash 结果码枚举定义如下：

```
typedef ql_errcode_spi_flash_e ql_errcode_spi_nand_e
```

```
typedef enum
{
    QL_SPI_FLASH_SUCCESS                    =    0,
    QL_SPI_FLASH_ERROR                      =    1 |
```

```
(QL_COMPONENT_STORAGE_EXTFLASH << 16),
    QL_SPI_FLASH_PARAM_TYPE_ERROR,
    QL_SPI_FLASH_PARAM_DATA_ERROR,
    QL_SPI_FLASH_PARAM_ACQUIRE_ERROR,
    QL_SPI_FLASH_PARAM_NULL_ERROR,
    QL_SPI_FLASH_DEV_NOT_ACQUIRE_ERROR,
    QL_SPI_FLASH_PARAM_LENGTH_ERROR,
    QL_SPI_FLASH_MALLOC_MEM_ERROR,
    QL_SPI_FLASH_NOT_FLASH_CONN_ERROR,
    QL_SPI_FLASH_NOT_SUPPORT_ERROR,
    QL_SPI_FLASH_OP_NOT_SUPPORT_ERROR,
    QL_SPI_FLASH_ECC_ERROR,
    QL_SPI_FLASH_PROGRAM_ERROR,
    QL_SPI_FLASH_ERASE_ERROR,
}ql_errcode_spi_flash_e;
```

- **参数**

参数	描述
QL_SPI_FLASH_SUCCESS	函数执行成功
QL_SPI_FLASH_ERROR	SPI 总线其他错误
QL_SPI_FLASH_PARAM_TYPE_ERROR	参数类型错误
QL_SPI_FLASH_PARAM_DATA_ERROR	参数数据错误
QL_SPI_FLASH_PARAM_ACQUIRE_ERROR	参数无法获取
QL_SPI_FLASH_PARAM_NULL_ERROR	参数 NULL 错误
QL_SPI_FLASH_DEV_NOT_ACQUIRE_ERROR	无法获取 SPI 总线
QL_SPI_FLASH_PARAM_LENGTH_ERROR	参数长度错误
QL_SPI_FLASH_MALLOC_MEM_ERROR	申请内存错误
QL_SPI_FLASH_NOT_FLASH_CONN_ERROR	未接入 Flash 芯片
QL_SPI_FLASH_NOT_SUPPORT_ERROR	Flash 型号不支持（仅适用于外接 NOR Flash）
QL_SPI_FLASH_OP_NOT_SUPPORT_ERROR	Flash 型号不支持此操作
QL_SPI_FLASH_ECC_ERROR	NAND Flash ECC 校验错误
QL_SPI_FLASH_PROGRAM_ERROR	Flash 程序错误
QL_SPI_FLASH_ERASE_ERROR	Flash 擦除错误

2.3.1.2. ql_spi_port_e

SPI 总线枚举定义如下：

```
typedef enum
{
    QL_SPI_PORT1,
    QL_SPI_PORT2,
}ql_spi_port_e;
```

- **参数**

参数	描述
QL_SPI_PORT1	SPI1 总线
QL_SPI_PORT2	SPI2 总线

2.3.1.3. ql_spi_clk_e

SPI 时钟频率枚举定义如下：

```
typedef enum
{
    QL_SPI_CLK_INVALID=-1,
    QL_SPI_CLK_781_25KHZ = 781250,
    QL_SPI_CLK_1_5625MHZ = 1562500,
    QL_SPI_CLK_3_125MHZ = 3125000,
    QL_SPI_CLK_5MHZ = 5000000,
    QL_SPI_CLK_6_25MHZ = 6250000,
    QL_SPI_CLK_10MHZ = 10000000,
    QL_SPI_CLK_12_5MHZ = 12500000,
    QL_SPI_CLK_20MHZ = 20000000,
    QL_SPI_CLK_25MHZ = 25000000,
    QL_SPI_CLK_33_33MHZ = 33000000,
    QL_SPI_CLK_50MHZ_MAX = 50000000,
}ql_spi_clk_e;
```

- **参数**

参数	描述
QL_SPI_CLK_INVALID	无效参数

QL_SPI_CLK_781_25KHZ	时钟频率为 781.25 kHz
QL_SPI_CLK_1_5625MHZ	时钟频率为 1.5625 MHz
QL_SPI_CLK_3_125MHZ	时钟频率为 3.125 MHz
QL_SPI_CLK_5MHZ	时钟频率为 5 MHz
QL_SPI_CLK_6_25MHZ	时钟频率为 6.25 MHz
QL_SPI_CLK_10MHZ	时钟频率为 10 MHz
QL_SPI_CLK_12_5MHZ	时钟频率为 12.5 MHz
QL_SPI_CLK_20MHZ	时钟频率为 20 MHz
QL_SPI_CLK_25MHZ	时钟频率为 25 MHz
QL_SPI_CLK_33_33MHZ	时钟频率为 33.33 MHz
QL_SPI_CLK_50MHZ_MAX	时钟频率为 50 MHz（最大时钟频率）

2.3.1.4. ql_spi_input_sel_e

数据输入引脚枚举定义如下：

```
typedef enum
{
    QL_SPI_DI_0 = 0,
    QL_SPI_DI_1,
    QL_SPI_DI_2,
}ql_spi_input_sel_e;
```

- **参数**

参数	描述
QL_SPI_DI_0	DI0 为数据输入引脚（暂不支持）
QL_SPI_DI_1	DI1 为数据输入引脚
QL_SPI_DI_2	DI2 为数据输入引脚（暂不支持）

2.3.1.5. ql_spi_transfer_mode_e

SPI 传输模式枚举定义如下：

```
typedef enum
{
    QL_SPI_DIRECT_POLLING = 0,
    QL_SPI_DIRECT_IRQ,
    QL_SPI_DMA_POLLING,
    QL_SPI_DMA_IRQ,
}ql_spi_transfer_mode_e;
```

- **参数**

参数	描述
QL_SPI_DIRECT_POLLING	FIFO 读写，轮询等待。
QL_SPI_DIRECT_IRQ	FIFO 读写，中断通知。暂不支持。
QL_SPI_DMA_POLLING	DMA 读写，轮询等待。此模式下，SPI 不可与 SD 卡同时使用。若已启用 SD 卡，则 SPI 将申请 DMA 通道失败、初始化失败。
QL_SPI_DMA_IRQ	DMA 读写，中断通知。暂不支持。

备注

使用 DMA 轮询模式时，SPI 与 SD 卡抢占 DMA 通道；若 SD 卡已使用 DMA 通道，SPI 初始化则会失败。

2.3.1.6. ql_spi_cs_sel_e

CS 引脚枚举定义如下：

```
typedef enum
{
    QL_SPI_CS0 = 0,
    QL_SPI_CS1,
    QL_SPI_CS2,
    QL_SPI_CS3,
}ql_spi_cs_sel_e;
```

- **参数**

参数	描述
QL_SPI_CS0	CS0 为 SPI 的 CS 引脚
QL_SPI_CS1	CS1 为 SPI 的 CS 引脚
QL_SPI_CS2	CS2 为 SPI 的 CS 引脚（暂不支持）
QL_SPI_CS3	CS3 为 SPI 的 CS 引脚（暂不支持）

2.3.2. ql_spi_nand_init_ext

该函数用于初始化配置 SPI，包括 SPI 总线、时钟频率、传输模式、数据输入引脚和 CS 引脚。调用此函数前，需调用 *ql_pin_set_func()*函数设置相关 GPIO 引脚为 SPI 功能，详情请参考***文档 [2]***。

- **函数原型**

```
ql_errcode_spi_nand_e ql_spi_nand_init_ext(ql_spi_nand_config_s nand_config)
```

- **参数**

nand_config：
[In] SPI 配置参数。详见***第 2.3.2.1 章***。

- **返回值**

详见***第 2.3.1.1 章***。

备注

若选择了错误的 SPI 总线进行初始化，将导致 NAND Flash 读、写、擦操作失败。因此，需对初始化返回值进行判定，若判定为初始化失败，应避免进行后续读、写、擦操作。

2.3.2.1. ql_spi_nand_config_s

SPI 配置参数结构体定义如下：

```
typedef ql_spi_flash_config_s ql_spi_nand_config_s
```

```
typedef struct
{
```

```
    ql_spi_port_e port;
    ql_spi_clk_e spiclk;
    ql_spi_input_sel_e input_sel;
    ql_spi_transfer_mode_e transmode;
    ql_spi_cs_sel_e cs;
} ql_spi_flash_config_s;
```

- **参数**

类型	参数	描述
ql_spi_port_e	*port*	SPI 总线。详见***第 2.3.1.2 章***。
ql_spi_clk_e	*spiclk*	SPI 时钟频率。详见***第 2.3.1.3 章***。
ql_spi_input_sel_e	*input_sel*	数据输入引脚。详见***第 2.3.1.4 章***。
ql_spi_transfer_mode_e	*transmode*	SPI 传输模式。详见***第 2.3.1.5 章***。
ql_spi_cs_sel_e	*cs*	CS 引脚。详见***第 2.3.1.6 章***。

2.3.3. ql_spi_nand_read_devid

该函数用读取 NAND Flash 设备 ID。

- **函数原型**

```
ql_errcode_spi_nand_e ql_spi_nand_read_devid(ql_spi_port_e port, unsigned char *mid)
```

- **参数**

port：
[In] SPI 总线。详见***第 2.3.1.2 章***。

mid：
[Out] NAND Flash 设备 ID。

- **返回值**

详见***第 2.3.1.1 章***。

2.3.4. ql_spi_nand_read_page_spare

该函数用于读取 NAND Flash 的页（2048 个字节）和备用区（64 个字节）数据。

- **函数原型**

```
ql_errcode_spi_nand_e ql_spi_nand_read_page_spare(ql_spi_port_e port, unsigned int page_addr, unsigned short column_addr, unsigned char *data, int len)
```

- **参数**

port：
[In] SPI 总线。详见***第 2.3.1.2 章***。

page_addr：
[In] 页地址。

column_addr：
[In] 列地址。

data：
[Out] 读取出的缓存数据。

len：
[In] 读取数据长度。单位：字节。

- **返回值**

详见***第 2.3.1.1 章***。

2.3.5. ql_spi_nand_write_page_spare

该函数用于写入数据到 NAND Flash 的页（2048 个字节）和备用区（64 个字节）。

- **函数原型**

```
ql_errcode_spi_nand_e ql_spi_nand_write_page_spare(ql_spi_port_e port, unsigned int page_addr, unsigned short column_addr, unsigned char *data, int len)
```

- **参数**

port：
[In] SPI 总线。详见***第 2.3.1.2 章***。

page_addr：
[In] 页地址。

column_addr：
[In] 列地址。

data：
[In] 需写入的数据。

len：
[In] 写入数据长度。单位：字节。

- **返回值**

详见***第 2.3.1.1 章***。

2.3.6. ql_spi_nand_read_status

该函数用于读取状态寄存器的值。

- **函数原型**

```
ql_errcode_spi_nand_e ql_spi_nand_read_status(ql_spi_port_e port, ql_spi_nand_status_reg_e reg, unsigned char *status)
```

- **参数**

port：
[In] SPI 总线。详见***第 2.3.1.2 章***。

reg：
[In] 状态寄存器。详见***第 2.3.6.1 章***。

status：
[Out] 读取的状态寄存器的值。

- **返回值**

详见***第 2.3.1.1 章***。

2.3.6.1. ql_spi_nand_status_reg_e

状态寄存器枚举定义如下：

```
typedef enum
{
    QL_NAND_FLASH_STATUS_1 = 0,
```

```
    QL_NAND_FLASH_STATUS_2,
    QL_NAND_FLASH_STATUS_3,
    QL_NAND_FLASH_STATUS_4,
}ql_spi_nand_status_reg_e;
```

- **参数**

参数	描述
QL_NAND_FLASH_STATUS_1	状态寄存器 1
QL_NAND_FLASH_STATUS_2	状态寄存器 2
QL_NAND_FLASH_STATUS_3	状态寄存器 3
QL_NAND_FLASH_STATUS_4	状态寄存器 4

2.3.7. ql_spi_nand_write_status

该函数用于写入状态寄存器的值。

- **函数原型**

```
ql_errcode_spi_nand_e ql_spi_nand_write_status(ql_spi_port_e port, ql_spi_nand_status_reg_e reg, unsigned char status)
```

- **参数**

port：
[In] SPI 总线。详见***第 2.3.1.2 章***。

reg：
[In] 状态寄存器。详见***第 2.3.6.1 章***。

status：
[In] 写入的状态寄存器的值。

- **返回值**

详见***第 2.3.1.1 章***。

2.3.8. ql_spi_nand_erase_block

该函数用于擦除 NAND Flash 的块。

- **函数原型**

```
ql_errcode_spi_nand_e ql_spi_nand_erase_block(ql_spi_port_e port, unsigned int page_addr)
```

- **参数**

port：
[In] SPI 总线。详见***第 2.3.1.2 章***。

page_addr：
[In] 块的第一个页页地址。

- **返回值**

详见***第 2.3.1.1 章***。

2.3.9. ql_spi_nand_reset

该函数用于复位 NAND Flash。

- **函数原型**

```
ql_errcode_spi_nand_e ql_spi_nand_reset(ql_spi_port_e port)
```

- **参数**

port：
[In] SPI 总线。详见***第 2.3.1.2 章***。

- **返回值**

详见***第 2.3.1.1 章***。

3 示例

本章节主要介绍在 APP 侧如何使用上述 API 进行 SPI NAND Flash 开发以及简单调试。

3.1. 开发示例

移远通信 ECx00U 系列和 EGx00U QuecOpen SDK 代码中提供了 SPI NAND Flash API 的示例文件 *\components\ql-application\spi_nand_flash\spi_nand_flash_demo.c*。该示例中主要包含 SPI NAND Flash 初始化、状态寄存器值读和写、页和备份区读和写、擦除块、设备 ID 读取及复位等操作。示例入口函数为 *ql_spi_nand_flash_demo_init()*，如下图所示。

```
QlOSStatus ql_spi_nand_flash_demo_init(void)
{
    QlOSStatus err = QL_OSI_SUCCESS;

    err = ql_rtos_task_create(&spi_nand_flash_demo_task, SPI_NAND_FLASH_DEMO_TASK_STACK_SIZE, SPI_NAND_FLASH_DEMO_TASK_PRIO, "ql_spi_nand", ql_spi_nand_flash_demo_task_pthread, NULL, SPI_NAND_FLASH_DEMO_TASK_EVENT_CNT);
    if(err != QL_OSI_SUCCESS)
    {
        QL_SPI_NAND_LOG("demo_task created failed");
        return err;
    }

    return err;
}
```

图 1：入口函数 ql_spi_nand_flash_demo_init()

如下图所示，上述示例已在 *ql_init_demo_thread* 线程中默认不启动。如需设置示例自启动，取消注释即可。

```
#ifdef QL_APP_FEATURE_SPI_NAND_FLASH
    //ql_spi_nand_flash_demo_init();
#endif
```

图 2：SPI NAND Flash API 示例默认不启动

3.2. 功能调试

调试 SPI NAND Flash 功能前，需在 LTE OPEN EVB 上外接 NAND Flash 芯片。

编译并烧录固件后，使用 USB 线连接 LTE OPEN EVB 的 USB 端口和 PC。模块开机后，PC 中“设备管理器”中显示端口如下图所示。使用 cooltools 工具抓取 log 后，可通过 USB AP Log 端口查看该示例

的调试信息。Log 抓取方法请参考*文档 [3]*。

图 3：USB 端口显示

模块开机后自动启动 *ql_spi_nand_flash_demo_init()*。通过比较 NAND Flash 页和保护区读写后的数据，判断读写是否操作成功；擦除操作后进行读取操作，若读取出的数据全为 FF，则可判断擦除操作成功。

如函数执行失败，可根据相应 log 信息查看失败原因。

Index	Received	Tick	Level	Description
71	09:58:44.532	41192	QOPN/I	[ql_SPI][ql_spi_write, 494] write sucess
72	09:58:44.532	41194	QOPN/I	[ql_SPI_NAND][ql_spi_nand_wait_write_finish, 244] status=2
73	09:58:44.532	41194	QOPN/I	[ql_SPI][ql_spi_write, 494] write sucess
74	09:58:44.532	41194	QOPN/I	[ql_SPI_NAND][ql_spi_nand_wait_write_finish, 244] status=0
99	09:58:44.563	41391	QOPN/I	[ql_SPI][ql_spi_write, 494] write sucess
100	09:58:44.563	41391	QOPN/I	[ql_SPI][ql_spi_write, 494] write sucess
103	09:58:44.579	41683	QOPN/I	[ql_SPI_NAND][ql_spi_nand_read_devid, 534] mid =efbf22
104	09:58:44.579	41684	QOPN/I	[ql_SPI_NAND_DEMO][ql_spi_nand_flash_demo_task_pthread, 186] sr1 = 0
105	09:58:44.579	41684	QOPN/I	[ql_SPI_NAND_DEMO][ql_spi_nand_flash_demo_task_pthread, 195] sr2 = 18
106	09:58:44.579	41684	QOPN/I	[ql_SPI_NAND_DEMO][ql_spi_nand_flash_demo_task_pthread, 204] sr3 = 0
107	09:58:44.579	41685	QOPN/I	[ql_SPI_NAND_DEMO][ql_spi_nand_flash_demo_task_pthread, 213] sr4 = 0
108	09:58:44.579	41685	QOPN/I	[ql_SPI_NAND_DEMO][ql_spi_nand_flash_demo_task_pthread, 222] a2 = 0xaaaaaaaa

图 4：Log 信息

4 附录 参考文档及术语缩写

表 3：参考文档

文档名称
[1] Quectel_ECx00U&EGx00U 系列_QuecOpen_CSDK_快速开发指导
[2] Quectel_ECx00U&EGx00U 系列_QuecOpen_GPIO_API_参考手册
[3] Quectel_ECx00U&EGx00U 系列_QuecOpen_Log_抓取操作指导

表 4：术语缩写

缩写	英文全称	中文全称
API	Application Programming Interface	应用程序接口
CS	Chip Select	片选
DI	Data Input	数据输入
DMA	Direct Memory Access	直接存储器访问
ECC	Error Correcting Code	错误检查和纠正
FIFO	First In First Out	先进先出
PC	Personal Computer	个人计算机
RTOS	Real-Time Operating System	实时操作系统
SD	Secure Digital Card	安全数字卡
SPI	Serial Peripheral Interface	串行外设接口
USB	Universal Serial Bus	通用串行总线