

# **ECx00U&EGx00U 系列 QuecOpen**

## **6 线 SPI NOR Flash 文件系统挂载**

## **API 参考手册**

**LTE Standard 模块系列**

版本：1.0

日期：2021-11-08

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](mailto: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](mailto: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。

版权所有 ©上海移远通信技术股份有限公司 2021，保留一切权利。

**Copyright © Quectel Wireless Solutions Co., Ltd. 2021.**

## 文档历史

### 修订记录

版本	日期	作者	变更表述
-	2021-09-07	Sum LI	文档创建
1.0	2021-11-08	Sum LI	受控版本

## 目录

文档历史 .....	3
目录 .....	4
表格索引 .....	5
图片索引 .....	6
<b>1 引言 .....</b>	<b>7</b>
1.1. 适用模块 .....	7
<b>2 hal_spi_flash_prop.csv .....</b>	<b>8</b>
2.1. 支持的 Flash 型号信息 .....	8
2.2. 相关参数含义 .....	9
2.2.1. halSpiFlash_t .....	9
2.2.1.1. mid .....	11
2.2.1.2. security register .....	11
2.2.1.3. halSpiFlashType_t .....	12
2.2.1.4. halSpiFlashUidType_t .....	13
2.2.1.5. halSpiFlashCpidType_t .....	14
2.2.1.6. halSpiFlashWpType_t .....	14
2.2.1.7. 其他特性参数 .....	15
<b>3 6 线 SPI NOR Flash 文件系统挂载相关 API .....</b>	<b>17</b>
3.1. 头文件 .....	17
3.2. 函数概览 .....	17
3.3. 函数详解 .....	17
3.3.1. ql_spi6_ext_nor_flash_init .....	17
3.3.2. ql_spi6_ext_nor_flash_deinit .....	18
3.3.3. ql_spi6_ext_nor_flash_sffs_mkfs .....	18
3.3.3.1. ql_errcode_spi6_nor_e .....	19
3.3.4. ql_spi6_ext_nor_flash_sffs_mount .....	20
3.3.5. ql_spi6_ext_nor_flash_sffs_unmount .....	20
<b>4 示例及调试 .....</b>	<b>21</b>
4.1. App 的 6 线 SPI NOR Flash 文件系统挂载开发示例 .....	21
4.1.1. 6 线 SPI NOR Flash 文件系统挂载应用程序示例说明 .....	21
4.1.2. 应用程序示例自启动 .....	23
4.2. 6 线 SPI NOR Flash 文件系统挂载相关调试 .....	23
4.2.1. 调试准备 .....	23
4.2.2. 6 线 SPI NOR Flash 文件系统挂载功能调试 .....	24
<b>5 附录 参考文档及术语缩写 .....</b>	<b>26</b>

## 表格索引

表 1: 适用模块 .....	7
表 2: 函数概览 .....	17
表 3: 参考文档 .....	26
表 4: 术语缩写 .....	26

## 图片索引

图 1: 支持的 Flash 型号的相关参数信息 (1) .....	8
图 2: 支持的 Flash 型号的相关参数信息 (2) .....	9
图 3: 型号 GD25LQ128D mid .....	11
图 4: 型号 W25Q128JW mid .....	11
图 5: 型号 GD25LQ128D 的部分信息 .....	12
图 6: 型号 W25Q128JW 的部分信息 .....	12
图 7: 特性介绍 .....	15
图 8: 状态存储器介绍 .....	16
图 9: 支持的命令 .....	16
图 10: 应用程序示例初始化 .....	21
图 11: 初始化引脚和挂载设备 .....	22
图 12: ql_init_demo_thread() .....	23
图 13: COM 设备图 .....	23
图 14: 6 线 SPI NOR Flash 文件系统挂载 Log 信息 .....	24
图 15: 未读取到 MID 打印 Log .....	24
图 16: 未找到型号信息 .....	24
图 17: 挂载所挂载的 NOR Flash 失败打印 Log .....	25

# 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]。

本文档主要提供移远通信 ECx00U 系列和 EGx00U QuecOpen®模块的 6 线 SPI NOR Flash 文件系统挂载的相关 API、开发示例及调试。

## 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 hal\_spi\_flash\_prop.csv

ECx00U 系列和 EGx00U QuecOpen CSDK 代码中提供了 NOR Flash 配置文件 *hal\_spi\_flash\_prop.csv*，位于 `\components\hal\src` 路径下，用户可以在该表格里自行添加相应的 Flash 型号信息。

本章节主要介绍目前支持的 Flash 型号的相关参数信息。

### 备注

ECx00U 系列和 EGx00U QuecOpen SDK 代码中未开放文件 *hal\_spi\_flash\_prop.csv*，所以不支持自行添加 Flash 型号信息。

### 2.1. 支持的 Flash 型号信息

	mid	capacity	type	wp_type	cpid_type	uid_type
XT25W32B	0x16600b	0x400000	HAL_SPI_FLASH_TYPE_XTX	HAL_SPI_FLASH_WP_GD	0	HAL_SPI_FLASH_UID_SFDP_194H_16
XT25W64B	0x17600b	0x800000	HAL_SPI_FLASH_TYPE_XTX	HAL_SPI_FLASH_WP_GD	0	HAL_SPI_FLASH_UID_SFDP_194H_16
XM25QU64A	0x173820	0x800000	HAL_SPI_FLASH_TYPE_XMCA	HAL_SPI_FLASH_WP_XMCA	0	HAL_SPI_FLASH_UID_SFDP_80H_12
XM25QU64B	0x175020	0x800000	HAL_SPI_FLASH_TYPE_XMCB	0	0	HAL_SPI_FLASH_UID_4BH_16
XM25QU32C	0x165020	0x400000	HAL_SPI_FLASH_TYPE_XMCC	HAL_SPI_FLASH_WP_GD	0	HAL_SPI_FLASH_UID_4BH_8
XM25QU16C	0x155020	0x200000	HAL_SPI_FLASH_TYPE_XMCC	HAL_SPI_FLASH_WP_GD	0	HAL_SPI_FLASH_UID_4BH_8
GD25LE64E	0x1760c8	0x800000	HAL_SPI_FLASH_TYPE_GD	HAL_SPI_FLASH_WP_GD	HAL_SPI_FLASH_CPID_4BH	HAL_SPI_FLASH_UID_4BH_8
GD25LQ128C	0x1860c8	0x1000000	HAL_SPI_FLASH_TYPE_GD	HAL_SPI_FLASH_WP_GD	HAL_SPI_FLASH_CPID_4BH	HAL_SPI_FLASH_UID_4BH_8
GD25Q127C	0x1840c8	0x1000000	HAL_SPI_FLASH_TYPE_GD	0	0	HAL_SPI_FLASH_UID_4BH_16
W25Q64JV	0x1740ef	0x800000	HAL_SPI_FLASH_TYPE_WINBOND	HAL_SPI_FLASH_WP_GD	HAL_SPI_FLASH_CPID_4BH	HAL_SPI_FLASH_UID_4BH_8
P25Q128L	0x186085	0x1000000	HAL_SPI_FLASH_TYPE_PUYA	HAL_SPI_FLASH_WP_GD	0	HAL_SPI_FLASH_UID_4BH_16
P25Q64L	0x176085	0x800000	HAL_SPI_FLASH_TYPE_PUYA	HAL_SPI_FLASH_WP_GD	0	HAL_SPI_FLASH_UID_4BH_16
P25Q32L	0x166085	0x400000	HAL_SPI_FLASH_TYPE_PUYA	HAL_SPI_FLASH_WP_GD	0	HAL_SPI_FLASH_UID_4BH_16
GD	0x0000C8	0x000000	HAL_SPI_FLASH_TYPE_GD	HAL_SPI_FLASH_WP_GD	HAL_SPI_FLASH_CPID_4BH	HAL_SPI_FLASH_UID_4BH_8
WINBOND	0x0000EF	0x000000	HAL_SPI_FLASH_TYPE_WINBOND	HAL_SPI_FLASH_WP_GD	HAL_SPI_FLASH_CPID_4BH	HAL_SPI_FLASH_UID_4BH_8
XM25QU128	0x184120	0x1000000	HAL_SPI_FLASH_TYPE_XMCC	HAL_SPI_FLASH_WP_XMCA	HAL_SPI_FLASH_CPID_4BH	HAL_SPI_FLASH_UID_4BH_8
W25Q128JW	0x1860ef	0x1000000	HAL_SPI_FLASH_TYPE_WINBOND	HAL_SPI_FLASH_WP_GD	HAL_SPI_FLASH_CPID_4BH	HAL_SPI_FLASH_UID_4BH_8
W25Q32JW	0x1660ef	0x400000	HAL_SPI_FLASH_TYPE_WINBOND	HAL_SPI_FLASH_WP_GD	HAL_SPI_FLASH_CPID_4BH	HAL_SPI_FLASH_UID_4BH_8
GD25LQ32E	0x1660c8	0x400000	HAL_SPI_FLASH_TYPE_GD	HAL_SPI_FLASH_WP_GD	HAL_SPI_FLASH_CPID_4BH	HAL_SPI_FLASH_UID_4BH_8

图 1：支持的 Flash 型号的相关参数信息（1）

	mid	sreg_block_size	sreg_min_num	sreg_max_num	volatile_sr_en	suspend_en	sfdp_en	write_sr12	has_sr2	has_sus1	has_sus2
XT25W32B	0x16600b	1024	0	0	1	0	1	1	1	0	0
XT25W64B	0x17600b	1024	0	0	1	0	1	1	1	0	0
XM25QU64A	0x173820	0	0	0	1	0	1	0	0	0	0
XM25QU64B	0x175020	256	0	3	0	1	1	0	0	0	0
XM25QU32C	0x165020	256	1	3	1	0	1	0	1	1	0
XM25QU16C	0x155020	256	1	3	1	0	1	0	1	1	0
GD25LE64E	0x1760c8	1024	1	3	1	1	1	1	1	1	1
GD25LQ128C	0x1860c8	512	1	3	1	1	1	1	1	1	1
GD25Q127C	0x1840c8	1024	1	3	0	1	1	0	1	1	1
W25Q64JV	0x1740ef	256	1	3	1	1	1	1	1	1	0
P25Q128L	0x186085	1024	1	3	1	0	1	0	1	1	1
P25Q64L	0x176085	1024	1	3	1	1	1	0	1	1	1
P25Q32L	0x166085	1024	1	3	1	1	1	0	1	1	1
GD	0x0000C8	256	1	3	1	1	1	1	1	1	1
WINBOND	0x0000EF	256	1	3	1	1	1	1	1	1	0
XM25QU128	0x184120	1024	1	3	1	1	1	1	1	1	1
W25Q128JW	0x1860ef	256	1	3	1	1	1	1	1	1	0
W25Q32JW	0x1660ef	256	1	3	1	1	1	1	1	1	0
GD25LQ32E	0x1660c8	1024	1	3	1	1	1	1	1	1	1

图 2：支持的 Flash 型号的相关参数信息（2）

## 备注

上图中所示的 Flash 型号未全都调试过，有关已调试过型号的具体信息，请咨询移远通信技术支持。

## 2.2. 相关参数含义

### 2.2.1. halSpiFlash\_t

文件 *hal\_spi\_flash\_prop.csv* 中的 Flash 型号的相关参数信息均与 *components\hal\include\hal\_spi\_flash.h* 中的结构体 *halSpiFlash\_t* 的参数相对应。结构体信息定义如下：

```
typedef struct halSpiFlash
{
    uintptr_t hwp;
    unsigned mid;
    unsigned capacity;
    uint16_t sreg_block_size;
    uint8_t type : 4;
    uint8_t wp_type : 4;
    uint8_t uid_type : 4;
    uint8_t cpid_type : 4;
    uint8_t sreg_min_num : 4;
    uint8_t sreg_max_num : 4;
    bool volatile_sr_en : 1;
    bool suspend_en : 1;
    bool sfdp_en : 1;
    bool write_sr12 : 1;
    bool has_sr2 : 1;
    bool has_sus1 : 1;
```

```
bool has_sus2 : 1;
} halSpiFlash_t
```

## ● 参数

类型	参数	描述
uintptr_t	<i>hwp</i>	6 线 SPI 硬件寄存器
unsigned	<i>mid</i>	mid = Manufacturer ID 后添加 Device ID。采用低位在前的方式记录。详情请参考第 2.2.1.1 章。
unsigned	<i>capacity</i>	容量大小。单位：字节。 如：型号 XM25QU32C：32 Mbit，即 0x400000 型号 XM25QU64B：64 Mbit，即 0x800000 型号 GD25LQ128C：128 Mbit，即 0x1000000
uint16_t	<i>sreg_block_size</i>	安全寄存器组大小。取值一般为 256、512、1024 和 2048。单位：字节。大小不超过 0xffff 字节。0 表示不支持。详情请参考第 2.2.1.2 章。
uint8_t	<i>type</i>	内置 Flash 的厂商型号。详情请参考第 2.2.1.3 章。
uint8_t	<i>wp_type</i>	保护类型。详情请参考第 2.2.1.7 章。
uint8_t	<i>uid_type</i>	UID 类型。详情请参考第 2.2.1.4 章。
uint8_t	<i>cpid_type</i>	CPID 类型。详情请参考第 2.2.1.5 章。
uint8_t	<i>sreg_min_num</i>	最小安全寄存器编号。0 表示不支持。
uint8_t	<i>sreg_max_num</i>	最大安全寄存器编号。0 表示不支持。
bool	<i>volatile_sr_en</i>	是否支持易失性状态寄存器。0：不支持；1：支持。
bool	<i>suspend_en</i>	是否支持读写操作的挂起与恢复。0：不支持；1：支持。
bool	<i>sfdp_en</i>	是否支持 5AH 命令读取 SFDP。0：不支持；1：支持。
bool	<i>write_sr12</i>	是否支持 01H 命令写 SR1 和 SR2 状态寄存器。 0：不支持；1：支持。
bool	<i>has_sr2</i>	是否支持 35H 命令读取 SR2 状态寄存器。 0：不支持；1：支持。
bool	<i>has_sus1</i>	状态寄存器的 S15 是否存在 SUS1。0：不存在；1：存在。
bool	<i>has_sus2</i>	状态寄存器的 S10 是否存在 SUS2。0：不存在；1：存在。

## 2.2.1.1. mid

mid = Manufacturer ID 后添加 Device ID，采用低位在前的方式记录。将从 NOR Flash 上读取到的 mid 值与表格中的 mid 值逐一比较，若对应的值相同则读出表格中的所有信息，若未找到相同值则表示不支持该 Flash 型号。可参考下文中的示例说明。

示例一：由下图型号 GD25LQ128D datasheet 中的 mid 定义，可知 mid 值为 0x1860c8（低位在前）。

**Table of ID Definitions:  
GD25LQ128D**

Operation Code	M7-M0	ID15-ID8	ID7-ID0
9FH	C8	60	18
90H	C8		17
ABH			17

图 3：型号 GD25LQ128D mid

示例二：由下图型号 W25Q128JW datasheet 中的 mid 定义，可知 mid 值为 0x1860ef（低位在前）。

### 8.1.1 Manufacturer and Device Identification

<b>MANUFACTURER ID</b>	<b>(MF7 - MF0)</b>	
Winbond Serial Flash	EFh	
<b>Device ID</b>	<b>(ID7 - ID0)</b>	<b>(ID15 - ID0)</b>
<b>Instruction</b>	<b>ABh, 90h, 92h, 94h</b>	<b>9Fh</b>
W25Q128JW-IQ/JQ	17h	6018h
W25Q128JW-IM/JM*	17h	8018h

图 4：型号 W25Q128JW mid

## 2.2.1.2. security register

Flash 型号是否支持安全寄存器，需根据该型号的 datasheet 来确认。具体方法可参考下文中的示例说明。

示例一：下图为型号 GD25LQ128D datasheet 中的部分信息。

The GD25LQ128D provides three 1024-byte Security Registers which can be erased and programmed individually. These registers may be used by the system manufacturers to store security and other important information separately from the main memory array.

Address	A23-16	A15-12	A11-10	A9-0
Security Register #1	00H	0 0 0 1	0 0	Don't Care
Security Register #2	00H	0 0 1 0	0 0	Don't Care
Security Register #3	00H	0 0 1 1	0 0	Don't Care

图 5：型号 GD25LQ128D 的部分信息

由上面 datasheet 中的信息可知：  
sreg\_block\_size: 1024。安全寄存器组大小为 1024 字节。  
sreg\_min\_num: 1。最小安全寄存器编号为 1。  
sreg\_max\_num: 3。最大安全寄存器编号为 3。

示例二：下图为 W25Q128JW datasheet 中的部分信息。

- At least one byte of data input is required for Page Program, Quad Page Program and Program Security Registers, up to 256 bytes of data input. If more than 256 bytes of data are sent to the device, the addressing will wrap to the beginning of the page and overwrite previously sent data.
- Write Status Register-1 (01h) can also be used to program Status Register-1&2, see section 8.2.5.
- Security Register Address:

Security Register 1: A23-16 = 00h; A15-8 = 10h; A7-0 = byte address  
Security Register 2: A23-16 = 00h; A15-8 = 20h; A7-0 = byte address  
Security Register 3: A23-16 = 00h; A15-8 = 30h; A7-0 = byte address

图 6：型号 W25Q128JW 的部分信息

由上面 datasheet 中的信息可知：  
sreg\_block\_size: 256。安全寄存器组大小为 256 字节。  
sreg\_min\_num: 1。最小安全寄存器编号为 1。  
sreg\_max\_num: 3。最大安全寄存器编号为 3。

### 2.2.1.3. halSpiFlashType\_t

内置 Flash 厂商型号枚举信息如下：

```
typedef enum
{
    HAL_SPI_FLASH_TYPE_UNKNOWN,
    HAL_SPI_FLASH_TYPE_GD,
    HAL_SPI_FLASH_TYPE_WINBOND,
    HAL_SPI_FLASH_TYPE_XTX,
    HAL_SPI_FLASH_TYPE_XMCA,
    HAL_SPI_FLASH_TYPE_XMCC,
```

```
HAL_SPI_FLASH_TYPE_XMCB,
HAL_SPI_FLASH_TYPE_PUYA,
} halSpiFlashType_t
```

#### ● 参数

参数	描述
<i>HAL_SPI_FLASH_TYPE_UNKNOWN</i>	未知类型
<i>HAL_SPI_FLASH_TYPE_GD</i>	GigaDevice
<i>HAL_SPI_FLASH_TYPE_WINBOND</i>	WINBOND
<i>HAL_SPI_FLASH_TYPE_XTX</i>	XTX
<i>HAL_SPI_FLASH_TYPE_XMCA</i>	XMC-A 批次
<i>HAL_SPI_FLASH_TYPE_XMCC</i>	XMC-C 批次
<i>HAL_SPI_FLASH_TYPE_XMCB</i>	XMC-B 批次
<i>HAL_SPI_FLASH_TYPE_PUYA</i>	PUYA

#### 备注

若所选 Flash 型号非上述的厂商型号，可以对比 **datasheet** 尝试选用一个比较相似的型号，但是需要验证是否能成功驱动 Flash。如未能成功驱动 Flash，可联系移远通信技术支持。

#### 2.2.1.4. halSpiFlashUidType\_t

UID 类型枚举如下：

```
typedef enum
{
    HAL_SPI_FLASH_UID_NONE,
    HAL_SPI_FLASH_UID_4BH_8,
    HAL_SPI_FLASH_UID_4BH_16,
    HAL_SPI_FLASH_UID_SFDP_80H_12,
    HAL_SPI_FLASH_UID_SFDP_94H_16,
    HAL_SPI_FLASH_UID_SFDP_194H_16,
} halSpiFlashUidType_t
```

- 参数

参数	描述
<i>HAL_SPI_FLASH_UID_NONE</i>	无 UID 信息
<i>HAL_SPI_FLASH_UID_4BH_8</i>	读取 UID 的命令为 <b>4BH</b> ，UID 长度为 8 bit
<i>HAL_SPI_FLASH_UID_4BH_16</i>	读取 UID 的命令为 <b>4BH</b> ，UID 长度为 16 bit
<i>HAL_SPI_FLASH_UID_SFDP_80H_12</i>	UID 在 SFDP Table 地址偏移 80H 处，长度为 12 bit
<i>HAL_SPI_FLASH_UID_SFDP_94H_16</i>	UID 在 SFDP Table 地址偏移 94H 处，长度为 16 bit
<i>HAL_SPI_FLASH_UID_SFDP_194H_16</i>	UID 在 SFDP Table 地址偏移 194H 处，长度为 16 bit

#### 2.2.1.5. halSpiFlashCpidType\_t

CPID 类型枚举如下：

```
typedef enum
{
    HAL_SPI_FLASH_CPID_NONE,
    HAL_SPI_FLASH_CPID_4BH, // byte [17:16]
} halSpiFlashCpidType_t
```

- 参数

参数	描述
<i>HAL_SPI_FLASH_CPID_NONE</i>	无 CPID 信息
<i>HAL_SPI_FLASH_CPID_4BH</i>	读取 CPID 的命令为 <b>4BH</b>

#### 2.2.1.6. halSpiFlashWpType\_t

写保护类型枚举信息如下：

```
typedef enum
{
    HAL_SPI_FLASH_WP_NONE,
    HAL_SPI_FLASH_WP_GD,
    HAL_SPI_FLASH_WP_XMCA,
} halSpiFlashWpType_t
```

● 参数

参数	描述
<code>HAL_SPI_FLASH_WP_NONE</code>	无写保护功能
<code>HAL_SPI_FLASH_WP_GD</code>	写保护兼容格式为 BP0/1/2/3/4/CMP
<code>HAL_SPI_FLASH_WP_XMCA</code>	写保护兼容格式为 SR2/3/4/5

### 2.2.1.7. 其他特性参数

以型号 GD25LQ128D 为例，对 `hal_spi_flash_prop.csv` 中的其他的参数进行分析。下图为 GD25LQ128D datasheet 的部分信息，分别是手册中特性介绍、状态寄存器介绍、支持的命令。

## 1. FEATURES

- ◆ 128M-bit Serial Flash
  - 16384K-byte
  - 256 bytes per programmable page
- ◆ Standard, Dual, Quad SPI, QPI
  - Standard SPI: SCLK, CS#, SI, SO, WP#, HOLD#
  - Dual SPI: SCLK, CS#, IO0, IO1, WP#, HOLD#
  - Quad SPI: SCLK, CS#, IO0, IO1, IO2, IO3
  - QPI: SCLK, CS#, IO0, IO1, IO2, IO3
- ◆ High Speed Clock Frequency
  - 120MHz for fast read with 30PF load
  - Dual I/O Data transfer up to 240Mbps/s
  - Quad I/O Data transfer up to 480Mbps/s
  - QPI Mode Data transfer up to 480Mbps/s
- ◆ Software/Hardware Write Protection
  - Write protect all/portion of memory via software
  - Enable/Disable protection with WP# Pin
  - Top/Bottom Block protection
- ◆ Allows XIP (execute in place) Operation
  - Continuous Read With 8/16/32/64-byte Wrap
- ◆ Minimum 100,000 Program/Erase Cycles
- ◆ Fast Program/Erase Speed
  - Page Program time: 0.5ms typical
  - Sector Erase time: 70ms typical
  - Block Erase time: 0.16/0.3s typical
  - Chip Erase time: 50s typical
- ◆ Flexible Architecture
  - Uniform Sector of 4K-byte
  - Uniform Block of 32/64K-byte
  - Erase/Program Suspend/Resume
- ◆ Low Power Consumption
  - 35µA typical stand-by current
  - 1µA typical power down current
- ◆ Advanced security Features
  - 128-bit Unique ID for each device
  - 3x1024-Byte Security Registers With OTP Lock
- ◆ Single Power Supply Voltage
  - Full voltage range: 1.65~2.0V
- ◆ Data Retention
  - 20-year data retention typical

图 7：特性介绍



## 6. STATUS REGISTER

S15	S14	S13	S12	S11	S10	S9	S8
SUS1	CMP	LB3	LB2	LB1	SUS2	QE	SRP1

S7	S6	S5	S4	S3	S2	S1	S0
SRP0	BP4	BP3	BP2	BP1	BP0	WEL	WIP

### SUS1, SUS2 bits

The SUS1 and SUS2 bits are read only bit in the status register (S15 and S10) that are set to 1 after executing an Program/Erase Suspend (75H) command (The Erase Suspend will set the SUS1 to 1, and the Program Suspend will set the SUS2 to 1). The SUS1 and SUS2 bits are cleared to 0 by Program/Erase Resume (7AH) command, software reset (66H+99H) command as well as a power-down, power-up cycle.

图 8：状态存储器介绍

WRITE ENABLE (WREN) (06H) .....
WRITE DISABLE (WRDI) (04H) .....
WRITE ENABLE FOR VOLATILE STATUS REGISTER (50H) .....
READ STATUS REGISTER (RDSR) (05H OR 35H OR 15H) .....
WRITE STATUS REGISTER (WRSR) (01H) .....
PROGRAM/ERASE SUSPEND (PES) (75H) .....
PROGRAM/ERASE RESUME (PER) (7AH) .....
READ UNIQUE ID (4BH) .....
ERASE SECURITY REGISTERS (44H) .....
PROGRAM SECURITY REGISTERS (42H) .....
READ SECURITY REGISTERS (48H) .....
READ SERIAL FLASH DISCOVERABLE PARAMETER (5AH) .....

图 9：支持的命令

由上面 datasheet 中红框框出来的信息可知：

*capacity*: 0x1000000。型号 GD25LQ128C: 128 Mbit。

*volatile\_sr\_en*: 1。支持易失性状态寄存器。

*suspend\_en*: 1。支持读写操作的挂起与恢复。

*sfdp\_en*: 1。此功能一般都支持。

*write\_sr12*: 1。支持 01H 命令写 SR1 和 SR2 状态寄存器。

*has\_sr2*: 1。支持 35H 命令读取 SR2 状态寄存器。

*has\_sus1*: 1。状态寄存器的 S15 存在 SUS1。

*has\_sus2*: 1。状态寄存器的 S10 存在 SUS2。

# 36 线 SPI NOR Flash 文件系统挂载相关 API

## 3.1. 头文件

`ql_api_spi6_ext_nor_flash.h` 为 6 线 SPI NOR Flash 文件系统挂载的头文件，位于 `components\ql-kernel\inc` 目录下。

## 3.2. 函数概览

表 2：函数概览

函数	说明
<code>ql_spi6_ext_nor_flash_init()</code>	初始化专用 6 线 SPI 引脚
<code>ql_spi6_ext_nor_flash_deinit()</code>	反初始化专用 6 线 SPI 引脚
<code>ql_spi6_ext_nor_flash_sffs_mkfs()</code>	格式化所挂载的 NOR Flash
<code>ql_spi6_ext_nor_flash_sffs_mount()</code>	挂载所挂载的 NOR Flash
<code>ql_spi6_ext_nor_flash_sffs_unmount()</code>	反挂载所挂载的 NOR Flash

## 3.3. 函数详解

### 3.3.1. ql\_spi6\_ext\_nor\_flash\_init

该函数用于初始化专用 6 线 SPI 引脚，配置成 6 线 SPI 功能。

- 函数原型

```
void ql_spi6_ext_nor_flash_init(void)
```

- 参数

无

- 返回值

无

### 3.3.2. ql\_spi6\_ext\_nor\_flash\_deinit

该函数用于反初始化专用 6 线 SPI 引脚，将专用 6 线 SPI 引脚恢复成默认功能。

- 函数原型

```
void ql_spi6_ext_nor_flash_deinit(void)
```

- 参数

无

- 返回值

无

### 3.3.3. ql\_spi6\_ext\_nor\_flash\_sffs\_mkfs

该函数用于格式化所挂载的 NOR Flash。函数执行成功后通过文件系统再次读取所挂载的 NOR Flash 时将显示为空。

- 函数原型

```
ql_errcode_spi6_nor_e ql_spi6_ext_nor_flash_sffs_mkfs(void)
```

- 参数

无

- 返回值

详情请参考第 3.3.3.1 章。

### 3.3.3.1. ql\_errcode\_spi6\_nor\_e

6 线 SPI NOR Flash 文件系统挂载的 API 错误码，表示 API 是否执行成功。若 API 执行失败，则返回错误原因，枚举信息定义如下：

```
typedef enum
{
    QL_SPI6_EXT_NOR_FLASH_SUCCESS,
    QL_SPI6_EXT_NOR_FLASH_NOT_INIT_ERROR,
    QL_SPI6_EXT_NOR_FLASH_CREATE_FBDEV2_ERROR,
    QL_SPI6_EXT_NOR_FLASH_REPEAT_MOUNT_ERROR,
    QL_SPI6_EXT_NOR_FLASH_FIND_BLOCK_DEVICE_ERROR,
    QL_SPI6_EXT_NOR_FLASH_SFFS_MOUNT_ERROR,
    QL_SPI6_EXT_NOR_FLASH_SFFS_UNMOUNT_ERROR,
    QL_SPI6_EXT_NOR_FLASH_PARTITION_INFO_ERROR
}ql_errcode_spi6_nor_e
```

#### ● 参数

参数	描述
QL_SPI6_EXT_NOR_FLASH_SUCCESS	API 执行成功
QL_SPI6_EXT_NOR_FLASH_NOT_INIT_ERROR	初始化失败
QL_SPI6_EXT_NOR_FLASH_CREATE_FBDEV2_ERROR	创建块设备错误
QL_SPI6_EXT_NOR_FLASH_REPEAT_MOUNT_ERROR	重复挂载错误
QL_SPI6_EXT_NOR_FLASH_FIND_BLOCK_DEVICE_ERROR	未找到块设备报错
QL_SPI6_EXT_NOR_FLASH_SFFS_MOUNT_ERROR	挂载文件系统错误
QL_SPI6_EXT_NOR_FLASH_SFFS_UNMOUNT_ERROR	反挂载文件系统错误
QL_SPI6_EXT_NOR_FLASH_PARTITION_INFO_ERROR	分区信息错误

### 3.3.4. ql\_spi6\_ext\_nor\_flash\_sffs\_mount

该函数用于挂载所挂载的 NOR Flash，挂载成功后，才能正常使用文件系统接口函数进行读写。

- 函数原型

```
ql_errcode_spi6_nor_e ql_spi6_ext_nor_flash_sffs_mount(void)
```

- 参数

无

- 返回值

详情请参考第 3.3.3.1 章。

### 3.3.5. ql\_spi6\_ext\_nor\_flash\_sffs\_unmount

该函数用于反挂载所挂载的 NOR Flash。

- 函数原型

```
ql_errcode_spi6_nor_e ql_spi6_ext_nor_flash_sffs_unmount(void)
```

- 参数

无

- 返回值

详情请参考第 3.3.3.1 章。

# 4 示例及调试

本章节主要介绍在 App 侧如何使用上述 API 进行 6 线 SPI NOR Flash 文件系统挂载开发以及简单的调试。

## 4.1. App 的 6 线 SPI NOR Flash 文件系统挂载开发示例

### 4.1.1. 6 线 SPI NOR Flash 文件系统挂载应用程序示例说明

ECx00U 系列和 EGx00U QuecOpen CSDK 代码中提供了 6 线 SPI NOR Flash 文件系统挂载的示例，即 `spi6_ext_nor_flash_demo.c` 文件，位于 `\components\ql-application\spi6_ext_nor_flash` 目录下。进行开发前要先确认所使用的 TE-A 是否焊接了 NOR Flash，是否与使用的 6 线 SPI 接口一致。

该应用程序示例需要先初始化专用 6 线 SPI 引脚，然后将其挂载到文件系统上，再调用文件系统接口操作所挂载的 NOR Flash，详细信息可参考文档 [2]。

```

QIosStatus ql_spi6_ext_nor_flash_demo_init(void)
{
    QIosStatus err = QL_OSI_SUCCESS;

    //QL_SPI6_EXT_NOR_LOG("enter ql_spi6_ext_nor_flash_demo_init");
    err = ql_rtos_task_create(&spi6_ext_nor_flash_demo_task, SPI6_EXT_NOR_FLASH_DEMO_TASK_STACK_SIZE, SPI6_EXT_NOR_FLASH_DEMO_TASK_Prio, "ql_spi6_nor", .
    if(err != QL_OSI_SUCCESS)
    {
        QL_SPI6_EXT_NOR_LOG("demo_task created failed");
        return err;
    }

    return err;
}
    
```

图 10: 应用程序示例初始化

```

static void ql_spi6_ext_nor_flash_demo_task_pthread(void *ctx)
{
    QLOSStatus erro = 0;
    ql_errcode_spi6_nor_e ret;

    ql_spi6_ext_nor_flash_init();
    ret = ql_spi6_ext_nor_flash_sffs_mount();
    if(QL_SPI6_EXT_NOR_FLASH_SUCCESS != ret)
    {
#ifdef QL_APP_FEATURE_FILE
        int64 free_size = ql_fs_free_size("EFS:");
        QL_SPI6_EXT_NOR_LOG("fs free_size :%d", free_size);
#endif
        goto _QL_SPI6_TASK_EXIT;
    }
#ifdef QL_APP_FEATURE_FILE
    >> int fd = 0;
    >> int64 err = 0;
    >> char buffer[100];
    >> char *str = QL_EFS_TEST_STR;

    int64 free_size = ql_fs_free_size("EFS:");
    QL_SPI6_EXT_NOR_LOG("fs free_size :%d", free_size);

    fd = ql_fopen(QL_EFS_FILE_PATH, "wb+");
    if(fd < 0)
    {
        >> QL_SPI6_EXT_NOR_LOG("open file failed");
        >> err = fd;
        >> goto _lexit;
    }

    err = ql_fwrite(str, strlen(str) + 1, 1, fd); //strlen not include '\0'
    if(err < 0)
    {
        >> QL_SPI6_EXT_NOR_LOG("write file failed");
        >> ql_fclose(fd);
        >> goto _lexit;>>>
    }

    err = ql_frewind(fd);
    if(err < 0)
    {
        >> QL_SPI6_EXT_NOR_LOG("rewind file failed");
        >> ql_fclose(fd);
        >> goto _lexit;>>> >> >>
    }

    err = ql_fread(buffer, ql_fsize(fd), 1, fd);

```

图 11: 初始化引脚和挂载设备

### 4.1.2. 应用程序示例自启动

上述应用程序示例已在 `ql_init_demo_thread()` 线程中默认不启动，如下图所示，需要测试时请取消注释。

```
static void ql_init_demo_thread(void *param)
{
    QL_INIT_LOG("init demo thread enter, param 0x%x", param);

    /*Caution:If the macro of secure boot and the function are opened, download firmware and restart will enable secure boot.
    the secret key cannot be changed forever*/
    #ifdef QL_APP_FEATURE_SECURE_BOOT
        //ql_dev_enable_secure_boot();
    #endif

    #ifdef QL_APP_FEATURE_SPI_NOR_FLASH
        //ql_spi_nor_flash_demo_init();
    #endif

    #ifdef QL_APP_FEATURE_SPI6_EXT_NOR
        ql_spi6_ext_nor_flash_demo_init();
    #endif

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

图 12: ql\_init\_demo\_thread()

## 4.2. 6 线 SPI NOR Flash 文件系统挂载相关调试

### 4.2.1. 调试准备

ECx00U 系列和 EGx00U QuecOpen 模块可通过使用移远通信 LTE OPEN EVB 板进行 6 线 SPI NOR Flash 文件系统挂载功能调试。

编译固件版本烧录到 ECx00U 系列和 EGx00U QuecOpen 模块中，使用 USB 线连接 LTE OPEN EVB 板的 USB 端口和 PC，USB AP log 端口主要用于显示系统调试信息，通过 `coolwatcher_debughost.exe` 的 Trace Tool 打印的 log 信息可以看到该应用程序示例的相关信息。

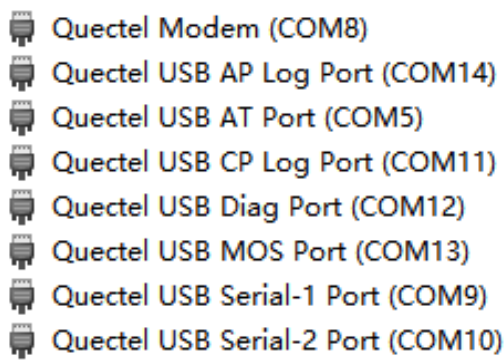


图 13: COM 设备图



## 备注

EC200U 系列模块的 TE-A 没有外挂 NOR Flash，需要自行飞线外接 NOR Flash；EG500U-CN 模块的 TE-A 需要修改兼容性电阻，具体可以咨询移远通信技术支持。

### 4.2.2. 6 线 SPI NOR Flash 文件系统挂载功能调试

模块开机后自动启动 `ql_spi6_ext_nor_flash_app_init()`，通过 log 信息可以看到 6 线 SPI NOR Flash 文件系统挂载的相关信息。AP log 端口调试信息如下图所示：

QUEV/I	[FS M][quec_set_spi6_ext_nor_flash_capacity, 156] ext flash capacity:0xf80000
QOPN/I	[ql_SPI6_NOR_DEMO][ql_spi6_ext_nor_flash_demo_task_pthread, 63] fs free_size :15797892
QOPN/I	[ql_SPI6_NOR_DEMO][ql_spi6_ext_nor_flash_demo_task_pthread, 97] file read result is 1234567890abcdefg
QOPN/I	[ql_SPI6_NOR_DEMO][ql_spi6_ext_nor_flash_demo_task_pthread, 108] open file failed,errcode is 83030260
QOPN/I	[ql_SPI6_NOR_DEMO][ql_spi6_ext_nor_flash_demo_task_pthread, 112] sffs mount ret 0
QOPN/I	[ql_SPI6_NOR_DEMO][ql_spi6_ext_nor_flash_demo_task_pthread, 127] close file
QOPN/I	[ql_SPI6_NOR_DEMO][ql_spi6_ext_nor_flash_demo_task_pthread, 147] ql_rtos_task_delete

图 14：6 线 SPI NOR Flash 文件系统挂载 Log 信息

如果模块初始化失败，一般为以下两种情况：

第一种：读取不到 MID 或型号不支持。此时初始化失败打印 log 如下：

```
/I : Not support nor mid:0x0
FSYS/I : failed to open spi flash SFL2
QUEV/I : [FS M][quec_spi6_ext_nor_flash_mount, 185] failed to find block device FEXT
QOPN/I : [ql_SPI6_NOR_DEMO][ql_spi6_ext_nor_flash_demo_task_pthread, 52] fs free_size :-2096954973
QOPN/I : [ql_SPI6_NOR_DEMO][ql_spi6_ext_nor_flash_demo_task_pthread, 147] ql_rtos_task_delete
```

图 15：未读取到 MID 打印 Log

如上图所示，若 mid 为 0x0，表示读不到实际的 MID 值，可能原因如下：

1. 不存在外挂所挂载的 NOR Flash；
2. 专用 6 线 SPI 引脚未初始化成功，可能原因如下：
  - 1) 未调用初始化函数；
  - 2) `quec_boot_pin_map.c` 中的 `quec_boot_spi6_pin_cfg_map[]` 数组中参数配置不正确；
  - 3) 配置的 6 线 SPI 接口与实际不一致。

16:32:49.011	5400	/I	Not support nor mid:0x218c4
--------------	------	----	-----------------------------

图 16：未找到型号信息

如上图所示，若 mid 不为 0，则说明目前 `hal_spi_flash_prop.csv` 文件中找不到该型号的信息，用户可以自行添加相应的信息到 `hal_spi_flash_prop.csv` 文件中。详情请参考第 2 章。

第二种：挂载所挂载的 NOR Flash 不成功。此时初始化失败打印 log 如下：

FSYS/I	FBDEV: scan EB quickinit/0 invalid/3968 same1b/0 erase count/15728640/0 next use/0
FSYS/I	SFFS mount, device block size/500 block count/31735 cache count 4
FSYS/E	SFFS mount wrong root tag
FSYS/E	failed to mount /ext
QOPN/I	[ql_SPI6_NOR_DEMO][ql_spi6_ext_nor_flash_demo_task_pthread, 52] fs free_size :1242392
QOPN/I	[ql_SPI6_NOR_DEMO][ql_spi6_ext_nor_flash_demo_task_pthread, 147] ql_rtos_task_delete
QOPN/I	[ql_OSI][ql_rtos_task_delete, 192] remove task 80BFC4E0 ok

图 17：挂载所挂载的 NOR Flash 失败打印 Log

若挂载设备不成功，需先格式化再重新进行设备挂载。解决方法如下：

1. 调用 `ql_spi6_ext_nor_flash_sffs_mkfs()` 接口进行格式化所挂载的 NOR Flash；
2. 使用 **AT+QEFSMKFS="SFFS"** 命令格式化所挂载的 NOR Flash。

## 5 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_文件系统 API_参考手册

表 4：术语缩写

缩写	英文全称	中文全称
AP	Access Point	接入点
API	Application Programming Interface	应用程序接口
CPID	Consumer Product Identification	消费产品标识
CSDK	Customized Software Development Kit	定制的软件开发工具包
EVB	Evaluation Board	评估板
MID	Manufacture Identification	制造识别
PC	Personal Computer	个人电脑
RTOS	Real-time Operating System	实时操作系统
SFDP	Serial Flash Discoverable Parameters	可供查询的串行 Flash 功能参数表
SPI	Serial Peripheral Interface	串行外设接口
UID	Unique Identification	唯一标识
USB	Universal Serial Bus	通用串行总线