QM10XD 分区一键修改操作 SOP 文档

文档信息

项目	内容
文档名称	分区一键修改操作 SOP
适用场景	项目分区配置修改（支持普通项目、nor/nand 兼容项目）
操作目的	标准化分区修改流程，确保分区配置修改准确、可追溯
操作权限	项目开发权限、脚本执行权限
文档版本	V1.0
编制日期	2025 年 12 月 24 日
编制人	[马激伟]
审批人	[]

目录

1. 操作前置条件

1.1 明确本次分区修改的需求（需提前确认具体修改内容，如分区大小调整）；

1.2 提前确认目标项目类型：通过查看 product/[项目名]目录下是否存在 partition-config-nor.json、partition-config-nand.json，判断是否为 nor/nand 兼容项目；

1.3 操作前备份关键文件：将项目中原有的 partition-config.json、partition-info.txt（或兼容项目的 partition-config-nor.json、partition-config-nand.json 等）复制到独立备份目录，命名格式建议为“文件名_备份_日期”（如 partition-config.json_backup_20251224），避免操作失误导致配置丢失；

2. 核心操作步骤

2.1 获取并修改 partition-config.json 配置文件

2.1.1 查找/生成基础配置文件

打开终端/命令行工具，切换至目标项目的根目录（示例命令）：

```
# Linux 系统
cd /home/user/projects/[项目名] 例如
cd ~/workspace/code/qm10xd/product/demo_ld/
```

分场景处理：

- 场景 1：找到该文件（确认为 nor/nand 兼容项目）→ 按以下步骤生成：
  i. 切换至 product/[项目名]目录（示例命令：cd product/demo_ld）；
  ii. 完成以下复制操作：
    - 若编译 nor flash 时：将 partition-config-nor.json 复制为 partition-config.json，将 partition-info-nor.txt 复制为 partition-info.txt；
    - 若编译 nand flash 时：将 partition-config-nand.json 复制为 partition-config.json，将 partition-info-nand.txt 复制为 partition-info.txt；

2.1.2 编辑 partition-config.json 配置文件

使用文本编辑器（如 VS Code、Notepad++、Sublime Text）或项目开发工具打开 partition-config.json 文件；

按本次修改需求，调整对应配置项（需严格遵循 JSON 格式规范，禁止出现注释、多余空格或语法错误），示例修改场景如下：

示例 1：调整“data”分区大小（单位：MB）{

```
  "partition_table": [
// 其他分区配置...
    {
      "name": "data",
      "size": "10MB",
      "offset": "0x3300000",
      "type": "userdata",
      "filesystem_type": "ubi"
    },
    // 其他分区配置...
  ]
}
```

修改完成后，执行以下操作：

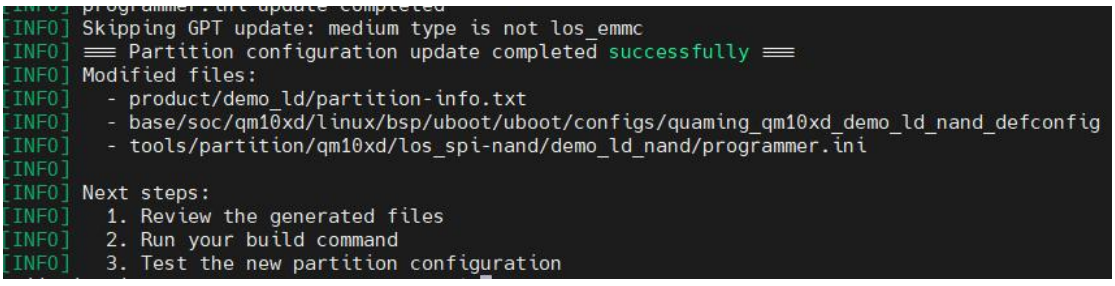
- 保存文件（快捷键：Ctrl+S）并关闭编辑器；

2.2 执行一键分区修改脚本

保持终端/命令行工具在项目根目录，执行以下命令（将[项目名]替换为实际项目名称）：

```
# 脚本绝对路径执行（推荐，避免路径错误）
./tools/partition/update-partition/update-partition.sh [项目名]
# 示例：./tools/partition/update-partition/update-partition.sh demo_ld
```

脚本执行过程中，终端会输出实时日志，请勿中断进程,输出 log 会提示修改了那些文件。



脚本执行完成后，重点核对输出日志：

- 确认是否存在成功标识（如日志里显示“Partition configuration update completed successfully”）；
- 记录脚本修改的明细信息，包括修改的文件路径等；

2.3 提交修改文件（含特殊兼容项目处理）

2.3.1 梳理修改文件清单

以 Git 版本控制为例，执行以下命令列出本次操作涉及的所有修改文件：

```
git status
```

核心修改文件清单（必含）：

- 基础配置文件：partition-config.json、partition-info.txt；
- 脚本自动更新文件：可能包含 quaming_qm10xd_demo_ld_nand_defconfig、programmer.ini 等（以脚本输出日志为准）；
- 兼容项目额外文件：partition-config-nor.json/partition-config-nand.json、partition-info-nor.txt/partition-info-nand.txt（需手动复制生成，见 2.3.2 节）。

2.3.2 nor/nand 兼容项目特殊处理

若项目为 nor/nand 兼容类型，需将修改后的基础文件同步至对应后缀的兼容文件，避免另一类配置失效，执行以下复制命令：

基于 nor 配置修改的场景：

```
# 复制修改后的文件，覆盖原有 nor 后缀文件
cp partition-config.json partition-config-nor.json
cp partition-info.txt partition-info-nor.txt
```

基于 nand 配置修改的场景：

```
# 复制修改后的文件，覆盖原有 nand 后缀文件
cp partition-config.json partition-config-nand.json
cp partition-info.txt partition-info-nand.txt
```

执行完成后，通过以下命令确认后缀文件已更新（查看文件修改时间）：

```
# Linux 系统
ls -l partition-config-*.json partition-info-*.txt
```

2.3.3 提交版本控制

执行添加命令，将所有修改文件纳入版本控制：

```
# 添加基础文件
git add partition-config.json partition-info.txt
# 兼容项目需额外添加后缀文件
git add partition-config-nor.json partition-info-nor.txt
```

或者

```
git add partition-config-nand.json partition-info-nand.txt
```

```
# 脚本自动修改的其他文件，按实际清单添加（示例）
git add quaming_qm10xd_demo_ld_nand_defconfig
git add programmer.ini
```

填写清晰的提交备注（便于后续追溯），示例:

```
git commit -m "分区配置修改：1. 调整 data 分区大小至 10MB；
```

将提交推送至远程仓库（若需）:

```
git push origin [分支名]  # 示例：git push origin develop
```

提交完成后，在版本控制平台（如 GitLab、GitHub、Gitee）确认提交记录无误，完成整个分区修改流程。