

SD Express 测试工具

1. 简介

SD Express Tester是SD Express卡的测试软件，同时向下兼容SD 4.0, SD 3.0, SD2.0等传统SD模式。工具支持图形界面和命令行两种操作模式，可进行控制器检测、基本读写、性能测试和稳定性测试等功能。

2. 用户指南

2.1. 功能特点

- 双模式操作：图形界面(GUI)和命令行(CLI)
- 自动检测SD卡 and 控制器状态
- 支持多种测试项目：
 - 控制器兼容性检测
 - 基本读写测试（读写数据比较）
 - 性能测试（读写速度）
 - 稳定性测试（随机读写）
- 支持循环测试
- 实时显示测试进度
- 自动生成测试报告
- 详细的日志记录

2.2. 系统要求

- Windows 10/11
- SD Express卡控制器（O2Micro/BayHub系列控制器）

2.3. GUI模式使用说明



1. 运行 SDExpressTester.exe

2. 程序自动检测SD卡和SD/NVMe控制器

3. 界面说明:

- 系统状态:
 - 控制器: 显示当前控制器型号
 - 控制器能力: 显示当前控制器支持的SD卡模式
 - 卡名称: 显示当前检测到的SD卡型号
 - 卡能力: 显示SD卡支持的速度模式
- 控制按钮:
 - 开始测试: 开始执行测试套件
 - 停止测试: 中断当前测试过程
 - 配置文件: 打开配置文件进行编辑
 - 日志文件: 打开日志文件查看详细日志
 - 关于: 显示软件版本和作者信息
- 进度显示:
 - 进度条: 显示当前测试项的完成进度
 - 状态栏: 显示当前测试项状态
- 结果区域:
 - 实时显示测试结果和详细信息
 - 显示测试汇总信息

4. 测试流程:

- 插入SD卡后自动检测
- 点击"开始测试"启动测试
- 测试过程中可随时停止
- 测试完成后自动生成报告

5. 配置说明:

- 循环测试:
 - 在config.yaml中设置enabled为true/false
 - 可设置循环次数(count)
- 性能测试参数:
 - total_size: 总测试数据大小(MB)
 - block_size: 单次读写块大小(MB)
 - iterations: 重复测试次数
- 界面设置:
 - always_on_top: 窗口是否置顶
- 日志设置:
 - level: 日志级别

2.4. CLI模式使用说明

1. 命令行运行:

```
./SDExpressTester.exe --cli --run # 使用配置文件运行测试  
./SDExpressTester.exe --cli --help # 显示帮助信息
```

2. 测试过程:

- 自动检测SD卡
- 显示测试进度
- 输出测试结果
- 生成测试报告

2.5. 配置文件说明

配置文件 `config.yaml` 包含以下主要设置(默认值), GUI模式和CLI模式都使用此配置文件

```
test:  
  # 循环测试配置  
  loop:  
    enabled: false # 是否启用循环测试  
    count: 1       # 循环次数  
  
  # 性能测试配置  
  performance:  
    total_size: 128 # 总数据大小(MB)  
    block_size: 1   # 块大小(MB)  
    iterations: 3   # 平均次数  
  
  # 界面配置  
  ui:  
    always_on_top: true # 窗口是否始终置顶  
  
  # 日志配置  
  logger:  
    level: INFO # 日志级别: DEBUG, INFO, WARNING, ERROR, CRITICAL
```

2.6. 测试报告说明

- 位置：程序运行目录下的 `test_report_YYYYMMDD_HHMMSS.txt`
- 内容：包含测试配置、测试结果汇总、详细测试数据

3. 开发者指南

3.1. 环境要求

- Python 3.8+
- PyQt5

3.2. 从源码运行

1. 克隆仓库：

```
git clone https://github.com/cursorhu/sd_express_tester.git
cd sd_express_tester
```

2. 安装依赖：

```
pip install -r requirements.txt
```

3. 运行程序：

```
GUI模式
python main.py
CLI模式
python main.py --cli --run
```

3.3. 打包说明

1. 安装PyInstaller：

```
pip install pyinstaller
```

2. 使用spec文件打包：

```
pyinstaller main.spec --clean
```

3.4. 项目结构

- `core/`：核心测试模块
- `gui/`：图形界面实现
- `cli/`：命令行实现
- `utils/`：工具类
- `main.py`：主程序入口

3.5. 技术要点

3.5.1. 技术栈

- GUI框架：PyQt5实现跨平台图形界面
- 系统接口：
 - WMI(Windows Management Instrumentation)：检测控制器和SD卡信息
 - Win32 API：底层文件读写操作
- 配置管理：YAML格式配置文件
- 日志系统：Python logging模块，支持多级别日志

3.5.2. 性能优化

1. 文件读写优化：

- 使用Win32 API直接操作文件：
 - CreateFile设置优化标志
 - ReadFile/WriteFile直接操作
 - 避免文件系统缓存造成误差
- 异步IO提升性能：
 - 使用FILE_FLAG_OVERLAPPED标志
 - 重叠IO操作并行处理
 - 使用完成端口(IOCP)处理异步结果
- 缓冲区优化：
 - 无缓冲写入(FILE_FLAG_NO_BUFFERING)
 - 直写模式(FILE_FLAG_WRITE_THROUGH)
 - 顺序扫描提示(FILE_FLAG_SEQUENTIAL_SCAN)

2. SD卡检测优化：

- 快速模式检测：
 - 根据1MB小数据读写速度判断卡类型
 - 仅当卡变化时才完整检测卡模式
- 轮询优化：
 - 使用WMI事件订阅
 - 异步处理设备变更通知
 - 减少轮询间隔(1秒)

3. 控制器检测优化：

- 缓存控制器信息，避免重复查询
- 异步检测，不阻塞UI

4. 内存管理：

- 大文件分块读写，避免内存溢出
- 及时释放不用的资源

5. UI响应优化：

- 使用QTimer延迟初始化
- 测试过程中通过信号机制更新UI

3.5.3. 可扩展设计

1. 测试框架：
 - 测试用例基类设计
 - 支持动态添加测试项
2. 报告生成：
 - 支持单次和循环测试报告
 - 结构化的报告格式
 - 详细的测试数据记录
3. 双模式支持：
 - GUI和CLI共用核心逻辑
 - 统一的配置管理
 - 一致的测试流程

3.5.4. 稳定性保障

1. 异常处理：
 - 全局异常捕获
 - 分级别的错误处理
 - 详细的错误日志
2. 状态管理：
 - 严格的状态检查
 - 测试过程可中断
 - 资源自动释放
3. 兼容性处理：
 - 支持多种SD卡规格
 - 控制器兼容性检查
 - 向下兼容传统SD模式

4. 许可证

GPLv3