

SD Express Test Tool

1. Introduction

SD Express Tester is a testing software for SD Express cards, with backward compatibility for SD 4.0, SD 3.0, SD2.0 and other traditional SD modes. The tool supports both graphical interface and command line operation modes, providing controller detection, basic read/write, performance testing and stability testing functions.

2. User Guide

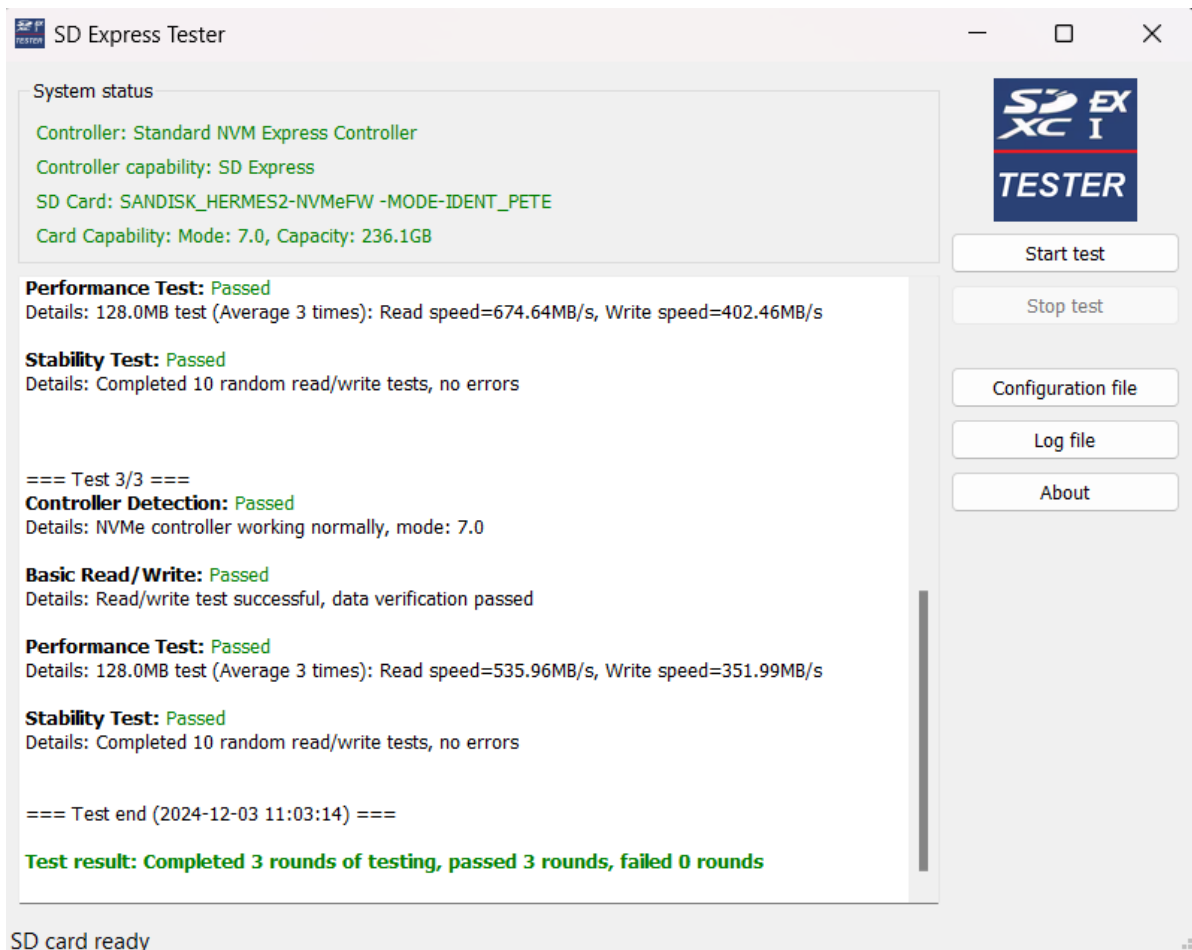
2.1. Features

- Dual operation modes: Graphical Interface (GUI) and Command Line (CLI)
- Automatic SD card and controller status detection
- Support multiple test items:
 - Controller compatibility detection
 - Basic read/write test (data comparison)
 - Performance test (read/write speed)
 - Stability test (random read/write)
- Support loop testing
- Real-time test progress display
- Automatic test report generation
- Detailed logging

2.2. System Requirements

- Windows 10/11
- SD Express card controller (O2Micro/BayHub series controllers)

2.3. GUI Mode Usage Instructions



1. Run `SDExpressTester.exe`

2. The program automatically detects SD cards and SD/NVMe controllers

3. Interface description:

- System status:
 - Controller: Display the current controller model
 - Controller capability: Display the current controller's supported SD card modes
 - Card name: Display the current detected SD card model
 - Card capability: Display the SD card's supported speed modes
- Control buttons:
 - Start test: Start executing the test suite
 - Stop test: Interrupt the current test process
 - Configuration file: Open the configuration file for editing
 - Log file: Open the log file to view detailed logs
 - About: Display software version and author information
- Progress display:
 - Progress bar: Display the completion progress of the current test item
 - Status bar: Display the status of the current test item
- Result area:
 - Real-time display of test results and detailed information
 - Display test summary information

4. Test process:

- Automatically detected after inserting the SD card
- Click "Start test" to start the test
- You can stop the test at any time during the test
- The test report is automatically generated after the test is complete

5. Configuration instructions:

- Loop test:
 - Set enabled to true/false in config.yaml
 - Set loop count (count)
- Performance test parameters:
 - total_size: Total test data size (MB)
 - block_size: Single read/write block size (MB)
 - iterations: Repeat test times
- Interface settings:
 - always_on_top: Whether the window is always on top
- Log settings:
 - level: Log level

2.4. CLI Mode Usage Instructions

1. Command line operation:

```
./SDExpressTester.exe --cli --run # Run test using config.yaml
./SDExpressTester.exe --cli --help # Display help information
```

2. Test process:

- Automatically detected SD card
- Display test progress
- Output test results
- Generate test report

2.5. Configuration File Description

The configuration file `config.yaml` contains the following main settings (default values), which are used by both GUI and CLI modes

```
test:
  # Loop test configuration
  loop:
    enabled: false # whether to enable loop test
    count: 1      # Loop count

  # Performance test configuration
  performance:
    total_size: 128 # Total data size (MB)
    block_size: 1  # Block size (MB)
    iterations: 3  # Average times

# Interface configuration
ui:
  always_on_top: true # whether the window is always on top

# Log configuration
logger:
  level: INFO # Log level: DEBUG, INFO, WARNING, ERROR, CRITICAL
```

2.6. Test Report Description

- Location: `test_report_YYYYMMDD_HHMMSS.txt` under the program running directory
- Content: Includes test configuration, test result summary, and detailed test data

3. Developer Guide

3.1. Environment Requirements

- Python 3.8+
- PyQt5

3.2. Running from Source Code

1. Clone the repository:

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

2. Install dependencies:

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

3. Run the program:

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

3.3. Packaging Instructions

1. Install PyInstaller:

```
pip install pyinstaller
```

2. Use spec file to package:

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

3.4. Project Structure

- `core/`: Core test module
- `gui/`: Graphical interface implementation
- `cli/`: Command line implementation
- `utils/`: Utility classes
- `main.py`: Main program entry

3.5. Technical Points

3.5.1. Technical Stack

- GUI framework: PyQt5 for cross-platform graphical interface
- System interface:
 - WMI (Windows Management Instrumentation): Detect controller and SD card information
 - Win32 API: Low-level file read/write operations
- Configuration management: YAML format configuration file
- Logging system: Python logging module, supporting multiple log levels

3.5.2. Performance Optimization

1. File read/write optimization:
 - Use Win32 API directly to operate files:
 - CreateFile setting optimization flags
 - ReadFile/WriteFile direct operations
 - Avoid errors caused by file system caching
 - Asynchronous IO to improve performance:
 - Use FILE_FLAG_OVERLAPPED flag
 - Overlapped IO operations are processed in parallel
 - Use completion ports (IOCP) to handle asynchronous results
 - Buffer optimization:
 - No buffer write (FILE_FLAG_NO_BUFFERING)
 - Direct write mode (FILE_FLAG_WRITE_THROUGH)
 - Sequential scan prompt (FILE_FLAG_SEQUENTIAL_SCAN)
2. SD card detection optimization:
 - Fast mode detection:
 - Determine card type based on 1MB small data read/write speed
 - Only detect card mode completely when the card changes
 - Polling optimization:
 - Use WMI event subscription
 - Asynchronous handling of device change notifications
 - Reduce polling interval (1 second)
3. Controller detection optimization:
 - Cache controller information to avoid repeated queries
 - Asynchronous detection, not blocking the UI
4. Memory management:
 - Large file read/write blocks to avoid memory overflow
 - Release unused resources in a timely manner
5. UI response optimization:
 - Use QTimer to delay initialization
 - Update UI through signal mechanisms during the test

3.5.3. Extensible Design

1. Test framework:
 - Design of test case base class
 - Support dynamic addition of test items
2. Report generation:
 - Support single and loop test reports
 - Structured report format
 - Detailed test data recording
3. Dual mode support:
 - GUI and CLI share core logic
 - Unified configuration management
 - Consistent test process

3.5.4. Stability Assurance

1. Exception handling:
 - Global exception capture
 - Error handling at different levels
 - Detailed error logs
2. State management:
 - Strict state check
 - Test process can be interrupted
 - Resources are automatically released
3. Compatibility handling:
 - Support multiple SD card specifications
 - Controller compatibility check
 - Backward compatibility with traditional SD modes

4. License

GPLv3