

# use Equip

## Equip 装备测试框架 — 使用手册

版本: 1.0.0

作者: mengfei.wu

日期: 2026-02-28

适用平台: Jetson AGX Orin / Orin NX

安装目录: `/opt/equip` (默认)

## 目录

- [1. 快速开始](#)
- [2. 全局选项](#)
- [3. 管理命令](#)
  - [3.1 list — 列出所有测试项](#)
  - [3.2 export — 导出能力清单](#)
  - [3.3 all — 执行全部测试](#)
  - [3.4 group — 执行测试组](#)
  - [3.5 plan — 执行测试计划](#)
  - [3.6 --clean — 清理日志与报告](#)
- [4. 测试命令与参数详解](#)
  - [4.1 ssd — SSD 磁盘读写测试](#)
  - [4.2 cpustress — CPU 压力测试](#)
  - [4.3 stress — CPU/GPU 满载压力测试](#)
  - [4.4 ethernet — 有线网口 iperf3 打流测试](#)
  - [4.5 netloop — 网口自环打流测试 \(1350\)](#)
  - [4.6 wifi — WIFI 打流测试](#)
  - [4.7 usb — USB 磁盘读写测试](#)
  - [4.8 can — CAN 回环测试](#)
  - [4.9 rs485 — RS485 回环测试](#)
  - [4.10 dido — DIDO 回环测试](#)
  - [4.11 rtc — RTC 时钟测试](#)
  - [4.12 mac — MAC 地址修改测试](#)
  - [4.13 camera\\_pressure — 摄像头压力测试](#)
  - [4.14 reboot — 系统重启稳定性测试](#)

- [4.15 powercycle — 掉电循环测试](#)
  - [5. 参数优先级机制](#)
  - [6. 配置文件说明](#)
  - [7. 输出与报告](#)
  - [8. 退出码说明](#)
- 

## 1. 快速开始

```
# 查看帮助
sudo equip --help

# 列出所有可用测试项
sudo equip list

# 执行单项测试（使用默认参数）
sudo equip ssd
sudo equip usb
sudo equip can

# 执行单项测试（指定参数）
sudo equip ssd --device /dev/nvme0n1p1 --tool fio --repeat 3

# 使用指定配置文件
sudo equip -c config/orin_config.json ssd

# 执行全部测试
sudo equip all

# 执行测试组
sudo equip group factory_flow

# 执行测试计划
sudo equip plan default
```

---

## 2. 全局选项

以下全局选项可在任何子命令前使用：

选项	缩写	默认值	说明
<code>--config</code> <code>CONFIG</code>	<code>-c</code>	<code>config/config.json</code>	指定配置文件路径
<code>--log-level</code> <code>LEVEL</code>	<code>-ll</code>	<code>INFO</code>	日志级别： <code>DEBUG</code> / <code>INFO</code> / <code>WARNING</code> / <code>ERROR</code>
<code>--verbose</code>	<code>-v</code>	关闭	启用详细日志输出（打印命令执行细节）
<code>--resume</code>	—	关闭	断点续跑模式：从上次中断处继续执行（适用于重启/掉电恢复）

**示例：**

```
# 使用 orin 专用配置 + DEBUG 级别日志
sudo equip -c config/orin_config.json -ll DEBUG ssd

# 断点续跑（重启后自动恢复上次进度）
sudo equip --resume ssd
```

### 3. 管理命令

#### 3.1 list — 列出所有测试项

```
sudo equip list
```

输出所有已注册的测试命令名称及功能描述，便于快速查阅可用测试项。

#### 3.2 export — 导出能力清单

```
sudo equip export
```

以 JSON 格式输出 Agent 完整能力清单，包含所有测试项的参数定义、默认值、数据类型等元信息。用于 Master 端动态渲染 UI 表单。

```
# 导出到文件
sudo equip export > manifest.json
```

#### 3.3 all — 执行全部测试

```
sudo equip all
```

按 `config.json` 中 `general_tests` 定义的所有测试项顺序执行，每项使用配置文件中的参数（无命令行覆盖）。

### 3.4 group — 执行测试组

```
sudo equip group <组名>
```

执行 `config.json` 中 `test_groups` 预定义的测试组。测试组支持并行/串行模式。

**示例：**

```
# 执行工厂流程组（cpustress + camera_pressure 并行）
```

```
sudo equip group factory_flow
```

```
# 执行稳定性流程组（ssd 串行）
```

```
sudo equip group stability_flow
```

**配置文件中的组定义：**

```
"test_groups": {
  "factory_flow": {
    "tests": ["cpustress", "camera_pressure"],
    "parallel": true
  },
  "stability_flow": {
    "tests": ["ssd"],
    "parallel": false
  }
}
```

### 3.5 plan — 执行测试计划

```
sudo equip plan <计划名>
```

执行 `config.json` 中 `test_plans` 预定义的测试计划。计划由多个测试组组成，按顺序依次执行。

**示例：**

```
# 执行默认计划（先 factory_flow 组，再 stability_flow 组）
sudo equip plan default
```

### 3.6 --clean — 清理日志与报告

```
# 清理所有（日志 + 报告）
sudo equip --clean

# 仅清理日志目录
sudo equip --clean --logs

# 仅清理报告目录
sudo equip --clean --reports

# 显式清理全部
sudo equip --clean --all
```

选项	缩写	说明
--clean	-cl	执行清理操作
--logs	-l	清理 logs/ 目录
--reports	-re	清理 reports/ 目录
--all	-a	清理所有生成文件（默认行为）

## 4. 测试命令与参数详解

**通用规则：**

- 所有参数均可选，不指定时使用默认值
- 命令行参数优先级最高，会覆盖配置文件和代码默认值
- 布尔参数使用 --参数名 启用（如 --bidir ），默认关闭
- 列表参数支持逗号分隔格式（如 --server\_ips 192.168.1.1,192.168.1.2 ）

### 4.1 ssd — SSD 磁盘读写测试

**功能：**对 NVMe SSD 进行读写性能测试，支持 fio（默认）和 dd 两种工具。fio 模式下读测试直接测试设备节点，写测试使用文件避免破坏文件系统。

```
sudo equip ssd
sudo equip ssd --device /dev/nvme0n1p1 --tool fio --repeat 3
sudo equip ssd --mode read --fio_bs 4k --fio_iodepth 32
```

参数	类型	默认值	说明
--device	str	/dev/nvme0n1	测试的磁盘设备节点路径
--tool	str	fio	测试工具: fio (推荐) 或 dd
--mode	str	rw	测试模式: read (只读) / write (只写) / rw (混合)
--write_threshold	float	200	写入速度合格阈值 (MB/s)
--read_threshold	float	100	读取速度合格阈值 (MB/s)
--size	str	1M	单次测试数据量 (仅 dd 模式)
--block_size	int	4096	数据块大小 (仅 dd 模式, 可选: 512/1024/2048/4096/8192)
--fio_runtime	int	10	fio 单次测试时长 (秒)
--fio_bs	str	128k	fio 块大小 (如 4k, 128k, 1M)
--fio_iodepth	int	16	fio IO 队列深度
--repeat	int	1	测试重复次数

## 4.2 cpustress — CPU 压力测试

**功能：**使用 stress 命令对 CPU 施加满载压力，验证系统稳定性。

```
sudo equip cpustress
sudo equip cpustress --timeout_seconds 120 --cpu_cores 12
```

参数	类型	默认值	说明
--cpu_cores	int	自动检测	压力测试使用的 CPU 核心数 (默认使用全部逻辑核心)
--timeout_seconds	int	60	压力测试持续时间 (秒)

### 4.3 stress — CPU/GPU 满载压力测试

**功能：**对 CPU 和 GPU 同时或分别进行满载压力测试。CPU 使用 `stress` 命令，GPU 使用 `gpu_burn` 工具。

```
sudo equip stress
sudo equip stress --duration 10 --test_gpu false
sudo equip stress --cpu_cores 12 --duration 30
```

参数	类型	默认值	说明
--test_cpu	bool	true	是否测试 CPU
--test_gpu	bool	true	是否测试 GPU
--cpu_cores	int	8	CPU 压力测试使用的核心数
--duration	int	5	测试持续时间（分钟）
--repeat	int	1	测试重复次数

### 4.4 ethernet — 有线网口 iperf3 打流测试

**功能：**使用 iperf3 对多个有线网口进行带宽打流测试，支持串行/并行模式和双向并发。需要远端 iperf3 服务端配合。

```
sudo equip ethernet
sudo equip ethernet --duration 120 --repeat 10 --min_rate 800
sudo equip ethernet --parallel --bidir
```

参数	类型	默认值	说明
--server_ips	list	192.168.138.200, 192.168.139.200, 192.168.140.200	服务端 IP 列表（逗号分隔）
--server_ports	list	8082, 8083, 8084	服务端端口列表（逗号分隔）
--client_ips	list	192.168.138.100, 192.168.139.100, 192.168.140.100	客户端本地 IP 列表（逗号分隔）
--duration	int	60	每次打流时长（秒）
--bandwidth	str	1000M	目标带宽

参数	类型	默认值	说明
<code>--parallel</code>	bool	false	并行模式开关（所有网口同时打流）
<code>--repeat</code>	int	60	测试重复次数
<code>--min_rate</code>	float	900.0	通过阈值（Mbits/sec）
<code>--bidir</code>	bool	false	iperf3 双向并发模式

**前置条件：**远端服务器需启动 iperf3 服务（`iperf3 -s -p 8082`），且本机 IP 已配置。

#### 4.5 netloop — 网口自环打流测试 (I350)

**功能：**Intel I350 四口千兆网卡物理自环测试。使用两根网线将四个接口两两互连，通过 Linux network namespace 隔离接收端口，使用 iperf3 进行打流。强制串行模式（因四口共享 PCIe x4 通道）。

```
sudo equip netloop
sudo equip netloop --duration 60 --min_rate 800 --repeat 5
```

参数	类型	默认值	说明
<code>--pair1_host_if</code>	str	enP1p1s0f0	第 1 组主机侧网口
<code>--pair1_ns_if</code>	str	enP1p1s0f1	第 1 组命名空间侧网口
<code>--pair1_host_ip</code>	str	192.168.10.10	第 1 组主机侧 IP
<code>--pair1_ns_ip</code>	str	192.168.10.11	第 1 组命名空间侧 IP
<code>--pair1_port</code>	int	5201	第 1 组 iperf3 端口
<code>--pair2_host_if</code>	str	enP1p1s0f2	第 2 组主机侧网口
<code>--pair2_ns_if</code>	str	enP1p1s0f3	第 2 组命名空间侧网口
<code>--pair2_host_ip</code>	str	192.168.20.20	第 2 组主机侧 IP
<code>--pair2_ns_ip</code>	str	192.168.20.21	第 2 组命名空间侧 IP
<code>--pair2_port</code>	int	5202	第 2 组 iperf3 端口
<code>--subnet_mask</code>	int	24	子网掩码长度
<code>--namespace</code>	str	ns_test	测试命名空间名称
<code>--duration</code>	int	30	每次 iperf3 打流时长（秒）
<code>--bandwidth</code>	str	1000M	iperf3 目标带宽



参数	类型	默认值	说明
--min_rate	float	900.0	单口最低通过带宽阈值 (Mbits/sec)
--repeat	int	1	测试重复次数
--bidir	bool	false	iperf3 双向并发开关
--netplan_config	str	/etc/netplan/01-network-manager-all.yaml	netplan 配置文件路径
--ping_count	int	3	连通性验证 ping 次数
--link_wait	int	3	链路协商等待时间 (秒)

**接线要求:** Port 0 ↔ Port 1 (一根网线) , Port 2 ↔ Port 3 (一根网线) 。

## 4.6 wifi — WIFI 打流测试

**功能:** 连接指定 WIFI 热点后，使用 iperf3 进行带宽打流测试，验证无线网络吞吐量是否达标。

```
sudo equip wifi
sudo equip wifi --ssid MyWifi --server_ip 192.168.1.100 --min_rate 100
```

参数	类型	默认值	说明
--server_ip	str	192.168.10.2	服务端 IP
--server_port	int	8085	服务端端口
--client_ip	str	192.168.10.3	客户端本地 IP
--duration	int	60	每次打流时长 (秒)
--bandwidth	str	1000M	目标带宽
--parallel	bool	false	并行模式开关
--repeat	int	60	测试重复次数
--min_rate	float	150.0	通过阈值 (Mbits/sec)
--ssid	str	CCCCDD111-5G	WIFI 名称 (SSID)
--password	str	空	WIFI 密码 (空表示开放网络)
--bidir	bool	false	iperf3 双向并发开关

**前置条件:** 远端服务器需启动 iperf3 服务，设备需具备 WIFI 网卡。

## 4.7 usb — USB 磁盘读写测试

**功能：**自动检测已插入的 USB 磁盘，使用 fio（默认）或 dd 进行读写性能测试，验证 USB 接口吞吐量。

```
sudo equip usb
sudo equip usb --ports 3 --tool fio --repeat 3
sudo equip usb --fio_bs 1M --fio_runtime 30
```

参数	类型	默认值	说明
--ports	int	5	待检测 USB 端口数量
--tool	str	fio	测试工具：fio（推荐）或 dd
--size	str	512M	单次测试数据量（仅 dd 模式）
--format_ext4	int	0	是否格式化为 ext4（仅 dd 模式，0=否，1=是）
--write_speed	str	60MB/s	写入速度合格阈值
--read_speed	str	100MB/s	读取速度合格阈值
--fio_runtime	int	10	fio 单次测试时长（秒）
--fio_bs	str	128k	fio 块大小
--fio_iodepth	int	16	fio IO 队列深度
--repeat	int	1	测试重复次数

## 4.8 can — CAN 回环测试

**功能：**CAN 总线回环测试，支持两组独立配对，支持 CAN FD 模式。将两个 CAN 接口物理互连，发送端发送数据后验证接收端是否正确收到。

```
sudo equip can
sudo equip can --pair1_tx can0 --pair1_rx can1 --fd_mode true
sudo equip can --repeat 10 --timeout 3
```

参数	类型	默认值	说明
--pair1_tx	str	can0	第 1 组发送端 CAN 接口
--pair1_rx	str	can1	第 1 组接收端 CAN 接口

参数	类型	默认值	说明
--pair1_bitrate	int	250000	第 1 组 CAN 波特率 (bps)
--pair1_dbitrate	int	1000000	第 1 组 CAN FD 数据段波特率 (bps)
--pair2_tx	str	can2	第 2 组发送端 CAN 接口
--pair2_rx	str	can3	第 2 组接收端 CAN 接口
--pair2_bitrate	int	250000	第 2 组 CAN 波特率 (bps)
--pair2_dbitrate	int	1000000	第 2 组 CAN FD 数据段波特率 (bps)
--timeout	int	5	CAN 消息接收超时时间 (秒)
--fd_mode	bool	true	是否启用 CAN FD 模式
--test_msg_id	int	0x123	测试消息 ID
--repeat	int	1	测试重复次数

**接线要求:** *can0 ↔ can1 (一根线)* , *can2 ↔ can3 (一根线)* , *H-H / L-L*。

## 4.9 rs485 — RS485 回环测试

**功能:** RS485 串口回环测试，支持三组独立配对。将 TX 和 RX 设备节点物理互连，发送测试数据后验证接收端是否一致。

```
sudo equip rs485
sudo equip rs485 --pair1_baudrate 9600 --test_data "HelloOrin"
sudo equip rs485 --repeat 5
```

参数	类型	默认值	说明
--pair1_tx	str	/dev/ttyTHS1	第 1 组发送端设备节点
--pair1_rx	str	/dev/ttyTHS2	第 1 组接收端设备节点
--pair1_baudrate	int	115200	第 1 组波特率 (bps)
--pair2_tx	str	/dev/ttyCH9344USB0	第 2 组发送端设备节点
--pair2_rx	str	/dev/ttyCH9344USB2	第 2 组接收端设备节点
--pair2_baudrate	int	115200	第 2 组波特率 (bps)
--pair3_tx	str	/dev/ttyCH9344USB1	第 3 组发送端设备节点
--pair3_rx	str	/dev/ttyCH9344USB3	第 3 组接收端设备节点
--pair3_baudrate	int	115200	第 3 组波特率 (bps)

参数	类型	默认值	说明
<code>--timeout</code>	int	5	串口读取超时时间（秒）
<code>--test_data</code>	str	Jetson_Orin_RS485_Test_Data	测试数据内容
<code>--repeat</code>	int	1	测试重复次数

### 4.10 dido — DIDO 回环测试

**功能：**通过 I2C 总线控制 DO（数字输出），读取 DI（数字输入）状态，验证 DIDO 回环是否正确。

```
sudo equip dido
sudo equip dido --bus 3 --address 0x22 --repeat 5
```

参数	类型	默认值	说明
<code>--bus</code>	int	2	I2C 总线号
<code>--address</code>	str	0x21	I2C 设备地址
<code>--config_reg</code>	str	0x06	配置寄存器地址
<code>--control_reg</code>	str	0x02	DO 控制寄存器地址
<code>--status_reg</code>	str	0x00	DI 状态寄存器地址
<code>--do_high</code>	str	0xF0	DO 高电平控制参数
<code>--do_low</code>	str	0x00	DO 低电平控制参数
<code>--config_value</code>	str	0x0F	DO 配置为输出、DI 配置为输入
<code>--expect_high</code>	str	0xF0	DO 高电平时 DI 预期输出
<code>--expect_low</code>	str	0x0F	DO 低电平时 DI 预期输出
<code>--timeout</code>	int	3	命令执行超时时间（秒）
<code>--repeat</code>	int	1	测试重复次数

### 4.11 rtc — RTC 时钟测试

**功能：**测试硬件 RTC 时钟的读写准确性。写入指定时间到 RTC，等待一段时间后回读，验证走时误差是否在允许范围内。

```
sudo equip rtc
sudo equip rtc --device /dev/rtc1 --tolerance 2.0 --sleep_duration 5
```

参数	类型	默认值	说明
--device	str	/dev/rtc0	RTC 设备路径
--tolerance	float	1.0	允许的读写误差范围（秒）
--sleep_duration	int	3	模拟走时的延时时间（秒）
--extern_time	str	空	外部指定的测试时间锚点（格式：YYYY-MM-DD HH:MM:SS，为空则使用当前系统时间）
--repeat	int	1	测试重复次数

### 4.12 mac — MAC 地址修改测试

**功能：**通过 ethtool 修改 Intel I350 四个网络设备的 MAC 地址到 EEPROM，系统重启后验证是否生效。此测试包含两个阶段：Phase 1 写入 → 请求重启 → Phase 2 验证。

```
sudo equip mac
sudo equip mac --enP1p1s0f0_mac AABBCDDEEF0
```

参数	类型	默认值	说明
--enP1p1s0f0_mac	str	10101A1A1A01	enP1p1s0f0 的目标 MAC 地址（12 位 16 进制，不含冒号）
--enP1p1s0f1_mac	str	10101A1A1A02	enP1p1s0f1 的目标 MAC 地址
--enP1p1s0f2_mac	str	10101A1A1A03	enP1p1s0f2 的目标 MAC 地址
--enP1p1s0f3_mac	str	10101A1A1A04	enP1p1s0f3 的目标 MAC 地址
--magic	str	0x15218086	ethtool EEPROM 写入魔术数（Intel 网卡）
--byte_write_delay	float	0.1	单字节写入后的等待时间（秒）
--interface_write_delay	float	1.0	单个接口写入完成后的等待时间（秒）
--repeat	int	1	测试重复次数

**注意：**此测试会请求系统重启（Exit Code 88），需配合 --resume 模式在重启后完成验证。

### 4.13 camera\_pressure — 摄像头压力测试

**功能：**自动扫描所有已连接的摄像头（ALY01–ALY06 及双目摄像头），进行长时间抓拍压力测试，检测 UVC 掉线、全绿/全黑异常图像等问题。

```
sudo equip camera_pressure
sudo equip camera_pressure --duration_hours 2 --interval 5 --save_images
```

参数	类型	默认值	说明
--duration_hours	float	1	测试总时长（小时）
--interval	int	1	每轮拍照间隔（秒）
--save_path	str	./report/camera_images	图片保存路径
--pass_rate	float	0.95	成功率阈值（0–1）
--save_images	bool	false	是否保存每轮图像
--parallel	bool	false	并行模式开关

### 4.14 reboot — 系统重启稳定性测试

**功能：**循环执行系统重启，验证每次重启后系统是否正常启动、文件系统是否完整。需配合 Master 端（equip\_master）或 --resume 模式使用。

```
sudo equip reboot
sudo equip reboot --uptime_threshold 180
```

参数	类型	默认值	说明
--uptime_threshold	int	300	判定重启成功的最大启动时间（秒），超过则告警
--check_file_persistence	bool	true	是否验证文件系统持久性（写入临时文件重启后检查）

**注意：**此测试会请求系统重启（Exit Code 88），循环执行 10 轮。

## 4.15 powercycle — 掉电循环测试

**功能：**通过外部继电器执行硬掉电循环测试，验证设备在反复断电上电后是否正常工作。需配合 Master 端控制继电器。

```
sudo equip powercycle
sudo equip powercycle --repeat 10 --uptime_threshold 180
```

参数	类型	默认值	说明
--repeat	int	5	掉电循环次数
--uptime_threshold	int	300	判定上电成功的最大启动时间（秒），超过则告警

**注意：**此测试会请求硬掉电（Exit Code 89），需 Master 端配合控制继电器。

## 5. 参数优先级机制

equip 框架采用 **三级参数合并** 机制，优先级从低到高：

L1（最低）	代码默认值	← Python dataclass 字段的 default 值
L2（中）	配置文件	← config.json 中 general_tests.<测试名>.params 的值
L3（最高）	命令行参数	← 执行时通过 --参数名 传入的值

**示例：**

```
# SSD 的 fio_runtime 默认值为 10（L1）
# 如果 config.json 中设置了 "fio_runtime": 30（L2），则使用 30
# 如果命令行指定了 --fio_runtime 60（L3），则最终使用 60
sudo equip ssd --fio_runtime 60
```

**只有用户显式传入的命令行参数才会覆盖，未传入的参数保持 L1/L2 层的值。**

## 6. 配置文件说明

默认配置文件位于 config/config.json，结构如下：

```
{
  "log_strategy": "per_group",
```

```

"log_level": "INFO",
"report_head": "Jetson AGX 单板测试",
"test_plans": {
  "default": ["factory_flow", "stability_flow"]
},
"test_groups": {
  "factory_flow": {
    "tests": ["cpustress", "camera_pressure"],
    "parallel": true
  }
},
"general_tests": {
  "ssd": {
    "name": "SSD测试",
    "params": { ... },
    "default_test_times": 1,
    "threshold": { "write_speed": "200MB/s", "read_speed": "100MB/s" },
    "pass_rate": 0.9,
    "is_selected": 1
  }
}
}

```

字段	说明
log_strategy	日志策略：per_group（按组分文件） / single（单文件）
log_level	全局日志级别
report_head	测试报告标题
test_plans	测试计划定义（由多个测试组组成）
test_groups	测试组定义（包含测试列表和并行/串行模式）
general_tests	各测试项的参数、阈值和选中状态

**Orin 专用配置：** config/orin\_config.json，可通过 -c 指定：

```
sudo equip -c config/orin_config.json ssd
```

## 7. 输出与报告

### 日志文件

测试过程中的日志输出到 logs/ 目录，按测试组或单项分别存放。



## HTML 报告

测试完成后自动生成 HTML 格式的测试报告，保存在 `reports/` 目录中：

```
reports/
└─ test_report_20260228_143025.html
```

报告包含每项测试的执行结果、详细参数、通过率等信息。

## 清理输出文件

```
sudo equip --clean          # 清理所有
sudo equip --clean --logs   # 仅清理日志
sudo equip --clean --reports # 仅清理报告
```

## 8. 退出码说明

退出码	含义	后续操作
0	正常退出（所有测试完成）	查看报告
1	异常退出（框架错误、参数错误等）	检查日志
88	请求软重启（ <code>reboot</code> / <code>mac</code> 测试触发）	Master 执行 <code>reboot</code> ，然后 <code>--resume</code> 恢复
89	请求硬掉电（ <code>powercycle</code> 测试触发）	Master 操作继电器断电再上电，然后 <code>--resume</code> 恢复

## 附录：常用组合命令速查

```
# ===== 基础测试 =====
sudo equip ssd                # SSD 默认测试
sudo equip usb --ports 3      # USB 3口测试
sudo equip cpustress --timeout_seconds 120 # CPU 压力 2分钟

# ===== 网络测试 =====
sudo equip ethernet --repeat 10 --min_rate 900 # 有线网口 10轮
sudo equip netloop --duration 60               # 自环打流 60秒
```

```

sudo equip wifi --ssid MyWifi --repeat 30      # WIFI 30轮

# ===== 总线测试 =====
sudo equip can --repeat 5                      # CAN 回环 5轮
sudo equip rs485 --repeat 5                    # RS485 回环 5轮
sudo equip dido --repeat 10                    # DIDO 回环 10轮

# ===== 系统级测试 =====
sudo equip stress --duration 30                # CPU+GPU 满载 30分钟
sudo equip rtc --tolerance 0.5                 # RTC 走时测试
sudo equip camera_pressure --duration_hours 4  # 摄像头 4小时压测

# ===== 批量执行 =====
sudo equip all                                # 全部测试
sudo equip group factory_flow                 # 工厂流程组
sudo equip plan default                       # 默认计划

# ===== 管理维护 =====
sudo equip list                               # 查看所有测试项
sudo equip export > manifest.json             # 导出能力清单
sudo equip --clean                             # 清理日志报告
sudo equip -c config/orin_config.json all     # 使用 Orin 配置执行

```

## 测试示例：SSD（其他测试项原理相同）

```

# 测试次数默认一次
cnit@cnit:~$ sudo equip ssd

```

正在启动测试项：SSD	
device:	/dev/nvme0n1p1
block_size:	4096
write_threshold:	200
read_threshold:	100
mode:	rw
tool:	fio
size:	2048M
repeat:	1
fio_runtime:	10
fio_bs:	128k
fio_iodepth:	16

```

2026-02-28 01:36:37 - INFO - [Main] ===== 开始测试 [ssd] =====

```

```
2026-02-28 01:36:37 - INFO - [Main] SSD设备节点检查通过: /dev/nvme0n1p1
2026-02-28 01:36:37 - INFO - [Main] [ssd] 执行第 1/1 次测试
2026-02-28 01:36:37 - INFO - [Main] 第1/1轮SSD测试开始
2026-02-28 01:36:37 - INFO - [Main] [SSD测试] 执行清空内存缓存命令: sudo echo 3 >
/proc/sys/vm/drop_caches
2026-02-28 01:36:37 - INFO - [Main] [SSD测试] 内存缓存清空成功
2026-02-28 01:36:37 - INFO - [Main] [SSD fio] 读测试命令: sudo fio --
allow_mounted_write=1 --ioengine=libaio --bs=128k --direct=1 --thread --rw=read --
filename=/dev/nvme0n1p1 --name="BS 128k read test" --iodepth=16 --runtime=10
2026-02-28 01:36:47 - INFO - [Main] [SSD fio] 读测试解析结果: 5032.0 MB/s
2026-02-28 01:36:47 - INFO - [Main] [SSD测试] 执行清空内存缓存命令: sudo echo 3 >
/proc/sys/vm/drop_caches
2026-02-28 01:36:47 - INFO - [Main] [SSD测试] 内存缓存清空成功
2026-02-28 01:36:47 - INFO - [Main] [SSD fio] 写测试命令: sudo fio --
name=seqwrite_128k_file --ioengine=libaio --iodepth=16 --rw=write --bs=128k --
direct=1 --size=5G --numjobs=1 --runtime=10 --time_based --group_reporting --
filename=test_128k.img
2026-02-28 01:36:57 - INFO - [Main] [SSD fio] 写测试解析结果: 3564.0 MB/s
2026-02-28 01:36:57 - INFO - [Main] [SSD fio] 已清理写测试中间文件: test_128k.img
2026-02-28 01:36:57 - INFO - [Main] 第1/1轮SSD测试完成 - 写速度: 3564.0MB/s, 读速度:
5032.0MB/s
2026-02-28 01:36:57 - INFO - [Main] [SSD测试] 测试结果: 总计 1 轮, 成功 1 轮 - 平均写
速度: 3564.00MB/s (阈值: 200.0MB/s), 平均读速度: 5032.00MB/s (阈值: 100.0MB/s) - 通过
2026-02-28 01:36:57 - INFO - [Main] ===== 测试 [ssd] 结束 (PASS) =====
2026-02-28 01:36:57 - INFO - [Main] 测试报告已生成:
/opt/equip/equip_test/reports/equip_test_report_20260228_013657.html
```

```
#测试次数指定3次
cnit@cnit:~$ sudo equip ssd --repeat 3
```

正在启动测试项: SSD	
device:	/dev/nvme0n1p1
block_size:	4096
write_threshold:	200
read_threshold:	100
mode:	rw
tool:	fio
size:	2048M
repeat:	3
fio_runtime:	10
fio_bs:	128k
fio_iodepth:	16

```
2026-02-28 01:47:09 - INFO - [Main] ===== 开始测试 [ssd] =====
2026-02-28 01:47:09 - INFO - [Main] SSD设备节点检查通过: /dev/nvme0n1p1
2026-02-28 01:47:09 - INFO - [Main] [ssd] 执行第 1/1 次测试
2026-02-28 01:47:09 - INFO - [Main] 第1/3轮SSD测试开始
2026-02-28 01:47:09 - INFO - [Main] [SSD测试] 执行清空内存缓存命令: sudo echo 3 >
/proc/sys/vm/drop_caches
2026-02-28 01:47:09 - INFO - [Main] [SSD测试] 内存缓存清空成功
2026-02-28 01:47:09 - INFO - [Main] [SSD fio] 读测试命令: sudo fio --
allow_mounted_write=1 --ioengine=libaio --bs=128k --direct=1 --thread --rw=read --
filename=/dev/nvme0n1p1 --name="BS 128k read test" --iodepth=16 --runtime=10
2026-02-28 01:47:20 - INFO - [Main] [SSD fio] 读测试解析结果: 4198.0 MB/s
2026-02-28 01:47:20 - INFO - [Main] [SSD测试] 执行清空内存缓存命令: sudo echo 3 >
/proc/sys/vm/drop_caches
2026-02-28 01:47:20 - INFO - [Main] [SSD测试] 内存缓存清空成功
2026-02-28 01:47:20 - INFO - [Main] [SSD fio] 写测试命令: sudo fio --
name=seqwrite_128k_file --ioengine=libaio --iodepth=16 --rw=write --bs=128k --
direct=1 --size=5G --numjobs=1 --runtime=10 --time_based --group_reporting --
filename=test_128k.img
2026-02-28 01:47:30 - INFO - [Main] [SSD fio] 写测试解析结果: 3575.0 MB/s
2026-02-28 01:47:30 - INFO - [Main] [SSD fio] 已清理写测试中间文件: test_128k.img
2026-02-28 01:47:30 - INFO - [Main] 第1/3轮SSD测试完成 - 写速度: 3575.0MB/s, 读速度:
4198.0MB/s
2026-02-28 01:47:30 - INFO - [Main] 第2/3轮SSD测试开始
2026-02-28 01:47:30 - INFO - [Main] [SSD测试] 执行清空内存缓存命令: sudo echo 3 >
/proc/sys/vm/drop_caches
2026-02-28 01:47:30 - INFO - [Main] [SSD测试] 内存缓存清空成功
2026-02-28 01:47:30 - INFO - [Main] [SSD fio] 读测试命令: sudo fio --
allow_mounted_write=1 --ioengine=libaio --bs=128k --direct=1 --thread --rw=read --
filename=/dev/nvme0n1p1 --name="BS 128k read test" --iodepth=16 --runtime=10
2026-02-28 01:47:40 - INFO - [Main] [SSD fio] 读测试解析结果: 5055.0 MB/s
2026-02-28 01:47:40 - INFO - [Main] [SSD测试] 执行清空内存缓存命令: sudo echo 3 >
/proc/sys/vm/drop_caches
2026-02-28 01:47:40 - INFO - [Main] [SSD测试] 内存缓存清空成功
2026-02-28 01:47:40 - INFO - [Main] [SSD fio] 写测试命令: sudo fio --
name=seqwrite_128k_file --ioengine=libaio --iodepth=16 --rw=write --bs=128k --
direct=1 --size=5G --numjobs=1 --runtime=10 --time_based --group_reporting --
filename=test_128k.img
2026-02-28 01:47:51 - INFO - [Main] [SSD fio] 写测试解析结果: 3575.0 MB/s
2026-02-28 01:47:51 - INFO - [Main] [SSD fio] 已清理写测试中间文件: test_128k.img
2026-02-28 01:47:51 - INFO - [Main] 第2/3轮SSD测试完成 - 写速度: 3575.0MB/s, 读速度:
5055.0MB/s
2026-02-28 01:47:51 - INFO - [Main] 第3/3轮SSD测试开始
2026-02-28 01:47:51 - INFO - [Main] [SSD测试] 执行清空内存缓存命令: sudo echo 3 >
/proc/sys/vm/drop_caches
```

```
2026-02-28 01:47:51 - INFO - [Main] [SSD测试] 内存缓存清空成功
2026-02-28 01:47:51 - INFO - [Main] [SSD fio] 读测试命令: sudo fio --
allow_mounted_write=1 --ioengine=libaio --bs=128k --direct=1 --thread --rw=read --
filename=/dev/nvme0n1p1 --name="BS 128k read test" --iodepth=16 --runtime=10
2026-02-28 01:48:01 - INFO - [Main] [SSD fio] 读测试解析结果: 4277.0 MB/s
2026-02-28 01:48:01 - INFO - [Main] [SSD测试] 执行清空内存缓存命令: sudo echo 3 >
/proc/sys/vm/drop_caches
2026-02-28 01:48:01 - INFO - [Main] [SSD测试] 内存缓存清空成功
2026-02-28 01:48:01 - INFO - [Main] [SSD fio] 写测试命令: sudo fio --
name=seqwrite_128k_file --ioengine=libaio --iodepth=16 --rw=write --bs=128k --
direct=1 --size=5G --numjobs=1 --runtime=10 --time_based --group_reporting --
filename=test_128k.img
2026-02-28 01:48:11 - INFO - [Main] [SSD fio] 写测试解析结果: 3585.0 MB/s
2026-02-28 01:48:11 - INFO - [Main] [SSD fio] 已清理写测试中间文件: test_128k.img
2026-02-28 01:48:11 - INFO - [Main] 第3/3轮SSD测试完成 - 写速度: 3585.0MB/s, 读速度:
4277.0MB/s
2026-02-28 01:48:11 - INFO - [Main] [SSD测试] 测试结果: 总计 3 轮, 成功 3 轮 - 平均写
速度: 3578.33MB/s (阈值: 200.0MB/s), 平均读速度: 4510.00MB/s (阈值: 100.0MB/s) - 通过
2026-02-28 01:48:11 - INFO - [Main] ===== 测试 [ssd] 结束 (PASS) =====
2026-02-28 01:48:11 - INFO - [Main] 测试报告已生成:
/opt/equip/equip_test/reports/equip_test_report_20260228_014811.html
```