



曦云[®] 系列通用 GPU

mx-smi 使用手册

OG-24049-020-F5_V14

2026-03-09

声明

版权所有 ©2024-2026 沐曦集成电路（上海）股份有限公司。保留所有权利。

本文档中呈现的信息属于沐曦集成电路（上海）股份有限公司和/或其附属公司（以下统称为“沐曦股份”），非经沐曦股份事先书面许可，任何实体或个人均不得获得本文档的副本，且无权以任何方式处理本文档，包括但不限于使用、复制、修改、合并、出版、发行、销售或传播本文档的部分或全部。

本文档内容仅供参考，不提供任何形式的、明示或暗示的保证，包括但不限于对适销性、适用于任何目的和/或不侵权的保证。在任何情况下，沐曦股份均不对因本文档引起的、由本文档造成的、或与之相关的任何索赔、损害或其他责任负责。

沐曦股份保留自行决定随时更改、修改、添加或删除本文档的部分或全部的权利。沐曦股份保留最终解释权。

沐曦、MetaX 和其他沐曦图标是沐曦股份的商标。本文档中提及的所有其他商标和商品名称均为其各自所有者的财产。

目录

1 概述	1
2 安装及使用方法	2
2.1 安装包安装与卸载	2
2.1.1 安装	2
2.1.2 卸载	2
2.2 在容器中使用	2
2.3 使用方法	2
2.4 命令补全使用方法	3
3 命令介绍	4
3.1 通用选项	4
3.2 归纳选项	4
3.3 查询选项	4
3.4 控制选项	8
3.5 RAS 选项	9
3.6 MXLK 选项	9
3.7 查询拓扑关系	10
3.8 VM 虚拟化管理	10
3.9 光模块信息（当前仅适用于 C500X）	11
3.10 轮询设备状态信息	11
3.11 MISC	11
3.12 ETH（仅适用于 C600 系列）	12
3.13 sGPU 切分选项	13
4 MXSML 使用说明	16
4.1 简介	16
4.2 安装	16
4.3 编程指南	16
5 Pymxsmi 使用说明	18
5.1 简介	18
5.2 安装	18
5.3 编程指南	18
6 附录	19
6.1 术语/缩略语	19

表目录

表 3.1	KMD 关键事件信息	6
表 3.2	创建子设备时资源设置项说明	14
表 3.3	子设备调度策略说明	15
表 4.1	MXSML 安装文件列表	16

更新记录

版本	日期	更新说明
V14	2026-03-09	更新以下章节： 3.11 MISC
V13	2026-01-30	更新以下章节： 3.3 查询选项 3.4 控制选项
V12	2025-12-24	更新以下章节： 3.3 查询选项 3.4 控制选项
V11	2025-11-27	更新以下章节： 3.3 查询选项 3.11 MISC
V10	2025-10-24	更新以下章节： 3.11 MISC
V09	2025-10-09	新增 C600 系列产品信息 更新以下章节： 3.2 归纳选项 3.11 MISC 新增以下章节： 3.12 ETH（仅适用于 C600 系列）
V08	2025-09-15	更新以下章节： 3.1 通用选项 3.3 查询选项 3.4 控制选项 3.5 RAS 选项 3.6 MXLK 选项 3.7 查询拓扑关系 3.13 sGPU 切分选项 新增以下章节： 3.11 MISC
V07	2025-08-01	更新以下章节： 3.2 归纳选项 3.3 查询选项 3.4 控制选项 3.13 sGPU 切分选项 新增以下章节： 3.6 MXLK 选项

下页继续

表 1 – 续上页

版本	日期	更新说明
V06	2025-06-25	更新以下章节： 2.2 在容器中使用 2.4 命令补全使用方法 3.3 查询选项 3.4 控制选项 3.7 查询拓扑关系 3.8 VM 虚拟化管理 3.13 sGPU 切分选项
V05	2025-04-21	更新以下章节： 3.3 查询选项 3.13 sGPU 切分选项
V04	2025-03-21	更新以下章节： 2.1.2 卸载 3.7 查询拓扑关系 4.2 安装 新增以下章节： 3.13 sGPU 切分选项
V03	2025-02-18	更新以下章节： 3.3 查询选项 3.7 查询拓扑关系
V02	2024-12-20	更新以下章节： 3.3 查询选项 4.2 安装 4.3 编程指南 新增以下章节： 2.4 命令补全使用方法
V01	2024-09-06	首次发布

1 概述

mx-smi (MetaX System Management Interface) 是用于管理曦云[®] 系列 GPU 的命令行工具，可以实时查询 GPU 的信息，包括版本，温度，使用率，功耗，时钟频率等，并具有启用和禁用 GPU 配置选项，执行固件升降级等功能。

2 安装及使用方法

2.1 安装包安装与卸载

2.1.1 安装

曦云系列 GPU 驱动包安装过程中会默认安装 mx-smi 工具。驱动包安装完成后，mx-smi 工具放置在 **/opt/mxdriver/bin/** 目录下，并会在 **/usr/bin** 目录下创建软链接。

2.1.2 卸载

曦云系列 GPU 的驱动卸载后，mx-smi 工具会自动卸载。驱动卸载参见《曦云® 系列通用 GPU 驱动安装指南》。

2.2 在容器中使用

若需在容器中使用 mx-smi，可在运行容器时传入以下参数：

```
docker run -v --device=/dev/dri -v /opt/mxdriver/bin/mx-smi:/opt/mxdriver/bin/mx-smi <IMAGE-ID>
```

其中 `--device=/dev/dri` 将主机上的沐曦设备挂载到容器，`-v /opt/mxdriver/bin/mx-smi:/opt/mxdriver/bin/mx-smi` 将 mx-smi 工具挂载到容器。

以上方式仅可在容器中用 `/opt/mxdriver/bin/mx-smi` 执行一些查询类的命令，如需执行设置类的命令，需在启动容器时添加 `--privileged=true`。

2.3 使用方法

mx-smi 是一个 Linux 命令行工具，其调用遵守如下格式。

```
mx-smi [选项1] [参数1]] [参数2] ...
```

2.4 命令补全使用方法

驱动包安装后，**/opt/mxdriver/share/mxsmi/mxsmi_completion.sh** 为 mx-smi 命令补全脚本，可以实现 mx-smi 命令自动补全。

需确保系统已安装 bash-completion 包，以 Ubuntu 为例，执行以下命令安装：

```
dpkg -i bash-completion
```

若为当前用户临时使用：

```
source /opt/mxdriver/share/mxsmi/mxsmi_completion.sh
```

若对所有用户生效（需 root 权限），复制脚本到 **/etc/bash_completion.d/** 目录并重新登录：

```
cp /opt/mxdriver/share/mxsmi/mxsmi_completion.sh /etc/bash_completion.d/
```

3 命令介绍

mx-smi 命令说明，可以通过在工作环境中执行 `man mx-smi` 获取。

3.1 通用选项

```
-h, --help
```

打印使用帮助信息。

```
-v, --version
```

打印 mx-smi 的版本信息。

```
-j, --json
```

以 JSON 格式显示板卡信息。

3.2 归纳选项

```
./mx-smi （直接回车）
```

mx-smi 命令不添加任何选项和参数，默认归纳并按照 Board 显示 GPU 的主要信息，显示信息包括功耗，内存，温度，版本，使用率，GPU 持久模式状态，sGPU 模式状态及进程等。

若开启 sGPU 且已有新建的 sGPU，则还会显示 sGPU 信息列表。

3.3 查询选项

```
--show-temperature
```

显示板卡上温度传感器读取的温度，单位是摄氏度。

```
--show-version
```

显示 BIOS，驱动和 Firmware 的版本信息。

```
--show-hbm-bandwidth
```

显示显存的动态带宽信息。

```
--show-pcie-bandwidth
```

显示 PCIe 动态带宽信息。

```
--show-usage
```

显示 GPU、VPU 的使用率。使用率反应过去一段采样周期内硬件资源的使用情况。C588 支持使用 `--die` 进行分 die 显示。

```
--show-memory
```

显示存储资源的使用情况，包括板卡显存和板卡可以访问的系统内存。C588 支持使用 `--die` 进行分 die 显示。驱动与固件会预留使用小于 1GB 的显存，所以在无进程时也会看到一部分显存已经被占用。

```
--show-board-power
```

显示板卡当前的电压，电流和功耗信息。

```
--show-pmbus-power
```

显示芯片内的电压，电流和功耗信息。C588 支持使用 `--die` 进行分 die 显示。

```
--show-EEPROM
```

显示 EEPROM 记录的信息，包括板卡版本，板卡序列号等。

```
--show-clock
```

显示板卡常用 IP（如 XCORE，MC 等）的时钟信息。C588 支持使用 `--die` 进行分 die 显示。

```
--show-clocks all
```

显示板卡的时钟信息。C588 支持使用 `--die` 进行分 die 显示。

```
--show-dpm cur
```

显示当前的性能等级。C588 支持使用 `--die` 进行分 die 显示。

```
--show-dpm all
```

显示支持的性能等级及其对应的时钟和电压信息，为设置 performance level 做参考。

```
--show-dpm-max
```

显示支持的性能等级的最高级别。

```
--show-pcie
```

显示 PCIe 当前及最大带宽和速率。

```
--show-sn
```

显示整板序列号和芯片序列号。

```
--show-process
```

显示当前运行的进程信息。C588 支持使用 `--die` 进行分 die 显示。

```
--show-power-mode
```

显示板卡的电源模式，分为普通模式和高功耗模式（增加电压），未用 `-i` 指定板卡时，默认显示所有板卡的电源模式信息。

```
--show-persistence-mode
```

显示板卡的持久模式，可使板卡即使在没有活动进程调用时，也保持在高性能状态，而不会自动切换到低功耗模式。未用 `-i` 指定板卡时，默认为 Disable。

```
--show-event {aer_ue|aer_ce|synfld|dbe|mmio|all}
```

显示驱动关键事件信息，可指定显示相应 pci 错误类型或全部类型的关键事件信息，包括大小，次数，第一次发生时间，事件内容等。未用 `-i` 指定板卡时，默认显示所有板卡的关键事件信息。详细信息参见表 3.1。

表 3.1 KMD 关键事件信息

事件类型	说明
aer_ue	PCIe AER uncorrectable error，仅 PF 模式支持查询。
aer_ce	PCIe AER correctable error，仅 PF 模式支持查询。
synfld	PCIe syncflood error，仅 PF 模式支持查询。
dbe	PCI device base error，仅 PF 模式支持查询。
mmio	PCI mmio invalid (disconnected)，PF 与 VF 模式均支持查询。

```
--show-clk-tr
```

显示板卡降频原因信息。Active 表示对应项为降频原因；全为 Not Active 表示当前未降频。未用 `-i` 指定板卡时，默认显示所有板卡的降频原因信息。

```
--show-unavailable-reason
```

当使用 `mx-smi -L` 命令查看设备状态为 `not available` 时，可用以上命令查看具体不可用原因以及参考建议。

```
--show-hwinfo
```

显示设备硬件汇总信息，包括产品名称，芯片系列，厂商信息，板卡 SN，板卡 PN，芯片序列号，Memory 类型及容量。

```
--show-sysinfo
```

显示设备系统汇总信息，包括驱动 GPU id，Node id，Render id 和 Card id。

```
--count-ecc
```

显示设备 ECC 错误及坏页的计数信息。ECC 错误包括 SRAM 及 DRAM 的可纠正和不可纠正的 ECC 计数，数值会在重启后清零。坏页计数信息重启不清零。

```
-i <GPU-ID>, --index <GPU-ID>
```

显示指定板卡的信息。如果没有指定，默认显示全部板卡的信息。GPU-ID 是从 0 开始的自然数，可以通过 -L，--list 获取板卡的 ID 信息。可指定一个或多个 GPU-ID，多个 GPU-ID 用 “,” 分隔，如 “0,1,5”，“0-2”，“0-4,6”，“all” 等。

```
-o, fileName, --output fileName
```

可以将命令输出写入 csv 格式的文件中，支持显示的命令有：

- --show-temperature
- --show-board-power
- --show-version
- --show-usage
- --show-memory
- --show-pcie
- --show-dpm cur
- --show-pmbus-power
- --show-vpu

```
-l ms, --loop ms
```

以特定的间隔周期持续显示板卡信息，单位是毫秒。执行 Ctrl+C 可退出持续查询。

示例：

以 500ms 周期持续显示所有卡的温度，功耗，使用率信息：

```
mx-smi --show-temperature --show-board-power --show-usage -l 500
```

以 1s 周期将 GPU0 - GPU3 的温度，功耗，使用率信息写文件：

```
mx-smi --show-temperature --show-board-power --show-usage -i 0-3 -l 1000 -o sample.  
↪ CSV
```

3.4 控制选项

```
-u, --vbios-upgrade <vbios_bin_file>
```

默认对所有板卡进行 BIOS 固件升级。需要 root 权限。当 MMio 地址空间不足时，会有相应提示，用户可选择继续升级（Y）或者不升级（N）。若继续升级有可能导致系统无法启动。

若当前 Bar 空间与要升级的 BIOS 需要的 Bar 空间不一致时，会分别列出需要的 Bar 空间。

需要对系统进行 reboot 或者 warm reset 以使新固件生效。

```
-r, --reset
```

对指定的卡执行 warm reset 重置。必须使用 `-i <GPU-ID>`，`--index <GPU-ID>` 选项以指定一个特定的板卡。

需要 root 权限。如果 warm reset 失败（设备无法恢复），需要重启服务器，对同一组使用 MetaXLink 连接的卡，只需做一次 warm reset，详情参见《曦云® 系列通用 GPU Warm Reset 使用指南》。

注解： warm reset 功能目前仅在 H3C UniServer R5300 G6 服务器型号中经过验证，在未经验证的机型中使用 warm reset 功能可能会导致功能失效等不可预期的结果。

```
--set-power-mode {0|1}
```

用 `-i` 指定板卡进行电源模式设置，0 为 Normal，1 为 High。C588 支持使用 `--die` 进行分 die 设置。

```
--set-persistence-mode {0|1}
```

用 `-i` 指定板卡进行 GPU 持久模式设置，0 为关闭，1 为打开。默认为关闭，重启后恢复默认值（关闭）。

```
--set-dpm-max <IP,DpmLevel>
```

对指定的 IP 设置最大性能等级。必须使用 `-i <GPU-ID>`，`--index <GPU-ID>` 选项以指定一个特定的板卡。需要 root 权限。在设置前可先使用 `--show-dpm-max` 查看有哪些可设置的 IP。

例如，执行以下命令，将板卡 1 的 xcore 模块的最大性能等级有效配置设置成 level 7：

```
mx-smi --set-dpm-max xcore,7 -i 1
```

3.5 RAS 选项

mx-smi ras 子命令用于显示与芯片可靠性，可用性，安全性相关的信息。

可显示 IP 的错误统计。例如，执行以下命令，显示板卡 0 相关的 corrected error 和 uncorrected error 统计数目：

```
ras --show-count -i 0
```

可显示 IP 的错误状态。例如，执行以下命令，显示板卡 0 相关的错误状态：

```
ras --show-status -i 0
```

3.6 MXLK 选项

mx-smi mxlk 子命令用于显示与 MetaXLink 相关的信息。

```
--show
```

显示曦云 GPU 设备之间 MetaXLink 带宽和速率。

```
--show-peer
```

显示通过 MetaXLink 与曦云 GPU 设备相连的设备信息。

```
--show-state
```

显示曦云 GPU 设备之间 MetaXLink 连接的状态（启用/禁用）和各端口的状态（**up/down**）。

```
--set-state {0|1}
```

启用/禁用 MetaXLink 连接，用于修复因训练失败等原因导致端口状态为 **down** 的情况。操作步骤为先禁用（0）再启用（1）。

```
--show-aer
```

显示 MetaXLink 的错误统计。

```
--show-traffic-stat
```

显示 MetaXLink port 的 TX（发送数据）与 RX（接收数据）字节总数。

3.7 查询拓扑关系

mx-smi topo 子命令用于查询主机上曦云 GPU 设备之间的拓扑关系。

```
-m, --matrix
```

显示曦云 GPU 设备之间的连接方式，以及 CPU 亲和性、NUMA node 亲和性等信息。

```
-d, --distance
```

显示曦云 GPU 设备之间连接的相对距离信息。

```
-t, --tree
```

显示曦云 GPU 设备拓扑树。

```
-n, --nic
```

显示曦云 GPU 设备与网卡之间的连接方式。

注解：MetaXLink 相关命令已移动到 mxlk 子命令中，请参见 [3.6 MXLK 选项](#)。

3.8 VM 虚拟化管理

mx-smi vm 子命令用于 GPU 卡虚拟化管理。

注解：此命令仅支持在物理机中操作。

```
--enable-vf {1|2|4|8}
```

开启虚拟化，一张 GPU 卡可支持虚拟化的个数为 1，2，4，8。

```
--disable-vf
```

关闭虚拟化功能。

```
--show-vf
```

显示虚拟化后的 GPU 信息。

3.9 光模块信息（当前仅适用于 C500X）

mx-smi om 子命令用于显示光模块信息。

```
--show-status
```

显示光模块电压，温度，模块状态，通道状态及接收状态信息。

```
--show-info
```

显示光模块固件信息。

```
--show-rx-status
```

显示光模块接收的详细信息。

3.10 轮询设备状态信息

mx-smi dmon 子命令用于轮询显示设备状态信息，包括温度，功耗，VPUE，VPUD 和 GPU 使用率，板卡显存及内存；时间间隔默认为 1000ms。执行 Ctrl+C 可退出轮询。

例如，执行以下命令，可轮询 100 次板卡 0 的温度及功耗信息：

```
mx-smi dmon --show-temperature --show-board-power -i 0 -c 100
```

3.11 MISC

mx-smi misc 子命令用于显示多种类的信息，如严重事件信息及日志级别等。

```
--show-board-type
```

显示板卡类型。

```
--show-critical-event
```

显示 PCIe 及 MetaXLink 严重错误信息。若有，会显示类型、连接状态、发生严重错误的原因和时间。

```
--show-mmio-state
```

显示 PCI MMIO 状态；当状态异常时显示第一次发生异常的时间。

```
--show-pcie-event
```

显示 PCIe 计数不为 0 的历史事件，需使用 root 权限查看。如需显示所有 PCIe 事件，可加 --detail。

```
--clear-pcie-event -i <device id>
```

清除 PCIe 历史事件，需指定 device id 并使用 root 权限操作。

```
--show-fw-loglevel
```

显示 Firmware (C500 系列 ip: smp0, smp1, ccx0, ccx1, ccx2, xcore; C600 系列 ip: smp0, smp1, xcore, eth) 的日志级别。

```
--set-fw-loglevel <ip>,<loglevel_number>
```

设置 Firmware (C500 系列 ip: smp0, smp1, ccx0, ccx1, ccx2, xcore; C600 系列 ip: smp0, smp1, xcore, eth) 的日志级别 (5:debug, 4:info, 3:warn, 2:error, 1:fatal, 0:none)。

```
--show-ecc-state
```

显示设备显存 ECC 模式，默认为 Enable。

```
--collect-chip-sn
```

收集服务器上的芯片 SN 并存入当前执行路径下的 **metax_chip_sns.txt** 文件中。

```
--show-op-mode
```

显示板卡的运维模式，默认为 Normal。

```
--set-op-mode
```

设置板卡的运维模式，1 为维护模式，0 为正常模式，进入维护模式后，不可运行程序。

3.12 ETH（仅适用于 C600 系列）

mx-smi eth 子命令用于显示 ETH 相关信息。

```
--show-bandwidth
```

显示 ETH 动态带宽信息。

```
--show-usage
```

显示 ETH 使用率。

```
--show-mac-addr
```

显示 ETH Mac 信息。

```
--show-ras-count
```

显示 ETH IP 的 corrected error 和 uncorrected error 统计数目。

```
--show-ras-status
```

显示 ETH IP 的错误状态。

```
--show-clock
```

显示 ETH 的时钟信息（需 root 权限）。

```
--show-status
```

显示 ETH 使能状态（需 root 权限）。

3.13 sGPU 切分选项

为了资源粒度可以灵活配置，sGPU（sliced GPU）功能可实现将一个父设备（物理设备）切分为多个子设备（sGPU），每个父设备最多可创建 16 个子设备（若按照默认配置显存 4GB，算力为父设备的 5% 创建，则由于显存大小的限制，无法创建出第 16 个子设备）。

注解：

- 虚拟化模式不支持此功能。
- 仅 C500 支持此功能。

```
sgpu -h
```

查看 sGPU 使用帮助信息。

```
sgpu --enable -i <GPU-ID>
```

启用设备 sGPU 功能，须用 -i 指定设备（需 root 权限）。默认为禁用。

```
sgpu --show-mode
```

查看设备 sGPU 功能是否启用，可用 -i 指定设备。

```
sgpu --disable
```

禁用设备 sGPU 功能，须用 -i 指定设备（需 root 权限），当子设备正在被使用时，则无法禁用 sGPU 功能。

```
sgpu -i <GPU-ID> --create
```

新建指定父设备的子设备，须用 -i 指定父设备。默认创建一个显存为 4 GB，算力为父设备 5% 的子设备，不创建别名。若需创建多个可用 -n 传入数量；可用 --vram xxx --compute xxx --alias xxx 指定创建的子设备的显存大小、算力占比、别名等。具体资源设置方法参见表 3.2。

表 3.2 创建子设备时资源设置项说明

可设置项	说明
--vram	分配给子设备的显存，粒度为 1M。例如，可设置为 128M、8G 或 5%
--compute	分配给子设备的父设备算力百分比，粒度为 1%，单位为%，例如，可设置为 22
--alias	分配给子设备的别名，最大长度为 31 字节，超出部分会被截断

```
sgpu -i <GPU-ID> --set <SGPU-ID> --vram <vram_size> --compute {0-100} --alias
↪<name>
```

修改已建子设备的显存配额，算力配额及别名。

```
sgpu
```

可查看所有父设备下的子设备信息，包括父设备编号、时间片大小、调度策略、子设备 SGPU-ID、MinorID、优先级、显存配额、算力配额、子设备调度策略以及别名等。可用 -i 指定父设备查看该父设备下所有子设备的信息。

```
sgpu --show-remain
```

查看父设备剩余可分配额度，可用 -i 指定父设备。

```
sgpu --show-usage
```

查看子设备使用率，可用 -i 仅指定父设备或用 -i 及 -s 指定父设备及子设备。

```
sgpu --show-memory
```

查看子设备显存使用情况，可用 -i 仅指定父设备或用 -i 及 -s 指定父设备及子设备。

```
sgpu --show-latency
```

查看子设备调度延时统计信息，可用 -i 指定父设备。

```
sgpu --show-timeslice
```

查看父设备时间片大小，可用 -i 指定父设备。默认为 10ms。

```
sgpu --set-timeslice xx -i <GPU-ID>
```

设置父设备时间片大小，单位为 ms，须用 -i 指定父设备。

```
sgpu --show-schedclass
```

查看调度策略，可用 -i 指定父设备。

表 3.3 子设备调度策略说明

调度类型	说明
0: (Best Effort) 争抢模式	默认调度策略。各子设备不限算力，父设备上有余算力即可使用
1: (Fixed Share) 固定配额	各子设备有固定的算力配额，无法超过固定配额使用
2: (Burst Share) 保证配额加弹性	各子设备有固定的算力配额，父设备上还有空闲算力可被使用

```
sgpu --set-sched-class {0|1|2} -i <GPU-ID>
```

设置调度策略（需 root 权限），须用 -i 指定设备。

```
sgpu --remove <SGPU-ID> -i <GPU-ID>
```

移除指定父设备的指定子设备（需 root 权限），须用 -i 指定设备。当子设备正在被使用时，则无法关闭 sGPU 功能。

```
sgpu --set-queue-priority [0|1] -s <SGPU-ID> -i <GPU-ID>
```

设置 sGPU 队列优先级（需 root 权限），须用 -i 指定父设备，-s 指定子设备。0 表示 Normal，1 表示 High，修改后可用 `mx-smi sgpu` 查看 Priority 列。

4 MXSML 使用说明

4.1 简介

MXSML (MetaX System Management Library) 是一套 C API 程序开发库。应用程序使用这套开发库，可以方便地对沐曦的设备进行状态查询和管理。MXSML 提供的 API 可以被使用 C/C++/PYTHON/GO 等语言开发的程序使用。

4.2 安装

MXSML 动态链接库是随 Driver 软件包一起发布，同时在 MXMACA SDK 包中会包含头文件，stub 库，示例代码，Python bindings 等文件。

Driver 软件包的安装参见《曦云® 系列通用 GPU 驱动安装指南》中“安装驱动”章节。Driver 软件包安装后，MXSML 动态链接库位于 `/opt/mxdriver/lib/libmxsml.so`。

MXMACA SDK 包的安装参见《曦云® 系列通用 GPU 快速上手指南》。SDK 安装后的文件列表参见表 4.1。

表 4.1 MXSML 安装文件列表

文件	说明
<code>/opt/maca/include/mxsml/MxSml.h</code>	MXSML 头文件
<code>/opt/maca/include/mxsml/MxSmlExtension.h</code>	MXSML 扩展头文件
<code>/opt/maca/include/mxsml/mxsmlBindings.py</code>	MXSML Python bindings
<code>/opt/maca/lib/stub/libmxsml.so</code>	MXSML stub 库
<code>/opt/maca/share/mxsml/pymxsml-xxx.whl</code>	Pymxsml 安装包
<code>/opt/maca/share/mxsml/mxsmlDemo.c</code>	C 示例程序
<code>/opt/maca/share/mxsml/mxsml_demo.py</code>	Python 示例程序
<code>/opt/maca/share/mxsml/mxsmlDemo.go</code>	Go 示例程序

4.3 编程指南

用 C 语言调用 MXSML API 读取所有卡的信息，示例如下：

```
#include <stdio.h>
#include "MxSml.h"

int main()
{
```

(下页继续)

(续上页)

```
// 调用任何MXSML API前都需要先调用mxSmlInit完成初始化
mxSmlReturn_t ret = mxSmlInit();

if (ret != MXSML_Success)
{
    // 可通过mxSmlGetErrorString将错误号转换为错误信息
    printf("mxSmlInit failed: %u, %s\n", ret, mxSmlGetErrorString(ret));

    return 1;
}

// 获取可用设备总数，设备ID从0开始依次编号：0,1,2...deviceNum-1
int deviceNum = mxSmlGetDeviceCount();
for (int i = 0; i < deviceNum; i++)
{
    mxSmlDeviceInfo_t deviceInfo;
    if (mxSmlGetDeviceInfo(i, &deviceInfo) == MXSML_Success)
    {
        printf("device id %d, bdf id: %s, type: %s\n", i, deviceInfo.bdfId,
↪deviceInfo.deviceName);
    }
}

return 0;
}
```

使用以下命令完成编译：

```
gcc example.c -I/opt/maca/include/mxsml -L/opt/maca/lib/stub -lmxsml -o example
```

运行 example：

```
export LD_LIBRARY_PATH=/opt/mxdriver/lib:$LD_LIBRARY_PATH
./example
```

5 Pymxsmi 使用说明

5.1 简介

Pymxsmi 是基于 MXSML 封装的一套 Python API 程序开发库。使用这套开发库，可以方便地对沐曦 GPU 设备进行状态查询和管理。

5.2 安装

操作步骤

1. Pymxsmi 依赖 MXSML 动态链接库，确保驱动包已安装，相关信息参见 [4.2 安装](#)。
2. Pymxsmi 以 wheel 包形式随 MXMACA SDK 包一起发布。MXMACA SDK 包安装后，Pymxsmi wheel 包位于 **/opt/maca/share/mxsml/pymxsmi-*.whl**。
3. 确保系统已安装 Python 3，使用以下命令安装 Pymxsmi wheel 包。

```
pip3 install /opt/maca/share/mxsml/pymxsmi-*.whl
```

5.3 编程指南

用 Python 语言导入 Pymxsmi 模块读取所有卡的信息，示例如下：

```
from pymxsmi import *

# 调用任何 Pymxsmi API 前都需要先调用 mxSmlInit 完成初始化
# 接口调用失败，会产生对应的 python 异常
mxSmlInit()

# 获取可用设备总数，设备 ID 从 0 开始依次编号：0, 1, 2...deviceNum-1
device_num = mxSmlGetDeviceCount()
for i in range(device_num):
    device_info = mxSmlGetDeviceInfo(i)
    print("device id", i, ", bdf id:", device_info.bdfId, ", type:", device_info.
↪deviceName)
```

运行 example：

```
python3 example.py
```

6 附录

6.1 术语/缩略语

术语/缩略语	全称	描述
DPM	Dynamic Power Management	动态电源管理
ECC	Error Checking and Correcting	错误检查与纠正
EEPROM	Electrically Erasable Programmable Read-Only Memory	带电可擦可编程只读存储器
MMIO	Memory-Mapped Input/Output	内存映射输入/输出
RAS	Reliability, Availability, Serviceability	
VM	Virtual Management	虚拟化管理