

Atlas 300I 推理卡

# 带外管理接口说明 (型号 3000, 3010)

文档版本 09  
发布日期 2022-07-29



版权所有 © 华为技术有限公司 2022。保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

## 商标声明



HUAWEI和其他华为商标均为华为技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

## 注意

您购买的产品、服务或特性等应受华为公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为公司对本文档内容不做任何明示或暗示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

## 华为技术有限公司

地址： 深圳市龙岗区坂田华为总部办公楼 邮编： 518129

网址： <https://e.huawei.com>

# 前言

## 概述

本文档详细的描述了Atlas 300I 推理卡MCU对外接口使用方式以及接口定义。






## 适用范围

本文档适用于以下产品型号，如型号内容一致，则不作区分，名称统称为Atlas 300I 推理卡。

- Atlas 300I 推理卡（型号 3000）
- Atlas 300I 推理卡（型号 3010）

## 符号约定

在本文中可能出现下列标志，它们所代表的含义如下。

符号	说明
	表示如不避免则将会导致死亡或严重伤害的具有高等级风险的危害。
	表示如不避免则可能导致死亡或严重伤害的具有中等级风险的危害。
	表示如不避免则可能导致轻微或中度伤害的具有低等级风险的危害。
	用于传递设备或环境安全警示信息。如不避免则可能会导致设备损坏、数据丢失、设备性能降低或其它不可预知的结果。 “须知”不涉及人身伤害。
	对正文中重点信息的补充说明。 “说明”不是安全警示信息，不涉及人身、设备及环境伤害信息。

# 修改记录

文档版本	发布日期	修改说明
09	2022-07-26	第九次正式发布。 修改3.21 0x001A 查询升级状态、3.31 0x0602 查询复位D芯片结果。
08	2022-02-14	第八次正式发布。 修改5.1 传感器名称约定。
07	2022-01-06	第七次正式发布。 修改2.2 Read Request格式和Write Request格式、2.3 Read Response格式和Write Response格式。
06	2021-11-22	第六次正式发布。 修改5.1 传感器名称约定。
05	2021-11-01	第五次正式发布。 修改3.17 0x0016 启动/停止装备测试、3.32 0x0605 获取Davinci信息、5.3 故障定义。
04	2021-08-24	第四次正式发布。 删除0x0008 查询Serial Number接口。
03	2021-04-29	第三次正式发布。

文档版本	发布日期	修改说明
02	2021-01-25	<p>第二次正式发布。</p> <p>新增：</p> <p>3.1 0x0000 查询设备能力。</p> <p>3.4 0x0003 查询芯片温度。</p> <p>3.5 0x0004 查询设备功耗。</p> <p>3.6 0x0005 查询MCU固件版本。</p> <p>3.7 0x0006 查询Vendor ID。</p> <p>3.8 0x0007 查询Device ID。</p> <p>0x0008 查询Serial Number。</p> <p>3.9 0x0009 查询Sub Vendor ID。</p> <p>3.10 0x000A 查询Sub Device ID。</p> <p>3.11 0x000B 查询芯片电压。</p> <p>3.12 0x000C 获取错误日志。</p> <p>3.13 0x000F 查询Board ID。</p> <p>3.14 0x0010 查询PCB ID。</p> <p>3.15 0x0011 查询MCU复位时间。</p> <p>3.16 0x0015 查询电子标签。</p> <p>3.17 0x0016 启动/停止装备测试。</p> <p>3.18 0x0017 查询装备测试结果。</p> <p>3.19 0x0018 传输MCU固件文件。</p> <p>3.20 0x0019 升级控制命令。</p> <p>3.21 0x001A 查询升级状态。</p> <p>3.22 0x001C 查询标卡设备电压。</p> <p>3.24 0x001F 查询ECC统计信息。</p> <p>3.25 0x0020 查询系统时间。</p> <p>3.26 0x0021 设置系统时间。</p> <p>3.27 0x0026 获取操作日志。</p> <p>3.28 0x0027 获取维护日志。</p> <p>3.29 0x0028 查询BOM ID。</p> <p>3.30 0x0601 复位D芯片。</p> <p>3.31 0x0602 查询复位D芯片结果。</p> <p>3.32 0x0605 获取Davinci信息。</p> <p>3.33 0x0614 查询MCU心跳计数。</p> <p>3.34 0x0616 交换MCU连接的I2C通道。</p> <p>3.35 0x0617 查询MCU用户自定义信息。</p> <p>3.36 0x0618 设置MCU用户自定义信息。</p> <p>0x0619 清除MCU用户自定义信息。</p> <p>修改：</p>

文档版本	发布日期	修改说明
		<a href="#">4.3 获取Atlas 300I 推理卡的温度数据示例</a> 章节修改产品名称。
01	2020-10-26	第一次正式发布。

# 目录

前言.....	ii
1 Atlas 300I 推理卡 I2C 地址.....	1
2 SMBus 协议接口定义.....	2
2.1 说明.....	2
2.2 Read Request 格式和 Write Request 格式.....	2
2.3 Read Response 格式和 Write Response 格式.....	4
2.4 错误码定义.....	5
3 Data Register Key 定义.....	6
3.1 0x0000 查询设备能力.....	7
3.2 0x0001 查询设备总体健康状态.....	9
3.3 0x0002 查询故障码.....	10
3.4 0x0003 查询芯片温度.....	11
3.5 0x0004 查询设备功耗.....	12
3.6 0x0005 查询 MCU 固件版本.....	13
3.7 0x0006 查询 Vendor ID.....	14
3.8 0x0007 查询 Device ID.....	15
3.9 0x0009 查询 Sub Vendor ID.....	16
3.10 0x000A 查询 Sub Device ID.....	17
3.11 0x000B 查询芯片电压.....	18
3.12 0x000C 获取错误日志.....	19
3.13 0x000F 查询 Board ID.....	20
3.14 0x0010 查询 PCB ID.....	21
3.15 0x0011 查询 MCU 复位时间.....	22
3.16 0x0015 查询电子标签.....	23
3.17 0x0016 启动/停止装备测试.....	24
3.18 0x0017 查询装备测试结果.....	26
3.19 0x0018 传输 MCU 固件文件.....	27
3.20 0x0019 升级控制命令.....	28
3.21 0x001A 查询升级状态.....	29
3.22 0x001C 查询标卡设备电压.....	30
3.23 0x001D 查询标卡设备温度.....	31
3.24 0x001F 查询 ECC 统计信息.....	32

3.25 0x0020 查询系统时间..... 34

3.26 0x0021 设置系统时间..... 35

3.27 0x0026 获取操作日志..... 36

3.28 0x0027 获取维护日志..... 37

3.29 0x0028 查询 BOM ID..... 38

3.30 0x0601 复位 D 芯片..... 39

3.31 0x0602 查询复位 D 芯片结果..... 40

3.32 0x0605 获取 Davinci 信息..... 41

3.33 0x0614 查询 MCU 心跳计数..... 42

3.34 0x0616 交换 MCU 连接的 I2C 通道..... 43

3.35 0x0617 查询 MCU 用户自定义信息..... 44

3.36 0x0618 设置 MCU 用户自定义信息..... 45

**4 请求和响应消息举例..... 46**

4.1 构造获取标卡设备温度的请求消息..... 46

4.2 构造获取标卡设备温度的响应消息..... 46

4.3 获取 Atlas 300I 推理卡的温度数据示例..... 46

**5 附录..... 49**

5.1 传感器名称约定..... 49

5.2 缩略语..... 51

5.3 故障定义..... 51



# 1 Atlas 300I 推理卡 I2C 地址

---

Atlas 300I 推理卡作为从设备，从设备地址为0xd8，连接到PCIe金手指上，和BMC（带外）对接。

# 2 SMBus 协议接口定义

- 2.1 说明
- 2.2 Read Request格式和Write Request格式
- 2.3 Read Response格式和Write Response格式
- 2.4 错误码定义

## 2.1 说明

- 请求和响应命令中所有字段都是小字节序优先（Least Significant Bit, LSB）。
- 根据smbus标准规范，PEC是可选项。
- smbus格式中，白色的框是master发送的数据，阴影部分是slave的数据。
- Command Code为0x20表示读写请求，0x21表示读写响应。
- Byte1…ByteN，其中N值为32。

## 2.2 Read Request 格式和 Write Request 格式

图 2-1 Block Write

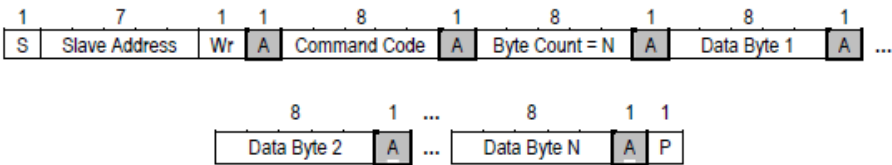
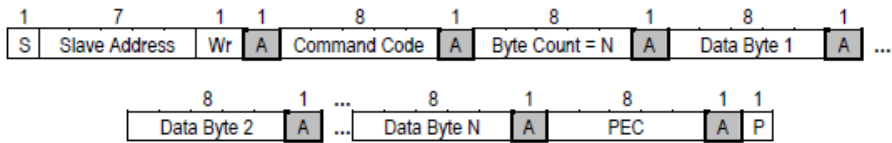


图 2-2 Block Write with PEC



Command code:0x20

请求数据结构

```
typedef struct tag_STD_SMBUS_REQ
{
    unsigned char lun;
    unsigned char arg;
    unsigned short opcode;
    unsigned int offset;
    unsigned int length;
    unsigned char data[20];
}STD_SMBUS_REQ;
```

请求数据说明

表 2-1 请求数据（Byte1~ByteN）

Request Data Structure	Bytes	Name	说明
	1	LUN	Bit0-Bit3=0：表示整卡。 Bit0-Bit3=(1-15)：表示MINI管理芯片的索引。 Bit7=1：表示最后一帧。 Bit7=0：表示数据未写完。
	2	arg	opcode的补充参数。
	3 ~ 4	opcode	命令字，详见3 Data Register Key定义。
	5~8	offset	数据读取偏移。从0开始，超长数据时需要分多次读取，如第一次读取填0，读取长度为32，则第二次读取时offset填32。

	9 ~ 12	length	请求数据长度。 <ul style="list-style-type: none"><li>读操作： 数据读长度（如果本规范定义opcode的数据总长度变长，则每次请求的数据读取长度为（32-12）bytes；如果本规范定义opcode的数据总长度固定，则除最后一帧请求外，其它帧的数据读取长度都必须是（32 -12）bytes。）</li><li>写操作： 数据写长度（本规范定义opcode对应的长度无论变长还是定长，除最后一帧请求外(如果只有一帧，那么第一帧即是最后一帧)，其它帧的数据读取长度都必须是（32 -12）bytes。）</li></ul>
	13 ~ N	data	数据区。

## 2.3 Read Response 格式和 Write Response 格式

图 2-3 Block Read

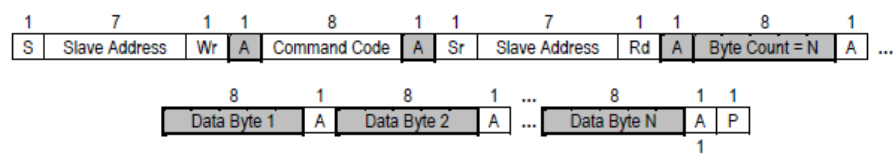
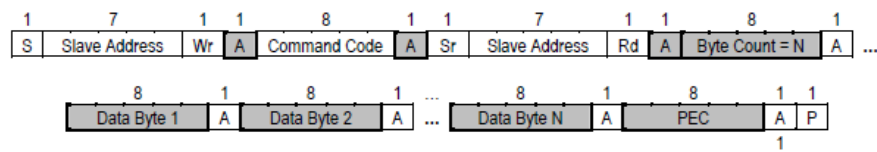


图 2-4 Block Read with PEC



Command code:0x21

### 响应数据结构

```
typedef struct tag_STD_SMBUS_RSP
{
    unsigned short error_code;
    unsigned short opcode;
    unsigned int total_length;
    unsigned int length;
```

```
unsigned char data[20];  
}STD_SMBUS_RSP;
```

响应数据结构说明

表 2-2 响应数据（Byte1~ByteN）

Response Data Structure	Bytes	Name	说明
	1 ~ 2	error_code	错误码，详见 <a href="#">2.4 错误码定义</a> 。  说明 对于所有的请求，不管响应的error_code是否为0，Slave都要返回total_length，length，data三个域的数据（全置为0），保持响应帧的总长度不变。
	3 ~ 4	opcode	命令字，详见 <a href="#">3 Data Register Key定义</a> 。
	5 ~ 8	total_length	opcode的数据总长度。
	9 ~ 12	length	数据区长度（如果本规范定义opcode的数据总长度变长，则每次响应（error_code到data）长度为32字节，如果是最后一帧并且数据区不足（32-12）bytes时，需要用0填充；如果本规范定义opcode的数据总长度固定，则除最后一帧响应外，其它帧响应数据区长度都必须是（32-12）bytes。）
	13 ~ N	data	数据区。如果数据少于32字节，则都填充0。

2.4 错误码定义

表 2-3 错误码定义

error_code	Description	备注
0	Success	无。
1	Opcode not support	Slave不响应数据，BMC只能通过驱动报错识别读取失败。
2	Parameter Error	无。
3	Internal Error	无。

# 3 Data Register Key 定义

---

- 3.1 0x0000 查询设备能力
- 3.2 0x0001 查询设备总体健康状态
- 3.3 0x0002 查询故障码
- 3.4 0x0003 查询芯片温度
- 3.5 0x0004 查询设备功耗
- 3.6 0x0005 查询MCU固件版本
- 3.7 0x0006 查询Vendor ID
- 3.8 0x0007 查询Device ID
- 3.9 0x0009 查询Sub Vendor ID
- 3.10 0x000A 查询Sub Device ID
- 3.11 0x000B 查询芯片电压
- 3.12 0x000C 获取错误日志
- 3.13 0x000F 查询Board ID
- 3.14 0x0010 查询PCB ID
- 3.15 0x0011 查询MCU复位时间
- 3.16 0x0015 查询电子标签
- 3.17 0x0016 启动/停止装备测试
- 3.18 0x0017 查询装备测试结果
- 3.19 0x0018 传输MCU固件文件
- 3.20 0x0019 升级控制命令
- 3.21 0x001A 查询升级状态
- 3.22 0x001C 查询标卡设备电压
- 3.23 0x001D 查询标卡设备温度

- 3.24 0x001F 查询ECC统计信息
- 3.25 0x0020 查询系统时间
- 3.26 0x0021 设置系统时间
- 3.27 0x0026 获取操作日志
- 3.28 0x0027 获取维护日志
- 3.29 0x0028 查询BOM ID
- 3.30 0x0601 复位D芯片
- 3.31 0x0602 查询复位D芯片结果
- 3.32 0x0605 获取Davinci信息
- 3.33 0x0614 查询MCU心跳计数
- 3.34 0x0616 交换MCU连接的I2C通道
- 3.35 0x0617 查询MCU用户自定义信息
- 3.36 0x0618 设置MCU用户自定义信息

3.1 0x0000 查询设备能力

请求

字节顺序 (LSB)	说明	域内容
1	LUN/flag	Bit0-bit3=0：表示整卡。 Bit0-bit3=(1-15)：表示MINI管理芯片的索引。 Bit7=1：表示最后一帧。 Bit7=0：表示数据未写完。 仅支持获取整卡数据信息。
2	Request parameter	请求的参数，作为opcode的补充参数，不能以bit位的形式占用。 0x00：本命令不使用此参数。
3~4	opcode	0x0000
5~8	offset	0
9~12	length	以实际为准。

响应

字节顺序 ( LSB )	说明	域内容
1~2	error_code	0：成功。 其它为错误码。
3~4	opcode	0x0000
5~8	total_length	以实际为准。
9~12	length	数据长度，以实际为准。
13~N	Response data	<div>1. Byte[0~1](2 Byte): Identify[BIT 0~11]: 固定值为0xeee, 定义该格式的识别码。 Version[BIT 12~15]: 初始版本为1, 小字节序。</div> <div>2. Byte[2](1 Byte): Type: PCIe类型定义。<ul style="list-style-type: none"><li>0: 无效卡。</li><li>1: RAID卡。</li><li>2: SSD卡。</li><li>3: NIC卡。</li><li>4: 压缩卡。</li><li>6: NPU卡。</li><li>7: 中端FPGA卡。</li></ul></div> <div>3. Byte[3~4](2 Byte): Opcode Capability Number: n个Opcode能力。</div> <div>4. Byte[3~4] 2*N Byte: 具体的能力。<ul style="list-style-type: none"><li>Opcode列表, 每个Opcode都是小字节序。</li><li>0x0001</li><li>0x0002</li><li>0x000E</li></ul></div>



### 3.2 0x0001 查询设备总体健康状况

#### 请求

字节顺序 (LSB)	说明	域内容
1	LUN/flag	Bit0-bit3=0: 表示整卡。 Bit0-bit3=(1-15): 表示MINI管理芯片的索引。 Bit7=1: 表示最后一帧。 Bit7=0: 表示数据未写完。
2	Request parameter	0x00
3~4	opcode	0x0001
5~8	offset	0
9~12	length	以实际为准。

#### 响应

字节顺序 (LSB)	说明	域内容
1~2	error_code	0: 成功。 其它为错误码。
3~4	opcode	0x0001
5~8	total_length	1字节。
9~12	length	1字节。
13~N	Response data	设备总体健康状况: 只代表本部件, 不包括与本部件存在逻辑关系的其它部件。 <ul style="list-style-type: none"><li>0: 正常。</li><li>1: 一般告警。</li><li>2: 重要告警。</li><li>3: 紧急告警。</li></ul> 类型: 整型。 Byte[0]: 小字节序优先 (LSB)。

## 3.3 0x0002 查询故障码

### 请求

字节顺序 (LSB)	说明	域内容
1	LUN/flag	Bit0-bit3=0: 表示整卡。 Bit0-bit3=(1-15): 表示MINI管理芯片的索引。 Bit7=1: 表示最后一帧。 Bit7=0: 表示数据未写完。 仅支持获取整卡数据信息。
2	Request parameter	0x00: 本命令不使用此参数。
3~4	opcode	0x0002
5~8	offset	0
9~12	length	以实际为准。

### 响应

字节顺序 (LSB)	说明	域内容
1~2	error_code	0: 成功。 其它为错误码。
3~4	opcode	0x0002
5~8	total_length	以实际为准。
9~12	length	数据长度, 以实际为准。
13~N	Response data	byte[0~3]: 小字节序优先 (LSB)。 类型: 整型。 故障码: 上面总体健康非0时的故障码, 有n个并发故障时, 需返回n个故障码, 大小为2*n; 若无故障, 则填0x0000, 小字节序。

### 3.4 0x0003 查询芯片温度

#### 请求

字节顺序 ( LSB )	说明	域内容
1	LUN/flag	Bit0-bit3=0：表示整卡。 Bit0-bit3=(1-15)：表示MINI管理芯片的索引。 Bit7=1：表示最后一帧。 Bit7=0：表示数据未写完。 <b>说明</b> MCU 20.1.1版本，仅支持获取整卡数据信息。
2	Request parameter	0x00：本命令不使用此参数。
3~4	opcode	0x0003
5~8	offset	0
9~12	length	以实际为准。

#### 响应

字节顺序 ( LSB )	说明	域内容
1~2	error_code	0：成功。 其它为错误码。
3~4	opcode	0x0003
5~8	total_length	2字节
9~12	length	2字节
13~N	Response data	byte[0~1]：小字节序优先（ LSB ）。 类型：整型。 芯片温度：单位摄氏度，精度为1摄氏度，有小数点则四舍五入。16位带符号类型，小字节序。设备侧返回的值就是实际温度。针对I2C协议，如果温度为无效数据则填0x7ffd，如果温度读取失败则填0x7fff。

## 3.5 0x0004 查询设备功耗

### 请求

字节顺序 (LSB)	说明	域内容
1	LUN/flag	Bit0-bit3=0: 表示整卡。 Bit0-bit3=(1-15): 表示MINI管理芯片的索引。 Bit7=1: 表示最后一帧。 Bit7=0: 表示数据未写完。
2	Request parameter	0x00: 本命令不使用此参数。
3~4	opcode	0x0004
5~8	offset	0
9~12	length	以实际为准。

### 响应

字节顺序 (LSB)	说明	域内容
1~2	error_code	0: 成功。 其它为错误码。
3~4	opcode	0x0004
5~8	total_length	2字节
9~12	length	2字节
13~N	Response data	byte[0~1]: 小字节序优先 (LSB)。 类型: 整型。 设备功耗: 单位为W, 精度为0.1W。16位无符号short类型, 小字节序。 计算公式: value=reading*0.1。 针对I2C协议, 如果功耗为无效数据则填0x7ffd, 如果功耗读取失败则填0x7fff。

## 3.6 0x0005 查询 MCU 固件版本

### 请求

字节顺序 (LSB)	说明	域内容
1	LUN/flag	Bit0-bit3=0: 表示整卡。 Bit0-bit3=(1-15): 表示MINI管理芯片的索引。 Bit7=1: 表示最后一帧。 Bit7=0: 表示数据未写完。
2	Request parameter	0x00: 本命令不使用此参数。
3~4	opcode	0x0005
5~8	offset	0
9~12	length	以实际为准。

### 响应

字节顺序 (LSB)	说明	域内容
1~2	error_code	0: 成功。 其它为错误码。
3~4	opcode	0x0005
5~8	total_length	以实际为准。
9~12	length	数据长度, 以实际为准。
13~N	Response data	固件版本: <ul style="list-style-type: none"> <li>byte1表示Major版本。</li> <li>byte2表示Minor版本。</li> <li>byte3表示Revision版本。</li> </ul> 如果Revision不涉及则填0xff。显示为x.y[z]格式, x表示Major版本, y表示Minor版本, z表示Revision版本(可选), 如: 2.5.26、2.5。byte1第一个传输, byte3最后一个传输。

### 3.7 0x0006 查询 Vendor ID

请求

字节顺序 ( LSB )	说明	域内容
1	LUN/flag	Bit0-bit3=0：表示整卡。 Bit0-bit3=(1-15)：表示MINI管理芯片的索引。 Bit7=1：表示最后一帧。 Bit7=0：表示数据未写完。 仅支持获取整卡数据信息。
2	Request parameter	0x00：本命令不使用此参数。
3~4	opcode	0x0006
5~8	offset	0
9~12	length	以实际为准。

响应

字节顺序 ( LSB )	说明	域内容
1~2	error_code	0：成功。 其它为错误码。
3~4	opcode	0x0006
5~8	total_length	2字节
9~12	length	2字节
13~N	Response data	byte[0~1]：小字节序优先（ LSB ）。 类型：整型。 Vendor ID：表示PCIe设备的Vendor ID，小字节序。

### 3.8 0x0007 查询 Device ID

请求

字节顺序 ( LSB )	说明	域内容
1	LUN/flag	Bit0-bit3=0：表示整卡。 Bit0-bit3=(1-15)：表示MINI管理芯片的索引。 Bit7=1：表示最后一帧。 Bit7=0：表示数据未写完。 仅支持获取整卡数据信息。
2	Request parameter	0x00：本命令不使用此参数。
3~4	opcode	0x0007
5~8	offset	0
9~12	length	以实际为准。

响应

字节顺序 ( LSB )	说明	域内容
1~2	error_code	0：成功。 其它为错误码。
3~4	opcode	0x0007
5~8	total_length	2字节
9~12	length	2字节
13~N	Response data	byte[0~1]：小字节序优先（ LSB ）。 类型：整型。 Device ID：表示PCIe设备的Device ID，小字节序。

### 3.9 0x0009 查询 Sub Vendor ID

请求

字节顺序 ( LSB )	说明	域内容
1	LUN/flag	Bit0-bit3=0：表示整卡。 Bit0-bit3=(1-15)：表示MINI管理芯片的索引。 Bit7=1：表示最后一帧。 Bit7=0：表示数据未写完。 仅支持获取整卡数据信息。
2	Request parameter	0x00：本命令不使用此参数。
3~4	opcode	0x0009
5~8	offset	0
9~12	length	以实际为准。

响应

字节顺序 ( LSB )	说明	域内容
1~2	error_code	0：成功。 其它为错误码。
3~4	opcode	0x0009
5~8	total_length	2字节
9~12	length	2字节
13~N	Response data	byte[0~1]：小字节序优先（ LSB ）。 类型：整型。 Sub VID：表示PCIe设备的Sub Vendor ID，小字节序。



### 3.10 0x000A 查询 Sub Device ID

请求

字节顺序 ( LSB )	说明	域内容
1	LUN/flag	Bit0-bit3=0：表示整卡。 Bit0-bit3=(1-15)：表示MINI管理芯片的索引。 Bit7=1：表示最后一帧。 Bit7=0：表示数据未写完。 仅支持获取整卡数据信息。
2	Request parameter	0x00：本命令不使用此参数。
3~4	opcode	0x000A
5~8	offset	0
9~12	length	以实际为准。

响应

字节顺序 ( LSB )	说明	域内容
1~2	error_code	0：成功。 其它为错误码。
3~4	opcode	0x000A
5~8	total_length	2字节
9~12	length	2字节
13~N	Response data	byte[0~1]：小字节序优先（LSB）。 类型：整型。 Sub DID：表示PCIe设备的Sub Device ID，小字节序。

### 3.11 0x000B 查询芯片电压

#### 请求

字节顺序 ( LSB )	说明	域内容
1	LUN/flag	Bit0-bit3=0：表示整卡。 Bit0-bit3=(1-15)：表示MINI管理芯片的索引。 Bit7=1：表示最后一帧。 Bit7=0：表示数据未写完。 <b>说明</b> MCU 20.1.1版本，仅支持获取整卡数据信息。
2	Request parameter	0x00：本命令不使用此参数。
3~4	opcode	0x000B
5~8	offset	0
9~12	length	以实际为准。

#### 响应

字节顺序 ( LSB )	说明	域内容
1~2	error_code	0：成功。 其它为错误码。
3~4	opcode	0x000B
5~8	total_length	2字节
9~12	length	2字节
13~N	Response data	byte[0~1]：小字节序优先（ LSB ）。 类型：整型。 芯片电压：单位为V，精度为0.01V。16位无符号short类型，小字节序。 计算公式：value=reading*0.01。 针对I2C协议，如果电压为无效数据则填0x7ffd，如果电压读取失败则填0x7fff。

### 3.12 0x000C 获取错误日志

#### 请求

字节顺序 ( LSB )	说明	域内容
1	LUN/flag	Bit0-bit3=0：表示整卡。 Bit0-bit3=(1-15)：表示MINI管理芯片的索引。 Bit7=1：表示最后一帧。 Bit7=0：表示数据未写完。 仅支持获取整卡数据信息。
2	Request parameter	0x00：本命令不使用此参数。
3~4	opcode	0x000C
5~8	offset	日志偏移地址。
9~12	length	以实际为准。

#### 响应

字节顺序 ( LSB )	说明	域内容
1~2	error_code	0：成功。 其它为错误码。
3~4	opcode	0x000C
5~8	total_length	日志总大小
9~12	length	单帧数据长度
13~N	Response data	日志数据

### 3.13 0x000F 查询 Board ID

请求

字节顺序 ( LSB )	说明	域内容
1	LUN/flag	Bit0-bit3=0：表示整卡。 Bit0-bit3=(1-15)：表示MINI管理芯片的索引。 Bit7=1：表示最后一帧。 Bit7=0：表示数据未写完。 仅支持获取整卡数据信息。
2	Request parameter	0x00：本命令不使用此参数。
3~4	opcode	0x000F
5~8	offset	0
9~12	length	以实际为准。

响应

字节顺序 ( LSB )	说明	域内容
1~2	error_code	0：成功。 其它为错误码。
3~4	opcode	0x000F
5~8	total_length	2字节
9~12	length	2字节
13~N	Response data	byte[0~1]：小字节序优先（ LSB ）。 类型：整型。 BoardId：板卡ID，小字节序。

### 3.14 0x0010 查询 PCB ID

请求

字节顺序 ( LSB )	说明	域内容
1	LUN/flag	Bit0-bit3=0：表示整卡。 Bit0-bit3=(1-15)：表示MINI管理芯片的索引。 Bit7=1：表示最后一帧。 Bit7=0：表示数据未写完。 仅支持获取整卡数据信息。
2	Request parameter	0x00：本命令不使用此参数。
3~4	opcode	0x0010
5~8	offset	0
9~12	length	以实际为准。

响应

字节顺序 ( LSB )	说明	域内容
1~2	error_code	0：成功。 其它为错误码。
3~4	opcode	0x0010
5~8	total_length	1字节
9~12	length	1字节
13~N	Response data	类型：整型，从1开始。 PCB Id：PCB版本ID，如，1表示PCB版本为A，2表示PCB版本为B，以此类推。

### 3.15 0x0011 查询 MCU 复位时间

请求

字节顺序 ( LSB )	说明	域内容
1	LUN/flag	Bit0-bit3=0：表示整卡。 Bit0-bit3=(1-15)：表示MINI管理芯片的索引。 Bit7=1：表示最后一帧。 Bit7=0：表示数据未写完。 仅支持获取整卡数据信息。
2	Request parameter	0x00：本命令不使用此参数。
3~4	opcode	0x0011
5~8	offset	0
9~12	length	以实际为准。

响应

字节顺序 ( LSB )	说明	域内容
1~2	error_code	0：成功。 其它为错误码。
3~4	opcode	0x0011
5~8	total_length	2字节
9~12	length	2字节
13~N	Response data	byte[0~1]：小字节序优先（ LSB ）。 类型：整型。 芯片复位时间：单位为秒。

### 3.16 0x0015 查询电子标签

请求

字节顺序 ( LSB )	说明	域内容
1	LUN/flag	Bit0-bit3=0：表示整卡。 Bit0-bit3=(1-15)：表示MINI管理芯片的索引。 Bit7=1：表示最后一帧。 Bit7=0：表示数据未写完。 仅支持获取整卡数据信息。
2	Request parameter	arg参数具体含义如下： <ul style="list-style-type: none"><li>• 0x10: Chassis Type。</li><li>• 0x11: Chassis Part Number。</li><li>• 0x12: Chassis Serial Number。</li><li>• 0x20: Mfg. Date / Time。</li><li>• 0x21: Board Manufacturer。</li><li>• 0x22: Board Product Name。</li><li>• 0x23: Board Serial Number。</li><li>• 0x24: Board Part Number。</li><li>• 0x25: Board FRU File ID。</li><li>• 0x30: Product Manufacturer Name。</li><li>• 0x31: Product Name。</li><li>• 0x32: Product Part/Model Number。</li><li>• 0x33: Product Version。</li><li>• 0x34: Product Serial Number。</li><li>• 0x35: Asset Tag。</li><li>• 0x36: Product FRU File ID。</li><li>• 0x60: System Manufacturer Name。</li><li>• 0x61: System Product Name。</li><li>• 0x62: System Version。</li><li>• 0x63: System Serial Number。</li></ul>
3~4	opcode	0x0015
5~8	offset	0
9~12	length	以实际为准。

响应

字节顺序 ( LSB )	说明	域内容
1~2	error_code	0：成功。 其它为错误码。
3~4	opcode	0x0015
5~8	total_length	以实际为准。
9~12	length	数据长度，以实际为准。
13~N	Response data	标签数据。

3.17 0x0016 启动/停止装备测试

请求

字节顺序 ( LSB )	说明	域内容
1	LUN/flag	Bit0-bit3=0：表示整卡。 Bit0-bit3=(1-15)：表示MINI管理芯片的索引。 Bit7=1：表示最后一帧。 Bit7=0：表示数据未写完。 仅支持设置整卡数据信息。
2	Request parameter	0x00：本命令不使用此参数。
3~4	opcode	0x0016
5~8	offset	0
9~12	length	以实际为准。



字节顺序 ( LSB )	说明	域内容
13	Request data	arg参数具体含义如下： <ul style="list-style-type: none"><li>• BYTE[0]：开始/停止测试。 Bit0-1： 1：开始测试。 0：停止测试。 Bit2-7：测试项编号。 0：ALL。 1~62：测试项编号。</li><li>• BYTE[1]：测试项ID。 2：外围设备自检。 4：点灯测试。 5：日志清除。 6：自愈功能。 7：降频中断测试。</li></ul> <b>说明</b> MCU 3.X.X系列版本：测试项ID仅包含2：外围设备自检。

响应

字节顺序 ( LSB )	说明	域内容
1~2	error_code	0：成功。 其它为错误码。
3~4	opcode	0x0016
5~8	total_length	以实际为准。
9~12	length	数据长度，以实际为准。

### 3.18 0x0017 查询装备测试结果

请求

字节顺序 (LSB)	说明	域内容
1	LUN/flag	Bit0-bit3=0：表示整卡。 Bit0-bit3=(1-15)：表示MINI管理芯片的索引。 Bit7=1：表示最后一帧。 Bit7=0：表示数据未写完。 仅支持获取整卡数据信息。
2	Request parameter	0x00：本命令不使用此参数。
3~4	opcode	0x0017
5~8	offset	0
9~12	length	以实际为准。
13	Request data	arg参数具体含义如下： BYTE[0]：测试项ID。 2：外围设备自检。 4：点灯测试。 5：日志清除。 6：自愈功能。 7：降频中断测试。 <b>说明</b> MCU 3.X.X系列版本：测试项ID仅包含2：外围设备自检。

响应

字节顺序 (LSB)	说明	域内容
1~2	error_code	0：成功。 其它为错误码。
3~4	opcode	0x0017
5~8	total_length	以实际为准。
9~12	length	数据长度，以实际为准。

字节顺序 (LSB)	说明	域内容
13~N	Response data	<ul style="list-style-type: none"> <li>BYTE[0]: 装备测试结果。 0: 表示成功。 1: 表示失败。 2: 测试中。</li> <li>BYTE[1]: 表示测试进度。</li> <li>BYTE[2:n]: 失败结果描述。 byte0为1时, byte1~byteN具体描述失败的装备测试结果。</li> </ul>

## 3.19 0x0018 传输 MCU 固件文件

### 请求

字节顺序 (LSB)	说明	域内容
1	LUN/flag	Bit0-bit3=0: 表示整卡。 Bit0-bit3=(1-15): 表示MINI管理芯片的索引。 Bit7=1: 表示最后一帧。 Bit7=0: 表示数据未写完。 仅支持设置整卡数据信息。
2	Request parameter	0x00: 本命令不使用此参数。
3~4	opcode	0x0018
5~8	offset	MCU固件数据偏移地址。
9~12	length	以实际为准。
13~14	Request data	MCU固件码流。

### 响应

字节顺序 (LSB)	说明	域内容
1~2	error_code	0: 成功。 其它为错误码。
3~4	opcode	0x0018
5~8	total_length	以实际为准。

字节顺序 (LSB)	说明	域内容
9~12	length	数据长度，以实际为准。
13~N	Response data	NA，不涉及，此命令无请求数据。

## 3.20 0x0019 升级控制命令

### 请求

字节顺序 (LSB)	说明	域内容
1	LUN/flag	Bit0-bit3=0：表示整卡。 Bit0-bit3=(1-15)：表示MINI管理芯片的索引。 Bit7=1：表示最后一帧。 Bit7=0：表示数据未写完。 仅支持设置整卡数据信息。
2	Request parameter	0x00：本命令不使用此参数。
3~4	opcode	0x0019
5~8	offset	0
9~12	length	以实际为准。
13~14	Request data	1：开始升级MCU固件。 3：激活升级MCU固件。 4：切换MCU镜像。

### 响应

字节顺序 (LSB)	说明	域内容
1~2	error_code	0：成功。 其它为错误码。
3~4	opcode	0x0019
5~8	total_length	以实际为准。
9~12	length	数据长度，以实际为准。
13~N	Response data	NA，不涉及，此命令无请求数据。

### 3.21 0x001A 查询升级状态

#### 请求

字节顺序 ( LSB )	说明	域内容
1	LUN/flag	Bit0-bit3=0：表示整卡。 Bit0-bit3=(1-15)：表示MINI管理芯片的索引。 Bit7=1：表示最后一帧。 Bit7=0：表示数据未写完。 仅支持获取整卡数据信息。
2	Request parameter	0x00：本命令不使用此参数。
3~4	opcode	0x001A
5~8	offset	0
9~12	length	以实际为准。
13~14	Request data	1：MCU。 <b>说明</b> MCU 3.3.0及以上版本支持该参数。

#### 响应

字节顺序 ( LSB )	说明	域内容
1~2	error_code	0：成功。 其它为错误码。
3~4	opcode	0x001A
5~8	total_length	以实际为准。
9~12	length	数据长度，以实际为准。

字节顺序 ( LSB )	说明	域内容
13~14	Response data	类型：整型。 升级（ 查询升级状态 ）。 BYTE[0]: <ul style="list-style-type: none"><li>0：空闲状态。</li><li>1：正在升级主区。</li><li>2：当前状态不支持升级，如主备同步时。</li><li>3：升级失败。</li></ul> BYTE[1]: byte0为1时显示升级进度，取值范围为 [0-100]。

### 3.22 0x001C 查询标卡设备电压

请求

字节顺序 ( LSB )	说明	域内容
1	LUN/flag	Bit0-bit3=0：表示整卡。 Bit0-bit3=(1-15)：表示MINI管理芯片的索引。 Bit7=1：表示最后一帧。 Bit7=0：表示数据未写完。
2	Request parameter	0x00：本命令不使用此参数。
3~4	opcode	0x001C
5~8	offset	第一次：0x00 第二次：0x14 第三次：0x28 .....
9~12	length	以实际为准。

响应

字节顺序 (LSB)	说明	域内容
1~2	error_code	0: 成功。 其它为错误码。
3~4	opcode	0x001C
5~8	total_length	以实际为准。
9~12	length	数据长度，以实际为准。
13~N	Response data	BYTE[0]: 电压个数。 BYTE[1:8]: 电压1名称。详细信息请参见 <a href="#">5.1 传感器名称约定</a> 。 BYTE[9:10]: 电压1的电压值。 ... BYTE[10n-9:10n-2]: 电压n名称。 BYTE[10n-1:10n]: 电压n的电压值。 电压值精确到0.01V，电压名称为所获取数据的ASCII码值。 针对I2C协议，如果电压为无效数据则填0x7ffd，如果电压读取失败则填0x7fff。

3.23 0x001D 查询标卡设备温度

请求

字节顺序 (LSB)	说明	域内容
1	LUN/flag	Bit0-bit3=0: 表示整卡。 Bit0-bit3=(1-15): 表示MINI管理芯片的索引。 Bit7=1: 表示最后一帧。 Bit7=0: 表示数据未写完。
2	Request parameter	0x00
3~4	opcode	0x001D
5~8	offset	第一次: 0x00 第二次: 0x14 第三次: 0x28 .....

字节顺序 ( LSB )	说明	域内容
9~12	length	以实际为准。

响应

字节顺序 ( LSB )	说明	域内容
1~2	error_code	0：成功。 其它为错误码。
3~4	opcode	0x001D
5~8	total_length	以实际为准。
9~12	length	数据长度，以实际为准。
13~N	Response data	Get peripheral Device Temperature(温度名称+温度值) Byte[0]：温度个数。 Byte[1:8]：温度1名称，详细信息请参见 <a href="#">5.1 传感器名称约定</a> 。 Byte[9:10]：温度1的温度值。 ... Byte[n-9:n-2]：温度n名称。 Byte[n-1:n]：温度n的温度值。 温度名称为所获取数据的ASCII码值。 针对I2C协议，如果温度为无效数据则填0x7ffd，如果温度读取失败则填0x7fff。

3.24 0x001F 查询 ECC 统计信息

请求

字节顺序 ( LSB )	说明	域内容
1	LUN/flag	Bit0-bit3=0：表示整卡。 Bit0-bit3=(1-15)：表示MINI管理芯片的索引。 Bit7=1：表示最后一帧。 Bit7=0：表示数据未写完。
2	Request parameter	0x00：本命令不使用此参数。



字节顺序 ( LSB )	说明	域内容
3~4	opcode	0x001F
5~8	offset	0
9~12	length	以实际为准。
13~14	Request data	Bit0: device type, DDR。 Bit6-7: error type, 固定为0x0。 0x00: ALL。 0x01: single-bit error counter。 0x02: double-bit error counter。

响应

字节顺序 ( LSB )	说明	域内容
1~2	error_code	0: 成功 其它为错误码。
3~4	opcode	0x001F
5~8	total_length	以实际为准。
9~12	length	数据长度，以实际为准。
13~N	Response data	如果Request parameter的 error type为 ALL: Byte[0~3]: int single-bit error counter Byte[4~8]: int double-bit error counter 如果Request parameter的error type为 1/2: Byte[0~3]: int single-bit/double-bit error counter

### 3.25 0x0020 查询系统时间

#### 请求

字节顺序 ( LSB )	说明	域内容
1	LUN/flag	Bit0-bit3=0：表示整卡。 Bit0-bit3=(1-15)：表示MINI管理芯片的索引。 Bit7=1：表示最后一帧。 Bit7=0：表示数据未写完。 仅支持获取整卡数据信息。
2	Request parameter	0x00：本命令不使用此参数。
3~4	opcode	0x0020
5~8	offset	0
9~12	length	以实际为准。

#### 响应

字节顺序 ( LSB )	说明	域内容
1~2	error_code	0：成功 其它为错误码。
3~4	opcode	0x0020
5~8	total_length	以实际为准。
9~12	length	数据长度，以实际为准。
13~N	Response data	byte[0~3]：小字节序优先 ( LSB ) the number of seconds from 00:00:00, January 1, 1970.

### 3.26 0x0021 设置系统时间

#### 请求

字节顺序 ( LSB )	说明	域内容
1	LUN/flag	Bit0-bit3=0：表示整卡。 Bit0-bit3=(1-15)：表示MINI管理芯片的索引。 Bit7=1：表示最后一帧。 Bit7=0：表示数据未写完。 仅支持设置整卡数据信息。
2	Request parameter	0x00：本命令不使用此参数。
3~4	opcode	0x0021
5~8	offset	0
9~12	length	4，数据长度，以实际为准。
13~N	Request data	byte[0~3]：小字节序优先（ LSB ） the number of seconds from 00:00:00, January 1, 1970.

#### 响应

字节顺序 ( LSB )	说明	域内容
1~2	error_code	0：成功。 其它为错误码。
3~4	opcode	0x0021
5~8	total_length	以实际为准。
9~12	length	数据长度，以实际为准。
13~N	Response data	NA，不涉及，此命令无请求数据。

### 3.27 0x0026 获取操作日志

#### 请求

字节顺序 ( LSB )	说明	域内容
1	LUN/flag	Bit0-bit3=0：表示整卡。 Bit0-bit3=(1-15)：表示MINI管理芯片的索引。 Bit7=1：表示最后一帧。 Bit7=0：表示数据未写完。 仅支持获取整卡数据信息。
2	Request parameter	0x00：本命令不使用此参数。
3~4	opcode	0x0026
5~8	offset	日志偏移地址。
9~12	length	以实际为准。

#### 响应

字节顺序 ( LSB )	说明	域内容
1~2	error_code	0：成功。 其它为错误码。
3~4	opcode	0x0026
5~8	total_length	日志总大小
9~12	length	单帧数据长度
13~N	Response data	日志数据

### 3.28 0x0027 获取维护日志

#### 请求

字节顺序 ( LSB )	说明	域内容
1	LUN/flag	Bit0-bit3=0：表示整卡。 Bit0-bit3=(1-15)：表示MINI管理芯片的索引。 Bit7=1：表示最后一帧。 Bit7=0：表示数据未写完。 仅支持获取整卡数据信息。
2	Request parameter	0x00：本命令不使用此参数。
3~4	opcode	0x0027
5~8	offset	日志偏移地址。
9~12	length	以实际为准。

#### 响应

字节顺序 ( LSB )	说明	域内容
1~2	error_code	0：成功。 其它为错误码。
3~4	opcode	0x0027
5~8	total_length	日志总大小
9~12	length	单帧数据长度
13~N	Response data	日志数据

### 3.29 0x0028 查询 BOM ID

#### 请求

字节顺序 ( LSB )	说明	域内容
1	LUN/flag	Bit0-bit3=0：表示整卡。 Bit0-bit3=(1-15)：表示MINI管理芯片的索引。 Bit7=1：表示最后一帧。 Bit7=0：表示数据未写完。 仅支持获取整卡数据信息。
2	Request parameter	0x00：本命令不使用此参数。
3~4	opcode	0x0028
5~8	offset	0
9~12	length	以实际为准。

#### 响应

字节顺序 ( LSB )	说明	域内容
1~2	error_code	0：成功。 其它为错误码。
3~4	opcode	0x0028
5~8	total_length	1字节
9~12	length	1字节
13~N	Response data	byte[0]：BOM ID

### 3.30 0x0601 复位 D 芯片

#### 请求

字节顺序 (LSB)	说明	域内容
1	LUN/flag	Bit0-bit3=0：表示整卡。 Bit0-bit3=(1-15)：表示MINI管理芯片的索引。 Bit7=1：表示最后一帧。 Bit7=0：表示数据未写完。
2	Request parameter	0x00：本命令不使用此参数。
3~4	opcode	0x0601
5~8	offset	0
9~12	length	0

#### 响应

字节顺序 (LSB)	说明	域内容
1~2	error_code	0：成功。 其它为错误码。
3~4	opcode	0x0601
5~8	total_length	以实际为准。
9~12	length	数据长度，以实际为准。
13~N	Response data	NA，不涉及，此命令无请求数据。

#### 约束说明

- 调用该接口前需要将PCIE设备与驱动解绑并移除
- 调用该接口后需要复位后重新扫描PCIE设备

### 3.31 0x0602 查询复位 D 芯片结果

#### 请求

字节顺序 ( LSB )	说明	域内容
1	LUN/flag	Bit0-bit3=0：表示整卡。 Bit0-bit3=(1-15)：表示MINI管理芯片的索引。 Bit7=1：表示最后一帧。 Bit7=0：表示数据未写完。 <b>说明</b> 仅支持获取MINI管理芯片数据信息。
2	Request parameter	0x00：本命令不使用此参数。
3~4	opcode	0x0602
5~8	offset	0
9~12	length	1

#### 响应

字节顺序 ( LSB )	说明	域内容
1~2	error_code	0：成功。 其它为错误码。
3~4	opcode	0x0602
5~8	total_length	以实际为准。
9~12	length	数据长度，以实际为准。
13~N	Response data	byte[0]： <ul style="list-style-type: none"><li>0：复位成功。</li><li>1：复位失败。</li></ul>



### 3.32 0x0605 获取 Davinci 信息

请求

字节顺序 (LSB)	说明	域内容
1	LUN/flag	Bit0-bit3=0: 表示整卡。 Bit0-bit3=(1-15): 表示MINI管理芯片的索引。 Bit7=1: 表示最后一帧。 Bit7=0: 表示数据未写完。
2	Request parameter	0x00: 本命令不使用此参数。
3~4	opcode	0x0605
5~8	offset	0
9~12	length	以实际为准。
13	Request data	bit 0~3: device type <ul style="list-style-type: none"><li>1: 内存。</li><li>2: CPU。</li></ul> bit 4~7: info type <ul style="list-style-type: none"><li>0: utilization_rate。</li><li>1: frequency。</li><li>2: memory_size (只针对内存)。</li></ul>

响应

字节顺序 (LSB)	说明	域内容
1~2	error_code	0: 成功。 其它为错误码。
3~4	opcode	0x0605
5~8	total_length	以实际为准。
9~12	length	数据长度, 以实际为准。
13~N	Response data	整型: 同时只有一个有效。 utilization_rate: 单位%。 frequency: 单位MHz。 memory_size大小 (只针对内存)。

### 3.33 0x0614 查询 MCU 心跳计数

#### 请求

字节顺序 ( LSB )	说明	域内容
1	LUN/flag	Bit0-bit3=0：表示整卡。 Bit0-bit3=(1-15)：表示MINI管理芯片的索引。 Bit7=1：表示最后一帧。 Bit7=0：表示数据未写完。 仅支持获取整卡数据信息。
2	Request parameter	0x00：本命令不使用此参数。
3~4	opcode	0x0614
5~8	offset	0
9~12	length	0

#### 响应

字节顺序 ( LSB )	说明	域内容
1~2	error_code	0：成功。 其它为错误码。
3~4	opcode	0x0614
5~8	total_length	4
9~12	length	4
13	Response data	当前心跳计数。

3.34 0x0616 交换 MCU 连接的 I2C 通道

请求

字节顺序 ( LSB )	说明	域内容
1	LUN/flag	Bit0-bit3=0：表示整卡。 Bit0-bit3=(1-15)：表示MINI管理芯片的索引。 Bit7=1：表示最后一帧。 Bit7=0：表示数据未写完。 仅支持设置整卡数据信息。
2	Request parameter	0x00：本命令不使用此参数。
3~4	opcode	0x0616
5~8	offset	0
9~12	length	1
13	Request data	1：mini1。 3：mini3。 其他：无效数据。

响应

字节顺序 ( LSB )	说明	域内容
1~2	error_code	0：成功。 其它为错误码。
3~4	opcode	0x0616
5~8	total_length	0
9~12	length	0

### 3.35 0x0617 查询 MCU 用户自定义信息

#### 请求

字节顺序 ( LSB )	说明	域内容
1	LUN/flag	Bit0-bit3=0：表示整卡。 Bit0-bit3=(1-15)：表示MINI管理芯片的索引。 Bit7=1：表示最后一帧。 Bit7=0：表示数据未写完。 仅支持获取整卡数据信息。
2	Request parameter	0x00：本命令不使用此参数。
3~4	opcode	0x0617
5~8	offset	自定义信息偏移地址。
9~12	length	以实际为准。

#### 响应

字节顺序 ( LSB )	说明	域内容
1~2	error_code	0：成功。 其它为错误码。
3~4	opcode	0x0617
5~8	total_length	以实际为准。
9~12	length	数据长度，以实际为准。
13~N	Response data	用户自定义信息。 用户自定义信息为所获取数据的ASCII码值。

3.36 0x0618 设置 MCU 用户自定义信息

请求

字节顺序 ( LSB )	说明	域内容
1	LUN/flag	Bit0-bit3=0：表示整卡。 Bit0-bit3=(1-15)：表示MINI管理芯片的索引。 Bit7=1：表示最后一帧。 Bit7=0：表示数据未写完。 仅支持设置整卡数据信息。
2	Request parameter	0x00：本命令不使用此参数。
3~4	opcode	0x0618
5~8	offset	自定义信息偏移地址。
9~12	length	以实际为准。
13~N	Request data	首帧包含以下8字节的头部信息： Byte[0~3]：版本号，当前为1。 Byte[4~5]：用户自定义信息的字节长度。 Byte[6~7]：用户自定义信息的CRC16校验和。 其余字节为用户自定义信息，长度以实际为准。

响应

字节顺序 ( LSB )	说明	域内容
1~2	error_code	0：成功。 其它为错误码。
3~4	opcode	0x0618
5~8	total_length	以实际为准。
9~12	length	数据长度，以实际为准。

# 4 请求和响应消息举例

## 4.1 构造获取标卡设备温度的请求消息

## 4.2 构造获取标卡设备温度的响应消息

## 4.3 获取Atlas 300I 推理卡的温度数据示例

### 4.1 构造获取标卡设备温度的请求消息

```
STD_SMBUS_REQ req = {0};
// 发送请求报文
req.lun = 0x80;
req.arg = (unsigned char)item_type;
req.opcode = (unsigned short)opcode;
req.offset = (unsigned int)offset;
req.length = (unsigned int)length;

i2c_smbus_write_block(dev_name, MCU_SLAVE_ADDR, SEND_REQUEST, DMP_MSG_HEAD_LENGTH +
req_data_len, (const unsigned char*)&req);
```

### 4.2 构造获取标卡设备温度的响应消息

```
STD_SMBUS_RSP rsp = {0};
i2c_smbus_read_block(dev_name, MCU_SLAVE_ADDR, SEND_RESPONSE, DMP_MSG_HEAD_LENGTH + length
+ 1, (unsigned char*)&rsp);
```

### 4.3 获取 Atlas 300I 推理卡的温度数据示例

Atlas 300I 推理卡的温度数据大于32字节，需要多次获取，示例代码如下。

```
int dcmi_mcu_get_info_dynamic(char *dev_name, int item_type, int opcode, int total_len, char *req_data, int
req_data_len, char *data_info, int *len)
{
    int ret;
    int offset;
    int len_curr = 0;
    int len_total = 0;
    int len_msg_curr = 0;
    int len_msg_total = 0;
    int count;
    int num_id;
    char buffer[DCMI_MCU_MSG_MAX_TOTAL_LEN] = {0};
    char *point = NULL;
```

```

int len_total_true;
point = &buffer[0];
ret = dcmi_mcu_get_info(dev_name, item_type, opcode, 0, total_len, req_data, req_data_len, point,
&len_curr, &len_total);
len_total_true = len_total;
if (ret != DCMI_OK) {
    printf("call dcmi_mcu_get_info fail.ret(%d)\n", ret);
    return DCMI_ERR;
}
if ((len_total <= len_curr) && (len_total < DCMI_MCU_MSG_MAX_TOTAL_LEN) && (len_total > 0)) {
    memcpy(data_info, buffer, len_total);
    *len = len_total;
    return DCMI_OK;
}
if (len_total > DCMI_MCU_MSG_MAX_TOTAL_LEN) {
    printf("total=%d bigger tran 256.\n", len_total);
    return DCMI_ERR;
}
len_total -= len_curr;
offset = len_curr;
point += len_curr;
count = len_total / DCMI_MCU_MSG_DATA_LEN;
if (len_total % DCMI_MCU_MSG_DATA_LEN != 0) {
    count++;
}
for (num_id = 0; num_id < count; num_id++)
{
    if (offset + DCMI_MCU_MSG_DATA_LEN > len_total)
    {
        len_curr = len_total - offset;
    }
    else
    {
        len_curr = DCMI_MCU_MSG_DATA_LEN;
    }
    ret = dcmi_mcu_get_info(dev_name, 0, opcode, offset, DCMI_MCU_MSG_DATA_LEN, req_data,
req_data_len, point, &len_msg_curr, &len_msg_total);
    if (ret != DCMI_OK)
    {
        printf("call dcmi_mcu_get_info fail.ret(%d)\n", ret);
        return DCMI_ERR;
    }
    offset += len_curr;
    point += len_curr;
}
if (len_total_true < DCMI_MCU_MSG_MAX_TOTAL_LEN)
{
    memcpy(data_info, buffer, len_total_true);
    *len = len_total + len_curr;
}
else
{
    return DCMI_ERR;
}
return DCMI_OK;
}

int show_chip_temp_info(char *dev_name, int opcode)
{
    char data_info[256] = { 0 }; // 返回的数据
    char tmp_data[9] = { 0 }; /* 临时存放温度名称 */
    int req_data = 0;
    int len = 0; // 返回当前数据的长度
    int temp_index, temp_num, temp_get;

    dcmi_mcu_get_info_dynamic(dev_name, 0, opcode, DCMI_MCU_MSG_DATA_LEN, (char *)&req_data, 0,
data_info, &len);

    temp_num = (int)data_info[0];
    for (temp_index = 0; temp_index < temp_num; temp_index++)

```

```
{
    temp_get = *(INT16*)(data_info + (temp_index * 10 + 9));
    /* PCIE带外管理规范, 温度名称长度为8(不含结束字符) */
    strncpy(tmp_data, data_info + temp_index * 10 + 1, 8);

    printf("\t%-8s%-22s :", tmp_data, " (C)");

    if (temp_get == 0x7fff) {
        printf(" %s\n", "Failed");
    } else if (temp_get == 0x7ffd) {
        printf(" %s\n", "NA");
    } else {
        printf(" %d\n", temp_get);
    }
}

return 0;
}
```



# 5 附录

- 5.1 传感器名称约定
- 5.2 缩略语
- 5.3 故障定义

## 5.1 传感器名称约定

表 5-1 BMC 与 Atlas 300I 推理卡（型号 3000，3010）的温度传感器名称约定

监控点	传感器名称
Mini0温度	MINI0
Mini1温度	MINI1
Mini2温度	MINI2
Mini3温度	MINI3
PCleSwitch温度	PCIESW（Atlas 300I 推理卡（型号：3000）无此传感器，MCU 3.1.0及以上版本返回无效值。）
DDR的LM75_1温度	DDR1
DDR的LM75_2温度	DDR2
PSIP的LM75温度	PSIP

表 5-2 BMC 与 Atlas 300I 推理卡（单板类型 IT21DMPC）的温度传感器名称约定

监控点	传感器名称
Mini0温度	MINI0
Mini1温度	MINI1

监控点	传感器名称
Mini2温度	MINI2
Mini3温度	MINI3
DDR的LM75_1温度	DDR1
DDR的LM75_2温度	DDR2
PSIP的LM75温度	PSIP

表 5-3 BMC 与 Atlas 300I 推理卡（型号 3000, 3010）的电压传感器名称约定

监控点	传感器名称
3.3V VDD电压	3V3 VDD (Atlas 300I 推理卡 (型号: 3000) 无此传感器, 返回无效值。)
0.9V VDD电压	0V9 VDD (Atlas 300I 推理卡 (型号: 3000) 无此传感器, 返回无效值。)
1.8V VDD电压	1V8 VDD (Atlas 300I 推理卡 (型号: 3000) 无此传感器, 返回无效值。)
VSYS VDD电压	VSYS VDD (Atlas 300I 推理卡 (型号: 3000) 无此传感器, 返回无效值。)
12V VDD电压	12V VDD (Atlas 300I 推理卡 (型号: 3000) 无此传感器, 返回无效值。)
MINI0电压	MINI0 VOL
MINI1电压	MINI1 VOL
MINI2电压	MINI2 VOL
MINI3电压	MINI3 VOL

表 5-4 BMC 与 Atlas 300I 推理卡（单板类型 IT21DMPC）的电压传感器名称约定

监控点	传感器名称
3.3V VDD电压	3V3 VDD
1.05V VDD电压	1V05 VDD
2.5V VDD电压	2V5 VDD
VSYS0 VDD电压	VSYS0 VDD
12V VDD电压	12V VDD
MINI0电压	MINI0 VOL

监控点	传感器名称
MINI1电压	MINI1 VOL
MINI2电压	MINI2 VOL
MINI3电压	MINI3 VOL
VSYS1 VDD电压	VSYS1 VDD
V3.3V VDD电压	V3V3 VDD

## 5.2 缩略语

<b>B</b>		
<b>BMC</b>	Baseboard Management Controller	主板管理控制单元
<b>L</b>		
<b>LSB</b>	Least Significant Bit	最低有效位
<b>M</b>		
<b>MCU</b>	Microcontroller Unit	微控制单元
<b>P</b>		
<b>PCIe</b>	Peripheral Component Interconnect Express	快捷外围部件互连标准
<b>PEC</b>	Packet Error Code	封包错误码

## 5.3 故障定义

故障码用2个字节表示，按如下位数进行编码：

故障码(10进制)	错误码描述	处理建议
7168	PCB或BOM版本信息异常	更换板卡
7196	板卡FLASH故障	更换板卡
7199	APP主备区损坏：片外FLASH中，APP主备区被破坏，MCU掉电后不能正常启动，提醒用户需要升级MCU更新片外FLASH的MCU主备区，若升级失败(FLASH异常)，提醒用户需要返厂烧写片外FLASH。	更换板卡

故障码(10进制)	错误码描述	处理建议
7500	PCIe Switch器件异常	尝试下电再上电恢复
7501	PCIe Switch内部故障	检查板卡是否安装正常
7571	LM75C温度传感器异常	尝试下电再上电恢复
7503	LM75B温度传感器异常	尝试下电再上电恢复
7502	LM75A温度传感器异常	尝试下电再上电恢复
7504	12V功率数据获取异常	尝试下电再上电恢复
7505	3V3功率数据获取异常	尝试下电再上电恢复
7506	VSYS电压获取异常	尝试下电再上电恢复
7507	VSYS电压过高严重告警	更换板卡
7508	VSYS电压过低严重告警	更换板卡
7509	0V9电压获取异常	尝试下电再上电恢复
7510	0V9电压过高严重告警	更换板卡
7511	0V9电压过低严重告警	更换板卡
7512	1V8电压获取异常	尝试下电再上电恢复
7513	1V8电压过高严重告警	更换板卡
7514	1V8电压过低严重告警	更换板卡
7515	12V电源功率过高告警	检查业务运行负载是否过大
7516	3V3电源功率过高告警	检查业务运行负载是否过大
7518	3V3电压获取异常	尝试下电再上电恢复
7519	查询PCIe Switch芯片温度失败	尝试下电再上电恢复
7520	MINI0芯片健康状态异常	联系华为工程师支持定位
7521	MINI1芯片健康状态异常	联系华为工程师支持定位
7522	MINI2芯片健康状态异常	联系华为工程师支持定位
7523	MINI3芯片健康状态异常	联系华为工程师支持定位
7528	MINI0芯片内部故障	联系华为工程师支持定位
7529	MINI1芯片内部故障	联系华为工程师支持定位
7530	MINI2芯片内部故障	联系华为工程师支持定位
7531	MINI3芯片内部故障	联系华为工程师支持定位
7536	MINI0芯片温度过高	检查设备散热是否正常

故障码(10进制)	错误码描述	处理建议
7537	MINI1芯片温度过高	检查设备散热是否正常
7538	MINI2芯片温度过高	检查设备散热是否正常
7539	MINI3芯片温度过高	检查设备散热是否正常
7572	MINI0芯片温度获取失败	尝试下电再上电恢复
7573	MINI1芯片温度获取失败	尝试下电再上电恢复
7574	MINI2芯片温度获取失败	尝试下电再上电恢复
7575	MINI3芯片温度获取失败	尝试下电再上电恢复
7544	LM75A温度过高触发降频	检查设备散热是否正常
7545	LM75B温度过高触发降频	检查设备散热是否正常
7546	LM75C温度过高触发降频	检查设备散热是否正常
7547	PCIE Switch温度过高触发降频	检查设备散热是否正常
7548	LM75A温度过高触发MINI芯片下电	检查设备散热是否正常
7549	LM75B温度过高触发MINI芯片下电	检查设备散热是否正常
7550	LM75C温度过高触发MINI芯片下电	检查设备散热是否正常
7551	PCIE Switch温度过高触发MINI芯片下电	检查设备散热是否正常
7570	LM75C温度轻微过高告警	检查设备散热是否正常
7560	PCIE Switch温度轻微过高告警	检查设备散热是否正常
7562	MINI0芯片PCIE速率异常	检查板卡是否安装正常
7563	MINI1芯片PCIE速率异常	检查板卡是否安装正常
7564	MINI2芯片PCIE速率异常	检查板卡是否安装正常
7565	MINI3芯片PCIE速率异常	检查板卡是否安装正常
7566	PCIE Switch上行链路速率异常	检查板卡是否安装正常
7567	板卡时钟器件异常	尝试下电再上电恢复

## 说明

MCU 20.1.1版本及3.X.X系列版本无7562、7563、7564、7565、7566故障码。