# AT 开放接口指南

## 1.1 概览

AT命令端口，支持从PowerWriter 两端控制输入和输出，即 USB-CDC 端口和扩展通用RX/TX 口输入和输入命令

- USB-CDC ： 用户可以通过USB-CDC 下发数据到PowerWriter ，也可以通过PowerWriter 支持的AT 命令控制PowerWriter 的行为，读取状态或者数据，这种模式适合用户二次开发桌面软件来集成PowerWriter 来控制生产流程。
- 硬件端UART（输出串口）： 用户除了使用USB -CDC 来控制PowerWriter 的行为之外，也可以将PowerWriter 集成到硬件控制系统中来来控制PowerWriter 的行为，数据上传和下载，这种模式适合脱离桌面系统的开发模式。

仓库地址：

```
#github
git clone https://github.com/openpowerwriter/PowerWriter_AT_Command.git
#gitee 码云(推介)
git clone https://gitee.com/openpowerwriter/PowerWriter_AT_Command.git
```

:::tip 科学上网提示

- Github Hosts：最新Github Hosts

:::

**刷新系统 DNS 缓存**

```
ipconfig /flushdns
```

## 1.2 AT 命令使能

硬件AT 命令接口默认为**关闭状态**，如需开启，请参考 **设备首选项**

## 1.3 AT 命令结构

标准的AT 命令 以 "AT" 开头，以回车CR(LF) 作为结尾，返回的响应样式是<回车><换行><响应内容><回车><换行>, 为纯字符串终端响应模式，在PowerWriter 上，由于部分接口需要发送大数据，以及数据加密需求，以及更好的实时性，降低通信带宽占用，综合权衡，**PowerWriter 开放AT接口采用统一的二进制格式，以便提供更好的性能**，指令结构如下所示：

| frame header | length | data | crc32 |
| --- | --- | --- | --- |
| "PWAT" | uint32_t | object | crc32 |

- frame header : 同步头信号，用于PowerWriter 从流中获取 AT 命令，**四字节，无 '\0' 结尾。**
- length: 命令帧长度，含：Frameheader、length、data、crc32。

Copyright ©2017-2020 All Rights Reserved. 创芯工坊科技（深圳）有限公司

data 的格式定义如下

```
typedef struct S_ATCmdObject{
    E_ATcmdType                 m_cmdType;          //Type

    uint32_t                    m_cmdSize;          //Size
    U_ATCmdProprety *           m_cmdPropretyp;     //Payload
}S_ATCmdObject;
```

m_cmdType: 表示命令类型，具体请参考E_ATcmdType的定义

m_cmdSize：表示当前命令的有效数据长度（预留）

U_ATCmdProprety *：表示当前命令的属性节点，每一个命令都有各自的属性节点，具体请参考 U_ATCmdProprety 的定义。

- crc32：命令帧的整帧校验值（frameheader、length、data）,参数配置：

> Polynomial : 0x04C11DB7
>
> Initial/Final remainder : 0xffffffff
>
> Input/Output reverse : yes

:::tip 提示

- 接口数据可选AES128CBC 加密，通讯密码需在PowerWriter 端手动设置。
- 纯命令行式 的AT 接口，请提交反馈。
- data 节点在定义时，默认强制以**字节对齐**，这么做的目的是去除不同平台的编译器特性差异(结构体填充)，会有一定的性能损失，为了弥补可能存在的性能损失（内存拷贝），所以在定义时尝试将属性节点默认对齐到 4字节（32位平台），实际上没有区别，但不能依赖编译器的默认特性。
- data 节点由于长度不统一，所以如果开启加密，将会检查 object 的总大小是否对齐到16 字节，如果没对齐，则会在尾部执行 Zero padding（某些开源的AES128CBC library 内部带了其他方式，如PKCS15，不依赖差异化特性是一个好的选择），故 m_cmdSize 实际可能小于 data 节点的长度，最多15个字节。

:::

## 1.4 AT 命令分类

当前PowerWriter AT 接口提供的操作接口如下：

```
// PowerWriter AT command type
   typedef enum E_ATcmdType
   {
       // Null command for reserved
       ATCmdNull = 0,

       // Query and configuration
       ATCmdGetWriterInfo = 1, // Get writer information
       ATCmdRspWriterInfo,     // Respone writer information

       ATCmdGetWriterCfg, // Get writer configuration
       ATCmdRspWriterCfg, // Respone writer configuration
```

Copyright ©2017-2020 All Rights Reserved. 创芯工坊科技（深圳）有限公司

```
15
16          // Online mode target chip operation
17          ATCmdConnectTarget = 100, // Connnect target chip (manual operation)
18          ATCmdGetTargetStatus,     // Get target chip status
19
20          ATCmdGetTargetChipID, // Get target chip id
21          ATCmdRspTargetChipID, // Respone target chip id
22
23          ATCmdReadTargetMemory, // Read target memory
24          ATCmdRspTargetMemory,  // Respone target memory content
25
26          ATCmdEraseTarget,        // Erase target chip (full)
27          ATCmdEraseTargetSectors, // Erase target sectors
28
29          ATCmdWriteTargetMemory, // Write target memory
30
31          ATCmdReadTargetOptionByte,        // Read target option byte
32          ATCmdRspTargetOptionByte,         // Respone target option byte
33          ATCmdWriteTargetVendorOptionByte, // Write vendor's default option
    byte to target
34          ATCmdWriteTargetUserOptionByte,   // Write user's option byte to
    target
35
36          // Offline mode target chip operation
37          ATCmdGetProjectInfo = 200, // Get project info from PowerWriter
38          ATCmdRspProjectInfo,
39
40          ATCmdLoadProject,   // Load project to PowerWriter
41          ATCmdLoadProjectSend,   //Load project to PowerWriter send package
42          ATCmdDisableProject, // Delete project from PowerWriter (Mark as
    invalid)
43
44          ATCmdStartOffline,     // Start offline programming
45          ATCmdGetOfflineStatus, // Get offline programming status
46
47          // Benchmark instruction
48          ATCmdFactoryRunSRAMFW = 300, // Load & run factory SRAM firmware
49          ATCmdFactoryRunFlashFW,      // Load & run factory Flash firmware
50
51          // Other command fields
52          ATCmdBroadcast = 400, // Broadcast data
53                                //...
54
55          // State instruction
56          ATCmdStatusOK = (UINT32_MAX - 100), // Status Ok
57          ATCmdStatusError,                   // Status error with error code
58          ATCmdStatusProgress,                // Status busy (progress)
59
60          // Align to 4 bytes, compiler independent
61          _ATCmdTypeMax = UINT32_MAX
62      } E_ATcmdType;
```

:::info 提示

Copyright ©2017-2020 All Rights Reserved. 创芯工坊科技（深圳）有限公司

:::

## 1.4.1 系统功能查询与配置

### 1.4.1.1 查询PowerWriter信息

ATCmdGetWriterInfo：

**调用参数**：无需参数，但有占位符(空结构体导致某些编译器警告)

```
typedef struct S_ATCmdObject{
    union powerwriter_at_type{
        uint32_t m_placeholder; // An placeholder for empty structure
   }property;
}S_ATCmdGetWriterInfo;
```

m_placeholder：占位符

:::tip

一部分命令只有命令类型，无参数，均以 S_ATCmdObject 表示，不再重复描述。

:::

**命令返回**：ATCmdRspWriterInfo

### 1.4.1.2 返回PowerWriter信息

ATCmdRspWriterInfo：

**调用参数**：ATCmdGetWriterInfo

**返回值:**

```
typedef struct S_ATCmdRspWriterInfo
{
    uint8_t m_oem[PW_OEM_LENGTH]; // PowerWriter oem, such as PW200,PW300,PWLINK etc...
    uint8_t m_sn[PW_SN_LENGTH];   // PowerWriter sn, formated as "xxxxxxxxxxx..."

    uint8_t m_hardwareVersion[PW_VERSION_LENGTH];   // PowerWriter hardware version, formated as "x.xx.xx"
    uint8_t m_bootloaderVersion[PW_VERSION_LENGTH]; // PowerWriter bootloader version, formated as "x.xx.xx"
    uint8_t m_interfaceVersion[PW_VERSION_LENGTH];  // PowerWriter interface version, formated as "x.xx.xx"
} S_ATCmdRspWriterInfo;
```

**m_oem**：PowerWriter OEM 型号。

**m_sn**：PowerWriter 序列号。

**m_hardwareVersion**: PowerWriter 硬件版本。

**m_bootloaderVersion**: PowerWriter  bootloader 版本。

Copyright ©2017-2020 All Rights Reserved. 创芯工坊科技（深圳）有限公司

#### 1.4.1.3 查询PowerWriter配置

ATCmdGetWriterCfg：

**调用参数**：S_ATCmdObject。

**命令返回**：ATCmdRspWriterCfg

#### 1.4.1.4 返回PowerWriter配置

ATCmdRspWriterCfg：

**调用参数**：ATCmdGetWriterCfg

**返回值：**

```
// Target names
    typedef struct S_TargetName
    {
        uint8_t m_targetVendor[PW_TARGET_NAME_MAX]; // target chip vendor,such as 'nations' (注：只能看，不能改)
        uint8_t m_targetSeries[PW_TARGET_NAME_MAX]; // target chip series,such as 'N32G020' (注：只能看，不能改)
        uint8_t m_targetModel[PW_TARGET_NAME_MAX];  // target chip model,such as 'N32G020G5' (注：只能看，不能改)
    } S_TargetName;

    // Base setting
    typedef struct S_BaseWriterCfg
    {
        uint8_t m_eraseType;      // Erase type: 0(no erase),1(chip erase),2(sector erase)
        uint8_t m_outputVextType; // Output VEXT: 0(1.8V),1(3.3V),2(5.0V),3(refer to external power supply)
        uint8_t m_clockType;      // Colok: 0(20Mhz fast mode),1(10Mhz default),2(5Mhz),3(2Mhz),4(1Mhz), etc ...
        uint8_t m_buzzerEnable;   // 0: disable 1: enable
        uint8_t m_obUpdateType;   // 0: No operation before programming, no operation after programming
                                  // 1: No operation is performed before programming, and user - defined OptionByte is      written after flash programming
                                  // 2: Restore default values is performed before programming, no operation after programming.
                                  // 3: Restore default values is performed before programming, and user - defined OptionByte is written after flash programming

    } S_BaseWriterCfg;
    // Offline configure
    typedef struct S_OfflineWriterCfg
    {
        uint8_t m_limitOfNumberEnable;  // 0 : disable 1: enable    (注：只能看，不能改)
        uint32_t m_limitOfNumber;       // Current remaining quantity (注：只
```

Copyright ©2017-2020 All Rights Reserved. 创芯工坊科技（深圳）有限公司

```
        uint16_t m_targetAutoInDebonce; // Target connected debonce time
        uint16_t m_chipAutoChkOut;      // Target disconnected debonce time
    } S_OfflineWriterCfg;
    // Output Ctrl configure
    typedef struct S_OutputCtrlWriterCfg
    {
        uint8_t m_disablePowerAP;    // 0 : disable 1:enable (Power off when programming is complete)
        uint8_t m_runAP;             // 0 : disbale 1:enable (Run when programming is complete)
        uint16_t m_poweronDebonce;   // Debonce the output power supply (ms)
        uint16_t m_powerdownDebonce; // Debonce the output power off (ms)
        uint8_t m_resetType;         // 0 : Keep low
                                     // 1 : Off, high resistance
                                     // 2 : Reset output then to off
    } S_OutputCtrlWriterCfg;

    // Writer configure for currrent project
    typedef struct S_ATCmdWriterCfg
    {
        S_TargetName m_targetName;             // Target names
        S_BaseWriterCfg m_baseCfg;             // Base config
        S_OfflineWriterCfg m_offlineCfg;       // Offline configure
        S_OutputCtrlWriterCfg m_outputCtrlCfg; // Output control configure
    } S_ATCmdRspWriterCfg, S_ATCmdSetWriterCfg;
```

### 1.4.1.4 更新PowerWriter配置

ATCmdSetWriterCfg

**调用参数**：S_ATCmdSetWriterCfg，同 ATCmdRspWriterCfg 的**返回值**

**返回值:**

ATCmdStatusOK、ATCmdStatusError。

:::tip 提示

- 处于安全考虑，部分PowerWriter 配置不允许更改(修改无效)，如：**m_targetName、m_limitOfNumberEnable、m_limitOfNumber**
- 烧录器配置信息，默认PowerWrietr 启动时加载离线项目的信息，如果没有预载离线项目, 则返回的是默认值。

:::

### 1.4.1.4 设置AT 接口波特率

ATCmdSetBaudrate

**调用参数**：S_ATCmdSetBaudrate

Copyright ©2017-2020 All Rights Reserved. 创芯工坊科技（深圳）有限公司

```
typedef struct S_ATCmdObject
{
    union powerwriter_at_type
    {
        uint32_t m_baudrate;    // Set UART AT new baudrate;
    }property;
}S_ATCmdSetBaudrate;
```

m_baudrate: 设置新的波特率

**返回值:**

ATCmdStatusOK、ATCmdStatusError。

:::caution 注意

- USB 端口无需更改波特率，为CDC 接口，支持所有的波特率，自动设置，UART 端口AT 为硬件UART 端口，**默认波特率为 9600。**
- PowerWriter 在接收到指令之后，按照设置以前的波特率返回响应，如果设置成功，PowerWriter 将自动进入新的波特率通信，如果失败，仍保持旧波特率设置。

:::

## 1.4.2 在线操作指令:

:::tip 提示

在线操作指令，需要先加载离线项目，进行数据预设才支持。

:::

### 1.4.2.1 连接目标芯片

ATCmdConnectTarget

**调用参数**：S_ATCmdObject

**返回值:**

ATCmdStatusOK、ATCmdStatusError。

### 1.4.2.2 获取目标芯片连接状态

ATCmdGetTargetStatus

**调用参数**：S_ATCmdObject

**返回值:**

ATCmdStatusOK、ATCmdStatusError。

### 1.4.2.3 获取目标芯片ID

ATCmdGetTargetChipID

**调用参数**：S_ATCmdObject

**返回值:**

Copyright ©2017-2020 All Rights Reserved. 创芯工坊科技（深圳）有限公司

### 1.4.2.4 返回目标芯片ID

ATCmdRspTargetChipID

**调用参数**：ATCmdGetTargetChipID

**返回值：**

```
    typedef struct S_ATCmdRspTargetChipID
    {
        uint8_t m_CIDSize;                    // Target chip ID size
        uint8_t m_CIDData[PW_TARGET_ID_MAX]; // Target chip ID data
    } S_ATCmdRspTargetChipID;
```

m_CIDSize：当前芯片的ID 长度

m_CIDData：当前芯片的ID Data Buffer ,预留缓冲区长度为 PW_TARGET_ID_MAX(16)

:::tip 提示

部分芯片的ID是非连续存储的, PowerWriter 会将非连续存储的芯片已连续存储的模式给出，去除冗余信息

:::

### 1.4.2.5 读取目标芯片数据

ATCmdReadTargetMemory

**调用参数**：S_ATCmdReadTargetMemory

```
    typedef struct S_ATCmdReadTargetMemory{
        uint32_t   m_address;           //Target chip address, support such as SRAM、Flash、EEPROM、OTP etc...
        uint32_t   m_size;              //Data size of a single read (Must be less than 256)
    }S_ATCmdReadTargetMemory;
```

m_address：读取数据的地址，支持SRAM，外设寄存器值(**部分模块可能需要特殊权限，请跟我们反馈**)，Flash，EEPROM，OTP等。

m_size：读取数据的大小。

:::tip 注意

单次读取的大小不要超过PW_BUFFER_MAX(256), 否则PowerWriter 只返回 PW_BUFFER_MAX的长度。PowerWriter 暂没提供连续读取的指令，连续虽然可以提高读写效率，但是需要比较大的存储空间来接收数据，这对UART 端的AT 接口并不友好，后续如有特殊需求，请反馈我们，此外也没有提供mem8、mem16、mem32 的接口。

:::

**返回值：**参见ATCmdRspTargetMemory、ATCmdStatusError。

Copyright ©2017-2020 All Rights Reserved. 创芯工坊科技（深圳）有限公司

### 1.4.2.6 返回目标芯片数据

ATCmdRspTargetMemory

**调用参数**：ATCmdReadTargetMemory

**返回值**：S_ATCmdReadTargetMemory

```
    // Respone target chip memory
    typedef struct S_ATCmdRspTargetMemory{
        uint32_t    m_address;              //Current data address
        uint32_t    m_size;                 //Current data size
        uint8_t     m_buffer[PW_BUFFER_MAX]; //Current data buffer (fixed
length)
    }S_ATCmdRspTargetMemory;
```

m_address: 实际返回的数据的起始地址

m_size：实际成功读取到的数据长度

m_buffer：数据缓冲区 (**注：缓冲区为固定长度**)

### 1.4.2.7 擦除目标芯片

ATCmdEraseTarget

**调用参数**：S_ATCmdObject

**返回值**：ATCmdStatusOK、ATCmdStatusError。

:::tip 提示

擦除目标芯片的时间不确定，在执行此命令时，需要预留足够的等待时间。

:::

### 1.4.2.8 擦除目标芯片扇区

ATCmdEraseTargetSectors

**调用参数**：ATCmdEraseTargetSectors

```
    // Erase target chip
    typedef struct S_ATCmdReadTargetMemory{
        uint32_t    m_address;          //Target chip address, support such
as SRAM、Flash、EEPROM、OTP etc...
        uint32_t    m_size;             //Data size of a single read (Must be
less than 256)
    }S_ATCmdReadTargetMemory,
    S_ATCmdEraseTargetSectors;
```

m_address: 擦除目标芯片的地址（可以擦除Flash，EEPROM等带扇区表的存储空间）

m_size：擦除长度

**返回值**：ATCmdStatusOK、ATCmdStatusError。

Copyright ©2017-2020 All Rights Reserved. 创芯工坊科技（深圳）有限公司

擦除的地址对齐到扇区起始地址，擦除整数个扇区是最优解，如果起始地址或者结束地址不对齐，**将会多擦除起始地址扇区(向下取整)，以及多擦除尾部一个扇区（向上取整）。**

擦除目标芯片的时间不确定，在执行此命令时，需要预留足够的等待时间。

:::

### 1.4.2.9 写入目标芯片数据

ATCmdWriteTargetMemory

**调用参数**：S_ATCmdWriteTargetMemory

```
    // Respone target chip memory
    // Write target chip memory
    typedef struct S_ATCmdTargetMemory{
        uint32_t   m_address;               //Current data address
        uint32_t   m_size;                  //Current data size
        uint8_t    m_buffer[PW_BUFFER_MAX]; //Current data buffer (fixed
length)
    }S_ATCmdRspTargetMemory,
    S_ATCmdWriteTargetMemory;
```

m_address: 实际写入的数据的起始地址

m_size：实际写入读取的数据长度

m_buffer：数据缓冲区 (**注：缓冲区为固定长度**)

**返回值：**ATCmdStatusOK、ATCmdStatusError。

:::tip 提示

写入数据支持写入SRAM，写入Flash、EEPROM、OTP等，**其他区域写入未经充分测试。**

:::

### 1.4.2.10 读取目标芯片选项字

ATCmdReadTargetOptionByte

**调用参数**：S_ATCmdReadTargetOptionByte

**返回值：**ATCmdRspTargetOptionByte、ATCmdStatusError。

### 1.4.2.11 返回目标芯片选项字

ATCmdRspTargetOptionByte

**调用参数**：ATCmdReadTargetOptionByte

**返回值：**

```
        // Respone target chip option byte
    typedef struct S_ATCmdRspTargetOptionByte{
        uint32_t    m_OBsize;                   //Option byte size
        uint8_t     m_OBData[PW_OB_MAX];        //Option byte data buffer
    }S_ATCmdRspTargetOptionByte;
```

Copyright ©2017-2020 All Rights Reserved. 创芯工坊科技（深圳）有限公司

m_OBsize：选项字节的长度

m_OBData：选项字节缓冲区 (**目前是固定长度,后续会优化**)

#### 1.4.2.12 写入目标芯片默认选项字

ATCmdWriteTargetVendorOptionByte

**调用参数**：S_ATCmdWriteTargetVendorOptionByte

**返回值**：ATCmdStatusOK、ATCmdStatusError。

:::caution 注意

写入选项字，数据来源于离线项目 (pkg)中数据，必须确保先预载

:::

#### 1.4.2.13 写入目标芯片用户选项字

ATCmdWriteTargetUserOptionByte

**调用参数**：S_ATCmdWriteTargetUserOptionByte

**返回值**：ATCmdStatusOK、ATCmdStatusError。

:::caution 注意

写入选项字，数据来源于离线项目(pkg)中数据，必须确保先预载

:::

### 1.4.3 离线操作指令：

#### 1.4.3.1 查询离线项目信息

ATCmdGetProjectInfo

**调用参数**：S_ATCmdGetProjectInfo

**返回值**：ATCmdRspProjectInfo、ATCmdStatusError。

#### 1.4.3.2 返回离线项目信息

ATCmdRspProjectInfo

**调用参数**：ATCmdGetProjectInfo

**返回值**：S_ATCmdRspProjectInfo。

```
// Respone offline project infomation
    typedef struct S_ATCmdRspProjectInfo{
        uint8_t     m_activate;         // 0 : Inactivate 1: Activate
        uint32_t    m_projectSize;      // Offline project size
        uint32_t    m_projectcrc32;     // Offline project crc32
    }S_ATCmdRspProjectInfo;
```

m_activate： 0 表示无效，1 表示有效

Copyright ©2017-2020 All Rights Reserved. 创芯工坊科技（深圳）有限公司

m_projectcrc32： 离线项目的CRC32 校验值

### 1.4.3.3 加载离线项目

ATCmdLoadProject

**调用参数**：S_ATCmdLoadProject

```
//Load offline project to PowerWriter
    typedef struct S_ATCmdLoadProject
    {
        uint8_t         m_password[PW_PROJECT_PWD_SIZE]     //Project password
        uint32_t        m_projectSize;                      //Project Size
        uint32_t        m_projectCRC32;                     //Project crc32
    }S_ATCmdLoadProject;
```

m_password: 项目文件密码

m_projectSize：项目文件大小

m_projectCRC32：项目文件的CRC32 校验值

**返回值**：ATCmdStatusOK、ATCmdStatusError。

:::danger 危险

- 项目文件密码最长为16个字符，如果**密码短于16 字节，剩余的部分请用 0x00 填充。**
- 项目文件密码传输是敏感数据，对安全有需求的用户，请开启 AT 通讯加密功能（**加密算法为 AES128 CBC**）。

:::

### 1.4.3.4 加载离线项目发送

ATCmdLoadProjectSend

**调用参数**：S_ATCmdLoadProjectSend

```
//Load offline project to PowerWriter send
    typedef struct S_ATCmdLoadProjectSend
    {
        uint32_t        m_offset;                           //offset of project
        uint32_t        m_Size;                             //currrent acitvate
        uint8_t         m_data[PW_PACKAGE_SIZE];            //data buffer
    }S_ATCmdLoadProjectSend;
```

m_offset：当前发送数据的偏移地址。

m_Size：当前发送的长度。

m_data：当前数据缓冲区。

**返回值**：ATCmdStatusOK、ATCmdStatusError。

Copyright ©2017-2020 All Rights Reserved. 创芯工坊科技（深圳）有限公司

#### 1.4.3.5 禁用离线项目

ATCmdDisableProject:

**调用参数**：S_ATCmdDisableProject

**返回值**：ATCmdStatusOK、ATCmdStatusError。

#### 1.4.3.6 启动离线烧录

ATCmdStartOffline:

**调用参数**：S_ATCmdStartOffline

**返回值**：ATCmdStatusOK、ATCmdStatusError。

#### 1.4.3.7 查询离线状态

ATCmdGetOfflineStatus

**调用参数**：S_ATCmdGetOfflineStatus

**返回值**：ATCmdStatusOK、ATCmdStatusError、ATCmdStatusProgress。

### 1.4.4 测试与生产

#### 1.4.4.1：运行FactoryTest SRAM 固件

ATCmdFactoryRunSRAMFW

**调用参数**：S_ATCmdFactoryRunSRAMFW

**返回值**：ATCmdStatusOK、ATCmdStatusError。

#### 1.4.4.2：运行FactoryTest Flash 固件

ATCmdFactoryRunFlashFW

**调用参数**：S_ATCmdFactoryRunFlashFW

**返回值**：ATCmdStatusOK、ATCmdStatusError。

### 1.4.5 扩展指令

#### 1.4.5.1 广播

ATCmdBroadcast

**调用参数**：S_ATCmdBroadcast

```
typedef enum S_ATCmdBroadcastDirection{
        DIR_CDC2UART,                                   //Forwarding from USB to UART
        DIR_UART2CDC,                                   //Forwarding from UART to USB

        _DIR_MAX_ = UINT32_MAX
    }S_ATCmdBroadcastDirection;
```

Copyright ©2017-2020 All Rights Reserved. 创芯工坊科技（深圳）有限公司

```
    {
        uint8_t                 m_keepATFrame;              //0 : Forwarding valid data, 1: forwarding Full AT frame data
        S_ATCmdBroadcastDirection   m_dirType;              //Direction
        uint32_t                m_size;                     //Activate size
        uint8_t                 m_data[PW_PACKAGE_SIZE]     //data
    }S_ATCmdBroadcast;
```

m_keepATFrame：0表示只转发有效数据，1 表示转发完整的AT 帧结构（**在其他系统上也需要解析AT 指令**）

m_dirType：广播的方向。

m_size：当前发送数据的长度

m_data：数据缓冲区

**返回值：**ATCmdStatusOK、ATCmdStatusError。

:::info 补充信息

- 为何要指定传输路径：部分产品上只有USB 和 UART 两个端口，但是其他产品上可能有多个转发端口，这是为了兼容考虑。

:::

## 1.4.6 调试指令

暂不提供(如有需要，请反馈给我们)...

## 1.4.7 IO控制指令

暂不提供(如有需要，请反馈给我们)...

## 1.4.8 状态

### 1.4.8.1 返回Status Ok

**调用参数**：主流无数据返回的命令，都返回Ok 响应。

**返回**：S_ATCmdObject

### 1.4.8.2 返回Status Error

**调用参数**：**所有命令都有可能返回** ATCmdStatusError，不再详细描述。

**返回**：

```
typedef struct S_ATCmdObject
{
    union powerwriter_at_type
    {
        uint32_t m_errorcode;   // Status error code
    }property;
} S_ATCmdStatusError;
```

Copyright ©2017-2020 All Rights Reserved. 创芯工坊科技（深圳）有限公司

### 1.4.8.3 返回Status Progress

**调用参数**：ATCmdGetOfflineStatus。

**返回**：

```
//ATCmdStatusProgress
typedef struct S_ATCmdStatusProgress{
    uint32_t    m_total;          //Total size
    uint32_t    m_processed;      //Processed size
}S_ATCmdStatusProgress;
```

m_total：处理总长度

m_processed：已处理的长度

:::tip 提示

**下载本页PDF文件**

:::

Copyright ©2017-2020 All Rights Reserved. 创芯工坊科技（深圳）有限公司