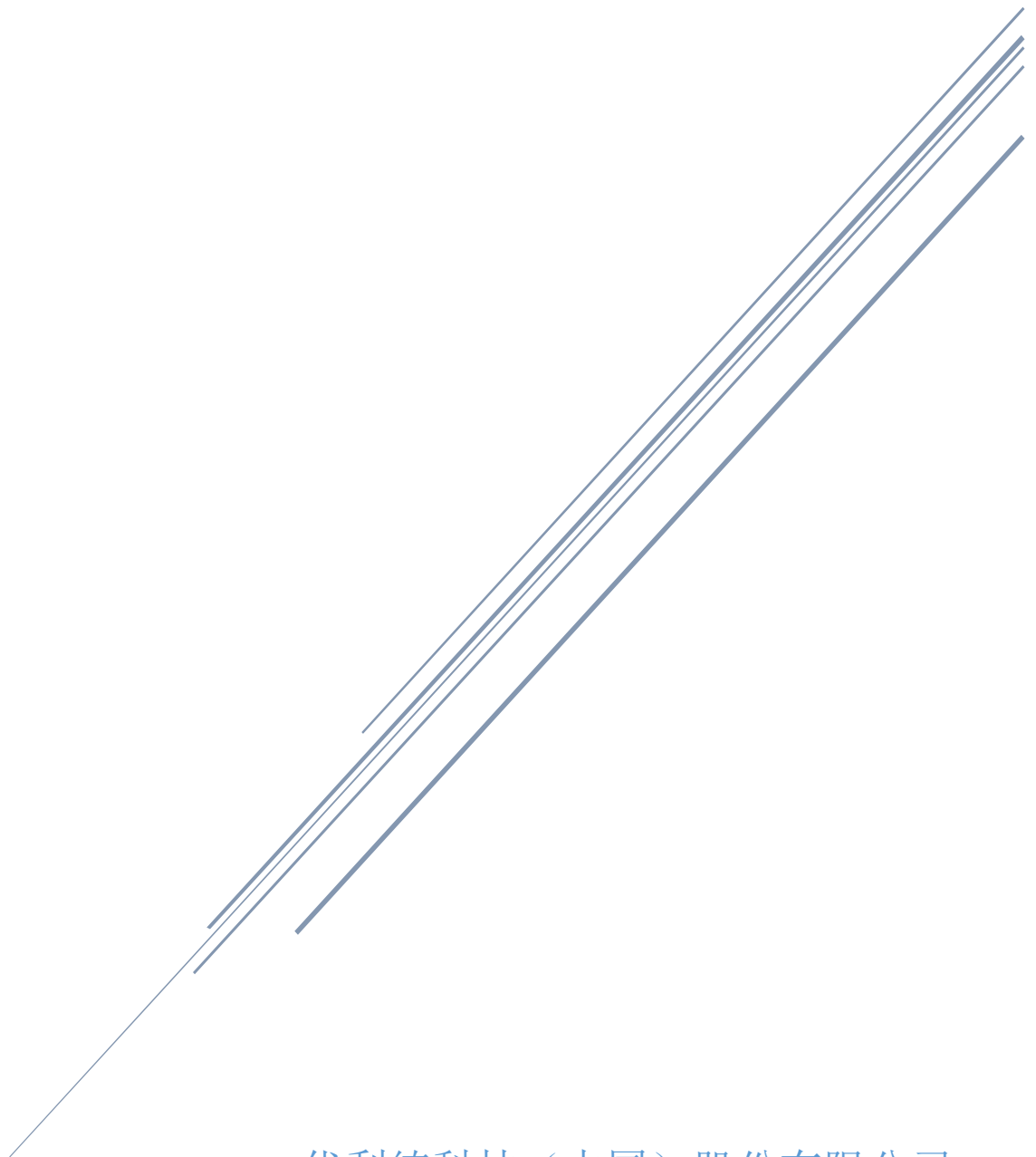


UTP3000C&UTP8000M 系列电源

编程手册



优利德科技（中国）股份有限公司

版本:

版本	修改项	日期
V1.0	正式版。	2016 年 6 月 18 日
V1.1	修改部分说明，新增示例。	2018 年 8 月 6 日

声明:

- 1、可直接使用 UCI 接口收发数据（USB\COM），也可以自己开发串口和 USB 驱动程序来下发指令，指令是明文下发的。
- 2、关于 UCI 接口的使用，请查看 doc\UCI 帮助文档.pdf 文件的介绍。

UCI 接口:

如果安装了 SDK\driver\DriverPack_Libusb 下的 USB 驱动程序，则电源被识别为 USB 设备；
如果安装了 USB 转串口驱动程序（可以使用通用的 USB 转串口驱动），则识别为 COM 口。

其中:

USB 通信:

可以使用 uci_QueryNodes 接口查找到该设备，也可以直接使用下面的 UCI 地址：

[C:POWER][D:P330XC][T:USB][PID:0x1234][VID:0x5345][EI:0x81][EO:0x2][CFG:1][I:1]

或

[C:POWER][D:P330XM][T:USB][PID:0x1234][VID:0x5345][EI:0x81][EO:0x2][CFG:1][I:1]

COM 通信:

目前不支持通过 uci_QueryNodes 来查询设备。但可以使用下面的 UCI 地址。

[T:COM][C:POWER][D:P3303C][PORT:2][BAUD:9600][PARITY:N][STOP:1][DATA:8]

其中可调参数为:

PORT： 串口端口号；

BAUD： 波特率；

PARITY： 校验方式 N： NONE； E：偶校验； O：奇校验；

STOP： 停止位；

DATA： 数据位。

也可以使用默认的配置:

地址简化为： [T:COM][C:POWER][D:P3303C][PORT:2]

对应的配置为： [BAUD:9600][PARITY:N][STOP:1][DATA:8]

打开设备：

使用 `uci_Open` 函数，传入上面的 UCI 地址，打开设备，返回会话 ID。

写指令：

直接使用 `uci_WriteX` 或 `uci_FormatWrite` 接口，下发字符串。

读参数（查询）：

先使用 `uci_WriteX` 写查询指令到设备；

然后使用 `uci_ReadX` 读取回复数据，接口超时设置 100ms 即可。

比如查询 IDN？

使用 `uci_Read` 读数据，该接口中的 `PRParams` 可以为空。只使用 `u_byte* _data, u_size _dataLen` 两个参数来读取数据流。

示例：

声明：

以下代码只做逻辑流程演示使用，不关心错误处理等细节。实际使用时，只需要一次打开，一次关闭，无需每次读写都要打开和关闭设备。

```
u_session m_session;
```

```
u_status r = uci_Open("[C:POWER][D:P330XC][T:USB][PID:0x1234][VID:0x5345][EI:0x81][EO:0x2][CFG:1][I:1]",
    &m_session, 1000);
```

```
auto r = uci_WriteX(m_session, "*IDN?", 1000, NULL, 0);
```

```
char buf[20];
```

```
ZeroMemory(buf, sizeof(buf));
```

```
auto r = uci_ReadX(m_session, "*IDN?", 100, buf, sizeof(buf)); //读取时的_msg可以为空！
```

```
uci_Close(m_session);
```

```
/*
```

```
//@brief          : 读取数据
```

```
//@param _session : 会话ID
```

```
//@param _msg      : 命令字符串，以'\0'结尾。具体格式见文档；
```

```
//@param _timeout  : 读取超时。
```

```
//@param _data     : 接收读取到的数据的缓冲区地址。
```

```
//@param _dataLen  : _data指向的缓冲区长度（Bytes）
```

```
//@return : 查看本文件中的“关于接口返回的状态值”说明块
//@remarks : 本接口是uci_Read的简化版本
u_status _UCIAPI uci_ReadX(u_session _session, u_cstring _msg,
u_uint32 _timeout, u_byte* _data, u_size _dataLen);
*/
```

支持的指令有:

1、 ISET<X>:<NR2>

描述: 设置电流输出。

示例: ISET1:2.225

设置 CH1 的电流输出为 2.225A

2、 ISET<X>?

描述: 回读指定通道的当前电流设置值。

示例: ISET1?

回读 CH1 的当前电流设置值。

3、 VSET<X>:<NR2>

描述: 设置指定通道的电压值。

示例: VSET1:20.50

设置 CH1 的电压值为 20.50V

4、 VSET<X>?

描述: 回读指定通道的电压设置值。

示例: VSET1?

回读 CH1 的电压设置值。

5、 IOOUT<X>?

描述： 回读指定通道实际输出电流值。

示例： IOOUT1?

回读 CH1 的当前实际输出的电流值。

6、 VOUT<X>?

描述： 回读指定通道的实际输出电压。

示例： VOUT1?

回读 CH1 的实际输出电压值。

7、 TRACK<NR1>

描述： 选择工作模式： 独立、串联和并联。

NR1 取值有：

0：独立

1：串联

2：并联

示例： TRACK0

8、 BEEP<Boolean>

描述： 打开/关闭蜂鸣器响

示例： BEEP1 打开蜂鸣器。

9、 OUT<Boolean>

描述： 打开/关闭通道输出

参数：0：OFF；1：ON

示例：OUT1 打开通道输出

10、 STATUS?

描述：查询设备 8 位状态码，状态码格式说明如下：

位置	位数	意义
D0	1	CH1: 0 = CC, 1 = CV
D1	1	CH2: 0 = CC, 1 = CV
D2~D3	2	00=独立模式, 01=跟踪模式, 10=并联
D4	1	OVP: 0 = OFF, 1 = ON
D5	1	OCP: 0 = OFF, 1 = ON
D6	1	CH1 : 0 = CH1 输出关闭, 1 = CH1 输出打开
D7	1	CH2 : 0 = CH2 输出关闭, 1 = CH2 输出打开

11、 *IDN?

描述：返回设备名称。

示例：*IDN? 结果：P330XM

12、 RCL<NR1>

描述：回调已存储的设置信息。

参数：NR1 1 - 5：存储位置，1~5。

示例：RCL1 回调存储位置位 1 的记忆信息。

13、 SAV<NR1>

描述：记忆设置数据

参数：NR1 1 - 5：存储位置，1~5

示例： SAV1 将当前设置操作存储到位置 1.

14、 OCP< Boolean >

描述： 打开/关闭 OCP 功能

参数： 0 = OFF, 1 = ON.

示例： OCP1 打开 OCP 功能。

15、 OVP< Boolean >

描述： 打开/关闭 OVP 功能

参数： 0 = OFF, 1 = ON

示例： OVP1 打开 OVP 功能。

16、 OCPSTE: <X>:<NR2>

描述： 设置当前过流保护值

示例： OCPSTE1: 5.100

17、 OVPSTE: <X>:<NR2>

描述： 设置当前过压保护值

示例： OVPSTE1: 31.00