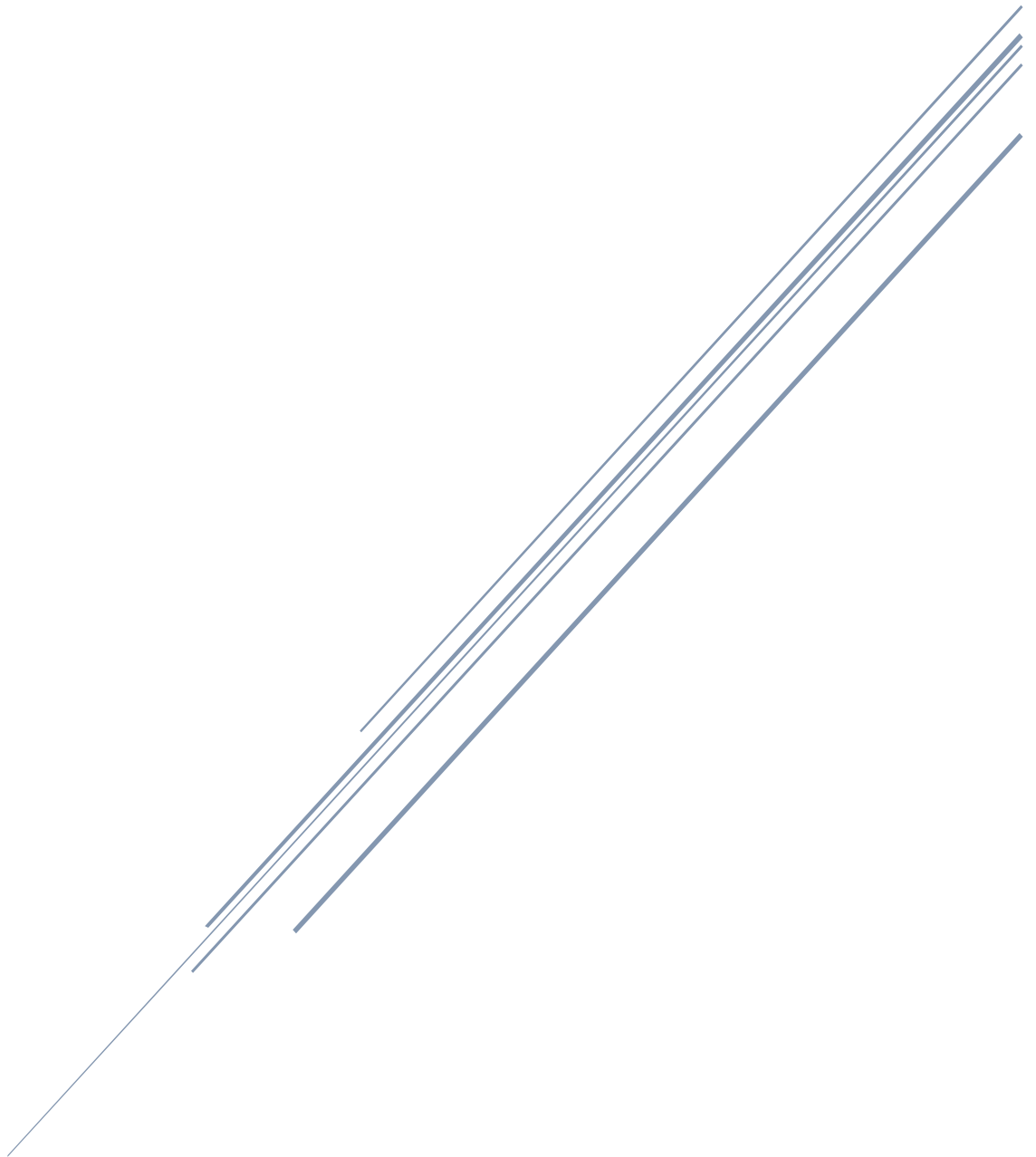


UTG2000A&UTG7000B 系列信号源

编程手册



优利德科技（中国）有限公司

版本:

版本	修改项
V1.0	正式版

简介:

使用的接口为 UCI 接口，UCI 接口的使用相关问题查看《UCI 帮助文档》。

具体操作可以参照所给的示例工程。

引用文件:

- 1、UTG2025Def.h :
给出该系列信号源的基础定义。
- 2、UCI 相关文件:
见《UCI 帮助文档》。

命令字符串基本格式:

“
命令名 1: 命令参数@属性 1:属性值@属性 2:属性值 。 。 。 @属性 n:属性值;
命令名 2: 命令参数@属性 1:属性值@属性 2:属性值 。 。 。 @属性 n:属性值;
。 。 。 。 。 。
命令名 n: 命令参数@属性 1:属性值@属性 2:属性值 。 。 。 @属性 n:属性值;
”

说明:

- 1、对字符大小写不敏感;
- 2、数值支持十六进制、八进制、十进制格式;
- 3、支持多条语句（具体视机型而定）
如果多条语句或者多个属性执行失败，请尝试使用单挑语句和单个属性;
- 4、每个语句必须以';'结尾;
- 5、名称、值和各标记间支持空格存在;

示例:

“wp@ch:0@addr:10@v:10;”
“key:c1;”

术语： SG - 信号源简称

通用命令

命令名	意义	I/O	数据	备注
Local	Lock Pad (锁键盘)	W	Enum:0/1 {远程状态/本地状态}	远程状态时设备键盘锁。
Local?	查询键盘是否锁定	R	Enum:0/1 {未锁定/已锁定}	
Lock?	查询所有按键的锁定状态	R	8Bytes, 64 有符号整数, 标记位	

写参数:

命令名	命令参数	命令参数类型
wp	无	无

属性名	意义	I/O	数据
CH	通道号	W	Enum(Integer) : 0/1 { CH1/ CH2 }
addr	参数地址	W	Enum(ParamNo): 查看 参数地址 定义。
v	参数值	W	一般以最小单位为量度的数值

示例:

"wp@CH:0@addr:9@v:1000;" - 设置 CH1 的频率为 1mHz;

注意:

UCI 接口对应: `uci_Write`

读参数：

命令名	命令参数	命令参数类型
rp	无	无

属性名	意义	I/O	数据
CH	通道号	W	Enum(Integer) : 0/1{ CH1/ CH2 }
addr	参数地址	W	Enum(ParamNo): 查看 参数地址 定义。

示例：

"rp@CH:0@addr:9;" - 读取 CH1 的频率；

注意：

UCI 接口对应： `uci_Read`， 对应的数据量是 8 个字节，double 类型！

按键

命令名	命令参数	命令参数类型
KEY	键值	见下面的键码表

KEY	字符编码	KEY	字符编码
底部功能键 1	AF1	0	0
底部功能键 2	AF2	1	1
底部功能键 3	AF3	2	2
底部功能键 4	AF4	3	3
底部功能键 5	AF5	4	4

底部功能键 6	AF6	5	5
F1	F1	6	6
F2	F2	7	7
F3	F3	8	8
F4	F4	9	9
Menu	MENU	.	.
Knob Left	FKNL	+/-	SIGN
Knob Right	FKNR	Trigger	TG
Knob Click	FKN	Utility	UTIL
Left	L	CH1	C1
Right	R	CH2	C2

属性名	意义	IO	数据
Lock	锁定该按键	W	无数据
Unlock	解锁该按键	W	无数据
Lock?	查询该按键锁状态	R	Integer<4Bytes>: 0 - 未锁; 1 - 已锁

示例:

```

“KEY:c1;”    -- CH1
“KEY:c2;”    -- CH2
“KEY:c2@lock;” -- CH2 按键锁
“KEY:c2@unlock;” -- CH2 按键解锁
“KEY:c2@lock?;” -- 查询按键是否锁

```

注意:

带问号的指令需要用 **uci_Read** 读取。状态通过接口返回值获取。

读写配置文件

命令名	命令参数	命令参数类型
dconfig	无	无

示例:

“dconfig;”

注意:

读取使用 `uci_Read` 接口, 缓冲区大小可设置为 1024Bytes, 实际有效数据量, 以接口返回值为准。

写配置数据使用 `uci_WriteFromFile` 接口, 不识别文件后缀, 只识别“dconfig;”命令, 写配置超时至少 6s。

抓取屏幕:

命令名	命令参数	命令参数类型
PrtScn	图像格式	Enum(String):空/zip/bmp {非压缩像素数据/压缩的像素数据/BMP 文件数据}

示例:

“PrtScn:bmp;” --- 截取屏幕保存为 bmp 文件数据;

“PrtScn;” --- 截取屏幕保存为像素数据;

“PrtScn:zip;” --- 截取屏幕保存为压缩像素数据;

注意:

使用 `uci_Read` 读取数据, 这个命令并没有将数据直接保存的文件, 而是返回到 `uci_Read` 指定的缓冲区里面。

如果要缓冲到本地文件, 请自行保存。

- 1、如果使用命令: “PrtScn;”, 则缓冲区大小必须是 $\geq 391680(480 \times 272 \times 3)$, 读到的是 24 位的像素数据;
- 2、如果使用命令: “PrtScn:bmp;”, 则缓冲区大小必须是 $480 \times 272 \times 3 + 54 = 391734$, 即位图文件大小。
- 3、如果使用命令: “PrtScn:zip;”, 则缓冲区大小可设置为 $\geq 391680(480 \times 272 \times 3)$ (最大数据量), 读到的是压缩像素数据。然后使用: `alg_UnCompressPixels_25` 接口解压缩数据。注意 `uci_Read` 返回的是压缩数据量。
- 4、使用 `uci_ReadToFile` 接口加 “prtsn:bmp;” 命令, 可以将位图保存到磁盘文件。

写任意波形文件：

命令名	命令参数	命令参数类型
WARB	无	无

属性名	意义	IO	数据
CH	通道号	W	Enum(Integer) : 0/1{ CH1/ CH2 }
Mode	加载模式	W	Enum(Integer): 0/1 {Carrier/Mod}

示例：

“WARB@CH:0@MODE:0;”

将波形文件以载波的形式加载到 CH1 中。

注意：

使用 `uci_WriteFromFile` 写任意波文件。超时设置为 **1000**。

附录：

1、参数地址：

数值单位：

频率设置单位： μHz ， 1mHz° 设置值为 $1*1000$ ；

电压设置单位： μV

时间设置单位： ns（ns、 μs 、ms、s） 比如设置 1s 下发值就是 $1*1000^3$

度数设置单位： 0.01° ， 比如设置相位为 90° ， 那么下发值就是 $90 * 100$

百分比设置单位： 0.01%， 比如占空比。 下发值=输入值*100

//@brief : UTG2025A 和 UTG2062A 的信号源各参数编码

```
typedef enum _ParamNo {
    Invalid_Cmd = -1,
    //@brief : 当前工作模式
    CurMode = 6,
    //@brief : 任意波播放模式
    ArbPlayMode = 7,
    //@brief : 波形类型
    WaveType = 8,
    //@brief : 频率 单位:  $\mu\text{Hz}^\circ$  即  $1\text{mHz}^\circ$ 设置值为  $1*1000$ 
    Freq = 9,
    //@brief : 周期
    Period = 10,
    //@brief : 幅度-峰峰值, 设置值以 mV 为单位
    AmpVpp = 11,
    //@brief : 幅度-有效值 单位:  $\mu\text{V}$ 
    AmpVrms = 12,
    //@brief : 幅度-功率
    AmpDbm = 13,
    //@brief : 直流偏移
    DCOffset = 14,
    //@brief : 高电平
    High = 15,
    //@brief : 低电平
    Low = 16,
    //@brief : 相位 , 设置值以 0.01 为基数, 如  $1^\circ$ 就要设置 100
    Phase = 17,
    //@brief : 占空比 单位: 0.01% 即 1%设置值为  $1*100$ 
    DutyCycle = 18,
    //@brief : 对称度 单位:  $0.01^\circ$  即  $1^\circ$ 设置值为  $1*100$ 
```

```

Symmetry = 19,
//@brief :
PulseWidth = 20,
//@brief : 脉冲波-上升沿时间
PulseRisingTime = 21,
//@brief : 脉冲波-下降沿时间
PulseFallingTime = 22,
//@brief :
ArbSampleRate = 23,
//@brief : ARB 文件名
ArbName = 24,
//@brief :
ArbLength = 25,
//@brief : 调制类型
ModulateType = 40,
//@brief : 调制频率
ModulateFreq = 28,
//@brief : 调制源
ModulateSource = 29,
//@brief : 调制波
ModulateShape = 30,
//@brief :
ModulateOnOff = 31,
//@brief : 调制深度
ModulateAMDepth = 32,
//@brief : 频偏
ModulateFMFreqDev = 33,
//@brief : 相偏
ModulatePMPhaseDev = 34,
//@brief : 跳跃频率
ModulateFSK_HopFreq = 35,
//@brief : 速率
ModulateFSKRate = 36,
//@brief : 相移键控-相位
ModulatePSKPhase = 37,
//@brief : 频移键控-载波频率
ModulateFSK_CarrierFreq = 38,
//@brief : 脉宽调制-占空比偏差
ModulatePWM_DutyDev = 39,

//@brief : 扫频-起始频率
SweepStartFreq = 47,
//@brief : 扫频-停止频率
SweepStopFreq = 48,

```

```

//@brief : 扫频-扫描时间
SweepTime = 49,
//@brief : 扫频-类型
SweepType = 50,

//@brief : 切换频率设置方式-频率或者周期
Wave_Freq_Or_Period = 51,
//@brief : 切换幅度设置方式-幅度或者高低电平
Wave_Amp_Or_HighLow = 52,
//@brief : 切换不同的幅度显示单位
Wave_Amp_Unit = 53,

//@brief : 脉冲串 - 起始相位
BurstStartPhase = 57,
//@brief : 脉冲串 - 猝发周期
BurstPeriod = 58,
//@brief : 脉冲串 - 循环数
BurstCycleCount = 59,
//@brief : 脉冲串 - 门控-极性
BurstPolarity = 60,
//@brief : 脉冲串 - 类型
BurstType = 61,

//@brief : 扫频-触发源
TriggerSource = 68,
SyncType = 69,
//@brief : 扫频-触发沿 0: 是上升沿 1: 下降沿
TriggerEdge = 70,
//@brief : 扫频-触发输出开关
TriggerOnOff = 71,

//@brief : CH1 通道使能
OutputEnable = 77,
//@brief : CH2 通道使能
OutputEnable2 = 78,
//@brief : 反相
Invert = 79,
//@brief : 输出阻抗
OutputExtRes = 80,
//@brief : 幅度限制
OutputLimit = 81,
//@brief : 幅度上限
OutputLimitHigh = 82,
//@brief : 幅度下限

```

```
OutputLimitLow = 83,  
//@brief : 同步输出  
SyncOut = 84,  
//@brief :  
Couple = 85,  
//@brief : IP 地址类型  
IPAddrType = 101,  
//@brief : IP 地址  
IPAddr = 102,  
//@brief : 子网掩码  
IPSubnetMask = 103,  
//@brief : 网关  
IPGateway = 104,  
//@brief : DNS  
IPDns = 105,  
//@brief : MAC 地址  
Mac = 106,  
//@brief : 语言  
Language = 107,  
//@brief : 时钟源  
ClkSource = 108,  
//@brief :  
PowerOnParam = 109,  
//@brief : 蜂鸣器开关  
BeepOnOff = 110,  
//@brief : 数字分隔符  
CharDivideInt = 111,  
//@brief : 载入设置  
LoadSettting = 112,  
//@brief : 保存设置  
SaveSetting = 113,  
//@brief : 背光调节  
Backlight = 114,  
//@brief : 关于  
About = 115,  
//@brief : 时钟输出  
ClkOut = 116  
}ParamNo;
```

2、按键锁定状态标记位：

//@brief : 查询UTG2025A 和 UTG2062A的信号源按键锁定状态时，存放于64位整数中设置位。

```
typedef enum _UTG2025Key_bit {
    IKEY_Unknown = -1,
    //@brief : 底部功能按键1
    IKEY_AF1 = 0,
    //@brief : 底部功能按键2
    IKEY_AF2,
    //@brief : 底部功能按键3
    IKEY_AF3,
    //@brief : 底部功能按键4
    IKEY_AF4,
    //@brief : 底部功能按键5
    IKEY_AF5,
    //@brief : 底部功能按键6
    IKEY_AF6,

    //@brief : 功能键1
    IKEY_F1,
    //@brief : 功能键2
    IKEY_F2,
    //@brief : 功能键3
    IKEY_F3,
    //@brief : 功能键4
    IKEY_F4,

    //@brief : Menu键
    IKEY_MENU,

    //@brief : 数字0键
    IKEY_NUM_0 = 14,
    //@brief : 数字1键
    IKEY_NUM_1,
    //@brief : 数字2键
    IKEY_NUM_2,
    //@brief : 数字3键
    IKEY_NUM_3,
    //@brief : 数字4键
    IKEY_NUM_4,
    //@brief : 数字5键
    IKEY_NUM_5,
    //@brief : 数字6键
```

```

IKEY_NUM_6,
//@brief : 数字7键
IKEY_NUM_7,
//@brief : 数字8键
IKEY_NUM_8,
//@brief : 数字9键
IKEY_NUM_9,
//@brief : 数字键盘-符号(+/-)键
IKEY_SIGN,
//@brief : 数字键盘“.”键
IKEY_DOT,
//@brief : 旋钮右旋
IKEY_PLUS,
//@brief : 旋钮左旋
IKEY_MINUS,
//@brief : 方向→键
IKEY_RIGHT,
//@brief : 方向←键
IKEY_LEFT,
//@brief : Trigger键
IKEY_TRIGGER,
//@brief : CH1键
IKEY_CH1,
//@brief : CH2键
IKEY_CH2,

//@brief : UTILITY键
IKEY_LOCAL =34,
//@brief : 功能旋钮按下(Enter/OK功能)
IKEY_OK,
}UTG2025KeyBit;

```