

KingstVIS 软件基于 Socket 通信的 API 说明

目前基于 Socket 通信的 API 尚未完全开放，仅实现了部分功能，但现已实现的接口命令已经过多方测试，可确保其可用性。Socket 是一种通用的接口，其编程实现不依赖于特定语言，主流编程语言皆可支持，我们在此提供了基于 Python 的实例代码，以供开发者参考与测试。其中的 `SocketAPI-Test.py` 文件可用于手动测试各命令的执行，`SocketAPI-AutoSave.py` 文件是一个自动重复采集并保存数据的实例。

基于 Socket API 的二次开发需要具备专业编程技能的人员依据本说明文档进行，恕我们不再提供额外的技术支持。

一、启用对 Socket API 的支持

KingstVIS 软件默认是没有启用对 Socket API 功能的支持的，需通过修改软件的配置文件来启用支持，各系统版本的配置文件所在位置如下：

- Windows 版：C:\Users\用户名\AppData\Local\kingst\vis.config
- Mac OSX 版：/Users/用户名/Library/Application Support/kingst/vis.config
- Linux 版：/home/用户名/.local/share/kingst/vis.config

配置文件采用标准 XML 格式，可以使用任意文本编辑软件打开。

- (1) 在关闭 KingstVIS 软件的状态下，打开配置文件；
- (2) 查找“enaSocket”项，将其值改为“1”即可启用 API 功能；

再查找“listenPort”项，指定监听端口号，默认值为 23367，可留空而不做修改；
如下图所示：



```
<?xml version="1.0" encoding="UTF-8"?>
<settings>
  <global>
    <version>3.4.3</version>
    <>windowPosition>384,208,3072,1664</windowPosition>
    <viewState/>
    <disAnimateZoom/>
    <autoWaitTrig/>
    <colorWaveform/>
    <smpProgPosition/>
    <userLang>zh-CN</userLang>
    <devModel>8</devModel>
    <>windowMaximized>0</windowMaximized>
    <anlyDbgPath/>
    <enaSocket>1</enaSocket>
    <listenPort/>
    <oscCursorEna/>
    <lgeMeasure/>
```

- (3) 保存、关闭配置文件；
- (4) 重新启动 KingstVIS 软件，即可通过 Socket 编程控制软件的部分功能。
- (5) 软件重启后可能会申请防火墙权限，请允许即可。

二、 命令综述：

Socket API 使用 TCP 连接，软件默认监听 23367 端口。

Socket API 命令使用与 Linux Shell 命令类似的风格。命令及参数均使用小写字母与中划线的组合，命令及参数之间用空格符隔开，如某项参数中包含空格，则需将该参数用双引号括起来。[命令及参数均区分大小写](#)。

软件接收到一个命令后，会首先检查是否支持该命令及命令参数是否正确，如命令有误则直接返回“NAK”，如命令无误则尝试执行该命令。如命令无法执行或执行过程中发生错误，则返回“NAK”，如命令正确执行完毕，则返回“ACK”。某些读取类命令（如获取当前采样率 get-sample-rate）会返回数据给用户，这些数据将跟在“ACK”之后，“ACK”与之后的数据用空格符隔开，多个数据之间同样用空格符隔开。

如果某命令返回“NAK”，用户可以通过特殊命令 [get-last-error](#) 获取最后的错误信息。如果某命令返回“ACK”后再执行 [get-last-error](#)，得到的是最后一次未正确执行的命令的错误信息，而不是当前命令的执行信息。

三、 各命令详述：

1、 [get-last-error](#)

获取最后一次出错时的错误信息。该命令的返回信息可能有以下几种：

- **no error**
无错误，自软件启动以来尚未发生任何错误
- **command is too long**
命令超长，命令+参数的总长度需限制在 2048 个字符以内
- **invalid command**
无效的命令
- **invalid parameter**
无效的参数
- **save file failed**
文件保存失败，可能由于文件名及路径错误或没有写入权限而导致
- **sampling has been started**
采样已经开始，在一次采样结束前再次收到启动采样命令时返回该信息
- **device unconnected, cannot start sample**
硬件设备未连接，不能启动采样
- **device connected, cannot start simulate**
硬件设备已连接，不能启动模拟采样
- **sampling in progress, cannot execute the command**
采样正在进行中，不能执行当前命令
- **there is no data to save or export**
没有有效的采样数据供保存或导出

2、 start [--simulate]

启动采样。接收到该命令后，软件会立即启动一次采样，如采样正确完成则返回"ACK"，如采样过程中发生错误，则返回"NAK"。该命令在采样结束后才会发回响应。

--simulate: 可选参数。

不带此参数时，软件将启动一次真实的采样，如当前未连接硬件设备，则直接返回"NAK"而不启动采样。

带上此参数时，软件将启动一次模拟采样，如当前已连接好硬件设备，则直接返回"NAK"而不启动采样。

设计此参数的目的是为了在用户编程实现自动化处理的过程中，能够随时得知本次采样数据是否真实有效，因为硬件设备在使用过程中有可能发生故障而导致软件失去连接，此时软件会进入模拟状态，如果不加区分则将得到模拟出的虚假的采样数据。

3、 stop

终止当前采样。如目前正在采样，则立即终止本次采样并返回"ACK"，如当前并未进行任何采样则直接返回"ACK"，该命令返回的"ACK"表示当前已经不处在采样状态了。

该命令执行时，如果前一次发送的 start 命令尚未执行完成，则该命令的执行将导致之前的 start 命令不再发回响应。

4、 set-sample-rate 采样率数值

命令后跟随要设定的采样率数值，比如要设置 10M 采样率，需输入：

set-sample-rate 10000000

如果输入的采样率不被当前设备所支持，则返回"NAK"。

5、 get-sample-rate

获取当前采样率，比如当前采样率为 5M，则返回：

ACK 5000000

6、 get-supported-sample-rate

获取当前设备所支持的所有采样率，比如 LA1010 会返回：

ACK 100000000 80000000 50000000 40000000 32000000 25000000 20000000
16000000 12500000 10000000 8000000 5000000 4000000 2000000 1000000 500000
200000 100000 50000 20000

7、 set-sample-depth 采样深度数值

命令后跟随要设定的采样深度数值，比如要设置 20M 的深度，需输入：

set-sample-depth 20000000

如果输入的采样深度超出可支持的范围，则返回"NAK"。比如 LA1010 支持的深度范围为 20K ~ 100G，即 20000 ~ 1000000000000。

8、 get-actual-sample-depth

获取实际的采样深度。一次采样完后实际的采样深度有可能不等于设定的深度（存储器提前耗尽或采样过程中出错），该命令用于获取实际的采样深度。比如返回：

ACK 15000000

表示本次采样的实际深度为 15M。

9、 set-sample-time 采样时间数值

该命令可以让用户使用时间参数来设定采样深度，时间参数使用以秒为单位的数值，比如要设置 0.5s 的时间，需输入：

set-sample-time 0.5

如果时间参数转换为采样深度后超出可支持的范围，则返回"NAK"。

10、 get-actual-sample-time

获取实际的采样时间。采样结果以实际的触发位置为时间的参考零点，向前为负值，向后为正值。该命令返回 2 个参数，即起始点和结束点所对应的实际采样时间。比如返回：

ACK -0.05 0.45

表示本次采样的实际时间为-0.05 ~ 0.45s

11、 set-threshold-voltage 阈值电压数值

命令后跟随要设定的 IO 阈值电压，比如被测信号为 3.3VCMOS 标准，则需输入：

set-threshold-voltage 1.65

设备支持的阈值电压范围为-4V ~ +4V，超出此范围将返回"NAK"。

12、 set-trigger [-reset] [触发类型 通道号]

触发类型参数包括：--low-level / --high-level / --pos-edge / --neg-edge

分别代表：低电平 / 高电平 / 上升沿 / 下降沿

在触发类型参数之后输入需要设置为该类型的通道号，可以连续输入多个通道，用空格隔开。

可以同时输入多个类型参数与通道号的组合，比如：

set-trigger --pos-edge 0 --high-level 1 2 --low-level 3 4

表示设置通道 0 为上升沿，同时设置通道 1、2 为高电平，通道 3、4 为低电平。

请注意：边沿触发类型只能设置一个通道，如果设置了多个通道，则只有最后一个通道号生效。

--reset: 该参数将重置全部通道，即清除全部已设置的触发条件。

该参数可以单独使用，如：set-trigger --reset，即可清除全部触发条件，然后再发送命令设置所需要的触发条件。

也可以跟触发类型参数组合使用，这样就可以只用一条命令完成清除原有触发条件而设置为新的触发条件，比如：

```
set-trigger --reset --pos-edge 0 --high-level 1 2 --low-level 3 4
```

表示先清除原有触发设置，然后设置通道 0 为上升沿，同时设置通道 1、2 为高电平，通道 3、4 为低电平。

13、 export-data 完整文件名 [-chn-select 通道号] [-time-span 起始时间 结束时间]

将当前的采样数据导出到文件中。

命令后第一个参数为包含路径与扩展名在内的完整文件名，命令通过扩展名来区分导出文件的类型，目前支持.kvdat / .txt / .csv / .bin。

--chn-select: 可选参数。

不带此参数时，导出全部通道的数据。

使用此参数时，其后跟随要导出的通道号，可以连续输入多个通道，用空格隔开。比如：

```
export-data "D:\save\data1.csv" --chn-select 0 1
```

表示只导出通道 0 和通道 1 的采样数据。

```
export-data "D:\save\data2.csv"
```

表示导出全部的采样数据。

--time-span: 可选参数。

不带此参数时，导出全部的采样数据。

使用此参数时，其后跟随起始和结束时间，时间以实际触发位置为参考零点。可以只给定一个时间参数，此即起始时间，结束时间将自动设定为实际采样数据的结束时间。如设定的时间参数超出实际采样所得的时间，则自动缩短至实际采样时间。比如：

```
export-data "D:\save\data01.txt" --time-span 0.01 0.5
```

表示导出 0.01s ~ 0.5s 时间内的采样数据。

```
export-data "D:\save\data02.txt" --time-span 0.01
```

表示导出 0.01s 至结束位置的采样数据。

```
export-data "D:\save\data03.txt"
```

表示导出全部的采样数据。