



南京凌鸥创芯电子有限公司

lkviewer 使用说明

© 2022, 版权归凌鸥创芯所有

机密文件，未经许可不得扩散



版本	修改时间	修改人	修改说明
1.0.0	2022-03-14	Wucf	文档创建
1.0.1	2022-04-02	Wucf	
1.0.2	2022-04-09	Wucf	修复部分功能
1.0.5	2022-06-29	Wucf	
1.0.7	2022-07-21	Wucf	开放 TCP 连接说明



目录

目录	3
1. 概述	4
2. 主界面	4
3. 配置管理	5
3.1. 配置文件	5
3.2. 新建工程	5
3.3. 打开已有工程	6
4. 设置	6
4.1. 基本设置	6
4.2. 下位机设置	7
4.3. 连接设置	7
4.3.1. JLink/ULink	8
4.3.2. 串口	8
4.3.3. 蓝牙	8
4.3.4. TCP Server	9
4.3.5. TCP Client	10
5. 变量管理	10
5.1. 变量导入	11
5.2. 自定义变量	12
6. 窗口导航	13
6.1. 基础窗口	13
6.1.1. RTT 命令行	14
6.1.2. 寄存器	16
6.1.3. 运行日志	16
6.2. 数值窗口	17
6.3. 曲线窗口	17
6.3.1. 基本曲线	17
6.3.2. 多子图显示	18
6.3.3. 导出图形	18
6.3.4. 导出数据	19
6.3.5. 导入数据	20
6.3.6. 数据查看	20
6.4. Recorder 窗口（后续版本开放）	20
6.5. Trigger 窗口	22
7. 下位机支持库	24
8. 通信协议说明	25
8.1. 读 Memory	25



8.1.1. 请求帧 25

8.1.2. 应答帧 26

8.2. 写 Memory 26

8.2.1. 请求帧 26

8.2.2. 应答帧 27

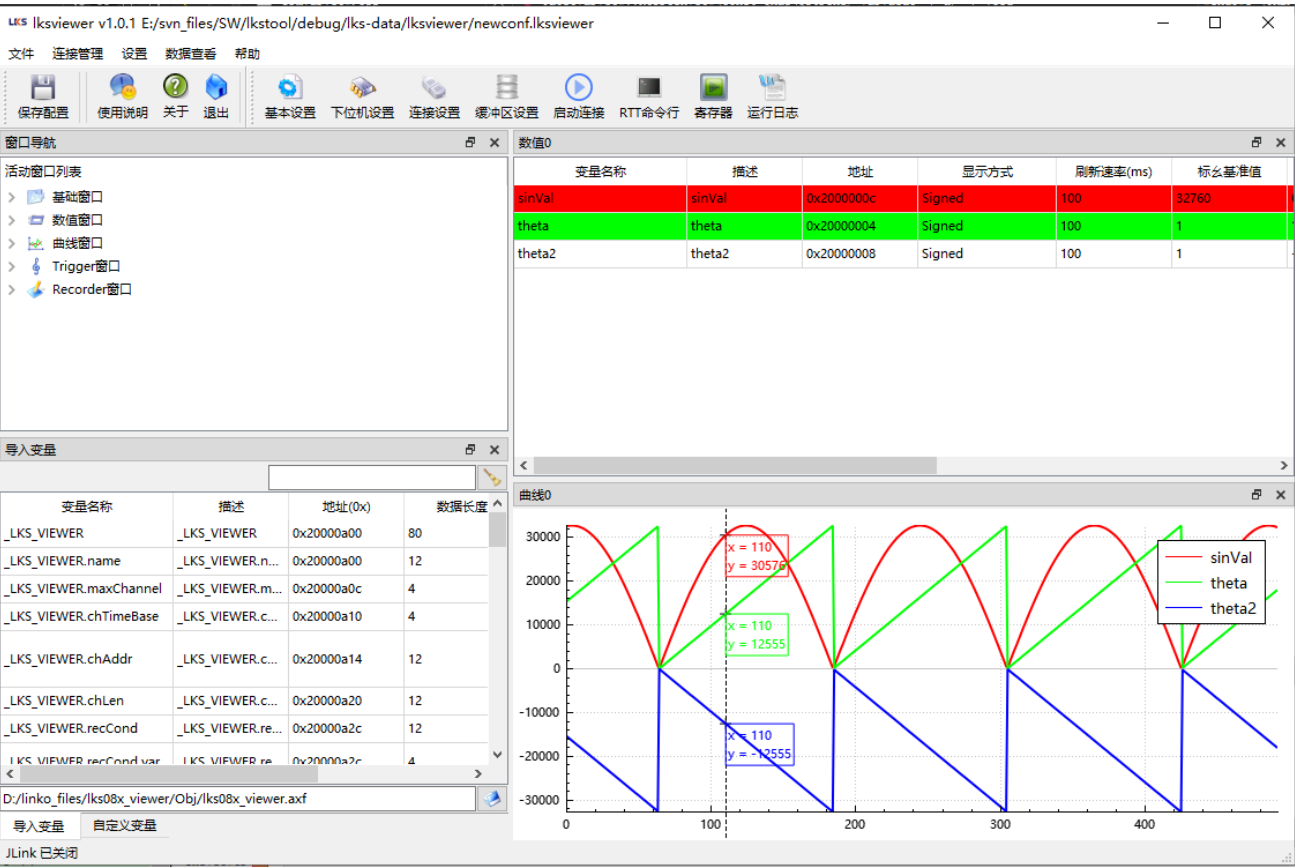
8.3. CRC 计算 27

1. 概述

lkviewer 是凌鸥公司开发的用于辅助嵌入式调试的工具，部分功能是基于 SEGGER_RTT 的二次开发，集合了 JScope 等常用调试工具的功能，并根据工程师的日常需求增加了一些实用功能。如有更多的意见或建议，可发送邮件至 wucf@linkosemi.com。

lkviewer 可以导入 axf/map/elf 文件的变量，在调试中实现数据的查看与修改、数据与波形的联动显示、数据触发捕获、寄存器读写、调试日志的输出与命令下发等。推荐使用 keil 编译的 axf 文件作为变量信息来源。

2. 主界面



上图是 lkviewer 的一个典型工作界面。其中打开了一个工程 “newconf.lkviewer”，导入了一个 .axf 文



件，用于识别变量信息。同时打开了数值查看及曲线绘制窗口，用于显示变量的当前值，以及数据曲线。

关于工程的管理及各类窗口的维护，将在后面详细介绍。

3. 配置管理

3.1. 配置文件

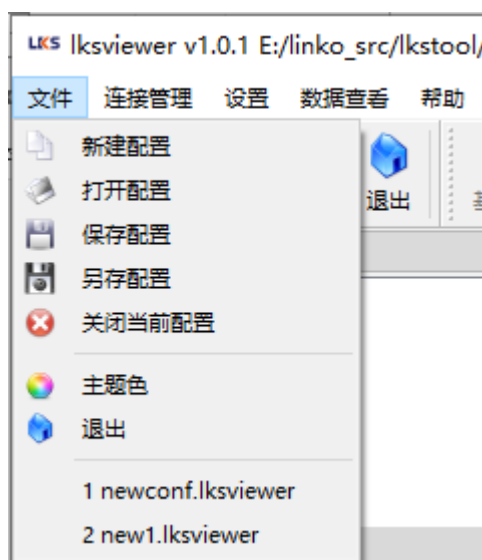
使用*.lkviewer 作为配置文件的名称，保存界面中各类子窗口和参数的配置情况，方便以后使用时恢复配置。

lkviewer 期望绑定为.lkviewer 文件的默认打开程序，所以需要修改您的注册表，请至少使用管理员权限运行一次 lkviewer.exe，以完成注册表的修改。之后，您可以双击直接打开.lkviewer 文件。如果在使用过程中您移动了 lkviewer.exe 文件的位置，则注册表需要更新，需要再执行一次管理员运行。

名称	修改日期	类型	大小
LKS new2.lkviewer	2022/3/27 11:59	LKSVIEWER 文件	1 KB
LKS newconf.lkviewer	2022/3/31 19:35	LKSVIEWER 文件	3 KB

.lkviewer 绑定打开方式

3.2. 新建工程



在开始一组调试前，您可以直接将 keil 工程下的.axf 或.uvproj(x)文件拖入到窗口左下角的列表框中，将自动解析其中的全局变量信息。然后点击保存按钮，配置文件的保存位置将默认指向 keil 工程所在的目录，方便与 keil 工程同步维护。



也可以使用菜单栏的“文件->新建配置”菜单，为要使用的配置内容指定存储文件。点击菜单栏的“文件->保存配置”菜单或工具栏的“保存配置”按钮，可以保存当前工程的设置。

3.3. 打开已有工程

点击菜单栏的“文件->打开配置”按钮，可以选择打开已保存的工程配置。同时，文件菜单下会显示最近打开的工程列表供快速打开。

4. 设置

lkviewer 的工具栏放置了一组设置按钮，用于界面显示，连接等参数的设置。

4.1. 基本设置



字体：选择 lkviewer 界面使用的字体。

默认采样周期：当您添加新的采样对象时，默认使用的采样周期。可以在数值窗口为采样对象设置独立的采样周期。

界面刷新周期：数值、曲线等显示内容的刷新周期。数据采集与刷新异步实现，刷新周期不影响数据采集。

曲线缓存测点数量：在曲线窗口中，所有的曲线数据缓存在内存中，如果采样频率高采样时间较长，则缓存占用内存过大时会导致界面卡顿，可根据使用情况调整，默认为 100（万点）。

自动检测模型变化：自动检测在变量导入窗口指定的 axf/map/elf 文件的属性，有更新时发出提示。



4.2. 下位机设置

LKS 下位机设置

RTT块地址

RTT搜索地址(0x)

如果已经指定axf/map文件, 则RTT数据块直接获得, 无需搜索。

RTT写入频率

RTT写缓冲频率(Hz)

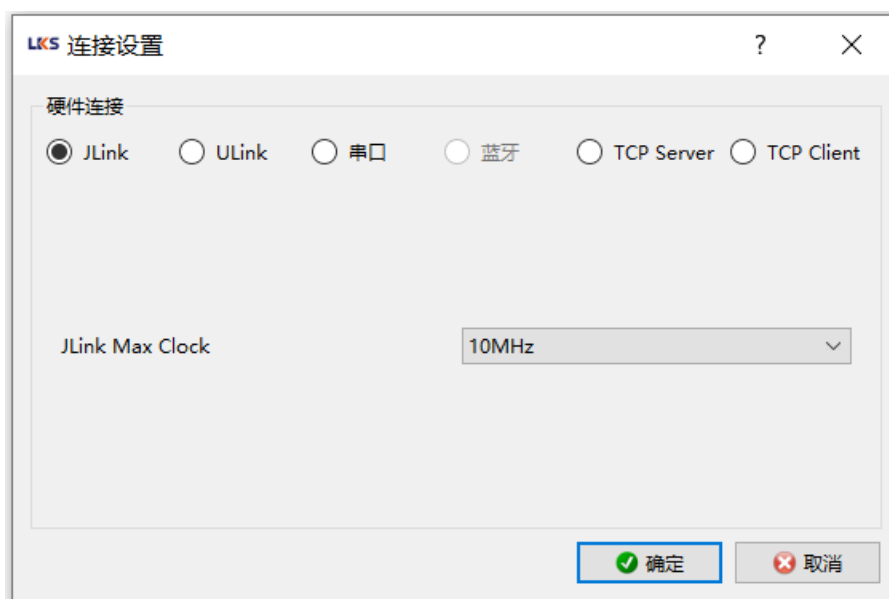
在主循环或者定时器中断中调用RTT写缓冲的频率, 用于RTT数据的时间轴显示。

- **RTT 块搜索地址：**当您无法获得 **axf** 文件时，支持自动搜索 RTT 块地址。如果已有 **axf** 文件，则忽略该参数。
- **RTT 写入频率：**填入下位机中实际调用 **RTT** 写缓冲函数的频率。该函数可能在您的主循环或定时器中断中调用。该参数用于显示缓冲数据点的相对时间。

4.3. 连接设置

lkviewer 当前支持 JLink/ULink/串口几种连接链路，不同的链路具有不同的连接方式。建议使用 JLink 连接，通信速度最快。如果不具备 JLink 的连接条件，再考虑选择其他连接方式，其他几种方式适用于采样频率不是特别高的场合。

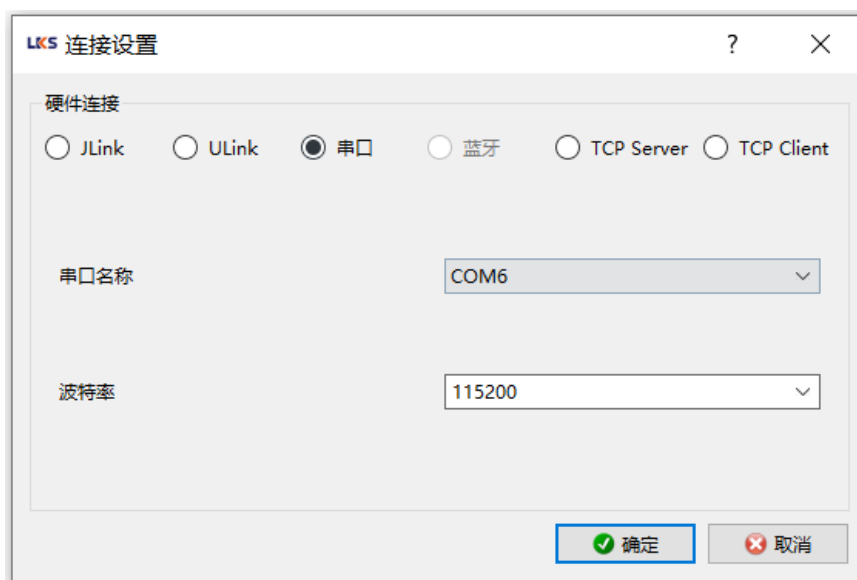
4.3.1. JLink/ULink



JLink 与 ULink 连接方式一样，通过 JLink/ULink 仿真器与目标板的 SWD 直接连接，与日常的 Keil 调试连接方式一致。

使用 JLink 连接时，Iksviewer 与 Keil 可以同时访问目标板，前提是二者设置的 JLink 时钟一致，否则会有访问冲突。

4.3.2. 串口



串口连接需要协议的支持，将在后面介绍。

计算机侧可能需要连接 USB 转串口的工具，与目标板开放的串口进行连接。

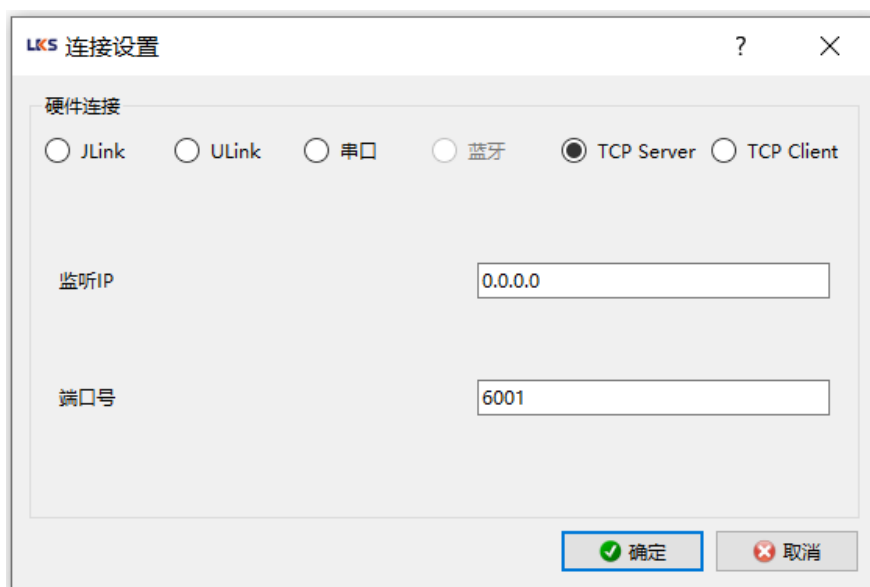
4.3.3. 蓝牙

蓝牙连接需要协议的支持，将在后面介绍。



为方便维护连接状态，建议在 windows 系统下配置虚拟蓝牙串口，将蓝牙连接当作本地串口使用。

4.3.4. TCP Server



TCP 连接需要协议的支持，将在后面介绍。

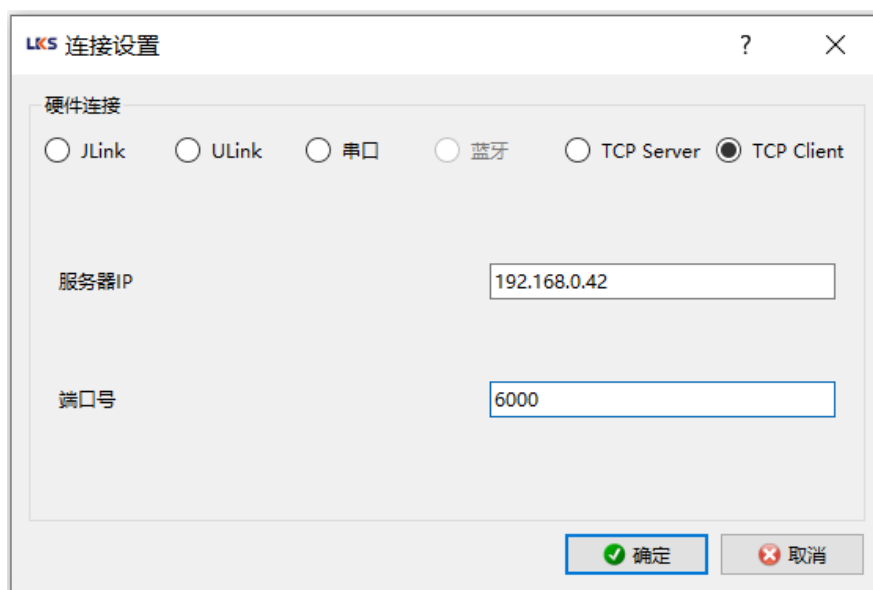
lkviewer 作为 TCP 的服务端使用，设备端可以使用 ESP8266 等模块，配置自动连接和透传，上电后自动连接 lkviewer 进行通信。ESP8266 在默认情况下每 20ms 透传一包数据，所以在使用 8266 模块时采样间隔不宜设置过低。

ESP8266 模块可以使用如下的 AT 命令配置为上电自动连接 wifi 和 lkviewer。其中的 wifi 名称和密码，lkviewer 监听的 IP 和端口号根据实际情况填写。

```
AT
AT+CWMODE=1 // 设置 STA 模式
AT+RST // 重启生效
AT+CWJAP="LINKO", "xxxxxxxx" // 连接路由器，设置 SSID 和密码
AT+CIFSR // 查询模块 IP
AT+SAVETRANSLINK=1, "192.168.0.42", 6001, "TCP" // 上电连接并透传，设置 IP 和端口

// 如果需要修改配置，发送+++关闭透传后重新配置。
+++ // 关闭透传
```

4.3.5. TCP Client



TCP 连接需要协议的支持，将在后面介绍。

lkviewer 作为 TCP 的客户端使用，设备端作为服务器监听指定端口。lkviewer 启动连接后向设备端发起连接，连接成功后按既定协议进行通信。

5. 变量管理

要实现变量的动态管理，需要首先告知 lkviewer 各变量的地址以及数据宽度。这里支持变量导入和自定义添加变量两种方式实现。

5.1.变量导入

导入变量					
变量名称	描述	地址(0x)	数据长度	类型	代码位置
_LKS_VIEWER.postNum	_LKS_VIEWER.p...	0x20000a48	4	u32	
_LKS_VIEWER.recStart	_LKS_VIEWER.re...	0x20000a4c	4	u32	
_LKS_VIEWER.recReset	_LKS_VIEWER.re...	0x20000a50	4	u32	
_LKS_VIEWER.recState	_LKS_VIEWER.re...	0x20000a54	4	u32	
_LKS_VIEWER.sampleNum	_LKS_VIEWER.sa...	0x20000a58	4	u32	
_LKS_VIEWER.lastValue	_LKS_VIEWER.la...	0x20000a5c	4	int	
recvData	recvData	0x20000900	255	array[255] of u8	
recvLen	recvLen	0x20000040	1	u8	
_ActiveTerminal	_ActiveTerminal	0x00000000	1	unsigned char	
_SEGGER_RTT	_SEGGER_RTT	0x20000448	168	SEGGER_RTT_CB	
_SEGGER_RTT.acID	_SEGGER_RTT.a...	0x20000448	16	array[16] of char	
_SEGGER_RTT.MaxNumU...	_SEGGER_RTT....	0x20000458	4	int	
_SEGGER_RTT.MaxNumD...	_SEGGER_RTT....	0x2000045c	4	int	
_SEGGER_RTT.aUp	_SEGGER_RTT.a...	0x20000460	72	array[3] of SEG...	
_SEGGER_RTT.aDown	_SEGGER_RTT.a...	0x200004a8	72	array[3] of SEG...	
_aTerminalId	_aTerminalId	0x00000000	16	array[16] of unsigned char	

D:/linko_files/lks08x_viewer/.Obj/lks08x_viewer.axf

导入变量 自定义变量

在主界面的左下角，切换“导入变量”选项卡，点击右侧的文件夹图标，可以选择指定的 **axf/map/elf** 文件，也可以直接将 **axf/map/elf** 或 **uvproj** 文件拖动到此窗口。**lkviewer** 将解析文件中的全局变量名称，显示在变量窗口中，供后续使用。

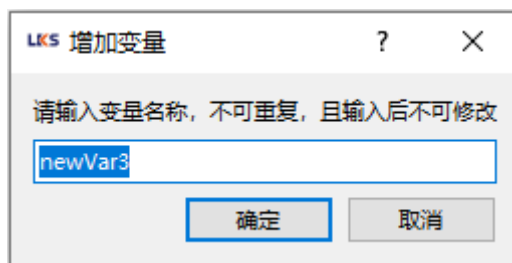
lkviewer 支持解析 **axf/map/elf** 三种文件格式，但仅对 **keil** 生成的 **axf** 文件能解析结构体下的成员，建议尽量使用 **axf** 文件，以实现更全面的数据观察。

您也可以拖入 **uvproj(x)** 文件，将根据 **uvproj** 中的 **OutputDirectory** 和 **OutputName** 定位到 **axf** 文件并加载。

您可以对列表中的描述信息进行修改，描述信息将保存在配置文件文件中，下次启动时加载使用。其他参数则统一从 **axf** 文件重新提取，以确保获得最新的参数列表。



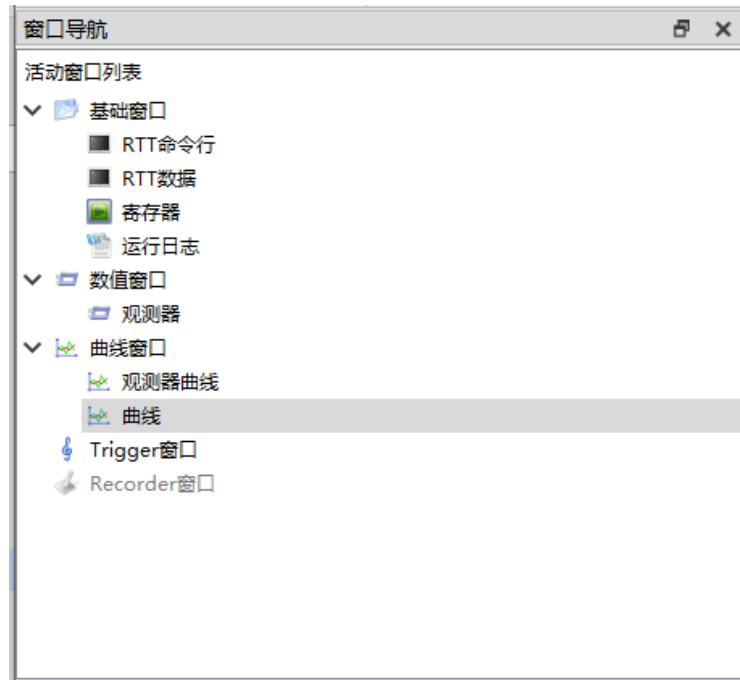
5.2. 自定义变量



如果已知某个变量的地址和宽度，可以切换到“自定义变量”选项卡进行自定义添加。在子窗口内部点击右键，将激活“增加”窗口。设置变量名称后，变量将被添加到列表中。变量名称设定后不允许修改，其他字段可以双击触发修改。

一般情况下不需要添加自定义变量。

6. 窗口导航

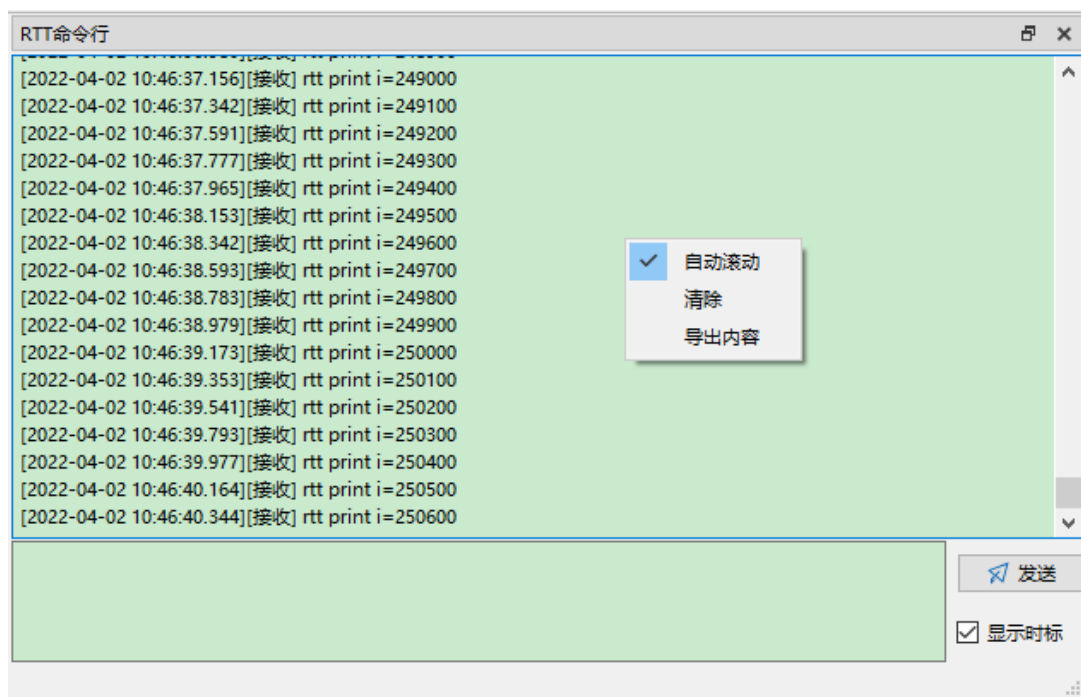


Iksviewer 集成了多种调试功能，不同的功能以不同的子窗口类型展现。每个子窗口以 **DockView** 的形式存在，可以根据需要拖动位置和缩放大小。如果您有双屏幕，也可以将子窗口拖到其他屏幕显示，以获得更大的显示空间。

6.1. 基础窗口

基础窗口定义为全局唯一存在的窗口，这类窗口在程序启动时就存在，双击对应条目可以打开。

6.1.1. RTT 命令行



该窗口的功能类似于 SEGGER_RTT 下的 RTT_VIEWER 组件,用于显示下位机通过 PRINTF()输出的内容。PRINTF()函数的定义与使用将在后面的下位机支持库章节具体解释。

您可以使用右键菜单导出窗口中的内容,保存到指定文件中,供后续查看。

为了方便 PRINTF 功能的调用,您可以创建头文件 rtt_debug.h 并将如下的宏定义添加到文件中,再将 rtt_debug.h 添加到您的工程中。通过开头的宏开关控制是否打印输出,以及是否采用 RTT 通道输出。

```
#include <stdio.h>

/*用于控制输出调试信息*/

#define PRINTF_DEBUG    1U

#define USE_RTT_PRINTF  1U

#if (PRINTF_DEBUG > 0U)

    #if (USE_RTT_PRINTF > 0U)

        #include "SEGGER_RTT.h"

        /*基于 SEGGER_RTT 不能输出浮点,中文,等一些特殊字符(优点速度快可在中断中调用)*/

        #define    PRINTF_Dbg(...)    \

            do{\
```

```

SEGGER_RTT_printf(0,"file:%s,line:%d",__FILE__,__LINE__);\\

        SEGGER_RTT_printf(0,__VA_ARGS__);\\

}while(0);


#define   PRINTF(...)   \\

        SEGGER_RTT_printf(0,__VA_ARGS__);

#else

/*使用串口格式化到 printf 输出*/

#define   PRINTF_Dbg(...)   \\

        do{\\

                printf("file:%s,line:%d",__FILE__,__LINE__);\\

                printf(__VA_ARGS__);\\

        }while(0);


#define   PRINTF(...)   \\

        printf(__VA_ARGS__);

#endif


#else

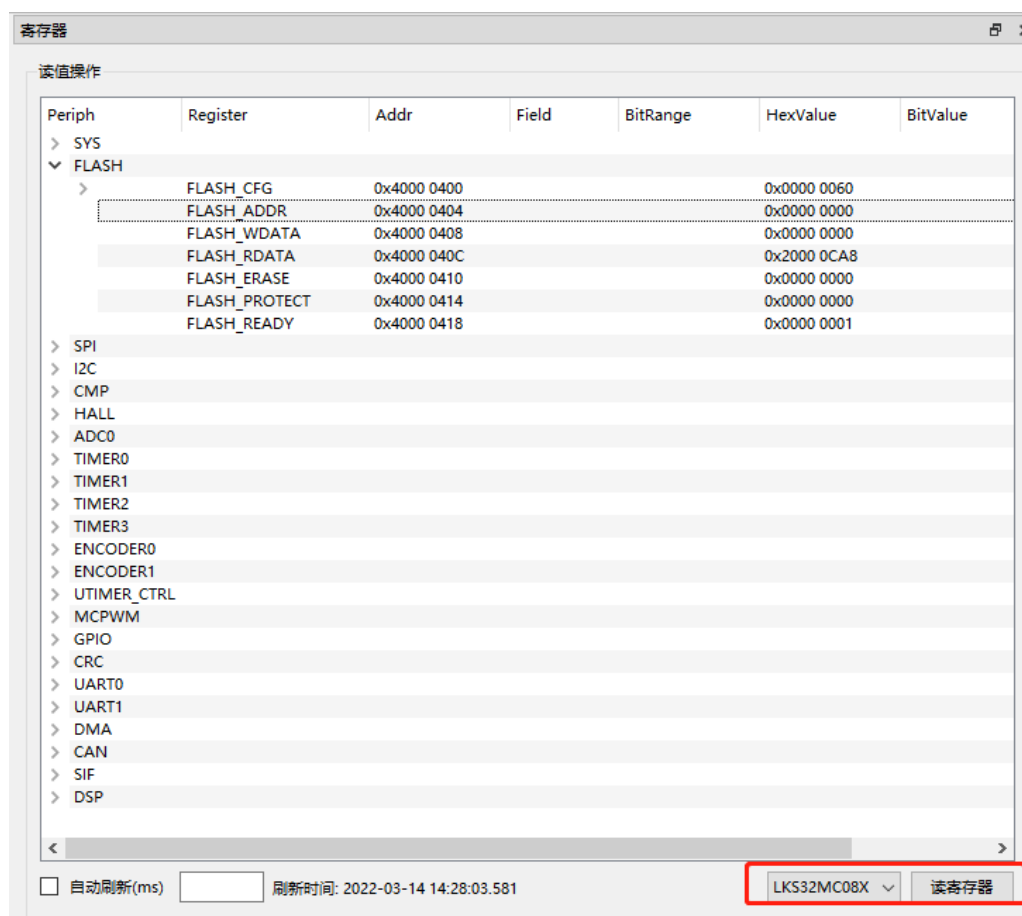
#define   PRINTF_Dbg(...)

#define   PRINTF(...)

#endif

```

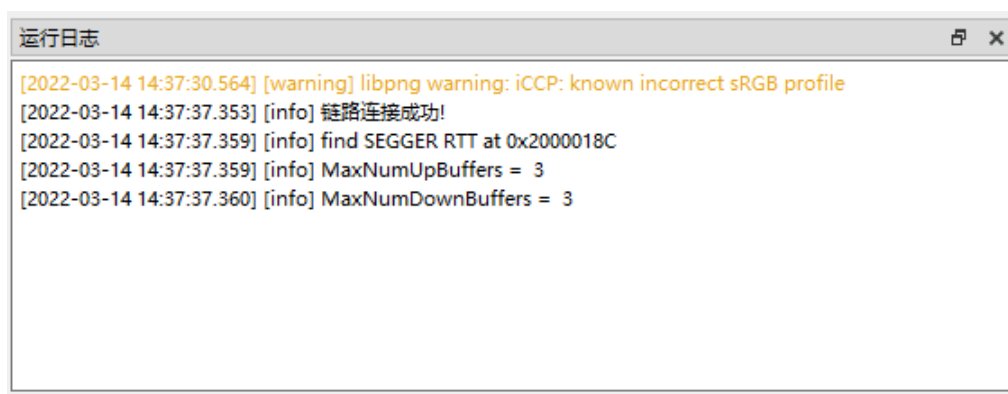
6.1.2. 寄存器



寄存器窗口用于查看 LKS 芯片当前所有寄存的值，及各 bit 位的含义。在使用时，注意选择与目标 MCU 相匹配的类型，点击“读寄存器”按钮更新数值。

双击寄存器对应的 HexValue 或 BitValue 列，可以触发写值。修改 Value 并回车，将修改寄存器值。

6.1.3. 运行日志



运行日志窗口输出 lkviewer 自身在运行过程中输出的信息。一般情况下无需关心该窗口内容，当遇到未知的问题时，可以打开该窗口查看内容，也许可以获得有用的提示。

6.2. 数值窗口

数值窗口用于查看目标变量的实时值。在新建的 **lkviewer** 工程中，默认创建一个数值窗口。如需管理多组窗口对象，可以在“窗口导航”界面中右键点击“数值窗口”来新建这类子窗口。

数值0									
变量名称	描述	地址	显示方式	刷新速率(ms)	标么基准值	当前值	最大值	最小值	背景色
sinVal	sinVal_desc	0x2000000c	Signed	100	32767	0.208625	0.999969	0.00170904	#ff0000
theta	theta_desc	0x20000004	Signed	100	1	2195	32750	20	#00ff00
theta2	theta2_desc	0x20000008	Signed	100	1	-2195	-20	-32750	

从主界面左下角的“导入变量”中，选中要监测的多个对象，拖入到“数值窗口”，即完成监测对象的添加。显示方式从变量的原始类型推算而来，刷新周期默认取自基本设置中的“默认采样周期”参数，您可以根据需要修改合适的显示方式和刷新速率。刷新速率仅在连接未启动时允许修改。

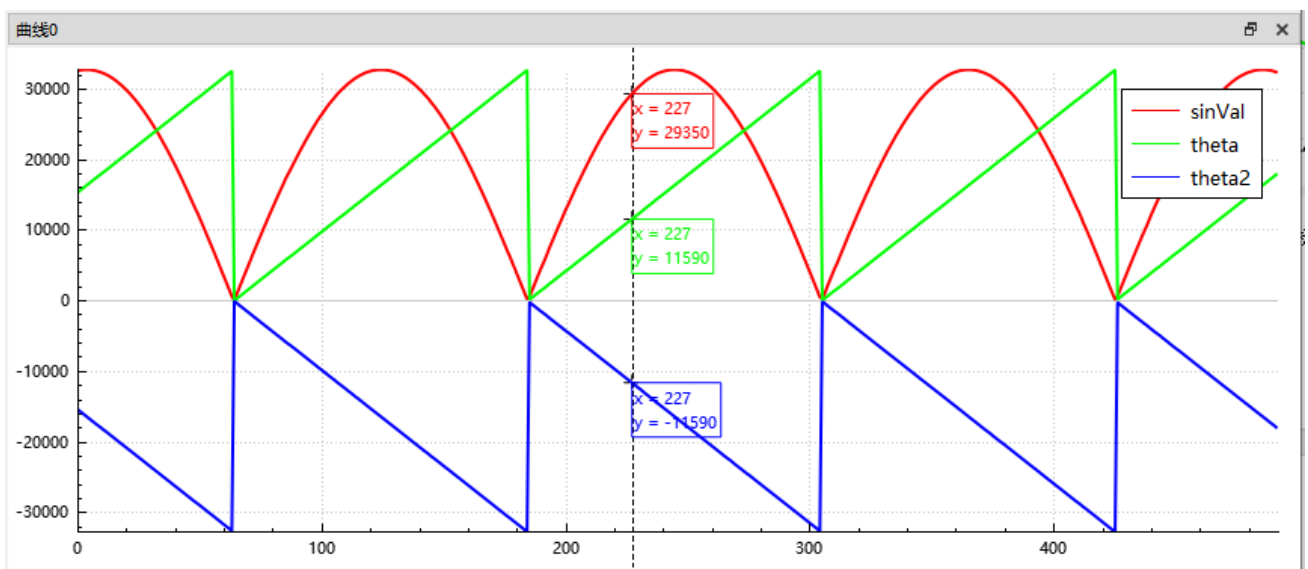
刷新速率当前支持最小间隔 **1ms**，如果要获得更高的数据刷新速率，建议在下位机使用 **RTT** 缓冲通道，数据刷新可以达到数十 **k**。

在启动连接后，最大值、最小值和当前值将在启动连接后保持刷新。选中当前值一列的表格元素，双击或输入任意键，可以触发修改，**Enter** 键下发写值，**Esc** 取消修改。

6.3. 曲线窗口

曲线窗口与数值窗口一样实现变量的采集，以曲线的形式展示数据。

6.3.1. 基本曲线

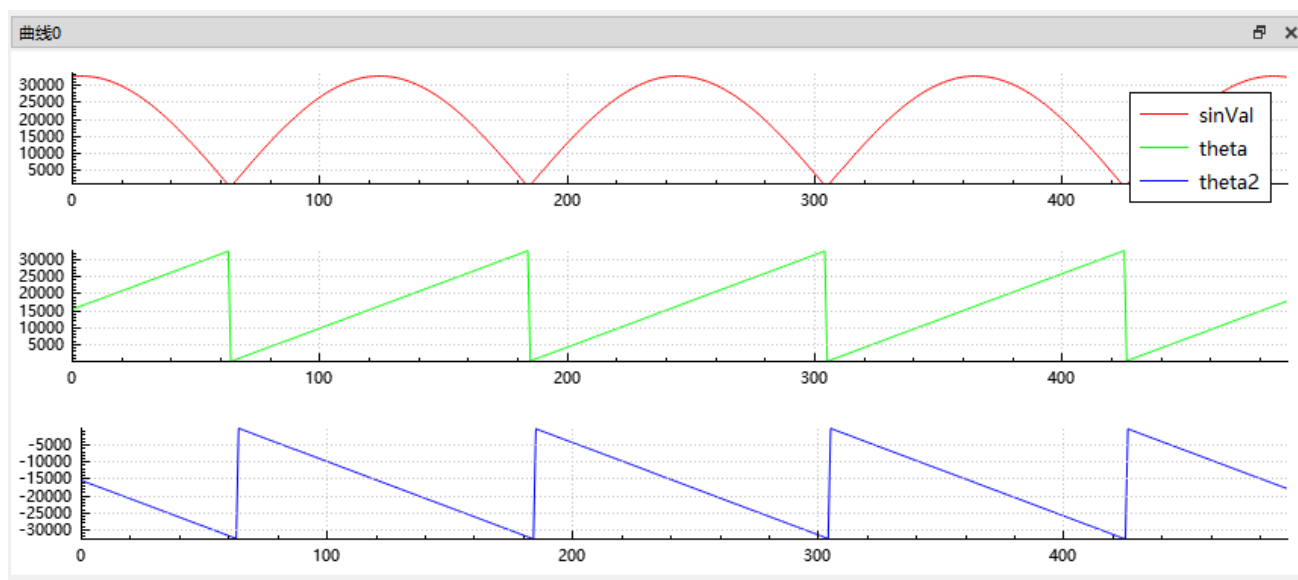


曲线窗口支持拖动的方式添加变量，右键菜单删除变量。点击选中变量后，显示游标跟随鼠标移动。



曲线窗口的右键菜单提供了多种查看及管理入口，后续还将继续添加完善。

6.3.2. 多子图显示

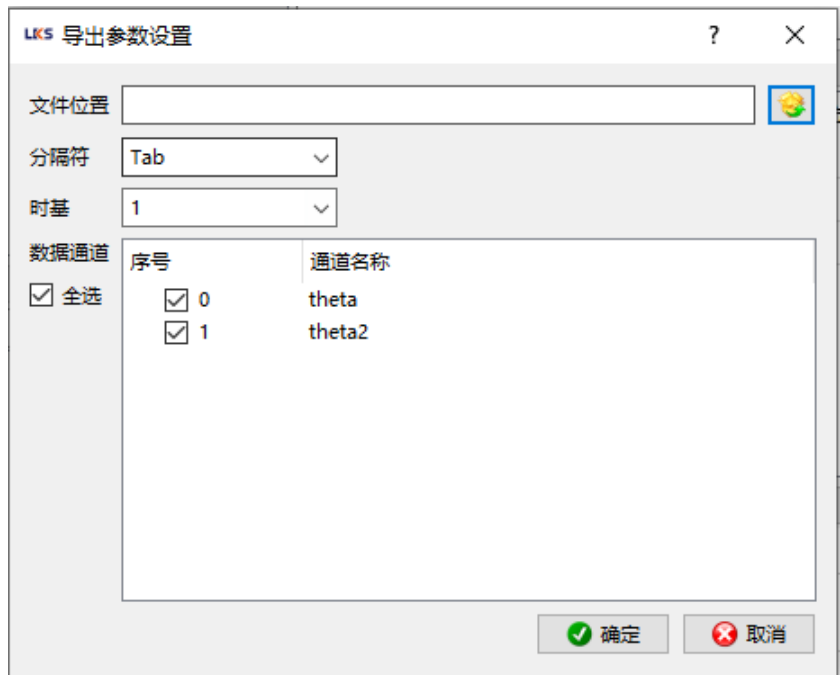


如果多条曲线的 Y 轴分布数量级不同，可以通过右键切换为多子图模式。该模式下每条曲线使用独立的绘制区域，有各自的 Y 轴范围，而 X 轴同步显示。

6.3.3. 导出图形

点击右键菜单的“导出图形”按钮，并指定保存位置后，将把当前窗口的图形内容保存到文件中。支持保存为 png/jpg/jpeg/bmp/pdf 类型的文件，通过后缀名指定类型。如果不指定类型，则默认保存为 png 图片。

6.3.4. 导出数据



点击右键菜单的“导出数据”按钮，设置导出参数及数据通道后，将把当前窗口的数据内容保存到文件中。

可以在导出参数中设置分隔符，时基，选择要导出的数据通道。

	0	1.0	2.0	3.0	4.0	5.0
1	sinVal	theta	theta2			
2	26968	10095	-10100			
3	27898	10635	-10640			
4	28784	11195	-11205			
5	29572	11750	-11755			
6	30258	12285	-12290			
7	30886	12840	-12845			
8	31410	13380	-13390			
9	31862	13930	-13935			
10	32212	14465	-14470			
11	32480	15015	-15020			
12	32654	15555	-15560			
13	32752	16095	-16105			
14	32756	16630	-16635			
15	32680	17175	-17180			
16	32500	17725	-17730			
17	32236	18270	-18275			
18	31892	18815	-18820			
19	31452	19360	-19365			
20	30934	19895	-19900			
21	30330	20440	-20445			
22	29642	20990	-20995			
23	28884	21525	-21530			
24	28034	22070	-22075			
25	27108	22615	-22620			
26	26098	23165	-23170			
27	25022	23715	-23720			
28	23888	24255	-24260			
29	22678	24805	-24810			
30	21388	25360	-25365			

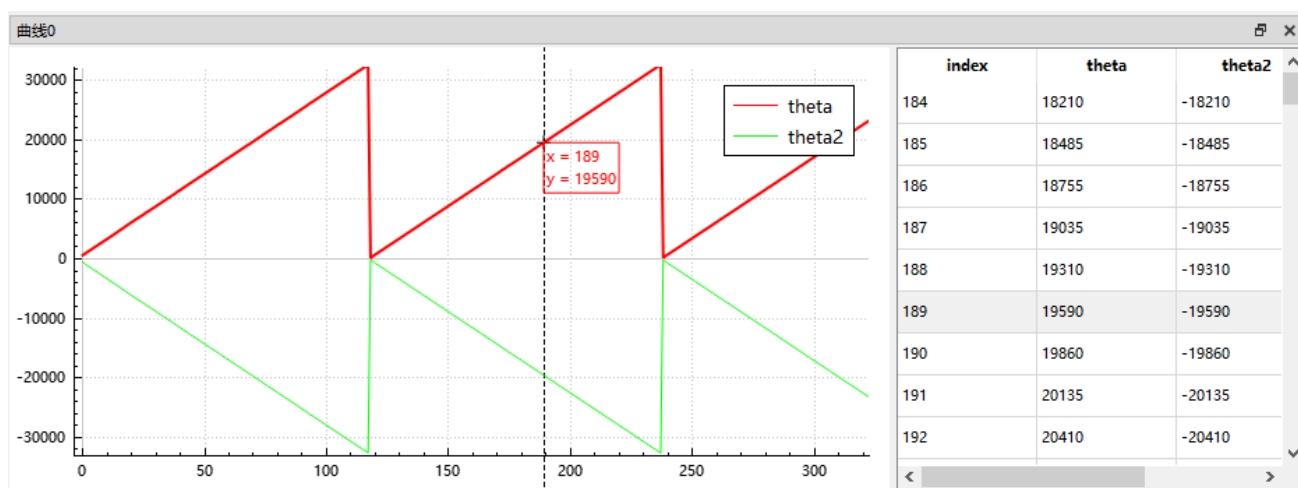
导出数据的第一行存储标题，后面各行存储每一组数据，每一行具有相同的分隔符，即导出参数设置中指定的分隔符。

6.3.5. 导入数据

曲线窗口未提供导入数据的右键菜单，但支持数据文件的拖动式导入。从操作系统的资源管理器选择数据文件后，拖动到曲线窗口，即实现数据的还原。

导入数据支持来自其他软件生成的文件，但应具有与导出数据类似的存储格式。曲线窗口自动识别文件中的分隔符，目前支持的分隔符有：[Tab Space , ; \$ /]。

6.3.6. 数据查看

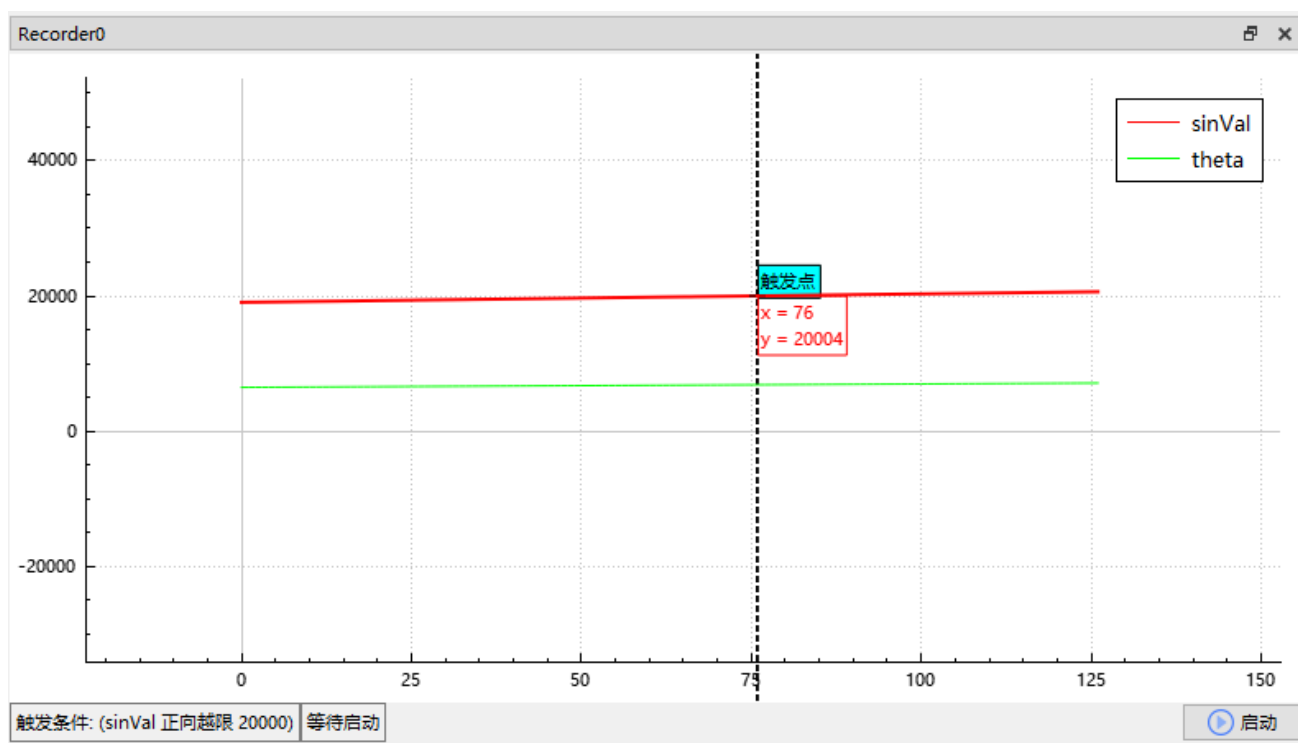


当需要具体查看曲线上某一区域的具体数值时，可以点击右键菜单下的“数据查看”按钮，打开数据查看窗口。

数据查看子窗口与选中曲线的 X 轴坐标联动显示，其分段加载和显示曲线数据，百万级的测点显示不卡顿。双击数据子窗口中的条目，也将触发曲线的自动滚动，并在 X 轴的对应位置上显示标尺。

6.4.Recorder 窗口（后续版本开放）

Recorder 是电机调试中常用的一种功能，经常用于排查故障。上位机将相应的参数设置完成后，下发给下位机，由下位机对触发条件进行判断，并在触发后采样指定数量的数值。待上位机将缓冲区数据拉取完成后，复位下位机的 Recorder 状态，恢复正常的的数据缓存功能。



启动连接后，点击右下角的启动按钮，将给下位机下发 Recorder 配置的各参数。然后由下位机对触发条件进行判断，缓存指定数量的数据后，上位机拉取并显示数据，复位 Recorder 状态。

6.5.Trigger 窗口

Trigger 窗口功能与 Recorder 类似，其实现过程简单，不需要下位机库的支持。由上位机自身对触发条件进行判断，记录触发位置，并在触发条件成立后，采集指定数量的数值并停止刷新，以记录触发前后的数据现场。缺点是不如 Recorder 对触发条件的检测及时性高。

The LKS Trigger参数设置 window contains three tabs: '记录参数' (Recording Parameters), '触发条件' (Trigger Conditions), and '记录数据' (Recording Data). The '记录参数' tab is active, showing the following settings:

- 记录周期(ms): 100
- 记录数量: 200
- 线型: 曲线 (Curve)

At the bottom right, there are '确定' (Confirm) and '取消' (Cancel) buttons.

?

×

LKS

Trigger参数设置

记录参数

触发条件

记录数据

序号	变量	变量地址	条件	阈值
0	theta	0x20000004	正向超限	20000

+

增加

-

删除

✓

确定

✗

取消

?

×

LKS

Trigger参数设置

记录参数

触发条件

记录数据

变量	变量地址	变量宽度	曲线颜色
theta2	0x20000008	4	#ff0000
sinVal	0x2000000c	4	#00ff00

+

增加

-

删除

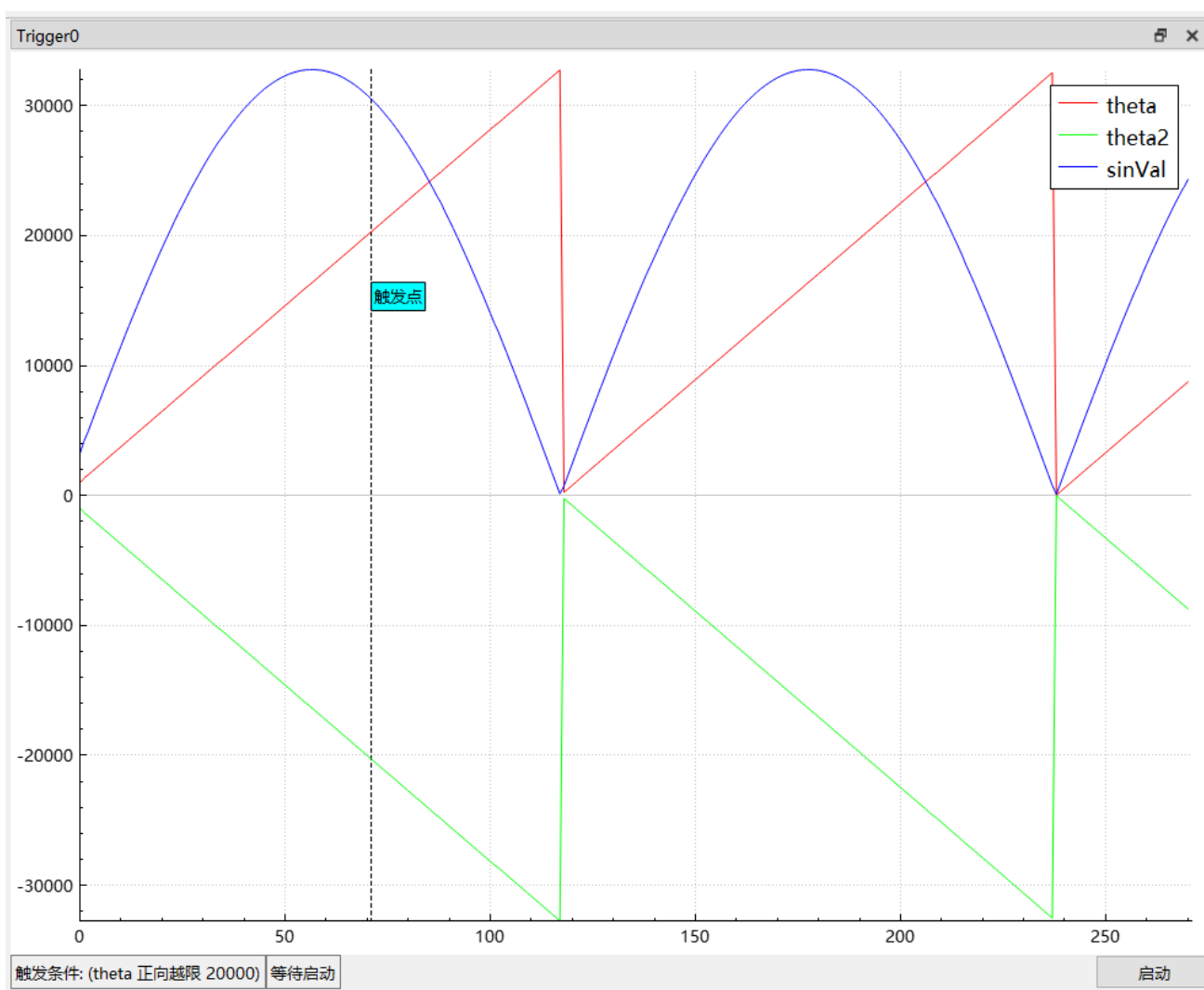
✓

确定

✗

取消

Trigger 的参数设置包括记录周期，触发后采样数量，触发条件以及记录对象。由于触发条件是由上位机自己判断，所有可以支持多组触发条件，且对触发点的数据类型无限制。



参数设置完毕，点击右下角的启动按钮后，将启动相关数据的采样。启动 **Trigger** 之前，需要先在工具栏中启动连接，否则将收到错误提示。

当采样点触发时，在图中标记触发位置。触发后采集指定数量的数值并停止刷新，保留数据现场。

7. 下位机支持库

lkviewer 的部分功能需要下位机库的支持，但即使不做任何下位机功能移植，您仍可以使用大部分功能。当前版本下，为了方便软件的使用和理解，建议在移植 **RTT** 库的情况下使用。凌鸥提供的工程模板一般自带 **RTT** 的内容。

其中 **SEGGER_RTT** 目录来自于 **JLink** 软件包自带文件，我们这里对其稍作了修改，以支持浮点数和中文的显示，整体功能与 **JLink** 保持兼容。**SEGGER** 的版权声明里允许对其进行二次开发。

与下位机支持相关的功能如下表所示，其中部分内容在后续版本更新中提供。

功能项	下位机无移植	下位机移植 RTT	下位机移植 RTT+LKSVIEWER	备注
JLink 链路	√	√	√	



ULink 链路	√	√	√	
串口链路	-	-	√	待添加
RTT 命令行	-	√	√	
寄存器	√	√	√	
数值窗口	√	√	√	
曲线窗口	√	√	√	
Recorder	-	-	√	待添加
Trigger	√	√	√	

8. 通信协议说明

当您使用串口、蓝牙、TCPserver、TCPClient 链路时，需要在下位机添加代码处理协议。Iksviewer 设定的协议内容较简单，仅包含读 Memory 和写 Memory 两组协议帧。

8.1. 读 Memory

8.1.1. 请求帧

序号	示例(0x)	含义
0	68	起始字节
1	01	帧序号。应答帧中序号必须一致。
2	06	数据区长度。CMD 及以后的字节数。
3	29	数据区校验码。CMD 及以后的字节参与计算。
4	01	命令内容：01-读。
5	0C	读内存的起始地址。 四字节表示，低字节在前。
6	00	
7	00	
8	20	
9	04	读内存的长度，每包 255 以内。

上面的一帧报文表示：读 0x2000000C 开始的 4 字节内存数据。



8.1.2. 应答帧

序号	示例(0x)	含义
0	68	起始字节
1	01	帧序号。与请求帧中的序号必须一致。
2	05	数据区长度。CMD 及以后的字节数。
3	CE	数据区校验码。CMD 及以后的字节参与计算。
4	01	命令内容：01-读。
5	84	内存数据，长度与请求帧中指定的一致。
6	4B	
7	00	
8	00	

上面的一帧报文表示：响应帧序号为 01 的读请求，返回四字节内存数据 84 4B 00 00。lkviewer 根据自己维护的变量属性解析内存数据并显示到界面中。

8.2. 写 Memory

8.2.1. 请求帧

序号	示例(0x)	含义
0	68	起始字节
1	1A	帧序号。应答帧中序号必须一致。
2	09	数据区长度。CMD 及以后的字节数。
3	4A	数据区校验码。CMD 及以后的字节参与计算。
4	02	命令内容：02-写。
5	0C	写内存的起始地址。 四字节表示，低字节在前。
6	00	
7	00	
8	20	
9	64	向内存写入的数据。
10	00	
11	00	



12	00	
----	----	--

上面的一帧报文表示：向 0x2000000C 开始的内存地址，写入四字节数据 64 00 00 00。

8.2.2. 应答帧

序号	示例(0x)	含义
0	68	起始字节
1	1A	帧序号。应答帧中序号必须一致。
2	02	数据区长度。CMD 及以后的字节数。
3	03	数据区校验码。CMD 及以后的字节参与计算。
4	02	命令内容：02-写。
5	01	写值结果。1-成功，0-失败。

上面的一帧报文表示：响应帧序号为 1A 的写请求，写成功。

8.3.CRC 计算

报文中的 CRC 计算方式如下。

```

uchar CalcCRC(uchar *pData, int len)
{
    uchar crc = 0;

    for (int i = 0; i < len; i++)
    {
        crc ^= pData[i];
    }

    return crc;
}

```

