

# MangoTree RobustRIO

## 上手指南



# 目录

一、 产品概述 .....	3
二、 开发环境搭建 .....	4
三、 开发第一个 RobustRIO 项目（RT系统） .....	10
1. 连接与发现设备 .....	10
3. LinuxRT系统语言配置 .....	11
4. 新建LabVIEW项目 .....	11
5. 新建终端设备 .....	12
6. 更改设备IP地址（如需） .....	13
7. 连接RobustRIO终端设备 .....	13
8. 新建RobustRIO机箱终端（以U808为例） .....	14
9. 新建RobustRIO FPGA终端 .....	15
10. 第一个Robust RIO项目！ .....	15
10.1 FPGA开发模式 .....	15
10.2 使用SCAN模式开发 .....	41
2. 部署RobustRIO 实时应用程序 .....	43
四、 开发第一个 RobustRIO 项目（Windows系统） .....	45
五、 开发第一个 RobustRIO 项目（雷电接口） .....	45

## 一、产品概述

MT RobustRIO 是一种坚固可靠的工业化控制和采集系统，采用可重新配置 I/O（Reconfigurable I/O, 缩写为 RIO）FPGA 技术实现超高性能和可自定义功能。MT RobustRIO 嵌入式系统可以使用LabVIEW图形化编程工具进行快速开发。利用 MT RobustRIO 可以快速建立嵌入式控制和采集系统，而且该系统的工作性能和优化特性可与专门定制的硬件电路媲美，用户可以使用LabVIEW、LabVIEW RT模块、LabVIEW FPGA模块、VISA驱动、RIO 驱动以及RobustRIO 驱动来开发 MT RobustRIO嵌入式系统。

用户可以通过LabVIEW FPGA的基本I/O功能，直接访问RobustRIO FPGA硬件上I/O，通过I/O来进行相应的操作。MT RobustRIO 嵌入式系统搭载2.4GHz四核四线程Intel处理器，最高128GB SSD，4GB DDR3 RAM，在保证可靠性和确定性的前提下顺利执行LabVIEW实时程序，分析，数据记录与通讯。控制器上集成千兆以太网，可在网络上进行通信编程。由于RIO 的并行特性，所以增加新的计算并不会降低 FPGA 程序的执行速度。

一般情况下，可重新配置的控制和采集系统包括四个主要部分：

- ✓ 用于输入，输出，通讯和控制的 RIO FPGA 核心部分；
- ✓ 用于浮点控制，信号处理，分析和逐点决策的实时循环；
- ✓ 用于内嵌数据记录，网络远程面板和以太网/串口的普通优先级的循环；
- ✓ 用于远程图形化用户界面，历史数据记录和事后处理的联网的主PC；

使用 MT RobustRIO，您可以快速的解决复杂问题的挑战，大大提高工作效率。

## 二、开发环境搭建

\*如果您购买的RobustRIO产品系统为Windows版本，出厂会预装好全部开发所需的软件及驱动，用户无需自行安装即可上手开发。

\*如果您购买的RobustRIO产品系统为Linux RT版本或者雷电接口版本，则需要自行在自己的PC下搭建开发环境，具体参照如下步骤：

**注意：**以下所有软件必须按照默认的路径安装（C盘），且按照先后顺序，否则会导致无法安装成功，共需要C盘20G左右的空间。

下面所有的安装过程，确保电脑没有开启任何杀毒软件及安全助手，以免导致安装失败。

### 步骤一：MT-Master软件安装

MT-Master下载链接：

<http://server.mangotree.cn:9000/Software/MangoTree/MT-Master/>，选择对应系统的安装包进行下载安装。



### 步骤二：LabVIEW相关软件安装

推荐安装LabVIEW2020版，可在下面的链接中进行下载安装，

LabVIEW2020下载：

[http://server.mangotree.cn:9000/Software/LabVIEW/LabVIEW2020/All\\_in\\_One\\_Install/](http://server.mangotree.cn:9000/Software/LabVIEW/LabVIEW2020/All_in_One_Install/)



安装过程中正常开发只需要安装以上勾选的五个软件，然后点击下一步，在下面的安装选择页面中，将Web服务器开发支持选择取消，否则可能会导致安装不成功，如下图：

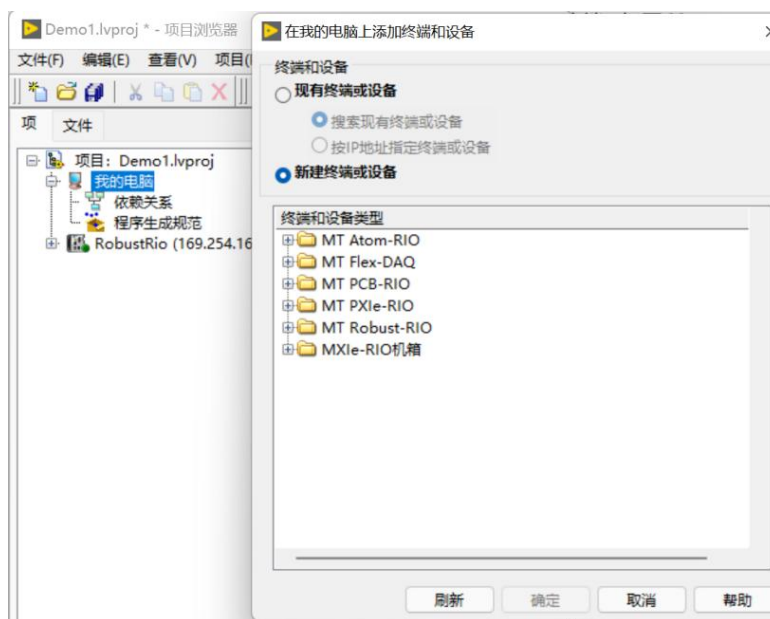


### 步骤三：MT RIO驱动安装

MT-RIO安装包下载：<http://server.mangotree.cn:9000/Software/MangoTree/MT-RIO/>

如何验证MT RIO安装包是否安装成功？

打开LabVIEW新建一个项目，在我的电脑下右键新建一个终端，如图所示：



如果能出现如图所示MT的产品选型即认为MT驱动软件已经安装完毕。

## 步骤四：FPGA编译器安装

FPGA程序需要安装编译器进行编译，编译器下载地址：

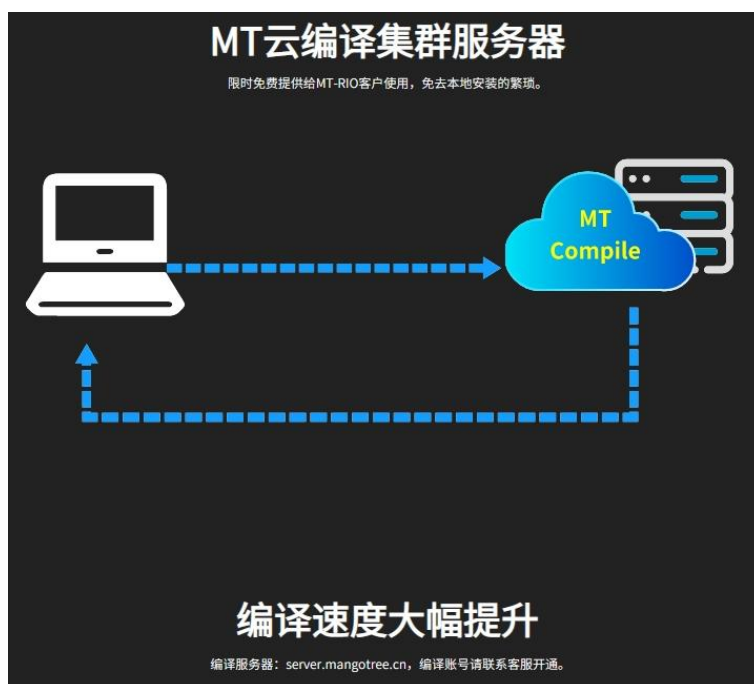
[http://server.mangotree.cn:9000/Software/LabVIEW/LabVIEW2020/Separate Install/](http://server.mangotree.cn:9000/Software/LabVIEW/LabVIEW2020/Separate%20Install/)



如上图提供两种编译器，购买RobustRIO的型号为MT-U800、MT-U801、MT-U804、MT-U808的安装ISE编译器，RobustRIO的型号为MT-U834、MT-U838的安装Vivado编译器。

注：ISE编译器在Win10下安装提示系统兼容问题可忽略继续安装，使用过程中可能会出现编译过程卡死情况，建议用户Win7安装ISE编译器，开发的PC可以通过远程的方式连接Win7中的编译器进行编译。

限时免费提供MT云编译集群服务器给MT-RIO客户使用，免去本地安装的繁琐，大幅提升编译速度，如下图所示：

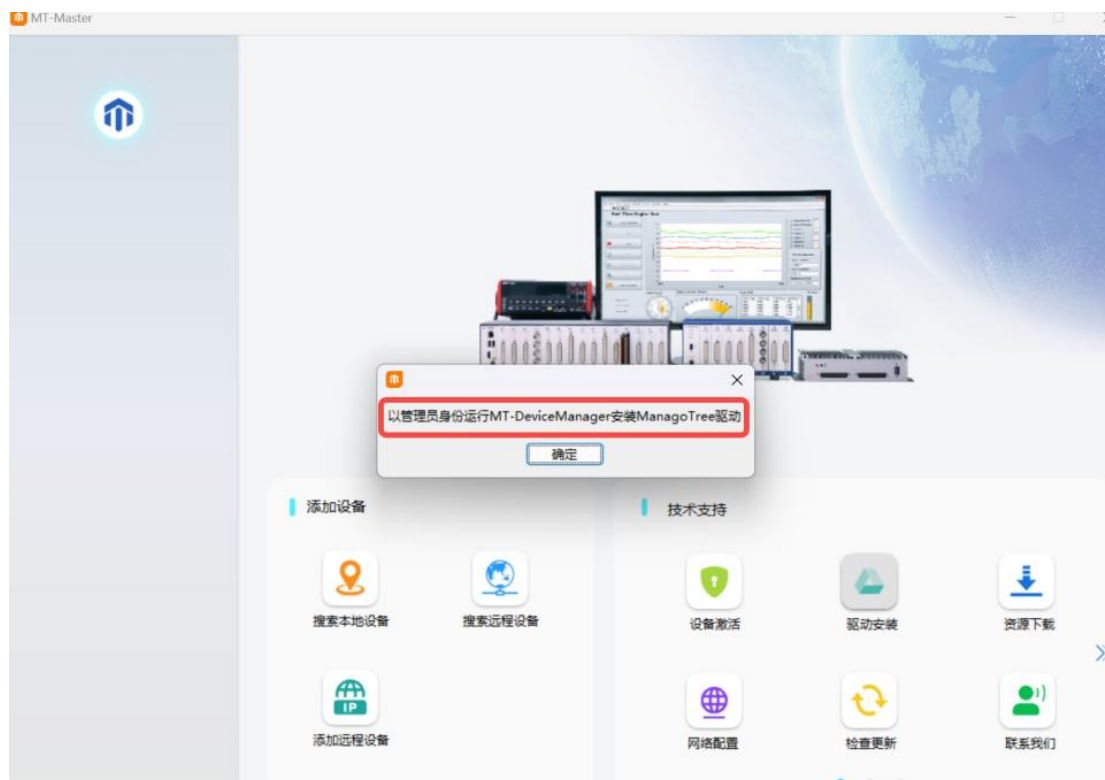


注：编译服务器：[server.mangotree.cn](http://server.mangotree.cn)，编译账号请联系客服开通。

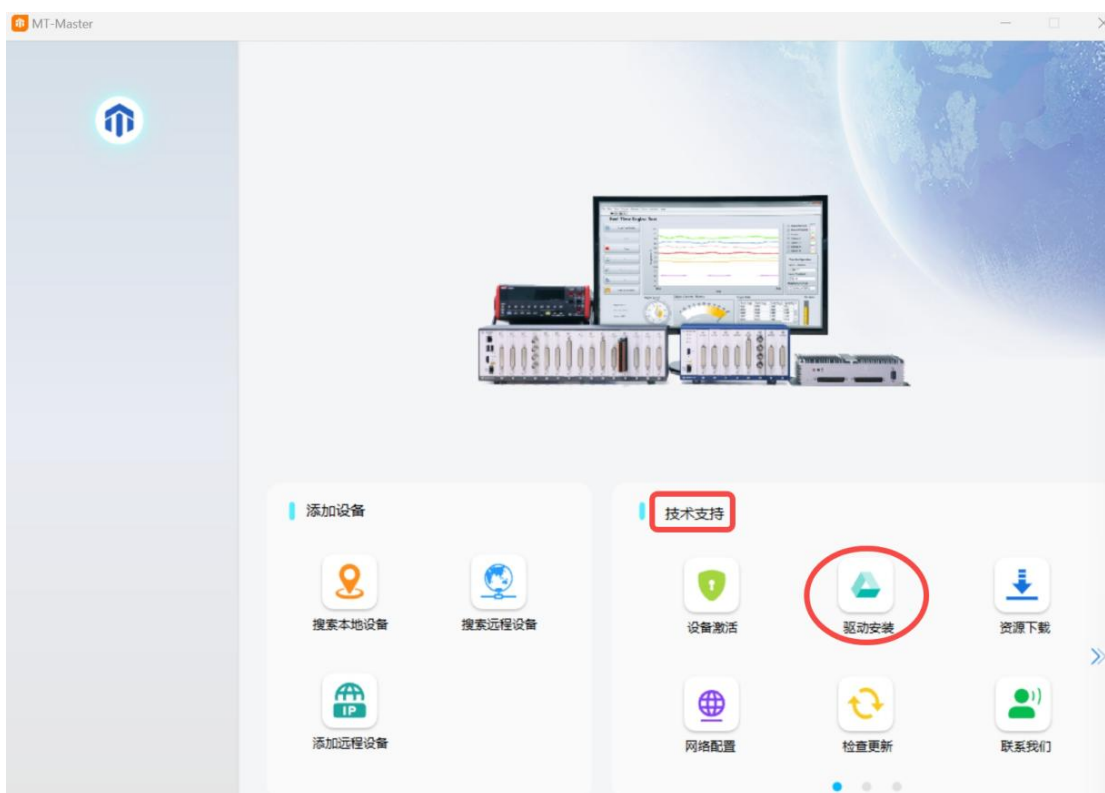


## 步骤五：雷电驱动安装（只针对雷电接口的RobustRIO）

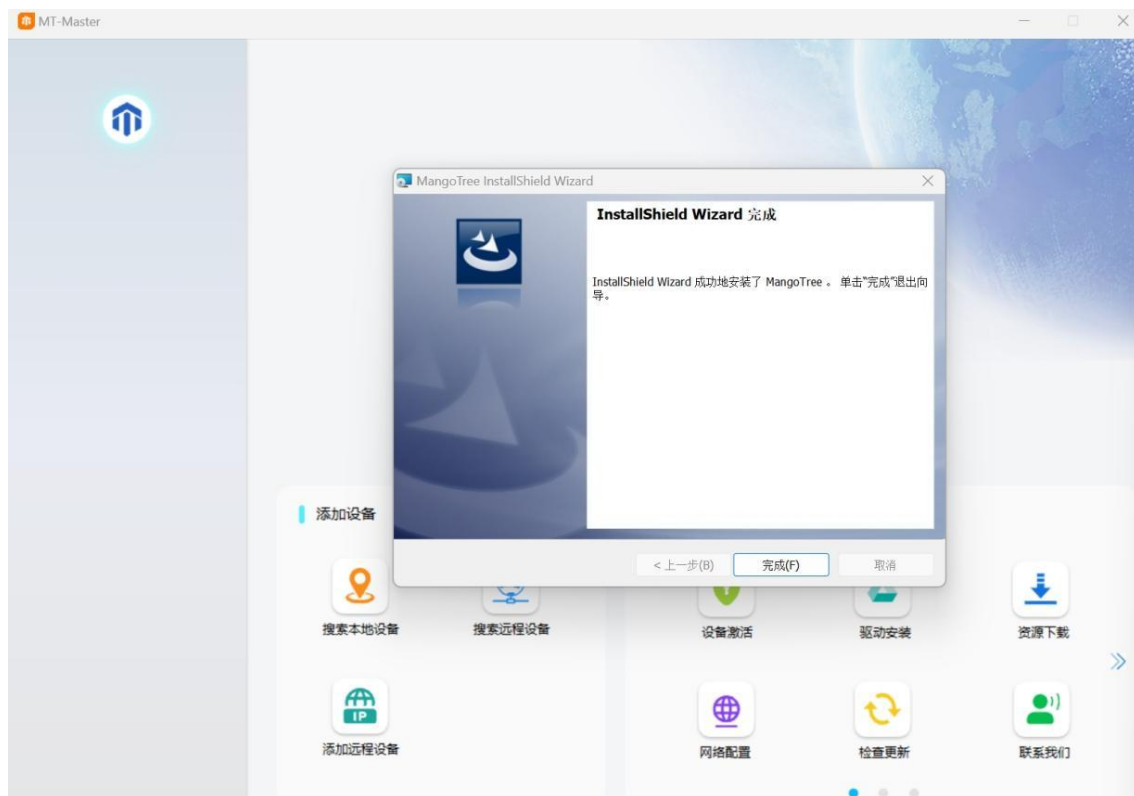
必须以管理员身份运行MT-Master软件，否则点击驱动安装图标会弹窗提示，如下图所示：



切换到主页中的技术支持，点击驱动安装图标，并根据提示完成驱动安装，如下图所示：

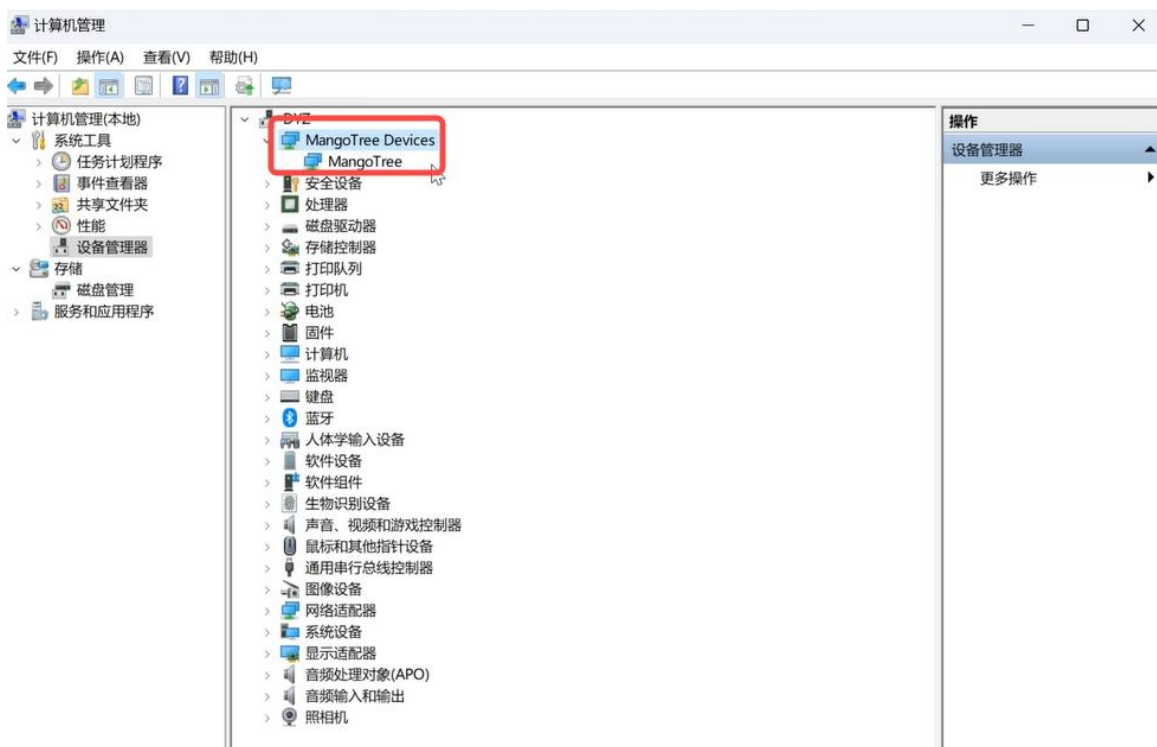






如何验证雷电驱动是否安装成功？

安装成功后，打开设备管理器，可以看到出现一台MangoTree设备，硬件连接成功，如下图所示：



### 三、开发第一个 RobustRIO 项目（RT系统）

#### 1. 连接与发现设备

Linux RT出厂IP设置是DHCP，用户需要确保RobustRIO的IP地址与PC主机的IP地址在同一个网段（需将PC也设置为DHCP），用户可以将RobustRIO通过网线直接连接上位机PC，或者通过交换机连接RobustRIO和上位机PC，正确连接之后都可以在MT-Master软件中通过搜索远程设备来发现设备，设备下显示远程设备的IP地址，如图3-1所示：

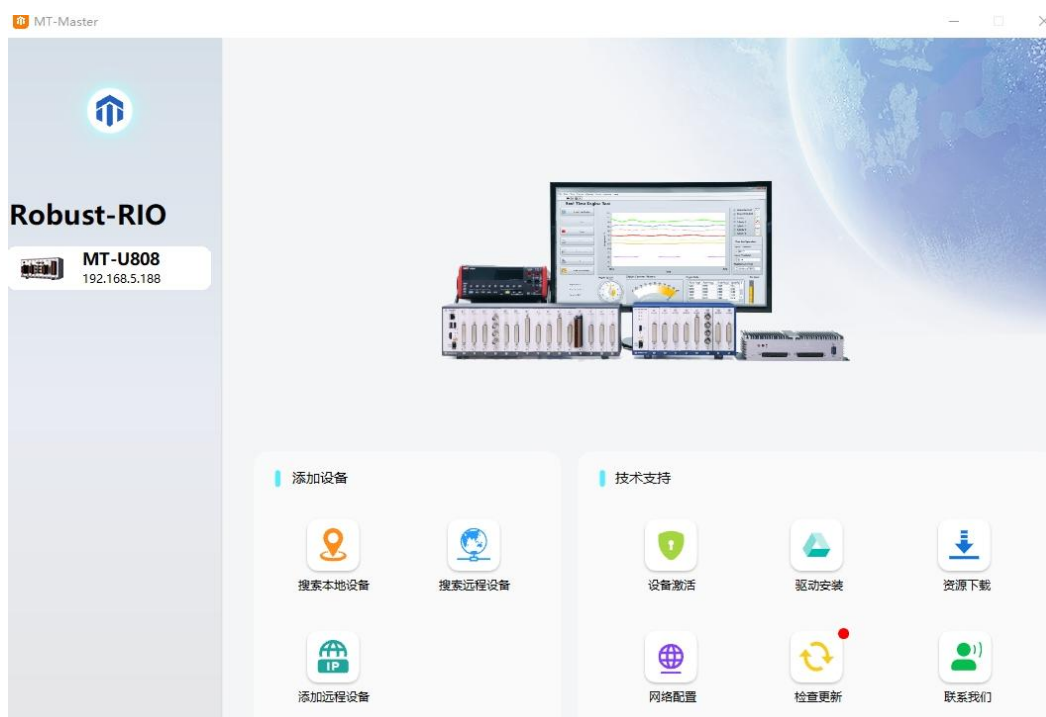


图3-1

在NI MAX中可以查看设备的操作系统，名称，IP地址和其他相关网络设置，也可以修改相关配置。如果没有找到设备，请检查设备网络接口连接是否有问题（网口灯是否闪烁）。

#### 2. 激活设备

购买后初次使用MT设备，需要用MT-Master发现后进行激活，具体激活步骤参考MT-Master上手指南：<http://server.mangotree.cn:9000/WebFile/Downloads/上手指南/MT-Master/>；

### 3. LinuxRT系统语言配置

在NI MAX中，单击选中远程系统中发现的RobustRIO-LinuxRT设备



图3-2：LinuxRT系统语言配置

LinuxRT系统语言中文和英文的区别在于是否支持UI界面，如上图所示，当LinuxRT语言环境为中文时，是无法启用嵌入式UI的，只有语言环境选择英文才能勾选启用嵌入式UI，启用嵌入式UI用户接上显示器之后可以看到上位机程序运行界面，更改语言后需要保存重启设备才能生效。

### 4. 新建LabVIEW项目

打开 LabVIEW，在LabVIEW中新建项目，如下图所示。

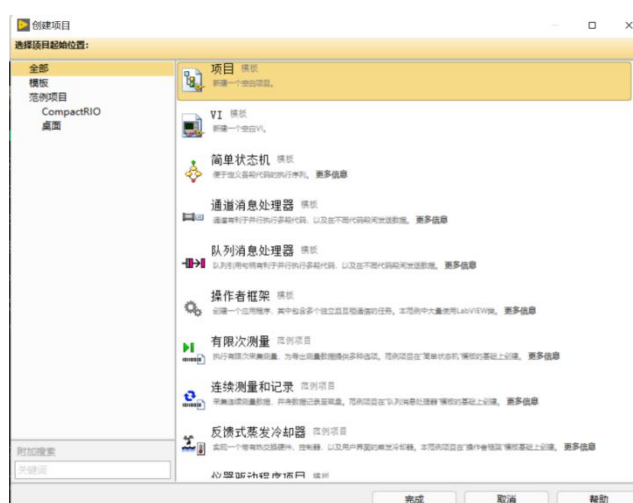


图3-3 新建一个LabVIEW项目

## 5. 新建终端设备

点击 RobustRIO项目浏览器，右击项目，选择新建一个终端如图所示。

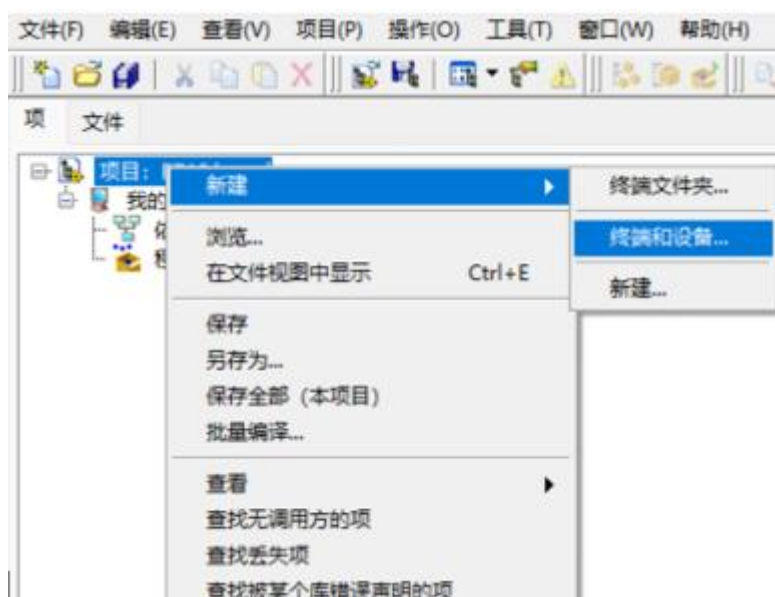


图3-4 新建一个终端设备

点击新建终端和设备，选择Real-Time CompactRIO下的RobustRIO Linux RT如下图所示。

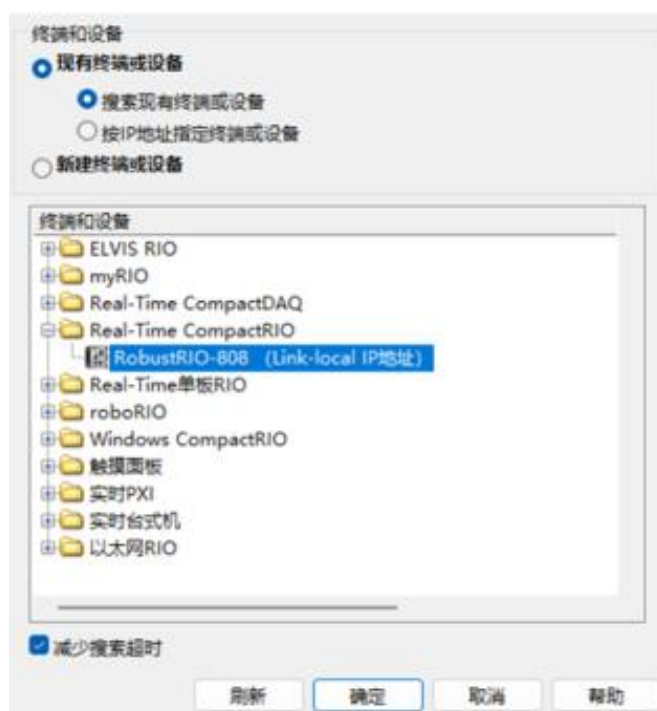


图3-5 选择RobustRIO终端

## 6. 更改设备IP地址（如需）

一般按照上述方法新建的设备，默认地址都是设备的IP地址，无需改动，但是可以通过下面的方式，确认和更改。通过终端属性选项更改IP 地址如下图所示，填写MAX中的设备IP地址。



图3-6 选择设备属性

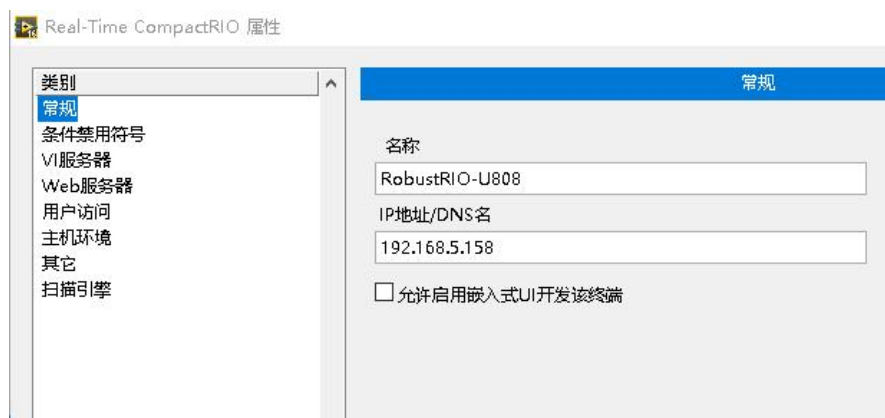


图3-7 更改IP地址

## 7. 连接RobustRIO终端设备

在项目浏览器中右击终端，选择连接，即可连接 Robust RIO 如下图所示。

注：需要确保上位机PC安装Labview语言版本与下位机LinuxRT的系统语言保持一致才能正常连接，修改LinuxRT系统语言见第二小节。



图3-8 连接RobustRIO终端

## 8. 新建RobustRIO机箱终端（以U808为例）

连接完成之后，为RT终端添加FPGA终端设备，右击RT终端RobustRIO 选择新建终端或者设备，然后选择MT-U808如下图所示。

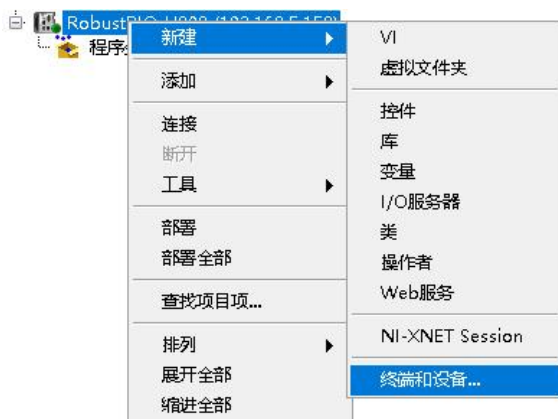


图3-9 新建U808机箱终端

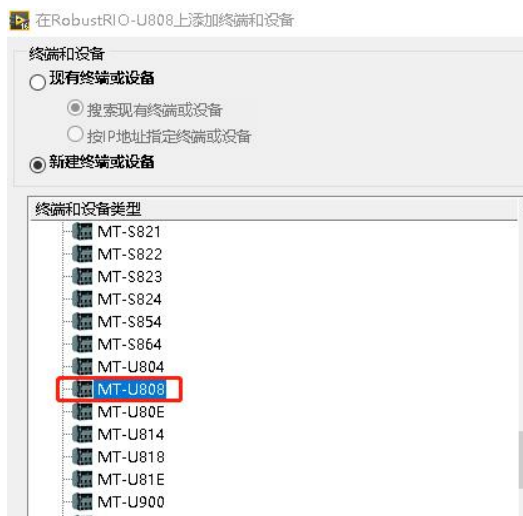


图3-10 选择MT-U808

## 9. 新建RobustRIO FPGA终端

右键U808机箱终端新建FPGA终端后如下图所示。

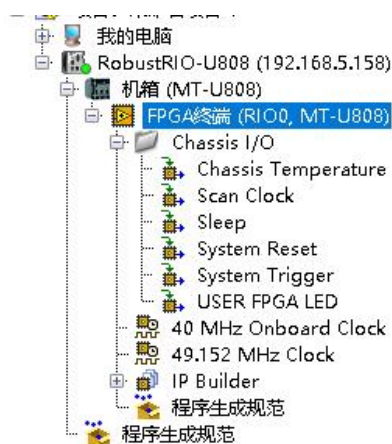


图3-11 U808 FPGA终端新建完成

总结：完成上述配置以及连接之后，我们就可以开始激动人心的第一个RobustRIO程序的开发。

## 10. 第一个Robust RIO项目！

RobustRIO支持所有的MT E系列模块及NI C系列模块，下面以RobustRIO搭配MT E710模块卡（八通道模拟量输入）为例，介绍如何开发RobustRIO。

RobustRIO的使用及开发有两种方式，分别是FPGA模式和SCAN模式，两种模式的区别在于SCAN模式无需开发FPGA程序，用户通过一个上位机程序即可快速实现板卡的采集功能，只是SCAN模式的数据交互速率较慢。FPGA模式则需要用户先开发FPGA程序，再开发上位机程序实现与板卡的数据交互，且用户可在FPGA使用DMA的方式实现高速的数据交互速率。

下面先介绍FPGA模式。

### 10.1 FPGA开发模式

首先新建一个MT E710模块，右击FPGA终端（MT-U808），新建->C系列模块，如下图所示。



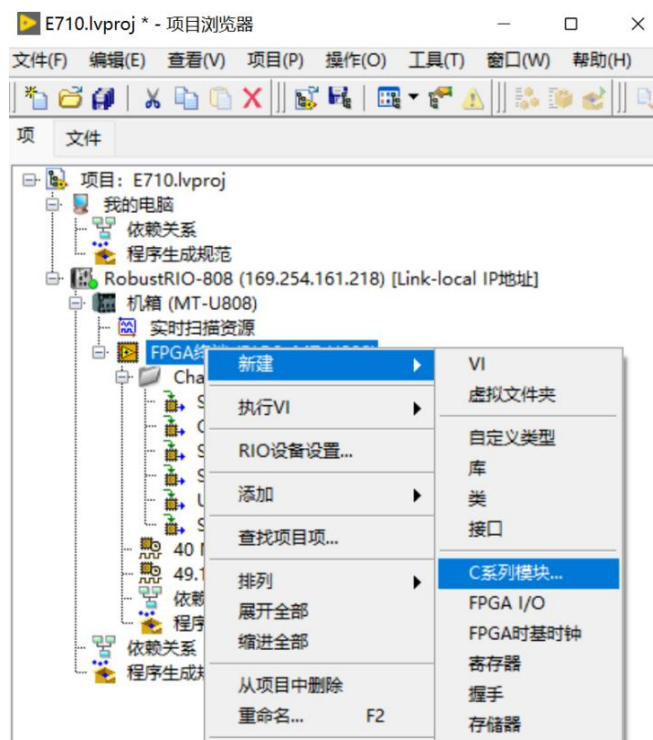


图3-12 新建模块卡



图3-13 选择MT E710及对应插槽

建好之后在FPGA终端下就会出现对应的模块及其包含的接口，E710包含了8个通道的AI，如下图所示。

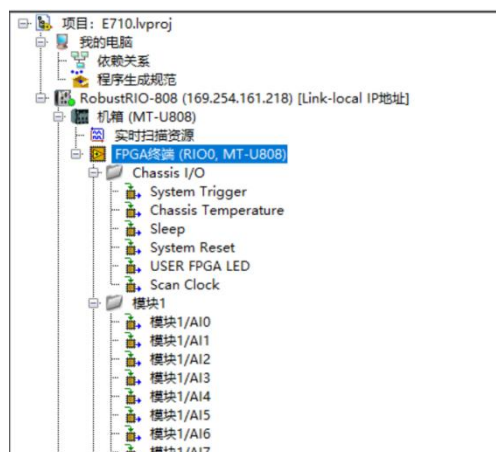


图3-14

在FPGA终端下新建VI，并保存在项目文件夹下：

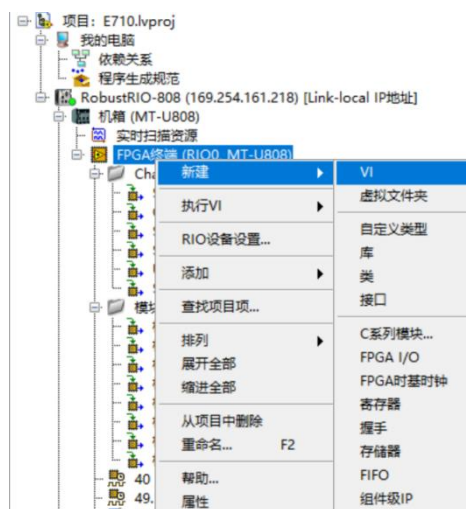


图3-15 新建FPGA VI

在后面的编程中，我们可以将模块的AI0接口拖到FPGA VI程序框图中，用于采集数据，

如下图：

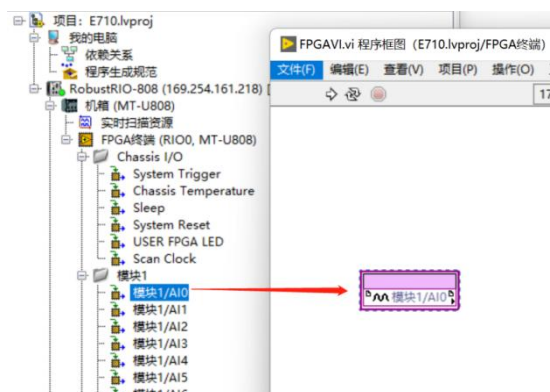


图3-16 E710 AI接口

然后下拉模块图标下方的蓝色点，得到全部8个AI通道的数据接口，如下图：

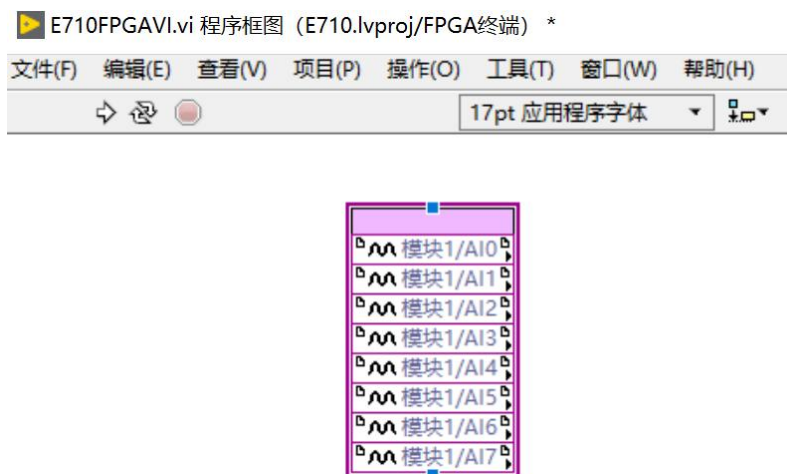


图3-17 8通道AI数据接口

在这里可以右键点击AI输出接口新建显示控件，然后右键查看该采集卡FIFO的数据属性，这里可以看到E710默认输出的数据类型为FXP类型，并且20位字长，其中5位是整数：



图3-18



图3-19

写FPGA程序，E710 FPGA VI程序框图总览：

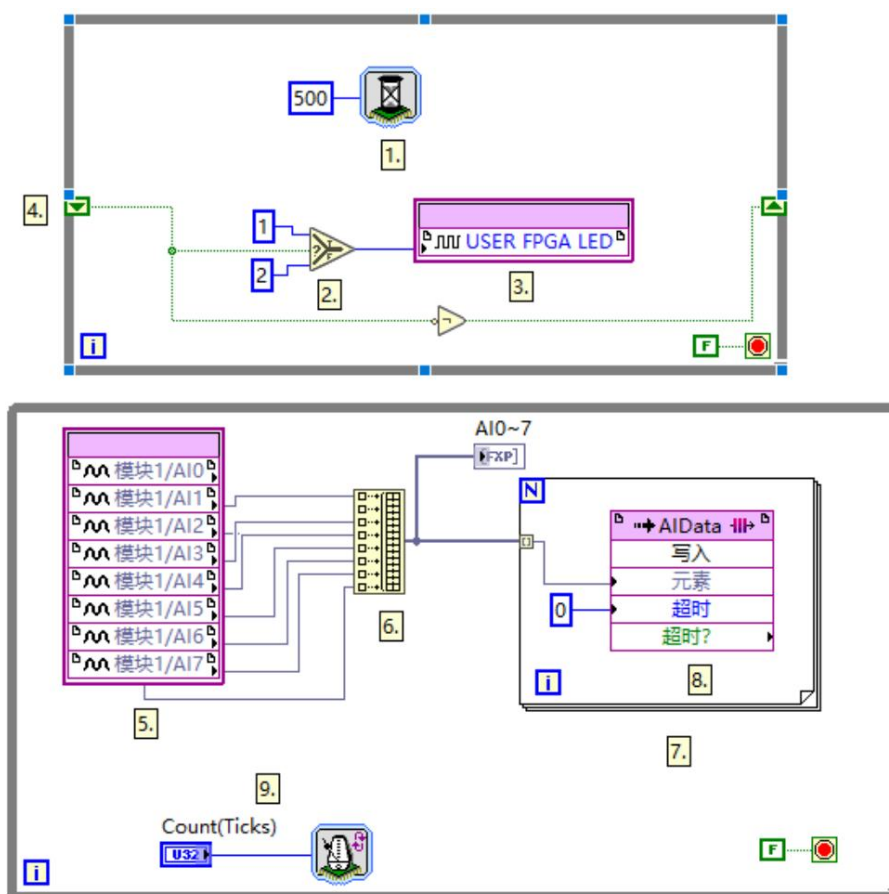


图3-20 E710 FPGA VI程序框图

程序说明（按照数字编号说明）：

1. 选版下的延时函数，此处用于控制用户灯的闪烁，闪烁间隔500ms, 设置方式如下：

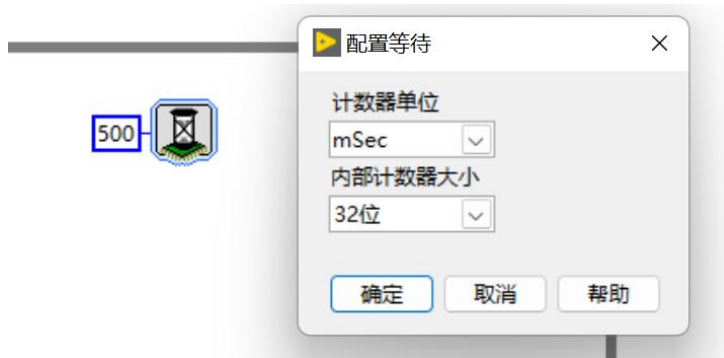


图3-21

2. 右键‘比较’选板下的‘选择’函数，此处写法用于控制灯闪烁

3. 用户灯节点

用户灯节点在项目浏览器下方Chassis I/O目录下(如下图)，在这里同样可以看到常用的温度节点，可以直接将其拖入程序框图中使用，温度节点度数是真实度数的4倍关系，通过除以4运算可得到可用的温度数据。用户灯可以呈现两种颜色，分别为：1（颜色1），2（颜色2）

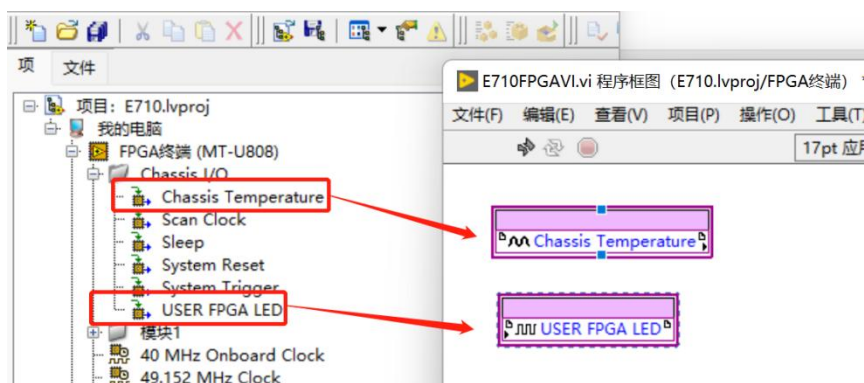


图3-22

4. 移位寄存器，右键While循环添加

5. 采集卡模块AI接口

6. 创建数组

7. FIFO节点

8. For循环

9. 循环定时器

另外，下面的While循环是实现数据采集并通过DMA FIFO的方式上传，新建DMA FIFO的方式，右键FPGA终端新建FIFO，然后命名此FIFO（此处我将FIFO命名为AIData）

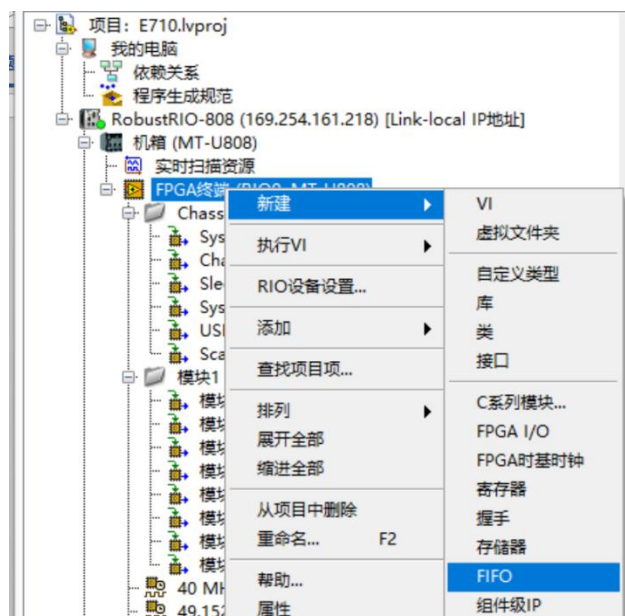


图3-23 新建FPGA FIFO

将类型改为终端至主机（因为E710为采集卡，数据由下位机传向上位机）

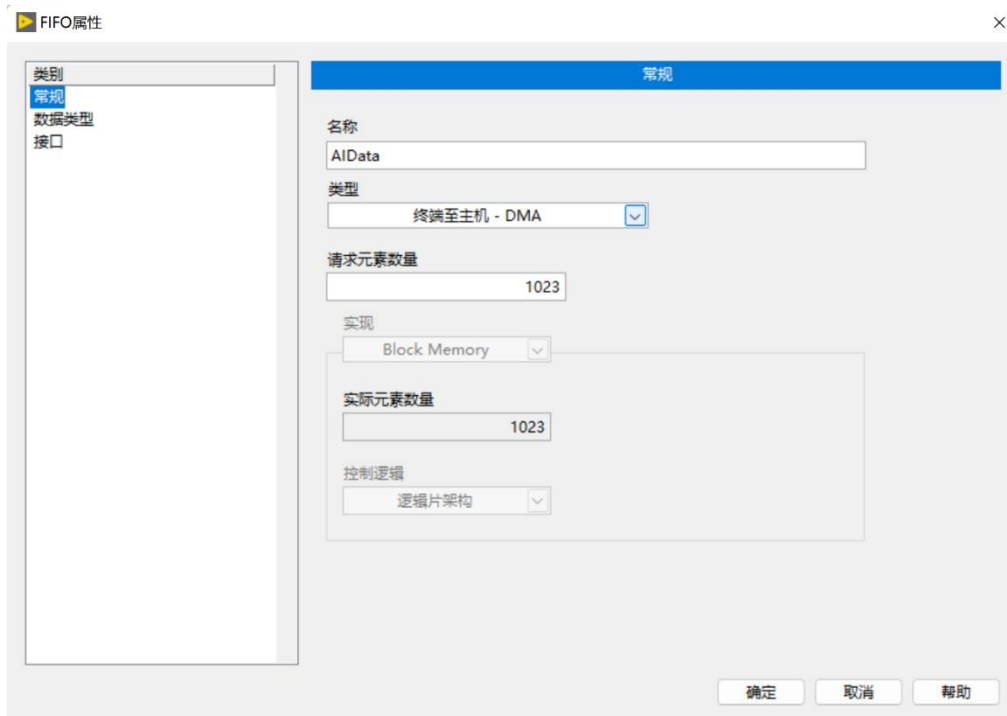


图3-24

并配置数据类型为上面查看到的板卡接口FIFO数据类型

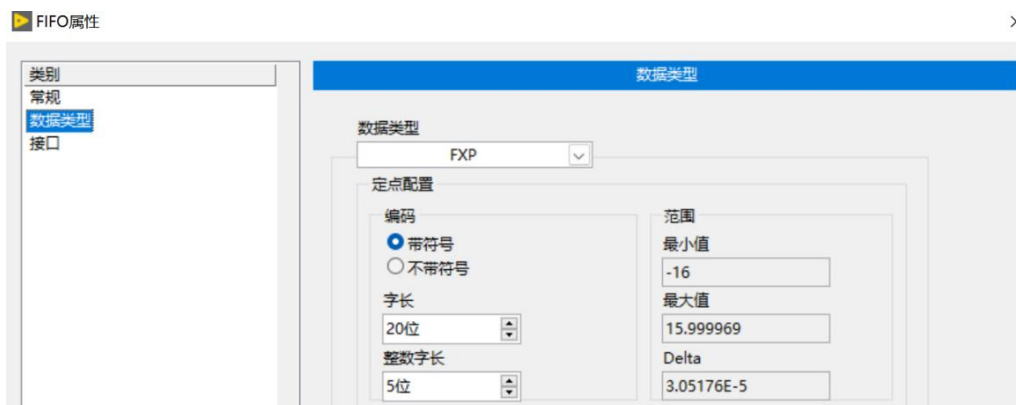


图3-25

或者通过‘数值’选板下的数据类型转化函数将采集卡输出的数据改为任意您需要的数据类型，如下图，不过需要注意，在FPGA VI和上位机VI之间的数据交互中，要保证数据类型的一致，否则会发生强制数据类型转换，消耗FPGA资源且影响传输速率

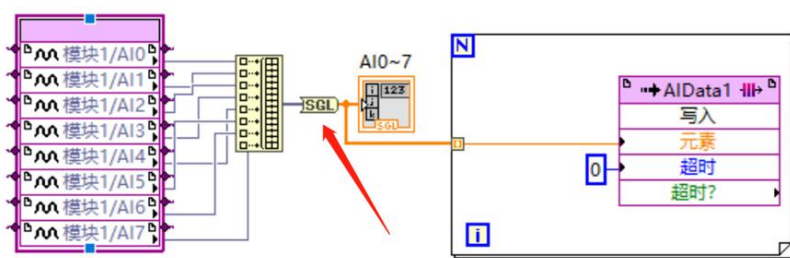


图3-26

建好之后在项目浏览器中能看到刚刚新建好的FIFO:

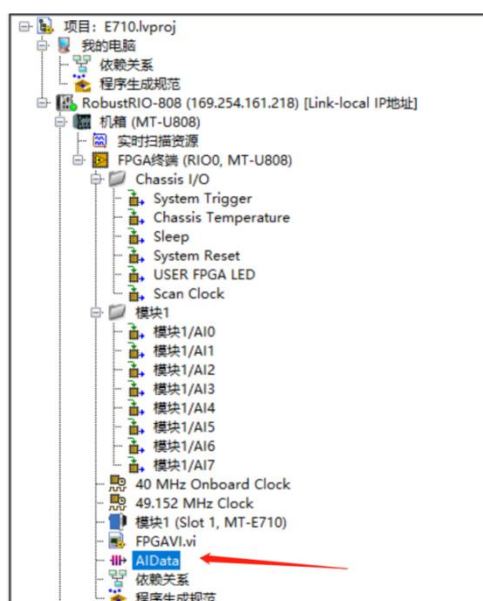


图3-27 出现新建好的FIFO



然后将采集模块接口和FIFO接口由左侧的项目目录中拖入程序框图，如下图所示：

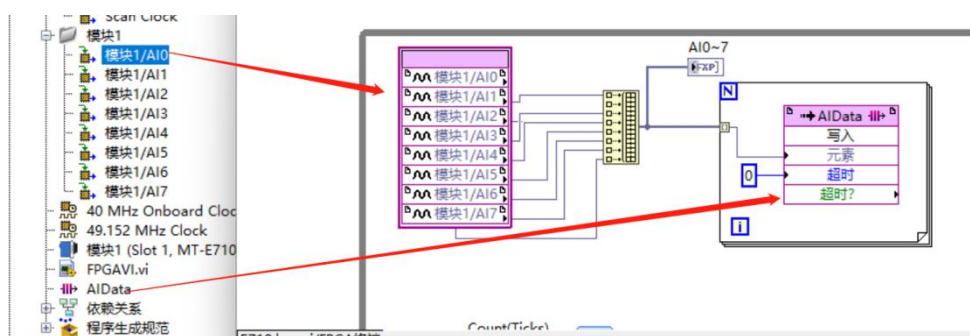


图3-28 数据写入FIFO操作

该While循环下侧的Count Ticks是用于控制采样率

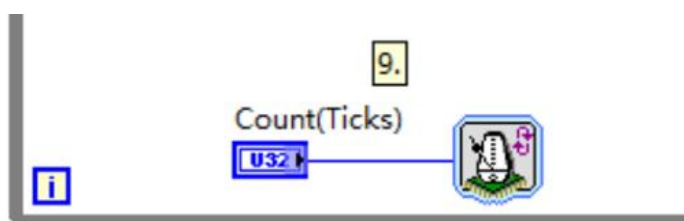


图3-29

采样率=40000000/Count (Ticks)，E710的最大采样率是200K/S/ch，即当Count设置小于等于200时，以最高采样率采样。循环定时器的属性设置如下图，单位选择滴答（tick），

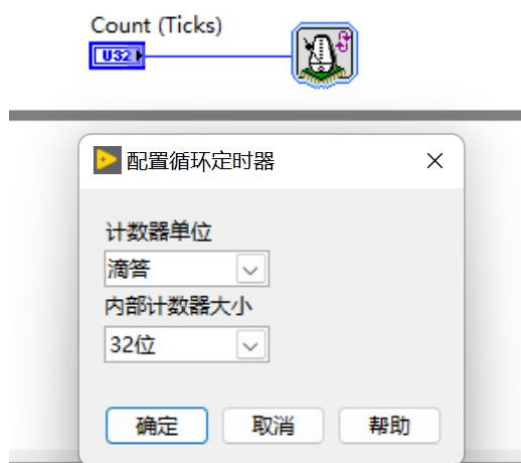


图3-30

FPGA VI程序写好之后点击左上角运行按钮，点击确认，选择编译器对FPGA程序进行编译，这里需要安装编译器才能编译



图3-31 点击运行进行编译

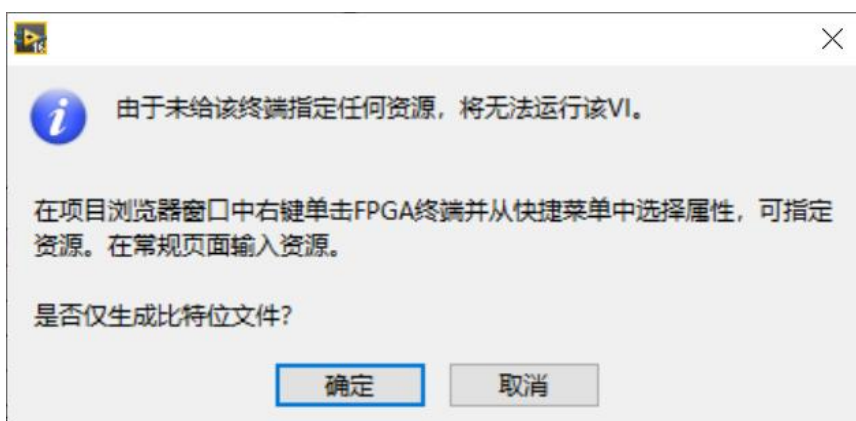


图3-32

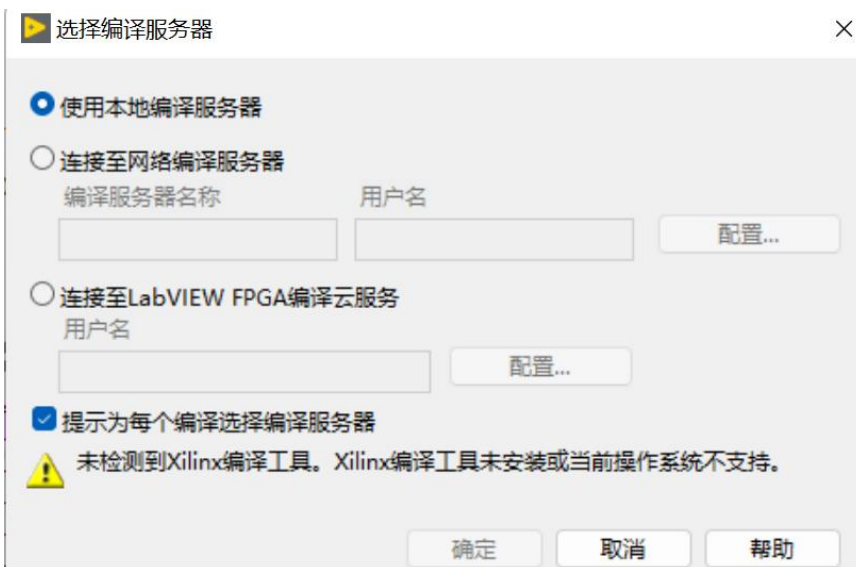


图3-33 选择编译器

编译好之后，在项目文件夹下会发现刚刚生成好的放置比特文件的文件夹：（编译好之后产生的错误1003弹窗不必在意，由于FPGA程序无法直接运行导致，请忽略）

名称	修改日期	类型	大小
FPGA Bitfiles	2022/5/24 17:58	文件夹	
E710.aliases	2022/5/24 17:33	ALIASES 文件	1 KB
E710.lvls	2022/5/24 17:20	LVLPS 文件	1 KB
E710	2022/5/24 17:20	LabVIEW Project	6 KB
E710RT.lvls	2022/5/24 17:17	LVLPS 文件	1 KB
FPGAVI	2022/5/24 17:52	LabVIEW Instru...	63 KB

图3-34

打开文件夹，看到类型为LVBITX的比特文件：

名称	修改日期	类型	大小
e710_FPGA终端_E710FPGAVI_7icGoq...	2022/5/20 14:42	LVBITX 文件	3,251 KB

图3-35

至此，FPGA程序开发已经完成，下面介绍上位机程序的开发。

在RT终端RobustRIO目录下新建VI，并保存，开始上位机VI程序（Host.vi）编写：

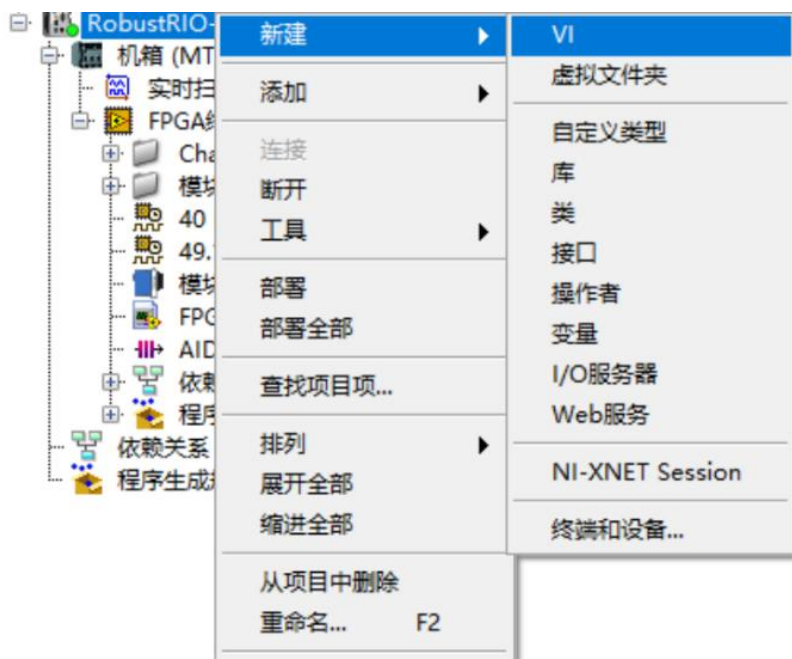


图3-36 新建上位机VI

新建和保存好之后可以看到网格状的前面板框图：

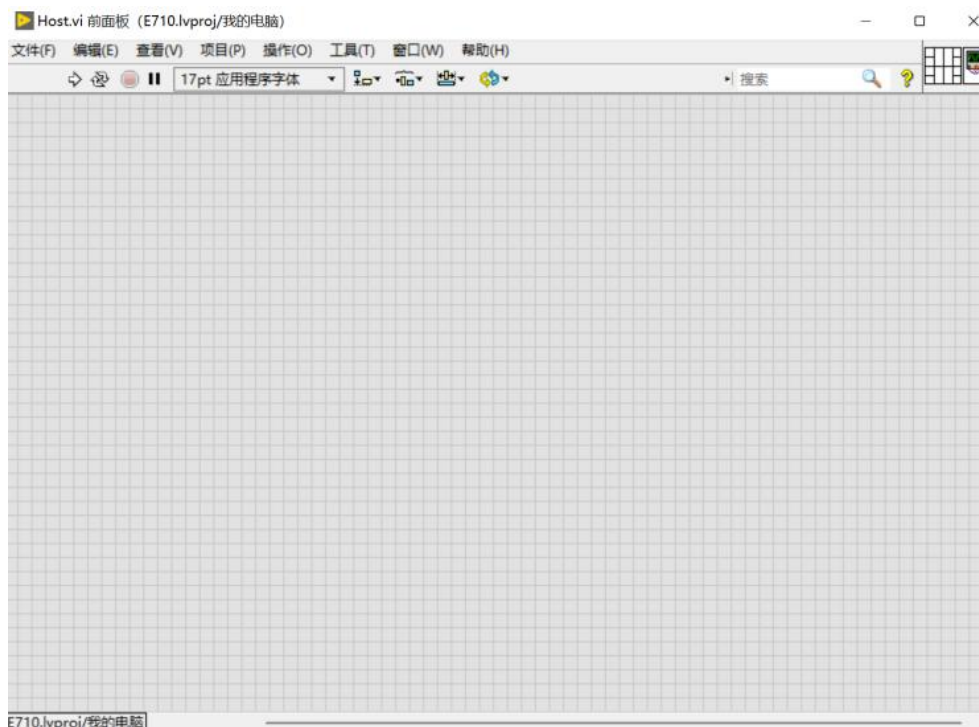


图3-37 前面板

Ctrl+E快捷键，可以在前面板和程序框图间互相切换：

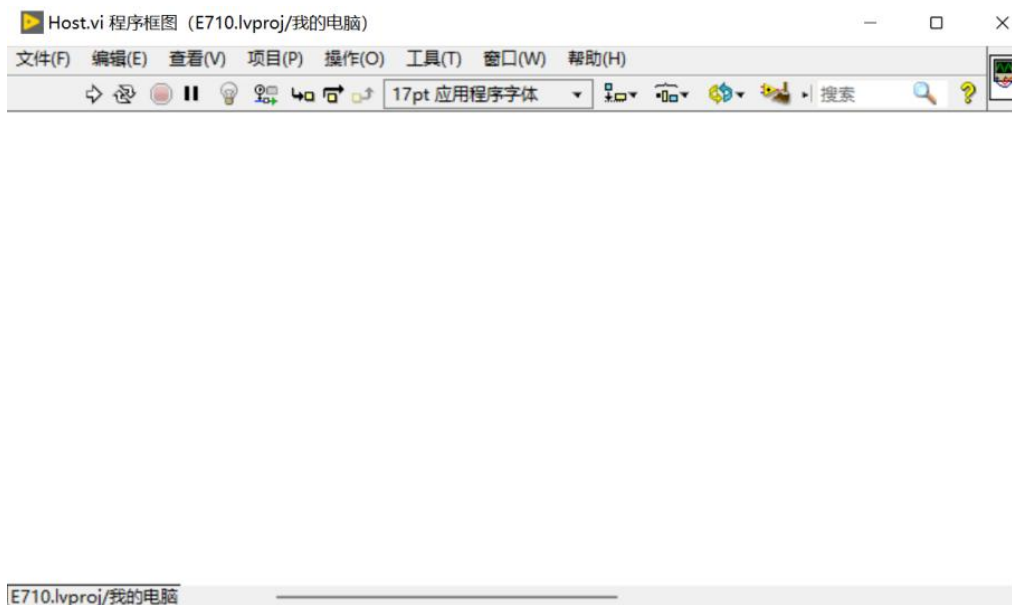


图3-38 程序框图

在程序框图空白处右击，通过下图所示MangoTree选板进行上位机程序的编写，MT提供上位机VI两种不同的程序写法，分别使用不同的函数包：

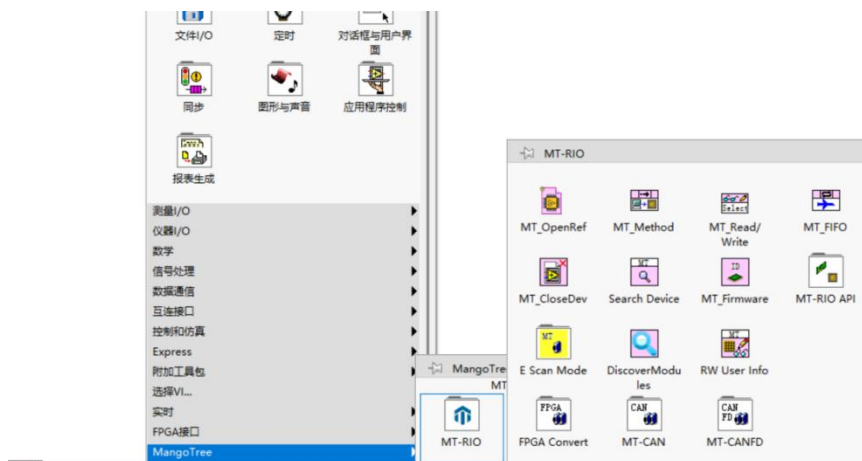


图3-39 MT-RIO函数

## 上位机VI写法A：（基础）

使用如下图所示MT OpenRef 函数，左键点击函数图标，然后点击放置到程序框图中，如图：

图：

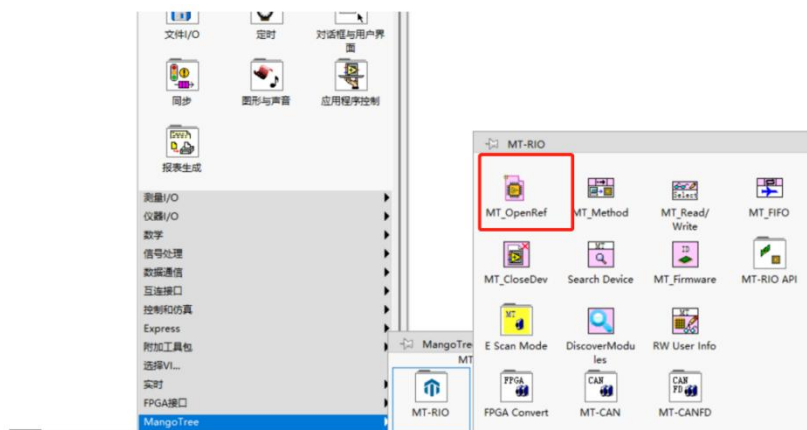


图3-40 OpenRef函数

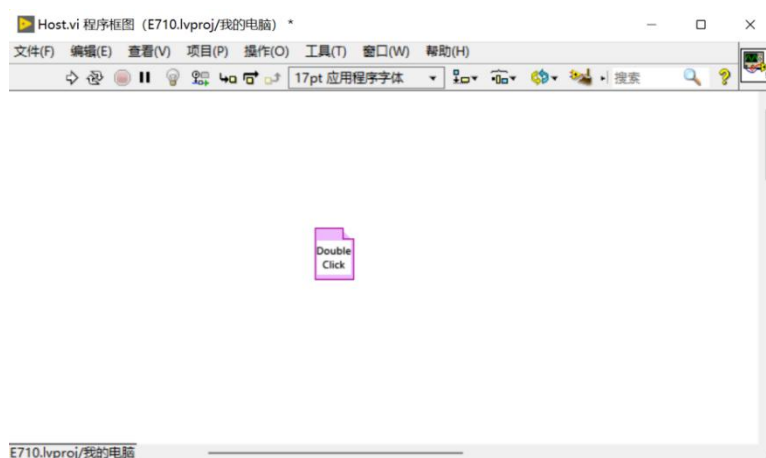


图3-41 放置函数

双击该函数图标，加载前面生成的比特文件：

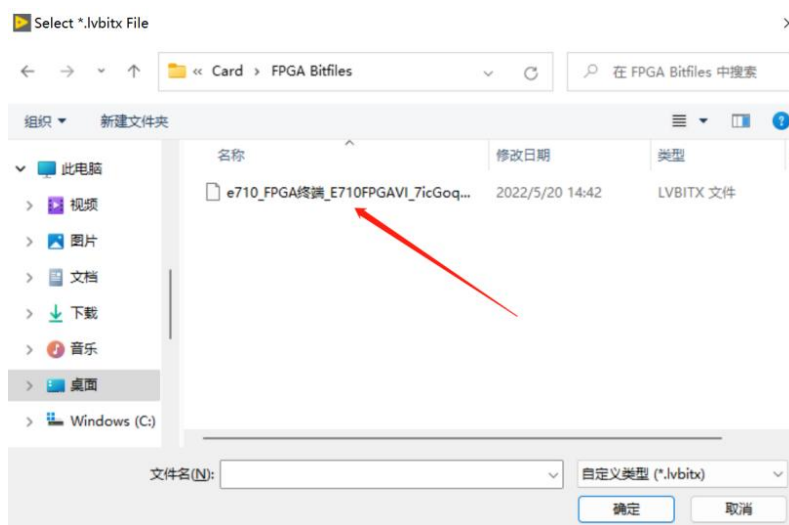


图3-42 选择bit文件

方法A的上位机程序框图全部如图所示：

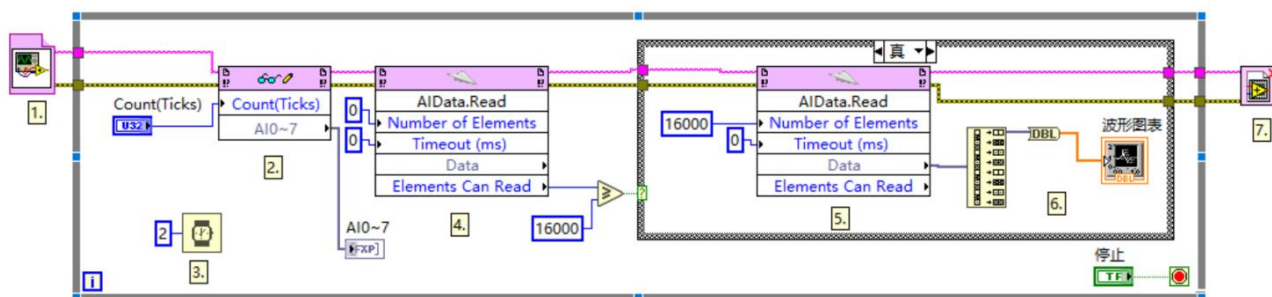


图3-43 上位机程序总览

程序说明：

1. MangoTree（后面简称MT）选板下MT OpenRef函数。

注：此处写法默认是上位机程序（Host.vi）写在RT终端目录下，如下图所示：



图3-44 RT终端目录



如果您是准备把上位机程序放在我的电脑路径下，如下图：



图3-45 我的电脑目录

则需要在Open Ref函数前填写RT终端的IP地址，如图所示：

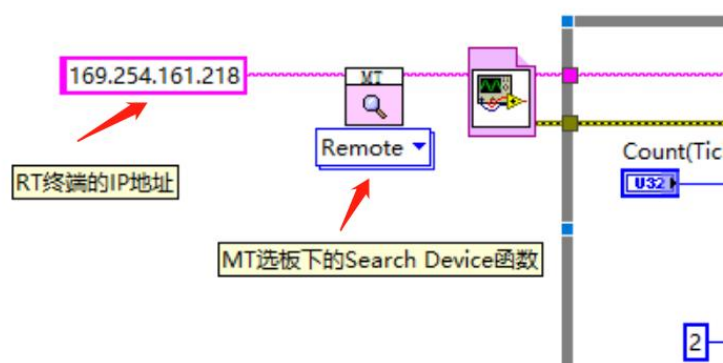


图3-46

2. MT选板的Read/Write函数，用来读写寄存器，此处用于设置采样率。

3. 为程序添加延时。

4. MT选板下MT FIFO函数，此处用于检查DMA缓存中的数据量。

5. 读取DMA中的数据。条件为真和假的两种情况程序框图如下列两图所示：（本文档后面

出现的类似条件‘假’框图都是如此连接）

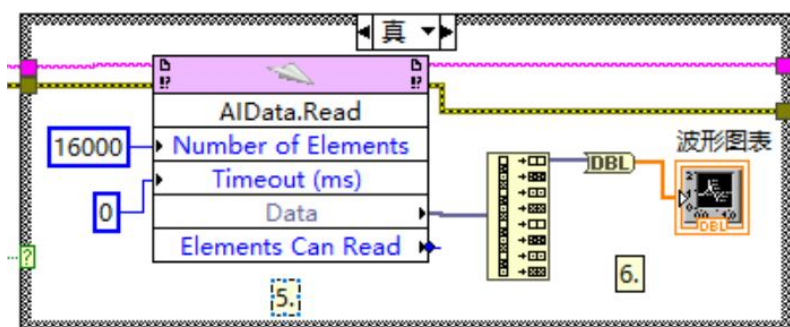


图3-47



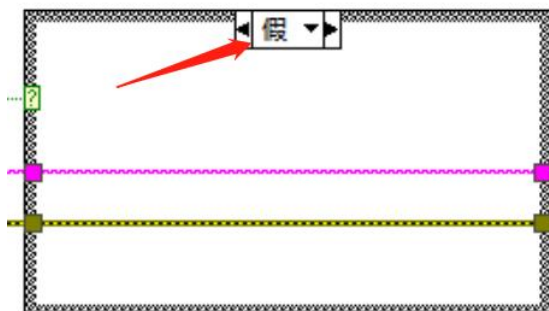


图3-48

6. 因为E710是8个通道，此处拆分出8组数据，此处演示只将其中第一个通道的数据显示到波形图表。

7. 关闭FPGA Ref。

## 上位机VI写法B：MT RIO API编程（进阶）

通过MT-RIO API选板下的函数编程，用更为底层的方式加载比特文件和读写寄存器及DMA，好处在于，程序比较复杂，涉及多层嵌套VI的项目，FPGA程序修改之后，极大的降低了上位机代码的维护工作量。另外，在比特文件发生改变后，生成好的运行程序（.exe）不需要更新，也可以直接运行。

函数包选板路径：

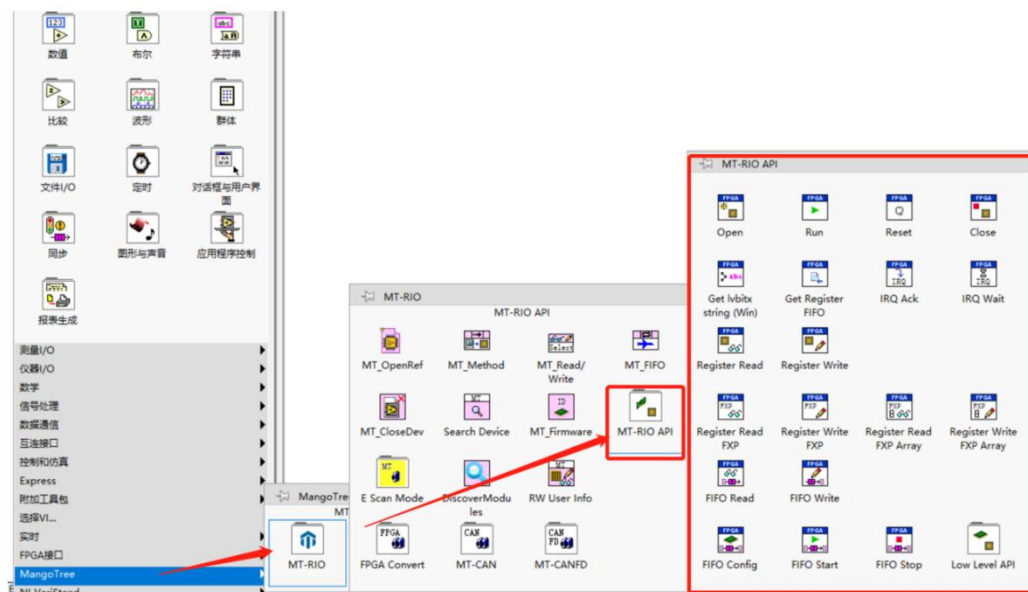


图3-49 MT-RIO API

## MT RIO API编程程序总览：

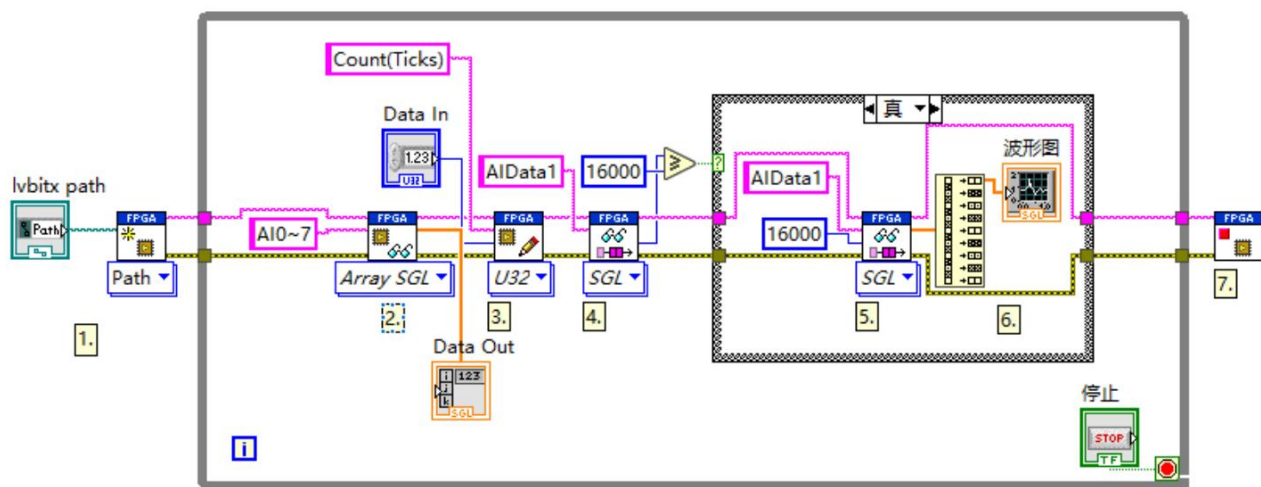


图3-50

## 程序说明：

1. MT-RIO API选板下加载比特文件路径函数，用来加载比特文件。

我们将Open函数放置到程序框图中

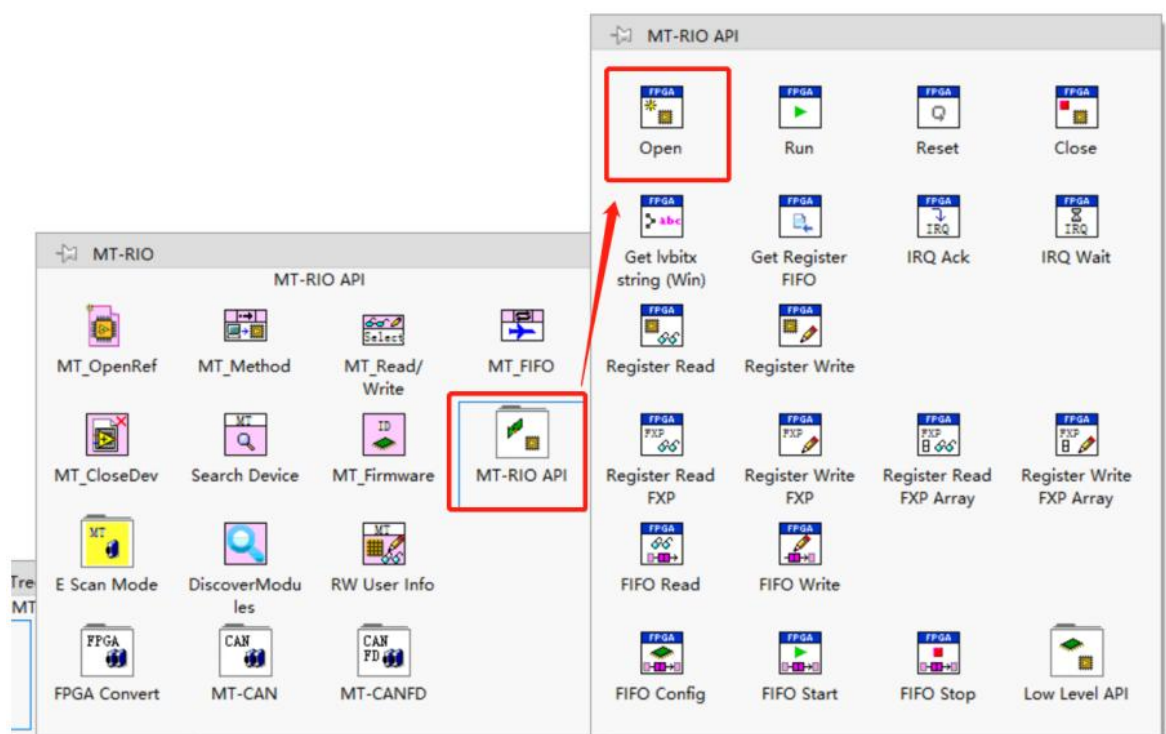


图3-51

可以发现它有两种加载比特文件的方法，分别是Path和String，如下图：



图3-52

### Path方法：

根据Host.vi运行目录位置分为两种情况。

1. 上位机程序（Host.vi）写在项目中RT终端路径下。如下图：



图3-53

具体操作是：将鼠标光标停在Open函数图标左侧的lvbitx path接口处右键点击选择创建输入控件，建好输入控件后如下图：

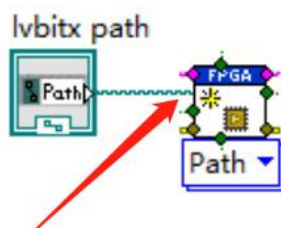


图3-54

然后双击新建的lvbitx path输入控件，会跳到前面板，在这里输入RT终端中比特文件的存放路径（需要事先将比特文件拷贝到RT终端中），**注意**：这里只能通过输入路径的方式，后方文件夹图标按钮只能浏览Windows系统路径所以不能使用。如下图：



图3-55 选择RT下存放比特文件的路径

2. Host. vi 写在**我的电脑**路径下，那么需要按照下图方式，填写RT终端的IP地址给Open函数：

数：

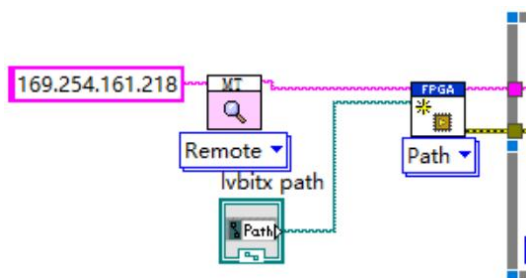


图3-56

此时，因为Host. vi和比特文件都在Windows系统下，所以加载比特文件路径时可以直接用这个按钮找到比特文件，如下图：

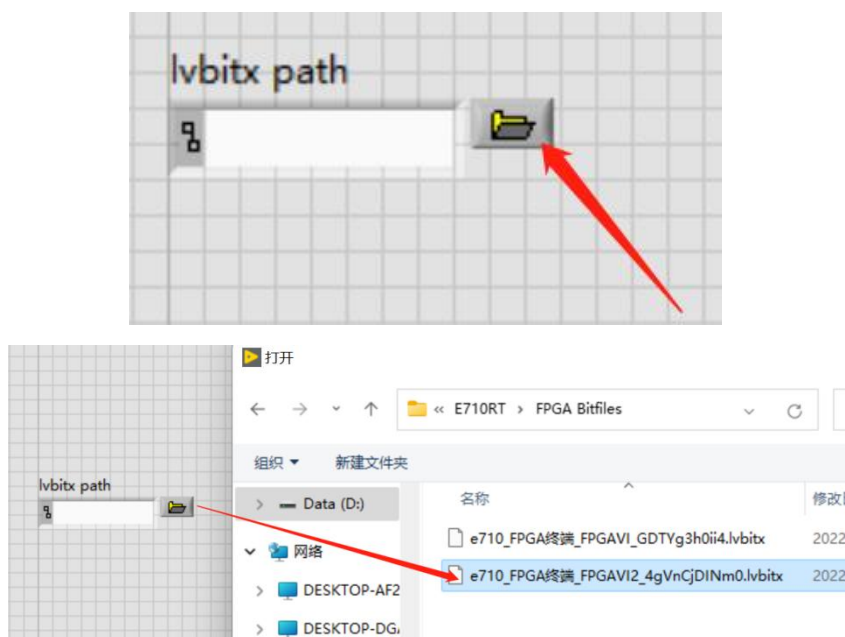


图3-57 选择Windows下比特文件存放路径

## String方法:

此方法是将比特文件转化成String字符串传递给Open函数。

同样根据Host.vi的文件位置分为两种情况。

1. Host.vi在RT终端路径目录下:

需要在我的电脑目录下新建一个VI并保存:

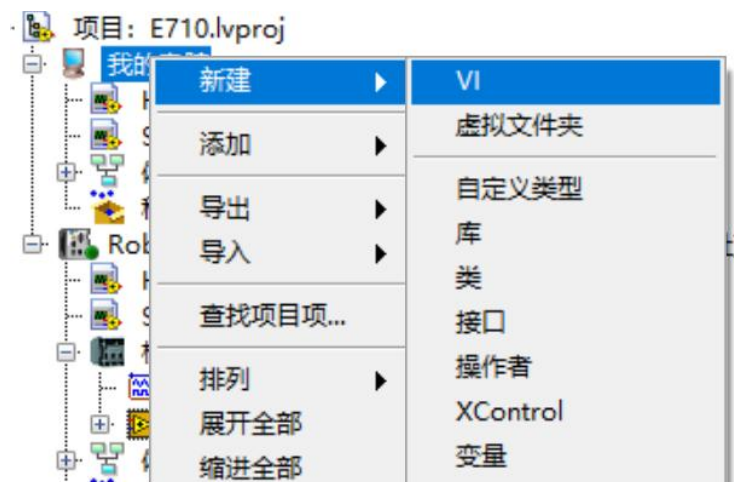


图3-58

在该VI中通过MT RIO API选板下的Get lvbitx String函数，将Windows系统中的比特文件转化成String，函数位置和程序如图所示：

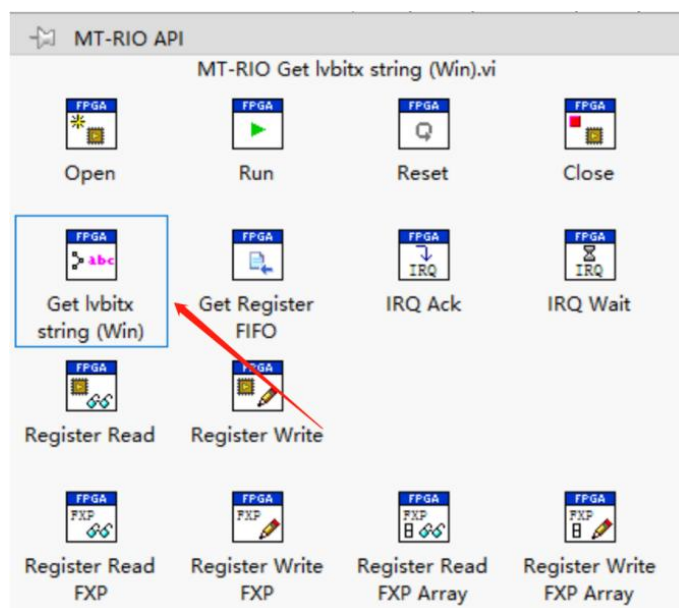
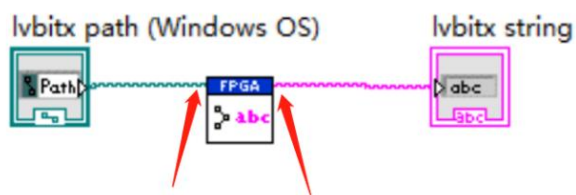


图3-59



右键创建输入和输出控件

图3-60



图3-61

点击左上角运行按钮，生成字符串：

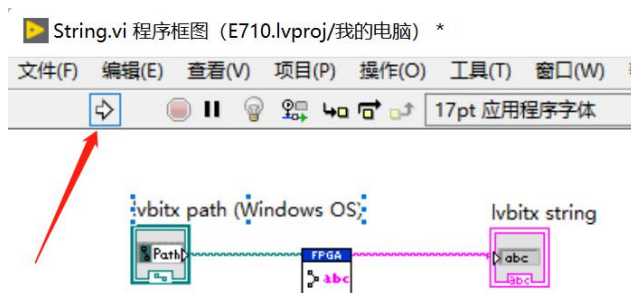


图3-62

然后我们可以在前面板看到，这里已经将比特文件路径生成String字符串：

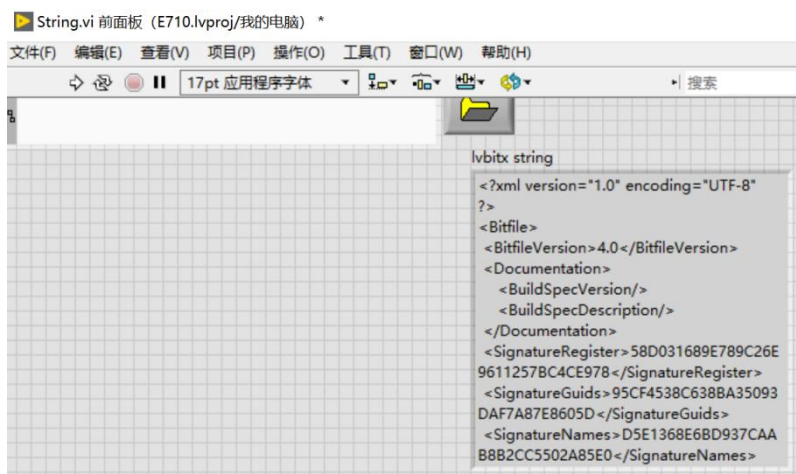


图3-63



左键单击选中显示控件，Ctrl+C 复制该控件，打开上位机程序VI，Ctrl+V将其复制到程序框图中：

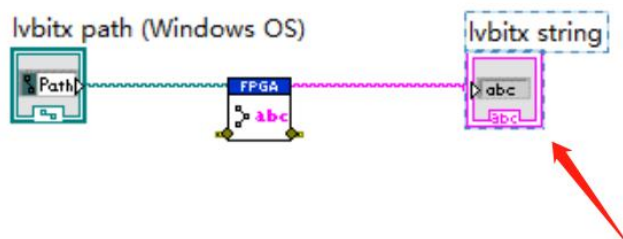


图3-64

右键点击复制好的lvbitx string控件，将其转换为输入控件，如下图：



图3-65

并将当前值设置为默认值：

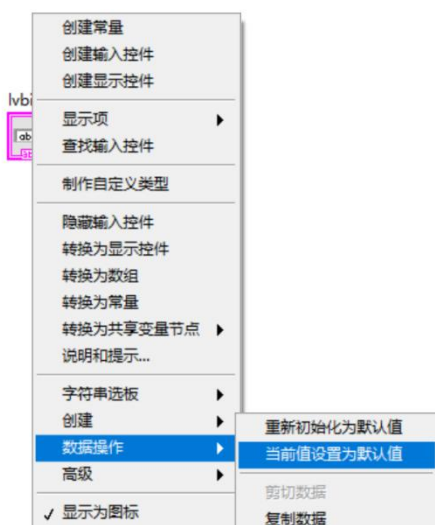


图3-66





2. MT RIO API选板下的读寄存器函数，如图：

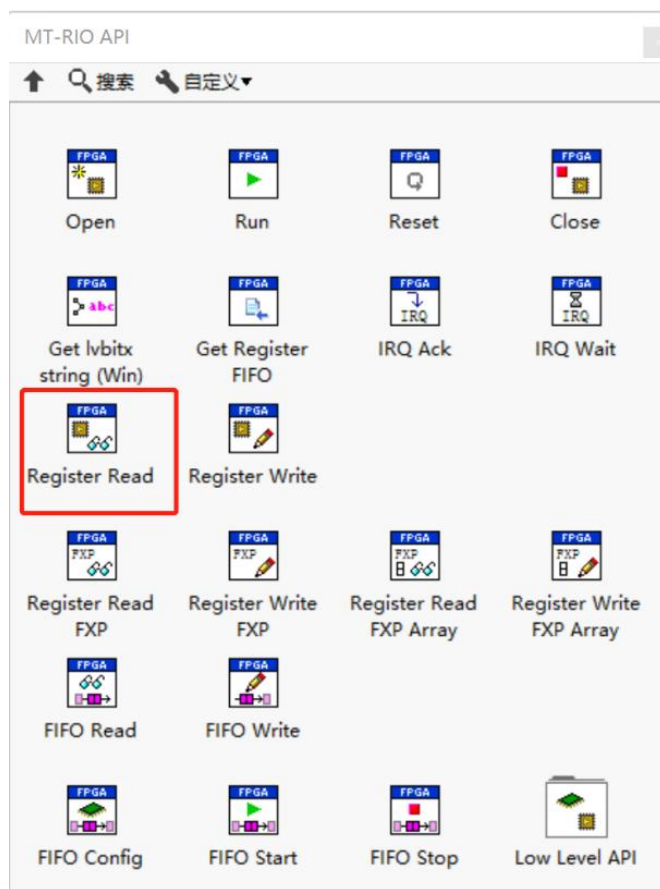


图3-70

图标左侧为String字符串，是FPGA程序中板卡采集到数据显示控件的名字，右侧右键新建的显示控件，此处是将采集到的数据实时显示在上位机前面板：

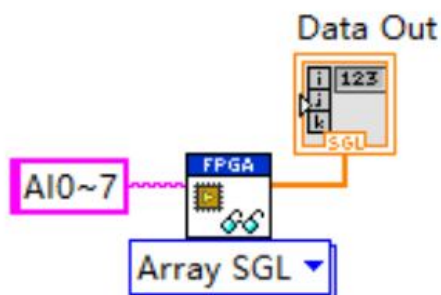


图3-71

将其放置于程序面板中，下拉其多态VI选择器，会发现其支持多种数据类型，但是不支持FXP类型数据：

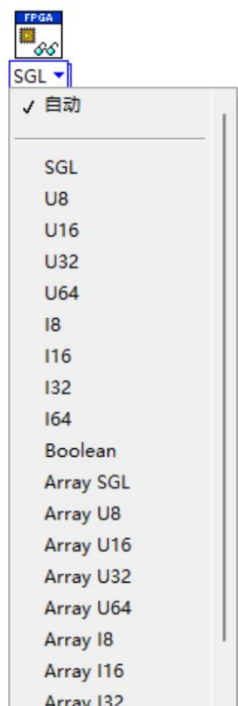


图3-72

还记得吗，上面我们的板卡默认输出的是FXP类型数据，如下图：

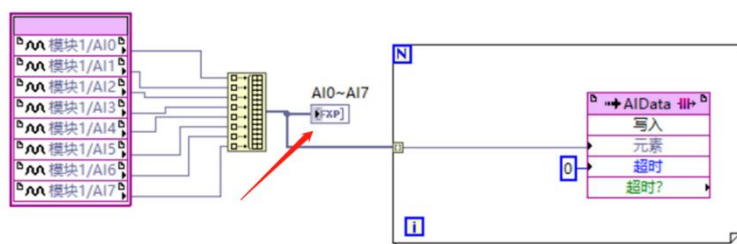


图3-73

所以在方法B这里，如果FPGA中数据类型为FXP，我们可以在写FPGA VI时将传递给FIFO的数据类型转换为其它可用的数据类型，如SGL，如下图：（此处演示改为SGL单精度浮点型）

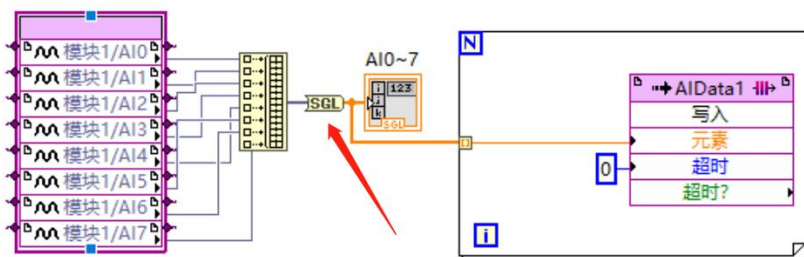


图3-74

或者利用单独的FXP读写函数（其本质上是在上位机将数据转换成所需FXP）

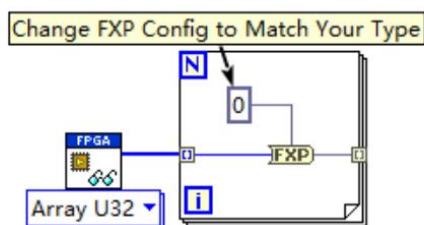


图3-75

3. MT RIO API选板下写寄存器函数，在这里是用来写入Count（ticks）的大小来改变采样率的。函数选板路径如下：

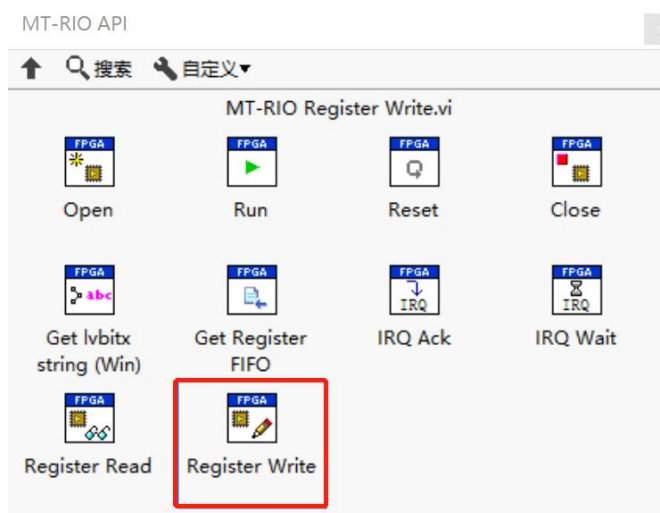


图3-76

4. MT RIO API选板下的读FIFO函数，此处用于检查DMA中的数据量

5. MT RIO API选板下的读FIFO函数，此处用于读DMA中的数据。

6. 将数据显示到波形图，因为E710是8个通道，所以拆分出8组数据，此处演示只将其中第一个通道的数据显示到波形图。

7. MT RIO API选板下Close函数

## 10.2 使用SCAN模式开发

在RT终端下新建VI并保存：

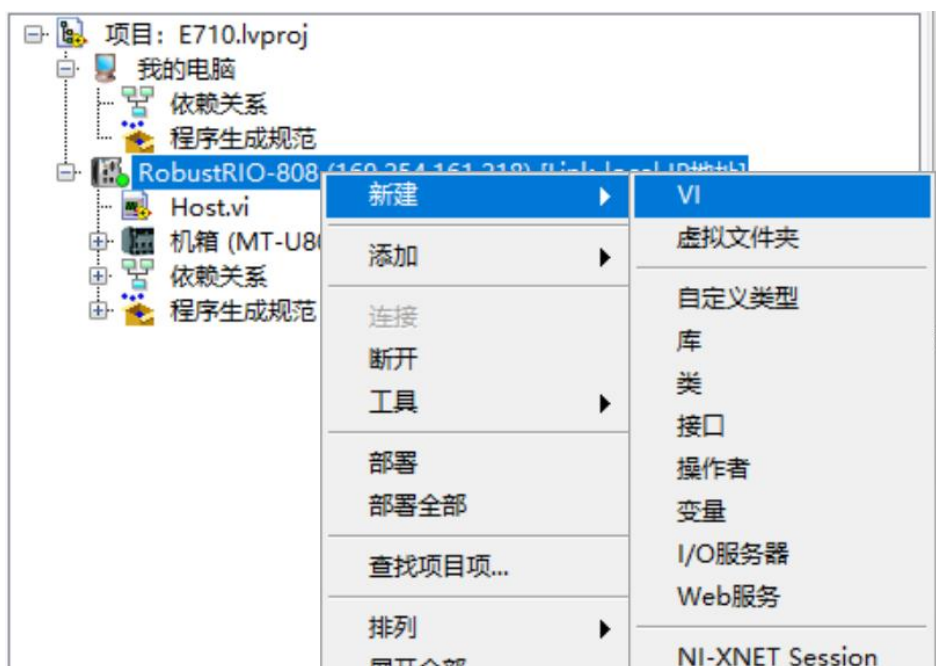


图3-77

在程序框图面板空白处右击，选择如下图所示的函数选板：

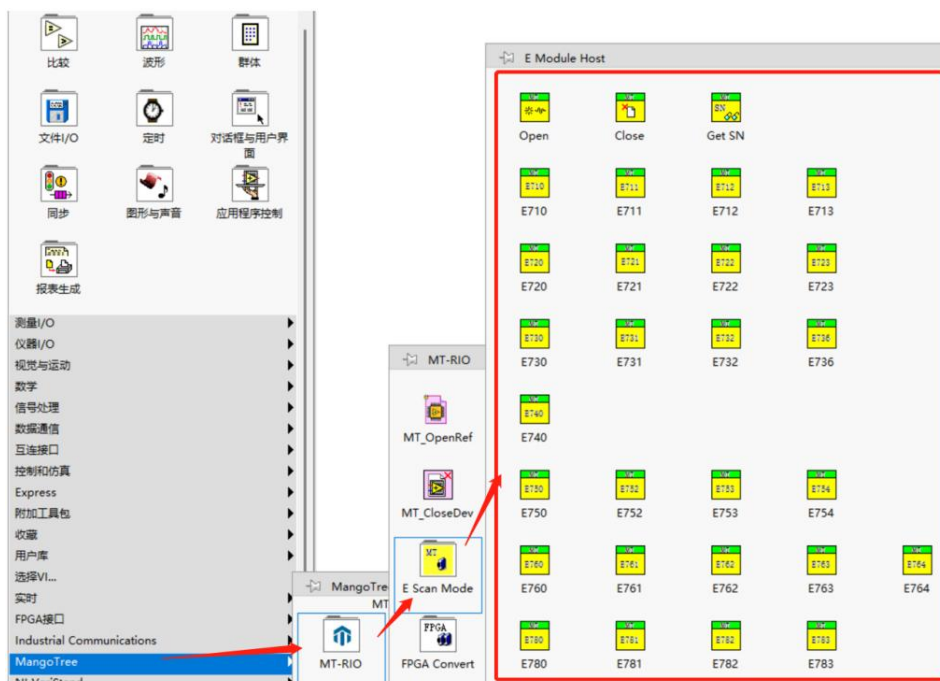


图3-78

然后开始Scan模式程序开发，全部程序框图如下图所示：

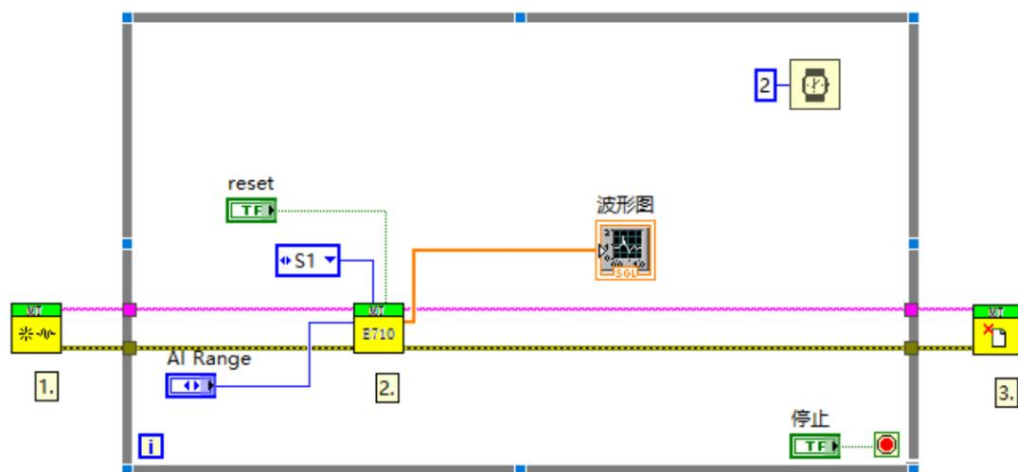


图3-79

程序说明：

1. E Scan Mode选板下的open函数。
2. E Scan Mode选板下的E710模块卡函数。

这里蓝色的S1控件是E710函数选择槽位的接口，这里我的E710插在第一槽位，所以下拉选项选择了S1。

通过MT选板下的Discover Modules函数，可以浏览所有槽位插卡的信息

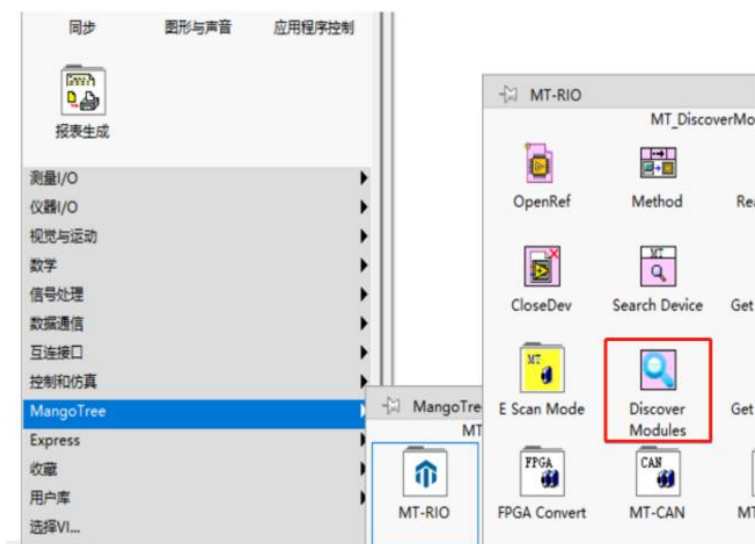


图3-80

具体步骤是：在RT终端路径下新建一个VI并保存，然后将Discover Modules函数放置，然后点击左上角运行按钮，就可以在前面板得到所有插槽的板卡型号信息，如下图：

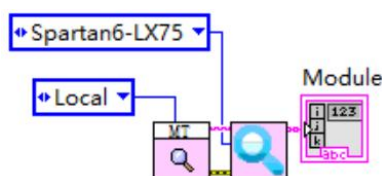


图3-81

转到前面板：

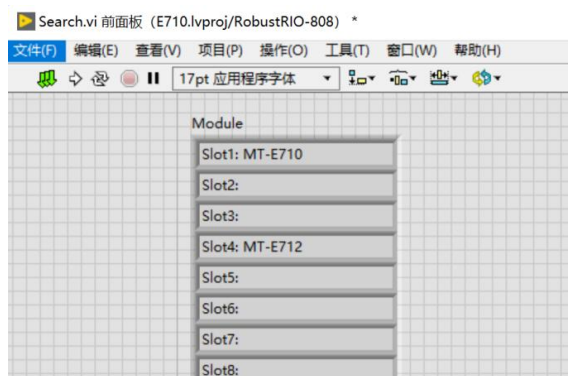


图3-82

3. E Scan Mode选板下的Close函数。

## 2. 部署RobustRIO 实时应用程序

在 RobustRIO 中生成实时应用程序，当RobustRIO上电时，将会自动运行生成的实时应用程序。首先在实时台式机终端下右击程序生成规范，选择新建，选择实时应用程序，如下图所示。

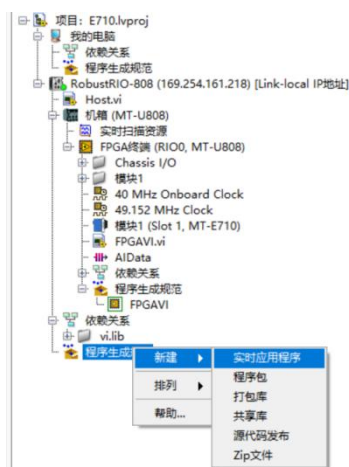


图3-83 新建实时应用程序



然后选择要生成程序的源文件，将上位机VI选择，添加到右侧启动VI，点击生成

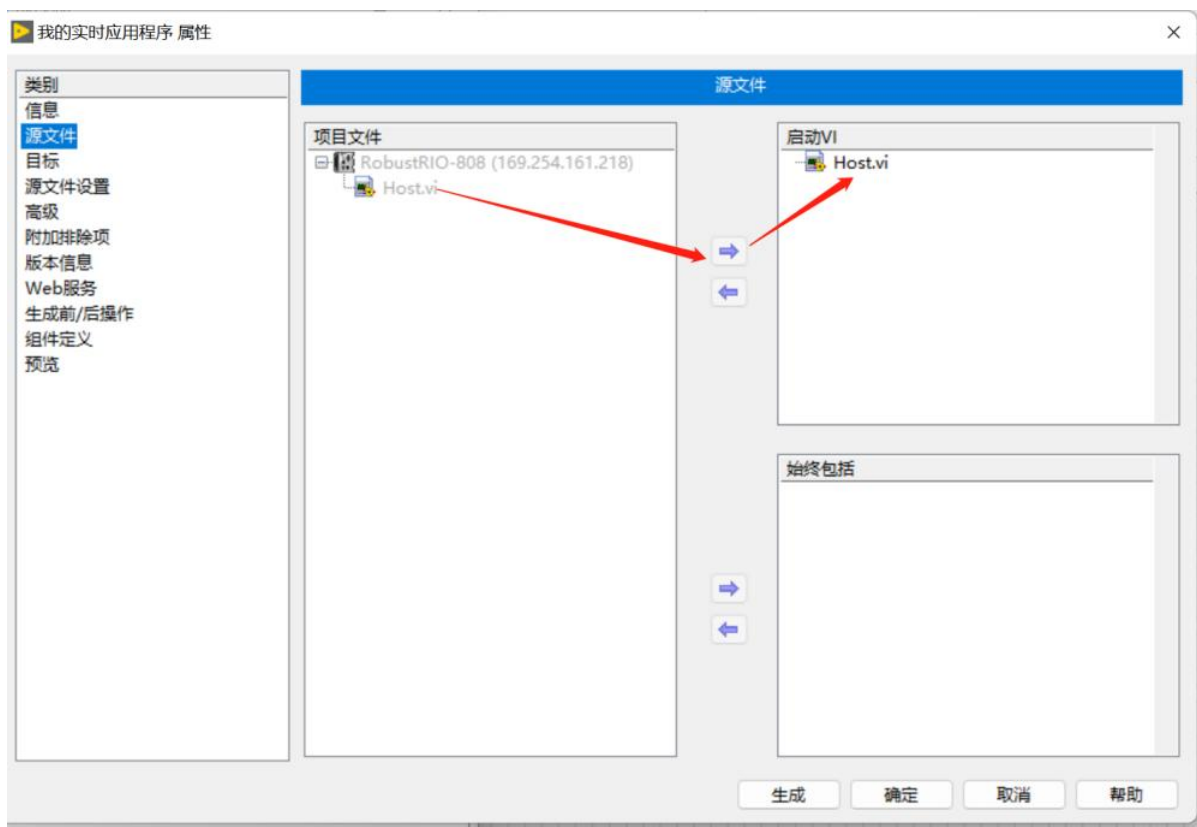


图3-84 生成实时应用程序

生成完成之后，在程序生成规范中，右击刚刚生成的应用程序，选择作为启动项运行，当RobustRIO 断电时，再重新上电，该实时应用程序会自动运行，如下图所示。



图3-85 作为启动项运行

## 四、开发第一个 RobustRIO 项目（Windows系统）

RobustRIO操作系统支持Windows和Linux RT两种，前面的使用开发流程都是基于Linux RT，下面简单介绍Windows版本的RobustRIO使用开发流程。

Windows版的RobustRIO用户开机接显示器后就可以在RobustRIO本地开发，无需借助其他PC，系统内已经安装好了全部的开发环境、RobustRIO驱动以及FPGA编译器。

开机后，首先打开LabVIEW新建工程项目，然后直接在我的电脑下新建终端设备MT-U808，如图4-1所示。

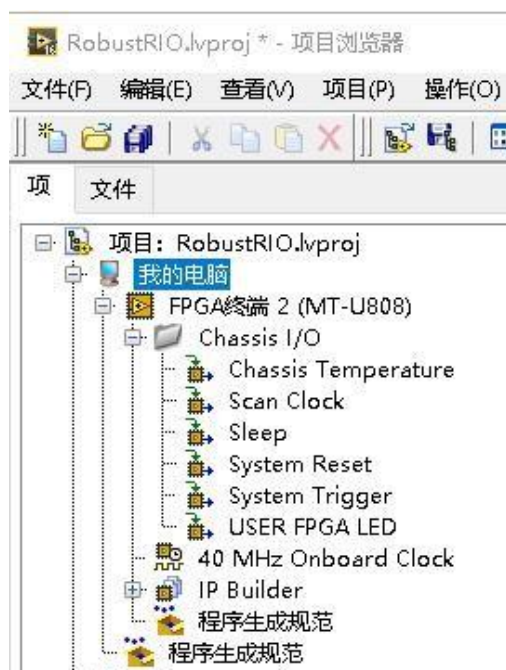


图4-1 在我的电脑下新建U808 FPGA设备

接下来开发FPGA程序及上位机程序，参见第三章的开发流程。

## 五、开发第一个 RobustRIO 项目（雷电接口）

MT雷电设备采用带有雷电标识的雷电3.0的接口，配合使用的PC机也需要带有雷电标识的雷电3.0以上接口。雷电接口版的RobustRIO用户开机通过带有雷电标识的雷电3.0以上的线将MT雷电设备的雷电接口和PC雷电接口连接，就可以在PC主机上开发，雷电标识如下图5-1所示：

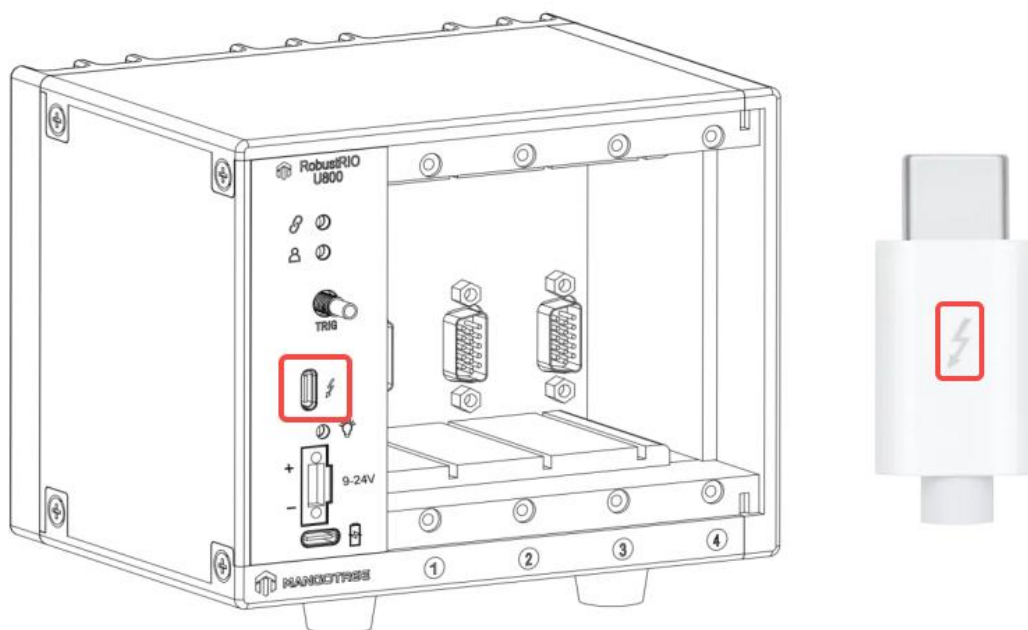
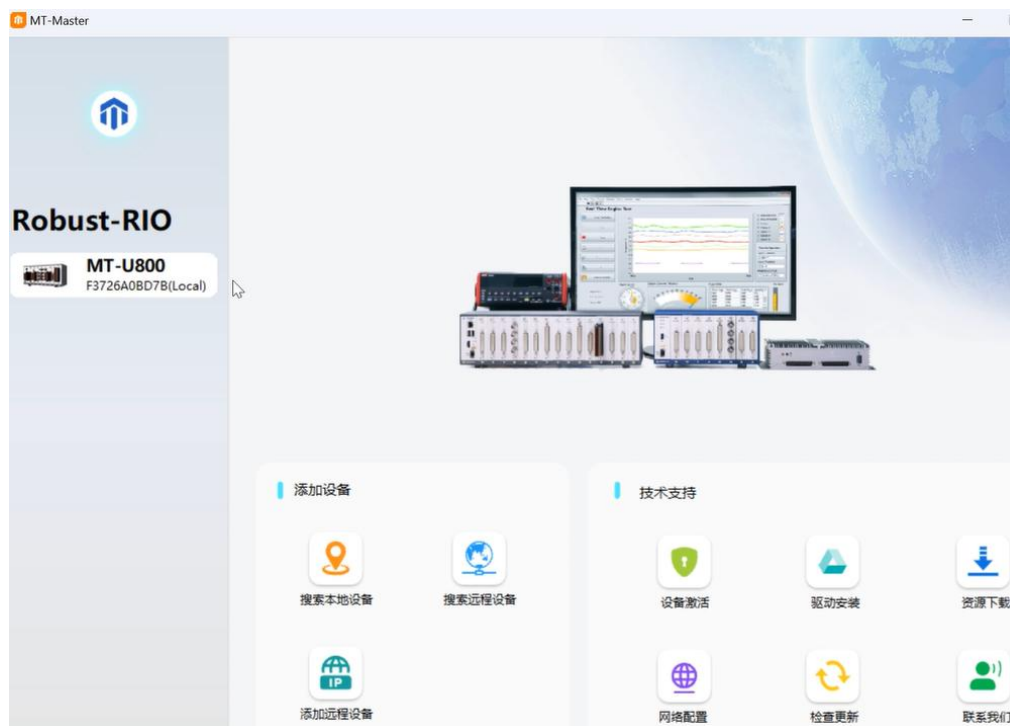


图5-1 U800雷电接口和雷电线

**注：雷电接口不是普通的Type-c接口，必须是带有雷电标识的雷电3.0以上的接口。**

在PC机系统内安装好全部的开发环境、RobustRIO驱动、雷电驱动以及FPGA编译器。安装好环境后，打开MT-Master软件，点击搜索本地设备图标，可以在左侧设备栏看到一台本地雷电设备，如下图5-2所示：



找到设备后，首先打开LabVIEW新建工程项目，然后直接在我的电脑下新建终端设备MT-U800，如图5-3所示：

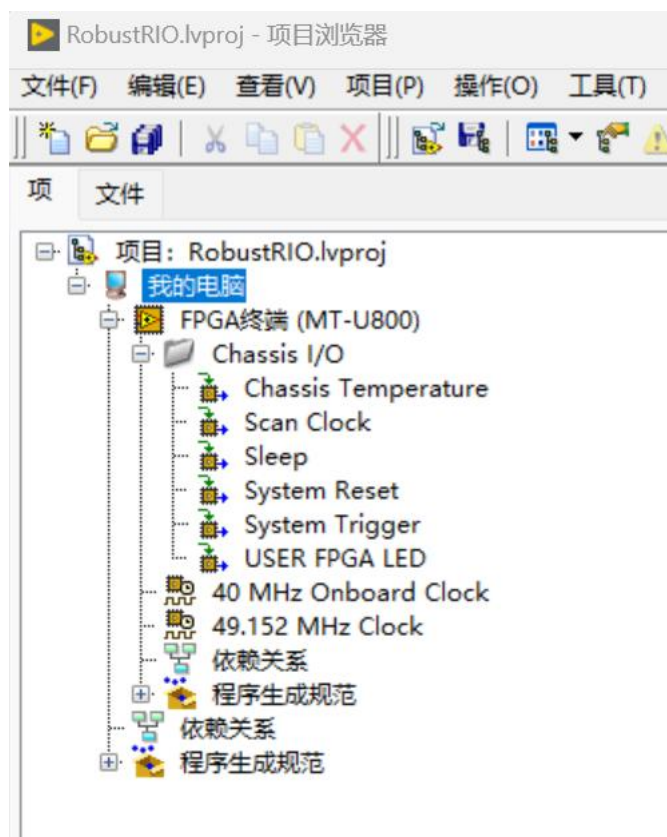


图5-3 在我的电脑下新建U800 FPGA设备

接下来开发FPGA程序及上位机程序，参见第三章的开发流程。