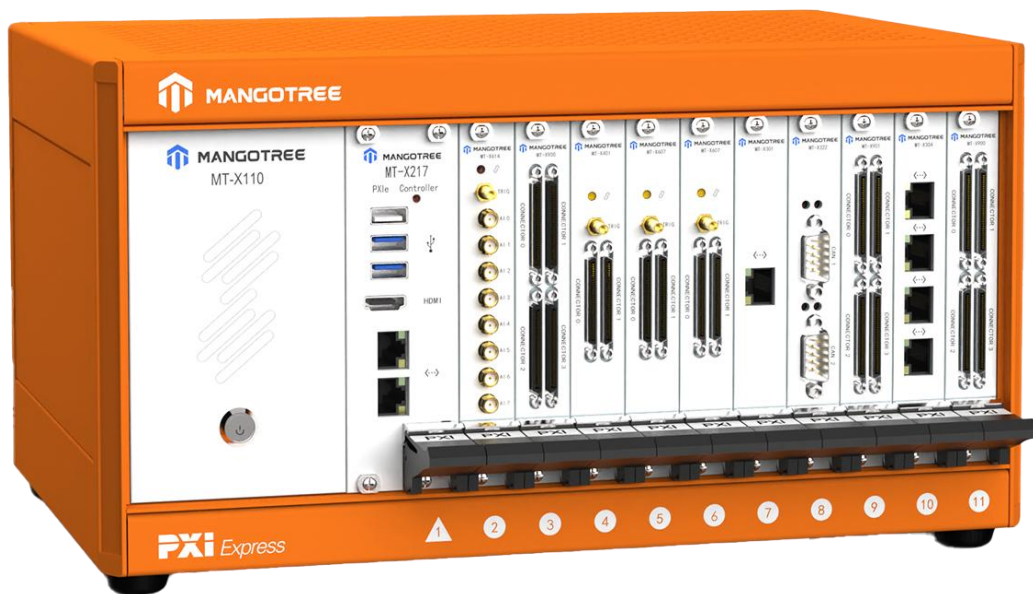


MangoTree PXIe-RIO

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PXIe-RIO项目（RT系统）	8
1. 连接与发现设备	8
2. 激活设备	8
3. LinuxRT系统语言配置	9
4. 新建LabVIEW项目	9
5. 新建终端设备	10
6. 更改设备IP地址（如需）	11
7. 连接PXIe-RIO终端设备	11
8. 新建PXIe-RIO FPGA终端（以X922为例）	12
9. 第一个PXIe RIO项目	13
10. 部署PXIe RIO实时应用程序	38
四、 开发第一个PXIeRIO项目（Windows系统）	40

一、产品概述

PXIe-RIO 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。PXIe-RIO 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 PXIe-RIO 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 MT-RIO 驱动来开发 PXIe-RIO 嵌入式系统。

用户可以通过 LabVIEW FPGA 的基本 I/O 功能，直接访问 PXIe-RIO FPGA 硬件上 I/O，通过 I/O 来进行相应的操作。PXIe-RIO 嵌入式系统由 PXIe 机箱、PXIe 控制器和 PXIe-RIO 模块卡组成。MT 提供 4 槽、11 槽和 17 槽的 PXIe 机箱可供选择，机箱槽位间电源独立，全直流供电。PXIe 控制器最高可搭载 4.9GHz 四核八线程 Intel 处理器，最高 1TB SSD，16GB 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。PXIe 板卡由前端 IO 模块和板载 FPGA/ASIC 组成。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 PXIe-RIO，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PXIe-RI0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PXIe-RI0产品系统为Linux RT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/



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：

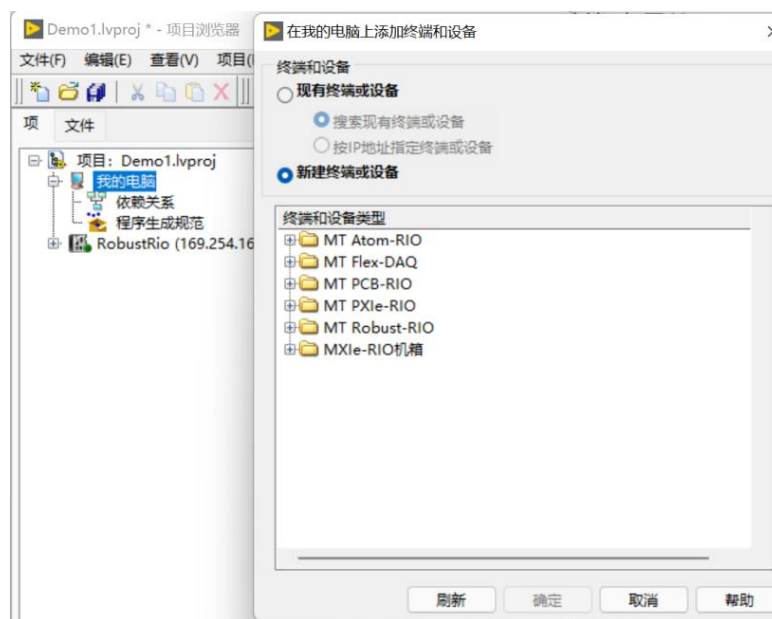


步骤三：MT RIO驱动安装

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/

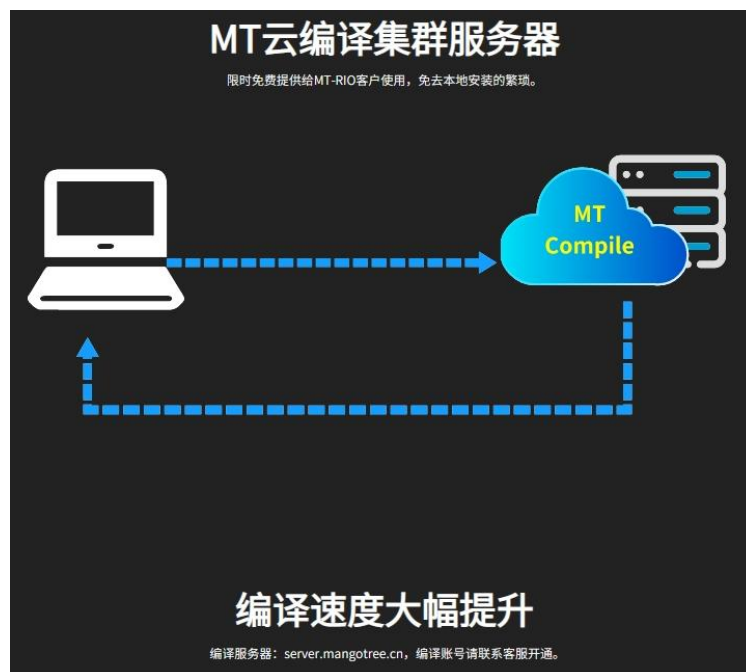
07-Xilinx ISE 14.7 2020.iso

08-Vivado 2019.1 2020.iso

如上图提供两种编译器，购买PX1e-R10设备安装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-R10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：compile.mangotree.cn，编译账号请联系客服开通。可以通过Master软件首页MT-R10栏下的编译器获取。

三、开发第一个PXIe-RIIO项目（RT系统）

1. 连接与发现设备

我们以X110机箱配备X217控制器和X922 PXIe-RIIO板卡为例，X217控制器内置Linux-RT系统，Linux RT出厂IP设置是DHCP，用户需要确保X217控制器的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X217控制器通过网线直接连接上位机PC，或者通过交换机连接X217控制器和上位机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：

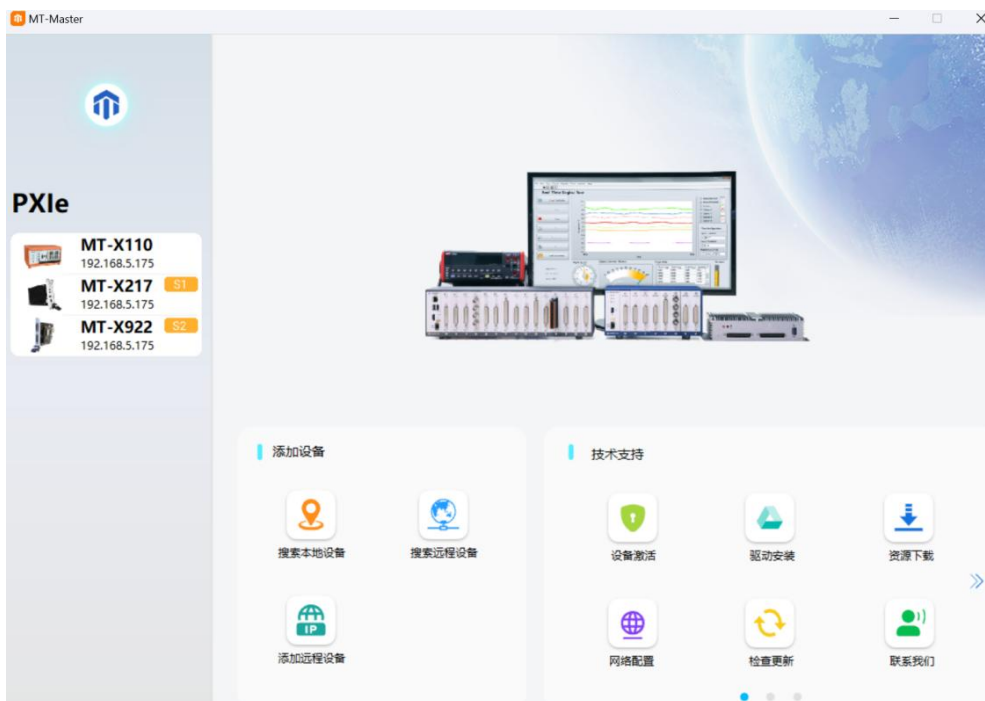


图3-1

在NI MAX中可以查看X217控制器设备的操作系统，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X217控制器(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。

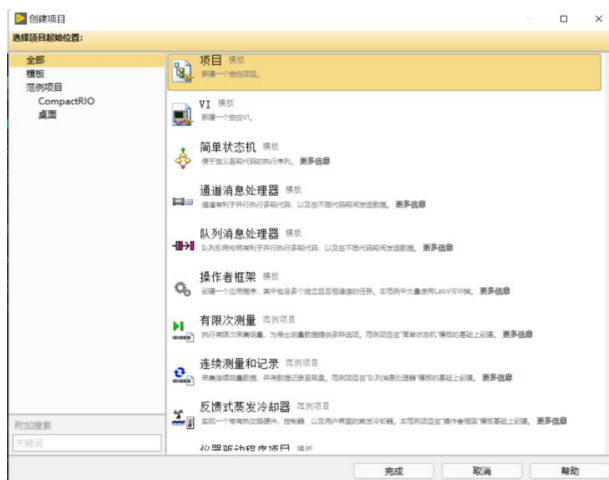


图3-3 新建一个LabVIEW项目

5. 新建终端设备

点击 PXIe-RIO项目浏览器，右击项目，选择新建一个终端如图所示。

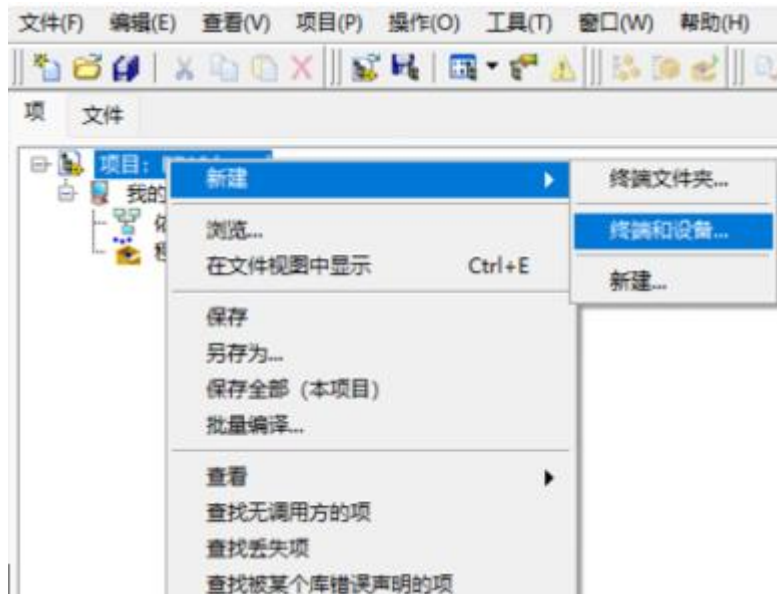


图3-4 新建一个终端设备

点击新建终端和设备，选择实时PXI下的X217控制器如下图所示。

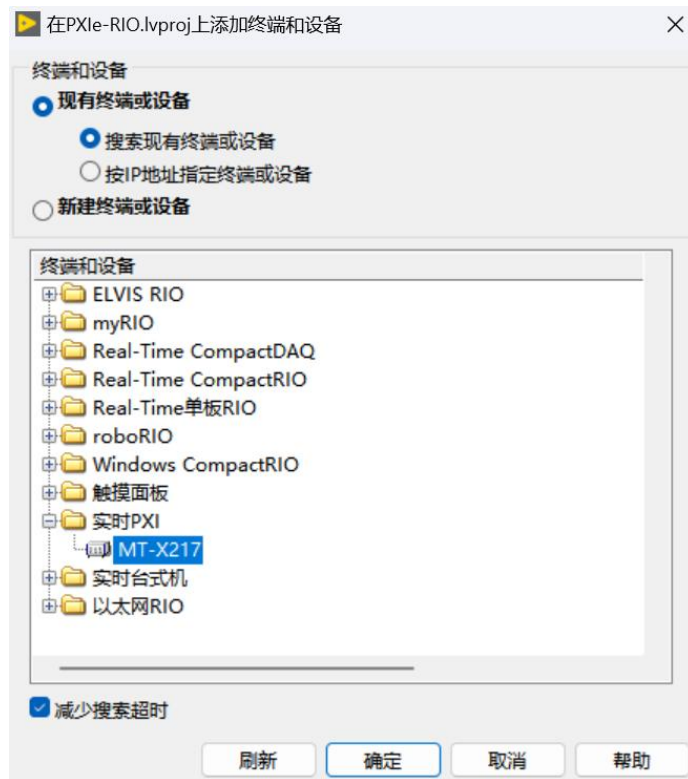


图3-5 选择X217控制器终端

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X217控制器设备IP地址。



图3-6 选择设备属性

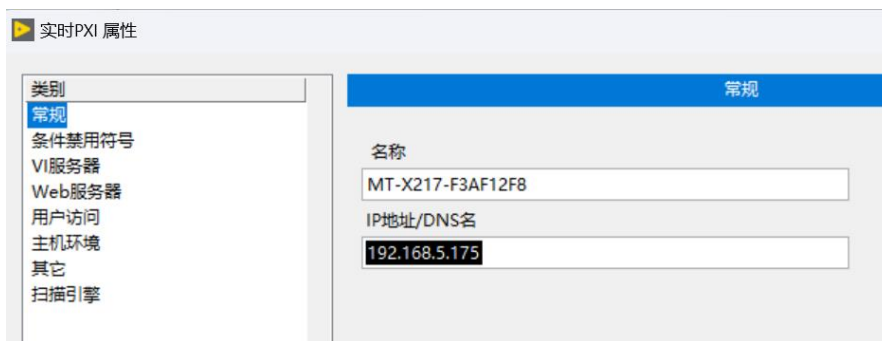


图3-7 更改IP地址

7. 连接PXIe-RIO终端设备

在项目浏览器中右击终端，选择连接，即可连接MT-X217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Linux RT的系统语言保持一致才能正常连接，修改Linux RT系统语言见第二小节。

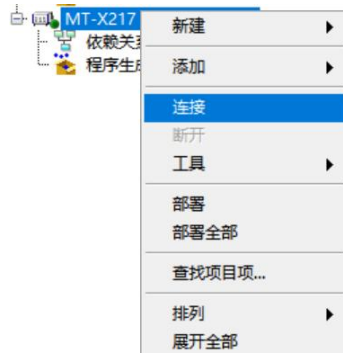


图3-8 连接PXIe控制器终端

8. 新建PXIe-RIO FPGA终端（以X922为例）

连接完成之后，为RT终端添加FPGA终端设备，右击RT终端MT-X217选择新建终端或者设备，然后选择MT-X922如下图所示。



图3-9 新建X922终端



图3-10 选择MT-X922

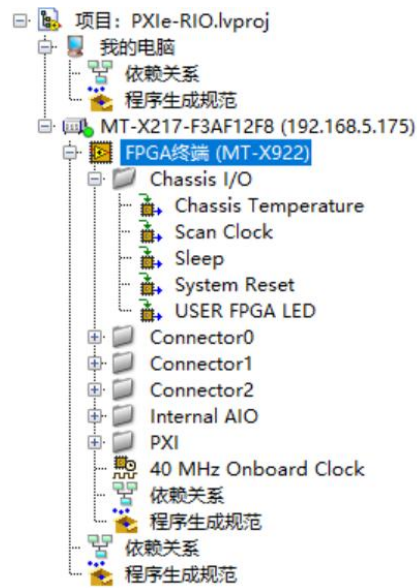


图3-11 X922 FPGA终端新建完成

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PXIe-RIO程序的开发。

9. 第一个PXIe RIO项目

完成上述准备工作之后便可以利用LabVIEW 开发环境开发第一个PXIeRIO项目，如图所示。

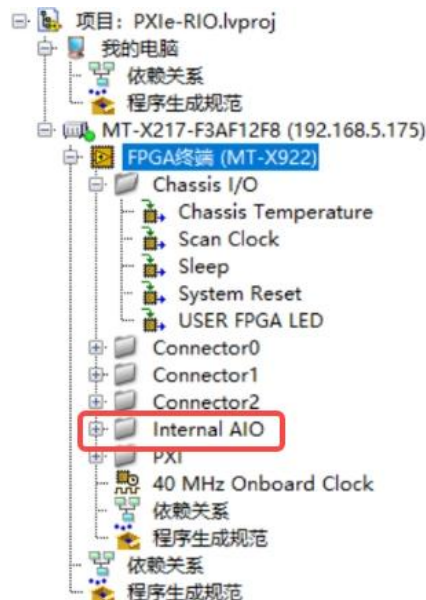


图3-12 PXIeRIO 的开发环境

注：X922有AI、A0和DIO接口，上图红框中的Internal AIO用户无法使用，AIO的功能我们提供

专门的VI，具体见下面步骤。其余的Chassis I/O、Connector0(DIO)、Connector1(DIO)、Connector2(DIO)、PXI下的I/O可以直接拖放到程序框图里使用。

新建120MHz时钟（由于X922是1M采样率，需要120M的参考时钟），如下图：

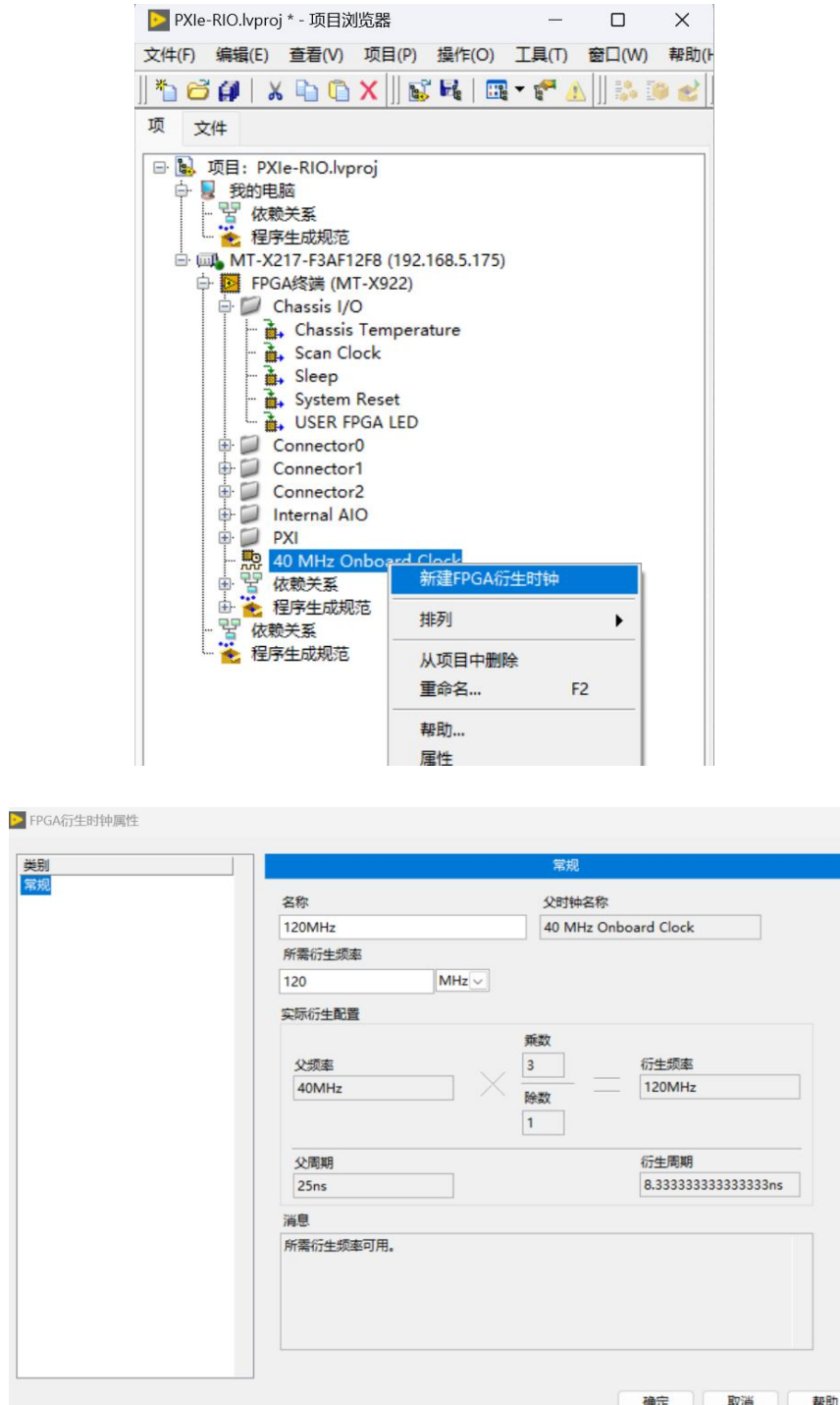


图3-13 新建FPGA衍生时钟

注：X920, X921, X922都是1M采样率，需要120M的参考时钟。

在FPGA终端下(MT-X922)新建VI，并保存在项目文件夹下：

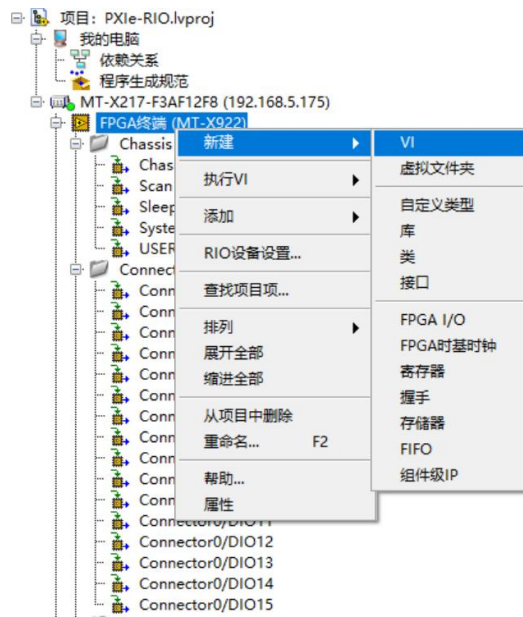


图3-14 新建FPGA VI

MT提供AI与AO的VI给用户实现模拟量的输入与输出功能，右键程序框图空白处选择MT PXIe，相应的VI如下图所示。

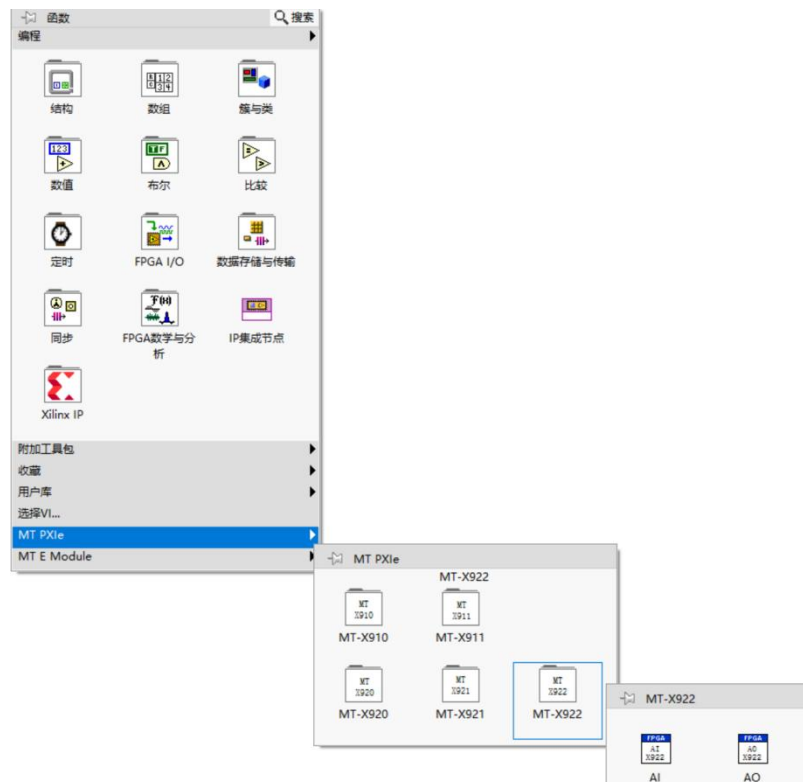


图3-15 X922 VI

然后分别放置X922的AI及AO VI 如下图所示：

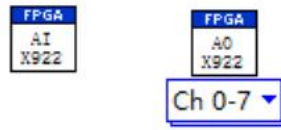


图3-16 放置 AI及AO VI

X922卡是八通道模拟量输入和二十四通道模拟量输出，所以AO VI通道可以选择，如下图所示：

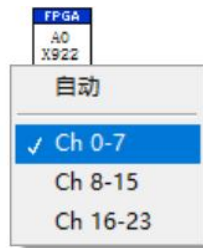


图3-17 X922 AO VI通道选择

X911和X921卡是二十四通道模拟量输入和八通道模拟量输出，所以AI VI通道可以选择，如下图所示：



图3-18 X911、X921 AI VI通道选择

在这里可以右键点击AI输出接口新建显示控件，然右键查看该采集卡FIFO的数据属性，这里可以看到X922默认输出的数据类型为SGL类型：

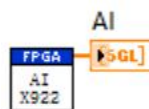


图3-19

新建一个终端至主机的 FIFO, 完成 FPGA 与 RT 之间的 DMA 通讯如图所示:

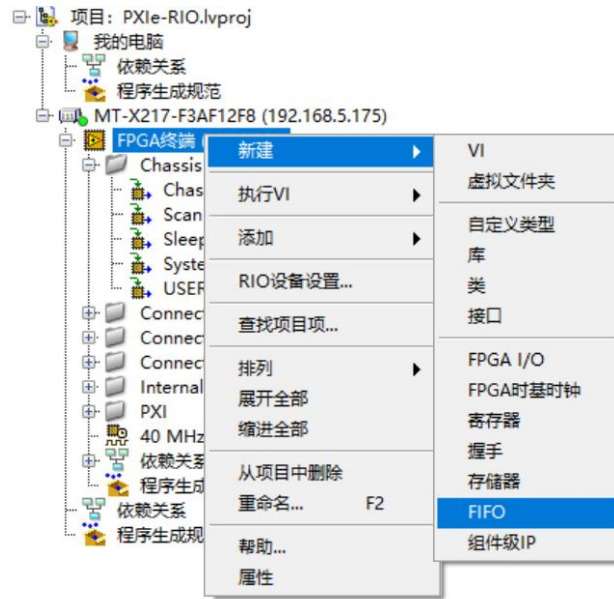


图3-20 新建FPGA FIFO

设置 FIFO 的属性, 在类型中设置成终端至主机, 同时设置上传数据类型为SGL, 用于上传AI的数据, 如图3-21和3-22所示。

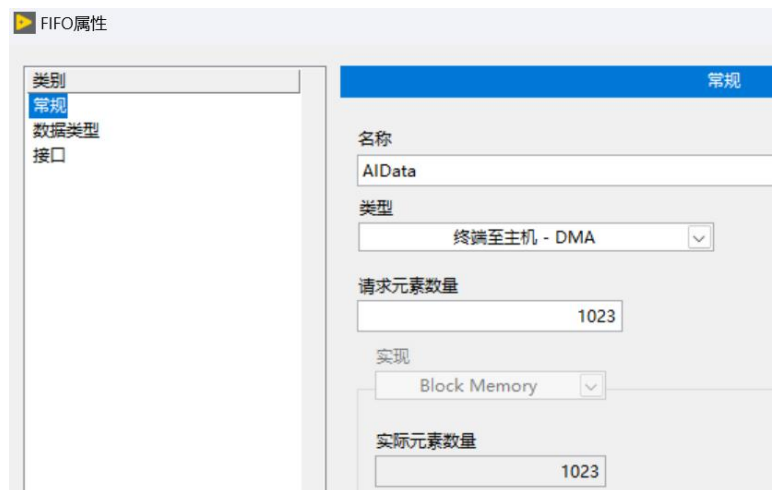


图3-21 设置类型至终端至主机

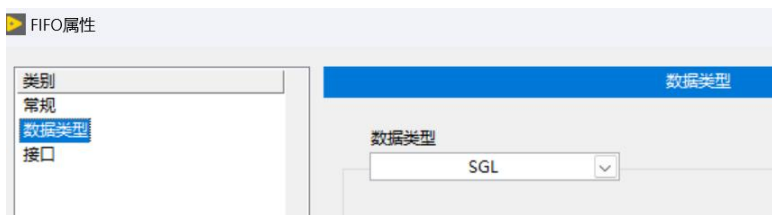


图3-22 设置上传的数据类型为SGL

FIFO创建之后，FPGA终端下将会出现，可以通过FIFO将FPGA终端中的数据上传给上位机RT，如图3-23所示：

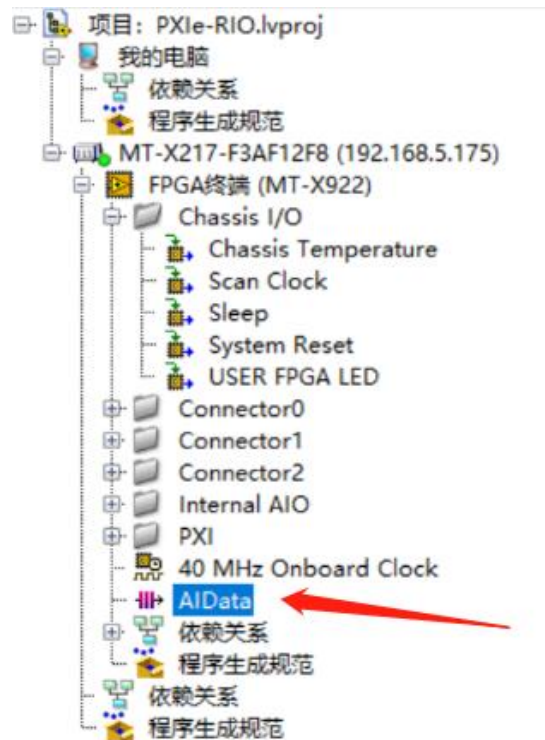


图3-23 新建的FIFO将会出现在FPGA终端下

将FPGA终端下的FIFO直接拖拽到FPGA下 VI的程序框图中如图所示：

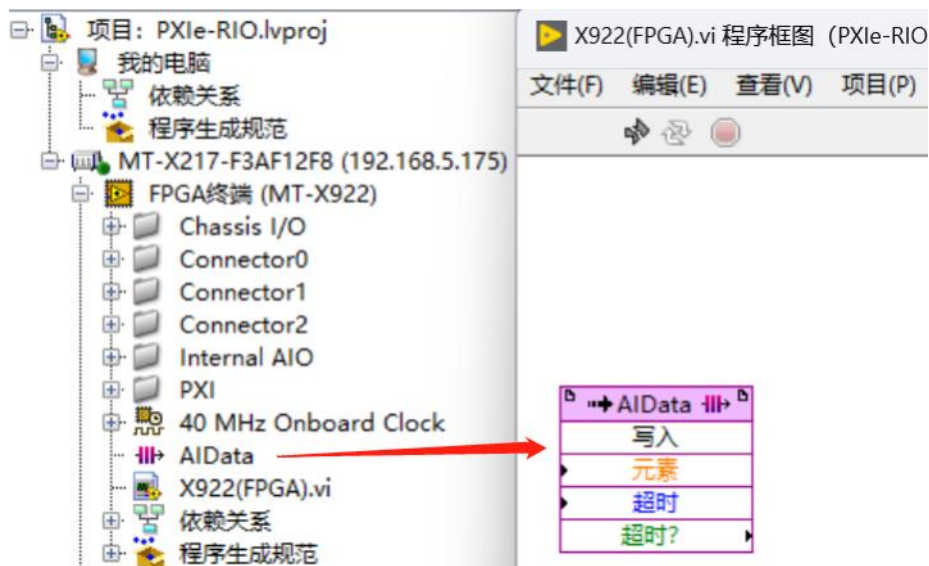


图3-24 将FIFO直接拖拽到程序框图中

利用FIFO将AI的数据通过DMA上传，给AO创建输入控件用于设置输出电压，如下图所示：

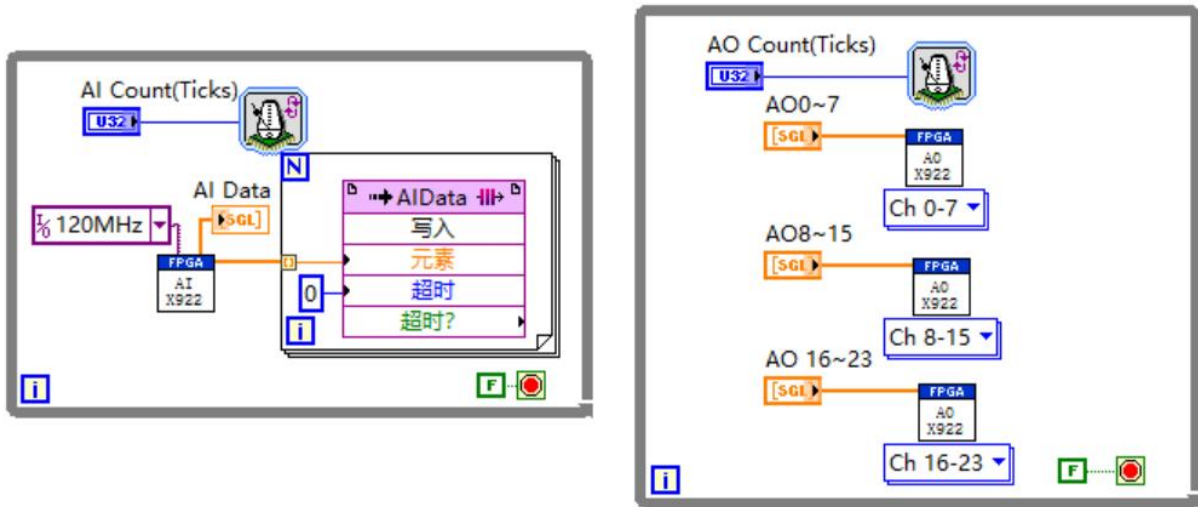


图3-25

采样率=4000000/Count (Ticks)，X922的最大采样率是1M/S/ch，即当Count设置小于等于40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)，

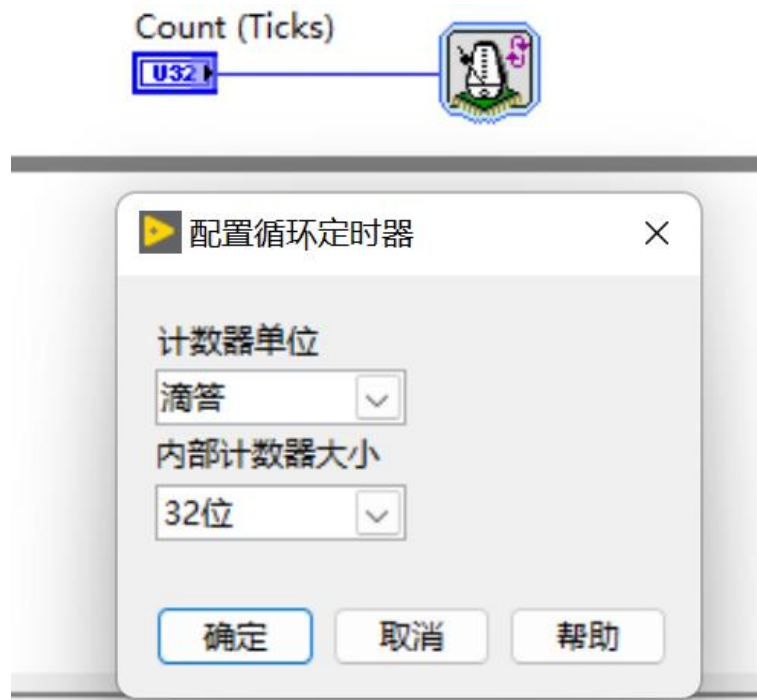


图3-26

将X922的Connector0 (D10)、Connector1 (D10)、Connector2 (D10) 下的I/O直接拖放到程序框图里使用，如下图所示：

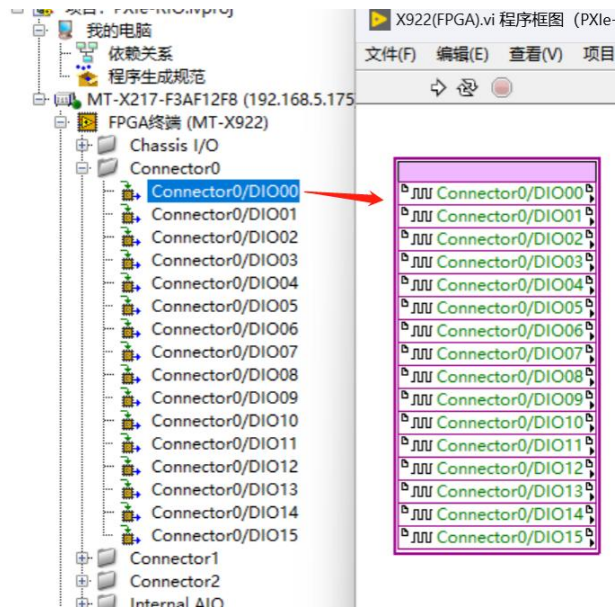


图3-27 将DIO直接拖拽到程序框图中

通过一个布尔值循环取反的方式控制DIO输出方波，如图所示：

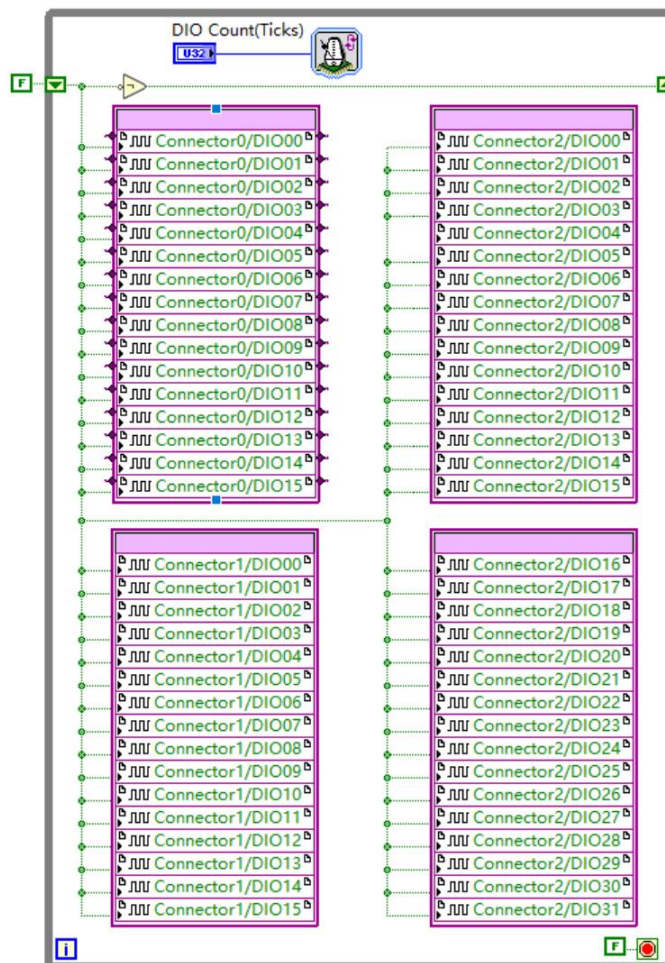


图3-28 DIO发方波程序

写FPGA程序，X922 FPGA VI程序框图总览：

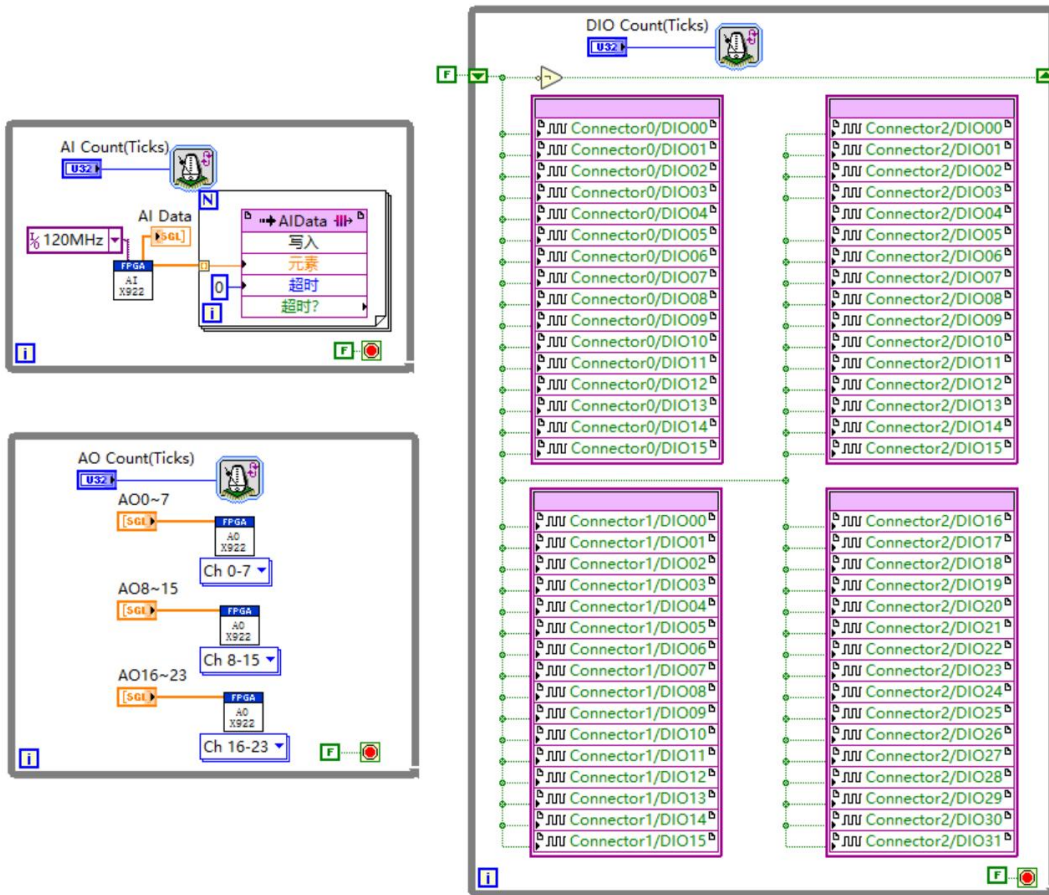


图3-29 完整FPGA Demo程序

FPGA 程序写完之后需要进行编译，点击左上角运行按钮，出现提示框，点击确定按钮，仅生成比特文件，如图所示：

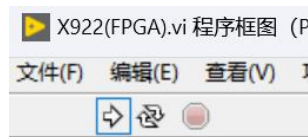


图3-30 点击运行进行编译

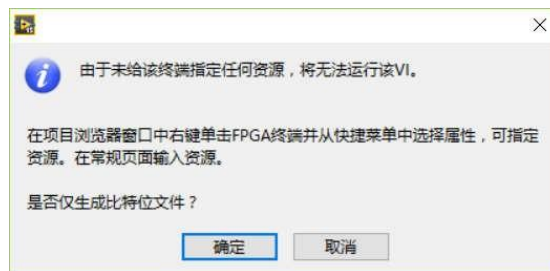


图3-31 点击确定按钮，仅生成比特文件

编译的时候，如果本地计算机安装了编译器可以选择本地编译，如果同个网络中其他计算机安装了编译器，本地没有安装编译器，可以选择连接至网络编译服务器，并输入相应计算机的IP地址如图所示。

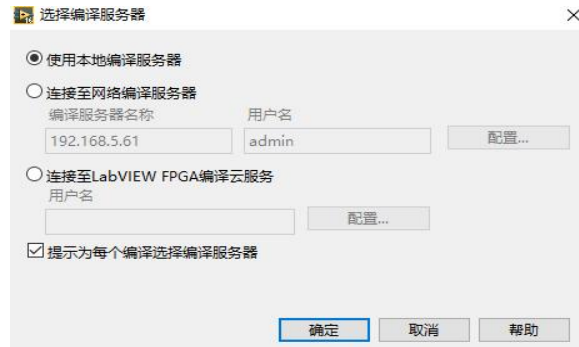


图3-32 选择编译服务器

最终的编译状态界面如图所示。

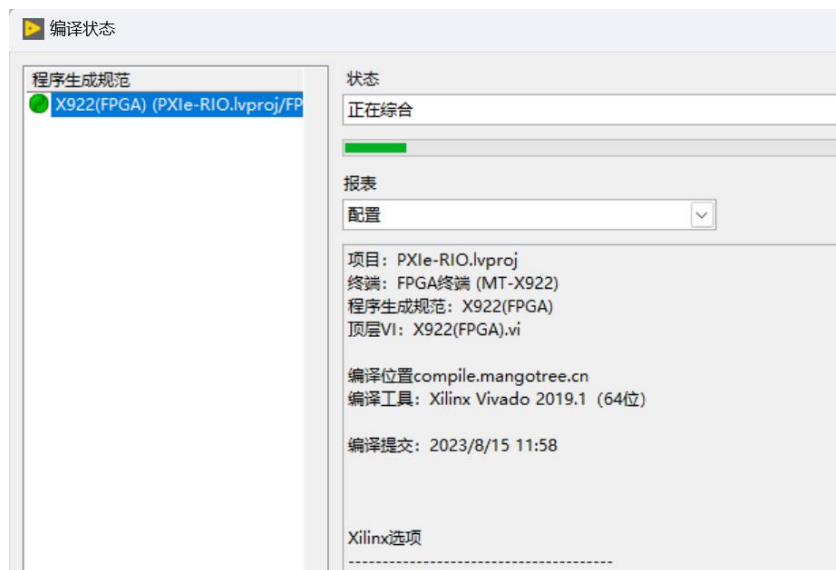


图3-33 编译窗口

编译完成之后，在刚刚保存的项目文件夹下，会出现 FPGA Bitfiles 文件夹，里面存有刚刚编译生成的文件夹如图所示。

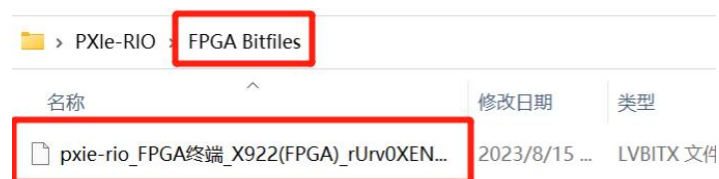


图3-34 生成的FPGA Bitfile

至此，FPGA程序开发已经完成，下面介绍上位机程序的开发。

右击RT终端新建上位机VI，并保存，开始上位机VI程序编写：

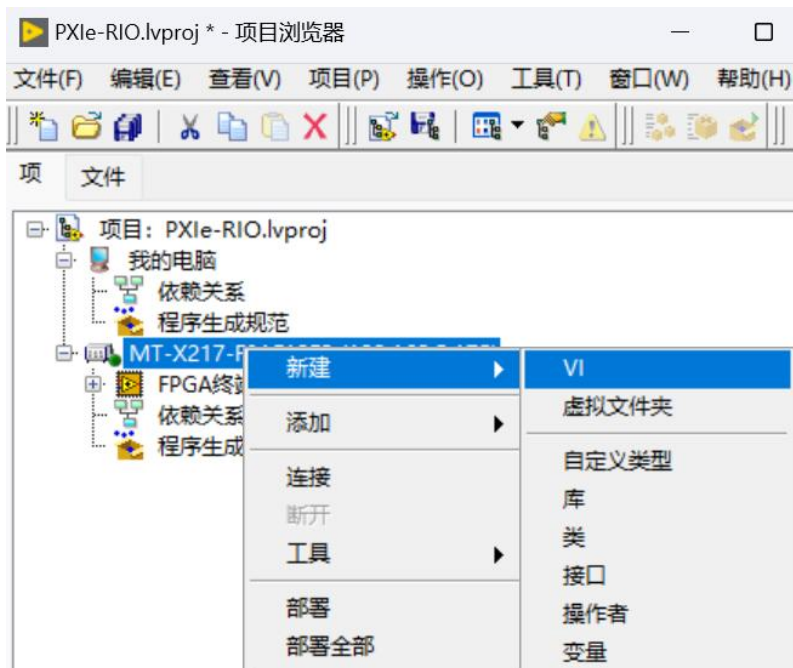


图3-35 右击RT终端，选择新建 VI

新建和保存好之后可以看到网格状的前面板框图：

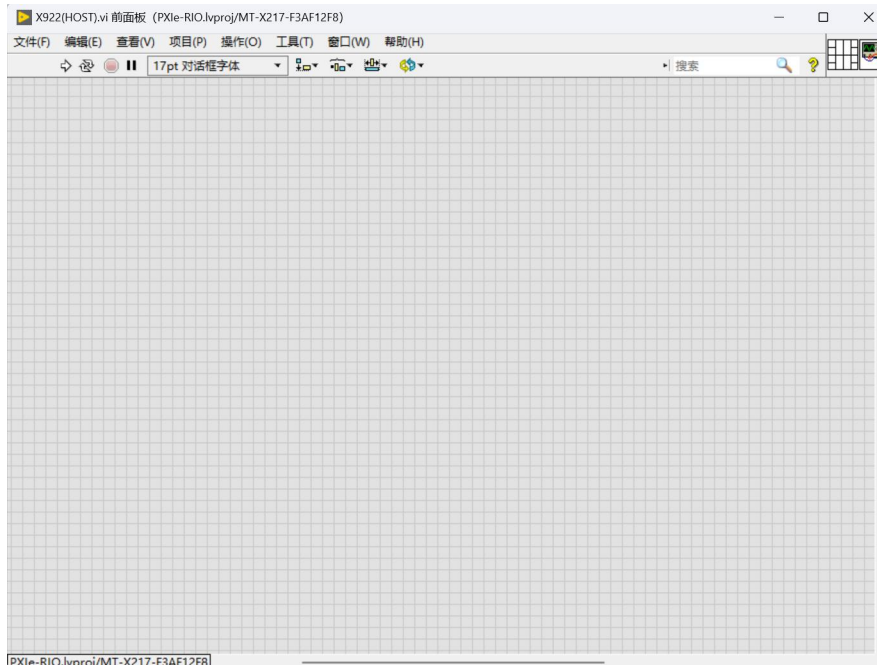


图3-36 前面板

Ctrl+E快捷键，可以在前面板和程序框图间互相切换：

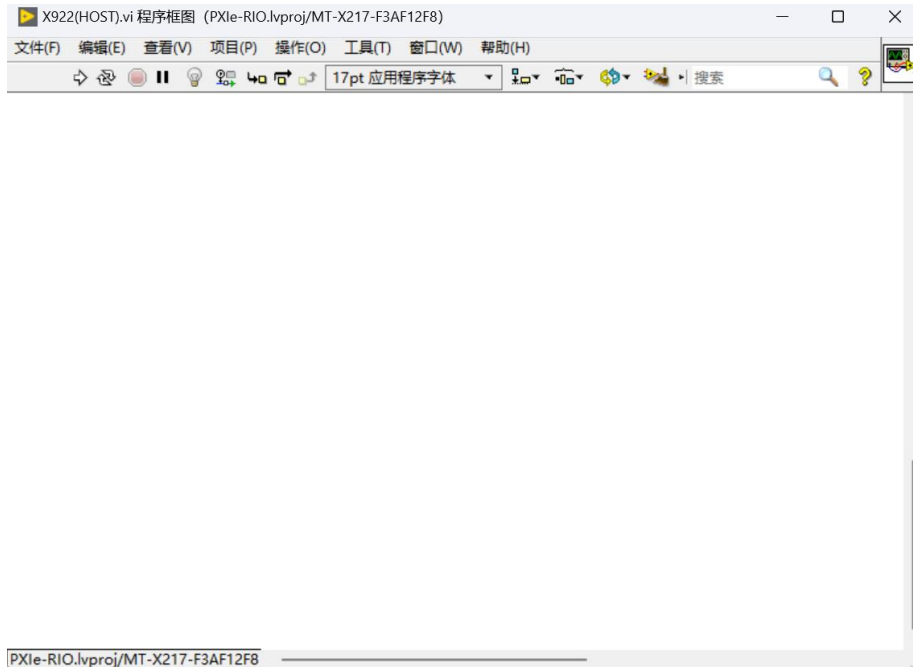


图3-37 程序框图

在程序框图空白处右击，通过下图所示MT-RIO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：

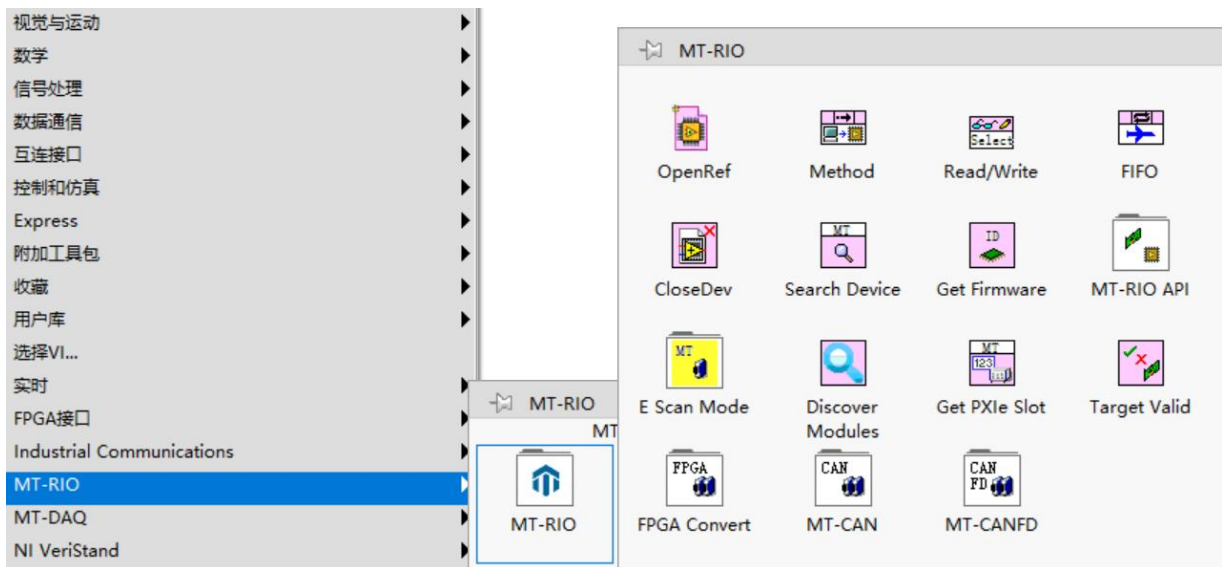


图3-38 MT-RIO选板

上位机VI写法A：（基础）

首先在程序框图中放置一个OpenRef，然后双击，在弹出的对话框中，选择刚刚 FPGA 编译生成的比特文件如图所示。

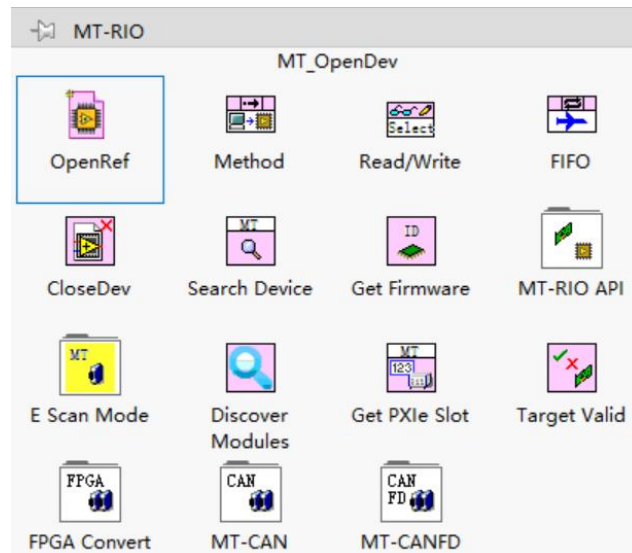


图3-39 放置OpenRef函数

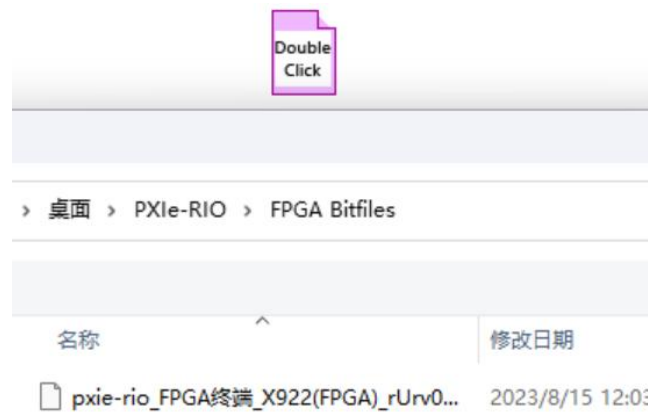


图3-40 选择FPGA比特文件

将Method放在程序框图中，并且和OpenRef连接，然后左击Method，选择Reset如图所示。

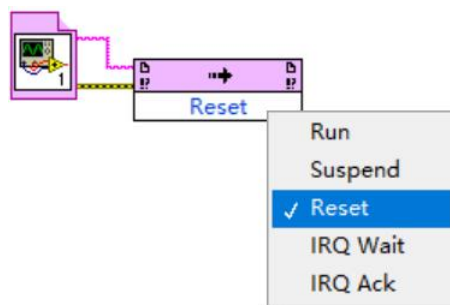


图3-41 左击Method，选择Reset

将FIFO放置到程序框图中，然后将FIFO连接到Method，左击FIFO，选择Configure，同时给开辟的深度一个较大的值，如图所示。

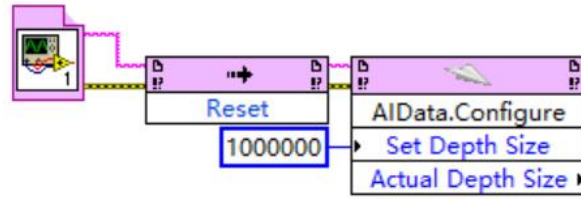


图3-42 配置FIFO通道

然后再添加Method和FIFO节点，Method左击选择Run，FIFO选择start如图所示。

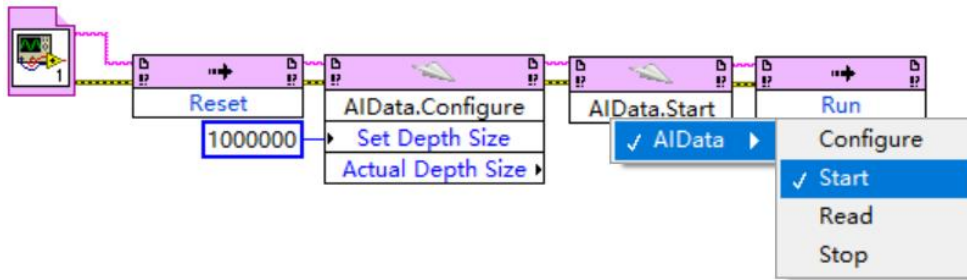


图3-43 在后面连接Run和Start

在程序框图中放置一个While循环，然后将Read/Write 放置在循环中，将Read/Write 与前面的节点相连接如图所示。

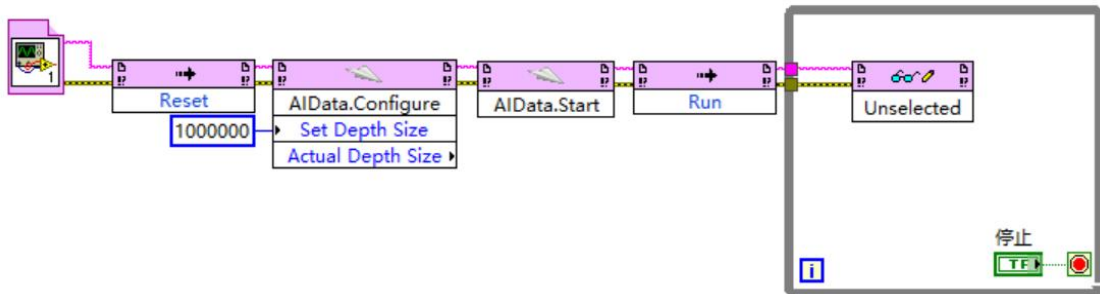


图3-44 将Read/Write放置在while循环内

左击Read/Write，可以显示FPGA VI 中的输入控件和显示控件如图所示。

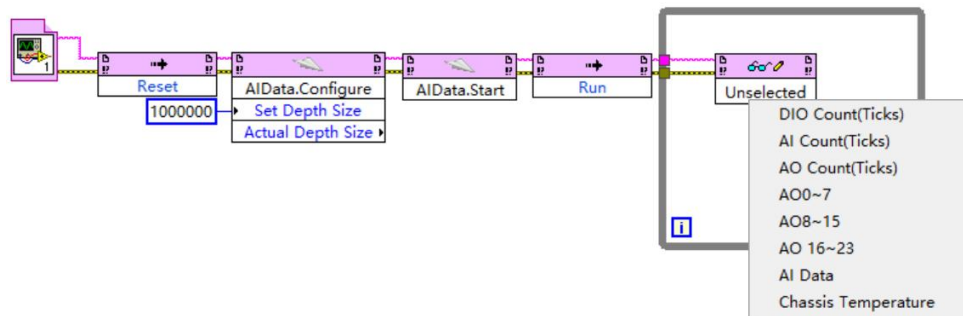


图3-45 左击Read/Write

可以将鼠标移动到Read/Write 的上边缘或者下边缘，当出现双向箭头的时候进行拖拽，自动出现其他参数如图所示：

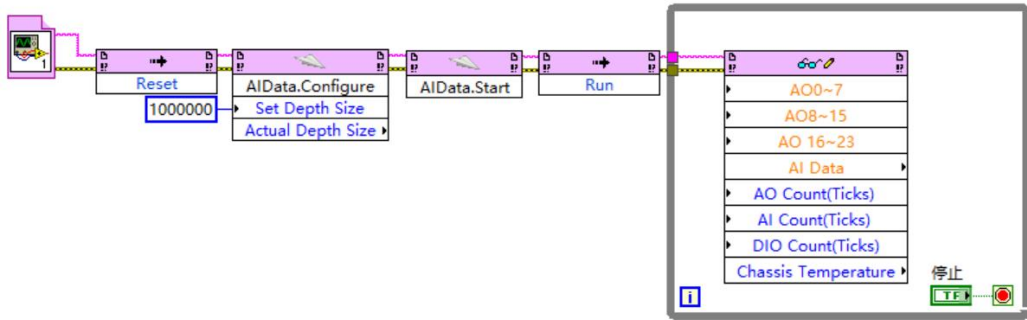


图3-46 进行控件的拖拽

通过在Chassic Temperature处进行除法运算，得到一个板载温度，给AI Count、AO Count、A00~7、A08~15、A016~23及AI Data创建输入输出控件，如图所示：

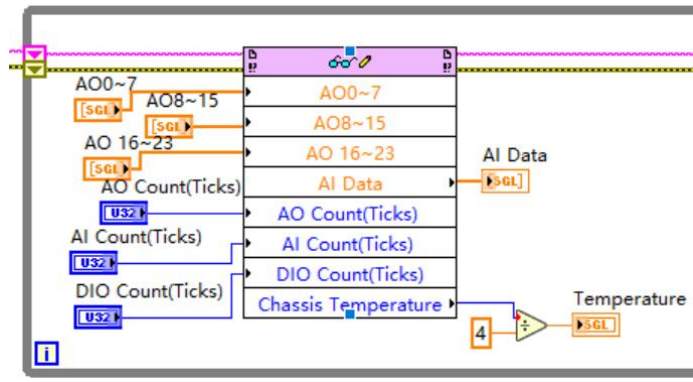


图3-47 与 FPGA 端控件交互

放置一个FIFO，左击选择Read，然后给Number of Elements赋0，读出其中Element Remaining 的值（DMA 缓存中可读数据个数）连接给另一个FIFO Read，然后将读出数据用波形图表实时显示，如图所示。

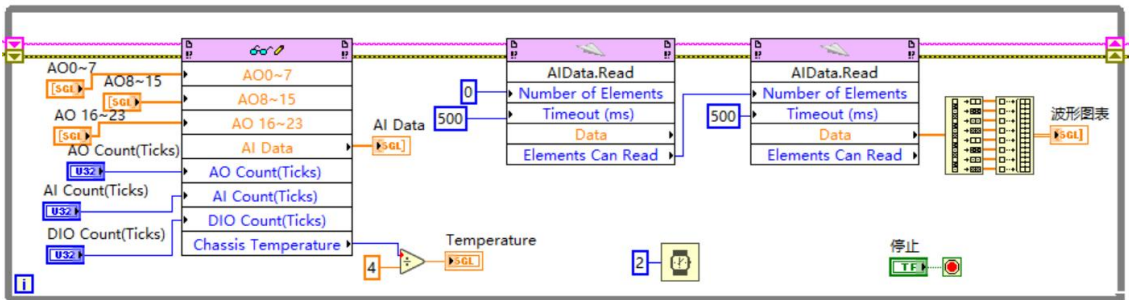


图3-48 显示 FIFO 传输上来的数据

最后放置一个FIFO，选择Stop，再放置一个CloseDev，完整Host程序如下图所示。

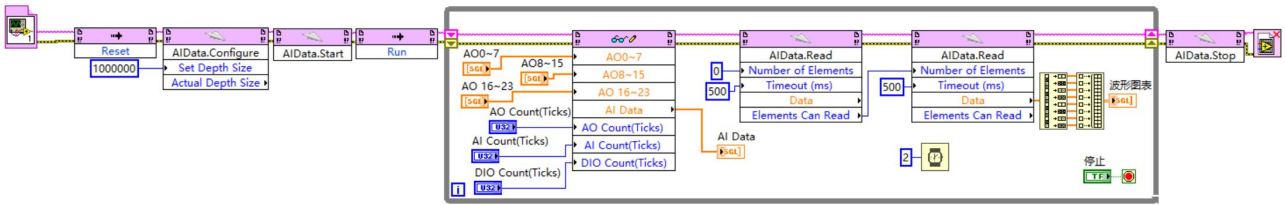


图3-49 完整Host程序

最后点击左上角运行按钮，Host程序即可运行。

注意：上述Host程序是运行在RT终端本地的，用户也可以将Host程序拖到我的电脑目录下，通过填写RT终端的IP地址来实现远程连接运行，具体操作如下：

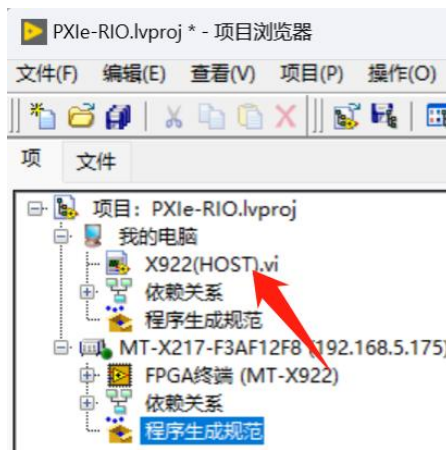


图3-50 将Host程序拖放到我的电脑下

在Host程序框图中添加 Search Device函数，选择Remote

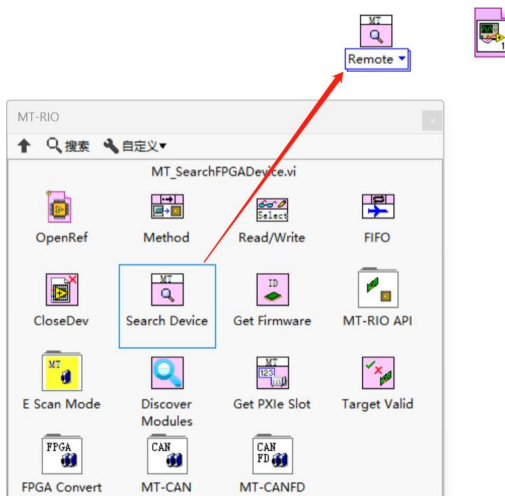


图3-51 添加Search Device

填写RT终端的IP地址，并与OpenRef函数连接

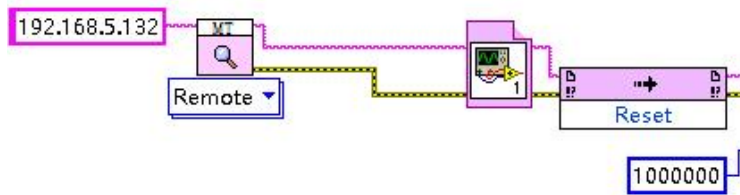


图3-52 填写IP地址

至此，基础的上位机Host程序开发介绍完毕，下面介绍另外一种上位机程序的写法，利用MT RIO API编程。如下图所示为相关API函数。

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：

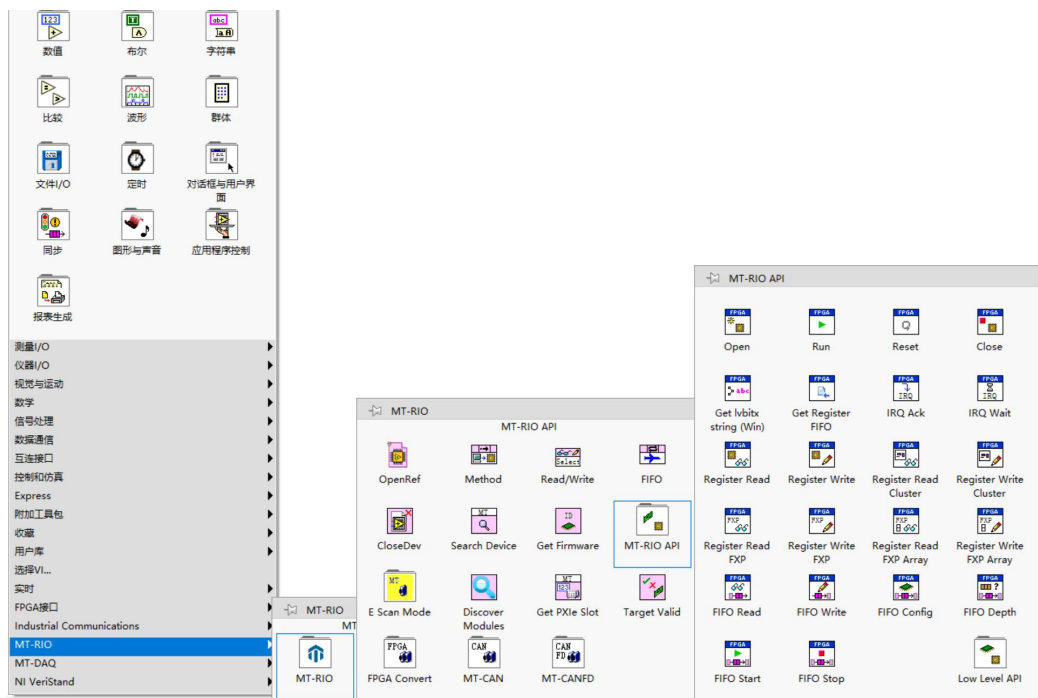


图3-53 MT-RIO API

MT RIO API编程程序总览:

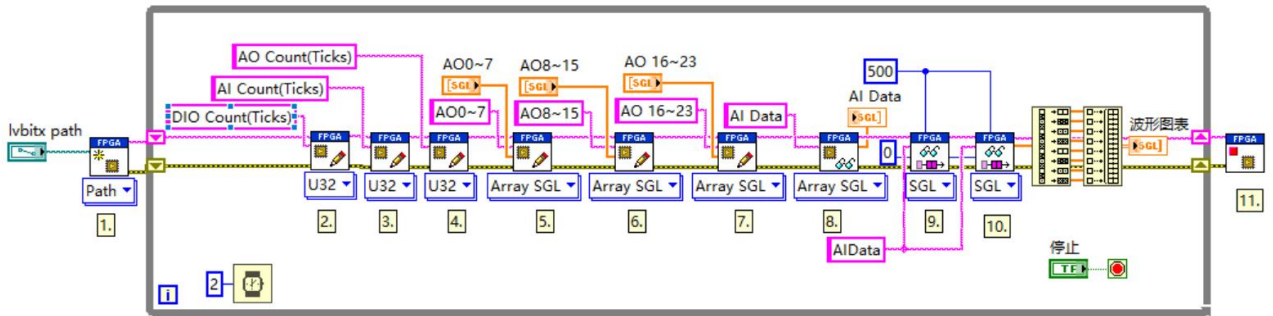


图3-54

程序说明:

1. Open函数: 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

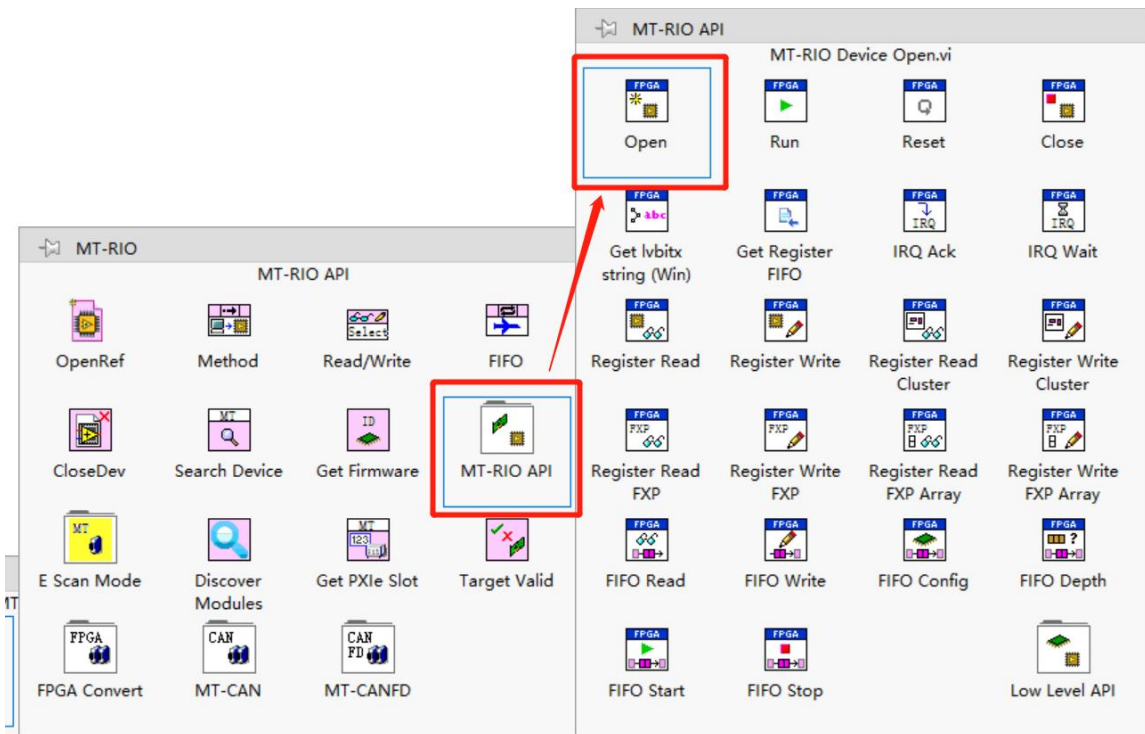


图3-55 Open函数

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3-56

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3-57

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：

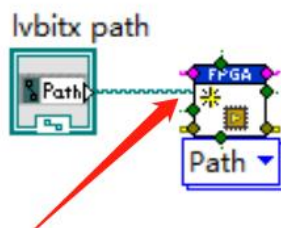


图3-58

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3-59 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：

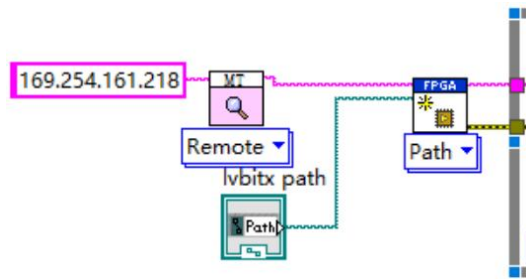


图3-60

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3-61 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:

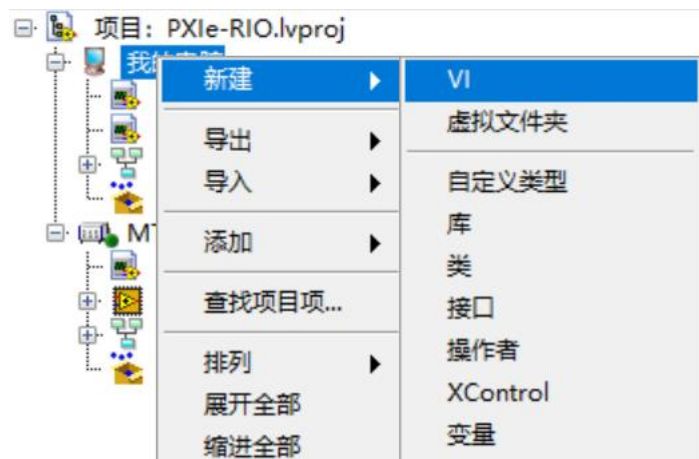


图3-62

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：

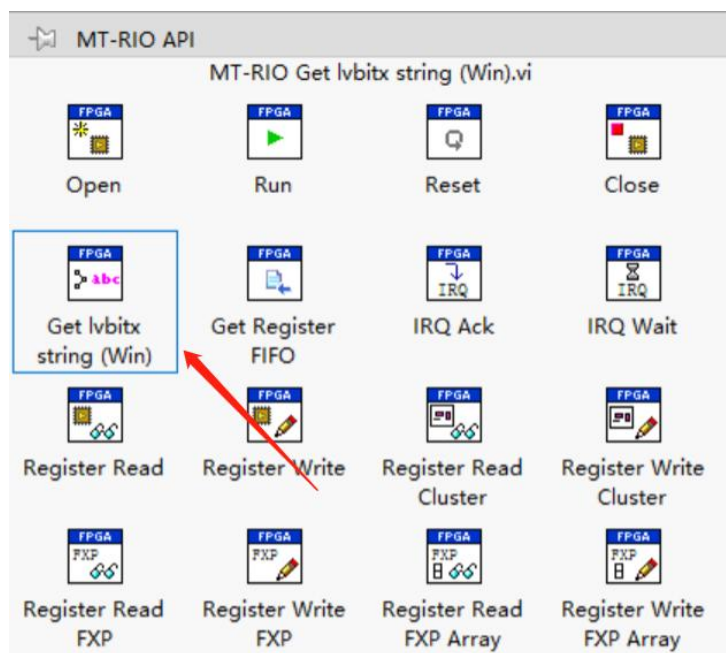
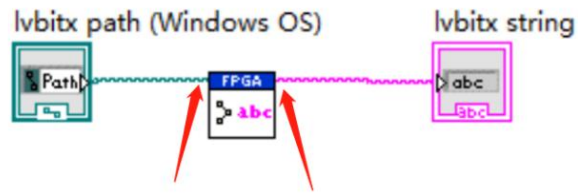


图3-63



右键创建输入和输出控件

图3-64



图3-65

点击左上角运行按钮，生成字符串：

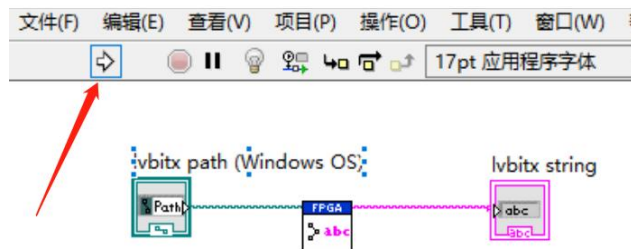


图3-66

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：

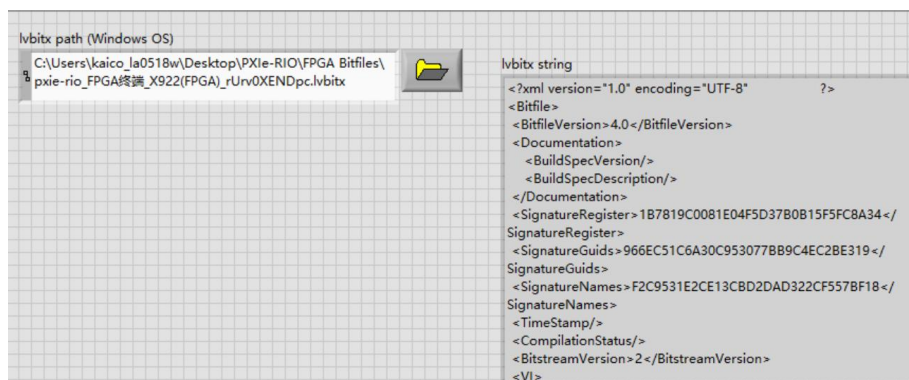


图3-67

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：

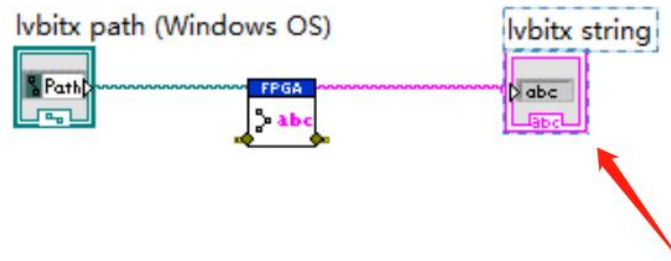


图3-68

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

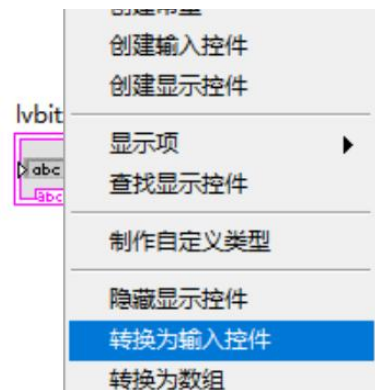


图3-69

并将当前值设置为默认值：

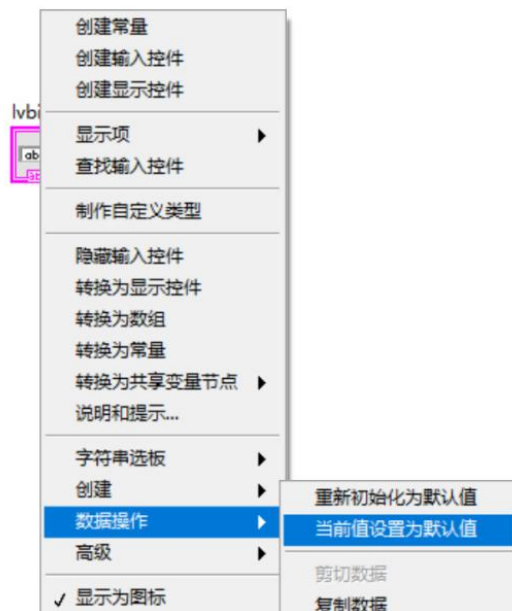


图3-70

然后将String传递给Open函数的lvbitx string接口，并将Open函数的方法改为String。

如下图：

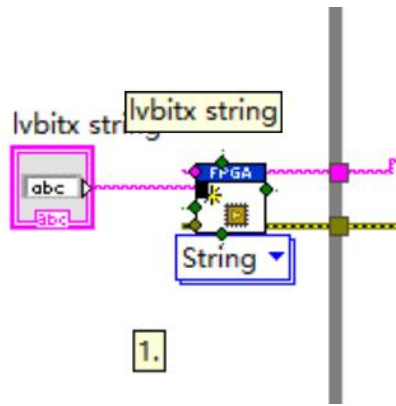


图3-71

2. Host.vi 写在**我的电脑**目录下:

那么除了做到上面 1 中的几步, 还需要按照下图方式, 在Open函数前面添加Search Device, 然后填写RT终端的IP地址给Open函数:

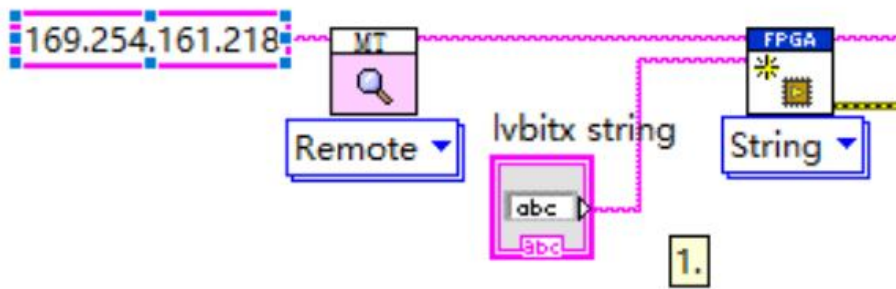


图3-72

继续介绍程序总览, 如下图所示:

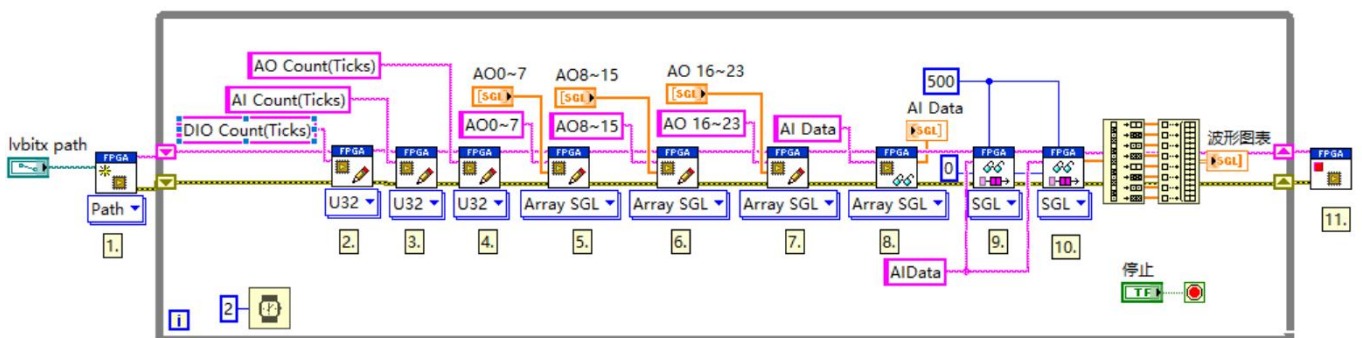


图3-73

2-8: 图中编号2-8为MT R10 API选板下寄存器的读写函数, 同于读写FPGA里的输入输出控件, 函数需要填写寄存器名称, 这里填写的名称要与FPGA程序里定义的一致, 如下图所示:

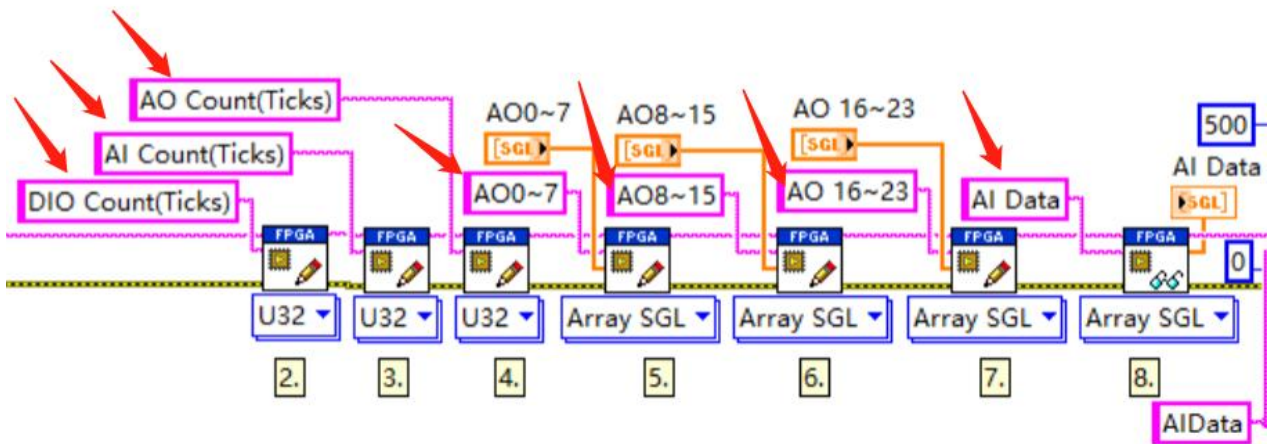


图3-74

读写寄存器函数还需选择对应的数据类型，与FPGA程序中的保持一致，通过下拉其多态VI选择器来手动选择，如下图所示：

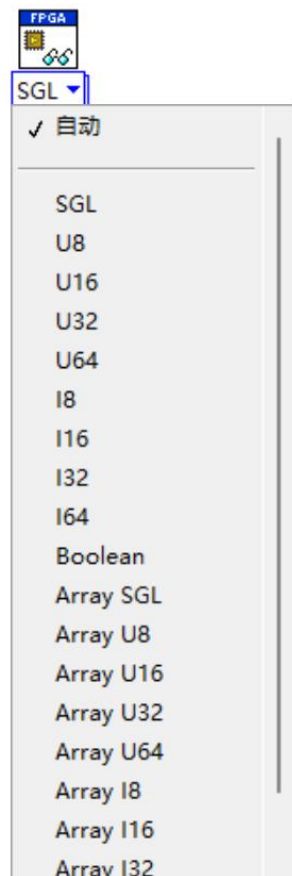


图3-75

9、10：程序中9和10的位置放置了两个FIFO Read函数，用于读写DMA通道的数据，同样

要填写FPGA 中FIFO的名称和DMA所传的数据类型，如下图所示：

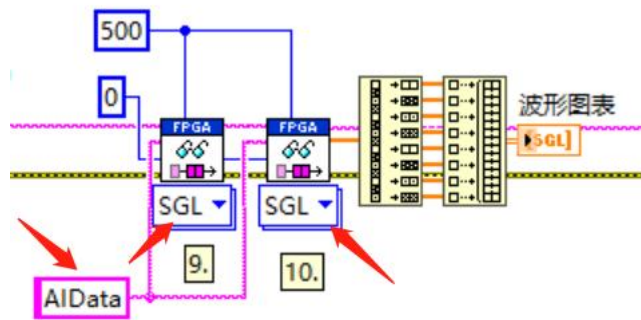


图3-76

11: MT RIO API选板下Close函数。

10. 部署PXIe RIO实时应用程序

在PXIeRIO中生成实时应用程序，当PXIe机箱(搭配PXIe控制器)上电时，将会自动运行生成的实时应用程序。首先在实时PXI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。

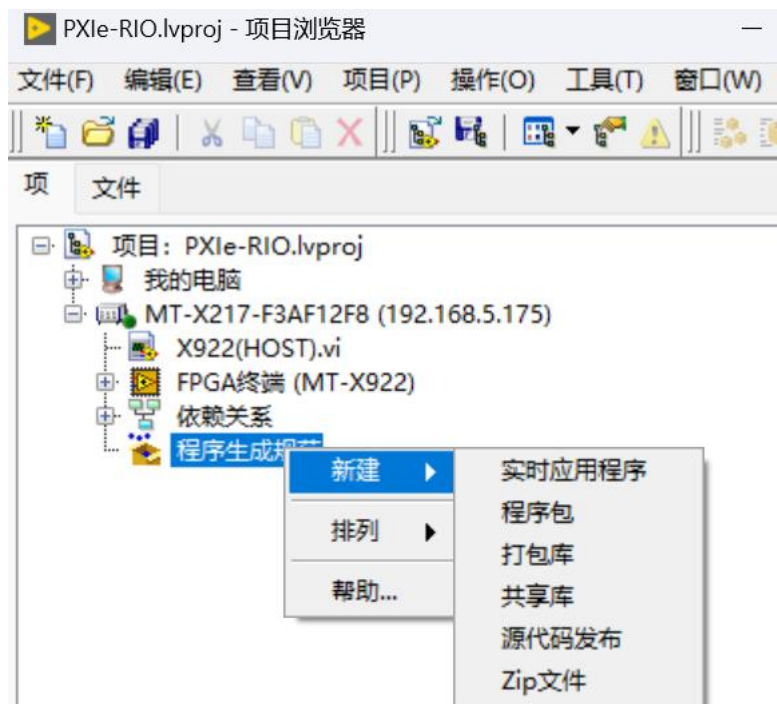


图3-77 新建实时应用程序

在弹出的对话框中，在源文件选项中，选择要生成程序的源文件，把上位机Host VI添加到右侧启动VI，然后点击生成，如下图所示：

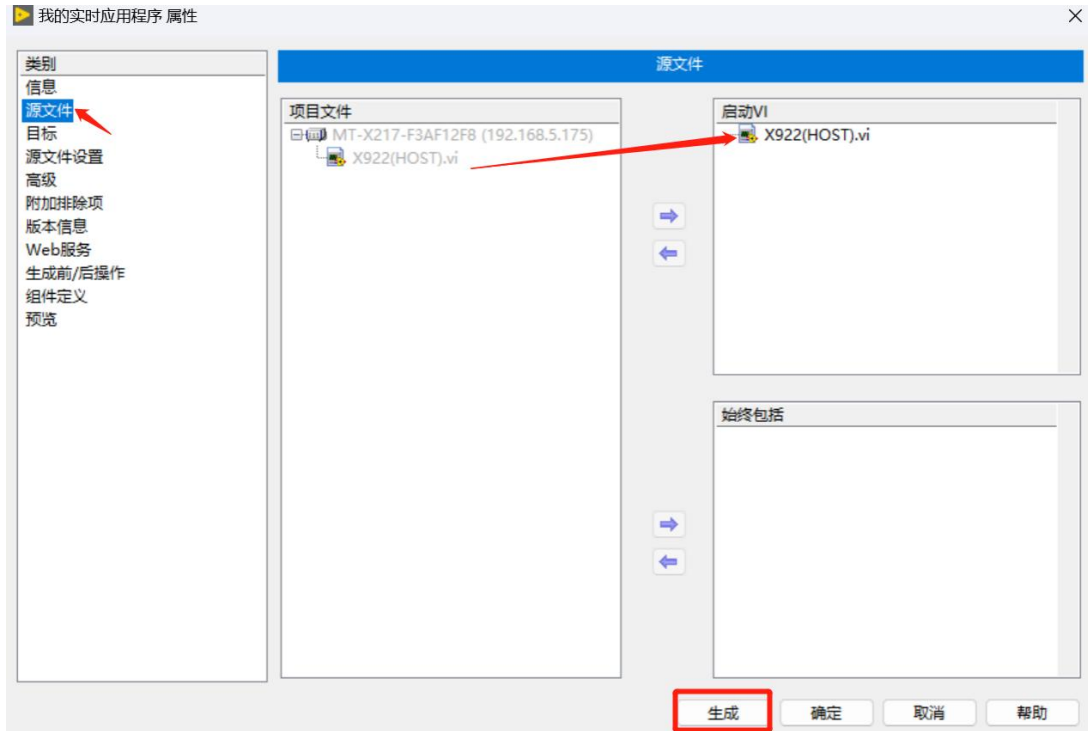


图3-78 生成实时应用程序

生成完成之后，在程序生成规范中，右击刚刚生成的实时应用程序，选择作为启动项运行，如下图所示。

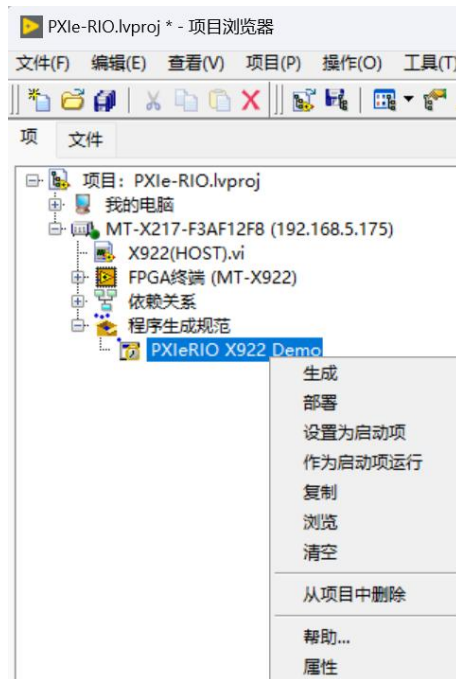


图3-79 作为启动项运行

四、开发第一个PXIeRIO项目（Windows系统）

PXIeRIO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PXIeRIO使用开发流程。

PXIe机箱搭配装有Windows版的PXIe控制器，用户开机接显示器后就可以在PXIeRIO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、PXIeRIO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FPGA终端MT-X922，如图4-1所示。

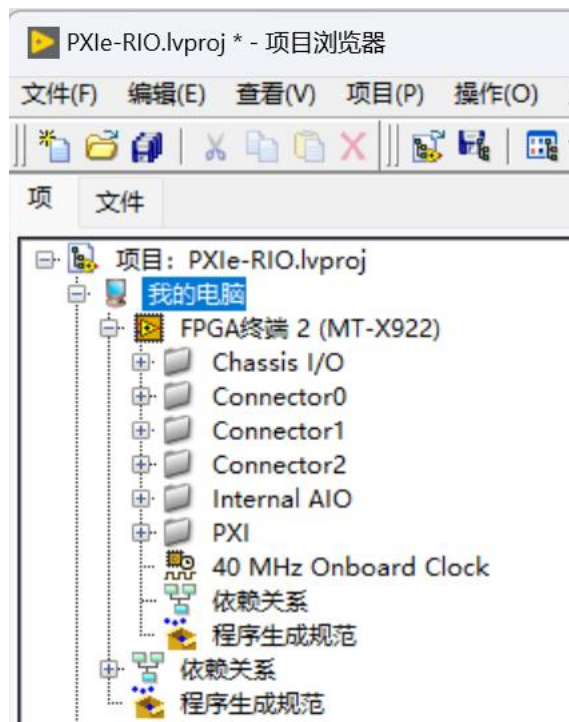


图4-1 在我的电脑下新建X922 FPGA终端

接下来开发FPGA程序及上位机程序，参见第三章的开发流程。