Modbus、S7 协议建立 PLC 通信

目录

一．安装环境

基于 Win10 家庭版系统 关闭防火墙
Visual Studio 2022（编写 C#脚本）

Modbus 通信测试工具:
Modbus Poll（推荐，需要破解）
HslCommunication（可选）

Modbus 通信协议库：
NModbus4 (开源，MIT 协议)

博途系列软件（均需要破解）:
TIA Portal V15（用于写入 PLC 程序）
S7-PLCSIM Advanced V3.0（模拟 PLC）
软件安装包:
链接：https://pan.baidu.com/s/1XX78RS8Kut01BXrW52wO3Q?pwd=916t
提取码：916t

TIA Portal V15 软件安装教程:
https://www.bilibili.com/video/BV1t34y1a75V/?spm_id_from=333.337.search-card.all.click&vd_source=83d2130f4c5f6c8184cc12a6c6b98a79

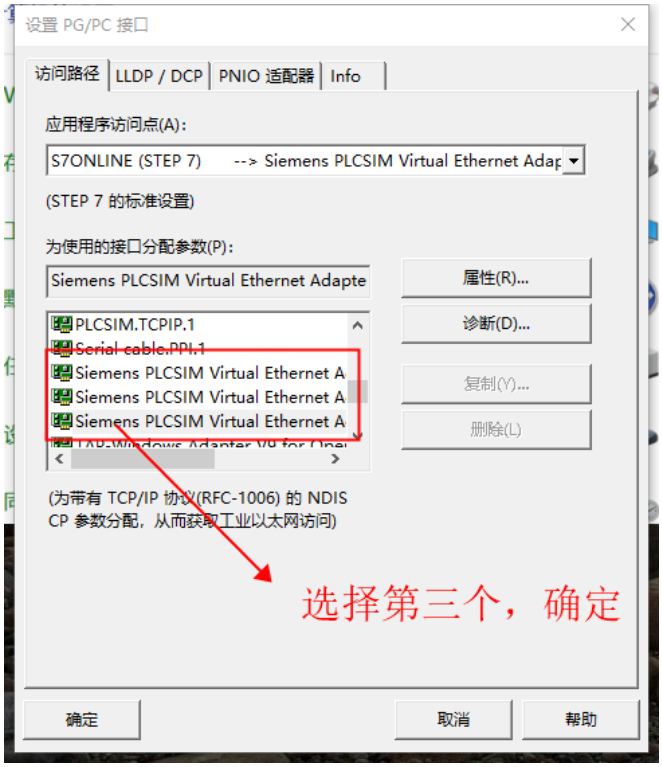
S7-PLCSIM Advanced V3.0 软件安装教程:
https://www.bilibili.com/video/BV1av4y1K7Uc/?spm_id_from=333.999.0.0

Modbus Poll 安装破解
https://blog.csdn.net/byxdaz/article/details/77979114

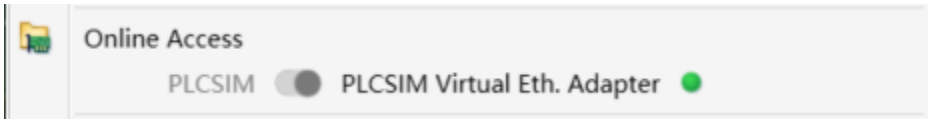
二．生成模拟 PLC

1．安装好 S7-PLCSIM Advanced V3.0 后可以试运行虚拟 PLC。打开控制面板界面的设置 PG/PC 接口

设置应用访问点为 S7ONLINE(STEP 7) →Simens PLCSIM 开头的第三个

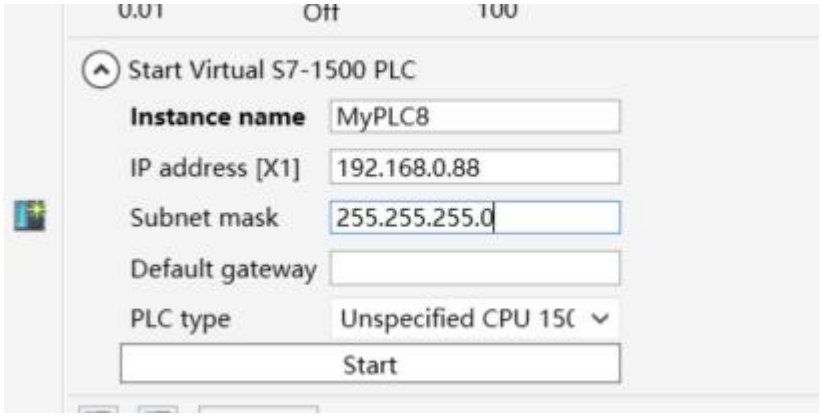


打开 S7-PLCSIM Advanced V3.0
切换为模拟 PLCSIM Virtual Eth. Adapter

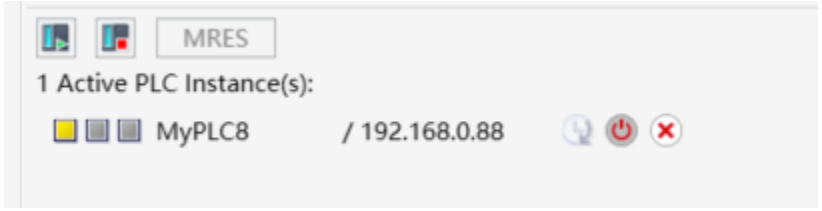


网络连接设置为以太网

目前只能模拟 S7-1500PLC
设置 plc 名称、ip 地址、网关后点击 start 开始模拟



下方出现 plc 代表运行成功

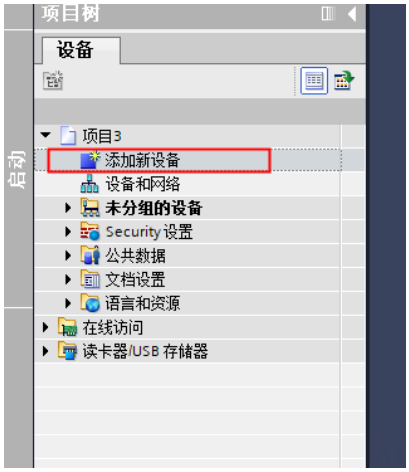


三．建立 PLC 项目

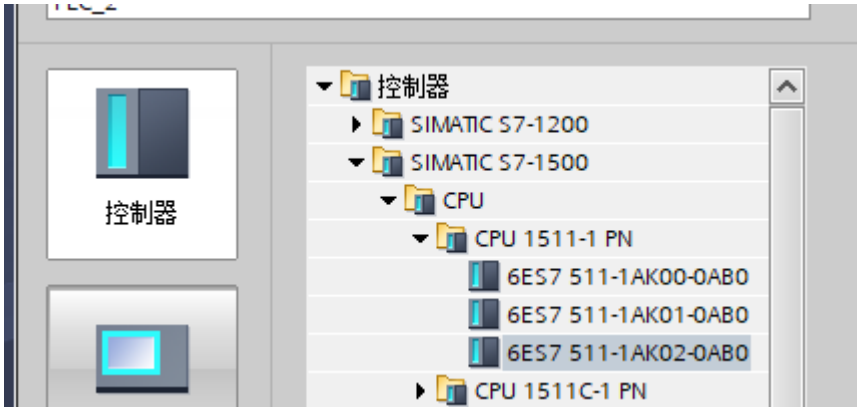
打开 TIA V15 创建新项目、

打开项目视图

双击左侧添加新设备



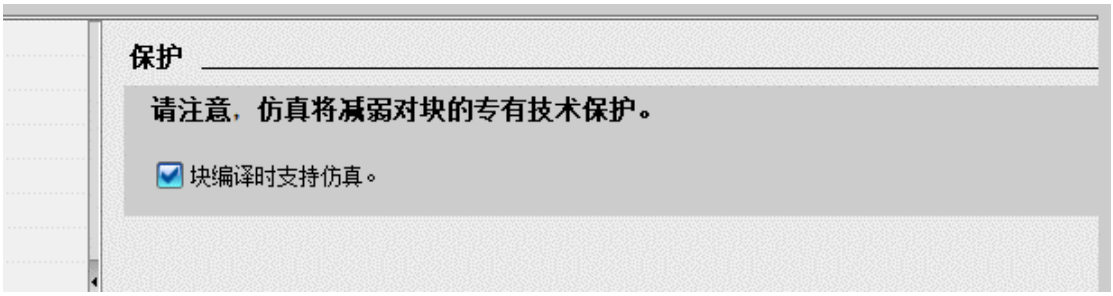
选择 S7-1500PLC 控制器（与之前模拟的 PLC 相同型号）选择下图所示 CPU



此外需要右键左边栏的 

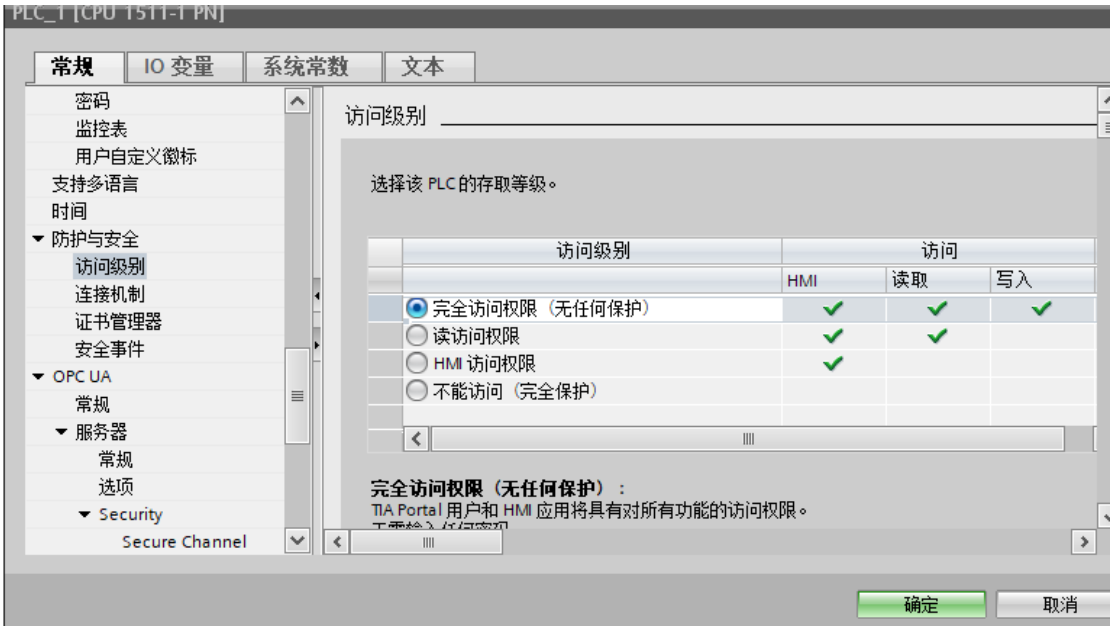
点击属性

将保护中的块编译支持仿真勾选上，用于支持模拟 PLC 环境

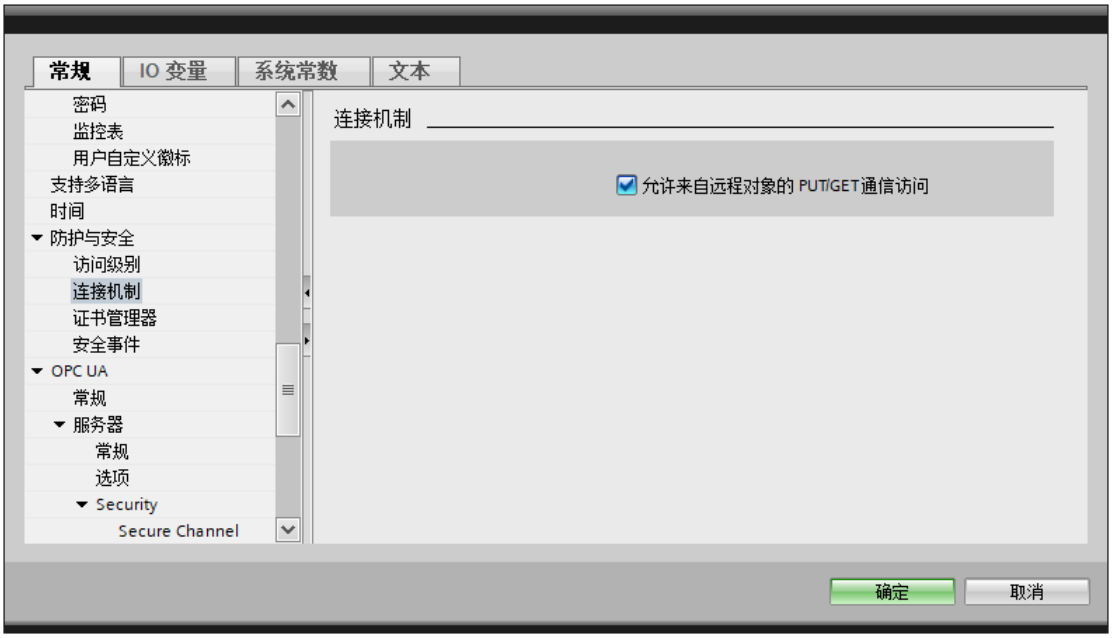


右键左侧栏 plc 设备，点击属性，将以太网地址更改为之前模拟 PLC 相同的 IP 地址

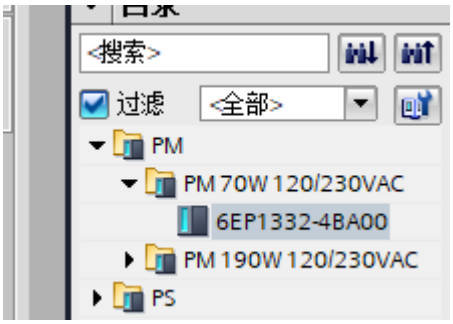
将防护与安全下的访问级别设置为完全访问权限



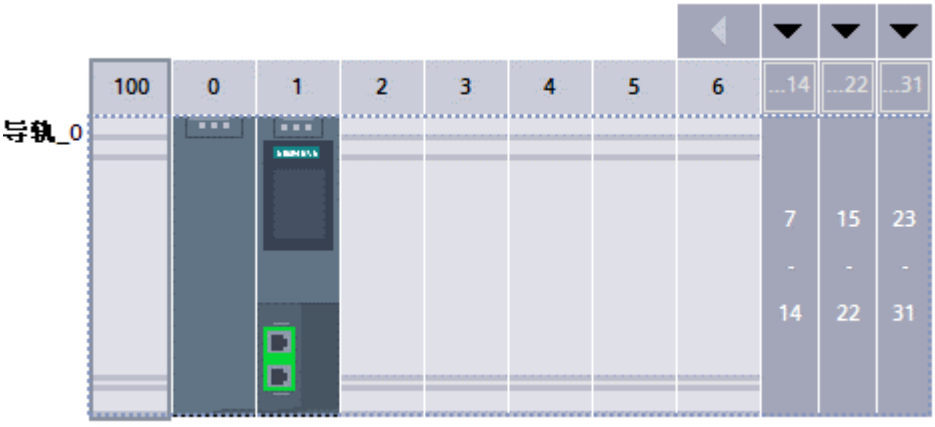
连接机制中勾选允许远程 PUT/GET 通信访问



设备视图中点击右侧 PM 选择电源



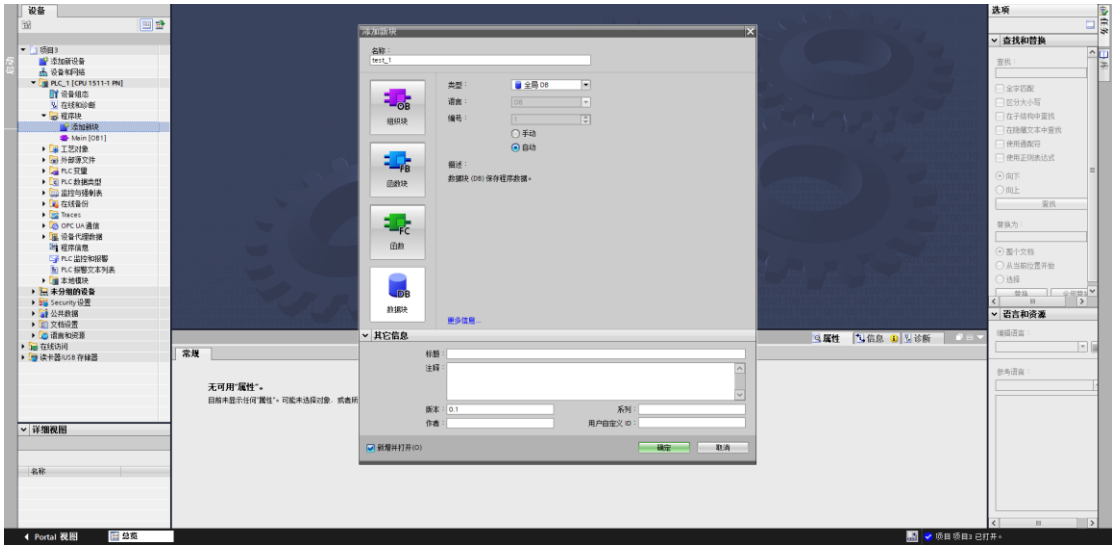
将其拖动至 plc 左侧



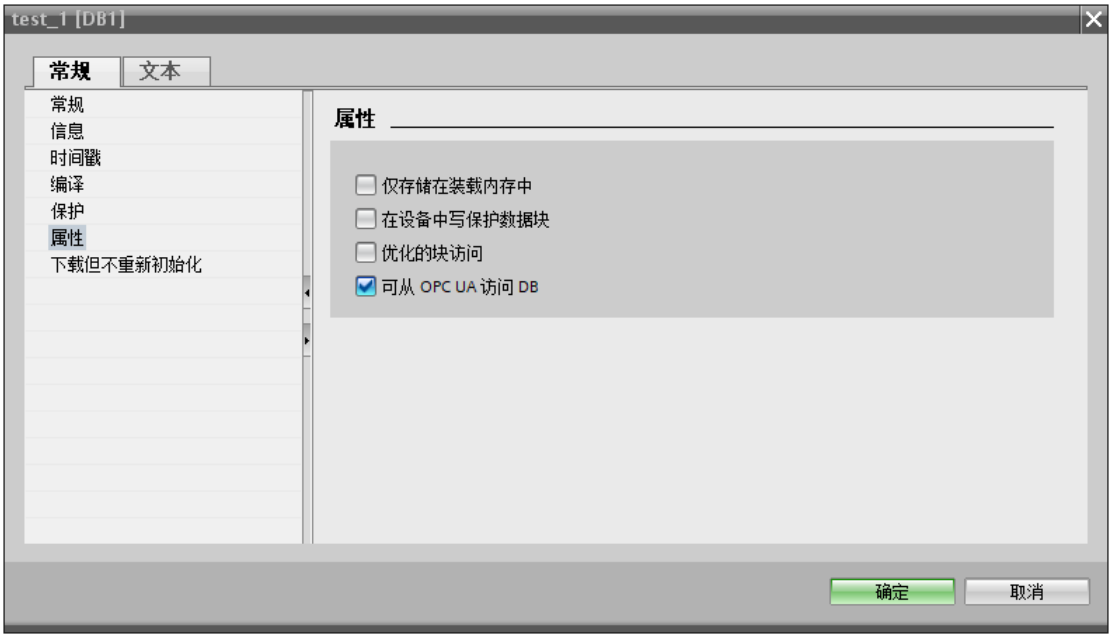
四．使用 S7 协议与 PLC 建立连接

S7 通信协议是西门子 S7 系列 PLC 内部集成的一种通信协议,。C#可以通过调用 S7 协议库中的 api 对 PLC 直接进行连接以及数据读写，对于西门子 S7 系列 PLC 侧不需要太多额外的配置。

点击左边栏 plc 设备下的程序块，点击添加新块，建立一个全局数据块 test_1



右键左边栏新建的 test_1 数据库，点击属性，将优化的块访问取消勾选
因为我们需要用 C#通过绝对地址访问数据，开启自动优化则无法访问绝对地址
在这之后创建的每一个数据块或函数块都需要进行这一步操作

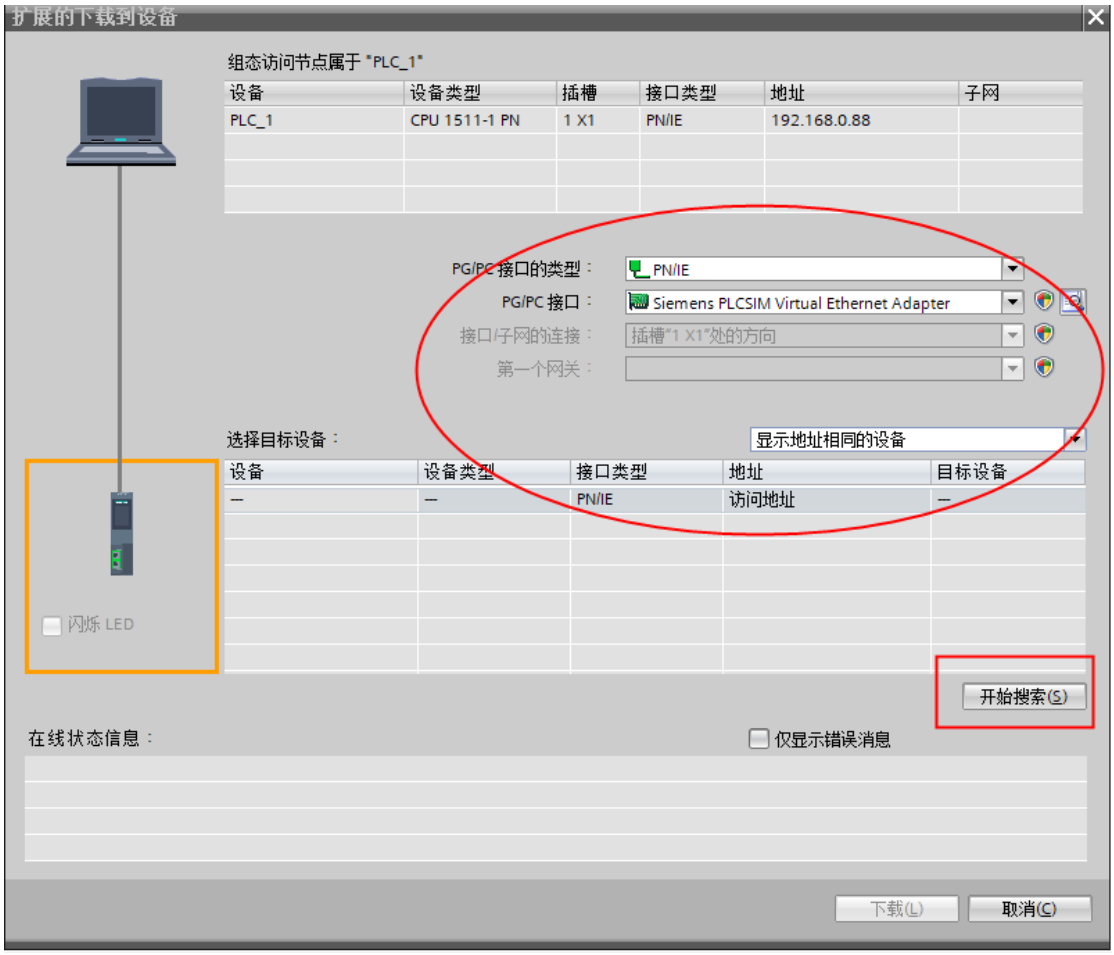


双击 test_1 数据块，创建一个布尔型变量 boo1，用于后续的更改

创建好数据块之后，需要将编写的内容下载安装至我们之前模拟的 PLC 中，点击上边栏的下载按钮

PG/PC 接口类型以及 PG/PC 接口选择如下，选择目标设备为显示地址相同的设备

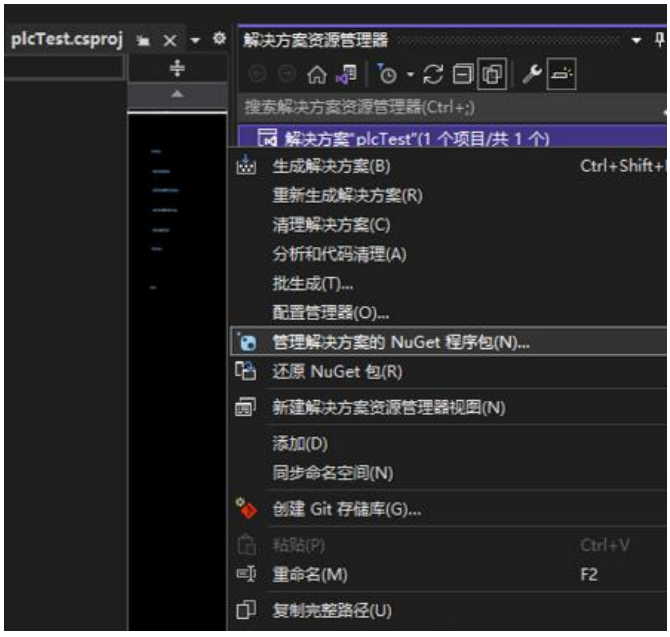
点击开始搜索



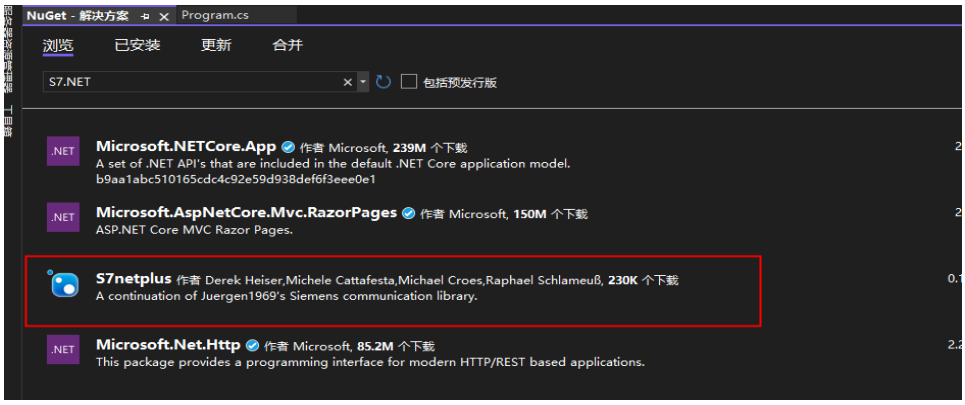
搜索到模拟 plc 后，选中点击下载即可，之后一路点击完成

打开 VS 2022 选择语言为 C#，创建控制台应用,框架选择.NET5.0 或者.NET6.0 均可

创建好后右键点击右边栏的解决方案，点击管理解决方案的程序包



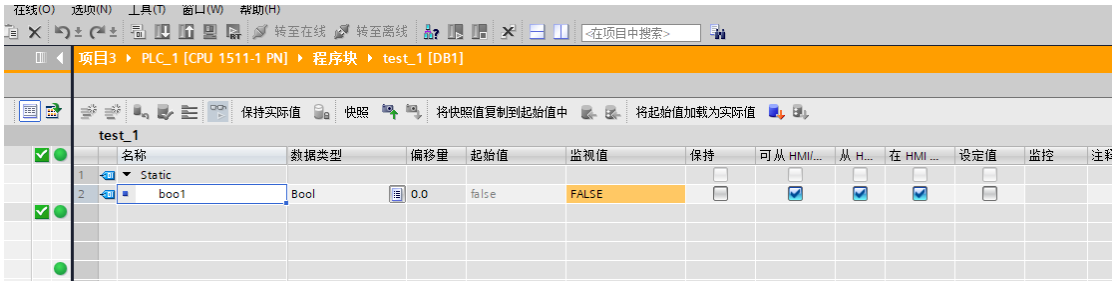
搜索 S7.NET,安装下图所示库



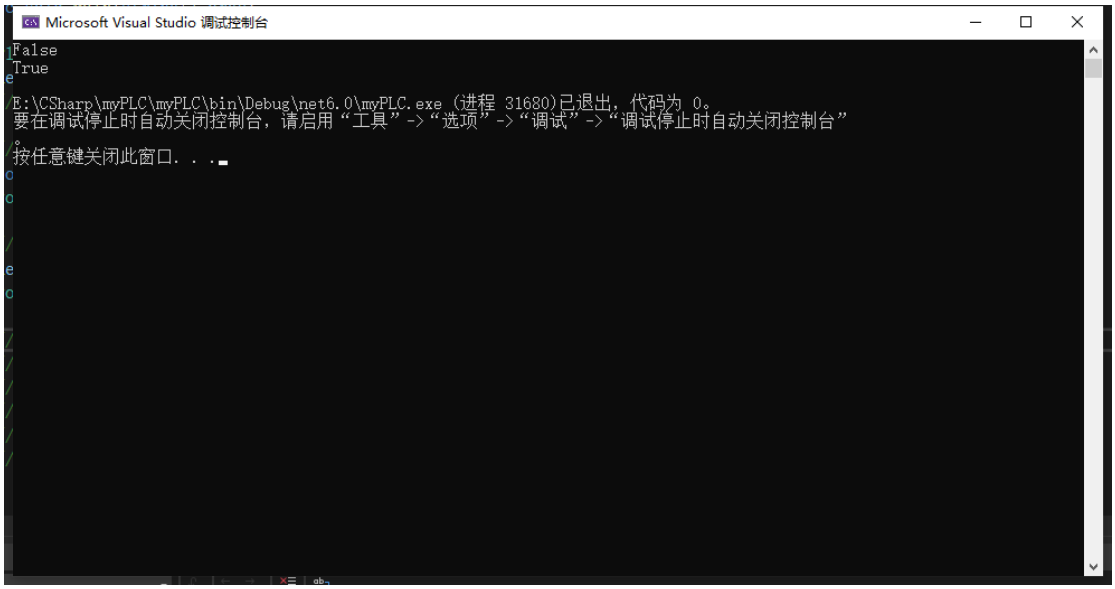
下载好后，代码中引入 S7 命名空间，并创建一个 plc 对象，其中 CpuType.S71500 代表 PLC 的型号，192.168.0.88 为之前模拟 PLC 的地址，0 为 u 机架号，1 为操作号
调用 api .Read 读取数据，其中 DB1 代表创建的数据块编号，偏移量（地址）为 0.0，因此之前创建的布尔值地址为 DB1.DBX0.0。最后用 console 在控制台中打印结果。

```
using S7.Net; // 引入命名空间
namespace myPLC
{
    0 个引用
    class Program
    {
        0 个引用
        static void Main(string[] args)
        {
            Plc newPlc = new Plc(CpuType.S71500, "192.168.0.88", 0, 1); // 创建一个PLC对象
            newPlc.Open(); // 打开连接plc
            // newPlc.Close(); // 断开连接

            // 读取
            bool B = (bool)newPlc.Read("DB1.DBX0.0");//强制转换为布尔类型赋值给B
            Console.WriteLine(B);

            // 写入
            newPlc.Write("DB1.DBX0.0", true);
            Console.WriteLine(newPlc.Read("DB1.DBX0.0"));
```

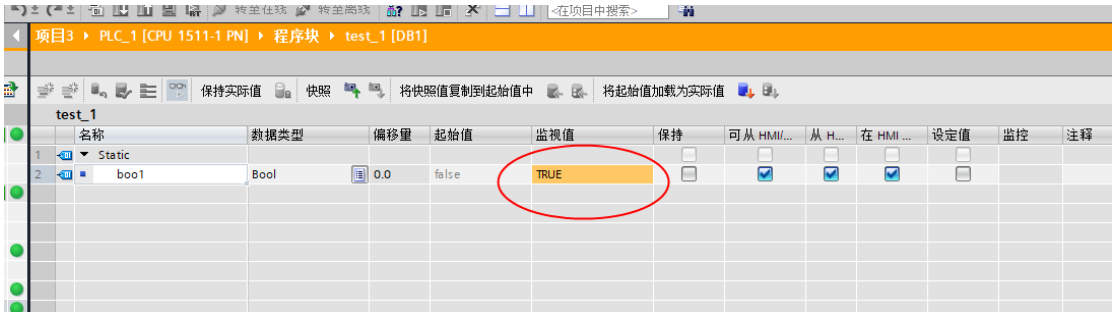
在 TIA 中打开对 DB1 的监控



运行控制台代码，可以看见打印了更改前和更改后的数据

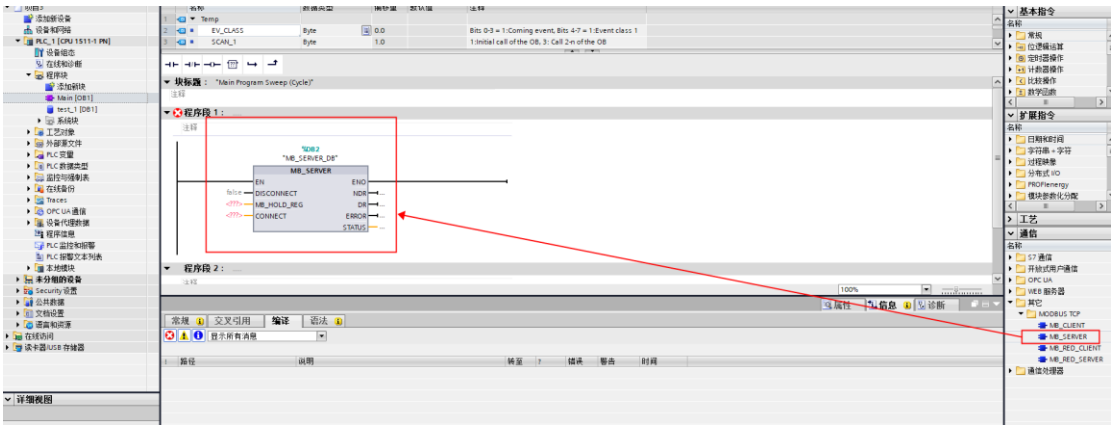


此外 TIA 中监控的对应数据也已经被更改



五 . 使用 Modbus TCP 协议与 PLC 通信

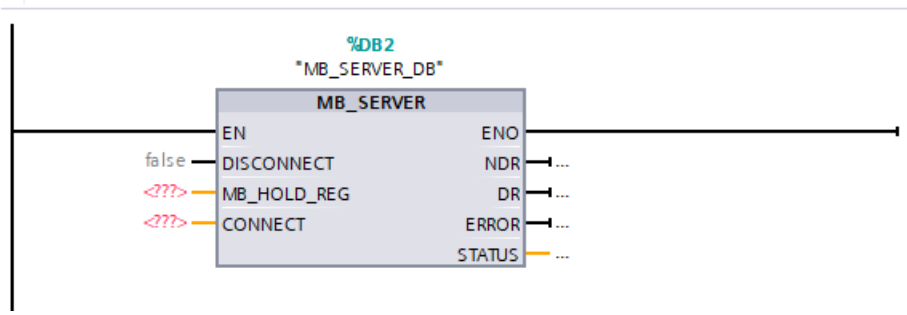
与 S7 协议不同，西门子 plc 要通过 modbus 协议通信需要编写程序端。
在此案例中，plc 作为服务器，c#脚本、通信工具作为客户端对数据进行修改
双击主程序块 Main[OB1]，点击右边栏-通信-其他-MODBUS TCP,找到 MB_SERVER，将其拖动至程序段 1 中，结果如下



可以看到程序段各个管脚参数。

DISCONNECT 代表着连接状态
False 代表在无通信连接时建立被动连接（个人理解为一直建立连接，选这个就好）
True 代表终止连接初始化。如果已置位该输入，那么不会执行其它操作。成功终止连接后，STATUS 参数将输出值 0003。

CONNECT 代表指向连接描述接口的指针，个人理解为用何种地址结构建立连接（通道定义），对应的参数应该是一个数据块，用来存放通道定义
其中可以用的包括 TCON_IP_v4、TCON_Configured（仅限 s7-1500），常用的即是 ipv4

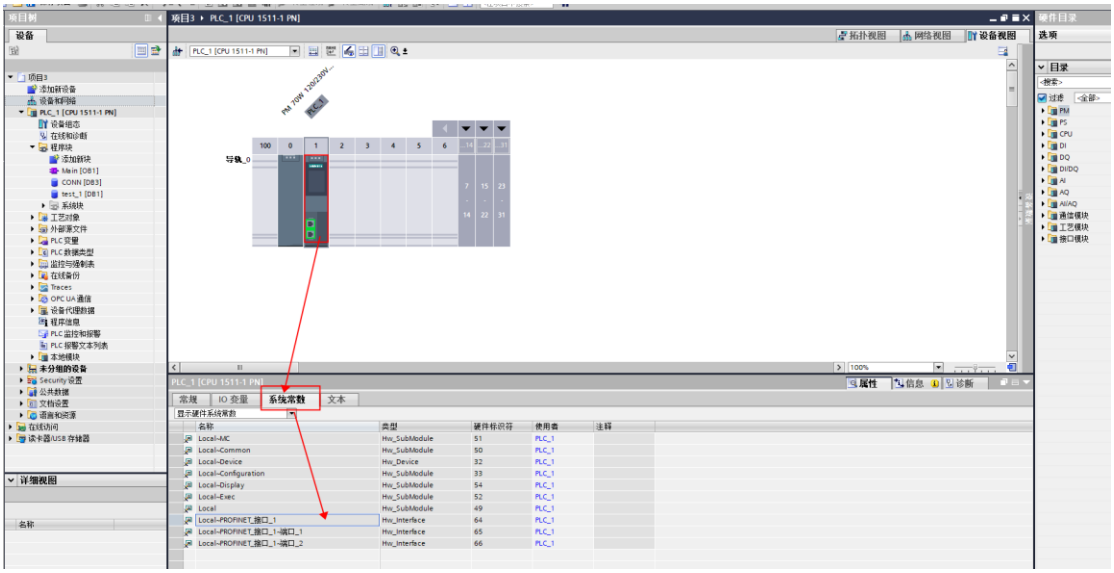


接下来需要配置 CONNECT 管脚的数据块，在左边栏 plc-程序块中新建新块，命名为 CONN

需要注意将新建的数据块属性中的优化块访问取消勾选（第四章第一步）

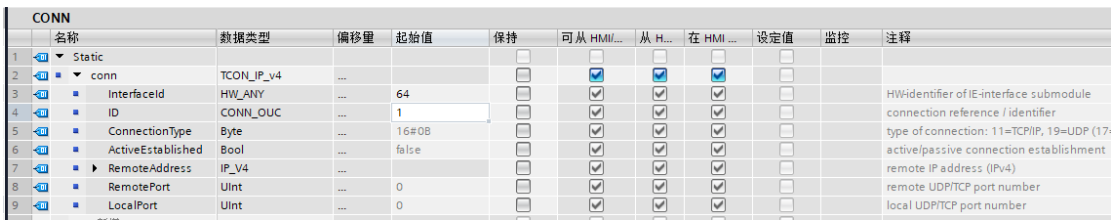
新建变量 conn，数据类型需要**手动输入** TCON_IP_v4，创建好后展开会自动出现以下变量

接下来对这些变量进行配置。对于真实 PLC 物理设备来说都需要一个网口建立连接，**InterfaceId** 则代表接口 id。打开 PLC 的设备视图、双击视图中的 plc，点击下边栏的系统常数



可以看到 PLC 的接口的硬件标识符为 64，65 和 66 则是对 64 的映射地址。因此我们将 conn 中的 interfaceId 设置为接口本身 64 即可。

ID 则代表 PLC 的 id，此处我们只有一个 plc，因此设置 1 即可



ConnectionType 代表来连接类型，TCP/IP=16#0B，UDP=16#13，默认为 16#0B；保持默认值 13 即可

ActiveEstablished 代表是否主动建立连接，demo 中 PLC 是作为服务器，不需要主动建立连接，因此保持默认值为 false

RemoteAddress 远端地址类似一个对象，里面包含一个 ADDR 的数组，组成一个 IP 地址，其作用为限定可访问 PLC 的远端地址。由于我们并不需要限定远端地址，因此将其四个值均设置为 0（默认值），代表任何地址均可以访问 PLC。
注意：如果限制了外部地址，外部地址必须与 PLC 的 IP 地址保持同一个网段

同理 **RemotePort** 远端端口为可访问 PLC 的端口号，此处我们也不需要做限制，默认为 0 代表均可检测。

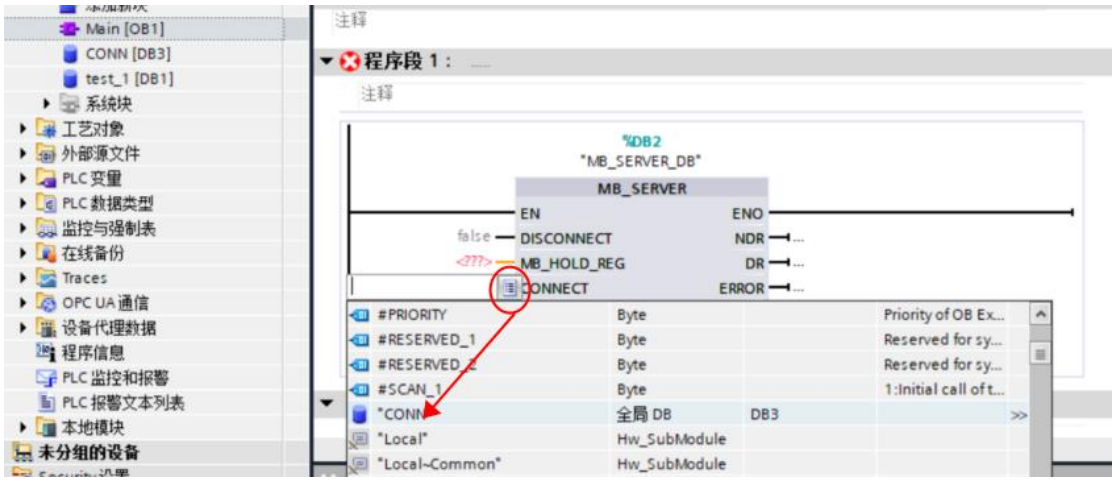
LocalPort 字段为设置 PLC 端口号，一般 PLC Modbus 默认端口号为 502，设置为 502 即可

最终配置好的 DB 块如下图所示

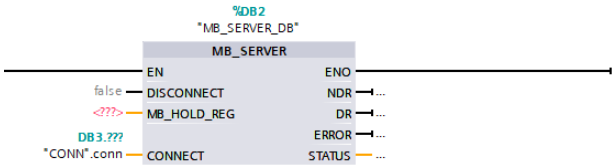
CONN

名称	数据类型	偏移量	起始值	保持	可从 HMI/...	从 H...	在 HMI ...	设定值	监控	注释
Static										
conn	TCON_IP_v4	...								
InterfaceId	HW_ANY	...	64							HW-identifier of IE-interface submodule
ID	CONN_OUC	...	1							connection reference / identifier
ConnectionType	Byte	...	16#0B							type of connection: 11=TCP/IP, 19=UDP (17=TCP
ActiveEstablished	Bool	...	false							active/passive connection establishment
RemoteAddress	IP_V4	...								remote IP address (IPv4)
ADDR	Array[1..4] of Byte	...								IPv4 address
ADDR[1]	Byte	...	16#0							IPv4 address
ADDR[2]	Byte	...	16#0							IPv4 address
ADDR[3]	Byte	...	16#0							IPv4 address
ADDR[4]	Byte	...	16#0							IPv4 address
RemotePort	UInt	...	0							remote UDP/TCP port number
LocalPort	UInt	...	502							local UDP/TCP port number
<新增>										

配置好后，回到 Main 程序块的程序段中，点击 CONNECT 管脚索引下滑找到之前配置的 CONN 全局 DB 块



选中 conn 变量再选择无即可，配置好后如下图所示



在 Modbus 规约中，上位机通过功能码如读（03）、写（06） PLC 下位机中的数据，实际上是先去读写 modbus 的保持寄存器里的数据，PLC 再去根据保持寄存器中的映射关系去更改 PLC 中的数据（个人理解）。
例如 PLC 地址 40001 对应寻址地址 0x0000，40002 对应寻址地址 0x0001。寄存器寻址地址一般使用十六进制描述。

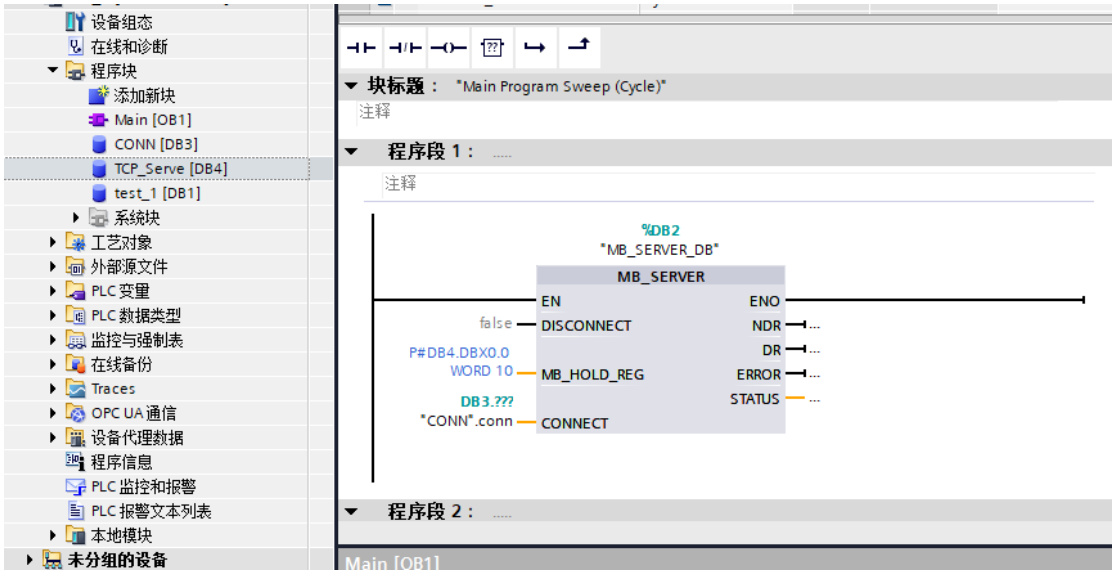
而管脚 **MB_HOLD_REG** 则是指向“MB_SERVER”指令中 Modbus 保持性寄存器的指针。保持性寄存器中包含 Modbus 客户端通过 Modbus 功能 3（读取）、6（写入）、16（多次写入）和 23（在一个作业中读写）可访问的值。

作为保持性寄存器，可以使用全局数据块，也可以使用位存储器的存储区。

因此我们可以直接创建一个全局 DB 块用来匹配管脚 **MB_HOLD_REG 用来做数据的读写。**

在 PLC 程序块下新增一个 DB 块 TCP_Serve，创建过程不再赘述，注意属性中右键取消块优化。在表中创建 10 个 word 类型的变量，默认为 0（可像 Excel 表一样下拉快速创建）
如下图所示

创建好后回到 Main 程序块中设置 MB_HOLD_REG 管脚
手动输入 P#DB4.DBX0.0 WORD 10 代表含义位使用 DB4（刚刚创建好的 TCP_Serve DB 块的编号），从 0.0 开始保存 WORD 数据类型，最后面的 10 代表之前创建变量的数量。
设置好的管脚如下图所示



程序块右侧主要代表状态，各个管脚的数值意义如下：

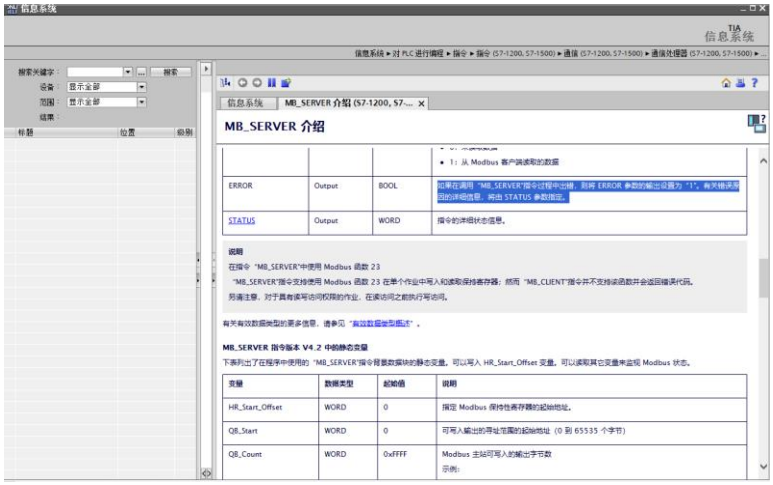
NDR： 是“New Data Ready”的缩写，0（false）代表无新数据，1(True)代表从 Modbus 客户端写入的新数据

DR： 是“Data Read”的缩写，0（false）代表未读取数据，1(True)代表从 Modbus 客户端写入的数据

ERROR： 报错状态位，如果程序段出现错误，状态则为 1(True)，否则为 0（false）

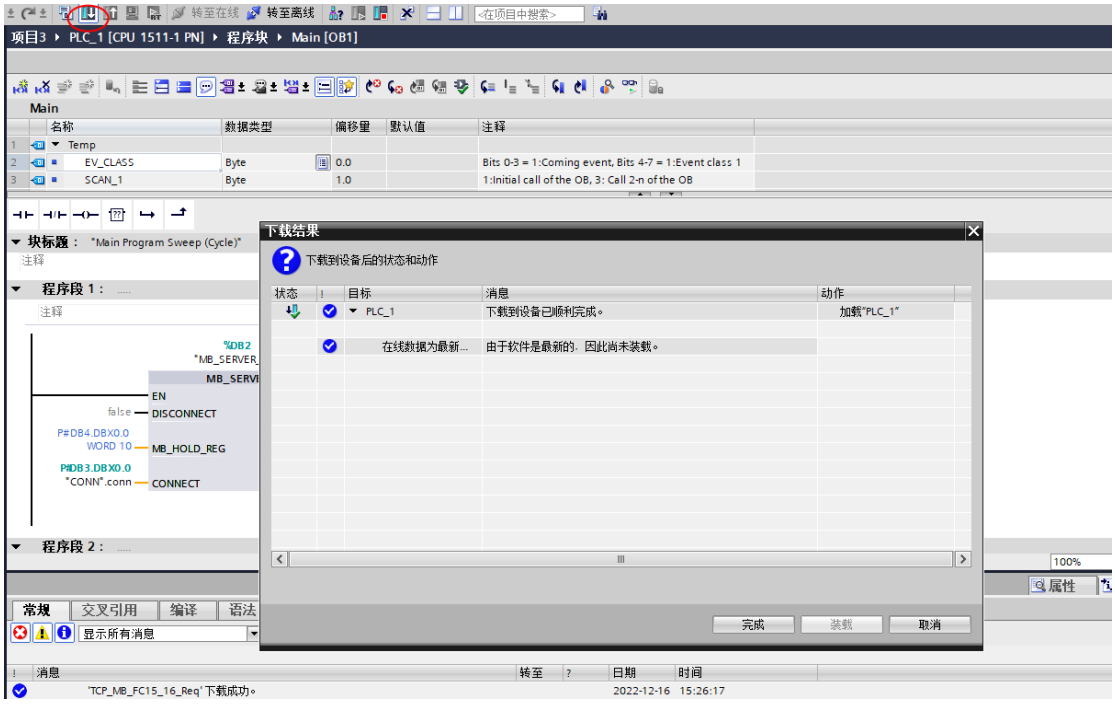
STATUS：代表指令的详细状态信息，（可理解为前端的错误码，如 404、502、201 等）

如想要了解更多信息，我们可以选中程序块，按 F1 显示帮助手册，对学习其他内容也很有帮助

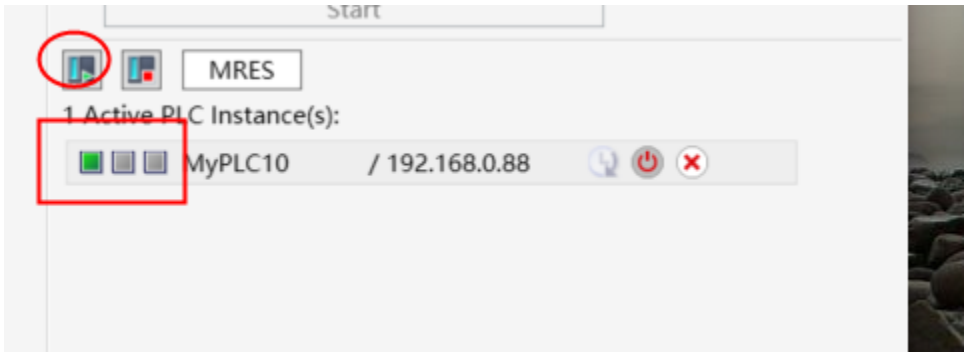


以上参数并不是必填项，由于我们并不需要去监测他们的状态，因此可以不配置。如想要检测可以再去创建一个全局 DB 去匹配。

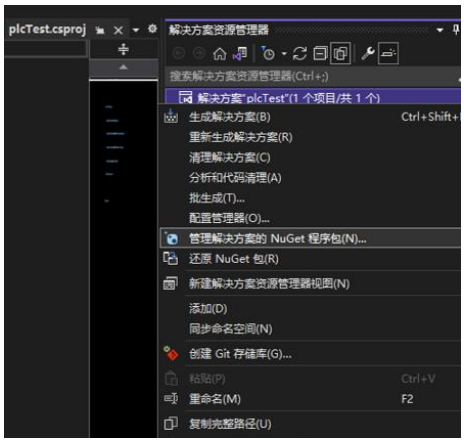
到此为止 PLC 程序配置完毕，将程序下载至我们虚拟好的 PLC 中（同第四章下载操作）



打开 S7-PLCSIM Advanced V3.0，选中虚拟 PLC 点击上方运行，会亮起绿色小灯代表成功



打开 VS 2022，新建 C# 控制台应用，右边栏右键点击管理程序包



搜索安装以下工具包

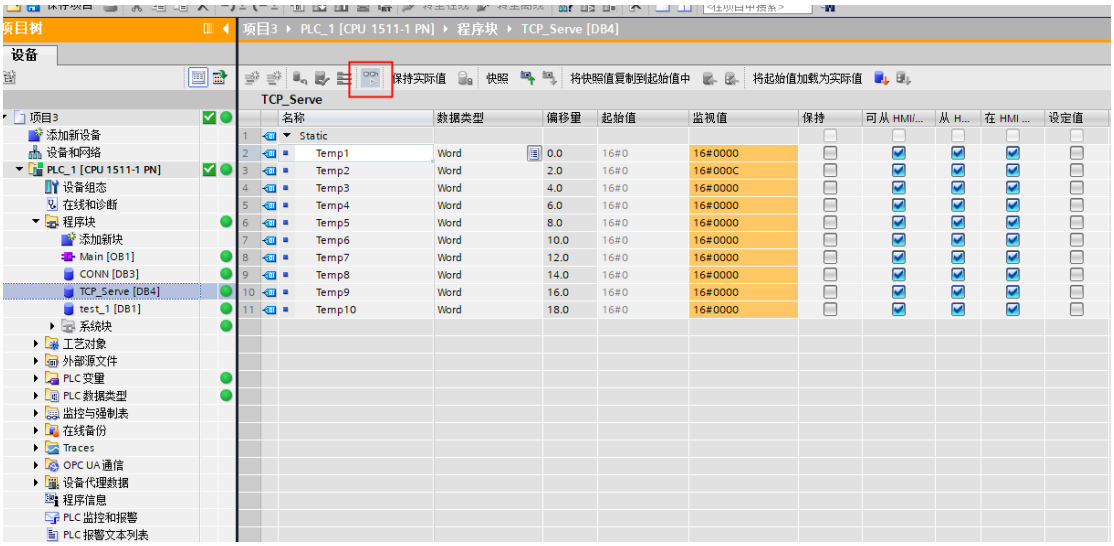
Nmodbus4 是使用 MIT 协议的 C#库，简便快捷

然后写入以下代码

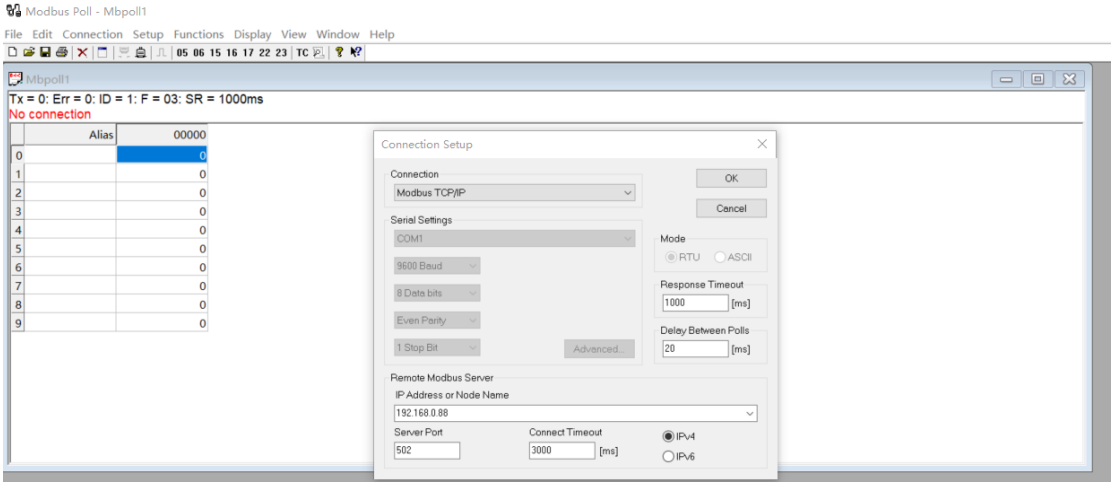
```
using Modbus.Device; //引入命名空间
using System.Net.Sockets;
namespace myPLC
{
    0 个引用
    class Program
    {

        0 个引用
        static void Main(string[] args)
        {
            ModbusIpMaster master;
            TcpClient client;
            Console.WriteLine("准备连接");
            client = new TcpClient("192.168.0.88", 502); //创建连接，IP地址、端口号与plc保持一致
            // client.Connect(IPAddress.Parse("192.168.0.88"), 502);
            master = ModbusIpMaster.CreateIp(client); //mastert原本为null，如果连接成功则不为nul
            if (master != null)
            {
                Console.WriteLine("连接成功"); //打印信息
            }
            else
            {
                Console.WriteLine("没有连接上");
            }
        }
    }
}
```

运行代码结果如下，连接成功

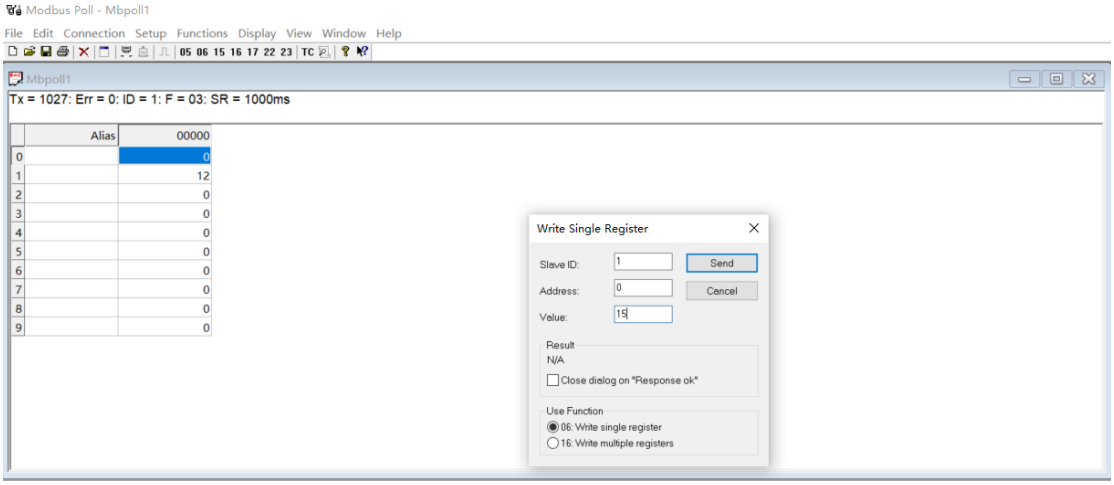
此外，我们还可以通过工具 Modbus Poll（见第一章）建立连接、进行数据读写。
首选打开 TIA v15
选择 TCP_Serve 进行监控数据



打开 Modbus Poll，点击上方栏 Connection-connect,弹出下图弹框，选择 connection 为 Modbus TCP/IP，下方输入虚拟 PLC 的 IP 地址、端口号。



双击表格弹出弹框，将地址 0（映射为 DBX.0.0）value 改为 15，点击发送



可以观察到对应数据改变，读写成功

六．参考学习资料

基于 PLCSIM-Advanced 搭建 ModbusTCP 通信仿真环境
https://zhuanlan.zhihu.com/p/213713802

C#通过 S7.Net 读写西门子 PLC 数据：
https://www.bilibili.com/video/BV1Rf4y1J7tX/?spm_id_from=333.788.recommend_more_video.0&vd_source=83d2130f4c5f6c8184cc12a6c6b98a79

用 C#通过 visual studio 来读写 PLC：
https://www.bilibili.com/video/BV1jK4y1t77a/?spm_id_from=autoNext&vd_source=83d2130f4c5f6c8184cc12a6c6b98a79

西门子 S7-1500 PLC 的 MODBUS TCP 通信：
https://zhuanlan.zhihu.com/p/403679566

C#Modbus 通讯协议和 PLC 数据交换详解：
https://www.bilibili.com/video/BV1Se4y1h7Sr/?spm_id_from=333.337.search-card.all.click&vd_source=83d2130f4c5f6c8184cc12a6c6b98a79