

# PLC实验 S7-300超详细硬件组态实验过程

原创

山重水复疑无路@ 于 2020-06-10 22:06:40 发布 6662 收藏 29

分类专栏: [PLC学习系列](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: [https://blog.csdn.net/qq\\_45252077/article/details/106675693](https://blog.csdn.net/qq_45252077/article/details/106675693)

版权



[PLC学习系列](#) 专栏收录该内容

25 篇文章 12 订阅

订阅专栏

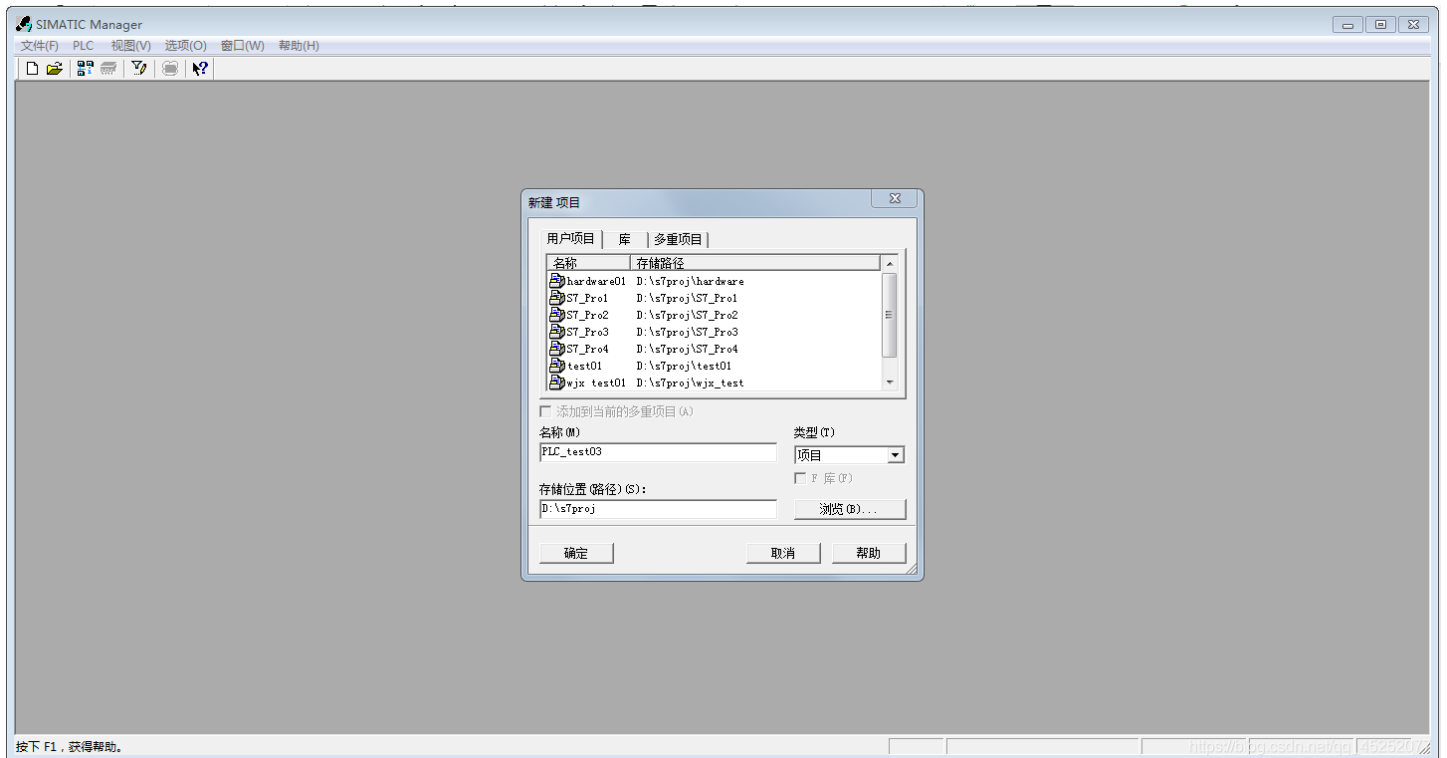
## PLC S7-300超详细硬件组态实验过程

实验要求:

- 1、组态s7-300站。
- 2、系统要求: CPU315-2DP, 1个16点DI(数字量输入), 一个16点DO(数字量输出), 一个8点AI(模拟量输入), 一个4点AO(模拟量输出), 需要分配IO地址。
- 3、描述组态过程。
- 4、描述DP地址分配和IO地址分配。

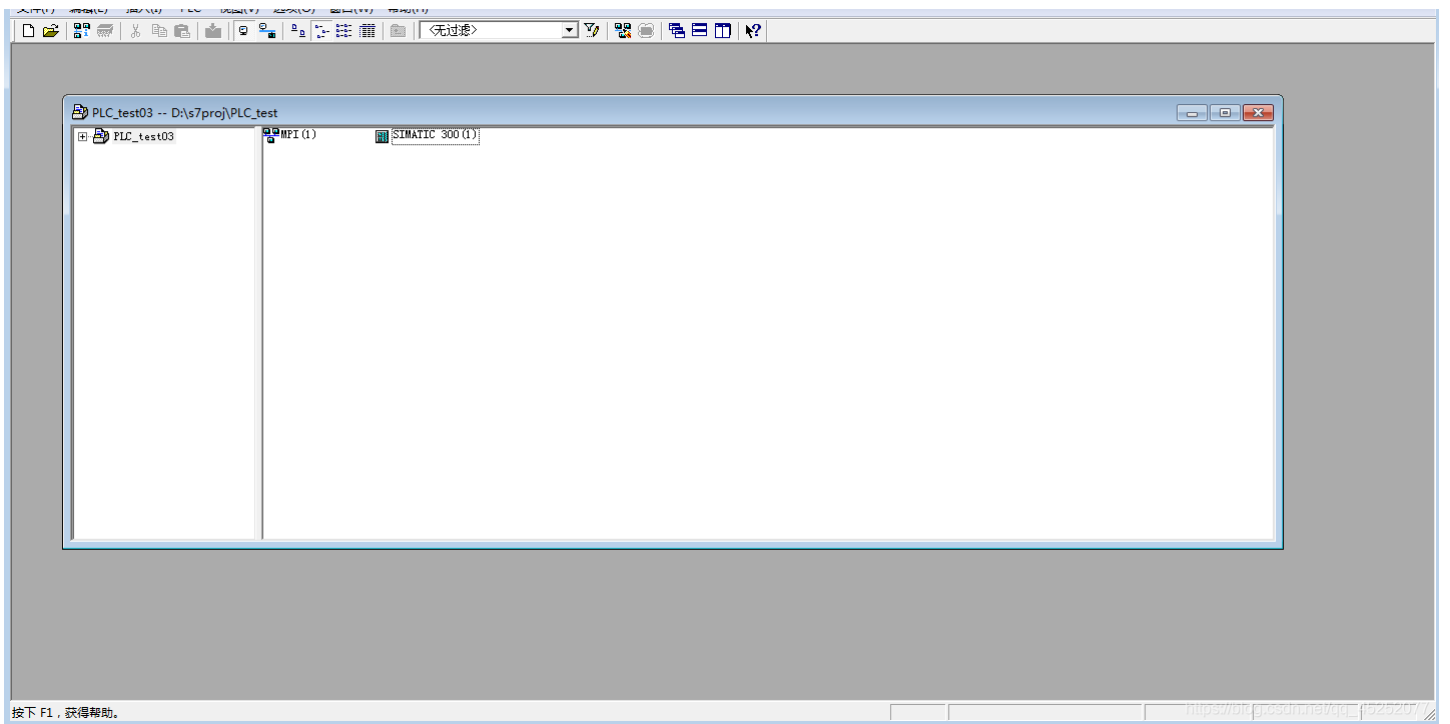
### 一、组态S7-300站

(1) 打开STEP7软件, 新建一个项目, 输入项目名称, 如图所示为PLC第三次实验PLC—test03

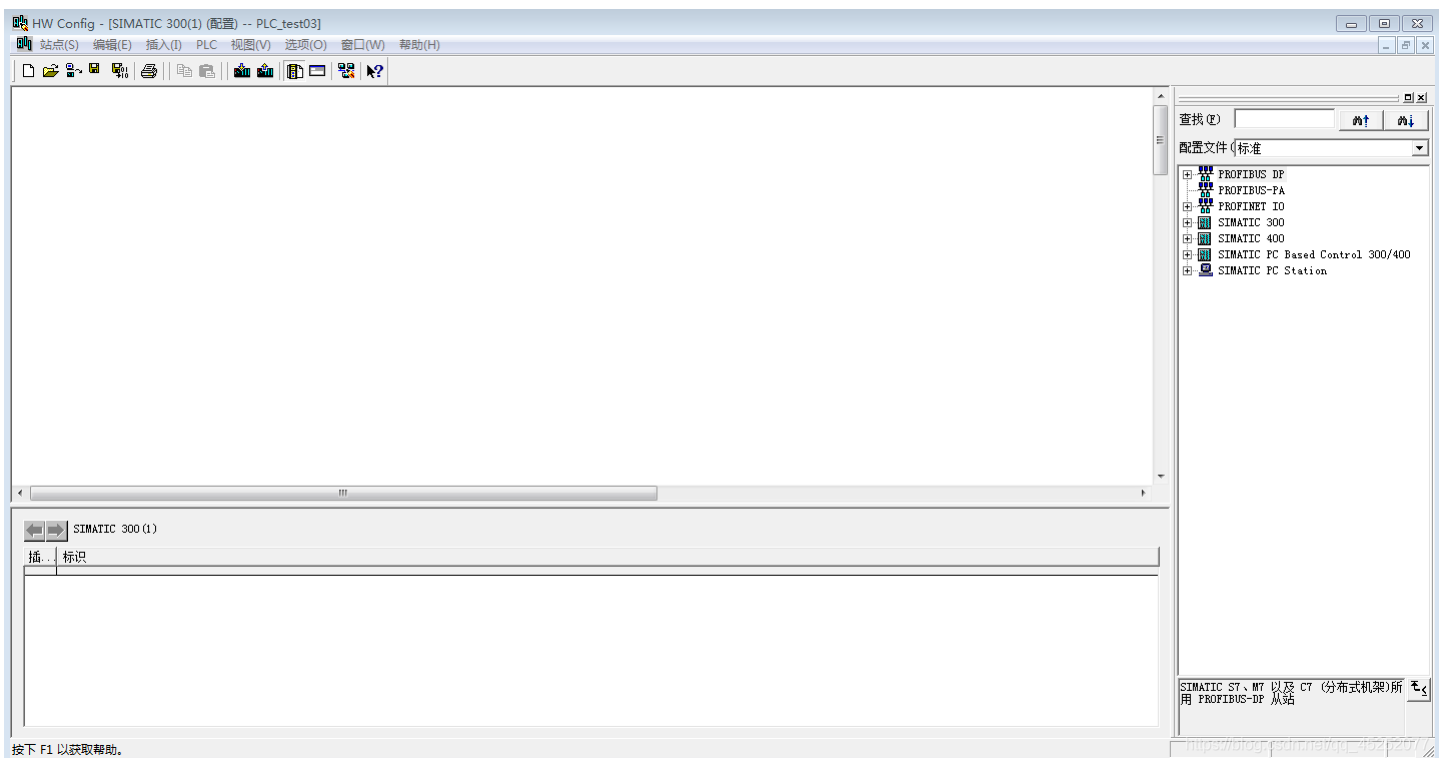


(2) 项目上右键选择插入新对象—>SIMATIC 300 站点

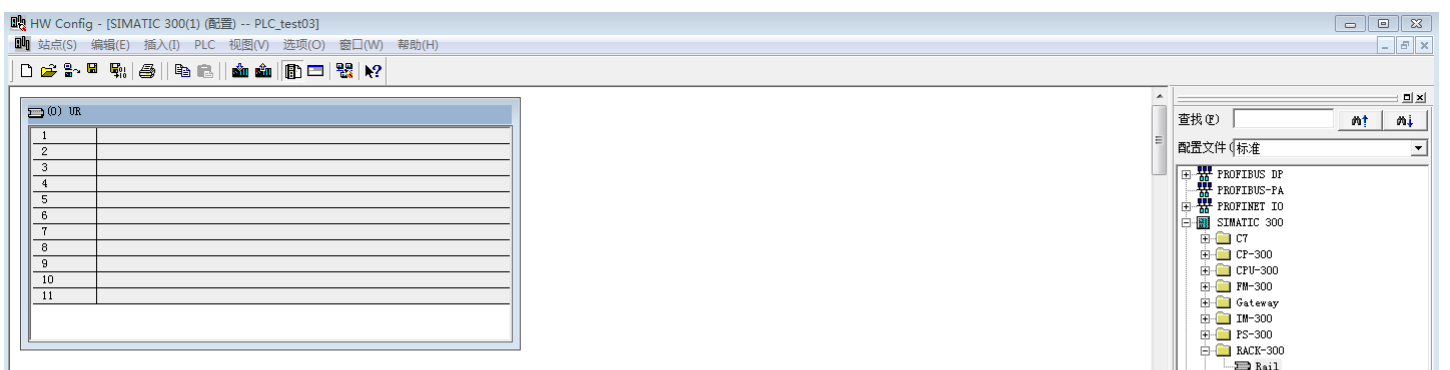


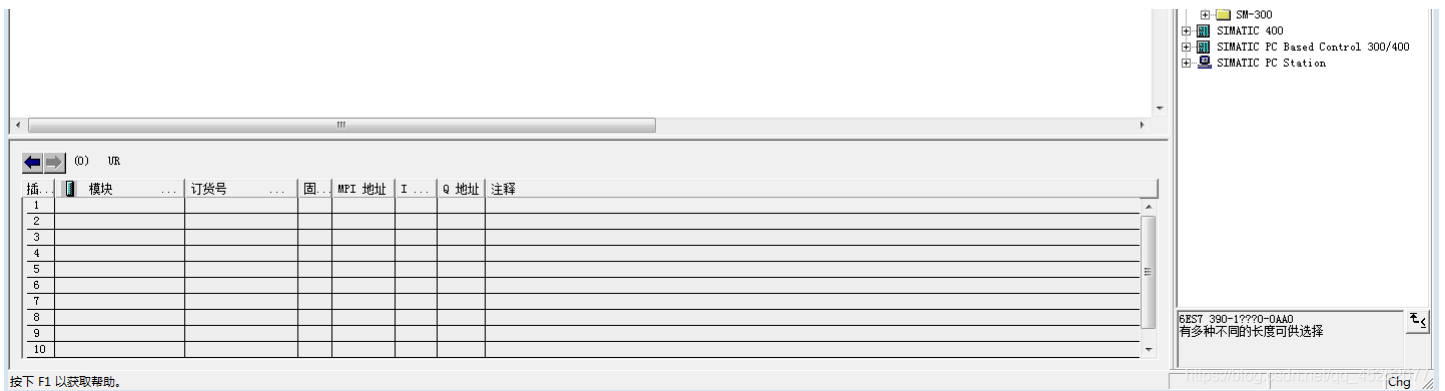


(3) 双击SIMATIC 300站点，找到硬件双击进入，开始硬件组态

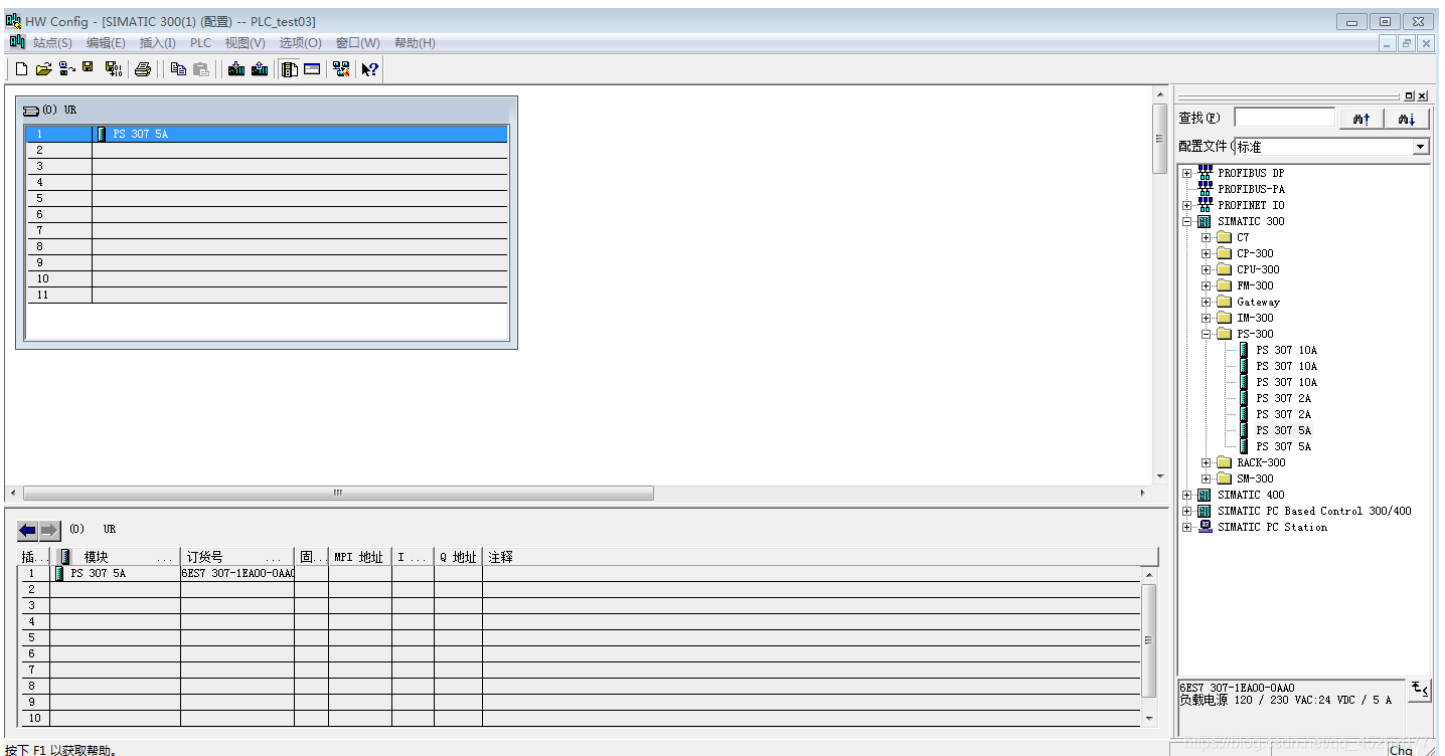


(4) 先插入导轨（所有模块的安装都是基于导轨的），选择SIMATIC 300—>选择RACK-300—>双击Rail—>插入导轨

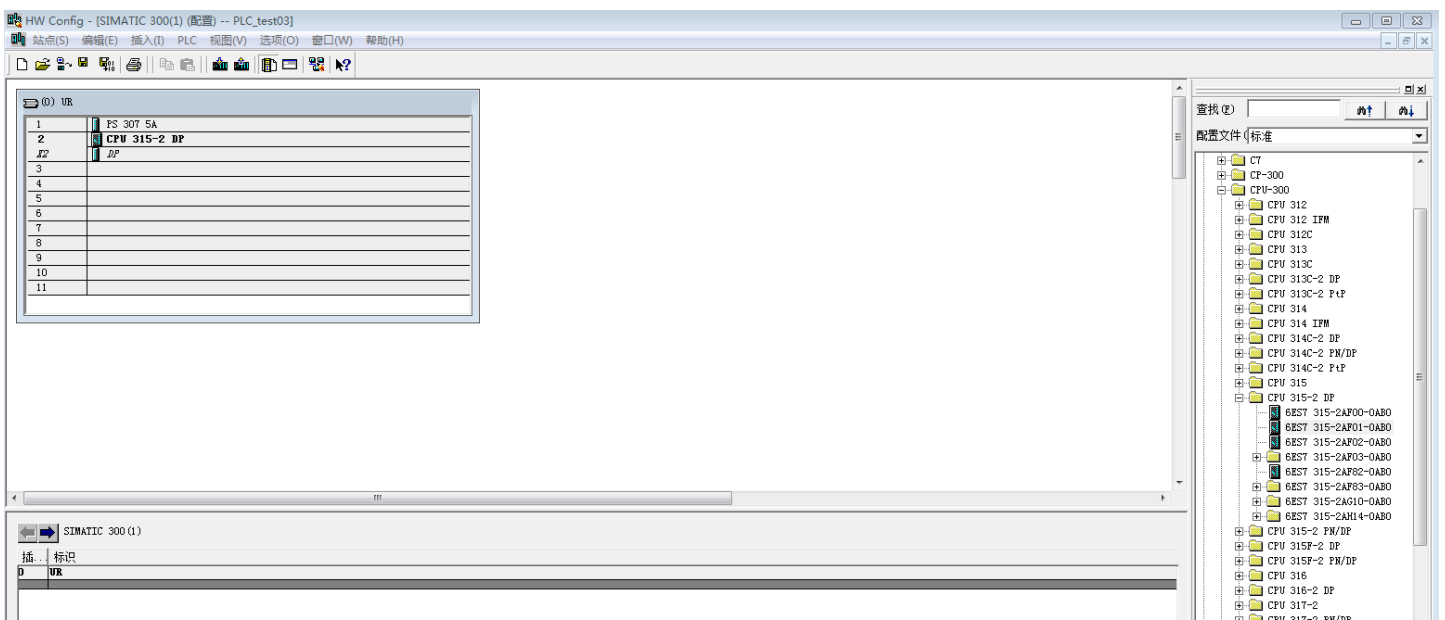


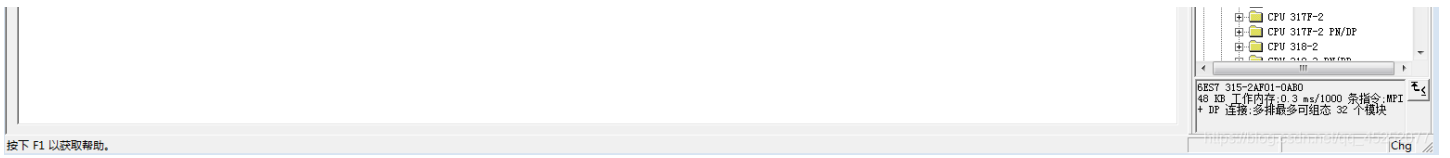


(5) 之后开始按顺序与实验要求插入各个模块，PS电源模块，CPU模块，以及四个SM信号模块，其中信号模块有（DI、DO、AI、AO）。先插入PS电源模块，实验并没有对电源模块进行特殊要求，我选择的是PS 307 5A电源模块。

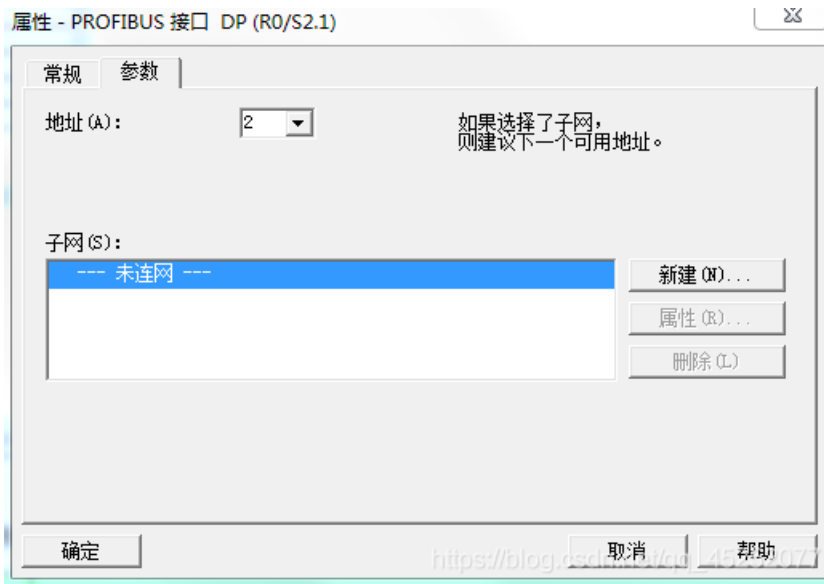


(6) 第二个位置我们选择CPU，实验要求选择CPU315-2DP，我选择的是CPU315-2DP目录下的6ES7 315-2AF01-0AB0，插入到第二个槽中。



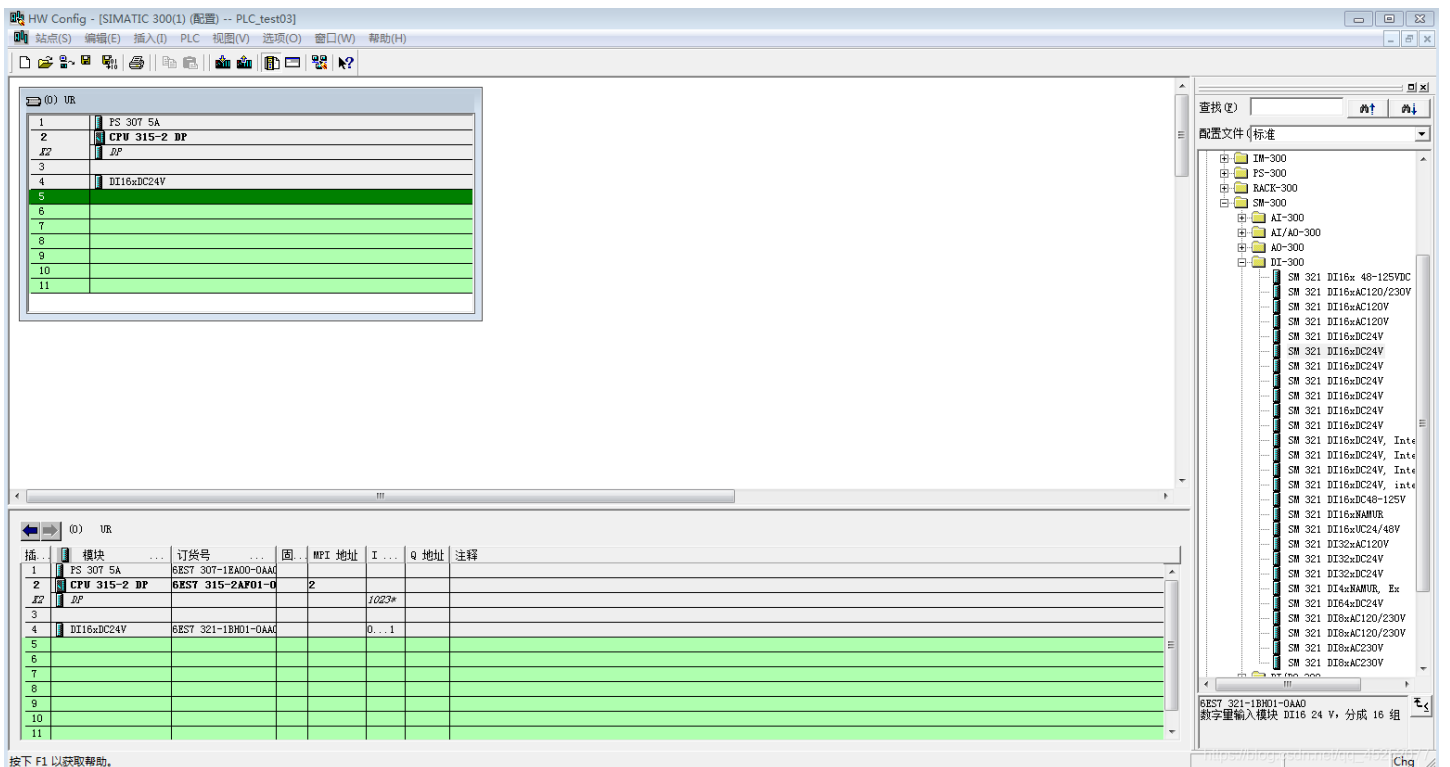


(7) 插入过程中让选择地址，先不管选择系统默认点击确定就OK

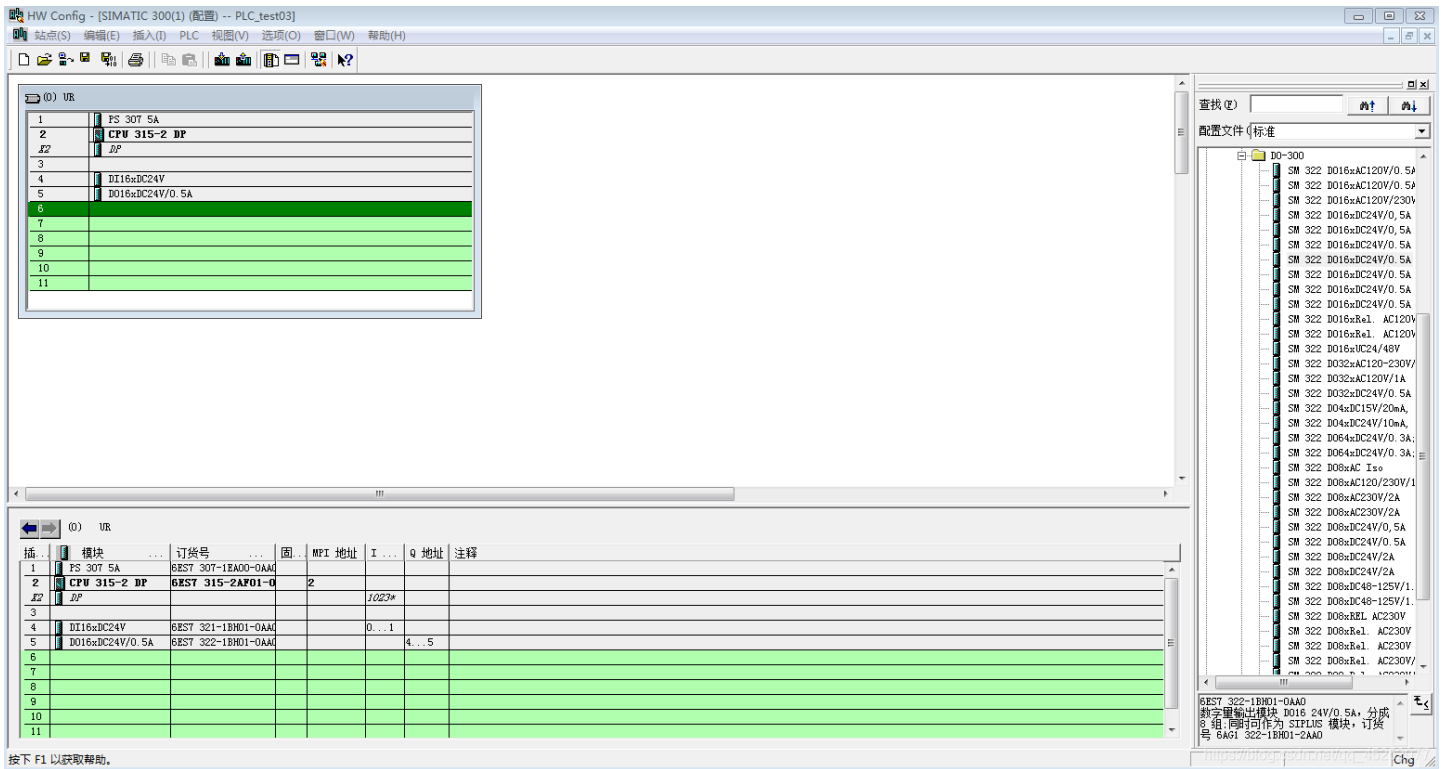


(8) 第三个位置我们空出来，作为接口模块插入 连接其他导轨，因为毕竟一个导轨上最多也只能插入8个信号模块，再进行大型项目时是远远不够的。

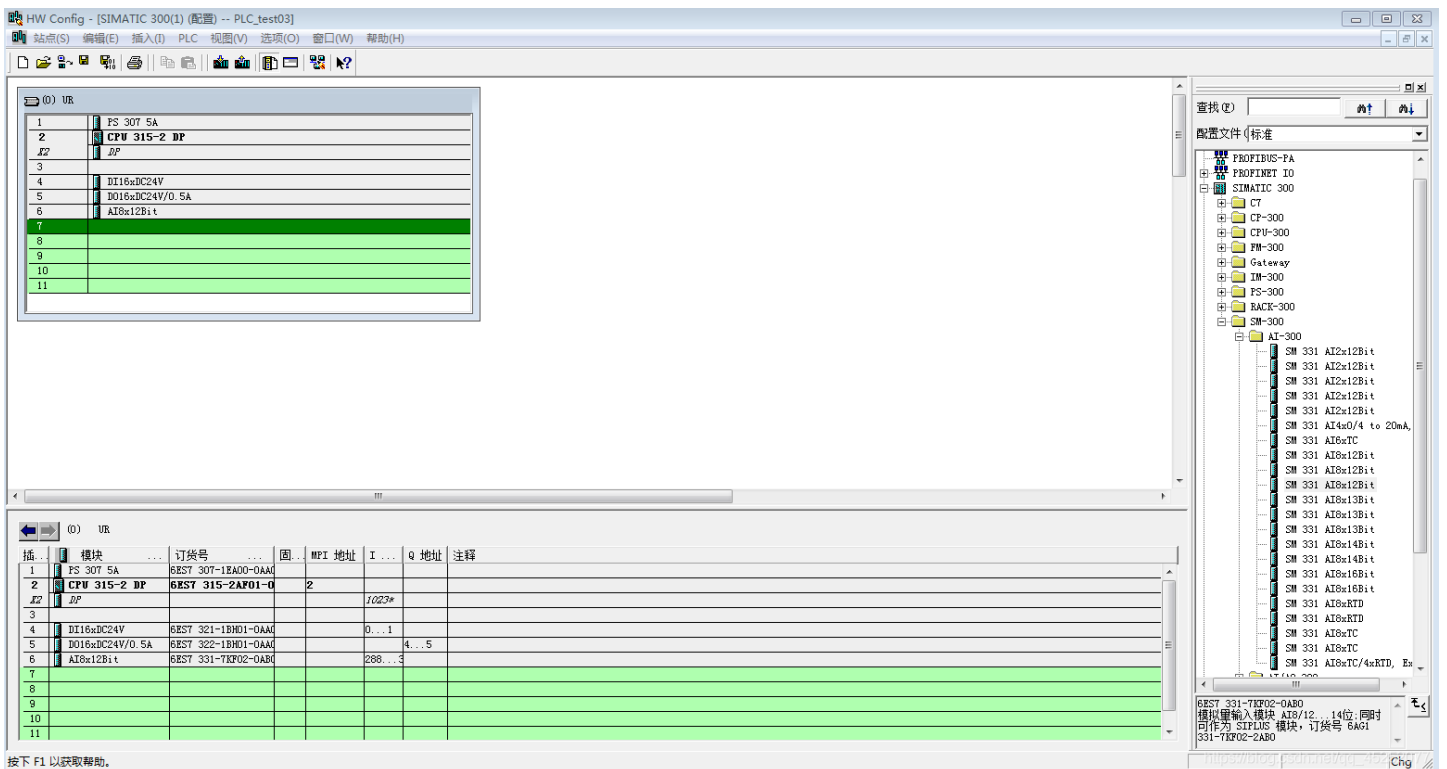
(9) 接下来组态信号模块，数字量输入DI，根据实验要求需要1个16点DI，先找到SM-300目录，可以很快的找到DI-300目录，点开之后可以发现非常多的DI选项，我们随便选一个16点的（满足题目要求）DI双击插入第四个槽位就OK，我选择的DI是SM 321 DI16xDC24V。



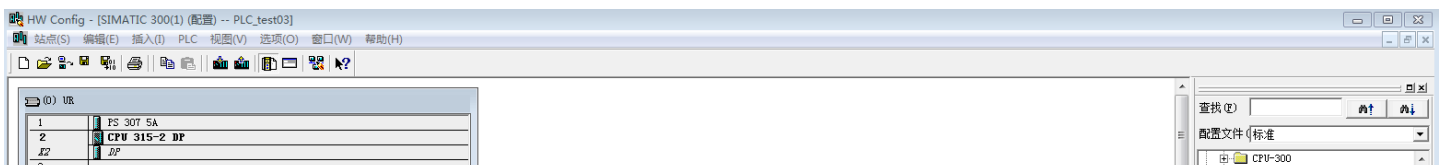
(10) 第五个插槽我们插入DO（数字量输出0）模块，实验要求也是16点DO，在DO-300模块里我们可以选择组态一个16个输出点的输出模块，我选择的是SM 322 DO16xDC24V/0.5A双击插入如图。

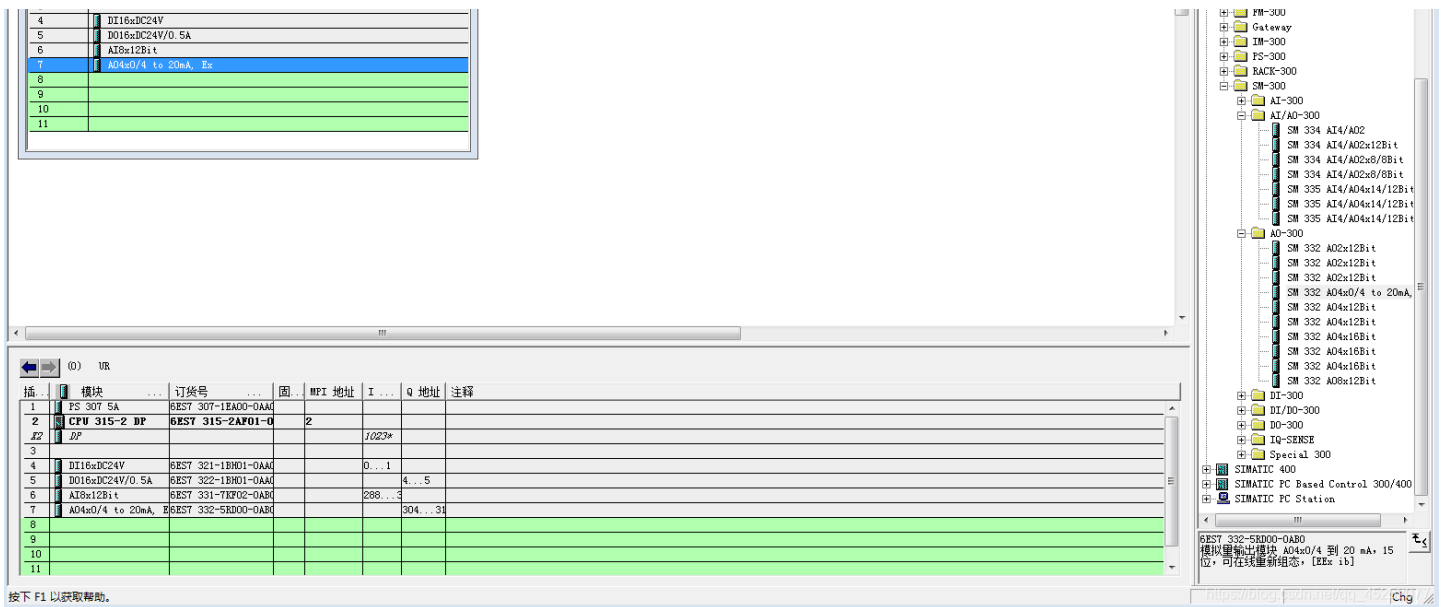


(11) 接下来我们来组态AI（模拟量输入）模块，实验要求是一个8点AI，在AI-300目录下选择一个8通道的AI模块，我选的是SM 331A8x16bit双击插入如图所示。



(12) 第七个槽位我们来插入AO模块（模拟量输出模块），题目要求是插入一个4通道的，在AO-300目录下选择一个4通道的AO模块插入，但是我找了找没有发现4通道的AO模块，思路都是一样的我就选择插入了SM 332 A04X0/4 to 20mA Ex,如图所示：





(13) 现在我们已经将硬件组态好了，接下来就是进行地址分配了。

## 二、地址分配

(1) 先打开一个D模块，点击地址如图：



### 分析描述：

**First:** 可以看到下面有一个系统默认的可勾选选项，选上之后系统会自动的给我们分配我们所需要的地址，默认地址开始是0，结束是1，也就是说从I0.0开始，结束的时候在第一字节I1.7。显示的只是所在的首字节和末字节的地址，不告诉我们具体那一位。我们可以将系统默认选项给去掉。之后就可以任意修改了。

**Second:** 硬件如何和地址进行关联，D模块是一个16点的，也就是说有16个输入点的D模块，我们输入起始地址位0，结束地址自动分配位1。从0.0开始在1.7结束，一个有16个点。0.0地址与我们D模块的第一个通道是关联的。0.2就是和我们模块的第二个通道相关联的。最后以此类推我们的1.7就是和第16个通道进行关联的。这就是我们如何将我们硬件的点位与我们实际的地址跟我们分配的地址进行关联的。

如果此处开始不用0换一个其他的也可以按照以上分析进行推算，举一个例子吧，如果我们将起始的改为3、结束默认自动就会变成4了。如图



这样我们编程第一个地址I点就变成3.0了，第二点位输入点就变成了3.1，以此类推最后一个点位是4.7。

(2) 对DO模块地址进行分配，同样双击DO模块进入地址栏，如图所示。

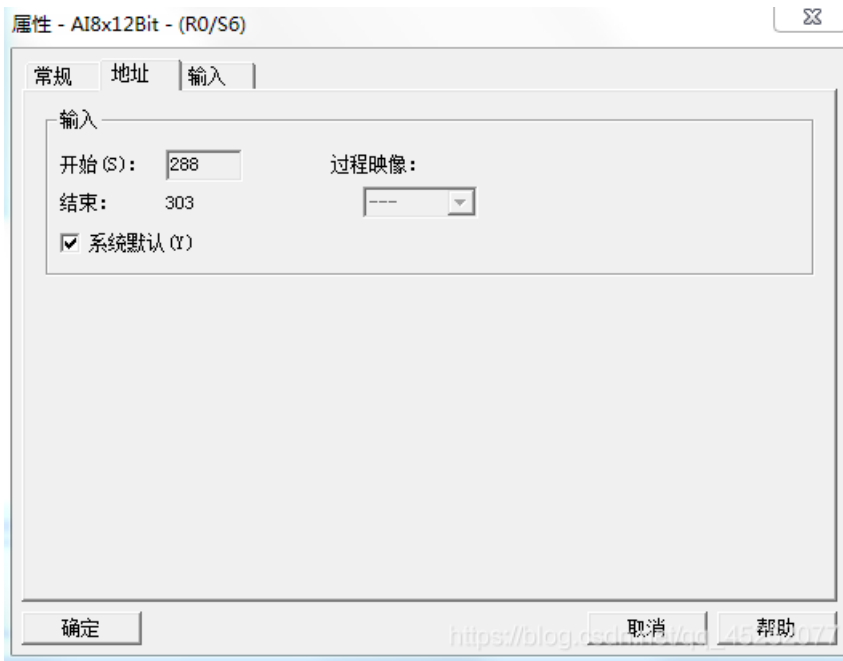


系统默认的起始地址是Q4.0，结束的首地址是5。在这里我想使它与输入I相对应就把他的起始地址也改为了0，默认地址首字节就自动变成了1如图所示

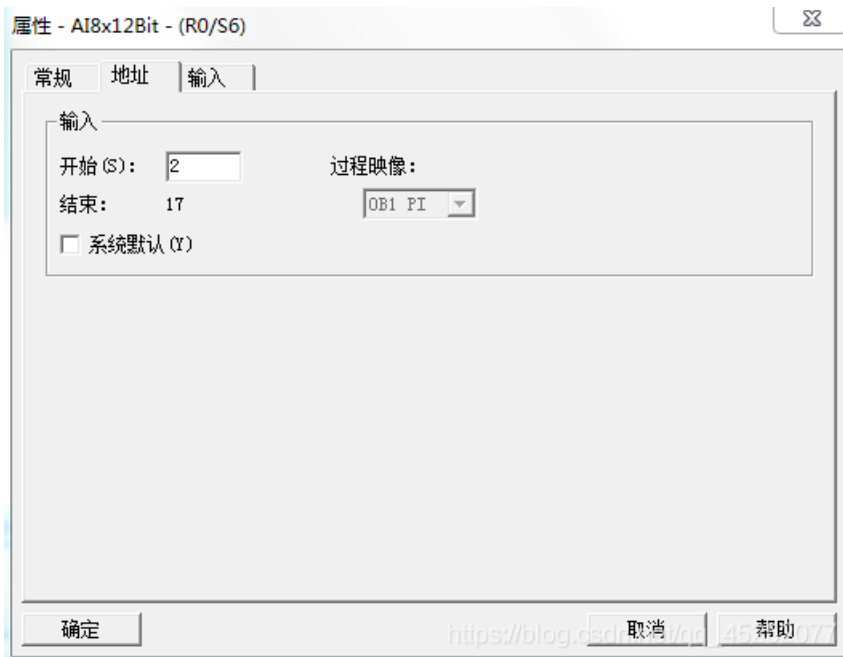




(3) 继续分配AI八通道的模拟量输入模块，每一个通道 (bit) 都占我们16个位的地址，先来看一下默认地址，如图所示

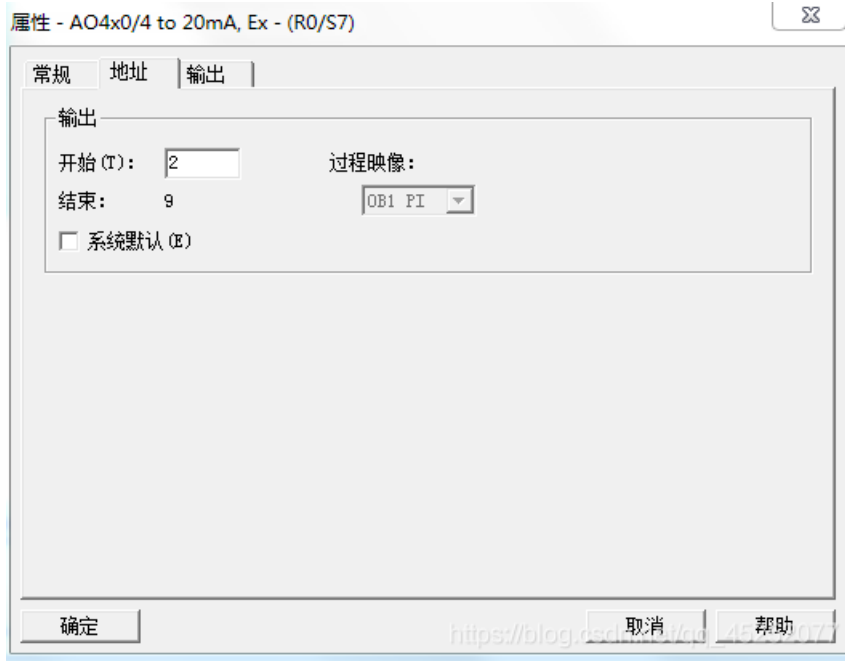


开始地址是288，结束地址是303，AI同为我们的输入模块在我们的输入区。与DI共享同一块区域，只不过AI每一个通道 (bit) 占16位的地址，DI每一个位只占一个位地址。在这里DI模块用到了0和1，在这里我就把AI的起始地址改为2，结束地址就是17。

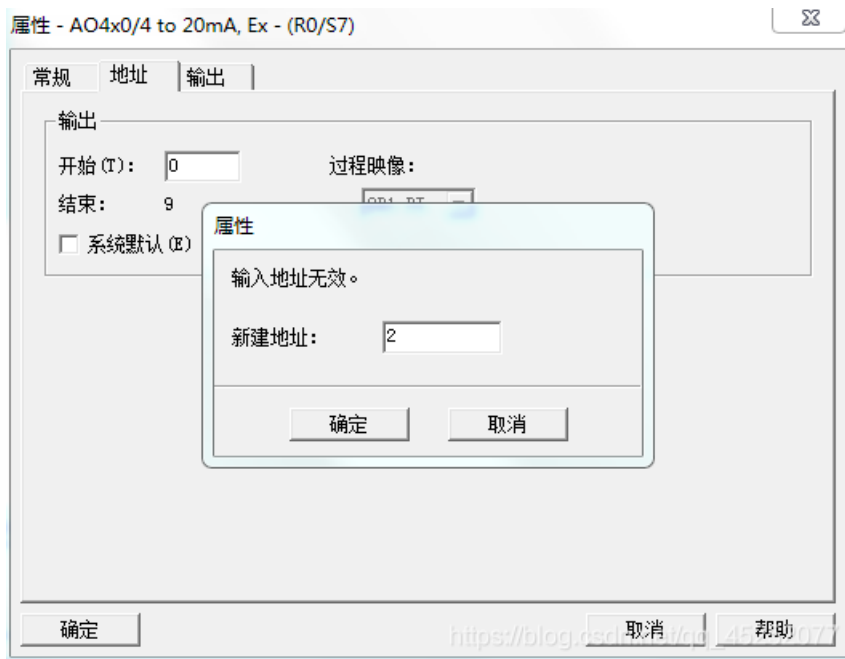


也就是说我们的AI模块第一个通道占用的地址就是Iw2.0到Iw3.7,以此类推的话最后一个通道占用的地址是I16.0到I16.7。

(4) 最后分配我们的AO模块，AO模块与AI模块类似。与DO模块一样占用的是Q区，一样一个通道占16个位的地址。我将AO模块的起始地址也改为2.0，如下图所示



(5) 当我们的地址选用冲突时即AO模块与DO模块占用地址相同时，会有提示，说我们的输入地址无效，提示我们重新输入，如图所示



(6) 这样我们的硬件就完整的组态好了。地址也分配完毕且进行了分析。最后可以进行保存和编译，编译就是将我们硬件组态中的看是否有错误，如果有错误会有提示可以进行修改。和C语言一样，没错误的话直接就会保存。实验圆满完成。