

Modbus 网关使用手册

产品型号：H01TCP-4



厦门海为科技有限公司

2012 年 10 月

目 录

1 前言	- 1 -
1.1 声明	- 1 -
1.2 技术支持	- 1 -
2 概述	- 1 -
2.1 功能描述	- 1 -
2.2 运行环境	- 2 -
2.3 支持寄存器类型及个数	- 2 -
2.4 应用领域	- 3 -
3 使用说明	- 3 -
3.1 选择操作语言	- 3 -
3.2 选择监控模式	- 4 -
3.3 添加驱动	- 5 -
3.4 添加通道	- 7 -
3.5 添加设备	- 10 -
3.6 添加标签	- 11 -
3.7 Modbus 服务器（从站）设置	- 18 -
3.9 上传工程	- 23 -
3.10 网关参数设置	- 25 -
3.11 下载工程	- 28 -
4 WEB 服务器	- 32 -
4.1 网页登陆	- 32 -
4.2 下载文件	- 33 -
4.3 用户管理	- 34 -
4.4 网口设置	- 35 -
4.5 串口设置	- 35 -
4.7 通讯状态	- 36 -
4.8 实时数据	- 37 -
5 Modbus 客户端（主站）访问	- 39 -
6 JS 脚本逻辑控制	- 40 -
6.1 操作步骤	- 40 -
6.2 函数说明	- 44 -
6.3 使用范例	- 46 -
6.4 JS 实现 PID 逻辑控制	- 47 -
7 硬件网关说明	- 50 -
8 恢复出厂 IP 设置	- 55 -

1 前言

1.1 声明

本手册属于厦门海为科技有限公司及授权许可者版权所有，保留一切权利，未经本公司书面许可，任何单位和个人不得擅自摘抄、复制本书内容的部分或全部。由于产品版本升级或其他原因，本手册内容有可能变更。厦门海为科技有限公司保留在没有任何通知或者提示的情况下对本手册的内容进行修改的权利。本手册仅作为使用指导，本公司尽全力在本手册中提供准确的信息。

1.2 技术支持

- 技术支持热线电话：0592-2230312
- 网址：www.haiwell.com

2.1 功能描述

- H01TCP 是一款功能很强大的协议转换网关，使得用户可以根据现场设备的通信协议进行配置，转成标准的 Modbus 协议。在 PC 端仿真运行无误后，上传到硬件协议转换网关。**注意本网关采集端与转发端的 Modbus 寄存器基地址都是从 1（Base1）开始。**
- 工作原理：H01TCP 相当于一座通信桥梁，将其它非标准通信协议转成 Modbus 标准通讯协议，使得支持 Modbus 协议的上位机软件（譬如西门子的 WinCC、Wonderware 的 Intouch、亚控的组态王、力控等组态软件）通过硬件协议网关可以和不同设备互相通信，方便系统集成。
- 本软件优点：
 1. 绿色免安装，可配置性强，操作简单，稳定可靠，故障诊断方便。
 2. [支持 JS 脚本，实现逻辑控制](#)。
 3. 支持多国语言切换，方便用户操作。

4. 在 PC 监控模式下，配置软件 H01TCP，可用于 PC 仿真。
5. 网关内置 WEB 服务器，用户可以通过浏览器查看实时变化的数据和通信状态，方便现场调试。另外，还可以下载配置软件 H01TCP 及工程文件。
6. 支持各种不同的协议同时转为 Modbus 协议。
7. 网关模拟量支持线性转换，支持取位功能，高低字节交换功能。
8. 支持用户权限管理

2.2 运行环境

- H01TCP 配置软件支持 Windows XP/2000/2003/Win 7/Win8。
- WEB 支持 IE8 及以上版本，Opera、苹果的 Safari、Google Chrome 以及火狐浏览器。

2.3 支持寄存器类型及个数

网关内部支持一个 ModbusTCP 服务器 和一个 ModbusRTU 服务器，4 种寄存器一共支持 1024 个点位，具体点位由用户自由分配，限制点数如下表 2-3-1。

Modbus 寄存器类型		Modbus 寄存器个数
0x (Coil Status)	可读可写	1024(1~1024)
1x (Input Status)	只 读	1024(1~1024)
4x (Holding Register)	可读可写	1024(1~1024)
3x (Input Register)	只 读	1024(1~1024)

表 2-3-1 支持 Modbus 寄存器个数

2.4 应用领域

硬件网关 H01TCP—04 用于解决 Citect、IFIX、RSVIEW、WINCC、组态王、

易控等组态软件无法连接一些不常见的控制设备。譬如工控里的组态软件要访问支持 BACnetIP 协议的网控器，通过硬件网关转换成 Modbus 标准协议，以供支持 Modbus 客户端软件进行访问。

3 使用说明

H01TCP 是我司硬件网关 H01TCP—04 的上位机配置软件；当然也可以单独放在 PC 上使用，无时间限制版本，需要购买 USB-KEY 加密狗或软授权。

3.1 选择操作语言

首先打开运行主程序 H01TCP.exe。进入主程序界面，点击视图菜单选择“语言设置”，如下图 3-1-1。

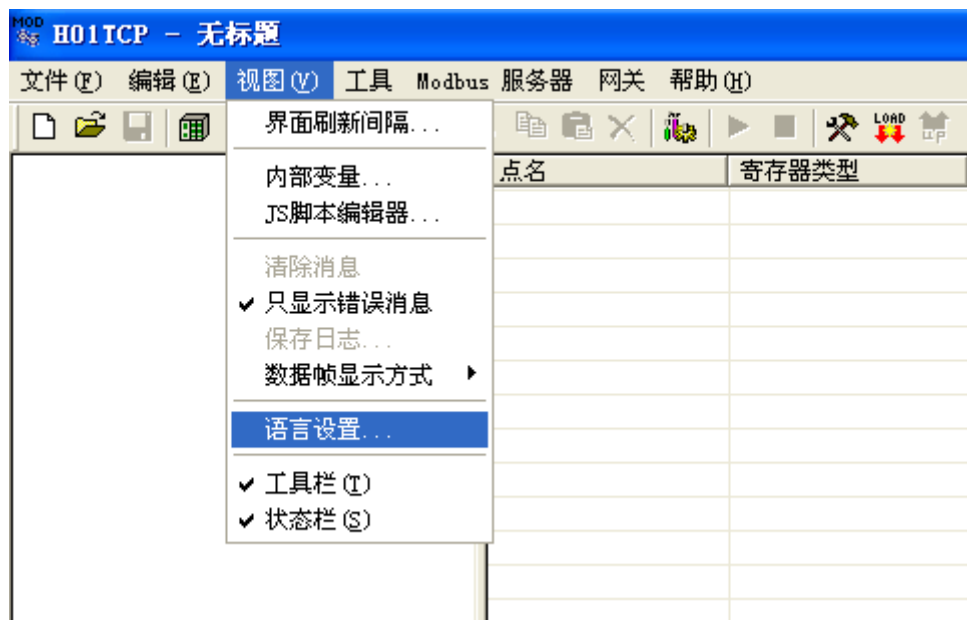


图 3-1-1 选择语言设置

注意：上位机配置软件 H01TCP 以及上传的工程可以从网关内部下载，操作步骤请查看第 4 章节 [WEB 服务器](#)

在弹出来的对话框中选择操作语言，如下图 3-1-2。

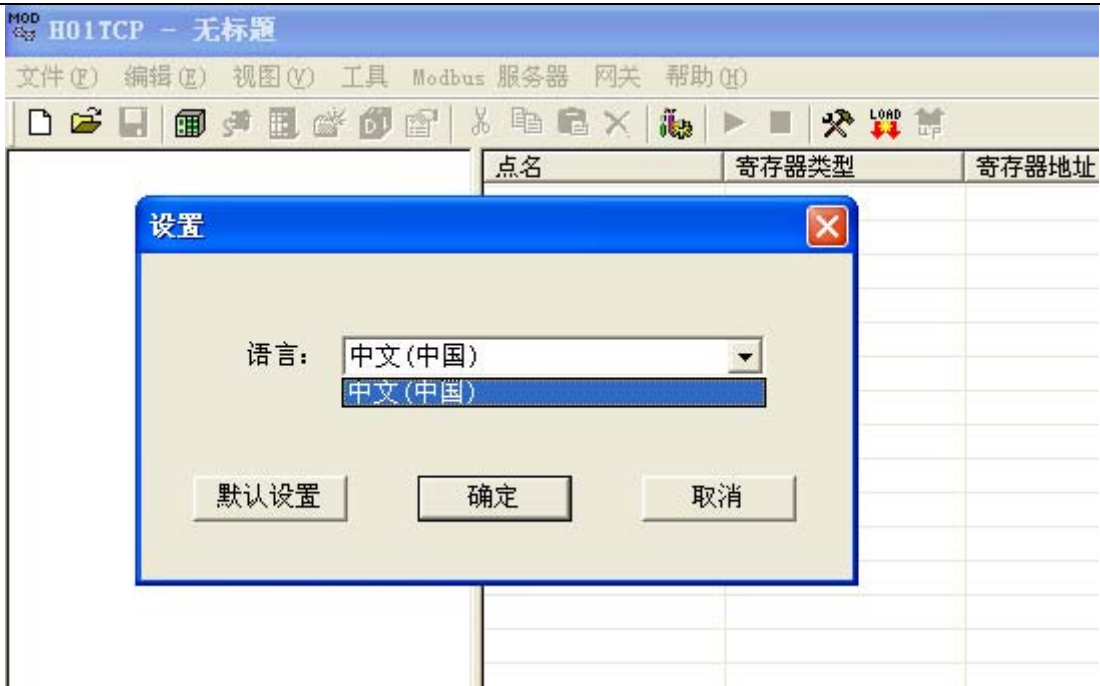


图 3-1-2 选择操作语言

3.2 选择监控模式

在工具栏下“监控模式”下可以选择监控模式，或者在最下面的状态栏双击“监控模式”即可切换模式，如下图 3-2。可以在 PC 上监控硬件网关的通讯状态。

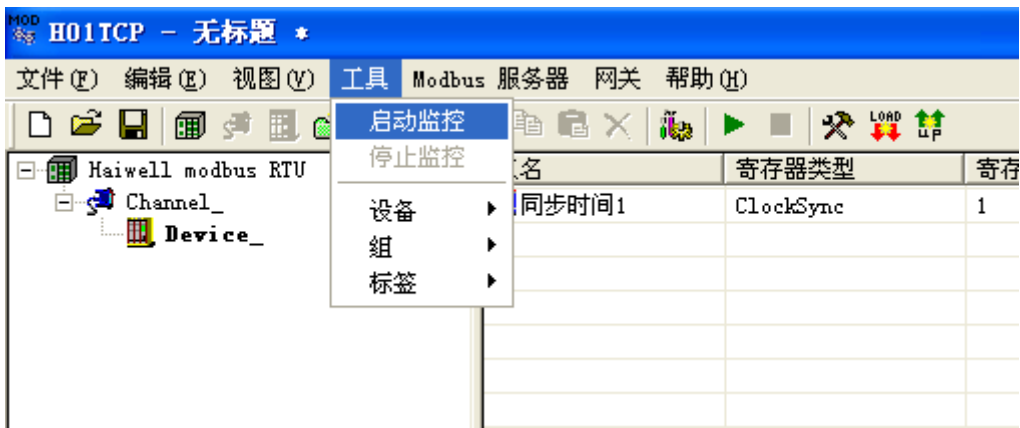


图 3-2 选择启动模式

3.3 添加驱动

点击编辑选择“添加驱动”或者点击工具栏图标，如下图 3-3-1。在这里作为 X 涵盖的众多协议，我们选择 Haiwell ES PLC 协议作为范例。如果需要了解其它协议的配置，打开《CommunicationManual-Ch.pdf》。

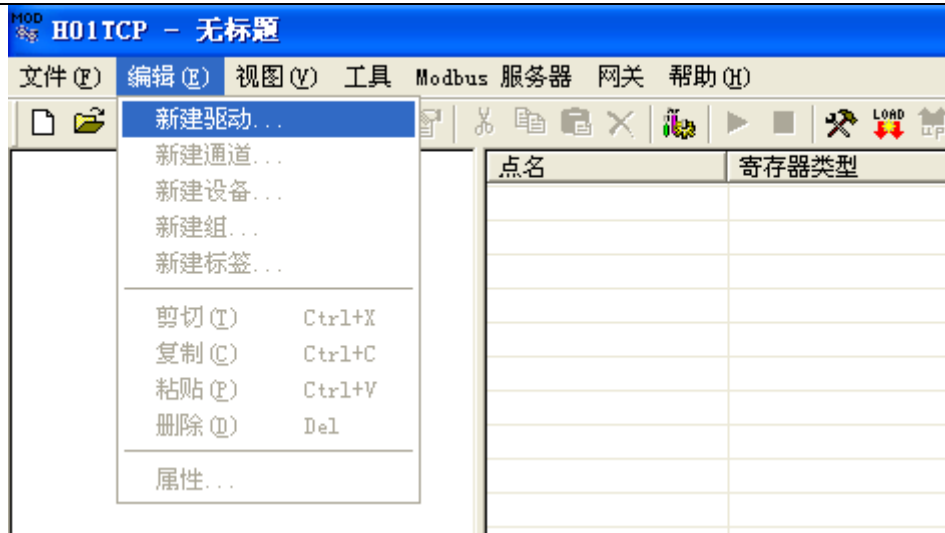


图 3-3-1 选择新建驱动

然后再弹出来的窗口选择驱动进行添加，如下图 3-3-2。

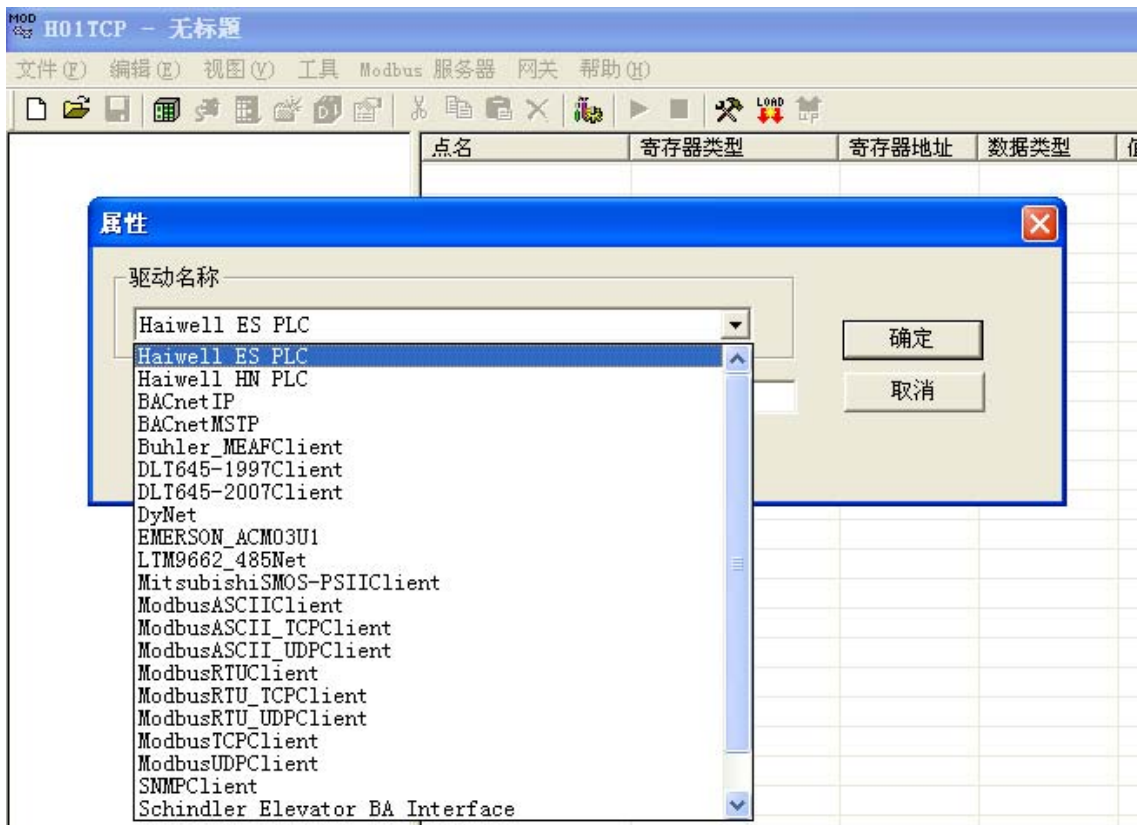


图 3-3-2 选择驱动

编辑所选驱动的属性，如下图 3-3-3。

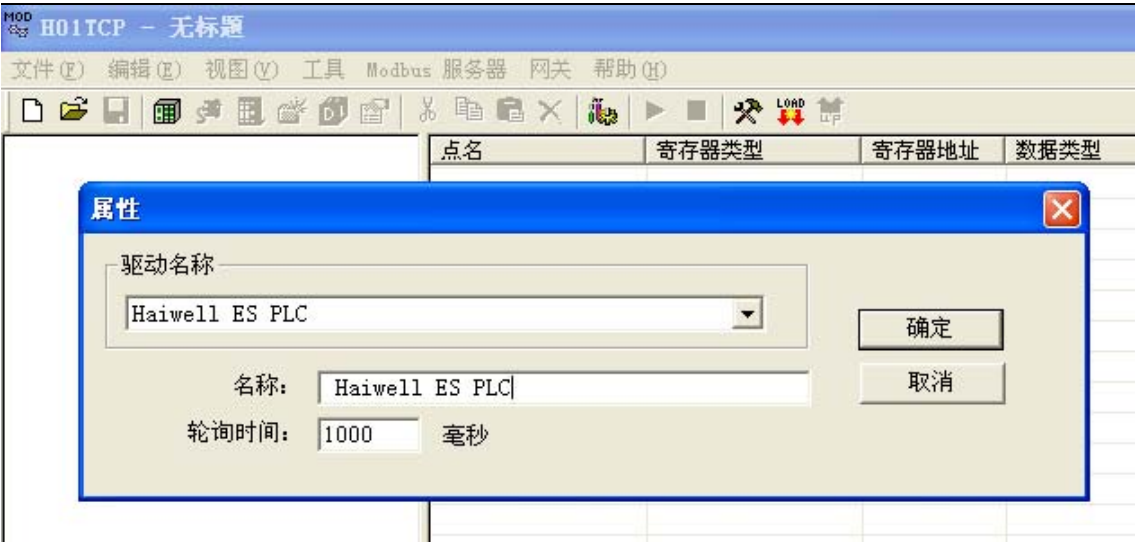
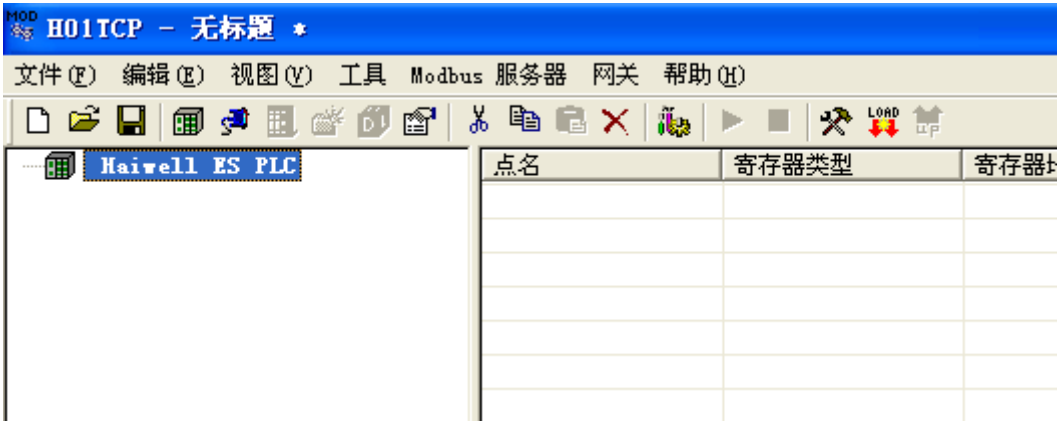


图 3-3-3 驱动属性

在名称项目里输入驱动名称，默认轮询时间是 1000 毫秒，用户可以根据实际情况，更改轮询时间。这里选择 Haiwell ES PLC 协议，添加后如下图 3-3-4。



图

3-3-4 添加驱动完成

3.4 添加通道

选择当前驱动，点击右键选择“新建通道”或者点击工具栏，如下图 3-4-1。

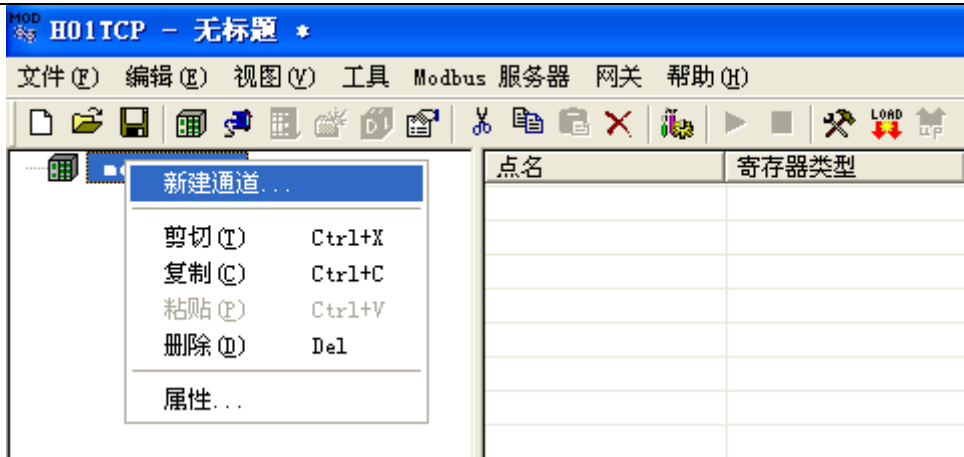


图 3-4-1 选择新建通道

在弹出来的窗口根据驱动通讯协议进行相应设置，通道名称可以任意命名，Haiwell PLC 默认 波特率 19200，数据位 8，停止位 2 无效验。如下图 3-4-2。



图 3-4-2 设置通道参数

通道添加完成后如下图 3-4-4。

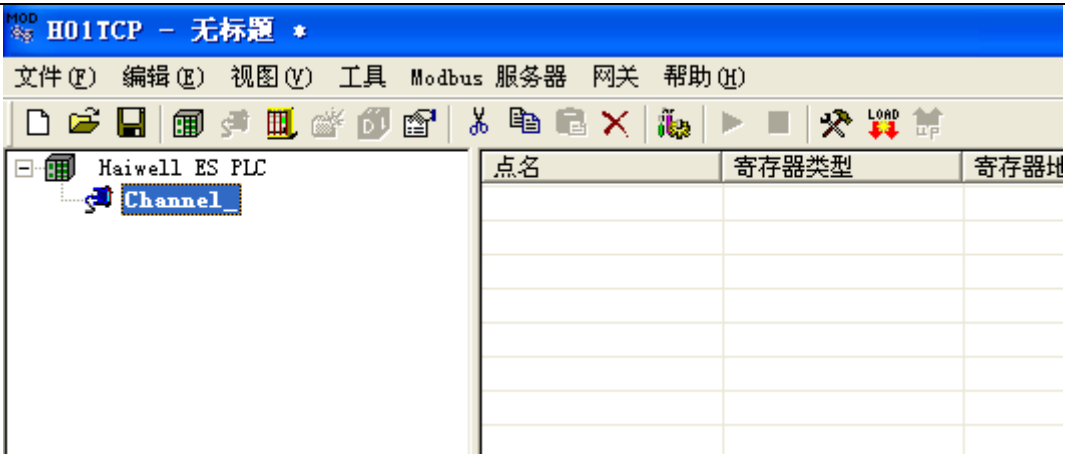


图 3-4-4 添加通道完成

3.5 添加设备

选择当前通道，点击右键选择“新建设备”或者点击工具栏, 如下图 3-5-1。

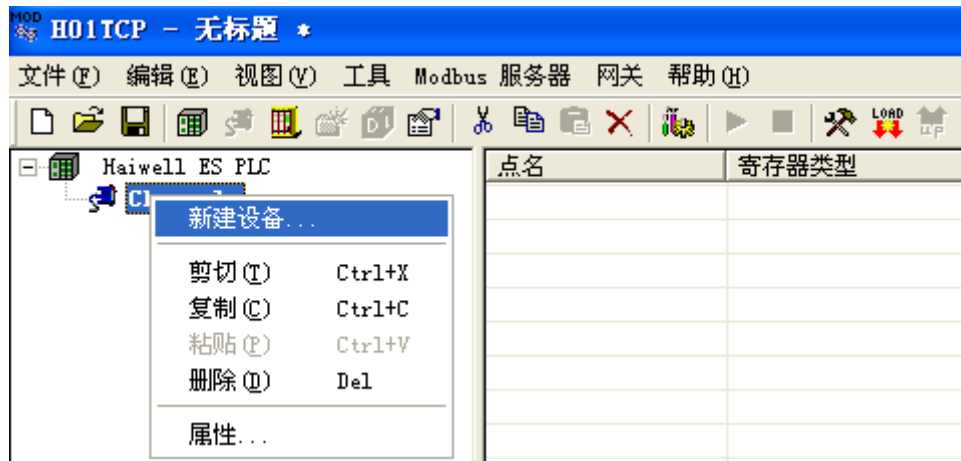


图 3-5-1 选择新建设备

在弹出的对话框中设置设备相关属性，如下图 3-5-2。

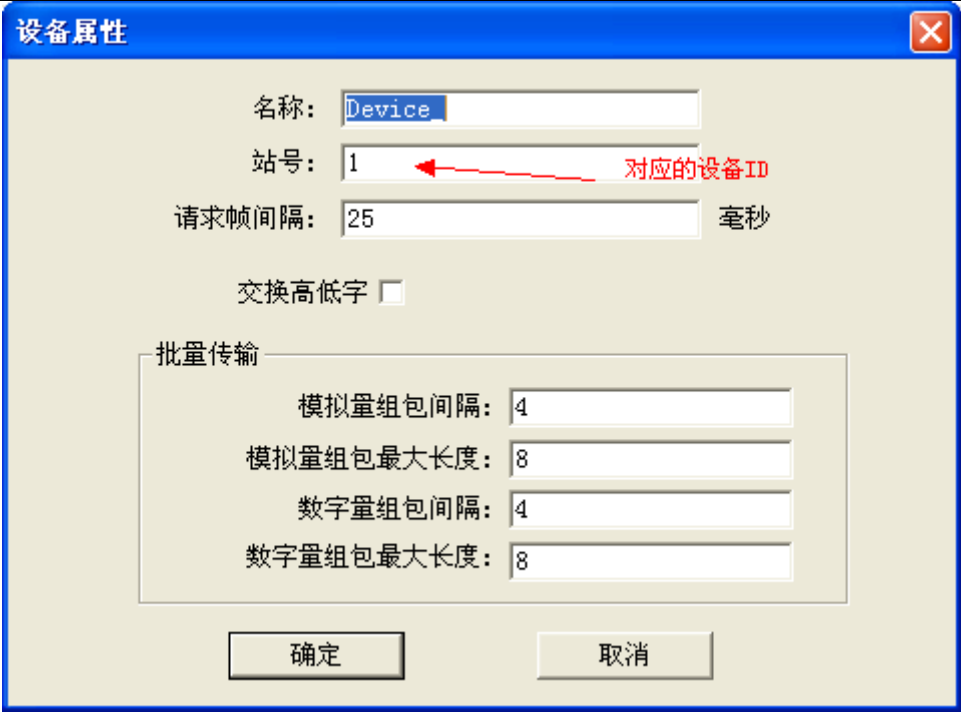


图 3-5-2 设置设备属性

在支持组包的通信协议下，为了提高通信速度，同样寄存器类型，在连续的寄存器地址下，可以实现组包通信。当设备不支持组包通信的情况下，应该把组包的参数都设置为 0。另外，当设备的响应时间比较慢时，可以设置数据帧与帧之间的时间间隔，，默认的帧间隔设置为 25 毫秒。

点击确定，完成添加设备，如下图 3-5-3。

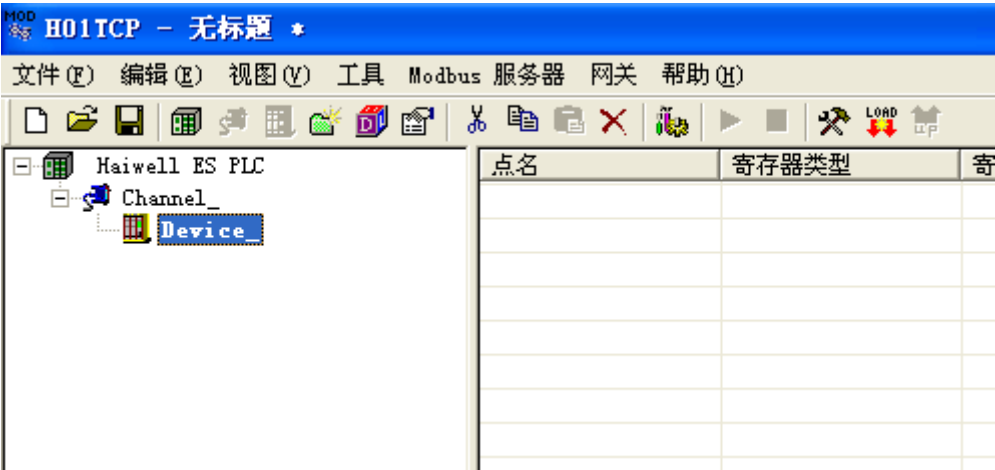


图 3-5-3 添加设备完成

3.6 添加标签

在设备下可以直接新建标签（也可先建立组，再在组中新建标签），选中设备点击右键选择新建标签或者点击工具栏图标，如下图 3-6-1。

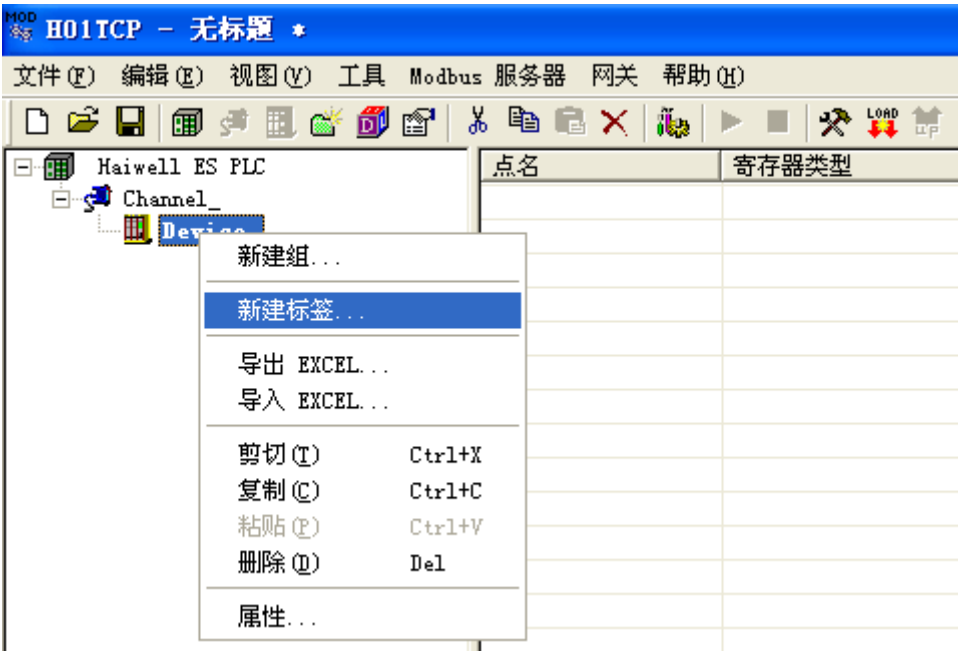


图 3-6-1 选择新建标签

在弹出的对话框中设置采集端和转发端的参数，如下图 3-6-2。

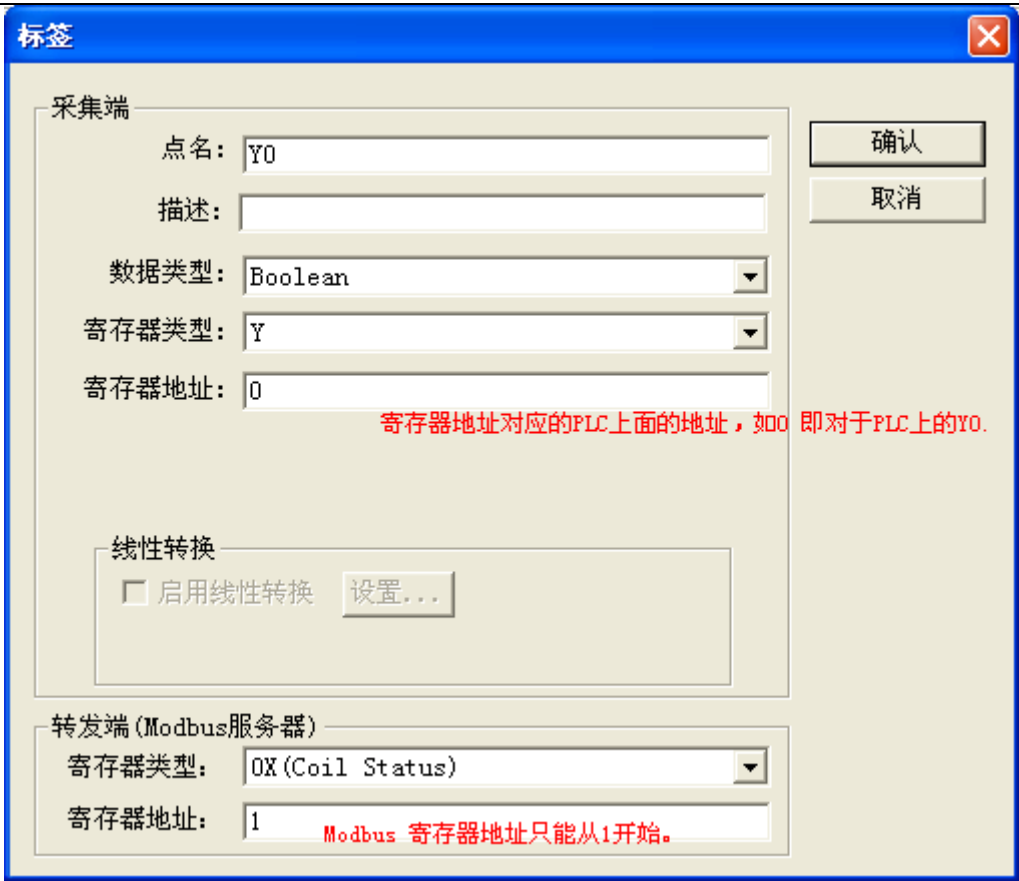


图 3-6-2 设置标签属性

在标签属性里设置采集端的点名、数据类型、寄存器类型、寄存器地址。上图选择的采集端寄存器地址是 0X0000，数据类型是 Boolean 类型。另外当数据类型是 Short、Word、Long 或者 DWord 的情况下，可以按字节的数据位位取值。对于一些特殊数据还可以启用线性转换功能，实现数据的线性放大与缩小。注意 Modbus 服务器寄存器地址的初始地址是从 1 开始的。点击确定完成添加标签，如下图 3-6-3。

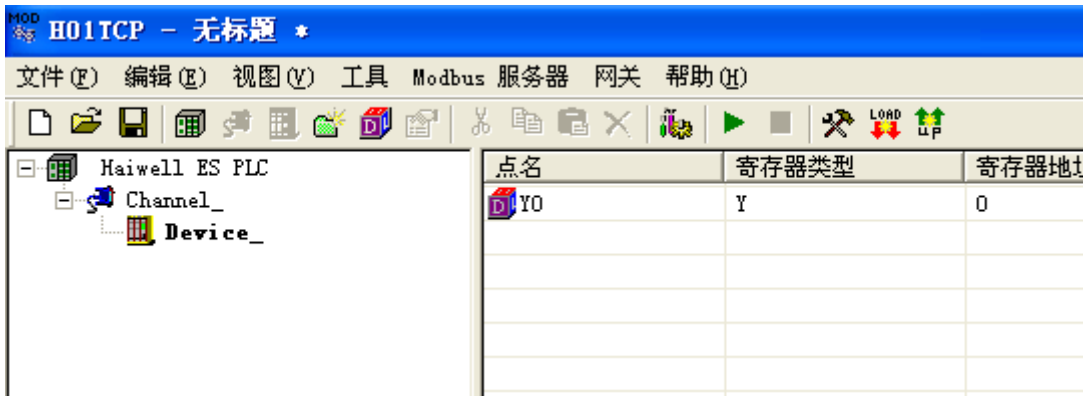


图 3-6-3 添加标签完成

可以继续上面的步骤进行逐个添加，建议使用工具栏的复制粘贴，具体操作如下：

选择要复制的标签，点击工具栏复制按钮，或者右键选择“复制”如下图 3-6-4。

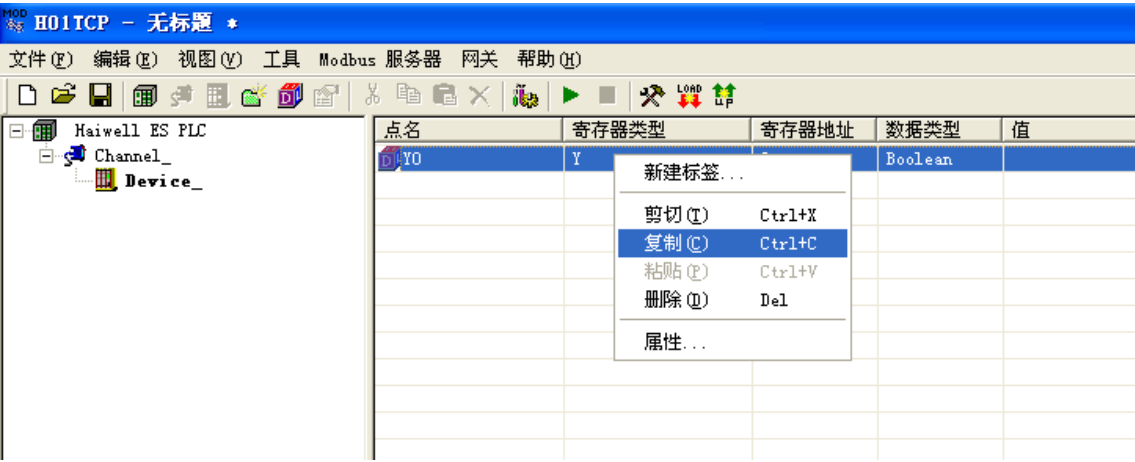


图 3-6-4 复制当前标签

然后按 Ctrl+v 键选择粘贴，如下图 3-6-5。

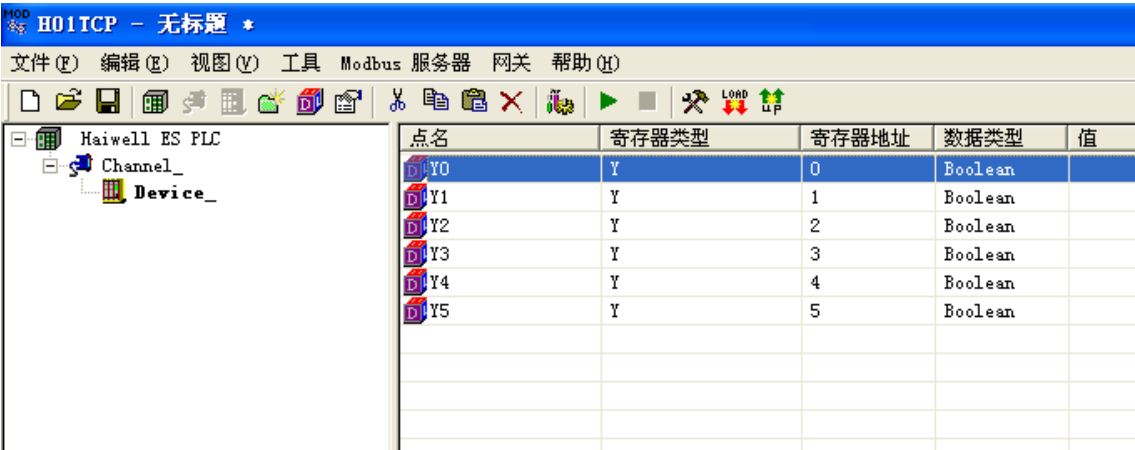


图 3-6-5 粘贴标签

就会自动添加新的标签，新的标签的部分参数（如 Modbus 寄存器的地址）会相应的自动生成，需要根据现场情况进行设置，如下图 3-6-6。

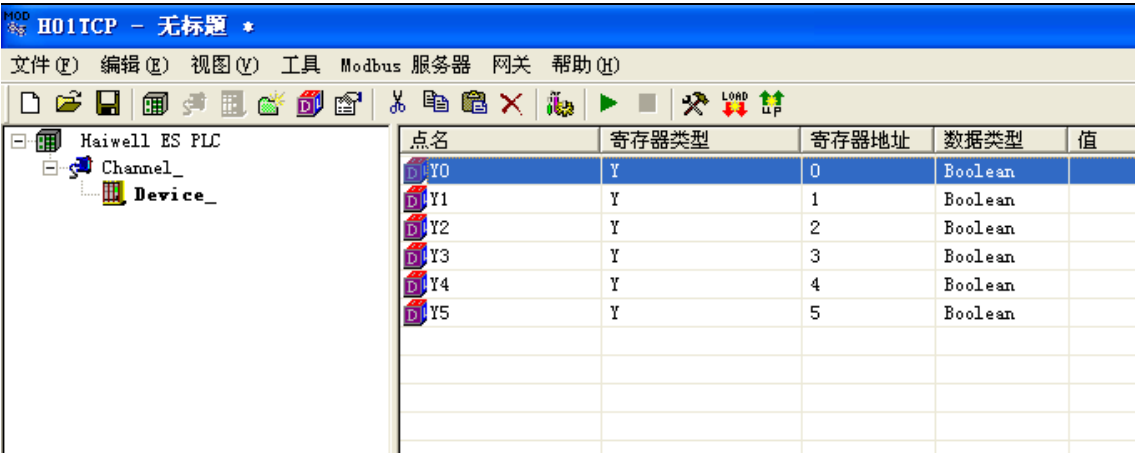


图 3-6-6 复制标签完成

另外也可以在 EXCEL 表中作编辑，然后通过导入导出功能编辑工程。

在新建的设备下新建标签，如下图 3-6-7。

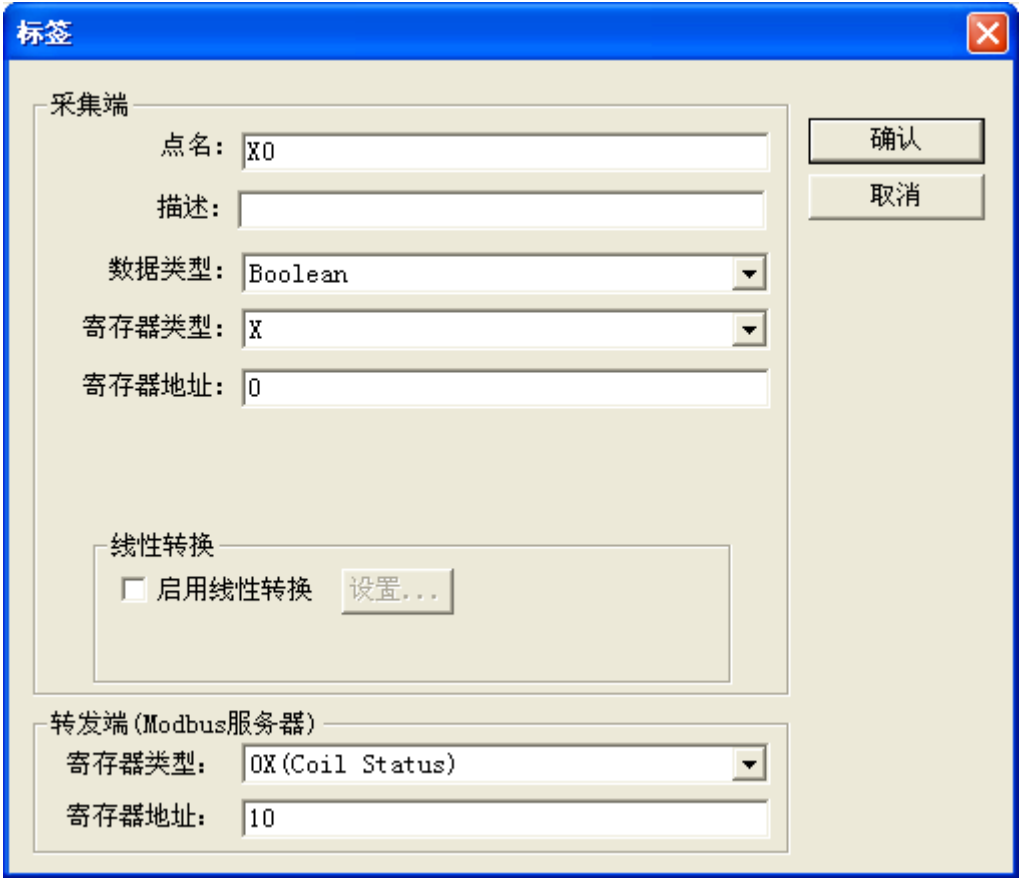


图 3-6-7 新建标签

这里以 Boolean 数据类型为例，新建好一个标签，如下图 3-6-8。

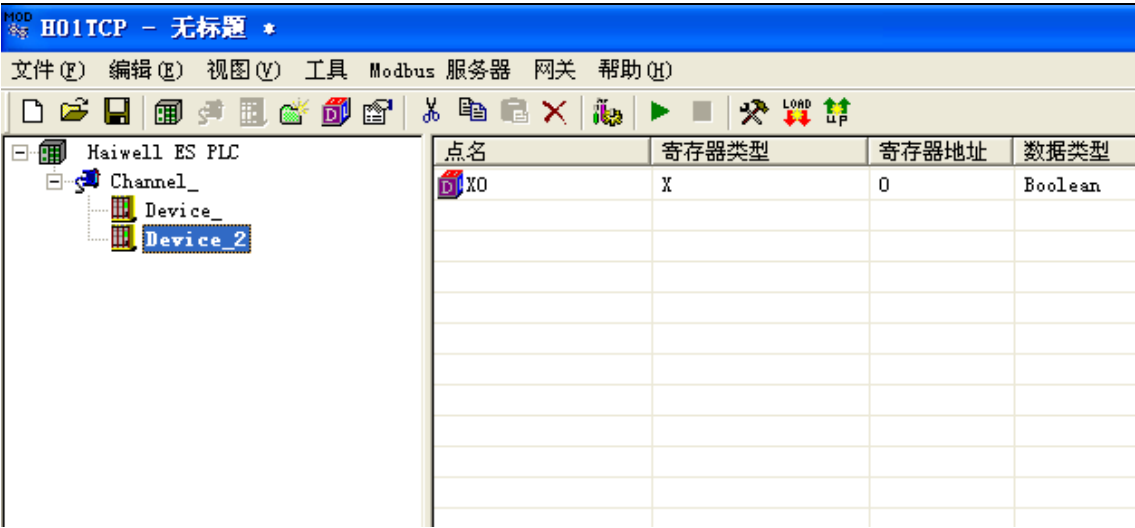


图 3-6-8 新建标签完成

选择当前设备，右键选择“导出 EXCEL”，如下图 3-6-9。

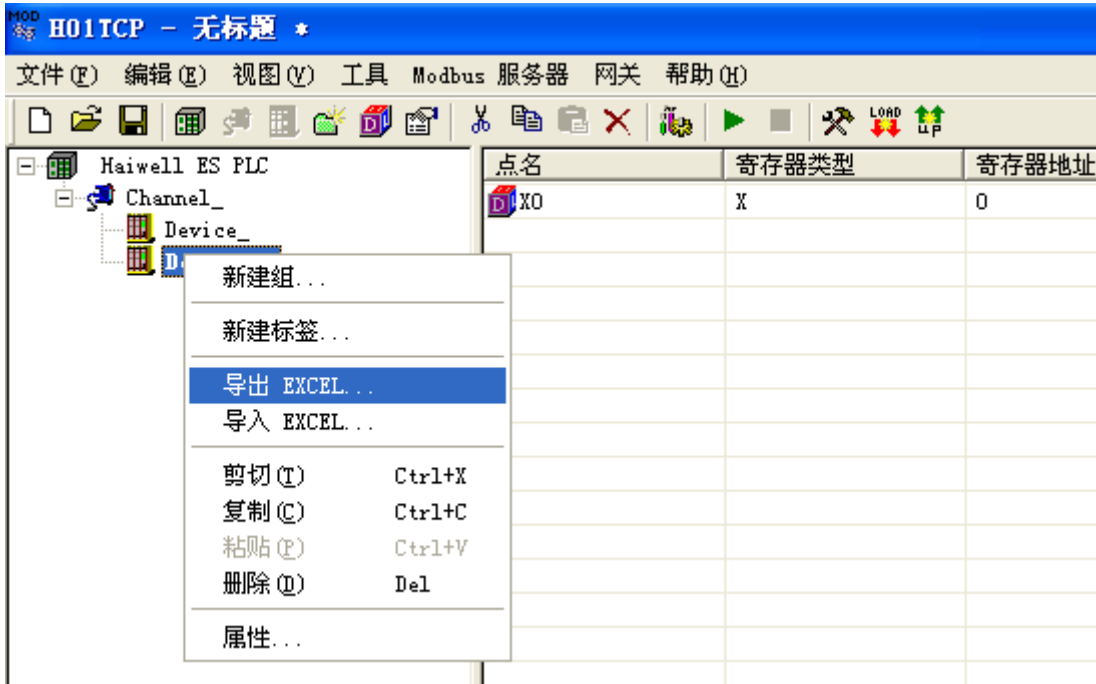


图 3-6-9 选择导出 EXCEL 表

保存好 EXCEL 表后，打开 EXCEL 表，如下图 3-6-10。

A	B	C	D	E	F	G	H	I	J	K
ID	Name	RegType	RegAddress	DataType	ModbusType	ModbusAddress	Description	Mask	Bit	Scale
1	X0	X	0	Boolean	0	10		0	0	0
2										
3										
4										
5										

图 3-6-10 打开 EXCEL 表

然后在 EXCEL 中进行批量编辑，如下图 3-6-11。

Microsoft Excel - Device_2.csv															
文件(F) 编辑(E) 视图(V) 插入(I) 格式(O) 工具(T) 数据(D) 窗口(W) 帮助(H)															
N8															
A	B	C	D	E	F	G	H	I	J	K	L	M	N		
1	ID	Name	RegType	RegAddress	DataType	ModbusType	ModbusAddress	Description	Mask	Bit	Scale	MinRawRate	MaxRawRate	MinEngRate	MaxEngRate
2	1	X0	X	0	Boolean	0	10		0	0	0	0	10	0	0
3	2	x1	x	1	Boolean	0	11		0	0	0	0	10	0	0
4															

图 3-6-11 编辑 EXCEL 表

编辑完成后保存 EXCEL 表，选择当前设备右键选择“导入 EXCEL 表”，如下图所示 3-6-12。

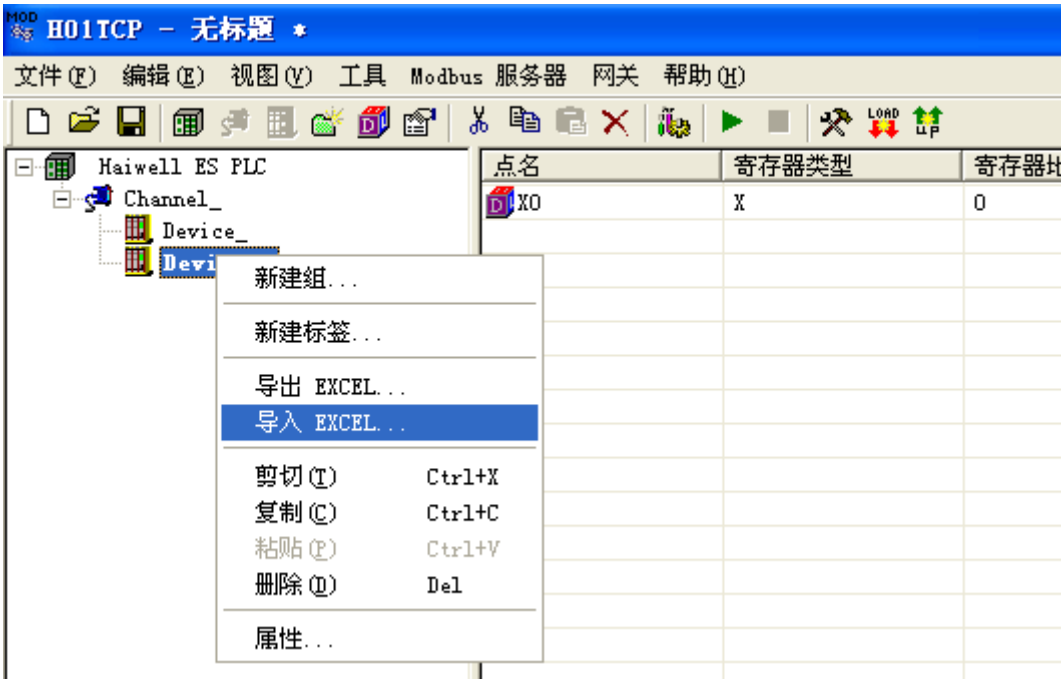


图 3-6-12 选择导入 EXCEL 表

导入 EXCEL 表完成如下图所示 3-6-13。

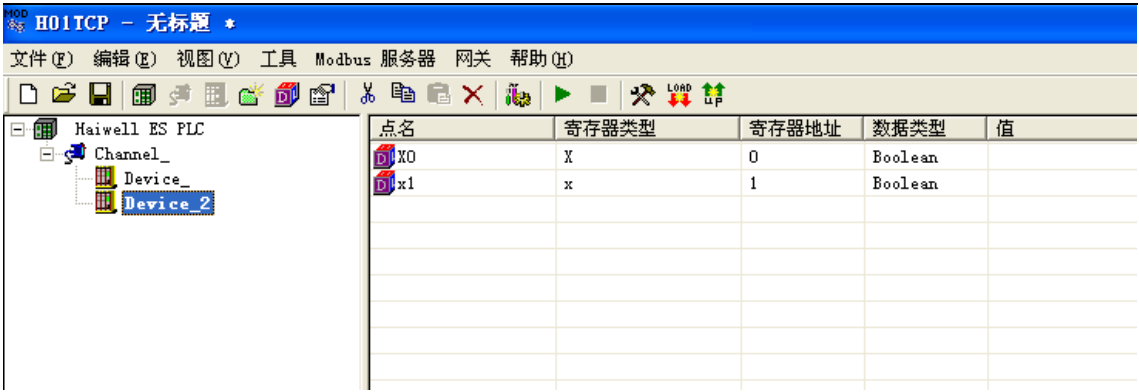


图 3-6-13 导入 EXCEL 表完成

3.7 Modbus 服务器（从站）设置

Modbus 服务器（从站）包括 ModbusRTU 和 ModbusTCP，Modbus 服务器出厂默认是支持 Modbus TCP 协议的，且端口号默认为 502，如若用户需要更改，可点击 Modbus 服务器，选择“Modbus TCP 设置”如下图 3-7-1。

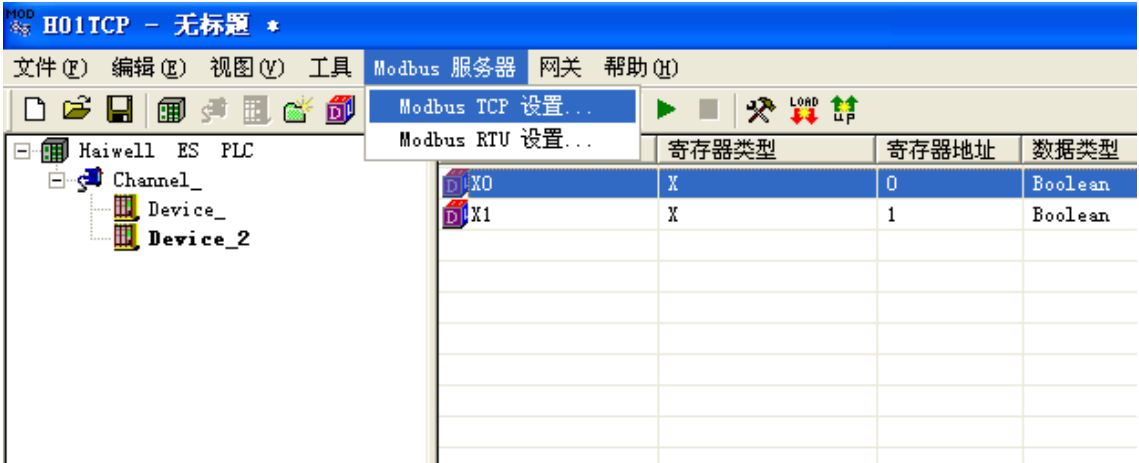
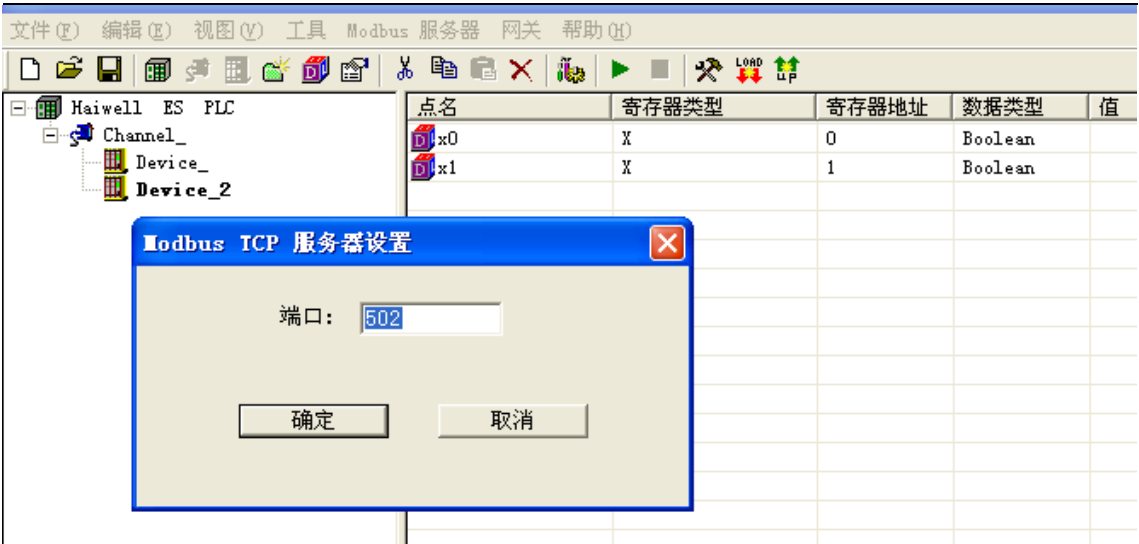


图 3-7-1 选择 ModbusTCP 设置

在弹出来的对话框中设置 ModbusTCP 服务器参数，如下图 3-7-2。



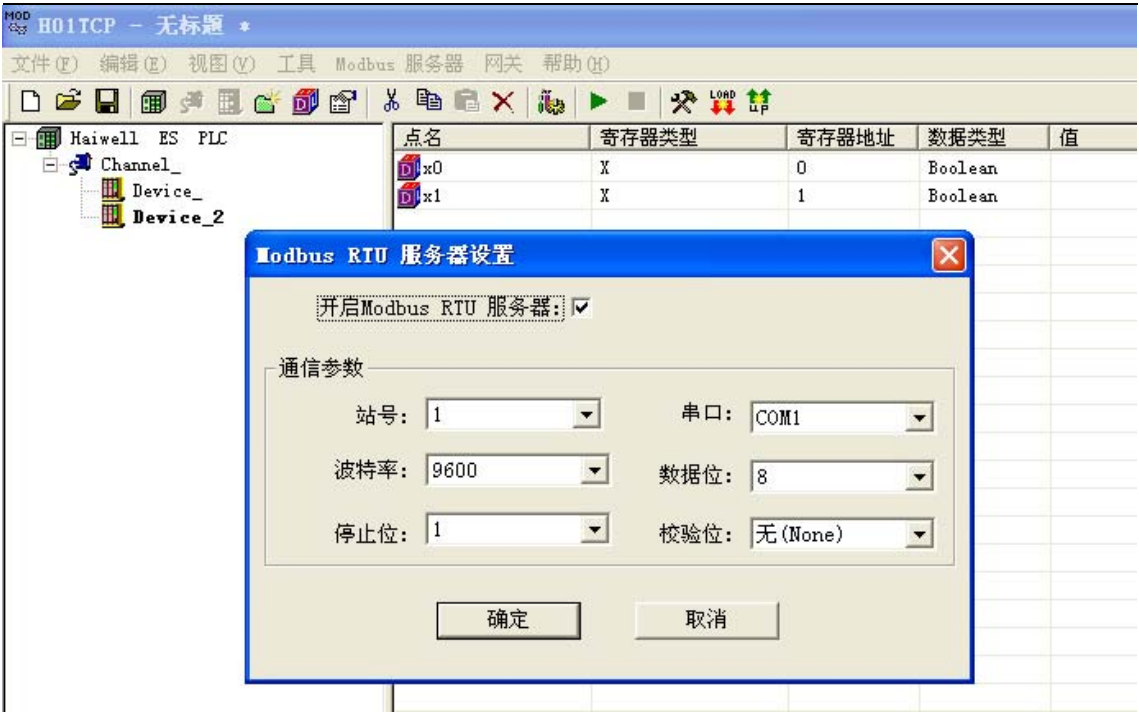
3-7-2 ModbusTCP 参数设置

若果设置 ModbusRTU，可点击“Modbus 服务器”，选择“ModbusRTU 设置”，如下图 3-7-3。



图 3-7-3 选择 ModbusRTU 设置

在弹出来的对话框中设置 ModbusRTU 服务器参数，如下图 3-7-4。



3-7-4 RTU 方式参数设定

注意“开启 Modbus RTU 服务器”后面的方框须勾上才会生效。

3.8 H01TCP 网关运行时

工程配置完成之后，点击菜单栏“工具”选择“启动监控”或者点击工具栏图标，如下图 3-8-1。

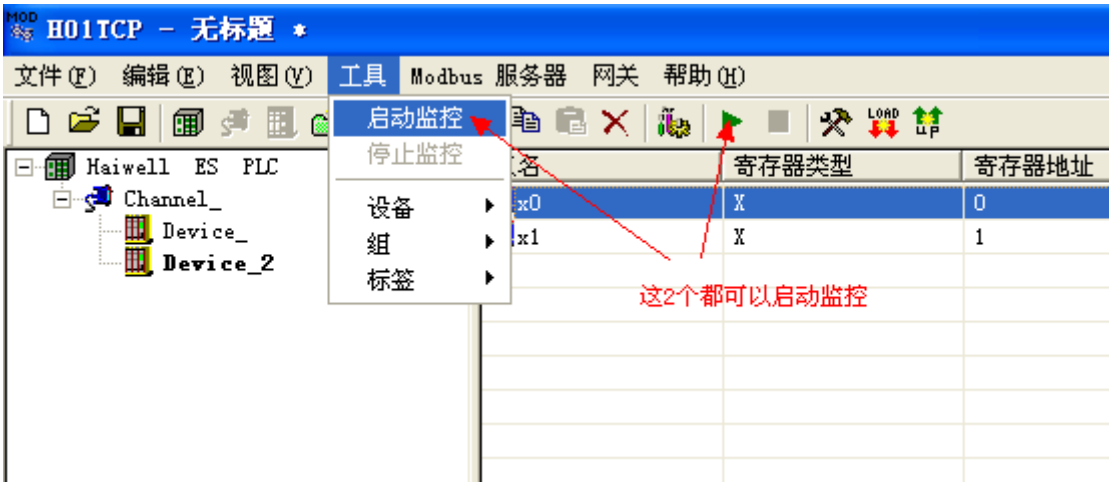


图 3-8-1 选择启动监控

3.9 上传工程

注意：上传工程功能在[网关监控模式](#)有效。

配置好工程后，在 PC 上测试没问题后，可将工程上传到下位机网关中，在网关模式下，点击菜单栏“网关”选择上传或者点击工具栏，如下图 3-9-1。

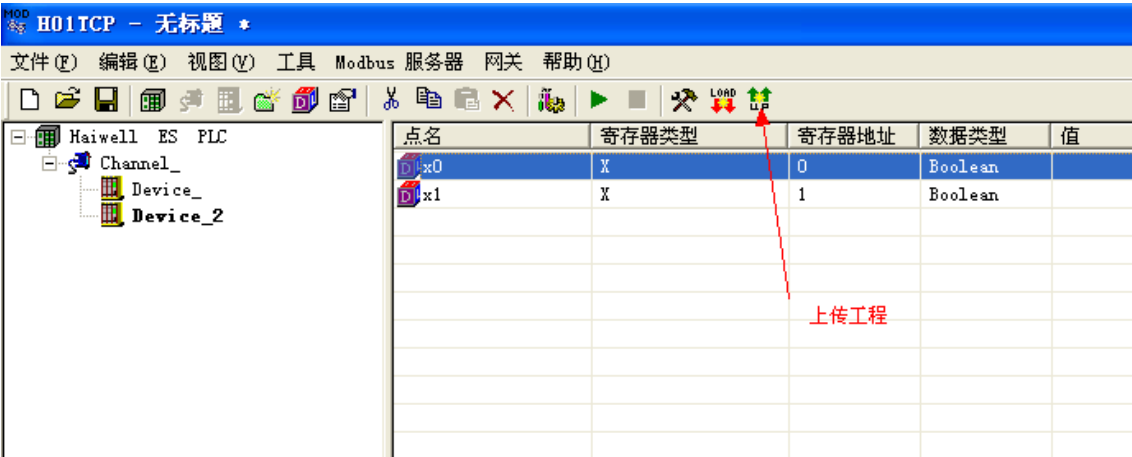


图 3-9-1 选择上传工程

在弹出来的对话框中输入网关 IP 地址，点击“上传”，如下图 3-9-2。

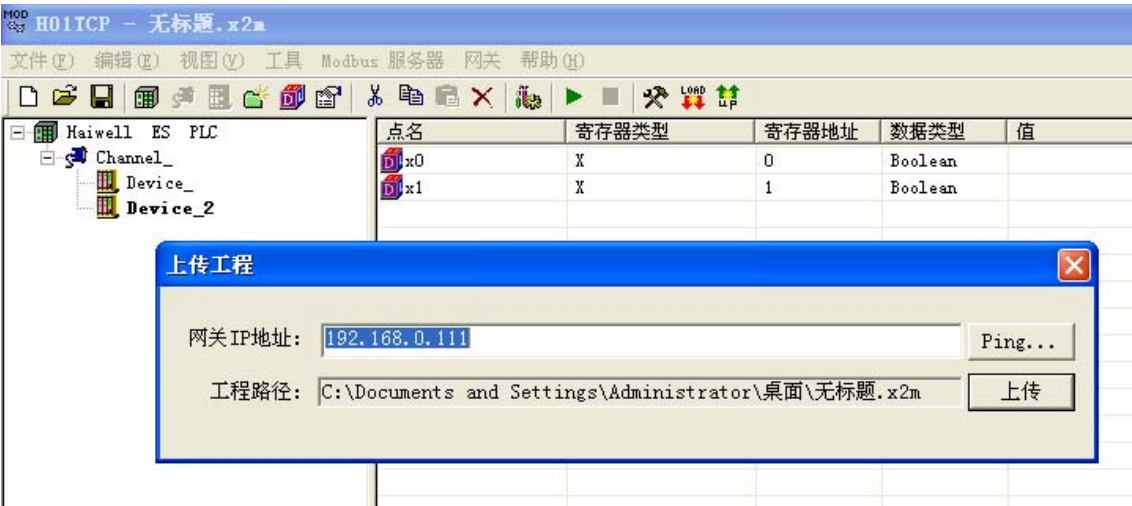


图 3-9-2 上传工程参数设置

上传成功后，会弹出对话框提示上传成功，如下图 3-9-3。

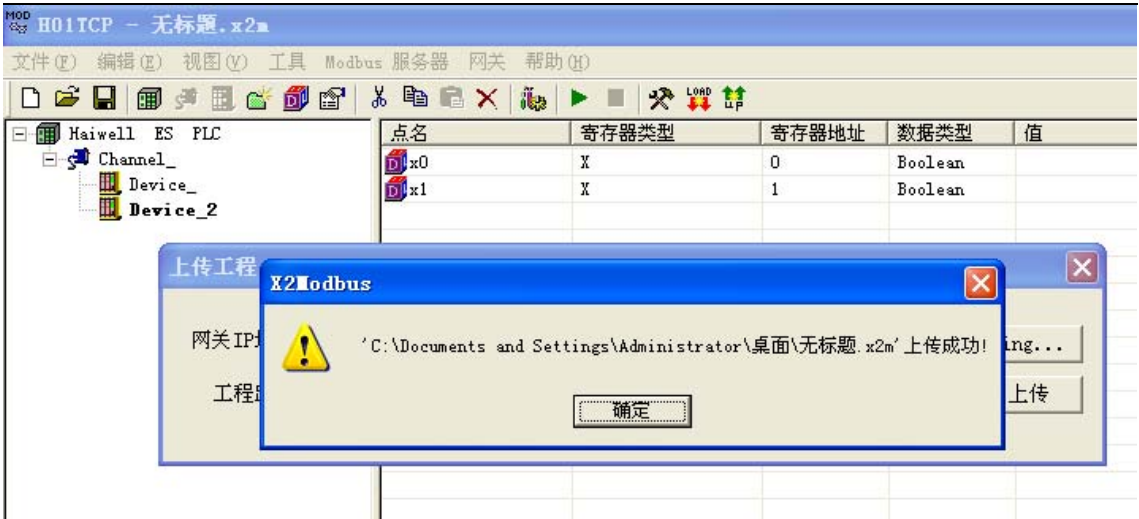


图 3-9-3 上传工程成功

注意网关的 IP 地址必须正确无误，网关出厂默认 IP 地址是 192.168.1.88，PC 上的 IP 地址要设置到同一个网段，Ping 通以后即可上传。

3.10 网关参数设置

在[网关监控模式](#)下，点击菜单栏“网关”，选择“参数设置”，在弹出的对话框里设置硬件网关配置参数，如下图 3-10-1。



图 3-10-1 选择网关参数设置

网口设置:可以更改硬件网关的 IP 地址，网关出厂默认 IP 地址为 192.168.1.88，子网掩码默认为 255.255.255.0，默认网关 192.168.1.1，设置完成点击确定即可。

“Ping”功能测试 ping 当前 IP 地址是否成功。“登陆网页功能”可以登陆到网关所在的 WEB 服务器上，如下图 3-10-2。



图 3-10-2 网口设置

注意：网关的出厂默认 IP 地址是 192.168.1.88，用户如果是第一次改 IP 地址，用户只需一根网线（交叉或直连都可以）和网关对接。需先将 PC 设置成和网关为同一个网段，然后再在当前 IP 地址文本框中输入网关当前 IP 地址 192.168.1.88，然后点击应用，使得网关参数设置操作对应当前 IP 地址的网关，然后再设置新的 IP 地址。

串口设置：设置硬件网关 COM 口通讯方式，其中 COM1 和 COM2 口支持 RS232，和 RS485 两种方式，如下图 3-10-3。



图 3-10-3 串口设置

系统时间：能够读取硬件网关当前系统时间，如果时间与正常时间不匹配，可以点击“写入”功能刷新硬件网关系统时间，自动更新周期是指将硬件网关的系统时间更新到底层末端设备上的周期间隔，如下图 3-10-4。

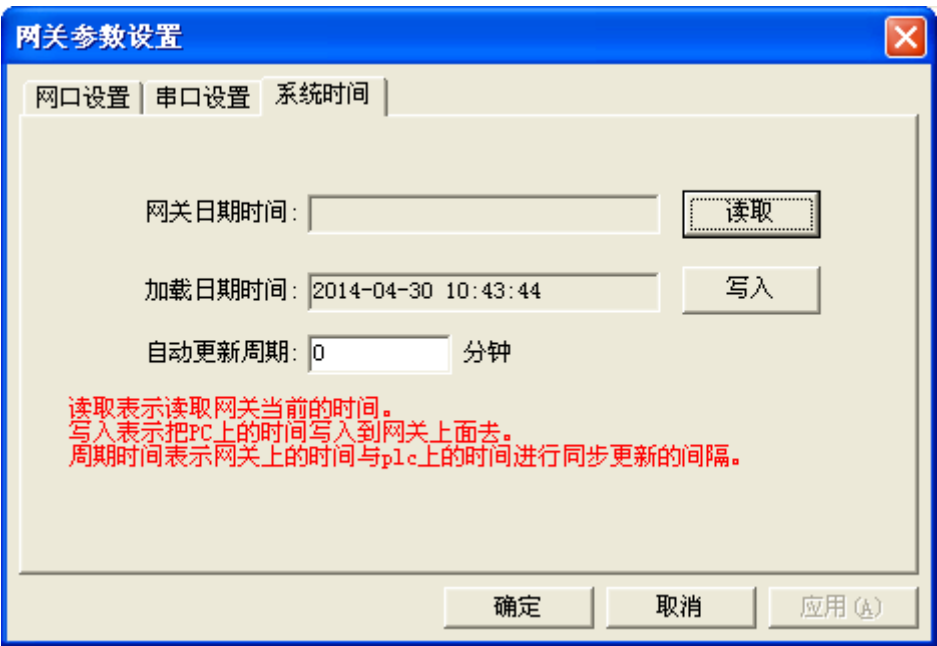


图 3-10-4 系统时间设置

3.11 下载工程

注意：下载工程功能在[网关监控模式](#)有效。

下载工程是指从硬件网关里把上一次配置的工程下载到 PC 上，在 PC 上可以编辑工程和查看实时数据，方便用户调试，点击“网关”选择“下载工程”，如下图 3-11-1。

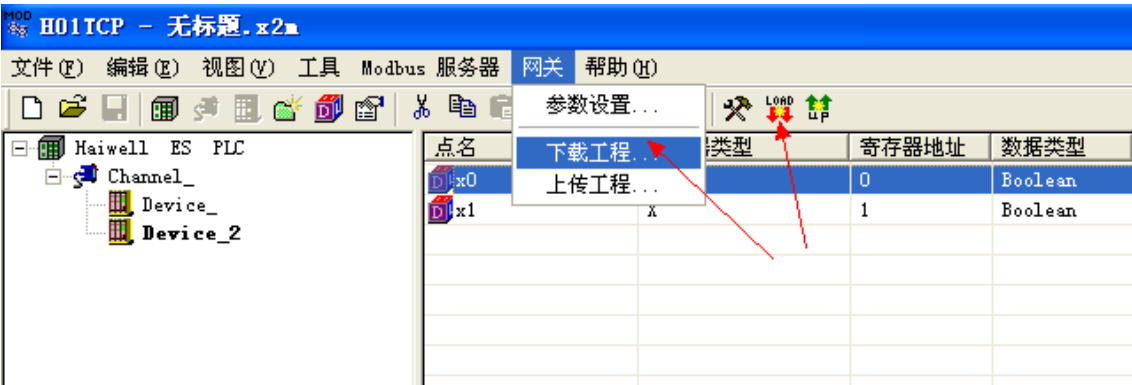


图 3-11-1 选择下载工程

在弹出的对话框中输入网关 IP 地址，可从网关中下载当前工程，如下图 3-11-2。

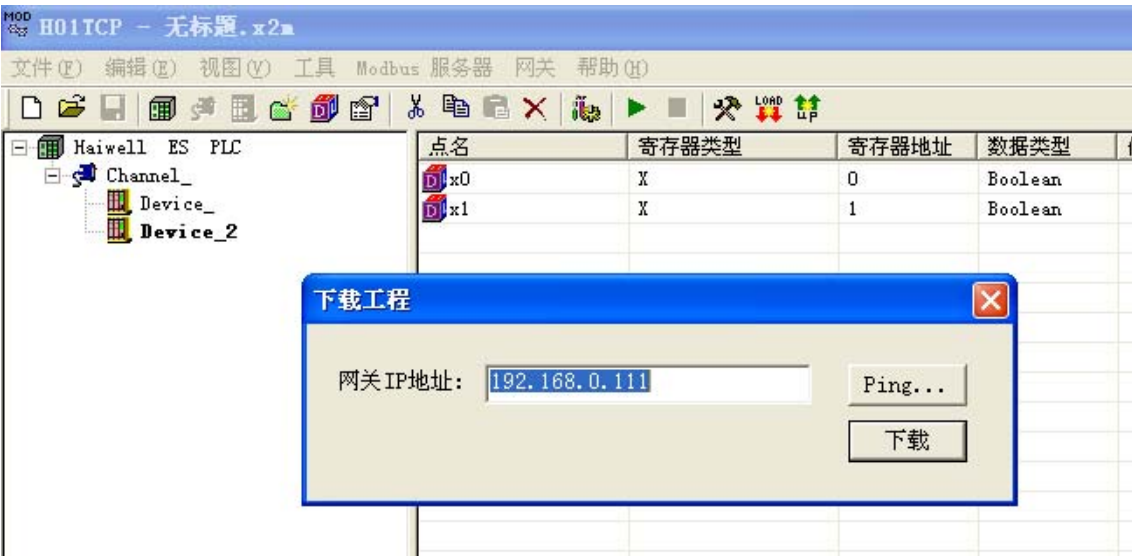


图 3-11-2 下载工程

用户也可以通过登录到网关的 [WEB 服务器](#)，下载工程。

3.12 软件授权

对于授权，硬件网关在出厂时已经授权，对于使用上位机纯软件网关的用户来说，授权方式主要有 USB 硬授权和注册码软授权两种方式。

USB 硬授权是本公司给用户一个即插即用的 U 盘，用户可以在电脑上插入 U 盘，即可使用本产品，硬授权不受电脑限制，即可以在多台电脑上使用本产品。

软授权是通过用户的机器码产生一个唯一对应的注册码，此注册码只能在一台电脑上使用，注册码和机器码是相对应的。

4 WEB 服务器

网关自带 WEB 服务器，默认端口固定为 80。用户可以通过浏览器就可以登录到 WEB 服务器，在网页里可以修改硬件网关的 IP 地址，串口的通信方式，查看实时数据，下载 H01TCP 上位机配置软件及工程文件等。

注意：网关的出厂默认 IP 地址是 192.168.1.88，用户如果是第一次改 IP 地址，用户只需一根网线（交叉或直连都可以）和网关直连。需先将 PC 设置成和网关为同一个网段，然后再在浏览器中输入 192.168.1.88 完成网关 IP 地址的修改。

4.1 网页登陆

在浏览器中输入下位机网关的 IP 地址，如下图 4-1-1。

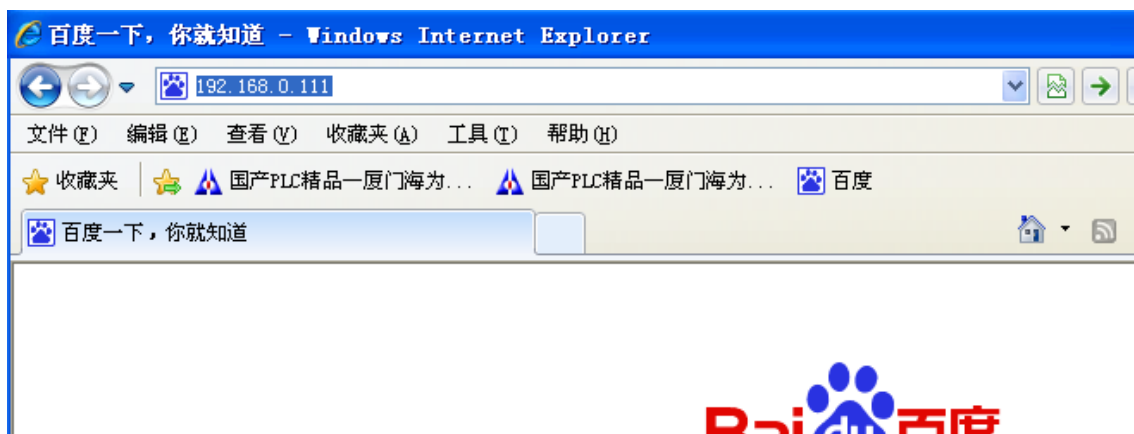


图 4-1-1 网页输入网关 IP 地址

在弹出来的窗口中输入用户名和密码，如下图 4-1-2。

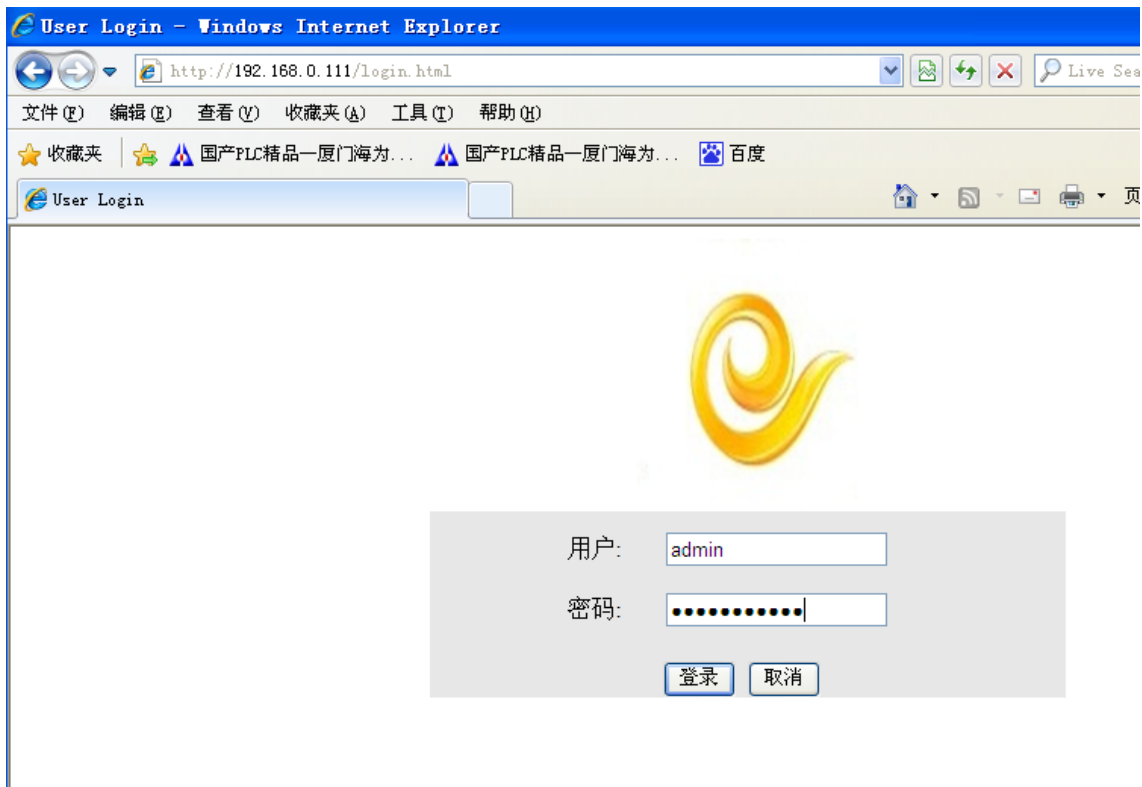


图 4-1-2 用户登录

注意出厂默认用户名为“**admin**”，密码为“**admin123456**”，用户在登陆成功之后在用户管理里可以添加个人账户。

4.2 下载文件

下载文件：下载和网关配套使用的上位机配置软件，当前的工程文件以及帮助文档，如下图 4-2-1。



图 4-2-1 下载文件

4.3 用户管理

用户管理：用户可以通过添加、修改、删除等方式管理自己的登陆账号，如下图所示 4-3-1。



图 4-3-1 用户管理

4.4 网口设置

网口设置：设置下位机硬件网关网口 IP 地址，如下图所示 4-4-1。



图 4-4-1 网口设置

4.5 串口设置

串口设置：设置 COM1 和 COM2 口通讯方式，分别是 RS232、RS422 和 RS485 三种方式，如下图所示 4-5-1。



图 4-5-1 串口设置

串口设置视网关串口实际接线方式而定，只有设置正确时，通信才能成功。

4.7 通讯状态

通讯状态：除此之外还可以在网页中查看实时动态的数据帧，方便用户直观的了解实时通讯状态，通过对数据帧的分析，可以找出故障的原因，给用户带来了很大的方便，如下图 4-7-1。

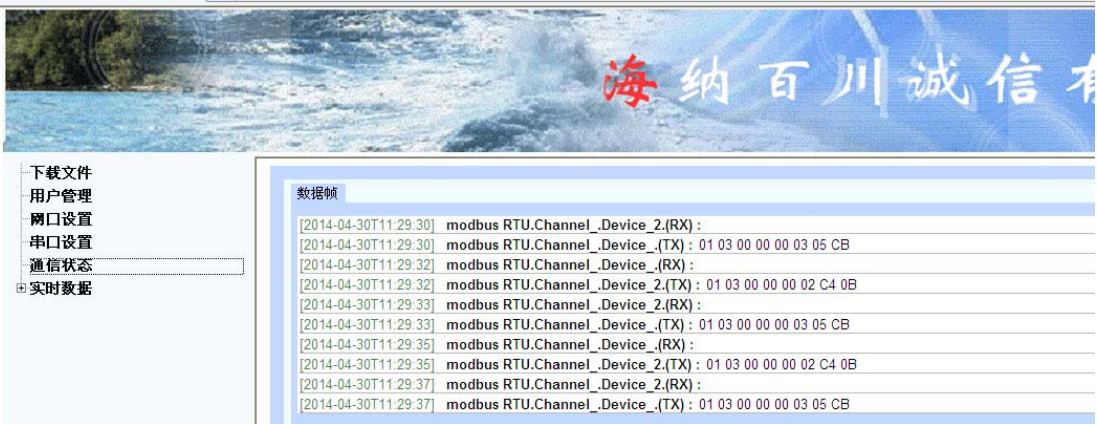


图 4-7-1 查看通讯状态

4.8 实时数据

实时数据：在网页上可以查看网关采集到的实时数据信息，如下图 4-8-1。



图 4-8-1 查看实时数据

也可以在网页上进行写值操作，如下图 4-8-2。



图 4-8-2 写值操作

5 Modbus 客户端（主站）访问

转出的 Modbus 协议可支持 ModbusTCP 和 ModbusRTU 协议，在这里我们用 Modbus Poll 软件通过 TCP 方式来访问我们的 H01TCP。

Modbus Poll(客户端软件)通过 TCP/IP 的方式访问我们的 H01TCP 从站，从而访问现场 ModbusRTU 设备，整个过程就是 ModbusRTU 协议通过 H01TCP 转成 ModbusIP/TCP 协议。



图 6-1-2 JS 脚本编辑器编辑界面

用户可以利用“内部变量”功能新建一个变量和相关函数配合使用，用来接收或存储变量数据，比如取 Tag 的质量戳，映射到一个新的内部变量。在脚本编辑器内点击“内部变量”，在弹出来的窗口中右键选择“新建”，如下图 6-1-3。



图 6-1-3 选择新建内部变量

注意上图中已经存在 2 个内部变量，此变量表示 **Device_1** 这个设备的通讯状态，如果该设备下所有的点通讯正常，则这个内部变量的值为 1，如果该设备下某些点通讯不正常，则这个内部变量的值为 0。如果当前有 **n** 个设备，则会自动产生 **n** 个内部变量，用户在配置工程的时候需注意避免 1x 类型的寄存器地址重复。

在弹出的窗口中，设置变量相关属性，如下图 6-1-4。如果要快速查看内部变量的当前值，可以点击菜单栏“视图”选项，选择“内部变量”即可在弹出来的窗口中查看实时数据。

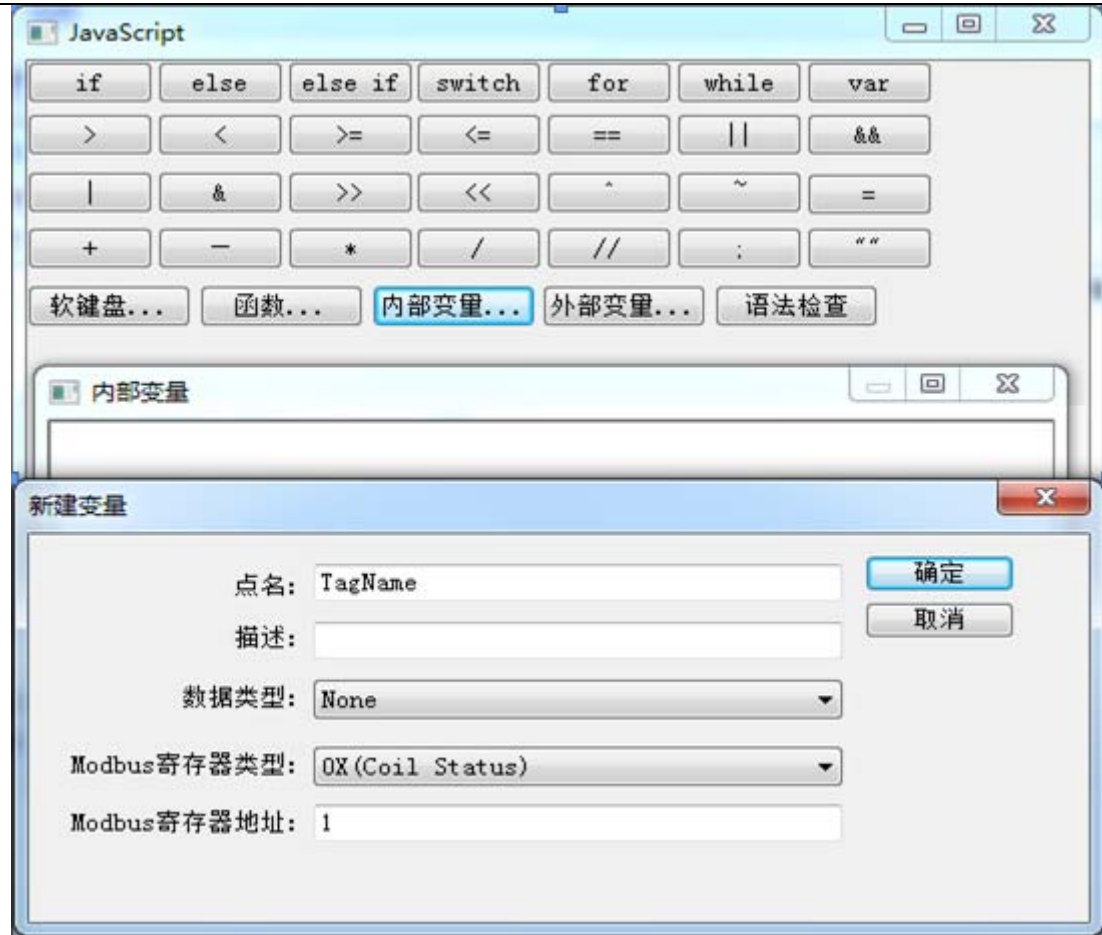


图 6-1-4 新建内部变量

另外，用户也可以根据外部变量的数据变化采用相关函数进行逻辑控制，譬如根据某一个外部变量的数值变化范围设置不同的逻辑控制。在脚本编辑器内点击“外部变量”，在展开的变量表中双击某一个变量即可使用，如下图 6-1-5。

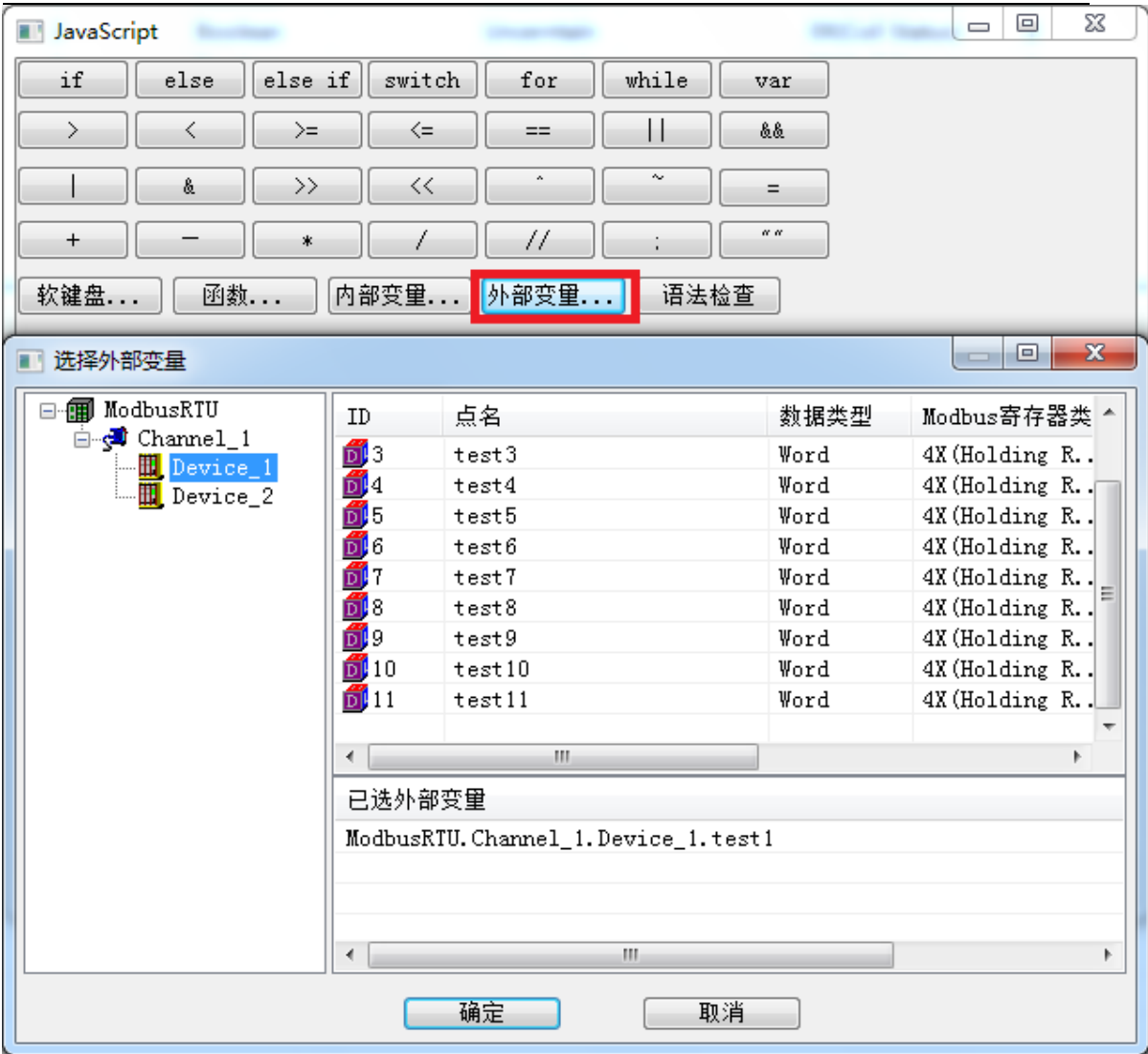


图 6-1-5 选择外部变量

完成编程后，用户可以使用“语法检查”功能检查是否存在语法错误问题，如果程序有问题，则会给出提示。

6.2 函数说明

JS 脚本编辑器内置的常用函数说明如下：

1. 定时器函数

setTimeout(); 是一个定时器函数，用来在运行后指定的毫秒数后触发一次的函数，只会使用一次；

var t = setTimeout("javascript 语句", 毫秒)

`setTimeout()` 方法会返回某个值。在上面的语句中，值被储存在名为 `t` 的变量中。

假如你希望取消这个 `setTimeout()`，你可以使用这个变量名来指定它。

`setTimeout()` 的第一个参数是含有 [JavaScript](#) 语句的字符串。这个语句可能诸如 `"alert('5 seconds!')"`，或者对函数的调用，诸如 `alertMsg()`。

第二个参数指示从当前起多少毫秒后执行第一个参数。

关闭 `setTimeout` 打开的定时器函数为 `clearTimeout()`;

语法:

`clearTimeout(id_of_settimeout)`

`id_of_settimeout` 是由 `setTimeout()` 返回的 ID 值。该值标识要取消的延迟执行代码块。

`setInterval()` 是一个定时器函数，用来在运行后指定的毫秒数后反复触发的函数，循环执行。它的语法是 `setInterval("函数名, 或者需要的操作", 1000)`;

`setInterval()` 方法会不停地调用函数，直到 `clearInterval()` 被调用或程序被关闭。由 `setInterval()` 返回的 ID 值可用作 `clearInterval()` 方法的参数。

关闭 `setInterval` 打开的定时器函数为 `clearInterval()`，语法和 `clearTimeout` 一致。

2.Tag 操作函数

`ReadFromTag` 是从 Tag 里取值函数。

语法:

`var szValue = ReadFromTag('szTagName');`

WriteToTag 是写值到 Tag 函数，用于对外部变量的 Tag 赋值，会执行对设备一个写操作。

语法:

```
var szValue = 100;
```

WriteToTag ('szTagName', szValue.toString());其中'szTagName'从外部变量中选取。

GetTagQuality 是从 Tag 里取质量戳。返回的结果符合 OPC 规范。即 Good 返回 192, Bad 返回 0, Uncertainn 为 64, 表示值未赋值, 尚未被更新。

var varQuality = GetTagQuality('szTagName'); 其中'szTagName'从外部变量中选取。

3.系统函数

alert()函数输出字符串，主要用于调试。返回的结果显示在 Runtime.exe 的日志上。也可以显示在网页通信状态上。

语法:

譬如输出质量戳的返回值

```
alert(varQuality.toString());
```

Sleep(1000)函数，用于延迟，表示延迟 1 秒钟。单位毫秒。

6.3 使用范例

打开脚本编辑器，默认会有如下代码：

```
setInterval("MainLoop()", 1000);
```

```
function MainLoop()
```

{}

实现每隔 1000 毫秒去执行 MainLoop()函数。用户可以在 MainLoop 函数内部实现自己的逻辑控制功能。

取 Tag 的质量戳，映射到内部变量 TagName1。

```
var wQuality = GetTagQuality("rtu.Channel_.Device_.4x004");
SetTagValue("TagName1", wQuality.toString());
```

此功能在 H01TCP 中特别重要，上位机想知道系统是否通信正常，可以通过把质量戳映射到某个 3X 或者 4X 寄存器地址。

脚本编辑器支持所有数学计算的语法，譬如取 0-255 之间的随机数等，

举例说明：

```
var rand = Math.round(Math.random() * 255);
alert (rand.toString());
```

6.4 JS 实现 PID 逻辑控制

PID 逻辑控制的实现在自动化控制中非常经典，原理图如下图 6-4-1 所示：

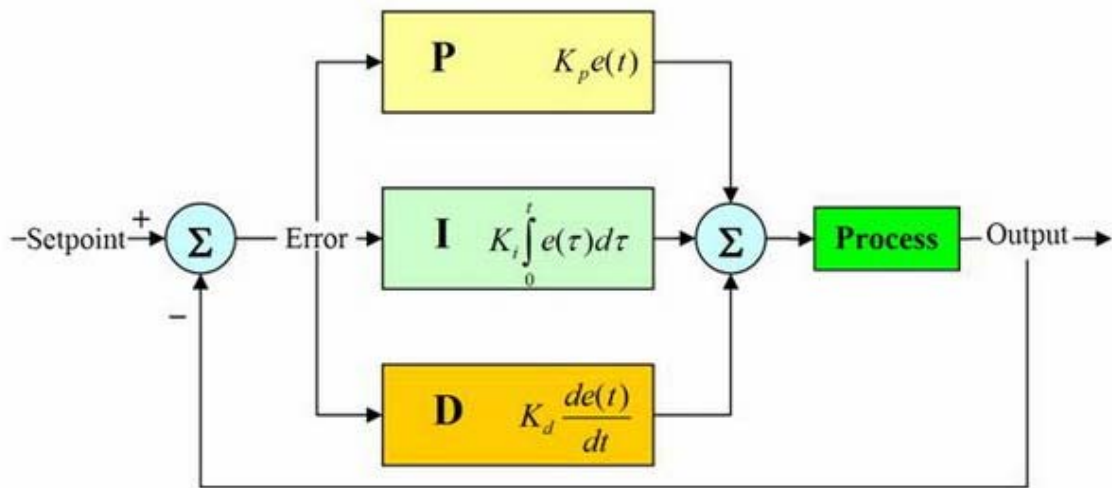


图 6-4-1 PID 逻辑控制原理图

具体实现的范例举例如下：

```

/**
 *   PID Controller.
 */
var Controller = function(k_p, k_i, k_d) {
    this.k_p = k_p || 1;
    this.k_i = k_i || 0;
    this.k_d = k_d || 0;

    this.sumError = 0;
    this.lastError = 0;
    this.lastTime = 0;

    this.target = 0; // default value, can be modified with .setTarget
};

Controller.prototype.setTarget = function(target) {
    this.target = target;
};

Controller.prototype.update = function(current_value) {
    this.current_value = current_value;

    var error = (this.target - this.current_value);

    this.sumError = this.sumError + error;

    var dError = error - this.lastError;

    this.lastError = error;

```

```

        return (this.k_p*error) + (this.k_i * this.sumError) + (this.k_d * dError);
    };

    var k_p = 0.5
        , k_i = 0.1
        , k_d = 0.2;

    // Create the controller
    var ctr = new Controller(k_p, k_i, k_d);

    var v = 120; // 120km/h
    ctr.setTarget(v);

    setInterval("MainLoop()", 100);

    var goalReached = false

    function  MainLoop(){

        var output =

        ReadFromTag("TCP.Channel_.Device_.4X1");//measureFromSomeSensor();

        var input  = ctr.update(output);

        goalReached = (Number(input) == 0) ? true : false;

        if(!goalReached){

            //applyInputToActuator(input);

            WriteToTag("TCP.Channel_.Device_.4X2', input.toString());

            alert(input.toString());

        }

    }

```

7 硬件网关说明

网关硬件参数如下：

硬件性能如下表 7-1

CPU	32 Bit 400MHz RISC ARM926EJ
内存	64M DDR2
存储器	128M Flash
操作系统	WINDOWS CE
存储扩展	SD 卡/USB
以太网	1 个 10M/100M 自适应以太网端口
USB 端口	1 个 USB Host 1.1 端口： 可支持 U 盘存储器/鼠标/键盘 1 个 USB Device 2.0 端口： 支持 USB 下载程序或更新固件
调试方式	支持通过 RS232 主控台调试 支持通过以太网 Telenet 调试
串行通讯 端口	COM1/COM2 端口： RS232/RS485 COM3 端口：RS232 COM4 端口：RS232 所有通讯口均为隔离通讯口

表 7-1 网关性能

电气规格和工作环境如下表 7-2

额定功率	< 8 W
额定电压	DC 24V，可工作范围 DC 18V~32V
电源保护	采用隔离电源模块，具备雷击浪涌保护
允许失电	< 5 ms
CE & ROHS	符合 EN61000-6-2:2005，EN61000-6-4:2007 标准，符合 ROHS 雷击浪涌±4KV，群脉冲±4KV；静电接触 4K，空气放电 8K

工作环境

工作温度	-30~70℃
存储温度	-30~85℃
环境温度	10~90%RH（无冷凝）
抗震性	10 ~ 25 Hz (X、Y、Z 方向 2G/30 分钟)
冷却方式	自然风冷
防护等级	前面板符合 IP65(配合平整盘柜安装)，机身后壳符合 IP20 整机通过 48 小时盐雾试验
机械机构	前面板采用铝合金，表面硬化氧化着色，后壳采用钣金， 表面烤漆处理
整机尺寸	263mm×110mm×28mm
整机重量	500g

表 7-2 网关电器规格和使用环境

网关的具体接口说明如下：



网关接口示意图

注意：

1. PWR（黄灯代表电源灯），正常通电时，黄灯是闪烁状态的。
2. COM（绿灯代表串口通讯指示灯）。
3. **电源（直流 24 伏）** 接口（从左到右）依次为：外壳接地线（FG），可以不接；直流负极（GND）；直流正极（DC24V）。
4. COM1/COM3（公口）和 COM2/COM4（母口）串口接线口。**其中 COM1、COM2 支持 RS485 和 RS232 方式通讯，COM3、COM4 仅支持 RS232 方式通讯。**
5. DIP 开关（用户不要随便拨动）。**正常两个开关置于“off”位置，当两个开关同时置于“on”位置时可以使硬件网关 IP 恢复到出厂默认地址。**
6. USB HOST（USB 接口）。
7. USB SLAVE（同步接口线）。
8. Ethernet（网络接口线）。

协议转换硬件网关串行通讯引脚定义

管脚排列图  COM1/COM3（公口） 第一排 （上边）从左到右依次对应12345； 第二排 （下边）从左到右依次对应6789；	COM1/COM3 通讯端口9 针D 型公座管脚排列图。这个端口用于协议转换网关连接具有RS232/485 通讯端口的控制器通讯。 注意：这个端口从外观上看是一个端口，实际上可以当 2 个端口来使用，COM1是通讯口，可以与PLC 以RS232/RS-485 通讯，COM3只支持RS232 通讯方式。			
	管脚	信号	[RS-485]	[RS-232]
	1	Rx- (B)	COM1： RS485B	
	2	RxD_PLC		COM1： RS232接收
	3	TxD_PLC		COM1： RS232发送
	4	Tx-		
	5	GND		信号地
	6	Rx+ (A)	COM1： RS485A	
	7	RxD_PC/ PLC		COM3： RS232 接收
	8	TxD_PC/ PLC		COM3： RS232 发送
	9	Tx+		

表 7-3 公口引脚定义

管脚排列图  COM2/COM4 (母口) 第一排 (上边) 从左到右依次对应 54321; 第二排 (下边) 从左到右依次对应 9876;	COM2/COM4 通讯端口9 针D 型母座管脚排列图。这个端口用于协议转换网关连接具有RS232/485 通讯端口的控制器，COM4 只支持RS232通讯方式。 此端口排列管脚排列如下表所示。			
	管脚	信号	[RS-485]	[RS-232]
	1	Rx- (B)	COM2: RS485B	
	2	Rx+ (A)	COM2: RS485A	
	3	Tx-		
	4	Tx+		
	5	GND		信号地
	6	RxD		Com4:RS232 接收
	7	RxD		Com2:RS232 接收
	8	TxD		Com2:RS232 发送
	9	TxD		Com4:RS232 发送

表 7-4 母口引脚定义

8 恢复出厂 IP 设置

如果用户在使用硬件网关的过程中遇到某些原因遗忘了网关的 IP 地址，也可以手动恢复硬件网关的出厂默认 IP 地址，具体方法如下：

首先，在通电状态下将硬件网关的两个 Dip 开关同时置于“ON”的位置，如下图所示 8-1，大约两秒之后会听到“嘀”的一声，然后再将两个 Dip 开关拨回去置于“Off”位置，最后将网关掉电重启即可。之后硬件网关的 IP 地址会恢复到出厂默认的 IP 地址 192.168.1.88，子网掩码 255.255.255.0，默认网关 192.168.1.1。



图 8-1 恢复出厂 IP 默认设置

