

# Modbus Plus 驱动程序

© 2022 PTC Inc. 保留所有权利。

# 目录

<b>Modbus Plus 驱动程序</b> .....	<b>1</b>
<b>目录</b> .....	<b>2</b>
Modbus Plus 驱动程序 .....	4
<b>概述</b> .....	<b>5</b>
外部依存关系 .....	5
<b>设置</b> .....	<b>6</b>
通道属性 - 常规 .....	6
标记计数 .....	7
通道属性 - 写入优化 .....	7
通道属性 - 高级 .....	8
通道属性 - 适配器 .....	8
设备属性 - 常规 .....	10
设备 ID .....	11
设备属性 - 扫描模式 .....	14
设备属性 - 定时 .....	14
设备属性 - 自动降级 .....	15
设备属性 - 标记生成 .....	16
设备属性 - 块大小 .....	17
设备属性 - 变量导入设置 .....	18
设备属性 - 设置 .....	20
设备属性 - 冗余 .....	22
<b>自动标记数据库生成</b> .....	<b>23</b>
<b>优化通信</b> .....	<b>24</b>
<b>数据类型说明</b> .....	<b>25</b>
<b>地址说明</b> .....	<b>26</b>
Modbus 寻址 .....	26
功能代码说明 .....	28
配置全局数据通信设备 .....	29
TIO 模块寻址 .....	30
<b>事件日志消息</b> .....	<b>31</b>
块中的地址错误。  块范围 = <开始> 至 <结束>。 .....	31
块中的地址错误。  块范围 = H<开始> 至 H<结束>。 .....	31
无法启动 MBPLUS.SYS 设备。 .....	31
无法检测卡或启动 Modbus Plus 服务。请验证是否已正确安装卡和 MBP *.sys 驱动程序。 .....	31
无法创建运行此驱动程序所需的系统资源。 .....	31
无法初始化通道。 .....	31
不良数组。  数组范围 = <开始> 至 <结束>。 .....	31
无法加载通道。每个 Hilscher 适配器仅允许一个通道。修改项目，使每个通道都具有唯一的适配器，然后重新加载。 .....	32
导入标记数据库时打开文件出错。  OS 错误 = '<错误>'。 .....	32
打开 MBPLUS 路径时出错。  路径 = '<路径>'。 .....	32

收到的块长度与预期的长度不匹配。  收到的长度 = <数字> (字节), 需要的长度 = <数字> (字节)。	32
无法从设备访问全局数据。 .....	32
从设备读取全局数据时出错。 .....	32
设备上的块请求响应异常。  块范围 = <开始> 至 <结束>, 异常 = <代码>。 .....	32
无法写入设备上的地址。设备响应异常。  地址 = '<地址>', 异常 = <代码>。 .....	33
无法从设备上的地址读取。设备响应异常。  地址 = '<地址>', 异常 = <代码>。 .....	33
块地址请求响应异常。  块范围 = H<开始> 至 H<结束>, 异常 = <代码>。 .....	33
警告: 全局数据已禁用, 访问需要 Modicon 的 4.0 低级系统驱动程序。 .....	33
无法打开适配器。  适配器 = <名称>。 .....	33
标记导入由于内存资源不足而失败。 .....	33
标记导入期间发生文件异常。 .....	34
解析导入文件中的记录出错。  记录编号 = <编号>, 字段 = <编号>。 .....	34
针对导入文件中的记录截断了说明。  记录编号 = <编号>。 .....	34
导入的标记名称无效, 且已经更改。  标记名称 = '<标记>', 更改后的标记名称 = '<标记>'。 .....	34
由于不支持数据类型, 因此无法导入标记。  标记名称 = '<标记>', 不支持的数据类型 = '<类型>'。 ..	34
无法写入设备上的地址。电路板响应异常。  地址 = '<地址>', 异常 = <代码>。 .....	35
无法从设备上的地址读取。板响应异常。  地址 = '<地址>', 异常 = <代码>。 .....	35
启动了 MBPLUS.SYS 设备 .....	35
正在导入标记数据库。  源文件 = '<文件名>' .....	35
<b>Modbus 异常代码</b> .....	<b>36</b>
<b>索引</b> .....	<b>37</b>

---

## Modbus Plus 驱动程序

---

帮助版本 1.061

### 目录

#### 概述

什么是 Modbus Plus 驱动程序？

#### 设置

如何配置使用此驱动程序的设备？

#### 自动生成标记数据库

如何配置 Modbus Plus 驱动程序 的标记？

#### 优化通信

如何从驱动程序获得最佳性能？

#### 数据类型说明

Modbus Plus 驱动程序 支持哪些数据类型？

#### 地址说明

如何对 Modbus Plus 设备上的数据位置进行寻址？

#### 事件日志消息

Modbus Plus 驱动程序 会产生哪些错误消息？

---

## 概述

Modbus Plus 驱动程序 提供将 Modbus Plus 设备连接 OPC 客户端应用程序的可靠方式；其中包括 HMI、SCADA、Historian、MES、ERP 和无数自定义应用程序。其适用于 Modicon SA85 或 Schneider PCI-85 接口卡。此驱动程序不支持 SA85/PCI-85 卡位于同一台计算机的配置。

---

## 外部依存关系

此驱动程序具有外部依存关系。

需要 SA85 或 PCI-85 接口适配器及其设备驱动程序软件 (MBPLUS 或 MBX 驱动程序) 才能与 Modbus Plus 网络进行通信。接口适配器和设备驱动程序可以从 Modicon/Schneider 购买。服务器上的每块 SA85 或 PCI-85 卡最多可支持 8 个通道。

## 设置

对于此驱动程序，术语“Modbus 服务器”和“主动提供”可交替使用。

### 通道和设备限制

此驱动程序支持的最大通道数量为 32。此驱动程序所支持设备的最大数量为每通道 8192 个。Modbus Plus 驱动程序 对于每个 SA85 卡适配器最多支持 8 个通道。

### 支持 接口卡

SA85 卡

● **注意：**用户还可以通过 USB 适配器从 Modbus RTU 串行计算机连接到 Modbus Plus 网络。

● 有关 SA85 卡相关要求的详细信息，请参阅[外部依赖项](#)。

### 支持的通信模式

Modbus Plus 驱动程序 支持三种用于从设备获取数据的通信模式。通过设备 ID 格式指定模式。选项包括“非主动”、“主动提供”和“邮箱”。模式说明如下：

- **非主动：**在此模式下，驱动程序生成向设备读取和/或写入数据的请求。可用地址包括输出线圈、输入线圈、内部寄存器和保存寄存器。输出线圈和保存寄存器地址具有读/写权限，而输入线圈和内部寄存器具有只读权限。用于非主动模式的设备 ID 格式是 *DM.r1.r2.r3.r4.r5* 或 *S.r1.r2.r3.r4.r5*。
- **主动提供：**在此模式下，驱动程序用作网络虚拟 PLC。读取和写入操作并非源自驱动程序。从主动提供设备上读取或写入操作的所有客户端应用程序将向分配给该设备的本地缓存传送数据，而非向物理设备传送。此本地缓存位于正在运行驱动程序的 PC 中。网络上的设备可通过主动提供命令向同一缓存进行读取和写入操作。用于主动提供模式的设备 ID 格式是 *DS.r1.r2.r3.r4.r5*。
- **邮箱：**在此模式下，驱动程序用作各定义邮箱设备的存储区域。接收主动提供命令后，驱动程序会检测到发出消息的路由路径，并将数据放置于为该设备分配的存储区域。如果发出消息的设备的路由路径尚未定义为邮箱设备，则不会处理该消息。从邮箱设备进行读取操作的所有客户端应用程序将读取驱动程序中的存储区域，而非物理设备；但是，写入将同时发送到物理设备以及本地缓存。此模式下仅支持保存寄存器。不支持双精度数据类型。用于邮箱模式的设备 ID 格式是 *U.r1.r2.r3.r4.r5*。
  - **注意：**主动提供的邮箱命令可通过某些 Modicon 设备中的 MSTR 指令来实现。有关详细信息，请参阅[设备 ID](#) 中的“示例 2 - 邮箱模式单个网络”和“示例 3 - 邮箱模式桥接网络”。

● 要了解 SA85 接口卡支持的通信模式，请参阅下表。

模式	SA85 卡
非主动	x
主动提供	x
邮箱	x

● 有关详细信息，请参阅[设备 ID](#)。

### 轮询多个设备

Modbus Plus 驱动程序 可以轮询 Modbus Plus 网络上的多个设备，也可以用作 Modbus Plus 网络上的单个 Modbus 服务器设备，对其他设备进行轮询。驱动程序限制为 8192 个设备，最多可支持 4 个适配器。

### 通道属性 - 常规

此服务器支持同时使用多个通信驱动程序。服务器项目中使用的各个协议或驱动程序称为通道。服务器项目可以由具有相同通信驱动程序或具有唯一通信驱动程序的多个通道组成。通道充当 OPC 链路的基础构建块。此组用于指定常规通道属性，如标识属性和操作模式。

属性组	<input type="checkbox"/> <b>标识</b>	
常规	名称	通道 1
写优化	说明	
高级	驱动程序	
持久存储	<input type="checkbox"/> <b>诊断</b>	
	诊断数据捕获	禁用

## 标识

**“名称”**: 指定此通道的用户定义标识。在每个服务器项目中，每个通道名称都必须是唯一的。尽管名称最多可包含 256 个字符，但在浏览 OPC 服务器的标记空间时，一些客户端应用程序的显示窗口可能不够大。通道名称是 OPC 浏览器信息的一部分。该属性是创建通道所必需的。

● 有关保留字符的信息，请参阅服务器帮助中的“如何正确命名通道、设备、标记和标记组”。

**“说明”**: 指定此通道的用户定义信息。

● 在这些属性中，有很多属性 (包括“说明”) 具有关联的系统标记。

**“驱动程序”**: 为该通道指定的协议/驱动程序。指定在创建通道期间选择的设备驱动程序。它在通道属性中为禁用设置。该属性是创建通道所必需的。

● **注意**: 服务器全天在线运行时，可以随时更改这些属性。其中包括更改通道名称以防止客户端向服务器注册数据。如果客户端在通道名称更改之前已从服务器中获取了项，那么这些项不会受到任何影响。如果客户端应用程序在通道名称更改之后发布项，并尝试通过原来的通道名称重新获取项，则该项将不被接受。一旦开发完成大型客户端应用程序，就不应对属性进行任何更改。采用适当的用户角色和权限管理来防止操作员更改属性或访问服务器功能。

## 诊断

**“诊断数据捕获”**: 启用此选项后，通道的诊断信息即可提供给 OPC 应用程序，。由于服务器的诊断功能所需的开销处理量最少，因此建议在需要时使用这些功能，而在不需要时禁用这些功能。默认设置为禁用状态。

● **注意**: 如果驱动程序不支持诊断，则该属性不可用。

● 有关详细信息，请参阅服务器帮助中的“通信诊断”和“统计信息标记”。

## 标记计数

**“静态标记”**: 提供此级别 (设备或通道) 上已定义静态标记的总数。此信息有助于排除故障和平衡负载。

## 通道属性 - 写入优化

服务器必须确保从客户端应用程序写入的数据能够准时发送到设备。为此，服务器提供了优化属性，用以满足特定需求或提高应用程序响应能力。

属性组	<input type="checkbox"/> <b>写优化</b>	
常规	优化方法	仅写入所有标记的最新值
写优化	占比	10
高级		
持久存储		

## 写入优化

**“优化方法”**: 控制如何将写入数据传递至底层通信驱动程序。选项包括：

- **“写入所有标记的所有值”(Write All Values for All Tags)**: 此选项可强制服务器尝试将每个值均写入控制器。在此模式下，服务器将持续收集写入请求并将它们添加到服务器的内部写入队列。服务器将对写入队列进行处理并尝试通过将数据尽快写入设备来将其清空。此模式可确保从客户端应用程序写入的所有数据均可发送至目标设备。如果写入操作顺序或写入项的内容必须且仅能显示于目标设备上，则应选择此模式。

- **“写入非布尔标记的最新值”**：由于将数据实际发送至设备需要一段时间，因此对同一个值的多次连续写入会存留于写入队列中。如果服务器要更新已位于写入队列中的某个写入值，则需要大大减少写入操作才能获得相同的最终输出值。这样一来，便不会再有额外的写入数据存留于服务器队列中。几乎就在用户停止移动滑动开关时，设备中的值达到其正确值。根据此模式的规定，任何非布尔值都会在服务器的内部写入队列中更新，并在下一个可能的时机发送至设备。这可以大大提高应用性能。
  - **注意**：该选项不会尝试优化布尔值的写入。它允许用户在不影响布尔运算的情况下优化 HMI 数据的操作，例如瞬时型按钮等。
- **“写入所有标记的最新值”**：该选项采用的是第二优化模式背后的理论并将其应用至所有标记。如果应用程序只需向设备发送最新值，则该选项尤为适用。此模式会通过当前写入队列中的标记发送前对其进行更新来优化所有的写入操作。此为默认模式。

**“占空比”(Duty Cycle)**：用于控制写操作与读操作的比率。该比率始终基于每一到十次写入操作对应一次读取操作。占空比的默认设置为 10，这意味着每次读取操作对应十次写入操作。即使在应用程序执行大量的连续写入操作时，也必须确保足够的读取数据处理时间。如果将占空比设置为 1，则每次读取操作对应一次写入操作。如果未执行任何写入操作，则会连续处理读取操作。相对于更加均衡的读写数据流而言，该特点使得应用程序的优化可通过连续的写入操作来实现。

● **注意**：建议在将应用程序投入生产环境前使其与写入优化增强功能相兼容。

## 通道属性 - 高级

此组用于指定高级通道属性。并非所有驱动程序都支持所有属性，因此不会针对不支持的设备显示“高级”组。

属性组	<input type="checkbox"/> <b>非规范浮点数处理</b>	
常规	浮点值	替换为零
以太网通信	<input type="checkbox"/> <b>设备间延迟</b>	
写优化	设备间延迟 (毫秒)	0
<b>高级</b>		
通信序列化		

**“非规范浮点数处理”**：非规范值定义为无穷大、非数字(NaN)或非规范数。默认值为“替换为零”。具有原生浮点数处理功能的驱动程序可能会默认设置为“未修改”。通过非规范浮点数处理，用户可以指定驱动程序处理非规范 IEEE-754 浮点数据的方式。选项说明如下：

- **“替换为零”**：此选项允许驱动程序在将非规范 IEEE-754 浮点值传输到客户端之前，将其替换为零。
- **“未修改”**：此选项允许驱动程序向客户端传输 IEEE-754 非规范、规范、非数字和无穷大值，而不进行任何转换或更改。

● **注意**：如果驱动程序不支持浮点值或仅支持所显示的选项，则将禁用此属性。根据通道的浮点规范设置，将仅对实时驱动程序标记(如值和数组)进行浮点规范。例如，此设置不会影响 EFM 数据。

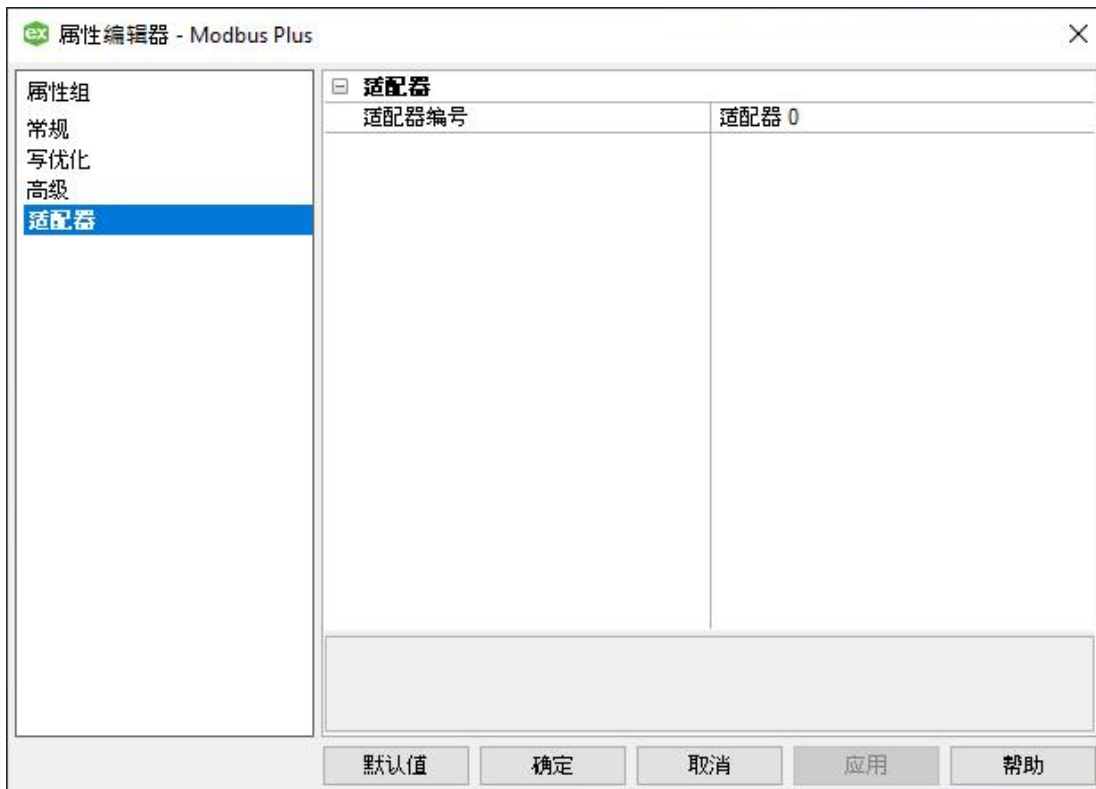
● 有关浮点值的详细信息，请参阅服务器帮助中的“如何使用非规范浮点值”。

**“设备间延迟”**：指定在接收到同一通道上的当前设备发出的数据后，通信通道向下一设备发送新请求前等待的时间。设置为零(0)将禁用延迟。

● **注意**：此属性并不适用于所有驱动程序、型号和相关设置。

## 通道属性 - 适配器





**“适配器编号”**: 指定 Modbus Plus 卡将使用的适配器编号。有效适配器编号为 0 至 3。有关卡的特定信息，请参阅[设置](#)。

## 设备属性 - 常规

标识	
名称	Modbus
说明	
驱动程序	Modbus Plus
型号	Modbus
通道分配	Modbus Plus
ID	DM.1.0.0.0.0
操作模式	
数据收集	启用
模拟	否
标记计数	
静态标记	1

**名称**  
指定此对象的标识。

默认值    确定    取消    应用    帮助

### 标识

**“名称”**: 此设备的用户定义标识。

**“说明”**: 有关此设备的用户定义信息。

**“通道分配”**: 该设备当前所属通道的用户定义名称。

**“驱动程序”**: 为该设备选择的协议驱动程序。

**“型号”**: 设备的特定版本。

**ID**: 指定网络上节点的路径。设备 ID 用于指定网络上节点的路径。除模式指示符外，它还包含五个连续的路由字节。

● 有关详细信息，请参阅 [设备 ID](#)。

### 操作模式

**“数据收集”**: 此属性控制设备的活动状态。尽管默认情况下会启用设备通信，但可使用此属性禁用物理设备。设备处于禁用状态时，不会尝试进行通信。从客户端的角度来看，数据将标记为无效，且不接受写入操作。可随时通过此属性或服务器帮助中的设备“系统标记”更改此属性。

**“模拟”**: 此选项可将设备置于模拟模式。在此模式下，驱动程序不会尝试与物理设备进行通信，但服务器将继续返回有效的 OPC 数据。模拟停止与设备的物理通信，但允许 OPC 数据作为有效数据返回到 OPC 客户端。在“模拟模式”下，服务器将所有设备数据处理为反射型：无论向模拟设备写入什么内容，都会读取回来，而且会单独处理每个 OPC 项。项的内存映射取决于组更新速率。如果服务器移除了项（如服务器重新初始化时），则不保存数据。默认值为“否”。

● **注意**：

1. “系统”标记 (**Simulated**) 为只读且无法写入，从而达到运行时保护的目的。“系统”标记允许从客户端监控此属性。
2. 在“模拟”模式下，项的内存映射取决于客户端更新速率 (OPC 客户端的“组更新速率”或本机和 DDE 接口的扫描速率)。这意味着，参考相同项、而采用不同更新速率的两个客户端会返回不同的数据。

模拟模式仅用于测试和模拟目的。该模式永远不能用于生产环境。

## 设备 ID

设备 ID 用于指定网络上节点的路径。它包含模式指示符和五个连续的路由字节。模式可能是数据客户端 (DM)、数据服务器 (DS) 或邮箱。对于此驱动程序，术语“Modbus 服务器”和“主动提供”可交替使用。

## 非主动模式 (数据客户端)

数据客户端路径以 DM 或 S 前缀开头，用于与网络上的另一个节点进行通信。主机系统在此类型会话中作为 Modbus 客户端使用。DM 路径可以标识 PLC 或可以响应 Modbus 读取和写入命令的任何其他设备，包括运行 Modbus Plus 驱动程序的另一个主机。DM 路径的格式为 *DM.r1.r2.r3.r4.r5* 或 *S.r1.r2.r3.r4.r5*。

## 主动模式 (数据服务器)

每个 SA85 卡可配置一个数据服务器路径，且格式为 *DS.1.0.0.0.0*。通过定义服务器路径，用户可启用运行 Modbus Plus 驱动程序的主机来模拟网络上的设备 (能够响应来自其他设备的读/写请求)。其他设备可以通过打开其数据客户端路径与此模拟设备进行通信。

模拟设备使用 Modbus 字节排序：对于 32 位和 64 位值，首字为双字型低位字，对于用于数据编码的 64 位值，首双字为低位双字。因此，必须将主动设备的“数据编码”选项设置如下：

- “Modbus 字节顺序”
- “首字排序为低”
- “首双字排序为低”

有关详细信息，请参阅 [设置](#)。

输出线圈、输入线圈、内部寄存器和保存寄存器实施地址 1 到 65536。驱动程序将响应从外部设备 (功能代码 [十进制] 01、02、03、04、05、06、15 和 16) 读取或写入这些值的任何有效请求。这些位置也可作为分配给 Modbus 服务器设备的标记通过主机 PC 本地访问。主动设备不支持只写访问权限。

启用 Modbus 服务器路径后，Modbus Plus 驱动程序会对每个 SA85 卡启用八个 Modbus 服务器路径。这样，远程 PLC 和其他 Modbus Plus 设备就可以利用这八个 Modbus 服务器路径中的任意一个来访问此驱动程序的 Modbus 服务器内存。在所有情况下，访问的内存都是相同的。根据 MSTR 指令，当定义在此驱动程序服务的 SA85 卡上连接的路径时，用户可指定路径 1 到 8。如果应用程序有大量向 PC 发送数据的远程设备，则此功能会很有用。在此情况下，用户可以利用八个 Modbus 服务器路径来从远程节点分配负载。此驱动程序中的每个 Modbus 服务器路径都有其自身的执行线程，以确保达到最佳性能水平。

如果项目中未定义任何 Modbus 服务器设备，则驱动程序会忽略其收到的任何主动读取或写入请求。

## 邮箱模式

邮箱路径以前缀 U 开头，并提供物理设备的路径。在项目中定义的 Modbus 服务器设备中，为此物理设备提供了存储区域。虽然物理设备会将主动写入发送至此存储区域，但这些写入也可作为分配给 Modbus 服务器设备的标记通过主机 PC 本地访问。邮箱路径的格式为 *U.r1.r2.r3.r4.r5*。

接收主动提供的邮箱数据时，驱动程序总是会打开 Modbus 服务器路径。驱动程序打开的路径为 *DS.1.0.0.0.0*。同一 Modbus Plus 网络上的设备通过打开数据客户端路径 DM.<本地节点>.1.0.0.0 与驱动程序进行通信，其中本地节点是在主机的 SA85 卡上设置的地址。有关在桥接的网络上使用路径设备的说明，请参阅 [示例 3](#)。

设备使用 Modbus Plus MSTR 指令以向驱动程序提供数据。对于能够将数据关联到特定设备的驱动程序，设备 ID 路径必须嵌入接收数据的前五个寄存器。如果数据的前五个寄存器与项目中设备的设备 ID 路径不匹配，则会丢弃接收到的数据。MSTR 指令仅支持写入命令。

注意：

1. 设备 ID 路径会嵌入主机 PC 到设备的路径，而不会嵌入设备到主机 PC 的路径。
2. TIO 模块设备不支持 Modbus 服务器网络地址。
3. OPC 客户端处于连接状态时，不应更改设备 ID。这时，只有在所有 OPC 客户端都断开连接然后再重新连接之后，更改才会生效。

### 示例 1 - 非主动模式

一个网络包含四个节点，其中节点 1 和节点 4 是运行使用 Modbus Plus 驱动程序的主机 PC。节点 2 和节点 3 是 PLC。下表显示了自每个节点的网络寻址。

自	到节点 1	到节点 2	到节点 3	到节点 4
节点 1	-----	DM.2.0.0.0.0	DM.3.0.0.0.0	DM.4.1.0.0.0
节点 2	DM.1.1.0.0.0	-----	DM.3.0.0.0.0	DM.4.1.0.0.0
节点 3	DM.1.1.0.0.0	DM.2.0.0.0.0	-----	DM.4.1.0.0.0
节点 4	DM.1.1.0.0.0	DM.2.0.0.0.0	DM.3.0.0.0.0	-----

● **注意：**要访问主机 PC 上的模拟设备，路径中的最后一个非零数字应该始终为一。这表示驱动程序所使用的 Modbus 服务器路径。

### 示例 2 - 邮箱模式 单个网络

将寄存器 40020 到 40029 从设备转移到主机 PC 的位置 40001 到 40010。控制块的位置可能有所不同。主机 PC 地址是 7.0.0.0.0。设备地址是 3.0.0.0.0。

#### MSTR 指令

控制块	40001	-
数据区域	40015	提早启动五个寄存器
长度	15	比实际数据多五

#### 控制块

寄存器	内容	说明
40001	1	写入操作
40002	0	保留错误代码
40003	15	要传输的寄存器数量
40004	1	主机 PC 中的起始位置 (寄存器 40001)
40005	7	至主机 PC 的路径
40006	1	至主机 PC 的路径
40007	0	至主机 PC 的路径
40008	0	至主机 PC 的路径
40009	0	至主机 PC 的路径

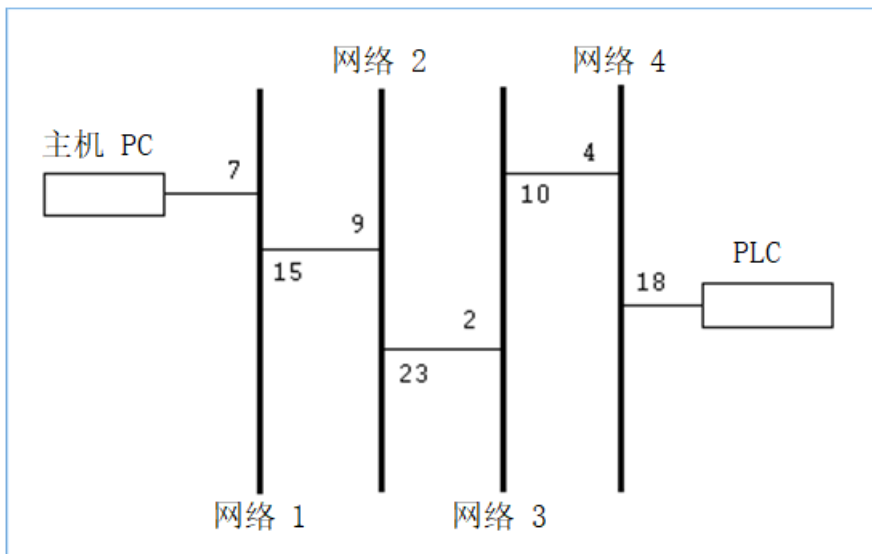
#### 数据区域

寄存器	内容	说明
40015	3	从主机 PC 返回设备的路径，设备 ID
40016	0	从主机 PC 返回设备的路径
40017	0	从主机 PC 返回设备的路径
40018	0	从主机 PC 返回设备的路径
40019	0	从主机 PC 返回设备的路径
40020	-	实际起始数据
40029	-	实际终止数据

在收到主动提供的消息时，驱动程序会执行以下任务：

1. 如果消息可理解，驱动程序会将确认发送给传送设备。如果收到除**预设多个寄存器** (代码 0x10) 以外功能的消息，驱动程序会返回“功能未执行”的响应。“预设多个寄存器”是由 MSTR 指令接收端上的设备使用的功能代码。如果该消息无法理解或不完整，驱动程序将返回异常响应。
2. 驱动程序会尝试将接收到的前五个寄存器的数据与项目中设备的设备 ID 路径进行匹配。如果未找到，则会将数据丢弃。如果数据不足六个寄存器，则会立即被丢弃。
3. 驱动程序会将 n - 5 个寄存器的数据 (从收到的数据的第六个寄存器开始) 复制到内部保留的设备图像映射 (从消息中指示的位置开始)。如果这是这些位置接收到的第一批数据，驱动程序可能需要为图像映射分配存储空间。
4. 该数据可提供给驱动程序的客户端。此示例中的数据会被参考为设备 (设备 ID 为 U.3.0.0.0) 的标记 (地址为 40001 到 40009)。客户端会使用在项目创建该设备时分配的逻辑名称来引用设备。数据也可以作为一个数组 (如 40001[10] 或 40001[2][5]) 来参考。

### 示例 3 - 邮箱模式 桥接网络



从 PLC 的角度来看，主机 PC 的地址是 4.2.9.7.1。从主机 PC 的角度来看，PLC 地址是 15.23.10.18.0。这是设备 ID 路径。如果已将相同寄存器从 PLC 转移到主机 PC 中的相同位置，则控制块和数据区域会根据下表用于 MSTR 指令。同样会对消息进行处理。

● **注意：**当使用此驱动程序时，除了设备以外，主机 PC 最多只能有三个网络。

#### MSTR 指令

控制块	40001	
数据区域	40015	尽早启动五个寄存器。
长度	15	比实际数据多五。

#### 控制块

寄存器	内容	说明
40001	1	写入操作
40002	0	保留错误代码
40003	15	要传输的寄存器数量
40004	1	主机 PC 中的起始位置 (寄存器 40001)
40005	4	至主机 PC 的路径
40006	2	至主机 PC 的路径
40007	9	至主机 PC 的路径
40008	7	至主机 PC 的路径
40009	1	至主机 PC 的路径

## 数据区域

寄存器	内容	说明
40015	15	从主机 PC 返回设备的路径, 设备 ID
40016	23	从主机 PC 返回设备的路径
40017	10	从主机 PC 返回设备的路径
40018	18	从主机 PC 返回设备的路径
40019	0	从主机 PC 返回设备的路径
40020	-	实际起始数据
40029	-	实际终止数据

## 设备属性 - 扫描模式

“扫描模式”为需要设备通信的标记指定订阅客户端请求的扫描速率。同步和异步设备的读取和写入会尽快处理; 不受“扫描模式”属性的影响。

属性组	扫描模式	
常规	扫描模式	遵循客户端指定的扫描速率
扫描模式	来自缓存的初始更新	禁用
定时		

“扫描模式”: 为发送到订阅客户端的更新指定在设备中扫描标记的方式。选项说明如下:

- “遵循客户端指定的扫描速率”: 此模式可使用客户端请求的扫描速率。
- “不超过扫描速率请求数据”: 此模式可将该数值集指定为最大扫描速率。有效范围为 10 至 99999990 毫秒。默认值为 1000 毫秒。  
● **注意:** 当服务器有活动的客户端和设备项且扫描速率值有所提高时, 更改会立即生效。当扫描速率值减小时, 只有所有客户端应用程序都断开连接, 更改才会生效。
- “以扫描速率请求所有数据”: 此模式将以订阅客户端的指定速率强制扫描标记。有效范围为 10 至 99999990 毫秒。默认值为 1000 毫秒。
- “不扫描, 仅按需求轮询”: 此模式不会定期轮询属于设备的标签, 也不会在一个项变为活动状态后为获得项的初始值而执行读取操作。OPC 客户端负责轮询以便更新, 方法为写入 `_DemandPoll` 标记或为各项发出显式设备读取。有关详细信息, 请参阅服务器帮助中的“设备需求轮询”。
- “遵循标签指定的扫描速率”: 此模式将以静态配置标记属性中指定的速率强制扫描静态标记。以客户端指定的扫描速率扫描动态标记。

“来自缓存的初始更新”: 启用后, 此选项允许服务器为存储(缓存)数据的新激活标签参考提供第一批更新。只有新项参考共用相同的地址、扫描速率、数据类型、客户端访问和缩放属性时, 才能提供缓存更新。设备读取仅用于第一个客户端参考的初始更新。默认设置为禁用; 只要客户端激活标记参考, 服务器就会尝试从设备读取初始值。

## 设备属性 - 定时

设备的“定时”属性允许调整驱动程序对错误条件的响应, 以满足应用程序的需要。在很多情况下, 需要更改环境的此类属性, 以便获得最佳性能。由电气原因产生的噪音、调制解调器延迟以及较差的物理连接等因素都会影响通信驱动程序遇到的错误数或超时次数。“定时”属性特定于每个配置的设备。

属性组	通信超时	
常规	连接超时(秒)	3
扫描模式	请求超时(毫秒)	1000
定时	超时前尝试次数	3

## 通信超时

**“连接超时”(Connect Timeout):** 此属性 (主要由基于驱动程序的以太网使用) 控制建立远程设备套接字连接所需的时间长度。设备的连接时间通常比针对同一设备的正常通信请求所花费时间更长。有效范围为 1 到 30 秒。默认值通常为 3 秒钟, 但可能会因驱动程序的具体性质而异。如果驱动程序不支持此设置, 则此设置将被禁用。

● **注意:** 鉴于 UDP 连接的性质, 当通过 UDP 进行通信时, 连接超时设置不适用。

**“请求超时”:** 指定一个所有驱动程序使用的间隔来决定驱动程序等待目标设备完成响应的的时间。有效范围是 50 至 9999 毫秒 (167 分钟)。默认值通常是 1000 毫秒, 但可能会因驱动程序而异。大多数串行驱动程序的默认超时是基于 9600 波特或更高的波特率来确定的。当以较低的波特率使用驱动程序时, 请增加超时, 以补偿获取数据所需增加的时间。

**“超时前的尝试次数”:** 指定在认定请求失败以及设备出错之前, 驱动程序发出通信请求的次数。有效范围为 1 到 10。默认值通常是 3, 但可能会因驱动程序的具体性质而异。为应用程序配置的尝试次数很大程度上取决于通信环境。此属性适用于连接尝试和请求尝试。

## 定时

**“请求间延迟”:** 指定驱动程序在将下一个请求发送到目标设备之前等待的时间。它会覆盖设备关联标记的一般轮询频率, 以及一次性读取和写入次数。在处理周转时间慢的设备时, 以及担心网络负载问题时, 这种延迟很有用。为设备配置延迟会影响与通道上所有其他设备的通信。建议用户尽可能将所有需要请求间延迟的设备隔离至单独的通道。其他通信属性 (如通信序列化) 可以延长此延迟。有效范围是 0 至 300,000 毫秒; 但是, 某些驱动程序可能因某项特别设计的功能而限制最大值。默认值为 0, 它表示对目标设备的请求之间没有延迟。

● **注意:** 不是所有的驱动程序都支持“请求间延迟”。如果不可用, 则此设置不会出现。

定时	☐ 定时	
自动降级	请求间延迟 (毫秒)	0

## 设备属性 - 自动降级

自动降级属性可以在设备未响应的情况下使设备暂时处于关闭扫描状态。通过将特定时间段内无响应的设备脱机, 驱动程序可以继续优化与同一通道上其他设备的通信。该时间段结束后, 驱动程序将重新尝试与无响应设备进行通信。如果设备响应, 则该设备会进入开启扫描状态; 否则, 设备将再次开始其关闭扫描时间段。

属性组	☐ 自动降级	
常规	故障时降级	启用
扫描模式	降级超时	3
定时	降级期间 (毫秒)	10000
自动降级	降级时放弃请求	禁用
标记生成		

**“故障时降级”:** 启用后, 将自动对设备取消扫描, 直到该设备再次响应。

● **提示:** 使用 `_AutoDemoted` 系统标记来监视设备的降级状态, 确定何时对设备取消扫描。

**“降级超时”:** 指定在对设备取消扫描之前, 请求超时和重试的连续周期数。有效范围是 1 到 30 次连续失败。默认值为 3。

**“降级期间”:** 指示当达到超时值时, 对设备取消扫描多长时间。在此期间, 读取请求不会被发送到设备, 与读取请求关联的所有数据都被设置为不良质量。当此期间到期时, 驱动程序将对设备进行扫描, 并允许进行通信尝试。有效范围为 100 至 3600000 毫秒。默认值为 10000 毫秒。

**“降级时放弃请求”:** 选择是否在取消扫描期间尝试写入请求。如果禁用, 则无论是否处于降级期间都始终发送写入请求。如果启用, 则放弃写入; 服务器自动将接收自客户端的写入请求视为失败, 且不会在事件日志中记录消息。

## 设备属性 - 标记生成

自动生成标记数据库功能可使设置应用程序成为一项即插即用操作。选择可以配置为自动构建标记列表的通信驱动程序 (标记与特定于设备的数据相对应)。可以从客户端浏览这些自动生成的标记 (这取决于支持驱动程序的性质)。

●并非所有设备和驱动程序都支持全自动生成标记数据库,也并非所有都支持相同的数据类型。有关具体信息,请参阅各驱动程序的数据类型说明或支持的数据类型列表。

如果目标设备支持其自身的本地标记数据库,则驱动程序会读取设备的标记信息,并使用该数据来在服务器中生成标记。如果该设备本身不支持已命名的标记,则驱动程序会根据特定于驱动程序的信息来创建标记列表。这两个条件的示例如下:

1. 如果数据采集系统支持其自身的本地标记数据库,则通信驱动程序将使用在设备中发现的标记名称来构建服务器的标记。
2. 如果以太网 I/O 系统支持其自身可用 I/O 模块类型的检测,则通信驱动程序会基于插入以太网 I/O 机架的 I/O 模块类型在服务器中自动生成标记。

●注意:自动生成标记数据库的操作模式可进行完全配置。有关详细信息,请参阅下方的属性说明。

属性组	☐ 标记生成	
常规	设备启动时	启动时不生成
扫描模式	对于重复标记	创建时删除
定时	父组	
自动降级	允许自动生成的子组	启用
标记生成	创建	创建标记
冗余		

“属性更改时”:如果设备支持在特定属性更改时自动生成标记,系统会显示“属性更改时”选项。默认情况下,该选项设置为“是”,但可以将其设置为“否”来控制何时生成标记。在此情况下,必须手动执行“创建标记”操作以执行标记生成。

“设备启动时”:指定自动生成 OPC 标记的时间。选项说明如下:

- “启动时不生成”:此选项可防止驱动程序向服务器的标记空间添加任何 OPC 标记。这是默认设置。
- “始终在启动时生成”:此选项可使驱动程序评估设备,以便获得标记信息。每次启动服务器时,它还会向服务器的标记空间添加标记。
- “首次启动时生成”:此选项可使驱动程序在首次运行项目时评估目标设备,以便获得标记信息。它还可以根据需要向服务器标记空间添加任何 OPC 标记。

●注意:如果选择自动生成 OPC 标记的选项,添加到服务器标记空间的任何标记都必须随项目保存。用户可以在“工具”|“选项”菜单中将项目配置为自动保存。

“对于重复标记”:启用自动生成标记数据库后,服务器需要了解如何处理先前已添加的标记,或在初始创建通信驱动程序后添加或修改的标记。此设置可控制服务器处理自动生成的以及当前存在于项目中的 OPC 标记的方式。它还可以防止自动生成的标记在服务器中累积。

例如,如果用户更改机架中的 I/O 模块,并且服务器配置为“始终在启动时生成”,则每当通信驱动程序检测到新的 I/O 模块时,新标记就会添加到服务器。如果未移除旧标记,则许多未使用的标记可能会在服务器的标记空间中累积。选项包括:

- “创建时删除”:此选项可在添加任何新标记之前,将先前添加到标记空间的任何标记删除。这是默认设置。
- “根据需要覆盖”:此选项可以指示服务器仅移除通信驱动程序要用新标记替换掉的标记。所有未被覆盖的标记仍将保留在服务器的标记空间中。
- “不覆盖”:此选项可以防止服务器移除任何之前生成的标记或服务器中已存在的标记。通信驱动程序只能添加全新的标记。
- “不覆盖,记录错误”:此选项与前一选项有相同效果,并且在发生标记覆盖时,也会将错误消息发布到服务器的事件日志。



● **注意:** 删除 OPC 标记会影响通信驱动程序已自动生成的标记以及使用匹配已生成标记的名称添加的任何标记。如果标记所使用的名称可能与驱动程序自动生成的标记相匹配, 则用户应避免将此类标记添加到服务器。

**“父组”:** 此属性通过指定将要用于自动生成标记的组, 来防止自动生成的标记与已手动输入的标记发生混淆。组名称最多可包含 256 个字符。此父组具有一个根分支, 可将所有自动生成的标记添加到其中。

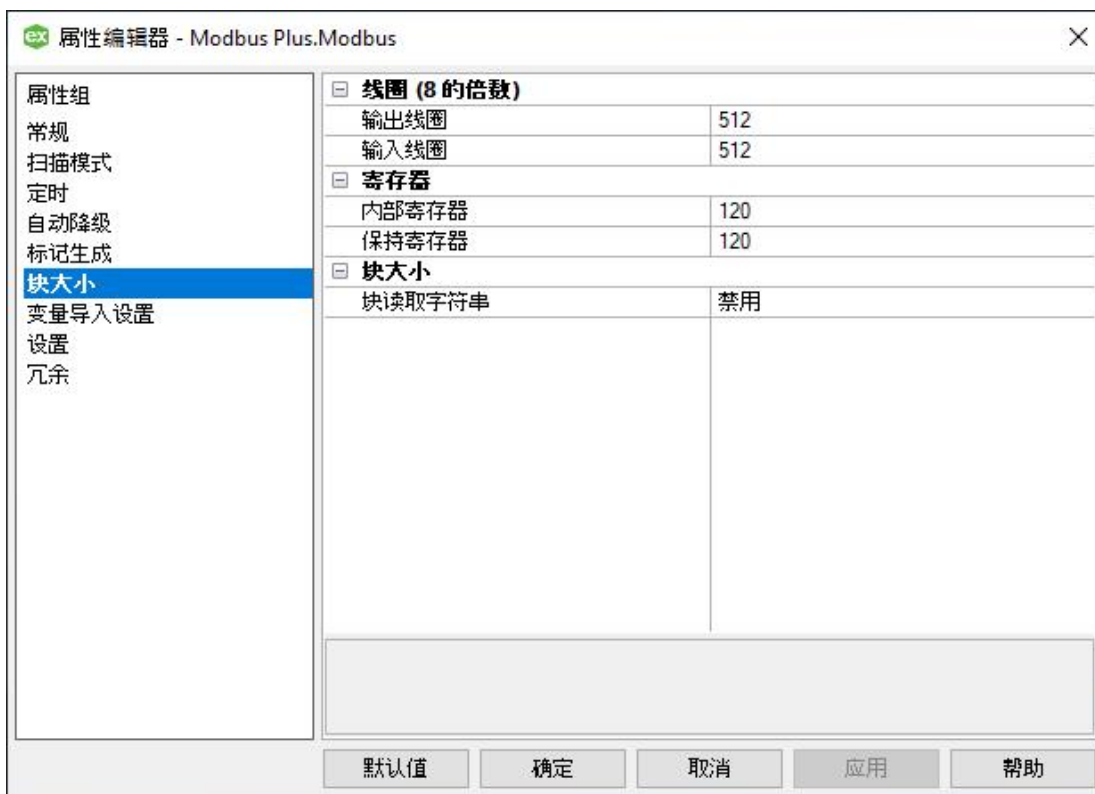
**“允许自动生成的子组”:** 此属性用于控制服务器是否为自动生成的标记自动创建子组。这是默认设置。如果禁用, 则服务器会在没有任何分组的简单列表中生成设备标记。在服务器项目中, 生成的标记使用地址值命名。例如, 生成过程中不会保留标记名称。

● **注意:** 如果在服务器生成标记的过程中, 分配给标记的名称与现有标记的名称相同, 则系统会自动递增到下一个最高数字, 以免标记名称发生重复。例如, 如果生成过程中创建了名为 "AI22" 的标记且该名称已存在, 则会将标记创建为 "AI23"。

**“创建”:** 开始创建自动生成的 OPC 标记。如果已修改设备的配置, 则**“创建标记”**可强制驱动程序重新评估设备以发现可能的标记更改。由于该选项可以通过系统标记进行访问, 这使得客户端应用程序能够启动标记数据库创建。

● **注意:** 当**“配置”**对项目进行离线编辑时, 会禁用**“创建标记”**。

## 设备属性 - 块大小



### “线圈”

**“输出线圈”:** 指定输出块大小, 以位为单位。可一次性读取线圈 8 到 2000 个点 (位)。

**“输入线圈”:** 指定输入块大小, 以位为单位。可一次性读取线圈 8 到 2000 个点 (位)。

● **注意:**

1. 线圈大小必须为 8 的倍数。
2. 对于 MBX、NETLIB 或 NONE, 默认值为 512, 最大值为 2000。
3. OPC 客户端处于连接状态时, 将禁用此属性。

## 寄存器

**“内部寄存器”**: 指定内部寄存器块大小, 以位为单位。可一次性读取 1 至 125 个 16 位标准 Modbus 寄存器。

**“保持寄存器”**: 指定保持寄存器块大小, 以位为单位。可一次性读取 1 至 125 个 16 位标准 Modbus 寄存器。

### 注意:

1. 对于 MBX、NETLIB 或 NONE, 默认值为 120, 最大值为 125。
2. OPC 客户端处于连接状态时, 将禁用此属性。
3. 对于 TIO 模块, 使用此设置可通知驱动程序在读取数据位置 400001 时返回的字节数。对于返回 2 字节的模块, 将其设置为 1。对于返回 3 字节的模块, 将其设置为 2。对于所有其他数据位置, 驱动程序使用固定的块长度 (独立于此设置)。
4. 设备可能不支持默认大小的块读取/写入操作。小型 Modicon PLC 和非 Modicon 设备可能不支持 MBPlus 网络支持的最大数据传输长度。
5. 设备可能包含非连续地址。在这种情况下, 如果驱动程序尝试读取包含未定义内存的数据块, 则设备可能会拒绝请求。

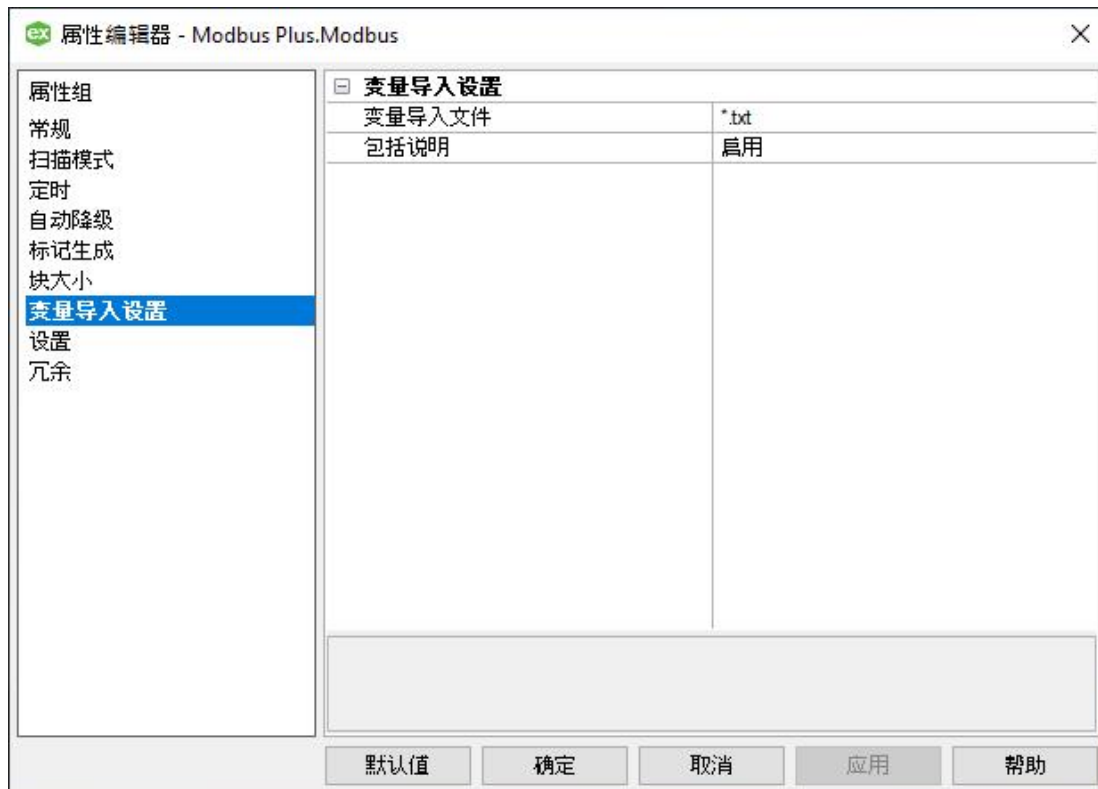
**警告:** 如果块大小值设置为大于 120, 并且将 32 位或 64 位数据类型用于任何标记, 则会出现“块中的地址错误”。请将块大小值减小到 120 以避免出现错误。

## 块大小

**“块读取字符串”**: 启用通常单独读取的字符串标记的组读取。字符串标记将根据块大小分组在一起。只能对 Modbus 型号字符串标记执行块读取。

## 设备属性 - 变量导入设置

有关 Modbus 驱动程序的 CSV 文件的详细信息, 请参阅 [创建 Modbus 驱动程序的 CSV 文件](#)。



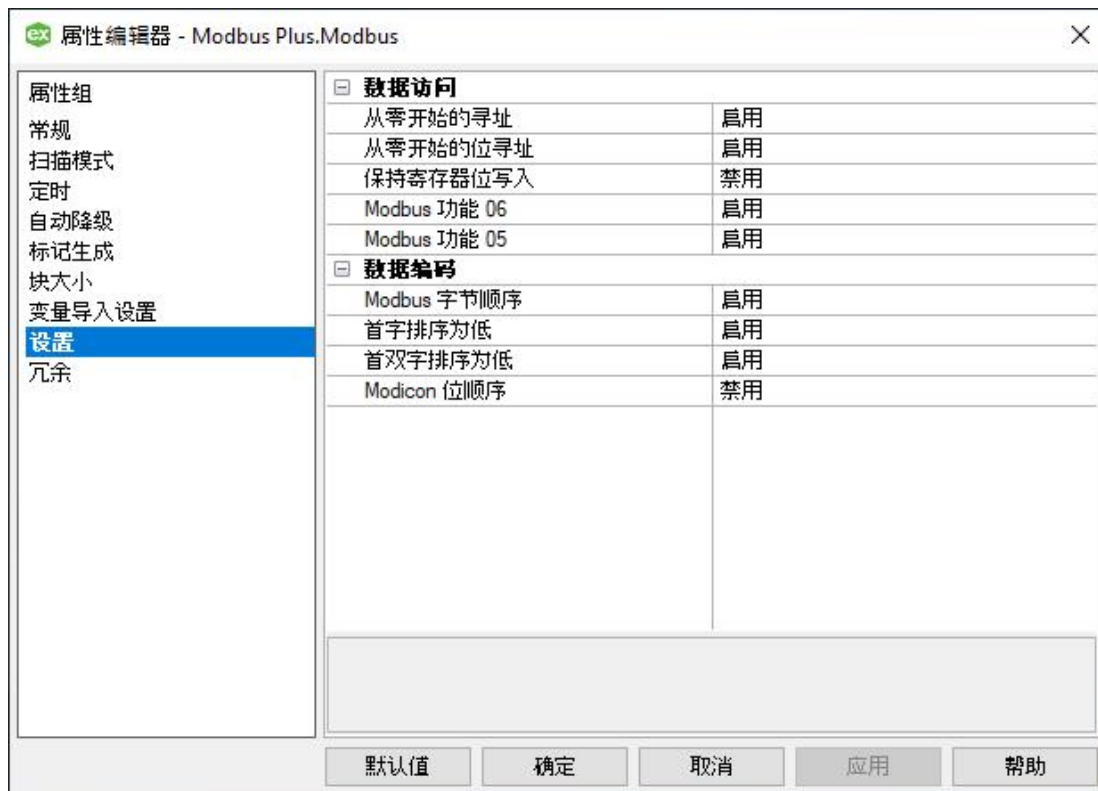
**“变量导入文件”**: 指定该设备启用自动标记数据库生成时驱动程序使用变量导入文件的确切位置。

**“包括说明”**: 启用后可使用导入的标记说明 (如果文件中存在)。

有关配置自动标记数据库生成以及如何创建变量导入文件的详细信息, 请参阅[自动标记数据库生成](#)。

有关从 *Concept* 和 *ProWORX* 创建变量导入文件的具体信息, 请参阅技术说明“为 *Modbus* 驱动程序创建 CSV 文件”。

## 设备属性 - 设置



### 数据访问

**“从零开始的寻址”**: 如果设备的地址编号约定从一开始而不是从零开始, 则应设置为禁用。默认情况下, 当将帧构建为与 Modbus 设备通信时, 用户输入的地址将减一。如果该设备不遵循此约定, 则禁用**“从零开始的寻址”**。默认行为为遵守 Modicon PLC 的约定。

**“从零开始的位寻址”**: 允许字内位的内存类型可参考为布尔型。寻址符号是: <地址>.<位>, 其中 <位> 表示字中的位数。寄存器内的“从零开始的位寻址”提供了两种对给定字的位进行寻址的方法: 从零开始和从一开始。寄存器中从零开始的位寻址意味着第一个位从 0 开始。如果是从一开始, 则第一个位从 1 开始。

**“保持寄存器位写入”**: 写入保持寄存器中的位位置时, 驱动程序仅应修改正在使用的位。某些设备支持用操纵寄存器中的单个位的命令 (功能码十六进制 0x16 或十进制 22)。如果设备不支持此功能, 则驱动程序必须执行“读取/修改/写入”操作, 以确保只更改单个位。默认设置为禁用。如果设备支持保存寄存器位访问, 并且驱动程序使用功能码 0x16, 则设置为启用, 而无需考虑“Modbus 功能码 06”的设置。如果禁用此设置, 则驱动程序会使用功能码 0x06 或 0x10, 具体取决于单个寄存器写入的“Modbus 功能码 06”属性。

● **注意:**

1. 当禁用“Modbus 字节顺序”时, 在命令中发送的掩码的字节顺序是 Intel 字节顺序。

**“Modbus 功能码 06”**: Modbus Plus 驱动程序可提供使用两种 Modbus 协议功能将保存寄存器数据写入目标设备的选项。在大多数情况下, 驱动程序可基于正在写入的寄存器数在这两个功能之间进行切换。当写入单个 16 位寄存器时, 驱动程序会使用 Modbus 功能码 06。当将 32 位值写入两个寄存器时, 驱动程序将使用 Modbus 功能码 16。对于标准 Modicon PLC, 使用这些功能中的任何一个都不是问题。然而, 已经有大量第三方设备执行了 Modbus 协议。对于其中的许多设备而言, 无论待写入的寄存器数有多少, 都仅支持使用 Modbus 功能码 16 写入保存寄存器。

如有需要, 可使用“Modbus 功能码 06”强制驱动程序仅使用 Modbus 功能码 16。默认情况下将启用此选项, 允许驱动程序根据需要在 06 和 16 之间切换。如果设备要求所有写入都只使用 Modbus 功能码 16 来完成, 则请禁用此选项。

● **注意:** 对于字内位写入,“保存寄存器位写入”属性优先于“Modbus 功能码 06”。如果启用了“保存寄存器位写入”,则会使用功能码 0x16,而无需考虑此属性如何选择。如果将其禁用,则此属性可确定将使用功能码 0x06 还是 0x10 进行字内位写入。

**Modbus 功能码 05:** Modbus Plus 驱动程序可使用两个 Modbus 协议功能将输出线圈数据写入目标设备。在大多数情况下,驱动程序可基于正在写入的线圈数在这两个功能之间进行切换。当写入单个线圈时,驱动程序将使用 Modbus 功能码 05。当写入线圈数组时,驱动程序将使用 Modbus 功能码 15。对于标准 Modicon PLC,使用这些功能中的任何一个都不是问题。然而,已经有大量第三方设备执行了 Modbus 协议。对于其中的许多设备而言,无论待写入的线圈数有多少,都仅支持使用 Modbus 功能码 15 写入输出线圈。

如有需要,可使用“Modbus 功能码 05”选项强制驱动程序仅使用 Modbus 功能码 15。默认情况下将启用此属性,允许驱动程序根据需要在 05 和 15 之间切换。如果设备要求所有写入都只使用 Modbus 功能码 15 完成,则请禁用此选择。

## 数据编码

**Modbus 字节顺序:** 使用此选项,可将 Modbus Plus 驱动程序使用的字节顺序从默认的 Modbus 字节排序更改为 Intel 字节排序。此选项默认为启用状态,这是 Modbus 兼容设备的常用设置。如果设备使用 Intel 字节排序,则禁用此选项后,驱动程序将能够正确读取 Intel 格式的数据。

**“首字排序为低”:** Modbus 设备中的两个连续寄存器地址用于 32 位数据类型。用户可指定驱动程序应该将首字假定为 32 位值的低位字还是高位字。默认设置(“首字排序为低”)遵守 Modicon Modsoft 编程软件的约定。

**“首双字排序为低”:** Modbus 设备中的四个连续寄存器地址用于 64 位数据类型。用户可指定驱动程序应将首双字假定为 64 位值的低位双字还是高位双字。默认设置(“首双字排序为低”)遵守 32 位数据类型的默认约定。

**“Modicon 位顺序”:** 启用后,驱动程序将反转读取和写入至寄存器的位顺序,以遵守 Modicon Modsoft 编程软件的约定。例如,启用此选项后,写入地址 40001.0/1 会影响设备的位 15/16。默认情况下,此选项处于禁用状态。

● **注意:** 对于以下示例,第 1 位至第 16 位表示 0-15 位或 1-16 位,具体取决于驱动程序在寄存器中是否使用“从零开始的位寻址”。

MSB = 最高有效位

LSB = 最低有效位

### 已启用“Modicon 位顺序”

MSB								LSB							
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

### 已禁用“Modicon 位顺序”

MSB								LSB							
16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1

## 数据编码选项的详细信息

- “Modbus 字节顺序”选项可用于设置每个寄存器/16 位值的数据编码。
- “首字排序为低”可设置每个 32 位值和每个 64 位值双字的数据编码。
- “首双字排序为低”可设置每个 64 位值的数据编码。

数据类型	“Modbus 字节顺序”	“首字排序为低”	“首双字排序为低”
字、短整型、BCD	适用	不适用	不适用
浮点型、双字、长整型、LBCD	适用	适用	不适用
双精度	适用	适用	适用

如果需要,请使用以下信息和特定设备的文档来确定“数据编码”选项的正确设置。默认设置适用于大多数 Modbus 设备。

已启用“Modbus 字节顺序”	高字节 (15..8)	低字节 (7..0)
已禁用“Modbus 字节顺序”	低字节 (7..0)	高字节 (15..8)
已禁用“首字排序为低”	高位字 (31..16) 64 位数据类型中双精度字的高位字 (63..48)	低位字 (15..0) 64 位数据类型中双字的低位字 (47..32)
已启用“首字排序为低”	低位字 (15..0) 64 位数据类型中双字的低位字 (47..32)	高位字 (31..16) 64 位数据类型中双字的高位字 (63..48)
已禁用“首双字排序为低”	高位双字 (63..32)	低位双字 (31..0)

## 设备属性 - 冗余

属性组	☐ 冗余	
常规	次级路径	...
扫描模式	操作模式	故障切换
定时	监视器项目	
自动降级	监视器间隔 (秒)	300
冗余	尽快返回至主要设备	是

Media-Level Redundancy 插件提供冗余。

● 有关详细信息，请参阅网站、向销售代表咨询或查阅[用户手册](#)。

## 自动标记数据库生成

Modbus Plus 驱动程序 将使用“自动标记数据库生成”，后者可以自动创建能够访问设备梯形图程序所用数据点的标记。尽管某些情况下可以在设备中查询构建标记数据库所需的信息，但此驱动程序必须改用变量导入文件。变量导入文件可以使用 Concept 和 ProWORX 设备编程应用程序生成。

### 创建变量导入文件

导入文件必须采用以分号分隔的 .TXT 格式，这是 Concept 设备编程应用程序的默认导出文件格式。

ProWORX 编程应用程序能够以此格式导出变量数据。

● 有关从 Concept 和 ProWORX 创建变量导入文件的具体信息，请参阅技术说明“为 Modbus 驱动程序创建 CSV 文件”。

### 服务器配置

可自定义“自动标记数据库生成”以满足应用程序的需求。可以在创建数据库期间通过向导或设备属性设置主要控制选项。

● 有关详细信息，请参阅服务器帮助文档。

除了所有支持“自动标记数据库生成”的驱动程序通用的基本设置外，Modbus Plus 驱动程序 还需要其他设置。这些专用设置包括要求提供变量导入文件名称及位置。此信息可在设备属性中的“变量导入设置”下指定。

● 有关详细信息，请参阅 [变量导入设置](#)。

### 操作

标记生成可在服务器项目打开时自动启动，也可在其他时间手动启动，具体取决于具体配置。“事件日志”会显示标记生成过程的开始时间、处理变量导入文件时发生的任何错误以及过程的完成时间。

## 优化通信

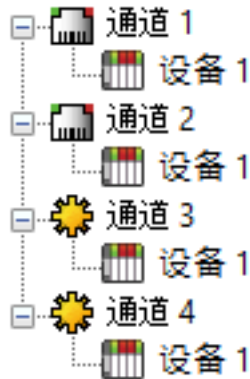
Modbus Plus 驱动程序旨在提供更好的吞吐量，并充分利用 SA85 卡。之前，Modbus Plus 驱动程序会限制用户在服务器项目中配置单通道，并要求在此通道下定义将要访问的所有 Modbus Plus 设备。这意味着，在设备之间移动时，驱动程序每次只能移动一个设备，以便发送请求。OPC 服务器的设计非常高效，因此，对于大多数应用程序，单通道方案可以提供足够的性能。作为一项使能技术的 OPC 随即出现，但是，项目大小已显著增加。为使性能保持较高水平，Modbus Plus 驱动程序的设计旨在能够高效率高性能地运行。

● **注意：**根据需要，在开始进行这些更改之前，备份服务器项目目录，以返回到先前的设置。



在本项目中，仅定义了一个通道。在该通道下定义需要访问的所有设备。Modbus Plus 驱动程序必须尽快从一个设备移动到下一个设备，以便高效地收集信息。随着更多设备的添加或从单个设备请求的信息的增加，更新速率会受到不利影响。

最新版本的 Modbus Plus 驱动程序使用多通道定义来提高应用程序的性能。在此配置中，服务器中的每个通道均表示单独的执行路径。通过添加额外通道，应用程序的工作负载即可分布到新通道中。这将创建独立运行的多个执行路径，从而使性能得到显著提高。下图显示了重新配置为使用多个通道的同一应用程序。



每个设备均可在其自身的通道下定义。在此配置中，服务器可将单个执行路径用于在单个设备中收集数据的任务中，原因在于每个设备都有其自己的专用通道。如果应用程序的设备数较少，则可按照显示的方法对其进行优化。

即使应用程序有更多设备，也同样存在增益。虽然设备数较小可能是理想情况，但附加通道仍会对应用程序有益。虽然这意味着，服务器必须在给定通道中从一个设备移动到另一个设备，但是，它现在可以用极少的设备在单个路径上进行处理。

● **注意：**通道限制应与制造商所设置的 SA85 卡的多路径限制相匹配。

即使在每个设备下均定义了大量标记，也可以对应用程序进行重新设计，以支持多个通道。有关详细信息，请参阅以下说明。

1. 在基于单通道的现有项目中，单击“**连接性**”(Connectivity)|“**新建通道**”(New Channel)，然后根据需要对其进行命名。
2. 从 ModbusPlus 通道剪切 PLC2。
3. 将其粘贴在新通道下。剪切和粘贴功能可快速修改应用程序，以利用新的 Modbus Plus 驱动程序。

这些示例强调了可通过 Modbus Plus 驱动程序实现的最明显的优化。其他可能的优化包括将单通道专用于全局数据。为此，在该新通道下，为想要访问的每个包含全局数据的设备定义一组新的设备名称。请记住，仅通过这些新定义的设备名称访问全局数据。



## 数据类型说明

数据类型	说明
布尔型	单个位
字	无符号 16 位值 位 0 是低位 位 15 是高位
短整型	有符号 16 位值 位 0 是低位 位 14 是高位 位 15 是符号位
双字	无符号 32 位值 位 0 是低位 位 31 是高位
长整型	有符号 32 位值 位 0 是低位 位 30 是高位 位 31 是符号位
BCD	两个字节封装的 BCD 值的范围是 0 至 9999。对于超出此范围的值，未定义行为。
LBCD	四个字节封装的 BCD 值的范围是 0 至 99999999。对于超出此范围的值，未定义行为。
字符串	空终止 ASCII 字符串 在 Modbus 型号上支持，包括由高到低、由低到高的字节顺序选择。
双精度*	64 位浮点值 驱动程序将四个连续寄存器解释为双精度值，方法将最后两个寄存器作为高位双字，将前两个寄存器作为低位双字。
双精度示例	如果将寄存器 40001 指定为双精度数，则寄存器 40001 的位 0 将是 64 位数据类型的位 0，寄存器 40004 的位 15 将是 64 位数据类型的位 63。
浮点型*	32 位浮点值 驱动程序将两个连续寄存器解释为单精度值，方法是将最后一个寄存器作为高位字，将第一个寄存器作为低位字。
浮点型示例	若将寄存器 40001 指定为浮点型，则寄存器 40001 的位 0 将是 32 位数据类型的位 0，寄存器 40002 的位 15 将是 32 位数据类型的位 31。

\*说明假定为默认，按首双字排序为低处理 64 位数据类型，按首字排序为低处理 32 位数据类型。

## 地址说明

地址规范因所使用的型号而异。从以下列表选择一个链接，以获取相关型号的具体地址信息。

[Modbus 寻址](#)

[TIO 模块寻址](#)

## Modbus 寻址

对于此驱动程序，术语“服务器”和“主动提供”可交替使用。

### 5 位寻址与 6 位寻址

在 Modbus 寻址中，地址的第一位指定了主表格。其余位表示设备的数据项。最大值是一个两字节的无符号整数 (65,535)。需要六位数字来表示整个地址表格和项。因此，在设备手册中指定为 0xxxx、1xxxx、3xxxx 或 4xxxx 的地址，当应用至 Modbus 标记的地址字段时，将额外填充一个零。

主表格	说明
0	输出线圈
1	输入线圈
3	内部寄存器
4	保持寄存器

### 以十进制格式表示的 Modbus 寻址 十进制格式

该功能代码以十进制显示。有关详细信息，请参阅[功能代码说明](#)。

地址类型	范围	数据类型	访问	功能代码
输出线圈	000001-065536	布尔型	读/写	01、05、15
输入线圈	100001-165536	布尔型	只读	02
内部寄存器	300001-365536	字、短整型、BCD 浮点型、双字型、长整型、LBCD 双精度	只读*	04
	300001-365535		只读*	04
	300001-365533		只读*	04
	3xxxxx.0/1-3xxxxx.15/16**	布尔型	只读	04
	300001.2H-365536.240H***	字符串	只读*	04
300001.2L-365536.240L***	字符串	只读	04	
保持寄存器	400001-465536	字、短整型、BCD 浮点型、双字、长整型、LBCD 双精度	读/写	03、06、16
	400001-465535		读/写	03、06、16
	400001-465533		读/写	03、06、16
	4xxxxx.0/1-4xxxxx.15/16**	布尔型	读/写	03、06、16、22
	400001.2H-465536.240H***	字符串	读/写	03、16
400001.2L-465536.240L***	字符串	读/写	03、16	
			读/	

地址类型	范围	数据类型	访问	功能代码
			写	
全局数据	G01-G32 G01-G31 G01-G29  Gxx.0/1-Gxx.15/16**	字、短整型、BCD 浮点型、双字、长整型、LBCD 双精度  布尔型	读/ 写 读/ 写 读/ 写  只读	不适用 不适用 不适用  不适用

\*对于 Modbus 服务器设备，这些位置是读/写。

\*\*有关详细信息，请参阅设置中的“基于零与一的寻址”。

\*\*\*.Bit 是字符串长度，范围介于 2 到 240 个字符之间。

### 以十进制格式表示的 Modbus 寻址 十六进制格式

地址类型	十进制范围	数据类型	访问
输出线圈	H000001-H010000	布尔型	读/写
输入线圈	H100001-H110000	布尔型	只读
内部寄存器	H300001-H310000 H300001-H30FFFF H300001-H30FFFD	字、短整型、BCD 浮点型、双字型、长整型、LBCD 双精度	只读* 只读* 只读*
	H3yyyyy.0/1-H3yyyyy.F/10	布尔型	只读*
	H300001.2H-H3FFFF.240H	字符串	只读
	H300001.2L-H3FFFF.240L	字符串	只读
保持寄存器	H400001-H410000 H400001-H40FFFF H400001-H40FFFD	字、短整型、BCD 浮点型、双字型、长整型、LBCD 双精度	读/写 读/写 读/写
	H4yyyyy.0/1-H4yyyyy.F/10	布尔型	读/写
	H400001.2H-H4FFFF.240H	字符串	读/写
	H400001.2L-H4FFFF.240L	字符串	读/写
全局数据	HG01-HG20 HG01-HG1F HG01-HG1D	字、短整型、BCD 浮点型、双字、长整型、LBCD 双精度	读/写 读/写 读/写
	HGyy.0/1-HGyy.F/10	布尔型	只读

\*对于 Modbus 服务器设备，这些位置是读/写。

\*\*..Bit 是字符串长度，范围介于 2 到 240 个字符之间。

### 封装线圈

封装线圈地址类型允许访问多个连续的线圈作为模拟值。仅当 Modbus 型号处于 Modbus 客户端模式下时，此功能才可用。语法如下：

输出线圈：0xxxx#nn

输入线圈：1xxxx#nn

其中

- xxxx 是第一个线圈的地址。允许使用十进制和十六进制值。
- nn 是封装进模拟值的线圈数。有效范围为 1 至 16，且仅允许使用十进制值。

● **注意:** 唯一有效的数据类型是“字”。输出线圈具有读/写访问权限，而输入线圈具有只读访问权限。位顺序使开始地址为模拟值的最低有效位 (LSB)。

### 只写访问

所有的读/写地址均可通过在地址前面加上前缀 "W" (如"W40001") 来设置为“只写”，这样便可阻止驱动程序读取位于指定地址的寄存器。客户端读取“只写”标记的任何尝试都会导致获得指定地址的上次成功写入值。如果没有任何成功写入，则客户端将接收到数值/字符串值 0/NULL 作为初始值。

**警告:** 将“只写”标记的客户端访问权限设置为“只读”将导致无法写入这些标记，且客户端将始终接收到数值/字符串值 0/NULL。

### 邮箱模式

在邮箱模式下仅支持“保存寄存器”。从客户端读取数据时，将从本地缓存而不是物理设备读取数据。当从客户端写入时，数据将同时写入本地缓存以及由设备 ID 路由路径确定的物理设备。有关详细信息，请参阅[邮箱模式](#)。

● **注意:** 不支持双精度数据类型。

### 字符串支持

Modbus 型号支持以 ASCII 字符串的形式读写保持寄存器内存。当将保持寄存器用于字符串数据时，每个寄存器包含 ASCII 数据的两个字节。定义字符串时，可以选择指定寄存器中 ASCII 数据的顺序。字符串的长度可以为 2 到 240 个字节，并代替位数输入。长度必须输入为偶数。向地址附加 "H" 或 "L" 可指定字节顺序。

#### 示例

- 对于从 40200 开始、长度为 100 字节并采用由高到低字节排序的字符串，要对其进行寻址，请输入："40200.100H"。
- 对于从 40500 开始、长度为 78 字节并采用由低到高字节排序的字符串，要对其进行寻址，请输入："40500.78L"。

● **注意:** 字符串长度可能受到设备允许的写入请求的最大大小的限制。使用字符串标记时，如果在服务器事件窗口中接收到错误消息“无法写入设备 <设备> 上的地址 <地址>: 设备响应异常，异常代码为 3”，则设备不支持字符串长度。用户应尽量缩短字符串。

### 数组支持

内部和保存寄存器位置 (包括除布尔型和字符串型之外的所有数据类型) 以及输入和输出线圈 (布尔型数据类型) 均支持数组。对数组进行寻址的方法有两种。以下示例适用于保持寄存器：

4xxxx [行数] [列数]

4xxxx [列数]，假定行数等于 1。

对于字型、短整型和 BCD 数组，基址 + (行数 \* 列数) 不能超过 65536。对于浮点型、双字型、长整型和长整型 BCD 数组，基址 + (行数 \* 列数 \* 2) 不能超过 65535。对于所有数组，正在请求的寄存器总数不能超过为此设备指定的保持寄存器块大小。

● **注意:** 全局数据的基址不能超过 32。

### 功能代码说明

十进制	十六进制	说明
01	0x01	读取线圈状态
02	0x02	读取输入状态
03	0x03	读取保持寄存器
04	0x04	读取内部寄存器
05	0x05	强制单个线圈
06	0x06	预设单个寄存器
15	0x0F	强制多个线圈
16	0x10	预设多个寄存器

十进制	十六进制	说明
22	0x16	已屏蔽的写入寄存器

## 配置全局数据通信设备

SA85 接口卡支持全局数据。它只能从单个网络访问。例如，“7.0.0.0.0”可访问全局数据，而“7.1.0.0.0”则不可访问。

●**注意：**主动模式不支持全局数据。

## 将全局数据写入设备

从 PLC 的角度来看，主机 PC 的地址是 2.0.0.0.0。从主机 PC 的角度来看，PLC 地址是 9.0.0.0.0。这是设备 ID 路径。用户必须配置地址以使设备可以读写编程软件。

### 控制块

寄存器	内容	说明
Control [1]	5	用于写入全局数据的功能代码
Control [2]	- 0 = 无错误	错误代码。这可能不会更改。
Control [3]	32	从状态 RAM 写入全局内存的字数；最大值为 32 位
Control [4]	-	保留*
Control [5]	2	向其发送数据的 Modbus Plus 节点地址
Control [6]	0	主机 PC 路径
Control [7]	0	主机 PC 路径
Control [8]	0	主机 PC 路径
Control [9]	0	主机 PC 路径

\*此寄存器特定于应用程序。

### 数据区域

寄存器	内容	说明
DataField [1]-DataField [32]	数据	不适用

## 从设备读取全局数据

从 PLC 的角度来看，主机 PC 的地址是 2.0.0.0.0。从主机 PC 的角度来看，PLC 地址是 9.0.0.0.0。这是设备 ID 路径。

### 控制块

寄存器	内容	说明
Control [1]	6	用于读取全局数据的功能代码
Control [2]	- 0 = 无错误	错误代码。这可能不会更改
Control [3]	32	从状态 RAM 写入全局内存的字数；最大值为 32 位
Control [4]	-	保留*
Control [5]	2	从中读取数据的 Modbus Plus 节点地址
Control [6]	0	主机 PC 路径
Control [7]	0	主机 PC 路径
Control [8]	0	主机 PC 路径
Control [9]	0	主机 PC 路径

\*此寄存器特定于应用程序。

### 数据区域

寄存器	内容	说明
DataField [1]-DataField [32]	数据	不适用

## TIO 模块寻址

此型号不支持“邮箱模式”。

### TIO 模块十进制寻址

地址类型	范围	数据类型	访问
数据 I/O*	400001 400001.0/1-400001.15/16**	字、短整型 布尔型	读/写 读/写
数据输入 - 锁存	400257 400257.0/1-400257.15/16**	字、短整型 布尔型	只读 只读
模块超时	461441 461441.0/1-461441.15/16**	字、短整型 布尔型	读/写 读/写
模块状态	463489-463497 4xxxxx.0/1-4xxxxx.15/16**	字、短整型 布尔型	只读 只读
模块 ASCII 标题	464513	字符串	只读

\*从“数据 I/O”位置读取的值来自模块的输入寄存器。写入此位置时，发送的值会修改模块的输出寄存器。因此，在此位置读取的值与先前写入此位置的值并不对应。

\*\*有关详细信息，请参阅 [设置](#) 中的“基于零与一的寻址”。

### TIO 模块十六进制寻址

地址类型	范围	数据类型	访问
数据 I/O*	H40001 H40001.0/1-H40001.F/10	字、短整型 布尔型	读/写 读/写
数据输入 - 锁存	H40101 H40101.0/1-40101.F/10	字、短整型 布尔型	只读 只读
模块超时	H4F001 H4F001.0/1-H4F001.F/10	字、短整型 布尔型	读/写 读/写
模块状态	H4F801-H4F809 H4yyyy.0/1-H4yyyy.F/10	字、短整型 布尔型	只读 只读
模块 ASCII 标题	H4FC01	字符串	只读

\*从“数据 I/O”位置读取的值来自模块的输入寄存器。写入此位置时，发送的值会修改模块的输出寄存器。因此，在此位置读取的值与先前写入此位置的值并不对应。

## 事件日志消息

以下信息涉及发布到主要用户界面中“事件日志”窗格的消息。。关于如何筛选和排序“事件日志”详细信息视图，请参阅 OPC 服务器帮助。服务器帮助包含许多常见的消息，因此也应对其进行搜索。通常，其中会尽可能提供消息的类型 (信息、警告) 和故障排除信息。

---

### 块中的地址错误。| 块范围 = <开始> 至 <结束>。

**错误类型：**

错误

**可能的原因：**

已尝试引用指定设备中不存在的位置。

**可能的解决方案：**

验证分配给设备的所有标记的地址，并去除参考无效位置的标记。

---

### 块中的地址错误。| 块范围 = H<开始> 至 H<结束>。

**错误类型：**

错误

**可能的原因：**

已尝试引用指定设备中不存在的位置。

**可能的解决方案：**

验证分配给设备的所有标记的地址，并去除参考无效位置的标记。

---

### 无法启动 MBPLUS.SYS 设备。

**错误类型：**

错误

**可能的原因：**

未正确配置 MBPLUS.SYS 驱动程序。

**可能的解决方案：**

验证是否可以利用“控制面板”|“设备小程序”手动启动或停止 MBPLUS 设备。当手动启动 MBPLUS.SYS 驱动程序时，modbus\_unsolicited.dll 驱动程序可以启动该驱动程序。

---

### 无法检测卡或启动 Modbus Plus 服务。请验证是否已正确安装卡和 MBP \*.sys 驱动程序。

**错误类型：**

错误

---

### 无法创建运行此驱动程序所需的系统资源。

**错误类型：**

错误

---

### 无法初始化通道。

**错误类型：**

错误

---

### 不良数组。| 数组范围 = <开始> 至 <结束>。

**错误类型：**

错误

**可能的原因：**

某一地址数组超越了地址空间末端。

**可能的解决方案：**

验证设备内存空间的大小，并相应地重新定义数组长度。

**无法加载通道。每个 Hilscher 适配器仅允许一个通道。修改项目，使每个通道都具有唯一的适配器，然后重新加载。**

---

**错误类型：**

错误

**导入标记数据库时打开文件出错。| OS 错误 = '<错误>'。**

---

**错误类型：**

错误

**打开 MBPLUS 路径时出错。| 路径 = '<路径>'。**

---

**错误类型：**

警告

**可能的原因：**

1. 未正确配置 MBPLUS.SYS 驱动程序。
2. 驱动程序无法打开指定适配器上的路径。

**可能的解决方案：**

1. 遵守安装和配置 MBPLUS 驱动程序的说明。
2. 验证是否为八个以上的通道分配了相同的适配器编号。

**收到的块长度与预期的长度不匹配。| 收到的长度 = <数字> (字节)，需要的长度 = <数字> (字节)。**

---

**错误类型：**

警告

**无法从设备访问全局数据。**

---

**错误类型：**

警告

**从设备读取全局数据时出错。**

---

**错误类型：**

警告

**设备上的块请求响应异常。| 块范围 = <开始> 至 <结束>，异常 = <代码>。**

---

**错误类型：**

警告

**可能的原因：**

请求的节点未响应。

**可能的解决方案：**



检查电缆、线路和引脚连接。

● **也可以看看：**

Hilscher CIF 异常代码

---

**无法写入设备上的地址。设备响应异常。| 地址 = '<地址>', 异常 = <代码>。**

**错误类型：**

警告

**可能的原因：**

有关异常代码的说明，请参阅“Modbus 异常代码”。

**可能的解决方案：**

请参阅“Modbus 异常代码”。

● **也可以看看：**

Modbus 异常代码

---

**无法从设备上的地址读取。设备响应异常。| 地址 = '<地址>', 异常 = <代码>。**

**错误类型：**

警告

**可能的原因：**

有关异常代码的说明，请参阅“Modbus 异常代码”。

**可能的解决方案：**

请参阅“Modbus 异常代码”。

● **也可以看看：**

Modbus 异常代码

---

**块地址请求响应异常。| 块范围 = H<开始> 至 H<结束>, 异常 = <代码>。**

**错误类型：**

警告

**可能的原因：**

请求的节点未响应。

**可能的解决方案：**

检查电缆、线路和引脚连接。

● **也可以看看：**

Hilscher CIF 异常代码

---

**警告：全局数据已禁用，访问需要 Modicon 的 4.0 低级系统驱动程序。**

**错误类型：**

警告

---

**无法打开适配器。| 适配器 = <名称>。**

**错误类型：**

警告

---

**标记导入由于内存资源不足而失败。**

**错误类型：**

警告

**可能的原因：**

驱动程序无法分配所需内存来处理变量导入文件。

**可能的解决方案：**

关闭所有不必要的应用程序，然后再试一次。

---

**标记导入期间发生文件异常。**

**错误类型：**

警告

**可能的原因：**

无法读取变量导入文件。

**可能的解决方案：**

重新生成变量导入文件。

---

**解析导入文件中的记录出错。| 记录编号 = <编号>, 字段 = <编号>。**

**错误类型：**

警告

**可能的原因：**

无法解析变量导入文件中的指定字段，因为其长于预期长度或无效。

**可能的解决方案：**

编辑变量导入文件以更改违规字段。

---

**针对导入文件中的记录截断了说明。| 记录编号 = <编号>。**

**错误类型：**

警告

**可能的原因：**

指定记录中的标记说明过长。

**可能的解决方案：**

驱动程序会根据需要截断说明。为了防止此错误，请编辑变量导入文件以缩短说明。

---

**导入的标记名称无效，且已经更改。| 标记名称 = '<标记>', 更改后的标记名称 = '<标记>'。**

**错误类型：**

警告

**可能的原因：**

在变量导入文件中发现的标记名称包含无效字符。

**可能的解决方案：**

驱动程序会根据变量导入文件构造有效的名称。为防止此错误并保持名称的一致性，请更改导出变量的名称。

---

**由于不支持数据类型，因此无法导入标记。| 标记名称 = '<标记>', 不支持的数据类型 = '<类型>'。**

**错误类型：**

警告

**可能的原因：**

变量导入文件中的数据类型不是此驱动程序支持的类型之一。

**可能的解决方案：**

将变量导入文件中的数据类型更改为支持的类型。如果变量用于结构，请手动编辑文件以定义结构所需的每个标记或在服务器中手动配置所需标记。

**也可以看看：**

从概念中导出变量

---

**无法写入设备上的地址。电路板响应异常。| 地址 = '<地址>', 异常 = <代码>。****错误类型：**

警告

**可能的原因：**

1. 适配器可能不存在。
2. 取决于提供的错误代码。

**可能的解决方案：**

验证是否已在通道属性中选择正确的适配器编号。使用 SyCon 来确定适配器排序。

**注意：**

不适用于 SA85 卡。Hilscher CIF 卡的代码为 -1、-33。

**也可以看看：**

SyCon 用户手册

---

**无法从设备上的地址读取。板响应异常。| 地址 = '<地址>', 异常 = <代码>。****错误类型：**

警告

**可能的原因：**

1. 适配器可能不存在。
2. 取决于提供的错误代码。

**可能的解决方案：**

验证是否已在通道属性中选择正确的适配器编号。使用 SyCon 来确定适配器排序。

**注意：**

不适用于 SA85 卡。Hilscher CIF 卡的代码为 -1、-33。

**也可以看看：**

SyCon 用户手册

---

**启动了 MBPLUS.SYS 设备****错误类型：**

信息化

---

**正在导入标记数据库。| 源文件 = '<文件名>'****错误类型：**

信息化

## Modbus 异常代码

以下数据来自 Modbus 应用程序协议规范文档。

代码 (十进制/ 十六进制)	名称	含义
01/0x01	ILLEGAL FUNCTION	在查询中接收的功能代码不是服务器允许的操作。这可能是因为功能代码仅适用于较新的设备，而未在选定单元中实施。该代码也可以指示服务器(或服务器设备)处理此类型请求的状态错误；例如服务器未配置但被要求返回寄存器值。
02/0x02	ILLEGAL DATA ADDRESS	在查询中接收的数据地址不是服务器允许的地址。更具体地说，参考数字和传输长度的组合是无效的。对于具有 100 个寄存器的控制器，采用偏移为 96 且长度为 4 的请求时会成功。采用偏移为 96 且长度为 5 的请求时将发生异常 02。
03/0x03	ILLEGAL DATA VALUE	包含于查询数据字段中的值不是服务器的允许值。这指示了复杂请求中其余部分结构的故障，例如隐含长度不正确。具体而言，它并不意味着提交用于存储在寄存器中的数据项的值超出应用程序预期，因为 Modbus 协议不知道任何特定寄存器的任何特定值的重要性。
04/0x04	SERVER DEVICE FAILURE	服务器在尝试执行请求的操作时发生不可恢复的错误。
05/0x05	ACKNOWLEDGE	服务器已接受请求并正在处理，但需要很长时间才能完成。返回此响应是为了防止客户端发生超时错误。客户端接下来可以发出轮询程序完成消息，从而确定是否已完成处理。
06/0x06	SERVER DEVICE BUSY	服务器正在处理持续时间较长的编程命令。当服务器设备空闲时，客户端稍后应重新发送消息。
07/0x07	NEGATIVE ACKNOWLEDGE	服务器不能执行查询中接收到的编程功能。使用功能代码 13 或 14 (十进制)的编程请求未成功时，将返回此代码。客户端应请求来自服务器的诊断或错误信息。
08/0x08	MEMORY PARITY ERROR	服务器尝试读取扩展内存，但检测到内存中的奇偶性错误。客户端可以重试请求，但服务器设备可能需要服务。
10/0x0A	GATEWAY PATH UNAVAILABLE	专门与网关结合使用，指示网关无法分配从输入端口到输出端口的内部通信路径来处理请求。通常，这意味着网关配置错误或过载。
11/0x0B	GATEWAY TARGET DEVICE FAILED TO RESPOND	专门与网关结合使用，指示没有从目标设备获得响应。通常，这意味着设备未存在于网络上。

注意：对于此驱动程序，术语“服务器”和“主动提供”可交替使用。

# 索引

“

“标识” 7

## B

BCD 25

## I

ID 10

## L

LBCD 25

## M

MBPLUS 5

MBX 5

Modbus 寻址 26

Modbus 异常代码 36

Modbus 字节顺序 20

Modicon 5

Modicon PLC 20

Modicon SA85 网卡 5

MSTR 11

MSTR 指令 13

## P

PCI-85 5

ProWORX 23

ProWORX 编程应用程序 23

## S

SA8 5

SA85 卡 6

Schneider 5

## T

TIO 模块 18

TIO 模块寻址 30

## 包

包括说明 19

## 保

保持寄存器 18, 20, 26

## 变

变量导入设置 18

变量导入文件 18, 23

## 标

标记导入期间发生文件异常。 34

标记导入由于内存资源不足而失败。 33

标记计数 7

标记生成 16

标识 10

## 不

不良数组。| 数组范围 = <开始> 至 <结束>。 31

不扫描, 仅按需求轮询 14

## 布

布尔型 25

## 超

超时前的尝试次数 15

## 创

创建 17

## 从

从零开始的位寻址 20

从零开始的寻址 20

从设备读取全局数据时出错。 32

## 打

打开 MBPLUS 路径时出错。| 路径 = '<路径>'。 32

## 单

单个网络 12

单通道 24

## 导

导入标记数据库时打开文件出错。| OS 错误 = '<错误>'。 32

导入的标记名称无效，且已经更改。| 标记名称 = '<标记>'，更改后的标记名称 = '<标记>'。 34

## 地

地址说明 26

## 定

定时 14

## 读

读取保持寄存器 28

读取内部寄存器 28

读取输入状态 28

读取线圈状态 28

## 短

短整型 25

## 对

对于重复标记 16

## 多

多个通道 24

## 非

非规范浮点数处理 8

非主动 6, 11

## 封

封装线圈 27

## 服

服务器 11

## 浮

浮点数 25

浮点型 25

## 父

父组 17

## 覆

覆盖 16



## 概

概念 23

概述 5

## 功

功能代码 26

功能代码说明 28

功能码 05 21

功能码 06 20

## 故

故障时降级 15

## 降

降级超时 15

降级期间 15

降级时放弃请求 15

## 接

接口卡 6

## 解

解析导入文件中的记录出错。| 记录编号 = <编号>, 字段 = <编号>。 34

## 警

警告

全局数据已禁用, 访问需要 Modicon 的 4.0 低级系统驱动程序。 33

## 客

客户端 11

## 控

控制块 13

## 块

块大小 17

块地址请求响应异常。| 块范围 = H<开始> 至 H<结束>, 异常 = <代码>。 33

块读取字符串 18

块中的地址错误。| 块范围 = <开始> 至 <结束>。 31

块中的地址错误。| 块范围 = H<开始> 至 H<结束>。 31

## 来

来自缓存的初始更新 14

## 连

连接超时 15

## 轮

轮询 6

## 名

名称 10

## 模

模拟 10

## 内

内部寄存器 18, 26

## 配

配置全局数据通信设备 29

## 启

启动了 MBPLUS.SYS 设备 35

## 强

强制单个线圈 28

强制多个线圈 28

## 桥

桥接网络 13

## 请

请求超时 15

## 驱

驱动程序 10

## 全

全局数据 29

## 冗

冗余 22

## 扫

扫描模式 14

## 删

删除 16

## 设

设备 ID (PLC 网络地址) 11

设备间延迟 8

设备启动时 16

设备上的块请求响应异常。| 块范围 = <开始> 至 <结束>, 异常 = <代码>。 32

设备属性 - 标记生成 16

设备属性 - 定时 14

设备属性 - 冗余 22

设备属性 - 自动降级 15

设置 6, 20

## 生

生成 16

## 十

十进制 26, 28, 30

十六进制 27-28, 30

## 使

使用 Modicon 位排序 21

## 事

事件日志消息 31

## 适

适配器 8

适配器编号 9

## 收

收到的块长度与预期的长度不匹配。| 收到的长度 = <数字> (字节), 需要的长度 = <数字> (字节)。 32

## 首

首双字排序为低 21

首字排序为低 21

## 输

输出线圈 17, 26

输入线圈 17, 26

## 属

属性更改时 16

## 数

数据编码 21

数据客户端 11

数据库创建 23

数据类型说明 25

数据收集 10

数组支持 28

## 双

双精度 25

双字 25

## 说

说明 10

## 锁

锁存 30

## 替

替换为零 8

## 通

通道分配 10

通道属性 - 常规 6  
通道属性 - 高级 8  
通道属性 - 写入优化 7  
通信超时 14

## 外

外部依存关系 5

## 未

未修改 8

## 无

无法初始化通道。 31  
无法创建运行此驱动程序所需的系统资源。 31  
无法从设备访问全局数据。 32  
无法从设备上的地址读取。板响应异常。| 地址 = '<地址>', 异常 = <代码>。 35  
无法从设备上的地址读取。设备响应异常。| 地址 = '<地址>', 异常 = <代码>。 33  
无法打开适配器。| 适配器 = <名称>。 33  
无法加载通道。每个 Hilscher 适配器仅允许一个通道。修改项目，使每个通道都具有唯一的适配器，然后重新加载。 32  
无法检测卡或启动 Modbus Plus 服务。请验证是否已正确安装卡和 MBP \*.sys 驱动程序。 31  
无法启动 MBPLUS.SYS 设备。 31  
无法写入设备上的地址。电路板响应异常。| 地址 = '<地址>', 异常 = <代码>。 35  
无法写入设备上的地址。设备响应异常。| 地址 = '<地址>', 异常 = <代码>。 33  
无符号 25

## 项

项目 24

## 写

写入非布尔标记的最新值 8  
写入所有标记的所有值 7  
写入所有标记的最新值 8

## 型

型号 10

## 性

性能 24

## 寻

寻址, 5 位 26

寻址, 6 位 26

## 已

已屏蔽的写入寄存器 29

## 优

优化方法 7

优化通信 24

## 由

由于不支持数据类型, 因此无法导入标记。| 标记名称 = '<标记>', 不支持的数据类型 = '<类型>'。 34

## 邮

邮箱 6, 11

邮箱模式 11, 28

## 有

有符号 25

## 预

预设单个寄存器 28

预设多个寄存器 28

## 允

允许子组 17

## 占

占空比 8

## 长

长整型 25

## 针

针对导入文件中的记录截断了说明。| 记录编号 = <编号>。 34

## 诊

诊断 7

## 正

正在导入标记数据库。| 源文件 = '<文件名>' 35

## 支

支持 6

## 只

只写访问 28

## 主

主动模式 11

主动提供 6

## 自

自动标记数据库生成 23

自动降级 15



## 字

字 25

字符串 25

字符串支持 28

字节顺序 21

## 遵

遵循标签指定的扫描速率 14