Modbus TCP/IP Ethernet 驱动程序

© 2025 PTC Inc. 保留所有权利。

目录

Modbus TCP/IP Ethernet 驱动程序 1
目录 2
Modbus TCP/IP Ethernet 驱动程序 5
概述 5
支持的设备型号 6
设置 7
通道属性 - 常规 8
标记计数 8
通道属性 - 以太网通信 9
通道属性 - 写入优化 9
通道属性 - 高级 10
通道属性 - 通信序列化 10
通道属性 - 以太网 11
设备属性 - 常规 14
设备属性 - 扫描模式 15
设备属性 - 定时 16
设备属性 - 自动降级 16
设备属性 - 标记生成 17
设备属性 - 变量导入设置 19
设备属性 - 非主动请求 19
Modbus 客户端和 Modbus 服务器注意事项 20
设备属性 - 错误处理 21
设备属性 - 以太网 22
设备属性 - 设置 22
设备属性 - 块大小 25
设备属性 - 冗余 27
Configuration API Modbus Ethernet 示例 28
枚举 29
设备型号枚举 30
自动标记数据库生成 31
从自定义应用程序导入 31
优化通信 32
数据类型说明 34
地址说明 35
驱动程序系统标记寻址 35
功能代码说明 35

Applicom 子模型和寻址 36
类属 Modbus 寻址 36
TSX Quantum 40
TSX Premium 43
CEG 寻址 45
Fluenta 寻址 45
Instromet 寻址 46
邮箱寻址 46
Modbus 寻址 46
Roxar 寻址 49
统计信息项 50
事件日志消息 52
无法启动 winsock 通信。 52
无法启动非主动请求通信。 52
针对未定义设备的非主动请求邮箱访问。正在关闭套接字。| IP 地址 = '<地址>'。 52
非主动请求邮箱接收到不支持的请求。| IP 地址 = '<地址>'。 52
非主动请求邮箱内存分配错误。| IP 地址 = '<地址>'。 53
无法创建套接字连接。 53
导入标记数据库时打开文件出错。| OS 错误 = '<错误>'。 53
不良数组。| 数组范围 = <开始> 至 <结束>。 53
块中的地址错误。| 块范围 = <地址> 至 <地址>。 54
无法解析主机。| 主机名 = '<名称>'。 54
指定的输出线圈块大小超过了最大块大小。| 指定的块大小 = <数字> (线圈)，最大块大小 = <数字> (线圈)。 54
指定的输入线圈块大小超过了最大块大小。| 指定的块大小 = <数字> (线圈)，最大块大小 = <数字> (线圈)。 54
指定的内部寄存器块大小超过了最大块大小。| 指定的块大小 = <数字> (寄存器)，最大块大小 = <数字> (寄存器)。 54
指定的保持寄存器块大小超过了最大块大小。| 指定的块大小 = <数字> (寄存器)，最大块大小 = <数字> (寄存器)。 54
块请求响应异常。| 块范围 = <地址> 至 <地址>，异常 = <代码>。 55
块请求响应异常。| 块范围 = <地址> 到 <地址>，功能码 = <代码>，异常 = <代码>。 55
接收到无效块长度。| 块范围 = <开始> 至 <结束>。 55
标记导入由于内存资源不足而失败。 55
标记导入期间发生文件异常。 56
解析导入文件中的记录时出错。| 记录编号 = <数字>，字段 = <字段>。 56
针对导入文件中的记录截断了说明。| 记录编号 = <编号>。 56
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>'。 56
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>'。 57

无法写入地址，设备响应异常。| 地址 = '<地址>'，异常 = <代码>。 57
以太网管理器已启动。 57
以太网管理器已停止。 57
正在导入标记数据库。| 源文件 = '<文件名>'。 57
客户端应用程序已通过系统 tag _CEGExtension 更改 CEG 扩展。| 扩展 = '<扩展>'。 57
正在启动非主动请求通信。| 协议 = '<名称>'，端口 = <数字>。 58
为 Modbus 服务器设备创建了内存。| Modbus 服务器设备 ID = <设备>。 58
所有通道都已订阅虚拟网络，或者所有设备均在侦听远程地址，正在停止非主动请求通信。 58
通道处于虚拟网络中，所有设备都会恢复到“每台设备使用一个套接字”。 58
与客户端相连时，无法将设备 ID 从 Modbus 客户端模式更改为服务器模式。 58
与客户端相连时，无法将设备 ID 从 Modbus 服务器模式更改为客户端模式。 58
当通道处于虚拟网络中时，不允许使用 Modbus 服务器模式。设备 ID 无法包含环回或本地 IP 地址。 58
当通道处于虚拟网络中时，不允许使用邮箱模型。 58
Modbus 异常代码 59
Modbus Ethernet 通道属性 60
Modbus Ethernet 设备属性 60
Modbus Ethernet 标记属性 61
索引 62

Modbus TCP/IP Ethernet 驱动程序

帮助版本 1.158

目录

概述
什么是 Modbus TCP/IP Ethernet 驱动程序？

设置
如何配置使用此驱动程序的通道和设备？

通过 API 进行配置
如何使用 Configuration API 配置通道和设备？

自动生成标记数据库
如何配置 Modbus TCP/IP Ethernet 驱动程序 的标记？

优化通信
如何从 Modbus TCP/IP Ethernet 驱动程序 获得最佳性能？

数据类型说明
Modbus TCP/IP Ethernet 驱动程序 支持哪些数据类型？

地址说明
如何引用 Modbus Ethernet 设备中的数据位置？

事件日志消息
Modbus TCP/IP Ethernet 驱动程序 会产生哪些消息？

概述

Modbus TCP/IP Ethernet 驱动程序 提供将 Modbus Ethernet 设备连接至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。要使用此驱动程序，用户必须正确安装 TCP/IP。*有关设置的详细信息，请参阅 Windows 文档。*

注意：在操作期间发生故障时，驱动程序将发送消息。

支持的设备型号

Applicom

该型号支持类属 Modbus、TSX Premium 和 TSX Quantum 设备的 Applicom 寻址语法。

以太网到 Modbus Plus 网桥

驱动程序还可通过"以太网到 Modbus Plus 网桥"与 Modbus Plus 设备进行通信。所使用的设备 ID 应为网桥的 IP 地址以及 Modbus Plus 网桥索引。例如，网桥 IP 205.167.7.12、网桥索引 5 相当于 205.167.7.12.5 的设备 ID。有关获取并设置 MBE 到 MBP 网桥的信息，请联系 Modicon/Schneider Automation 分销商。

CEG

该型号支持 CEG 设备的扩展块大小。

Fluenta

该型号支持 Fluenta FGM 100/130 流量计算机的非标准 Modbus 映射。

Instromet

该型号支持 Instromet 设备的非标准 Modbus 映射。

邮箱

该模型将影响非主动请求的处理方式。通过定义邮箱设备，驱动程序所起的作用与网络上的 PLC 不同。相反，它将作为所定义的每个邮箱设备的存储区域。驱动程序接收非主动请求的命令后，驱动程序会检测到发出消息的 IP 地址，并将数据置于为该设备分配的存储区域中。如果发出消息的设备 (带有 IP 地址) 尚未定义为邮箱设备，则不会处理该消息。读取或写入此类型设备的所有客户端应用程序将读取或写入驱动程序中的存储区域，而非物理设备。

有关将非主动请求发送到 Modbus TCP/IP Ethernet 驱动程序 的信息，请参阅关于 MSTR 说明的 Modicon 文档。

注意：Modbus 邮箱不支持功能代码 22 (0x16)。仅支持 0x10 (保持寄存器写入多次) 和 0x6 (保持寄存器写入单次)。用户可通过禁用设备属性中的"保存寄存器位写入"来写入单个位。这将强制其使用"读取/修改/写入"序列，而不是直接写入该位。只有客户端 Modbus 设备 (不是邮箱) 必须更改其设置才能解决此问题。

邮箱设备型号的邮箱客户端权限

Modbus 客户端

大多数项目被配置为 Modbus 客户端。在此模式下，驱动程序将访问物理设备 (如 TSX Quantum 或任何其他 Modbus 开放以太网兼容设备)。

Modbus 非主动请求或服务器模式

当 Modbus 为选定型号且配置了等同于主机 IP 地址的设备 ID 时，Modbus TCP/IP Ethernet 驱动程序 将用作网络上的设备。驱动程序接受收到的所有非主动请求的命令，并尝试将其视为另一 PLC 对其进行处理。网络上的所有 Modbus 客户端均可使用其 IP 地址与此模拟设备进行通信。

Modbus 服务器设备的设备 ID 被指定为 *YYY*。*YYY*。*YYY*。*YYY*。*XXX*。*YYY* 可以是运行驱动程序的 PC 的环回地址或本地 IP 地址。*XXX* 指定 Modbus 服务器的工作站 ID，且可以在 0 到 255 的范围内。

多个 Modbus 服务器设备可以具有相同的工作站 ID。在这种情况下，共享工作站 ID 的所有设备都将指向一个公共模拟设备。如果远程 Modbus 客户端从不存在的 Modbus 服务器设备 (工作站 ID) 请求数据，则响应将包含来自工作站 0 的数据。在项目中创建 Modbus 服务器设备后，该 Modbus 服务器随即启用，且将在服务器关闭之前一直保持启用状态。更改工作站 ID 可启用新的 Modbus 服务器设备，该设备在服务器关闭之前一直处于启用状态。

针对输出线圈、输入线圈、内部寄存器和保存寄存器使用地址 1 到 65536。在非主动请求模式下，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。此外，在此驱动程序中已经实现了环回 (也称为功能代码 08，子代码 00)。这些位置可作为分配给 Modbus 服务器设备的标记通过主机 PC 本地访问。

注意：非主动请求的设备不支持只写访问权限。

Roxar

该型号支持 Roxar RFM 水切割仪的非标准 Modbus 映射。

另请参阅：*设备型号*和*设备型号枚举*。

设置

通道和设备限制

此驱动程序支持的最大通道数量为 1024。此驱动程序所支持设备的最大数量为每通道 8192 个。

提示：通道级别设置适用于在此通道上配置的所有设备。

注意：Modbus TCP/IP Ethernet 驱动程序 需要 Winsock 1.1 或更高版本。

通信序列化

Modbus TCP/IP Ethernet 驱动程序 支持通信序列化，该功能可用于指定是否每次都将数据传输限制为单通道传输。

有关详细信息，请参阅通信序列化。

注意：

- 当“通道序列化”启用时，非主动请求的通信和**“每个设备的最大套接字数”**属性将被禁用。“邮箱模式”不适用于“通道序列化”。
- 并非所有属性都可用且适用于所有型号。

另请参阅：*使用 Configuration API 命令配置设备*、*含有 Modbus 的 API 示例*

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。

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。

请知悉：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用于 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。

请知悉：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"。

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 - 以太网通信

以太网通信可用于与设备进行通信。

以太网设置

"网络适配器"：指定要绑定的网络适配器。如果留空或选择"默认"，则操作系统将选择默认适配器。

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。

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。此模式会通过在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。

“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 - 通信序列化

服务器的多线程架构使通道能够与设备并行通信。尽管这十分高效，但在存在物理网络限制 (如以太网无线电) 的情况下，通信可能会进行序列化。通信序列化将限制在虚拟网络中每次仅使用一个通道进行通信。

术语“虚拟网络”是指使用同一管线进行通信的通道和相关设备的集合。例如，以太网无线电管线是客户端无线电。使用同一客户端无线电的所有通道均与同一虚拟网络相关联。通道能够以“循环”方式轮流进行通信。默认情况下，通道在向另一通道传递通信前，可处理一个事务。一个事务中可包括一个或多个标记。如果控制通道包含的设备未响应请求，则在事务超时之前，通道无法释放控制权。这会导致虚拟网络中其他通道的数据更新延迟。

通道级别设置

“虚拟网络”：指定通道的通信序列化模式。选项包括“无”和“网络 1 - 网络 500”。默认值为“无”。选项说明如下：

- **“无”**：此选项禁用通道的通信序列化。
- **“网络 1 - 网络 500”**：此选项可指定分配通道的虚拟网络。

“每周期的事务数”：指定通道中可能发生的单一分块/非分块读/写事务的数量。当通道可以进行通信时，将尝试该事务数。有效范围为 1 到 99。默认值为 1。

全局设置

“网络模式”：此属性用于控制委派通道通信的方式。在**“负载平衡”**模式下，每个通道可以逐一轮流进行通信。在**“优先级”**模式下，通道可以根据以下规则 (优先级由高到低) 进行通信：

1. 具有待处理写入操作的通道具有最高优先级。
2. 具有待处理显式读取操作 (通过内部插件或外部客户端接口) 的通道的优先级基于读取的优先级。
3. 扫描读取和其他定期事件 (特定于驱动程序)。

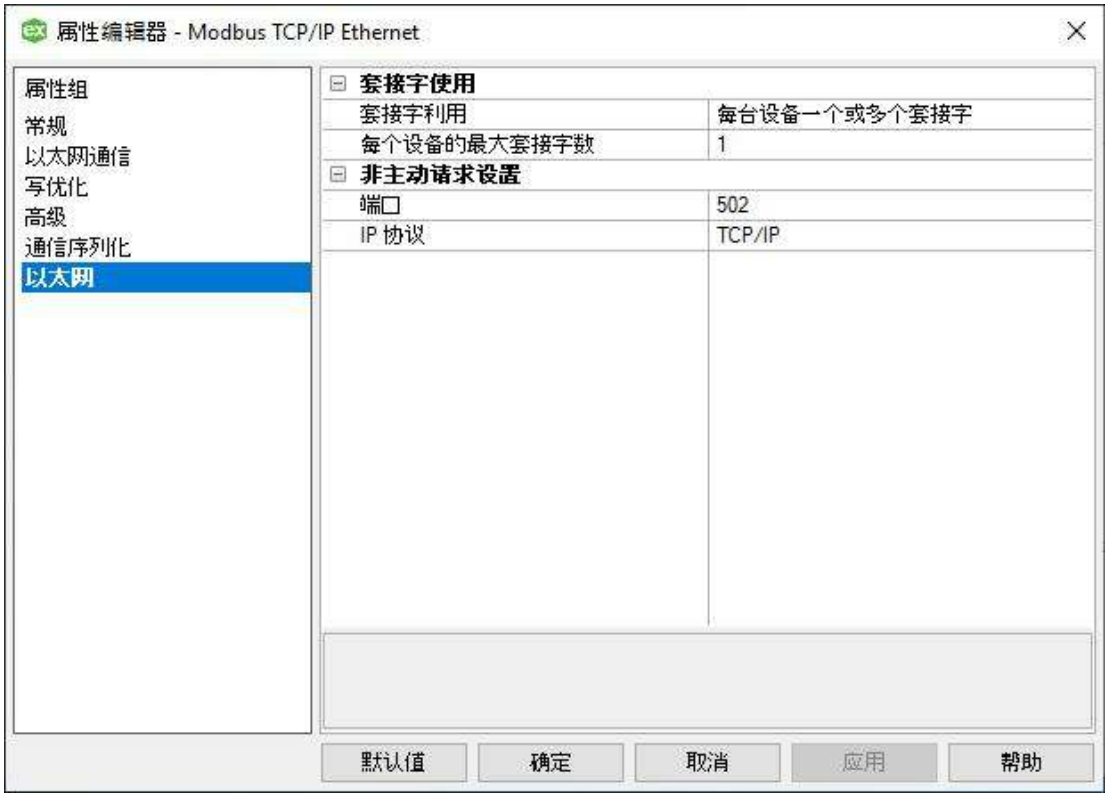
默认设置为“负载平衡”，这并影响*所有*虚拟网络和通道。

依赖于非主动请求响应的设备不应置于虚拟网络中。在必须进行通信序列化的情况下，建议启用“自动降级”。

由于驱动程序的数据读取和写入方式的差异 (如单一、分块或非分块事务)，可能需要调整应用程序的“每周期的事务数”属性。执行此操作时，请考虑以下因素：

- 必须从每个通道读取多少标记？
- 数据写入各个通道的频率如何？
- 通道使用串行驱动程序还是以太网驱动程序？
- 驱动程序是读取单独请求中的标记还是读取块中的多个标记？
- 设备的定时属性 (如请求超时和 *x* 次连续超时后失败) 是否针对虚拟网络通信媒介进行了优化？

通道属性 - 以太网



"套接字使用"(Socket Usage)

"套接字利用"：指定驱动程序是应针对此通道上的所有设备使用单一套接字还是应使用多个套接字与设备进行通信。

- **"每台设备一个或多个套接字"**：指定驱动程序针对网络上的每台设备使用一个或多个套接字，并将此套接字保持为活动连接。这是默认设置和行为。
- **"每个通道一个套接字"**：指定驱动程序仅使用单一套接字为通道中的所有设备提供通信。

"每个设备的最大套接字数"：指定设备可用的最大套接字数量。默认值为 1。

有关这些设置如何影响性能的详细信息，请参阅优化通信。

非主动请求的设置

非主动请求设置同时适用于 Modbus 客户端和 Modbus 服务器模式。

"端口"：指定驱动程序在监听非主动请求时使用的端口号。有效范围为 0 到 65535。默认值为 502。

"IP 协议"：指定驱动程序在监听非主动请求时使用的协议。选项包括"用户数据报协议"(UDP) 或"传输控制协议"(TCP/IP)。默认值为 TCP/IP。

注意：

1. 当任何外部 IP 用作设备 ID 时，都会启用 Modbus 客户端模式。当驱动程序处于客户端模式时，它可以接受非主动请求设置。一旦被 OPC 服务器加载，驱动程序就会启动非主动请求数据的监听线程。此线程是 OPC 服务器中配置的所有通道的全局线程。例如，OPC 服务器项目定义了三个通道，如果更改了其中一个通道的任何一个设置，其他两个通道的设置也会发生同样的更改。在应用更改后，监听线程会重新启动。事件日志会发布一个重新启动事件。

2. Modbus 服务器模式在以下情况下启用：型号设为 Modbus 且设备 ID 设为 IP_Address.yyy，其中 IP_Address 可以是通道中分配的网络适配器的本地 IP 地址，也可以是运行驱动程序的 PC 的本地环回 IP 地址。例如，127.0.0.1.1 或 127.xxx.xxx.xxx，其中 xxx= 0-255，yyy (站 ID) = 0-255。驱动程序处于 Modbus 服务器模式时，一旦被 OPC 服务器加载，驱动程序就会启动非主动请求数据的监听线程。此线程监听单一端口，是 OPC 服务器中配置的所有通道的全局线程。一个服务器项目中的多个设备可设为 Modbus 服务器，但项目中的所有 Modbus 服务器设备都可以使用唯一的设备 ID 单独进行访问，而同一项目中的所有 Modbus 服务器都在同一端口上监听或被访问。

有关非主动请求模式的详细信息，请参阅 ***Modbus 非主动请求或服务器模式****。*

设备属性 - 常规

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。
有关特定设备型号的详细信息，请参阅支持的设备型号。

"型号"：设备的特定版本。

ID：指定设备 IP 地址和以太网上的 Modbus 网桥索引。设备 ID 被指定为 <*主机*>.*XXX*，其中*主机* 是标准的 UNC/DNS 名称或 IP 地址。*XXX* 用于指定设备的 Modbus 服务器 ID，且其范围可介于 0 到 255 之间。如果未使用网桥，则应将索引设为 0。设备可配置为客户端或服务器，具体取决于型号和设备 ID。
*有关非主动请求模式的详细信息，请参阅 **Modbus 非主动请求或服务器模式**。*

示例

1. 当向 IP 地址为 205.167.7.19 的 Modicon TSX Quantum 设备请求数据时，设备 ID 应为 205.167.7.19.0。
2. 当向 IP 地址为 205.167.7.50 且与 Modbus Ethernet 网桥的网桥索引 5 连接的 Modbus Plus 设备请求数据时，设备 ID 应为 205.167.7.50.5。

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。通过此属性或设备系统标记可随时更改此属性。

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"。

请知悉：

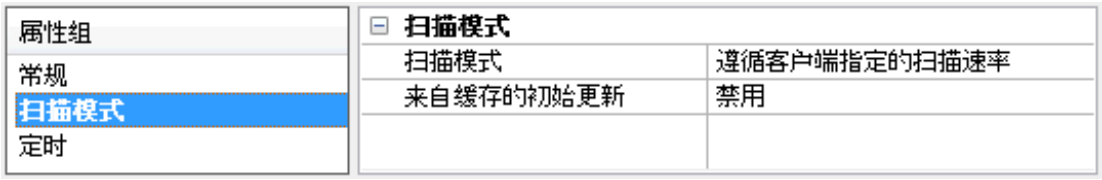
1. "系统"标记 (_Simulated) 为只读且无法写入，从而达到运行时保护的目的。"系统"标记允许从客户端监控此属性。
2. 在"模拟"模式下，项的内存映射取决于客户端更新速率 (OPC 客户端的"组更新速率"或本机和 DDE 接口的扫描速率)。这意味着，参考相同项、而采用不同更新速率的两个客户端会返回不同的数据。

"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。

另请参阅：使用 Configuration API 命令配置设备、含有 Modbus 的 API 示例

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。



"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。

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。

请知悉：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。

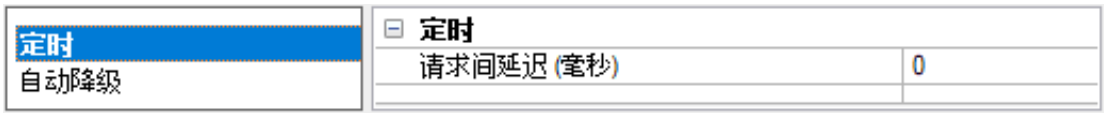
“请求超时”：指定一个所有驱动程序使用的间隔来决定驱动程序等待目标设备完成响应的时间。有效范围是 50 到 9999999 毫秒 (167 分钟)。默认值通常是 1000 毫秒，但可能会因驱动程序而异。大多数串行驱动程序的默认超时是基于 9600 波特或更高的波特率来确定的。当以较低的波特率使用驱动程序时，请增加超时，以补偿获取数据所需增加的时间。

“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 到 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。

请知悉：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。



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。

"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。

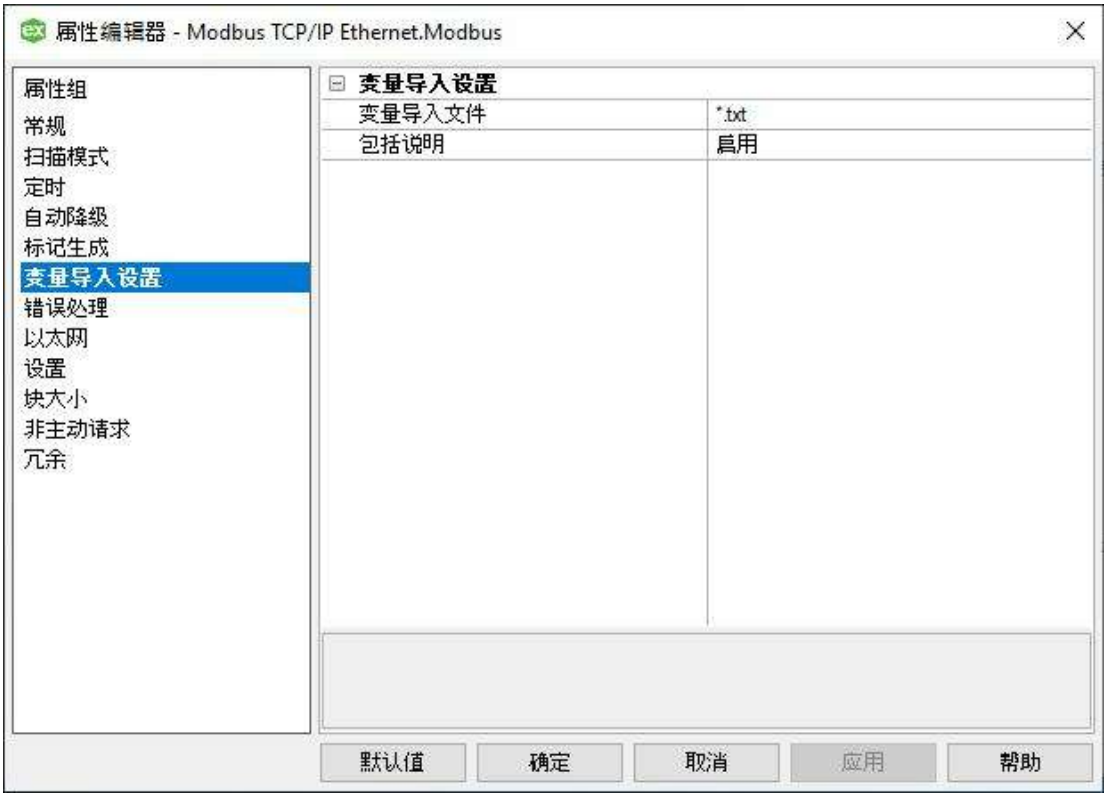
注意：如果在服务器生成标记的过程中，分配给标记的名称与现有标记的名称相同，则系统会自动递增到下一个最高数字，以免标记名称发生重复。例如，如果生成过程中创建了名为 "AI22" 的标记且该名称已存在，则会将标记创建为 "AI23"。

"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 - 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。

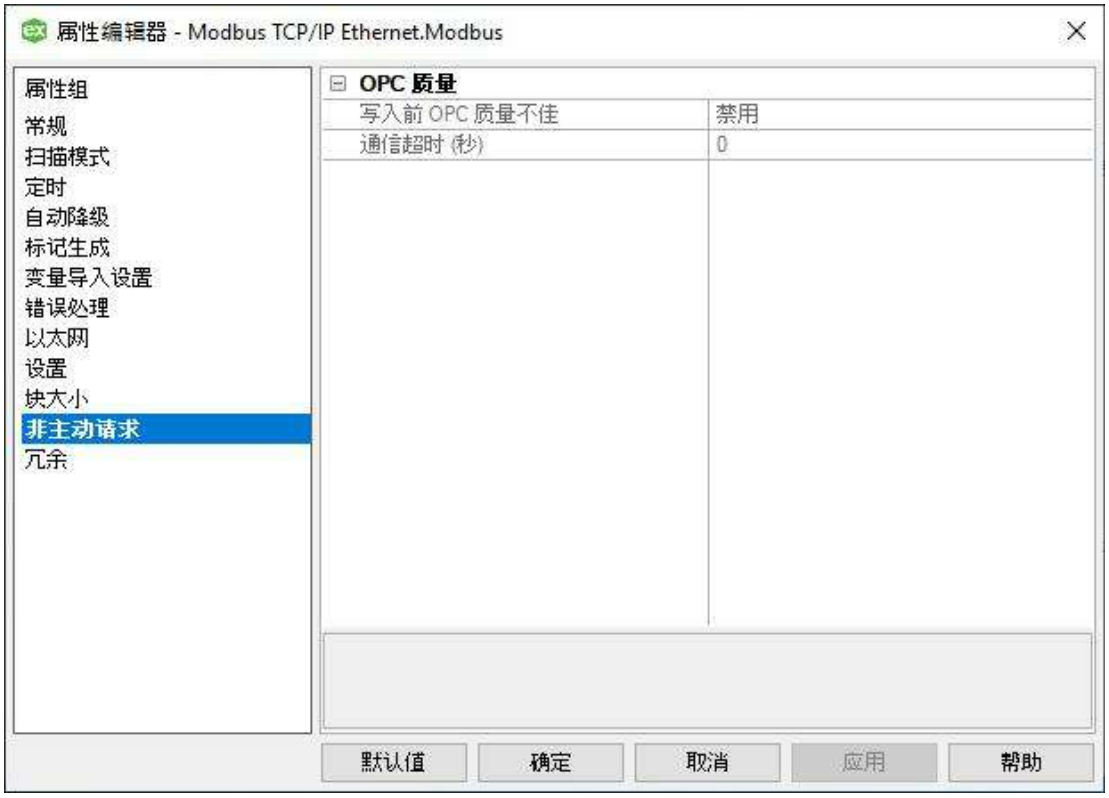


"变量导入文件"：该参数指定启用自动生成标记数据库功能时驱动程序所用变量导入文件的确切位置。

"包括说明"(Include Descriptions)：启用时，该选项可导入标记说明 (如果文件中存在)。

有关配置自动标记数据库生成功能 (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。

设备属性 - 非主动请求



OPC 质量

"写入前 OPC 质量不佳"：控制附加到该驱动程序的标记的初始 OPC 质量。禁用此选项后，所有标记的初始值均为 0，且 OPC 质量良好。此为默认条件。启用此选项后，所有标记的初始值均为 0，且 OPC 质量不佳。除非 Modbus 客户端或客户端应用程序写入所有被标记参考的线圈或寄存器，否则标记质量仍然很低。例如，地址为 400001、数据类型为"双字型"的标记参考两个保持寄存器：400001 和 400002。只有写入这两个保存寄存器后，此标记的质量才会变好。

注意：如果设备未处于服务器模式下，则此选项将灰显。

"通信超时"(Communications Timeout)：设置驱动器在将设备标记质量设置为"不佳"之前等待传入请求的时长，单位为秒。发生超时后，重置超时并允许正常处理所有标记的唯一方法是重新建立与远程客户端的通信或通过将其设置为 0 来禁用通信超时。启用后，有效范围为 1 到 64,800 秒 (18 小时)。

注意：

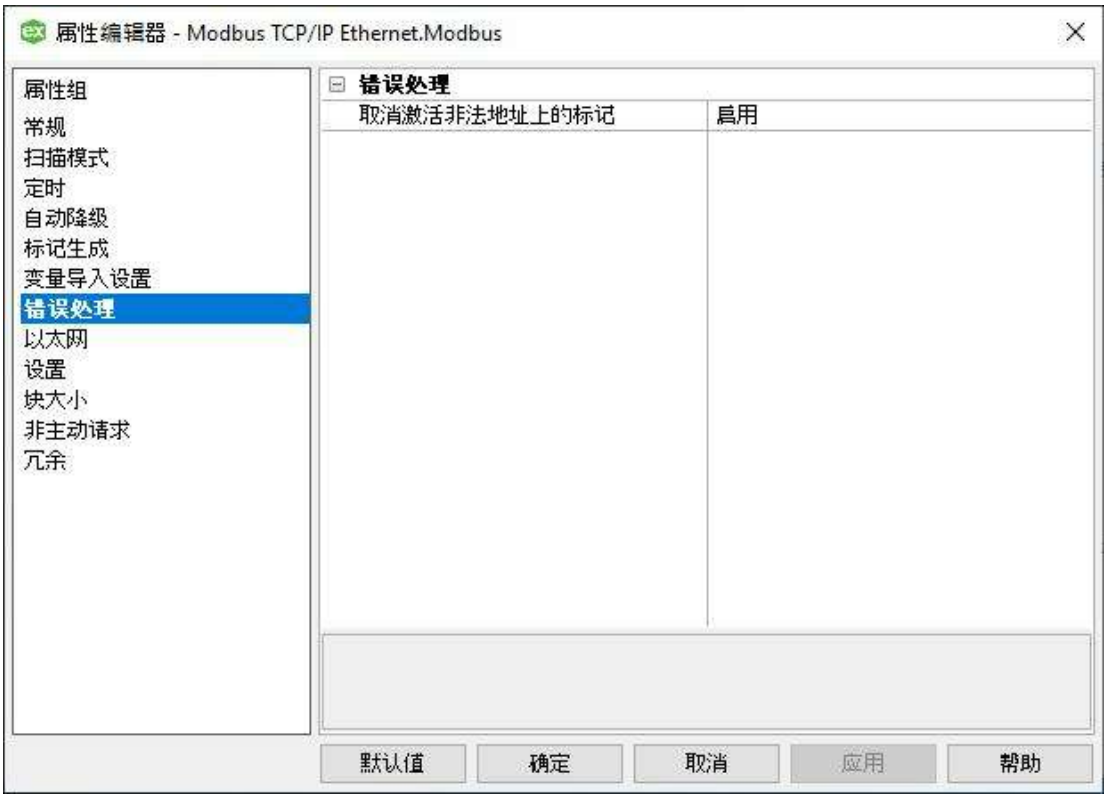
1. 如果传入请求到不存在的服务器设备 (站 ID)，则请求将定向到站 0。在这种情况下，即使在超时期间没有明确收到任何远程通信，站 ID 为 0 的服务器设备超时也不会发生。
2. 服务器设备要求型号为 Modbus 且设备 ID 为 *IP_Address.yyy*，其中 *IP_Address* 可以是通道中分配的网络适配器的 IP 地址，或运行驱动程序的 PC 的本地 IP 回环 IP 地址。例如，127.0.0.1.1 或 127.xxx.xxx.xxx，其中 *xxx*= 0-255，*yyy* (站 ID) = 0-255。
3. 当收到第一个服务器设备的非主动请求时，事件日志将显示以下信息性消息："<日期>__<时间>__<级别>__<源>__<事件>"。例如，"2/4/2011__4:53:10 PM__信息__Modbus TCP/IP Ethernet__为服务器设备 <服务器编号> 创建的内存"。

Modbus 客户端和 Modbus 服务器注意事项

下列注意事项适用于 Modbus 客户端和 Modbus 服务器设备。

- 不建议邮箱设备和 Modbus 设备在同一台计算机上。由于客户端一次仅从其中一台设备中获取数据，因此不确定其具体从哪一台设备获取数据。
- 建议将客户端和服务器设备置于服务器项目中的单独通道上，以达到处理服务器设备标记的最佳效果。
- 连接 OPC 客户端后，仅在不会导致更改设备模式 (客户端到服务器，或服务器到客户端) 的情况下，才可更改设备 ID。通过将环回地址或本地 IP 地址更改为其他不同的 IP 地址来更改模式，反之亦然。(运行驱动程序的 PC 的) 环回地址和本地 IP 地址表示服务器 (非主动请求) 模式，而任何其他 IP 地址表示设备的客户端模式。未连接 OPC 客户端时，可以任意更改模式 (例如，客户端到客户端、客户端到服务器、服务器到服务器或服务器到客户端)。
  注意：格式为 127.xxx.xxx.xxx 的所有地址均为环回地址，其中 *xxx* 在 0 到 255 范围内。
- 客户端和服务器设备中的“数据编码”组设置必须相同。例如，当配置为 Modbus 客户端的设备与设置为 Modbus 服务器的设备进行通信时。
- 服务器项目整体上最多允许 255 个服务器设备，均有唯一的服务器 ID。同一服务器 ID 不能跨多个通道使用。
- 服务器会将任何环回地址 (127.x.x.x) 或 localhost IP 视为对自身引用并创建服务器 ID 专用的共享内存空间。多个通道中相同的 ID 为使用相同寄存器内存的相同服务器设备。
- 如果项目中必须多次使用相同的服务器 ID，请选择与相同服务器设备 ID 的其他实例不重合的标记地址范围。多个通道/设备如果在同一服务器 ID 中使用相同标记地址范围，会发生串扰和数据损坏。
- 对于此驱动程序，术语“服务器”和“非主动请求”可交替使用。

设备属性 - 错误处理



“取消激活非法地址上的标记”：选择“启用”可让驱动程序在设备读取某个数据块的响应过程中返回 Modbus 异常代码 2 (非法地址) 或 3 (点数之类的非法数据) 时，停止轮询该数据块。选择“禁用”可让驱动程序继续轮询该数据块而不管发生了什么错误。默认设置为已启用。

设备属性 - 以太网

"端口": 指定配置远程使用的端口号。有效范围为 0 到 65535。默认值为 502。当设备对象处于 Modbus 客户端模式并向设备发出主动请求时，将使用该端口号。

如果使用了端口系统标记，则端口号设置会更改。有关详细信息，请参阅驱动程序系统标记地址。

"IP 协议": 指定驱动程序应使用用户数据报协议(UDP) 还是传输控制协议(TCP/IP) 连接到远程设备。客户端和服务器设置必须匹配。例如，如果服务器 IP 协议设置为 TCP/IP，则该设备的客户端 IP 协议设置也必须为 TCP/IP。

注意: 此驱动程序需要 Winsock V1.1 或更高版本。

"超时后关闭套接字": 指定驱动程序是否应在设备未在超时时间内响应时关闭 TCP 套接字连接。启用时，在默认情况下，驱动程序会在超时后关闭套接字连接。禁用时，驱动程序将继续使用相同的 TCP 套接字，直到接收到的错误、物理设备关闭套接字或驱动程序关闭。

注意: 发生套接字错误时，Modbus TCP/IP Ethernet 驱动程序 会关闭套接字连接。

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 1 开始而不是零，可在定义设备参数时指定该值。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。如果设备不遵守此约定，请选择禁用。默认行为遵守 Modicon PLC 的约定。

"从零开始的位寻址"：在寄存器内，允许字内位的内存类型可作为布尔型引用。寻址符号为 *<address>.<位>*，其中 *<位>* 代表字内的位数。此选项提供了两种对给定字内的位进行寻址的方法；从零开始或从 1 开始。从零开始意味着第一位从 0 开始 (范围 = 0-15)；从 1 开始意味着第一位从 1 开始 (范围 = 1-16)。

"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。某些设备支持一个特殊命令，来操纵寄存器中的单个位 (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。启用后，驱动程序将使用功能码 0x16，与单个寄存器写入操作的设置无关。禁用后，驱动程序将根据是否对单个寄存器写入操作选择了 Modbus 功能码 06 来使用功能码 0x06 或 0x10。默认设置为禁用状态。

请知悉：如果禁用 Modbus 字节顺序，则命令中发送的掩码的字节顺序将为 Intel 字节顺序。

"Modbus 功能码 06"：此驱动程序支持 Modbus 协议功能，以将保持寄存器数据写入到目标设备。在大多数情况下，驱动程序可基于正在写入的寄存器数在功能码 06 和 16 之间进行切换。当写入单个 16 位寄存器时，驱动程序通常使用 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。但是，使用 Modbus 协议的第三方设备很多，并且许多第三方设备仅支持 Modbus 功能码 16 来写入保持寄存器。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，则请禁用此选择。

请知悉：对于字内位写入，"保持寄存器位写入"属性优先于此选项。如果启用了"保存寄存器位写入"，则将使用功能码 0x16，而无需考虑此属性。如果未启用，则功能码 0x06 或 0x10 将用于位内字写入。

“Modbus 功能 05”：此驱动程序支持 Modbus 协议功能，可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，使用这些功能都不是问题。但是，使用 Modbus 协议的第三方设备很多，并且许多第三方设备无论输出线圈数为多少都仅支持使用 Modbus 功能码 15 写入输出线圈。默认情况下将启用此选项，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，则请禁用此选择。

“CEG 扩展”：Modbus 驱动程序能够与支持扩展块大小的 CEG 设备或使用 CEG 型号配置的 Modbus 设备进行通信。此属性仅可用于 CEG 型号。默认设置为“启用”状态，表示该设备是具有扩展块大小的 CEG 设备。“已禁用”表示该设备不支持扩展块大小。

请知悉：当存在启用的 OPC 客户端连接时，可以修改此属性。在此情况下，禁用此选项会导致块大小范围的更改。如果存在任何块大小属性超过了最大值，则它们会自动调整为新的最大值。

“邮箱客户端权限”：Modbus 驱动程序可与邮箱客户端进行通信，选项如下：

- **“内存映射只读”**：客户端应用程序只能从邮箱内存映射读取。
- **“内存映射读-写”**：客户端应用程序可读取和写入邮箱内存映射。
- **“设备写入-内存映射读取”**：客户端应用程序仅可写入设备；并可从内存映射读取。

数据编码

“Modbus 字节顺序”：设置每个寄存器/16 位值的数据编码。使用此选项，字节顺序可以从默认的 Modbus 字节排序更改为 Intel 字节排序。默认设置为启用状态，这是 Modbus 兼容设备的常用设置。如果设备使用 Intel 字节排序，请禁用此属性以读取 Intel 格式的数据。

“首字排序为低”：设置 32 位值的数据编码和 64 位值的双字。Modbus 设备中的两个连续寄存器地址用于 32 位数据类型。驱动程序可根据此选项读取第一个字作为 32 位值的低位字或高位字读取。默认设置为启用状态，即首字排序为低，以遵守 Modicon Modsoft 编程软件的约定。

“首双字排序为低”：设置 64 位值的数据编码。Modbus 设备中的四个连续寄存器地址用于 64 位数据类型。驱动程序可将第一个双字作为 64 位值的低位双字或高位双字读取。默认设置为启用状态，即首双字排序为低，以遵守 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 (已禁用)。

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使用从零开始还是从 1 开始的位寻址。
MSB = 最高有效位
LSB = 最低有效位

启用的 Modicon 位顺序

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

禁用的 Modicon 位顺序

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

“将长整型作为十进制处理”: 启用后，驱动程序会将双精度无符号长整型和双字型数据类型编码和解码为介于 0 到 99999999 之间的值。此格式指定了每个字表示介于 0 到 9999 之间的值。超过指定范围的值读取不受限制，但行为未定义。所有读取值都将使用以下公式进行解码：[读取值] = HighWord * 10000 + LowWord。大于 99999999 的写入值将限制为最大值。所有写入值都使用以下公式进行编码：原始数据 = [写入值]/10000 + [写入值] % 10000。

有关设置的提示

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，请使用以下信息和设备的文档来确定数据编码选项的正确设置。

默认设置对于大多数 Modbus 设备都是可接受的。

数据编码选项	数据编码	
Modbus 字节顺序	高字节 (15..8)	低字节 (7..0)
Modbus 字节顺序	低字节 (7..0)	高字节 (15..8)
首字排序为低	高位字 (31..16) 64 位数据类型中双字的高位字 (63..48)	低位字 (15..0) 64 位数据类型中双字的低位字 (47..32)
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)
首双字排序为低	高位双字 (63..32)	低位双字 (31..0)
首双字排序为低	低位双字 (31..0)	高位双字 (63..32)

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)。默认值为 32。

"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)。默认值为 32。

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 120 个 16 位标准 Modbus 寄存器。默认值为 32。

"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 120 个 16 位标准 Modbus 寄存器。默认值为 32。

块

"块读取字符串": 启用通常单独读取的字符串标记的组/块读取。字符串标记根据所选块大小分组在一起。只能对 Modbus 型号字符串标记执行块读取。

注意：

1. Instromet、Roxar 和 Fluenta 模型 (支持 32 位和 64 位寄存器) 需要特殊考虑。Modbus 协议将块大小限制为不超过 256 个字节。这将转换为这些型号的 64 个 32 位寄存器或 32 个 64 位寄存器的最大块大小。
2. CEG 型号支持介于 8 到 8000 (8 的倍数) 之间的线圈块大小和介于 1 到 500 之间的寄存器块大小。此型号必须仅用于 CEG 设备。

3. 如果寄存器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”错误。为了防止此错误，请将块大小值减小到 120。
4. 某些设备可能不支持默认大小的块读取操作。小型 Modicon PLC 和非 Modicon 设备可能不支持 Modbus 以太网支持的最大数据传输长度。
5. 某些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，则请求可能会被拒绝。

设备属性 - 冗余

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。

Configuration API Modbus Ethernet 示例

有关通道及设备定义和枚举的列表，请通过 REST 客户端访问以下端点。

通道定义

端点 (GET)：

```
https://<主机名_或_ip>:<端口
>/config/v1/doc/drivers/Modbus%20TCP%2FIP%20Ethernet/channels
```

设备定义

端点 (GET)：

```
https://<主机名_或_ip>:<端口
>/config/v1/doc/drivers/Modbus%20TCP%2FIP%20Ethernet/devices
```

创建 Modbus 通道

端点 (POST)：

```
https://<主机名_或_ip>:<端口>/config/v1/project/channels
```

Body:

```
{ "common.ALLTYPES_NAME": "MyChannel", "servermain.MULTIPLE_TYPES_DEVICE_DRIVER":
"Modbus TCP/IP Ethernet" }
```

另请参阅：附录 A *查看通道属性列表。*

创建 Modbus 设备

端点 (POST)：

```
https://<主机名_或_ip>:<端口>/config/v1/project/channels/MyChannel/devices
```

Body:

```
{ "common.ALLTYPES_NAME": "MyDevice", "servermain.DEVICE_ID_STRING":
"<192.160.0.1>.0", "servermain.MULTIPLE_TYPES_DEVICE_DRIVER": "Modbus TCP/IP
Ethernet" }
```

另请参阅：附录 B *查看设备属性列表。*

设备 ID 更新

使用 REST 客户端的“PUT”命令更新设备 ID。
以下端点示例参考了“demo-project.json”项目配置的“ModbusTCPIP”通道名称和“ModbusDevice”设备名称。

设备 ID 示例

端点 (PUT)：

```
https://<主机名_或_ip>:<端口
>/config/v1/project/channels/ModbusTCPIP/devices/ModbusDevice
```

Body:

```
{ "project_id": <项目_ID_来自_GET>, "servermain.DEVICE_ID_STRING": "<IP 地址>" }
```

创建 Modbus 标记

端点 (POST)：

```
https://<主机名_或_ip>:<端口
>/config/v1/project/channels/MyChannel/devices/MyDevice/tags
```

Body:

```
[ { "common.ALLTYPES_NAME": "MyTag1", "servermain.TAG_ADDRESS": "40001" } {
"common.ALLTYPES_NAME": "MyTag2", "servermain.TAG_ADDRESS": "40002" } ]
```

*另请参阅：*附录 C *查看标记属性列表。*

有关通过 Configuration API 进行项目配置的详细信息，请参阅服务器帮助。

枚举

某些属性 (例如设备型号) 具有映射到枚举的值。要查找有效的枚举和值列表，可通过 content=property_definitions 查询设备端点，也可通过查询文档定义端点来查找。

例如，要在名为 “MyChannel”的通道下查看名为“MyDevice”的设备的属性定义，GET 请求应发送到：

```
https://<主机名_或_ip>:<端口
>/config/v1/project/channels/MyChannel/devices/MyDevice/?content=property definitions
```

属性定义也可用于其他对象，例如通道或标记。

或者，如果在 Configuration API 的设置中启用了，则该驱动程序的通道和设备属性定义也可在以下位置查看：

```
https://<主机名_或_ip>:<端口>/config/v1/doc/drivers/<drivername>/Channels
```

```
https://<主机名_或_ip>:<端口>/config/v1/doc/drivers/<drivername>/Devices
```

数据类型枚举示例

查询标记数据类型的文档端点提供以下枚举：

```
{
   "Default": -1,
   "String": 0,
   "Boolean": 1,
   "Char": 2,
   "Byte": 3,
   "Short": 4,
   "Word": 5,
   "Long": 6,
   "DWord": 7,
   "Float": 8,
   "Double": 9,
   "BCD": 10,
   "LBCD": 11,
   "Date": 12,
```

```
   "LLong": 13,
   "QWord": 14,
   "String Array": 20,
   "Boolean Array": 21,
   "Char Array": 22,
   "Byte Array": 23,
   "Short Array": 24,
   "Word Array": 25,
   "Long Array": 26,
   "DWord Array": 27,
   "Float Array": 28,
   "Double Array": 29,
   "BCD Array": 30,
   "LBCD Array": 31,
   "Date Array": 32,
   "LLong Array": 33,
   " QWord Array": 34
}
```

注意：受支持的数据类型因协议和驱动程序而异。

设备型号枚举

设备型号属性的值映射到以下枚举。下表仅供参考；设备端点处的信息应是完整和最新的信息源：

```
https://<主机名_或_ip>:<端口
>/config/v1/doc/drivers/Modbus%20TCP%2FIP%20Ethernet/Channels
```

```
https://<主机名_或_ip>:<端口
>/config/v1/doc/drivers/Modbus%20TCP%2FIP%20Ethernet/Devices
```

枚举	设备型号
0	Modbus
1	邮箱
2	Instromet
3	Roxar RFM
4	Fluenta FGM
5	Applicom
6	CEG

自动标记数据库生成

此驱动程序支持“自动生成标记数据库”，从而能够自动创建可访问设备梯形图程序所用数据点的标记。标记生成可在服务器项目启动时自动启动，也可在其他时间手动启动，具体取决于配置。“事件日志”将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。

有关详细信息，请参阅服务器帮助文档。

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。可以使用 Concept 和 ProWORX 等设备编程应用程序生成变量导入文件。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。

另请参阅：从自定义应用程序导入

有关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”。

从自定义应用程序导入

可以使用以下 CSV 文件格式导入自定义标记：

[记录类型]；[变量名称]；[数据类型]；[地址]；[设置值]；[注释]，其中：

- **“记录类型”：**Concept 软件中使用的一种标志，也是另一种导入标记的方式。可以为 N 或 E：两个标志的处理方式相同。
- **“变量名称”：**服务器中静态标记的名称。其长度不能超过 256 个字符。
- **“数据类型”：**标记的数据类型。支持的数据类型如下：
  - BOOL
  - DINT
  - INT
  - 实型 (32 位浮点数)
  - 无符号双整型
  - 无符号整型
  - WORD
  - 字节
  - 时间 (视为双字型)
  - STRING
- **“地址”：**标记的 Modbus 地址。其长度不能超过 16 个字符。
- **“设置值”：**已被忽略，应将其留空。
- **“备注”：**服务器中标记的说明。其长度不能超过 255 个字符。

示例

- N;Amps;WORD;40001;;Current in
- N;Volts;WORD;40003;;Volts in
- N;Temperature;REAL;40068;;Tank temp

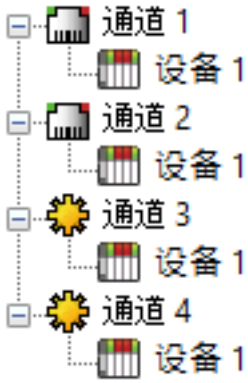
优化通信

Modbus TCP/IP Ethernet 驱动程序 旨在提供最佳性能，使得其对系统的整体性能影响最小。即使驱动程序速度很快，也可以利用一系列指南来控制和优化应用程序，并获得最佳性能。

服务器将诸如 Modbus 以太网等通信协议称为通道。应用程序中定义的每个通道都表示服务器中一个单独的执行路径。一旦定义了通道，则必须在该通道下定义一系列设备。每一个此类设备都代表一个可从中收集数据的单一 Modbus 控制器。虽然这种定义应用程序的方法提供了高水平的性能，但它不能充分利用驱动程序或网络。下面显示了使用单个通道配置时应用程序所呈现效果的示例。

每个均在单一 Modbus Ethernet 通道下定义。在此配置中，驱动程序必须尽快从一个设备移动到下一个设备，以有效速率收集信息。随着更多设备的添加或从单个设备请求的信息的增加，整体更新速率会受到不利影响。

如果 Modbus TCP/IP Ethernet 驱动程序 只能定义一个通道，则上述示例将是唯一可用的选项；但是，驱动程序最多可以定义 1024 个通道。使用多个通道，可通过同时向网络发出多个请求来分发数据集合工作载荷。下面显示了使用多个通道来提高性能时相同应用程序所呈现效果的示例。



每个设备均可在其自身的通道下定义。在此配置中，单个执行路径专用于从每个设备收集数据。如果应用程序的设备数小于等于 1024，则可用此方法对其进行精确优化。

即使应用程序设备数较大，也可改善性能。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。尽管在全部通道上分散设备载荷会使服务器再次从一个设备移动到另一个设备，但是，它可以用极少的设备在单通道上进行处理。

块大小

块大小会影响 Modbus TCP/IP Ethernet 驱动程序 的性能。块大小参数适用于在设备属性的"块大小"设置下定义下的每个设备。块大小是指一次可能需要从设备请求的寄存器数或位数。要优化此驱动程序的性能，请将块大小配置为 1 至 120 个寄存器 (8 至 2000 位)。

提示：

- 若要实现其他性能增益，可启用**"超时后关闭套接字"**属性以及调整超时和定时属性。
- 性能还受到套接字使用情况的影响，下文将对此进行讨论。

有关详细信息，请参阅以太网属性、通信超时和计时。

套接字使用

用户可指定驱动程序是应针对此通道上的所有设备使用单一套接字还是使用多个套接字与单一通道上的设备进行通信。如果驱动程序使用端口，则其他系统不得访问目标设备，因此在某些环境中可能需要考虑平衡对设备的访问。如果不希望驱动程序保持连接，或者设备的可用连接数有限，用户可以限制驱动程序可使用的套接字数量。

使用驱动程序与 Modbus-以太网到 Modbus-RTU 桥产品进行通信时，将驱动程序置于单套接字模式的功能非常重要。大部分产品允许将多个 RS-485 串行设备连接到一个 Modbus 以太网到 Modbus RTU 桥。为此，

可设置“每个通道一个套接字”，或设置“每台设备一个或多个套接字”，并将“每个设备的最大套接字数”设为 1 (*参阅通道以太网设置*)。

若配置了多个套接字，驱动程序会在目标设备所用的所有可用套接字中传播数据以读取目标设备的操作或向其写入操作。驱动程序**仅**使用处理数据所需的套接字数，最多不超过指定的最大数量。

如果通道下设备对象的目标 IP 地址和/或端口不同，则会关闭套接字，只在需要时才重新打开。如果以太网转串行网关设备要处理多个串行设备，建议使用此方法。网关 (和设备) 通常会限制并发连接数，以防止通信冲突。请勿超过这些限制。如果超过这些限制，则驱动程序会发布连接失败的消息。

驱动程序可能会同时接收到设备响应消息。设备的响应采用通道级的单线程按序处理；然而，这种通道级的数据处理速度非常快 (毫秒)，因此，若配置为使用多个套接字，可显著提升性能。

提示：性能还受块大小的影响，如上文所讨论，块大小会影响套接字间的数据传输。

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	两字节压缩 BCD 值的范围是 0 至 9999。对于超出此范围的值，未定义行为。
LBCD	四字节压缩 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；即，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。从以下列表中选择一个链接，以获取相关型号的具体地址信息。

Applicom 寻址
CEG 寻址
Fluenta 寻址
Instromet 寻址
邮箱寻址
Modbus 寻址
Roxar 寻址

驱动程序系统标记寻址

内部标记

标记	说明	数据类型	访问
端口	端口系统标记允许客户端应用程序读取和写入端口号设置。写入此标记会导致驱动程序与设备的连接断开，并尝试重新连接到指定的端口。	字、短整型、双字型、长整型	读/写

注意：

- 驱动程序不会将设备端口设置用于服务器通信。
- 对于此驱动程序，术语“服务器”和“非主动请求”可交替使用。
- 对此标记进行更改会修改项目，这将引起服务器在关闭时提示保存项目。

系统标记

标记	说明	数据类型	访问
_CEGExtension	此标记仅用于 CEG 型号设备。它允许从客户端应用程序中更改**“CEG 扩展”**设备属性。	布尔型	读/写
_InputCoilBlockSize	此标记允许从客户端应用程序中更改“输入线圈”块大小属性。	双字型	读/写
_OutputCoilBlockSize	此标记允许从客户端应用程序中更改“输出线圈”块大小属性。	双字型	读/写
_InternalRegisterBlockSize	此标记允许从客户端应用程序中更改“内部寄存器”块大小属性。	双字型	读/写
_HoldingRegisterBlockSize	此标记允许从客户端应用程序中更改“保存寄存器”块大小属性。	双字型	读/写

注意：对这些标记进行更改会修改项目，这将引起服务器在关闭时提示保存项目。

另请参阅：以太网

功能代码说明

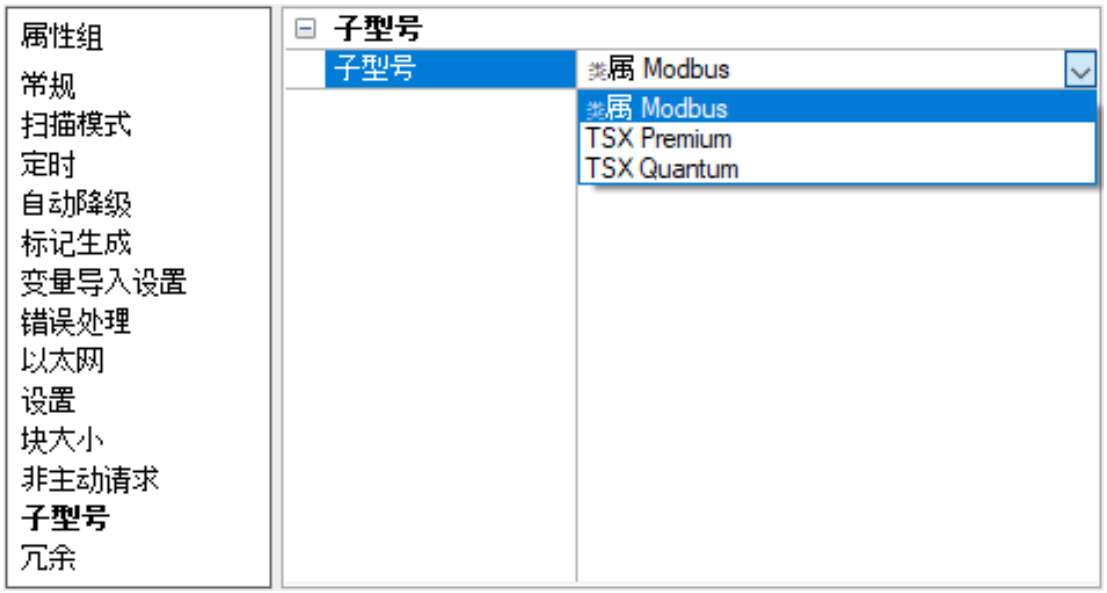
Modbus 和 Applicom 设备型号 支持下表中 显示的功能代码。

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
22	0x16	已屏蔽的写入寄存器

Applicom 子模型和寻址

Applicom 设备支持三个 Applicom 子模型。为要连接的设备选择适当的子模型。单击下方的子模型链接以获得地址信息。

类属 Modbus
TSX Premium
TSX Quantum



类属 Modbus 寻址

所有功能代码均以十进制显示。有关详细信息，请参阅功能代码说明。

输出线圈

地址	范围	数据类型	访问	功能代码
Bxxxxx	0-65535	布尔型	读/写	01、05、15

数组支持

输出线圈地址支持数组。用于声明数组的语法如下：
Bxxxxx_列数，假定行数为 1。
Bxxxxx_行数_列数。

基本地址 + (行数*列数) 不能超过 65535。正在请求的线圈总数不能超过为此设备指定的输出线圈块大小。

输入线圈

地址	范围	数据类型	访问	功能代码
BIxxxxx	0-65535	布尔型	只读	02

数组支持
输入线圈地址支持数组。用于声明数组的语法如下：
BIxxxxx_列数，假定行数为 1。
BIxxxxx_行数_列数。

基本地址 + (行数*列数) 不能超过 65535。正在请求的线圈总数不能超过为设备指定的输入线圈块大小。

内部寄存器

默认数据类型以**粗体**显示。

除了布尔型和字符串之外，所有数据类型的内部寄存器位置都支持数组。

注意：对于服务器设备，只读位置为读位置/写位置。

地址	范围	数据类型	访问	功能代码
WIxxxxx	0-65535 0-65534 0-65532	**字**、短整型、BCD 浮点型、双字型、长整型、LBCD 双精度	只读	04
WIxxxxx.bb	xxxxx=0-65535 bb=0/1-15/16*	**布尔型**	只读	04
WIxxxxx:Xbb	xxxxx=0-65535 bb=0/1-15/16*	**布尔型**	只读	04
DIxxxxx	0-65534	**双字型**	只读	04
FIxxxxx	0-65534	**浮点型**	只读	04
WIxxxxx_S	0-65535	**短整型**	只读	04
WIxxxxx_B	0-65535	**BCD**	只读	04
WIxxxxx_A**	0-65535	**字符串**	只读	04
WIxxxxx_X<1, 2, 3>***	0-65535 0-65534	**字**、短整型、BCD 浮点型、双字、长整型、LBCD	只读	04
DIxxxxx_S	0-65534	**长整型**	只读	04
DIxxxxx_B	0-65534	**LBCD**	只读	04
DIxxxxx_X<1, 2, 3>***	0-65534	**双字型**	只读	04
FIxxxxx_X<1, 2,	0-65534	**浮点型**	只读	04

地址	范围	数据类型	访问	功能代码
3>***				
M_WIxxxxx_n(H) 采用由高到低字节排序的字符串 (H 可选)	xxxxx=0-65535 n 是字符串长度，范围为 1 至 120 个字	**字符串**	只读	04
M_WIxxxxx_nL 采用由低到高字节排序的字符串	xxxxx=0-65535 n 是字符串长度，范围为 1 至 120 个字	**字符串**	只读	04

*有关详细信息，请参阅设置下的“从零开始的位寻址”。
**字符串的长度为 2 个字节。
有关详细信息，请参阅字节切换后缀。

数组支持
内部寄存器地址支持数组。用于声明数组的语法如下：
WIxxxxx_列数，假定行数为 1。
WIxxxxx_行数_列数。

对于字、短整型和 BCD 数组，基本地址 + (行数*列数) 不能超过 65535。
对于浮点型、双字型、长整型和长 BCD 数组，基本地址 + (行数*列数*2) 不能超过 65534。
对于所有数组，正在请求的寄存器总数不能超过为设备指定的内部寄存器块大小。

保持寄存器

默认数据类型以**粗体**显示。

除了布尔型和字符串之外，所有数据类型的保持寄存器位置都支持数组。

注意：对于服务器设备，只读位置为读位置/写位置。

地址	范围	数据类型	访问	功能代码
Wxxxxx	0-65535 0-65534 0-65532	**字**、短整型、BCD 浮点型、双字型、长整型、LBCD 双精度	读/写	03、06、16
Wxxxxx.bb	xxxxx=0-65535 bb=0/1-15/16*	**布尔型**	读/写	03、06、16、22
Wxxxxx:Xbb	xxxxx=0-65535 bb=0/1-15/16*	**布尔型**	读/写	03、06、16、22
Dxxxxx	0-65534	**双字型**	读/写	03、06、16
Fxxxxx	0-65534	**浮点型**	读/写	03、06、16
Wxxxxx_S	0-65535	**短整型**	读/写	03、06、16
Wxxxxx_B	0-65535	**BCD**	读/写	03、06、16
Wxxxxx_A**	0-65535	**字符串**	只读	03、16
Wxxxxx_X<1, 2, 3>***	0-65535 0-65534	**字**、短整型、BCD 浮点型、双字型、长整型、LBCD	读/写	03、06、16

地址	范围	数据类型	访问	功能代码
Dxxxxx_S	0-65534	长整型	读/写	03、06、16
Dxxxxx_B	0-65534	**LBCD**	读/写	03、06、16
Dxxxxx_X<1, 2, 3>***	0-65534	双字型	读/写	03、06、16
Fxxxxx_X<1, 2, 3>***	0-65534	浮点型	读/写	03、06、16
M_Wxxxxx_n(H) 采用由高到低字节排序的字符串 (H 可选)	xxxxx=0-65535 n 是字符串长度，范围为 1 至 120 个字	字符串	读/写	03、16
M_Wxxxxx_nL 采用由低到高字节排序的字符串	xxxxx=0-65535 n 是字符串长度，范围为 1 至 120 个字	字符串	读/写	03、16

*有关详细信息，请参阅设置下的“从零开始的位寻址”。
**字符串的长度为 2 个字节。
有关详细信息，请参阅字节切换后缀。

数组支持
保持寄存器地址支持数组。使用十进制寻址声明数组的语法如下：
Wxxxxx_列数，假定行数为 1。
Wxxxxx_行数_列数。

对于字、短整型和 BCD 数组，基本地址 + (行数*列数) 不能超过 65535。
对于浮点型、双字型、长整型和长 BCD 数组，基本地址 + (行数*列数*2) 不能超过 65534。
对于所有数组，正在请求的寄存器总数不能超过为设备指定的保持寄存器块大小。

字符串支持
Applicom 型号支持以 ASCII 字符串的形式读取和写入保持寄存器内存。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。字符串的长度可以为 1 到 120 个字。有关对字符串标记执行块读取的详细信息，请参阅块大小。

注意：字符串长度可能受设备允许的写入请求的最大大小限制。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。要解决这一问题，请将字符串缩短为受支持的长度。

字节切换后缀

这些后缀用于切换合成 16 位字、32 位双字型或 32 位浮点型数据的字节。应用“Modbus 字节顺序”和“首字排序为低”的设备级设置后，将进行字节切换。有关详细信息，请参阅设置。

字节切换后缀只能用于内部寄存器和保持寄存器。有关取决于项后缀和数据类型的各种切换类型的信息，请参阅下表。

后缀	**16 位数据类型 (字、短整型、BCD)**	**32 位数据类型 (双字型、长整型、LBCD、浮点型)**
_X1	O1 O2 -> O2 O1 (字节切换)	O1 O2 O3 O4 -> O4 O3 O2 O1 (字节切换)
_X2	O1 O2 -> O2 O1 (字节切换)	O1 O2 O3 O4 -> O3 O4 O1 O2 (字节切换)

后缀	**16 位数据类型 (字、短整型、BCD)**	**32 位数据类型 (双字型、长整型、LBCD、浮点型)**
_X3	O1 O2 -> O2 O1 (字节切换)	O1 O2 O3 O4 - > O2 O1 O4 O3 (字中的字节切换)

TSX Quantum

所有功能代码均以十进制显示。*有关详细信息，请参阅功能代码说明。*

输出线圈

地址	范围	数据类型	访问	功能代码
0xxxxx	1-65536	布尔型	读/写	01、05、15

数组支持
输出线圈地址支持数组。用于声明数组的语法如下：
0xxxxx_列数，假定行数为 1。
0Xxxxx_行数_列数。

基本地址 + (行数*列数) 不能超过 65536。正在请求的线圈总数不能超过为设备指定的输出线圈块大小。

输入线圈

地址	范围	数据类型	访问	功能代码
1xxxxx	1-65536	布尔型	只读	02

数组支持
输入线圈地址支持数组。用于声明数组的语法如下：
1xxxxx_列数，假设行数为 1。
1xxxxx_行数_列数。

基本地址 + (行数*列数) 不能超过 65536。正在请求的线圈总数不能超过为设备指定的输入线圈块大小。

内部寄存器

默认数据类型以**粗体**显示。

除了布尔型和字符串之外，所有数据类型的内部寄存器位置都支持数组。

注意：对于服务器设备，只读位置为读位置/写位置。

地址	范围	数据类型	访问	功能代码
3xxxxx	1-65536 1-65535 1-65533	**字**、短整型、BCD 浮点型、双字、长整型、LBCD 双精度	只读	04
3xxxxx.bb	xxxxx=1-65536 bb=0/1-15/16*	**布尔型**	只读	04
3xxxxx:Xbb	xxxxx=0-65535 bb=0/1-15/16*	**布尔型**	只读	04

地址	范围	数据类型	访问	功能代码
D3xxxxx	1-65535	**双字型**	只读	04
F3xxxxx	1-65535	**浮点型**	只读	04
3xxxxx_S	1-65536	**短整型**	只读	04
3xxxxx_B	1-65536	**BCD**	只读	04
3xxxxx_A**	1-65536	**字符串**	只读	04
3xxxxx_X<1, 2, 3>***	1-65536 1-65535	**字**、短整型、BCD 浮点型、双字、长整型、LBCD	只读	04
D3xxxxx_S	1-65535	**长整型**	只读	04
D3xxxxx_B	1-65535	**LBCD**	只读	04
D3xxxxx_X<1, 2, 3>***	1-65535	**双字型**	只读	04
F3xxxxx_X<1, 2, 3>***	1-65535	**浮点型**	只读	04
M_3xxxxx_n(H) 采用由高到低字节排序的字符串 (H 可选)	xxxxx=1-65536 n 是字符串长度，范围为 1 至 120 个字	**字符串**	只读	04
M_3xxxxx_nL 采用由低到高字节排序的字符串	xxxxx=1-65536 n 是字符串长度，范围为 1 至 120 个字	**字符串**	只读	04

*有关详细信息，请参阅设置下的“从零开始的位寻址”。
**字符串的长度为 2 个字节。
有关详细信息，请参阅字节切换后缀。

数组支持
内部寄存器地址支持数组。用于声明数组的语法如下：
3xxxxx_列数，假设行数为 1。
3Xxxxx_行数_列数。。

对于字、短整型和 BCD 数组，基本地址 + (行数*列数) 不能超过 65536。
对于浮点型、双字型、长整型和长 BCD 数组，基本地址 + (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设备指定的内部寄存器块大小。

保持寄存器

默认数据类型以**粗体**显示。

除了布尔型和字符串之外，所有数据类型的保持寄存器位置都支持数组。

注意：对于服务器设备，只读位置为读位置/写位置。

地址	范围	数据类型	访问	功能代码
4xxxxx	1-65536 1-65535 1-65533	**字**、短整型、BCD 浮点型、双字型、长整型、LBCD	读/写	03、06、16

地址	范围	数据类型	访问	功能代码
		双精度		
4xxxxx.bb	xxxxx=1-65536 bb=0/1-15/16*	**布尔型**	读/写	03、06、16、22
4xxxxx:Xbb	xxxxx=0-65535 bb=0/1-15/16*	**布尔型**	读/写	03、06、16、22
D4xxxxx	1-65535	**双字型**	读/写	03、06、16
F4xxxxx	1-65535	**浮点型**	读/写	03、06、16
4xxxxx_S	1-65536	**短整型**	读/写	03、06、16
4xxxxx_B	1-65536	**BCD**	读/写	03、06、16
4xxxxx_A**	1-65536	**字符串**	只读	03、16
4xxxxx_X<1, 2, 3>***	1-65536 1-65535	**字**、短整型、BCD 浮点型、双字、长整型、LBCD	读/写	03、06、16
D4xxxxx_S	1-65535	**长整型**	读/写	03、06、16
D4xxxxx_B	1-65535	**LBCD**	读/写	03、06、16
D4xxxxx_X<1, 2, 3>***	1-65535	**双字型**	读/写	03、06、16
F4xxxxx_X<1, 2, 3>***	1-65535	**浮点型**	读/写	03、06、16
M_4xxxxx_n(H) 采用由高到低字节排序的字符串 (H 可选)	xxxxx=1-65536 n 是字符串长度，范围为 1 至 120 个字	**字符串**	读/写	03、16
M_4xxxxx_nL 采用由低到高字节排序的字符串	xxxxx=1-65536 n 是字符串长度，范围为 1 至 120 个字	**字符串**	读/写	03、16

*有关详细信息，请参阅设置下的“从零开始的位寻址”。
**字符串的长度为 2 个字节。
有关详细信息，请参阅字节切换后缀。

数组支持
保持寄存器地址支持数组。使用十进制寻址声明数组的语法如下：

4xxxxx_列数，假设行数为 1。
4xxxxx_行数_列数。

对于字、短整型和 BCD 数组，基本地址 + (行数*列数) 不能超过 65536。
对于浮点型、双字型、长整型和长 BCD 数组，基本地址 + (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设备指定的保持寄存器块大小。

字符串支持
Applicom 型号支持以 ASCII 字符串的形式读取和写入保持寄存器内存。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。字符串的长度可以为 1 到 120 个字。
有关对字符串标记执行块读取的信息，请参阅块大小。

注意：字符串长度可能受设备允许的写入请求的最大大小限制。如果在服务器事件窗口中接收到错误消息“无法在写入设备 <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。要解决这一问题，请将字符串缩短为受支持的长度。

字节切换后缀

这些后缀用于切换合成 16 位字、32 位双字型或 32 位浮点型数据的字节。应用“Modbus 字节顺序”和“首字排序为低”的设备级设置后，将进行字节切换。有关详细信息，请参阅设置。

字节切换后缀只能用于内部寄存器和保持寄存器。有关取决于项后缀和数据类型的各种切换类型的信息，请参阅下表。

后缀	16 位数据类型 (字、短整型、BCD)	32 位数据类型 (双字型、长整型、LBCD、浮点型)
_X1	O1 O2 -> O2 O1 (字节切换)	O1 O2 O3 O4 -> O4 O3 O2 O1 (字节切换)
_X2	O1 O2 -> O2 O1 (字节切换)	O1 O2 O3 O4 -> O3 O4 O1 O2 (字节切换)
_X3	O1 O2 -> O2 O1 (字节切换)	O1 O2 O3 O4 - > O2 O1 O4 O3 (字中的字节切换)

TSX Premium

所有功能代码均以十进制显示。有关详细信息，请参阅功能代码说明。

输出线圈

地址	范围	数据类型	访问	功能代码
%MXxxxxx	0-65535	布尔型	读/写	01、05、15
%Mxxxxx	0-65535	布尔型	读/写	01、05、15

数组支持

输出线圈地址支持数组。用于声明数组的语法如下：

%MXxxxxx_列数，假定行数为 1。

%MXxxxxx_行数_列数。

基本地址 + (行数*列数) 不能超过 65535。正在请求的线圈总数不能超过为设备指定的输出线圈块大小。

保持寄存器

默认数据类型以**粗体**显示。

除了布尔型和字符串之外，所有数据类型的保持寄存器位置都支持数组。

注意：对于服务器设备，只读位置为读位置/写位置。

地址	范围	数据类型	访问	功能代码
%MWxxxxx	0-65535 0-65534 0-65532	**字**、短整型、BCD 浮点型、双字、长整型、LBCD 双精度	读/写	03、06、16
%MWxxxxx.bb	xxxxx=0-65535 bb=0/1-15/16*	**布尔型**	读/写	03、06、16、22

地址	范围	数据类型	访问	功能代码
%MWxxxxx:Xbb	xxxxx=0-65535 bb=0/1-15/16*	布尔型	读/写	03、06、16、22
%DWxxxxx 或 %MDxxxxx	0-65534	双字型	读/写	03、06、16
%FWxxxxx 或 %MFxxxxx	0-65534	浮点型	读/写	03、06、16
%MWxxxxx_S	0-65535	短整型	读/写	03、06、16
%MWxxxxx_B	0-65535	**BCD**	读/写	03、06、16
%MWxxxxx_A**	0-65535	字符串	只读	03、16
%MWxxxxx_X<1, 2, 3>***	0-65535 0-65534	字、短整型、BCD 浮点型、双字、长整型、LBCD	读/写	03、06、16
%DWxxxxx_S	0-65534	长整型	读/写	03、06、16
%DWxxxxx_B	0-65534	**LBCD**	读/写	03、06、16
%DWxxxxx_X<1, 2, 3>*** 或 %MDxxxxx_X<1, 2, 3>***	0-65534	双字型	读/写	03、06、16
%FWxxxxx_X<1, 2, 3>*** 或 %MFxxxxx_X<1, 2, 3>***	0-65534	浮点型	读/写	03、06、16
M_%MWxxxxx_n(H) 采用由高到低字节排序 的字符串 (H 可选)	xxxxx=0-65535 n 是字符串长度，范围为 1 至 120 个字	字符串	读/写	03、16
M_%MWxxxxx_nL 采用由低到高字节排序的字符串	xxxxx=0-65535 n 是字符串长度，范围为 1 至 120 个字	字符串	读/写	03、16

*有关详细信息，请参阅设置下的“从零开始的位寻址”。
**字符串的长度为 2 个字节。
有关详细信息，请参阅字节切换后缀。

数组支持
保持寄存器地址支持数组。使用十进制寻址声明数组的语法如下：
%MWxxxxx_列数，假定行数为 1。
%MWxxxxx_行数_列数。

对于字、短整型和 BCD 数组，基本地址 + (行数*列数) 不能超过 65535。
对于浮点型、双字型、长整型和长 BCD 数组，基本地址 + (行数*列数*2) 不能超过 65534。
对于所有数组，正在请求的寄存器总数不能超过为设备指定的保持寄存器块大小。

字符串支持

Applicom 型号支持以 ASCII 字符串的形式读取和写入保持寄存器内存。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。字符串的长度可以为 1 到 120 个字。有关对字符串标记执行块读取的详细信息，请参阅块大小。

注意：字符串长度可能受设备允许的写入请求的最大大小限制。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。要解决这一问题，请将字符串缩短为受支持的长度。

字节切换后缀

这些后缀用于切换合成 16 位字、32 位双字型或 32 位浮点型数据的字节。应用“Modbus 字节顺序”和“首字排序为低”的设备级设置后，将进行字节切换。有关详细信息，请参阅设置。

字节切换后缀只能用于内部寄存器和保持寄存器。有关取决于项后缀和数据类型的各种切换类型的信息，请参阅下表。

后缀	16 位数据类型 (字、短整型、BCD)	32 位数据类型 (双字型、长整型、LBCD、浮点型)
_X1	O1 O2 -> O2 O1 (字节切换)	O1 O2 O3 O4 -> O4 O3 O2 O1 (字节切换)
_X2	O1 O2 -> O2 O1 (字节切换)	O1 O2 O3 O4 -> O3 O4 O1 O2 (字节切换)
_X3	O1 O2 -> O2 O1 (字节切换)	O1 O2 O3 O4 - > O2 O1 O4 O3 (字中的字节切换)

CEG 寻址

CEG 设备型号的寻址与 Modbus 设备型号的寻址相同。

有关详细信息，请参阅 ***Modbus 寻址***。

Fluenta 寻址

默认数据类型以**粗体**显示。

地址	范围	数据类型	访问
系统	400000-409999	**浮点型**、双精度	读/写
输出	410000-410999 420000-420999 430000-430999	**浮点型**、双精度	只读
用户	411000-411999 421000-421999 431000-431999	**浮点型**、双精度	读/写
服务	412000-412999 422000-422999 432000-432999	**浮点型**、双精度	读/写
累积	413000-413999 423000-423999 433000-433999	**浮点型**、双精度	只读

Instromet 寻址

默认数据类型以**粗体**显示。

地址	范围	数据类型	访问
短整数	400000-400199	**字**、短整型	只读
长整型	400200-400399	**双字**、长整型	只读
浮点数	400400-400599	**浮点型**	只读

邮箱寻址

默认数据类型以**粗体**显示。

十进制寻址

地址	范围	数据类型	访问
4xxxxx	1-65536	字、短整型、BCD	读/写
4xxxxx.bb	xxxxx=1-65536 bb=0-15	**布尔型**	读/写
4xxxxx	1-65535	浮点型、双字型、长整型、LBCD	读/写

十六进制寻址

地址	范围	数据类型	访问
H4yyyyy	1-10000	字、短整型、BCD	读/写
H4yyyyy.c	yyyyy = 1-10000 c=0-F	**布尔型**	读/写
H4yyyy	1-FFFF	浮点型、双字型、长整型、LBCD	读/写

请知悉：Modbus 邮箱不支持功能代码 22 (0x16)。仅支持 0x10 (保持寄存器写入多次) 和 0x6 (保持寄存器写入单次)。要写入单个位，可以关闭设置选项卡下的设备属性中的**"保持寄存器位写入"**。这将强制其使用"读取/修改/写入"序列，而不是直接写入该位。只有客户端 Modbus 设备 (不是邮箱) 必须更改其设置才能解决此问题。

数组

保存寄存器地址也支持数组。声明数组 (使用十进制寻址) 的语法如下：

4xxxx[列数]，假定行数为 1。
4xxxx[行数][列数]。

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。

Modbus 寻址

对于此驱动程序，术语"服务器"和"非主动请求"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。数据项的最大值是一个两字节的无符号整型值 (65,535)。驱动程序内部需要六位数字来表示整个地址表格和项。请务必注意，许多 Modbus 设备可能不支持数据项的整个范围。为避免在输入此类设备的地址时出现混乱，此驱动程序将根据地址字段中输入的内容来"填充"地址 (添加数字)。如果主表格类型后跟最多 4 位数字 (示例：4 x、4xx、4xxx 或 4xxxx)，则地址仍可保留或使用额外的零填充到五 (5) 位数字。如果主表格类型后跟五 (5) 位数字 (示例：4xxxxx)，则该地址不变。在内部，输入为 41、401、4001、40001 或 400001 的地址是指定主表格类型 4 和数据项 1 的地址的所有对等表示。

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

功能代码均以十进制显示。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 300001-365535 300001-365533 xxxxx=1-65536 bb=0/1-15/16** 300001.2H-365536.240H*** 300001.2L-365536.240L***	**字**、短整型、BCD 浮点型、双字、长整型、LBCD 双精度 **布尔型** 字符串 字符串	只读 只读 只读 只读 只读 只读	04 04 04 04 04 04
保持寄存器	400001-465536 400001-465535 400001-465533 xxxxx=1-65536 bb=0/1-15/16* 400001.2H-465536.240H*** 400001.2L-465536.240L***	**字**、短整型、BCD 浮点型、双字、长整型、LBCD 双精度 **布尔型** **字符串** **字符串**	读/写 读/写 读/写 读/写 读/写 读/写	03、06、16 03、06、16 03、06、16 03、06、16、22 03、16 03、16

*对于服务器设备，只读位置为读位置/写位置。
***有关详细信息，请参阅"设置"中的"从零开始的寻址"。*
***.Bit 是字符串长度，范围介于 2 到 240 个字符之间。

以十六进制格式表示的 Modbus 寻址

地址类型	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 yyyyy = 1-10000 cc=0/1-F/10 H300001.2H-H3FFFF.240H** H300001.2L-H3FFFF.240L**	**字**、短整型、BCD 浮点型、双字、长整型、LBCD 双精度 **布尔型** 字符串 字符串	只读 只读 只读 只读 只读 只读
保持寄存器	H400001-H410000 H400001-H40FFFF H400001-H40FFFD yyyyy = 1-10000 cc=0/1-F/10 H400001.2H-H4FFFF.240H H400001.2L-H4FFFF.240L	**字**、短整型、BCD 浮点型、双字、长整型、LBCD 双精度 **布尔型** **字符串** **字符串**	读/写 读/写 读/写 读/写 读/写 读/写

*对于服务器设备，只读位置为读位置/写位置。
**.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。此功能只有在轮询模式下才适用于输入线圈和输出线圈。此功能不适用于为访问非主动请求的内存映射而配置的设备或位于邮箱模式下的设备。十进制语法为 0*xxxxx#nn*，其中：
*对于服务器设备，只读位置为读位置/写位置。
**.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。此功能只有在轮询模式下才适用于输入线圈和输出线圈。此功能不适用于为访问非主动请求的内存映射而配置的设备或位于邮箱模式下的设备。十进制语法为 0*xxxxx#nn*，其中：

- *xxxxx* 是第一个线圈的地址 (其范围为 000001-065521)。
- *nn* 是封装进模拟值的线圈数 (范围为 01-16)。

十六进制语法为 H0*yyyyy#nn*，其中：

- *xxxxx* 是第一个线圈的地址 (其范围为 H000001-H000FFF1)。
- *nn* 是封装进模拟值的线圈数 (范围为 01-16)。

注意：

1. 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。在十进制寻址中，输出线圈支持功能代码 01 和 15，而输入线圈支持功能代码 02。

2. 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/写地址均可通过将 "W" 添加为地址前缀 (如 "W40001") 设置为“只写”，这会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所有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。

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。

有关对 Modbus 型号的字符串标记执行块读取的详细信息，请参阅块大小。

示例

1. 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"。
2. 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：字符串长度可能受设备允许的写入请求的最大大小限制。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。如果可能，请尝试缩短字符串。

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：

4*xxxx* [行数] [列数]

4*xxxx* [列数]，假定列数为 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。

Roxar 寻址

默认数据类型以**粗体**显示。

地址	范围	数据类型	访问
短整数	403000-403999	**字**、短整型	读/写
浮点数	407000-407999	**浮点型**	读/写
浮点数	409000-409999	**浮点型**	只读

统计信息项

统计信息项使用通过其他诊断信息收集默认情况下不收集的数据。要使用统计信息项，必须启用“通信诊断”。要启用“通信诊断”，在项目视图中右键单击通道，然后单击**“属性”| “启用诊断”**。或者，双击通道，然后选择**“启用诊断”**。

通道级统计信息项

通道级统计信息项的语法为 *<通道>._Statistics*。

注意：通道级别的统计信息是设备级别中相同项的汇总。

项	数据类型	访问	说明
_CommFailures	双字型	读/写	通信失败 (或耗用的重试) 的总次数。
_ErrorResponses	双字型	读/写	收到有效错误响应的总数。
_ExpectedResponses	双字型	读/写	收到预期响应的总数。
_LastResponseTime	字符串	只读	收到最后一次有效响应的时间。
_LateData	双字型	读/写	读取标记的时间晚于预期的总次数 (基于指定的扫描速率)。由于 DNR 错误状态，此值不会增加。如果首次读取发生在通信中断之后，则标记不会被计入延迟次数 (即使事实如此)。这是设计如此的。
_MsgResent	双字型	读/写	作为重试发送的消息总数。
_MsgSent	双字型	读/写	初始发送的消息总数。
_MsgTotal	双字型	只读	发送的消息总数 (_MsgSent + _MsgResent 二者之和)。
_PercentReturn	浮点型	只读	预期响应 (“已接收”) 与初始发送 (“已发送”) 的比例，以百分比形式表示。
_PercentValid	浮点型	只读	收到的有效响应总数 (_TotalResponses) 与发送的总请求数 (_MsgTotal) 的比例，以百分比形式表示。
_Reset	布尔型	读/写	重置所有诊断计数器。写入“_Reset 标记”会导致此级别的所有诊断计数器重置。
_RespBadChecksum*	双字型	读/写	校验和错误的响应总数。
_RespTimeouts	双字型	读/写	无法接收任何响应类型的消息总数。
_RespTruncated	双字型	读/写	仅接收部分响应的消息总数。
_TotalResponses	双字型	只读	收到有效响应的总数 (_ErrorResponses + _ExpectedResponses)。

* 未执行 _RespBadChecksum 统计；数据包校验和由 TCP 协议处理。

统计信息项不会在模拟模式下更新 *(请参阅设备常规属性)*。

设备级统计信息项

设备级统计信息项的语法为 *<通道>.<设备>._Statistics*。

项	数据类型	访问	说明
_CommFailures	双字型	读/写	通信失败 (或耗用的重试) 的总次数。
_ErrorResponses	双字型	读/写	收到有效错误响应的总数。
_ExpectedResponses	双字型	读/写	收到预期响应的总数。
_LastResponseTime	字符串	只读	收到最后一次有效响应的时间。
_LateData	双字型	读/写	读取标记的时间晚于预期的总次数 (基于指定的扫描速率)。由于 DNR 错误状态，此值不会增加。如果首次读取发生在通信中断之后，则标记不会被计入延迟次数 (即使事实如此)。这是设计如此的。
_MsgResent	双字型	读/写	作为重试发送的消息总数。
_MsgSent	双字型	读/写	初始发送的消息总数。
_MsgTotal	双字型	只读	发送的消息总数 (_MsgSent + _MsgResent 二者之和)。
_PercentReturn	浮点型	只读	预期响应 (“已接收”) 与初始发送 (“已发送”) 的比例，以百分比形式表示。
_PercentValid	浮点型	只读	收到的有效响应总数 (_TotalResponses) 与发送的总请求数 (_MsgTotal) 的比例，以百分比形式表示。
_Reset	布尔型	读/写	重置所有诊断计数器。写入“_Reset 标记”会导致此级别的所有诊断计数器重置。
_RespBadChecksum*	双字型	读/写	校验和错误的响应总数。
_RespTimeouts	双字型	读/写	无法接收任何响应类型的消息总数。
_RespTruncated	双字型	读/写	仅接收部分响应的消息总数。
_TotalResponses	双字型	只读	收到有效响应的总数 (_ErrorResponses + _ExpectedResponses)。

* 未执行 _RespBadChecksum 统计；数据包校验和由 TCP 协议处理。

注意：统计信息项在模拟模式下不更新*(请参阅设备常规属性)*。

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。

提示：来自数据源 (如第三方软件，包括数据库) 的消息通过事件日志显示。故障排除步骤应包括在网上和供应商文档中研究这些消息。

无法启动 winsock 通信。

错误类型：
错误

无法启动非主动请求通信。

错误类型：
错误

可能的原因：
驱动程序无法为非主动请求通信创建侦听套接字。

可能的解决方案：
验证在通道级别定义的端口是否正在由系统上的另一应用程序使用。

注意：
对于此驱动程序，术语“Modbus 服务器”和“非主动请求”可互换使用。

针对未定义设备的非主动请求邮箱访问。正在关闭套接字。| IP 地址 = '<地址>'。

错误类型：
错误

可能的原因：

1. 具有指定 IP 地址的设备尝试向服务器发送了邮箱消息。由于并未在“邮箱项目”中配置具有该 IP 的设备，因此该消息未通过验证。
2. 具有指定 IP 地址的设备尝试向服务器发送了邮箱消息。尽管已配置设备，但由于没有任何客户端从其中请求数据，因此该消息未通过验证。

可能的解决方案：
对于接受邮箱消息的服务器，必须在项中配置指定的设备 IP。而且必须由客户端从该设备请求至少一个数据项。

非主动请求邮箱接收到不支持的请求。| IP 地址 = '<地址>'。

错误类型：
错误

可能的原因：

从指定的设备 IP 接收到不支持的请求。请求格式无效，且不在 Modbus 规范中。

可能的解决方案：

验证配置为发送邮箱数据的设备是否发送了有效请求。

非主动请求邮箱内存分配错误。| IP 地址 = '<地址>'。

错误类型：

错误

可能的原因：

1. 具有指定 IP 地址的设备尝试向服务器发送了邮箱消息。由于并未在“邮箱项目”中配置具有该 IP 的设备，因此该消息未通过验证。
2. 具有指定 IP 地址的设备尝试向服务器发送了邮箱消息。尽管已配置设备，但由于没有任何客户端从其中请求数据，因此该消息未通过验证。

可能的解决方案：

对于接受邮箱消息的服务器，必须在项中配置指定的设备 IP。而且必须由客户端从该设备请求至少一个数据项。

无法创建套接字连接。

错误类型：

错误

可能的原因：

服务器无法建立与指定设备的 TCP/IP 套接字连接，但是会继续尝试连接。

可能的解决方案：

1. 验证设备是否在线。
2. 验证设备 IP 是否在绑定服务器的 IP 子网内。验证有效网关是否可用，以允许连接至另一网络。

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

不良数组。| 数组范围 = <开始> 至 <结束>。

错误类型：

错误

可能的原因：

定义了一个地址数组，该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。

块中的地址错误。| 块范围 = <地址> 至 <地址>。

错误类型：
错误

可能的原因：
驱动程序尝试读取了 PLC 中不存在 (可能是超出范围了) 的位置。例如，在仅具有地址为 40001 至 41400 的保持寄存器的 PLC 中，请求地址 41405 即会产生此错误。产生此错误后，驱动程序不会再次从 PLC 请求指定的数据块。从此同一块请求的任何其他地址都将被视为无效。

可能的解决方案：
将客户端应用程序更新到设备范围内的请求地址。

也可以看看：
错误处理

无法解析主机。| 主机名 = '<名称>'。

错误类型：
错误

可能的原因：
设备被配置为使用 DNS 主机名，而不是 IP 地址。服务器无法将主机名解析为 IP 地址。

可能的解决方案：
验证设备是否在线并已注册到域中。

指定的输出线圈块大小超过了最大块大小。| 指定的块大小 = <数字> (线圈)，最大块大小 = <数字> (线圈)。

错误类型：
错误

指定的输入线圈块大小超过了最大块大小。| 指定的块大小 = <数字> (线圈)，最大块大小 = <数字> (线圈)。

错误类型：
错误

指定的内部寄存器块大小超过了最大块大小。| 指定的块大小 = <数字> (寄存器)，最大块大小 = <数字> (寄存器)。

错误类型：
错误

指定的保持寄存器块大小超过了最大块大小。| 指定的块大小 = <数字> (寄存器)，最大块大小 = <数字> (寄存器)。

错误类型：

错误

块请求响应异常。| 块范围 = <地址> 至 <地址>，异常 = <代码>。

错误类型：
警告

可能的原因：
设备返回了异常代码。

可能的解决方案：
请参阅异常代码文档。

也可以看看：
Modbus 异常代码

块请求响应异常。| 块范围 = <地址> 到 <地址>，功能码 = <代码>，异常 = <代码>。

错误类型：
警告

可能的原因：
设备返回了异常代码。

可能的解决方案：
请参阅异常代码文档。

也可以看看：
Modbus 异常代码

接收到无效块长度。| 块范围 = <开始> 至 <结束>。

错误类型：
警告

可能的原因：
驱动程序尝试读取 PLC 中的内存块。PLC 响应错误，但未向驱动程序提供请求的数据块大小。

可能的解决方案：
确保 PLC 的内存范围存在。

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：
驱动程序无法分配处理变量导入文件所需的内存。

可能的解决方案：

关闭所有不必要的应用程序，然后重试。

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时出错。| 记录编号 = <数字>，字段 = <字段>。

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：
如有可能，请编辑变量导入文件以更改有问题的字段。

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：
指定记录中提供的标记说明过长。

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建有效的名称。为防止此错误发生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：
将在变量导入文件中指定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：
设备返回了异常代码。

可能的解决方案：
请参阅异常代码文档。

也可以看看：
Modbus 异常代码

以太网管理器已启动。

错误类型：
信息化

以太网管理器已停止。

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

客户端应用程序已通过系统 tag _CEGExtension 更改 CEG 扩展。| 扩展 = '<扩展>'。

错误类型：
信息化

可能的原因：
连接至服务器的客户端应用程序将指定设备上的 CEG 扩展更改为 0 (用于 Modbus) 或 1 (用于 CEG)。

可能的解决方案：
此设备属性仅适用于 CEG 型号设备。更改不会影响其他型号。要限制客户端应用程序更改此属性，请通过 OPC DA 设置禁用客户端写入系统级标记的功能。

正在启动非主动请求通信。| 协议 = '<名称>'，端口 = <数字>。

错误类型：
信息化

为 Modbus 服务器设备创建了内存。| Modbus 服务器设备 ID = <设备>。

错误类型：
信息化

所有通道都已订阅虚拟网络，或者所有设备均在侦听远程地址，正在停止非主动请求通信。

错误类型：
信息化

通道处于虚拟网络中，所有设备都会恢复到“每台设备使用一个套接字”。

错误类型：
信息化

与客户端相连时，无法将设备 ID 从 Modbus 客户端模式更改为服务器模式。

错误类型：
信息化

与客户端相连时，无法将设备 ID 从 Modbus 服务器模式更改为客户端模式。

错误类型：
信息化

当通道处于虚拟网络中时，不允许使用 Modbus 服务器模式。设备 ID 无法包含环回或本地 IP 地址。

错误类型：
信息化

当通道处于虚拟网络中时，不允许使用邮箱模型。

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“非主动请求”可交替使用。

Modbus Ethernet 通道属性

以下是所有 Modbus Ethernet 通道级属性的完整列表。


```
{ "common.ALLTYPES_NAME": "MyChannel", "common.ALLTYPES_DESCRIPTION": "",
"servermain.MULTIPLE_TYPES_DEVICE_DRIVER": "Modbus TCP/IP Ethernet",
"servermain.CHANNEL_DIAGNOSTICS_CAPTURE": false, "servermain.CHANNEL_UNIQUE_ID":
721923342, "servermain.CHANNEL_ETHERNET_COMMUNICATIONS_NETWORK_ADAPTER_STRING": "",
"servermain.CHANNEL_WRITE_OPTIMIZATIONS_METHOD": 2, "servermain.CHANNEL_WRITE_
OPTIMIZATIONS_DUTY_CYCLE": 10, "servermain.CHANNEL_NON_NORMALIZED_FLOATING_POINT_
HANDLING": 0, "servermain.CHANNEL_COMMUNICATIONS_SERIALIZATION_VIRTUAL_NETWORK": 0,
"servermain.CHANNEL_COMMUNICATIONS_SERIALIZATION_TRANSACTIONS_PER_CYCLE": 1,
"servermain.CHANNEL_COMMUNICATIONS_SERIALIZATION_NETWORK_MODE": 0, "modbus_
ethernet.CHANNEL_USE_ONE_OR_MORE_SOCKETS_PER_DEVICE": 1, "modbus_ethernet.CHANNEL_
MAXIMUM_SOCKETS_PER_DEVICE": 1 }
```


Modbus Ethernet 设备属性

以下是所有 Modbus Ethernet 设备级属性的完整列表。


```
{ "common.ALLTYPES_NAME": "MyDevice", "common.ALLTYPES_DESCRIPTION": "",
"servermain.MULTIPLE_TYPES_DEVICE_DRIVER": "Modbus TCP/IP Ethernet",
"servermain.DEVICE_MODEL": 0, "servermain.DEVICE_UNIQUE_ID": 70949968,
"servermain.DEVICE_CHANNEL_ASSIGNMENT": "MyChannel", "servermain.DEVICE_ID_FORMAT":
0, "servermain.DEVICE_ID_STRING": "<0.0.0.0>.0", "servermain.DEVICE_ID_HEXADECIMAL":
0, "servermain.DEVICE_ID_DECIMAL": 0, "servermain.DEVICE_ID_OCTAL": 0,
"servermain.DEVICE_DATA_COLLECTION": true, "servermain.DEVICE_SIMULATED": false,
"servermain.DEVICE_SCAN_MODE": 0, "servermain.DEVICE_SCAN_MODE_RATE_MS": 1000,
"servermain.DEVICE_SCAN_MODE_PROVIDE_INITIAL_UPDATES_FROM_CACHE": false,
"servermain.DEVICE_CONNECTION_TIMEOUT_SECONDS": 3, "servermain.DEVICE_REQUEST_
TIMEOUT_MILLISECONDS": 1000, "servermain.DEVICE_RETRY_ATTEMPTS": 3,
"servermain.DEVICE_INTER_REQUEST_DELAY_MILLISECONDS": 0, "servermain.DEVICE_AUTO_
DEMOTION_ENABLE_ON_COMMUNICATIONS_FAILURES": false, "servermain.DEVICE_AUTO_DEMOTION_
DEMOTE_AFTER_SUCCESSIVE_TIMEOUTS": 3, "servermain.DEVICE_AUTO_DEMOTION_PERIOD_MS":
10000, "servermain.DEVICE_AUTO_DEMOTION_DISCARD_WRITES": false, "servermain.DEVICE_
TAG_GENERATION_ON_STARTUP": 0, "servermain.DEVICE_TAG_GENERATION_DUPLICATE_HANDLING":
0, "servermain.DEVICE_TAG_GENERATION_GROUP": "", "servermain.DEVICE_TAG_GENERATION_
ALLOW_SUB_GROUPS": true, "modbus_ethernet.DEVICE_VARIABLE_IMPORT_FILE": "normal.txt",
"modbus_ethernet.DEVICE_VARIABLE_IMPORT_INCLUDE_DESCRIPTIONS": 1, "modbus_
ethernet.DEVICE_DEACTIVATE_TAGS_ON_ILLEGAL_ADDRESS": 1, "modbus_ethernet.DEVICE_SUB_
MODEL": 1, "modbus_ethernet.DEVICE_ETHERNET_PORT_NUMBER": 502, "modbus_
ethernet.DEVICE_ETHERNET_IP_PROTOCOL": 1, "modbus_ethernet.DEVICE_ETHERNET_CLOSE_TCP_
SOCKET_ON_TIMEOUT": true, "modbus_ethernet.DEVICE_ZERO_BASED_ADDRESSING": true,
"modbus_ethernet.DEVICE_ZERO_BASED_BIT_ADDRESSING": true, "modbus_ethernet.DEVICE_
HOLDING_REGISTER_BIT_MASK_WRITES": true, "modbus_ethernet.DEVICE_MODBUS_FUNCTION_06":
true, "modbus_ethernet.DEVICE_MODBUS_FUNCTION_05": true, "modbus_ethernet.DEVICE_
MODBUS_BYTE_ORDER": true, "modbus_ethernet.DEVICE_FIRST_WORD_LOW": true, "modbus_
ethernet.DEVICE_FIRST_DWORD_LOW": true, "modbus_ethernet.DEVICE_MODICON_BIT_ORDER":
false, "modbus_ethernet.DEVICE_TREAT_LONGS_AS_DOUBLE_PRECISION_UNSIGNED_DECIMAL":
false, "modbus_ethernet.DEVICE_OUTPUT_COILS": 32, "modbus_ethernet.DEVICE_INPUT_
COILS": 32, "modbus_ethernet.DEVICE_INTERNAL_REGISTERS": 32, "modbus_ethernet.DEVICE_
HOLDING_REGISTERS": 32, "modbus_ethernet.DEVICE_PERFORM_BLOCK_READ_ON_STRINGS": 0 }
```


注意：servermain.DEVICE_MODEL 参数默认为类属 Modbus 型号。如果不需要，请确保正确定义此参数。

Modbus Ethernet 标记属性

以下是所有 Modbus Ethernet 标记属性的完整列表。


```
{ "common.ALLTYPES_NAME": "MyTag", "common.ALLTYPES_DESCRIPTION": "",
"servermain.TAG_ADDRESS": "400001", "servermain.TAG_DATA_TYPE": 5, "servermain.TAG_
READ_WRITE_ACCESS": 1, "servermain.TAG_SCAN_RATE_MILLISECONDS": 100, "servermain.TAG_
AUTOGENERATED": false, "servermain.TAG_SCALING_TYPE": 0, "servermain.TAG_SCALING_RAW_
LOW": 0, "servermain.TAG_SCALING_RAW_HIGH": 1000, "servermain.TAG_SCALING_SCALED_
DATA_TYPE": 9, "servermain.TAG_SCALING_SCALED_LOW": 0, "servermain.TAG_SCALING_
SCALED_HIGH": 1000, "servermain.TAG_SCALING_CLAMP_LOW": false, "servermain.TAG_
SCALING_CLAMP_HIGH": false, "servermain.TAG_SCALING_NEGATE_VALUE": false,
"servermain.TAG_SCALING_UNITS": "" }
```

索引

“

“标识” 8

5

5 位寻址 47

6

6 位寻址 47

A

Applicom 30
Applicom 寻址 36

B

BCD 34
BOOL 31

C

CEG 30
CEG 扩展 24
CEG 寻址 45
CEGExtension 35
CSV 31

D

DINT 31

F

Fluenta 6
Fluenta FGM 30
Fluenta 寻址 45

H

HoldingRegisterBlockSize 35

I

ID 14
InputCoilBlockSize 35
Instromet 6, 30
Instromet 寻址 46
INT 31
InternalRegisterBlockSize 35
IP 协议 12, 22

L

LBCD 34

M

Modbus 非主动请求 6
Modbus 功能码 05 24
Modbus 功能码 06 23
Modbus 客户端 6
Modbus 客户端和服务器注意事项 20
Modbus 寻址 46
Modbus 异常代码 59
Modbus 邮箱 46
Modbus 字节顺序 24
Modicon 位顺序 24

O

OPC 质量不佳 20
OutputCoilBlockSize 35

R

REAL 31
Roxar 7
Roxar RFM 30
Roxar 寻址 49

S

STRING 31

T

TSX Premium 43
TSX Quantum 40

W

WORD 31

帮

帮助内容 5

包

包括说明 19

保

保持寄存器 26, 38, 41, 43, 47
保持寄存器位写入 23

备

备注 31

变

变量 31
变量导入设置 19

标

标记导入期间发生文件异常。 56
标记导入由于内存资源不足而失败。 55
标记计数 8
标记生成 17

不

不良数组。| 数组范围 = <开始> 至 <结束>。 53
不扫描，仅按需求轮询 15

布

布尔型 34

常

常规 14

超

超时后关闭套接字 22
超时前的尝试次数 16

创

创建 19

从

从零开始的位寻址 23
从零开始的寻址 23
从自定义应用程序导入 31

错

错误处理 21

当

当通道处于虚拟网络中时，不允许使用 Modbus 服务器模式。设备 ID 无法包含环回或本地 IP 地址。 58
当通道处于虚拟网络中时，不允许使用邮箱模型。 58

导

导入标记数据库时打开文件出错。| OS 错误 = '<错误>'。 53
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>'。 56

地

地址 31
地址说明 35

定

定时 16

读

读取保持寄存器 36
读取内部寄存器 36
读取输入状态 36
读取线圈状态 36

端

端口 12, 22, 35

短

短整数 46, 49
短整型 34

对

对于重复标记 18

非

非规范浮点数处理 10
非主动请求 19
非主动请求邮箱接收到不支持的请求。| IP 地址 = '<地址>'。 52
非主动请求邮箱内存分配错误。| IP 地址 = '<地址>'。 53

服

服务 45

浮

浮点数 46, 49
浮点型 34

父

父组 18

负

负载平衡 11

覆

覆盖 18

概

概述 5

功

功能代码说明 35

故

故障时降级 17

记

记录 31

将

将长整型作为十进制处理 25

降

降级超时 17
降级期间 17
降级时放弃请求 17

接

接收到无效块长度。| 块范围 = <开始> 至 <结束>。 55

解

解析导入文件中的记录时出错。| 记录编号 = <数字>，字段 = <字段>。 56

客

客户端应用程序已通过系统 tag _CEGExtension 更改 CEG 扩展。| 扩展 = '<扩展>'。 57

块

块大小 25
块读取字符串 26
块请求响应异常。| 块范围 = <地址> 到 <地址>，功能码 = <代码>，异常 = <代码>。 55
块请求响应异常。| 块范围 = <地址> 至 <地址>，异常 = <代码>。 55
块中的地址错误。| 块范围 = <地址> 至 <地址>。 54

来

来自缓存的初始更新 16

类

类属 Modbus 寻址 36

累

累积 45

连

连接超时 16

枚

枚举 30

每

每个设备的最大套接字数 12
每周期的事务数 11

名

名称 14

模

模拟 15

内

内部标记 35
内部寄存器 26, 37, 40, 47

强

强制单个线圈 36
强制多个线圈 36

请

请求超时 16

驱

驱动程序 14
驱动程序系统标记寻址 35

取

取消激活非法地址上的标记 21

全

全局设置 11

冗

冗余 27

扫

扫描模式 15

删

删除 18

设

设备间延迟 10
设备启动时 18
设备属性 - 标记生成 17
设备属性 - 定时 16
设备属性 - 冗余 27
设备属性 - 自动降级 16
设置 7, 22
设置值 31

生

生成 18

十

十进制寻址 46
十六进制寻址 46

时

时间 31

事

事件日志消息 52

首

首双字排序为低 24
首字排序为低 24

输

输出 45
输出线圈 26, 36, 40, 43, 47
输入线圈 26, 37, 40, 47

属

属性更改时 18

数

数据编码 24
数据访问 23
数据类型说明 34
数据收集 15
数组 46
数组支持 36-39, 43-44, 49

双

双精度 34
双字 34

说

说明 14

所

所有通道都已订阅虚拟网络，或者所有设备均在侦听远程地址，正在停止非主动请求通信。 58

套

套接字使用 12, 32

替

替换为零 10

通

通道处于虚拟网络中，所有设备都会恢复到“每台设备使用一个套接字”。 58
通道分配 14
通道级别设置 11
通道属性 - 常规 8
通道属性 - 高级 10
通道属性 - 通信序列化 10
通道属性 - 写入优化 9
通道属性 - 以太网通信 9
通信超时 16, 20

统

统计信息项 50

网

网络 1 - 网络 500 11
网络模式 11
网络适配器 9

为

为 Modbus 服务器设备创建了内存。| Modbus 服务器设备 ID = <设备>。 58

未

未修改 10

无

无法创建套接字连接。 53
无法解析主机。| 主机名 = '<名称>'。 54
无法启动 winsock 通信。 52
无法启动非主动请求通信。 52
无法写入地址，设备响应异常。| 地址 = '<地址>'，异常 = <代码>。 57
无符号双整型 31
无符号整型 31

系

系统 45
系统标记 35

写

写入非布尔标记的最新值 9
写入所有标记的所有值 9
写入所有标记的最新值 9

型

型号 6, 14

虚

虚拟网络 11

已

已屏蔽的写入寄存器 36

以

以太网 11, 22
以太网到 Modbus Plus 网桥 6
以太网管理器已启动。 57
以太网管理器已停止。 57
以太网设置 9

用

用户 45

优

优化 Modbus 以太网通信 32
优化方法 9
优先级 11

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>'。 57

邮

邮箱 6, 30
邮箱客户端权限 24
邮箱模式 49
邮箱寻址 46

与

与客户端相连时，无法将设备 ID 从 Modbus 服务器模式更改为客户端模式。 58
与客户端相连时，无法将设备 ID 从 Modbus 客户端模式更改为服务器模式。 58

预

预设单个寄存器 36

预设多个寄存器 36

允

允许子组 18

占

占空比 10

长

长整型 34, 46

针

针对导入文件中的记录截断了说明。| 记录编号 = <编号>。 56
针对未定义设备的非主动请求邮箱访问。正在关闭套接字。| IP 地址 = '<地址>'。 52

诊

诊断 8, 50

正

正在导入标记数据库。| 源文件 = '<文件名>'。 57
正在启动非主动请求通信。| 协议 = '<名称>'，端口 = <数字>。 58

支

支持 6

只

只写访问 49

指

指定的保持寄存器块大小超过了最大块大小。| 指定的块大小 = <数字> (寄存器)，最大块大小 = <数字> (寄存器)。 54

指定的内部寄存器块大小超过了最大块大小。| 指定的块大小 = <数字> (寄存器)，最大块大小 = <数字> (寄存器)。 54

指定的输出线圈块大小超过了最大块大小。| 指定的块大小 = <数字> (线圈)，最大块大小 = <数字> (线圈)。 54

指定的输入线圈块大小超过了最大块大小。| 指定的块大小 = <数字> (线圈)，最大块大小 = <数字> (线圈)。 54

自

自定义标记 31

自动标记数据库生成 31

自动降级 16

字

字 34

字符串 34

字符串支持 39, 44, 49

字节 31

字节切换后缀 39, 45

遵

遵循标签指定的扫描速率 15