

Omron Host Link 驱动程序

© 2018, PTC Inc. 保留所有权利。

目录

Omron Host Link 驱动程序	1
目录	2
Omron Host Link 驱动程序	3
概述	3
设置	4
信道属性	4
信道属性 - 常规	4
通道属性 - 串行通信	5
通道属性 - 写入优化	7
通道属性 - 高级	8
驱动程序设备属性	8
设备属性 - 常规	8
设备属性 - 扫描模式	10
设备属性 - 定时	10
设备属性 - 自动降级	11
设备属性 - 字符间延迟	11
设备属性 - 冗余	12
数据类型说明	13
地址说明	14
C20H 寻址	14
C200H 寻址	17
CQM1 寻址	20
开放寻址	23
事件日志消息	27
请求的类型与接收的类型不匹配。 标记地址 = '<地址>', 请求的类型 = <类型>, 接收的类型 = <类型>。	27
块中的地址错误。 标记地址 = '<地址>', 块 = <开始> 至 <结束>。	27
错误 掩码定义	27
索引	28

Omron Host Link 驱动程序

帮助版本 [1.037](#)

目录

[概述](#)

什么是 Omron Host Link 驱动程序？

[设备设置](#)

如何配置使用此驱动程序的设备？

[数据类型说明](#)

此驱动程序支持哪些数据类型？

[地址说明](#)

如何处理 Omron Host Link 设备上的数据位置？

[事件日志消息](#)

Omron Host Link 驱动程序 会产生哪些消息？

概述

Omron Host Link 驱动程序 提供将 Omron Host Link 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。其适用于 SYSMAC C 系列设备。

设置

支持的设备

C20H
C200H
CQM1
无限制

通信协议

Omron Host Link

支持的通信参数

波特率: 9600
奇偶校验: 偶或奇
数据位: 7 或 8
停止位: 1 或 2

信道和设备的最大数量

此驱动程序支持的最大信道数量为 100。所支持设备的最大数量为每信道 32 个。

以太网封装

此驱动程序支持“以太网封装”，允许驱动程序使用终端服务器与连接到以太网的串行设备进行通信。可以通过信道属性进行设置。有关详细信息，请参阅服务器帮助文档。

流量控制

使用 RS232/RS485 转换器时，所需的流量控制类型取决于转换器的需求。一些转换器不需要任何流量控制，而其他转换器则需要 RTS 流量。请参阅转换器的文档，以确定其流量要求。建议使用提供自动流量控制功能的 RS485 转换器。

● 注意：

1. 在使用制造商提供的通信电缆时，有时需要在“信道属性”中选取流量控制设置 **RTS** 或“**始终为 RTS**”。
2. 在未规定正确流量控制的平台上运行时，可能需要在服务器通信设置中设置流量控制。

[通道属性](#)

[设备属性](#)

信道属性

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信道。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信道组成。信道充当 OPC 链路的基础构建块。

与信道相关联的属性分为多个逻辑分组。某些组特定于给定的驱动程序或协议，而以下是通用组：

[常规](#)

[以太网或串行通信](#)

[写入优化](#)

[高级](#)

信道属性 - 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信道。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信道组成。信道充当 OPC 链路的基础构建块。此组用于指定常规信道属性，如标识属性和操作模式。

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

“标识”

“名称”: 此信道的用户定义标识。在每个服务器项目中，每个信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。信道名称是 OPC 浏览器信息的一部分。

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

“说明”: 有关此信道的用户定义信息。

● 这些属性 (包括 Description) 当中有很多具有关联的系统标记。

“驱动程序”(Driver): 为该信道选择的协议/驱动程序。该属性指定在信道创建期间选择的设备驱动程序。它在信道属性中为禁用设置。

● **注意**: 服务器全天在线运行时，可以随时更改这些属性。其中包括更改信道名称以防止客户端向服务器注册数据。如果客户端在信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信道名称更改之后发布项，并尝试通过原来的信道名称重新获取项，则该项将不被接受。考虑到这一点，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。利用“用户管理器”可防止操作员更改属性并限制对服务器功能的访问权限。

诊断

“诊断数据捕获”(Diagnostics Capture): 启用此选项后，信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。

● **注意**: 如果驱动程序不支持诊断，则该属性将被禁用。

● 有关详细信息，请参阅服务器帮助中的“通信诊断”。

通道属性 - 串行通信

串行通信属性可用于串行驱动程序，且随驱动程序、连接类型以及所选选项的不同而变化。以下是可能具有的属性的超集。

单击跳转至下列其中一个部分：[“连接类型”](#)、[“串行端口设置”](#)或[“以太网设置”](#)以及[“操作行为”](#)。

● **注意**: 服务器全天在线运行时，可以随时更改这些属性。由于对这些属性进行更改后可能会暂时中断通信，因此可通过“用户管理器”来限制对服务器功能的访问权限。

属性组	<input type="checkbox"/> 连接类型	
常规	物理媒体	COM 端口
串行通信	已共享	否
写优化	<input type="checkbox"/> 串行端口设置	
高级	COM ID	2
通信序列化	波特率	19200
链接设置	数据位	8
	奇偶性	无
	停止位	1
	流量控制	无
	<input type="checkbox"/> 操作行为	
	报告通信错误	启用

连接类型

“物理媒体”：选择用于数据通信的硬件设备的类型。选项包括“COM 端口”、“无”、“调制解调器”和“以太网封装”。默认选项为 COM 端口。

- **“无”**：选择“无”表示没有物理连接，此时将显示“[无通信的操作](#)”部分。
- **“COM 端口”**：选择“Com 端口”可显示和配置“[串行端口设置](#)”部分。
- **“调制解调器”**：当用电话线进行通信时，选择“调制解调器”，并在“[调制解调器设置](#)”部分中对该选项进行配置。
- **“以太网封装”**：选择是否将“以太网封装”用于通信，此时将显示“[以太网设置](#)”部分。
- **“共享”**：验证是否已将连接正确标识为与其他通道共享当前配置。为只读属性。

串行端口设置

“COM ID”：指定在与分配给通道的设备进行通信时要使用的通信 ID。有效范围为 1 至 9991 至 16。默认值为 1。

“波特率”：指定用于配置选定通信端口的波特率。

“数据位”：指定每个数据字的数据位数。选项包括 5、6、7 或 8。

“奇偶性”：指定数据的奇偶类型。选项包括“奇”(Odd)、“偶”(Even) 或“无”(None)。

“停止位”：指定每个数据字的停止位数。选项包括 1 或 2。

“流量控制”：选择 RTS 和 DTR 控制线的使用方式。在与一些串行设备进行通信时需要流量控制。选项包括：

- **“无”**：此选项不会切换或添加控制线。
- **“DTR”**：当通信端口打开并保持开启状态时，此选项将添加 DTR 线路。
- **“RTS”**：此选项指定，如果字节适用于传输，则 RTS 线路为高电平。在发送所有缓冲字节后，RTS 线路变为低电平。这通常用于 RS232/RS485 转换器硬件。
- **“RTS, DTR”**：此选项是 DTR 和 RTS 的组合选项。
- **“始终 RTS”**：当通信端口打开并保持开启状态时，此选项将添加 RTS 线路。
- **“RTS 手动”**：此选项将基于为“RTS 线路控制”输入的定时属性添加 RTS 线路。该选项仅在驱动程序支持手动 RTS 线路控制 (或属性共享且至少有一个通道属于提供此类支持的驱动程序) 时可用。“RTS 手动”添加“[RTS 线路控制](#)”属性时具有如下选项：
 - **“上升”**：该属性用于指定在数据传输前 RTS 线路上升为高电平所需的时间量。有效范围为 0 至 9999 毫秒。默认值为 10 毫秒。
 - **“下降”**：该属性用于指定在数据传输后 RTS 线路保持高电平的时间量。有效范围为 0 至 9999 毫秒。默认值为 10 毫秒。
 - **“轮询延迟”**：该属性用于指定通信轮询的延迟时间量。有效范围为 0 到 9999。默认值为 10 毫秒。

提示：在使用双线 RS-485 时，通信线路上可能会出现“回波”。由于此类通信不支持回波抑制，因此建议禁用回波或使用 RS-485 转换器。

操作行为

- **“报告通信错误”**：启用或禁用报告低级通信错误。启用时，如果出现低级错误，则会将其发布到“事件日志”。禁用时，即使正常请求失败，也不会发布这些相同的错误。默认设置为“启用”。
- **“关闭空闲连接”**：当通道上的客户端不再引用任何标记时，选择关闭通道连接。默认设置为“启用”。
- **“关闭前空闲时间”**：指定在移除所有标记后服务器在关闭 COM 端口前所等待的时间。默认值为 15 秒。

以太网设置

注意：不是所有的串行驱动程序都支持以太网封装。若此组未出现，则无法支持相关功能。

如果要同与以太网终端服务器相连的串行设备进行通信，则可通过“以太网封装”来实现。终端服务器本质上是将以太网上的 TCP/IP 消息转换为串行数据的虚拟串行端口。消息转换完毕后，用户可将支持串行通信的

标准设备连接到终端服务器。必须对终端服务器的串行端口进行正确配置，以满足所连串行设备的要求。有关详细信息，请参阅服务器帮助中的“如何使用以太网封装”。

- **“网络适配器”**：用于指示此通道中以太网设备绑定的网络适配器。选择要绑定的网络适配器，或者允许操作系统选择默认项。
 ● 某些特定的驱动程序可能会显示其他“以太网封装”属性。有关详细信息，请参阅“信道属性 - 以太网封装”。

“调制解调器设置”

- **“调制解调器”**：指定用于通信的已安装调制解调器。
- **“连接超时”**：指定读取或写入失败前建立连接所等待的时间。默认值为 60 秒。
- **“调制解调器属性”**：配置调制解调器硬件。单击该选项后，将打开供应商特定的调制解调器属性。
- **“自动拨号”**：启用自动拨打电话簿中的条目。默认设置为“禁用”。有关详细信息，请参阅服务器帮助中的“调制解调器自动拨号”。
- **“报告通信错误”**：启用或禁用报告低级通信错误。启用时，如果出现低级错误，则会将其发布到“事件日志”。禁用时，即使正常请求失败，也不会发布这些相同的错误。默认设置为“启用”。
- **“关闭空闲连接”**：当通道上的客户端不再引用任何标记时，选择关闭调制解调器连接。默认设置为“启用”。
- **“关闭前空闲时间”**：指定在移除所有标记后服务器在关闭调制解调器连接前所等待的时间。默认值为 15 秒。

“无通信的操作”

- **“读取处理”**：选择要在请求显式设备读取时执行的操作。选项包括“忽略”和“失败”。“忽略”不执行任何操作；“失败”会为客户端提供一条指示失败的更新信息。默认设置为“忽略”。

通道属性 - 写入优化

与任何 OPC 服务器一样，将数据写入设备可能是应用程序应具备的最重要的功能。服务器旨在确保从客户端应用程序写入的数据能够准时发送到设备。为了达到此目标，服务器提供了可用来满足特定需求以提高应用程序响应能力的优化属性。

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

写入优化

“优化方法”(Optimization Method)：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”(Write Only Latest Value for Non-Boolean Tags)**：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”(Write Only Latest Value for All Tags)**：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

此组用于指定高级信道属性。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"/> 非规范浮点数处理	
常规	浮点值	替换为零
以太网通信	<input type="checkbox"/> 设备间延迟	
写优化	设备间延迟 (毫秒)	0
高级		
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) 或不正规编号。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 不正规、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。根据信道的浮点规范化设置，将仅对实时驱动程序标记 (如值和数组) 进行浮点规范化。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“如何使用非规范化浮点值”。

“设备间延迟”：指定在接收到同一信道上的当前设备发出的数据后，通信信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。

驱动程序设备属性

设备属性将被分成组。有关该组中设置的详细信息，请单击下面的链接。

[标识](#)

[操作模式](#)

[扫描模式](#)

[通信超时](#)

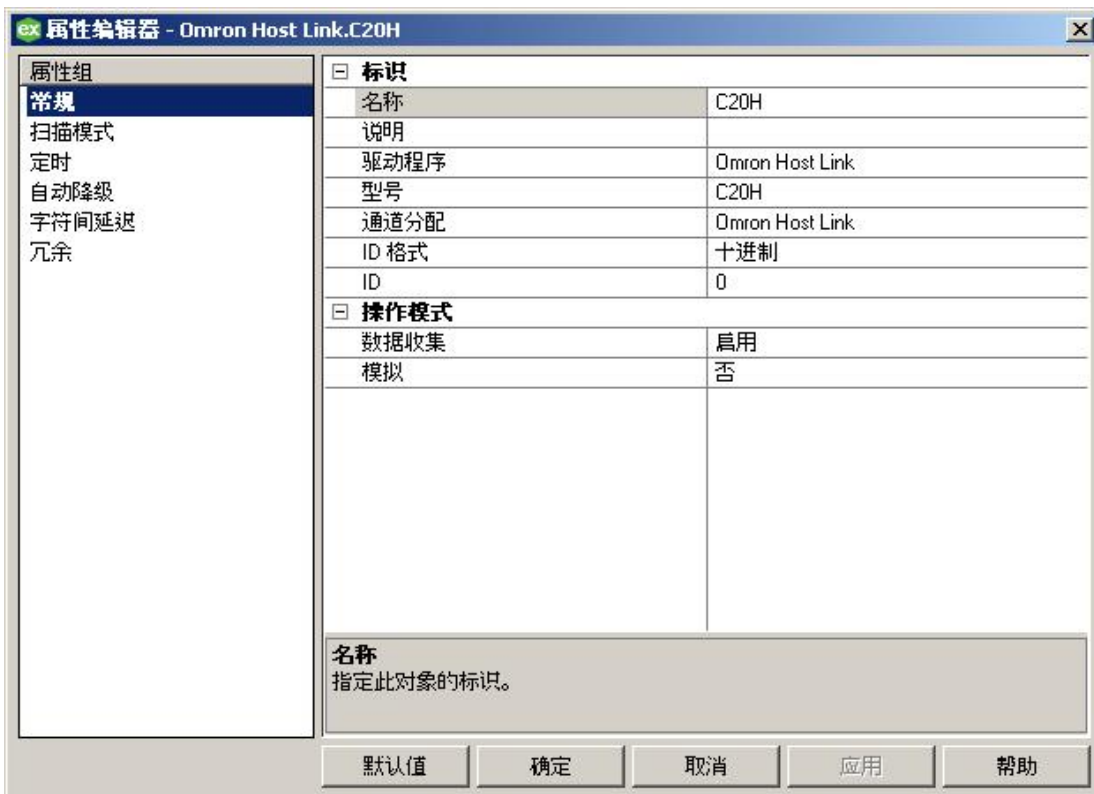
[自动降级](#)

[字符间延迟](#)

[通信参数](#)

[冗余](#)

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。

“型号”: 设备的特定版本。

●有关支持 *FINS* 通信服务的型号列表, 请参阅制造商网站。

“ID 格式”: 选择设备标识采用的格式。选项包括“十进制”、“八进制”和“十六进制”。

ID: ID 指定设备的唯一单元编号。有效范围为 0 到 31。

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”。

●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。 “系统”标记允许从客户端监控此属性。

- 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	<input checked="" type="checkbox"/> 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”：此模式可指定要使用的最大扫描速率。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。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	<input checked="" type="checkbox"/> 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	重试次数	3
自动降级	<input checked="" type="checkbox"/> 定时	
冗余	请求间延迟 (毫秒)	0

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”(Request Timeout)：此属性可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 9,999,999 毫秒 (167.6667 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”：此属性用于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”(Inter-Request Delay)：此属性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信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信道。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。

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信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。

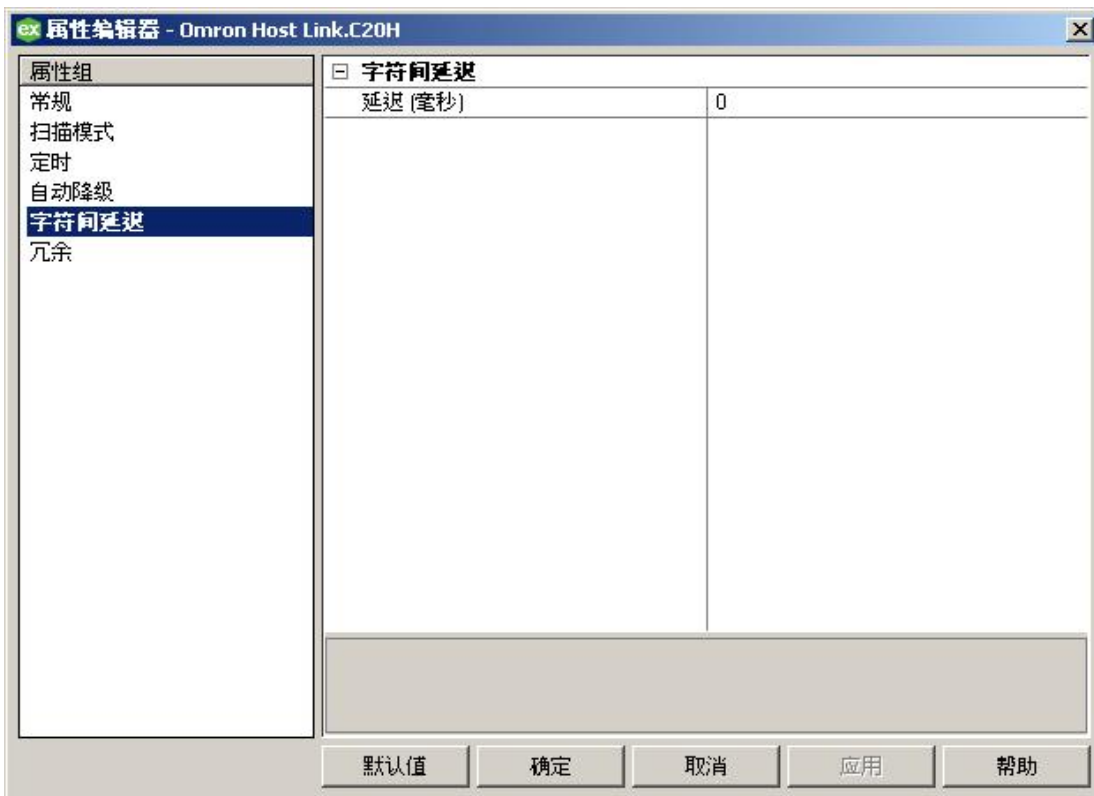
“降级超时”：指定在对设备取消扫描之前，请求超时和重试的连续周期数。有效范围是 1 到 30 次连续失败。默认值为 3。

“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 - 字符间延迟

字符间延迟适用于调制解调器通信，但某些特定设备在通过调制解调器通信时可能不需要延迟。此值带来的延迟会降低通信速度。



“延迟”: 用于指定驱动程序发送给 Omron 设备的每个字符之间的控制延迟。有效范围为 0 至 200 毫秒。默认设置为 0 毫秒。

提示: 通常建议将此值保留为默认设置; 然而, 在通过调制解调器使用驱动程序时可调整该值以确保成功通信。对于波特率为 1200 的调制解调器连接, 典型设置为 30 毫秒。

设备属性 - 冗余



Media-Level Redundancy 插件提供冗余。

提示: 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。

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
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。
长整型	有符号 32 位值
双字型	无符号 32 位值
浮点型	32 位浮点值 驱动程序将两个连续寄存器解释为浮点值，方法是将第二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 DM100 指定为浮点型，则寄存器 DM100 的位 0 将是 32 位浮点数的位 0，寄存器 DM101 的位 15 将是 32 位浮点数的位 31。
字符串	空终止 ASCII 字符串。 支持多达 58 个字符的字符串长度，并支持选择由高到低的字节排序、由低到高的字节排序、仅高位字节和仅低位字节。

地址说明

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

[C20H 寻址](#)

[C200H 寻址](#)

[CQM1 寻址](#)

[开放寻址](#)

C20H 寻址

默认数据类型以**粗体**显示。

● 有关详细信息，请参阅[字符串支持](#)和[数组支持](#)。

设备类型	范围	数据类型	访问
辅助继电器	AR00-AR27 AR00-AR26 ARxx.00-ARxx.15	字 、短整型、BCD、 长整型、 双字型、浮点型、 LBCD 布尔型	读/ 写
辅助继电器为字符串型 (字节排序由高到低)	AR00.56H-AR27.02H .I 是字符串长度，范围介于 2 到 56 个字符之间	字符串	读/ 写
辅助继电器为字符串型 (字节排序由低到高)	AR00.56L-AR27.02L .I 是字符串长度，范围介于 2 到 56 个字符之间	字符串	读/ 写
辅助继电器为字符串型 (仅使用每个字的高位字节)	AR00.28D-AR27.01D .I 是字符串长度，范围介于 1 到 28 个字符之间	字符串	读/ 写
辅助继电器为字符串型 (仅使用每个字的低位字节)	AR00.28E-AR27.01E .I 是字符串长度，范围介于 1 到 28 个字符之间	字符串	读/ 写
数据内存	DM0000-DM0999 DM0000-DM0998 DMxxxx.00-DMxxxx.15 DM1000-DM1999 DM1000-DM1998 DMxxxx.00-DMxxxx.15	字 、短整型、BCD、 长整型、 双字型、浮点型、 LBCD 布尔型 字 、短整型、BCD、 长整型、 双字型、浮点型、 LBCD 布尔型	读/ 写 只读
数据内存为具有高低点字节顺序字符串	DM0000.58H-DM0999.02H .I 是字符串长度，范围介于 2 到 58 个字符之间 DM1000.58H-DM1999.02H	字符串 字符串	读/ 写 只读

设备类型	范围	数据类型	访问
	.I 是字符串长度, 范围介于 2 到 58 个字符之间		
数据内存为具有 LoHi 字节顺序字符串	DM0000.58L-DM0999.02L .I 是字符串长度, 范围介于 2 到 58 个字符之间	字符串	读/写
	DM1000.58L-DM1999.02L .I 是字符串长度, 范围介于 2 到 58 个字符之间	字符串	只读
数据内存为字符串型 (仅使用每个字的高位字节)	DM0000.29D-DM0999.01D .I 是字符串长度, 范围介于 1 到 29 个字符之间	字符串	读/写
	DM1000.29D-DM1999.01D .I 是字符串长度, 范围介于 1 到 29 个字符之间	字符串	只读
数据内存为字符串型 (仅使用每个字的低位字节)	DM0000.29E-DM0999.01E .I 是字符串长度, 范围介于 1 到 29 个字符之间	字符串	读/写
	DM1000.29E-DM1999.01E .I 是字符串长度, 范围介于 1 到 29 个字符之间	字符串	只读
保持继电器	HR00-HR99 HR00-HR98 HRxx.00-HRxx.15	字、短整型、BCD、长整型、双字型、浮点型、LBCD 布尔型	读/写
保持继电器为字符串型 (字节排序由高到低)	HR00.58H-HR99.02H .I 是字符串长度, 范围介于 2 到 58 个字符之间	字符串	读/写
保持继电器为字符串型 (字节排序由低到高)	HR00.58L-HR99.02L .I 是字符串长度, 范围介于 2 到 58 个字符之间	字符串	读/写
保持继电器为字符串型 (仅使用每个字的高位字节)	HR00.29D-HR99.01D .I 是字符串长度, 范围介于 1 到 29 个字符之间	字符串	读/写
保持继电器为字符串型 (仅使用每个字的低位字节)	HR00.29E-HR99.01E .I 是字符串长度, 范围介于 1 到 29 个字符之间	字符串	读/写
内部继电器	IR000-IR255 IR000-IR254 IRxxx.00-IRxxx.15	字、短整型、BCD、长整型 双字型、浮点型、LBCD	读/写

设备类型	范围	数据类型	访问
		布尔型	
内部继电器为字符串型 (字节排序由高到低)	IR000.58H-IR255.02H .I 是字符串长度, 范围介于 2 到 58 个字符之间	字符串	读/写
内部继电器为字符串型 (字节排序由低到高)	IR000.58L-IR255.02L .I 是字符串长度, 范围介于 2 到 58 个字符之间	字符串	读/写
内部继电器为字符串型 (仅使用每个字的高位字节)	IR000.29D-IR255.01D .I 是字符串长度, 范围介于 1 到 29 个字符之间	字符串	读/写
内部继电器为字符串型 (仅使用每个字的低位字节)	IR000.29E-IR255.01E .I 是字符串长度, 范围介于 1 到 29 个字符之间	字符串	读/写
链路继电器	LR00-LR63 LR00-LR62 LRxx.00-LRxx.15	字、短整型、BCD、长整型、双字型、浮点型、LBCD 布尔型	读/写
链路继电器为字符串型 (字节排序由高到低)	LR00.58H-LR63.02H .I 是字符串长度, 范围介于 2 到 58 个字符之间	字符串	读/写
链路继电器为字符串型 (字节排序由低到高)	LR00.58L-LR63.02L .I 是字符串长度, 范围介于 2 到 58 个字符之间	字符串	读/写
链路继电器为字符串型 (仅使用每个字的高位字节)	LR00.29D-LR63.01D .I 是字符串长度, 范围介于 1 到 29 个字符之间	字符串	读/写
链路继电器为字符串型 (仅使用每个字的低位字节)	LR00.29E-LR63.01E .I 是字符串长度, 范围介于 1 到 29 个字符之间	字符串	读/写
计时器/计数器	RC000-RC511 RCxxx.00-RCxxx.15	字、短整型、BCD 布尔型	读/写
计时器/计数器状态	TC000-TC511	布尔型	读/写

字符串支持

C20H 模型支持以 ASCII 字符串的形式读取和写入辅助继电器 (AR)、数据内存 (DM)、保持继电器 (HR)、内部继电器 (IR) 和链路继电器 (LR)。当将其中任何一种设备类型用于字符串数据时, 每个寄存器可以包含 ASCII 数据的两个字节 (两个字符) 或一个。定义字符串时, 可以选择指定寄存器中 ASCII 数据的顺序或要在指定寄存器中使用的字节。

如果每个寄存器使用两字节 ASCII 数据, 则字符串长度可以为 2 到 58 个字符 (对于 AR 为 2 到 56), 并代替位数输入。长度必须输入为偶数。字符串跨越的寄存器范围不得超过设备类型的范围。通过将 "H" 或 "L" 附加到地址来指定字节顺序。

如果每个寄存器使用一字节 ASCII 数据，则字符串长度可以为 1 到 29 个字符（对于 AR 为 1 到 28），并代替位数输入。字符串跨越的寄存器范围不得超过设备类型的范围。通过将 "D" 或 "E" 附加到地址来指定要在寄存器内使用的字节。

示例

- 对于从 DM100 开始、长度为 50 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：DM100.50H
- 对于从 DM110 开始、长度为 8 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：DM110.08L
- 对于从 DM200 开始、长度为 15 字节并采用仅高位字节的字符串，要对其进行寻址，请输入：DM200.15D
- 对于从 DM220 开始、长度为 7 字节并采用仅低位字节的字符串，要对其进行寻址，请输入：DM220.07E

数组支持

除了“布尔型”和“字符串”之外，所有数据类型都支持数组。有关数组寻址的方法有两种。以下示例为根据数据内存位置。

DMxxxx [行数] [列数]
DMxxxx [列数]*

*此方法假定“行数”等于 1。

行数、列数与以字节为单位的数据大小的乘积不能超过 116 个字节。此限制由协议规定。由于此驱动程序使用 ASCII 协议，因此每个字、短整型和 BCD 对应 4 个字节，而每个双字型、长整型、LBCD 和浮点型对应 8 个字节。例如，对于数组大小为 28 字的 4 X 7 数组，如果每个字乘以 4 字节，那么数组大小将为 112 字节，即在 116 字节的最大请求大小范围内。

✿ 修改 32 位值（双字型、长整型、LBCD 和浮点型）时请小心。允许这些数据类型的每个地址在设备内的字偏移处开始。因此，双字型 DM0 和 DM1 在字 DM1 处重叠。写入 DM0 还会修改保存在 DM1 中的值。建议用户使用这些数据类型的，以避免发生重叠。例如，在使用双字型时，用户最好使用 DM0、DM2、DM4 等字，以防止“字”重叠。

C200H 寻址

默认数据类型以**粗体**显示。

● 有关详细信息，请参阅[字符串支持](#)和[数组支持](#)。

设备类型	范围	数据类型	访问
辅助继电器	AR000-AR999 AR000-AR998 ARxxx.00-ARxxx.15	字、短整型、BCD、 长整型、 双字型、浮点型、 LBCD 布尔型	读/ 写
辅助继电器为字符串型 (字节排序由高到低)	AR000.58H-AR999.02H .I 是字符串长度，范围介于 2 到 58 个字符之间	字符串	读/ 写
辅助继电器为字符串型 (字节排序由低到高)	AR000.58L-AR999.02L .I 是字符串长度，范围介于 2 到 58 个字符之间	字符串	读/ 写
辅助继电器为字符串型 (仅使用每个字的高位字节)	AR000.29D-AR999.01D .I 是字符串长度，范围介于 1 到 29 个字符之间	字符串	读/ 写

设备类型	范围	数据类型	访问
辅助继电器为字符串型 (仅使用每个字的低位字节)	AR000.29E-AR999.01E .I 是字符串长度, 范围介于 1 到 29 个字符之间	字符串	读/写
数据内存	DM0000-DM9999 DM0000-DM9998 DMxxxx.00-DMxxxx.15	字、短整型、BCD、长整型、双字型、浮点型、LBCD 布尔型	读/写
数据内存为字符串型 (字节排序由高到低)	DM0000.58H-DM0999.02H .I 是字符串长度, 范围介于 2 到 58 个字符之间	字符串	读/写
数据内存为具有 LoHi 字节顺序字符串	DM0000.58L-DM0999.02L .I 是字符串长度, 范围介于 2 到 58 个字符之间	字符串	读/写
数据内存为字符串型 (仅使用每个字的高位字节)	DM0000.29D-DM0999.01D .I 是字符串长度, 范围介于 1 到 29 个字符之间	字符串	读/写
数据内存为字符串型 (仅使用每个字的低位字节)	DM0000.29E-DM0999.01E .I 是字符串长度, 范围介于 1 到 29 个字符之间	字符串	读/写
保持继电器	HR000-HR999 HR000-HR998 HRxxx.00-HRxxx.15	字、短整型、BCD、长整型、双字型、浮点型、LBCD 布尔型	读/写
保持继电器为字符串型 (字节排序由高到低)	HR000.58H- HR999.02H .I 是字符串长度, 范围介于 2 到 58 个字符之间	字符串	读/写
保持继电器为字符串型 (字节排序由低到高)	HR000.58L-HR999.02L .I 是字符串长度, 范围介于 2 到 58 个字符之间	字符串	读/写
保持继电器为字符串型 (仅使用每个字的高位字节)	HR000.29D-HR999.01D .I 是字符串长度, 范围介于 1 到 29 个字符之间	字符串	读/写
保持继电器为字符串型 (仅使用每个字的低位字节)	HR000.29E-HR999.01E .I 是字符串长度, 范围介于 1 到 29 个字符之间	字符串	读/写
内部继电器	IR000-IR999 IR000-IR998 IRxxx.00-IRxxx.15	字、短整型、BCD、长整型、双字型、浮点型、LBCD 布尔型	读/写

设备类型	范围	数据类型	访问
内部继电器为字符串型 (字节排序由高到低)	IR000.58H-IR999.02H .I 是字符串长度, 范围介于 2 到 58 个字符之间	字符串	读/写
内部继电器为字符串型 (字节排序由低到高)	IR000.58L-IR999.02L .I 是字符串长度, 范围介于 2 到 58 个字符之间	字符串	读/写
内部继电器为字符串型 (仅使用每个字的高位字节)	IR000.29D-IR999.01D .I 是字符串长度, 范围介于 1 到 29 个字符之间	字符串	读/写
内部继电器为字符串型 (仅使用每个字的低位字节)	IR000.29E-IR999.01E .I 是字符串长度, 范围介于 1 到 29 个字符之间	字符串	读/写
链路继电器	LR000-LR999 LR000-LR998 LRxxx.00-LRxxx.15	字、短整型、BCD、长整型、双字型、浮点型、LBCD 布尔型	读/写
链路继电器为字符串型 (字节排序由高到低)	LR000.58H-LR999.02H .I 是字符串长度, 范围介于 2 到 58 个字符之间	字符串	读/写
链路继电器为字符串型 (字节排序由低到高)	LR000.58L-LR999.02L .I 是字符串长度, 范围介于 2 到 58 个字符之间	字符串	读/写
链路继电器为字符串型 (仅使用每个字的高位字节)	LR000.29D-LR999.01D .I 是字符串长度, 范围介于 1 到 29 个字符之间	字符串	读/写
链路继电器为字符串型 (仅使用每个字的低位字节)	LR000.29E-LR999.01E .I 是字符串长度, 范围介于 1 到 29 个字符之间	字符串	读/写
计时器/计数器	RC000-RC999 RCxxx.00-RCxxx.15	字、短整型、BCD 布尔型	读/写
计时器/计数器状态	TC000-TC999	布尔型	读/写

字符串支持

C200H 模型支持以 ASCII 字符串的形式读取和写入辅助继电器 (AR)、数据内存 (DM)、保持继电器 (HR)、内部继电器 (IR) 和链路继电器 (LR)。当将其中任意一种设备类型用于字符串数据时, 每个寄存器可以包含 ASCII 数据的两个字节 (两个字符) 或一个。定义字符串时, 可以选择指定寄存器中 ASCII 数据的顺序或要在指定寄存器中使用的字节。

如果每个寄存器使用两字节 ASCII 数据, 则字符串的长度可以为 2 到 58 个字符, 并代替位数输入。长度必须输入为偶数。字符串跨越的寄存器范围不得超过设备类型的范围。通过将 "H" 或 "L" 附加到地址来指定字节顺序。

如果每个寄存器使用一字节 ASCII 数据，则字符串的长度可以为 1 到 29 个字符，并代替位数输入。字符串跨越的寄存器范围不得超过设备类型的范围。通过将 "D" 或 "E" 附加到地址来指定要在寄存器内使用的字节。

示例

- 对于从 DM100 开始、长度为 50 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：DM100.50H
- 对于从 DM110 开始、长度为 8 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：DM110.08L
- 对于从 DM200 开始、长度为 15 字节并采用仅高位字节的字符串，要对其进行寻址，请输入：DM200.15D
- 对于从 DM220 开始、长度为 7 字节并采用仅低位字节的字符串，要对其进行寻址，请输入：DM220.07E

数组支持

除了“布尔型”和“字符串”之外，所有数据类型都支持数组。有关数组寻址的方法有两种。以下示例为根据数据内存位置。

DMxxxx [行数] [列数]
DMxxxx [列数]*

*此方法假定“行数”等于 1。

行数、列数与以字节为单位的数据大小的乘积不能超过 116 个字节。此限制由协议规定。由于此驱动程序使用 ASCII 协议，因此每个字、短整型和 BCD 对应 4 个字节，而每个双字型、长整型、LBCD 和浮点型对应 8 个字节。例如，对于数组大小为 28 字的 4 X 7 字数组，如果每个字乘以 4 字节，那么数组大小将为 112 字节，即在 116 字节的最大请求大小范围内。

✿ 修改 32 位值 (双字型、长整型、LBCD 和浮点型) 时请小心。允许这些数据类型的每个地址在设备内的字偏移处开始。因此，双字型 DM0 和 DM1 在字 DM1 处重叠。写入 DM0 还会修改保存在 DM1 中的值。建议用户使用这些数据类型，以避免发生重叠。例如，当使用双字型时，用户可能希望使用 DM0、DM2、DM4 等字，以防止“字”重叠。

CQM1 寻址

默认数据类型以**粗体**显示。

● 有关详细信息，请参阅[字符串支持](#)和[数组支持](#)。

设备类型	范围	数据类型	访问
辅助继电器	AR00-AR27 AR00-AR26 ARxx.00-ARxx.15	字 、短整型、BCD、 长整型 双字型、浮点型、 LBCD 布尔型	读/ 写
辅助继电器为字符串型 (字节排序由高到低)	AR00.56H-AR27.02H .I 是字符串长度，范围介于 2 到 56 个字符之间	字符串	读/ 写
辅助继电器为字符串型 (字节排序由低到高)	AR00.56L-AR27.02L .I 是字符串长度，范围介于 2 到 56 个字符之间	字符串	读/ 写
辅助继电器为字符串型 (仅使用每个字的高位字节)	AR00.28D-AR27.01D .I 是字符串长度，范围介于 1 到 28	字符串	读/ 写

设备类型	范围	数据类型	访问
	个字符之间		
辅助继电器为字符串型 (仅使用每个字的低位字节)	AR00.28E-AR27.01E .I 是字符串长度, 范围介于 1 到 28 个字符之间	字符串	读/写
数据内存	DM0000-DM6655 DM0000-DM6654 DMxxxx.00-DMxxxx.15	字、短整型、BCD、长整型 双字型、浮点型、LBCD 布尔型	读/写
数据内存为具有高低点字节顺序字符串	DM0000.58H-DM6655.02H .I 是字符串长度, 范围介于 2 到 58 个字符之间	字符串	读/写
数据内存为具有 LoHi 字节顺序字符串	DM0000.58L-DM6655.02L .I 是字符串长度, 范围介于 2 到 58 个字符之间	字符串	读/写
数据内存为字符串型 (仅使用每个字的高位字节)	DM0000.29D-DM6655.01D .I 是字符串长度, 范围介于 1 到 29 个字符之间	字符串	读/写
数据内存为字符串型 (仅使用每个字的低位字节)	DM0000.29E-DM6655.01E .I 是字符串长度, 范围介于 1 到 29 个字符之间	字符串	读/写
保持继电器	HR00-HR99 HR00-HR98 HRxx.00-HRxx.15	字、短整型、BCD、长整型 双字型、浮点型、LBCD 布尔型	读/写
保持继电器为字符串型 (字节排序由高到低)	HR00.58H-HR99.02H .I 是字符串长度, 范围介于 2 到 58 个字符之间	字符串	读/写
保持继电器为字符串型 (字节排序由低到高)	HR00.58L-HR99.02L .I 是字符串长度, 范围介于 2 到 58 个字符之间	字符串	读/写
保持继电器为字符串型 (仅使用每个字的高位字节)	HR00.29D-HR99.01D .I 是字符串长度, 范围介于 1 到 29 个字符之间	字符串	读/写
保持继电器为字符串型 (仅使用每个字的低位字节)	HR00.29E-HR99.01E .I 是字符串长度, 范围介于 1 到 29 个字符之间	字符串	读/写
内部继电器	IR000-IR255	字、短整型、BCD、长整型	读/写

设备类型	范围	数据类型	访问
	IR000-IR254 IRxxx.00-IRxxx.15	双字型、浮点型、LBCD 布尔型	
内部继电器为字符串型 (字节排序由高到低)	IR000.58H-IR255.02H .I 是字符串长度, 范围介于 2 到 58 个字符之间	字符串	读/写
内部继电器为字符串型 (字节排序由低到高)	IR000.58L-IR255.02L .I 是字符串长度, 范围介于 2 到 58 个字符之间	字符串	读/写
内部继电器为字符串型 (仅使用每个字的高位字节)	IR000.29D-IR255.01D .I 是字符串长度, 范围介于 1 到 29 个字符之间	字符串	读/写
内部继电器为字符串型 (仅使用每个字的低位字节)	IR000.29E-IR255.01E .I 是字符串长度, 范围介于 1 到 29 个字符之间	字符串	读/写
链路继电器	LR00-LR63 LR00-LR62 LRxx.00-LRxx.15	字、短整型、BCD、长整型 双字型、浮点型、LBCD 布尔型	读/写
链路继电器为字符串型 (字节排序由高到低)	LR00.58H-LR63.02H .I 是字符串长度, 范围介于 2 到 58 个字符之间	字符串	读/写
链路继电器为字符串型 (字节排序由低到高)	LR00.58L-LR63.02L .I 是字符串长度, 范围介于 2 到 58 个字符之间	字符串	读/写
链路继电器为字符串型 (仅使用每个字的高位字节)	LR00.29D-LR63.01D .I 是字符串长度, 范围介于 1 到 29 个字符之间	字符串	读/写
链路继电器为字符串型 (仅使用每个字的低位字节)	LR00.29E-LR63.01E .I 是字符串长度, 范围介于 1 到 29 个字符之间	字符串	读/写
计时器/计数器	RC000-RC511 RCxxx.00-RCxxx.15	字、短整型、BCD 布尔型	读/写
计时器/计数器状态	TC000-TC511	布尔型	读/写

字符串支持

CQM1 模型支持以 ASCII 字符串的形式读取和写入辅助继电器 (AR)、数据内存 (DM)、保持继电器 (HR)、内部继电器 (IR) 和链路继电器 (LR)。当将其中任意一种设备类型用于字符串数据时, 每个寄存器可以包含 ASCII 数据的两个字节 (两个字符) 或一个。定义字符串时, 可以选择指定寄存器中 ASCII 数据的顺序或要在指定寄存器中使用的字节。

如果每个寄存器使用两字节 ASCII 数据，则字符串长度可以为 2 到 58 个字符（对于 AR 为 2 到 56），并代替位数输入。长度必须输入为偶数。字符串跨越的寄存器范围不得超过设备类型的范围。通过将 "H" 或 "L" 附加到地址来指定字节顺序。

如果每个寄存器使用一字节 ASCII 数据，则字符串长度可以为 1 到 29 个字符（对于 AR 为 1 到 28），并代替位数输入。字符串跨越的寄存器范围不得超过设备类型的范围。通过将 "D" 或 "E" 附加到地址来指定要在寄存器内使用的字节。

示例

- 对于从 DM100 开始、长度为 50 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：DM100.50H
- 对于从 DM110 开始、长度为 8 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：DM110.08L
- 对于从 DM200 开始、长度为 15 字节并采用仅高位字节的字符串，要对其进行寻址，请输入：DM200.15D
- 对于从 DM220 开始、长度为 7 字节并采用仅低位字节的字符串，要对其进行寻址，请输入：DM220.07E

数组支持

除了“布尔型”和“字符串”之外，所有数据类型都支持数组。有关数组寻址的方法有两种。以下示例为根据数据内存位置。

DMxxxx [行数] [列数]
DMxxxx [列数]*

*此方法假定“行数”等于 1。

行数、列数与以字节为单位的数据大小的乘积不能超过 116 个字节。此限制由协议规定。由于此驱动程序使用 ASCII 协议，因此每个字、短整型和 BCD 对应 4 个字节，而每个双字型、长整型、LBCD 和浮点型对应 8 个字节。例如，对于数组大小为 28 字的 4 X 7 字数组，如果每个字乘以 4 字节，那么数组大小将为 112 字节，即在 116 字节的最大请求大小范围内。

● 修改 32 位值（双字型、长整型、LBCD 和浮点型）时请小心。允许这些数据类型的每个地址在设备内的字偏移处开始。因此，双字型 DM0 和 DM1 在字 DM1 处重叠。写入 DM0 还会修改保存在 DM1 中的值。建议用户使用这些数据类型，以避免发生重叠。例如，当使用双字型时，用户可能希望使用 DM0、DM2、DM4 等字，以防止“字”重叠。

开放寻址

对于所有内存类型，以下内存映射均处于打开状态，以便支持较新的设备。请参阅制造商文档来了解特定于设备的地址范围。默认数据类型以**粗体**显示。

● 有关详细信息，请参阅[字符串支持](#)和[数组支持](#)。

设备类型	范围	数据类型	访问
辅助继电器	AR0000-AR9999 AR0000-AR9998 ARxxxx.00-ARxxxx.15	字、短整型、BCD、长整型、 双字型、浮点型、LBCD 布尔型	读 / 写
辅助继电器为字符串型 (字节排序由高到低)	AR0000.58H-AR9999.02H .I 是字符串长度，范围介于 2 到 58 个字符之间	字符串	读 / 写
辅助继电器为字符串型 (字节排序由低到高)	AR0000.58L-AR9999.02L .I 是字符串长度，范围介于 2 到 58 个字符之间	字符串	读 / 写

设备类型	范围	数据类型	访问
辅助继电器为字符串型 (仅使用每个字的高位字节)	AR0000.29D-AR9999.01D .I 是字符串长度, 范围介于 1 到 29 个字符之间	字符串	读 / 写
辅助继电器为字符串型 (仅使用每个字的低位字节)	AR0000.29E-AR9999.01E .I 是字符串长度, 范围介于 1 到 29 个字符之间	字符串	读 / 写
数据内存	DM0000-DM9999 DM0000-DM9998 DMxxxx.00-DMxxxx.15	字、短整型、BCD、长整型、双字型、浮点型、LBCD 布尔型	读 / 写
数据内存为字符串型 (字节排序由高到低)	DM0000.58H-DM9999.02H .I 是字符串长度, 范围介于 2 到 58 个字符之间	字符串	读 / 写
数据内存为字符串型 (字节排序由低到高)	DM0000.58L-DM9999.02L .I 是字符串长度, 范围介于 2 到 58 个字符之间	字符串	读 / 写
数据内存为字符串型 (仅使用每个字的高位字节)	DM0000.29D-DM9999.01D .I 是字符串长度, 范围介于 1 到 29 个字符之间	字符串	读 / 写
数据内存为字符串型 (仅使用每个字的低位字节)	DM0000.29E-DM9999.01E .I 是字符串长度, 范围介于 1 到 29 个字符之间	字符串	读 / 写
保持继电器	HR0000-HR9999 HR0000-HR9998 HRxxxx.00-HRxxxx.15	字、短整型、BCD、长整型、双字型、浮点型、LBCD 布尔型	读 / 写
保持继电器为字符串型 (字节排序由高到低)	HR0000.58H-HR9999.02H .I 是字符串长度, 范围介于 2 到 58 个字符之间	字符串	读 / 写
保持继电器为字符串型 (字节排序由低到高)	HR0000.58L-HR9999.02L .I 是字符串长度, 范围介于 2 到 58 个字符之间	字符串	读 / 写
保持继电器为字符串型 (仅使用每个字的高位字节)	HR0000.29D-HR9999.01D .I 是字符串长度, 范围介于 1 到 29 个字符之间	字符串	读 / 写
保持继电器为字符串型 (仅使用每个字的低位字节)	HR0000.29E-HR9999.01E .I 是字符串长度, 范围介于 1 到 29 个字符之间	字符串	读 / 写
内部继电器	IR0000-IR9999 IR0000-IR9998 IRxxxx.00-IRxxxx.15	字、短整型、BCD、长整型、双字型、浮点型、LBCD 布尔型	读 / 写

设备类型	范围	数据类型	访问
内部继电器为字符串型 (字节排序由高到低)	IR0000.58H-IR9999.02H .I 是字符串长度, 范围介于 2 到 58 个字符之间	字符串	读 / 写
内部继电器为字符串型 (字节排序由低到高)	IR0000.58L-IR9999.02L .I 是字符串长度, 范围介于 2 到 58 个字符之间	字符串	读 / 写
内部继电器为字符串型 (仅使用每个字的高位字节)	IR0000.29D-IR9999.01D .I 是字符串长度, 范围介于 1 到 29 个字符之间	字符串	读 / 写
内部继电器为字符串型 (仅使用每个字的低位字节)	IR0000.29E-IR9999.01E .I 是字符串长度, 范围介于 1 到 29 个字符之间	字符串	读 / 写
链路继电器	LR0000-LR9999 LR0000-LR9998 LRxxxx.00-LRxxxx.15	字、短整型、BCD、长整型、双字型、浮点型、LBCD 布尔型	读 / 写
链路继电器为字符串型 (字节排序由高到低)	LR0000.58H-LR9999.02H .I 是字符串长度, 范围介于 2 到 58 个字符之间	字符串	读 / 写
链路继电器为字符串型 (字节排序由低到高)	LR0000.58L-LR9999.02L .I 是字符串长度, 范围介于 2 到 58 个字符之间	字符串	读 / 写
链路继电器为字符串型 (仅使用每个字的高位字节)	LR0000.29D-LR9999.01D .I 是字符串长度, 范围介于 1 到 29 个字符之间	字符串	读 / 写
链路继电器为字符串型 (仅使用每个字的低位字节)	LR0000.29E-LR9999.01E .I 是字符串长度, 范围介于 1 到 29 个字符之间	字符串	读 / 写
计时器/计数器	RC0000-RC9999 RCxxxx.00-RCxxxx.15	字、短整型、BCD 布尔型	读 / 写
计时器/计数器状态*	TC0000-TC9999	布尔型	读 / 写

● *注意: 某些模型不支持写入到计时器/计数器状态 (TCxxxx)。

字符串支持

开放模型支持以 ASCII 字符串的形式读取和写入辅助继电器 (AR)、数据内存 (DM)、保持继电器 (HR)、内部继电器 (IR) 和链路继电器 (LR)。当将其中任意一种设备类型用于字符串数据时, 每个寄存器可以包含 ASCII 数据的两个字节 (两个字符) 或一个。定义字符串时, 可以选择指定寄存器中 ASCII 数据的顺序或要在指定寄存器中使用的字节。

如果每个寄存器使用两字节 ASCII 数据, 则字符串的长度可以为 2 到 58 个字符, 并代替位数输入。长度必须输入为偶数。字符串跨越的寄存器范围不得超过设备类型的范围。通过将 "H" 或 "L" 附加到地址来指定字节顺序。

如果每个寄存器使用一字节 ASCII 数据, 则字符串的长度可以为 1 到 29 个字符, 并代替位数输入。字符串跨越的寄存器范围不得超过设备类型的范围。通过将 "D" 或 "E" 附加到地址来指定要在寄存器内使用的字节。

示例

- 对于从 DM100 开始、长度为 50 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: DM100.50H
- 对于从 DM110 开始、长度为 8 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: DM110.08L
- 对于从 DM200 开始、长度为 15 字节并采用仅高位字节的字符串, 要对其进行寻址, 请输入: DM200.15D
- 对于从 DM220 开始、长度为 7 字节并采用仅低位字节的字符串, 要对其进行寻址, 请输入: DM220.07E

数组支持

除了“布尔型”和“字符串”之外, 所有数据类型都支持数组。有关数组寻址的方法有两种。以下示例为根据数据内存位置。

DMxxxx [行数] [列数]

*DMxxxx [列数]**

*此方法假定“行数”等于 1。

行数、列数与以字节为单位的数据大小的乘积不能超过 116 个字节。此限制由协议规定。由于此驱动程序使用 ASCII 协议, 因此每个字、短整型和 BCD 对应 4 个字节, 而每个双字型、长整型、LBCD 和浮点型对应 8 个字节。例如, 对于数组大小为 28 字的 4 X 7 字数组, 如果每个字乘以 4 字节, 那么数组大小将为 112 字节, 即在 116 字节的最大请求大小范围内。

● 修改 32 位值 (双字型、长整型、LBCD 和浮点型) 时请小心。允许这些数据类型的每个地址在设备内的字偏移处开始。因此, 双字型 DM0 和 DM1 在字 DM1 处重叠。写入 DM0 还会修改保存在 DM1 中的值。建议用户使用这些数据类型, 以避免发生重叠。例如, 当使用双字型时, 用户可能希望使用 DM0、DM2、DM4 等字, 以防止“字”重叠。

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。请参阅有关筛选和排序“事件日志”详细信息视图的服务器帮助。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。

请求的类型与接收的类型不匹配。| 标记地址 = '<地址>', 请求的类型 = <类型>, 接收的类型 = <类型>。

错误类型：

错误

块中的地址错误。| 标记地址 = '<地址>', 块 = <开始> 至 <结束>。

错误类型：

警告

可能的原因：

已尝试参考指定设备中不存在的位置。

可能的解决方案：

验证分配给标记的地址是否在设备上的指定范围内。去除或更正参考无效位置的标记。

错误 掩码定义

B = 检测到 检测到硬件断点

F = 帧错误

E = I/O 错误

O = 字符缓冲区溢出

R = RX 缓冲区溢出

P = 已接收字节奇偶校验错误

T = TX 缓冲区已满

索引

B

BCD 13

C

C200H 寻址 17

C20H 14

COM ID 6

CQM1 寻址 20

I

I/O 错误 27

ID 9

IEEE-754 浮点 8

L

LBCD 13

R

RX 缓冲区溢出 27

T

TX 缓冲区已满 27

廖

报告通信错误 6-7

柺

标识 8

沂

波特率 4, 6

庭

不超过扫描速率请求数据 10

不扫描, 仅按需求轮询 10

焯

布尔型 13

摭

操作行为 6

败

超时前的尝试次数 11

串行端口设置 6

串行通信 5

钹

错误掩码定义 27

嗽

地址说明 14

讷

读取处理 7

瞍

短整型 13

雾

非规范浮点数处理 8

泊

浮点型 13

栖

概述 3

駉

高级信道属性 8

孽

故障时降级 11

儻

关闭空闲连接 6-7

关闭前空闲时间 6-7

闭

降级超时 11

降级期间 11

降级时放弃请求 11

庠

开放寻址 23

困

块中的地址错误。| 标记地址 = '<地址>', 块 = <开始> 至 <结束>。 27

担

框架 27

竭

来自缓存的初始更新 10

轻

连接超时 10

连接类型 5

泡

流量控制 4, 6

搞

模拟 9

壩

奇偶校验 4, 6, 27

講

请求超时 10

请求的类型与接收的类型不匹配。| 标记地址 = '<地址>', 请求的类型 = <类型>, 接收的类型 = <类型>。
27

请求间延迟 11

駢

驱动 5, 9

儼

冗余 12

戔

扫描模式 10

譚

设备属性 8

设备属性 - 自动降级 11

设置 4

冫

事件日志消息 27

攘

数据类型说明 13

数据收集 9

数据位 4,6

印

双字型 13

讠

调制解调器 7

脩

停止位 4,6

辺

通道分配 9

通道属性 - 常规 4
通道属性 - 写入优化 7
通信超时 10-11
通信协议 4

缩

网络 4
网络适配器 7

熾

物理媒体 6

償

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

侁

信道属性 4

坻

型号 9

庖

延迟 12

捉

掩码 27

丿

以扫描速率请求所有数据 10

以太网封装 4

灣

溢出 27

砌

硬件断点 27

躡

优化方法 7

勻

占空比 8

鋸

长整型 13

藹

诊断 5

膊

自动拨号 7

媯

字 13

字符串 13

字符间延迟 11

逕

遵循标签指定的扫描速率 10

遵循客户端指定的扫描速率 10