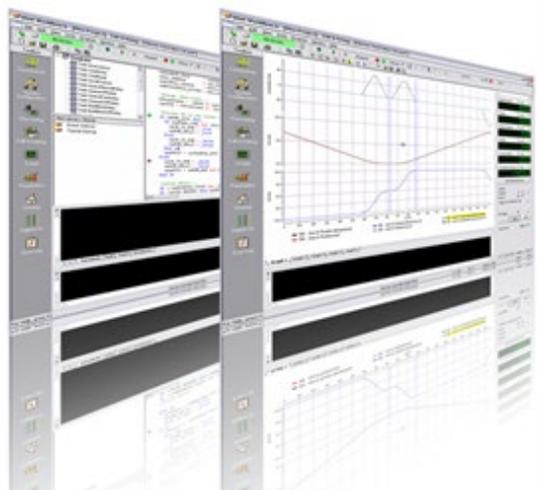


Modbus 是一种工业标准协议，允许各种自动化设备（如可编程逻辑控制器和人机界面）相互通信。Modbus 协议定义了一个独立于通信底层的简单协议数据单元 (PDU)。

控制器使用一种主从技术（通过 RS232/422/485）进行通信，其中只有一个设备（主设备）可以启动事件（称为“查询”）。其他设备（从设备）通过向主设备提供请求的数据或采取查询中请求的操作来响应。典型的主设备包括主机处理器和编程板。典型的从设备包括可编程控制器。



简介

主设备可以向单个从设备发送信息，也可以对所有从设备发起广播消息。从设备对单独发送给自己的查询返回一个消息（称为“响应”）。从设备不对主设备的广播查询返回响应。

Modbus 协议通过在其中放置设备（或广播）地址、定义要求的动作的功能代码、所有要发送的数据和一个错误校验字段来确定主设备查询的格式。从设备的响应消息也使用 Modbus 协议构建。包括确认已采取的行动的字段、任何要返回的数据和一个错误校验字段。如果消息接收过程中发生了错误，或从设备不能执行要求的行动，从设备将构建一个错误消息并将其作为回应发送出去。

用 Mint 语言编写了一个 Modbus 解释器。这不仅展示了 Mint 编程语言的强大和灵活性，还允许为您的应用程序量身定制 Modbus 功能。Modbus 有两种变体：RTU 和 ASCII。基于 Mint 的解释器仅在 Mint 控制器作为从设备运行的情况下提供对 ASCII 变体的支持。Mint 示例程序允许 Modbus 主设备读/写 Mint 通信阵列，从 Mint 控制器读取数字输入，并写入 Mint 控制器上的数字输出。

Modbus ASCII

数据格式

当控制器被设置为使用 ASCII（美国信息交换标准代码）模式在 Modbus 网络上通信时，消息中的每个 8 位字节都作为两个 ASCII 字符发送。这种模式的主要优点是允许字符之间出现长达一秒的时间间隔，而不会导致错误。

Mint 解释器支持具有以下通信设置的主设备：

- 7 个数据位、1 个停止位、偶数奇偶校验
- 7 个数据位、1 个停止位、奇数奇偶校验
- 8 个数据位、1 个停止位、无奇偶校验

波特率仅限于 Mint 控制器支持的波特率（当然，必须与 Modbus 主设备上的波特率设置相匹配）。

Mint 启动块定义了控制器的串行端口设置（示例程序说明了终端通道 1 上的 19200 波特设置 - 例如 Nextmove ESB 控制器上的串行端口）……

```
SERIALBAUD(_TERM1) = 19200
```

Mint 任务'ModbusSlave'声明一个常量来设置所需的奇偶校验……

```
Const _nParity As Integer = 0 '0=偶数, 1=奇数
```

此任务还声明一个常量来设置正在使用的数据位数……

```
Const _nBits As Integer = 7
```

如果该常数设置为 8，则 _nParity 的设置变得不相关（即可以设置为 0 或 1）。

消息成帧

在 ASCII 模式下，消息以“冒号”（:）字符（ASCII 3A 十六进制）开始，以“回车换行”(CRLF)对（ASCII 0D 和 0A 十六进制）结束。为所有其他字段传输的允许字符为十六进制 0-9, A-F。

网络设备连续监控网络总线的冒号字符。当接收到一个时，每个设备解码下一个字段（地址字段），以确定它是否是被寻址的设备。

消息中的字符之间最多可以间隔一秒。如果出现更大的间隔，则接收设备假定发生了错误。

以下所示为一个典型的消息帧。

开始	地址	功能	数据	LRC 检查	结束
1 CHAR	2 CHARS	2 CHARS	n CHARS	2 CHARS	2 CHARS
:	02	03	001D	DE	<CR><LF>

地址字段

ASCII 消息帧的地址字段包含两个字符。有效的从设备地址在十进制数 0-247 的范围内。

当从设备发送响应时，它将自己的地址放在响应的地址字段中，让主设备知道是哪个从设备在响应。

地址 0 用于广播地址，所有从设备都能识别该地址。因此，在配置 Mint 控制器用作 Modbus 从设备时，最好避免将该控制器设置为 Mint 节点 0。

功能字段

ASCII 消息帧的功能码字段包含两个字符。（主设备）有效的代码在十进制数 1-127 的范围内。

当从主设备发送消息到从设备时，功能码字段告知从设备要执行哪种操作。例如读取一组离散输入的开/关状态；读取一组寄存器的数据内容；读取从设备的诊断状态或写入指定寄存器。

当从设备响应主设备时，它使用功能码字段指示正常（无错误）响应或发生某种错误（称为异常响应）。对于正常响应，从设备只是回显原始功能码。

对于异常响应，从设备返回一个代码，该代码相当于原始功能码，其最高有效位设置为逻辑 1。

Mint Modbus ASCII 解释器支持完整 Modbus 功能集的子集：-

01（十进制）- 读取输出

该功能读取从设备上离散数字输出的开/关状态。不支持广播（即功能请求将被忽略）。

来自 Mint 控制器的查询消息指定要从 Mint 控制器读取的起始输入和输出数量。输出从零开始寻址（就像它们在 Mint 中一样）。

响应消息中的输出状态打包为每位数据字段一个输出。状态表示为：1 = 开；0 = 关。第一个数据字节的 LSB 包含查询中寻址的输出。其他输出朝向该字节的高位末端，并在后续字节中从“低位到高位”。

如果返回的输出量不是 8 的倍数，则最后一个数据字节中的剩余位将用零填充（朝向该字节的高位末端）。Byte Count 字段指定数据的完整字节数量。

02（十进制）- 读取输入

该功能读取从设备中离散数字输入的开/关状态。不支持广播（即功能请求将被忽略）。

来自 Mint 控制器的查询消息指定要从 Mint 控制器读取的起始输入和输入数量。输入从零开始寻址（就像它们在 Mint 中一样）。

响应消息中的输入状态打包为每位数据字段一个输入。状态表示为：1 = 开；0 = 关。第一个数据字节的 LSB 包含查询中寻址的输入。其他输入朝向该字节的高位末端，并在后续字节中从“低位到高位”。

如果返回的输入量不是 8 的倍数，则最后一个数据字节中的剩余位将用零填充（朝向该字节的高位末端）。Byte Count 字段指定数据的完整字节数量。

03（十进制）- 读取保持寄存器

该功能读取从设备中保持寄存器（本例中为 Mint Comms 位置）的二进制内容。不支持广播。

查询消息指定要读取的起始寄存器和寄存器数量。寄存器从零开始寻址。然而，Mint Comms 位置的寻址范围为 1-99（另请参见应用说明 An00110），因此应注意不要尝试从 Mint 控制器上的寄存器 0 读取。

数据被处理为 16 位值（范围 +/-32767）。

05（十进制）- 强制单个线圈

此功能强制单个线圈（Mint 控制器数字输出）开启或关闭。当主设备广播（即发送到节点 0）时，该功能强制在所有连接的从设备中提供相同的数字输出。

查询消息指定要强制的数字输出。输出从零开始寻址（就像它们在 Mint 中一样）。

请求的开/关状态由查询数据字段中的常量指定:

- FF00 十六进制值请求线圈开启
- 0000 值请求线圈关闭

所有其他值都是非法的, 不会影响线圈。

正常的响应是查询的回显, 在强制输出后返回。

16 (十进制) - 预置多寄存器

该功能将值预置 (写入) 到一系列保持寄存器中 (本例中为 Mint Comms 位置)。广播时, 该功能在所有附加的从设备中预置相同的寄存器引用。

查询消息指定要预置的寄存器引用。寄存器从零开始寻址。然而, Mint Comms 位置的寻址范围为 1-99 (另请参见应用说明 An00110), 因此应注意不要尝试写入 Mint 控制器上的寄存器 0。

数据被处理为 16 位值 (范围 +/-32767)。

23 (十进制) - 读/写多寄存器

此功能将值预置 (写入) 到一系列保持寄存器 (本例中为 Mint Comms 位置) 中, 并回读一系列保持寄存器 (本例中为 Mint Comms 位置), 所有操作都在一个事务中完成 (在 PLC 需要确认已接收/处理写入的数据时, 此功能非常有用)。不支持广播。

查询消息指定要预置的寄存器引用和要读取的寄存器引用。寄存器从零开始寻址。然而, Mint Comms 位置的寻址范围为 1-99 (另请参见应用说明 An00110), 因此应注意不要尝试读取或写入 Mint 控制器上的寄存器 0。

数据被处理为 16 位值 (范围 +/-32767)。

数据字段

数据字段是使用两个十六进制数字的集合构造的, 范围为十六进制数 00 到 FF。它们由一对 ASCII 字符组成。从主设备发送到从设备的消息的数据字段包含额外的信息, 从设备必须使用这些信息来执行功能码定义的操作。这可以包括像离散地址和寄存器地址这样的项、要处理的项的数量以及字段中的实际数据字节数。

例如, 如果主设备请求从设备读取一组保持寄存器 (功能码 03), 则数据字段指定起始寄存器和要读取的寄存器数量。如果主设备写入从设备中的一组寄存器 (功能码十进制数 16), 则数据字段指定起始寄存器、写入寄存器的数量、数据字段后面的数据字节数以及要写入寄存器的数据。

如果没有发生错误, 则从设备到主设备的响应的数据字段包含所请求的数据。如果发生错误, 则该字段包含一个异常代码, 主设备的应用程序可以使用该异常代码确定要采取的下一个操作。

错误检查字段

当字符成帧使用 ASCII 模式时，错误检查字段包含两个 ASCII 字符。错误检查字符是对消息内容执行纵向冗余检查 (LRC) 计算的结果，不包括开头的“冒号”和结尾的 <CR><LF> 字符。

LRC 字符作为 <CR><LF> 字符之前的最后一个字段追加到消息中。

异常响应

除了广播消息外，当主设备向从设备发送查询时，它希望得到正常的响应。主设备的查询中可能发生以下四种事件之一：

- 如果从设备接收到的查询没有通信错误，并且能够正常处理查询，则返回正常响应。
- 如果从设备由于通信错误而没有接收到查询，则不返回响应。主设备程序最终将处理查询的超时条件。
- 如果从设备接收到查询，但检测到通信错误（奇偶校验，LRC），则不返回响应。主设备程序最终将处理查询的超时条件。
- 如果从设备接收到的查询没有通信错误，但无法处理（例如，如果请求是读取一个不存在的寄存器），则从设备将返回异常响应，通知主设备错误的性质。

异常响应消息有两个字段将异常响应与正常响应区分开来：

功能码字段：在正常响应中，从设备在响应的功能码字段中回显原始查询的功能码。所有功能码的最高有效位 (MSB) 为 0（它们的值都低于十六进制数 80）。

在异常响应中，从设备将功能码的 MSB 设置为 1。这使得异常响应中的功能码值比正常响应的值高出十六进制数 80。

利用功能码的 MSB 设置，主设备的应用程序可以识别异常响应，并可以检查数据字段中的异常代码。

数据字段：在正常响应中，从设备可能返回数据字段中的数据或统计信息（查询中请求的任何信息）。在异常响应中，从设备在数据字段中返回异常代码。这定义了导致异常的从设备条件。

Mint 解释器程序只提供了三个异常响应……

```
Const _nlllegalFunction As Integer = 1
Const _nlllegalAddress As Integer = 2
Const _nlllegalValue As Integer = 3
```

如果主设备请求解释器所支持功能以外的功能，则会产生非法功能响应。

如果主设备试图访问 1-99 范围之外的 Comms 位置、试图读取组 0 的控制器输入范围之外的输入或试图写入组 0 的控制器输出范围之外的输出，则会产生非法地址响应。

如果主设备试图将非法值写入数字输出（即 0xFF00 或 0x0000 以外的值），则会产生非法值响应。

欲了解 Modbus ASCII 协议的更多详细信息，请访问 www.modbus.org

调试

变量 bDebug 被定义为允许一些简单的调试。如果将其设置为 _TRUE，则在执行解释器期间将通过终端通道 2（例如 NextMove ESB 上的 USB 连接）显示各种消息。对于仅支持单个终端通道的 Mint 控制器（例如，仅具有单个串行接口的 MintDrive li），应编辑示例程序以删除对终端通道 2 (#2) 的引用。

控制器规范

本应用说明附带的示例代码专门针对非以太网 Powerlink 控制器。当使用 NextMove e100 控制器（例如 BUSNODE(6) 替换 NODE）时，请参考代码中的注释以了解关键字的微小差异。

时序

以 19200 波特率工作且只有 ModbusSlave 任务在 Nextmove ESB 上运行时，功能 01、02、03、05、16 和 23（十进制数）的时序如下：

01 - 读取输出

1 个输出：10ms

16 个输出：10ms

02 - 读取输入

1 个输入：10ms

16 个输入：10ms

03 - 读取寄存器

1 个位置：10ms

99 个位置：420ms

05 - 写入输出

1 个输出：10ms

16 - 写入寄存器

1 个位置：10ms

99 个位置：240ms

23 - 读/写寄存器

1 个位置：14ms

99 个位置：500ms

如果可以在主设备上配置超时设置，则此信息可能很有用。

联系我们

欲了解详情，请联系
当地 ABB 代表或：

new.abb.com/motion
new.abb.com/drives
new.abb.com/drivespartners
new.abb.com/PLC

© 2012 年 ABB 版权所有。保留所有权利。
规格如有更改，恕不另行通知。