采用 Unity Pro 的 Quantum
140 ESI 062 10 ASCII 接口模块
用户手册

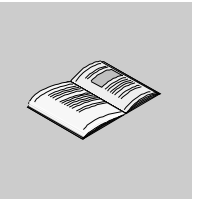
06/2006

35012189.00

a brand of Schneider Electric

Telemecanique

目录



关于本书 7
重要信息 5

章 1 140 ESI 062 10 硬件描述 9
概览 9
介绍 10
指示灯 11
外部连接器和开关 13
规格 15

章 2 Quantum 寻址模式 19
概述 19
平面寻址 — Modicon Quantum 800 系列 I/O 模块 20
拓扑寻址 — 带 Unity 的 Modicon Quantum 800 系列 I/O 模块 21
寻址示例 22
离散 I/O 位编号 23
寻址 24

章 3 配置概述 25
概览 25
ESI 配置 26
ASCII 消息格式 28
数据流 35
参数配置 38

章 4 ESI 命令行编辑器 41
概述 41
配置编辑器 42
ASCII 消息编辑器 46

章 5 ESI 命令 47
概览 47
ESI 命令概述 48
ESI 命令结构 49
命令处理 50
命令 0 — NO OPERATION 52

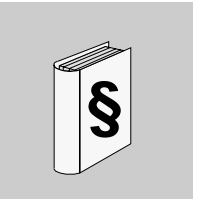
命令 1 — READ ASCII MESSAGE 53
命令 2 — WRITE ASCII MESSAGE 55
命令 3 — GET DATA（模块至控制器） 58
命令 4 — PUT DATA（控制器至模块） 60
命令 5 — GET TOD（时间） 62
命令 6 — SET TOD（时间） 64
命令 7 — SET MEMORY REGISTERS 67
命令 8 — FLUSH BUFFER 69
命令 9 — ABORT 71
命令 A — GET BUFFER STATUS 73
非法命令的响应结构 75
模块状态字（字 11） 76
读取超过有效寄存器范围 78

附录 79
概览 79

附录 A 字符集 81
ASCII 字符集 82

附录 B ESI 062 10 简介 85
概览 85
ESI 模块简介 86
应用标准 87
模块描述 88
ESI 模块结构图 89

安全信息



重要信息

注意

在尝试安装、操作或维护设备之前，请仔细阅读下述说明并通过查看来熟悉设备。下述特别信息可能会在本文其他地方或设备上出现，提示用户潜在的危险，或者提供阐明或简化某一过程的信息。

在“危险”或“警告”安全标签上添加此符号表示存在触电危险，如果不遵守使用说明，将导致人身伤害。

这是提醒注意安全的符号。提醒用户可能存在人身伤害的危险。请遵守所有带此符号的安全注意事项，以避免可能的人身伤害甚至死亡。

⚠ 危险

“危险”表示可能存在危险，如果不遵守说明，可**导致**严重的人身伤害甚至死亡，或设备损坏。

⚠ 警告

“警告”表示可能存在危险，如果不遵守说明，可**导致**严重的人身伤害甚至死亡，或设备损坏。

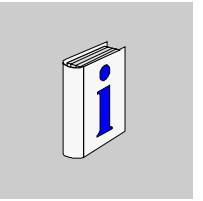
⚠ 注意

“注意”表示可能存在危险，如果不遵守说明，可**导致**严重的人身伤害或设备损坏。

请注意

电气设备的维修仅限于合格人员。对于使用本资料所引发的任何后果，Schneider Electric 概不负责。本文档不可用作未经培训人员的使用手册。

关于本书



浏览

文档范围

本文档解释 ASCII 接口模块的安装和使用。

本文档适用于 Unity Pro 1.1 及更高版本。

有效性

此文档中给出的数据和示意图并不是一成不变的。我们保留根据持续产品开发策略修改我们的产品的权利。本文档中的信息如有更改，恕不另行通知，并且不应理解为 Schneider Electric 承担的义务。

相关的文件

文件名称	参考编号
Quantum Hardware Reference Manual（Quantum 硬件参考手册）	UNYUSE10010
Quantum Discrete and Analog I/O Reference Manual（Quantum 离散量和模拟量 I/O 参考手册）	UNYUSE10010
Quantum Experts and Communication Reference Manual（Quantum 专家和通讯参考手册）	UNYUSE10010
Grounding and Electromagnetic Compatibility of PLC Systems User Manual（PLC 系统的接地和电磁兼容性用户手册）	UNYUSE10010
Quantum and Premium Communication Architecture Reference Manual（Quantum 和 Premium 通讯体系结构参考手册）	本软件包的一部分

您可以从我们的网站下载这些技术出版物以及其他技术信息，网址为：www.telemecanique.com。

与产品相关的警告

对于本文档中可能出现的任何错误，Schneider Electric 概不负责。如果您有关于改进或更正此出版物的任何建议，或者从中发现错误，请通知我们。

未经 Schneider Electric 明确书面许可，不得以任何形式、通过任何电子或机械手段（包括影印）复制本文档的任何部分。

在安装和使用本产品时，必须遵守国家、地区和当地的所有相关的安全法规。出于安全方面的考虑和为了确保符合归档的系统数据，只允许制造商对各个组件进行维修。

当控制器用于具有技术安全要求的应用时，请遵守有关的使用说明。

如果没有在我们的硬件产品上使用 Schneider Electric 软件或得到认可的软件，则可能导致人身伤害、损害或不正确的操作结果。

不遵守本产品的相关警告可能导致人身伤害或设备损坏。

用户意见

欢迎对本书提出意见。您可以给我们发邮件，我们的邮件地址是 techpub@schneider-electric.com

140 ESI 062 10 硬件描述

1

概览

简介

本章中的内容描述 140 ESI 062 10 模块的硬件特性。本章末尾将介绍产品规格。

本章包含了哪些内容？

本章包含了以下主题：

主题	文件集
介绍	10
指示灯	11
外部连接器和开关	13
规格	15

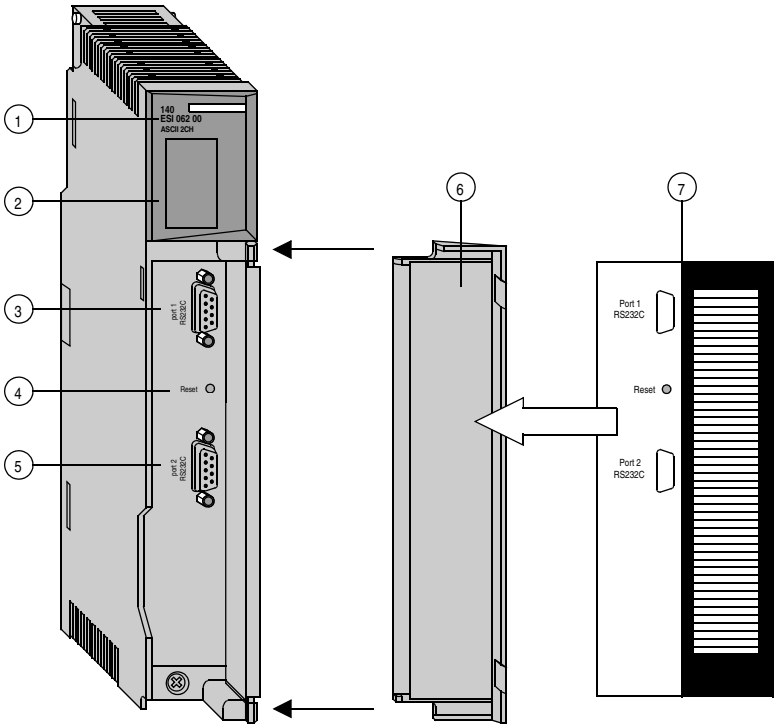
介绍

功能

140 ESI 062 10 模块是一个 Quantum 通讯接口模块，用于将来自 ASCII 设备的消息和/或数据输入到 CPU，将来自 CPU 的消息和/或数据输出到 ASCII 设备，或者在 ASCII 设备和 CPU 之间双向交换消息和/或数据。

示意图

下图显示 140 ESI 062 10 模块及其组件。

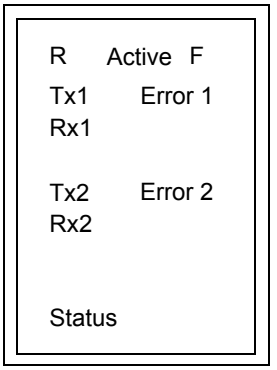


1 型号、模块描述、颜色代码
2 LED 显示
3 端口 1 连接器
4 复位按钮
5 端口 2 连接器
6 可拆除门
7 客户标识标签（折叠标签并将标签放置于门内）

指示灯

示意图

下表显示 140 ESI 062 10 模块的 LED 指示灯。



描述

下表描述了 140 ESI 062 10 模块的 LED 说明。

LED	颜色	灯亮时指示
R	绿色	模块已通过加电诊断。
活动	绿色	存在总线通讯。
F	红色	模块检测到故障。
RX1	绿色	RS-232 端口 1 上接收到数据
TX1	绿色	RS-232 端口 1 上已传输数据
RX2	绿色	RS-232 端口 2 上接收到数据
TX2	绿色	RS-232 端口 2 上已传输数据
状态	黄色	状态
错误 1	红色	端口 1 上有错误状况
错误 2	红色	端口 2 上有错误状况

LED 闪烁顺序

下表显示 F、状态、错误 1 和错误 2 LED 的闪烁顺序。

LED				描述
F	状态	错误 1	错误 2	
闪烁	闪烁	闪烁	闪烁	ASCII 模块正在初始化（第一次加电）
关	开	关	关	编程模式
关	关	开	无	串行端口 1 发生缓冲区溢出
关	关	无	开	串行端口 2 发生缓冲区溢出
无	闪烁（参见崩溃代码）	关	关	ASCII 模块处于内核模式下，可能有错误

状态 LED 崩溃代码

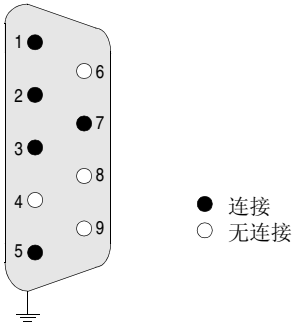
下表显示状态 LED 的崩溃代码。

闪烁次数	代码（十六进制）	错误
稳定	0000	请求内核模式
4	6631	错误的微控制器中断
5	6503	RAM 地址测试错误
8	6402	RAM 数据测试错误
7	6300	PROM 校验和错误（没有加载 EXEC）
	6301	PROM 校验和错误
	630A	闪存消息校验和错误
	630B	执行警戒时钟超时错误
8	8000	其他内核错误
	8001	内核 PROM 校验和错误
	8002	闪存程序错误
	8003	意外执行返回

外部连接器和开关

RS-232 串行端口

ESI 有两个串行端口，用于与串行设备通讯。下面是 ASCII 模块串行端口的引脚连接。



RS-232 端口引脚：

引脚	信号名	描述
1	DCD	载波检测
2	RXD	接收数据
3	TXD	传输数据
4	无	未连接
5	GND	信号接地
6	无	未连接
7	RTS	请求发送
8	无	未连接
9	无	未连接
屏蔽	无	机箱接地

编程端口

端口 1 也用作编程端口（端口 0）。按住复位按钮超过 4 秒可进入该模式。进入编程模式后，串行端口设置为标准终端通讯配置。

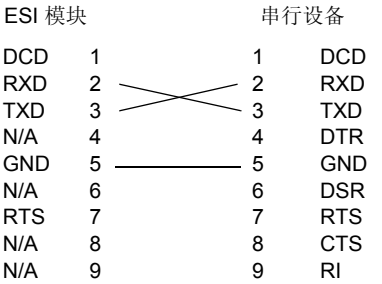
编程模式使用以下端口设置：

参数	设置
波特率	9600
数据位	8
停止位	1
校验位	无（禁用）
键盘模式	ON（字符回显）
XON/XOFF	ON

以这种方式设置串行端口配置，使得该端口配置是一个已知配置，并且可与模块运行时使用的配置相同或不同。

最小电缆布局

下面的示意图显示将 ESI 模块连接到外部设备或编程终端 (PC) 所需的最小电缆布局：



复位按钮

一个凹陷按钮位于模块前部。

复位按钮具有两个功能：

- 通过短按复位模块
- 通过按住按钮超过 4 秒进入编程模式

规格

数据接口

数据接口

RS-232	2 个串行端口（9 针 D 型），无隔离	
布线 （最大电缆长度 20 米，带屏蔽）	990 NAA 263 20，Modbus 可编程 RS 232 电缆，12 英尺（2.7 米）	
	990 NAA 263 50，Modbus 可编程 RS 232 电缆，50 英尺（15.5 米）	

固件

固件规格

端口性能	突发速度： 连续速度：	19.2 千波特/端口 取决于应用
嵌套消息深度	8	
缓冲区大小	255 输入 255 输出	
消息数	255	
最大消息长度	127 个字符加 1 位校验和	

存储器

存储器规格

RAM	256 kb（用于数据和程序）+ 2 kb 双端口 RAM
Flash-ROM	128 kb（用于程序和固件）

电源

电源规格

功耗	2 W（最大值）
总线电流要求	300 mA

熔断器

熔断器要求

内部	无
外部	用户自行决定

I/O) 映射

地址要求

输入	12 个字
输出	12 个字

兼容性

兼容性

编程软件	Concept 2.0 或更高版本、ProWorx NxT、ProWorx 32、Modsoft、Unity Pro
支持的数据格式	文本、十进制、定点、嵌套写入消息、设置寄存器指针、打印时间/日期、重复、空格、换行、控制代码、刷新缓冲区
Quantum 控制器	所有，至少为 Executive V2.0
电池备份模块	140 XCP 900 00

机械

机械

重量	1 kg（最大值）
尺寸 (H x D x W)	250 mm x 103.85 mm x 40.34 mm
材料	（机箱和前盖）Lexan
空间要求	1 个背板插槽

电气

电气

RFI 抗干扰 (IEC 1000-4-3)	27 ... 500 MHz，10 V/m
静电释放 (IEC 1000-4-2)	8 kV 空气 / 4 kV 触点
快速瞬态 (IEC 1000-4-4)	0.5 kV 共模
阻尼振荡瞬态	1 kV 共模 0.5 kV 差模
抗浪涌能力（瞬态）(IEC 1000-4-5)	1 kV 共模 0.5 kV 差模

环境条件

操作的环境条件

温度	0 ... 60°C (32 ... 140°F)
湿度	0 ... 95% RH 无冷凝 @ 60°C
化学反应	机箱和前盖由 Lexan 制成，这是一种不耐强碱溶剂的聚碳酸酯。
海拔高度	2,000 m
振动	10 ... 57 Hz @ 0.075 mm d.a. 57 ... 150 Hz @ 1 g
震动	+/-15 g 峰值，11 ms，半正弦波

储存条件

储存条件

温度	-40 ... 85°C (-40 ... 185°F)
湿度	0 ... 95% RH 无冷凝 @ 60°C
自由下落	1 m

机构批准

机构批准

UL 508 CSA 22.2-142 Factory Mutual 1 类，2 分类 关于 EMC 89/336/EEC 的欧洲指令

Quantum 寻址模式

2

概述

目的

在此专用模块的功能描述中，广泛使用在 Quantum 产品中建立的寄存器寻址模式 (3x, 4x)。为了使用户方便地转换到 Unity Pro 提供的寻址模式，本章说明了 Unity Pro 所允许的从 Quantum 模块寻址数据的不同模式：

- 平面寻址
- 拓扑寻址

本章包含了哪些内容？

本章包含了以下主题：

主题	文件集
平面寻址 — Modicon Quantum 800 系列 I/O 模块	20
拓扑寻址 — 带 Unity 的 Modicon Quantum 800 系列 I/O 模块	21
寻址示例	22
离散 I/O 位编号	23
寻址	24

平面寻址 — Modicon Quantum 800 系列 I/O 模块

简介

Modicon Quantum with Unity 模块遵循平面寻址映射系统。为正常工作，每个模块都需要确定数目的位和（或）字。IEC 寻址系统等效于 984LL 寄存器寻址。请按以下方式指定：

- 0x 现在为 %Mx
- 1x 现在为 %Ix
- 3x 现在为 %IWx
- 4x 现在为 %MWx

请依据下表将 984LL 表示形式转换为 IEC 表示形式。

输出和输入	984LL 表示形式寄存器地址	IEC 表示形式		
		系统位和系统字	存储器地址	I/O 地址
输出	0x	系统位	%Mx	%Qx
输入	1x	系统位	%Ix	%Ix
输入	3x	系统字	%IWx	%IWx
输出	4x	系统字	%MWx	%QWx

访问模块的 I/O 数据，

步骤	操作
1	在配置屏幕中输入地址范围。

示例

以下示例显示了 984LL 寄存器寻址与 IEC 寻址的关系：

000001 现在为 %M1

100101 现在为 %I101

301024 现在为 %IW1024

400010 现在为 %MW10

拓扑寻址 — 带 Unity 的 Modicon Quantum 800 系列 I/O 模块

简介

可使用拓扑寻址访问 I/O 数据项。请使用下面的表示形式标识模块在 Modicon Quantum with Unity 系统中的拓扑位置：

```
%<Exchangetype><Objecttype>[\b.e\]r.m.c[.rank]
```

所用的缩写：

- **b** = bus（总线）
- **e** = equipment (drop)（设备（子站））
- **r** = rack（机架）
- **m** = module slot（模块插槽）
- **c** = channel（通道）

注意：寻址时，
1. 在本地机架中 [\b.e\] 缺省为 \1.1\，不需要指定。
2. rank（序号）是一个索引，用于标识具有相同数据类型的对象的不同属性（值、警告级别、错误级别）。
3. 序号从零开始，如果序号为零，则忽略该项。

有关 I/O 变量的详细信息，请参考 *Unity Pro 参考手册*。

示例

读取值

目的	操作
从位于本地机架的插槽 6 中的模拟量模块的通道 7 中读取输入值（序号 = 0）：	输入 `%IW1.6.7[.0]`
从位于 RIO 总线 2 的子站 3 中的模拟量模块的通道 7 中读取输入值（序号 = 0）：	输入 `%IW\2.3\1.6.7[.0]`
从位于本地机架的插槽 6 中的模拟量模块的通道 7 中读取 "超范围" 值（序号 = 1）：	输入 `%I1.6.7.1[.0]`

寻址示例

3 种寻址模式的示例

以下示例将比较 3 种可能的寻址模式。一个 8 通道热电偶 140 ATI 030 00 模块，使用以下配置数据：

- 安装在 CPU 机架（本地机架）的插槽 5 中
- 起始输入地址为 201（输入字 %IW201）
- 结束输入地址为 210（输入字 %IW210）

要访问该模块中的 I/O 数据，可以使用以下语法：

模块数据	平面寻址	拓扑寻址	IODDT 寻址	Concept 寻址
通道 3 温度	%IW203	%IW1.5.3	My_Temp.VALUE	300203
通道 3 溢出	%IW209.5	%I1.5.3.1	My_Temp.ERROR	300209 位 5 由用户逻辑抽取
通道 3 范围警告	%IW209.13	%I1.5.3.2	My_Temp.WARNING	300209 位 13 由用户逻辑抽取
模块内部 温度	%IW210	%IW1.5.10	不可通过 IODDT 访问	300210

注意：对于 IODDT，使用数据类型 `T_ANA_IN_VWE`，并定义了地址为 `%CH1.5.10` 的变量 `My_Temp`。

出于对比的目的，Concept 使用的寄存器寻址添加在最后一列。因为 Concept 不支持直接对字中的位进行寻址，所以位抽取必须在用户程序中执行。

离散 I/O 位编号

简介

I/O 模块的通道编号通常从 1 开始递增计数，直到达到所支持通道的最大数目为止。但是，该软件从 0 开始编号，对应于字中的最低有效位 (LSB)。此外，Quantum I/O 模块将自己的最低通道映射到最高有效位 (MSB)。

下图显示 I/O 通道与字中的各位的映射关系：

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	I/O 通道
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	位编号

MSB LSB

字寻址与位寻址

大体上，离散 I/O 模块可以配置为使用字格式或位格式发送其 I/O 数据。这可以在配置时选择 %IW (%MW) 或 %I (%M) 来实现。如果需要从配置为使用 I/O 字的 I/O 模块访问单个位，可以使用语法 %word.bit。下表提供了 I/O 点编号和以位以及字寻址表示的所属 I/O 地址之间的联系。

该表显示主机架中的一个 32 点输入模块，插槽 4 配置为起始地址为 %I1 或 %IW1：

I/O 通道	位地址 （平面寻址）	位地址 （拓扑寻址）	位地址 从字中抽取 （平面寻址）	位地址 从字中抽取 （拓扑寻址）
1	%I1	%I1.4.1[.0]	%IW1.15	%IW1.4.1.1.15
2	%I2	%I1.4.2[.0]	%IW1.14	%IW1.4.1.1.14
3	%I3	%I1.4.3[.0]	%IW1.13	%IW1.4.1.1.13
• • •				
15	%I15	%I1.4.15[.0]	%IW1.1	%IW1.4.1.1.1
16	%I16	%I1.4.16[.0]	%IW1.0	%IW1.4.1.1.0
17	%I17	%I1.4.17[.0]	%IW2.15	%IW1.4.1.2.15
18	%I18	%I1.4.18[.0]	%IW2.14	%IW1.4.1.2.14
• • •				
31	%I31	%I1.4.31[.0]	%IW2.1	%IW1.4.1.2.1
32	%I32	%I1.4.32[.0]	%IW2.0	%IW1.4.1.2.0

寻址

平面寻址

该模块需要 12 个连续的 16 位输入字 (%IW) 和 12 个连续的 16 位输出字 (%QW)。

拓扑寻址

140ESI06210 模块的拓扑地址：

点	I/O 对象	注释
输入 1	%IW[\b.e\]r.m.1.1	响应字
	• • •	
输入 12	%IW[\b.e\]r.m.1.12	数据
输出 1	%QW[\b.e\]r.m.1.1	命令字
	• • •	
输出 12	%QW[\b.e\]r.m.1.12	数据

所用的缩写：**b** = 总线，**e** = 设备（子站），**r** = 机架，**m** = 模块插槽。

注

输入/输出字 2 ... 12 用于模块与 CPU 间的数据交换，具体情况取决于活动命令。

配置概述

3

概览

简介

本章描述 ESI 模块的配置模式的基本信息。本章末尾描述外部设备和 PLC 之间的数据流。

本章包含了哪些内容？

本章包含了以下主题：

主题	文件集
ESI 配置	26
ASCII 消息格式	28
数据流	35
参数配置	38

ESI 配置

概述

ESI 模块包含一个内置命令行编辑器，用于配置端口通讯设置、内部时钟和 ASCII 消息。

编程端口

ESI 模块支持两个 RS 232 硬件端口，这两个端口在运行时拥有各自的参数设置。第一个端口也用作编程端口。在此模式下它拥有自己的一组参数。

如何进入配置模式

要进入 ESI 的配置模式，需要执行以下步骤：

步骤	操作
1	将哑终端或 PC 终端仿真（例如超级终端）连接到端口 1。有关合适电缆的信息，请参见 RS -232 串行端口。
2	将该终端的通讯参数设置为 9600 波特、8 个数据位、无校验、1 个停止位和 XON/XOFF 流控制。
3	按住 ESI 模块前面的复位按钮 4 秒以上。

命令行编辑器

当您按照以上步骤进入配置模式后，ESI 前面板上的黄色状态 LED 将亮起，终端屏幕上将显示下面的消息：

```
欢迎使用
MODICON QUANTUM ASCII 模块
正在进入编程模式 ...
当前日期为：Wed 01-01-2002
当前时间为：09:15:10a
CLI> _
```

可用命令

CLI 的命令结构：

命令		描述	示例
CLI		将编程模式设置为命令行解释程序。	无
HELP		显示可用命令以及命令的简要描述，或者显示所请求的命令的帮助（例如 CLI> HELP ASCII 显示关于 ASCII 命令的帮助）。	无
RUN		复位模块并进入正常运行模式。	无
CONFIG		将编程模式设置为配置解释程序。	无
	DATE	显示或设置模块中的当前日期。	有关示例，请参见 "配置编辑器" 一章
	TIME	显示或设置模块中的当前时间。	
	PORT	显示或设置端口参数设置。	
ASCII		将编程模式设置为 ASCII 消息解释程序。	无
	NEW	进入消息编辑器并将新消息保存在工作缓冲区中。	`ASCII>new`
	EDIT	显示指定消息，进入消息编辑器，并在完成后保存该消息。	`ASCII>edit (message #)`
	VIEW	显示现有消息以供查看。	`ASCII>view (message #)`
	SAVE	将对指定消息所做的更改保存在其工作缓冲区中。	`ASCII>save (message #)`
	CLR	清除指定消息。	`ASCII>clr (message #)`
	COPY	将指定消息复制到其他消息。	`ASCII>copy (message #) (message #)`
	SIM	仿真指定消息。显示使用的寄存器数量（有助于创建用户逻辑时进行映射）以及嵌套消息的最大深度（用于额外调试工具）。如果最大深度大于 8，则发送通知并显示嵌套的消息路径。	`ASCII>sim (message #)`
	DIR	显示所有可用消息的目录。可使用 CNTL S 和 CNTL Q 来停止和继续向终端显示数据。	无
	DLOAD	将消息从 PC 下载到模块。有关更多详细信息，请参见 "ASCII 消息传输"。	无
	ULOAD	上载所有编程后的消息 (1 ... 255)。	`ASCII>uload`
		将指定的编程后的消息从模块上载到 PC。有关更多详细信息，请参见 "ASCII 消息传输"。	`ASCII>uload (message # - message #)`

ASCII 消息格式

概述

ASCII 消息用于将信息从 ESI 模块发送到 ASCII 设备，例如终端程序。

ASCII 消息格式定义 CPU 中包含的数据如何转换为连续字符流，以及如何反向转换。

下表列出可用的消息格式：

格式	方向	描述
文本	输出	静态文本
ASCII	输出/输入	ASCII 字符
十六进制	输出/输入	十六进制数字
八进制	输出/输入	八进制数字
二进制	输出/输入	二进制数字
整数	输出/输入	整数数字
定点十进制	输出/输入	定点十进制数字
时间/日期	输出	时间/日期信息
控制字符	输出	空格和换行字符
控制序列	输出	3 位八进制控制字符
嵌套	输出/输入	消息嵌套

文本

包含在单引号内的任意 ASCII 字符串（例如，'message string'）是仅输出格式。任何包含此格式的消息将输出文本，无论消息从读取还是写入消息命令开始。

'. . .（文本）. . .'

ASCII 字符

带有寄存器数量和字段长度的 ASCII 格式变量字段：

n**A**m

n 为寄存器数量 1..99（格式重复）

m 为字段长度 1..2（字符数）

示例：2A2 作为输入表示 2 个寄存器，每个包含 2 个 ASCII 字符。

十六进制

带有寄存器数量和字段长度的十六进制格式变量字段：

n**H**m

n 为寄存器数量 1..99（格式重复）

m 为字段长度 1..4（数字个数）

示例：2H3 作为输入表示 2 个寄存器，每个包含 3 个十六进制数字。

八进制

带有寄存器数量和字段长度的八进制格式变量字段：

n**O**m

n 为寄存器数量 1..99（格式重复）

m 为字段长度 1..6（数字个数）

示例：3O4 作为输入表示 3 个寄存器，每个包含 4 个八进制数字。

二进制

带有寄存器数量和字段长度的二进制格式变量字段：

n**B**m

n 为寄存器数量 1..99（格式重复）

m 为字段长度 1..16（数字个数）

示例：1B8 作为输入表示 1 个包含 8 个二进制数字的寄存器。

整数前导空格

带有寄存器数量和字段长度的整数/十进制格式的变量字段，作为输出时使用前导空格。作为输入时，此格式接受前导零和空格作为 0（零）。

n**I**m

n 为寄存器数量 1..99（格式重复）

m 为字段长度 1..5（数字个数）

示例：2I5 作为输入表示 2 个寄存器，每个包含 5 个整数/十进制数字。最大值为 65,535。

整数前导零

带有寄存器数量和字段长度的整数/十进制格式的变量字段，作为输出时使用前导零。作为输入时，此格式接受前导零和空格作为 0（零）。

n**L**m

n 为寄存器数量 1..99（格式重复）

m 为字段长度 1..5（数字个数）

示例：3L5 作为输入表示 3 个寄存器，每个包含 5 个整数/十进制数字。最大值为 65,535。

定点十进制

带有寄存器数量和字段长度的定点十进制格式的变量字段，作为输出时使用前导空格。作为输入时，此格式接受前导零和空格作为 0（零）。

n**P**m.q

n 为寄存器数量 1..99（格式重复）

m 为数字个数 + '.'3..8

q 为小数部分的数字个数 1..5

示例：1P7.2 作为输入表示 1 个寄存器，其中包含 4 个十进制数字，后接`.'，其后是 2 个十进制数字（小数部分）。

注意：不应把此格式与浮点格式混淆。小数点的放置是用于输入/输出格式，对 PLC 寄存器中的值没有任何影响（例如 23.456、234.56 和 23456 三个值都指寄存器值 23456）。

嵌套消息

嵌套消息格式允许消息调用其他消息。此格式可在重复格式中使用；重复格式可在嵌套消息中使用以允许间接嵌套重复。允许的最大嵌套消息层为 8 层。不允许递归嵌套。

Mn

n 为消息编号 1..255

示例：M6 将运行编号为 6 的那条消息。

时间

有两种不同的时间显示格式。一种用于 12 小时时间，另一种用于 24 小时时间。这是仅输出格式。

T12 > hh:mm:ss AM/PM（12 小时时间）

T24 > hh:mm:ss（24 小时时间）

日期

有 5 种不同的日期显示格式，每种提供 2 种显示年份的格式类型。这是仅输出格式。

Dnm

n 为日期和月份类型 1..5

m 为年份类型 2 或 4

D12 > dd/mm/yy

D14 > dd/mm/yyyy

D22 > mm/dd/yy

D24 > mm/dd/yyyy

D32 > dd mmm yy

D34 > dd mmm yyyy

D42 > mmm dd, yy

D44 > mmm dd, yyyy

D52 > dd.mm.yy

D54 > dd.mm.yyyy

dd = 日期 (1..31)

mm = 月份 (1..12)

mmm = 月份（一月、二月 ..、十二月）

yy = 年份 (0..99)（90 - 99 表示 1990 - 1999，0 - 89 表示 2000 - 2089）

yyyy = 年份 (1990..2089)

重复

多个格式的重复；嵌套重复括号无效。

n(...)

n 是要重复 ( ) 中的内容的次数 1..99

示例：6('Item',1I2,4X,1I5,/) 将生成 6 行，每行均包含字段 'Item'、1I2、4X、1I5 和一个 <CR, LF>。

空格

ASCII 消息中表示空格的符号是 X。这是仅输出格式。

nX

n 为空格的数量 1..99

换行

ASCII 消息中表示 '回车符' 的符号是 **/**。这是仅输出格式。

控制代码

一个包含在双引号分隔符中的 3 位八进制控制字符（范围为 000 到 377）。这是仅输出格式。

"###"

是字符的八进制格式

示例："033"。

刷新

刷新当前正在运行的串行端口的输入缓冲区有以下四种方法：整个缓冲区、一定数量的字符、最多一对字符或者可重复最多一对字符。

<0>	刷新整个缓冲区
<1;bbb**>**	刷新直至删除一定数量的字符
<2;hhhh**>**	刷新直至字符对匹配
<3;rrr;hhhh**>**	刷新直至字符对重复匹配

bbb = 字符数量 (1..255)

hhhh = 十六进制的字符对 (0000..FFFF)

rrr = 重复次数 (1..255)

注意：端口缓冲区大小为 255 个字符。

ASCII 消息语法规则

在输入使用模块的 ASCII 消息编辑器创建或使用 ASCII 消息传输下载的消息后，将检查消息是否违反了常规语法以及格式语法。如果发现任何不符合的情况，将不保存消息（ASCII 消息传输）或者将通知用户并指出不符合的内容（ASCII 消息编辑器）。

- 必须使用格式分隔符 (,) 来分隔每个格式。
- 所有文本格式必须封闭。
- 格式 A、H、O、B、I、L、P、X 和 (可以有寄存器重复/数字值，为 1 至 99。
- 格式 A、H、O、B、I 和 L 的总字段大小为 1 至 8。
- 格式 P 总字段大小为 3 至 8，小数部分字段大小 1 至 5，但总字段大小必须至少比小数部分字段大小大 2。
- 格式 M（嵌套消息）只要不递归，可以使用 1 至 255（十进制）的任意消息编号。
- 格式 T 可以使用以下 2 种格式之一：T12 或 T24。
- 格式 D 可以使用以下 10 种格式之一：D12、D14、D22、D24、D32、D34、D42、D44、D52 和 D54。
- 控制代码格式 "###" 仅接受 000 至 377 间的 3 位八进制值。
- 刷新格式可以使用以下 4 种格式之一：<0>、<1;bbb>、<2;hhhh> 或 <3;rrr;hhhh>，其中 bbb = 1 至 255，hhhh = 0000 至 FFFF，rrr = 1 至 255。

标准 ASCII 消息预处理规则

在输入使用模块的 ASCII 消息编辑器创建或使用 ASCII 消息传输下载的消息后，将对消息进行预处理，以节约空间并标准化消息以便于在仿真或运行模式时进行解释。

- 文本不进行处理。
  示例：>'这是文本...' > >'这是文本...'
- 删除第一个格式前的空格。
  示例：> 1A4,2X > >1A4,2X
- 删除最后一个格式后的空格。
  示例：>1A4,2X（结束） > >1A4,2X（结束）
- 删除格式和分隔符前后的空格。
  示例：>1A4 , 2X > >1A4,2X
- 删除最后一个格式后的逗号。
  示例：>1A4,2X,,, > >1A4,2X
- 删除重复格式中最后一个格式后的逗号。
  示例：>1A4,2X,3(1I2,1X,,),/ > >1A4,2X,3(1I2,1X),/
- 非文本字符要大写。
  示例：>'text ',1a4,2x,/ > >'text ',1A4,2X,/
- 删除数字中的所有前导 0，刷新格式的重复/数字值和字符对值中的 0 除外。
  示例：>01A004,0002X > >1A4,2X

数据流

概述

在 Quantum 处理器和 ESI 模块的串行端口间交换数据包括以下步骤：

传输方向：

- 数据从 PLC 寄存器通过 I/O 配置中分配给 ESI 模块的 12 个输出寄存器传输到 ESI 寄存器区域。
- 根据 ASCII 消息解释 ESI 寄存器中的数据并将其传输到端口传输缓冲区。

接收方向：

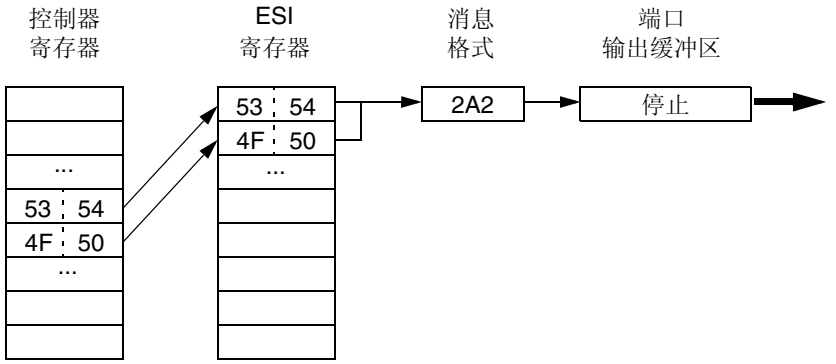
- 根据 ASCII 消息解释端口接收缓冲区中的数据并将其传输到 ESI 寄存器区域。
- 数据从 ESI 寄存器区域通过 I/O 配置中分配给 ESI 模块的 12 个输入寄存器传输到 PLC 寄存器。

ASCII 消息

ASCII 消息表示如何设置 ESI 寄存器中的数据格式以通过 RS-232 端口以任一方向传输的中央机制。例如，一个 16 位寄存器可以表示两个 ASCII 字符并作为两个字符传输，它也可以表示作为带有前导空格的整数（产生一个 5 个字符的字符串）传输的单个数字。有关可用格式的详细描述，请参见 *ASCII 消息格式*。

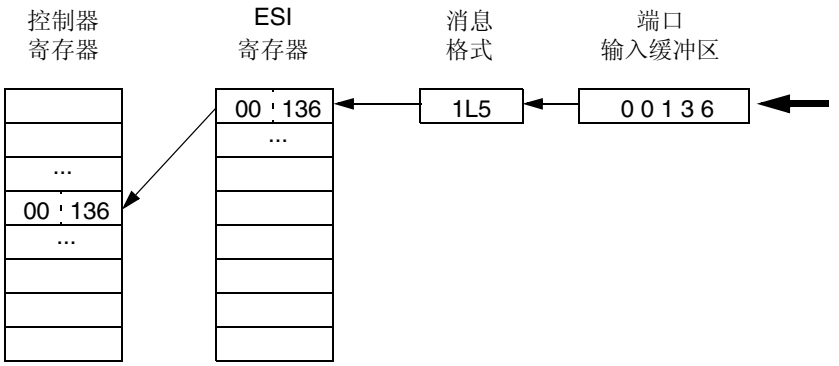
传输示例

下图是使用 "2A2" 消息格式（2 个寄存器，每个包含 2 个字符）从 Quantum 控制器传输 4 个字符的示例。端口缓冲区内容为 ASCII 格式，寄存器内容为十六进制：



接收示例

下图是使用 "1L5" 消息格式（1 个寄存器，带有前导零的 5 位数字）从 RS-232 端口接收 1 个数值的示例。端口缓冲区内容为 ASCII 格式，寄存器内容为十六进制：

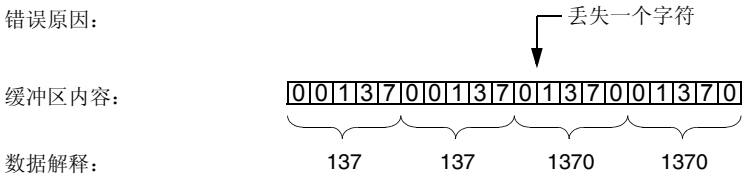


注意：请确保传入的字符数与 ASCII 消息中定义的数字一致。如果上面的示例中设备仅发送 "0013"，ESI 模块将无法完成接收命令，并将等待直到收到第 5 个字符。

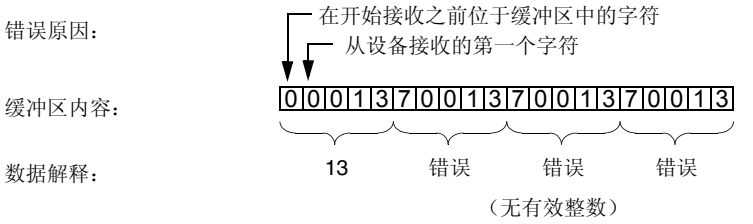
可能出现的同步问题

由于 ESI 模块仅支持没有开始或终止字符的固定长度消息格式，丢失任何字符（或增加预期之外的字符）都可导致对接收到的数据生成错误解释。以下示例显示 3 种不同错误类型的结果。假定消息格式为 "1L5"（最大值为 65,535）：

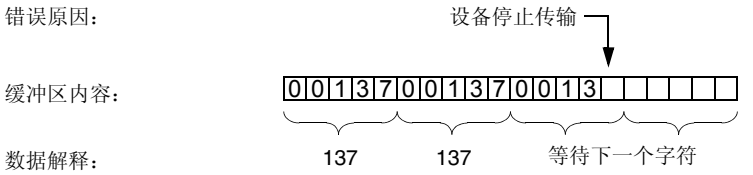
丢失字符的结果：



开始接收时缓冲区不为空的结果：



终止接收的结果：



FLUSH、ABORT、GET STATUS

为避免错误解释数据或锁定模块，应使用与缓冲区相关的命令 FLUSH BUFFER、ABORT、GET BUFFER STATUS 来控制数据交换。

有关这些命令的详细信息，请参见 *ESI 命令列表*。

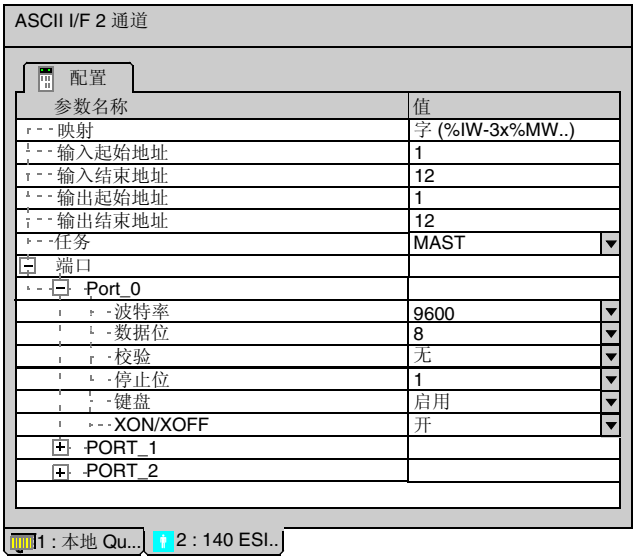
参数配置

概述

参数编辑器是 ESI 062 10 模块的 Unity Pro 配置的一部分。用户能够设置有关输入/输出寄存器和端口参数的多项信息。下图显示模块的不同设置。

参数和缺省值

"参数配置" 窗口



名称	缺省值	选项	描述
映射	字 (%IW-3X%MW-4X)	-	
输入起始地址	1	-	
输入结束地址	4	-	
输出起始地址	1	-	
输出结束地址	2	-	
任务 （如果模块不在本地则灰显）	MAST	FAST AUX0 AUX1 AUX2 AUX3	如果模块不在本地则始终为 MAST
端口			
PORT_0、PORT_1、PORT_2			
波特率	9600	300-19200	
数据位	8	7	
校验	无 (PORT_0) 偶 (PORT_1,2)	奇	
停止位	1	2	
键盘	启用 (PORT_0) 禁用 (PORT_1,2)	禁用	
XON/XOFF	开	关	

ESI 命令行编辑器

4

概述

概览

ESI 固件包含可通过端口 1 连接的哑终端来访问的编辑环境。本章描述如何使用此编辑器配置模块和编辑 ASCII 消息格式。

本章包含了哪些内容？

本章包含了以下主题：

主题	文件集
配置编辑器	42
ASCII 消息编辑器	46

配置编辑器

概述

配置编辑器接口是编程模式的一部分。它用于配置模块的串行端口和时间时钟。

注意：也可以通过 I/O 映射配置串行端口。I/O 映射会覆盖在配置编辑器中输入的任何串行端口配置。

注意：时间时钟还可以使用 SET TOD 命令配置。

要进入配置编辑器，请在 CLI> 提示符处键入 CONFIG。配置编辑器显示提示符 CONFIG>。

端口命令

端口命令显示或设置端口参数。可接受的命令格式变体包括：

```
PORT [n[: [b] [,p] [,d] [,s] [,k] [,x]]]
PORT [n[: [BAUD=b] [,PARITY=p] [,DATA=d] [,STOP=s] [,KEYBOARD=k] [,XON/XOFF=x]]
```

PORT 命令中使用的元素的描述和范围：

索引	描述	范围
n	端口号	0, 1, 2
b	波特率	50, 75, 110, 134.5, 150, 300, 600, 1200, 1800, 2000, 2400, 3600, 4800, 7200, 9600, 19200
p	校验设置	N, O, E
d	数据位的数目	5, 6, 7, 8
s	停止位的数目	1, 2
k	键盘模式（字符回显模式）	on, off
x	XON/XOFF 模式（软件流控制）	on, off

示例：

```
PORT 0:1200,n,8,1,on,on
PORT 0:baud=1200, parity=n, data=8, stop=1, keyboard=on, XON/XOFF=on
PORT 0
```

当前端口参数为：`PORT 0:BAUD=1200, PARITY=NONE ...`

输入新参数：`4800,n,8,1,off,on`

更改完模块的端口设置后，将显示下面的消息：

注：在此编程会话期间端口设置是临时的。

注意：端口 0 和 1 并不支持所有的波特率和数据位选项。请参考 "模块配置" 屏幕了解可用选项。

日期命令

显示或设置模块中的当前日期。可接受的命令格式变体包括：

```
DATE      [mm dd [  yy]]
DATE      [mm/dd [/ yy]]
DATE      [mm.dd [.yy]]
DATE      [mm dd [ YYYY]]
DATE      [mm/dd [/YYYY]]
DATE      [mm.dd [.YYYY]]
```

DATE 命令中使用的元素的描述和范围：

索引	描述	范围
mm	月份	1 ... 12
dd	日期	1 ... 31
yy	年份	00 ... 99
yyyy	年份	1990 ... 2089

示例：

```
DATE 3 30 95
DATE 3/3 0/1995
DATE
当前日期为 Wed 3 29 1995
输入新日期：3.30
```

注意：如果不需要更改年份，则只需要输入月份和日期。星期几由固件自动计算得出。yy 年份按此方法映射，即 00..89 = 2000..2089 和 90..99 = 1990..1999。

时间命令

显示或设置模块中的当前时间。可接受的命令格式变体包括：

```
TIME [hh:mm[:ss][x]]
TIME [hh.mm[.ss][x]]
```

TIME 命令中使用的元素的描述和范围：

索引	描述	范围
hh	时	1 ... 23
mm	分	1 ... 59
ss	秒	1 ... 59
x	上下午	a, p

示例：

```
TIME 3:26p
TIME 3.26.30p
TIME 15.26
TIME
当前时间为 3:15:26p
输入新时间：3.26.30p
```

注意：时间可以 12 或 24 小时时间格式输入。除非小时为 0 或 13 至 23，否则不输入上下午即表示 AM。

ASCII 消息编辑器

概述

ASCII 消息编辑器接口用于对模块中的 ASCII 消息格式进行编程。此接口包含一个简单的命令行解释程序（也类似于 Modicon B885 002 模块中的 CLI），该解释程序包含用于显示、创建、编辑、传输、保存、清除和测试 ASCII 消息的命令。命令集中还包含帮助命令，它给出可用命令的在线列表以及每个命令的含义。

要进入 ASCII 消息编辑器，请在 CLI> 提示符处键入 ASCII。ASCII 消息编辑器使用提示符 ASCII>。

ESI 命令

5

概览

简介

本章描述由 CPU 发送用以控制 ESI 模块的通讯功能的命令，以及包含数据和状态信息的 ESI 模块的响应。

本章包含了哪些内容？

本章包含了以下主题：

主题	文件集
ESI 命令概述	48
ESI 命令结构	49
命令处理	50
命令 0 — NO OPERATION	52
命令 1 — READ ASCII MESSAGE	53
命令 2 — WRITE ASCII MESSAGE	55
命令 3 — GET DATA（模块至控制器）	58
命令 4 — PUT DATA（控制器至模块）	60
命令 5 — GET TOD（时间）	62
命令 6 — SET TOD（时间）	64
命令 7 — SET MEMORY REGISTERS	67
命令 8 — FLUSH BUFFER	69
命令 9 — ABORT	71
命令 A — GET BUFFER STATUS	73
非法命令的响应结构	75
模块状态字（字 11）	76
读取超过有效寄存器范围	78

ESI 命令概述

ESI 命令列表

共有 11 个 ASCII 模块命令，用于管理 ESI 模块串行通讯和其他日常管理实用程序。Quantum 控制器将这些命令发送到 ESI 模块。ASCII 设备与 Quantum 控制器之间的数据交换集成在本节描述的 READ/WRITE 命令结构中。输出数据（第一个 4x 寄存器）包含命令；第一个输入寄存器 (3x) 包含响应以及命令回显。

下表总结了 ESI 模块命令：

命令	名称	描述
0	NO OPERATION	不进行操作
1	READ ASCII MESSAGE	开始读取 ASCII 消息
2	WRITE ASCII MESSAGE	开始写入 ASCII 消息
3	GET DATA	将数据从模块传输到 PLC
4	PUT DATA	将数据从 PLC 传输到模块
5	GET TOD	从模块获取时间
6	SET TOD	在模块中设置时间
7	SET MEMORY REGISTERS	设置寄存器值
8	FLUSH BUFFER	刷新串行端口缓冲区
9	ABORT	中止当前正在运行的 ASCII 消息
A	GET BUFFER STATUS	获取端口输入缓冲区

ESI 命令结构

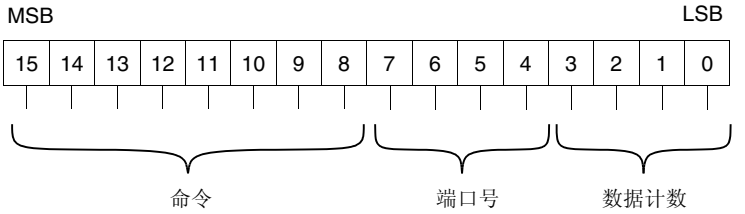
命令字格式

命令字是映射到模块的第一个输出寄存器。

ESI 模块的命令字格式如下：

- 位 0 ... 3 — 包含数据计数（以字为单位），范围是 0 ... 9
- 位 4 ... 7 — 包含端口号，范围是 1 ... 2
- 位 8 ... 15 — 包含命令，范围是 0 ... A

命令字结构：



注意：位顺序基于 IEC 标准，其中位 15 是最高有效位。

命令处理

寄存器

寄存器 3:x（PLC 输入寄存器）和 4:x（PLC 输出寄存器）用于处理控制 ESI 模块的命令。x 表示 PLC 硬件配置中 ESI 模块的起始地址。

ESI 模块处理的命令数据存放在输出寄存器 (4:x) 中，可能的响应信息存放在输入寄存器 (3:x) 中。

下面的示例显示命令 5（上载 ESI 系统时间）和命令 6（设置 ESI 系统时间）的寄存器占用情况。

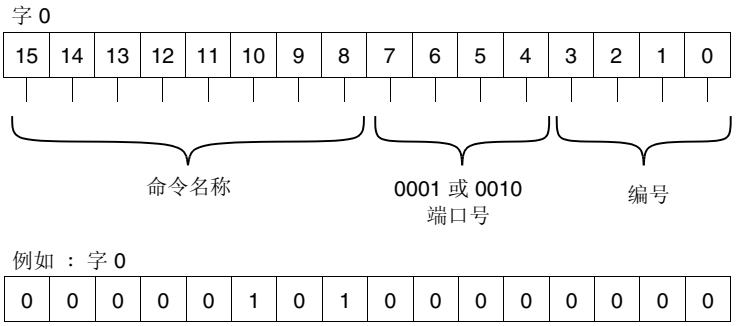
示例 5 GET TOD

命令 5 用于上载系统时间。为使此命令正确执行，必须在 ESI 模块输出寄存器的字 0 中写入命令参数。字 0 是模块硬件配置（PLC 配置）中的第一个输出寄存器。

注意：寻址 PLC 配置中从起始地址 4:1 到结束地址 4:12 的硬件时，命令字 0 对应于地址 4:1。

命令结构

命令字 0 分为以下几个区域：



命令字描述：

区域（位）	描述	示例值
0 - 3	要上载或输出的寄存器的编号。输出寄存器 (3:x) 的编号由命令 5 定义。这里将该值设置为 0。	0
4 - 7	端口号。执行命令 5 或 6 时不使用这些端口。数据只在模块内部使用寄存器处理。	0
8 - 15	位格式的命令名称。设置命令值后会直接处理该命令。	5

注意：可借助移动块或通过外部开关设置命令 0。也可以有其他变化。

结果

作为操作结果，ESI 系统时间数据被存放在寄存器 1 到 7 中。（见*页 63 响应结构*）

数据返回通过 PLC 3:x 寄存器执行。它对应于模块硬件配置（PLC 配置）中的输入寄存器。

注意：寄存器 0（状态寄存器）显示命令处理的状态。命令已正确执行时该寄存器对应于命令字 0。如果出现错误数据，MSB（最高有效位）的状态从 0 更改为 1。

示例 6 SET TOD

命令 6 用于设置系统时间。与命令 5 一样，需要的命令参数必须写入 ESI 模块输出寄存器 (4:x) 的字 0 中。设置系统时间时，时间和日期参数单独传送。该参数存放在命令字 0 后面的寄存器中。（见*页 65 命令结构*）

注意：设置命令字 0 之前必须将时间和日期信息存放在对应的 4:x 寄存器中。
注意：借助状态寄存器可以在处理过程中监控命令的成功执行。

命令 0 — NO OPERATION

概述

NO OPERATION 命令不对 ESI 模块进行任何操作。它用于允许多个扫描命令生成（设置命令字 1 至 11，然后设置命令字 0 开始执行命令），以及切换不连续运行的重复命令。

此命令持续执行，直到命令字 0 更改为 NO OPERATION 以外的其他命令。

命令结构

命令 0 的命令结构

字 0　0000（十六进制）

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

注意：对于命令 0，不使用命令字 1 至 11。

响应结构

命令 0 的响应结构

字 0　0000（十六进制）　回显命令字 0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

注 ： 位 15 是状态字有效位。

●
●
●

字 11　XXXX 十六进制　模块状态

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

注意：对于命令 0，命令字 1 至 10 返回 0。

命令 1 — READ ASCII MESSAGE

概述

READ ASCII MESSAGE 命令用于开始在模块上运行读取消息，即从串行端口的输入/接收缓冲区中取出 ASCII 字符，以符合消息的各种格式。所有的仅输出格式仍将 ASCII 字符发送到串行端口。

要开始读取消息，模块需要了解以下信息：

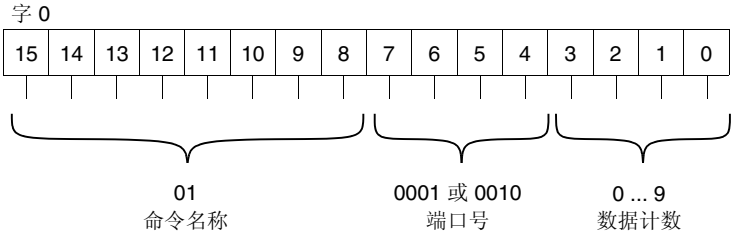
- 要使用的端口号
- 要处理的数据的起始模块寄存器编号
- 要运行的消息编号

除开始读取消息外，此命令还可以在消息完成后将最多 9 个（这是数据计数）寄存器数据从模块传输到控制器。返回的数据从命令字 1 中提供的起始寄存器编号获得。

此命令仅在第一次接收时执行。要再次执行该命令，需要更改命令字 0、1 或 2。这样，同一消息将不会持续运行，直到命令字 0 更改为 READ ASCII MESSAGE 以外的其他命令。

命令结构

命令 1 的命令结构



字 1　XXXX 十六进制 (XXXX = 0 ... 3FFF)　起始寄存器编号

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

字 2　XXXX 十六进制 (XX = 1 ... FF)　消息编号

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

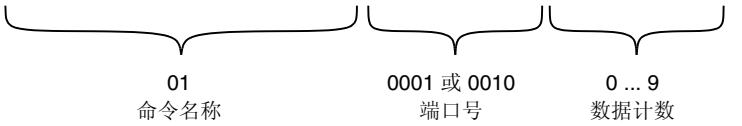
注意：对于命令 1，不使用命令字 3 至 11。

响应结构

命令 1 的响应结构

字 0　回显命令字 0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0



注 ： 位 15 是状态字有效位。

字 1　XXXX 十六进制 (XXXX = 0 ... 3FFF)　回显起始寄存器编号

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

字 2　XXXX 十六进制 (XX = 1 ... FF)　回显消息编号

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

字 3　XXXX 十六进制　数据字 1

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

●
●
●

字 11　XXXX 十六进制　模块状态或数据字 9

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

命令 2 — WRITE ASCII MESSAGE

概述

WRITE ASCII MESSAGE 命令用于开始在模块上运行写入消息，即将 ASCII 字符放入串行端口的输出/传输缓冲区。

要开始写入消息，模块需要了解以下信息：

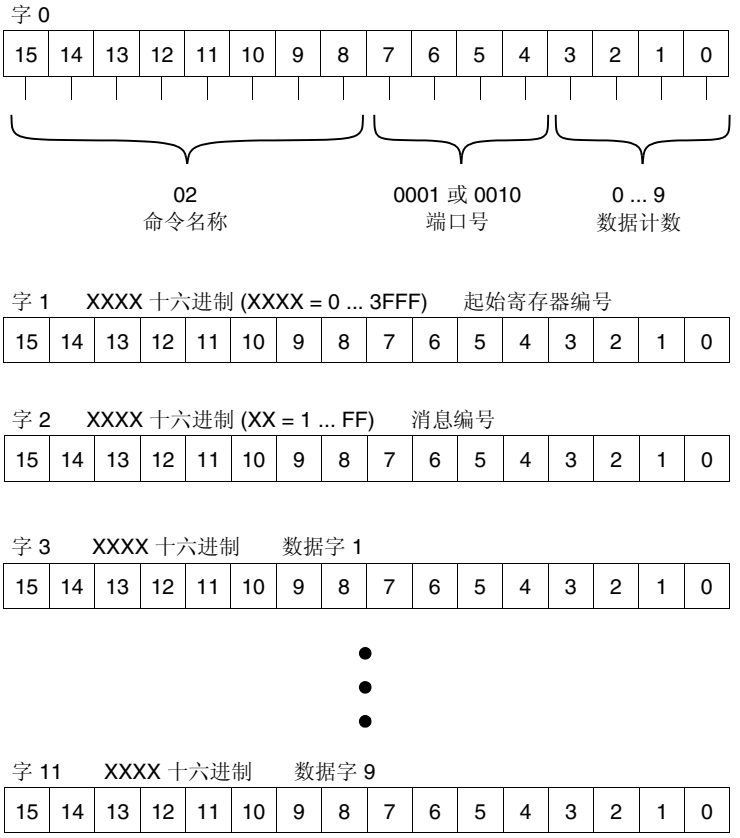
- 要使用的端口号
- 要处理的数据的起始模块寄存器编号
- 要运行的消息编号

除开始写入消息外，此命令还可以在消息开始写入前将最多 9 个（这是数据计数）寄存器数据从控制器传输到模块。发送的数据从命令字 1 中提供的起始寄存器编号开始存储。

此命令仅在第一次接收时执行。要再次执行该命令，需要更改命令字 0、1 或 2（加上任何发送的数据字 — 切断数据计数）。这样，同一消息将不会持续运行，直到命令字 0 更改为 WRITE ASCII MESSAGE 以外的其他命令。

命令结构

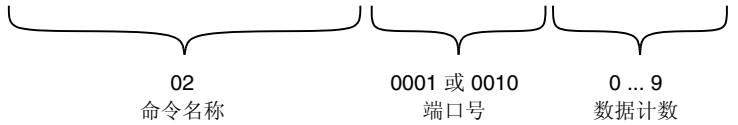
命令 2 的命令结构



响应结构　　命令 2 的响应结构

字 0　回显命令字 0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0



注 ： 位 15 是状态字有效位。

字 1　XXXX 十六进制 (XXXX = 0 ... 3FFF)　回显起始寄存器编号

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

字 2　XXXX 十六进制 (XX = 1 ... FF)　回显消息编号

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

字 3　XXXX 十六进制

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

●
●
●

字 11　XXXX 十六进制　模块状态

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

注意：对于命令 2，命令字 3 至 10 返回 0。

命令 3 — GET DATA（模块至控制器）

概述

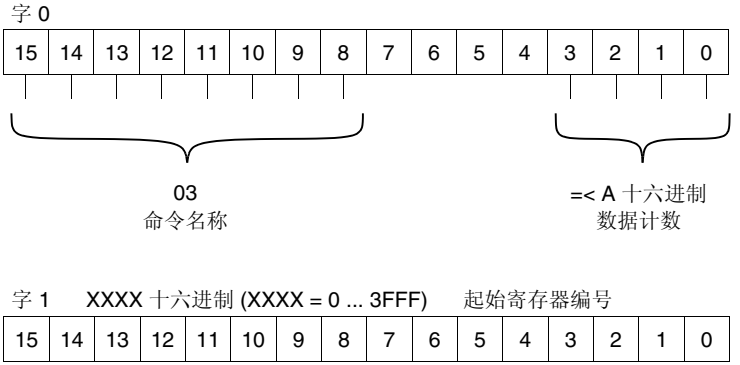
GET DATA 命令从命令字 1 提供的起始寄存器编号开始，从模块读取最多 10 个字/寄存器数据。命令字 0 提供的数据计数决定要读取的字数。数据在响应字 2 至 11 中返回。

注意：如果报告了错误状态（并且不是命令语法错误），且命令请求 10 个寄存器数据，则模块将仅返回 9 个字数据并使用响应字 11 作为模块状态。如果响应字 11 是模块状态，则将设置状态字数据位。

此命令持续执行，直到命令字 0 更改为 GET DATA 以外的其他命令。

命令结构

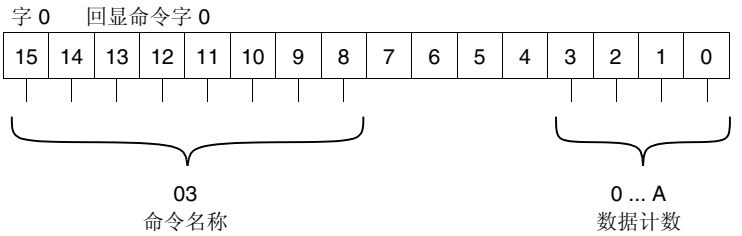
命令 3 的命令结构



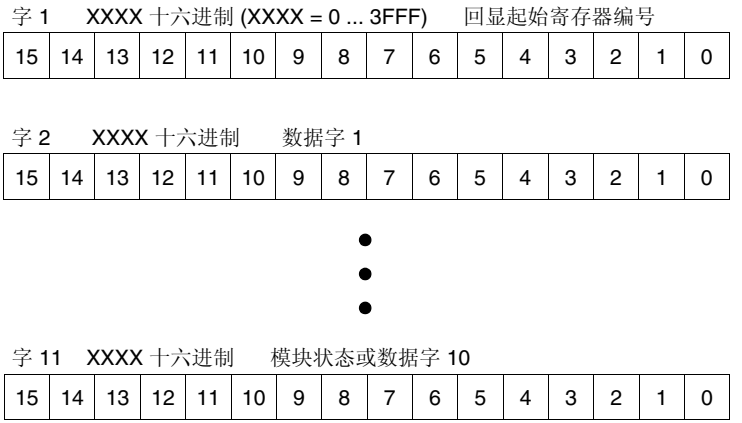
注意：对于命令 3，不使用命令字 2 至 11。

响应结构

命令 3 的响应结构



注 ： 位 15 是状态字有效位。



命令 4 — PUT DATA（控制器至模块）

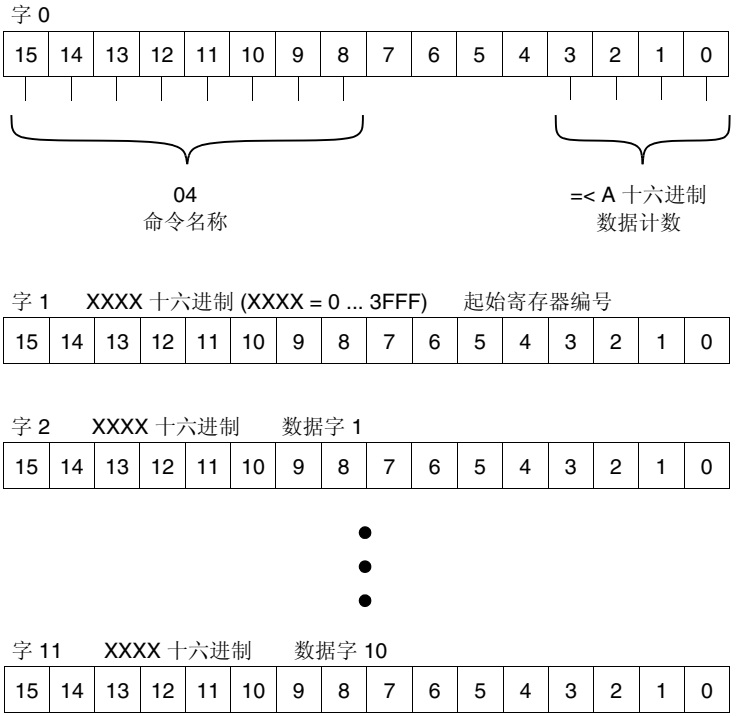
概述

PUT DATA 命令从命令字 1 提供的起始寄存器编号开始，将最多 10 个字/寄存器数据写入模块。数据在命令字 2 至 11 中发送。

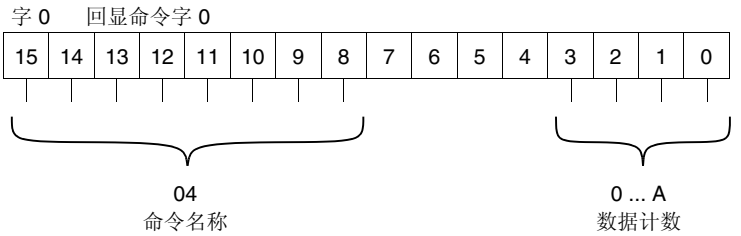
此命令持续执行，直到命令字 0 更改为 GET DATA 以外的其他命令。

命令结构

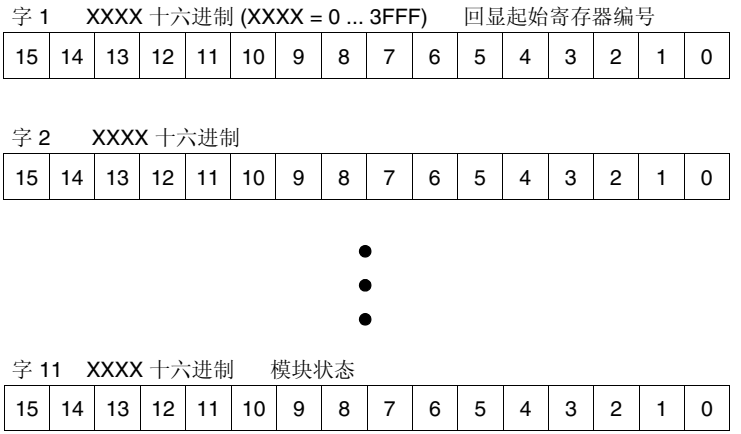
命令 4 的命令结构



响应结构　　命令 4 的响应结构



注 ： 位 15 是状态字有效位。



注意：对于命令 4，命令字 2 至 10 返回 0。

命令 5 — GET TOD（时间）

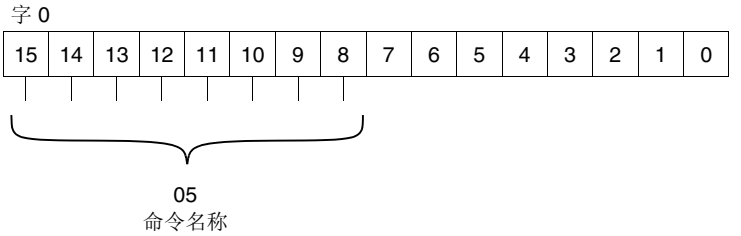
概述

GET TOD 命令读取模块 TOD 时钟并在响应字 1 至 7 中返回时间和日期。时间和日期的格式与 PLC 时间/日期寄存器使用的格式相同。

此命令持续执行，无需更改任何命令字。

命令结构

命令 5 的命令结构



注意：对于命令 5，不使用命令字 1 至 11。

响应结构

命令 5 的响应结构

字 0　回显命令字 0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

05
命令名称

注 ： 位 15 是状态字有效位。

字 1　XXXX 十六进制　星期几（1 = 星期日 ... 7 = 星期六）

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

字 2　XXXX 十六进制　月（一月 = 1 ... 十二月 = C [十进制的 12]）

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

字 3　XXXX 十六进制　日（1 ... 1F [十进制的 31]）

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

字 4　XXXX 十六进制　年（00 ... 63 [十进制的 99]）

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

字 5　XXXX 十六进制　小时（0 ... 17 [十进制的 23]）

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

字 6　XXXX 十六进制　分钟（0 ... 3B [十进制的 59]）

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

字 7　XXXX 十六进制　秒钟（0 ... 3B [十进制的 59]）

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

●
●
●

字 11　XXXX 十六进制　模块状态

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

注意：对于命令 5，命令字 8 至 10 返回 0。

命令 6 — SET TOD（时间）

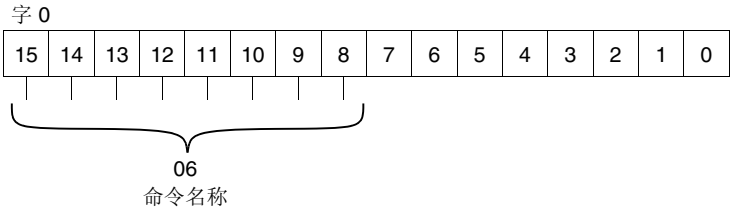
概述

SET TOD 命令用命令字 1 至 7 中提供的时间和日期加载模块 TOD 时钟。时间和日期的格式与 PLC 时间/日期寄存器使用的格式相同。

注意：为同步模块和 PLC 的 TOD 时钟，请对命令字 1 至 7 执行 PLC 的 7 个时间/日期寄存器的块移动，并将命令字 0 设置为 0600（十六进制）。

此命令仅在第一次接收时执行。要再次执行该命令，需要更改命令字 0 至 7 中的一个。这样，同一时间将不会持续加载，直到命令字 0 更改为 SET TOD 以外的其他命令。

命令结构　　命令 6 的命令结构



注 ： 位 15 是状态字有效位。

字 1　XXXX 十六进制　星期几（1 = 星期日 ... 7 = 星期六）

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

字 2　XXXX 十六进制　月（一月 = 1 ... 十二月 = C [十进制的 12]）

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

字 3　XXXX 十六进制　日（1 ... 1F [十进制的 31]）

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

字 4　XXXX 十六进制　年（00 ... 63 [十进制的 99]）

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

字 5　XXXX 十六进制　小时（0 ... 17 [十进制的 23]）

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

字 6　XXXX 十六进制　分钟（0 ... 3B [十进制的 59]）

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

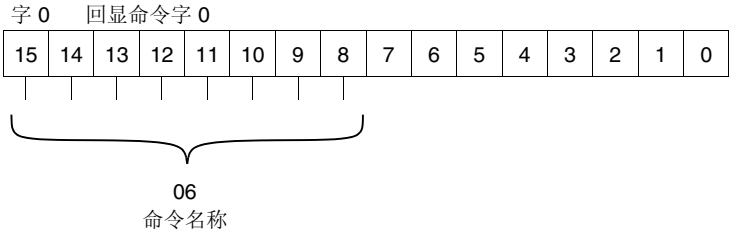
字 7　XXXX 十六进制　秒钟（0 ... 3B [十进制的 59]）

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

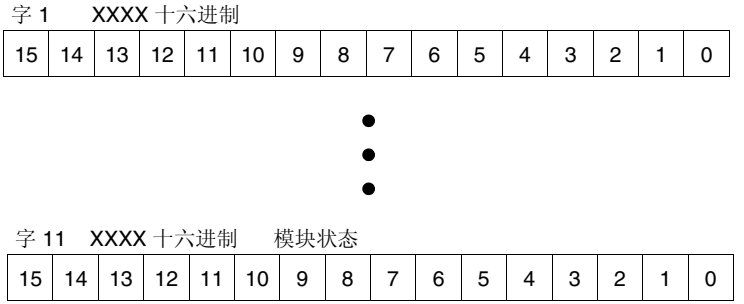
注意：对于命令 6，不使用命令字 8 至 11。

响应结构

命令 6 的响应结构



注 ： 位 15 是状态字有效位。



注意：对于命令 6，命令字 1 至 10 返回 0。

命令 7 — SET MEMORY REGISTERS

概述

SET MEMORY REGISTERS 命令将模块寄存器设置为命令字 3 中提供的值。设置的寄存器由下面的内容指定：起始寄存器编号和结束寄存器编号。从起始寄存器编号至结束寄存器编号（包含结束寄存器）的所有寄存器均设置为提供的值。

命令结构

命令 7 的命令结构

字 0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

07
命令名称

字 1　XXXX 十六进制 (XXXX = 0 ... 3FFF)　起始寄存器编号

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

字 2　XXXX 十六进制 (XXXX = 0 ... 3FFF)　结束寄存器编号

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

字 3　XXXX 十六进制　寄存器中设置的值

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

注意：对于命令 7，不使用命令字 4 至 11。

响应结构　　命令 7 的响应结构

字 0　回显命令字 0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

07
命令名称

注 ： 位 15 是状态字有效位。

字 1　XXXX 十六进制 (XXXX = 0 ... 3FFF)　回显起始寄存器编号

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

字 2　XXXX 十六进制 (XX = 1 ... FF)　回显消息编号

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

字 3　XXXX 十六进制　数据字 1

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

字 4　XXXX 十六进制

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

●
●
●

字 11　XXXX 十六进制　模块状态

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

注意：对于命令 7，命令字 4 至 10 返回 0。

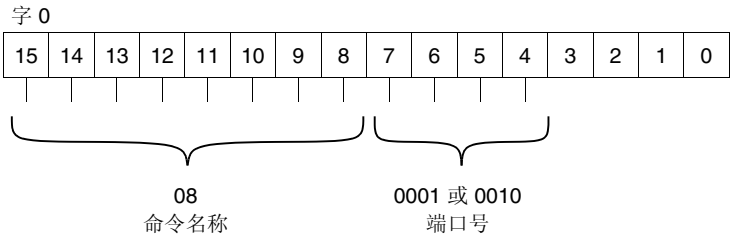
命令 8 — FLUSH BUFFER

概述

FLUSH BUFFER 命令为命令字中提供的串行端口刷新输入缓冲区。此命令不影响输出缓冲区。

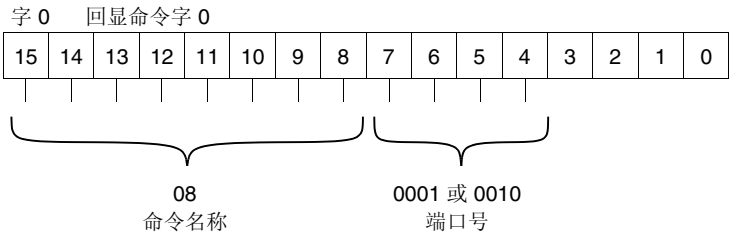
命令结构

命令 8 的命令结构

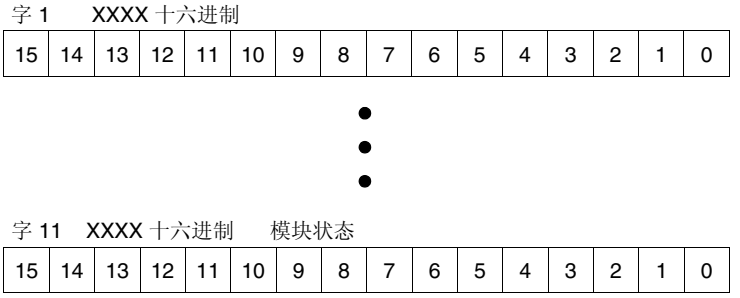


注意：对于命令 8，不使用命令字 1 至 11。

响应结构　　命令 8 的响应结构



注 ： 位 15 是状态字有效位。



注意：对于命令 8，命令字 3 至 10 返回 0。

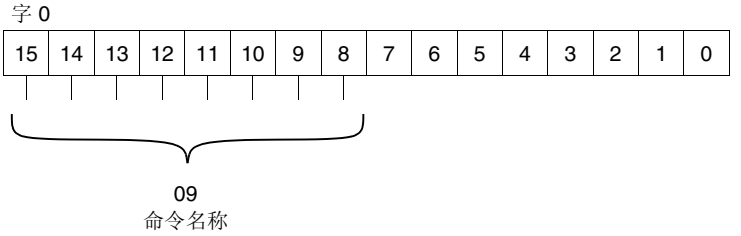
命令 9 — ABORT

概述

ABORT 命令中止运行中的 READ 或 WRITE ASCII MESSAGE，并且模块不再处于繁忙状态。此命令不影响模块的串行端口缓冲区，仅影响当前运行的消息。

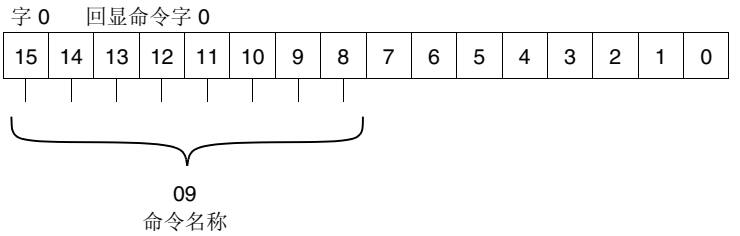
命令结构

命令 9 的命令结构

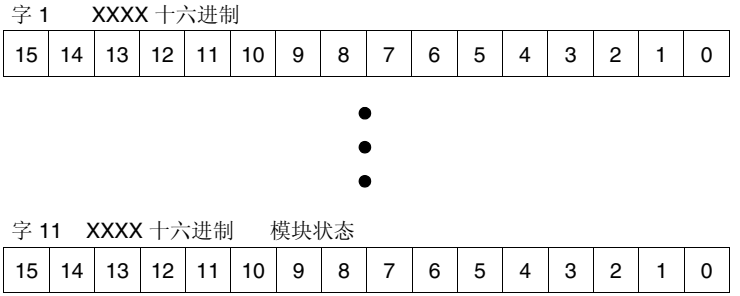


注意：对于命令 9，不使用命令字 1 至 11。

响应结构　　命令 9 的响应结构



注 ： 位 15 是状态字有效位。



注意：对于命令 9，命令字 3 至 10 返回 0。

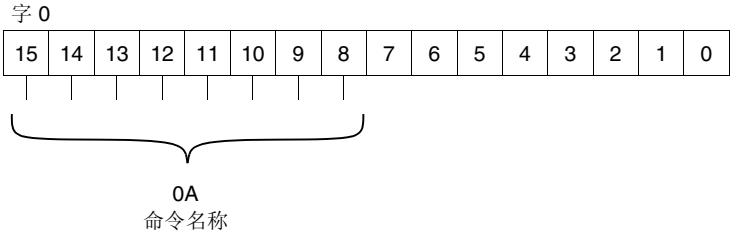
命令 A — GET BUFFER STATUS

概述

GET BUFFER STATUS 命令读取每个端口的输入缓冲区中的字符数。字符数范围为 1 ... 255。

命令结构

命令 A 的命令结构



注意：对于命令 A，不使用命令字 1 至 11。

响应结构　　命令 A 的响应结构

字 0　回显命令字 0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

0A
命令名称

注 ： 位 15 是状态字有效位。

字 1　端口 1 缓冲区状态

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

字 2　端口 2 缓冲区状态

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

字 3　XXXX 十六进制

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

●
●
●

字 11　XXXX 十六进制　模块状态

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

注意：对于命令 A，命令字 3 至 10 返回 0。

非法命令的响应结构

响应结构　　非法命令的响应结构

字 0　回显命令字 0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

注 ： 位 15 是状态字有效位。

字 11　XXXX 十六进制　模块状态

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

注意：命令字 1 至 10 返回 0。

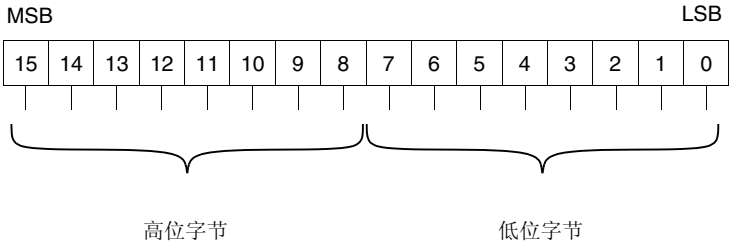
模块状态字（字 11）

概述

字 0（响应结构中）的位 15 置位时，模块状态字（响应结构中的字 11）包含有效的模块状态信息。此位的状态可区别响应结构中的字 11 正用于数据还是状态。

状态字的组成

状态字的组成：



注意：正常操作过程中，当字 11 用于 READ ASCII MESSAGE 或 GET DATA 命令中的模块状态或返回的数据时，模块状态信息尤其重要。

状态字内容

低字节

低字节的位								低字节（十六进制）	描述
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**		
0	0	0	0	0	0	0	1	0001	繁忙；模块上正在运行命令
0	0	0	0	0	0	1	0	0002	命令运行期间出现无效的消息数据
0	0	0	1	0	0	0	0	0100	命令运行期间到达寄存器末尾
0	0	1	0	0	0	0	0	0200	串行缓冲区溢出错误
0	1	0	0	0	0	0	0	0400	存储区域中的消息校验和错误 请参见消息编号的高字节
1	0	0	0	0	0	0	0	8000	错误；请参见消息编号的高字节

高字节

高字节的位								高字节（十六进制）	描述
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**		
0	0	0	0	0	0	0	1	0001	无效的用户逻辑参数
0	0	0	0	0	0	1	0	0002	无效的用户逻辑命令
0	0	0	1	0	0	0	0	0100	计数超出范围
0	0	0	1	0	0	0	1	0101	起始寄存器超出范围
0	0	0	1	0	0	1	0	0102	结束寄存器超出范围
0	0	0	1	0	0	1	1	0103	无效的寄存器编号顺序（结束编号在开始编号之前）
0	0	0	1	0	1	0	0	0104	请求的串行端口号无效
0	0	0	1	0	1	0	1	0105	请求的消息编号无效
0	0	0	1	0	1	1	0	0106	请求的消息编号未编程
0	0	0	1	0	1	1	1	0107	请求的消息编号位于错误的存储区域中
0	0	0	1	1	0	0	0	0108	配置参数错误
0	0	1	0	0	0	0	0	0200	星期几不正确

读取超过有效寄存器范围

概述

如果起始寄存器编号和数据计数有效，但一些要访问的寄存器超出有效寄存器范围，则仅读取/写入有效寄存器范围内的寄存器的数据。返回的数据计数是返回的有效寄存器数据数，并且在模块状态字中返回错误代码 1280（十六进制），表示结束寄存器编号超出范围。

示例

下面的示例尝试使用 GET 命令从 ESI 模块的 3FFA（十六进制）寄存器开始读取 10 个寄存器：

用户逻辑命令 = 030A（十六进制）

起始寄存器 = 3FFA（十六进制）

因此数据计数为 10，并返回 6 个有效寄存器（3FFA、3FFB、3FFC、3FFD、3FFE 和 3FFF，十六进制）数据。命令字中返回的数据计数为 6（8306，十六进制）。

假定下面的数据位于 ESI 寄存器中：

ESI 寄存器	**内容（十六进制）**
3FFA	1111
3FFB	2222
3FFC	3333
3FFD	4444
3FFE	5555
3FFF	6666

下表显示发送给 ESI 模块的命令及响应：

用户逻辑命令		**用户逻辑响应**	
寄存器	**内容**	**寄存器**	**内容**
4x+0	030A（十六进制）	3x+0	8306（十六进制）
4x+1	3FFA（十六进制）	3x+1	3FFA（十六进制）
4x+2	0000（十六进制）	3x+2	1111（十六进制）
4x+3	0000（十六进制）	3x+3	2222（十六进制）
4x+4	0000（十六进制）	3x+4	3333（十六进制）
4x+5	0000（十六进制）	3x+5	4444（十六进制）
4x+6	0000（十六进制）	3x+6	5555（十六进制）
4x+7	0000（十六进制）	3x+7	6666（十六进制）
4x+8	0000（十六进制）	3x+8	0000（十六进制）
4x+9	0000（十六进制）	3x+9	0000（十六进制）
4x+10	0000（十六进制）	3x+10	0000（十六进制）
4x+11	0000（十六进制）	3x+11	1280（十六进制）

概览

附录

概述

本附录提供额外的常规信息。

本附录包含了哪些内容？

本附录包含了以下章节：

章	章节标题	文件集
A	字符集	81
B	ESI 062 10 简介	85

字符集

A

概览

简介

本章描述标准 ASCII 字符集。

本章包含了哪些内容？

本章包含了以下主题：

主题	文件集
ASCII 字符集	82

ASCII 字符集

不可打印的 ASCII 字符

下表以十进制、十六进制、字符和控制字符值的形式定义 ASCII 字符集。

十进制	八进制	十六进制	字符	字符控制
0	00	00	NUL	空格
1	01	01	SOH	标题开始
2	02	02	STX	正文开始
3	03	03	ETX	正文结束
4	04	04	EOT	传输结束
5	05	05	ENQ	查询
6	06	06	ACK	确认
7	07	07	BEL	响铃
8	10	08	BS	退格
9	11	09	HT	水平制表符
10	12	0A	LF	换行
11	13	0B	VT	垂直制表符（主）
12	14	0C	FF	换页
13	15	0D	CR	回车
14	16	0E	SO	移出
15	17	0F	SI	移入
16	20	10	DLE	数据链路转义
17	21	11	DC1	设备控制一
18	22	12	DC2	设备控制二
19	23	13	DC3	设备控制三
20	24	14	DC4	设备控制四
21	25	15	NAK	否认
22	26	16	SYN	同步闲置
23	27	17	ETB	传输块结束
24	30	18	CAN	取消
25	31	19	EM	媒体结束
26	32	1A	SUB	替换
27	33	1B	ESC	转义
28	34	1C	FS	文件分隔符（右光标）
29	35	1D	GS	组分隔符（左光标）
30	36	1E	RS	记录分隔符（上光标）
31	37	1F	US	单元分隔符（下光标）

可打印的 ASCII 字符

下表以十进制、十六进制和字符的形式定义 ASCII 字符集。

十进制	八进制	十六进制	字符		十进制	八进制	十六进制	字符
32	40	20	空格		58	72	3A	:
33	41	21	!		59	73	3B	;
34	42	22	"		60	74	3C	<
35	43	23	#		61	75	3D	=
36	44	24	$		62	76	3E	>
37	45	25	%		63	77	3F	?
38	46	26	&		64	100	40	@
39	47	27	'		65	101	41	A
40	50	28	(		66	102	42	B
41	51	29	)		67	103	43	C
42	52	2A	*		68	104	44	D
43	53	2B	+		69	105	45	E
44	54	2C	,		70	106	46	F
45	55	2D	-		71	107	47	G
46	56	2E	.		72	110	48	H
47	57	2F	/		73	111	49	I
48	60	30	0		74	112	4A	J
49	61	31	1		75	113	4B	K
50	62	32	2		76	114	4C	L
51	63	33	3		77	115	4D	M
52	64	34	4		78	116	4E	N
53	65	35	5		79	117	4F	O
54	66	36	6		80	120	50	P
55	67	37	7		81	121	51	Q
56	70	38	8		82	122	52	R
57	71	39	9		83	123	53	S

可打印的 ASCII 字符集（续）：

十进制	八进制	十六进制	字符		十进制	八进制	十六进制	字符
84	124	54	T		106	152	6A	j
85	125	55	U		107	153	6B	k
86	126	56	V		108	154	6C	l
87	127	57	W		109	155	6D	m
88	130	58	X		110	156	6E	n
89	131	59	Y		111	157	6F	o
90	132	5A	Z		112	160	70	p
91	133	5B	[		113	161	71	q
92	134	5C	\		114	162	72	r
93	135	5D	]		115	163	73	s
94	136	5E	^		116	164	74	t
95	137	5F	_		117	165	75	u
96	140	60	`		118	166	76	v
97	141	61	a		119	167	77	w
98	142	62	b		120	170	78	x
99	143	63	c		121	171	79	y
100	144	64	d		122	172	7A	z
101	145	65	e		123	173	7B	{
102	146	66	f		124	174	7C	l
103	147	67	g		125	175	7D	}
104	150	68	h		126	176	7E	~
105	151	69	i		127	177	7F	

ESI 062 10 简介

B

概览

简介

本章概述了 140 ESI 062 10 ASCII 通讯模块的功能，并提供有关如何区分模块是否适合给定应用的帮助。

本章包含了哪些内容？

本章包含了以下主题：

主题	文件集
ESI 模块简介	86
应用标准	87
模块描述	88
ESI 模块结构图	89

ESI 模块简介

概述

Quantum ASCII 接口模块是一个通用的 ASCII 接口模块，它提供与第三方设备进行通讯和交换数据的能力。这些设备通常用于无法使用工业自动化惯用的标准通讯方法的工业环境中。标准通讯方法使用行业标准的 Modbus 通讯，该通讯方法定义数据查询和必要的响应字符串，以及实现可编程设备间通讯所需的物理接口。

目前的工业自动化中有许多通讯标准和现场总线可用。这些标准中很少有对串行数据流使用 RS 232C 物理介质的。多数串行数据信息都不是基于这些可用标准之一的；因此，需要使用 ASCII 接口。ASCII 通讯基于一种使用 RS232 或 RS422/485 物理介质的自定义串行协议。

物理介质

不同物理介质的特性：

标准	最长距离	物理属性	数据速率范围
RS232	15.24 米	点到点 使用调制解调器的多子站	180 bps 至 19200 bps
RS422	121.92 米	点到点 使用调制解调器的多子站	180 bps 至 19200 bps
RS485	范围广	多子站（内部调制解调器） 2 线或 4 线标准	180 bps 至 19200 bps

串行设备应用

这些 ASCII 应用中的多数都直接与打印机、条形码读卡器和扫描仪、串行设备（如磅秤、仪表和其他测量设备），以及工业自动化应用中使用的其他控制系统进行通讯。

这些第三方设备要求以它们能理解的语言进行通讯，以便能在第三方设备和 ASCII 模块之间传输数据。

例如，当测量包裹总重的磅秤接收到一个 "控制 A" ASCII 字符 <^A> 时，即返回包裹重量来进行响应。此数据存放在 ASCII 模块的存储器中，Quantum 控制器随后会读取该存储器。该控制器可能需要执行一个逻辑决策，即当重量大于一个特定的预定义值时将包裹送到哪里。因此，ASCII 模块只需知道外部设备进行通讯所需的协议和语言即可允许集成自动化应用中常见的数据。

应用标准

简介

Quantum PLC 系列提供了多种用于与外部设备通讯的解决方案。根据应用需要，用户可以选择软件解决方案（使用 CPU Modbus 端口的 XMIT 功能块）或硬件解决方案（ESI 模块或 ASCII Basic 模块）。下面的信息可帮助您为给定应用找到合适的解决方案。

应用标准

下表介绍典型应用和作为解决方案的建议使用的产品。在查找解决应用问题的方法时，此信息只作为一个参考，并不是解决应用问题的唯一答案。

应用	描述	建议的解决方案
打印机接口	使用来自控制器或 ASCII 模块的嵌入数据生成本地报告。	ESI 模块、J892 或 ASCIIBasic 模块
与简单设备通讯	发送控制字符并从测量设备接收数据。	ESI 模块、J892 或 XMIT
条形码接口	从条形码读取器/扫描仪发送和接收数据。	ESI 模块或 ASCII Basic 模块
与设备通讯	发送控制字符并从测量设备接收数据，设备可以发送前导零或前导空格。	ESI 模块或 J892
控制器到控制器接口	模仿支持多种子功能的制造商协议。生成复杂设备协议。	ASCII Basic 模块
外部数据存储	在控制器之外存储数据。	ESI 模块或 ASCII Basic 模块
Modbus 主站和/或调制解调器支持	使用控制字符生成所有 Modbus 主站命令和/或支持拨号调制解调器。	XMIT 功能块和控制器本地 Modbus 端口
多个 RS-232 端口	需要与外部设备通讯的多个端口	ESI 模块或 ASCII Basic 模块
分布式 I/O 中的 RS-232 端口	外部设备必须与分布式 I/O 连接	ESI 模块或 ASCII Basic 模块

模块描述

概述

ESI 模块包括 5 个主要的功能元素：

- 用于设备通讯的串行端口
- 通过背板与 Quantum 控制器连接的接口
- 端口缓冲区
- 寄存器存储器
- ASCII 消息存储器
- 固件

串行端口

ESI 模块已实现了 3 个逻辑通讯端口。端口 1 和端口 2 用于与外部串行设备通讯，而端口 0 用于对模块进行编程。端口 0 和端口 1 共享一个物理端口。所有 3 个端口都可独立设置。有关端口设置的详细描述，请参见*端口命令*。

到 Quantum 控制器的接口

ESI 模块与 Quantum 控制器交换数据，12 个输出字用于来自 Quantum 控制器的命令和数据，12 个输入字用于传输给 Quantum 控制器的数据以及命令回显和状态信息。有关命令结构和响应结构的详细信息，请参见 *ESI 命令结构*。

端口缓冲区

ESI 模块的 2 个物理端口各有一个输入缓冲区和一个输出缓冲区，缓冲区大小都为 255 个字符。这些缓冲区的设备端由可选的 XON/XOFF 信号交换自动维护。对于进出 Quantum 控制器的数据传输，有多个命令可用于缓冲区控制和状态测试，详见*数据流*。

寄存器存储器

ESI 模块有一个 32 kB 的存储器，该存储器组织为 16k 个 16 位寄存器。这些寄存器存放来自串行端口和发送到串行端口的所有数据。可以通过 PUT 命令和 GET 命令对它们进行访问。

ASCII 消息存储

ESI 模块最多可存放 255 条 ASCII 消息，每条消息包含 127 个字符和校验和字符。这些 ASCII 消息可以是要发送到外部设备的静态文本，或者是说明如何在寄存器区域中包含的数据和串行 ASCII 字符流两者之间相互转换的定义，也可是两者的组合。

固件

ESI 模块的固件可通过本地 I/O 背板加载。可通过更新 ESI 模块内的闪存执行固件来支持升级和更改功能。用户应知道更新过程只能通过本地 I/O 背板完成，尽管模块可以放置在本地、远程或分布位置。如果是在远程或分布式背板中使用 ESI 模块，请确保本地背板中有一个可用的空插槽，或者有一个备用的控制器系统以便将来进行升级。

ESI 模块结构图

ESI 模块的元素　　下图显示 ESI 模块的元素：