

使用 EcoStruxure™ Control Expert 的 Quantum 140 ESI 062 10 ASCII 接口模块 用户手册

原始指令翻译

10/2019

本文档中提供的信息包含有关此处所涉及产品之性能的一般说明和/或技术特性。本文档并非用于(也不代替)确定这些产品对于特定用户应用场合的适用性或可靠性。任何此类用户或设备集成商都有责任就相关特定应用场合或使用方面对产品执行适当且完整的风险分析、评估和测试。

Schneider Electric 或其任何附属机构或子公司对于误用此处包含的信息而产生的后果概不负责。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。

本手册可用于法律所界定的个人以及非商业用途。在未获得施耐德电气书面授权的情况下，不得翻印传播本手册全部或部分相关内容、亦不可建立任何有关本手册或其内容的超文本链接。施耐德电气不对个人和非商业机构进行非独占许可以外的授权或许可。请遵照本手册或其内容原义并自负风险。与此有关的所有其他权利均由施耐德电气保留。

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帮助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。

当设备用于具有技术安全要求的应用场合时，必须遵守有关的使用说明。

未能使用施耐德电气软件或认可的软件配合我们的硬件，则可能导致人身伤害、设备损坏或不正确的运行结果。

不遵守此信息可能导致人身伤害或设备损坏。

© 2019 Schneider Electric. 保留所有权利。



	安全信息	5
	关于本书	9
第1章	140 ESI 062 10 硬件描述	11
	介绍	12
	LED 指示灯	13
	外部连接器和开关	15
	规格	17
第2章	Quantum 寻址模式	21
	平面寻址—800 系列 I/O 模块	22
	拓扑寻址 - 800 系列 I/O 模块 (使用 Control Expert)	23
	寻址示例	24
	离散量 I/O 位编号	25
	140 ESI 062 10 模块寻址	26
第3章	配置概述	27
	140 ESI 062 10 配置	28
	ASCII 消息格式	30
	数据流	36
	参数配置	39
第4章	ESI 命令行编辑器	41
	配置编辑器	42
	ASCII 消息编辑器	45
第5章	ESI 命令	47
	ESI 命令概述	48
	ESI 命令字	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 (从模块到控制器)	58
	命令 4 - PUT DATA (从控制器到模块)	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	70

	命令 A - GET BUFFER STATUS	71
	非法命令的响应结构	73
	模块状态字 (字 11)	74
	读取超出有效寄存器范围	76
附录	79
附录 A	字符集	81
	ASCII 字符集	81
附录 B	ESI 062 10 简介	85
	ESI 模块简介	86
	应用标准	87
	模块描述	88
	ESI 模块结构图	89
索引	91



重要信息

声明

在试图安装、操作、维修或维护设备之前，请仔细阅读下述说明并通过查看来熟悉设备。下述特定信息可能会在本文其他地方或设备上出现，提示用户潜在的危險，或者提醒注意有关阐明或简化某一过程的信息。



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



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險。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。

危險

危險表示若不加以避免，将会导致严重人身伤害甚至死亡的危險情况。

警告

警告表示若不加以避免，可能会导致严重人身伤害甚至死亡的危險情况。

小心

小心表示若不加以避免，可能会导致轻微或中度人身伤害的危險情况。

注意

注意用于表示与人身伤害无关的危害。

请注意

电气设备的安装、操作、维修和维护工作仅限于有资质的人员执行。施耐德电气不承担由于使用本资料所引起的任何后果。

有资质的人员是指掌握与电气设备的制造和操作及其安装相关的技能和知识的人员，他们经过安全培训能够发现和避免相关的危险。

开始之前

不得将本产品在没有有效作业点防护的机器上使用。如果机器上缺少有效的作业点防护，则有可能导致机器的操作人员严重受伤。

警告

未加以防护的设备

- 不得将此软件及相关自动化设备用在不具有作业点防护的设备上。
- 在操作期间，不得将手放入机器。

不遵循上述说明可能导致人员伤亡或设备损坏。

此自动化设备及相关软件用于控制多种工业过程。根据所需控制功能、所需防护级别、生产方法、异常情况、政府法规等因素的不同，适用于各种应用的自动化设备的类型或型号会有所差异。在某些应用情况下，如果需要后备冗余，则可能需要一个以上的处理器。

只有用户、机器制造商或系统集成商才能清楚知道机器在安装、运行及维护过程中可能出现的各种情况和因素，因此，也只有他们才能确定可以正确使用的自动化设备和相关安全装置及互锁设备。在为特定应用选择自动化和控制设备以及相关软件时，您应参考适用的当地和国家标准及法规。National Safety Council's Accident Prevention Manual（美国全国公认）同样提供有非常有用的信息。

对于包装机等一些应用而言，必须提供作业点防护等额外的操作人员防护。如果操作人员的手部及其他身体部位能够自由进入夹点或其他危险区域内，并且可导致人员严重受伤，则必须提供这种防护。仅凭软件产品自身无法防止操作人员受伤。因此，软件无法被取代，也无法取代作业点防护。

在使用设备之前，确保与作业点防护相关的适当安全设备与机械/电气联锁装置已经安装并且运行。与作业点防护相关的所有联锁装置与安全设备必须与相关自动化设备及软件程序配合使用。

注意：关于协调用于作业点防护的安全设备与机械/电气联锁装置的内容不在本文档中功能块库、系统用户指南或者其他实施的范围之内。

启动与测试

安装之后，在使用电气控制与自动化设备进行常规操作之前，应当由合格的工作人员对系统进行一次启动测试，以验证设备正确运行。安排这种检测非常重要，而且应该提供足够长的时间来执行彻底并且令人满意的测试。

警告

设备操作危险

- 验证已经完成所有安装与设置步骤。
- 在执行运行测试之前，将所有元器件上用于运送的挡块或其他临时性支撑物拆下。
- 从设备上拆下工具、仪表以及去除碎片。

不遵循上述说明可能导致人员伤亡或设备损坏。

执行设备文档中所建议的所有启动测试。保存所有设备文档以供日后参考使用。

必须同时在仿真与真实的网络境中进行软件测试。

按照地方法规（例如：依照美国 National Electrical Code）验证所完成的系统无任何短路且未安装任何临时接地线。如果必须进行高电位电压测试，请遵循设备文档中的建议，防止设备意外损坏。

在对设备通电之前：

- 从设备上拆下工具、仪表以及去除碎片。
- 关闭设备柜门。
- 从输入电源线中拆除所有的临时接地线。
- 执行制造商建议的所有启动测试。

操作与调节

下列预防措施来自于NEMA Standards Publication ICS 7.1-1995（以英文版本为准）：

- 无论在设计与制造设备或者在选择与评估部件时有多谨慎，如果对此类设备造作不当，将会导致危险出现。
- 有时会因为对设备调节不当而导致设备运行不令人满意或不安全。在进行功能调节时，始终以制造商的说明书为向导。进行此类调节的工作人员应当熟悉设备制造商的说明书以及与电气设备一同使用的机器。
- 操作人员应当只能进行操作人员实际所需的运行调整。应当限制访问其他控件，以免对运行特性进行擅自更改。

关于本书



概览

文档范围

本文档介绍 ASCII 接口模块的安装和使用。

有效性说明

此文档适用于 EcoStruxure™ Control Expert 14.1 或更高版本。

本文中描述的设备技术特性在网站上也有提供。要在线访问此信息：

步骤	操作
1	访问 Schneider Electric 主页 www.schneider-electric.com 。
2	在 Search 框中键入产品参考号或产品系列名称。 <ul style="list-style-type: none">勿在参考号或产品系列中加入空格。要获得有关类似模块分组的信息，请使用星号 (*)。
3	如果您输入的是参考号，则转至 Product Datasheets 搜索结果，单击您感兴趣的参考号。如果您输入产品系列的名称，则转到 Product Ranges 搜索结果，单击您感兴趣的产品系列。
4	如果 Products 搜索结果中出现多个参考号，请单击您感兴趣的参考号。
5	根据屏幕大小，您可能需要向下滚动查看数据表。
6	要将数据表保存为 .pdf 文件或打印数据表，请单击 Download XXX product datasheet 。

本手册中介绍的特性应该与在线显示的那些特性相同。依据我们的持续改进政策，我们将不断修订内容，使其更加清楚了，更加准确。如果您发现手册和在线信息之间存在差异，请以在线信息为准。

相关文档

文档标题	参考号
EcoStruxure™ Control Expert 程序语言和结构参考手册	35006144 (英语)、35006145 (法语)、 35006146 (德语)、35013361 (意大利语)、 35006147 (西班牙语)、35013362 (简体中文)
使用 EcoStruxure™ Control Expert 的 Quantum 硬件参考手册	35010529 (英语)、35010530 (法语)、 35010531 (德语)、35013975 (意大利语)、 35010532 (西班牙语)、35012184 (简体中文)
使用 EcoStruxure™ Control Expert 的 Quantum 离散量和模拟量 I/O 参考手册	35010516 (英语)、35010517 (法语)、 35010518 (德语)、35013970 (意大利语)、 35010519 (西班牙语)、35012185 (简体中文)

文档标题	参考号
使用 EcoStruxure™ Control Expert 的 Quantum 专家和通讯参考手册	35010574 (英语)、35010575 (法语)、 35010576 (德语)、35014012 (意大利语)、 35010577 (西班牙语)、35012187 (简体中文)
Electrical installation guide	EIGED306001EN (English)
通讯服务和架构参考手册	35010500 (英语)、35010501 (法语)、 35006176 (德语)、35013966 (意大利语)、 35006177 (西班牙语)、35012196 (简体中文)

您可以在我们的网站 www.schneider-electric.com/en/download 下载这些技术出版物和其他技术信息。

第1章

140 ESI 062 10 硬件描述

简介

本章描述 140 ESI 062 10 ASCII 接口模块的硬件特性。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	页
介绍	12
LED 指示灯	13
外部连接器和开关	15
规格	17

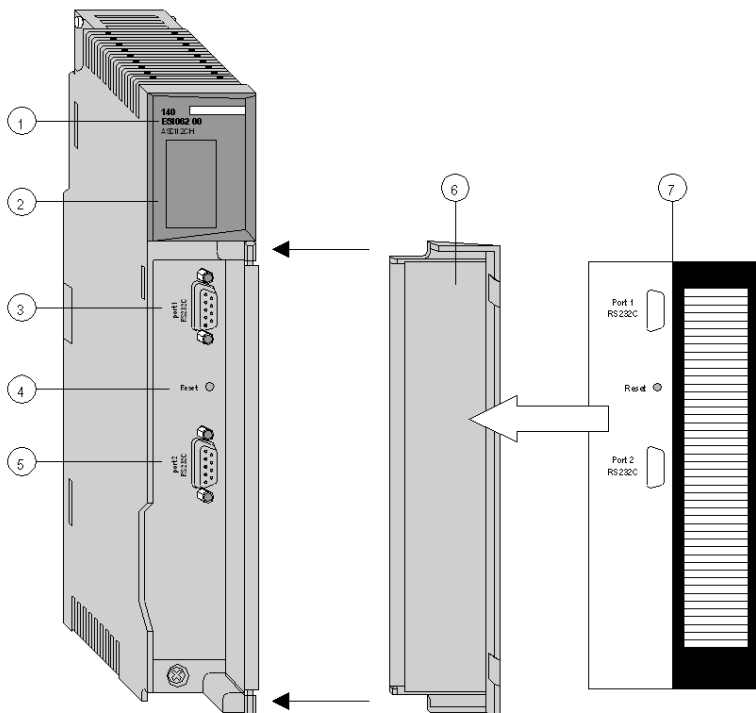
介绍

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。

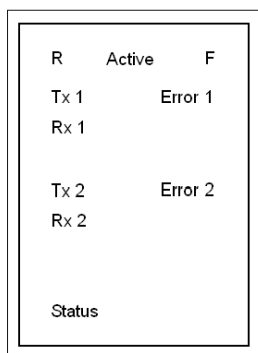


- 1 型号、模块描述、颜色代码
- 2 LED 显示
- 3 端口 1 连接器
- 4 复位按钮
- 5 端口 2 连接器
- 6 可拆除门
- 7 客户标识标签（折叠标签并将标签放置于门内）

LED 指示灯

LED 显示位置

LED 显示包含位于 140 ESI 062 10 模块正上方的 10 个指示灯。



指示内容

下表描述各个 LED 亮起时所指示的内容。

LED	颜色	指示
R	绿色	模块已通过加电诊断。
Active	绿色	总线通讯存在。
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
Status	黄色	状态
Error 1	红色	端口 1 上出现错误情况
Error 2	红色	端口 2 上出现错误情况

闪烁序列

F、Status、Error 1 和 Error 2 LED 可按序列闪烁以指示以下情况：

F	Status	Error 1	Error 2	状况
闪烁	闪烁	闪烁	闪烁	ASCII 模块正在初始化 首次加电
灭	亮	灭	灭	编程模式
灭	灭	亮	N/A	串行口 1 发生缓冲区溢出
灭	灭	N/A	亮	串行口 2 发生缓冲区溢出
N/A	闪烁 (参见 崩溃代码)	灭	灭	该模块处于内核模式下，可能出现错误

崩溃代码指示

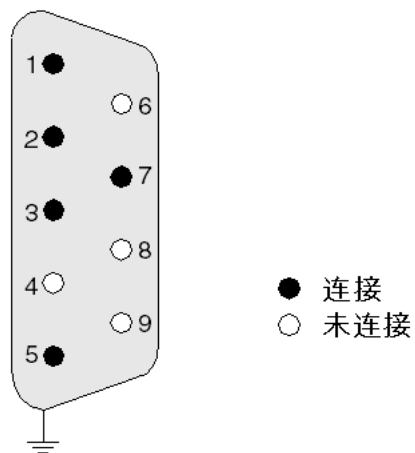
状态 LED 以各种模式闪烁以指示模块的崩溃代码。

闪烁次数	代码 (以十六 进制表示)	错误
常亮	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 (未加载执行程序)
	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	意外的执行返回

外部连接器和开关

RS-232 串行口

ASCII 模块有两个 RS-232 串行口，用于与串行设备通讯。



下面是这两个串行口的引脚连接：

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

编程端口

端口 1 也可用作编程端口（端口 0）。按住**复位**按钮超出 4 秒可进入编程模式。在编程模式下，串行口设置为终端通讯配置。

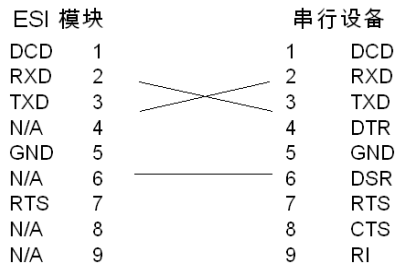
在编程模式下，该端口使用以下参数：

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

端口配置已设置为使其成为一个已知配置；这一配置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。此**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 个串行口 (9 针 D 型) , 无隔离
接线 (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)

功率

电源规格

功耗	2 W (最大值)
总线电流要求	300 毫安

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

In	12 个字
输出	12 个字

兼容性

兼容性

编程软件	Concept 2.5 或更高版本、ProWorx NxT、ProWorx 32、Modsoft、Control Expert
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、控制码、刷新缓冲区
Quantum 控制器	所有，最低 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 (高 x 长 x 宽)	250 mm x 103.85 mm x 40.34 mm
材料	(机箱和前盖) Lexan
空间要求	1 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
快速瞬变 (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% 相对湿度，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 米
振动	10...57 Hz (0.075 mm d.a.) 57...150 Hz (1 g)
冲击	+/-15 g (峰值)，11 毫秒，半正弦波

存储条件

存储条件

温度	-40...85°C (-40...185°F)
湿度	0...95% 相对湿度，无冷凝 (60°C)
自由下落	1 米

机构核准

机构核准

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

第2章

Quantum 寻址模式

概述

在此专用模块的功能描述中，广泛地使用了在 Quantum 产品中建立的 %IW/%MW (3x/4x) 寄存器寻址模式。本章描述在 Control Expert 中用于从 Quantum 模块对数据进行寻址的不同模式。

注意： 拓扑地址重叠 (%IW_{r.m.c}) 不受 Quantum 应用程序支持，在需要存储器重叠控制时使用平面寻址 (%IW_x)。

本章包含了哪些内容？

本章包含了以下主题：

主题	页
平面寻址—800 系列 I/O 模块	22
拓扑寻址 - 800 系列 I/O 模块 (使用 Control Expert)	23
寻址示例	24
离散量 I/O 位编号	25
140 ESI 062 10 模块寻址	26

平面寻址—800 系列 I/O 模块

简介

在 Control Expert 中，800 系列 I/O 模块遵循平面地址映射系统。为确保正常工作，每个模块都需要确定位和（或）字的数量。IEC 寻址系统等效于 984LL 寄存器寻址。请按以下方式指定：

- 0x 现对应于 %Mx
- 1x 现对应于 %Ix
- 3x 现对应于 %IWx
- 4x 现对应于 %MWx

下表显示了 984LL 表示形式和 IEC 表示形式之间的关系。

输出和输入	984LL 符号 寄存器地址	IEC 符号		
		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 现对应于 %M1

100101 现对应于 %I101

301024 现对应于 %IW1024

400010 现对应于 %MW10

拓扑寻址 - 800 系列 I/O 模块 (使用 Control Expert)

访问 I/O 数据值

可以使用拓扑寻址来访问 I/O 数据项。请使用下面的表示形式在 Control Expert 中标识模块在 800 系列 I/O 模块中的拓扑位置：

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

其中：

- **b** = 总线
- **e** = 设备 (子站)
- **r** = 机架
- **m** = 模块插槽
- **c** = 通道

注意：寻址时，

1. 在本地机架中 [b.e] 缺省设置为 \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，请参阅 *EcoStruxure™ Control Expert 程序语言和结构参考手册*。

读取值：示例

读取	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)：	Enter %IW1.6.7[.0]
从位于 RIO 总线 2 的子站 3 的插槽 6 中的模拟量模块的通道 7 中读取输入值 (序号 = 0)：	Enter %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)：	Enter %I1.6.7.1[.0]

寻址示例

比较 3 种寻址模式

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 超出范围	%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。

为进行比较，将 Concept 寄存器寻址添加在最后一列中。因为 Concept 不支持直接对字中的位进行寻址，所以位抽取操作必须在用户程序中执行。

离散量 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)。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。这可以在配置时通过选择 %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，则可以使用语法 %word.bit。下表提供了 I/O 点编号和以位以及字寻址表示的相关 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 %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

140 ESI 062 10 模块寻址

平面寻址

140 ESI 062 10 ASCII 接口模块需要 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

140 ESI 062 10 模块的拓扑地址如下所示：

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

其中：**b** = 总线，**e** = 设备（子站），**r** = 机架，**m** = 模块插槽

注意： I/O 字 2 ...12 用于模块与 CPU 间的数据交换，具体取决于活动的命令。

第3章

配置概述

概述

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	页
140 ESI 062 10 配置	28
ASCII 消息格式	30
数据流	36
参数配置	39

140 ESI 062 10 配置

概述

140 ESI 062 10 模块具有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

140 ESI 062 10 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。

进入配置模式

要进入配置模式，请执行以下步骤：

步骤	操作
1	将鼓终端或 PC 终端仿真器（如超级终端）连接到端口 1。有关适用电缆的消息，请参见 <i>RS-232 串行口</i> ，第 15 页
2	将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	按住模块前面的 复位 按钮达 4 秒以上。

命令行编辑器

进入配置模式后，前面板上的黄色**状态** LED 将亮起，终端屏幕上将显示以下消息：

欢迎

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

命令行编辑器中提供以下命令结构：

命令	描述	示例
CLI	将编程模式设置为命令行解释程序。	N/A
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 将显示关于 ASCII 命令的帮助）。	N/A
RUN	复位模块并进入正常运行模式。	N/A
CONFIG	将编程模式设置为配置解释程序。	N/A

命令	描述	示例	
	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。	N/A	
	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>upload
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>upload (message # - message #)	

ASCII 消息格式

ASCII 消息用于从 140 ESI 062 10 模块向 ASCII 设备（如终端程序）发送信息。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	输入/输出	ASCII 字符
十六进制	输入/输出	十六进制数字
八进制	输入/输出	八进制数字
二进制	输入/输出	二进制数字
整数	输入/输出	整数数字
定点十进制	输入/输出	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	输入/输出	消息嵌套

文本格式

包含在单引号内的任意 ASCII 字符串（例如，'message string'）是纯输出格式。任何包含此格式的消息都将发送文本，无论消息从读取消息命令还是写入消息命令开始。

'... (文本) ...'

ASCII 格式

带有寄存器数量和字段长度的 ASCII 格式变量字段：

nAm

其中：

- n 为寄存器数量 1..99（格式重复）
- m 为字段长度 1..2（字符数）

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

十六进制格式

带有寄存器数量和字段长度的十六进制格式的变量字段：

nHm

其中：

- 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)

例如，2H3 作为输入表示 2 个寄存器，每个包含 3 个十六进制数字。

八进制格式

带有寄存器数量和字段长度的八进制格式的变量字段：

nOm

其中：

- 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)

例如，3O4 作为输入表示 3 个寄存器，每个包含 4 个八进制数字。

二进制格式

带有寄存器数量和字段长度的二进制格式的变量字段：

nBm

其中：

- 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)

例如，1B8 作为输入表示 1 个包含 8 个二进制数字的寄存器。

整数格式，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零。

nIm

其中：

- 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)

例如，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。

整数格式，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零。

nLm

其中：

- 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)

例如，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。

定点十进制格式

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零。

nPm.q

其中：

- 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

例如，1P7.2 作为输入表示 1 个寄存器，其中包含 4 个十进制数字，后接十进制小数点和 2 个十进制数字 (小数部分)。

注意：请勿将此格式与浮点格式混淆。小数点的位置用于输入/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)。

嵌套消息格式

嵌套消息格式允许一条消息调用其他消息。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，并允许间接嵌套重复。允许的最大嵌套消息层为 8 层。不允许递归嵌套。

Mn

其中 n 为消息编号 1..255

例如，M6 将运行编号为 6 的消息。

时间格式

可使用两种不同的格式来显示时间，即 12 小时格式和 24 小时格式。这是纯输出格式。

T12 > hh:mm:ss AM/PM (12 小时时间)

T24 > hh:mm:ss (24 小时时间)

日期格式

可使用 5 种不同的格式来显示日期，每种提供 2 种显示年份的格式类型。这是纯输出格式。

Dnm

其中：

- 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)

多个格式的重复

嵌套重复括号无效。

n(...)

其中 n 是要重复 () 中的内容的次数 1..99

例如：6('Item',112,4X,115,/) 将生成 6 行，每行均包含字段 'Item'、112、4X、115 和一个 <CR, LF>。

空格

ASCII 消息中表示空格的符号是 X。这是纯输出格式。

nX

其中 n 为空格的数量 1..99

换行

ASCII 消息中表示回车符的符号是 `\r`。这是纯输出格式。

控制代码

控制代码显示为包含在双引号分隔符中的 3 位八进制字符 (范围为 000 到 377) 这是纯输出格式。

`"###"`

其中, `###` 是字符的八进制格式

例如: `"033"`。

刷新

刷新当前正在运行的串行口的输入缓冲区有四种方法: 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

其中:

- `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，就可以使用 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，以节约空间并对消息进行标准化，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：

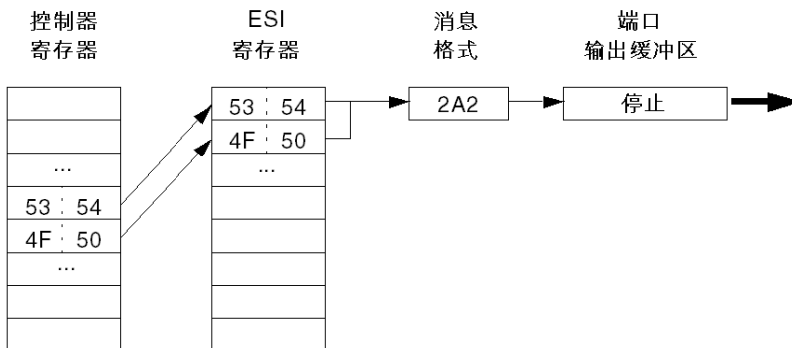
- 根据 ASCII 消息解释端口接收缓冲区中的数据并将其传输到 ESI 寄存器区域。
- 数据从 ESI 寄存器区域通过 I/O 配置中分配给 ESI 模块的 12 个输入寄存器传输到 PLC 寄存器。

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*，第 30 页。

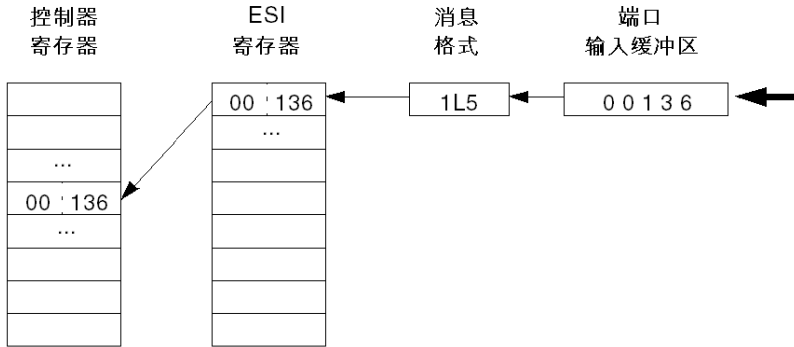
传输示例

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

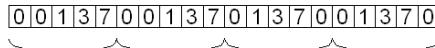
由于 ESI 模块仅支持没有开始或终止字符的固定长度消息格式，丢失任何字符（或增加预期之外的字符）都可导致对接收到的数据生成错误解释。以下示例显示 3 种不同错误类型的结果。假定消息格式为 "1L5"（最大值为 65,535）：

丢失字符的结果：

错误原因：

丢失一个字符

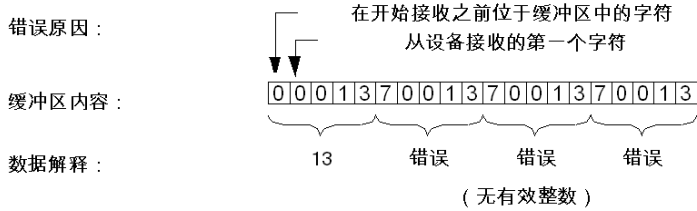
缓冲区内容：



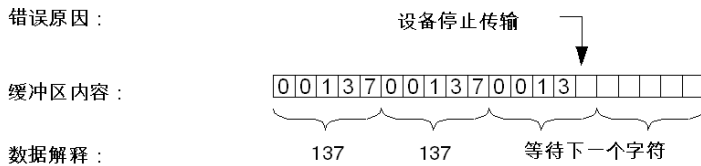
数据解释：

137 137 1370 1370

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*, 第 48 页。

参数配置

概述

参数编辑器是 ESI 062 10 模块的 Control Expert 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	说明
映射	字(%IW-3X%MW-4X)	-	
输入起始地址	1	-	
输入结束地址	12	-	
输出起始地址	1	-	
输出结束地址	12	-	
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			

名称	缺省值	选项	说明
PORT_0、PORT_1、PORT_2			
波特率	9600	300-19200	
数据位	8	7	
校验位	无 (PORT_0) 偶 (PORT_1,PORT_2)	奇	
停止位	1	2	
键盘	开 (PORT_0) 关 (PORT_1,PORT_2)	开/关	
XON/XOFF	启用	禁用	

注意： 以下两种配置不得应用于端口 1：

- 配置 1：
 - 数据位参数设置为 8
 - 校验位参数设置为启用，或设置为偶校验或奇校验
 - 停止位参数设置为 2
- 配置 2：
 - 数据位参数设置为 7
 - 校验位参数设置为无
 - 停止位参数设置为 1

如果这两种配置之一使用端口 1，则会出现数据传输错误。

第4章

ESI 命令行编辑器

概述

ESI 固件包含可通过端口 1 连接的虚拟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	页
配置编辑器	42
ASCII 消息编辑器	45

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口和时间时钟。

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

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	校验位设置	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)	开、关
x	XON/XOFF 模式 (软件流控制)	开、关

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	1 ... 12
dd	日	1 ... 31
yy	年	00 ... 99
yyyy	年	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>

第5章

ESI 命令

简介

本章描述由 CPU 发送以控制 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	页
ESI 命令概述	48
ESI 命令字	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 (从模块到控制器)	58
命令 4 - PUT DATA (从控制器到模块)	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	70
命令 A - GET BUFFER STATUS	71
非法命令的响应结构	73
模块状态字 (字 11)	74
读取超出有效寄存器范围	76

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 命令字

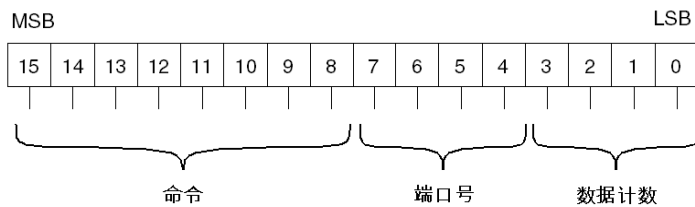
命令字格式

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

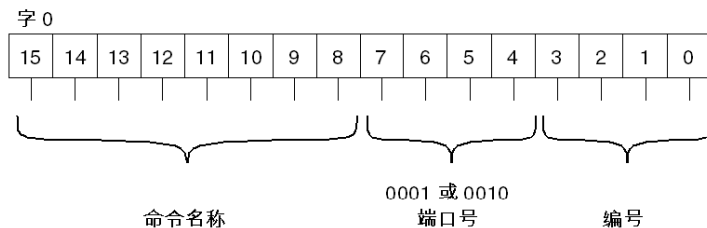
示例 5 GET TOD

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：



例如: 字 0

0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

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 (参见第 63 页) 中。

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 (参见第 65 页) 后面的寄存器中。

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。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。它的作用是允许生成多个扫描命令（设置命令字 1 至 11，然后设置命令字 0 以开始执行命令），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

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

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，即从串行口的输入/接收缓冲区取出 ASCII 字符，以实现各种格式的消息。所有的纯输出格式仍将 ASCII 字符发送到串行口。

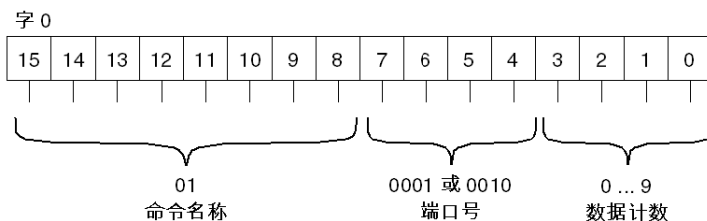
要开始写入消息，模块需要了解以下信息：

- 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

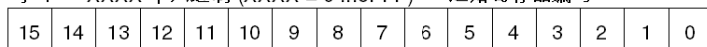
除开始读取消息外，此命令还可以在消息完成后，将最多 9 个（这是数据计数）寄存器数据从模块传输到控制器。返回的数据从命令字 1 中提供的起始寄存器编号获得。

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。

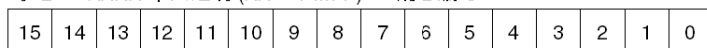
命令结构



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

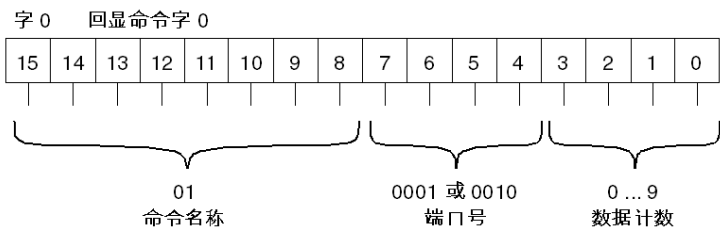


字 2 XXXX 十六进制 (XX = 1...FF) 消息编号



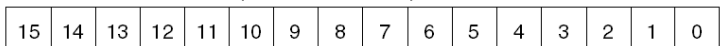
注意： 对于命令 1，不使用命令字 1 至 11。

响应结构

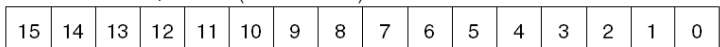


注: 位 15 是状态字有效位。

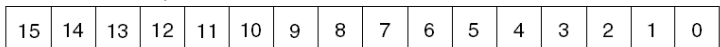
字 1 XXXX 十六进制 (XXXX = 0 ...3FFF) 回显起始寄存器编号



字 2 XXXX 十六进制 (XX = 1 ...FF) 回显消息编号

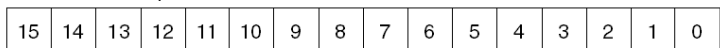


字 3 XXXX 十六进制 数据字 1



•
•
•

字 11 XXXX 十六进制 模块状态或数据字 9



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，也就是将 ASCII 字符放入串行口的输出/传输缓冲区。

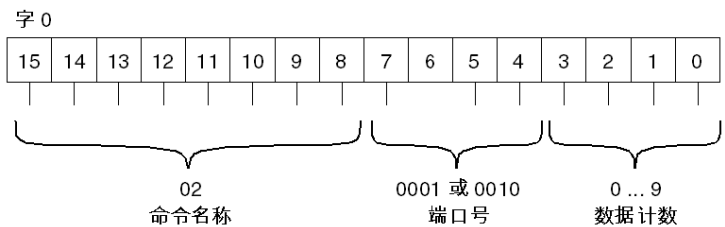
要开始写入消息，模块需要了解以下信息：

- 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

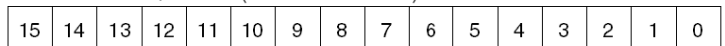
除开始写入消息外，此命令还可以在消息开始写入前将最多 9 个（这是数据计数）寄存器数据从控制器传输到模块。发送的数据从命令字 1 中提供的起始寄存器编号开始存储。

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。

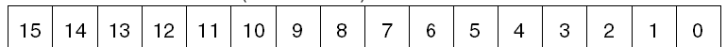
命令结构



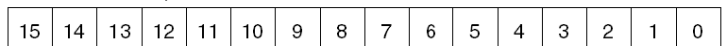
字 1 XXXX 十六进制 (XXXX = 0...3FFF) 起始寄存器编号



字 2 XXXX 十六进制 (XX = 1...FF) 消息编号

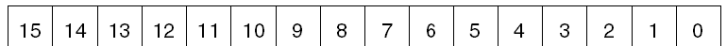


字 3 XXXX 十六进制 数据字 1

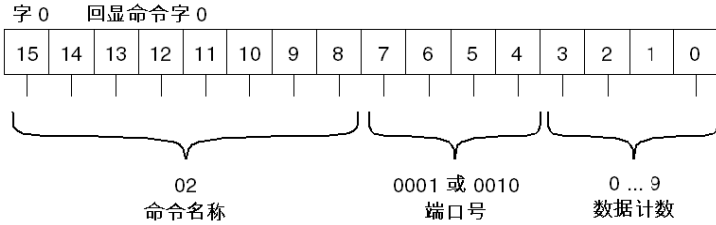


•
•
•

字 11 XXXX 十六进制 数据字 9



响应结构



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 (从模块到控制器)

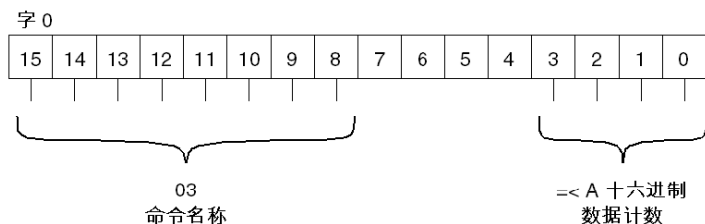
概述

GET DATA 命令从命令字 1 中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。命令字 0 中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。

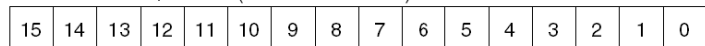
注意： 如果报告了错误状态（并且不是命令语法错误），且命令请求 10 个寄存器数据，则模块将仅返回 9 个字数数据并使用响应字 11 作为模块状态。如果响应字 11 是模块状态，则将设置状态字数据位。

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

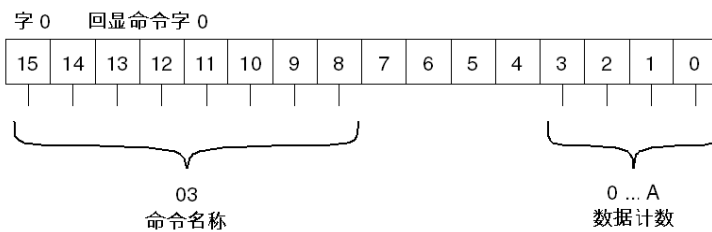


字 1 XXXX 十六进制 (XXXX = 0 ...3FFF) 起始寄存器编号

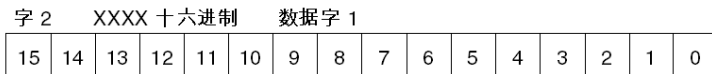
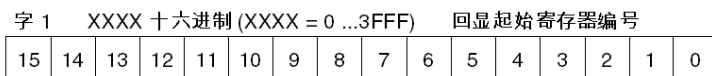


注意： 对于命令 3，不使用命令字 2 至 11。

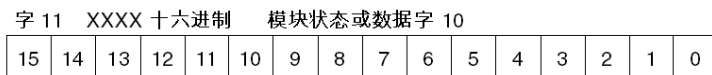
响应结构



注:位 15 是状态字有效位。



•
•
•



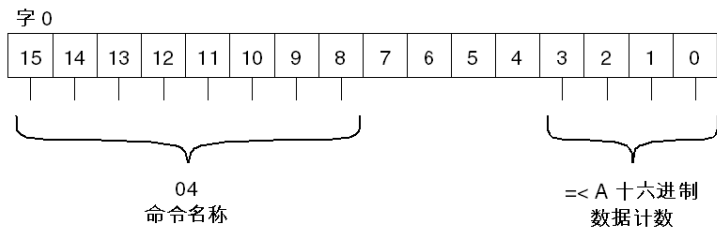
命令 4 - PUT DATA (从控制器到模块)

概述

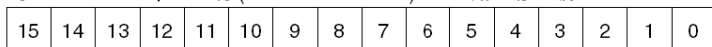
PUT DATA 命令从命令字 1 中提供的起始寄存器编号开始，将最多 10 个字/寄存器的数据写入模块。数据在命令 2 至 11 中发送。

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。

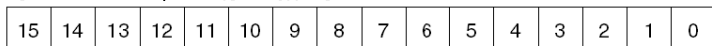
命令结构



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

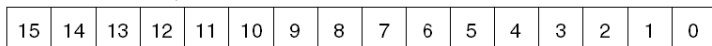


字 2 XXXX 十六进制 数据字 1

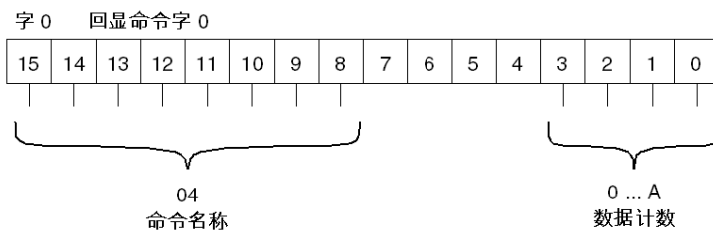


•
•
•

字 11 XXXX 十六进制 数据字 10



响应结构



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

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 4，命令字 2 至 10 返回 0。

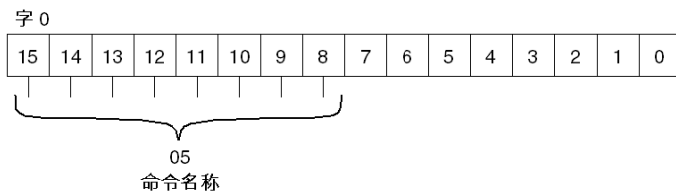
命令 5 - GET TOD (时间)

概述

GET TOD 命令读取模块的 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完全相同。

此命令持续执行，无需更改任何命令字。

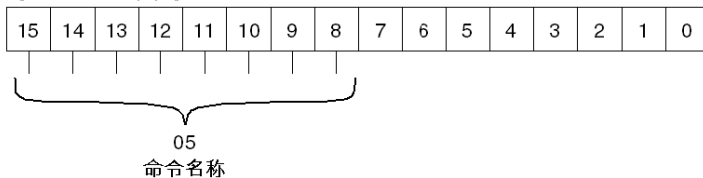
命令结构



注意：对于命令 5，不使用命令字 1 至 11。

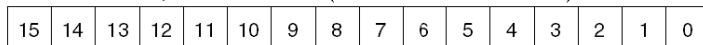
响应结构

字 0 回显命令字 0

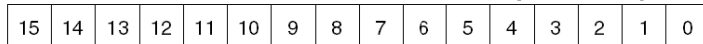


注: 位 15 是状态字有效位。

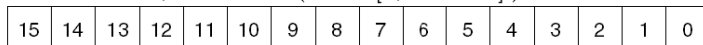
字 1 XXXX 十六进制 星期几(1 = 星期日 ...7 = 星期六)



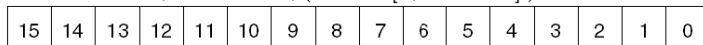
字 2 XXXX 十六进制 月(一月 = 1 ... 十二月 = C [十进制的 12])



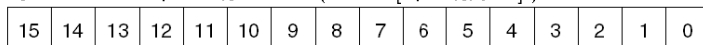
字 3 XXXX 十六进制 日(1 ...1F [十进制的 31])



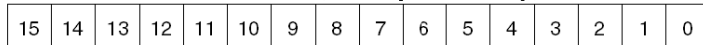
字 4 XXXX 十六进制 年(00 ...63 [十进制的 99])



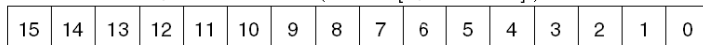
字 5 XXXX 十六进制 小时(0 ...17 [十进制的 23])



字 6 XXXX 十六进制 分钟(0 ...3B [十进制的 59])



字 7 XXXX 十六进制 秒钟(0 ...3B [十进制的 59])

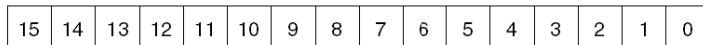


•

•

•

字 11 XXXX 十六进制 模块状态

**注意：**对于命令 5，命令字 8 至 10 返回 0。

命令 6 - SET TOD (时间)

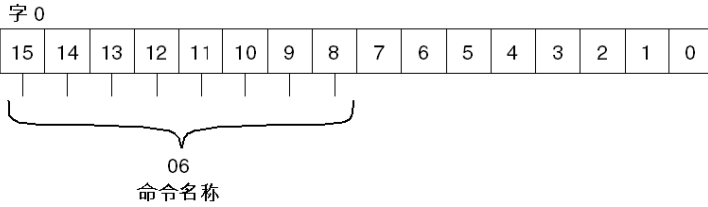
概述

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完全相同。

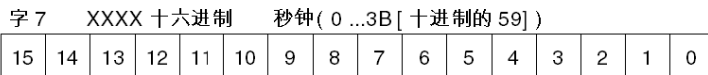
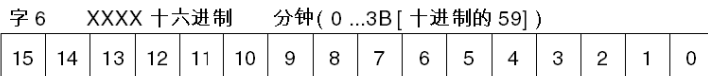
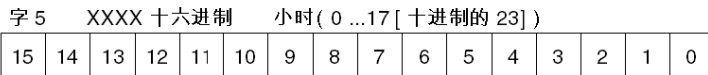
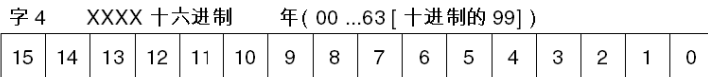
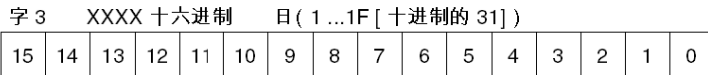
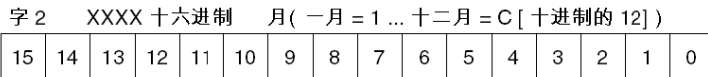
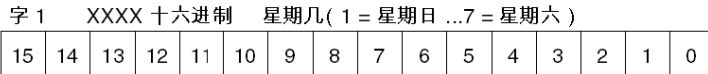
注意：为同步模块和 PLC 的 TOD 时钟，请对命令字 1 至 7 执行 PLC 的 7 个时间/日期寄存器的块移动，并将命令字 0 设置为 0600 (十六进制)。

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

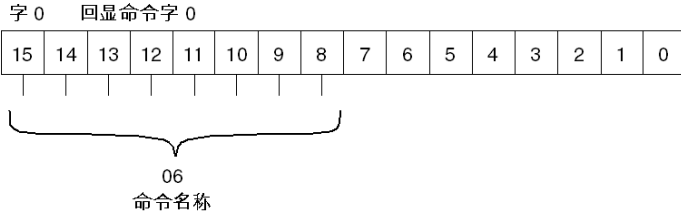


注: 位 15 是状态字有效位。

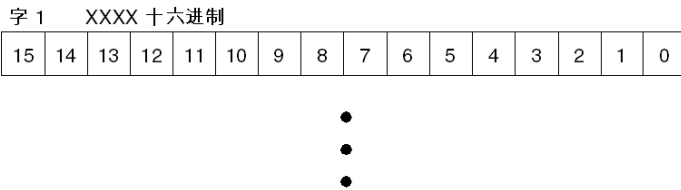


注意： 对于命令 6，不使用命令字 8 至 11。

响应结构



注:位 15 是状态字有效位。



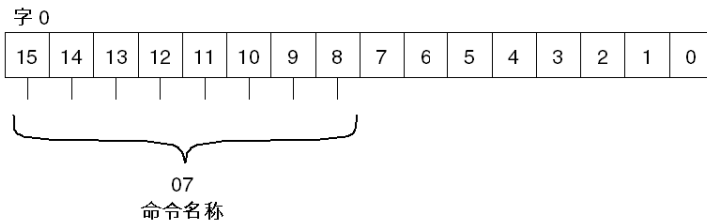
注意：对于命令 6，命令字 1 至 10 返回 0。

命令 7 - SET MEMORY REGISTERS

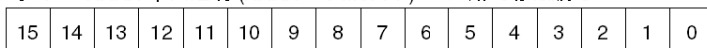
概述

SET MEMORY REGISTERS 命令将模块寄存器设置为命令字 3 中提供的值。将由起始寄存器编号和结束寄存器编号来指定寄存器范围。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。

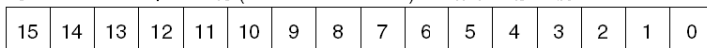
命令结构



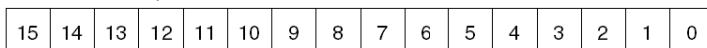
字 1 XXXX 十六进制 (XXXX = 0 ...3FFF) 起始寄存器编号



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



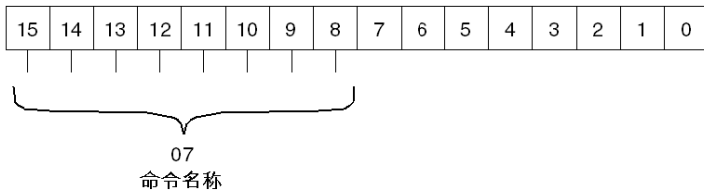
字 3 XXXX 十六进制 寄存器中设置的值



注意： 对于命令 7，不使用命令字 4 至 11。

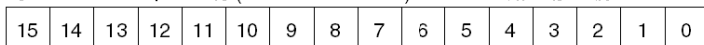
响应结构

字 0 回显命令字 0

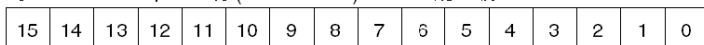


注:位 15 是状态字有效位。

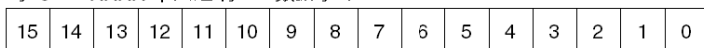
字 1 XXXX 十六进制 (XXXX = 0 ...3FFF) 回显起始寄存器编号



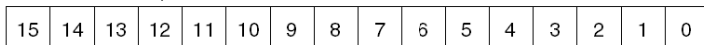
字 2 XXXX 十六进制 (XX = 1 ...FF) 回显消息编号



字 3 XXXX 十六进制 数据字 1

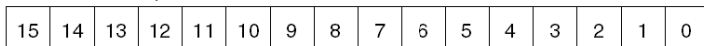


字 4 XXXX 十六进制



•
•
•

字 11 XXXX 十六进制 模块状态

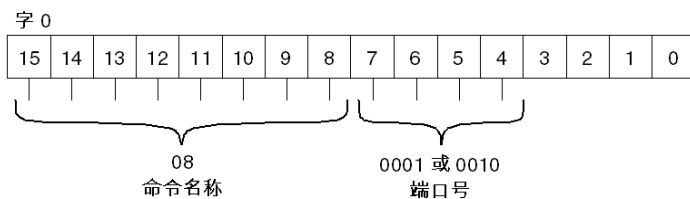
**注意：**对于命令 7，命令字 1 至 10 返回 0。

命令 8 - FLUSH BUFFER

概述

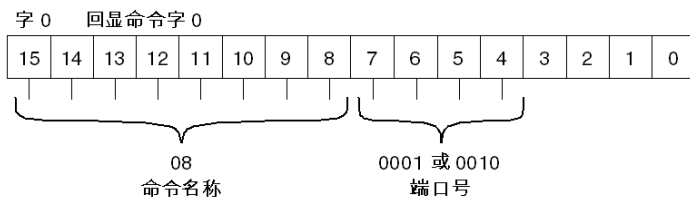
FLUSH BUFFER 命令为命令字中提供的串行口刷新输入缓冲区。此命令不影响输出缓冲区。

命令结构



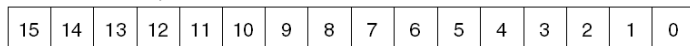
注意：对于命令 8，不使用命令字 1 至 11。

响应结构



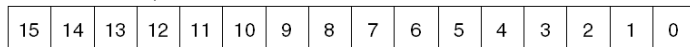
注: 位 15 是状态字有效位。

字 1 XXXX 十六进制



•
•
•

字 11 XXXX 十六进制 模块状态



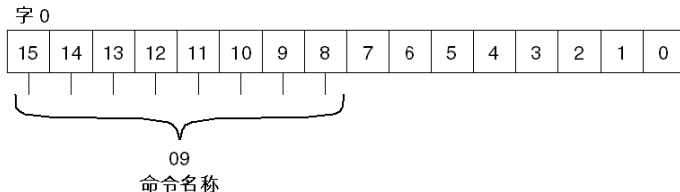
注意：对于命令 8，命令字 3 至 10 返回 0。

命令 9 - ABORT

概述

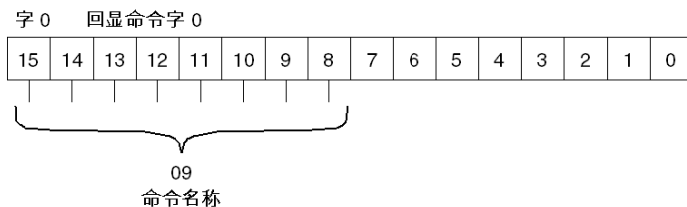
ABORT 命令中止正在运行的 READ ASCII MESSAGE 命令或 WRITE ASCII MESSAGE 命令，模块不再处于忙碌状态。此命令不影响模块的串行口缓冲区，仅影响当前运行的消息。

命令结构



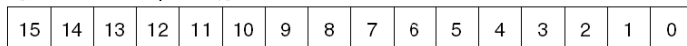
注意：对于命令 9，不使用命令字 1 至 11。

响应结构



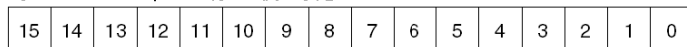
注：位 15 是状态字有效位。

字 1 XXXX 十六进制



•
•
•

字 11 XXXX 十六进制 模块状态



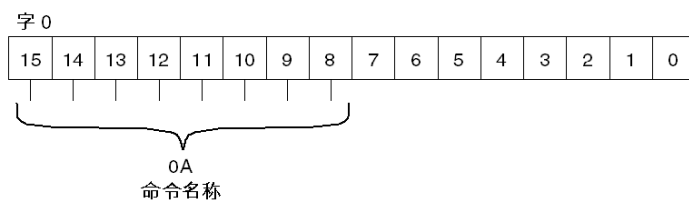
注意：对于命令 9，命令字 3 至 10 返回 0。

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。

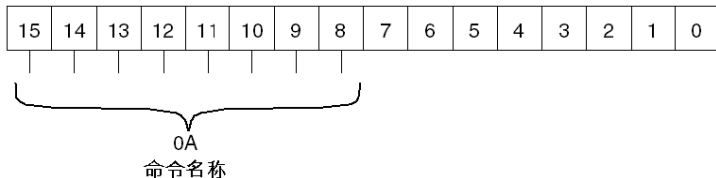
命令结构



注意：对于命令 A，不使用命令字 1 至 11。

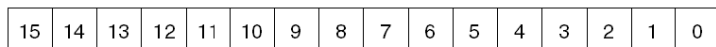
响应结构

字 0 回显命令字 0

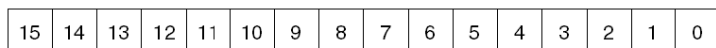


注: 位 15 是状态字有效位。

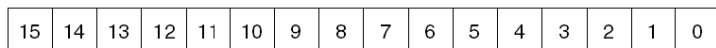
字 1 端口 1 缓冲区状态



字 2 端口 2 缓冲区状态

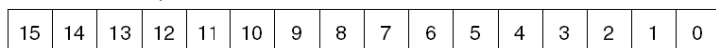


字 3 XXXX 十六进制



●
●
●

字 11 XXXX 十六进制 模块状态



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

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
----	----	----	----	----	----	---	---	---	---	---	---	---	---	---	---

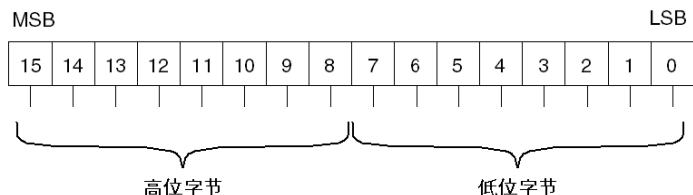
注意：命令字 1 至 10 返回 0。

模块状态字 (字 11)

概述

设置响应结构中字 0 的位 15 时，模块状态字 (响应结构中的字 11) 将包含有效的模块状态信息。此位的状态可用于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



注意： 在正常操作过程中，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	0	1	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

读取超出有效寄存器范围

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，则仅对于有效寄存器范围内的寄存器读取/写入数据。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）

用户逻辑命令		用户逻辑响应	
寄存器	内容	寄存器	内容
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)

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	页
A	字符集	81
B	ESI 062 10 简介	85

附录 A

字符集

ASCII 字符集

非打印 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	负确认
22	26	16	SYN	同步闲置
23	27	17	ETB	传输块结束
24	30	18	CAN	取消

十进制	八进制	十六进制	字符	字符控制
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	
103	147	67	g	125	175	7D	}
104	150	68	h	126	176	7E	~
105	151	69	i	127	177	7F	

附录 B

ESI 062 10 简介

简介

本章概述 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信息。

本章包含了哪些内容？

本章包含了以下主题：

主题	页
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 ASCII Basic 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。生成复杂的设备协议。	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*，第 42 页。

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 命令字*，第 49 页。

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*，第 36 页。

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。

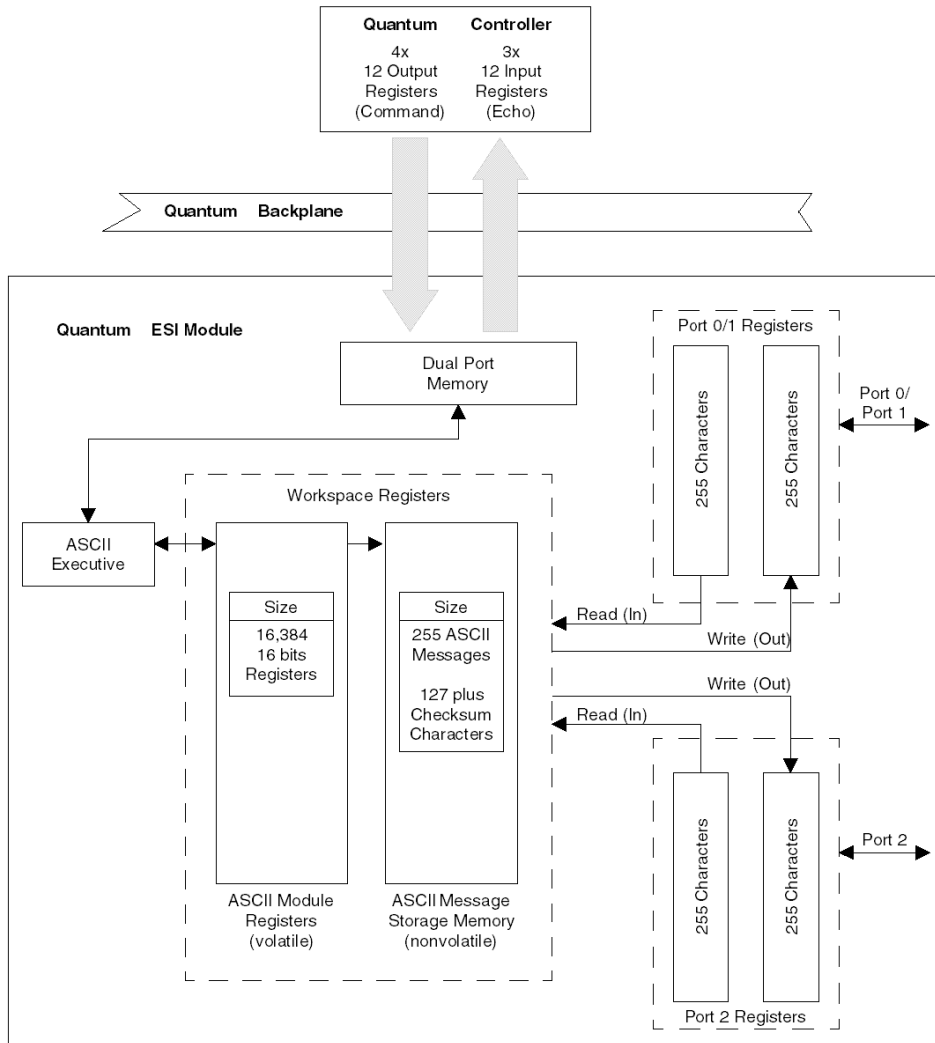
ASCII 消息存储

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





140ESI06210, *11, 85*
ASCII 字符集, *81*
ESI 命令
 ABORT, *70*
 FLUSH BUFFER, *69*
 GET BUFFER STATUS, *71*
 GET DATA, *58*
 GET TOD, *62*
 NO OPERATION, *52*
 PUT DATA, *60*
 READ ASCII MESSAGE, *53*
 SET MEMORY REGISTERS, *67*
 SET TOD, *64*
 WRITE ASCII MESSAGE, *55*
NO OPERATION, *52*
中止读/写消息, *70*
从模块获取数据, *58*
写入 ASCII 消息, *55*
刷新输入缓冲区, *69*
命令, *47*
命令行编辑器, *41*
实时时钟, *62, 64*
寻址
 平面, *21, 22*
 拓扑, *21*
将数据写入模块, *60*
崩溃代码, *14*
消息格式, *30*
离散量 I/O 的位序, *21*
设置存储器寄存器, *67*
设置模块的实时时钟, *64*
读取 ASCII 消息, *53*
读取模块的实时时钟, *62*
读取输入缓冲区中的字符, *71*
配置 ASCII 接口模块, *27*

