西门子S7-400模块控制器CPU414-2西门子中国一级总代理 西门子PLC代理商

产品名称	西门子S7-400模块控制器CPU414-2西门子中国一级总代理 西门子PLC代理商
公司名称	浔之漫智控技术（上海）有限公司
价格	888.00/台
规格参数	西门子:西门子代理商 西门子CPU:西门子plc 德国:全新原装
公司地址	上海市松江区石湖荡镇塔汇路755弄29号1幢一层A区213室
联系电话	195****8569 195****8569

产品详情

西门子S7-400模块控制器CPU414-2西门子中国一级总代理 西门子PLC代理商

西门子S7-400模块控制器CPU414-2西门子中国一级总代理 西门子PLC代理商

西门子S7-400模块控制器CPU414-2西门子中国一级总代理 西门子PLC代理商

西门子S7-400模块控制器CPU414-2西门子中国一级总代理 西门子PLC代理商

西门子PLC工作原理以及内部构造介绍

西门子PLC工作原理以及内部构造介绍

西门子plc是一种专业应用于企业的计算机，全名为可编程控制器。 在西门子PLC投入运行时，其工作过程一般分为三个阶段，即输入采样、用户程序执行和输出刷新三个阶段。完成上述三个阶段称作一个扫描周期。在整个运行期间，西门子PLC的CPU以一定的扫描速度重复执行上述三个阶段。

中央处理器是西门子PLC正常工作的神经中枢，当PLC投入运行时，首先它以扫描的方式接收现场各输入

装置的状态和数据，并分别存入I/O映象区，然后从用户程序存储器中逐条读取用户程序，经过命令解释后按指令的规定执行逻辑或算数运算的结果送入I/O映象区或数据寄存器内。等所有的用户程序执行完毕之后，最后将I/O映象区的各输出状态或输出寄存器内的数据传送到相应的输出装置，如此循环运行，直到停止运行。

其次是存储器。存储器是存放系统软件的称之为系统程序存储器;存放应用程序的存储器则被我们成为是用户成粗存储器。

其三是电源。当PLC投入运行后，其工作过程一般分为三个阶段，即输入采样、用户程序执行和输出刷新三个阶段。完成上述三个阶段称作一个扫描周期。

工作原理

当PLC投入运行后，其工作过程一般分为三个阶段，即输入采样、用户程序执行和输出刷新三个阶段。完成上述三个阶段称作一个扫描周期。在整个运行期间，PLC的CPU以一定的扫描速度重复执行上述三个阶段。

输入采样

在输入采样阶段，PLC以扫描方式依次地读入所有输入状态和数据，并将它们存入I/O映象区中的相应得单元内。输入采样结束后，转入用户程序执行和输出刷新阶段。在这两个阶段中，即使输入状态和数据发生变化，I/O映象区中的相应单元的状态和数据也不会改变。因此，如果输入是脉冲信号，则该脉冲信号的宽度必须大于一个扫描周期，才能保证在任何情况下，该输入均能被读入。

用户程序执行

在用户程序执行阶段，PLC总是按由上而下的顺序依次地扫描用户程序(梯形图)。在扫描每一条梯形图时，又总是先扫描梯形图左边的由各触点构成的控制线路，并按先左后右、先上后下的顺序对由触点构成的控制线路进行逻辑运算，然后根据逻辑运算的结果，刷新该逻辑线圈在系统RAM存储区中对应位的状态;或者刷新该输出线圈在I/O映象区中对应位的状态;或者确定是否要执行该梯形图所规定的特殊功能指令。即，在用户程序执行过程中，只有输入点在I/O映象区内的状态和数据不会发生变化，而其他输出点和软设备在I/O映象区或系统RAM存储区内的状态和数据都有可能发生变化，而且排在上面的梯形图，

其程序执行结果会对排在下面的凡是用到这些线圈或数据的梯形图起作用;相反，排在下面的梯形图，其被刷新的逻辑线圈的状态或数据只能到下一个扫描周期才能对排在其上面的程序起作用。

输出刷新

当扫描用户程序结束后，PLC就进入输出刷新阶段。在此期间，CPU按照I/O映象区内对应的状态和数据刷新所有的输出锁存电路，再经输出电路驱动相应的外设。这时，才是PLC的真正输出。同样的若干条梯形图，其排列次序不同，执行的结果也不同。另外，采用扫描用户程序的运行结果与继电器控制装置的硬逻辑并行运行的结果有所区别。当然，如果扫描周期所占用的时间对整个运行来说可以忽略，那么二者之间就没有什么区别了。

西门子S7协议笔记

参考模型

TPKT（ISO Transport Service ontop of the TCP）TPKT是一种“ 封装 ”协议。它在自己的数据包中携带OSI数据包的数据有效载荷(负载)，然后将结果结构传递给TCP，然后将该数据包作为TCP / IP数据包处理。将数据传递给TPKT的OSI程序不会察觉它们的数据将通过TCP / IP传输，因为TPKT模拟OSI协议传输服务访问点（TSAP：Transport Service Access Poin）。

TPKT(Transport Service ontop of the TCP)：通过TCP的传输服务。介于TCP和COTP之间。属于传输服务类的协议，它为上层的COTP和下层TCP进行了过渡。功能为在COTP和TCP之间建立桥梁，其内容包含了上层协议数据包的长度。一般与COTP一起发送，当作Header段。我们常用的RDP协议（remote desktop protocol，windows的远程桌面协议）也是基于TPKT的，TPKT的默认TCP端口为102（RDP为3389），其实它本身为payload增加的数据并不多，主要就是以下几个：

version reserved length payload

version，1byte，表明版本信息

reserved，1byte，看到这个名字就知道是保留的了

length，2byte，包括payload和这三部分在内的总长度

COTP（Connection Oriented Transport Protocol/面向连接的传输协议），比较TCP与COTP两种协议，因为它们都是用于通过网络可靠地传输用户数据，基于数据流的与基于数据包的：COTP将数据包从一个用户传输到另一个用户，所以接收者将获得与发送者传输完全相同的数据边界。TCP将连续的数据流传输到接收器，因此TCP上的协议通常必须自己添加这样的边界（如TPKT协议）。

西门子通信场景

西门子设备使用多种不同现场总线协议，例如：MPI、Profibus、IE 、Profinet 等。Profinet用于将PLC连接到IO模块，而不是设备的管理协议。S7以太网通信协议，主要用于将PLC连接到(i)pc站(PG/PC - PLC通信)。

大多数情况下，西门子通信遵循传统的主从模式(master-slave)或者CS模式(client-server)。其中PC（master/client）将S7请求发送到现场设备（slave/server)。这些请求用于从设备查询或向设备发送数据或发出某些命令。当PCL作为通信主站时（master）有一些例外，通过FB14/FB15设备可以向其他设备发起GET和PUT请求。

S7 协议工作流程

client与server通过socket建立连接，过程是标准的TCP连接方式，这一步完成连接的建立

client发送COTP，请求连接PLC，报文中包含CR Connect Request和Destination TSAP，从而标识出CPU的机架号和槽号

PLC返回COTP，确认连接，报文中包含CC Connect Confirm，此时server已经明确client与哪个CPU进行通讯

client发送S7 Communication给server，报文中包含Setup communication，即通讯请求

server返回S7 Communication给client，报文的ROSCTR为ACK_DATA，有确认的意思，包含了对作业请求的回复

client与server发送交换数据的报文，仍以S7 Communication完成

备注：2-5的步骤中，2与3完成数据传输前连接的功能，4与5则完成连接之后的通讯请求，如果绕过通讯请求的建立，在有TCP时就进行数据交换，服务器一般会直接断连

2. S7 PDU

S7协议TCP/IP实现依赖于面向块的ISO传输服务,S7协议包含在TPKT和ISO-COTP协议中，允许PDU（协议数据单元）通过TCP承载。ISO overTCP通信定义在RFC1006中，ISO-COTP定义在RFC2126其是基于ISO 8073协议（RFC905）。该结构如下图。

S7协议是面向功能/命令的，这意味着传输由S7请求和适当的应答组成（极少数例外）。在连接建立期间协商并行传输的数量和PDU的最大长度。

S7 PDU由三个主要部分组成：

头(Header)：包含长度信息，PDU参考和消息类型常量

参数(Parameters)：内容和结构根据PDU的消息和功能类型而有很大差异

数据(Data)：该数据是一个可选字段来携带数据，例如存储器值，块代码，固件数据等。

2.1 头(Header)

头长度为10-12个字节，确认消息包含两个额外的错误代码字节。除此之外，标头格式在所有PDU中是一致的。

字段：

Protocol ID： [1b]协议常量，始终设置为0x32

Message Type：
[1b]消息的一般类型（有时称为ROSCTR类型），消息的其余部分在很大程度上取决于Message Type和功能代码。

0x01-Job Request：主站发送的请求（例如读/写存储器，读/写块，启动/停止设备，通信设置）

0x02-Ack：从站发送的简单确认没有数据字段（从未见过它由S300 / S400设备发送）

0x03-Ack-Data：带有可选数据字段的确认，包含对作业请求的回复

0x07-Userdata：原始协议的扩展，参数字段包含请求/响应id，（用于编程/调试，SZL读取，安全功能，时间设置，循环读取…）

Reserved： [2b]始终设置为0x0000（但可能忽略）

PDU reference： [2b]由主站生成，每次新传输递增，用于链接对其请求的响应，Little-Endian（注意：这是WinCC，Step7和其他西门子程序的行为，它可能是随机的生成后，PLC只将其复制到回复中）

Parameter Length： [2b]参数字段的长度，Big-Endian

Data Length： [2b]数据字段的长度，Big-Endian

(Error class)： [1b]仅存在于Ack-Data消息中，可能的错误常量查看协议常量

(Error code)： [1b]仅出现在Ack-Data消息中，可能的错误常量查看协议常量

2.2 parameter

Parameter.Function中的取值及其对应的功能：

Function与Code对应关系

Parameters Code Parameters Function

0x00 diagnostics

0x04 read

0x05 write

0x1a request_download

0x1b download_block

0x1c end_download

0x1d start_download

0x1e upload

0x1f end_upload

0x28 plc_control

0x29 plc_stop

0xf0 setup communication

parameter参数取决于Message Type和功能代码。这里分析仅针对Message Type为：

（1）0x01- Job Request：主站发送的请求（例如读/写存储器，读/写块，启动/停止设备，通信设置）

（2）0x03- Ack-Data：带有可选数据字段的确认，包含对作业请求的回复

Job和Ack Data消息的parameter都是以功能代码开头，其余字段的结构取决于此值。

2.2.1.设置通信[0xF0]

在可以交换任何其他消息之前，在每个会话开始时会发送该消息对（Job 和Ack Data）。它用于协商Ack队列的大小和最大PDU长度，双方都声明其支持的值。Ack队列的长度决定了可以在没有确认的情况下同时启动的并行作业的数量。PDU和队列长度字段都是大端。

参数结构如下图所示：

字段：

Function Code：功能代码，通信设置为0xf0

Reserverd：保留字段，默认为ox00

Max AmQ Caller ：Ack队列的大小（主叫）

Max AmQ Callee：Ack队列的大小（被叫）

PDU length： PDU长度。

2.2.1.1 S7认证和保护

在配置CPU期间，可以设置三种保护模式。

没有保护：正如人们所期望的那样，不需要身份验证。

写保护：对于某些数据写入和配置更改操作，需要进行身份验证。

读/写保护：就像前一个一样，但某些读操作也需要验证。

必须注意的是，即使启用了读/写保护，也会允许某些操作，例如读取SZL列表或读取和写入标记区域。其他操作（如读取或写入对象/功能/数据块）应返回权限错误。

有两个与CPU关联的保护级别，他们是：分配保护级别和实际保护级别。

分配保护级别是配置期间设置的保护级别；

而实际的保护级别是适用于通信会话的当前保护级别。

在正常操作期间，在通信设置之后，客户端通过查询SZL读取（SZL ID：0x0132
SZL索引：0x0004）得到读写权限，查询实际保护级别和分配保护级别。

如果需要认证的密码以用户数据消息被发送到设备，这会降低有效的保护水平。

任何人认为这至少提供了一点点安全性之前，但它不是。密码是六个字节，几乎以明文形式发送（与常量进行异或并移位）。它是可重放的，可以强制执行。该协议还不提供完整性或机密性保护，可以进行消息注入和修改。S7安全性的一般经验法则是，如果您可以ping设备，则可以拥有它。

这里必须注意的是，S7-1200/1500系列设备使用略有不同的方法，保护级别处理稍有不同，发送的密码明显更长（实际上是密码的哈希），但它仍然是不变的，可重放。

2.2.2 读/写变量[0x04/0x05]

通过指定变量的存储区域，地址（偏移量）及其大小或类型来执行数据读取和写入操作。在进入协议细节之前，先简要介绍一下S7寻址模型。

如前所述，通过指定地址来访问变量，该地址由三个主要属性组成。

存储区域：

Merker： [M]任意标记变量或标志寄存器驻留在这里。

Data Block： [DB]
DB区域是存储设备不同功能所需数据的最常见位置，这些数据块编号为地址的一部分。

Input： [I]数字和模拟输入模块值，映射到存储器。

Output： [Q]类似的存储器映射输出。

Counter： PLC程序使用的不同计数器的[C]值。

Timer： PLC程序使用的不同定时器的[T]值。

还有其他不太常见的存储区域（例如本地数据[L]和外设访问[P]等）。

变量的类型决定了它的长度以及它应该如何解释。一些例子是：

BIT：[X]一位。

WORD：两个字节宽的无符号整数。

DINT：四字节宽的有符号整数。

REAL：四个字节宽的IEEE浮点数。

COUNTER：PLC程序计数器使用的计数器类型。

示例：

变量的示例地址是DB123X 2.1，它访问数据块#123的第三个字节的第二个位。

在这个简短的介绍之后，回到协议的变量读/写的实现。S7协议支持使用不同的寻址模式在单个消息中查询多个变量读/写。主要有三种模式：

any-type：这是默认的寻址模式，用于查询任意变量。为每个寻址变量指定所有三个参数（区域，地址，类型）。

db-type：这是专门用于解决DB区域变量的特殊模式，它比any-type地址更紧凑。

symbolic-addressing： S7-1200/1500系列设备使用此模式，允许使用预定义的符号名称寻址某些变量。此处不会详细介绍此模式。

对于每种寻址模式，Parameters头的结构方式相同：

Function Code： [1b]读取的常量值0x04或写入作业和回复的0x05。

Item Count： [1b] Request Item结构的数量。

Request Item：此结构用于解决实际变量，其长度和字段取决于所使用的寻址类型。这些项目仅存在于作业请求中，无论寻址模式是什么，还是读取或写入请求，都会从相应的Ack数据中发出。

S7 PDU 的Data 部分根据消息的类型（read/write）和方向（Job/Ack Data）而变化：

Read Request：该Data 部分是空的。

Read Response： Ack数据消息的Data 部分由Data Item结构组成，每个结构对应于原始请求中存在的每个Request Items。这些项包含读变量的实际值，格式取决于寻址模式。

Write Request：包含与读响应类似的 Data Items，一个用于 Parameter头中的每个Request Items。类似地，这些包含要在从设备上写入的变量值。

Write Response：Ack数据消息的数据部分只包含原始 Write Request中每个Request Items的一个字节错误代码。有关错误代码值，请参阅协议常量。

总而言之，Request Item 始终包含变量的描述，其中多个可以在作业请求中发送，而 Data Item包含所描述变量的实际值。所述 Data Item的结构必须开始于偶数字节，所以如果它们的长度是奇数且有以下 Data Item然后将它们填充以零字节。

剩下要讨论的是Request/Data
Item结构的格式。如前所述，它们依赖于所使用的寻址模式，因此将基于此介绍它们。

2.2.2.1 具有any-type任何类型寻址的项目结构

下图显示了Request和Data Item结构：

请求项的字段：

Specification Type：
[1b]该字段确定结构的主要类型，对于读/写消息，它总是具有值0x12，代表变量规范。

Length： [1b]此项目其余部分的长度。

Syntax ID： [1b]此字段确定寻址模式和项结构其余部分的格式。它具有任意类型寻址的常量值0x10 。

Variable Type： [1b]用于确定变量的类型和长度（使用常用的S7类型，如REAL，BIT，BYTE，WORD，D WORD，COUNTER等）。

Count： [2b]可以用单个项结构选择整个相似变量数组。这些变量必须具有相同的类型，并且必须在内存中连续，并且count字段确定此数组的大小。对于单变量读或写，它设置为1。

DB Number： [2b]数据库的地址，如果该区域未设置为DB，则忽略该数据库（参见下一个字段）。

Area： [1b]选择寻址变量的存储区域。见协议常量的内存区域的常数。

Address： [3b]包含所选存储区中寻址变量的偏移量。实质上，地址被转换为位偏移并在网络（大端）字节顺序中的3个字节上编码。实际上，由于地址空间小于5位，所以从不使用最重要的5位。作为一个例子，DBX40.3将是0x000143 40 * 8 + 3。

数据项字段

类似地，相关 Data Item的字段：

Error Code： [1b]操作的返回值，0xff信号成功。在“ 写入请求 ”消息中，此字段始终设置为零。

Variable Type and Count： [1b 2b]与 Request Item中的相同。

Data：此字段包含已寻址变量的实际值，其大小为len(variable) * count。

2.2.2.2 具有db-type寻址的项目结构

下图显示了Reques Itemt和Data Item结构

Request Item字段：

Specification Type： [1b]与任何类型寻址相同。

Length： [1b]此项目其余部分的长度。

Syntax ID： [1b]确定寻址模式，db-type的常量值为0xb0 。

Number of Subitems： [1b] 以下子项的数量。

Subitem：

Size： [1b]指定从所选地址读取或写入的字节数。

DB Number： [2b]寻址变量所在的DB。

Address： [2b]将变量的字节偏移量放入给定的DB中。

Data Item的字段：

Error Code： [1b]操作的返回值，0xff信号成功。

Variable Type： [1b]始终设置为0x09（八位字符串）。

Length： [2b]剩余子响应数据的长度。

Subresponse：

Error Code： [1b]与Subitem请求关联的返回值。

Data：要读取或写入的实际数据，解释这需要相应的Subitem。

2.2.3 阻止/下载[0x1a-1f]

在西门子术语中，下载是主机将块数据发送到从机。上载刚好与此方向相反。在西门子设备上，程序代码和（大多数）程序数据以块的形式存储，这些块具有自己的头和编码格式，这里不再详细讨论。从协议的角度来看，它们是需要传输的二进制blob.

西门子设备认可七种不同类型的块：

OB：组织块，存储主程序。

(s)DB :（系统）数据块，存储PLC程序所需的数据。

(s)FC :(系统）功能，无状态功能（没有自己的内存），可以从其他程序调用。

(s)FB :(系统）功能块，有状态的功能，通常有关联的（S）DB。

这些块在up/download请求中使用特殊的ASCII文件名进行寻址。此文件名的结构如下：

File Identifier： [1 char]据我所知，它总是具有’_'的值。

Block Type： [2个字符]确定块类型。

Block Number： [5个字符]十进制格式的给定块的编号。

Destination File System： [1 char]此字段的值可以是“ A ”表示活动，“ P ”表示被动文件系统。复制到活动文件系统的块会立即链接，这意味着它们会在PLC执行恢复后立即生效。另一方面，需要首先激活复制到被动文件系统的块。

示例:

文件名是_0800001P，用于将OB 1复制到被动文件系统或从被动文件系统复制OB 1。

** 这里快速介绍一下块编码和内容保护。有两种措施可以保护程序和数据的内容，并允许程序库的分发。第一个称为专有技术保护，如果设置，则防止STEP7或TIA显示块的实际内容。不幸的是，这很容易绕过，因为它只是在块的标题中设置了两位并且可以很容易地被清除。另一个保护措施是块“ 加密 ”，实际上它只是一个线性变换的混淆（逐字节xoring和旋转常数），再次应该是微不足道的绕过。因此，不要依赖这些“ 安全 ”机制来保护您的专有技术。否则，数据块包含存储器的原始初始化图像。 **

上传和下载块涉及3-3种不同类型的消息对。下面列出了相关的功能代码：

请求下载 - 0x1a

下载块 - 0x1b

下载结束 - 0x1c

开始上传 - 0x1d

上传块 - 0x1e

结束上传 - 0x1f

这些消息的结构非常简单，但是消息序列（特别是下载）需要一些解释。