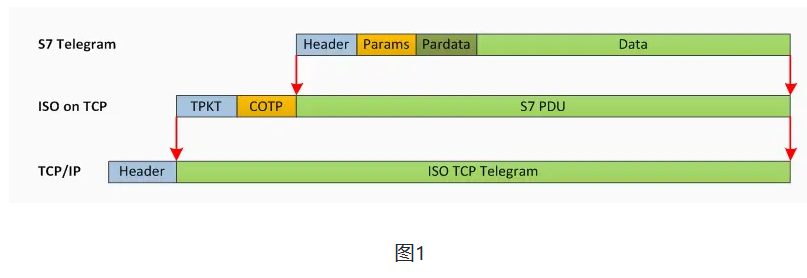
# **西门子S7协议介绍**

## 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)。

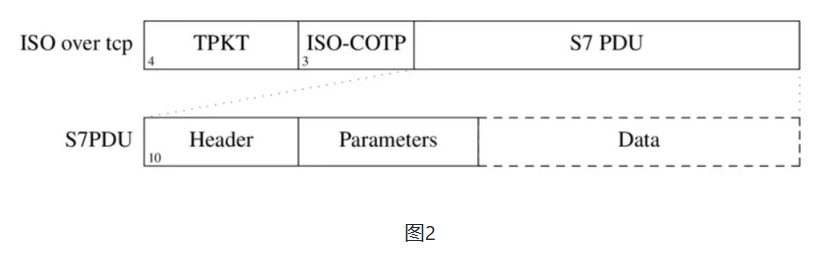
大多数情况下，西门子通信遵循传统的主从模式(****master-slave****)或者CS模式(****client-server**** )。其中PC（****master****/****client****）将S7请求发送到现场设备（****slave****/****server****)。这些请求用于从设备查询或向设备发送数据或发出某些命令。当PCL作为通信主站时（master）有一些例外，通过FB14/FB15设备可以向其他设备发起****GET****和****PUT****请求。

在S400系列中，实现了所谓的****循环数据I/O****功能，这类似于传统的****发布者 - 订阅****模型。PC可以订阅某些事件，而不是PLC ****定期推送****所请求的数据送到网络。还有一个****合作伙伴（Partner ）****或****点对点****模型，当一个活动的合作伙伴请求连接并调用****阻塞发送（Block Send），****与此同时被动合作伙伴调用****阻塞接收（Block Receive）****方法。



## **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：

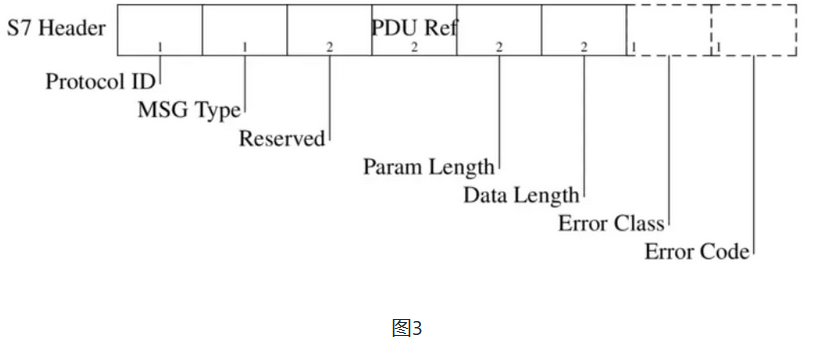
****头(Header)：****包含长度信息，PDU参考和消息类型常量

****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](https://maling.blog.csdn.net/article/details/133135043" \o "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](https://maling.blog.csdn.net/article/details/133135043" \o "协议常量)

#### ****2.2 parameter****

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）

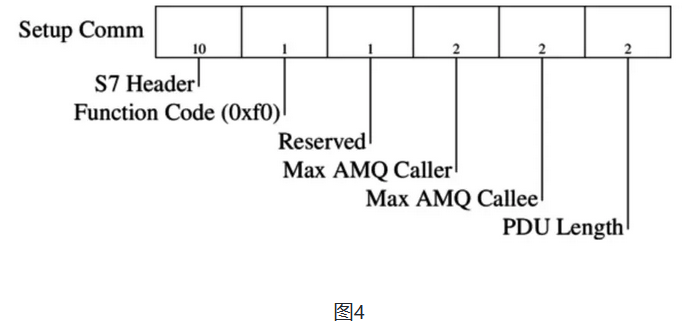
      （2） ****0x03- Ack-Data：****带有可选数据字段的确认，包含对作业请求的回复

*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。

该****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。

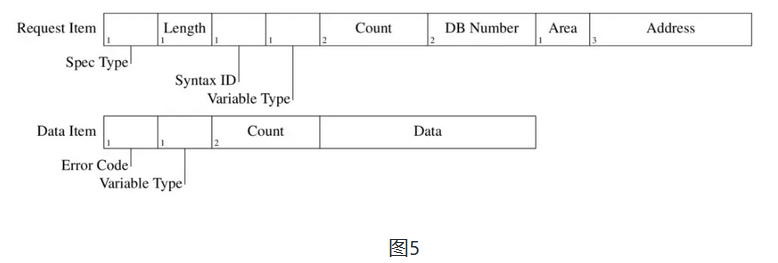
****Write Response：Ack数据****消息的数据部分只包含原始 ****Write Request****中每个****Request Items****的一个字节错误代码。有关错误代码值，请参阅[协议常量](https://maling.blog.csdn.net/article/details/133135043" \o "协议常量)。

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，DWORD，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](https://maling.blog.csdn.net/article/details/133135043" \o "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：

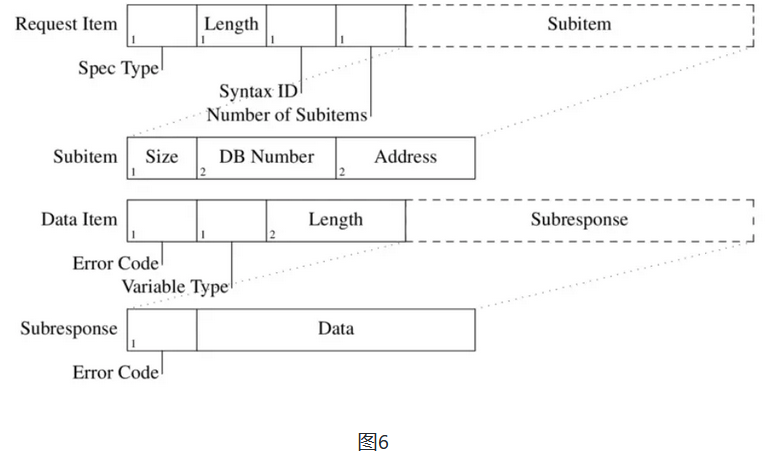
****Error Code：**** [1b]操作的返回值，****0xff信号成功****。在“ ****写入请求”****消息中，此字段始终设置为零。

****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，设置（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

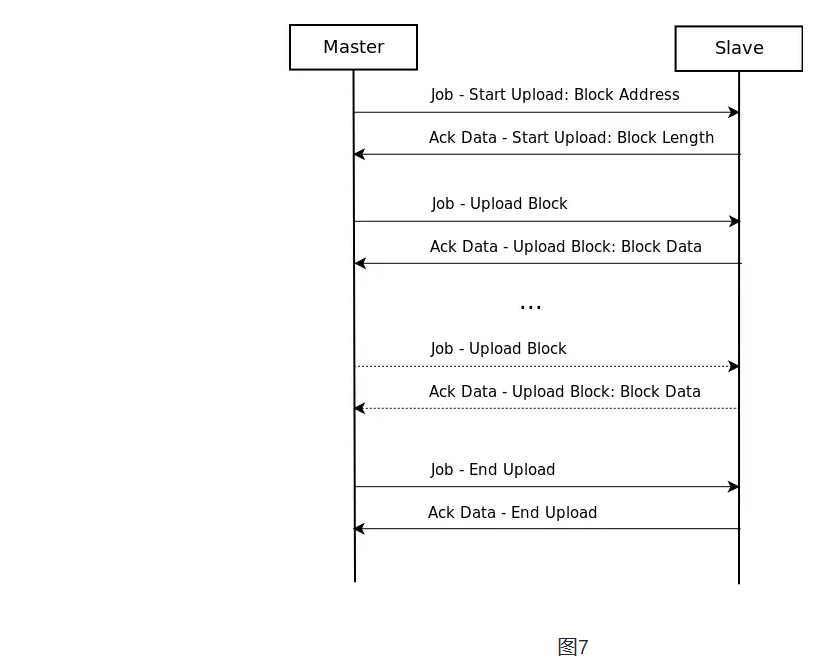
       上传块 - 0x1e

       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。

###### ****2.2.3.1 上传块(0x1a)****

上传块序列非常直观，如下所示：



在****Ack Data - Start Upload****消息中，从设备告知块的长度，然后主设备继续发送****Job - Upload Block****消息，直到接收到所有字节。最后，它使用****Job - End Upload****消息关闭上传序列。块的实际数据由从****Ack Data - Upload Block****消息中发送。

****Job - Start Upload****参数头：

****Function Code：**** [1b] 0x1d用于*开始上传*。

****Function Status：**** [1b]仅用于*上传*消息，如果要发送更多数据，则设置为0x01。

****Unknown：**** [2b]总是0x0000。

****Session ID：**** [4b]与每个上传序列相关联的唯一ID，它在*Ack Data - Start Upload*消息中设置。

****Filename Length：**** [1b]以下文件名的长度。

****Filename：****标识上面引入的块的文件名。

****Ack Data - Start Upload****参数标题：

****Function Code：**** [1b] 0x1d用于*开始上传*。

****Function Status：**** [1b]与上述相同。

****Unknown：**** [2b]总是0x0100。

****Session ID：**** [4b] 此处设置*会话ID*，连续消息使用相同的值。

****Length String Length：**** [1b]以下*块长度字符串的长度*。

****Length String：****编码为ASCII C字符串的块的十进制长度（不要问我为什么......）。

****作业 - 上传****参数头：

包含****功能代码****（0x1e），****功能状态****，****未知****（0x0000）和****会话ID****字段，如上所述。

****确认数据 - 上传****参数和数据部分：

****Function Code：**** [1b] 0x1e用于*上传*。

****Function Status：**** [1b]如果要发送更多数据，则设置为0x01。

****Data**** 部分：

****Length：**** [2b] *块数据*的长度。

****Unknown：**** [2b]总是0x00fb。

****Block Data：****上传数据块的一部分。

****Job - End Upload****参数头：

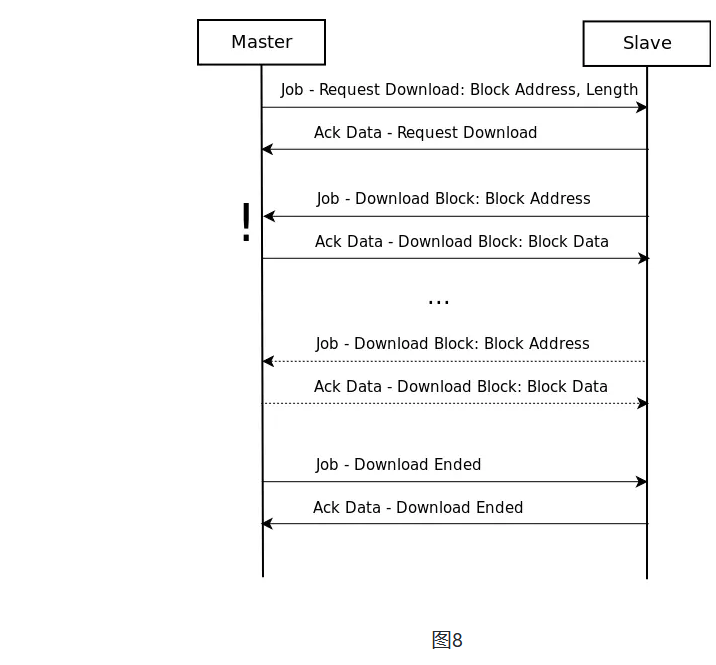
        包含****功能代码****（0x1f），****功能状态****，****未知****（0x0000）和****会话ID****字段，如上所述。

****Ack Data - End Upload****参数头：

       只包含****功能代码****（0x1f）

###### ****2.2.3.2 下载块(0x1b)****

上传和下载之间的关键区别在于，在下载过程中，通信方向发生变化，从站成为主站（很好）。在初始*请求下载*交换后，从站发送*作业*消息，主站回复*Ack数据*，这是“仅从站回复”规则的唯一例外。发送*完* 所有字节后，主服务器（原始服务器）发送*下载结束作业*以关闭下载会话。请参见下面的序列图。



实际消息的结构与上传消息非常相似，所以这里只介绍差异。

该 ****Job - Request Download**** 消息中包含了两个附加字段，该****Block Length****下载的块和****Payload Length****块（不块头的长度）。这两个字段都是编码为ASCII字符串的十进制数字。响应 ****Ack Data - Request Download**** 仅包含*功能代码*。

另一个显着的区别是，虽然*会话ID*字段存在但不使用（仍为0x00000000）而是在每个 ****Job - Download Block****传输*文件名*。其余消息的结构与前面讨论的相同。

##### ****2.2.4 PLC控制[0x28]****

PLC控制消息用于在从设备上执行修改其执行/存储状态的不同例程。这些命令用于启动或停止PLC控制程序的执行，激活或删除设备上的程序块或将其配置保存到永久存储器。这些消息的结构非常简单，它们将在不讨论确切细节的情况下进行解释。

****Job - PLC Control**** 消息由两个主要部分组成，被调用的方法和它的参数（也被编码为ASCII字符串）的ASCII名字。方法名称的结构与块传输部分中引入的文件名的方式类似。参数取决于方法类型，可以将它们视为参数。该 ****Ack Data****消息仅包含*PLC控制*功能码。

一些示例函数名称及其相关参数：

           \_INSE：激活设备上下载的块，参数是块的名称（例如OB1）。

           \_DELE：从设备的文件系统中删除一个块，该参数也是块的名称。

           P\_PROGRAM：设置设备的运行状态（启动，停止，mem重置）。它在没有参数的情况下发送以启动设备，但是停止plc程序使用不同的功能代码（参见下一节）。

           \_GARB：压缩PLC内存。

           \_MODU：将ram复制到rom，该参数包含文件系统标识符（A/E/P）。

##### ****2.2.5 PLC停止[0x29]****

所述****PLC Stop****消息是基本相同的****PLC Control**** 消息。唯一的区别是消息中没有参数，并且例程部分始终设置为P\_PROGRAM。不知道为什么它有单独的类型而不是使用参数来确定它是一个开始或停止消息。