

# 西门子S7-300授权总经销商 6ES7355-2CH00-0AE0 闭环温度控制模块

产品名称	西门子S7-300授权总经销商 6ES7355-2CH00-0AE0 闭环温度控制模块
公司名称	浔之漫智控技术(上海)有限公司
价格	.00/件
规格参数	西门子:现货 S7-300:正品 德国:全新
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	15801997124 15801997124

## 产品详情

西门子S7-300授权总经销商 6ES7355-2CH00-0AE0 闭环温度控制模块

[6ES7355-2CH00-0AE0](#)

SIMATIC S7-300, 温度 步进电机/伺服电机的 FM 355-2 C, 4 通道, 连续, 4 模拟输入+8 数字输入+4 模拟输出 包括多种语言项目组态包, 手册和入门指南 (德语、en, it)保存在 CD 只读光盘上

下图显示了使用后续响应消息帧获取数据时的传输顺序。6' (%'6'' (%'LUN LU LU LU LU LU \$ LU LU LU LU aCl#0 ^ \* I\* Q,\ \ %7P> ± 1/P &%4#(] 64(O ü 8 I\* O ü 8 I\* 3Yu(Q,\ \ ^ \* I\* Q,\ \ %7P > ± 1/P fBJP?% LU LU LUN O ü 8 I\* O ü 8 I\* Q,\ \ a0# > ± / \* ab ó a0#% > ± / \* FQ 6 ab ó 3Yu( %7P%/ %7P&%4 图 2-16 使用后续响应消息帧的后续 FETCH 消息帧的顺序 串行数据传输的基本原理 2.5 使用 RK 512 计算机连接的数据传输 2.5.3 半全双工模式 56 半全双工操作 半全双工模式的含义是：伙伴可随时发送命令和响应消息帧，只要其它伙伴当前未发送。命令和响应消息帧的\*大嵌套深度为“1”。因此，在用响应消息帧应答上一个命令消息帧之前，不能处理下一个命令消息帧。在特定情况下可以实现—如果两个伙伴在收到响应消息帧之前都要发送（传输）伙伴的 SEND 消息帧。例如，如果在收到响应消息帧之前伙伴的 SEND 消息帧被输入到 CP 341 的输出缓冲区中。在下图中，直到收到伙伴的 SEND 消息帧后才发送第一个 SEND 消息帧的后续响应消息帧。图 2-17 半全双工模式 CP 341 点到点通讯、安装和参数分配 设备手册, 04/2011, A5E01156022-03 串行数据传输的基本原理 使用 RK 512 计算机连接的数据传输 2.5 2.5.4 RK 512 CPU 请求 当提出 CPU 请求时 RK 512 中涉及的过程 下图显示了 CPU 提出请求时 RK 512 计算机连接中涉及的过程。图 2-18

CPU 提出请求时使用 RK 512 的数据传输流程图 CP 341 点到点通讯、安装和参数分配 设备手册, 04/2011, A5E01156022-03 57 串行数据传输的基本原理 2.5 使用 RK 512 计算机连接的数据传输 接收 FB 处的扩展错误显示 激活参数 “在 FB 中显示接收错误” 也可以在函数块 P\_RCV\_RK 的状态输出中显示接收出 错的消息帧。如果取消激活该参数, 则只在 CP 341 的诊断缓冲区中创建一个条目。RK 512 伙伴请求 58 下图显示了伙伴提出请求时 RK 512 计算机连接中涉及的程序。图 2-19 伙伴提出请求时使用 RK 512 的数据传输流程图 CP 341 点到点通讯、安装和参数分配 设备手册, 04/2011, A5E01156022-03 串行数据传输的基本原理 使用 RK 512 计算机连接的数据传输 2.5 接收 FB 处的扩展错误显示 激活参数 “在 FB 中显示接收错误” 也可以在函数块 P\_RCV\_RK 的状态输出中显示接收出 错的消息帧。如果取消激活该参数, 则只在 CP 341 的诊断缓冲区中创建一个条目。CP 341 点到点通讯、安装和参数分配 设备手册, 04/2011, A5E01156022-03 59 串行数据传输的基本原理 2.6 使用 ASCII 驱动程序的数据传输 2.6.2.6.1 2.6.2 使用 ASCII 驱动程序的数据传输 使用 ASCII 驱动程序的数据传输的原理 ASCII

驱动程序通过通讯处理器和通讯伙伴之间的点对点连接控制数据传输。该驱动程序包含物理层 (第 1 层)。在 S7 用户将完整的发送消息帧传输给通讯处理器的过程中, 消息帧的结构保持开放。对于接收方向, 必须组态消息的结束标准。发送消息帧的结构与接收消息帧的结构可能会不同。ASCII 驱动程序允许发送和接收任何结构的数据 (包括所有可打印的 ASCII 字符以及从 00 到 FFH [带有 8 个数据位字符帧] 的所有其它字符或从 00 到 7FH [带有 7 个数据位字符帧] 的所有其它字符)。使用 ASCII 驱动程序发送数据 使用 ASCII 驱动程序发送数据 对于发送, 您指定当 P\_SND\_RK 功能块作为 “LEN” 参数调用时要传输的用户数据字节数。如果您使用结束标准 “字符延迟时间结束”, 则 ASCII 驱动程序将在发送时在两个消息帧之间暂停。可以随时调用 FB P\_SND\_RK, 但是因为 \*后一个消息帧已发送, 所以只有分配的字符延迟时间过后, ASCII 驱动程序才开始输出。如果您使用 “文本结束字符” 结束标准, 则有三种选择: 发送并包含有文本结束字符 文本结束字符必须包含在要发送的数据中。发送的数据到文本结束字符为止并包含该字符, 即使 FB 中指定的数据长度更长。按 FB 中分配的长度发送数据 按 FB 中分配的长度发送数据。\*后一个字符必须是文本结束字符。即使要发送的数据不包括文本结束字符, 也将发送消息帧且无错误消息。按 FB 中分配的长度发送数据, 并自动附加一个或多个文本结束字符 按 FB 中分配的长度发送数据。自动附加文本结束字符, 即, 不得将文本结束字符包含在要发送的数据中。根据文本结束字符的个数, 超出 FB 中指定字符数的 1 或 2 个字符将发送给伙伴。CP 341 点到点通讯、安装和参数分配 60 设备手册, 04/2011, A5E01156022-03 串行数据传输的基本原理 使用 ASCII 驱动程序的数据传输 2.6 如果您使用结束标准 “固定消息帧长度”, 则发送方向上传输的数据量与 FB P\_SND\_RK 的 “LEN” 参数指定的一样。接收方向上传输的 (即, 接收 DB 中的) 数据量与在接收器端使用参数分配接口中的 “固定消息帧长度” 参数指定的一样。两个参数设置必须相同, 以确保正确进行数据通信。发送时将在两个消息帧之间暂停 (暂停时间等于字符延迟时间), 以使伙伴可以同步 (识别消息帧开始)。如果使用其它同步方法, 则可通过参数分配接口取消激活发送暂停。说明在分配 XON/XOFF 流控制后, 用户数据不得包含任何已分配的 XON 或 XOFF 字符。缺省设置为 DC1 = 11H (XON) 和 DC3 = 13H (XOFF)。发送数据 下图说明了一个发送操作。图 2-20 发送操作的流程图 CP 341 点到点通讯、安装和参数分配 设备手册, 04/2011, A5E01156022-03 61 串行数据传输的基本原理 2.6 使用 ASCII 驱动程序的数据传输 2.6.3 使用 ASCII 驱动程序接收数据 使用 ASCII 驱动程序接收数据 使用 ASCII 驱动程序进行数据传输时, 可以在三种不同的数据接收结束标准中进行选择。结束标准定义何时接收完整的消息帧。可能的结束标准如下: 字符延迟时间结束 消息帧没有固定长度, 也没有定义的文本结束字符; 消息帧的结束由行的暂停 (字符延迟时间结束) 定义。接收文本结束字符 消息帧的结束由一个或两个定义的文本结束字符标记。接收固定数目的字符 接收消息帧的长度始终相同。代码透明度 62 程序的代码透明度取决于所分配的结束标准和流控制的选择: 带有一个或两个文本结束字符 - 非代码透明 当结束标准为字符延迟时间或固定消息帧长度时 - 代码透明 当使用流控制 XON/XOFF 时, 无法实现代码透明操作。代码透明是指用户数据中可以包含任意字符组合, 而无需识别结束标准。CP 341 点到点通讯、安装和参数分配 设备手册, 04/2011, A5E01156022-03 串行数据传输的基本原理 使用

## ASCII 驱动程序的数据传输 2.6 结束标准“字符延迟时间结束”

接收数据时，字符延迟时间结束时识别为消息帧结束。CPU 接受所接收的数据。

在这种情况下，必须设置字符延迟时间，以确保其在两个连续的消息帧之间结束。但是该字符延迟时间应该足够长，这样，在消息帧中发生链接发送暂停时，不会错误地识别消息帧结束。

下图说明了使用结束标准“字符延迟时间结束”的接收操作。图 2-21

使用结束标准“字符延迟时间结束”接收的流程图 CP 341 点到点通讯、安装和参数分配 设备手册, 04/2011, A5E01156022-03 63 串行数据传输的基本原理 2.6 使用 ASCII 驱动程序的数据传输

结束标准“文本结束字符”接收数据时，在接收到分配的文本结束字符时识别为消息帧的结束。

CPU 接受所接收的、包括文本结束字符的数据。

如果在接收数据的同时字符延迟时间结束，则接收操作将结束。将发出一条出错消息并丢弃消息帧碎片。如果使用文本结束字符，则传输为非代码透明。

还要必须确保结束字符不含在用户的用户数据中。

如果接收到消息帧中的\*后一个字符不是文本结束字符，请注意以下情况。

文本结束字符在消息帧中的其它位置：所有字符(包括文本结束字符)都被写入接收 DB。

文本结束字符后的字符 – 被丢弃（如果字符延迟时间在消息帧结束时结束）。 –

与下一个消息帧合并（如果在字符延迟时间结束前收到新的消息帧）。

文本结束字符不包括在消息帧中：消息帧将： –

被丢弃（如果字符延迟时间在消息帧结束时结束）。 –

与下一个消息帧合并（如果在字符延迟时间结束前收到新的消息帧）