

SIEMENS S7-400 PLC 及 WinCC 实现高速数据采集

王前厚

[摘 要] 本文描述了 WinCC 端利用原始数据方式, S7-400 PLC 端使用数据块作为数据缓冲区, 实现高速数据采集的方法。数据采集周期达到 10ms。

[关 键 词] WinCC 数据采集 原始数据 缓冲区

在使用 PLC 机制实现控制的领域, 大多使用 WinCC 等组态工具实现对现场信号的采集功能并将数据存储到上位机数据库, 但由于受到通讯性能等原因的限制, 通常的方法所能实现的数据采集周期局限于秒级。这其实也是正常的, 因为对于一般意义的现场信号, 我们对它的测控所要求的数据周期要求并不高, 通常 1~2s 的采集周期, 就足以满足我们对现场信号的监控精度。

但在某些特定领域, 譬如快速反应系统, 其整个反应过程<5s, 要分析了解其在整个反应过程中的参数变化情况, 即便以 500ms 的采集周期, 数据采集量仍然太少, 分析结果仍然不够精确。而在此情况下, 上位机用于与控制器通讯的系统资源耗费已经非常高, 再加上数据处理、画面、数据库等任务, 在变量规模尚不算多的情况下, 上位计算机的负荷已经不堪重负。

对此类的数据处理, 通常的解决方案是使用带高速处理功能的数据采集板卡, 数据卡插在计算机插槽中, 通过板卡提供的驱动程序接口, 使用高级语言如 C++等编程实现。

但同时, 此解决方案的缺点又是非常明显的, 首先, 整个系统的规模受到非常大的限制。即便系统的其他控制对象均为普通信号, 但由于计算机的扩展插槽数量受限制, 导致能实现的点数往往仅限于最多不超过几十个。这样, 如果面对的是稍大规模的项目, 数据采集卡的模式就无能为力了。

因而我们仍旧尝试使用 PLC+WinCC 的模式下问题的解决方法。

首先, 我们回过头来, 分析研究数据采集卡的实现方式。在数据采集卡中, 根据数据采集的周期不同, 会分为普通采集卡和高速采集卡, 而在高速采集卡的性能指标中, 非常重要的就是它提供的缓存区的容量。许多采集卡会有如 1KB 或者 2KB 的数据缓冲区, 用于实现高速采集数据的缓存。采集卡从现场获得信号并进行 A/D 转换之后, 将数据存储于缓存区中, 而 PC 通过它的扩展 PCI 插槽从板卡获取数据。这个过程说到底其实也是一种通讯的过程。同时为了保证数据采集的连续性,

通常的方法是设定一个缓冲区半满的标志，或者说将整个缓冲区分成两部分，采集来的数据，首先堆栈到缓冲区 1 中，当缓冲区 1 存满，则发出一个半满标志到计算机，计算机收到这个半满信号，则启动读取程序，将缓冲区 1 内的数据读取到计算机内存中。当数据读取结束，清空缓冲区 1 内的数据，而在此期间，高速采集的数据，缓存在缓冲区 2 中。同样的道理，当缓冲区 2 满之后，同样发出半满信号，数据从缓冲区 2 转移到计算机。如此形成循环。

整个逻辑实现的基础就是，数据的集中通讯比分散通讯的效率高得多，在远远低于数据堆满一个缓冲区的时间内，已满数据能够有充分的时间转移到计算机中。

根据缓冲区大小的不同，在缓冲区比较小的情况下，为了保证数据半满之后，能够尽快将数据读出，以满足下一个周期的使用，半满信号往往需要较高的优先级，在计算机内以中断的方式，通知后续的处理过程。但如果缓冲区尺寸能够提供得比较大，留给主机与板卡通讯的查询时间能够足够长，那么即便是用普通 I/O 方式来处理半满信号，也同样可行。

基于上述思想同样的思路，我们可以应用到 WinCC + S7-400 PLC 的系统中来。

首先，WinCC 中如何才能支持连续数据的读取？我们注意到，在 WinCC 中可以建立原始数据类型的变量，所谓的原始数据类型，就是在 WinCC 中定义变量的时候，只需要指定数据在 PLC 中的开始地址和长度，而具体数据的排列方式定义，可以由用户自行定义。而同时在 WinCC 的趋势变量记录中支持 AR_SEND 功能块的原始数据类型，它分别设计了趋势变量记录可以接受的将近 10 种原始数据类型的结构，这些数据结构原理大同小异，无非所定义的变量数不同，有针对一个采集数的，也有针对多个采集数的；时间标签的记录方式不同，有不带时间标签的，也有带时间标签的，有的每条记录均带时间标签，而有的仅仅首条记录带有时间标签，后续的记录以相同的时间间隔平移后推。具体的类型描述，参见” [wincc V6.0 信息系统\使用 wincc\通讯\SIMATIC S7 PROTOCL SUITE\特殊功能\使用 S7 功能块“AR_SEND”进行数据交换\AR_SEND 变量-属性概述]

我们来做简单的性能测试，只对一个归档变量进行记录，同时，为了节省数据存储的空间和节省通讯耗时，我们选择其中的第二种数据类型，即“具有等间隔时间标志的过程值”。这种数据类型的描述如下：

DB 块地址	内 容			
0.0	报文类型=1			
2.0	年		月	
4.0	日		小时	
6.0	分钟		秒	
8.0	0.1s	0.01s	001s	工作日
10.0	周期=10			

12.0		
14.0	单元（类型）=1	单位（范围）=3
16.0	AR_ID 子编号=0	
18.0	过程数据：数据类型	
20.0	过程值数量=2000	
22.0	过程值 1	
24.0		
26.0	过程值 2	
28.0		
30.0	.	
32.0		
34.0	.	
36.0		
38.0	.	
40.0		
.....		

其中 22.0 之前的 22 个 BYTE 存放的报文头，之后为实质的数据序列。

报文头的各参数含义如下：

报头类型=1

报头类型定义了报头中所包含的信息的类型。

报头类型	时间标志	AR_ID 子编号
0	不带时间标志的报头	不带 AR_ID 子编号的报头
1	带时间标志的报头	不带 AR_ID 子编号的报头
8	不带时间标志的报头	带 AR_ID 子编号的报头
9	带时间标志的报头	带 AR_ID 子编号的报头

时间标志

时间标志包含 SIMATIC S7 BCD 格式的日期和时间。WinCC 不使用工作日条目。

周期=10

在周期中读取过程值。该参数是在单位(范围)处指定的时间单位的因子。数据长度：双字。

“周期” = 10; “单位(范围)” = 3 意思是: 过程值读取周期 = 10ms

单元(类型)=1

指定时间信息的类型并修改参数“过程值的数目”。

编号	含义
1	在相等的时间间隔内读取过程值。 启动时间是在报头时间标志中给出的, 并且是强制性的。对于过程值之间的时间间隔由“单位(范围)”的时间单位和“周期”因素来定义。
2	每个过程值包含一个时间标志。 不评估输入到报头的任何时间标志。其结构对应于具有 8 个字节长度的报头中的时间标志。
3	每个过程值均拥有一个相对时间差, 该时间差具有两个字的数据长度的时间单位。 绝对时间是报头中的时间标记的总和(=启动时间), 而相对时间差则使用在“单位(范围)”中设置的时间单位。报头的时间标志条目是强制性的。
4	每个过程值包含 AR_ID 子编号。 在报头中给定的时间标志应用于过程值。报头的时间标志条目是强制性的。

单位(范围)=3

指定用于单位(范围)的时间单位 = 1 或 3。

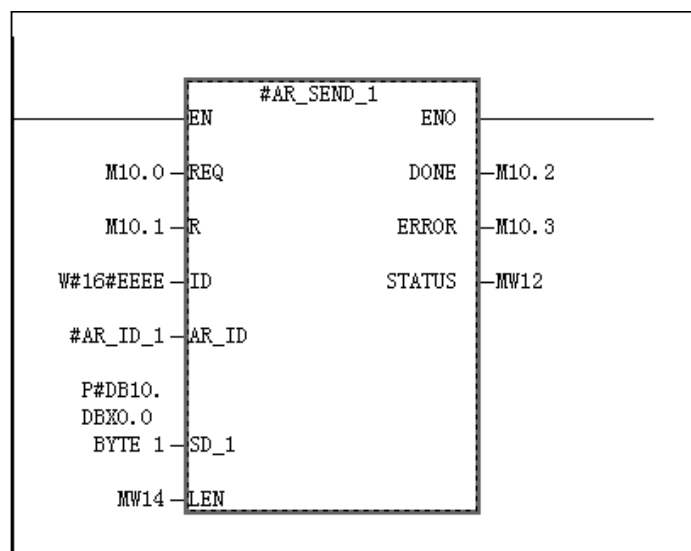
编号	含义
1	保留
2	保留
3	毫秒
4	秒
5	分
6	小时
7	日

过程数据 - 数据类型=5

过程值直接以 S7 格式存储。

编号	S7 数据类型	WinCC 数据类型
0	字节	字节
1	字	字
2	INT	SWORD
3	DWORD	DWORD
4	DINT	SDWORD
5	实型	FLOAT

根据以上的数据类型定义，我们在 S7-400 PLC 中建立数据块 DB10，参考 WinCC KNOWLEDGE BASE V5.1\Technical Information\TagLogging\Process-Controlled Measured Value Archive in the SIMATIC S7 - AS 400 章节的描述，分别在 STEP7 和 WinCC 中做相应的调用。建立 FB37 调用 SFC37 AR_SEND 函数发送数据



其中 MW14=8022，表示整个发送数据区长度为 8022 个字节，除了包含 22 个报文头之外，我们建立 2000 个 REAL 型的数据，长度 8000 个字节。

所使用的变量 M10.0 即为归档的使能位，使能之后 PLC 主动与 WinCC 进行通讯，完成后 M10.2=1。

在 OB1 调用的 FC 块中调用 FB37，自动分配 AR_ID=16#1。

然后在 WinCC 中建立一个 RAW DATA TYPE 的变量，变量记录中，选择过程变量，并制定 AR_ID 一致。

这样，通过使能位 M10.0 的控制命令，就可以将数据区中的 2000 个数据发送到 WinCC 数据库中。

这里的数据区尚没有任何数据，需要我们编程实现数据序列。

下面为双缓冲区模式实现连续记录的过程：

1. 生成信号

正常应用时，数据大多直接从模拟量模块采集，实验中为了方便检查效果，在 OB35（10ms）中生成正弦曲线，周期为 20s，2000 个数据。

```

L    MD    400
L    L#1
+D
T    MD    400
DTR
L    3.141592e-003
*R
SIN
L    1.000000e+003
*R
L    1.000000e+003
+R
T    MD    404

```

采集的数据放在 MD404 中，实数形式。

2. 数据记录到缓冲区

首先建立长度 4000 个 REAL 的数据块 DB400，作为我们的双缓冲区：

Address	Name	Type	Initial value
0.0		STRUCT	
+0.0	BUFFER_A	ARRAY[1..2000]	
*4.0		REAL	
+8000.0	BUFFER_B	ARRAY[1..2000]	
*4.0		REAL	
=16000.0		END_STRUCT	

OB35 中使用间接寻址编程，每采集一个数据，依次放入缓冲区，并在指针分别达到 2000 和 4000 的时候，生成半满信号。

Network 1: Title:

Comment:

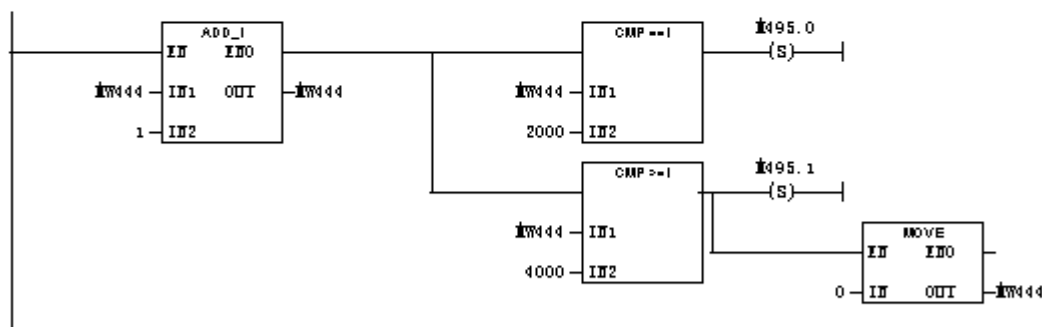
```

OPN DE 400
// MD440=F#0.0 //OE100
I MD 404
I DED [MD 440]

```

Network 2: Title:

M495.0: BUFFER 1 已满标志
M495.1: BUFFER 2 满标志
OB1程序中复制buffer数据到raw数据区。并记录系统时间作为下一次raw数据段的开始时间



Network 3: 生成下一次数据记录的buffer位置指针

Comment:

```

I MW 444
I 32
*D
I MD 440

```

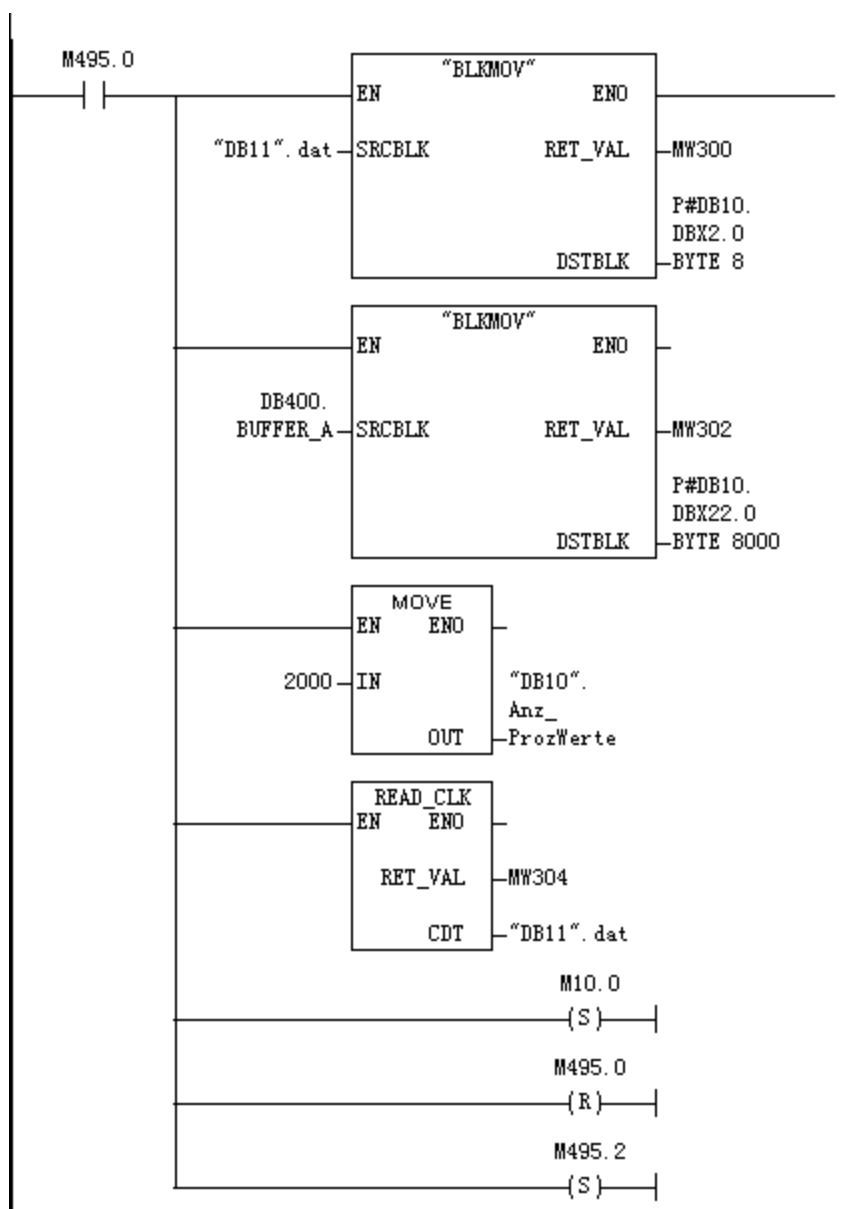
M495.0 和 M495.1 分别为缓冲区 1 和缓冲区 2 的满标志，当缓冲区 2 满的同时，切换指针回 0。另外，在每个缓冲区记录首行数据的时候，记录下当前的系统时间。

3. 数据从缓冲区发送到 WinCC

在 OB1 调用的 FC 中编程实现：根据半满标志位状态，分别完成的工作：记录开始时间值送到数据区 8 个字节。2000 个数据送到数据区 8000 个字节。指定数据长度 2000，使能通讯标志。完成后此半满标志复位。

4. 其他

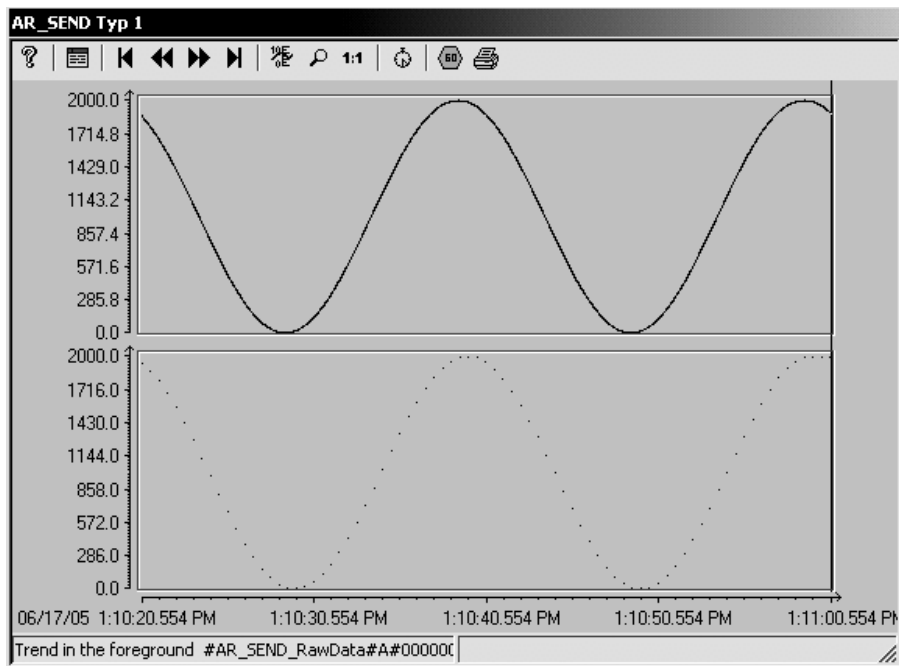
由于高速数据采集所站用的系统资源较多，采集来的数据占用空间也很大，所以通常情况下，这种高速数据的采集应当有一个开关，就是只有在需要的时候才进行数据采集。所以程序还需要做一些相应的处理，使用一个 WinCC 能控制的中间量作为触发开关。在触发的上升沿，数据指针指向首行，下降沿时，判断指针位置，判断在缓冲区 1 还是缓冲区 2，并且在缓冲区中的有效数据数量，在发送任务中根据实际的数量，而不是正常半满下的 2000。



5. 效果

在 WinCC 的趋势图画面中添加对此归档的监视, 可以看到动态效果, 每过 20s, 图像刷新一次, 一个完整的正弦曲线周期被添加, 2000 个数据记录被保存。效果图中蓝线为采集结果的点图, 对比的红色曲线为按照传统方式所能达到的最小周期 500ms 采集的结果。

另外, 以表格方式记录的采样结果:



WinCC?CSV-Provider - RT - Tables

Copyright ?1994-2004 by SIEMENS AG

\\WT401\\WinCC60_Project_CompSupport-AR_SEND\\CompSupport-AR_SEND.MCP

Date/Time	列 1
06/17/05 1:15:24.836 PM	1196.73
06/17/05 1:15:24.846 PM	1193.65
06/17/05 1:15:24.856 PM	1190.56
06/17/05 1:15:24.865 PM	1187.48
06/17/05 1:15:24.876 PM	1184.39
06/17/05 1:15:24.886 PM	1181.30
06/17/05 1:15:24.896 PM	1178.21
06/17/05 1:15:24.906 PM	1175.11
06/17/05 1:15:24.916 PM	1172.02
06/17/05 1:15:24.926 PM	1168.92
06/17/05 1:15:24.935 PM	1165.82
06/17/05 1:15:24.946 PM	1162.72
06/17/05 1:15:24.956 PM	1159.62
06/17/05 1:15:24.966 PM	1156.51
06/17/05 1:15:24.976 PM	1153.41
06/17/05 1:15:24.986 PM	1150.30
06/17/05 1:15:24.996 PM	1147.19
06/17/05 1:15:25.005 PM	1144.08
06/17/05 1:15:25.016 PM	1141.00
06/17/05 1:15:25.026 PM	1137.89
06/17/05 1:15:25.036 PM	1134.78
06/17/05 1:15:25.046 PM	1131.66

06/17/05 1:15:25.056 PM	1128.54
06/17/05 1:15:25.066 PM	1125.43
06/17/05 1:15:25.075 PM	1122.31
06/17/05 1:15:25.086 PM	1119.19
06/17/05 1:15:25.096 PM	1116.07
06/17/05 1:15:25.106 PM	1112.94
06/17/05 1:15:25.116 PM	1109.82
06/17/05 1:15:25.126 PM	1106.69
06/17/05 1:15:25.136 PM	1103.57

2005-06-17 13:15:58

Page 1 / 1

Layout:@CCTable ControlContents.RPL

_CompSupport-AR_SEND\CompSupport-AR_SEND.MCP

可以看到，每条记录之间的时间间隔为足够精确的 10ms，说明所有按照设计精度记录的数据，全都没有丢失的被记录下来了。

6. 结论和探讨

使用 S7-400PLC 配合 WinCC，我们成功实现了高速数据的采集，采集周期达到了 10ms，而且通过不同的通讯协议接口，也做了耗时统计，结果在使用 MPI 接口 187.5kbit/s，时，数据量为 6000 个时，写入时间大概要 10s，而使用 10M 工业以太网 ISO 协议，需要的时间大概为 600ms。可见在高速采集情况下，工业以太网的效果要好得多，应当尽量采用工业以太网。

参数设置表中，我们也可以使用比 10ms 更小的周期，理论上的最高限能达到 1ms，但这时除了 PLC 的定时中断需要使用 1ms 周期，从而占用了 PLC 更大的运行资源之外，缓冲区的尺寸设计也需要非常谨慎。

参考文献

1. WinCC 信息系统 V6.0
2. WinCC Comprehensive Support V6
3. WINCC KNOWLEDGE BASE V5.1
4. STEP7 编程手册