

文献标识码: B 文章编号: 1003-0492 (2022) 04-076-06 中图分类号: TP216

基于MODBUS-ASCII通讯协议的仪表数据采集

Instrument Data Acquisition Based on Modbus-ASCII Communication Protocol

★ 张钰 (北京碧水源科技股份有限公司, 北京 102206)

摘要: 流量计仪表的累计流量需要PLC通过通讯方式进行数据采集, 通讯接口为RS485接口, 通讯协议为Modbus-ASCII。本文介绍了仪表数据采集系统的结构以及所需软、硬件, 并对仪表的参数设置及通讯协议也做了详细阐述。最后以实例介绍了PLC的硬件组态与主要程序设计。

关键词: 数据采集; Modbus-ASCII协议; 程序设计

Abstract: The cumulative flow of the flowmeter requires PLC to collect data through communication. The communication interface is RS485, and the communication protocol is Modbus-ASCII. This paper introduces the structure of the instrument data acquisition system as well as the software and hardware required, and expounds the parameter setting and communication protocol of the instrument in detail. Finally, the hardware configuration and main program programming of PLC are introduced with an example.

Key words: Data acquisition; Modbus-ASCII protocol; Programming

1 引言

污水处理厂每天都要采集和存储大量的仪表数据。例如, 用于评判水质好坏的总磷、总氮、pH值、化学需氧量(COD)、氨氮等, 用于设备控制的液位、压力、流量等。这些重要的仪表数据, 通常由相应的仪表利用自带的传感器现场测量得到, 然后由可编程逻辑控制器(PLC)读取并存储仪表数据, PLC将数据传送给中控室内的工控机或服务器, 最后由工控机或服务器上传到网站或环保局。这些采集到的仪表数据可供民众进行查阅, 给科研和环境保护提供准确数据。由此可见, 仪表数据采集是很重要的环节。大部分仪表信号可由PLC的数字量或模拟量模块进行采集, 如果需要采集累计流量就必须使用通讯的方式进行数据采集, 通常PLC与仪表之间的通讯接口都为RS485接口, 通讯协议是Modbus-RTU或Modbus-ASCII等。本文以RS485接口和Modbus-ASCII协议为例论述

PLC与仪表之间通过通讯实现数据采集的过程。

2 数据采集系统的结构

本文以日处理规模5万吨的污水厂为例, 该水厂主站采用西门子S7-300系列PLC作为数据采集、设备控制的核心部件, 现场有10台开封威利的电磁流量计(电磁流量计的型号为MagneW3000)。西门子S7-300系列PLC中的CP340作为通讯模块, 此模块的通讯接口为RS485, 通讯协议为Modbus-ASCII。本系统为CP340通过通讯的方式采集10台电磁流量计的累计流量, 其网络结构如图1所示, 图中PLC(或者CP340模块)作为通讯的主站, 电磁流量计为从站。

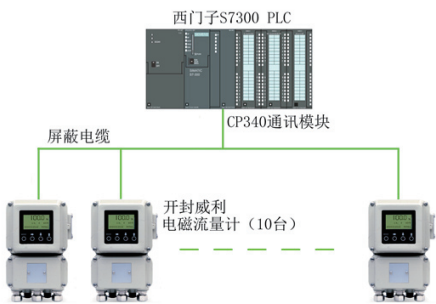


图1 网络结构

2.1 数据采集系统所需的软件

数据采集系统所需软件如下:

- (1) PLC的编程软件STEP7, 版本为STEP7 V5.6 Professional;
- (2) 驱动软件包PtP Param V5.1;
- (3) Modbus Master (ASCII) 驱动软件包。

2.2 数据采集系统所需的硬件

系统所需的主要硬件如表1所示。

表1 系统所需硬件

序号	名称	订货号称/型号	数量	备注
1	CPU模块	6ES7 315-2EH14-0AB0	1	西门子
2	电源模块	6SE7 307-1KA00-0AA0	1	西门子
3	通讯模块	6ES7 340-1CH00-0AE0	1	西门子
4	电磁流量计	MagneW3000（转换器的型号）	10	开封威利
5	存储卡	6ES7 952-1GJ00-0AE0	1	西门子

3 仪表的参数设置和Modbus-ASCII通讯协议

3.1 仪表的通讯参数设置

MagneW3000型流量计设置主要包括通讯参数、波特率、数据位数、停止位数和奇偶校验、Modbus站地址、通讯接口等。以1#电磁流量计为例，其参数设置如表2所示。

表2 流量计参数设置

传输模式	ASCII通讯
站地址	01
波特率	9600bps
编码格式	ASCII码（‘0’ - ‘9’ ‘A’ - ‘F’）
起始位	1
数据位	8
校验位	无
停止位	1
帧校验	LRC

3.2 仪表通讯协议

MagneW3000型仪表的ASCII码通讯协议，其主要包括数据传输模式、寄存器和数据类型、数据帧格式定义、数据帧校验算法、仪表变量地址定义。这里主要介绍数据帧格式定义。

3.2.1 数据帧格式

MagneW3000型仪表的协议包括4个基本命令：CMD=0x03（读1个或多个寄存器）、CMD=0x05（写COIL变量）、CMD=0x06（写单个寄存器）、CMD=0x10（写多个寄存器）。每个命令都对应的请求帧和应答帧数据格式。要读取仪表的累计流量，下面以CMD=0x03（读1个或多个寄存器）为例，来说明请求帧数据格式的含义，应答帧数据格式与请求帧格式类似，故本文不作赘述。

命令CMD=0x03，表示读1个或多个寄存器，本例数据为读取瞬时流量的数据帧，仪表地址=1。

瞬时流量的寄存器起始地址=0x0253，但数据帧中的寄存器起始地址必须是0x0253-0x0001=0x0252。

对数据格式说明时，需参考ASCII表，表3是ASCII码的一部分。表4是请求帧的数据格式。

表3 ASCII码

ASCII码值	16进制	控制字符	ASCII码值	16进制	控制字符
48	30H	0	56	38H	8
49	31H	1	57	39H	9
50	32H	2	65	41H	A
51	33H	3	66	42H	B
52	34H	4	67	43H	C
53	35H	5	68	44H	D
54	36H	6	69	45H	E
55	37H	7	70	46H	F

表4 请求帧的数据格式

序号	数据名称	ASCII数值(十六进制)	存储器代号
1	包头	3A	①
2	仪表地址	30H 31H	②③
3	功能码	30H 33H	④⑤
4	寄存器起始地址高字节	30H 32H	⑥⑦
5	寄存器起始地址低字节	35H 32H	⑧⑨
6	寄存器数量高字节	30H 30H	⑩⑪
7	寄存器数量低字节	30H 32H	⑫⑬
8	帧校验	41H 36H	⑭⑮
9	包尾	0D 0A	⑯⑰

请求帧的数据格式的说明如下：

- （1）通讯的起始符（包头），占用1字节，ASCII码值：3A；
- （2）通讯仪表地址，占用2字节，由2位的十六进制数转换而成，地址范围1~255。假设这个2位的十六进制数，表示为AB（A是十位，B是个位）。首先把“A”当作ASCII码表内的控制字符，此控制字符对应着唯一的ASCII码十六进制的值，并把ASCII码十六进制的值存入②中。同理把“B”按同样的方法进行转换，并且分别存入③中。例如，仪表地址为10（十进制），转换成2位的十六进制为0AH，把“0”、“A”看作ASCII码中的控制字符，对应着唯一的ASCII码十六进制的值分别是30H、41H。所以②内的值是30H，③内的值是41H。
- （3）功能码，占用2字节。由2位的十六进制数转换而成。转换方式与（2）同理，例如，03（十进制）表示读1个或多个寄存器，转换成2位的十六进制为03H，把“0”、“3”看作ASCII码中的控制字符，对应着唯一的ASCII码十六进制的值分别是30H、33H。所以④内的值是30H，⑤内的值是33H。
- （4）寄存器起始地址高字节，占用2字节。寄存器起始地址高字节和低字节，共占用4个字节。由4位的十六进制数转换而成。假设这个4位的十六进制数，表示为ABCD（A是千位，B是百位，C是十位，D是个位）。首先把“A”当作ASCII码

表内的控制字符，此控制字符对应着唯一的ASCII码十六进制的值，并把ASCII码十六进制的值存入⑥中。同理把“B”、“C”、“D”按同样的方法进行转换，并且分别存入⑦、⑧、⑨中。例如，读取的起始地址为0594（十进制），转换成4位的十六进制为0252H，把“0”、“2”、“5”、“2”看作ASCII码中的控制字符，对应着唯一的ASCII码十六进制的值分别是30H、32H、35H、32H。所以⑥内的值是30H，⑦内的值是32H，⑧内的值是35H，⑨内的值是32H。

（5）寄存器起始地址低字节，占用2字节。说明同（4）。

（6）寄存器数量高字节。占用2字节。寄存器数量高字节和低字节，共占用4个字节。由4位的十六进制数转换而成。假设这个4位的十六进制数，表示为ABCD（A是千位，B是百位，C是十位，D是个位）。首先把“A”当作ASCII码表内的控制字符，此控制字符对应着唯一的ASCII码十六进制的值，并把ASCII码十六进制的值存入⑩中。同理把“B”、“C”、“D”按同样的方法进行转换，并且分别存入⑪、⑫、⑬中。例如，读取的数据的个数为4（十进制）个字节，转换成4位的十六进制为0004H，把“0”、“0”、“0”、“4”看作ASCII码中的控制字符，对应着唯一的ASCII码十六进制的值分别是30H、30H、30H、34H。所以⑩内的值是30H，⑪内的值是30H，⑫内的值是30H，⑬内的值是34H。

（7）寄存器数量低字节，占用2字节。说明同上。

（8）帧校验。校验方式采用纵向冗余校验（Longitudinal Redundancy Check, LRC）：从“仪表地址”到LRC帧校验码的前一个字节，LRC校验数据被分成高4位和低4位，并被转换成ASCII码，当校验码错误，仪表将没有响应，设有如下的发送数据。LRC计算方式举例（以表4中的数据为例）：

第一步：根据LRC校验范围，把从“仪表地址”到“寄存器数量低字节”的ASCII码十六进制数值，转换成十六进制数值，结果如表5所示。

表5 ASCII转十六进制

ASCII(十六进制)	转换后(十六进制)
30H 31H	01
30H 33H	03
30H 32H	02
35H 32H	52
30H 30H	00
30H 32H	02

第二步：把表5中转换后的十六进制数相加，即01+03+02+52+00+02=5A。

第三步：相加的结果为十六进制数据，用100减去十六进制结果的后两位组成的数据，即100-5A=A6。然后，把A6转换成ASCII码再分配到⑭⑮中，所以⑭中的数据为41H，⑮中的数据为36H。

（9）包尾，占用2字节，为固定ASCII码十六进制值0D和0A。

3.2.2 MagneW3000型仪表寄存器地址

MagneW3000型仪表的寄存器地址定义如表6所示。

表6 MagneW3000型仪表的寄存器地址

变量名	寄存器地址	寄存器长度	读指令	写指令
COIL类型				
清累计	0003	---	---	5
INT类型				
流量单位	0042	1	3	6
累计单位	0046	1	3	6
功率单位	6003（热能表）	1	3	6
热能单位	6004（热能表）	1	3	6
报警	0419	1	3	---
LONG类型				
扩展正向累计	0309	2	3	---
正向累计	0311	2	3	---
扩展反向累计	0313	2	3	---
反向累计	0315	2	3	---
扩展正向累计热量	0317（热能表）	2	3	---
正向累计热量	0319（热能表）	2	3	---
扩展反向累计热量	0321（热能表）	2	3	---
反向累计热量	0323（热能表）	2	3	---
FLOAT类型				
主变量（瞬时流量）	0253	2	3	----
瞬时功率	2000（热能表）	2	3	----
入口温度	2002（热能表）	2	3	----
出口温度	2004（热能表）	2	3	----
阻尼时间s	0189	2	3	10
小信号切除%	0197	2	3	10
刻度流量	0209	2	3	10
4-20mA电流测试（mA）	0143	2	---	10
输出电流值mA	0203	2	3	---
输出频率值Hz	0229	2	3	---
频率上限Hz	0223	2	3	10
脉冲当量L/p	1103	2	3	10
脉冲宽度ms	0227	2	3	10

4 软件编程

S7-300 PLC要通过CP340通讯模块采集MagneW3000型仪表的累计流量，通讯的物理接口为RS485，通讯协议为Modbus-ASCII。编程软件为STEP7 V5.6 Professional。

4.1 PLC硬件组态

(1) 进入STEP7硬件组态，并加入硬件模块，如图2所示。

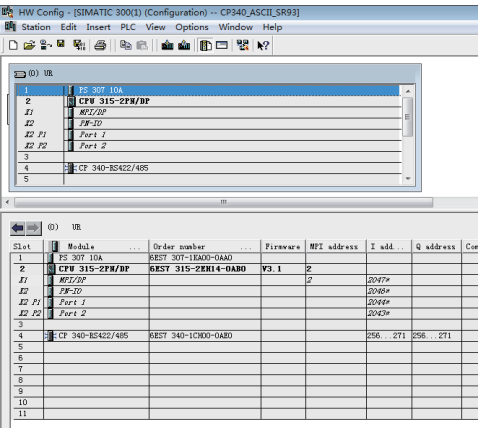


图2 硬件组态1

(2) 双击CP 340-RS422/485，弹出属性设置弹窗如图3所示。

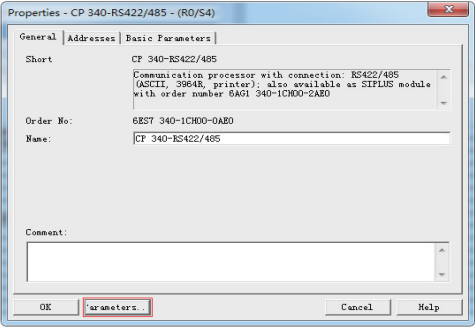


图3 CP340属性设置

(3) 点击Parameters（参数），选择ASCII协议，如图4所示。

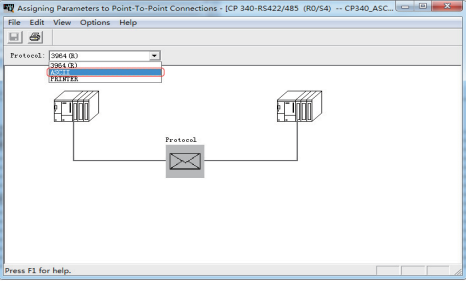


图4 协议选择

(4) ASCII协议参数设置：波特率=9600、数据位=8、停止位=1、奇偶校验=无。接口选择RS485。如图5所示。

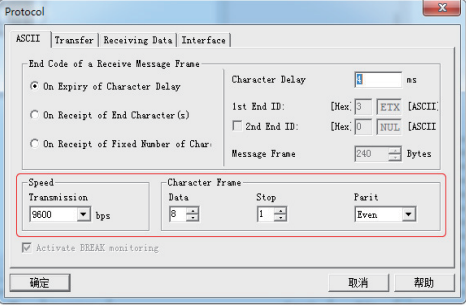


图5 ASCII协议参数设置

4.2 主控程序

PLC的工作原理是巡回扫描，周而复始地执行用户程序和其他任务。使用西门子STEP7软件编写的PLC程序是由若干块组成的，例如OB块、FB块、DB块等。因为PLC每执行一个周期任务，就会调用一次OB块，所以主控程序在OB1中编写。OB1包括时间脉冲程序、发送程序、接收程序、赋值程序、转换程序。OB1中的程序如图6所示。

```
OB1: "Main Program Sweep (Cycle)"
Comment:
Network 1: 生成1个1秒为周期的脉冲。
AN T 2
L S5T#500MS
SD T 1
A T 1
L S5T#500MS
SD T 2
Network 2: 块FB3为发送命令，向仪表发出读取累计流量的命令
A T 1
= L 20.0
BLD 103
CALL FB 3, DB21
REQ :=L20.0
R :=
LADDR :=256
DB_NO :=13
DEB_NO:=0
LEN :=17
DONE :=M1.0
ERROR :=M1.1
STATUS:=HW10
NOP 0
Network 3: 块FB2为接收命令，接收从仪表返回的数据帧
A M 1.2
= L 20.0
BLD 103
CALL FB 2, DB22
ENR_N :=L20.0
R :=
LADDR :=256
DB_NO :=HW100
DEB_NO:=0
NDR :=M1.4
ERROR :=M1.5
LEN :=HW104
STATUS:=HW14
NOP 0
Network 4: 分别给10块仪表的发送数据帧内的各个字节赋值，且给FB2赋值。
A T 1
= L 20.0
BLD 103
CALL FB 31, DB31
pulse_a:=L20.0
NOP 0
Network 5: 分别为读取的10个流量计的累计流量进行换算
A T 1
= L 20.0
BLD 103
CALL FB 30, DB30
pulse_a :=L20.0
Total_Flow1 :=MD200
Total_Flow2 :=MD204
Total_Flow3 :=
Total_Flow4 :=
Total_Flow5 :=
Total_Flow6 :=
Total_Flow7 :=
Total_Flow8 :=
Total_Flow9 :=
Total_Flow10:=
NOP 0
```

图6 主控程序

4.3 发送程序（FB3）的设计

4.3.1 发送功能块FB3的说明

图6中的Network2是发送程序，即向仪表发出读取累计流量的命令。FB3的各个参数含义如表7所示。

表7 FB3参数

LAD DR	硬件组态中的起始逻辑地址，本例中为256
DB_NO	发送数据块号，本例中为13（DB13）
DBB_NO	发送数据的起始地址，本例中为0（DB13.DBB0）
LEN	发送数据的长度，本例中为17
REO	发送数据触发位，上升沿触发，本例中为T1
R	取消通讯，本例中不用
DONE	发送完成位，发送完成且没有错误时为TRUE
ERROR	错误位，为TRUE说明有错误
STATUS	状态字，标识错误代码

4.3.2 DB13块中的数据说明

考虑到系统有10块仪表，采用轮询的方式访问每块仪表，而对于发送读取数据的DB块仅是仪表地址和LRC校验码不相同，因此，所有的仪表采用一个相同的DB块（本例DB13）来执行，根据不同的仪表号修改相应的LRC校验码和仪表地址。发送数据的DB块（本例DB13）结构如图7所示。

Address	Name	Type	Initial value	Actual value	Comment
0.0 DB_VAR0		BYTE	#B16#3A	#B16#3A	包头
1.0 DB_VAR1		BYTE	#B16#30	#B16#30	仪表地址，十六进制数据的十位
2.0 DB_VAR2		BYTE	#B16#31	#B16#31	仪表地址，十六进制数据的个位
3.0 DB_VAR3		BYTE	#B16#30	#B16#30	功能码，十六进制数据的十位
4.0 DB_VAR4		BYTE	#B16#33	#B16#33	功能码，十六进制数据的个位
5.0 DB_VAR5		BYTE	#B16#30	#B16#30	寄存器起始地址高字节，十六进制数据的千位
6.0 DB_VAR6		BYTE	#B16#30	#B16#30	寄存器起始地址高字节，十六进制数据的百位
7.0 DB_VAR7		BYTE	#B16#31	#B16#31	寄存器起始地址低字节，十六进制数据的十位
8.0 DB_VAR8		BYTE	#B16#30	#B16#30	寄存器起始地址低字节，十六进制数据的个位
9.0 DB_VAR9		BYTE	#B16#30	#B16#30	寄存器数量高字节，十六进制数据的千位
10.0 DB_VAR10		BYTE	#B16#30	#B16#30	寄存器数量高字节，十六进制数据的百位
11.0 DB_VAR11		BYTE	#B16#30	#B16#30	寄存器数量低字节，十六进制数据的十位
12.0 DB_VAR12		BYTE	#B16#30	#B16#30	寄存器数量低字节，十六进制数据的个位
13.0 DB_VAR13		BYTE	#B16#45	#B16#45	帧校验，十六进制数据的十位
14.0 DB_VAR14		BYTE	#B16#31	#B16#31	帧校验，十六进制数据的个位
15.0 DB_VAR15		BYTE	#B16#00	#B16#00	包尾
16.0 DB_VAR16		BYTE	#B16#00	#B16#00	
17.0 DB_VAR17		BYTE	#B16#00	#B16#00	
18.0 DB_VAR18		BYTE	#B16#00	#B16#00	
19.0 DB_VAR19		BYTE	#B16#00	#B16#00	
20.0 DB_VAR20		BYTE	#B16#00	#B16#00	
21.0 DB_VAR21		BYTE	#B16#00	#B16#00	
22.0 DB_VAR22		BYTE	#B16#00	#B16#00	
23.0 DB_VAR23		BYTE	#B16#00	#B16#00	
24.0 DB_VAR24		BYTE	#B16#00	#B16#00	
25.0 DB_VAR25		BYTE	#B16#00	#B16#00	

图7 DB块结构

第1行的3A为包头。

第2、3行是设定仪表地址。B#16#30、B#16#34转换后为“0”和“1”，表示仪表的地址为01。

第4、5行是设定功能码。B#16#30、B#16#33转换后为“0”和“3”。03表示要读取一个或多个寄存器。

第6、7、8、9行是设定寄存器的首地址。B#16#30、B#16#33、B#16#31和B#16#30转换后为“0”、“3”、“1”和“0”。地址0310为仪表的累计地址。

第10、11、12、13行是设定读取寄存器的数量。B#16#30、B#16#30、B#16#30和B#16#38转换后为“0”、“0”、“0”和“8”。0008表示读取8个字节。

第14行为帧校验（LRC校验）：B#16#45、B#16#31转换

后为“E”和“1”，即校验码为E1。

第15、16行为包尾。

对于写数据发送命令程序块，主要完成填写设置数据地址、BCC码计算、仪表地址的修改和设置数据的填写，最后调用发送FB3将数据发送到对应的仪表。

4.4 接收程序（FB2）的设计

4.4.1 接收功能块FB2的说明

图7中的Network3是接收程序，接收仪表应答帧数据，数据帧内包含累计流量的数据。FB2的各个参数含义如表8所示。

表8 功能块FB2参数

LAD DR	硬件组态中的起始逻辑地址，本例中为256
DB_NO	接收数据块号，本例中为MW100
DBB_NO	接收数据的起始地址，本例中为0（DBX.DBB0，X为MW100内的值）
LEN	接收数据的长度，本例中为MW104
EN_R	使能接收位，本例中为M1.2
R	取消通讯，本例中不用
NDR	接收完成位，接收完成并没有错误为TRUE
ERROR	错误位，为TRUE说明有错误
STATUS	状态字，标识错误代码

4.4.2 接收数据块DB1块中的数据说明

10块仪表，采用轮询的方式访问每块仪表，因为每个流量计的累计流量是不同的，所以把接收的10个仪表的数据分别存放在10个数据块中，本例为DB1~DB10。以DB1内的数据为例，图8为接收到1号仪表的数据。

Address	Name	Type	Initial value	Actual value	Comment
0.0 DB_VAR0		BYTE	#B16#3A	#B16#3A	包头
1.0 DB_VAR1		BYTE	#B16#30	#B16#30	仪表地址
2.0 DB_VAR2		BYTE	#B16#31	#B16#31	仪表地址
3.0 DB_VAR3		BYTE	#B16#30	#B16#30	功能码
4.0 DB_VAR4		BYTE	#B16#33	#B16#33	功能码
5.0 DB_VAR5		BYTE	#B16#30	#B16#30	数据长度
6.0 DB_VAR6		BYTE	#B16#38	#B16#38	数据长度
7.0 DB_VAR7		BYTE	#B16#30	#B16#43	寄存器0310的寄存器高字节，累计流量十六进制数据的千位
8.0 DB_VAR8		BYTE	#B16#30	#B16#31	寄存器0310的寄存器高字节，累计流量十六进制数据的百位
9.0 DB_VAR9		BYTE	#B16#30	#B16#34	寄存器0310的寄存器低字节，累计流量十六进制数据的十位
10.0 DB_VAR10		BYTE	#B16#30	#B16#38	寄存器0310的寄存器低字节，累计流量十六进制数据的个位
11.0 DB_VAR11		BYTE	#B16#43	#B16#43	寄存器0311的寄存器高字节，累计流量十六进制数据的千位
12.0 DB_VAR12		BYTE	#B16#31	#B16#30	寄存器0311的寄存器高字节，累计流量十六进制数据的百位
13.0 DB_VAR13		BYTE	#B16#34	#B16#30	寄存器0311的寄存器低字节，累计流量十六进制数据的十位
14.0 DB_VAR14		BYTE	#B16#38	#B16#30	寄存器0311的寄存器低字节，累计流量十六进制数据的个位
15.0 DB_VAR15		BYTE	#B16#45	#B16#45	帧校验
16.0 DB_VAR16		BYTE	#B16#42	#B16#42	帧校验
17.0 DB_VAR17		BYTE	#B16#00	#B16#00	
18.0 DB_VAR18		BYTE	#B16#00	#B16#00	
19.0 DB_VAR19		BYTE	#B16#00	#B16#00	
20.0 DB_VAR20		BYTE	#B16#00	#B16#00	
21.0 DB_VAR21		BYTE	#B16#00	#B16#00	
22.0 DB_VAR22		BYTE	#B16#00	#B16#00	
23.0 DB_VAR23		BYTE	#B16#00	#B16#00	
24.0 DB_VAR24		BYTE	#B16#00	#B16#00	
25.0 DB_VAR25		BYTE	#B16#00	#B16#00	

图8 1号仪表数据

4.5 赋值程序

复制程序代码如图9所示。

4.6 数据转换程序

从仪表中读取的数据进行一定的转换才能得到需要的累计流量。根据仪表协议可知，MagneW3000型仪表发送回来的流量计的数据共有8个字节。每一个字节内的数值为一个ASCII码十六进制的值。需要将此值转换成ASCII码表中的“控制字符”。根据协议可知，流量计的数据共有8个字节中的前4个字

```
REGIN
//上升沿脉冲
pulse:=pulse_a AND NOT pulse_b;
pulse_b:=pulse_a;
//*****使地址代码每秒加1,大于等于11后再返回到1*****//
IF pulse THEN addr_code:=addr_code+1;END_IF;
IF addr_code=11 OR addr_code=0 THEN addr_code:=1;END_IF;

//*****将请求帧的参数写入到数据块DB13中,由FB3将参数发送给仪表*****//
IF addr_code=1 THEN //如果地址代码为01,则执行下面的命令
db13.dbb0:=B#16#3A; //包头
db13.dbb1:=B#16#30;db13.dbb2:=B#16#31; //2个字节内的ASCII码转换后为01,表示仪表地址是01
db13.dbb3:=B#16#30;db13.dbb4:=B#16#33; //2个字节内的ASCII码转换后为03,表示读取命令
db13.dbb5:=B#16#30;db13.dbb6:=B#16#33; //2个字节内的ASCII码转换后为03,表示要读取的4位十六进制
//寄存器的首地址的前两位
db13.dbb7:=B#16#31;db13.dbb8:=B#16#30; //2个字节内的ASCII码转换后为10,表示要读取的4位十六进制
//寄存器的首地址的后两位
db13.dbb9:=B#16#30;db13.dbb10:=B#16#30; //2个字节内的ASCII码转换后为00,表示要读取的4位十六进制
//寄存器数量的前两位
db13.dbb11:=B#16#30;db13.dbb12:=B#16#38; //2个字节内的ASCII码转换后为08,表示要读取的4位十六进制
//寄存器数量的后两位
db13.dbb13:=B#16#45;db13.dbb14:=B#16#31; //校验码LRC:51
db13.dbb15:=B#16#0D;db13.dbb16:=B#16#0A; //包尾
//*****设置应答帧参数*****//
nv100:=1; //设置应答帧数据的存储数据块号
nv104:=25; //设置应答帧的数据长度(字节)
a1.2:=1; //把FB2的使能端设置成接收状态
END_IF;
```

图9 赋值程序

节,应该转换成第一个十进制的INT值,后4个字节转换成转换成第二个十进制的INT值。然后,把得到的第一个十进制的值,当作某个双整数的高16位,第二个十进制的值,当作这个双整数的低16位,这样合成一个新的双整数(DINT)值。这个新的双整数(DINT)值就是最终的累计流量的值。转换的详细步骤如下:

第一步,把8个字节中的前4个字节按照顺序,转换成ASCII码表中的“控制字符”;

第二步,把这4个“控制字符”按顺序转化成4个十进制无符号整数(WORD);

第三步,把第一个十进制无符号整数(WORD),左移12位,第二个数左移8位,第三个数左移4位,第四个数不移位。移位之后把这4个十进制无符号整数(WORD),转换成整数(INT);

第四步,把4个数据相加,得到一个整数(INT)值;

第五步,按第一、二、三、四步的方法把8个字节中的后4个字节,转换成另一个整数;

第六步,把得到的2个整数转换成双整数(DINT)。把第一个双整数,左移16位,然后与第二个双整数相加,这样就得到最终结果。

编写的程序如图10所示。

```
BEGIN//程序开始
//上升沿脉冲
pulse:=pulse_a AND NOT pulse_b;
pulse_b:=pulse_a;
//*****使地址代码每秒加1,大于等于11后再返回到1*****//
IF pulse THEN addr_code:=addr_code+1;END_IF;
IF addr_code=11 OR addr_code=0 THEN addr_code:=1;END_IF;

IF addr_code=1 THEN
in_data1_first:=DB1.DBB7; in_data1_second:=DB1.DBB8;
in_data1_third:=DB1.DBB9; in_data1_fourth:=DB1.DBB10;
in_data2_first:=DB1.DBB11; in_data2_second:=DB1.DBB12;
in_data2_third:=DB1.DBB13; in_data2_fourth:=DB1.DBB14;
END_IF;

IF addr_code=2 THEN
in_data1_first:=DB2.DBB7; in_data1_second:=DB2.DBB8;
in_data1_third:=DB2.DBB9; in_data1_fourth:=DB2.DBB10;
in_data2_first:=DB2.DBB11; in_data2_second:=DB2.DBB12;
in_data2_third:=DB2.DBB13; in_data2_fourth:=DB2.DBB14;
END_IF;
:
//应管帧内的累计流量共占有8个字节,前4个字节都为0,这里忽略转换过程。
//下面是后4个字节,既第5、6、7和8个字节的转换程序。
//本小段为第6个字节由ASCII码值转换成10进制数的程序//
data2_first:=BYTE_TO_INT(in_data2_first)-48;
IF (data2_first > 9) THEN
data2_first:=data2_first-7;
END_IF;
data2_first:=WORD_TO_INT(SHL(IN:=INT_TO_WORD(data2_first),N:=8));

//本小段为第7个字节由ASCII码值转换成10进制数的程序//
data2_third:=BYTE_TO_INT(in_data2_third)-48;
IF (data2_third > 9) THEN
data2_third:=data2_third-7;
END_IF;
data2_third:=WORD_TO_INT(SHL(IN:=INT_TO_WORD(data2_third),N:=4));

//本小段为第8个字节由ASCII码值转换成10进制数的程序//
data2_fourth:=BYTE_TO_INT(in_data2_fourth)-48;
IF (data2_fourth > 9) THEN
data2_fourth:=data2_fourth-7;
END_IF;
data2_fourth:=WORD_TO_INT(SHL(IN:=INT_TO_WORD(data2_fourth),N:=0));
//转换后的8个10进制数,分别为流量计的累计流量的“千万位”、“百万位”、“十万位”、“万位”、
//“千位”、“百位”、“十位”、“个位”,并合并在一起。
data1:=WORD_TO_DINT(SHL(IN:=INT_TO_WORD(data1_first+data1_second+data1_third+data1_fourth),N:=16));
data2:=INT_TO_DINT(data2_first+data2_second+data2_third+data2_fourth);

//分别给10个仪表发送转换数据
IF addr_code=1 THEN Total_Flow1:=data1+data2;END_IF;
IF addr_code=2 THEN Total_Flow2:=data1+data2;END_IF;
IF addr_code=3 THEN Total_Flow3:=data1+data2;END_IF;
IF addr_code=4 THEN Total_Flow4:=data1+data2;END_IF;
IF addr_code=5 THEN Total_Flow5:=data1+data2;END_IF;
IF addr_code=6 THEN Total_Flow6:=data1+data2;END_IF;
IF addr_code=7 THEN Total_Flow7:=data1+data2;END_IF;
IF addr_code=8 THEN Total_Flow8:=data1+data2;END_IF;
IF addr_code=9 THEN Total_Flow9:=data1+data2;END_IF;
IF addr_code=10 THEN Total_Flow10:=data1+data2;END_IF;

END_FUNCTION_BLOCK
```

图10 数据转换程序

5 结束语

数据采集是自动化系统的重要组成部分,利用通讯的方式对仪表进行数据采集,在现代化的污水厂里是最普遍的数据采集方式。在污水处理厂中,一般的通讯端口都为RS485,但是通讯协议有许多种,例如PROFIBUS-DP、工业以太网等。使用Modbus-ASCII通讯协议采集仪表数据,投入成本更低,调试更加简单方便,是数据采集较好的选择方式。**AP**

作者简介:

张钰(1988-),女,四川人,中级工程师,硕士,现就职于北京碧水源科技股份有限公司,主要从事污水厂的电气自动化系统设计、编程与调试工作。

参考文献:

- [1] 西门子(中国)有限公司. CP340/CP341/CP440/CP441communication and programming[Z].
- [2] 西门子(中国)有限公司. S7300和S7400语句表编程手册[Z].
- [3] 西门子(中国)有限公司. S7-SCL编程[Z].
- [4] 西门子(中国)有限公司. SIMATIC用STEP7-V5.5编程手册[Z].
- [5] 开封威利流量仪表有限公司. MagneW3000电磁流量计转换器MODBUS协议用户手册(2016版)[Z].
- [6] 开封威利流量仪表有限公司. MagneW3000操作手册[Z].