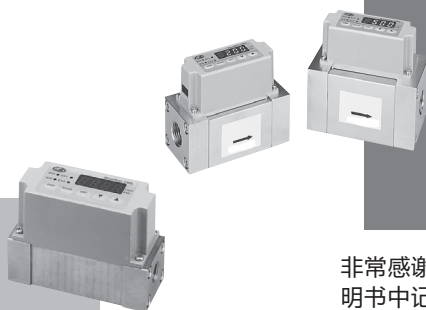




气体质量流量计

使用说明书 通讯功能篇



非常感谢您购买本公司产品。本使用说明书中记述了正确安全使用本机通讯功能的必要事项。

对于承担使用本机通讯功能的操作盘、装置的设计、维护的工作人员请务必在阅读理解本书的基础上使用。

此外，本使用说明书不只在安装时，在维护和故障维修时也是必不可少的。请常备此手册以供参考。

阿自倍尔株式会社

在订货和使用时，请务必登入以下网站，仔细阅读“关于订购与使用的承诺事项”。
<http://www.azbil.com/cn/products/factory/order.html>

重要事项

通讯中频繁写入数据的场合，请对RAM的地址进行写入。写入EEPROM的场合，写入次数有限制。限制次数是10 万次。
另外，写入 RAM 的场合，当本产品停电时，RAM 的数据将消失，回到 EEPROM 上的数据。

要求

请务必把本使用说明书送到本产品使用者手中。

禁止擅自复印和转载全部或部分本使用说明书的内容。今后内容变更时恕不事先通知。

本使用说明书的内容，经过仔细审查校对，万一有错误或遗漏，请向本公司提出。

对客户应用结果，本公司有不能承担责任的场合，敬请谅解。

© 2007-2024 Azbil Corporation All Rights Reserved.

μF™是阿自倍尔株式会社的注册商标。

本使用说明书的标记

- 为避免给您及他人造成人体伤害及财产损失，防患于未然，按照以下分类对安全注意事项进行说明。



警告

当错误使用本机时，可能会造成使用者死亡或重伤的危险情况。



注意

当错误使用本机时，可能会造成使用者轻伤或财物损失的危险情况。

- 本书中使用了如下的记号及对标记方法进行说明。



: 本符号表示使用上必须“注意”的内容。



: 本符号表示必须“禁止”的内容。



: 本符号表示必须执行的“指示”内容。

 使用上的注意事项: 表示在使用时敬请注意的事项。

 参考: 表示知道该内容后易于理解。

 : 表示参考的项目及页码。

①②③: 表示操作的顺序或对图等进行相应说明的部分。

>>: 表示操作的结果及操作后的状态。

03、P-07: 表示设定显示部的 7 段显示。

[MODE] 键: 表示设定显示部的键。

安全注意事项

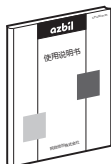
⚠注意

	请务必在全部切断本机及连接机器的电源后再进行本机的安装・拆卸及接线。 否则有触电的危险。
	请勿分解本机。 否则有发生故障的危险。
	请按规定的标准、指定的电线及施工方法正确配线。 否则可能造成本机故障。
	请在规格书中记载的使用条件(温度、湿度、电压、振动、撞击、安装方向、环境等) 范围内使用本产品。 否则有发生故障的危险。
	请不要让线头、铁粉、水等进入本机的机箱内。 否则有产生误动作或故障的危险。

本使用说明书的定位

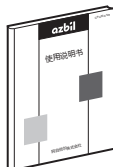
与气体质量流量计通讯功能篇相关的使用说明书共分 5 册。请根据需要阅读相应的使用说明书。

如果您手中无相关的使用说明书时，请向本公司或代理店索取。



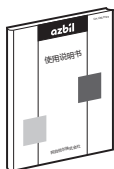
气体质量流量计 CMS0010/0050/0200/0500/1000/2000
氢气・氦气对应 资料编号 CP-UM-5207C

用本机进行装置设计、制作的人员，请务必阅读。
本机对使用上的安全注意事项、安装、接线、故障时的处理及主要规格进行说明。



气体质量流量计 CMS9500/0002/0005/0020/0050/0200/0500
SUS/SUS316 型 资料编号 CP-SP-5384C

用本机进行装置设计、制作的人员，请务必阅读。
本机对使用上的安全注意事项、安装、接线、故障时的处理及主要规格进行说明。



气体质量流量计 通讯功能篇 资料编号 CP-SP-1184C

本书。
使用本机通讯功能的人员请务必阅读。
对通讯的概要、接线、通讯步骤及本机的通讯数据一览、故障时的处理及通讯规格进行说明。

本使用说明书的构成

本使用说明书的构成如下。

第 1 章 概 要

本机的通讯功能的说明。

第 2 章 接 线

本机与其他机器间用 RS-485 通讯时的连接方法的说明。

第 3 章 设 定

本机通讯设定的说明。

第 4 章 通讯步骤

通讯步骤、电文的构成、数据的读出、写入与信号的时间的说明。

第 5 章 通讯数据地址一览

本机通讯中使用的各种数据的地址一览。

第 6 章 主站用通讯程序例

本机制作通讯程序时的注意事项及程序例的说明。

第 7 章 故障处理

本机的通讯不正常动作时的检查要点的说明。

第 8 章 通讯规格

本机的通讯规格的说明。

附 录

代码表。

目 录

本使用说明书的标记
安全注意事项
本使用说明书的定位
本使用说明书的构成

第 1 章	概 要	1-1
	■ 特 长	1-1
	■ 构成例	1-1
第 2 章	接 线	2-1
第 3 章	设 定	3-1
	■ 设定方法	3-1
第 4 章	通讯步骤	4-1
4 - 1	通讯步骤及电文的概要	4-1
	■ 通讯步骤	4-1
	■ 电文的构成	4-1
	■ 数据链层	4-2
	■ 应用层	4-4
4 - 2	命令的说明	4-5
	■ 连续数据地址读出命令 (RS 命令)	4-5
	■ 连续数据地址写入命令 (WS 命令)	4-6
4 - 3	应用层的数值表现	4-7
4 - 4	结束代码一览	4-8
	■ 正常及警告结束	4-8
	■ 异常结束	4-8
4 - 5	时间规格	4-9
	■ 命令电文、应答电文时间规格	4-9
	■ RS-485 驱动控制时间规格	4-9

第 5 章	通讯数据地址一览	5-1
5 - 1	通讯数据使用的预备知识	5-1
	■ 通讯数据的种类和形式	5-1
	■ 通讯数据的存储内存	5-1
	■ 数据地址	5-2
	■ 数据的单位・小数点位置	5-2
5 - 2	通讯数据地址一览	5-3
	■ 机器种类关连数据	5-3
	■ 动作状态关连数据	5-4
	■ 瞬时流量关连数据	5-5
	■ 累计流量关连数据	5-5
	■ 功能设定关连数据	5-6
	■ 参数设定关连数据	5-8
第 6 章	主站用通讯程序例	6-1
	■ 程序执行前	6-1
	■ 执行程序	6-1
	■ 数据读出/写入样板程序	6-2
第 7 章	故障处理	7-1
	■ 不能通讯时的确认项目	7-1
第 8 章	通讯规格	8-1
附 录		附-1
	■ 代码表	附-1

第1章 概要

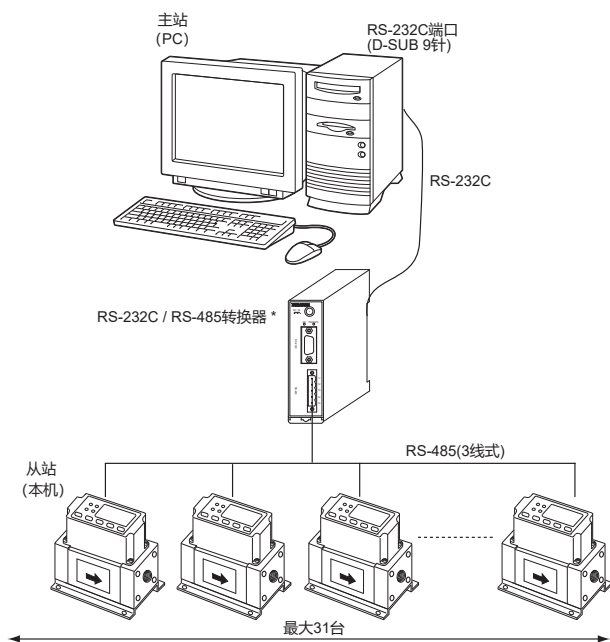
本产品(称为本机或从站)可通过通讯与PC或PLC等上位机器(称为主站)进行设定值或数据等的交换。另外,通讯程序可由用户制作。

■ 特 长

本机的通讯功能有如下特长。

- 对1台作为主站的上位机器,最大可连接31台本机。
- 上位机器的通讯规格为RS-232C的场合,需要使用另售的通讯转换器。可实现RS-232C $\leftrightarrow$ RS-485的转换。
- 传输速度最大可对应9600bps。

■ 构成例



* RS-232C / RS-485转换器可使用本公司产品。

第2章 接线

⚠注意



请务必在全部切断本机及连接机器的电源后再进行本机的安装·拆卸及接线。否则有触电的危险。



请勿分解本机。否则有发生故障的危险。



请按规定的标准、指定的电线及施工方法正确配线。否则可能造成本机故障。



请不要让线头、铁粉、水等进入本机的机箱内。否则有产生误动作或故障的危险。



参考

- 关于 RS-485 通讯线以外的接线方法，请参阅

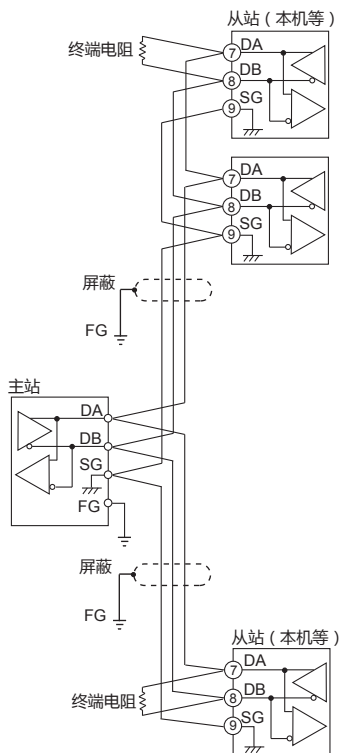


气体质量流量计 CMS0010/0050/0200/0500/1000/2000
氢气·氦气对应 使用说明书 CP-UM-5207C、



气体质量流量计 CMS9500/0002/0005/0020/0050/0200/
0500 SUS/SUS316 型使用说明书 CP-UM-5384C。

以下是 RS-485 连接方法的一例。



通讯线路的两端上，请安装 $150\Omega \pm 5\%$ 的 $1/2W$ 以上的终端电阻。
屏蔽的FG接地不是在屏蔽的两端，而是单侧1点接地。

❗ 使用上的注意事项

- 请务必连接 SG。如果不连接，将不能获得稳定的通讯。

第3章 设定

为了实现通讯功能，请事先按下记顺序对本机的通讯条件及机器地址进行设定，使其与本站的设定相吻合。

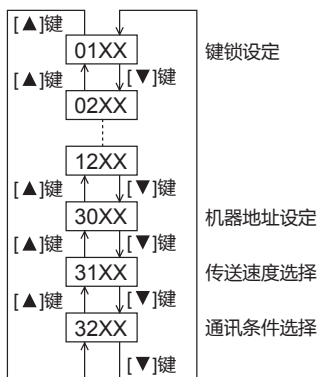
■ 设定方法

按下记步骤进行功能的设定。

① 按 [MODE] 键。

》显示部的上位 2 位变成闪烁状态。显示部的上位 2 位是设定项目、下位 2 位显示其内容。

② 按 [▲] 键或 [▼] 键，选择希望的设定模式编号（下页）。



③ 按 [ENT] 键。

》设定项目被选择，下位 2 位变成闪烁状态。

④ 按 [▲] 键或 [▼] 键，下位 2 位处选择希望的设定值，按 [ENT] 键。

》4 位全部灯亮，确定设定值。

⑤ 继续设定时，请回到①重复操作。
如果没有其他设定的项目，进入⑥。

⑥ 按 [DISP] 键。

》从功能设定模式回到瞬时流量显示或累计流量显示。

❗ 使用上的注意事项

- 在④的操作中（设定闪烁中）按 [DISP] 键时，设定值不更新，保持前次值。

● 设定项目

模式	项目内容	设定编号 及内容	初始值	参 考
30	机器地址设定	0 : 不使用通讯功能 1 ~ 99: 机器地址	0	0 时通讯功能不动作 请设定与其他从站不同的地址
31	传送速度选择	0: 9600bps 1: 4800bps 2: 2400bps	0	
32	通讯条件选择	0: 8位数据、偶数校验、 停止位1 1: 8位数据、无校验、 停止位 2	0	

第 4 章

通讯步骤

4 - 1

通讯步骤及电文的概要

在此对通讯步骤的概要及电文构成的概念进行说明。

■ 通讯步骤

通讯的步骤如下。

- ① 主站向要通讯的 1 台从站发送命令电文。
- ② 收到命令电文的从站，根据电文的内容进行读出或写入处理。
- ③ 然后，从站把根据处理内容的电文作为应答电文返信。
- ④ 主站接收应答电文。

■ 电文的构成

电文构成如下。

电文大致分为数据链层、应用层。

- 数据链层
含有通讯必要的基本信息的层。含有命令电文的去向或电文的检查信息。
- 应用层
读出写入数据的层。内容根据目的的不同而不同。

电文由下图的①～⑨构成。
应用层中，存储来自主站的命令电文的命令和来自子局的应答电文的响应



■ 数据链层

● 数据链层的概要

数据链层是固定长,各数据的位置、文字数都有规定。但ETX以后的数据链层的数据位置按应用层的文字数移动,但文字长不变化。

● 应答开始条件

只有当数据链层的电文构成(机器地址、子地址、检查和、1帧的电文长等)全部正确的场合,从站才发送应答电文。其中任一个发生错误的场合,应答电文不被发送,处于STX收信等待状态。

● 数据链层的数据定义一览

以下是数据链层的数据定义一览。

数据名	字符串代码	字节数	数据的含义
STX	02H	1	电文的起始
机器地址	用 16 进制数的字符串代码表示	2	通讯对象机器的区别
子地址	用 16 进制数的字符串代码表示	2	固定为 00
设备区分代码	"X"(58H) 或 "x"(78H)	1	机器的种别
ETX	03H	1	应用层的结束
检查和	把 00H ~ FFH 用 2 位 16 进制数的字符串代码表示	2	电文的检查和
CR	0DH	1	电文的结束 (1)
LF	0AH	1	电文的结束 (2)

● 数据的说明

• STX(02H)

从站收到STX的场合,判断为发送电文的起始。
所以,不管之前处于何种接收状态,都回到初始状态,收到第1文字为STX的电文后开始处理。这是考虑到由于干扰等发生命令电文异常的场合,从主站侧的下次正确的电文(例如,RETRY电文)开始,恢复从站的应答。

- 机器地址

从站根据主站的命令电文,当有自己机器地址的场合,处理命令,作成应答电文。机器地址=0(30H30H)的场合,即使机器地址一致,也作为无应答。另外,从站在应答电文上加上自己的机器地址回信。

机器地址用16进制数的2位字符串表示。

例: 通讯对象的机器地址为10时

① 10(10进制数) = 0AH(16进制数)

② 转换成字符串代码

0 = 30H

A = 41H

③ ②得出的「0A」(30H 41H)是机器地址。

- 子地址

由于本机不使用子地址,设定为"00"(30H 30H)。从站在应答电文上加上子地址回信。

- 设备区分代码

本机的设备区分代码设定为X(58H)或x(78H)。根据机器系列进行规定,不能使用其他的文字。从站在应答电文上加上与接收电文相同的设备区分代码后回信。初次使用X(58H),为了区分再送电文,使用x(78H)很方便。

- ETX

ETX表示应用层的结束。

- 检查和

校验电文在通讯途中是否因为某种异常(例如干扰)而未产生变化的值。16进制数的2文字。

- 检查和的作成方法

① 从STX到ETX的电文按1字节单位进行加算。

② 加算结果取2的补码。

③ 转换成字符串代码。

例: 以P4-1的电文例求出。

① 从STX到ETX的字符串代码按每1字节加算

计算结果的下位1字节是76H。

② 76H的2的补码为8AH。

③ 把8AH转换成字符串代码,结果为(38H)与(41H)。

- CR/LF

表示电文的最后。LF接收结束后立即变成接收电文处理的许可状态。

■ 应用层

● 应用层的构成

应用层的构成如下。

项 目	内 容
命令	"RS"(10 进制数形式的连续数据地址读出命令)
	"WS"(10 进制数形式的连续数据地址写入命令)
数据分隔符	RS,WS: ","(逗号)
数据地址	RS,WS: "501W" 等
读出数	"1" 等的字符串表示的数值
写入数值	RS,WS: "100" 等字符串表示的数值

● 1 帧可访问的数据数

种类	命令的说明	RAM 领域	EEPROM 领域
RS	10 进制数形式的读出命令	8	8
WS	10 进制数形式的写入命令	4	4

4 - 2 命令的说明

■ 连续数据地址读出命令(RS 命令)

是读出连续地址的数据的命令。

可从指定的读出起始地址开始,用一个命令就能读出连续地址的内容。

● 命令电文

R	S	,	1	5	0	1	W	,	1
①	②		③					②	④

应用层									
-----	--	--	--	--	--	--	--	--	--

- ① 命令
- ② 数据分隔符
- ③ 读出起始地址
- ④ 读出数据数

● 应答电文

• 正常时(1数据读出)

0	0	,				
①	②		③			

• 正常时(多个数据读出)

0	0	,												
①	②		③				②		④		②	⑤		

• 异常时

X	X
①	

XX中含有结束代码(异常)。

- ① 结束代码*
- ② 数据分隔符
- ③ 数据1
- ④ 数据2 ~ (n-1)
- ⑤ 数据n

*有关结束代码的内容,请参阅

👉 4-4 结束代码一览 (4-8页)。

■ 连续数据地址写入命令 (WS 命令)

对连续地址进行数据写入的命令。

● 命令电文

W	S	.	1	5	0	1	W	.	1	.	6	5
①	②				③			②	④	②		⑤

- ① 命令
- ② 数据分隔符
- ③ 写入起始地址
- ④ 写入数据(第1字)
- ⑤ 写入数据(第2字)

● 应答电文

- 正常时

0:0
①

- 异常/警告时

X:X
①

XX中含有结束代码(异常)。

- ① 结束代码*

*有关结束代码的内容,请参阅
👉 4-4 结束代码一览 (4-8页)。

4 - 3 应用层的数值表现

各数值需要进行去零表示。

以下是包含了不进行去零表示的规格。主站的命令电文必需全部发送去零表示后的数据。

● RS、WS 命令の場合

项 目	规 格	异常时处理
不要的空格	不可附加	中断电文处理，把异常结束 代码作为应答电文回信
不要的零	不可附加	
数值 = 0	不能省略 务必使用 "0"	
其他不要的文字	表示负数时，在数值的开始附加 " - "，不能附加其他文字 正数值的场合，不能附加 " + "	
可使用的数值范围	- 32768 ~ + 32767 不能超过该范围	

4 - 4 结束代码一览

对应答电文，必须返回结束代码。

■ 正常及警告结束

结束代码	种 别	内容及动作
00	正常	正常结束
21	警告	通讯向不可设定的地址写入 该地址里不会写入任何数据，处理继续
23	警告	访问了范围外的地址，终止读出 但之前的写入处理已完成

■ 异常结束

结束代码	种 别	内容及动作
40	异常	地址设定不正确 废弃所有电文
41	异常	WS 或 RS 未设定 废弃所有电文
43	异常	ETX(03H) 未设定到正确的位置 地址后的「,」未设定 读出字数异常 废弃所有电文
46	异常	地址异常 废弃所有电文
47	异常	读出字数异常 废弃所有电文
48	异常	写入数值异常 访问了范围外的地址，终止写入 除了异常的地址外，进行了写入操作。
99	异常	未定义命令或其他的电文异常 废弃所有电文

4 - 5 时间规格

■ 命令电文、应答电文时间规格

有关主站的命令电文发送与从站的应答电文发送的时间规格,需要注意以下事项。

● 应答监视时间

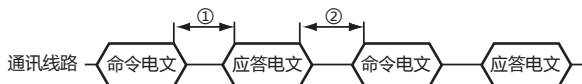
主站发送命令电文完毕后到开始接收从站的应答电文为止的最长应答时间为2秒(①的部分)。

所以,请把应答监视时间设定为2秒。通常当应答监视时间到的场合,执行命令电文的再发送。

详见

👉 第6章 主站用通讯程序。

● 发送开始时间



① 主站发送完毕后到从站开始发送为止的时间 = 2000ms以下

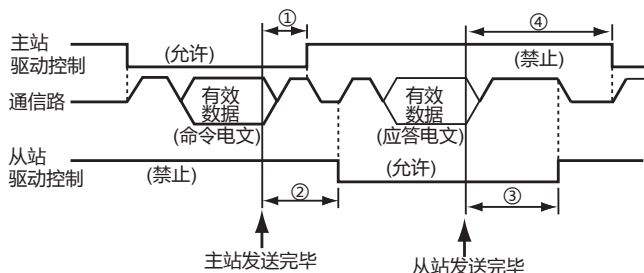
(请把主站的命令电文发送完毕后的应答监视时间设定成2s)

② 从站完成发送后到主站发送开始为止的时间 = 50ms以上

(从主站的应答电文接收完毕后到下一个发送开始为止,请等待50ms以上)

■ RS-485 驱动控制时间规格

由主站对RS-485的发送/接收直接控制的场合,请注意以下时序。



① 主站发送完毕 - 驱动不允许时间 = 500μs以下

② 从站接收完毕 - 驱动允许时间 = 30ms以上

③ 从站发送完毕 - 驱动不允许时间 = 10ms以下

④ 主站接收完毕 - 驱动允许时间 = 50ms以上

第 5 章 通讯数据地址一览

5 - 1 通讯数据使用的预备知识

■ 通讯数据的种类和形式

● 通讯数据的种类

通讯数据可分为以下几种类型。

- 机器种类关连数据
- 动作状态关连数据
- 瞬时流量关连数据
- 累计流量关连数据
- 功能设定关连数据
- 参数设定关连数据

● 通讯数据的形式

通讯数据的形式划分如下。

- 数值数据 : 表示数值的数据。(瞬时流量值等)
- 位数据 : 每个位都有含义的数据。
(报警等)
位数据需要在发送时合成, 接收时分解。

重要事项

- 通过通讯写入模式设定或参数设定时, 请勿通过本体操作键更改模式设定或参数设定。内部处理可能会冲突, 导致模式设定或参数设定无法正确执行。
- 对 EEPROM 的写入次数限制在 10 万次以下。如果需要通过通讯频繁重复写入数据, 请将目标内存设置为 RAM。

■ 通讯数据的存储内存

● 内存种类

通讯数据存储在机器的内存(记忆单元)中。本机可使用的内存有 2 种。

- RAM : 切断电源后数据消失, 但可反复写入。
- EEPROM : 切断电源后数据也不消失。但记忆单元的特性对写入次数有限制。
限制次数为 10 万次。

● 通讯对象内存

在通讯中，对上述2种内存，有必要按目的及用途进行读出/写入。
对象内存的差异如下。

- RAM : 仅对RAM进行读出/写入。对RAM写入后,电源切断后再通电时,EEPROM内的数据复制到RAM中,RAM的数据与EEPROM相同。
- EEPROM : 对RAM及EEPROM两者进行写入

■ 数据地址

数据地址的规定如下表所示。

通讯数据	RAM		EEPROM	
	偏移值	地址	偏移值	地址
机器种类关连数据	1000	1001 ~ 1199	4000	4001 ~ 4199
动作状态关连数据	1200	1201 ~ 1399	4200	4201 ~ 4399
瞬时流量关连数据	1400	1401 ~ 1599	4400	4401 ~ 4599
累计流量关连数据	1600	1601 ~ 1799	4600	4601 ~ 4799
功能设定关连数据	2000	2001 ~ 2199	5000	5001 ~ 5199
参数设定关连数据	2200	2201 ~ 2399	5200	5201 ~ 5399

■ 数据的单位・小数点位置

读出/写入数据中，不附加小数点。
单位或小数点位置由各数据决定。
有关各数据的单位・小数点位置，请参阅机器本体的使用说明书。

5 - 2 通讯数据地址一览

对各数据的地址及可读出/写入(R/W)状态的规定如下表所示。

R/W 栏记号的含义
○：可能
×：不可

■ 机器种类关连数据

项 目		数据范围	RAM			EEPROM			备 注
			地址	R	W	地址	R	W	
各系 列的 气 体 种 类	CMS SUS/SUS316	0: 氮气/空气 1: 氩气 2: 二氧化碳气 3: 氧气 4: 城市煤气 13A(46MJ) 5: 沼气 6: 丙烷 7: 丁烷 8: 用户设定 11: 城市煤气13A(45MJ)	1001	○	×	4001	×	×	气体种类设定的变 更可通过功能设定 (地址 2008、5008) 进行
	CMS 氢气・ 氦气对应产品	8: 用户设定 9: 氢气 10: 氦气							
未定义			1002	○	×	4002	×	×	
瞬时流量小数点 表示位置		0: 无小数点 1: xxxx. 2: xxx.x 3: xx.xx 4: x.xxx	1003	○	×	4003	×	×	
累计流量小数点 表示位置		0: 无小数点 1: xxxxxxxx. 2: xxxxxxx.x 3: xxxxxx.xx 4: xxxxx.xxx	1004	○	×	4004	×	×	
瞬时流量单位		0: mL/min 1: L/min	1005	○	×	4005	×	×	
累计流量单位		0: mL 1: L 2: m ³	1006	○	×	4006	×	×	

■ 动作状态关连数据

项 目	数据范围	RAM			EEPROM			备 注
		地址	R	W	地址	R	W	
报警状态	*2	1201	○	×	4201	×	×	状态用 10 进制数表示
事件状态	*3	1202	○	×	4202	×	×	状态用 10 进制数表示
未定义		1203	○	×	4203	×	×	
未定义		1204	○	×	4204	×	×	
累计流量下位	0 ~ 9999	1205	○	○	4205	○	○	与累计流量关连数据的地址 1603 及 4603 相同 *1
累计流量上位	0 ~ 9999	1206	○	○	4206	○	○	与累计流量关连数据的地址 1604 及 4604 相同 *1
瞬时流量	0 ~ 9999	1207	○	×	4207	×	×	与瞬时流量关连数据的地址 1401 相同

*1 对 CMF，不能写入

*2 报警状态位构成 (地址 1201) 0 : 正常 1 : 异常

位 No.	内 容
0	ALHI 超量程
1	未定义 (通常为 0)
2	未定义 (通常为 0)
3	未定义 (通常为 0)
4	ERR1 传感器异常
5	ERR2 调整数据异常
6	ERR3 传感器加热器异常
7	ERR4 传感器安全回路动作

*3: 事件状态位构成 (地址 1202) 0 : OFF 1 : ON

位 No.	内 容
0	事件输出 1 状态
1	事件输出 2 状态
2	未定义 (通常为 0)
3	外部接点输入状态
4	未定义 (通常为 0)
5	未定义 (通常为 0)
6	未定义 (通常为 0)
7	未定义 (通常为 0)

■ 瞬时流量关连数据

项 目	数据范围	RAM			EEPROM			备 注
		地址	R	W	地址	R	W	
瞬时流量	0 ~ 9999	1401	○	×	4401	×	×	
事件输出 1 瞬时流量设定	0 ~ 9999	1402	○	○	4402	○	×	与参数设定的地址 2201 及 5201 相同 *
事件输出 2 瞬时流量设定	0 ~ 9999	1403	○	○	4403	○	×	与参数设定的地址 2204 及 5204 相同 *

* 请用参数设定数据的地址向 EEPROM 执行写入。

■ 累计流量关连数据

项 目	数据范围	RAM			EEPROM			备 注
		地址	R	W	地址	R	W	
未定义		1601	○	×	4601	○	×	
未定义		1602	○	×	4602	○	×	
累计流量下位 4 位	0 ~ 9999	1603	○	○	4603	○	○	
累计流量上位 4 位	0 ~ 9999	1604	○	○	4604	○	○	
事件输出 1 累计流量设定 下位 4 位	0 ~ 9999	1605	○	○	4605	○	×	与参数设定的地址 2202 及 5202 相同 *1
事件输出 1 累计流量设定 上位 4 位	0 ~ 9999	1606	○	○	4606	○	×	与参数设定的地址 2203 及 5203 相同 *1
事件输出 2 累计流量设定 下位 4 位	0 ~ 9999	1607	○	○	4607	○	×	与参数设定的地址 2205 及 5205 相同 *1
事件输出 2 累计流量设定 上位 4 位	0 ~ 9999	1608	○	○	4608	○	×	与参数设定的地址 2206 及 5206 相同 *1
反向累计流量 初期值设定 下位 4 位	0 ~ 9999	1609	○	○	4609	○	×	与参数设定的地址 2211 及 5211 相同 *1
反向累计流量 初期值设定 上位 4 位	0 ~ 9999	1610	○	○	4610	○	×	与参数设定的地址 2212 及 5212 相同 *1

*1 请用参数设定数据的地址向 EEPROM 执行写入。

■ 功能设定关连数据

项 目	数据范围	RAM			EEPROM			备 注
		地址	R	W	地址	W	R	
键锁设定	0:无键锁 1:有键锁	2001	○	○	5001	○	○	
测量模式设定	0:仅瞬时流量 1:瞬时流量/累计流量 2:瞬时流量/反向累计流量	2002	○	○	5002	○	○	数据 1: 固定为瞬时流量 / 累计流量, 不可写入
事件输出 1 (EV1) 种类设定	0:不使用 1:瞬时流量上限 2:瞬时流量下限 3:累计流量计数增加 4:累计流量计数减少 5:串行输出 6:错误输出	2003	○	○	5003	○	○	累计计数增加、反向累计计数减少及累计输出脉冲的动作的测量模式被设定为 1 或 2 时有效 累计计数增加与反向累计计数减少不可进行组合设定
事件输出 2 (EV2) 种类设定	0:不使用 1:瞬时流量上限 2:瞬时流量下限 3:累计流量计数增加 4:累计流量计数减少 5:累计流量脉冲比率1 6:累计流量脉冲比率2 7:累计流量脉冲比率3	2004	○	○	5004	○	○	不可设定为反向累计计数减少 累计流量脉冲比率根据机器种类, 其值不同  请参阅各系列的使用说明书
ON 延迟设定 (EV1)	0:不使用 1:使用	2005	○	○	5005	○	○	仅当事件输出 1 的设定为 1 或 2 时有效
ON 延迟设定 (EV2)	0:不使用 1:使用	2006	○	○	5006	○	○	仅当事件输出 2 的设定为 1 或 2 时有效
事件待机设定	0:不使用 1:使用	2007	○	○	5007	○	○	仅当事件输出 1 或事件输出 2 的设定为 2 时有效

项 目		数据范围	RAM			EEPROM			备 注
			地址	R	W	地址	R	W	
各系列的气体种类选择	CMS SUS/SUS316	0: 氮气/空气 1: 氦气 2: 二氧化碳气 3: 氧气 4: 城市煤气 13A(46MJ) 5: 沼气 6: 丙烷 7: 丁烷 8: 用户设定 11: 城市煤气 13A(45MJ)	2008	○	○	5008	○	○	非氧气对应产品, 不能指定为氧气  请参阅各系列的使用说明书
	CMS 氢气・氦气对应产品	8: 用户设定 9: 氢气 10: 氦气							
模拟输出量程设定		0: 量程 1 1: 量程 2 2: 量程 3 3: 量程 4 4: 任意量程	2009	○	○	5009	○	○	根据机器种类, 量程 1 ~ 4 的值不同  请参阅各系列的使用说明书
模拟输出种类选择		0: 0 ~ 5V 1: 1 ~ 5V 2: 4 ~ 20mA	2010	○	○	5010	○	○	
流量换算基准温度		0 ~ 35	2011	○	○	5011	○	○	0 ~ 35 °C (1°C刻度)
低流量切除设定		0: 无低流量切除 1: 小于最小显示 * 2: 小于 1%FS 3: 小于 2.5%FS 4: 小于 5%FS	2012	○	○	5012	○	○	*根据机器种类, 最小显示的值不同  请参阅各系列的使用说明书
机器地址		0: 不使用通讯功能 1 ~ 99: 通讯地址	2030	○	×	5030	×	×	
传送速度选择		0: 9600bps 1: 4800bps 2: 2400bps	2031	○	×	5031	×	×	
通讯条件选择		0: 8 位数据、偶数校验、停止位 1 1: 8 位数据、无校验、停止位 2	2032	○	×	5032	×	×	

■ 参数设定关连数据

项 目	数据范围	RAM			EEPROM			备 注
		地址	R	W	地址	R	W	
事件输出 1 瞬时流量	0 ~ 9999	2201	○	○	5201	○	○	
事件输出 1 累计流量下位	0 ~ 9999	2202	○	○	5202	○	○	
事件输出 1 累计流量上位	0 ~ 9999	2203	○	○	5203	○	○	
事件输出 2 瞬时流量	0 ~ 9999	2204	○	○	5204	○	○	
事件输出 2 累计流量下位	0 ~ 9999	2205	○	○	5205	○	○	
事件输出 2 累计流量上位	0 ~ 9999	2206	○	○	5206	○	○	
事件输出 1 回差	0 ~ 100	2207	○	○	5207	○	○	
事件输出 2 回差	0 ~ 100	2208	○	○	5208	○	○	
事件输出 1 ON 延迟	0 ~ 60	2209	○	○	5209	○	○	单位 (S)
事件输出 2 ON 延迟	0 ~ 60	2210	○	○	5210	○	○	单位 (S)
反向累计流量 初期设定下位 4 位	0 ~ 9999	2211	○	○	5211	○	○	不可读出及写入
反向累计流量 初期设定上位 4 位	0 ~ 9999	2212	○	○	5212	○	○	不可读出及写入
用户设定 气体种类补偿系数	100 ~ 8000	2213	○	○	5213	○	○	根据功能设定的气体种类选择， 仅当选择了用户设定时，设定 值才有效 相当于 0.100 ~ 8.000
模拟输出 定标任意设定	10 ~ 250	2214	○	○	5214	○	○	

功能设定数据与各参数设定对应的场合，设定的数据才有效。

第 6 章 主站用通讯程序例

本章节所示程序例是用 Micro Focus 公司的 Windows 95/98/NT/2000 版 C++ Builder 5.0 或 Borland C ++ Compiler 5.5 编制的。

该程序仅作为客户编制程序时的参考，并不保证所有操作。Borland C ++ Compiler 5.5 可从 Micro Focus 公司网站下载。

■ 程序执行前

- 请确认机器的通讯条件、机器地址。
- 从站的最长应答时间为 2 秒。请把主站的应答监视时间设定为 2 秒。2 秒以内无应答时，请再次发送相同的电文。



- 命令电文的设备区分代码上使用了「X」时，应答电文的设备区分代码则为「X」。同样，命令电文上使用「x」时，应答电文也为「x」。
主站再次发送电文时，通过交替使用设备区分代码的「X」和「x」，可以方便的知道收到的应答电文是前次的还是本次的。

■ 执行程序

该程序执行数据的读出及数据的写入。

执行后，显示通讯的命令电文、应答电文的应用层。

```
命令 RS, 1000W, 2  
执行结果 00, 0, 0  
命令 WS, 1000W, 2  
执行结果 00
```

执行结果显示例

● 执行通讯的设定

调出 Open ()，初始化 RS-232C。

● 执行命令

Command 中设定要执行的文字列，调出 AppCPL()。

■ 数据读出/写入样板程序



使用上的注意事项

- 使用本程序例所产生的不良后果，本公司不负任何责任。

```
//-----
// C++ Builder 5
// Borland C++
// bcc32 cpl.cpp
//
// cygwin + gcc
// gcc cpl.cpp
//
#include <stdio.h>
#include <windows.h>
#pragma hdrstop

#define COMRESENDNUM 2
#define BUFFERSIZE 4096
#define TIMEOUT 2000

HANDLE handle;
unsigned long ErrorCode;
bool CheckSum;

bool AppCPL( char* tosend, char * received );
int Open( void );
int Close( void );
bool Write( unsigned char *Mesg, unsigned long Size );
bool Read( unsigned char *Buffer, unsigned long SizeToRead,
           unsigned long *ReadSize, unsigned long *ErrFlag );
void CPLSum( unsigned char *str, int len, unsigned char *buf );

//-----
int main(int argc, char* argv[])
{
    char command[255];
    char recieve[255];

    handle = (void *)0xffffffff;
    ErrorCode = 0;
    CheckSum = true;

    if(Open()==0){
        strcpy(command,"RS,1000W,2");
        AppCPL(command,recieve);
        printf("命令:%s\n",command);
        printf("执行结果:%s\n",recieve);

        strcpy(command,"WS,1000W,2");
        AppCPL(command,recieve);
        printf("命令:%s\n",command);
        printf("执行结果:%s\n",recieve);
        Close();
        getchar();
    }
    return 0;
}

//-----
int Open(void)
{
    COMMTIMEOUTS Timeouts;
    _DCB DCB;

    handle = CreateFile( "\\\\.\\COM1", GENERIC_READ|GENERIC_WRITE,
                        0, 0, OPEN_EXISTING, FILE_ATTRIBUTE_NORMAL, 0 );
    if( handle==(void *)0xffffffff ) return 3;

    if( !SetupComm( handle, BUFFERSIZE, BUFFERSIZE ) ){
        CloseHandle( handle );
        handle = (void *)0xffffffff;
        return 4;
    }
}
```

```

if( !PurgeComm( handle, PURGE_TXABORT|PURGE_RXABORT|
                PURGE_TXCLEAR|PURGE_RXCLEAR ) ){
    CloseHandle( handle );
    handle = (void*)0xffffffff;
    return 5;
}

Timeouts.ReadIntervalTimeout      = 0xFFFFFFFF;
Timeouts.ReadTotalTimeoutMultiplier = 0;
Timeouts.ReadTotalTimeoutConstant = 0;
Timeouts.WriteTotalTimeoutMultiplier = 0;
Timeouts.WriteTotalTimeoutConstant = 0;

if( !SetCommTimeouts( handle, &Timeouts ) ){
    CloseHandle( handle );
    handle = (void*)0xffffffff;
    return 6;
}

if( !GetCommState( handle, &DCB ) ){
    CloseHandle( handle );
    handle = (void*)0xffffffff;
    return 7;
}

DCB.BaudRate=CBR_9600;
DCB.fBinary=1;
DCB.fParity= 1;
DCB.ByteSize=0x08;
DCB.Parity=EVENPARITY;
DCB.StopBits=ONESTOPBIT;

if( !SetCommState( handle, &DCB ) ){
    CloseHandle( handle );
    handle = (void*)0xffffffff;
    return 8;
}

return 0;
}

int Close( void )
{
    if( handle!=(void *)0xffffffff ){
        if( !SetCommMask( handle, 0 ) ){
            CloseHandle( handle );
            handle = (void*)0xffffffff;
            return 2;
        }
        if( !EscapeCommFunction( handle, CLRDR ) ){
            CloseHandle( handle );
            handle = (void*)0xffffffff;
            return 3;
        }
        if( !CloseHandle( handle ) ) return 4;
        handle = (void*)(0xffffffff);
    }else{
        return 1;
    }

    return 0;
}

//-----
bool Write( unsigned char *Msg, unsigned long Size )
{
    unsigned long    Error;
    unsigned long    Errors;
    unsigned long    SizeWritten;
    COMSTAT          Stat;

    if( handle==(void *)0xffffffff ){
        return false;
    }
    if( Size>BUFFERSIZE ){
        printf( " 发送数据太长 " );
        return false;
    }
}

```

```

do{
    ClearCommError( handle, &Errors, &Stat );
}while( BUFFERSIZE < Stat.cbOutQue + Size );

if( WriteFile( handle, Mesg, Size, &SizeWritten, 0 )==false ){
    if( GetLastError()==ERROR_IO_PENDING ){
        while( GetOverlappedResult( handle, 0, &SizeWritten, true ) ){
            Error = GetLastError();
            if( Error==ERROR_IO_INCOMPLETE ){
                continue;
            }else{
                ClearCommError( handle, &Errors, &Stat );
                break;
            }
        }
    }else{
        ClearCommError( handle, &Errors, &Stat );
    }
}

if( Size==SizeWritten ){
    return true;
}else{
    return false;
}
}

//-----
bool Read( unsigned char *Buffer, unsigned long SizeToRead,
           unsigned long *ReadSize, unsigned long *ErrFlag )
{
    BOOL          rt;
    unsigned char  RecvMesg[BUFFERSIZE];
    unsigned char  *cptr;
    unsigned long  i;
    unsigned long  stime;
    unsigned long  dtime;
    unsigned long  Error=0;
    unsigned long  Size;
    COMSTAT        Stat;

    *ReadSize = *ErrFlag = dtime = 0;

    if( handle==(void *)0xffffffff ){
        return false;
    }

    stime = GetTickCount();
    do{
        ClearCommError( handle, &Error, &Stat );
        if( Stat.cbInQue>0 ){
            ZeroMemory( RecvMesg, BUFFERSIZE );
            rt = ReadFile( handle, RecvMesg, Stat.cbInQue, &Size, 0 );
            RecvMesg[Stat.cbInQue] = 0x00;
            if( !rt ){
                // 清除错误标志
                ClearCommError( handle, &Error, &Stat );
                // 设置引数错误标志
                *ErrFlag = Error;
                return false;
            }
        }

        for( cptr=&RecvMesg[0],i=0 ; cptr<&RecvMesg[Stat.cbInQue] ; cptr++,i++ ){
            if( SizeToRead==0xffffffff ){
                if( *cptr==0x02 ){
                    *ReadSize = 0;
                    Buffer[( *ReadSize )++] = *cptr;
                }else{
                    if( (*cptr=='\n') ){
                        Buffer[( *ReadSize )++] = cptr[0];
                        if( Buffer[( *ReadSize )-2]=='\r' ){
                            Buffer[( *ReadSize )] = 0x00;
                            goto OutOfWhile;
                        }
                    }
                }
                Buffer[( *ReadSize )++] = *cptr;
            }
        }
    }
}

```

```

        }else{
            Buffer[(*ReadSize)++] = *cptr;
            if( (*ReadSize)>=SizeToRead ){
                goto OutOfWhile;
            }
        }
    }
    Sleep(1);
    dtime = GetTickCount() - stime;
}while( (SizeToRead>*ReadSize) && (dtime<TIMEOUT) );

OutOfWhile:

    if( ((SizeToRead>*ReadSize)&&(SizeToRead!=0xFFFFFFFF))||(*ReadSize==0) ){
        *ErrFlag = 0x00010000;
        return false;
    }
    return true;
}
//-----
void CPLSum( unsigned char *str, int len, unsigned char *buf )
{
    int         num=0;
    unsigned char *ch;

    for( ch=&str[0] ; ch<&str[len] ; ch++ ) num += *ch;
    num = (~(num&0x000000FF)&0x000000FF);
    sprintf( (char*)buf, "%02X", num );
}
//-----

bool AppCPL( char* tosend, char * received )
{
    unsigned char  theMsg[BUFFERSIZE];
    unsigned char  theApp[BUFFERSIZE];
    unsigned char  theHdr[16];
    unsigned long  SzToSnd;
    unsigned long  RdSz;
    unsigned long  ErrFlg;
    bool           rt;
    int            Cnt=0;
    unsigned char  Sum[4];
    char           *etx;

    if( handle==(void*)(0xFFFFFFFF) ){
        return false;
    }

    ZeroMemory( theMsg, BUFFERSIZE );
    sprintf( (char*)theHdr, "%x02Yx30Yx31Yx30Yx30Yx58" );
    sprintf( (char*)theMsg, "%s%sYx03", theHdr, tosend );

    if( CheckSum ){
        SzToSnd = strlen( (char*)theMsg );
        CPLSum( theMsg, SzToSnd, &theMsg[SzToSnd] );
    }
    strcat( (char*)theMsg, "YrYn" );
    SzToSnd = strlen( (char*)theMsg );

resend:
    Write( theMsg, SzToSnd );
    rt = Read( theMsg, 0xFFFFFFFF, &RdSz, &ErrFlg );

    if( !rt ){
        if( Cnt++ < COMRESENDNUM ){
            goto resend;
        }
        if( ErrFlg ){
            ErrorCode = ErrFlg;
        }else{
            ErrorCode = 0x000f0000;
        }
        return false;
    }
}

```

```

    }else{
        if( strcmp( (char*)theMsg, (char*)theHdr, 6 )!=0 ){
            ErrorCode = 0x00020000;
            if( Cnt++<COMRESENDNUM ){
                goto resend;
            }
            return false;
        }
        if( CheckSum ){
            CPLSum( theMsg, RdSz-4, Sum );
            if( (theMsg[RdSz-4]!=Sum[0])|| (theMsg[RdSz-3]!=Sum[1]) ){
                ErrorCode = 0x00040000;
                if( Cnt++<COMRESENDNUM ){
                    goto resend;
                }
                return false;
            }
        }
        ZeroMemory( theApp,BUFFERSIZE );
        CopyMemory( theApp, &theMsg[6], RdSz-6 );
        etx = strstr( (char*)theApp, "%x03" );

        if( etx==NULL ){
            ErrorCode = 0x00200000;
            if( Cnt++<COMRESENDNUM ){
                goto resend;
            }
            return false;
        }else{
            *etx = 0x00;
        }
        strcpy( received, (char*)&theApp[0] );
    }
    return true;
}
//-----

```

第 7 章 故障处理

■ 不能通讯时的确认项目

- ① 装置电源是否合上。
- ② 接线有无错误。
- ③ 主站与本机的通讯条件设定是否一致。
下记设定项目只要有一个不同,就不能通讯。
下划线部分表示在本机侧可设定。
 传送速度 : 9600、4800、2400bps
 数据长 : 8
 校验 : 无校验、偶数校验
 停止位 : 1 位、2 位
- ④ 主站发送的命令电文的机器地址与本机设定的地址是否一致。
出厂时,本机的机器地址设为 0,即使命令电文的机器地址设定为 00(30H、30H),本机也不应答。
- ⑤ 多分支连接的机器,是否设定了不同的机器地址。
- ⑥ 通讯时间与  4-5 时间规格 (4-9 页)是否相符。
- ⑦ 设备区分代码(本机的场合,“X”或“x”)以外的字符串代码,请全部使用大写字母。

第 8 章 通讯规格

名 称	备 注
传送形态	平衡形
传送线路	3 线式
传送速度 (bps)	9600、4800、2400
传送距离	500m 以下 (但与 MA500DIM、CMC410 连接时为 300m)
通讯方式	半双工
同步方式	调步同期
数据形式	数据 8 位 偶数校验 1 停止位 数据 8 位 无奇偶校验 2 停止位
错误检测	校验检查、校验和
机器地址	0 ~ 99 (0 的场合通讯功能无效)
连接形式	1:N (31 台以下)
其他	基于 RS-485

附 录

■ 代码表

上位 下位	0	1	2	3	4	5	6	7
0			空格	0	@	P	、	p
1			!	1	A	Q	a	q
2	STX		"	2	B	R	b	r
3	ETX		#	3	C	S	c	s
4			\$	4	D	T	d	t
5			%	5	E	U	e	u
6			&	6	F	V	f	v
7			'	7	G	W	g	w
8			(	8	H	X	h	x
9			)	9	I	Y	i	y
A	LF		*	:	J	Z	j	z
B			+	;	K	[	k	{
C			,	<	L	¥	l	
D	CR		-	=	M	]	m	}
E			.	>	N	^	n	~
F			/	?	O	_	o	

的部分不可用于通信系。
(使用的代码因机器而异)

改版履历

印刷年月	资料编号	种 类	改订页	改订内容
07-12	CP-SP-1184C	初 版		
12-04		第2版		变更公司名
16-11		第3版	1-1、附-2 1-1	WORD → INDESIGN处理 由 CMC10L 变更到 CMC10L001A000 ■构成例 计算机 → 主站 (PC) 本机 → 从站 (本机)
24-02		第4版	封面 iii、2-1 1-1、2-2 4-8 5-1 5-2、5-7 5-5 5-6、5-8 附2-附4	删除部分图形 删除“医疗用气体仪表 CMC0152/0155”说明书 删除产品名称“CMC10L001A000” 部分内容变更 “重要事项”的内容变更 删除“CMS1500”和“CMF”的内容 删除注释*2 删除CMF 删除页面



本资料所记内容如有变更恕不另行通知

阿自倍尔株式会社
Advanced Automation Company

阿自倍尔自控工程（上海）有限公司

总 部 上海市徐汇区宜山路 700 号 B2 栋 8 楼

邮编：200233

电话：021-50905580

传真：021-50909810

<https://acn.azbil.com.cn/>

2024 年 2 月 中文第 4 版 日文第 7 版