



感热式涡流流量计 MVF050/080/100/150

使用说明书 通讯功能篇



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

对于承担使用本产品通讯功能的操作盘、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。

阿自倍尔仪表(大连)有限公司

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”。
<https://www.azbil.com/cn/products/factory/order.html>

重要事项

通讯中频繁变更本产品的参数的场合，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 1 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。

©2013-2020 Azbil Corporation. All Rights Reserved.

μ F™ 是阿自倍尔株式会社的注册商标。

安全注意事项

■ 关于图示

为了避免给您及他人造成人身损害及财产损失，请务必遵守本使用说明书中记述的安全注意事项。

本书使用各种图示。

其含义表示如下。请在理解的基础上仔细阅读本使用说明书。

■ 图示例

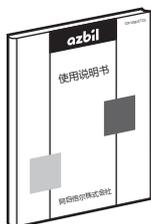
	本符号表示使用上必须“注意”的内容。
	本符号表示必须“禁止”的内容。
	本符号表示必须执行的“指示”内容。

注意

	请务必在切断供给电源的情况下对本产品进行接线。 否则可能会产生故障。
	请勿对本产品进行分解。 否则可能会产生故障。
	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。
	请勿让断线头、铁粉、水进入机箱内。 否则有产生误动作或故障的危险。
	请按规格记载的力矩可靠地紧固端子螺丝。 紧固不充分会引起故障。
	请勿把本机未使用的端子作为中继端子使用。 否则会引起故障。

本使用说明书的定位

与感热式涡流流量计相关的使用说明书共分 2 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。



感热式涡流流量计 MVF050/080/100/150 使用说明书

资料编号 CP-UM-5380CD

初次使用本机的用户、负责硬件设计、维护的人员，请务必阅读。本书对硬件包括的内容、产品的概要、与本机组合使用的产品群有哪些、安装在装置上的设置及配线`方法、维护点检、故障时的处理、硬件的规格进行说明。



感热式涡流流量计 MVF050/080/100/150 使用说明书 通讯功能篇

资料编号 CP-SP-1183CD

本书。
使用本机通讯功能的用户务必阅读。本书对通讯的概要、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的说明。

本使用说明书的标记

本书中使用了如下的记号及标记方法进行说明。

 使用上的注意事项：表示在使用时敬请注意的事项。

 参考：表示知道该项内容后易于理解。

：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。

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、写入与讯号的时间的说明。

第 5 章 通讯数据一览

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 规 格

本机的通讯规格的说明。

附 录

代码表。

目 录

安全注意事项
本使用说明书的定位
本使用说明书的构成
本使用说明书的标记

第 1 章 概 要 1-1

第 2 章 接 线 2-1

第 3 章 设 定 3-1

第 4 章 通讯步骤

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
■ 具体例	4-2
■ 数据地址的概念	4-2
4 - 2 数据链层	4-3
■ 数据链层的说明	4-3
4 - 3 应用层	4-6
■ 应用的概要	4-6
4 - 4 数据的读出	4-7
■ 读出命令的说明	4-7
■ 读出应答	4-8
■ 10 进制值的表现 (数值数据)	4-9
4 - 5 数据的写入	4-10
■ 写入命令的说明	4-10
■ 写入应答	4-11
4 - 6 结束代码	4-12
■ 正常结束及警告结束	4-12
■ 异常结束	4-12
4 - 7 时间规格	4-13
■ 命令电文、应答电文时间规格	4-13
■ RS-485 驱动控制时间规格	4-13
■ 其它注意事项	4-13

第 5 章 通讯数据一览

5 - 1 通讯数据使用的予备知识	5-1
■ 通讯数据的种类及形式	5-1
■ 通讯数据的存储内存	5-1

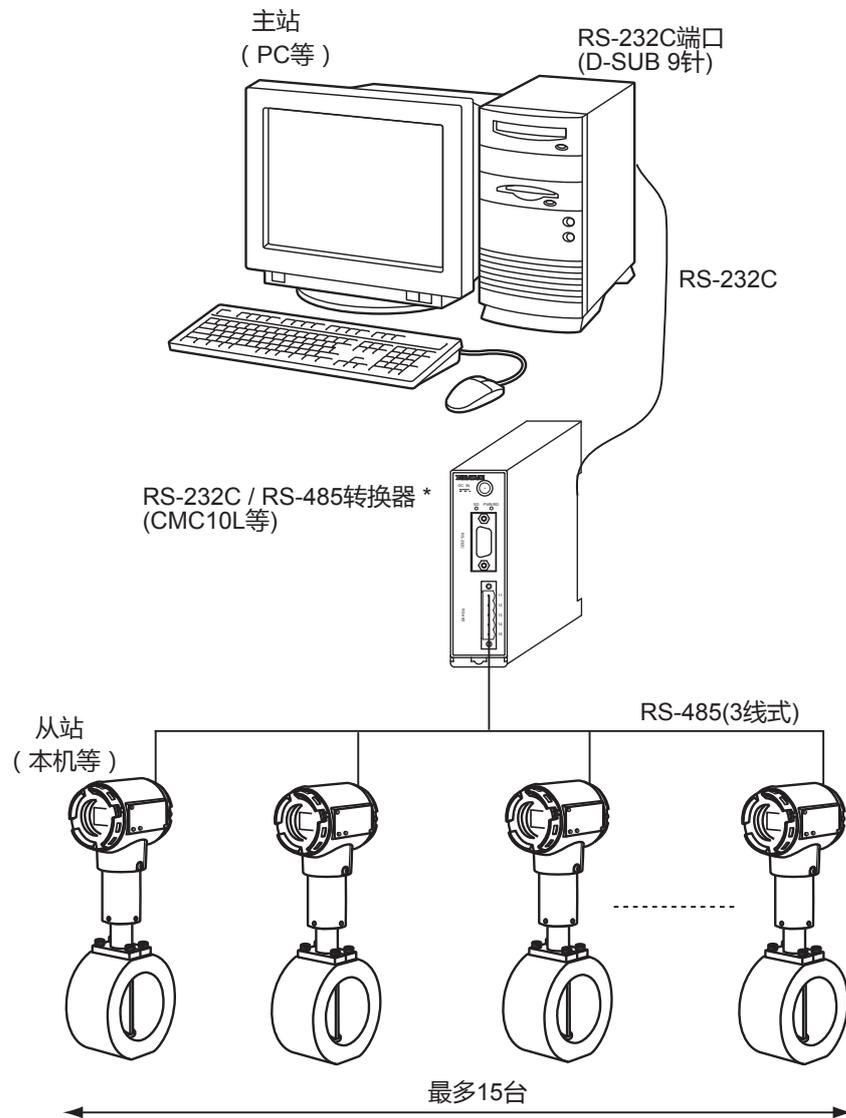
■ 数据地址	5-2
■ 读出 / 写入数据数	5-2
■ 数据的单位 · 小数点位置	5-2
5 - 2 通讯数据一览	5-3
■ 机器种类关连数据	5-3
■ 动作状态关连数据	5-4
■ 累积流量关连数据	5-5
■ 功能设定关连数据	5-6
■ 参数设定关连数据	5-8
第 6 章 主站用通讯程序	
6 - 1 编制时的注意事项	6-1
6 - 2 通讯程序例	6-2
■ 程序执行前	6-2
■ 执行程序	6-2
■ 数据读出 / 写入样板程序	6-3
第 7 章 故障处理	
■ 不能通讯时的确认项目	7-1
第 8 章 规格	8-1
附 录	
■ 代码表	附 -1
■ 与 CMC10L001A000 的连接	附 -2

第1章 概要

本机(也叫从站)可经由RS-232C/RS-485转换器与计算机或PLC等上位机(也称为主站)通讯,进行设定值或数据等数据交换。1台主机经由RS-485通讯可最多与15台(含其它机器最多31台)从站连接。使用「机器地址」以便确定要通讯的从站。

此外,为了能够从主站向各从站进行设定的写入、状态的读出等,请由客户编制通讯程序。

- 下述步骤成立时,可进行对方机器的各种数据的读出及写入。
 - ①主站(计算机)向从站(本机)发送命令电文
 - ②主站接收来自从站的应答电文
- 主站向从站发送的命令有「读出」及「写入」的2种。
- 读写的数据种类可按「数据地址」自由选择。



* : RS-232C / RS-485转换器可使用本公司产CMC10L001A000。

第2章 接 线

注意



请务必切断电源后再进行接线作业。
否则可能产生故障。



请不要分解本机。
否则有产生故障的可能。



请按规定的标准、指定的电线及施工方法进行正确配线。
否则可能会产生故障。



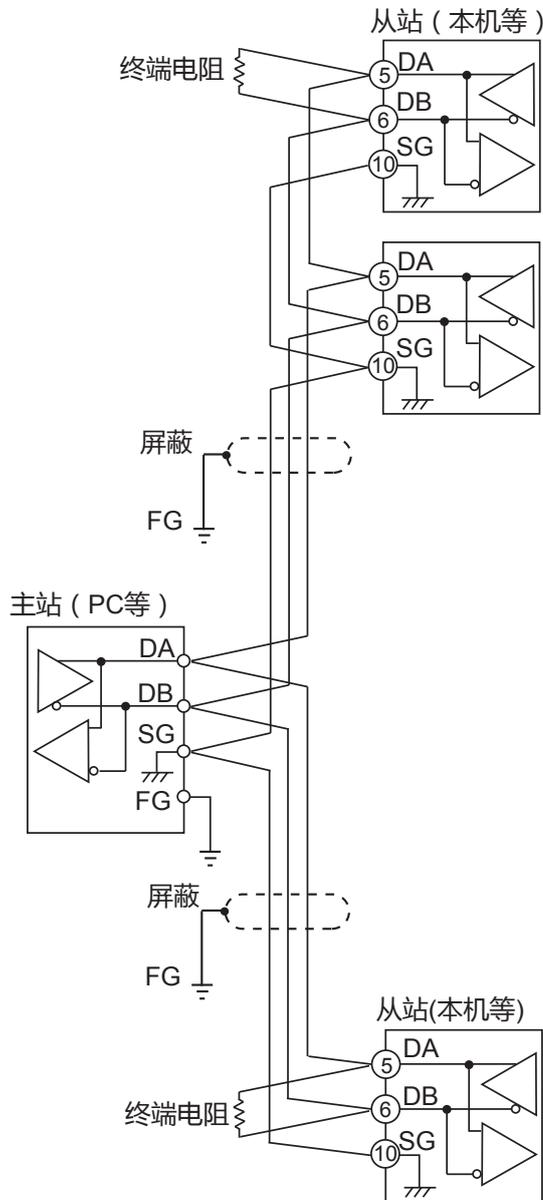
请勿让断线头、铁粉、水进入机箱内。
否则有产生误动作、故障的危险。

参 考

• 关于 RS-485 通讯线以外的接线方法，请参阅

 感热式涡流流量计 MVF050/080/100/150 使用说明书 CP-UM-5380CD

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。
作为主站的转换器，可使用本公司的产品 CMC10L001A000。

! 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。

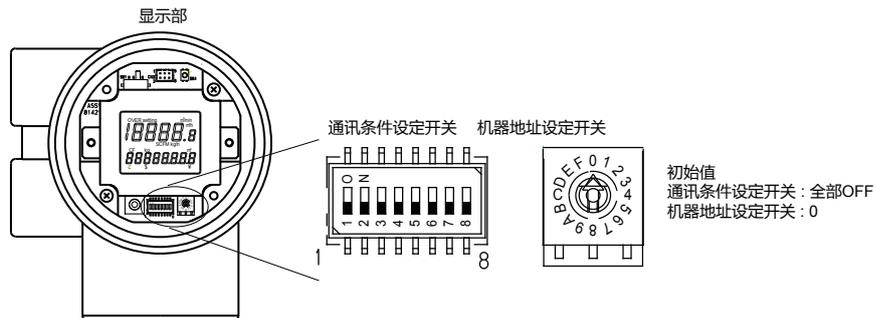
第3章 设定

⚠注意

❗ 设定请务必在切断电源后再进行设定。
否则可能会引起故障。

为了使 RS-485 通讯功能动作，请事先按以下步骤根据主站的设定对本机的通讯条件及机器地址进行设定。

● 设定方法



❗ 使用上的注意事项

- 设定的场合，请务必把通讯条件设定开关的「1」置为 ON。在 OFF 状态下设定无效。



① 传送速度设定

请把通讯条件设定开关的「2」、「3」按下表设定。

传送速度 (bps)	通讯条件设定开关	
	2	3
19200	OFF	OFF
9600	ON	OFF
4800	OFF	ON
2400	ON	ON

② 通讯条件设定

请把通讯条件设定开关的「4」按下表设定。

通讯条件	通讯条件设定开关 4
数据 8 位 1 停止位 偶数校验	OFF
数据 8 位 2 停止位 无校验	ON

③ 机器地址设定

请用机器地址设定开关把机器地址设定为 1 ~ 15(1 ~ F) 的范围。设定为 0 时，RS-485 通讯功能不起作用。此外，含其它机器在内连接台数大于 16 台的场合，请在机器侧设定。

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

通讯步骤可用下述最简单的表述方法。

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。

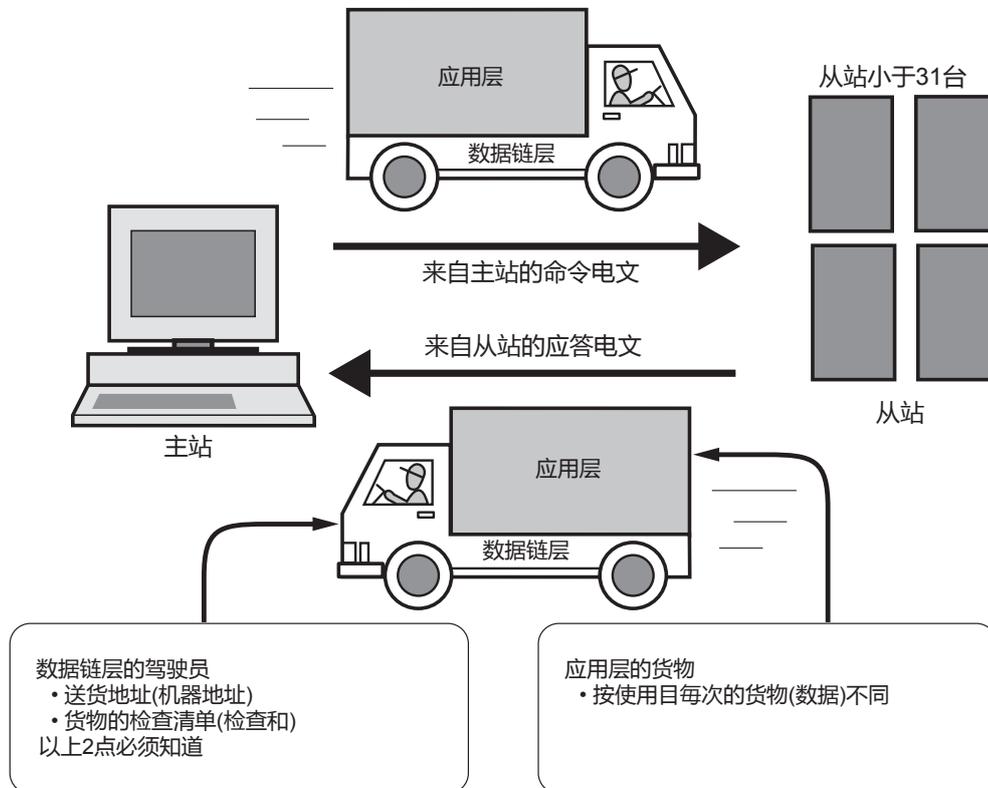
■ 电文的构成

一个电文由如下的 2 层构成。

从主站的命令电文及从站的应答电文都相同。

- 数据链层
 - 含有通讯所必要的基本信息的层。
 - 含有通讯电文的去向及电文的检查信息。
- 应用层
 - 写入数据的层。
 - 内容因使用目的而异。

各层的详细说明如下。



■ 具体例

电文的具体内容如下。

● 读出命令的场合

• 命令电文



• 应答电文



● 写入命令的场合

• 命令电文



• 应答电文



数据链层、应用层的详细说明如下项。

■ 数据地址的概念

本机使用称为数据地址的概念。数据的写入是与对各数据的地址相对应。

数据 A	1001W
数据 B	1001W
数据 C	1001W
⋮	⋮

有关实际的数据与数据地址的对应关系，请参阅

👉 5-2 通讯数据一览 (5-3 页)。

4 - 2 数据链层

■ 数据链层的说明

- 数据链层含有发送电文所需的 8 个基本信息。
- 数据链层中命令电文与应答电文的构造相同。



数据链层的各功能如下。

● STX(Start of TeXt)

◆功能 : 表示电文的起始。

- ◇说明
- 02H 固定。
 - 当机器收到「STX」时，即使在电文的中途，也判断为命令电文的第 1 文字。

● 机器地址

◆功能 : 指定送信目标的机器。可与指定的 1 台机器进行通讯。

- ◇说明
- 指定 0 的机器其通讯功能将停止。所以，要进行通讯的场合，请务必设定大于 1 以上的值。
 - 16 进制 2 文字。详细内容请参阅例子。

□例 : 通讯对方的机器地址为 10 的场合

- ① 10(10 进制)=0AH(16 进制)
- ② 转换成字符代码
0=30H
A=41H
- ③ 由②求出的「0A」(30H、41H)作为机器地址使用

📖 参考

设定的详细内容请参阅
👉第 3 章 设定。

❗ 使用上的注意事项

- 与应用层的数据地址的功能完全不同。请注意其差别。

● 子地址

◇说明 : 对本机固定。与机器地址的形式相同,请务必设定为「00」(30H、30H)。

● 设备区分代码

◇说明 : 本机只可指定字符代码「X」(58H)或「x」(78H)。

● ETX (End of TeXt)

◆功能 : 在此之前表示应用层。

◇说明 • 03H 固定。

● 检查和

◆功能 : 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。。

◇说明 • 16 进制 2 文字。

• 检查和的成方法

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 转换成字符代码

□例 : 以前页的命令电文例的场合进行说明。

① 从 STX 到 ETX 的电文按 1 字节单位进行加算的结果的下位 1 字节为 76H

② 加算结果取 2 的补码后为 8AH

③ 转换成字符代码。把其作为检查和

结果为「8A」,即(38H)与(41H)

字符代码的转换请参阅机器地址的例。

❗ 使用上的注意事项

• 命令电文的检查和不可省略。

● CR 与 LF (Carrige Return / Line Feed)

◆功能 : 表示电文的最后。

◇说明 · 「CR」为 (0DH)、 「LF」为 (0AH)。

· 请务必把 CR 及 LF 成对使用。

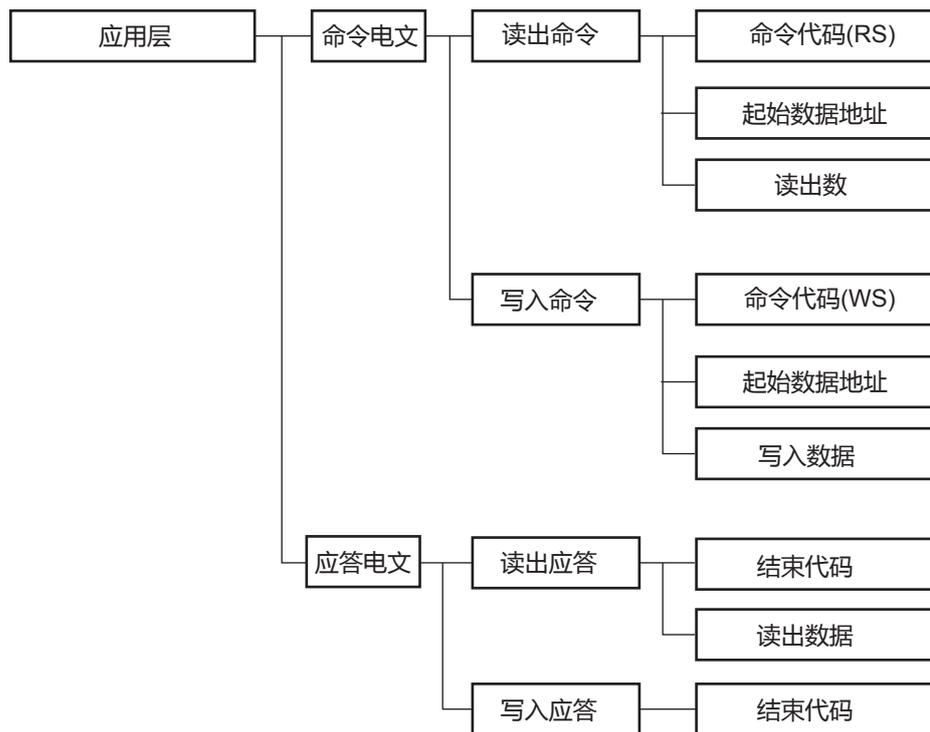
❗ 使用上的注意事项

- 数据链层的内容中发生如下异常的场所，机器将不会应答。
 - 与通讯条件不符 (传送速度不符、校验错误等)
 - STX、ETX、CR、LF 未在规定的位置
 - 设备区分代码不是 「X」 或 「x」
 - 机器地址、子地址、检查和不是 2 个文字
 - 检查和的计算与电文的检查和不符
 - 含有规定以外的文字
 - 发送的机器地址与对象机器的机器地址不同
 - 机器地址为 「00」
- 除检查和外，数据链层的内容设定了与机器的命令电文相同的应答电文。
- 机器地址及检查和所使用的 16 进制的数值部分中，请使用 「A」 ~ 「F」的大文字。

4 - 3 应用层

■ 应用的概要

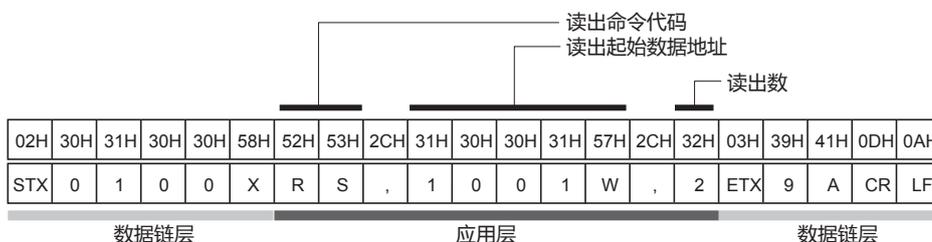
- 应用层中含有命令、数据、数据数及电文的判定信息 (结束代码)。
- 应用层的命令电文与应答电文的构造不同。
- 命令电文中有「读出命令」与「写入命令」。
应答电文根据各自的命令产生相应的应答。
- 命令电文根据结束代码可判断该进行如何的处理。



4 - 4 数据的读出

■ 读出命令的说明

- 从指定的读出起始数据地址起把连续的数据地址的内容用 1 个电文读出的命令。
- 读出命令的应用层由如下的 3 种数据构成。



- 各数据间用逗号「,」(字符代码的 2CH) 区别。
- 应用层的各数值及文字全部用大写字符代码。
- 各数值用 10 进制。
- 各数据中不可附加多余的「0」及空格。
 - 例：「RS,01001W,2」的下划线部不可。
 - 例：「RS, 1001W,02」的下划线部不可。
 - 例：上图是把从 1001W 起的 2 个数据的信息用 1 个电文读出的例。

● 读出命令代码 (RS)

- ◆功能：表示读出的命令。
- ◇说明
 - RS
 - (52H、53H) 的 2 文字。

● 读出起始数据地址

- ◆功能：指定开始读出的起始数据地址。
- ◇说明
 - 数据地址与读出数据的对应，请参阅 5-2 通讯数据一览 (5-3 页)。
 - 数据地址的数值后必须附加「W」(57H)。

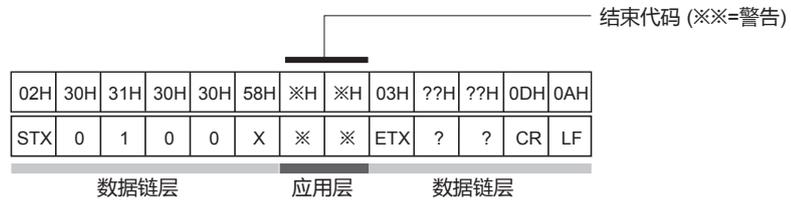
● 读出数据数

- ◆功能：指定的数据地址读出连续的几个数据。
- ◇说明
 - 1 个电文可读出的数据数量有限制。详见 ■ 读出 / 写入数据数 (5-2 页)。

● 异常应答

命令电文中有异常、不能正常读出时，将返回异常应答。

下图示异常应答的例。(※※ 是异常代码对应的数值)



■ 10 进制值的表现 (数值数据)

◆规定：数据地址的数值部、读出数、写入值、读出数据全部依据以下规定。

① 数值为负の場合，数值前附加负记号「-」(2DH)。

□例：「- 123」(2DH、31H、32H、33H)

② 数值为 0 の場合，按 1 个 0 处理。

□例：「0」(30H)

□例：「00」(30H、30H) 不可

③ 数值为正的場合，数值前不附加正记号「+」。

□例：「+ 123」(2BH、31H、32H、33H) 不可

④ 数值前不附加多余的 0 或空格。

□例：「0123」(30H、31H、32H、33H) 不可

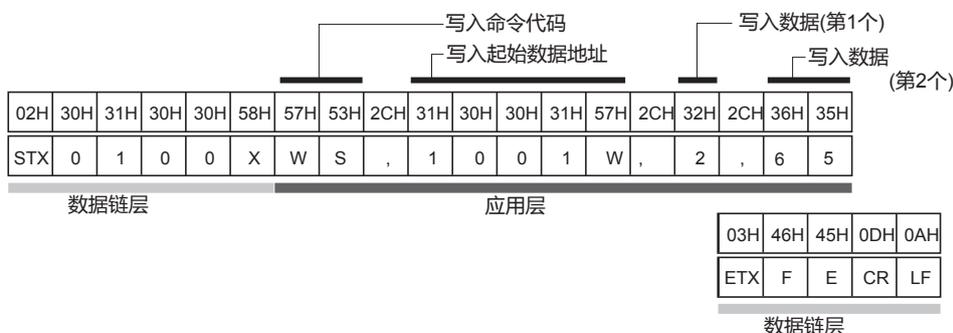
□例：「 123」(20H、31H、32H、33H) 不可

4 - 5 数据的写入

■ 写入命令的说明

从指定的写入起始数据地址起、把连续的数据地址的内容用 1 个电文同时进行多个写入的命令。

写入命令的应用层由如下 3 种数据构成。



各数据间用「,」(字符代码的 2CH) 区别。

不必指定写入数据的个数。

应用层的各数值及文字全部使用大写字母代码。

各数值使用 10 进制。

各数据中不可附加多余的「0」(30H) 及空格。

□例：「WS,01001W,2」的下划线部不可

□例：「WS,1001W,02」的下划线部不可

□例：上图是用 1 个电文向 1001W 地址及 1002W 地址分别写入 2 与 65 的例

● 写入命令代码 (WS)

◆功能：表示写入的命令。

◇说明 「WS」(57H、53H) 的 2 文字。

● 写入起始数据地址

◆功能：指定开始写入的起始数据地址。

• 数据地址与写入数据的对应关系，请参阅

➡ 5-2 通讯数据一览 (5-3 页)。

• 数据地址的数值后必须附加「W」(57H)。

● 写入数据

- ◆功能 : 从指定的数据地址起向连续的数据地址中写入数据。
- ◇说明
 - 能写入的数值的范围因各数据地址而异。
 - 各数据用 (2CH) 区别。
 - 各数据对应的写入数据地址、起始数据地址起按顺序每次加 1 后的值。(参照前页的例)
 - 1 个电文可写入的数据的数量有限制。详见 ■ 读出 / 写入数据数 (5-2 页)。

■ 写入应答

● 结束代码

- ◆功能 : 数据链层的电文不正确的场合, 只返回结束代码。
- ◇说明 结束代码的分类如下。



● 正常应答

- ◆功能 : 写入正常结束的场合, 返回正常应答。
- 例 : 正常应答的例 (全部的数据已正确写入)

											结束代码 (00=正常)					
02H	30H	31H	30H	30H	58H	30H	30H	03H	38H	32H	0DH	0AH				
STX	0	1	0	0	X	0	0	ETX	8	2	CR	LF				
数据链层						应用层		数据链层								

● 警告应答

下图中是警告应答时的例。(※※ 是异常代码对应的数值)

											结束代码 (※※=警告)					
02H	30H	31H	30H	30H	58H	※H	※H	03H	??H	??H	0DH	0AH				
STX	0	1	0	0	X	※	※	ETX	?	?	CR	LF				
数据链层						应用层		数据链层								

● 异常应答

- ◆功能 : 命令电文有异常、未进行写入处理的场合, 返回异常应答。
- 例 : 异常应答时 (※※ 为异常代码对应的数值)

											结束代码 (※※=异常)					
02H	30H	31H	30H	30H	58H	※H	※H	03H	??H	??H	0DH	0AH				
STX	0	1	0	0	X	※	※	ETX	?	?	CR	LF				
数据链层						应用层		数据链层								

4 - 6 结束代码

■ 正常结束及警告结束

全部的写入数据已写入的场合则为正常结束代码。

此外，当存在已写入的数据及未写入的数据的场合，则为警告结束代码。

结束代码	种 类	内容及动作
00	正常	正常结束
20	警告	数据数异常 (写入数据的个数不正确, 之前的写入处理执行)
21	警告	数据地址异常 (数据地址的中途或结束不正确, 之前的写入处理执行)
22	警告	数据范围异常 (写入命令时的数值数据的值超过正常范围, 之前的写入处理执行)
23	警告	根据机器条件不可写入 (本机内部的设定值的条件是不可写入)

■ 异常结束

没有写入 1 个数据的场合，则为异常结束代码。

结束代码	种 类	内容及动作
40	异常	数据数异常 读出数据数的值不正确或写入数据的个数不正确
41	异常	数据地址异常 (数据地址的起始不正确)
42	异常	数据范围异常 (写入命令的数值数据的值超出正常范围)
43	异常	根据机器条件全部数据不可写入 (本机内部的设定值的条件是不可写入)
99	异常	未定义命令 (应用层的起始 2 文字是未定义的命令)

4 - 7 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。

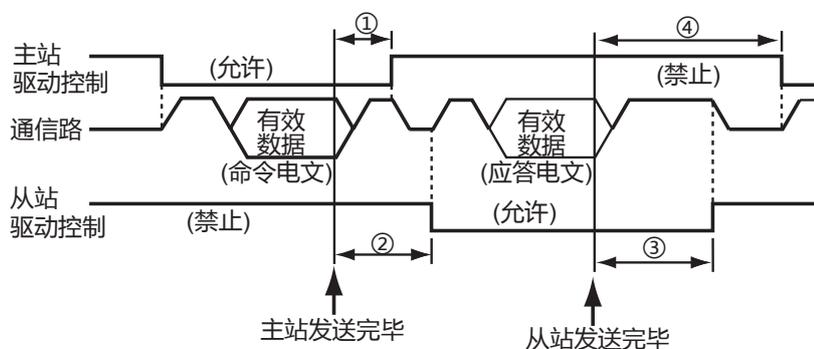
● 发送开始时间



- 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 = 10ms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10ms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 = 15ms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 = 10ms以上

■ 其它注意事项

- 从用1个电文进行读出 / 写入的数据数量越多，则从主站发送命令电文结束后起到从站开始发送应答电文为止的时间越长。
要使从站的应答快的场合，请用1个电文进行读出 / 写入并且减少数据数量。
- 用1个电文进行读出 / 写入的数据数为1个数据时，主站从发送命令电文结束后起到从站发送应答电文为止的时间约为30ms。

第 5 章 通讯数据一览

5 - 1 通讯数据使用的预备知识

■ 通讯数据的种类及形式

● 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 (瞬时流量及其它)
- 累积流量关连数据
- 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 : 显示数值的数据。
- 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

通讯中频繁变更本产品的参数的场合, 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 1 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。

■ 通讯数据的存储内存

● 内存种类

通讯数据存储于机器的内存 (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 1 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
累积流量关连数据	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

■ 读出/写入数据数

1次通讯可连续读出/写入的数据数如下表。

	RAM	EEPROM
读出	1 ~ 10 字	1 ~ 10 字
写入	1 ~ 10 字	1 ~ 10 字

■ 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。

5 - 2 通讯数据一览

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

显示	数据范围	RAM			EEPROM			备注
		地址	R	W	地址	R	W	
气体种类	0: 空气/窒氮气/氩气 1: 氧气 2: 二氧化碳 3: 城市煤气 13A(LNG)/甲烷 4: 丙烷 5: 丁烷 7: 用户设定气体	1001	○	×	4001	×	×	可通过功能设定 (地址 2001 及 5001) 变更气体种类
连接口径	0: 50A 1: 80A 2: 100A 3: 150A	1002	○	×	4002	×	×	—
瞬时流量单位	1: 0.1 2: 0.2 5: 0.5 10: 1.0	1003	○	×	4003	×	×	该值与瞬时流量值 (地址 1201) 之积为瞬时流量
累积流量小数点显示位置	0: XXXXXXXX (无小数点) 1: XXXXXXXX.X	1004	○	×	4004	×	×	上位 4 位 (地址 1603)+ 中位 4 位 (地址 1602) 的 8 位显示时的小数点位置

■ 动作状态关连数据

显示	数据范围	RAM			EEPROM			备注
		地址	R	W	地址	R	W	
瞬时质量流量值	0 ~ 65535	1201	○	×	4201	×	×	该值与瞬时流量单位 (地址 1003) 之积为瞬时流量值
瞬时体积流量值	MVF050: 0 ~ 3900 MVF080: 0 ~ 8600 MVF100: 0 ~ 13250 MVF150: 0 ~ 28500	1202	○	×	4202	×	×	该值乘以 0.1(m³/h) 后的值为瞬时体积流量值
温度	- 15 ~ + 60(°C)	1203	○	×	4203	×	×	-
压力	- 75 ~ + 1100(kPa)	1204	○	×	4204	×	×	-
错误状态位	*1	1205	○	×	4205	×	×	状态用 10 进制显示
报警状态位	*2	1206	○	×	4206	×	×	状态用 10 进制显示

*1 : 错误状态位构成 (地址 1205) 0: 正常 1: 错误发生

位 No.	内容
0	处理器异常
1	温度传感器异常
2	压力传感器异常
3	内存数据异常
4	未定义 (常时 0)
5	未定义 (常时 0)
6	未定义 (常时 0)
7	未定义 (常时 0)

*2 : 报警状态位构成 (地址 1206) 0: 正常 1: 报警发生

位 No.	内容
0	流量上限报警 (流速 45m/s 以上)
1	温度下限报警 (- 15 °C 以下)
2	温度上限报警 (60 °C 以上)
3	压力下限报警 (- 50kPa 以下)
4	压力上限报警 (1000kPa 以上)
5	未定义 (常时 0)
6	未定义 (常时 0)
7	未定义 (常时 0)

■ 累积流量关连数据

显示	数据范围	RAM			EEPROM			备注
		地址	R	W	地址	R	W	
累积流量下位 2 位 *1、*2	(BCD 数据) 0 ~ 99 m ³ 或 kg	1601	○	×	4601	×	×	上位 4 位 + 中位 4 位的 8 位显示时的小数点位置为地址 1004 的累积流量小数点位置
累积流量中位 4 位 *2	(BCD 数据) 0 ~ 9999 m ³ 或 kg	1602	○	×	4602	×	×	
累积流量上位 4 位 *2	(BCD 数据) 0 ~ 9999 m ³ 或 kg	1603	○	×	4603	×	×	
换算费用下位 4 位	(BCD 数据) 0 ~ 9999	1604	○	×	4604	×	×	累积流量上位 4 位 + 中位 4 位之和乘以费用换算系数 (地址 2211) 后的值
换算费用上位 4 位	(BCD 数据) 0 ~ 9999	1605	○	×	4605	×	×	
累积流量复位	0 ~ 1	1606	○	○	4606	×	×	数据读出时总返回 0 复位累积流量时请写入 1

*1: 当电源 OFF 时, 累积流量下位 2 位 (地址 1601) 不备份。

*2: 累积流量采用如下的形式。

显示位 (MVF050) (MVF050 以外)	10 ⁶	10 ⁵	10 ⁴	10 ³	10 ²	10 ¹	1	10 ⁻¹	10 ⁻²	10 ⁻³
	10 ⁷	10 ⁶	10 ⁵	10 ⁴	10 ³	10 ²	10 ¹	1	10 ⁻¹	10 ⁻²
通讯数据	上位 4 位地址 1603			中位 4 位地址 1602				下位 4 位 地址 1601		

显示部的初始设定是上位 4 位 + 中位 4 位的共计 8 位, 通过累积流量显示分辨率 (地址 2012 及 5012) 设定, 也可显示下位 2 位。

例) MVF080 时的累积流量为 12345678.90m³ 的场合

地址	数据 (BCD)
1601	90
1602	5678
1603	1234

■ 功能设定关连数据

显示	数据范围	RAM			EEPROM			备注
		地址	R	W	地址	R	W	
气体种类设定	0: 空气/氮气/氩气 1: 氧气 2: 二氧化碳 3: 城市气体 13A(LNG)/甲烷 4: 丙烷 5: 丁烷 7: 用户设定气体	2001	○	○	5001	○	○	氧气型号的机型以外时不可选择 1: 氧气 7: 由用户设定进行选择的情况, 为 气体比重(地址2207及5207)所 设定的值
温度压力 补偿模式设定	0: 无补偿 1: 只有温度补偿 2: 只有压力补偿 3: 有温度压力补偿	2002	○	○	5002	○	○	-
流量显示模式	0: m ³ /h 显示 1: kg/h 显示	2003	○	○	5003	○	○	-
未定义	0	2004	○	×	5004	○	×	*1
4 ~ 20mA 输出 模式设定	0: 瞬时质量流量 1: 瞬时体积流量 2: 温度 3: 压力	2005	○	○	5005	○	○	0: 选择瞬时质量流量的场合, 4mA、20mA 输出时设定质量流 量(地址 2209、2210 及 5209、 5210) 所设定的值为输出量程 1: 选择瞬时体积流量的场合, 由 体积流量输出量程(地址 2215 及 5215) 设定的值为输出量程
4 ~ 20mA 输出 断线设定	0: 量程上限 1: 量程下限	2006	○	○	5006	○	○	1: 选择量程上限的场合, 由断线 设定值(地址 2211 及 5211) 所 设定的值为输出值
未定义	0	2007	○	×	5007	○	×	*1
未定义	0	2008	○	×	5008	○	×	*1
累积脉冲 权重设定 (MVF050)	0: 0.01m ³ 或 0.01kg 1: 0.1m ³ 或 0.1kg 2: 1m ³ 或 1kg 3: 10m ³ 或 10kg	2009	○	○	5009	○	○	-
累积脉冲 权重设定 (MVF050 以外)	0: 0.1m ³ 或 0.1kg 1: 1m ³ 或 1kg 2: 10m ³ 或 10kg 3: 100m ³ 或 100kg							
LCD 上段显示 模式设定	0: 瞬时质量流量 1: 瞬时体积流量 2: 温度 3: 压力	2010	○	○	5010	○	○	-
LCD 下段显示 模式设定	0: 累积流量 1: 换算费用 2: 温度/压力 3: 累积流量及温度/压力 交替显示	2011	○	○	5011	○	○	-

*1: 对未定义地址读出时总为 0。另外, 向未定义地址写入时能正常应答, 但数据不可写入。

显示	数据范围	RAM			EEPROM			备注
		地址	R	W	地址	R	W	
累积流量显示分辨率 (MVF050)	0: XXXXXX.XXX 1: XXXXXX.XX 2: XXXXXXXX.X	2012	○	○	5012	○	○	-
累积流量显示分辨率 (MVF050 以外)	0: XXXXXX.XX 1: XXXXXXXX.X 2: XXXXXXXXX							
未定义	0	2013	○	×	5013	○	×	*1
通过单位模式设定	0: 日元 1: 美元 2: 欧元	2014	○	○	5014	○	○	-
用户设定温度	0: 测量值使用 1: 用户设定值使用	2015	○	○	5015	○	○	1: 选择用户设定值使用的场合, 则按用户设定温度(地址2216及5216)进行温度补偿
用户设定压力	0: 测量值使用 1: 用户设定值使用	2016	○	○	5016	○	○	1: 选择用户设定值使用的场合, 则按用户设定压力(地址2217及5217)进行压力补偿
未定义	0	2017	○	×	5017	○	×	*1
未定义	0	2018	○	×	5018	○	×	*1
未定义	0	2019	○	×	5019	○	×	*1
未定义	0	2020	○	×	5020	○	×	*1
未定义	0	2021	○	×	5021	○	×	*1
未定义	0	2022	○	×	5022	○	×	*1
未定义	0	2023	○	×	5023	○	×	*1
未定义	0	2024	○	×	5024	○	×	*1
未定义	0	2025	○	×	5025	○	×	*1
未定义	0	2026	○	×	5026	○	×	*1
未定义	0	2027	○	×	5027	○	×	*1
未定义	0	2028	○	×	5028	○	×	*1
未定义	0	2029	○	×	5029	○	×	*1
机器地址设定	0: 不使用通讯功能 1 ~ 15: 机器地址	2030	○	×	5030	○	×	设定方法请参阅第3章 设定
传送速度设定	0: 19200bps 1: 9600bps 2: 4800bps 3: 2400bps	2031	○	×	5031	○	×	
通讯条件选择	0: 8位数据 偶数校验 1 停止位 1: 8位数据 无校验 2 停止位	2032	○	×	5032	○	×	

*1: 对未定义地址读出时总为0。另外, 向未定义地址写入时能正常应答, 但数据不可写入。

■ 参数设定关连数据

显示	数据范围	RAM			EEPROM			备注
		地址	R	W	地址	R	W	
基准温度	0 ~ 35 °C	2201	○	○	5201	○	○	—
基准压力	90.0 ~ 300.0kPa	2202	○	○	5202	○	○	*1
大气压	90 ~ 110kPa	2203	○	○	5203	○	○	—
不感流量	(0 ~ 30 %FS)m ³ /h	2204	○	○	5204	○	○	数据范围为满量程流量 *2 乘以括号内的比率后的流量 (根据机型数据范围不同)
偏置流量	(-10 ~ + 10%FS)m ³ /h	2205	○	○	5205	○	○	数据范围为满量程流量 *2 乘以括号内的比率后的流量 (根据机型数据范围不同)
转换系数	0.100 ~ 9.999	2206	○	○	5206	○	○	*1
气体比重	0.01 ~ 99.99	2207	○	○	5207	○	○	气体种类设定 (地址 2001 及 5001) 选择为 7: 用户设定气体的场合, 设定值才有效 *1
费用换算系数	0.100 ~ 9.999	2208	○	○	5208	○	○	*1
4mA 输出时设定 质量流量	(0 ~ 99 %FS)m ³ /h	2209	○	○	5209	○	○	数据范围是满量程流量 *2 乘以括号内的比率后的流量 (根据机型数据范围不同) 仅当 4 ~ 20mA 输出模式设定 (地址 2005 及 5005) 选择为 0: 瞬时质量流量的场合, 设定值才有效
20mA 输出时设定 质量流量	(1 ~ 100%FS)m ³ /h	2210	○	○	5210	○	○	数据范围是满量程流量 *2 乘以括号内的比率后的流量 (根据机型数据范围不同) 仅当 4 ~ 20mA 输出模式设定 (地址 2005 及 5005) 选择为 0: 瞬时质量流量的场合, 设定值才有效
断线设定值	0 ~ 125%	2211	○	○	5211	○	○	仅当 4 ~ 20mA 输出断线设定 (地址 2006 及 5006) 选择 1: 量程上限的场合, 设定值才有效
未定义	0	2212	○	×	5212	○	×	*3
未定义	0	2213	○	×	5213	○	×	*3
未定义	0	2214	○	×	5214	○	×	*3
体积流量输出量程	10 ~ 150 %FS	2215	○	○	5215	○	○	仅当 4 ~ 20mA 输出模式设定 (地址 2005 及 5005) 选择 1: 瞬时体积流量的场合, 设定值才有效 体积流量满量程请参阅 *2
用户设定温度	- 15 ~ + 60 °C	2216	○	○	5216	○	○	仅当用户设定温度 (地址 2015 及 5015) 选择 1: 用户设定值使用的场合, 设定值才有效
用户设定压力	- 50 ~ + 1000kPa	2217	○	○	5217	○	○	仅当用户设定压力 (地址 2016 及 5016) 选择为 1: 用户设定值使用的场合, 设定值才有效

*1 :数据是除小数点外的值。

*2 :满量程流量

型号	质量流量满量程 m ³ /h 或 kg/h	体积流量满量程 m ³ /h
MVF050	8000	240
MVF080	16000	520
MVF100	24000	800
MVF150	48000	1700

*3 :未定义地址总读数为 0。另外, 向未定义地址写入时, 会有正常应答但不能写入。

第 6 章 主站用通讯程序

6 - 1 编制时的注意事项

- 机器的最长应答时间为 2 秒。所以请把应答监视时间设定为 2 秒。
- 2 秒以内没有应答的场合，请再次发送同一电文。
- 通讯中由于干扰的影响不能正确传送电文时，请务必再次进行上述电文的再发送。

参考

- 命令电的设备区分代码中使用「X」时，应答电文的设备区分代码也为「X」，同样，命令电文中使用「x」时应答电文也为「x」。
主站再次发送电文时，通过设备区分代码交替使用「X」及「x」，便于区分接收到的应答电文是前次的还是本次的。

6 - 2 通讯程序例

本章节所示程序例是用 Micro Focus 公司的 Windows95/98/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。

■ 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, CLRDRTR ) ){
            CloseHandle( handle );
            handle = (void*)0xffffffff;
            return 3;
        }
        if( !CloseHandle( handle ) ) return 4;
        handle = (void*)(0xffffffff);
    }else{
        return 1;
    }

    return 0;
}

//-----
bool    Write( unsigned char *Mesg, 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, "%x02%x30%x31%x30%x30%x58" );
    sprintf( (char*)theMsg, "%s%s%x03", theHdr, tosend );

    if( CheckSum ){
        SzToSnd = strlen( (char*)theMsg );
        CPLSum( theMsg, SzToSnd, &theMsg[SzToSnd] );
    }
    strcat( (char*)theMsg, "%r%n" );
    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( strncmp( (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 装置电源是否接通。
- 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：19200、9600、4800、2400bps
数据长：8、7 位
校验：无校验、偶数校验、奇数校验
停止位：1、2 停止位
- 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
👉 4-7 时间规格 (4-13 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 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)	19200、9600、4800、2400
传送距离	300m 以下
通讯方式	半双工
同步方式	调步同期
数据形式	数据 8 位 1 停止位 偶数校验 数据 8 位 2 停止位 无奇偶校验
错误检测	奇偶检查、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

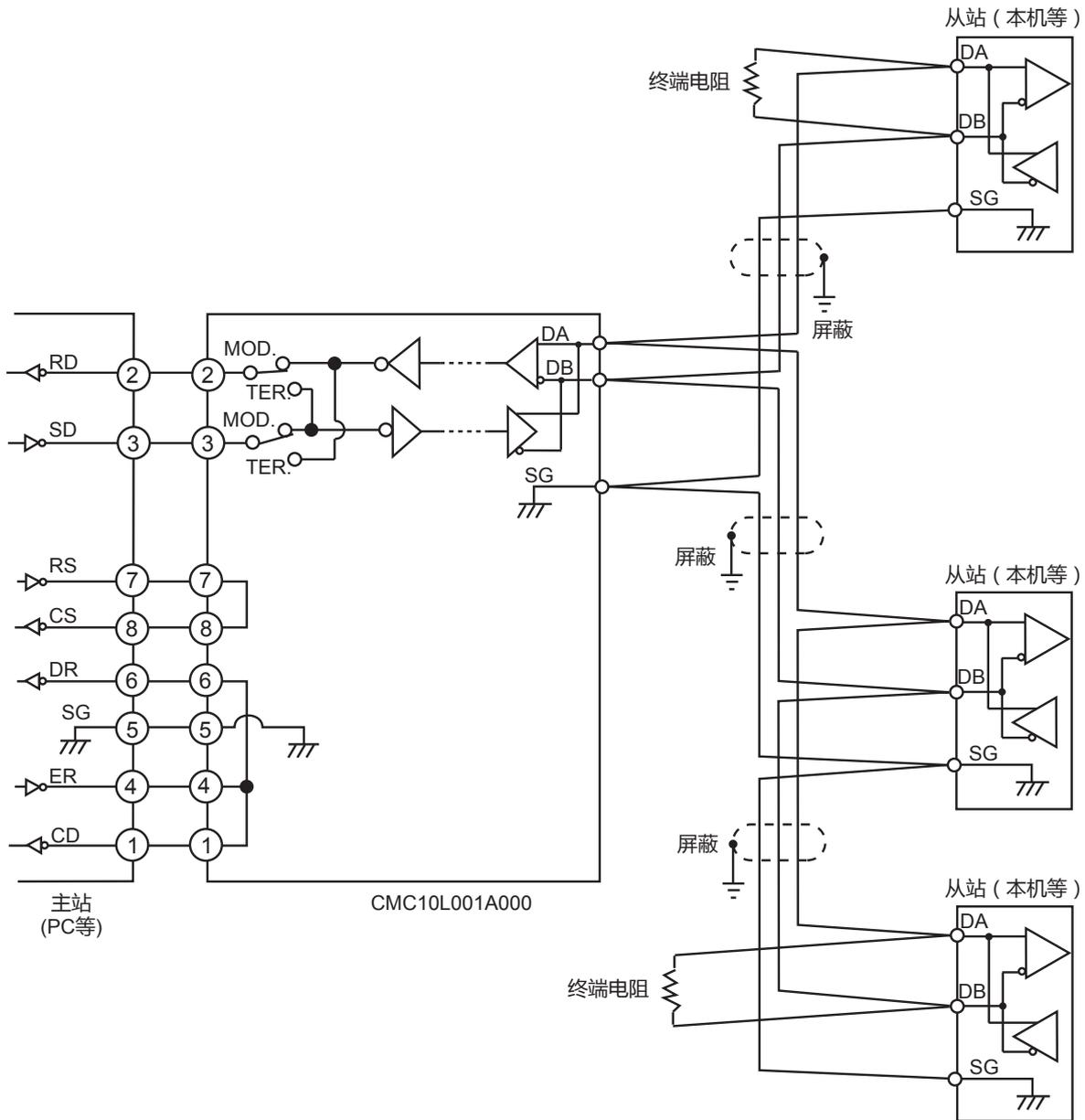
■ 代码表

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)

■ 与 CMC10L001A000 的连接

下图是使用直通电缆线 D-SUB(9 针) 与终端模式的主站 (计算机) 连接的例子。



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

❗ 使用上的注意事项

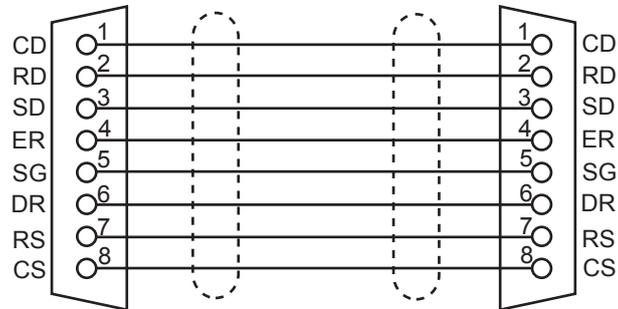
- 请务必连接 SG。如果不连接，将不能获得稳定的通讯。

请把主站的 SD 与从站的 RD、主站的 RD 与从站的 SD 连接。
为此，主计算机（主站）侧 RS-232C 插口的针排列（调制解调器 / 终端）要与使用的缆线种类（交叉 / 直通）匹配，并按下表所示，对 CMC10L001A000 内部的 MODE 开关进行设定。

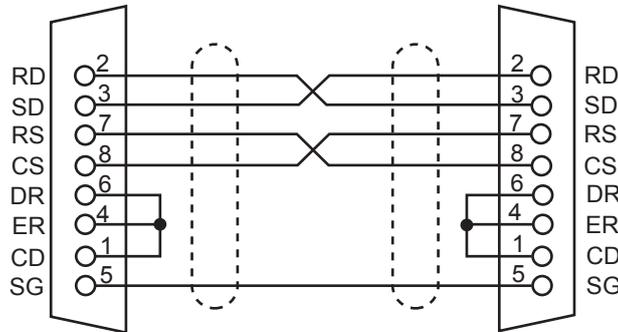
RS-232C	缆线种类	MODE 开关
TERMINAL	直通	MODEM
TERMINAL	交叉	TERMINAL
MODEM	直通	TERMINAL
MODEM	交叉	MODEM

● RS-232C 缆线

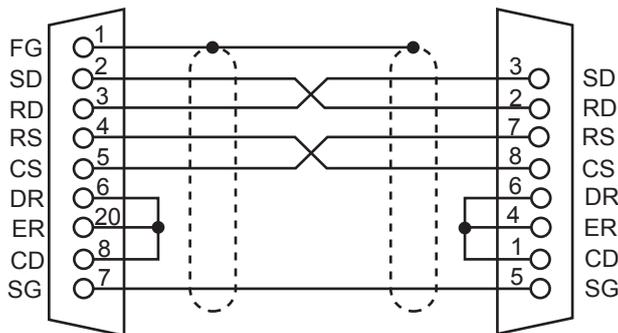
直通：如② - ②、③ - ③那样，把相同编号的针连接的两侧带 D-SUB(9 针) 插口的 RS-232C 缆线。



交叉：如② - ③、③ - ②那样，各信号与不同的针号连接的两侧带 D-SUB(9 针) 插口的 RS-232C 缆线。



D-SUB25 针 -9 针转换缆线：如图所示，把 D-SUB(25 针) 转换成 D-SUB(9 针) 的 RS-232C 缆线。



改订履历

印刷年月	资料编号	种 类	改订页	改订内容
13-05	CP-SP-1183CD	初 版		
16-11		第 2 版	附 -2、附 -3 全书	CMC10L → CMC10L001A000 MVF 系列 → MVF050/080/100/150
20-03		第 3 版	封面	删除 CMC 标志、编号

阿自倍尔仪表（大连）有限公司

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

azbil

阿自倍尔自控工程（上海）有限公司

上海市徐汇区宜山路 700 号 B2 栋 8 楼

电 话：021-50905580 传 真：021-50909810

邮 编：200233 <https://acn.cn.azbil.com/>