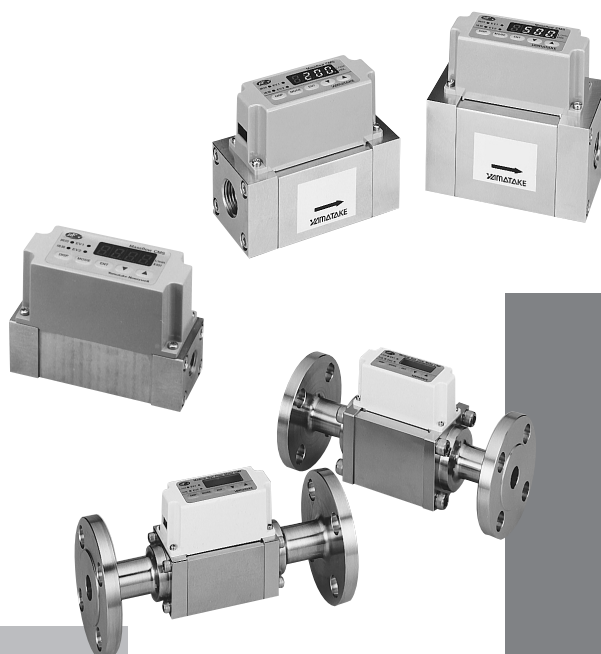




CMS/CMF Series Gas Mass Flow Meter User's Manual for Communication Functions



Thank you for purchasing the CMS/CMF Series Gas Mass Flow Meter.

This manual contains information for ensuring correct use of the CMS/CMF Series communication functions.

This manual should be read by those who design and maintain devices that use the CMS/CMF Series communication functions. Be sure to keep this manual nearby for handy reference.

Yamatake Corporation

IMPORTANT

If the communication involves frequent data writing, the data should be written to RAM addresses. EEPROM is subject to a limit on the number of times it can be written. The limit is 100,000 rewrites.

Note that, for the CMS/CMF series, data written to RAM is lost when the power is turned off, and the system settings default back to the EEPROM data.

Please read the "Terms and Conditions" from the following URL before ordering or use:

<http://www.yamatake.com/products/bi/order.html>

NOTICE

Be sure that the user receives this manual before the product is used.

Copying or duplicating this user's manual in part or in whole is forbidden. The information and specifications in this manual are subject to change without notice.

Considerable effort has been made to ensure that this manual is free from inaccuracies and omissions. If you should find an error or omission, please contact Yamatake Corporation.

In no event is Yamatake Corporation liable to anyone for any indirect, special or consequential damages as a result of using this product.

©2010 Yamatake Corporation ALL RIGHTS RESERVED

Micro Flow™, μ F™ are trademarks of Yamatake Corporation.

Conventions Used in This Manual

- To prevent injury to the operator and others, and to prevent property damage, the following types of safety precautions are indicated:



CAUTION

Cautions are indicated when mishandling this product might result in minor injury to the user, or only physical damage to the product.

- In describing the product, this manual uses the icons and conventions listed below.



Use caution when handling the product.



The indicated action is prohibited.



Be sure to follow the indicated instructions.



Handling Precautions:

Handling Precautions indicate items that the user should pay attention to when handling the CMS/CMF Series.



Note:

Notes indicate information that might benefit the user.



:

This indicates the item or page that the user is requested to refer to.

(1), (2), (3):

Numbers within parentheses indicate steps in a sequence or parts of an explanation.

>>:

Indicates the result of an operation, details displayed on the personal computer or other devices, or the state of the device after operation.

CAUTION



Before doing any wiring work, be sure to disconnect the power. Failure to do so might cause device failure.



Do not disassemble. Device failure could result.



Wire the device properly according to instructions. Use the type of wire specified in the user's manual and follow recognized installation methods. Failure to do so might cause electric shock, fire or faulty operation.



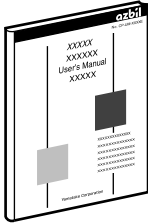
Use the CMS/CMF Series within the operating ranges recommended in the specifications (temperature, humidity, voltage, vibration, shock, mounting direction, atmosphere, etc.). Failure to do so might cause malfunction.



Make sure that wire scraps, shavings, water, etc. do not enter inside the case. Otherwise, malfunction or device failure could result.

The Role of This Manual

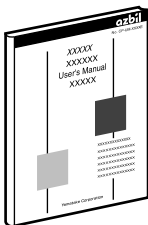
Five manuals have been prepared for the CMS/CMF Series. Read the manual according to your specific requirements. The below lists all the manuals that accompany the CMS/CMF Series and gives a brief outline of the manual. If you do not have the required manual, contact Yamatake Corporation or your dealer.



CMS Series Gas Mass Flow Meter (For Hydrogen and Helium Gases) User's Manual **Manual No. CP-SP-1118E**

This manual is supplied with the device. Personnel in charge of design and/or manufacture of a system using this device must thoroughly read this manual.

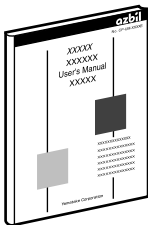
This manual describes the safety precautions, installation, wiring, list of parameters, and primary specifications.



CMS1500 Gas Mass Flow Meter User's Manual **Manual No. CP-SP-1119E**

This manual is supplied with the device. Personnel in charge of design and/or manufacture of a system using this device must thoroughly read this manual.

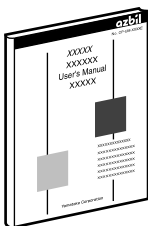
This manual describes the safety precautions, installation, wiring, list of parameters, and primary specifications.



CMS Series SUS/SUS316 Gas Mass Flowmeters CMS9500/0002/0005/0020/0050/0200/0500 User's Manual **Manual No. CP-SP-1302E**

This manual is supplied with the device. Personnel in charge of design and/or manufacture of a system using this device must thoroughly read this manual.

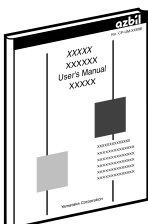
This manual describes the safety precautions, installation, wiring, list of parameters, and primary specifications.



CMF0152/0155 Medical Gas Flow Meter User's Manual **Manual No. CP-SP-1189E**

This manual is supplied with the device. Personnel in charge of design and/or manufacture of a system using this device must thoroughly read this manual.

This manual describes the safety precautions, installation, wiring, list of parameters, and primary specifications.



CMS/CMF Series Gas Mass Flow Meter User's Manual for Communication Functions **Manual No. CP-SP-1184E**

This manual.

Those using the communications functions of the CMS/CMF Series should read this manual.

This manual describes an outline of communications, wiring, communications procedures, CMS/CMF Series communications data, trouble-shooting, and communications specifications.

Organization of This User's Manual

This manual is organized as follows:

Chapter 1. OUTLINE

This chapter briefly describes communication functions of the CMS/CMF Series.

Chapter 2. WIRING

This chapter describes RS-485 wiring methods to make a communication link between the CMS/CMF Series and other instruments.

Chapter 3. SETTING

This chapter describes CMS/CMF Series communication settings.

Chapter 4. COMMUNICATION PROCEDURE

This chapter describes communication procedures, message configuration, data read/write and signal timing operations.

Chapter 5. COMMUNICATION DATA TABLE

This chapter provides various data address tables for communications on the CMS/CMF Series.

Chapter 6. COMMUNICATION PROGRAM FOR MASTER STATION

This chapter gives precautions for programming as well as an example of a CMS/CMF series communication program.

Chapter 7. TROUBLESHOOTING

This chapter describes checkpoints to diagnose failures in CMS/CMF Series communications.

Chapter 8. SPECIFICATIONS

This chapter lists communication specifications for the CMS/CMF Series.

APPENDIX

Contains the code table.

Contents

Conventions Used in This Manual	
SAFETY PRECAUTIONS	
The Role of This Manual	
Organization of This User's Manual	

Chapter 1. OVERVIEW

■ Features	1
------------------	---

Chapter 2. WIRING

2

Chapter 3. SETTING

■ Communication function setup	4
--------------------------------------	---

Chapter 4. COMMUNICATION PROCEDURE

4-1 Communication Procedure and Message Structure	6
■ Communication procedures	6
■ Message structure	6
■ Data link layer	7
■ Application layer	9
4-2 Description of Commands	10
■ Continuous data read command (RS command)	10
■ Continuous data write command (WS command)	11
4-3 Representation of Numeric Data in the Application Layer	12
4-4 Termination Code Table	13
■ Normal and warning termination codes	13
■ Error termination	13
4-5 Timing Specifications	14
■ Timing specifications for instruction and response message	14
■ RS-485 driver control timing specifications	14

Chapter 5. COMMUNICATION DATA TABLE

5-1 Basic Communication Data Processing	15
■ Communication data types and formats	15
■ Communication data storage memory	15
■ Data address	16
■ Data unit and decimal point position	16

5-2	Communication Data Address Table	17
■	Model-related data	17
■	Operating status data	18
■	Instantaneous flow rate data	19
■	Integrated flow rate data	19
■	Function setup data	20
■	Parameter setup data	22
Chapter 6.	COMMUNICATION PROGRAM FOR MASTER STATION	23
■	Before running the program	23
■	Running the program	23
■	Sample data read/write program	24
Chapter 7.	TROUBLESHOOTING	
■	Check items in case communication is disabled	29
Chapter 8.	SPECIFICATIONS	30
APPENDIX		
■	Code table	31
■	Connection with CMC10L	32

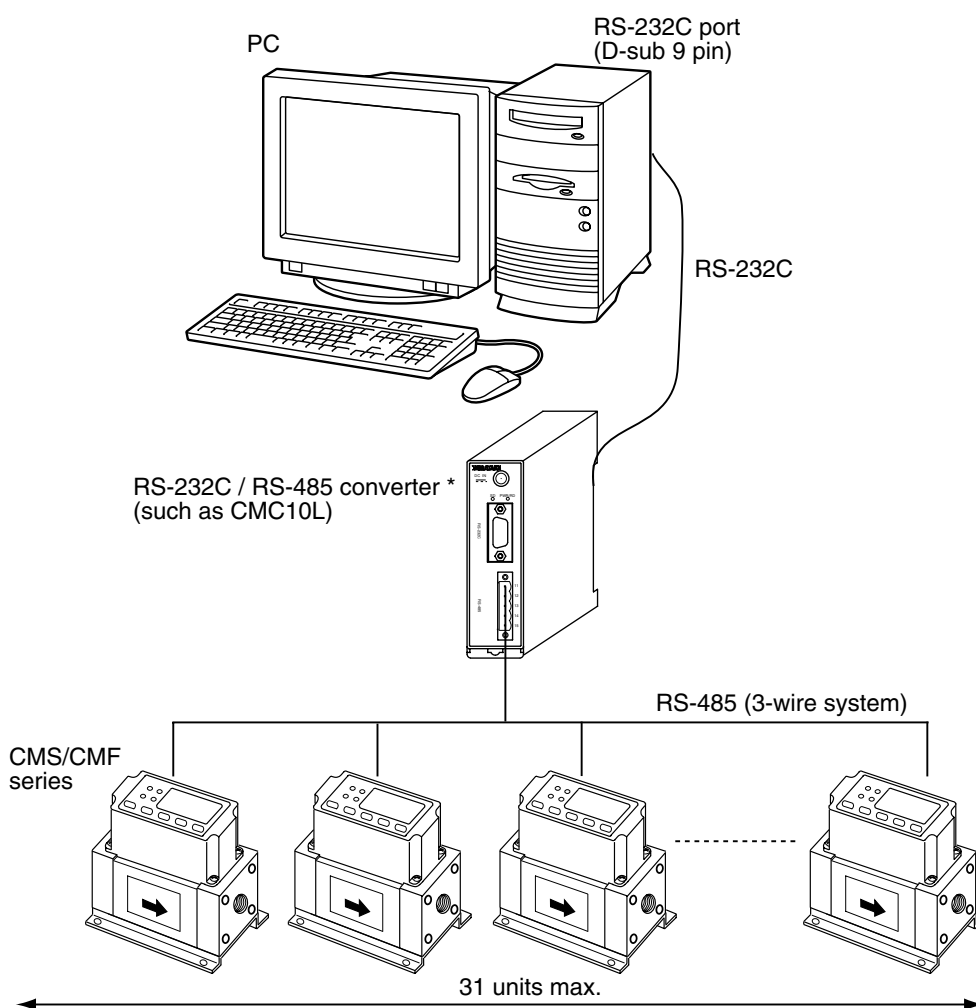
Chapter 1. OVERVIEW

CMS/CMF series mass flow meters (herein also called the slave station) has a communication function that can exchange set points and other data with a PC, PLC or other host device (also called the master station). The communication program must be provided by the customer.

■ Features

The features of the CMS/CMF series' communication function are as follows:





- Up to 31 units can be connected to a single master station (host device).
- A CMC10L communication converter (sold separately) is required if the host device uses RS-232C communications. The CMC10L handles RS-232C/RS-485 conversion.
- The baud rate can be up to 9600 bps.







*: A Yamatake CMC10L001A000 may also be used as an RS-232C/RS-485 converter.

Chapter 2. WIRING

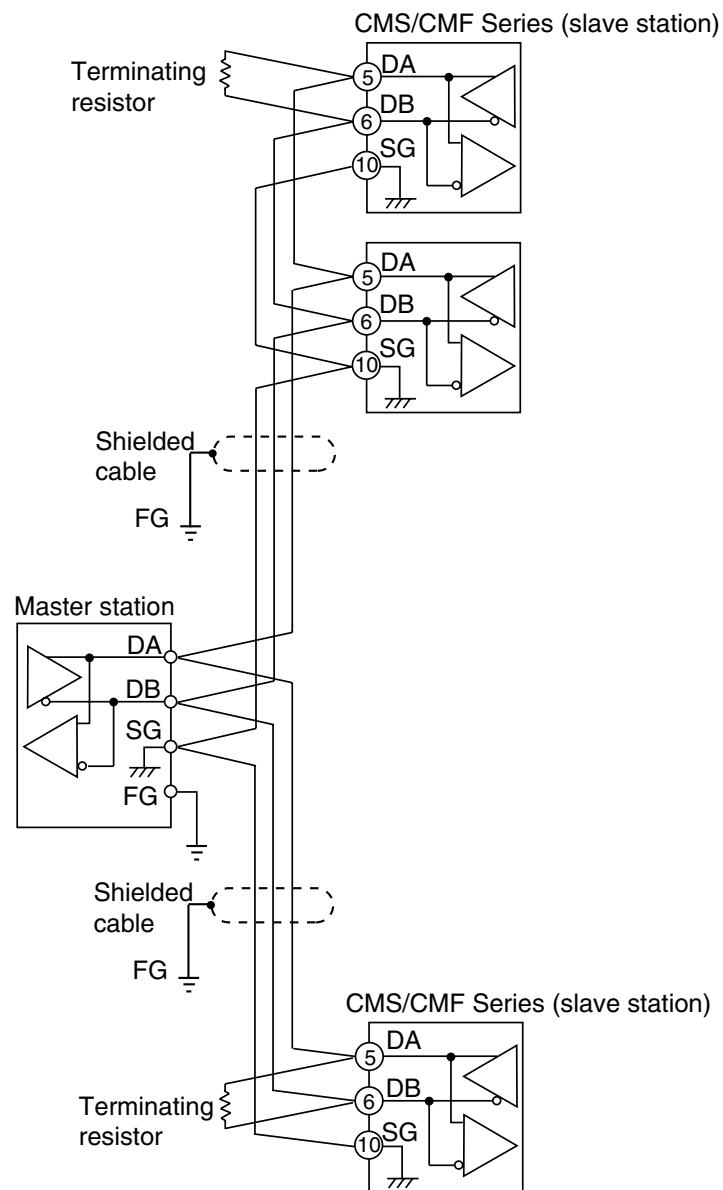
CAUTION

-  Before doing any wiring work, be sure to disconnect the power. Failure to do so might cause device failure.
-  Do not disassemble. Device failure could result.
-  Use the CMS/CMF Series within the operating ranges recommended in the specifications (temperature, humidity, voltage, vibration, shock, mounting direction, atmosphere, etc.). Failure to do so might cause malfunction.
-  Make sure that wire scraps, shavings, water, etc. do not enter inside the case. Otherwise, malfunction or device failure could result.

Note

- For wiring methods other than for RS-485
 -  CMS Series Gas Mass Flow Meter (For Hydrogen and Helium Gases) User's Manual, CP-SP-1118E
 -  CMS1500 Gas Mass Flow Meter User's Manual, CP-SP-1119E
 -  CMF0152/0155 Medical Gas Flow Meter User's Manual, CP-SP-1302E
 -  CMS Series SUS/SUS316 Gas Mass Flowmeters CMS9500/0002/0005/0020/0050/0200/0500 User's Manual, CP-SP-1189E

Example of RS-485 connection



Connect one terminating resistor of $150\ \Omega \pm 5\%$, $1/2W$ min. to the device at each end of the transmission line.

The FG grounding must not be made at the both shielded wire ends but only at one location.

A Yamatake CMC10L001A000 can be used as the converter for the master station.

! Handling Precautions

- Be sure to connect SG. Communications may not operate reliably if SG is not connected.

Chapter 3. SETTING

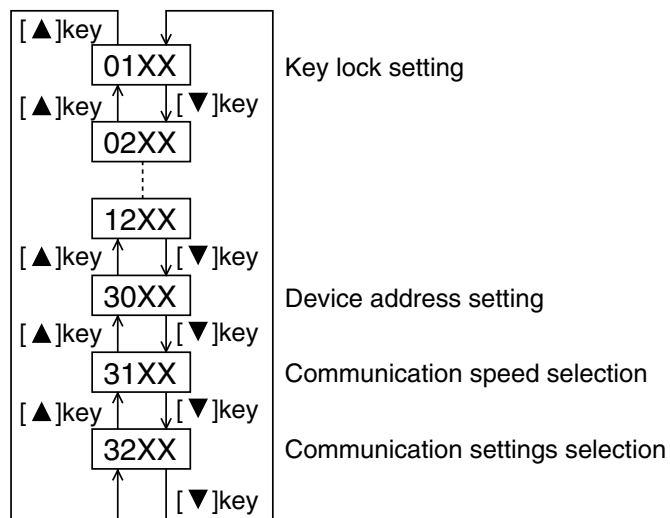
To operate the RS-485 communications functions, suit the communication parameters and station address of this device to this device to the setting of the master station.

■ Communication function setup

● Setting method

Setting are as follows:

- (1) Press the [MODE] key.
>> The left-most two digits on the display start blinking. The two digits on the left indicate the item being set and the two digits on the right indicate the setting value.
- (2) Press the [▲] and [▼] keys to select the desired setting mode number.
At this moment, the lower 2 digits indicate the current setting value. Also, the function type (number) blinks.



- (3) Press the [ENT] key.
>> This selects the item to be set. The two digits on the right start blinking.
- (4) Press the [▲] and [▼] keys to adjust the two right-hand digits to the desired setting value, and then press the [ENT] key.
>> All four digits light up and the specified value is set.
- (5) To continue configuring, return to step (1) and repeat the procedure.
When you have finished with settings, proceed to step (6)
- (6) Press the [DISP] key.
>> This changes the display back from function setting mode to instantaneous flow rate display or integrated flow rate display.

! Handling Precautions

- Pressing the [DISP] key in the middle of step 4 (with the setting blinking) cancels the update and leaves the previous setting value unchanged.

● Setup item

Mode	Function	Setup	Factory setting	Note
30	Station address	0: Communication function is disabled. 1 to 99: Station address	0	No communication when set at 00. Set an address different from other slave stations.
31	Transmission speed	0: 9600 bps 1: 4800 bps 2: 2400 bps	0	
32	Data format	0: 8-bit data, even parity, 1 stop bit 1: 8-bit data, no parity, 2 stop bits	0	

Chapter 4. COMMUNICATION PROCEDURE

4 - 1 Communication Procedure and Message Structure

■ Communication procedures

The communication procedure is as follows:

- (1) The instruction message is sent from the host device (master station) to one unit (slave station) to communicate with.
- (2) The slave station receives the instruction message, and performs read or write processing according to the content of the message.
- (3) The slave station sends a message corresponding to the processing content as a response message.
- (4) The master station receives the response message.

■ Message structure

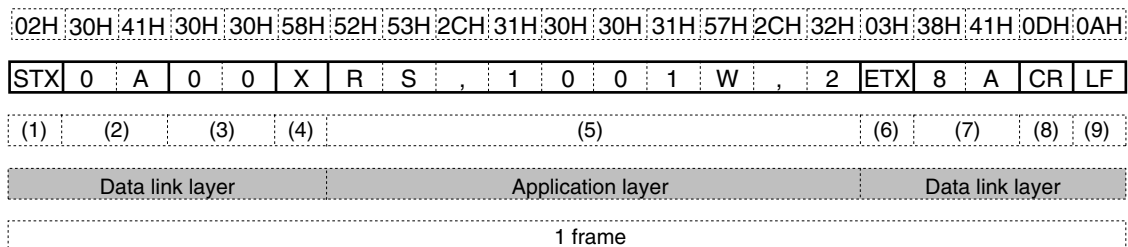
The message structure is as follows:

Messages are broadly classified into two layers; the data link layer and the application layer.

- Data link layer
This layer contains the basic information required for communication, such as the destination of the instruction message and the message check data.
- Application layer
Data is read and written in this layer. The content of the layer varies according to the purpose of the message.

Messages comprise parts (1) to (9) as shown in the figure below.

The command (details sent from the master station) and the response (details returned from the slave station) are stored in the application layer.



- | | |
|--|-----------------------------------|
| (1) STX (start of message) | (6) ETX (end of command/response) |
| (2) Station address | (7) Checksum |
| (3) Sub-address (always 00) | (8) CR (delimiter) |
| (4) Device code | (9) LF (delimiter) |
| (5) Send message = command,
response message = response | |

■ Data link layer

● Outline

The data link layer is of a fixed length. The position of each data item and the number of its characters are already decided. Note, however, that the data positions of the data link layer from ETX onwards shift according to the number of characters in the application layer. The character length, however, remains unchanged.

● Response start conditions

The slave station sends a response only if the data link layer message structure (station address, sub-address, checksum, frame length, etc.) is entirely correct. If one or more of these is invalid, the slave station enters the STX receive wait state without sending a response.

● Data link layer data definitions

The table below lists the data definitions for the data link layer.

Data item	Character code	No. of bytes	Meaning
STX	02H	1	Start message
Station address	Hexadecimal character code	2	Identifies the recipient station
Sub-address	Hexadecimal character code	2	Always 00
Device code	"X" (58H) or "x" (78H)	1	Station type
ETX	03H	1	End of application layer
Checksum	Two-digit hexadecimal character code (00H to FFH)	2	Message checksum
CR	0DH	1	End of message (1)
LF	0AH	1	End of message (2)

● Description of data items

• STX (02H)

When STX is received, the device judges this to be the start of the send message. Accordingly, the device returns to the initial state regardless of the reception state it was in and commences processing on the assumption that the STX was the first character of a new message. The purpose of this is to ensure that, if electrical noise (etc.) causes an error in an instruction message, at the next valid message from the master station (e.g., a RETRY message) the slave station will have recovered and will be able to respond.

• Station address

The slave station only processes and replies to instructions if they contain its own station address. However, the slave station does not reply if the station address is set to 0 (30H 30H), even if this is the slave station address. The slave station also adds its own station address to its reply.

The station address is a two-digit hexadecimal value.

Example: If the station address of the receiver station is 10:

(1) 10 (decimal) = 0AH (hexadecimal)

(2) Convert to character code

0=30H

A=41H

(3) Set the above value ("0A" = 30H 41H) as the station address.

- Sub-address

As this unit does not use the sub-address, set it to "00" (30H 30H). The slave station also adds the sub-address to its reply.

- Device code

The unit sets "X" (58H) or "x" (78H) as its device code. This code is determined for each device series, and no other character may be used. The slave station returns the received message with the station's device code added. It is recommended that "X" (58H) be used for first messages and "x" (78H) be used to indicate resent messages.

- ETX

ETX indicates the end of the application layer.

- Checksum

This value is for checking whether or not some abnormality (e.g. noise) causes the message content to change during communications.

The checksum is expressed as two hexadecimal characters.

- How to calculate a checksum

(1) Add the character codes in the message from STX through ETX in single byte units.

(2) Take the two's complement of the result.

(3) Convert to character codes.

Example: The checksum for the example on page 4-1 is as follows:

(1) Sum the character codes from STX to ETX one byte at a time. The lower-order byte of the result is 76H.

(2) The two's complement of 76H is 8AH.

(3) Converting 8AH to character codes gives 38H and 41H.

- CR/LF

This indicates the end of the message. After receiving LF, the device is immediately ready to process received messages.

■ Application layer

● Application layer structure

The table below shows the structure of the application layer.

Item	Description
Command	"RS" (command to read data addresses sequentially in decimal format)
	"WS" (command to write data addresses sequentially in decimal format)
Data delimiter	RS, WS: "," (comma)
Data address	RS, WS: "501W," etc.
Number of read data items	A number represented in character format (e.g., "1")
data items	RS, WS: a number represented in character format (e.g., "100")

● Number of data items able to be accessed in one frame

Type	Command	RAM area	EEPROM area
RS	Decimal format read command	8	8
WS	Decimal format write command	4	4

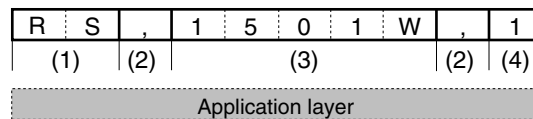
4 - 2 Description of Commands

■ Continuous data read command (RS command)

This command reads data at sequential addresses. It instructs the slave station to read the data as one message from sequential addresses starting from the specified start address.

● Send message

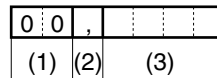
This command enables the content of continuous data addresses starting with the specified read start address to be read as a single message. The figure below shows the structure of the application layer of the send message when the data is read.



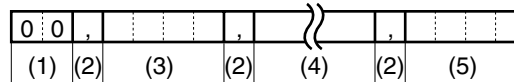
- (1) Continuous read command
- (2) Data delimiter
- (3) Data address
- (4) Number of read data

● Response message

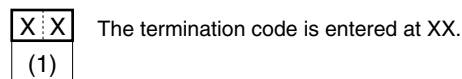
- Normal termination (reading of single data item)



- Normal termination (reading of multiple data items)



- Warning termination



- (1) Termination code
- (2) Data delimiter
- (3) Data
- (4) Data 2 to (n-1)
- (5) Data n

* For details of codes, refer to;

4-4, Termination Code Table (on page 13).

■ Continuous data write command (WS command)

This command writes data to continuous addresses.

● Instruction message

W	S	,	1	5	0	1	W	,	1	,	6	5
(1)	(2)		(3)				(2)	(4)	(2)		(5)	

- (1) Write command
- (2) Data delimiter
- (3) Start write data address
- (4) Write data (first word)
- (5) Write data (second word)

● Response message

The figure below shows the structure of the application layer of the response message for the data write command.

● Normal termination

0	0
(1)	


● Warning termination or error termination

X	X
(1)	

The termination code is entered at XX.

(1) Termination code

* For details of codes, refer to;

 4-4, Termination Code Table (on page 13).

4 - 3 Representation of Numeric Data in the Application Layer

All values must be specified with zeros suppressed.

The following describes the specifications, including cases when zero suppression is not used. Instruction messages from the master station must always send data with zeros suppressed.

- **RS and WS commands**

Item	Specification	Error handling
Unnecessary spaces	Exclude	Message processing is aborted and an error termination code is returned in the response.
Unnecessary zeros	Exclude	
Value = zero	Do not omit. Always use "0".	
Other unnecessary characters	Prefix negative values with a "-" sign. Do not use any other prefixes. Do not prefix positive values with "+".	
Permitted range	-32768 to +32767 Values outside this range are prohibited.	

4 - 4 Termination Code Table

The response message must include a termination code.

■ Normal and warning termination codes

Termination code	Type	Contents and action
00	Normal	Normal end
21	Warning	Processing continued without writing a value to the address. The communication function tried to write to an unwritable address.
23	Warning	Reading halted because of an access to an out-of-range address. Writing halted because of an access to an out-of-range address. However, write operations prior to the error were completed.

■ Error termination

Termination code	Type	Contents and action
40	Error	"W" was not set in the address. All messages are aborted.
41	Error	"WS" or "RS" has not been set. All messages are aborted.
43	Error	"ETX" is not set on the correct position. "." is not set after the address. All messages are aborted.
46	Error	The address is erroneous. All messages are aborted.
47	Error	The number of read data is error. All messages are aborted.
48	Error	The written data is error. Writing has been executed except for the error address.
99	Error	An undefined command or other message error. All messages are aborted.

4 - 5 Timing Specifications

■ Timing specifications for instruction and response message

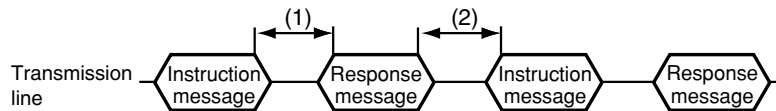
The cautions below are required with regard to the timing to transmit a instruction message from the master station and a response message from the slave station.

● Response monitor time

The maximum response time from the end of the instruction message transmission by the master station until when the master station receives a response message from the slave station is two seconds ((1) in the figure below). So, the response monitor time should be set to two seconds.

Generally, when a response time-out occurs, resend the instruction message. See Chapter 6, "Communication Program for Master Station," for details.

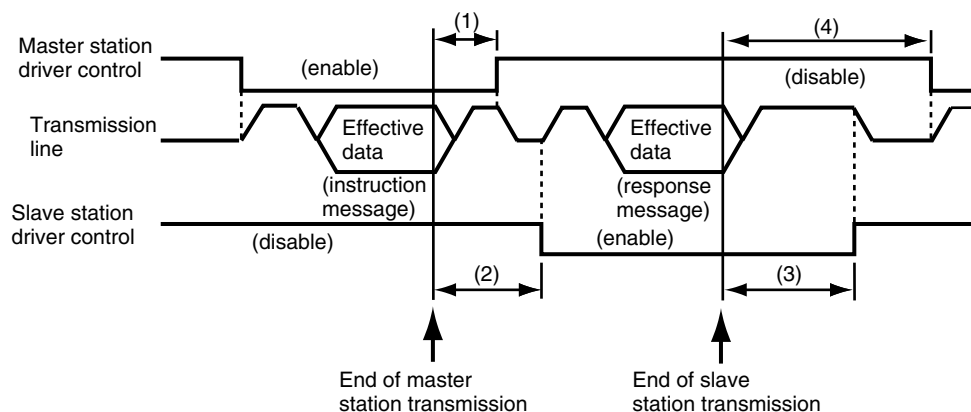
● Transmission start time



- (1) Time from completion of master station transmission to start of slave station transmission = 2000 ms or less.
(Use a response monitor time of 2 s from the time the master station completes sending of the instruction message.)
- (2) Time from completion of slave station transmission to start of master station transmission = 50 ms or more.
(The master station must wait for at least 50 ms after receiving a response from the slave station before starting to send a new message.)

■ RS-485 driver control timing specifications

When the transmission/reception on the RS-485 is directly controlled by the master station, care should be paid to the following timing:



- (1) End of master station transmission - Driver disable time = 500 μ s max.
- (2) End of slave station reception - Driver enable time = 30ms min.
- (3) End of slave station transmission - Driver disable time = 10ms max.
- (4) End of master station reception - Driver enable time = 50ms min.

Chapter 5. COMMUNICATION DATA TABLE

5 - 1 Basic Communication Data Processing

■ Communication data types and formats

● Types of communication data

The communications data are categorized as follow:

- Model-related data
- Operating status data
- Instantaneous flow rate data
- Integrated flow data
- Function setup data
- Parameter setup data

● Format of communication data

Communication data is classified into the following formats:

- Numeric data: Data indicating a numeric value (PV, SP, etc.).
- Bit data: Data where each bit is significant (alarms, etc.). Bit data must be composed by transmission and decomposed by reception.

IMPORTANT

The number of times EEPROM can be rewritten is 100,000 times or less.
If data is updated frequently and repetitively via communications, write the data to a RAM address.

■ Communication data storage memory

● Memory type

Data sent by the communication function is saved in the unit's memory. The unit has the following two types of memory.

- RAM: Stored data is cleared when the power is turned OFF. However, data can be written to this memory any number of times.
- EEPROM: Stored data is retained even when the power is turned OFF. However, the nature of the memory elements means that it is subject to a limit on the number of times it can be written. The limit is 100,000 rewrites.

● Communication data memory

The communication function can read and write data using the above two types of memory, one of which should be selected depending on the purpose.

The differences between the two types of memory are as follows.

- RAM: Reads and writes data only from/to RAM. If the power is turned off after data is written to RAM, the EEPROM data is copied to RAM when the unit is next turned on, making the RAM data the same as the EEPROM data.
- EEPROM: Writes data in RAM and EEPROM.

■ Data address

The table below lists the data addresses

Communication data	RAM		EEPROM	
	Offset	Address	Offset	Address
Model-related data	1000	1001 to 1199	4000	4001 to 4199
Operating status data	1200	1201 to 1399	4200	4201 to 4399
Instantaneous flow rate data	1400	1401 to 1599	4400	4401 to 4599
Integrated flow rate data	1600	1601 to 1799	4600	4601 to 4799
Function setup data	2000	2001 to 2199	5000	5001 to 5199
Parameter setup data	2200	2201 to 2399	5200	5201 to 5399

■ Data unit and decimal point position

Read/write data is not appended with a decimal point.

The unit and decimal point position is determined for each data item.

Refer to the user's manual for the mass flow meter for details about the units and decimal point position.

5 - 2 Communication Data Address Table

The enabling conditions for the address and R/W (Read/Write) of each data are specified in the following table:

The meaning of R/W column marks:

○ Possible

X Impossible

■ Model-related data

Item	Range	RAM			EEPROM			Remarks	
		アドレス	R	W	Address	R	W		
Gas type for each series	CMS SUS/SUS316	0: Nitrogen/air 1: Argon 2: Carbon dioxide 3: Oxygen 4: City gas 13A (46MJ) 5: Methane 6: Propane 7: Butane 8: User setting 11: City gas 13A (45MJ)	1001	○	X	4001	X	X	The gas setting can be modified in the function setup (address 2008, 5008)
	CMS hydrogen/ helium model	8: User setting 9: Hydrogen 10: Helium							
	CMS1500	0: Nitrogen/air 1: Argon 2: Carbon dioxide gas 8: User setting							
	CMF	0: Nitrogen/artificial air 1: Oxygen 2: Laughing gas							
Undefined		1002	○	X	4002	X	X		
Decimal point display position for instantaneous flow rate	0: No decimal point 1: xxxx. 2: xxx.x 3: xx.xx 4: x.xxx	1003	○	X	4003	X	X		
Unit for integrated flow rate	0: No decimal point 1: xxxxxxxx. 2: xxxxxxxx.x 3: xxxxxx.xx 4: xxxxx.xxx	1004	○	X	4004	X	X		
Decimal point display position for integrated flow rate	0: mL/min 1: L/min	1005	○	X	4005	X	X		
Unit for instantaneous flow rate	0: mL 1: L 2: m ³	1006	○	X	4006	X	X		

■ Operating status data

Item	Range	RAM			EEPROM			Remarks
		Address	R	W	Address	R	W	
Alarm status	*1	1201	○	X	4201	X	X	Status is represented as a decimal
Event status	*2	1202	○	X	4202	X	X	Status is represented as a decimal
Undefined		1203	○	X	4203	X	X	
Undefined		1204	○	X	4204	X	X	
Lower digits of integrated flow rate	0 to 9999	1205	○	○	4205	○	○	This is the same as the integrated flow rate data at addresses 1603 and 4603 *
Upper digits of integrated flow rate	0 to 9999	1206	○	○	4206	○	○	This is the same as the integrated flow rate data at addresses 1604 and 4604 *
Instantaneous flow rate	0 to 9999	1207	○	X	4207	X	X	This is the same as the instantaneous flow rate data at address 1401

* Writing to these addresses is not permitted in the CMF series.

*1. Alarm status bits (address 1201) 0: OK 1: Error

Bit No.	Description
0	ALHI range exceeded
1	Undefined (always 0)
2	Undefined (always 0)
3	Undefined (always 0)
4	ERR1: sensor abnormal
5	ERR2: adjustment data abnormal
6	ERR3: sensor heater abnormal
7	ERR4: sensor safety circuit triggered

*2. Event status bits (address 1202) 0: Off 1: On

Bit No.	Description
0	Event output 1 status
1	Event output 2 status
2	Undefined (always 0)
3	External input status
4	Undefined (always 0)
5	Undefined (always 0)
6	Undefined (always 0)
7	Undefined (always 0)

■ Instantaneous flow rate data

Item	Range	RAM			EEPROM			Remarks
		Address	R	W	Address	R	W	
Instantaneous flow rate	0 to 9999	1401	○	X	4401	X	X	
Event output 1 instantaneous flow rate setting	0 to 9999	1402	○	○	4402	○	X	Same as data at parameter setup addresses 2201 and 5201 *
Event output 2 instantaneous flow rate setting	0 to 9999	1403	○	○	4403	○	X	Same as data at parameter setup addresses 2204 and 5204 *

* To write to EEPROM, use the addresses for the parameter setup data.

■ Integrated flow rate data

Item	Range	RAM			EEPROM			Remarks
		Address	R	W	Address	R	W	
Undefined		1601	○	X	4601	○	X	
Undefined		1602	○	X	4602	○	X	
Lower 4 digits of integrated flow rate	0 to 9999	1603	○	○	4603	○	○	
Upper 4 digits of integrated flow rate	0 to 9999	1604	○	○	4604	○	○	
Lower 4 digits of integrated flow rate setting for event output 1	0 to 9999	1605	○	○	4605	○	X	Same as data at parameter setup addresses 2202 and 5202 *1
Upper 4 digits of integrated flow rate setting for event output 1	0 to 9999	1606	○	○	4606	○	X	Same as data at parameter setup addresses 2203 and 5203 *1
Lower 4 digits of integrated flow rate setting for event output 2	0 to 9999	1607	○	○	4607	○	X	Same as data at parameter setup addresses 2205 and 5205 *1
Upper 4 digits of integrated flow rate setting for event output 2	0 to 9999	1608	○	○	4608	○	X	Same as data at parameter setup addresses 2206 and 5206 *1
Lower 4 digits of initial value for reverse-integrated flow rate	0 to 9999	1609	○	○	4609	○	X	Same as data at parameter setup addresses 2211 and 5211 *1 *2
Upper 4 digits of initial value for reverse-integrated flow rate	0 to 9999	1609	○	○	4610	○	X	Same as data at parameter setup addresses 2212 and 5212 *1 *2

*1 To write to EEPROM, use the addresses for the parameter setup data.

*2 Reading and writing to these addresses is not permitted in the CMF series.

■ Function setup data

Item	Range	RAM			EEPROM			Remarks
		Address	R	W	Address	R	W	
Key lock setup	0: Key lock OFF 1: Key lock ON	2001	○	○	5001	○	○	
Measurement mode setup	0: Instantaneous flow rate 1: Instantaneous flow rate/integrated flow rate 2: Instantaneous flow rate/reverse integrated flow rate	2002	○	○	5002	○	○	For the CMF series, the value is fixed at "1: Instantaneous flow rate/integrated flow rate" and writing is prohibited.
Event output 1 (EV1) setup	0: Do not use 1: Instantaneous flow rate upper limit 2: Instantaneous flow rate lower limit 3: Integrated flow rate count up 4: Integrated flow rate count down 5: Serial output 6: Error output	2003	○	○	5003	○	○	The operation of integrated count up, reverse-integrated count down, and integrated output pulse are available when the measurement mode is set to 1 or 2. Setting integrated count up and reverse-integrated count down at the same time is prohibited. On the CMF series, the reverse integrated count down setting is prohibited. The value of the integrated flow rate pulse rate is different on different models.
Event output 2 (EV2) setup	0: Do not use 1: Instantaneous flow rate upper limit 2: Instantaneous flow rate lower limit 3: Integrated flow rate count up 4: Integrated flow rate count down 5: Integrated flow rate pulse rate 1 6: Integrated flow rate pulse rate 2 7: Integrated flow rate pulse rate 3	2004	○	○	5004	○	○	Refer to the user's manuals for each series.
On delay setting (EV1)	0: Do not use 1: Use	2005	○	○	5005	○	○	This setting is only effective if the setting for event output 1 is 1 or 2.
On delay setting (EV2)	0: Do not use 1: Use	2006	○	○	5006	○	○	This setting is only effective if the setting for event output 2 is 1 or 2.
Event standby setting	0: Do not use 1: Use	2007	○	○	5007	○	○	This setting is only effective if the setting for event output 1 and event output 2 are 2.

Item	Range	RAM			EEPROM			Remarks	
		Address	R	W	Address	R	W		
Gas type for each series	CMS SUS/SUS316	0: Nitrogen/air 1: Argon 2: Carbon dioxide 3: Oxygen 4: City gas 13A (46MJ) 5: Methane 6: Propane 7: Butane 8: User setting 11: City gas 13A (45MJ)	2008	○	○	5008	○	○	Oxygen can only be specified on models with an oxygen measurement capability. Refer to the user's manuals for each series.
	CMS hydrogen/helium model	8: User setting 9: Hydrogen 10: Helium							
	CMS1500	0: Nitrogen/air 1: Argon 2: Carbon dioxide . gas 8: User setting							
	CMF	0: Nitrogen/ artificial air 1: Oxygen 2: Laughing gas							
Analog output scaling	0: Scaling 1 1: Scaling 2 2: Scaling 3 3: Scaling 4 4: User-defined scaling	2009	○	○	5009	○	○	The scaling 1 to 4 values are different for different models. Refer to the user's manuals for each series.	
Analog output type selection	0: 0 to 5 V 1: 1 to 5 V 2: 4 to 20 mA	2010	○	○	5010	○	○		
Reference temperature for flow rate conversion	0 to 35	2011	○	○	5011	○	○	0 to 35 °C (1 °C increments)	
Low-flow cut setting	0: Low-flow cutoff 1: Less than minimum display value * 2: Less than 1% FS 3: Less than 2.5% FS 4: Less than 5% FS	2012	○	○	5012	○	○	* The minimum display value is different for different models. Refer to the user's manuals each series.	
Station address	0: Do not use communications 1 to 99: Communication address	2030	○	X	5030	○	X		
Communication speed selection	0: 9600 bps 1: 4800 bps 2: 2400 bps	2031	○	X	5031	○	X		
Communication settings	0: 8 data bits, even parity, 1 stop bit 1: 8 data bits, no parity, 2 stop bits	2032	○	X	5032	○	X		

■ Parameter setup data

Item	Range	RAM			EEPROM			Remarks
		Address	R	W	Address	R	W	
Event output 1 instantaneous flow rate	0 to 9999	2201	<input type="radio"/>	<input type="radio"/>	5201	<input type="radio"/>	<input type="radio"/>	
Event output 1 integrated flow rate lower limit	0 to 9999	2202	<input type="radio"/>	<input type="radio"/>	5202	<input type="radio"/>	<input type="radio"/>	
Event output 1 integrated flow rate upper limit	0 to 9999	2203	<input type="radio"/>	<input type="radio"/>	5203	<input type="radio"/>	<input type="radio"/>	
Event output 2 instantaneous flow rate	0 to 9999	2204	<input type="radio"/>	<input type="radio"/>	5204	<input type="radio"/>	<input type="radio"/>	
Event output 2 instantaneous flow rate lower limit	0 to 9999	2205	<input type="radio"/>	<input type="radio"/>	5205	<input type="radio"/>	<input type="radio"/>	
Event output 2 instantaneous flow rate upper limit	0 to 9999	2206	<input type="radio"/>	<input type="radio"/>	5206	<input type="radio"/>	<input type="radio"/>	
Event output 1 hysteresis	0 to 100	2207	<input type="radio"/>	<input type="radio"/>	5207	<input type="radio"/>	<input type="radio"/>	
Event output 2 hysteresis	0 to 100	2208	<input type="radio"/>	<input type="radio"/>	5208	<input type="radio"/>	<input type="radio"/>	
Event output 1 on delay	0 to 60	2209	<input type="radio"/>	<input type="radio"/>	5209	<input type="radio"/>	<input type="radio"/>	unit (S)
Event output 2 on delay	0 to 60	2210	<input type="radio"/>	<input type="radio"/>	5210	<input type="radio"/>	<input type="radio"/>	unit (S)
Lower 4 digits of initial value for reverse-integrated flow rate	0 to 9999	2211	<input type="radio"/>	<input type="radio"/>	5211	<input type="radio"/>	<input type="radio"/>	On the CMF series, reading, and writing of the lower 4 digits of the initial value are prohibited.
Upper 4 digits of initial value for reverse-integrated flow rate	0 to 9999	2212	<input type="radio"/>	<input type="radio"/>	5212	<input type="radio"/>	<input type="radio"/>	On the CMF series, reading, and writing of the lower 4 digits of the initial value are prohibited.
User-defined gas conversion factor	100 to 8000	2213	<input type="radio"/>	<input type="radio"/>	5213	<input type="radio"/>	<input type="radio"/>	This setting is only effective if "user setting" was specified as the gas type in the function setup data. Setting value corresponds to a value of 0.100 to 8.000
User setting for analog output scaling	100 to 250	2214	<input type="radio"/>	<input type="radio"/>	5214	<input type="radio"/>	<input type="radio"/>	

When the function setup data corresponds to a parameter setting, the specified data is used.

Chapter 6. Communication Program for Master Station

The sample program in this chapter was written with Borland C++ Builder 5.0 or Borland C++ Compiler 5.5 for Windows 95, 98, NT, or 2000.

The program is provided as an example to help you write your own program and its operation is not subject to any guarantees. Borland C++ Compiler 5.5 can be downloaded from the Borland web site.

■ Before running the program

- Check the communication settings and station address on the mass flow meter.
- The maximum response time from the slave station is 2 seconds. Please use a monitoring time of 2 s on the master station. If no response is received within 2 s, resend the message.

Note

- If "X" is used as the device code in an instruction message, the device code returned in the response message will also be "X." Similarly, if the device code in the instruction message is "x," the response message will also contain "x."

By changing the device code from "X" to "x" or vice versa when resending a message, the master station can determine whether a received response message belongs to the initial message or the resent message.

■ Running the program

This program reads and writes data.

Running the program displays the application layer instruction and response messages exchanged between the stations.

```
command:RS, 1000W, 2
result:00, 0, 0
command:WS, 1000W, 2
result:00
```

Sample execution results

● Setting up communications

Call `Open()` to initialize RS-232C.

● Executing commands

Set the character string to execute in `Command` and then call `AppCPL()`.

■ Sample data read/write program

! Handling Precautions

- Yamatake is not liable for any damage resulting from the use of this program.

```
//-----
// 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("command:%s\n",command);
        printf("result:%s\n",recieve);

        strcpy(command,"WS,1000W,2");
        AppCPL(command,recieve);
        printf("command:%s\n",command);
        printf("result:%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_19200;
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( "unsigned long size" );
        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 ){
                // Clear Error Flag
                ClearCommError( handle, &Error, &Stat );
                // Set Error Flag
                *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( strcmp( (char*)theMsg, (char*)theHdr, 6 )!=0 ){
        ErrorCode = 0x0020000;
        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 = 0x0040000;
            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 = 0x0020000;
        if( Cnt++<COMRESENDNUM ){
            goto resend;
        }
        return false;
    }else{
        *etx = 0x00;
    }
    strcpy( received, (char*)&theApp[0] );
}
return true;
}
//-----

```

Chapter 7. TROUBLESHOOTING

■ Check items in case communication is disabled

- (1) Check the power supply.
- (2) Check the wiring.
- (3) Check if the communication conditions for the CMS/CMF meet those for the host computer.
Communications will not function correctly if any of the following items are set differently.
The underlined values are the settings that can be set on the device.
Baud rate: 9600, 4800, 2400 bps
No. of data bits: 8
Parity: No parity, even parity
No. of stop bits: 1 bit, 2 bits
- (4) Check if the destination address of the command frame transmitted from the host computer meets the address set to the CMS/CMF Series.
The address of the CMS/CMF set to "0" for factory setting.
Even when the destination address of the command frame is set to 00 (30H, 30H), the CMS/CMF does not respond to such a message.
- (5) Are those multi-dropped units belong operated themselves with different station address setups.
- (6) Is the communication timing conformed with the Section 4-5 Timing Specifications (page 14).
- (7) Use the upper-case character codes for all the character codes other than the device code ("X" or "x").

Chapter 8. SPECIFICATIONS

Item	Remarks
Transmission mode	Balanced
Transmission line	3-wire system
Transmission speed (bps)	9600, 4800, 2400
Transmission distance	500m max. (For the MA500DIM and CMC410, the maximum distance is 300 m)
Communications flow	Half duplex
Synchronization	Start-stop synchronization
Data format	8 data bits, even parity, 1 stop bit 8 data bits, no parity, 2 stop bits
Error detection	Parity check, checksum
Station address	0 to 99 (setting 0 disables the communication function)
Network type	1: N (31 units max.)
Other	Follows the RS-485 standard

APPENDIX

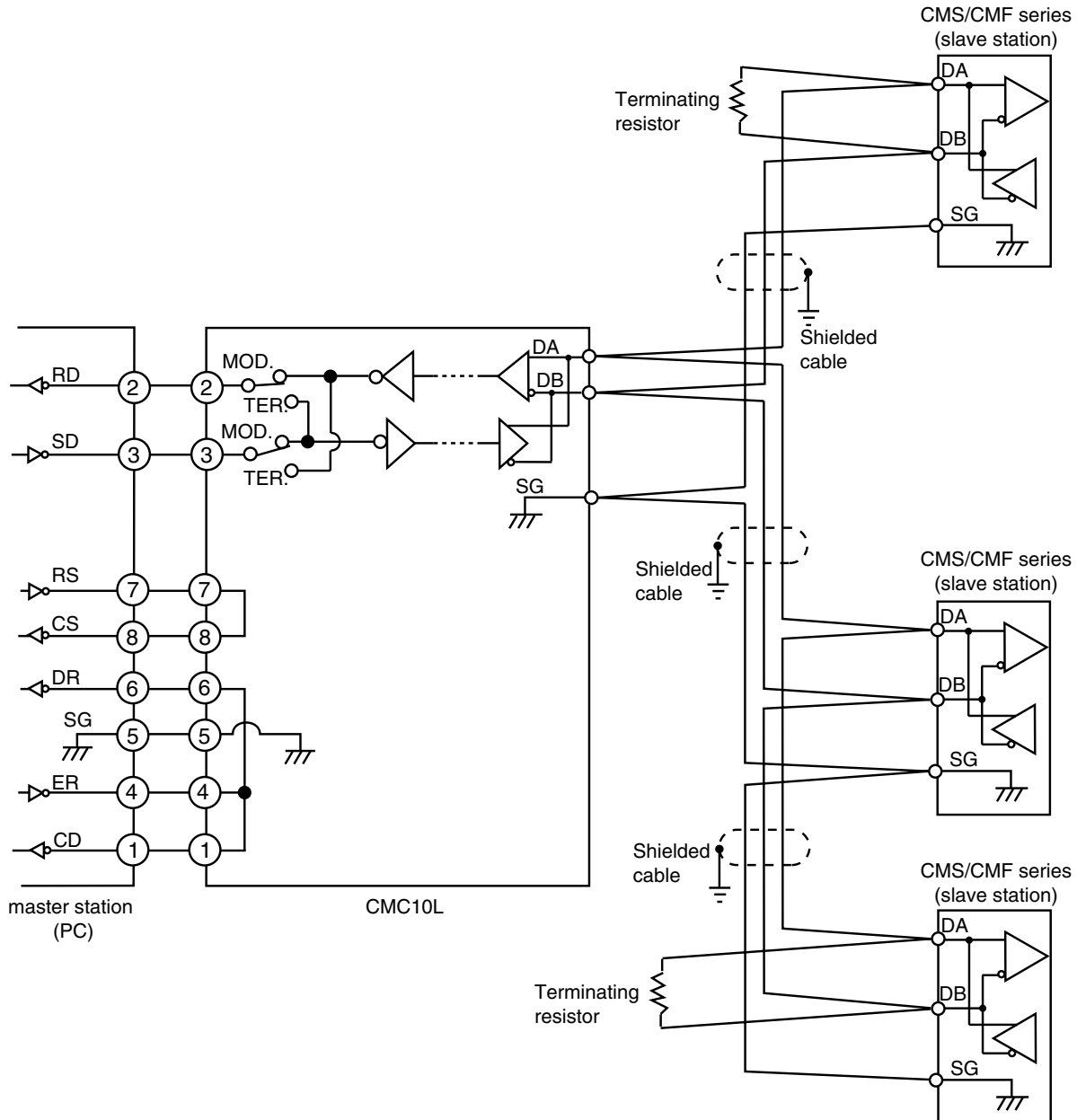
■ Code table

UPPER LOWER	0	1	2	3	4	5	6	7
0			SPACE	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	^

The shaded part () is not used for this communication system. (The codes to be used change every instrument.)

■ Connection with CMC10L

The diagram below shows an example of wiring using a D-SUB (9-pin) straight-through cable connecting to a master station (PC) in terminal mode.



Connect terminating resistors of $150 \Omega \pm 5 \%$, 1/2 W or more to the both ends of the communication path.
 Ground the shield to the FG at one end of the shield. Make sure that the shield is not grounded at both ends of the shield.

❗ Handling Precautions

- Be sure to connect SG terminals each other.
 Failure to do so might cause unstable communications.

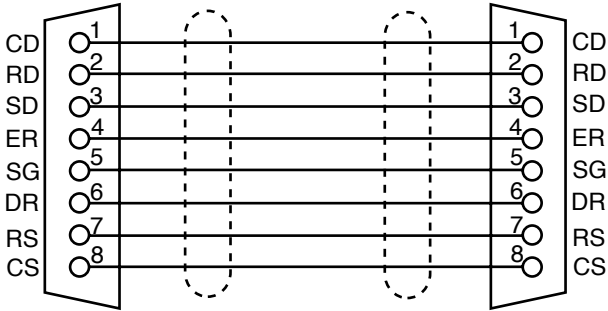
Connect the master station SD to the slave station RD and vice versa. Also, set the MODE switch in the CMC10L as shown in the table below to match the pin layout (modem or terminal) of the RS-232C connector on the host computer (master station) and the type of cable used (crossover (null-modem) or straight-through).

RS-232C	Cable type	MODE switch
TERMINAL	Straight-through	MODEM
TERMINAL	Crossover (null-modem)	TERMINAL
MODEM	Straight-through	TERMINAL
MODEM	Crossover (null-modem)	MODEM

● RS-232C cable

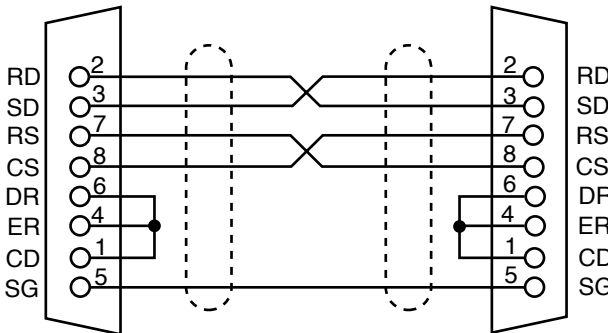
Straight through:

RS-232C cable with D-SUB (9-pin) connectors at each end connected such that each pin number connects to the same pin number at the other end (for example, pin 2 to pin 3, and pin 3 to pin 3).



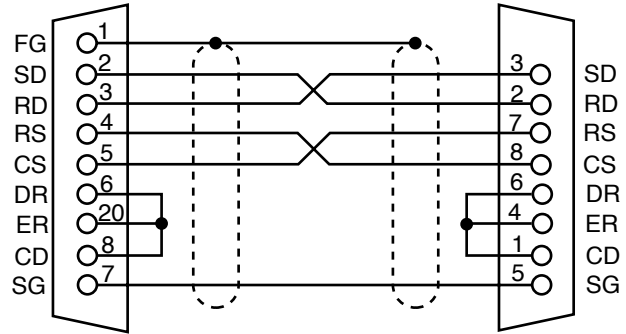
Crossover (null-modem):

RS-232C cable with D-SUB (9-pin) connectors at each end connected such that each signal pin connects to the opposite signal pin at the other end (for example, pin 2 to pin 3, and pin 3 to pin 2).



D-SUB 25-pin/9-pin converter cable:

RS-232C cable connected as shown below to convert from D-SUB (25-pin) to D-SUB (9-pin).



azbil

Yamatake Corporation
Advanced Automation Company

1-12-2 Kawana, Fujisawa
Kanagawa 251-8522 Japan

URL: <http://www.azbil.com>

Specifications are subject to change without notice. (08)

1st Edition: Issued in Sep. 2010 (G)