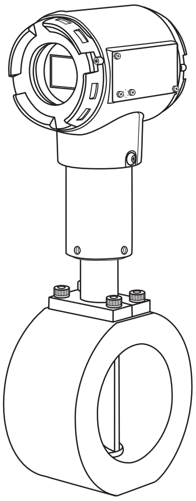
azbil

No. CP-SP-1183E

MVF Series
Micro Flow Vortex Gas Flowmeter

User's Manual
Communications



Thank you for purchasing the MVF Series Micro Flow Vortex Gas Flowmeter.

This manual contains information for ensuring correct use of the MVF Series communication functions.

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

Azbil Corporation

IMPORTANT

If it is necessary to change the parameters of the MVF Series frequently by communication, write data at addresses of RAM. The guaranteed data write count at the EEPROM addresses is limited to 100,000 times.
Note that the data in RAM is cleared, and the data in EEPROM is copied on RAM if the power supply to the MVF Series interrupted.

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

http://www.azbil.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 the azbil Group.

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

SAFETY PRECAUTIONS

■ About Icons

The safety precautions described in this manual are indicated by various icons. Please be sure you read and understand the icons and their meanings described below before reading the rest of the manual.

Safety precautions are intended to ensure the safe and correct use of this product, to prevent injury to the operator and others, and to prevent damage to property. Be sure to observe these safety precautions.

⚠WARNING Warnings are indicated when mishandling this product might result in death or serious injury.

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

■ Examples

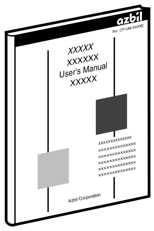
⚠	Use caution when handling the product.
🛇	The indicated action is prohibited.
❗	Be sure to follow the indicated instructions.

⚠CAUTION

❗	**Use the MVF 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.**
❗	**Be sure to turn the power off before connecting the unit. Failure to do so might cause malfunction.**
❗	**Wire this unit in compliance with the predetermined standards. Also wire the unit with specified cables and recognized installation methods. Failure to do so might cause malfunction.**
❗	**Make sure that wire scraps, chips or water do not enter inside the case of the unit. Failure to heed this caution may lead to malfunction.**
❗	**Firmly tighten the terminal screws at the torque listed in the specifications. Insufficient tightening of terminal screw might cause faulty operation.**
⊘	**Do not use unused terminals on the MVF as relay terminals.** **Doing so might cause faulty operation.**
⊘	**Do not disassemble the unit. Doing so might cause malfunction.**

The Role of This Manual

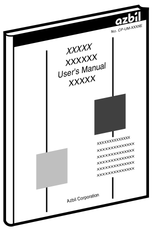
Two manuals have been prepared for the MVF Series. Read the manual according to your specific requirements. The below lists all the manuals that accompany the MVF Series and gives a brief outline of the manual. If you do not have the required manual, contact the azbil Group or your dealer.



MVF Series Micro Flow Vortex Gas Flowmeter
Manual No. CP-SP-1190E

First-time users of the MVF Series, and those in charge of maintenance or hardware design for incorporating a MVF Series flowmeter in instrumentation should read this manual.

This manual outlines the product, tells how to install, wire, and incorporate the product into instrumentation, and describes its operation, inspection and maintenance, troubleshooting, and hardware specifications.



MVF Series Micro Flow Vortex Gas Flowmeter : Communications
Manual No.CP-SP-1183E

This manual.
Those using the communications functions of the MVF Series should read this manual.
This manual describes an outline of communications, wiring, communications procedures, MVF Series communications data, troubleshooting, 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 MVF Series.

Chapter 2. WIRING

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

Chapter 3. SETTING

This chapter describes MVF 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 MVF Series.

Chapter 6. COMMUNICATION PROGRAM FOR MASTER STATION

This chapter gives precautions for programming and an example of a communication program for the MVF Series.

Chapter 7. TROUBLESHOOTING

This chapter describes checkpoints to diagnose failures in MVF Series communications.

Chapter 8. SPECIFICATIONS

This chapter lists communication specifications for the MVF Series.

APPENDIX

The appendix provides code tables.

Conventions Used in This Manual

The following conventions are used in this manual:

! Handling Precautions

: Handling Precautions indicate items that the user should pay attention to when handling the MVF Series.

📖 Note : Notes indicate useful information that the user might benefit by knowing.

(1), (2), (3) : The numbers with the parenthesis indicate steps in a sequence or indicate corresponding parts in an explanation.

Contents

Chapter 5. COMMUNICATION DATA TABLE

Chapter 6. COMMUNICATION PROGRAM FOR MASTER STATION

Chapter 7. TROUBLESHOOTING

Chapter 8. SPECIFICATIONS

APPENDIX

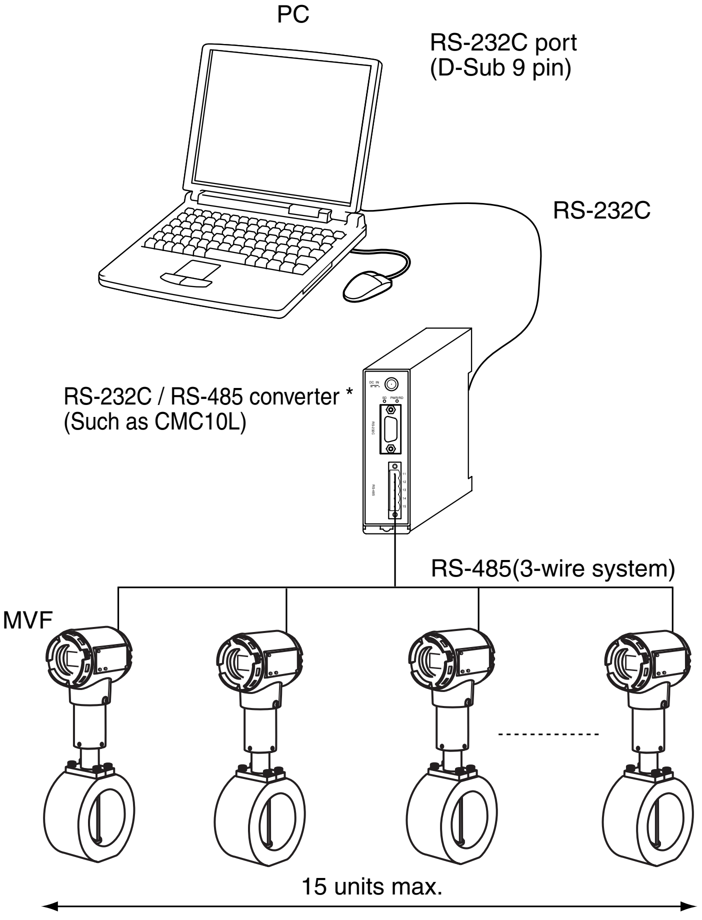
Chapter 1. OUTLINE

The MVF is able to communicate the set points and data with a PC and PLC through an RS-232C/RS-485 converter using a user-configured program. Also, the MVF is called a slave station and a PC or PLC is called a master station. The communication protocol can be selected from the Controller Peripheral Link (CPL) communications (Azbil Corporation's host communication protocol). This chapter describes the CPL communications.

■ Features

The features of the MVF's communications function are as follows:

- Up to 15 units can be connected to a single master station as a host device.
- When the communication specifications of the host device conform to the RS-232C interface, the communication converter CMC10L (sold separately) is required. The CMC10L allows the conversion between RS-232C and RS-485.
- Almost all of the device parameters can be communicated.
  For details on communication parameters, refer to;
  ☞ Chapter 5, COMMUNICATION DATA TABLE.



* : The CMC10L 001A000 is provided as an RS-232C / RS-485 converter.

Chapter 2. WIRING

⚠CAUTION

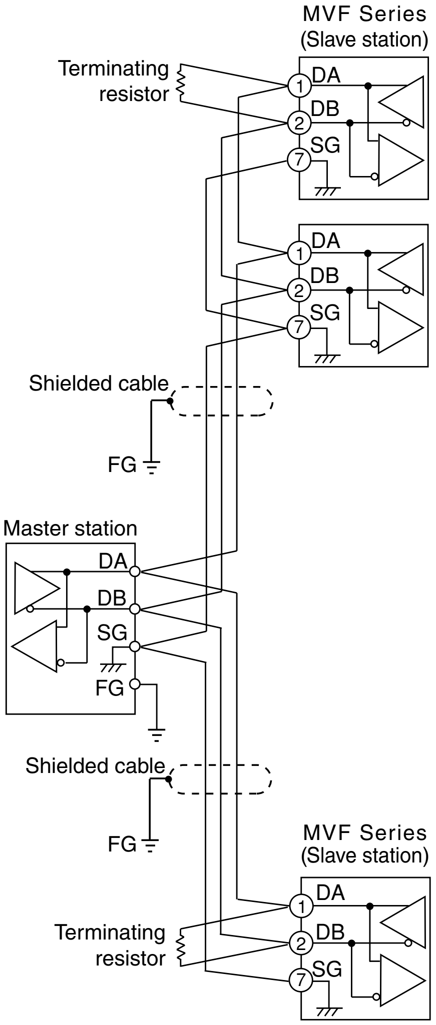
●	Be sure to turn the power off before connecting the unit. Failure to do so might cause malfunction.
●	Wire this unit in compliance with the predetermined standards. Also wire the unit with specified cables and recognized installation methods. Failure to do so might cause malfunction.
●	Make sure that wire scraps, chips or water do not enter inside the case of the unit. Failure to heed this caution may lead to malfunction.
●	Be sure to check that the wiring is correct before turning the power on. Incorrect wiring could cause damage or malfunction.
⊘	Do not disassemble the unit. Doing so might cause malfunction.

! Handling Precautions

- Wiring way except the RS-485 communications wiring, refer to; MVF Series Micro Flow Vortex Gas Flowmeter user's manual No.CP-SP-1190E.

■ RS-485 connection

An example of connection methods in such a case is shown below.

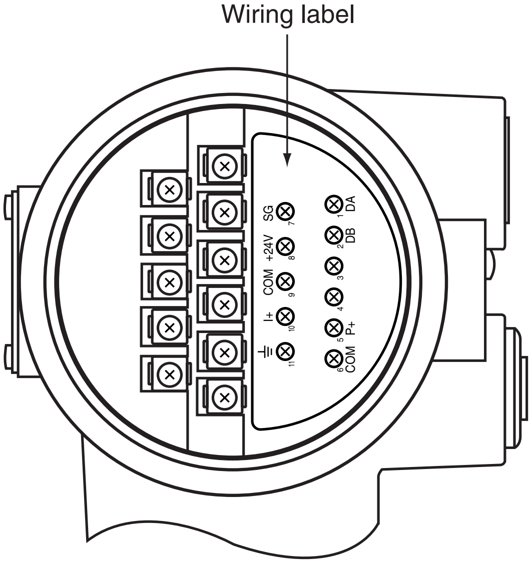


- Connect terminating resistors of 150Ω±5%, 1/2W min. at each end of the transmission line.
- The FG grounding must not be made at the both shielded wire ends but only at one location.
- Azbil Corporation's CMC10L001A000 can be used as a converter of the host station.

! Handling Precautions

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

■ Terminal layout



Terminal No.	Signal name	Description
1	DA	RS-485 communication DA
2	DB	RS-485 communication DB
3	Unused	Do not use
4	Unused	Do not use
5	P+	Pulse output (NPN open collector)
6	COM	Common
7	SG	RS-485 communication common
8	+24V	24V power
9	COM	Common
10	I+	4 to 20mA output
11	⏚	Ground terminal

! Handling Precautions

- Use the twisted-pair shielded cables for the wiring of RS-485 communications. Be sure to connect terminating resistors (150Ω 1/2W).
- Connect each terminal securely using crimp type terminal lugs to ensure firm contact area.
- Use crimp type terminal lugs applicable to M4 screw.
- Be sure that the tightening torque of terminal screw is less than 0.8N•m.
- For the wiring except RS-485 communications, use the JIS C 3401 cables for control (CVV etc.) of less than 2.2mm dia.

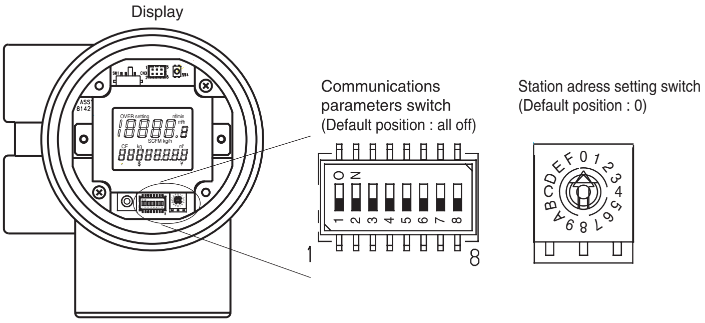
Chapter 3. SETTING

⚠CAUTION

❗ Be sure to turn the power off before setting the unit. Failure to do so might cause malfunction.

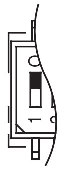
To operate the RS-485 communications functions, set the communications parameters and station adress of this unit to the setting of the master station.

■ Setting method


Display
Communications parameters switch (Default position : all off)
Station adress setting switch (Default position : 0)
ON
1 2 3 4 5 6 7 8
1
8
0 1 2 3 4 5 6 7 8 9 A B C D E F

❕ Handling Precautions

- Befor setting, set the [1] of the communications parameters switch to on. If the switch remaind off, setting will become invalid.



(1) Transmission speed setup

Set the [2] and [3]of communications parameters switch according to the transmission speed of the master station.

Transmission speed	Communications parameters switch	
	2	3
19200	OFF	OFF
9600	ON	OFF
4800	OFF	ON
2400	ON	ON

(2) Data format setup

Set the data format according to the data format of the master station.

Data format	Communications parameters switch 4
8-bit data, 1 stop bit, even parity	OFF
8-bit data, 2 stop bits, no parity	ON

(3) Station adress setup

Set the station adress from 1 to 15 with the station adress setting switch. The RS-485 communications do not work when the station adress is remining 0.
If the connecting equipments exceed 16 units or more, set the station address of over 16 to the equipment except the MVF.

Chapter 4. COMMUNICATION PROCEDURE

4 - 1 Outline of Communication

■ 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.

4 - 2 Message Structure

■ Message structure

The following shows the message structure:
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 the communication such as the destination of the communication message and the check information of the message.
- 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.

02H			58H		03H		0DH	0AH
STX			X		ETX		CR	LF
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)
Data link layer				Application layer	Data link layer			
1 frame								

(1) STX (start of message)
(2) Station address
(3) Sub-address
(4) Device code
(5) Send message = command,
response message = response

(6) ETX (end of command/response)
(7) Checksum
(8) CR (delimiter)
(9) LF (delimiter)

■ 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 device sends the response message only when (1) message structure, station address, sub-address, checksum and message length of a single frame in the data link layer are all correct. If even one of these is incorrect, no response messages are sent, and the device waits for new message.

- Number of word addresses accessible by a single frame

Type	Description of command	RAM area	EEPROM area
RS	Decimal format read command	1 to 10	1 to 10
WS	Decimal format write command	1 to 10	1 to 10

● List of data link layer data definitions

The following list shows the definitions for data in the data link layer:

Data name	Character code	Number of characters	Meaning of data
STX	02H	1	Start of message
Station address	0 to 0FH are expressed as hexadecimal character codes.	2	Identification of device to communicate with
Sub-address	"00" (30H, 30H)	2	No function
Device code	"X" (58H) or "x" (78H)	1	Device type
ETX	ETX (03H)	1	End position of the application layer
Checksum	00H to FFH are expressed as two-digit hexadecimal character codes.	2	Checksum of message
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. For this reason, the device returns to the initial state whatever reception state it was in, and processing is started on the assumption that the STX, the first character, has been received. The purpose of this is to enable recovery of the device's response at the next correct message (e.g. RETRY message) from the master station in the event that noise, for example, causes an error in the sent message.
- Station address

  Of the messages sent by the master station, the device creates response messages only when station addresses are the same. Station addresses in the messages are expressed as two-digit hexadecimal characters.

  The station address is set up by the station address setting switch. However, when the station address is set to 0, the device creates no response even if station addresses match.

  The device returns the same station address as that of the received message.
- Sub-address

  This unit does not use the sub-address. For this reason, set "00" (30H 30H).

  The device returns the same sub-address as that of the received message.
- Device code

  The device sets X (58H) or x (78H) as the device code. This code is determined for each device series, and other codes cannot be selected. The device returns the same device code as that of the received message. X (58H) is used as the default, and x (78H) is used for judging the message as the resend message.
- 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 two's complement of the low-order one byte of the addition result.
(3) Convert the obtained two's complement to a two-byte ASCII code.

The following is a sample checksum calculation:
[Sample message]

STX:	02H
'0':	30H (first byte of the station address)
'1':	31H (second byte of the station address)
'0':	30H (first byte of the sub-address)
'0':	30H (second byte of the sub-address)
'X':	58H (device code)
'R':	52H (first byte of the command)
'S':	53H (second byte of the command)
(omitted)	
ETX:	03H

(1) Add the character codes in the message from STX through ETX in single byte units.
The add operation in single byte units is as follows:
02H + 30H + 31H + 30H + 30H + 58H + 52H + 53H + • • • + 03H.
Assume that the result is 376H.
(2) The low-order one byte of the addition result 376H is 76H. The two's complement of 76H is 8AH.
(3) Convert the obtained 8AH to a two-byte ASCII code.
The result is:
'8': 38H
'A': 41H,
and the two bytes, '8'(38H) and 'A'(41H), are the checksum.

- CR/LF

This indicates the end of the message. Immediately after LF is received, the device enters a state allowed to process the received message.

■ Application layer

The table below shows the configuration of the application layer.

Item	Description
Command	"RS" (decimal number format continuous address data read command)
	"WS" (decimal number format continuous address data write command)
Data delimiter	RS, WS: "," (comma)
Data address	RS, WS: "501W", etc.
Number of read data	Numerical value of characters expressed as "1" for example
Numerical value to be written	RS, WS: Numerical value of characters expressed as "100" for example

4 - 3 Description of Commands

■ Continuous data read command (RS command)

This command reads data of continuous addresses by a single command.

● 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.

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

Application layer

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

● Response message

If the message is correctly received, a response message corresponding to the command content is returned.
The figure below shows the structure of the application layer of the response message when the data is read.

• Normal termination (reading of single data item)

0	0	,				
(1)		(2)	(3)			

• Normal termination (reading of multiple data items)

0	0	,					,		,				
(1)		(2)	(3)				(2)	(4)	(2)	(5)			

• Warning termination

X	X	,					,		,				
(1)		(2)	(3)				(2)	(4)	(2)	(5)			

• Error termination

X	X
(1)	

The termination code is entered at XX.
For details of codes, refer to;
☞ 4-4, Termination Code Table (on page 12).

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

● Maximum number of read data per message

Up to 10 words for both RAM and EEPROM areas

■ Continuous data write command (WS command)

This command writes data to continuous addresses.

● Send message

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

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.
For details of codes, refer to;
☞ 4-4, Termination Code Table (on page 12).

(1) Termination code

● Maximum number of write data per message

Up to 10 words for both RAM and EEPROM areas

4 - 4 Termination Code Table

■ Normal termination and warning termination

If the command is processed successfully, the MVF sends normal termination code. When an error occurred in the application layer, an abnormal termination code is returned as a response message.

Termination code	Type	Contents and action
00	Normal	Normal termination
20	Warning	Wrong number of data Processing is continued except for the relevant address.
21	Warning	Data address alarm Processing is continued except for the relevant address.
22	Warning	Excess range of data Though the numeric data exceeds the normal range in write command, processing is continued except for the relevant address.
23	Warning	Write disable by the unit The setting of the unit disables a write command.

■ Error termination

Termination code	Type	Contents and action
40	Error	Wrong number of data Number of read data or number of write data is incorrect.
41	Error	Data address alarm The front of data address is incorrect
42	Error	Excess range of data Numeric data exceeds the normal range in write command.
43	Error	Write disable entirely by the unit The setting of the unit disables a write command.
99	Error	An undefined command The 2 letters in the front of application layer is an undefined command.

If no data is written, the MVF sends an error termination code.

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

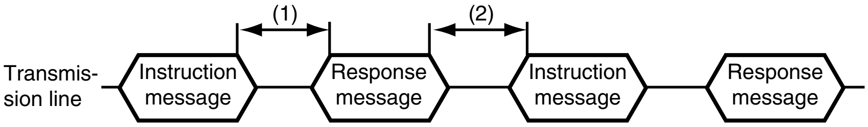
The maximum response time from the end of the instruction message transmission by the master station until when the master station receives a response message from the slave station is two seconds ((1) in the figure below). So, the response monitor time should be set to two seconds.

Generally, when a response time-out occurs, resent the instruction message.

For details, see Chapter 6 "COMMUNICATION PROGRAM FOR MASTER STATION."

● Transmission start time

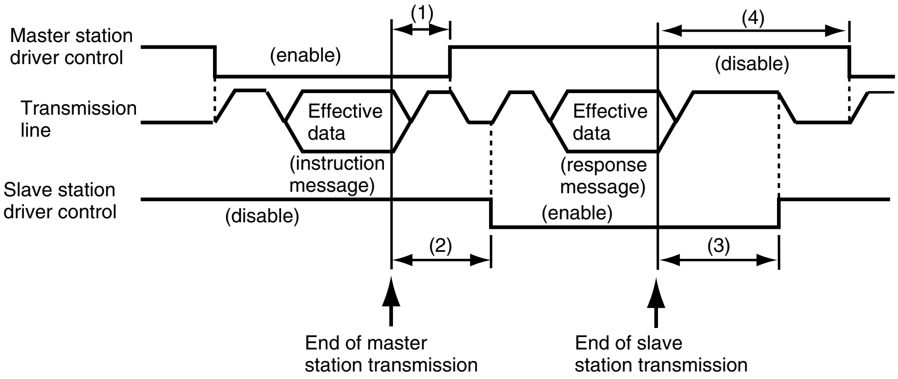
A wait time of 10ms is required before the master station starts to transmit the next instruction message (to the same slave station or a different slave station) after the end of receiving response message ((2) in the figure below).



(1) End of master station transmission - Transmission start time of slave station = Max. 2000ms
(2) End of slave station transmission - Transmission start time of master station = Min. 10ms

■ RS-485 driver control timing specifications

When the transmission/reception on the RS-485 3-wire system is directly controlled by the master station, care should be paid to the following timing:



(1) End of master station transmission - Driver disable time = 500µs max.
(2) End of slave station reception - Driver enable time = 15ms min.
(3) End of slave station transmission - Driver disable time = 10ms max.
(4) End of master station reception - Driver enable time = 10ms min.

■ Other precautions

- •The time required for the master station to finish the transmittal of instruction message and for the slave station to start the transmittal of response message becomes longer if the number of data to write and read increases. When the faster response time is required by the slave station, make sure to keep the number of data to read / write at the minimum in one message.
- •When the number of data is one data to read / write in one message, the time required for the master station to finish the instruction message and for the slave station to transmit the response message is about 30ms.

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:

- Device data
- Operating status data
- Instantaneous flow rate data (such as instantaneous flowrate)
- 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

If it is necessary to change the parameters of the MVF Series frequently by communication, write data at addresses of RAM. The guaranteed data write count at the EEPROM addresses is limited to 100,000 times.
Note that the data in RAM is cleared, and the data in EEPROM is copied on RAM if the power supply to the MVF Series interrupted.

■ Communication data storage memory

● Memory type

The communication data are stored in 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, whereas data write operations are limited to a total of 100,000 times owing to device characteristics.

● Communication object memory

In communication, it is necessary to read/write data from/into the abovementioned two types of memory according to the purpose and use. There is a difference between the object memories as follows:

- RAM: Data is read/written from/into RAM only. If the power supply is turned off after writing data into RAM, and then it is turned on again, the data in EEPROM is copied on RAM, so the data in RAM becomes the same as in EEPROM.
- EEPROM: Data are written in both RAM and EEPROM.

■ Data address

The data addresses are allocated as shown in the table below.

Communication data	RAM		EEPROM	
	Offset value Decimal	Address Decimal	Offset value Decimal	Address Decimal
Device data	1000	1001 to 1199	4000	4001 to 4199
Operating status data	1200	1201 to 1399	4200	4201 to 4399
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

■ Number of data read / write word

The number of data which can be continuously read/written by once communication is as shown in the table below.

	RAM	EEPROM
Read	1 to 10	1 to 10
Write	1 to 10	1 to 10

■ 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.

5 - 2 Communication Data 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:
❍ : Enable
× : Disable

■ Device data

Item	Data range	RAM			EEPROM			Remarks
		Address	R	W	Address	R	W	
Gas type	0: Air/Nitrogen/Argon 1: Oxygen 2: Carbon dioxide 3: Natural gas 13A (LNG) / Methane 4: Propane 5: Butane 7: User setting gas	1001	❍	×	4001	×	×	Gas type can be changed by function setup (address 2001 and 5001)
Pipe size	0: 50A 1: 80A 2: 100A 3: 150A	1002	❍	×	4002	×	×	-
Multiplier of instantaneous flowrate	1 : 0.1 2 : 0.2 5 : 0.5 10: 1.0	1003	❍	×	4003	×	×	Multiplying an instantaneous mass flowrate (address 1201) and this multiplier together becomes an instantaneous flowrate.
Decimal point position of integrated flowrate	0: xxxxxxx.x 1: xxxxxxxx	1004	❍	×	4004	×	×	This item shows the decimal point position of 8-digit indication (upper 4-digit and middle 4-digit). The upper 4-digit is gained from address 1603 and the lower 4-digit is from address 1602.

■ Operating status data

Item	Data range	RAM			EEPROM			Remarks
		Address	R	W	Address	R	W	
Instantaneous mass flowrate	0 to 65535	1201	❍	×	4201	×	×	Multiplying this value (address 1201) and the multiplier (address 1003) together becomes an instantaneous flowrate.
instantaneous volume flowrate	MVF050: 0 to 3900 MVF080: 0 to 8600 MVF100: 0 to 13250 MVF150: 0 to 28500	1202	❍	×	4202	×	×	Multiplying this value (address 1202) and the 0.1 (m^3/h) together becomes an instantaneous volume flowrate.
Temperature	-15 to +60 (˚C)	1203	❍	×	4203	×	×	-
Pressure	-75 to +1100 (kPa)	1204	❍	×	4204	×	×	-
Error status	* 1	1205	❍	×	4205	×	×	The status is indicated in decimal.
Alarm status	* 2	1206	❍	×	4206	×	×	The status is indicated in decimal.

* 1: Bit configuration of error status (address 1205) 0: Normal
1: Error occurs

Bit No.	Description
0	Flow sensor error
1	Temperature sensor error
2	Pressure error sensor
3	Memory data error
4	Undefined (fixed 0)
5	Undefined (fixed 0)
6	Undefined (fixed 0)
7	Undefined (fixed 0)

* 2: Bit configuration of alarm status (address 1206) 0: Normal
1: Alarm occurs

Bit No.	Description
0	Flowrate upper limit alarm (45 m/s or more)
1	Temperature lower limit alarm (-15 ˚C or less)
2	Temperature upper limit alarm (60 ˚C or more)
3	Pressure lower limit alarm (-50 kPa or less)
4	Pressure upper limit alarm (1000 kPa or more)
5	Undefined (fixed 0)
6	Undefined (fixed 0)
7	Undefined (fixed 0)

■ Integrated flowrate data

Item	Data range	RAM			EEPROM			Remarks
		Address	R	W	Address	R	W	
Integrated flowrate lower 2-digit	0 to 99 (BCD code) Unit: m^3 or kg	1601	○	×	4601	×	×	The decimal point position of 8-digit integrated flowrate (upper 4-digit and middle 4-digit) is designated by an address 1004.
Integrated flowrate middle 4-digit	0 to 9999 (BCD code) Unit: m^3 or kg	1602	○	×	4602	×	×	
Integrated flowrate upper 4-digit	0 to 9999 (BCD code) Unit: m^3 or kg	1603	○	×	4603	×	×	
Converted rate lower 4-digit	0 to 9999 (BCD code)	1604	○	×	4604	×	×	Multiplying the 8-digit integrated flowrate and the rate conversion factor (address 2208) becomes converted rate.
Converted rate upper 4-digit	0 to 9999 (BCD code)	1605	○	×	4605	×	×	
Integrated flowrate reset	0 or 1	1606	○	○	4606	×	×	When reading this address, the zero is read out at all times. To reset the integrated flowrate, write 1 to this address.

* 1: The integrated flowrate lower 2-digit (address 1601) is not backed up during the power off.

* 2: The integrated flowrate configured as follow:

Display digit (MVF050)	10^6	10^5	10^4	10^3	10^2	10	1	10^{-1}	10^{-2}	10^{-3}
(Except MVF050)	10^7	10^6	10^5	10^4	10^3	10^2	10	1	10^{-1}	10^{-2}
Communication data	Upper 4-digit Address 1603				Middle 4-digit Address 1602				Lower 2-digit Address 1601	

Though the integrated flowrate is set the 8-digit display (upper 4-digit and middle 4-digit) in the default, the lower 2-digit can be displayed by the display resolution of integrated flowrate (address 2012 or 5012).

Example: The integrated flowrate is 12345678.90 m^3 in the MVF080.

Address	Data (BCD)
1601	90
1602	5678
1603	1234

■ Function setup data

Item	Data range	RAM			EEPROM			Remarks
		Address	R	W	Address	R	W	
Gas type setting	0: Air/Nitrogen/Argon 1: Oxygen 2: Carbon dioxide 3: Natural gas 13A (LNG) / Methane 4: Propane 5: Butane 7: User setting gas	2001	○	○	5001	○	○	The oxygen could not be set except the oxygenic model. When the user setting gas is selected in this setting, the setting of gas specific gravity (address 2207 or 5207) is used as a specific gravity.
Temperature and pressure correction setting	0: No correction 1: Temperature correction only 2: Pressure correction only 3: Both temperature and pressure correction	2001	○	○	5001	○	○	-
Display mode	0: m^3/h and m^3 1: kg/h and kg	2003	○	○	5003	○	○	-
Undefined	0	2004	○	×	5004	○	×	* 1
Mode setup of 4-20 mA output	0: Instantaneous mass flowrate 1: instantaneous volume flowrate 2: Temperature 3: Pressure	2005	○	○	5005	○	○	When "0" is selected, the mass flowrate at 4mA output (address 2209 or 5209) and the mass flowrate at 20mA (address 2210 or 5210) are set as output range. When "1" is selected, the volume flowrate output range (address 2215 or 5215) is set as output range.
Burnout setup of 4-20 mA output	0: Downscale 1: Upscale	2006	○	○	5006	○	○	When "1" is selected, the burnout setup value (address 2211 or 5211) is output.
Undefined	0	2007	○	×	5007	○	×	*1
Undefined	0	2008	○	×	5008	○	×	*1
Integrated pulse setup (for MVF50)	0: 0.01 m^3 or 0.01 kg 1: 0.1 m^3 or 0.1 kg 2: 1 m^3 or 1 kg 3: 10 m^3 or 10 kg	2009	○	○	5009	○	○	-
Integrated pulse setup (except for MVF50)	0: 0.1 m^3 or 0.1 kg 1: 1 m^3 or 1 kg 2: 10 m^3 or 10 kg 3: 100 m^3 or 100 kg							
Upper LCD display mode setup	0: Instantaneous mass flowrate 1: Instantaneous volume flowrate 2: Temperature 3: Pressure	2010	○	○	5010	○	○	-
Lower LCD display mode setup	0: Integrated flowrate 1: Conversion rate 2: Temp. / Pressure 3: Alternate display of integrated flowrate and temp. / pressure	2011	○	○	5011	○	○	-

Item	Data range	RAM			EEPROM			Remarks
		Address	R	W	Address	R	W	
Display resolution of integrated flowrate	0: xxxxx.xxx 1: xxxxxx.xx 2: xxxxxxx.x (for MVF050) 0: xxxxxx.xx 1: xxxxxxx.x 2: xxxxxxxx (except for MVF050)	2012	❍	❍	5012	❍	❍	-
Undefined	0	2013	❍	×	5013	❍	×	* 1
Monetary unit setup	0: Yen 1: Dollar 2: Euro	2014	❍	❍	5014	❍	❍	-
Temperature correction	0: Measured temp. 1: User setting temp.	2015	❍	❍	5015	❍	❍	When "1" is selected, the flowrate is corrected by the user setting temperature (address 2216 or 5216).
Pressure correction	0: Measured pressure 1: User setting pressure	2016	❍	❍	5016	❍	❍	When "1" is selected, the flowrate is corrected by the user setting pressure (address 2217 or 5217).
Undefined	0	2017	❍	×	5017	❍	×	*1
Undefined	0	2018	❍	×	5018	❍	×	
Undefined	0	2019	❍	×	5019	❍	×	
Undefined	0	2020	❍	×	5020	❍	×	
Undefined	0	2021	❍	×	5021	❍	×	
Undefined	0	2022	❍	×	5022	❍	×	
Undefined	0	2023	❍	×	5023	❍	×	
Undefined	0	2024	❍	×	5024	❍	×	
Undefined	0	2025	❍	×	5025	❍	×	
Undefined	0	2026	❍	×	5026	❍	×	
Undefined	0	2027	❍	×	5027	❍	×	
Undefined	0	2028	❍	×	5028	❍	×	
Undefined	0	2029	❍	×	5029	❍	×	
Station address	0 to 15	2030	❍	×	5030	❍	×	These settings are read-only. To set the station address and communications parameters, see chapter 3 SETTING.
Transmission speed	0: 19200 bps 1: 9600 bps 2: 4800 bps 3: 2400 bps	2031	❍	×	5031	❍	×	
Data format	0: 8-bit, even parity, 1 stop bit 1: 8-bit, no parity, 2 stop bits	2032	❍	×	5032	❍	×	

*1: When reading an undefined address, the zero is read out at all times. And, when writing an undefined address, the unit returns normal response though no data is written.

■ Parameter setup data

Item	Data range	RAM			EEPROM			Remarks
		Address	R	W	Address	R	W	
Reference temperature	0 to 35 ˚C	2201	❍	❍	5201	❍	❍	-
Reference pressure	90.0 to 300.0 kPa	2202	❍	❍	5202	❍	❍	* 1
Atmospheric pressure	90 to 110 kPa	2203	❍	❍	5203	❍	❍	-
Flow dead band setting	(0 to 30 %FS) m³/h or kg / h	2204	❍	❍	5204	❍	❍	Multiplying the full-scale flowrate by the percentage in the parentheses becomes the flow dead band. (It's depend on a model)* 2
Bias flowrate	(-10 to +10 %FS) m³/h or kg / h	2205	❍	❍	5205	❍	❍	Multiplying the full-scale flowrate by the percentage in the parentheses becomes the bias flowrate. (It's depend on a model)* 2
Conversion factor	0.100 to 9.999	2206	❍	❍	5206	❍	❍	* 1
Specific gravity of gas	0.100 to 9.999	2207	❍	❍	5207	❍	❍	This setting parameter is enable when "7" is selected in the gas type setting (address 2001 or 5001). * 1
Rate conversion factor	0.01 to 99.99	2208	❍	❍	5208	❍	❍	* 1
Mass flowrate at 4mA output	(0 to 99 %FS) m³/h or kg / h	2209	❍	❍	5209	❍	❍	Multiplying the full-scale flowrate by the percentage in the parentheses becomes the mass flowrate at 4mA output. (It's depend on a model.) This setting parameter is enable when "0" is selected in the mode setup of 4-20 mA output (address 2005 or 5005). * 2
Mass flowrate at 20mA output	(1 to 100 %FS) m³/h or kg / h	2210	❍	❍	5210	❍	❍	Multiplying the full-scale flowrate by the percentage in the parentheses becomes the mass flowrate at 20mA output. (It's depend on a model.) This setting parameter is enable when "0" is selected in the mode setup of 4-20 mA output (address 2005 or 5005). * 2
Burnout setup	0 to 125 %	2211	❍	❍	5211	❍	❍	This setting parameter is enable when "1" is selected in the burnout setup of 4-20 mA output (address 2006 or 5006).

Item	Data range	RAM			EEPROM			Remarks
		Address	R	W	Address	R	W	
Undefined	0	2212	❍	×	5212	❍	×	* 3
Undefined	0	2213	❍	×	5213	❍	×	* 3
Undefined	0	2214	❍	×	5214	❍	×	* 3
Volume flowrate output range	10 to 150 %FS	2215	❍	❍	5215	❍	❍	This setting parameter is enable when "1" is selected in the mode setup of 4-20 mA output (address 2005 or 5005). About the volume flowrate full-scale, see *2.
User setting temperature	-15 to +60 C˚	2216	❍	❍	5216	❍	❍	This setting parameter is enable when "1" is selected in the temperature correction (address 2015 or 5015).
User setting pressure	-50 to +1000 kPa	2217	❍	❍	5217	❍	❍	This setting parameter is enable when "1" is selected in the pressure correction (address 2016 or 5016).

* 1: Setup data must be taken off the decimal point.

* 2: Full-scale flowrate

Model No.	Mass flowrate full-scale [m^3/h] or [kg/h]	Volume flowrate full-scale [m^3/h]
MVF050	8000	240
MVF080	16000	520
MVF100	24000	800
MVF150	48000	1700

* 3: When reading an undefined address, the zero is read out at all times. And, when writing an undefined address, the unit returns normal response though no data is written.

Chapter 6. COMMUNICATION PROGRAM FOR MASTER STATION

6 - 1 Precautions for Programming

- The maximum response moniter time of the instrument is 2 seconds. So, set the response time-out to 2 seconds.
- If no response is obtained within 2 seconds, retransmit the same message.
- The above retransmission is required since a message may not be properly transmitted due to noise or the like during communications.

Note

When “X” is used as the device code in the instruction message, the device code of the response message also becomes “X”. Likewise, the code in the response message becomes “x” when “x” is used as the device code in the instruction message.

Use the “X” and “x” device codes alternately during message retransmission from the master station, to make it easier to identify whether the received response message is the current one or the preceding one.

6 - 2 Examples of Communication Program

The program in this section is written in Borland's C++Builder5.0 or Borland C++Compiler5.5 for Windows95/98/NT/2000.

This program is given here as a reference when the user makes a program, and does not assure all the operations.

You can download Borland C++Compiler5.5 from Borland Home Page.

■ Before executing the program

Make sure to check the settings for communications type, station address, transmission speed and data format of the MVF.

■ Executing the program

This program is used for reading and writing data. When the program is executed, the application layers of the instruction message and response message communicated are indicated.

```
command:RS,1001W,2
result:00,0,0
command:WS,2001W,2
result:00
```

Sample indication of execution results

● Processing of the sample program

- Communication settings
  Call open() and initialize the RS-232C serial port.
- Command execution
  Set a desired character string in 'command' and call AppCPL().

■ Data read/write sample program

Handling Precautions

Azbil Corporation won't be absolutely responsible for any trouble caused by applying this program sample.

```
//---------------------------------------------------------------------------
// 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, CLRDTR ) ){
            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( "unsıgned long sıze" );
        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( 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;
}
//---------------------------------------------------------------------------
```

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 MVF meet those for the host computer.

If any one of the following setting items is different between both stations, communication is disabled:

The underlined items mean that they can be set on the MVF side.

Transmission seed	: 19200, 9600, 4800, 2400bps
Data length	: 7, 8 bits
Parity	: No parity, odd parity, even parity
Stop bit	: 1 stop bit, 2 stop bits

(4) Check if the destination address of the command frame transmitted from the host computer meets the address set to the MVF Series.

The address of the MVF set to “0” for factory setting.

Even when the destination address of the command frame is set to 00 (30H, 30H), the MVF does not respond to such a message.

(5) Use the upper-case character codes for all the character codes other than the device code (“X” or “x”).

(6) Are those multi-dropped units belong operated themselves with different station address setups?

(7) Is the communication timing conformed with the Chapter 4-5 Timing Specifications (page 13)?

Chapter 8. SPECIFICATIONS

■ RS-485 specifications

Item	Remarks
Transmission mode	Balanced
Transmission line	3-wire system
Transmission speed (bps)	2400, 4800, 9600, 19200
Transmission distance	300m max.
Communications flow	Half duplex
Synchronization	Start-stop synchronization
Data format	8 data bits, 1 stop bit, even parity 8 data bits, 2 stop bits, no parity
Error detection	Parity check, checksum
Station address	0 to 15 (Communication function is inhibited when set to “0”.)
Network type	1: N (31 units max.)
The other items	Conforms to RS-485 interface specifications.

APPENDIX

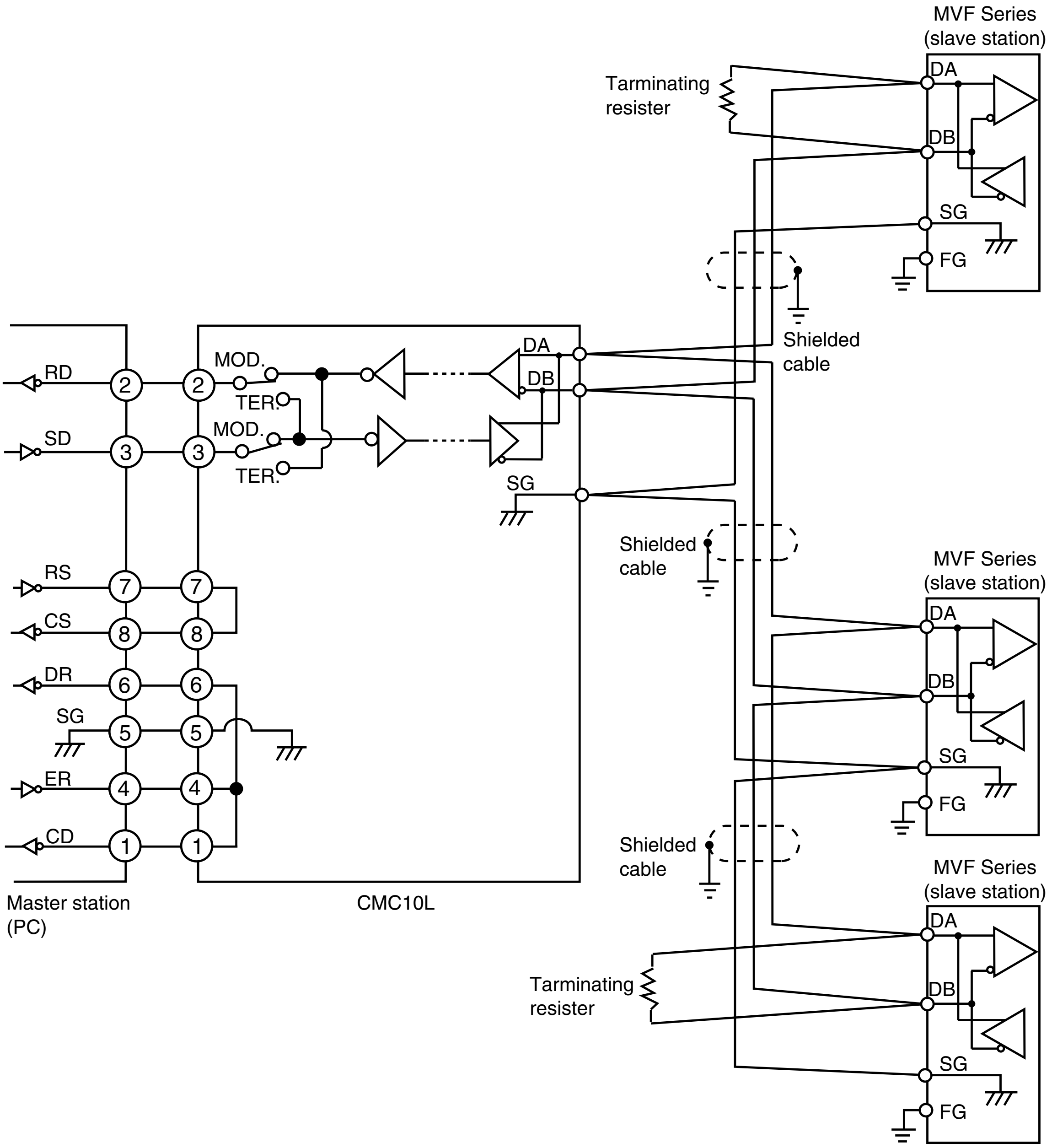
■ Code table

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 parts of the table above (▨) are not used for this communication system.

■ Connection with CMC10L

The CMC10L001A000 is available as an RS-232C/RS-485 (3-wire system) converter from Azbil Corporation's. The following diagram shows an example of wiring using a straight cable for a host computer in the terminal mode:



- Connect terminating resistors of 150Ω±5%, 1/2W min. 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.

! Handling Precautions

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

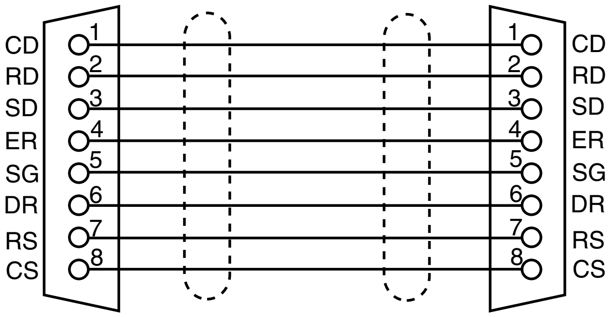
Connect the master station SD to the slave station RD, and the master station RD to the slave station SD.

To execute this connection, set the MODE switch provided in the CMC10L as shown in the following table in accordance with the host computer side RS-232C connector pin arrangement (modem/terminal) and the type of cable (crossover/straight) used:

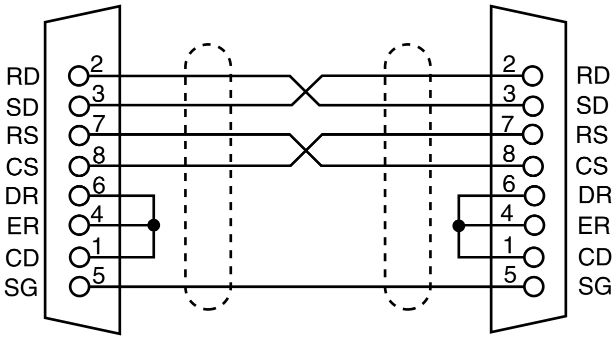
RS-232C	Cable type	MODE switch
TERMINAL	Straight	MODEM
TERMINAL	Crossover	TERMINAL
MODEM	Straight	TERMINAL
MODEM	Crossover	MODEM

● **RS-232C cable**

Straight: An RS-232C cable with a D-Sub (9-pin) connector at each end where pins with the same number are mutually connected (for example, pin 2 to pin 2, and pin 3 to 3)

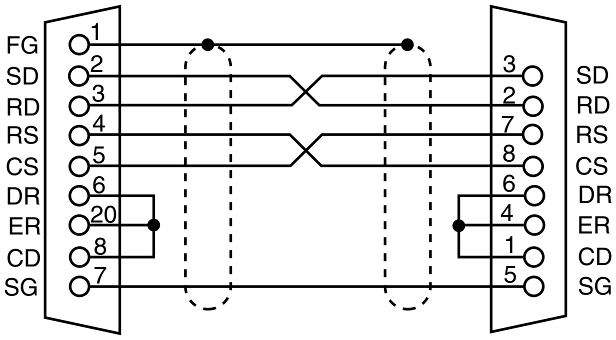


Crossover: An RS-232C cable with a D-Sub (9-pin) connector at each end where different number pins are connected (for example, pin 2 to pin 3, and pin 3 to pin 2)



D-Sub (25-pin) – D-Sub (9-pin) conversion cable:

An RS-232C cable for conversion between D-Sub (25-pin) and D-Sub (9-pin)



Revision History

Printed date	Manual Number	Edition	Revised pages	Description
June 2008	**CP-SP-1183E**	**1st Edition**		
Apr. 2012		**2nd Edition**		Company name changed.

Azbil 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. (09)

1st edition: June 2008 (A)
2nd edition: Apr. 2012 (M)