

**Introducing the G9000 series**

# **Capabilities of Motionnet**

***NPM*** Nippon Pulse Motor Co., Ltd.

<June, 2005>

## Introduction

The G9000 series (Motionnet) is a group of ASIC products that provide high-speed serial communication.

By employing the proprietary "Motionnet" protocol, the series can be used to easily construct a field bus based on the RS485 standard.

The targets and uses of this technology are in the following applications.

- **Reduce the wiring needed for axis control, such as in stepper motors**

Axis control is the field in which NPM shows its greatest advantage. Technologies requiring axis control devices, such as the PCL series, can significantly reduce the wiring required.

- **Reduce the wiring required for peripheral devices communicating with the CPU**

NPM has a device that emulates the operational control signals from a CPU. Using this device, a peripheral device located far away can be controlled with a reduced number of wires.

- **Reduce the wiring needed for data exchange between CPUs.**

If you want to use data stored on another CPU located somewhere else, you can do that with this technology.

- **As a general field bus**

Of course, this technology can also be used to control general I/O lines.

The "Motionnet" protocol is designed to manage the "motion" of each individual function located in separate places on a field bus. For example it can control motor operation and reproduce CPU operation signals. That is why this protocol is named "Motionnet."

We want to take this opportunity to convey the features and capabilities of Motionnet.

"Motionnet" is a registered trademark of Nippon Pulse Motor Co., Ltd.

## Table of index

Basic version .....	1
1. What products are available in the G9000 series and what functions do they have? .....	2
1-1. G9001: Center device .....	2
1-2. G9002: I/O device .....	2
1-3. G9003: PCL device .....	2
1-4. G9004: CPU emulation device .....	3
2. How is a Motionnet constructed? .....	4
3. How many local devices can be connected .....	4
4. How is communication done? .....	5
4-1. Cyclic communication .....	5
4-2. Transient communication = Data communication .....	6
5. What kind of data can be sent using cyclic communication? .....	7
6. What is status? .....	8
7. How fast is a communication cycle? .....	8
8. What is the longest communication distance possible using Motionnet? .....	9
9. Do device addresses have to be allocated sequentially without gaps? .....	9
10. What is RS485? .....	10
11. Why are pulse transformers needed? .....	10
12. What is the role of the termination resistance? .....	11
13. Types of CPUs that can be used to control the center device .....	12
14. Why does the center device have two serial inputs (SIA, SIB)? .....	13
Operation version .....	15
1. How easy is it to use Motionnet? .....	16
2. How is data communication (transient communication) performed? .....	17
3. How is data written to the send FIFO buffer? .....	18
4. How is data read from the receiving FIFO buffer? .....	18
5. How is data about the G9002 (I/O device) input port read? .....	19
6. How is data output from the G9002 output port (I/O device)? .....	20
7. How can you check the status information for the G9003 and G9004? .....	21
8. How to check the device type for devices that are connected to a communication circuit .....	22
9. How can an input port change on an I/O device be identified using the interrupt function? .....	23
9-1. Assign a device to be monitored for a change in its input value .....	23
9-2. Identify a device whose input value has changed .....	24
10. How can a status change in a G9003 or G9004 be identified? .....	25
10-1. Assign a device whose status you want to monitor .....	25
10-2. Precautions for monitoring port 0 (when using the change interrupt function) .....	26
11. How is the G9003 (PCL device) operated? .....	28
12. How is data exchanged with the G9004 (CPU emulation device)? .....	29
12-1. Precautions about the G9004 status .....	29
12-2. Message communication mode1 .....	29
12-3. Message communication mode2 .....	30
12-4. Message communication mode3 .....	32
12-5. Commands in the CPU emulation mode .....	34
12-6. CPU emulation mode 1 (Write sequential data) .....	35
12-7. CPU emulation mode 2 (Reading sequential data) .....	36
12-8. CPU emulation mode 3 (combined process) .....	38
12-9. CPU emulation mode: What is the communication standby setting? .....	40
12-10. CPU emulation mode: Data format that is returned by the G9004 when using a read command .....	41
12-11. CPU emulation mode: When the data returned by a read command exceeds 128 words .....	41
13. If I try to perform data communication (transient communication) with an I/O device, what will happen? .....	42
14. If a different PMD is placed between an I/O device and a center device, what will happen? .....	42
15. What happens if more data is sent to the PCL device than the receiving FIFO can handle? .....	43
16. What will happen if the G9004 does not read the data in the receiving FIFO? .....	43
17. What happens to each device when communication is not possible? .....	44
18. When a local device stops functioning, how can it be identified? .....	45

19. How can a new device be added during operation?.....	46
20. How can the communication status be monitored from the outside? .....	47
21. How can CPUs with no wait control terminal be used? .....	48
22. How to control the G9001 using only two address lines .....	49
23. What are the precautions for using a 40 MHz clock? .....	49
24. What is the selection standard for related parts? .....	50
24-1. Cable selection .....	50
24-2. Selecting an RS485 device.....	50
24-3. Selecting pulse transformers .....	50
24-4. Determining the proper termination resistance.....	50
25. When is a level shifter is needed? .....	51
Alphabetical index.....	53

# Basic version

---

# 1. What products are available in the G9000 series and what functions do they have?

The G9000 series consists of the following four device types (as of April 2004).

Model name	Product name	Classification
G9001	Center device	Center device (master device)
G9002	I/O device	Local device (slave device)
G9003	PCL device	Local device (slave device)
G9004	CPU emulation device	Local device (slave device)

## 1-1. G9001: Center device

It controls all aspects of serial communication. One center device is required to construct a Motionnet. (Only one center device can be used in a Motionnet.)

This center device controls communication with all the other devices connected to the Motionnet. (e.g. the G9002, G9003, and G9004. They are also referred to as local devices.)

The communication speed can be set to 20Mbps as the maximum speed, but it can also be set to 2.5, 5 or 10Mbits per second.

The center device is connected to a CPU.

## 1-2. G9002: I/O device

It has 32 input/output terminals.

These input/output terminals are arranged in groups of 8 bits per port (in other words, there are a total of 4 ports).

The four ports on the G9002 can be set as inputs or outputs using the external terminals PMD2, PMD1, and PMD0.

PMD2	PMD1	PMD0	Port 0	Port 1	Port 2	Port 3
L	L	L	Output	Output	Output	Output
L	L	H	Input	Output	Output	Output
L	H	L	Input	Input	Output	Output
L	H	H	Input	Input	Input	Output
H	L	L	Input	Input	Input	Input

A port that is set to output mode can be returned to its initial logic after a reset (output equals H or L) using the external terminals.

## 1-3. G9003: PCL device

The G9003 is used to control motion on a single axis.

Its functions are based on our PCL6045. It allows complicated motion control with simple user operations.

## 1-4. G9004: CPU emulation device

The G9004 has the following two selectable functions.

## 1) Message communication mode

By connecting a CPU to the G9004, it can exchange a maximum of 127 words of data (254 bytes) with the CPU that controls the center device. There is no restriction on the data format, therefore the G9004 offers highly flexible data exchange.



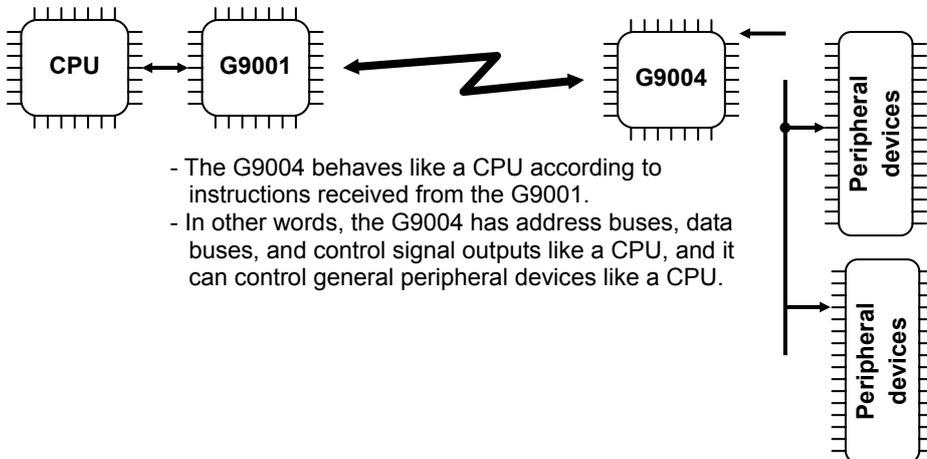
- A maximum of 127 words of data can be sent to the CPU that controls the G9004 from the CPU that controls the center device.
- On the other hand, the CPU used to control the G9004 can send a maximum of 128 words of data to the CPU that controls center device. However, the data transfer timing depends on the operation of the CPU of the center device.

## 2) CPU emulation mode

By connecting general CPU peripheral devices to the G9004, the CPU that controls the center device can operate these peripheral devices remotely.

For example, two PCL6045s (our own ASIC) can be connected to one G9004. (There are 6 address bus lines for use by peripheral devices on the G9004. Five address bus lines are needed to control the PCL6045.)

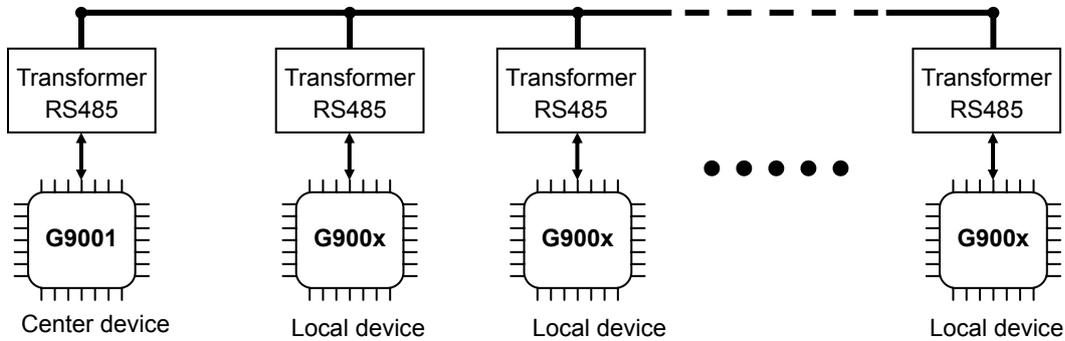
- The CPU that controls the G9001 can control the peripheral devices that are connected to the G9004 using serial communications. This allows the CPU to exchange data with the peripheral devices.



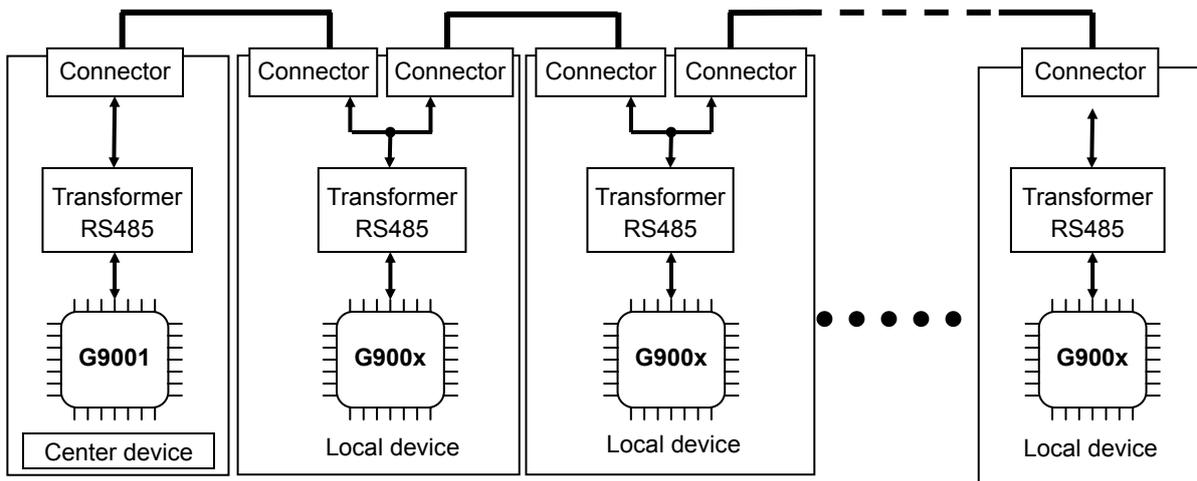
- The G9004 behaves like a CPU according to instructions received from the G9001.
- In other words, the G9004 has address buses, data buses, and control signal outputs like a CPU, and it can control general peripheral devices like a CPU.

## 2. How is a Motionnet constructed?

In a Motionnet, multiple local devices can be connected as a multi-drop system, like the one shown below, to a serial bus that is connected to a single center device.



Multi-drop refers to an arrangement in which multiple systems are connected to a single path. The simplest illustration of this system is shown in the figure above. However, the following chain arrangement is actually a common configuration.



## 3. How many local devices can be connected?

Up to 64 local devices can be controlled by one center device. The local devices must not have overlapping device addresses (the device address is set using the external terminals on each device). If addresses are duplicated, the center device cannot communicate normally. The center device itself does not have an address.

## 4. How is communication done?

The center device initiates all data communication. There are two primary communication methods: "cyclic communication" and "transient communication" (data communication). "System communication" is also possible, but this guide does not include a description of this communication type.

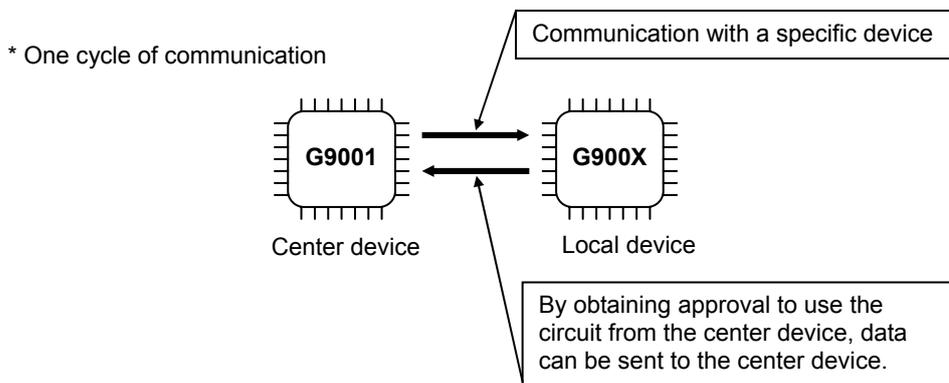
### 4-1. Cyclic communication

The basic Motionnet protocol resembles the "polling/selecting" method of communication.

In all communication circuit conditions, the center device controls the traffic. This is necessary in order to prevent data collisions on the circuit. Local devices cannot send data without approval by the center device.

In order to communicate effectively with each local device, the center device starts communications with the local device that has the lowest device address, and then communicates with all the rest in increasing device address order.

Only one device on the communication circuit can receive a given communication because duplicate device addresses are not allowed. The local device which receives the communication then has the right to use the communication circuit and can communicate with the center device. One cycle of communication is complete when all of the local devices have had a chance to communicate with the center device.



This simple query and response is carried out with all of the local devices, one by one, in the order of their device numbers.

When the center device receives a response from the local device with the highest device number, it repeats the communication cycle starting with the local device with the lowest device number.

The repeated communication method described above is referred to as "cyclic communication."

4-2. Transient communication = Data communication

**Note:** From this point on, this guide will refer to "transient communication" as "data communication."

The maximum amount of data that can be exchanged using the cyclic communication is 4 bytes. (The reason for this will be covered later.)

This is too small an amount for operating instructions to control a PCL device or for controlling CPU emulation devices. With cyclic communication, there is a restriction on the time because a specific device cannot make another communication until the communication with all of the other devices is complete.

These problems can be solved by using data communication.

Data communication can interrupt cyclic communication using any desired timing (the timing is controlled by a CPU).

For example, suppose that a cyclic communication is being carried out in the following order.



"Address n" is for a local device with address "n." The arrows in the figure show the order of the cyclic communication.

If a data communication command is issued at point "A" in the timing, the communication order will be as follows.



When interrupts occur, the total time for one cycle of cyclic communication will vary. However, data exchange with an arbitrary device can be performed immediately.

Although we said that the interrupt can be performed at any time, there are actually some restrictions.

To perform continuous data communication, or to retry a data communication after a communication error, one cycle of cyclic communication must be performed.



Data communication (transient communication) cannot be done continuously.

This is to prevent long pauses in cyclic communication caused by uninterrupted data communications.

A maximum of 128 words (256 bytes) can be sent in one data communication transfer.

Just like with cyclic communication, in data communication the center device sends data to a specific local device and the local device sends a response to the center device. A maximum of 256 bytes of data can be included in this response (currently, this is possible only with G9004).

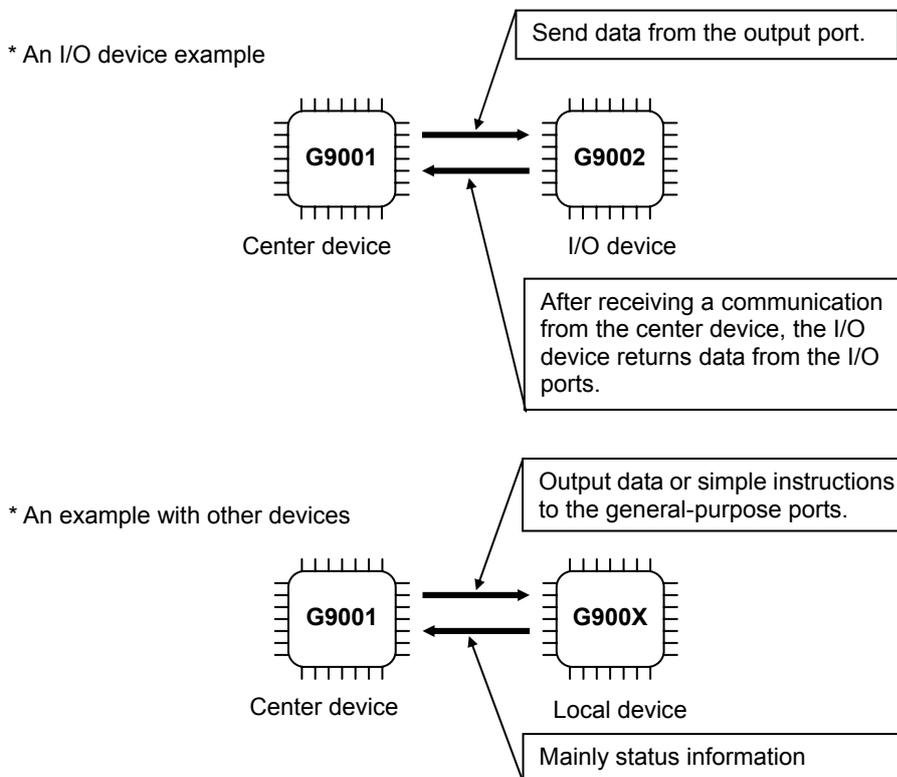
By the way, the I/O device does not respond to data communication (since it does not need this function).

## 5. What kind of data can be sent using cyclic communication?

Cyclic communication is carried out continuously unless there is a stop command from the CPU. There are upper limits on the amount of data using this communication method. This rule keeps cyclic communications within a certain amount of time. (Due to this restriction, data communication must be used to send large amounts of data.)

A maximum of 4 bytes can be sent in cyclic communication. The data that can be communicated within this number of bytes varies with the local device.

Device name	Communication details from the center device to local devices	Response details from the local device to center devices
G9002	Data output from the output ports	Data input on the input ports.
G9003	Output data from general-purpose output ports	Input data from general-purpose input ports and status information from the PCL device
G9004	Operating instructions, etc.	Status information for the CPU emulation device



Cyclic communication was designed for effective data sending and receiving with a number of I/O devices (insuring that the data is always refreshed within a certain interval). Therefore, a maximum of 4 bytes can be sent.

## 6. What is status?

In order to monitor the conditions of the G9003 (PCL device) and G9004 (CPU emulation device), each of these devices contains status information. (The G9002 I/O device does not produce status information.) For example, this function can be used to learn whether a PCL device has started or stopped and to handle data transfer requests from a CPU emulation device.

The status information is sent to the G9001 (center device) from each device using cyclic communication. The information is stored in the RAM in the center device.

By referring to this information from the CPU, the status of each device can be monitored.

Also, the G9001 itself maintains status information about itself and it can be read from the CPU. The G9004 has two types of status information: the status information sent to the center device and to be referred by the CPU of the center device, and status referred by a CPU that is connected to G9004. (For details, see "Precautions for G9004 status" section.)

## 7. How fast is a communication cycle?

In cyclic communication, a communication cycle is as follows when a 20 Mbps speed is selected.

Number of local devices	Communication cycle	Remarks
8	0.12 ms	If all of the local devices connected are I/O devices, 256 input/output points can be used.
16	0.24 ms	If all of the local devices connected are I/O devices, 512 input/output points can be used.
32	0.49 ms	If all of the local devices connected are I/O devices, 1024 input/output points can be used.
65	0.97 ms	If all of the local devices connected are I/O devices, 2048 input/output points can be used.

If a different number of local devices is connected, or when the communication cycle is interrupted by data communications, refer to the calculation formulas in the user's manual to calculate the time.

## 8. What is the longest communication distance possible using Motionnet?

Our experimental communication results are shown below. These experiments were done in our plant and the environmental conditions are shown.

These are only for reference when you construct your system. Systems with configurations different from the one below are also possible. Please use the information below for reference only.

Conditions						Results
Transfer rate	Number of local devices	Cable type	Termination resistance	Pulse transformer	I/F chip	Max. length
20 M-bps	32	CAT5	100 ohms	1000 $\mu$ H	RS485	100 m
20 M-bps	64	CAT5	100 ohms	1000 $\mu$ H	RS485	50 m
10 M-bps	64	CAT6	100 ohms	1000 $\mu$ H	RS485	100 m

Note: In the data above, the maximum length given is for the results in ideal conditions in an experimental room. Please note that even when using an RS485 and the cable is laid in the same conditions, the same results may not be obtained in actual use.

The G9001 (center device) has two sets of serial data terminals. If the desired communication distance cannot be obtained using one terminal, try the other terminal.

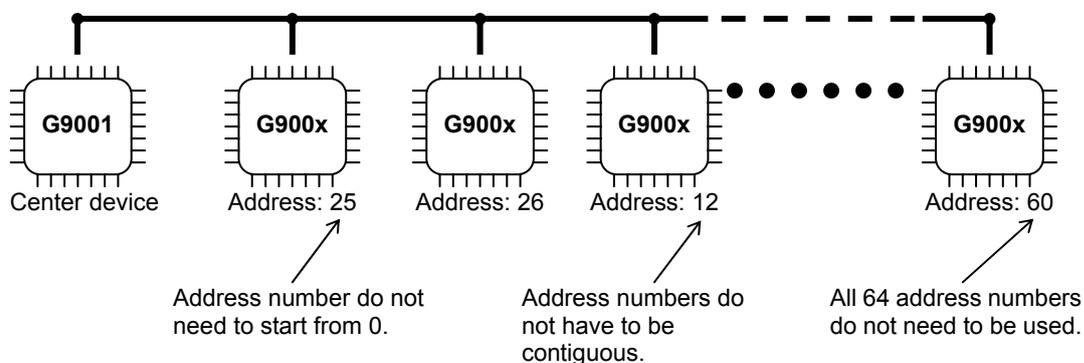
## 9. Do device addresses have to be allocated sequentially without gaps?

You can use 64 device addresses from 0 to 63. However, there is no requirement to start from address 0. (Do not use the same address twice.)

There will not be a problem if some address numbers are not used.

Normally, the center device determines the addresses of the local device connected to the communication line, and only communicates with the addresses of connected devices.

These addresses can be determined automatically. (The user can also manually determine the addresses of the connected devices.)



## 10. What is RS485?

RS485 is an agreed international standard for serial communication. It is also referred to as "EIA-485." Motionnet recommends that systems be built using this standard because this standard is superior to others at present as a standard for high speed, long distance serial communication, and driver chips conforming the RS485 standard can easily be obtained. "Driver chip" refers to an IC with a transceiver function that has both serial signal sending and receiving functions.

The RS485 standard only specifies the electrical characteristics needed for communication. However, RS232C also specifies signal status that transfer on a circuit. This is a distinctive difference between RS485 and RS232C and other standards.

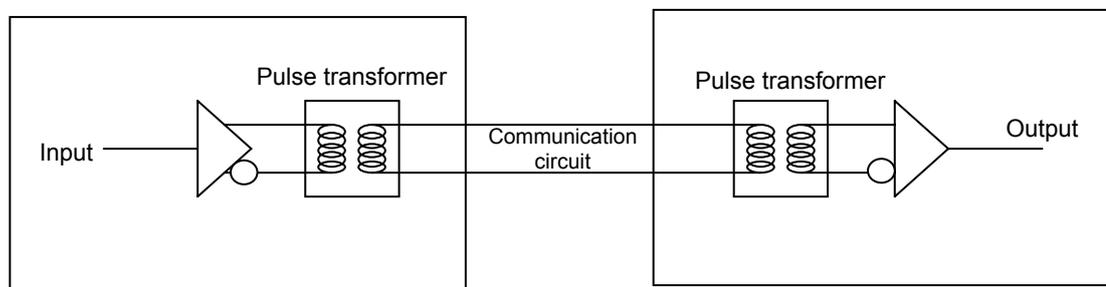
## 11. Why are pulse transformers needed?

The term "pulse transformer" is used for transformers designed for communication applications. However, their construction is the same as a normal transformer.

By taking advantage of the characteristic that transformers can only pass AC current signals, it isolates the power source of each communication system. Motionnet recommends constructing a communication system using pulse transformers.

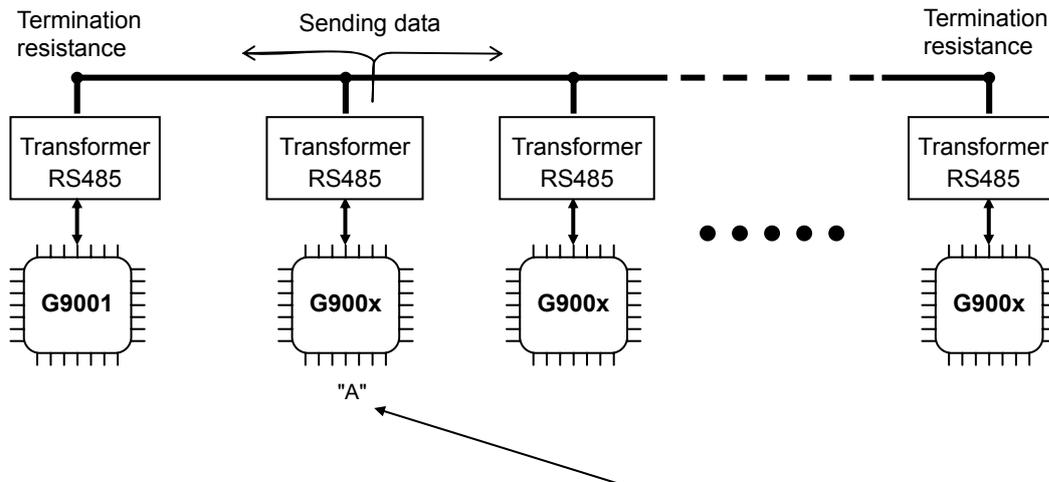
The use of bare RS485 devices and communicating without pulse transformers is possible. However, if you do this, the ground must be common between the data sending and receiving sides. If the ground levels are not identical, the voltage level of the signals transferred through RS485 devices may exceed allowable voltage range of the RS485 devices and they may be destroyed. In order to make the ground common, you must add one parallel ground line to the communication circuit. (A total of three lines are needed for this communication circuit.)

This means that, technically, communication circuits that don't use pulse transformers can be constructed. However, since an extra ground line must be used, a large ground loop may be created. This may create a system that cannot cope with electrical noise.



The communication circuits are isolated by the pulse transformers.  
Therefore, the two systems in the figure above are electrically isolated.

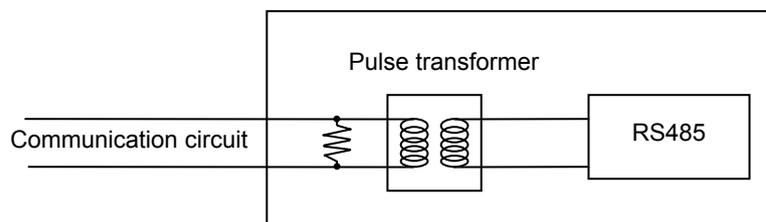
## 12. What is the role of the termination resistance?



For example, suppose that serial data is sent from device "A."

Signals are transferred to both sides of the communication circuit and reach the ends of the communication circuit. The data rebounds at the ends and reverses the flow through the communication circuit. This is referred to as "reflection" and is a major source of communication problems. Termination resistance is used to prevent this reflection. Normally, resistors of 100 ohms or so are placed at both ends of communication circuit. They do not need to be installed elsewhere in the communication circuit.

The figure below shows a terminating resistor inserted in the communication circuit after the pulse transformer. If one is installed on the other side of the circuit (the RS485 device side of the pulse transformer), you may get the same effect.



### 13. Types of CPUs that can be used to control the center device

The G9000 series devices that need a CPU are the G9001 and G9004 (when used in the message communication mode). The G9002, G9003, and G9004 (when used in the emulation mode) are controlled by the G9001 (center device).

The type of CPU that can be connected varies a little between the G9001 and G9004.

Shown below are the CPUs that can be connected to each device. (Basically, most CPUs can be connected.) To select the CPU type, set the external terminals "IF0" and "IF1."

- CPUs that can be connected to the G9001

Setting		I/F mode	CPU type	Example of the CPU type
IF1	IF0			
L	L	I/F mode 1	Motorola family 16-bit CPU Addressing must be big endian.	68000
L	H	I/F mode 2	Other 16-bit CPU Addressing must be big endian.	H8
H	L	I/F mode 3	Intel family 16-bit CPU Addressing must be big endian.	8086
H	H	I/F mode 4	8-bit CPU	Z80

- CPUs that can be connected to the G9004

Setting		I/F mode	CPU type	Example of the CPU type
IF1	IF0			
L	L	I/F mode 1	Motorola family 16-bit CPU	68000
L	H	I/F mode 2	Other 16-bit CPU	H8, 8086
H	L	I/F mode 3	8-bit CPU (Intended as an LC controller for IC control)	6809
H	H	I/F mode 4	8-bit CPU	Z80

All of the G9000 series devices run on a single 3.3V power supply. However, 5V can be applied to most of the terminals (there are some exceptions for communication lines on the I/O devices) so that they can be connected directly to 5V type CPUs.

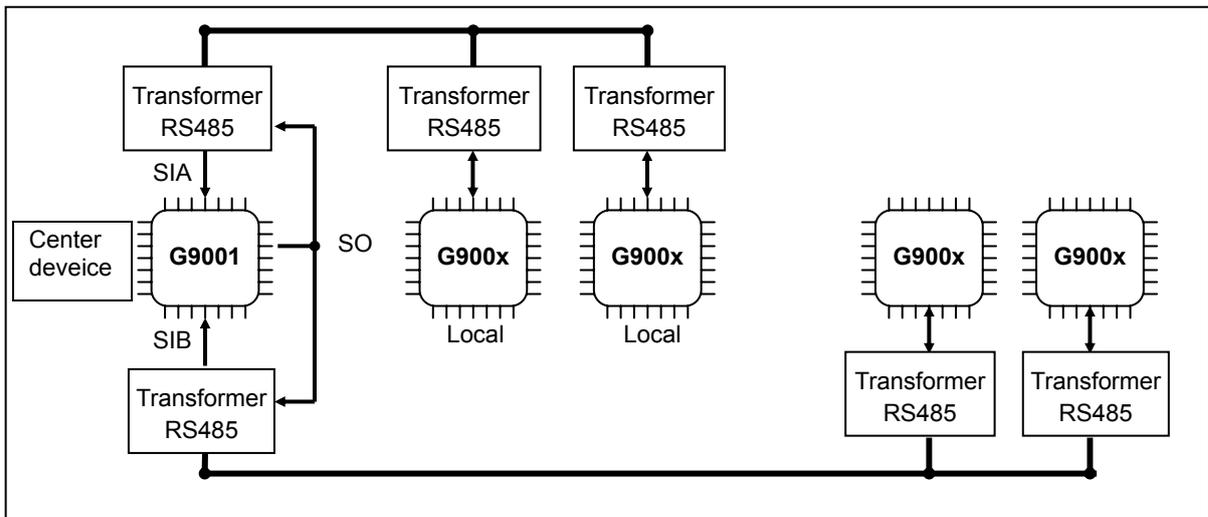
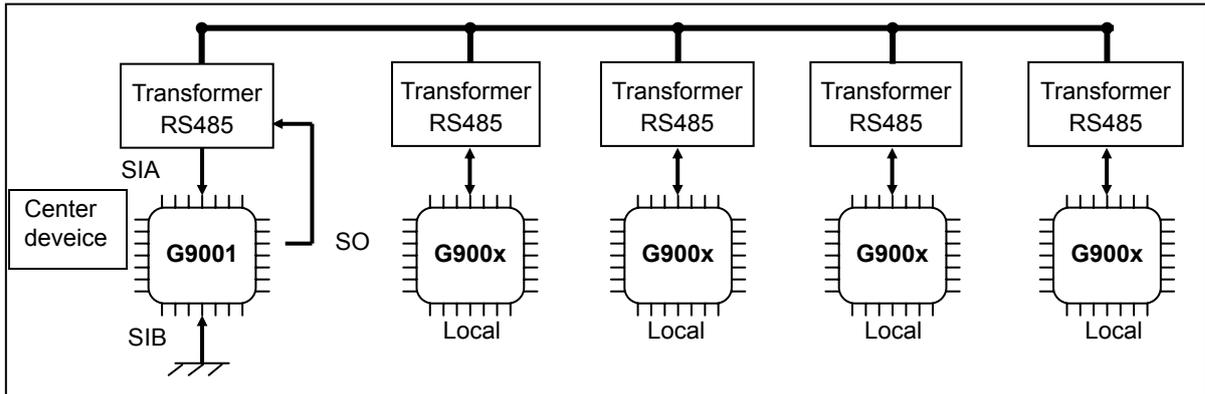
## 14. Why does the center device have two serial inputs (SIA, SIB)?

One serial input for communication is used to receive data from a communication line that is handled by a device such as an RS485.

The other terminal is used to output data to a communication circuit that is "SO" only. The number of lines is one.

Normally, you only use one serial input (SIA as main line) and pull up or down the other serial input terminal so that you can communicate.

The reason there are two input lines is shown in the figure below.



The CPU operating methods with the two different configurations described above are the same because the output data line (SO) is common. But, by separating a circuit, the load on one line can be decreased and a long distance cable configuration can easily be established. For example, when connecting 32 local devices, using two 50m lines is better than using only one 100m line, and you have better noise immunity. To get better communication quality, two lines are preferable.



# Operation version

---

## 1. How easy is it to use Motionnet?

To communicate using a properly connected Motionnet, the items that the CPU needs to tell the G9001 (center device) are very simple.

The G9001 will start cyclic communication with just the following two instructions.

### 1) Issue a "System communication" command

By issuing this command to the G9001 (center device), the G9001 will check what devices are connected to the communication circuit.

The items it checks are as follows: the device type (G9002 or other devices); if appropriate, the setting conditions of the I/O ports; the device address of each device.

With this command the G9001 can automatically obtain and set the device information on the circuit.

The results can be referred to and changed by the CPU.

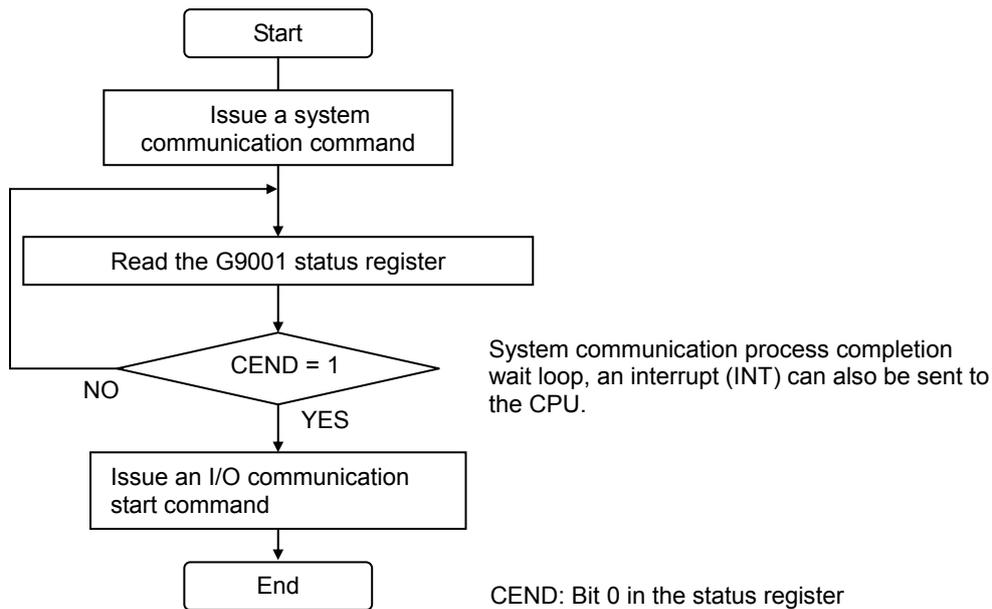
When the checks are complete, the "CEND" (bit 0) in the G9001 status register changes. After seeing this bit change, the next instruction can be issued.

### 2) Issue an "I/O communication start" command

The G9001 will start cyclic communication with the devices whose addresses were obtained in step 1) above.

That's all there is to it. For example, an I/O device that is set up as an input port sends input data to the internal RAM in the G9001 (center device). On the other hand, by simply writing data into the internal RAM, an I/O device that is set up as an output port will receive the data.

The addresses in the internal RAM are allocated values corresponding to the device numbers. The details of this function are described in the following section. For even more detailed information, see the user's manual. (The address map is shown in the manual.)

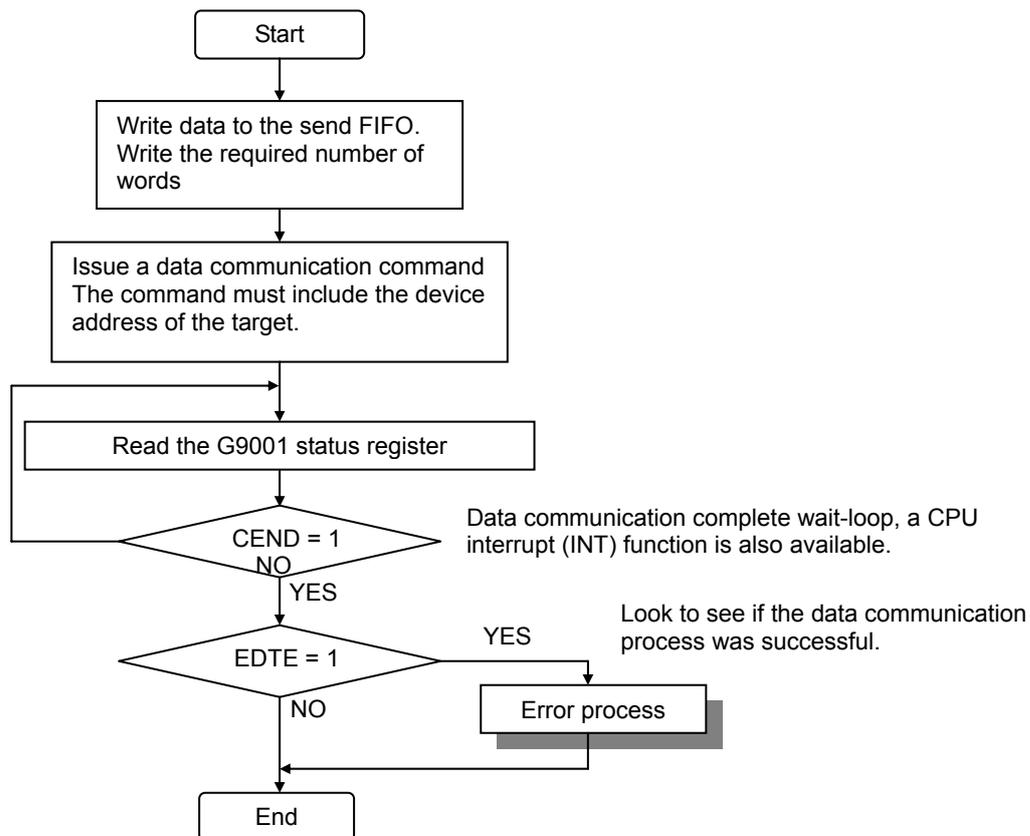


## 2. How is data communication (transient communication) performed?

In order to start transferring data you must start cyclic communication. Once cyclic communication has been started, data communication can be started any time.

The procedure for data communication is as follows.

- 1) Place the data to send in the G9001 FIFO (Center device).  
The G9001 has 2 FIFO buffers (256 bytes each) for sending and receiving.  
The data to be sent using data communication is written to the send FIFO in units of one word (8 bits or 16 bits).
- 2) Issue a data communication command using a specific device address.  
The data communication command can contain any device address. By issuing this command to the G9001 (Center device), the data previously placed in the send FIFO will be sent to the target device.  
Then you can check that the data was sent without problems by watching the G9001 status.



CEND: Bit 0 of the status register

EDTE: Bit 4 of the status register

### 3. How is data written to the send FIFO buffer?

The FIFO is a RAM buffer that does not have specific address. An entry point is provided for input and an exit point for output. Data written to the entry point are stored inside the RAM in the order they are written. The stored data is then output from the exit point. Data can only be removed in the order it was written. Since this RAM outputs the data that was input first, it is referred to a FIFO (First In First Out) buffer. The entry point for the send FIFO buffer is as follows on the G9001 address map.

Address	Details (16 bits)
000h 002h 004h	Used for other purposes
006h	Data write port on the send FIFO buffer.
008h through 1FEh	Used for other purposes

Note: The address map is as described above when I/O mode 3 (IF1 = H, IF0 = L) is selected.

When a 16-bit CPU writes data to address "006h," the data is stored in the send FIFO buffer.

### 4. How is data read from the receiving FIFO buffer?

The data exit point on the receiving FIFO buffer is located as follows on the G9001 address map. As you can see from the address map, it is the same address as is used by the send FIFO buffer. So, when you want to write data, it is written to the send FIFO buffer. When you want to read data, it is read from the receiving FIFO buffer.

Address	Details (16 bits)
000h 002h 004h	Used for other purposes
006h	Data write port on the receiving FIFO buffer.
008h through 1FEh	Used for other purposes

Note: The address map is as described above when I/O mode 3 (IF1 = H, IF0 = L) is selected.

When a 16-bit CPU reads data from address "006h," the CPU is actually removing the data from the receiving FIFO buffer.

### 5. How is data about the G9002 (I/O device) input port read?

Once cyclic communication has started, the input port status of the G9002 is read during each cycle automatically. These values are stored in the G9001 RAM (center device).

Address	Details (16 bits)
000h	Used for other purposes
through	
0FEh	
100h	Port 1 and port 0 of device address 00.
102h	Port 3 and port 2 of device address 00.
104h	Port 1 and port 0 of device address 01.
106h	Port 3 and port 2 of device address 01.
through	through
1F8h	Port 1 and port 0 of device address 62.
1FAh	Port 3 and port 2 of device address 62.
1FCh	Port 1 and port 0 of device address 63.
1FEh	Port 3 and port 2 of device address 63.

} 16bit x 128word

Note: The address map is as described above when I/O mode 3 (IF1 = H, IF0 = L) is selected.

The G9002 has four, 8-bit I/O ports (port 0 to 3). These ports are located in the G9001 (center device) RAM as shown in the map above.

Therefore, in order to obtain the status of any G9002 port, you may read the data from the RAM which corresponds to the device address in the G9002.

For example, to obtain the status of port 1 for device address 62, read address "1F8h" in the G9001 RAM (center device) and then refer to the upper 8 bits of this data.

However, each of the four ports on the G9002 can be set as an input or output with external terminals (PMD2, PMD1, and PMD0).

PMD2	PMD1	PMD0	Port 0	Port 1	Port 2	Port 3
L	L	L	Output	Output	Output	Output
L	L	H	Input	Output	Output	Output
L	H	L	Input	Input	Output	Output
L	H	H	Input	Input	Input	Output
H	L	L	Input	Input	Input	Input

Ports set as outputs write data to the RAM as shown in the next section. By reading these data, data placed in the output buffer are read out. (The default port value is undefined.)

### 6. How is data output from the G9002 output port (I/O device)?

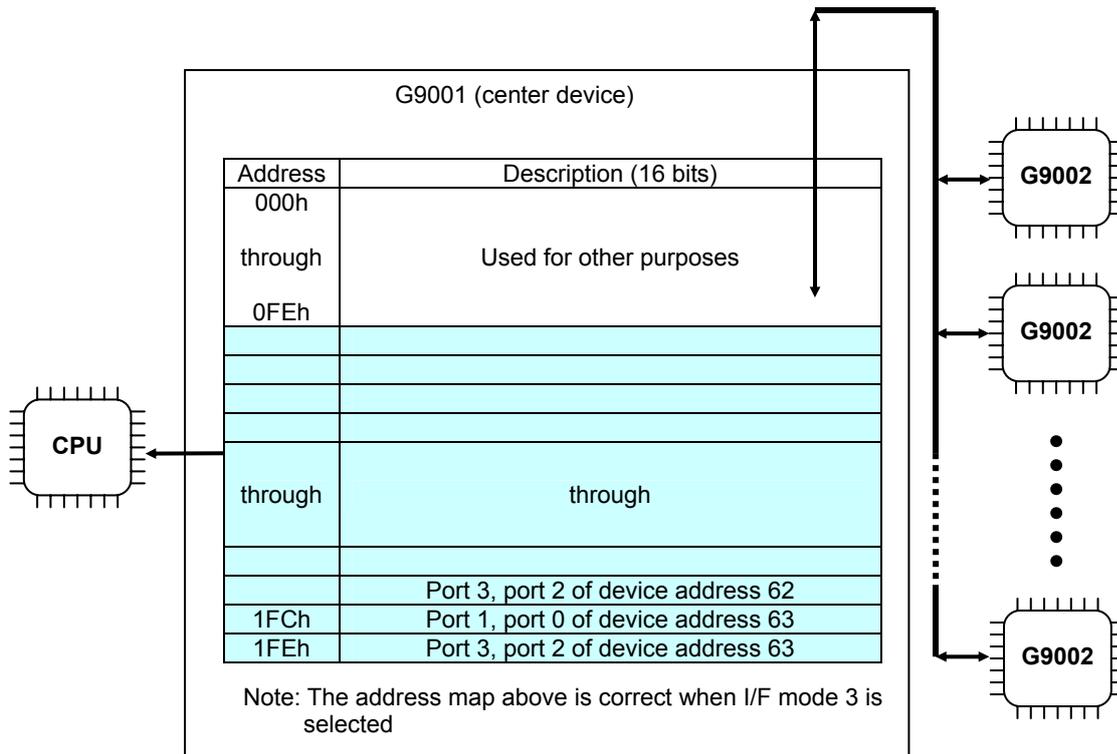
When cyclic communication starts each cycle, the G9001 (center device) automatically sends the data stored in its RAM to the G9002 output port.

The address map for this operation is the same as described in the previous section. Write the data to send to the RAM address that corresponds to the desired device address.

However, any port that is put into input mode on the G9002 will ignore any data written to its address in the RAM. (The input data from the G9002 are always reflected.)

Summing up the above: port information for a local device that is connected to the G9001 (center device) is refreshed cyclically through the internal RAM.

Looking at this operation from the CPU side, it handles the data the same as when accessing the RAM.



## 7. How can you check the status information for the G9003 and G9004?

The G9001 (center device) provides its own status information to the CPU.

Like the G9001, the G9003 (PCL device) and G9004 (CPU emulation device) maintain their own status information.

The status information about these local devices is transferred to the G9001 RAM during cyclic communication. This is the same system used to exchange port information about the G9002 (I/O device) through the RAM. Devices that do not have I/O ports send their status information to the G9001 during cyclic communication. (The G9003 has an 8-bit general-purpose port and communicates status information during cyclic communication.)

In other words, cyclic communication is used to exchange I/O data with the I/O devices and to exchange status information about other devices. Therefore, the location of the I/O information and status information is in the same area in the RAM.

The table described in the section "How to obtain data about the G9002 I/O ports (I/O device)" is shown below.

A local I/O device with any device address will have information stored in the corresponding RAM address. If a local device is not an I/O device, status information will be stored at that location.

Address	Description (16 bit)
000h	Used for other purposes
through	
0FEh	
100h	I/O or status info for device address 00
102h	I/O or status info for device address 00
104h	I/O or status info for device address 01
106h	I/O or status info for device address 01
through	through
1F8h	I/O or status info for device address 62
1FAh	I/O or status info for device address 62
1FCh	I/O or status info for device address 63
1FEh	I/O or status info for device address 63

} 16bit x 128word

Note: The address map above is correct when I/O mode 3 (IF1=H, IF0=L) is selected.

### 8. How to check the device type for devices that are connected to a communication circuit

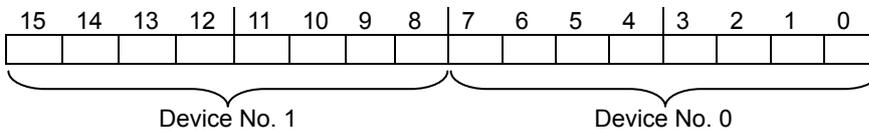
Local devices connected to the communication circuit can consist of any combination of G9002, G9003, and G9004s.

Information about local devices that are connected to the communication circuit is collected automatically by the "system communication command." The collected data is stored in the G9001 internal RAM as follows. (This is referred to as the device information area.) This area can be read or changed by the CPU.

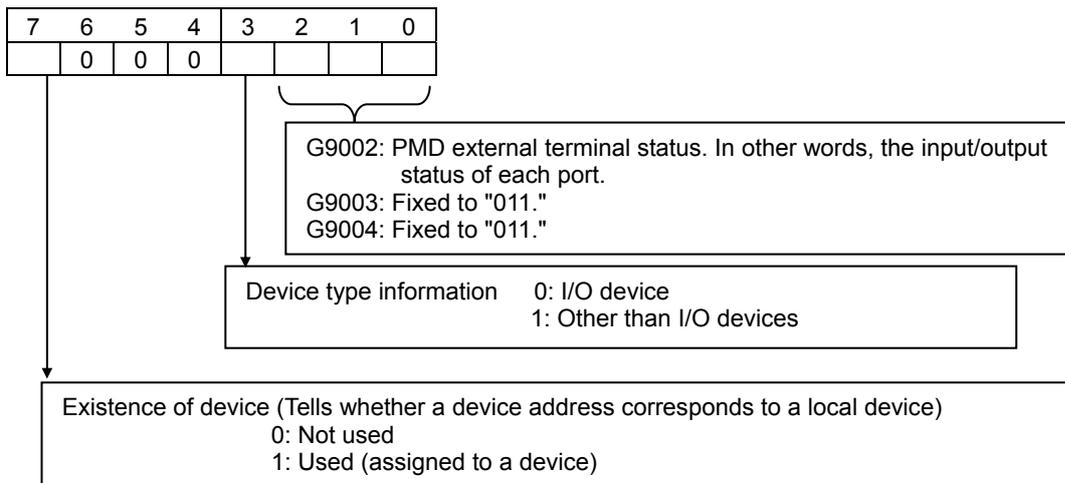
Address	Description (16 bit)
000h through 076h	Used for other purposes
078h	Device information about device address 01, 00.
07Ah	Device information about device address 03, 02.
through	through
0BCh	Device information about device address 61, 60.
0BEh	Device information about device address 63, 62.
0B8h through 1FEh	Used for other purposes

Note: The address map above is correct when I/F mode 3 (IF1=H, IF0=L) is selected.

The details of address "078h" in the address map are shown below.



When a 16-bit CPU looks at address "078h," it will look like the figure above. Each 8 bits corresponds to one device address. Each device address is as follows:



## 9. How can an input port change on an I/O device be identified using the interrupt function?

The I/O port status of the G9002 (I/O device) is always refreshed with the latest data during cyclic communication. However, it does not necessarily change all the time. In order to monitor ports which might change at any time, there is a function that can notify the CPU of a change in the input status. It uses an input change interrupt.

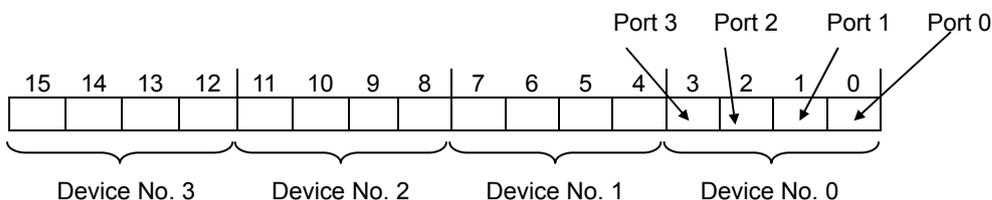
### 9-1. Assign a device to be monitored for a change in its input value

You have to assign which port on which I/O device will be monitored for changes in its input. To assign a device, use the "input change interrupt setting area" inside the G9001 RAM (center device).

Address	Description (16 bit)
000h through 0BEh	Used for other purposes
0C0h	Put the device addresses to be monitored for changes in locations 03, 02, 01, and 00.
0C2h	Put the device addresses to be monitored for changes in locations 07, 06, 05, and 04.
through	through
0DCh	Put the device addresses to be monitored for changes in locations 59, 58, 57, and 56.
0DEh	Put the device addresses to be monitored for changes in locations 63, 62, 61, and 60.
0E0h through 1FEh	Used for other purposes

Note: The address map above is correct when I/F mode 3 (IF1=H, IF0=L) is selected.

The details for address "000h" are shown below.



When a 16-bit CPU looks at address "000h," it will look like the description above. Each 4 bits correspond to one device address.

The 4 bits correspond to port 0, port 1, port 2, and port 3 respectively.

With this configuration, when a bit corresponding to a port number at the device address you are monitoring changes to "1," the CPU will start monitoring the input value on the port. For example, if you want to monitor port 2 at device address 5, do the following.

G9001: RAM address = 0C2h  
 Set it to = 0040h

If you want to monitor a port's status, assign the port on the device address you want to monitor as an input.

By writing the information above, the G9001 will send an interrupt (INT) to the CPU if there is a change in the port being monitored.

9-2. Identify a device whose input value has changed

In the section above, we described how the G9001 monitors changes in an input signal and outputs an interrupt signal to the CPU.

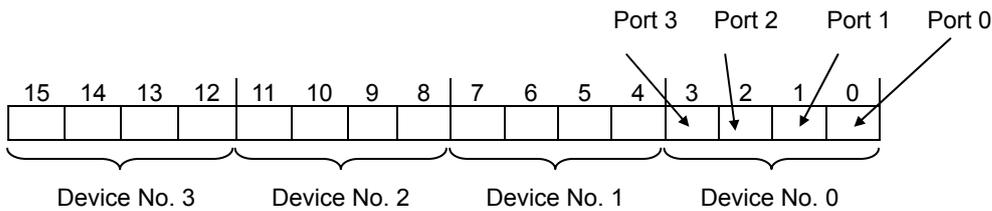
However, the G9001 has to know which device and which port input signal actually changed (especially when monitoring a large number of ports).

In this case, the G9001 can check the location that changed by analyzing the "input change interrupt flag area" in the following RAM locations.

Address	Description (16 bit)
000h through 0DEh	Used for other purposes
0E0h	Input change interrupt flag for device addresses 03, 02, 01, and 00.
0E2h	Input change interrupt flag for device addresses 07, 06, 05, and 04.
through	through
0FCh	Input change interrupt flag for device addresses 59, 58, 57, and 56.
0FEh	Input change interrupt flag for device addresses 63, 62, 61, and 60.
100h through 1FEh	Used for other purposes

Note: The address map above is correct when I/F mode 3 (IF1=H, IF0=L) is selected.

Shown below are the details for address "0E0h" in the address map. This is the same configuration as the settings for monitoring changes in the input.



When a 16-bit CPU looks at address "0E0h", it will look like the figure above.

Each 4 bits correspond to one device address.

The 4 bits correspond to port 0, port 1, port 2, and port 3, starting from the low bit.

If any of these bits changes to a "1," it means that the input value changed on the corresponding port number of that device address.

For example, suppose that the following address has the value given below, it can be determined that the input status of port 3 and port 0 at device address 58 have changed.

G9001: RAM address = 0FCh (portion of device addresses 59, 58, 57, and 56)  
 Data read = 0900h

When bit changes to "1," it must be changed back to "0" by the program (If it is not returned to 0, no further interrupt will occur. However, the G9001 will not be able to identify the device.)

To return this bit to "0," write "1" to this bit. If this is too much trouble, write back the data you just read so that the bit will be returned to "0."

## 10. How can a status change in a G9003 or G9004 be identified?

Basically, the input change interrupt function on the I/O port is also used to identify a status change in a G9003 or G9004. However, there are some precautions. You will need to understand them well for things to work as you expect.

### 10-1. Assign a device whose status you want to monitor

The setting method is the same as when "Assigning a device to be monitored for a change in its input value" on a G9002 (I/O device).

As described before, local devices other than the G9002 send their own status information to the center device using cyclic communication. It has also been noted that these data are stored in the same area as the I/O information.

So, what kind of data is assigned to the area which corresponds to each port on the G9002 (I/O device)? These details are shown below.

#### G9003 (PCL device)

Port 0	Port 1	Port 2	Port 3
Main status (lower bits)	Main status (upper bits)	General-purpose I/O or input data	General-purpose I/O or output data

#### G9004 (CPU emulation device)

Port 0	Port 1	Port 2	Port 3
Status 0	Status 1	Status 2	Operating instructions

The shaded cells are the information sent from a local device. The other cells are data that the center device sends to local devices. In other words, ports 0 to 2 are used by the local device to send status information, etc. This information is stored in RAM on the center device.

The relationship between a device address and the RAM address is described in the "How to obtain data from the G9002 (I/O device) input port"

In other words, assume that cells from port 0 to port 2 are input data, and that the system is set to monitor these cells (the same as in the section "Assigning a device to be monitored for a change in its input value").

10-2. Precautions for monitoring port 0 (when using the change interrupt function)

The method for monitoring the status of the area corresponding to port 0 on the G9003 (PCL device) or G9004 (CPU emulation device) (G9003: Lower 8 bits of the main status, G9004: Status 0) is a little different from other cases. Other ports trigger an interrupt when any of the 8 bits of data change. However, in the area corresponding to port 0, only bit 0 is monitored. (Furthermore, the center device only monitors a status change from "0" to "1.") (When an I/O device is connected, the center device will monitor all of the bits.)

As you will see when you read the user's manual for each device, bit 0, which corresponds to port 0, is a logical "AND" of all the other bits (in order to indicate the cause of an interrupt in each local device). In other words, monitoring bit 0 is effectively monitoring the interrupt request from each local device.

So, what a user needs to be careful about is that, once a status change has occurred and caused an interrupt, further status changes which cause interrupts will not be seen unless the cause of the interrupt from the first local device is resolved. This is shown in the figures below.

Let us describe the case when using a G9003 (PCL device).

- 1) Before a PCL device interrupt occurs, the input change flags from the lower bits of the PCL device's main status and the corresponding port 0 are as follows.

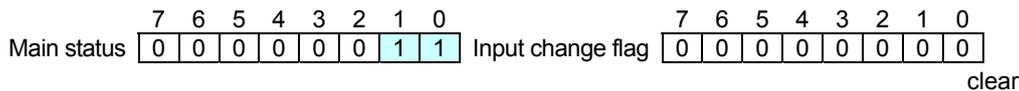


- 2) Imagine that the PCL device main status changes because the PCL device stops operating.



Bit 1 in the PCL device main status is used to cause an interrupt if the operation stops, so this bit becomes "1." In addition, bit 0 is a logical "AND" of bits 1 to 3, so that this bit also becomes "1." This status is then stored in the port 0 area (in the G9001 RAM) due to normal cyclic communication. At this stage, an interrupt will occur and the input change flag will change to "1."

- 3) The CPU clears the input change flag inside the G9001 RAM. This can be done by writing back the input change flag after reading the data related to the input change.



- 4) Next, suppose that an error interrupt occurs.

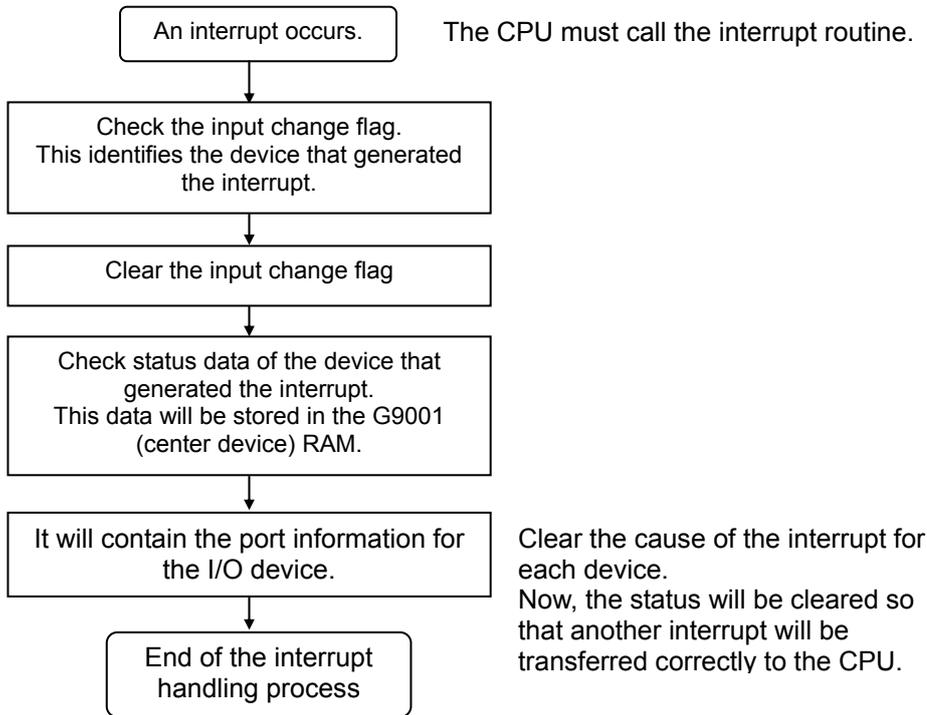


Bit 2 of the G9003 main status changes to "1." Bit 0 should also change to "1." However, since it is already a "1," it looks like there was no change (port 0 only monitors bit 0). In other words, the second interrupt that occurred is not transferred to the CPU.

At this point the important operation had not yet been completed. Many users believe that the cause of the interrupt can only be resolved by clearing the input change flag using the CPU.

However, the interrupt cause must actually be resolved on each device. In this example, we have to clear the "1" that appears in the G9003 (PCL device) main status. (Clearing the input change flag only clears the information used to identify the device that had the interrupt. It does not clear the interrupt cause itself.)

So, the procedures we actually have to perform to check the cause are shown below.



Next, the operations used to clear the cause of an interrupt are shown for each device.

Local device	Bit position	Interrupt cause	Methods for clearing the flag
G9003 PCL device	1	Operation stopped interrupt	Send an interrupt reset command (0008h)
	2	Error interrupt	Read the RESET register of G9003
	3	Event interrupt	Read the RIST register of G9003
G9004 CPU Emulation device	1	Local side: Data receive process complete interrupt	Send a receive process complete interrupt reset command (0400h)
	2	Local side: LIRQ input interrupt	Send an LIRQ input interrupt reset command (0500h) Process the peripheral chips which the G9004 is controlling.
	3	Local side: Data send request interrupt	Send a message communication command (0001h)

All of the send commands and register reads are performed using data communication. For details, see the user's manual for each device.

## 11. How is the G9003 (PCL device) operated?

The functions of the G9003 are the same as for our PCL series ASICs. It is a chip that can output pulse trains to control motors.

The basic command system and operating methods resemble a conventional PCL. The difference from conventional PCLs is that the G9003 does not require a CPU to control devices. Instead it operates by receiving commands from the G9001 (center device).

What commands are used and how does the G9001 send them to the G9003?

As described before, the commands sent to the G9003 from the G9001 resemble those from a conventional PCL. The G9001 sends these commands to the G9003 using data communication (transient communication).

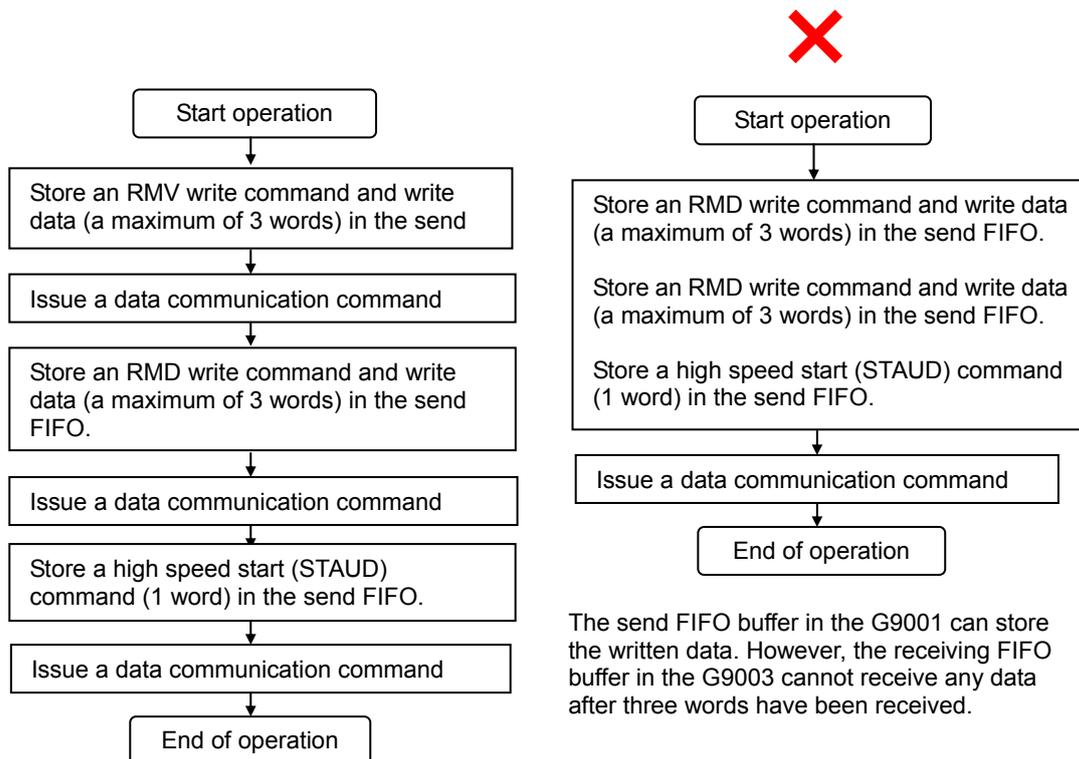
The G9003 can accept both one word (16 bits) and three word type commands.

The G9001 stores these commands in the data send FIFO buffer and then issues a send command. See the section "How is data communication (transient communication) performed?" for more details. For further details, also see the user's manual for the G9001 or G9003.

Now, pay special attention because the commands that can be stored in the send FIFO buffer are only one possible source of instruction to the G9003. If you want to put data into more than one register at once, and if you store this data in the FIFO buffer all at once and send it, it may cause an error in the G9003.

This is because the G9003 only has 8 bytes in its FIFO buffer for a send or receive sequence, but the G9001 has a send FIFO buffer of 256 bytes.

For example, if you put data in the RMV and RMD registers on the G9003 and try to trigger a high speed start (STAUD), the communication will not be carried out correctly using the procedures on the right below.



The send FIFO buffer in the G9001 can store the written data. However, the receiving FIFO buffer in the G9003 cannot receive any data after three words have been received.

Before issuing a next data send command, the CPU has to check if the previous data communication is complete.

## 12. How is data exchanged with the G9004 (CPU emulation device)?

The basic operation methods are the same as in the section "How is data communication (transient communication) performed?"

The procedures shown below are based on the assumption that input change interrupts and the other functions are set properly.

### 12-1. Precautions about the G9004 status

In the G9004 manual, two kinds of status information are described. In order to prevent confusion, let us make the differences between the two clear.

**The first status information** is sent to the G9001 (center device) using cyclic communication. See the section "Setting the status and operation information for the G9004" in the G9004 user's manual.

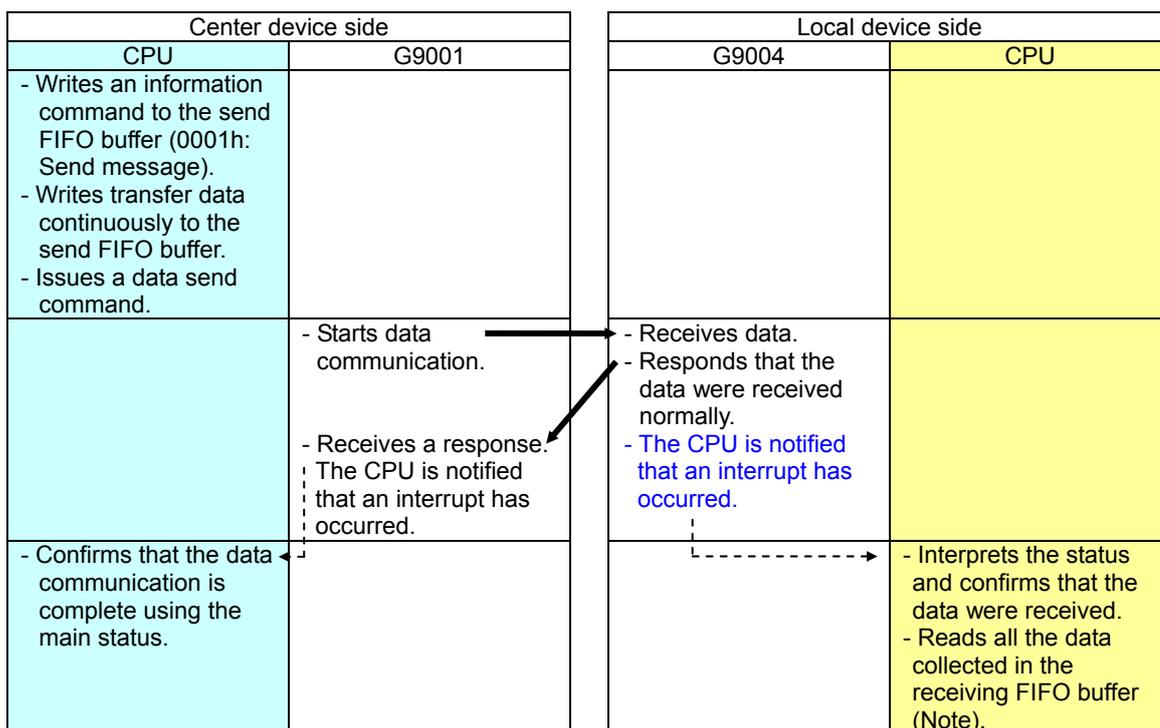
A CPU connected to the G9001 checks this status information.

**The second status information** occurs when using the G9004 in the message communication mode.

See "Command and status information that can be controlled by a local CPU", in the G9004 user's manual.

A CPU connected to the G9004 checks this status information.

### 12-2. Message communication mode1 (Transfer data from a CPU connected to the center device to a CPU connected to the G9004)



Note: When reading data from the receiving FIFO buffer, repeat the read process until the following flags change to "0", which will mean that you have read all the data in the FIFO buffer.

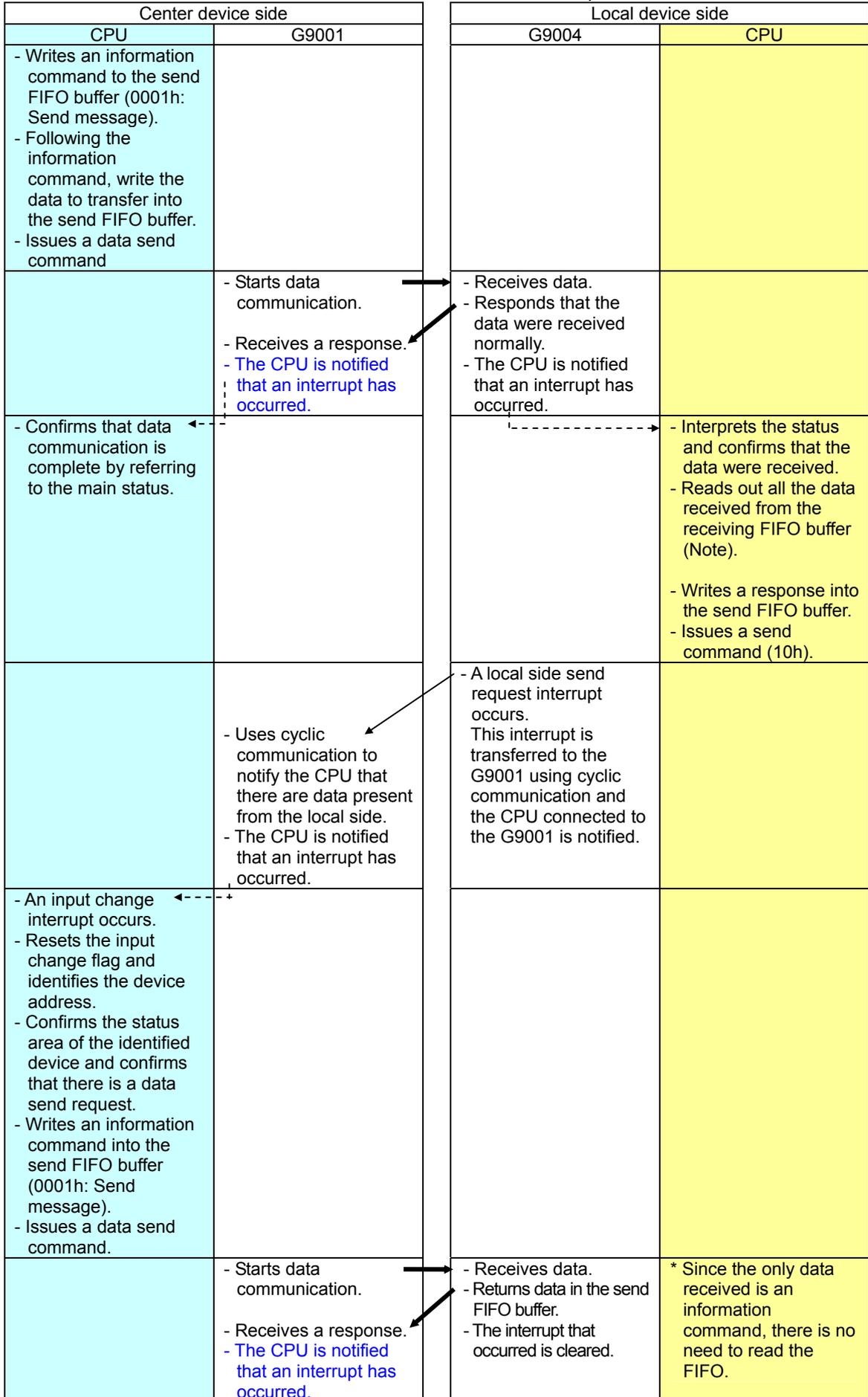
G9001: Bit 10 of the main status

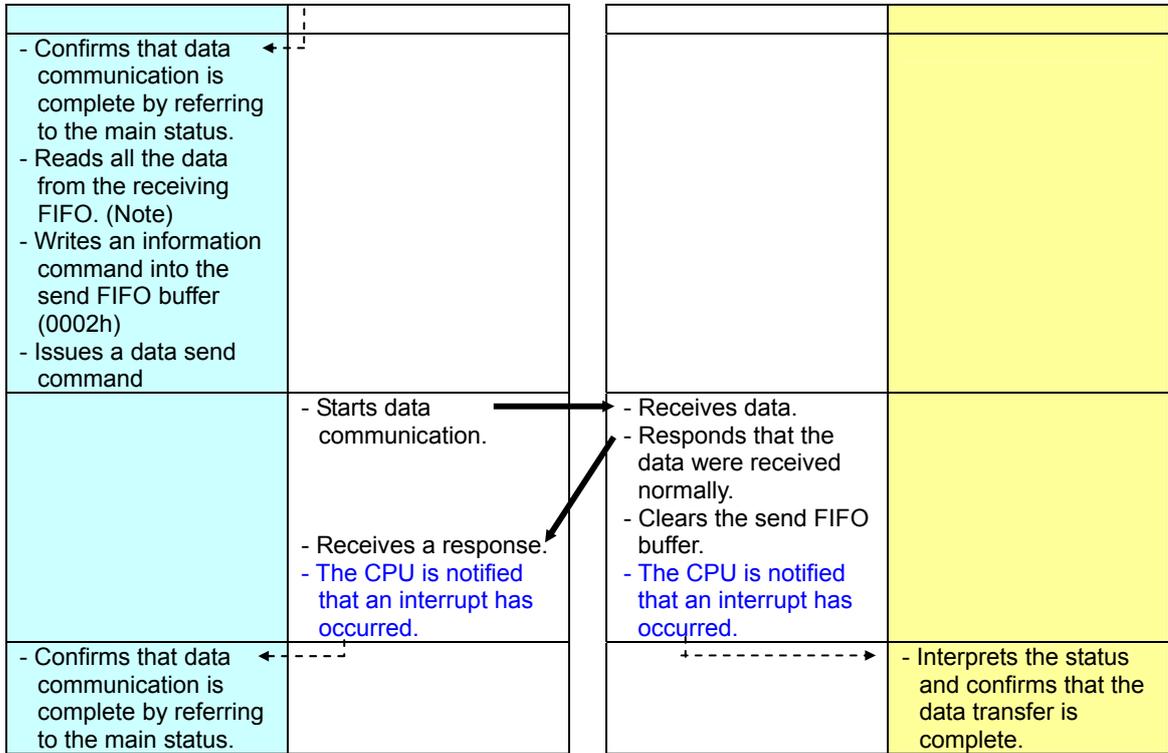
G9004: Bit 5 of the status

The bold arrows in the figure above indicate the data communication direction.

Dotted lines show an interrupt sent to a CPU.

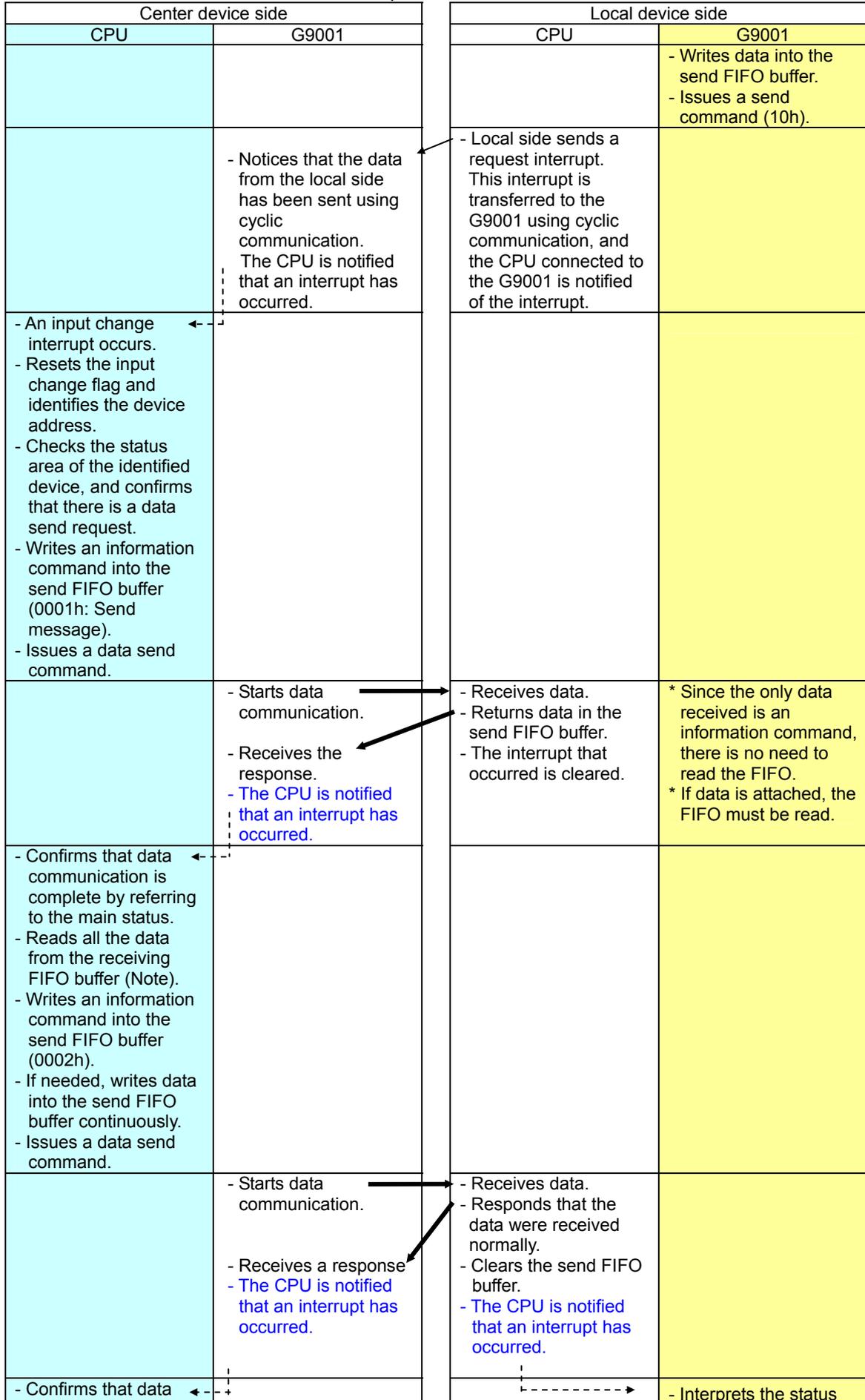
12-3. Message communication mode 2 (the CPU connected to center device sends data to the CPU connected to a G9004 and receives the data sent back).





Note: When reading data from the receiving FIFO buffer, repeat the read process until the following flags change to "0", which will mean that you have read all the data in the FIFO buffer.  
 G9001: Bit 10 of the main status  
 G9004: Bit 5 of the status  
 The bold arrows in the figure above indicate the data communication direction.  
 Dotted lines show an interrupt sent to a CPU.

12-4. Message communication mode 3 (Transfer data from the CPU connected to the G9004 to the CPU connected to the center device)



communication is complete by referring to the main status			and confirms that the data transfer completed normally. - Interprets the status and if there is received data, reads the data from the receiving FIFO. * If the received data is only an information command, there is no need to read the FIFO.
---	--	--	--

Note: When reading data from the receiving FIFO buffer, repeat the read process until the following flags change to "0", which will mean that you have read all the data in the FIFO buffer.

G9001: Bit 10 of the main status

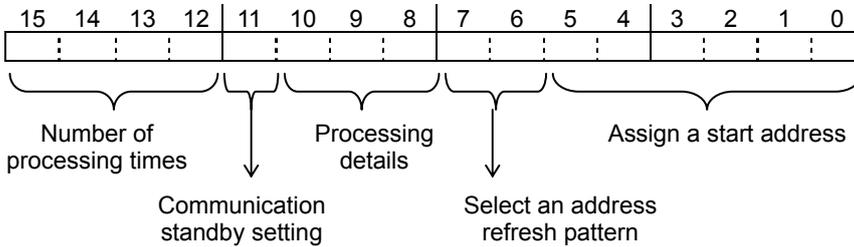
G9004: Bit 5 of the status

The bold arrows in the figure above indicate the data communication direction.

Dotted lines show an interrupt sent to a CPU.

12-5. Commands in the CPU emulation mode

This section describes the commands that the G9001 (center device) can send to the G9004. In the message communication mode, only three commands can be used and they are simple. Therefore, we have omitted a description of these commands. However, in the CPU emulation mode, the commands that can be used are a little bit more complicated. The details are given below.



1) Number of processing times

When you want to start a process from a specified start address and use the specified processing details repeatedly, enter the number of times to repeat it here (the required number of times -1). If you enter zero here, the process will be done once.

2) Communication standby setting

Enter a response pattern for the G9004 when it receives a command. The details are described later.

3) Processing details

The following processes can be used.

- 000: NOP ..... Invalid command
- 001: Write process ..... The G9004 writes data to peripheral devices it controls.
- 010: Read process ..... The G9004 reads data from peripheral devices it controls.
- 011: Data send process ..... Requests data that is stored in the G9004 FIFO buffer to be sent.
- 100: Reset the "receive process complete" interrupt ..... Clears "receive process complete" interrupt that occurred on the G9004.
- 101: Reset the "LIRQ input" interrupt ..... Clear "LIRQ input" interrupt that occurred on the G9004.

4) Select an address refresh pattern

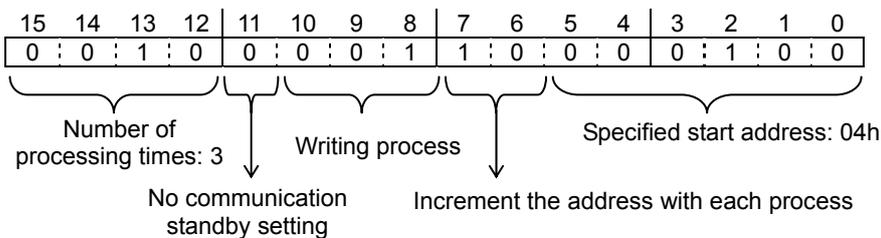
When the number of times to repeat the process is specified, the following options can be selected.

- 0x: Fixed address
- 10: Increment the address with each repeat of the process (+2 for 16-bit addresses)
- 11: Decrement the address with each repeat of the process (-2 for 16-bit addresses)

5) Assign a start address

Specify a top address at which to start processing.

For example, when "2184h" is entered, the G9004 will do the following.

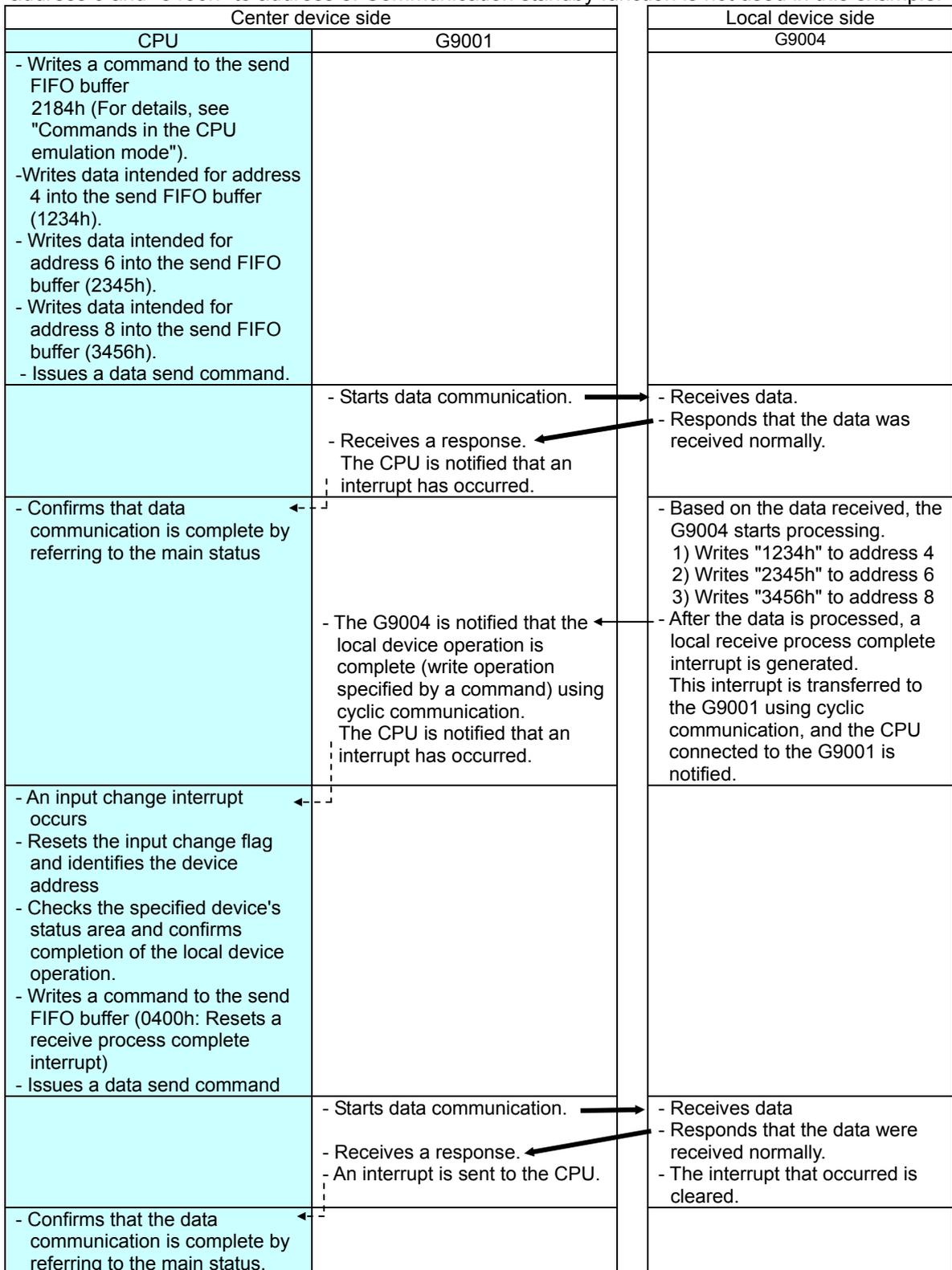


With the operation above, the G9004 will write information three times, starting at address 4, without waiting for a response.

The data to write must follow this command. For details, see the next page.

12-6. CPU emulation mode 1 (Write sequential data)

Suppose that you want the G9004 to write "1234h" to address 4 (address bus: 000100b) on a peripheral device it controls. You want it to repeat the process 2 more times by writing "2345h" to address 6 and "3456h" to address 8. Communication standby function is not used in this example.



Note: When reading data from the receiving FIFO buffer, repeat the read process until the following flags change to "0", which will mean that you have read all the data in the FIFO buffer.

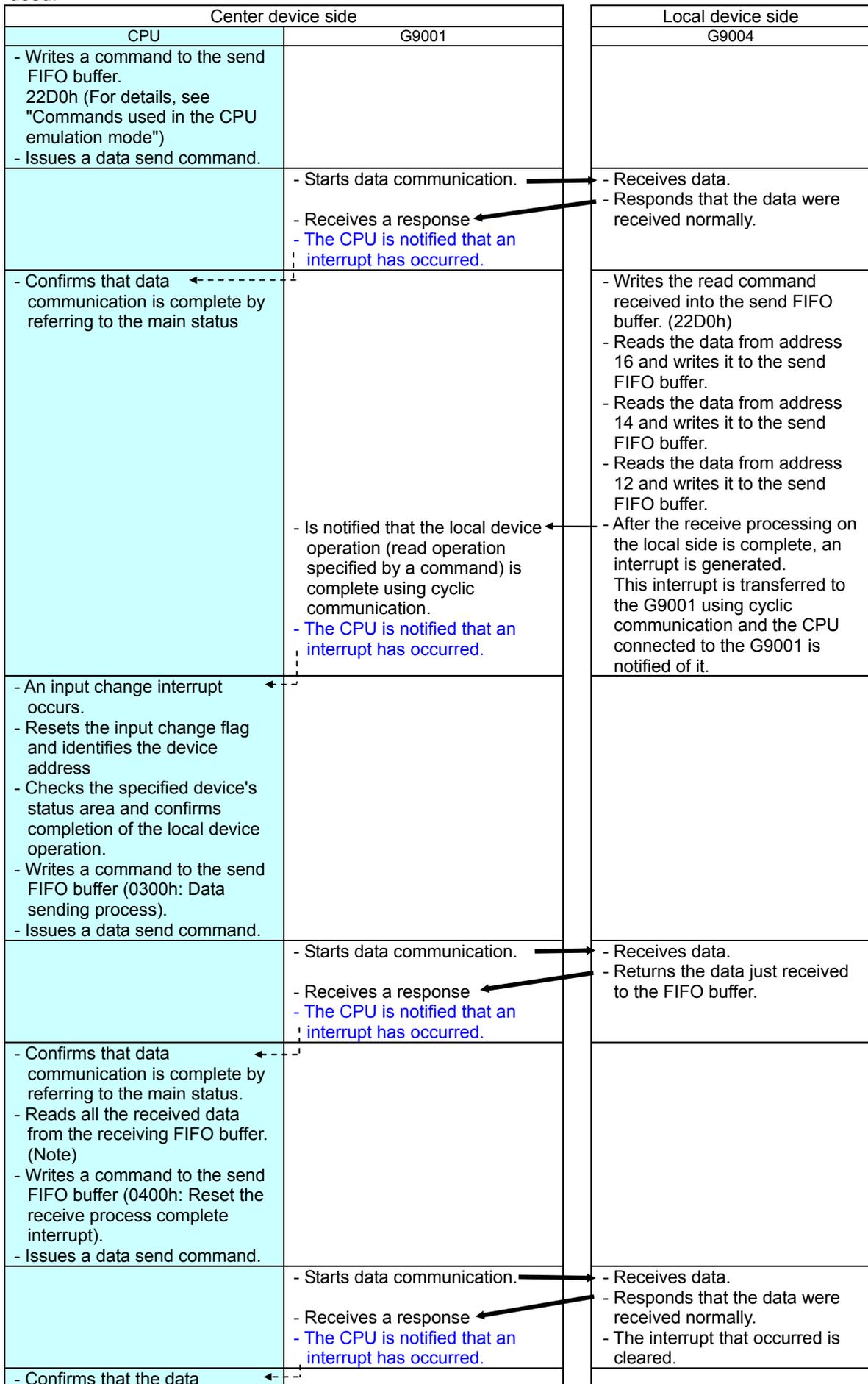
G9001: Bit 10 of the main status

The bold arrows in the figure above indicate the data communication direction.

Dotted lines show an interrupt sent to a CPU.

12-7. CPU emulation mode 2 (Reading sequential data)

Imagine that the G9004 reads 3 words of data from addresses 16 (address bus: 010000b), 14 and 12 on the peripheral device it controls. Also assume that the communication standby function is not used.



communication is complete by referring to the main status.		
--	--	--

Note: When reading data from the receiving FIFO buffer, repeat the read process until the following flags change to "0", which will mean that you have read all the data in the FIFO buffer.

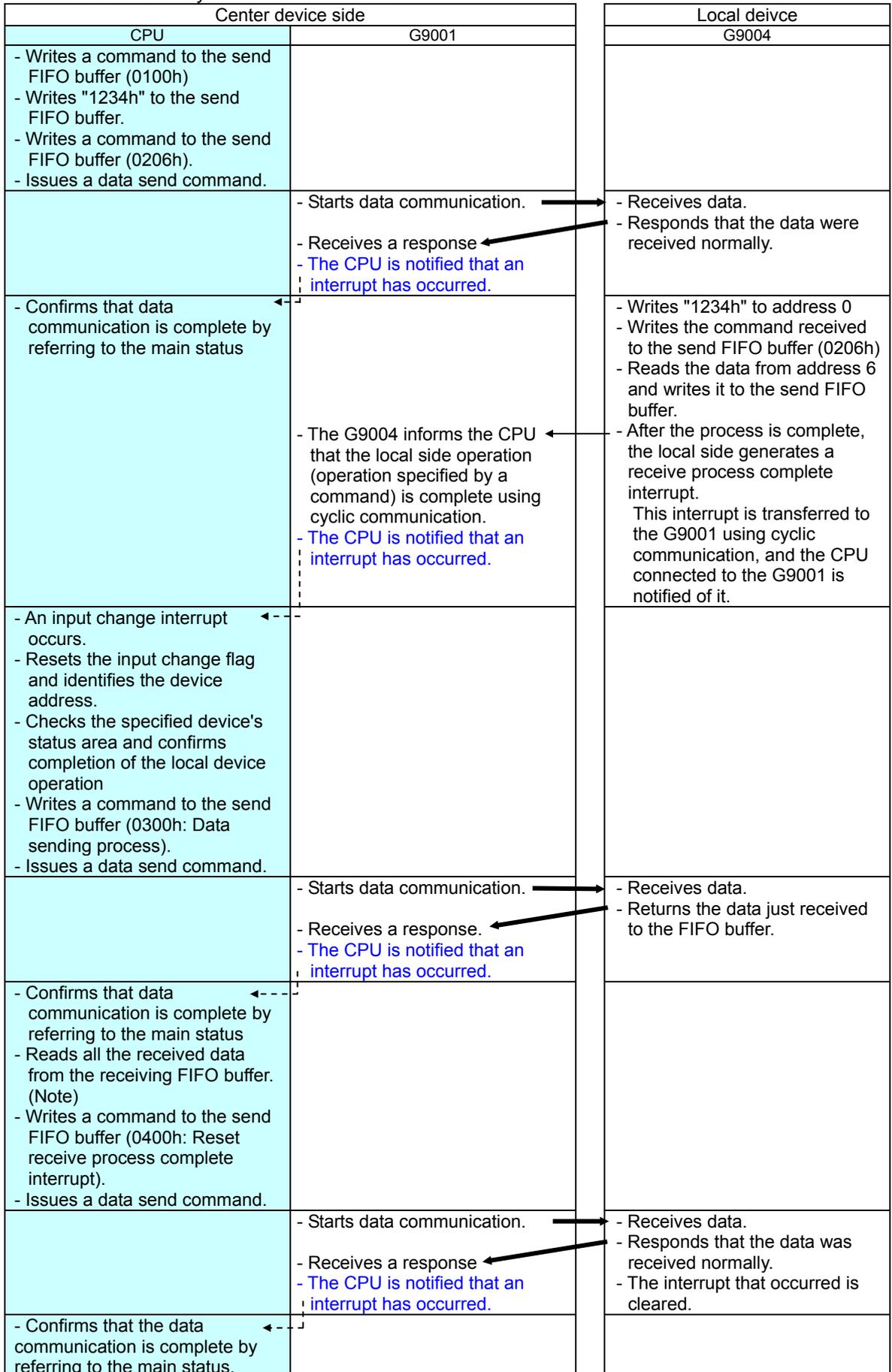
G9001: Bit 10 of the main status

The bold arrows in the figure above indicate the data communication direction.

Dotted lines show an interrupt sent to a CPU.

12-8. CPU emulation mode 3 (combined process)

Imagine that the G9004 writes "1234h" to address 0 (address bus: 000000b) of the peripheral device that it controls and reads 1 word of data from address 6. Also assume that the communication standby function is not used.



Note: When reading data from the receiving FIFO buffer, repeat the read process until the following flags change to "0", which will mean that you have read all the data in the FIFO buffer.

G9001: Bit 10 of the main status

The bold arrows in the figure above indicate the data communication direction.

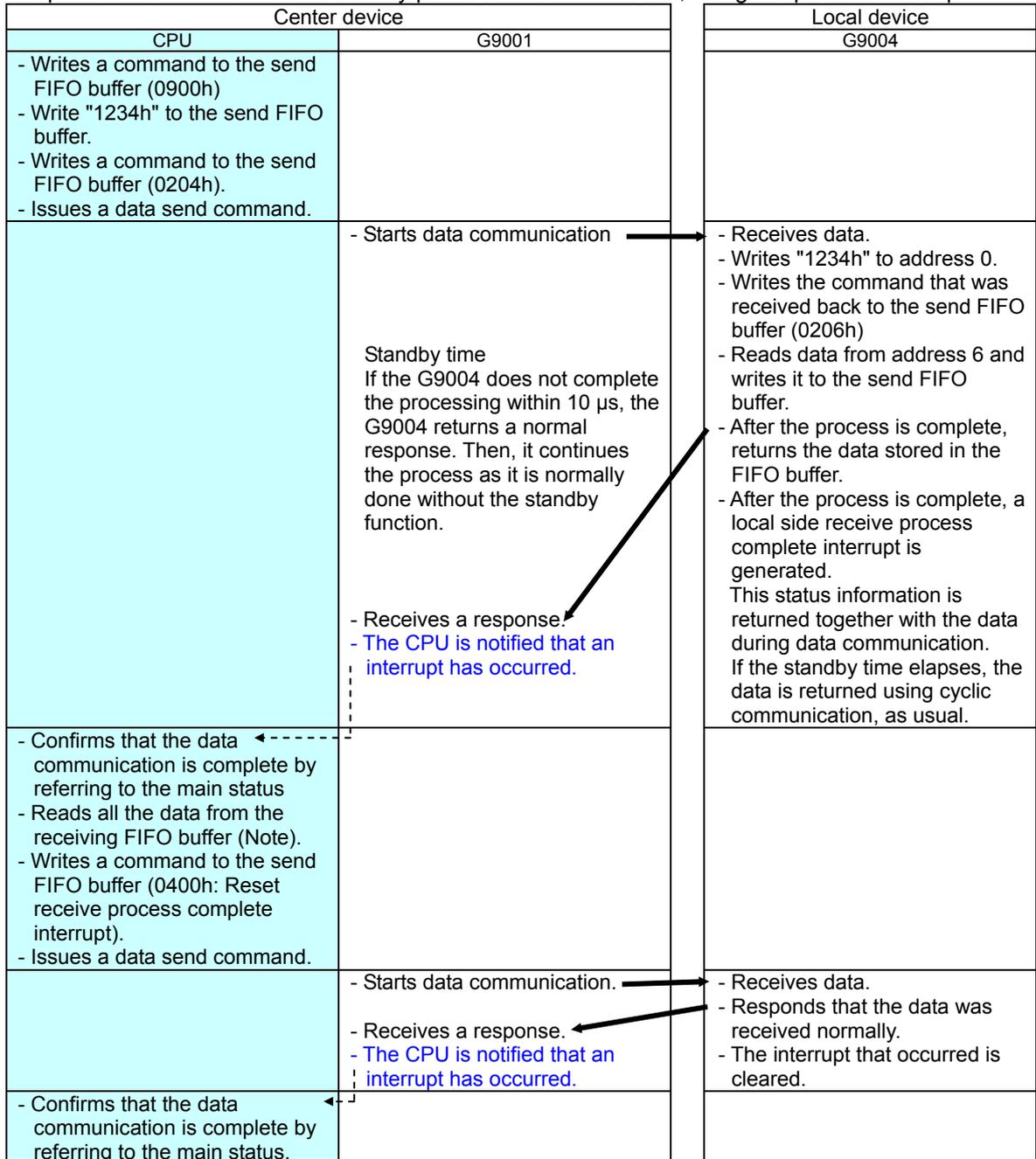
Dotted lines show an interrupt sent to a CPU.

12-9.CPU emulation mode: What is the communication standby setting?

If the standby process is not selected, after receiving a command from the G9001, the G9004 immediately returns a response that it received the data normally. Then it interprets the command and executes the operation.

On the other hand, if the standby process is selected, after receiving a command, the G9004 does not respond immediately. First, it interprets the command and executes the operation. After executing the command, the G9004 returns the response that the operation ended normally (together with the data read, if any). However, if execution of the command takes 10µs (at 20 Mbbs) or more, the G9004 returns an intermediate response that the command was received normally, and then carries out the procedure the same as described just above.

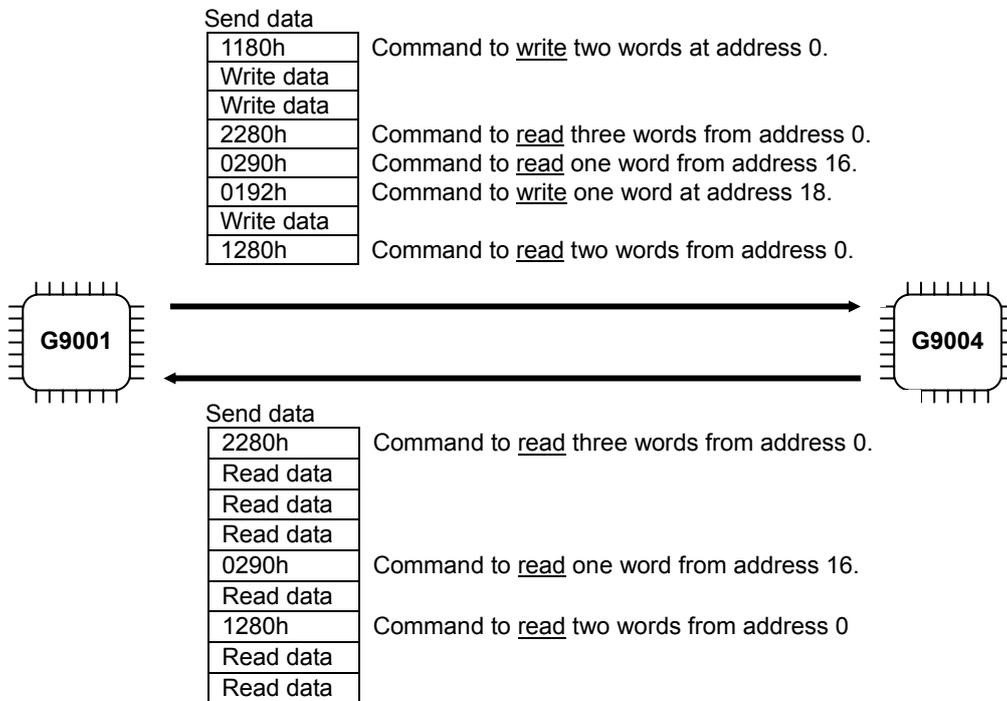
The procedures involved in the standby process are shown below, using the previous example.



The first command received determines whether the G9004 will use the communication standby function or not.

Therefore, if the first command does not require use of the communication standby function, the G9004 will ignore the communication standby function setting in the next command (and vice versa).

12-10. CPU emulation mode: Data format that is returned by the G9004 when using a read command  
 The G9004 reads data from the peripheral devices it controls. However, please note that the returned data contains not only the data read but also the commands used to get the peripheral device to send this data.  
 For example, if write commands and read commands are mixed (the same is also true just with read commands), the G9004 will execute the group of commands it receives.  
 After execution, the data read are returned to the G9001. The returned data will be formatted as shown below.



First, the read commands are sent to the G9001 without any changes. Then, the data actually read follows.

The CPU connected to the G9001 can carry out various processes while comparing the read commands it sent with the data that was read. (There is no need for the CPU to store the read commands it sent.)

12-11. CPU emulation mode: When the data returned by a read command exceeds 128 words  
 When the G9004 reads data from a peripheral device it controls using read commands, the read commands are attached at the top of the data returned. This was described in the previous section.

As above, more data may be returned from the G9004 than was expected. (When setting the amount of processing time, the amount of data will be even larger.)

If more than 128 words of data are returned, the condition of the excess data cannot be guaranteed.

Errors may occur. Pay close attention to the amount of data that is returned.

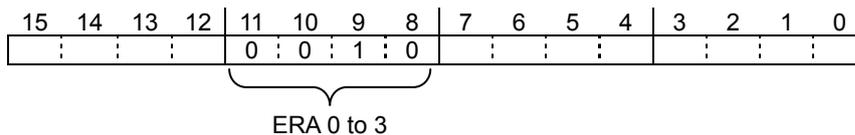
**13. If I try to perform data communication (transient communication) with an I/O device, what will happen?**

Do not try doing data communication with a G9002 (I/O device).

The G9002 does not have a receiving FIFO, so it cannot receive any data you send.

If you try data communication with a G9002, the G9002 will return an error. In this case, the interrupt status of the G9001 (center device) will be as shown below (In the figure below, unrelated bits are left blank, since they have no meaning in this description), and an interrupt occurs.

Interrupt status (ISTW)



**14. If a different PMD is placed between an I/O device and a center device, what will happen?**

The G9002 (I/O device) determines the I/O attributes of the input/output terminals on the four ports according to the setting of the external PMD terminals.

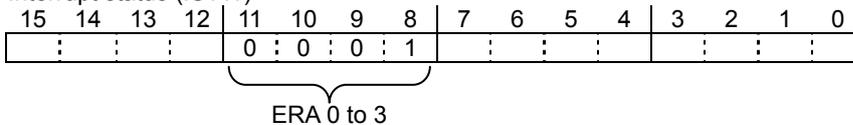
The G9001 (center device) has an area in RAM to keep track of the PMD terminal status of each G9002. During cyclic communication, the G9001 exchanges port data after referring to the PMD status memory area in RAM. In other words, the PMD data in the G9001 RAM and the actual PMD terminal status of the G9002 should always match.

Before starting data communication, the G9001 performs a system communication. When the G9001 automatically collects the information about local devices connected to the bus, it also collects the PMD terminal status of each G9002 that is connected. Therefore, the PMD information inside the RAM should never be different from actual PMD terminal status of a G9002.

However, if the PMD area in RAM is set manually by the CPU, the settings may be different from the actual condition of the G9002. If the G9001 starts cyclic communication while the settings between the RAM and actual PMD terminals are different, the G9002 will receive the wrong cyclic communication and send back an error.

In this case, the interrupt status of the G9001 (center device) will be as shown below (In the figure below, unrelated bits are left blank, since they have no meaning in this description), and an interrupt occurs.

Interrupt status (ISTW)



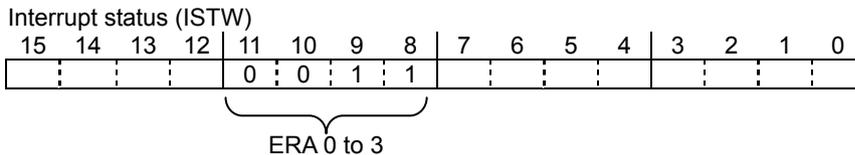
## 15. What happens if more data is sent to the PCL device than the receiving FIFO can handle?

The send FIFO in the G9001 (center device) can store 128 words (256 bytes).

On the other hand, the receiving FIFO of the G9003 (PCL device) can only hold 4 words (8 bytes).

If the CPU writes more than 9 bytes of data to the send FIFO of the G9001 and then try to send the data to the G9003, the G9003 will return an error.

In this case, the interrupt status of the G9001 (center device) will be as shown below (In the figure below, unrelated bits are left blank, since they have no meaning in this description), and an interrupt occurs.



Note: Although the G9003 FIFO has a 4 word capacity, it actually uses a maximum of 3 words. If 4 words of data are sent, the G9003 will not declare an error. However, the communication will not be guaranteed.

Basically, data should be sent to the PCL device in units of one command.

## 16. What will happen if the G9004 does not read the data in the receiving FIFO?

When the G9004 is used in message communication mode, the G9004 is controlled by a CPU that is different from the CPU controlling the center device.

Then when the G9004 receives a data communication from the G9001 and the data received are stored in the receiving FIFO, the CPU in the G9004 has to read the data in the receiving FIFO.

Now, suppose the CPU in the G9004 is busy with other processing. If the process that reads data in the receiving FIFO is delayed, what will happen?

The G9004 will not receive the next data communication if data are left in the receiving FIFO (it ignores the data sent next). When the G9001 tries to send data, the G9004 will not respond. The G9001 will stop trying to send data after making three attempts and the data communication will be considered to have failed.

Therefore, if the G9004 receives data, the data in the receiving FIFO should be read as soon as possible. (Resetting the receiving FIFO is also useful.)

However, even if data remain in the G9004 receiving FIFO, it will normally respond to cyclic communication.

**17. What happens to each device when communication is not possible?**

If a problem occurs with the communication circuit, and the G9001 (center device) cannot communicate with the local devices, what will happen?

The local devices cannot do anything since no communication is coming from the G9001.

In this case, the watchdog timer integrated into each local device will be triggered.

This timer counts the time specified on the external terminals of each device. If a local device receives communication before the counter reaches the specified elapsed time, the counter will clear itself.

(Actually the timer is cleared after responding to the cyclic communication.)

If there is no communication before the counter reaches the specified time, the device does the following. (The process is selectable using the external TUD terminal.)

TUD external terminal	Processing
H	When latched, if the local device status times out, it just waits for communication. In this case, the external terminal outputs are held in the condition they were in when the timer timed out. The PCL and other devices will continue to operate if they have internal processes being executed.
L	Resets the device circuit and waits for communication. The status of the external terminals is initialized.

### 18. When a local device stops functioning, how can it be identified?

If any local device loses power or is disconnected from the communication line (accidents, intentional disconnects, etc.) during cyclic communication, the device will not respond to a cyclic communication from the G9001 (center device).

If there is no response after three sequential cycles of cyclic communication, the G9001 assumes that the local device is no longer present. (This device will not be removed from the cyclic communication list. To remove this device from the cyclic communication list, erase it using the CPU.)

In this case, bit 2 (EIOE) of the G9001 main status will change to "1."

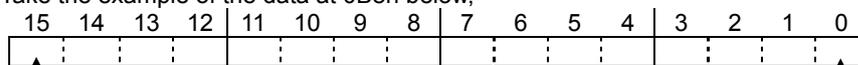
The device address can be confirmed by checking the following area of the G9001 internal RAM (I/O communication error flag).

Address	Description (16 bit)
000h through 0B6h	Used for other purposes
0B8h	I/O communication error flag for device addresses 00 to 15.
0BAh	I/O communication error flag for device addresses 16 to 31.
0BCh	I/O communication error flag for device addresses 32 to 47.
0BEh	I/O communication error flag for device addresses 48 to 63.
0C0h through 1FEh	Used for other purposes

Note: The address map above is correct when I/F mode 3 (IF1=H, IF0=L) is selected.

This data can be interpreted as follows.

Take the example of the data at 0B8h below,



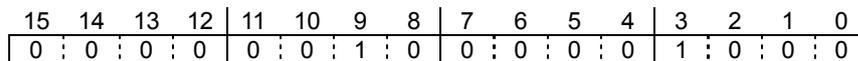
The device at address 15 has an error.

The device at address 00 has an error.

As seen above, the lowest bit of the data indicates an I/O communication error for the device at the lowest address.

If this bit is "1," the G9001 cannot perform cyclic communication with the corresponding local device.

Now assume that the data at 0BAh is as follows.



Device address 25

Device address 19

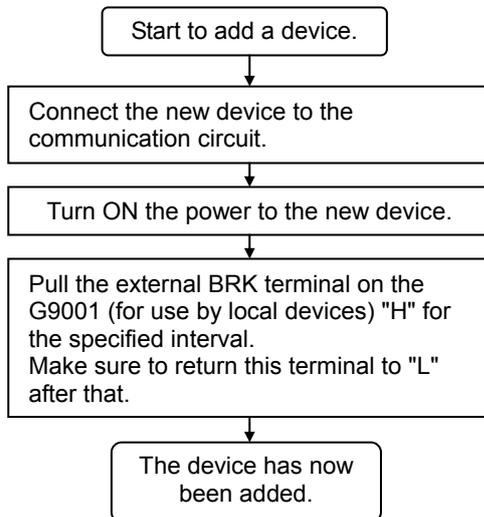
The lowest bit of data at 0BAh corresponds to device address 16. Counting up from this bit in the order, the bits that are "1" are at device addresses 19 and 25.

To clear the bits that have changed to "1," write a "1" to these bits.

## 19. How can a new device be added during operation?

When a device that has dropped out of communication becomes available again, or if a new local device is added to the circuit, cyclic communication can be done continuously, without stopping.

The procedure for adding a device is shown below.



The G9001 (center device) monitors whether new devices have been added at certain intervals. If a device outputs an "H" to the external BRK terminal for a certain period (3.2 ms or longer), the G9001 will see that a new device is available.

After noticing that there is a new device, the G9001 will turn bit 1 (BRKF) in the main status to "1," and generate an interrupt to the CPU.

The CPU, which receives this interrupt, will issue a "system communication" command to the G9001.

Now the G9001 can get the information about the new local device and include it in cyclic communication.

**Note:** If the local device's electrical system is not insulated from the communication circuit using a pulse transformer or similar device, its line driver may be broken by noise on the circuit.

If this occurs, turn off the power to all of the systems connected to the communication circuit, and then restart the connection operation.

## 20. How can the communication status be monitored from outside?

Devices have external terminals for evaluating whether there is an abnormality in the communication. The terminal status of local devices is different from the G9001 (center device). See the table below for the differences.

Device type	Terminal name	Description
Center device	MCRY	Indicates the status of the communication circuit. If data is present in the circuit, this terminal is "L."
	MERR	In the following cases, the terminal goes "L" for 3.2 $\mu$ s. 1) Data could not be received normally from a local device (such as an error in the check sum). 2) After completing communication to a local device, if there is no response after 20 $\mu$ s (this is treated as no response).
Local device	MSEL	Succeeded in receiving data If the local device receives communication from the center device normally, this terminal goes "L" for 3.2 $\mu$ s.
	MRER	Failed to receive Could not receive communication from the center device normally (such as an error in the check sum), this terminal goes "L" for 3.2 $\mu$ s.
	TOUT	Stopped communication If there is no communication by a local device for a certain period, the watchdog timer times out. At that time, this terminal goes "L." When communication is resumed, this terminal goes "H."

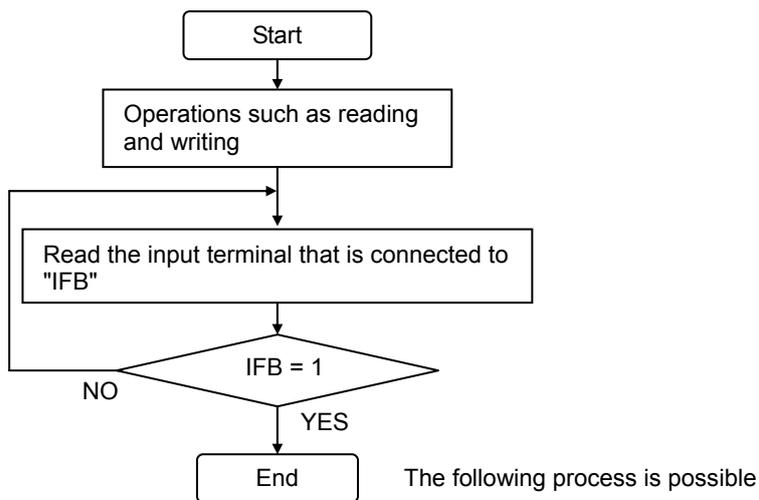
## 21. How can CPUs with no wait control terminal be used?

The G9001 (center device) and G9004 (CPU emulation device) are controlled by CPUs. For this reason, internal processing time is needed for read and write operations from the CPUs.

The G9001 has a "WRQ" terminal ("LWRQ" terminal on the G9004) so that during this internal processing the CPU will not keep going. Normally, when this signal is connected to a wait terminal on the CPU, the CPU will automatically wait.

However, some CPUs do not have a wait terminal. For these CPUs, the CPU can be made to wait using the following procedures (you may also make a large dummy loop to make it wait).

Preconditions: Connect any CPU input port terminal (any terminal that is available) to the "IFB" terminal ("LIFB" on the G9004).



Please see "Connection to a CPU that does not have a wait function" (Section1-1-5 in Chapter 3) in the G9001 user's manual. It contains a detailed description of this issue.

## **22. How to control the G9001 using only two address lines**

The G9001 (center device) has 9 address lines.

This is used to access the 512 bytes of internal RAM directly. Normally, all 9 address lines from a CPU (8 if a 16-bit CPU is used) are connected to these address lines.

However, the G9001 can be controlled by connecting just two address lines (3 with an 8-bit CPU).

This method uses a function that accesses the RAM with a command to the G9001, and the same operation can be used with any of our ASIC PCL series products.

Please see "Command" (Section 4-8 in Chapter 1) in the G9001 user's manual. It contains a detailed description of this issue.

## **23. What are the precautions for using a 40 MHz clock?**

The G9000 series devices can select a system clock speed of 40MHz or 80MHz. With either of these clock rates, the functions are the same. So, what criteria should be used to select a system clock?

Our recommended clock rate is 80MHz. A filter may be inserted on this clock line.

However, in the small scale systems shown below, a 40MHz clock may also be used.

- The total number of local devices is 10 or so and the total cable length less than 20m.

However, in these conditions, if the cable or parts qualities are lower, communication may not be normal, even at 80MHz.

## 24. What is the selection standard for related parts?

Communication parts quality greatly affects communication quality. Whenever possible, select better quality parts. However, if the definition of high quality is not clear, you cannot select good parts. Shown below are some rough selection standards.

### 24-1. Cable selection

Select shielded LAN cables. (If twisted pair lines will be quite long, use pairs with the loosest twist.) We recommend category 5 or better cables.

Cables specially designed for use as field buses, such as CC-Link, LOWORK etc., are also high quality (and high cost).

Quality warranted cables come with a performance characteristic table. A performance characteristic table contains the warranted quality values. (A simple description of the materials is not a performance characteristic table. At a minimum, a specification for attenuation is required.)

Cables that don't have a performance characteristic table should not be used.

### 24-2. Selecting an RS485 device

You must choose devices that can send and receive data at 20Mbps or faster.

Also, select devices that have a large input resistance, 24k-ohms or more is desirable (12k-ohms is standard).

Either 5V or 3.3V devices can be selected. However, 5V devices are generally better (although the G9000 series will run on 3.3V).

Also, choose devices that have a fail-safe function.

### 24-3. Selecting pulse transformers

You must choose devices that can handle 20Mbps.

The transformer must respond cleanly to 20Mbps and must not saturate at 4Mbps.

Use transformers with about 1000 $\mu$ H of inductance. (Transformers with a larger inductance than this are acceptable. Transformers with only a few hundred  $\mu$ H of inductance cannot support 64 local devices. These recommendations are based on the condition that 5V RS485 devices are used.)

### 24-4. Determining the proper termination resistance

Termination resistors are indispensable at both ends of the communication circuit.

The proper resistance value is determined by the cables used.

The "Characteristic impedance" described in the characteristic table for the cable you use will be the termination resistor value.

## **25. When is a level shifter is needed?**

The G9000 series can run from a single voltage of 3.3V.

However, in most cases, 5V devices can be connected directly to the G9000 series models. (For details, see the individual user's manual.)

However, 5V devices cannot be connected directly to the terminals related to communications (outputting signals such as "SO," "SOEL," and "SOEH, and receiving signals such as "SI"), since high speed communication is needed.

In other words, when a 5V product is selected as the RS485 device, a change to the appropriate voltage is needed using a level shifter, etc.

Summed up below are the features of Motionnet.

- <> Communication cycle and number of I/O points  
Motionnet can support up to 2048 I/O points and the communication speed is as high as 20Mbps.
- <> Storage of I/O data  
The MKYseries (NPM's conventional model) has external RAM. With Motionnet the RAM is internal, which really helps to reduce the number of parts and cost of constructing a communication system.
- <> Address settings for local devices  
Since Motionnet can automatically set the address for a local device using the system communication mode, the address can be selected at random.
- <> Plug and play function  
When adding a new local device, the G9001 notifies the CPU that a new device has been identified when a pulse is input on the G9001 break terminal. Therefore, it supports plug and play operation.
- <> Monitor communication status  
The watchdog timer function provides a monitor of the communication status.  
If the data send interval from the G9001 (center device) to a local device exceeds the specified time, the local device watchdog timer will time out. When a timeout occurs, the current status can be latched and the reset status can be specified using the terminals.
- <> Setting the input/output ports  
Compared with the MKY series which has fixed input/output ports (2 input ports and 2 output ports) Motionnet devices can specify ports as inputs or outputs, so it can be used efficiently and it is easy to change or add connections.
- <> Axis control
  - Motionnet is available with a G9003 single axis controller. It offers a variety of control patterns including continuous operation, positioning operation, and zero return operation using constant speed, linear acceleration/deceleration, and S-curve acceleration/deceleration.  
If all of the local devices connected are G9003s, motion control can be accomplished for 64 axes while reducing the wiring substantially.
  - If a G9004 is used in the CPU emulation mode, two PCL6045s (NPM's 4-axis control LSI) can be connected to one G9004. Therefore, if all of the local devices are G9004s, motion control of 512 axes is possible (64 x 4 x 2).
- <> Data communication between CPUs  
If the G9004 is used in the message communication mode, it can send a maximum of 127 words of message data (in any format) in one cycle, which means that the CPU controlling the G9001 and a G9004 can pass data freely.

## Alphabetical Index

BRK.....	45	MRER .....	46
Center device.....	2	MSEL .....	46
Characteristic impedance .....	49	Multi-drop system.....	4
Communication cycle.....	8	PCL device.....	2
Communication standby setting .....	34, 40	PMD .....	2
CPU emulation device .....	3	Polling/selecting system .....	5
Cyclic communication .....	5	Power isolation.....	10
Data communication .....	6	Pulse transformer.....	10
Data communication command .....	16	Reflection .....	12
Device address .....	4	RS485 .....	10
Device information area.....	22	SIA .....	13
EIA-485 standard.....	10	SIB .....	13
FIFO.....	18	Status .....	8
G9004 status .....	28	SO .....	13
IF (I/F) .....	12	System clock.....	49
I/F mode.....	12	System communication.....	15
Input change interrupt.....	22	Termination resistance .....	11
Input change interrupt flag area.....	23	Time up .....	43
Input change interrupt setting area.....	22	TOUT .....	46
I/O communication error flag .....	45	Transceiver function.....	10
I/O communication start.....	15	Transient communication.....	6
I/O device.....	2	Transformer.....	10
Local device.....	2	TUD.....	43
LWRQ .....	47	Wait control terminal .....	48
MERR .....	46	Watchdog timer .....	43
Message communication mode.....	3	WRQ .....	48
MCRY .....	46		

1st version created on June, 2005

No. DZ10001-0