Motionnet

RemoteI/O & RemoteMotion

<u>G9103A</u>

(Motor control LSI with interpolation function)

User's Manual

NPM Nippon Pulse Motor Co., Ltd.

[Preface]

Thank you for considering our super high-speed serial communicator LSI, the "G9000 series".
To learn how to use the G9000 series, read this manual to become familiar with the product.
The handling precautions for mounting this LSI are described at the end of this manual. Make sure to read them before mounting the LSI.

[What the Motionnet is]

As a next generation communication system, the Motionnet can construct faster, more volume large scale, wire saving systems than conventional T-NET systems (conventional LSI product to construct serial communication system by NPM). Further, it has data communication function, which the T-NET does not have, so that the Motionnet can control data control devices such as in the PCL series (pulse train generation LSI made by NPM).

The Motionnet system consists of one center LSI connected to a CPU bus, and maximum 64 local LSIs, and they are connected by using cables of two or three conductive cores.

[Cautions]

(1) Copying all or any part of this manual without written approval is prohibited.

(2) The specifications of this LSI may be changed to improve performance or quality without prior notice.

(3) Although this manual was produced with the utmost care, if you find any points that are unclear, wrong, or have inadequate descriptions, please let us know.

(4) We are not responsible for any results that occur from using this LSI, regardless of item (3) above.

■ Descriptions of indicators that are used in this manual

1. In description of register bits, "n" refers to a bit position and "0" refers to a bit position that can be written with only "0." It also means that a bit will always be read as "0".
   Specified bit of specified register is referred to as (resister name).(bit name). (ex. RMD.MSDE)

2. Unless otherwise described, time description affected by the reference clock frequency discussed in this manual is in the case of CSK=40 MHz.

3. Terminals with a bar (an upper line) over the name use negative logic.

   Example: $\overline{\text{TOUT}}$ means that TOUT terminal uses negative logic.

Table of Contents

1. Outline ···· 1

2. Features ···· 1

3. Specifications ···· 6

3-1. Basic specifications ···· 6

3-2. Communication system specifications ···· 6

3-3. Specifications for the axis control section ···· 7

4. Hardware description ···· 8

4-1. A list of terminals ···· 8

4-2. Terminal allocation diagram ···· 11

4-3. Entire block diagram ···· 12

4-4. Functions of terminals (The number in the parentheses [ ] is the terminal number.) ···· 13

5. Description of the software ···· 20

5-1. Outline of control ···· 20

5-1-1. Communication control ···· 20

5-1-2. Communication type ···· 20

5-2. Functional settings for the G9103A ···· 24

5-2-1. I/O port ···· 24

5-3. Command (Operation commands & Control commands) ···· 26

5-3-1. Operation commands ···· 26

5-3-2. Control commands ···· 28

5-3-3. Register control commands ···· 31

5-4. Register and pre-register ···· 39

5-4-1. Pre-register ···· 39

5-4-2 Register for setting speeds ···· 40

5-4-3. Feed amount setting registers ···· 43

5-4-4. Environment setting registers ···· 45

6. Operation Mode ···· 67

6-1. Continuous operation mode using command control ···· 67

6-2. Positioning operation mode ···· 67

6-2-1. Positioning operation (specify a target position using an incremental value) (MOD: 41(h))67

6-2-2. Positioning operation (specify the absolute position in COUNTER 1) (MOD: 42(h)) ···· 67

6-2-3. Positioning operation (specify the absolute position in COUNTER 2) (MOD: 43(h)) ···· 68

6-2-4. Command position 0 return operation (MOD: 44(h)) ···· 68

6-2-5. Machine position 0 return operation (MOD: 45(h)) ···· 68

6-2-6. One pulse operation (MOD: 46(h), 4E(h)) ···· 68

6-2-7. Timer operation (MOD: 47(h)) ···· 68

6-3. Pulsar (PA/PB) input mode ···· 69
6-3-1. Continuous operation synchronized with PA/PB (MOD: 01(h)) ···· 72
6-3-2. Positioning operations synchronized with PA/PB (specify incremental position) (MOD: 51(h)) ···· 73
6-3-3. Positioning operation synchronized with PA/PB (specify the absolute position to COUNTER 1) (MOD: 52(h)) ···· 73
6-3-4. Positioning operation synchronized with PA/PB (specify the absolute position in COUNTER 2) (MOD: 53(h)) ···· 73
6-3-5. Command position zero return operation synchronized with PA/PB (MOD: 54(h)) ···· 73
6-3-6. Mechanical position zero return operation synchronized with PA/PB (MOD: 55(h)) ···· 73
6-3-7. Interpolated operation synchronized with PA/PB ···· 74
6-4. Origin position operation mode ···· 75
6-4-1. Origin return operation ···· 76
6-4-2. Leaving the origin position operations ···· 89
6-4-3. Origin search operation ···· 89
6-5. EL or SL operation mode ···· 91
6-5-1. Feed until reaching an EL or SL position ···· 91
6-5-2. Leaving an EL or SL position ···· 91
6-6. EZ count operation mode ···· 92
6-7. Interpolation operation mode ···· 93
6-7-1. Linear interpolation operation ···· 94
6-7-2. Circular interpolation operation ···· 95
6-7-3. Continuous operation of interpolation operation ···· 97

7. Speed patterns ···· 99
7-1. Speed patterns ···· 99
7-2. Speed pattern settings ···· 100
7-3. Manual FH correction ···· 105
7-4. Example of setting up an acceleration/deceleration speed pattern ···· 109
7-5. Changing speed patterns while in operation ···· 110

8. Description of the functions ···· 111
8-1. Reset ···· 111
8-2. Position override ···· 111
8-2-1. Target position override 1 ···· 111
8-2-2. Target position override 2 (PCS signal) ···· 112
8-3. Output pulse control ···· 113
8-3-1. Output pulse mode ···· 113
8-3-2. Control the output pulse width and operation complete timing ···· 114
8-4. Idling control ···· 115

8-5. Mechanical external input control ········ 116
8-5-1. +EL, -EL signal ········ 116
8-5-2. SD signal ········ 116
8-5-3. ORG, EZ signals ········ 119
8-6. Motor driver I/F signal ········ 121
8-6-1. INP signal (for servomotor) ········ 121
8-6-2. ERC signal (for servomotor) ········ 122
8-6-3. CDWN signal (for stepper motor) ········ 124
8-6-4. ALM signal ········ 125
8-7. External start, simultaneous start ········ 126
8-8. External stop / simultaneous stop ········ 128
8-9. Emergency stop ········ 130
8-10. Counter ········ 131
8-10-1. Counter type and input method ········ 131
8-10-2. Counter reset ········ 134
8-10-3. Latch the counter and count condition ········ 135
8-10-4. Stop the counter ········ 136
8-11. Comparator ········ 137
8-11-1. Comparator types and functions ········ 137
8-11-2. Software limit function ········ 140
8-11-3. Out of step detection function for stepper motors ········ 141
8-11-4. IDX (synchronous) signal output function ········ 142
8-11-5. Ring counter function ········ 143
8-12. Backlash correction ········ 144
8-13. Vibration restriction function ········ 144
8-14. Excitation sequence for stepper motors ········ 145
8-15. General-purpose I/O terminals (P0 to P7) ········ 147
8-16. Interrupt output ········ 148
8-17. Synchronization function with other axes ········ 153
8-17-1. Synchronization function of the clock for motor control ········ 153
8-17-2. Simultaneous stop function ········ 156
8-18. Unit ID control function ········ 157
8-18-1. EEPROM control command ········ 158
8-19. CPU connection function ········ 159
8-19-1. Timing to read (When reading 4 bytes） ········ 160
8-19-2. Timing to write (when writing 4 bytes） ········ 161

9. How to calculate the communication cycle time ········ 162
9-1. Cyclic communication ········ 162
9-2. Time required for one complete data communication ········ 162
9-3. Total cycle time (including data communication) ········ 163

10. Electrical Characteristics 164
10-1. Absolute maximum ratings 164
10-2. Recommended operating conditions 164
10-3. DC characteristics 164
10-4. AC characteristics 165
10-4-1. Reference clock 165
10-4-2. Reset timing 165
10-5. Operation timing 166

11. Communication example 168
11-1. G9103A, line transceiver, and pulse transformer 168
11-2. Complete configuration 170

12. Recommended environment 171
12-1. Cable 171
12-2. Terminating resistor 171
12-3. Pulse transformer 171
12-4. I/F -IC 171
12-5. Parts used in our experiments 171
12-6. Other precautions 172

13. Software example 173
13-1. Environment and precautions used for the descriptions 173
13-2. Commands used 173
13-3. Center LSI (G9001A) address map 174
13-4. Center LSI status 175
13-5. Center LSI Interrupt status 176
13-6. Center LSI command 177
13-6-1. Operation command 177
13-6-2. Indirect access command 179
13-6-3 Register access command 179
13-7. Center LSI (G9001A) registers 181
13-7-1 RENV0 register 181
13-7-2 Error count register (RERCNT) 181
13-7-3 Cycle of cyclic communication register (RSYCNT) 181
13-7-4 Recipient address register (RDJADD) 182
13-7-5 Version inforamation register (RVER) 182
13-8. Examples of control software 183
13-8-1. Start of the simplest cyclic communication 183
13-8-2. Communication with port data 184
13-8-3. Data communication 185

14. External dimensions 188

15. Designing Precautions 189
  15-1. Precautions for hardware design 189
  15-2. Precautions for software design 189
  15-3. Mechanical precautions 189

16. Handling Precautions 190
  16-1. Design precautions 190
  16-2. Precautions for transporting and storing LSIs 190
  16-3. Precautions for mounting 190
  16-4. Other precautions 191

1. Outline

This LSI is an axis motor control local LSI for the Motionnet system. On receiving a command from the center LSI (G9001A), it can output high-speed pulses to drive stepper motors and servomotors.
Using a variety of speed patterns, including constant speed, linear acceleration/deceleration, and S-curve acceleration/deceleration, this LSI affords control of various actions including continuous feeding, positioning, origin return operations, linear interpolation, and circular interpolation.
If all LSIs connected to the center LSI are the G9103As, the system can be constructed to control up to 64 axes while saving wiring. Using communications, the system allows you to check the operation status and output interrupt request signals with various conditions.
This system for G9103A has high degree of compatibility with the system for G9003 with respect to both hardware and software. Therefore, G9103A can be mounted on a board for G9003 and operated by software for G9003. (There may be the case that software change is needed.)

2. Features

- Communications
  Maximum communication rate is 20 Mbps.
  The system can control up to 64 axes.
  If a communication is disconnected, the G9103A can stop outputting pulses and reset output ports.

- Power supply
  Single power supply voltage: +3.3 V.
  Interfaces with TTL-ICs are possible.

- Synchronization of the clock for motor control
  When G9103As are dispersed, quartz oscillators for the G9103As exist individually. In this case, the frequency error causes to make errors of the speed of each axis, and affects interpolation operation. G9103A has synchronization circuit of the clock for motor control and synchronize operation timing between more than one G9103A by fine adjustment for clock frequency. You can confirm the amount of fine adjustment and also set the limit value of the fine adjustment.

- Interrupt request signal output
  An interrupt request ($\overline{\text{INT}}$) can be output from the center LSI by various factors.

- Acceleration/Deceleration speed control
  Linear acceleration/deceleration and S-curve acceleration/deceleration are available.
  Linear acceleration/deceleration can be inserted in the middle of an S-curve acceleration/deceleration curve. (S-curve range setting)
  The S-curve range setting can specify each acceleration and deceleration independently. Therefore, you can create an acceleration/deceleration profile that consists of linear acceleration and S-curve deceleration, or vice versa.

- Speed override
  Feed speed can be changed in the middle of any feed operation.

- Overriding target position 1) and 2)
  1) Target position (feed amount) can be changed while the axis is operating in positioning mode. If the current position exceeds the newly entered position, the axis decelerates and stops (immediately stop when feeding at a constant speed), and then feed in the reverse direction for positioning.
  2) The motor starts operation like in the continuous mode and, when G9103A receives an external signal, it stops outputting after outputting the specified number of pulses.

- Pre-register function
  The pre-register for operation data is built in. During operation, the G9001A can send a next operation data and G9103A can continue the next operation after the current operation completes. The comparator 3 also has a pre-register function.

- Multiple registers package communication
  G9003 can write into or read from only one register in one communication. G9103A can write into or read from up to 21 registers in one communication. When multiple registers are selected, it allows for shorter total communication time.

- Broadcast communication function
  Commands for start, stop, acceleration, and deceleration can be sent from G9001A to G9103A of the specified groups (1 to 7) or all groups simultaneously. The groups are specified by terminals or software.

- Unit ID management function
  To distinguish the kind of product (unit), a 32-bit register is added. This register can be downloaded the initial value from external EEPROM automatically. (RGN0).
  The 32 bits consist of company ID (upper 20 bits) and model ID (lower 20 bits). Company ID is managed by NPM and model ID is managed by customers.

- CPU connection function
  Parameters for driver, such as gain adjustment, can be set through Motionnet when G9103A is mounted in a motor driver.
  Three 32-bit general-purpose registers (RGN1 to RNG3) are added. These registers can be accessed from an external CPU through 4-line serial communication.

- Triangle drive elimination (FH correction function)
  In the positioning mode, when there are a small number of output pulses, this function automatically lowers the maximum speed and eliminates triangle driving.

- Simultaneous start function
  Multiple axes controlled by the G9103As can be started at the same time using broadcast communication or an external signal.

- Simultaneous stop function
  Multiple axes controlled by the G9103As can be stopped at the same time using broadcast communication or an external signal. Additionally, when other axis stops by errors, the G9103A can also stop its axis.

- Excitation sequence for stepper motors
  This LSI can output excitation sequences for 2-phase unipolar and for 2-phase bipolar systems.

- A variety of counter circuits
  The following three counters are available separately for each axis.

Counter	Intended use	Count Input
COUNTER1	28-bit counter for control of the command position	Output pulses
COUNTER2	28-bit counter for mechanical position control (Can be used as a general-purpose counter)	Output pulses EA/EB input PA/PB input
COUNTER3	16-bit counter for controlling the deviation between the command position and the machine's current position, or 16-bit general-purpose counter with the synchronous signal output function.	Output pulses EA/EB input PA/PB input 1/4096 division clock of 40MHz Output pulses and EA/EB input Output pulses and PA/PB input EA/EB input and PA/PB input

  All counters can be reset by writing a command or by providing a CLR signal.
  The counter data can also be latched by writing a command, or by providing an LTC or ORG signal.
  All counters have a ring count function that repeats counting through a specified counting range.

- Comparator
  There are three comparator circuits. They can be used to compare target values and internal counter values.
  The counter to compare can be selected from COUNTER 1 (command position counter), COUNTER 2 (mechanical position counter) and COUNTER 3 (deviation, general-purpose counter).
  Comparators 1 and 2 can also be used as software limits (+SL, -SL).

- Software limit function
  You can set software limits using 2 comparator circuits.
  When the mechanical position arrives in the software limit range, the axes stop immediately or decelerate and stop. Then, the axes can only be moved in the direction opposite to its previous travel.

- Backlash correction function
  The LSI has a backlash correction function.
  Backlash correction corrects the feed amount each time the feed direction is changed.

- Synchronous signal output function
  This LSI can output pulse signals at the specified constant interval.

- Vibration restriction function
  When you specify a control constant in advance, G9103A adds one pulse for reverse feed and one for forward feed just before the motor stops.
  This function can decrease vibration that occurs when the motor stops.

- Manual pulsar input function
  By applying manual pulse signals (PA/PB), you can rotate a motor directly.
  The input signals can be 90 degree phase difference signals (1x, 2x, or 4x) or Two-pulse signals (Up and Down pulse).
  In addition to the magnification rates above, the LSI contains an integral pulse number magnification circuit that multiplies by 1x to 32x and a pulse quantity division circuit (1/2048 to 2048/2048).
  EL signal and software limit setting are enabled. When the LSI stops outputting pulses by this function, the axes can feed only in the opposite direction.

- Out-of-step detection function
  This LSI has a deviation counter which can be used to compare command pulses and encoder signals (EA/EB).
  It can be used to detect out-of-step operations and to confirm a position by using a comparator.

- Output pulse specifications
  The way how to output pulses can be selected form the Common pulse mode, the Two-pulse mode, or the 90 degree phase pulse difference mode (4x). The output logic can also be selected. The relation between the direction of operation (+/-) and motor rotation (CW/CCW) can be changed easily.

- Current down control
  In controlling stepper motors, the LSI can output the current down signals automatically for reducing excitation current while operation is stopping. At the start, G9103A restores the changed current value to the original and begins to rotate axes after a specified time has elapsed. After the operation has stopped and a specified time has elapsed, the motor becomes in the condition of the current down.

- Idling pulse output function
  This function outputs a specified number of pulses at the initial speed (FL) before a high-speed start acceleration operation.
  Even if value near to the maximum starting pulse rate is set during acceleration, this function is effective in preventing out-of-step operation for stepper motors.

- Operation mode

  The basic operations of this LSI consist of continuous, positioning, origin return and interpolation operation.

  By setting the optional operation mode bits, you can use a variety of operations.

  <Examples of the operation modes>

  1) Start/stop by command.
  2) Continuous operation, positioning operation, linear interpolation and circular interpolation operation using PA/PB inputs (manual pulsar).
  3) Origin return operation.
  4) Positioning operation using commands.
  5) Hardware start of the positioning operation using the $\overline{\text{STA}}$ input.
  6) Specified amount feed after turning ON the PCS input. (Override 2 for the target position)
  7) Linear interpolation by multiple G9103As.
  8) Circular interpolation by two G9103As.

- Variety of origin return sequences
  1) The axis feeds at constant speed and stops when the ORG signal is turned ON
  2) The axis feeds at constant speed and stops when a specified number of EZ signals are received after the ORG signal is turned ON.
  3) The axis feeds at constant speed, reverses when the ORG signal is turned ON, and stops when a specified number of EZ signals are received.
  4) The axis feeds at constant speed and stops when the EL signal is turned ON. (Normal stop)
  5) The axis feeds at constant speed, reverses when the EL signal is turned ON, and stops when a specified number of EZ signals are received.
  6) The axis feeds at high speed, decelerates when the SD signal is turned ON, and stops when the ORG signal is turned ON.
  7) The axis feeds at high speed, decelerates when the ORG signal is turned ON, and stops when a specified number of EZ signals are received.
  8) The axis feeds at high speed, decelerates and stops after the ORG signal is turned ON. Then, it reverses feeds and stops when a specified number of EZ signals are received.
  9) The axis feeds at high speed, decelerates and stops by memorizing the position when the ORG signal is turned ON, and stops at the memorized position.
  10) The axis feeds at high speed, decelerates to the position stored in memory when a specified number of EZ signals are received after the ORG signal is turned ON. Then, it returns to the memorized position if an overrun occurs.
  11) The axis feeds at high speed, reverses after a deceleration stop triggered by the EL signal, and stops when a specified number of EZ signals are received.

- Mechanical input signals

  The following four signals can be input.

  1) +EL: When this signal is turned ON while the axis is feeding in the positive direction, the axis stops immediately (with deceleration). When this signal is ON, no further movement occurs on the axis in the positive direction. (The axis can be rotated in the negative direction.)
  2) -EL: This signal's function is the same as the +EL signal except that it works in the negative direction.
  3) SD: This signal can be used as a deceleration signal or a deceleration stop signal, according to the software setting. When this is used as a deceleration signal, and when this signal is turned ON during a high speed feed operation, this axis will decelerate to the FL speed. If this signal is ON and movement on the axis is started, this axis will run at the FL constant speed. When this signal is used as a deceleration stop signal, and when this signal is turned ON during a high speed feed operation, this axis will decelerate to the FL speed and then stop.
  4) ORG: Input signal for an origin return operation.

  For safety, make sure the +EL and -EL signals stay on from the EL position until the end of each stroke.

  The input logic for these signals can be changed using the ELL terminal.

  The input logic of the SD and ORG signals can be changed using software.

- Digital servomotor I/F

The following three signals can be used as an interface.

1) INP: Inputs positioning complete signal that is output by a servomotor driver.
2) ERC: Outputs deviation counter clear signal to a servomotor driver.
3) ALM: Regardless of the direction of operation, when this signal is ON, this axis stops immediately (deceleration stop). When this signal is ON, no movement can occur on this axis.

The input logic of the INP, ERC, and ALM signals can be changed using software.

The ERC signal is a pulsed output. The pulse length can be set. (12 μsec to 95 msec. A level output is also available.)

- Emergency stop signal ($\overline{\text{EMG}}$) input

When this signal is turned ON, the axe stops immediately. While this signal is ON, no movement is allowed.

3. Specifications

3-1. Basic specifications

Item	Description
General-purpose input/output	General-purpose input/output port. (1 port = 8 bits) Individual bit can be set for input or output.
Communication data length	1 to 64 words / frame (one word = 16 bits)
Data buffer length	64 words
Data communication time	When using three-word communication (written to one register in the G9103A) --- 19.3 µs 63-word communication (written to 21 registers together)--- 91.3 µs
Communication control system	I/O port control: Cyclic communication Data communication: Transient communication.
Package type	80 pins QFP (Mold section: 12 x 12 x 1.4 mm) same as G9003
Power supply	3.0 V to 3.6V
Storage temperature range	-40 to +125˚C
Operating temperature range	-40 to +85˚C

3-2. Communication system specifications

Item		Description
Reference clock	Note 1	40 MHz or 80 MHz
Communication rate	Note 2	2.5 M, 5 M, 10 M, or 20 Mbps
Communication codes		NRZ code
Communication protocol		NPM original
Communication method		Half-duplex communication
Communication I/F	Note 3	RS-485 or pulse transformer
Connection method		Multi-drop connection
Number of local LSIs		64 devices max.

Note 1:When to transfer data with 20 Mbps speed, and if the clock duty can be maintained to ideal "50:50" condition, G9103A can be operated by inputting 40 MHz clock signal.

The ideal condition is that each G9103A is connected to one oscillator. Actually, even this good condition cannot establish 50:50. However, a duty proximate to the ideal one will be established. Even if the ideal duty is broken a little, when signal lines are shorter and/or the number of local LSIs is smaller, Motionnet system can operate without any trouble. (For the details, see the section for the "CLK" terminal.)

When signal lines are longer and/or many local LSIs are connected and it is difficult to warrant the clock duty, you should input an 80 MHz signal.

Which clock rate you use is selected by setting the CKSL terminal. In either clock rate, the maximum speed of 20 Mbps is the same.

Note 2:The communication rate is selected by the SPD0 and SPD1 terminal. It is not necessary to change the input clock rate for changing communication rate.

Note 3:NPM recommends using a pulse transformer.

3-3. Specifications for the axis control section

Item	Description
Positioning control range	-134,217,728 to +134,217,727 (28-bit)
Ramping-down point setting range	0 to 16,777,215 (24-bit)
Number of registers used for setting speeds	Three for each axis (FL, FH, and FA (correction speed))
Speed setting step range	1 to 100,000 (17-bits) Note1
Speed magnification range	Multiply by 0.1 to 66.6 Multiply by 0.1 = 0.1 to 10,000.0 pps Multiply by 1 = 1 to 100,000 pps Multiply by 50 = 50 to 5,000,000 pps
Acceleration /deceleration characteristics	Selectable acceleration/deceleration pattern for both increasing and decreasing speed separately, using Linear and S-curve acceleration/deceleration.
Acceleration rate setting range	1 to 65,535 (16-bit) Ex: 1-> 100,000 pps acceleration time: 80 msec (“1” is set.) to 2621 sec (“65535” is set.)
Deceleration rate setting range	1 to 65,535 (16-bit)
Ramping-down point automatic setting	Automatic setting within the range of (deceleration time) < (acceleration time x 2)
FH correction function (Eliminates triangle pattern driving)	If the feed amount is too small, the LSI has to start decelerating before it has completed the acceleration, and this will create a triangular shaped speed pattern. In order to eliminate this triangular speed pattern, this function automatically reduces the operation speed so that the triangle speed pattern will be avoided.
Manual operation input	Manual pulsar input
Counter	COUNTER1: Command position counter (28-bit) COUNTER2: Mechanical position counter (28-bit) COUNTER3: General-purpose deviation counter (16-bit)
Comparators	28-bits x 3 circuits

Note 1:Values above 100,000 cannot be entered. Even if a value over 100,000 is entered, the register value will only be 100,000.

4. Hardware description

4-1. A list of terminals

No.	Signal name	I/O	Logic	Description	TTL interface
1	VDD			Power supply +3.3 V	
2	$\overline{\text{DN0}}$/$\overline{\text{DNSI}}$	I_U	Negative	Device address bit 0 (Common with the serial input)	Possible
3	$\overline{\text{DN1}}$	I_U	Negative	Device address bit 1	Possible
4	$\overline{\text{DN2}}$	I_U	Negative	Device address bit 2	Possible
5	$\overline{\text{DN3}}$/ROMC	B_U	Negative	Device address bit 3 / Clock output for EEPROM connection	Possible
6	$\overline{\text{DN4}}$/ROMO	B_U	Negative	Device address bit 4 / Data output for EEPROM connection	Possible
7	$\overline{\text{DN5}}$/ROMI	I_U	Negative	Device address bit 5 / Data input for EEPROM connection	Possible
8	VDD			Power supply input +3.3 V	
9	$\overline{\text{GRP0}}$	I_U	Negative	Group setting bit 0 (VDD on the G9003)	Possible
10	$\overline{\text{GRP1}}$	I_U	Negative	Group setting bit 1 (VDD on the G9003)	Possible
11	$\overline{\text{GRP2}}$	I_U	Negative	Group setting bit 2 (VDD on the G9003)	Possible
12	ROME	I_D	Positive	H: EEPROM connection mode (GND on the G9003)	Possible
13	$\overline{\text{DNSO}}$/$\overline{\text{ROMS}}$	O	Negative	Serial output of the next LSI device address / Chip select output for EEPROM connection	Possible
14	DNSM	I_U		Device address setting mode	Possible
15	SOEI	I_D	Positive	Enable serial output	Possible
16	$\overline{\text{SOEL}}$	O	Negative	Enable serial output	Possible
17	SOEH	O	Positive	Enable serial output	Possible
18	SO	O	Positive	Serial output	Possible
19	GND			GND	
20	SI	I	Positive	Serial input	Possible
21	GND			GND	
22	VDD			Power supply input +3.3V	
23	VDD			Power supply input +3.3V	
24	EA	I_U		Encoder A phase signal	Possible
25	EB	I_U		Encode B phase signal	Possible
26	EZ	I_U	Negative#	Encoder Z phase signal	Possible
27	PA	I_U		Manual Pulsar A phase signal	Possible
28	PB	I_U		Manual Pulsar B phase signal	Possible
29	GND			GND	
30	ERC/CDWN	O	Negative#	Request to clear a deviation counter in a driver/current-down	Possible
	OUT	O	Negative#	Pulse train output	Possible
32	DIR	O		Feed direction	Possible
33	VDD			Power supply input +3.3 V	
34	$\overline{\text{CP1}}$	O	Negative	Comparator 1 output	Possible
35	$\overline{\text{CP2}}$	O	Negative	Comparator 2 output	Possible
36	$\overline{\text{CP3}}$	O	Negative	Comparator 3 output	Possible
37	PCS	I_U	Negative#	Start positioning control	Possible
38	LTC	I_U	Negative#	Counter value latch signal	Possible
39	CLR	I_U	Negative#	Counter clear signal	Possible
40	INP	I_U	Negative#	In-position (Positioning complete)	Possible

No.	Signal name	I/O	Logic	Description	TTL interface
41	GND			GND	
42	$\overline{\text{EMG}}$	I_U	Negative	Emergency stop	Possible
43	+EL	I_U	Negative%	(+) end limit	Possible
44	-EL	I_U	Negative%	(-) end limit	Possible
45	SD	I_U	Negative#	Deceleration (Deceleration stop) signal	Possible
46	ORG	I_U	Negative#	Origin position signal	Possible
47	ALM	I_U	Negative#	Alarm signal (Stop request)	Possible
48	VDD			Power supply input +3.3 V	
49	$\overline{\text{STA}}$	B_U	Negative	External start, simultaneous start	Possible
50	$\overline{\text{STP}}$	B_U	Negative	External stop, simultaneous stop	Possible
51	P0	B_U		General-purpose I/O terminal 0	Possible
52	GND			GND	
53	P1	B_U		General-purpose I/O terminal 1	Possible
54	P2	B_U		General-purpose I/O terminal 2	Possible
55	P3	B_U		General-purpose I/O terminal 3	Possible
56	P4/SIFC	B_U		General-purpose I/O terminal 4 / Clock input for CPU connection	Possible
57	P5/$\overline{\text{SIFS}}$	B_U		General-purpose I/O terminal 5 / Selection input for CPU connection	Possible
58	P6/SIFI	B_U		General-purpose I/O terminal 6 / Data input for CPU connection	Possible
59	P7/SIFO	B_U		General-purpose I/O terminal 7 / Data output for CPU connection	Possible
60	VDD			Power supply input +3.3 V	
61	$\overline{\text{BSY}}$/PH1	O	Negative/ Positive	Operation-in-progress signal / Excitation sequence output 1	Possible
62	$\overline{\text{FUP}}$/PH2	O	Negative/ Positive	Acceleration monitor output / Excitation sequence output 2	Possible
63	$\overline{\text{FDW}}$/PH3	O	Negative/ Positive	Deceleration monitor output /Excitation sequence output 3	Possible
64	$\overline{\text{MVC}}$/PH4	O	Negative/ Positive	Constant speed monitor output /Excitation sequence output 4	Possible
65	ELL	I_U		±EL input logic setting	Possible
66	GND			GND	
67	$\overline{\text{MSEL}}$	O	Negative	Goes LOW for a certain interval while this LSI is sending/receiving data.	Possible
68	$\overline{\text{MRER}}$	O	Negative	Goes LOW for a certain interval when an abnormal communication has been received.	Possible
69	$\overline{\text{TOUT}}$	O	Negative	Watchdog timer output	Possible
70	BRK	I_D	Positive	Break signal	Possible
71	TUD	I_U		Select operation method for outputting watchdog timer signal	Possible
72	TMD	I_U		Watchdog timer setting	Possible
73	VDD			Power supply input +3.3 V	
74	CLK	I		Reference clock	Possible
75	GND			GND	
76	CKSL	I_U		Clock rate selection	Possible
77	SPD0	I_U		Communication rate setting 0	Possible
78	SPD1	I_U		Communication rate setting 1	Possible
79	GND			GND	
80	$\overline{\text{RST}}$	I_U	Negative	Reset	Possible

Note 1: "I" in the I/O column expresses input, "O" as output, and "B" as both directions.

Note 2: All signal terminals have 5V interface and can be connected to 3.3 V-CMOS, TTL, and LVTTL devices. However, even if the output terminals are pulled up to 5V, more than 3.3V is not realized.

Note 3: 5V interface inputs are not equipped with an over voltage prevention diode for the 3.3 V lines. If over voltage may be applied due to a reflection, ringing, or to inductive noise, we recommend inserting a diode to protect against over voltage.

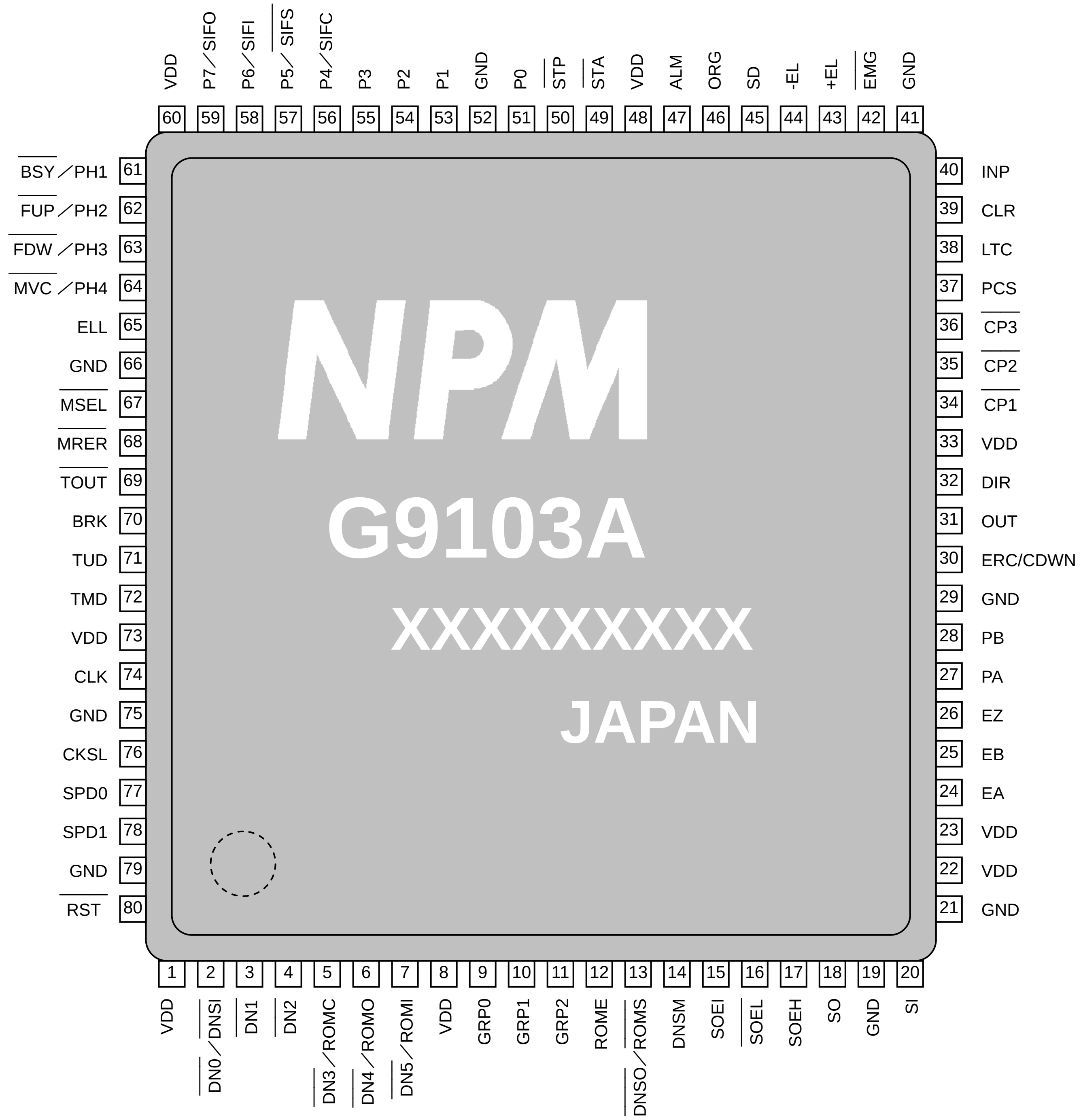
Note 4: "I_U" and "B_U" in the table indicate terminals with a pull up resistor to prevent floating. "I_D" indicates terminals with a pull down resistor to prevent floating.

Input terminals that are not used and which have internal pull up/ pull down resistors can be left open. However, we recommend pulling these unused terminals up to 3.3 V (5k to 10k ohm) externally. Input terminals can be connected to 3.3V or GND. However, terminals for both directions should be connected using a resistor.

Note 5: Leave the unused output terminals open.

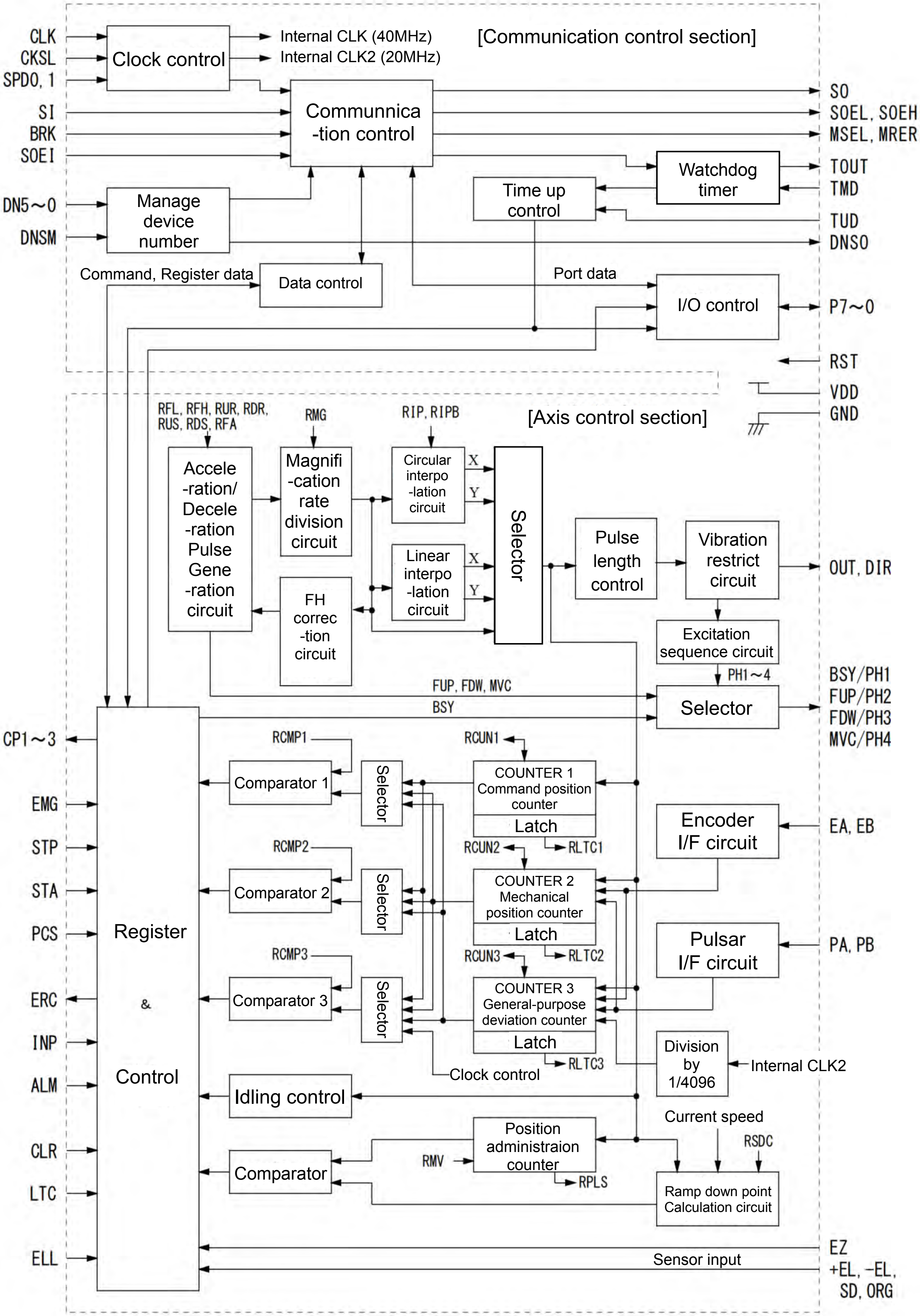
Note 6: "Negative" means negative logic and "Positive" means positive logic. In addition, a "#" means that the terminal's logic can be changed by software. A "%" means that the terminal's logic can be changed by terminal setting. The logic shown in the table is the default condition. The DIR terminal logic shown is when it is used in Two-pulse mode.

4-2. Terminal allocation diagram



Note: As you can see in the figure above, pin number 1 is at the left lower of the LSI model name marked on the LSI.

4-3. Entire block diagram



4-4. Functions of terminals (The number in the parentheses [ ] is the terminal number.)

4-4-1. CLK [74]

This is an input terminal of the reference clock. By setting the CKSL terminal, either of the following clock rate signals can be connected.

CKSL = L: 40 MHz
CKSL = H: 80 MHz

By selecting either of these clock rates, the serial communication rate does not change. This clock rate selection affects communication precision.
For a small-scale serial communication and communication rate below 10 Mbps, use of the center LSI with 40 MHz does not give any restriction.
With 20 Mbps communication rate; however, a longer communication line or a large number of connected local LSIs may deteriorate communication precision due to collapse of signals on the circuit. This deterioration of communication quality can be corrected inside the LSI if the deterioration level is not much.
In order to improve correction precision; however, evenness of the clock duty is required. In other words, if the duty is ideal (50:50), the capacity to correct collapse of the signals in the communication lines can be improved. On the contrary, if the duty is not ideal, the LSI cannot cope with collapses of the communication line.
As a result, if the duty is close to ideal, the LSI can be used with 40 MHz. When connecting more than one LSI to one oscillator, the duty may not be ideal. In this case, use 80 MHz. This LSI divides the clock rate inside and creates 40 MHz frequency.
If you do not want to use 80 MHz clock rate, you may prepare a separate 40 MHz oscillator for each this LSI.
We recommend that 3.3 V signal is input because duty rate may get worse due to threshold voltage, even though 5V signal can be input.

4-4-2. $\overline{\text{RST}}$ [80]

This is an input terminal for a reset signal.
By input L level signal, the internal circuit of G9103A is reset. The reference clock is used to reset it in the LSI. Therefore, please do not disconnect the clock even if resetting. Longer than 10 clock cycles is required during resetting.

4-4-3. CKSL [76]

This is used to select clock rate.
L: Connect 40 MHz clock rate to the CLK terminal.
H: Connect 80 MHz clock rate to the CLK terminal.
Select this when the duty of the 40 MHz clock collapses a lot.

4-4-4. ROME [12]

H level signal is input to this terminal when serial EEPROM with unit ID is connected. On the G9003, the terminal with this pin number is the GND terminal. For the detail, see 8-18, “Unit ID control function”.

4-4-5. $\overline{\text{DN0}}$/ $\overline{\text{DNSI}}$, $\overline{\text{DN1}}$, $\overline{\text{DN2}}$, $\overline{\text{DN3}}$/ROMC, $\overline{\text{DN4}}$/ROMO $\overline{\text{DN5}}$/ROMI [2 to 7]

Input terminals for setting device address.
Since these terminals use negative logic, setting all the terminals to LOW level calls up device address "3F(h)."
There are two methods for entering a device address. Select the input method using the DNSM terminal.
$\overline{\text{DN3}}$ to are also used as terminals for EEPROM connection. For the detail, see 8-18. “Unit ID control function”.

4-4-6. DNSM [14]

This terminal is to select the input method of device addresses.

1) When the DNSM = H
Specify an address from 00(h) to 3F(h) using the $\overline{\text{DN0}}$ to $\overline{\text{DN5}}$ terminals.

2) When the DNSM = L
When the $\overline{\text{DNSO}}$ signal that is output by another local device is input to the $\overline{\text{DN0}}$ / $\overline{\text{DNSI}}$ terminal on this LSI, this LSI has an address equal to another LSI's address plus one.
Terminals $\overline{\text{DN1}}$ to $\overline{\text{DN5}}$ should be pulled up or pulled down.
$\overline{\text{DN0}}$ / $\overline{\text{DNSI}}$ terminal is input serialized device address value repeatedly. When the same values are input twice in a row, this value is recognized as a correct device address.

4-4-7. $\overline{\text{DNSO}}$ / $\overline{\text{ROMS}}$ [13]

The numeric equivalent to the device address on this LSI + 1 is output to the next LSI and the LSI outputs the numeric equivalent to the device address +1 to the next LSI continuously in order.
Connect this output to another local LSI's $\overline{\text{DN0}}$ / $\overline{\text{DNSI}}$ terminal (make all the other DNSM terminals of that local LSI LOW), so that another LSI can get the continuous number address.
Please note that the next address after "3F(h)" (DN(5:0) = " LLLLLL") is "00(h)." If you set continuous number address by the $\overline{\text{DNSO}}$ signal, it needs at least 50 μsec to determine next LSI's address.
When the ROME terminal (12) is H level, it is used as the terminals for EEPROM connection. Therefore, when serial setting of multiple local LSI's address, please make G9103A connected with EEPROM the last LSI of the communication line.

4-4-8. $\overline{\text{GRP0}}$ to $\overline{\text{GRP2}}$ [9 to 11]

This terminal is to specify the group number used for broadcast communication. Because of negative logic input, "7" is set as the group number when all terminals are set to low level. The group number can be changed by software (RENV2.GN0 to GN2)

4-4-9. SPD0, SPD1 [77, 78]

These terminals are to set the communication rate.
All of the LSIs on the same communication line must be set to be same speed.

SPD1	SPD0	Communication rate
L	L	2.5 Mbps
L	H	5 Mbps
H	L	10 Mbps
H	H	20 Mbps

4-4-10. TUD [71]

A watchdog timer is included on the LSI to assist in control of the communication status (see the "TMD" terminal section).
When the data transmission interval from the center LSI to this LSI exceeds the set time, the watchdog timer times out.
This terminal is used to set output conditions when the watchdog timer times out.
When TUD = HIGH --- The LSI keeps its current status.
When the TUD = LOW --- The LSI resets I/O port output and immediately stops pulse output (stop operation).

4-4-11. TMD [72]

Specify operation time for the watchdog timer.
The watchdog timer is used to control the communication status.
When the interval of data sent from the center LSI is longer than the specified interval, the watchdog timer times out (the timer starts its count at the end of each data packet received from the center LSI). The time out may occur because of a problem on the communication circuit, such as disconnection, or simply because the center LSI has stopped communicating.
The time used by the watchdog timer varies with communication rate selected.

TMD terminal	Watchdog timer setting			
	20 Mbps	10 Mbps	5 Mbps	2.5 Mbps
L	5 ms	10 ms	20 ms	40 ms
H	20 ms	40 ms	80 ms	160 ms

4-4-12. $\overline{\text{TOUT}}$ [69]

When the watchdog timer has timed out, this terminal goes LOW.

4-4-13. SO [18]

Serial output signal for communication. (Positive logic, tri-state output)

4-4-14. SOEH [17], $\overline{\text{SOEL}}$ [16]

Output enable signal for communication.
The difference between SOEH and $\overline{\text{SOEL}}$ is that the logic is inverted.
When sending, SOEH = HIGH and $\overline{\text{SOEL}}$ = LOW.

4-4-15. SOEI [15]

When using more than one G9103A, connect SOEH output of the other local LSIs to this terminal.
By logical OR operation with the output enable signal from this LSI, the LSI outputs an enable signal to SOEH and $\overline{\text{SOEL}}$. When connected like this, make the number of LSIs less than 4 because that OR processing time of each LSI is accumulated.

4-4-16. SI [20]

Serial input signal for communication. (Positive logic)

4-4-17. $\overline{\text{MRER}}$ [68]

This terminal is monitor output to check communication quality.
When the G9103A receives an error frame such as a CRC error, this terminal goes LOW for exactly 3.2 μsec.
By timing this interval using a counter, you can check the quality of the communication.

4-4-18. $\overline{\text{MSEL}}$ [67]

Communication status monitor output.
When the G9103A receives a frame intended for this LSI and everything is normal (when $\overline{\text{MRER}}$ is H), this terminal goes LOW for exactly 3.2μsec. This signal cycle is one cycle of the cyclic communication.

4-4-19. BRK [70]

By providing HIGH pulses, the G9103A waits for a break frame.
When the G9103A receives a request for sending a break frame from the center LSI (G9001A), it immediately sends a break frame.
The break frame is 60 bits long.
The BRK input pulse needs at least 3200 μsec long. (positive logic).

4-4-20. P0 to P3, P4/SIFC, P5/$\overline{\text{SIFS}}$ P6/SIFI, P7/SIFO [51, 53 to 59]

Using software, these terminals can be general-purpose input or output terminals (RENV2.P0M to P7M). These terminals have built-in pull up resistors to prevent floating. When these terminals are not used, they can be left open. However, if you want to improve noise resistance of the LSI, pull them up (5 to 10 K-ohms). Using software, P4 to P7 can be used as serial communication terminals to connect to the CPU. For the detail, see 8-19. “CPU connection function”.

4-4-21. $\overline{\text{STA}}$ [49], $\overline{\text{STP}}$ [50]

If you want to start multiple LSIs simultaneously, connect the $\overline{\text{STA}}$ terminals of all the LSIs together.
If you want to stop multiple LSIs simultaneously when an error occurs, connect the $\overline{\text{STP}}$ terminals of all the LSIs together.
These terminals have built in pull up resistors to prevent floating. When these terminals are not used, they can be left open. However, if you want to improve the noise resistance, pull them up (5 to 10 K-ohms). These terminals can be used to start/stop operation by an external signal.

4-4-22. $\overline{\text{EMG}}$ [42]

This is the emergency stop input terminal.
While this terminal is LOW, the G9103A prohibits operation. If this terminal goes LOW while the motor is operating, the motor will stop immediately.
This terminal has a built-in pull up resistor to prevent floating. When not used, it can be left open. However, if you want to improve the noise resistance, pull it up (5 to 10 K-ohms), or connect it to VDD.

4-4-23. ELL [65]

This terminal is to set the input logic of the +EL and -EL signals. When this terminal is LOW, the respective signal is set for positive logic input.

4-4-24. +EL [43], -EL [44]

Provide the stroke end signals to these terminals. Their input logic can be changed using the ELL terminals. When this signal (for the feed direction) turns ON, the motor stops immediately, or decelerates and stops, depending on the conditions.
These terminals have built in pull up resistors to prevent floating. When not used, they can be left open. If you want to improve noise resistance, pull them up (5 to 10 K-ohms) to the VDD or connect them to the VDD

4-4-25. SD [45]

This terminal is to input the deceleration signal (deceleration and stop signal). Software can be used to change the input logic of this terminal (RENV1.SDL).
This input has a latch function.
This terminal has a built in pull up resistor to prevent floating. When this terminal is not used, it can be left open. However, if you want to improve noise resistance of the chip, pull it up (5 to 10 K-ohms) to the VDD, or connect it to the VDD.

4-4-26. ORG [46]

This terminal is to input for an origin return signal. Software can be used to change the input logic of this terminal (RENV1.ORGL).
This is used in origin return operation. (Edge detection)
This terminal has a built-in pull up resistor to prevent floating. When this terminal is not used, it can be left open. However, if you want to improve noise resistance, pull it up (5 to 10 K-ohms) to the VDD, or connect it to the VDD.

4-4-27. ALM [47]

This terminal is to input an alarm signal. Software can be used to change the input logic of this terminal (RENV1.ALML).
When this signal turns ON, the motor stops immediately, or decelerates and stops, depending on the conditions.
This terminal has a built-in pull up resistor to prevent floating. When this terminal is not used, it can be left open. However, if you want to improve noise resistance, pull it up (5 to 10 K-ohms) to the VDD, or connect it to the VDD.

4-4-28. OUT [31], DIR [32]

While the G9103A is in the common pulse mode, it sends feed pulses from the OUT terminal, and supplies a direction signal from the DIR terminal.
While the G9103A is in Two-pulse mode, it outputs positive direction feed pulses from the OUT terminal, and negative direction feed pulses from the DIR terminal.

4-4-29. PA [27], PB [28]

These terminals are to operate the motor from external pulses, such as a manual pulsar.
90 degree phase difference signals or Two-pulse signal (Up pulse and Down pulse) can be supplied to these terminals. The 90 degree phase difference signals can be multiplied by 2 or by 4.
These terminals have built-in pull up resistors to prevent floating. When these terminals are not used, they can be left open. If you want to improve noise resistance, pull them up (5 to 10 K-ohms) to the VDD or connect them to the VDD

4-4-30. EA [24], EB [25], EZ [26]

These terminals are to control the current position using an encoder.
90 degree phase difference signals or Two-pulse (UP pulse and DOWN pulse) can be input on these terminals. The 90 degree phase difference signals can be multiplied by 2 or by 4.
The EZ input is used for origin return operations. Software can be used to change the input logic of these terminals. These terminals have built-in pull up resistors to prevent floating. When these terminals are not used, they can be left open. If you want to improve noise resistance of the chip, pull them up (5 to 10 K-ohms) or connect them to the VDD.

4-4-31. PCS [37]

The G9103A can start positioning control when the signal is input to this terminal (Override 2 of the target position).
Software can be used to change the input logic of this terminal (RENV1.PCSL).
This terminal has a built in pull up resistor to prevent floating. When this terminal is not used, it can be left open. However, if you want to improve noise resistance, pull it up (5 to 10 K-ohms) to the VDD or connect it to the VDD.

4-4-32. INP [40]

Input an In position (positioning complete) signal from a servo driver.
The output of an interrupt request signal can be delayed until this signal is input from the center LSI.
Software can be used to change the input logic of this terminal.
This terminal has a built-in pull up resistor to prevent floating. When this terminal is not used, it can be left open. However, if you want to improve noise resistance, pull it up (5 to 10 K-ohms) to the VDD or connect it to the VDD.

4-4-33. CLR [39]

This terminal is to reset the specified counter (COUNTER 1 to 3) by inputting a signal. (It can reset more than one counter).
Software can be used to change the input logic of this terminal (RENV1.CLR0 to 1).
This terminal has a built-in pull up resistor to prevent floating. When this terminal is not used, it can be left open. However, if you want to improve noise resistance, pull it up (5 to 10 K-ohms) to the VDD or connect it to the VDD.

4-4-34. LTC [38]

This terminal is to latch the specified counter (COUNTER 1 to 3) by inputting a signal. (It can latch more than one counter).
Software can be used to change the input logic of this terminal (RENV1.LTCL).
This terminal has a built-in pull up resistor to prevent floating. When this terminal is not used, it can be left open. However, if you want to improve noise resistance, pull it up (5 to 10 K-ohms) to the VDD or connect it to the VDD.

4-4-35. ERC / CDWN [30]

This terminal is to output a one-shot pulse to clear a deviation counter for a servo driver.
This terminal is also to output terminal of a current down signal at the control of stepper motor.
The output logic and pulse length can be set using software (RENV1.ERCL, RENV1.EPW).
If this terminal is not used, leave it open.

4-4-36. $\overline{\mathrm{BSY}}$ / PH1 [61]

This terminal doubles as $\overline{\mathrm{BSY}}$ and PH1 output. The functions are selected by the RMD register (RMD.MPH).
When the $\overline{\mathrm{BSY}}$ is selected (RMD.MPH=0), the G9103A outputs LOW level from this terminal while the motor is operating.
When the PH1 is selected (RMD.MPH=1), the G9103A outputs excitation sequence for a 2-phase stepper motor.
If this terminal is not used, leave it open.

4-4-37. $\overline{\mathrm{FUP}}$ / PH2 [62]

This terminal doubles as $\overline{\mathrm{FUP}}$ and PH2 output. The functions are selected by the RMD register (RMD.MPH).
When the $\overline{\mathrm{FUP}}$ is selected (RMD.MPH=0), the G9103A outputs a LOW level signal from this terminal while the motor is accelerating.
When the PH2 is selected (RMD.MPH=1), the G9103A outputs an excitation sequence for a 2-phase stepper motor.
If this terminal is not used, leave it open.

4-4-38. $\overline{\mathrm{FDW}}$ / PH3 [63]

This terminal doubles as $\overline{\mathrm{FDW}}$ and PH3 output. The functions are selected by the RMD register (RMD.MPH).
When the $\overline{\mathrm{FDW}}$ is selected, the G9103A outputs a LOW level signal from this terminal while the motor is decelerating.
When the PH3 is selected, the G9103A outputs an excitation sequence for a 2-phase stepper motor.
Select FDW/PH3 using the RMD register.
If this terminal is not used, leave it open.

4-4-39. $\overline{\mathrm{MVC}}$ / PH4 [64]

This terminal doubles as $\overline{\mathrm{MVC}}$ and PH4 output. The functions are selected by the RMD register (RMD.MPH).
When the $\overline{\mathrm{MVC}}$ is selected (RMD.MPH=0), the G9103A outputs a LOW level signal from this terminal while the motor is operating at a constant speed.
When the PH4 is selected (RMD.MPH=1), the G9103A outputs an excitation sequence for a 2-phase stepper motor.
If this terminal is not used, leave it open.

4-4-40. $\overline{\mathrm{CP1}}$ [34]

When the conditions for Comparator 1 are met, the G9103A outputs a LOW level signal from this terminal.
If this terminal is not used, leave it open.

4-4-41. $\overline{\mathrm{CP2}}$ [35]

When the conditions for Comparator 2 are met, the G9103A outputs a LOW level signal from this terminal. If this terminal is not used, leave it open.

4-4-42. $\overline{\mathrm{CP3}}$ [36]

When the conditions for Comparator 3 are met, the G9103A outputs a LOW level signal from this terminal. If this terminal is not used, leave it open.

4-4-43. VDD, GND

Input terminal for power supply. Input 3.3V±10% to the VDD. Please make sure to use all VDD and GND terminals.

5. Description of the software

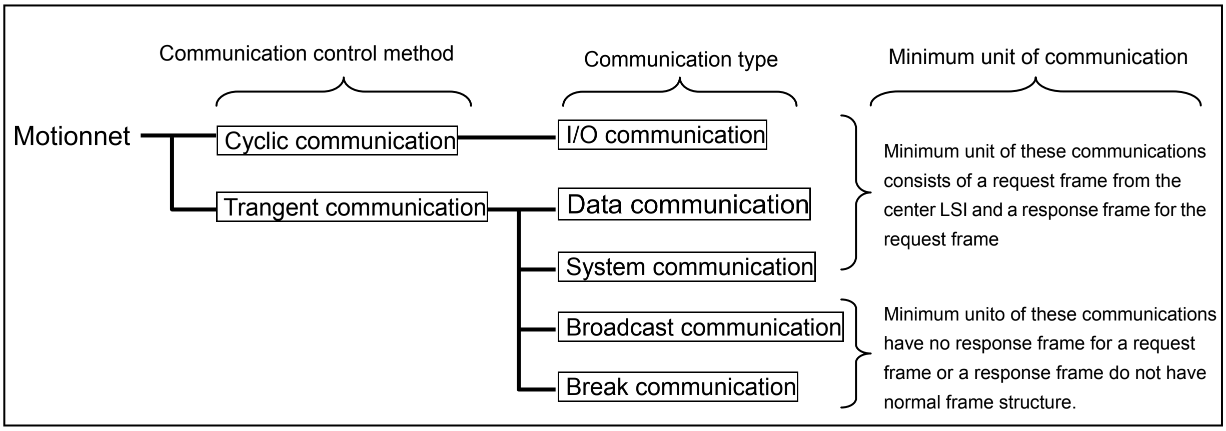
5-1. Outline of control

5-1-1. Communication control

- The center LSI (G9001A) controls all communication.
- One set of communication consists of a communication from the center LSI to the local LSIs (request frame), and a communication from the local LSIs back to the center LSI (response frame).
- The response from the local LSIs may include I/O information and data.
- The G9103A is a local LSI for data control.

5-1-2. Communication type

System communications, I/O communications, data communications and broadcast communications are available.


Communication control method
Communication type
Minimum unit of communication
Motionnet
Cyclic communication
Trangent communication
I/O communication
Data communication
System communication
Broadcast communication
Break communication
Minimum unit of these communications consists of a request frame from the center LSI and a response frame for the request frame
Minimum unito of these communications have no response frame for a request frame or a response frame do not have normal frame structure.

G9103A controls I/O communication, data communication, system communication, broadcast communication and break communication.

5-1-2-1. Cyclic communication

I/O communication is to send and receive port and status information of local LSI.
I/O communication starts with the local LSI that has the lowest number device address and proceeds through all the LSIs that are present. When the communication with the LSI that has the highest number device address is complete, the center LSI starts again to communicate with the local LSI that has the lowest number device address. Therefore, it is also called "cyclic communication".
By writing an I/O communication start command to the center LSI, the center LSI communicates only with devices whose "device in use" bit in "device information" area is set to 1.
This communication continues until a cyclic communication stop command is written.
This G9103A checks status and inputs and outputs information of general-purpose I/O terminals using cyclic communications.

5-1-2-2. Data communication

In data communication, the center LSI communicates with other data control LSIs, such as the G9103A, etc.
Normally, the center LSI sends I/o information in cyclic communications. A data communication command from a CPU allows you to perform data communications by interrupting the cyclic communications.
After writing data to the data transmission FIFO of the center LSI, write a send data command. The center LSI will interrupt by the data communication when the current cyclic communication is complete.
After a local LSI has received data, it will ignore any further data received until it has read out all of the data received, and it will not send any response to the center LSI while reading the data. The center LSI recognizes that no response error has occurred in this case and retries the communication (maximum three times).

5-1-2-3. System communication

With the system communication, the center LSI automatically confirms the connection status, device type, and I/O port settings of each local LSI.
By starting the system communication, the center LSI performs polling communication to all of the local LSIs (device address 0 to 63) one by one, and refreshes the "device information" area according to the response from the local LSIs.

5-1-2-4. Broadcast communications

The center LSI (G9001A) sends information to multiple data LSIs simultaneously. Broadcast communication frame consists of group number (Group numbers are set by terminals $\overline{\text{GRP0}}$ to $\overline{\text{GRP2}}$ of data LSI and can be changed by software) and broadcast commands. Only the data LSIs that are specified by group number of communication frame perform a broadcast command received. When group number (ggg in the following table) of communication frame is “000”, all groups are the objects of the command.

Broadcast commands that G9103A can perform (ggg=group number)

Broadcast command	Description
0010 0ggg 0000 0001	Start（CMSTA command for multiple axes）
0010 0ggg 0000 0010	Stop（CMSTP command for multiple axes）
0010 0ggg 0000 0011	Emergency stop（CMEMG command for multiple axes）
0010 0ggg 0000 0100	Reset local LSI（SRST command for multiple axes）
0010 0ggg 0000 0101	Latch counter value（LTCH command for multiple axes）
0010 0ggg 0000 0110	Stop immediately（STOP command for multiple axes）
0010 0ggg 0000 0111	Decelerate and stop（SDSTP command for multiple axes）
0010 0ggg 0000 1000	Change to FL speed immediately（FCHGL command for multiple axes）
0010 0ggg 0000 1001	Change to FH speed immediately（FCHGH command for multiple axes）
0010 0ggg 0000 1010	Decelerate to FL speed（FSCHL command for multiple axes）
0010 0ggg 0000 1011	Accelerate to FH speed（FSCHH command for multiple axes）
0010 0ggg 0000 1100	Copy a pre-register for operation to a register (speed change, etc.)（PRESHF command for multiple axes）

Note: Broadcast command is a command that is written into G9001A.

[How to set group numbers]

To set group numbers, there are two following ways:

1. Set group numbers to the $\overline{\text{GRP0}}$ to $\overline{\text{GRP2}}$ input terminals by negative logic. You can confirm the setting value by main status (MSTS.SGP0 to 2).

2. Set group numbers by writing to the registers (RENV2.GN0 to 2). You can confirm the setting value by reading registers.

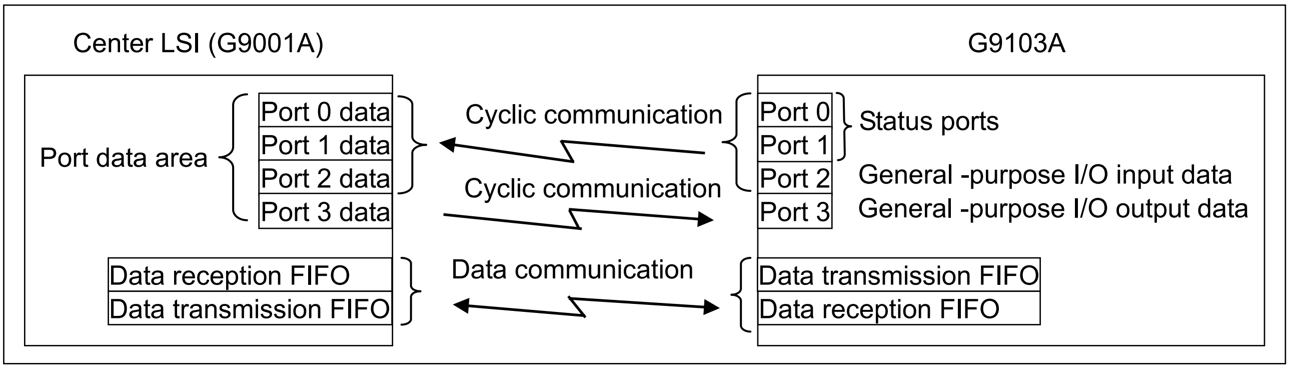
Only when RENV2.GN0 to 2=000, the setting value of the above 1 is used. Other than the case of 000, the setting value of the above 2 is used.

5-1-2-5. Break communications

Motionnet performs communication with the local LSIs that are registered in G9001A before starting communication, by system communication or CPU software. Therefore, even if local LSIs are turned on after the start of communication, the local LSIs cannot perform communication. The local LSIs are needed to be informed to the center LSI and registered additionally to the center LSI.
Request communication for additional registration is called as break communication. Normally, G9001A sends a request frame for sending a break frame every approximately 250ms. When you input a break signal to the BRK terminal of the G9103A using such as switches, the G9103A returns a break frame at the receipt of the next request frame for sending a break frame.
When G9001A receives a break frame, the status (STSW.BRKF) becomes 1 and G9001A outputs an interrupt request signal ($\overline{\text{INT}}$) to the CPU
Then, G9001A searches the source of request using CPU software and register it as the communication target additionally.

Notes: G9001A sends a request frame for sending a break frame at specified time intervals in default. However, this process disturbs cycle of cyclic communication. If you set RENV0.BK0F=1 on the G9001A, the automatic sending a request frame for sending a break frame stops and cycle of cyclic communication can be made constant. However, automatic break function becomes unable to be used. In this case, CPU can issue "a break communication command (0610h)" at a given point in time.

[Conceptual communication diagram]
G9103A sends and receives information of port 0 to 3 by I/O communication (cyclic communication) and access to registers and write commands by data communication.



Note: Response frame of data communication (from G9103A to G9001A) contains information of port 0 to 3 so that status change by data communication can be transferred to the center LSI immediately.

G9103A can perform data communication up to 64 words at once. Therefore, it can make total communication time shorter. The data to set registers consists of a 1-word command and a 2-word command. Therefore, 21 registers can be set at once at a maximum.
It can also contain control commands and register read commands as well as register settings. However, when you send multiple register-read commands together at once, please avoid that response information (from G9103A to G9001A) exceeds 64 words.

- Example of setting 1
  Writing into one register of G9103A (setting on the G9001A side)
    RMV (Register for setting feed amount) ←"01234567(h)"

  When using a 16-bit CPU
  1) First, write an RMV write command (0090(h)) to the transmitting FIFO (006(h)).
  2) Next, write the lower 16 bits data (4567(h)) to be sent to the RMV register into the transmission FIFO (006(h)).
  3) Then, write the upper 16 bits data (0123(h)) to be sent to the RMV register into the transmission FIFO (006(h)).
  4) Finally, write the data communication command (4000(h)) into the command area (000(h)).

	Details of the data transmission FIFO	Description
1st word	0090(h)	RMV write command
2nd word	4567(h)	RMV lower data
3rd word	0123(h)	RMV upper data

When writing into only one register, you can omit the third word when the register length is less than 16 bits or when the upper data is "0000(h)".

- Example of setting 2.

Writing into multiple registers of G9103A together at once.（Setting on the G9001A side）

PRMV (Pre-register for setting feed amount) ←“01234567(h)”
PRFL (Pre-register for setting FL speed) ←“00000001(h)”
PRFH (Pre-register for setting FH speed) ←“00001000(h)”
PPUR (Pre-register for setting acceleration rate) ←“00000010(h)”
PRMG (Pre-register for setting speed magnification) ←“000000C7(h)”
PPMD (Pre-register for setting operation mode) ←“00000041(h)”
High-speed start command (STAUD) ←“00000053(h)”

Write the data communication command (4000(h)) into the command area (000(h)).

	Details of the data transmission FIFO	Description
1st word	00B0(h)	PRMV write command
2nd word	4567(h)	PRMV lower data
3rd word	0123(h)	PRMV upper data
4th word	00B1(h)	PRFL write command
5th word	0001(h)	PRFL lower data
6th word	0000(h)	PRFL upper data
7th word	00B2(h)	PRFH write command
8th word	1000(h)	PRFH lower data
9th word	0000(h)	PRFH upper data
10th word	00B3(h)	PRUR write command
11th word	0010(h)	PRUR lower data
12th word	0000(h)	PRUR upper data
13th word	00B5(h)	PRMG write command
14th word	00C7(h)	PRMG lower data
15th word	0000(h)	PRMG upper data
16th word	00B7(h)	PRMD write command
17th word	0041(h)	PRMD lower data
18th word	0000(h)	PRMD upper data
19th word	0053(h)	STAUD start command

- Example of setting 3

Latch the contents of COUNTER 1 to 3 and read the values（setting on the G9001A side）

Latch command（LTCH） ←“0029(h)”
RLTC1 (COUNTER 1 Latch register) ←“00ED(h)”
RLTC2 (COUNTER 2 Latch register)Read ←“00EE(h)”
RLTC3 (COUNTER 3 Latch register)Read ←“00EF(h)”

Write the data communication command (4000(h)) into the command area (000(h)).

	Details of the data transmission FIFO	Description
1st word	0029(h)	LTCH command
2nd word	00Ed(h)	RRLTC1 command
3rd word	00EE(h)	RRLTC2 command
4th word	00Ef(h)	RRLTC3 command

When communication completes, the following data is stored in the data reception FIFO.

(In the case of RLTC1=01234567h, RLTC2=01234566h, RLTC3=00000001h)

	Data of the data reception FIFO	Description
1st word	00ED(h)	RRLTC1 response command
2nd word	4567(h)	RLTC1 upper data
3rd word	0123(h)	RLTC1 lower data
4th word	00B1(h)	RRLTC2 response command
5th word	4566(h)	RLTC2 lower data
6th word	0123(h)	RLTC2 upper data
7th word	00B2(h)	RRLTC3 response command
8th word	0001(h)	RLTC3 lower data
9th word	0000(h)	RLTC3 upper data

5-2. Functional settings for the G9103A

5-2-1. I/O port

G9103A has four I/O ports. The highest port, Port 3, is used to output. Ports 0, 1, and 2 are used to input. As shown in the figure below, they are arranged from the highest to the lowest port: for a general-purpose I/O terminal output data, for a general-purpose I/O terminal input data, for main status upper byte, and for main status lower byte.
The general-purpose I/O terminals are selected between input or output using register RENV2. Therefore, the settings for the general-purpose I/O output data will be effective only when the general-purpose I/O terminals are set up as outputs.
The general-purpose I/O output data terminals can be reset by setting of the TUD terminal when the watchdog timer times out
The output status can be checked by reading the general-purpose I/O input data.

Port 3	Port 2	Port 1	Port 0
General-purpose I/O output data [IOPOB]	General-purpose I/O input data [IOPIB]	Main status (upper byte) [MSTSB1]	Main status (lower byte) [MSTSB0]

5-2-1-1. Main status (MSTS)

The bit definition varies depending on the setting status of RSYN.SYNC and RSYN.SYNE.

[In the case of RSYN.SYNC=0 and RSYN.SYNE=0 (default)]

MSTSB1								MSTSB0							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	SGP2	SGP1	SGP0	0	0	0	SBSY	SPDF	SPRF	0	0	SEVT	SERR	SEND	SINT

[In the case of RSYN.SYNC=1]

MSTSB1								MSTSB0							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SYN6	SYN5	SYN4	SYN3	SYN2	SYN1	SYN0	SBSY	SPDF	SPRF			SEVT	SERR	SEND	SINT

[In the case of RSYN.SYNE=1]

MSTSB1								MSTSB0							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							SBSY	SPDF	SPRF	SPS1	SPS0	SEVT	SERR	SEND	SINT

Bit	Bit name	Description
0	SINT	This bit is set to 1 when an interrupt request occurs. (When any of bits 1, 2, or 3 is 1.)
1	SEND	When an operation stops, an interrupt request occurs and this bit is set to 1. This bit is returned to 0 by the Interrupt Reset command (0008(h)).
2	SERR	This bit is set to 1 when an error interrupt occurs. It is set back to 0 by reading the REST or writing into the REST.
3	SEVT	This bit is set to 1 when an event interrupt occurs. It is set back to 0 by writing the RIST.
4 to 5	SPS0 to 1	For simultaneous stop processing. Please ignore these when reading status.
6	SPRF	This bit is set to 1 when the pre-register for next operation is full.
7	SPDF	This bit is set to 1 when the pre-register for comparator 3 is full.
8	SBSY	This bit is set to 1 when the LSI starts to output pulses. It is set back to 0 by stopping operation. (=BSY)
12 to 14	SGP0 to 2	The status of the terminals (GRP0 to 2) to specify group number for broadcast communication. (1=L level) When RENV2.GN0 to GN2=000, these bits are set same as GRP0 to 2. When other than 000, these bits are set same as GN0 to GN2.
9 to 15	SYN0 to 6	For clock synchronous processing. Please ignore these when reading status.

5-2-1-2. General I/O terminal input data (IOPIB)

IOPIB

7	6	5	4	3	2	1	0
IP7	IP6	IP5	IP4	IP3	IP2	IP1	IP0

Bit	Bit name	Description
0 to 7	IP 0 to 7	These bits are to read the status of terminals P0 to P7. (0: LOW, 1: HIGH)

Note: In case of RENV2.SIFM=1, IP4 to 7 are set to 0.
In case of RMD.MIOR=1, bits set as output terminals are set to 0.

5-2-1-3. General I/O terminal output data (IOPOB)

IOPOB

7	6	5	4	3	2	1	0
OP7	OP6	OP5	OP4	OP3	OP2	OP1	OP0

Bit #.	Bit name	Description
0 to 7	OP 0 to 7	These bits are to set the output status of terminals P0 to P7. (0: LOW, 1: HIGH)

Note: When RENV2.SIFM=1, the setting for IP4 to 7 is disabled.

5-3. Command (Operation commands & Control commands)

The G9103A can control axes through data communications from the center LSI by using the following commands. Two command types are available, "Commands without data" and "Commands with data (register data)".
The buffer memory size for sending and receiving is 64 words. Two communication types are available, “communication with one command” that is to communicate one command at once and “communication with multiple commands” that is to communicate multiple commands together at once. Please pay attention to the followings.

1. When “commands with data” is sent in the “communication with one command”, the upper word is recognized as 0000(h) if the 3rd word is omitted. If both the 2nd word and 3rd word are omitted, both the lower data and upper data are recognized as “0000(h)”.
2. If “commands without data” is sent with data by mistakes, the 2nd word is recognized as the second command and malfunction occurs.
3. When “commands with data” is sent in the “communication with multiple commands”, please send commands followed by 2-word data continuously. If the number of words is different, data is recognized as command.

The buffer size for sending and receiving is both 64 words. If larger data than this is sent, the processing contents may not be ensured.

Note : Please do not perform one-word communication that sends only one command without data.
Please add a NOP command to the one word and perform at least two-word communication.
If you perform one-word communication shortly after cyclic communication error occurs, G9103A may malfunction.

5-3-1. Operation commands

5-3-1-1. Start command

1) Start command

To start operation, write one of these commands while the motor is stopping.

Symbol	1st word	2nd word	3rd word	Description	Response frame
STAFL	0050(h)			FL constant speed start	Response only
STAFH	0051(h)			FH constant speed start	Response only
STAD	0052(h)			High-speed start 1 (FH constant speed ->Deceleration)	Response only
STAUD	0053(h)			High-speed start 2 (Acceleration -> FH constant speed -> Deceleration)	Response only

2) Residual amount start command

If a positioning operation is halted without completing and uses these commands to drive the motor, operation of the residual number of pulses in the positioning counter is executed.

Symbol	1st word	2nd word	3rd word	Description	Response frame
CNTFL	0054(h)			Residual amount by FL constant speed start	Response only
CNTFH	0055(h)			Residual amount by FH constant speed start	Response only
CNTD	0056(h)			Residual amount by high speed start1 (FH constant speed ->Deceleration)	Response only
CNTUD	0057(h)			Residual amount by high speed start 2 (Acceleration -> FH constant speed -> Deceleration)	Response only

3) Start command with a feed amount

Write a feed amount and a start command while the motor is stopping. The feed amount is useful for positioning operations.

Symbol	1st word	2nd word	3rd word	Description	Response frame
RMSTFL	0058(h)	Lower byte data	Upper byte data	Write the RMV register + FL constant speed start	Response only
RMSTFH	0059(h)	Lower byte data	Upper byte data	Write the RMV register + FH constant speed start	Response only
RMSTD	005A(h)	Lower byte data	Upper byte data	Write the RMV register + High-speed start 1	Response only
RMSTUD	005B(h)	Lower byte data	Upper byte data	Write the RMV register + High-speed start 2	Response only

Note 1: If the command is sent without any data, the RMV register will be set to 0 and the motor will operate "0" feed amount. Data is not set to the pre-register (PRMV), therefore, do not set operation data for next operation until operation completes.

4) Simultaneous start command

By setting RMD.MSY=1, and writing a start command, several LSIs are waiting for an STA signal.
By writing this command in this condition, the motors start to operate.
When several LSIs are waiting for an STA signal to arrive by setting the RMD register, write these commands to start multiple axes simultaneously.

Symbol	1st word	2nd word	3rd word	Description	Response frame
CMSTA	0006(h)			STA output (simultaneous start)	Response only
SPSTA	002A(h)			Substitute for STA input	Response only

5-3-1-2. Speed change command

If any of these commands are written while the motor is operating, the operation speed will be changed. If these commands are written while the motor is stopping, the LSI will ignore the commands.

Symbol	1st word	2nd word	3rd word	Description	Response frame
FCHGL	0040(h)			Change to FL constant speed immediately	Response only
FCHGH	0041(h)			Change to FH constant speed immediately	Response only
FSCHL	0042(h)			Decelerate to FL speed	Response only
FSCHH	0043(h)			Accelerate to FH speed	Response only

5-3-1-3. Stop command

1) Stop command
When one of these commands is written while the motor is operating, the operation stops.

Symbol	1st word	2nd word	3rd word	Description	Response frame
STOP	0049(h)			Immediate stop	Response only
SDSTP	004A(h)			Decelerate and stop	Response only

2) Simultaneous stop command
When this command is input, the LSIs stops any axis whose stop by STP input is enabled by setting the RMD register.

Symbol	1st word	2nd word	3rd word	Description	Response frame
CMSTP	0007(h)			STP output (simultaneous stop)	Response only

3) Emergency stop command
This command stops an axis in an emergency

Symbol	1st word	2nd word	3rd word	Description	Response frame
CMEMG	0005(h)			Emergency stop	Response only

5-3-2. Control commands

These commands are to control various items such as resetting counters.

5-3-2-1. NOP (no operation) command

Symbol	1st word	2nd word	3rd word	Description	Response frame
NOP	0000(h)			Invalid command	Response only

5-3-2-2. SEND interrupt reset command

Symbol	1st word	2nd word	3rd word	Description	Response frame
INTRS	0008(h)			Reset MSTS.SEND (Bit 1 of main status)	Response only

5-3-2-3. Software reset command

This command resets registers and cancels commands stored in this LSI (except for communication related items)

Symbol	1st word	2nd word	3rd word	Description	Response frame
SRST	0004(h)			Reset IC by software	Response only

5-3-2-4. Counter reset command

These commands resets the specified counter to 0.

Symbol	1st word	2nd word	3rd word	Description	Response frame
CUN1R	0020(h)			Reset COUNTER 1 (command position)	Response only
CUN2R	0021(h)			Reset COUNTER 2 (mechanical position)	Response only
CUN3R	0022(h)			Reset COUNTER 3 (general-purpose, deviation)	Response only

5-3-2-5. ERC output control command

These commands control the ERC signal using commands.

Symbol	1st word	2nd word	3rd word	Description	Response frame
ERCOUT	0024(h)			Output an ERC signal	Response only
ERCRST	0025(h)			Reset the ERC signal	Response only

5-3-2-6. PCS input command

This command has the same results as turning on the PCS input.

Symbol	1st word	2nd word	3rd word	Description	Response frame
STAON	0028(h)			Substitute for PCS input	Response only

5-3-2-7. LTC input (counter latch) command

This command has the same results as turning on the LTC input

Symbol	1st word	2nd word	3rd word	Description	Response frame
LTCH	0029(h)			Substitute for LTC input	Response only

5-3-2-8. Pre-register control command

These commands control the pre-register for operation and RCMP3.

Symbol	1st word	2nd word	3rd word	Description	Response frame
PRECAN	0026(h)			This command disables the pre-register setting for operation	Response only
PCPCAN	0027(h)			This command disables the pre-register setting for RCMP3.	Response only
PRESHF	002B(h)			This command copies the pre-register setting value for operation to the operation register.	Response only
PCPSHF	002C(h)			This command copies the pre-register setting value to the RCMP3 register.	Response only
PFCCAN	002D(h)			This command disables the pre-register and register setting for RCMP3.	Response only

Note. PCPCAN command cancels only the pre-register (PRCP3) written flag and does not cancel the register (RCMP3) written flag.
PFCCAN command cancels both the pre-register (PRCP3) written flag and the register (RCMP3) written flag.

5-3-2-9. EEPROM control commands

Symbol	Cord	Description
ROMPE	000A(h)	This command prohibits writing to external EEPROM G9103A performs the following processing for EEPROM. 1. It sends a command to permit writing (00000110) to EEPROM. 2. It ensures 3.2 µs interval time when ROMS=H. 3. It writes 11111100 into the status register of EEPROM
ROMPD	000B(h)	This command permits writing to external EEPROM G9103A performs the following processing for EEPROM. 1. It sends a command to permit writing (00000110) to EEPROM. 2. It ensures 3.2 µs interval time when ROMS=H. 3. It writes 11110000 into the status register of EEPROM
ROMWR	000C(h)	This command writes the contents of RGN0 to 3 into EEPROM G9103A performs the following processing for EEPROM. 1. It sends command to permit writing (00000110) to EEPROM. 2. It ensures 3.2 µs interval time when ROMS=H. 3. It offers Page write of 16 byte of RGN0 to RGN3 into EEPROM.
ROMRD	000D(h)	This command reads EEPROM and sets to RGN0 to 3 G9103A performs the following processing for EEPROM. 1. Reads 16 bytes from address 0 of EEPROM together at once and writes into RGN0 to RGN3 registers.

5-3-3. Register control commands

Contents	Register					Pre-register				
	Name	Read command		Write command		Name	Read command		Write command	
		Code	Symbol	Code	Symbol		Code	Symbol	Code	Symbol
Position override				0080(h)	WRMVOR					
Feed amount, target position	RMV	00D0(h)	RRMV	0090(h)	WRMV	PRMV	00C0(h)	RPRMV	00B0(h)	WPRMV
Initial speed	RFL	00D1(h)	RRFL	0091(h)	WRFL	PRFL	00C1(h)	RPRFL	00B1(h)	WPRFL
Operation speed	RFH	00D2(h)	RRFH	0092(h)	WRFH	PRFH	00C2(h)	RPRFH	00B2(h)	WPRFH
Acceleration rate	RUR	00D3(h)	RRUR	0093(h)	WRUR	PRUR	00C3(h)	RPRUR	00B3(h)	WPRUR
Deceleration rate	RDR	00D4(h)	RRDR	0094(h)	WRDR	PRDR	00C4(h)	RPRDR	00B4(h)	WPRDR
Speed magnification	RMG	00D5(h)	RRMG	0095(h)	WRMG	PRMG	00C5(h)	RPRMG	00B5(h)	WPRMG
Ramp-down point	RDP	00D6(h)	RRDP	0096(h)	WRDP	PRDP	00C6(h)	RPRDP	00B6(h)	WPRDP
Operation mode	RMD	00D7(h)	RRMD	0097(h)	WRMD	PRMD	00C7(h)	RPRMD	00B7(h)	WPRMD
Center of axis X in circular interpolation	RIP	00D8(h)	RRIP	0098(h)	WRIP	PRIP	00C8(h)	RPRIP	00B8(h)	WPRIP
S-curve range while acceleration	RUS	00D9(h)	RRUS	0099(h)	WRUS	PRUS	00C9(h)	RPRUS	00B9(h)	WPRUS
S-curve range while deceleration	RDS	00DA(h)	RRDS	009A(h)	WRDS	PRDS	00CA(h)	RPRDS	00BA(h)	WPRDS
Speed to correct feed amount	RFA	00DB(h)	RRFA	009B(h)	WRFA					
Environmental setting 1	RENV1	00DC(h)	RRENV1	009C(h)	WRENV1					
Environmental setting 2	RENV2	00DD(h)	RRENV2	009D(h)	WRENV2					
Environmental setting 3	RENV3	00DE(h)	RRENV3	009E(h)	WRENV3					
Environmental setting 4	RENV4	00DF(h)	RRENV4	009F(h)	WRENV4					
Environmental setting 5	RENV5	00E0(h)	RRENV5	00A0(h)	WRENV5					
Environmental setting 6	RENV6	00E1(h)	RRENV6	00A1(h)	WRENV6					
COUNTER1	RCUN1	00E3(h)	RRCUN1	00A3(h)	WRCUN1					
COUNTER2	RCUN2	00E4(h)	RRCUN2	00A4(h)	WRCUN2					
COUNTER3	RCUN3	00E5(h)	RRCUN3	00A5(h)	WRCUN3					
Data for comparator 1	RCMP1	00E7(h)	RRCMP1	00A7(h)	WRCMP1					
Data for comparator 2	RCMP2	00E8(h)	RRCMP2	00A8(h)	WRCMP2					
Data for comparator 3	RCMP3	00E9(h)	RRCMP3	00A9(h)	WRCMP3	PRCP3	00CB(h)	RPRCP3	00BB(h)	WPRCP3

Contents	Register					Pre-register				
	Name	Read command		Write command		Name	Read command		Write command	
		Code	Symbol	Code	Symbol		Code	Symbol	Code	Symbol
Setting event INT	RIRQ	00EC(h)	RRIRQ	00AC(h)	WRIRQ					
COUNTER1 latch data	RLTC1	00ED(h)	RRLTC1							
COUNTER2 latch data	RLTC2	00EE(h)	RRLTC2							
COUNTER3 latch data	RLTC3	00EF(h)	RRLTC3							
Extended status	RSTS	00F1(h)	RRSTS							
Error INT status	REST	00F2(h)	RREST	00AD(h)	WREST					
Event INT status	RIST	00F3(h)	RRIST	00AE(h)	WRIST					
Positioning counter	RPLS	00F4(h)	RRPLS							
EZ counter, speed monitor	RSPD	00F5(h)	RRSPD							
Ramp down point monitor	RSDC	00F6(h)	RRSDC							
Stepping counter in circular interpolation	RCIC	00FB(h)	RRCIC							
Number of stepping in circular interpolation	RCI	00FC(h)	RRCI	008C(h)	WRCI	PRCI	00CC(h)	RPRCI	00BC(h)	WPRCI
Target point of axis Y	RMVY	00FD(h)	RRMVY	008D(h)	WRMVY	PRMVY	00CD(h)	RPRMVY	00BD(h)	WPRMVY
Center of axis Y in circular interpolation	RIPY	00FE(h)	RRIPY	008E(h)	WRIPY	PRIPY	00CE(h)	RPRIPY	00BE(h)	WPRIPY
Synchronous control	RSYN	00FF(h)	RRSYN	008F(h)	WRSYN					
General-purpose register 0	RGN0	00F7(h)	RRGN0	0087(h)	WRGN0					
General-purpose register 1	RGN1	00F8(h)	RRGN1	0088(h)	WRGN1					
General-purpose register 2	RGN2	00F9(h)	RRGN2	0089(h)	WRGN2					
General-purpose register 3	RGN3	00FA(h)	RRGN3	008A(h)	WRGN3					

Note: Please dedicate the general-purpose register 0 for setting Unit ID.

5-3-3-1. Register write commands

Note: If you set several registers in one communication, please set all words (1st, 2nd and 3rd words).

Symbol	1st word	2nd word	3rd word	Description	Response frame
WRMVOR	0080(h)	Lower byte data	Upper byte data	Write override to RMV register	Response only
WRGN0	0087(h)	Lower byte data	Upper byte data	Write to RGN0 register	Response only
WRGN1	0088(h)	Lower byte data	Upper byte data	Write to RGN1 register	Response only
WRGN2	0089(h)	Lower byte data	Upper byte data	Write to RGN2 register	Response only
WRGN3	008A(h)	Lower byte data	Upper byte data	Write to RGN3 register	Response only
WRCI	008C(h)	Lower byte data	Upper byte data	Write to RCI register	Response only
WRMVY	008D(h)	Lower byte data	Upper byte data	Write to RMVY register	Response only
WRIPY	008E(h)	Lower byte data	Upper byte data	Write to RIPY register	Response only
WRSYN	008F(h)	Lower byte data	Upper byte data	Write to RSYN register	Response only
WRMV	0090(h)	Lower byte data	Upper byte data	Write to RMV register	Response only
WRFL	0091(h)	Lower byte data	Upper byte data	Write to RFL register	Response only
WRFH	0092(h)	Lower byte data	Upper byte data	Write to RFH register	Response only
WRUR	0093(h)	Data	0	Write to RUR register	Response only
WRDR	0094(h)	Data	0	Write to RDR register	Response only
WRMG	0095(h)	Data	0	Write to RMG register	Response only
WRDP	0096(h)	Lower byte data	Upper byte data	Write to RDP register	Response only
WRMD	0097(h)	Lower byte data	Upper byte data	Write to RMD register	Response only
WRIP	0098(h)	Lower byte data	Upper byte data	Write to RIP register	Response only
WRUS	0099(h)	Data	0	Write to RUS register	Response only
WRDS	009A(h)	Data	0	Write to RDS register	Response only
WRFA	009B(h)	Lower byte data	Upper byte data	Write to RFA register	Response only
WRENV1	009C(h)	Lower byte data	Upper byte data	Write to RENV1 register	Response only
WRENV2	009D(h)	Lower byte data	Upper byte data	Write to RENV2 register	Response only
WRENV3	009E(h)	Lower byte data	Upper byte data	Write to RENV3 register	Response only
WRENV4	009F(h)	Lower byte data	Upper byte data	Write to RENV4 register	Response only
WRENV5	00A0(h)	Lower byte data	Upper byte data	Write to RENV5 register	Response only
WRENV6	00A1(h)	Lower byte data	Upper byte data	Write to RENV6 register	Response only
WRCUN1	00A3(h)	Lower byte data	Upper byte data	Write to RCUN1 register	Response only
WRCUN2	00A4(h)	Lower byte data	Upper byte data	Write to RCUN2 register	Response only
WRCUN3	00A5(h)	Data	0	Write to RCUN3 register	Response only
WRCMP1	00A7(h)	Lower byte data	Upper byte data	Write to RCMP1 register	Response only
WRCMP2	00A8(h)	Lower byte data	Upper byte data	Write to RCMP2 register	Response only
WRCMP3	00A9(h)	Lower byte data	Upper byte data	Write to RCMP3 register	Response only
WRIRQ	00AC(h)	Lower byte data	Upper byte data	Write to RIRQ register	Response only
WREST	00AD(h)	Lower byte data	Upper byte data	Write to REST register	Response only

Symbol	1st word	2nd word	3rd word	Description	Response frame
WRIST	00AE(h)	Lower byte data	Upper byte data	Write to RIST register	Response only
WPRMV	00B0(h)	Lower byte data	Upper byte data	Write to pre-register for RMV.	Response only
WPRFL	00B1(h)	Lower byte data	Upper byte data	Write to pre-register for RFL.	Response only
WPRFH	00B2(h)	Lower byte data	Upper byte data	Write to pre-register for RFH.	Response only
WPRUR	00B3(h)	Data	0	Write to pre-register for RUR.	Response only
WPRDR	00B4(h)	Data	0	Write to pre-register for RDR.	Response only
WPRMG	00B5(h)	Data	0	Write to pre-register for RMG.	Response only
WPRDP	00B6(h)	Lower byte data	Upper byte data	Write to pre-register for RDP.	Response only
WPRMD	00B7(h)	Lower byte data	Upper byte data	Write to pre-register for RMD.	Response only
WPRIP	00B8(h)	Lower byte data	Upper byte data	Write to pre-register for RIP.	Response only
WPRUS	00B9(h)	Data	0	Write to pre-register for RUS.	Response only
WPRDS	00BA(h)	Data	0	Write to pre-register for RDS.	Response only
WPRCP3	00BB(h)	Lower byte data	Upper byte data	Write to pre-register for RCMP3.	Response only
WPRCI	00BC(h)	Lower byte data	Upper byte data	Write to pre-register for RCI.	Response only
WPRMVY	00BD(h)	Lower byte data	Upper byte data	Write to pre-register for RMVY.	Response only
WPRIPY	00BE(h)	Lower byte data	Upper byte data	Write to pre-register for RIPY.	Response only

5-3-3-2. Register read controls

Symbol	1st word	2nd word	3rd word	Description	Response frame
RPRMV	00C0(h)			Read pre-register for RMV.	APRMV +data
RPRFL	00C1(h)			Read pre-register for RFL	APRFL +data
RPRFH	00C2(h)			Read pre-register for RFH.	APRFH +data
RPRUR	00C3(h)			Read pre-register for RUR.	APRUR +data
RPRDR	00C4(h)			Read pre-register for RDR.	APRDR +data
RPRMG	00C5(h)			Read pre-register for RMG.	APRMG +data
RPRDP	00C6(h)			Read pre-register for RDP.	APRDP +data
RPRMD	00C7(h)			Read pre-register for RMD.	APRMD +data
RPRIP	00C8(h)			Read pre-register for RIP	APRIP +data
RPRUS	00C9(h)			Read pre-register for RUS.	APRUS +data
RPRDS	00CA(h)			Read pre-register for RDS.	APRDS +data
RPRCP3	00CB(h)			Read pre-register for RCMP3.	APRCP3 +data
RPRCI	00CC(h)			Read pre-register for RCI.	APRCI +data
RPRMVY	00CD(h)			Read pre-register for RMVY.	APRMVY +data
RPRIPY	00CE(h)			Read pre-register for RIPY.	APRIPY +data
RRMV	00D0(h)			Read RMV register	ARMV+data
RRFL	00D1(h)			Read RFL register	ARFL+ data
RRFH	00D2(h)			Read RFH register	ARFH+ data
RRUR	00D3(h)			Read RUR register	ARUR+ data
RRDR	00D4(h)			Read RDR register	ARDR+ data
RRMG	00D5(h)			Read RMG register	ARMG+ data
RRDP	00D6(h)			Read RDP register	ARDP+ data
RRMD	00D7(h)			Read RMD register	ARMD+ data
RRIP	00D8(h)			Read RIP register	ARIP+data
RRUS	00D9(h)			Read RUS register	ARUS+ data
RRDS	00DA(h)			Read RDS register	ARDS+ data
RRFA	00DB(h)			Read RFA register	ARFA+ data
RRENV1	00DC(h)			Read RENV1 register	ARENV1+ data
RRENV2	00DD(h)			Read RENV2 register	ARENV2+ data
RRENV3	00DE(h)			Read RENV3 register	ARENV3 +data
RRENV4	00DF(h)			Read RENV4 register	ARENV4+ data
RRENV5	00E0(h)			Read RENV5 register	ARENV5+ data
RRENV6	00E1(h)			Read RENV6 register	ARENV6+ data
RRCUN1	00E3(h)			Read RCUN1 register	ARCUN1+ data
RRCUN2	00E4(h)			Read RCUN2 register	ARCUN2+ data
RRCUN3	00E5(h)			Read RCUN3 register	ARCUN3+ data
RRCMP1	00E7(h)			Read RCMP1 register	ARCMP1+ data
RRCMP2	00E8(h)			Read RCMP2 register	ARCMP2+ data
RRCMP3	00E9(h)			Read RCMP3 register	ARCMP3+ data

Symbol	1st word	2nd word	3rd word	Description	Response frame
RRIRQ	00EC(h)			Read RIRQ register	ARIRQ+ data
RRLTC1	00ED(h)			Read RLTC1 register	ARLTC1+ data
RRLTC2	00EE(h)			Read RLTC2 register	ARLTC2+ data
RRLTC3	00EF(h)			Read RDTC3 register	ARLTC3+ data
RRSTS	00F1(h)			Read RSTS register	ARSTS+ data
RREST	00F2(h)			Read REST register	AREST+ data
RRIST	00F3(h)			Read RIST register	ARIST+ data
RRPLS	00F4(h)			Read RPLS register	ARPLS+ data
RRSPD	00F5(h)			Read RSPD register	ARSPD+ data
RRSDC	00F6(h)			Read RSDC register	ARSDC+ data
RRGN0	00F7(h)			Read RGN0 register	ARGN0 + data
RRGN1	00F8(h)			Read RGN1 register	ARGN1 +data
RRGN2	00F9(h)			Read RGN2 register	ARGN2 +data
RRGN3	00FA(h)			Read RGN3 register	ARGN3 + data
RRCIC	00FB(h)			Read RCIC register	ARCIC +data
RRCI	00FC(h)			Read RCI register	ARCI +data
RRMVY	00FD(h)			Read RMVY register	ARMVY + data
RRIPY	00FE(h)			Read RIPY register	ARIPY + data
RRSYN	00FF(h)			Read RSYN register	ARSYN +data

5-3-3-3. Response data from read commands

Symbol	1st word	2nd word	3rd word	Description	Effective number of bits
APRMV	00C0(h)	Lower byte data	Upper byte data	Read pre-register for RMV.	28
APRFL	00C1(h)	Lower byte data	Upper byte data	Read pre-register for RFL.	17
APRFH	00C2(h)	Lower byte data	Upper byte data	Read pre-register for RFH.	17
APRUR	00C3(h)	Data	0	Read pre-register for RUR.	16
APRDR	00C4(h)	Data	0	Read pre-register for RDR.	16
APRMG	00C5(h)	Data	0	Read pre-register for RMG.	11
APRDP	00C6(h)	Lower byte data	Upper byte data	Read pre-register for RDP.	24
APRMD	00C7(h)	Lower byte data	Upper byte data	Read pre-register for RMD.	31
APRIP	00C8(h)	Lower byte data	Upper byte data	Read pre-register for RIP.	28
APRUS	00C9(h)	Data	0	Read pre-register for RUS.	16
APRDS	00CA(h)	Data	0	Read pre-register for RDS.	16
APRCP3	00CB(h)	Lower byte data	Upper byte data	Read pre-register for RCMP3.	28
APRCI	00CC(h)	Lower byte data	Upper byte data	Read pre-register for RCI.	31
APRMVY	00CD(h)	Lower byte data	Upper byte data	Read pre-register for RMVY.	28
APRIPY	00CE(h)	Lower byte data	Upper byte data	Read pre-register for RIPY.	28
ARMV	00D0(h)	Lower byte data	Upper byte data	Read RMV register	28
ARFL	00D1(h)	Lower byte data	Upper byte data	Read RFL register	17
ARFH	00D2(h)	Lower byte data	Upper byte data	Read RFH register	17
ARUR	00D3(h)	Data	0	Read RUR register	16
ARDR	00D4(h)	Data	0	Read RDR register	16
ARMG	00D5(h)	Data	0	Read RMG register	11
ARDP	00D6(h)	Lower byte data	Upper byte data	Read RDP register	24
ARMD	00D7(h)	Lower byte data	Upper byte data	Read RMD register	31
ARIP	00D8(h)	Lower byte data	Upper byte data	Read RIP register	28
ARUS	00D9(h)	Data	0	Read RUS register	16
ARDS	00DA(h)	Data	0	Read RDS register	16
ARFA	00DB(h)	Lower byte data	Upper byte data	Read RFA register	17
ARENV1	00DC(h)	Lower byte data	Upper byte data	Read RENV1 register	32
ARENV2	00DD(h)	Lower byte data	Upper byte data	Read RENV2 register	27
ARENV3	00DE(h)	Lower byte data	Upper byte data	Read RENV3 register	31
ARENV4	00DF(h)	Lower byte data	Upper byte data	Read RENV4 register	32
ARENV5	00E0(h)	Lower byte data	Upper byte data	Read RENV5 register	32
ARENV6	00E1(h)	Lower byte data	Upper byte data	Read RENV6 register	32
ARCUN1	00E3(h)	Lower byte data	Upper byte data	Read RCUN1 register	28
ARCUN2	00E4(h)	Lower byte data	Upper byte data	Read RCUN2 register	28
ARCUN3	00E5(h)	Data	Code extension	Read RCUN3 register	16
ARCMP1	00E7(h)	Lower byte data	Upper byte data	Read RCMP1 register	28

Symbol	1st word	2nd word	3rd word	Description	Effective number of bits
ARCMP2	00E8(h)	Lower byte data	Upper byte data	Read RCMP2 register	28
ARCMP3	00E9(h)	Lower byte data	Upper byte data	Read RCMP3 register	28
ARIRQ	00EC(h)	Lower byte data	Upper byte data	Read RIRQ register	17
ARLTC1	00ED(h)	Lower byte data	Upper byte data	Read RLTC1 register	28
ARLTC2	00EE(h)	Lower byte data	Upper byte data	Read RLTC2 register	28
ARLTC3	00EF(h)	Lower byte data	Upper byte data	Read RLTC3 register	17
ARSTS	00F1(h)	Lower byte data	Upper byte data	Read RSTS register	32
AREST	00F2(h)	Lower byte data	Upper byte data	Read REST register	18
ARIST	00F3(h)	Lower byte data	Upper byte data	Read RIST register	17
ARPLS	00F4(h)	Lower byte data	Upper byte data	Read RPLS register	28
ARSPD	00F5(h)	Lower byte data	Upper byte data	Read RSPD register	27
ARSDC	00F6(h)	Lower byte data	Upper byte data	Read RSDC register	24
ARGN0	00F7(h)	Lower byte data	Upper byte data	Read RGN0 register	32
ARGN1	00F8(h)	Lower byte data	Upper byte data	Read RGN1 register	32
ARGN2	00F9(h)	Lower byte data	Upper byte data	Read RGN2 register	32
ARGN3	00FA(h)	Lower byte data	Upper byte data	Read RGN3 register	32
ARCIC	00FB(h)	Lower byte data	Upper byte data	Read RCIC register	31
ARCI	00FC(h)	Lower byte data	Upper byte data	Read RCI register	31
ARMVY	00FD(h)	Lower byte data	Upper byte data	Read RMVY register	28
ARIPY	00FE(h)	Lower byte data	Upper byte data	Read RIPY register	28
ARSYN	00FF(h)	Lower byte data	Upper byte data	Read RSYN register	32

5-4. Register and pre-register

Note 1. The bits shown by * are ignored while they are written and become 0 while they are read in the explanation hereinafter described.

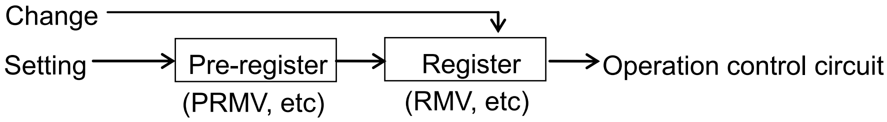
Note 2. The bits shown by & are ignored while they are written and become the same as the highest bit shown by a blank while they are read (code extension).

Note3. The default value of all registers and pre-registers are “0”. There may be cases that “0” is out of the setting range depending on register.

Note 4. If the new value you want to set is the same as the previous value, you do not need to overwrite it.

5-4-1. Pre-register

RMV, RMVY, RFL, RFH, RUR, RDR, RMG, RDP, RMD, RIP, RIPY, RUS, RDS, RCI, RCMP3 registers and start command have pre-registers.
Pre-register is a register to set data for the next operation while the motor is operating. The pre-registers of this IC are as follows. They operate like FIFO.
Pre-register consists of pre-register for operation (RMV, PRMVY, PRFL, PRFH, PRUR, PRDR, PRMG, PRDP, PRMD, PRIP, PRIPY, PRUS, PRDS, and PRCI) and for comparator (PRCP3).



Note: Normally, operation data is set through pre-register. However, operation data can be written into register directly, when pre-register is not used (when the data for next operation is not written while the motor is operating.)

5-4-1-1. Writing into pre-register for operation

There are a pre-register and a register, and up to two data for operation can be had. Write data is written into pre-register (P register name). You do not have to write again about the register with no change.
If the G9103A is stopping the motor control, the written data is shifted and recognized as a current data. If the G9103A is operating, it is stored as a pre-register data. This data is defined by writing a start command. After the current operation is complete, the data is shifted and the operation starts automatically. You can confirm a status of pre-register using RSTS.PFM0 to PFM1.
When RSTS.PFM=10 (the pre-register is full), SPRF of main status (MSTS) becomes 1. If the pre-register is full, writing data is invalid.
In the case that the current operation status is changed because of the reason such as a speed change, new data can be written in the register.

The relation between writing status of pre-register and the PFM value is as follows.

Process	Pre-register	Register	PFM	SPRF
Initial status	0 (Unfixed)	0 (Unfixed)	00	0
Write operation data 1	Operation data1 (Unfixed)	Operation data1 (Unfixed)	00	0
Write the start command (The motor starts operation by operation data 1)	Operation data1 (Unfixed)	Operation data1 (Fixed)	01	0
Write operation data 2 while the motor is operating	Operation data2 (Unfixed)	Operation data1 (Fixed)	01	0
Write the start command while the motor is operating	Operation data 2 (Fixed)	Operation data1 (Fixed)	10	1
Operation of operation data 1 is complete. The data is automatically shifted and operation starts by operation data 2.	Operation data2 (Unfixed)	Operation data2 (Fixed)	01	0
Operation of operation data 2 is complete.	Operation data2 (Unfixed)	Operation data2 (Unfixed)	00	0

In the case to set RIRQ.IRNM=1, when a status of pre-register changed from “fixed” to “unfixed” after operation is complete, the center LSI can output an interrupt request signal ($\overline{\text{INT}}$).

Note: Please set operation complete timing to “End of Cycle (RMD.METM=0)” if the next operation starts automatically using the pre-register. If you select “End of Pulse (RMD.METM=1)”, the interval between the last pulse and the initial pulse of the next operation become narrow (750ns). For the detail, see “8-3-2. Control the output pulse width and operation complete timing”

5-4-1-2. Cancellation of pre-register data for operation

By pre-register cancel command (PRECAN:0026(h)) and stop command (STOP:0049(h), SDSTP:004A(h)), all pre-register data is cancelled and the status becomes unfixed (RSTS.PFM=00). It is also cancelled by error stop.

5-4-1-3. Writing into pre-register for comparator

Comparator 3 has a pre-register. Data written to the register is written into RCMP3 directly and data written to the pre-register is written into PRCP3. Writing into PRCP3 makes comparative data fixed. The status of the pre-register for comparator can be fixed by RSTS.PFC0 to PFC1. When RSTS.PFC=10, SPDF of main status (MSTS) become 1. If the pre-register is full, writing data is invalid. Data for comparator in the pre-register is shifted when the condition turns true to false after the condition is met.
You can write data for comparator any time, regardless of stopping or operating.

The relation between writing status of the pre-register and the PFC value is as follows.

Process	Pre-register	Register	PFC	SPDF
Initial status	0 (Unfixed)	0 (Unfixed)	00	0
Write comparison data 1 into PRCP3	Comparison data 1 (Unfixed)	Comparison data 1 (Fixed)	01	0
Write comparison data 2 into PRCP3	Comparison data 2 (Fixed)	Comparison data 1 (Fixed)	10	1
Comparative result is changed from true to false by comparison data 1	Comparison data 2 (Unfixed)	Comparison data 2 (Fixed)	01	0
Comparative result is changed from true to false by comparison data 2	Comparison data 2 (Unfixed)	Comparison data 2 (Unfixed)	00	0

The center LSI can output the interrupt request signal ($\overline{\text{INT}}$) when comparator 3 turns true by the setting RIRQ.IRC3=1.

5-4-1-4. Cancellation of pre-register data for comparator.

By pre-register cancel command (RCPCAN:0027(h)), the pre-register data is cancelled and the status becomes unfixed (RSTS.PFC=00). However, please note that register is not made unfixed.

To return to RSTS.PFC=00, use PFC cancel command (PFCCAN:002D(h)).

5-4-2 Register for setting speeds

These registers are used to set the operating speeds.
Please note that with some registers "0" may be outside the allowed setting range. For details about speed setting, see "7-2. Speed pattern settings".

5-4-2-1. PRFL (RFL) : FL speed setting registers (17 bits)

This register is used to specify the initial speed (stopping speed) in the high-speed operation (with acceleration and deceleration). PRFL is the pre-register for RFL.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
*	*	*	*	*	*	*	*	*	*	*	*	*	*	*																	

Specify the start speed of FL constant operation and high-speed operation (acceleration and deceleration) within the range 1 to 100,000(186A0(h). The range 100,000 to 131,071 (186A0(h) to 1FFFF(h)) is regarded as 100,000. The actual operation speed is calculated by the formula with the setting value of RMG

5-4-2-2. PRFH (RFH) : FH speed setting registers (17 bits)

This register is used to set the operating speed. PRFH is the pre-register for RFH.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
*	*	*	*	*	*	*	*	*	*	*	*	*	*	*																	

While the motor is operating, the speed can be overridden by changing RFH register. Specify the speed of FH constant operation and the operation speed of high-speed operation (acceleration and deceleration) within the range 1 to100,000 (186A0(h)). The range 100,000 to 131,071 (186A0(h) to 1FFFF(h)) is regarded as 100,000.
In the case of high-speed operation (acceleration and deceleration), specify a larger value than the setting value of RFL. The actual operation speed is calculated by the formula with the setting value of RMG.

5-4-2-3. PRUR (RUR) : Acceleration rate setting registers (16 bits)

This register is used to set the acceleration rate. PRDR is the pre-register for RDR.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*																

Specify the characteristic of acceleration in the case of the high-speed operation (acceleration and deceleration) within the range 1 to 65,535 (0FFFF(h)).

5-4-2-4. PRDR (RDR) : Deceleration rate setting registers (17 bits)

This register is used to set the deceleration rate. PRDR is the pre-register for RDR.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*																

Specify the characteristic of deceleration in the case of the high-speed operation (acceleration and deceleration) within the range 1 to 65,535 (0FFFF(h)). Even when “Automatic setting” (RMD.MSDP=0) is selected for the ramp-down point, the value set in RDR register is be used as the deceleration rate. When RDR=0, the value set in RUR double as the deceleration rate.

5-4-2-5. PRMG (RMG): Multiplication rate register (11 bits)

This register is used to set the speed magnification rate. PRMG is the pre-register for RMG.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*											

Specify the relationship between the setting value of RFL, RFH, RFA and the operating speed, within the range of 2 to 2,047 (07FF(h)). The higher the multiplication rate is, the coarser the speed steps can be selected. Normally, use as small a multiplication rate as possible. The operation speed [PPS] is the product of multiplying the speed rate by the speed register setting.

5-4-2-6. PRDP (RDP): Ramp down point setting register (24 bits)

This register is used to set a ramping-down point (deceleration start point) for positioning operations. PRDP is the pre-register for RDP.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
#	#	#	#	#	#	#	#																								

Specify the value used to determine the deceleration start point in an acceleration/deceleration or positioning operation.
Bits shown by a "#" symbol are ignored when they are written and setting details varies according to the setting of RMD.MSDP when these bits are read.

MSDP	Setting details	bit #
0	Offset for automatically set values. When a positive value is entered, the G9103A will start deceleration earlier and the FL speed range will be used longer. When a negative value is entered, the G9103A will start deceleration later and the speed of the axis will not reach the FL speed.	Same as bit 23.
1	When number of pulses left drops to less than a set value, the motor starts to decelerate.	0

5-4-2-7. PRUS (RUS): Acceleration S-curve range setting register (16 bits)

This register is used to specify the S-curve range of the S-curve acceleration. PRUS is the pre-register for RUS.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*																

Specify the S-curve acceleration value for an S-curve acceleration/deceleration operation, within the range of 1 to 50,000 (0C350(h)).
All values from 50,000 to 65,535 (0C350(h) to 0FFFF(h)) are treated as 50,000.
The S-curve acceleration range S_{SU} is calculated by the formula with the RMG value.
If "0" is entered, the G9103A substitutes the value calculated by (RFH - RFL) / 2, and operates using an S-curve acceleration that does not have any linear sections.
If a value larger than (RFH - RFL) / 2 is entered, the speed does not reach the maximum acceleration speed and the acceleration time does not match the speed calculated by the formula. Therefore enter a value smaller than "(RFH - RFL) / 2."

5-4-2-8. PRDS (RDS): Deceleration S-curve range setting register (16 bits)

This register is used to specify the S-curve range of the S-curve deceleration. PRDS is the pre-register for PRDS.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*																

Specify the S-curve deceleration value for an S-curve acceleration/deceleration operation, within the range of 1 to 50,000 (0C350(h)).
All values from 50,000 to 65,535 (0C350(h) to 0FFFF(h)) are treated as 50,000.
The S-curve deceleration range S_{SD} is calculated by the formula with the RMG value.
If "0" is entered, the G9103A substitutes the value calculated by (RFH - RFL) / 2, and operates using an S-curve deceleration that does not have any linear sections.
If a value larger than (RFH - RFL) / 2 is entered, the speed does not reach the maximum deceleration speed and the deceleration time does not match the speed calculated using the formula. Therefore enter a value smaller than "(RFH - RFL) / 2."

5-4-2-9. RFA: FA speed setting register (17 bits)

This register is used to set the constant speed for backlash correction.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
*	*	*	*	*	*	*	*	*	*	*	*	*	*	*																	

Set the correction speed (FA speed) to feed a specific amount during backlash correction, within the range of 1 to 100,000 (186A0(h)).
ALL values from 100,000 to 131,071 (186A0(h) to 1FFFF(h)) are treated as 100,000.
The actual operation speed is be calculated by the formula with RMG value.
This value can also be used for the reverse constant speed during an origin return operation.

5-4-3. Feed amount setting registers

5-4-3-1. PRMV (RMV) :Feed amount for positioning setting registers (28 bits)

This register is used to specify the target position for positioning operation. In linear interpolation and circular interpolation, specify an X-coordinate of the target position with an incremental value.
PRMV is a pre-register for RMV.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
&	&	&	&																												

The details for setting the register may vary according to the operation mode.
Setting range: -134,217,728 to +134,217,727.
By changing RMV register while in positioning operation, the feed amount can be overridden.

5-4-3-2. PRMVY (RMVY) : Feed amount setting register for Y-coordinate in interpolation (28 bits)

In the operation of linear interpolation and circular interpolation, specify Y-coordinate of the target position with an incremental value.
PRMVY is the pre-register for RMVY.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
&	&	&	&																												

Setting range: -134,217,728 to +134,217,727.

5-4-3-3. PRIP (RIP): Positioning setting register for the center of X-coordinate of the center in circular interpolation (28 bits)

In circular interpolation, specify the center of X-coordinate with an incremental value
PRIP is the pre-register for RIP.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
&	&	&	&																												

Setting range: -134,217,728 to +134,217,727.

5-4-3-4. PRIPY (RIPY): Positioning setting register for the center of Y-coordinate in circular interpolation (28 bits)

In circular interpolation, specify the center of Y-coordinate with an incremental value
PRIPY is the pre-register for RIPY.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
&	&	&	&																												

Setting range: -134,217,728 to +134,217,727.

5-4-3-5. PRCI (RCI): Setting register for number of steps necessary to complete circular interpolation (31 bits)

For circular interpolation with acceleration / deceleration, specify the data to control automatic ramp down. For circular interpolation in constant speed, the setting value is invalid.
PRCI is the pre-register for RCI.

By one-step operation in circular interpolation, the axis moves in one of the 8 directions of 45 angular units. Using this register, specify the number of steps necessary to complete circular interpolation operation. CPU calculates this setting value. For the detail, see “6-7-2-2. Number of stepping in circular interpolation”.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0																															

Setting range: 0 to +2,147,483,647.

5-4-3-6. RCIC: Step counter for circular interpolation (31 bits)

This register is used to read step count value for circular interpolation (read only). When the motors start circular interpolation, value in RCI register is loaded and counted down at every steps for circular interpolation, However, after count value reaches 0, the value is not be counted down.
The count value after completing circular interpolation is memorized until the next start of circular interpolation operation.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0																															

Setting range: 0 to +2,147,483,647.

5-4-4. Environment setting registers

The environment setting registers consist of registers used to set and monitor operation mode, counters, comparators, environment, and interrupt controls.

Register	Description	Bit length	Set range	R/W
RMD	Set the operation mode	31		R/W
RENV1	Environment Register 1	32		R/W
RENV2	Environment Register 2	27		R/W
RENV3	Environment Register 3	31		R/W
RENV4	Environment Register 4	32		R/W
RENV5	Environment Register 5	32		R/W
RENV6	Environment Register 6	32		R/W
RCUN1	COUNTER1 (Command position)	28	-134,217,728 to 134,217,727 (8000000(h)) (7FFFFFF(h))	R/W
RCUN2	COUNTER2 (Mechanical position)	28	-134,217,728 to 134,217,727 (8000000(h)) (7FFFFFF(h))	R/W
RCUN3	COUNTER3 (General-purpose, Deviation)	16	-32,768 to 32,767 (8000(h)) (7FFF(h))	R/W
RCMP1	Comparison data for Comparator 1	28	-134,217,728 to 134,217,727 (8000000(h)) (7FFFFFF(h))	R/W
RCMP2	Comparison data for Comparator 2	28	-134,217,728 to 134,217,727 (8000000(h)) (7FFFFFF(h))	R/W
RCMP3	Comparison data for Comparator 3	28	-134,217,728 to 134,217,727 (8000000(h)) (7FFFFFF(h))	R/W
RIRQ	Set event interrupt factors	17		R/W
RCUN1	COUNTER1 latched data	28	-134,217,728 to 134,217,727 (8000000(h)) (7FFFFFF(h))	R/W
RCUN2	COUNTER2 latched data	28	-134,217,728 to 134,217,727 (8000000(h)) (7FFFFFF(h))	R/W
RCUN3	COUNTER3 latched data	17	-32,768 to 32,767 (8000(h)) (7FFF(h))	R/W
RSTS	Extension status	32		R
REST	Error interrupt status	18		R/W
RIST	Event interrupt status	17		R/W
RPLS	Positioning Counter (residual number of pulses to feed)	28	0 to 134,217,728 (8000000(h))	R
RSPD	EZ counter, current speed monitor	27	1 to 100,000 (186A0(h)), etc.	R
RSDC	Ramp down value by automatic calculation	24	0 to 16,777,215 (0FFFFFF(h))	R
RSYN	Synchronous control register	15		R/W

5-4-4-1. RMD (RMD) registers

These registers are used to set the operation mode.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MSPE	MSY	MPCS	MSDP	METM	MSMD	MINP	MSDE	MENI	MOD						

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	MDMY	MERO	MERI	MIPF	MDSC	MBIM	MIOR	MFH	MUB	MMPH	MPH	MINT	MMSK	MADJ	MSPO

Bits	Bit name	Description
0 to 6	MOD	These bits are to set operation mode 000 0000 (00(h)): Positive direction continuous operation controlled by command control. 000 1000 (08(h)): Negative direction continuous operation controlled by command control. 000 0001 (01(h)): Continuous operation controlled by pulsar (PA/PB) input. 001 0000 (10(h)): Origin return operation in the positive direction. 001 1000 (18(h)): Origin return operation in the negative direction. 001 0010 (12(h)): Leaving the origin position in the positive direction. 001 1010 (1A(h)): Leaving the origin position in the negative direction. 001 0101 (15(h)): Origin search in the positive direction 001 1101 (1D(h)): Origin search in the negative direction 010 0000 (20(h)): Feed to +EL or +SL position. 010 1000 (28(h)): Feed to -EL or -SL position. 010 0010 (22(h)): Leaving the -EL or -SL position. 010 1010 (2A(h)): Leaving the +EL or +SL position. 010 0100 (24(h)): Feed in the positive direction for a specified number of EZ counts. 010 1100 (2C(h)): Feed in the negative direction for a specified number of EZ counts. 100 0001 (41(h)): Positioning operation (specify the incremental target position) 100 0010 (42(h)): Positioning operation (specify the absolute position in COUNTER 1) 100 0011 (43(h)): Positioning operation (specify the absolute position in COUNTER 2) 100 0100 (44(h)): Zero return of command position (COUNTER 1). 100 0101 (45(h)): Zero return of mechanical position (COUNTER 2). 100 0110 (46(h)): Single pulse operation in the positive direction. 100 1110 (4E(h)): Single pulse operation in the negative direction. 100 0111 (47(h)): Timer operation 101 0001 (51(h)): Positioning operation synchronized with PA/PB. (Specify the target incremental position) 101 0010 (52(h)): Positioning operation synchronized with PA/PB (Specify the absolute position of COUNTER 1) 101 0011 (53(h)): Positioning operation synchronized with PA/PB (Specify the absolute position of COUNTER 2) 101 0100 (54(h)): Zero return of the command position synchronized with PA/PB. 101 0101 (55(h)): Zero return of the mechanical position synchronized with PA/PB. 110 0000 (60(h)): Continuous linear interpolation (output X axis pulses). 110 0001 (61(h)): Linear interpolation (output X axis pulses). 110 0100 (64(h)): Circular interpolation in the CW direction (output X axis pulses). 110 0101 (65(h)): Circular interpolation in the CCW direction (output X axis pulses). 110 1000 (68(h)): Continuous linear interpolation synchronized with PA/PB (output X axis pulses). 110 1001 (69(h)): Linear interpolation synchronized with PA/PB (output X axis pulses). 110 1100 (6C(h)): Circular interpolation in the CW direction synchronized with PA/PB (output X axis pulses).

Bits	Bit name	Description
		110 1101 (6D(h)): Circular interpolation in the CCW direction synchronized with PA/PB (output X axis pulses). 111 0000 (70(h)): Continuous linear interpolation (output Y axis pulses). 111 0001 (71(h)): Linear interpolation (output Y axis pulses). 111 0100 (74(h)): Circular interpolation in the CW direction (output Y axis pulses). 111 0101 (75(h)): Circular interpolation in the CCW direction (output Y axis pulses). 111 1000 (78(h)): Continuous linear interpolation synchronized with PA/PB (output Y axis pulses). 111 1001 (79(h)): Linear interpolation synchronized with PA/PB (output Y axis pulses). 111 1100 (7C(h)): Circular interpolation in the CW direction synchronized with PA/PB (output Y axis pulses). 111 1101 (7D(h)): Circular interpolation in the CCW direction synchronized with PA/PB (output Y axis pulses).
7	MENI	1: When the pre-register for operation is fixed, MSTS.SEND changes to 1 after the current operation has completed.
8	MSDE	0: SD input is ignored. (Checking terminal condition can be done with RSTS.SDIN) 1: Decelerates (deceleration stop) by turning ON the SD input.
9	MINP	0: Delay using INP input will be invalid. (Checking can be done with RSTS.SINP) 1: Operation completes by turning ON the INP input.
10	MSMD	Specify an acceleration/deceleration type for high speed operation. (0: Linear acceleration/deceleration. 1: S-curve acceleration/deceleration.)
11	METM	Specify the operation complete timing. (0: End of cycle. 1: End of pulse.) When using vibration reduction function, select "End of pulse."
12	MSDP	Specify the ramping-down point for high speed operation. (Effective during positioning operations.) (0: Automatic setting. 1: Manual setting.)
13	MPCS	Enable or disable PCS input. (1: When positioning, the G9103A controls the number of pulses after the PCS input turns ON.) [Override 2 for the target position.]
14	MSY	After writing a start command, the G9103A can delay the start until some other event occurs. 0: Start immediately. 1: Start when the STA input is supplied, the command 0006(h) or 002A(h) is written, or when the group number is correspond to one set in the simultaneous command of broadcast communication.
15	MSPE	1: Decelerate and stop or stop the motor immediately when the STP input is supplied or when the group number corresponds with one set in the simultaneous start command of broadcast communication.
16	MSPO	1: Output an STP (simultaneous stop) signal when the motor has stopped due to an error.
17	MADJ	Specify an FH correction function. (0: ON. 1: OFF.)
18	MMSK	1: Mask the pulse output.
19	MINT	1: Mask the interrupt output (SINT). (The interrupt status continues to change.)
20	MPH	Select the signal output for the following terminals: $\overline{\text{BSY}}$/PH1, $\overline{\text{FUP}}$/PH2, $\overline{\text{FDW}}$/PH3, and $\overline{\text{MVC}}$/PH4. 0: Output $\overline{\text{BSY}}$, $\overline{\text{FUP}}$, $\overline{\text{FDW}}$ and $\overline{\text{MVC}}$ signals. 1: Output PH1, PH2, PH3, and PH4 signals.
21	MMPH	Mask PH1, PH2, PH3, and PH4 signals. 0: Output “LLLL” or “LLHH” level from PH1, PH2, PH3, and PH4. 1: Output the PH1, PH2, PH3, and PH4 signals.
22	MUB	Select the excitation method for the PH1, PH2, PH3, and PH4 signals. 0: Output excitation sequence for a 2-phase unipolar motor. 1: Output excitation sequence for a 2-phase bipolar motor.
23	MFH	Select the excitation sequence for the PH1, PH2, PH3, and PH4 signals. 0: Output excitation sequence for full-step. 1: Output excitation sequence for half-step.

Bits	Bit name	Description
24	MIOR	Select a monitoring method for output setting bits of a general-purpose I/O port. This is used when you do not want the input change interrupt in the center LSI to function when an output port changes. 0: Read the output bit status from Port 2. 1: Regardless of status of the output bit, the respective bit for Port 2 becomes "0."
25	MBIM	1: When MMPH=0 (mask PH1 to 4), change the output of PH1 to 4 to “LLHH”.
26	MSDC	1: Fix the method to set the ramp down point automatically, to “the count method”.
27	MIPF	1: In interpolation, make the synthesized speed constant while recognizing as the interpolation between 2 axes.
28	MERI	1: When the axis of the G9103A that is specified by RSYN.DNSTP stops abnormally, the G9103A stops operation of own axis.
29	MERO	1: When the axis stops abnormally, outputs the stop request to other axes.
30	MDMY	1: Masks the output pulses and make counters ignore the pulses. (Dummy operation) In continuous interpolation using the pre-register, use this bit to switch the interpolated axis.
31	Not defined	(Always set to 0.)

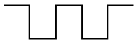
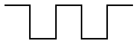
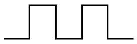
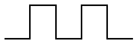
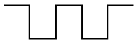
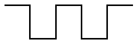
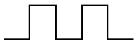
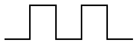
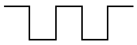
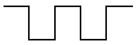
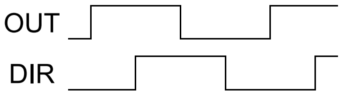
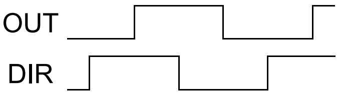
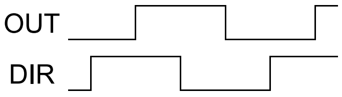
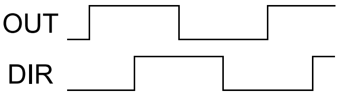
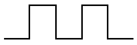
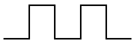
5-4-4-2. RENV1 register

This register is used for the environment setting 1. This is mainly used to set the specifications for input/output terminals.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ERCL	EPW2	EPW1	EPW0	EROR	EROE	ALML	ALMM	ORGL	SDL	SDLT	SDM	ELM	PMD2	PMD1	PMD0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CDWN	MREV	PDTC	SEDR	SEDM	DTMF	FLTR	PCSL	LTCL	INPL	CLR1	CLR0	STPM	STAM	ETW1	ETW0

Bits	Bit name	Description (The following is an example when RENV1.MREV=0.)
0 to 2	PMD 0 to 2	These bit is to specify output pulse details

PMD 0〜2	Operation in (+) direction		Operation in (-) direction	
	OUT output	DIR output	OUT output	DIR output
000		High		ow
001		High		Low
010		Low		High
011		Low		High
100		High	High	
101				
110				
111		Low	Low	

When RENV1.MREV=1, operation in (+) direction and operation in (-) direction is switched.

Bits	Bit name	Description
3	ELM	Specify the process to occur when the EL input is turned ON. (0: Immediate stop. 1: Deceleration stop.) Note 1
4	SDM	Specify the process to occur when the SD input is turned ON. (0: Deceleration only. 1: Deceleration stop.)
5	SDLT	Specify the latch function of the SD input. (0: OFF. 1: ON.) Turns ON when the SD signal width is short. When the SD input is OFF while the motor is starting, the latch signal is reset. The latch signal is also reset when SDLT = 0.
6	SDL	Specify the SD signal input logic. (0: Negative logic. 1: Positive logic.)
7	ORGL	Specify the ORG signal input logic. (0: Negative logic. 1: Positive logic.)
8	ALMM	Specify the process to occur when the ALM input is turned ON. (0: Immediate stop. 1: Deceleration stop.)
9	ALML	Specify the ALM signal input logic. (0: Negative logic. 1: Positive logic.)

Bits	Bit name	Description
10	EROE	1: G9103A outputs an ERC signal automatically when the axis is stopped immediately by a +EL, -EL, ALM, or EMG input signal. However, G9103A does not output the ERC signal when a deceleration stop occurs on the axis. When the EL signal is specified for a normal stop, by setting RMD.MOD = "010 X000" (feed to the EL position) in RMD register, the ERC signal is output if an immediate stop occurs.
11	EROR	1: G9103A outputs the ERC signal automatically when the axis completes an origin return.
12 to 14	EPW0 to 2	Specify the pulse width of the ERC output signal (RENV1.CDWN=0) or the current recover time (RENV1.CDWN=1). (There is a margin of error of approximate 4% one way or the other) 1. ERC signal pulse width (RENV1.CDWN=0) 000: 12 μsec 001: 93 μsec 010: 371 μsec 011:1.5 msec 100: 12 msec 101: 48 msec 110: 95 msec 111: Level output 2. Current recover time (RENV1.CDWN=1) 000: 6.4 msec 001: 13 msec 010: 26 msec 011:51 msec 100: 102 msec 101: 205 msec 110: 410 msec 111: Prohibited cooking
15	ERCL	Specify the ERC signal output logic. (0: Negative logic. 1: Positive logic.)
16 to 17	ETW0 to 1	Specify the ERC signal OFF timer time (RENV1.CDWN=0) or the current down delay time (RENV1.CDWN=1). (There is a margin of error of approximate 4% one way or the other) 1. ERC signal OFF timer time (RENV1.CDWN=0) 00: 0 μsec 10: 1.5 msec 01: 12 μsec 11: 95 msec 2. Current down delay time (RENV1.CDWN=1) 00: 51 msec 10: 205 msec 01: 102 msec 11: 410 msec
18	STAM	Specify the STA signal input type. (0: Level trigger. 1: Edge trigger.)
19	STPM	Specify a stop method using STP input. (0: Immediate stop. 1: Deceleration stop.)
20 to 21	CLR0 to 1	Specify the CLR input. 00: Clear on the falling edge 10: Clear on a LOW. 01: Clear on the rising edge 11: Clear on a HIGH.
22	INPL	Specify the INP signal input logic. (0: Negative logic. 1: Positive logic.)
23	LTCL	Specify the operation edge for the LTC signal. (0: Falling. 1: Rising)
24	PCSL	Specify the PCS signal input logic. (0: Negative logic. 1: Positive logic.)
25	FLTR	0: Apply a filter to the +EL, -EL, SD, ORG, ALM, INP, or EMG inputs. When a filter is applied, signal pulses shorter than 4 μsec are ignored.
26	DTMF	1: Turn OFF the direction change timer (0.2 msec) function.
27	SEDM	1: Mask the interrupt by SEND output. (Even if MSTS.SEND=1, MSTS.SINT does not be set to "1".
28	SEDR	1: Reset SEND when the motor starts. (SEND is reset by writing the command 0040(h) to 005B(h).)
29	PDTC	1: Always make the pulse width a 50% duty cycle.
30	MREV	1: Reverse the rotation direction of the motor
31	CDWN	1: Output the current down signal from the ERC/CDWN terminal.

Note1: When a deceleration stop (ELM = 1) has been specified to occur when the EL input turns ON, the motor starts the deceleration when the EL input is turned ON. Therefore, the motor stops after the mechanical position passes over the EL position. In this case, be careful to avoid collisions of mechanical systems.

5-4-4-3. RENV2 register

This is a register for the Environment 2 settings. Specify the function of general-purpose I/O terminals, EA/EB input, and PA/PB input.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PIM1	PIM0	PINF	EZL	EDIR	EIM1	EIM0	EINF	P7M	P6M	P5M	P4M	P3M	P2M	P1M	P0M

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	GN2	GN1	GN0	SIFM	IDL2	IDL1	IDL0	ROMB	POFF	EOFF	PDIR

Bits	Bit name	Description
0	P0M	Specify the operation of P0 terminal. 0: General-purpose input 1: General-purpose output
1	P1M	Specify the operation of P1 terminal. 0: General-purpose input 1: General-purpose output
2	P2M	Specify the operation of P2 terminal. 0: General-purpose input 1: General-purpose output
3	P3M	Specify the operation of P3 terminal. 0: General-purpose input 1: General-purpose output
4	P4M	Specify the operation of P4 terminal. 0: General-purpose input 1: General-purpose output.
5	P5M	Specify the operation of P5 terminal. 0: General-purpose input 1: General-purpose output
6	P6M	Specify the operation of P6 terminal. 0: General-purpose input 1: General-purpose output
7	P7M	Specify the operation of P7 terminal. 0: General-purpose input 1: General-purpose output
8	EINF	0: Apply a noise filter to the EA/EB/EZ input. Note 1.
9 to 10	EIM0 to 1	Specify the EA/EB input operation. 00: Multiply a 90 degree phase difference by 1 (Count forward when the EA input phase is ahead.) 01: Multiply a 90 degree phase difference by 2 (Count forward when the EA input phase is ahead.) 10: Multiply a 90 degree phase difference by 4 (Count forward when the EA input phase is ahead.) 11: Count forward when the EA signal rises, count backward when the EB signal rises.
11	EDIR	1: Reverse the counting direction of the EA/EB inputs.
12	EZL	Specify the EZ signal input logic. (0: Falling edge. 1: Rising edge.)
13	PINF	0: Apply a noise filter to the PA/PB input. Note 1
14 to 15	PIM0 to 1	Specify the PA/PB input operation. 00: Multiply a 90 degree phase difference by 1 (Count forward when the PA input phase is ahead.) 01: Multiply a 90 degree phase difference by 2 (Count forward when the PA input phase is ahead.) 10: Multiply a 90 degree phase difference by 4 (Count forward when PA input phase is ahead.) 11: Count forward when the PA signal rises, count backward when the PB signal rises.
16	PDIR	1: Reverse the counting direction of the PA/PB input.

Bits	Bit name	Description
17	EOFF	1: Disable the EA/EB input.
18	POFF	1: Disable the PA/PB input.
19	ROMB	This bit is for read only and ignored when writing. 1. G9103A is accessing to EEPROM that is connected externally.
20 to 22	IDL 0 to 2	Set the number of idling pulses (0 to 7 pulses).
23	SIFM	1: Make the general-purpose I/O terminals P4 to P7, the terminals to control serial bus.
24 to 26	GN0 to 2	Specify a group number to be used for broadcast communication. (Ex “100”=4) Only when this bits are “000”, the setting vale of $\overline{\text{GRP0}}$ to $\overline{\text{GRP2}}$ is enabled.x
27 to 31	Not defined	(Always set to 0.)

Note 1: Signals shorter than 150 ns are ignored.

5-4-4-4. RENV3 register

This is a register for the Environment 3 settings. Origin return methods and counter operation specifications are the main function of this register.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	BSYC	CI32	CI31	CI30	CI21	CI20	EZD3	EZD2	EZD1	EZD0	ORM3	ORM2	ORM1	ORM0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	CU3H	CU2H	CU1H	0	CU3B	CU2B	CU1B	0	CU3R	CU2R	CU1R	0	CU3C	CU2C	CU1C

Bit	Bit name	Description
0 to 3	ORM0 to 3	Specify an origin return method. 0000: Origin return operation 0 - The axis will stop immediately (or make a deceleration stop when feeding at high speed) when the ORG input turns ON. - COUNTER reset timing: When the ORG input turns ON. 0001: Origin return operation 1 - The axis will stop immediately (or decelerate and stop when feeding at high speed) when the ORG input turns ON. Then, it feeds in the opposite direction at RFA constant speed until the ORG input turns OFF. Then, the axis will move back in the original direction at RFA speed and stop immediately when the ORG input turns ON again. - COUNTER reset timing: When the ORG input signal turns ON. 0010: Origin return operation 2 - After the ORG input turns ON when feeding at constant speed, the LSI will start counting EZ pulses. The axis will stop immediately when the LSI finishes counting the specified number of EZ pulses. After the ORG input turns ON when feeding at high speed, the axis will start decelerating. At the same time, the LSI will start counting the EZ pulses. When the LSI finishes counting the specified number of EZ pulses, the axis will stop immediately. - COUNTER reset timing: When finishing counting the specified number of EZ pulses. 0011: Origin return operation 3 - After the ORG signal turns ON when feeding at constant speed, the LSI will start counting EZ pulses. The axis will stop immediately when the LSI finishes counting the specified number of EZ pulses. After the ORG signal turns ON when feeding at high speed, the LSI will start counting EZ pulses. When the LSI finishes counting the specified number of EZ pulses, the axis will decelerate and stop. COUNTER reset timing: When finishing counting the specified number of EZ pulses. 0100: Origin return operation 4 - After the ORG input turns ON when feeding at constant speed, the axis will stop immediately (or decelerate and stop when feeding at high speed). Then, the axis will start feeding in the opposite direction at RFA constant speed. After the ORG input turns OFF, the LSI will start counting EZ pulses. After the LSI finishes counting the specified number of EZ pulses, the axis will stop immediately. - COUNTER reset timing: When finishing counting the specified number of EZ pulses.

Bit	Bit name	Description
		0101: Origin return operation 5 - After the ORG input turns ON when feeding at constant speed, the axis will stop immediately (or decelerate and stop when feeding at high speed). Then, the axis will start feeding in the opposite direction. After the ORG input turns OFF, the LSI will start counting EZ pulses. After the LSI finishes counting the specified number of EZ pulses, the axis will stop immediately (or decelerate and stop when feeding at high speed). - COUNTER reset timing: When finishing counting the specified number of EZ pulses. 0110: Origin return operation 6 - After the EL input turns ON when feeding at constant speed, the axis will stop immediately (or decelerate and stop when ELM is 1). Then, the axis will start feeding in the opposite direction at RFA constant speed. When the EL signal turns OFF, the axis will stop immediately when the LSI finishes counting the specified number of EZ pulses. - COUNTER reset timing: When the EL input is OFF. 0111: Origin return operation 7 - After the EL signal turns ON when feeding at constant speed, the axis will stop immediately (or decelerate and stop when ELM is 1). Then, the axis will start feeding in the opposite direction at RFA constant speed. After the EL signal turns OFF, the LSI will start counting EZ pulses. After the LSI finishes counting the specified number of EZ pulses, the axis will stop immediately. - COUNTER reset timing: When stopped by finishing counting the specified number of EL pulses. 1000:Origin return operation 8 - After the EL signal turns ON when feeding at constant speed, the axis will stop immediately (or decelerate and stop when ELM is 1). Then, the axis will start feeding in the opposite direction at RFL constant speed. After the EL signal turns OFF, the LSI will start counting EZ pulses. After the LSI finishes counting the specified number of EZ pulses, the axis will stop immediately. - CONTER reset timing: When finishing counting the specified number of EZ signal. 1001: Origin return operation 9 - After the process in origin return operation 0 has executed, the motor operates until COUNTER2 = 0. 1010: Origin return operation 10 - After the process in origin return operation 3 has executed, the motor operates until COUNTER2 = 0. 1011: Origin return operation 11 - After the process in origin return operation 5 has executed, the motor operates until COUNTER2 = 0. 1100: Origin return operation 12 - After the process in origin return operation 8 has executed, the motor operates until COUNTER2 = 0).
4 to 7	EZD0 to 3	Specify the EZ count value that is used for origin return operations. 0000 (1st count) to 1111 (16th count)
8 to 9	CI20 to 21	Select the input count for COUNTER 2 (mechanical position). 00: EA/EB input 01: Output pulse 10: PA/PB input
10 to 12	CI30 to 32	Select the input count for COUNTER 3 (general-purpose, deviation)

Bit	Bit name	Description
		000: Output pulse 001: EA/EB input 010: PA/PB input 011: 1/4096 division clock of 40MHz. 100: Output pulse and EA/EB input (deviation count) 101: Output pulse and PA/PB input (deviation count) 110: EA/EB input and PA/PB input (deviation count)
13	BSYC	1: Operate COUNTER 3 only while the LSI is operating (BSY is LOW).
14 to 15	Not defined	(Always set to 0.)
16	CU1C	1: Reset COUNTER 1 (command position) when the CLR input turns ON.
17	CU2C	1: Reset COUNTER 2 (mechanical position) when the CLR input turns ON.
18	CU3C	1: Reset COUNTER 3 (general-purpose, deviation counter) when the CLR input turns ON.
19	Not defined	(Always set to 0.)
20	CU1R	1: Reset COUNTER 1 (command position) at the origin position when the origin return is complete.
21	CU2R	1: Reset COUNTER 2 (mechanical position) at the origin position when the origin return is complete.
22	CU3R	1: Reset COUNTER 3 (general-purpose, deviation counter) at the origin position when the origin return is complete.
23	Not defined	(Always set to 0.)
24	CU1B	1: Operate COUNTER 1 (command position) while in backlash correction mode.
25	CU2B	1: Operate COUNTER 2 (mechanical position) while in backlash correction mode.
26	CU3B	1: Operate COUNTER 3 (general-purpose, deviation counter) while in backlash correction mode.
27	Not defined	(Always set to 0.)
28	CU1H	1: Stop the counting operation on COUNTER 1 (command position)
29	CU2H	1: Stop the counting operation on COUNTER 2 (mechanical position).
30	CU3H	1: Stop the counting operation on COUNTER 3 (general-purpose, deviation counter).
31	Not defined	(Always set to 0.)

5-4-4-5. RENV4 register

This register is used for Environment 4 settings. Set up comparators 1 to 3.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
C2RM	C2D1	C2D0	C2S2	C2S1	C2S0	C2C1	C2C0	C1RM	C1D1	C1D0	C1S2	C1S1	C1S0	C1C1	C1C0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ISMR	CU3L	CU2L	CU1L	LTOF	LTFD	LTM1	LTM0	C3D1	C3D0	C3S3	C3S2	C3S1	C3S0	C2C1	C3C0

Bit	Bit name	Description
0 to 1	C1C0 to 1	Select a comparison counter for comparator 1. Note 1 00: COUNTER1 (command position) 01: COUNTER2 (mechanical position) 10: COUNTER3 (general-purpose, deviation counter) 11: The comparison conditions are not satisfied.
2 to 4	C1S0 to 2	Select a comparison method for comparator 1. Note 2 001: RCMP1 data = Comparison counter (regardless of counting direction) 010: RCMP1 data = Comparison counter (while counting forward) 011: RCMP1 data = Comparison counter (while counting backward) 100: RCMP1 data > Comparison counter data 101: RCMP1 data < Comparison counter data 110: Use as positive end software limit (RCMP1< COUNTER1) Others: Treat that the comparison conditions are not satisfied. However, set 000 when C1RM=1
5 to 6	C1D0 to 1	Select a process to execute when the Comparator 1 conditions are met. 00: None (use to output $\overline{\text{INT}}$ from the center LSI.) 01: Immediate stop. 10: Deceleration stop. 11: Change operation data to the value of pre-register (speed change).
7	C1RM	1:Using the comparator 1, make COUNTER 1 operate as a ring counter.
8 to 9	C2C0 to 1	Select a comparison counter for Comparator 2. Note 1. 00: COUNTER1 (command position) 01: COUNTER2 (mechanical position) 10: COUNTER3 (general-purpose, deviation) 11: The comparison conditions are not satisfied at any time.
10 to 12	C2S0 to 2	Select a comparison method for Comparator 2. Note 2. 001: RCMP2 data = Comparison counter (regardless of counting direction) 010: RCMP2 data = Comparison counter (while counting up) 011: RCMP2 data = Comparison counter (while counting down) 100: RCMP2 data > Comparison counter data 101: RCMP2 data < Comparison counter data 110: Use as negative end software limit (RCMP2>COUNTER1) Others: The comparison conditions are not satisfied at any time. However, set 000 when C2RM=1.
13 to 14	C2D0 to 1	Select a process to execute when the Comparator 2 conditions are met. 00: None (use to output $\overline{\text{INT}}$ from the center LSI) 01: Immediate stop. 10: Deceleration stop. 11: Change operation data to the value of pre-register (speed change).
15	C2RM	1:Using the comparator 2, make counters 2 operate as a ring counter.
16 to 17	C3C0 to 1	Select a comparison counter for Comparator 3. Note 1 00: COUNTER1 (command position) 01: COUNTER2 (mechanical position) 10: COUNTER3 (deviation counter) 11: The comparison conditions are not satisfied at any time.
18 to 21	C3S0 to 3	Select a comparison method for comparator 3. Note 3 0001: RCMP3 data = Comparison counter (regardless of counting direction) 0010: RCMP3 data = Comparison counter (while counting forward) 0011: RCMP3 data = Comparison counter (while counting backward)

Bit	Bit name	Description
		0100: RCMP3 data > Comparison counter data 0101: RCMP3 data < Comparison counter data 0111: Prohibited setting 1000: Use as an output for the IDX (synchronous) signal (regardless of the count direction) 1001: Use as an output for the IDX (synchronous) signal (while counting forward) 1010: Use as an output for the IDX (synchronous) signal (while counting backward) Others: The comparison conditions are not met at any time.
22 to 23	C3D0 to 1	Select a process to execute when the Comparator 3 conditions are met. 00: None (use to output $\overline{\text{INT}}$ from the center LSI) 01: Immediate stop. 10: Deceleration stop. 11: Change operation data to the value of pre-register (speed change).
24 to 25	LTM0 to 1	Select latch timing of counter (COUNTER 1 to 3) 00: When LTC input goes ON from OFF 01: ORG input 10: When comparator 2 conditions are met. 11: When comparator 3 conditions are met.
26	LTFD	1: Latch current speed data instead of COUNTER 3.
27	LTOF	1: Stop latching by hardware timing. (Effective only for software)
28	CU1L	1: Immediately after latching COUNTER 1, reset COUNTER 1.
29	CU2L	1: Immediately after latching COUNTER 2, reset COUNTER 2.
30	CU3L	1: Immediately after latching COUNTER 3, reset COUNTER 3.
31	ISMR	1: Stop automatically reset function after reading REST and RIST register and reset by writing into REST and RIST register.

Note 1: When COUNTER3 (setting as a deviation counter) is selected as the comparison counter, the LSI compares the counted absolute value and the comparator data. (Absolute value range: 0 to 32,767.)

Note 2: When you specify C1S0 to 2 = 110 (positive software limit) or C2S0 to 2 = 110 (negative software limit), select COUNTER 1 (specified position) as the comparison counter.
When the software limit is set, the motor will stop, regardless of the settings on C1D0 to 1 and C2D0 to 1.
(When deceleration stop is selected, the motor will decelerate and stop when it is started with high-speed.)

Note 3: When C3S0 to 3 is set to 1000 to 1010 (synchronous signal output), select COUNTER 3 (setting as a general-purpose counter for the comparison counter. The other counters cannot be selected.
To set the comparator, select a positive value.

5-4-4-6. RENV5 register

This is a register for the Environment 5 settings. It is primarily used to set feed amount correction data.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSTP	0	0	ADJ	BR11	BR10	BR9	BR8	BR7	BR6	BR5	BR4	BR3	BR2	BR1	BR0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PMG4	PMG3	PMG2	PMG1	PMG0	PD10	PD9	PD8	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0

Bit	Bit name	Description
0 to 11	BR0 to 11	Enter a backlash correction amount. (Setting range: 0 to 4095)
12	ADJ	Select a feed amount correction method. 00: Turn OFF the correction function. 01: Backlash correction
13 to 14	Not defined	(Always set to 0.)
15	PSTP	1: Even if a stop command is written, the G9103A will operate for the number of pulses that are already input on the PA/PB.
16 to 26	PD0 to 10	Specify the division ratio for pulses on the PA/PB input. The number of pulses is divided using the set value/2048. When 0 is entered, the division circuit will be OFF. (= 2048/2048) [Setting range: 0 to 2,047]
27 to 31	PMG0 to 4	Specify the magnification rate for pulses on the PA/PB input. The number of pulses is multiplied by the set value + 1. [Setting range: 0 to 31]

5-4-4-7. RENV6 register

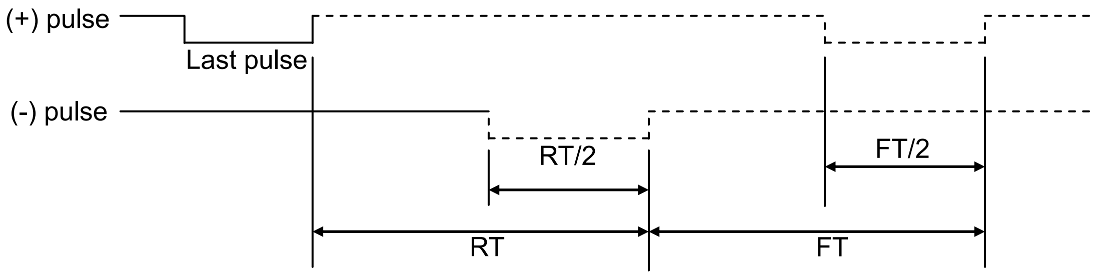
This is a register for the Environment 6 settings. It is primarily used to enter time for the vibration reduction function. If both RT and FT data are other than zero, the vibration reduction function is turned ON.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RT15	RT14	RT13	RT12	RT11	RT10	RT9	RT8	RT7	RT6	RT5	RT4	RT3	RT2	RT1	RT0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FT15	FT4	FT13	FT12	FT11	FT10	FT9	FT8	FT7	FT6	FT5	FT4	FT3	FT2	FT1	FT0

Bit	Bit name	Description
0 to 15	RT0 to 15	Enter the RT time shown in the figure below. [Setting range: 0 to 65,535] The setting unit is 1.6 μs.
16 to 31	FT0 to 15	Enter the FT time shown in the figure below. [Setting range: 0 to 65,535] The setting unit is 1.6 μs.

The dotted lines in the figure below are pulses added by the vibration reduction function.



Set the time [RT, FT] = Enter a value x 1.6 (μs)

5-4-4-8. RCUN1 register

This is a register used for COUNTER 1 (command position counter).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
&	&	&	&																												

This is a counter used exclusively for command pulses.
Setting rage: -134,217,728 to +134,217,727.

5-4-4-9. RCUN2 register

This is a register used for COUNTER 2 (mechanical position counter).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
&	&	&	&																												

Counter 2 can count three types of pulses: Command pulses, encoder signals (EA/EB input) and pulsar signals (PA/PB input). Setting range: -134,217,728 to +134,217,727.

5-4-4-10. RCUN3 register

This is a register used for COUNTER 3 (deviation, general-purpose counter).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
&	&	&	&	&	&	&	&	&	&	&	&	&	&	&	&																

Counter 3 can count three types of deviations: between command pulses and encoder signals, between command pulses and pulsar signals, and between encoder signals and pulsar signals.
Setting range: -32,768 to +32,767.
Counter 3 will not count values exceeding the setting and it shows the maximum value.

5-4-4-11. RCMP1 register

Specify the comparison data for Comparator 1.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
&	&	&	&																												

Setting range: -134,217,728 to +134,217,727.

5-4-4-12. RCMP2 register

Specify the comparison data for Comparator 2.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
&	&	&	&																												

Setting range: -134,217,728 to +134,217,727.

Note 1: Bits marked with an "*" (asterisk) will be ignored when written and are 0 when read.
Note 2: Bits marked with an "&" symbol will be ignored when written and will be the same value as the upper most bit among bits having no marks when read. (Code extension)

5-4-4-13. PRCP3 (RCMP3) register

Specify the comparison data for Comparator 3. PRCP3 is the pre-register for RCMP.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
&	&	&	&																												

Setting range: -134,217,728 to +134,217,727.

5-4-4-14. RIRQ register

This register is to set event interrupt cause.
Set bits that you want to enable event interrupts, to 1.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IRNM	IRNP	IRNA	IRSA	IRSD	IROL	IRLT	IRCL	IRC3	IRC2	IRC1	IRDE	IRDS	IRUE	IRUS	IREN

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	IRBE

Bit	Bit name	Description
0	IREN	When the axis stops normally.
1	IRUS	When the axis starts acceleration.
2	IRUE	When the axis ends acceleration.
3	IRDS	When the axis starts deceleration.
4	IRDE	When the axis ends deceleration.
5	IRC1	When Comparator 1 conditions are met.
6	IRC2	When Comparator 2 conditions are met.
7	IRC3	When Comparator 3 conditions are met.
8	IRCL	When the count value is reset by a CLR signal input.
9	IRLT	When the count value is latched by an LTC signal input.
10	IROL	When the count value is latched by an ORG signal input.
11	IRSD	When the SD input is ON.
12	IRSA	When the STA input is ON.
13	IRNA	When the motor starts by the start command (2x01(h)) of broadcast communication.
14	IRNP	When the motor stops by the stop command (2x02(h)) of broadcast communication.
15	IRNM	When writing into the pre-register for operation is ready (MSTS.SPRF changes from 1 to 0.)
16	IRBE	When completing the current operation while the pre-register is not fixed. (When the next operation data is not set in time while continuous interpolation is performed.)
17 to 31	Not defined	(Always set to 0.)

5-4-4-15. RLTC1 register

Latched data for COUNTER 1 (command position). (Read only.)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
&	&	&	&																												

The contents of COUNTER 1 are copied when triggered by the LTC, an ORG input, or an LTCH command.
Data range: -134,217,728 to +134,217,727.

5-4-4-16. RLTC2 register

Latched data for COUNTER2 (mechanical position). (Read only.)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
&	&	&	&																												

The contents of COUNTER2 are copied by the LTC, an ORG input, or an LTCH command.
Data range: -134,217,728 to +134,217,727.

5-4-4-17. RLTC3 register

Latched data for COUNTER 3 (deviation counter) or current speed. (Read only.)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
$	$	$	$	$	$	$	$	$	$	$	$	$	$	$	%																

The contents of COUNTER 3 or the current speed are copied by LTC, ORG input, or LTCH command.
When RENV4.LTFD=0, the register latches the COUNTER 3 data. When RENV4.LTFD=1, the register latches the current speed.
When the LTFD is 1 and movement on the axis is stopped, the latched data will be 0.
Data range when LTFD is 0: -32,768 to +32,767.
Data range when LTDF is 1: 0 to 100,000.

When the G9103A latches COUNTER 3 data (RENV4.LTFD=0), bits shown as "$" and "%" will have the same code extension as bit 15.
When the G9103A latches the current speed data (RENV4.LTFD=1), bits shown as "$" will become "0," and the lower 17 bits with "%" will contain the current speed data.

Note 1: Bits marked with an "*" (asterisk) will be ignored when written and are 0 when read.
Note 2: Bits marked with an "&" symbol will be ignored when written and will be the same value as the upper most bit among bits having no marks when read. (Sign extension)

5-4-4-18. RSTS register

The extension status can be checked. (Read only.)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SERC	SPCS	SEMG	SSTP	SSTA	SDIN	SSD	SORG	SMEL	SPEL	SALM	SDIR	CND3	CND2	CND1	CND0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PFM1	PFM0	PFC1	PFC0	SPH4	SPH3	SPH2	SPH1	SPLS	SCP3	SCP2	SCP1	SINP	SLTC	SCLR	SEZ

Bit	Bit name	Description
0 to 3	CND0 to 3	These bits report the operation status. 0000: Stopping 0001: Waiting for STA input 0010: Waiting for a completion of ERC timer 0011: Waiting for a completion of direction change timer 0100: Correcting backlash 0101: Waiting for PA/PB input 0110: Feeding at FA constant speed 0111: Feeding at FL constant speed 1000: Accelerating 1001: Feeding at FH constant speed. 1010: Decelerating 1011: Waiting for INP input or reverse braking 1100: Not defined 1101: Not defined 1110: Not defined 1111: Others (controlling start)
4	SDIR	Operation direction (0: Positive direction. 1: Negative direction.)
5	SALM	This bit becomes 1 when the ALM input signal is ON.
6	SPEL	This bit becomes 1 when the +EL input signal is ON.
7	SMEL	This bit becomes 1 when the -EL input signal is ON.
8	SORG	This bit becomes 1 when the ORG input signal is ON.
9	SSD	This bit becomes 1 when the SD input signal is ON. (SD latch status.)
10	SDIN	This bit becomes 1 when the SD input signal is ON. (SD terminal input status.)
11	SSTA	This bit becomes 1 when the $\overline{\text{STA}}$ input signal is ON.
12	SSTP	This bit becomes 1 when the $\overline{\text{STP}}$ input signal is ON.
13	SEMG	This bit becomes 1 when the $\overline{\text{EMG}}$ input signal is ON.
14	SPCS	This bit becomes 1 when the PCS input signal is ON.
15	SERC	This bit becomes 1 when the ERC input signal is ON.
16	SEZ	This bit becomes 1 when the EZ input signal is ON.
17	SCLR	This bit becomes 1 when the CLR input signal is ON.
18	SLTC	This bit becomes 1 when the LTC input signal is ON.
19	SINP	This bit becomes 1 when the INP input signal is ON.
20	SCP1	This bit becomes 1 when the CMP1 comparison conditions are met.
21	SCP2	This bit becomes 1 when the CMP2 comparison conditions are met.
22	SCP3	This bit becomes 1 when the CMP3 comparison conditions are met.
23	SPLS	This bit becomes 1 when the pulse output (±) is ON. Note 1
24	SPH1	This bit becomes 1 when the PH1 excitation signal output is H level.
25	SPH2	This bit becomes 1 when the PH2 excitation signal output is H level.
26	SPH3	This bit becomes 1 when the PH3 excitation signal output is H level.
27	SPH4	This bit becomes 1 when the PH4 excitation signal output is H level.
28 to 29	PFC0 to 1	Monitor the usage of the pre-register for comparator 3. Note.2
30 to 31	PFM0 to 1	Monitor the usage of the pre-register for operation. Note 2.

Note 1: Logical sum output of the OUT/DIR signals. When the 90 degree phase difference signal output is selected, the G9103A monitors the original pulse output.

Note 2:00:Both register and pre-register are unfixed.
01:Register is fixed and pre-register is unfixed.
1x:Both register and pre-register are fixed. (New writing is impossible.)

5-4-4-19. REST register

Reading this register, you can confirm an error interrupt cause
The corresponding bit will be "1" when that an error interrupt has occurs.
This register is reset by the following process.

1. When RENV4.ISMR=0 (initial status).
   This register is reset automatically by reading this register and is also reset by writing data that only bits to be reset are 1
2. When RENV4.ISMR=1.
   This register is reset by writing data that only bits to be reset are 1. It is reset by writing data that had read.

Note. When RENV4.ISMR=0, if automatic retry occurs because of communication error in the communication to read this register, data is read again and the information after reset is sent again. Therefore, we recommend the setting RENV4.ISMR=1.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ESDT	ESPE	ESEE	ESOR	0	ESNT	ESPO	ESSD	ESEM	ESSP	ESAL	ESML	ESPL	ESC3	ESC2	ESC1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	ESAO	ESIP

Bit	Bit name	Description
0	ESC1	Stopped when Comparator 1 conditions had met. (+SL)
1	ESC2	Stopped when Comparator 2 conditions had met. (-SL)
2	ESC3	Stopped when Comparator 3 conditions had met.
3	ESPL	Stopped by the +EL input being turned ON.
4	ESML	Stopped by the -EL input being turned ON.
5	ESAL	Stopped by the $\overline{\text{ALM}}$ input being turned ON.
6	ESSP	Stopped by the $\overline{\text{STP}}$ input being turned ON.
7	ESEM	Stopped by the $\overline{\text{EMG}}$ input being turned ON.
8	ESSD	Decelerated and stopped by the SD input being turned ON.
9	ESPO	An overflow occurred in the PA/PB input buffer counter.
10	ESNT	Stopped by watchdog timer with communication error. (Communication line disconnection)
11	Not defined	(Always set to 0.)
12	ESOR	Position override could not be executed.
13	ESEE	When EA and EB input changed simultaneously.
14	ESPE	When PA and PB input changed simultaneously.
15	ESDT	Simultaneously stopped by data error for interpolation operation (Note.1)
16	ESIP	Simultaneously stopped by abnormal stop of other axes.
17	ESAO	Stopped by exceeding the circular interpolation range (incremental position >28 bits)
18	EFAJ	A clock synchronous error occurred
19 to 31	Not defined	(Always set to 0.)

Note.1 :When the operation is started by the following setting in circular interpolation.

1. Central coordinate (RIP, RIPY) is (0,0).
2. Central coordinate (RIP, RIPY) is the same as the endpoint coordinate (RMV, RMVY).

5-4-4-20. RIST register

Reading this register, you can confirm an error interrupt cause
The corresponding bit will be "1" when that an event interrupt has occurs.
This register is reset by the following process.

3. When RENV4.ISMR=0 (initial status).
   This register is reset automatically by reading this register and is also reset by writing data that only bits to be reset are 1
4. When RENV4.ISMR=1.
   This register is reset by writing data that only bits to be reset are 1. It is reset by writing data that had read.

Note. When RENV4.ISMR=0, if automatic retry occurs because of communication error in the communication to read this register, data is read again and the information after reset is sent again. Therefore, we recommend the setting RENV4.ISMR=1.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ISNM	ISNP	ISNA	ISSA	ISSD	ISOL	ISLT	ISCL	ISC3	ISC2	ISC1	ISDE	ISDS	ISUE	ISUS	ISEN

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	ISBE

Bit	Bit name	Description
0	ISEN	Stopped automatically.
1	ISUS	Starting acceleration.
2	ISUE	Ending acceleration.
3	ISDS	Starting deceleration.
4	ISDE	Ending deceleration.
5	ISC1	When the comparator 1 conditions are met.
6	ISC2	When the comparator 2 conditions are met.
7	ISC3	When the comparator 3 conditions are met.
8	ISCL	When the count value is reset by a CLR signal input.
9	ISLT	When the count value is latched by an LTC input.
10	ISOL	When the count value is latched by an ORG input.
11	ISSD	When the SD input is ON.
12	ISSA	When the STA input is ON.
13	ISNA	When the start of broadcast communication is input.
14	ISNP	When the stop of broadcast communication is input.
15	ISNM	When the pre-register for the next operation is ready for writing.
16	ISBE	When completing the current operation while the pre-register is not fixed. (When RSTS.PFM turns to 0).
17 to 31	Not defined	(Always set to 0.)

5-4-4-21. RPLS register

This register is used to check the value of the positioning counter (number of pulses left for feeding). (Read only.)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0																												

At the start, this value will be the absolute value in RMV register. Each pulse that is output will decrease this value by one.

5-4-4-22. RSPD register

This register is used to check the EZ count value and the current speed. (Read only.)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AS15	AS14	AS13	AS12	AS11	AS10	AS9	AS8	AS7	AS6	AS5	AS4	AS3	AS2	AS1	AS0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	IDC2	IDC1	IDC0	ECZ3	ECZ2	ECZ1	ECZ0	0	0	0	AS16

Bit	Bit name	Description
0 to 16	AS0 to 16	Read the current speed as a step value (same units as for RFL and RFH). When stopped the value is 0.
17 to 19	Not defined	(Always set to 0.)
20 to 23	ECZ0 to 3	Read the count value of EZ input that is used for an origin return.
24 to 26	IDC0 to 2	Read the idling count value.
27 to 31	Not defined	(Always set to 0.)

5-4-4-23. RSDC register

This register is used to check ramping-down point value that is calculated automatically, for the positioning operation. (Read only.)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0																								

5-4-4-24. RSYN register

This register is used to set the information synthesized between the numeral G9103As.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	SYNE	DNSTP						SYON	SYNC	DNMST					

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FAM7	FAM6	FAM5	FAM4	FAM3	FAM2	FAM1	FAM0	FAWL	FAL6	FAL5	FAL4	FAL3	FAL2	FAL1	FAL0

Bit	Bit name	Description
0 to 5	DNMST	This bit to set device address of the master G9103A for synchronizing clock for motor control.
6	SYNC	1: Enable the DNMST setting. (Synchronize clock for control)
7	SYON	1: Clock for motor control is synthesized. (This bit is read only and ignored when written.)
8 to 13	DNSTP	These bits are to set the device address of other G9103A to monitor for simultaneous stop with other axes.
14	SYNE	1:Enable the DNSTP setting.
15	not defined	(Always set to 0.)
16 to 22	FAL0 to 6	These bits are to set the limit o f frequency correction. When the correction frequency is over the limit, this condition is regarded as a synchronized error. However, when 0 is set, this function to detect errors turns OFF.
23	FAWL	1: When frequency fluctuation range is more than plus or minus 7, this condition is regarded as a synchronized error.
24 to 31	FAM0 7	These bits are to monitor the frequency correction amount for clock synchronization. (Read only)

For the detail, see “8-17. Synchronous function with other axes”

5-4-4-25. RGNO register (32 bits)

This register is a general-purpose register and does not affect motor control. This register is used to set company ID and model ID. Company ID is managed by NPM and model ID is managed by customers.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Company ID																				Model ID											

For the details, see “8-18. Unit ID control function”.

5-4-4-26. RGN1 to 3 register (each 32 bits)

This register is a general-purpose register and does not affect motor control.
This register can be used to transfer the data between G9103A and CPU by serial bus

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

For the details, see “8-19. CPU connection function”.

6. Operation Mode

Specify the basic operation mode using the MOD area (bits 0 to 6) in RMD (operation mode) register.

6-1. Continuous operation mode using command control

This is a mode of continuous operation. After a start command is written, operation continues until a stop command is written.

MOD	Operation method	Direction of movement
00(h)	Continuous operation by a command	Positive direction
08(h)	Continuous operation by a command	Negative direction

This mode is to stop by turning ON the EL signal corresponding to the direction of operation.
When operation direction is positive, +EL can be used. When operation direction is negative, -EL is used.
In order to start operation in the reverse direction after stopping the motion by turning ON the EL signal, a new start command must be written.

6-2. Positioning operation mode

The following seven operation types are available for positioning operations.

MOD	Operation method	Direction of movement
41(h)	Positioning operation (specify the target increment position)	Positive direction when RMV > 0 Negative direction when RMV < 0
42(h)	Positioning operation (specify the absolute position in COUNTER1)	Positive direction when RMV > COUNTER1 Negative direction when RMV < COUNTER1
43(h)	Positioning operation (specify the absolute position in COUNTER2)	Positive direction when RMV > COUNTER2 Negative direction when RMV < COUNTER2
44(h)	Return operation to command position 0 (COUNTER1)	Positive direction when COUNTER1 < 0 Negative direction when COUNTER1 > 0
45(h)	Return operation to machine position 0 (COUNTER2)	Positive direction when COUNTER2 < 0 Negative direction when COUNTER2 > 0
46(h)	One pulse operation	Positive direction
4E(h)	One pulse operation	Negative direction
47(h)	Timer operation	

6-2-1. Positioning operation (specify a target position using an incremental value) (MOD: 41(h))

This is a positioning mode used by placing a value in the RMV (target position) register.
The feed direction is determined by the sign set in the RMV register.
When starting, the RMV register setting is loaded into the positioning counter (RPLS). The positioning counter counts down with each pulse output and the G9103A stops feeding when the counter reaches 0.
When you set the RMV register value to 0 and start a positioning operation, the LSI will stop immediately without outputting pulses.

6-2-2. Positioning operation (specify the absolute position in COUNTER 1) (MOD: 42(h))

This mode only uses the difference between the RMV (target position) register value and COUNTER 1.
Since the COUNTER 1 value is stored when starting to move, the target position cannot be overridden by changing the COUNTER 1 value. But, the target position can be overridden by changing the RMV value.
The direction of movement can be set automatically by evaluating the relative relationship between the RMV register setting and the value in COUNTER 1.
At start up, the difference between the RMV setting and the value stored in COUNTER 1 is loaded into the positioning counter (RPLS). The positioning counter counts down with each pulse output, and when the positioning counter value reaches zero, it stops operation.
If the RMV register value is made equal to the COUNTER 1 value and the positioning operation is started, the G9103A will immediately stop operation without outputting any command pulses.

6-2-3. Positioning operation (specify the absolute position in COUNTER 2) (MOD: 43(h))

This mode only uses the difference between the RMV (target position) register setting and the value in COUNTER 2.
Since the COUNTER 2 value is stored when starting a positioning operation, the target position cannot be overridden by changing the value in COUNTER 2; however, it can override the target position by changing the value in RMV.
The direction of movement can be set automatically by evaluating the relationship between the RMV register setting and the value in COUNTER 2.
At start up, the difference between the RMV setting and the value stored in COUNTER 2 is loaded into the positioning counter (RPLS). The positioning counter counts down with each pulse output, and when the positioning counter value reaches zero, it stops operation.
If the RMV register value is made equal to the COUNTER 2 value and the positioning operation is started, the G9103A will immediately stop operation without outputting any command pulses.
Also, this operation does not use feedback control. So, if encoder signals are input to COUNTER 2, the value of COUNTER 2 at the completion of the feed may be different from the target position.

6-2-4. Command position 0 return operation (MOD: 44(h))

This mode continues operation until the COUNTER 1 (command position) value becomes zero.
The direction of movement is set automatically by the sign for the value in COUNTER 1 when starting.
This operation is the same as when positioning (specify the absolute position in COUNTER 1) by entering zero in the RMV register; however, there is no need to specify RMV register.

6-2-5. Machine position 0 return operation (MOD: 45(h))

This mode is used to continue operations until the value in COUNTER 2 (mechanical position) becomes zero.
The direction of movement is set automatically by the sign for the value in COUNTER 2 when starting.
This operation is the same as when positioning (specify the absolute position in COUNTER 2) by entering zero in the RMV register. However, there is no need to specify RMV register.

6-2-6. One pulse operation (MOD: 46(h), 4E(h))

This mode outputs a single pulse.
This operation is identical to a positioning operation (incremental target positioning) that writes a "1" (or "-1") to RMV register. However, with this operation, there is no need to specify RMV register.

6-2-7. Timer operation (MOD: 47(h))

This mode allows the internal operation time to be used as a timer.
The internal effect of this operation is identical to the positioning operation. However, the LSI does not output any pulses (they are masked).
Therefore, the internal operation time using the constant speed start command will be a product of the frequency of the output pulses and the RMV register setting. (Ex.: When the frequency is 1000 pps and the RMV register is set to 120 pulses, the internal operation time will be 120 msec.)
Write a positive number (1 to 134,217,727) into RMV register.
The ±EL input signal, SD input signal, and software limits are ignored. (These are always treated as OFF.)
The ALM input signal, STP input signal, and EMG input signals are effective.
The backlash correction, vibration restriction function, and change direction timer function stop.
The count of COUNTER 1 (command position) stops.
Regardless of RMD.MINP setting, an operation complete delay controlled by the INP signal will not occur.
In order to eliminate errors in the internal operation time, set RMD.METM=0 and use the cycle completion timing of the output pulse as the operation complete timing.

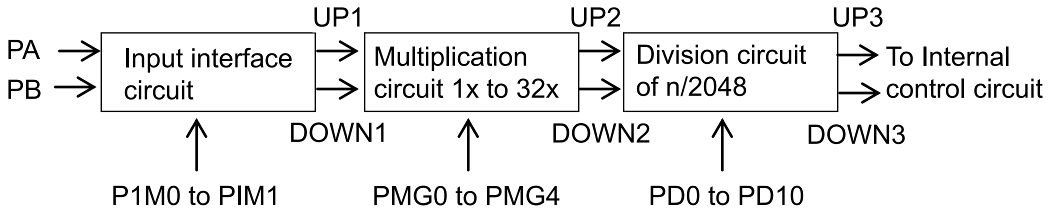
6-3. Pulsar (PA/PB) input mode

This mode is used to allow operations from a pulsar input.
In order to enable pulsar input, set RENV2.POFF=0.
It is also possible to apply a filter on the PA/PB input.
After writing a start command, when a pulsar signal is input, the LSI will output pulses to OUT terminal.
Use an FL constant speed start (STAFL: 0050(h)) or an FH constant speed start (STAFH: 0051(h)) as a start command.
Four methods are available for inputting pulsar signals through the PA/PB input terminal by setting RENV2 (environmental setting 2) register.

- Input a 90 degree phase difference signal (1x, 2x, or 4x).
- Input Two pulse signal (Up pulse and down pulse).

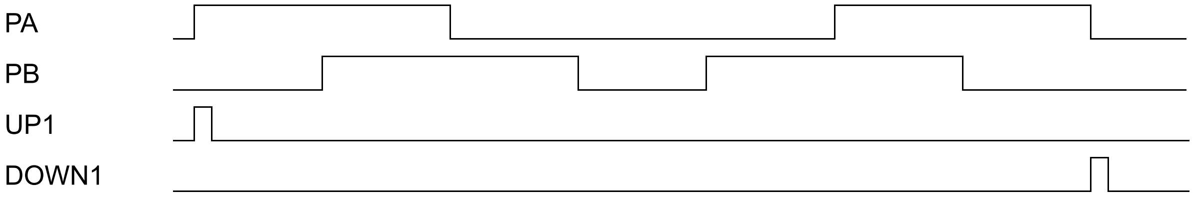
Note: The backlash correction function is available with the pulsar input mode. However, reversing pulsar input while the backlash correction is unavailable.

Besides the above 1x to 4x multiplication, the G9103A has a multiplication circuit of 1x to 32x and division circuit of (1 to 2048)/2048. For setting the multiplication from 1x to 32x, specify RENV5.PMG0 to PMG4 and for setting the division of n/2048, specify RENV5.PD0 to PD10.

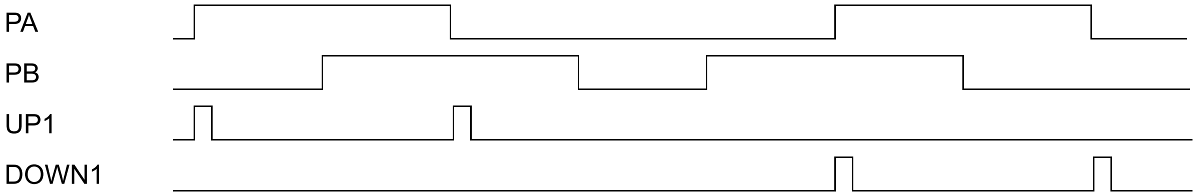


The timing of the UP1 and DOWN1 signals will be as follows by setting RENV2.PIM0 to PIM1.

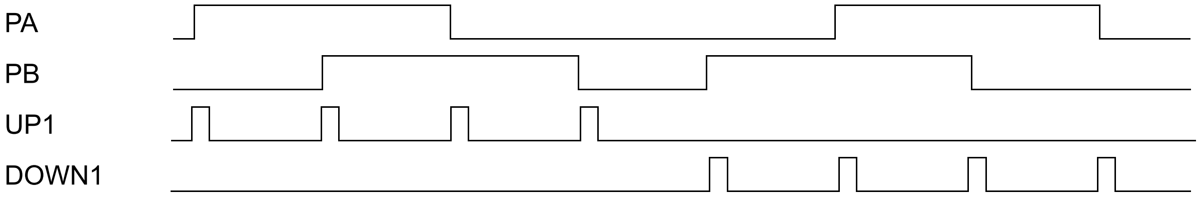
1) When using 1x input of 90 degree phase difference signals (PIM = 00)



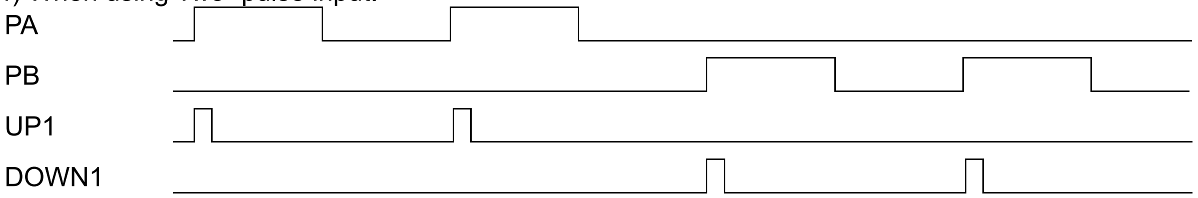
2) When using 2x input of 90 degree phase difference signals (PIM = 01)



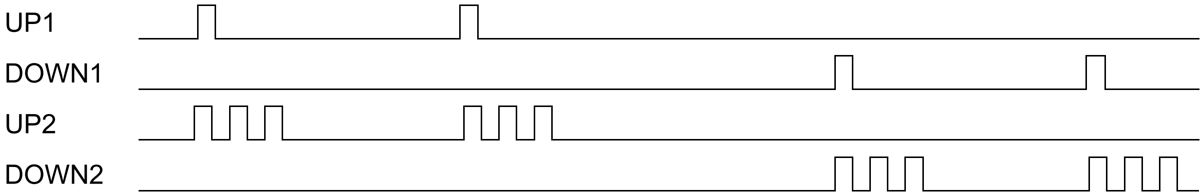
3) When using 4x input of 90 degree phase difference signals (PIM = 10)



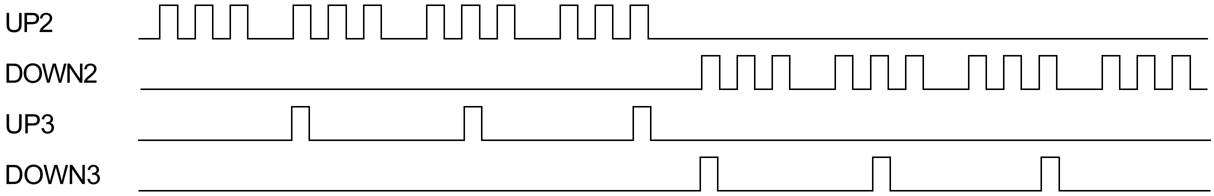
4) When using Two- pulse input.



When 3x (RENV5.PMG = 2) is set as the multiplication rate of 1x to 32x multiplication circuit, operation timing will be as follows.



When 512/2048 (RENV5.PD =512) is set as the divided rate of n/2048 division circuit, operation timing will be as follows.



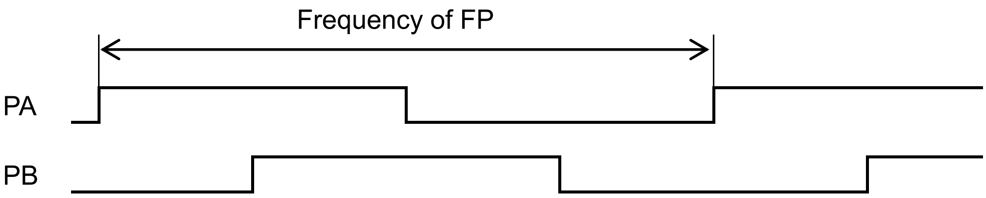
The pulsar input mode is started by an FL constant speed start command (0050(h)) or by an FH constant speed start command (0051(h)).
Pulsar input causes to output pulses that are omitted from the FL speed or FH speed pulse outputs. Therefore, there may be a difference in the timing between the pulsar input and output pulses, up to an internal pulse frequency period.

The maximum input frequency for pulsar signals (FP) is restricted by the FL speed when an FL constant speed start is used and by the FH speed when an FH constant speed start is used. The LSI generates an interrupt as errors when both the PA and PB inputs change simultaneously, or when the input frequency exceeded and the input/output buffer counter (16-bit) overflows. This can be monitored by REST (error interrupt cause) register.

FP<(speed)/(input I/F phase magnification value) / (PMG setting value+1) / (PD setting value/2048),
PD setting value ≠ 0
FP<(speed)/(input I/F phase magnification value) / (PMG setting value +1)
PD setting value = 0

<Examples of the relationship between the FH (FL) speed [pps] and the pulsar input frequency FP [pps]>

PA/PB input I/F setting	PMG setting value	PD setting value	Usable range
Two-pulse input	0 (1x)	0	FP < FH (FL)
	0 (1x)	1024	FP < FH (FL) x 2
	2 (3x)	0	FP < FH (FL) / 3
90 degree phase difference 1x	0 (1x)	0	FP < FH (FL)
	0 (1x)	1024	FP < FH (FL) x 2
	2 (3x)	0	FP < FH (FL) / 3
90 degree phase difference 2x	0 (1x)	0	FP < FH (FL) / 2
	0 (1x)	1024	FP < FH (FL)
	2 (3x)	0	FP < FH (FL) / 6
90 degree phase difference 4x	0 (1x)	0	FP < FH (FL) / 4
	0 (1x)	1024	FP < FH (FL) / 2
	2 (3x)	0	FP < FH (FL) / 6



Note: When the PA/ PB input frequency fluctuates, take the shortest frequency, not average frequency, as "Frequency of FP" above.

<Setting relationship of PA/PB input>

Notes 1. When describing register bit in the right column of the following table, “n” refers to the bit position and “0” refers to a bit position where it is prohibited to write any value except zero and the bit will always be zero when read.

Specify the PA/PB input <RENV2.PIM0 to 1(bit 14 to 15)> 00: 90 degree phase difference, 1x 10: 90 degree phase difference, 4x 01: 90 degree phase difference, 2x 11: Up and down pulse (Two-pulse) input	[RENV2] (WRITE) 15 8 n n - - - - - -
Specify the PA/PB input count direction <RENV2.PDIR(bit 16)> 0: Count forward when the PA phase is leading or on the rising edge of PA. 1: Count forward when the PB phase is leading or on the rising edge of PB	[RENV2] (WRITE) 23 16 - - - - - - - n
Enable/disable PA/PB input <RENV2.POFF (bit 18)> 0: Enable PA/PB input 1: Disable PA/PB input.	[RENV2] (WRITE) 23 16 - - - - - n - -
Set PA/PB input filter <RENV2.PINF(bit 13)> 0: Insert a filter on PA/PB input By setting the filter, the G9103A ignores signals shorter than 150ns.	[RENV2] (WRITE) 15 8 - - n - - - - -
Reading operation status <RSTS.CND (bit 0 to 3)> 0101: Wait for PA/ PB input.	[RSTS] (READ) 7 0 - - - - n n n n
Reading PA/PB input error <REST.ESPE (bit 14) > ESPE (bit 14) = 1: Occurs a PA/PB input error	[REST] (READ) 15 8 - n - - - - - -
Reading PA/PB input buffer counter (16 bit) status <REST.ESP0 (bit 9)> ESPO (bit 9) = 1: Overflow occurs. (When PA/PB frequency is higher than operation speed and buffer counter overflows.)	[REST] (READ) 15 8 - - - - - - n -

The pulsar input mode has the following 14 operation types.
The direction of movement for continuous operation can be changed by setting RENV2.PDIR, without changing the wiring connections for the PA/PB inputs.

MOD	Operation mode	Direction of movement
01(h)	Continuous operation synchronized with PA/PB (pulsar).	Determined by the PA/PB input.
51(h)	Positioning operation (incremental position) synchronized with PA/PB.	Determined by the sign of the RMV value.
52(h)	Positioning operation (COUNTER 1 absolute position) synchronized with PA/PB.	Determined by the relationship of the RMV and COUNTER 1 values.
53(h)	Positioning operation (COUNTER 2 absolute position) synchronized with PA/PB.	Determined by the relationship of the RMV and COUNTER 2 values.
54(h)	Zero return operation of command position (COUNTER 1) synchronized with PA/PB.	Determined by the sign of the value in COUNTER 1.
55(h)	Aero return operation of mechanical position (COUNTER 2) synchronized with PA/PB.	Determined by the sign of the value in COUNTER 2.
68(h)	Continuous linear interpolation synchronized with PA/PB (output the X axis pulse)	Determined by the direction of interpolation.
69(h)	Linear interpolation synchronized with PA/PB (output the X axis pulse)	Determined by the direction of interpolation.
6C(h)	CW directional circular linear interpolation synchronized with PA/PB (output the X axis pulse)	Determined by the direction of interpolation.
6D(h)	CCW directional circular interpolation synchronized with PA/PB (output the X axis pulse)	Determined by the direction of interpolation.
78(h)	Continuous linear interpolation synchronized with PA/PB (output the Y axis pulse)	Determined by the direction of interpolation.
79(h)	Linear interpolation synchronized with PA/PB (output the Y axis pulse)	Determined by the direction of interpolation.
7C(h)	CW directional circular interpolation synchronized with PA/PB (output the Y axis pulse)	Determined by the direction of interpolation.
7D(h)	CCW directional circular interpolation synchronized with PA/PB (output the Y axis pulse)	Determined by the direction of interpolation.

6-3-1. Continuous operation synchronized with PA/PB (MOD: 01(h))

This mode allows continuous operation using a pulsar input.
When PA/PB signals are input after writing a start command, the LSI will output pulses to OUT terminal.
The feed direction depends on the PA/PB signal input method and the value set in RENV2.PDIR.

PA/PB input method	PDIR	Feed direction	PA/PB input
90 degree phase difference signal (1x, 2x, and 4x)	0	Positive direction	When the PA phase leads the PB phase.
		Negative direction	When the PB phase leads the PA phase.
	1	Positive direction	When the PB phase leads the PA phase.
		Negative direction	When the PA phase leads the PB phase.
Two-pulse input	0	Positive direction	PA input rising edge.
		Negative direction	PB input rising edge.
	1	Positive direction	PB input rising edge.
		Negative direction	PA input rising edge.

The G9103A stops operation when the EL signal in the current feed direction is turned ON. But the G9103A can be operated in the opposite direction without writing a restart command.
When stopped by the EL input, no error interrupt (MSTS.SERR) will occur.
To release the operation mode, write an immediate stop command (0049(h)).

Note: When the "immediate stop command (0049(h))" is written while the G9103A is performing a multiplication operation (caused by setting PIM 0 to 1 and PMG 0 to 4), the G9103A will stop operation immediately and the total number of pulses that are output may not be an integer multiple of the magnification. When RENV5.PSTP is set to 1, the G9103A delays the stop timing until an integer multiple of pulses has been output.
However, if RENV5.PSTP=1, the G9103A maintains the stop command pending status, regardless of the operation mode selected. When MSTS.SBSY = 0, the stop command will be disabled. When using PA/PB input operation and RENV5.PSTP = 1, check the main status before writing a stop command. When MSTS.SBSY=0, return to RENV5.PSTP=0 and write the stop command.
However, after a stop command is sent by setting RENV5.PSTP to 1, If MSTS.SBSY=0, set to RENV5.PSTP=0. (When PSTP=1 and SBSY=0 , the G9103A will continue the stop command pending status.)

6-3-2. Positioning operations synchronized with PA/PB (specify incremental position) (MOD: 51(h))

The G9103A positioning is synchronized with the pulsar input by using the RMV setting as incremental position data.
The feed direction is determined by the sign in the RMV (target position) register.
When the RMV register value is loaded to the position counter at start and PA/PB signals are input, the LSI outputs pulses and the positioning counter counts down. When the value in the positioning counter reaches zero, movement on the axis will stop and any further PA/ PB input will be ignored.
Set the RMV register value to zero and start the positioning operation. The LSI will stop movement on the axis immediately, without outputting any command pulses.

Note. When operation is complete, stop interrupt (MSTS.SEND) is not output.
If interrupt is needed, please use normal stop of event interrupt (RIRQ.IREN).

6-3-3. Positioning operation synchronized with PA/PB (specify the absolute position to COUNTER 1) (MOD: 52(h))

The G9103A positioning is synchronized with the pulsar input by using the RMV setting as the absolute value for COUNTER 1.
The direction of movement is determined by the relationship between the value in RMV and the value in COUNTER 1.
When starting, the difference between the values in the RMV and COUNTER 1 is loaded into the positioning counter. When a PA/PB signal is input, the G9103A outputs pulses and the positioning counter counts down. When the value in the positioning counter reaches "0," the G9103A ignores any further PA/PB input. If you try to start with RMV = COUNTER 1, the G9103A will not output any pulses and it will stop immediately.

Note. When operation is complete, stop interrupt (MSTS.SEND) is not output.
If interrupt is needed, please use normal stop of event interrupt (RIRQ.IREN).

6-3-4. Positioning operation synchronized with PA/PB (specify the absolute position in COUNTER 2) (MOD: 53(h))

The operation procedures are the same as MOD= 52(h), except that this function uses COUNTER 2 instead of COUNTER 1.

6-3-5. Command position zero return operation synchronized with PA/PB (MOD: 54(h))

This mode is used to feed the axis using a pulsar input until the value in COUNTER1 (command position) becomes zero. The number of pulses output and the feed direction are set automatically by internal calculation, using the COUNTER1 value when starting.
Set the COUNTER1 value to zero and start the positioning operation, the LSI will stop movement on the axis immediately, without outputting any command pulses.

Note. When operation is complete, stop interrupt (MSTS.SEND) is not output.
If interrupt is needed, please use normal stop of event interrupt (RIRQ.IREN).

6-3-6. Mechanical position zero return operation synchronized with PA/PB (MOD: 55(h))

Except for using COUNTER2 instead of COUNTER1, the operation details are the same as for MOD = 54(h).

6-3-7. Interpolated operation synchronized with PA/PB

This mode is used to operate interpolation synchronized with a pulsar input. The feed direction is the same as the normal interpolation. Even If a pulse is reversed, the direction of interpolation is not changed. For the detail of interpolated operation, see “6-7. Interpolation operation mode”.

Note. When operation is complete, stop interrupt (MSTS.SEND) is not output.
If interrupt is needed, please use normal stop of event interrupt (RIRQ.IREN).

6-4. Origin position operation mode

The following six origin position operation modes are available.

MOD	Operation mode	Direction of movement
10(h)	Origin return operation	Positive direction
18(h)	Origin return operation	Negative direction
12(h)	Leaving the origin position operation	Positive direction
1A(h)	Leaving the origin position operation	Negative direction
15(h)	Origin search operation	Positive direction
1D(h)	Origin search operation	Negative direction

Depending on the operation method, the origin position operation uses the ORG, EZ, or ±EL inputs.
Specify the input logic of the ORG input signal in the RENV1 (environment 1) register. This register's terminal status can be monitored with the RSTS (extension status) register.
Specify the input logic of the EZ input signal in the RENV2 (environment 2) register. Specify the number for EZ to count for an origin return complete condition in the RENV3 (environment 3) register. This register's terminal status can be monitored by reading RSTS register.
Specify the logic for the ±EL input signal using the ELL input terminals. Specify the operation to execute when the signal turns ON (immediate stop/deceleration stop) in the RENV1 register. This register's terminal status can be monitored with RSTS (extension status) register.
An input filter can be applied to the ORG input signal and ±EL input signal by setting the RENV1 register.

Set the ORG signal input logic <RENV1.ORGL (bit 7)> 0: Negative logic 1: Positive logic	[RENV1] (WRITE) 7 0 n - - - - - - -
Read the ORG signal <RSTS.SORG (bit 8)> 0: Turn OFF the ORG signal 1: Turn ON the ORG signal	[RSTS] (READ) 15 8 - - - - - - - n
Set the EZ signal input logic <RENV2.EZL (bit 12)> 0: Falling edge 1: Rising edge	[RENV2] (WRITE) 15 8 - - - n - - - -
Set the EZ count <RENV3.EZD0 to 3 (bits 4 to 7)> Specify the number for EZ to count that will indicate an origin return complete condition. Enter the value (the count minus 1) in EZD0 to EZD3. Setting range: 0 to 15.	[RENV3] (WRITE) 7 0 n n n n - - - -
Read the EZ signal <RSTS.SEZ (bit 16)> 0: Turn OFF the EZ signal 1: Turn ON the EZ signal	[RSTS] (READ) 23 16 - - - - - - - n
Set the ±EL signal input logic <ELL input terminal> L: Positive logic input H: Negative logic input	
Specify a method for stopping when the ±EL signal turns ON <RENV1.ELM (bit 3)> 0: Immediate stop when the ±EL signal turns ON. 1: Deceleration stop when the ±EL signal turns ON.	[RENV1] (WRITE) 7 0 - - - - n - - -

Read the ±EL signal <RSTS.SPEL (bit 6), SMEL (bit 7)>
SPEL=0: Turn OFF + EL signal SPEL=1: Turn ON + EL signal
SMEL=0: Turn OFF - EL signal SMEL=1: Turn ON - EL signal
	[RSTS] (READ)
	7 0
	n n - - - - - -

Applying an input filter to the ±EL and ORG inputs <RENV1.FLTR (bit 25)> 0: Apply a filter to the ±EL and ORG inputs. By applying a filter, pulses shorter than 4 μsec will be ignored.	[RENV1] (WRITE) 31 24 - - - - - - n -

6-4-1. Origin return operation

After writing a start command, the axis will continue feeding until an origin return complete condition are met.
MOD: 10h Positive direction origin return operation
18h Negative direction origin return operation
When an origin return is completed, the LSI will reset the counter at the origin position and output an ERC (deviation counter clear) signal while the axis is stopping.
The RENV3 register is used to set the basic origin return method and whether or not to reset the counter when the origin return is completed. Specify whether or not to output the ERC signal in the RENV1 register. For details about the ERC signal, see 8-6-2, "ERC signal."

Set the an origin return method <RENV3.ORM0 to 3 (bits 0 to 3)>

[RENV3] (WRITE)

7							0
-	-	-	-	n	n	n	n

0000: Origin return operation 0
- The axis will stop immediately (or make a deceleration stop when feeding at high speed) when the ORG input turns ON.
- COUNTER reset timing: When the ORG input turns ON.

0001: Origin return operation 1
- The axis will stop immediately (or decelerate and stop when feeding at high speed) when the ORG input turns ON. Then, it feeds in the opposite direction at RFA constant speed until the ORG input turns OFF. Then, the axis will move back in the original direction at RFA speed and stop immediately when the ORG input turns ON again.
- COUNTER reset timing: When the ORG input signal turns ON.

0010: Origin return operation 2
- After the ORG input turns ON when feeding at constant speed, the LSI will start counting EZ pulses. The axis will stop immediately when the LSI finishes counting the specified number of EZ pulses.
  After the ORG input turns ON when feeding at high speed, the axis will start decelerating. At the same time, the LSI will start counting the EZ pulses. When the LSI finishes counting the specified number of EZ pulses, the axis will stop immediately.
- COUNTER reset timing: When finishing counting the specified number of EZ pulses.

0011: Origin return operation 3
- After the ORG signal turns ON when feeding at constant speed, the LSI will start counting EZ pulses. The axis will stop immediately when the LSI finishes counting the specified number of EZ pulses. After the ORG signal turns ON when feeding at high speed, the LSI will start counting EZ pulses. When the LSI finishes counting the specified number of EZ pulses, the axis will decelerate and stop.
  COUNTER reset timing: When finishing counting the specified number of EZ pulses.

0100: Origin return operation 4
- After the ORG input turns ON when feeding at constant speed , the axis will stop immediately (or decelerate and stop when feeding at high speed). Then, the axis will start feeding in the opposite direction at RFA constant speed. After the ORG input turns OFF, the LSI will start counting EZ pulses. After the LSI finishes counting the specified number of EZ pulses, the axis will stop immediately...
- COUNTER reset timing: When finishing counting the specified number of EZ pulses.

0101: Origin return operation 5

- After the ORG input turns ON when feeding at constant speed, the axis will stop immediately (or decelerate and stop when feeding at high speed). Then, the axis will start feeding in the opposite direction. After the ORG input turns OFF, the LSI will start counting EZ pulses. After the LSI finishes counting the specified number of EZ pulses, the axis will stop immediately (or decelerate and stop when feeding at high speed).
- COUNTER reset timing: When finishing counting the specified number of EZ pulses.

0110: Origin return operation 6

- After the EL input turns ON when feeding at constant speed, the axis will stop immediately (or decelerate and stop when ELM is 1). Then, the axis will start feeding in the opposite direction at RFA constant speed. When the EL signal turns OFF, the axis will stop immediately when the LSI finishes counting the specified number of EZ pulses.
- COUNTER reset timing: When the EL input is OFF.

0111: Origin return operation 7

- After the EL signal turns ON when feeding at constant speed, the axis will stop immediately (or decelerate and stop when ELM is 1). Then, the axis will start feeding in the opposite direction at RFA constant speed. After the EL signal turns OFF, the LSI will start counting EZ pulses. After the LSI finishes counting the specified number of EZ pulses, the axis will stop immediately.
- COUNTER reset timing: When stopped by finishing counting the specified number of EL pulses.

1000:Origin return operation 8

- After the EL signal turns ON when feeding at constant speed, the axis will stop immediately (or decelerate and stop when ELM is 1). Then, the axis will start feeding in the opposite direction at RFL constant speed. After the EL signal turns OFF, the LSI will start counting EZ pulses. After the LSI finishes counting the specified number of EZ pulses, the axis will stop immediately.
- CONTER reset timing: When finishing counting the specified number of EZ signal.

1001: Origin return operation 9

- After the process in origin return operation 0 has executed, the motor operates until COUNTER2 = 0.

1010: Origin return operation 10

- After the process in origin return operation 3 has executed, the motor operates until COUNTER2 = 0.

1011: Origin return operation 11

- After the process in origin return operation 5 has executed, the motor operates until COUNTER2 = 0.

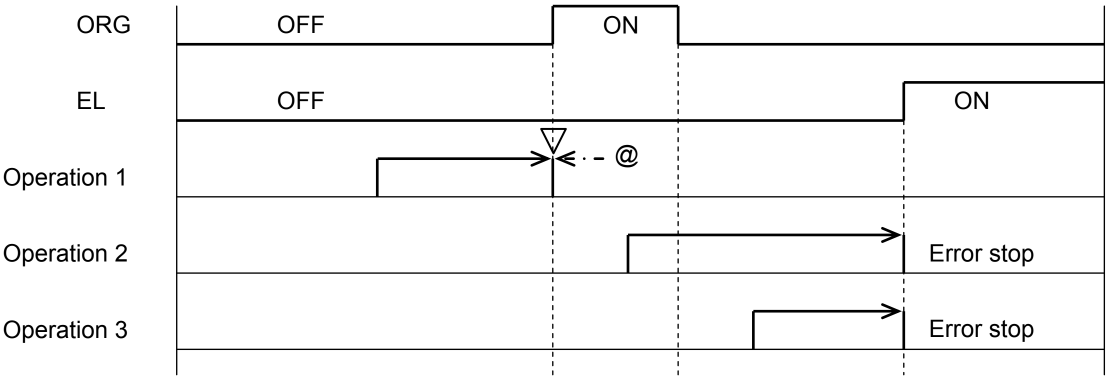
1100: Origin return operation 12

- After the process in origin return operation 8 has executed, the motor operates until COUNTER2 = 0).

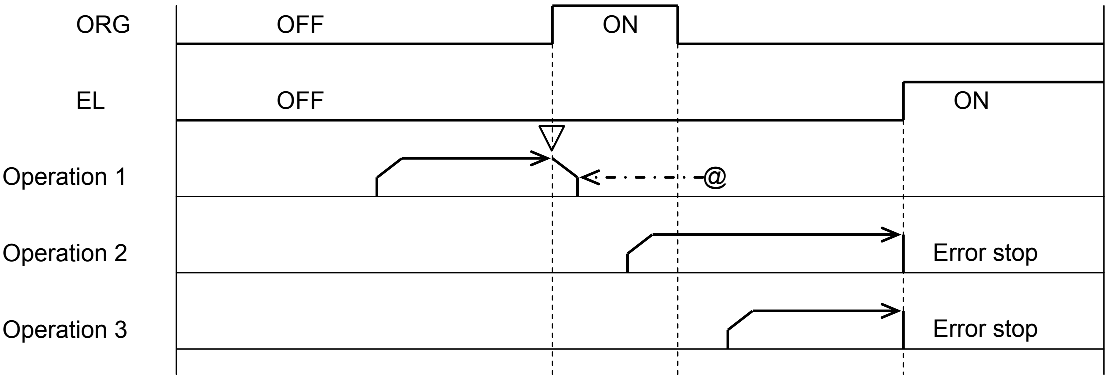
Settings after an origin return is completed <RENV3.CU1R to 3R (bits 20 to 22)> CU1R (bit 20) =1: Reset COUNTER 1 (command position) CU2R (bit 21) =1: Reset COUNTER 2 (mechanical position) CU3R (bit 22) =1: Reset COUNTER 3 (general-purpose, deviation counter)	[RENV3] (WRITE) 23 16 - n n n - - - -
Setting the ERC signal for automatic output <RENV1.EROR (bit 11)> 0: G9103A does not output an ERC signal when a zero return is completed. 1:G9103A outputs an ERC signal automatically when a zero return is completed.	[RENV1] (WRITE) 15 8 - - - - n - - -
Setting the counter reset at the origin position <RENV3.CU1R (bit 20)> 0: G9103A does not reset COUNTER 1 at the origin position. 1: G9103A resets Counter 1 at the origin position in origin return operation.	[RENV3] (WRITE) 23 16 - - - n - - - -
Setting the counter reset at the origin position <RENV3.CU2R (bit 21)> 0: G9103A does not reset COUNTER 2 at the origin position. 1: G9103A resets Counter 2 at the origin position in origin return operation.	[RENV3] (WRITE) 23 16 - - n - - - - -
Setting the counter reset at the origin position <RENV3.CU3R (bit 22)> 0: G9103A does not reset COUNTER 3 at the origin position. 1: G9103A resets Counter 3 at the origin position in origin return operation.	[RENV1] (WRITE) 23 16 - n - - - - - -

6-4-1-1. Origin return operation 0 (ORM = 0000)

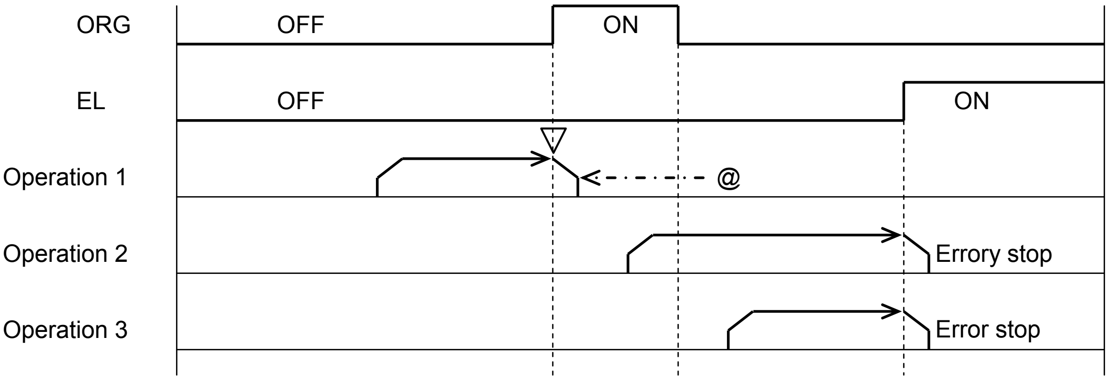
□ Constant speed operation <Sensor: EL (ELM = 0), ORG>
[Starting from here, □ indicates constant speed operation, and ■ indicates high speed operation.]



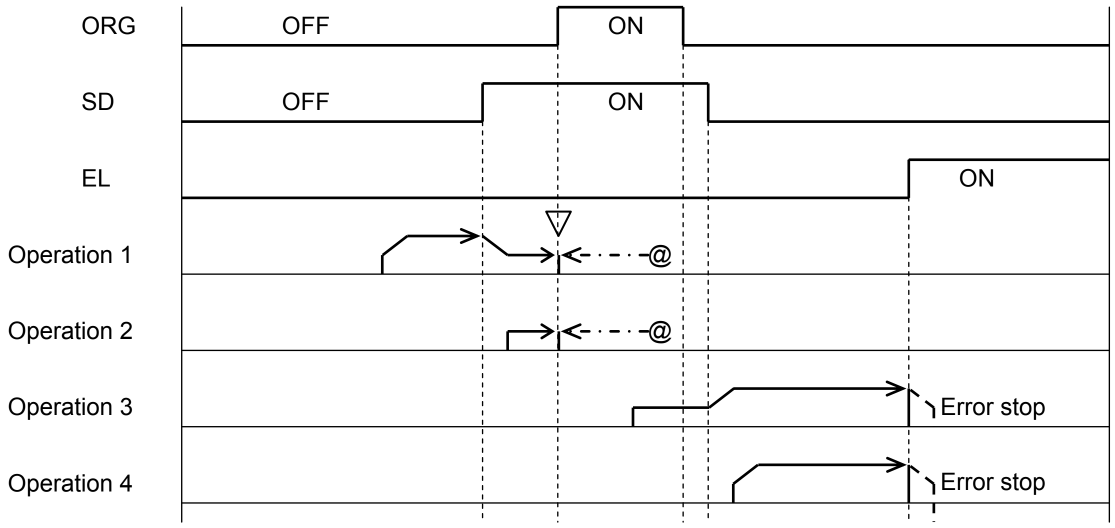
■ High speed operation <Sensor: EL (ELM = 0), ORG>
Even if the axis stops normally, it may not be at the origin position. However, COUNTER 2 (mechanical position) provides a reliable value



■ High speed operation <Sensor: EL (ELM = 1), ORG>
Even if the axis stops normally, it may not be at the origin position. However, COUNTER 2 (mechanical position) provides a reliable value.



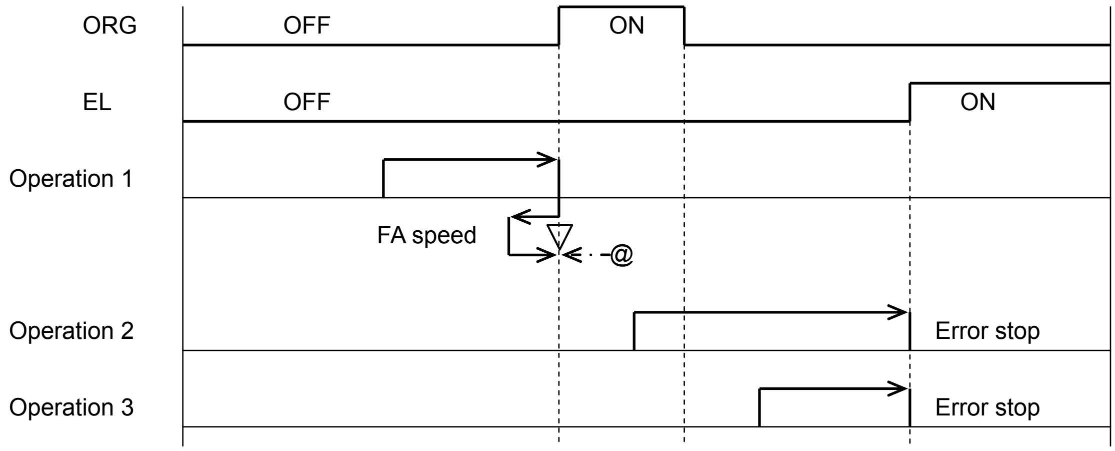
■ High speed operation <Sensor: EL (ELM = 1), SD (SDM = 0, SDLT = 0), ORG>



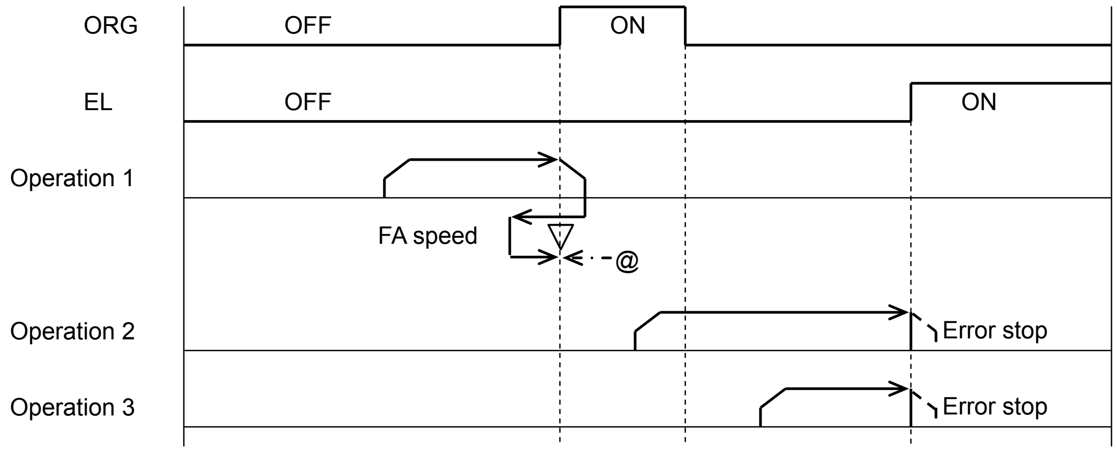
Note: Positions marked with ▽ reflect the counter reset timing and @ reflects the ERC signal output timing.

6-4-1-2. Origin return operation 1 (ORM=0001)

□ Constant speed operation <Sensor: EL (ELM = 0), ORG>

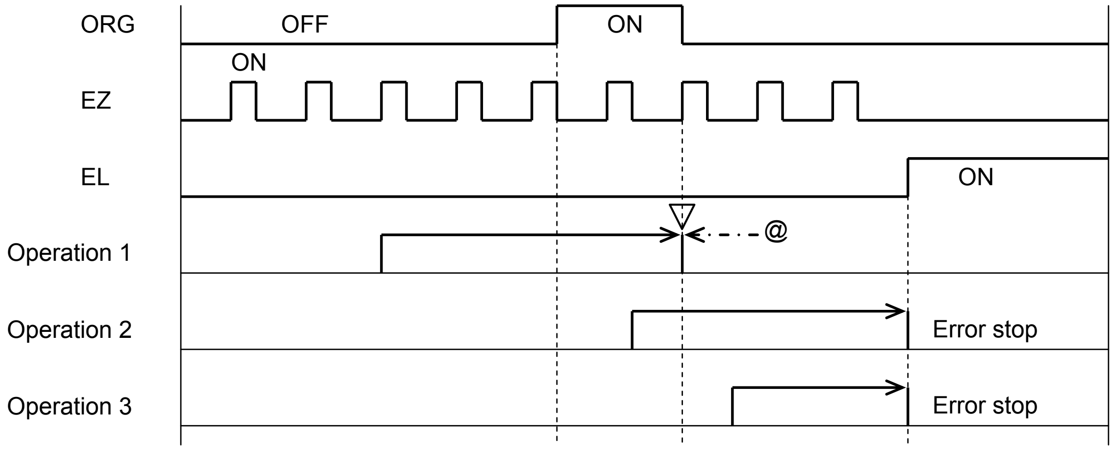


■ High speed operation <Sensor: EL, ORG>

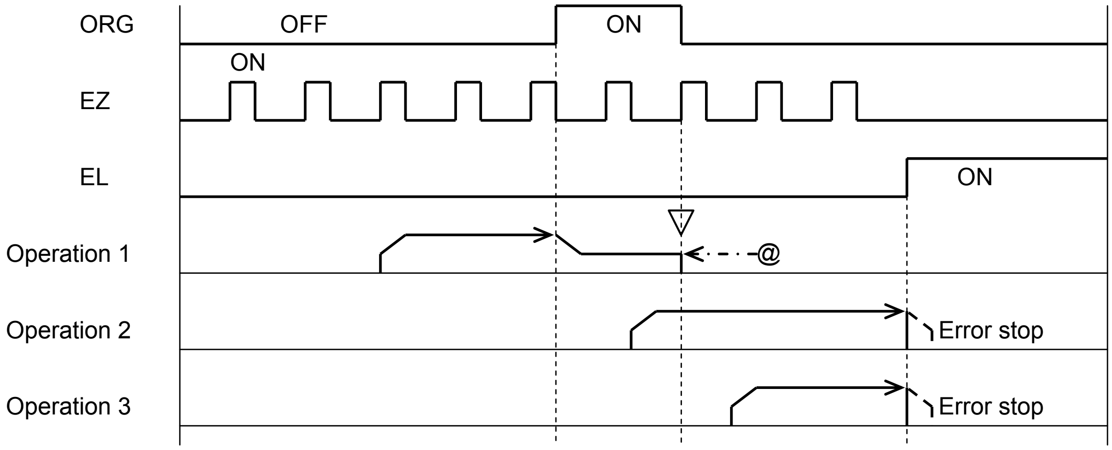


6-4-1-3. Origin return operation 2 (ORM = 0010)

□ Constant speed operation <Sensor: EL (ELM = 0), ORG, EZ (EZD = 0001)>



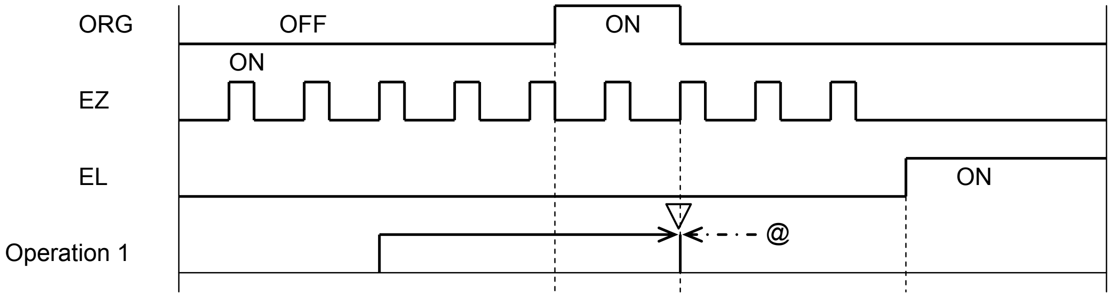
■ High speed operation <Sensor: EL, ORG, EZ (EZD = 0001)>



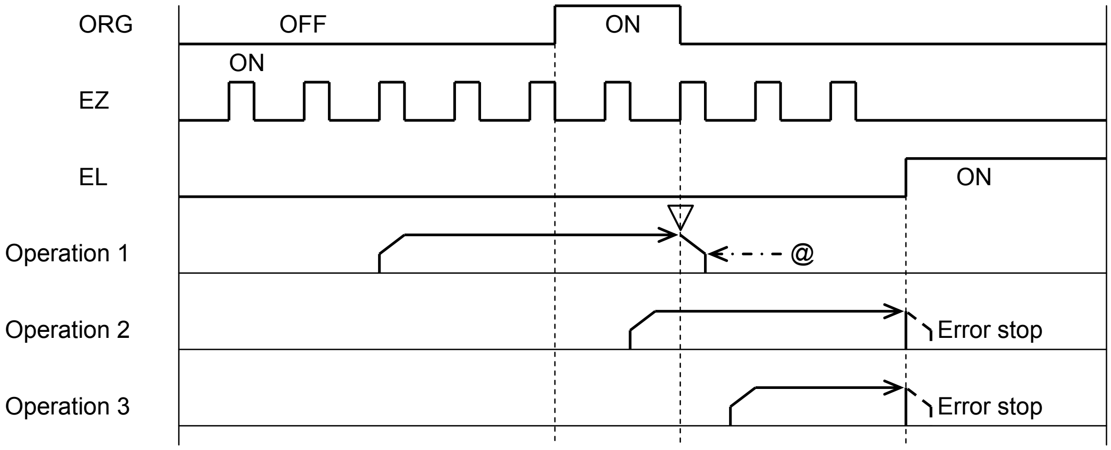
Note: Positions marked with▽ reflect the counter reset timing and @ reflects the ERC signal output timing.

6-4-1-4. Origin return operation 3 (ORM = 0011)

□ Constant speed operation <Sensor: EL, ORG, EZ (EZD = 0001)>

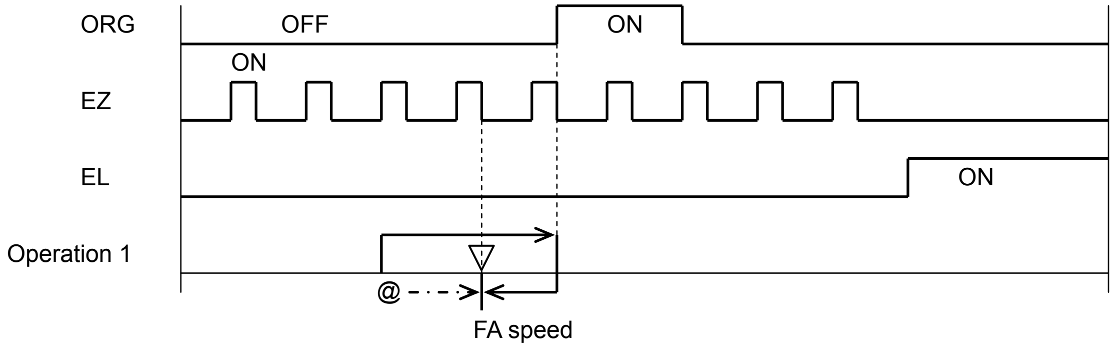


■ High speed operation <Sensor: EL,ORG, EZ (EZD = 0001)>

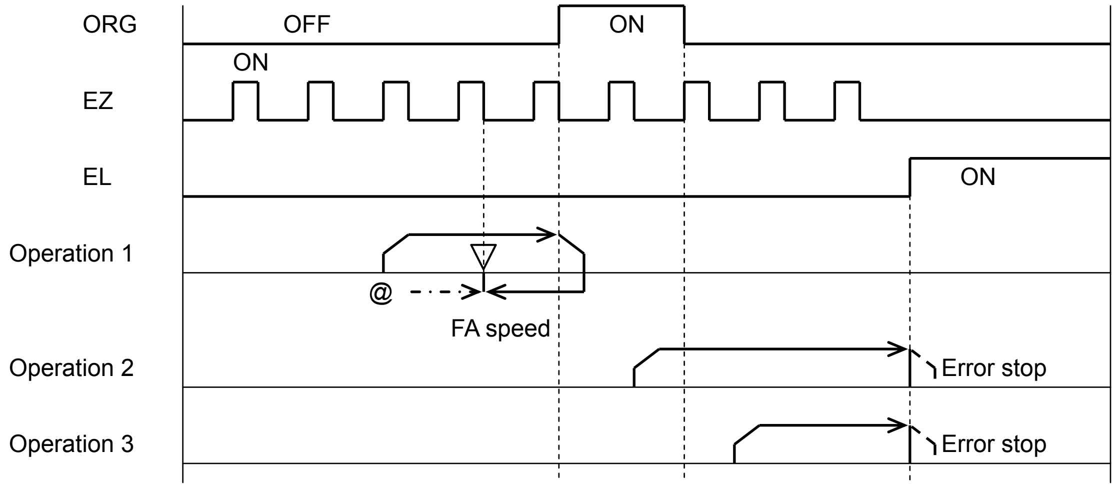


6-4-1-5. Origin return operation 4 (ORM = 0100)

□ Constant speed operation <Sensor: EL, ORG, EZ (EZD = 0001)>



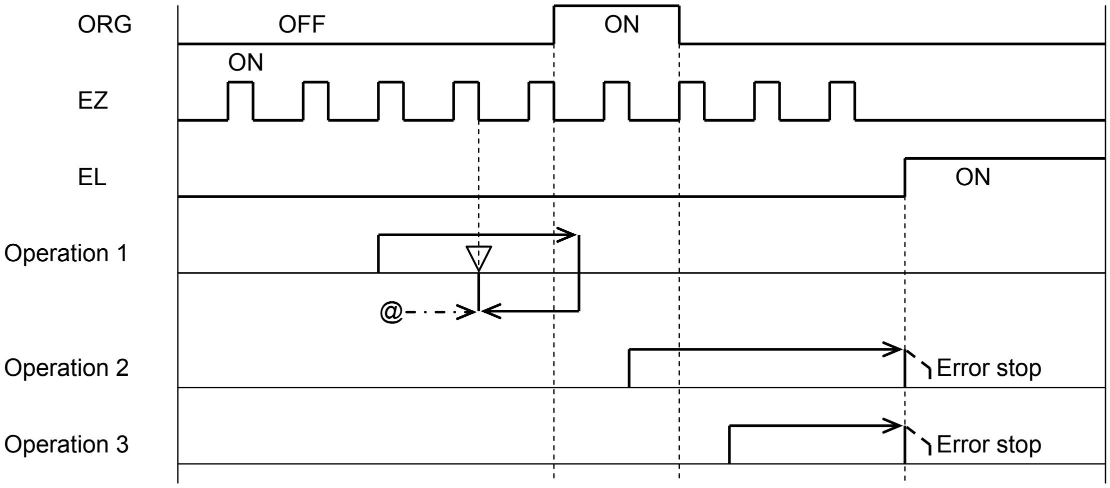
■ High speed operation <Sensor: EL, ORG, EZ (EZD = 0001)>



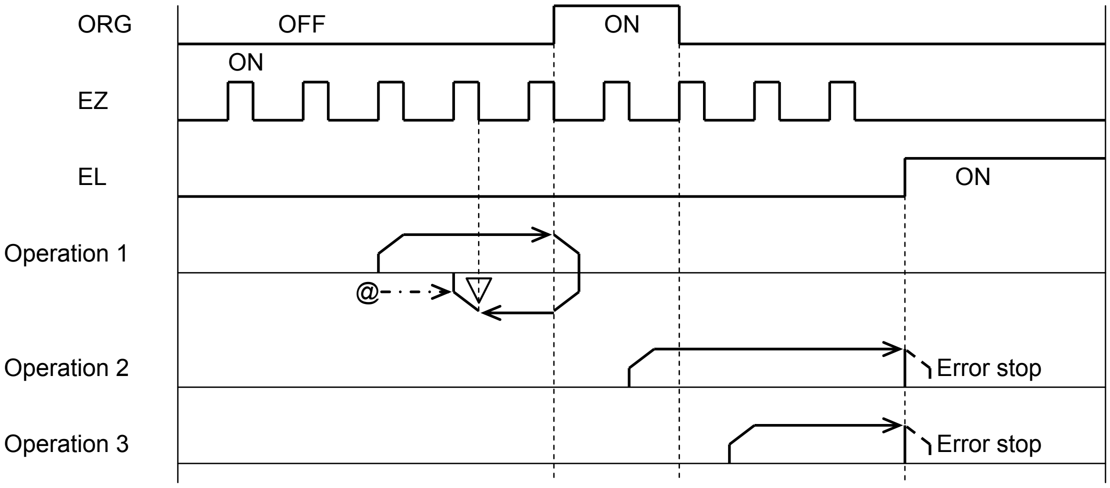
Note: Positions marked with ∇ reflect the counter reset timing and @ reflects the ERC signal output timing.

6-4-1-6. Origin return operation 5 (ORM = 0101)

□ Constant speed operation <Sensor: EL, ORG, EZ (EZD = 0001)>

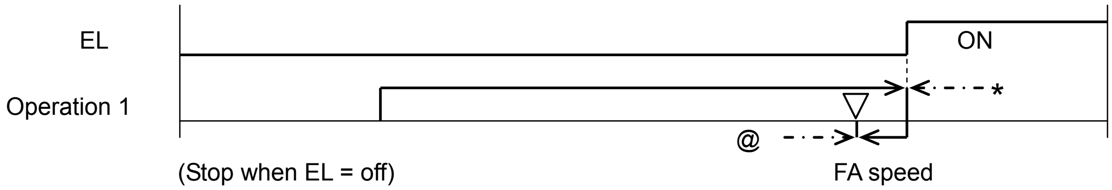


■ High speed operation <Sensor: EL, ORG, EZ (EZD = 0001)>

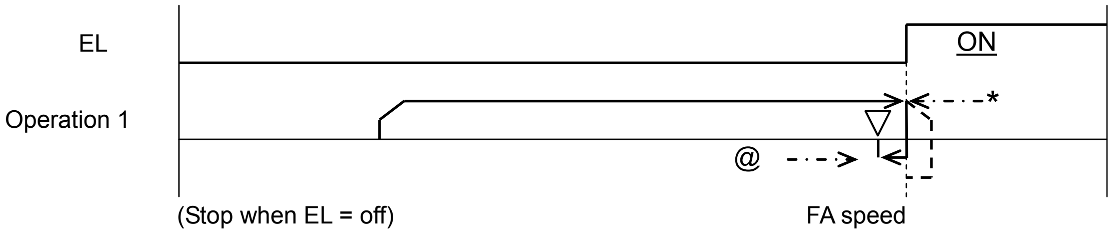


6-4-1-7. Origin return operation 6 (ORM = 0110)

□ Constant speed operation <Sensor: EL>



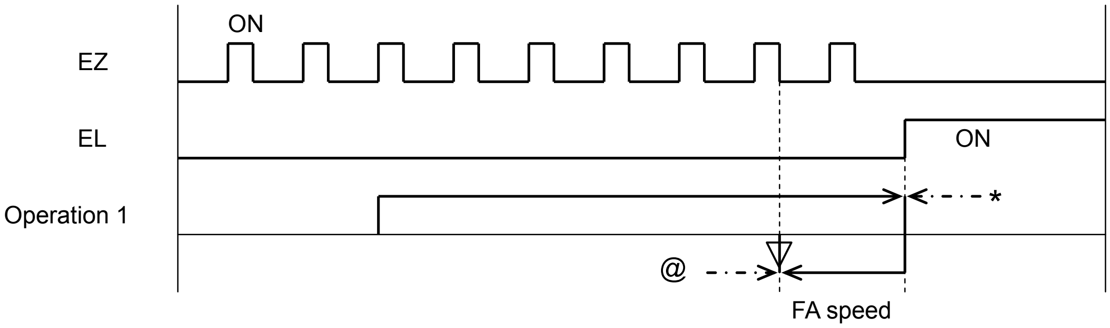
■ High speed operation <Sensor: EL>



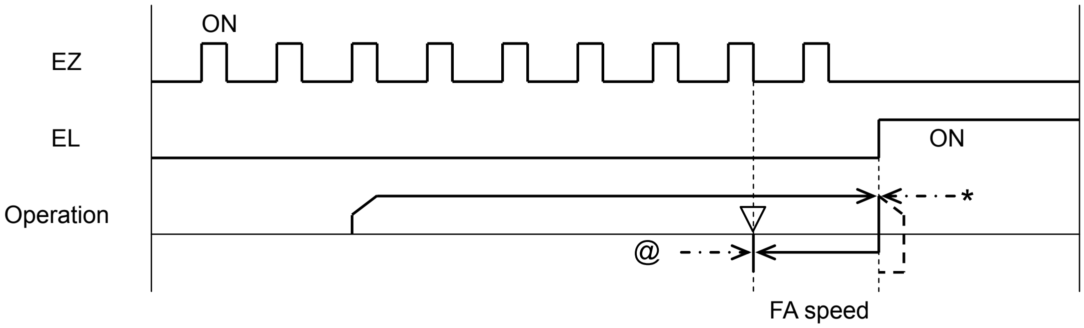
Note: Positions marked with ∇ reflect the counter reset timing and @ reflects the ERC signal output timing. Also, when RENV1.EROE=1 and RENV1.ELM=0, the LSI will output an ERC signal at positions marked with an asterisk (*).

6-4-1-8. Origin return operation 7 (ORM = 0111)

□ Constant speed operation <Sensor: EL, EZ (EZD = 0001)>

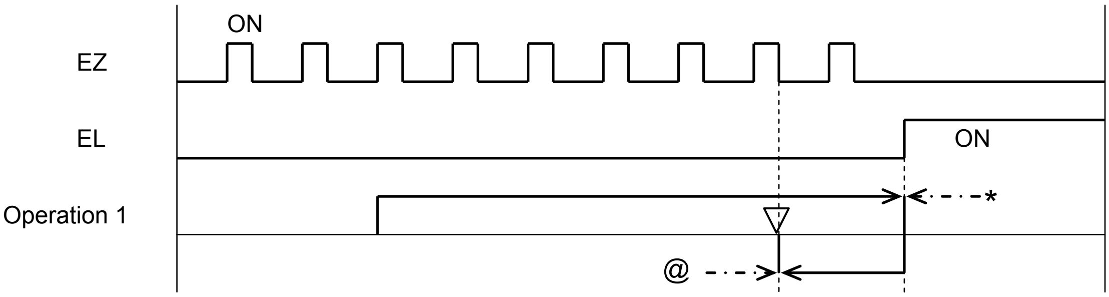


■ High speed operation <Sensor: EL, EZ (EZD = 0001)>

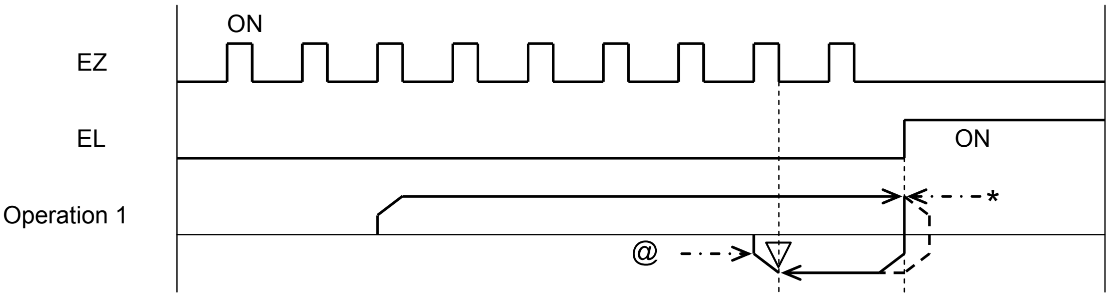


6-4-1-9. Origin return operation 8 (ORM=1000)

□ Constant speed operation <Sensor: EL, EZ (EZD = 0001)>

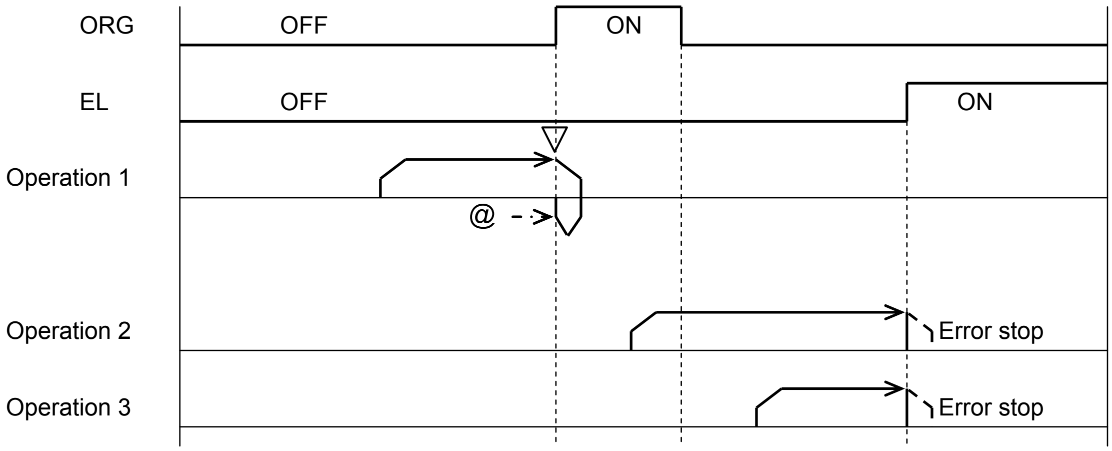


■ High speed operation <Sensor: EL, EZ (EZD = 0001)>



6-4-1-10. Origin return operation 9 (ORM = 1001)

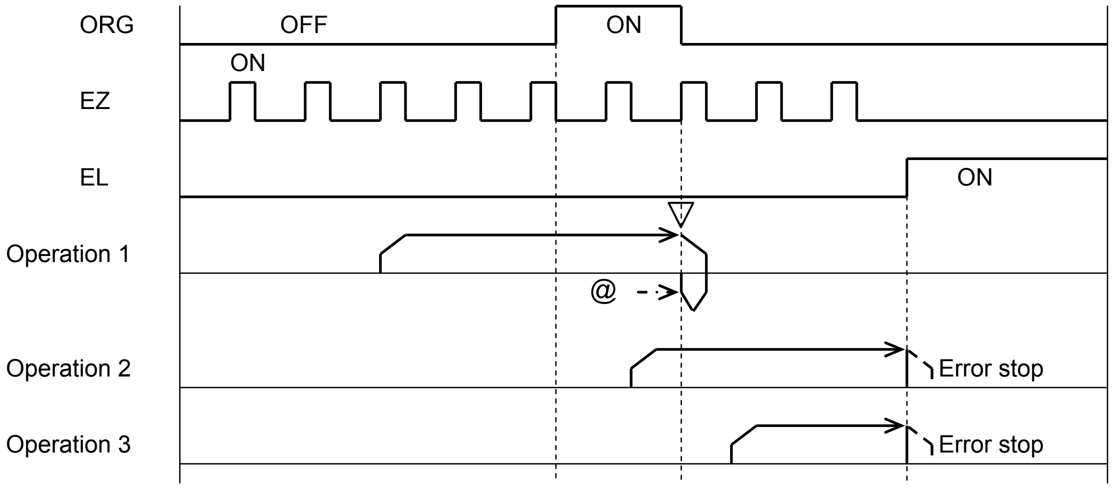
■ High speed operation <Sensor: EL, ORG)>



Note: Positions marked with ∇ reflect the counter reset timing and @ reflects the ERC signal output timing. Also, when RENV1.EROE=1 and RENV1.ELM=0, the LSI will output an ERC signal at positions marked with an asterisk (※).

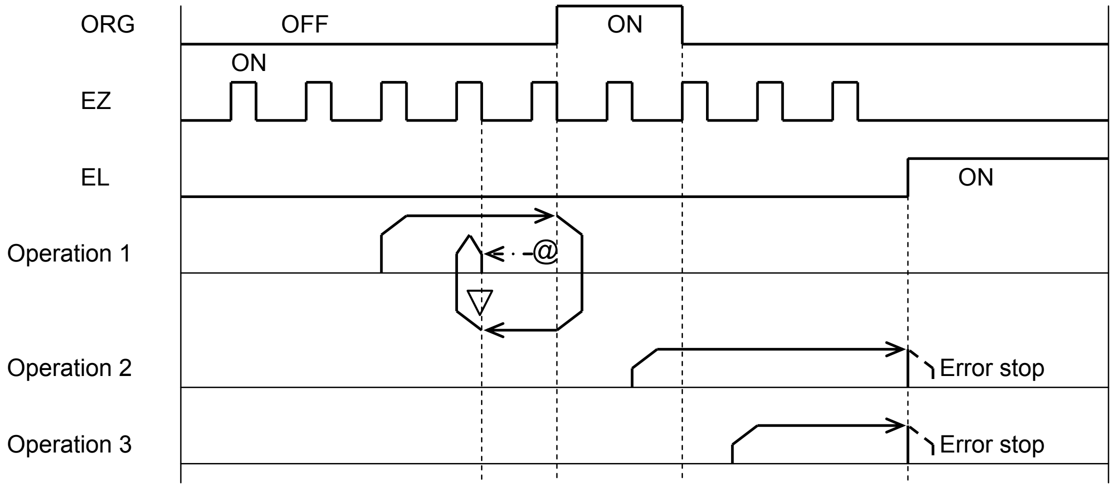
6-4-1-11. Origin return operation 10 (ORM = 1010)

■ High speed operation <Sensor: EL, ORG, EZ (EZD = 0001)>



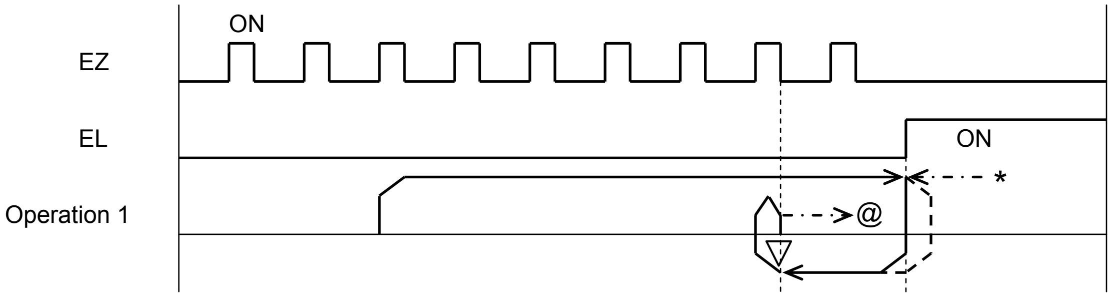
6-4-1-12. Origin return operation 11 (ORM = 1011)

■ High speed operation <Sensor: EL, ORG, EZ (EZD = 0001)>



6-4-1-13. Origin return operation 12 (ORM = 1100)

■ High speed operation <Sensor: EL, EZ (EZD = 0001)>

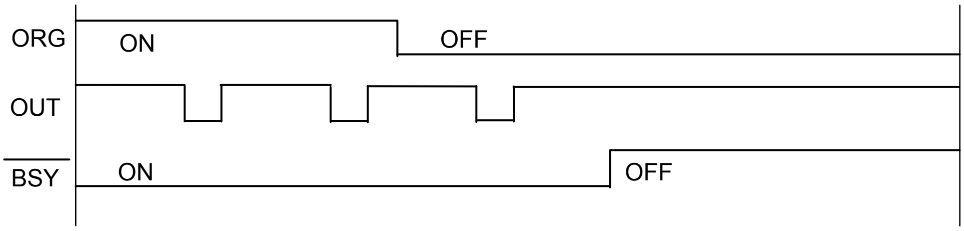


Note: Positions marked with "@" reflect the ERC signal output timing when "Automatically output an ERC signal" is selected for the zero stopping position.
Also, when EROE (bit 10) is 1 in the RENV1 register and ELM (bit 3) is 0, the LSI will output an ERC signal at positions marked with an asterisk (*).

6-4-2. Leaving the origin position operations

After writing a start command, the axis will leave the origin position (when the ORG input is ON).
Make sure to use the "Constant speed start command (0050(h), 0051(h))" when leaving the origin position.
When you write a start command while the ORG input is OFF, the LSI will stop the movement on the axis as a normal stop, without outputting any pulses.
Since the ORG input status is sampled when outputting pulses, if the G9103A starts at constant speed while the ORG signal is ON, it will stop operation after outputting one pulse after the ORG input is turned OFF. (Normal stop)

MOD: 12h Leave the origin position in the positive direction
1Ah Leave the origin position in the negative direction



6-4-3. Origin search operation

This mode is used to add functions to an origin return operation. It consists of the following possibilities.

1) An "Origin return operation" is performed in the opposite direction to the one specified.
2) A "Leaving the origin position using positioning operations" is executed in the opposite direction to the one specified.
3) An "Origin return operation" is executed in the specified direction.

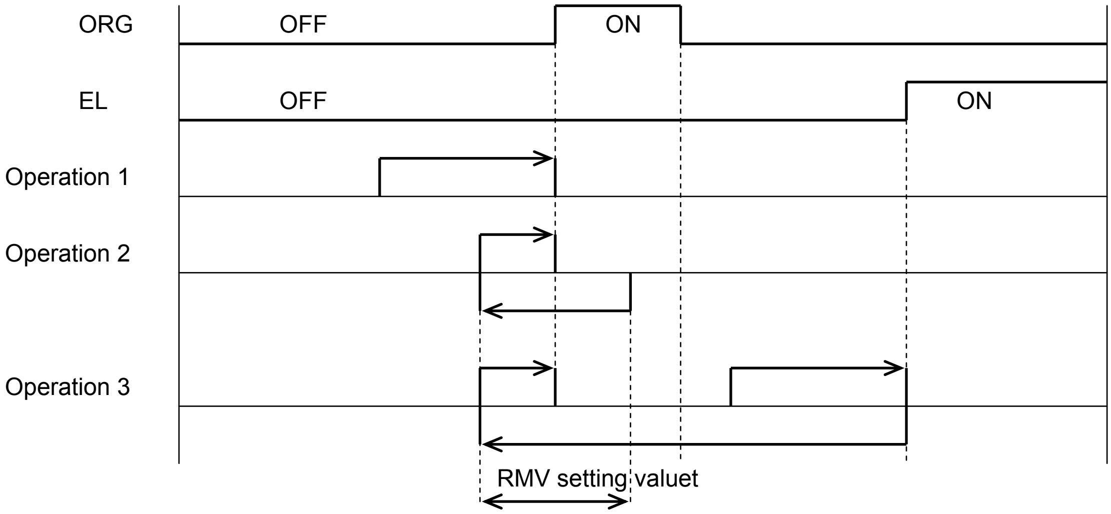
Operation 1: If the ORG input is turned ON after starting, the axis will stop normally.
Operation 2: If the ORG input is already turned ON when starting, the axis will leave the origin position using positioning operations, and then begin an "origin return operation."
Operation 3: If the axis is stopped by an EL signal while operating in the specified direction, the LSI will execute an "origin return operation (ORM = 0000)" and a "leaving the origin position by positioning" in the opposite direction. Then it will execute an "origin return operation" in the specified direction.

When "leaving the origin position by positioning," the LSI will repeat the positioning operation for the number of pulses specified in the RMV (target position) register, until the origin position has been left. Enter a positive number (1 to 134,217,727) in RMV register.

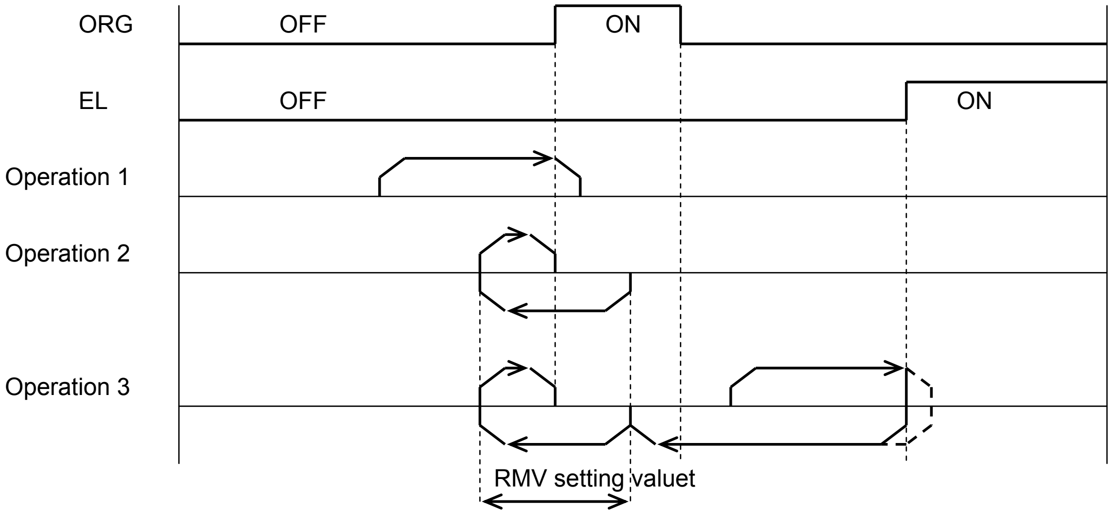
MOD: 15(h) Origin search operation in the positive direction
1D(h) Origin search operation in the negative direction

6-4-3-1. Origin return operation 0 (ORM=0000)

□ Constant speed operation <Sensor: EL, ORG>



■ High speed operation <Sensor: EL, ORG>
Even if the axis stops normally, it may not be at the origin position. However, COUNTER 2 (mechanical position) provides a reliable value.



6-5. EL or SL operation mode

The following four modes of EL or SL (soft limit) operation are available.

MOD	Operation mode	Direction of movement
20(h)	Operate until reaching the +EL or +SL position.	Positive direction
28(h)	Operate until reaching the -EL or -SL position.	Negative direction
22(h)	Leave the -EL or -SL positions.	Positive direction
2A(h)	Leave the +EL or +SL positions.	Negative direction

To specify the ±EL input signal, set the input logic using ELL input terminal. Select the operation type (immediate stop / deceleration stop) when the input from that terminal is ON in the RENV1 (Environment setting 1) register. The status of the terminal can be monitored using the RSTS (extension status) register. For details about setting the SL (software limit), see section 8-11-2, "Software limit function."

Select the ±EL signal input logic <ELL input terminal> L: Positive logic input H: Negative logic input	
Select the stop method to use when the ±EL signal is turned ON <RENV1.ELM (bit 3)> 0: Stop immediately when the ±EL signal turns ON. 1: Decelerates and stops when the ±EL signal turns ON.	[RENV1] (WRITE) 7 0 - - - - n - - -

Reading the ±EL signal <RSTS.SPEL (bit 6), SMEL (bit 7)>
SPEL=0: Turn OFF +EL signal SPEL=1: Turn ON +EL signal
SMEL=0: Turn OFF -EL signal SMEL=1: Turn ON -EL signal
	[RSTS] (WRITE)
	7 0
	n n - - - - - -

Setting the ±EL input filter <RENV1.FLTR (bit 25)> 0: Apply a filter to the ±EL, ORG input. After applying a filter, signals shorter than 4 µsec will be ignored.	[RENV1] (WRITE) 31 24 - - - - - - n -

6-5-1. Feed until reaching an EL or SL position

This mode is used to continue feeding until the EL or SL (soft limit) signal turns ON and then the operation stops normally.
When a start command is written on the position where the EL or SL signal turns ON, the LSI will not output pulses and it will stop the axis normally. When a start command is written while the EL and SL signals are OFF, the axis will stop when the EL or SL signal turn ON. (Normal stop)

MOD: 20(h) Feed until reaching the +EL or +SL position.
28(h) Feed until reaching the -EL or -SL position.

6-5-2. Leaving an EL or SL position

This mode is used to continue feeding until the EL or SL (software limit) signal turns OFF.
When a start command is written on the position where the EL and SL signals turn OFF, the LSI will not output pulses and it will stop the axis normally.
When starting an operation while the EL input or SL signal is ON, the G9103A will stop operation normally when both the EL input and SL signal are OFF.

MOD: 22(h) Leave a -EL or -SL position
2A(h) Leave a + EL or +SL position

6-6. EZ count operation mode

This mode is to operate until EZ signal counts reaches the number (EZD set value +1) written into RENV3 register.

MOD: 24(h) Feed in positive direction until the EZ count is completed.
2C(h) Feed in negative direction until the EZ count is completed.

After a start command is written, the axis stops immediately (or decelerates and stops when feeding at high speed) after the EZ count equals the number stored in the register.

The EZ count can be set from 1 to 16.

Use the constant speed start command (0050(h), 0051(h)) for this operation. When the high speed start command is used, the motor will start decelerating and stop when the EZ signal turns ON, Therefore, the machine position overruns the EZ position of specified number of times.

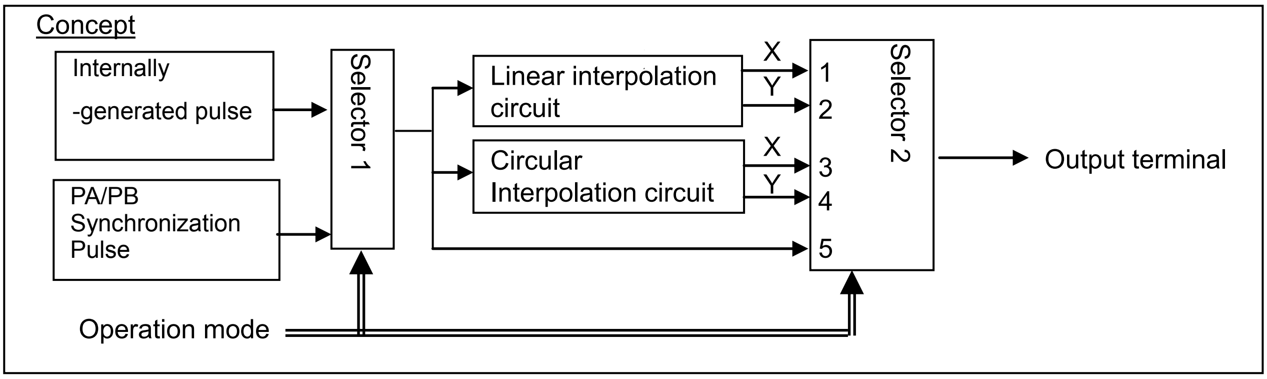
Specify input logic for the EZ signal in RENV2 (environment setting 2) register, and the EZ number to count in RENV3 (environment setting 3) register. The terminal status can be monitored by reading the RSTS (extension status) register.

Setting the input logic of the EZ signal <Set RENV2.EZL (bit 12)> 0: Falling edge 1: Rising edge	[RENV2] (WRITE) 15 8 - - - n - - - -
Setting the EZ count number <Set RENV3.EZD0 to 3 (bits 4 to 7)> Specify the EZ count number that is an origin return complete condition. Enter a value (the number to count to minus 1) in EZD 0 to 3. Setting range: 0 to 15.	[RENV3] (WRITE) 7 0 n n n n - - - -
Reading the EZ signal < RSTS.SEZ (bit 16)> 0: Turn OFF the EZ signal 1: Turn ON the EZ signal	[RSTS] (READ) 23 16 - - - - - - - n
Reading the residual number of the EZ signal <RSPD.ECZ (bit 20 to 23)> Note. RENV3.EZD value is set again after the operation stops.	[RSTS] (READ) 23 16 n n n n - - -

6-7. Interpolation operation mode

Using numeral G9103As allows executing linear interpolation among any more than two axes and circular interpolation between any two axes. G9103A has linear interpolation circuit for 2 axes and circular interpolation circular for 2 axes. In explanation, output axes of interpolation circuits are referred to as “Interpolated X axis” and “Interpolated Y axis”. G9103A is the LSI to control single axe. Therefore, in interpolation, it outputs pulse trains of either “Interpolated X axis” or “Interpolated Y axis” to a terminal. Select which axis signal is output by operation mode (PRMD.MOD.)

By setting the same to multiple G9103As, multiple motors can also control interpolated X axis like gantry structure.



Selector 1 : Select whether control by synchronization with PA/PB or control by internal pulse.
Selector 2 : Select the following five types of operations

1. X axis output of linear interpolation
2. Y axis output of linear interpolation
3. X axis output of circular interpolation
4. Y axis output of circular interpolation
5. Output of operation mode except interpolation

Note 1. When interpolation operation block is executed continuously, the start timing between each block on each G9103A is not synchronized. Each G9103A controls the start timing on the assumption that operation time of each G9103A the same. Therefore, interpolation operation cannot be used with the functions that operation time is different among G9103As such as backlash correction, vibration restriction function, direction change timer, and delay by INP input.

Note 2. In interpolation between 2 axes, PRMD.MIPF=1 allows to control synthesized speed so as to be constant. However, in interpolation among more than 3 axes, this control cannot be executed. Calculate the speed value so as to make synchronized speed constant using software.

Note 3. The interpolation operation is started by broadcast communication. However, a slight error of the start timing occurs by the time difference that the broadcast communication frame (electric signals) is transferred in a cable. (Approximately 50n seconds per 10 meters).
In the case of 100 meters, the time difference becomes approximately 500n seconds.
Approximately 1μ seconds occurs each time passing through the Motionnet HUB.

Note 4. Make sure to use the synchronization function of clock for motor control in the system for interpolation operation. G9001A has two terminals SIA and SIB as serial input terminal and the communication line can be separated into two. However, G9103A of the interpolated axes should not be separate into two lines.

Note 5 Each G9103A’s circuit to control speed does not synchronize between LSIs and but only execute interpolation by same operation at the same speed pattern. Therefore, note that interpolation trajectory may be distorted if one axis decelerates by SD input, etc.

There are 16 interpolation operation modes as follows.

PRMD.MOD	Description	Interpolated output axis
60(h)	Continuous linear interpolation	X axis
61(h)	Linear interpolation	X axis
64(h)	Circular interpolation in CW direction	X axis
65(h)	Circular interpolation in CCW direction	X axis
68(h)	Continuous linear interpolation synchronized with PA/PB	X axis
69(h)	Linear interpolation synchronized with PA/PB	X axis
6C(h)	Circular interpolation in CCW direction synchronized with PA/PB	X axis
6D(h)	Circular interpolation in CCW direction synchronized with PA/PB	X axis
70(h)	Continuous linear interpolation	Y axis
71(h)	Linear interpolation	Y axis
74(h)	Circular interpolation in CW direction	Y axis
75(h)	Circular interpolation in CCW direction	Y axis
78(h)	Continuous linear interpolation synchronized with PA/PB	Y axis
79(h)	Linear interpolation synchronized with PA/PB	Y axis
7C(h)	Circular interpolation in CCW direction synchronized with PA/PB	Y axis
7D(h)	Circular interpolation in CCW direction synchronized with PA/PB	Y axis

Set the same value for the registers (PRFL, PRFH, PRUR, PRDR, PRMG, PRDP, PRUS and PRDS) that are related to the speed of axes interpolated by G9103A.

Note. When interpolation operation synchronized with PA/PB is complete, stop interrupt (MSTS.SEND) is not output.
If interrupt is needed, please use normal stop of event interrupt (RIRQ.IREN).

6-7-1. Linear interpolation operation

6-7-1-1. Linear interpolation between 2 axes

Set the feed amount in PRMV and PRMVY.
Select linear interpolation (output x axis pulses) and linear interpolation (output Y axis pulses) with holding start. (RMD.MSY=1)
Set the same value for the same speed pattern registers (PRFL, PRFH, PRUR, PRDR, PRMG, PRDP, PRUS, PRDS) of the interpolated X axis and Y axis.
Write the same start command (0050(h) to 0053(h)) on each axis and start with the start command (2001(h)) of the broadcast communication.

Here is the example of the linear interpolation between 2 axes feeding amount 1000 and 2000.
In addition to this method that PRMD is changed from one axis to another, linear interpolation among numeral axes is available (See 6-7-2-1).

Register	G9103A-X	G9103A-Y
PRMD	00004461(h)	00004471(h)
PRMV	1000	1000
PRMVY	2000	2000
PRFL	1	1
PRFH	100000	10000
PRUR	20	20
PRDR	0	0
PRMG	1 99	199
PRDP	0	0
PRUS	0	0
PRDS	0	0

6-7-1-2. Linear interpolation among numeral axes (2 to 64 axes).

Set the feed amount of the longest axis in PRMVY and the feed amount of own axis in PRMV. Select linear interpolation (output X axis pulses) for operation mode of all axes with holding start.
Set the same value for the same speed pattern registers (PRFL, PRFH, PRUR, PRDR, PRMG, PRDP, PRUS and PRDS) of all axes.
Write the same start command (0050(h) to 0053(h)) on each axis and start with the start command (2001(h)) of the broadcast communication.

Here is the example of the linear interpolation among 4 axes feeding amount 1000, 2000, 3000 and 4000.

Register	G9103A-1	G9103A-2	G9103A-3	G9103A-4
PRMD	00004461(h)	00004461(h)	00004461(h)	00004461(h)
PRMV	1000	2000	3000	4000
PRMVY	4000	4000	4000	4000
PRFL	1	1	1	1
PRFH	10000	10000	10000	10000
PRUR	20	20	20	20
PRDR	0	0	0	0
PRMG	199	199	199	199
PRDP	0	0	0	0
PRUS	0	0	0	0
PRDS	0	0	0	0

6-7-2. Circular interpolation operation

Set the incremental position of the ending point in PRMV and PRMVY based on start position.
The ending point of a perfect circle is (0, 0). In the case that an ending point is set at the point except the circumference of the circle, linear movement to the ending point is operated after circular interpolation. This operation is referred to as ending point lead-in operation. The ending point lead-in operation is described below.
Set the incremental position of the center for circular in PRMV and PRMV.
Select linear interpolation (output X axis pulses) and linear interpolation (output Y axis pulses) for operation mode with holding start.
Set the same value for the same speed pattern register (PRFL, PRFH, PRUR, PRDR, PRMG, PRDP, PRUS and PRDS).
Write the high-speed start command (0053(h)) on each axis and start with the start command (2001(h)) of the broadcast communication.

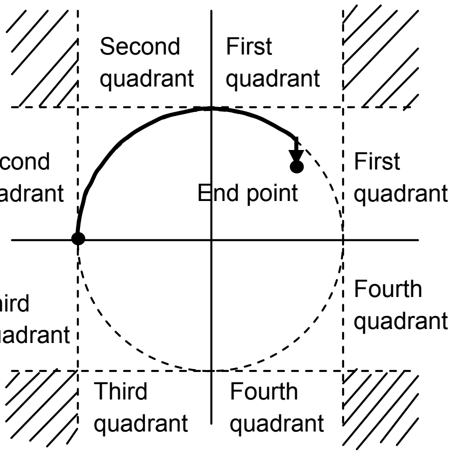
Here is the example of the circular interpolation of 180 degrees in CW direction with the center (100,0) and ending point (200,0). The method for calculation of PRCI setting value is described later.

Register	G9103A-X	G9103A-Y
PRMD	00004464(h)	00004474(h)
PRMV	200	200
PRMVY	0	0
PRIP	100	100
PRIPY	0	0
PRCI		
PRFL	1	1
PRFH	10000	10000
PRUR	20	20
PRDR	0	0
PRMG	199	199
PRDP	0	0
PRUS	0	0
PRDS	0	0

6-7-2-1. Ending point lead-in operation

In the case that an ending point is set at the point except the circumference of the circle, linear movement to the ending point is operated after circular interpolation. This operation is referred to as ending point lead-in operation.
In circular interpolation, when one axis reaches the ending point in the quadrant of end point, the interpolation operation completes and the axis move to the end point coordinate. The speed of ending point lead-in operation is the same as the speed of circular interpolation.

Avoid that an end point coordinate of circular interpolation is set in the shaded areas in the right figure because the axis does not stop and continue circular interpolation eternally.

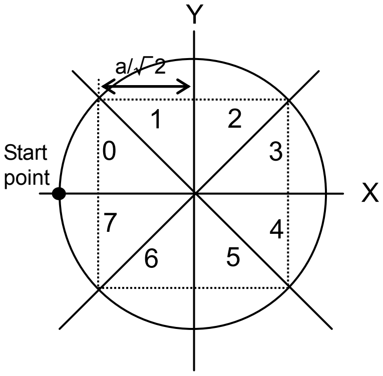


6-7-2-2. Number of stepping in circular interpolation

When circular interpolation is executed with acceleration / deceleration, it is needed to set the number of pulses (the number of stepping for circular interpolation) necessary for circular interpolation in PRCI register of axis controlled.
In calculating the number of pulses necessary for circular interpolation, assume that the plane containing X-axis and Y-axis is divided into 8 areas (0 to 7) regarding the circular coordinate of the circular as the center.

The status of pulses output by each axis in each area is as follows.

Area	X axis output pulse	Y axis output pulse
0	Output based on a result of calculation for interpolation	Always output
1	Always output	Output based on a result of calculation for interpolation
2	Always output	Output based on a result of calculation for interpolation
3	Output based on a result of calculation for interpolation	Always output
4	Output based on a result of calculation for interpolation	Always output
5	Always output	Output based on a result of calculation for interpolation
6	Always output	Output based on a result of calculation for interpolation
7	Output based on a result of calculation for interpolation	Always output

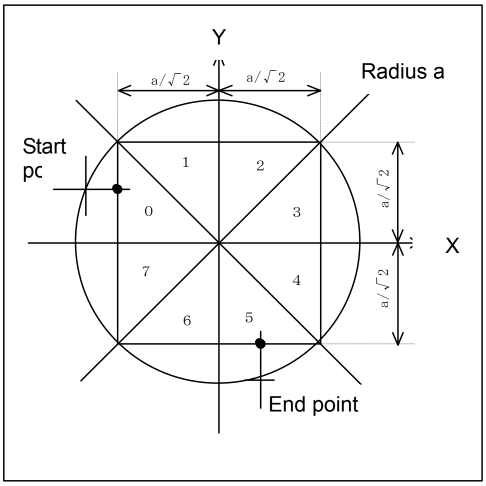


According to the above table, you can see that either axis output pulses in any areas.
Therefore, The number of pulses (number of stepping in circular interpolation) necessary for the distribution of circular interpolation is equal to the number of pulses that moves on the trajectory of square inscribed in the circle.
For example, in the case that 90 degrees of circular with radius "a" is depicted, the number of pulses necessary for circular interpolation is $(a/\sqrt{2})x2$. Set this value in PRCI register.

In order to calculate the number of stepping on arbitrary start point and end point, the procedure is as follows.

1. Discriminate that the start point is in which area (area 0 to 7) based on the center coordinate and obtain the intersection of the perpendicular from the start point to the inscribed square.
2. Discriminate that the end point is in which are (area 0 to 7) based on the end and center coordinate and obtain the intersection of the perpendicular from the end point to the inscribed square.
3. On the inscribed square, obtain the length from the intersection with the start point perpendicular to the intersection with the end point and set this value in PRCI register.

   In the case of that the end point is not on the circumference of the circle, add the number of pulses necessary for the end point lead-in to the above value and set the value in PRCI register.



Note 1. The PRCI register value is used for the occurrence of timing to start deceleration. If a small value is entered according to the error of calculation, the deceleration starts early and the time to feed at FL constant speed appear. If a bigger value is entered, the deceleration starts late and the axis stops at more than FL speed. However, in any cases, the interpolation trajectory is the same as constant circular interpolation.

Note 2. To specify the ramp-down point manually, you can use the PRDP formula in positioning operation if the PRCI setting value is regarded as the number of output pulses.
However, when the function that makes synthesized speed constant is ON, the above formula cannot be used. The value has to be obtained by the change of RCIC in experiment.

6-7-3. Continuous operation of interpolation operation

While interpolation operation, the data for the next data can be set. Therefore, interpolation operation can be executed continuously.
However, note that it is necessary to set new data for operation one by one and operation cannot be continued if operation time for interpolation is shorter than communication process time.
If setting that RIRQ.IRBE is 1, G9001A can outputs an interrupt request signal ($\overline{\text{INT}}$) when operation is stopping because sending the data for next operation is not in time.
In the case with using circular interpolation in addition to linear interpolation in continuous interpolation among more than 3 axes, make axes that do not operate in circular interpolation operate the dummy circular interpolation with setting PRMD.MDMY=1.

Example of continuous operation of interpolation

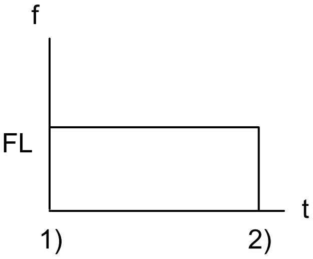
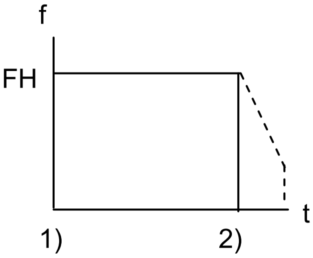
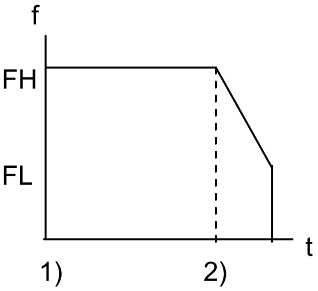
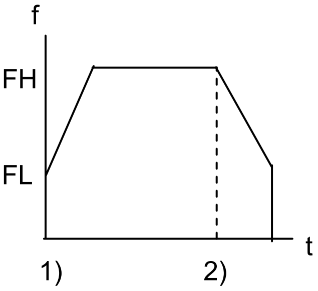
Operation	Register	G9103A_X	G9103A_Y	G9103A_Z	Notes
1	PRMV	1000	0	1500	Linear interpolation between 2 axes with X-axis and Y-axis. The end point is X=1000, Z=1500. Set PRMV=0 because Y-axis does not operate. Initially make the operation hold start and start by the broadcast command.
	PRMVY	1500	1500	1500	
	PRIP	Invalid	Invalid	Invalid	
	PRIPY	Invalid	Invalid	Invalid	
	PRMD	00004061	00004061	00004061	
2	PRMV	3000	100	2000	Linear interpolation among 3 axes (X, Y and Z). The end point: X=3000, Y=100, Z=2000.
	PRMVY	3000	3000	3000	
	PRIP	Invalid	Invalid	Invalid	
	PRIPY	Invalid	Invalid	Invalid	
	PRMD	00000061	00000061	00000061	
3	PRMV	-2000	-2000	-2000	Circular interpolation in CW direction with X-axis and Z-axis. Center:X=0, Z=2000. End point:X=-2000, Z=2000, Y-axis: dummy operation
	PRMVY	2000	2000	2000	
	PRIP	0	0	0	
	PRIPY	2000	2000	2000	
	PRMD	00000064	40000064	00000074	
4	PRMV	0	0	0	Circular interpolation in CCW direction with X-axis and Y-axis. Center: X=100, Y=2000 End point: X=0, Y=0 (perfect circle) Z-axis_3: dummy operation。
	PRMVY	0	0	0	
	PRIP	100	100	100	
	PRIPY	2000	2000	2000	
	PRMD	00000065	00000075	40000065	
5	PRMV	0	0	1000	Only Z-axis moves +1000. Make the feed amount 0 in linear interpolation operation so as to make the operation time of other axes same.
	PRMVY	1000	1000	1000	
	PRIP	Invalid	Invalid	Invalid	
	PRIPY	Invalid	Invalid	Invalid	
	PRMD	00000061	00000061	00000061	

Procedure

1. Set the data for operation 1 data and the start command to each interpolated axis. Pulses are not output because holding start is set in PRMD register.
2. Set the data for operation 2 and the start command to each interpolated axis.
3. Send the broadcast communication command (0010 0ggg 0000 0001). The operation 1 starts.
4. After the operation 1 completed, write the data for the operation 3 and the start command into each interpolated axis.
5. After the operation 2 completed, write the data for the operation 4 and the start command into each interpolated axis.
6. After the operation 3 completed, write the data for the operation 5 and the start command into each interpolated axis.

7. Speed patterns

7-1. Speed patterns

Speed pattern	Continuous mode	Positioning operation mode
FL constant speed operation  	1) Write an FL constant speed start command (0050(h)). 2) Stop feeding by writing an immediate stop (0049(h)) or deceleration stop (004A(h)) command.	1) Write an FL constant speed start command (0050(h)). 2) Stop feeding when the positioning counter reaches zero, or by writing an immediate stop (0049(h)) or deceleration stop (004A(h)) command.
FH constant speed operation  	1) Write an FH constant speed start command (0051(h)). 2) Stop feeding by writing an immediate stop command (0049(h)).	1) Write an FH constant speed start command (0051(h)). 2) Stop feeding when the positioning counter reaches zero, or by writing an immediate stop (0049(h)) command.
	* When the deceleration stop command (004Ah) is written to the register, the G9103A starts deceleration.	
High speed operation 1  	1) Write high speed start command 1 (0052(h)). 2) Start deceleration by writing a deceleration stop command (004A(h))	1) Write high speed start 1 command (0052(h)). 2) Start deceleration when a ramping-down point is reached or by writing a deceleration stop command (004A(h)). * The automatic ramp down point (PRDP) setting is not possible at High speed operation 1. Please select manual setting (PRMD.MSDP=1), and set the value to PRDP.
	* When the immediate stop command (0049(h)) is written to the register, stops immediately.	
High speed operation 2  	1) Write high speed start command 2 (0053(h)). 2) Start deceleration by writing a deceleration stop command (004A(h)).	1) Write high speed start command 2 (0053(h)). 2) Start deceleration when a ramping-down point is reached or by writing a deceleration stop command (004A(h)). * When the automatic ramp down point setting is set to manual (RMD.MSDP = 1), and the ramp down point value (RDP) is set to "0," the G9103A immediately stops the motor.
	* When the immediate stop command (0049(h)) is written to the register, stops immediately.	

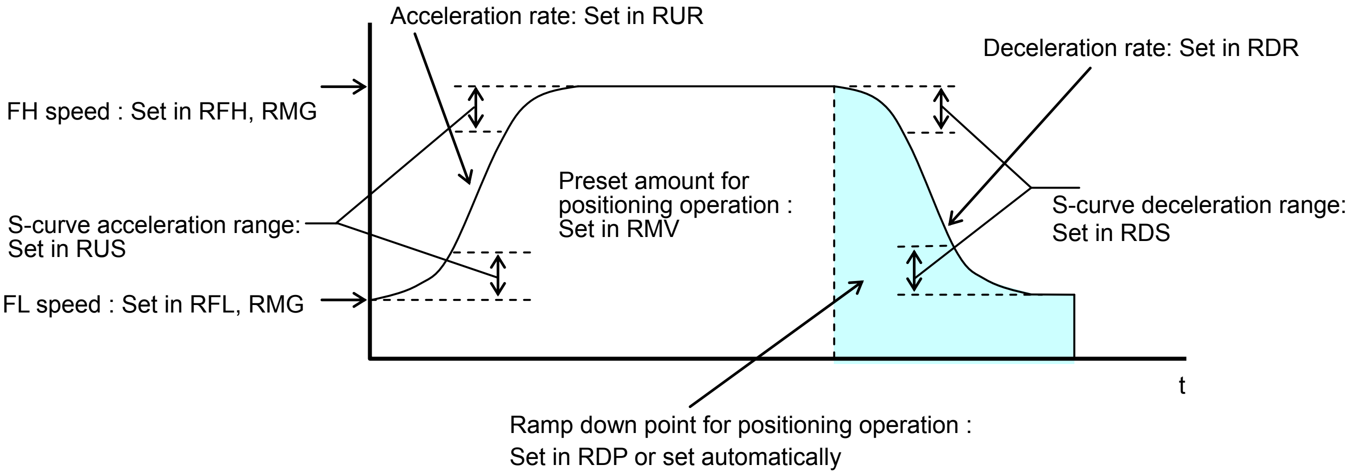
7-2. Speed pattern settings

Specify the speed pattern using the registers shown in the table below.
If the next register setting is the same as the current value, there is no need to write to the register again.
Please note that with some registers, a setting of "0" may be outside the allowable range.

Register	Description	Bit length setting range	Setting range	R/W
RMV	Positioning amount	28	-134,217,728 to 134,217,727 (8000000h) (7FFFFFFh)	R/W
RFL	Initial speed	17	1 to 100,000 (186A0(h)) Note2	R/W
RFH	Operation speed	17	1 to 100,000 (186A0(h)) Note2	R/W
RUR	Acceleration rate	16	1 to 65,535 (0FFFF(h))	R/W
RDR	Deceleration rate Note 1	16	0 to 65,535 (0FFFF(h))	R/W
RMG	Speed magnification rate	11	2 to 2,047 (7FF(h))	R/W
RDP	Ramping-down point	24	0 to 16,777,215 (0FFFFFF(h))	R/W
RUS	S-curve acceleration range	16	0 to 50,000 (0C350(h)) Note 3	R/W
RDS	S-curve deceleration range	16	0 to 50,000 (0C350(h)) Note 3	R/W
RFA	Feed amount correction speed	17	0 to 100,000 (186A0(h)) Note 2	R/W

Note 1: If RDR is set to zero, the value set in RUR is used as the deceleration rate.
Note 2: All values from 186A0(h) to 1FFFF(h) will be treated as 186A0(h).
Note 3: All values from 0C350(h) to 0FFFF(h) will be treated as 0C350(h).

[Relative positions of each register setting for acceleration and deceleration factors]


Acceleration rate: Set in RUR
Deceleration rate: Set in RDR
FH speed : Set in RFH, RMG
Preset amount for positioning operation : Set in RMV
S-curve acceleration range: Set in RUS
S-curve deceleration range: Set in RDS
FL speed : Set in RFL, RMG
t
Ramp down point for positioning operation : Set in RDP or set automatically

♦ PRMV (RMV): Positioning feed amount setting register (28 bits)
This register is used to set the target position for positioning operations.
The details for setting may vary according to the operation mode selected.
Setting rage: -134,217,728 to +134,217,727
By changing RMV register during operation, you can override the feed amount.

♦ PRFL (RFL): FL speed setting register (17 bits)
This register is used to set the initial speed (and stopping speed) in a high-speed operation (with acceleration/deceleration).
Specify the speed for FL constant speed operations and the start speed for high speed operations (acceleration/deceleration operations) in the range of 1 to 100,000 (186A0(h)). All values from 100,000 to 131,071 (186A0(h) to 1FFFF(h)) will be treated as 100,000.
The actual operation speed will be obtained from the following calculating formula with RMG value.

$$\text{FL speed [pps]} = \text{RFL} \times \frac{40{,}000{,}000}{(\text{RMG} + 1) \times 200{,}000}$$

- PRFH (RFH): FH speed setting register (17 bits)

  This register is used to set the operation speed.

  The speed can be changed in the middle of an operation by changing RFH register setting.

  Specify the speed for FH constant speed operations and the start speed for high speed operations (acceleration/deceleration operations) in the range of 1 to 100,000 (186A0(h)). All values from 100,000 to 131,071 (186A0(h) to 1FFFF(h)) will be treated as 100,000.

  In high speed operations (acceleration/deceleration operations), specify a value larger than RFL.

  The actual operation speed will be obtained from the following calculating formula with RMG value.

$$\text{FH speed [pps]} = \text{RFH} \times \frac{40{,}000{,}000}{(\text{RMG} + 1) \times 200{,}000}$$

- PRUR (RUR): Acceleration rate setting register (16 bits)

  This register is used to set the acceleration rate.

  Specify the acceleration characteristic for high speed operations (acceleration/deceleration operations), in the range of 1 to 65,535 (0FFFF(h))

  Relationship between the value entered and the acceleration time will be as follows:

1) Linear acceleration (MSMD = 0 in the RMD register)

$$\text{Acceleration time [s]} = \frac{(\text{RFH - RFL}) \times (\text{RUR} + 1) \times 8}{40{,}000{,}000}$$

2) S-curve without a linear range (RMD.MSMD=1 and RUS register = 0)

$$\text{Acceleration time [s]} = \frac{(\text{RFH - RFL}) \times (\text{RUR} + 1) \times 16}{40{,}000{,}000}$$

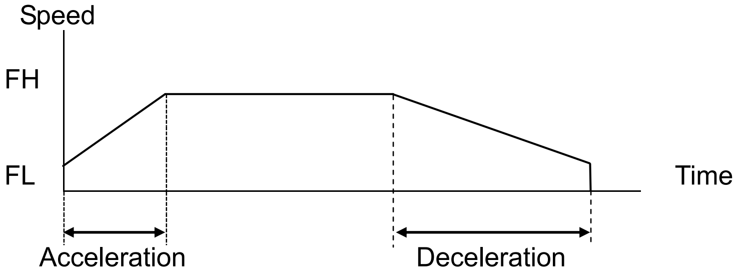
3) S-curve with a linear range (RMD.MSMD=1 and RUS register > 0)

$$\text{Acceleration time [s]} = \frac{(\text{RFH - RFL} + 2 \times \text{RUS}) \times (\text{RUR} + 1) \times 8}{40{,}000{,}000}$$

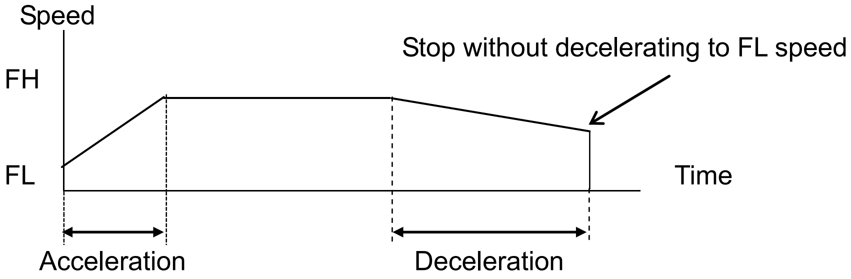
- PRDR (RDR): Deceleration rate setting register (16 bits)
  This register is used to set the deceleration rate.
  Normally, specify the deceleration characteristics for high speed operations (acceleration/deceleration operations) in the range of 1 to 65,535 (0FFFF(h)).
  Even if the ramping-down point is set to automatic (RND.MSDP = 0), the value placed in RDR register will be used as the deceleration rate.
  However, when RDR = 0 , the value placed in RUR is used as the deceleration rate

  When you want to set the auto ramp-down point, adjust it so that
  (deceleration time) ≦ (acceleration time x 2).
  If the (deceleration time) > (acceleration time x 2), the motor may not be able to decelerate to the FL speed when stopping. In this case, select the manual ramp-down point setting method (MSDP = 1 in the RMD register).

  < When (deceleration time) ≦ (acceleration time x 2) using an automatic ramping-down point >



  <When (deceleration time) > (acceleration time x 2) using an automatic ramping-down point>



The relationship between the value entered and the deceleration time is as follows.

1) Linear deceleration (RMD.MSMD)

$$\text{Deceleration time [s]} = \frac{(RFH - RFL) \times (RDR + 1) \times 8}{40{,}000{,}000}$$

2) S-curve deceleration without a linear range (RMD.MSMD=1 and RDS register = 0)

$$\text{Deceleration time [s]} = \frac{(RFH - RFL) \times (RDR + 1) \times 16}{40{,}000{,}000}$$

3) S-curve deceleration with a linear range (RMD.MSMD=1 and RDS register >0)

$$\text{Deceleration time [s]} = \frac{(RFH - RFL + 2 \times RDS) \times (RDR + 1) \times 8}{40{,}000{,}000}$$

♦ PRMG (RMG): Magnification rate register (11 bits)
This register is used to set the speed magnification rate.
Specify the relationship between the RFL, RFH and RFA settings and the speed, in the range of 2 to 2,047 (07FF(h)). As the magnification rate is increased, the speed setting units tend to be coarse. Normally set the magnification rate as low as possible.
The relationship between the value entered and the magnification rate is as follows.

$$\text{Magnification rate} = \frac{40{,}000{,}000}{(\text{RMG} + 1) \times 200{,}000}$$

[Magnification rate setting example] (Output speed unit: pps)

Setting	Magnification rate	Output speed range	Setting	Magnification rate	Output speed range
1999 (7CF(h))	0.1	0.1 to 10,000.0	39 (27(h))	5	5 to 500,000
999 (3E7(h))	0.2	0.2 to 20,000.0	19 (13(h))	10	10 to 1,000,000
399 (18F(h))	0.5	0.5 to 50,000.0	9 (09(h))	20	20 to 2,000,000
199 (0C7(h))	1	1 to 100,000	3 (3(h))	50	50 to 5,000,000
99 (63(h))	2	2 to 200,000	2 (2(h))	66.6	66.6…to 6,666,666.6…

♦ PRDP(RDP): Ramp down point setting register (24 bits)
This register is used to set the ramp-down point (deceleration starting point).
Specify the value used to determine the deceleration starting point for positioning operations that include acceleration and deceleration.
The meaning of the value specified in RDP varies according to the "ramping-down point setting method," (RMD.MSDP).

<When set to manual setting (RMD.MSDP=1)>
Set the number of pulses at which to start deceleration in the range of 0 to16,777,215 (0FFFFFF(h)).
The optimum value for the ramp-down point can be calculated as shown in the equation below.

1) Linear deceleration (RMD.MSMD=0)

$$\text{Optimum value [Number of pulses]} = \frac{(\text{RFH}^2 - \text{RFL}^2) \times (\text{RDR} + 1)}{(\text{RMG} + 1) \times 50{,}000}$$

However, the optimum value for a triangle start, without changing the value in RFH register while turning OFF the FH correction function (RMD.MADJ=1) is calculated as shown in the equation below.
(When using idling control, modify the value for RMV in the equation below by deducting the number of idling pulses from the value placed in RMV register. The number of idling pulses will be "1 to 6" when RENV2.IDL = 0 to 7).

$$\text{Optimum value [Number of pulses]} = \frac{\text{RMV} \times (\text{RDR} + 1)}{\text{RUR} + \text{RDR} + 2}$$

2) S-curve deceleration without a linear range (RMD.MSMD=1 and the RDS register=0)

$$\text{Optimum value [Number of pulses]} = \frac{(\text{RFH}^2 - \text{RFL}^2) \times (\text{RDR} + 1) \times 2}{(\text{RMG} + 1) \times 50{,}000}$$

3) S-curve deceleration with a linear range (RMD.MSMD=1 and the RDS register>0)

$$\text{Optimum value [Number of pulses]} = \frac{(\text{RFH} + \text{RFL}) \times (\text{RFH} - \text{RFL} + 2 \times \text{RDS}) \times (\text{RDR} + 1)}{(\text{RMG} + 1) \times 50{,}000}$$

Start deceleration at the point when the (positioning counter value) ≦ (RDP set value).

<When set to automatic setting (RMD.MSDP = 0)>
This is an offset value for the automatically set ramping-down point. Set in the range of -8,388,608 (800000(h)) to 8,388,607 (7FFFFF(h)).
The axis will start deceleration at an earlier stage when the offset value is a positive number than when the offset value is set to and will feed at the FL speed after decelerating. When a negative number is entered, the deceleration start timing will be delayed. If the offset is not required, set to zero.
When the value for the ramp-down point is smaller than the optimum value, the speed when stopping will be faster than the FL speed. On the other hand, if it is larger than the optimum value, the axis will feed at FL constant speed after decelerating completes.

♦ PRUS (RUS): S-curve acceleration range register (16 bits)
This register is used to specify the S-curve range in S-curve acceleration.
Specify the S-curve acceleration range for S-curve acceleration/deceleration operations in the range of 1 to 50,000 (0C350(h)).
All values from 50,000 to 65,535 (0C350(h) to 0FFFF(h)) will all be treated as 50,000.
The S-curve acceleration range S_{SU} will be calculated by the following formula with the value placed in RMG.

$$S_{SU}\,[\text{pps}] = \text{RUS} \times \frac{40{,}000{,}000}{(\text{RMG} + 1) \times 200{,}000}$$

In other words, speeds between the FL speed and (FL speed + S_{SU}), and between (FH speed - S_{SU}) and the FH speed, will be S-curve acceleration operations. Intermediate speeds will use linear acceleration.
However, if zero is specified, "(RFH - RFL)/2" will be used for internal calculations, and the operation will be an S-curve acceleration without a linear component.
If the minimum value "1" is specified, the G9103A will operate with nearly linear acceleration.
If a larger value than "(RFH - RFL) / 2" is specified, the motor will not reach the maximum acceleration speed and the acceleration time will be different from the calculated value. Therefore, enter a value smaller than "(RFH - RFL) / 2."

♦ PRDS (RDS): S-curve deceleration range setting register (16 bits)
This register is used to specify the S-curve range in an S-curve deceleration
Specify the S-curve deceleration range for S-curve acceleration/deceleration operations in the range of 1 to 50,000 (0C350(h)).
All values from 50,000 to 65,535 (0C350(h) to 0FFFF(h)) will all be treated as 50,000.
The S-curve acceleration range S_{SD} will be calculated by the following formula with the value placed in RMG.

$$S_{SD}\,[\text{pps}] = \text{RDS} \times \frac{40{,}000{,}000}{(\text{RMG} + 1) \times 200{,}000}$$

In other words, speeds between the FL speed and (FL speed + S_{SD}), and between (FH speed - S_{SD}) and the FH speed, will be S-curve deceleration operations. Intermediate speeds will use linear deceleration.
However, if zero is specified, "(RFH - RFL)/2" will be used for internal calculations, and the operation will be an S-curve deceleration without a linear component.
If the minimum value "1" is specified, the G9103A operates with nearly linear acceleration.
If a larger value than "(RFH - RFL) / 2" is specified, the motor will not reach the maximum acceleration speed and the deceleration time will be different from the calculated value. Therefore, enter a value smaller than "(RFH - RFL) / 2."

- ♦ RFA: FA speed setting register (17bits)

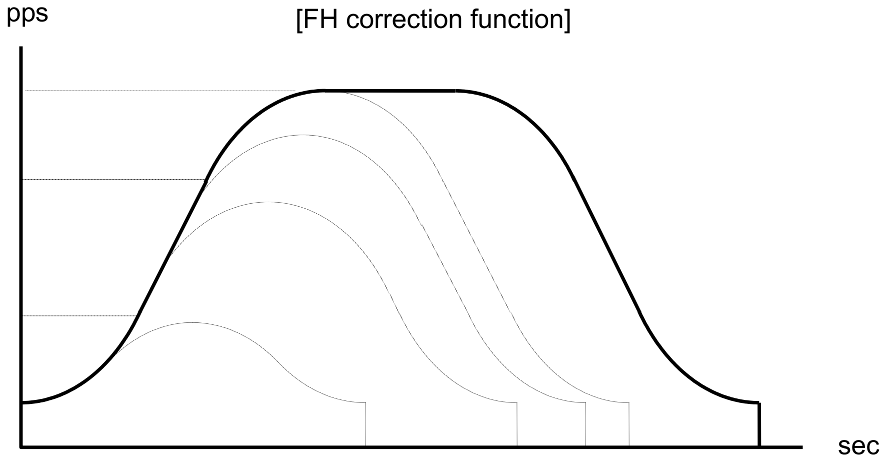
This register is used to set the constant speed during backlash correction.
Set the correction speed feed amount for use during backlash within the range of 1 to 100,000 (186A0(h)).
All values from 100,000 to 131,071 (186A0(h) to 1FFFF(h)) will all be treated as 100,000.
The actual operating speed will be the value calculated using RMG setting.
This register value is also used for the reverse constant speed during origin return operation

$$\text{FA speed [pps]} = \text{RFA} \times \frac{40{,}000{,}000}{(\text{RMG} + 1) \times 200{,}000}$$

7-3. Manual FH correction

When the FH correction function is turned ON (RMD.MADJ = 0), and when the feed amount is too small for a normal acceleration and deceleration operation, the LSI will automatically lower the FH speed to eliminate triangle driving.
However, if values in the RUR and RDR registers are set so that the (deceleration time) > (acceleration time x 2), do not use the FH correction function.

In order to eliminate triangle driving without using the FH correction function (RMD.MADJ = 1), lower the FH speed before starting the acceleration/deceleration operation.
When using idling control, enter a value for RMV in the equation below after deducting the number of idling pulses. The number of idling pulses will be 1 to 6 when RENV2.IDL = 2 to 7.



Automatic correction of the maximum speed for changing the feed amount

< To execute FH correction manually>

1) Linear acceleration/deceleration speed (RMD.MSMD=0)
When

$$RMV \leqq \frac{(RFH^2 - RFL^2) \times (RUR + RDR + 2)}{(RMG + 1) \times 50000}$$

$$RFH \leqq \sqrt{\frac{(RMG + 1) \times 50000 \times RMV}{RUR + RDR + 2} + RFL^2}$$

2) S-curve acceleration without linear acceleration (RMD.MSMD=1 and RUS register=0, RDS registers = 0)
When

$$RMV \leqq \frac{(RFH^2 - RFL^2) \times (RUR + RDR + 2)}{(RMG + 1) \times 50000}$$

$$RFH \leqq \sqrt{\frac{(RMG + 1) \times 50000 \times RMV}{(RUR + RDR + 2) \times 2} + RFL^2}$$

3) S-curve acceleration/deceleration with linear acceleration/deceleration (RMD.MSMD=1 and RUS register > 0, RDS register > 0)
(3)-1. When RUS = RDS
(i) Set up a small linear acceleration range
When

$$RMV \leqq \frac{(RFH + RFL) \times (RFH - RFL + 2 \times RUS) \times (RUR + RDR + 2)}{(RMG + 1) \times 50000} \quad \text{and}$$

$$RMV > \frac{(RUS + RFL) \times RUS \times (RUR + RDR + 2) \times 8}{(RMG + 1) \times 50000}$$

$$RFH \leqq - RSU + \sqrt{(RUS - RFL)^2 + \frac{(RMG + 1) \times 50000 \times RMV}{(RUR + RDR + 2)}}$$

(ii) Eliminate the linear acceleration/deceleration range
When

$$RMV \leqq \frac{(RUS + RFL) \times RUS \times (RUR + RDR + 2) \times 8}{(RMG + 1) \times 50000}$$

Change to S-curve acceleration/deceleration without a linear acceleration/deceleration range (RUS = 0, RDS = 0),

$$RFH \leqq \sqrt{\frac{(RMG + 1) \times 50000 \times RMV}{(RUR + RDR + 2) \times 2} + RFL^2}$$

RMV: Positioning amount	RFL: Initial speed	RFH: Operation speed
RUR: Acceleration rate	RDR: Deceleration rate	RMG: Speed magnification rate
RUS: S-curve acceleration range	RDS: S-curve deceleration range	

(3)-2. When RUS < RDS

(i) Set up a small linear acceleration/deceleration range
When

$$RMV \leqq \frac{(RFH+RFL) \times \{(RFH-RFL) \times (RUR + RDR + 2) + 2 \times RUS \times (RUR+1) + 2 \times RDS \times (RDR + 1)\}}{(RMG + 1) \times 50000}$$

and

$$RMV > \frac{(RDS+RFL) \times \{RDS \times (RUR + 2 \times RDR + 3) + RUS \times (RUR + 1)\} \times 4}{(RMG + 1) \times 50000} \quad ,$$

$$RFH \leqq \frac{-A + \sqrt{A^2 + B}}{RUR + RDR + 2}$$

However, $A = RUS \times (RUR + 1) + RDS \times (RDR + 1)$
$B= \{(RMG + 1) \times 50000 \times RMV - 2 \times A \times RFL + (RUR + RDR + 2) \times RFL^2\} \times (RUR + RDR + 2)$

(ii) Eliminate the linear acceleration/deceleration range and set up a small linear acceleration section.
When

$$RMV \leqq \frac{(RDS + RFL) \times \{RDS \times (RUR + 2 \times RDR + 3)\} + RUS \times (RUR +1)\} \times 4}{(RMG + 1) \times 50000} \quad \text{and}$$

$$RMV > \frac{(RUS + RFL) \times RUS \times (RUR + RDR + 2) \times 8}{(RMG + 1) \times 50000} \quad ,$$

Change to S-curve acceleration/deceleration without any linear acceleration/deceleration (RUS>0, RDS=0)

$$RFH \leqq \frac{-A + \sqrt{A^2 + B}}{RUR + 2 \times RDR + 3}$$

However, $A = RUS \times (RUR + 1)$,
$B= \{(RMG + 1) \times 50000 \times RMV - 2 \times A \times RFL + (RUR + 2 \times RDR + 3) \times RFL^2\} \times (RUR + 2 \times RDR + 3)$

(iii) Eliminate the linear acceleration/deceleration range
When

$$RMV \leqq \frac{(RUS + RFL) \times RUS \times (RUR + RDR + 2) \times 8}{(RMG + 1) \times 50000}$$

Change to S-curve acceleration/deceleration without any linear acceleration/deceleration (RUS=0, RDS=0),

$$RFH \leqq \sqrt{\frac{(RMG + 1) \times 50000 \times RMV}{(RUR + RDR + 2) \times 2} + RFL^2}$$

RMV: Positioning amount	RFL: Initial speed	RFH: Operation speed
RUR: Acceleration rate	RDR: Deceleration rate	RMG: Speed magnification rate
RUS: S-curve acceleration range	RDS: S-curve deceleration range	

(3)-3. When RUS>RDS

(i) Set up a small linear acceleration/deceleration range

When

$$\text{RMV} \leqq \frac{(\text{RFH} + \text{RFL}) \times \{(\text{RFH} - \text{RFL}) \times (\text{RUR} + \text{RDR} + 2) + 2 \times \text{RUS} \times (\text{RUR} + 1) + 2 \times \text{RDS} \times (\text{RDR} + 1)\}}{(\text{RMG} + 1) \times 50000}$$

and

$$\text{RMV} > \frac{(\text{RUS} + \text{RFL}) \times \{\text{RUS} \times (2 \times \text{RUR} + \text{RDR} + 3) + \text{RDS} \times (\text{RDR} + 1) \times 4}{(\text{RMG} + 1) \times 50000} \quad ,$$

Then,

$$\text{RFH} \leqq \frac{-A + \sqrt{A^2 + B}}{\text{RUR} + \text{RDR} + 2}$$

However, A = RUS x (RUR + 1) + RDS x (RDR + 1),
B= {(RMG + 1) x 50000 x RMV - 2 x A x RFL + (RUR + RDR + 2) x RFL^2} x (RUR + RDR + 2)

(ii) Eliminate the linear acceleration section and set up a small linear deceleration range.

When

$$\text{RMV} \leqq \frac{(\text{RUS} + \text{RFL}) \times \{\text{RUS} \times (2 \times \text{RUR} + \text{RDR} + 3) + \text{RDS} \times (\text{RDR} + 1)\} \times 4}{(\text{RMG} + 1) \times 50000} \quad \text{and}$$

$$\text{RMV} > \frac{(\text{RDS} + \text{RFL}) \times \text{RDS} \times (\text{RUR} + \text{RDR} + 2) \times 8}{(\text{RMG} + 1) \times 50000} \quad ,$$

Change to S-curve acceleration/deceleration without any linear acceleration (RUS = 0, RDS > 0)

$$\text{RFH} \leqq \frac{-A + \sqrt{A^2 + B}}{2 \times \text{RUR} + \text{RDR} + 3}$$

However, A = RDS x (RDR + 1),
B= {(RMG + 1) x 50000x RMV - 2 x A x RFL + (2 x RUR + RDR + 3) x RFL^2} x (2 x RUR + RDR + 3)

(iii) Eliminate the linear acceleration/deceleration range

When

$$\text{RMV} \leqq \frac{(\text{RUS} + \text{RFL}) \times \text{RUS} \times (\text{RUR} + \text{RDR} + 2) \times 8}{(\text{RMG} + 1) \times 50000}$$

Change to S-curve acceleration/deceleration without any linear acceleration/deceleration (RUS = 0, RDS = 0),

$$\text{RFH} \leqq \sqrt{\frac{(\text{RMG} + 1) \times 50000 \times \text{RMV}}{(\text{RUR} + \text{RDR} + 2) \times 2} + \text{RFL}^2}$$

RMV: Positioning amount	RFL: Initial speed	RFH: Operation speed
RUR: Acceleration rate	RDR: Deceleration rate	RMG: Speed magnification rate
RUS: S-curve acceleration range	RDS: S-curve deceleration range	

7-4. Example of setting up an acceleration/deceleration speed pattern

Ex. When the start speed =10 pps, the operation speed =110 kpps, and the acceleration/deceleration time = 300 msec,

1) Select the 2x mode for multiplier rate in order to get 110 kpps output
RMG = 99 (63h)

2) Since the 2x mode is selected to get an operation speed 110 kpps,
RFH = 55000 (D6D8h)

3) In order to set a start speed to 10 pps, the rate magnification is set to the 2x mode.
RFL = 5 (0005h)

4) In order to make the acceleration/deceleration time 300 ms, set RUR = 26.275, from the equation for the acceleration time and RUR value.

$$\text{Acceleration time [s]} = \frac{(\text{RFH} - \text{RFL}) \times (\text{RUR} + 1) \times 8}{40{,}000{,}000}$$

$$0.3 = \frac{(55000 - 5) \times (\text{RUR} + 1) \times 8}{40{,}000{,}000}$$

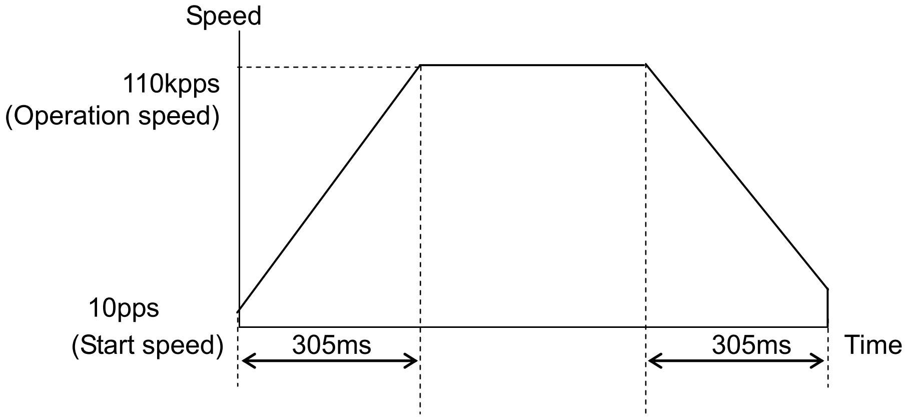
Then,

$$\text{RUR} = \frac{40{,}000{,}000 \times 0.3}{(55000 - 5) \times 8} - 1$$

RUR = 26.275

However, since only integers can be entered for RUR, use "26" or "27." The actual acceleration/deceleration time will be 297 msec if RUR = "26", or 308 msec if RUR = "27."

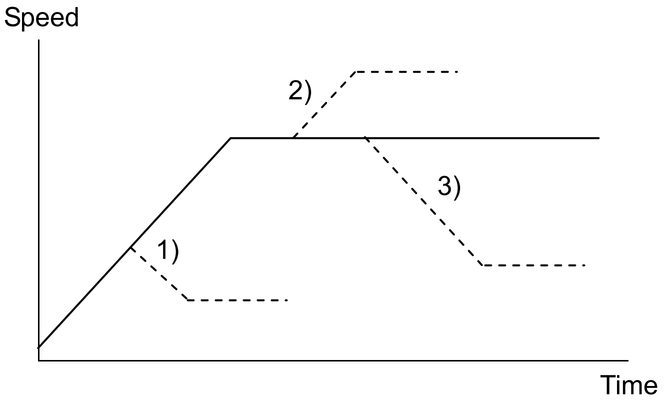
An example of the speed pattern when RUR = 27



7-5. Changing speed patterns while in operation

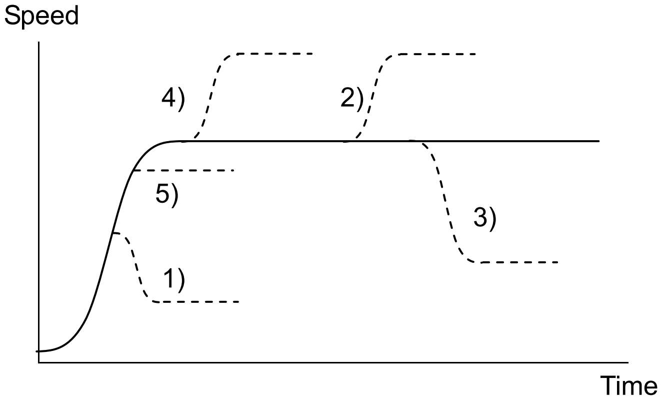
By changing RFH, RUR, RDR, RUS, or RDS registers during operation, the speed and acceleration can be changed on the fly. However, if the ramp-down point is set to automatic (RMD.MSDP = 0) for the positioning mode, do not change the values for RFL, RUR, RDR, RUS, or RDS. The automatic ramp-down point function will not work correctly.

An example of changing the speed pattern by changing the speed, during a linear acceleration/deceleration operation



1) Use a small RFH while accelerating the axis until it reaches the correct speed.
2), 3) Change RFH after the acceleration/deceleration is complete. The axis will continue accelerating or decelerating until it reaches the new speed.

An example of changing the speed pattern by changing the speed during S-curve acceleration/deceleration operation



1) Use a smaller RFH while accelerating and if ((change speed) < (speed before change)) and the axis will decelerate using an S-curve until it reaches the correct speed.
5) Use a smaller RFH while accelerating and if ((change speed) ≦ (speed before change)) and the axis will accelerate without changing the S-curve's characteristic until it reaches the correct speed.
4) Use a larger RFH while accelerating and the axis will accelerate to the original speed entered without changing the S-curve's characteristic. Then it will accelerate again until it reaches the newly set speed.
2), 3) If RFH is changed after the acceleration/deceleration is complete, the axis will accelerate/decelerate using an S-curve until it reaches the correct speed.

8. Description of the functions

8-1. Reset

After turning ON the power, make sure to reset the LSI before beginning to use it.
To reset the LSI, hold the $\overline{RST}$ terminal LOW while supplying at least 10 cycles of a reference clock signal.
After a reset, the status of the LSI will be as follows.

Item	Reset status (initial status)
Internal registers	0
Control commands	0
P0 to P7 terminals	Input terminal
$\overline{STA}$, $\overline{STA}$ terminals	HIGH
OUT, DIR terminals	HIGH
ERC terminal	HIGH
$\overline{BSY}$/PH1, $\overline{FUP}$/PH2, $\overline{FDW}$/PH3, $\overline{MVC}$/PH4 terminals	HIGH

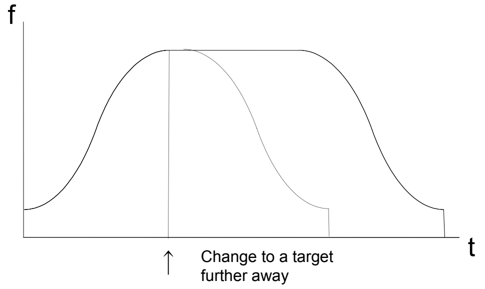
8-2. Position override

This LSI can override (change) the target position freely during operation.
There are two methods for overriding the target position.

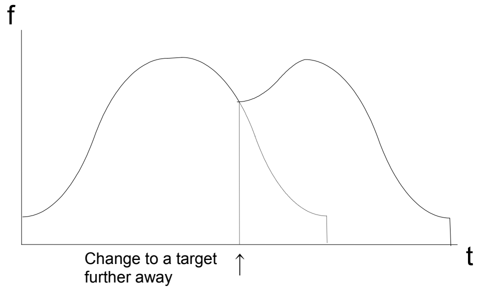
8-2-1. Target position override 1

By rewriting the target position data (RMV register value), the target position can be changed.
The starting position is used as a reference to change target position.

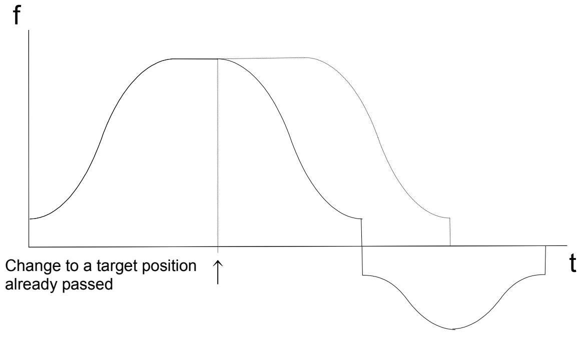
1) If the new target position is further away from the original target position during acceleration or constant speed operation, the motor will maintain the operation using the same speed pattern and it will complete the positioning operation at the position specified in the new data (new RMV value).



2) If the new target position is further away from the original target position during deceleration, the motor will accelerate from the current position to FH speed and complete the positioning operation at the position specified in the new data (new RMV value).
Assume that the current speed is Fu, and when RFL = Fu, a curve of next acceleration will be equal to a normal acceleration curve.

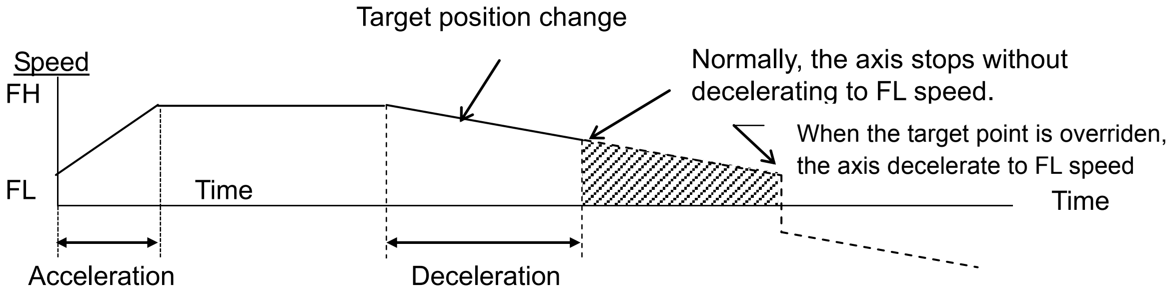


3) If the axis has already passed over the new target position,or the target position is changed to a position that is closer than the original position during deceleration, the motor will decelerate and stop. Then, the operation will reverse and complete the positioning operation at the position specified in the new data (new RMV value).



The motor accelerates/decelerates only when starting in high speed. The target position data (RMV register value) can be rewritten any number of times until the positioning operation is complete.

Note1: If the ramp-down point is set to automatic and the (deceleration time) > (acceleration time x 2), it may be the case that the axis cannot reduce the speed to the FL level, as shown below. In this case, if the target position is set closer than original position and the axis is decelerating, the axis will decelerate along the deceleration curve to the new override position, and then slow to the FL speed and finally stop. Then the axis will start moving to the new position.
Therefore, the axis will overrun the original target position during deceleration (shaded area).



To avoid creating an overrun condition, make sure that the deceleration time is less than two times the acceleration time, or if the deceleration time is more than double the acceleration time, set the ramp-down point manually.

Note 2: The position override is only effective during operation (FL/FH constant speed operation, during accelerating/decelerating, or during backlash correction).
If the speed is overridden before the motor stops, the speed override may not be accepted. If you need to override a speed just before a stop, you must determine if the G9103A can accept the override or not by the rotation position of the motor at the stopping.
By using write override (0080(h)) to RMV register, the center LSI outputs an interrupt request signal ($\overline{\text{INT}}$) when the G9103A fails to override. In this case too, you must determine whether the override is accepted or not by the stop position. The cause of the interrupt can be monitored in REST (error interrupt cause) register.

The center LSI recognized that override had failed when an override is written (0080(h)) to RMV register while the axis stops. Therefore, if you try to write an override (0080(h)) to RMV register before the axis starts, the center LSI outputs an interrupt request signal ($\overline{\text{INT}}$).

8-2-2. Target position override 2 (PCS signal)

By RMD.MPCS=1, the G9103A will perform positioning operations for the amount specified in RMV register, based on the timing of this command after the operation start (after it starts outputting command pulses) or on the "ON" timing of the PCS input signal.
The logic of a PCS input signal can be changed. The PCS terminal status can be monitored using RSTS register (extension status).

Setting pulse control using the PCS input <RMD.MPCS (bit 13)> 1: Positioning for the number of pulses stored in RMV, starting from the time at which the PCS input signal is turned ON.	[RMD] (WRITE) 15 8 - - n - - - - -
Setting the PCS input logic <RENV1.PCSL (bit 24)> 0: Negative logic 1: Positive logic	[RENV1] (WRITE) 31 24 - - - - - - - n
Reading the PCS signal <RSTS.SPCS (bit 14)> 0: Turn OFF PCS 1: Turn ON PCS	[RSTS] (READ) 15 8 - n - - - - - -
Substitute for PCS input <Command:STAON> Perform processes that are identical to those performed by supplying a PCS signal.	[Command] 0028(h)

8-3. Output pulse control

8-3-1. Output pulse mode

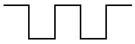
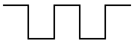
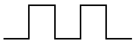
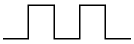
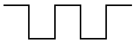
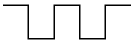
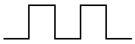
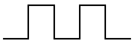
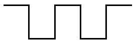
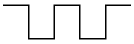
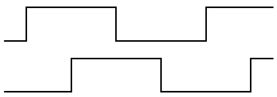
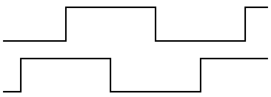
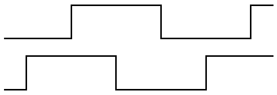
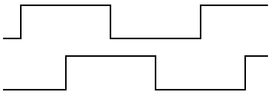
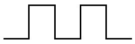
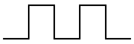
There are four types of common command pulse modes, two types of Two-pulse modes and two types of 90-degree phase difference pulse mode.

Common pulse mode:	This mode is to output operation pulses from OUT terminal and the direction signal from DIR terminal.
Two-pulse mode:	This mode is to output positive direction operation pulses from OUT terminal, and outputs negative direction operation pulses from DIR terminal.
90 degree phase Difference pulse mode:	This mode is to output 90 degree phase difference pulses through OUT and DIR terminals. One 90-degree phase difference is equivalent to 4 pulses in the common or two-pulse mode.

The output mode for command pulses is set in REMV.PMD.
If motor drivers using the common pulse mode need a lag time (since the direction signal changes, until receiving a command pulse), use a direction change timer.
When REMV.DTMF=0, the operation can be delayed for one direction change timer unit (0.2 msec), after changing the direction signal.

When RENV.MREV=1, the wave profiles of "When feeding in the positive direction", and "When feeding in the negative direction" in the following figure are switched and the direction of the motor rotation can be reversed. (The direction of the motor rotation is reversed when excitation sequence is output.)

Setting the pulse output mode(RENV1.MREV=0) <Set RENV1.PMD2～0(bit0 to 2)>

PMD2 to 0	When feeding in the positive direction		When feeding in the negative direction	
	OUT output	DIR output	OUT output	DIR output
000		High		Low
001		High		Low
010		Low		High
011		Low		High
100		High	High	
101	OUT DIR 		OUT DIR 	
110	OUT DIR 		OUT DIR 	
111		Low	Low	

[RENV1] (WRITE)

7							0
-	-	-	-	-	n	n	n

Setting the direction change timer (0.2 msec) function <RENV1.DTMF (bit 26)>
0: ON
1: OFF

[RENV1] (WRITE)

31							24
-	-	-	-	-	n	-	-

Setting the direction of motor rotation <Set to RENV1.MREV(bit 30)>
0: Normal direction
1: Reverse direction

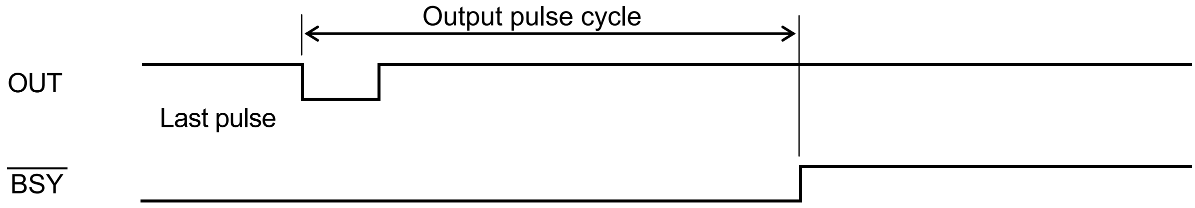
[RENV1] (WRITE)

31							24
-	n	-	-	-	-	-	-

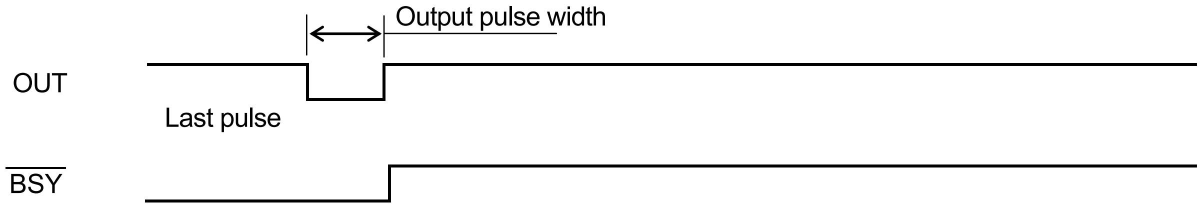
8-3-2. Control the output pulse width and operation complete timing

In order to put the timing of the stopping forward, this LSI controls the output pulse width.
When the output pulse speed is slower than approximately 2.44Kpps, the pulse width is constant (Approx. 200 μsec). For faster pulse speeds than this, the duty cycle is kept constant (approx. 50%). By setting RENV.PDTC=1, the output pulse width can be set to make duty cycle (50%) constant.
Also, when setting RMD.METM, the operation complete timing can be changed.

1) When RMD.METM = 0 (the point at which the output pulse cycle is complete)



2) When RMD.METM = 1 (when the output pulse is OFF)

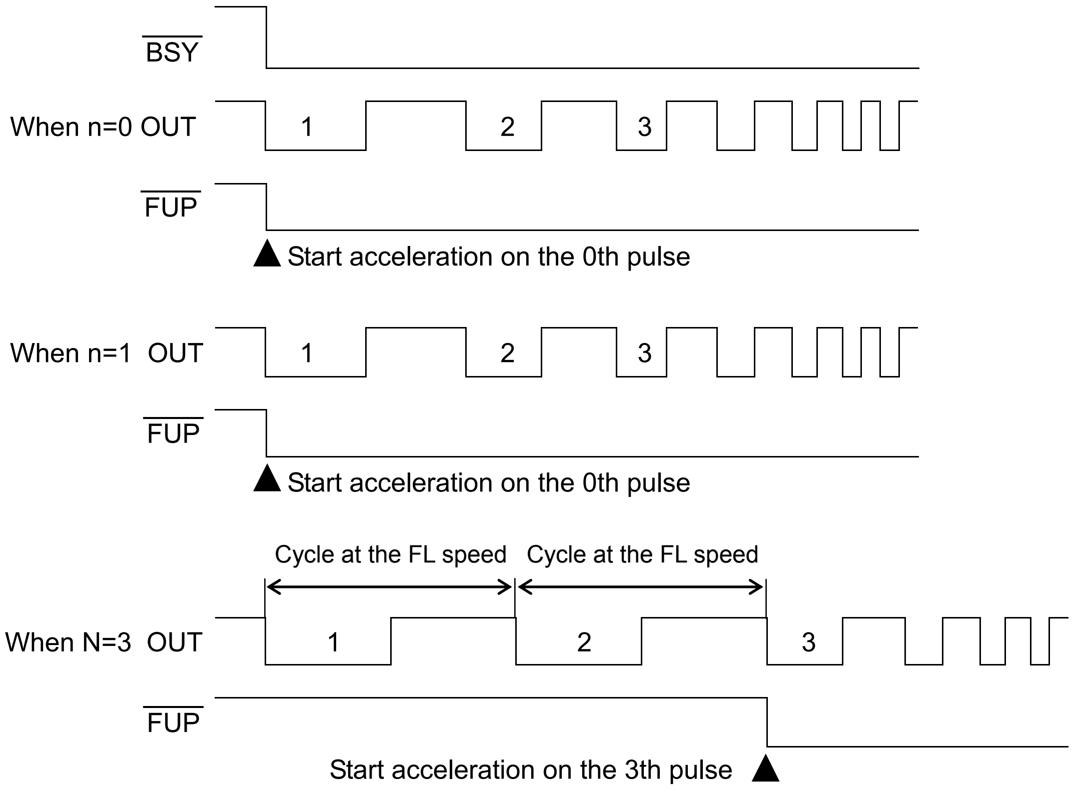


Setting the operation complete timing <RMD.METM (bit 11)> 0: At the end of a cycle of output pulse 1: Complete when the output pulse turns OFF.	[RMD] (WRITE) 15 8 - - - - n - - -
Setting the output pulse width <RENV1.PDTC (bit 29)> 0: Automatically switch between a constant output pulse and a constant duty cycle (approx. 50%) in accord with variations in speed. 1: Keep the output pulse width at a constant duty cycle (approx. 50%).	[RENV1] (WRITE) 31 24 - - n - - - - -

8-4. Idling control

When starting acceleration or deceleration operation, it can be started after the output of a few pulses at FL speed. This pulse is referred to as "idling pulse". Set the number of pulses for idling in RENV5.IDL.
If you do not use this function, enter a value "n" of 0 or 1. The LSI will start the acceleration at the same time it begins outputting pulses. Therefore, the start speed obtained from an initial 2-pulse frequency will be faster than the FL speed.
To use this function, enter a value "n" of 2 to 7. The LSI will start the acceleration by beginning its output on the "n"th pulse. Therefore, the start speed will be the FL speed and the FL speed can be set to start automatically at upper speed limit.
If this function is used with the positioning mode, the total feed amount will not change.

[Setting idling pulses and the acceleration start timing]



▲ Start acceleration on the 0th pulse

▲ Start acceleration on the 0th pulse

Start acceleration on the 3th pulse ▲

Set the number of idling pulses < RENV2.IDL0 to 2 (bits 20 to 22)> Specify the number of idling pulses, from 0 to 7. Start accelerating at FL speed after outputting the specified number of pulses.	[RENV2] (WRITE) 23 16 - n n n - - - -
Read the idling control counter value <RSPD.IDC0 to 2 (bits 24 to 26)> Read the idling control counter.	[RSPD] (READ) 31 24 - - - n n n

8-5. Mechanical external input control

8-5-1. +EL, -EL signal

When an end limit signal (a +EL signal when feeding in the + direction) in the feed direction turns ON while operating, the axis will stop immediately or decelerate and stop. After stopping, even if the EL signal is turned OFF, the axis will remain stopped. For safety, keep the EL signal ON until the axis reaches the end of the stroke.
If the EL signal is ON when writing a start command, the axis cannot start moving in the direction of the particular EL signal that is ON.
By setting RENV1.ELM, the stopping pattern for use when the EL signal is turned ON can be selected from immediate stop or deceleration stop (with high speed start only). However, when the deceleration stop is selected, please note to have room mechanically because the motor stops after passing through the EL position.

The minimum pulse width of the EL signal is 4 μsec when the input filter is ON. When the input filter is turned OFF, the minimum pulse width is 0.1 μsec.
The EL signal can be monitored by reading RSTS (extension status) register.
By reading REST register, you can check for an error interrupt caused by the EL signal turning ON.
When in the timer mode, this signal is ignored. Even in this case, the EL signal can be monitored by reading RSTS (extension status) register.
The input logic of the EL signal can be set for each axis using the ELL input terminal.

Set the input logic of the ±EL signal <ELL input terminal> L: Positive logic input H: Negative logic input	
Stop method to be used when the ±EL signal turns ON <RENV1.ELM (bit 3)> 0: Immediate stop by turning ON the ±EL signal 1: Deceleration stop by turning ON the ±EL signal	[RENV1] (WRITE) 7 0 - - - - n - - -

Reading the ±EL signal <RSTS.SPEL (bit 6), SMEL (bit 7)>
SPEL = 0:Turn OFF the +EL signal SPEL = 1: Turn ON the +EL signal
SMEL = 0:Turn OFF the -EL signal SMEL = 1: Turn ON the -EL signal
	[RMD] (READ)
	7 0
	n n - - - - - -

Setting the ±EL input filter <RENV1.FLTR (bit 25)> 0: Apply a filter to the ±EL and ORG input Apply a filter and any signals shorter than 4 μsec pulse width are ignored.	[RENV1] (WRITE) 31 24 - - - - - - n -

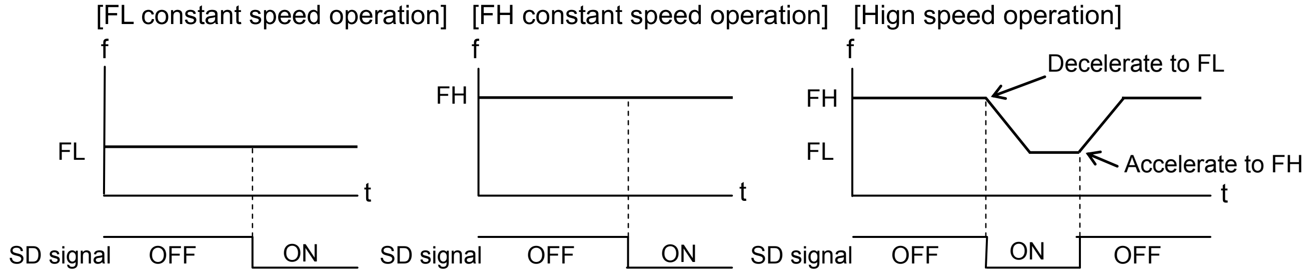
Note 1: Operation after turning ON the EL signal may be different from the above for the origin return operation (6-4-1), the origin search operation (6-4-3), and the EL or SL operation mode (6-5). See the description of each operation mode.

8-5-2. SD signal

When RMD.MSDE=0, the SD signal is ignored.
If the SD signal input is enabled and the SD signal is turned ON while in operation, the axis will: 1) decelerate, 2) latch and decelerate, 3) decelerate and stop, or 4) latch and perform a deceleration stop, according to the setting of RENV1.SDM and RENV1.SDLT.

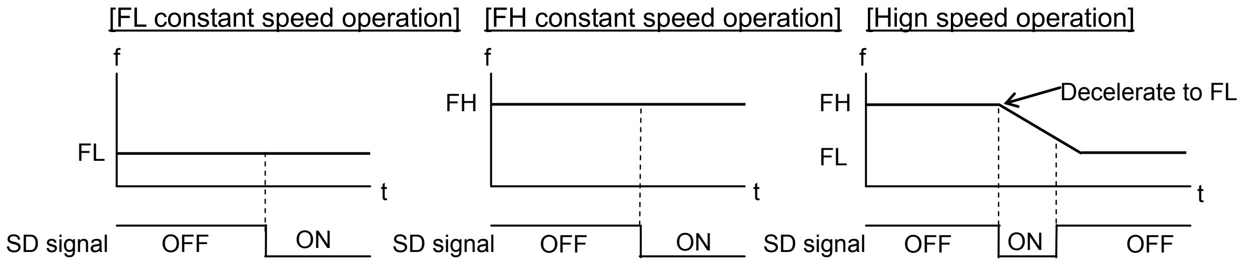
1) Deceleration <RENV1.SDM = 0, RENV1.SDLT = 0>

- While feeding at constant speed, the SD signal is ignored. While in high speed operation the axis decelerates to the FL speed when the SD signal is turned ON. After decelerating, or while decelerating, if the SD signal turns OFF, the axis will accelerate to the FH speed.
- If the SD signal is turned ON when the high speed command is written, the axis will operate at the FL speed. When the SD signal is turned OFF, the axis will accelerate to the FH speed.



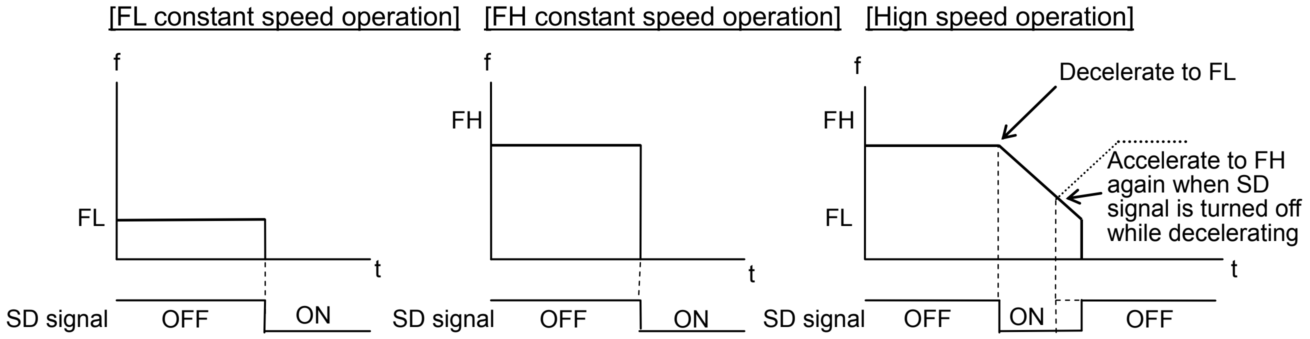
2) Latch and decelerate <RENV1.SDM=0, RENV1.SDLT=1>

- While feeding at constant speed, the SD signal is ignored. While in high speed operation, the axis decelerates to FL speed by turning the SD signal ON. Even if the SD signal is turned OFF after decelerating or while decelerating, the axis will continue moving at the FL speed and will not accelerate to the FH speed.
- If the SD signal is turned ON while writing a high speed command, the axis will feed at FL speed. Even if the SD signal is turned OFF, the axis will not accelerate to FH speed.



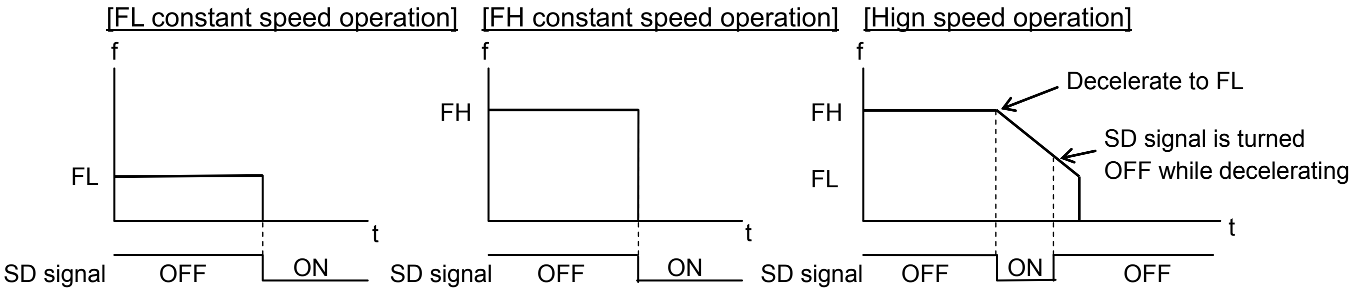
3) Deceleration stop <RENV1.SDM=1, RENV1.SDLT = 0>

- If the SD signal is turned ON while in constant speed operation, the axis stops. While in high speed operation, the axis decelerates to FL speed when the SD signal is turned ON, and then stops. If the SD signal is turned OFF during deceleration, the axis accelerates to FH speed.
- If the SD signal is turned ON after writing a start command, the axis completes its operation without another start.
- The center LSI outputs an interrupt request ($\overline{\text{INT}}$) when the axis stops. (REST.ESSD=1)


[FL constant speed operation]
[FH constant speed operation]
[Hign speed operation]
f
FL
t
SD signal
OFF
ON
f
FH
t
SD signal
OFF
ON
f
FH
FL
Decelerate to FL
Accelerate to FH again when SD signal is turned off while decelerating
t
SD signal
OFF
ON
OFF

4) Latched, deceleration stop <RENV1.SDM = 1, RENV1.SDLT=0>

- If the SD signal is turned ON while in constant speed operation, the axis stops. If the SD signal is turned ON while in high speed operation, the axis decelerates to FL speed and then stop. Even if the SD signal is turned OFF during deceleration, the axis does not accelerate.
- If the SD signal is turned ON while writing a start command, the axis completes its operation without another start.
- The center LSI outputs an interrupt request signal ($\overline{\text{INT}}$) when the axis stops. (REST.ESSD=1)


[FL constant speed operation] [FH constant speed operation] [Hign speed operation]

The input logic of the SD signal can be changed. If the latched input is set to accept input from the SD signal, and if the SD signal is OFF at the next start, the latch will be reset. The latch is also reset when the latch input (RENV1.SDLT) is set to zero.
The minimum pulse width of the SD signal is 4 µsec when the input filter is ON. When the input filter is turned OFF, the minimum pulse width is 0.1 µsec.
The latch signal of the SD signal can be monitored by reading RSTS (extension status) register. The SD signal terminal status can be monitored by reading RSTS (extension status) register. By reading REST register, you can check for an error interrupt caused by the SD signal turning ON.

Enable/disable SD signal input <RMD.MSDE (bit 8)> 0: Enable SD signal input 1: Disable SD signal input	[RMD] (WRITE) 15 8 - - - - - - - n
Input logic of the SD signal <RENV1.SDL (bit 6)> 0: Negative logic 1: Positive logic	[REMV1] (WRITE) 7 0 - n - - - - - -
Set the operation pattern when the SD signal is turned ON <RENV1.SDM (bit 4)> 0: The motor decelerates on receiving the SD signal and feeds at FL constant speed 1: The motor decelerates and stops on receiving the SD signal.	[RENV1] (WRITE) 7 0 - - - n - - - -
Select the SD signal input type <RENV1.SDLT (bit 5)> 0: Level input 1: Latched input To release the latch, turn OFF the SD input when next start command is written or select level input.	[RENV1] (WRITE) 7 0 - - n - - - - -
Reading the latch status of the SD signal <RSTS.SSD (bit 9)> 0: The SD latch signal is OFF 1: The SD latch signal is ON	[RENV1] (WRITE) 7 0 - - - - - - n -
Reading the SD signal <RSTS.SDIN (bit 10)> 0: The SD signal is OFF 1: The SD signal is ON	[RSTS] (READ) 15 8 - - - - - n - -
Reading the cause of an interrupt when stopped by the SD signal <REST.ESSD (bit 8)> 1: Deceleration stop caused by the SD signal turning ON	[REST] (READ) 15 8 - - - - - - - n
Apply an input filter to SD input <RENV1.FLTR (bit 25)> 0: Apply a filter to the SD input 1: Apply no filter. By applying a filter, signals with a pulse width of 4 µsec or less will be ignored.	[RENV1] (WRITE) 31 24 - - - - - - n -

8-5-3. ORG, EZ signals

These signals are enabled in the origin return modes (origin return, leaving origin position, and origin position search) and in the EZ count operation modes. Specify the operation mode and the operation direction using RMD register (operation mode).

The G9103A latches the ORG signal on the rising edge of the output pulse (negative logic). The minimum pulse length of the ORG signal is one cycle period of the output pulses.

Since the ORG signal input is latched internally, there is no need to keep the external signal ON.

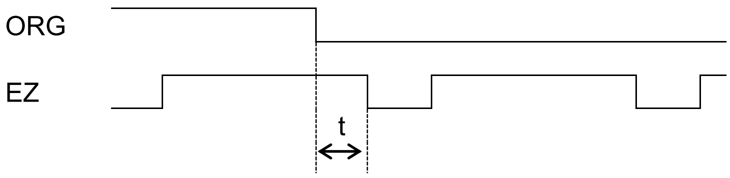
The ORG latch signal is reset when the axis stops.

The input logic of the ORG signal can be changed by RENV1.ORGL and EZ signals can be changed using RENV2.EZL.

By reading RSTS (extension status) register, you can monitor the status of ORG and EZ terminals.

For details about the origin return operation modes, see 6-4, "Origin position operation mode."

ORG signal and EZ signal timing



(i) When t ≧ 100μs, the EZ count is counted.

(ii) When 50μs < t < 100μs, it is undetermined to count the EZ signal.

(iii) When t 50Tμs, the EZ count is not counted.

Enabling the ORG and EZ signals <RMD.MOD (bits 0 to 6)> 001 0000: Origin return in the positive direction 001 0010: Leaving origin position in the positive direction 001 0101: Origin position search in the positive direction 010 0100: EZ count operation in the positive direction 001 1000: Origin return in the negative direction 001 1010: Leave origin position in the negative direction 001 1101: Origin position search in the negative direction 010 1100: EZ count operation in the negative direction	[RMD] (WRITE) 7 0 0 n n n n n n n
Set the origin return method <RENV3.ORM0 to 3 (bits 0 to 3)> See the RENV3 register description	[REMV3] (WRITE) 7 0 - - - - n n n n
Set the input logic for the ORG signal <RENV1.ORGL (bit 7)> 0: Negative logic 1: Positive logic	[RENV1] (WRITE) 7 0 n - - - - - - -
Read the ORG signal <RSTS.SORG (bit 8)> 0: The ORG signal is OFF 1: The ORG signal is ON	[RSTS] (READ) 15 8 - - - - - - - n
Set the EZ count number <RENV3.EZD0 to 3 (bits 4 to 7)> Set the origin return completion condition and the EZ count number for counting. in EZ Specify the value (the number to count to -1) in EZD0 to 3. The setting range is 0 to 15.	[RENV3] (WRITE) 7 0 n n n n - - - -
Specify the input logic of the EZ signal <RENV2.EZL (bit 12)> 0: Falling edge 1: Rising edge	[RENV2] (WRITE) 15 8 - - - n - - - -

Read the EZ signal <RSTS.SEZ (bit 16)> 0: The EZ signal is OFF 1: The EZ signal is ON	[RSTS] (READ) 23 16 - - - - - - - n
Apply an input filter to EZ input <RENV2.EINF (bit 8)> 0: Apply a filter to the EZ input 1: Apply no filter. By applying a filter, input signal shorter than 6 cycles of the CLK input will be ignored.	[RENV2] (WRITE) 15 8 - - - - - - - n

8-6. Motor driver I/F signal

8-6-1. INP signal (for servomotor)

The servo driver systems to accept pulse strings input have a deviation counter to count the difference between command pulse inputs and feedback pulse inputs. The driver controls motor so as to adjust the difference to zero. In other words, the servomotors operates behind command pulses and, even after the command pulses stop, the servomotor systems keep feeding until the count in the deviation counter reaches zero.

This LSI can receive a positioning complete signal (INP signal) from a servo driver in place of the pulse output complete timing to determine when an operation is complete.
When the INP signal input is used to indicate the completion status of an operation, the $\overline{\text{BSY}}$ signal when an operation is complete, the stop conditions of the main status (bit 0 to 3 and 8 of the MSTS), and the operation status of the extension status (RSTS.CND0 to 3) will also change when the INP signal is input.

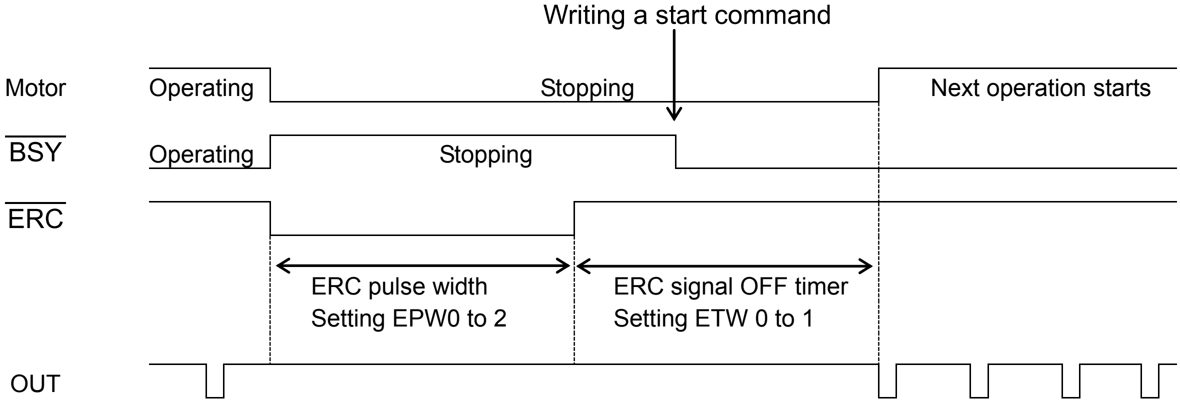
The input logic of the INP signal can be changed by setting RENV.INPL..
The minimum pulse width of the INP signal is 4 μsec when the input filter is ON. If the input filter is OFF, the minimum pulse width will be 0.1 μsec.
If the INP signal is already ON when the G9103A is finished outputting pulses, it treats the operation as complete, without any delay.
The INP signal can be monitored by reading the RSTS register (extension status).

Set the operation complete delay using the INP signal <RMD.MINP (bit 9)> 0: No operation complete delay waiting for the INP signal. 1: Operation complete (status, $\overline{\text{BSY}}$) delay until the INP signal goes ON.	[RMD] (WRITE) 15 8 - - - - - - n -
Input logic of the INP signal <RENV1.INPL (bit 22)> 0: Negative logic 1: Positive logic	[RENV1] (WRITE) 23 16 - n - - - - - -
Reading the INP signal <RSTS.SINP (bit 19)> 0: The INP signal OFF 1: The INP signal ON	[RSTS] (READ) 23 16 - - - - n - - -
Set the INP input filter <RENV1.FLTR (bit 25)> 0: Apply a filter to the INP input. By applying a filter, pulses less than 4 μsec in width are ignored.	[RENV2] (WRITE) 31 24 - - - - - - n -

Note. When you want to continue interpolation operation using pre-register, the operation time of all interpolated axes should be same. Therefore, do not use delay by the INP signals, 8-6-2. ERC signal (for servomotor)

The servomotor does not stop until the deviation counter in the servo driver reaches “0” even if command pulse is stopped. In order to stop the servomotor immediately when completing origin return operation, etc., it is necessary to clear the deviation counter in the servo driver.
This LSI can output the ERC signal to clear the deviation counter in the servo driver. The ERC signal is output as one-shot signal or level signal and can be selected by RENV1.EPW. If a lag time is needed until the servo driver accept command pulses after the ERC signal returns to high level (off status), the time for the ERC signal OFF timer can be set by RENV1.ETW.

Note. ERC signal is output from the ERC/CDWN terminal. Only when RENV1.CDWN=0, ERC signal is output. When RENV1.CDWN=1, CDWN signal is output.



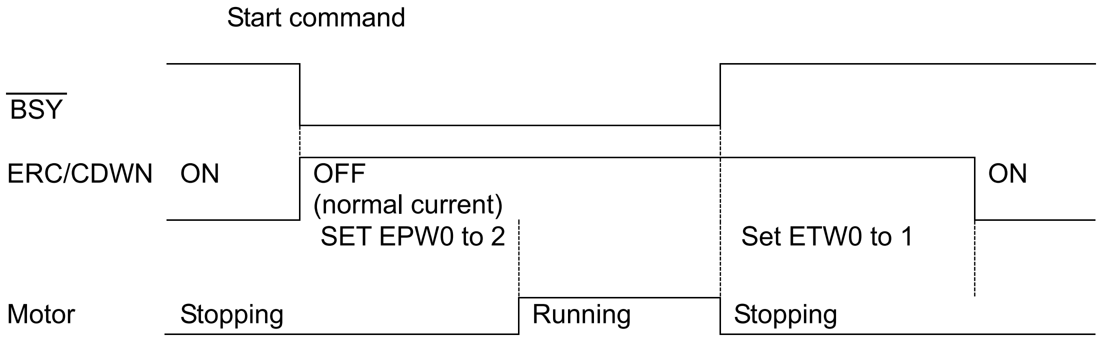
In order to output the ERC signal at completing origin return operation, set the ERC signal to be output automatically by setting RENV1.EROR=1. For the output timing of the ERC signal, see the timing wave shape in “6-4-1. Origin return operation”.
In order to output the ERC signal at the stopping by inputting of the EL signal, ALM signal or EMG signal, and the immediate stopping by emergency stop command (0005(h)), set the ERC signal to be output automatically by setting RENV1.EROE = 1. (When the motor decelerates and stops, the ERC signal is not output even if automatic output is selected.)
The ERC signal can be output by writing the ERC output command (0024(h)),
The output logic of the ERC signal can be changed by the setting RENV 1 register and the ERC signal can be monitored by reading RSTS register (extension status).

Select ERC/CDWN signal output <RENV1.CDWN (bit 31)> 0: The ERC signal is output from ERC/CDWN terminal. 1: The CDWN signal is output from ERC/CDWN terminal.	[REMV1] (WRITE) 31 24 | n | - | - | - | - | - | - | - |
Set to output the ERC signal automatically <RENV1.EROE (bit 10)> 0: The ERC signal is not output at the stop by inputting EL, ALM, EMG signals. 1: The ERC signal is automatically output at the stop by inputting EL, ALM and EMG signal.	[REMV1] (WRITE) 15 8 | - | - | - | - | - | n | - | - |
Set to output the ERC signal automatically <RENV1.EROR (bit 11)> 0: The ERC signal is not output at completing origin return. 1: The ERC signal is output automatically at completing origin return.	[RENV1] (WRITE) 15 8 | - | - | - | - | n | - | - | - |
Set the width of the ERC signal output <RENV1.EPW 0 to 2 (bit 12 to 14)> 000:12μs 100:12ms 001:93μs 101:48ms 010:371μs 110:95ms 011:1.5ms 111:Level output	[RENV1] (READ) 15 8 | - | n | n | n | - | - | - | - |
Output logic of the ERC signal <RENV1.ERCL (bit 15)> 0: Negative logic 1: Positive logic	[RENV1] (WRITE) 15 8 | n | - | - | - | - | - | - | - |
Set the time for the ERC signal off timer <RENV1.ETW 0 to 1(bit 16 to 17)> 00:0μs 10:1.5ms 01:12μs 11:95ms	[RENV1] (WRITE) 23 16 | - | - | - | - | - | - | n | n |
Reading the ERC signal <REST.SERC (bit 15)> 0: ERC signal OFF 1: ERC signal ON	[RSTS] (READ) 15 8 | n | - | - | - | - | - | - | - |
Emergency stop command <Command: CMEMG> G9103A outputs the ERC signal after emergency stop.	[Command] 0005(h)
ERC signal output command <Command: ERCOUT> G9103A turn ON the ERC signal	[Command] 0024(h)
Reset command for output the ERC signal <Command: ERCRST> G9103A turns OFF the ERC signal	[Command] 0025(h)

8-6-3. CDWN signal (for stepper motor)

In control stepper motor, there is a case that you want to turn current down to reduce motor's heat generation at stand still. (Current down control)
This LSI has a function for outputting current down signals (CDWN).
By writing a start command, turn the CDWN signal off (normal curren). After the setting time has elapsed, start the motor. If the setting time elapses after the motor stops, this LSI turns CDWN signal ON (Current down).
When you write the next start command while the CDWN signal still be off after motor stops, this LSI starts to operate the motor immediately.
The timer at the start and stop doubles as the timer to control ERC. However, The time range that you can select for EPW and ETW is different from the timer range to control ERC.

Note. The CDWN signal is output from the ERC/CDWN terminal. When RENV1.CDWN=1, the CDWN signal is output and when RENV1.CDWN=0, the ERC signal is output.



The output logic of the CDWN signal can be changed by setting RENV1 register and the CDWN signal can be monitors by reading RSTS (extended status) register.

Select ERC/CDWN signal output <RENV1.CDWN (bit 31)> 0: The ERC signal is output from ERC/CDWN terminal. 1: The CDWN signal is output from ERC/CDWN terminal.	[REMV1] (WRITE) 31 24 n - - - - - - -
Set the time to return current <RENV1.EPW0 to 2 (bit 12 to 14)> 000: 6.4ms 100:102ms 001: 13ms 101:205ms 010: 26ms 110: Always CDWN=OFF 011: 51ms 111: This setting is prohibited.	[REMV1] (WRITE) 15 8 - n n n - - - -
Set the output logic of the CDWN signal <RENV1.ERCL (bit 15)> 0: Negative logic 1: Positive logic	[REMV1] (WRITE) 15 8 n - - - - - - -
Set the time to delay current down <RENV1.ETW0 to 1 (bit 16 to 17)> 00: 51ms 10: 205ms 01: 102ms 11: 410ms	[REMV1] (WRITE) 23 16 - - - - - - n n
Read the CDWN signal <RSTS.SERC (bit 15)> 0: CDWN signal OFF 1: CDWN signal ON	[REMV1] (WRITE) 15 8 n - - - - - - -

8-6-4. ALM signal

Input the alarm signal (ALM signal). If the ALM signal turns ON while operating, the motor stops immediately or decelerates and stops. However, the motor decelerates and stops by inputting the ALM signal only at the high-speed start. At the constant speed start, the motor stops immediately.
In the case of deceleration stop, keep the alarm input ON until stopping. If the ALM signal is ON at writing the start command, any pulses are not output.
The minimum pulse width of the ALM signal is 4 μsec when the input filter is ON. If the input filter is OFF, the minimum pulse width is 0.1 μsec.
The input logic of the ALM signal can be changed and the ALM signal can be monitored by reading RSTS register (extension status).

Set the method to stop the motor when the ALM signal turns ON. <Set REMV1.ALMM (bit 8)> 0: Immediately stop when the ALM signal goes ON. 1: Decelerate and stop when the ALM signal goes ON. (Only with high-speed start)	[RMD] (WRITE) 15 8 - - - - - - - n
Input logic of the INP signal <Set RENV1.ALML (bit 9)> 0: Negative logic 1: Positive logic	[RENV1] (WRITE) 15 8 - - - - - - n -
Read the ALM signal <RSTS.SALM (bit 5)> 0: The ALM signal OFF 1: The ALM signal ON	[RSTS] (READ) 7 0 - - n - - - - -
Read the stop cause when the ALM signal turns ON <RSTS.ESAL (bit 5)> 1: Stop by the ALM signal ON	[RSTS] (READ) 7 0 - - n - - - - -
Set the ALM input filter <RENV1.FLTR (bit 25)> 0: Apply a filter to the ALM input. By applying a filter, pulses less than 4 μsec in width are ignored.	[RENV1] (WRITE) 31 24 - - - - - - n -

8-7. External start, simultaneous start

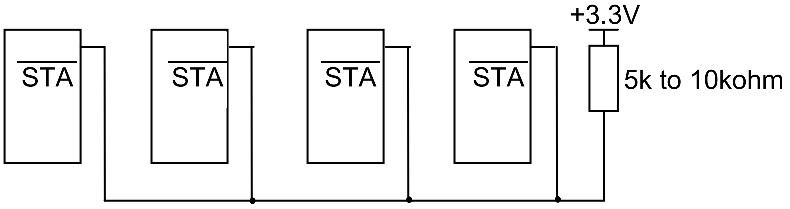
This LSI can start operation by an external signal using $\overline{\text{STA}}$ terminals. Set RMD.MSY=1 and the LSI will start feeding when the $\overline{\text{STA}}$ turns LOW.
When you want to control multiple axes using more than one LSI, connect the $\overline{\text{STA}}$ terminal on each LSI with another $\overline{\text{STA}}$ terminal and set the axes to "waiting for $\overline{\text{STA}}$ input", to start them all at the same time. In spite of the external signal, start signal can be output through $\overline{\text{STA}}$ inal.
The input logic on $\overline{\text{STA}}$ terminals cannot be changed.
By setting RIRQ register (event interrupt cause), an interrupt occurs together with a simultaneous start (when the $\overline{\text{STA}}$ input is ON).
By reading RIST register, the cause of an event interrupt can be checked. The operation status (waiting for $\overline{\text{STA}}$ input), and status of the $\overline{\text{STA}}$ terminal can be monitored by reading RSTS register.

<How to make a simultaneous start>
Set RMD.MSY=1 for the axes you want to start. Write a start command and put the axes in the "waiting for $\overline{\text{STA}}$ input" status. Then, start the axes simultaneously by either of the methods described below.

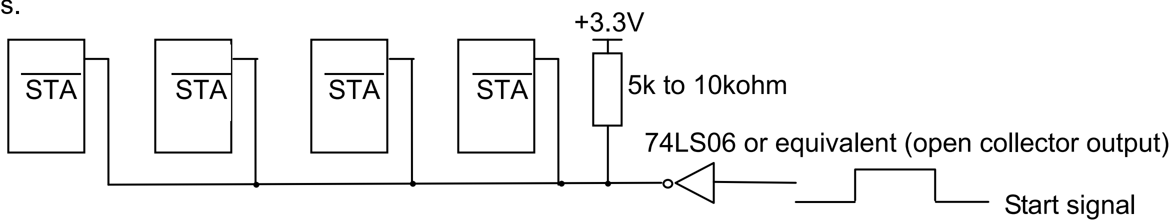
1) By writing a simultaneous start command (CMSTA:0006h), the LSI will output a one shot signal of approx. 0.4 μsec from $\overline{\text{STA}}$ terminal.
2) Input hardware signal from outside.
   Supply a start signal by driving the terminal with open collector output (74LS06 or equivalent).

$\overline{\text{STA}}$ signal can be supplied as level trigger or edge trigger inputs. However, when level trigger input is selected, if $\overline{\text{STA}}$=L and a start command is written, the axis starts immediately.
After connecting terminal on each LSI, each axis can still be started independently using start commands.
To release the "waiting for $\overline{\text{STA}}$ input" condition, write an immediate stop command (0049(h)).

1) To start axes controlled by different LSIs simultaneously, connect the LSIs as follows.



2) To start simultaneously from an external circuit, or use a single axis as an external start, connect the LSIs as follows.



For start signal, supply a one shot input signal with a pulse width of at least approx. 0.2 μsec.

$\overline{\text{STA}}$ input <RMD.MSY (bits 14)> 01: Start by inputting a $\overline{\text{STA}}$ signal	[RMD] (WRITE) 15 8 - n - - - - - -
Specify the input specification for the $\overline{\text{STA}}$ signal <RENV1.STAM (bit 18)> 0: Level trigger input for the $\overline{\text{STA}}$ signal : Edge trigger input for the $\overline{\text{STA}}$ signal	[RENV1] (WRITE) 23 16 - - - - - n - -
Read the $\overline{\text{STA}}$ signal <RSTS.SSTA (bit 11)> 0: The $\overline{\text{STA}}$ signal is OFF (H level) 1: The $\overline{\text{STA}}$ signal is ON (L level)	[RSTS] (READ) 15 8 - - - - n - - -
Read the operation status <RSTS.CND (bits 0 to 3)> 0001: Waiting for $\overline{\text{STA}}$ input	[RSTS] (READ) 7 0 - - - - n n n n
Set an event interrupt cause <RIRQ.IRSA (bit 12)> 1: Generates an interrupt when the $\overline{\text{STA}}$ input is ON.	[RIRQ] (WRITE) 15 8 - - - n - - - -
Reading the event interrupt cause <RIST.ISSA (bit 12)> 1: When the $\overline{\text{STA}}$ signal is ON.	[RSTS] (READ) 15 8 - - - n - - - -
Simultaneous start command <Command: CMSTA> Output a one shot pulse of 0.4μs from $\overline{\text{STA}}$ terminal. ($\overline{\text{STA}}$ terminal is bi-directional and inputs the output signal again.)	[Command] 0006(h)
For own axis only, simultaneous start command <Command: SPSTA> Used the same way as when an $\overline{\text{STA}}$ signal is supplied, for own axis only.	[Command] 002A(h)

8-8. External stop / simultaneous stop

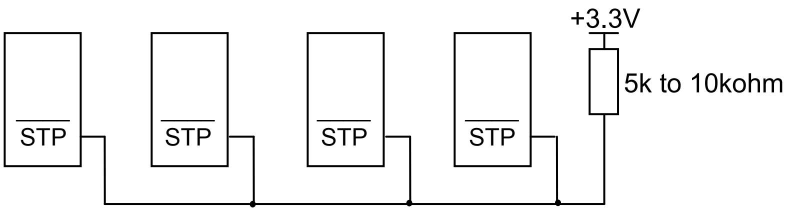
This LSI can execute an immediate stop or a deceleration stop triggered by an external signal using $\overline{\text{STP}}$ terminal. Set RMD.MSPE=1 to enable a stop from a $\overline{\text{STP}}$ input. The axis stops immediately or decelerates and stops when STP terminal is LOW. However, a deceleration stop is only used for a high speed start. When the axis starts at constant speed, the signal on $\overline{\text{STP}}$ terminal causes an immediate stop.
The input logic of the $\overline{\text{STP}}$ terminal cannot be changed.
When multiple LSIs are used to control multiple axes, connect the $\overline{\text{STP}}$ terminal on each LSI with another $\overline{\text{STP}}$ terminal and input the same signal so that the axes that are set to stop on a $\overline{\text{STP}}$ input can be stopped simultaneously. In this case, a stop signal can also be output from the $\overline{\text{STP}}$ terminal.
When an axis stops because the $\overline{\text{STP}}$ signal turns ON, an interrupt occurs. By reading REST register, you can check the cause of an error interrupt. You can monitor $\overline{\text{STP}}$ terminal status by reading RSTS register (extension status).

<How to make a simultaneous stop>
Set RMD.MSPE=1 for each of the axes that you want to stop simultaneously. Then start these axes.
Stop these axes using any of the following three methods.

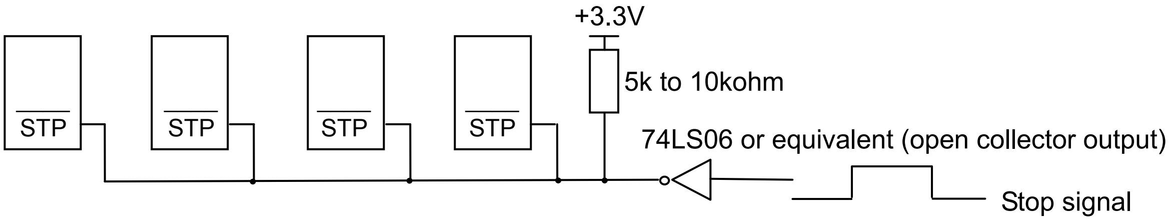
1) By writing a simultaneous stop command, the $\overline{\text{STP}}$ terminal will output a one shot signal that is approx. 0.4 μsec.
2) Input an external hardware signal
   Input a hardware signal using an open collector output (74LS06 or equivalent).
3) The $\overline{\text{STP}}$ terminal will output a one shot signal that is approximately 0.4 μsec when a stop caused by an error occurs on an axis that has RMD.MSPO=1.

Even when $\overline{\text{STP}}$ terminals on LSIs are connected together, each axis can still be stopped independently by using the stop command.
Without using $\overline{\text{STP}}$ terminals, stop command for broadcast communication can stop axes simultaneously. (Please see 5-1-2-4. Broadcast communications.)

1) Connect the terminals as follows for a simultaneous stop among numeral LSIs.



2) To stop simultaneously using an external circuit, connect as follows.



As a stop signal, input a one shot signal that is more than 0.2 μsec in length.

Setting to enable $\overline{\text{STP}}$ input <Set RMD.MSPE (bit 15)> 1. Enable a stop from the $\overline{\text{STP}}$ input. (Immediate stop, deceleration stop)	[RMD] (WRITE) 15 8 n - - - - - - -
Auto output setting for the $\overline{\text{STP}}$ signal <Set RMD.MSPO (bit 16)> 1: When an axis stops because of an error, the G9103A will output the $\overline{\text{STP}}$ signal. (Output signal width: 16 cycles of the CLK signal)	[RMD] (WRITE) 23 16 - - - n - - - -
Specify the stop method to use when the $\overline{\text{STP}}$ signal turns ON. <RENV1.STPM (bit 19)> 0: Immediate stop when the $\overline{\text{STP}}$ signal is turned ON. 1: Deceleration stop when the $\overline{\text{STP}}$ signal is turned ON.	[RENV1] (WRITE) 23 16 - - - - n - - -
Read the $\overline{\text{STP}}$ signal <RSTS.SSTP (bit 12)> 0: The $\overline{\text{STP}}$ signal is OFF 1: The $\overline{\text{STP}}$ signal is ON	[RSTS] (READ) 15 8 - - - n - - - -
Read the cause of an error input < REST.ESSP (bit 6)> 1. When stopped because the $\overline{\text{STP}}$ signal turned ON.	[REST] (READ) 7 0 - n - - - - - -
Simultaneous stop command <Command: CMSTP> Outputs a one shot pulse of 0.4μs in length from the $\overline{\text{STP}}$ terminal. (The $\overline{\text{STP}}$ terminal is bi-directional. It can input the output signal again.)	[Command] 00007(h)

8-9. Emergency stop

This LSI has an $\overline{\text{EMG}}$ input terminal for use as an emergency stop signal.
While in operation, if the $\overline{\text{EMG}}$ input turns LOW or if you write an emergency stop command, the axis stops immediately. While the $\overline{\text{EMG}}$ input remains LOW, the axis cannot be operated.
The logical input of $\overline{\text{EMG}}$ terminal cannot be changed.

When the axes are stopped because the $\overline{\text{EMG}}$ input is turned ON, the LSI will generate an interrupt. By reading REST register, the cause of the error interruption can be checked.
The minimum pulse width of the $\overline{\text{EMG}}$ signal is 4 μsec when the input filter is ON. If the input filter is OFF, the minimum pulse width will be 0.1 μsec.
The status of $\overline{\text{EMG}}$ terminal can be monitored by reading RSTS register (extension status).

Read the $\overline{\text{EMG}}$ signal 0: The $\overline{\text{EMG}}$ signal is OFF 1: The $\overline{\text{EMG}}$ signal is ON	<RSTS.SEMG (bit 13)>	[RSTS] (READ) 15 7 - - n - - - - -
Read the cause of an error interrupt 1. Stopped when the $\overline{\text{EMG}}$ signal turns ON.	<REST.ESEM (bit 7)>	[REST] (READ) 7 0 n - - - - - - -
Set the $\overline{\text{EMG}}$ input filter 0: Apply a filter to the $\overline{\text{EMG}}$ input. By applying a filter, pulses less than 4 μsec in width are ignored.	<RENV1.FLTR (bit 25)>	[RENV1] (WRITE) 31 24 - - - - - - n -
Emergency stop command The operation is the same as when an $\overline{\text{EMG}}$ signal is input.	<Command: CMEMG>	[Command] 0005(h)

Note: In a normal stop operation, the final pulse width is normal. However, in an emergency stop operation, the final pulse width may not be normal. It can be a glitch. Motor drivers do not recognize the glitch pulse, and therefore only the G9103A's internal counter may count this pulse. (Deviation from the command position control). Therefore, after an emergency stop, you must perform an origin return to match the command position with the mechanical position.

8-10. Counter

8-10-1. Counter type and input method

In addition to the positioning counter, this LSI contains three other counters. These counters offer the following functions.

- Control command position and mechanical position
- Detect a stepper motor that is "out of step" using COUNTER3 (general-purpose, deviation counter) and output the synchronous signal using a comparator.

The positioning counter is loaded with an absolute value for RMV register (target position) with each start command, regardless of the operation mode selected. It decreases the value with each pulse that is output. However, if RMD.MPCS=1 and a position override 2 is executed, the counter does not count down until the PCS input turns ON.

Input to COUNTER 1 is exclusively for output pulses. However COUNTERS 2 to 3 can be selected as follows by setting RENV3 register (environment setting 3).

	COUNTER1	COUNTER2	COUNTER3
Counter name	Command position	Mechanical position	General-purpose, deviation
Counter type	Up/down counter	Up/down counter	Deviation counter
Number of bits	28	28	16
Output pulse	Available	Available	Available
Encoder (EA/EB) input		Available	Available
Pulsar (PA/PB) input		Available	Available
1/4096 division clock of 40MHz			Available

Note: When using pulsar input, use the internal signal result after multiplying or dividing.

Specify COUNTER2 (mechanical position) input <RENV3.CI20 to 21 (bit 8 to 9)> 00: EA/EB input 01: Output pulses 10: PA/PB input	[RENV3] (WRITE) 15 8 - - - - - - n n
Set COUNTER3 (deviation) input <RENV3.CI30 to 32 (bit 10 to 12)> 000:Output pulses 001: EA/EB input 010: PA/PB input 011: 1/4096 division clock of 40Mhz 100: Count the deviation between output pulses and EA/EB input 101: Count the deviation between output pulses and PA/PB input 110: Count the deviation between EA/EB input and PA/PB input	[RENV3] (WRITE) 15 8 - - - n n n - -

The EA/EB and PA/PB input terminal that are used as inputs for the counter, can be set for one of two signal input types by setting RENV2 (environment setting 2) register.

1) Signal input method: Input 90 degree phase difference signals (1x, 2x, 4x).
Counter direction: Count forward when the EA input phase is leading. Count backward when the EB input phase is leading.
2) Signal input method: Input count forward pulses or count backward pulses (Two-pulse input).
Count direction: Count forward when the EA input is rising. Count backward when the EB input is rising.

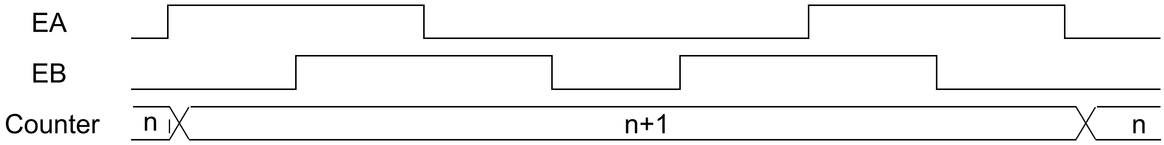
The counter direction or EA/EB and PA/PB input signals can be reversed.

The LSI can be set to sense an error when both the EA and EB input, or both the PA and PB inputs change simultaneously, and this error can be detected using REST (error interrupt cause) register.

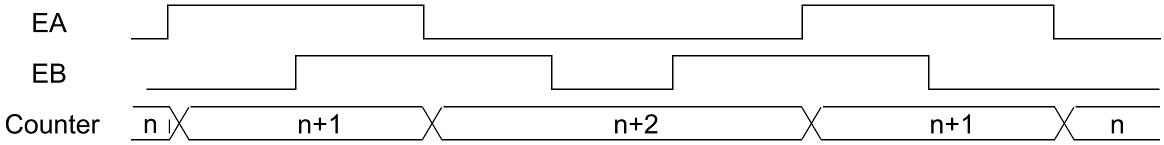
Set the input signal filter for EA/EB/EZ <RENV2.EINF (bit 8)> 1: Turn OFF the filter function 0: Turn ON the filter function (Input signals shorter than 0.3 μ sec are ignored.)	[RENV2] (WRITE) 15 8 - - - - - - - n
Setting the EA/EB input <RENV2.EIM0 to 1 (bit 9 to 10)> 00: 90 degree phase difference, 1x 10: 90 degree phase difference, 4x 01: 90 degree phase difference, 2x 11: Input up pulses and down pulses (Two-pulse input)	[RENV2] (WRITE) 15 8 - - - - - n n -
Specify the EA/EB input count direction <RENV2.EDIR (bit 11)> 0: Count forward when the EA phase is leading. Or, count forward on the rising edge of EA. 1: Count forward when the EB phase is leading. Or, count forward on the rising edge of EB.	[RENV2] (WRITE) 15 8 - - - - n - - -
Enable/disable EA/EB input <RENV2.EOFF (bit 17)> 0: Enable EA/EB input 1: Disable EA/EB input. (EZ input is valid.)	[RENV2] (WRITE) 23 16 - - - - - - n -
Set the input signal filter for PA/PB <RENV2.PINF (bit 13)> 1: Turn OFF the filter function. 0: Turn ON the filter function (Input signals shorter than 6 reference clock cycles are ignored.)	[RENV2] (WRITE) 15 8 - - n - - - - -
Specify the PA/PB input <RENV2.PIM0 to 1 (bit 14 to 15)> 00: 90 degree phase difference, 1x 10: 90 degree phase difference, 4x 01: 90 degree phase difference, 2x 11: Input up and down pulses (Two-pulse input)	[RENV2] (WRITE) 15 8 n n - - - - - -
Specify the PA/PB input count direction <RENV2.PDIR (bit 16)> 0: Count forward when the PA phase is leading. Or, count forward on the rising edge of PA. 1: Count forward when the PB phase is leading. Or, count forward on the rising edge of PB.	[RENV2] (WRITE) 23 16 - - - - - - - n
Enable/disable PA/PB input <RENV2.POFF (bit 18)> 0: Enable PA/PB input 1: Disable PA/PB input.	[RENV2] (WRITE) 23 16 - - - - - n - -
Reading EA/EB, PA/PB input error <REST.ESEE (bit 13), ESPE (bit 14)> ESEE (bit 13) = 1: An EA/EB input error occurred. ESPE (bit 14) = 1: A PA/PB input error occurred.	[REST] (READ) 15 8 - n n - - - - -

When RENV2.EDIR =0, the EA/EB input and count timing are as follows.
For details about the PA/PB input, see section "6-3. Pulsar input mode."

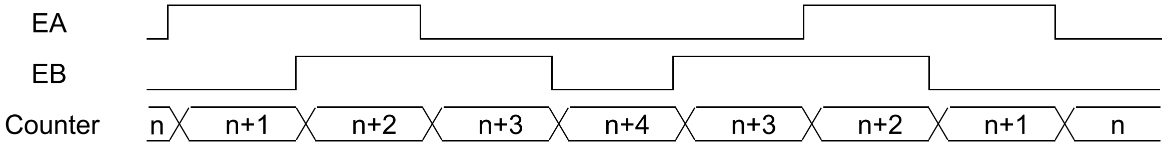
1) When using 90 degree phase difference signals and 1x input



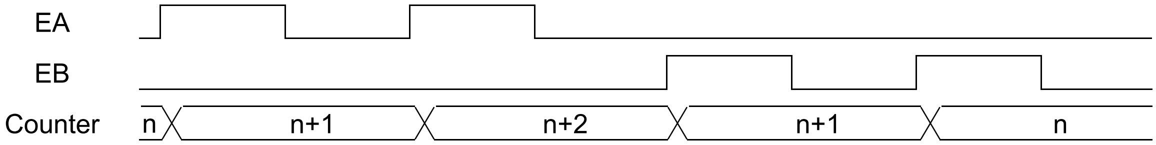
2) When using 90 degree phase difference signals and 2x input



3) When using 90 degree phase difference signals and 4x input



4) When two pulses are input (count on the rising edge)



8-10-2. Counter reset

All the counters can be reset using any of the following four methods.

1) When the CLR input signal turns ON (set in RENV3).
2) When an origin return is executed (set in RENV3).
3) When a command is written.
4) Just after the counter value is latched.

The CLR input timing can be set in RENV1 (environment setting 1). As an event interrupt cause, an interrupt can be generated when inputting the CLR signal.

Action when the CLR signal turns ON <RENV3.CU1C to 3C (bit 16 to 18) > CU1C (bit 16) =1: Reset COUNTER 1 (command position). CU2C (bit 17) =1: Reset COUNTER 2 (mechanical position). CU3C (bit 18) =1: Reset COUNTER 3 (general-purpose, deviation).	[RENV3] (WRITE) 23 16 - - - - 0 n n n
Set counter reset at the origin position <RENV3.CU1R to 3R (bit 20 to 22)> CU1R (bit 20) =1: Reset COUNTER 1 (command position). CU2R (bit 21) =1: Reset COUNTER 2 (mechanical position) CU3R (bit 22) =1: Reset COUNTER 3 (general-purpose, deviation)	[RENV3] (WRITE) 23 16 0 n n n - - - -
Set to reset the counter just after a latch. <RENV4.CU1L to 3L (bit 28 to 30)> CU1L (bit 20) =1: Reset COUNTER 1 (command position) just after a latch. CU2L (bit 21) =1: Reset COUNTER 2 (mechanical position) just after a latch. CU3L (bit 22) =1: Reset COUNTER 3 (general-purpose, deviation) just after a latch.	[RENV4] (WRITE) 31 24 0 n n n - - - -
Action for the CLR signal <RENV1.CLR0 and 1 (bit 20 to 21)> 00: Clear on the falling edge 10: Clear on a LOW level 01: Clear on the rising edge 11: Clear on a HIGH level	[RENV1] (WRITE) 23 16 - - n n - - - -
Reading the CLR signal <RSTS.SCLR (bit 17)> 0: The CLR signal is OFF 1: The CLR signal is ON	[RSTS] (READ) 23 16 - - - - - - n -
Set event interrupt cause <RIRQ.IRCL (bit 8)> 1: Generate an interrupt signal when resetting the counter value by turning the CLR signal ON.	[RIRQ] (WRITE) 15 8 - - - - - - - n
Read the event interrupt cause <RIST.ISCL (bit 8)> 1: When you want to reset the counter value by turning ON the CLR signal.	[RIST] (READ) 15 8 - - - - - - - n

Counter reset command <Command:CUN1R to CUN4R>
CUN1R(0020(h)): Reset COUNTER 1 (command position).
CUN2R(0021(h)): Reset COUNTER 2 (mechanical position).
CUN3R(0022(h)): Reset COUNTER 3 (general-purpose, deviation).
	[Command]
	0020(h)
	0021(h)
	0022(h)

Note: When the count forward (backward) timing and reset timing match, the counter will be set to 0.

8-10-3. Latch the counter and count condition

All the counters can latch their counts using any of the following five methods. The setting is made in RENV4 (environment setting 4) register. The latched values can be output from RLTC1 to 3 registers.

1) When the LTC signal turns ON..
2) When the ORG signal turns ON.
3) When the conditions for Comparator 2 are satisfied.
4) When the conditions for Comparator 3 are satisfied.
5) When a command is written.

The current speed can also be latched instead of COUNTER 3 (general-purpose, deviation). Latch of (above Items 1) to 4)) can also be stopped by hardware timing.
The LTC input timing can be set by in RENV1 (environment setting 1). An interrupt can be generated when a counter value is latched by turning ON the LTC signal or the ORG signal. This allows you to identify the cause of an event interrupt.

Specify the latch method for a counter (1 to 4) <RENV4.LTM0 to 1 (bit 24 to 25) > 00: When the LTC signal turns ON. 01: When the ORG signal turns ON. 10: When the conditions for Comparator 2 are satisfied. 11: When the conditions for Comparator 3 are satisfied	[RENV4] (WRITE) 31 24 - - - - - - n n
Specify the latch method for the current speed <RENV4.LTFD (bit 26)> 1: Latch the current speed instead of COUNTER 3 (general-purpose, deviation).	[RENV4] (WRITE) 31 24 - - - - - n - -
Specify latching using hardware <RENV4.LTOF (bit 27)> 1: Do not latch 1) to 4) above with hardware timing.	[RENV] (WRITE) 31 24 - - - - n - - -
Specify the LTC signal mode <RENV1.LTCL (bit 23)> 0: Latch on the falling edge. 1: Latch on the rising edge.	[RENV1] (WRITE) 23 16 n - - - - - - -
Set an event interrupt cause <RIRQ.IRLT (bit 9) and IROL (bit 10)> IRLT = 1: Generate an interrupt when the counter value is latched by the LTC signal being turned ON. IROL = 1: Generate an interrupt when the counter value is latched by the ORG signal being turned ON.	[RIRQ] (WRITE) 15 8 - - - - - n n -
Read the event interrupt cause <RIST.ISLT (bit 9), ISOL (bit 10)> ISLT = 1: Latch the counter value when the LTC signal turns ON. ISOL = 1: Latch the counter value when the ORG signal turns ON.	[RIST] (READ) 15 8 - - - - - n n -
Read the LTC signal <RSTS.SLTC (bit 18)> 0: The LTC signal is OFF 1: The LTC signal is ON	[RSTS] (READ) 23 16 - - - - - n - -
Counter latch command <Command: LTCH> Latch the contents of the counters (COUNTER 1 to 3).	[Command] 0029(h)

8-10-4. Stop the counter

COUNTER 1 (command position), COUNTER 2 (mechanical position), and COUNTER 3 (general-purpose, deviation) stop when RENV3 (environment setting 3) register can be set to stop.
COUNTER 1 (command position) stops while in timer mode operation.
By setting RENV3 register, you can stop counting pulses while performing a backlash correction.
COUNTER 3 (general-purpose) can be set to count only during operation ($\overline{\text{BSY}}$ = low) using RENV3 register.
By specifying 1/4096 of 40MHz signal, the time after the start can be controlled.

Specify the counting operation for COUNTERS 1 to 3 <EWNC3.CU1H to 3H (bits 28 to 30)> CU1H (bit 28) = 1: Stop COUNTER 1 (mechanical position) CU2H (bit 29) = 1: Stop COUNTER 2 (deviation) CU3H (bit 30) = 1: Stop COUNTER 3 (general-purpose, deviation)	[RENV3] (WRITE) 31 24 | - | n | n | n | - | - | - | - |
Setting the counters for backlash correction <REVN3.CU1B to 3B (bits 24 to 26)> CU1B (bit 24) = 1: Enable COUNTER 1 (command position) CU2B (bit 25) = 1: Enable COUNTER 2 (mechanical position) CU3B (bit 26) = 1: Enable COUNTER 3 (general-purpose, deviation)	[RENV4] (WRITE) 31 24 | - | - | - | - | - | n | n | n |
Specify the counting conditions for COUNTER 3 <RENV3.BSYC (bit 13)> 1. Enable COUNTER3 (general-purpose, deviation) only while operating ($\overline{\text{BSY}}$ = L).	[RENV] (WRITE) 15 8 | - | - | n | - | - | - | - | - |

8-11. Comparator

8-11-1. Comparator types and functions

This LSI has 3 circuits of 28-bit comparators. It compares the values set in RCMP1 to 3 registers with the counter values.
Comparators 1 to 3 can be used as comparison counters. There are many comparison methods and 3 processing methods that can be used when the conditions are met.
Specify the comparator conditions in RENV4 (environment 4) registers. By using these comparators, you can perform the following.

- Generate an interrupt and output the comparison result externally.
- Immediate stop and deceleration stop operations.
- Change the operation data to the same value in the pre-register (while changing the speed operating.)
- Software limit function using Comparators 1 and 2.
- Ring count function using COUNTER1 (command position) and Comparator 1.
- Ring count function using COUNTER2 (mechanical position) and Comparator 2.
- Detect out of step stepper motors using COUNTER 3 (deviation) and a comparator.
- Output a synchronous signal (IDX) using COUNTER 3 (general-purpose) and a Comparator 3.

[Comparison data] Each comparator can select the data for comparison from the items in the following table.

Comparison data	Comparator 1		Comparator 2		Comparator 3	
		C1C0 to 1		C2C0 to 1		C3C0 to 1
COUNTER 1 (command position)	O	"00"	O	"00"	O	"00"
COUNTER 2 (mechanical position)	O	"01"	O	"01"	O	"01"
COUNTER 3 (general-purpose, deviation)	O	"10"	O	"10"	O	"10"
Major application	+SL		-SL		Out-of-step detection, IDX signal output	

- O: Comparison is possible.
- +SL, -SL are used for software limits.
- If COUNTER 3 (deviation) that was specified as deviation counter is selected as comparison counter, the LSI will compare the absolute value of the counter with the comparator data. (Absolute value range: 0 to 32,767)
- Choose the comparison data by RENV4.C1C0 to 1 (bits 0 to 1), C2C0 to 1 (bits 8 to 9), and C3C0 to 1 (bits 16 to 17).

[Comparison method] Each comparator can be assigned a comparison method from the table below.

Comparison method	Comparator 1			Comparator 2			Comparator 3	
		C1S0 to 2	C1RM		C2S0 to 2	C2RM		C3S0 to 3
Comparator = Comparison counter (regardless of count direction)	O	"001"	0	O	"001"	0	O	"0001"
Comparator = Comparison counter (Count forward only)	O	"010"	0	O	"010"	0	O	"0010"
Comparator = Comparison counter (count backward only)	O	"011"	0	O	"011"	0	O	"0011"
Comparator > Comparison counter	O	"100"	0	O	"100"	0	O	"0100"
Comparator < Comparison counter	O	"101"	0	O	"101"	0	O	"0101"
Use as software limits	O	"110"	0	O	"110"	0		
IDX (synchronous signal) output (regardless of counting direction)							O	"1000"
IDX (synchronous signal) output (count forward only)							O	"1001"
IDX (synchronous signal) output (count backward only)							O	"1010"
Making Counter 1 as a ring counter	O	“000”	1					
Making Counter 2 as a ring counter				O	“000”	1		

- O: Comparison is possible. Blank: Comparison is not possible.
- C3S0 to 3 of comparator 3 is prohibited to be set to a value of “0111”. Setting any of the values may result in failing to satisfy the comparison conditions.
- When C3S0 to 3=1000 to 1010 for Comparator 3 <IDX (synchronous signal) output>, select COUNTER 3 (general-purpose, deviation) for use as the comparison counter. Other counters cannot be used for this function. Enter a positive value for the comparator setting.
- When using the comparator function as a software limit, Comparator 1 will be the positive limit value. Then, the G9103A looks for the "Comparator < Comparison counter." Comparator 2 will be the negative limit value. Then, the G9103A compares "Comparator > Comparison counter." Select COUNTER 1 (command position) as the comparison counter.
- Choose the comparison method from RENV1.C1S0 to 2 (bits 2 to 4), C2S0 to 2 (bits 10 to 12), and C3S0 to 3 (bits 18 to 21).

[Processing method when comparator conditions are satisfied]
The processing method that is used when the conditions are satisfied can be selected from the table below.

Processing method when the conditions are met	Comparator 1	Comparator 2	Comparator 3
	C1D0 to 1	C2D0 to 1	C3D0 to 1
No operation	"00"	"00"	"00"
Immediate stop operation	"01"	"01"	"01"
Deceleration stop operation	"10"	"10"	"10"
Change operation data to the value set in the pre-register	"11"	"11"	"11"

- The bit assignments to select a processing method are as follows.
RENV4.C1D0 to 1 (bits 5 to 6), C2D0 to 1 (bits 13 to 14), and C3D0 to 1 (bits 22 to 23).

Set an event interrupt cause RIRQ.IRC1 to 3 (bit 5 to 7)> IRC1 (bit 5) = 1: Generate an interrupt when the Comparator 1 conditions are satisfied. IRC2 (bit 6) = 1: Generate an interrupt when the Comparator 2 conditions are satisfied. IRC3 (bit 7) = 1: Generate an interrupt when the Comparator 3 conditions are satisfied.	[RIRQ] (WRITE) 7 0 n n n - - - - -
Read the event interrupt cause <RIST.ISC1 to 3 (bit 5 to 7)> IRC1 (bit 5) = 1: When the Comparator 1 conditions are satisfied. IRC2 (bit 6) = 1: When the Comparator 2 conditions are satisfied. IRC3 (bit 7) = 1: When the Comparator 3 conditions are satisfied.	[RIST] (READ) 7 0 n n n - - - - -
Read the comparator condition status <RSTS.SCP1 to 3 (bits 20 to 22)> SCP1 (bit 20) = 1: When the Comparator 1 conditions are satisfied. SCP2 (bit 21) = 1: When the Comparator 2 conditions are satisfied. SCP3 (bit 22) = 1: When the Comparator 3 conditions are satisfied.	[RSTS] (READ) 23 16 - n n n - - - -
Read the error interrupt cause <REST.ESC1 to 3 (bits 0 to 2)> ESC1 (bit 0) = 1: When stopped by a match of the comparator 1 conditions. (+SL) ESC2 (bit 1) = 1: When stopped by a match of the comparator 2 conditions. (-SL) ESC3 (bit 2) = 1: When stopped by a match of the comparator 3 conditions.	[REST] (READ) 7 0 - - - - - n n n

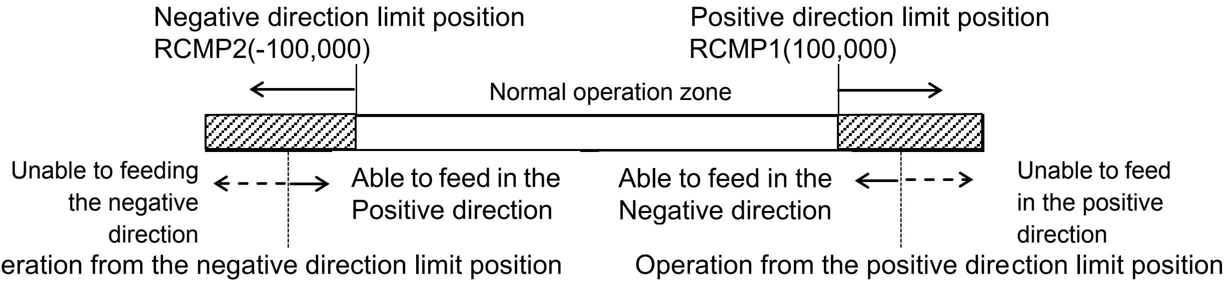
[Change speed using comparator]
If set to CND0 to 1 (n=1 to 3)=1, the value in the pre-registers for operation (PRMV, PRFL, PRFH, PRUR, PRDR, PRMG, PRDP, PRMD, PRIP, PRUS, PRDS, PRCI, PRMVY, PRIPY) is copied in the registers. If PRFH register is set in advance, the speed can be changed when the conditions are satisfied.
In order to change the speed using a comparator, the data for the next operation cannot be set because the pre-register is used.

8-11-2. Software limit function

A software limit function can be set up using comparators 1 and 2.
Select COUNTER 1 (command position) as a comparison counter for comparators 1 and 2.
Use Comparator 1 as a positive direction limit and Comparator 2 as a negative direction limit to stop the axis based on the results of the comparator and the operation direction.
When the software limit function is used, the following process can be executed.
1) Stop pulse output immediately
2) Decelerate and then stop pulse output
While using the software limit function, if a deceleration stop is selected as the process to be used when the comparator conditions are met (C1D, C2D), when an axis reaches the software limit while in a high speed start (STAUD: 0053(h)), the axis decelerates and stops. When some other process is specified for use when the conditions are met, or while in a constant speed start, the axis stops immediately.
If a software limit is ON while writing a start command, the axis does not start to move in the direction in which the software limit is enabled. However, it can start in the opposite direction.

[Setting example]

RENV4=00003838h:	Comparator 1 is used as a positive direction software limit. Comparator 2 is used as a negative direction software limit. Set to stop immediately when the software limit is reached.
RCMP1= 100,000:	Positive direction limit value
RCMP2= -100,000:	Negative direction limit value


Negative direction limit position
RCMP2(-100,000)
Positive direction limit position
RCMP1(100,000)
Normal operation zone
Unable to feeding the negative direction
Able to feed in the Positive direction
Able to feed in the Negative direction
Unable to feed in the positive direction
Operation from the negative direction limit position
Operation from the positive direction limit position

Setting the comparison method for Comparator 1 <RENV4.C1S0 to C1S2 (bits 2 to 4)> 001: RCMP1 data = Comparison data (Regardless of count direction) 010: RCMP1 data = Comparison data (While counting forward) 011: RCMP1 data = Comparison data (While counting backward) 100: RCMP1 data > Comparison counter 101: RCMP1 data < Comparison counter 110: Use as a positive direction software limit (RCMP 1< COUNTER 1) Others: Always assumes that the comparison conditions are not met.	[RENV4] (WRITE) 7 0 - - - n n n - -
Specify the process to use when the Comparator 1 conditions are met <RENV4.C1D0 to C1D1 (bits 5 to 6)> 01: Immediate stop 10: Deceleration stop	[RENV4] (WRITE) 7 0 - n n - - - - -
Specify the comparison method for Comparator 2 <RENV4.C2S0 to C2S2 (bits 10 to 12)> 001: RCMP2 data = Comparison data (Regardless of count direction) 010: RCMP2 data = Comparison data (While counting forward) 011: RCMP2 data = Comparison data (While counting backward) 100: RCMP2 data > Comparison counter 101: RCMP2 data < Comparison counter 110: Use as a negative direction software limit. (RCMP 2 > COUNTER 1) Others: Always assumes that the comparison conditions are not met.	[RENV4] (WRITE) 15 8 - - - n n n - -
Specify the process to use when the Comparator 2 conditions are met <RENV4.C2D0 to C2D1 (bits 13 to 14)> 01: Immediate stop 10: Deceleration stop	[RENV4] (WRITE) 15 8 - n n - - - - -

Note: The parts underlined mean that the settings in the example above are allowed.

8-11-3. Out of step detection function for stepper motors

If the deviation counter value controlled by the motor command pulses and the feedback pulses from the encoder on a stepper motor exceeds the maximum deviation value, the LSI will declare that the stepper motor is out of step. The LSI monitors stepper motor operation using COUNTER 3 (general-purpose, deviation counter) and a comparator.

The process which takes place after an out of step condition is detected can be selected from the above table [Processing method when comparator conditions are satisfied].

For this function, use an encoder with the same resolution as the stepper motor.

COUNTER 3 (general-purpose, deviation) can be cleared by writing a reset command to the deviation counter.

There are methods for inputting a feedback signal: Input 90 degree phase difference signals (1x, 2x, 4x) on the EA/EB terminals and Two-pulse input (input a count forward pulse and a count backward pulse).

When both the EA and EB signals change at the same time, the LSI generates an interrupt regarding as an error.

[Setting example]

RENV3 = 00001000h:	Set COUNTER 3 as an EA/EB deviation counter.
RENV4 = 00560000h:	Satisfy the conditions of Comparator 3 < COUNTER 3 (deviation) Stop immediately when the conditions are satisfied.
RCMP3 = 32:	The maximum deviation value is "32" pulses.
RIRQ = 00000080h:	Generate an interrupt when the comparator 3 conditions are met.

Set COUNTER 3 (deviation) input <RENV3.CI30 to 32 (bits 10 to 12)> 000: Output pulse 001: EA/EB input 010: PA/PB input 011: 1/4096 division clock of 40Mhz 100: Count deviation using output pulses and the EA/EB input 101: Count deviation using output pulses and the PA/PB input 110: Count deviation using the EA/EB and PA/PB inputs.	[RENV3] (WRITE) 15 8 - - - n n n - -
Specify the EA/EB input <RENV2.EIM0 to 1 (bits 9 to 10)> 00: 90 degree phase difference, 1x 01: 90 degree phase difference, 2x 10: 90 degree phase difference, 4x 11: Two-pulse mode	[RENV2] (WRITE) 15 8 - - - - - n n -
Specify the EA/EB input count direction <RENV2.EDIR (bit 11)> 0: When the EA phase is leading, or count forward on the EA rising edge. 1: When the EB phase is leading, or count forward on the EB rising edge	[RENV2] (WRITE) 15 8 - - - - n - - -
Read the EA/EB input error <REST.ESEE (bit 13)> 1: An EA/EB input error has occurred.	[REST] (READ) 15 8 - - n - - - - -
Set the EA/EB/EZ input filter <RENV2.EINF (bit 8)> 0: Turn OFF the filter function 1: Turn ON the filter function (Ignore input signals shorter than 6 CLK cycles)	[RENV2] (WRITE) 15 8 - - - - - - - n
Enable/disable EA/EB input <RENV2.EOFF (bit 17)> 0: Enable EA/EB input 1: Disable EA/EB input (EZ input is left enabled.)	[RENV2] (WRITE) 23 16 - - - - - - n -
Counter reset command <Command: CUN3R> Clear COUNTER 3 (general-purpose, deviation) to zero.	[Command] 0022(h)

Note: The parts underlined mean that the settings in the example above are allowed.

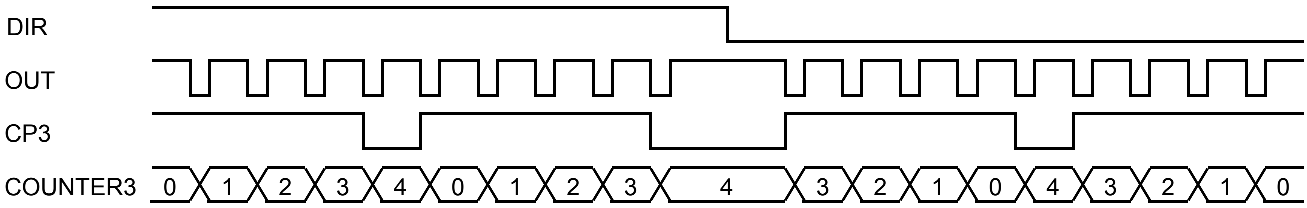
8-11-4. IDX (synchronous) signal output function

Using Comparator 3 and COUNTER3 (general-purpose, deviation counter) that was specified to "general-purpose counter," the LSI can output signals to CP3 terminal at specified intervals.
If setting to that C3C0 toC2C1="10" (COUNTER 3) and C3S0 toC3C3="1000 to "1010" (IDX output), the G9103A can be used for IDX (index) operation.
The counter range of COUNTER 3 will be 0 to the value set in RCMP3 [Max. 32,767]. If counting down from 0, the next counter value will be the value set in RCMP3, and if counting up from the value set in RCMP3, the next counter value will be 0.
The input for COUNTER 3 can be set the value CI30-CI32 in the RENV3.

Set COUNTER 3 (deviation) <RENV3.CI30 to 32 (bits 10 to 12)> 000: Output pulse 001: EA/EB input 010: PA/PB input 011: 1/4096 division clock of 40 MHz 100: Count deviation using output pulses and the EA/EB input 101: Count deviation using output pulses and the PA/PB input 110: Count deviation using the EA/EB and PA/PB inputs.	[RENV3] (WRITE) 15 8 - - - n n n - -
Select Comparator 3 comparison counter < RENV4.C3C0 to 1 (bits 16 to 17)> 00: COUNTER 1 (command position) 01: COUNTER 2 (mechanical position) 10: COUNTER 3 (general-purpose, deviation) 11: Always assumes that the comparison conditions are not met.	[RENV4] (WRITE) 23 16 - - - - - - n n
Set the comparison method for comparator 3 <RENV4.C3S0 to 3 (bits 18 to 21)> 0001: RCMP3 data = Comparison data (regardless of count direction) 0010: RCMP3 data = Comparison data (while counting up) 0011: RCMP3 data = Comparison data (while counting down) 0100: RCMP3 data > Comparison counter 0101: RCMP3 data < Comparison counter 0111: Prohibited 1000: Use as an IDX (synchronous) signal output (regardless of count direction) 1001: Use as an IDX (synchronous) signal output (while counting forward) 1010: Use as an IDX (synchronous) signal output (while counting backward) Others: Always assumes that the comparison conditions are not met.	[RENV4] (WRITE) 23 16 - - n n n n - -

Note: The parts underlined mean that the settings in the example above are allowed.

Output example:
Regardless of the feed direction, the G9103A will output the IDX signal using negative logic for the output pulses. (Counting range: 0 to 4.)
Settings: RENV3 = 00000000h, RENV4 = 00220000h, RCMP3 = 4



8-11-5. Ring counter function

Ring counter is an up / down counter to repeat from 0 to (setting value). When the count value is (the setting value) and counts forward, it will be 0. When the count value is 0 and counts backward, it will be the setting value. Mainly it is used to control the current angle of the rotation table.
Using comparator 1, COUNTER 1 (command position counter) can be made a ring counter, and using comparator 2, Counter 2 (mechanical position counter) can be made a ring counter.
When making COUNTER 1 a ring counter, set to that RENV4.C1C0 to 1 =00, RENV4.C1S20 to 2=000, RENV4.C1RM=1 and set the setting value in RCMP1 register. (positive only)
When making COUNTER 2 a ring counter, set to that RENV4.C2C0 to 1 =01, RENV4.C2S20 to 2=000, RENV4.C2RM=1 and set the setting value in RCMP2 register. (positive only)

The counter value for a ring counter should be within the range from 0 to (the setting value). If the counter value is out of the range, write a data to RCUN1 or RCUN2 and make the counter value within the range, and make count.

Even if the setting value for PRMV in the positioning operation is out of the above range, it operates.
For example, when the rotation table of the 3600 pulses per rotation is operated at RCMP1=3599 and PRMV=7200, the table rotates two times and the counter 1 after stopping will be the same value before starting.

Set operation for COUNTER 1 <RENV4.C1RM (bits 7)> 0: Normal count operation 1: Ring count operation using comparator 1	[RENV4] (WRITE) 7 0 n - - - - - - -
Set comparison counter for Comparator 1 <RENV4.C1C0 to 1 (bits 0 to 1)> When making counter 1 operate as ring counter, fix to 00.	[RENV4] (WRITE) 7 0 - - - - - - 0 0
Set the comparison method for comparator 1 <RENV4.C1S0 to 2 (bits 2 to 4)> When making counter 1 operate as ring counter, fix to 000.	[RENV4] (WRITE) 7 0 - - - 0 0 0 - -
Set operation when comparator 1 conditions are met <RENV4.C1D0 to 1 (bits 5 to 6)> When making counter 1 operate as ring counter, fix to 00.	[RENV4] (WRITE) 7 0 - 0 0 - - - - -
Set operation for COUNTER 2 <RENV4.C2RM (bits 15)> 0: Normal count operation 1: Ring count operation using comparator 2	[RENV4] (WRITE) 15 8 n - - - - - - -
Set comparison counter for Comparator 2 <RENV4.C2C0 to 1 (bits 8 to 9)> When making counter 1 operate as ring counter, fix to 00.	[RENV4] (WRITE) 15 8 - - - - - - 0 0
Set the comparison method for comparator 2 <RENV4.C2S0 to 2 (bits 10 to 12)> When making counter 1 operate as ring counter, fix to 000.	[RENV4] (WRITE) 15 8 - - - 0 0 0 - -
Set operation when comparator 2 conditions are met <RENV4.C2D0 to 1 (bits 13 to 14)> When making counter 2 operate as ring counter, fix to 00.	[RENV4] (WRITE) 15 8 - 0 0 - - - - -

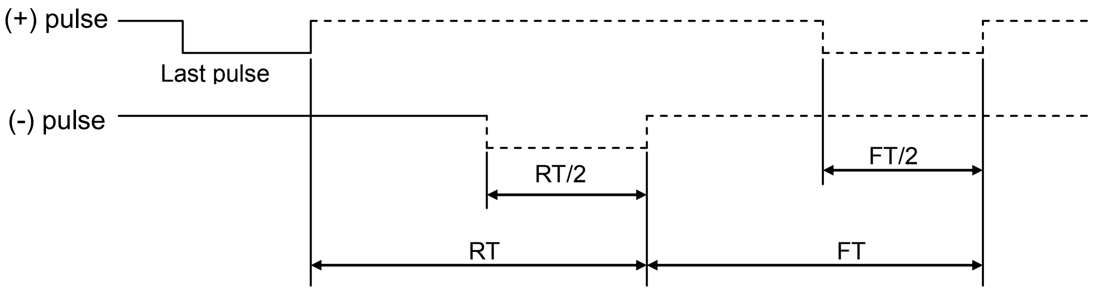
8-12. Backlash correction

This LSI has a backlash correction functions. This function outputs the number of command pulses that are specified for the correction value just before the command operation, in the speed set in the RFA (correction speed) register.
The backlash correction is performed each time the direction of operation changes. The correction amount and method is specified in RENV5 (environment setting 5) register.
The operation of the counter (COUNTER 1 to 3) can be operated and monitored using RENV3 (environment setting 3) register even while correction.

Enter the correction value <RENV5.BR0 to 11 (bits 0 to 11)> Backlash correction amount value (0 to 4,095)	[RENV5] (WRITE) 15 8 - - - - n n n n 7 0 n n n n n n n n
Set the correction method <RENV5.ADJ (bits 12)> 0: The correction function turns OFF 1: Backlash correction	[RENV5] (WRITE) 15 8 - - - n - - - -
Action for backlash correction <RENV3.CU1B to 3B (bit 24 to 26)> CU1B (bit 24) = 1: Enable COUNTER 1 (command position) CU2B (bit 25) = 1: Enable COUNTER 2 (mechanical position) CU4B (bit 26) = 1: Enable COUNTER 3 (general-purpose, deviation)	[RENV3] (WRITE) 31 24 - - - - - n n n

8-13. Vibration restriction function

This LSI has a function to restrict vibration when stopping by adding one pulse of reverse operation and one pulse of forward operation shortly after completing a command operation.
Specify the output timing for additional pulses in RENV6 (environment setting 6) register.
When both the reverse timing (RT) and the forward timing (FT) are non-zero, the vibration restriction function is enabled.
The dotted lines below are pulses added by the vibration restriction function. (An example in the positive direction)



Specify the reverse operation timing <RENV6.RT0 to 15 (bits 0 to 15)> RT range: 0 to 65,535 The unit is 1.6 μs Settable range: 0 to approx. 0.1 sec.	[RENV6] (WRITE) 15 8 n n n n n n n n 7 0 n n n n n n n n
Specify the forward operation timing <RENV6.FT0 to 15 (bits 16 to 31)> FT range: 0 to 65,535 The unit is 1.6 μs Settable range: 0 to approx. 0.1 sec.	[RENV6] (WRITE) 31 24 n n n n n n n n 23 16 n n n n n n n n

Note: The optimum values for RT and FT varies according to each piece of machinery and load. Therefore, it is best to obtain these values by experiment.

8-14. Excitation sequence for stepper motors

This LSI can generate 2-2 phase and 1-2 phase excitation sequences for 2-phase stepper motors to provide unipolar and bipolar driving.

The LSI uses $\overline{\text{BSY}}$ / PH1, $\overline{\text{FUP}}$ / PH2, $\overline{\text{FDW}}$, and $\overline{\text{MVC}}$ / PH4 terminals to output these signal sequences. To switch signals outputs, set RMD.MPH register. (Rotation can be reversed by RENV1.MREV=1).
When excitation sequence is specified to PH1, PH2, PH3, and PH4 output terminals, the output can be masked using RMD.MMPH register.
Output level of PH1 to 4 when masked is selected between LLLL/LLHH using RMD.MMPH setting.
To switch between unipolar and bipolar signals, set RMD.MUB register. To switch between 2-2 and 1-2 phase excitation, set RMD.MFH register.
While the LSI is producing an excitation signal for a single phase in 1-2 phase excitation (steps 1, 3, 5, and 7 in the table below), if you switch to 2-2 phase excitation, the LSI will change to 2 phase excitation status starting with the next output pulse.
By reading RSTS (extension status) register, you can monitor the excitation sequence status.

[Unipolar excitation sequence]

2-2 phase excitation					
STEP	0	1	2	3	0
PH1	H	H	L	L	H
PH2	L	H	H	L	L
PH3	L	L	H	H	L
PH4	H	L	L	H	H
(-) Operation direction (+)					

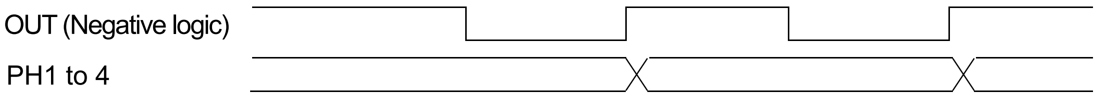
1-2 phase excitation									
STEP	0	1	2	3	4	5	6	7	0
PH1	H	H	H	L	L	L	L	L	H
PH2	L	L	H	H	H	L	L	L	L
PH3	L	L	L	L	H	H	H	L	L
PH4	H	L	L	L	L	L	H	H	H
(-) Operation direction (+)									

[Bipolar excitation sequence]

2-2 phase excitation					
STEP	0	1	2	3	0
PH1	H	H	L	L	H
PH2	L	H	H	L	L
PH3	L	L	L	L	L
PH4	L	L	L	L	L
(-) Operation direction (+)					

1-2 phase excitation									
STEP	0	1	2	3	4	5	6	7	0
PH1	H	H	H	H	L	L	L	L	H
PH2	L	L	H	H	H	H	L	L	L
PH3	L	L	L	H	L	L	L	H	L
PH4	L	H	L	L	L	H	L	L	L
(-) Operation direction (+)									

[Change the timing of the excitation sequence]

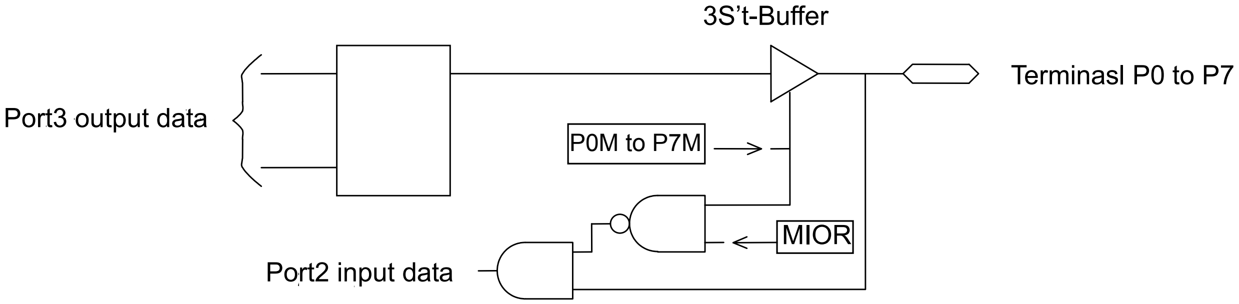


Selection of the $\overline{BSY}$ / PH1, $\overline{FUP}$ / PH2, $\overline{FDW}$ / PH3, $\overline{MVC}$ / PH4 terminal output <RMD.MPH (bit 20)> 0: Output $\overline{BSY}$, $\overline{FUP}$, $\overline{FDW}$, and $\overline{MVC}$ signals. 1: Output PH1, PH2, PH3, and PH4 signals.	[RMD]	(WRITE)	23	16		-	-	n	-	-	-	-
Mask PH1, PH2, PH3, and PH4 signals <RMD.MMPH (bit 21)> 0: Output the level specified in RMD.MBIM from PH1, PH2, PH3, and PH4 1: Output PH1, PH2, PH3, and PH4 signals.	[RMD]	(WRITE)	23	16	-	-	n	-	-	-	-	-
Change the status of mask PH1, PH2, PH3, and PH4 signals <RMD.MBIM (bit 25)> 0:PH1=L, PH2=L, PH3=L, PH4=L 1:PH1=L, PH2=L, PH3=H PH4=H	[RMD]	(WRITE)	31	24	-	-	-	-	-	-	n	-
Select the excitation method using PH1, PH2, PH3, and PH4 signals <RMD.MUB (bit 22) > 0: Output excitation sequence for 2-phase unipolar 1: Output excitation sequence for 2-phase bipolar	[RMD]	(WRITE)	23	16	-	n	-	-	-	-	-	-
Select the excitation method using PH1, PH2, PH3, and PH4 signals <RMD.MFH (bit 23)> 0: Output excitation sequence for full step 1: Output excitation sequence for half step	[RMD]	(WRITE)	23	16	n	-	-	-	-	-	-	-
Read the excitation sequence signal <RSTS.SPH1 to 4 (bits 24 to 27)> SPH1 = 1: PH1 is ON (HIGH) status SPH2 = 1: PH2 is ON (HIGH) status SPH3 = 1: PH3 is ON (HIGH) status SPH4 = 1: PH4 is ON (HIGH) status	[RSTS]	(READ)	31	24	-	-	-	-	n	n	n	n

8-15. General-purpose I/O terminals (P0 to P7)

Although these ports are set as input ports by default, by setting RENV2.P0M to P7M (bits 0 to 7), they can be set individually for input or output every bit.

The internal arrangement of these terminals is roughly as shown below. Although they are used primarily as input terminals, they can be set to act as a latched output circuit. If they are changed to function as output terminals, the LSI will output a latched status.



(The initial status of latched output in the above figure is “LOW” level.)

When they are set for use as output ports, the status is changed by writing data to Port 3 of the I/O port, their corresponding bits are set to "1" will go HIGH by the data written into Port 3.

The terminal status can be checked by reading Port 2.

The status data can be masked when the output port is selected using RMD.MIOR (bit 24).

To enable the input change interrupt on the center LSI using Port 2, set MI0R = 0. Then, even if the output port status is changed, the corresponding port 2 bit will also change so that an interrupt occurs. When MI0R is 1, if the output port status changes, the corresponding port 2 bit will be left at "0."

Setting the general I/O terminals <RENV2.P0M to P7M (bits 0 to 7)> 0: Make the terminal that corresponds to the bit an input terminal. 1: Make the terminal that corresponds to the bit an output terminal.	[RENV2] (WRITE) 7 0 n n n n n n n n
Select the monitoring method for the output setting bits in a general-purpose I/O terminal. <RMD.MIOR (bits 24)> 0: Read the setting of the output bits on Port 2. 1: Regardless of the setting of the output bits, the bits corresponding to Port 2 will be "0."	[RMD] (WRITE) 31 24 - - - - - - - n
Set the general-purpose I/O terminal data to be output <Set IOPOB (bits 0 to 7) on I/O Port 3> 0: LOW level (when specified as output port) 1: HIGH level (when specified as output port)	[Port3] (WRITE) 7 0 n n n n n n n n
Read the general-purpose I/O terminal input data <Set IOPIB (bits 0 to 7) on I/O Port 2> 0: LOW level 1: HIGH level	[Port2] (WRITE) 7 0 n n n n n n n n

8-16. Interrupt output

This LSI can output an interrupt request signal ($\overline{\text{INT}}$) from the center LSI.: There are 17 types of errors, 17 types of events, and interrupt requests by the change from operating to stop, to the center LSI. Each error cause will always generates an interrupt request without conditions, and each event cause can be set in RIRQ register.
If any of the interrupts by errors occur, SERR of the Main Status (MSTS) changes 0 to 1.
If any of the interrupts by events occur, MSTS.SEVT changes to 1.
If a stop interrupt occurs, MSTS.SEND changes to 1.
Normally, when any of SERR, SEVT, SEND is 1, MSTS.SINT changes to 1.

[Cautions]
Motionnet LSI series have CPU emulation local LSI, G9004A. Data communication length of the center LSI (G9001A) and G9004 is up to 256 bytes length.
To control the synchronization function of clock for motor control and simultaneous stop function, the G9103A captures communications to all local LSIs.
However, its reception buffer has only 128 bytes. Therefore, if the G9103A captures more than 128 bytes of data communication, the reception buffer overflows and G9103A sends the error information to G9001A at the cyclic communication or data communication for itself (G9103A) just after the overflows.
For example, the center LSI performs data communication of more than 128 bytes to G9004A, this communication succeeds. However, after that G9103A sends the error information to G9001A in communication between G9001Aand G9103A, and a communication error (STSW.ERAE=1, ISTW.ERA=0011) occurs on the G9001A.
In the case that there are both G9103A and G9004A in the Motionnet system, please make length of data communication with G9004A less than 129 bytes.

Please note that software change may be needed to short data communication length to G9004A in the case that you change G9003 to G9103A.

[Interrupt sequence]
G9103A's main status is sent (revised) to the address that corresponds to G9103A that is in port data area of the center LSI, as input data for port0 and 1 in G9103A in cyclic communication

On the center LSI, by an input change interrupt setting, set so as to generate an interrupt at the time to change port 0. Because of that, when bit 0 (SINT) of the port 0 changes 0 to 1, G9103A can issue an interrupt to CPU.

If you set a port 0 change interrupt on a local LSI that can perform data communication, the case that bit 0 of port 0 changes 0 to 1 is only a cause to occur interrupt. The change of bit 1 to 7 is ignored.

[Error interrupt]
You can confirm about error interrupt cause.
If any bit of REST register is 1, MSTS.SERR changes to 1.

To clear REST register, there are two the following ways .
1. When RENV4.ISMR=0, all bits are cleared by reading REST register.
2. Regardless of RENV4.ISMR, you write a data that the you make bit you want to clear 1 to REST register, the specified bit is cleared.

When RENV4.ISMR=0, if communication error occur at the communication for reading the rest register, data with all bit 0 is transferred because the register is read again to retry. Therefore, we recommend that you set RENV4.ISMR to 1 and use the above 2 to clear.

[Event interrupt]
Event interrupt is effective by the causes specified in RIRQ register. You can confirm the event interrupt cause, and the correcponsing bit goes 1.
When any bit of RIST register is 1, MSTS.SEVT changes to 1.

To clear the RIST register, there are two the following ways .
1. When RENV4.ISMR=0, all bits are cleared by reading RST register.

2. Regardless of RENV4.ISMR, you write a data that the you make bit you want to clear 1 to RIST register, the specifiedd bit is cleared.

When RENV4.ISMR=0, if communication error occur at the communication for reading the rest register, data with all bit 0 is transferred because the register is read again to retry. Therefore, we recommend that you set RENV4.ISMR to 1 and use the above 2 to clear.

[Stop interrupt]
A stop interrupt (SEND) is a simple interrupt function that produces an interrupt without discriminating between normal stop and error stop.
For a normal stop interrupt to be issued, the confirmation process to read the RIST register is necessary as described in the Cause of an Event section. If your system needs to provide a stop interrupt only when a stop occurs, it is easy to use the stop interrupt function.

The stop interrupt is cleared by writing "Command to reset SEND interrupt (0008h).

By setting RENV1 (Environment Setting Register 1).SEDM, you can choose not to reflect the occurrence of SEND interrupts in the Main Status bit 0 (SINT) or to reset the SEND interrupt when starting an operation by setting RENV1.SEDR.
When the data for the next operation is set by RMD.MENI=1 as a continuous operation using pre-register, you can choose not to issue SEND interrupt even if the current operation completes.

Note. When operations synchronized with PA/PB (RMD.MOD=51h to 55h, 68h to 6Dh, 78h to 7Dh) is complete, SEND does not become 1.

[Interrupt control on the G9001A side]
As you can see, only one G9103A have 35 (max) types of interrupt causes. the more axes there are, the more causes the whole system has.

Additionally, In interrupt cause of G9001A, there are other interrupt causes as follows, in addition to Input change interrupt that is shown in the term [input sequence].

1. When G9001A is available for writing to the DATA transmission FIFO. (CEND)
2. When G9001A receive a break frame. (BRKF)
3. When I/O communication errors occur. (EIOE)
4. When an error occur in data communication or system communication. (EDTE)
5. When an local side has an error to receive data. (ERAE)
6. When an error occur in accessing to CPU. (CAER)

While the G9001A handles interrupts, other interrupt may occur. Therefore, to create a program to handle interrupts, you should pay attention to such case.
If you do not need high response, please consider the way that does not use interrupt processing.

[Cause of an interrupt evaluation process]
The causes of an interrupt from G9103A can be evaluated as follows below.
1) Read the main status and check whether bits 1, 2, or 3 is "1."
2) If bit 1 (SEND) = "1”, a Stop interrupt occurs. Reset the LSI using a reset command (0008(h)).
3) If bit 2 (SERR) = "1”, read REST register to identify the cause of the interrupt.
4) If bit 3 (SEVT) = "1", read RIST register to identify the cause of the interrupt.
With these procedures, you can identify an interrupt cause and turn OFF the occurrence of the interrupt.

Note 1:G9001A outputs an Interrupt request signal for CPU through $\overline{\text{INT}}$ terminal at the timing when G9001A performs I/O communication to the local LSI that requested the interrupt.
Therefore, even while CPU is writing normal sending data to FIFO in G9001A or it is reading from reception FIFO, a interrupt request may occur.

Using the interrupt routine, if the center LSI tries to make communication with the local LSI that requested the interrupt, data of the transmission FIFO and reception FIFO change and an error occur in the process after it returns to perform a main routine.

Therefore, in the interrupt routine, please make only that interrupt request occur stored in flags and check the condition of the flag in the main routine and perform data communication with the local LSI that requested interrupt.

Note 2:While processing in the step 4) above, it is possible that SEND or SERR may change to 1 again. After the steps 1) to 4) complete, please check whether all SEND, SERR, SEVT is 0.

Note 3.When several G9103As are used, interrupt process is performed axis by axis. However, an interrupt cause may occur on the axis that already completed interrupt process. After the CPU returns to ready to receive an interrupt, please read out the Main status and confirm that all G9103As do not receive interrupts and then terminate the interrupt routine.

Note 4:Information of port 0 to 3 including the Main Status is revised in I/O communication under normal conditions. However, it is also revised at the time to send response frame of data communication to G9103A.

The Main Status bit 0 (SINT) can be masked by setting RMD (operation mode) register.
When masked (RMD.MINT = 1), even though the status changes, bit 0 (SINT) in the Main Status will remain "0" and will not change to "1" when the interrupt conditions are satisfied.
When an interrupt occurs, if the output mask is turned OFF (RMD.MINT = 0), bit 0 (SINT) in the Main Status will change to "1."

Read the interrupt status <MSTB0.SINT, SEND, SERR, SEVT> SINT (bit 0) = 1: turns 1 when SEND = 1 or SERR = 1 or SEVT = 1. SEND (bit 1) = 1: turns 1 when stopped and becomes 0 by an interrupt reset command (0008(h)). SERR (bit 2) = 1: Turns 1 when an error interrupt occurs. Turns 0 by reading REST. SEVT (bit 3) = 1: Becomes 1 when an event interrupt occurs. Becomes 0 by reading RIST.	[MSTSB0] (WRITE) 7 0 - - - - n n n n
Set the interrupt mask <RMD.MINT (bit 19)> 1: Mask SINT (bit 0) in the Main Status.	[RMD] (WRITE) 23 16 - - - - n - - -
Set the mask on the stop interrupt <RENV1.SEDM (bit 27)> 0: Enable stop interrupts (reflected to SINT) 1: Mask the stop interrupt (only the SEND status change)	[RENV1] (WRITE) 31 24 - - - - n - - -
Set to reset when a stop interrupt occurs. <RENV1.SEDR (bit 28)> 1: Reset the stop interrupt bit (SEND) when starts.	[RENV1] (WRITE) 31 24 - - - n - - - -
Set stop interrupt while continuous operation <RMD.MENI (bit 7)> 1: Mask the stop interrupt when pre-register for operation is fixed at normal stop. (Only available when RENV1.SEDM=0)	[RMD] (WRITE) 7 0 n - - - - - - -
Reset the stop interrupt <Command: INTRS> Reset the SEND bit (stop interrupt).	[Command] 0008(h)
Read the cause of the error interrupt <Command: RREST> Read the data of REST register (error interrupt cause).	[Command] 00F2(h)
Read the event interrupt cause <Command: RRIST> Read the data of RIST register (event interrupt cause).	[Command] 00F3(h)
Set the event interrupt cause <Command: WRIRQ> Write the FIFO data into RIRQ register (event interrupt cause).	[Command] 00AC(h)
Reset the error interrupt cause <Command: WREST> Only the bit 1 is reset by writing into REST register (error interrupt cause).	[Command] 00AD(h)
Reset the event interrupt cause <Command: WRIST> Only the bit 1 is reset by writing into RIST register (event interrupt cause).	[Command] 00AE(h)

[Error interrupt causes] <Detail of REST: The cause of an interrupt makes the corresponding bit "1">

Error interrupt cause	Cause (REST)	
	Bit	Bit name
Stopped by Comparator 1 conditions being satisfied (+SL)	0	ESC1
Stopped by Comparator 2 conditions being satisfied (-SL)	1	ESC2
Stopped by Comparator 3 conditions being satisfied	2	ESC3
Stopped by turning ON the +EL input	3	ESPL
Stopped by turning ON the -EL input	4	ESML
Stopped by turning ON the ALM input	5	ESAL
Stopped by turning ON the STP input	6	ESSP
Stopped by turning ON the EMG input	7	ESEM
Decelerated and stopped by turning ON the SD input	8	ESSD
Stopped by an overflow occurrence of PA/PB input buffer counter	9	ESPO
When stopped by a communication error.	10	ESNT
(Always 0)	11	Not defined
When a position override cannot be executed.	12	ESOR
An EA/EB input error occurred (does not stop).	13	ESEE
A PA/PB input error occurred (does not stop).	14	ESPE
Stopped by data error for interpolation operation Note1	15	ESDT
Simultaneously stopped by error stop of other axes (simultaneous stop by RSYN setting)	16	ESIP
Simultaneously stopped by beyond the Operation range (28 Bit) while circular interpolation.	17	ESAO
Synchronization of the clock for motor control is collapsed.	18	ERFJ

Note1. When start command is written with the following data setting, error interrupt occurs.
When linear interpolation: RMV=0 and RMVY=0
When circular interpolation (RIP=0 and RIPY=0) or (RMV=RIP and RMVY=RIPY)

[Event interrupt causes] < The corresponding interrupt bit is set to 1 and then an interrupt occurred>

Event interrupt cause	Set cause (RIRQ)		Cause (RIST)	
	Bit	Bit name	Bit	Bit name
Automatic stop	0	IREN	0	ISEN
When acceleration starts	1	IRUS	1	ISUS
When acceleration ends	2	IRUE	2	ISUE
When deceleration starts	3	IRDS	3	ISDS
When deceleration ends	4	IRDE	4	ISDE
When the Comparator 1 conditions are satisfied	5	IRC1	5	ISC1
When the Comparator 2 conditions are satisfied	6	IRC2	6	ISC2
When the Comparator 3 conditions are satisfied	7	IRC3	7	ISC3
When the counter value is reset by a CLR signal input	8	IRCL	8	ISCL
When the counter value is latched by an LTC input	9	IRLT	9	ISLT
When the counter value is latched by an ORG input	10	IROL	10	ISOL
When the SD input turns ON	11	IRSD	11	ISSD
When the STA input turns ON	12	IRSA	12	ISSA
When operation starts in receiving start command broadcast communication.	13	IRNA	13	ISNA
When stopping by receiving broadcast communication stop command	14	IRNP	14	ISNP
When it is possible to write into the pre-register for operation	15	IRNM	15	ISNM
When setting data for the next operation does not in time	16	IRBE	16	ISBE

8-17. Synchronization function with other axes

There are two functions as the synchronous function with other axes as follows.

1. Synchronization function of the clock for motor control
2. Simultaneous stop when stop by errors.

8-17-1. Synchronization function of the clock for motor control

The output pulse train of G9103A is generated to be synchronized with reference clock (40MHz or 80 MHz). Because G9103A is a LSI to control for single axis, use more than one G9103A when interpolation is executed. Normally each crystal oscillator provides reference clock to each G9103A independently.

In order to operate synchronization such as interpolation, note that the frequency error of this reference clock.

For example, if accuracy for two reference clock rate (frequency) is +100ppm and -100 ppm, accuracy of output pulse frequency from G9103A is also +100ppm and –100 ppm.
Therefore, the number of the pulse output for 5 seconds at 10kpps constant speed is 50005 pulses and 49995 pulses and the error of interpolation trajectory become big.
In order to reduce this error, G9103A has the synchronization function of the clock for motor control.
When using several G9103As are connected, use any one of G9103A as a master device for synchronization of the clock for motor control and other G9103As can be synchronized with the clock for motor oontrol of the master LSI.

When the G9103A which device address is 4 is used as a master.

	G9103A-1	G9103A-2	G9103A-3	G9103A-4	G9103A-5
Device address	1(01(h))	4(04(h))	6(06(h))	10(0A(h))	20(14(h))
RSYN.SYNC	1	1	1	1	1
RSYN.DNMST	04(h)	04(h)	04(h)	04(h)	04(h)
Setting order	(2)	(1)	(3)	(4)	(5)

Setting process

1. Set RSYN register of a master LSI
   RSYN.DNMST ← the device address of a master LSI (=04h)
   RSYN.SYNC ← 1
2. Set RSYN registers of other G9103As to the same as a master LSI.
3. It takes approximately 4 m sec until synchronous control starts. If synchronous control starts, RSYN.SYON becomes “1”.

Note 1.Be sure to set RSYN of master LSI first. For other G9103As, any order is available.
Note 2. Synchronization control synchronize only control the clock for motor control. Control clock for communication is not synchronized and is influenced by the frequency error of CLK input. Therefore, use crystal oscillator that the frequency permissible deviation is within ±500ppm.
Note 3: Operation timing with output pulse width may fluctuate ±25ns by synchronization control.

[Monitor of status to synchronize of the clock for motor control]

The reference clock frequency that is input to G9103A is 40 MHz or 80 MHz. The motor control circuit is operate at 40MHz, and the axis control circuit operates at 20MHz.
The synchronization function of the clock for motor control corrects 20 MHz clock that is used for the axis control circuit synchronized with 20MHz of G9103A (Clock master LSI) specified by RSYN.DNMST.

You can monitor the correction amount by RSYN.FAM0 to 7. Writing to RSYN.FAM0 to 7 is invalid.

FAM value is shown by 8 bit integer with sign (-128 to +127). With no correction adjustment, FAM value is 0 .
The formula of CLK input clock rate (f_{CLK}) and the clock frequency used for motor control circuit (OSC_CLK) relation is as follows.

1. When CKSL=L (40MHz input) ・・OSC_CLK＝f_{CLK}×(32768＋FAM value)／65536
2. When CKSL=H (80MHz input)) ・・OSC_CLK＝f_{CLK}×(32768＋FAM value)／131072

When clock synchronization function is operating, the difference with the center value of correction varies in -1 to +1 range.
You can monitor the difference with the crystal oscillation frequency of G9103A (clock master). Under normal condition, you do not need to monitor the clock.

Tha formula of FAM value (-128 to +127) relation with correction amount (ppm) is as follows.
The correction amount [ppm] = (FAM value ×1,000,000) / 32,768 ･･･(formula 8-17-1)

Tha range of FAM value is -128 to +127. Therefore, the range to be monitored is from -3906ppm to +3875 ppm.

[Detection of synchronization error of the clock for motor control]

In clock synchronization, the master 9103A sends clock frequency information for motor control with the main status (MSTS.SYN0 to SYN6) to other G9103As and other G9103A monitor the information and operate synchronization using the information.

However, if the synchronization function of the master G9103A turns OFF (RSYN.SYNC=0) by mistake in synchronization, the master G9103A outputs the setting information of GRP terminals (SGP0 to 2) in stead of the synchronization information. Other G9103A recognizes the setting information of GRP terminals as clock frequency information by mistake. Therefore, because G9103As synchronizes based on the information, the clock frequency for motor control deviates considerably.

If you use the system under this condition, the error of motor speed of operation became big. In interpolation operation, the interpolation trajectory became misshapen considerately. Threfore, G9103A has a feature to detect this obnormal condition.

There are two following circuits for detection : a circuit to control rate correction amount and a circuit to control a band of fluctuation of the rate.

1. Circuit to control rate correction amount (mainly to detect the difference of crystal oscillator)

   When the absolute value of FAM value exceeds the setting value, it recognized the condition as an error.
   Set RSYN.FAL0 to 6. (1 to 127) The smaller the setting value is, the more sensitive the detection sensitivity. Under normal condition, set a value that is approximately four-times of frequency allowable deviation of a crystall oscillator. For example, when the allowable deviation is plus mainus 100ppm and, the correction amount is 400 using the above formula 8-17-1, calculate the FAM value. The result become 13.1. Therefore, please set 13 or 14 to the setting value. In the case that the setting value is 0 (default), G9103A does not control rate correction amount.

2. Circuit to control a band of fluctuation of the rate (mainly to detect RSYN.SYNC=0 of the master G9103A)

   The FAM value in synchronization of the clock for motor control fluctuates within plus or minus 1 of the center correction value. When the band of the fluctuation exceeds plus or minus 7, G9103A recognizes this condition as an error.
   This circuit detects a rapid change of clock frequency information of the master G9103A.
   This function operates only when setting RSYN.FAWL=1.
   When one circuit of the above 1 or 2 detects an error, G9103A occurs an error interrupt request (REST.EFAJ=1) after turning the clock synchronization function OFF (RSYN.SYNC=0) automatically and switching the rate to a default.

Set a clock master LSI for motor control. <RSYN.DNMST(bit 0 to 5)> Set the device address of G9103A that used as the clock master. (0 to 63)	[RSYN] (WRITE) 7 0 - - n n n n n n
Set the clock synchronization for motor control. <RSYN.SYNC(bit 6)> 0: Clock synchronization OFF 1: Clock synchronization ON	[RSYN] (WRITE) 7 0 - n - - - - - -
Check the status of clock synchronization for motor control. <RSYN.SYON(bit 7)> 0: The satus not to synchronize clock 1: The status to syncronize clock	[RSYN] (READ) 7 0 n - - - n - - -
Set the limit of correction of clock frequency for motor control. <RSYN.FAL(bit 16 to 22)> Set the limit to correct the frequency at the synchronization. An error interrupt occurs when the correction exceeds the limit. If you set 0, the limit is not controlled.	[RSYN] (WRITE) 23 16 - n n n n n n n
Set control function of clock frequency range for motor control. <RSYN.FAWL(bit 23)> 0: Control function :OFF 1: Control function ON (An error interrupt occur by large frequency change)	[RSYN] (WRITE) 23 16 n - - - - - - -
Monitor the correction amount of clock freqneuncy for motor control. <RSYN.FAM (bit 24 to 31)> You can monitor correction amount (-128 to +127) of the clock frequency. When the correction amount is big, please check a crystal oscillator frequency.	[RSYN] (READ) 31 24 n n n n n n n n

8-17-2. Simultaneous stop function

When the spcified G9103A is stopped by an error, own axis can be stopped by an error.
However, the stop timing delays one cycle of cyclic communication at maximum.

Setting process

1. Specify a device address of G9103A to monitor in RSYN register.
   RSYN.DNSTP ← Device address of G9103A to monitor
   RSYN.SYNE ←1
2. Set RSYN of other G9103As. Set G9103As to be monitored so as to make a loop.
3. In order to transmit the information that own axis stops to other G9103As, start operation with PRMD.MERO=1.
4. In order to stop own axis when other axis stops, start with operation PRMD.MERI=1.

Example for setting

The following is an example when using six G9103As and the device address is 3,5,6,50,55 and 60. In this example, either the device address 3 or 5 of G9103A stops by an error, the device address 3, 5 and 6 are all stopped. If the device address 6 is stopped, the device address 3 and 5 are not stopped. If one of the device addresss 50, 55 or 60 is stopped, the device address 50, 55 and 60 are all stopped.

	G9103A_3	G9103A_5	G9103A_6
RSYN.DNSTP	000110(6)	000011(3)	000101(5)
RSYN.SYNE	1	1	1
PRMD.MERO	1	1	0
PRMD.MERI	1	1	1

	G9103A_50	G9103A_55	G9103A_60
RSYN.DNSTP	111100(60)	110010(50)	110111(55)
RSYN.SYNE	1	1	1
PRMD.MERO	1	1	1
PRMD.MERI	1	1	1

Note. The information of error stop is transmitted by the cyclic communication. Therefore, set the setting value for RSYN.DNSTP to the device address of G9103A that executes cyclic communication to us just before own axis's cyclic communication.
In the case that error interrupt occurs and stops motor by the interrupt, the cause is error stop.

Set a device address of G9103A to be monitor. <RSYN.DNSTP(bit 8 to 13)> G9103A monitors that the G9103A that has a device address (0 to 63) set in this register, stops or not.	[RSYN] (WRITE) 15 8 0 - n n n n n n
Select whether G9103A sends simultaeous stop information. <RSYN.SYNE(bit14)> 0: G9103A does not send simultaneous stop information. 1: G9103A send simultaneous stop information.	[RSYN] (WRITE) 15 8 0 n - - - - - -
Select whether G9103A stops its own axis when it receives simultaneous stop information. <RMD.MERI(bit28)> 0: Even if G9103A that is monitored sends simultaneous stop information, G9103A does not stop its own axis. 1: When G9103A that is monitored sends simultaneous stop information, G9103A stops its own axis.	[RMD] (WRITE) 31 24 - - - n - - - -
Select whether G9103A sends simultaneous stop information or not if own axis stops. <RMD.MERO(bit 29)> 0: Even if own axis stops abnormally, G9103A does not send simultaneous stop information. 1: When own axis stops abnormally, G9103A sends simultaneous stop information.	[RMD] (WRITE) 31 24 - - n - - - - -

8-18. Unit ID control function

If making ROME terminal (Pin12) H level, EEPROM connection mode for ID is available and ID information within EEPROM is downloaded into the general-purpose register (RGN0 to 3) in G9103A automatically after releasing reset. The general-purpose register can be changed and the content of the register is written into EEPROM by the data communication. Pin 12 of G9003 is GND terminal.

Please use an EEPROM that satisfies the following specifications.

- Serial interface with four lines. (SPI)
- Address length is 8 bits.
- Data length is 8 bits.
- More than 16 bytes should be written by one writing command. (Only use 16 bytes.)
- 3.3 V power supply and should be operate with more than 2.5MHz.
- The bit 2 and 3 can be written when “00” is set and cannot be written when “11” is set. when writing status command.
- The commands are as follows.

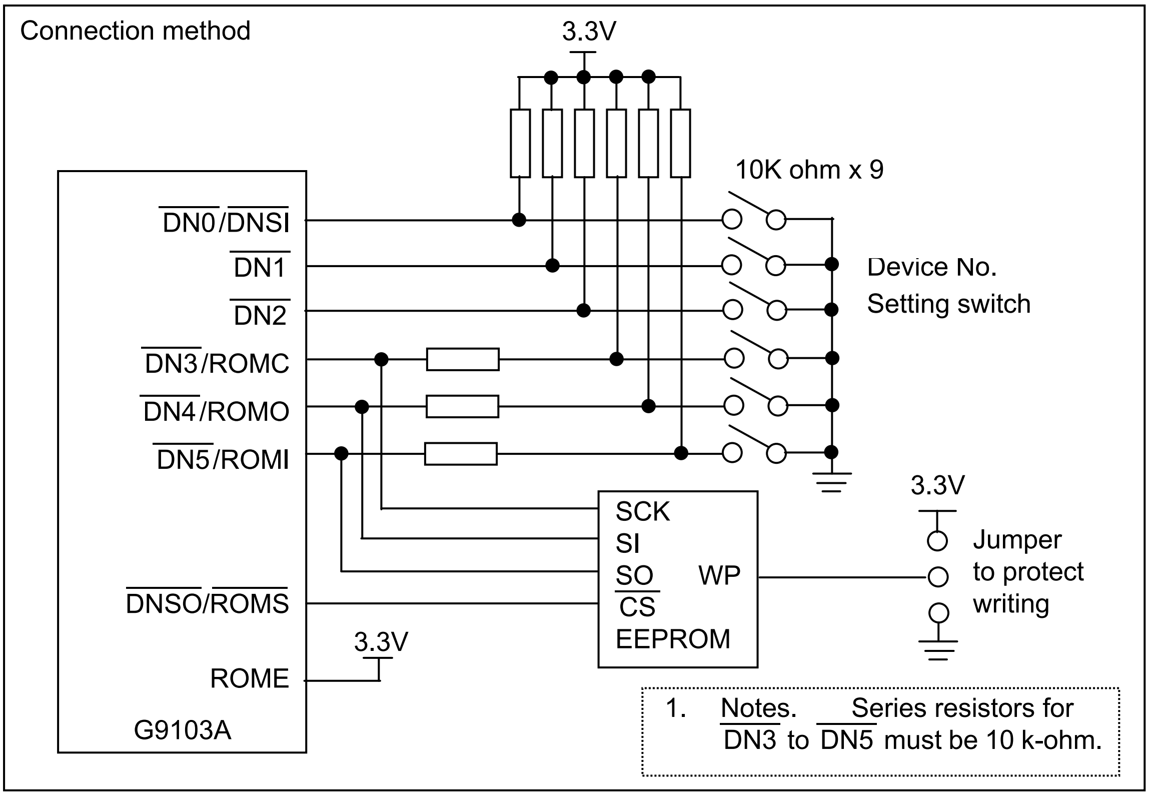
Command	Process
0000 0001	Write to status register
0000 0010	Write data
0000 0011	Read data
0000 0110	Be ready for writing

Example of EEPROM（The capacity to be control is only 16 word/8Bit.）

Manufacturer	Model name	Memory capacity	Power	Speed	Writing time	WP terminal
ROHM	BR25H010-W	128Word/8Bit(1Kbit)	2.5V～5.5V	5MHz	5mS	With
SII	S-25C010A	128Word/8Bit(1Kbit)	2.5V～5.5V	5MHz	4mS	With
Renesas	HN58X2502	256Word/8Bit(2Kbit)	2.5V～5.5V	5MHz	5mS	With
MICROCHIP	25LC010A	128Word/8Bit(1Kbit)	2.5V～5.5V	10MHz	5mS	With
MICROCHIP	25AA010A	128Word/8Bit(1Kbit)	1.8V～5.5V	10MHz	5mS	With

Note. The above information is as of August, 2010.
When you adopt the above, please check each manufacturer's model name and specifications, again.

Connection method



1. Notes. Series resistors for DN3 to DN5 must be 10 k-ohm.

Notes 1. Make the terminal for writing protection to EEPROM available to be change (protection ON or OFF) not as to change the ID within EEPROM by mistakes.

Notes 2. When ROME terminal (PIN 12) is HIGH level, the default value of RGN0 to 3 is 0.

Notes3. In the communication during accessing EEPROM, use device address stored in G9103A

8-18-1. EEPROM control command

There are only 4 commands of G9103A for EEPROM control.

Command name	Code	Contents
ROMPE	000A(h)	It is prohibited to write to the external EEPROM. G9103A performs the following to EEPROM. 1. Send the command (00000110) to permit writing to EEPROM. 2. Secure 3.2μs interval time with ROMS=H. 3. Write 11111100 to EEPROM status register.
ROMPD	000B(h)	Make the external EEPROM permitted to be written. 1. Send the command (00000110) to permit writing to EEPROM. 2. Secure 3.2μs interval time with ROMS=H. 3. Write 11110000 to EEPROM status register 1.
ROMWR	000C(h)	Write the content of RGN0 to 3 to EEPROM. G9103A oerfirns the following to EEPROM. 1. Send the command (00000110) to permit writing to EEPROM. 2. Secure 3.2μs interval time with ROMS=H. 3. Write 16 bytes of RGN0 to 3 to EEPROM.
ROMRD	000D(h)	Read EEPROM and set to RGN 0 to 3. G9103A performs the following to EEPROM. 1. Read from 16 bytes from address 0 at once and write to the RGN0 to 3 registers.

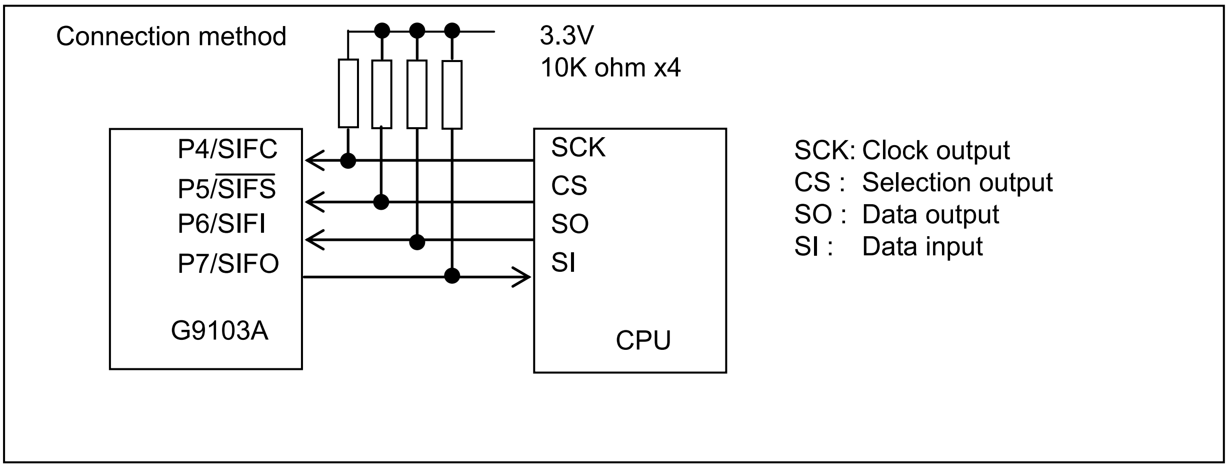
Notes 1. When ROME terminal (PIN12) is HIGH level, the ROMRD command is executed automatically under the following conditions.

1. Just after being reset by the $\overline{\text{RST}}$ input.
2. Just after being reset by the software reset command (0004h)
3. Just after being reset by the broadcast communication command (0010 0ggg 0000 0100)

Note 2. Communication time and writing time to EEPROM is needed until process within EEPROM completes after G9001A sends ROMPE, ROMPD, ROMWR command. Make sure not to send next EEPROM control command or not to power off for this period.
The maximum communication time at 20Mbps is 168μs. For the time to write into EEPROM, see the data sheet for EEPROM. (Normally the time from the sending start to completing to write is less than 10 ms.

8-19. CPU connection function

The general-purpose terminal P4 to 7 are changed to serial I/F control terminals with RENV2.SIFM=1. Writing and reading to RGN0 to 3 by 4 lines serial interface are available. (Other registers except RGN0 to 3 cannot be accessed.) However, RGN0 is for the exclusive use of unit ID control. Therefore, do not write from CPU.



Note 1. After power-on, G9103A's four terminals are input ports until REMV2.SIFM is set to 1 through Motionnet and becomes High level by pull-up.
Note 2. While RENV2.SIFM=1, setting for I/O of RENV2.P4M, P5M, P6M and P7M is disabled.
Note 3. Set communication clock frequency (SCK) 2.5MHz or less.

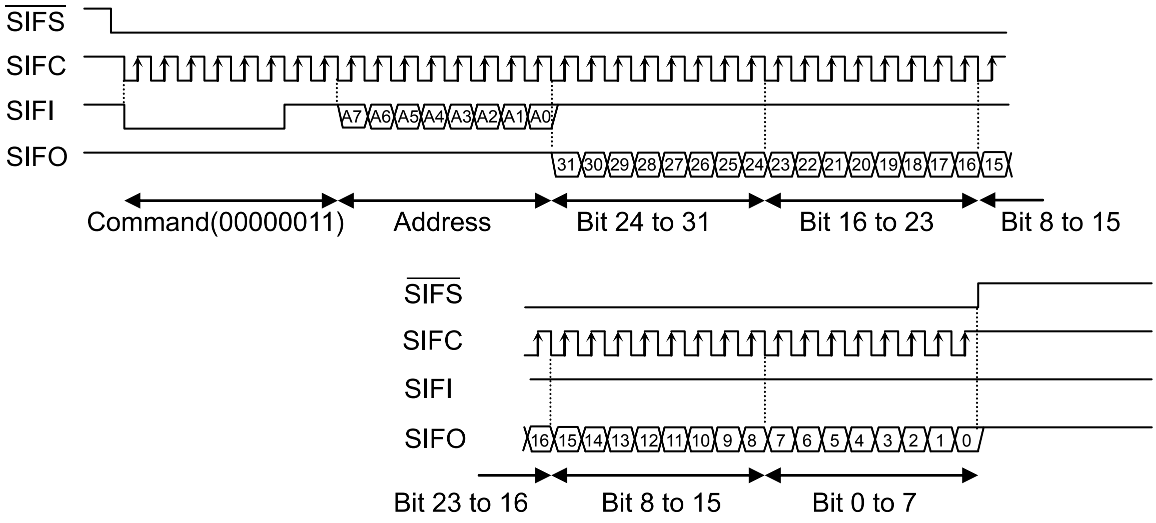
G9103A can be accessed from CPU like serial RAM with 16-bite capacity. Address map is as follows.

Address	Contents		Address	Contents
0	RGN0 bit 24 to 31		8	RGN2 bit 24 to 31
1	RGN0 bit 16 to 23		9	RGN2 bit 16 to 23
2	RGN0 bit 8 to 15		10	RGN2 bit 8 to 15
3	RGN0 bit 0 to 7 (LSB)		11	RGN2 bit 0 to 7 (LSB)
4	RGN1 bit 24 to 31		12	RGN3 bit 24 to 31
5	RGN1 bit 16 to 23		13	RGN3 bit 16 to 23
6	RGN1 bit 8 to 15		14	RGN3 bit 8 to 15
7	RGN1 bit 0 to 7 (LSB)		15	RGN3 bit 0 to 7 (LSB)

Notice for access

Do not access only the specific bits of a register. Access in the unit of 31-bit (0-31). For example, even if you want to read bit 8 to 15 of RGN1, read address 4 to 7. you can access numeral registers at once.

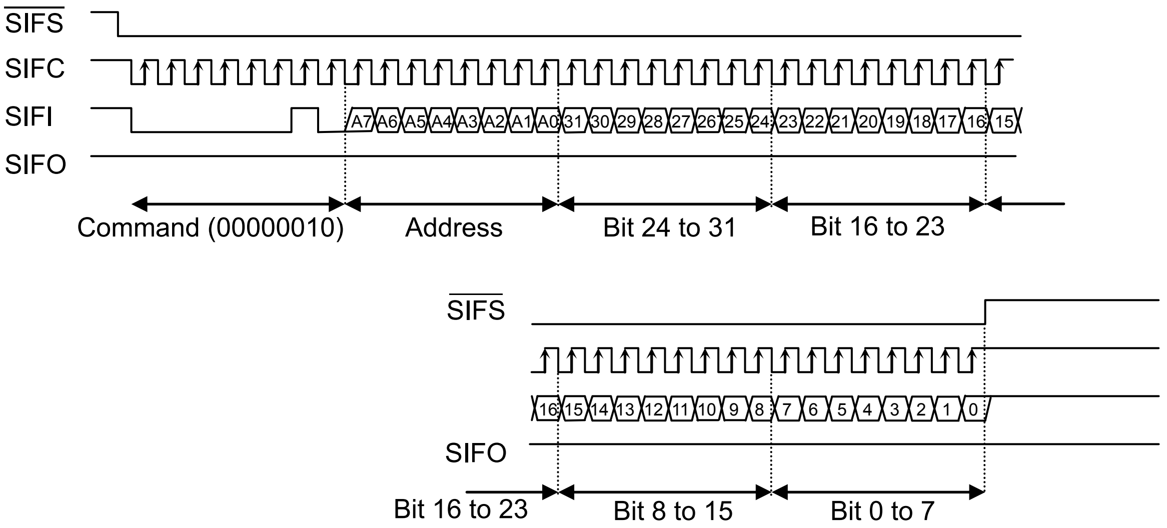
8-19-1. Timing to read (When reading 4 bytes）



1. After set $\overline{\text{SIFS}}$=L, CPU synchronized with the falling of SIFC, input read command (00000011) to G9103A.
2. After that, Synchronized with the falling of SIFC, input the 8 bit read start address from the upper.
3. After that, synchronized with the falling of SIFC, the upper bits outputs 8 bit data.
4. If SIFC is continued to input, address is added +1 and up to 16 bites can be read.
5. CPU read SIF0 data as synchronized with the rising of SIFC.
6. After read to the D0 bit of the last data, set $\overline{\text{SIFS}}$=H.

Note. The upper addresses (A4 to 7) are ignored. The address next to address 15 returns address 0.

8-19-2. Timing to write (when writing 4 bytes）



1. After set $\overline{\text{SIFS}}$=L, CPU synchronized with the falling of SIFC, input read command (00000010) to G9103A.
2. After that, synchronized with the falling of SIFC, input 8-bit read start address from the upper.
3. After that, synchronized with the falling of SIFC, input 8-bit data from the upper.
4. If SIFC is continued to input, address is added +1 and up to 16 bites can be read.
5. CPU read SIFC data as synchronized with the rising of SIFC.
6. After read to the D0 bit of the last data, set $\overline{\text{SIFS}}$=H.

Note. The upper address (A4 to 7) are ignored. The address next to address15 returns address 0.

9. How to calculate the communication cycle time

9-1. Cyclic communication

Cyclic communication is repetition of I/O communication. Therefore, first of all, calculate I/O communication time.
I/O communication consists of a request frame and a response frame. The data that received and send in this communication is controlled so as to be always 4 bytes. Because data amount is constant, communication time is always constant. (However, if the communication rate varies, I/O communication time, of course varies.)

One I/O communication time is shown as follows.

I/O communication time ＝（ 7.7 × K ）+ 7.4 [μsec]

"K" is a communication rate coeffcient. Please refer to the following table.

Commnunication rate(Mbps)	K
20	1
10	2
5	4
2.5	8

Because cyclic communication controls I/O communication so as to repeat I/O communication in order of the device address of the connected local LSIs. Therefore, one cycle of cyclic communication is calculated as follows.

Cycle of cyclic communicaiton =(I/O communication time)x N

＝((7.7xK)+7.4)xN [μsec]

"N"shows the number of local LSIs that are connected.

Ex.: Calculating the cycle time with a communication rate of 20 Mbps and 30 local LSIs.
(7.7 x 1 + 7.4) x 30 = 453μs

9-2. Time required for one complete data communication

There are two types of data communications as follows:
1) When there is a response frame from a local LSI (the data length is variable).
2) When there is no response frame from a local LSI.

Basic item	Required time (μs)
Data sending time (ST)	(B x 0.6 + 3.25) x K
Command execution time (RT)	(C x 0.3)
Response time with data (JT)	(B x 0.6 + 5.65) x K
Response time without data (JT)	5.05 x K

"B" is number of bytes of data. "C" is number of commands.

One complete data communication cycle = ST +RT + JT + 7.4 (μs)

9-3. Total cycle time (including data communication)

The total time can be obtained by adding the data communication times to the ordinary communication cycle time.

Ex.1: In the case of communication rate = 20 Mbps, 34 local LSIs are connected and that data communication (consisted of 2 bytes for sending and 6 bytes for receiving) to read one register in G9103A is performed

Cycle time = Cyclic time + (Data communication time) x Number of times of data communication
= (7.7 x 1 + 7.4) x 34 + {(2 x 0.6 + 3.25) x 1 + (6 x 0.6 + 5.65) x 1 + 7.4} x 4
= 513.4 + 21.1 x 4
= 597.8 μs

Note: The formula above contains some margin. In actual operation, a shorter total time can be obtained. However, when a communication error occurs, a total time may be longer than the above.

10. Electrical Characteristics

10-1. Absolute maximum ratings

Item	Symbol	Rating	Unit
Power supply voltage	V_{DD}	-0.3 to +4.0	V
Input voltage	V_{IN}	-0.3 to +7.0	V
Output voltage	V_{out}	-0.3 to V_{DD} +0.5	V
Input current / pin	I_{out}	±30	mA
Storage temperature	T_{Stg}	-65 to +150	^{o}C

10-2. Recommended operating conditions

Item	Symbol	Rating	Unit
Power supply voltage	V_{DD}	+0.3 to +3.6	V
Input voltage	V_{IN}	-0.3 to 5.5	V
Storage temperature	T_a	-40 to +85	^{o}C

10-3. DC characteristics

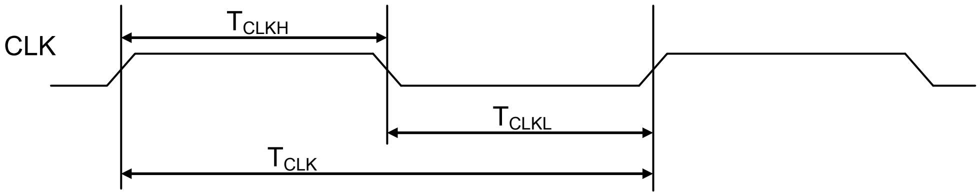
Item	Symbol	Condition	Min.	Max.	Unit
Current consumption	I_{dd}	CLK = 80 MHz, Output 5MHz		80	mA
Off-state leakage current	I_{OZ}	*4,*5	-1	1	μA
Terminal capacitance		*1,*2,*3,*4,*5,*6		10	pF
LOW input current	I_{IL}	V_{IL}=0.0V *1,*2	-1		μA
		V_{IL}=0.0V *3,*4	-166		
		V_{IL}=0.0V *5	-84		
HIGH input current	I_{HL}	V_{IL}=V_{DD} *1,*3,*4,*5		1	μA
		V_{IL}=V_{DD} *2		165	
LOW input voltage	V_{IL}	*1		0.6	V
		*2,*3,*4,*5		0.8	
HIGH input voltage	V_{IH}	*1	2.4		V
		*2,*3,*4,*5	2.0		
LOW output voltage (Note)	V_{OL}	IOL=6mA *6		0.4	V
		IOL=12mA *4,*5		0.4	
HIGH output voltage (Note)	V	IOH=-6mA *6	V_{DD}-0.4		V
		I_{OH}＝-12mA *4,*5	V_{DD}-0.4		
LOW output current	I_{OL}	$V_{OL\,L}$＝0.4V *6		6	mA
		V_{OL}＝0.4V *4,*5		12	
HIGH output current	I_{OH}	V_{OH}＝V_{DD}－0.4V *6	-6		mA
		V_{OH}＝V_{DD}－0.4V *4,*5	-12		
Internal pull up resistance	R_{UP}	* *3,*4	20	120	K-ohm
		*5	40	240	
Internal pull down resistance	R_{DWN}	*2	20	120	K-ohm

Characteristics varies depends on terminals.

*1	SI,CLK
*2	ROME,SOEI,BRK
*3	$\overline{\text{DN0}}$, $\overline{\text{DN1}}$, $\overline{\text{DN2}}$, $\overline{\text{DN5}}$ / ROMI, $\overline{\text{GRP0}}$, $\overline{\text{GRP1}}$, $\overline{\text{GRP2}}$,DNSM, EA, EB, EZ, PA, PB, $\overline{\text{PCS}}$, LTC, CLR, INP, EMG, +EL, -EL, SD, ORG, ALM, ELL, TUD, TMD, CKSL, SPD0, SPD1, $\overline{\text{RST}}$
*4	$\overline{\text{STA}}$, $\overline{\text{STP}}$, P0, P1, P2, P3, P4 / SIFC, P5 / $\overline{\text{SIFS}}$, P6 / SIFI, P7/SIFO
*5	$\overline{\text{DN3}}$ / ROMC, $\overline{\text{DN4}}$ / ROMO
*6	$\overline{\text{DNSO}}$ / $\overline{\text{ROMS}}$, $\overline{\text{SOEL}}$, SOEH, SO, ERC, OUT, DIR, CP1, CP2, CP3, $\overline{\text{BSY}}$ / PH1, $\overline{\text{FUP}}$ / PH2, $\overline{\text{FDW}}$ / PH3, $\overline{\text{MVC}}$ / PH4, $\overline{\text{MSEL}}$, $\overline{\text{MRER}}$, $\overline{\text{TOUT}}$

10-4. AC characteristics

10-4-1. Reference clock



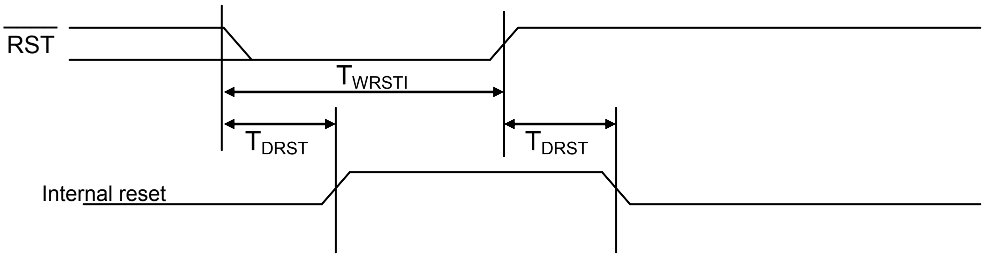
1) When setting CKSL = L

Item	Symbol	Min.	Max.	Unit
Frequency	f_{CLK}	-	40	MHz
Cycle	T_{CLK}	25	-	ns
HIGH duration	T_{CLKH}	10	15	ns
LOW duration	T_{CLKL}	10	15	ns

2) When setting CKSL = H

Item	Symbol	Min.	Max.	Unit
Frequency	f_{CLK}	-	80	MHz
Cycle	T_{CLK}	12.5	-	ns
HIGH duration	T_{CLKH}	-	-	ns
LOW duration	T_{CLKL}	-	-	ns

10-4-2. Reset timing



Item	Symbol	Min.	Max.	Unit
Reset length	T_{WRSTI}	10	-	Clock cycles
Delay time	T_{DRST}	-	10	Clock cycles

Note 1: The G9103A is ready to be used after the internal reset goes LOW.

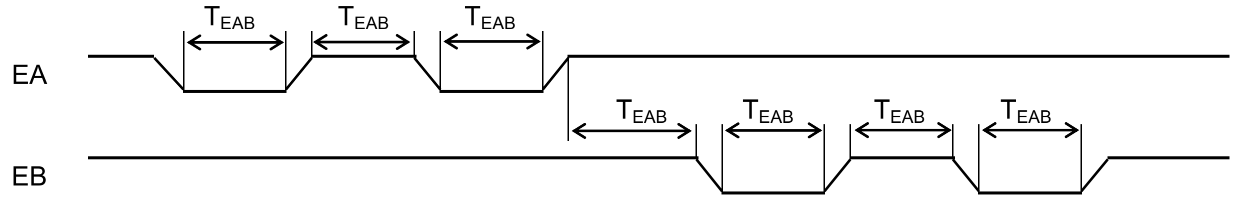
Note 2: While $\overline{RST}$ terminal is low, please input at least 10 cycles of the reference clock.
Even while resetting, make sure the clock signal is continuously available to the LSI.
If the clock is stopped while resetting, the LSI cannot be reset normally.

10-5. Operation timing

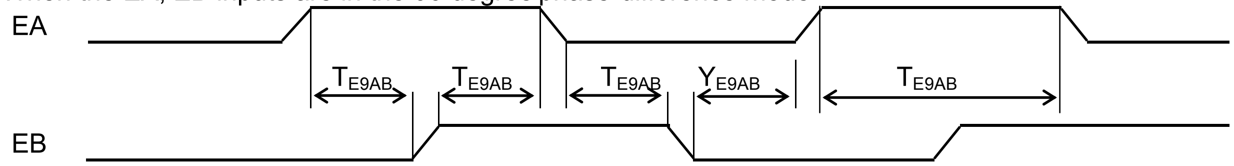
Item		Symbol	Condition	Min.	Max.	Unit
$\overline{\text{RST}}$ input signal width			Note 1	$10T_{CLK}$		ns
CLR input signal width			Note 2	$4T_{ICK}$		ns
EA, EB input signal width (Two-pulse)		T_{EAB}	Note 3	$2T_{ICK}$ ($6T_{ICK}$)		ns
EA, EB input signal phase (90 degree)		T_{E9AB}		$2T_{ICK}$		ns
EA, EB input signal width (90 degree)		T_{E9W}	Note 3	$4T_{ICK}$ ($6T_{ICK}$)		ns
EZ input signal width			Note 3	$2T_{ICK}$ ($6T_{ICK}$)		ns
PA, PB input signal width (2 pulses)		T_{PAB}	Note 4	$2T_{ICK}$ ($6T_{ICK}$)		ns
PA, PB input signal phase (90 degree)		T_{P9AB}		$2T_{ICK}$		ns
PA, PB input signal width (90 degree)		T_{P9W}	Note 4	$4T_{ICK}$ ($6T_{ICK}$)		ns
ALM input signal width			Note 5	$4T_{ICK}$		ns
INP input signal width			Note 5	$4T_{ICK}$		ns
ERC output signal width			RENV1 bit 12 to 14 = 000	254 x $2T_{ICK}$	255 x $2T_{ICK}$	ns
			RENV1 bit 12 to 14 = 001	254 x $16T_{ICK}$	255 x $16T_{ICK}$	
			RENV1 bit 12 to 14 = 010	254 x $64T_{ICK}$	255 x $64T_{ICK}$	
			RENV1 bit 12 to 14 = 011	254 x $256T_{ICK}$	255 x $256T_{ICK}$	
			RENV1 bit 12 to 14 = 100	254 x $2048T_{ICK}$	255 x $2048T_{ICK}$	
			RENV1 bit 12 to 14 = 101	254 x $8192T_{ICK}$	255 x $8192T_{ICK}$	
			RENV1 bit 12 to 14 = 110	254 x $16384T_{ICK}$	255 x $16384T_{ICK}$	
			RENV1 bit 12 to 14 = 111	LEVEL output		
+EL, -EL input signal width			Note 5	4 T_{ICK}		ns
SD input signal width			Note 5	4 T_{ICK}		ns
ORG input signal width			Note 5	4 T_{ICK}		ns
PCS input signal width				4 T_{ICK}		ns
LTC input signal width				4 T_{ICK}		ns
$\overline{\text{STA}}$	Output signal width			16 T_{ICK}		ns
	Input signal width			10 T_{ICK}		ns
$\overline{\text{STP}}$	Output signal width			16 T_{ICK}		ns
	Input signal width			10 T_{ICK}		ns
$\overline{\text{BSY}}$ signal ON delay time		T_{SOEBSY}			-12 T_{ICK}	ns
		T_{STABSY}			14 T_{ICK}	ns
Start delay time		T_{SOEPLS}			8 T_{ICK}	ns
		T_{STAPLS}			34 T_{ICK}	ns
Output port delay time		T_{SOEPRT}			$-26T_{ICK}$	ns

Note 1: "T_{ICK}" in the table above means one cycle (25 ns) of the internal clock 40 MHz.
Note 2: The actual CLK input signal should be input more than 10 cycles while $\overline{\text{RST}}$ terminal is LOW.
Note 3: If the input filter is ON < RENV2.EINF=0>, the minimum time will be $6T_{ICK}$.
Note 4: If the input filter is ON < RENV2.PINF=0>, the minimum time will be $6T_{ICK}$.
Note 5: If the input filter is ON < RENV2.FLTR=0>, the minimum time will be $160T_{ICK}$.
Note 6. If the synchronization function of the clock for motor control is operating, ±25ns error occurs.

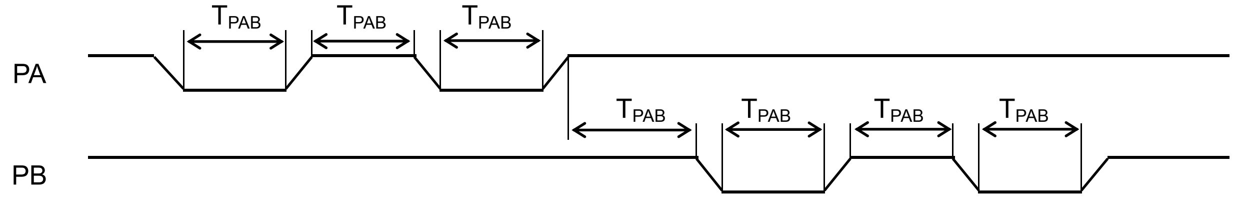
1) When the EA, EB inputs are in the Two-pulse mode



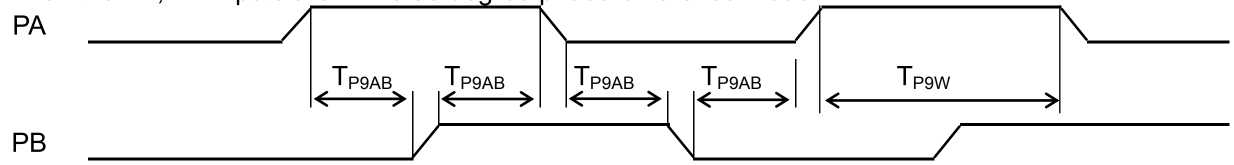
2) When the EA, EB inputs are in the 90 degree phase-difference mode



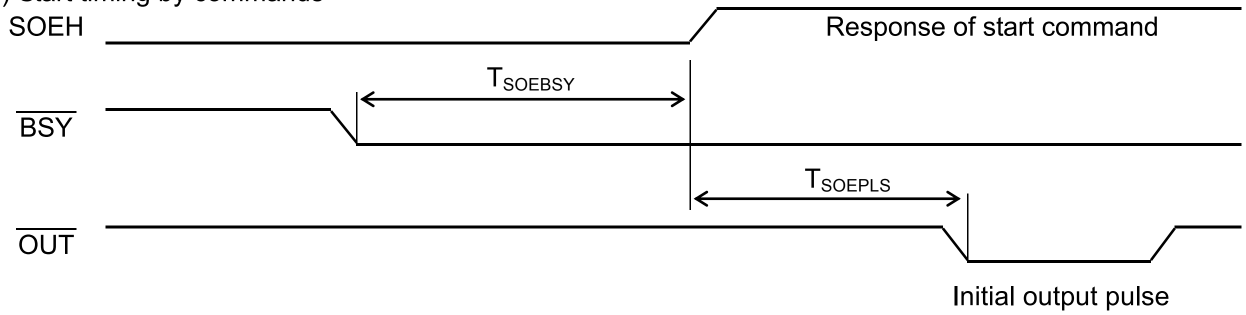
3) When the PA, PB inputs are in the Two-pulse mode



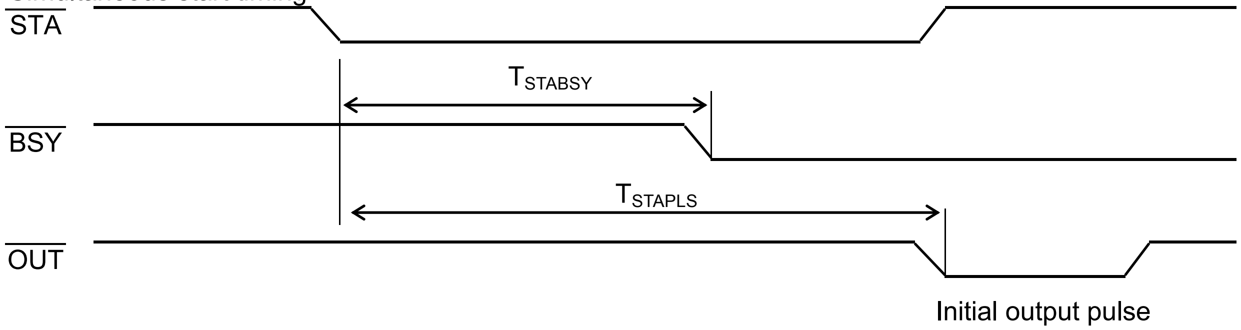
4) When the PA, PB inputs are in the 90 degree phase-difference mode



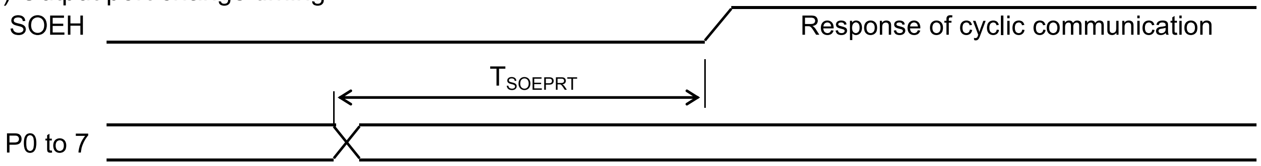
5) Start timing by commands



6) Simultaneous start timing



7) Output port change timing

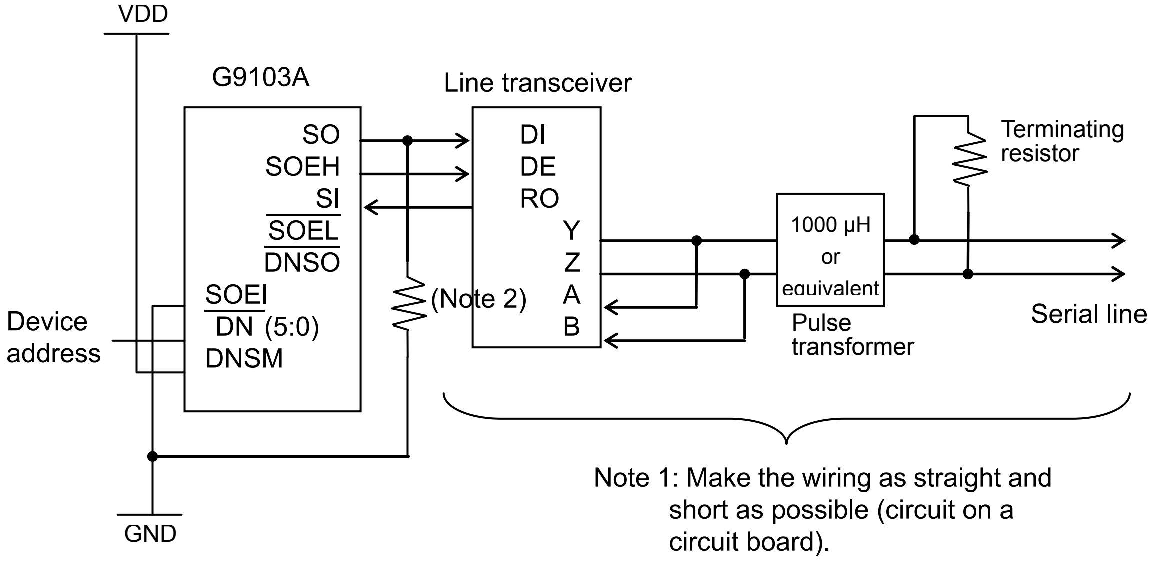


11. Communication example

11-1. G9103A, line transceiver, and pulse transformer

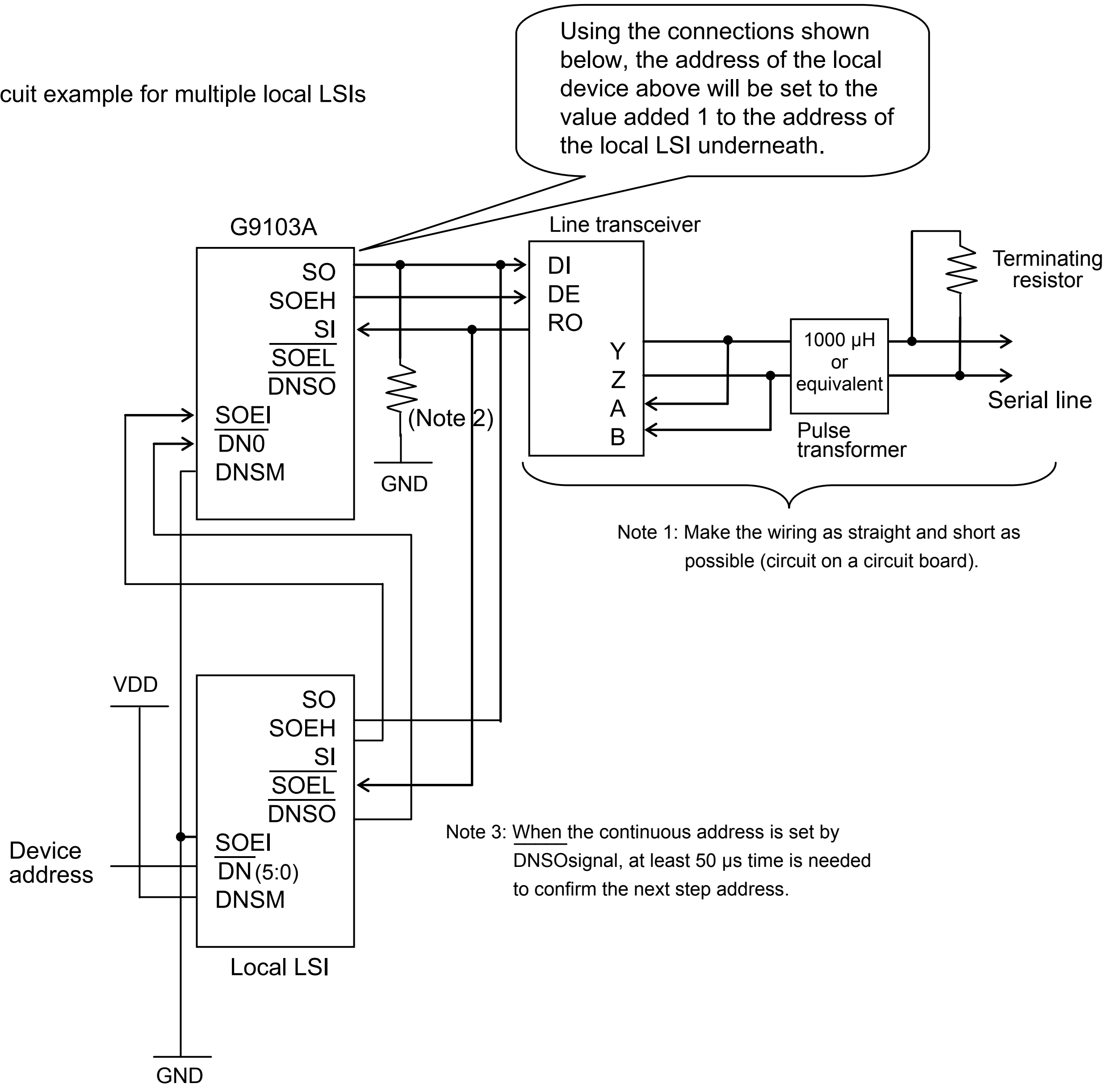
Use RS-485 line transceivers and pulse transformers (1000 μH or equivalent) to make serial communication connections.
Connect the line transceivers as shown below.
Connect terminating resistors (which match the cable impedance) at both ends of the transmission line.
The terminating resistors can be either at front or back of the pulse transformer. The same effect will be obtained at either position.
SO, SI, SOEH, $\overline{\text{SOEL}}$ terminals have 5V tolerant feature, they can be connected with TTL level 5V line transceiver directly.

(1) Circuit example for a single local LSI



Note 1: Make the wiring as straight and short as possible (circuit on a circuit board).

(2) Circuit example for multiple local LSIs


Using the connections shown below, the address of the local device above will be set to the value added 1 to the address of the local LSI underneath.

Note 1: Make the wiring as straight and short as possible (circuit on a circuit board).

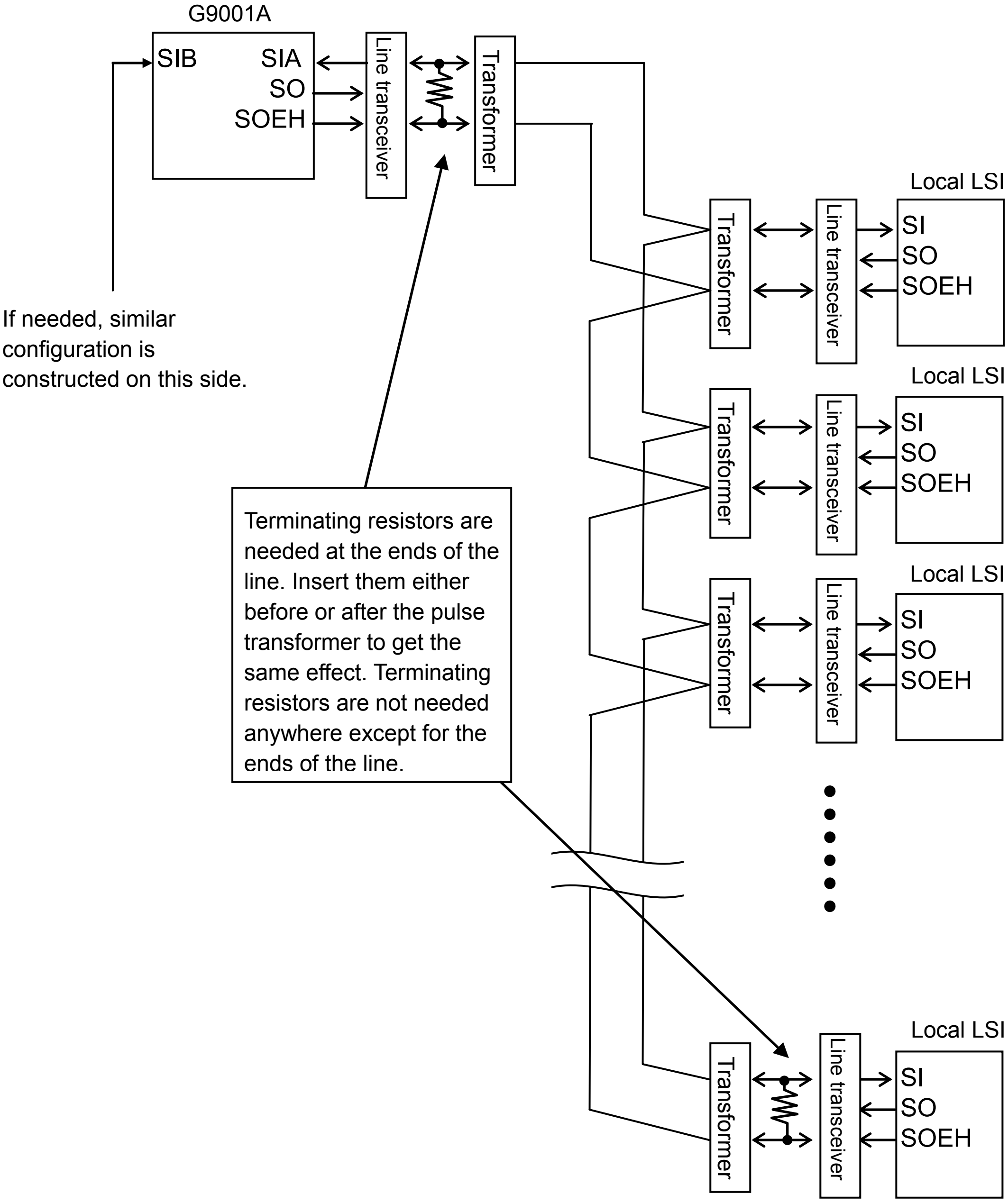
Note 3: When the continuous address is set by DNSOsignal, at least 50 μs time is needed to confirm the next step address.

Note 1: When connecting the serial lines (actually, connectors) to line transceivers, make the path as short and straight as possible. Running these lines on a PC board could deteriorate the communication performance.

Note 2: Pull down resistors to GND should be 5 to 10 k-ohms.

11-2. Complete configuration

We recommend a configuration with the center LSI (G9001A) at one end of the line and the local LSIs at the other end, as shown below.
If you want to place the center LSI in the middle of the system, use two communication lines so that the center LSI is effectively at the end of each line.
However, when using synchronization function with other axes, connect all LSIs synchronized in one communication line.



G9001A
SIB
SIA
SO
SOEH
Line transceiver
Transformer
Local LSI
SI
SO
SOEH
If needed, similar configuration is constructed on this side.
Terminating resistors are needed at the ends of the line. Insert them either before or after the pulse transformer to get the same effect. Terminating resistors are not needed anywhere except for the ends of the line.

12. Recommended environment

Shown below are the results of our experimental communication results and the environment used for the experiment.
These results can be used to design your own system. However, other system configurations are possible.
The example below is only for your reference.

Conditions						Results
Communication rate	Number of local LSIs	Cable used	Terminating resistor	Pulse transformer	I/F-IC	Max. length
20 Mbps	32	CAT5	100 ohm	1000 μH	RS485	100 m
20 Mbps	64	CAT5	100 ohm	1000 μH	RS485	50 m
10 Mbps	64	CAT6	100 ohm	1000 μH	RS485	100 m

Note: In the figures above, the maximum length figures are results from ideal conditions in a laboratory. In actual use, the results may not be the same.

12-1. Cable

Commercially available LAN cables were used.
CAT5: Category 5
CAT6: Category 6
We used these LAN cables because they are high quality, inexpensive, and easy availability. Lower quality cables (such as cheap instrumentation cables) may significantly reduce the effective total length of the line.
LAN cables normally consist of several pairs of wires. Make sure to use wires from the same pair for one set of communication lines.
Even in the same category and rating, performance of each cable may be different according to manufacturers. Always use the higher quality cables.

12-2. Terminating resistor

Select resistors that match with the impedance of the cable used.
Normally, a 100-ohm resistor is recommended. Therefore, we used terminating resistors with this value.
Adjusting this resistor value may improve the transmission line quality.

12-3. Pulse transformer

We recommend using a pulse transformer, in order to isolate the GND of each local LSI.
By isolating the GNDs, the system will have greater resistance to electrical noise. If pulse transformers are not used, the transmission distance may be less.
We used 1000 μH transformers in our experiments.

12-4. I/F -IC

We selected I/C chips with specifications better than the RS485 standard.
In the experiment, we used 5 V line transceivers. When 5 V line transceivers are used, level shifters are needed to make the connections. (To use G9103A, do not need level shifters.)

12-5. Parts used in our experiments

Show below is a list of the parts used in the interface circuits of our experiments.
Use of other parts may change the system's response. This list is only for your reference.

Parts	Manufacturer	Model name
CAT5	Oki Wire Co., Ltd.	F-DTI-C5 (SLA)
CAT6	Oki Wire Co., Ltd.	DTI-C6X
Pulse transformer	Nippon Pulse Motor, Co., Ltd.	NPT102F
Line transceiver	TEXAS INSTRUMENTS	SN75LBC180AP

Note. The above information is as of August, 2010.
When you adopt the above, please check each manufacturer's model name and specifications, again.

12-6. Other precautions

- Cables
When you are planning long distance transmission, cable quality will be the most important factor.
Specialized cables designed for use as field busses, such as those by CC-Link and LONWORKS, have guaranteed quality and may be easier to use.

- Pulse transformers
Needless to say, the pulse transformers should handle 20 Mbps (10 MHz) without becoming saturated.
The transformer's inductance is also important.
Since up to 64 pulse transformers may be connected, that should be considered for selection.
We used 1000 μH transformers in our experiments. If there is a product that satisfy necessary response capacity and have bigger value, the product is worth being tried to use.

- Line transceivers
We used TEXAS Instruments ICs for the experiments.
Other possibilities are available from MAXIM and LINEAR TECHNOLOGY, who offer very high performance transceivers.

- Connectors
If possible, the connectors should match the cable characteristics.
Although we did not use them, modular type connecters must be better for LAN cables.

- Cable connections
Do not connect one cable to another cable (using connectors etc.).
In a multi-drop system, the number of cables increases as the number of local LSIs increase. However, connecting a cable just to extend the line should be avoided.

- Processing of excess cable
If cables is too long, cut the excess length of cable if possible.
Unneeded cable length may restrict the line overall usable length, and may introduce electrical noise.

- Circuit board
Create circuits on a board with 4 or more layers, to prevent the introduction of noise.

13. Software example

This chapter outlines software for the center LSI (G9001A) using flow charts.
In the flow charts, required variables are used for convenience. The following address map and status command is an example of the center LSI. For the detail, see the manual for the center LSI.

13-1. Environment and precautions used for the descriptions

The descriptions below assume that I/F mode 3 is selected for the CPU mode of the center LSI. Therefore, a 16-bit data bus is used.
Also, these descriptions are based on the assumption that the wiring connections around the center LSI have been properly prepared and that the connected local LSIs are turned on. And, of course, we presume that connections to the serial line and the termination resistances are all correct.

13-2. Commands used

We will use the following two commands to access the address map in the center LSI.

1) Write command to the center LSI

Outpw (Address, Data)	
Address	Value corresponding to the address map in the center LSI The lowest bit is fixed to 0.
Data	Write data (16 bits)
Return value	None

2) Read command from the center LSI

Inpw (Address)	
Address	Value corresponding to the address map in the center LSI The lowest bit is fixed to 0.
Return value	Read data (16 bits)

13-3. Center LSI (G9001A) address map

For details, see the user's manual for the center LSI.

Interface mode: I/F mode 3

A1 to A8		Writing	Reading
0 0000 000	000(h)	Command bits 0 to 15	Status bits 0 to 15
0 0000 001	002(h)	Invalid	Interrupt status bits 0 to 15
0 0000 010	004(h)	Input buffer bits 0 to 15	Output buffer bits 0 to 15
0 0000 011	006(h)	Data transmission FIFO bits 0 to 15	Data reception FIFO bits 8 to 15

0 0000 100
|
0 0111 011
	008(h)
	|
	076(h)
		Not defined (56 words)
		(Any data written here will be ignored.)
			Not defined (56 words)
			(Always read as 00(h).)

0 0111 100	078(h)	Device information (Device No. 0, 1)	Device information (Device No.0, 1)
|	|	|	|
0 1011 011	0B6(h)	Device information (Device No. 62, 63)	Device information (Device No.62, 63)
0 1011 100	0B8(h)	I/O communication error flags (Device No. 0 to 15)	I/O communication error flags (Device No. 0 to 15)
|	|	|	|
0 1011 111	0BE(h)	I/O communication error flags (Device No. 48 to 63)	I/O communication error flags (Device No. 48 to 63)
0 1100 000	0C0(h)	Input change interrupt settings (Device No. 0 to 3)	Input change interrupt settings (Device No. 0 to 3)
|	|	|	|
0 1101 111	0DE(h)	Input change interrupt settings (Device No. 60 to 63)	Input change interrupt settings (Device No. 60 to 63)
0 1110 000	0E0(h)	Input change interrupt flags (Device No. 0 to 3)	Input change interrupt flags (Device No. 0 to 3)
|	|	|	|
0 1111 111	0FE(h)	Input change interrupt flags (Device No. 60 to 63)	Input change interrupt flags (Device No. 60 to 63)
1 0000 000	100(h)	Port data No. 0,1 (Device No.0 - Port 0, 1)	Port data No. 0,1 (Device No.0 - Port 0, 1)
1 0000 001	102(h)	Port data No. 2,3 (Device No.0 - Port 2, 3)	Port data No. 2, 3 (Device No.0 - Port 2, 3)
|	|	|	|
1 1111 110	1FC(h)	Port data No. 252, 253 (Device No.63 - Port 0, 1)	Port data No. 252, 253 (Device No.63 - Port 0, 1)
1 1111 111	1FE(h)	Port No.254, 255 (Device No.63 - Port 2, 3)	Port data No. 254, 255 (Device No.63 - Port 2, 3)

Note: The hexadecimal notation for the addresses above are written with the assumption that A0 = 0.

13-4. Center LSI status

For details, see the user's manual for the center LSI.

STSW

STSB1

15	14	13	12	11	10	9	8
0	DBSY	RBSY	SBSY	0	RDBB	TDBB	REF

STSB0

7	6	5	4	3	2	1	0
0	CAER	ERAE	EDTE	EIOE	IOPC	BRKF	CEND

Bit	Symbol	Description
0	CEND	Becomes 1 when ready for data to be written to the transmission FIFO buffer. When the system communication or data communication is complete and the next data can be sent to the transmission FIFO buffer, this bit becomes 1 and the center LSI outputs an interrupt request signal ($\overline{\text{INT}}$). The way to clear this bit varies on the setting of RENV0.MCLR.
1	BRKF	When the center LSI receives a break frame, this bit becomes 1 and an interrupt request signal ($\overline{\text{INT}}$) is output. The way to clear this bit varies on the setting of RENV0.MCLR.
2	IOPC	Becomes 1 when the status of any input port that had enabled the “input change interrupt setting” changed. Then, the center LSI outputs an interrupt request signal ($\overline{\text{INT}}$). This signal is an OR of 256-bit "input port change interrupt flag". When all the bits return to 0, this bit returns to 0.
3	EIOE	Becomes 1 when an I/O communication error occurs. Then, the center LSI then outputs an interrupt request signal ($\overline{\text{INT}}$). This signal is an OR of 64 bit "I/O communication error flag". When all the bits return to 0, this bit returns to 0.
4	EDTE	Becomes 1 when a data or system communication error occurs. Then, the center LSI outputs an interrupt request signal ($\overline{\text{INT}}$). The way to clear this bit varies on the setting of RENV0.MCLR. Note 1
5	ERAE	Becomes 1 when a "local LSI reception processing error" occurs. Then, the center LSI outputs an interrupt request signal ($\overline{\text{INT}}$). Once the status of this bit is read, it returns to 0. Then, the device address and details where the reception processing error occurred can be checked by reading the interrupt status. The way to clear this bit varies on the setting of RENV0.MCLR.
6	CAER	A CPU access error occurred. When there is a problem accessing to a CPU, such as a data sending command being written when there is no data to send, this bit becomes 1. Then, the center LSI outputs an interrupt signal request ($\overline{\text{INT}}$). Once the status of this bit is read, it returns to 0. The details of the error can be checked by reading the interrupt status.
7	(Not defined)	Always 0.
8	REF	When there is unsent output port data, this bit becomes 1. If data is written to the output port area, this bit becomes 1. When cyclic communication to all the ports has completed, this bit returns to 0.
9	TDBB	When there is data to send in the transmission FIFO, this bit becomes 1. After data is written to the transmissionn FIFO, this bit becomes 1. Once a data sending command or a transmission FIFO reset command is written, this bit returns to 0.
10	RDBB	When data has been received in the reception FIFO, this bit becomes 1. When receiving data from a local LSI, this bit becomes 1. After a CPU has read all of the data received, this bit returns to 0.
11	(Not defined)	Always 0.
12	SBSY	Becomes 1 when I/O communication (cyclic communication) starts.
13	RBSY	Is 1 during a reset.
14	DBSY	Is 1 during system communication or data communication.
15	(Not defined)	Always 0.

Note 1: The device address of a local LSI that an error has occurred can be checked by an interrupt status.

13-5. Center LSI Interrupt status

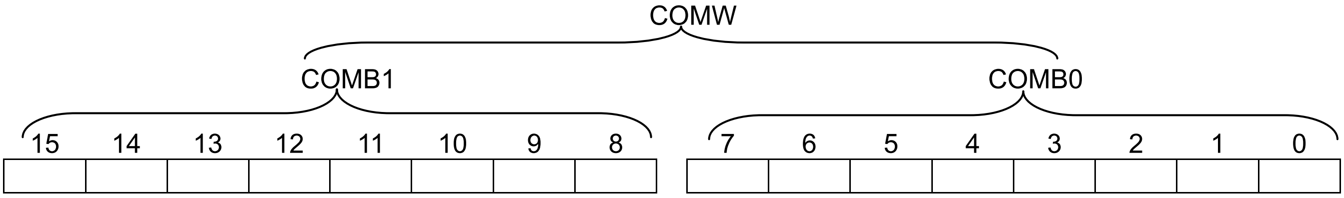
For details, see the user's manual for the center LSI.

ISTW															
ISTB1								ISTB0							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAE3	CAE2	CAE1	CAE0	ERA3	ERA2	ERA1	ERA0	LNRV	0	EDN5	EDN4	EDN3	EDN2	EDN1	EDN0

Bit	Symbol	Description
0 to 5	EDN0 to 5	Contains the device address of the device with an EDTE = 1 or ERAE = 1 (error from receiving I/O data that is different from PMD setting). These details are stored until an error occurs next time.
6	(Not defined)	Always 0.
7	LNRV	When a local LSI is not receiving data, this bit is 1. When the data communication or system communication terminates with an error (EDTE = 1) (only when receiving attribute information), and if a local LSI cannot receive data from the center LSI, this bit becomes 1. When the local LSI has received the data, this bit returns to 0. This value is stored until an error occurs next time.
8 to 11	ERA0 to 3	These are identification codes for received data processing errors on a local LSI. The code is stored until an error occurs next time. 0001: Received I/O data is different from the PMD settings. 0010: An local LSI for I/O communication receives a data communication frame. 0011: Frames larger than the receiving buffer capacity is received in data communication.
12 to 15	CAE0 to 3	These are access error codes from a CPU. The code is stored until an error occurs next time. 0001: When no local LSIs are used an I/O communication start command is written. 0010: Try to write data with a start sending command without any data to send. 0011: While DBSY = 1, a LSI tries to do one of the following: (1) Reading or writing to the transmission or receiving FIFO. (2) Wrote a system start command or a data communication start command. 0100: Tried to send data to device address of a LSI that is not is in use.

13-6. Center LSI command

For details, see the user's manual for the center LSI.



Note1: Write to the 8-bit CPU I/F (IF0=H, IF1=1) in the following order: COMB0 then COMB1.
Note2: # symbol bit: The upper bits of the device address should be set in order, starting from the left end of the # bits.
& symbol bit: When the port is 0 or 1, set the bit to 0. When the port is 2 or 3, set the bit to 1
x symbol bit: Either 0 or 1 may be used.

13-6-1. Operation command

Command	Description
0000 0000 0000 0000 (0000(h))	NOP Invalid command.
0000 0001 0000 0000 (0100(h))	Resets by software. Resets the G9001A. This is the same function as $\overline{\text{RST}}$ input.
0000 0010 0000 0000 (0200(h))	Resets the transmssion FIFO. Resets only the data transmission FIFO.
0000 0011 0000 0000 (0300(h))	Resets the receiving FIFO. Resets only the data receiving FIFO.
0000 0100 $$$ 00$$ (04xxh)	Command to clear INT system status. $ bits of this command corresponds to the followings. 7: 0 6: CAER 5: ERAE 4: EDTE 3: 0 2: 0 1: BRKF 0: CEND By setting each bit to 1, correcpondent status is cleared. However, when RENV0.MCLR=0, this command is invalid.
0000 0110 0000 0000 (0600h)	Command to clear an error counter Reset a register of error count to 0.
0000 0110 0001 0000 (0610h)	Break communication command. Turning off automatic break function by setting RENV0.BKOF = 1, you can perform break communication at artibrary timing by this command.
0001 0000 0000 0000 (1000(h))	System communication to all local LSIs. Polls all of the local LSIs (device No. 0 to 63) one by one, and refreshes the "device information" areas that correspond to each device address. The "device information" contains the following: - Device in use: 0 when no response, and 1 when it responds. - Device type: Reset to 1 when it is a LSI that can perform data communication. - I/O setting information
0001 0001 0000 0000 (1100(h))	System communication to all local LSIs except those devices excluded from cyclic communication. After checking the "device information" area, the center LSI polls all the local LSIs whose device-in-use bit is set to 0, one by one, and refreshes the "device information" areas that correspond to each device address. The details refreshed are the same as by writing a command 1000(h).
0001 0010 00## #### (1200(h) to 123F(h))	System communication to specified local LSIs. Device information area of the local LSIs specified with ###### is revised. The details refreshed are the same as by writing a command 1000(h).

Command	Description
0001 0011 00## #### (1300(h) to 133F(h))	Obtain attribute information for the specified devices. Attribute information of the local LSIs specified with ###### is c opied to the Data reception FIFO. "Device information" area does not change. The details of the Data reception FIFO is as follows. Bits 0 to 4: (Number of bytes for the longest piece of data) / 8 -1 Bits 5 to 7: Not used (not defined) Bits 8 to 15: Device type code (LSI for I/O communication: 01h, LSI that can perform data communication: 81h) Bits 16 to 18: Set the I/O port (PMD terminal information when an LSI for I/O communication is selected) Bit 19: Always 0 Bits 20 to 31: Type of an LSI that can perform data communication (G9003: 000h, G9004A: 001h, G9103A:002h)
0010 0ggg cccc cccc (2000(h) to 270C(h))	Broadcast communication Send “cccc cccc” command to the groups specified by ggg all at once. Set the group number on $\overline{\text{GRP0}}$ to $\overline{\text{GRP2}}$ terminals or RENV2.GN0 to GN2 for G9103A. All 0 to 7 groups are intended for ggg = 000 Command (“cccc cccc”) 0000 0001: Start (CMSTA command to numeral axes) 0000 0010: Stop (CMSTP command to numeral axes) 0000 0011: Emergency stop (CMEMG command to numeral axes) 0000 0100: Reset local LSIs (SRST command to numeral axes) 0000 0101: Latch counter (LTCH command to numeral axes) 0000 0110: Immediately stop (STOP command to numeral axes) 0000 0111: Decelerate and stop (SDSTP command to numeral axes) 0000 1000: Change to FL speed immediately (FCHGL command to numeral axes) 0000 1001: Change to FH speed immediately (FCHGH command to numeral axes) 0000 1010: Decelerate to FL speed (FSCHL command to numeral axes) 0000 1011: Accelerate to FH speed (FSCHH command to numeral axes) 0000 1100: Copy pre-register for operation to register (speed change, etc.) (PRESHF command to numeral axes.)
0011 0000 0000 0000 (3000(h))	Start I/O communication Start I/O communication with devices that have a 1 in the "device-in-use" bit in the "device information".
0011 0001 0000 0000 (3100(h))	Stop I/O communication. Stop the current I/O communication.
0100 0000 00## #### (4000(h) to 403F(h))	Data communication. Sends data in the transmission FIFO to the specified devices with ######. The data received in response will be stored in the reception FIFO.
0100 0001 0000 0000 (4100(h))	Cancel data communication Halts the data communication and resets the transmssion FIFO. This command will be ignored after the data has been sent.

13-6-2. Indirect access command

If CPU can access to all address map (512 bytes) that the center LSI requests, you do not have to use indirect access commands.
You can use these commands to access to the center LSI only in the range of 8 bytes.

<table>
<tr><th>Command</th><th>Description</th></tr>
<tr><td>0101 0000 0### ##xx
(5000(h) to 507F(h))</td><td>Write to the "Device information" area.
The contents of the I/O buffer are written into a word in the device information area.
As an example, the relationship between the I/O buffer details and the device information area are listed below.
<table>
<tr><th>Command</th><th>I/O buffer</th><th>Address</th><th>Local No.</th></tr>
<tr><td rowspan="2">5000(h)</td><td>Bits 0 to 7</td><td>078(h)</td><td>0</td></tr>
<tr><td>Bits 8 to 15</td><td>079(h)</td><td>1</td></tr>
<tr><td rowspan="2">5004(h)</td><td>Bits 0 to 7</td><td>07A(h)</td><td>2</td></tr>
<tr><td>Bits 8 to 15</td><td>07B(h)</td><td>3</td></tr>
<tr><td rowspan="2">5008(h)</td><td>Bits 0 to 7</td><td>07C(h)</td><td>4</td></tr>
<tr><td>Bits 8 to 15</td><td>07D(h)</td><td>5</td></tr>
<tr><td rowspan="2">500C(h)</td><td>Bits 0 to 7</td><td>07E(h)</td><td>6</td></tr>
<tr><td>Bits 8 to 15</td><td>07F(h)</td><td>7</td></tr>
</table>
</td></tr>
<tr><td>0101 0001 0##x xxxx
(5100(h) to 517F(h))</td><td>Write to the "I/O communication error flag" area.
The contents of the I/O buffer are written into a word in this area.</td></tr>
<tr><td>0101 0010 0### #xxx
(5200(h) to 527F(h))</td><td>Write to the "input change interrupt setting" area.
The contents of the I/O buffer are written into a word in this area.</td></tr>
<tr><td>0101 0011 0### #xxx
(5300(h) to 537F(h))</td><td>Write to the "input change interrupt flag" area.
The contents of the I/O buffer are written into a word in this area.</td></tr>
<tr><td>0101 0100 0### ###&
(5400(h) to 547F(h))</td><td>Write to the "port data" area.
The contents of the I/O buffer are written into a word in this area.</td></tr>
<tr><td>0110 0000 0### ##xx
(6000(h) to 607F(h))</td><td>Read the "device information" area.
The contents of the word in this area are copied to the I/O buffer.</td></tr>
<tr><td>0110 0001 0##x xxxx
(6100(h) to 617F(h))</td><td>Read the "I/O communication error flag" area.
The contents of the word in this area are copied to the I/O buffer.</td></tr>
<tr><td>0110 0010 0### #xxx
(6200(h) to 627F(h))</td><td>Read the "input change interrupt setting" area.
The contents of the word in this area are copied to the I/O buffer.</td></tr>
<tr><td>0110 0 11 0### #xxx
(6300(h) to 637F(h))</td><td>Read the "input change interrupt flag" area.
The contents of the word in this area are copied to the I/O buffer.</td></tr>
<tr><td>0110 0100 0### ###&
(6400(h) to 647F(h))</td><td>Read the "port data" area.
The contents of the word in this area are copied to the I/O buffer.</td></tr>
</table>

13-6-3 Register access command

Command	Description
0101 0101 0000 0000 (5500h)	RENV0 write command When CPU sets data to input/output buffer and issues this command, the content of the input/output buffer is copied to RENV0 register.
0110 0101 0000 0000 (6500h)	RENV0 read command When this command is issued, the value of RENV0 register is copied to input/output buffer.
0110 0101 0000 0001 (6501h)	Error counter read command When this command is issued, the value of error counter register is copied to input/output buffer.
0110 0101 0000 0010 (6502h)	Cycle of cyclic communication read command When this command is issued, the value of register for cycle of cyclic communication is copied to input/output buffer.

0110 0101 0000 0011 (6503h)	Recipient address register read command When this command is issued, the value of register for receipient address is copied to input/output buffer.
0110 0101 0000 0100 (6504h)	Version information register read command When this command is issued, the value of register for version information is copied to input/output buffer.

Note: The above registers are not allocated on the address mapp.
Therefore, please access by the above commands.

13-7. Center LSI (G9001A) registers

13-7-1 RENV0 register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	MCLR	BKOF	0	MCSE	MERE	MEDE	MEIE	MIOP	MBRK	MCED

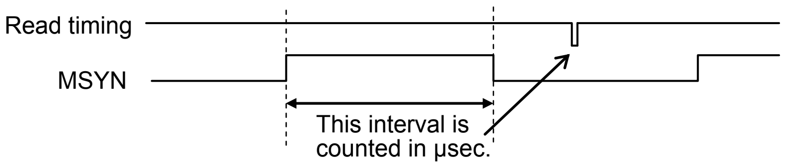
Bit	Bit name	Description
0	MCED	1: Masks CEND interrupt. (Status changes.)
1	MBRK	1: Masks BRKF interrupt. (Status changes.)
2	MIOP	1: Masks IOPC interrupt. (Status changes.)
3	MEIE	1: Masks EIOE interrupt. (Status changes.)
4	MEDE	1: Masks EDTE interrupt. (Status changes.)
5	MERE	1: Masks ERAE interrupt. (Status changes.)
6	MCSE	1: Masks CAER interrupt. (Status changes.)
7	—	Always "0".
8	BKOF	0: G9103A confirms a break request at a certain time intervals (250ms at 20Mbps.) 1: G9103A turns off automatic break function.
9	MCLR	Select the way to clear the following status bits. CEND, BRKF, EDTE, ERAE, CAER 0: These status bits are cleared by reading status. (dafault) 1: These status bits are not cleared by reading status. These status bits are cleared by clear command (04xxh) of INT status.
10 to 15	—	Always "0".

13-7-2 Error count register (RERCNT)

This is a 16-bit register for error counter. It is a read-only register.
This register counts total number of communication errors sucha as no response and CRC errors.
When error counts exceed more than 65534 times, the counter stops at 65535.
To clear the counter, issue a counter clear command (0600h).
Please note that it counts no response to system communication as an error. (No response to system communication does not handled as an error, basically.)
All bits are 0 after reset.

13-7-3 Cycle of cyclic communication register (RSYCNT)

This is a 16-bit register for measuring cycle of cyclic communication.
It counts the time when MSYN keeps change in μsec. It is a read-only register.
It always counts them, and you can refer the count before the last MSYN changes.
The upper limit of the counts is 65535 (approximate 65.5ms). The width of the MSYN signals more than the limit cannot be measured.



13-7-4 Recipient address register (RDJADD)

This register retains the device address of a local LSI that has received normal data communication finally.

This is read-only register.

All bits are 0 after reset.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0						

Device address of a local LSI that has received normal data communication finally.

13-7-5 Version inforamation register (RVER)

You can confirm the version information of G9001A. This is read-only register.

This register is always the following value (0001h).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

This register distinguish between G9001 and G9001A by software.

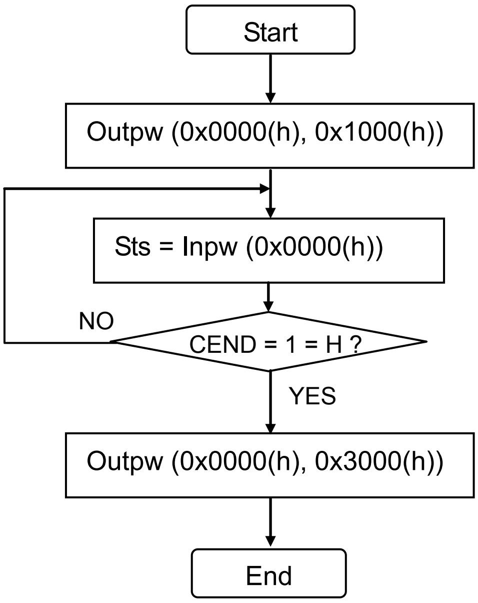
The process is as follows.

1. Write "0000h" to I/O buffer.
2. Issue Read commandd (6504h) of RVER register.
3. If data that is read from I/O buffer is "000h", G9001 is used. If "0001", G9001A is used.

13-8. Examples of control software

13-8-1. Start of the simplest cyclic communication

The simplest example is to issue a system communication command, let the center LSI automatically collect data from the local LSIs, and then start cyclic communication.



Send a system communication command to all local LSIs.

Read status

Wait for completion of the system communication
Until system communication is completed, the center LSI cannot start the next operation.
This step can also be used to wait for an interrupt.

Start a cyclic communication

13-8-2. Communication with port data

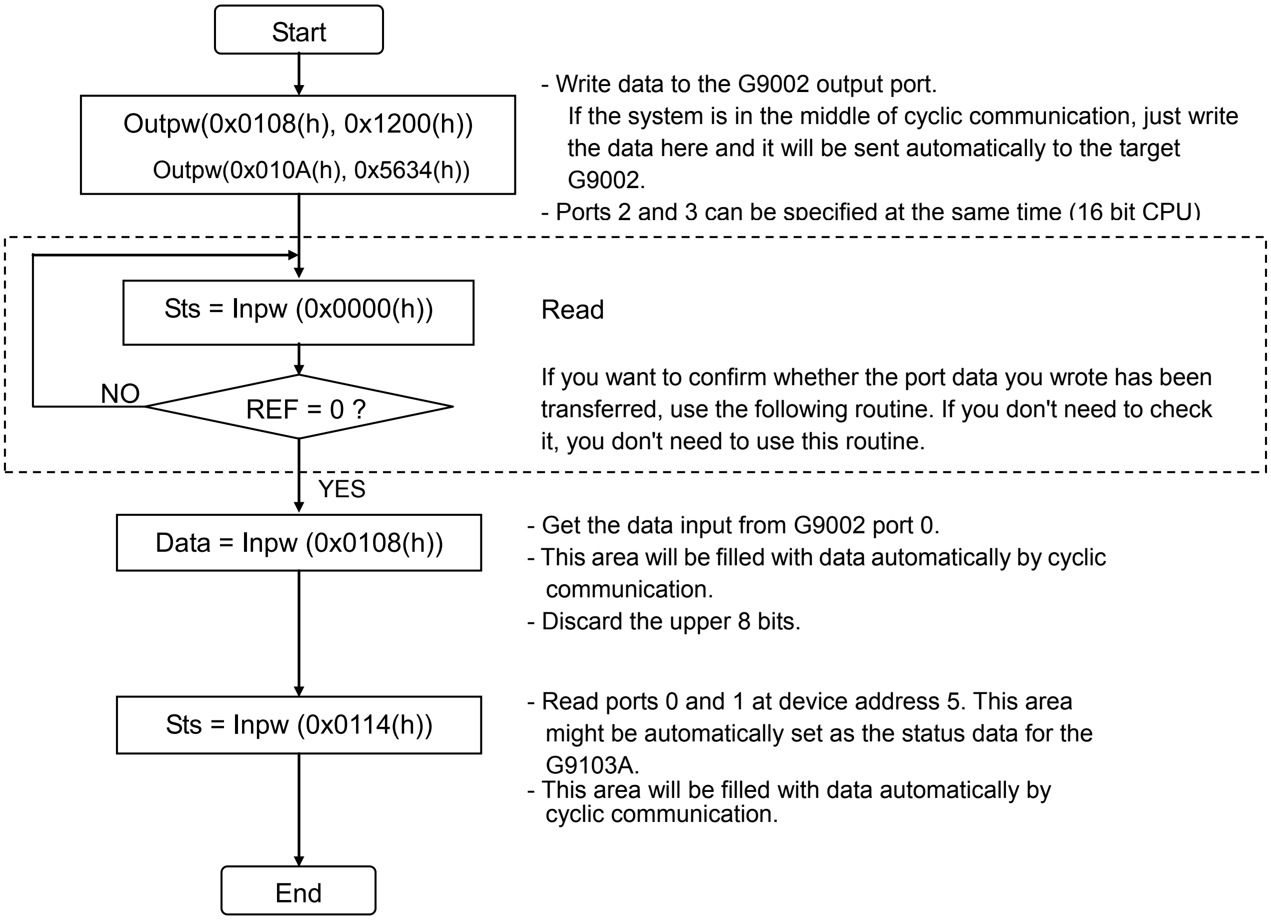
This section describes how to exchange data using the I/O port on the I/O device (G9002), and how to obtain the status of a data control LSI (G9103A).
Assume that the local LSIs to be used are as follows:
The example of the G9103A is one of only read status.

LSI type	Item to configure	Configuration data	Output data
I/O device (G9002)	Device address	2	-
	Port 0	Input	-
	Port 1	Output	12h
	Port 2	Output	34h
	Port 3	Output	56h
G9103A	Device address	5	

Note: The port area configuration of the G9103A is always as follows (always fixed).

Port No.	Mode	Description
Port 0	Input	Main status (MSTSB0) lower 8 bits
Port 1	Input	Main status (MSTSB1) upper 8 bits
Port 2	Input	Input value from the general-purpose I/O terminal (IOPIB)
Port 3	Output	Output value to the general-purpose I/O terminal (IOPIB)

1) When the whole address map can be used



- Write data to the G9002 output port.
  If the system is in the middle of cyclic communication, just write the data here and it will be sent automatically to the target G9002.
- Ports 2 and 3 can be specified at the same time (16 bit CPU)

Read

If you want to confirm whether the port data you wrote has been transferred, use the following routine. If you don't need to check it, you don't need to use this routine.

- Get the data input from G9002 port 0.
- This area will be filled with data automatically by cyclic communication.
- Discard the upper 8 bits.

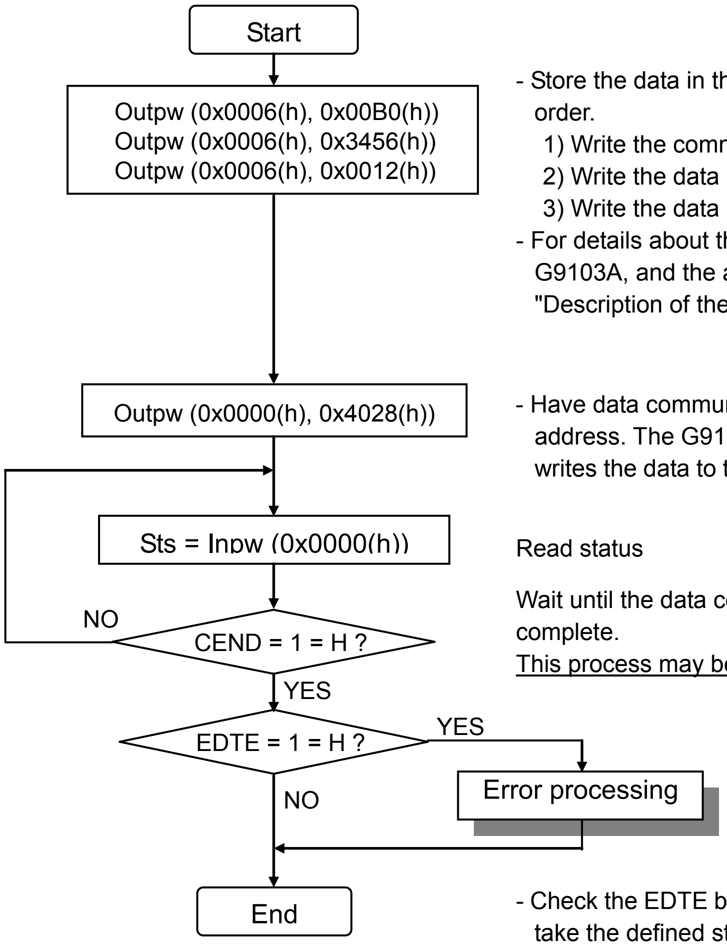
- Read ports 0 and 1 at device address 5. This area might be automatically set as the status data for the G9103A.
- This area will be filled with data automatically by cyclic communication.

13-8-3. Data communication

13-8-3-1. Set a value to a register of G9103A.

The data communication example below shows data being placed in a register that is integrated in the G9103A.
Assume that the local LSIs to be used are as follows.
Assume that "00123456h" will be placed in "PRMV" register of the G9103A.

LSI type	Configuration item	Device address
G9103A	Device address	40 (28h)


Start
Outpw (0x0006(h), 0x00B0(h))
Outpw (0x0006(h), 0x3456(h))
Outpw (0x0006(h), 0x0012(h))
Outpw (0x0000(h), 0x4028(h))
Sts = Inpw (0x0000(h))
NO
CEND = 1 = H ?
YES
EDTE = 1 = H ?
YES
Error processing
NO
End

- Store the data in the receiving FIFO in the following order.
  1) Write the command to the G9103A register.
  2) Write the data (lower 16 bits)
  3) Write the data (upper 16 bits)
- For details about the write order for the FIFO in G9103A, and the access command, see Chapter 5. "Description of the software."

- Have data communication with the specified device address. The G9103A that receives this communication writes the data to the register specified in the data details.

Read status

Wait until the data communication is complete.
This process may be waiting for an interrupt.

- Check the EDTE bit. If the data communication failed, take the defined steps.
- Note that the EDTE bit will be cleared by reading the status.
- The EDTE bit changes with the same timing as the CEND bit.

A data communication command is constructed as follows:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	0	0	0	0	0	0	0	#	#	#	#	#	#

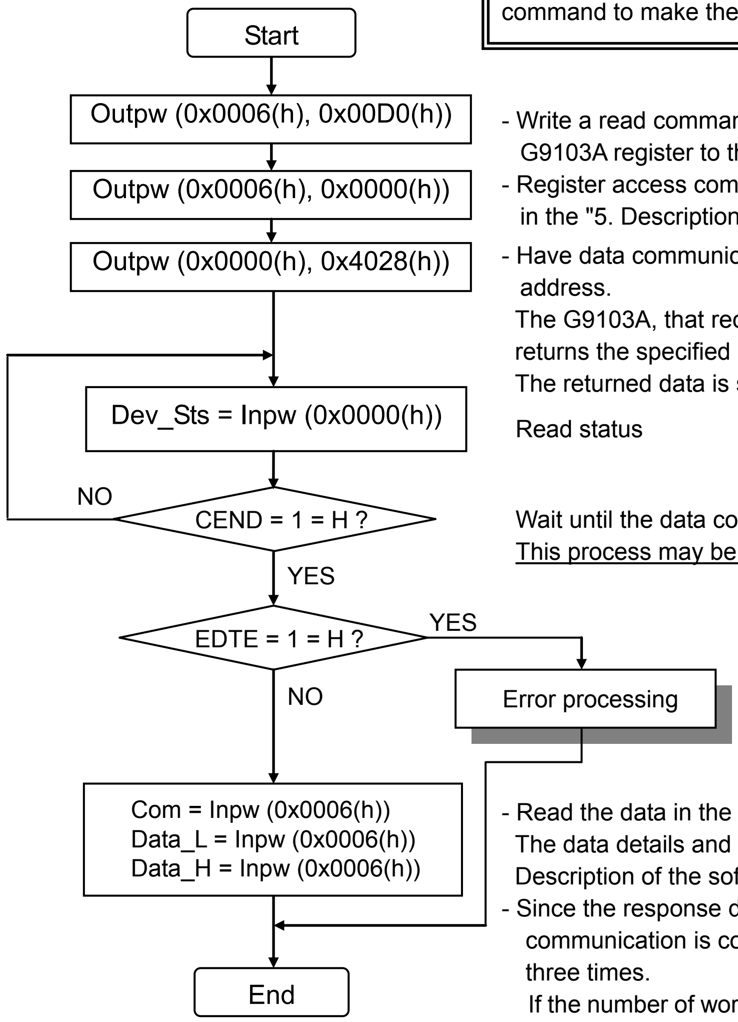
$\underbrace{\qquad\qquad\qquad\qquad}$ Specify the address in these bits.

13-8-3-2. Read registers of G9103A.

Assume that the local LSIs to be used are as follows.
Assume you want to read “RMV” register value in the G9103A.

LSI type	Configuration item	Configuration data
G9103A	Device address	40 (28h)

Note :
If FIFO write data has only one word, please add a NOP command to make the data two-word.



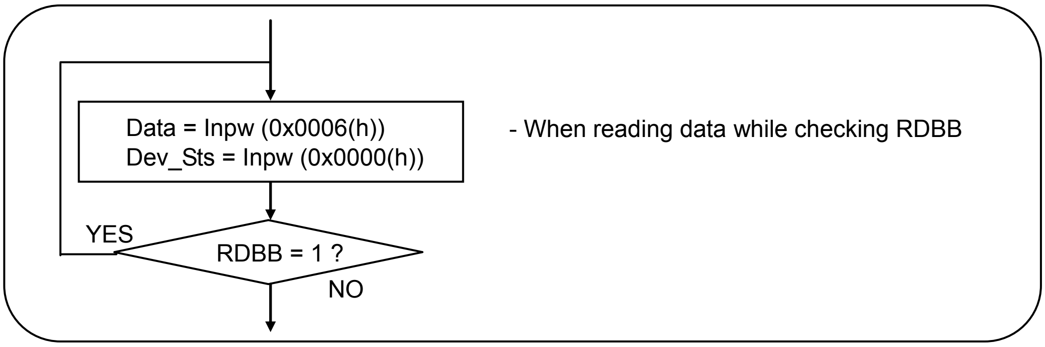
Start
Outpw (0x0006(h), 0x00D0(h))
Outpw (0x0006(h), 0x0000(h))
Outpw (0x0000(h), 0x4028(h))
Dev_Sts = Inpw (0x0000(h))
NO
CEND = 1 = H ?
YES
EDTE = 1 = H ?
YES
Error processing
NO
Com = Inpw (0x0006(h))
Data_L = Inpw (0x0006(h))
Data_H = Inpw (0x0006(h))
End

- Write a read command and a NOP command from G9103A register to the FIFO.
- Register access command of the G9103A is specified in the "5. Description of the software."
- Have data communication with the specified device address.
  The G9103A, that received this communication, returns the specified register data to the center LSI.
  The returned data is stored in the reception FIFO.

Read status

Wait until the data communication is complete.
This process may be waiting for an interrupt.

- Read the data in the reception FIFO.
  The data details and order are specified in "5. Description of the software."
- Since the response data is specified as 3 words, the communication is completed by reading the FIFO three times.
  If the number of words in the return data is not known, read the status in the center LSI. Keep reading the reception FIFO until "RDBB" bit goes LOW.

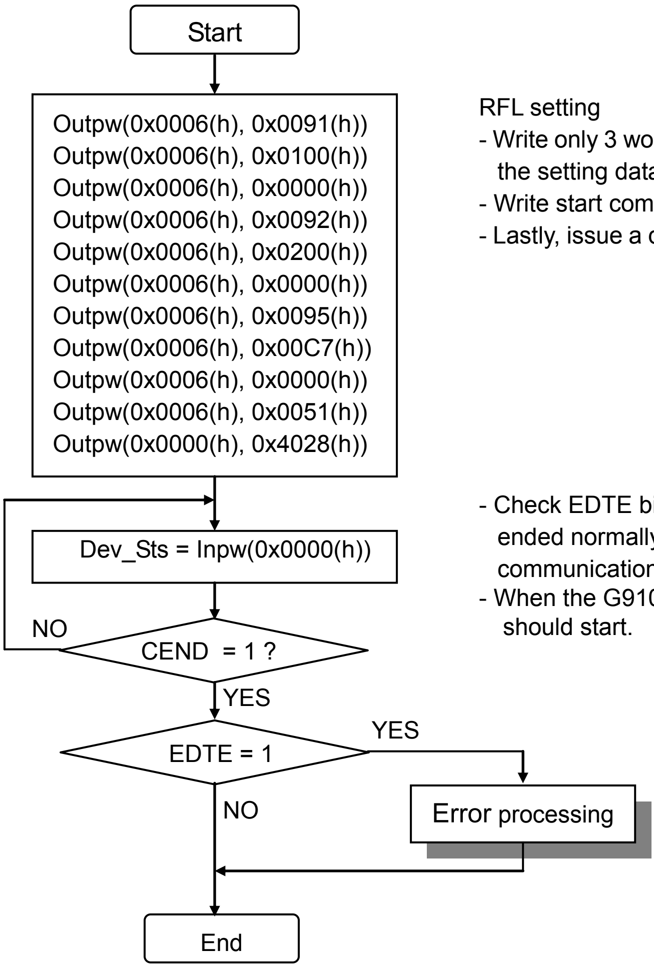


- When reading data while checking RDBB

13-8-3-3. Starting G9103A

As an example of data communication, the following shows an example to set various register of G9103A and start G9103A.
The data that is specified to G9103A is as follows. (This is a simple data to output pulses.)

Register name	Set value	Command	Remarks
RFL	00000100(h)	0091(h)	
RFH	00000200(h)	0092(h)	
RMG	00C7(h)	0095(h)	Multiplication rate = 1



```
Outpw(0x0006(h), 0x0091(h))
Outpw(0x0006(h), 0x0100(h))
Outpw(0x0006(h), 0x0000(h))
Outpw(0x0006(h), 0x0092(h))
Outpw(0x0006(h), 0x0200(h))
Outpw(0x0006(h), 0x0000(h))
Outpw(0x0006(h), 0x0095(h))
Outpw(0x0006(h), 0x00C7(h))
Outpw(0x0006(h), 0x0000(h))
Outpw(0x0006(h), 0x0051(h))
Outpw(0x0000(h), 0x4028(h))
```

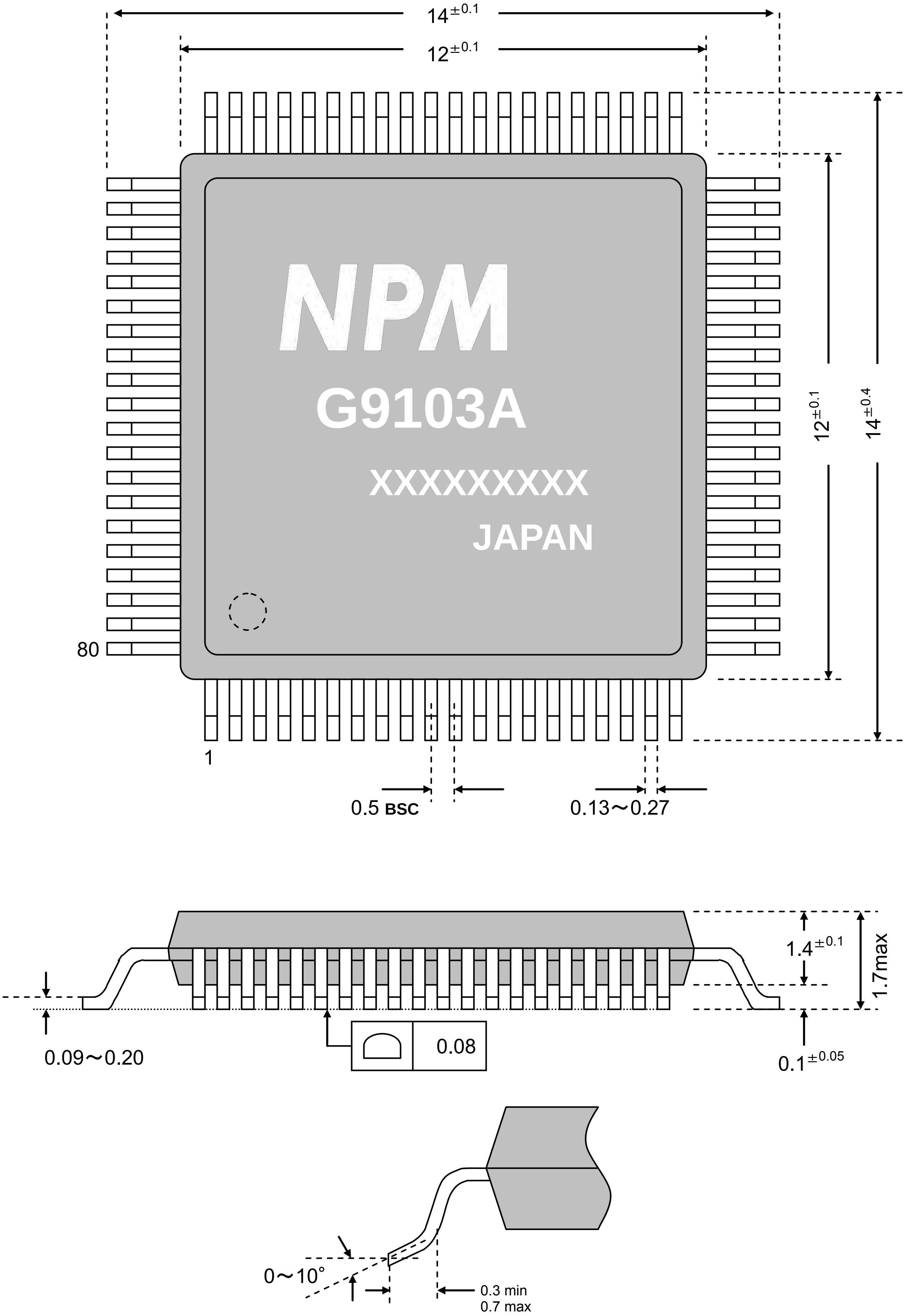
```
Dev_Sts = Inpw(0x0000(h))
```

RFL setting
- Write only 3 words amount of a register write command and the setting data (2 words) in the FIFO .
- Write start command(0x0051h) into the FIFO.
- Lastly, issue a device communication command (0x4028h).

- Check EDTE bit to see if this data communication ended normally before starting the next data communication.
- When the G9103A receives the data correctly, it should start.

14. External dimensions

80-pin, LQFP, Unit: mm



15. Designing Precautions

The following precautions are described above. The precautions that need attention especially are shown below again.

15-1. Precautions for hardware design

1) All signal terminals have 5V interface and can be connected to 3.3 V-CMOS, TTL, and LVTTL devices. However, even if the output terminals are pulled up to 5V, more than 3.3V is not realized.
   Input terminals are not equipped with an over voltage prevention diode for the 3.3 V lines. If over voltage may be applied due to a reflection, ringing, or to inductive noise, we recommend inserting a diode to protect against over voltage.

15-2. Precautions for software design

1) Please do not perform one-word communication that sends only one command without data.
   Please add a NOP command to the one word and perform at least two-word communication.
   If you perform one-word communication shortly after cyclic communication error occurs, G9103A may execute the command several times.

2) In the case that there are both G9103A and G9004A in the Motionnet system, please make length of data communication with G9004A less than 129 bytes.

3) When setting multiple registers at once in one communication, set three-words per register.

4) Please set operation complete timing to “End of Cycle (RMD.METM=0)” if the next operation starts automatically using the pre-register. If you select “End of Pulse (RMD.METM=1)”, the interval between the last pulse and the initial pulse of the next operation become narrow (750ns).

5) When interpolation operation block is executed continuously, the start timing between each block on each G9103A is not synchronized. Each G9103A controls the start timing on the assumption that operation time of each G9103A the same. Therefore, interpolation operation cannot be used with the functions that operation time is different among G9103As such as backlash correction, vibration restriction function, direction change timer, and delay by INP input.

6) Make sure to use the synchronization function of the clock for motor control in the system for interpolation operation. G9001A has two terminals SIA and SIB as serial input terminal and the communication line can be separated into two lines. However, G9103A of the interpolated axis should not be separate into two lines. Be sure to set RSYN of master LSI first. For other G9103As, any order is available.

7) When positioning operation synchronized with PA/PB or interporation operation (MOD=51h～55h,68h,69h,6Ch,6Dh,78h,79h,7Ch,7Dh) is complete, stop interrupt (MSTS.SEND) is not output. If interrupt is needed, please use normal stop of event interrupt (RIRQ.IREN)

15-3. Mechanical precautions

1) When a deceleration stop (ELM = 1) has been specified to occur when the EL input turns ON, the motor starts the deceleration when the EL input is turned ON. Therefore, the motor stops after the mechanical position passs over the EL position. In this case, be careful to avoid collisions of mechanical systems.

16. Handling Precautions

16-1. Design precautions

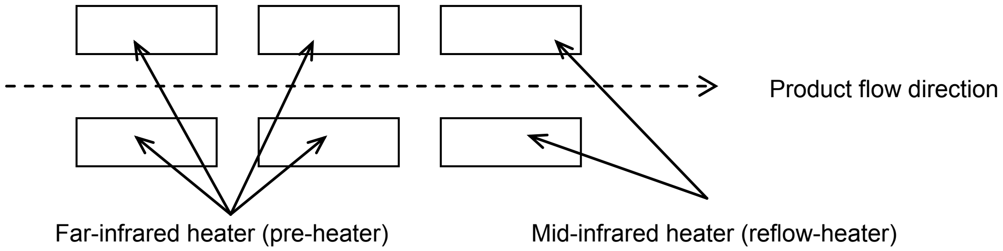
1) Never exceed the absolute maximum ratings, even for a very short time.
2) Take precautions against the influence of heat in the environment, and keep the temperature around the LSI as cool as possible.
3) Please note that ignoring the following may result in latching up and may cause overheating and smoke.
   - Do not apply a voltage greater than +5.5V to the input terminals and do not pull them below GND.
   - Make sure you consider the input timing when power is applied.
   - Be careful not to introduce external noise into the LSI.
   - Hold the unused input terminals to +3.3 V or GND level.
   - Do not short-circuit the outputs.
   - Protect the LSI from inductive pulses caused by electrical sources that generate large voltage surges, and take appropriate precautions against static electricity.
4) Provide external circuit protection components so that over voltages caused by noise, voltage surges, or static electricity are not fed to the LSI.
5) Using 5V interface, make the time difference between turning on and off 3.3 V and 5V power supply as short as possible. (There is no problem about the time difference of turning on and off the power supply such as to generate 3.3V power supply from 5 V power by a regulator IC or DC-DC converter.)

16-2. Precautions for transporting and storing LSIs

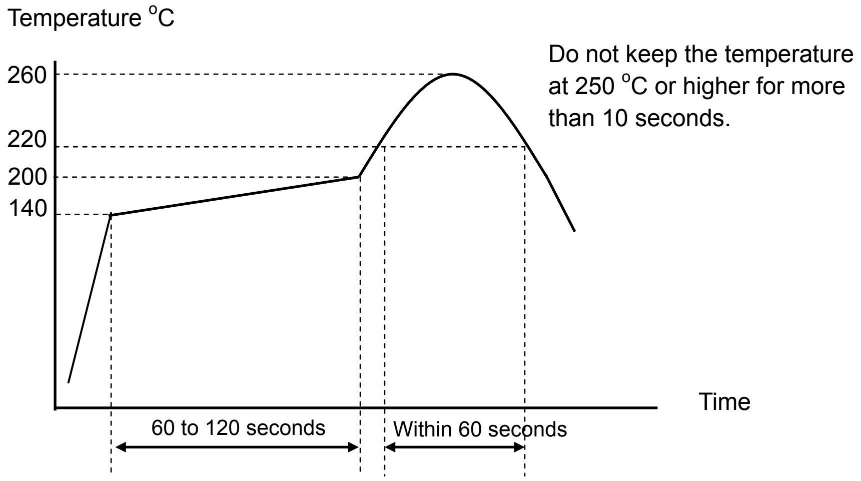
1) Always handle LSIs and their packages carefully. Throwing or dropping LSIs may damage them.
2) Do not store LSIs in a location exposed to water droplets or direct sunlight.
3) Do not store the LSI in a location where corrosive gases are present, or in excessively dusty environments.
4) Store the LSIs in an anti-static storage container, and make sure that no physical load is placed on the LSIs.

16-3. Precautions for mounting

1) In order to prevent damage caused by static electricity, pay attention to the following.
   - Make sure to ground all equipment, tools, and jigs that are present at the work site.
   - Ground the work desk surface using a conductive mat or similar apparatus (with an appropriate resistance factor). Do not allow to work on a metal surface, which can cause a rapid discharge and damage on LSI (if the charged LSI touches the surface directly) due to extremely low resistance.
   - When using a vacuum pickup tool, provide anti-static protection using a conductive rubber for the pickup tip. Anything which contacts with the LSI lead terminals should have as high a resistance as possible.
   - When using a pincer that may make contact with the LSI terminals, use an anti-static model. Do not use a metal pincer, if possible.
   - Store unused LSIs in a PC board storage box that is protected against static electricity, and make sure there is adequate clearance between the LSIs. Never directly stack them on each other, as it may cause friction that can develop an electrical discharge.
2) Operators must wear wrist straps that are grounded through approximately 1M-ohm of resistance.
3) Use low voltage soldering devices and make sure the tips are grounded.
4) Do not store or use LSIs, or a container filled with LSIs, near high-voltage electrical fields, such those produced by a CRT.
5) To preheat LSIs for soldering, we recommend keeping them at a high temperature in a completely dry environment, i.e. 125±5^{o}C for 20 to 36 hours. The LSI must not be exposed to the high temperature and completely dry environment more than 2 times.
6) When using an infrared reflow system to apply solder, we recommend the use of a far-infrared pre-heater and mid-infrared reflow devices, in order to ease the thermal stress on the LSIs. Do not reflow more than 2 times.


Product flow direction
Far-infrared heater (pre-heater)
Mid-infrared heater (reflow-heater)

Package and board surface temperatures must never exceed 260°C and do not keep the temperature at 250 °C or higher for more than 10 seconds.



[Recommended temperature profile of a far/mid-infrared heater and hot air reflow]

7) When using hot air for solder reflow, the restrictions are the same as for infrared reflow equipment.
8) If you will use a slodering iron, the temperature at the leads must not exceed 350 degrees or higher and the time must not exceed for more than 5 seconds and more than twice per each terminal.

16-4. Other precautions

1) When the LSI will be used in poor environments (high humidity, corrosive gases, or excessive amounts of dust), we recommend applying a moisture prevention coating.
2) The package resin is made of fire-retardant material; however, it can burn. When baked or burned, it may generate gases or fire. Do not use it near ignition sources or flammable objects.
3) This LSI is designed for use in commercial apparatus (office machines, communication equipment, measuring equipment, and household appliances). If you use it in any device that may require high quality and reliability, or where faults or malfunctions may directly affect human survival or injure humans, such as in nuclear power control devices, aviation devices or spacecraft, traffic signals, fire control, or various types of safety devices, we will not be liable for any problem that occurs, even if it was directly caused by the LSI. Customers must provide their own safety measures to ensure appropriate performance in all circumstances.

Notes

* The specifications may be changed without notice for improvement.

NPM Nippon Pulse Motor Co., Ltd.
Head Office: No.16-13, 2-chome, Hongo, Bunkyo-ku, Tokyo 113-0033, Japan
Phone: +81-3-3813-8841 Fax: +81-3-3813-8665
Web: http://www.pulsemotor.com E-mail: int-l@npm.co.jp