# Motionnet®

Remote I/O & Remote Motion

# G9103C

(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]

The Motionnet® system is a wire-saving system that consists of one center LSI connected to a CPU bus, and maximum 64 local LSIs, and is connected by using cables of two or three conductive cores.

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

## [Cautions]

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

(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 the above.

■ Descriptions of indicators that are used in this manual

1. In description of register bits, "n" refers to a bit position. "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".

2. Specified bit of register is referred to as “resister name”.“bit name”. (ex. RMD.MSDE)

3. This manual does not use overlines with negative logic terminals and signals. Please refer to “4.1 A list of terminals”.

4. Unless otherwise described, time description in this manual is in the case of CLK=40 MHz.

5. Signal “ON” or “OFF” is shown by “H level” or “1” in the case of positive logic and “L level” or “0” in the case of negative logic.

6. Suffix “b” with numerical number shows binary and suffix “h” shows hex. Decimal notation is used no suffix. Even though binary or hex notation is used, suffix may not be attached in some figures or in the case of that the same as decimal notation.

Table of Contents

## *1. Outline*

This LSI is an axis motor control local LSI with interpolation function 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. Using communications, the system allows you to check operation status and output interrupt request signals with various conditions. If all LSIs connected to the center LSI are the G9103Cs, the system can be constructed to control up to 64 axes while saving wiring.
This system for G9103C has high degree of compatibility with G9003, a conventional local LSI for single axis control without interpolation function. Therefore, G9103C 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 G9103C can stop outputting command pulses and reset general-purpose output terminals.

- Power supply
  Single power supply voltage: +3.3 V.
  G9103C can interface with TTL-ICs because input terminals are 5 V tolerant.

- Synchronizing of the clock for motor control
  When G9103Cs are dispersed, quartz crystal oscillators are needed individually. In this case, the frequency error causes to make errors of the speed of each axis, and affects interpolation operation. G9103C has synchronizing circuit of the clock for motor control. Using this circuit, clock frequency for motor control is adjusted between more than one G9103C and operation is synchronized. 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 signal can be output from the center LSI by various causes.

- 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. Using this S-curve range setting enables S-curve deceleration after linear acceleration and linear deceleration after S-curve acceleration.

- 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 a motor is operating in positioning mode. If the current position exceeds a newly entered position, the motor 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 G9103C receives an external signal, it stops outputting after outputting a specified number of pulses.

- Pre-register function
  The pre-register for operation data is built in (one layer). During operation, the center LSI can send a next operation data and G9103C 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. G9103C can write into or read from up to 21 registers in one communication. When multiple registers are written or read out, total communication time can be shorter.

- Broadcast communication function
  Commands for start, stop, acceleration, and deceleration can be sent from a center LSI to G9103C of the specified groups (1 ~ 7) or all groups simultaneously. The groups are specified by terminals or software.

- Unit ID control function
  G9103C, a 32-bit register RGN0, can read default value from an external EEPROM to distinguish a kind of product (unit).
  The 32 bits consist of company ID (upper 20 bits) and model ID (lower 12 bits). Company ID is controlled by NPM and model ID is controlled by customers.

- CPU connection function
  G9103C has three 32-bit general-purpose registers (RGN1 ~ RGN3) that can be accessed from an external CPU through 4-line serial communication so that parameters for driver, such as gain adjustment, can be set through **Motionnet**® when G9103C is mounted in a motor driver.

- FH correction function (Triangle pattern drive elimination)
  In the positioning mode, when number of command pulses is not enough to accelerate, this function automatically lowers the maximum speed (FH speed) and eliminates triangle driving.

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

- Simultaneous stop function
  Multiple axes can be stopped at the same time using broadcast communication or an external signal. Additionally, when other axis stops by errors, the G9103C can also stop its axis.

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

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

| Counter | Intended use | Count Input |
|---|---|---|
| COUNTER 1 | 28-bit counter for control of a command position | Command pulses |
| COUNTER 2 | 28-bit counter for mechanical position control<br>(It can be used as a general-purpose counter) | Command pulses<br>EA and EB signals<br>PA and PB signals |
| COUNTER 3 | 16-bit counter for controlling deviation between a command position and a machine's current position, or 16-bit general-purpose counter with synchronizing signal output function. | Command pulses<br>EA and EB signals<br>PA and PB signals<br>1/4096 division clock of 40 MHz<br>Command pulses and EA and EB signals<br>Command pulses and PA and PB signals<br>EA, EB signals and PA, PB signals |

  All counters can be cleared 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, COUNTER 2 and COUNTER 3.

  Comparators 1 and 2 can also be used as software limits (+SL, −SL signal).

- Software limit function

  You can set software limits using 2 comparator circuits.

  When the mechanical position arrives in the software limit range, a motor stops immediately or decelerates and stops. Then, the machine can only be moved in the direction opposite to its previous travel.

- Backlash correction function

  The G9103C has a backlash correction function.

  Backlash correction corrects a feed amount each time the feed direction is changed.

- Synchronizing signal output function

  The G9103C can output pulse at the specified constant interval.

- Vibration restriction function

  When you specify a control constant in advance, G9103C adds one pulse for reverse feed and one for forward feed just before a motor stops.

  This function can decrease vibration that occurs when a stepper motor stops.

- Manual pulsar signal input function

  By applying manual pulse signals (PA, PB), you can operate a motor directly.

  The input signals can be 90 degree phase difference signals (1x, 2x, or 4x) or Two-pulse signals (count forward and backward pulse).

  In addition to magnification above, the LSI contains an integral pulse multiplier that multiplies by 1x ~ 32x and a pulse divider (1 ~ 2048) / 2048.

  Stop function by +EL or –EL signals and software limit setting is enabled. When the LSI stops outputting pulses by this function, the motor can feed only in the opposite direction.

- Stepper motor out-of-step detection function

  This LSI has a deviation counter which can be operated by command pulses and encoder signals (EA, EB signal).

  It can be used to detect out-of-step operations of stepper motor and to confirm a position by using a comparator.

- Output pulse mode

  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 current down signals automatically for reducing excitation current while operation is stopping. At the start, G9103C restores a changed current value to the original and begins to rotate the shaft after a specified time has elapsed. The G9103C can make current value in the condition of the current down after the operation has stopped and a specified time has elapsed.

- Idling pulse output function

  The G9103C can output a specified number of pulses at the initial speed (FL speed) before a high-speed start acceleration operation.

  Even if value near to the maximum speed 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 operation modes>

1) Start and stop of current operation by command.
2) Continuous operation, positioning operation, linear interpolation and circular interpolation operation using PA or PB signal).
3) Origin return operation.
4) Positioning operation using commands.
5) Hardware start of positioning operation using STA signal.
6) Specified amount feed after PCS signal turning ON. (Override 2 for target position)
7) Linear interpolation by multiple G9103Cs.
8) Circular interpolation by two G9103Cs.

- 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 or –EL signal is turned ON. (Normal stop)
5) The axis feeds at constant speed, reverses when the +EL or –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 or –EL signal, and stops when a specified number of EZ signals are received.

- Mechanical input signals (+EL, –EL, SD, ORG)

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, or decelerates and stops. When this signal is already ON at the start, no further movement occurs on the axis in the positive direction. (Reverse operation can available.)
2) −EL: This signal's function is the same as a +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 software setting. When deceleration is set, and when this signal is turned ON during a high speed feed operation, the motor will decelerate to the FL speed. If this signal is ON and movement on the axis is started, the motor will operate at the FL constant speed. When deceleration stop is set, and when this signal is turned ON during a high speed feed operation, the motor will decelerate to the FL speed and then stop. When this signal is already ON at the start, no further movement occurs.
4) ORG: Input signal for an origin return operation.

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

The input logic for +EL and −EL signals can be changed using the ELL terminal.

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

- Servomotor I/F (INP, ERC, ALM)

  The following three signals can be connected.

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

  The input logic of the INP, ERC, and ALM signals can be changed using software setting
  The ERC signal is a pulsed output. The pulse length can be set. (12 µs ~ 102 ms, a level output is also available.)

- Emergency stop signal (EMG)

  When this signal is turned ON, a motor stops immediately.
  While this signal is already ON at the start, no movement is allowed.

## 3. Specifications

### 3.1 Basic specifications

| Item | Description |
|---|---|
| General-purpose input/output | General-purpose input/output terminals (8 terminals)<br>Individual terminal can be set as input or output. |
| Communication data length | 1 ~ 64 words / frame (one word = 16 bits) |
| Data buffer length | 64 words |
| Data communication time | When using three-word communication (written into one register) --- 19.3 µs<br>63-word communication (written to 21 registers together)--- 91.3 µs |
| Communication control system | I/O communication: Cyclic communication control system<br>Data communication: Transient communication control system. |
| Attribute value | “0000 0000 0010 0011 1000 0001 ggg0 1111” (ggg = Group No.) |
| Package type | 80 pins QFP (Mold section: 12 x 12 x 1.4 mm) same as G9003 |
| Weight | 0.50 g Note 1 |
| Power supply | 3.0 V ~ 3.6 V |
| Storage temperature range | −40 ~ +125 °C |
| Operating temperature range | −40 ~ +85 °C |

Note 1: Individual variability may be observed.

### 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 duty ratio of CLK signal can be maintained to ideal condition (50%), G9103C can be operated by 40 MHz CLK signals.
The ideal condition is that each G9103C is connected to one oscillator closely.
(Actually, even this good condition cannot establish 50% duty ratio. However, a duty ratio 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 small, this system can operate without any troubles.
(For details, see “4.4.1 CLK [74]”.)
When signal lines are longer and/or many local LSIs are connected and it is difficult to warrant an ideal clock duty, you should input 80 MHz CLK signals.
Which clock rate you use is selected by setting CKSL terminal. In either clock rate, the maximum communication rate is 20 Mbps.

Note 2: The communication speed is selected by the SPD0 and SPD1 terminals. It is not necessary to change a clock rate (CKSL terminal) even if changing communication speed.

Note 3: NPM recommends a system using a pulse transformer.

## *3.3 Specifications for the axis control section*

| Item | Description |
|---|---|
| Positioning control range | −134,217,728 ~ +134,217,727 (28 bits) |
| Ramping-down point setting range | 0 ~ 16,777,215 (24 bits) |
| Number of registers used for setting speeds | Three for each axis (FL, FH, and FA (correction speed)) |
| Speed setting step range | 1 ~ 100,000 (17 bits) Note1 |
| Speed magnification range | Multiply by 0.1 ~ 66.6<br>Multiply by 0.1 = 0.1 ~ 10,000.0 pps<br>Multiply by 1 = 1 ~ 100,000 pps<br>Multiply by 50 = 50 ~ 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 ~ 65,535 (16 bits)<br>Ex: Acceleration time (from 1 pps ~ 100,000 pps) :<br>Approximate 80 ms when “1” is set.<br>Approximate 2,621 s when “65,535” is set. |
| Deceleration rate setting range | 1 ~ 65,535 (16 bits) |
| 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 operation speed (FH speed) so that the triangle speed pattern will be avoided. |
| Manual operation input | Input by a manual pulse generator |
| Counters | COUNTER 1: Command position counter (28 bits)<br>COUNTER 2: Mechanical position counter (28 bits)<br>COUNTER 3: General-purpose deviation counter (16 bits) |
| 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 be 100,000.

## 4. Hardware description

### 4.1 A list of terminals

| No. | Signal name | I/O | Logic | Description |
|---|---|---|---|---|
| 1 | VDD | - | - | +3.3 V |
| 2 | DN0/DNSI | $I_U$ | Negative | Device number setting bit 0 / Serial input of device number |
| 3 | DN1 | $I_U$ | Negative | Device number setting bit 1 |
| 4 | DN2 | $I_U$ | Negative | Device number setting bit 2 |
| 5 | DN3/ROMC | $B_U$ | Negative | Device number setting bit 3 / Clock output for EEPROM connection |
| 6 | DN4/ROMO | $B_U$ | Negative | Device number setting bit 4 / Data output for EEPROM connection |
| 7 | DN5/ROMI | $I_U$ | Negative | Device number setting bit 5 / Data input for EEPROM connection |
| 8 | VDD | - | - | +3.3 V |
| 9 | GRP0 | $I_U$ | Negative | Group number setting bit 0 (VDD on the G9003) |
| 10 | GRP1 | $I_U$ | Negative | Group number setting bit 1 (VDD on the G9003) |
| 11 | GRP2 | $I_U$ | Negative | Group number setting bit 2 (VDD on the G9003) |
| 12 | ROME | $I_D$ | Positive | H: EEPROM connection mode (GND on the G9003) |
| 13 | DNSO/ROMS | O | Negative | Serial output of next LSI device number / Chip select output for EEPROM connection |
| 14 | DNSM | $I_U$ | - | L : Device number automatic setting mode |
| 15 | SOEI | $I_D$ | Positive | Enable serial output (input terminal for multi-layer connection) |
| 16 | SOEL | O | Negative | Enable serial output (Negative logic output terminal) |
| 17 | SOEH | O | Positive | Enable serial output (Positive logic output terminal) |
| 18 | SO | O | Positive | Serial output |
| 19 | GND | - | - | GND |
| 20 | SI | I | Positive | Serial input (Schmitt trigger) |
| 21 | GND | - | - | GND |
| 22 | VDD | - | - | +3.3 V |
| 23 | VDD | - | - | +3.3 V |
| 24 | EA | $I_U$ | - | Encoder phase A signal |
| 25 | EB | $I_U$ | - | Encode phase B signal |
| 26 | EZ | $I_U$ | Negative# | Encoder phase Z signal |
| 27 | PA | $I_U$ | - | Manual pulse generator phase A signal |
| 28 | PB | $I_U$ | - | Manual pulse generator phase B signal |
| 29 | GND | - | - | GND |
| 30 | ERC/CDWN | O | Negative# | Request to clear a deviation counter in a driver /current-down |
| 31 | OUT | O | Negative# | Pulse train output |
| 32 | DIR | O | - | Operation direction output |
| 33 | VDD | - | - | +3.3 V |
| 34 | CP1 | O | Negative | Comparator 1 output |
| 35 | CP2 | O | Negative | Comparator 2 output |
| 36 | CP3 | O | Negative | Comparator 3 output |
| 37 | PCS | $I_U$ | Negative# | Start positioning control |
| 38 | LTC | $I_U$ | Negative# | Counter value latch signal |
| 39 | CLR | $I_U$ | Negative# | Counter clear signal |
| 40 | INP | $I_U$ | Negative# | In-position (Positioning complete) |
| 41 | GND | - | - | GND |
| 42 | EMG | $I_U$ | Negative | Emergency stop |
| 43 | +EL | $I_U$ | Negative% | (+) end limit |

| No. | Signal name | I/O | Logic | Description |
|---|---|---|---|---|
| 44 | −EL | $I_U$ | Negative% | (−) end limit |
| 45 | SD | $I_U$ | Negative# | Deceleration (Deceleration stop) signal |
| 46 | ORG | $I_U$ | Negative# | Origin position signal |
| 47 | ALM | $I_U$ | Negative# | Alarm signal (Stop request) |
| 48 | VDD | - | - | +3.3 V |
| 49 | STA | $B_U$ | Negative | External start, simultaneous start |
| 50 | STP | $B_U$ | Negative | External stop, simultaneous stop |
| 51 | P0 | $B_U$ | - | General-purpose I/O terminal 0 |
| 52 | GND | - | - | GND |
| 53 | P1 | $B_U$ | - | General-purpose I/O terminal 1 |
| 54 | P2 | $B_U$ | - | General-purpose I/O terminal 2 |
| 55 | P3 | $B_U$ | - | General-purpose I/O terminal 3 |
| 56 | P4/SIFC | $B_U$ | - | General-purpose I/O terminal 4 / Serial clock input for CPU connection |
| 57 | P5/SIFS | $B_U$ | - | General-purpose I/O terminal 5 / Slave selection input for CPU connection |
| 58 | P6/SIFI | $B_U$ | - | General-purpose I/O terminal 6 / Data input at slave side for CPU connection |
| 59 | P7/SIFO | $B_U$ | - | General-purpose I/O terminal 7 / Data output at slave side for CPU connection |
| 60 | VDD | - | - | +3.3 V |
| 61 | BSY/PH1 | O | Negative/ Positive | Output operating signal / Excitation sequence output phase 1 |
| 62 | FUP/PH2 | O | Negative/ Positive | Output accelerating signal / Excitation sequence output phase 2 |
| 63 | FDW/PH3 | O | Negative/ Positive | Output decelerating signal /Excitation sequence output phase 3 |
| 64 | MVC/PH4 | O | Negative/ Positive | Output constant speed operating /Excitation sequence output phase 4 |
| 65 | ELL | $I_U$ | - | +EL and –EL input logic setting |
| 66 | GND | - | - | GND |
| 67 | MSEL | O | Negative | Goes L level for a certain interval while this LSI is sending/receiving data. |
| 68 | MRER | O | Negative | Goes L level for a certain interval when an abnormal communication has been received. |
| 69 | TOUT | O | Negative | Watchdog timer output |
| 70 | BRK | $I_D$ | Positive | Break signal |
| 71 | TUD | $I_U$ | - | Select operation method for outputting watchdog timer signal |
| 72 | TMD | $I_U$ | - | Watchdog timer setting |
| 73 | VDD | - | - | +3.3 V |
| 74 | CLK | I | - | Reference clock (Schmitt trigger) |
| 75 | GND | - | - | GND |
| 76 | CKSL | $I_U$ | - | Clock rate selection |
| 77 | SPD0 | $I_U$ | - | Communication rate setting 0 |
| 78 | SPD1 | $I_U$ | - | Communication rate setting 1 |
| 79 | GND | - | - | GND |
| 80 | RST | $I_U$ | Negative | Reset |

Note 1: In the I/O column, “I”, “O” and “B” express input, output and both direction, respectively.

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

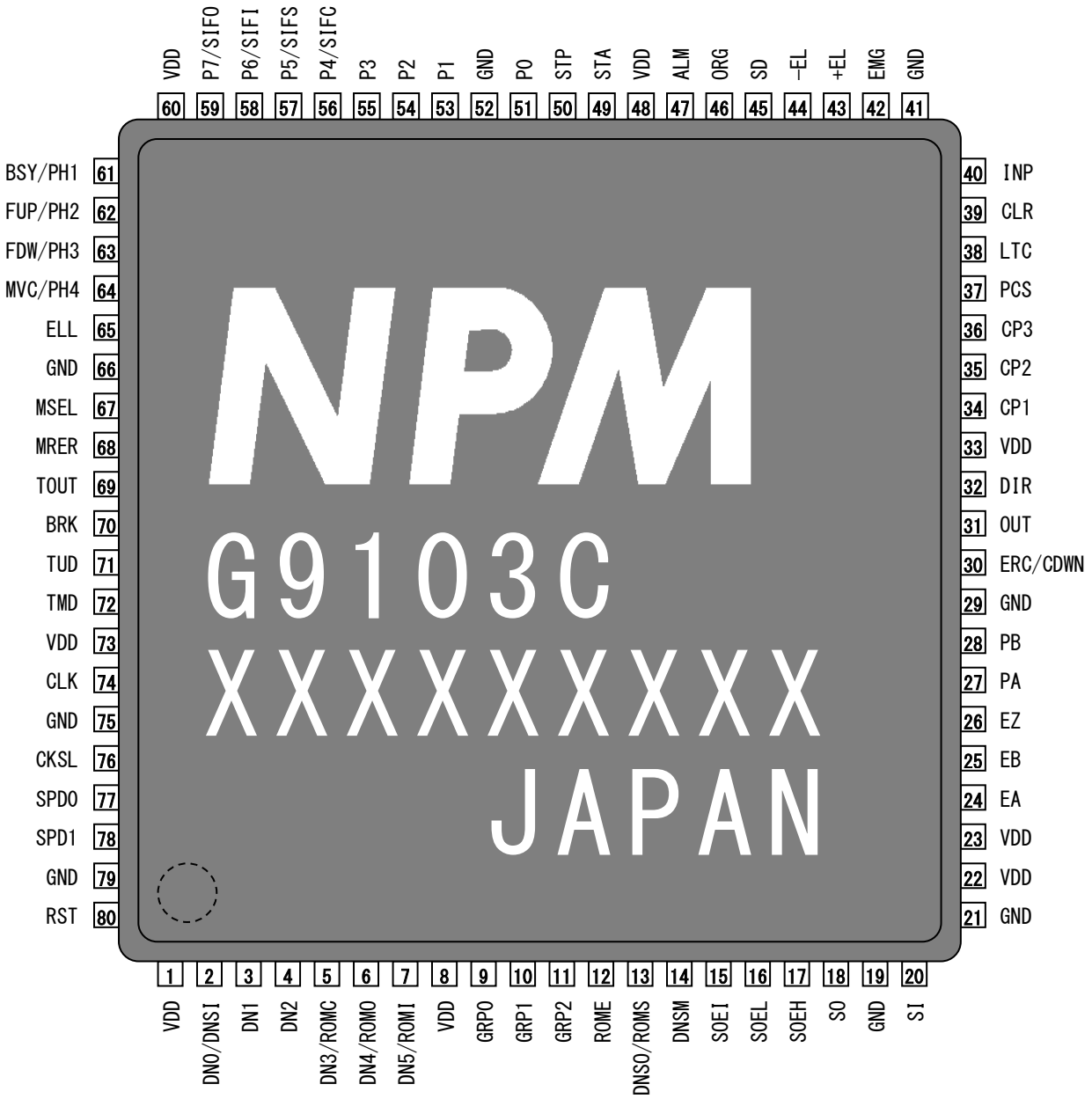
Note 3: 5 V interface inputs are not equipped with an over voltage prevention diode for the 3.3 V lines. If over voltage may be applied due to a reflection, ringing, or to inductive noise, we recommend inserting a diode to protect against over voltage.

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 terminal and bidirectional terminals that are not used and which have an internal pull up or pull down resistor can be left open. However, we recommend pulling these unused terminals up to 3.3 V (5k ~ 10k ohm) externally or pulling down to GND. Input terminals can be connected to 3.3 V or GND without an external resistor. However, terminals for both directions should be connected using a resistor.

Note 5: Leave unused output terminals open.

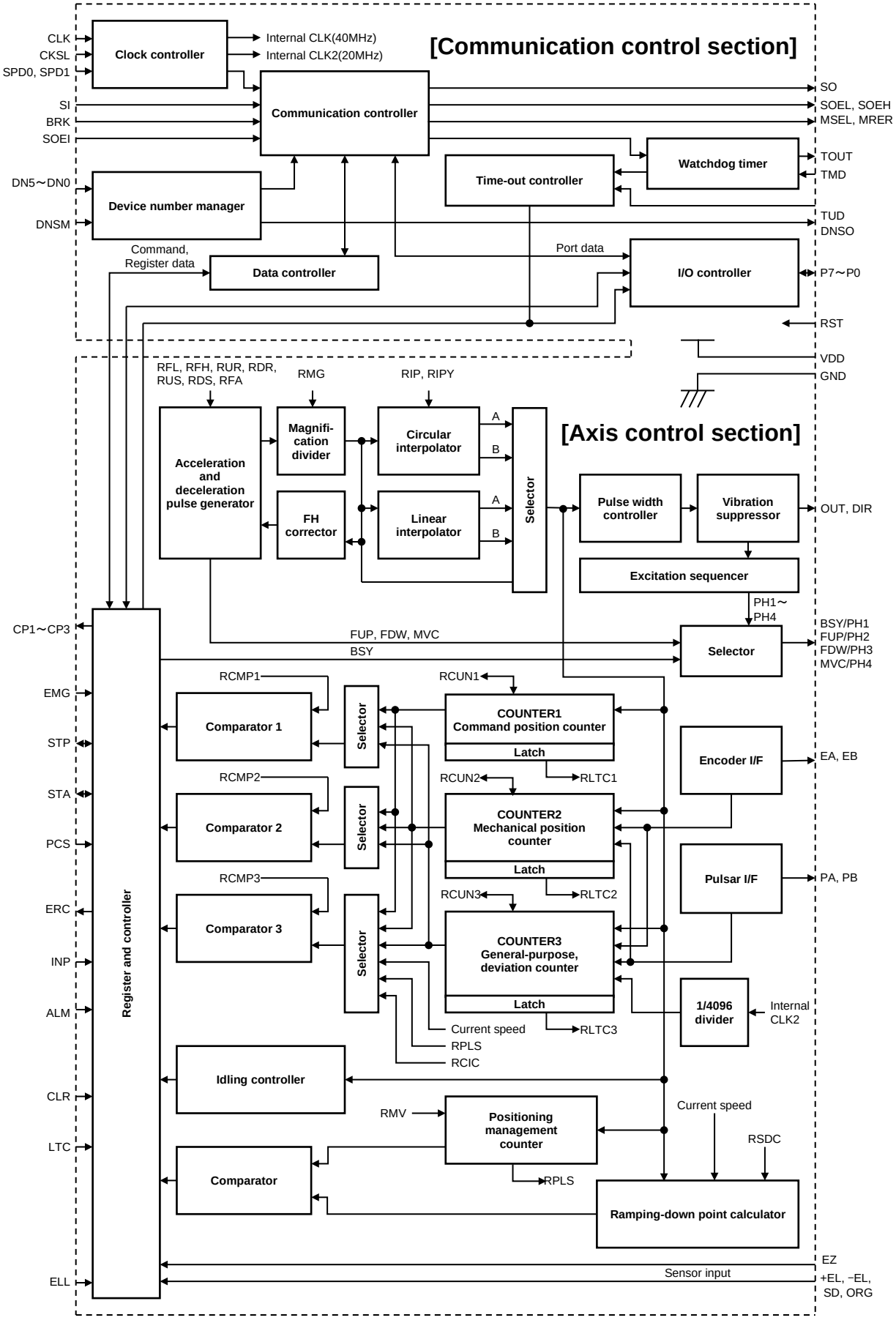
Note 6: Positive" means positive logic and "Negative" means negative logic. In addition, "#" means that the terminal’s logic can be changed by software. "%" means that the terminal’s logic can be changed by terminal setting. The logic shown in the table is a default condition. DIR terminal logic shown 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 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. Input specification is Schmitt trigger. For details, see “10.3 DC characteristics”. By setting CKSL terminal, either of the following clock rate signals can be connected.

CKSL = L level: Please connect 40 MHz with CLK terminal.
CKSL = H level: Please connect 80 MHz with CLK terminal.

By selecting either of these clock rates, serial communication rate does not change.
A reason to select high clock rate is to improve communication quality.
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.
However, a 20 Mbps communication rate and a longer communication line or a large number of connected local LSIs may deteriorate communication quality due to collapse of signals in communication line. This deterioration of communication quality can be corrected inside the LSI if the deterioration is in a certain level. However, in order to improve correction precision, evenness of the duty ration of CLK signal is required.
If duty ratio is close to ideal (50%), the capacity to correct collapse of signals in the communication lines can be improved. On the contrary, if duty is not ideal, the LSI cannot cope with collapses of communication line. As a result, if a duty ratio is close to ideal, the LSI can be used with 40 MHz. When connecting more than one LSI to one crystal oscillator, the duty ratio may not be ideal. In this case, use 80 MHz. This LSI divides a clock rate inside and creates 40 MHz frequency.
If you do not use 80 MHz clock rate, you can prepare a separate 40 MHz crystal oscillator for each G9103C. We recommend that 3.3 V signal is input because duty rate may get worse due to threshold voltage, even though 5 V signal can be input.

### 4.4.2 RST [80]

This is an input terminal for a reset signal.
While RST = L level, the internal circuit of G9103C is reset.
During reset, input longer than 10 clock cycles of CLK signal.
When CKSL=H level, please input longer than 20 clock cycles of CLK signal.
After the power is turned on, reset signals should be input before G9103C is used.
This terminal has a built-in pull up resistor to prevent floating.

### 4.4.3 CKSL [76]

This is used to select clock rate.
CKSL = L level: Connect 40 MHz clock rate to CLK terminal.
CKSL = H level: Connect 80 MHz clock rate to CLK terminal.

Select 80 MHz CLK signal when a duty ratio of 40 MHz clock collapses a lot.
This terminal has a built-in pull up resistors to prevent floating.
When using H level, please pull up to VDD (5k ~ 10k ohm) or connect directly to improve noise resistance.

### 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 GND terminal. For details, see “8.18 Unit ID control function”.
Though this terminal has a built-in pull up resistor to prevent floating, when using H level, pull up to VDD (5K ~ 10K ohm) or connect directly to improve noise resistance.

### 4.4.5 DN0 / DNSI [2], DN1 [3], DN2 [4], DN3 / ROMC [5], DN4 / ROMO [6], DN5 / ROMI [7]

Input terminals to set device number.
Since these terminals use negative logic, setting all the terminals to L level calls up device number "3Fh."
There are two methods for entering a device number. Select input method using DNSM terminal.
DN3 ~ DN5 are also used as terminals for EEPROM connection. For details, see “8.18 Unit ID control function”. When using H level, pull up to VDD (5k ~ 10k ohm) or connect directly to improve noise resistance.

#### 4.4.6 DNSM [14]

This terminal is to select input method of device numbers.
1) When DNSM = H level
   Specify an address from 00h ~ 3Fh using DN0 ~ DN5 terminals.

2) When DNSM = L level
   When DNSO signal that is output by another local device is input to DN0 / DNSI terminal on this LSI, this LSI has a device number equal to another LSI's device number +1.
   DN1 ~ DN5 terminals should be pulled up or pulled down.
   DN0 / DNSI terminal is input serialized device number value repeatedly.
   When same values are input twice in a row, this value is recognized as a correct device number.

This terminal has a built-in pull up resistor to prevent floating.
When using H level, pull up to VDD (5k ~ 10k ohm) or connect directly to improve noise resistance.

#### 4.4.7 DNSO / ROMS [13]

This terminal outputs a numeric equivalent this LSI's device number + 1 to the next LSI and the LSI outputs the numeric equivalent to the device number +1 to the next LSI continuously in order.
Connect this output to another local LSI's DN0 / DNSI terminal, so that another LSI can get a continuous device number. Please set DNSM terminal of other local LSI to L level.
The next device number after "3Fh" (DN5 ~ DN0 = "LLLLLL") is "00h."
If you set consecutive device number by DNSO terminal, it needs at least 50 µs to determine a next device number.
When ROME terminal is H level, it is used as the terminals for EEPROM connection. Therefore, when serial setting of multiple local LSIs' device numbers, please make G9103C connected with EEPROM the last LSI of the communication line.

#### 4.4.8 GRP0 ~ GRP2 [9 ~ 11]

This terminal is to specify a group number used for broadcast communication.
Because of negative logic input, "7" is set as a group number when all terminals are set to L level. A group number can be changed by software (RENV2.GN).
This terminal has a built-in pull up resistor to prevent floating
When using H level, pull up to VDD (5k ~ 10k ohm) or connect directly to improve noise resistance.

#### 4.4.9 SPD0 [77], SPD1 [78]

These terminals are to set a communication rate.
All communication rates of the LSIs on the same communication line must be set to be same.
This terminal has a built-in pull up resistor to prevent floating.
When using H level, pull up to VDD (5k ~ 10k ohm) or connect directly to improve noise resistance when this terminal is set to H level.

| 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 communication status. (For details, see "4.4.11 TMD [72]".)
When a data transmission interval from a center LSI to this LSI exceeds a set time, the watchdog timer times out.
This terminal is used to set output conditions when the watchdog timer times out.
When TUD = H level: The LSI keeps its current status.
When TUD = L level: The LSI resets output of a general-purpose I/O terminal and stops immediately pulse output (stop operation).
This terminal has a built-in pull up resistor to prevent floating and can be open when it is not used. Pull up to VDD (5k ~ 10K ohm) to improve noise resistance.

4.4.11 TMD [72]

Specify operation time for the watchdog timer.
The watchdog timer is used to monitor 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 receiving the last data packet from the center LSI.
The reason for time out is problems such as disconnection of communication line or stop of communication by a center LSI.
The setting 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 |

This terminal has a built-in pull up resistor to prevent floating and can be open when it is not used. Pull up to VDD (5k ~ 10k ohm) to improve noise resistance.

4.4.12 TOUT [69]

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

4.4.13 SO [18]

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

4.4.14 SOEH [17], SOEL [16]

Output enable signal for communication.
The difference between SOEH signal and SOEL signal is that the logic is reverse.
When sending, SOEH = H level and SOEL = L level.

4.4.15 SOEI [15]

When using more than one G9103C for multi-layer connection, connect it with SOEH signal from other local LSIs to this terminal.
By logical “OR” operation with the output enable, the LSI outputs to SOEH and SOEL terminals.
Make the number of LSIs less than 4 because that OR processing time of each LSI is accumulated.
This terminal has a built-in pull up resistor to prevent floating and can be open when it is not used. Pull up to VDD (5k ~ 10k ohm) to improve noise resistance.

4.4.16 SI [20]

Serial input signal for communication. (Positive logic)
Input specification is Schmitt trigger. For details, see “10.3 DC characteristics”.

4.4.17 MRER [68]

This terminal is monitor output to check communication quality.
When the G9103C receives error frames such as CRC errors, this terminal goes L level for exactly 3.2 µs.
By timing this interval using a counter, you can check quality of communication.
When it is used as LED display, you can change L level time to approximate 100 ms using software setting (RENV2.EXER).

4.4.18 MSEL [67]

Communication status monitor output.
When the G9103C receives a communication frame normally (MRER = H level), this terminal goes L level for exactly 3.2 µs. The longest time of this signal cycle is cycle time.

#### 4.4.19 BRK [70]

By providing an H level pulse, the G9103C waits for a break frame.
When the G9103C receives a request for sending a break frame from the center LSI, it immediately sends a break frame.
The length of a break frame is 60 bits.
BRK signal pulse needs at least 3.2 µs long. (Positive logic)
This terminal has a built-in pull up resistor to prevent floating and can be open when it is not used. Pull up to VDD (5k ~ 10k ohm) to improve noise resistance.

#### 4.4.20 P0 [51], P1~P3 [53~55], P4/SIFC [56], P5/SIFS [57], P6/SIFI [58], P7/SIFO [59]

Using software, these terminals can be general-purpose input or output terminals (RENV2.P0M ~ P7M).
These terminals have built-in pull up resistors to prevent floating. These terminals can be left open when they are not used. Pull them up (5k ~ 10k ohm) to improve noise resistance.
Using software, P4 ~ P7 can be used as serial communication terminals to connect to the CPU. For details, see “8.19 CPU connection function”.

#### 4.4.21 STA [49], STP [50]

If you want to start multiple LSIs simultaneously, connect STA terminal of each LSI together.
If you want to stop multiple LSIs simultaneously when an error occurs, connect STP terminal of each LSI together.
These terminals have built in pull up resistors to prevent floating. These terminals can be left open when they are not used. Pull them up (5k ~ 10k ohm) to improve noise resistance. These terminals can be used to start or stop operation by an external signal.
It can be enabled using software setting (RMD.MSY, RMD.MSPE).

#### 4.4.22 EMG [42]

Input an emergency stop signal. (Level detection)
Because the signal is a high priority stop one, the final pulse width may not be normal.
While this terminal is L level, the G9103C prohibits operation. If this terminal goes L level while the motor is operating, the motor will stop immediately.
This terminal has a built-in pull up resistor to prevent floating and it can be left open when it is not used.
Pull it up (5k ~ 10 k ohm), or connect it to VDD directly to improve noise resistance.

#### 4.4.23 ELL [65]

This terminal is to set input logic of +EL and −EL signals. When this terminal is L level, the respective signal is set for positive logic input.
This terminal has a built-in pull up resistor to prevent floating and can be open when it is not used.
Pull it up (5k ~ 10 k ohm), or connect it to VDD directly to improve noise resistance.

#### 4.4.24 +EL [43], −EL [44]

Input EL signal (+EL signal in + direction operation and –EL signal in – direction operation). (Level detection)
Their input logic can be changed using the ELL terminals.
When the EL signal (in feed direction) turns ON, the motor stops immediately, or decelerates and stops. (Level detection)
These terminals have built in pull up resistors to prevent floating. When not used, they can be left open.
For improving noise resistance, pull it up (5k ~ 10 k ohm), or connect it to VDD directly with ELL terminal.

#### 4.4.25 SD [45]

This terminal is to input a deceleration signal (deceleration and stop signal). (Level detection)
Software can be used to change input logic of this terminal (RENV1.SDL). This terminal has a latch function.
This terminal has a built in pull up resistor to prevent floating and when it is not used, it can be left open.
Pull it up (5k ~ 10 k ohm), or connect it to VDD directly to improve noise resistance.

#### 4.4.26 ORG [46]

Input an origin return signal. (Edge detection)
Software can be used to change the input logic of this terminal (RENV1.ORGL).
This is used in origin return operation. Change from OFF to ON is detected as the origin. (Edge detection)
This terminal has a built-in pull up resistor to prevent floating and it can be left open when this terminal is not used.
Pull it up (5k ~ 10 k ohm) or connect it to VDD directly to improve noise resistance.

#### 4.4.27 ALM [47]

Input an alarm signal. (Level detection)
Software can be used to change input logic of this terminal (RENV1.ALML).
When this signal turns ON, a motor stops immediately, or decelerates and stops.
This terminal has a built-in pull up resistor to prevent floating and it can be left open when this terminal is not used.
Pull it up (5k ~ 10 k ohm) or connect it to VDD directly to improve noise resistance.

#### 4.4.28 OUT [31], DIR [32]

While the G9103C is in the common pulse mode, it sends feed pulses from OUT terminal, and supplies direction signals from DIR terminal.
While the G9103C is in Two-pulse mode, it outputs positive direction feed pulses from OUT terminal, and negative direction feed pulses from DIR terminal.

#### 4.4.29 PA [27], PB [28]

These terminals are used to operate a motor by external pulses, such as a manual pulse generator.
90 degree phase difference signals or Two-pulse signal (count forward pulse and backward pulse) can be input to these terminals. In the case of 90 degree phase difference signals, multiplication by 1, 2 or 4 can be selected.
These terminals have built-in pull up resistors to prevent floating and can be left open when these terminals are not used.
Pull it up (5k ~ 10 k ohm) or connect it to VDD directly to improve noise resistance.

#### 4.4.30 EA [24], EB [25], EZ [26]

These terminals are used to control current position using an encoder.
90 degree phase difference signals or Two-pulse (count forward pulse and backward pulse) can be input on these terminals. In the case of 90 degree phase difference signals, multiplication by 1, 2 or 4 can be selected.
EZ input is used for origin return operations. Input logic can be changed by software setting (RENV2.EZL).
These terminals have built-in pull up resistors to prevent floating and can be left open when these terminals are not used,
Pull it up (5k ~ 10 k ohm) or connect it to VDD directly to improve noise resistance.

#### 4.4.31 PCS [37]

The G9103C can start positioning control when the signal is input to this terminal (Override 2 of the target position).
Input logic can be changed by software setting (RENV1.PCSL).
This terminal has a built in pull up resistor to prevent floating and it can be left open when it is not used.
Pull it up (5k ~ 10 k ohm) or connect it to VDD directly to improve noise resistance.

#### 4.4.32 INP [40]

Input in-position (positioning complete) signal from a servo driver.
Positioning complete is delayed until this signal is input.
Input logic can be changed by software setting (RENV1.INPL).
This terminal has a built in pull up resistor to prevent floating and it can be left open when it is not used.
Pull it up (5k ~ 10 k ohm) or connect it to VDD directly to improve noise resistance.

#### 4.4.33 CLR [39]

Clear the counter specified by RENV3.CU1C ~ CU3C bits among COUNTER 1 ~ COUNTER 3, using inputting a signal.
Input logic can be changed by software setting (RENV1.CLR).
This terminal has a built in pull up resistor to prevent floating and it can be left open when it is not used. Pull it up (5k ~ 10 k ohm) or connect it to VDD directly to improve noise resistance.

#### 4.4.34 LTC [38]

Latch the values of all counters (COUNTER 1 ~ COUNTER 3) to RLTC1 ~ RLTC3 registers using inputting a signal. Counters can be clear by setting RENV4.CU1L ~ CU3L bits just after latching. Input logic can be changed by software setting (RENV1.LTCL).
This terminal has a built in pull up resistor to prevent floating and it can be left open when it is not used. Pull it up (5k ~ 10 k ohm) or connect it to VDD directly to improve noise resistance.

#### 4.4.35 ERC / CDWN [30]

Output a one-shot pulse to clear a deviation counter for a servo driver.
This terminal is also to be an output terminal (RENV1.CDWN) of a current down signal at the control of stepper motor.
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 BSY / PH1 [61]

This terminal doubles as BSY and PH1 output. The functions are selected by RMD register (RMD.MPH).
When BSY is selected (RMD.MPH = 0), the G9103C outputs L level from this terminal while a motor is operating.
When PH1 is selected (RMD.MPH = 1), the G9103C outputs excitation sequence for a 2-phase stepper motor.
If this terminal is not used, leave it open.

#### 4.4.37 FUP / PH2 [62]

This terminal doubles as FUP and PH2 output. The functions are selected by RMD register (RMD.MPH).
When FUP is selected (RMD.MPH = 0), the G9103C outputs an L level signal from this terminal while a motor is accelerating.
When PH2 is selected (RMD.MPH = 1), the G9103C outputs an excitation sequence for a 2-phase stepper motor.
If this terminal is not used, leave it open.

#### 4.4.38 FDW / PH3 [63]

This terminal doubles as FDW and PH3 output. The functions are selected by RMD register (RMD.MPH).
When FDW is selected (RMD.MPH = 0), the G9103C outputs an L level signal from this terminal while the motor is decelerating.
When PH3 is selected (RMD.MPH = 1), the G9103C outputs an excitation sequence for a 2-phase stepper motor.
If this terminal is not used, leave it open.

#### 4.4.39 MVC / PH4 [64]

This terminal doubles as MVC and PH4 output. The functions are selected by RMD register (RMD.MPH).
When MVC is selected (RMD.MPH = 0), the G9103C outputs an L level signal from this terminal while the motor is operating at a constant speed.
When PH4 is selected (RMD.MPH = 1), the G9103C outputs an excitation sequence for a 2-phase stepper motor.
If this terminal is not used, leave it open.

#### 4.4.40 CP1 [34]

When the conditions for Comparator 1 are met, the G9103C outputs an L level signal from this terminal.
If this terminal is not used, leave it open.

#### 4.4.41 CP2 [35]

When the conditions for Comparator 2 are met, the G9103C outputs an L level signal from this terminal. If this terminal is not used, leave it open.

#### 4.4.42 CP3 [36]

When the conditions for Comparator 3 are met, the G9103C outputs an L 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 3.0 V ~ 3.6 V to the VDD. Please make sure to use all VDD and GND terminals.

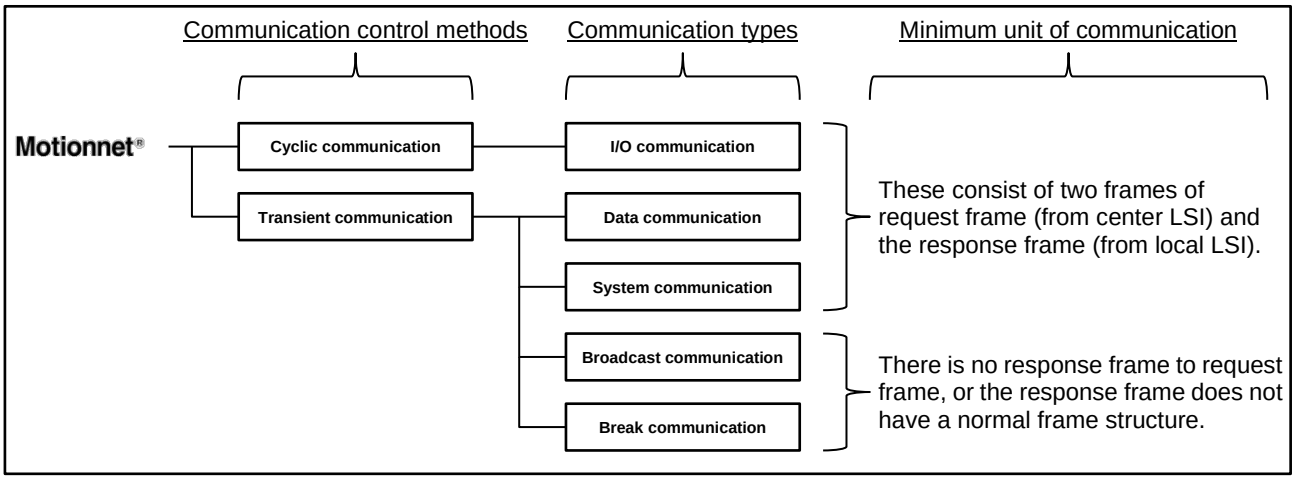
# 5. Description of software

## 5.1 Outline of control

### 5.1.1 Communication control

- The center LSI controls all communication.
- One set of communication consists of a request from the center LSI to the local LSIs (request frame), and a response 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 G9103C is a local LSI for data control.

### 5.1.2 Communication type




G9103C controls I/O communication, data communication, system communication, broadcast communication, and break communication.

#### 5.1.2.1 I/O communication (Cyclic communication)

I/O communication is to send and receive port and status information of a local LSI.
I/O communication starts with the local LSI that has the smallest device number and proceeds through all the local LSIs that are present. When the communication with the LSI that has the biggest device number is complete, the center LSI starts again to communicate with the local LSI that has the smallest device number. 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 G9103C 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 local LSIs, such as the G9103C, etc.
Normally, the center LSI forwards I/O information in cyclic communications. A data communication command from a CPU allows you to perform data communications by interrupting 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 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. The center LSI recognizes that no response error has occurred. In this case, it retries data communication (maximum three times).

5.1.2.3 System communication

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

5.1.2.4 Broadcast communication

The center LSI sends information to multiple local LSIs for data control simultaneously. Broadcast communication frame consists of group number and broadcast commands. Only local LSIs for data control that are specified by group numbers (ggg n the below table) f communication frame perform a broadcast command received. When a group number in communication frame is 000b, all groups are objects of the command.

Broadcast commands that G9103C can perform (ggg = group number)

| Broadcast command | Description |
|---|---|
| 0010 0ggg 0000 0001 | Start (STA signal substitute input command for multiple axes) * |
| 0010 0ggg 0000 0010 | Stop (STP signal substitute input command for multiple axes) * |
| 0010 0ggg 0000 0011 | Emergency stop (EMG signal substitute input 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 a center LSI. Three commands with * are enable to G9003. G9003's group number is always 000b.

[How to set group numbers]

To set group numbers, there are two following ways:

1. Set group numbers input terminals (GRP) by negative logic. You can confirm a setting value by main status (MSTS.SGP).
2. Set a value by writing to the group number setting registers (RENV2.GN). You can confirm the setting value by reading registers.

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

5.1.2.5 Break communications

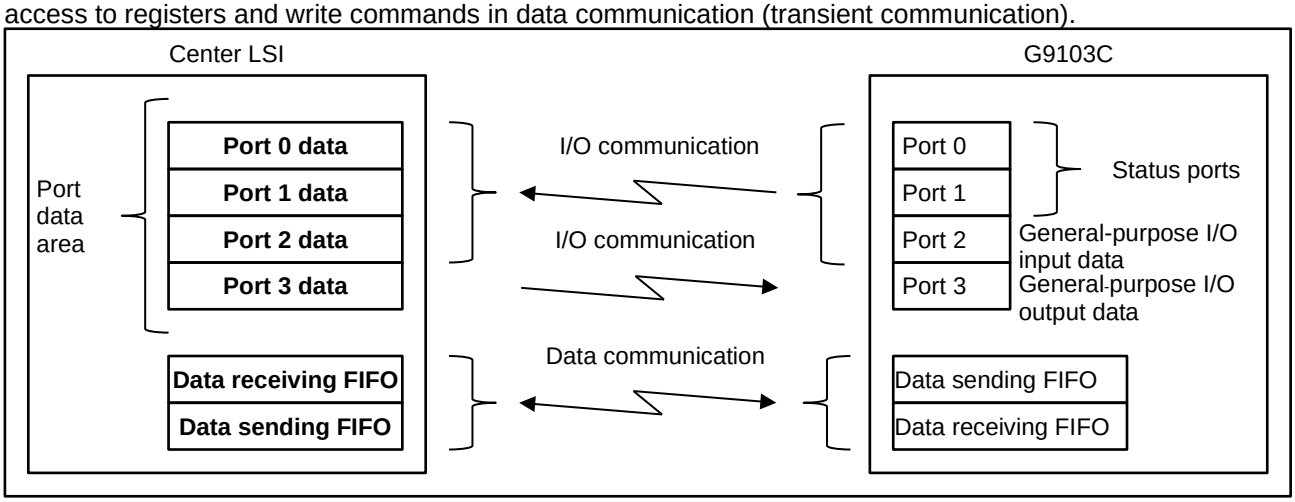
**Motionnet**® performs communication with the local LSIs that are registered in a center LSI before starting cyclic communication, by system communication or CPU software. Therefore, even if there are local LSIs that turned on after the start of communication, the local LSIs cannot perform communication until the local LSIs informs to the center LSI and are resisted additionally. Request communication for this additional registration is called as break communication.
Normally, a center LSI sends a request frame for sending a break frame every approximately 250 ms. When you input a break signal to the BRK terminal using such as switches, a local LSI returns a break frame at the receipt of the next request frame for sending a break frame.
When a center LSI receives a break frame, the break frame receiving bit (STSW.BRKF) of the status becomes 1 and it outputs an interrupt request signal to CPU
Then, the center LSI searches the source of request using CPU software and register it as the communication target additionally.

Notes: A center LSI sends a request frame for sending a break frame at specified time intervals (250 ms in 20 Mbps) in default. However, this process may disturb cycle of cycle time. If you set "break frame sending request" is to "Stop" ("RENV0.BK0F = 1), the automatic sending a request frame

for sending a break frame stops and cycle time 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.

5.1.2.6 Conceptual communication and setting examples

[Conceptual communication diagram]
G9103C sends and receives information of port 0 ~ 3 in I/O communication (cyclic communication) and access to registers and write commands in data communication (transient communication).




Note: Response frame of data communication (from G9103C to center LSI) contains information of port 0 to 3 after data processing is completed, so that status change by data communication can be transferred to the center LSI immediately.

G9103C 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, up to 21 registers can be set at once.
It can also contain control commands and read register commands as well as register settings. However, when you send multiple read register commands together at once, please avoid that response frame (from G9103C to center LSI) exceeds 64 words.

| Notes. In the examples described below, pre-registers (PRMV etc.) are used. When you use the system that pre-registers are not used at all like software for G9003, you can set registers instead of pre-registers (RMV, etc.). |
|---|

- Example 1
  Writing into one register of G9103C (setting on the center LSI)
  PRMV (Pre-register for setting feed amount) ←"01234567h"

  In the case of a 16-bit CPU (center LSI: I/F mode 3)
  1) First, write a write PRMV command (00B0h) to the sending FIFO (006h) (in the case that device number is 0).
  2) Next, write the lower 16-bit data (4567h) to be sent to PRMV register into the transmission FIFO (006h).
  3) Then, write the upper 16-bit data (0123h) to be sent to PRMV register into the transmission FIFO (006h).
  4) Finally, write a data communication command (4000h) into a command area (000h).

| | Details of data transmission FIFO | Description |
|---|---|---|
| 1st word | 00B0h | PRMV register write command |
| 2nd word | 4567h | PRMV register lower data |
| 3rd word | 0123h | PRMV register 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 0000h.

- Example 2

Writing into multiple registers of G9103C together at once (Setting on the center LSI)

PRMV (Pre-register for setting feed amount) ← 01234567h
PRFL (Pre-register for setting FL speed) ← 00000001h
PRFH (Pre-register for setting FH speed) ← 00001000h
PPUR (Pre-register for setting acceleration rate) ← 00000010h
PRMG (Pre-register for setting speed magnification) ← 000000C7h
PPMD (Pre-register for setting operation mode) ← 00000041h
STAUD (High-speed start 2 command ← 00000053h

Write a data communication command (4000h) into a command area (000h) (in the case that device number is 0).

| | Details of data transmission FIFO | Description |
|---|---|---|
| 1st word | 00B0h | Write PRMV register command |
| 2nd word | 4567h | PRMV register lower data |
| 3rd word | 0123h | PRMV register upper data |
| 4th word | 00B1h | Write PRFL register command |
| 5th word | 0001h | PRFL register lower data |
| 6th word | 0000h | PRFL register upper data |
| 7th word | 00B2h | Write PRFH register command |
| 8th word | 1000h | PRFH register lower data |
| 9th word | 0000h | PRFH register upper data |
| 10th word | 00B3h | Write PRUR register command |
| 11th word | 0010h | PRUR register lower data |
| 12th word | 0000h | PRUR register upper data |
| 13th word | 00B5h | Write PRMG register command |
| 14th word | 00C7h | PRMG register lower data |
| 15th word | 0000h | PRMG register upper data |
| 16th word | 00B7h | Write PRMD register command |
| 17th word | 0041h | PRMD register lower data |
| 18th word | 0000h | PRMD register upper data |
| 19th word | 0053h | STAUD start command |

- Example 3

Latch the contents of RCUN1 (COUNTER 1) ~ RCUN3 (COUNTER 3) and read the values (setting on the center LSI)

LTCH (Latch command) ← 0029h
RLTC1 (COUNTER 1 Latch data) Read ← 00EDh
RLTC2 (COUNTER 2 Latch data) Read ← 00EEh
RLTC3 (COUNTER 3 Latch data) Read ← 00EFh

Write a data communication command (4000h) into a command area (000h) (in the case that device number is 0).

| | Details of the data transmission FIFO | Description |
|---|---|---|
| 1st word | 0029h | LTCH command |
| 2nd word | 00Edh | RRLTC1 command |
| 3rd word | 00EEh | RRLTC2 command |
| 4th word | 00Efh | RRLTC3 command |

When communication completes, the following data is stored in the data reception FIFO of center LSI.
(In the case of RLTC1 = 01234567h, RLTC2 = 01234566h, RLTC3 = 00000001h)

| | Data of the data reception FIFO | Description |
|---|---|---|
| 1st word | 00EDh | RRLTC1 response command |
| 2nd word | 4567h | RLTC1 register lower data |
| 3rd word | 0123h | RLTC1 register upper data |
| 4th word | 00EEh | RRLTC2 response command |
| 5th word | 4566h | RLTC2 register lower data |
| 6th word | 0123h | RLTC2 register upper data |
| 7th word | 00EFh | RRLTC3 response command |
| 8th word | 0001h | RLTC3 register lower data |
| 9th word | 0000h | RLTC3 register upper data |

## *5.2 Functional settings*

### 5.2.1 I/O port

G9103C has four I/O ports for I/O communication (cyclic communication). The highest port, Port 3, is used to output. Ports 2, 1, and 0 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 from input or output using register RENV2.PnM bit (n = 0~7). Therefore, the settings for the general-purpose I/O output data will be enabled when the general-purpose I/O terminals are set up as outputs.
Output data setting of the general-purpose I/O output terminals can be reset by setting of TUD terminal when watchdog timer times out
Output status can be checked by reading 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)

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

**[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 | SGP | | | 0 | 0 | 0 | SBSY | SPDF | SPRF | 0 | 0 | SEVT | SERR | SEND | SINT |

**[In the case of RSYN.SYNC = 1] (Bits 5 and 4 are changed by RSYN.SYNE bit setting.)**

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

**[In the case of RSYN.SYNE = 1] (Bits 15 ~ 9 are changed by RSYN.SYNC bit setting.)**

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

| Bit | Bit name | Description |
|---|---|---|
| 0 | SINT | This bit becomes 1 when an interrupt request turns ON. (When any of bits 1, 2, or 3 is 1.) |
| 1 | SEND | When an operation stops, an interrupt request turns ON and this bit becomes 1. This bit is returned to 0 by an INTRS (0008h) command. |
| 2 | SERR | This bit becomes 1 when an error interrupt turns ON. It is returned to 0 by reading REST register or writing into REST register. |
| 3 | SEVT | This bit becomes 1 when an event interrupt turns ON. It is return set to 0 by reading RIST register or writing into RIST register. |
| 5, 4 | SPS | For simultaneous stop processing. Please ignore these when reading status. |
| 6 | SPRF | This bit becomes 1 when the pre-register for operation is full. |
| 7 | SPDF | This bit becomes 1 when the pre-register for comparator 3 is full. |
| 8 | SBSY | This bit becomes 1 when the LSI starts to output pulses. It is set back to 0 by stopping operation. (= BSY signal) |
| 14 ~ 12 | SGP | The status of group number for broadcast communication.<br>When RENV2.GN = 000b, these bits are set same as GRP terminal. When other than 000b, these bits are set same as RENV2.GN. |
| 15 ~ 9 | SYN | For clock synchronizing processing. Please ignore these when reading status. |

Note: When interrupt request is disabled (RMD.MINT = 1), MSTS.SINT bit is always 0 although MSTS.SEND, SERR and SEVT bit become 1.

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 |
|---|---|---|
| 7 ~ 0 | IP7 ~ 0 | These bits are to read status of terminals P7 ~ P0. (0: L level, 1: H level) |

Note: In the case that terminals P7 ~ P4 is for serial bus control (RENV2.SIFM = 1), IP7 ~ 4 are set to 0.
When the general-purpose I/O terminals are set as output terminals, the status of the output terminals can be read.
In the case that monitor for input data of general purpose input/output terminal is disabled (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 |
|---|---|---|
| 7 ~ 0 | OP7 ~ OP0 | These bits are to set the output status of terminals P7 ~ P0. (0: L level, 1: H level) |

Note: In the case that terminals P7 ~ P4 is for serial bus control (RENV2.SIFM = 1), the setting for OP7 ~ OP4 is disabled.

## *5.3 Command (Operation commands and Control commands)*

The G9103C can control an axis through data communications from a 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 single command” that is to communicate one command at once independently 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 “communication with single command”, the upper word is recognized as 0000h 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 “0000h”.
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 a command and the G9013C malfunctions.

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.

### 5.3.1 Operation commands

#### 5.3.1.1 Start command

1) Start command
To start operation, write one of these commands while a motor is stopping.

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

2) Remaining amount start command
If these commands are written after a positioning operation (including linear interpolation and circular interpolation) is halted without completing, operation of the remaining number of pulses in the positioning counter is executed.

| Symbol | 1st word | Description | Response frame |
|---|---|---|---|
| CNTFL | 0054h | Remaining amount by FL constant speed start | Response only |
| CNTFH | 0055h | Remaining amount by FH constant speed start | Response only |
| CNTD | 0056h | Remaining amount by high speed start 1<br>(FH constant speed →Deceleration) | Response only |
| CNTUD | 0057h | Remaining amount by high speed start 2<br>(Acceleration → FH constant speed → Deceleration) | Response only |

3) Start command with a feed amount (G9003 compatible function)

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

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

Note 1: If the command is sent without any data, RMV register will be set to 0 and the motor will operate feed amount "0".

Note 2: Because this command is compatible with G9003, data is not set to PRMV register but to RMV register. Therefore, do not write to a pre-register or a start command while a motor is running. Though these commands are written while a motor is running, they are ignored.

4) Simultaneous start command

By making holding start enabled (RMD.MSY = 1), and writing a start command, several LSIs are waiting or an STA signal.

By writing this command in this condition, a motors start to operate.

To start multi axes simultaneously, write CMSTA (0006h) command.

To start only own axis, write SPSTA (002Ah) command.

| Symbol | 1st word | Description | Response frame |
|---|---|---|---|
| CMSTA | 0006h | STA signal output (Because output signal is input again, own axis also starts simultaneously.) | Response only |
| SPSTA | 002Ah | Substitute for STA signal input (Because output signal is not output, only own axis starts.) | Response only |

5) Start command for next operation

This command is to continue interpolation operation. If this command is written while a motor is stopping, error interrupt occurs and a motor does not start.

| Symbol | 1st word | Description | Response frame |
|---|---|---|---|
| NSTAFL | 005Ch | FL constant speed start for next operation | Response only |
| NSTAFH | 005Dh | FH constant speed start for next operation | Response only |
| NSTAD | 005Eh | High-speed start 1 for next operation (FH constant speed → deceleration) | Response only |
| NSTAUD | 005Fh | High-speed start 2 for next operation (Acceleration → FH constant speed → deceleration) | Response only |

5.3.1.2 Speed change command

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

| Symbol | 1st word | Description | Response frame |
|---|---|---|---|
| FCHGL | 0040h | Change to FL constant speed immediately | Response only |
| FCHGH | 0041h | Change to FH constant speed immediately | Response only |
| FSCHL | 0042h | Decelerate to FL speed | Response only |
| FSCHH | 0043h | Accelerate to FH speed | Response only |

#### 5.3.1.3 Stop command

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

| Symbol | 1st word | Description | Response frame |
|---|---|---|---|
| STOP | 0049h | Stop immediately. | Response only |
| SDSTP | 004Ah | Decelerate and stop | Response only |

2) Simultaneous stop command
This command stops axes that signal input are enabled (RMD.MSPE = 1).

| Symbol | 1st word | Description | Response frame |
|---|---|---|---|
| CMSTP | 0007h | Output of STP signal (simultaneous stop) | Response only |

3) Emergency stop command
This command is written while motor is running, motor stops in emergency.

| Symbol | 1st word | Description | Response frame |
|---|---|---|---|
| CMEMG | 0005h | Emergency stop | Response only |

### 5.3.2 Control commands

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

#### 5.3.2.1 NOP (no operation) command

| Symbol | 1st word | Description | Response frame |
|---|---|---|---|
| NOP | 0000h | Disabled command. | Response only |

#### 5.3.2.2 Operation stop interrupt reset command

| Symbol | 1st word | Description | Response frame |
|---|---|---|---|
| INTRS | 0008h | Reset operation stop interrupt (MSTS.SEND) | Response only |

#### 5.3.2.3 Software reset command

This command resets registers and commands (except for communication related items)

| Symbol | 1st word | Description | Response frame |
|---|---|---|---|
| SRST | 0004h | Software reset | Response only |

#### 5.3.2.4 Counter clear command

These commands clear the specified counter to 0.

| Symbol | 1st word | Description | Response frame |
|---|---|---|---|
| CUN1R | 0020h | Clear COUNTER 1 (RCUN1) | Response only |
| CUN2R | 0021h | Clear COUNTER 2 (RCUN2) | Response only |
| CUN3R | 0022h | Clear COUNTER 3 (RCUN3) | Response only |
| CKMECR | 0030h | Clear an error counter (RMEC.CKEC) for receiving clock synchronizing communication | Response only |
| SPMECR | 0031h | Clear an error counter (RMEC.SPEC) for receiving simultaneous stop communication | Response only |
| SWMECR | 0032h | Clear an error counter (RMEC.SWEC) for receiving sensor substitute input communication | Response only |

5.3.2.5 ERC signal and CDWN signal output control command

These commands control ERC signal and CDWN signal output.

| Symbol | 1st word | Description | Response frame |
|---|---|---|---|
| ERCOUT | 0024h | Output ERC signal | Response only |
| ERCRST | 0025h | Reset ERC signal / Reset CDWN signal | Response only |

5.3.2.6 PCS signal input command

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

| Symbol | 1st word | Description | Response frame |
|---|---|---|---|
| STAON | 0028h | Substitute for PCS signal input | Response only |

5.3.2.7 LTC signal input command

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

| Symbol | 1st word | Description | Response frame |
|---|---|---|---|
| LTCH | 0029h | Substitute for LTC signal input | Response only |

5.3.2.8 Pre-register control command

These commands control pre-registers for operation and RCMP3.

| Symbol | 1st word | Description | Response frame |
|---|---|---|---|
| PRECAN | 0026h | This command cancels determined status of pre-register for operation | Response only |
| PCPCAN | 0027h | This command cancels determined status of PRCP3 register. | Response only |
| PRESHF | 002Bh | This command copies a pre-register setting value for operation to an operation register. | Response only |
| PCPSHF | 002Ch | This command copies PRCP3 register value to RCMP3 register. | Response only |
| PFCCAN | 002Dh | This command cancels determined status of PRCP3 register and RCMP3 register. | Response only |

Note: PCPCAN command cancels only determined status of PRCP3 register.
PFCCAN command cancels both determined status of PRCP3 register and RCMP3 register.

5.3.2.9 EEPROM control commands

| Symbol | Cord | Description | Response frame |
|---|---|---|---|
| ROMPE | 000Ah | This command prohibits writing to external EEPROM<br>Performs the following processing for EEPROM.<br>1. It sends a command to permit writing to EEPROM (00000110).<br>2. It ensures 3.2 μs interval time when ROMS = H level.<br>3. It writes 11111100 into status register of EEPROM | Response only |
| ROMPD | 000Bh | This command permits writing to external EEPROM<br>Performs the following processing for EEPROM.<br>1. It sends a command to permit writing to EEPROM (00000110).<br>2. It ensures 3.2 μs interval time when ROMS = H level.<br>3. It writes 11110000 into status register of EEPROM | Response only |
| ROMWR | 000Ch | This command writes the contents of RGN0 ~ RGN3 registers into EEPROM<br>Performs the following processing for EEPROM.<br>1. It sends a command to permit writing to EEPROM (00000110).<br>2. It ensures 3.2 μs interval time when ROMS = H level.<br>3. Writes 16 bytes of RGN0 ~ RGN3 registers into EEPROM. | Response only |
| ROMRD | 000Dh | This command reads EEPROM and sets to RGN0 ~ RGN3 registers.<br>Performs the following processing for EEPROM.<br>1. Reads 16 bytes from address 0 of EEPROM together at once and writes into RGN0 ~ RGN3 registers. | Response only |

5.3.2.10 ID code check command

Sets ID code on the upper 16 bits of RMG.

| Symbol | 1st word | Description | Response frame |
|---|---|---|---|
| IDMON | 0003h | Make ID code read by RRMG command | Response only |

### 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 | RMV | - | - | 0080h | WRMVOR | - | - | - | - | - |
| Positioning amount | RMV | 00D0h | RRMV | 0090h | WRMV | PRMV | 00C0h | RPRMV | 00B0h | WPRMV |
| FL speed | RFL | 00D1h | RRFL | 0091h | WRFL | PRFL | 00C1h | RPRFL | 00B1h | WPRFL |
| FH speed | RFH | 00D2h | RRFH | 0092h | WRFH | PRFH | 00C2h | RPRFH | 00B2h | WPRFH |
| Acceleration rate | RUR | 00D3h | RRUR | 0093h | WRUR | PRUR | 00C3h | RPRUR | 00B3h | WPRUR |
| Deceleration rate | RDR | 00D4h | RRDR | 0094h | WRDR | PRDR | 00C4h | RPRDR | 00B4h | WPRDR |
| Speed magnification | RMG | 00D5h | RRMG | 0095h | WRMG | PRMG | 00C5h | RPRMG | 00B5h | WPRMG |
| Ramping-down point | RDP | 00D6h | RRDP | 0096h | WRDP | PRDP | 00C6h | RPRDP | 00B6h | WPRDP |
| Operation mode | RMD | 00D7h | RRMD | 0097h | WRMD | PRMD | 00C7h | RPRMD | 00B7h | WPRMD |
| X-coordinate of the center in circular interpolation | RIP | 00D8h | RRIP | 0098h | WRIP | PRIP | 00C8h | RPRIP | 00B8h | WPRIP |
| S-curve range while acceleration | RUS | 00D9h | RRUS | 0099h | WRUS | PRUS | 00C9h | RPRUS | 00B9h | WPRUS |
| S-curve range while deceleration | RDS | 00DAh | RRDS | 009Ah | WRDS | PRDS | 00CAh | RPRDS | 00BAh | WPRDS |
| FA speed | RFA | 00DBh | RRFA | 009Bh | WRFA | - | - | - | - | - |
| Environmental setting 1 | RENV1 | 00DCh | RRENV1 | 009Ch | WRENV1 | - | - | - | - | - |
| Environmental setting 2 | RENV2 | 00DDh | RRENV2 | 009Dh | WRENV2 | - | - | - | - | - |
| Environmental setting 3 | RENV3 | 00DEh | RRENV3 | 009Eh | WRENV3 | - | - | - | - | - |
| Environmental setting 4 | RENV4 | 00DFh | RRENV4 | 009Fh | WRENV4 | - | - | - | - | - |
| Environmental setting 5 | RENV5 | 00E0h | RRENV5 | 00A0h | WRENV5 | - | - | - | - | - |
| Environmental setting 6 | RENV6 | 00E1h | RRENV6 | 00A1h | WRENV6 | - | - | - | - | - |
| COUNTER 1 | RCUN1 | 00E3h | RRCUN1 | 00A3h | WRCUN1 | - | - | - | - | - |
| COUNTER 2 | RCUN2 | 00E4h | RRCUN2 | 00A4h | WRCUN2 | - | - | - | - | - |
| COUNTER 3 | RCUN3 | 00E5h | RRCUN3 | 00A5h | WRCUN3 | - | - | - | - | - |
| Comparison data for comparator 1 | RCMP1 | 00E7h | RRCMP1 | 00A7h | WRCMP1 | - | - | - | - | - |
| Comparison data for comparator 2 | RCMP2 | 00E8h | RRCMP2 | 00A8h | WRCMP2 | - | - | - | - | - |
| Comparison data for comparator 3 | RCMP3 | 00E9h | RRCMP3 | 00A9h | WRCMP3 | PRCP3 | 00CBh | RPRCP3 | 00BBh | WPRCP3 |

| Contents | Register | | | | | Pre-register | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Name | Read command | | Write command | | Name | Read command | | Write command | |
| | | Code | Symbol | Code | Symbol | | Code | Symbol | Code | Symbol |
| Event interrupt cause | RIRQ | 00ECh | RRIRQ | 00ACh | WRIRQ | - | - | - | - | - |
| COUNTER 1 latched data | RLTC1 | 00EDh | RRLTC1 | - | - | - | - | - | - | - |
| COUNTER 2 latched data | RLTC2 | 00EEh | RRLTC2 | - | - | - | - | - | - | - |
| COUNTER 3 latched data | RLTC3 | 00EFh | RRLTC3 | - | - | - | - | - | - | - |
| Extended status | RSTS | 00F1h | RRSTS | - | - | - | - | - | - | - |
| Error interrupt status | REST | 00F2h | RREST | 00ADh | WREST | - | - | - | - | - |
| Event interrupt status | RIST | 00F3h | RRIST | 00AEh | WRIST | - | - | - | - | - |
| Positioning counter | RPLS | 00F4h | RRPLS | - | - | - | - | - | - | - |
| Current speed monitor | RSPD | 00F5h | RRSPD | - | - | - | - | - | - | - |
| Ramping-down point auto setting value | RSDC | 00F6h | RRSDC | - | - | - | - | - | - | - |
| Stepping counter in circular interpolation | RCIC | 00FBh | RRCIC | - | - | - | - | - | - | - |
| Number of stepping in circular interpolation | RCI | 00FCh | RRCI | 008Ch | WRCI | PRCI | 00CCh | RPRCI | 00BCh | WPRCI |
| Axis Y positioning amount in interpolation | RMVY | 00FDh | RRMVY | 008Dh | WRMVY | PRMVY | 00CDh | RPRMVY | 00BDh | WPRMVY |
| Y-coordinate of the center in circular interpolation | RIPY | 00FEh | RRIPY | 008Eh | WRIPY | PRIPY | 00CEh | RPRIPY | 00BEh | WPRIPY |
| Synchronizing control | RSYN | 00FFh | RRSYN | 008Fh | WRSYN | - | - | - | - | - |
| Synchronizing control 2 | RSYN2 | 00EBh | RRSYN2 | 00ABh | WRSYN2 | - | - | - | - | - |
| Error counter for receiving communication | RMEC | 00CFh | RRMEC | - | - | - | - | - | - | - |
| General-purpose register 0 | RGN0 | 00F7h | RRGN0 | 0087h | WRGN0 | - | - | - | - | - |
| General-purpose register 1 | RGN1 | 00F8h | RRGN1 | 0088h | WRGN1 | - | - | - | - | - |
| General-purpose register 2 | RGN2 | 00F9h | RRGN2 | 0089h | WRGN2 | - | - | - | - | - |
| General-purpose register 3 | RGN3 | 00FAh | RRGN3 | 008Ah | WRGN3 | | - | - | - | - |

Note: Please use general-purpose register 0 as a dedicated unit ID setting.

5.3.3.1 Register write commands

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

| Symbol | 1st word | 2nd word | 3rd word | Description | Response frame |
|---|---|---|---|---|---|
| WRMVOR | 0080h | Lower byte data | Upper byte data | Write override to RMV register | Response only |
| WRGN0 | 0087h | Lower byte data | Upper byte data | Write to RGN0 register | Response only |
| WRGN1 | 0088h | Lower byte data | Upper byte data | Write to RGN1 register | Response only |
| WRGN2 | 0089h | Lower byte data | Upper byte data | Write to RGN2 register | Response only |
| WRGN3 | 008Ah | Lower byte data | Upper byte data | Write to RGN3 register | Response only |
| WRCI | 008Ch | Lower byte data | Upper byte data | Write to RCI register | Response only |
| WRMVY | 008Dh | Lower byte data | Upper byte data | Write to RMVY register | Response only |
| WRIPY | 008Eh | Lower byte data | Upper byte data | Write to RIPY register | Response only |
| WRSYN | 008Fh | Lower byte data | Upper byte data | Write to RSYN register | Response only |
| WRMV | 0090h | Lower byte data | Upper byte data | Write to RMV register | Response only |
| WRFL | 0091h | Lower byte data | Upper byte data | Write to RFL register | Response only |
| WRFH | 0092h | Lower byte data | Upper byte data | Write to RFH register | Response only |
| WRUR | 0093h | Data | 0 | Write to RUR register | Response only |
| WRDR | 0094h | Data | 0 | Write to RDR register | Response only |
| WRMG | 0095h | Data | 0 | Write to RMG register | Response only |
| WRDP | 0096h | Lower byte data | Upper byte data | Write to RDP register | Response only |
| WRMD | 0097h | Lower byte data | Upper byte data | Write to RMD register | Response only |
| WRIP | 0098h | Lower byte data | Upper byte data | Write to RIP register | Response only |
| WRUS | 0099h | Data | 0 | Write to RUS register | Response only |
| WRDS | 009Ah | Data | 0 | Write to RDS register | Response only |
| WRFA | 009Bh | Lower byte data | Upper byte data | Write to RFA register | Response only |
| WRENV1 | 009Ch | Lower byte data | Upper byte data | Write to RENV1 register | Response only |
| WRENV2 | 009Dh | Lower byte data | Upper byte data | Write to RENV2 register | Response only |
| WRENV3 | 009Eh | Lower byte data | Upper byte data | Write to RENV3 register | Response only |
| WRENV4 | 009Fh | Lower byte data | Upper byte data | Write to RENV4 register | Response only |
| WRENV5 | 00A0h | Lower byte data | Upper byte data | Write to RENV5 register | Response only |
| WRENV6 | 00A1h | Lower byte data | Upper byte data | Write to RENV6 register | Response only |
| WRCUN1 | 00A3h | Lower byte data | Upper byte data | Write to RCUN1 register | Response only |
| WRCUN2 | 00A4h | Lower byte data | Upper byte data | Write to RCUN2 register | Response only |
| WRCUN3 | 00A5h | Data | 0 | Write to RCUN3 register | Response only |
| WRCMP1 | 00A7h | Lower byte data | Upper byte data | Write to RCMP1 register | Response only |
| WRCMP2 | 00A8h | Lower byte data | Upper byte data | Write to RCMP2 register | Response only |
| WRCMP3 | 00A9h | Lower byte data | Upper byte data | Write to RCMP3 register | Response only |
| WRSYN2 | 00ABh | Lower byte data | Upper byte data | Write to RSYN2 register | Response only |
| WRIRQ | 00ACh | Lower byte data | Upper byte data | Write to RIRQ register | Response only |

| Symbol | 1st word | 2nd word | 3rd word | Description | Response frame |
|---|---|---|---|---|---|
| WREST | 00ADh | Lower byte data | Upper byte data | Write to REST register | Response only |
| WRIST | 00AEh | Lower byte data | Upper byte data | Write to RIST register | Response only |
| WPRMV | 00B0h | Lower byte data | Upper byte data | Write to pre-register for RMV. | Response only |
| WPRFL | 00B1h | Lower byte data | Upper byte data | Write to pre-register for RFL. | Response only |
| WPRFH | 00B2h | Lower byte data | Upper byte data | Write to pre-register for RFH. | Response only |
| WPRUR | 00B3h | Data | 0 | Write to pre-register for RUR. | Response only |
| WPRDR | 00B4h | Data | 0 | Write to pre-register for RDR. | Response only |
| WPRMG | 00B5h | Data | 0 | Write to pre-register for RMG. | Response only |
| WPRDP | 00B6h | Lower byte data | Upper byte data | Write to pre-register for RDP. | Response only |
| WPRMD | 00B7h | Lower byte data | Upper byte data | Write to pre-register for RMD. | Response only |
| WPRIP | 00B8h | Lower byte data | Upper byte data | Write to pre-register for RIP. | Response only |
| WPRUS | 00B9h | Data | 0 | Write to pre-register for RUS. | Response only |
| WPRDS | 00BAh | Data | 0 | Write to pre-register for RDS. | Response only |
| WPRCP3 | 00BBh | Lower byte data | Upper byte data | Write to pre-register for RCMP3. | Response only |
| WPRCI | 00BCh | Lower byte data | Upper byte data | Write to pre-register for RCI. | Response only |
| WPRMVY | 00BDh | Lower byte data | Upper byte data | Write to pre-register for RMVY. | Response only |
| WPRIPY | 00BEh | Lower byte data | Upper byte data | Write to pre-register for RIPY. | Response only |

5.3.3.2 Register read controls

Note: If you read from several registers in one communication, please write 1 word command continuously.

| Symbol | 1st word | Description | Response frame |
|---|---|---|---|
| RPRMV | 00C0h | Read pre-register for RMV | APRMV + data |
| RPRFL | 00C1h | Read pre-register for RFL | APRFL + data |
| RPRFH | 00C2h | Read pre-register for RFH | APRFH + data |
| RPRUR | 00C3h | Read pre-register for RUR | APRUR + data |
| RPRDR | 00C4h | Read pre-register for RDR | APRDR + data |
| RPRMG | 00C5h | Read pre-register for RMG | APRMG + data |
| RPRDP | 00C6h | Read pre-register for RDP | APRDP + data |
| RPRMD | 00C7h | Read pre-register for RMD | APRMD + data |
| RPRIP | 00C8h | Read pre-register for RIP | APRIP + data |
| RPRUS | 00C9h | Read pre-register for RUS | APRUS + data |
| RPRDS | 00CAh | Read pre-register for RDS | APRDS + data |
| RPRCP3 | 00CBh | Read pre-register for RCMP3 | APRCP3 + data |
| RPRCI | 00CCh | Read pre-register for RCI | APRCI + data |
| RPRMVY | 00CDh | Read pre-register for RMVY | APRMVY + data |
| RPRIPY | 00CEh | Read pre-register for RIPY | APRIPY + data |
| RRMEC | 00CFh | Read RMEC register | ARMEC + data |
| RRMV | 00D0h | Read RMV register | ARMV + data |
| RRFL | 00D1h | Read RFL register | ARFL + data |
| RRFH | 00D2h | Read RFH register | ARFH + data |
| RRUR | 00D3h | Read RUR register | ARUR + data |
| RRDR | 00D4h | Read RDR register | ARDR + data |
| RRMG | 00D5h | Read RMG register | ARMG + data |
| RRDP | 00D6h | Read RDP register | ARDP + data |
| RRMD | 00D7h | Read RMD register | ARMD + data |
| RRIP | 00D8h | Read RIP register | ARIP + data |
| RRUS | 00D9h | Read RUS register | ARUS + data |
| RRDS | 00DAh | Read RDS register | ARDS + data |
| RRFA | 00DBh | Read RFA register | ARFA + data |
| RRENV1 | 00DCh | Read RENV1 register | ARENV1 + data |
| RRENV2 | 00DDh | Read RENV2 register | ARENV2 + data |
| RRENV3 | 00DEh | Read RENV3 register | ARENV3 + data |
| RRENV4 | 00DFh | Read RENV4 register | ARENV4 + data |
| RRENV5 | 00E0h | Read RENV5 register | ARENV5 + data |
| RRENV6 | 00E1h | Read RENV6 register | ARENV6 + data |
| RRCUN1 | 00E3h | Read RCUN1 register | ARCUN1 + data |
| RRCUN2 | 00E4h | Read RCUN2 register | ARCUN2 + data |
| RRCUN3 | 00E5h | Read RCUN3 register | ARCUN3 + data |
| RRCMP1 | 00E7h | Read RCMP1 register | ARCMP1 + data |
| RRCMP2 | 00E8h | Read RCMP2 register | ARCMP2 + data |

| Symbol | 1st word | Description | Response frame |
|---|---|---|---|
| RRCMP3 | 00E9h | Read RCMP3 register | ARCMP3 + data |
| RRSYN2 | 00EBh | Read RSYN2 register | ARSYN2 + data |
| RRIRQ | 00ECh | Read RIRQ register | ARIRQ + data |
| RRLTC1 | 00EDh | Read RLTC1 register | ARLTC1 + data |
| RRLTC2 | 00EEh | Read RLTC2 register | ARLTC2 + data |
| RRLTC3 | 00EFh | Read RLTC3 register | ARLTC3 + data |
| RRSTS | 00F1h | Read RSTS register | ARSTS + data |
| RREST | 00F2h | Read REST register | AREST + data |
| RRIST | 00F3h | Read RIST register | ARIST + data |
| RRPLS | 00F4h | Read RPLS register | ARPLS + data |
| RRSPD | 00F5h | Read RSPD register | ARSPD + data |
| RRSDC | 00F6h | Read RSDC register | ARSDC + data |
| RRGN0 | 00F7h | Read RGN0 register | ARGN0 + data |
| RRGN1 | 00F8h | Read RGN1 register | ARGN1 + data |
| RRGN2 | 00F9h | Read RGN2 register | ARGN2 + data |
| RRGN3 | 00FAh | Read RGN3 register | ARGN3 + data |
| RRCIC | 00FBh | Read RCIC register | ARCIC + data |
| RRCI | 00FCh | Read RCI register | ARCI + data |
| RRMVY | 00FDh | Read RMVY register | ARMVY + data |
| RRIPY | 00FEh | Read RIPY register | ARIPY + data |
| RRSYN | 00FFh | Read RSYN register | ARSYN + data |

5.3.3.3 Response data by read commands

| Symbol | 1st word | 2nd word | 3rd word | Description | Effective number of bits |
|---|---|---|---|---|---|
| APRMV | 00C0h | Lower byte data | Upper byte data | Read pre-register for RMV | 28 |
| APRFL | 00C1h | Lower byte data | Upper byte data | Read pre-register for RFL | 17 |
| APRFH | 00C2h | Lower byte data | Upper byte data | Read pre-register for RFH | 17 |
| APRUR | 00C3h | Data | 0 | Read pre-register for RUR | 16 |
| APRDR | 00C4h | Data | 0 | Read pre-register for RDR | 16 |
| APRMG | 00C5h | Lower byte data | Upper byte data | Read pre-register for RMG | 32 |
| APRDP | 00C6h | Lower byte data | Upper byte data | Read pre-register for RDP | 24 |
| APRMD | 00C7h | Lower byte data | Upper byte data | Read pre-register for RMD | 31 |
| APRIP | 00C8h | Lower byte data | Upper byte data | Read pre-register for RIP | 28 |
| APRUS | 00C9h | Data | 0 | Read pre-register for RUS | 16 |
| APRDS | 00CAh | Data | 0 | Read pre-register for RDS | 16 |
| APRCP3 | 00CBh | Lower byte data | Upper byte data | Read pre-register for RCMP3 | 28 |
| APRCI | 00CCh | Lower byte data | Upper byte data | Read pre-register for RCI | 31 |
| APRMVY | 00CDh | Lower byte data | Upper byte data | Read pre-register for RMVY | 28 |
| APRIPY | 00CEh | Lower byte data | Upper byte data | Read pre-register for RIPY | 28 |
| ARMEC | 00CFh | Lower byte data | Upper byte data | Read RMEC register | 24 |
| ARMV | 00D0h | Lower byte data | Upper byte data | Read RMV register | 28 |
| ARFL | 00D1h | Lower byte data | Upper byte data | Read RFL register | 17 |
| ARFH | 00D2h | Lower byte data | Upper byte data | Read RFH register | 17 |
| ARUR | 00D3h | Data | 0 | Read RUR register | 16 |
| ARDR | 00D4h | Data | 0 | Read RDR register | 16 |
| ARMG | 00D5h | Data | 0 | Read RMG register | 11 |
| ARDP | 00D6h | Lower byte data | Upper byte data | Read RDP register | 24 |
| ARMD | 00D7h | Lower byte data | Upper byte data | Read RMD register | 31 |
| ARIP | 00D8h | Lower byte data | Upper byte data | Read RIP register | 28 |
| ARUS | 00D9h | Data | 0 | Read RUS register | 16 |
| ARDS | 00DAh | Data | 0 | Read RDS register | 16 |
| ARFA | 00DBh | Lower byte data | Upper byte data | Read RFA register | 17 |
| ARENV1 | 00DCh | Lower byte data | Upper byte data | Read RENV1 register | 32 |
| ARENV2 | 00DDh | Lower byte data | Upper byte data | Read RENV2 register | 29 |
| ARENV3 | 00DEh | Lower byte data | Upper byte data | Read RENV3 register | 31 |
| ARENV4 | 00DFh | Lower byte data | Upper byte data | Read RENV4 register | 32 |
| ARENV5 | 00E0h | Lower byte data | Upper byte data | Read RENV5 register | 32 |
| ARENV6 | 00E1h | Lower byte data | Upper byte data | Read RENV6 register | 32 |
| ARCUN1 | 00E3h | Lower byte data | Upper byte data | Read RCUN1 register | 28 |
| ARCUN2 | 00E4h | Lower byte data | Upper byte data | Read RCUN2 register | 28 |
| ARCUN3 | 00E5h | Data | Code extension | Read RCUN3 register | 16 |
| ARCMP1 | 00E7h | Lower byte data | Upper byte data | Read RCMP1 register | 28 |

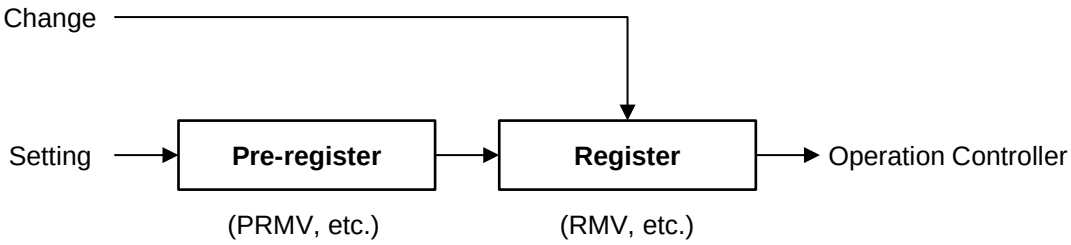
| Symbol | 1st word | 2nd word | 3rd word | Description | Effective number of bits |
|---|---|---|---|---|---|
| ARCMP2 | 00E8h | Lower byte data | Upper byte data | Read RCMP2 register | 28 |
| ARCMP3 | 00E9h | Lower byte data | Upper byte data | Read RCMP3 register | 28 |
| ARSYN2 | 00EBh | Lower byte data | Upper byte data | Read RSYN2 register | 28 |
| ARIRQ | 00ECh | Lower byte data | Upper byte data | Read RIRQ register | 17 |
| ARLTC1 | 00EDh | Lower byte data | Upper byte data | Read RLTC1 register | 28 |
| ARLTC2 | 00EEh | Lower byte data | Upper byte data | Read RLTC2 register | 28 |
| ARLTC3 | 00EFh | Lower byte data | Upper byte data | Read RLTC3 register | 17 |
| ARSTS | 00F1h | Lower byte data | Upper byte data | Read RSTS register | 32 |
| AREST | 00F2h | Lower byte data | Upper byte data | Read REST register | 23 |
| ARIST | 00F3h | Lower byte data | Upper byte data | Read RIST register | 17 |
| ARPLS | 00F4h | Lower byte data | Upper byte data | Read RPLS register | 28 |
| ARSPD | 00F5h | Lower byte data | Upper byte data | Read RSPD register | 27 |
| ARSDC | 00F6h | Lower byte data | Upper byte data | Read RSDC register | 24 |
| ARGN0 | 00F7h | Lower byte data | Upper byte data | Read RGN0 register | 32 |
| ARGN1 | 00F8h | Lower byte data | Upper byte data | Read RGN1 register | 32 |
| ARGN2 | 00F9h | Lower byte data | Upper byte data | Read RGN2 register | 32 |
| ARGN3 | 00FAh | Lower byte data | Upper byte data | Read RGN3 register | 32 |
| ARCIC | 00FBh | Lower byte data | Upper byte data | Read RCIC register | 31 |
| ARCI | 00FCh | Lower byte data | Upper byte data | Read RCI register | 31 |
| ARMVY | 00FDh | Lower byte data | Upper byte data | Read RMVY register | 28 |
| ARIPY | 00FEh | Lower byte data | Upper byte data | Read RIPY register | 28 |
| ARSYN | 00FFh | Lower byte data | Upper byte data | Read RSYN register | 32 |

## *5.4 Register and pre-register*

Note 1: The bits shown with * are ignored while they are written and become 0 while they are read in the explanation hereinafter described.
Note 2: The bits shown with & are ignored while they are written and become the same as the highest bit shown by a blank while they are read (code extension).
Note 3: The default value of all registers and pre-registers are 0. However, there may be cases that 0 is out of the setting range depending on register.
Note 4: If a 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 commands have pre-registers.
Pre-register is a register to set data for next operation while a motor is operating. The pre-registers of the G9103C are as follows. They operate like FIFO.
Pre-register consists of pre-register for operation (PRMV, PRMVY, PRFL, PRFH, PRUR, PRDR, PRMG, PRDP, PRMD, PRIP, PRIPY, PRUS, PRDS, and PRCI) and for comparator 3 (PRCP3).




Note: Normally, operation data is set through a pre-register. However, operation data can be written into a register directly, when a pre-register is not used (when the data for next operation is not written while a motor is operating.) (G9003 compatibility)

#### 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 (register name starting with “P”).
You do not have to write a pre-register about the register with no change.
The data written during motor control stopping is shifted to register and recognized as a current data. The data written while G9103C is operating motor control become a pre-register data.
This data is fixed by writing a start command (STAFL, STAFH, STAD, and STAUD) and start command for next operation (NSTAFL, NSTAFH, NSTAD, and NSTAUD).

After the current operation is complete, the data is shifted to the register and the operation starts automatically.

You can confirm a status of pre-register using RSTS.PFM bit.
When a pre-register is full (RSTS.PFM = 10b), pre-register is also full (MSTS.SPRF = 1) in main status. If the pre-register is full, writing data is disabled.
In the case that the current operation status is changed (overridden) because of the reason such as a speed change, write a new data can be written to the register.

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

| Process | Pre-register | Register | RSTS. PFM | MSTS. SPRF |
|---|---|---|---|---|
| Default status | 0 (Unfixed) | 0 (Unfixed) | 00 | 0 |
| Write operation data 1 | Operation data 1 (Unfixed) | Operation data 1 (Unfixed) | 00 | 0 |
| Write a start command (A motor starts operation by operation data 1) | Operation data 1 (Unfixed) | Operation data 1 (Fixed) | 01 | 0 |
| Write operation data 2 while the motor is operating | Operation data 2 (Unfixed) | Operation data 1 (Fixed) | 01 | 0 |
| Write a start command for next operation while the motor is operating | Operation data 2 (Fixed) | Operation data 1 (Fixed) | 10 | 1 |
| Operation of operation data 1 is complete. The data is automatically shifted and operation starts by operation data 2. | Operation data 2 (Unfixed) | Operation data 2 (Fixed) | 01 | 0 |
| Operation of operation data 2 is complete. | Operation data 2 (Unfixed) | Operation data 2 (Unfixed) | 00 | 0 |

In the case to set event interrupt cause (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.

Note 1: Please set operation complete timing to “When output pulse cycle is complete (RMD.METM = 0)” if the next operation starts automatically using the pre-register. If you select “When output pulse width is complete (RMD.METM = 1)”, the interval between the last pulse and the initial pulse of the next operation become narrow (750ns). For details, see “8.3.2 Control the output pulse width and operation complete timing”

Note 2: In the case to run continuous operation using pre-registers, you can use start commands for next operation (NSTAFL, NSTAFH, NSTAD, and NSTAUD). Start commands for next operation occurs an error interrupt by writing while stopping. In starting interpolation operation continuously, you can stop motors simultaneously by error interrupt in the case that command to either axis does not in time during operating. This allows us to prevent starting continuous operation while failing simultaneous start.

#### 5.4.1.2 Cancellation of pre-register data for operation

By PRECAN (0026h), STOP (0049h) and SDSTP (004Ah) commands, all pre-register data is cancelled and the status becomes unfixed (RSTS.PFM = 00b).
When pre-register is full (MSTS.SPRF = 1), only the pre-register is canceled and the pre-register can be fixed (RSTS.PMM = 01b) by PRECAN (0026h) command.
The pre-register is canceled by error stop and the pre-register is unfixed (RSTS.PMF = 00b).

#### 5.4.1.3 Writing into pre-register for comparator 3

Comparator 3 (RCMP3) has a pre-register.
Data is written to the register by writing into RCMP3 register and to the pre-register by writing into PRCP3 register.
You can write data for comparator any time, regardless of stopping or operating.
Data is fixed only by writing RCMP3 register or PRCP3 register.

Data for Comparator 3 in the pre-register will be shifted to the register when data become true to false to true after conditions are met.

The status of the pre-register for comparator can be checked by RSTS.PFC bit.
If the pre-register is full (RSTS.PFC = 10b), pre-register in main status is full (MSTS.SPDF = 1).
Writing data to the pre-register is disabled when the pre-register is full.

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

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

The center LSI can output interrupt request signal when comparator 3 become true with setting event interrupt cause (RIRQ.IRC3 = 1).

5.4.1.4 Cancellation of pre-register data for Comparator 3.

By PCPCAN (0027h) command, the pre-register data is cancelled and the status becomes unfixed (RSTS.PFC = 00b). However, please note that RCMP3 register is not made unfixed.

To return to unfixed (RSTS.PFC = 00b), use PFCCAN (002Dh) command.

### 5.4.2 Register for setting speeds

These registers are used to set the operating speeds.

| Register | Description | Bit length | Set range | R/W |
|---|---|---|---|---|
| RFL(PRFL) | FL speed setting | 17 | 1 ~ 100,000 (186A0h) | R/W |
| RFH(PRFH) | FH speed setting | 17 | 1 ~ 100,000 (186A0h) | R/W |
| RUR(PRUR) | Acceleration rate setting | 16 | 1 ~ 65,535 (FFFFh) | R/W |
| RDR(PRDR) | Deceleration rate setting | 16 | 1 ~ 65,535 (FFFFh) | R/W |
| RMG(PRMG) | Speed magnification setting | 11 | 2 ~ 2,047 (7FFh) | R/W |
| | Get ID code | 16 | - | |
| RDP(PRDP) | Ramping-down point setting | 24 | −8,388,608 (80000h) ~ +8,388,607 (7FFFFFh)<br>0 ~ 16,777,215 (FFFFFFh) | R/W |
| RUS(PRUS) | Acceleration S-curve range setting | 16 | 1 ~ 50,000 (C350h) | R/W |
| RDS(PRDS) | Deceleration S-curve range setting | 16 | 1 ~ 50,000 (C350h) | R/W |
| RFA | FA speed setting | 17 | 1 ~ 100,000 (186A0h) | R/W |

Please note that there may be cases that 0 is output the setting range depending registers. For details, see "7.2 Speed pattern settings".

#### 5.4.2.1 RFL (PRFL): FL speed setting register (17 bits)

[WPRFL: 00B1h, RPRFL: 00C1h]
[WRFL: 0091h, RRFL: 00D1h]

This register is used to specify initial speed (stopping speed) in high-speed operation (with acceleration and deceleration). PRFL register 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| * | * | * | * | * | * | * | * | * | * | * | * | * | * | * | | | | | | | | | | | | | | | | | |

While a motor is operating, the speed can be overridden by changing RFL register.
Specify a start speed (Stop speed) of FL constant operation and high-speed operation (with acceleration and deceleration) within the range 1 ~ 100,000(186A0h). The range 100,000 ~ 131,071 (186A0h ~ 1FFFFh) is replaced as 100,000. The actual operation speed is calculated by the formula with a setting value of RMG register

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

[WPRFH: 00B2h, RPRFH: 00C2h]
[WRFH: 0092h, RRFL: 00D2h]

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 a motor is operating, the speed can be overridden by changing RFH register. Specify the speed of FH constant operation and operation speed of high-speed operation (with acceleration and deceleration) within the range 1 ~ 100,000 (186A0h). The range 100,000 ~ 131,071 (186A0h ~ 1FFFFh) is replaced as 100,000. In the case of high-speed operation (with acceleration and deceleration), specify a larger value than a setting value of RFL register. The actual operation speed is calculated by the formula with a setting value of RMG register.

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

[WPRUR: 00B3h, RPRUR: 00C3h]
[WRUR: 0093h, RRUR: 00D3h]

This register is used to set an acceleration rate. PRDR register 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 characteristic of acceleration in the case of the high-speed operation (with acceleration and deceleration) within the range of 1 ~ 65,535 (FFFFh).

5.4.2.4 RDR (PRDR): Deceleration rate setting register (16 bits)

[WPRDR: 00B4h, RPRDR: 00C4h]
[WRDR: 0094h, RRDR: 00D4h]

This register is used to set a deceleration rate. PRDR register 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 characteristic of deceleration in the case of the high-speed operation (with acceleration and deceleration) within the range 1 ~ 65,535 (FFFFh). Even when “Automatic setting” (RMD.MSDP = 0) is selected for the ramping-down point, the value set in RDR register is be used as a deceleration rate. When RDR is set to 0, deceleration rate refers to RUR register value.

5.4.2.5 RMG (PRMG): Magnification rate register (11-bit)

[WPRMG: 00B5h, RPRMG: 00C5h]
[WRMG: 0095h, RRMG: 00D5h]

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

ID code can be read out from the upper 16 bits.
For details, see “8.20 ID code confirmation”.

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

| Bit | Bit name | Description |
|---|---|---|
| 10 ~ 0 | MG | Speed magnification setting bit |
| 15 ~ 11 | Undefined | (Always 0) |
| 31 ~ 16 | IDCD | ID code bit<br>0 can be read normally. Only when RMG (00D0h) command is used following IDMON (0003h) command, ID code can be read. (Writing is disabled). * There is not PRMG.IDCD bit. |

RMG.MG bit specifies the relationship between setting values of PFL, RFH, RFA registers and speed, within the range of 2 ~ 2,047. The higher the magnification rate is, the coarser the speed steps selected is. Normally, use as small a magnification rate as possible. The operation speed [pps] is the product of multiplying the speed magnification by a speed register value.

5.4.2.6 (RDP) PRDP: Ramping-down point setting register (24 bits)

[WPRDP: 00B6h, RPRDP: 00C6h]
[WRDP: 0096h, RRDP: 00D6h]

This register is used to set a ramping-down point (with deceleration start point) for positioning operations. PRDP register 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 a deceleration start point in positioning operation of high seed operation (with acceleration/deceleration operation).
Bits shown with "#" symbol are ignored when they are written and setting meaning varies according to the setting of RMD.MSDP bit when these bits are read.

| RMD.MSDP | Setting details | “#” Bit |
|---|---|---|
| 0 | Offset for automatically set values. (−8,388,608 ~ +8,388,607)<br>When a positive value is entered, a motor will start deceleration earlier and the FL speed range will be longer.<br>When a negative value is entered, a motor will start deceleration later and the speed will not reach the FL speed. | Same as bit 23. |
| 1 | When number of remaining pulses drops to less than a set value, the motor starts to decelerate. (0 ~ 16,777,215) | 0 |

5.4.2.7 PRUS (RUS): Acceleration S-curve range setting register (16 bits)

[WPRUS: 00B9h, RPRUS: 00C9h]
[WRUS: 0099h, RRUS: 00D9h]

This register is used to specify the S-curve range of the S-curve acceleration.
PRUS register 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 an S-curve acceleration value for an S-curve acceleration/deceleration operation, within the range of 1 ~ 50,000 (C350h).
All values from 50,000 ~ 65,535 (0C350h ~ 0FFFFh) are replaced as 50,000.
S-curve acceleration range $S_{SU}$ is calculated by the formula with a RMG register value.
If "0" is entered, the G9103C substitutes a 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 RDS (PRDS): Deceleration S-curve range setting register (16 bits)

[WPRDS: 00BAh, RPRDS: 00CAh]
[WRDS: 009Ah, RRDS: 00DAh]

This register is used to specify the S-curve range of the S-curve deceleration. PRDS register 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 an S-curve deceleration value for an S-curve acceleration/deceleration operation, within the range of 1 ~ 50,000 (C350h).
All values from 50,000 ~ 65,535 (0C350h ~ 0FFFFh) are replaced as 50,000.
The S-curve deceleration range $S_{SD}$ is calculated by the formula with a RMG register value.
If "0" is entered, the G9103C 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-bit)

[WRFA: 009Bh, RRFA:00DBh]

This register is used to set a constant speed for backlash correction.
This value is also used as reverse constant speed during an origin return 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| * | * | * | * | * | * | * | * | * | * | * | * | * | * | * | | | | | | | | | | | | | | | | | |

Set a correction speed (FA speed) to feed a specific amount during backlash correction, within the range of 1 ~ 100,000 (186A0h).
ALL values from 100,000 ~ 131,071 (186A0h ~ 1FFFFh) are replaced as 100,000.
The actual operation speed is be calculated by the formula with RMG register value.

### 5.4.3 Feed amount setting registers

These registers are used to set feed amount.

| Register | Description | Bit length | Set range | R/W |
|---|---|---|---|---|
| RMV(PRMV) | Positioning amount | 28 | −134,217,728 (8000000h) ~ +134,217,727 (7FFFFFFh) | R/W |
| RMVY(PRMVY) | Axis Y positioning amount in interpolation | 28 | −134,217,728 (8000000h) ~ +134,217,727 (7FFFFFFh) | R/W |
| RIP(PRIP) | X-coordinate of the center in circular interpolation | 28 | −134,217,728 (8000000h) ~ +134,217,727 (7FFFFFFh) | R/W |
| RIPY(PRIPY) | Y-coordinate of the center in circular interpolation | 28 | −134,217,728 (8000000h) ~ +134,217,727 (7FFFFFFh) | R/W |
| RCI(PRCI) | Number of stepping in circular interpolation | 31 | 0 ~ +2,147,483,647 (7FFFFFFFh) | R/W |
| RCIC | Stepping counter in circular interpolation | 31 | 0 ~ +2,147,483,647 (7FFFFFFFh) | R |

#### 5.4.3.1 RMV (PRMV): Feed amount for positioning setting register (28 bits)

[WPRMV: 00B0h, RPRMV: 00C0h]
[WRMV: 0090h, RRMV: 00D0h]

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

Meaning of setting value may vary according to operation mode.
Setting range is −134,217,728 ~ +134,217,727.
By changing RMV register while in positioning operation, feed amount can be overridden.
For details, see “8.2.1 Target position override 1”.

5.4.3.2 RMVY (PRMVY): Positioning amount setting register for Y-coordinate in interpolation (28 bits)

[WPRMVY: 00BDh, RPRMVY: 00CDh]
[WRMVY: 008Dh, RRMVY: 00FDh]

In operation of linear interpolation and circular interpolation, specify Y-coordinate of a target position with an incremental value.
PRMVY register 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 is −-134,217,728 ~ +134,217,727.

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

[WPRIP: 00B8h, RPRIP: 00C8h]
[WRIP: 0098h, RRIP: 00D8h]

In circular interpolation, specify the center of X-coordinate with an incremental value
PRIP register 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 is −134,217,728 ~ +134,217,727.

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

[WPRIPY: 00BEh, RPRIPY: 00CEh]
[WRIPY: 008Eh, RRIPY: 00FEh]

In circular interpolation, specify the center of Y-coordinate with an incremental value
PRIPY register 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 is –134,217,728 ~ +134,217,727.

5.4.3.5 RCI (PRCI): Setting register for number of steps necessary to complete circular interpolation (31 bits)

[WPRCI: 00BCh,RPRCI: 00CCh]
[WRCI: 008Ch, RRCI: 00FCh]

Specify a number of steps necessary to complete circular interpolation operation.
PRCI register is the pre-register for RCI.

| 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 is 0 ~ +2,147,483,647. (Set more than 1 without fail when ramping-down point auto setting function is used.)

In circular interpolation mode, setting can be omitted when acceleration / deceleration, a remaining start command or comparator functions are not used.
For calculation of setting value, see “6.7.3.2 Number of stepping in circular interpolation”.

5.4.3.6 RCIC: Step counter for circular interpolation (31 bits)

[RRCIC: 00FBh]

This register is used to read step counter value for circular interpolation (read only).
This is also used as a counter to read remaining pulse in circular and linear interpolation mode.

| 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 is 0 ~ +2,147,483,647.

Step counter for circular interpolation copies RCI register value in circular interpolation.
Step counter for circular interpolation counts down to 0 every step of circular interpolation.

Remaining pulse counter copies RCI register value in circular interpolation mode.
In linear interpolation mode, it copies a bigger absolute value (long axis) after comparing with absolute values of RMV register value or RMVY register value.
Remaining pulse counter in circular interpolation mode counts down every step of circular interpolation or every pulse output on a long axis.

Note : Remaining pulse counter (RCIC) in circular interpolation mode regards counter values that are more than 134,217,727 as 134,217,727 and compares them with comparator 3 (RCMP3) internally. Therefore, if “RCMP3 comparison counter” is selected as comparison conditions with setting the maximum value (134,217,727) to RCMP3 register, comparison condition is always true in the case that RCIC register value is more than 134.217,727.

### 5.4.4 Environmental setting registers

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

| Register | Name | Bit length | Set range | R/W |
|---|---|---|---|---|
| RMD(PRMD) | Set operation mode | 31 | - | R/W |
| RENV1 | Environment setting 1 | 32 | - | R/W |
| RENV2 | Environment setting 2 | 29 | - | R/W |
| RENV3 | Environment setting 3 | 31 | - | R/W |
| RENV4 | Environment setting 4 | 32 | - | R/W |
| RENV5 | Environment setting 5 | 32 | - | R/W |
| RENV6 | Environment setting 6 | 32 | - | R/W |
| RCUN1 | COUNTER 1 | 28 | −134,217,728 (8000000h) ~ +134,217,727 (7FFFFFFh) | R/W |
| RCUN2 | COUNTER 2 | 28 | −134,217,728 (8000000h) ~ +134,217,727 (7FFFFFFh) | R/W |
| RCUN3 | COUNTER 3 | 16 | −32,768 (8000h) ~ +32,767 (7FFFh) | R/W |
| RCMP1 | Comparison data for Comparator 1 | 28 | −134,217,728 (8000000h) ~ +134,217,727 (7FFFFFFh) | R/W |
| RCMP2 | Comparison data for Comparator 2 | 28 | −134,217,728 (8000000h) ~ +134,217,727 (7FFFFFFh) | R/W |
| RCMP3 (PRCP3) | Comparison data for Comparator 3 | 28 | −134,217,728 (8000000h) ~ +134,217,727 (7FFFFFFh) | R/W |
| RIRQ | Set event interrupt causes | 17 | −134,217,728 (8000000h) ~ +134,217,727 (7FFFFFFh) | R/W |
| RCUN1 | COUNTER 1 latched data | 28 | −134,217,728 (8000000h) ~ +134,217,727 (7FFFFFFh) | R/W |
| RCUN2 | COUNTER 2 latched data | 28 | −134,217,728 (8000000h) ~ +134,217,727 (7FFFFFFh) | R/W |
| RCUN3 | COUNTER 3 latched data | 17 | −32,768 (08000h) ~ +32,767 (7FFFh)<br>0 ~ 100,000 (186A0h) | R/W |
| RSTS | Extension status | 32 | - | R |
| REST | Error interrupt status | 23 | - | R/W |
| RIST | Event interrupt status | 17 | - | R/W |
| RPLS | Positioning counter | 28 | 0 ~ +134,217,728 (8000000h) | R |
| RSPD | Current speed monitor | 17 | 1 ~ 100,000 (186A0h) | R |
| | EZ counter | 4 | 0 ~ 15 | |
| | Idling count | 3 | 0 ~ 7 | |
| RSDC | Ramp down value by automatic calculation | 24 | 0 ~ 16,777,215 (FFFFFFh) | R |
| RSYN | Synchronizing control | 32 | - | R/W |
| RSYN2 | Synchronizing control 2 | 28 | - | R/W |
| RMEC | Receiving error counter | 24 | - | R |
| RGN0 | General-purpose register 0 (Unit ID manager) | 32 | - | R/W |
| RGN1 | General-purpose register 1 | 32 | - | R/W |
| RGN2 | General-purpose register 2 | 32 | - | R/W |
| RGN3 | General-purpose register 3 | 32 | - | R/W |

5.4.4.1 RMD (PRMD): Operation mode setting register

[WPRMD: 00B7h,RPRMD: 00C7h]
[WRMD: 0097h, RRMD: 00D7h]

This register is used to set operation mode. PRMD is the pre-register for RMD.

| 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 |
|---|---|---|
| 6~0 | MOD | These bits are to select operation mode<br>000 0000 (00h): (+) direction continuous operation controlled by command control.<br>000 1000 (08h): (−) direction continuous operation controlled by command control.<br>000 0001 (01h): Continuous operation controlled by pulse generator (PA, PB) input.<br><br>001 0000 (10h): Origin return operation in (+) direction.<br>001 1000 (18h): Origin return operation in (−) direction.<br>001 0010 (12h): Leaving the origin position in (+) direction.<br>001 1010 (1Ah): Leaving the origin position in (−) direction.<br>001 0101 (15h): Origin search in (+) direction<br>001 1101 (1Dh): Origin search in (−) direction<br><br>010 0000 (20h): Feed to +EL or +SL (software limit) position.<br>010 1000 (28h): Feed to −EL or −SL (software limit) position.<br>010 0010 (22h): Leaving from −EL or −SL (software limit) position.<br>010 1010 (2Ah): Leaving from +EL or +SL (software limit) position.<br><br>010 0100 (24h): Feed in (+) direction for a specified number of EZ counts.<br>010 1100 (2Ch): Feed in (−) direction for a specified number of EZ counts.<br><br>100 0001 (41h): Positioning operation (specify an incremental target position)<br>100 0010 (42h): Positioning operation (specify an absolute position in COUNTER 1)<br>100 0011 (43h): Positioning operation (specify an absolute position in COUNTER 2)<br>100 0100 (44h): Zero return of command position (COUNTER 1).<br>100 0101 (45h): Zero return of mechanical position (COUNTER 2).<br>100 0110 (46h): Single pulse operation in (+) direction.<br>100 1110 (4Eh): Single pulse operation in (−) direction.<br>100 0111 (47h): Timer operation<br><br>101 0001 (51h): Positioning operation synchronized with PA and PB.<br>(Specify a target incremental position)<br>101 0010 (52h): Positioning operation synchronized with PA and PB<br>(Specify an absolute position of COUNTER 1)<br>101 0011 (53h): Positioning operation synchronized with PA and PB<br>(Specify an absolute position of COUNTER 2)<br>101 0100 (54h): Zero return of a command position synchronized with PA and PB.<br>101 0101 (55h): Zero return of a mechanical position synchronized with PA and PB.<br><br>110 0000 (60h): Continuous linear interpolation (output X axis pulses).<br>110 0001 (61h): Linear interpolation (output X axis pulses).<br>110 0100 (64h): Circular interpolation in CW direction (output X axis pulses).<br>110 0101 (65h): Circular interpolation in CCW direction (output X axis pulses).<br>110 1000 (68h): Continuous linear interpolation synchronized with PA and PB (output X axis pulses).<br>110 1001 (69h): Linear interpolation synchronized with PA and PB (output X axis pulses). |

| Bits | Bit name | Description |
|---|---|---|
| | | 110 1100 (6Ch): Circular interpolation in CW direction synchronized with PA and PB (output X axis pulses).<br>110 1101 (6Dh): Circular interpolation in CCW direction synchronized with PA and PB (output X axis pulses).<br><br>111 0000 (70h): Continuous linear interpolation (output Y axis pulses).<br>111 0001 (71h): Linear interpolation (output Y axis pulses).<br>111 0100 (74h): Circular interpolation in CW direction (output Y axis pulses).<br>111 0101 (75h): Circular interpolation in CCW direction (output Y axis pulses).<br>111 1000 (78h): Continuous linear interpolation synchronized with PA and PB (output Y axis pulses).<br>111 1001 (79h): Linear interpolation synchronized with PA and PB (output Y axis pulses).<br>111 1100 (7Ch): Circular interpolation in CW direction synchronized with PA and PB (output Y axis pulses).<br>111 1101 (7Dh): Circular interpolation in CCW direction synchronized with PA and PB (output Y axis pulses). |
| 7 | MENI | Set function to occur stop interrupt (MSTS.SEND) while a pre-register for operation is fixed.<br>0: Enabled.<br>1: Disabled.<br>While pre-register for operation is fixed, stop interrupt does not occur.<br>(MSTS.SEND does not change to 1) |
| 8 | MSDE | Set the function to input SD signal to<br>0: Disabled. Terminal condition can be checked with RSTS.SDIN bit.)<br>1: Enabled. Decelerates (or decelerates and stops) by SD signal turning ON. |
| 9 | MINP | Set the function to input INP signal to<br>0: Disabled. Terminal condition can be checked with RSTS.SINP bit.)<br>1: Enabled. Delay operation complete until INP signal turns ON. |
| 10 | MSMD | Select acceleration / deceleration characteristics.<br>0: Linear acceleration/deceleration<br>1: S-curve acceleration/deceleration |
| 11 | METM | Specify operation complete timing.<br>0: When output pulse cycle is complete.<br>1: When output pulse width is complete.<br>*When using vibration reduction function, "When output pulse width is complete" is selected even if "when output pulse cycle is complete" (RMD.METM = 0) is selected. |
| 12 | MSDP | Select a method to set ramping-down point for high speed operation.<br>0: Automatic setting<br>1: Manual setting.<br>*This setting is enabled only in positioning operations. |
| 13 | MPCS | Set the function to input PCS signal.<br>0: Disabled.<br>1: Enabled. After PCS signal turns ON, G9103C controls number of pulses.<br>*This setting is enabled only in positioning operations.<br>See "8.2.2 Target position override 2 (PCS signal)". |
| 14 | MSY | Set the feature to hold start to<br>0: Disabled.<br>1: Enabled.<br>After writing a start command, the G9103C starts operation by turning STA signal ON, or writing CMSTA(0006h) or SPSTA(002Ah) commands or when the group number is correspond to the value set in the start command of broadcast communication. |
| 15 | MSPE | Set the feature to input STP signal to<br>0: Disabled<br>1: Enabled.<br>Decelerate and stop or stop a motor immediately by turning STP signal ON, or by writing CMSTP (00007h) or when the group number corresponds to the value set in the stop command of broadcast communication. |
| 16 | MSPO | Set to output STP signal at abnormal emergent stop to<br>0: Disabled.<br>1: Enabled. |

| Bits | Bit name | Description |
|---|---|---|
| 17 | MADJ | Set FH correction function.<br>0: Enabled.<br>1: Disabled. |
| 18 | MMSK | Set the function to mask command pulses to<br>0: Disabled.<br>1: Enabled. Does not output command pulses. |
| 19 | MINT | Set the feature of interrupt request (MSTS.SINT) to<br>0: Enabled.<br>1: Disabled. Interrupt request is fixed to 0.<br>Statuses of operation stop interrupt, error interrupt, event interrupt are changed. |
| 20 | MPH | Select output signal of the following terminals: BSY/PH1, FUP/PH2, FDW/PH3, and MVC/PH4.<br>0: Output BSY, FUP, FDW and MVC.<br>1: Output PH1, PH2, PH3, and PH4. |
| 21 | MMPH | Set output mask function of PH1, PH2, PH3, and PH4 signals to<br>0: Disabled. Output a level selected by RMD.MBIM bit.<br>1: Enabled. Output PH1, PH2, PH3, PH4 signals. |
| 22 | MUB | Select a driving method for PH1, PH2, PH3, and PH4 signals.<br>0: Driving method for a 2-phase unipolar motor.<br>1: Driving method for a 2-phase bipolar motor. |
| 23 | MFH | Select excitation sequence for PH1, PH2, PH3, and PH4 signals.<br>0: 2-phase excitation method<br>1: 1-2 phase excitation method |
| 24 | MIOR | Set monitor function of general-purpose input/output data (IOPIB) to<br>0: Enabled.<br>1: Disabled. The corresponding bit of port 2 is 0 regardless of condition of output terminals.<br>*When making “Input change interrupt” of a center LSI by change of output terminal disabled, set this disabled. |
| 25 | MBIM | Select output mask status of PH1, PH2, PH3 and PH4.<br>0: Output “LLLL” level<br>1: Output “LLHH” level<br>*This setting is enabled when RMD.MMPH = 0. |
| 26 | MSDC | Select a method to set ramping-down point automatically to<br>0: Calculating method.<br>In addition to the case that acceleration time and deceleration time is same, correct setting can be made in the case that they are different.<br>The error from the calculated values may be bigger than 1 pulse at 0.01% frequency. Please note that when you use a speed pattern with a bit number of deceleration pulses.<br>Count method is selected when function to make synthesized speed constant is enabled (RMD.MIPF = 1) in interpolation.<br>In this case, match acceleration time with deceleration time.<br>1: Count method<br>Enabled only when acceleration time is the same as deceleration time. In the different case, setting cannot be made correctly. The error from a calculated value will be smaller than 1 pulse.<br>*See “7.2 Speed pattern settings”. |
| 27 | MIPF | Set the function to make synthesized speed constant in interpolation operation regarded as 2 axis interpolation operation.<br>1: Disabled.<br>2: Enabled.<br>*See “6.7 Interpolation operation mode”. |
| 28 | MERI | Set the function to receive stop request when other axis stops by error.<br>0: Disabled.<br>1: Enabled.<br>*See “8.17.2 Simultaneous stop function”. |

| Bits | Bit name | Description |
|---|---|---|
| 29 | MERO | Set the function to send stop request when own axis stops by error.<br>0: Disabled.<br>1: Enabled.<br>*See “8.17.2 Simultaneous stop function”. |
| 30 | MDMY | Set the function of dummy operation to mask command pulse output and not count.<br>0: Disabled.<br>1: Enabled.<br>*This setting is used in the case that interpolated axis will be changed in interpolation operation using pre-registers. |
| 31 | Not defined | (Always set to 0.) |

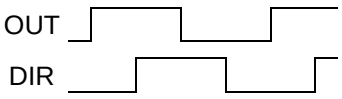
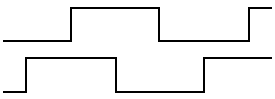
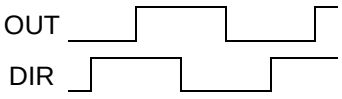
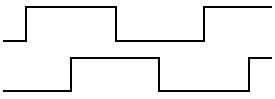
5.4.4.2 Environmental setting 1 register

[WRENV1: 009Ch, RRENV1: 00DCh]

This register is used for environmental 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 | EPW | | | EROR | EROE | ALML | ALMM | ORGL | SDL | SDLT | SDM | ELM | PMD | | |

| 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 | CLR | | STPM | STAM | ETW | |

<table>
<tr><th>Bit</th><th>Bit name</th><th>Description</th></tr>
<tr><td>2 ~ 0</td><td>PMD</td><td>These bits are to select an output pulse mode. (The following is an example when the function to reverse motor rotation direction is disabled (RENV1.MREV = 0)).
<table>
<tr><th rowspan="2">PMD</th><th colspan="2">Operation in (+) direction</th><th colspan="2">Operation in (−) direction</th></tr>
<tr><th>OUT signal</th><th>DIR signal</th><th>OUT signal</th><th>DIR signal</th></tr>
<tr><td>000</td><td>  </td><td>High</td><td>  </td><td>Low</td></tr>
<tr><td>001</td><td>  </td><td>High</td><td>  </td><td>Low</td></tr>
<tr><td>010</td><td>  </td><td>Low</td><td>  </td><td>High</td></tr>
<tr><td>011</td><td>  </td><td>Low</td><td>  </td><td>High</td></tr>
<tr><td>100</td><td>  </td><td>High</td><td>High</td><td>  </td></tr>
<tr><td>101</td><td colspan="2">  </td><td colspan="2">  </td></tr>
<tr><td>110</td><td colspan="2">  </td><td colspan="2">  </td></tr>
<tr><td>111</td><td>  </td><td>Low</td><td>Low</td><td>  </td></tr>
</table>
When the function to reverse motor rotation direction is enabled (RENV1.MREV = 1), output pulse modes in (+) direction and operation in (−) direction are switched. 000b, 001b, 010b, 011b are common pulse mode, 100b, 111b are 2 pulse mode, and 101b, 110b are 90 degree phase difference mode. One cycle of 90 degree phase difference corresponds to 4x (4 pulse) of common pulse mode and 2 pulse mode.</td></tr>
<tr><td>3</td><td>ELM</td><td>Select process when +EL and –EL signals turn ON.<br>0: Stop immediately.<br>1: Decelerate and stop.<br>*When “Decelerate and stop (RENV1.ELM = 1)” is selected as a process made when +EL and –EL signals turn ON, a motor starts to decelerate when the signal turns ON. Therefore, the machine goes past the position turning ON and stops. Please pay attention to collision to mechanical systems.</td></tr>
<tr><td>4</td><td>SDM</td><td>Select process when SD signal turn ON.<br>0: Only decelerate.<br>1: Decelerate and stop.</td></tr>
<tr><td>5</td><td>SDLT</td><td>Set the function to input SD signal.<br>0: Disabled.<br>1: Enabled.<br>*SD latch signal is reset when starting at turning SD signals OFF and input latch function of SD signal is set to disabled (RENV1.SDLT = 0).</td></tr>
</table>

| Bits | Bit name | Description |
|---|---|---|
| 6 | SDL | Select input logic of SD signal.<br>0: Negative logic.<br>1: Positive logic. |
| 7 | ORGL | Select input logic of ORG signal.<br>0: Negative logic.<br>1: Positive logic. |
| 8 | ALMM | Select processing when ALM signals turn ON.<br>0: Stop immediately.<br>1: Decelerate and stop. |
| 9 | ALML | Select input logic of ALM signal.<br>0: Negative logic.<br>1: Positive logic. |
| 10 | EROE | Set the function to output ERC signal automatically when motor stops by error.<br>0: Disabled<br>1: Enabled. G9103C outputs an ERC signal automatically when the movement on the axis is stopped immediately by +EL, −EL, ALM, or EMG input signals. However, G9103C does not output the ERC signal when a motor decelerate and stop. Even in the operation mode that EL signal is specified as a normal stop (RMD.MOD = 20h, 28h), G9103C outputs ERC signal automatically when a motor stops immediately. |
| 11 | EROR | Set the function to output ERC signal automatically at completion of origin return operation.<br>0: Disabled<br>1: Enabled |
| 14 ~ 12 | EPW | In the case of ERC signal output (RENV1.CDWN = 0), “ERC signal pulse width” is selected. In the case of CDWN signal output (RENV1.CDWN = 1), “Current recovery” is selected.<br><br>1. ERC signal pulse width (when RENV1.CDWN = 0)<br>000: 12.5μs 001: 100μs 010: 400 μs 011:1.6 ms<br>100: 13 ms 101: 51 ms 110: 102 ms 111: Level output<br><br>2. Current recovery time (when RENV1.CDWN = 1)<br>000: 6.4 ms 001: 13 ms 010: 25.6 ms 011:51 ms<br>100: 102 ms 101: 205 ms 110: 410 ms 111: Prohibited setting<br><br>There are error as much as ±4%. |
| 15 | ERCL | Select output logic of ERC signal.<br>0: Negative logic.<br>1: Positive logic. |
| 17, 16 | ETW | Select OFF timer time of ERC signal in the case of ERC signal output (RENV1.CDWN = 0).<br>Select current down delay time in the case of CDWN signal output (RENV1.CDWN = 1).<br><br>1. OFF timer time of ERC signal (RENV1.CDWN = 0)<br>00: 0 μs 01: 13 μs 10: 1.6 ms 11: 102 ms<br><br>2. Current down delay time (RENV1.CDWN = 1)<br>00: 51 ms 01: 102 ms 10: 205 ms 11: 410 ms |
| 18 | STAM | Select input type of STA signal<br>0: Level trigger<br>1: Edge trigger (falling edge) |
| 19 | STPM | Select processing made when STP signal turns ON.<br>0: Stop immediately.<br>1: Decelerate and stop |

| Bits | Bit name | Description |
|---|---|---|
| 21, 20 | CLR | Select input type of CLR signal.<br>00: Falling edge<br>01: Rising edge<br>10: L level.<br>11: H level. |
| 22 | INPL | Select input logic of INP signal.<br>0: Negative logic<br>1: Positive logic |
| 23 | LTCL | Select input type of LTC signal.<br>0: Falling edge<br>1: Rising edge. |
| 24 | PCSL | Select input logic of PCS signal.<br>0: Negative logic<br>1: Positive logic |
| 25 | FLTR | Set input filter function of +EL −EL SD, ORG, ALM and IMIP signals.<br>0: Enabled. Pulses shorter than 4 μs will be ignored.<br>1: Disabled. |
| 26 | DTMF | Set direction change timer.<br>0: Enabled.<br>1: Disabled<br>See “8.3.1 Output pulse mode”. |
| 27 | SEDM | Set the function that interrupt request turns ON (MSTS.SINT = 1) by operation stop interrupt ON (MSTS.SEND = 1)<br>0: Enabled. Operation stop interrupt ON makes interrupt request ON.<br>1: Disabled. Operation stop interrupt ON does not make interrupt request ON. |
| 28 | SEDR | Set the function to reset operation stop interrupt (MSTS.SEND = 1) when a start command is written.<br>0: Disabled<br>1: Enabled. When a start command (0050h ~ 005Fh) is written, G9103C makes operation stop interrupt OFF.<br>* In the case that a start command is written at the moment that operation stop interrupt occurs (MSTS.SEND = 1), operation stop interrupt may be reset (MSTS.SEND = 0) before interrupt request signal is confirmed.<br>In the case to use pre-register in operating, pay attention to that. |
| 29 | PDTC | Set the function to control output pulse width.<br>0: Enabled. When output speed of command pulse is below about 2.44kpps, output pulse width is fixed to about 200 μs.<br>1: Disabled Output pulse width is fluctuated to about 50% duty ratio regardless of output speed of command pulse. |
| 30 | MREV | Set the function to reverse motor rotation<br>0: Disabled<br>1: Enabled. Change motor rotation direction to a reverse direction of output pulse mode. |
| 31 | CDWN | Select output signal of ERC/SDWN terminal<br>0: ERC signal<br>1: CDWN signal |

5.4.4.3 RENV2: Environmental setting 2 register

[WRENV2: 009Dh, RRENV2: 00DDh]

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

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PIM | | PINF | EZL | EDIR | EIM | | 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 | IMSK | EXER | GN | | | SIFM | IDL | | | ROMB | POFF | EOFF | PDIR |

| Bit | Bit name | Description |
|---|---|---|
| 0 | P0M | Select function of P0 terminal.<br>0: General-purpose input<br>1: General-purpose output |
| 1 | P1M | Select function of P1 terminal.<br>0: General-purpose input<br>1: General-purpose output |
| 2 | P2M | Select function of P2 terminal.<br>0: General-purpose input<br>1: General-purpose output |
| 3 | P3M | Select function of P3 terminal.<br>0: General-purpose input<br>1: General-purpose output |
| 4 | P4M | Select function of P4 terminal.<br>0: General-purpose input<br>1: General-purpose output |
| 5 | P5M | Select function of P5 terminal.<br>0: General-purpose input<br>1: General-purpose output |
| 6 | P6M | Select function of P6 terminal.<br>0: General-purpose input<br>1: General-purpose output |
| 7 | P7M | Specify function of P7 terminal.<br>0: General-purpose input<br>1: General-purpose output |
| 8 | EINF | Set input filter to EA, EB and EZ signal.<br>0: Enabled. Pulse shorter than 150 ns will be ignored.<br>1: Disabled. |
| 10, 9 | EIM | Select input type of EA and EB signal.<br>00: Multiply a 90 degree phase difference by 1<br>(Count forward when EA signal phase is ahead.)<br>01: Multiply a 90 degree phase difference by 2<br>(Count forward when EA signal phase is ahead.)<br>10: Multiply a 90 degree phase difference by 4<br>(Count forward when EA signal phase is ahead.)<br>11: Count forward when EA signal rises and count backward when the EB signal rises. |
| 11 | EDIR | Set function to reverse count direction of EA and EB signals.<br>0: Disabled.<br>1: Enabled. |
| 12 | EZL | Select input type of EZ signal.<br>0: Falling edge<br>1: Rising edge |
| 13 | PINF | Set input filter to the PA and PB signals to<br>0: Enabled. Pulse shorter than 150 ns will be ignored.<br>1: Disabled. |

| Bits | Bit name | Description |
|---|---|---|
| 15, 14 | PIM | Select input type of PA and PB signal.<br>00: Multiply a 90 degree phase difference by 1<br>(Count forward when PA signal phase is ahead.)<br>01: Multiply a 90 degree phase difference by 2<br>(Count forward when PA signal phase is ahead.)<br>10: Multiply a 90 degree phase difference by 4<br>(Count forward when PA signal phase is ahead.)<br>11: Count forward when PA signal rises and count backward when the PB signal rises. |
| 16 | PDIR | Set function to reverse count direction of PA and PB signals.<br>0. Disabled.<br>1: Enabled. |
| 17 | EOFF | Set function to input of EA and EB signals.<br>0: Enabled.<br>1: Disabled. (EZ signal input is enabled.) |
| 18 | POFF | Set function to input of PA and PB signals.<br>0: Enabled.<br>1: Disabled. |
| 19 | ROMB | Monitor access status of EEPROM connected externally.<br>0: Not accessing<br>1: Accessing<br>*This bit is only for read. When it is written, it will be ignored. |
| 22 ~ 20 | IDL | Set number of idling pulses (0 ~ 6 pulses)<br>0 : Disabled.<br>1~7: n – 1 pulse. “n” is a setting value. |
| 23 | SIFM | Select function of terminals P4 ~ P7.<br>0: General-purpose I/O terminals<br>1: Terminals for serial bus control |
| 26 ~ 24 | GN | Select a group number to be used for broadcast communication<br>000 : Status GRP terminal is used.<br>001~ 111: The setting value of RENV2.GN is used. (e.g. “100b” = 4) |
| 27 | EXER | Select output pulse width of MRER signal.<br>0: 3.2 µs<br>1: Approximately 100 ms |
| 28 | IMSK | Set the function to reset interrupt request for maximum one cycle of cyclic communication so that a center LSI makes “Input change interrupt” occurs at reset of interrupt cause.<br>0: Disabled.<br>1: Enabled. |
| 31 ~ 29 | Not defined | (Always set to 0.) |

5.4.4.4 RENV3: Environmental setting 3 register

[WRENV3: 009Eh, RRENV3: 00DEh]

This register is used for Environmental setting 3. The main functions are to set origin return methods and counter operation specifications

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | BSYC | CI3 | | | CI2 | | EZD | | | | ORM | | | |

| 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 | G03B | CU3C | CU2C | CU1C |

| Bit | Bit name | Description |
|---|---|---|
| 3 ~ 0 | ORM | Select an origin return method.<br>0000: Origin return operation 0<br>- When ORG signal turns ON, a motor will stop immediately at constant start or decelerate and stop at high speed start.<br>- COUNTER clear timing: When ORG signal turns ON.<br><br>0001: Origin return operation 1<br>- After ORG signal turns ON, a motor will stop immediately at constant start or decelerate and stop at high speed start. Then, it feeds in the opposite direction at FA constant speed until ORG signal turns OFF. Then, the motor will move back in the original direction at FA speed and stop immediately when ORG signal turns ON again.<br>- COUNTER clear timing: When ORG signal turns ON after move at FA speed.<br><br>0010: Origin return operation 2<br>- A motor operates until ORG signal turns ON, a motor will stop immediately at constant start when the LSI finish counting the specified number of EZ pulses. At high speed start, the motor decelerates during counting and stops when the LSI finishes counting the specified number of EZ pulses.<br>COUNTER clear timing: When finishing counting the specified number of EZ pulses.<br><br>0011: Origin return operation 3<br>- A motor operates until ORG signal turns ON. A motor will stop immediately at constant start or decelerate and stop at high speed start when the LSI finishes counting the specified number of EZ pulses,<br>COUNTER clear timing: When finishing counting the specified number of EZ pulses.<br><br>0100: Origin return operation 4<br>- After the ORG signal turns ON, a motor will stop immediately at constant start or decelerate and stop at high speed start. Then, it feeds in the opposite direction at FA constant speed until the ORG signal turns OFF. Then, when the LSI finishes counting the specified number of EZ pulses, the motor will stop immediately at constant start or decelerate and stop at high speed start<br>- COUNTER clear timing: When finishing counting the specified number of EZ pulses. |

| Bit | Bit name | Description |
|---|---|---|
| 3 ~ 0 | ORM | 0101: Origin return operation 5<br>- After the ORG signal turns ON, a motor will stop immediately at constant start or decelerate and stop at high speed start. Then, it feeds in the opposite direction at FA constant speed until ORG input turns OFF. Then, when the LSI finishes counting the specified number of EZ pulses, the motor will stop immediately at constant start or decelerate and stop at high start speed.<br>- COUNTER clear timing: When finishing counting the specified number of EZ pulses.<br><br>0110: Origin return operation 6<br>- After the ORG signal turns ON, a motor will stop immediately at constant start or decelerate and stop at high speed start and RENV1.ELM = 1. Then, it feeds in the opposite direction at FA constant speed until ORG input turns OFF and stops immediately.<br>- COUNTER clear timing: When the EL signal is OFF.<br><br>0111: Origin return operation 7<br>- After the ORG input turns ON, a motor will stop immediately at constant start or decelerate and stop at high speed start and RENV1.ELM = 1. Then, it feeds in the opposite direction until the ORG input turns OFF. Then, when the LSI finishes counting the specified number of EZ pulses, the motor will stop immediately.<br>- COUNTER clear timing: When stopped by finishing counting the specified number of EZ pulses.<br><br>1000:Origin return operation 8<br>- After the ORG input turns ON, a motor will stop immediately at constant start or decelerate and stop at high speed start and RENV1.ELM = 1. Then, the motor will start feeding in the opposite direction. 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 motor will stop immediately and decelerates and stops at high start speed.<br>- CONTER clear timing: When finishing counting the specified number of EZ signal.<br><br>1001: Origin return operation 9<br>- After origin return operation 0 has executed, a motor operates until COUNTER2 = 0.<br><br>1010: Origin return operation 10<br>- After origin return operation 3 has executed, a motor operates until COUNTER2 = 0.<br><br>1011: Origin return operation 11<br>- After origin return operation 5 has executed, a motor operates until COUNTER2 = 0.<br><br>1100: Origin return operation 12<br>- After origin return operation 8 has executed, a motor operates until COUNTER2 = 0). |
| 7 ~ 4 | EZD | Specify EZ count value that is used for origin return operations.<br>0000 ~ 1111: 1st ~ 16th count (e.g. 0100b = 5$^{th}$ count) |
| 9, 8 | CI2 | Select the input signal of COUNTER 2.<br>00: EA, EB signals, 01: Command pulses, 10: PA, PB signals,<br>11: Prohibited setting |

| Bit | Bit name | Description |
|---|---|---|
| 12 ~ 10 | CI3 | Select input signal for COUNTER 3.<br>000: Command pulses 001: EA and EB signals<br>010: PA and PB signals 011: 1/4096 division pulse of 40MHz<br>100: Command pulses and EA, EB signals (deviation count)<br>101: PA and PB signals and command pulses (deviation count)<br>110: PA and PB signals and EA and EB signals(deviation count)<br>111: Prohibited setting |
| 13 | BSYC | Set count function of COUNTER 3 only while the LSI is operating (BSY = L level).<br>0: Disabled.<br>1: Enabled. |
| 15, 14 | Not defined | (Always set to 0.) |
| 16 | CU1C | Set COUNTER 1 clear function when CLR signal turns ON.<br>0: Disabled.<br>1: Enabled. |
| 17 | CU2C | Set COUNTER 2 clear function when CLR signal turns ON.<br>0: Disabled.<br>1: Enabled. |
| 18 | CU3C | Set COUNTER 3 clear function when CLR signal turns ON.<br>0: Disabled.<br>1: Enabled. |
| 19 | G03B | Set function to change the timing to check error stop cause (G9103B compatibility).<br>0: Disabled.<br>1: Enabled. Change to the time when writing a start command like G9103B, not at start. |
| 20 | CU1R | Set COUNTER 1 clear function at the origin position when the origin return is complete.<br>0: Disabled.<br>1: Enabled. |
| 21 | CU2R | Set COUNTER 2 clear function at the origin position when the origin return is complete.<br>0: Disabled.<br>1: Enabled. |
| 22 | CU3R | Set COUNTER 3 clear function at the origin position when the origin return is complete.<br>0: Disabled.<br>1: Enabled. |
| 23 | Not defined | (Always set to 0.) |
| 24 | CU1B | Set COUNTER 1 count function while in backlash correction mode.<br>0: Disabled.<br>1: Enabled. |
| 25 | CU2B | Set COUNTER 2 count function while in backlash correction mode.<br>0: Disabled.<br>1: Enabled. |
| 26 | CU3B | Set COUNTER 3 count function while in backlash correction mode.<br>0: Disabled.<br>1: Enabled. |
| 27 | Not defined | (Always set to 0.) |
| 28 | CU1H | Set COUNTER 1 count function.<br>0: Disabled.<br>1: Enabled. |
| 29 | CU2H | Set COUNTER 2 count function.<br>0: Disabled.<br>1: Enabled. |
| 30 | CU3H | Set COUNTER 3 count function.<br>0: Disabled.<br>1: Enabled. |
| 31 | Not defined | (Always set to 0.) |

5.4.4.5 RENV4: Environmental setting 4 register

[WRENV4: 009Fh, RRENV4: 00DFh]

This register is used for Environmental 4 settings. Set up comparators 1 ~ 3.

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| C2RM | C2D | | | C2S | | | C2C | C1RM | C1D | | C1S | | | C1C | |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ISMR | CU3L | CU2L | CU1L | LTOF | LTFD | LTM | | C3D | | C3S | | | | C3C | |

| Bit | Bit name | Description |
|---|---|---|
| 1, 0 | C1C | Select a comparison counter for comparator 1. Note 1<br>00: COUNTER 1 01: COUNTER 2<br>10: COUNTER 3 11: Comparison conditions are not satisfied at any time. |
| 4 ~ 2 | C1S | Select a comparison condition for Comparator 1. Note 2<br>000: Comparison conditions are not satisfied at any time.<br>Select 000b when using ring counter operation (RENV4.C1RM =1).<br>001: RCMP1 data = Comparison counter (regardless of counting direction)<br>010: RCMP1 data = Comparison counter (while counting forward)<br>011: RCMP1 data = Comparison counter (while counting backward)<br>100: RCMP1 data > Comparison counter data<br>101: RCMP1 data < Comparison counter data<br>110: Use as (+) software limit (RCMP1< RCUN1)<br>111: Comparison conditions are not satisfied at any time. |
| 6, 5 | C1D | Select processing to execute when Comparator 1 condition is met.<br>00: None (Event interrupt can occur.)<br>01: Stop immediately.<br>10: Decelerate and stop.<br>11: Execute processing that is the same as writing PRESHF (002B) command. |
| 7 | C1RM | Select count specification of COUNTER 1.<br>0: Count forward and downward counter.<br>1: Ring counter. |
| 9, 8 | C2C | Select a comparison counter for Comparator 2. Note 1.<br>00: COUNTER 1 01: COUNTER 2<br>10: COUNTER 3 11: Comparison conditions are not satisfied at any time. |
| 12 ~ 10 | C2S | Select a comparison condition for Comparator 2. Note 2.<br>000: Comparison conditions are not always satisfied.<br>Select 000b when using ring counter operation (RENV4.C2RM = 1).<br>001: RCMP2 data = Comparison counter (regardless of counting direction)<br>010: RCMP2 data = Comparison counter (while counting forward)<br>011: RCMP2 data = Comparison counter (while counting backward)<br>100: RCMP2 data > Comparison counter data<br>101: RCMP2 data < Comparison counter data<br>110: Use as (−) end software limit (RCMP2>RCUN1).<br>111: Comparison conditions are not satisfied at any time. |
| 14, 13 | C2D | Select processing to execute when the Comparator 2 condition is met.<br>00: None (Event interrupt can occur.)<br>01: Stop immediately.<br>10: Decelerate and stop.<br>11: Execute processing that is the same as writing PRESHF (002B) command. |
| 15 | C2RM | Select count specification of COUNTER 2.<br>0: Count forward and backward counter<br>1: Ring counter |

| Bit | Bit name | Description |
|---|---|---|
| 17, 16 | C3C | Select a comparison counter for Comparator 3. Note 1<br>00: COUNTER 1 01: COUNTER 2<br>10: COUNTER 3<br>11: Comparison conditions are not satisfied at any time or remaining pulse counter. Selected by RENV5.C3C2. |
| 21 ~ 18 | C3S | Select comparison conditions for comparator 3. Note 3<br>0001: RCMP3 data = Comparison counter (regardless of counting direction)<br>0010: RCMP3 data = Comparison counter (while counting forward)<br>0011: RCMP3 data = Comparison counter (while counting backward)<br>0100: RCMP3 data > Comparison counter data<br>0101: RCMP3 data < Comparison counter data<br>0110: Comparison conditions are not satisfied at any time.<br>0111: Prohibited setting<br>1000: Use as an output for IDX (synchronizing) signal (regardless of count direction).<br>1001: Use as an output for IDX (synchronizing) signal (while counting forward).<br>1010: Use as an output for IDX (synchronizing) signal (while counting backward).<br>Others: Comparison conditions are not met at any time. |
| 23, 22 | C3D | Select processing to execute when Comparator 3 condition is met.<br>00: None (Event interrupt can occur.)<br>01: Stop immediately.<br>10: Decelerate and stop.<br>11: Execute processing that is the same as writing PRESHF (002B) command. |
| 25, 24 | LTM | Select latch timing of COUNTER 1 ~ 3<br>00: When LTC signal turns ON.<br>01: When ORG signal turns ON.<br>10: When comparator 2 condition is met.<br>11: When comparator 3 condition is met. |
| 26 | LTFD | Select COUNTER 3 latch data.<br>0: Latch RCUN3 register (COUNTER 3).<br>1: Latch RSPD.AS bit (current speed). |
| 27 | LTOF | Set latch function by LTC signal, or ORG signal ON and when comparative conditions are met.<br>0: Disabled.<br>1: Enabled. |
| 28 | CU1L | Set COUNTER 1 clear function immediately after latching COUNTER 1.<br>0: Disabled.<br>1: Enabled. |
| 29 | CU2L | Set COUNTER 2 clear function immediately after latching COUNTER 2.<br>0: Disabled.<br>1: Enabled. |
| 30 | CU3L | Set COUNTER 3 clear function immediately after latching COUNTER 3.<br>0: Disabled.<br>1: Enabled. |
| 31 | ISMR | Set the function to reset REST and RIST registers automatically after reading.<br>0: Disabled.<br>1: Enabled. |

Note 1: When COUNTER 3 is selected as a comparison counter, the LSI compares counted absolute value and comparator data. (Absolute value range: 0 ~ 32,768)

Note 2: When selecting (+) software limit (RENV4.C1S = 110b) as a comparison condition for comparator 1 and (−) software limit (RENV4.C2S = 110b) as a comparison condition for comparator 2, select COUNTER 1 as the comparison counter.
When software limit is set, a motor will stop immediately other than the case to select deceleration stop when a condition for comparator 1 are met (RENV4.C1D = 10b) or to select deceleration stop when a condition for comparator 2 is met (RENV4.C2D = 10b). A motor will stop immediately at constant start or decelerate and stop at high speed start when a condition for comparator 1 is met (RENV4.C2D = 10b) and in the case to select deceleration stop when condition for comparator 2 is met (REN4.C2D = 10b).

At the moment that overflow or underflow of comparison counter occur, comparison status during count up or countdown reverses. Please note that if an upper limit value or lower limit value of comparison counter is set as comparison data.

Note 3: When IDX (synchronizing) signal output is selected as comparator 3 (RENV4.C3S = 1000b, 1001b, 1010b), select counter 3 as a comparison counter. Counter 3 is a dedicated counter for IDX (synchronizing) signal output. Set a positive number as a comparator value.

5.4.4.6 RENV5: Environment setting 5 register

[WRENV5: 00A0h, RRENV5: 00E0h]

This register is used for the Environmental 5 settings. It is mainly 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 | C3S2 | ADJ | BR | | | | | | | | | | | |

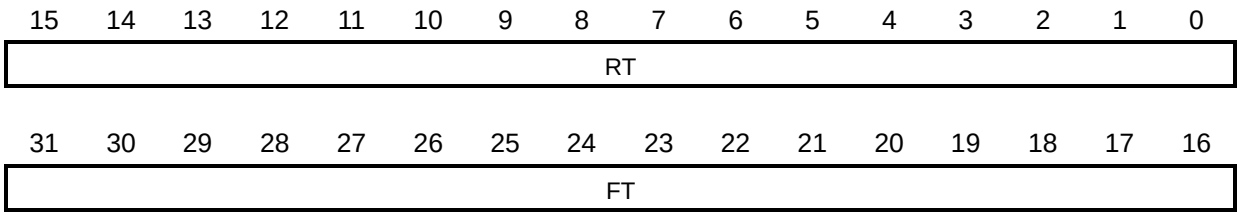
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PMG | | | | | PD | | | | | | | | | | |

| Bit | Bit name | Description |
|---|---|---|
| 11~ 0 | BR | Enter a backlash correction amount. (Setting range: 0 ~ 4095) |
| 12 | ADJ | Set backlash correction function.<br>0: Disabled.<br>1: Enabled. |
| 13 | C3C2 | When RENV4.C3C = 11b, remaining pulse counter can be selected as a comparison counter of Comparator 3.<br>0: Comparison conditions are not met at any time.<br>1: Remaining pulse counter. RPLS or RCIC register will be selected automatically. |
| 14 | Not defined | (Always set to 0.) |
| 15 | PSTP | Set the function to output remaining pulsar signal input.<br>0: Disabled.<br>1: Enabled.<br>*Disabled in interpolation operation mode. |
| 26 ~ 16 | PD | Specify division ratio for pulses on PA and PB signals. The number of pulses is divided using the set value/2048. When 0 is entered, the frequency divider will be OFF. (= 2048/2048)<br>[Setting range: 0 ~ 2,047] |
| 31 ~ 27 | PMG | Specify magnification for pulses on PA and PB signals. The number of pulses is multiplied by the set value + 1.<br>[Setting range: 0 ~ 31] |

5.4.4.7 RENV6: Environmental setting 6 register

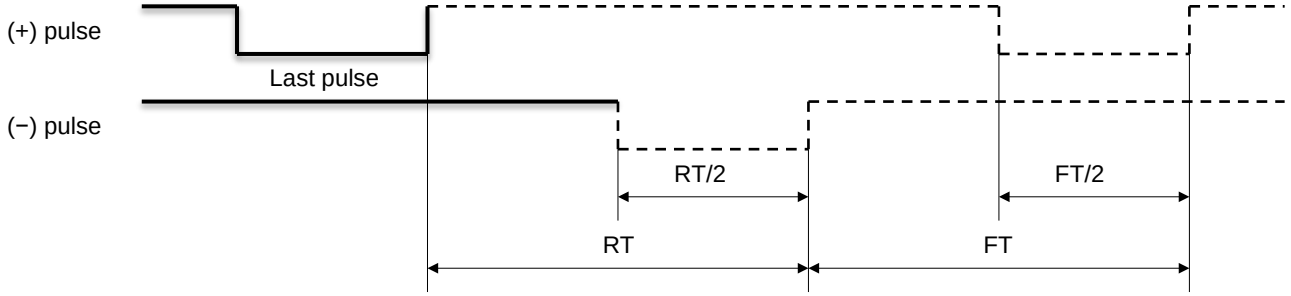
[WRENV6: 00A1h, RRENV6: 00E1h]

This register is used for the Environmental 6 settings. It is mainly 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.




| Bit | Bit name | Description |
|---|---|---|
| 15 ~ 0 | RT | Enter reverse rotation time (RT) shown in the figure below.<br>Setting unit is 1.6 µs. Setting time range is 0 ~ approximately 0.1 s.<br>[Setting range: 0 ~ 65,535] |
| 31 ~ 16 | FT | Enter the normal rotation time (FT) shown in the figure below.<br>Setting unit is 1.6 µs. Setting time range is 0 ~ approximately 0.1 s.<br>[Setting range: 0 ~ 65,535] |

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




Setting time [RT, FT] = (a setting value) x 1.6 (µs)

5.4.4.8 RCUN1: COUNTER 1 (28 bits)

[WRCUN1: 00A3h, RRCUN1: 00E3h]

This is a 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 range: −134,217,728 ~ +134,217,727.

5.4.4.9 RCUN2: COUNTER 2 (28 bits)

[WRCUN2: 00A4h, RRCUN2: 00E4h]

This is a 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 and pulse generator signals.
Setting range: −134,217,728 ~ +134,217,727.

5.4.4.10 RCUN3: COUNTER 3 (16 bits)

[WRCUN3: 00A5h, RRCUN3: 00E5h]

This is a deviation and 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 pulse generator signals, and between encoder signals and pulse generator signals. Counter 3 will not count values exceeding the setting and it shows the maximum or minimum value.
Setting range: −32,768 ~ +32,767.

5.4.4.11 RCMP1: Comparison data for Comparator 1 (28 bit)

[WRCMP1: 00A7h, RRCMP1: 00E7h]

Specify 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 ~ +134,217,727.

5.4.4.12 RCMP2: Comparison data for Comparator 2 (28 bit)

[WRCMP2: 00A8h, RRCMP2: 00E8h]

Specify 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 ~ +134,217,727.

5.4.4.13 PRCP3 (RCMP3): Comparison data for Comparator 3 (28 bit)

[WPRCP3: 00BBh, RPRCP3: 00CBh]
[WRCMP3: 00A9h, RRCMP3: 00E9h]

Specify 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 ~ +134,217,727.

5.4.4.14 RIRQ: Event interrupt cause setting register

[WRIRQ: 00ACh, RRIRQ: 00ECh]

This register is to set event interrupt causes.
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 a motor stops normally. |
| 1 | IRUS | When a motor starts acceleration. |
| 2 | IRUE | When a motor finishes acceleration. |
| 3 | IRDS | When a motor starts deceleration. |
| 4 | IRDE | When a motor finishes deceleration. |
| 5 | IRC1 | When Comparator 1 condition is met. |
| 6 | IRC2 | When Comparator 2 condition is met. |
| 7 | IRC3 | When Comparator 3 condition is met. |
| 8 | IRCL | When the count value is cleared when a CLR signal turning ON. |
| 9 | IRLT | When the count value is latched when an LTC signal turning ON. |
| 10 | IROL | When the count value is latched when an ORG signal turning ON. |
| 11 | IRSD | When SD signal turns ON. |
| 12 | IRSA | When STA signal turns ON. |
| 13 | IRNA | When the motor starts by the start command (2x01h) of broadcast communication. |
| 14 | IRNP | When the motor stops by the stop command (2x02h) of broadcast communication. |
| 15 | IRNM | When writing into the pre-register for operation is ready (Change to MSTS.SPRF = 0.) |
| 16 | IRBE | When completing the current operation while the pre-register is not fixed. (Change to RSTS.PFM = 00b.) |
| 31 ~ 17 | Not defined | (Always set to 0.) |

5.4.4.15 RLTC1: Latch data for COUNTER 1 (28 bits)

[RRLTC1: 00EDh]

Latch data for COUNTER 1. (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 by LTC, an ORG signal turning ON, or LTCH (0029h) command. Data range: −134,217,728 ~ +134,217,727.

5.4.4.16 RLTC2: Latch data for COUNTER 2 (28 bits)

[RRLTC2: 00EEh]

Latch data for COUNTER 2. (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 2 are copied by LTC, ORG signal turning ON or an LTCH (0029h) command. Data range: −134,217,728 to +134,217,727.

5.4.4.17 RLTC3: Latch data for COUNTER 3 (16 bits)

[RRLTC3: 00EFh]

Latch data for COUNTER 3 or RSPD.AS (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 RSPD.AS (current speed) are copied by the LTC, an ORG signal turning ON, or an LTCH (0029h) command.

| RENV4.LTFD | 0 | 1 |
|---|---|---|
| Latched data | COUNTER 3 | RSPD.AS (0 while stopping) |
| Data range | −32,768 ~ +32,767 | 0 = 100,000 |
| “%” bit | Same as bit 15 (Code extension) | Latched data (~ bit 16) |
| “$” bit | Same as bit 15 (Code extension) | 0 |

5.4.4.18 RSTS: Extension status

[RRSTS: 00F1h]

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 | CND | | | |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PFM | | PFC | | SPH4 | SPH3 | SPH2 | SPH1 | SPLS | SCP3 | SCP2 | SCP1 | SINP | SLTC | SCLR | SEZ |

| Bit | Bit name | Description |
|---|---|---|
| 3 ~ 0 | CND | These bits check operation status.<br>0000: Stopping<br>0001: Waiting for STA signal input<br>0010: Waiting for completion of ERC timer<br>0011: Waiting for a completion of direction change timer<br>0100: Correcting backlash<br>0101: Waiting for PA and PB signals input<br>0110: Feeding at FA constant speed<br>0111: Feeding at FL constant speed<br>1000: Accelerating<br>1001: Feeding at FH constant speed.<br>1010: Decelerating<br>1011: Waiting for INP or during vibration restriction<br>1100: Not defined<br>1101: Not defined<br>1110: Not defined<br>1111: Others (controlling start or stop)<br>*When FH correction function is enabled, operation speed may not become be FH speed even in FH constant speed feeding (RSTS.CND = 1001).See “7.3 FH correction”. |
| 4 | SDIR | Select operation direction<br>0: (+) direction.<br>1: (−) direction. |
| 5 | SALM | 1: ALM signal is ON. |
| 6 | SPEL | 1: +EL signal is ON. |
| 7 | SMEL | 1: −EL signal is ON. |
| 8 | SORG | 1: ORG signal is ON. |
| 9 | SSD | 1: SD signal is ON. (SD latch status.) |
| 10 | SDIN | 1: SD signal is ON. (SD terminal input status.) |
| 11 | SSTA | 1: STA signal is ON. |
| 12 | SSTP | 1: STP signal is ON. |
| 13 | SEMG | 1: EMG signal is ON. |
| 14 | SPCS | 1: PCS signal is ON. |
| 15 | SERC | 1: ERC signal is ON. |
| 16 | SEZ | 1: EZ signal is ON. |
| 17 | SCLR | 1: CLR signal is ON. |
| 18 | SLTC | 1: LTC signal is ON. |
| 19 | SINP | 1: INP signal is ON. |
| 20 | SCP1 | 1: When a CMP1 comparison condition is met. |
| 21 | SCP2 | 1: When a CMP2 comparison condition is met. |
| 22 | SCP3 | 1: When a CMP3 comparison condition is met. |
| 23 | SPLS | 1: When pulse output is ON. |
| 24 | SPH1 | 1: When PH1 signal is H level. |
| 25 | SPH2 | 1: When PH2 signal is H level. |
| 26 | SPH3 | 1: When PH3 signal is H level. |
| 27 | SPH4 | 1: When PH4 signal is H level. |

| Bit | Bit name | Description |
|---|---|---|
| 29, 28 | PFC | Monitor the usage of pre-register for comparator 3.<br>00: Both register and pre-register are unfixed.<br>01: Register is fixed and pre-register is unfixed.<br>1X: Both register and pre-register are fixed. (New writing is impossible.) |
| 31, 30 | PFM | Monitor the usage of pre-register for operation.<br>00: Both register and pre-register are unfixed.<br>01: Register is fixed and pre-register is unfixed.<br>1X: Both register and pre-register are fixed. (New writing is impossible.) |

5.4.4.19 REST: Error interrupt status

[WREST: 00ADh, RREST: 00F2h]

Reading this register, you can check an error interrupt cause
The corresponding bit will be "1" when that an error interrupt has occurs.

This register is reset by the following processing.

1. When RENV4.ISMR = 0 (default status).
   All bits are reset automatically by reading this register. Writing data that only bits to be reset are 1 resets this register.
2. When RENV4.ISMR = 1.
   This register is reset by writing data that only bits to be reset are 1. That is, all are 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 after automatic reset and send the data. Therefore, we recommend the setting RENV4.ISMR = 1 that does not reset automatically by reading.

| 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 | ENST | ESWM | ESPM | ECKM | EFAJ | ESAO | ESIP |

| Bit | Bit name | Description |
|---|---|---|
| 0 | ESC1 | 1: Stopped when a Comparator 1 condition is met. (+SL) |
| 1 | ESC2 | 1: Stopped when a Comparator 2 condition is met. (−SL) |
| 2 | ESC3 | 1: Stopped when a Comparator 3 condition is met. |
| 3 | ESPL | 1: Stopped by +EL signal turning ON. |
| 4 | ESML | 1: Stopped by −EL signal turning ON. |
| 5 | ESAL | 1: Stopped by ALM signal turning ON. |
| 6 | ESSP | 1: Stopped by STP signal turning ON. |
| 7 | ESEM | 1: Stopped by EMG signal turning ON. |
| 8 | ESSD | 1: Decelerated and stopped by SD signal turning ON. |
| 9 | ESPO | 1: An overflow occurred in PA and PB signal input buffer counter. |
| 10 | ESNT | 1: Stopped by watchdog timer with communication error. (Communication line disconnection, etc.) |
| 11 | Not defined | (Always set to 0.) |
| 12 | ESOR | 1: When writing position override command (WRMVOR) while a motor is stopping. |
| 13 | ESEE | 1: When EA and EB signals changed simultaneously. (The motor does not stop.) |
| 14 | ESPE | 1: When PA and PB signals changed simultaneously. (The motor does not stop.) |
| 15 | ESDT | 1: Simultaneously stopped by data error for interpolation operation (Note.1) |
| 16 | ESIP | 1: Simultaneously stopped by abnormal stop of other axes. |
| 17 | ESAO | 1: Stopped by exceeding the circular interpolation range (28 bits) |
| 18 | EFAJ | 1: When synchronizing status of clock for motor control is lost. (A motor does not stop.) |
| 19 | ECKM | 1: Stopped when communication error had occurred the specified times in receiving communication for synthesizing clock continuously. |
| 20 | ESPM | 1: Stopped when communication error had occurred the specified times in receiving communication for simultaneous stop continuously. |
| 21 | ESWM | 1: Stopped when communication error had occurred the specified times in receiving communication for senor signal substitute input |
| 22 | ENST | 1: Writing start command for next operation (NSTAFL, NSTAFH, NSTAD, NSTAUD) while a motor is stopping |
| 31 ~ 23 | Not defined | (Always set to 0.) |

Note 1: An error interrupt occurs when a start command is written with the following data setting.
1. In linear interpolation: (RMV = 0) and (RMVY = 0)
2. In circular interpolation: ((RIP = 0) and (RIPY = 0)) or ((RMV = RIP) and (RMVY = RIPY))

5.4.4.20 RIST: Even interrupt status

[WRIST: 00AEh, RRIST: 00F3h]

Reading this register, you can check an even interrupt cause.
The corresponding bit will be "1" when that an event interrupt has occurs.
You can select event interrupt causes by RIRQ register.

This register is reset by the following process.

1. When RENV4.ISMR=0 (default status).
   All bits are reset automatically by reading this register. Writing data that only bits to be reset are 1 resets this register.
2. When RENV4.ISMR=1.
   This register is reset by writing data that only bits to be reset are 1. That is, all are 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 after automatic reset and send the data. Therefore, we recommend the setting RENV4.ISMR=1 that does not reset automatically by reading.

| 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 | 1: Stopped normally. |
| 1 | ISUS | 1: Starting acceleration. |
| 2 | ISUE | 1: Ending acceleration. |
| 3 | ISDS | 1: Starting deceleration. |
| 4 | ISDE | 1: Ending deceleration. |
| 5 | ISC1 | 1: When a comparator 1 condition is met. |
| 6 | ISC2 | 1: When a comparator 2 condition is met. |
| 7 | ISC3 | 1: When a comparator 3 condition is met. |
| 8 | ISCL | 1: When CLR signal turning ON. |
| 9 | ISLT | 1: When LTC signal turning ON. |
| 10 | ISOL | 1: When ORG signal turning ON. |
| 11 | ISSD | 1: When SD signal turns ON. |
| 12 | ISSA | 1: When STA signal turns ON. |
| 13 | ISNA | 1: Started by a start command of broadcast communication (2x01h). |
| 14 | ISNP | 1: Stopped by a stop command of broadcast communication (2x02h). |
| 15 | ISNM | 1: When the pre-register for the next operation is ready for writing. (Change to MSTS.SPRF = 0). |
| 16 | ISBE | 1: When completing the current operation while the pre-register is not fixed. (Change to RSTS.PFM = 00b). |
| 31 ~ 17 | Not defined | (Always set to 0.) |

5.4.4.21 RPLS: Positioning counter (28 bits)

[RRPLS: 00F4h]

This register is used to check a value of a positioning counter. (Read only)
This is used as a remaining pulse counter of positioning operation mode.

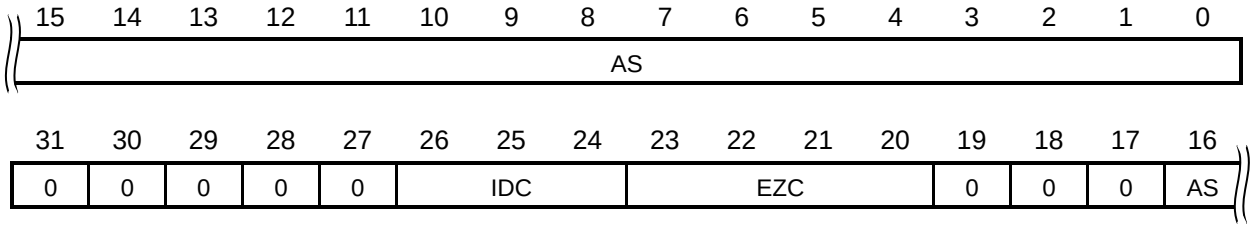
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | |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

Positioning counter sets feed amount calculated from RMV register value in positioning operation mode.
Positioning counter count down to 0 every pulse output. Remaining pulse counter's specification in positioning operation mode is the same as one of positioning counter.

5.4.4.22 RSPD: Current speed monitor

[RRSPD: 00F5h]

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

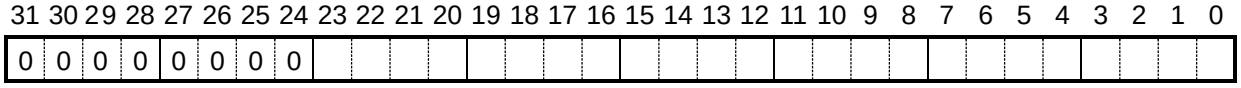



| Bit | Bit name | Description |
|---|---|---|
| 16 ~ 0 | AS | The current speed can be checked by a step value (same units as for RFL register and RFH register). When a motor is stopped, the value is 0. |
| 19 ~ 17 | Not defined | (Always set to 0.) |
| 23 ~ 20 | EZC | The value of EZ signal input counter that is used for an origin return can be checked. For details, see “6.6 EZ count operation mode”. |
| 26 ~ 24 | IDC | An idling count value can be checked. |
| 31 ~ 27 | Not defined | (Always set to 0.) |

5.4.4.23 RSDC: Ramping-down point auto calculated value register (24 bit)

[RRSDC: 00F6h]

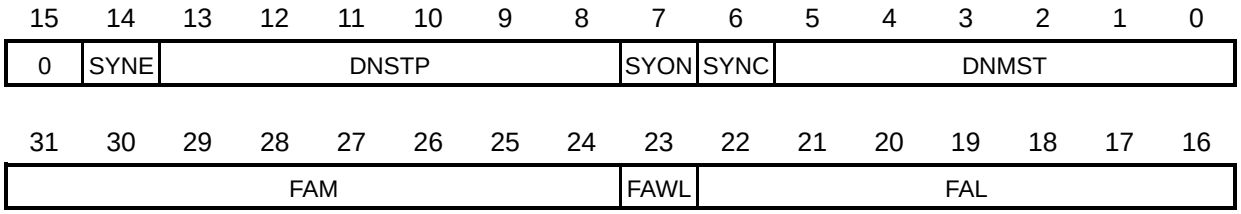
This register is used to check ramping-down point value that is calculated automatically, for the positioning operation. (Read only.)




5.4.4.24 RSYN: Synchronizing control register

[WRSYN: 008Fh, RRSYN: 00FFh]

This register is used to set synchronizing information among the multiple G9103Cs.
For details, see “8.17 Synchronizing function with other axes”.




| Bit | Bit name | Description |
|---|---|---|
| 5 ~ 0 | DNMST | Set device number of the clock master LSI for synchronizing clock for motor control. |
| 6 | SYNC | Set clock synchronizing function for motor control.<br>0: Disabled.<br>1: Enabled. (Bit definition of main status changes.) |
| 7 | SYON | Check clock synchronizing status for motor control.<br>0: Not synchronized.<br>1: Synchronized.<br>This bit is read only and ignored when written.<br>When synchronizing control in interpolation operation, etc. is used, check RSYN.SYON = 1. |
| 13 ~ 8 | DNSTP | Set a device number of monitor target LSI for synchronizing function of simultaneous stop. |
| 14 | SYNE | Set a synchronizing function of simultaneous stop.<br>0: Disabled.<br>1: Enabled (Bit definition of main status changes.) |

| Bit | Bit name | Description |
|---|---|---|
| 15 | not defined | (Always set to 0.) |
| 22 ~ 16 | FAL | Set a limit o f frequency correction. When correction frequency is over the limit, this condition is regarded as a synchronizing error.<br>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 synchronizing error.<br>0: Disabled.<br>1: Enabled |
| 31 ~ 24 | FAM | Monitor the frequency correction amount for clock synchronization. This bit is read only and ignored when written. For details, see “8.17.1 Clock Synchronizing function of clock for motor control.” |

5.4.4.25 RSYN 2: Synchronizing control register

[WRSYN2:00ABh, RRSYN2:00EBh]

This register is used to set the synthesizing information among the multiple G9103Cs.
For details, see “8.17 Synchronizing function with other axes”.

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SPME | SAME | ALME | EMME | ORME | SDME | MEME | PEME | PNSWM | | DNSWM | | | | | |

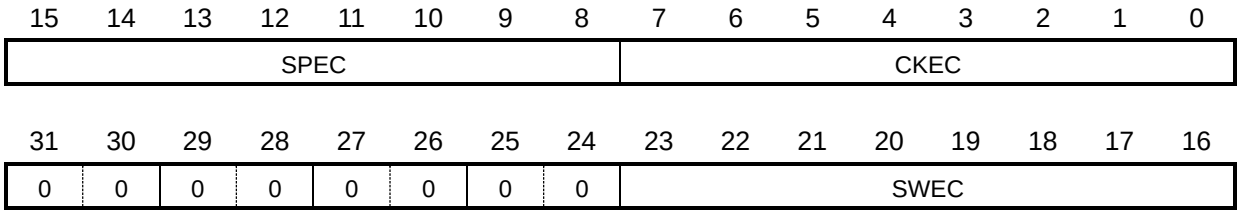
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | SWMEV | | | | SPMEV | | | | CKMEV | | | |

| Bit | Bit name | Description |
|---|---|---|
| 5 ~ 0 | DNSWM | Specify device number of I/O device that substitutes for sensor signal (STP, STA, ALM, EMG, ORG, SD, −EL, +EL) |
| 7, 6 | PNSWM | Specify a port number of I/O device that substitutes for a sensor signal. |
| 8 | PEME | Select input of +EL signal.<br>0: Use +EL terminal.<br>1: Use bit 0 input of I/O device port. |
| 9 | MEME | Select input of –EL signal.<br>0: Use −EL terminal.<br>1: Use bit 1 input of I/O device port. |
| 10 | SDME | Select input of SD signal.<br>0: Use SD terminal.<br>1: Use bit 2 input of I/O device port. |
| 11 | ORME | Select input of ORG signal.<br>0: Use ORG terminal.<br>1: Use bit 3 input of I/O device port. |
| 12 | EMME | Select input of EMG signal.<br>0: Use EMG terminal.<br>1: Use bit 4 input of I/O device port. |
| 13 | ALME | Select input of ALM signal.<br>0: Use ALM terminal.<br>1: Use bit 5 input of I/O device port. |
| 14 | SAME | Select input of STA signal.<br>0: Use STA terminal.<br>1: Use bit 6 input of I/O device port. |
| 15 | SPME | Select input of STP signal.<br>0: Use STP terminal.<br>1: Use bit 7 input of I/O device port. |
| 19 ~ 16 | CKMEV | Specify the maximum number of continuous error occurrence for monitoring clock synchronizing communication.<br>If monitoring errors occur continuously specified times, a motor stops as error stop. When 0 is selected as a specified value, error stop does not occur. |
| 23 ~ 20 | SPMEV | Specify the maximum number of continuous error occurrence for monitoring simultaneous stop communication.<br>If monitoring errors occur continuously specified times, motor stops as error stop. When 0 is selected as a specified value, error stop does not occur. |
| 27 ~ 24 | SWMEV | Specify the maximum number of continuous error occurrence for monitoring sensor substitute communication.<br>If monitoring errors occur continuously specified times, motor stops as error stop. When 0 is selected as a specified value, error stop does not occur. |
| 31 ~ 28 | not defined | Always set to 0. |

5.4.4.26 RMEC: communication monitoring error counter

[RRMEC:00CFh]

This register is used to read an error count to monitor errors in communication for clock synchronization, simultaneous stop and sensor signal substitute input.
For details, see “8.17 Synchronizing function with other axes”.

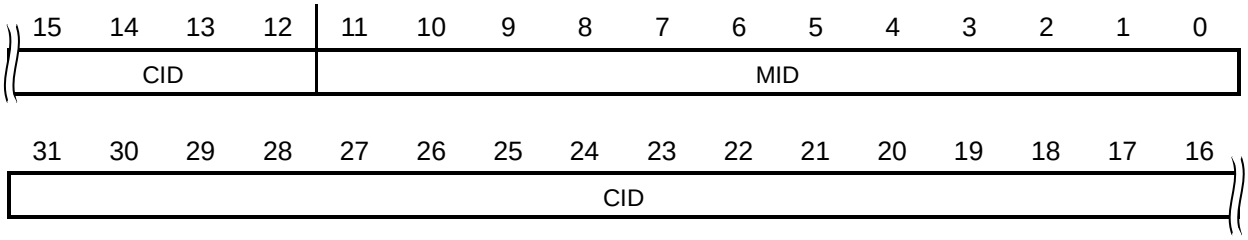



| Bit | Bit name | Description |
|---|---|---|
| 7 ~ 0 | CKEC | The number of error occurrence for monitoring communication for clock synchronization. (0 ~ 255)<br>This value is cleared to 0 by default and after CKMECR (0030h) command is issued. Count value stops at 255. |
| 15 ~ 8 | SPEC | The number of error occurrence for monitoring communication for simultaneous stop. (0 ~ 255)<br>This value is cleared to 0 by default and after SPMECR (0031h) command is issued. Count value stops at 255. |
| 23 ~ 16 | SWEC | The number of error occurrence for monitoring communication for sensor substitute input. (0 ~ 255)<br>This value is cleared to 0 by default and after SWMECR (0032h) command is issued. Count value stops at 255. |
| 31 ~ 24 | not defined | Always set to 0. |

5.4.4.27 RGNO: General-purpose register (32 bits)

[WRGN0:0087h, RRGN0:00F7h]

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 controlled by NPM and model ID is controlled by customers.
For details, see “8.18 Unit ID control function”.

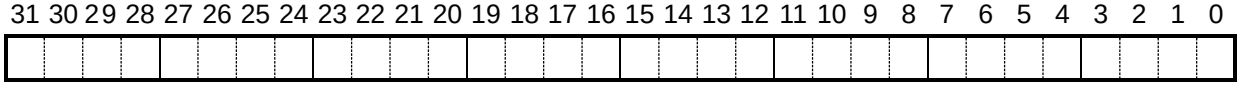



| Bit | Bit name | Description |
|---|---|---|
| 11 ~ 0 | MID | This is used for model ID. |
| 31 ~ 12 | CID | This is used for company ID. |

5.4.4.28 RGN1 ~ 3: General-purpose registers 1 ~ 3 (each 32 bits)

[WRGN1:0088h, RRGN1:00F8h]
[WRGN2:0089h, RRGN2:00F9h]
[WRGN3:008Ah, RRGN3:00FAh]

This register is a general-purpose register and does not affect motor control.
This register can be used to transfer parameters to a local CPU when G9103C and CPU are connected by serial bus. For details, see “8.19 CPU connection function”.



## *6. Operation Mode*

Specify basic operation mode using operation mode register (RMD.MOD).

### *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.

| RMD.MOD | Operation mode | Direction of movement |
|---|---|---|
| 00h | Continuous operation by a command | (+) direction |
| 08h | Continuous operation by a command | (−) direction |

A motor stops by EL signal corresponding to the direction of operation turning ON.
When operation direction is positive, +EL is enabled.
When operation direction is negative, −EL is enabled.
In order to start operation in the reverse direction after stopping the motion by +EL or –EL signal turning ON, a new start command must be written.

### *6.2 Positioning operation mode*

The following seven operation types are available for positioning operations.

| RMD.MOD | Operation mode | Direction of movement |
|---|---|---|
| 41h | Positioning operation (specify the incremental position) | (+) direction when RMV > 0<br>(−) direction when RMV < 0 |
| 42h | Positioning operation (specify the absolute position in COUNTER 1) | (+) direction when RMV > RCUN1<br>(−) direction when RMV < RCUN1 |
| 43h | Positioning operation (specify the absolute position in COUNTER 2) | (+) direction when RMV > RCUN2<br>(−) direction when RMV < RCUN2 |
| 44h | Return operation to command position 0 (COUNTER1) | (+) direction when RCUN1 < 0<br>(−) direction when RCUN1 > 0 |
| 45h | Return operation to machine position 0 (COUNTER 2) | (+) direction when RCUN2 < 0<br>(−) direction when RCUN2 > 0 |
| 46h | One pulse operation | (+) direction |
| 4Eh | One pulse operation | (−) direction |
| 47h | Timer operation | |

#### 6.2.1 Positioning operation (specify incremental position)

This is a positioning mode to feed a value specified in RMV register.
The feed direction is automatically specified by a sign set in RMV register.

RMD.MOD: 41h positioning operation (specify incremental position)

At the start, the RMV register setting is loaded into RPLS register.
RPLS register counts down each pulse output and stops when the counter reaches 0.
When you set a RMV register value to 0 and start positioning operation, the LSI will stop immediately without outputting pulses.

Target position can be overridden by writing a new target position into RMV register while motor is running.

#### 6.2.2 Positioning operation (specify an absolute position in COUNTER 1)

This mode is to feed only the difference between a RMV register value and a COUNTER 1 register value.
The feed direction is automatically specified by the magnitude relation between a RMV register value and a RCUN1 register value.

RMD.MOD: 42h positioning operation (specify an incremental position in COUNTER 1)

At the start, an absolute value of the difference between a RMV setting and a value stored in RCUN1 is loaded into RPLS register
RPLS register counts down each pulse output and stops when the counter reaches 0.

When you set a RMV register value to 0 and start a positioning operation, the LSI will stop immediately without outputting pulses.

Target position can be overridden by writing a new target position into RMV register while motor is running. Because RCUN1 register value is saved at the start, target position cannot be override by writing a value to RCUN1 register.

### 6.2.3 Positioning operation (specify absolute position in COUNTER 2)

This mode is to feed only the difference between a RMV register value and a COUNTER 1 register value. The feed direction is automatically specified by the magnitude relation between a RMV register value and a RCUN2 register value.

RMD.MOD: 43h positioning operation (specify incremental position in COUNTER 1)

At the start, the absolute value of the difference between a RMV setting and a value stored in RCUN2 register is loaded into RPLS register
RPLS register counts down each pulse output and stops when the counter reaches 0.
When you set RMV register value to 0 and start a positioning operation, the LSI will stop immediately without outputting pulses.

Target position can be overridden by writing a new target position into RMV register while motor is running. Because RCUN2 register value is saved at the start, target position cannot be override by writing a value to RCUN2 register.

This is not a feedback control. Therefore, a RCUM2 register value may not be the same as a RMV register value at operation complete in the case that an encoder is selected as RCUN 2 register input.

### 6.2.4 Command position 0 return operation

This mode continues operation until a RCUN1 register value becomes zero.
The feed direction is automatically specified by a sign of RCUN1 register value at the start.

RMD.MOD: 44h command position 0 return operation

This operation is the same as when positioning (specify the absolute position in COUNTER 1) by entering zero in RMV register.

### 6.2.5 Mechanical position 0 return operation

This mode is used to continue operations until a RCUN2 register value becomes zero.
The feed direction is automatically specified by a sign of RCUN2 register value at the start.

RMD.MOD: 45h command position 0 return operation

This operation is the same as when positioning (specify the absolute position in COUNTER 2) by entering zero in RMV register.

### 6.2.6 One pulse operation

This mode outputs a single pulse.

RMD.MOD: 46h (+) direction one pulse operation
4Eh (−) direction one pulse operation

This operation is identical to a positioning operation (incremental target positioning) that writes a "1" (or "−1") to RMV register.

### 6.2.7 Timer operation

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.

RMD.MOD: 47h (+) timer operation

The internal operation time using a constant speed start command will be a product of an output frequency of command pulses and RMV register setting.
(Ex.: When the frequency is 1000 pps and RMV register is set to 120 pulses, the internal operation time will be 120 ms.)

Set a positive number (1 ~ 134,217,727) into RMV register.

+EL signal, −EL signal, SD signal, and software limits are ignored. These are always treated as OFF.
ALM signal, STP signal, and EMG input signals are enabled.

Backlash correction, vibration restriction function, and change direction timer function stop.
Count of COUNTER 1 also stops.
Regardless of RMD.MINP setting, an operation complete delay controlled by INP signal will not occur.

In order to eliminate errors in the internal operation time, set “When output pulse cycle is complete” (RMD.METM = 0) as the operation. Even if “When output pulse cycle is complete” is selected, an error $T_{err} = 750 - \left(\frac{5,000,000}{RFH}\right)$ ns occurs. If STAFL (0050h) command is used, replace the above a RFH register value with a RFL register value.

## *6.3 Pulse generator (PA, PB) input mode*

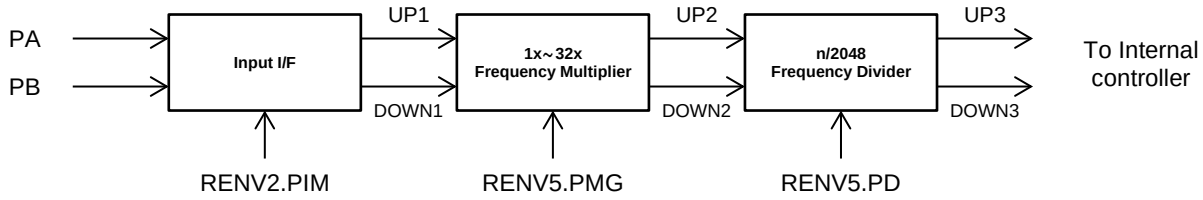
This mode is used to allow operations synchronized with a pulse generator signal input.
Input of pulse generation signals can be enabled (RENV2.POFF=0) and applying a filter can be also enabled (RENV2.PINF=0).
After writing a start command, the LSI will output pulses from OUT terminal or DIR terminal according to output pulse mode when pulse generator signal is input to PA and PB terminals.
Use STAFL (0050h) command or STAFH (0051h) command as a start command.

Four I/Fs are available to input pulse generator signals from PA and PB terminals.

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

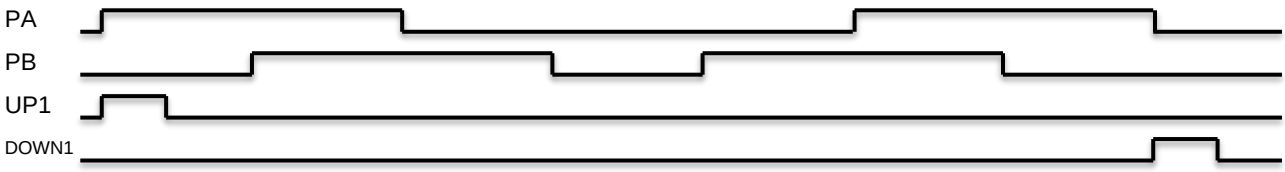
Note: Backlash correction function is available with pulse generator input mode. However, reversing pulse generator input while the backlash correction is unavailable.

Besides the above 1x to 4x input I/F, the G9103C has a multiplication circuit of 1x ~ 32x frequency multiplier and n/2048 frequency divider. For setting the multiplication from 1x ~ 32x, specify RENV5.PMG bit and for setting the divider of n/2048, specify RENV5.PD bit. “n” is “1 ~ 2048.

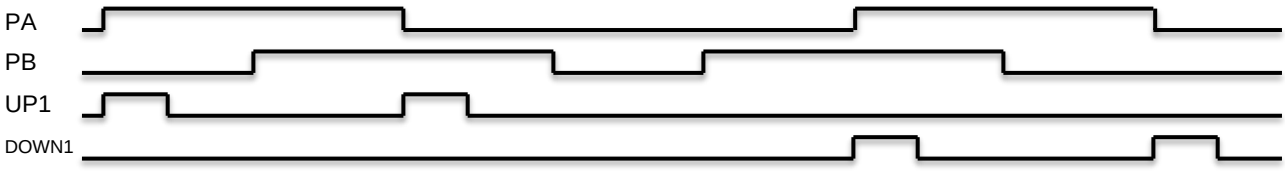



UP1 and DOWN1 signals will be as follows by setting RENV2.PIM bit.

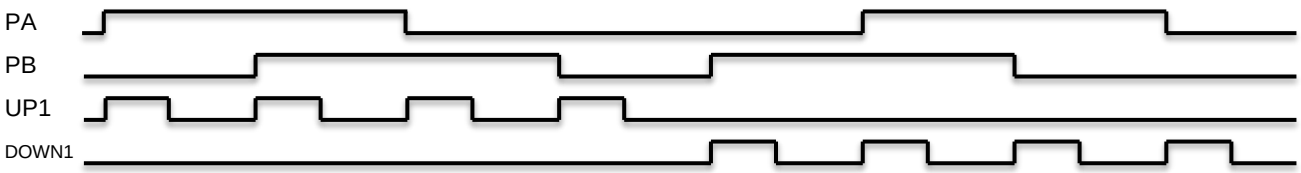
1) When inputting 1x input of 90 degree phase difference signals (RENV2.PIM = 00b)



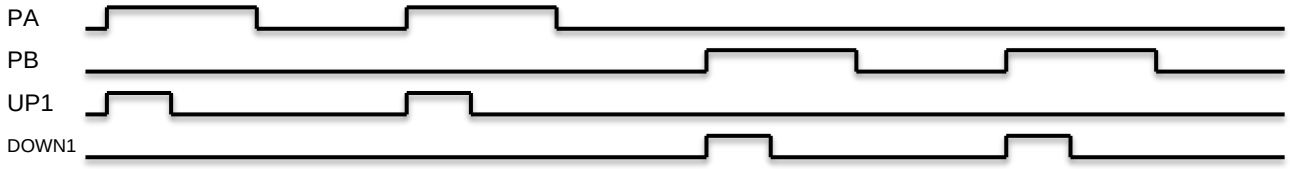

2) When inputting 2x input of 90 degree phase difference signals (RENV2.PIM = 01b)



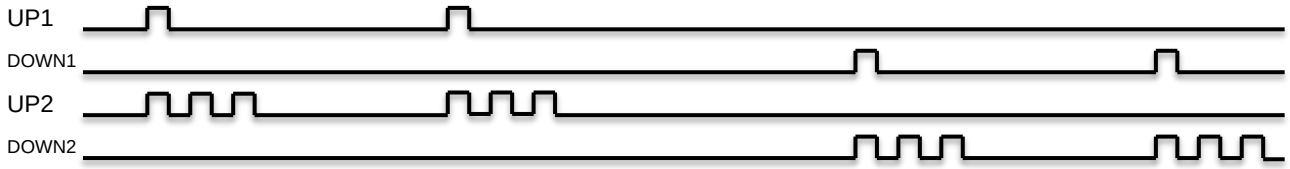

3) When inputting 4x input of 90 degree phase difference signals (RENV2.PIM = 10b)



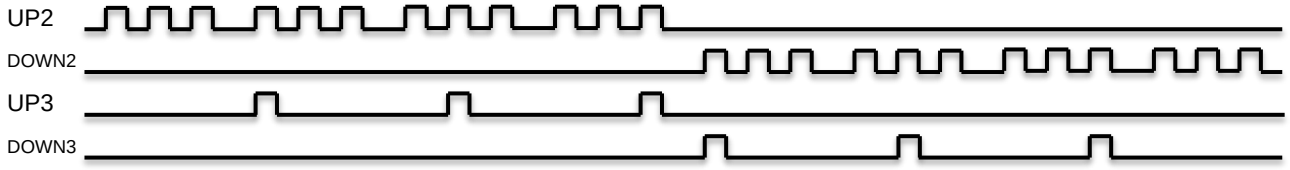
4) When inputting Two-pulse input (RENV2.PIM=11b).




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




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




Input of pulse generator signal causes output internal pulses at FL speed or FH speed partially discrete. Therefore, there may be a difference in the timing between pulse generator input and output pulses, up to an internal pulse frequency period.

The maximum input frequency of pulse generator signals (FP) is restricted by FL speed when STAFL (0050H) command is used and by the FH speed when STAFH (0051H) command is used.
The LSI generates an interrupt as errors when both PA and PB signals inputs change simultaneously, or when the input frequency exceeded FP and the input/output buffer counter (16-bit) overflows. Error interrupt cause can be monitored by REST register.

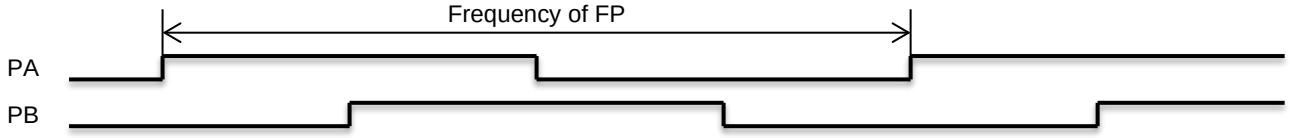
1. In the case of RENV5.PD ≠0
   FP < (FL speed or FH speed) / PIM / (RENV5.PMG+1) / (RENV5.PD / 2048)
2. In the case of RENV5.PD = 0
   FP < (FL speed or FH speed) / PIM / (RENV5.PMG+1)

“PIM” in the above FP calculation formula is a setting value in RENV2.PIM bit and as follows.

| Pulse generator signal input I/F | RENV2.PIM | PIM |
|---|---|---|
| 90 degree phase deference 1x | 00 | 1 |
| 90 degree phase deference 2x | 01 | 2 |
| 90 degree phase deference 4x | 10 | 4 |
| Two pulse signal | 11 | 1 |

<Examples of the relationship between FH or FL speed and the maximum pulsar input frequency FP>

| RENV2.PIM setting | PMG setting value | PD setting value | Usable range |
|---|---|---|---|
| 90 degree phase difference 1x | 0 (1x) | 0 | FP < FH (FL) / 1 / 1 |
| | 0 (1x) | 1024 | FP < FH (FL) / 1 / 1 x 2 |
| | 2 (3x) | 0 | FP < FH (FL) / 1 / 3 |
| 90 degree phase difference 2x | 0 (1x) | 0 | FP < FH (FL) / 2 / 1 |
| | 0 (1x) | 1024 | FP < FH (FL) / 2 / 1 x 2 |
| | 2 (3x) | 0 | FP < FH (FL) / 2 / 3 |
| 90 degree phase difference 4x | 0 (1x) | 0 | FP < FH (FL) / 4 / 1 x 2 |
| | 0 (1x) | 1024 | FP < FH (FL) / 2 |
| | 2 (3x) | 0 | FP < FH (FL) / 4 / 3 |
| Two-pulse input | 0 (1x) | 0 | FP < FH (FL) / 1 / 1 |
| | 0 (1x) | 1024 | FP < FH (FL) / 1 / 1 x 2 |
| | 2 (3x) | 0 | FP < FH (FL) / 1 / 3 |




Note: When pulse generator signals input frequency fluctuates, take the shortest frequency, not average frequency, as "Frequency of FP" above.

<Setting relationship of PA and PB signals>

Note 1. When describing register bit in the right column of the following table, “n” refers to a bit position.

| | |
|---|---|
| PA and PB signal input <RENV2.PIM><br>00: 90 degree phase difference, 1x<br>(Count forward when the PA signal input phase is leading).<br>01: 90 degree phase difference, 2x<br>(Count forward when the PA signal input phase is leading).<br>10: 90 degree phase difference, 4x<br>(Count forward when the PA signal input phase is leading).<br>11: Count forward on the rising edge of PA.<br>Count backward on the rising edge of PB. | [RENV2] (WRITE)<br>15 8<br>n n - - - - - - |
| Function to reverse count direction by PA and PB signal input <RENV2.PDIR><br>0: Disabled.<br>1: Enabled. | [RENV2] (WRITE)<br>23 16<br>- - - - - - - n |
| Function to input PA and PB signals. <RENV2.POFF><br>0: Disabled<br>1: Enabled. | [RENV2] (WRITE)<br>23 16<br>- - - - - n - - |
| Input filter function of PA and PB signals. <RENV2.PINF><br>0: Enabled. Set a filter on PA/PB input. G9103C ignores signals shorter than 150ns.<br>1: Disabled. | [RENV2] (WRITE)<br>15 8<br>- - n - - - - - |
| Operation status <RSTS.CND><br>0101: Wait for PA and PB signal input. | [RSTS] (READ)<br>7 0<br>- - - - n n n n |
| Error interrupt status <REST.ESPE><br>1: When PA and PB signals changed simultaneously. (A motor does not stop). | [REST] (READ)<br>15 8<br>- n - - - - - - |
| Error interrupt status <REST.ESP0><br>1: When overflow of buffer counter to input PA and PB signals occurs. | [REST] (READ)<br>15 8<br>- - - - - - n - |

The pulse generator input mode has the following 14 operation types.

| RMD.MOD | Operation mode | Direction of movement |
|---|---|---|
| 01h | Continuous operation synchronized with PA and PB | Determined by the PA and PB input. |
| 51h | Positioning operation (incremental position) synchronized with PA and PB | Determined by the sign of RMV value. |
| 52h | Positioning operation (COUNTER 1 absolute position) synchronized with PA and PB | Determined by the relationship of RMV and RCUN1 register. |
| 53h | Positioning operation (COUNTER 2 absolute position) synchronized with PA and PB | Determined by the relationship of RMV and RCUN2 register. |
| 54h | Zero return operation of command position (COUNTER 1) synchronized with PA and PB | Determined by a sign of a value in COUNTER 1. |
| 55h | Zero return operation of mechanical position (COUNTER 2) synchronized with PA and PB | Determined by a sign of a value in COUNTER 2. |
| 68h | Continuous linear interpolation synchronized with PA and PB (output X axis pulse) | Determined by a direction of interpolation. |
| 69h | Linear interpolation synchronized with PA and PB (output X axis pulse) | Determined by a direction of interpolation. |
| 6Ch | CW directional circular linear interpolation synchronized with PA and PB (output X axis pulse) | Determined by a direction of interpolation. |
| 6Dh | CCW directional circular interpolation synchronized with PA and PB (output X axis pulse) | Determined by a direction of interpolation. |
| 78h | Continuous linear interpolation synchronized with PA and PB (output Y axis pulse) | Determined by a direction of interpolation. |
| 79h | Linear interpolation synchronized with PA and PB (output Y axis pulse) | Determined by a direction of interpolation. |
| 7Ch | CW directional circular interpolation synchronized with PA and PB (output Y axis pulse) | Determined by a direction of interpolation. |
| 7Dh | CCW directional circular interpolation synchronized with PA and PB (output Y axis pulse) | Determined by a direction of interpolation. |

Note: In the case of multiplication operation by RENV2.PIM bit and RENV5.PMG bit, a motor will stop immediately by STOP (0049h) command. In the case, output pulse amount may not be integral multiple of multiplication.
If you want to delay stop until output pulse amount become an integral multiple of multiplication, set remaining output function of pulse generator signal input to enabled (RENV5.PSTP = 1) and write stop (0049h) command.
If you want to stop operation before output pulse amount become integral multiple of multiplication, set remaining output function of pulse generator signal input to disabled (RENV5.PSTP = 0) and write STOP (0049h) command.
However, in interpolation mode (RMD.MOD = 68h, 69h, 6ch, 6Dh, 78h, 79h, 7Ch, 7Dh), a motor will stop immediately by STOP (0049h) command even though remaining output function of pulse generator signal input to enabled (RENV5.RSTP = 1).

### 6.3.1 Continuous operation synchronized with PA and PB

This mode allows continuous operation using pulse generator signals.
Without PA and PB signal connection change, feed direction can be changed by RENV2.PDIR bit.

RMD.MOD: 01h PA and PB synchronizing continuous operation

| Pulse generator input I/F | RENV2.PDIR | Feed direction | PA and PB signal input |
|---|---|---|---|
| 90 degree phase difference signal (1x, 2x, and 4x) | 0 | (+) direction | When PA signal phase leads PB signal phase. |
| | | (−) direction | When PB signal phase leads PA signal phase. |
| | 1 | (+) direction | When PB signal phase leads PA signal phase. |
| | | (−) direction | When PA signal phase leads PB signal phase. |
| Two-pulse signal | 0 | (+) direction | PA signal rising edge |
| | | (−) direction | PB signal rising edge |
| | 1 | (+) direction | PB signal rising edge |
| | | (−) direction | PA input rising edge |

The G9103C stops operation when EL signal in the current feed direction turns ON. However, the G9103C can be operated in the opposite direction without writing a start command again.
When the motor is stopped by EL turning ON, no error interrupt (MSTS.SERR) will occur and error interrupt status (REST) is not changed.
To release operation mode, write STOP (0049h) command.

### 6.3.2 Positioning operations synchronized with PA and PB (specify incremental position)

This positioning is synchronized with pulse generator input by using RMV setting as an absolute position data.
The direction of movement is determined by a sign of RMV register

RMD.MOD: 51h PA and PB synchronizing positioning operation (specify incremental position)

At the start, a RMV register value is loaded into RPLS register. When pulse generator signals are input, the G9103C outputs pulses and counts down a RPLS register value. When a RPLS register value reaches "0," operation stops and the G9103C ignores any further PA/PB input. If you try to start with setting RMV register to 0, the G9103C will not output any pulses and it will stop immediately.

### 6.3.3 Positioning operation synchronized with PA and PB signals (specify the absolute position to COUNTER 1)

This positioning is synchronized with pulse generator input by using RMV setting as the absolute value for RCUN1 register.
The direction of movement is determined by the relationship between a value in RMV and a value in RCUN1 register.

RMD.MOD : 52h PA and PB synchronizing positioning operation (specify absolute position)

At the start, the difference between a value in RMV and RCUN1 register is loaded into RPLS register. When PA and PB signals are input, the G9103C outputs pulses and counts down a RPLS register value. When a RPLS register value reaches "0," the G9103C ignores any further PA/PB input. If you try to start with RMV = RCUN1, the G9103C will not output any pulses and it will stop immediately.

### 6.3.4 Positioning operation synchronized with PA and PB signals (specify the absolute position in COUNTER 2)

The operation procedures are the same as positioning operation synchronized with PA and PB signals (RMD.MOD = 52h) except that this function uses RCUN2 register instead of RCUN 1 register.

RMD.MOD : 53h PA and PB synchronizing positioning operation (specify absolute position)

### 6.3.5 Command position zero return operation synchronized with PA and PB signals

This operation is synchronized with pulse generator signals and operates until a RCUN register value reaches 0. Command pulses and operation direction is automatically calculated internally by a RCUN register value at the start.

RMD.MOD: 54h command position zero return operation synchronized with PA and PB

When the RCUN1 register is set to zero and operation is started, the LSI will stop movement on the axis immediately, without outputting any command pulses.

### 6.3.6 Mechanical position zero return operation synchronized with PA and PB signals

The operation is the same as command position zero return operation synchronized with PA and PB signals (RMD.MOD = 54h) except that this function uses RCUN2 register instead of RCUN 1 register.

RMD.MOD : 55h mechanical position zero operation synchronized with PA and PB signals

### 6.3.7 Interpolated operation synchronized with PA and PB signals

This mode is used to operate interpolation synchronized with a pulse generator input.
The feed direction is the same as the normal interpolation. Even If a pulse is reversed, the direction of interpolation is not changed.

RMD.MOD : 68h continuous linear interpolation synchronized with PA and PB (X axis pulse)
69h linear interpolation synchronized with PA and PB (X axis pulse)
6Ch CW direction circular interpolation synchronized with PA and PB (X axis pulse)
6Dh CCW direction circular interpolation synchronized with PA and PB (X axis pulse)
78h Continuous linear interpolation synchronized with PA and PB (Y axis pulse)
79h Linear interpolation synchronized with PA and PB (Y axis pulse)
7Ch CW direction circular interpolation synchronized with PA and PB (Y axis pulse)
7Dh CCW direction circular interpolation synchronized with PA and PB (Y axis pulse)

For details, see “6.7 Interpolation operation mode”.

## *6.4 Origin position operation mode*

The following six origin position operation modes are available.

| RMD.MOD | Operation mode | Direction of movement |
|---|---|---|
| 10h | Origin return operation | (+) direction |
| 18h | Origin return operation | (−) direction |
| 12h | Leaving from origin position operation | (+) direction |
| 1Ah | Leaving from origin position operation | (−) direction |
| 15h | Origin search operation | (+) direction |
| 1Dh | Origin search operation | (−) direction |

Depending on operation method, origin position operation uses ORG, EZ, +EL or −EL signals.
Specify input logic of ORG signal in the RENV1 ORGL bit. This register's terminal status can be checked with RSTS.SORG bit.
Specify input logic of the EZ signal in RENV2.EZL bit. Specify the number of EZ count that indicates an origin return complete in RENV3.EZD bit. This register's terminal status can be checked by reading the RSTS.SEZ bit.
Specify logic for +EL and –EL signals using ELL terminals. Specify the operation to execute when the signal turns ON (immediate stop/deceleration stop) in RENV1.ELM bit. This terminal status can be monitored with RSTS.SPEL bit (+EL signal) and RSTS.SMEL bit (−EL signal).
An input filter can be applied to the ORG, EL and −EL signal in RENV1.FLTR bit.

| | |
|---|---|
| Input logic of ORG signal <RENV1.ORGL><br>0: Negative logic<br>1: Positive logic | [RENV1] (WRITE)<br>7 0<br>n - - - - - - - |
| Input status of ORG signal <RSTS.SORG><br>0: OFF<br>1: ON | [RSTS] (READ)<br>15 8<br>- - - - - - - n |
| Input type of EZ signal <RENV2.EZL><br>0: Falling edge<br>1: Rising edge | [RENV2] (WRITE)<br>15 8<br>- - - n - - - - |
| EZ count value used in origin return <RENV3.EZD><br>0000 ~ 1111 : 1st ~ 16th (e.g. 0100b = 5th) | [RENV3] (WRITE)<br>7 0<br>n n n n - - - - |
| Input status of EZ signal <RSTS.SEZ><br>0: OFF<br>1: ON | [RSTS] (READ)<br>23 16<br>- - - - - - - n |
| Input logic of EZ signal <ELL input terminal><br>L: Positive logic<br>H: Negative logic | |
| Processing used when the +EL or −EL signal turns ON <RENV1.ELM><br>0: Stop immediately.<br>1: Decelerate and stop. | [RENV1] (WRITE)<br>7 0<br>- - - - n - - - |
| Input status of +EL signal <RSTS.SPEL><br>0: OFF<br>1: ON | [RSTS] (WRITE)<br>7 0<br>- n - - - - - - |
| Input status of −EL signal <RSTS.SMEL><br>0: OFF<br>1: ON | [RSTS] (WRITE)<br>7 0<br>n - - - - - - - |
| Input filter function of +EL, −EL and ORG signals <RENV1.FLTR><br>0: Enabled. Pulses shorter than 4 µs will be ignored.<br>1: Disabled. | [RENV1] (WRITE)<br>31 24<br>- - - - - - n - |

### 6.4.1 Origin return operation

After writing a start command, a motor will continue feeding until an origin return complete condition are met.

RMD.MOD: 10h (+) direction origin return operation
18h (−) direction origin return operation

When an origin return is completed, the LSI will clear the counter at the origin position and output an ERC (deviation counter clear) signal while the motor is stopping.

RENV3 register is used to set a basic origin return method and whether or not to clear the counter when the origin return is completed. RENV1 register is used to set whether or not to output an ERC signals.

For details about the ERC signal, see “8.6.2 ERC signal”.

Select an origin return method. <RENV3.ORM>

0000: Origin return operation 0
- When ORG signal turns ON, a motor will stop immediately at constant start or decelerate and stop at high speed start.
- COUNTER clear timing: When ORG signal turns ON.

0001: Origin return operation 1
- After ORG signal turns ON, a motor will stop immediately at constant start or decelerate and stop at high speed start. Then, it feeds in the opposite direction at FA constant speed until ORG signal turns OFF. Then, the motor will move back in the original direction at FA speed and stop immediately when ORG signal turns ON again.
- COUNTER clear timing: When ORG signal turns ON after move at FA speed.

0010: Origin return operation 2
- A motor operates until ORG signal turns ON, a motor will stop immediately at constant start when the LSI finish counting the specified number of EZ pulses. At high speed start, the motor decelerates during counting and stops when the LSI finishes counting the specified number of EZ pulses.
  COUNTER clear timing: When finishing counting the specified number of EZ pulses.

0011: Origin return operation 3
- A motor operates until ORG signal turns ON. A motor will stop immediately at constant start or decelerate and stop at high speed start when the LSI finishes counting the specified number of EZ pulses,
  COUNTER clear timing: When finishing counting the specified number of EZ pulses.

0100: Origin return operation 4
- After the ORG signal turns ON, a motor will stop immediately at constant start or decelerate and stop at high speed start. Then, it feeds in the opposite direction at FA constant speed until the ORG signal turns OFF. Then, when the LSI finishes counting the specified number of EZ pulses, the motor will stop immediately at constant start or decelerate and stop at high speed start
- COUNTER clear timing: When finishing counting the specified number of EZ pulses.

0101: Origin return operation 5
- After the ORG signal turns ON, a motor will stop immediately at constant start or decelerate and stop at high speed start. Then, it feeds in the opposite direction at FA constant speed until ORG input turns OFF. Then, when the LSI finishes counting the specified number of EZ pulses, the motor will stop immediately at constant start or decelerate and stop at high start speed.
- COUNTER clear timing: When finishing counting the specified number of EZ pulses.

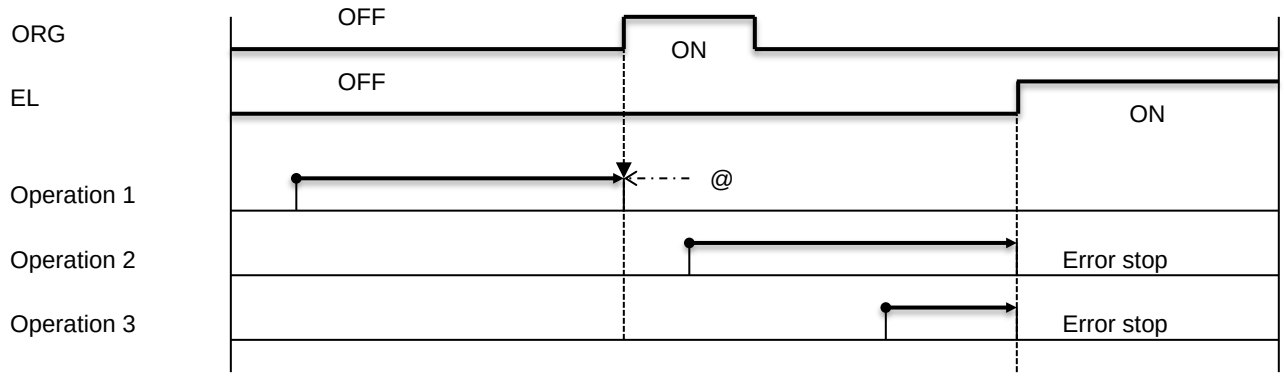
[RENV3] (WRITE)

| 7 | | | | | | | 0 |
|---|---|---|---|---|---|---|---|
| - | - | - | - | n | n | n | n |

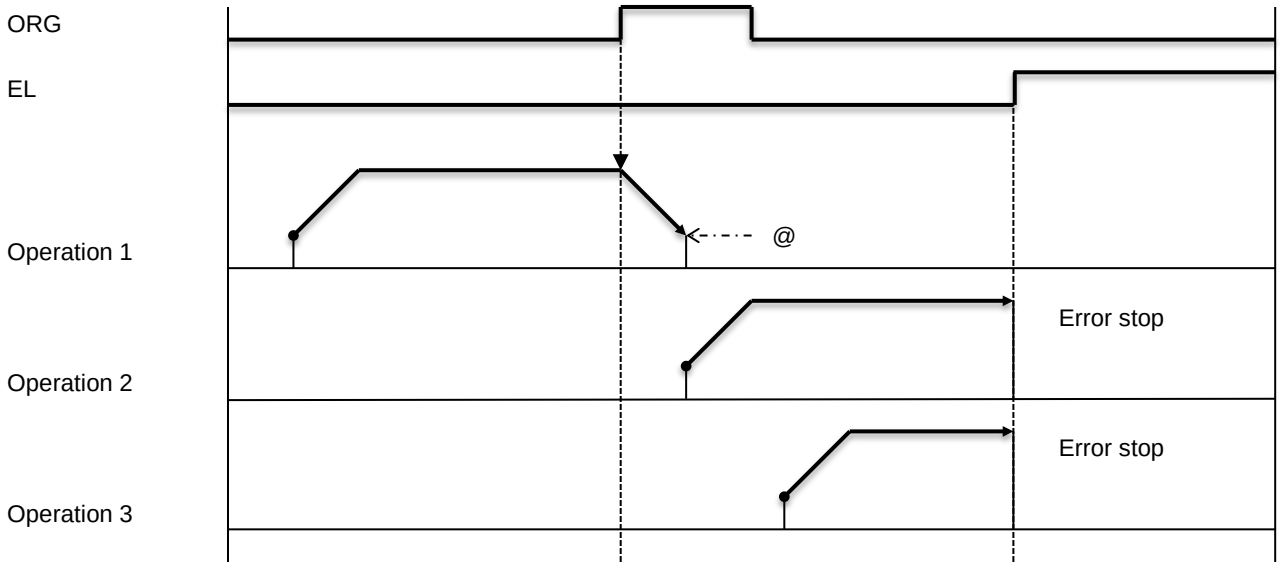
| | |
|---|---|
| 0110: Origin return operation 6<br>- After the ORG signal turns ON, a motor will stop immediately at constant start or decelerate and stop at high speed start and RENV1.ELM = 1. Then, it feeds in the opposite direction at FA constant speed until ORG input turns OFF and stops immediately.<br>- COUNTER clear timing: When the EL signal is OFF.<br><br>0111: Origin return operation 7<br>- After the ORG input turns ON, a motor will stop immediately at constant start or decelerate and stop at high speed start and RENV1.ELM = 1. Then, it feeds in the opposite direction until the ORG input turns OFF. Then, when the LSI finishes counting the specified number of EZ pulses, the motor will stop immediately.<br>- COUNTER clear timing: When stopped by finishing counting the specified number of EZ pulses.<br><br>1000:Origin return operation 8<br>- After the ORG input turns ON, a motor will stop immediately at constant start or decelerate and stop at high speed start and RENV1.ELM = 1. Then, the motor will start feeding in the opposite direction. 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 motor will stop immediately and decelerates and stops at high start speed.<br>- CONTER clear timing: When finishing counting the specified number of EZ signal.<br><br>1001: Origin return operation 9<br>- After origin return operation 0 has executed, a motor operates until COUNTER 2 = 0.<br><br>1010: Origin return operation 10<br>- After origin return operation 3 has executed, a motor operates until COUNTER 2 = 0.<br><br>1011: Origin return operation 11<br>- After origin return operation 5 has executed, a motor operates until COUNTER 2 = 0.<br><br>1100: Origin return operation 12<br>- After origin return operation 8 has executed, a motor operates until COUNTER 2 = 0). | |
| Clear COUNTER 1 at origin position in origin return operation <RENV3.CU1R><br>0: Disabled.<br>1: Enabled. | [RENV1] (WRITE)<br>23 16<br>- - - n - - - - |
| Clear COUNTER 2 at origin position in origin return operation <RENV3.CU2R><br>0: Disabled.<br>1: Enabled. | [RENV3] (WRITE)<br>23 16<br>- - n - - - - - |
| Clear COUNTER 3 at origin position in origin return operation <RENV3.CU3R><br>0: Disabled.<br>1: Enabled. | [RENV3] (WRITE)<br>23 16<br>- n - - - - - - |
| Function to output ERC signal automatically at origin return completed <RENV1.EROR><br>0: Disabled.<br>1: Enabled. | [RENV1] (WRITE)<br>15 8<br>- - - - n - - - |

6.4.1.1 Origin return operation 0 (RENV3.ORM = 0000b)

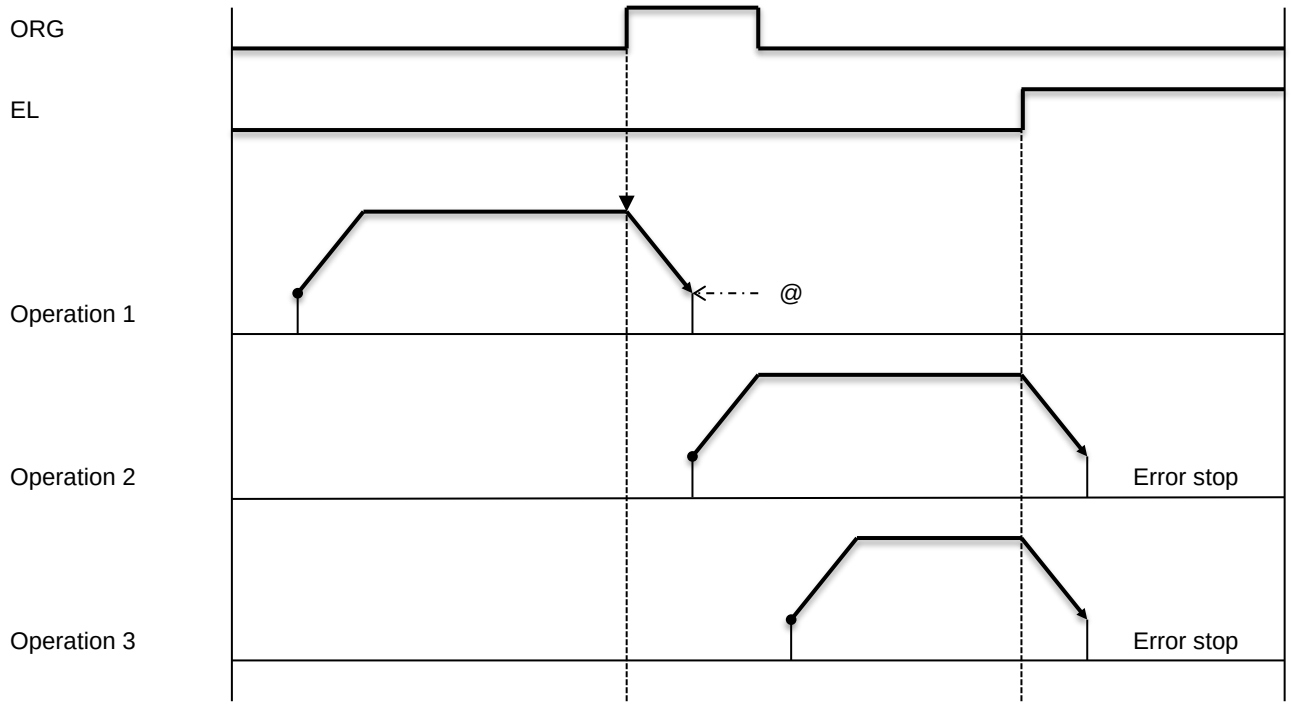
☐ Constant speed operation <Sensor: EL (RENV1.ELM = 0), ORG>
[Starting from here, ☐ indicates constant speed operation, and ■ indicates high speed operation.]



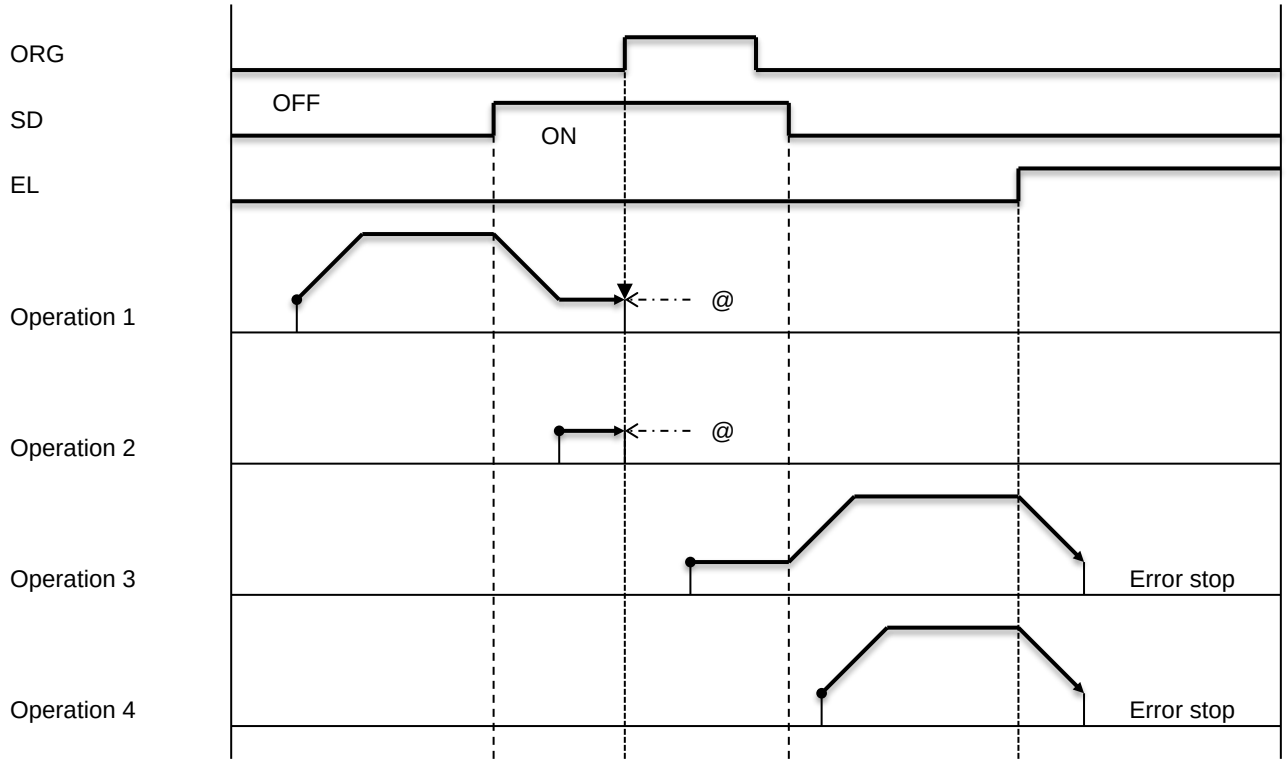

■ High speed operation <Sensor: EL (RENV1.ELM = 0), ORG>
Even if a motor stops normally, stop position may not be at the origin position. However, COUNTER 2 (mechanical position) provides a reliable value



■ High speed operation <Sensor: EL (RENV1.ELM = 1), ORG>
Even if a motor stops normally, stop position may not be at the origin position. However, COUNTER 2 (mechanical position) provides a reliable value.



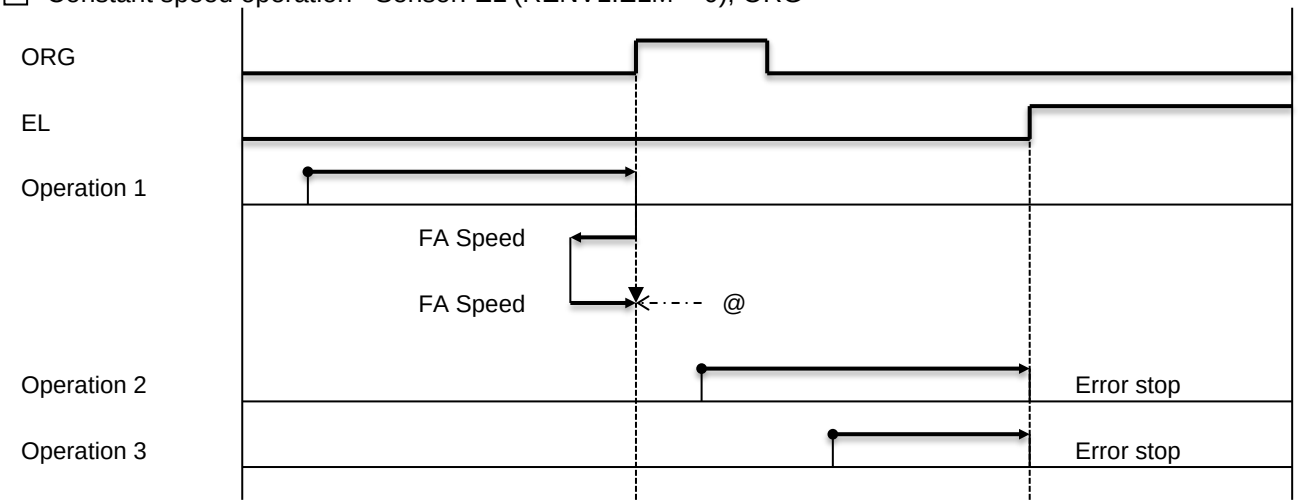

■ High speed operation <Sensor: EL (RENV1.ELM = 1), SD (RENV2.SDM = 0, RENV1.SDLT = 0), ORG>



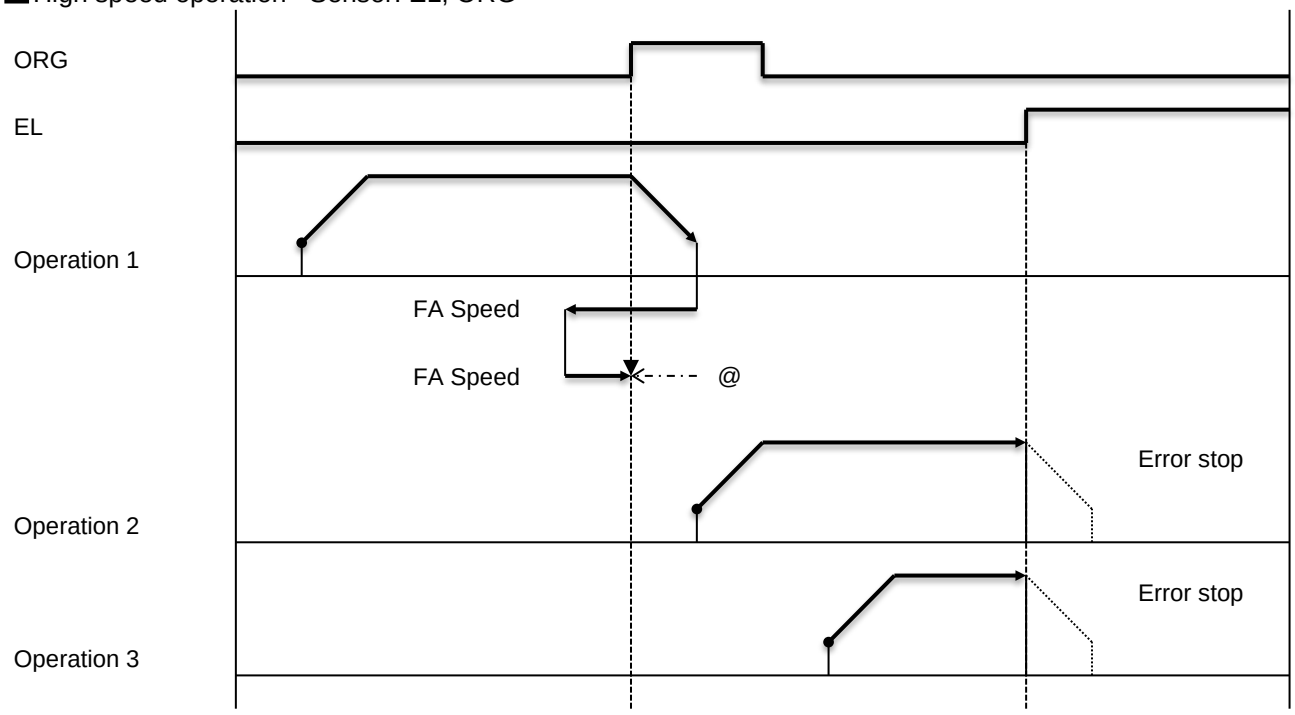

Note: The position marked with ▼reflects a counter clear timing and @ reflects an ERC signal output timing.

6.4.1.2 Origin return operation 1 (RENV3.ORM=0001b)

☐ Constant speed operation <Sensor: EL (RENV1.ELM = 0), ORG>



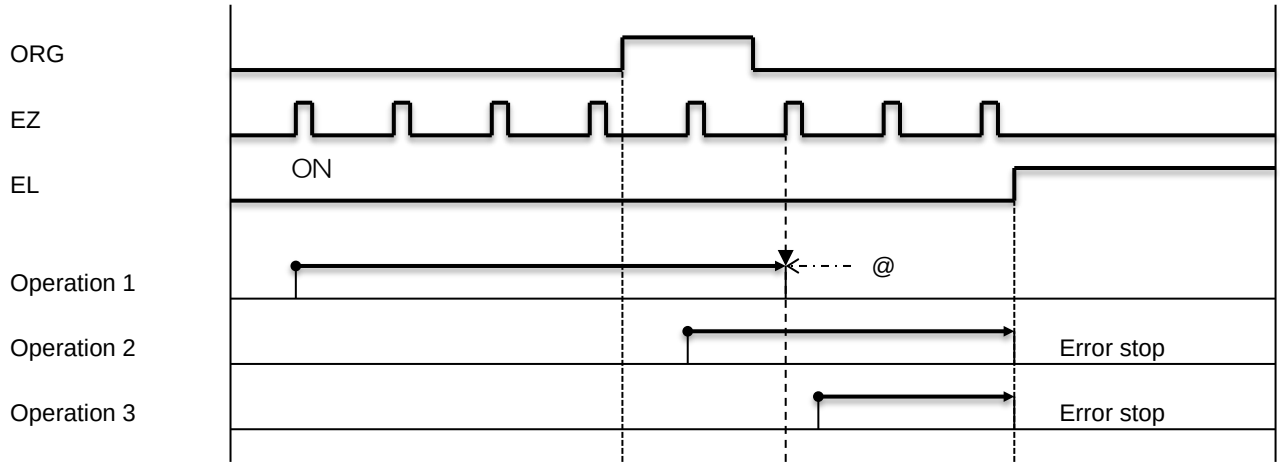

■High speed operation <Sensor: EL, ORG>



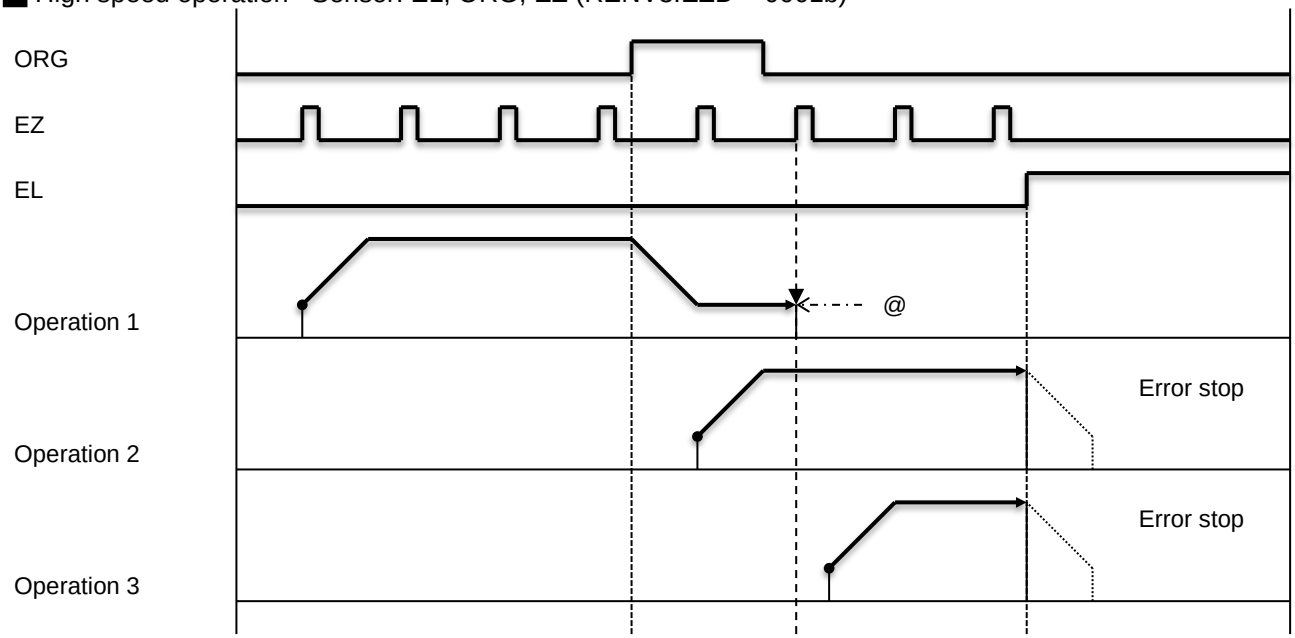

Note: The position marked with ▼reflects a counter clear timing and @ reflects an ERC signal output timing.

6.4.1.3 Origin return operation 2 (RENV3.ORM = 0010b)

☐ Constant speed operation <Sensor: EL (RENV1.ELM = 0), ORG, EZ (RENV3.EZD = 0001b)>



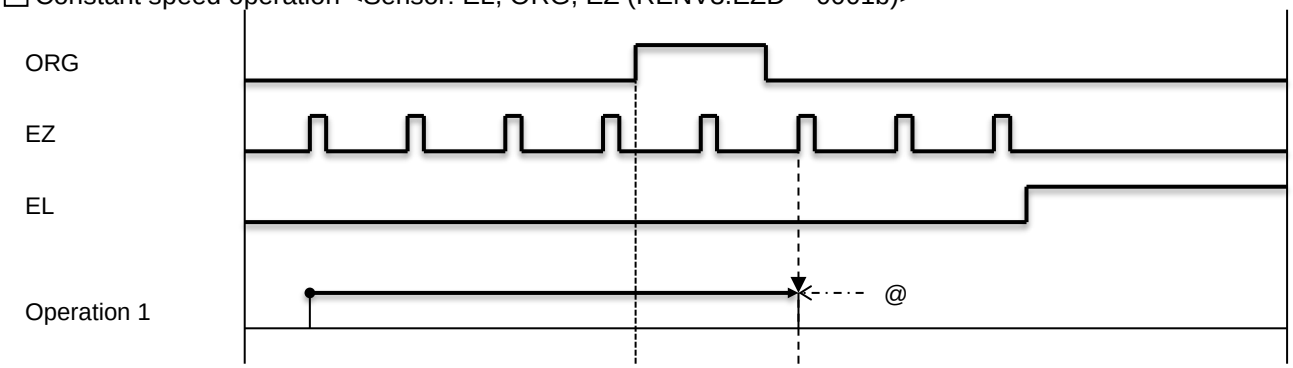

■ High speed operation <Sensor: EL, ORG, EZ (RENV3.EZD = 0001b)>



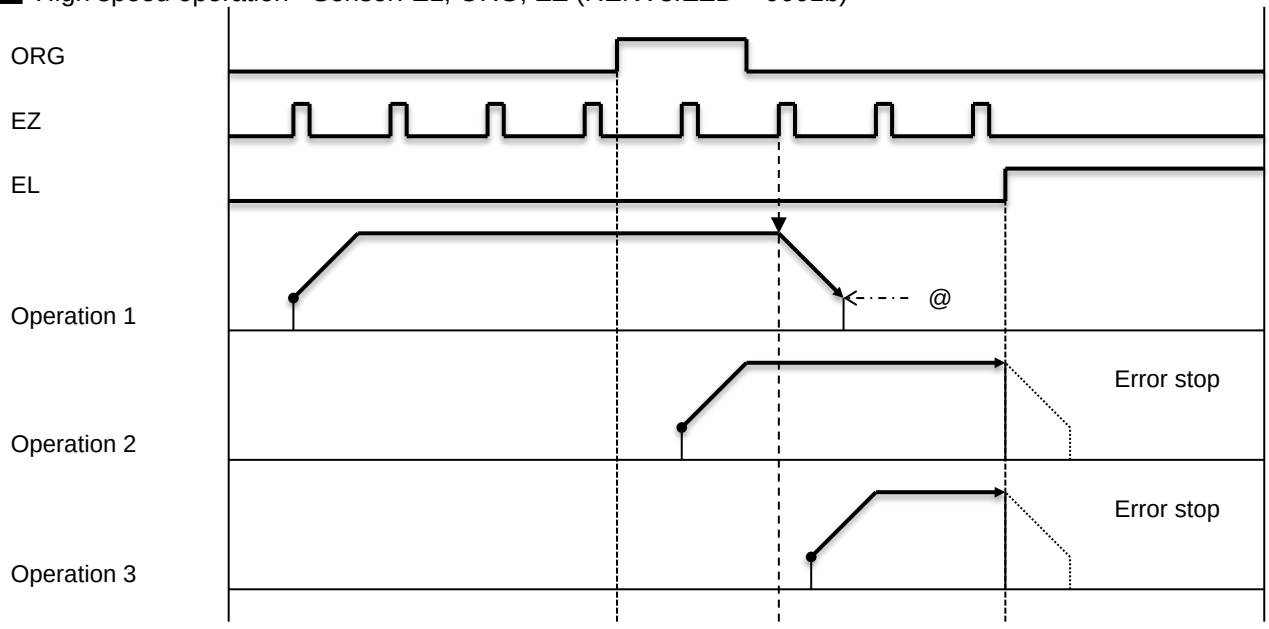

Note: The position marked with ▼ reflects a counter clear timing and @ reflects an ERC signal output timing.

6.4.1.4 Origin return operation 3 (RENV3.ORM = 0011b)

☐ Constant speed operation <Sensor: EL, ORG, EZ (RENV3.EZD = 0001b)>



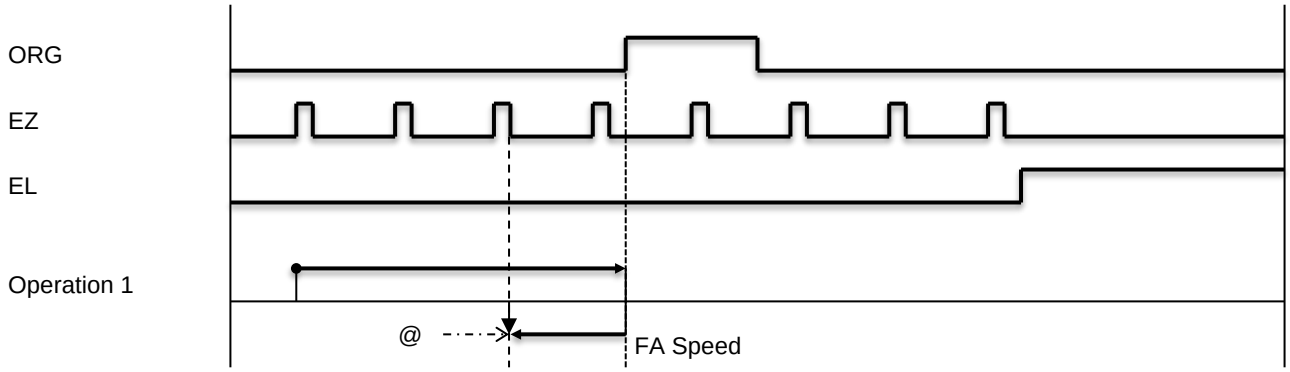

■ High speed operation <Sensor: EL, ORG, EZ (RENV3.EZD = 0001b)>



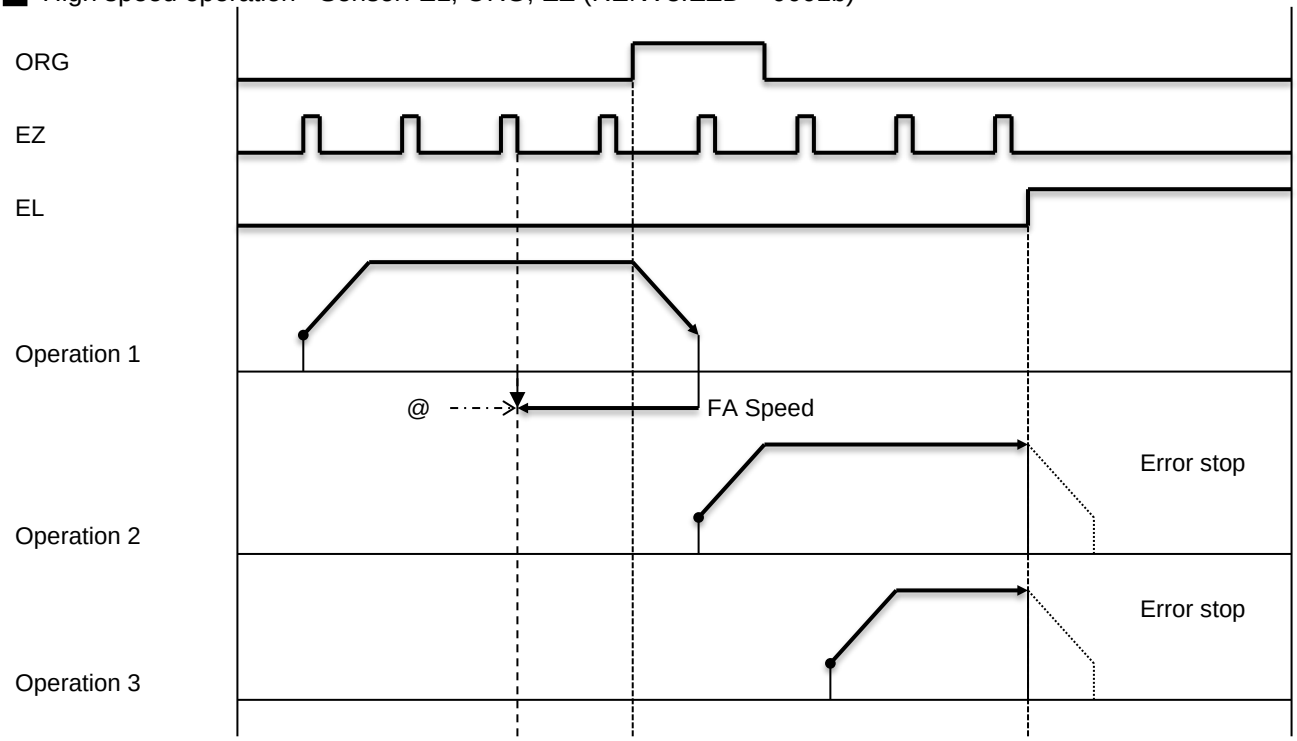

Note: The position marked with ▼reflects a counter clear timing and @ reflects an ERC signal output timing.

6.4.1.5 Origin return operation 4 (RENV3.ORM = 0100b)

□ Constant speed operation <Sensor: EL, ORG, EZ (RENV3.EZD = 0001b)>



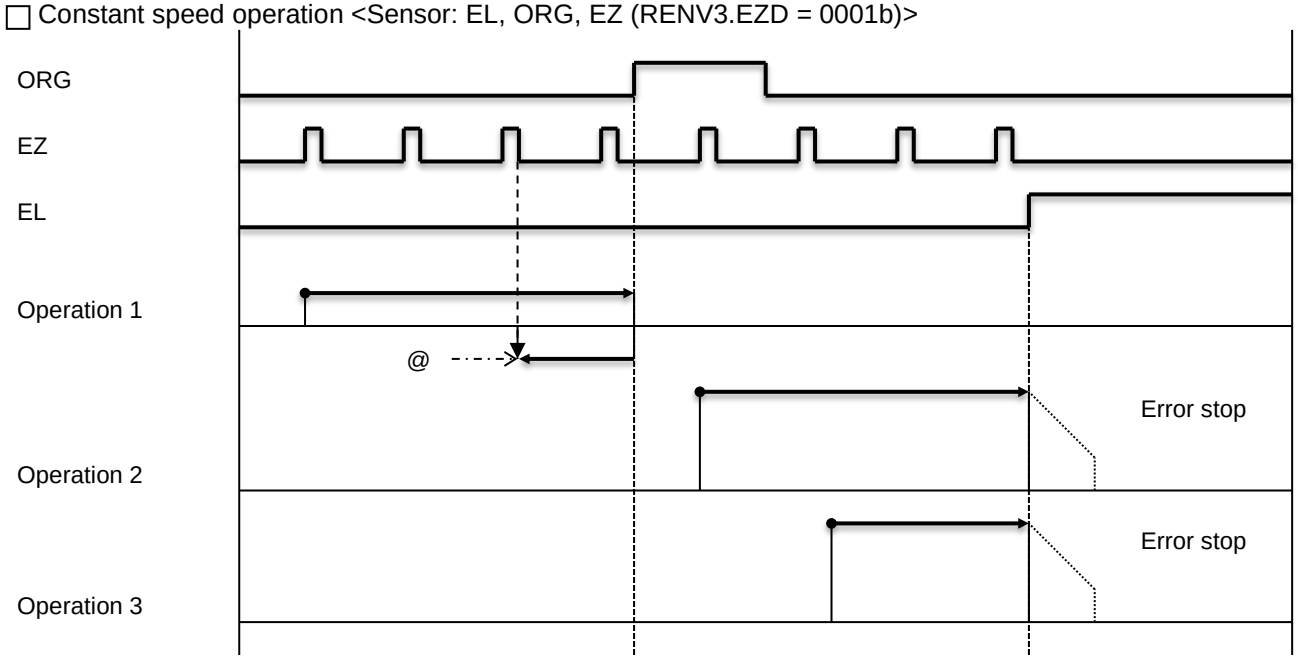

■ High speed operation <Sensor: EL, ORG, EZ (RENV3.EZD = 0001b)>



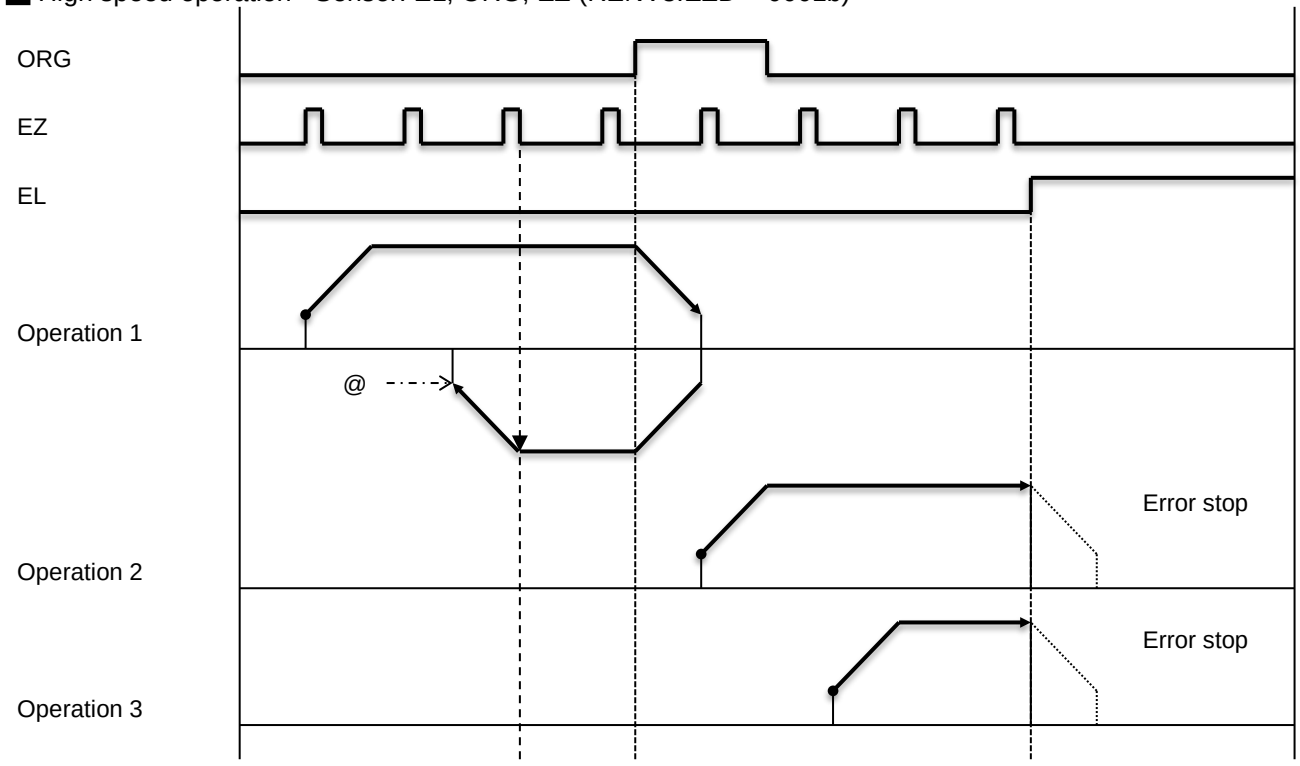

Note: The position marked with ▼reflects a counter clear timing and @ reflects an ERC signal output timing.

6.4.1.6 Origin return operation 5 (RENV3.ORM = 0101b)

□ Constant speed operation <Sensor: EL, ORG, EZ (RENV3.EZD = 0001b)>




■ High speed operation <Sensor: EL, ORG, EZ (RENV3.EZD = 0001b)>

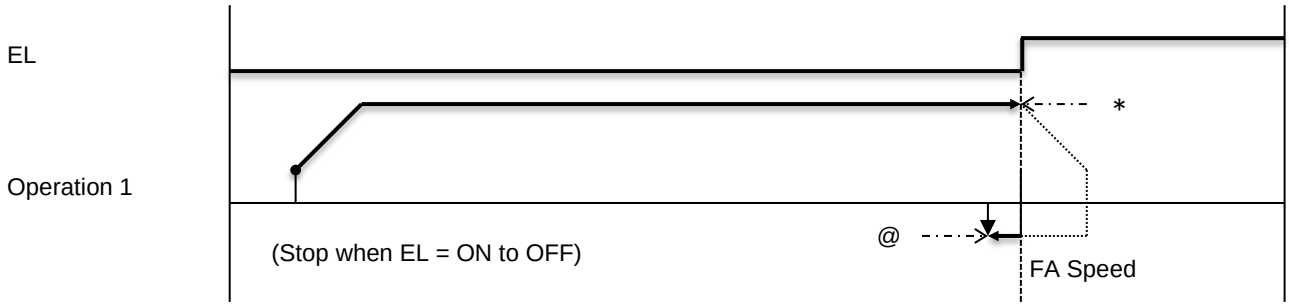



Note: The position marked with ▼ reflects a counter clear timing and @ reflects an ERC signal output timing.

6.4.1.7 Origin return operation 6 (RENV3.ORM = 0110b)

☐ Constant speed operation <Sensor: EL>

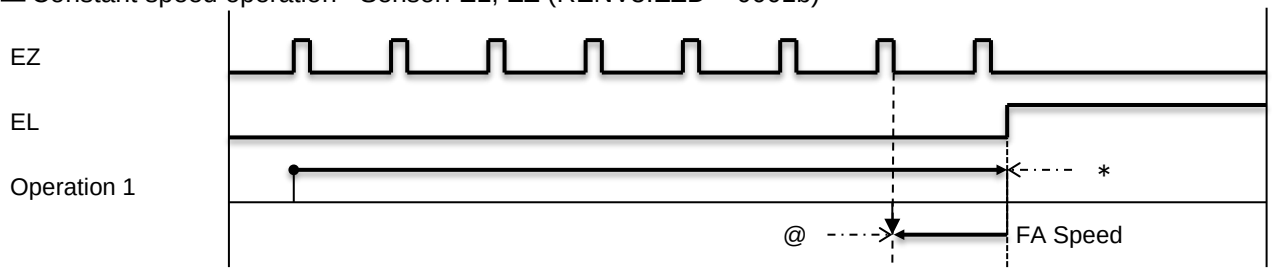

■ High speed operation <Sensor: EL>



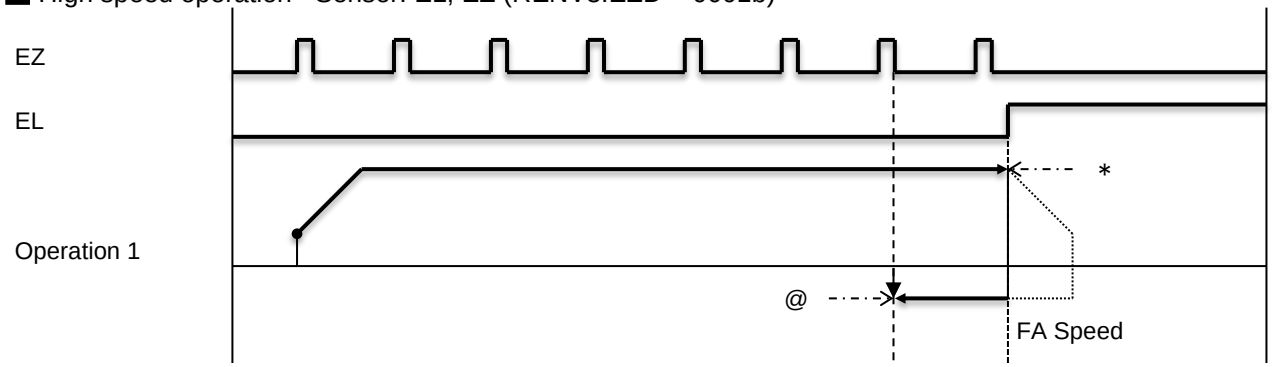

Note: The position marked with ▼ reflects a counter clear timing and @ reflects an ERC signal output timing. Also, when ERC auto output function is enabled (RENV1.EROE = 1), the LSI will output ERC signal at a position marked with an asterisk (*).

6.4.1.8 Origin return operation 7 (RENV3.ORM = 0111b)

☐ Constant speed operation <Sensor: EL, EZ (RENV3.EZD = 0001b)>



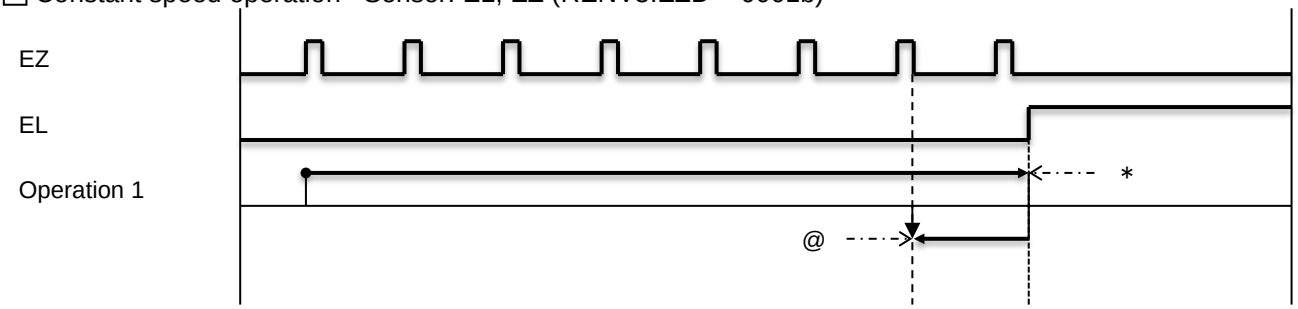

■ High speed operation <Sensor: EL, EZ (RENV3.EZD = 0001b)>



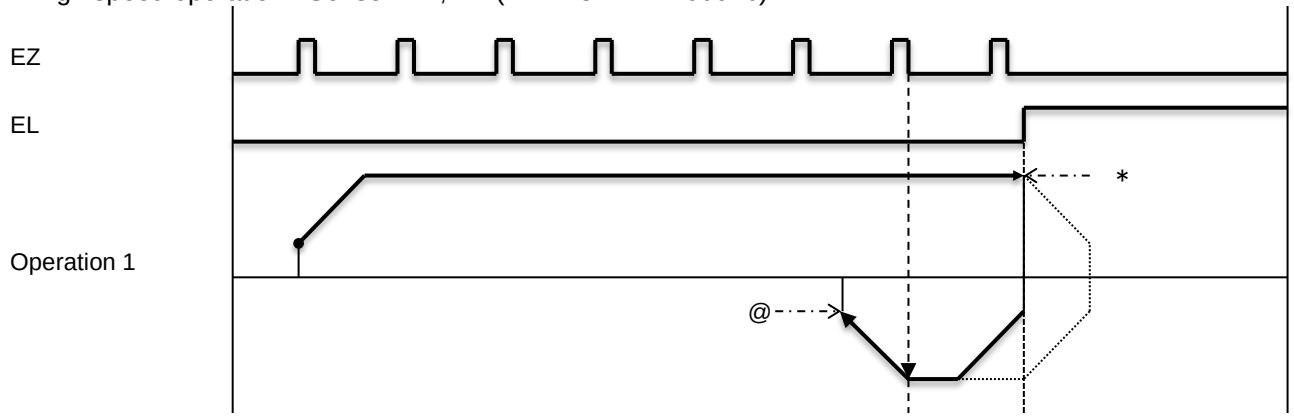

Note: The position marked with ▼ reflects a counter clear timing and @ reflects an ERC signal output timing. Also, when ERC auto output function is enabled (RENV1.EROE = 1), the LSI will output ERC signal at a position marked with an asterisk (*).

6.4.1.9 Origin return operation 8 (RENV3.ORM = 1000b)

☐ Constant speed operation <Sensor: EL, EZ (RENV3.EZD = 0001b)>



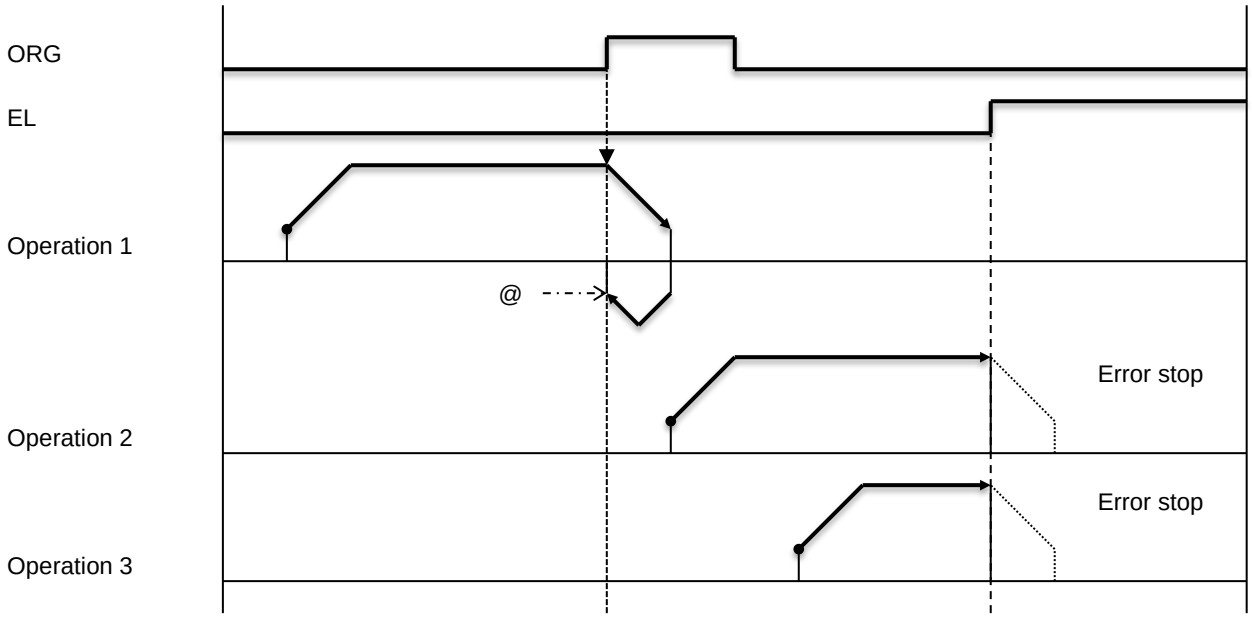

■ High speed operation <Sensor: EL, EZ (RENV3.EZD = 0001b)>




Note: The position marked with ▼ reflects a counter clear timing and @ reflects an ERC signal output timing. Also, when ERC auto output function is enabled (RENV1.EROE=1), the LSI will output ERC signal at a position marked with an asterisk (*).

6.4.1.10 Origin return operation 9 (RENV3.ORM = 1001b)

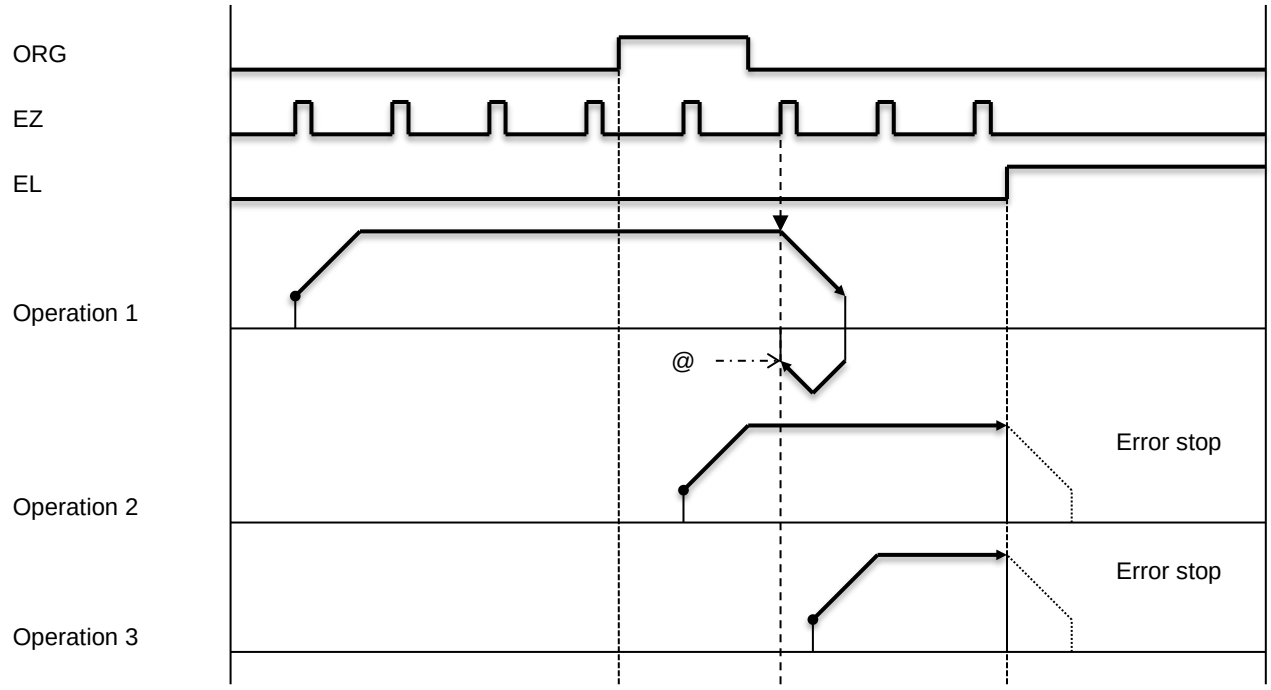
■ High speed operation <Sensor: EL, ORG)>




Note: The position marked with ▼reflect a counter reset timing and @ reflects an ERC signal output timing.

6.4.1.11 Origin return operation 10 (RENV3.ORM = 1010b)

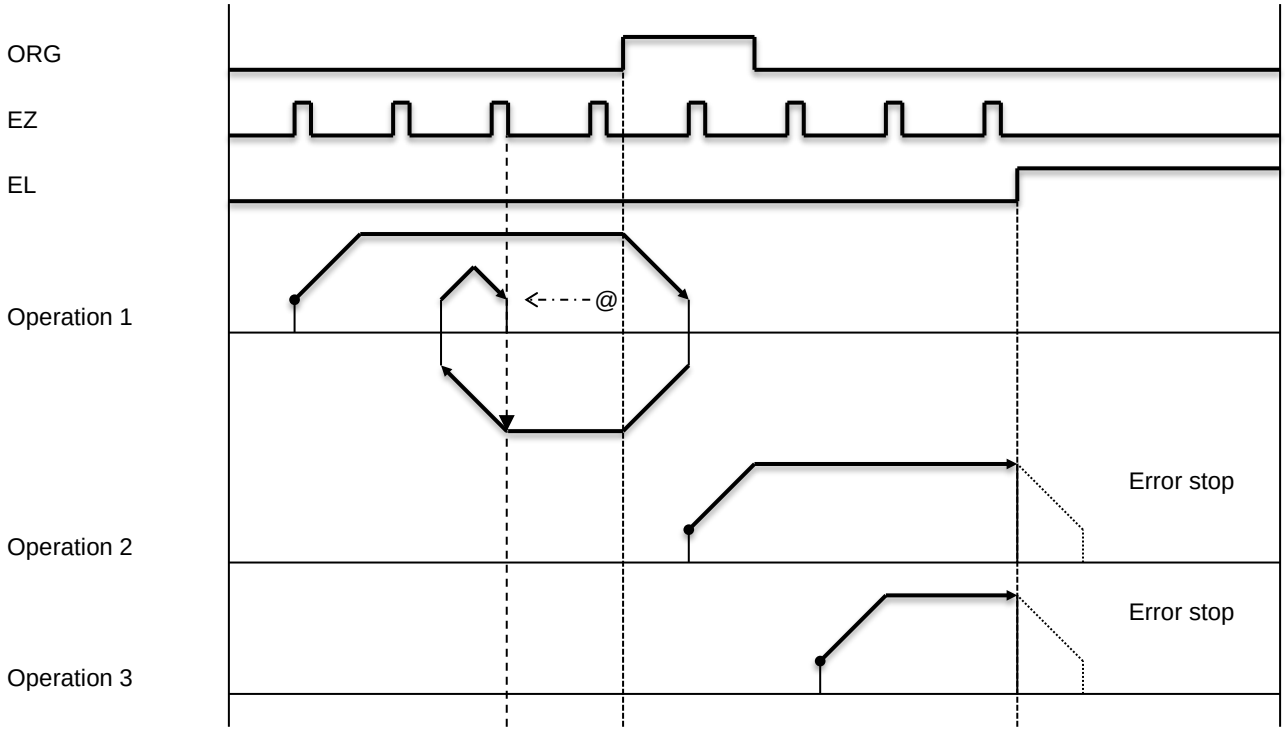
■ High speed operation <Sensor: EL, ORG, EZ (RENV3.EZD = 0001b)>




Note: The position marked with ▼reflects a counter clear timing and @ reflects an ERC signal output timing.

6.4.1.12 Origin return operation 11 (REMV3/ORM = 1011b)

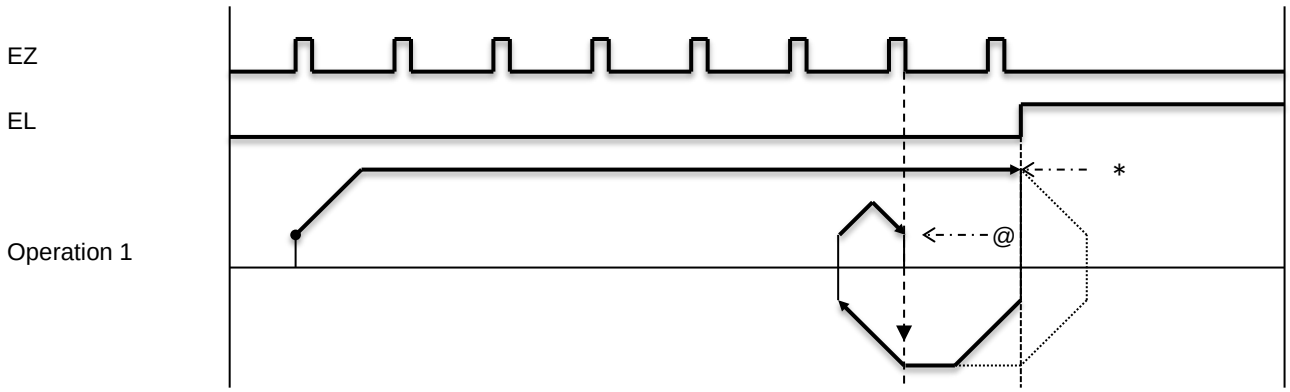
■ High speed operation <Sensor: EL, ORG, EZ (RENV3.EZD = 0001b)>




Note: The position marked with ▼ reflects a counter clear timing and @ reflects an ERC signal output timing.

6.4.1.13 Origin return operation 12 (RENV3.ORM = 1100b)

■ High speed operation <Sensor: EL, EZ (RENV3.EZD = 0001b)>



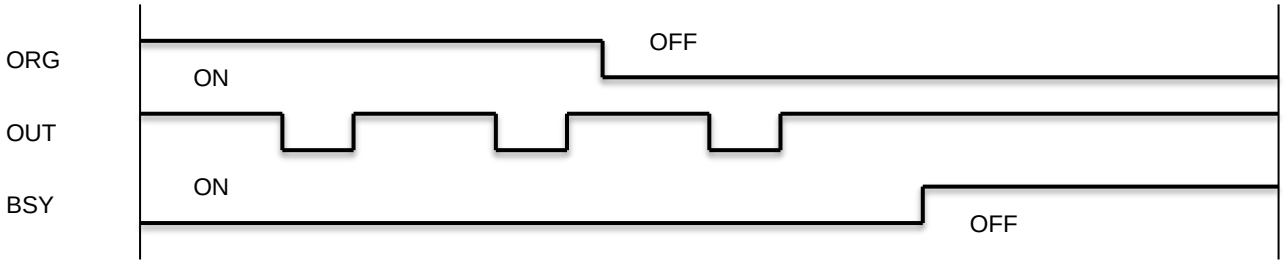

Note: The position marked with ▼ reflect the counter reset timing and @ reflects the ERC signal output timing. Also, when EROE (bit 10) is 1 in the RENV1 register and ELM (bit 3) is 0, the LSI will output an ERC signal at positions marked with an asterisk (*).

### 6.4.2 Leaving from the origin position operations

After writing a start command, a mechanical position will leave the origin position (ORG signals ON).

RMD.MOD:12h; escape from origin position operation in (+) direction
1Ah; escape from origin position operation in (−) direction

Make sure to use a constant start command STAFL (0050h) or STAFH (0051h) for leaving from the origin position.
When you write a start command while the ORG signal OFF, the LSI will stop the movement on the axis as a normal stop, without outputting any pulses.
If a start command is written while the ORG signal is ON, the G9103C will stop operation after outputting one pulse after the ORG input is turned OFF (Normal stop), since input status of ORG signal is sampled when the G9103C is outputting pulses.




### 6.4.3 Origin search operation

This is a mode that a function is added to an origin return operation.

RMD.MOD:15h; origin search operation in (+) direction
1Dh; origin search operation in (−) direction

It consists of the following possibilities.

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

Operation 1: If ORG signal is OFF at the start, a motor operates origin return operation in the specified direction.

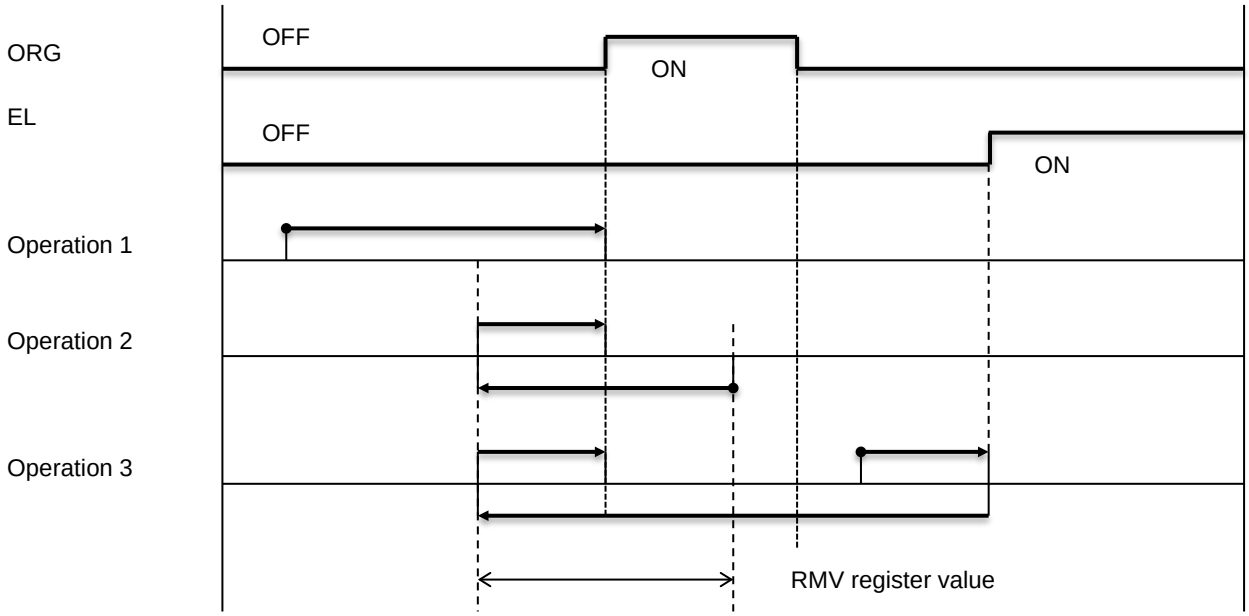
Operation 2: If ORG signal is already turned ON at the start, the motor will operate leaving from the origin position using positioning operations in the opposite direction to the one specified.
After the ORG signal turns OFF, the motor will operate origin return operation in the specified direction.

Operation 3: If ORG signal is OFF at the start, the motor operates origin return operation in the specified direction.
If the motor is stopped by an EL signal while operating in the specified direction, the LSI will execute an "origin return operation" in the opposite direction to the one specified. After that, it will execute "leaving from the origin position by positioning" in the opposite direction to the one specified. After the ORG signal turns OFF, it will execute an "origin return operation" in the specified direction.

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

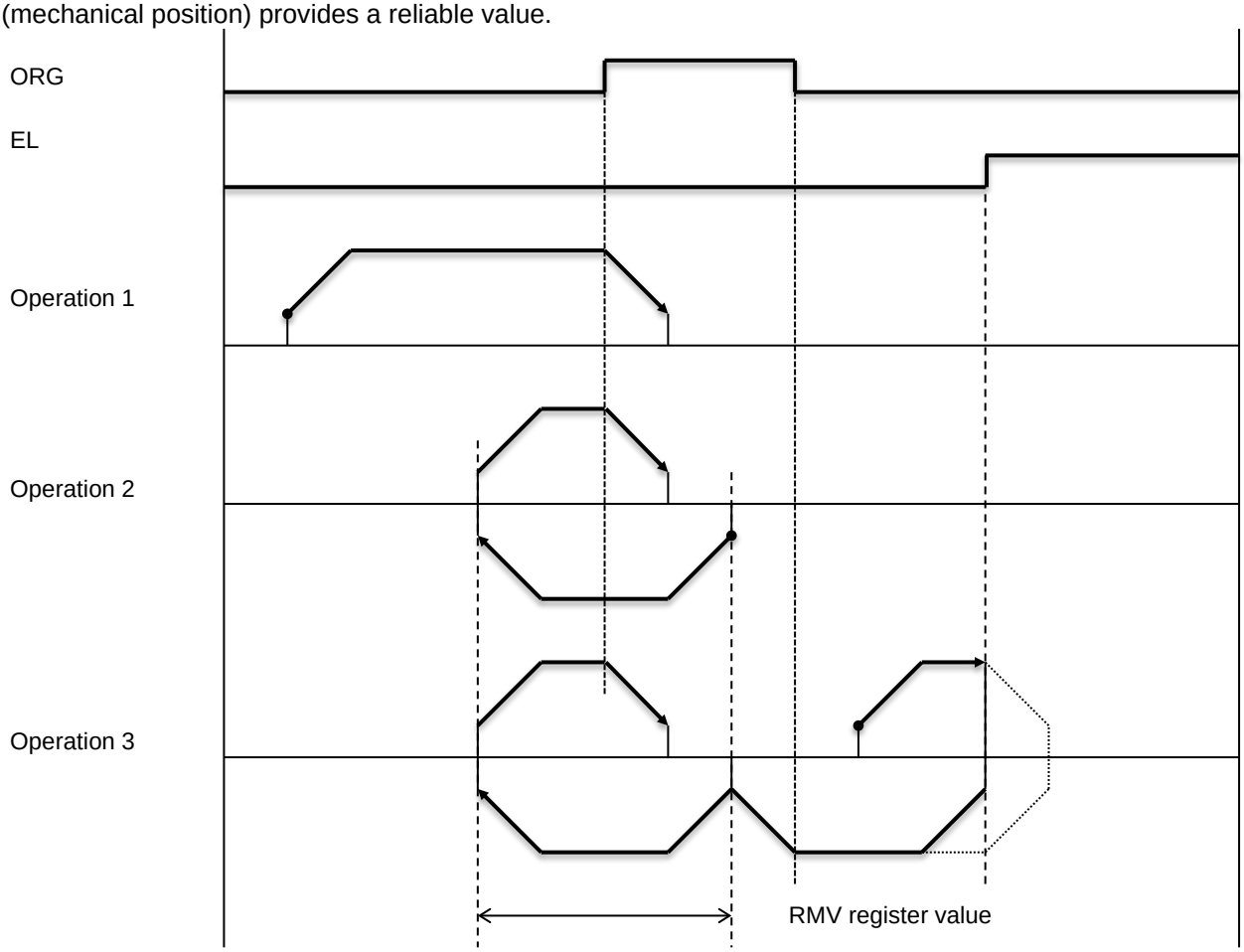
6.4.3.1 Origin return operation 0 (RENV3.ORM = 0000b)

☐ Constant speed operation <Sensor: EL, ORG>




■ High speed operation <Sensor: EL, ORG>

Even if a motor stops normally, stop position 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 end limit (EL) or software limit (SL) operation are available.

| RMD.MOD | Operation mode | Direction of movement |
|---|---|---|
| 20h | Operate until reaching +EL or +SL position. | (+) direction |
| 28h | Operate until reaching –EL or –SL position. | (–) direction |
| 22h | Leave from –EL or –SL positions. | (+) direction |
| 2Ah | Leave from +EL or +SL positions. | (–) direction |

To specify the EL signal (+EL signal in + direction operation and +EL signal in – direction operation), set input logic using ELL terminal.
Select a stop method (immediate stop or decelerate and stop) to use when EL signal is ON in RENV1.ELM bit.
Input status of +EL signal can be monitored by RSTS.SPEL bit and input status of –EL signal can be checked by RSTS.SMEL bit.
For details about setting the SL, see section “8.11.2 Software limit function”.

| | |
|---|---|
| Input logic of EL signal <ELL terminal><br>L: Positive logic.<br>H: Negative logic. | |
| Stop method to use when EL signal is turned ON <RENV1.ELM><br>0: Stop immediately.<br>1: Decelerate and stop. | [RENV1] (WRITE)<br>7 0<br>- - - - n - - - |
| Input status of +EL signal <RSTS.SPEL><br>0: OFF.<br>1: ON. | [RSTS] (READ)<br>7 0<br>- n - - - - - - |
| Input status of –EL signal <RSTS. SMEL><br>0: OFF.<br>1: ON. | [RSTS] (READ)<br>7 0<br>n - - - - - - - |
| EL signal input filter function <RENV1.FLTR><br>0: Enabled. Signals shorter than 4 µs will be ignored.<br>1: Disabled. | [RENV1] (WRITE)<br>31 24<br>- - - - - - n - |

### 6.5.1 Feed until reaching an EL or SL position

This mode is used to continue feeding until reaching EL or SL position.

RMD.MOD: 20h; Feed until reaching +EL or +SL position.
28h; Feed until reaching –EL or –SL position.

When a start command is written at the position where the +EL or –EL signal turns ON or within the range of +SL or -SL, the LSI will not command pulses and the motor will stop normally.
When a start command is written at the position where the +EL or –EL signal turns OFF or outside the range of +SL or -SL, the motor operates until +EL or –El signal turns ON, or reaching the range of +SL or –SL.

### 6.5.2 Leaving from an EL or SL position

This mode is used to continue feeding until the EL or SL signal turns OFF.

RMD.MOD: 22h; Leave a –EL or –SL position
2Ah; Leave a +EL or +SL position

When a start command is written at the position where the +EL or –EL signal turns OFF and outside the range of +SL or –SL, the LSI will not output pulses and the motor will stop normally.
When a start command is written at the position where the +EL or –EL signal turns ON or within the range of +SL or –SL, the motor operates until +EL or –El signal turns OFF, or leaving from the range of +SL or –SL.

## *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: 24h; Feed in (+) direction until the EZ count is completed.
2Ch; Feed in (–) direction until the EZ count is completed.

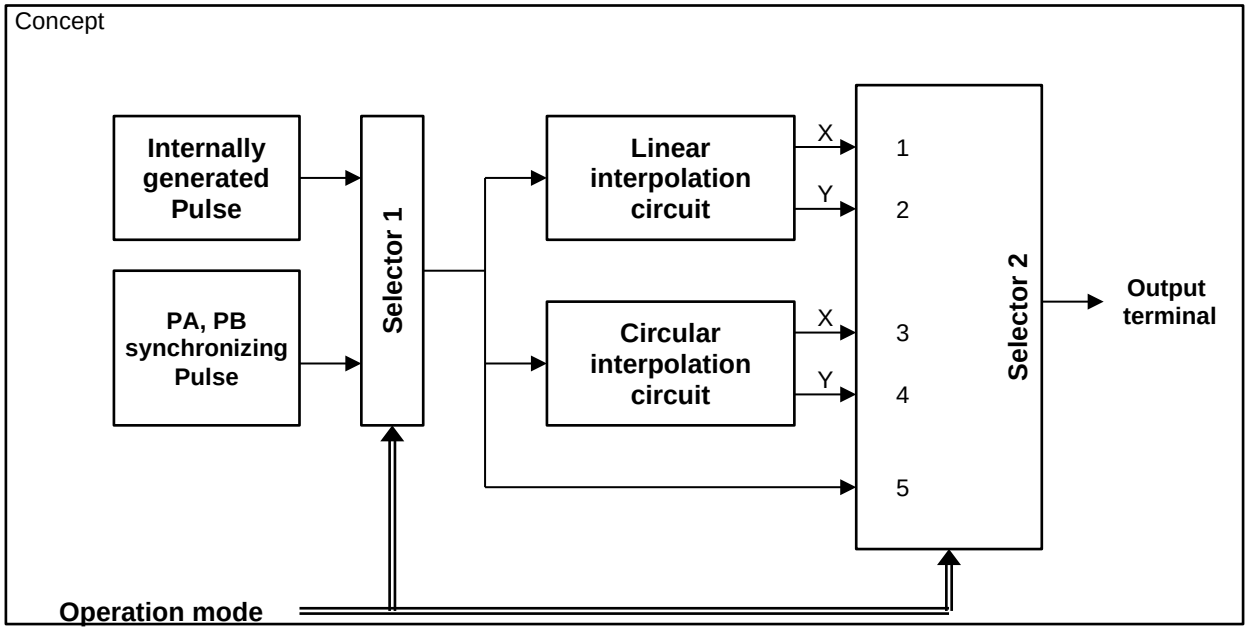
After a start command is written, a motor stops immediately or decelerates and stops after the EZ count equals the number stored in the register.
An EZ count can be set from 1 ~ 16.
If you want a motor to stop at the point where EZ signal turns ON, use constant speed start commands STAFL (0050h) or STAFH (0051h). When the high speed start command STAD (0052h) or STAUD (0053h) are used, the motor will start decelerating when the EZ signal turns ON and stop. Therefore, a mechanical position overruns the position where the EZ signals turns ON.
Specify input logic for EZ signal in RENV2.EZL bit, and EZ count number in RENV3.EZD bit. The input status can be checked by reading RSTS.SEZ bit.

| | |
|---|---|
| Input specification of EZ signal <RENV2.EZL><br>0: Falling edge<br>1: Rising edge | [RENV2] (WRITE)<br>15 8<br>- - - n - - - - |
| EZ count number to use for origin return <RENV3.EZD><br>0000 ~ 1111: 1 ~ 16th. (e.g. 0100b = 5th). | [RENV3] (WRITE)<br>7 0<br>n n n n - - - - |
| Input status of EZ signal <RSTS.SEZ><br>0: OFF.<br>1: ON. | [RSTS] (READ)<br>23 16<br>- - - - - - - n |
| Input count of EZ signal to use for origin return <RSPD.EZC><br>Count down from a value in RENV3.EZD bit.<br>Next to count 0, return to a value in RENV3.EZD bit. | [RSTS] (READ)<br>23 16<br>n n n n - - - |

## *6.7 Interpolation operation mode*

Using multiple G9103Cs, G9103C has linear interpolation circuit for more than 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”. G9103C is an LSI to control single axe. Therefore, in interpolation, it outputs pulse trains of either “Interpolated X axis” or “Interpolated Y axis”. Select which axis signal is output by operation mode (RMD.MOD.)

By setting the same to “multiple interpolated Y axes” of multiple G9103Cs, multiple motors can also be controlled by one “interpolated axis”.




Selector 1: Select control method between control by internal pulse and synchronizing with PA and PB.

Selector 2: Select the following five outputs.

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: Start timing between interpolation operation blocks is not synchronized. If you continue interpolation operation blocks, do not use functions that difference operation time of each G9103C occurs, such as backlash correction, vibration restriction control, direction change timer, delay by INP signal input.

Note 2: In interpolation between 2 axes, the function to make synthesized speed constant is enabled (RMD.MIPF = 1), can be used. In interpolation among more than 3 axes, calculate a speed value so as to make synchronized speed constant using software.

Note 3: When the interpolation operation is started by broadcast communication, a slight error of start timing occurs by the time difference that occurs by the broadcast communication frame (electric signals) transferred in a cable. Because of the difference, approximately 50 ns per 10 meters occurs, in the case of 100 meters, the time difference becomes approximately 500 n seconds.
Approximately 1 μs delay occurs each time passing through the **Motionnet**® hub.

Note 4: Make sure to use the synchronizing function of clock for motor control in the system for interpolation operation.

Note 5: Center LSI has two SIA and SIB terminals as serial input terminals and the communication line can be separated into two systems. However, G9103Cs of the interpolated axes should not be separate into two systems.

Note 6: Each G9103C's circuit to control speed does not synchronize among multiple 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 signal, etc.

There are 16 interpolation operation modes as follows.

| RMD.MOD | Description | Interpolated output axis |
|---|---|---|
| 60h | Continuous linear interpolation | X axis |
| 61h | Linear interpolation | X axis |
| 64h | Circular interpolation in CW direction | X axis |
| 65h | Circular interpolation in CCW direction | X axis |
| 68h | Continuous linear interpolation synchronized with PA and PB | X axis |
| 69h | Linear interpolation synchronized with PA/PB | X axis |
| 6Ch | Circular interpolation in CCW direction synchronized with PA and PB | X axis |
| 6Dh | Circular interpolation in CCW direction synchronized with PA and PB | X axis |
| 70h | Continuous linear interpolation | Y axis |
| 71h | Linear interpolation | Y axis |
| 74h | Circular interpolation in CW direction | Y axis |
| 75h | Circular interpolation in CCW direction | Y axis |
| 78h | Continuous linear interpolation synchronized with PA and PB | Y axis |
| 79h | Linear interpolation synchronized with PA and PB | Y axis |
| 7Ch | Circular interpolation in CCW direction synchronized with PA and PB | Y axis |
| 7Dh | Circular interpolation in CCW direction synchronized with PA and PB | Y axis |

Set a same value in the registers (RFL, RFH, RUR, RDR, RMG, RDP, RUS, and RDS) that are related to the speed of axes interpolated by multiple G9103Cs.

### 6.7.1 Continuous linear interpolation operation

#### 6.7.1.1 Continuous linear interpolation operation between 2 axes

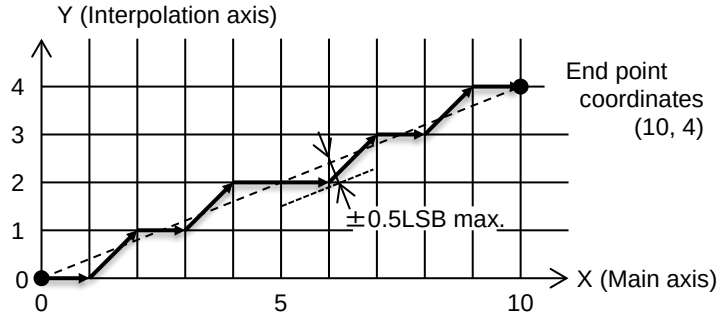
Set a speed ratio in RMV register and RMVY register. Operation is made at the speed corresponding to a ratio between the RMV and RMVY register value. Set operation mode to continuous linear interpolation of interpolated X axis (RMD.MOD = 60h) and continuous linear interpolation of interpolated Y axis (RMD.MOD = 70D) and set holding start to enabled (RMD.MSY = 1). Set a same value to the speed setting registers (RFL, RFH, RUR, RMG, RDP, RUS, and RDS).
Write the same command (STAFL, STAFH, STAD, and STAUD) and starts simultaneously by a command such as broadcast communication start command (2x01h) etc. Like continuous operation mode (RMD.MOD = 01h), pulse output continues until stop command is written.

#### 6.7.1.2 Continuous linear interpolation operation (2 ~ 64 axes) among multiple axes

Set a ratio between own axis's speed and the fastest axis in RMV register and RMVY register. Operation is made at the speed corresponding to a ratio between the RMV and RMVY register value. Set an operation mode to continuous linear interpolation of interpolated X axis (RMD.MOD = 60h) and set holding start to enabled (RMD.MSY = 1). Set a same value to the speed setting registers (RFL, RFH, RUR, RMG, RDP, RUS, and RDS).
Write same commands (STAFL, STAFH, STAD, and STAUD) and starts simultaneously by a command such as broadcast communication start command (2x01h) etc. Like continuous operation mode (RMD.MOD = 01h), pulse output continues until stop command is written.

### 6.7.2 Linear interpolation operation

[Linear interpolation precision]
In linear interpolation, a straight line is drawn from a current coordinate to end point coordinate as shown in the right figure. The right figure shows an example of a straight line to the end point coordinate (10, 4).
Position precision to the specified straight line in linear interpolation is within ±0.5 LSB in all interpolated area. LSB is the smallest feed unit of RMV register value and shown by one cell in the right figure. It corresponds to mechanical resolution.




#### 6.7.2.1 Linear interpolation between 2 axes

Set a feed amount in PRMV and PRMVY registers. Operation is made at the speed corresponding to a ratio between the RMV and RMVY register value. Set operation mode to linear interpolation of interpolated X axis (RMD.MOD = 61h) and linear interpolation of interpolated Y axis (RMD.MOD = 71h) and set holding start to enabled (RMD.MSY=1). Set the same value to the speed setting registers (RFL, RFH, RUR, RMG, RDP, RUS, and RDS).
Write a same command (STAFL, STAFH, STAD, and STAUD) and starts simultaneously by a command such as broadcast communication start command (2x01h) etc.

The below table shows an example to set X axis feed amount 1000 and Y axis feed amount 2000 in linear interpolation between 2 axes.

| Pre-register for WRITE | Interpolated x axis | Interpolated Y axis |
|---|---|---|
| PRMD | 00004461h | 00004471h |
| PRMV | 1000 | 1000 |
| PRMVY | 2000 | 2000 |
| PRFL | 1 | 1 |
| PRFH | 100000 | 10000 |
| PRUR | 20 | 20 |
| PRDR | 0 | 0 |
| PRMG | 199 | 199 |
| PRDP | 0 | 0 |
| PRUS | 0 | 0 |
| PRDS | 0 | 0 |

6.7.2.2 Linear interpolation among multiple axes (2 ~ 64 axes)

Set a feed amount in PRMV and PRMVY registers. Operation is made at the speed corresponding to a ratio between the RMV and RMVY register value. Set operation mode to linear interpolation of interpolated X axis (RMD.MOD = 61h) and set holding start to enabled (RMD.MSY = 1). Set the same value to the speed setting registers (RFL, RFH, RUR, RMG, RDP, RUS, and RDS).
Write a same command (STAFL, STAFH, STAD, and STAUD) and starts simultaneously by a command such as broadcast communication start command (2x01h) etc.

The below table shows an example to set 1st axis feed amount 1000, 2nd axis feed amount 2000, 2rd axis feed amount 3000 and 4th feed amount 4000 (longest) in linear interpolation amount 4 axes.

| Pre-register for WRITE | Interpolated 1st axis | Interpolated 2nd axis | Interpolated 3rdt axis | Interpolated 4th axis |
|---|---|---|---|---|
| PRMD | 00004461h | 00004461h | 00004461h | 00004461h |
| 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 |

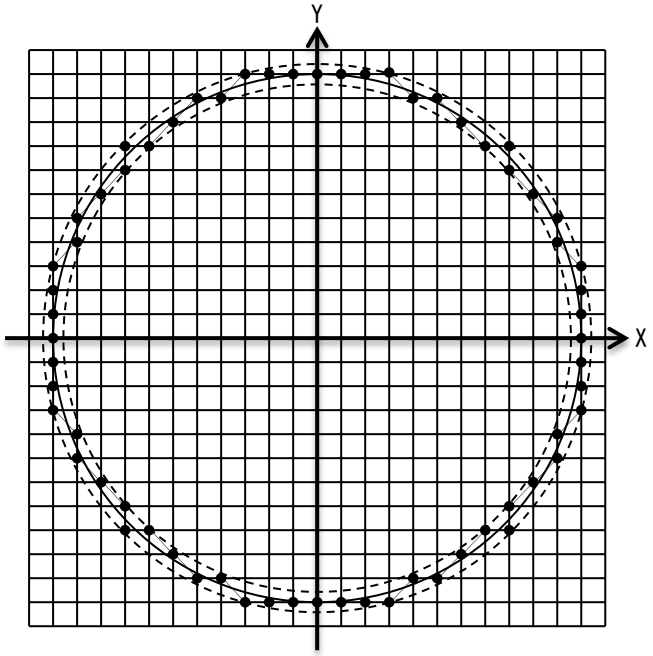
6.7.3 Circular interpolation operation

Use a start point as a base position (0,0) and set the incremental position of an ending point in RMV and RMVY registers.
In the case of a perfect circle, the ending point of a perfect circle is the same as the base point (0, 0). When 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 operation mode of CW direction to CW direction circular interpolation of interpolated X axis (RMD.MOD=64h) and CW direction circular interpolation of interpolated Y axis (RMD.MOD=74h).
Set operation mode of CCW direction to CCW direction circular interpolation of interpolated X axis (RMD.MOD=65h) and CCW direction circular interpolation of interpolated Y axis (RMD.MOD=75h).
Set holding start to enabled (RMD.MSY=1). Set a same value to the speed setting registers (RFL, RFH, RUR, RMG, RDP, RUS, and RDS).
Write a same command (STAFL, STAFH, STAD, and STAUD) and starts simultaneously by a command such as broadcast communication start command (2x01h) etc.

The followings are an example of circular interpolation of 180 degrees in CW direction with end point (200, 0) and the circular center (100, 0). The method for calculation of RCI setting value is described later.

| Pre-register for WRITE | Interpolated X axis | Interpolated Y axis |
|---|---|---|
| PRMD | 00004464h | 00004474h |
| 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 |

[Circular interpolation precision]
In circular interpolation, an arc is drawn from a current coordinate to end point coordinate as shown in the right figure. The right figure shows an example of a perfect circle with radius 11 cm.
Position precision to the specified arc in circular interpolation is within ±0.5LSB in all interpolated area. LSB is the smallest feed unit of RMV register value and shown by one cell in the right figure. It corresponds to mechanical resolution.



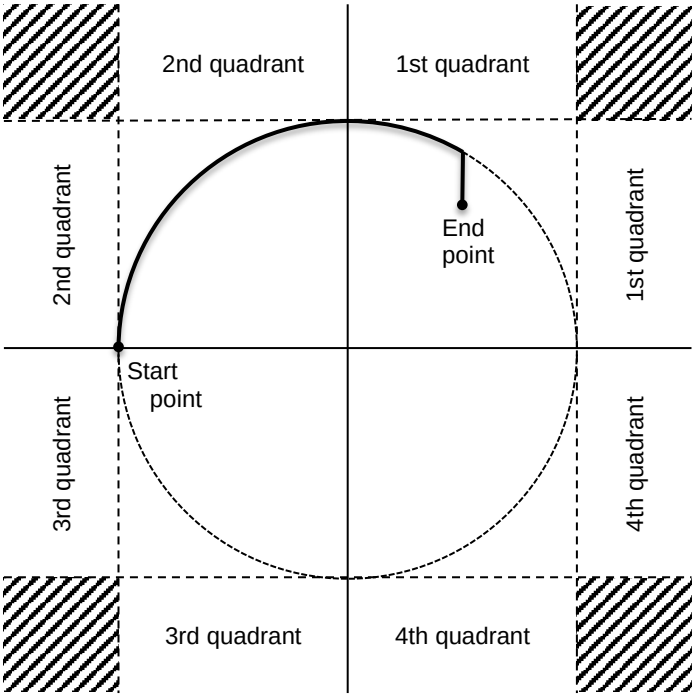

Black dots: Interpolation trajectory
Solid line: A circle of radius 11
Dashed line: A circle of radius 11 ± 0.5

6.7.3.1 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 other axis moves to the end point. The speed of ending point lead-in operation is the same as the speed of circular interpolation.

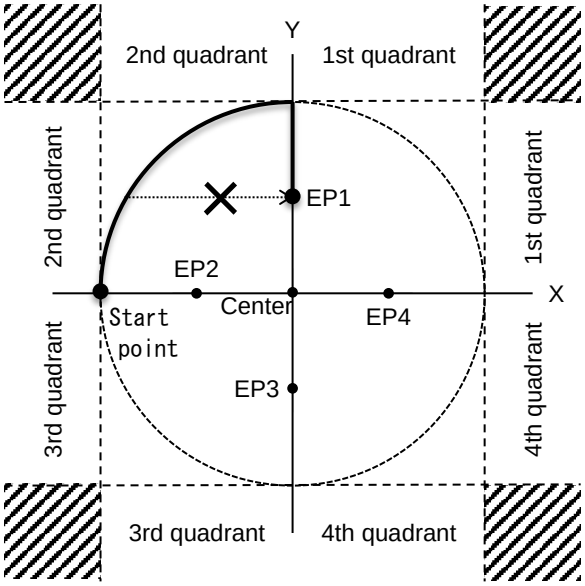
Avoid that an end point coordinate of circular interpolation is set in the shaded areas in the right figure because a motor does not stop and continue circular interpolation eternally.



6.7.3.1.1 Ending point lead-in operation on coordinate axis

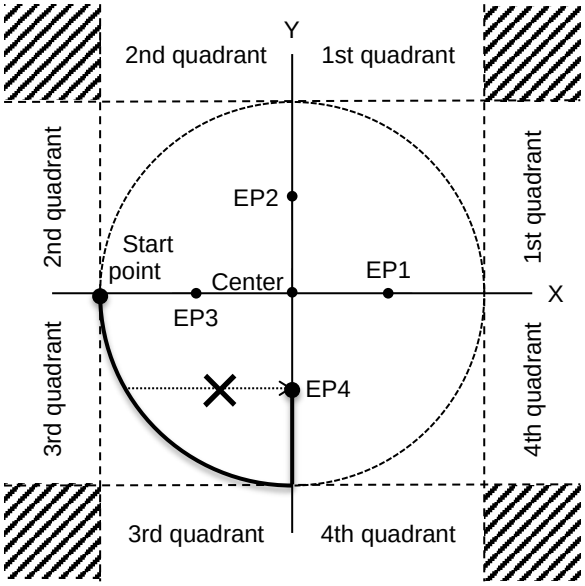
In the case that an ending point is on coordinate axis, lead-in operation is executed regarding the quadrant that is ahead of the coordinate axis on which the end point exists as the end point quadrant.

For example, in CW direction, if end point is “EP1” on coordinate Y in the right figure “Circular interpolation that end point is on the coordinate axis in CW direction”, end point quadrant is not 2nd quadrant that Y coordinate becomes the same in first, but 1st quadrant that is exist ahead of coordinate Y.
Similarly, if end point is “EP4”, end point quadrant is 4th quadrant. If end point is “EP3”, end point quadrant is 3rd quadrant. If end point is “EP2”, end point quadrant is 2nd quadrant.
Therefore, arc is drawn right above coordinate axis that the end point exists in CW direction and end point lead-in operation is executed along the coordinate axis.




Circular interpolation that end point is on the coordinate axis in CW direction

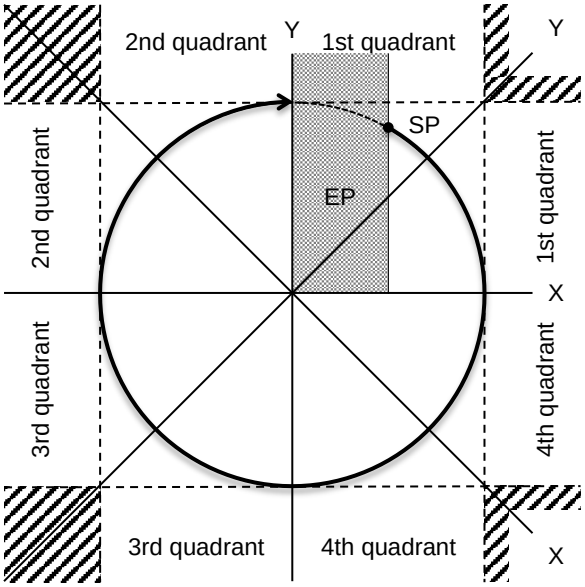
In CCW direction, if end point is “EP4” on coordinate Y in the right figure “circular interpolation that end point is on the coordinate axis in CCW direction”, end point quadrant is not 3rd quadrant that coordinate Y becomes the same in first, but 4th quadrant that is exist ahead of coordinate Y.
Similarly, if end point is “EP1”, end point quadrant is 1st quadrant. If end point is “EP2”, end point quadrant is 2nd quadrant. If endpoint is “EP3”, end point quadrant is 3rd quadrant. Therefore, arc is drawn right above coordinate axis that the end point exists in CCW direction and end point lead-in operation is executed along the coordinate axis.

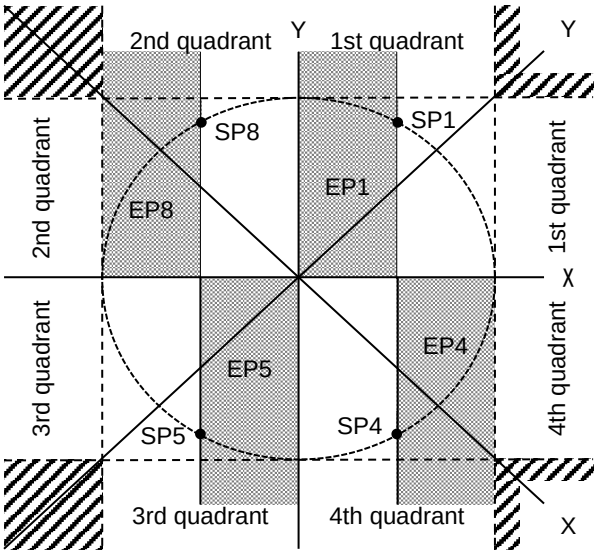


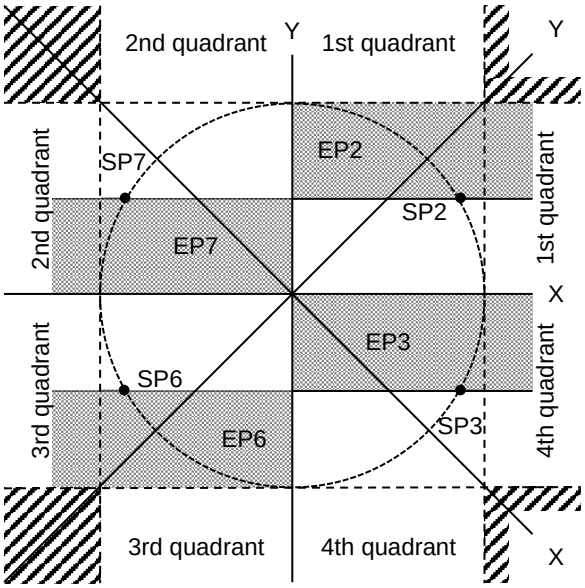

Circular interpolation that end point is on the coordinate axis in CCW direction

6.7.3.1.2 Ending point lead-in operation in the same quadrant

When end point exists in the same quadrant as start point, the timing to determine an end point quadrant varies by the position relation between the start point and the end point. For example, in the right figure of circular interpolation in CW direction, if a start point "SP" exists the area between Y axis of 1st quadrant and XY axis and end point exists in the shaded "EP" area, end point quadrant not reached yet. As the result, an arc is drawn passing through 4th quadrant and to 1st quadrant and then, end point lead-in operation is executed.
If end point exists out of the shaded "EP" area in 1st quadrant, end point quadrant has reached. As the result, a short arc is drawn in 1st quadrant or end point lead-in operation is executed without drawing any arc.




| Quadrant | Start point range | End point range |
|---|---|---|
| 1 | \|SP1.X\| ≤ \|SP1.Y\| | EP1.X < SP1.X |
| 4 | \|SP4.X\| ≤ \|SP4.Y\| | EP4.X ≥ SP4.X |
| 3 | \|SP5.X\| ≤ \|SP5.Y\| | EP5.X ≥ SP5.X |
| 2 | \|SP8.X\| ≤ \|SP8.Y\| | EP8.X < SP8.X |

| Quadrant | Start point range | End point range |
|---|---|---|
| 1 | \|SP2.X\| > \|SP2.Y\| | EP2.Y ≥ SP2.Y |
| 4 | \|SP3.X\| > \|SP3.Y\| | EP3.Y ≤ SP3.Y |
| 3 | \|SP6.X\| > \|SP6.Y\| | EP6.Y < SP6.Y |
| 2 | \|SP7.X\| > \|SP7.Y\| | EP7.Y < SP7.Y |

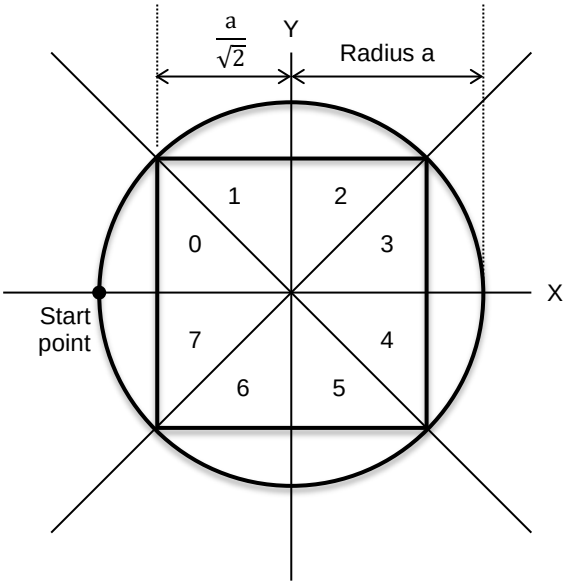
"|SP1.X|" in the table is an absolute value of coordinate X that a start point "SP1" exists. Absolute coordinate of start point is "0.0", however, the start point "SP" in the table is an incremental coordinate.
Coordinate X of the center = 100, "SPL.X = 100".
Coordinate Y of the center = 200, "SPL.Y = 200".

Note. Even if a start point and end point exist at a similar position, the result varies according to a start point and rotation direction. For example, if in CW direction, start point is |SP1.X| ≤ |SP1.Y| and end point is "EP1.X = SP1.X", end point lead-in operation is executed without drawing an arc. If start point is |SP4.X| ≤ |SP4.Y| and end point is "EP4.X=SP4.X", end point lead-in operation is executed with drawing an arc. Check the quadrant of arc and end point and their position relation. However, if a start point and end point exist and the same coordinate, perfect circle is drawing regardless of the above.

6.7.3.2 Number of stepping in circular interpolation

If you use circular interpolation with acceleration / deceleration, remaining amount start command (CNTFL, CNTFH, CNTD, and CNTUD) of counter for remaining pulses, it is needed to set the number of pulses (the number of stepping for circular interpolation) necessary for circular interpolation in RCI register of an axis controlled.
In calculating a number of pulses necessary for circular interpolation, assume that the plane containing X-axis and Y-axis is divided into 8 areas regarding the circular coordinate of the circular as the center.
The status of command pulses output by each axis in each area is as follows.

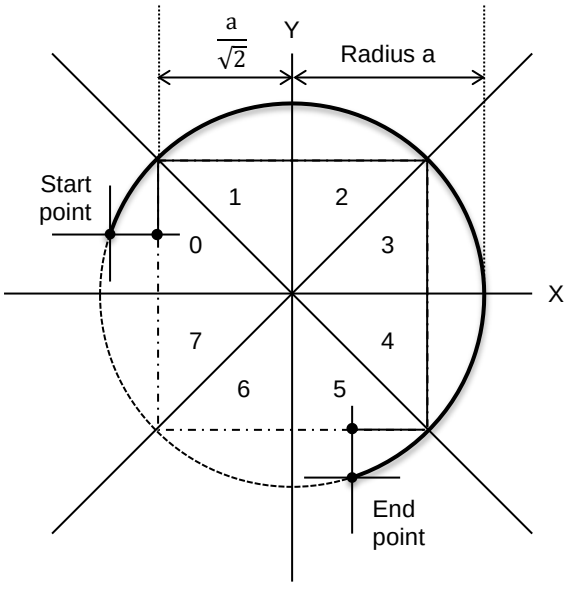
| Area | Interpolated X axis output pulse | Interpolated Y axis output pulse |
|---|---|---|
| 0 | Output based on a result of calculation for interpolation | Always output |
| 1 | Always output | Output based on a result of calculation for interpolation |
| 2 | Always output | Output based on a result 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, you can see that either axis outputs pulses in any areas.
Therefore, the number of stepping in 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 / √2) x 2. Set this value in PRCI register.

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

1. Discriminate that the start point is in which area (area 0 ~ 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 ~ 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 RCI register.



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

Note 1: The RCI register value is used for the occurrence of timing to start deceleration and counter for remaining pulse used for remaining start command. If a smaller value than calculated value is entered, the deceleration starts early and the time to feed at FL constant speed appear even after the speed reached FL. If a bigger value than calculated value is entered, the deceleration starts late and the movement on 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 a ramping-down point manually, you can use RDP register formula in positioning operation regarding RCI setting value as a feeding amount.
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 register in experiment.

Note 3: Make sure to set 1 or more than 1 to RCI register value in circular interpolation with acceleration / deceleration.

### 6.7.4 Continuous operation of interpolation operation

While interpolation operation, the next data can be set. Therefore, interpolation operation can be executed continuously.
However, note that it is necessary to set a 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 event interrupt cause is enabled (RIRQ.IRBE = 1), the center LSI can outputs an interrupt request signal when operation is stopping because sending the data for next operation is not in time.

When using start command for next operation (NSTAFL, NSTAFH, NSTAD, and NSTAUD), a motor stops by error if sending data for next operation is not in time.
In the case with using continuous circular interpolation in addition to linear interpolation in continuous interpolation among more than 3 axes, set circular interpolation dummy operation to enabled (RMD.MDMY = 1) in order to match operation time with axes that are not used for circular interpolation.

Example of continuous operation of interpolation

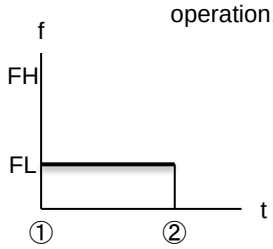
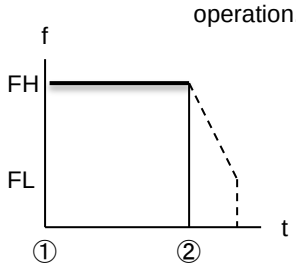
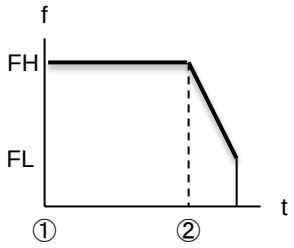
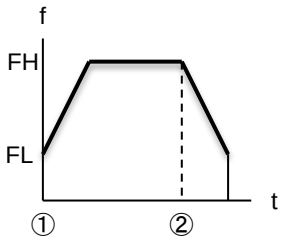
| Ope-ration | Pre-register for WRITE | Interpolated axis X | Interpolated axis Y | Interpolated axis Z | Notes |
|---|---|---|---|---|---|
| 1 | PRMV | 1000 | 0 | 1500 | Linear interpolation between 2 axes with X-axis and Z-axis.<br>End point is X = 1000, Z = 1500.<br>Set feed amount 0 (PRMV = 0) because axis Y does not operate.<br>Initially make operation holding start and start by a broadcast communication command, etc. |
| | PRMVY | 1500 | 1500 | 1500 | |
| | PRIP | Disabled | Disabled | Disabled | |
| | PRIPY | Disabled | Disabled | Disabled | |
| | PRMD | 00004061h | 00004061h | 00004061h | |
| 2 | PRMV | 3000 | 100 | 2000 | Linear interpolation among 3 axes (X, Y and Z).<br>End point: X = 3000, Y = 100, Z = 2000. |
| | PRMVY | 3000 | 3000 | 3000 | |
| | PRIP | Disabled | Disabled | Disabled | |
| | PRIPY | Disabled | Disabled | Disabled | |
| | PRMD | 00000061h | 00000061h | 00000061h | |
| 3 | PRMV | -2000 | -2000 | -2000 | Circular interpolation in CW direction with axis X and axis Z.<br>Center position: X = 0, Z = 2000.<br>End point position: X = −2000, Z = 2000.<br>Set dummy operation to enabled (PRMD.MDMY = 1) on axis Y in order to match operation time with axes that are not used for circular interpolation. |
| | PRMVY | 2000 | 2000 | 2000 | |
| | PRIP | 0 | 0 | 0 | |
| | PRIPY | 2000 | 2000 | 2000 | |
| | PRMD | 00000064h | 40000064h | 00000074h | |
| 4 | PRMV | 0 | 0 | 0 | Circular interpolation in CCW direction with X-axis and Y-axis.<br>Center position: X = 100, Y = 2000<br>End point position: X = 0, Y = 0 (perfect circle)<br>Set dummy operation to enabled (PRMD.MDMY = 1) on axis Z in order to match operation time with axes that are not used for circular interpolation. |
| | PRMVY | 0 | 0 | 0 | |
| | PRIP | 100 | 100 | 100 | |
| | PRIPY | 2000 | 2000 | 2000 | |
| | PRMD | 00000065h | 00000075h | 40000065h | |
| 5 | PRMV | 0 | 0 | 1000 | Only axis Z moves + 1000.<br>Set feed amount 0 (PRMV = 0) on axis X and axis Y in linear interpolation operation in order to match operation time with axes that are not used. |
| | PRMVY | 1000 | 1000 | 1000 | |
| | PRIP | Disabled | Disabled | Disabled | |
| | PRIPY | Disabled | Disabled | Disabled | |
| | PRMD | 00000061h | 00000061h | 00000061h | |

Procedure

1. Set a data for operation 1 data and a start command to each interpolated axis. Pulses are not output because holding start is set in PRMD register.
2. Send a broadcast communication command (2001h). Operation 1 starts.
3. Write data for operation 2 and a start command dedicated for next operation to each interpolated axis. Pre-register become full (MSTS.SPRF = 1).
4. After operation 1 is completed, pre-register become empty (MSTS.SPRF = 0).
   Write data for operation 3 and a start command dedicated for next operation to each interpolated axis. Pre-register become full (MSTS.SPRF = 1).
5. After operation 2 is completed, pre-register become empty (MSTS.SPRF = 0).
   Write data for operation 4 and a start command dedicated for next operation to each interpolated axis. Pre-register become full (MSTS.SPRF = 1).
6. After operation 3 is completed, pre-register become empty (MSTS.SPRF = 0).
   Write data for operation 5 and a start command dedicated for next operation to each interpolated axis. Pre-register become full (MSTS.SPRF = 1).
7. After operation 4 is completed, pre-register become empty (MSTS.SPRF = 0).
8. Operation 5 is completed.

## 7. Speed patterns

### 7.1 Speed patterns

| Speed pattern | Continuous mode | Positioning operation mode |
|---|---|---|
| FL constant speed operation.<br>  | 1). The point that operation starts by writing STAFL (0050h) command.<br>2). The point that operation stops by writing STOP (0049h) or SDSTP (004Ah) command. | 1) The point that operation starts by writing STAFL (0050h) command.<br>2) The point that operation stops by RPLS=0, writing STOP (0049h) or SDSTP (004Ah) command. |
| FH constant speed operation.<br>  | 1). The point that operation starts by writing STAFH (0051h) command.<br>2). The point that operation stops by writing STOP (0049h) command.<br>* When SDSPT (004Ah) command is written, operation decelerates and stops. | 1) The point that operation starts by writing STAFH (0051h) command.<br>2) The point that operation stops by RPLS=0 or writing STOP (0049h).<br>* When SDSPT (004Ah) command is written, operation decelerates and stops. |
| High speed operation 1.<br>  | 1). The point that operation starts by writing STAD (0052h) command.<br>2). The point that operation stops by writing SDSTP (004Ah) command.<br>* When STOP (0049h) command is written, operation stops immediately. | 1). The point that operation starts by writing STAD (0052h) command.<br>2). The point that operation starts to decelerate and stop by writing SDSTP (004Ah) command.<br>* The automatic ramping-down point setting cannot be used. Please select manual setting (RMD.MSDP=1) and set an appropriate value to RDP (ramping-down point).<br>* When STOP (0049h) command is written, operation stops immediately. |
| High speed operation 2.<br>  | 1). The point that operation starts by writing STAUD (0053h) command.<br>2). The point that operation stops by writing SDSTP (004Ah) command.<br>* When STOP (0049h) command is written, operation stops immediately. | 1). The point that operation starts by writing STAUD (0053) command.<br>2). The point that operation starts to decelerate and stop by writing SDSTP (004Ah) command.<br>* The automatic ramping-down point setting can be used. In the case that RDP (ramping-down point) register is set to 0 with auto manual setting (RMD.MSDP = 1), operation stops immediately without automatic setting (RMD.MSDP = 0).<br>* When STOP (0049h) command is written, operation stops immediately. |

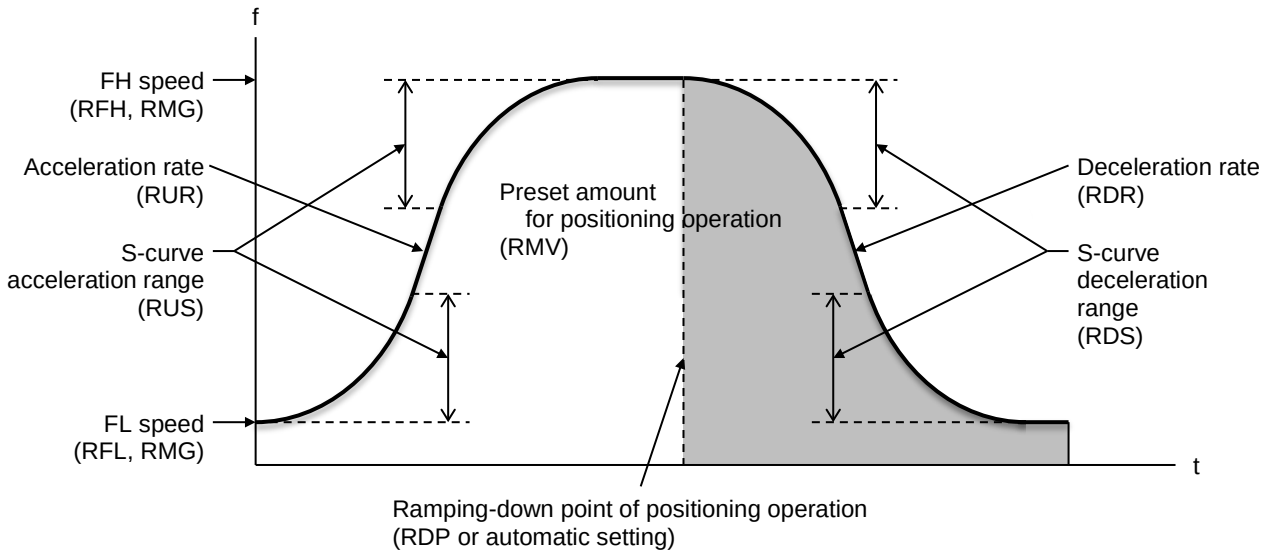
## *7.2 Speed pattern settings*

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

| Register | Description | Bit length | Setting range |
|---|---|---|---|
| RMV | Positioning amount | 28 | −134,217,728 (8000000h) ~ 134,217,727 ( 7FFFFFFh) |
| RFL | FL speed | 17 | 1 ~ 100,000 ( 186A0h) Note 2 |
| RFH | FH speed | 17 | 1 ~ 100,000 ( 186A0h) Note 2 |
| RUR | Acceleration rate | 16 | 1 ~ 65,535 ( FFFFh) |
| RDR | Deceleration rate Note 1 | 16 | 0 ~ 65,535 ( FFFFh) |
| RMG | Speed magnification rate | 11 | 2 ~ 2,047 ( 7FFh) |
| RDP | Ramping-down point | 24 | 0 ~ 16,777,215 ( FFFFFFh) |
| RUS | S-curve acceleration range | 16 | 0 ~ 50,000 ( C350h) Note 3 |
| RDS | S-curve deceleration range | 16 | 0 ~ 50,000 ( C350h) Note 3 |
| RFA | FA speed | 17 | 0 ~ 100,000 ( 186A0h) Note 2 |

Note 1: If RDR = 0, a value set in RUR is used as a deceleration rate.
Note 2: All values from 186A0h ~ 1FFFFh will be replaced with 186A0h.
Note 3: All values from 0C350h ~ 0FFFFh will be replaced with 0C350h.

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




### 7.2.1 RMV (PRMV): Positioning feed amount setting register (28 bits)

This register is used to set a target position for positioning operations.
The meaning of setting may vary according to an operation mode selected.
Setting range: −134,217,728 ~ +134,217,727
By changing RMV register value during operation, you can override position.

### 7.2.2 RFL (PRFL): FL speed setting register (17 bits)

This register is used to set an initial speed (and stopping speed)
By changing RFL register value during operation, you can override speed.
Specify a speed for FL constant speed operations and high speed operations (acceleration/deceleration operations) in the range of 1 ~ 100,000 (186A0h). All values from 100,000 ~ 131,071 (186A0h ~ 1FFFFh) will be replaced with 100,000.
The actual operation speed will be a product of RFL register value and speed magnification (RMG register calculated value).

FL speed [pps] = RFL x Speed magnification

### 7.2.3 RFH (PRFH): FH speed setting register (17 bits)

This register is used to set the initial speed (and stopping speed)
By changing RFH register value during operation, you can override speed.
Specify the speed for FH constant speed operations and the start speed for high speed operations (acceleration/deceleration operations) in the range of 1 ~ 100,000 (186A0h). All values from 100,000 ~ 131,071 (186A0h ~ 1FFFFh) will be replaced with 100,000.
The actual operation speed will be the product of RFH register value and speed magnification (RMG register calculated value).

FH speed [pps] = RFH x Speed magnification

### 7.2.4 RUR (PRUR): Acceleration rate setting register (16 bits)

This register is used to set an acceleration rate.
Specify an acceleration characteristic for high speed operations (acceleration/deceleration operations), in the range of 1 ~ 65,535 (0FFFFh)
Relationship between a value entered and an acceleration time will be as follows:

1) Linear acceleration (RMD.MSMD = 0)

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

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

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

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

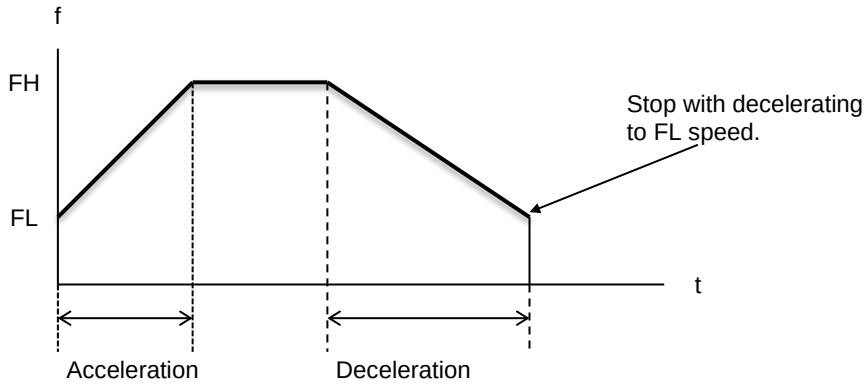
$$\text{Acceleration time [s]} = \frac{(RFH - RFL + 2 \times RUS) \times (RUR + 1) \times 8}{40{,}000{,}000}$$

### 7.2.5 RDR (PRDR): Deceleration rate setting register (16 bits)

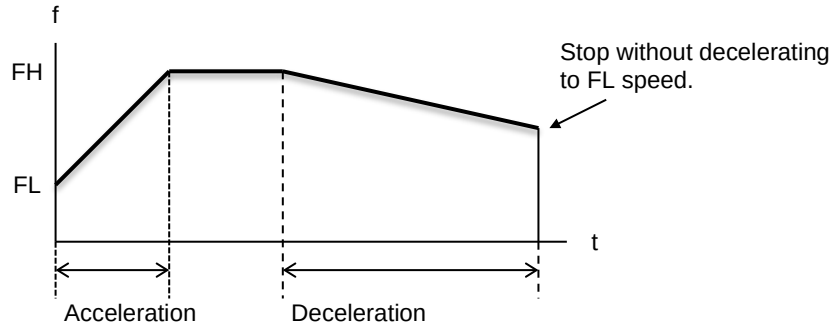
This register is used to set a deceleration rate.
Specify deceleration characteristics for high speed operations (acceleration/deceleration operations) in the range of 0 ~ 65,535 (FFFFh).
Even if auto ramping-down point setting is enabled (RMD.MSDP = 0), a value placed in RDR register will be used as a deceleration rate.
However, when RDR = 0, a value placed in RUR is used as a deceleration rate

When you want to set auto ramping-down point setting (RMD.MSDP=0), adjust it so that (deceleration time) ≦ (acceleration time x 2).
If the (deceleration time) > (acceleration time x 2) is set, a motor may not be able to decelerate to FL speed when stopping. In this case, select manual ramping-down point setting (RMD.MSDP = 1).

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




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




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

1) Linear deceleration (RMD.MSMD = 0)

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

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

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

### 7.2.6 RMG (PRMG): Speed magnification register (11 bits)

This register is used to set the speed magnification.
Specify a relationship between RFL, RFH and FA settings and the speed, in the range of 2 ~2,047 (07FFh). As the magnification becomes higher, the speed setting units tend to be coarser. Normally set magnification as low as possible.
Operation speed [pps] will be a product of speed magnification rate and speed register value.
The relationship between a register value to set of speed magnification and speed magnification is as follows.

$$\text{Speed magnification [times]} = \frac{40{,}000{,}000}{(\text{RMG} + 1) \times 200{,}000}$$

[Setting example of speed magnification]

| Setting | Magnification rate | Output speed range [pps] | Setting | Magnification rate | Output speed range[pps] |
|---|---|---|---|---|---|
| 1999 (7CFh) | 0.1 | 0.1 ~ 10,000.0 | 39 (27h) | 5 | 5 ~ 500,000 |
| 999 (3E7h) | 0.2 | 0.2 ~ 20,000.0 | 19 (13h) | 10 | 10 ~ 1,000,000 |
| 399 (18Fh) | 0.5 | 0.5 ~ 50,000.0 | 9 (09h) | 20 | 20 ~ 2,000,000 |
| 199 (0C7h) | 1 | 1 ~ 100,000 | 3 (3h) | 50 | 50 ~ 5,000,000 |
| 99 (63h) | 2 | 2 ~ 200,000 | 2 (2h) | 66.6 | 66.6…~ 6,666,666.6… |

### 7.2.7 RDP (PRDP): Ramping-down point setting register (24 bits)

This register is used to set a ramping-down point (deceleration starting point).
Specify a value used to determine a deceleration starting point for positioning operations in high speed operation (with acceleration and deceleration).
According to "ramping-down point setting method" (RMD.MSDP), the meaning of a value varies.

<When set to manual setting (RMD.MSDP = 1)>
Set a number of pulses at which to start deceleration in the range of 0 ~ 16,777,215 (0FFFFFFh).
The optimum value for a ramping-down point can be calculated as shown 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 pattern start, without changing the value in RFH register while the auto FH correction function turning OFF (RMD.MADJ = 1) is calculated as shown in the equation below.

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

Note: When using idling control, modify the value for RMV in the equation above by deducting the number of idling pulses from the value placed in RMV register. The number of idling pulses will be "1 ~ 6" pulses when RENV2.IDL = 0 ~ 7).

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

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

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

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

If RMD.MSDC=0, ramping-down point is automatically set by formula. If the function to make synthesized speed constant is enabled (RMD.MIPF = 1) in interpolation operation, count method is selected. In addition to the case that acceleration time is equal to deceleration time, ramping-down point is automatically set even in the case that acceleration time and deceleration time are different. The error from calculated value may occur at 0.01% frequency and the error may exceed one pulse. Please note that when you use a speed pattern that decelerating pulses number become big.
If RMD.MSDC=1, ramping-down point is automatically specified by count method. This is available only in the case that acceleration time is equal to deceleration time. If they are different, auto setting cannot be disabled. The error from calculated value is within one pulse.

### 7.2.8 RUS (PRUS): S-curve acceleration range register (16 bits)

This register is used to specify S-curve range in S-curve acceleration.
Specify S-curve acceleration range for S-curve acceleration/deceleration operations in the range of 1 ~ 50,000 (0C350h).
All values from 50,000 ~ 65,535 (0C350h ~ 0FFFFh) will all be replaced with 50,000.
The S-curve acceleration range $S_{SU}$ will be a product of RUS register value and speed magnification (RMG register calculated value)

$S_{SU}$ [pps] = RUS x Speed magnification

Speeds between FL speed and (FL speed + $S_{SU}$), and between (FH speed - $S_{SU}$) and FH speed, will be S-curve acceleration operations. Intermediate speeds will use linear acceleration.
However, if zero is set, "(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 set, the G9103C will operate with nearly linear acceleration.
If a larger value than "(RFH − RFL) / 2" is set, the speed will not reach the maximum acceleration speed and acceleration time will be different from a calculated value. Therefore, enter a value smaller than "(RFH − RFL) / 2" to the RUS register.

### 7.2.9 RDS (PRDS): S-curve deceleration range setting register (16 bits)

This register is used to specify S-curve range in S-curve deceleration
Specify the S-curve deceleration range for S-curve acceleration/deceleration operations in the range of 1 ~ 50,000 (0C350h).
All values from 50,000 ~ 65,535 (0C350h ~ 0FFFFh) will all be replaced with 50,000.
The S-curve acceleration range $S_{SD}$ will be a product of RDS register value and speed magnification rate (RMG register calculated value).

$S_{SD}$ [pps] = RDS x Speed magnification

Speeds between 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 set, "(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 G9103C operates with nearly linear acceleration.
If a larger value than "(RFH − RFL) / 2" is specified, a motor will not reach the maximum acceleration speed and deceleration time will be different from a calculated value. Therefore, enter a value smaller than "(RFH − RFL) / 2" to the RDS register.

### 7.2.10 RFA: FA speed setting register (17 bits)

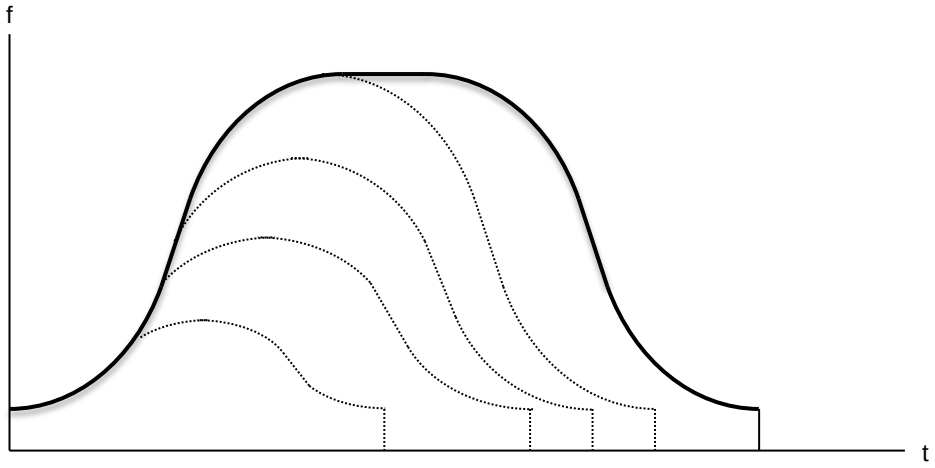
This register is used to set speed during backlash correction.
Set the correction speed feed amount for use during backlash within the range of 1 ~ 100,000 (186A0h).
All values from 100,000 ~ 131,071 (186A0h ~ 1FFFFh) will all be replaced with 100,000.
The actual operating speed will be a product of RFA register value and speed magnification the rate (RMG register calculated value).
This register value is also used for a reverse speed during origin return operation

FA speed [pps] = RFA x Speed magnification

## *7.3 FH correction*

When FH correction function is enabled (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 pattern driving.
However, if values in RUR and RDR registers are set so that the (deceleration time) > (acceleration time x 2), do not use FH correction function.
In order to eliminate triangle driving without using the FH correction function (RMD.MADJ = 1), use manual FH correction. If you use this setting to avoid triangle driving, FH speed has to be changed in advance.

[FH correction]




Automatic correction of the maximum speed for changing feed amount

Note: When using idling control, modify a value for RMV in the equation above by deducting a number of idling pulses from the value placed in RMV register. The number of idling pulses will be "1 ~ 6" pulses when RENV2.IDL = 0 ~ 7).

< 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 = 0, RDS = 0)
When

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

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

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

3) S-curve acceleration/deceleration with linear acceleration/deceleration (RMD.MSMD=1 and RUS > 0, RDS > 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 |
| 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

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

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

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

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

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

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

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

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

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

(iii) Eliminate linear acceleration/deceleration range
When

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

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 |
| 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

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

and

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

Then,

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

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 linear acceleration section and set up a small linear deceleration range.
When

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

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

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

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

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

(iii) Eliminate 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 | |

## *7.4 Example of setting up an acceleration/deceleration speed pattern*

Ex.: When the initial speed =10 pps, the operation speed = 110 kpps, and the acceleration/deceleration time = 300 ms,

1) Select a value so that magnification is 2x mode in order to output 110 kpps.
RMG = 99 (63h)

2) Since 2x mode is selected, set 1/2 of operation speed (110kpps) to the register.
RFH = 55000 (D6D8h)

3) Since 2x mode is set, set 1/2 of the initial speed (10pps) to the register.
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 register value.

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

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

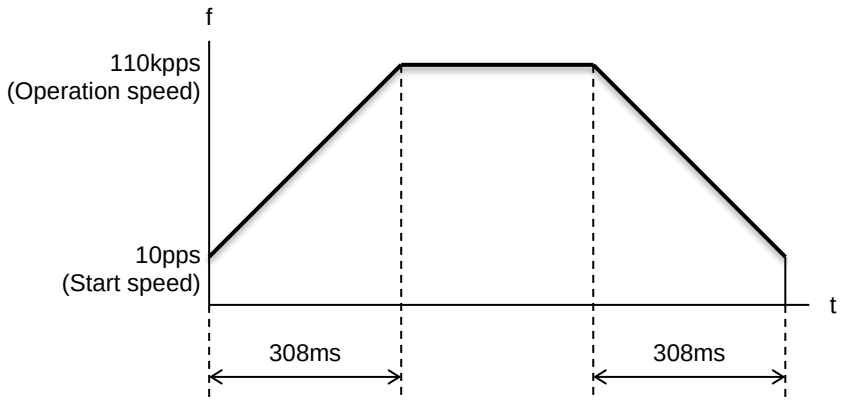
Then,

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

RUR = 26.275

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

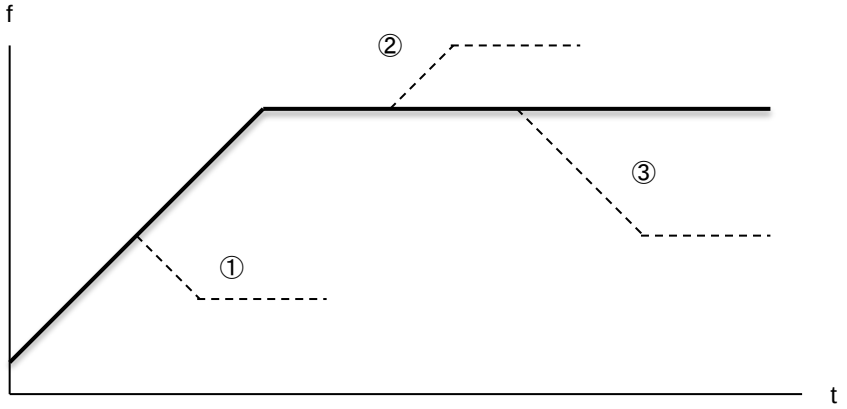
An example of the speed pattern when RUR = 27



## *7.5 Changing speed patterns while in operation*

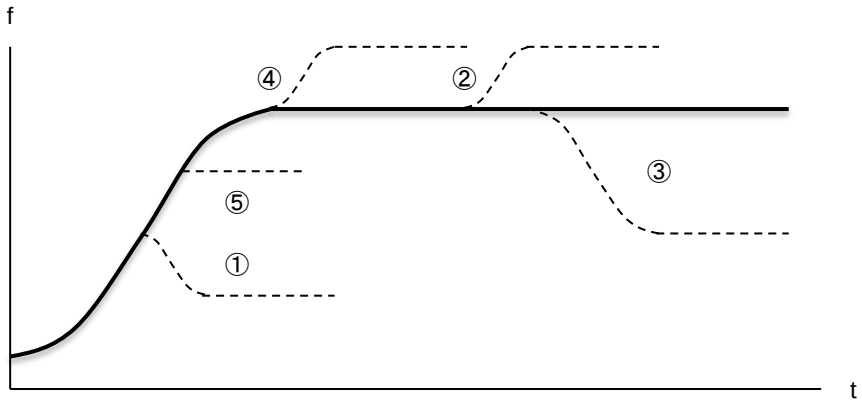
By changing RFL register value or RFH register value during operation, operation speed can be changed on the fly (speed override). By changing RUR, RDR, RUS, RDS register values, acceleration speed can be changed.
However, if auto ramping-down point setting is enabled (RMD.MSDP = 0) in such operation as positioning mode (RMD.MOD = 41h), do not change values for RFL, RUR, RDR, RUS, or RDS registers. The auto ramping-down point setting function will not work correctly.

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




1) When you use a smaller RFH register value while accelerating, a motor accelerates / decelerates until it reaches the set speed.
2), 3) When you change RFH register value after acceleration/deceleration is complete, a motor will continue accelerating or decelerating until it reaches a new speed.

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




1) When you use a smaller RFH register value while accelerating and if ((changed speed) < (speed before change)), a motor will decelerate using an S-curve until it reaches the set speed.
5) When you use a smaller RFH register value while a motor is accelerating and if ((changed speed) ≦ (speed before change)),the motor will accelerate without changing the S-curve's characteristic until it reaches the set speed.
4) When you use a larger RFH register value while accelerating, the motor 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 register value is changed after the acceleration/deceleration is complete, the motor will accelerate/decelerate using an S-curve until it reaches a set 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.
Reset the G9103C while RST= L level.
While resetting, input at least 10 cycles of CLK signal when CKSL= L level. When CKSL = H level, input at least 20 cycles of CLK 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 ~ P7 terminals | Input terminal |
| STA, STP terminals | H level |
| OUT, DIR terminals | H level |
| ERC terminal | H level |
| BSY/PH1, FUP/PH2, FDW/PH3, MVC/PH4 terminals | H level |

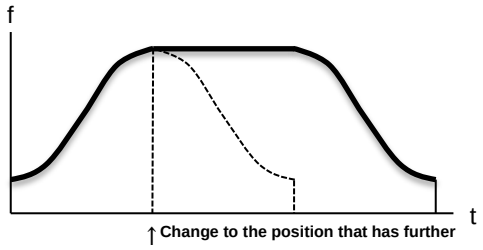
## 8.2 Position override

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

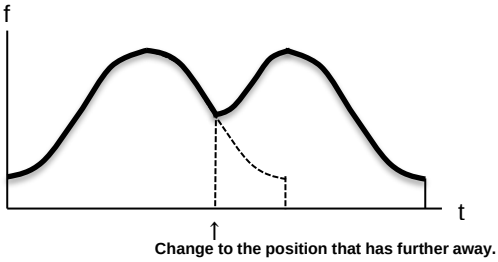
### 8.2.1 Target position override 1

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

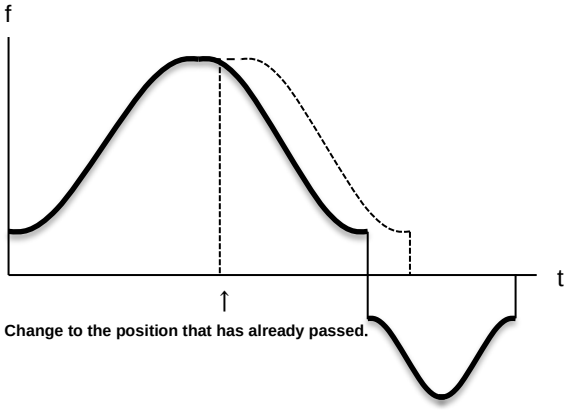
1) If a new target position is further away from the original target position during acceleration or constant speed operation, the motor will maintain the operation using the same speed pattern and it will complete the positioning operation at the position specified in the new data.




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

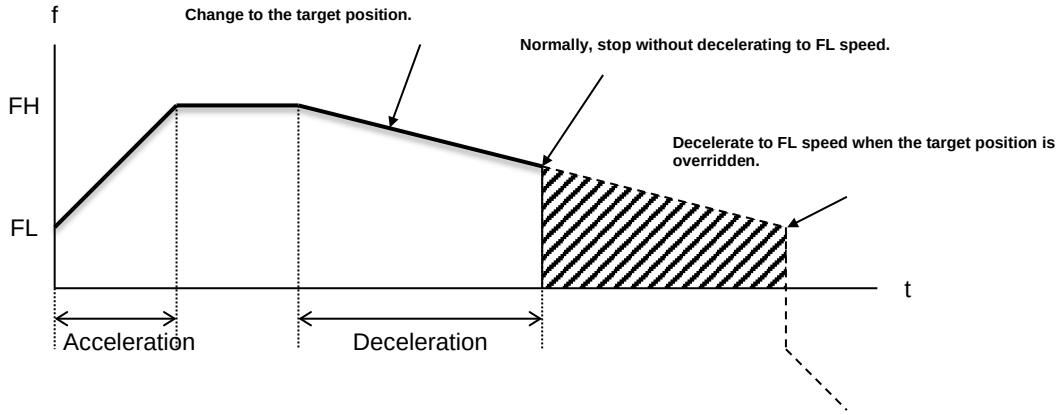



3) If the current position has already passed over a new target position, or the target position is changed to a position that is closer than the original position during deceleration, a motor will decelerate and stop. Then, the operation will reverse and complete the positioning operation at the position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




The motor accelerates/decelerates only when starting in high speed. Target position data (RMV) can be rewritten any number of times until positioning operation is complete.

Note1: If auto ramping-down point setting function is enabled and the "deceleration time > acceleration time x 2", it may be the case that the motor cannot reduce the speed to FL speed, as shown below. In this case, if a target position is set closer than original position while a motor is decelerating, a motor will decelerate along the deceleration curve from original target position, and then decelerate to FL speed and finally stop. Then the motor will start moving in opposite direction to the new position.
Therefore, the overrun beyond the original target position will occur during deceleration (shaded area in the below figure).




To avoid creating an overrun condition, make sure that deceleration time is less than two times of the acceleration time. If the deceleration time is more than double of the acceleration time, set the ramping-down point manually.

Note 2: The position override is only enabled during operation (FL, constant speed, FH constant speed, accelerating, decelerating, or backlash correction). If the speed is overridden just before the motor stops, the speed override may not be accepted. If you need to override a speed just before stop, you must determine if the G9103C can accept the override or not by the stop position.

By using WRMVOR (0080h) command when writing to RMV register, the center LSI outputs an interrupt request signal when the G9103C fails to override. In this case, determine whether the override is accepted or not by the stop position, too. The cause of error interrupt can be checked in REST register.

The center LSI recognized that override had failed when an WRMVOR (0080h) command is written while a motor stops. Therefore, if you try to write WRMVOR (0080h) command before the motor starts, the center LSI outputs an interrupt request signal.

### 8.2.2 Target position override 2 (PCS signal)

If PCS signal is set to be enabled (RMD.MPCS = 1) in positioning operation (RMD.MOD=41h), the G9103C will perform positioning operations for an amount specified in RMV register, based on the timing when PCS signal turns ON and writing STAON (0028h) command after pulse output start.
The logic of a PCS signal can be changed. The PCS terminal status can be checked using RSTS.SPCS bit.

| | |
|---|---|
| Input function of PCS signal <PRMD.MPCS><br>0: Disabled.<br>1: Enabled. Number of pulses is controlled after PCS signal turns ON. | [RMD] (WRITE)<br>15 8<br>- - n - - - - - |
| Input logic of PCS signal <RENV1.PCSL><br>0: Negative logic.<br>1: Positive logic. | [RENV1] (WRITE)<br>31 24<br>- - - - - - - n |
| Input status of PCS signal <RSTS.SPCS><br>0: OFF.<br>1: ON. | [RSTS] (READ)<br>15 8<br>- n - - - - - - |
| Input substitute for PCS signal <STAON><br>Perform processing that is identical to those performed by supplying a PCS signal. | [Command]<br>0028h |

## *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 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 (+) direction operation pulses from OUT terminal, and outputs (−) direction operation pulses from DIR terminal. |
| 90 degree phase difference mode: | This mode is to output 90 degree phase difference through OUT and DIR terminals. One cycle 90-degree phase difference is equivalent to 4x (4 pulses) of common or two-pulse mode. |

The output mode for command pulses is set in RENV1.PMD.
If motor drivers using the common pulse mode need a lag time bit since direction signal changes, until receiving a command pulse, use a direction change timer.
If direction change timer function is enabled (RENV1.DTMF = 0), operation can be delayed for one direction change timer unit (0.2 ms), after changing the direction signal.

If function to reverse motor rotation direction is enabled (RENV1.MREV = 1), the wave profiles of "When feeding in (+) direction", and "When feeding in (−) direction" in the following figure are switched and the direction of the motor rotation can be reversed. (The direction of the motor rotation is also reversed when excitation sequence is output.)

Output pulse mode <RENV1.PMD>

[RENV1 (WRITE)

7 0

| - | - | - | - | - | n | n | n |
|---|---|---|---|---|---|---|---|

| PMD | When feeding in the positive direction | | When feeding in the negative direction | |
|---|---|---|---|---|
| | OUT signal | DIR signal | OUT signal | DIR signal |
| 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 | |

Direction change timer function <RENV1.DTMF>
0: Enabled.
1: Disabled.

[RENV1] (WRITE)

31 24

| - | - | - | - | - | n | - | - |
|---|---|---|---|---|---|---|---|

Function to reverse motor rotation direction <RENV1.MREV>
0: Enabled.
1: Disabled.

[RENV1] (WRITE)

31 24

| - | n | - | - | - | - | - | - |
|---|---|---|---|---|---|---|---|

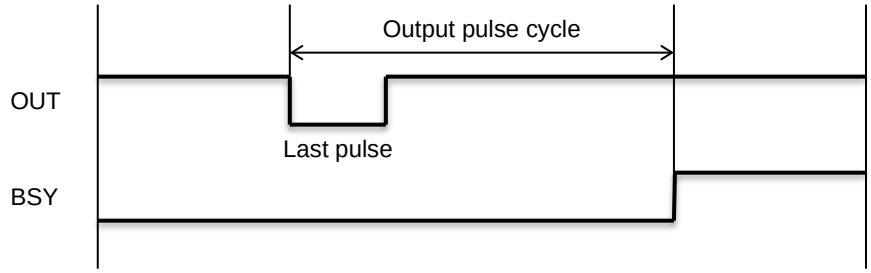
8.3.2 Control the output pulse width and operation complete timing

In order to put the timing of stopping forward, this LSI controls output pulse width.

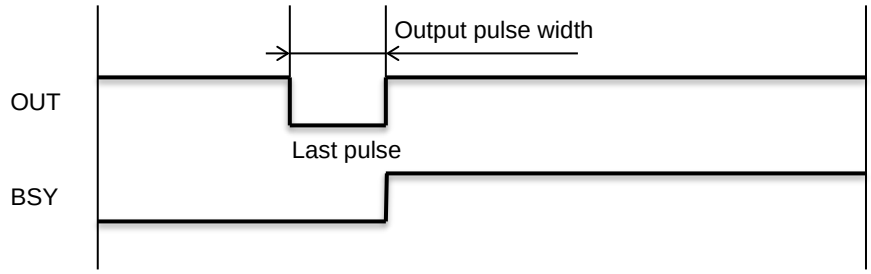
When control of output pulse width is enabled (RENV1.PDTC = 0), in the case that output speed of command pulse is slower than approximately 2.44kpps. Output pulse width is kept constant approx. 200 µs. For faster pulse speeds than this, output pulse width fluctuates at approximate 50% duty rate. By setting control of input pulse speed is disabled (RENV1.PDTC = 1), the output pulse width fluctuates at approximate 50% duty rate even if output speed of command pulse is slower than approximately 2.44kpps. In this case, output pulse width may exceed approximately 200 µs.

Also, when setting RMD.METM bit, the operation complete timing can be changed. If control of output pulse speed is enabled (RENV1.PDTC = 0) and operation complete timing is set to the time when output pulse width is complete (RMD.METM = 1), operation can finished within approximately 200µs after the G9103C outputs the last pulse regardless of output speed of command pulse.

1) When RMD.METM = 0 (When output pulse cycle is complete)




2) When RMD.METM = 1 (when output pulse width is complete)




Note: When you start automatically next operation using pre-registers, set operation complete timing to “when output pulse cycle is complete”(RMD.METM = 0). If you set it to “When output pulse width is complete” (RMD.METM = 1), the interval between the last pulse and the initial pulse of next operation will be narrow such as approximately 750 ns.

| | |
|---|---|
| Operation complete timing <RMD.METM><br>0: When output pulse cycle is complete.<br>1: When output pulse width is complete. | [RMD] (WRITE)<br>15 8<br>- - - - n - - - |
| Function to control output pulse width <RENV1.PDTC><br>0: Enabled. In the case that output speed of command pulse is slower than approximately 2.44kpps, pulse width is kept constant (Approx. 200 µs).<br>1: Disabled. Output pulse width is fluctuated at approximately 50% duty rate. | [RENV1] (WRITE)<br>31 24<br>- - n - - - - - |

## *8.4 Idling control*

When starting acceleration or deceleration operation of stepper motor, acceleration can be started after the output of a few pulses at FL speed and that eliminates occurrence of output of step by quick acceleration. This pulse is referred to as "idling pulse". Set the number of pulses for idling in RENV5.IDL bit.
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 command pulses. Therefore, the initial speed will be faster than the FL speed.
To use this function, enter a value "n" of 2 ~ 7. The LSI will start the acceleration by beginning its output on the nth pulse. Therefore, the initial speed will be the FL speed and the FL speed can be set to start automatically at upper speed limit of stepper motor. The number of idling pulses will be “n−1” pulse.

Even if this function is used with the positioning mode, the total feed amount will not change.

[Setting idling pulses and the acceleration start timing]

| | |
|---|---|
| Number of idling pulses (0 ~ 6 pulses) < RENV2.IDL><br>0 : Disabled.<br>1 ~ 7: n−1 pulse. “n” is a setting value. | [RENV2] (WRITE)<br>23 16<br>- n n n - - - - |
| Idling control counter value <RSPD.IDC><br>Read idling control counter value. | [RSPD] (READ)<br>31 24<br>- - - - n n n |

## *8.5 Mechanical external input control*

### 8.5.1 +EL, −EL signal

When an end limit signal (+EL signal in (+) direction and −EL signal in (−) direction) in the feed direction turns ON while operating, a motor will stop immediately or decelerate and stop. After stopping, even if EL signal is turned OFF, the motor will remain stopped. For safety, keep EL signal ON until the position reaches the end of the stroke.
If EL signal is ON when writing a start command, a motor cannot start moving in the direction of the EL signal that is ON.
By setting RENV1.ELM bit , the stopping pattern to be used when 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 a motor stops after passing through the EL position.

The minimum pulse width of EL signal is 4 µs when input filter is ON. When input filter is disabled, the minimum pulse width is 0.1 µs.
EL signal can be monitored by reading RSTS register.
By reading REST register, you can check for an error interrupt caused by EL signal turning ON.
When in the timer mode, this signal is ignored. Even in this case, EL signal can be checked by reading RSTS register.
Input logic of the EL signal can be set using the ELL terminal.

| | |
|---|---|
| Input logic of EL signal <ELL terminal><br>L: Positive logic input<br>H: Negative logic input | |
| Stop method to be used when EL signal turns ON <RENV1.ELM><br>0: Stop immediately<br>1: Decelerate and stop | [RENV1] (WRITE)<br>7 0<br>- - - - n - - - |
| Input status of +EL signal <RSTS.SPEL><br>0: OFF.<br>1: ON. | [RMD] (READ)<br>7 0<br>n n - - - - - - |
| Input status of −EL signal <RSTS.SMEL><br>0: OFF.<br>1: ON. | [RMD] (READ)<br>7 0<br>n n - - - - - - |
| Input filter function of EL signal <RENV1.FLTR><br>0: Enabled. Signal shorter than 150 ns will be ignored.<br>1: Disabled. | [RENV1] (WRITE)<br>31 24<br>- - - - - - n - |

Note 1: Operation after EL signal turning ON 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.

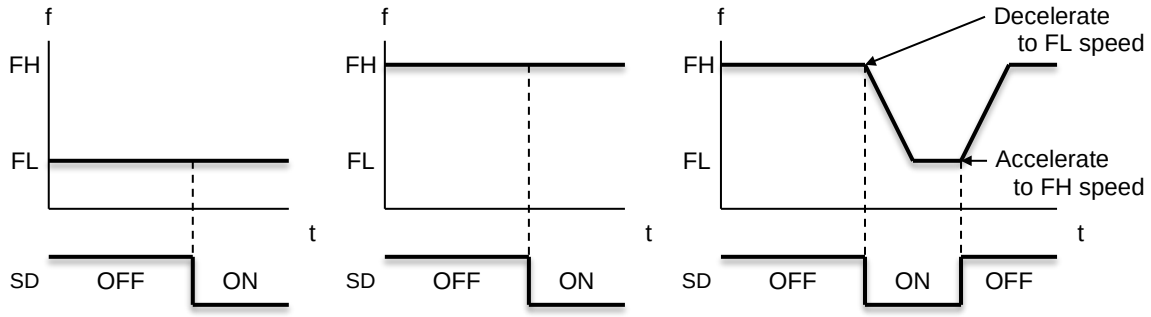
### 8.5.2 SD signal

If SD signal input is disabled (RMD.MSDE=0), SD signal input is ignored. If SD signal input is enabled (RMD.MSDE = 1) and the SD signal is turned ON while in operation, the LSI makes a motor to : 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 bit and RENV1.SDLT bit.

1) Deceleration <RENV1.SDM = 0, RENV1.SDLT = 0>

- While feeding at constant speed, SD signal is ignored. While in high speed operation a motor decelerates to FL speed when SD signal is turned ON. After decelerating, or while decelerating, if the SD signal turns OFF, the motor will accelerate to FH speed.
- If SD signal is turned ON when a high speed command is written, a motor will operate at FL speed. When SD signal is turned OFF, the motor will accelerate to FH speed.

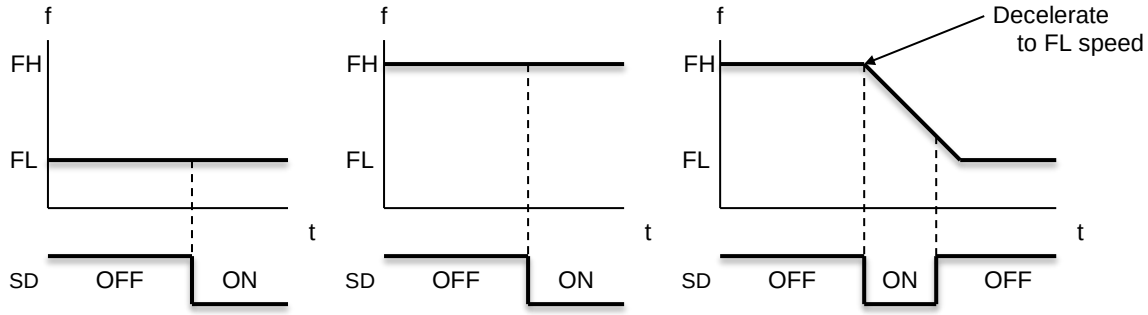
[FL constant speed operation] [FH constant speed operation] [High speed operation]




2) Latch and decelerate <RENV1.SDM = 0, RENV1.SDLT = 1>

- While feeding at constant speed, SD signal is ignored. While in high speed operation, a motor decelerates to FL speed by turning SD signal ON. Even if SD signal is turned OFF after decelerating or while decelerating, a motor will continue moving at FL speed and will not accelerate to FH speed.
- If SD signal is turned ON while writing a high speed command, a motor will feed at FL speed. Even if SD signal is turned OFF, a motor will not accelerate to FH speed.

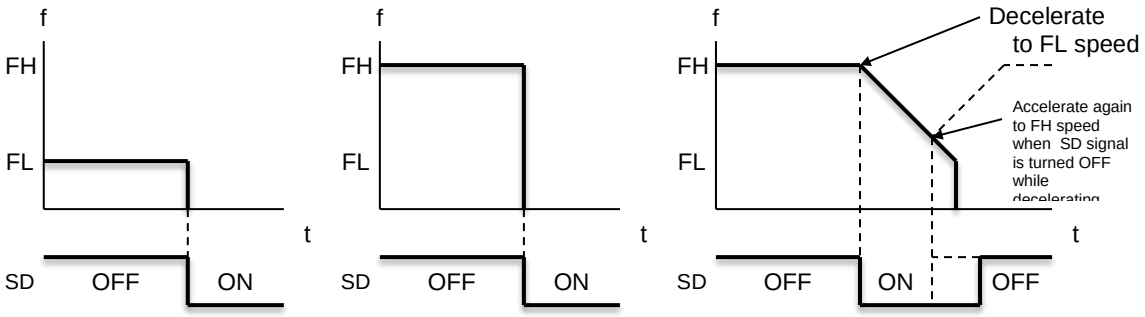
[FL constant speed operation] [FH constant speed operation] [High speed operation]



3) Deceleration stop <RENV1.SDM=1, RENV1.SDLT = 0>

- If SD signal is turned ON while in constant speed operation, a motor stops. While in high speed operation, a motor decelerates to FL speed when a SD signal is turned ON, and then stops. If SD signal is turned OFF during deceleration, a motor accelerates to FH speed.
- If SD signal is turned ON after writing a start command, a motor completes its operation without another start.
- When a motor stops, error interrupt occurs (REST.ESSD=1) and the center LSI outputs an interrupt request signal.

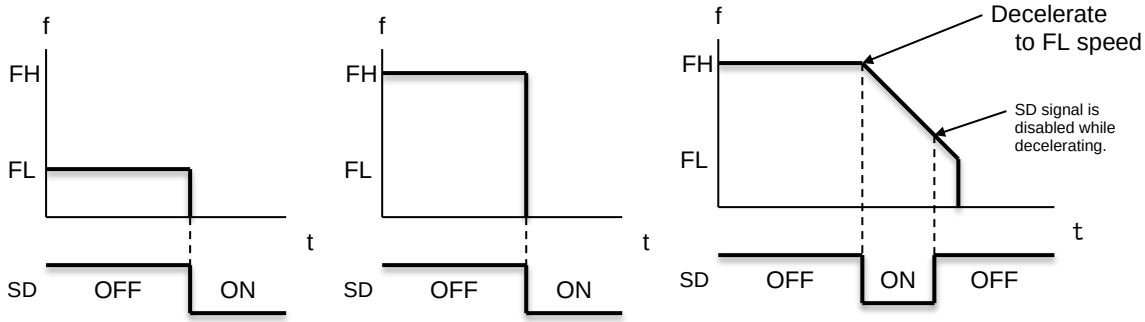
[FL constant speed operation] [FH constant speed operation] [High speed operation]




4) Latched, deceleration stop <RENV1.SDM = 1, RENV1.SDLT=1>

- If SD signal is turned ON while in constant speed operation, a motor stops. If SD signal is turned ON while in high speed operation, a motor decelerates to FL speed and then stop. Even if SD signal is turned OFF during deceleration, the motor does not accelerate.
- If SD signal is turned ON while writing a start command, a motor completes its operation without another start.
- When a motor stops, error interrupt occurs and the center LSI outputs an interrupt request signal.

[FL constant speed operation] [FH constant speed operation] [High speed operation]



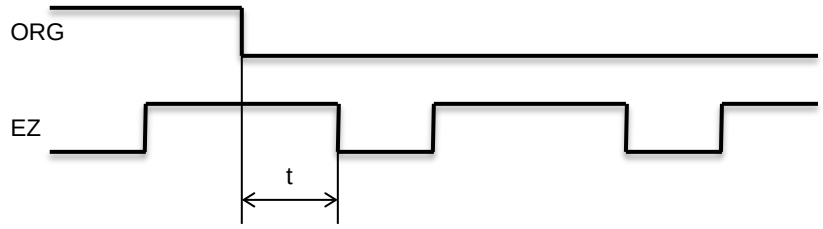

Input logic of SD signal can be changed. If SD latched input is enabled (RENV1.SDLT=1) as input method of SD signal, Latch signal will be reset if SD signal is OFF at the next start. Latch signal is also reset when input of SD latch signal is disabled (RENV1.SDLT=0).
The minimum pulse width of the SD signal is 4 µs when input filter is enabled (RENV1.FLTR=0) and 0.1 µs when input filter is disabled (RENV1.FLTR=0).
SD signal and SD latch signal can be checked by reading RSTS register. SD signal terminal status can be checked by reading RSTS register. By reading REST register, you can check for an error interrupt caused by the SD signal turning ON.

| | |
|---|---|
| Input function of SD signal <RMD.MSDE><br>0: Enabled.<br>1: Disabled. | [RMD] (WRITE)<br>15 8<br>- - - - - - - n |
| Input logic of SD signal <RENV1.SDL><br>0: Negative logic<br>1: Positive logic | [REMV1] (WRITE)<br>7 0<br>- n - - - - - - |
| Operation to be used when SD signal is turned ON <RENV1.SDM><br>0: Decelerates and feeds at FL constant speed<br>1: Decelerates and stops. | [RENV1] (WRITE)<br>7 0<br>- - - n - - - - |
| Input function of SD latch signal <RENV1.SDLT><br>0: Disabled.<br>1: Enabled.<br>SD latch signal will be reset when SD signal is OFF at the start and when SD latch signal input is enabled (RENV1.SDLT). | [RENV1] (WRITE)<br>7 0<br>- - n - - - - - |
| Input status of SD latch signal <RSTS.SSD><br>0: OFF<br>1: ON | [RSTS] (READ)<br>15 8<br>- - - - - - n - |
| Input status of SD signal <RSTS.SDIN><br>0: OFF<br>1: ON | [RSTS] (READ)<br>15 8<br>- - - - - n - - |
| Error interrupt status <REST.ESSD><br>1: Decelerate and stop caused by the SD signal turning ON | [REST] (READ)<br>15 8<br>- - - - - - - n |
| Input filter function of SD signal <RENV1.FLTR><br>0: Enabled. Signals shorter than 150 ns will be ignored.<br>1: Disabled. | [RENV1] (WRITE)<br>31 24<br>- - - - - - n - |

### 8.5.3 ORG and EZ signals

These signals are enabled in the origin return modes (origin return, leaving origin position, and origin position search) and in the operation mode to feed EZ count amount. Specify operation mode and an operation direction using RMD.MOD bit.
The G9103C latches ORG signal on the rising edge of command pulses (negative logic). The minimum pulse length of ORG signal is one cycle period of command pulses.
Since ORG signal input is latched internally, there is no need to keep the signal ON externally.
The ORG latch signal is reset when the motor stops.
Input logic of ORG signal can be changed by RENV1.ORGL bit and EZ signals can be changed using RENV2.EZL bit.
By reading RSTS register, you can check the status of ORG and EZ terminals.
For details about origin return operation modes, see “6.4 Origin position operation mode”.

The following figure shows the timing to count the first EZ signal ON after ORG signal turns ON in the operation mode to count EA signal ON.




(i) When t ≥ 100μs, EZ signal is counted.
(ii) When 50μs < t < 100μs, it is undetermined to count EZ signal.
(iii) When t ≤ 50μs, EZ signal is not counted.

| | |
|---|---|
| Operating mode to enable ORG and EZ signals <RMD.MOD><br>001 0000: Origin return in (+) direction<br>001 0010: Leaving from the origin position in (+) direction<br>001 0101: Origin position search in (+) direction<br>010 0100: Operation to feed EZ count amount in (+) direction<br>001 1000: Origin return in (−) direction<br>001 1010: Leave from the origin position in (−) direction<br>001 1101: Origin position search in (−) direction<br>010 1100: Operation to feed EZ count amount in (−) direction | [RMD] (WRITE)<br>7 0<br>0 n n n n n n n |
| Origin return method <RENV3.ORM><br>See “5.4.4.4. RENV3: Environmental setting 3 register”. | [REMV3] (WRITE)<br>7 0<br>- - - - n n n n |
| Input logic of ORG signal <RENV1.ORGL><br>0: Negative logic.<br>1: Positive logic | [RENV1] (WRITE)<br>7 0<br>n - - - - - - - |
| Input status of ORG signal <RSTS.SORG><br>0: OFF.<br>1: ON. | [RSTS] (READ)<br>15 8<br>- - - - - - - n |
| EZ count value to be used for Origin return <RENV3.EZD><br>Set an EZ count number as a condition to complete origin return using EZ count.<br>Specify a value (count number –1) in RENV3.EZD bit. Setting range is 0 ~ 15. | [RENV3] (WRITE)<br>7 0<br>n n n n - - - - |
| Input logic of EZ signal <RENV2.EZL><br>0: Falling edge<br>1: Rising edge | [RENV2] (WRITE)<br>15 8<br>- - - n - - - - |

| | |
|---|---|
| Input status of EZ signal <RSTS.SEZ><br>0: OFF<br>1: ON | [RSTS] (READ)<br>23 16<br>- - - - - - - n |
| Input filter function of EZ signal. <RENV2.EINF><br>0: Enabled. Signals shorter than 150 ns will be ignored.<br>1: Disabled. | [RENV2] (WRITE)<br>15 8<br>- - - - - - - 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, a servomotor 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 pulse output complete timing to determine when an operation is complete.
When INP signal input is used to indicate completion status of an operation, BSY signal when an operation is complete, a stop condition of the main status (MSTS.SINT, MSTS.SEND, MSTS.SERR, MSTS.SEVT, MSTS.SBSY), and an operation status of extension status (RSTS.CND) will also change when the INP signal is input.

Input logic of INP signal can be changed by setting RENV1.INPL bit.
The minimum pulse width of INP signal is 4 µs when the input filter is enabled (RENV1.FLTR=0). If the input filter is disabled (RENV1.FLTR = 1), the minimum pulse width will be 0.1 µs.
If INP signal is already ON when the G9103C is finished outputting pulses, it treats the operation as complete, without any delay.
INP signal can be checked by reading RSTS register.

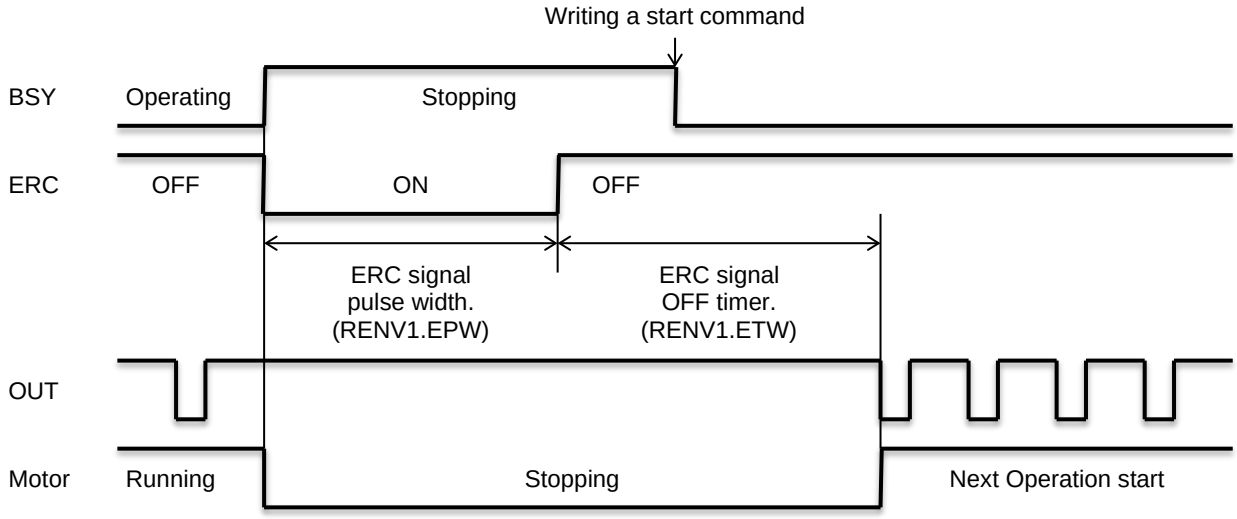
| | |
|---|---|
| Input function of INP signal <RMD.MINP><br>0: Disabled. The terminal status can be checked by RSTS.SINP bit.<br>1: Enabled. Operation complete delays until INP signal turns ON. | [RMD] (WRITE)<br>15 8<br>- - - - - - n - |
| Input logic of INP signal <RENV1.INPL><br>0: Negative logic<br>1: Positive logic | [RENV1] (WRITE)<br>23 16<br>- n - - - - - - |
| Input status of INP signal <RSTS.SINP><br>0: OFF<br>1: ON | [RSTS] (READ)<br>23 16<br>- - - - n - - - |
| Input filter function of INP signal <RENV1.FLTR><br>0: Enabled. Signals shorter than 4 µs will be ignored.<br>1: Disabled. | [RENV1] (WRITE)<br>31 24<br>- - - - - - 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 a 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 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 bit. If a lag time is needed until the servo driver accept command pulses after the ERC signal returns to OFF, the time for the ERC signal OFF timer can be set by RENV1.ETW bit.

Note. ERC signal is output from the ERC/CDWN terminal. When ERC signal is selected (RENV1.CDWN=0). When CDWN signal is selected (RENV1.CDWN=1), the CDWN signal is output.



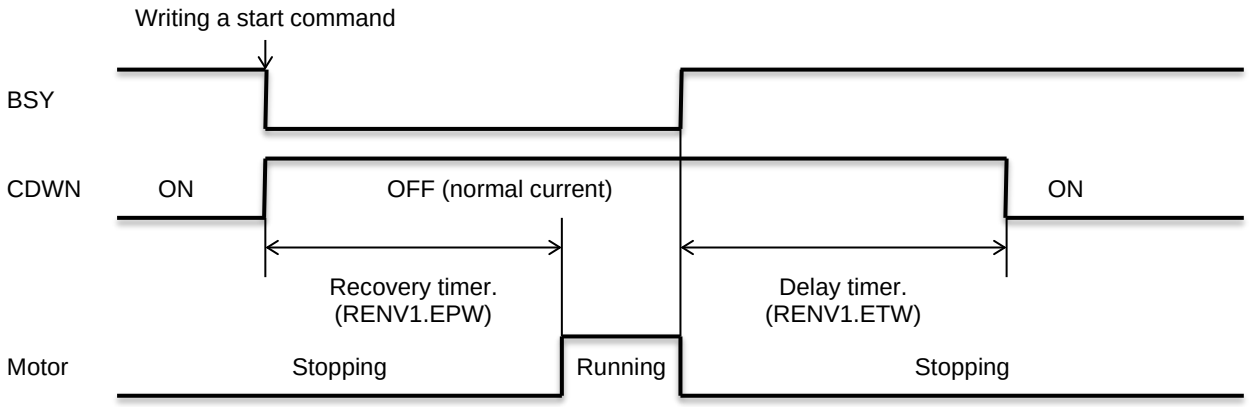

In order to output ERC signal at completing origin return operation, set auto output of ERC signal to enabled (RENV1.EROR=1). For the output timing of ERC signal, see the timing wave shape in “6.4.1 Origin return operation”.
In order to output ERC signal at the stopping by inputting of EL, ALM or EMG signals, and a CMEMG (0005h) command, set auto output of ERC signal to enabled (RENV1.EROE = 1). When the motor decelerates and stops, ERC signal is not output even if automatic output is selected.
ERC signal can be output by writing ERCOUT (0024h) command.
The output logic of the ERC signal can be changed by the setting RENV1 register and ERC signal can be monitored by reading RSTS register.

| | |
|---|---|
| Output signal of ERC/CDWN terminal <RENV1.CDWN><br>0: ERC signal.<br>1: CDWN signal. | [REMV1] (WRITE)<br>31 24<br>n - - - - - - - |
| Function to output ERC signal automatically at error stop <RENV1.EROE><br>0: Disabled.<br>1: Enabled. | [REMV1] (WRITE)<br>15 8<br>- - - - - n - - |
| Function to output ERC signal automatically at origin return complete <RENV1.EROR><br>0: Disabled.<br>1: Enabled. | [RENV1] (WRITE)<br>15 8<br>- - - - n - - - |
| Set the width of ERC signal output <RENV1.EPW><br>000: 12.5 µs. 001: 100 µs. 010: 400 µs. 011: 1.6 ms.<br>100: 13 ms. 101: 51 ms. 110: 102 ms. 111: Level output. | [RENV1] (READ)<br>15 8<br>- n n n - - - - |
| Output logic of ERC signal <RENV1.ERCL><br>0: Negative logic.<br>1: Positive logic. | [RENV1] (WRITE)<br>15 8<br>n - - - - - - - |
| Off timer time of ERC signal <RENV1.ETW><br>00:0 µs. 10:1.6 ms.<br>01:13 µs. 11:102 ms. | [RENV1] (WRITE)<br>23 16<br>- - - - - - n n |
| Output status of ERC signal <REST.SERC><br>0: OFF.<br>1: ON. | [RSTS] (READ)<br>15 8<br>n - - - - - - - |
| Output command of ERC signal <ERCOUT><br>Output ERC signal. | [Command]<br>0024h |
| Reset command of ERC signal <Command><br>Reset ERC signal. | [Command]<br>0025h |

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 while stopping. (Current down control)
This LSI has a function for outputting current down signals (CDWN).
By a start command is written, G9103C turns CDWN signal off (normal current). After the setting time has elapsed after the motor stops, G9103C starts the motor. If the setting time elapses after the motor stops, this LSI turns CDWN signal ON (Current down).
When you write a next start command while the CDWN signal still be off after a motor stops, this LSI starts to operate the motor immediately.
The timer at the start and stop doubles as the timer to control ERC signal. However, the time range that you can select by RENV1.EPW bit and RENV1.ETW.bit is different from the timer range to control ERC signal.

Note. CDWN signal is output from the ERC/CDWN terminal. When CDWN signal is selected (RENV1.CDWN=1), the CDWN signal is output. When ERC signal is selected (RENV1.CDWN=0), the ERC signal is output.




Output logic of the CDWN signal can be changed by setting RENV1 register and CDWN signal can be checked by reading RSTS register.

| | |
|---|---|
| Output signal of ERC/CDWN <RENV1.CDWN><br>0: ERC signal.<br>1: CDWN signal. | [REMV1] (WRITE)<br>31 24<br>n - - - - - - - |
| Time to return current <RENV1.EPW><br>000: 6.4 ms 001: 13.0 ms 010: 25.6 ms 011: 51.0 ms<br>100: 102.0 ms 101: 205.0 ms 110: 410.0 ms 111: Prohibited setting. | [REMV1] (WRITE)<br>15 8<br>- n n n - - - - |
| Output logic of CDWN signal <RENV1.ERCL><br>0: Negative logic<br>1: Positive logic | [REMV1] (WRITE)<br>15 8<br>n - - - - - - - |
| Time to delay current down <RENV1.ETW><br>00: 51 ms 01: 102 ms 10: 205 ms 11: 410 ms | [REMV1] (WRITE)<br>23 16<br>- - - - - - n n |
| Output status of CDWN signal <RSTS.SERC><br>0: OFF<br>1: ON | [REMV1] (WRITE)<br>15 8<br>n - - - - - - - |
| Reset command of CDWN signal <ERCRST><br>Reset CDWN signal. | [Command]<br>0025h |

### 8.6.4 ALM signal

Inputs the alarm signal (ALM). If the ALM signal turns ON while operating, a motor stops immediately or decelerates and stops. However, the motor decelerates and stops by inputting ALM signal only at the high-speed start. At constant speed start, a motor stops immediately.
If ALM signal is ON at writing a start command, any pulses are not output.
The minimum pulse width of ALM signal is 4 µs when the input filter is enabled (RENV1.FLTR = 0). If the input filter is disabled (RENV1.FLTR = 0), the minimum pulse width is 0.1 µs.
The input logic of the ALM signal can be changed and the ALM signal can be checked by reading RSTS register.

| | | |
|---|---|---|
| Operation to be used when ALM signal turns ON.<br>0: Stop immediately.<br>1: Decelerate and stop. (Only at high-speed start) | <REMV1.ALMM> | [RENV1] (WRITE)<br>15 8<br>- - - - - - - n |
| Input logic of INP signal<br>0: Negative logic<br>1: Positive logic | <RENV1.ALML> | [RENV1] (WRITE)<br>15 8<br>- - - - - - n - |
| Read ALM signal<br>0: OFF<br>1: ON | <RSTS.SALM> | [RSTS] (READ)<br>7 0<br>- - n - - - - - |
| Error interrupt status<br>1: Stop by ALM signal ON | <RSTS.ESAL> | [RSTS] (READ)<br>7 0<br>- - n - - - - - |
| Input filter function of ALM signal<br>0: Enabled. Pulses shorter than 4 µs are ignored.<br>1: Disabled. | <RENV1.FLTR> | [RENV1] (WRITE)<br>31 24<br>- - - - - - n - |

## *8.7 External start and simultaneous start*

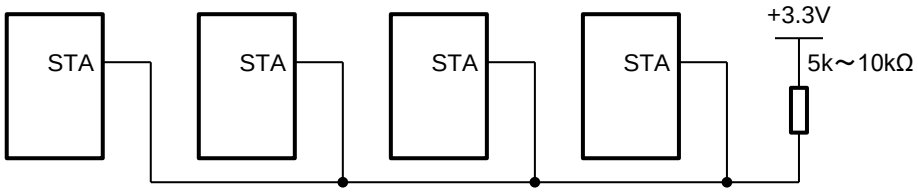
This LSI can start operation by external STA signal using STA terminals. Set holding start to enabled (PRMD.MSY = 1) and the LSI will start feeding when STA turns ON after writing a start command.
Logic of STA signal cannot be changed.
When you want to control multiple axes using more than one LSI, connect the STA terminal on each LSI with another STA terminal and input external STA signal, axes that are "waiting for STA input" can be started simultaneously. STA signal can be output from STA terminal, instead of external STA signal.
Event interrupt cause ON (RIST.ISSA = 1) and event interrupt ON (MSTS.SEVT = 1) can be checked when STA signal turns ON. In order to occur event interrupt when STA signal ON, the setting “when STA signal is ON” is enabled (RIRQ.IRSA = 1) needs to be set.
Operation status waiting for STA signal input (RSTS.CND) and input status of STA signal (RSTS.SSTA) can be checked by reading RSTS register.

<How to make a simultaneous start>
After setting holding start of the axes you want to start simultaneously is enabled (RMD.MSY = 1), write a start command and put the axes in the "waiting for STA signal" status. Then, start the axes simultaneously by either of the methods described below.

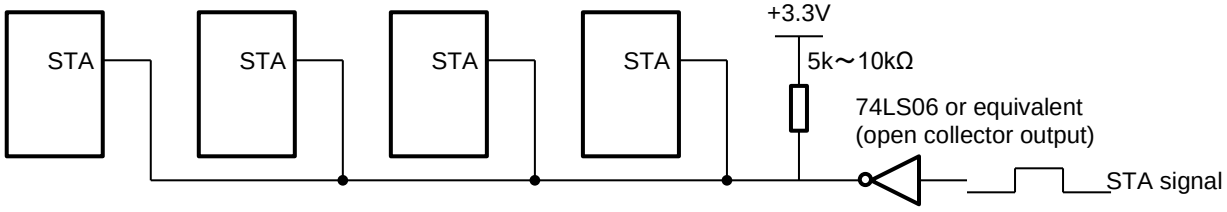
1) By writing a CMSTA (0006h) command, the LSI will output a one shot signal of approx. 0.4 µs from STA terminal.
2) Input STA signal from outside to STA terminal.
   Supply STA signal after driving the terminal with open collector output (74LS06 or equivalent).

The STA signal can be supplied as level trigger or edge trigger inputs. However, when level trigger input is selected, if start command is written when STA signal turns ON and a, a motor starts immediately.
After connecting terminal on each LSI, movement of each axis can still be started independently using start commands.
To release the "waiting for STA input" condition, write an immediate stop command (0049h).
Movements on the axes can be started simultaneously by a start command of broadcast communication without STA terminal.
(Regarding broadcast communication commands, see “5.1.2.4 Broadcast communication”)

1) To start axes controlled by different LSIs simultaneously, connect the LSIs as follows.




2) To start simultaneously from an external circuit, connect the LSIs as follows.




As start signal, input a one shot input signal with a pulse width of at least approx. 0.2 µs.

| | |
|---|---|
| Holding start function <RMD.MSY><br>0: Disabled.<br>1: Enabled. | [RMD] (WRITE)<br>15 8<br>- n - - - - - - |
| Input type of STA signal <RENV1.STAM><br>0: Level trigger (L level)<br>1: Edge trigger (falling edge) | [RENV1] (WRITE)<br>23 16<br>- - - - - n - - |
| Input status of STA signal <RSTS.SSTA><br>0: OFF (H level)<br>1: ON (L level) | [RSTS] (READ)<br>15 8<br>- - - - n - - - |
| Operation status <RSTS.CND><br>0001: Waiting for STA input | [RSTS] (READ)<br>7 0<br>- - - - n n n n |
| Event interrupt cause <RIRQ.IRSA><br>1: When STA input is ON. | [RIRQ] (WRITE)<br>15 8<br>- - - n - - - - |
| Event interrupt status <RIST.ISSA><br>1: When STA signal is ON. | [RSTS] (READ)<br>15 8<br>- - - n - - - - |
| Output of STA signal <CMSTA><br>Output a one shot signal of pulse width 0.4 µs from STA terminal.<br>(Own axis starts simultaneously because a signal is input again.) | [Command]<br>0006h |
| Input substitute of STA signal <SPSTA><br>Only own axis starts because of signal is not output. | [Command]<br>002Ah |

## *8.8 External stop / simultaneous stop*

This LSI can execute an immediate stop or a deceleration stop triggered by an external signal using STP terminal. If input of STP signal is enabled (RMD.MSPE = 1), a motor stops immediately or decelerates and stops when STP signal turns ON. Input logic of STP terminal cannot be changed.

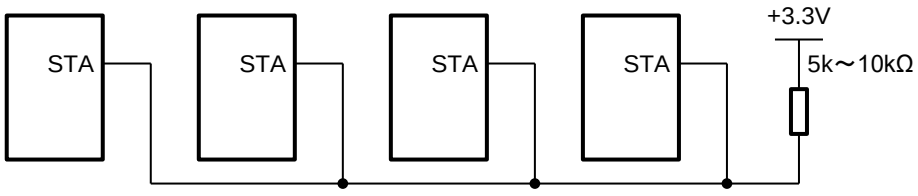
When multiple LSIs are used to control multiple axes, connect STP terminal on each LSI with another STP terminal and input the same signal so that the axes that are set to stop on a STP signal can be stopped simultaneously. Instead of external STP signal, STP signal can also be output from STP terminal.
The cause motor stops by STP signal turning ON can be checked by error interrupt cause turning ON (REST.ESSP = 1) and error interrupt turning ON (MSTS.SERR = 1). Input status of STP signal (RSTS.SSTP) is checked by Reading RSTS register.

<How to make a simultaneous stop>
After input of STP signal to stop axes to be stopped simultaneously is enabled (RMD.MSPE = 1), start movement on these axes.
Stop movements on these axes using any of the following three methods.

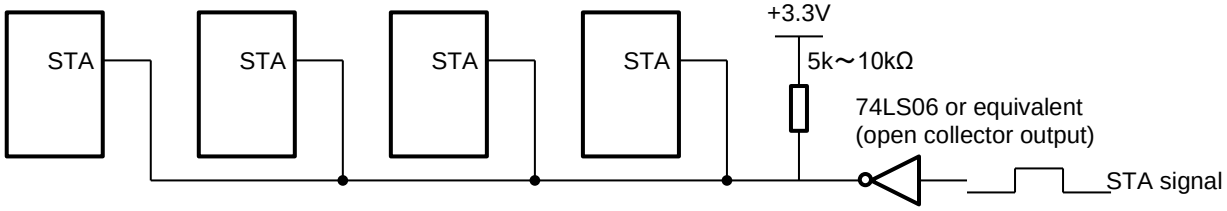
1. By writing a CMSTP (0007h) command, STP terminal will output a one shot signal that is approx. 0.4 µs.
2. Input STP signal from outside to STP signal.
   Input STP signal after driving by open collector output (74LS06 or equivalent).
3. The axis that STP signal input is enabled (RMD.MSP0 = 1) outputs a one shot signal that is approximately 0.4 µs when a stop caused by an error.

Even when STP terminals on LSIs are connected together, movement on each axis can still be stopped independently by using the stop command.
Without using STP terminals, stop command for broadcast communication can stop movements on axes simultaneously.
(Regarding broadcast communication commands, see “5.1.2.4 Broadcast communication”.)

1) To stop multiple LSIs simultaneously, connect the terminals as follows.




2) To stop simultaneously using an external circuit, connect as follows.




As STP signal, input a one shot signal with a pulse width of al least 0.2 µs.

| | |
|---|---|
| Input function of STP signal <RMD.MSPE><br>0: Disabled.<br>1: Enabled. (Stop immediately or decelerate and stop) | [RMD] (WRITE)<br>15 8<br>n - - - - - - - |
| Output of STP signal at error stop <RMD.MSPO><br>0: Disabled. (STP signal is not output automatically).<br>1: Enabled. (Output signal width 16 cycles of CLK signal is output automatically.) | [RMD] (WRITE)<br>23 16<br>- - - - - - - n |
| Operation to be used when STP signal turns ON. <RENV1.STPM)><br>0: Stop immediately.<br>1: Decelerate and stop. | [RENV1] (WRITE)<br>23 16<br>- - - - n - - - |
| Input status of STP signal <RSTS.SSTP><br>0: OFF. (H level)<br>1: ON. (L level) | [RSTS] (READ)<br>15 8<br>- - - n - - - - |
| Error interrupt status <REST.ESSP><br>1: When stopping by STP turning ON. | [REST] (READ)<br>7 0<br>- n - - - - - - |
| Output of STP signal <CMSTP><br>Outputs a one shot pulse of pulse width 0.4 µs in length from STP terminal.<br>Because output signal is input again, the own axis stops simultaneously. | [Command]<br>0007h |

## *8.9 Emergency stop*

This LSI has EMG terminal as an input terminal of emergency stop (EMG) signal.
While in operation, if EMG input turns ON or writing a CMEMG (0005h) command, a motor stops immediately. While EMG input remains ON, a motor cannot be operated.
Logical input of EMG terminal cannot be changed.

When a motor stops because the EMG signal 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 EMG signal is 4 µs when input filter is enabled (RENV1.FLTR = 0). If input filter is disabled (RENV1.FLTR = 0), the minimum pulse width will be 0.1 µs.
Input status of EMG signal can be checked by reading RSTS register.

| | |
|---|---|
| Input status of EMG signal <RSTS.SEMG><br>0: OFF.<br>1: ON. | [RSTS] (READ)<br>15 7<br>- - n - - - - - |
| Read the cause of an error interrupt <REST.ESEM><br>1. Stopped when the EMG signal turns ON. | [REST] (READ)<br>7 0<br>n - - - - - - - |
| Input filter function of EMG signal <RENV1.FLTR><br>0: Enabled. Pulses shorter than 4 µs will be ignored.<br>1: Disabled. | [RENV1] (WRITE)<br>31 24<br>- - - - - - n - |
| Emergency stop command <CMEMG><br>The operation is the same as when an EMG signal is input. | [Command]<br>0005h |

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 may not recognize the glitch pulse, and therefore only the G9103C's internal counter may count command pulses (deviation from command position control). Therefore, after an emergency stop, you must perform an origin return to match a command position with a mechanical position.

## *8.10 Counter*

8.10.1 Counter type and input method

In addition to a positioning counter (RPLS), this LSI has three other counters. These counters offer the following functions.

- Control a command position by COUNTER 1.
- Control a mechanical position by COUNTER 2.
- Detect a stepper motor that is "out of step" using COUNTER 3.
- Output a synchronizing signal by COUNTER 3 and comparator 3.

RPLS register counts down every pulse output by reading an absolute value of target position (RMV) at the start even other than positioning operation. However, it does not count down until PCS signal turns ON while target position override 2 is executed (RMD.MPCS = 1).

Input to COUNTER 1 is exclusively for output pulses. However COUNTERS 2 to 3 can be selected as follows by setting RENV3 register.

| | COUNTER 1 (RCUN1) | COUNTER 2 (RCUN2) | COUNTER 3 (RCUN3) |
|---|---|---|---|
| Counter name | Command position | Mechanical position | General-purpose, deviation |
| Counter type | Count forward and backward counter | Count forward and backward counter | Count forward and backward counter Deviation counter |
| Number of bits | 28 | 28 | 16 |
| Command pulse | O | O | O |
| Encoder (EA, EB) signal input | | O | O |
| Pulse generator (PA and PB) signal input | | O | O |
| 1/4096 division clock of 40 MHz | | | O |

O---Counted. Space: Cannot be counted

Note: When using count to input pulsar signal, use an internal signal result after multiplying or dividing.

Input signal of COUNTER 2 <RENV3.CI2>

00: EA and EB input 01: Output pulses
10: PA and PB input 11 : Prohibited setting

[RENV3] (WRITE)

| 15 | | | | | | | 8 |
|---|---|---|---|---|---|---|---|
| - | - | - | - | - | - | n | n |

Input signal of COUNTER 3 <RENV3.CI3>

000: Command pulses
001: EA, EB signals
010: PA and PB signals
011: 1/4096 division clock of 40 MHz
100: Count the deviation between command pulses and EA and EB signals
101: Count the deviation between command pulses and PA and PB signals.
110: Count the deviation between EA and EB input and PA and PB signals.
111: Prohibited setting.

[RENV3] (WRITE)

| 15 | | | | | | | 8 |
|---|---|---|---|---|---|---|---|
| - | - | - | n | n | n | - | - |

There are two input type to be selected to input EA and EB, PA and PB signals by setting RENV2 registers.

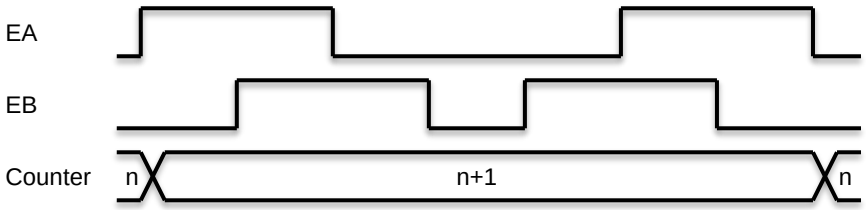
1) Signal input method: Input 90 degree phase difference signals (1x, 2x, and 4x).
   Counter direction: Count forward when EA input phase is leading. Count backward when EB input phase is leading.
2) Signal input method: Input count forward pulses or count backward pulses (Two-pulse input).
   Count direction: Count forward when EA signal is rising. Count backward when EB signal rising.

The counter direction or EA/EB and PA/PB input signals can be reversed.
When the EA and EB signal, or the PA and PB signal change simultaneously, the LSI regards it as an error. This error can be found in the REST register.

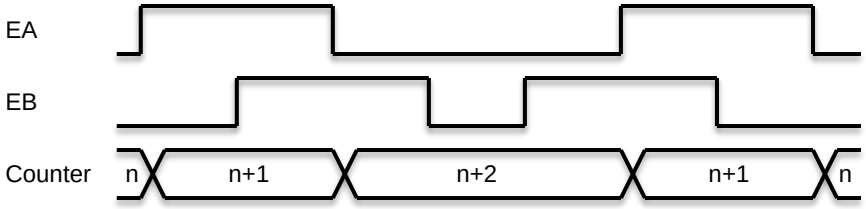
| | |
|---|---|
| Input filter function of EA, EB and EZ signals <RENV2.EINF><br>0: Enabled. Signals shorter than 150 ns are ignored.<br>1: Disabled. | [RENV2] (WRITE)<br>15 8<br>- - - - - - n - |
| Input type of EA and EB signal <RENV2.EIM><br>00: 90 degree phase difference, 1x 10: 90 degree phase difference, 4x<br>01: 90 degree phase difference, 2x 11: Count forward and backward pulses (Two-pulse input) | [RENV2] (WRITE)<br>15 8<br>- - - - - - n n |
| Function to reverse count direction of EA and EB signals <RENV2.EDIR><br>0: Disabled.<br>1: Enabled. | [RENV2] (WRITE)<br>15 8<br>- - - - n - - - |
| Input function of EA and EB signals. <RENV2.EOFF><br>0: Enable EA and EB input<br>1: Disable EA and EB input. (EZ input is enabled.) | [RENV2] (WRITE)<br>23 16<br>- - - - - - n - |
| Input filter function of PA and PB <RENV2.PINF><br>1: Enabled. Signals shorter than 150 ns are ignored.<br>0: Disabled. | [RENV2] (WRITE)<br>15 8<br>- - n - - - - - |
| Input type of PA and PB signals <RENV2.PIM><br>00: 90 degree phase difference, 1x 10: 90 degree phase difference, 4x<br>01: 90 degree phase difference, 2x 11: Count forward and backward pulses (Two-pulse input) | [RENV2] (WRITE)<br>15 8<br>n n - - - - - - |
| Function to reverse count direction of PA and PB signals. <RENV2.PDIR><br>0: Enabled.<br>1: Disabled. | [RENV2] (WRITE)<br>23 16<br>- - - - - - - n |
| Input function of PA and PB signals <RENV2.POFF><br>0: Enabled.<br>1: Disabled. | [RENV2] (WRITE)<br>23 16<br>- - - - - n - - |
| Error interrupt status <REST.ESEE><br>1: Occurrence of EA and EB signals input error. | [REST] (READ)<br>15 8<br>- - n - - - - - |
| Error interrupt status <REST.ESPE><br>1: Occurrence of PA and PB signals input error. | [REST] (READ)<br>15 8<br>- n - - - - - - |

When input of EA and EB signal is normal (RENV2.EDIR=0), count timing is as follows.
For details about PA and PB signals, see "6.3 Pulse generator (PA and PB) input mode."

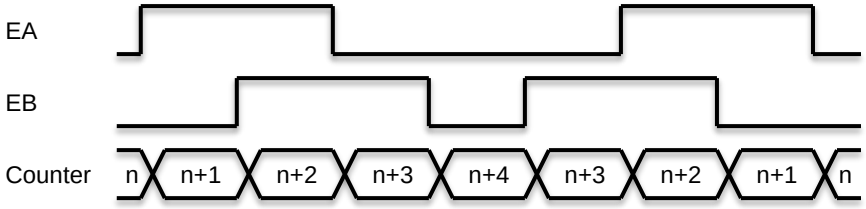
1) When using 90 degree phase difference signals 1x input



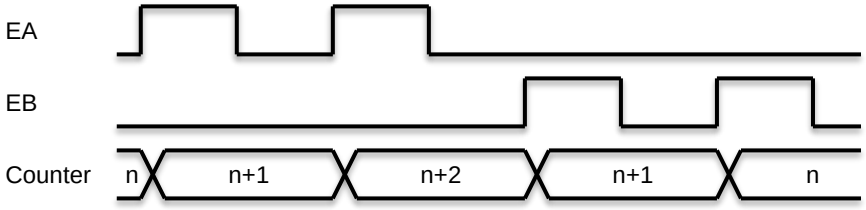

2) When using 90 degree phase difference signals 2x input




3) When using 90 degree phase difference signals 4x input




4) When two pulses are input (count on the rising edge)



8.10.2 Counter clear

All counters can be cleared using any of the following four methods.

1) When CLR signal turns ON (set to RENV3 register).
2) When an origin return completes in origin return operation (set to RENV3 register).
3) When writing a command to clear counter.
4) Just after counter value is latched (Set to RENV4 register).

CLR input timing can be set in RENV1 register. As an event interrupt cause, an interrupt can be generated when inputting CLR signal.

| | |
|---|---|
| Function to clear COUNTER 1 when CLR signal turns ON. <RENV3.CU1C><br>0: Disabled.<br>1: Enabled. | [RENV3] (WRITE)<br>23 16<br>0 - - - 0 - - n |
| Function to clear COUNTER 2 when CLR signal turns ON. <RENV3.CU2C><br>0: Disabled.<br>1: Enabled. | [RENV3] (WRITE)<br>23 16<br>0 - - - 0 - n - |
| Function to clear COUNTER 3 when CLR signal turns ON. <RENV3.CU3C><br>0: Disabled.<br>1: Enabled. | [RENV3] (WRITE)<br>23 16<br>0 - - - 0 n - - |
| Function to clear COUNTER 1 when reaching origin point in origin return operation. <RENV3.CU1R><br>0: Disabled.<br>1: Enabled. | [RENV3] (WRITE)<br>23 16<br>0 - - n 0 - - - |
| Function to clear COUNTER 2 when reaching origin point in origin return operation. <RENV3.CU2R><br>0: Disabled.<br>1: Enabled. | [RENV3] (WRITE)<br>23 16<br>0 - n - 0 - - - |
| Function to clear COUNTER 3 when reaching origin point in origin return operation. <RENV3.CU3R><br>0: Disabled.<br>1: Enabled. | [RENV3] (WRITE)<br>23 16<br>0 n - - 0 - - - |
| Function to clear COUNTER 1 just after counter 1 is latched. <RENV4.CU1L><br>0: Disabled.<br>1: Enabled. | [RENV4] (WRITE)<br>31 24<br>- - - n - - - - |
| Function to clear COUNTER 2 just after counter 2 is latched. <RENV4.CU2L><br>0: Disabled.<br>1: Enabled. | [RENV4] (WRITE)<br>31 24<br>- - n - - - - - |
| Function to clear COUNTER 3 just after counter 3 is latched. <RENV4.CU3L><br>0: Disabled.<br>1: Enabled. | [RENV4] (WRITE)<br>31 24<br>- n - - - - - - |
| Input type of CLR signal <RENV1.CLR><br>00: Clear on the falling edge 01: Clear on the rising edge<br>10: Clear on a L level 11: Clear on a H level | [RENV1] (WRITE)<br>23 16<br>- - n n - - - - |
| Input status of CLR signal <RSTS.SCLR><br>0: OFF<br>1: ON | [RSTS] (READ)<br>23 16<br>- - - - - - n - |
| Event interrupt cause <RIRQ.IRCL><br>1: When counter value is cleared by turning CLR signal ON. | [RIRQ] (WRITE)<br>15 8<br>- - - - - - - n |

| | |
|---|---|
| Event interrupt status <RIST.ISCL><br>1: When counter value is cleared by turning CLR signal ON. | [RIST] (WRITE)<br>15 8<br>- - - - - - - n |
| COUNTER 1 clear command < CUN1R><br>Clear COUNTER 1. | [Command]<br>0020h |
| COUNTER 2 clear command < CUN2R><br>Clear COUNTER 2. | [Command]<br>0021h |
| COUNTER 3 clear command < CUN3R><br>Clear COUNTER 3. | [Command]<br>0022h |

Note: When count timing and clear timing match, the counter will be set to 0.

8.10.3 Latch counter and count condition

All counters can latch their counts using any of the following five methods. The setting is made in RENV4 register. The latched values can be read from RLTC1 ~RLTC3 registers.

1) When LTC signal turns ON.
2) When ORG signal turns ON.
3) When a condition for Comparator 2 is satisfied.
4) When a condition for Comparator 3 is satisfied.
5) When writing a command.

The current speed can also be latched instead of COUNTER 3. Latch of (above Items 1) ~ 4)) can also be stopped by hardware timing.
LTC signal timing can be set by RENV1 register. An interrupt can be generated as the cause of an even interrupt when a counter value is latched by LTC signal or ORG signal turning ON.

| | |
|---|---|
| Latch timing of COUNTER 1 ~ COUNTER 3 <RENV4.LTM><br>00: When LTC signal turns ON.<br>01: When ORG signal turns ON.<br>10: When a condition for Comparator 2 is satisfied.<br>11: When a condition for Comparator 3 is satisfied | [RENV4] (WRITE)<br>31 24<br>- - - - - - n n |
| COUNTER 3 latch data <RENV4.LTFD><br>0: Latch RCUN3 register (COUNTER 3).<br>1: Latch RSPD.AS bit (current speed). | [RENV4] (WRITE)<br>31 24<br>- - - - - n - - |
| Latch function by turning LTC signal or ORG signal ON and a comparison condition is met. <RENV4.LTOF><br>0: Enabled.<br>1: Disabled. | [RENV] (WRITE)<br>31 24<br>- - - - n - - - |
| Input type of LTC signal. <RENV1.LTCL><br>0: Falling edge.<br>1: Rising edge. | [RENV1] (WRITE)<br>23 16<br>n - - - - - - - |
| Event interrupt cause (LTC signal) <RIRQ.IRLT ><br>1: When LTC signal turns ON. | [RIRQ] (WRITE)<br>15 8<br>- - - - - - n - |
| Event interrupt cause (ORG signal) <RIRQ.IROL><br>1: When ORG signal turns ON. | [RIRQ] (WRITE)<br>15 8<br>- - - - - - n - |
| Event interrupt status (LTC signal) <RIST.ISLT ><br>1: When LTC signal turns ON. | [RIST] (WRITE)<br>15 8<br>- - - - - n - - |
| Event interrupt status (ORG signal) <RIST.ISOL ><br>1: When ORG signal turns ON. | [RIST] (WRITE)<br>15 8<br>- - - - - n - - |
| Input status of LTC signal <RSTS.SLTC ><br>0: OFF.<br>1: ON. | [RSTS] (READ)<br>23 16<br>- - - - - n - - |
| Counter latch command <LTCH><br>Latch the contents of COUNTER 1 ~ COUNTER 3. | [Command]<br>0029h |

8.10.4 Stop the counter

COUNTER 1, COUNTER 2, and COUNTER 3 stop when RENV3 register can be set to stop.
COUNTER 1 stops while in timer mode operation.
By setting RENV3 register, you can stop counting pulses while performing a backlash correction.
COUNTER 3 can be set to count only during operation (BSY = L level) using RENV3 register. By specifying 1/4096 of 40 MHz signal, a time from a start can be controlled.

| | |
|---|---|
| Counting operation for COUNTER 1 <RENV3.CU1H><br>0: Enabled.<br>1: Disabled. | [RENV3] (WRITE)<br>31 24<br>- - - n - - - - |
| Counting operation for COUNTER 2 <RENV3.CU2H><br>0: Enabled.<br>1: Disabled. | [RENV3] (WRITE)<br>31 24<br>- - n - - - - - |
| Counting operation for COUNTER 3 <RENV3.CU3H><br>0: Enabled.<br>1: Disabled. | [RENV3] (WRITE)<br>31 24<br>- n - - - - - - |
| Counting operation for COUNTER 1 during backlash correction <RENV3.CU1B><br>0: Disabled.<br>1: Enabled. | [RENV4] (WRITE)<br>31 24<br>- - - - - - - n |
| Counting operation for COUNTER 2 during backlash correction <RENV3.CU2B><br>0: Disabled.<br>1: Enabled. | [RENV4] (WRITE)<br>31 24<br>- - - - - - n - |
| Counting operation for COUNTER 3 during backlash correction <RENV3.CU3B><br>0: Disabled.<br>1: Enabled. | [RENV4] (WRITE)<br>31 24<br>- - - - - n - - |
| Counting conditions for COUNTER 3 only during operation (BSY = L level) <RENV3.BSYC><br>0: Disabled.<br>1: Enabled. | [RENV3] (WRITE)<br>15 8<br>- - 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 ~ RCMP3 registers with counter values.
COUNTER 1 ~ COUNTER 3, RPLS register or RCIC register can be selected as comparison counters.
There are many comparison methods and 3 processing methods that can be used when a condition is met.
Specify the comparator conditions in RENV4 registers. By using these comparators, you can perform the following.

- Generate an interrupt and output a comparison result externally.
- Operation stops immediately or decelerates and stops.
- Same operation is executed as writing PRESHF (002B) command (used for changing speed in operating.)
- Software limit function using Comparators 1 and 2.
- Ring count function using COUNTER 1 and Comparator 1.
- Ring count function using COUNTER 2 and Comparator 2.
- Detect out of step stepper motors using COUNTER 3 and a comparator.
- Output a synchronizing signal (IDX) using COUNTER 3 and a Comparator 3.

[Comparison data] Each comparator can select comparison counter in the following table.

| Comparison counter | Comparator 1 | | Comparator 2 | | Comparator 3 | | |
|---|---|---|---|---|---|---|---|
| | | RENV4.C1C | | RENV4.C2C | | RENV5.C3C2 | RENV4.C3C |
| COUNTER 1 | O | 00 | O | 00 | O | * | "00" |
| COUNTER 2 | O | 01 | O | 01 | O | * | "01" |
| COUNTER 3 | O | 10 | O | 10 | O | * | "10" |
| Remaining pulse counter (RPLS) in other than interpolation mode<br>Remaining pulse counter (RCIC) in interpolation mode. | - | - | - | - | O | 1 | 11 |
| Special purpose | +SL | | -SL | | Out-of-step detection,<br>IDX signal output | | |

- O: Comparable. * No setting value required.
- +SL, -SL are used for software limits.
- If COUNTER 3 that was specified as deviation counter is selected as comparison counter, the LSI will compare between the absolute value and comparator data. (Absolute value range: 0 ~ 32,768)
- If remaining pulse counter is selected, PRLS register or RCIC register is selected automatically according to operation mode.
- Choose the comparison data by RENV4.C1C bit, RENV4.C2C bit, and RENV4.C3C bit and RENV5.C3C2 bit.

[Comparison conditions] Each comparator can be assigned a comparison conditions from the table below.

| Comparison conditions | Comparator 1 | | | Comparator 2 | | | Comparator 3 | |
|---|---|---|---|---|---|---|---|---|
| | | RENV4.C1S | RENV4.C1RM | | REBV4.C2S | RENV4.C2RM | | RENV4.C3S |
| 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 (synchronizing signal) output (regardless of counting direction) | - | - | - | - | - | - | O | 1000 |
| IDX (synchronizing signal) output (count forward only) | - | - | - | - | - | - | O | 1001 |
| IDX (synchronizing 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: Comparable. Blank: Comparison is impossible.
- Comparator 3 has a prohibited setting (RENV4.C3S = 0111b). Other selection does not satisfy comparison conditions.
- If Comparator 3 is set to IDX (synchronizing) signal output (RENV4.C3S = 1000b, 1001b and 1010b), select COUNTER 3 for use as a comparison counter. Other counters cannot be used for this function. Enter a positive value for the comparator setting.
- When using a comparator function as a software limit, Comparator 1 will be a positive limit value and the comparison condition is "Comparator < Comparison counter." Comparator 2 will be the negative limit and the comparison method is "Comparator > Comparison counter." Select COUNTER 1 as the comparison counter.
- RENV4.C1S bit, RENV4.C2S bit and RENV4.C3S bits are used to select operation.

Note. Remaining pulse counter (RCIC) in circular interpolation mode regards all values that are more than 134.217.727, as 134.217.727 and compares it with comparator 3 internally. Therefore, when you set the maximum value (134.217.727) to RCMP register and select comparator and comparator counter as a comparison condition, please note that this condition is always true in the case that RCIC register is more than 134.217.727.

[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 |
|---|---|---|---|
| | RENV4.C1D | RENV4.C2D | RENV4.C3D |
| No operation | 00 | 00 | 00 |
| Immediate stop operation | 01 | 01 | 01 |
| Deceleration stop operation | 10 | 10 | 10 |
| Same operation as writing PRESHF(002B) command | 11 | 11 | 11 |

- RENV4.C1D bit, RENV4.C2D bit and RENV4.C3Dbit are used to select operation.

| | | |
|---|---|---|
| Event interrupt cause (Comparator 1)<br>1: When a Comparator 1 condition is satisfied. | <RIRQ.IRC1> | [RIRQ] (WRITE)<br>7 0<br>- - n - - - - - |
| Event interrupt cause (Comparator 2)<br>1: When a Comparator 2 condition is satisfied. | <RIRQ.IRC2> | [RIRQ] (WRITE)<br>7 0<br>- n - - - - - - |
| Event interrupt cause (Comparator 3)<br>1: When a Comparator 3 condition is satisfied. | <RIRQ.IRC3> | [RIRQ] (WRITE)<br>7 0<br>n - - - - - - - |
| Event interrupt cause (Comparator 1)<br>1: When a Comparator 1 condition is satisfied. | <RIST.ISC1> | [RIST] (READ)<br>7 0<br>- - n - - - - - |
| Event interrupt cause (Comparator) 2<br>1: When a Comparator 2 condition is satisfied. | <RIST.ISC2> | [RIST] (READ)<br>7 0<br>- n - - - - - - |
| Event interrupt cause (Comparator 3)<br>1: When a Comparator 3 condition is satisfied. | <RIST.ISC3> | [RIST] (READ)<br>7 0<br>n - - - - - - - |
| Event interrupt cause (Comparator 1)<br>1: When a Comparator 1 condition is satisfied. | <RSTS.SCP1> | [RSTS] (READ)<br>7 0<br>- - - n - - - - |
| Event interrupt cause (Comparator 2)<br>1: When a Comparator 2 condition is satisfied. | <RSTS.SCP2> | [RSTS] (READ)<br>7 0<br>- - n - - - - - |
| Event interrupt cause (Comparator 3)<br>1: When a Comparator 3 condition is satisfied. | <RSTS.SCP3> | [RSTS] (READ)<br>7 0<br>- n - - - - - - |
| Event interrupt cause (Comparator 1)<br>1: When a Comparator 1 condition is satisfied. | <REST.ESC1> | [REST] (READ)<br>7 0<br>- - - - - - - n |
| Event interrupt cause (Comparator 2)<br>1: When a Comparator 2 condition is satisfied. | <REST.ESC2> | [REST] (READ)<br>7 0<br>- - - - - - n - |
| Event interrupt cause (Comparator 3)<br>1: When a Comparator 3 condition is satisfied. | <REST.ESC3> | [REST] (READ)<br>7 0<br>- - - - - n - - |

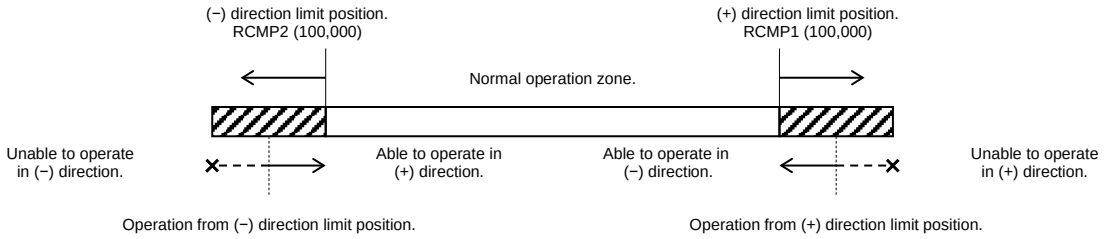
[Change speed using comparator]
If RENV4.Cnd bit is set to 11b, the same processing will be mode as writing a PRESHE (002B) command when a condition is satisfied. For example, any the data that a register value running currently and PRFH register value is different is set to the pre-register, FH speed can be changed when a condition is satisfied.

When changing register with computer, next operation data cannot be set because pre-register is used.

8.11.2 Software limit (SL) function

A software limit function can be set up using comparators 1 and comparator 2.
Select COUNTER 1 as a comparison counter for comparators 1 and comparator 2.
Use Comparator 1 as a (+) direction limit and Comparator 2 as a (-) direction limit to stop the motor 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 a comparator condition is met (RENV4.C1D, C2D), when movement on the axis reaches the software limit while in a high speed start, the motor decelerates and stops. When some other process is specified for use when a condition is met, or while in a constant speed start, the motor stops immediately.
If a software limit is ON while writing a start command, the motor does not start to move in the direction in which the software limit is ON. However, it can start in the opposite direction.

[Setting example]
RENV4 = 00003838h: Comparator 1 is used as a (+) direction software limit. Comparator 2 is used as a (−) direction software limit.
Set to stop immediately for both limit when the software limit is reached.
RCMP1 = 100,000: Positive direction limit value
RCMP2 = −100,000: Negative direction limit value




| | |
|---|---|
| Comparison condition for Comparator 1 <RENV4.C1S><br>110: Use as a positive direction software limit (RCMP1< COUNTER 1)<br>*This function does not use other than the above setting. | [RENV4] (WRITE)<br>7 0<br>- - - n n n - - |
| Process to use when a comparator 1 condition is met. <RENV4.C1D><br>01: Stop immediately.<br>10: Decelerate and stop<br>*This function does not use other than the above setting. | [RENV4] (WRITE)<br>7 0<br>- n n - - - - - |
| Comparison condition for Comparator 2 <RENV4.C2S><br>110: Use as a negative direction software limit. (RCMP2 > COUNTER 1)<br>*This function does not use other than the above setting. | [RENV4] (WRITE)<br>15 8<br>- - - n n n - - |
| Process to use when Comparator 2 conditions is met. <RENV4.C2D><br>01: Stop immediately<br>10: Decelerate and stop<br>*This function does not use other than the above setting. | [RENV4] (WRITE)<br>15 8<br>- n n - - - - - |

8.11.3 Out of step detection function for stepper motors

If the deviation counter value controlled by 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 and a comparator.
The process which takes place after an out of step condition is detected can be selected from the processing when comparator conditions are satisfied (RENV4.C1D, C2D, C3D).
For this function, use an encoder with the same resolution as the stepper motor.
COUNTER 3 can be cleared by writing CUN3R (0022h) command to the deviation counter.
Feedback signal that input to EA and EB terminals can be selected from 90 degree phase difference signals (1x, 2x, 4x) and Two-pulse input (a count forward pulse and a count backward pulse).
When both 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 deviation counter with EA and EB signals. |
| RENV4 = 00560000h: | Satisfy a condition of Comparator 3 < COUNTER 3<br>Stop immediately when a condition is met. |
| RCMP3 = 32: | The maximum deviation value is "32" pulses. |
| RIRQ = 00000080h: | Generate an interrupt when a comparator 3 condition is met. |

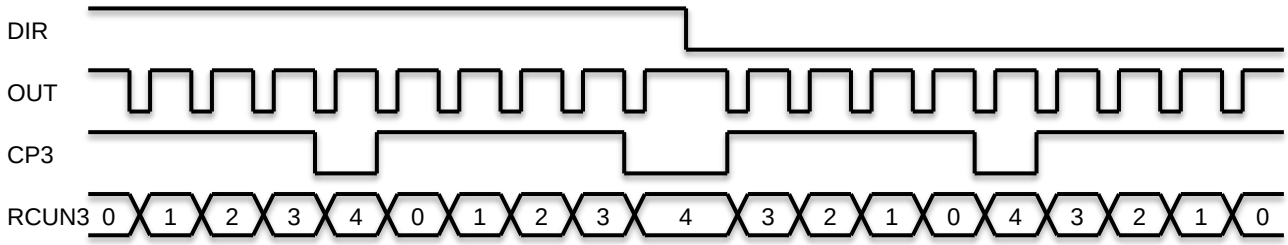
| | |
|---|---|
| Input signal of COUNTER 3 <RENV3.CI3><br>100: Count deviation using output pulses and EA and EB input<br>*This function does not use other than the above setting. | [RENV3] (WRITE)<br>15 8<br>- - - n n n - - |
| Input type of EA and EB signals <RENV2.EIM><br>00: 90 degree phase difference, 1x<br>01: 90 degree phase difference, 2x<br>10: 90 degree phase difference, 4x<br>11: Count forward pulse and count backward pulse (Two-pulse input) | [RENV2] (WRITE)<br>15 8<br>- - - - - n n - |
| Function to reverse count direction of EA and EB signals <RENV2.EDIR><br>0: Enabled.<br>1: Disabled. | [RENV2] (WRITE)<br>15 8<br>- - - - n - - - |
| Error interrupt status <REST.ESEE><br>1: When EA and EB signals change simultaneously. (Does not stop) | [REST] (READ)<br>15 8<br>- - n - - - - - |
| Input filter function of EA, EB and EA signals. <RENV2.EINF><br>0: Enabled. Pulse shorter than 150 ns will be ignored.<br>1: Disabled. | [RENV2] (WRITE)<br>15 8<br>- - - - - - - n |
| Function to input EA and EB signals. <RENV2.EOFF><br>0: Enabled<br>1: Disabled (EA signal input is enabled.) | [RENV2] (WRITE)<br>23 16<br>- - - - - - n - |
| COUNTER 3 clear <CUN3R><br>Clear COUNTER 3. | [Command]<br>0022h |

8.11.4 IDX (synchronizing) signal output function

Using Comparator 3 (RCMP3) and COUNTER 3 (RCUN3), the LSI can output IDX (synchronizing) signals to CP3 terminal at specified intervals.
If IDX (index) operation that output IDX (synchronizing) signal from CP3 terminal is used, select counter 3 as comparison counter (RENV3.C3C=10b) and IDX (synchronizing) signal output as a comparison condition (RENV4.C3S=1000b, 1001b, 1010b).
The counter range of COUNTER 3 will be 0 ~ the value set in RCMP3 register (MAX: 32,767). If counter value of COUNTER 3 counts forward from the value set in RCMP3 and if it counts backward from 0, the next counter value will be the value set in RCMP3.
The input for COUNTER 3 can be select by RENV3.CI3 bit.

| | |
|---|---|
| Set COUNTER 3 <RENV3.CI3><br>000: Command pulse<br>001: EA and EB signals<br>010: PA and PB signals<br>011: 1/4096 division pulse of 40 MHz<br>*This function does not use other than the above setting. | [RENV3] (WRITE)<br>15 8<br>- - - n n n - - |
| Comparison counter of Comparator 3 < RENV4.C3C><br>10: COUNTER 3<br>*This function does not use other than the above setting. | [RENV4] (WRITE)<br>23 16<br>- - - - - - n n |
| Comparison condition of Comparator 3 <RENV4.C3S><br>1000: Use as an IDX (synchronizing) signal output (regardless of count direction)<br>1001: Use as an IDX (synchronizing) signal output (while counting forward)<br>1010: Use as an IDX (synchronizing) signal output (while counting backward)<br>*This function does not use other than the above setting. | [RENV4] (WRITE)<br>23 16<br>- - n n n n - - |

[Example of setting]
RENV3 = 00000000h : Input command pulses to COUNTER 3.
RENV4 = 00220000h : Select COUNTER 3 as a comparison counter and regardless of count direction as count direction for IDX (synchronizing) signal output.
RCMP3 = 4 : Count range is 4.



8.11.5 Ring counter function

Ring counter is a count forward or backward counter to repeat from “0 ~ (setting value)”. When a count value is “the setting value” and it counts forward, it will be 0. When the count value is 0 and it counts backward, it will be the setting value. Mainly it can be used to control a current angle of a rotation table.

Using comparator 1 (RCMP1), COUNTER 1 (RCUN1) can be made as a ring counter, and using comparator 2(RCMP2), Counter 2 (RCUN2) can be made as a ring counter.

When making COUNTER 1 a ring counter, select RENV4.C1C = 00b, RENV4.C1S = 000b, RENV4.C1RM = 1 and set a “setting value” in RCMP1 register. (positive only)

When making COUNTER 2 a ring counter, select RENV4.C2C = 00b, RENV4.C2S = 000b, RENV4.C2RM = 1 and set a “setting value” in RCMP2 register. (positive only)

The counter value used for a ring counter should be within the range from 0 ~ “setting value”. If the counter value is out of the range, use the counter after writing a counter value within the range to RCUN1 register or RCUN2 register.

Even if a setting value for RMV register in the positioning mode is out of the above range, a motor operates. For example, when a rotation table of the 3600 pulses per rotation is operated with setting RCMP1 = 3599 and RMV = 7200, the table rotates two times and the counter 1 after stopping will be the same value before starting.

| | |
|---|---|
| Count specification of COUNTER 1 <RENV4.C1RM><br>1: Ring count operation.<br>*This function does not use other than the above setting. | [RENV4] (WRITE)<br>7 0<br>n - - - - - - - |
| Comparison counter for Comparator 1 <RENV4.C1C><br>00: COUNTER 1.<br>*This function does not use other than the above setting. | [RENV4] (WRITE)<br>7 0<br>- - - - - - 0 0 |
| Comparison conditions for comparator 1 <RENV4.C1S><br>00: Comparison conditions are not satisfied at any time.<br>*This function does not use other than the above setting. | [RENV4] (WRITE)<br>7 0<br>- - - n n n - - |
| Processing when comparison a condition of Comparator 1 is not met. <RENV4.C1D><br>No processing.<br>*This function does not use other than the above setting. | [RENV4] (WRITE)<br>7 0<br>- n n - - - - - |
| Count specification of COUNTER 2 <RENV4.C2RM><br>1: Ring count operation<br>*This function dos not use other than the above setting. | [RENV4] (WRITE)<br>15 8<br>n - - - - - - - |
| Comparison counter for Comparator 2 <RENV4.C2C><br>00: COUNTER 2.<br>*This function does not use other than the above setting. | [RENV4] (WRITE)<br>15 8<br>- - - - - - n n |
| Comparison conditions for comparator 2. <RENV4.C2S><br>000: A comparison condition is not met at any time.<br>*This function does not use other than the above setting. | [RENV4] (WRITE)<br>15 8<br>- - - n n n - - |
| Processing when comparison a condition of Comparator 2 is not met. <RENV4.C2D><br>No processing.<br>*This function does not use other than the above setting. | [RENV4] (WRITE)<br>15 8<br>- n n - - - - - |

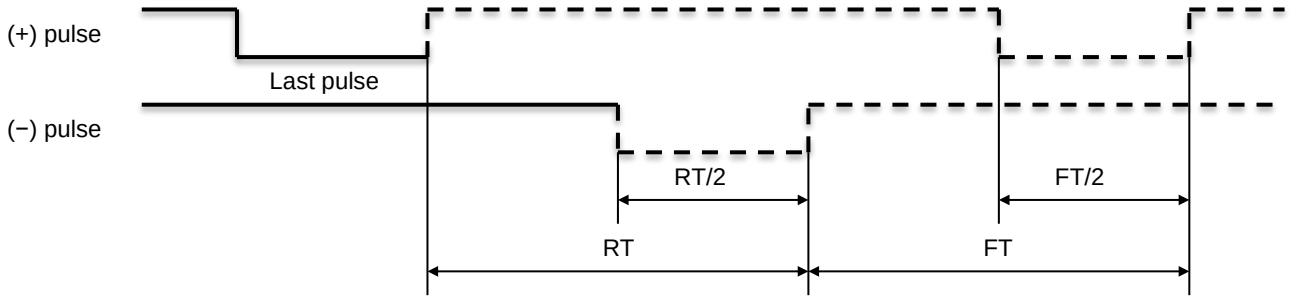
## *8.12 Backlash correction*

This LSI has a backlash correction functions. This function outputs the number of command pulses that are specified for backlash correction amount (RENV5.BR) in correction speed (RFA) just before start
The backlash correction is performed just before start each time the direction of operation changes. Number of pulses of backlash correction amount and selection to enable or disable backlash correction are specified in RENV5 register.
Counters (COUNTER 1 ~ COUNTER 3) can be operated and checked using RENV3 register even during backlash correction.

| | | |
|---|---|---|
| Backlash correction amount<br><br>Set backlash correction amount<br>[Setting range : 0 ~ 4,095] | <RENV5.BR> | [RENV5] (WRITE)<br>15 8<br>- - - - n n n n<br>7 0<br>n n n n n n n n |
| Backlash correction function<br>0: Disabled.<br>1: Enabled. | <RENV5.ADJ> | [RENV5] (WRITE)<br>15 8<br>- - - n - - - - |
| Count function of COUNTER 1 during backlash correction<br>0: Disabled.<br>1: Enabled. | <RENV3.CU1> | [RENV3] (WRITE)<br>31 24<br>- - - - - - - n |
| Count function of COUNTER 2 during backlash correction<br>0: Disabled.<br>1: Enabled. | <RENV3.CU1> | [RENV3] (WRITE)<br>31 24<br>- - - - - - n - |
| Count function of COUNTER 3 during backlash correction<br>0: Disabled.<br>1: Enabled. | <RENV3.CU1> | [RENV3] (WRITE)<br>31 24<br>- - - - - n - - |

## *8.13 Vibration restriction function*

This LSI has a function to restrict vibration occurs when a stepping motor is stopping by adding one pulse of reverse operation and one pulse of forward operation shortly after completing a command operation.
Specify output timing for additional pulses in RENV6 register.
When both reverse timing (RT) and forward timing (FT) are not 0, vibration restriction function is enabled.
The dotted lines below are pulses added by vibration restriction function. (An example in (+) direction)




| | |
|---|---|
| Reverse operation timing <RENV6.RT><br>Reverse time (RT) range: 0 ~ 65,535<br>Setting unit: 1.6 µs.<br>Settable range: 0 ~ approx. 0.1 s. | [RENV6] (WRITE)<br>15 8<br>n n n n n n n n<br>7 0<br>n n n n n n n n |
| Forward operation timing <RENV6.FT><br>Forward time (FT) range : 0 ~ 65,535<br>Setting unit: 1.6 µs.<br>Settable range: 0 ~ approx. 0.1 s. | [RENV6] (WRITE)<br>31 24<br>n n n n n n n n<br>23 16<br>n n n n n n n n |

Note 1: The optimum values for reverse time (RT) and forward time (FT) varies according to each piece of machinery and load. Therefore, obtain these values by experiments.
Note 2: When vibration restriction function is enabled, the timing to complete operation will be “when pulse width is complete” even if selecting “when pulse cycle is complete” (RMD.METM = 0).

## *8.14 Excitation sequence for stepper motors*

This LSI can generate excitation sequences of unipolar and bipolar driving for 2-phase stepper motors in 2-phase or 1-2 phase excitation method.
Use BSY/PH1, FUP/PH2, FDW/PH3, and MVC / PH4 terminals to output these signal sequences. To switch output signals, set RMD.MPH bit. (Rotation can be reversed by RENV1.MREV = 1).
When output of BSY/PH1, FUP/PH2, FDW/PH3, MVC/OH4 terminals are set to PH1 ~ PH4 signals, output can be masked using RMD.MMPH bit.
Output level of PH1 ~ PH4 when masked can be selected from “LLLL” or “LLHH” using RMD.MBIM bit.
To switch between unipolar and bipolar signals, set RMD.MUB bit. To switch between 2-phase and 1-2 phase excitations, set RMD.MFH bit.
While the LSI is producing an excitation signal for a single phase in 1-2 phase excitation (steps 1, 3, 5, and 7 in Excitation sequence for unipolar driving method table below), if you switch to 2 phase excitation, the LSI will change to 2-phase excitation status starting with a next output pulse.
By reading RSTS register, you can check excitation sequence status.

[Excitation sequence for Unipolar driving method]

| 2-phase excitation method | | | | | |
|---|---|---|---|---|---|
| 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 → (+) | | | | | |

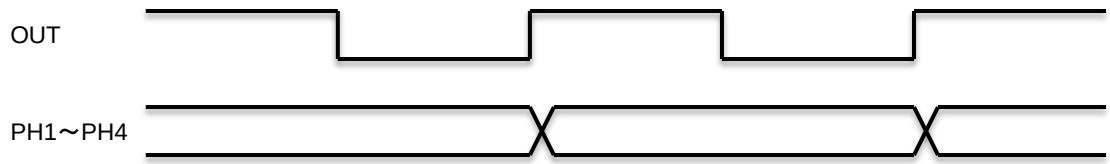
| 1-2 phase excitation method | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| STEP | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 0 |
| PH1 | H | H | H | L | L | L | L | L | H |
| PH2 | L | L | H | H | H | L | L | L | L |
| PH3 | L | L | L | L | H | H | H | L | L |
| PH4 | H | L | L | L | L | L | H | H | H |
| (−) ← Operation direction → (+) | | | | | | | | | |

[Excitation sequence for Bipolar driving method]

| 2-phase excitation method | | | | | |
|---|---|---|---|---|---|
| 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 method | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 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 → (+) | | | | | | | | | |

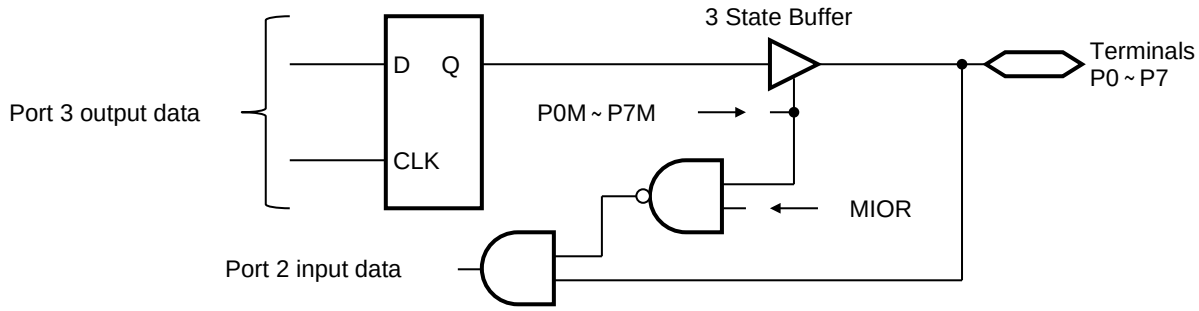
[The change timing of excitation sequence]



| | |
|---|---|
| Output signals of BSY/PH1, FUP/PH2, FDW/PH3, MVC/PH4 terminals <RMD.MPH><br>0: Output BSY, FUP, FDW, and MVC signals.<br>1: Output PH1, PH2, PH3, and PH4 signals. | [RMD] (WRITE)<br>23 16<br>\| \| - \| - \| n \| - \| - \| - \| - \| |
| Function to mask output of PH1, PH2, PH3, and PH4 signals <RMD.MMPH><br>0: Enabled. Output level specified in RMD.MBIM bit.<br>1: Disabled. Output PH1, PH2, PH3, and PH4 signals. | [RMD] (WRITE)<br>23 16<br>\| - \| - \| n \| - \| - \| - \| - \| - \| |
| Output masked status of PH1, PH2, PH3, and PH4 signals <RMD.MBIM><br>0: Output “LLLL” level.<br>1: Output “LLHH” level. | [RMD] (WRITE)<br>31 24<br>\| - \| - \| - \| - \| - \| - \| n \| - \| |
| Driving method using PH1, PH2, PH3, and PH4 signals <RMD.MUB><br>0: Excitation sequence for 2-phase unipolar<br>1: Excitation sequence for 2-phase bipolar | [RMD] (WRITE)<br>23 16<br>\| - \| n \| - \| - \| - \| - \| - \| - \| |
| Excitation method using PH1, PH2, PH3, and PH4 signals <RMD.MFH><br>0: 2 phase excitation method<br>1: 1-2 phase excitation method | [RMD] (WRITE)<br>23 16<br>\| n \| - \| - \| - \| - \| - \| - \| - \| |
| Output status of PH1 signal. <RSTS.SPH1><br>0: OFF. (L level)<br>1: ON. (H level) | [RSTS] (READ)<br>31 24<br>\| - \| - \| - \| - \| - \| - \| - \| n \| |
| Output status of PH2 signal. <RSTS.SPH2><br>0: OFF. (L level)<br>1: ON. (H level) | [RSTS] (READ)<br>31 24<br>\| - \| - \| - \| - \| - \| - \| n \| - \| |
| Output status of PH3 signal. <RSTS.SPH3><br>0: OFF. (L level)<br>1: ON. (H level) | [RSTS] (READ)<br>31 24<br>\| - \| - \| - \| - \| - \| n \| - \| - \| |
| Output status of PH4 signal. <RSTS.SPH4><br>0: OFF. (L level)<br>1: ON. (H level) | [RSTS] (READ)<br>31 24<br>\| - \| - \| - \| - \| n \| - \| - \| - \| |

## *8.15 General-purpose I/O terminals (P0 ~ P7)*

Although general purpose I/O terminals (P0 ~ P7) are set as input terminals by default, by setting RENV2.PnM bit (n = 0~7), they can be set individually for input or output every bit.
The internal arrangement of these terminals is roughly as shown below. Even when they are used 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 default status of latched output in the above figure is L level.)

When they are set for use as output terminals, the status is changed by writing data to Port 3 of the I/O port.
The terminals whose corresponding bits are set to "1" will go H level by the data written into Port 3.
The terminal status can be checked by reading Port 2.
The status data can be masked when is set as output terminals using RMD.MIOR bit.
The center LSI can use Port 2 for input change interrupt.
When reading of general-purpose output terminals is enabled (RMD.MIOR = 0), an input change interrupt occurs when output terminal changes.
When reading of general-purpose output terminals is disabled (RMD.MIOR = 1), an input change interrupt does not occur when output terminal changes (because port 2 does not change).

| | |
|---|---|
| Function of general purpose I/O terminals < RENV2.P7M ~ P0M><br>0: General purpose input<br>1: General purpose output | [RENV2] (WRITE)<br>7 0<br>n n n n n n n n |
| Monitor function for the output setting bits in general-purpose I/O terminals. <RMD.MIOR><br>0: Enabled.<br>1: Disabled. Regardless of the setting of the output terminal, the bits corresponding to Port 2 will be "0." | [RMD] (WRITE)<br>31 24<br>- - - - - - - n |
| Set the general-purpose I/O terminal data to be output <I/O PORT3.IOPOB><br>0: L level (when specified as output port)<br>1: H level (when specified as output port) | [PORT3] (WRITE)<br>7 0<br>n n n n n n n n |
| Set the general-purpose I/O terminal data to be output <I/O PORT3.IOPIB><br>0: L level<br>1: H level | [PORT2] (WRITE)<br>7 0<br>n n n n n n n n |

## *8.16 Interrupt output*

This LSI can output an interrupt request signal from the center LSI. There are 22 types of errors, 17 types of events, and interrupt requests by the change from operating to stop. Each error interrupt cause will always generates an interrupt request without conditions, and each event cause can be selected in RIRQ register. Operating stop interrupt will generates an interrupt without conditions; however, interrupt request signal (MSTS.SINT) can be masked.

If any of interrupt by error occur, MSTS.SERR = 1.
If any of interrupts by event occur, MSTS.SEVT = 1.
If a stop interrupt occurs, MSTS.SEND = 1.
Normally, when any of MSTS.SERR, SEVT, SEND bit is 1, interrupt request occurs and MSTS.SINT = 1.

[Cautions]
When MSTS.SINT bit of G9103C turns from 0 to 1, an interrupt request signal can be output from the center LSI. However, several interrupt causes of MSTS.SERR, SEVT, and SEND bit occur with time lag, MSTS.SINT may keep 1 without change and an interrupt request signal is not output from the center LSI. Therefore, when either MSTS.SERR, SENT or SEND is reset, 0 is returned to the center LSI instead of MSTS.SINT bit value for up to one cycle of cyclic communication cycle and that causes a change from 0 to 1 at the next cyclic communication.
Because of that, when several interrupt requests occur almost simultaneously, interrupt request signals on the G9001A side may occur several times with time lag of up to one cycle of cyclic communication.

[Interrupt sequence]
G9103C's main status (MSTS) is sent (revised) to the address that corresponds to G9103C that is in port data area of the center LSI, as input data for port 0 and 1 in G9103C in I/O 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, G9103C can issue an interrupt to CPU.

If you set a port 0 change interrupt on a local LSI for data control that includes G9103C, it output an interrupt request signal and only when bit 0 of port 0 changes from 0 to 1 and it will ignore the change of bit 1 ~ 7.

[Error interrupt]
You can confirm about error interrupt cause by REST register.
If any bit of REST register is 1, MSTS.SERR = 1.

To reset REST register, there are two the following ways.

1. When function to reset interrupt cause automatically is enabled (RENV4.ISMR = 0), all bits are cleared by reading REST register.
2. Regardless of a function to reset interrupt automatically (RENV4.ISMR), when you write a data that you make bit you want to clear 1 to REST register, the specified bit is cleared.

When a function to reset interrupt automatically is enabled (RENV4.ISMR = 0), if communication error occur at the communication for reading REST register, data with all bit 0 is transferred because the register is read again by retry. Therefore, we recommend that you set a function to reset interrupt automatically is disabled (RENV4.ISMR = 1) and use the above 2 method to clear.

[Event interrupt]
Event interrupt is enabled by a cause specified in RIRQ register. You can confirm an event interrupt cause, and the corresponding bit goes 1.
When any bit of RIST register is 1, MSTS.SEVT = 1.

To reset the RIST register, there are two the following ways.

1. When function to reset interrupt cause automatically is enabled (RENV4.ISMR = 0), all bits are reset by reading RIST register.
2. Regardless of the setting to reset interrupt causes, if you write a data that you make bit to clear 1 to RIST register, the specified bit is reset.

When function to reset interrupt cause automatically is disabled (RENV4.ISMR = 1), if communication error occur at the communication for reading RIST register, data with all bits 0 is transferred because the register is read again by retry. Therefore, we recommend that you set a function to reset an interrupt cause

automatically is disabled (RENV4.ISMR = 1) and use the above 2 to reset.

[Operation stop interrupt]
An operation stop interrupt (MSTS.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 check process to read RIST register is necessary as described in the event interrupt cause section. If your system needs to provide an operation stop interrupt only when a stop occurs, use operation stop interrupt function.

The operation stop interrupt is reset by writing INTRS (0008h) command.

By setting RENV1.SEDM bit, you can choose not to reflect occurrence of MSTS.SEND bit operation stop interrupt to MSTS.SIND bit or to reset MSTS. SEND bit when writing a start command by setting RENV1.SEDR bit.
If a function to reset operation stop interrupt automatically is enabled (RENV1.SEDR = 1), you do not have to write INTRS (0008h) command before writing a start command. However, If you write a start command at the timing when operation stop interrupt turns ON (MSTS.SEND = 1), operation stop interrupt may turn OFF (MSTS.SEND = 0) before interrupt request signal is checked. Please note that if you use pre-registers in operation.
When a function to generate operation stop interrupt while pre-registers for operation is fixed is disabled (RMD.MENI = 1) and you set data for next operation, operation stop interrupt does not turn ON (MSTS.SEND = 1) even if the current operation completes.

[Interrupt control on the center LSI]
As you can see, only one G9103C have 40 (max) types of interrupt causes. The more axes there are, the more interrupt causes the whole system has.
Additionally, in interrupt cause of the center LSI, there are other interrupt causes as follows, in addition to Input change interrupt that is shown in the section [input sequence].

1. When DATA transmission FIFO is available for writing. (CEND)
2. When receiving 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 a local side has an error to receive data. (ERAE)
6. When an error occur in accessing to CPU. (CAER)

While an interrupt is handled, 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 a way that does not use interrupt processing.

[Evaluation process of interrupt cause]
The causes of an interrupt can be evaluated as follows below.
1) Read the main status and check whether either bit 1, 2, or 3 is "1."
2) If bit 1 (SEND) = "1", an operation stop interrupt occurs. Reset using INTRS (0008h) command.
3) If bit 2 (SERR) = "1", read the REST register to identify the cause of the error interrupt.
4) If bit 3 (SEVT) = "1", read the RIST register to identify the cause of the even interrupt.

Note 1: The center LSI outputs an interrupt request signal for CPU through INT terminal at the timing when G9001A performs I/O communication (cyclic communication) to the local LSI that requested the interrupt.

Therefore, even while CPU is writing normal sending data to FIFO in the center LSI or it is reading data from reception FIFO, an interrupt request may occur.

Using an 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. In the main routine, check the condition of the flag and perform data communication with the local LSI that requested interrupt.

Note 2: While processing in the step 4) above, it is possible that MSTS.SEND or SERR, SEVT bit may change to 1. After the steps 1) ~ 4) complete, please check whether all MSTS.SEND, SERR, SEVT bit are 0 before finishing the interrupt routine.

Note 3: When several G9103Cs 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 check whether interrupts on all G9103Cs do not occur.

Note 4: Information of port 0 ~ 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 G9103C.

The main status bit 0 (MSTS.SINT) can be masked by a function to mask interrupt request signal (RMD.MINT).
When a function to mask interrupt request signal is masked (RMD.MINT = 1), even though the status changes, MSTS.SINT bit will remain "0" and will not change to "1" when interrupt conditions are satisfied.
When a function to mask interrupt request signal is disabled (RMD.MINT=0), while interrupt condition are met, MSTS.SINT bit will change to 1.

| | |
|---|---|
| Interrupt request (MSTS.SINT) <MSTS.SINT><br>0: OFF. When SEND = 0 and or SERR = 0 and SEVT = 0.<br>1: ON. When SEND = 1 or SERR = 1 or SEVT = 1. | [MSTS] (READ)<br>7 0<br>- - - - - - - n |
| Operation stop interrupt (MSTS.SEND) <MSTS.SEND><br>0: OFF. Operation stop interrupt does not occur.<br>1: ON. Operation stop interrupt occurs.<br>Become 0 by writing INTRS (0008h) command. | [MSTS] (READ)<br>7 0<br>- - - - - - n - |
| Error interrupt (MSTS.SERR) <MSTS.SERR><br>0: OFF. Error interrupt does not occur.<br>1: ON. Error interrupt occurs.<br>Become 0 by reading or writing of REST register. | [MSTS] (READ)<br>7 0<br>- - - - - n - - |
| Event interrupt (MSTS.SEVT) <MSTS.SEVT><br>0: OFF. Event interrupt does not occur.<br>1: ON. Event interrupt occurs.<br>Become 0 by reading or writing of RIST register. | [MSTS] (READ)<br>7 0<br>- - - - n - - - |
| Interrupt request function <RMD.MINT><br>0: Enabled.<br>1: Disabled. Interrupt request will be fixed 0. | PRMD] (WRITE)<br>23 16<br>- - - - n - - - |
| Interrupt request ON when operation stop interrupt turns ON <RENV1.SEDM><br>0: Enable. Interrupt request turns ON when operation stop interrupt turns ON.<br>1: Disable. Interrupt request does not turn ON when operation stop interrupt turns ON. | [RENV1] (WRITE)<br>31 24<br>- - - - n - - - |
| Function to generate operation stop interrupt while pre-register for operation is fixed. <RMD.MENI><br>0: Enabled.<br>1: Disabled. Operation stop interrupt does not turn ON while pre-register for operation is fixed. | [RMD] (WRITE)<br>31 24<br>n - - - - - - - |
| Reset operation stop interrupt <INTRS><br>Operation stop interrupt OFF (MSTS.SEND = 0). | [Command]<br>0008h |
| Error interrupt status <RREST><br>Read REST register. | [Command]<br>00F2h |
| Event interrupt status. <PRIST><br>Read RIST register. | [Command]<br>00F3h |

| | |
|---|---|
| Set event interrupt cause <WRIRQ><br>Enable the bits set to 1 by writing RIRQ register. | [Command]<br>00ACh |
| Reset error interrupt status <WREST><br>Make the bits set to 1 by writing REST register to 0. | [Command]<br>00ADh |
| Reset event interrupt status <WRIST><br>Make the bits set to 1 by writing RIST register to 0. | [Command]<br>00AEh |

[Error interrupt causes] <Detail of REST register: The cause of an interrupt makes the corresponding bit "1">

| Error interrupt cause | Cause (REST) | |
|---|---|---|
| | Bit | Bit name |
| Stopped by a Comparator 1 condition being satisfied. (+SL) | 0 | ESC1 |
| Stopped by a Comparator 2 condition being satisfied. (−SL) | 1 | ESC2 |
| Stopped by a Comparator 3 condition being satisfied. | 2 | ESC3 |
| Stopped by +EL signal turning ON. | 3 | ESPL |
| Stopped by −EL signal turning ON | 4 | ESML |
| Stopped by ALM signal turning ON | 5 | ESAL |
| Stopped by STP signal turning ON | 6 | ESSP |
| Stopped by turning EMG signal ON | 7 | ESEM |
| Decelerated and stopped by turning SD signal ON | 8 | ESSD |
| When an overflow occurrence of PA and PB signals input buffer counter | 9 | ESPO |
| When stopped by error by watch dog timer. (Communication line disconnects etc.) | 10 | ESNT |
| (Always 0) | 11 | Not defined |
| When writing position override command (WRMVOR) while a motor is stopping. | 12 | ESOR |
| Both EA and EB signals change simultaneously. (A motor does not stop.) | 13 | ESEE |
| Both PA and PB signals change simultaneously. (A motor does not stop.) | 14 | ESPE |
| Stopped simultaneously by data error for interpolation operation Note 1 | 15 | ESDT |
| Stopped simultaneously by error stop of other axes | 16 | ESIP |
| Stopped by beyond the circular interpolation operation range (28 bits) while | 17 | ESAO |
| Synchronizing of the clock for motor control is collapsed. | 18 | EFAJ |
| When stopped by communication errors that occur continuously specified times for receiving clock synchronizing communication. | 19 | ECKM |
| When stopped by communication errors that occur continuously specified times for receiving synchronizing stop communication | 20 | ESPM |
| When stopped by communication errors that occur continuously specified times for receiving sensor substitute communication | 21 | ESWM |
| When writing a start command for dedicated next operation (NSTAFL, NSTAFH, NSTAD, and NSTAUD). | 22 | ENST |

Note1: When start command is written with the following data setting, error interrupt occurs.
When linear interpolation: (RMV = 0) and (PMVY = 0)
When circular interpolation ((RIP = 0) and (RIPY = 0)) or ((RMV = RIP) and (RMVY = RIPY))

[Event interrupt causes] < Detail of RIST register: 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 |
| Normal 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 a Comparator 1 condition is satisfied | 5 | IRC1 | 5 | ISC1 |
| When a Comparator 2 condition is satisfied | 6 | IRC2 | 6 | ISC2 |
| When a Comparator 3 condition is satisfied | 7 | IRC3 | 7 | ISC3 |
| When a counter value is reset by CLR signal turning ON | 8 | IRCL | 8 | ISCL |
| When a counter value is latched by LTC signal turning ON | 9 | IRLT | 9 | ISLT |
| When a counter value is latched by ORG signal turning ON | 10 | IROL | 10 | ISOL |
| When SD signal turns ON | 11 | IRSD | 11 | ISSD |
| When STA signal turns ON | 12 | IRSA | 12 | ISSA |
| When operation starts in receiving a start command (2x01h) for broadcast communication. | 13 | IRNA | 13 | ISNA |
| When stopping by receiving a stop command (2x02h) for broadcast communication | 14 | IRNP | 14 | ISNP |
| When it is available for writing into the pre-register for operation (Change to MSTS.SPRF = 0) | 15 | IRNM | 15 | ISNM |
| When pre-register for operation is not fixed and current operation complete (Change to RSTS.PFM = 00b) | 16 | IRBE | 16 | ISBE |

## *8.17 Synchronizing function with other axes*

There are two functions as the synchronizing function with other G9103C as follows.

1. Clock synchronizing function for motor control
2. Simultaneous stop when stop by errors.

There is one function as synchronizing function with an I/O device.

1. Substitute input as for sensor signals.

### 8.17.1 Clock synchronizing function for motor control

Output pulse train is generated to be synchronized with reference clock (40MHz or 80 MHz). Because G9103C is a LSI to control for single axis, use more than one G9103C when interpolation is executed. Normally each crystal oscillator provides reference clock to each G9103C independently.

In order to operate synchronizing such as interpolation, note that frequency error of this reference clock.

For example, if accuracy for two reference clock rate (frequency) is +100 ppm and −100 ppm, accuracy of output pulse frequency from G9103C is also +100 ppm (+0.01%) and –100 ppm (−0.01%).
Therefore, the number of the pulse output for 5 seconds at 10 kpps constant speed is 50005 pulses and 49995 pulses and the error of interpolation trajectory become large.
In order to reduce this error, G9103C has clock synchronizing function for motor control. When using several G9103Cs are connected, use any one G9103C as a master device for clock synchronizing for motor control and other G9103Cs can be synchronized with the clock for motor control of the master LSI.

When device number 4 is set to a clock master LSI.

| | No.1 | No.2 | No.3 | No.4 | No.5 |
|---|---|---|---|---|---|
| Device number | 1(01h) | 4(04h) | 6(06h) | 10(0Ah) | 20(14h) |
| RSYN.SYNC | 1 | 1 | 1 | 1 | 1 |
| RSYN.DNMST | 04h | 04h | 04h | 04h | 04h |
| Setting order | (2) | (1) | (3) | (4) | (5) |

Setting process

1. Set RSYN register of a clock master LSI
   RSYN.DNMST ← a device number of a clock master LSI (04h)
   RSYN.SYNC ← 1
2. Set RSYN registers of other G9103Cs to the same as a clock master LSI.
3. It takes approximately 4 ms until synchronizing control starts (RSYN.SYON = 1). Make sure to check synchronizing control start (RSYN.SYON = 1).

Note 1: Be sure to set RSYN register of clock master LSI first. For other G9103Cs, any order is available.
Note 2: Synchronizing control synchronizes only control the clock for motor control. Clock for communication control is not synchronized and is influenced by the frequency error of CLK signal. Therefore, use crystal oscillators whose permissible frequency deviation is within ±500ppm.
Note 3: Operation timing with output pulse width may fluctuate ±25ns by synchronizing control.

[Monitor of status to synchronize clock for motor control]

The reference clock frequency that is input to G9103C is 40 MHz or 80 MHz. The motor control circuit operates at 40MHz, and the axis control circuit operates at 20 MHz.
Clock synchronizing function for motor control corrects 20 MHz CLK signals to the same frequency as a clock frequency 20 MHz of a clock master LSI specified by RSYN.DNMST bit.

You can monitor correction amount by RSYN.FAM bit (FAM value). Writing to this bit is disabled.

A FAM value is shown by 8 bit integer with a sign (−128 ~ +127). With no correction adjustment, FAM value is 0. The formula that shows the relation between CLK signal clock rate ($f_{CLK}$) and a clock frequency used for motor control circuit ($OSC_{CLK}$) is as follows.

1. When CKSL = L level (40MHz input)---$OSC_{CLK} = f_{CLK} \times (32{,}768 + \text{FAM value}) / 65536$
2. When CKSL = H level (80MHz input)---$OSC_{CLK} = f_{CLK} \times (32{,}768 + \text{FAM value}) / 131072$

When clock synchronizing function is operating, the difference with the center value of correction varies within −1 to +1 range.
You can monitor the difference with a crystal oscillation frequency of a clock master LSI. Under normal condition, you do not need to monitor the clock.

The formula of FAM value (−128 ~ +127) relation with frequency correction amount (ppm) is as follows.

Frequency correction amount = (FAM value ×1,000,000) / 32,768･･･(formula of 8.17.1)

The range of FAM value is −128 ~ +127. Therefore, the range to be monitored is from −3906 ppm ~ +3875 ppm.

[Detection of synchronizing error of the clock for motor control]

In clock synchronization, the clock master LSI sends clock frequency information (MSTS.SYN) for motor control with the main status to other G9103Cs and other G9103C monitors the information and operate synchronizing using the information.

However, if the synchronizing function of a clock master LSI turns OFF (RSYN.SYNC = 0) by mistake in synchronizing, the clock master LSI outputs the setting information of GRP terminals (MSTS.SGP) instead of synchronizing information. Other G9103C recognizes the setting information of GRP terminals as clock frequency information by mistake. Therefore, because the clock master LSI 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 becomes large. In interpolation operation, the interpolation trajectory becomes collapsed considerately. Therefore, G9103C has a feature to detect this abnormal condition.

There are two following circuits for detection: a circuit to control frequency correction amount and a circuit to control fluctuating range of frequency.

1. Circuit to control frequency correction amount (mainly to detect the difference of crystal oscillator)

   When an absolute value of FAM value exceeds the setting value, it recognizes the condition as an error. Set a value in RSYN.FAL bit (1 ~ 127). The smaller the setting value is, the more sensitive the detection sensitivity is. Under normal condition, set a value that is approximately four-times of frequency allowable deviation of a crystal oscillator. For example, when the allowable deviation is plus minus 100ppm and correction amount is 400 using the above formula 1 of 8.17.1, the FAM value becomes 13.1. Therefore, please set 13 or 14 to the setting value. In the case that the setting value is 0 (default), G9103C does not control frequency correction amount.

2. Circuit to control fluctuating range of frequency (mainly to detect RSYN.SYNC = 0 of the clock master LSI)

   The FAM value in clock synchronization for motor control fluctuates within plus or minus 1 of the center correction value. When the fluctuating range exceeds plus or minus 7, the clock master LSI recognizes this condition as an error.
   This circuit detects a rapid change of clock frequency information of the clock master LSI.
   This function operates only when synchronizing error detection is enabled (RSYN.FAWL = 1).

   When one circuit of the above 1 or 2 detects an error, G9103C occurs an error interrupt request (REST.EFAJ=1) after turning the clock synchronizing function is disabled (RSYN.SYNC = 0) automatically and switching the frequency to a default.

[Detect errors of monitoring communication for clock synchronization.]

In clock synchronization, the clock master LSI main status information that is sent to a center LSI is monitored regularly by cyclic communication.
If signals on the communication line cannot be monitored because of reflection and noise, clock synchronization cannot be operated.
G9103C has a function to count errors of monitoring communication for a clock master LSI and a function to stop motor as an error stop when monitoring errors occur specified times continuously.
Error counter has a function to stop counting at the maximum value (255) and you can check a value by

RMEC.CKEC bit and cleared by CKMECR (0030h) command.
The number of occurrences of continuous communication monitoring error is specified with values 0 ~ 15 in RSYN2.CKMEV bit. When 0 is selected, error stop function does not operate. When a value other than 0 is selected, motor stops because of error stop when specified number of errors interrupt (REST.ECKM = 1) occur. If monitoring communication is done normally even once before the number of continuous errors reaches to the setting value, the count of continuous error restart from 0.

| | |
|---|---|
| Device Number of clock master LSI <RSYN.DNMST><br>Set a device number of a clock master LSI (0 ~ 63). | [RSYN] (WRITE)<br>7 0<br>- - n n n n n n |
| Set clock synchronization for motor control <RSYN.SYNC><br>0: Disabled.<br>1: Enabled. | [RSYN] (WRITE)<br>7 0<br>- n - - - - - - |
| Clock synchronizing status for motor control. <RSYN.SYON><br>0: Not synchronizing.<br>1: Synchronizing.<br>With synchronizing control in such as interpolation operation, be sure to check RSYN.SYON = 1. | [RSYN] (READ)<br>7 0<br>n - - - - - - - |
| Limit of correction of clock frequency for motor control <RSYN.FAL><br>When correction exceeds a limit, it is regarded as a synchronizing error.<br>If you set 0, error detection function will be disabled. | [RSYN] (WRITE)<br>23 16<br>- n n n n n n n |
| Control function of clock frequency range for motor control. <RSYN.FAWL><br>0: Disabled.<br>1: Enabled. : Output a synchronizing error when width of fluctuation is more than ±7. | [RSYN] (WRITE)<br>23 16<br>n - - - - - - - |
| Correction amount of clock frequency for motor control <RSYN.FAM><br>Correction amount (−128 ~ +127) of clock frequency can be monitored. When frequency correction amount is large, please check a crystal oscillator frequency. | [RSYN] (READ)<br>31 24<br>n n n n n n n n |
| Number of communication monitoring error for clock synchronization <RMEC.CKEC><br>Count number of error that occurred on monitoring communication (0 ~ 255).<br>The number is reset to 0 at default setting and by CKMECR (0030h) command and the count number stops at 255. | [RMEC] (READ)<br>7 0<br>n n n n n n n n |
| Clear number of error that occurred on monitoring for clock synchronization. <CKMECR><br>RMEC.CKEC = 0. | [Command]<br>0030h |
| Number of continuous error for monitoring clock synchronization. <RSYN2.CKMEV><br>When a setting value is 0, error stop by continuous errors does not occur.<br>When a value (1 ~15) is selected, error stop occurs when the setting value become the same as continuous errors count. | [RSYN2] (WRITE)<br>23 16<br>- - - - n n n n |
| Error interrupt status <REST.ECKM ><br>1: When communication monitoring error for clock synchronization specified times, motor stops as an error. | [REST] (READ)<br>23 16<br>- - - - n - - - |

### 8.17.2 Simultaneous stop function

When the specified other motor 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 number of a monitoring target LSI in RSYN register.
   RSYN.DNSTP ← Device number of a monitoring target LSI
   RSYN.SYNE ←1
2. Set RSYN register of other Axes. Set a monitoring target LSI so as to make a loop.
3. In order to transmit error stop information when own axis stops to other axes, set error stop signal output is enabled (RMD.MERO = 1).
4. In order to stop own axis when other axis stops, set error stop signal input is enabled (RMD.MERI = 1).

Example for setting

The following is an example when using six G9103Cs and the device numbers are 3, 5, 6, 50, 55 and 60. In this example, either device number 3 or 5 of G9103C stops by an error, the device numbers 3, 5 and 6 are all stopped. If the device number 6 is stopped, the device number 3 and 5 are not stopped. If one of the device numbers 50, 55 or 60 is stopped, the device number 50, 55 and 60 are all stopped.

| Device No. | No 3 | No 5 | 6 |
|---|---|---|---|
| RSYN.DNSTP | 000110(6) | 000011(3) | 000101(5) |
| RSYN.SYNE | 1 | 1 | 1 |
| PRMD.MERO | 1 | 1 | 0 |
| PRMD.MERI | 1 | 1 | 1 |

| Device No. | 50 | 55 | 60 |
|---|---|---|---|
| RSYN.DNSTP | 111100(60) | 110010(50) | 110111(55) |
| RSYN.SYNE | 1 | 1 | 1 |
| RMD.MERO | 1 | 1 | 1 |
| RMD.MERI | 1 | 1 | 1 |

Note. The information of error stop is transmitted by cyclic communication. Therefore, set the setting value in RSYN.DNSTP bit as the device number of LSI that executes cyclic communication just before own LSI's cyclic communication.
Error stop is a case that an error interrupt occurs while a motor stops.

[Error detections of monitoring communication for simultaneous stop]

In simultaneous stop function, a monitoring target LSI's main status information that is sent to center LSI regularly is monitored by cyclic communication.
If there is a high probability to fail monitoring signals on a communication line because of reflection and noise, it takes times to communicate simultaneous stop information.
G9103C has an error count function for monitoring communication for simultaneous stop and error stop function when monitoring errors occur specified times.
Error stop counter has a count function up to the maximum value (255). You can check count by reading RMEC.SPEC bit. The count number can be cleared by SMECR (0031h) command.
Error continuous occurrence number is specified with a value 0 ~ 15 in RSYN2.SPMEV bit.
When 0 is selected, error stop function does not operate. When a value other than 0 is selected, the motor stops because of error stop when specified number of errors occur continuously and error interrupt (REST.ESPM = 1).
If communication monitoring is done normally even once before the number of continuous errors reaches to the setting value, the count of continuous error restart from 0.

| | |
|---|---|
| Device number of a monitoring target LSI <RSYN.DNSTP><br>Set a device number of a monitoring target LSI. (0 ~ 63) | [RSYN] (WRITE)<br>15 8<br>- - n n n n n n |
| Synchronizing function of simultaneous stop <RSYN.SYNE><br>0: Disabled.<br>1: Enabled. | [RSYN] (WRITE)<br>15 8<br>- n - - - - - - |
| Function to receive a stop request when other axis stops by an error.<RMD.MERI><br>0: Disabled.<br>1: Enabled. | [RMD] (WRITE)<br>31 24<br>- - - n - - - - |
| Function to send stop request when own axis stops by an error <RMD.MERO><br>0: Disabled.<br>1: Enabled. | [PRMD] (WRITE)<br>31 24<br>- - n - - - - - |
| Number of communication monitoring error for simultaneous stop <RMEC.CKEC><br>Count a number of error that occurred on monitoring communication (0 ~ 255).<br>Cleared by default setting and SPMECR (0031h) command and the count number stops at 255. | [RMEC] (READ)<br>15 8<br>n n n n n n n n |
| Clear number of communication monitoring error for simultaneous stop. <SPMECR><br>Set RMEC.CPEC = 0. | [Command]<br>0031h |
| Number of continuous communication monitoring error for simultaneous stop. <RSYN2.SPMEV><br>When the specified value is 0, error stop because of continuous errors does not occur.<br>When a value (1 ~15) is selected, error stop occurs when continuous error count become the same as the setting value. | [RSYN2] (WRITE)<br>23 16<br>n n n n - - - - |
| Error interrupt status <REST.ESPM><br>1: When communication monitoring errors for simultaneous stop a specified times, motor stops as an error continuously. | [REST] (READ)<br>23 16<br>- - - n - - - - |

8.17.3 Substitute input function for sensor signals

Usually, input of +EL, −EL, SD, ORG, EMG, ALM, STA, STP signals are connected to input terminals of G9103C.
However, if high responsiveness does not needed, save wiring is used by inputting these signals to other I/O devices (local LSI).
By cyclic communication, communication information that is sent from I/O device to a center device (a center LSI) can be use as substitute signals by monitoring.

Setting procedure
1. Specify information of an I/O device that inputs sensor signals to RSYN2 register.

RSYN2.DNSWM ← Device number of I/O device
RSYN2.PNSWM ← Port number of I/O device
RSYN2.PEME ← Specify a method to input +EL signals.
(0: Input from G9103C's terminals, 1: Input from bit 0 terminal of I/O device port.)

RSYN2.MEME ← Specify a method to input −EL signals.
(0: Input from G9103C's terminals, 1: Input from bit 1 terminal of I/O device port.)

RSYN2.SDME ← Specify a method to input SD signals.
(0: Input from G9103C's terminals, 1: Input from bit 2 terminal of I/O device port.)
RSYN2.ORME ← Specify a method to input ORG signals.
(0: Input from G9103C's terminals, 1: Input from bit 3 terminal of I/O device port.)

RSYN2.EMME ← Specify a method to input EMG signals.
(0: Input from G9103C's terminals, 1: Input from bit 4 terminal of I/O device port.)

RSYN2.ALME ← Specify a method to input ALM signals.
(0: Input from G9103C's terminals, 1: Input from bit 5 terminal of I/O device port.)

RSYN2.SAME ← Specify a method to input STA signals.
(0: Input from G9103C's terminals, 1: Input from bit 6 terminal of I/O device port.)

RSYN2.SPME ← Specify a method to input STP signals.
(0: Input from G9103C's terminals, 1: Input from bit 7 terminal of I/O device port.)

RSYN2.SWMEV ← Specify a number of continuous error of communication monitoring for substitute input for sensor signals. (0 ~ 15)

G9103C terminal input signals and signals to monitor communication are switched just after G9103C terminal input buffer.
Therefore, setting of terminal input specifications such as inserting a noise filter or input logic transformation is enabled when using a substitute input.

[Error detection when monitoring communication for substitute input]

Substitute input function monitors specified I/O port information to be sent to a center LSI regularly by cyclic communication.
When there may be high probability that signals on the communication line to fail in monitoring because of reflection or noise, it takes times to deliver alternate signals.
To check quality, G9103C has a function to count communication monitoring error for a substitute input and a function to stop a motor as error stop when monitor errors occur continuously specified time.
Error counter can stop counting when the count reach the max value 255 and can be read by RMEC.SWEC bit and can be cleared SWMECR (0032h) command.
Error continuous occurrence time is set with 0 ~ 15 in RSYN2.SWMEV bit. When the counter value is 0, error stop function does not operate. If the value is other than 0, error stop operate when errors occur continuously specified time, and error interrupt occurs (REST.ESWM = 1). If monitoring normally even once before the error count reach the specified times even once, the counter value starts 0 again.

Considering safety, please set other than 0 to RSYN2.SWMEV bit.

| | |
|---|---|
| Device number of a substitute input device <RSYN2.DNSWM><br>Specify I/O device number (0 ~ 63) that input sensor signals alternately. | [RSYN2] (WRITE)<br>7 0<br>- - n n n n n n |
| Specify port number of a substitute input device <RSYN2.PNSWM><br>Specify ports number (0 to 3) of I/O device that input sensor signals alternately.<br>0: P00 ~ P07, 1: P10 ~ P17, 2: P20 ~ P27, 3: P30 ~ P37 | [RSYN2] (WRITE)<br>7 0<br>n n - - - - - - |
| Input of EL signals. <RSYN2.PEME><br>0: Input from EL terminal of G9103C.<br>1: Input from Px0 terminal of I/O device (x = 0 ~ 3) | [RSYN2] (WRITE)<br>15 8<br>- - - - - - - n |
| Input of −EL signals. <RSYN2.MEME><br>0: Input from – EL terminal of G9103C<br>1: Input from Px1 terminal of I/O device (x = 0 ~ 3) | [PSYN2] (WRITE)<br>15 8<br>- - - - - - n - |
| Input of SD signals <RSYN2.SDME><br>0: Input from SD terminals of G9103C.<br>1: Input from Px2 terminal of I/O device (x = 0 ~ 3) | [RSYN2] (WRITE)<br>15 8<br>- - - - - n - - |
| Input of ORG signals. <RSYN2.ORME><br>0: Input from the ORG terminal of G9103C.<br>1: Input from Px3 terminal of I/O device (x = 0 ~ 3) | [RSYN2] (WRITE)<br>15 8<br>- - - - n - - - |
| Input of EMG signals. <RSYN2.EMME><br>0: Input from the EMG terminal of G9103C.<br>1: Input from Px4 terminal of I/O device. (x = 0 ~ 3) | [RSYN2] (WRITE)<br>15 8<br>- - - n - - - - |
| Input of ALM signals. <RSYN2.ALME><br>0: Input from the –EL terminal of G9103C.<br>1: Input from the Px1 terminal of I/O device. (x = 0 ~ 3) | [RSYN2] (WRITE)<br>15 8<br>- - n - - - - - |
| Input of STA signals. <RSYN2.SAME><br>0: Input from the SD terminal of G9103C<br>1: Input from the Px2 terminal of I/O device (x = 0 ~ 3) | [RSTY2] (WRITE)<br>15 8<br>- n - - - - - - |
| Input of STP signals. <RSYN2.SPME><br>0: Input from the ORG terminal of G9103C.<br>1: Input from the Px3 terminal of I/O device (x = 0 ~ 3) | [RSYN2] (WRITE)<br>7 0<br>- - n n n n n n |
| Number of monitoring error for a substitute input <RMEC.SWEC><br>Count a number of error the occurred on monitoring communication (0 ~ 255).<br>The number of reset to at default setting and by SWMECR command and the count number stops at 255. | [RMEC] (READ)<br>23 16<br>n n n n n n n n |
| Clear number of error that occurred on monitoring for a substitute input. <SWMECR><br>Set to RMEC.SWEC = 0 | [Command]<br>0032h |
| Number of continuous errors for monitoring substitute input. <RSYN2.SWMEV><br>When the setting value is 0, error stop because of continuous errors does not occur.<br>When a value (1 ~ 15) is selected, error stop occurs when the setting value is become the same as continuous error count. | [RSYN2] (WRITE)<br>31 24<br>- - - - n n n n |
| Error interrupt status <REST.ESWM ><br>1: When monitoring error for substitute occurs specified times continuously, motor stops as an error. | [REST] (READ)<br>23 16<br>- - n - - - - - |

## *8.18 Unit ID control function*

If setting ROME terminal (Pin12) to H level, EEPROM connection mode for unit ID is available. Unit ID information mode within EEPROM is downloaded into the general-purpose register (RGN0 ~RGN3) automatically after G9103C is 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 a GND terminal.

Please use an EEPROM that satisfies the following specifications.

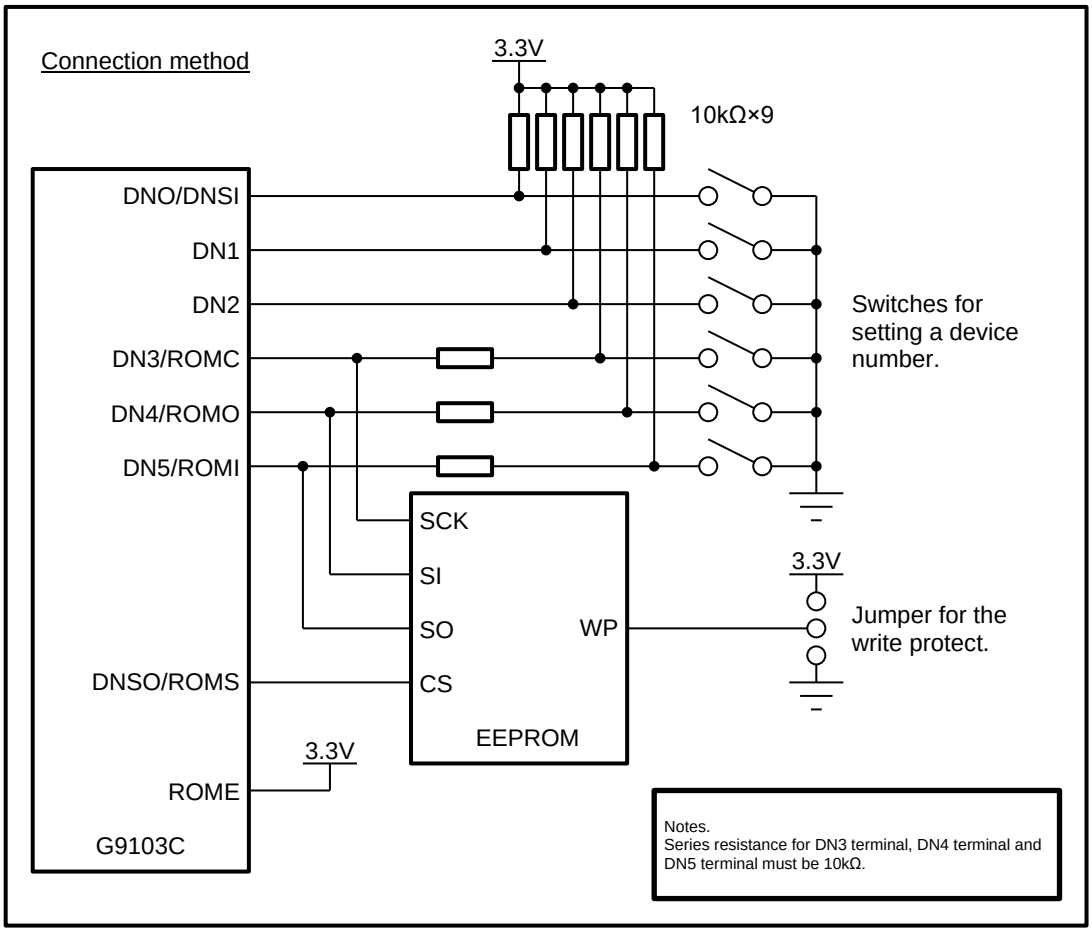
- 4-wire serial interface. (Equivalent to SPI)
- Address length is 8 bits.
- Data length is 8 bits.
- More than 16 bytes should be written by one writing command. (Only use 16 bytes.)
- 3.3 V power supply and should be operate with more than 2.5MHz.
- If bits 2 and 3 are “00b”, writing is available. If bits 2 and 3 are “11b”, writing is prohibited.
- 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/8 bit.）

| 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. Please confirm specifications with manufactures' latest information.



Note 1: Specify EEPROM write protect terminals not as to change the Unit ID within EEPROM by mistakes.
Note 2: When ROME terminal (Pin 12) is H level, the default value of RGN0 ~ RGN3 is 0.
Note 3: In the communication during accessing EEPROM, use a device number stored in G9103C

8.18.1 EEPROM control command

There are only 4 commands of G9103C for EEPROM control.

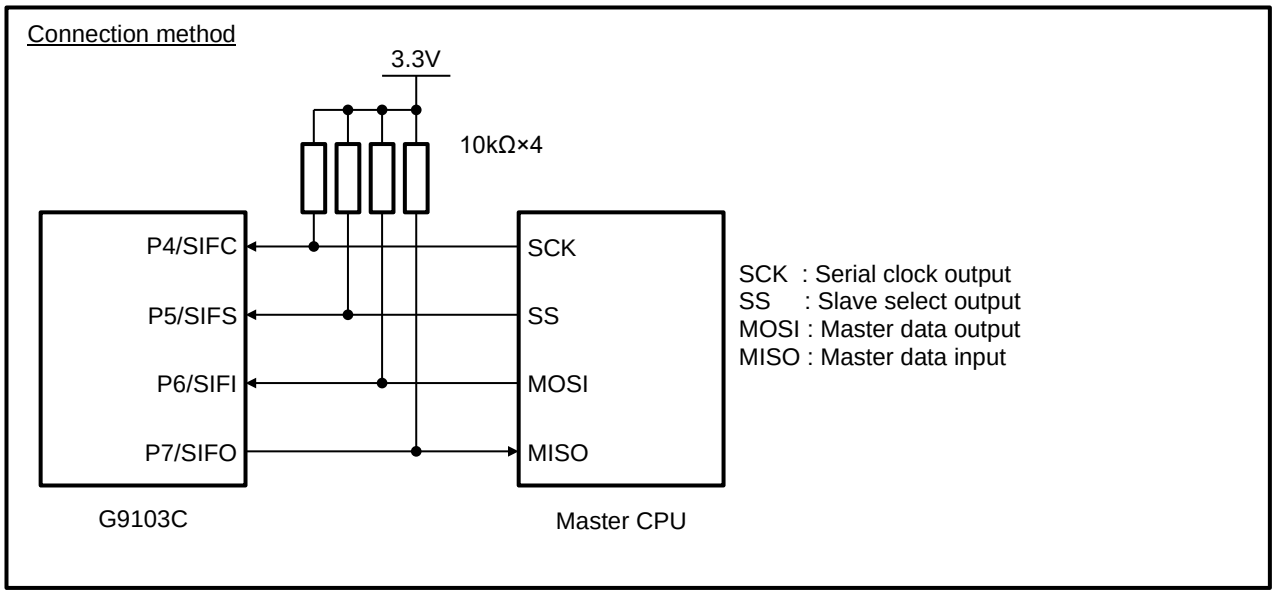
| Command name | Code | Contents |
|---|---|---|
| ROMPE | 000Ah | Prohibits writing to the external EEPROM.<br>G9103C performs the following to EEPROM.<br>1. Send a command (00000110) to permit writing to EEPROM.<br>2. Secure 3.2 µs interval time with ROMS = H level.<br>3. Write “11111100” to EEPROM status register. |
| ROMPD | 000Bh | Make the external EEPROM permitted to be written.<br>1. Send the command (00000110) to permit writing to EEPROM.<br>2. Secure 3.2 µs interval time with ROMS = H level.<br>3. Write “11110000” to EEPROM status register. |
| ROMWR | 000Ch | Write the content of RGN0 ~ RGN3 to EEPROM.<br>G9103C performs the following to EEPROM.<br>1. Send the command (00000110) to permit writing to EEPROM.<br>2. Secure 3.2 µs interval time with ROMS = H level.<br>3. Write 16 bytes of RGN0 ~ RGN3 to EEPROM. |
| ROMRD | 000Dh | Read EEPROM and set to RGN0 ~ RGN3.<br>G9103C performs the following to EEPROM.<br>1. Read from 16 bytes from address 0 at once and write to the RGN0 ~ RGN3. |

Note 1: When ROME terminal (Pin 12) is H level, ROMRD command is executed automatically under the following conditions.
1. Just after being reset by RST signal.
2. Just after being reset by SRST (0004h) command
3. Just after being reset by broadcast communication command (ggg = group number)

Note 2: Communication time and writing time to EEPROM is needed until process within EEPROM completes after the center LSI sends ROMPE, ROMPD, ROMWR commands. Make sure not to send next EEPROM control command or not to power off for this period.
The maximum communication time at 20 Mbps is maximum 168 µs. 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*

If CPU connection mode is enabled (RENV2.SIFM = 1), general-purpose I/O terminals (P4 ~ P7) are changed to serial I/F control terminals (SIFC, SIFS, SIFI, and SIFO). Access to RGN0 ~ RGN3 by 4-wire serial interface are available. (Other registers except RGN0 ~ RGN3 cannot be accessed.) However, RGN0 register is for the exclusive use of unit ID control. Therefore, do not write from CPU.




Note 1: After power-on, G9103C's four general-purpose terminals are input terminals until CPU connection mode is enabled (REMV2.SIFM = 1) through **Motionnet**® and becomes High level by pull-up.
Note 2: While CPU connection mode is enabled (RENV2.SIFM = 1), setting for I/O of RENV2.P4M, P5M, P6M and P7M bit is disabled.
Note 3: For frequency of the serial clock (SCK signal), use 2.5 MHz or less.

G9103C can be accessed from CPU like serial RAM with 16-byte capacity. Address map is as follows.

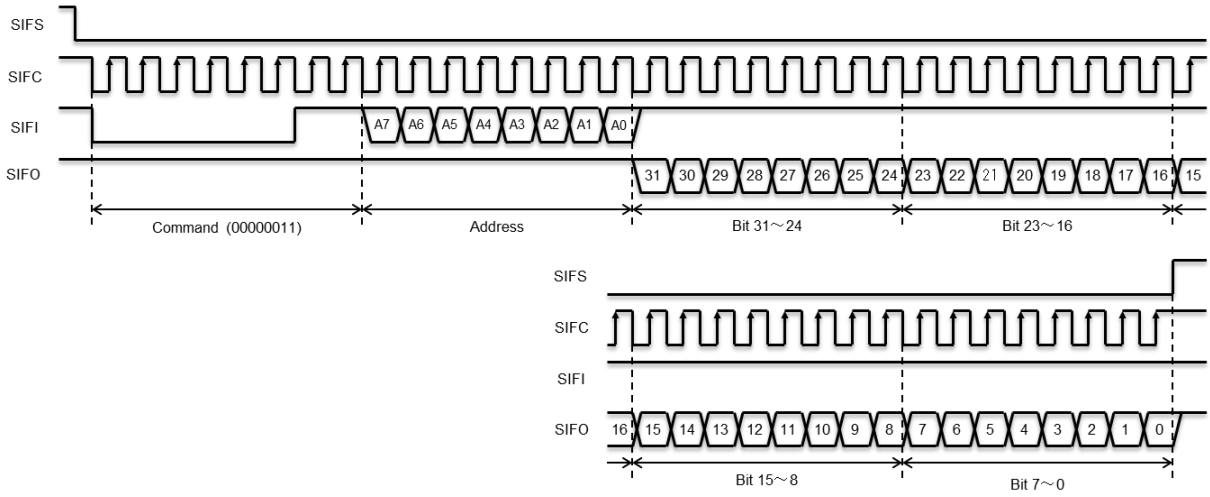
| Address | Contents |
|---|---|
| 0 | RGN0 bits 31 ~ 24 |
| 1 | RGN0 bits 23 ~ 16 |
| 2 | RGN0 bits 15 ~ 8 |
| 3 | RGN0 bits 7 ~ 0 (LSB) |
| 4 | RGN1 bits 31 ~ 24 |
| 5 | RGN1 bits 23 ~ 16 |
| 6 | RGN1 bits 15 ~ 8 |
| 7 | RGN1 bits 7 ~ 0 (LSB) |

| Address | Contents |
|---|---|
| 8 | RGN2 bits 31 ~ 24 |
| 9 | RGN2 bits 23 ~ 16 |
| 10 | RGN2 bits 15 ~ 8 |
| 11 | RGN2 bits 7 ~ 0 (LSB) |
| 12 | RGN3 bits 31 ~ 24 |
| 13 | RGN3 bits 23 ~ 16 |
| 14 | RGN3 bits 15 ~ 8 |
| 15 | RGN3 bits 7 ~ 0 (LSB) |

[Notice for access]
Do not access only the specific bits of a register. Register must be access in 32-bit units. For example, even if you want to read bit 15 ~ 8 (address 6) of RGN1, read bit 31 ~ 0 (address 4 ~ 7). You can access multiple registers at once.

| Function of terminals P4 ~ P7. <REST2.SIFM> | [RENV2] (WRITE) |
|---|---|
| 0: General-purpose input terminal.<br>1: Terminals for serial bus control. | 23 16<br>n - - - - - - - |

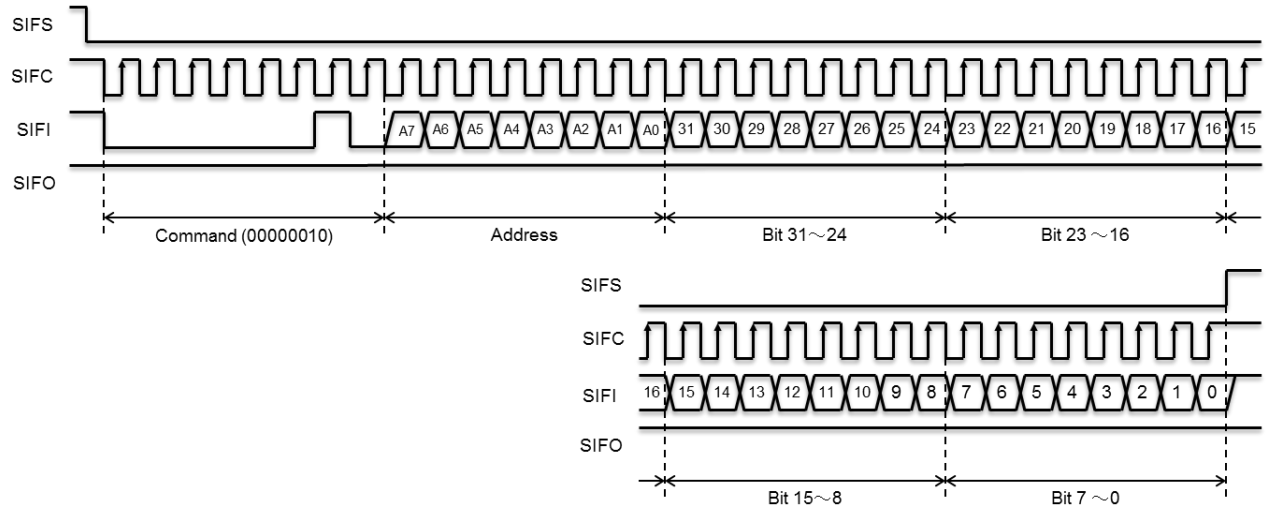
8.19.1 Timing to read (when reading 4 bytes）




1. After set SIFS = L level, CPU synchronizes with the falling of SIFC signal, input read command (03h) to G9103C.After that, as synchronized with the falling of SIFC signal, 8 bit data is output from the upper bit.
2. After that, as synchronized with the falling of SIFC signal, the upper bits outputs 8 bit data.
3. If SIFC signal is continued to input, address is added +1 and up to 16 bites can be read.
4. CPU reads SIF0 data as synchronized with the rising of SIFC signal.
5. After read to the D0 bit of the last data, set SIFS = H level.

Note. The upper addresses (A7 ~ A4) are ignored. The address next to address 15 returns address 0.

8.19.2 Timing to write (when writing 4 bytes）




1. After set SIFS = L level, CPU synchronizes with the falling of SIFC, input read command (02h) to G9103C.After that, as synchronized with the falling of SIFC signal, input 8-bit read start address from the upper.
2. After that, as synchronized with the falling of SIFC, input 8-bit data from the upper bit.
3. If SIFC signal is continued to input, address is added +1 and up to 16 bites can be read.
4. CPU reads SIFC data as synchronized with the rising of SIFC signal.
5. After read to the D0 bit of the last data, set SIFS = H level.

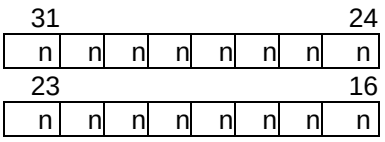
Note. The upper address (A4 ~ 7) are ignored. The address next to address 15 returns address 0.

## *8.20 ID code confirmation*

To distinguish from other LSI, ID code is included.

ID code can be checked by the following process.
1. Writes IDMON (0003h) command.
2. Writes PRMG (00D5h) command.
3. Check the upper 16 bit of data read.

Only when IDMON command and PRMG command are used continuously, you can read ID cord.
Except the case of the upper procedure to use RRMG command, ID code part becomes “0”.
G9103C’s ID code is “03C0h”.

| | |
|---|---|
| ID code <RMG.IDCD><br><br>It is an ID cord to distinguish from other LSIs.<br>G9103C’s ID code is "03C0h". | [RMG] (READ)<br>  |
| Command to confirm ID code <IDMON><br><br>Using RRMG command, make ID code to be read out. | [Command]<br>0003h |
| RMG readout command <RRMG><br>Read out RMG register.<br>If IDMON command is used continuously, the upper 16-bit becomes ID code. | [Command]<br>00D5h |

# 9. How to calculate communication cycle time

## 9.1 Cyclic communication

Cyclic communication is a control to repeat 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 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 communication rate varies, I/O communication time, of course varies.)

One I/O communication time is shown as follows.

| I/O communication time [µs] = (7.7 x K ) + 7.4 |
|---|

"K" is a communication rate coefficient. Please refer to the following table.

| Communication 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 a device number of the connected local LSIs. Therefore, one cycle of cyclic communication is calculated as follows.

| Cycle of cyclic communication [µs] = (I/O communication time) x N<br>= ((7.7 x K) + 7.4) x N |
|---|

"N" shows a number of local LSIs that are connected.

Ex.: Calculating cycle time with a communication rate of 20 Mbps and 30 local LSIs.
(7.7 x 1 + 7.4) x 30 = 453 µs

## 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 (data length is variable).
2) When there is no response frame from a local LSI.

| Basic item | Required time (µs) |
|---|---|
| ST: Data sending time | (B x 0.6 + 3.25) x K |
| RT: Time to execute command | (C x 0.3) |
| JT: Response time with returning data | (B x 0.6 + 5.65) x K |
| JT: Response time without returning data | 5.05 x K |

"B" is number of bytes of data. "C" is a number of commands.

| One complete data communication cycle [µs] = ST +RT + JT + 7.4 |
|---|

## *9.3 Total cycle time including data communication*

The total time can be obtained by adding data communication times to an ordinary communication cycle time.

Ex.1: Communication rate = 20 Mbps. 34 local LSIs are connected. Data communication of reading one the G9103C and register (2 bytes for sending and 6 bytes for receiving) is performed four times in cyclic communication.

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+ (1 x 0.3) + (6 x 0.6 + 5.65) x 1 + 7.4} x 4
= 513.4 + 21.4 x 4
= 599 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 ~ +4.0 | V |
| Input voltage | $V_{IN}$ | −0.3 ~ $V_{DD}$ +0.5 Note 1 | V |
| Output voltage | $V_{out}$ | −0.3 ~ $V_{DD}$ +0.5 Note 1 | V |
| Input current / terminal | $I_{out}$ | −30 ~ +30 | mA |
| Storage temperature | $T_{Stg}$ | −65 ~ +150 | °C |

Note 1: All signal terminals except “VDD” and “GND” can be connected up to +7.0V.

### *10.2 Recommended operating conditions*

| Item | Symbol | Rating | Unit |
|---|---|---|---|
| Power supply voltage | $V_{DD}$ | +3.0 ~ +3.6 | V |
| Input voltage | $V_{IN}$ | −0.3 ~ $V_{DD}$ +0.3 Note 1 | V |
| Storage temperature | Ta | −40 ~ +85 | °C |

Note 1: All signal terminals except “VDD” and “GND” can be connected up to +5.8V.

### *10.3 DC characteristics*

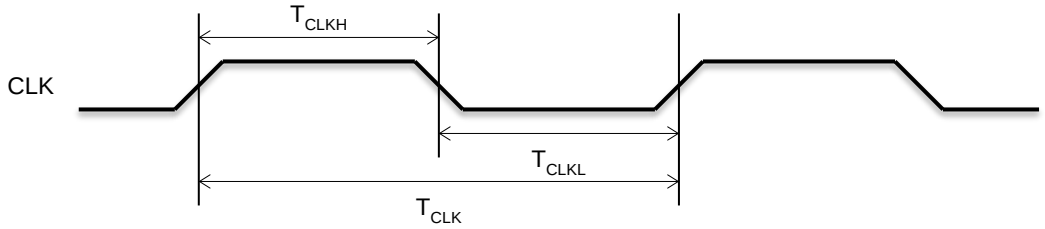
| Item | Symbol | Condition | Min. | Max. | Unit |
|---|---|---|---|---|---|
| Current consumption | $I_{dd}$ | CLK = 80 MHz | - | 67 | mA |
| | | CLK = 40 MHz | - | 47 | |
| Terminal capacitance | | | - | 10 | pF |
| H level input current | $I_{HI}$ | $V_{IH} = V_{DD}$ *3,*4, *5 | - | 1 | µA |
| | | $V_{IH}$ = 5.5 V *3,*4 | - | 30 | |
| L level input current | $I_{LI}$ | $V_{IL}$ = 0.0 V | −1 | - | µA |
| H level input voltage | $V_{IH}$ | *2, *3, *4, *5 | 2.0 | 5.8 | V |
| L level input voltage | $V_{IL}$ | *2, *3, *4, *5 | −0.3 | 0.8 | V |
| Rising input threshold voltage | $V_{T2+}$ | *1 | 1.1 | 2.4 | V |
| Rising input threshold voltage | $V_{T2-}$ | *1 | 0.6 | 1.8 | V |
| Hysteresis voltage | $V_{H2}$ | *1 | 0.1 | - | V |
| 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 |
| H level output current | $I_{OH}$ | $V_{OH} = V_{DD}$ − 0.4 V *6 | -6 | - | mA |
| | | $V_{OH} = V_{DD}$ − 0.4 V *4,*5 | -12 | - | |
| L level output current | $I_{OL}$ | $V_{OL}$ = 0.4 V *6 | - | 6 | mA |
| | | $V_{OL}$ = 0.4 V *4,*5 | - | 12 | |
| H level output voltage | $V_{OH}$ | IOH = −6 mA *6 | $V_{DD}$ − 0.4 | - | V |
| | | $I_{OH}$ = −12 mA *4,*5 | $V_{DD}$ − 0.4 | - | |
| L level output voltage | $V_{OL}$ | $I_{OL}$=6 mA *6 | - | 0.4 | V |
| | | $I_{OL}$=12 mA *4,*5 | - | 0.4 | |
| Off-state leakage current | $I_{OZ}$ | *6 | -1 | 1 | µA |

Characteristics varies depends on terminals.

| | |
|---|---|
| *1 | SI,CLK |
| *2 | ROME,SOEI,BRK |
| *3 | DN0, DN1, DN2, DN5 / ROMI, GRP0, GRP1, GRP2, DNSM, EA, EB, EZ, PA, PB, PCS, LTC, CLR, INP, EMG, +EL, −EL, SD, ORG, ALM, ELL, TUD, TMD, CKSL, SPD0, SPD1, RST |
| *4 | STA, STP, P0, P1, P2, P3, P4 / SIFC, P5 / SIFS, P6 / SIFI, P7/SIFO |
| *5 | DN3 / ROMC, DN4 / ROMO |
| *6 | DNSO / ROMNS, SOEL, SOEH, SO, ERC, OUT, DIR, CP1, CP2, CP3, BSY / PH1, FUP / PH2, FDW / PH3, MVC / PH4, MSEL, MRER, TOUT |

## *10.4 AC characteristics*

### 10.4.1 Reference clock



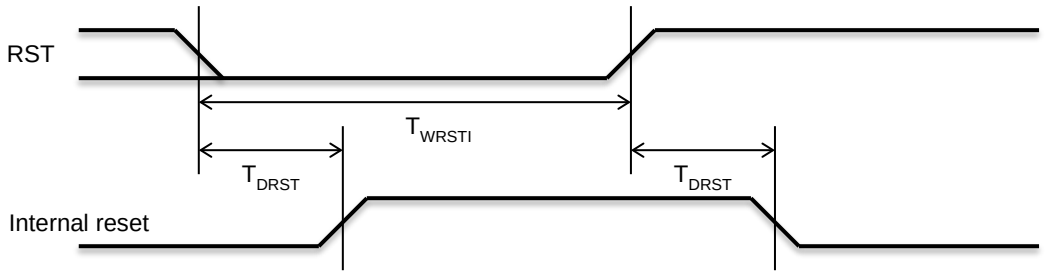

1) When setting CKSL = L level (40MHz)

| Item | Symbol | Min. | Max. | Unit |
|---|---|---|---|---|
| Frequency | $f_{CLK}$ | - | 40 | MHz |
| Cycle | $T_{CLK}$ | 25 | - | ns |
| H level duration | $T_{CLKH}$ | 10 | 15 | ns |
| L level duration | $T_{CLKL}$ | 10 | 15 | ns |

2) When setting CKSL = H level (80MHz)

| Item | Symbol | Min. | Max. | Unit |
|---|---|---|---|---|
| Frequency | $f_{CLK}$ | - | 80 | MHz |
| Cycle | $T_{CLK}$ | 12.5 | - | ns |
| H level duration | $T_{CLKH}$ | - | - | ns |
| L level duration | $T_{CLKL}$ | - | - | ns |

### 10.4.2 Reset timing




1) When setting CKSL = L level (40MHz)

| Item | Symbol | Min. | Max. | Unit |
|---|---|---|---|---|
| Reset length | $T_{WRSTI}$ | 10 | - | Clock |
| Delay time | $T_{DRST}$ | - | 10 | Clock |

1) When setting CKSL = H level (80MHz)

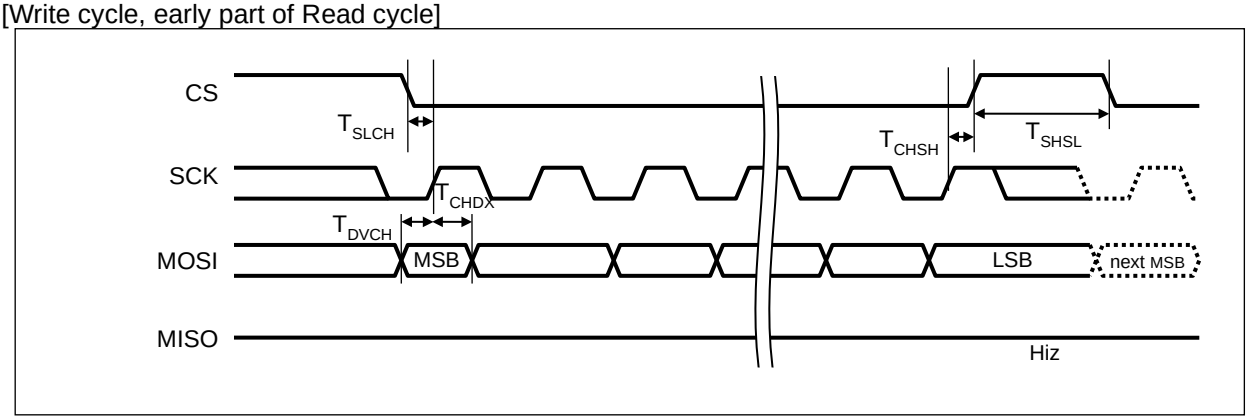
| Item | Symbol | Min. | Max. | Unit |
|---|---|---|---|---|
| Reset length | $T_{WRSTI}$ | 20 | - | Clock |
| Delay time | $T_{DRST}$ | - | 20 | Clock |

Note 1: The G9103C is ready to be used after the internal reset goes L level.
Note 2: While RST = L level, reset processing is made. During reset, please input at least 10 cycle of CLK signal when CKSL = L level. When CKSL = H level, input at least 20 cycle of CLK signal.

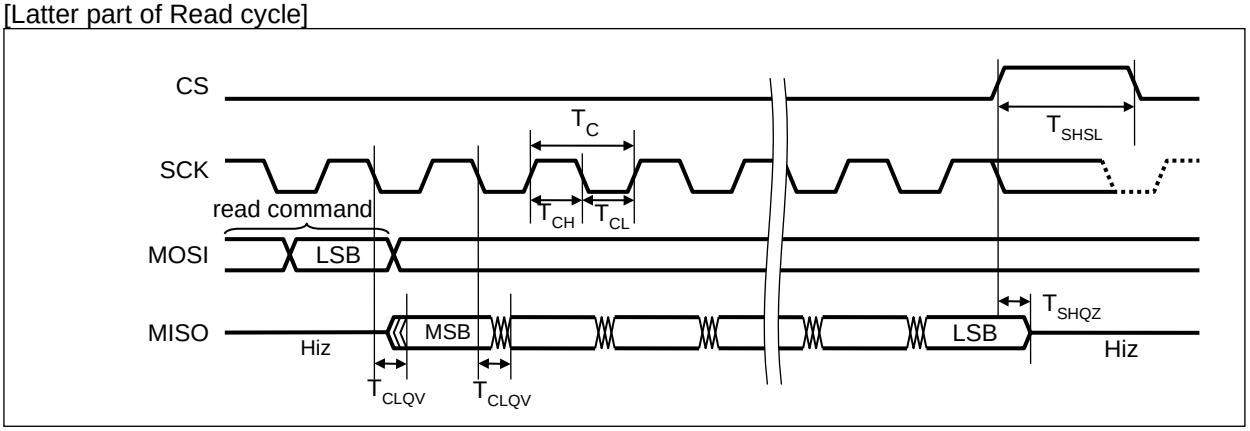
10.4.3 Serial I/F Access

10.4.3.1 Access from external CPU to G9103C

[Write cycle, early part of Read cycle]




[Latter part of Read cycle]

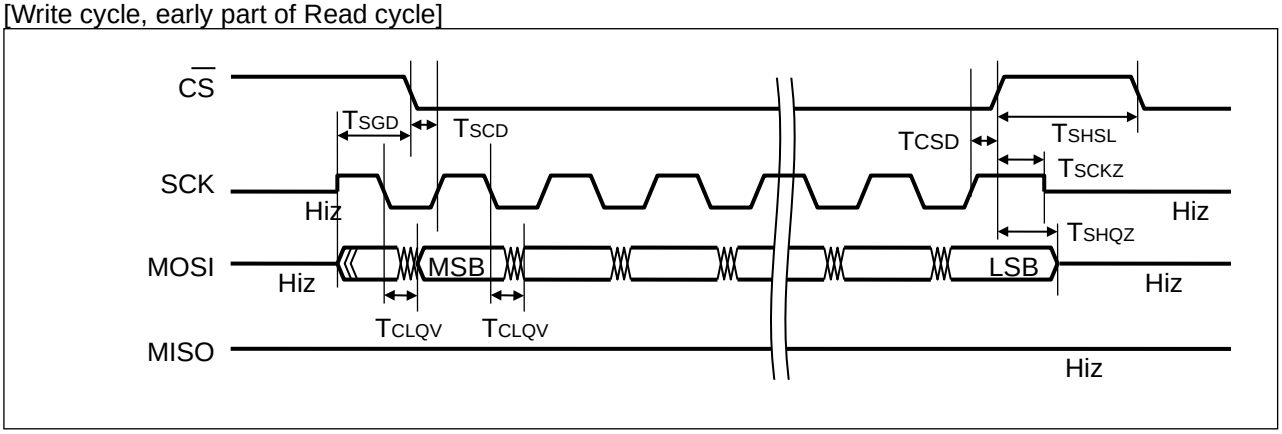



Access method of 4-wire serial interface is raising latch in mode 0 or mode 3.
While CS = H, fix to SCK = L level (in mode 1) or SCK = H level (in mode 3).

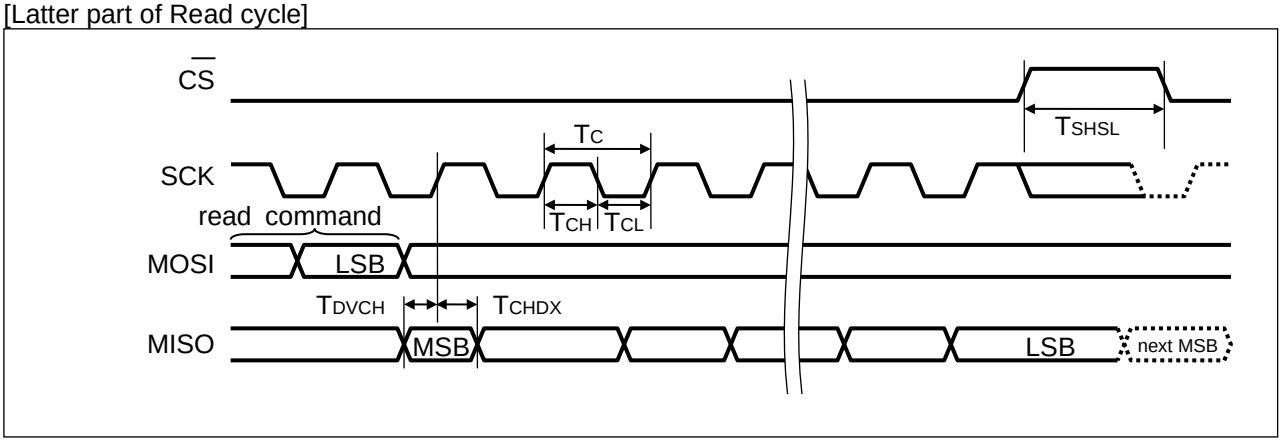
| Item | Symbol | Condition | Min | Max | Unit |
|---|---|---|---|---|---|
| Serial clock frequency | - | $C_L$=40pF | - | 2.5 | MHz |
| Serial clock cycle | $T_C$ | $C_L$=40pF | 400 | - | ns |
| Serial clock High pulse width | $T_{CH}$ | - | 160 | - | ns |
| Serial clock Low pulse width | $T_{CL}$ | - | 200 | - | ns |
| CS active set up | $T_{SLCH}$ | - | 10 | - | ns |
| CS deselect time | $T_{SHSL}$ | - | 60 | - | ns |
| CS active hold time | $T_{CHSH}$ | - | 35 | - | ns |
| Data setup time | $T_{DVCH}$ | - | 0 | - | ns |
| Data hold time | $T_{CHDX}$ | - | 35 | - | ns |
| Output disable time for CS ↑ | $T_{SHQZ}$ | $C_L$=40pF | - | 41 | ns |
| Output delay time | $T_{CLQV}$ | $C_L$=40pF | - | 87 | ns |

10.4.3.2 Access from G9103C to EEPROM

[Write cycle, early part of Read cycle]




[Latter part of Read cycle]




Access method of 4-wire serial interface is raising latch in mode 0 or mode 3.
While CS = H, fix to SCK = L level (in mode 1) or SCK = H level (in mode 3).

| Item | Symbol | Condition | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|---|
| Serial clock frequency | - | $C_L$ = 40pF | - | 2.5 | - | MHz |
| Serial clock cycle | $T_C$ | $C_L$ = 40pF | - | 16$T_{CLK}$ | - | ns |
| Serial clock High pulse width | $T_{CH}$ | $C_L$ = 40pF | - | 8$T_{CLK}$ | - | ns |
| Serial clock Low pulse width | $T_{CL}$ | $C_L$ = 40pF | - | 8$T_{CLK}$ | - | ns |
| Signal output start time for CS ↓ | $T_{SGD}$ | $C_L$ = 40pF | - | 6$T_{CLK}$ | - | ns |
| Serial clock delay time for CS ↓ | $T_{SCD}$ | $C_L$ = 40pF | - | 9$T_{CLK}$ | - | ns |
| CS deselect time | $T_{SHSL}$ | $C_L$ = 40pF | - | 128$T_{CLK}$ | - | ns |
| CS delay time for SCK ↑ | $T_{CSD}$ | $C_L$ = 40pF | - | 7$T_{CLK}$ | - | ns |
| Output delay time | $T_{CLQV}$ | $C_L$ = 40pF | - | - | 10 | ns |
| Output disable time for CS ↑ | $T_{SHQZ}$ | $C_L$ = 40pF | - | - | 238 | ns |
| Serial clock disable time for CS ↑ | $T_{SCKZ}$ | $C_L$ = 40pF | - | - | 238 | ns |
| Data setup time | $T_{DVCH}$ | - | 27 | - | - | ns |
| Data hold time | $T_{CHDX}$ | - | 0 | - | - | ns |

$T_{CLK}$ shows one cycle (25 ns) of internal CLK signal 40 MHz.

## *10.5 Operation timing*

| Item | | Symbol | Condition | Min. | Max. | Unit |
|---|---|---|---|---|---|---|
| RST signal width | | - | Note 1 | $10T_{CLK}$ ($20T_{CLK}$) | - | ns |
| CLR signal width | | - | - | $4T_{CLK}$ | - | ns |
| EA and EB signal width (Two-pulse) | | $T_{EAB}$ | Note 2 | $2T_{CLK}$ ($6T_{CLK}$) | - | ns |
| EA and EB signal phase (90 degree) | | $T_{EW}$ | Note 2 | $4T_{ICK}$ ($6T_{CLK}$) | | ns |
| EA, EB signal width (90 degree) | | $T_{EAB}$ | - | $2T_{CLK}$ | - | ns |
| EZ signal width | | - | Note 2 | $2T_{CLK}$ ($6T_{CLK}$) | - | ns |
| PA, PB signal width (2 pulses) | | $T_{PAB}$ | Note 3 | $2T_{CLK}$ ($6T_{CLK}$) | - | ns |
| PA, PB signal phase (90 degree) | | $T_{PW}$ | Note 3 | $4T_{CLK}$ ($6T_{CLK}$) | - | ns |
| PA, PB signal width (90 degree) | | $T_{PAW}$ | - | $2T_{CLK}$ | - | ns |
| ALM signal width | | - | Note 4 | $4T_{CLK}$($160T_{CLK}$) | - | ns |
| INP signal width | | - | Note 4 | $4T_{CLK}$($160T_{CLK}$) | - | ns |
| ERC signal width | | - | RENV1 bits 14~12 = 000b | 254 x $2T_{CLK}$ | 255 x $2T_{CLK}$ | ns |
| | | | RENV1 bits 14~12 = 001b | 254 x $16T_{CLK}$ | 255 x $16T_{CLK}$ | |
| | | | RENV1 bits 14~12 = 010b | 254 x $64T_{CLK}$ | 255 x $64T_{CLK}$ | |
| | | | RENV1 bits 14~12 = 011b | 254 x $256T_{CLK}$ | 255 x $256T_{CLK}$ | |
| | | | RENV1 bits 14~12 = 100b | 254 x $2048T_{CLK}$ | 255 x $2048T_{CLK}$ | |
| | | | RENV1 bits 14~12 = 101b | 254 x $8192_{CLK}$ | 255 x $8192T_{CLK}$ | |
| | | | RENV1 bits 14~12 = 111b | 254 x $16384T_{CLK}$ | 255 x $16384T_{CLK}$ | |
| | | | | LEVEL output | - | |
| +EL, −EL signal width | | - | Note 4 | $4T_{CLK}$($160T_{CLK}$) | - | ns |
| SD signal width | | - | Note 4 | $4T_{CLK}$($160T_{CLK}$) | - | ns |
| ORG signal width | | - | Note 4 | $4T_{CLK}$($160T_{CLK}$) | - | ns |
| PCS signal width | | - | - | 4 $T_{ICK}$ | - | ns |
| LTC signal width | | - | - | 4 $T_{ICK}$ | - | ns |
| STA | Output signal width | - | - | 16 $T_{ICK}$ | - | ns |
| | Input signal width | - | - | 10 $T_{ICK}$ | - | ns |
| STP | Output signal width | - | - | 16 $T_{ICK}$ | - | ns |
| | Input signal width | - | - | 10 $T_{ICK}$ | - | ns |
| BSY signal ON delay time | | $T_{SOEBSY}$ | - | | -12 $T_{CLK}$ | ns |
| | | $T_{STABSY}$ | - | | 14 $T_{CLK}$ | ns |
| Start delay time | | $T_{SOEPLS}$ | - | | 8 $T_{CLK}$ | ns |
| | | $T_{STAPLS}$ | - | | 34 $T_{CLK}$ | ns |
| General-purpose output terminal delay time | | $T_{SOEPRT}$ | - | | -26$T_{CLK}$ | ns |

"$T_{CLK}$" shows one cycle (25 ns) of the internal clock 40 MHz. While clock synchronization for motor control, a ±25ns error occurs.

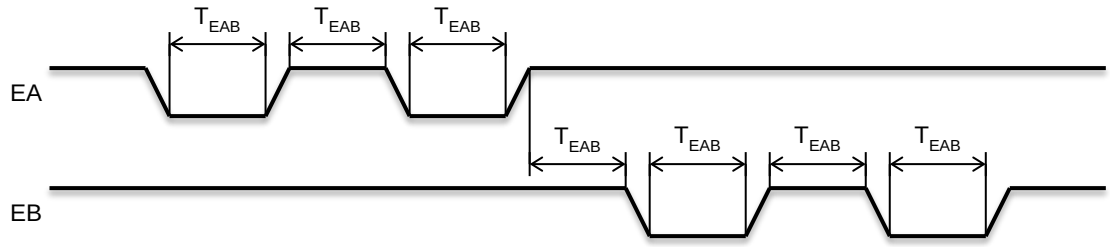
Note 1: During RST = L level, reset process will be made. During reset processing, input more than 10 clocks of CLK signal when CKSL=L level. When CKSL = H, input more than 20 clocks of CLK signals.

Note 2: If input filter is enabled (RENV2.EINF = 0), Minimum time = $6T_{CLK}$.

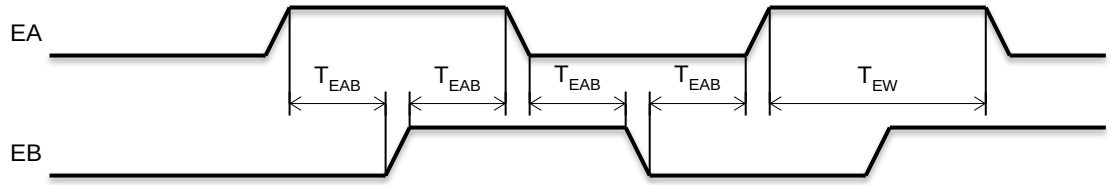
Note 3: If input filter is enabled (RENV2.PINF = 0), Minimum time = $6T_{CLK}$.

Note 4: If input filter is enabled (RENV1.FLTR = 0), Minimum time = $160T_{CLK}$.

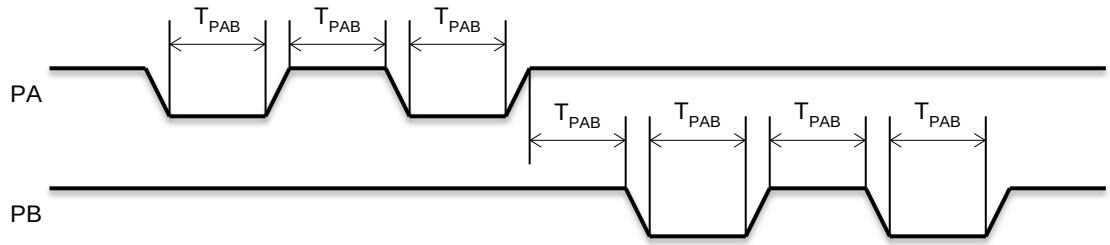
1) When EA and EB signals are in Two-pulse mode



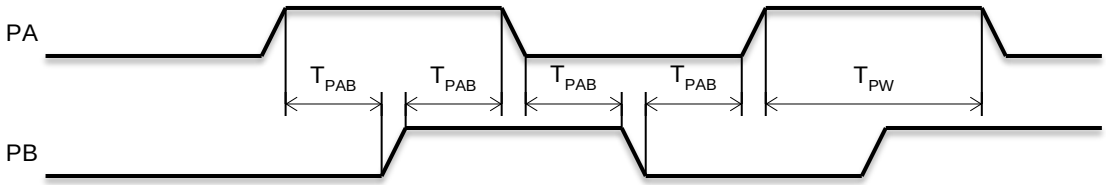

2) When EA and EB signals are in 90 degree phase-difference mode



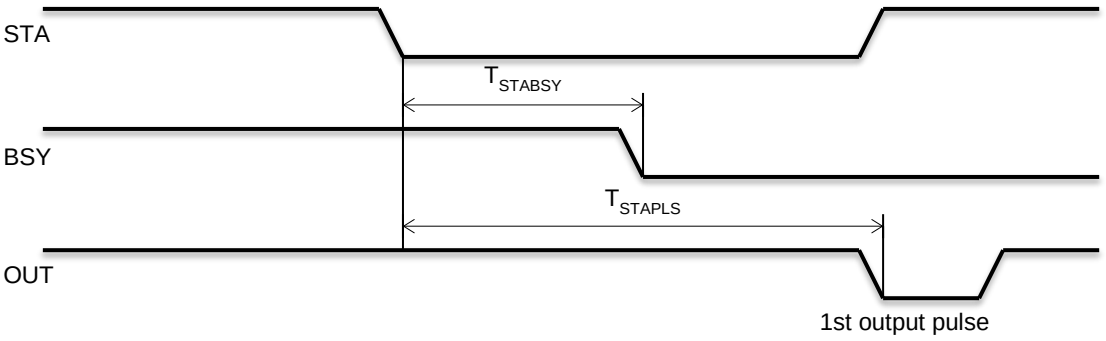

3) When PA and PB signals are in Two-pulse mode



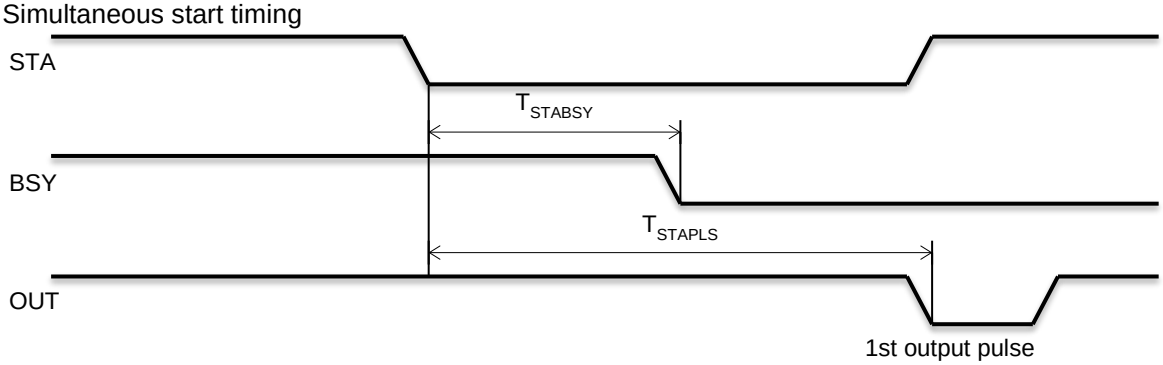

4) When PA and PB signals are in 90 degree phase-difference mode



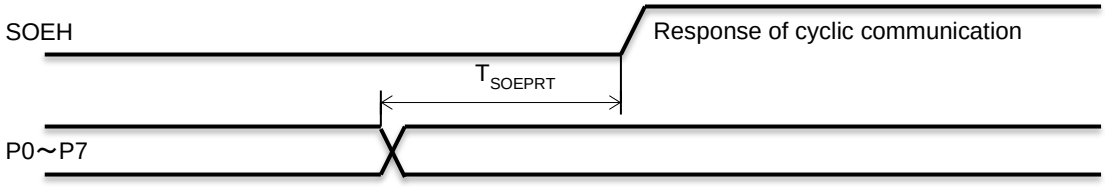

5) Start timing by commands




6) Simultaneous start timing



7) Output change timing of general output terminals

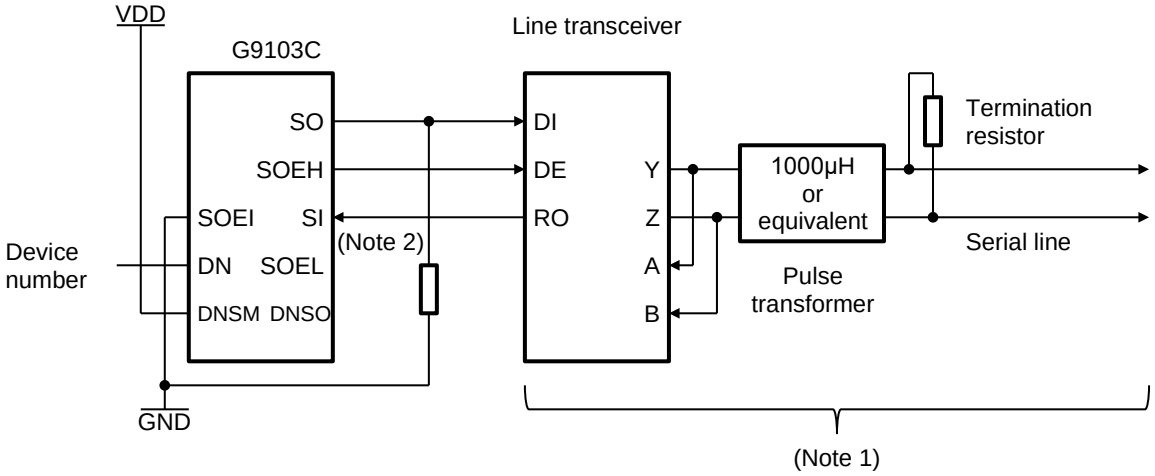



## 11. Connection example

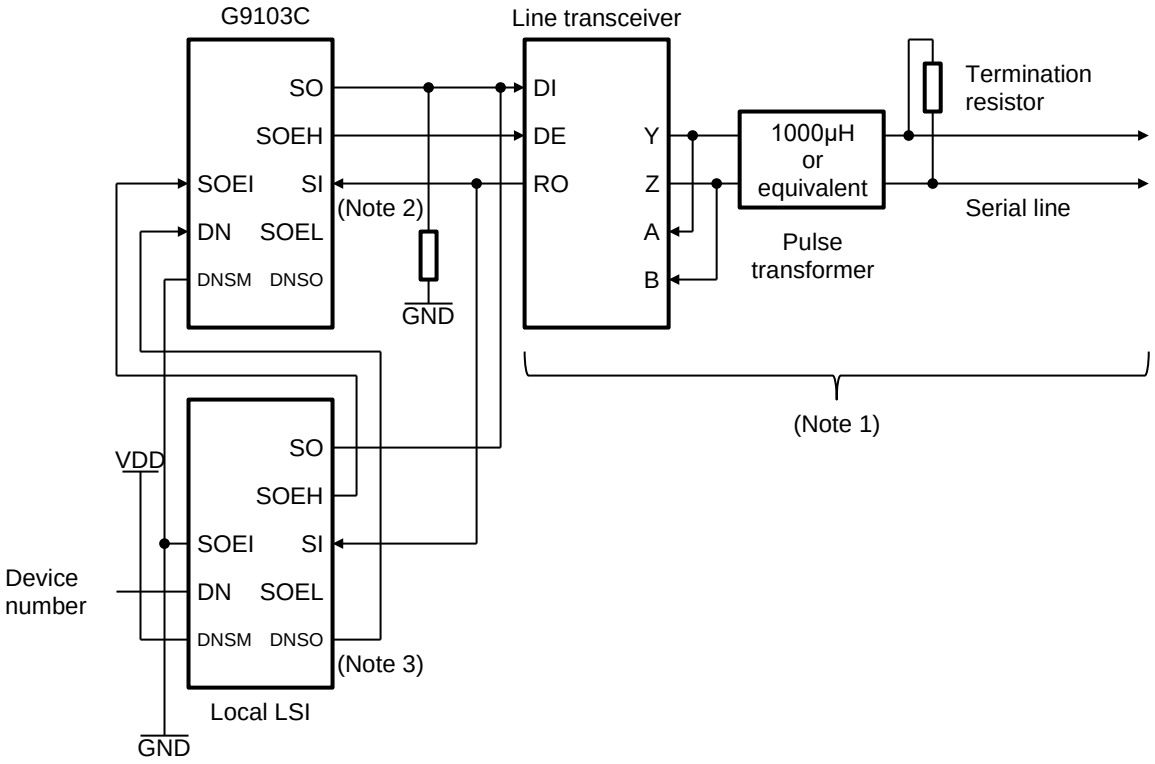
### 11.1 G9103C, line transceiver, and pulse transformer

Use RS-485 line transceivers and pulse transformers (1000 µH or equivalent) to make serial communication connections.
Connect a line transceiver as shown below.
Connect a termination resistor (which matches cable impedance) at both ends of the transmission line.
A termination resistor can be either at front or rear of a pulse transformer. The same effect will be obtained at either position.
Because SO, SI, SOEH, SOEL terminals have 5 V tolerant feature, they can be connected with TTL level 5V line transceiver directly.

(1) Circuit example for a single local LSI



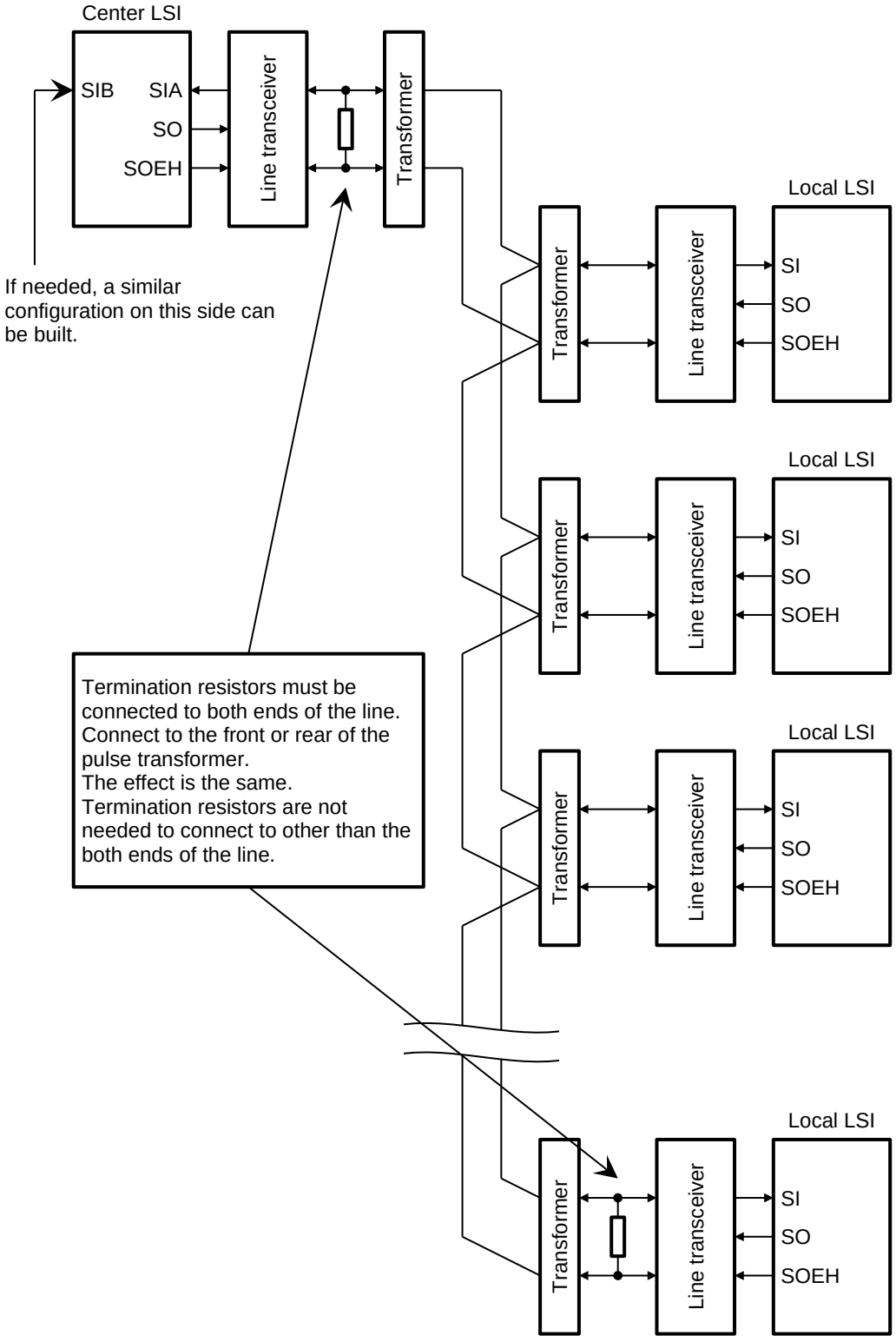
(2) Circuit example for multiple local LSIs (When a continuous device number setting is used.)




Note 1: When connecting the serial lines to line transceivers, make the path as short and straight as possible. Running these lines on a PC board complexly could deteriorate communication performance.
Note 2: Pull down resistors to GND should be 5 ~ 10 k-ohms.
Note 3: When a continuous device number is set by DNSO signal, at least 50 μs interval is needed to fix the next device number.

## *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 you use synchronizing function with other axes, connect all LSIs synchronized in one communication line.



## 12. Recommended environment

Shown below are the results of our experimental communication and the environment used for the experiment.
Please refer these when you design your system.

| 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.
A 100-ohm resistor is commonly used. Therefore, we used terminating resistors with this value.
Adjusting this resistor value may improve transmission line quality.

### 12.3 Pulse transformer

We recommend using a pulse transformer, in order to isolate GND of each local LSI.
By isolating GNDs, the system will have greater resistance to electrical noise. If pulse transformers are not used, the transmission distance may be shorter.
We used 1000 µH transformers in our experiments.

### 12.4 I/F -IC

We selected I/F-chips with specifications better than RS485 standard.
In our experiment, we used 5 V line transceivers. When you connect 3.3 V IC to 5 V line transceivers, level shifters are needed to make the connections. (To use G9103C, 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.
This list is only for your reference because other parts can be used.

| Parts | Manufacturer | Model name |
|---|---|---|
| CAT5 | Oki Wire Co., Ltd. | F-DTI-C5 (SLA) |
| CAT6 | Oki Wire Co., Ltd. | DTI-C6X |
| Pulse transformer | JPC Co., Ltd. | NPT102F,<br>NPT102G |
| Line transceiver | TEXAS INSTRUMENTS | SN75LBC180AP |

Note. Please check specifications with manufactures' latest information.

## *12.6 Other precautions*

- Cables
  When you are planning long distance transmission, cable quality will be the most important cause.
  Specialized cables designed for use as field busses, such as CC-Link and LONWORKS, have guaranteed quality.

- Pulse transformers
  The pulse transformers should handle 20 Mbps (10 MHz) without becoming saturated for a certain amount of time. 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 satisfies necessary response capacity and has bigger value, the product is worth being tried to use.

- Line transceivers
  We used TEXAS Instruments ICs for our experiments.
  Other possibilities are available from MAXIM and LINEAR TECHNOLOGY, who offer very high performance transceivers.

- Connectors
  Connectors should match cable characteristics.
  If using LAN cables, there are modular type connecters.

- 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 a cable is too long, cut the excess length of cable if possible. (Minimum 0.6 m)
  Unneeded cable length may restrict the line overall usable length, and may introduce electrical noise.

- Circuit board
  Create circuit board with 4 or more layers including VDD and GND layers to prevent noise.

## 13. Software example

This chapter outlines software for a center LSI 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 details, see the manual for the center LSI.

### 13.1 Environment and precautions used for the descriptions

The descriptions below assume the following system as a target.
-CPU mode of a center LSI is I/F mode 3.
-16-bit data bus
-The circuit around a center LSI is properly connected.
-Power used for local LSIs has already applied.
-Connection to a serial line and termination process are appropriate.

### 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<br>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<br>The lowest bit is fixed to 0. |
| Return value | Read data (16 bits) |

## *13.3 Center LSI address map*

For details, see the user's manual for the center LSI.

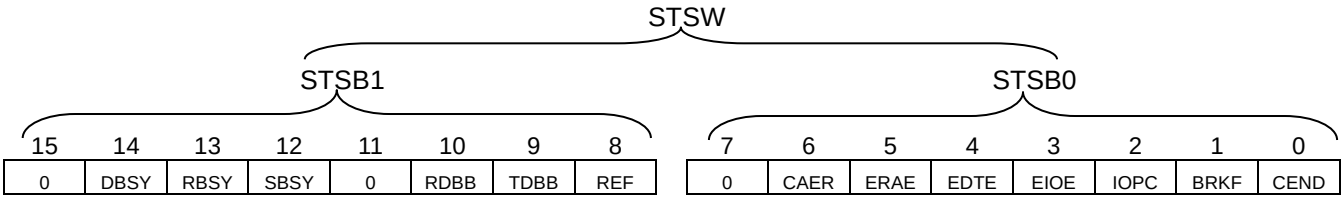
Interface mode: I/F mode 3

| A8 ~ A1 | | Writing | Reading |
|---|---|---|---|
| 0 0000 000 | 000h | Command<br>Bits 15 ~ 0 | Status<br>Bits 15 ~ 0 |
| 0 0000 001 | 002h | Disabled | Interrupt status<br>Bits 15 ~ 0 |
| 0 0000 010 | 004h | Input buffer<br>Bits 15 ~ 0 | Output buffer<br>Bits 15 ~ 0 |
| 0 0000 011 | 006h | Data transmission FIFO<br>Bits 15 ~ 0 | Data reception FIFO<br>Bits 15 ~ 8 |
| 0 0000 100<br>\|<br>0 0111 011 | 008h<br>\|<br>076h | Not defined (56 words)<br>(Any data written here will be ignored.) | Not defined (56 words)<br>(Always read as 00h.) |
| 0 0111 100 | 078h | Device information<br>(Device No.1, No.0) | Device information<br>(Device No.1, 0) |
| \| | \| | \| | \| |
| 0 1011 011 | 0B6h | Device information<br>(Device No.63, No.62) | Device information<br>(Device No.63, 62) |
| 0 1011 100 | 0B8h | Cyclic communication error flags<br>(Device No.15 ~ 0) | Cyclic communication error flags<br>(Device No.15 ~ 0) |
| \| | \| | \| | \| |
| 0 1011 111 | 0BEh | Cyclic communication error flags<br>(Device No.63 ~ 48) | Cyclic communication error flags<br>(Device No.63 ~ 48) |
| 0 1100 000 | 0C0h | Input change interrupt settings<br>(Device No.3 ~ No.0) | Input change interrupt settings<br>(Device No.3 ~ 0) |
| \| | \| | \| | \| |
| 0 1101 111 | 0DEh | Input change interrupt settings<br>(Device No.63 ~ 60) | Input change interrupt settings<br>(Device No.63 ~ 60) |
| 0 1110 000 | 0E0h | Input change interrupt flags<br>(Device No.3 ~ 0) | Input change interrupt flags<br>(Device No.3 ~ 0) |
| \| | \| | \| | \| |
| 0 1111 111 | 0FEh | Input change interrupt flags<br>(Device No.63 ~ 60) | Input change interrupt flags<br>(Device No.63 ~ 60) |
| 1 0000 000 | 100h | Port data No.1, No.0<br>(Device No.0 - Port 0, 1) | Port data No. 0,1<br>(Device No.0 - Port 1, 0) |
| 1 0000 001 | 102h | Port data No.3, 2<br>(Device No.0 - Port 3, 2) | Port data No. 3, 2<br>(Device No.0 - Port 3, 2) |
| \| | \| | \| | \| |
| 1 1111 110 | 1FCh | Port data No. 253, 252<br>(Device No.63 - Port 1, 0) | Port data No. 253, 252<br>(Device No.63 - Port 1, 0) |
| 1 1111 111 | 1FEh | Port data No.255, 254<br>(Device No.63 - Port 3, 2) | Port data No. 255, 254<br>(Device No.63 - Port 3, 2) |

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.

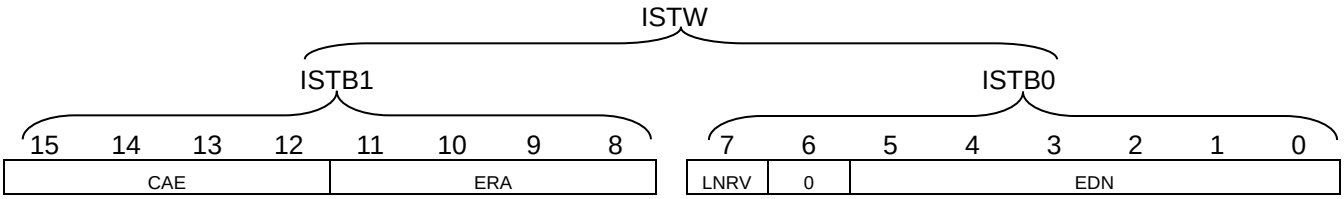



| Bit | Symbol | Description |
|---|---|---|
| 0 | CEND | Becomes 1 when ready for data to be written to the transmission FIFO buffer.<br>When the system communication or data communication is complete and the next data can be sent to the transmission FIFO buffer, this bit becomes 1 and the center LSI outputs an interrupt request signal.<br>The way to reset this bit varies on the setting of RENV0.MCLR bit. |
| 1 | BRKF | When the center LSI receives a break frame, this bit becomes 1 and an interrupt request signal is output. The way to reset this bit varies on the setting of RENV0.MCLR bit. |
| 2 | IOPC | Becomes 1 when the status of any input port that had set “input change interrupt setting” changed to 1. Then, the center LSI outputs an interrupt request signal. This signal is an OR of 256-bit "input port change interrupt flag".<br>When all the bits return to 0, this bit returns to 0. |
| 3 | EIOE | Becomes 1 when a cyclic communication (I/O communication) error occurs. Then, the center LSI then outputs an interrupt request signal.<br>This signal is an OR of 64 bit "I/O communication error flag".<br>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.<br>The way to reset this bit varies on the setting of RENV0.MCLR bit. Note 1 |
| 5 | ERAE | Becomes 1 when a "local LSI reception processing error" occurs. Then, the center LSI outputs an interrupt request signal.<br>Once the status of this bit is read, it returns to 0.<br>Then, the device number and details where the reception processing error occurred can be checked by reading interrupt status.<br>The way to clear this bit varies on the setting of RENV0.MCLR bit. |
| 6 | CAER | A CPU access error occurred.<br>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.<br>Once the status of this bit is read, it returns to 0.<br>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.<br>If data is written to the output port area, this bit becomes 1. When cyclic communication (I/O 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.<br>After data is written to the transmission 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.<br>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 cyclic communication (I/O communication) starts. |
| 13 | RBSY | During reset, this bit becomes 1. |
| 14 | DBSY | During system communication or data communication, this bit becomes 1. |
| 15 | (Not defined) | Always 0. |

Note 1: The device number of a local LSI that an error has occurred can be checked by an interrupt status.

## *13.5 Center LSI Interrupt status*

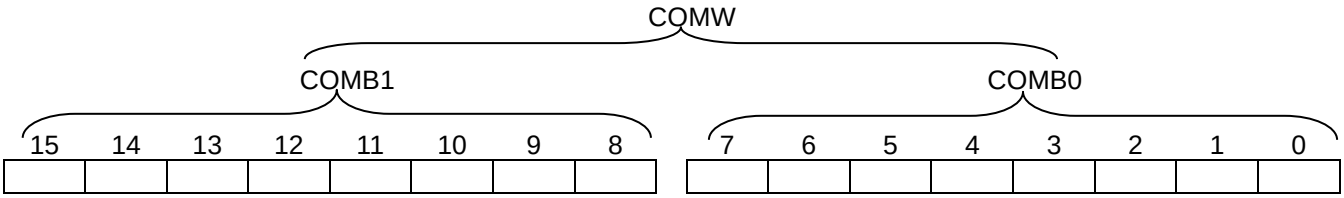
For details, see the user's manual for a center LSI.




| Bit | Symbol | Description |
|---|---|---|
| 5 ~ 0 | EDN | Contains the device number 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.<br>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.<br>This value is stored until an error occurs next time. |
| 11 ~ 8 | ERA | These are identification codes for received data processing errors on a local LSI. The code is stored until an error occurs next time.<br>0001: Received I/O data that is different from the PMD settings.<br>0010: When an I/O device receives a data communication frame.<br>0011: Frames larger than the receiving buffer capacity is received in data communication. |
| 15 ~ 12 | CAE | These are access error codes from a CPU. The code is stored until an error occurs next time.<br>0001: When no local LSIs are used a cyclic communication (I/O communication) start command is written.<br>0010: Try to write data with a start sending command without any data to send.<br>0011: While DBSY = 1, a LSI tries to do one of the followings:<br>(1) Reading or writing to the transmission or receiving FIFO.<br>(2) Wrote a system start command or a data communication start command.<br>0100: Tried to send data to device number 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 8-bit CPU I/F (IF0 = H level, IF1 = H level) in the following order: COMB0 then COMB1.
Note2: # symbol bit: The upper bits of the device number 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<br>(0000h) | NOP<br>Disabled command. |
| 0000 0001 0000 0000<br>(0100h) | Resets by software.<br>Resets a center LSI. This is the same function as RST signal input. |
| 0000 0010 0000 0000<br>(0200h) | Resets the transmission FIFO.<br>Resets only the data transmission FIFO. |
| 0000 0011 0000 0000<br>(0300h) | Resets the receiving FIFO.<br>Resets only the data receiving FIFO. |
| 0000 0100 $$$ 00$$<br>(04xxh) | Command to reset INT system status.<br>$ bits (bits 6, 5, 4, 1, 0) of this command corresponds to the followings.<br>7 6 5 4 3 2 1 0<br>0 CAER ERAE EDTE 0 0 BRKF CEND<br>By setting each bit to 1, correspondent status is reset.<br>However, when RENV0.MCLR = 0, this command is disabled. |
| 0000 0110 0000 0000<br>(0600h) | Command to clear an error counter<br>Reset error count register to 0. |
| 0000 0110 0001 0000<br>(0610h) | Break communication command.<br>When turning off automatic break function by setting RENV0.BKOF = 1, you can perform break communication at arbitrary timing by this command. |
| 0001 0000 0000 0000<br>(1000h) | System communication to all local LSIs.<br>Polls all of the local LSIs (device No. 0 ~ 63) one by one, and refreshes the "device information" areas that correspond to each device number.<br>The "device information" contains the following:<br>- Device in use: 0 when no response, and 1 when a response is returned.<br>- Device type: Refreshes to 1 when it is a LSI that can perform data communication.<br>- I/O setting information |
| 0001 0001 0000 0000<br>(1100h) | System communication to all local LSIs that are excluded from cyclic communication.<br>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 number.<br>The details refreshed are the same as by writing a command 1000h. |
| 0001 0010 00## ####<br>(1200h ~ 123Fh) | System communication to specified local LSIs.<br>Device information area of the local LSIs specified with ###### is revised. The details refreshed are the same as by writing a command 1000h. |

| Command | Description |
|---|---|
| 0001 0011 00## ####<br>(1300h ~ 133Fh) | Obtain attribute information for the specified devices.<br>Attribute information of the local LSIs specified with ###### is copied to the Data reception FIFO. "Device information" area does not change. The details of the Data reception FIFO is as follows.<br>Bits 4 ~ 0: (Number of bytes for the longest piece of data) / 8 − 1<br>Bits 7 ~ 5: Not used (not defined)<br>Bits 15 ~ 8: Device type code<br>(I/O device: 01h, Data device: 81h)<br>Bits 18 ~ 16: Set an I/O port (PMD terminal information when I/O device is selected)<br>Bit 19: Always 0<br>Bits 31 ~ 20: Type of data device (G9003: 000h, G9004A: 001h, G9103A:002h,G9103B: 002h, G9103C:002h) |
| 0010 0ggg cccc cccc<br>(2000h ~ 270Ch) | Broadcast communication<br>Send “cccc cccc” command to the groups specified by “ggg” all at once.<br>Set the group number on GRP0 ~ GRP2 terminals or RENV2.GN bit for G9103C.<br>All 0 ~ 7 groups are intended for ggg=000b<br>Command (“cccc cccc”)<br>0000 0001: Start (STA signal substitute input command to multiple axes)<br>0000 0010: Stop (STP signal substitute input command to multiple axes)<br>0000 0011: Emergency stop<br>(EMG signal substitute input command to multiple axes)<br>0000 0100: Reset local LSIs (SRST command to multiple axes)<br>0000 0101: Latch counter (LTCH command to multiple axes)<br>0000 0110: Stop immediately (STOP command to multiple axes)<br>0000 0111: Decelerate and stop (SDSTP command to multiple axes)<br>0000 1000: Change to FL speed immediately<br>(FCHGL command to multiple axes)<br>0000 1001: Change to FH speed immediately<br>(FCHGH command to multiple axes)<br>0000 1010: Decelerate to FL speed (FSCHL command to multiple axes)<br>0000 1011: Accelerate to FH speed (FSCHH command to multiple axes)<br>0000 1100: Copy pre-register for operation to register (speed change, etc.)<br>(PRESHF command to multiple axes.) |
| 0011 0000 0000 0000<br>(3000h) | Start cyclic communication<br>Start cyclic communication with devices that have a 1 in the "device-in-use" bit in the "device information". |
| 0011 0001 0000 0000<br>(3100h) | Stop cyclic communication.<br>Stop the current I/O communication. |
| 0100 0000 00## ####<br>(4000h ~ 403Fh) | Data communication.<br>Sends data in the transmission FIFO to the devices specified with ######. The data received in response will be stored in the reception FIFO. |
| 0100 0001 0000 0000<br>(4100h) | Cancel data communication<br>Halts the data communication and resets the transmission FIFO.<br>This command will be ignored after the data has been sent. |

13.6.2 Indirect access command

Indirect access commands are used when the CPU can not directly access to the address maps of 512 bytes that are required from the center LSI.
You can use these commands to access to the center LSI only within the range of 8 bytes.

<table>
<tr><th>Command</th><th>Description</th></tr>
<tr><td>0101 0000 0### ##xx<br>(5000h ~ 507Fh)</td><td>Write to the "Device information" area.<br>The contents of the I/O buffer are written into a word in the device information area.<br>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">5000h</td><td>Bits 7 ~ 0</td><td>078h</td><td>0</td></tr>
<tr><td>Bits 15 ~ 8</td><td>079h</td><td>1</td></tr>
<tr><td rowspan="2">5004h</td><td>Bits 7 ~ 0</td><td>07Ah</td><td>2</td></tr>
<tr><td>Bits 15 ~ 8</td><td>07Bh</td><td>3</td></tr>
<tr><td rowspan="2">5008h</td><td>Bits 7 ~ 0</td><td>07Ch</td><td>4</td></tr>
<tr><td>Bits 15 ~ 8</td><td>07Dh</td><td>5</td></tr>
<tr><td rowspan="2">500Ch</td><td>Bits 7 ~ 0</td><td>07Eh</td><td>6</td></tr>
<tr><td>Bits 15 ~ 8</td><td>07Fh</td><td>7</td></tr>
</table></td></tr>
<tr><td>0101 0001 0##x xxxx<br>(5100h ~ 517Fh)</td><td>Write to the "I/O communication error flag" area.<br>The contents of the I/O buffer are written into a word in this area.</td></tr>
<tr><td>0101 0010 0### #xxx<br>(5200h ~ 527Fh)</td><td>Write to the "input change interrupt setting" area.<br>The contents of the I/O buffer are written into a word in this area.</td></tr>
<tr><td>0101 0011 0### #xxx<br>(5300h ~ 537Fh)</td><td>Write to the "input change interrupt flag" area.<br>The contents of the I/O buffer are written into a word in this area.</td></tr>
<tr><td>0101 0100 0### ###&<br>(5400h ~ 547Fh)</td><td>Write to the "port data" area.<br>The contents of the I/O buffer are written into a word in this area.</td></tr>
<tr><td>0110 0000 0### ##xx<br>(6000h ~ 607Fh)</td><td>Read the "device information" area.<br>The contents of the word in this area are copied to the I/O buffer.</td></tr>
<tr><td>0110 0001 0##x xxxx<br>(6100h ~ 617Fh)</td><td>Read the "I/O communication error flag" area.<br>The contents of the word in this area are copied to the I/O buffer.</td></tr>
<tr><td>0110 0010 0### #xxx<br>(6200h ~ 627Fh)</td><td>Read the "input change interrupt setting" area.<br>The contents of the word in this area are copied to the I/O buffer.</td></tr>
<tr><td>0110 0 11 0### #xxx<br>(6300h ~ 637Fh)</td><td>Read the "input change interrupt flag" area.<br>The contents of the word in this area are copied to the I/O buffer.</td></tr>
<tr><td>0110 0100 0### ###&<br>(6400h ~ 647Fh)</td><td>Read the "port data" area.<br>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<br>(5500h) | RENV0 register write command<br>When CPU sets data to input/output buffer and this command is written, the content of the input/output buffer is copied to RENV0 register. |
| 0110 0101 0000 0000<br>(6500h) | RENV0 register read command<br>When this command is written, a value of the RENV0 register is copied to input/output buffer. |
| 0110 0101 0000 0001<br>(6501h) | Error counter read command<br>When this command is written, a value of error counter is copied to input/output buffer. |
| 0110 0101 0000 0010<br>(6502h) | Cycle time register read command<br>When this command is written, a register value of cycle time register is copied to input/output buffer. |
| 0110 0101 0000 0011<br>(6503h) | Recipient address register read command<br>When this command is written, a register value of register for receiving address is copied to input/output buffer. |
| 0110 0101 0000 0100<br>(6504h) | Version information register read command<br>When this command is written, a register value of register for version information is copied to input/output buffer. |

Note: The above registers are not allocated on the address map.
Therefore, please access by using the above commands.

## *13.7 Center LSI registers*

### 13.7.1 Environmental register (RENV0)

| 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 |

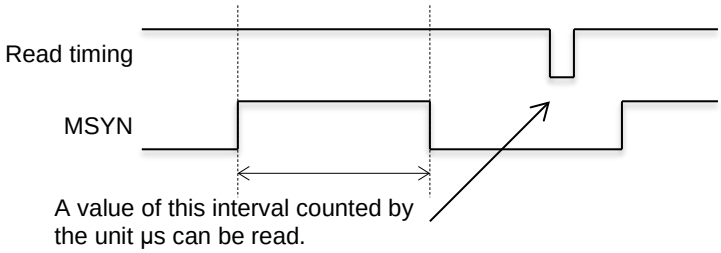
| Bit | Bit name | Description |
|---|---|---|
| 0 | MCED | 1: Masks CEND interrupt. (Status changes.) |
| 1 | MBRK | 1: Masks BRKF interrupt. (Status changes.) |
| 2 | MIOP | 1: Masks IOPC interrupt. (Status changes.) |
| 3 | MEIE | 1: Masks EIOE interrupt. (Status changes.) |
| 4 | MEDE | 1: Masks EDTE interrupt. (Status changes.) |
| 5 | MERE | 1: Masks ERAE interrupt. (Status changes.) |
| 6 | MCSE | 1: Masks CAER interrupt. (Status changes.) |
| 7 | - | Set always "0". |
| 8 | BKOF | 0: Confirm a break request at a certain time intervals (250ms at 20Mbps.)<br>1: Turn off automatic break function. |
| 9 | MCLR | Select the way to clear the following status bits.<br>CEND, BRKF, EDTE, ERAE, CAER<br>0: These status bits are reset by reading status. (default)<br>1: These status bits are not reset by reading status. These status bits are reset by reset command (04xxh) of interrupt status. |
| 15 ~ 10 | - | 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 such as no response and CRC errors.
When error counts exceed more than 65534 times, the counter stops at 65535.
To clear the counter, issue an error counter clear command (0600h).
Please note that the G9001A counts no response to system communication as an error. (Not-response to system communication is not handled as an error, basically.)
All bits are 0 after reset.

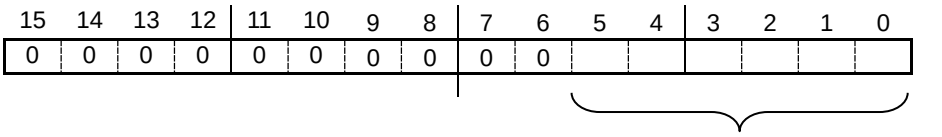
### 13.7.3 Cycle time register (RSYCNT)

This is a 16-bit register for measuring cycle time.
It counts the time when MSYN signal keeps change in µs. It is a read-only register.
It always counts them, and you can refer a count before the last MSYN signal changes.
The upper limit of the count is 65535 (approximate 65.5 ms). The width of the MSYN signals more than the limit cannot be measured.



13.7.4 Receiving address register (RDJADD)

This register retains the device number of a local LSI that has received normal data communication finally.
This is read-only register.
All bits are 0 after reset.




13.7.5 Version information register (RVER)

You can check the version information of a center LSI. This is read-only register.
This register is always the following value (0001h).

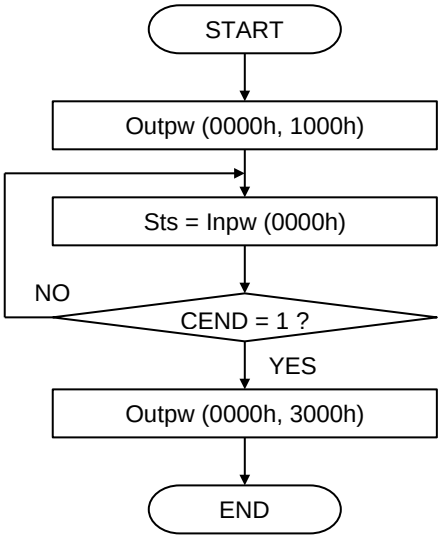
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

This register is used to distinguish between G9001 and G9001A by software.
The process is as follows.
1. Write "0000h" to I/O buffer.
2. Issue Read command (6504h) of RVER register.
3. If data that is read from I/O buffer is "0000h", G9001 is used. If the data is "0001h", 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 the system communication command for all of the local LSIs.
- Read status.
- Wait for completion of the system communication. Until the system communication has been completed, operation can not shift to the next one. This process can also wait an interrupt.
- Start cyclic communication.

13.8.2 Communication with port data area

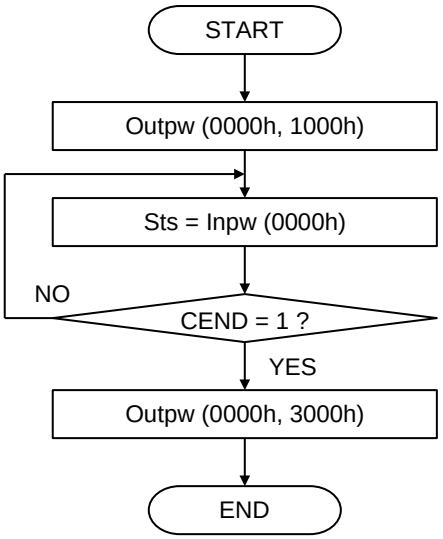
This section describes how to get data I/O port information on the I/O device (G9002A), and a status of a data device (G9103C).
Assume that the local LSIs to be used are as follows:
The example of the G9103C is one of only read status.

| LSI type | Item to configure | Configuration data | Output data |
|---|---|---|---|
| G9002A | Device number | 2 | - |
| | Port 0 | Input | - |
| | Port 1 | Output | 12h |
| | Port 2 | Output | 34h |
| | Port 3 | Output | 56h |
| G9103C | Device number | 5 | |

Note: The port area configuration of the G9103C 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) |

When whole address map can be used



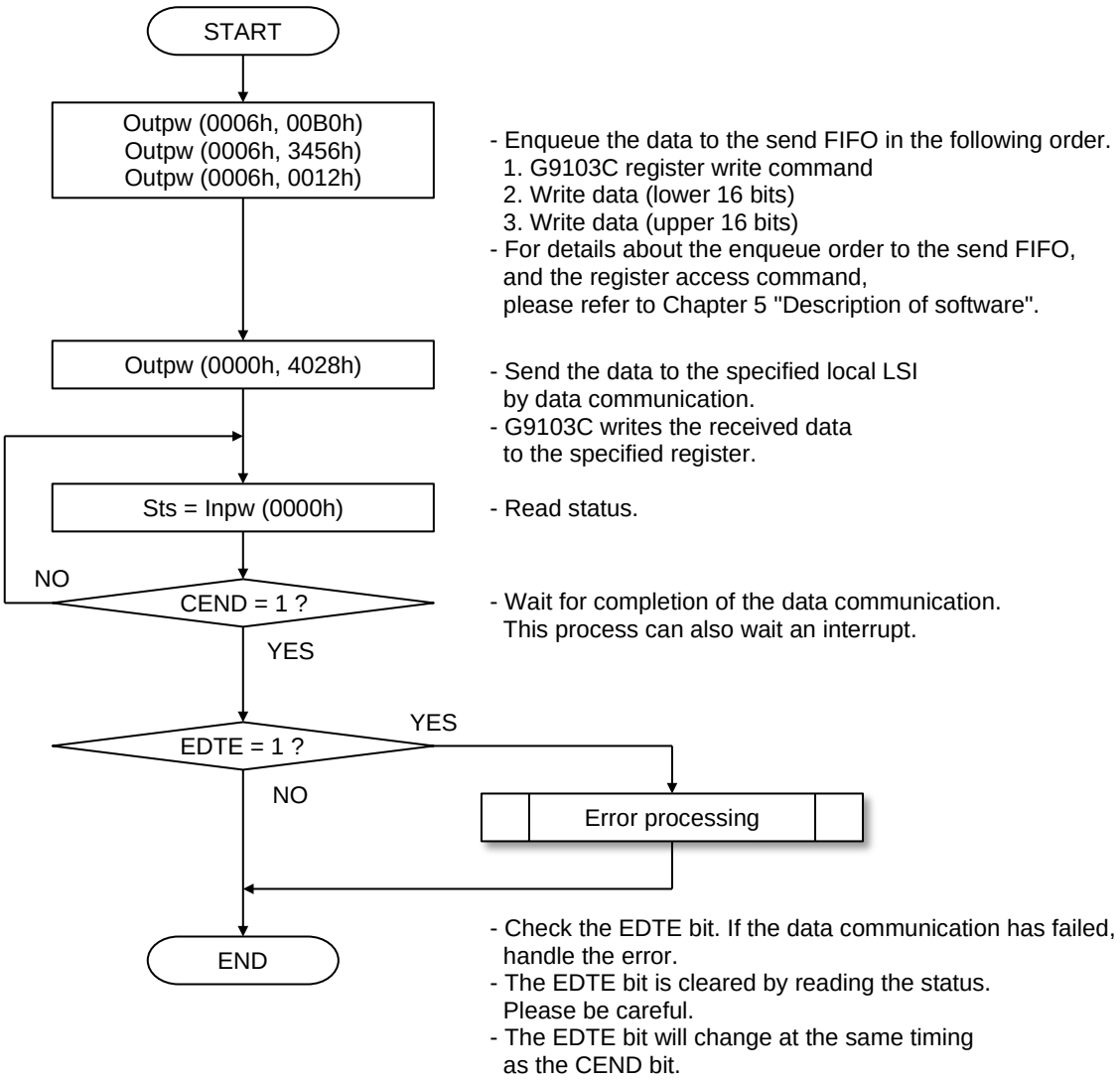

- Send the system communication command for all of the local LSIs.
- Read status.
- Wait for completion of the system communication. Until the system communication has been completed, operation can not shift to the next one. This process can also wait an interrupt.
- Start cyclic communication.

13.8.3 Data communication

13.8.3.1 Write register.

The data communication example below shows data being placed in a register that is integrated in the G9103C.
Assume that the local LSIs to be used are as follows.
Write "00123456h" in "PRMV" register of the G9103C.

| LSI type | Configuration item | Device number |
|---|---|---|
| G9103C | Device number | 40 (28h) |




A data communication command is determined as follows:

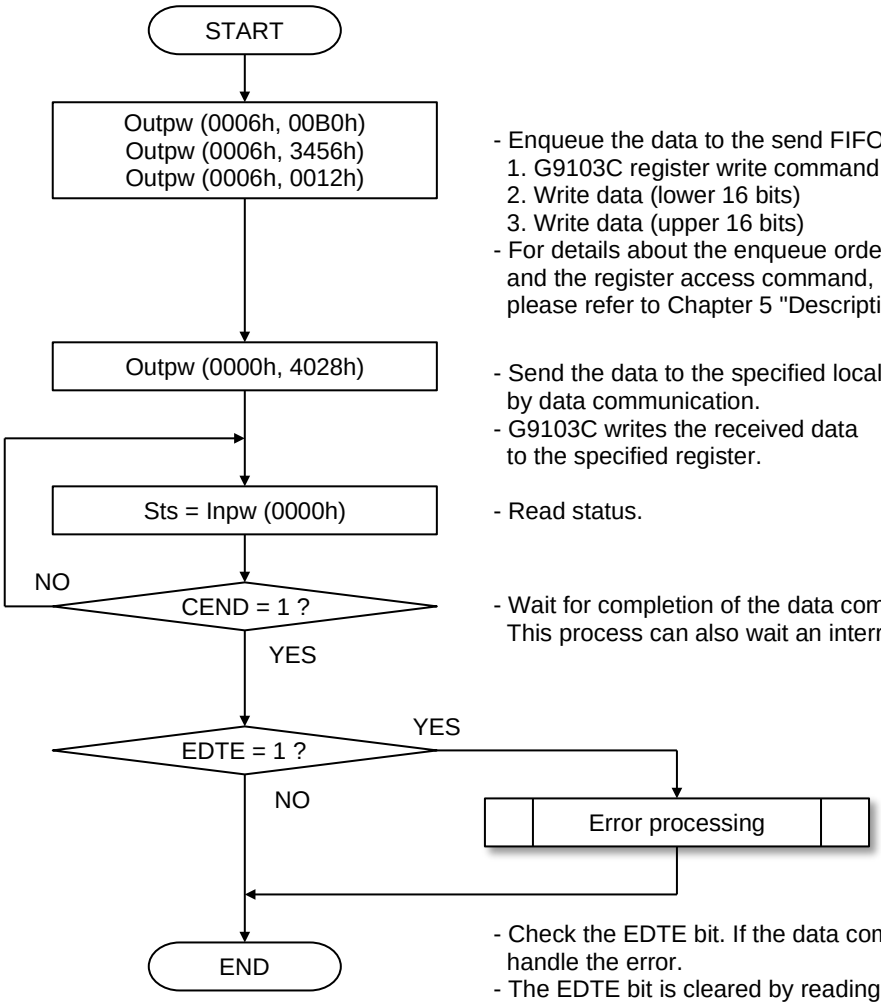
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | # | # | # | # | # | # |

Specify device number in these bits.

13.8.3.2 Read registers.

Assume that the local LSIs to be used are as follows.
Read a “RMV” register value.

| LSI type | Configuration item | Configuration data |
|---|---|---|
| G9103C | Device number | 40 (28h) |




- Enqueue the data to the send FIFO in the following order.
  1. G9103C register write command
  2. Write data (lower 16 bits)
  3. Write data (upper 16 bits)
- For details about the enqueue order to the send FIFO, and the register access command, please refer to Chapter 5 "Description of software".

- Send the data to the specified local LSI by data communication.
- G9103C writes the received data to the specified register.

- Read status.

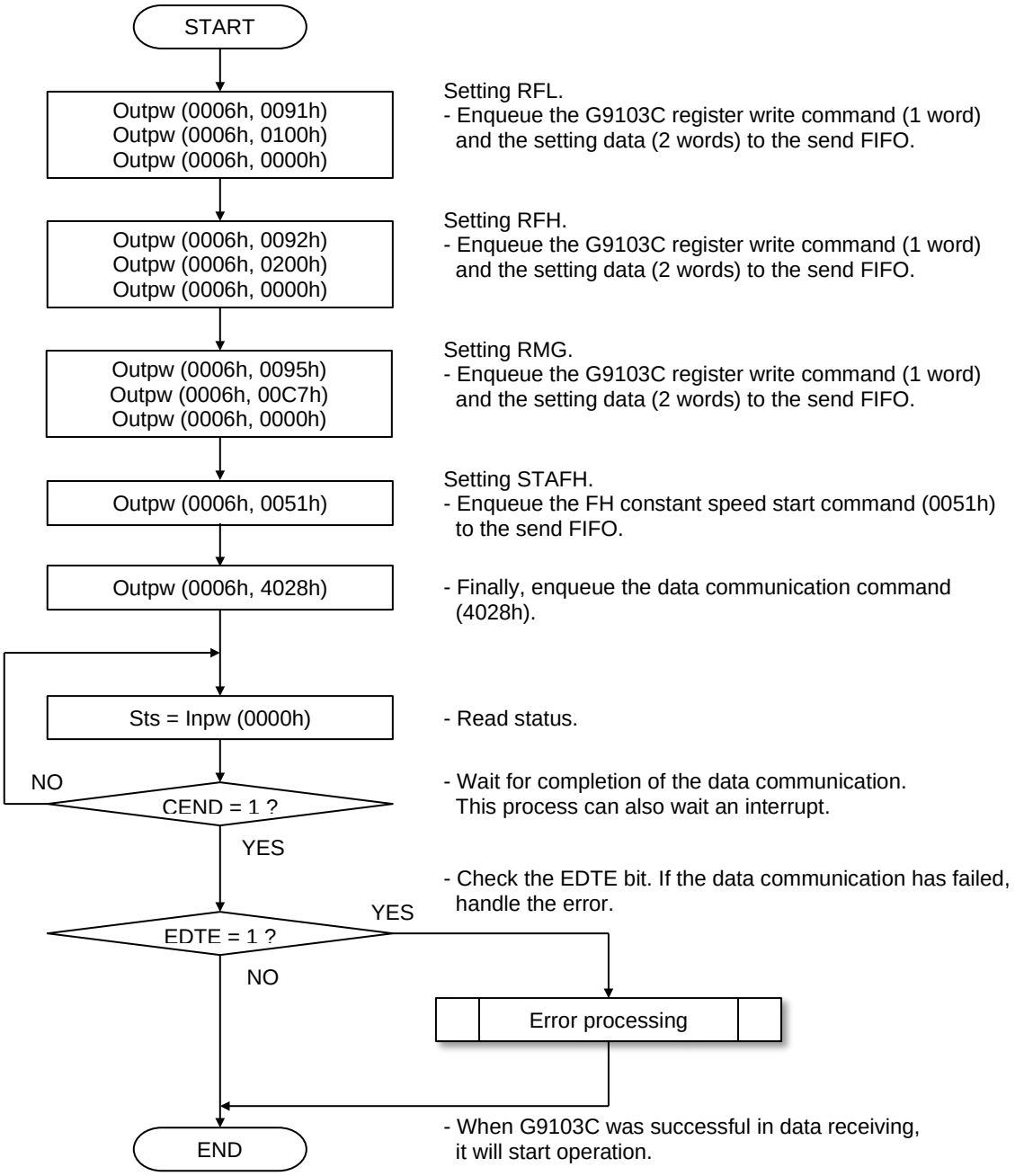
- Wait for completion of the data communication. This process can also wait an interrupt.

- Check the EDTE bit. If the data communication has failed, handle the error.
- The EDTE bit is cleared by reading the status. Please be careful.
- The EDTE bit will change at the same timing as the CEND bit.

13.8.3.3 Starting operation mode

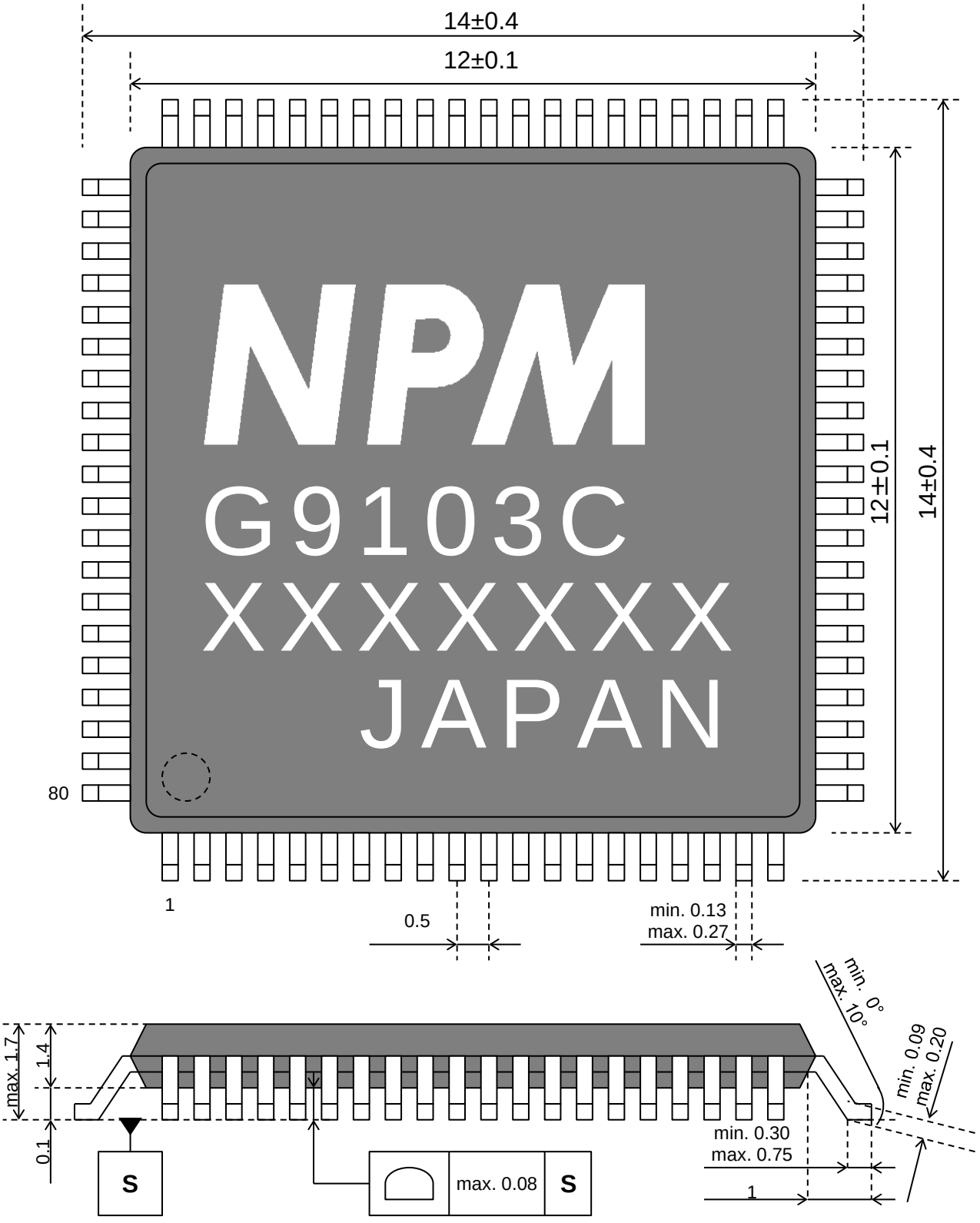
As an example of data communication, the following shows an example to set various register of G9103C and start pulse output.
The data that is specified to G9103C is as follows. (This is a simple data to output pulses.)

| Register name | Set value | Command | Remarks |
|---|---|---|---|
| RFL | 00000100h | 0091h | |
| RFH | 00000200h | 0092h | |
| RMG | 00C7h | 0095h | Multiplication rate = 1 |



## ***14. External dimensions***

80-pin, LQFP, Unit: mm



## 15. Designing Precautions

The following precautions are described in context. The precautions need attentions especially are shown below again.

### 15.1 Precautions for hardware design

1) All signal terminals have TTL level (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 with 5V interface 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) When setting multiple registers at once in one communication, set three-words per register.

2) Please set operation complete timing to “When output cycle is complete (RMD.METM = 0)” if the next operation starts automatically using the pre-register. If you select “When pulse width is complete (RMD.METM = 1)”, the interval between the last pulse and the initial pulse of the next operation become narrow (750ns).

3) When interpolation operation block is executed continuously, the start timing between each block on each G9103C is not synchronized. Each G9103C controls the start timing on the assumption that operation times of each G9103C the same. Therefore, interpolation operation cannot be used with the functions that operation time is different among G9103Cs such as backlash correction, vibration restriction function, direction change timer, and delay by INP signal input.

4) Make sure to use the clock synchronizing function for motor control in the system for interpolation operation. A center LSI has two terminals SIA and SIB as serial input terminals and the communication line can be separated into two lines. However, G9103C of the interpolated axis should not be separate into two lines. Be sure to set RSYN register of a clock master LSI first. For other G9103s, any order is available.

5) If you use a sensor substitute function by I/O device, make sure to specify a value other than zero in RSYN2.SWME bit and set to operate error stop when communication cannot be monitored.

### 15.3 Mechanical precautions

If the processing when EL signal (+EL signal in (+) direction and –EL signal in (−) direction) turns ON is to decelerate and stop (RENV1.ELM = 1), mechanical allowance is needed because a motor stops after a mechanical position passes over EL position.

## 16. Handling Precautions

### 16.1 Design precautions

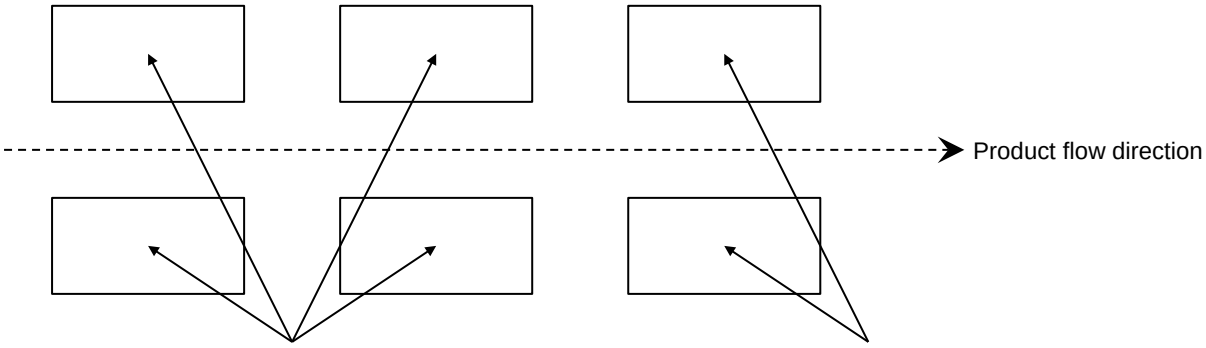
1) Never exceed the absolute maximum ratings, even for a very short time.
2) Take precautions against the influence of heat in the environment, and keep the temperature around the LSI as cool as possible.
3) Please note that ignoring the following may result in latching up and may cause overheating and smoke.
   - Do not apply a voltage greater than +5.5 V 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 5 V interface, make the time difference between turning on and off 3.3 V and 5 V 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 3.3 V power supply from 5 V power by a regulator IC or DC-DC converter.)

### 16.2 Precautions for transporting and storing LSIs

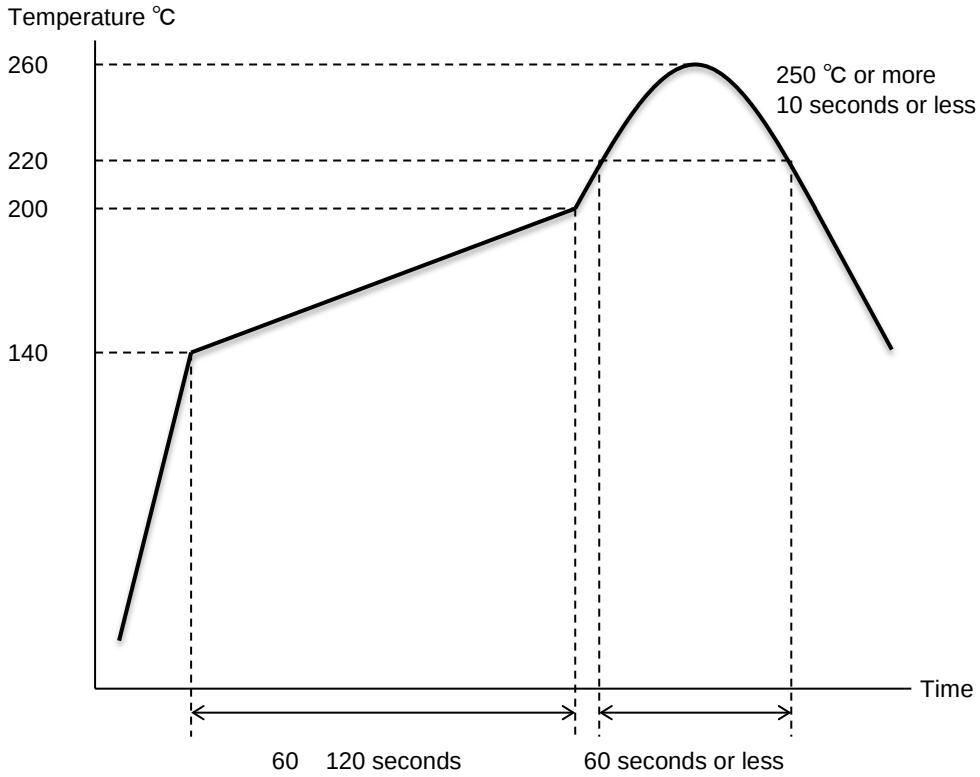
1) Always handle LSIs and their packages carefully. Throwing or dropping LSIs may damage them.
2) Do not store LSIs in a location exposed to water droplets or direct sunlight.
3) Do not store the LSI in a location where corrosive gases are present, or in excessively dusty environments.
4) Store the LSIs 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 cause). 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°C for 20 ~ 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.




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 reflow soldering, the restrictions are the same as for infrared reflow.
8) If you will use a soldering 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.

# *17. Difference from G9103B*

In the G9103C, many functions were added and improved and defects were corrected. If you use the same functions as G9103B, you can mount a G9103C on a board for G9103B and use software for G9103B.

## *17.1 Added functions*

The following four functions were added to the G9103C.

### 17.1.1 Target of comparator 3 for comparison

As target of comparator 3 for comparison, remaining pulse counter (RPLS or RCIC) was added.
For details, please refer to “8.11.1 Comparator types and functions.”

### 17.1.2 ID code confirmation

To distinguish between G9103B and G9103C, ID code is added.
For details, please refer to “8.20 ID code confirmation”.

### 17.1.3 Start command for next operation

Start commands for next operation (NSTAFL, NSTAFH, NSTAD, NSTAUD) that occur error interrupt while motor is stopping and error interrupt cause (REST.ENST) were added.
At the continual start of interpolation operation, if a write command to either axis does not make it while operating, simultaneous stop can be occurred by error interrupt. That prevents from motor running with simultaneous start failing.
For details, please refer to “5.3.1.1 Start command” or “5.4.1.1 Writing into pre-register for operation”, and ”8.16 Interrupt output”.

### 17.1.4 Remaining start in linear interpolation operation and circular interpolation operation

Residual start command that could be used only in single axis positioning operation was added to multi axes linear interpolation operation mode and circular interpolation mode.
For details, please refer to “5.3.1.1 Start command”, “5.4.3.5 RCI (PRCI): Setting register for number of steps necessary to complete circular interpolation (31 bits)” and “5.4.3.6 RCIC: Step counter for circular interpolation (31 bits)”.

## *17.2 Improved functions*

The following four functions are improved in the G9103C.

### 17.2.1 Improvement of timing to check error stop cause

The following phenomenon had occurred in G9103B.

1. Set “holding start” to both own axis and other axis by “writing a start command”.
2. Set “Simultaneous stop by synchronized with other LSIs”.
3. In the case that EMG signal turns on and you write a start command to own axis, the own axis stops by error with detecting “error stop cause” and sends simultaneous stop request to other axes.
4. Other axes receive simultaneous stop request, however ignore the request because they are stopping.
5. After that, you write a start command to other axes, the status of other axes become “holding starts”.
6. If you write broad communication start command to own axis and other axes, only other axes starts operation.

Therefore, before writing broad communication start command, you need to confirm whether there are not axes that stop by error using check all axis status.
G9103C has been improved so that “error stop cause” is checked at operation start.
As the result, G9103C sends a “simultaneous stop request” by error stop after writing broad communication start command, and other axes in holding start receive “simultaneous stop request” while running and stop simultaneously.
“Error stop cause” can be checked at writing start command like G9103B by setting a function to change check timing of error stop cause” is enabled (RENV3.G03B = 1).

### 17.2.2 Improvement of RSDC sign extension

In the G9103C, the upper 8 bits of RSDC register are fixed to 0.
Therefore, Attentions are required not as to misidentify 24-bit negative number as 32-bit positive number at developing software. G9103C was corrected so that the upper 8 bits of RSDC register is the same

specifications of code extension as RDP register. As the result, 24-bit negative number can be regarded as a negative number in 32-bit.

#### 17.2.3 Improvement of pre-register for comparator 3

The following phenomenon had occurred in G9103B.

1. Make PRCP3 register and RCMP3 register undetermined by PFCCAN (002Dh) command.
2. If writing data to PRCP3 register, written data is shifted from PRCP3 register to RCMP register and data in RCMP2 register become determined.
3. If original RCMP3 register is true, it change by written data and return to undetermined.
4. If writing new data to RCMP3 register at that time, the new data is shifted from PRCP3 register to RCMP3 register because the data in RCMP3 register is undetermined.

G9103C was improved to prevent that written data is not shifted from PRCP3 register to RCMP3 register even if PRCMP3 register and RCMP3 register are undetermined, data is written to PRCP3 register and comparison condition is changed by written data.
As the result, even if new data is written in PRCP3 register, new written data is not shifted from PRCP3 register to RCMP3 register and RCMP register can keep original data as determined.

#### 17.2.4 Improvement of pre-register for continuous operation

G9103B copies pre-register value to register after stopping and starts operation only when a start command is written while a motor is running. However, writing a start command for next operation delays and the timing is after a motor stops, a pre-register value is not copies and G9103B starts with a register value.
G9103C was improved so as to copy pre-register value to register regardless of writing a start command when pre-register is written while a motor is running.
As the result, even if writing a start command for next operation delays and the timing become after a motor stops, G9103C starts operation after it copies a re-register value to register.

#### 17.2.5 Improvement of start command with feeding amount

When writing a start command with feeding amount that is a G9003 compatible function while a motor is running, G9103B overrides the feeding amount (RMV register) and may malfunction.
G9103C was corrected to ignore writing when writing a start command with feeding amount while a motor is running.
As the result, RMV register is not overridden by a start command with feeding amount while a motor is running and the G9103C does not malfunction.

#### 17.2.6 Improvement of monitoring communication error

In G9103B, when a count number of monitoring communication error (RMEC.CKEC, RMEC.SPEC, RMEC.SWEC) is 0 and writing 0 after writing other than 0 as the maximum times of continuous error occurrence, error stop occurs even if number of occurrences of monitoring communication error is 0.
G9103C was improved to ignore satisfied conditions even if the number of occurrence of monitoring communication error matches a written value at writing the maximum number of continuous error
As the result, error stop does not occur when number of occurrence of monitoring communication error is 0.

### *17.3 Corrected defects*

The following defect was corrected in G9103C.

#### 17.3.1 Set error stop method to stop after decelerate in Linear interpolation and circular interpolation

G9103C was corrected so as to stop after deceleration as error stop even if “to decelerate and stop” is selected as error stop method in linear and circular interpolation.

## 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