

***Basic of PCD Series
Pulse Control LSIs***

Table of Contents

1. What is a PCD?	1
2. Reviewing common terms	1
(1) Reference clock	1
(2) Operating patterns and registers	1
(3) Commands	2
(4) Writing and reading data	2
3. What types of operating patterns are available?	3
<Basic operating patterns>	3
<Examples of operating patterns>	3
(1) Immediate stop operation	3
(2) Deceleration stop operation	3
(3) Preset operation (positioning operation)	4
(4) Zero return operation	4
4. How is data exchanged with a CPU?	5
(1) \overline{CS} (Chip Select)	5
(2) \overline{WR} (Write)	5
(3) \overline{RD} (Read)	5
(4) A0 to A3 (Address lines)	5
(5) \overline{INT} (Interrupt Request)	5
(6) D0 to D7 (Data lines)	6
5. What commands are available?	7
(1) Start Mode command	7
(2) Control Mode command	8
(3) Register Select command	8
(4) Output Mode command	8
6. What registers are available?	9
(1) R0: Preset amount register	10
(2) R1: FL speed register	10
(3) R2: FH speed register	10
(4) R3: Acceleration/deceleration rate register	10

(5) R4: Magnification rate register	11
(6) R5: Ramping-down point register	12
(7) R6: Idling pulse register	12
7. How is the motor driver connected to mechanical systems?	13
(1) $\overline{+PO}$, $\overline{-PO}$ (Pulse Out)	13
(2) OTS (Output Signal)	13
(3) \overline{ORG} (Origin)	14
(4) $\overline{+SD}$, $\overline{-SD}$ (Ramp-down)	14
(5) $\overline{+EL}$, $\overline{-EL}$ (End Limit)	14
8. What is the Status byte?	15
9. Terminology	16
10. The following tools are available	18
<Reference documents>	19
1. What kinds of applications can the PCD series be used for?	19
2. Basic digital circuits and stepper motor operation	19
(1) What are high and low signal levels?	19
(2) Signals (terminal names) with a bar over the name	19
(3) Terms used with stepper motors	20
1) Self-start frequency	20
2) Unipolar/bipolar	20
3) 2-phase/5-phase	21
4) Full step (2-2 phase excitation)/half step (1-2 phase excitation)	21

Basic of PCD series pulse control LSIs

This document is intended for customers who will use our LSIs specifically to control motors for the first time. It explains operations in terms of the simplest NPM LSIs, the "PCD series."

1. What is a PCD?

- A PCD divides a string of reference clock pulses and creates a variety of pulse strings at different frequencies that are appropriate for controlling specific motors.
- By sending operating patterns and commands from a CPU, you can leave the control of a motor up to the PCD.
=> This decreases the overhead imposed on the CPU.
- This PCD series provides most of the functions required to control stepper motors.
- It also has an excitation sequence output function used with 2-phase stepper motors.
- You can select a suitable model for the number of axes you need to control (1 axis: PCD4500, PCD4511, 2 axes: PCD4521, 4 axes: PCD4541).

2. Reviewing common terms

(1) Reference clock

This is the basic frequency used when operating the PCD. => Similar to CPUs, LSI operation is also based on a reference clock.

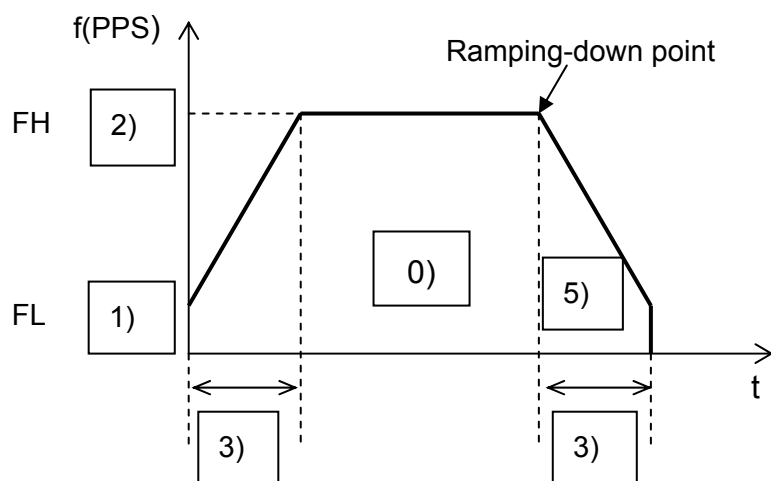
CPUs used in general equipment do not need to be as fast as the CPUs in personal computers.

(Lower speeds are fast enough.)

The PCD series recommends using a reference clock frequency of 4.9152 MHz.

=> Although 4.9152 is a fractional value, the pulse string frequencies will not be fractional values and fractions to the right of the decimal can be eliminated.

(2) Operating patterns and registers



<Figure 1>

Usually, six values are used to control a motor, as shown in the figure 1.

- 0): Feed amount (preset amount) = Number of pulses
- 1): Self start frequency (FL speed)
- 2): Frequency when running at high speed (FH speed)
- 3): Acceleration/deceleration time
- 4): Magnification rate
- 5): Ramping-down point (number of pulses after beginning deceleration until the motor operation is stopped)

The areas where these values are stored are referred to as "Registers."

(3) Commands

These are instructions telling the motor what to do.

For example, the following commands may be used.

- 1) "Start operation by accelerating from the self start frequency."
- 2) "Operate at the self start frequency and do not accelerate."
- 3) "Do not start until an external signal is received."
- 4) "When an operation is finished, send an Operation Complete signal to the CPU."
- 5) "Stop immediately, even if running."
- 6) "Start deceleration immediately and stop when the motor speed reaches the self start frequency."
- 7) "Run the motor counter clockwise."
- 8) "Accelerate and decelerate using an S-curve."
- 9) "Stop operation immediately if a zero position signal is received."
- 10) "Mask the output pulse signal."
- 11) "Output an excitation sequence signal, not pulses."
- 12) "Hold speed."

(4) Writing and reading data

The CPU writes to registers and sends commands.

Meanwhile, the CPU also reads the current PCD status (it retrieves the settings and determines the current status). The following read operations may be used.

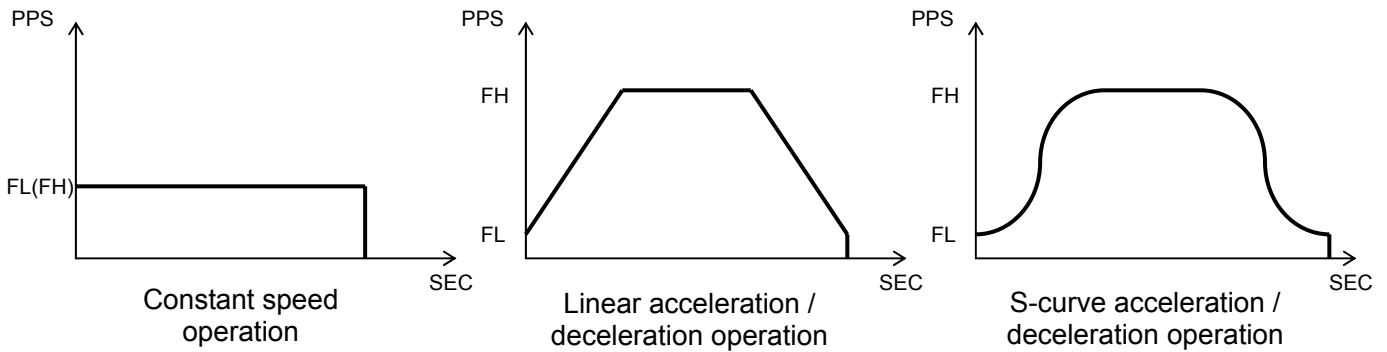
- 1) "What is the current PCD self start frequency?"
- 2) "Is the PCD running or stopped?"
- 3) "Is the PCD accelerating or operating at rated speed?"
- 4) "Has an external signal been received?"

3. What types of operating patterns are available?

<Basic operating patterns>

The following basic PCD series operating patterns are available.

In addition to "Constant speed operation" from start to stop, the PCD series can accelerate and decelerate a motor according to a specified linear rate or by using S-curves.

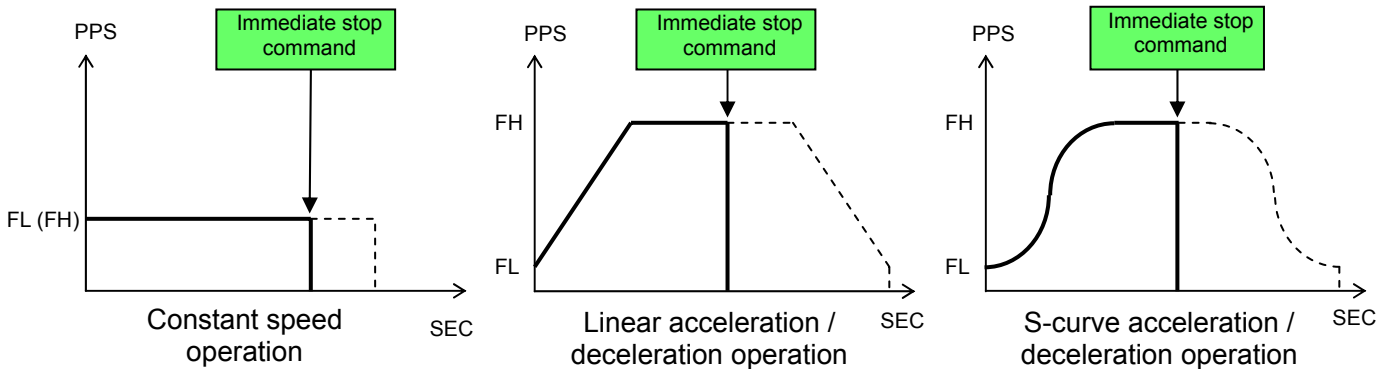


<Figure 2>

<Examples of operating patterns>

(1) Immediate stop operation

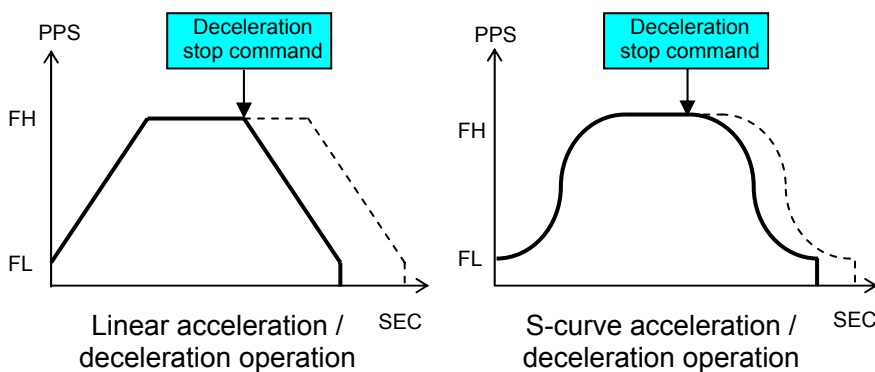
When an immediate stop command or immediate stop signal is received while operating, the PCD will stop operation regardless of its current state.



<Figure 3>

(2) Deceleration stop operation

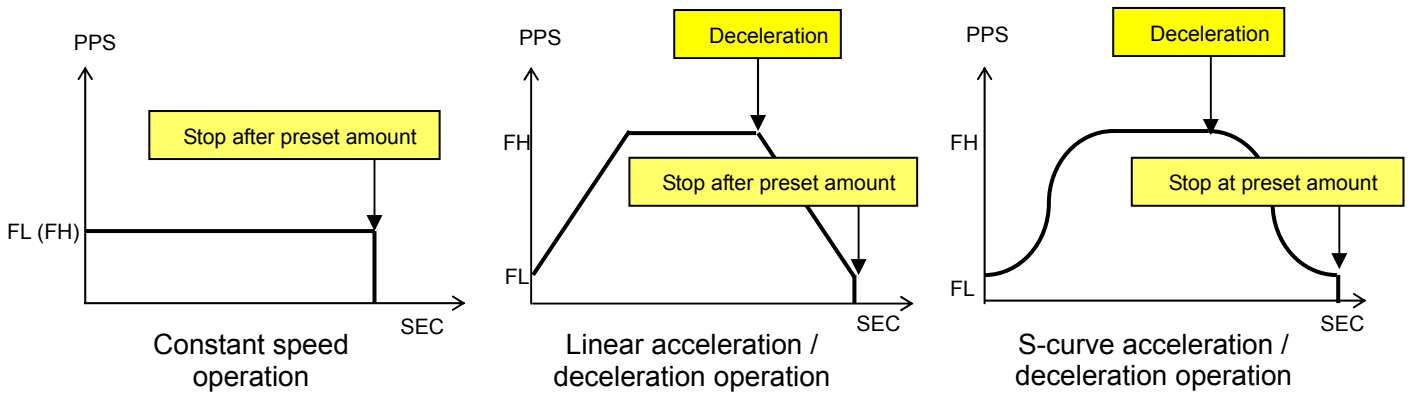
When a deceleration stop command or deceleration signal is received while operating, the motor will start decelerating and stop when it reaches the self start speed.



<Figure 4>

(3) Preset operation (positioning operation)

The PCD can stop after feeding a specified number of pulses or a specified number of rotations.

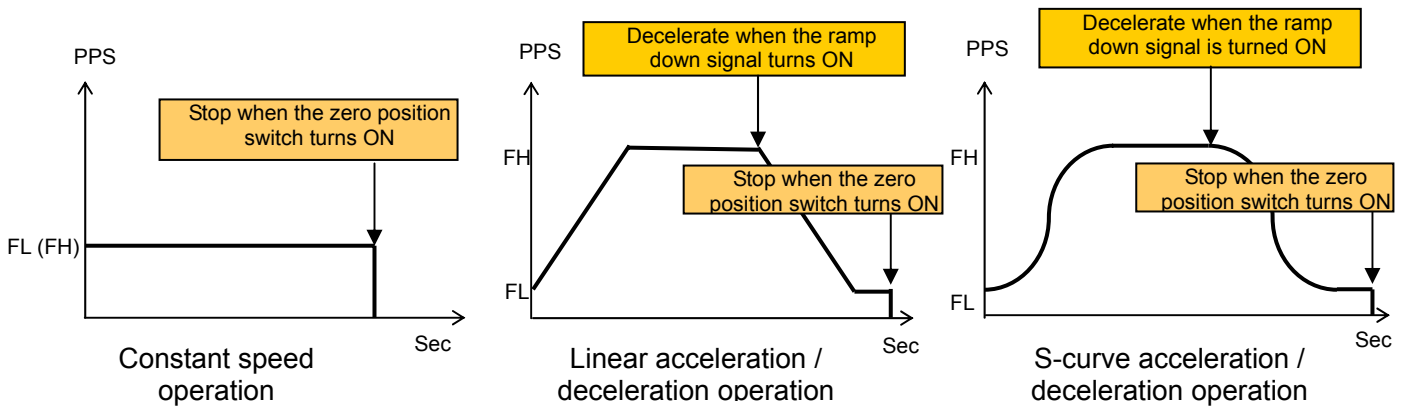


<Figure 5>

(4) Zero return operation

The PCD can stop when an ORG (zero position) switch at the zero position turns ON.

When you want to reduce the shock while stopping during an acceleration/deceleration operation, start decelerating the motor with the SD (ramp-down) switch. After the motor speed has dropped, turn ON the ORG switch.



<Figure 6>

* The PCD4500 cannot use S-curves for acceleration or deceleration.

*** From this point on, you may also want to refer to the user's manual for the PCD4500 or PCD4511/4521/4541, in order to gain a deeper understanding.**

4. How is data exchanged with a CPU?

First, a description of the PCD terminals that are connected to a CPU.

(1) \overline{CS} (Chip Select)

The chip select signal is used to specify with which device the CPU will be communicating, such as a PCD or memory. When this signal is LOW, the CPU can exchange data with the PCD.

(2) \overline{WR} (Write)

To write a command or send register data to the PCD, this signal line is brought LOW, together with the \overline{CS} line. Then, the PCD will read the data placed on terminals D0 to D7 by the CPU (described in item (6) below).

(3) \overline{RD} (Read)

To read the current PCD status or to get specific values from the PCD, the CPU will make this signal LOW, together with the \overline{CS} line. Then, the PCD will place data on terminals D0 to D7 for the CPU to read (described in item (6) below).

(4) A0 to A3 (Address lines)

These are address line terminals. Various combinations of HIGH/LOW on these four terminals determine which of the following operations is selected.

- Write a command
- Write data to a register
- Read the PCD status or data

With the PCD4521/4541, these lines are also used to select an axis for writing or reading data.

(5) \overline{INT} (Interrupt Request)

This is an interrupt signal sent to the CPU to tell the CPU when the PCD has finished its positioning operation, stopped operation for any reason, or has started to decelerate.

This signal causes the CPU to interrupt its current task. When the CPU receives an \overline{INT} signal, it takes action according to the cause of interrupt. For example, when the X-axis finishes operation, the CPU is notified by an interrupt signal and could then output a start signal to Y-axis.

(6) D0 to D7 (Data lines)

These are data bus signal lines. These eight terminals are used to exchange data (writing/reading) with the CPU. This is referred to as an 8-bit data bus.

Each of these terminals is "1" when ON, and "0" when OFF. For example, the following signal levels are translated as 10011100b (9Chex).

D7	D6	D5	D4	D3	D2	D1	D0
H	L	L	H	H	H	L	L
1	0	0	1	1	1	0	0

<Table 1>

The data bus is only 8 bits wide. So, how can you read or write 9 bits or more of data?

=> Data is written in 8-bit groups, by separating 8 bits at a time, counting from the lowest bit. For example, suppose you want to write the decimal number 1,000,000 into a 24-bit register called R0. If you look at reference documents table 8, the decimal/binary/hexadecimal comparison table, you will see that this is a 20-bit binary number. When expressed as 24 bits, it will look like the binary number shown below.

0000 1111 (0Fh) 0100 0010 (42h) 0100 0000 (40h)
(Upper 8 bits) (Middle 8 bits) (Lower 8 bits)

The order for writing this data is to write the upper 8 bits first, then the middle 8 bits, and finally the lower 8 bits.

* Either the upper 8 bits or middle 8 bits can be written first. However, the lower 8 bits must be written last, due to the internal architecture of the PCD.

- Table 2 below sums up the relationship described above.

\overline{CS}	\overline{RD}	\overline{WR}	A1	A0	Description
L	H	L	L	L	Place a command on the D0 to D7 data lines (write).
L	H	L	L	H	Place the lower 8 bits of register data on the D0 to D7 data lines (write).
L	H	L	H	L	Place the middle 8 bits of register data on the D0 to D7 data lines (write).
L	H	L	H	H	Place the upper 8 bits of register data on the D0 to D7 data lines (write).
L	L	H	L	L	Place zeroes on the D0 to D7 data lines (write).
L	L	H	L	H	Fetch the lower 8 bits of internal PCD data from the D0 to D7 data lines (read).
L	L	H	H	L	Fetch the middle 8 bits of internal PCD data from the D0 to D7 data lines (read).
L	L	H	H	H	Fetch the upper 8 bits of internal PCD data from the D0 to D7 data lines (read).
L	L	L	x	x	The setting shown on the right is prohibited.
H	x	x	x	x	Disable reading and writing on the D0 to D7 data lines.

Boxes marked with an "x" mean the L or H status for that line is ignored.

<Table 2>

5. What commands are available?

Commands consist of 8 bits. The commands understood by the PCD series are grouped into four categories, which are quite simple.

In the PCD series, a role is assigned to each bit. (Changing any bit to 1 will change the command.)

The address used for writing commands is $A(1:0) = 11$.

(1) Start Mode command

This command is used to indicate the type of start or stop.

The instructions in items 1) to 6) of section 2, subsection (3) above explain how to write a Start Mode command.

If you want an \overline{INT} signal to be output when the PCD stops, the CPU must also write this command.

The Start Mode command is always 00b, placed on data lines D6 and D7.

For other examples of this command and descriptions of the role of each bit, see the user manual for the PCD series (pages 32 to 34 for the PCD4511/4521/4541).

* When a Start Mode command is written to the PCD, it will start operation immediately. In practice, write each of the commands and register data described below first. Then, write the Start Mode command last.

(2) Control Mode command

This command is used to specify conditions in advance, such as whether or not to stop when the $\overline{\text{ORG}}$ signal (described later) turns ON, whether or not to go to a certain position (preset operation), or whether or not to apply S-curve acceleration/deceleration.

The instructions in items 7) to 9) of section 2, subsection (3) explain how to write a Control Mode command.

The Control Mode command is always 10b, placed on data lines D6 and D7.

For other examples of this command and descriptions of the role of each bit, see the user manual for the PCD series (pages 34 to 36 for the PCD4511/4521/4541).

(3) Register Select command

This command is used to select the register to which you want to write data (described later) and whether or not to output an $\overline{\text{INT}}$ signal at the deceleration start point.

This command is also used to enable/disable the preset counter (described later) for positioning operations (preset operation).

The Register Select command is always 01b, placed on data lines D6 and D7.

For other examples of this command and descriptions of the role of each bit, see the user manual for the PCD series (pages 37 to 38 for the PCD4511/4521/4541).

(4) Output Mode command

This command is used to select whether or not to output pulses and whether or not to halt acceleration/deceleration.

The instructions in items 10) to 12) of section 2, subsection (3) explain how to write an Output Mode command.

The Output Mode command is always 11b, placed on data lines D6 and D7.

For other examples of this command and descriptions of the role of each bit, see the user manual for the PCD series (page 39 to 40 for the PCD4511/4521/4541).

6. What registers are available?

The PCD series has only seven common registers (the PCD4541 has eight registers). This means that operation is easy and simple. The register names are prefixed with an "R": R0 to R6 (R0 to R7 for the PCD4541).

The number of bits that each register can handle varies from 1 to 24 bits.

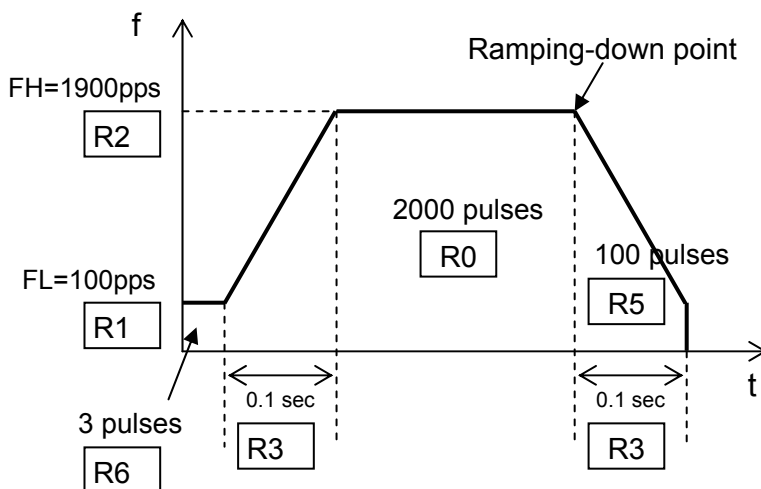
To write data to a register, follow the procedures below.

- Select the register using the register select command.
- Write the data.

- When you want to write more than 8 bits of data, write the data as follows.

- 1) A(1:0) = 11: Write the 8 upper bits of data from the CPU.
- 2) A(1:0) = 10: Write the 8 middle bits of data from the CPU.
- 3) A(1:0) = 01: Write the 8 lower bits of data from the CPU.

- As an example, let's examine the operation pattern in figure 7 below.



<Figure 7>

R0: Specify a feed amount (preset amount) (= number of pulses); 2000

R1: Specify an initial speed; 50

R2: Specify an operation speed; 950

R3: Specify the acceleration/deceleration time: 546

R4: Specify the magnification rate; 300

R5: Specify the ramping-down point (number of pulses left when deceleration starts); 100

R6: Specify the number of idling pulses; 3

(1) R0: Preset amount register (24 bits (18 bits for the PCD4500))

This register is used to specify the feed amount (preset amount). The range is 0 to 16,777,215 pulses, in decimal notation (0 to FFFFFFFh).

On the PCD4500, the range is 1 to 262,143 pulses (1h to 3FFFFh).

The "Positioning pulse range" shown in the user's manual specifications corresponds to this value. For example, enter 2000 (0000 0000 0000 0111 1101 0000b (= 7D0h)) as 24 bits of data. When the motor is started, the counter will decrement the number of remaining pulses, one at a time, e.g. 1999 -> 1998 -> 1997 ->... and so forth. When the number of remaining pulses reaches zero, the motor will stop turning.

(2) R1: FL speed register (13 bits)

This register is used to specify initial speed. The range is 1 to 8191, in decimal notation (1hex to 1FFFh).

When you want to accelerate the motor, it will accelerate from the FL speed (initial speed) until it reaches FH speed (operation speed).

(3) R2: FH speed register (13 bits)

This register is used to specify operation speed.

Just like R1, the range is 1 to 8191, in decimal notation (1h to 1FFFh).

* Note

The values in R1 and R2 are not equal to the pulse output frequency of the PCD. The actual pulse frequency will be obtained by multiplying the "magnification rate" described in item 6. (5) below by these register values.

(4) R3: Acceleration/deceleration rate register (10 bits)

This register is used to specify the acceleration and deceleration time. The range is 2 to 1023, in decimal notation (2h to 3FFh). In the PCD series, the acceleration/deceleration times are both specified in this register, which means that the acceleration time and deceleration time will be the same. The value entered does not refer to an actual amount of time. The relationship between the actual time and the value entered will be as follows.

$$R3 = \frac{\text{Reference clock frequency} \times \text{Acceleration/deceleration time}}{R2 - R1}$$

(This equation only applies when linear acceleration/deceleration is selected. The equation for S-curve acceleration/deceleration will be a little different. For details, see the User's Manual.)

For example, when the reference clock frequency is 4.9152 MHz, R1 = 50, R2 = 950, and the desired acceleration/deceleration time is 0.1 sec., the value to put in R3 shall be as follows.

$$R3 = \frac{4915200 \times 0.1}{950 - 50} = 546.13 = 546$$

(5) R4: Magnification rate register (10 bits)

This register is used to specify the magnification rate. The range is 2 to 1023, in decimal notation (2hex to 3FFh).

Multiply this magnification rate by the R1 and R2 settings to obtain the actual output pulse frequency. For example, when R1 = 50, R2 = 950, and the magnification rate = 2x, FL will be 100 pps, and FH will be 1900 pps. (pps means: pulses per second)

The R4 value itself is not the magnification rate. You can obtain the value to use for R4 from the equation below.

$$R4 = \frac{\text{Reference clock frequency}}{\text{Magnification} \times 8192}$$

Therefore, when the reference clock frequency is 4.9152 MHz (4915200 Hz) and you want a magnification rate of 2x, the R4 value should be 300 decimal (01 0010 1100b (= 12Chex) written as 10 bits of data).

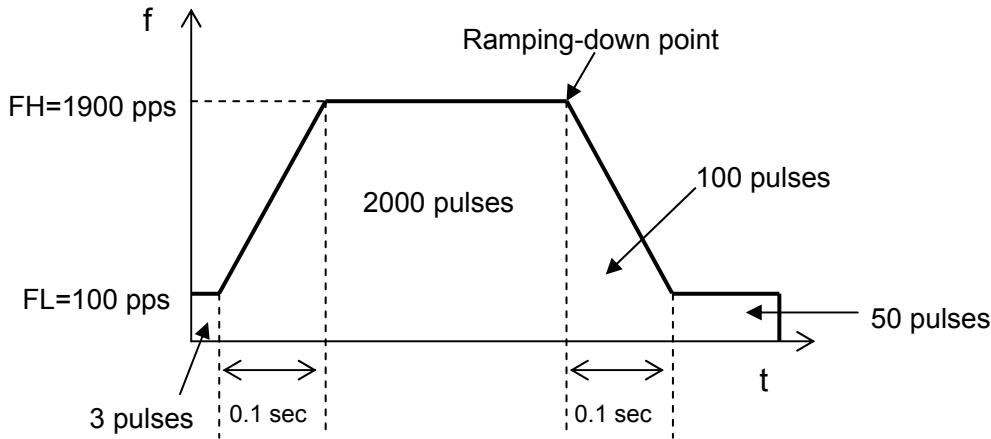
The actual output pulse rate varies with the magnification rate you use, as follows.

Magnification rate	Unit	Output pulse rate range	R4
1x	1 pps	1 to 8191 pps	600 (258h)
2x	2 pps	2 to 16381 pps	300 (12Ch)
10x	10 pps	10 to 81910 pps	60 (3Ch)
50x	50 pps	50 to 409550 pps	12 (0Ch)

In other words, R1 and R2 can be used to select any one of 8191 speeds.

(6) R5: Ramping-down point register (16 bits)

When using a preset operation with acceleration/deceleration, this register is used to specify number of pulses left when deceleration starts. The range is 0 to 65535, in decimal notation (0 to FFFFh). For example, the settings on figure 7 show R0 = 2000, and R5 = 100. Therefore, the motor starts to decelerate 1900 pulses after the start. It will decelerate to the FL speed while 100 pulses are received and then stop.



<Figure 8>

When R5 = 150, the motor will start to decelerate after receiving 1850 pulses. Since it uses 100 pulses to decelerate to FL speed, the motor will operate at FL speed for the balance of 50 pulses as shown in figure 8.

(7) R6: Idling pulse register (3 bits)

The range is 0 to 7, in decimal notation (0 to 7h).

Stepper motors cannot self-start at speeds higher than the maximum self-start speed. If you try, they will not rotate normally. This is referred to as being "out of step." (The meaning of "self-start speed" is described in the "Motor terminology" section)

When starting a stepper motor using various acceleration/deceleration patterns, the stepper motor starts to accelerate at the same time it starts rotating. The result is that the first two pulses will be shorter than the length of one FL speed pulse (they will be at a higher frequency). This means that even though the stepper motor can self start in theory, it may be out of step.

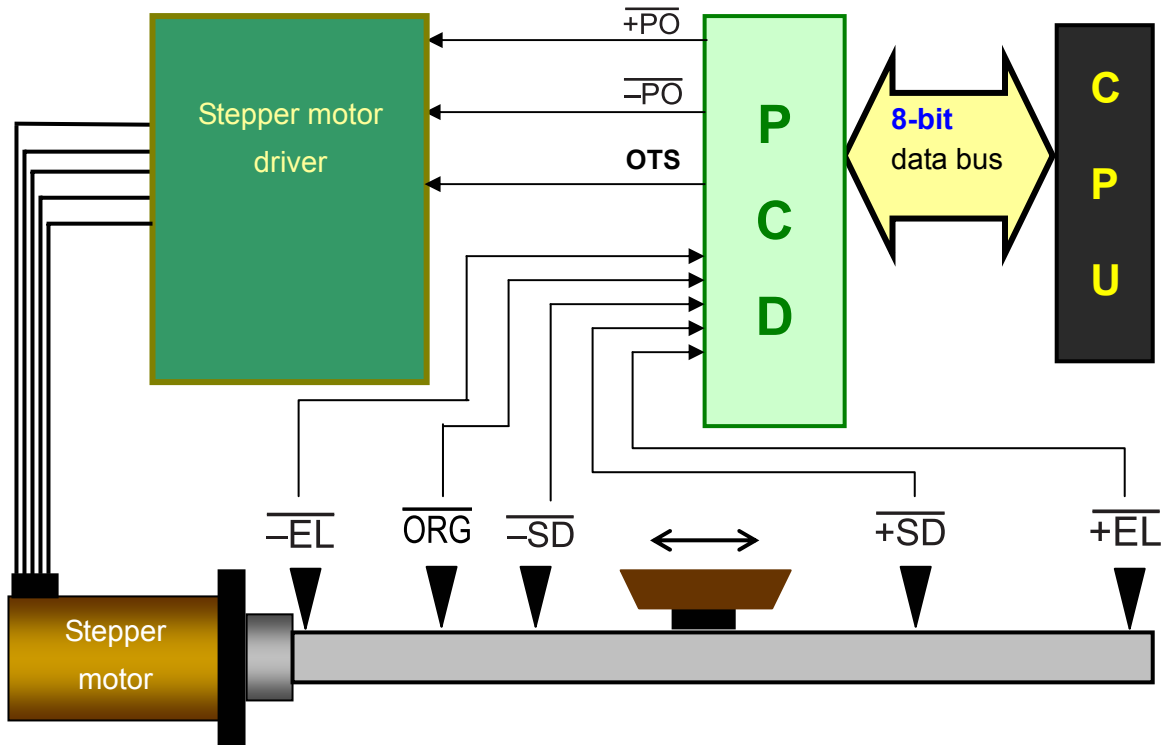
Therefore, in order to self start the stepper motor, the PCD can instruct the motor to rotate by sending some pulses at FL speed while starting, and not accelerate immediately. These pulses, that allow a slower start, are referred to as idling pulses.

Figures 7 and 8 above show an example using three pulses of rotation at FL speed while starting.

7. How is the motor driver connected to mechanical systems?

Although there are many different types of mechanisms, figure 9 shows a basic configuration for a linear operation using a ball screw.

A sample basic configuration is shown in figure 9 below.



<Figure 9>

Each terminal connected to the CPU was described before. This section describes the terminals connected to a motor driver and to mechanical systems.

(1) $\overline{+PO}$, $\overline{-PO}$ (Pulse Out)

These terminals output pulses to a motor driver. "+" indicates clockwise rotation and "-" indicates counterclockwise rotation pulses (a dual pulse system). (The PCD4541 can output pulse strings or direction signals (a single pulse system) by setting the R7 register.

(2) OTS (Output Signal)

This is a general-purpose output terminal. If bit 4 in the Control Mode command is set to "1," an H level signal is output. If bit 4 is set to "0," an L level signal is output. In other words this is a signal that can be turned ON/OFF with the control command. This is usually used to turn a motor driver excitation circuit ON/OFF. But, it can be used for other applications.

(3) $\overline{\text{ORG}}$ (Origin)

This is the zero position terminal. When this signal is LOW, the motor immediately stops. When you want to position a motor using a linear operation pattern, as shown in figure 9 above, you have to specify a zero position.

By installing a switch at a zero position and connecting it to this terminal, the motor will stop immediately when this signal is input.

(4) $\overline{+SD}$, $\overline{-SD}$ (Ramp-down)

These are deceleration start signals. When the motor is rotating at FH speed and this signal goes LOW, the motor will start to decelerate. These terminals are mainly used in zero return operations. When an $\overline{\text{ORG}}$ signal is input while feeding at FH speed, the motor must be stopped quickly using a brake and it may be possible to overrun the desired position. Therefore, the motor needs to be decelerated in advance, before the $\overline{\text{ORG}}$ signal is received, by using a switch installed a little in front of the $\overline{\text{ORG}}$ switch and connected to this terminal.

(5) $\overline{+EL}$, $\overline{-EL}$ (End Limit)

These are end limit signals. When either of these signals goes LOW, the motor stops immediately. These are usually connected to switches at both ends of a linear operation. In other words, if either of these signals becomes active, the PCD is instructed to not to advance any more.

* The signals in items (3), (4), and (5) are active low signals. In other words, when these switches are not turned ON, the terminals have a 5 volt signal level (HIGH). When the switch is turned ON, the voltage drops to 0 and is considered active.

Why are these terminal active low?

=> It is especially important that the motor must stop, no matter what, when the $\overline{\text{EL}}$ signal is active. If a cable connected the $\overline{\text{EL}}$ switch becomes disconnected, the $\overline{\text{EL}}$ signal line will go LOW, which allows it to function in a fail-safe manner.

The other reason for this arrangement is that an "active low" circuit is a better design for resisting electric noise.

8. What is the Status byte?

The Status byte is 8 bits of read-only data. By reading this data, the CPU can monitor the status of the PCD. Similar to the commands, each bit in the Status byte has its own role.

In section 2, subsection (4), "Reading and writing data," some examples of reading data were described. Among them, the status of the following items can be read.

- 2) "Is the PCD operating or stopped?"
- 3) "Is the PCD performing an acceleration or operating at rated speed?"
- 4) "Has an external signal been received?"

There are also some other items whose status can be checked.

For example, when the CPU requests the data for Status 0, and if the values in table 3 below are read from the data bus (ADh),

D7	D6	D5	D4	D3	D2	D1	D0
1	0	1	0	1	1	0	1

<Table 3>

The meaning of the bits is as follows.

D7 = 1: Currently decelerating.

D6 = 0: Currently not accelerating.

D5 = 1: The value in the preset counter (PC) is smaller than the value of R5.

D4 = 0: The preset counter has not reached zero.

D3 = 1: Currently operating.

D2 = 1: The $\overline{\text{INT}}$ signal will not be output on an external start.

D1 = 0: The $\overline{\text{INT}}$ signal was output when deceleration started, and it is still being output.

D0 = 1: The $\overline{\text{INT}}$ signal for a stop operation has not been output.

For a detailed description of each bit, see the user's manual (pages 43 and 44 for the PCD4511/4521/4541).

The address used for reading Status 0 is A (1:0) = 00.

There is another byte of status data that can be read, Status 1. It can be used to confirm whether the $\overline{\text{EL}}$, $\overline{\text{SD}}$, or $\overline{\text{ORG}}$ signals are ON or OFF. The PCD 4511/4521/4541 also have data bytes called Status 2 and Status 3, that are used to confirm whether the $\overline{\text{PO}}$ signal is currently being output, and which phase is currently being output.

9. Terminology

What follows are descriptions of terms used for the PCD series, such as items that must be specified and function names.

Number of axes	Number of axes that can be controlled by one chip.
Reference clock	Basic clock frequency that can be used as standard. For the PCD series, we recommend a 4.9152 MHz clock. (A clock speed up to 10 MHz can be used.)
Max. Output frequency	Maximum frequency of pulses that can be output. For the PCD series, this is shown as a nominal "400 kpps" on the leaflet. This is the actually useful value when controlling a stepper motor. By setting the "speed magnification rate" below, a maximum of 2.4 Mpps can be output. (When you want to accelerate and decelerate a motor, it should be decelerated with a value within the "ramp-down point range" shown below, so that the actual applicable frequency will be approximately 400 kpps.
Number of speed registers	Two speeds are possible: FL (self start) and FH (high speed). By rewriting the FH register, the feed speed can be changed on the fly.
Number of speeds to specify	Express number of speeds that can be selected.
Speed magnification rate range	The actual number of pulses per second is obtained by multiplying the magnification rate by the value in the speed register. For the PCD series, the magnification rates are "1x" and "2x." However, the magnification rate can be set larger.
Acceleration/deceleration rate range	The range for setting the slope angle used in acceleration/deceleration. The acceleration/deceleration time can be calculated from the setting for this rate.
Number of positioning pulses range (count-down counter)	The minimum and maximum number of preset output pulses that can be used for positioning operations.
Ramp-down point range	The minimum and maximum number of pulses left when deceleration starts in preset operation. Keep this value within the allowable range, even if you are decelerating a motor from a higher frequency to FL speed.
Timer operation	While in preset operation, you can stop outputting pulses for a certain period so that the PCD can be used as a timer.
Idling pulse output	The PCD can start acceleration after outputting a number of pulses at FL speed in the beginning. Using this function, you can specify an FL speed near the upper limit of the self-start speed, to control a stepper motor more accurately.

S-curve acceleration/deceleration	<p>A method of controlling acceleration and deceleration with an S-curve shape. Use of S-curves reduces the mechanical shock of linear acceleration/deceleration. (The amount of reduction depends on the motor, the mechanism, operating pattern, and other conditions.)</p> <p>The PCD series employs a quadratic equation to create this S-curve.</p>
External start/stop function	<p>The PCD can be started/stopped by a signal from external equipment, in addition to using the start command. To allow an external start, set bit D1 in the start mode command to 1 while stopped. The PCD will wait for an external signal to start. When a signal is seen on the STA terminal, the PCD will start operation.</p>
Mechanical external signal	<p>The following signals can be read by the PCD as position detection signals from a mechanical system.</p> <ol style="list-style-type: none"> 1) $\overline{\text{ORG}}$ signal: Mechanical zero position switch input. Use for zero return operation. 2) $\overline{+SD}/\overline{-SD}$ signals: Ramp-down input signal from a mechanical system. When these signals are enabled and a signal for a feed direction turns ON, the PCD will decelerate to FL speed. Then, when the signal turns OFF, the PCD will accelerate again. 3) $\overline{+EL}/\overline{-EL}$ signals: Limit input signals from a mechanical system. When a signal for a feed direction turns ON, the PCD will stop immediately. After that, even if the signal turns OFF, the PCD will remain stopped.
Status for monitoring operations	<p>By reading status bytes from the PCD, you can monitor the current PCD operation and the input condition of external signals.</p>
Excitation output for a 2-phase stepper motor	<p>Software can be used to change the PCD output pulses to a 2-phase excitation sequence signal (unipolar/bipolar with combinations of full or half steps). In particular, if you are using a 2-phase stepper motor, you can easily construct a controller and driver circuit system by connecting a transistor array to a device such as one of our drive ICs (NP and SLA series). You can control 3-phase and 5-phase stepper motors with input pulse signals sent to the drivers for these motors.</p>
Excitation system selection input signal ($\overline{\text{U}}/\text{B}$)	<p>When using the excitation output for 2-phase stepper motors discussed above, this signal is used to select between unipolar and bipolar.</p> <p>LOW selects unipolar and HIGH selects bipolar.</p>
Excitation sequence selection input signal ($\overline{\text{F}}/\text{H}$)	<p>When using the excitation output for 2-phase stepper motors discussed above, this signal is used to select between full and half step.</p> <p>LOW selects full step (2-2 phase excitation) and HIGH selects half step (1-2 phase excitation).</p>

Interrupt signal output ($\overline{\text{INT}}$)	Interrupt signal sent to the CPU.
General-purpose output terminal (OTS)	This terminal can be controlled by the CPU. Use it for general-purpose tasks such as turning the excitation for a stepper motor driver ON/OFF, or as a start/stop signal for another axis.
Busy signal output ($\overline{\text{BSY}}$)	This signal is output while the PCD is operating. Use it to monitor whether the PCD is currently operating or not.
IC Packages	All the PCD series ICs are supplied in "QFP" packages. Our PCL series are supplied in the following packages. QFP: Quad Flat Package SOP: Small Outline Package DIP: Dual In-line Package.

<Table 4>

10. The following tools are available.

"Register setting calculations" and "Operation setting flow" files are available as EXCEL files.

(1) Register setting calculations

Enter values for the reference clock used, the magnification rate, preset amount (feed amount), FL, FH, and acceleration/deceleration time, on a trapezoid form. The values needed for each register are calculated automatically and shown in decimal and hexadecimal notation.

(2) Operation setting flow

This file depends partly on the file above. But, it allows you to give an example of the procedure you want for trapezoid operation as a flow chart. To calculate each of the register values, enter the required values and specify the command by entering a 1 or 0 for each bit. The values on the right of the flow chart are calculated and displayed in hexadecimal notation.

These files are included in the CD-ROM together with the instruction manuals.

<Reference documents>

1. What kinds of applications can the PCD series be used for?

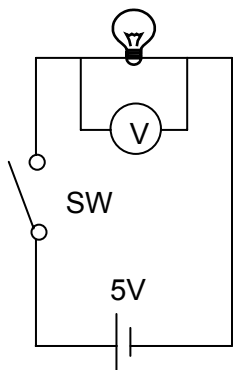
The following applications can be anticipated.

- (1) Medical instruments such as blood inspection devices, stool examination devices, or applying labels to blood sample vials.
- (2) Various analysis devices.
- (3) Autofocus systems in laboratories.
- (4) Card readers
- (5) High class or large scale OA equipment (facsimile machines, printers, and copy machine finishers for business use)
- (6) Semiconductor manufacturing devices.

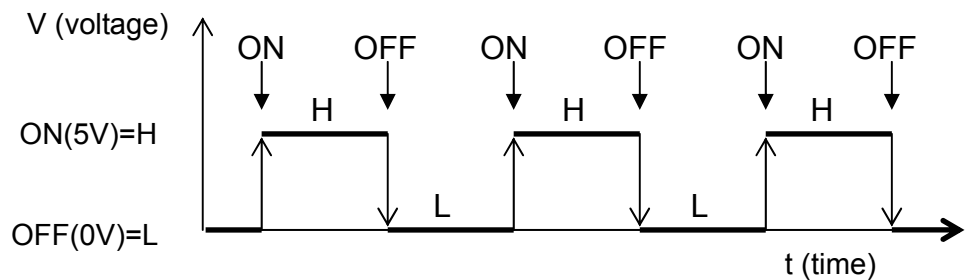
2. Basic digital circuits and stepper motor operation

(1) What are high and low signal levels?

When the SW in the circuit on figure 10 is turned ON and OFF repeatedly, the voltage will change as shown in figure 11 below.



<Figure 10>



<Figure 11>

When the SW is ON => This is referred to as a "high level (H level)."

When the SW is OFF => This is referred to as a "low level (L level)."

(2) Signals (terminal names) with a bar over the name

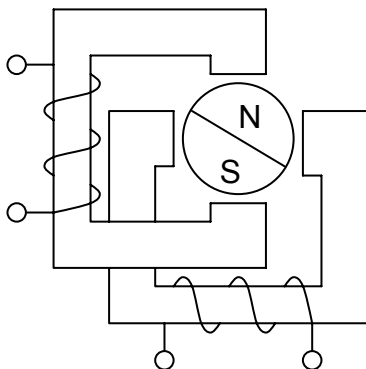
Normally, as shown in figure 11 above, when a SW is turned ON and creates an H level voltage, the signal is considered active. However, some signals are active when at the L level. (This is referred to as an active low or negative logic signal.)

=> Signal names with a bar over them like this " $\overline{\text{INT}}$ ", are active low signals.

Why are these signals used? => Active low signals are used to eliminate the effects of noise, or to stop a motor if a cable becomes disconnected.

(3) Terms used with stepper motors

1) Self-start frequency



<Figure 12>

Most stepper motors are constructed using a permanent magnet for the rotor and several electromagnets arranged around the rotor, as shown in figure 12. By turning these electromagnets ON and OFF, the permanent magnet in the middle rotates.

A motor driver turns the electromagnets ON and OFF according to pulses output from the PCD or according to excitation signal timing that is output by the PCD. If the electromagnets are turned ON and OFF too quickly, the rotor cannot keep up and the motor shaft will not rotate.

Stepper motors have a natural rate at which the electromagnets can be switched ON and OFF. This rate is expressed as the frequency for input pulses. The term "self-start frequency" refers to the frequency at which a motor can be started (or will stop) in response to a string of input pulses. In particular, the maximum frequency that can be used to start a motor without a load is referred to as the "maximum self-start frequency."

Once a motor has started to rotate, a stepper motor can be rotated at a speed higher than the maximum self-start frequency by increasing the pulse frequency. Therefore, in order to control the motor over a wide range of speeds higher than the maximum self-start frequency, the PCD series has an acceleration/deceleration control function. The acceleration and deceleration conditions vary with motor type and the load being driven by the motor.

2) Unipolar/bipolar

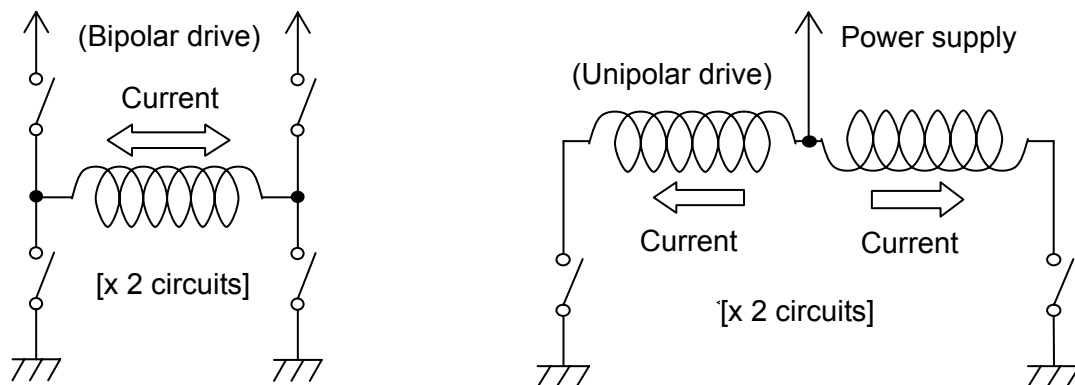
As described in the paragraph about the "self-start frequency" above, stepper motors contain electromagnets. There are two types of current flow in these electromagnets. When the current in a motor only flows in a single direction, it is referred to as a "unipolar" drive. When the current in a motor flows both forward and backward, it is referred to as a "bipolar" drive.

There are two methods for changing the polarity of the electromagnets. One is to use a single coil and change the direction of the current flow through the coil. This is how a bipolar drive system works. The other way is to wind two coils (one for rotating forward and one for rotating backward) and provide current to one of the two coils. This is how a unipolar drive system works.

Stepper motors with one coil (as found in a bipolar system) usually need a complicated circuit to drive the motor. The motor driver must have a number of ON/OFF elements (8 elements) to change the direction of the current flow.

Stepper motors with two coils (as found in a unipolar drive system) can work with simpler circuit since they only need to control the current flow in one direction and just have to select one of the

two coils. The motor driver for a unipolar system only needs 4 elements to control the current. However, twice as many coils are needed, when compared with a bipolar system.



<Figure 13>

3) 2-phase/5-phase

The phrase "stepper motor" refers to an "n-phase stepper motor," where n is the number of electromagnets used in its construction. The most familiar stepper motor is a "2-phase stepper motor," with two electromagnets. The 2-phase stepper motor is easy to control. However, it does not offer precise resolution.

5-phase stepper motors have five electromagnets and offer finer resolution than 2-phase stepper motors. However, they require more complicated controls and the drivers must be more sophisticated than for 2-phase stepper motors.

4) Full step (2-2 phase excitation)/half step (1-2 phase excitation)

These terms are used for the 2-phase stepper motors. One of the two electromagnets is excited (current flows and the electromagnet is turned ON), which is referred to as "1-phase excitation." When both two electromagnets are excited, this is referred to as "2-phase excitation."

There are two ways to rotate these stepper motors: 2-2 phase excitation rotates the motor by switching the 2-phase excitation conditions, while 1-2 phase excitation rotates the motor by combining the use of 1-phase and 2-phase excitation.

Compared with 2-2 phase excitation, 1-2 phase excitation offers double the resolution for fine control. Therefore, 2-2 phase excitation is also referred to as "full step" control and 1-2 phase excitation can be called "half step" control.



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

Asia/Europe: Nippon Pulse Motor Co., Ltd.

2-16-13 Hongo, Bunkyo-ku, Tokyo 113-0033, Japan

TEL: 81-3-3813-8841

FAX: 81-3-3813-7049

Web: <http://www.pulsemotor.com>

E-mail: int-l@npm.co.jp

North/South America: Nippon Pulse America, Inc.

1073 East Main Street, Radford, VA 24141, U.S.A.

TEL: 1-540-633-1677

FAX: 1-540-633-7049

Web: <http://www.nipponpulse.com> E-mail: info@nipponpulse.com