

No. QT32-12001

**MYCOM**

Electric Cam Controller

**ECM-011**

---

---

---

**【User's manual】**

**MYCOM,INC.**

# Attention on Safety

Please read this user's manual carefully before using this device.



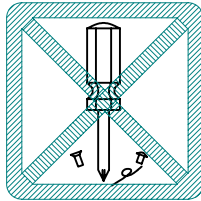
## Warning

Please keep the followings and use safely to avoid a fatal accident.

- **Do not use or leave it unattended after disassembling or damaging this unit.**

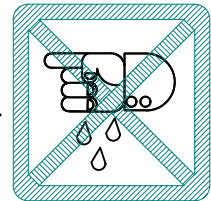
The above might cause an electrical shock or an accident.

We will not be responsible for repair if the unit is disassembled by the customer.



- **Do not touch the unit with wet hand.**

The above might cause electric shock or accidents.



- **Avoid having metals and other foreign objects enter into the device.**

The above might cause fire, electric shock or accidents.



# Attention on safety

Please read this user's manual carefully before using this unit.

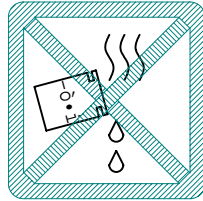


## Attention

Please keep the followings and use safely to avoid an accident.

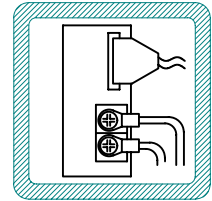
- Do not use or store the unit near by corrosive gas, watery or chemical place.

It may cause fire, electric shock or accident.



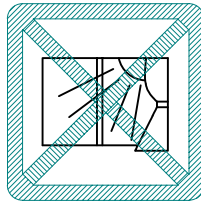
- Confirm the input voltage and wiring before power on.

Wrong wiring or input voltage may cause fire or accident.



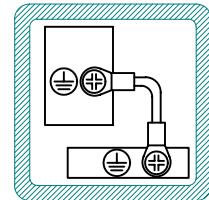
- Do not use or store this unit in a place of direct rays of the sun.

It may cause an accident.



- Earth

Earth (D class earth) or connect with protective earth in case of the product having protective earth terminal to avoid electric shock.



# Attention on usage

Please read this user's manual before using this device.

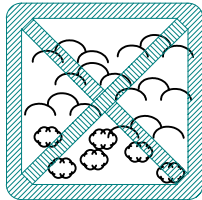


## Attention

Please keep the following and use safely to avoid an accident.

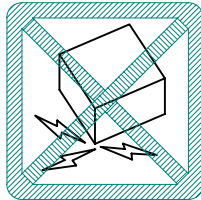
**Do not use or store this device in a dusty place.**

This device is not constructed dustproof. Operating in dusty place causes the accident.



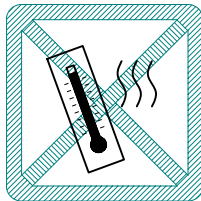
● **Do not give the device a big shock.**

This is a precision device and it should not be dropped or given a big shock. It causes the accident.



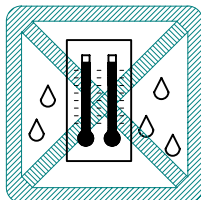
● **Do not use or store it in a place where the temperature might rise too high or too low.**

The operating temperature is 0°~40°C and the storage temperature is 0°~60°C. Avoid extreme temperature changes.



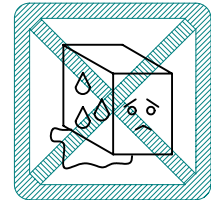
● **Do not use or store it in a place where the humidity might rise too high or too low.**

The operating humidity should be within 30% - 80%. The storage humidity should be within 20% -



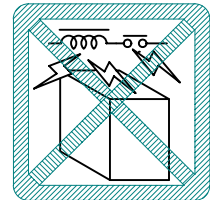
● **Avoid condensation of dews.**

Bringing the device into a place of high humidity or a place where temperature changes suddenly will cause condensation of dews.

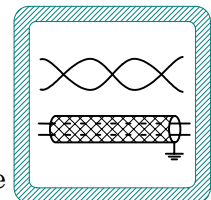


● **Take measures against noises.** Noises will cause malfunctions and accidents.

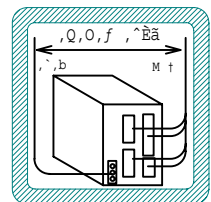
Use the power supply which is not connected to a noise generating device.



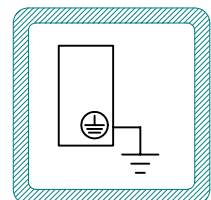
Use shielded wires for wiring. Make them as short as possible. Use the twisted pair wires for the clock outputs.



In case of AC type products, separate power cable (AC) from signal line (DC) more than 20cm.



Earth if the products have earth terminal.



90%.

-Contents-

Electric Cam Controller .....	1
1. Model name.....	1
2. Outline .....	2
<b>3. Configurations</b> .....	<b>3</b>
3-1 System Configuration .....	3
3-2. Dimensions .....	4
3-3. Description of parts.....	5
4. Specification.....	6
5. Functions .....	7
5-1. Electric cam function .....	7
5-2. Function of Program .....	9
5-3. Parameter setting function.....	11
6. Command system.....	13
6-1. MOV_PTP.....	14
6-2. MOV_HOM.....	14
6-3. MOV_SCN .....	15
6-4. MOV_CPX .....	16
6-5. MOV_CAM (Without cam shape).....	17
6-6. MOV_CAM (Shape change specified).....	18
6-7. SET_TBL.....	20
6-8. SET_MAX.....	20
6-9. SET_MIN.....	20
6-10. SET_ACC.....	20
6-11. SET_DEC.....	20
6-12. SET_ENC .....	19
6-13. CPX_SP1 .....	22
6-14. CPX_SP2 .....	22
6-15. CPX_SP3 .....	22
6-16. CPX_PS1 .....	22
6-17. CPX_PS2 .....	22
6-18. CPX_PS3 .....	22
6-19. STP_SDW .....	23
6-20. STP_EMR.....	23
6-21. STP_CAM.....	24
6-22. STP_PAS.....	24
6-23. SRV_ON.....	25
6-24. SRV_OFF.....	25
6-25. CLR_ON .....	25
6-26. ALL_RST .....	25
6-27. MSK_SIN.....	26
6-28. PRG_STR.....	26
6-29. PRG_END.....	26

6-30. PRG_TSK .....	26
6-31. PRG_JMP .....	27
6-32. PRG_LIN .....	27
6-33. PRG_CAL .....	28
6-34. L00 to L99 .....	28
6-35. WAIT.....	29
6-36. IF .....	28
6-37. REG_POS. ....	32
6-38. REG_R00 to R99.....	32
6-39. REG_TM0 to TM7 .....	34
6-40. REG_O00 to 15.....	34
6-41. REG_OUT.....	35
6-42. REG_I00 to 15 .....	35
6-43. REG_IN .....	37
6-44. REG_STS.....	37
6-45. REG_SPI.....	38
6-46. REG_SPO .....	38
6-47. REG_ERR.....	39
6-48. REG_ENC.....	39
6-49. REG_SWT.....	39
Home search sequence .....	41
7-1. Home search sequence 0 .....	41
7-2. Home search sequence 1. ....	41
7-3. Homse search sequence 2.....	42
7-4. Homse search sequence 3.....	42
8. On-line Operation .....	43
9. Connector .....	44
9-1. I/O Connector .....	44
9-2. RS-232C Connector .....	47
9-3. DC5V Connector.....	47
9-4. I/O2 Connector .....	48
10. Example of wiring .....	49
10-1. Connectiong example of I/O1 connector .....	49
10-2. Example of RS-232C connecting .....	50
11. Timing .....	51
11-1. External signal timing .....	51
11-2. Command execution time .....	51
12. Error code .....	52



## 2. Outline

This is an electronic CAM controller unit used to control the stepping motors or the pulse train input servo motors. In addition to the electronic CAM operation, this unit carries programs which enable positioning and simple PLC functions.

The main features are described below.

### 1. CAM Shape of 32 Pattern Memorized

The electronic CAM operation synchronizes to the encoder signal of the main axis and operates the secondary axis according to the preset operating patterns (the shape of CAM). The shape of CAM is generated by the computer and down-loaded to this machine.

### 2. S Curve Acceleration and Deceleration

It is possible to accelerate and decelerate using the S-curve.

There are 8 types of S-curve available. (The user can modify one of them.)

### 3. 128 Steps and 32 Programs Are Memorized

The command of the Simple PLC functions of 4. is memorized in addition to the electronic CAM, Index, Scan, and Home Search.

### 4. Multi-tasking Used for Simple PLC Functions

The Simple PLC functions such as the branching and timer functions of programs using the input terminals and calculation using the registers are realized by multi-tasking.

### 5. Up to 16 Axes Can be Connected Using the Daisy Chain Connection

It is possible to share up to 16 units by daisy chaining of RS-232C communication.

Upload and download of on-line operation and program are available by communication.

### 6. Memorizing Programs for Emergency Sequence is available.

It is possible to execute automatically a special emergency program when an emergency develops.

### 7. Parameters Can Be Set Up according to the Systems

Parameters such as the motor types, the forward directions, the clock systems, the sensor logic's, acceleration/deceleration shapes, speed modes, etc. can be set up according to the systems.

### 8. Comparison with Former model ECM-010

Item	New model (ECM-011)	Former model (ECM-010)
General purpose I/O	16 inputs and 16 outputs	8 inputs and 8 outputs
Program execution	Same time operation of program execution and online execution are available.	Online execution is not available during program execution.
Lines of program	Up to 256 lines	Up to 128 lines.
Encoder counter	There are registers for encoder counter	There is no register for encoder counter.
Switching of encoder	Switching of external encoder and internal clock for main axis is available.	Switching is available by parameter setting.
Modification of electric cam shape	It is available to change into different cam shape, changing certain cam shape gradually.	Only one shape moves by specification.
Speed of electric cam moving	Secondary axis speed is available up to 4 times of main axis speed during electric cam moving..	Available up to twice.
Temporary stop of cam moving	Temporary stop and resumption are available during cam moving.	Temporary stop and resumption are unavailable.
Numbers of Cam memory	32 cams	16 cams
Daisy chain	There is no daisy chain signal of program start, cam start nor emergency stop.	There is the daisy chain signal of program start, cam start nor emergency stop.
Dimension	160 × 85 × 25	160 × 160 × 20

### 3. Configurations

#### 3-1 System Configuration

This machine is a one axis unit. However, it can be used with multiple axes as shown below.

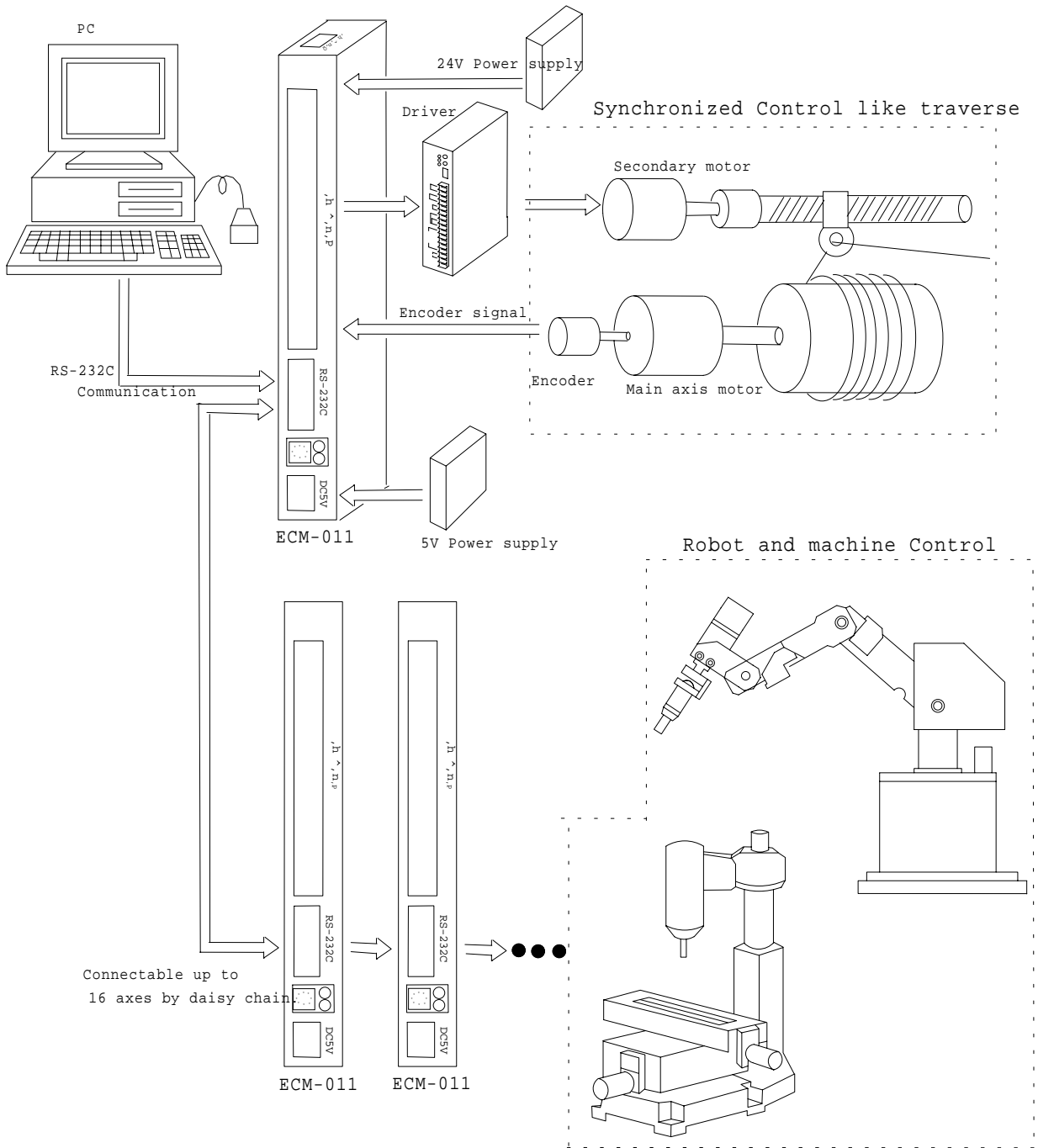
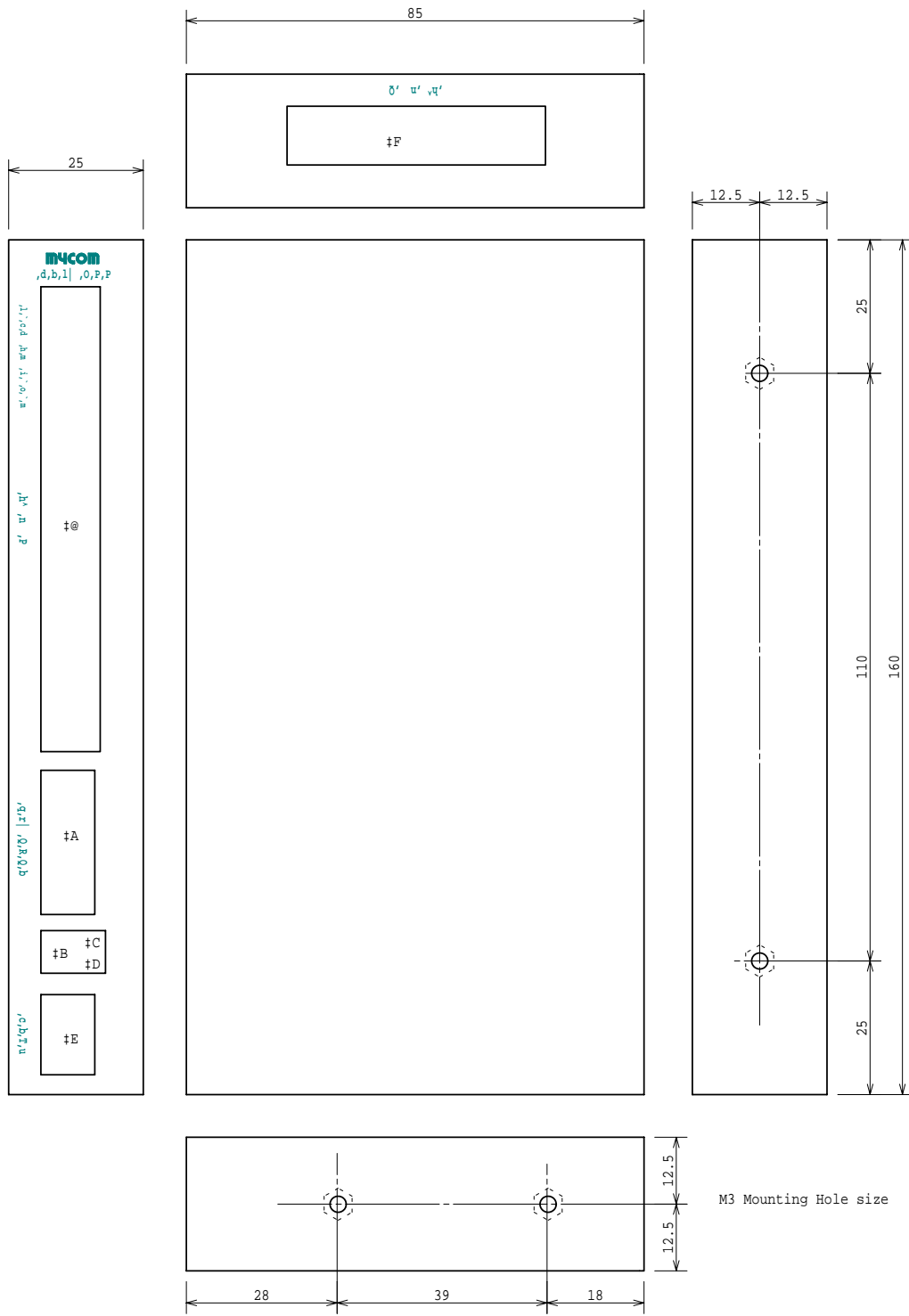


Fig. 3-1 System Configuration

### 3-2. Dimensions



**Fig. 3-2. Dimensions**

### 3-3. Description of parts

(1) I/O Connectors

I/O connectors are used to connect the general purpose/dedicated I/O, sensor input, driver I/O, and power supply for sensors.

(2) RS-232C communication connector (RS-232C)

This is used when RS-232C is used for communication.

(3) Axis Name Setting Switch

This switch is used to set the name of axes when multiple axes are used.

The switch setting corresponds to the axis name as below.

Switch setting	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Axis name	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P

When one axis is used, use 0 (Axis A).

(4) Error Lamp (Red LED)

When there is an error, the red lamp will be lit.

(5) Power Supply Lamp (Green LED)

The green lamp is lit when the power is turned on.

(6) Power supply connector

Connect the DC5V power supply.

(7) I/O Connectors 2 (I/O2)

Connect the general purpose I/Os (8 to 15)

#### 4. Specification

Items	Description	
Control system	Microprocessor control system. This unit carries the high-performance pulse generator, MPG2031 made by MYCOM. The exclusive electronic CAM control system made by MYCOM is used.	
Number of axes controlled	One axis (Up to 16 axes can be controlled using the daisy chain connection) The pulse output systems used are the photocoupler output and the RS-422 output. Multiple number of units can be connected.	
Motors used	Stepping motor or pulse train input servo motor	
Data store	Cam shape, Program and System parameter are stored. (Flash memory is used. It can be re-written 100,000 times.)	
Cam shape	32 patterns can be memorized. The CAM shapes are generated by the computer and downloaded. One CAM shape is composed of maximum 510 plotting points.	
Program	32 programs, 256 steps are stored. It is possible to store the emergency sequence program which is executed when an emergency occurs. Moving command (Electric Cam, Index, Scan, Home search, Complex trapezoidal drive) Complementary command (Timer, AND, OR, Arithmetic four rules calculation, Register, Multi-task, Various setting)	
Parameter	Speed mode, S curve acceleration/deceleration, Motor type, Forward direction, Clock type, Sensor logic, Backlash, Encoder setting, Moving speed, Memory initialization.	
Input interface	Program start, Cam start, Emergency stop, 16 General purpose inputs (Photo coupler isolated)	
Output interface	Ready, Moving, Error, 16 General purpose outputs (Photo coupler isolated)	
Sensor interface	Forward, Reverse, Near Home, Home, A phase, B phase (Photo coupler isolated)	
Driver interface	CW/CCW: Forward direction switchable, Selectable 2 clock or 1 clock, Open collector, RS422 Servo On: On/Off controllable, Photo coupler isolated Counter clear: One shot output (about 50ms), Photo coupler isolated Alarm: Logic changeable In-position	
Communication	RS-232C One channel 9600 baud, data bit 8, stop bit 1, without parity check Connectable up to 16 pieces of this unit	
Multiple axes control	Common use of RS-232C is available up to 16 units. There is a switch to set axis name for multi-axis control.	
Display	Power(Green LED), Error(Red LED)	
Dimension	160 × 85 × 25 [mm]	
Weight	About 340g	
Power input	Input	Less than 5V, 0.5A for unit Range of voltage; DC4.75V-DC5.25V Less than 24V, 0.2A for encoder Range of voltage; DC22.8V-DC25.2V
	Output	Less than 5V, 0.2A for encoder Range of voltage; DC4.75V-DC5.25V

Table 4-1. Specification

## 5. Functions

### 5-1. Electric cam function

The electronic CAM function takes in the encoder signal and synchronizes with this signal. Then, it outputs the preset CAM shape pulses.

The encoder signal is called the main axis and the axis corresponding to the CAM shape is called the secondary axis. It is possible to use the internal clock for operation even when the encoder signal is not connected.

The CAM shape is generated by the personal computer and downloaded and stored in the main controller by RS-232C interface. 32 types of CAM shapes can be stored.

When operating the CAM shapes, the personal computer is used to generate programs, which contain CAM operation start command. The programs are downloaded to the main controller and executed. If the CAM activating signal is inputted when the program is being executed, the CAM shape of the secondary axis is executed.

When changing the CAM shape, the CAM shape is generated again by the personal computer and is downloaded.

The above procedure is described by flowchart.

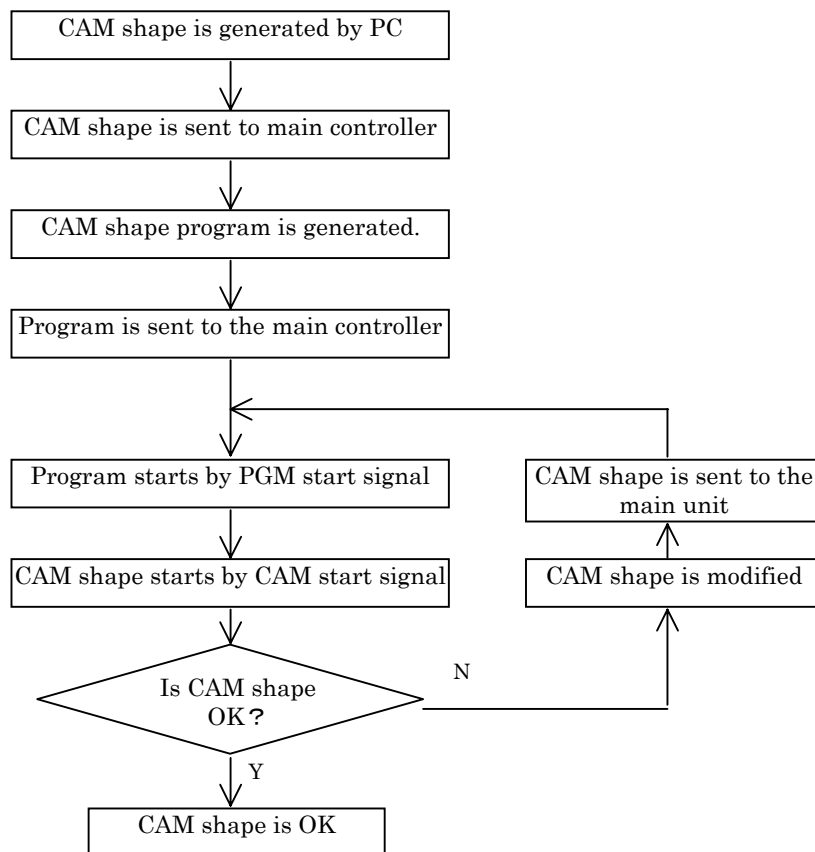


Fig. 5-1, Flowchart of Electric CAM operation

If the CAM start signal is inputted as shown below, the secondary axis will operate according to the CAM shape.

When the CAM shape movement is finished once, if the CAM start signal is OFF (See the left side figure), standby until the next start signal; if the CAM signal is ON (See the right figure), repeat the CAM shape movement.

It is possible to repeat the movement of the CAM shape continuously.

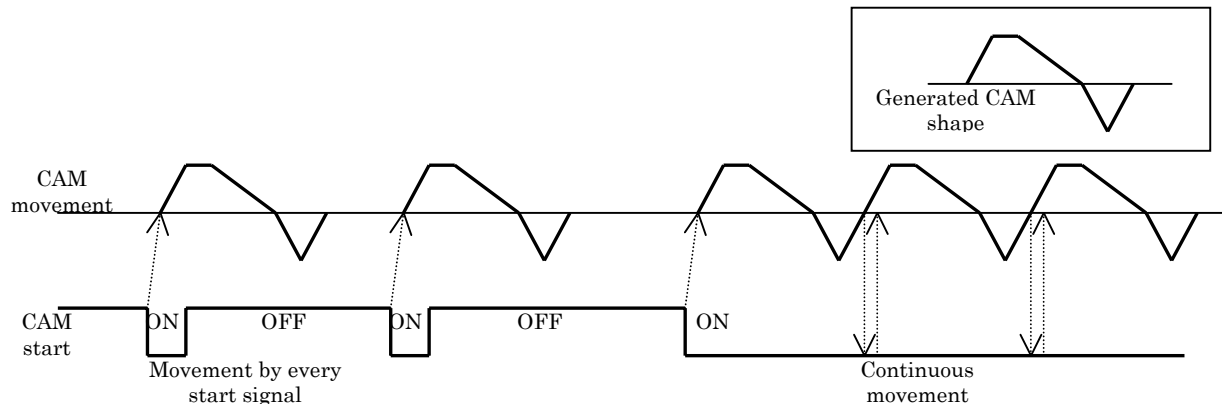


Fig. 5-2. Operations for each starting signal and continuous movement

Also it is possible to operate with changing gradually from one CAM shape to another CAM shape as below figure.

The below figure is a sample which specifies to finish the changing by four times from shape a to shape b.

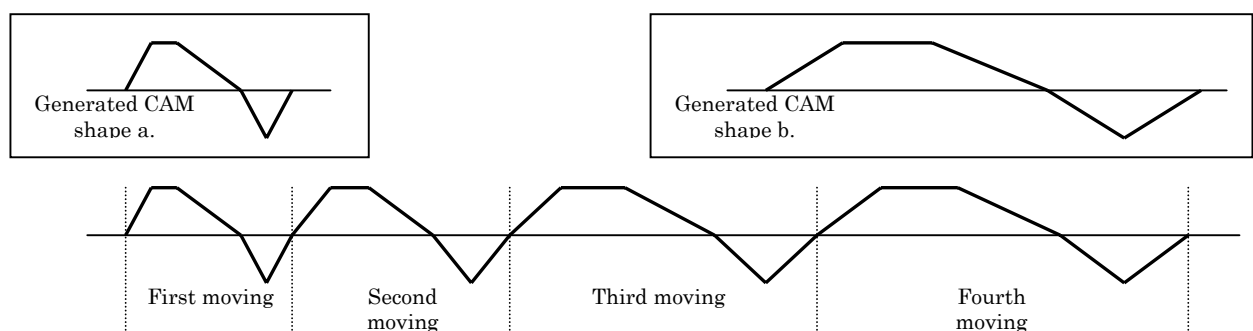


Fig. 5-3. Moving with changing

There are following limits as attention when CAM shape is generated.

Item	Limitation
Ratio of main axis and secondary axis	Set secondary axis pulses less 4 times than main axis pulses in one plot.
Increment of main axis pulses	Set less than 16,383 pulses for one plot increment.
Increment of secondary axis pulses	Set less than 32,767 pulses for one plot increment.
CAM moving without changing	Set less than 10KHz for encoder speed $\div$ increment of main axis pulses.
CAM moving with change	Set less than 6KHz for encoder speed $\div$ increment of main axis pulses.

## 5-2. Function of Program

This machine contains 32 programs of 256 steps.

The 32 programs are numbered 0 ~ 31 and managed. However, the program numbered 31 has limited use.

The program numbered 31 is executed automatically in case of an emergency. Therefore, an flexible response in case of an emergency is possible by having a special sequence stored in this #31 program.

Do not use #31 program when there is no need for it.

The program numbered 0 is executed when the program executing signal is inputted.

It is possible to branch from one program to another by using the condition of general purpose input and registers.

Moreover, it is possible to execute up to 8 programs in parallel by using the multi-tasking method.

Programs are generated by the PC and downloaded to the main controller by the RS-232C interface.

The program generating sequence is shown by the following flowchart.

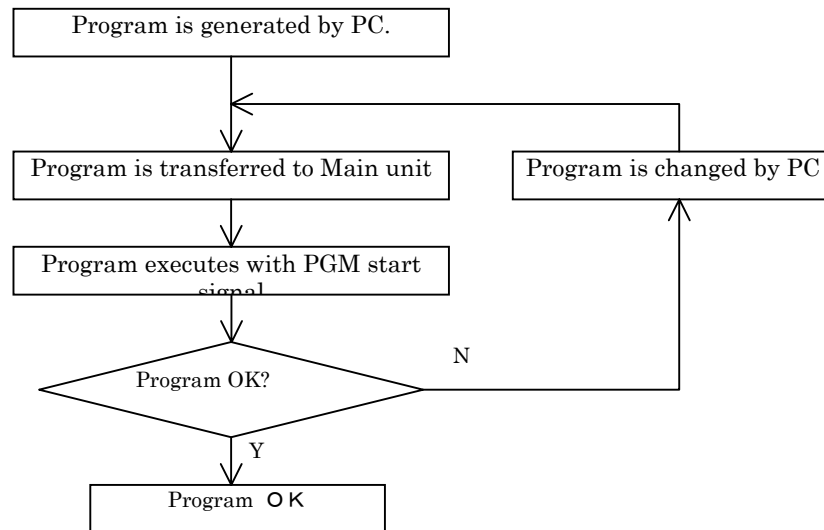


Fig. 5-4. Flowchart for generating programs

There are three types of commands used in program, namely, register commands, program commands, and program assisting commands.

**Registers:** These can memorize numbers and calculate arithmetic. These are used in combination with other commands.

**Programs:** These are commands to set up and execute actual operations.

**Program assist :** These are supplemental commands used to branch programs, etc.

The following is the list of commands.

Type	Command	Description
Register commands	REG_POS	Current position
	REG_R??	General purpose register (??: 00 to 99)
	REG_TM?	General purpose timer (? : 0 to 7)
	REG_OUT	General purpose output all terminals
	REG_IN	General purpose input all terminals
	REG_O??	General purpose output – terminal (?: 00 to 15)
	REG_I??	General purpose input – terminal (?: 00 to 15)
	REG_STS	Sensor status (B, A, CR, SON, INP, ALM, HOME, NEAR, REV, FOR)
	REG_SPI	Dedicated input (EMER, C_ST, P_ST)
	REG_SPO	Dedicated output (ERROR, MOVE, RDY)
	REG_ERR	Error status
REG_ENC	Encoder counter	
REG_SWT	Status of axis setting switch	
Program commands	MOV_PTP	PTP movement
	MOV_HOM	Home search movement
	MOV_SCN	Scan movement
	MOV_CPX	Complex trapezoidal movement
	MOV_CAM	CAM Movement
	SET_TBL	Set up coordinate system
	SET_MAX	Set up the maximum speed
	SET_MIN	Set up the start speed
	SET_ACC	Set up acceleration slope
	SET_DEC	Set up deceleration slope
	SET_ENC	Switching of internal or external encoder
	CPX_SP?	Set the Complex trapezoidal speed (? : 0 to 3)
	CPX_PS?	Set the Complex trapezoidal position (? : 0 to 3)
	STP_SDW	Decelerated stop
	STP_EMR	Instant stop
	STP_CAM	CAM moving stop
	STP_PAS	CAM moving temporary stop
	SRV_ON	Servo-on
SRV_OFF	Servo-off	
CLR_ON	Deviation counter reset output (One shot: 50ms.)	
ALL_RST	Reset (Recover to the power on status)	
MSK_SIN	Set up valid or invalid dedicated input (P_ST, C_ST, EMER)	
Program assist	PRG_STR	Program start
	PRG_END	Program end
	PRG_TSK	Multi task program start
	PRG_JMP	Program number jump
	PRG_LIN	Program line jump
	PRG_CAL	Program number call
	L??	Label
	WAIT (Conditional sentence)	Wait while a condition is being established (during true).
	IF(Conditional sentence) f1, f2	If the condition is true, execute f1; if it is false, execute f2.
	Conditional sentence of WAIT or IF	REG ??? is any of above registers. It is possible to enter a numeric value (decimal) directly. If the values of right side and left side are the same, it is true; if not, false. If the values of right side and left side are not the same, it is true; if not, false. If the left side is greater than the right side, it is true; if it is less, it is false. If the left side is the same or more than the right side, it is true; if not, it is false. If the left side is less than the right side, it is true; if it is more, it is false. false. If the left side is the same or less than the right side, it is true; if not, it is false. If the AND of the left and the right is other than 0, it is true; if it is 0, it is false. If the OR of the left and the right is other than 0, it is true; if it is 0, it is false.
	REG_???=REG_???	If the values of right side and left side are the same, it is true; if not, false.
	REG_???!=REG_???	If the values of right side and left side are not the same, it is true; if not, false.
	REG_???>REG_???	If the left side is greater than the right side, it is true; if it is less, it is false.
REG_???>=REG_???	If the left side is the same or more than the right side, it is true; if not, it is false.	
REG_???<REG_???	If the left side is less than the right side, it is true; if it is more, it is false.	
REG_???<=REG_???	false.	
REG_???&REG_???	If the left side is the same or less than the right side, it is true; if not, it is false.	
REG_??? REG_???	If the AND of the left and the right is other than 0, it is true; if it is 0, it is false. If the OR of the left and the right is other than 0, it is true; if it is 0, it is false.	
Calculation substitute	REG ??? is any of above registers. It is possible to enter a numeric value (decimal) directly. Addition Subtraction Multiplication Division AND of bits OR of bits	
REG_???=REG_??#+REG_???	Addition	
REG_???=REG_??#-REG_???	Subtraction	
REG_???=REG_??#*REG_???	Multiplication	
REG_???=REG_??#/REG_???	Division	
REG_???=REG_??#&REG_???	AND of bits	
REG_???=REG_??# REG_???	OR of bits	

### 5-3. Parameter setting function

The Parameter Setting Function is used to set up information necessary to operate this machine. A personal computer is used to set up conditions and the RS-232C interface is used to download them to the main body of the controller.

They are shown below.

Items set up	Contents	Default set
Speed mode	Low speed, Standard speed, High speed	Low speed
Shapes of slope	0, 1, 2, 3, 4, 5, 6, 7, 8	4
Motor type	Stepping motor / Servo motor	Stepping motor
Forward direction	CW / CCW	CCW
Clock type	2 clocks / one clock	2 clocks
Overrun logic	Positive logic / Negative logic	Positive logic
Near home logic	Positive logic / Negative logic	Positive logic
Home logic	Positive logic / Negative logic	Positive logic
Alarm logic	Positive logic / Negative logic	Positive logic
Backlash	With compensation / Without compensation	Without compensation
Encoder setting	Internal clock / External clock	Internal clock
Internal encoder speed	5 to 81,918Hz.	10000
Maximum speed	Speed mode, Low speed: 1 to 65535	10000
Start speed	Speed mode, Standard: 12.5 to 819187 Speed mode, High speed: 25 to 1638375Hz	100
Acceleration slope	Speed mode, Low speed: 10 to 1000	100
Deceleration slope	Speed mode, Standard: 12.5 to 12500 Speed mode, High speed: 250 to 25000Hz/ms	100
Pulse width of CAM move	0 to 12 $\mu$ s (500ns in case of 0)	10

The following is the detailed description of each parameter

#### (1) Speed mode

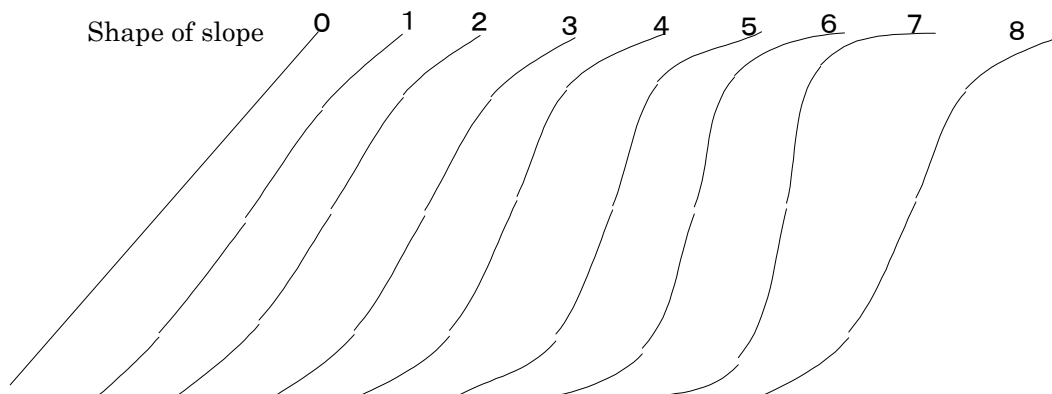
The pulse rate range which this controller can output is selected from low speed, standard speed and high speed.

The range is 1Hz~65535Hz in case of low speed, 12.5Hz~819187Hz in case of standard speed or 25Hz~1638375Hz in case of high speed.

#### (2) Shape of slope

There are 9 types of slope patterns, 0 ~ 8. One of them is selected.

The type 0 is a straight line. Types 1 through 7 are shown below. The type 8 is open to the user and available to set freely.



(3) Motor type

This sets motor type to control either stepping motor or servo motor.

When the servo motor is set, the in-position signal input becomes valid.

If the servo motor is set, when Home Search is completed, the Deviation Counter Reset signal (50ms. one shot) is automatically outputted.

(4) Forward direction

The forward direction is set either CW or CCW.

(5) Clock types

This switches 2 clock and 1 clock of clock output.

There are four types of CW/CCW output terminal as below table according to the combination of (4) above.

Forward direction	Clock type	Function of CW terminal	Function of CCW terminal
CW	2 clock	Clock output for forward direction	Clock output for reversed direction
	1 clock	Clock output	Direction output (On during forward)
CCW	2 clock	Clock output for reversed direction	Clock output for forward direction
	1 clock	Clock output	Direction output (On during reverse)

(6) Overrun logic, Near home logic, Home logic, Alarm logic

These set logic of each signal.

When the positive logic is set, A connection is made. (Photocoupler is active in conductive state).

When the negative logic is set, B connection is made. (Photocoupler is active in non-conductive state)

(7) Backlash

This sets whether the backlash compensation is used or not.

The backlash value is measured automatically when the first Home Search takes place after the power is turned on.

(8) Encoder setting

This sets whether the external encoder signal (A phase or B phase) is going to be used or the internal clock is used.

It is possible to use the single phase of A phase and B phase, the speed will be 1/2 of the phase difference signal. Please pay attention.

(9) Internal encoder speed

This becomes valid when the internal clock is set by (8) above.

The value set here takes place of the encoder signal.

(10) Maximum speed, Start speed, Acceleration slope, Deceleration slope

Speed and slope other than electronic CAM are set.

These settings are used to move if speed and slope are not set by program.

The range of setting differs according to the speed mode, however, when the set value exceeds the range, the machine moves within the range.

(11) Pulse width of CAM movement

The clock output is one shot output during cam movement. This sets the width of output pulse.

The setting is available within 0 to 12uS. If 0 is set, the pulse width is 0.5uS.

## 6. Command system

The commands used by this machine use simple language originated by us. There are three types of commands.

Type of commands	Description
Program	This includes actual moving, setting, operation, etc.
Auxiliary commands	This includes starting, ending, and branching of programs.
Register	This includes data to store and operate values and reflect the state of this machine to the programs. They include current position, the timer, I/O, and general purpose registers.

Each command is described in detail in the following format.

Form	This shows the format of a command.
Function	This outlines the function.
Description	This describes commands in detail. Related command (Related commands are described)
Example	This gives examples of a command.

## 6-1. MOV\_PTP

Form	MOV_PTP= $\pm$ d +9999999)	The value is specified directly. (d: -9999999~
	MOV_PTP=REG_R??	General purpose register is specified (?: 00~99)
Function	Coordinates are specified to determine a position.	
Description	<p>The coordinates can be specified by numerical values or by a general purpose register.</p> <p>When the coordinate system is specified in the absolute coordinates, the data specified is used as the absolute coordinates for moving a position.</p> <p>When the coordinate system is specified in the relative coordinates, the data specified sets the amount of movement. In case of + sign, it moves to forward direction. In case of - sign, it moves to reverse direction. Setting the absolute or relative coordinates can be switched using SET_TBL command.</p> <p>This command runs under the conditions set by the parameters or the value set by SET_MAX/MIN/ACC/DEC command of the program.</p> <p>Related command &lt;SET_TBL,SET_MAX/MIN/ACC/DEC&gt;</p>	
Example	<p>SET_TBL=0 (1)The coordinate system uses the absolute coordinates.</p> <p>MOV_PTP=+10 (2) Moves to the coordinates+1000.</p> <p>00 (3) Moves to thecoordinates-500.</p> <p>MOV_PTP=-500 (4) The relative coordinates is used as the coordinate system.</p> <p>SET_TBL=1 (5) Moves to the coordinates+500 (-500+1000)</p> <p>MOV_PTP=+10 (6) Moves to the coordinates+250 (+500-250)</p> <p>00</p> <p>MOV_PTP=-250</p>	

## 6-2. MOV\_HOM

Form	MOV_HOM=d (d: 0 to 3)
Function	The sequence specified is used to return to the Home Search.
Description	<p>The Home Search is executed according to the Home Search sequence specified by d.</p> <p>When this command ends normally, the current position (REG_POS) is cleared to 0.</p> <p>This command runs under the conditions set by the parameters or the value set by SET_MAX/MIN/ACC/DEC command of the program.</p> <p>The slope is set to the linear acceleration deceleration slope.</p> <p>Related commands &lt;SET_MAX/MIN/ACC/DEC&gt;</p>
Example	<p>MOV_HOM=1 (1) The Home Search sequence 1 is executed.</p> <p>MOV_PTP=+1000 (2) Moves to the coordinates+1000</p> <p>REG_POS=0 (3) The current position is set to 0.</p>



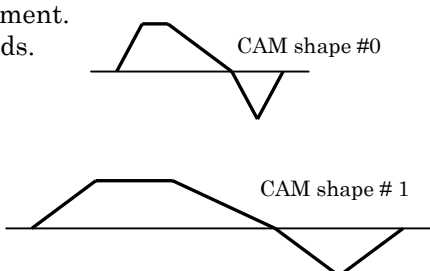
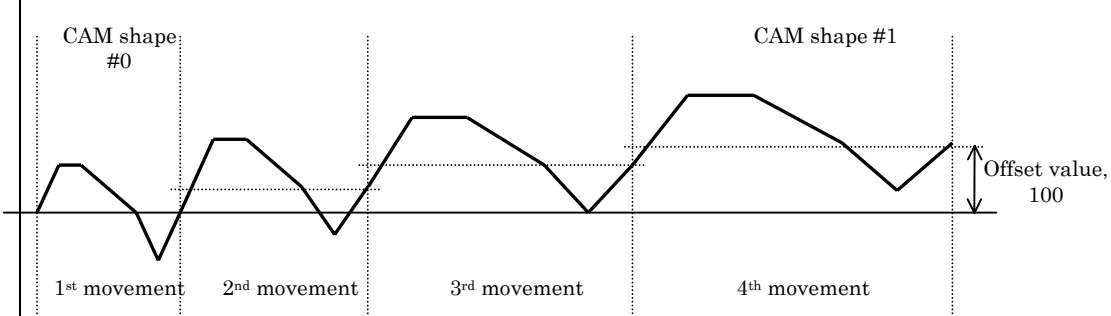
### 6-4. MOV\_CPX

Form	MOV_CPX=±	
Function	This executes the complex trapezoidal moving to the specified direction.	
Description	<p>The complex trapezoid drive specifies the speeds in three steps and position together with speed change as shown below.</p> <p>If the sign specified is +, the movement is in forward direction. If the sign is -, the movement is in reverse direction.</p> <p>The speed of this movement is set by CPX_SP1 /SP2/SP3 commands. The moving amount of this movement is set by CPX_PS1/PS2/PS3 commands. The starting speed and the slope are set by the parameter setting or by the SET_MAX/MIN /ACC /DEC commands. The acceleration and deceleration is liner.</p> <p>When the movement starts, the speed accelerates up to the speed of SP1. When the number of pulses of PS1 is reached, the speed changes in the direction of SP2. When the amount of movement of PS2 is reached, the speed changes in the direction of SP3. The total moving amount is PS1+PS2+PS3. The slopes of SP1- SP2 and SP2 – SP3 change according to the acceleration setting, SET_ACC</p> <p>Related commands &lt; CPX_SP1/SP2/SP3/PS1/PS2/PS3,SET_MIN/ACC/DEC &gt;</p>	
Example	<p>CPX_SP1=5000 CPX_SP2=10000    When the program CPX_SP3=3000    shown left is CPX_PS1=2500    executed, the image CPX_PS2=6000    shown to the right CPX_PS3=4000    results. MOV_CPX=+</p>	

6-5. MOV\_CAM (Without cam shape)

Form	MOV_CAM=d      CAM shape specified by direct numerical value (d: 0 – 31)	This goes to next line after moving stop.
	MOV_CAM=REG_R??    CAM shape specified by register (?: 00 – 31)	
	MOV_CAM=dE      CAM shape specified by direct numerical value (d: 0 – 31)	This specifies CAM movement and goes to next line.
	MOV_CAM=REG_R??E    CAM shape specified by register (?: 00 – 31)	
Function	This command specifies the CAM movement.	
Description	<p>The number of CAM shape is specified so that starting the CAM movement becomes possible.  The CAM shape number is specified either numerically directly or by the general purpose register.  Use numbers within 0~31 for the number of the CAM shape.</p> <p>When the CAM starting signal is inputted after this command is executed, actual movement will start.  When one CAM movement has ended, if the CAM starting signal is ON, the movement is repeated from the beginning.  If a CAM movement is to be finished, a STP_CAM command from another program is needed.  If there is no “E” at the end of a command, the machine will stop according to the stop command from another program and goes to the next step.  If there is an “E” at the end of a command, the next step follows as long as the machine is in operative condition.</p> <p>It is necessary that the shape of the CAM is generated and downloaded to the controller in advance.</p> <p>Related command &lt;STP_CAM&gt;</p>	
Example 1	<p>The below program uses multi tasking function to specify and stop the CAM movement.  This program specifies CAM shape number 8 and stops CAM movement when the general purpose input 00 is ON. This is series of moving.</p> <pre> PRG_STR=0            (1) Program No. 0 starts. PRG_TSK=10          (2) Program No. 10 starts. MOV_CAM=8           (3) CAM movement of the CAM shape No. 8 is specified. PRG_END=0           (4) Program No. 4 ends.  PRG_STR=10          (1) Program No. 10 starts. WAIT(REG_I00==0)   (2) The machine waits when the general purpose input 00 is OFF. It goes to the next line when 00 goes ON STP_CAM             (3) This stops CAM movement. PRG_END=10          (4) Program No. 10 ends. </pre>	
Example 2	<pre> PRG_STR=0            (1) Program No. 0 starts. MOV_CAM=8E          (2) CAM movement of the CAM shape No. 8 is specified. It goes to next step after it turns to CAM moving possible. WAIT(REG_I00==0)   (3) The machine waits when the general purpose input 00 terminal is OFF. It goes to the next line when 00 goes ON. STP_CAM             (4) This stops CAM movement. PRG_END=0           (5) Program No. 0 ends. </pre>	

6-6. MOV\_CAM (Shape change specified)

Form	<p>MOV_CAM=d1,d2,d3,d4      This does not proceed the next line until moving stop.</p> <p>MOV_CAM=d1,d2,d3,d4E      This specifies CAM movement and proceed the next line.</p>
Function	<p>This command moves with changing gradually from one CAM shape to another CAM shape</p>
Description	<p>This moves with changing from certain CAM shape to another CAM shape.</p> <p>d1 is CAM shape number when starting movement. (Setting range is 0 to 31)  d2 is final CAM shape number. (Setting is 0 to 31)  d3 specifies the times of changing from the movement start to the final CAM shape.  d4 specifies the offset value from the movement start to the final CAM shape.</p> <p>After execution of this command, the actual movement is started by CAM start signal input.  When the movement of the final CAM shape has ended, if the CAM starting signal is ON, the movement is repeated from the beginning.  If a CAM movement is to be finished, a STP_CAM command from another program is needed.  If there is no "E" at the end of a command, the machine will stop according to the stop command from another program and goes to the next step.  If there is an "E" at the end of a command, the next step follows as long as the machine is in operative condition.</p> <p>It is necessary that the shape of the CAM is generated and downloaded to the controller in advance.</p> <p>Related command &lt;STP_CAM&gt;</p>
Example	<p>PRG_STR=0      (1) Program number 0 starts.</p> <p>MOV_CAM=0,1,4,100E      (2) This specifies that starting CAM shape number is 0, final CAM shape number is 1, times are 4 and the offset is 100.</p> <p>WAIT(REG_I00==0)      (3) The machine waits when the general purpose input 00 terminal is OFF. It goes to the next line when 00 goes ON.</p> <p>STP_CAM      (4) This stops CAM movement.</p> <p>PRG_END=0      (5) Program number 0 ends.</p>  <p>It is assumed that the generated CAM shape number 0 and 1 are right shapes.</p> <p>The below is the image of above program, times are 4 and offset is 100.</p> 



6-7. SET\_TBL

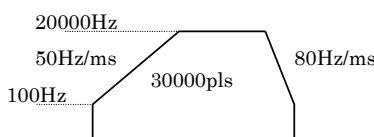
Form	MOV_TBL=d (d: 0 or 1)
Function	This sets absolute or incremental coordinate.
Description	<p>“d” is set to 0 in order to use the absolute coordinate.  “d” is set to 1 in order to use the incremental coordinate.</p> <p>This setting changes the coordinate system of MOV_PTP command.  The absolute coordinate system is in effect until this command is executed.</p> <p>Related command &lt;MOV_PTP&gt;</p>
Example	<p>SET_TBL=0 (1) Coordinates will use the absolute coordinates.  MOV_PTP=+1000 (2) Move to coordinates +1000  MOV_PTP=-500 (3) Move to coordinates -1000  SET_TBL=1 (4) The coordinate system will use the incremental coordinate.  MOV_PTP=+1000 (5) Move to coordinates +500 (-500+1000)  MOV_PTP=-250 (6) Move to coordinates +250 (+500-250)</p>

6-8. SET\_MAX

6-9. SET\_MIN

6-10. SET\_ACC

6-11. SET\_DEC

Form	<p>SET_MAX=d (d: 1 to 1638375)  SET_MIN=d (d: 1 to 1638375)  SET_ACC=d (d: 10 to 25000)  SET_DEC=d (d: 10 to 25000)</p>
Function	This sets moving speed and slope.
Description	<p>SET_MAX sets the maximum speed. (Unit: Hz)  SET_MIN sets the starting speed. (Unit: Hz)  SET_ACC sets the slope of acceleration (Unit: Hz)  SET_DEC sets the slope of deceleration (Unit: Hz)</p> <p>The speed and the slope set by the parameters are used until this command is executed.  This setting changes speed and slope of MOV_PTP, MOV_HOM, MOV_SCN, MOV_CPX commands.</p> <p>Related commands&lt;MOV_PTP,MOV_HOM,MOV_SCN,MOV_CPX&gt;</p>
Example	<p>SET_MAX=20000 (1) The maximum speed is set to 20000Hz.  SET_MIN=100 (2) The starting speed is set to 100Hz.  SET_ACC=50 (3) The acceleration slope is set to 50Hz/ms  SET_DEC=80 (4) The deceleration slope is set to 80Hz/ms.  MOV_PTP=+30000 (5) The machine moves to +30000.</p>  <p>The graph shows a trapezoidal velocity profile. It starts at a starting speed of 100Hz, accelerates linearly at a rate of 50Hz/ms to reach a maximum speed of 20000Hz. It then maintains this maximum speed for a distance of 30000 pulses before decelerating linearly at a rate of 80Hz/ms until it comes to a stop.</p>

6-12. SET\_ENC

Form	SET_ENC=d (d: 0 to 3)																			
Function	This sets counting type of encoder counter (REG_ENC).																			
Description	<p>The counting type of encoder counter and cam operation are set by this command as below.</p> <table border="1"> <thead> <tr> <th>Value of d</th> <th>Counter type of REG_ENC</th> <th>cam operation</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>One by one</td> <td>Internal clock used</td> <td rowspan="2">Any of two columns are set until this command is executed according to the parameter set</td> </tr> <tr> <td>1</td> <td>One by one</td> <td>External encoder used</td> </tr> <tr> <td>2</td> <td>Two-clock</td> <td>External encoder used</td> <td>A phase is + count. B phase is - count.</td> </tr> <tr> <td>3</td> <td>Four times</td> <td>External encoder used</td> <td>Available from Version 1.10 or after.</td> </tr> </tbody> </table> <p>Related command &lt;REG_ENC&gt;</p>	Value of d	Counter type of REG_ENC	cam operation	Description	0	One by one	Internal clock used	Any of two columns are set until this command is executed according to the parameter set	1	One by one	External encoder used	2	Two-clock	External encoder used	A phase is + count. B phase is - count.	3	Four times	External encoder used	Available from Version 1.10 or after.
Value of d	Counter type of REG_ENC	cam operation	Description																	
0	One by one	Internal clock used	Any of two columns are set until this command is executed according to the parameter set																	
1	One by one	External encoder used																		
2	Two-clock	External encoder used	A phase is + count. B phase is - count.																	
3	Four times	External encoder used	Available from Version 1.10 or after.																	

Example	SET_ENC=3 REG_ENC=0	(1) The encoder counter, REG_ENC counts four times. (2) The encoder counter becomes 0.
---------	------------------------	---

6-13. CPX\_SP1

6-14. CPX\_SP2

6-15. CPX\_SP3

Form	CPX_SP1=d (d: 1 to 1638375) CPX_SP2=d (d: 1 to 1638375) CPX_SP3=d (d: 1 to 1638375)
Function	This sets the speeds of the complex trapezoidal moving.
Description	<p>The complex trapezoidal moving specifies the speeds in three steps and position together with speed change as shown below.</p> <p>This command sets the speeds of three step movements. CPX_SP1 is the first step speed. CPX_SP2 is the second step. SPX_SP3 is the third. The range of setting is 1 to 1638375 in Hz. See MOV_CPX command for the operation for more.</p> <p>Related command &lt;MOV_CPX,CPX_PS1/PS2/PS3,SET_MIN/ACC/DEC&gt;</p>
Example	Refer MOV_CPX

6-16. CPX\_PS1

6-17. CPX\_PS2

6-18. CPX\_PS3

Form	CPX_PS1=d (d: 1 to 9999999) CPX_PS2=d (d: 1 to 9999999) CPX_PS3=d (d: 1 to 9999999)
Function	This sets the moving amount of the speed switching of the complex trapezoidal system.
Description	<p>The complex trapezoidal moving specifies the speeds in three steps and position together with speed change as shown below.</p> <p>This command sets the moving amount at the speed switching point. CPX_PS1 determines the moving amount when starting to change speed in the direction of the second step. CPX_PS2 determines the moving amount when starting to change speed in the direction of the third step. CPX_PS3 determines the amount of the last movement. The total moving amount is CPX_PS1+CPX_PS2+CPX_PS3.</p> <p>See MOV_CPX command for the contents of the operation.</p> <p>Related command &lt;MOV_CPX,CPX_SP1/SP2/SP3,SET_MIN/ACC/DEC&gt;</p>
Example	Refer MOV_CPX

6-19. STP\_SDW

6-20. STP\_EMR

Form	STP_SDW STP_EMR
Function	If the machine is in motion, these commands decelerate and stop or stop immediately.
Description	<p>When STP_SDW is executed while the machine is in operation, the machine will decelerate and stop.</p> <p>When STP_EMR is executed while the machine is in operation, the machine will stop immediately.</p> <p>No action will be taken if the machine is not operating.</p> <p>These commands will stop the operation of MOV_PTP, MOV_HOM, MOV_SCN and MOV_CPX.</p> <p>In order to stop the CAM operation, STP_CAM should be used.</p> <p>Related commands &lt;MOV_PTP/HOM/SCN/CPX&gt;</p>
Example	<pre> PRG_STR=0          (1) This starts the Program No. 0. MOV_SCN=+E        (2) This starts scanning for forward direction. WAIT(REG_I00==(   (3) The machine waits while the general purpose input 00 IF(REG_I01==1)     terminal is OFF. (It goes to next step when the above goes to STP_SDW,STP_E      ON.) MR                (4) If the general purpose input terminal 01 is ON, the machine PRG_END=15        will decelerate and stop. If it is OFF, it will stop                   immediately.                   (5) This ends the program No. 0. </pre> <p>While scanning, if the general purpose input 00 goes ON, this program stops instantly or stops with deceleration after checking the general purpose input 01.</p>

6-21. STP\_CAM

6-22. STP\_PAS

Form	STP_CAM STP_PAS=d (d: 0 or 1)
Function	This stops or stops temporarily the CAM movement
Description	<p>When STP_CAM is executed while the machine is in operation, the machine will stop instantly and end CAM movement.</p> <p>STP_PAS is a command to stop temporarily and resume CAM movement.</p> <p>STP_PAS=1 stops temporarily CAM movement and STP_PAS=0 releases the temporary stop and resumes CAM movement.</p> <p>When the machine is not operation, this command will not do anything.</p> <p>This command stops only the movement of MOV_CAM command.</p> <p>Use STP_SDW or STP_EMR command to stop the movement excepting CAM movement.</p> <p>Related command &lt;MOV_CAM&gt;</p>
Example	<p>PRG_STR=0 (1) This starts program #0.</p> <p>MOV_CAM=5E (2) This specifies the movement of CAM shape #5.</p> <p>L00 (3) Label, L00</p> <p>IF(REG_I00==0)L02,L01 (4) If the universal input terminal 00 is OFF, it goes to label L02. If it is ON, it goes to label L01.</p> <p>L01 (5) Label 01</p> <p>STP_PAS=1 (6) This stops the CAM movement temporarily.</p> <p>WAIT(REG_I00==1) (7) This waits while the universal input 00 terminal is OFF. (It goes to next step when the above goes to ON.)</p> <p>STP_PAS=0 (8) This releases the temporary stop of CAM movement and resumes the movement.</p> <p>L02 (9) Label L02</p> <p>IF(REG_I01==0)L00,L03 (10) If the universal input terminal 01 is OFF, it goes to label L02. If it is ON, it goes to label L03.</p> <p>L03 (11) Label L03</p> <p>STP_CAM (12) This stops and ends CAM movement.</p> <p>PRG_END=15 (13) This ends the program #0.</p> <p>This program stops temporarily and resumes the CAM movement by the universal input terminal 00 and ends CAM movement if the universal input 01 goes ON.</p>

## 6-23. SRV\_ON

## 6-24. SRV\_OFF

Form	SRV_ON SRV_OFF
Function	The servo-ON output is turned ON or OFF.
Description	The SRV_ON command turns the servo ON (SON) ON. The SRV_OFF command turns the servo ON (SON) OFF.  The servo ON output turns ON automatically after the power is turned ON and it goes available to use. Please use this command when the servo ON output needs operating. This command can be used as a general purpose output also.  Related command <CLR_ON>
Example	SRV_OFF (1) This turns OFF the servo ON signal. WAIT(REG_I02==0) (2) This waits as long as the general purpose input 02 terminal is OFF. SRV_ON (3) This turns ON the servo ON signal.

## 6-25. CLR\_ON

Form	CLR_ON
Function	The deviation counter RESET output is turned ON using a one-shot.
Description	This turns ON the deviation counter RESET output (CR) used for the servo motor. The above signal is turned ON approximately 50ms. using the one-shot output. The deviation counter RESET output is also outputted automatically when the Home Search is completed.  Related Command <SRV_ON/OFF>
Example	PRG_STR=31 (1) This starts the Program No. 31. (Program No. 31 is used for the error sequence) CLR_ON (2) The deviation counter RESET output is outputted for 50ms. REG_ERR=0 (3) This resets the error. REG_O03=1 (4) This turns ON the general purpose output 03 terminal. PRG_END=31 (5) This ends the program.

## 6-26. ALL\_RST

Form	ALL_RST
Function	This is used to reset the machine to the state when the power is turned ON.
Description	When this command is executed, the machine is placed in the state when the power is turned ON.
Example	PRG_STR=31 (1) This starts the Program No. 31. (Program No. 31 is used for the error sequence) ALL_RST (2) The machine is reset to the state when the power is turned ON. PRG_END=31 (3) This ends the program.

### 6-27. MSK\_SIN

Form	MSK_SIN=d (d: 0 or 1)
Function	This is used to set if the special inputs (P_ST, C_ST, EMER) are valid or not.
Description	This is used to set if the signals of Program Start (P_ST), the CAM Start (C_ST), and the Emergency Stop (EMER) are valid or not. When d is 0, it is valid; when d is 1, the special input is invalid. This is set as valid when the power is turned ON. This can be used, for instance, when the special input needs to be ignored, as in the case of the online operation.
Example	MSK_SIN=1 (1) The special input becomes invalid. REG_TM2=1000 (2) 1000 is set to the Timer Register 2. WAIT(REG_TM2>0) (3) The machine waits until the Timer Register 2 goes to 0 SIN_MSK=0 (Wait 1 second). (4) The special input becomes valid.

### 6-28. PRG\_STR

### 6-29. PRG\_END

Form	PRG_STR=d (d: 0 to 31) PRG_END=d (d: 0 to 31)
Function	This command starts or ends the program.
Description	The program must start with the PRG_STR command and end with the PRG_END command. “d” is the program number and its range is 0 to 31.
Example	PRG_STR=10 (1) Program No. 10 starts. MOV_PTP=+1000 (2) Moving +1000 takes place. REG_O02=1 (3) The general purpose output 02 terminal is turned ON. PRG_END=10 (4) The Program No. 10 ends.

### 6-30. PRG\_TSK

Form	PRG_TSK=d,d,d,d,d,d,d,d (d: 0 to 31)
Function	Multiple programs are started using the multitask command.
Description	This unit is capable of starting multiple programs simultaneously in parallel. Eight programs can be operated in parallel at the same time. “d” is the program number. A command using one line can start seven programs.
Example	PRG_STR=0 (1) Program No. 0 starts. PRG_TSK=1,2,3 (2) Programs, 1, 2, and 3 are operated in parallel. WAIT(REG_I05==0) (3) The controller waits as long as the general purpose input PRG_TSK=4,5 05 terminal is OFF. (Programs 1, 2, and 3 are being executed PRG_END=0 in succession.) (4) Programs No. 4 and 5 are operated in parallel. (5) The Program No. 0 ends.  In the above example, Programs 1, 2, and 3 are operated on and Programs 4 and 5 are operated on after the general purpose input 05 terminal is turned ON. Above (5) ends the Program No. 0 but other programs, 1, 2, 3, 4, and 5 are executed in succession until ends of each program.

### 6-31. PRG\_JMP

Form	PRG_JMP=d (d: 0 to 31) PRG_JMP=REG_R?? (?: 0 to 99)
Function	This is used to jump to another program number.
Description	This causes the program to jump to the beginning of the program specified. The program number can be specified directly by the numerical value (d) or by a general purpose register. “d” is program number and the range of d is 0 to 31. Use PRG_LIN command when a jump takes place within the same program.  Related command <PRG_STR,PRG_LIN,PRG_CAL>
Example	<pre>PRG_STR=0 IF(REG_I00==0)REG_R05=8,REG_R05=10 PRG_JMP=REG_R05 PRG_END=0</pre> <p>(1) Program No. 0 starts. (2) If the general purpose input 00 terminal is OFF, enter 8 to the general purpose register 05; if it is ON, enter 10 to the same. (3) Jump to the program specified by the value of the general purpose register REG_R05. (4) The Program No. 0 ends.</p> <p>In the above example, a jump takes place to No. 8 program if the general purpose input 00 terminal is OFF; it will jump to program No. 10 if it is ON.</p>

### 6-32. PRG\_LIN

Form	PRG_LIN=L?? (?: 00 to 99)
Function	This jump is used to jump to the labeled line specified within the same program.
Description	This is used to execute a program by jumping to the labeled line specified within the same program. The label can be specified within L00 to L99. If there is no label specified, it is an error. Use PRG_JMP command if the jump is to another program number.  Related command <PRG_JMP,PRG_CAL,L??>
Example	<pre>PRG_STR=0 IF(REG_I04==0)L00,L01 L00 MOV_PTP=+100 PRG_LIN=L02 L01 MOV_PTP=-100 L02 PRG_END=0</pre> <p>(1) Program No. 0 starts. (2) If the general purpose input 04 terminal is OFF, jump to L00; if it is ON, jump to L01. (3) Label L00 (4) Move +100 (5) Jump to Label L02. (6) Label L01 (7) Move -100 (8) Label L02 (9) The Program No. 0 ends.</p> <p>In the example above, moving of +100 takes place if the general purpose input 04 terminal is OFF; if it is ON, moving to -200 will take place.</p>

### 6-33. PRG\_CAL

Form	PRG_CAL=d (d: 0 to 31) PRG_CAL=REG_R?? (?: 00 to 99)
Function	This is used to call a specified program number.
Description	This calls a specified program number. When the program called is executed completely, the program returns to its original program. The program number can be specified directly by the numerical value (d) or by a general purpose register. “d” is program number and the range of d is 0 to 31. The nesting is 1.  Related command <PRG_STR, PRG_JMP, PRG_LIN>
Example	PRG_STR=0 (1) Program No. 0 starts. PRG_CAL=5 (2) Program No. 5 is called. (The program of No. 5 is executed.) PRG_CAL=10 (3) Program No. 10 is called. (The program of No. 10 is executed.) PRG_END=0 (4) The Program No. 0 ends.

### 6-34. L00 to L99

Form	L?? (?: 00 to 99)
Function	This sets labels.
Description	This sets labels. This is used to set the labels which point to the place where programs will branch. This program itself does not do anything.  Related command <PRG_LIN, IF>
Example	PRG_STR=0 (1) Program No. 0 starts. IF(REG_I04==0)L00,L01 (2) The program jumps to L00 if the general purpose input 04 terminal is OFF; If it is ON, the program will jump to L01. L00 (3) Label L00 MOV_PTP=+100 (4) Moves +100 PRG_LIN=L02 (5) The program jumps to Label 02. L01 (6) Label L01 MOV_PTP=-100 (7) Moves -100 L02 (8) Label L02 PRG_END=0 (9) The Program No. 0 ends.  In the above example, if the general purpose input 04 terminal is OFF, +100 moves; if ON, -200 moves.

6-35. WAIT

Form	WAIT (Conditional expression)																										
Function	The program waits while the conditions are being established.																										
Description	<p>This command makes the program wait while the conditions of () are being established. If the conditions fail to be established, the program of the following line is executed.</p> <p>The conditional expression is composed of the “right side”, the “conditional symbol”, and the “right side” and required to enclose by (). The conditional symbols are listed below.</p> <ul style="list-style-type: none"> <li>== The conditions are established if the left and the right sides are the same. If they are different, the conditions are not established.</li> <li>!= The conditions are established if the left and the right sides are different. If they are the same, the conditions are not established.</li> <li>&gt; The conditions are established if the left side is larger than the right sides. If the left side is less than the right side, the conditions are not established.</li> <li>&gt;= The conditions are established if the left side is equal or larger than the right sides. If the left side is less than the right side, the conditions are not established.</li> <li>&lt; The conditions are established if the left side is less than the right sides. If the left side is equal or larger than the right side, the conditions are not established.</li> <li>&lt;= The conditions are established if the left side is equal or less than the right sides. If the left side is larger than the right side, the conditions are not established.</li> <li>&amp; The conditions are established if the AND of each bit of the left side and the right side is other than 0. If it is 0, the conditions are not established.</li> <li>  The conditions are established if the OR of each bit of the left side and the right side is other than 0. If it is 0, the conditions are not established.</li> </ul> <p>The “left side” and the “right side” will have registers or numerical values. The registers include the following.</p> <table border="0"> <tr><td>REG_POS</td><td>Current position</td></tr> <tr><td>REG_R00 to R99</td><td>General purpose register</td></tr> <tr><td>REG_TM0 to TM7</td><td>Timer</td></tr> <tr><td>REG_OUT</td><td>General purpose output register (All terminals)</td></tr> <tr><td>REG_O00 to O15</td><td>General purpose output register (One terminal)</td></tr> <tr><td>REG_IN</td><td>General purpose input register (All terminals)</td></tr> <tr><td>REG_I00 to I15</td><td>General purpose input register (One terminal)</td></tr> <tr><td>REG_STS</td><td>Information related to the sensor and driver control signals.</td></tr> <tr><td>REG_SPI</td><td>Special purpose input</td></tr> <tr><td>REG_SPO</td><td>Special purpose output</td></tr> <tr><td>REG_ERR</td><td>Contents of error</td></tr> <tr><td>REG_ENC</td><td>Encoder counter</td></tr> <tr><td>REG_SWT</td><td>Switch to set axis</td></tr> </table> <p>Related command &lt;IF, REG_??&gt;</p>	REG_POS	Current position	REG_R00 to R99	General purpose register	REG_TM0 to TM7	Timer	REG_OUT	General purpose output register (All terminals)	REG_O00 to O15	General purpose output register (One terminal)	REG_IN	General purpose input register (All terminals)	REG_I00 to I15	General purpose input register (One terminal)	REG_STS	Information related to the sensor and driver control signals.	REG_SPI	Special purpose input	REG_SPO	Special purpose output	REG_ERR	Contents of error	REG_ENC	Encoder counter	REG_SWT	Switch to set axis
REG_POS	Current position																										
REG_R00 to R99	General purpose register																										
REG_TM0 to TM7	Timer																										
REG_OUT	General purpose output register (All terminals)																										
REG_O00 to O15	General purpose output register (One terminal)																										
REG_IN	General purpose input register (All terminals)																										
REG_I00 to I15	General purpose input register (One terminal)																										
REG_STS	Information related to the sensor and driver control signals.																										
REG_SPI	Special purpose input																										
REG_SPO	Special purpose output																										
REG_ERR	Contents of error																										
REG_ENC	Encoder counter																										
REG_SWT	Switch to set axis																										

Example	<pre> PRG_STR=0 REG_TM5=1000 WAIT(REG_TM5&gt;0) WAIT(REG_I02==1) MOV_SCN=+E WAIT(REG_POS&lt;100 00) STP_SDW PRG_END=0 </pre>	<ol style="list-style-type: none"> <li>(1) The program No 0 starts.</li> <li>(2) 1000 is placed in the Timer Register REG_TM5.</li> <li>(3) The machine waits until REG_TM5 becomes 0. (One second waiting) REG_TM5 is made -1 every 1ms.)</li> <li>(4) The machine waits as long as the general purpose input terminal 02 stays ON.</li> <li>(5) Scanning to forward direction</li> <li>(6) The machine waits until the current position is bigger than 10,000.</li> <li>(7) Decelerated and stop.</li> <li>(8) The Program No. 0 ends.</li> </ol>
---------	--	---

### 6-36. IF

Form	IF(Conditional Expression ) f1,f2 (f1 and f2 are a command) IF(Conditional Expression ) L??.L?? (L?? is a label)
Function	Program branches according to conditions.
Description	<p>When a condition is established, f1 is executed. If a condition is not established, f2 is executed. If f1 and f2 are labels (L00~99), the program jumps to that label line.</p> <p>If f1 and f2 are other than labels, the IF command uses an area for three lines. Therefore, it should be warned that the number of steps should not exceed the limit when making programs.</p> <p>The conditional expression is composed of the “right side”, the “conditional symbol”, and the “right side” and required to enclose by (). The conditional symbols are listed below.</p> <ul style="list-style-type: none"> <li>== The conditions are established if the left and the right sides are the same. If they are different, the conditions are not established.</li> <li>!= The conditions are established if the left and the right sides are different. If they are the same, the conditions are not established.</li> <li>&gt; The conditions are established if the left side is larger than the right sides. If the left side is less than the right side, the conditions are not established.</li> <li>&gt;= The conditions are established if the left side is equal or larger than the right sides. If the left side is less than the right side, the conditions are not established.</li> <li>&lt; The conditions are established if the left side is less than the right sides. If the left side is equal or larger than the right side, the conditions are not established.</li> <li>&lt;= The conditions are established if the left side is equal or less than the right sides. If the left side is larger than the right side, the conditions are not established.</li> <li>&amp; The conditions are established if the AND of each bit of the left side and the right side is other than 0. If it is 0, the conditions are not established.</li> <li>  The conditions are established if the OR of each bit of the left side and the right side is other than 0. If it is 0, the conditions are not established.</li> </ul> <p>The “left side” and the “right side” will have registers or numerical values. The registers include the following.</p> <ul style="list-style-type: none"> <li>REG_POS Current position</li> <li>REG_R00 to R99 General purpose register</li> <li>REG_TM0 to TM7 Timer</li> <li>REG_OUT General purpose output register (All terminals)</li> <li>REG_O00 to O15 General purpose output register (One terminal)</li> <li>REG_IN General purpose input register (All terminals)</li> <li>REG_I00 to I15 General purpose input register (One terminal)</li> <li>REG_STS Information related to the sensor and driver control signals.</li> <li>REG_SPI Special purpose input</li> <li>REG_SPO Special purpose output</li> <li>REG_ERR Contents of error</li> <li>REG_ENC Encoder counter</li> <li>REG_SWT Switch to set axis</li> </ul> <p>Related command &lt;WAIT, REG_??&gt;</p>

Example	<pre> PRG_STR=0 IF(REG_I00==0)MOV_PTP=500,MOV_PTP=800 MOV_SCN=-E L00 IF(REG_POS&lt;100)L01,L00 L01 STP_SDW PRG_END=0 </pre>	<p>(1) The program No 0 starts.</p> <p>(2) If the general purpose input 00 terminal is OFF, moving 500 will take place; if it is ON, moving 800 will take place.</p> <p>(3) Scanning in the reverse direction is started.</p> <p>(4) Label L00</p> <p>(5) If the current position is less than 100, go to L01; if it is more than 100, go to L00.</p> <p>(6) Label L01</p> <p>(7) Decelerated and stop.</p> <p>(8) The Program No. 0 ends.</p>
---------	---	--

### 6-37. REG\_POS.

Form	REG_POS
Function	This register indicates the current position.
Description	<p>This register indicates the current position. It shows 0 when the power is turned ON.</p> <p>Moreover, it is cleared to 0 when the Home Return is completed.</p> <p>The range allowed to this register is -2147483648 ~ +2147483647.</p> <p>This is used for the arithmetic calculation with other registers and numerical values and conditional branching.</p> <p>When REG_POS is used singly, the value of this register is sent by RS-232C interface.</p> <p>The form of transmission uses REG_POS=d. The d is the value of the register.</p> <p>Related command &lt;WAIT, IF, REG_??&gt;</p>
Example	<pre> MOV_HOM=1 REG_POS=-100 REG_POS REG_R00=REG_POS+1000 </pre> <p>(1) This performs the Home Search. The current position is cleared to 0.</p> <p>(2) The current position makes -100. (It is not moved.)</p> <p>(3) The current position is sent to the host using RS-232C.(REG_POS=-100 is set)</p> <p>(4) 1000 is added to the current position and assigned to the general purpose register REG_R00.</p>

### 6-38. REG\_R00 to R99

Form	REG_R?? (??: 00 to 99)
Function	This register can be used in the manner the general purpose registers are used.
Description	<p>This register can be used in the manner the general purpose registers are used.</p> <p>This unit uses 100 general purpose registers. ?? of REG_R?? is replaced by numbers 00~99.</p> <p>The range of this register is -2147483648~2147483647.</p> <p>This is used for the arithmetic calculation with other registers and numerical values and also conditional branching.</p> <p>When REG_R?? is used singly, the value of this register is sent through RS-232C interface.</p> <p>The form of transmission uses REG_R??=d. The d is the value of the register.</p> <p>Related command &lt;WAIT, IF, REG_??&gt;</p>

Example	REG_R0=2000 REG_R10=REG_POS+100 REG_R10 WAIT(REG_R20<=500)	<ol style="list-style-type: none"><li>(1) 2000 is assigned to the general purpose register REG_R00.</li><li>(2) +100 is added to the current position and assigned to the general purpose register REG_R10.</li><li>(3) The value of general purpose register REG_R10 is sent to the host using RS-232C.</li><li>(4) While the value of general purpose register REG_R20 is less than 500, this unit waits.</li></ol>
---------	---	---

6-39. REG\_TM0 to TM7

Form	REG_TM? (?: 0 to 7)
Function	This register functions as a timer.
Description	<p>This register functions as a timer. There are 8 registers using REG_TM?. ?=0~7. These timer registers operate independent of each other. Numbers are reduced by 1 every ms. When the value of the timer goes to 0, the operation stops.</p> <p>The range of values these registers can handle is 0 to 2147483647.</p> <p>This is used for the arithmetic calculation with other registers and numerical values and also conditional branching.</p> <p>When REG_TM? is used singly, the value of this register is sent through RS-232C interface.</p> <p>The form of transmission uses REG_TM?=d. The d is the value of the register.</p> <p>Related command &lt;WAIT, IF, REG_??&gt;</p>
Example	<p>REG_TM5=2000 (1) 2000 is assigned to the Timer Register REG_TM5.</p> <p>WAIT(REG_TM5&gt;0) (2) This unit waits until the Timer Register REG_TM5 becomes 0. (Because 2000 is assigned at (1), this unit waits 2 seconds.)</p>

6-40. REG\_O00 to 15

Form	REG_O?? (??: 00 to 15)
Function	This register is used to indicate the terminals of the general purpose outputs.
Description	<p>This unit has 16 general purpose output terminals. This register is used to indicate each terminal (1 terminal only)</p> <p>?? of REG_O?? is the terminal number. The numbers, 00 to 15 are used.</p> <p>When the general purpose output terminal concerned is ON, the value is 1; when it is OFF, the value is 0.</p> <p>When 1 is assigned to this register, the output terminal goes ON; when 0 is assigned, it goes OFF.</p> <p>The range of values this register can handle is 0 and 1 only.</p> <p>If all terminals are wished to be handled in a lump, please use REG_OUT.</p> <p>This is used for the arithmetic calculation with other registers and numerical values and also with the conditional branching.</p> <p>When only REG_O?? Is used singly, the value of this register is sent through RS-232C interface.</p> <p>The form of transmission uses REG_O??=d. The d is the value of the register.</p> <p>Related command &lt;WAIT, IF, REG_??&gt;</p>
Example	<p>REG_O01=1 (1) This turns ON the general purpose output terminal 01.</p> <p>REG_O14=0 (2) This turns OFF the general purpose output terminal 04.</p>

### 6-41. REG\_OUT

Form	REG_OUT
Function	This register indicates all the general purpose terminals.
Description	<p>This unit has 16 general purpose output terminals. This register indicates all the terminals.</p> <p>The general purpose output terminals 00~15 correspond to bits 0~15. When they are ON, it is 1; when they are OFF, it is 0.</p> <p>The range of values this register can handle is 0 to 65535.</p> <p>When only one terminal is to be operated on, REG_O?? is used.</p> <p>This is used for the arithmetic calculation with other registers and numerical values and also with the conditional branching.</p> <p>When only REG_OUT is used singly, the value of this register is sent through RS-232C interface.</p> <p>The form of transmission uses REG_OUT=d. The d is the value of the register.</p> <p>Related command &lt;WAIT, IF, REG_??&gt;</p>
Example	<p>REG_OUT=65535                   (1) All the general purpose output terminals go ON.</p> <p>REG_OUT=0                       (2) All the general purpose output terminals go OFF.</p> <p>REG_OUT=REG_OUT 21           (3) Terminals 00 and 02 and 04 of the general purpose output go ON.(21=00010101B)</p> <p>REG_OUT=REG_OUT&amp;16           (4) All the general purpose output terminals go OFF except 04 terminal. (16=00010000B)</p>

### 6-42. REG\_I00 to 15

Form	REG_I??                   (??: 00 to 15)
Function	This register indicates individual general purpose input terminals
Description	<p>This unit has 16 general purpose input terminals. This register indicates each terminal (1 terminal only) of them.</p> <p>?? of REG_I?? is the terminal number and 00~15 are used.</p> <p>When the specified general input terminal is ON, it is 1; when it's terminal is OFF, it is 0.</p> <p>This register is for reading only. Values can not be assigned.</p> <p>The range of values this register can handle is limited to 0 or 1.</p> <p>Use REG_IN when all terminals are to be operated in a lump.</p> <p>This is used for the arithmetic calculation with other registers and numerical values and also with the conditional branching.</p> <p>When only REG_I?? is used singly, the value of this register is sent through RS-232C interface.</p> <p>The form of transmission uses REG_I??=d. The d is the value of the register.</p> <p>Related command &lt;WAIT, IF, REG_??&gt;</p>
Example	<p>REG_R02=REG_I01                   (1) The state of the general purpose input terminal 01 is assigned to the general purpose register REG_R02.</p> <p>WAIT(REG_I02==0)                   (2) This unit waits as long as the general purpose input terminal 02 is OFF.</p> <p>IF(REG_I07==0)MOV_PTP=+30,MOV_PTP=-30           (3) When the general purpose input terminal 07 is OFF, +30 movement takes place; when it is ON, -30 movement takes place.</p>



6-43. REG\_IN

Form	REG_IN
Function	This register indicates all terminals of the general purpose input.
Description	<p>This unit has 16 general purpose input terminals. This register indicates the status of all terminals.</p> <p>The register numbers of the general purpose output terminals 00~15 correspond to bits 0~15. When they are ON, it is 1; when they are OFF, it is 0.</p> <p>This register is for reading only. Values can not be assigned.</p> <p>The range of values this register can handle is 0 to 65535.</p> <p>When only one terminal is to be operated on, REG_I?? is used.</p> <p>This is used for the arithmetic calculation with other registers and numerical values and also with the conditional branching.</p> <p>When only REG_IN is used singly, the value of this register is sent through RS-232C interface.</p> <p>The form of transmission uses REG_IN=d. The d is the value of the register.</p> <p>Related command &lt;WAIT, IF, REG_??&gt;</p>
Example	<p>REG_R00=REG_IN (1) The state of the general purpose input terminal is assigned to the general purpose register, REG_R00.</p> <p>WAIT(REG_IN!=0) (4) This unit waits until all the general purpose inputs are OFF.</p>

6-44. REG\_STS

Form	REG_STS																								
Function	This register indicates the status of the sensor of the controller.																								
Description	<p>This register indicates the status of the sensor of the controller. The function of bits is shown below.</p> <table border="1" style="margin-left: 40px;"> <tr> <td>Bit Assignment</td> <td>bit9</td> <td>bit8</td> <td>bit7</td> <td>bit6</td> <td>bit5</td> </tr> <tr> <td>Signal name</td> <td>B phase</td> <td>A phase</td> <td>Counter reset</td> <td>Servo-on</td> <td>In-position</td> </tr> <tr> <td>Bit Assignment</td> <td>bit4</td> <td>bit3</td> <td>bit2</td> <td>bit1</td> <td>bit0</td> </tr> <tr> <td>Signal name</td> <td>Alarm</td> <td>Home</td> <td>Near home</td> <td>Reverse</td> <td>Forward</td> </tr> </table> <p>When each signal is active, the corresponding bit is 1.</p> <p>This register is for reading only. Values can not be assigned.</p> <p>The range of values this register can handle is 0 to 1023.</p> <p>This is used for the arithmetic calculation with other registers and numerical values and also with the conditional branching.</p> <p>When only REG_STS is used singly, the value of this register is sent through RS-232C interface.</p> <p>The form of transmission uses REG_STS=d. The d is the value of the register.</p> <p>Related command &lt;WAIT, IF, REG_??&gt;</p>	Bit Assignment	bit9	bit8	bit7	bit6	bit5	Signal name	B phase	A phase	Counter reset	Servo-on	In-position	Bit Assignment	bit4	bit3	bit2	bit1	bit0	Signal name	Alarm	Home	Near home	Reverse	Forward
Bit Assignment	bit9	bit8	bit7	bit6	bit5																				
Signal name	B phase	A phase	Counter reset	Servo-on	In-position																				
Bit Assignment	bit4	bit3	bit2	bit1	bit0																				
Signal name	Alarm	Home	Near home	Reverse	Forward																				
Example	<p>REG_R00=REG_STS (1) The status of the sensor is assigned to the general purpose register REG_R00.</p> <p>WAIT(REG_STS&amp;4) (2) This unit waits as long as the Near Home is ON.</p>																								

### 6-45. REG\_SPI

Form	REG_SPI								
Function	This register indicates the status of the dedicated input.								
Description	<p>This register has the following bit assignment showing the status of these dedicated inputs.</p> <table border="1"> <thead> <tr> <th>Bit</th> <th>bit2</th> <th>bit1</th> <th>bit0</th> </tr> </thead> <tbody> <tr> <td>Signal name</td> <td>Emergency stop</td> <td>CAM start</td> <td>Program start</td> </tr> </tbody> </table> <p>When each signal is ON, the corresponding bit is 1.  This register is used exclusively for reading and any values can not be assigned to it.  The range of values this register can handle is 0 to 7.</p> <p>This is used for the arithmetic calculation with other registers and numerical values and also with the conditional branching.  When only REG_SPI is used singly, the value of this register is sent through RS-232C interface.  The form of transmission uses REG_SPI=d. The d is the value of the register.</p> <p>Related command &lt;WAIT, IF, REG_??&gt;</p>	Bit	bit2	bit1	bit0	Signal name	Emergency stop	CAM start	Program start
Bit	bit2	bit1	bit0						
Signal name	Emergency stop	CAM start	Program start						
Example	<p>REG_R00=REG_SPI (1) The status of the dedicated input is assigned to the general purpose register, REG_R00.  WAIT(REG_SPI&amp;2) (2) This unit waits as long as the CAM activating signal is ON.</p>								

### 6-46. REG\_SPO

Form	REG_SPO								
Function	This register indicates the status of the dedicated output of this unit.								
Description	<p>This register indicates the status of the dedicated outputs of this unit and bit assignment is as shown below.</p> <table border="1"> <thead> <tr> <th>Bit</th> <th>bit2</th> <th>bit1</th> <th>bit0</th> </tr> </thead> <tbody> <tr> <td>Signal name</td> <td>Error</td> <td>In motion</td> <td>Ready</td> </tr> </tbody> </table> <p>When each signal is ON, the corresponding bit is 1.  This register is rewritable, however, execution of programs may rewrite them.  The range of values this register can handle is 0 to 7.</p> <p>This is used for the arithmetic calculation with other registers and numerical values and also with the conditional branching.  When only REG_SPO is used singly, the value of this register is sent through RS-232C interface.  The form of transmission uses REG_SPO=d. The d is the value of the register.</p> <p>Related command &lt;WAIT, IF, REG_??&gt;</p>	Bit	bit2	bit1	bit0	Signal name	Error	In motion	Ready
Bit	bit2	bit1	bit0						
Signal name	Error	In motion	Ready						
Example	<p>REG_R00=REG_SPO (1) The status of the dedicated output is assigned to the general purpose register, REG_R00.  WAIT(REG_SPI&amp;2) (2) This unit waits while the in-motion signal is ON.</p>								

#### 6-47. REG\_ERR

Form	REG_ERR
Function	This register indicates the content of an error when it is generated.
Description	<p>When an error is generated in this unit, this register stores the contents of the error (Error code).</p> <p>This register is cleared to 0 when the power is turned ON.</p> <p>This register is dedicated to reading and no values can be assigned. However, it is possible to write 0 into it when an error is generated. When a 0 is written into it, the error output is also reset.</p> <p>It is also possible to clear the register to 0 using ALL_RST command. Refer to the Error Codes for the contents of errors.</p> <p>Related command &lt;WAIT, IF, REG_???, All_RST&gt;</p>
Example	<p>PRG_STR=31 (1) Program 31 starts. (No. 31 is the program used for Error Sequence.)</p> <p>REG_ERR=0 (2) The Error Register is cleared to 0. (The Error Output is also reset.)</p> <p>REG_O02=1 (3) The general purpose output terminal, 02, is turned ON</p> <p>PRG_END=31 (4) Program ends.</p> <p>The Program No. 31 is executed automatically when an error is generated. In the above example, the error was reset and the general purpose output terminal, 02, is turned ON.</p>

#### 6-48. REG\_ENC

Form	REG_ENC
Function	This register indicates the value of encoder counter.
Description	<p>This register indicates the value of encoder counter. This register is cleared to 0 when the power is turned ON.</p> <p>The range of values this register can handle is 2147483648 to +2147483647.</p> <p>This is used for the arithmetic calculation with other registers and numerical values and also with the conditional branching.</p> <p>When only REG_ENC is used singly, the value of this register is sent through RS-232C interface.</p> <p>The form of transmission uses REG_ENC=d. The d is the value of the register.</p> <p>Related command &lt;WAIT, IF, REG_???,&gt;</p>
Example	<p>PRG_STR=0 (1) Program 0 starts.</p> <p>REG_ENC=0 (2) The Encoder counter is cleared to 0.</p> <p>MOV_SCN=+E (3) Scanning in the forward direction is started.</p> <p>WAIT(REG_ENC&lt;1000) (4) This unit waits until the encoder counter becomes 1000.</p> <p>STP_SDW (5) Decelerated and stop.</p> <p>PRG_END=0 (6) Program 0 ends.</p>

#### 6-49. REG\_SWT

Form	REG_SWT
Function	This register indicates the status of switch of axis.
Description	<p>This register is used to know the status of axis setting of this unit.</p> <p>The range of values this register can handle is 0 to 15.</p> <p>This is used for the arithmetic calculation with other registers and numerical values and also with the conditional branching.</p> <p>When only REG_SWT is used singly, the value of this register is sent through RS-232C interface.</p> <p>The form of transmission uses REG_SWT=d. The d is the value of the register.</p> <p>Related command &lt;WAIT, IF, REG_???,&gt;</p>

Example	REG_R00=REG_SWT (1) The status of switch of axis is assigned to the general purpose register, REG_R00.
---------	--

## Home search sequence

This unit uses the following four types of Home Search sequences. They should be selected according to the direction of search and the number of sensors.

Home search sequence	Direction of search	Number of sensors used
Home search sequence 0	Reverse	Near home
Home search sequence 1	Reverse	Near home, Home
Home search sequence 2	Forward	Near home
Home search sequence 3	Forward	Near home, Home

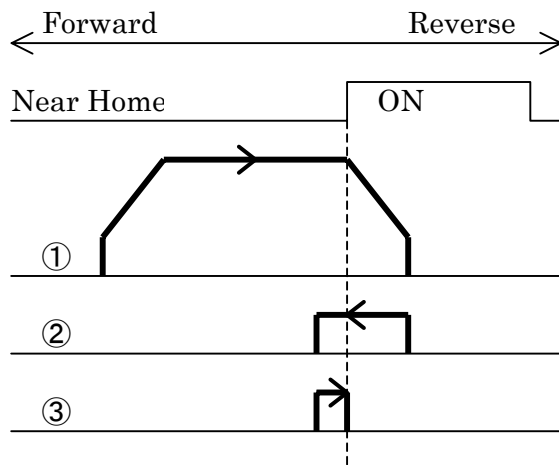
Table 7-1. List of Home search sequence

### 7-1. Home search sequence 0

The search is in reverse direction and only the Near Home is used.

- (1) The movement is in reverse. When the Near Home goes ON, the motor decelerates and stops.
- (2) The motor moves forward at the starting speed and stops after it reaches the forward side of the Near Home.
- (3) The motor moves to the reverse side and stops when the Near Home goes ON.

In case that the reverse overrun issues at (1), the motor stops instantly and goes to (2).

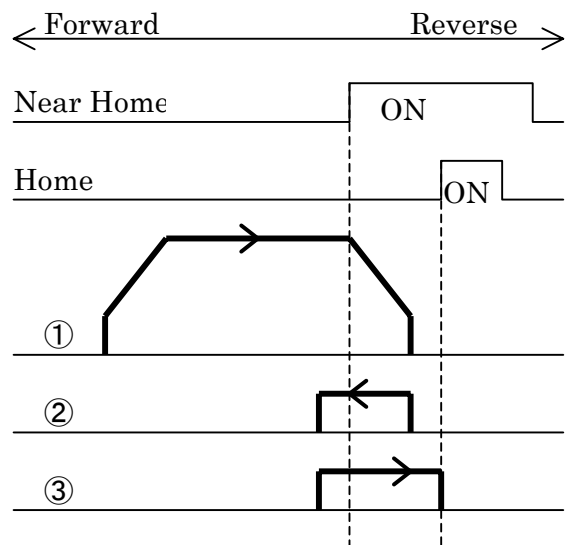


### 7-2. Home search sequence 1.

The direction of search is in reverse direction and Near Home and Home are used.

- (1) The movement is in reverse. When the Near Home goes ON, the motor decelerates and stops.
- (2) The motor moves forward with starting speed and stops after it reaches the forward side of the Near Home.
- (3) The motor moves to the reverse side with start speed and stops when the Home goes ON after the Near Home goes ON.

In case that the reverse overrun issues at (1), the motor stops instantly and goes to (2).

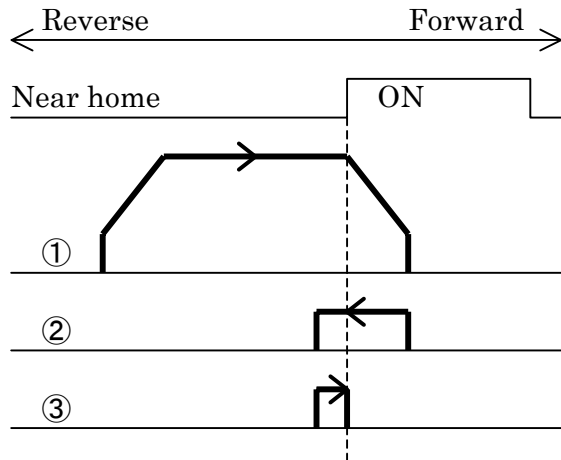


### 7-3. Home search sequence 2.

The search is in forward direction and only the Near Home is used.

- (1) The movement is in forward. When the Near Home goes ON, the motor decelerates and stops.
- (2) The motor moves reverse at the starting speed and stops after it reaches the reverse side of the Near Home.
- (3) The motor moves to the forward side and stops when the Near Home goes ON.

In case that the forward overrun issues at (1), the motor stops instantly and goes to (2).

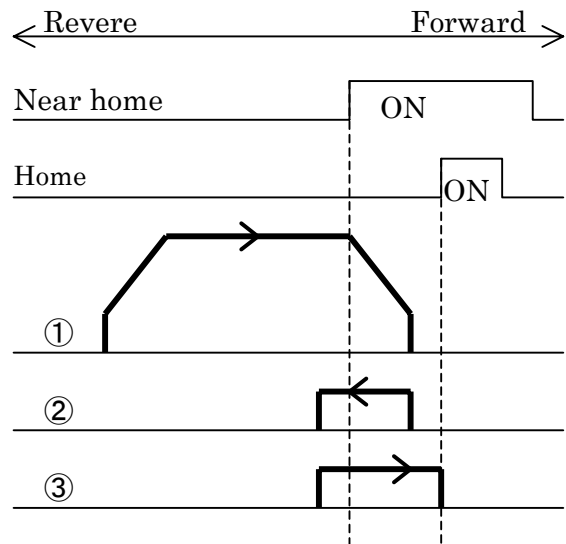


### 7-4. Home search sequence 3

The direction of search is in forward direction and Near Home and Home are used.

- (1) The movement is in forward. When the Near Home goes ON, the motor decelerates and stops.
- (2) The motor moves reverse with starting speed and stops after it reaches the reverse side of the Near Home.
- (3) The motor moves to the forward side with start speed and stops when the Home goes ON after the Near Home goes ON.

In case that the forward overrun issues at (1), the motor stops instantly and goes to (2).



## 8. On-line Operation

The Online Operation Mode is executed by receiving commands from the host through RS-232C. Program functions shown in the list of commands of chapter 5 can be used.

The following communication parameters are used.

Communication speed	9600bps
Data bits	8 bits
Stop bit	1 bit
Parity check	None

When a command is to be transmitted to this unit, first, one byte which specifies the axis is transmitted.

Next, the command itself is transmitted. Lastly, CR (ODH in hex) is transmitted.

The one byte code that specifies the axis uses the character code of the axis names, A to P, in addition to 80H.

<Example> When a command, MOV\_PTP=1000 is sent to the C-axis, the axis specifying code for the C-axis will be 'C' + 80H = 43H + 80H = C3H. Therefore, C3H is transmitted as the axis specifying code. Then, MOV\_PTP=1000 is transmitted. Lastly, CR (ODH in hex) is sent.

When this unit receives a command by online, this unit analyzes the command. If nothing is wrong, it sends NEXT to the host and executes the command received.

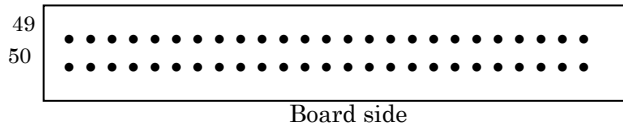
If this unit receives only the register name (starting with REG\_) as a command, this unit sends the received register name and its value to the host instead of sending NEXT.

<Example> When this unit receives a command, REG\_POS, REG\_POS is a register which indicates the current position. If the current position is 1000, this unit sends REG\_POS=1000 to the host.

When this unit sends, the axis specifying code is not sent. However, the last CR is added and sent.

# 9. Connector

## 9-1. I/O Connector



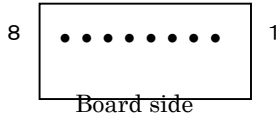
1 Connector: XG4A-5034(OMRON) or equivalent  
 Socket: XG4M-5030(OMRON) or equiv. (Accessory)  
 2 Cover: XG4T-5004(OMRON) or equiv. (Accessory)

Type	No.	Signal name	Contents	Circuit of interface and remarks
Output system	1	O00	General purpose output 00	
	2	O01	General purpose output 01	
	3	O02	General purpose output 02	
	4	O03	General purpose output 03	
	5	O04	General purpose output 04	
	6	O05	General purpose output 05	
	7	O06	General purpose output 06	
	8	O07	General purpose output 04	
	9	RDY	READY output	
	10	MOVE	MOVING output	
	11	ERR	ERROR output	
Input system	12	I00	General purpose input 00	
	13	I01	General purpose input 01	
	14	I02	General purpose input 02	
	15	I03	General purpose input 03	
	16	I04	General purpose input 04	
	17	I05	General purpose input 05	
	18	I06	General purpose input 06	
	19	I07	General purpose input 07	
	20	P_ST	Program start	
	21	C_ST	Cam start	
	22	EMER	Emergency stop	
Sensor system	23	FOR	Forward overrun	A power supply capable of 24VDC at 0.2A or more is needed. The voltage range is 22.8V to 25.2VDC
	24	REV	Reverse overrun	
	25	NEAR	Near Home	
	26	HOME	Home	
Power system	27,28	24V	24V input	The interface circuit is the same as Pin 23 to Pin 26.
	29,30	0V	0V input	
Driver system	31	INPOS	In-position	
	32	ALM	Alarm	
	33	+SON	Servo on output +	
	34	-SON	Servo on output -	
	35	+CR	Counter reset +	
	36	-CR	Counter reset-	
	37	+CW	CW Clock output +	
	38	-CW	CW Clock output -	
Encoder	39	+CCW	CCW Clock output +	
	40	-CCW	CCW Clock output -	
	41	+RSCW	CW Clock output +	
	42	-RSCW	CW Clock output -	
Power system	43	+RSCCW	CCW Clock output +	
	44	-RSCCW	CCW Clock output -	
	45	+A	A phase input +	
	46	-A	A phase input -	
Encoder	47	+B	B phase input +	
	48	-B	B phase input -	
	49	5V	5V output	
	50	0V	0V output	

- (1) General Purpose Outputs, 0 to 7, (O00~O07), Pin numbers: 1 to 8  
These are terminals used for general purpose output in program.  
These can be used for driving small relays under 35V and up to 0.5A.
- (2) Ready output (RDY), Pin number: 9  
This is turned ON when the program start signal is acceptable.
- (3) In-moving output (MOVE), Pin number 10  
This signal goes ON when a pulse is being outputted.
- (4) Error output (ERR), Pin number 11  
This signal is turned ON when an error is generated within this unit.  
The Program No. 31 is executed after this signal is turned ON.
- (5) General Purpose Inputs, 0~7, (I00 to I07) Pin numbers: 12 to 19.  
These are terminals used for general purpose input in program.  
These are, for example, used for branching programs.
- (6) Program start (P\_ST), Pin number: 20  
When this signal goes ON, the program No. 0 is executed.  
When the Ready Output signal is OFF, the Program Start signal is not accepted.
- (7) Cam start input (C\_ST), Pin number 21  
When this signal goes ON, the CAM movement starts.  
It is necessary to execute the program and specify the number of the CAM shape to be used in advance.
- (8) Emergency stop (EMER), Pin number 22  
If this signal goes ON while pulses are being outputted, the machine stops immediately and an error signal is outputted.
- (9) Forward overrun (FOR), Pin number 23  
Forward side overrun sensor is connected to this pin.  
If this signal is inputted when movement in forward direction is going on, an error signal is outputted.  
The input logic of this signal can be changed.
- (10) Reverse overrun (REV), Pin number 24  
Reverse side overrun sensor is connected to this pin.  
If this signal is inputted when movement in reverse direction is going on, an error signal is outputted.  
The input logic of this signal can be changed.
- (11) Near home (NEAR), Pin number 25  
Near home sensor is connected to this pin.  
The Home Search operation uses this sensor.  
The input logic of this signal can be changed.
- (12) Home (HOME), Pin number 26  
Home sensor is connected to this pin.  
This sensor is decided to be used or not depending on home search sequence.  
The input logic of this signal can be changed.

- (13) 24V input (24V), Pin numbers: 27, 28
- (14) 0V input (0V), Pin number: 29, 30  
Please supply 24VDC at 0.2A or more externally.  
The voltage range is 22.8V~25.2VDC.
- (15) In-position (INP), Pin number 31  
This connects to the In-position signal of the Servo Driver (Operation Complete signal).  
This is effective when the motor setting is servo motor.  
This unit does not advance to the next step until this signal goes ON after the pulse is outputted.  
The input logic of this signal can be changed.
- (16) Alarm (ALM), Pin number 32  
Alarm signal of driver is connected to this pin.  
If this signal is inputted in the middle of movement, an error is generated.  
The input logic of this signal can be changed.
- (17) Servo-on output (+SON, -SON), Pin numbers: 33, 34  
This signal is used with the servo motor. If the operation becomes possible after the power is turned ON, this is turned ON.  
This signal can be turned ON or OFF by program.
- (18) Counter clear output (+CR, -CR), Pin numbers: 35, 36  
This is the signal for servo motor and one-shot output.  
When the Home Search is completed, this is outputted automatically.  
It is possible to use this one-shot output by program.
- (19) CW clock output (+CW, -CW), Pin numbers: 37, 38
- (20) CCW clock output (+ CCW, -CCW), Pin numbers: 39, 40
- (21) CW clock output (+RSCW, -RSCW), Pin numbers: 41, 42
- (22) CCW clock output (+ RSCCW, -RSCCW), Pin numbers: 43, 44  
Clocks are outputted to drivers.  
Among the output circuits, Pins 37 to 40 are open collector output; pins 41 to 44 are RS-422 line driver output.  
When one clock is used, the CW terminal is the clock output and the CCW terminal is direction output.
- (23) A phase input (+A, -A), Pin numbers: 45, 46
- (24) B phase input (+B, -B), Pin numbers: 47, 48  
These are Encoder signal inputs.  
The input circuits are RS-422 line receivers.
- (25) 5V output (5V), Pin number: 49
- (26) 0V output (0V), Pin number: 50  
The 5V power is generated from the 24V of pins 27 to 30 and supplied externally.  
This can be used as the power for the encoder system. Use within 0.2A.  
The range of the output voltage is 4.75V to 5.25VDC.

### 9-2. RS-232C Connector

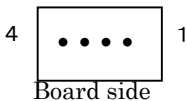


Post with base : BS8P-SHF-1AA (JST) Equivalent  
 Housing : H8P-SHF-AA (JST) Equivalent (Accessory)  
 Contact : SHF-001T-0.8 (JST) Equivalent (Accessory)

Pin #	Signal name	Contents	Circuit of interface and remarks
1	RXD	Receive data	
2	RET	Return	
3	GND	Ground	
4	TXD	Send data	
5	+DTXD	TXD+ for multiple axes	
6	-DTXD	TXD- for multiple axes	
7	+DRXD	RXD+ for multiple axes	
8	-DRXD	RXD- for multiple axes	

- (1) Receive data (RXD), Pin number: 1  
This terminal receives serial data.
- (2) Return (RET), Pin number: 2  
This terminal is used to check the state of connector connection.  
When RS-232C is used, make sure that this is connected to the GROUND of (3).
- (3) Ground (GND), Pin number 3  
This terminal is the signal GROUND.
- (4) Send data (TXD), Pin number 4  
This terminal sends serial data.
- (5) TXD± for Multiple axes (+DTXD, -DTXD), Pin numbers: 7, 8
- (6) RXD± for Multiple axes (+DRXD, -DRXD), Pin number 9, 10  
This signal is used when RS-232C is commonly used for multiple axes.  
Please refer the wiring example.

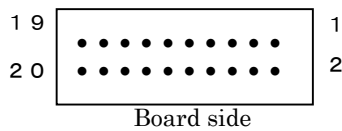
### 9-3. DC5V Connector



Post with base : BS4P-SHF-1AA (JST) or equivalent  
 Housing : H4P-SHF-AA (JST) or equivalent (Accessory)  
 Contact : SHF-001T-0.8 (JST) or equivalent

Pin #	Signal name	Contents	Description
1	FG.	Frame ground	Grounded to frame.
2	NC	No connection	Not used.
3	GND	0V input	5VDC of 0.5A or more is supplied from outside. The voltage range is 4.75V to 5.25VDC.
4	DC5V	5V input	

### 9-4. I/O2 Connector



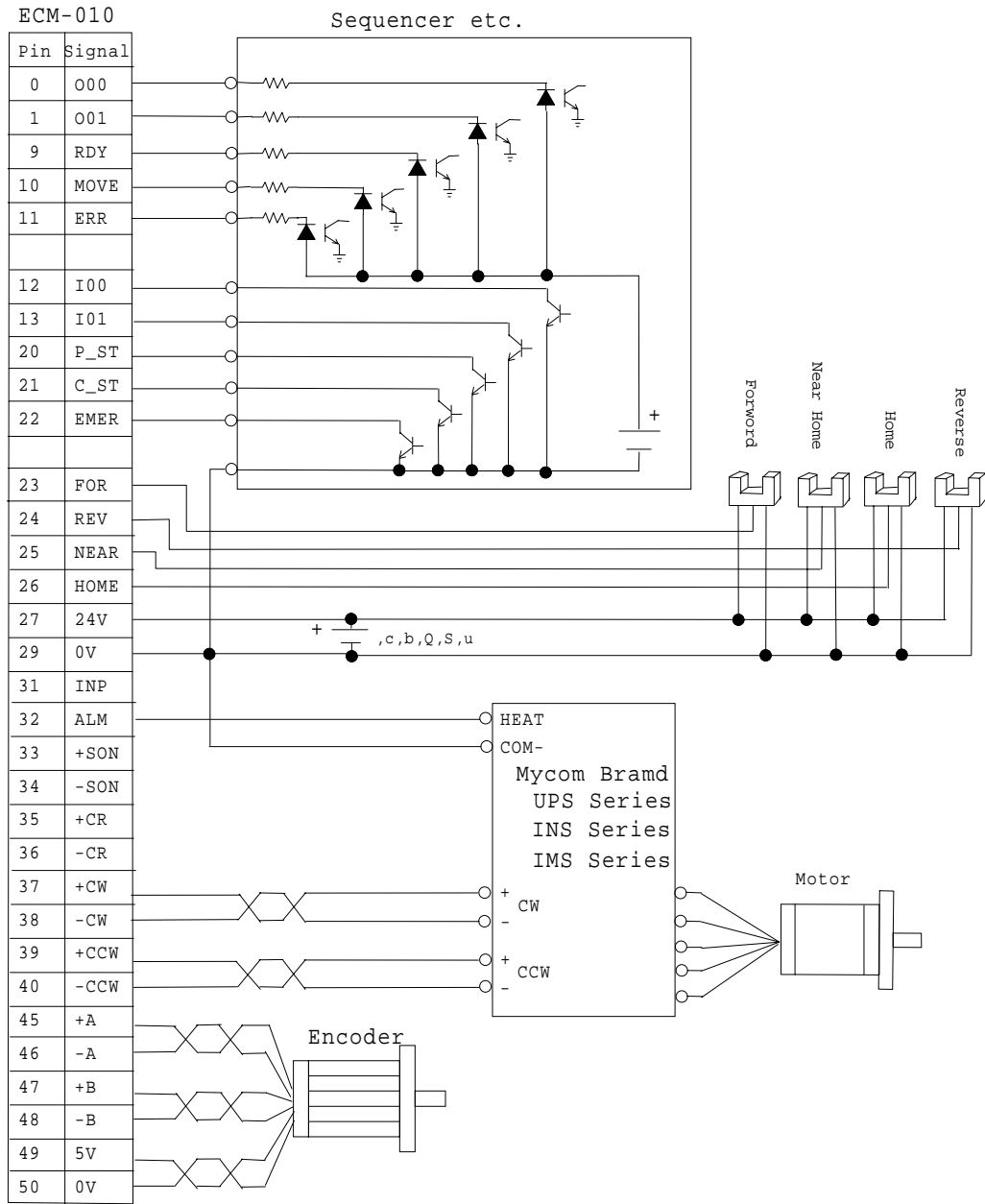
Connector : XG4A-2034(OMRON) or equivalent  
 Socket : XG4M-2030(OMRON) or equivalent (Accessory)  
 Cover : XG4T-2004(OMRON) or equivalent (Accessory)

Type	Number	Signal name	Contents	Circuit of interface and remarks
Output series	1	O08	General purpose output 08	
	2	O09	General purpose output 09	
	3	O10	General purpose output 10	
	4	O11	General purpose output 11	
	5	O12	General purpose output 12	
	6	O13	General purpose output 13	
	7	O14	General purpose output 14	
	8	O15	General purpose output 15	
	9	0V	0V	
	10	0V	0V	
Input series	11	I08	General purpose input 08	
	12	I09	General purpose input 09	
	13	I10	General purpose input 10	
	14	I11	General purpose input 11	
	15	I12	General purpose input 12	
	16	I13	General purpose input 13	
	17	I14	General purpose input 14	
	18	I15	General purpose input 15	
	19	0V	0V	
	20	0V	0V	

- (1) General purpose output 08 to 15 (O08 to O15), Pin numbers : 1 to 8  
 These are terminals used for general purpose output in program.  
 Please use under 24V and up to 20mA
  
- (1) General purpose input 08 to 15 (I08 to i15), Pin numbers : 11 to 18  
 These are terminals used for general purpose input in program.  
 These are, for example, used for branching programs.

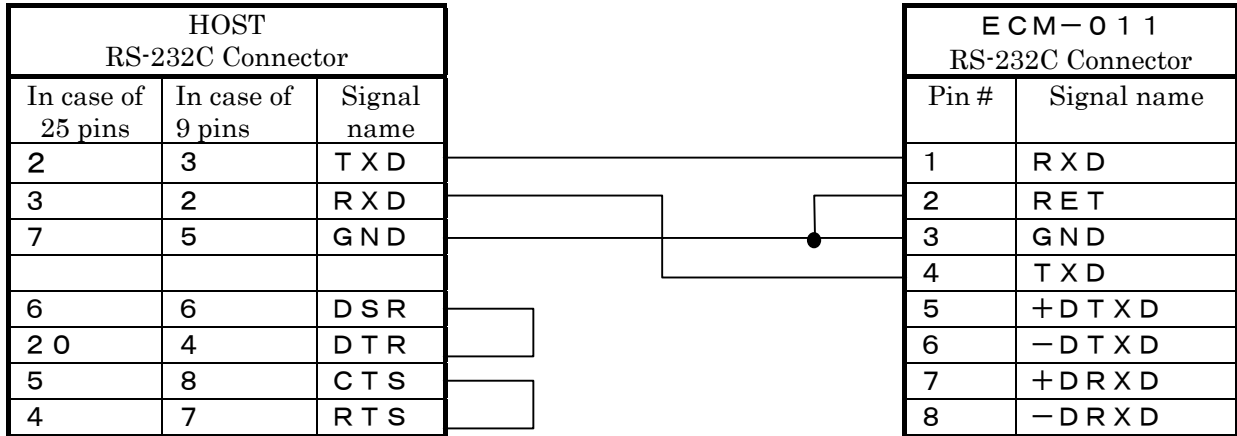
# 10. Example of wiring

## 10-1. Connecting example of I/O1 connector

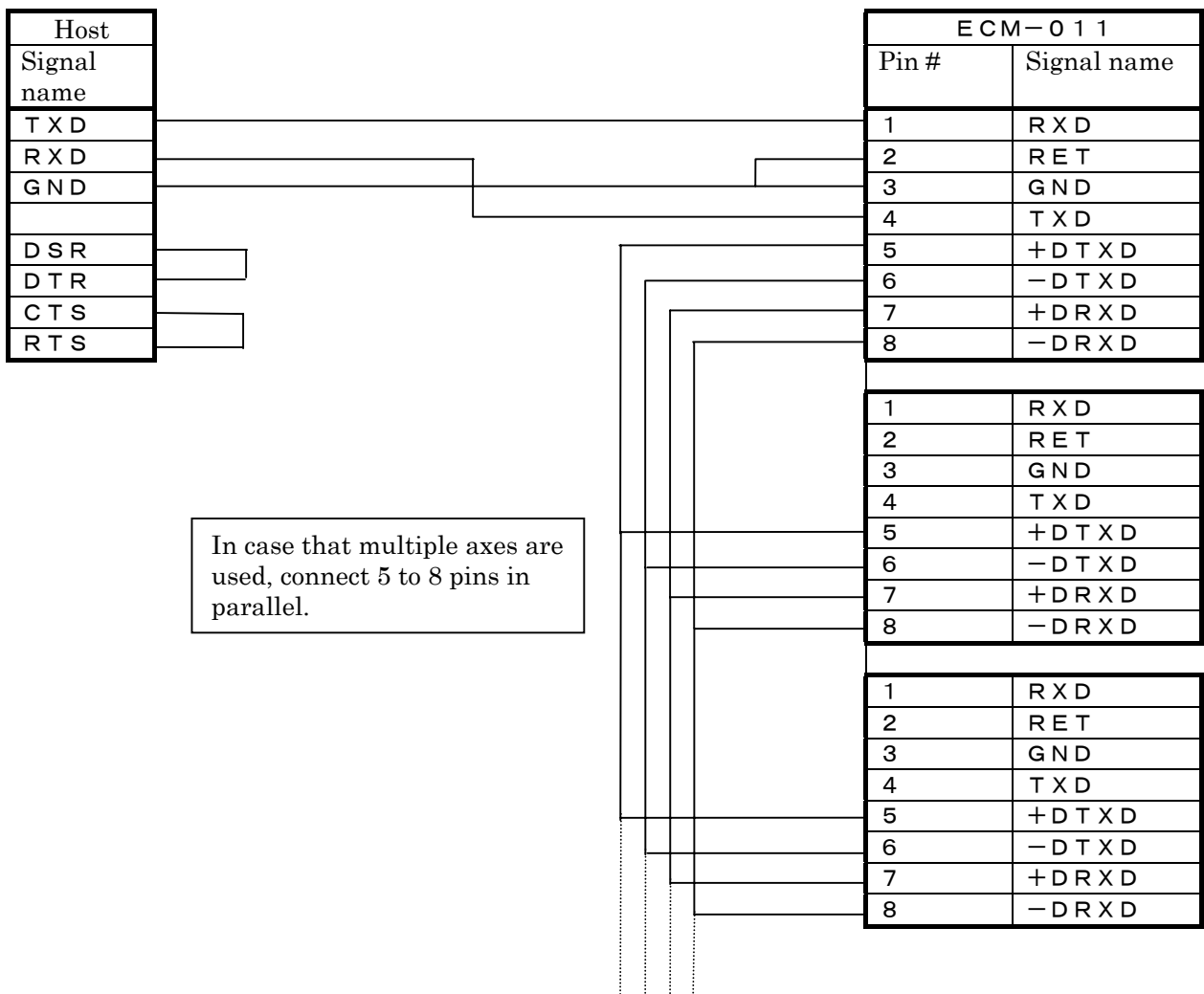


## 10-2. Example of RS-232C connecting

In case of one axis connection

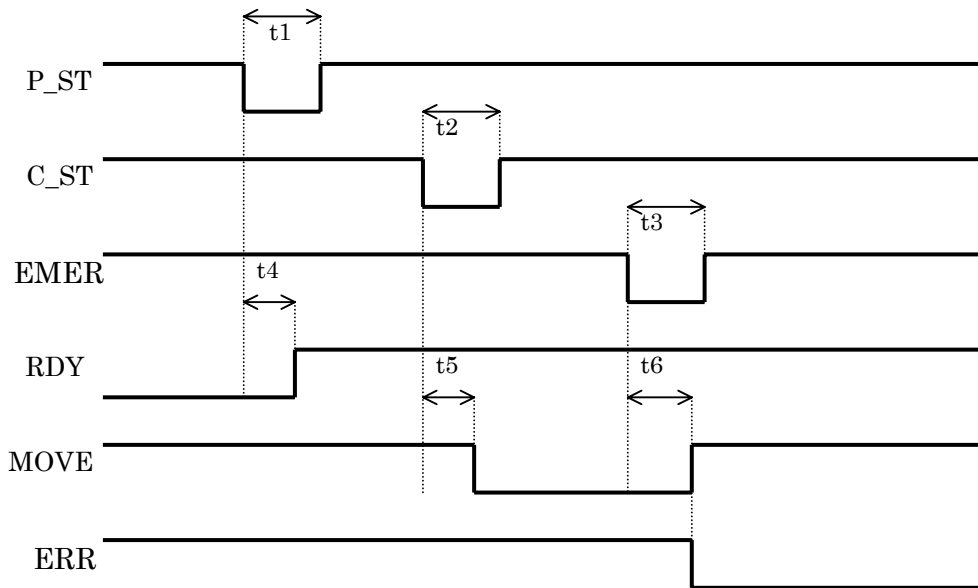


In case that multiple axes are connected:



## 11. Timing

### 11-1. External signal timing



Signal	Description	Time
t1	Program start keeping time	2ms. or more
t2	Cam start keeping time	2ms. or more
t3	Emergency stop keeping time	2ms. or more
t4	From program start to Ready output off	100us. or less
t5	From cam start to in-moving output off.	50us. or less
t6	From emergency stop to error output on.	1ms. or less

### 11-2. Command execution time

Command	Time[ $\mu$ s]
PRG_STR,PRG_END,PRG_TSK,PRG_JMP,PRG_LIN,PRG_CAL, WAIT(),IF()	100
SET_TBL,SET_ENC,CPX_SP1/SP2/SP3,CPX_PS1/PS2/PS3	100
SET_MAX,SET_MIN,	500
SET_ACC,SET_DEC,	750
STP_SDW,STP_EMR,STP_CAM,STP_PAS	150
SRV_ON,SRV_OFF	100
CLR_ON	50000
ALL_RST	4000
MSK SIN	150
Register substitution (example: REG_POS=100)	100
Register calculation (example: REG_POS=REG_R00+200)	120
Register sending (example: REG_POS)	11000

## 12. Error code

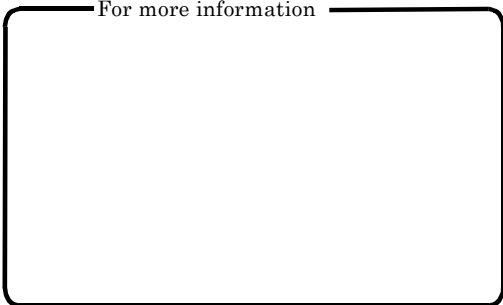
When an error is generated, the red LED on the panel lights and the error output goes ON. The error codes shown below go into the REG\_ERR register.

Error code	Contents of error
101	This is the command form error. For example, this is generated when wrong command is received through on-line.
102	This is unexcitable error. For example, duplicate operation, wrong register value and so on.
201	Emergency stop error generated externally.
202	Driver alarm error
203	Forward overrun error.
204	Reverse overrun error.
205	Home search error

Table 12-1. Table of error codes

Please understand that we may make modifications to our products without notification in order to improve the capabilities and external appearance of our products.

For more information



## **MYCOM, INC.**

Head office 12, S. Shimobano, Saga hirosawa, Ukyo, Kyoto,  
Japan 616-8303

TEL: 075(882)3601

FAX 075(882)6531

HomePage <http://www.mycom-japan.co.jp/>

Sales office Kyoto, Tokyo, Oita

Factory Kyoto

Subsidiary Taiwan, Korea, U.S.A., Singapore, Malaysia