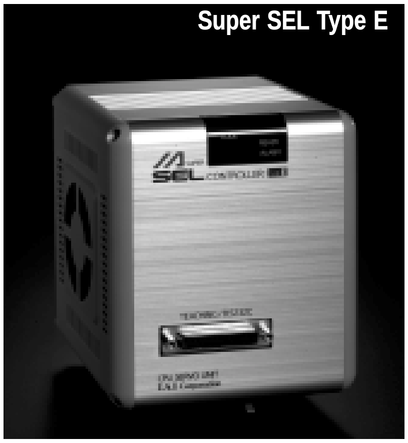
INTELLIGENT ACTUATOR

Super SEL Controller (Type E & G)

Operating Manual


Super SEL Type E

Intelligent Actuator Inc.

This publication was written to assist you in better understanding this part of your IA system. If you require further assistance, please contact IA Technical Support. For Central and East Coast Time Zones, please call our Itasca, IL office at 1-800-944-0333 or FAX 630-467-9912. For Mountain and Pacific Time Zones, please call our Torrance, CA office at 1-800-736-1712 or FAX 310-891-0815; Monday thru Friday from 8:00 AM to 5:00PM.

Intelligent Actuator, Inc.
U.S. Headquarters
2690 W. 237th Street
Torrance, CA 90501
310-891-6015 / 310-891-0815 fax

Intelligent Actuator, Inc.
Midwest Regional Office
1261 Hamilton Parkway
Itasca, IL 60143
630-467-9900 / 630-467-9912 fax

www.intelligentactuator.com

Disclaimer
The information and technical data contained herein are subject to change without notice. Intelligent Actuator, Inc. assumes no responsibility for any errors or omissions regarding the accuracy of the information contained in this publication.

Foreword

Thank you very much for purchasing the IA Super SEL Controller E·G Type. Without knowing beforehand how to correctly use or operate the controller, not only will the user be unable to take full advantage of all the functions built into this product but he might inadvertently cause damage to the controller or shorten its life. Please read this manual carefully to acquire an understanding of the proper method of handling and operating the controller. Keep the manual handy so that you can refer to the appropriate sections as the need arises.

Your Super SEL Controller E·G Type has a built-in 32 bit RISC (Reduced Instruction Set Computer) CPU and uses SEL language developed exclusively by IAI for programming. The Super SEL is a highly advanced, new generation of controller that eliminates the need for a PLC, allows you to do multi-tasking (parallel processing), and allows you to network the controllers with the SEL NET option.

As with our other Super SEL controllers, the Super SEL E·G Type can be used with all IAI actuators. Type E is designed for single axis control and Type G for multiple axes control. Please use the special cable provided to connect the actuator with the controller.

*All precautions have been taken to ensure the accuracy of the contents of this manual. However, if you become aware of any inaccuracies or discrepancies, please contact your IAI sales representative or technical service department.

Table of Contents

Table of Contents

Before You Begin

Differences between the Super SEL Controller Type E/G and Type A/B

(Be sure to read this before you begin)

1. There is no built-in DC24V power supply for the I/O in the Super SEL Controller Type E/G. The DC24V power must be supplied externally. Connect +24V to I/O connector Pin 1A, and 0V to Pin 25B. (Refer to "Supplement 5. I/O DC24V power Supply" for more details.)

2. For the Super SEL Controller Type E·G, the Emergency Stop is normally closed. To release the Emergency Stop, Pin 2B and 0V must be short-circuited. To release the Emergency Stop for testing, short-circuit the jumper post (ST1) placed at the bottom of the controller CPU UNIT or CPU SERVO UNIT with a jumper pin. Even in this case, the Emergency Stop function from the teaching pendant is still effective. (Refer to Supplement 6. Emergency Stop)
   Note: After testing, please make sure to take the jumper pin out so that the Emergency Stop operates externally.

3. The Super SEL Type E·G Controllers are designed to be mounted inside of a control panel. For cooling, the heat sink method or forced air current method is recommended. (Refer to Supplement 4. Heat Dissipation)

A Word of Caution

① Please read this manual carefully to operate the controller properly.
② You are not allowed to use or reproduce this manual or any portion thereof without permission.
③ We cannot accept any responsibility for possible damage resulting from the use of this manual.
④ We reserve the right to change the information contained in this manual without prior notice.

Emergency Procedures

If hazardous conditions arise while using the controller, immediately turn OFF all power switches for the controller and any devices connected to it, or pull all the power plugs from the electric outlet. ("Hazardous condition" refers to excessive heat, smoke or flames coming from the controller or any conditions which might lead to fire or cause damage to the controller.)

Chapter 1. Setting Up

Part 1 Safety Precautions

The IA Super SEL Controller Type E was designed to control any type single axis IAI actuator and Type G was designed to control any type actuator in assembly configurations using a maximum of 8 axes or integrated with other peripheral devices. It is capable of controlling everything from a simple single axis system to large scale FA (factory automation) system. As systems become more complicated, the possibility of incorrect operation or accidents arising from carelessness also increases. Please take sufficient care when operating your system.

Please follow the following safety precautions when operating your IA system:

(1) Any operation not specifically addressed in this manual should not be attempted. If you have any questions, please contact your IA sales representative or contact IA technical support at: 1-800-736-1712.

(2) Use only IA cables when connecting IA actuators and controllers. IA cables are matched for use with IA actuators and are specially designed to withstand repeated bending.

(3) Stand clear of your IA system when operating or preparing to operate. Surround your IA system with safety partitions if there is any possibility that someone may become injured by an operating IA system.

(4) Before assembling, adjusting, or performing maintenance on your IA system, please make sure that people around you are aware that the system is not to be powered up or turned on. You may want to disconnect the power cable completely, keep the power cable close to the operator, or use a safety plug to ensure that the power cable will not be plugged in inadvertently.

(5) When more than one person is working on your IA system, use signs to inform everyone of the operating status of the equipment. Make sure that everyone stands clear prior to operation. Operate your system only after you are sure that everyone knows that you are initiating system start-up and that everyone is clear of the system.

(6) In situations where the cables must be lengthened, be sure to double check all connections before powering up your IA system.

Part 2 Warranty Period and Scope

The Super SEL controller undergoes stringent testing before it is shipped from our factory. IAI provides the following warranty.

1. Warranty Period

The warranty period is 12 months from the date the unit is shipped to the customer.

2. Scope of Warranty

If within the period specified above, a breakdown occurs while operating the controller under normal conditions and is clearly the responsibility of the manufacturer, IAI will repair the unit at no cost. However, the following items are not covered by this warranty.

- Faded paint or other changes that occur naturally over time.
- Consumable components that wear out with use (battery, etc.).
- Unit seems to be noisy or similar impressions that do not affect machinery performance.
- Damage resulting from improper handling or use.
- Damage resulting from user error or failure to perform proper maintenance.
- Any alterations not authorized by IAI or its representatives.
- Damage caused by fire and other natural disasters or accidents.

The warranty pertains to the purchased product itself and does not cover any loss that might arise from a breakdown of the product. Any repairs will be done at our factory.

3. Service

The purchase price of the product does not include programming or expenses for sending technicians to the customer's site. Even if the product is still under the warranty period, separate charges will be assessed for the following services.

- Assistance with unit installation or trial operation.
- Inspection and maintenance.
- Technical instruction and training for controller operation and wiring.
- Writing programs or technical instruction and training for programming.
- Any other services or work for which IAI normally assesses separate charges.

Part 3   Installation Environment and Noise Measures

1. Installation Environment

(1) Do **NOT** block the air vents of your controller when installing your IA system.

(2) Your Super SEL Controller is NOT dust, water, or oil proof. Take steps to prevent foreign matter from getting into the controller air vents. Avoid using your IA system in environments subject to contamination by dust, oil mist, or cutting oil.

(3) Do not expose your IA system to direct sunlight or place it near a heat source.

(4) Avoid placing your IA system under conditions of extreme tempreratures above 50°C (120°F) or below 0°C (32°F). The area level of humidity should not be above 85%. Do NOT expose to corrosive or inflammable gas.

(5) Avoid external vibration, unnecessary impact, or excessive shocks to your IA system.

(6) Take steps to shield all cables and wires from electromagnetic noise.

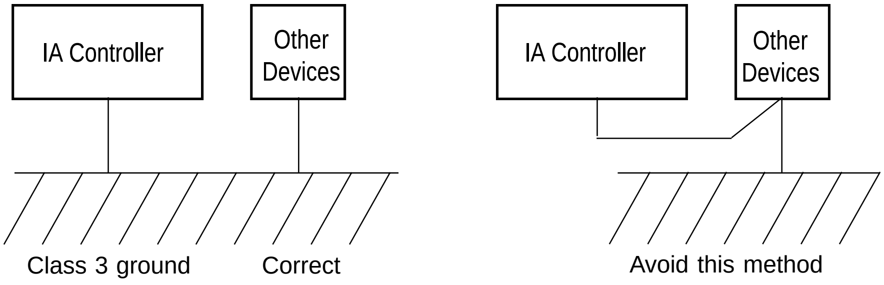
2. Power Source

Make certain that an AC line voltage of 90 ~ 127V (Rated 100 ~ 120V) is maintained. If the power supply tends to fluctuate substantially, use a constant-voltage transformer.

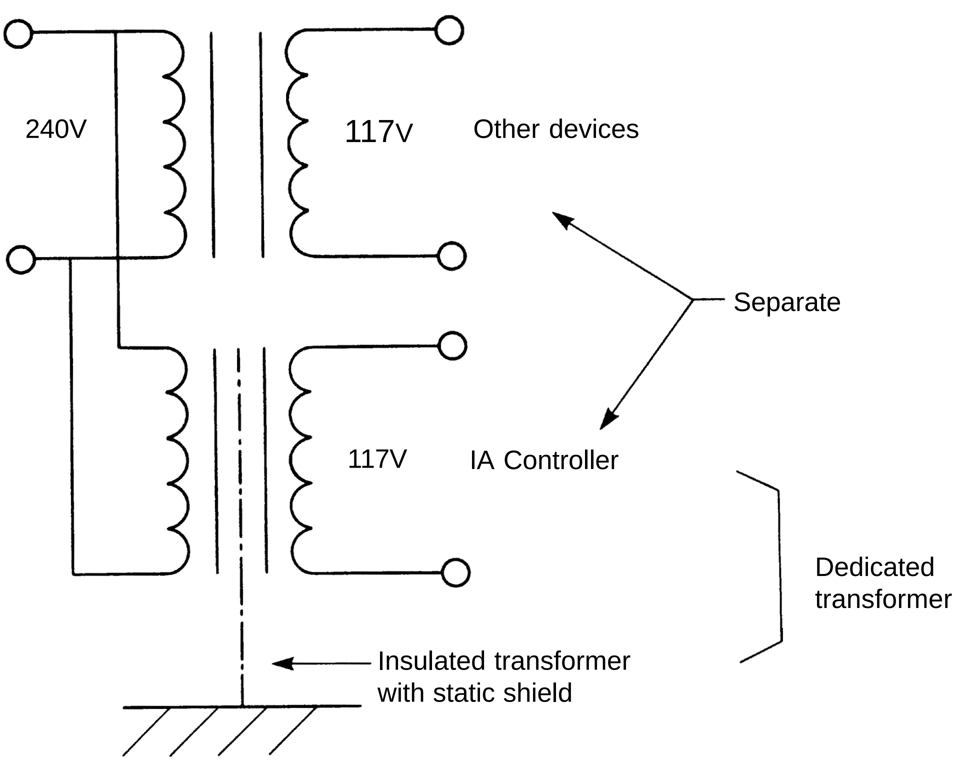
3. Electromagnetic Noise Supression

(1) Wiring and Power Supply

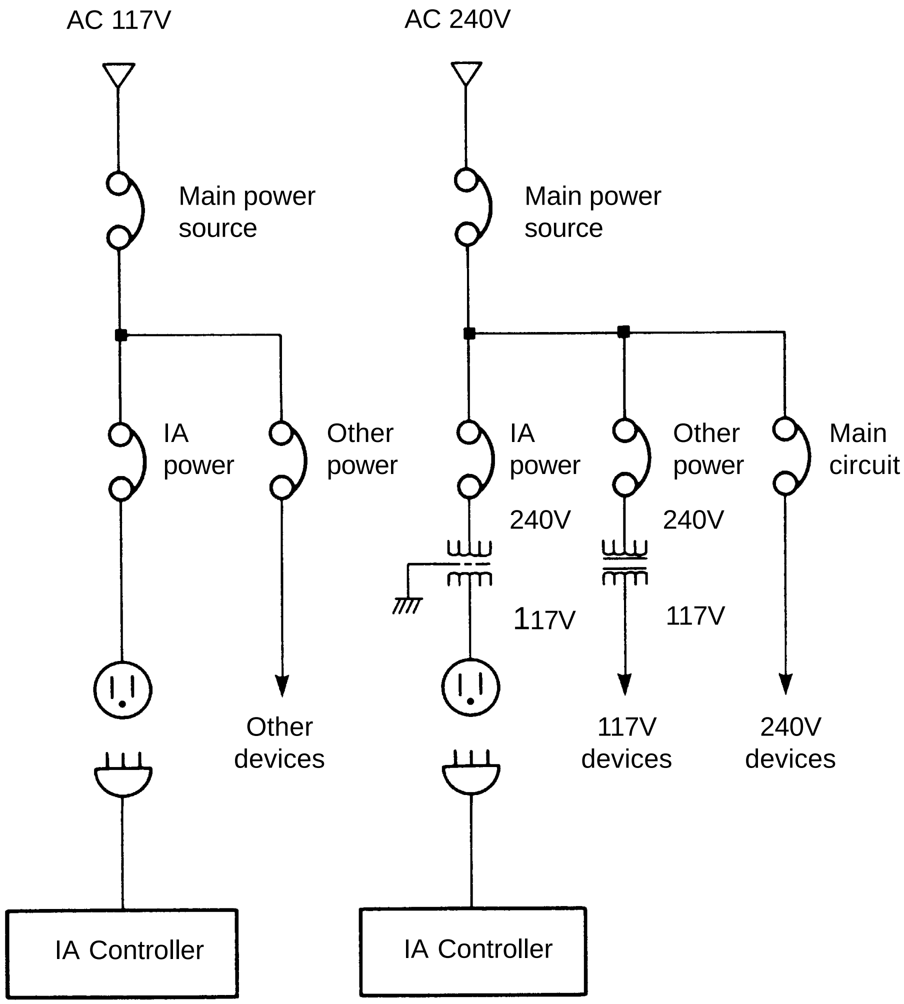
① For grounding, please use a dedicated ground of Class 3 or better. The thickness of the cable should be 2.0~5.5mm^2 or larger.



② If you need to use a stepdown transformer to lower the power voltage from 240V to 117V, use a dedicated, insulated transformer for the IA controller. (For further details, please contact your IA sales representative or technical support).



③ Please use a dedicated and insulated power transformer when the system has a 240V power source.



④ Wiring Notes

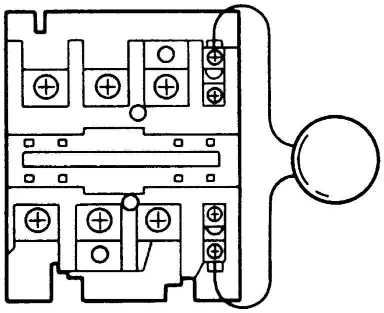
1. To reduce noise problems, the AC117V and the DC24V external power cable should be a twisted pair.
2. Isolate the SEL cables from the power line.
3. For DC motors, isolate the encoder cable from the motor cable.
4. For AC motors, the motor and encoder cables are partially wired together but when the length of the cable extends more than 5m, please wire them separately.
5. Consult with IAI if you need to extend the motor and encoder cables beyond the length that comes with the controller.

(2) Noise Source and Noise Suppression

When using electrical components such as electromagnets, solenoids, or relays which create electromagnetic noise, some type of noise supression device should be used.

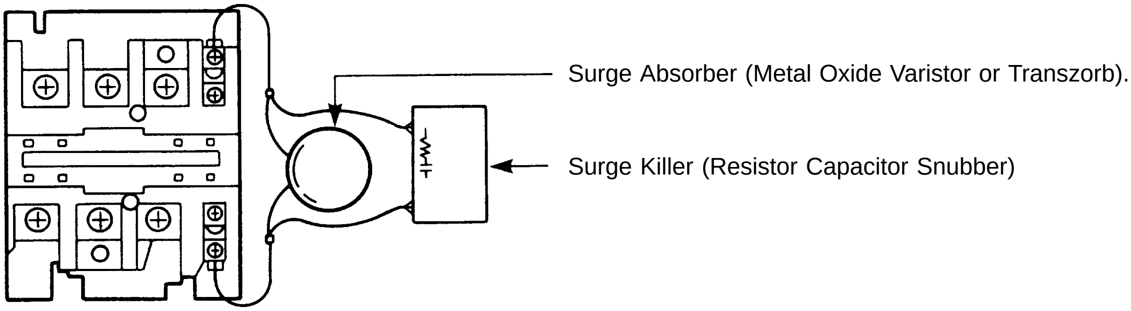
① AC solenoid valve · magnetic switch · relay

- Install a surge absorber parallel to the reactance load (solenoid and relay coils).



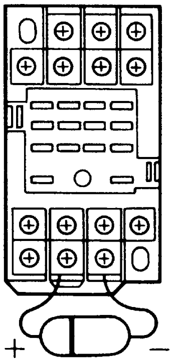
Note Use the shortest possible wiring between the surge absorber and the noise-creating device. Use of excessively long wiring will decrease the performance of the surge absorber.

- The most effective method is to install a surge absorber and surge killer in parallel to the reactance load (solenoid and relay coils). This will reduce noise in a wide band of frequencies.


Surge Absorber (Metal Oxide Varistor or Transzorb).
Surge Killer (Resistor Capacitor Snubber)

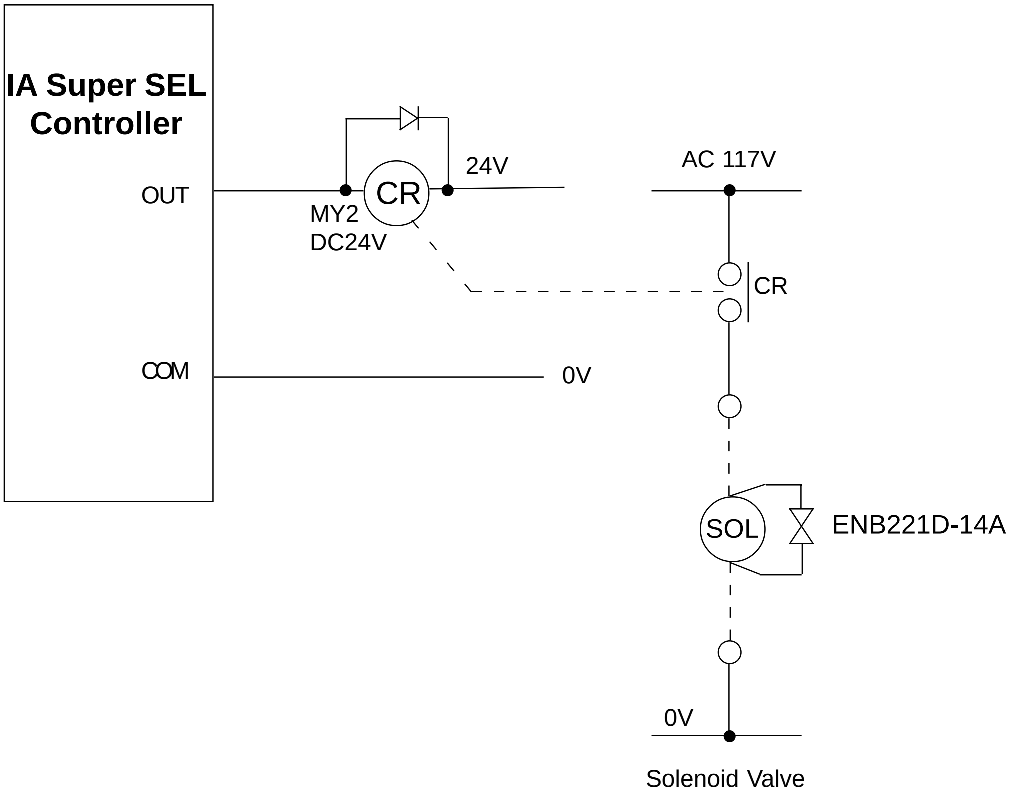
② DC solenoid valve · magnetic switch · relay

- Install a diode parallel with a reactive/inductive load.



- Select a diode with the proper voltage rating. The voltage rating is determined by the loading capacity of the system.
- When installing the diode, pay careful attention to the polarity of the diode. A diode installed in reverse polarity could damage your IA System's internal circuitry.
- If the Controller output will be driving a 24V relay or a 120V solenoid valve, the diode must be installed to reduce any noise made by these devices.

Circuit Example



Chapter 1. Setting Up

Part 4 Cabling Precautions

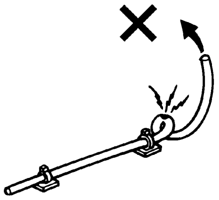
When using the IA actuator and controller to build an application system, it is important to position and lay out the cable correctly. If this is not done, the cable may snap or have a faulty connection that could lead to a variety of problems which in turn could cause the actuator to run out of control. Below, we explain the things not to do to ensure that the cables are connected in the correct fashion.

Ten "Do's and Don'ts" When Laying Out Cable (Please make sure to observe these rules!)

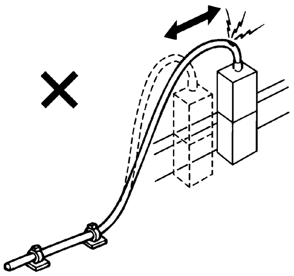
1 Make sure there is no excessive bending at one spot.

×
Steel band
(piano wire)
○
Bundle loosely

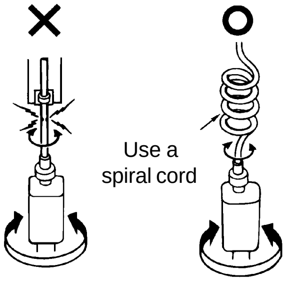
2 Do not twist or crease the cable.



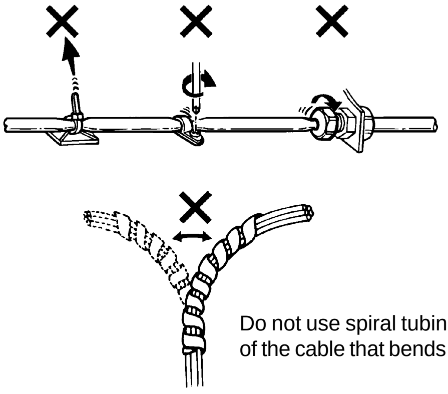
3 Do not stretch the cable too tautly.



4 Do not exert rotational force at a single spot on the cable.

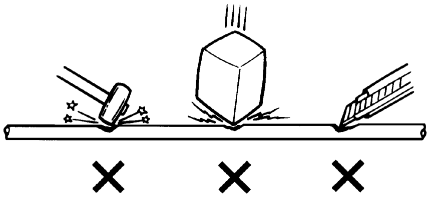


5 When affixing the cable, do not clamp it too tightly.

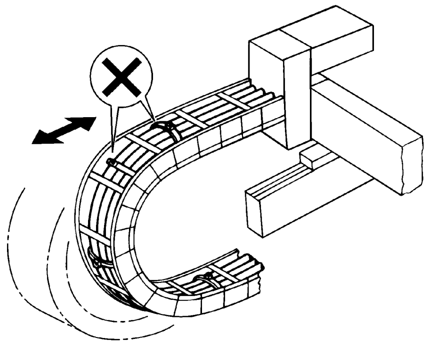


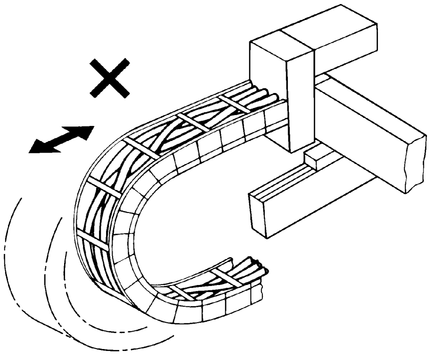
Do not use spiral tubing on a section of the cable that bends frequently.

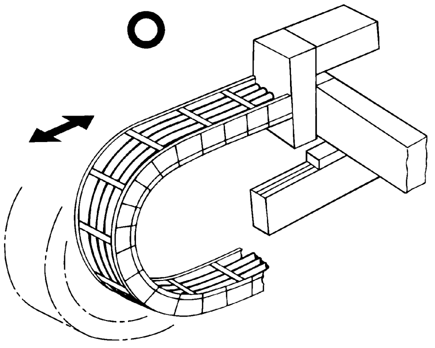
6 Do not cut, dent or let the cable get caught in something.



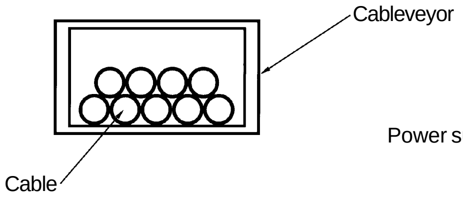
7 If placing cable in a cableveyor or flexible tube, make sure it does not twist around. Also, make sure the cables have some freedom of movement and are not bunched up (cable should not project out at bending points).



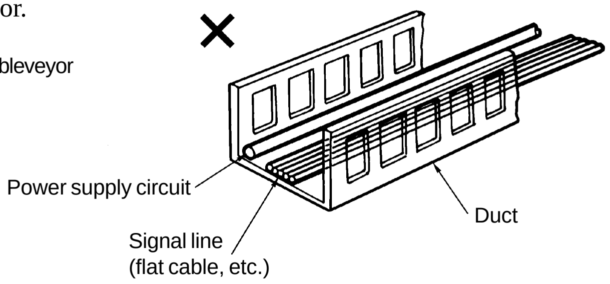




8 The amount of cable placed inside a cableveyor should be about 60% of the space capacity of the cableveyor.

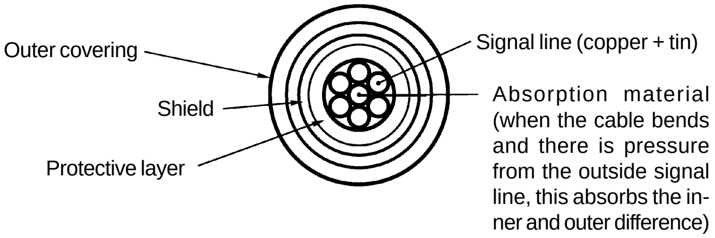


9 Do not mix the signal line with a high voltage circuit.



10 In a case where the cable will be subject to forced bending, always use robot cable.

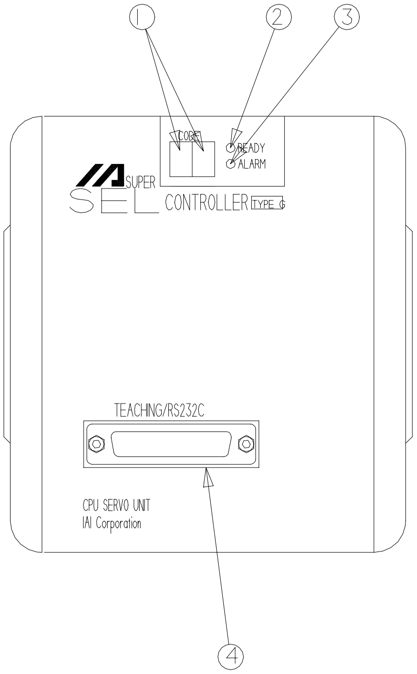
[Standard structure]
Varies according to manufacturer and type


Outer covering
Shield
Protective layer
Signal line (copper + tin)
Absorption material (when the cable bends and there is pressure from the outside signal line, this absorbs the inner and outer difference)

* When to use Robot Cable
When assembling two or three axes and connecting cable to the moving parts, bending weight will be repeatedly applied to the base of the cable. In this case, the cable core is very likely to snap. To prevent this from happening, we strongly recommend the use of robot cable which has greatly improved capacity to withstand bending.

Part 5 Part Names and Functions

1. IA Controller Front View



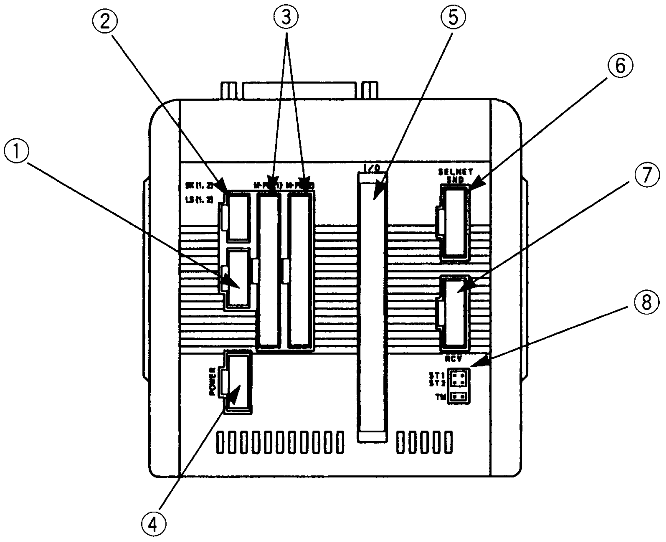
① **Code** (Code Display)
Controller operating status display (2 digit, 7 segment)

② **Ready** (Ready Display LED)
Indicates if the controller is operable.

③ **Alarm** (Alarm Display LED)
Alerts the operator of any abnormality in the system.

④ **Teaching/RS232C Connector**
Connector for Teaching Pendant or PC.

2. Controller Bottom View



① **LS** (Limit Switch Connector) (Option)
5 pins, limit switch connector (Model: Nippon Molex 53258-0520)

② **BK** (Brake Connector) (Option)
4 pins, brake output connector to a brake unit (option) (Model: Nippon Molex 53258-0420)

③ **M·PG** (Servo Motor Output/Encoder Input Connector)
14 pins, servo motor output/encoder input combined type connector (Axis 1 is on the left and Axis 2 is on the right) (Model: Nippon Molex 53258-1420)

④ **Power** (Power Cable Connector)
3 pins, AC power supply cable connector
Since Type E and Type G are designed to be mounted inside of a controller panel, no plug is provided for the power cable on the other end of the controller. (Terminal Type) (Model: Nippon Molex 53265-0320)

*Power Cable, Terminal

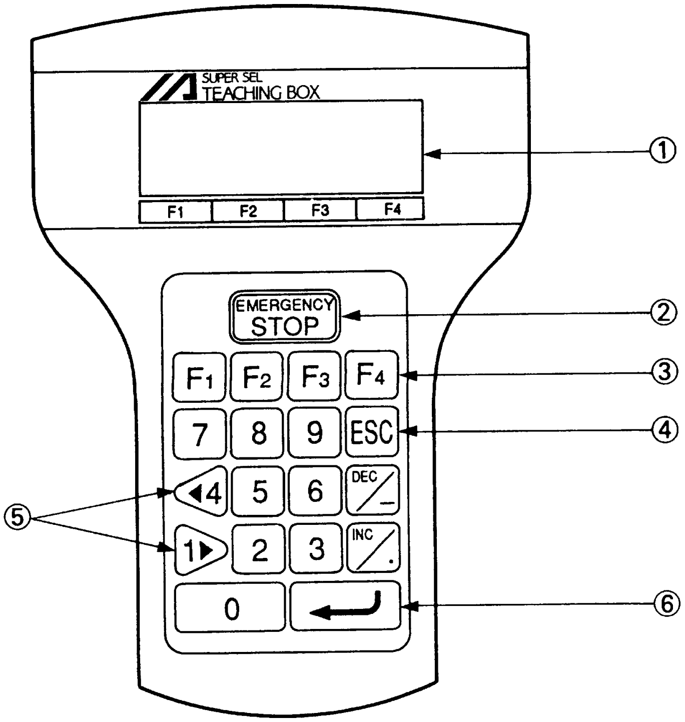
No.	Color	Signal
1	Black	AC117V
2	---	---
3	White	AC117V
4	---	---
5	Green	FG

⑤ **I/O** (I/O Connector)
50 pins, external I/O connector. This is used for Type E and Type G (2 axis) up to 100W only. For other models, it is attached to the I/O Unit Box.
(Model: Sumitomo 3M 7950-6500FL)

⑥⑦ **SEL NET** (Optional SEL Network Connector)
SEL. Network transmission connector. (Model: Nippon Molex 53259-0620)

⑧ **ST1**
Emergency Stop release jumper post. Refer to "Supplement 6. Emergency Stop".

3. Teaching Pendant (Option)



① **LCD Display**
4 lines with a 20 character per line capacity display. Shows program and Motion status.

② **Emergency Stop**
When the emergency stop button is pressed, servos will disengage and all programmable outputs will be turned OFF. To release the emergency stop, press Restart (F1) on the LCD Display.

③ **F1, F2, F3, F4** (Function Key)
Multi-function keys which correspond with the LCD Display.

④ **ESC** (Escape)
The escape key allows the operator to go backwards in one-step increments to previous displays to make corrections or to switch to different modes.

⑤ ◂ ▸
Dual function keys for use in data input and axis Jog functions.

⑥ ↲ (Return Key)
Return key is used to change operations and to move the cursor position.

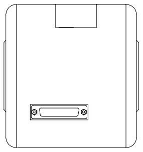
Part 6 Specifications

1. AC Specifications

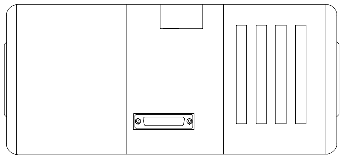
(1) Type E (1 Axis)

Number of Axes		AC60W·100Wx1	AC200Wx1	AC400Wx1
Power Voltage		AC 117V±10%		
Power Frequency		50/60Hz		
Power Consumption		≈175W (100Wx1)	≈345W	≈600W
Ambient Temperature and Humidity		Temperature 0~40°C, Humidity less than 85%RH		
Ambient Environment		No corrosive gas, minimal dust		
Isolation Resistance		500V 10MΩ or more		
Noise Immunity*1		1500V 1μsec pulse by noise simulator		
Unit Weight (kg)		1.2	2.5	3.0
Protective Functions		Driver temperature check Overload check Software limit check		
Motor Capacity		AC servo motor 60W · 100W	AC servo motor 200W	AC servo motor 400W
Memory Capacity		Total 3,000 steps 2,000 positions *2		
Number of Programs		64 programs, 16 multi-tasking programs		
Memory Device		CMOS RAM battery backup		
I/O (DC24V)	Dedicated Input	External start · Emergency stop · Limit switch		
	Standard I/O	24 inputs (including dedicated inputs) 24 outputs (including dedicated outputs)		
	Expansion I/O	Extended I/O: 1 module 24 inputs, 24 outputs		
		---	3	3
	Maximum I/O (including dedicated I/O)	24 inputs *3 24 outputs	96 inputs 96 outputs	96 inputs 96 outputs
		Output maximum load current 100mA/1 point (Recommended 20mA/1 point) Transistor array: TD62083AF compatible		
Data Input Method		Teaching pendant or RS232C communications		
Communications		EIA RS232C Asynchronous, Full duplex		
Network		SEL NET RS232C network		

*1 Controller unit test
*2 The maximum number of steps that can be compiled per unit is 1999.
*3 96 inputs and 96 outputs are available by connecting an optional expansion unit box.



60W·100W

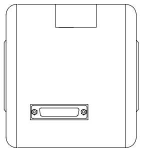


200W
400W

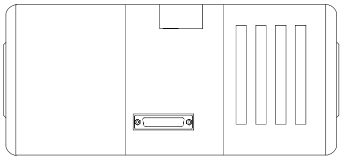
(2) Type G (2-Axis)

Number of Axes		AC60W·100Wx2	AC200Wx2	AC400Wx2
Power Voltage		AC 117V±10%		
Power Frequency		50/60Hz		
Power Consumption		≈300W (100Wx2)	≈590W	≈1100W
Ambient Temperature and Humidity		Temperature 0~40°C, Humidity less than 85%RH		
Ambient Environment		No corrosive gas, minimal dust		
Isolation Resistance		500V 10MΩ or more		
Noise Immunity*1		1500V 1μsec pulse by noise simulator		
Unit Weight (kg)		1.2	2.7	3.5
Protective Functions		Driver temperature check Overload check Software limit check		
Motor Capacity		AC servo motor 60W · 100Wx2	AC servo motor 200Wx2	AC servo motor 400Wx2
Memory Capacity		Total 3,000 steps 2,000 positions *2		
Number of Programs		64 programs, 16 multi-tasking programs		
Memory Device		CMOS RAM battery backup		
I/O (DC24V)	Dedicated Input	External start · Emergency stop · Limit switch		
	Standard I/O	24 inputs (including dedicated inputs) 24 outputs (including dedicated outputs)		
	Expansion I/O	Extended I/O: 1 module 24 inputs, 24 outputs		
		---	3	3
	Maximum I/O (including dedicated I/O)	24 inputs *3 24 outputs	96 inputs 96 outputs	96 inputs 96 outputs
		Output maximum load current 100mA/1 point (Recommended 20mA/1 point) Transistor array: TD62083AF compatible		
Data Input Method		Teaching pendant or RS232C communications		
Communications		EIA RS232C Asynchronous, Full duplex		
Network		SEL NET RS232C network		

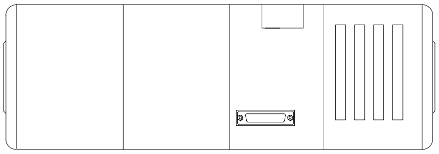
*1 Controller unit test
*2 The maximum number of steps that can be compiled per unit is 1999.
*3 96 inputs and 96 outputs are available by connecting an expansion unit box (option).



60W·100Wx2



200Wx2



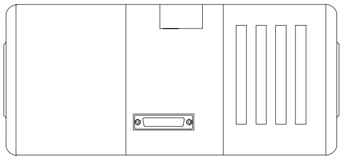
400Wx2

(3) Type G (4-Axis)

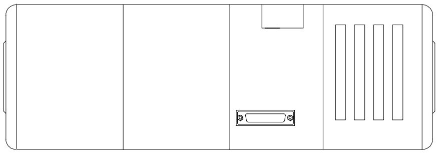
Number of Axes		AC60W·100Wx4	AC200Wx4	AC400Wx4
Power Voltage		AC 117V±10%		
Power Frequency		50/60Hz		
Power Consumption		≈600W (100Wx4)	≈1080W	≈2100W
Ambient Temperature and Humidity		Temperature 0~40°C, Humidity less than 85%RH		
Ambient Environment		No corrosive gas, minimal dust		
Isolation Resistance		500V 10MΩ or more		
Noise Immunity*1		1500V 1μsec pulse by noise simulator		
Unit Weight (kg)		2.7	3.5	4.7
Protective Functions		Driver temperature check Overload check Software limit check		
Motor Capacity		AC servo motor 60W · 100Wx4	AC servo motor 200Wx4	AC servo motor 400Wx4
Memory Capacity		Total 3,000 steps 2,000 positions *2		
Number of Programs		64 programs, 16 multi-tasking programs		
Memory Device		CMOS RAM battery backup		
I/O (DC24V)	Dedicated Input	External start · Emergency stop · Limit switch		
	Standard I/O	24 inputs (including dedicated inputs) 24 outputs (including dedicated outputs)		
	Expansion I/O	Extended I/O: 1 module 24 inputs, 24 outputs		
		3	3	3
	Maximum I/O (including dedicated I/O)	96 inputs 96 outputs	96 inputs 96 outputs	96 inputs 96 outputs
		Output maximum load current 100mA/1 point (Recommended 20mA/1 point) Transistor array: TD62083AF compatible		
Data Input Method		Teaching pendant or RS232C communications		
Communications		EIA RS232C Asynchronous, Full duplex		
Network		SEL NET RS232C network		

*1 Controller unit test

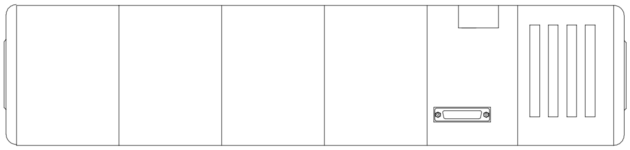
*2 The maximum number of steps that can be compiled per unit is 1999.



60W·100Wx4



200Wx4



400Wx4

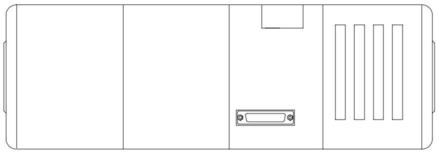
(4) Type G (8-Axis)

Number of Axes		AC60W·100Wx8	AC200Wx8
Power Voltage		AC 117V±10%	
Power Frequency		50/60Hz	
Power Consumption		≈1100W (100Wx8)	≈2000W
Ambient Temperature and Humidity		Temperature 0~40°C, Humidity less than 85%RH	
Ambient Environment		No corrosive gas, minimal dust	
Isolation Resistance		500V 10MΩ or more	
Noise Immunity*1		1500V 1μsec pulse by noise simulator	
Unit Weight (kg)		4.5	5.7
Protective Functions		Driver temperature check Overload check Software limit check	
Motor Capacity		AC servo motor 60W · 100Wx8	AC servo motor 200Wx8
Memory Capacity		Total 3,000 steps 2,000 positions *2	
Number of Programs		64 programs, 16 multi-tasking programs	
Memory Device		CMOS RAM battery backup	
I/O (DC24V)	Dedicated Input	External start · Emergency stop · Limit switch	
	Standard I/O	48 inputs (including dedicated inputs) 48 outputs (including dedicated outputs)	
	Expansion I/O	Extended I/O: 1 module 24 inputs, 24 outputs	
		10	10
	Maximum I/O (including dedicated I/O)	288 inputs 288 outputs	288 inputs 288 outputs
		Output maximum load current 100mA/1 point (Recommended 20mA/1 point) Transistor array: TD62083AF compatible	
Data Input Method		Teaching pendant or RS232C communications	
Communications		EIA RS232C Asynchronous, Full duplex	
Network		SEL NET RS232C network	

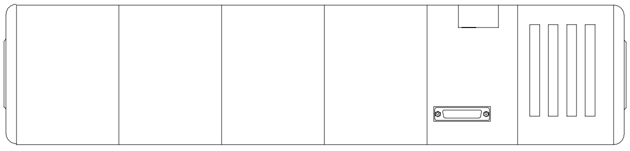
*1 Controller unit test

*2 The maximum number of steps that can be compiled per unit is 1999.

⌖ A 12-slot expansion unit is available as an option (please see P. 61).



60W·100Wx8



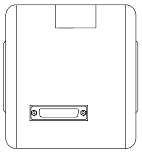
200Wx8

2. DC Specifications

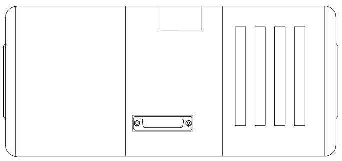
(1) Type E (1 Axis)

Number of Axes		DC20W~100Wx1	DC200Wx1
Power Voltage		AC 117V±10%	
Power Frequency		50/60Hz	
Power Consumption		≈210W (100Wx1)	≈370W
Ambient Temperature and Humidity		Temperature 0~40°C, Humidity less than 85%RH	
Ambient Environment		No corrosive gas, minimal dust	
Isolation Resistance		500V 10MΩ or more	
Noise Immunity*1		1500V 1μsec pulse by noise simulator	
Unit Weight (kg)		1.2	2.5
Protective Functions		Driver temperature check Overload check Software limit check	
Motor Capacity		DC servo motor 20~100W	DC servo motor 200W
Memory Capacity		Total 3,000 steps 2,000 positions *2	
Number of Programs		64 programs, 16 multi-tasking programs	
Memory Device		CMOS RAM battery backup	
I/O (DC24V)	Dedicated Input	External start · Emergency stop · Limit switch	
	Standard I/O	24 inputs (including dedicated inputs) 24 outputs (including dedicated outputs)	
	Expansion I/O	Extended I/O: 1 module 24 inputs, 24 outputs	
		---	3
	Maximum I/O (including dedicated I/O)	24 inputs *3 24 outputs	96 inputs 96 outputs
		Output maximum load current 100mA/1 point (Recommended 20mA/1 point) Transistor array: TD62083AF compatible	
Data Input Method		Teaching pendant or RS232C communications	
Communications		EIA RS232C Asynchronous, Full duplex	
Network		SEL NET RS232C network	

*1 Controller unit test
*2 The maximum number of steps that can be compiled per unit is 1999.
*3 96 inputs and 96 outputs are available by connecting an expansion unit box (option).



20W~100W

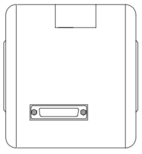


200W

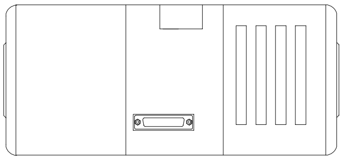
(2) Type G (2-Axis)

Number of Axes		DC20W~100Wx2	DC200Wx2
Power Voltage		AC 117V±10%	
Power Frequency		50/60Hz	
Power Consumption		≈370W (100Wx2)	≈640W
Ambient Temperature and Humidity		Temperature 0~40°C, Humidity less than 85%RH	
Ambient Environment		No corrosive gas, minimal dust	
Isolation Resistance		500V 10MΩ or more	
Noise Immunity*1		1500V 1μsec pulse by noise simulator	
Unit Weight (kg)		1.2	2.7
Protective Functions		Driver temperature check Overload check Software limit check	
Motor Capacity		DC servo motor 20~100Wx2	DC servo motor 200Wx2
Memory Capacity		Total 3,000 steps 2,000 positions *2	
Number of Programs		64 programs, 16 multi-tasking programs	
Memory Device		CMOS RAM battery backup	
I/O (DC24V)	Dedicated Input	External start · Emergency stop · Limit switch	
	Standard I/O	24 inputs (including dedicated inputs) 24 outputs (including dedicated outputs)	
	Expansion I/O	Extended I/O: 1 module 24 inputs, 24 outputs	
		---	3
	Maximum I/O (including dedicated I/O)	24 inputs *3 24 outputs	96 inputs 96 outputs
		Output maximum load current 100mA/1 point (Recommended 20mA/1 point) Transistor array: TD62083AF compatible	
Data Input Method		Teaching pendant or RS232C communications	
Communications		EIA RS232C Asynchronous, Full duplex	
Network		SEL NET RS232C network	

*1 Controller unit test
*2 The maximum number of steps that can be compiled per unit is 1999.
*3 96 inputs and 96 outputs are available by connecting an expansion unit box (option).



20W~100W

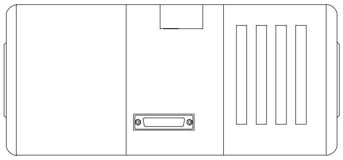


200W

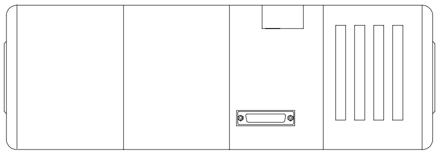
(3) Type G (4-Axis)

Number of Axes		DC20W~100Wx4	DC200Wx4
Power Voltage		AC 117V±10%	
Power Frequency		50/60Hz	
Power Consumption		≈740W (100Wx4)	≈1180W
Ambient Temperature and Humidity		Temperature 0~40°C, Humidity less than 85%RH	
Ambient Environment		No corrosive gas, minimal dust	
Isolation Resistance		500V 10MΩ or more	
Noise Immunity*1		1500V 1μsec pulse by noise simulator	
Unit Weight (kg)		2.7	3.5
Protective Functions		Driver temperature check Overload check Software limit check	
Motor Capacity		DC servo motor 20~100Wx4	DC servo motor 200Wx4
Memory Capacity		Total 3,000 steps 2,000 positions *2	
Number of Programs		64 programs, 16 multi-tasking programs	
Memory Device		CMOS RAM battery backup	
I/O (DC24V)	Dedicated Input	External start · Emergency stop · Limit switch	
	Standard I/O	24 inputs (including dedicated inputs) 24 outputs (including dedicated outputs)	
	Expansion I/O	Extended I/O: 1 module 24 inputs, 24 outputs	
		3	3
	Maximum I/O (including dedicated I/O)	96 inputs 96 outputs	96 inputs 96 outputs
		Output maximum load current 100mA/1 point (Recommended 20mA/1 point) Transistor array: TD62083AF compatible	
Data Input Method		Teaching pendant or RS232C communications	
Communications		EIA RS232C Asynchronous, Full duplex	
Network		SEL NET RS232C network	

*1 Controller unit test
*2 The maximum number of steps that can be compiled per unit is 1999.



20W~100Wx4



200Wx4

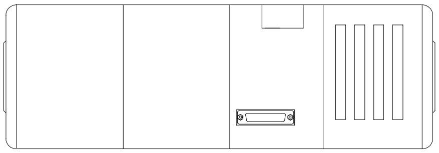
(4) Type G (8-Axis)

Number of Axes		DC20W~100Wx8	DC200Wx8
Power Voltage		AC 117V±10%	
Power Frequency		50/60Hz	
Power Consumption		≈1380W (100Wx8)	≈2260W
Ambient Temperature and Humidity		Temperature 0~40°C, Humidity less than 85%RH	
Ambient Environment		No corrosive gas, minimal dust	
Isolation Resistance		500V 10MΩ or more	
Noise Immunity*1		1500V 1μsec pulse by noise simulator	
Unit Weight (kg)		4.5	5.7
Protective Functions		Driver temperature check Overload check Software limit check	
Motor Capacity		DC servo motor 20~100Wx8	DC servo motor 200Wx8
Memory Capacity		Total 3,000 steps 2,000 positions *2	
Number of Programs		64 programs, 16 multi-tasking programs	
Memory Device		CMOS RAM battery backup	
I/O (DC24V)	Dedicated Input	External start · Emergency stop · Limit switch	
	Standard I/O	48 inputs (including dedicated inputs) 48 outputs (including dedicated outputs)	
	Expansion I/O	Extended I/O: 1 module 24 inputs, 24 outputs	
		10	10
	Maximum I/O (including dedicated I/O)	288 inputs 288 outputs	288 inputs 288 outputs
		Output maximum load current 100mA/1 point (Recommended 20mA/1 point) Transistor array: TD62083AF compatible	
Data Input Method		Teaching pendant or RS232C communications	
Communications		EIA RS232C Asynchronous, Full duplex	
Network		SEL NET RS232C network	

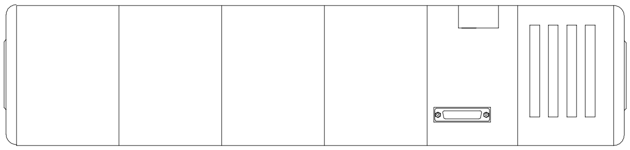
*1 Controller unit test

*2 The maximum number of steps that can be compiled per unit is 1999.

⌖ A 12-slot expansion unit is available as an option (please see P. 61).



20W~100Wx8



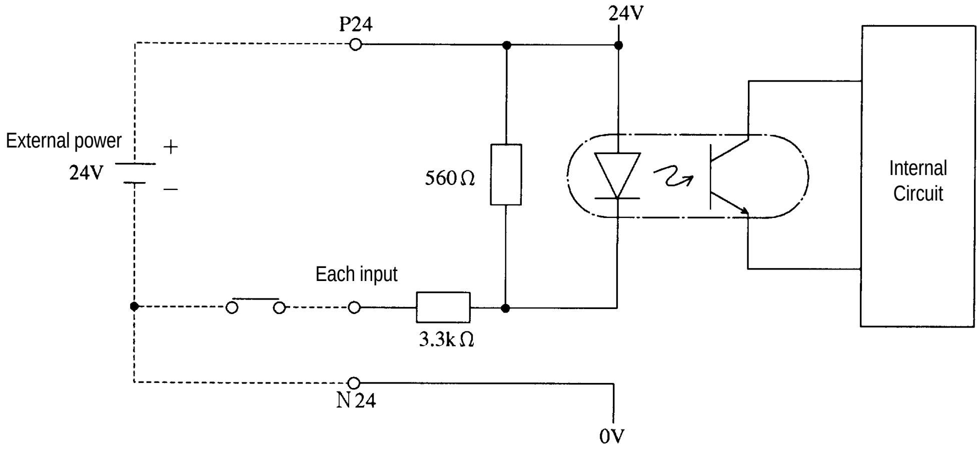
200Wx8

3. External I/O Specifications

(1) Input

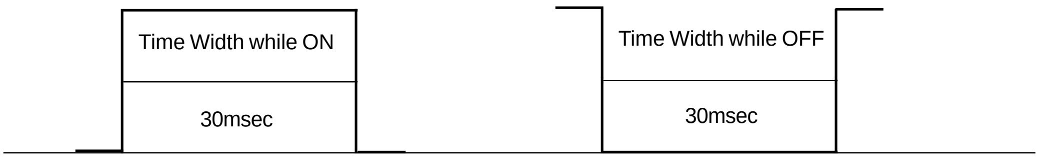
Input	Specifications
Point	Dedicated Input...4 Points User Input....20 Points Expansion Input (Option)....Max 72 Points
Power Voltage	DC24V +/-20%
Current	7mA/DC24V
ON/OFF Power Voltage	ON....Min DC16.0V OFF....Max DC5.0V
ON/OFF Response Time	ON....Max 20m sec OFF....Max 20m sec
Isolation Method	Photocoupler

Internal Circuit



1) The power must be supplied externally for the standard I/O board and the expansion I/O board.
2) For the external circuit connection (no contact point), the leakage per 1point must be kept lower than 1mA when the switch is OFF.

Guaranteed operation width of the Super SEL Controller input signal

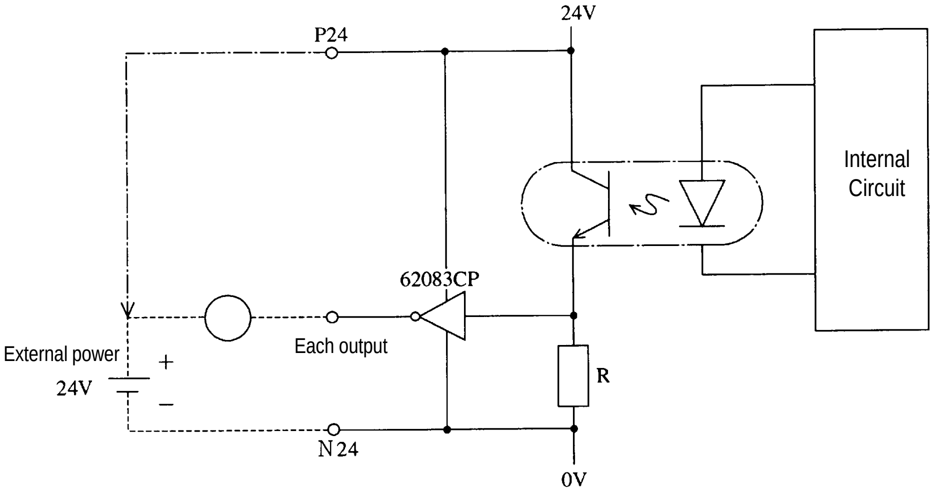


Generally, the input signal operates within 25msec but the guaranteed operation width is 30msec.

(2) Output

	Specifications
Output Point	Dedicated output.....2 points User output.....22 points Expansion output (option).....maximum72 points
Rated Power	DC24V
Maximum Load Current	100mA/1point
Recommended Load Current	20mA/1 point
Leakage	0.1mA (maximum)
Residual Voltage	3.1V/40mA (maximum)
Insulation Method	Photocoupler

Internal Circuit



The power must be supplied externally for the standard I/O board and the expansion I/O board.

*You will damage the output element if you overload or underload it.

4. Servo

	Specifications
Control Method	Semi-closed loop control
Position Feedback	Rotary encoder A · B · Z phase
Repeatability	± 2 pulses
Velocity	1m/sec ~ 1500m/sec *
Acceleration	0.01G ~ 1G

*Max velocity differs depending on actuator specifications.

5. Precautions When Using the Emergency Stop

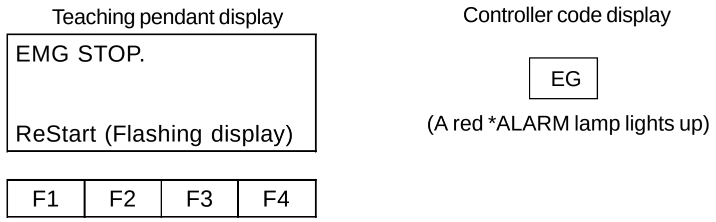
As a rule, emergency stops should only be applied from the I/O.
Do not turn the power (AC117V) ON/OFF to effect an emergency stop.
If you stop the actuator by turning the power OFF, wait at least 15 seconds before turning the power ON again. If you disregard this warning, and repeatedly turn the power ON/OFF without waiting a sufficient amount of time, you may damage the controller.

6. Restarting the Controller After an Emergency Stop (refer to part 3, 1-4 "Emergency Stop Release" for details)

The Super SEL controller and Table Top type (TT-300) both use a "hard reset" to restart after an emergency stop. The operation is nearly the same as turning the power OFF/ON. (Homing is required).

(1) Emergency Stop from the teaching pendant

① Press EMERGENCY STOP on the teaching pendant. Continue pressing and the screen will display the following.



Teaching pendant display

```
EMG STOP.

ReStart (Flashing display)
```

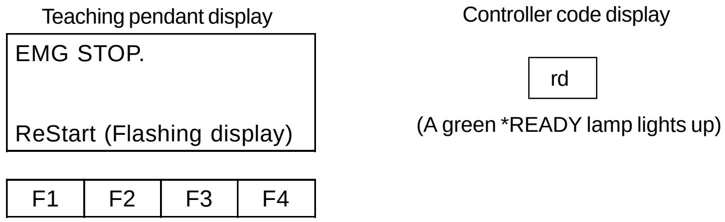
F1	F2	F3	F4

Controller code display

EG

(A red *ALARM lamp lights up)

② Take your finger off the EMERGENCY STOP button to do a hard reset and the following screen appears.



Teaching pendant display

```
EMG STOP.

ReStart (Flashing display)
```

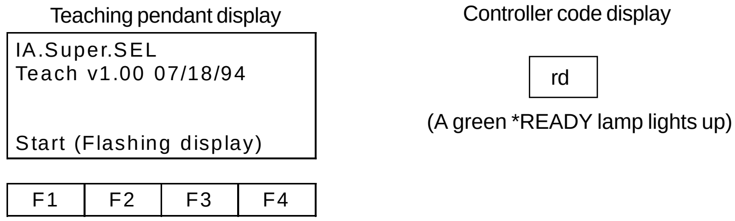
F1	F2	F3	F4

Controller code display

rd

(A green *READY lamp lights up)

③ If you press the F1 key (ReStart) on the teaching pendant, the initial screen reappears.



Teaching pendant display

```
IA.Super.SEL
Teach v1.00 07/18/94

Start (Flashing display)
```

F1	F2	F3	F4

Controller code display

rd

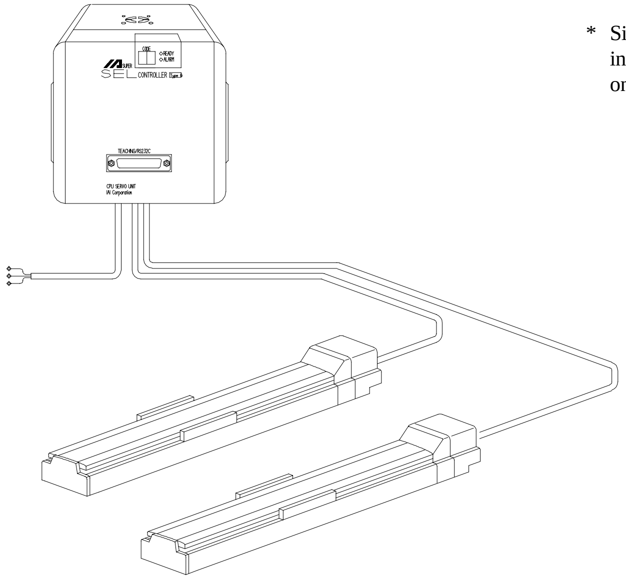
(A green *READY lamp lights up)

(2) Pressing the controller emergency stop button or an emergency stop condition caused by an external signal
When the emergency stop is released after pressing the emergency stop button on the controller front panel, you must follow the same procedure as described above or the teaching pendant will not reset (you cannot operate the teaching box if the code display on the controller front panel reads $\boxed{\text{EG}}$.

> ! **Warning**
> If you are using the Auto Start PRG in the system program parameter mode, always write the program so that movement will not resume unless there is some kind of input condition. This is to avoid sudden startup of movement because of the automatic start program right after the emergency stop is released.

Part 7 System Setup

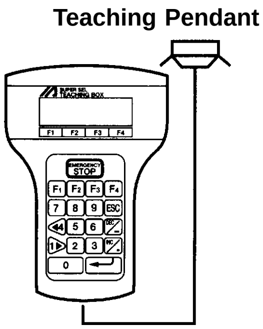
1. Connecting the IA Controller and Actuator



* Since Type E and Type G Controllers are designed to be mounted inside of a control panel, no plug is provided for the power cable on the other end of the controller.

*Power Cable·Terminal

No.	Color	Signal
1	Black	AC117V
2	---	---
3	White	AC117V
4	---	---
5	Green	FG



① Connect controller and actuator cables. Use only IA supplied cables. These cables include:
   a. Motor cable
   b. Encoder cable
   c. Teaching pendant cable
   d. Brake cable (optional)
   e. Limit switch cable (optional)

② Connect the power cord.

③ The CODE display shows oP → Su → SC → PS → rd in sequence. The Super SEL controller is ready to operate. If the CODE display is EG release the Emergency Stop input.
   Note: The Emergency Stop is normally closed (b contact point input). In order to release the emergency stop, short the jumper post (ST1) at the bottom of the controller CPU UNIT or CPU SERVO UNIT with the jumper pin. Refer to Supplement 6 for the actual procedure.

2. Interface List

I/O Connector (NPN-Sinking)

Pin No.	Category	Port No.	Function	Cable
1A	P24		---	1-Brown
1B	Input	000	External Start Input	1-Red
2A		001	User Input	1-Orange
2B		002	Emergency Stop b Contact Input *	1-Yellow
3A		003	SystemReserve	1-Green
3B		004	SystemReserve	1-Blue
4A		005	User Input	1-Purple
4B		006	User Input	1-Gray
5A		007	User Input	1-White
5B		008	PRG No. 1 (User Input)	1-Black
6A		009	PRG No. 2 (User Input)	2-Brown
6B		010	PRG No. 4 (User Input)	2-Red
7A		011	PRG No. 8 (User Input)	2-Orange
7B		012	PRG No. 10 (User Input)	2-Yellow
8A		013	PRG No. 20 (User Input)	2-Green
8B		014	PRG No. 40 (User Input)	2-Blue
9A		015	User Input	2-Purple
9B		016	User Input	2-Gray
10A		017	User Input	2-White
10B		018	User Input	2-Black
11A		019	User Input	3-Brown
11B		020	User Input	3-Red
12A		021	User Input	3-Orange
12B		022	User Input	3-Yellow
13A		023	User Input	3-Green
13B	Output	300	Emergency Stop/Alarm Output	3-Blue
14A		301	Ready Output	3-Purple
14B		302	User Output	3-Gray
15A		303	User Output	3-White
15B		304	User Output	3-Black
16A		305	User Output	4-Brown
16B		306	User Output	4-Red
17A		307	User Output	4-Orange
17B		308	User Output	4-Yellow
18A		309	User Output	4-Green
18B		310	User Output	4-Blue
19A		311	User Output	4-Purple
19B		312	User Output	4-Gray
20A		313	User Output	4-White
20B		314	User Output	4-Black
21A		315	User Output	5-Brown
21B		316	User Output	5-Red
22A		317	User Output	5-Orange
22B		318	User Output	5-Yellow
23A		319	User Output	5-Green
23B		320	User Output	5-Blue
24A		321	User Output	5-Purple
24B		322	User Output	5-Gray
25A		323	User Output	5-White
25B	N24		---	5-Black

* Emergency Stop (normally closed). To release the emergency stop, short circuit the jumper post with a jumper pin or connect pin 2B and pin 25B. Refer to Supplement 6.

* Pin No.3A (Port No.003) and Pin No.3B (Port No.004) cannot be used as user input.

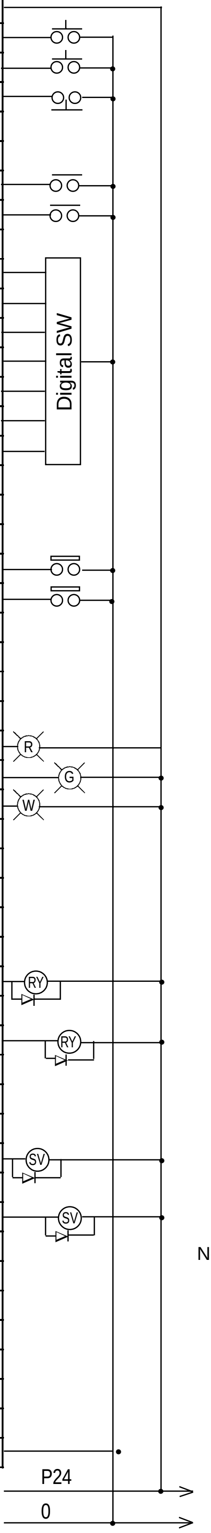
* Connector:
Sumitomo 3M7950-6500SC or
Yamaichi FAP-5001-1202

3. I/O Wiring Diagram

Caution ! NPN I/O wirings and PNP wirings are DIFFERENT

Standard I/O (NPN - Sinking)

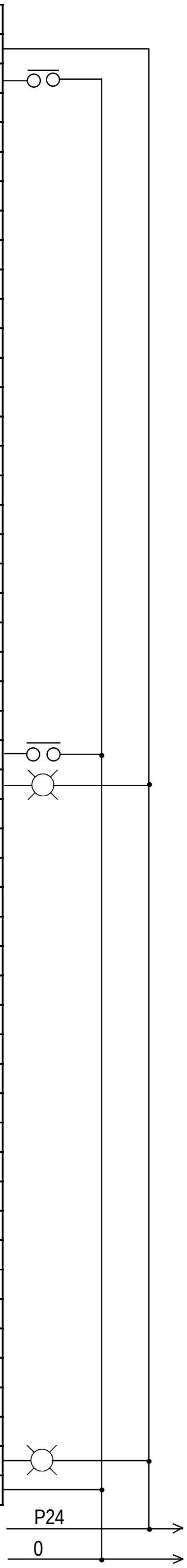
Pin No.	Category	Port No.	Function
1A	P24		---
1B	Input	000	External Start Input
2A		001	User Input
2B		002	Emergency Stop b Contact Input *
3A		003	SystemReserve
3B		004	SystemReserve
4A		005	User Input
4B		006	User Input
5A		007	User Input
5B		008	PRG No. 1 (User Input)
6A		009	PRG No. 2 (User Input)
6B		010	PRG No. 4 (User Input)
7A		011	PRG No. 8 (User Input)
7B		012	PRG No. 10 (User Input)
8A		013	PRG No. 20 (User Input)
8B		014	PRG No. 40 (User Input)
9A		015	User Input
9B		016	User Input
10A		017	User Input
10B		018	User Input
11A		019	User Input
11B		030	User Input
12A		021	User Input
12B		022	User Input
13A		023	User Input
13B	Output	300	Emergency Stop/Alarm Output
14A		301	Ready Output
14B		302	User Output
15A		303	User Output
15B		304	User Output
16A		305	User Output
16B		306	User Output
17A		307	User Output
17B		308	User Output
18A		309	User Output
18B		310	User Output
19A		311	User Output
19B		312	User Output
20A		313	User Output
20B		314	User Output
21A		315	User Output
21B		316	User Output
22A		317	User Output
22B		318	User Output
23A		319	User Output
23B		320	User Output
24A		321	User Output
24B		322	User Output
25A		323	User Output
25B	N24		---



Note: The same cable colors are used for the standard cables and the expansion cables.
*Standard I/O Pin No.3A (Port No.003), and Pin No.3B(Port No.004) cannot be used as user inputs.

Expansion I/O * (NPN - Sinking)

Pin No.	Category	Port No.	Function
1A	P24		External Power Supply +24V Input
1B	Input	024	User Input
2A		025	User Input
2B		026	User Input
3A		027	User Input
3B		028	User Input
4A		029	User Input
4B		030	User Input
5A		031	User Input
5B		032	User Input
6A		033	User Input
6B		034	User Input
7A		035	User Input
7B		036	User Input
8A		037	User Input
8B		038	User Input
9A		039	User Input
9B		040	User Input
10A		041	User Input
10B		042	User Input
11A		043	User Input
11B		044	User Input
12A		045	User Input
12B		046	User Input
13A		047	User Input
13B	Output	324	User Output
14A		325	User Output
14B		326	User Output
15A		327	User Output
15B		328	User Output
16A		329	User Output
16B		330	User Output
17A		331	User Output
17B		332	User Output
18A		333	User Output
18B		334	User Output
19A		335	User Output
19B		336	User Output
20A		337	User Output
20B		338	User Output
21A		339	User Output
21B		340	User Output
22A		341	User Output
22B		342	User Output
23A		343	User Output
23B		344	User Output
24A		345	User Output
24B		346	User Output
25A		347	User Output
25B	N24		External Power Supply 0V



* This is the first expansion I/O. In the second expansion I/O, the port numbers continue on from these numbers.

4. Interface List (PNP Sourcing)

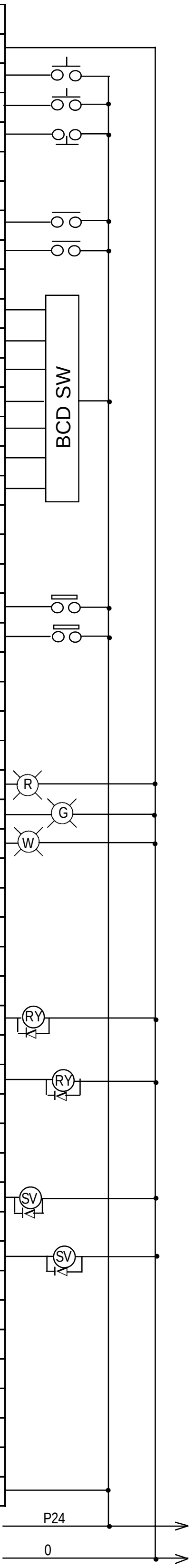
Pin No.	Category	Port No.	Function	Cable Color
1A	N24		External power 0V	1-Brown
1B	Input	000	External start input	1-Red
2A		001	User input	1-Orange
2B		002	Emergency Stop b contact input *1	1-Yellow
3A		003	System reserve	1-Green
3B		004	System reserve	1-Blue
4A		005	User input	1-Purple
4B		006	User input	1-Gray
5A		007	User input	1-White
5B		008	PRG No. 1 (user input)	1-Black
6A		009	PRG No. 2 (user input)	1-Brown
6B		010	PRG No. 4 (user input)	2-Red
7A		011	PRG No. 8 (user input)	2-Orange
7B		012	PRG No. 10 (user input)	2-Yellow
8A		013	PRG No. 20 (user input)	2-Green
8B		014	PRG No. 40 (user input)	2-Blue
9A		015	User input	2-Purple
9B		016	User input	2-Gray
10A		017	User input	2-White
10B		018	User input	2-Black
11A		019	User input	3-Brown
11B		020	User input	3-Red
12A		021	User input	3-Orange
12B		022	User input	3-Yellow
13A		023	User input	3-Green
13B	Output	300	Emergency stop/Alarm output	3-Blue
14A		301	Ready output	3-Purple
14B		302	User output	3-Gray
15A		303	User output	3-White
15B		304	User output	3-Black
16A		305	User output	4-Brown
16B		306	User output	4-Red
17A		307	User output	4-Orange
17B		308	User output	4-Yellow
18A		309	User output	4-Green
18B		310	User output	4-Blue
19A		311	User output	4-Purple
19B		312	User output	4-Gray
20A		313	User output	4-White
20B		314	User output	4-Black
21A		315	User output	5-Brown
21B		316	User output	5-Red
22A		317	User output	5-Orange
22B		318	User output	5-Yellow
23A		319	User output	5-Green
23B		320	User output	5-Blue
24A		321	User output	5-Purple
24B		322	User output	5-Gray
25A		323	User output	5-White
25B	P24		External power +24V	5-Black

* Emergency Stop is a b-contact input (normally closed). To release the emergency stop, short circuit the jumper post with the jumper pin. Refer to Supplement 7.
* Pin No.3A (Port No.003), and Pin No.3B (Port No.004) cannot be used as user inputs.
* Connector: Sumitomo 3M 7950-6500SC or Yamaichi FAP-5001-1202.

5. Wiring Diagram

Standard I/O (PNP - Sourcing)

Pin No.	Category	Port No.	Function	Cable Color
1A	N24		External power 0V	1-Brown
1B	Input	000	External start input	1-Red
2A		001	User input	1-Orange
2B		002	Emergency Stop b contact input	1-Yellow
3A		003	System reserve	1-Green
3B		004	System reserve	1-Blue
4A		005	User input	1-Purple
4B		006	User input	1-Gray
5A		007	User input	1-White
5B		008	PRG No. 1 (user input)	1-Black
6A		009	PRG No. 2 (user input)	1-Brown
6B		010	PRG No. 4 (user input)	2-Red
7A		011	PRG No. 8 (user input)	2-Orange
7B		012	PRG No. 10 (user input)	2-Yellow
8A		013	PRG No. 20 (user input)	2-Green
8B		014	PRG No. 40 (user input)	2-Blue
9A		015	User input	2-Purple
9B		016	User input	2-Gray
10A		017	User input	2-White
10B		018	User input	2-Black
11A		019	User input	3-Brown
11B		020	User input	3-Red
12A		021	User input	3-Orange
12B		022	User input	3-Yellow
13A		023	User input	3-Green
13B	Output	300	Emergency stop/Alarm output	3-Blue
14A		301	Ready output	3-Purple
14B		302	User output	3-Gray
15A		303	User output	3-White
15B		304	User output	3-Black
16A		305	User output	4-Brown
16B		306	User output	4-Red
17A		307	User output	4-Orange
17B		308	User output	4-Yellow
18A		309	User output	4-Green
18B		310	User output	4-Blue
19A		311	User output	4-Purple
19B		312	User output	4-Gray
20A		313	User output	4-White
20B		314	User output	4-Black
21A		315	User output	5-Brown
21B		316	User output	5-Red
22A		317	User output	5-Orange
22B		318	User output	5-Yellow
23A		319	User output	5-Green
23B		320	User output	5-Blue
24A		321	User output	5-Purple
24B		322	User output	5-Gray
25A		323	User output	5-White
25B	P24		External power +24V	5-Black

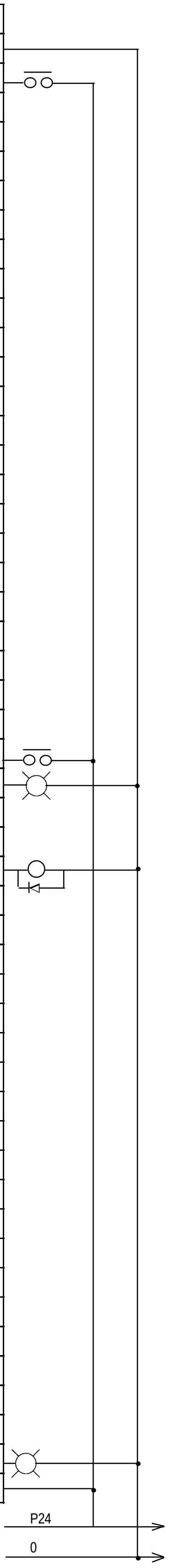


Note: The same cable colors are used for the standard cables and the expansion cables.
*Standard I/O Pin No.3A (Port No.003), and Pin No.3B(Port No.004) cannot be used as user inputs.

NOTE:
For the SEL EU controller, if the motor/encoder cables are not connected to the controller, the controller will be in a permanent emergency stop condition (even if the E-stop input is jumpered to 24VDC for a PNP I/O board).

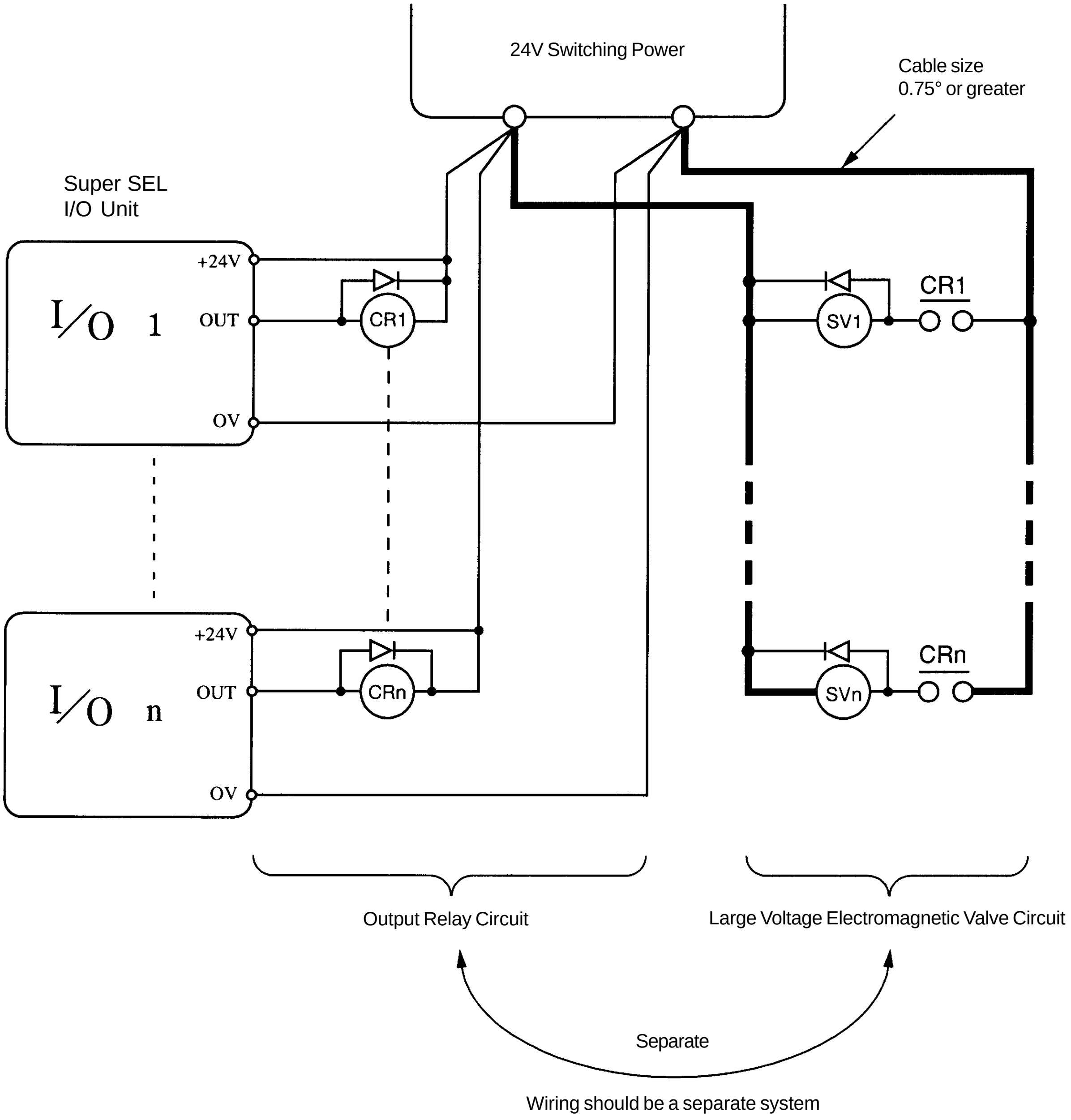
I/O Expansion * (PNP-Sourcing)

Pin No.	Category	Port No.	Function	Cable Color
1A	N24		External power 0V	1-Brown
1B	Input	024	User input	1-Red
2A		025	User input	1-Orange
2B		026	User input	1-Yellow
3A		027	User input	1-Green
3B		028	User input	1-Blue
4A		029	User input	1-Purple
4B		030	User input	1-Gray
5A		031	User input	1-White
5B		032	User input	1-Black
6A		033	User input	1-Brown
6B		034	User input	2-Red
7A		035	User input	2-Orange
7B		036	User input	2-Yellow
8A		037	User input	2-Green
8B		038	User input	2-Blue
9A		039	User input	2-Purple
9B		040	User input	2-Gray
10A		041	User input	2-White
10B		042	User input	2-Black
11A		043	User input	3-Brown
11B		044	User input	3-Red
12A		045	User input	3-Orange
12B		046	User input	3-Yellow
13A		047	User input	3-Green
13B	Output	324	User output	3-Blue
14A		325	User output	3-Purple
14B		326	User output	3-Gray
15A		327	User output	3-White
15B		328	User output	3-Black
16A		329	User output	4-Brown
16B		330	User output	4-Red
17A		331	User output	4-Orange
17B		332	User output	4-Yellow
18A		333	User output	4-Green
18B		334	User output	4-Blue
19A		335	User output	4-Purple
19B		336	User output	4-Gray
20A		337	User output	4-White
20B		338	User output	4-Black
21A		339	User output	5-Brown
21B		340	User output	5-Red
22A		341	User output	5-Orange
22B		342	User output	5-Yellow
23A		343	User output	5-Green
23B		344	User output	5-Blue
24A		345	User output	5-Purple
24B		346	User output	5-Gray
25A		347	User output	5-White
25B	P24		External power +24V	5-Black



* This is the first expansion I/O. In the second expansion I/O, the port numbers continue on from these numbers.

DC24V Electromagnetic Valve Wiring Precautions (Example)



In a situation where the I/O unit drives the relay and the relay drives the electromagnetic valve, separate the output relay circuit and the large voltage electromagnetic valve circuit as shown in the diagram above.

4. Teaching/RS232C Connector (D-Sub 25 DTE Special *)

Pin No.	Signal
1	FG
2	TXD
3	RXD
4	(RTS)
5	(CTS)
6	DSR
7	SG
8	NC
9	NC
10	NC
11	NC
12	NC
13	NC

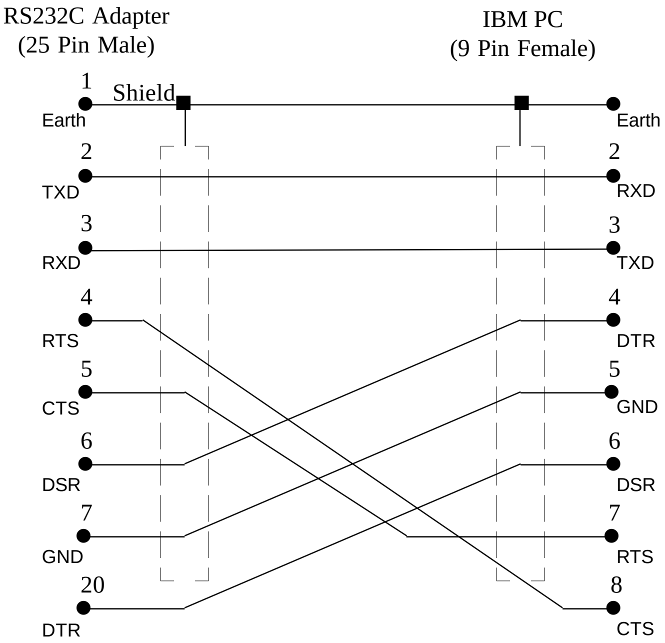
Pin No.	Signal	
14	NC	
15	NC	
16	NC	
17	NC	
18	+6.2V Output	*
19	NC	
20	DTR	
21	NC	
22	NC	
23	Emergency Stop Switch (EMG.SW)	*
24	NC	
25	0V (+6.2V)	*

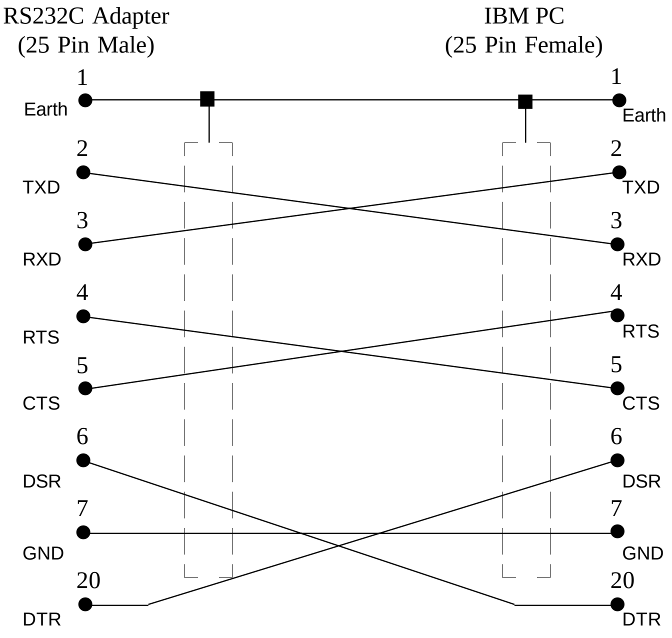
* Pin numbers 18, 23, and 25 are for use with the teaching pendant signal. Do not connect these pins.

- Pin numbers 4 and 5 are short-circuited.

■RS232C Cable

Use RS232C cable pin configuration (between controller and computer serial port)

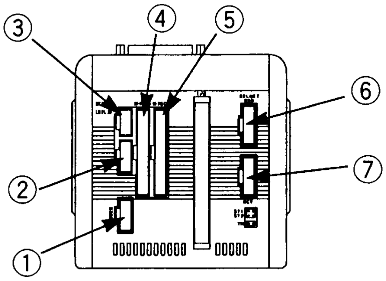




5. Connector Pin Assignment

The bottom view of the Super SEL Type E/G on the left shows the placement of the connectors. Please refer to Page 29 and 31 for the I/O connector and the I/O wiring (including the I/O expansion).

The connector pin assignment for the other parts are shown below.



(1) AC100W 2-Axis specifications

① Power Supply Connector

Pin No.	Signal
1	AC117V
3	AC117V
5	FG

Nippon Molex 53265-0320 (3P) (Body side)
51067-0500 Housing (5P)
50217-8100 x 3 Terminal

② LS Connector (Option)

Pin No.	Signal
1	P24V
2	N
3	XLS
4	YLS
5	EMG stop contact point input*

* The Emergency Stop is normally open.

Nippon Molex 53258-0520 (5P) (Body side)
51067-0500 Housing (5P)
50217-8100 x 5 Terminal

③ BK Connector (Option)

Pin No.	Signal
1	60V
2	GD
3	XBK
4	YBK

Nippon Molex 53258-0420 (4P) (Body side)
51067-0500 Housing (4P)
50217-8100 x 4 Terminal

④ M·PG Connector (Motor/Encoder Signal)

Pin No.	Signal	
1	U	X (1) Axis
2	V	
3	W	
4	NC	
5	FG	
6	PV5	
7	GD	
8	A	X (1) Axis
9	$\overline{A}$	
10	B	
11	$\overline{B}$	
12	Z	
13	$\overline{Z}$	
14	FG	

Nippon Molex 53258-1420 (14P) (Body side)
51067-1400 Housing (14P)
50217-8100 x 14 Terminal

⑤ M·PG Connector (Motor/Encoder Signal)

Pin No.	Signal	
1	U	
2	V	Y (2) Axis
3	W	
4	NC	
5	FG	
6	PV5	
7	GD	
8	A	
9	$\overline{A}$	
10	B	
11	$\overline{B}$	Y (2) Axis
12	Z	
13	$\overline{Z}$	
14	FG	

Nippon Molex	53258-1420	(14P) (Body side)
	51067-1400	Housing (14P)
	50217-8100	x 14 Terminal

⑥ SEL NET SND Connector (Option)

Pin No.	Signal
1	NC
2	NC
3	RD
4	TD
5	GD
6	FG

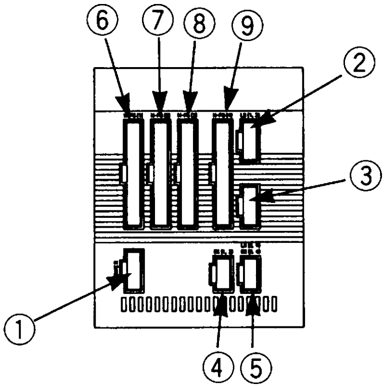
Nippon Molex	53259-0620	(6P) (Body side)
	51067-0600	Housing (6P)
	50217-8100	x 6 Terminal

⑦ SEL NET RCV Connector (Option)

Pin No.	Signal
1	NC
2	NC
3	RD
4	TD
5	GD
6	FG

Nippon Molex	53259-0620	(6P) (Body side)
	51067-0600	Housing (6P)
	50217-8100	x 6 Terminal

(2) AC100W 4-Axis specifications



① Power Supply Connector

Pin No.	Signal
1	AC117V
3	AC117V
5	FG

Nippon Molex 53265-0320 (3P) (Body side)
51067-0500 Housing (5P)
50217-8100 x 3 Terminal

② LS Connector (Option)

Pin No.	Signal
1	P24V
2	N
3	XLS
4	YLS
5	EMG stop contact point input*

* The Emergency Stop is normally open.

Nippon Molex 53258-0520 (5P) (Body side)
51067-0500 Housing (5P)
50217-8100 x 5 Terminal

③ LS Connector (Option)

Pin No.	Signal
1	P24V
2	N
3	ZLS
4	θLS
5	EMG stop contact point input*

* The Emergency Stop is normally open.

Nippon Molex 53258-0520 (5P) (Body side)
51067-0500 Housing (5P)
50217-8100 x 5 Terminal

④ BK Connector (Option)

Pin No.	Signal
1	60V
2	GD
3	XBK
4	YBK

Nippon Molex 53258-0420 (4P) (Body side)
51067-0400 Housing (4P)
50217-8100 x 4 Terminal

⑤ BK Connector (Option)

Pin No.	Signal
1	60V
2	GD
3	XBK
4	θBK

Nippon Molex	53258-0420	(4P) (Body side)
	51067-0400	Housing (4P)
	50217-8100	x 4 Terminal

⑥ M·PG Connector (Motor/Encoder Signal)

Pin No.	Signal	
1	U	X (1) Axis
2	V	
3	W	
4	NC	
5	FG	
6	PV5	
7	GD	
8	A	X (1) Axis
9	$\overline{A}$	
10	B	
11	$\overline{B}$	
12	Z	
13	$\overline{Z}$	
14	FG	

Nippon Molex	53258-1420	(14P) (Body side)
	51067-1400	Housing (14P)
	50217-8100	x 14 Terminal

⑦ M·PG Connector (Motor/Encoder Signal)

Pin No.	Signal	
1	U	Y (2) Axis
2	V	
3	W	
4	NC	
5	FG	
6	PV5	
7	GD	
8	A	Y (2) Axis
9	$\overline{A}$	
10	B	
11	$\overline{B}$	
12	Z	
13	$\overline{Z}$	
14	FG	

Nippon Molex	53258-1420	(14P) (Body side)
	51067-1400	Housing (14P)
	50217-8100	x 14 Terminal

⑧ M·PG Connector (Motor/Encoder Signal)

Pin No.	Signal	
1	U	Z (3) Axis
2	V	
3	W	
4	NC	
5	FG	
6	PV5	
7	GD	
8	A	Z (3) Axis
9	$\overline{A}$	
10	B	
11	$\overline{B}$	
12	Z	
13	$\overline{Z}$	
14	FG	

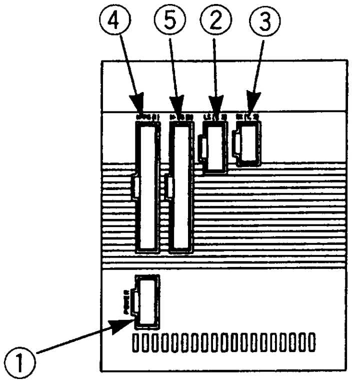
Nippon Molex	53258-1420	(14P) (Body side)
	51067-1400	Housing (14P)
	50217-8100	x 14 Terminal

⑨ M·PG Connector (Motor/Encoder Signal)

Pin No.	Signal	
1	U	θ (4) Axis
2	V	
3	W	
4	NC	
5	FG	
6	PV5	
7	GD	
8	A	θ (4) Axis
9	$\overline{A}$	
10	B	
11	$\overline{B}$	
12	Z	
13	$\overline{Z}$	
14	FG	

Nippon Molex	53258-1420	(14P) (Body side)
	51067-1400	Housing (14P)
	50217-8100	x 14 Terminal

(3) AC200W 2-Axis specifications



① Power Supply Connector

Pin No.	Signal
1	AC117V
3	AC117V
5	FG

Nippon Molex	53265-0320	(3P) (Body side)
	51067-0500	Housing (5P)
	50217-8100	x 3 Terminal

② LS Connector (Option)

Pin No.	Signal
1	P24V
2	N
3	XLS
4	YLS
5	EMG stop contact point input*

* The Emergency Stop is normally open.

Nippon Molex	53258-0520	(5P) (Body side)
	51067-0500	Housing (5P)
	50217-8100	x 5 Terminal

③ BK Connector (Option)

Pin No.	Signal
1	60V
2	GD
3	XBK
4	YBK

Nippon Molex	53258-0420	(4P) (Body side)
	51067-0400	Housing (4P)
	50217-8100	x 4 Terminal

④ M·PG Connector (Motor/Encoder Signal)

Pin No.	Signal	
1	U	
2	V	X (1) Axis
3	W	
4	NC	
5	FG	
6	PV5	
7	GD	
8	A	
9	$\overline{A}$	
10	B	
11	$\overline{B}$	X (1) Axis
12	Z	
13	$\overline{Z}$	
14	FG	

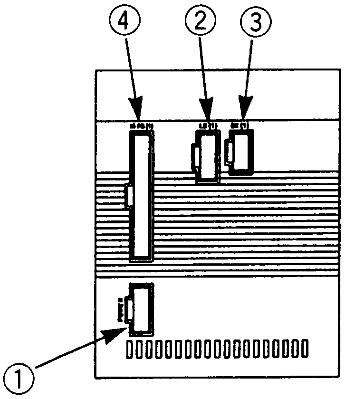
Nippon Molex	53258-1420	(14P) (Body side)
	51067-1400	Housing (14P)
	50217-8100	x 14 Terminal

⑤ M·PG Connector (Motor/Encoder Signal)

Pin No.	Signal	
1	U	
2	V	Y (2) Axis
3	W	
4	NC	
5	FG	
6	PV5	
7	GD	
8	A	
9	$\overline{A}$	
10	B	
11	$\overline{B}$	Y (2) Axis
12	Z	
13	$\overline{Z}$	
14	FG	

Nippon Molex	53258-1420	(14P) (Body side)
	51067-1400	Housing (14P)
	50217-8100	x 14 Terminal

(4) AC400W 1-Axis specifications



① Power Supply Connector

Pin No.	Signal
1	AC117V
3	AC117V
5	FG

Nippon Molex	53265-0320	(3P) (Body side)
	51067-0500	Housing (5P)
	50217-8100	x 3 Terminal

② LS Connector (Option)

Pin No.	Signal
1	P24V
2	N
3	XLS
4	YLS
5	EMG stop contact point input*

* The Emergency Stop is normally open.

Nippon Molex	53258-0520	(5P) (Body side)
	51067-0500	Housing (5P)
	50217-8100	x 5 Terminal

③ BK Connector (Option)

Pin No.	Signal
1	60V
2	GD
3	XBK
4	YBK

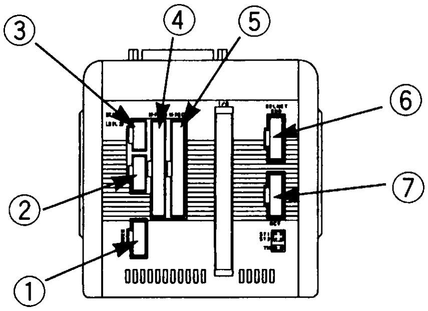
Nippon Molex	53258-0420	(4P) (Body side)
	51067-0400	Housing (4P)
	50217-8100	x 4 Terminal

④ M·PG Connector (Motor/Encoder Signal)

Pin No.	Signal	
1	U	
2	V	X (1) Axis
3	W	
4	NC	
5	FG	
6	PV5	
7	GD	
8	A	
9	$\overline{A}$	
10	B	
11	$\overline{B}$	X (1) Axis
12	Z	
13	$\overline{Z}$	
14	FG	

Nippon Molex	53258-1420	(14P) (Body side)
	51067-1400	Housing (14P)
	50217-8100	x 14 Terminal

(5) DC100W 2-Axis specifications



① Power Supply Connector

Pin No.	Signal
1	AC117V
3	AC117V
5	FG

Nippon Molex 53265-0320 (3P) (Body side)
51067-0500 Housing (5P)
50217-8100 x 3 Terminal

② LS Connector (Option)

Pin No.	Signal
1	P24V
2	N
3	XLS
4	YLS
5	EMG stop contact point input*

* The Emergency Stop is normally open.

Nippon Molex 53258-0520 (5P) (Body side)
51067-0500 Housing (5P)
50217-8100 x 5 Terminal

③ BK Connector (Option)

Pin No.	Signal
1	60V
2	GD
3	XBK
4	YBK

Nippon Molex 53258-0420 (4P) (Body side)
51067-0500 Housing (4P)
50217-8100 x 4 Terminal

④ M·PG Connector (Motor/Encoder Signal)

Pin No.	Signal	
1	MB	
2	MB	X (1) Axis
3	MA	
4	MA	
5	FG	
6	PV5	
7	GND	
8	A	
9	AGND	
10	B	
11	BGND	X (1) Axis
12	Z	
13	ZGND	
14	FG	

Nippon Molex 53258-1420 (14P) (Body side)
51067-1400 Housing (14P)
50217-8100 x 14 Terminal

⑤ M·PG Connector (Motor/Encoder Signal)

Pin No.	Signal	
1	MB	Y (2) Axis
2	NC	
3	NC	
4	MA	
5	FG	
6	PV5	
7	GD	
8	A	Y (2) Axis
9	AGND	
10	B	
11	BGND	
12	Z	
13	ZGND	
14	FG	

Nippon Molex	53258-1420	(14P) (Body side)
	51067-1400	Housing (14P)
	50217-8100	x 14 Terminal

⑥ SEL NET SND Connector (Option)

Pin No.	Signal
1	NC
2	NC
3	RD
4	TD
5	GD
6	FG

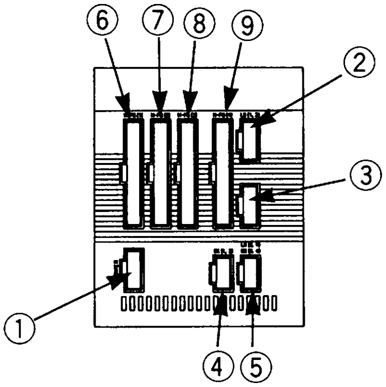
Nippon Molex	53259-0620	(6P) (Body side)
	51067-0600	Housing (6P)
	50217-8100	x 6 Terminal

⑦ SEL NET RCV Connector (Option)

Pin No.	Signal
1	NC
2	NC
3	RD
4	TD
5	GD
6	FG

Nippon Molex	53259-0620	(6P) (Body side)
	51067-0600	Housing (6P)
	50217-8100	x 6 Terminal

(6) DC100W 4-Axis specifications



① Power Supply Connector

Pin No.	Signal
1	AC117V
3	AC117V
5	FG

Nippon Molex 53265-0320 (3P) (Body side)
51067-0500 Housing (5P)
50217-8100 x 3 Terminal

② LS Connector (Option)

Pin No.	Signal
1	P24V
2	N
3	XLS
4	YLS
5	EMG stop contact point input*

* The Emergency Stop is normally open.

Nippon Molex 53258-0520 (5P) (Body side)
51067-0500 Housing (5P)
50217-8100 x 5 Terminal

③ LS Connector (Option)

Pin No.	Signal
1	P24V
2	N
3	ZLS
4	θLS
5	EMG stop contact point input*

* The Emergency Stop is normally open.

Nippon Molex 53258-0520 (5P) (Body side)
51067-0500 Housing (5P)
50217-8100 x 5 Terminal

④ BK Connector (Option)

Pin No.	Signal
1	60V
2	GD
3	XBK
4	YBK

Nippon Molex 53258-0420 (4P) (Body side)
51067-0400 Housing (4P)
50217-8100 x 4 Terminal

⑤　BK Connector (Option)

Pin No.	Signal
1	60V
2	GD
3	ZBK
4	θBK

Nippon Molex	53258-0420	(4P) (Body side)
	51067-0400	Housing (4P)
	50217-8100	x 4 Terminal

⑥　M·PG Connector (Motor/Encoder Signal)

Pin No.	Signal	
1	MB	
2	NC	X (1) Axis
3	NC	
4	MA	
5	FG	
6	PV5	
7	GD	
8	A	
9	AGND	
10	B	
11	BGND	X (1) Axis
12	Z	
13	ZGND	
14	FG	

Nippon Molex	53258-1420	(14P) (Body side)
	51067-1400	Housing (14P)
	50217-8100	x 14 Terminal

⑤　M·PG Connector (Motor/Encoder Signal)

Pin No.	Signal	
1	MB	
2	NC	Y (2) Axis
3	NC	
4	MA	
5	FG	
6	PV5	
7	GD	
8	A	
9	AGND	
10	B	
11	BGND	Y (2) Axis
12	Z	
13	ZGND	
14	FG	

Nippon Molex	53258-1420	(14P) (Body side)
	51067-1400	Housing (14P)
	50217-8100	x 14 Terminal

⑧ M·PG Connector (Motor/Encoder Signal)

Pin No.	Signal	
1	MB	
2	NC	Z (3) Axis
3	NC	
4	MA	
5	FG	
6	PV5	
7	GD	
8	A	
9	AGND	
10	B	
11	BGND	Z (3) Axis
12	Z	
13	ZGND	
14	FG	

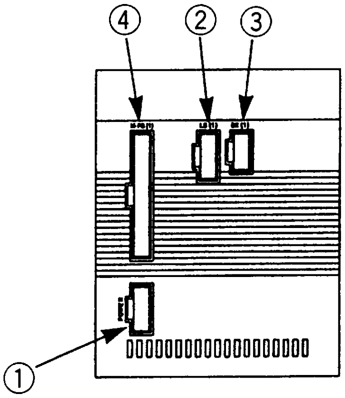
Nippon Molex	53258-1420	(14P) (Body side)
	51067-1400	Housing (14P)
	50217-8100	x 14 Terminal

⑨ M·PG Connector (Motor/Encoder Signal)

Pin No.	Signal	
1	MB	
2	NC	θ (4) Axis
3	NC	
4	MA	
5	FG	
6	PV5	
7	GD	
8	A	
9	AGND	
10	B	
11	BGND	θ (4) Axis
12	Z	
13	ZGND	
14	FG	

Nippon Molex	53258-1420	(14P) (Body side)
	51067-1400	Housing (14P)
	50217-8100	x 14 Terminal

(7) DC200W 2-Axis specifications



① Power Supply Connector

Pin No.	Signal
1	AC117V
3	AC117V
5	FG

Nippon Molex	53265-0320	(3P) (Body side)
	51067-0500	Housing (5P)
	50217-8100	x 3 Terminal

② LS Connector (Option)

Pin No.	Signal
1	P24V
2	N
3	XLS
4	YLS
5	EMG stop contact point input*

* The Emergency Stop is normally open.

Nippon Molex	53258-0520	(5P) (Body side)
	51067-0500	Housing (5P)
	50217-8100	x 5 Terminal

③ BK Connector (Option)

Pin No.	Signal
1	60V
2	GD
3	XBK
4	YBK

Nippon Molex	53258-0420	(4P) (Body side)
	51067-0400	Housing (4P)
	50217-8100	x 4 Terminal

④ M·PG Connector (Motor/Encoder Signal)

Pin No.	Signal	
1	MB	X (1) Axis
2	NC	
3	NC	
4	MA	
5	FG	
6	PV5	
7	GD	
8	A	X (1) Axis
9	AGND	
10	B	
11	BGND	
12	Z	
13	ZGND	
14	FG	

Nippon Molex	53258-1420	(14P) (Body side)
	51067-1400	Housing (14P)
	50217-8100	x 14 Terminal

⑤ M·PG Connector (Motor/Encoder Signal)

Pin No.	Signal	
1	MB	Y (2) Axis
2	NC	
3	NC	
4	MA	
5	FG	
6	PV5	
7	GD	
8	A	Y (2) Axis
9	AGND	
10	B	
11	BGND	
12	Z	
13	ZGND	
14	FG	

Nippon Molex	53258-1420	(14P) (Body side)
	51067-1400	Housing (14P)
	50217-8100	x 14 Terminal

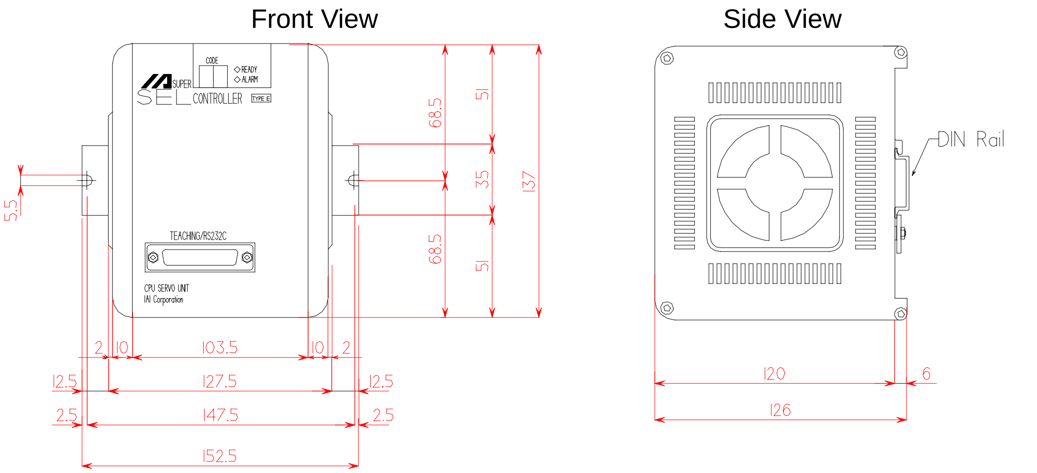
6. Controller Dimensions

Type E (Single Axis)

AC 60W • 100W
DC 20W ~ 100W

Type G (2 Axis)

AC 60W • 100W x 2
DC 20W ~ 100W x 2

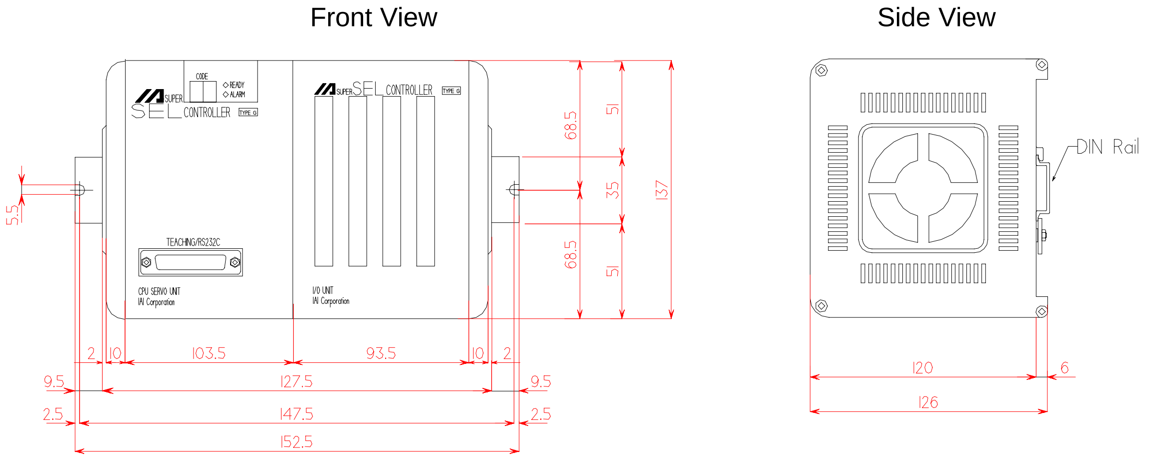


Type E (Single Axis) + Expansion Unit Box (Option)

AC 60W • 100W
DC 20W ~ 100W

Type G (2 Axis) + Expansion Unit Box (Option)

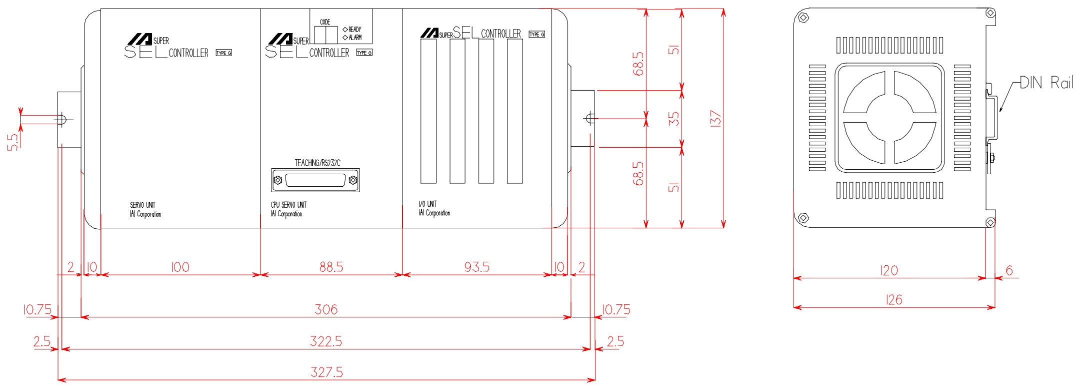
AC 60W • 100W x 2
DC 20W ~ 100W x 2



Type E (Single Axis)
AC 200W
AC 400W
DC 200W

Type G (2 Axis)
AC 200W x 2
DC 200W x 2

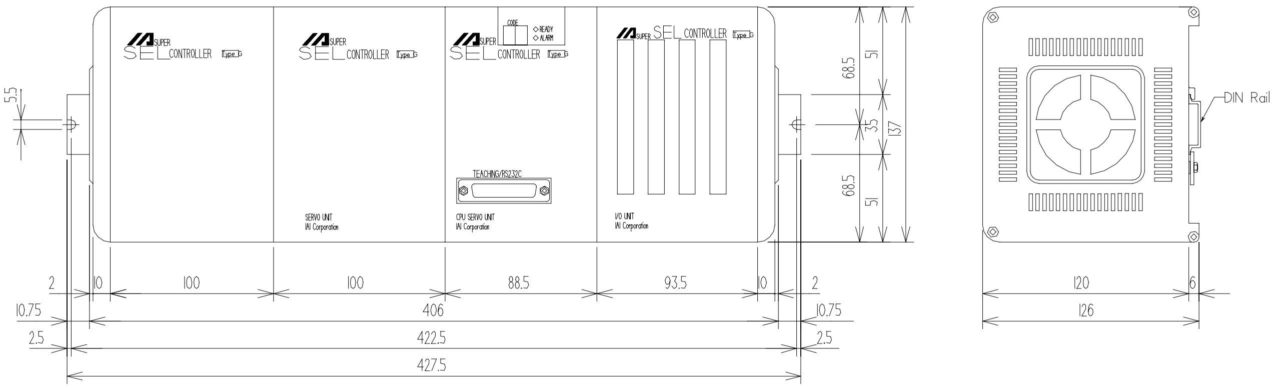
Type G (4 Axis)
AC 60W • 100W x 4
DC 20W ~ 100W x 4


Front View

Side View

Type G (2 Axis)
AC 400W

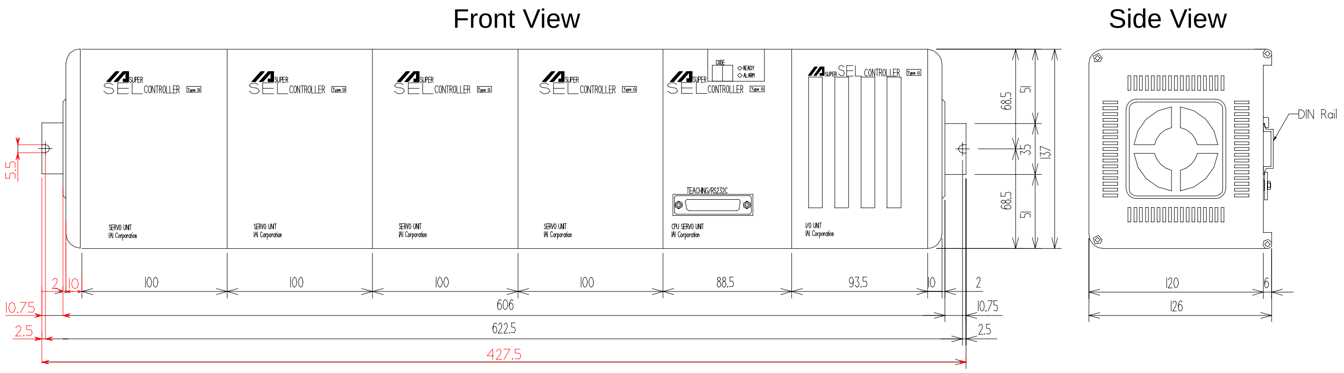
Type G (2 Axis)
AC 200W x 4
DC 200W x 4


Front View

Side View

Type G (4 Axis)

AC 400W x 4

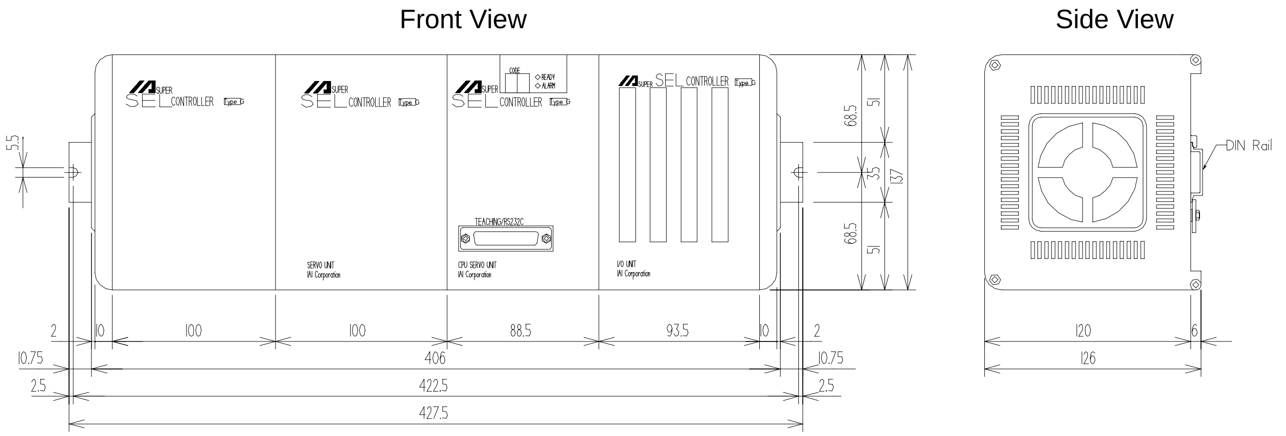

Front View

Side View

Type G (8 Axis)

AC 60W • 100W x 4

DC 20W ~ 100W x

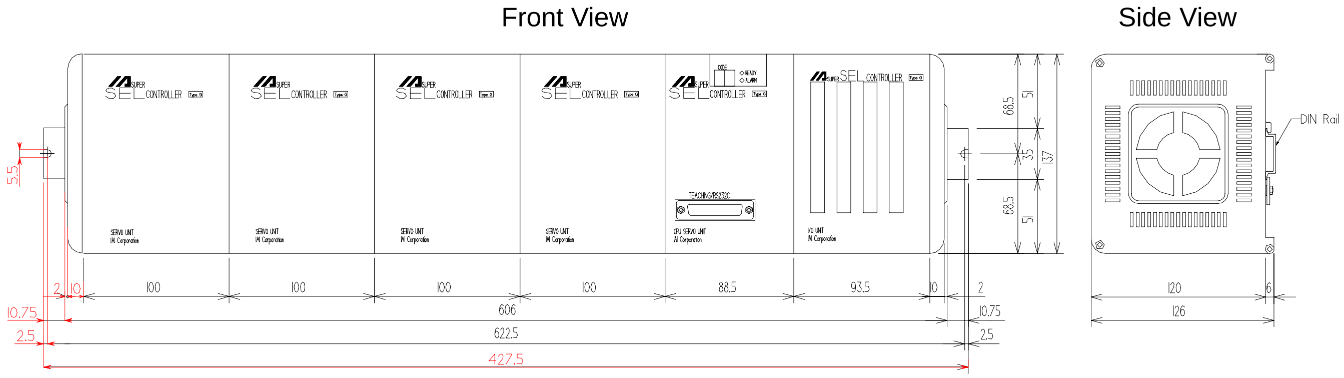

Front View

Side View

Type G (8 Axis)

AC 200W x 8

DC 200W x 8

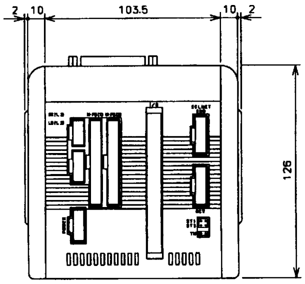

Front View

Side View

7. Unit Configurations

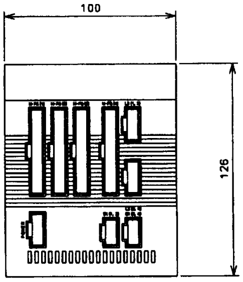
1 Servo Driver Units: Basic Modules

Module 1 - Both CPU and servo driver in a single unit. Representative of 1 or 2-axis Type-E and Type-G controllers with 100W maximum motor capacity per axis.



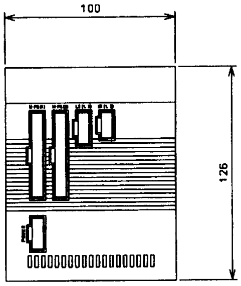
100W 2-axis dedicated

Module 2 - 4-axis servo driver unit with 100W maximum motor capacity per axis. This servo driver unit must be used along with a separate CPU unit. (See examples)



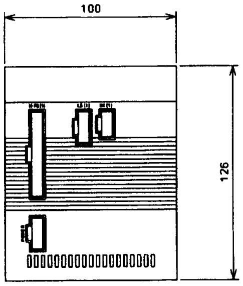
Servo driver unit
(100W 4-axis)

Module 3 - 2-Axis servo driver unit with 200W maximum motor capacity per axis. This servo driver unit must be used along with a separate CPU unit. (See examples)



Servo driver unit
(200W 2-axis)

Module 4 - 1-Axis servo driver unit with 400W motor capacity. This servo driver unit must be used along with a separate CPU unit. (See examples)

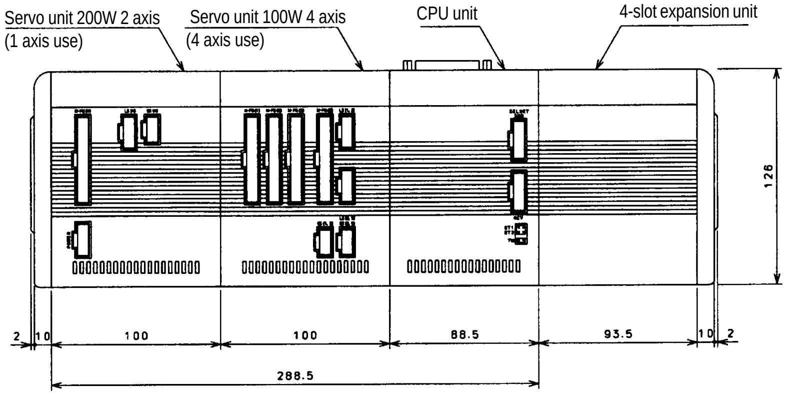


Servo driver unit
(400W 1-axis)

2 5-Axis configuration with various motor capacities

CPU Unit
+
Axis 1 =60W
Axis 2 =100W → 100W 4-axis servo driver unit
Axis 3 =100W
+
Axis 4 =200W → 200W 2-axis servo driver unit (using 1 axis only)
Axis 5 =60W

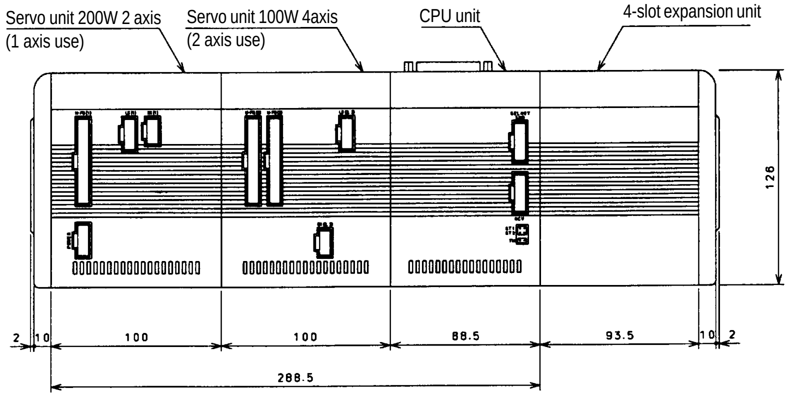
* The unit with the largest motor capacity is placed at the furthest left of the CPU unit. This is to reduce noise interference. A 12-slot expansion unit is available as an option.



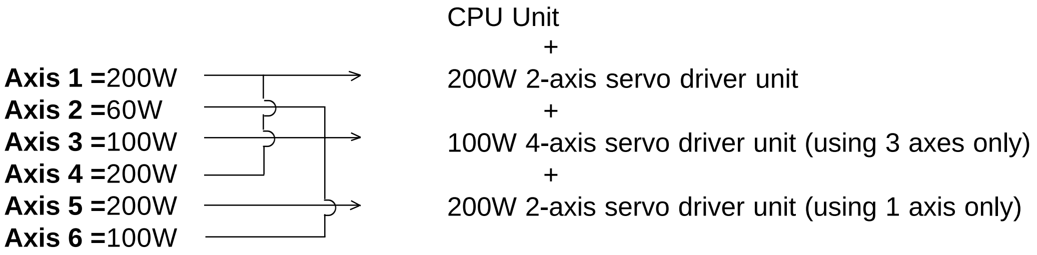
3 3-Axis configuration with various motor capacities

CPU Unit
+
Axis 1 =200W → 200W 2-axis servo driver unit (using 1 axis only)
+
Axis 2 =60W
Axis 3 =60W → 100W 4-axis servo driver unit (using 2 axes only)

* The unit with the largest motor capacity is placed at the furthest left of the CPU unit. This is to reduce noise interference.



4 6-Axis configuration with various motor capacities



CPU Unit

+

Axis 1 =200W

Axis 2 =60W

Axis 3 =100W

Axis 4 =200W

Axis 5 =200W

Axis 6 =100W

200W 2-axis servo driver unit

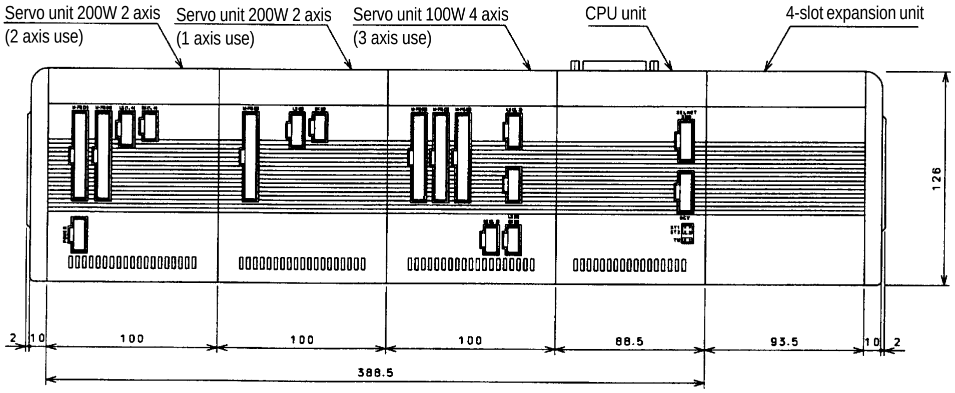
+

100W 4-axis servo driver unit (using 3 axes only)

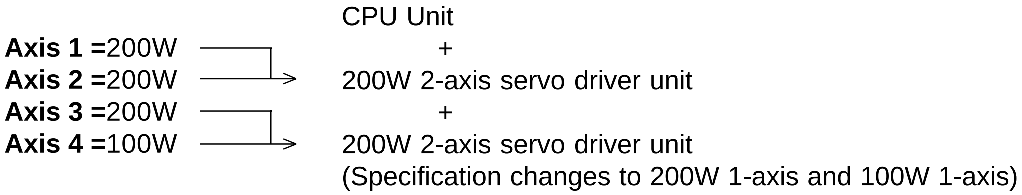
+

200W 2-axis servo driver unit (using 1 axis only)

* The unit with the largest motor capacity is placed at the furthest left of the CPU unit. This is to reduce noise interference. A 12-slot expansion unit is available as an option.



5 4-Axis configuration with various motor capacities



CPU Unit

+

Axis 1 =200W

Axis 2 =200W

Axis 3 =200W

Axis 4 =100W

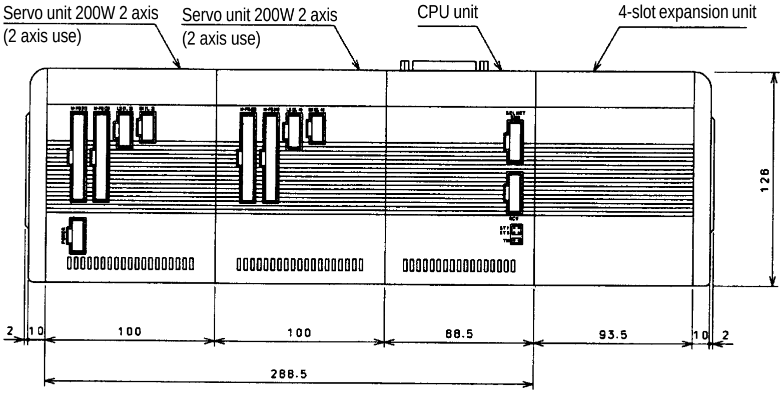
200W 2-axis servo driver unit

+

200W 2-axis servo driver unit

(Specification changes to 200W 1-axis and 100W 1-axis)

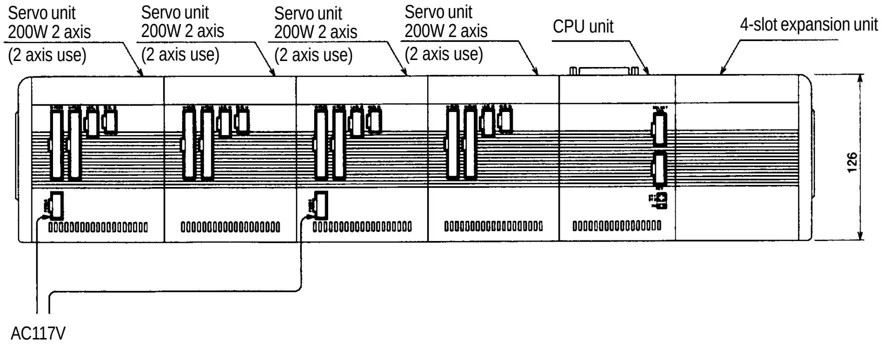
* The unit with the largest motor capacity is placed at the furthest left of the CPU unit. This is to reduce noise interference.



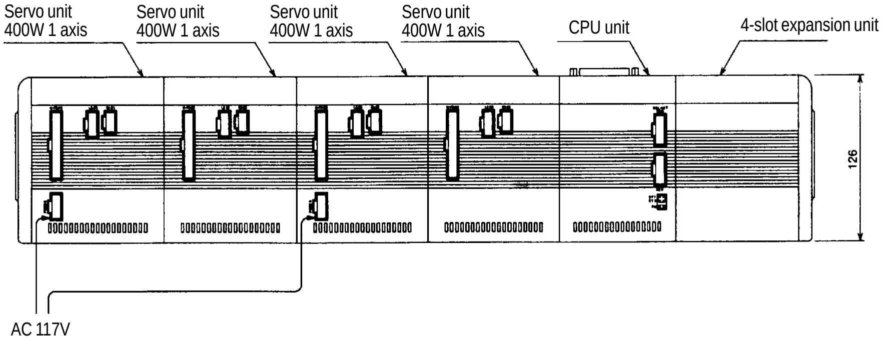
6 Controller configuration limitations

(1) Limitation on number of axes
Motor Capacity 100W– – – – (Maximum 8 axes)
Motor Capacity 200W– – – – (Maximum 8 axes)*
Motor Capacity 400W– – – – (Maximum 4 axes)*
* In the case of 8 axes with 200W motor or 4 axes with 400W motor, there are two power supply locations so power must be supplied to both.

200W 8 Axis Specification



400W 4 Axis Specification



(2) Limitation on axis expansion for dedicated 2-axis unit
The dedicated 2-axis Super SEL controller unit cannot be expanded to handle more than 2 axes.

(3) Limitation on unit configuration
Up to a maximum for four servo driver units can be connected.

[Example 1]
400W x 3 axes – – – – – – – – 3 units
+ = 4 units → ok
200W x 1 axis – – – – – – – – 1 unit

[Example 2]
400W x 3 axes – – – – – – – – 3 units
+ = 5 units → no good
200W x 3 axes – – – – – – – – 2 units

8. Type G (AC) Unit Configurations

Once the axis alignment and motor wattage (60W, 100W, 200W, 400W) have been decided, the controller unit configuration can be determined. We will explain the tables given below for determining the unit configuration from the model type.

1. Check how many axes there are by motor output. In this case, any motor output below 100W will be considered as 100W.

Example 200W - 100W - 60W

⇩

200W x 1
100W x 2 ←60W is less than 100W so it is considered as 100W

2. Looking at the front of the unit, arrange the units with the largest output starting at the left, refer to charts 1 ~ 7 on the following pages and confirm the configuration type.

EXAMPLE 1
2-axis system

200W-400W Assembly

Model type is M-SEL-G-2-AC-200-400 · · · (1)

↓

200W x 1
400W x 1
Arrange from the largest output,
400 + 200
According to Table 1, the configuration type is C

Note: Because the unit is arranged in order of largest output, please be aware that the order in which the axes are arranged may differ from what was specified at the time the unit was ordered (in actual usage, there will be no changes to the specifications designated by the customer). However, for output under 100W, the units will be arranged in axis order regardless of the wattage.

EXAMPLE 2
4-axis system

60W-200W-100W-60W Assembly

Model type is M-SEL-G-4-AC-60-200-100-60

↓

100W x 3
200W x 1
Arrange from the largest output,
(200 x 1) + (100 x 3)
According to Table 3, the configuration type is C

EXAMPLE 3
6-axis system

400W-200W-400W-200W-100W-60W Assembly

Model type is M-SEL-G-6-AC-400-200-400-200-100-60

↓

400W x 2
200W x 2
100W x 2
Arrange from the largest output,
(400 x 2) + (200 x 2) + (100 x 2)
According to Table 5, the configuration type is H

*There are other limiting factors on the assembly types. When the total output is 1600W or less, up to a maximum of 4 servo driver units can be connected. For an assembly consisting entirely of 400W output, only up to four axes can be assembled. Configurations where the above restrictions apply, are excluded from the tables that follow.

[Unit Configuration Tables]
*Any output less than 100W is considered as 100W.

Table 1

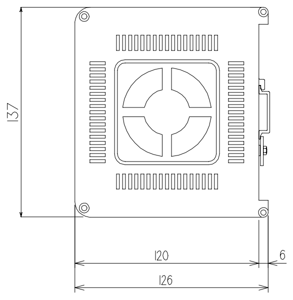
2-axis (6 combinations)	Configuration
100-100	A
200-200	B
200-100	B
400-400	C
400-200	C
400-100	C

Table 2

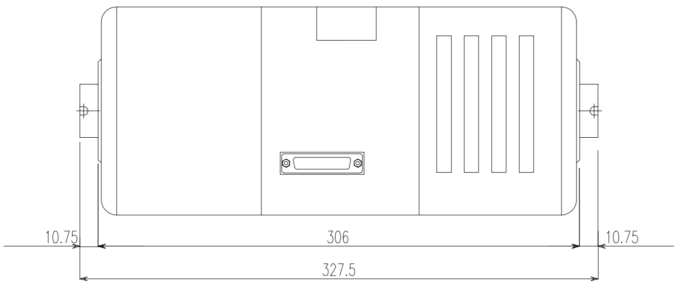
3-axis (10 combinations)	Configuration
100-100-100	B
200-200-200	C
200-200-100	C
200-100-100	C
400-400-400	D
400-400-200	D
400-400-100	D
400-200-200	C
400-200-100	C
400-100-100	C

Dimension Drawings

(Side view)

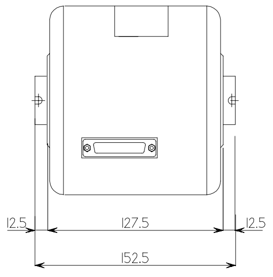


B



A

Front view



C

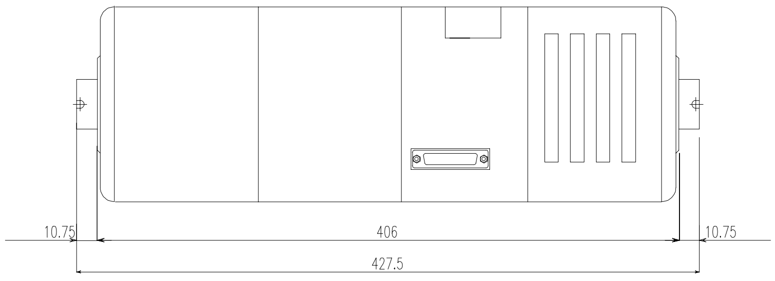
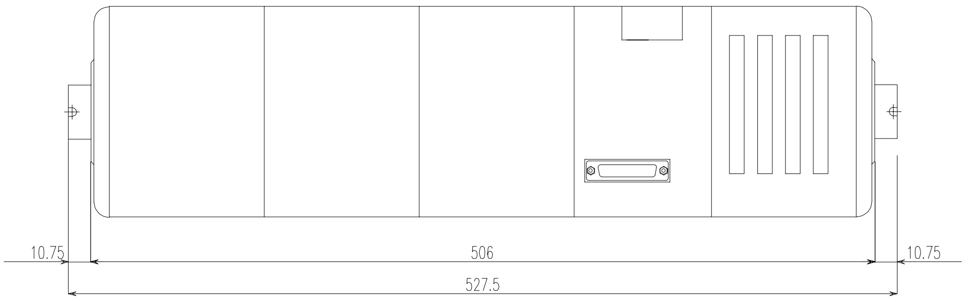


Table 3

4-axis (15 combinations)	Configuration
100x4	B
200x4	C
200x3 + 100x1	C
200x2 + 100x2	C
200x1 + 100x3	C
400x4	E
400x3 + 200x1	E
400x3 + 100x1	E
400x2 + 200x2	D
400x2 + 200x1 + 100x1	D
400x2 + 100x2	D
400x1 + 200x3	D
400x1 + 200x2 + 100x1	D
400x1 + 200x1 + 100x2	D
400x1 + 100x3	C

D



E

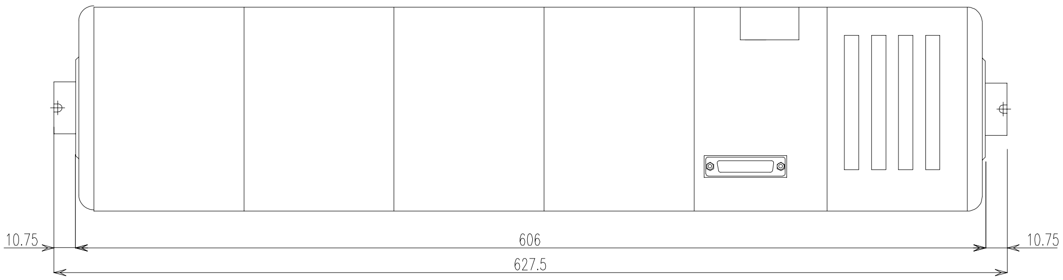


Table 4

5-axis (18 combinations)	Configuration
100x5	F
200x5	G
200x4 + 100x1	G
200x3 + 100x2	G
200x3 + 100x3	F
200x1 + 100x4	F
400x3 + 200x2	H
400x3 + 200x1 +100X1	H
400x3 + 100x2	H
400x2 + 200x3	H
400x2 + 200x2 + 100x1	H
400x2 + 200x1 + 100x2	H
400x2 + 100x3	G
400x1 + 200X4	G
400x1 + 200x3 + 100x1	G
400x1 + 200x2 + 100x2	G
400x1 + 200x1 + 100x3	G
400x1 + 100x4	F

Table 5

6-axis (19 combinations)	Configuration
100x6	F
200x6	G
200x5 + 100x1	G
200x4 + 100x2	G
200x3 + 100x3	G
200x2 + 100x4	F
200x1 + 100x5	F
400x3 + 100x3	H
400x2 + 200x4	H
400x2 + 200x3 + 100x1	H
400x2 + 200x2 + 100X2	H
400x2 + 200x1 + 100X3	H
400x2 + 100x4	G
400x1 + 200x5	H
400x1 + 200x4 + 100x1	H
400x1 + 200x3 + 100x2	H
400X1 + 200x2 + 100x3	G
400x1 + 200x1 + 100x4	G
400x1 + 100x5	G

Table 6

7-axis (19 combinations)	Configuration
100x7	F
200x7	H
200x6 + 100x1	H
200x5 + 100x2	H
200x4 + 100x3	G
200x3 + 100x4	G
200x2 + 100x5	G
200x1 + 100x6	G
400x3 + 100x4	H
400x2 + 200x2 + 100X3	H
400x2 + 200x1 + 100x4	H
400x2 + 100x5	H
400x1 + 200x6	H
400x1 + 200x5 + 100x1	H
400x1 + 200x4 + 100x2	H
400x1 + 200x3 + 100x3	H
400X1 + 200x2 + 100x4	G
400x1 + 200x1 + 100x5	G
400x1 + 100x6	G

Table 7

8-axis (17 combinations)	Configuration
100x8	F
200x8	H
200x7 + 100x1	H
200x6 + 100x2	H
200x5 + 100x3	H
200x4 + 100x4	G
200x3 + 100x5	G
200x2 + 100x6	G
200x1 + 100x7	G
400x2 + 200x2 + 100X4	H
400x2 + 200x1 + 100X5	H
400x2 + 100x6	H
400x1 + 200x4 + 100x3	H
400x1 + 200x3 + 100x4	H
400X1 + 200x2 + 100x5	H
400x1 + 200x1 + 100x6	H
400x1 + 100x7	G

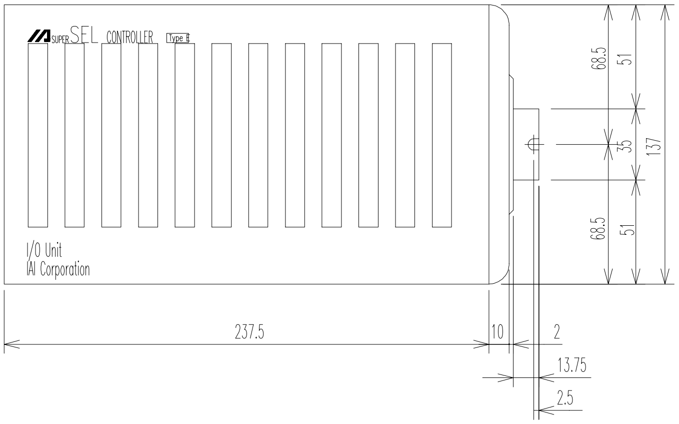
9. 12-slot Expansion Unit

A 12-slot expansion unit is available as an option and is used in the following situations.

(1) When the total number of inputs and outputs exceeds 96.
(2) When the total number of inputs and outputs is 96 and a flash memory card is used.
(3) When the total number of inputs and outputs is 96 and a 2-channel RS232 unit is used.

Please refer to pages 58 and 59 for the servo unit and CPU unit assembly configuration.

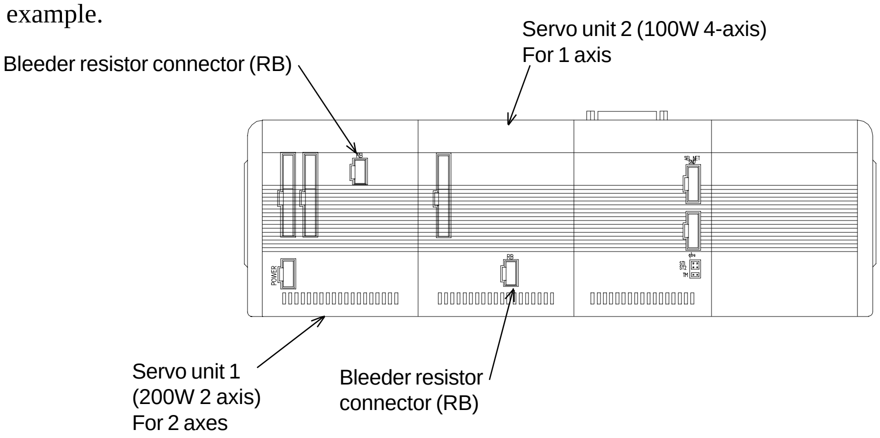
The drawing below gives the external dimensions for the 12-slot expansion unit.



10. Bleeder Resistor

If the selected module uses a large amount of voltage, there will also be a large amount of regenerative voltage which in some cases may require you to attach a bleeder resistor. The determination of whether to attach a bleeder resistor is made for each servo unit separately and is dependent on the total wattage of all motors for that unit. However, if the servo unit has a brake specification, there is a brake box (option) with a built-in discharge circuit.

Below, we illustrate the connection of a bleeder resistor to the controller using an M-SEL-G-3-AC-200-200-100 module as an example.

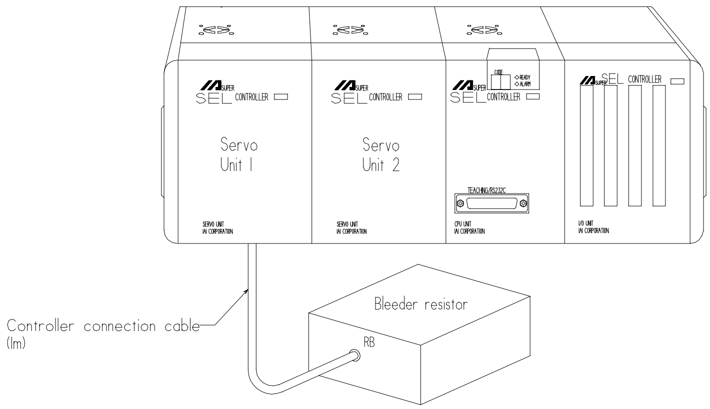


In the above module, servo unit 1 has a large combined motor capacity, which may require a discharge circuit so a bleeder resistor is attached. In this case, the cable for connecting the resistor to the controller is plugged into the resistor connector RB. Servo unit 2 does not require a resistor since the motor capacity is small. The resistor connector (RB) becomes a reserve connector.

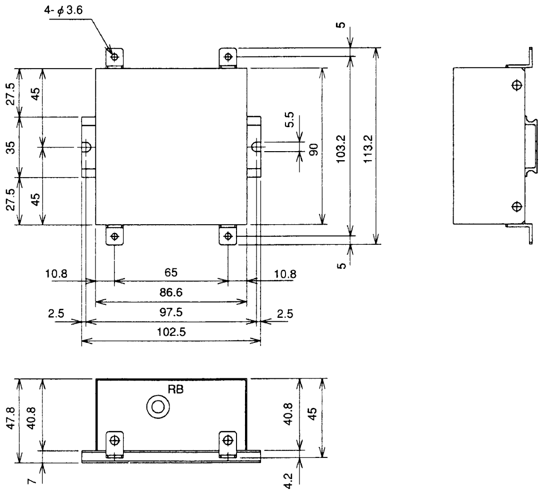
* At present, the general rule is that a bleeder resistor is attached for each servo unit in which the total motor capacity of all the axes is 200W or greater for an AC motor and 300W or greater for a DC motor. Also, if you are using the actuator as a vertical axis, the optional bleeder resistor is necessary even if you are not using a brake (box).

Note: If you forget to attach a bleeder resistor when the actuator is carrying a vertical load, excessive regenerative voltage will cause an "A1" driver alarm. If this happens, turn the power OFF, then turn the power back ON to home the actuator.

M-SEL-G-3-AC-200·200·100



Dimensions



Part 8 Super SEL Controller Maintenance

To ensure safe and trouble-free operation of your Super SEL system, a regular maintenance and inspection program should be implemented. Be sure to turn OFF the power before initiating any maintenance or inspection work. An inspection is recommended at least once every 6 to 12 months. However, depending on the environment, a more frequent inspection schedule may be advisable.

(1) Inspection Guidelines

- Check and make sure that the power supply to your Super SEL Controller is maintained at 90 ~ 127V.
- Check the controller vents and clean any accumulated dirt or dust.
- Check the controller cable (controller → axis) and make sure there are no loose screws or disconnections.
- Check for loose controller mounting screws. Tighten if necessary.
- Check each cable (axis cables, general I/O cables, system I/O cables, power supply cable). Check for loose connections, damage, or excessive wear. Replace if necessary.

(2) Recommended Spare Parts

It is advisable to keep a small supply of spare parts. In critical applications, it may be advisable to keep an entire spare Super SEL Controller on hand. The following spare parts are recommended:

- Cables
- Batteries* (Ni-Cd batteries have a general shelf life of about 6 years but this varies depending on use conditions and environment)

(3) Memory Backup

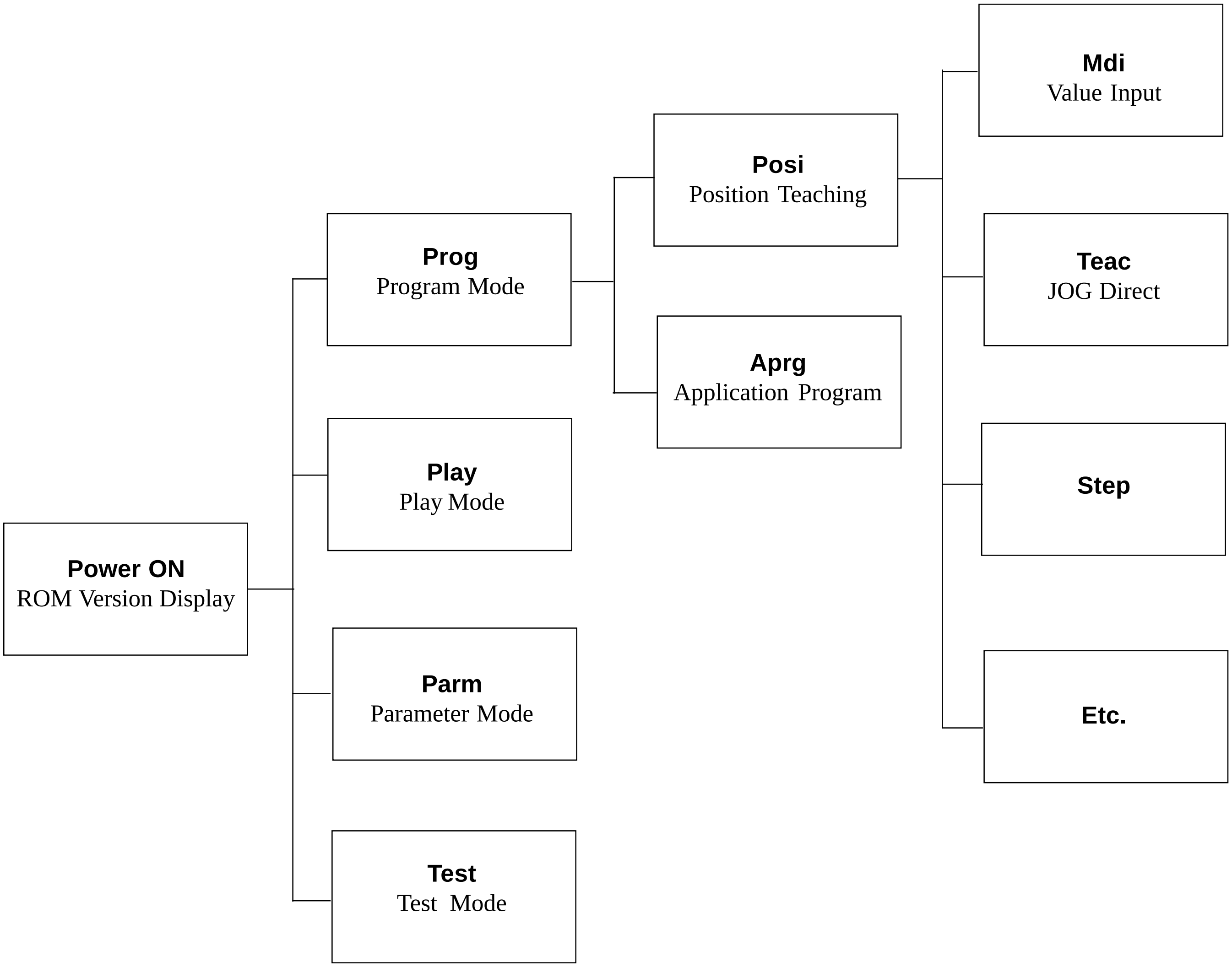
When the controller is fully charged, backup memory is guaranteed for 3 months. In actuality, the backup memory is not erased for 6-8 months but if the controller is to be left for a long period (more than 3 months) without having current run through it, please take precautions to save your program, position data, and parameters. To fully charge the controller if it does not contain any data, you will need to leave the controller with the power ON for 3 days.
If the memory is erased, the system's preset parameters will be set but the actuator will not run properly in this condition.

Chapter 2. Operation

Part 1 Basics in Operating Your Super SEL Controller

1. Summary of Teaching Pendant Operation

The tree structure below illustrates the teaching pendant mode structure.



The basics of Teaching Pendant operation of your Super SEL Controller will display the above messages on the LCD display screen. Press F1 ~ F4 to move through the branch modes and press ESC to go back to the main branch.

2. Teaching Pendant Key Functions

Teaching pendant key functions are as follows:

01 **Cursor:** Numbers can be changed when the cursor is positioned underneath.

"**Inc**" **Increment:** This key increases the step number or point number.

"**Dec**" **Decrement:** This key decreases the step number or point number.

"**Esc**" **Escape:** This key is used to go back to the previous display.

" **.** " **Decimal Point Key:** When creating an application program, the display will go to the next command menu.

"-" **Minus Key:** When creating an application program, the display will return to the previous command menu.

" ◄ ► " **Data Key, JOG Key:** Dual function key for use in data input and axis Jog functions.

" ↵ " **Return Key:** This key registers the value at the cursor, and makes it effective.

Note: If the main power is turned OFF, all programs and data will be stored for a guaranteed period of 3 months.

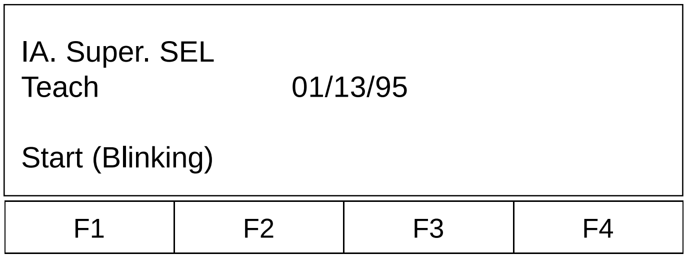
Chapter 2. Operation

Part 2 Teaching Pendant Operation

1. Mode Selection

(LCD Screen Display)

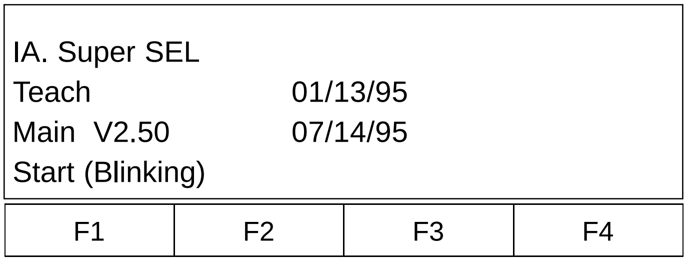
(Operation)



```
IA. Super. SEL
Teach                01/13/95

Start (Blinking)
```

1. Initial Display

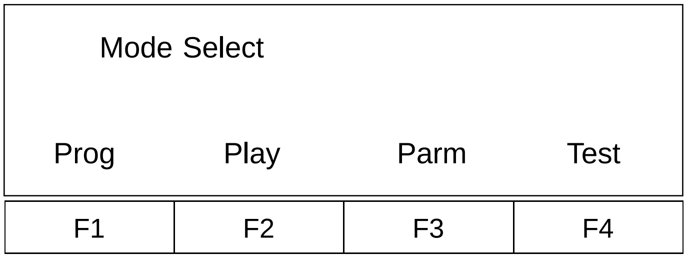
   Press F1 (Start) (To P66. 2)



```
IA. Super SEL
Teach                 01/13/95
Main  V2.50           07/14/95
Start (Blinking)
```

2. Controller ROM Version Display

   Press F1 (Start) (To P66. 3)



```
      Mode Select


   Prog        Play        Parm        Test
```

3. Mode Selection Display

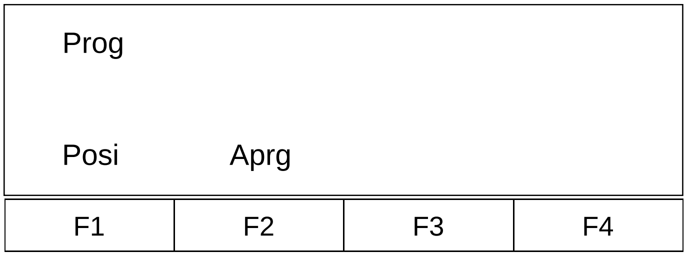
   F1- Prog (Program) Mode (P67. 4)
   F2- Play (Play/Execution) Mode (P81. 22)
   F3- Parm (Parameter) Mode (P75. 18)
   F4- Test (Test) Mode (P79. 21)

Note In case of an input error, press the **ESC** key to return to the previous screen and then resume operation. For any operation, if you continue pressing the **ESC** key, you will eventually return to this initial display.

2. Program Mode

(LCD Screen Display)

(Operation)



Prog

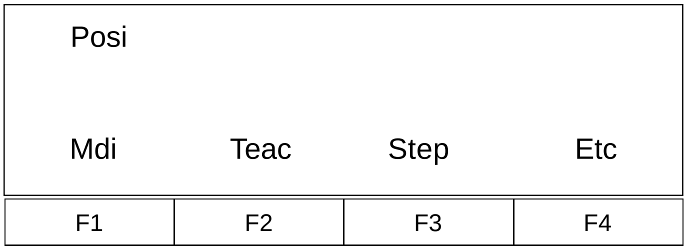
Posi Aprg

F1	F2	F3	F4

4. Program Mode (Press F1 at Step 3)

F1- (Posi: Position) Position Data Input (P67. 5)
F2- (Aprg: Application Program) Super SEL Programming Edit Display (P72. 15)
Esc- Return to Mode Selection Display (P66.3)

① Creating position data



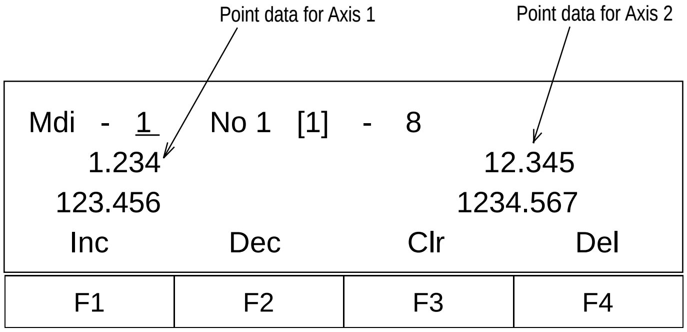
Posi

Mdi Teac Step Etc

F1	F2	F3	F4

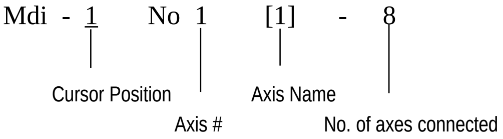
5. Position (Position Data) Input Display (Step 4. Press F1)

F1- Mdi Mode (Position Data Direct Input) (P67. 6)
F2- Teac (Position Data Teaching) Mode (P70. 12)
F3- Point Step (Position Data Step) Mode (P69. 10)
F4- Return to Etc Edit Display 2 (P68. 6-2)
Esc- Return to Program Mode (P67. 4)



```
Mdi  -  1    No 1  [1]  -  8
   1.234                    12.345
 123.456                  1234.567
 Inc       Dec       Clr       Del
```

F1	F2	F3	F4

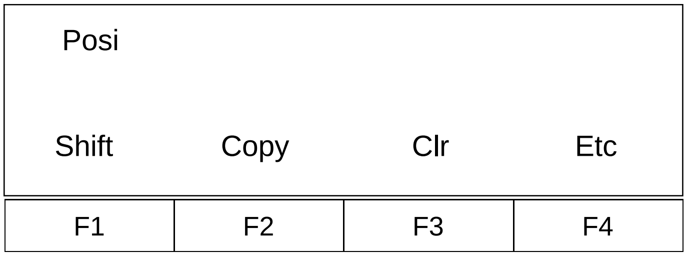


*This screen can only display up to a maximum of four axes.

6. Mdi Mode

6-1 Position Number Input Mode (Step 5. Press F1)

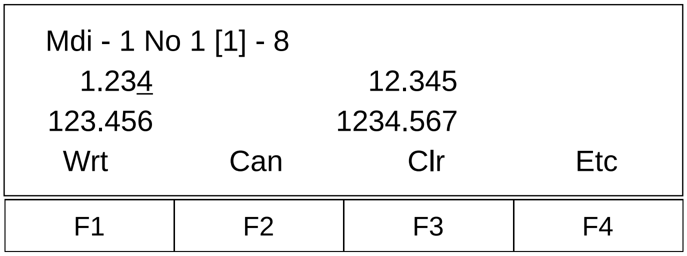
Input 4-digit numbers.
F1- (Inc: Increment) Increments the position number displayed by 1.
F2- (Dec: Decrement) Decrements the position number displayed by 1.
F3- (Clr: Clear) Clears all data (return to 0) for reentry.
F4- (Del: Delete) Deletes input data
Return - Position Data Input (P68. 7)
Esc- Returns to Position Data Input Display (P67. 5)



```
Posi

Shift      Copy      Clr      Etc
F1         F2        F3       F4
```

6-2 Position (Position Data) Input Display 2 (Go back to the display in Step 5. Press F4)

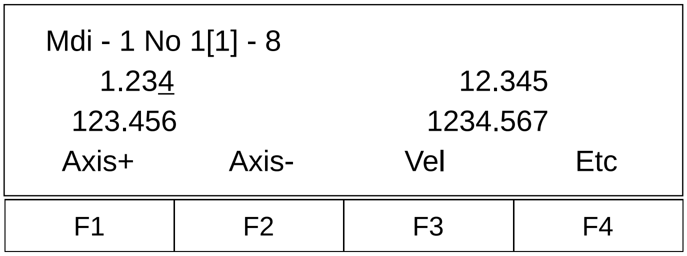
F1- (Shift) Position Shift Mode (P69. 11-1)
F2- (Copy) Position Data Copy (P69. 11-1)
F3- (Clr: Clear) Position Data Clear (Deletion) (P69. 11-2)
F4- Position (Position Data) Input Display (P67. 5)
Esc- Return to Program Mode (P67. 4)



```
Mdi - 1 No 1 [1] - 8
   1.234            12.345
 123.456          1234.567
 Wrt       Can      Clr      Etc
 F1        F2       F3       F4
```

7. Position Data Input Mode (Step 6-1, Return Key)

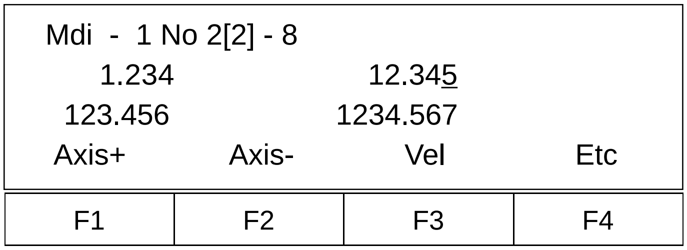
Direct input (5-digit integers before decimal point and 3 digits after the decimal point.)
F1- (Wrt: Write) Saving position data
F2- (Can: Cancel) displays xxxx.xxx.
F3- (Clr: Clear) displays 0.000 for data reentry.
F4- Position data input sub menu. (P68. 8)



```
Mdi - 1 No 1[1] - 8
    1.234                12.345
  123.456              1234.567
 Axis+     Axis-     Vel      Etc
 F1        F2        F3       F4
```

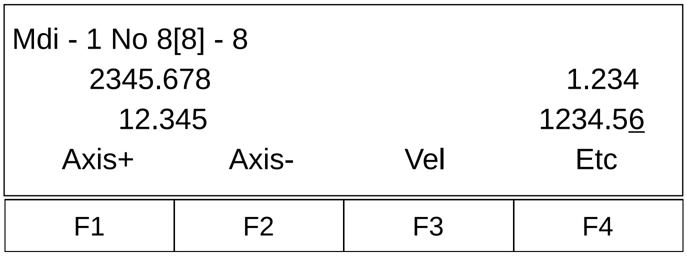
8. Position Data Input Sub Menu (Step 7. F4)

Direct Input (5-digit integers before decimal point and 3-digits after the decimal point)
F1- (Axis+) Increments axis number by 1. (P68. 8-1)
F2- (Axis-) Decrement axis number by 1. (P68. 8-2)
F3- (Vel: Velocity) Set velocity and acceleration for the position number. (P69. 9)
F4- (Etc) Returns to Position Data Input Mode (P68. 7)



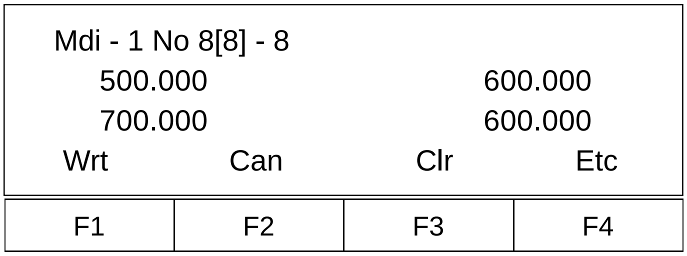
```
Mdi  -  1 No 2[2] - 8
    1.234           12.345
  123.456         1234.567
 Axis+     Axis-     Vel      Etc
 F1        F2        F3       F4
```

8-1 Increments axis number by 1. (Step 8, F1)



```
Mdi - 1 No 8[8] - 8
    2345.678                  1.234
      12.345               1234.56
 Axis+     Axis-     Vel      Etc
 F1        F2        F3       F4
```

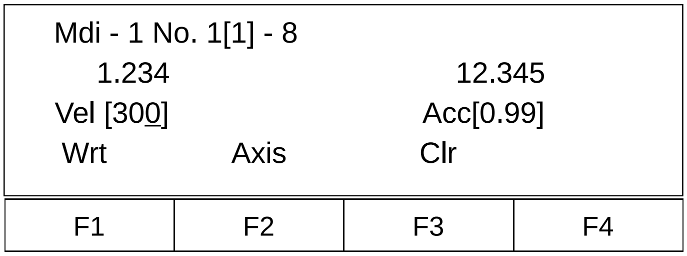
8-2. Decrements axis number by 1. (Step 8, F2)



```
Mdi - 1 No 8[8] - 8
   500.000              600.000
   700.000              600.000
 Wrt      Can       Clr        Etc
 F1       F2        F3         F4
```

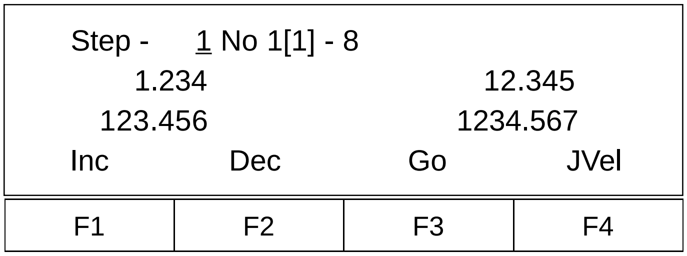
8-3 Use the F1 key (Wrt) to save data for up to 8 axes. However, do not write data while a program is running (cannot guarantee data will be saved).

F1- Wrt: Write
F2 - Can: Cancel
F3- Clr: Clear



```
Mdi - 1 No. 1[1] - 8
   1.234              12.345
Vel [300]            Acc[0.99]
 Wrt      Axis      Clr
 F1       F2        F3         F4
```

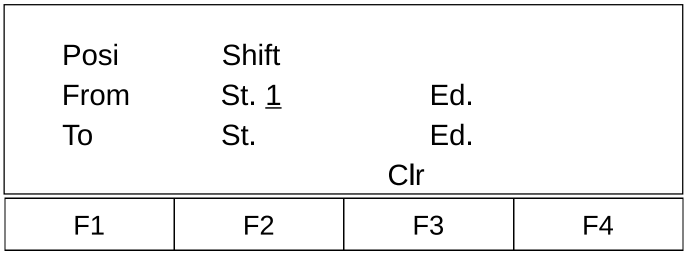
9. Position Data Velocity and Acceleration Setting (Step 8. F3) (Velocity: 3-digits, Acceleration 9.99, 2-digits after decimal point.)



```
 Step -   1 No 1[1] - 8
     1.234              12.345
   123.456            1234.567
 Inc      Dec       Go         JVel
 F1       F2        F3         F4
```

10. Point Step Mode (Go back to the display in step number 5, then press F3.)

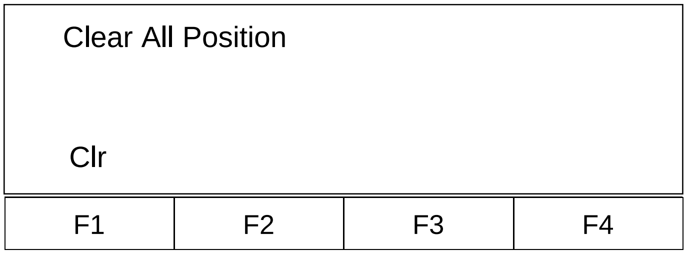
F1- Position No. +1
F2- Position No. -1
F3- Move to designated position
F4- Designate velocity



```
 Posi      Shift
 From      St. 1          Ed.
 To        St.            Ed.
                       Clr
 F1       F2        F3         F4
```

11-1 Position Shift (Move) Mode (6-2. F1 key)

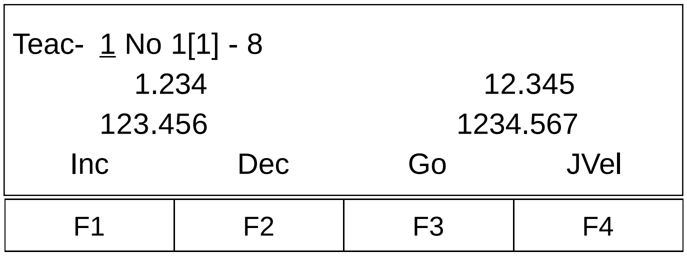
Moves the specified multiple position data. Input the beginning and the ending position No. of the original position (From), then input the beginning and the ending position No. of the target position (To).
*Position Copy Mode (6-2. F2 key) for duplication can be operated in the same manner.



```
 Clear All Position


 Clr
 F1       F2        F3         F4
```

11-2 Position Clear Mode (Go back to the display in Step 6-2. Then, press F3.)

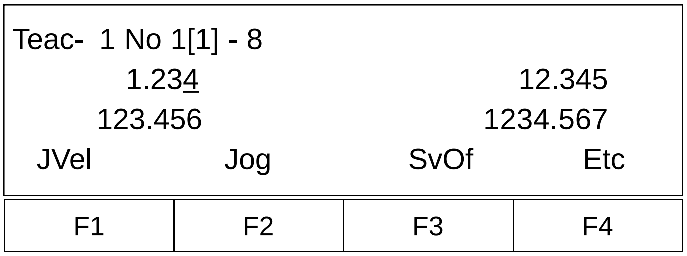
F3- Clears all position numbers.
F1- Execute



```
Teac-  1 No 1[1] - 8
       1.234                12.345
     123.456              1234.567
   Inc        Dec        Go        JVel
|   F1    |   F2    |   F3    |   F4   |
```

12. Teaching Mode (Go back to the display in Step 5, then press F2.

    Position Number Input Mode (2-digit number input)
    F1- Increments position number by 1.
    F2- Decrements position number by 1.
    F3- Moves actuators from current position to the position designated in the display.
    F4- Sets velocity.
    Return- Performs teaching for each axis. (P70. 13)

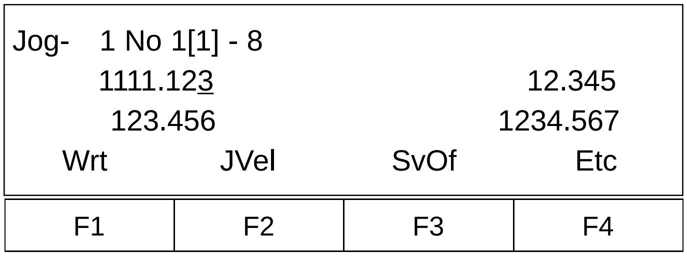


```
Teac-  1 No 1[1] - 8
       1.234                  12.345
     123.456                1234.567
 JVel         Jog          SvOf        Etc
|   F1    |   F2    |   F3    |   F4   |
```

13. Teaching Selection Mode (12. Return Key)

    Select Jog or SvOf (Servo OFF, Manual)
    The cursor is located under axis number 1. Press Etc to switch to another axis.

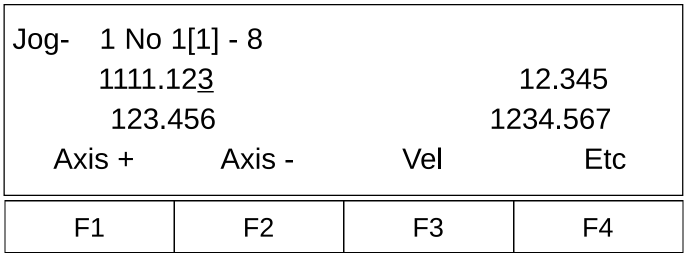
    F1- Select jog velocity.
    F2- Set for jog movement. (P70. 13-1)
    F3- Set for manual (direct) teaching. (P71. 14)
    F4- Velocity setting and axis changing mode. (P70. 13-3)



```
Jog-   1 No 1[1] - 8
     1111.123                 12.345
      123.456               1234.567
  Wrt        JVel        SvOf        Etc
|   F1    |   F2    |   F3    |   F4   |
```

13-1 Jog Mode (Step 13. F2)

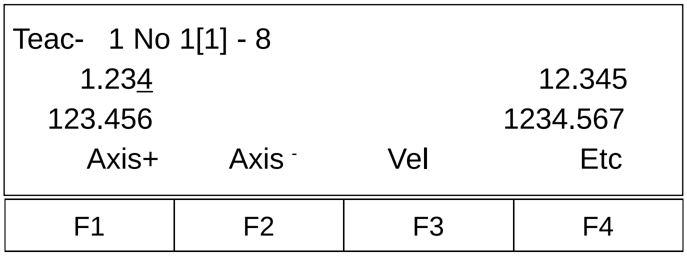
Turns all servos ON and also indicates the present position.
When Wrt is selected, the present position will be saved.
F4- Velocity setting and axis selection mode (P70. 13-2)



```
Jog-   1 No 1[1] - 8
     1111.123                12.345
      123.456              1234.567
  Axis +      Axis -       Vel        Etc
|   F1    |   F2    |   F3    |   F4   |
```

13-2 Jog Mode (Step 13-1, F4)
Velocity, axis selection.

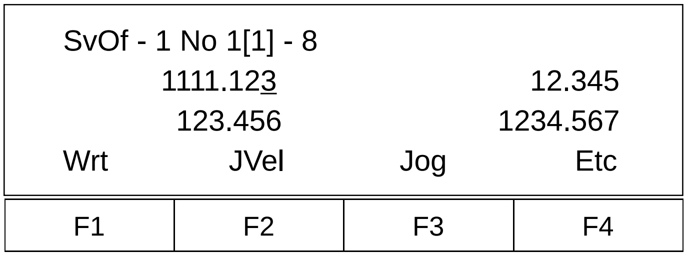
When selecting Axes to jog:
F1- Axis+ moves cursor forward 1 axis.
F2- Axis- moves cursor backward 1 axis.



```
Teac-   1 No 1[1] - 8
     1.234                      12.345
   123.456                    1234.567
     Axis+      Axis -      Vel        Etc
|   F1    |   F2    |   F3    |   F4   |
```

13-3 Teaching Velocity Set, Axis Change Mode (Step 13. F4)

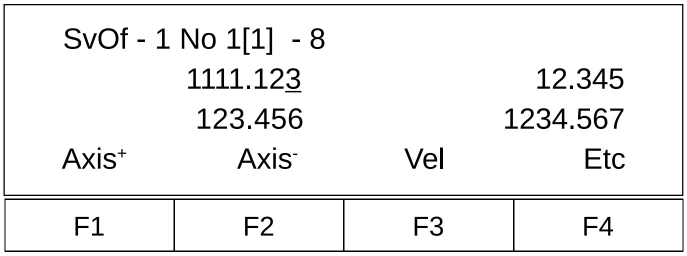
F1- Teach axis number + 1.
F2- Teach axis number - 1.
F3- Set velocity and acceleration for point data.
F4- Return to the display before selection.



```
SvOf - 1 No 1[1] - 8
      1111.123            12.345
       123.456          1234.567
Wrt        JVel       Jog       Etc
F1         F2         F3        F4
```

14. SvOf (Servo OFF, Manual • Direct Teaching) Mode (Go back to the display in Step 13, then press F3)

Servo OFF for all connected axes.
Moves axis position manually (direct teaching).
Wrt: Saves data.
Return: Changes axis number.

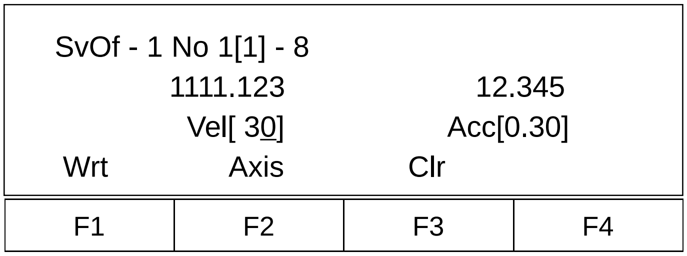


```
SvOf - 1 No 1[1]  - 8
        1111.123           12.345
         123.456         1234.567
Axis+      Axis-      Vel       Etc
F1         F2         F3        F4
```

14-1 SvOF Mode (Step 14. F4)

Velocity Setting, Axis No. Selection
Use to set point position velocity or to change axis number.

F3- →Velocity Setting Selection Display (P71. 14-2)

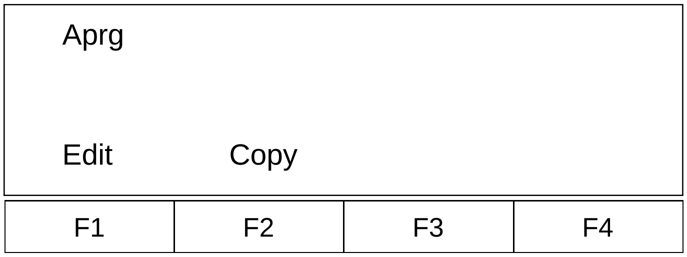


```
SvOf - 1 No 1[1] - 8
      1111.123          12.345
        Vel[ 30]      Acc[0.30]
Wrt        Axis       Clr
F1         F2         F3        F4
```

14-2 SvOf Mode (Step 14-1. F3)

Velocity Setting Selection
Input velocity and accerelation, then press Return key.
F2- Return to SvOf Mode (P71. 14)
F3- Clr (Clear) clears input value, for re-entry.

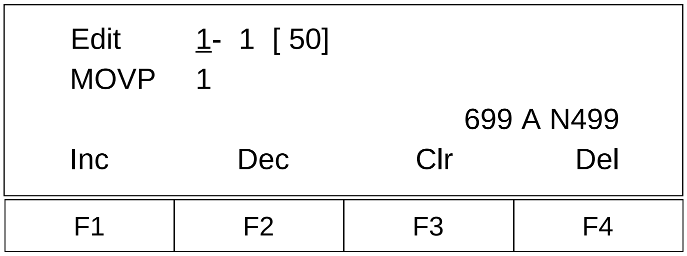
② Application Programming



```
  Aprg


  Edit        Copy

   F1    |    F2    |    F3    |    F4
```

15. Super SEL Programming Edit Display (Go back to the display in Step 4, then press F2)

    F1- (Edit Mode) Perform SEL programming, editing, addition, insertion, and deletion. (P72. 16)
    F2- (Copy Mode) Copies or overwrites programs. (P74. 17)



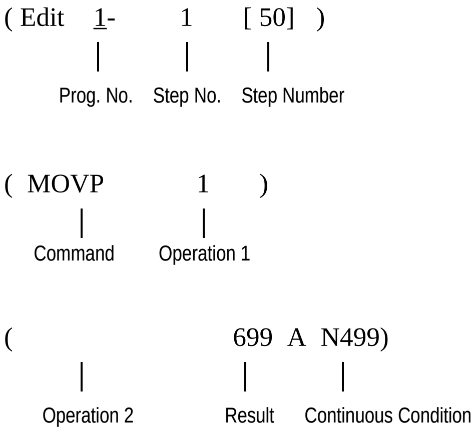
```
  Edit    1-  1  [ 50]
  MOVP    1
                          699 A N499
  Inc         Dec       Clr       Del

   F1    |    F2    |    F3    |    F4
```

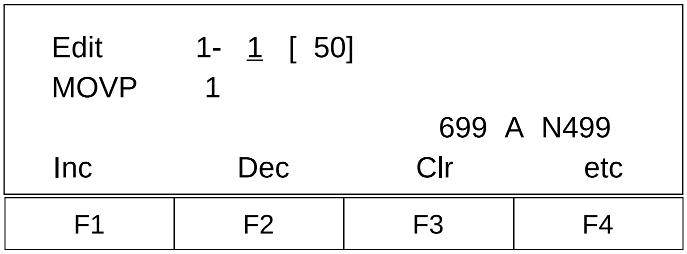
16. Super SEL Programming Edit Mode (15. F1)

16-1 Program Number Input Mode

F1- Increments program number by 1.
F2- Decrements program number by 1.
F3- Clears input value for re-entry.
F4- Deletes program number.
Return- Switches to step number input mode when the program number input is within the appropriate range. (P73. 16-2)

When a program number exists, step number 1 will be displayed. When a program number does not exist, a space will be displayed.



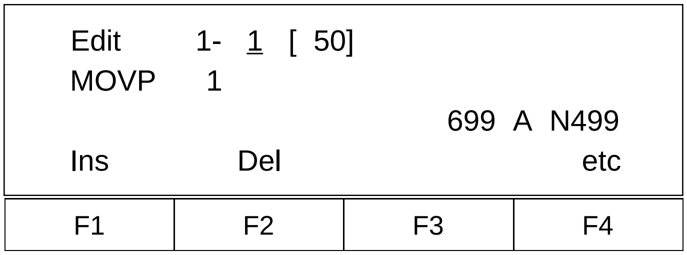


```
Edit       1-  1  [  50]
MOVP        1
                            699  A  N499
Inc           Dec          Clr          etc
```

F1	F2	F3	F4

16-2 Step Number Input Mode (Step 16-1. Return Key)

F1- (Inc) Increments step number by 1.
F2- (Dec) Decrements step number by 1.
F3- (Clr) Clears data for re-entry.
F4- Additional mode (Insertion mode, deletion mode, etc.) (P73. 16-2-1)

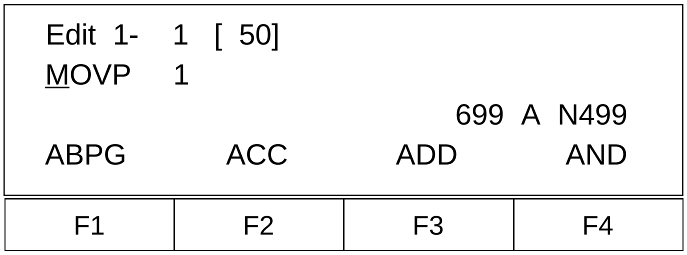


```
  Edit     1-  1  [  50]
  MOVP      1
                             699  A  N499
  Ins         Del                       etc
```

F1	F2	F3	F4

16-2-1 Step Number Input Mode 1 (Step 16-2. F4)

F1- (Insertion) Add steps.
F2- (Delete) Delete steps.

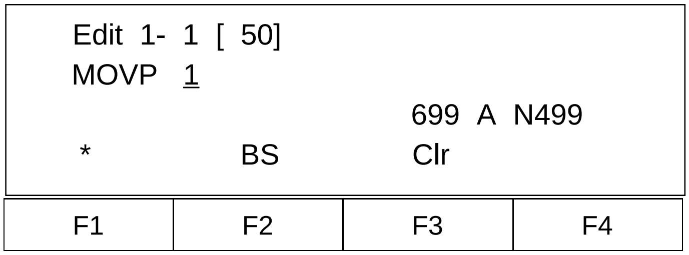


```
Edit  1-   1  [  50]
MOVP     1
                            699  A  N499
ABPG         ACC         ADD          AND
```

F1	F2	F3	F4

16-3 Edit Command Input Mode (Step 16-2. Return Key)
4 commands will be displayed from F1 to F4.

Pressing the decimal point (.) will display the commands in alphabetical order.
Pressing the minus sign (-) will display the commands in reverse alphabetical order.
Select a command by pressing F1 ~ F4, **.** (decimal point), or - (minus) sign.
Press CR (Return key) to display Operation 1 Input Mode (P73. 16.4).

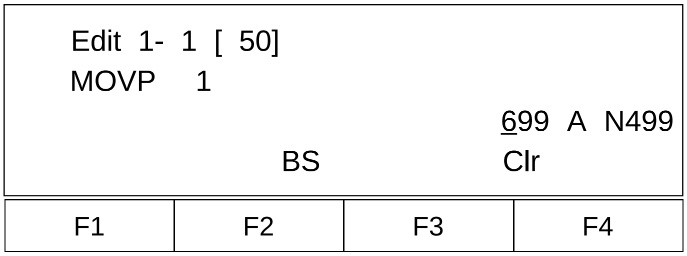


```
   Edit  1-  1  [  50]
   MOVP   1
                         699  A  N499
    *          BS          Clr
```

F1	F2	F3	F4

16-4 Operation 1 Input Mode (Step 16-3. Return key)

Input operation 1 of the selected command.
F1- (*) Designate variables indirectly.
F2- (BS: Back Space) Clears last input value. Cursor will move backwards.
F3- (Clr: Clear) Clears all the values input for new data entry.
Return- Input values will be saved, then operation 2 input mode will be displayed. (P74. 16-5)
ESC- Command input mode



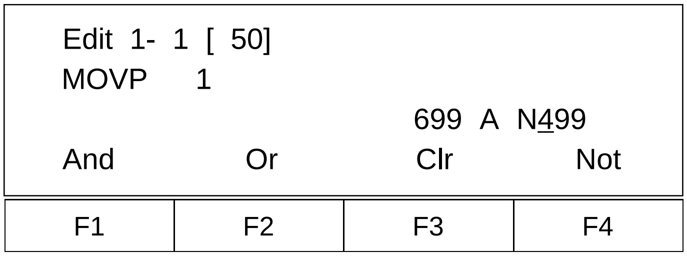
```
Edit  1-  1  [  50]
MOVP    1
                                  699  A  N499
                    BS            Clr
```

16-5   Operation 2 Input Mode (Step 16-4, Return key)

Same as 16-4

16-6   Result Input Mode (16-4. Return key)

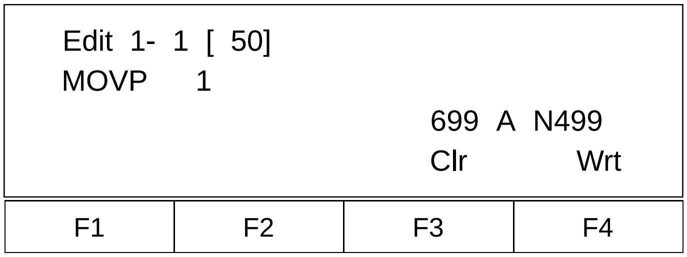
Input result, output and flag.
F2- Clears the last entry and moves cursor backwards using BS (back space).
F3- Clears all data.
Return- Continuous Condition Input Mode (P74. 16-7)



```
Edit  1-  1  [  50]
MOVP     1
                            699  A  N499
And          Or             Clr         Not
```

16-7   Continuous Condition Input Mode (Step 16-6, Return key)

F1- Select And
F2- Select Or
F3- Clears data entry
F4- Select Not
Return- Confirmation Mode (P74. 16-8)

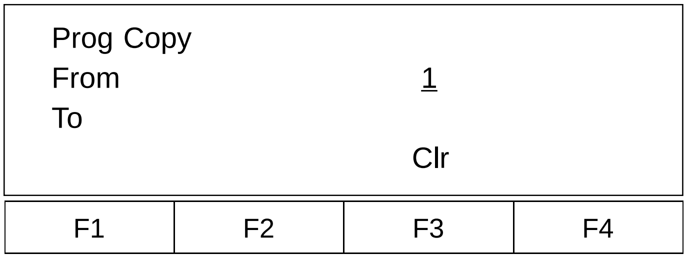


```
Edit  1-  1  [  50]
MOVP     1
                              699  A  N499
                              Clr         Wrt
```

16-8   Confirmation Mode (Step 16-7. Return key)

F3- Re-input from command
F4- Proceeds to the next step number after saving (Wrt) the current step number. (In the case of insertion, the step number remains the same).

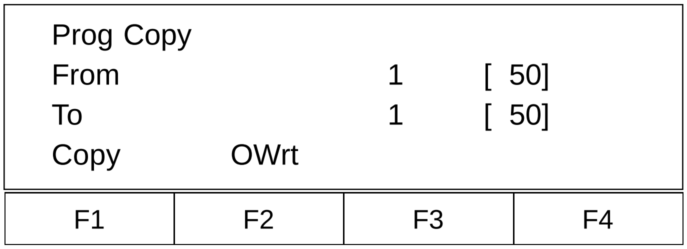
17.   Copy Mode (Go back to the display in Step No. 15, then press F2.)



```
Prog Copy
From                        1
To
                            Clr
```

17-1 Copies program. (Step 15. F2)

Input program number to be copied after "From", then input a new program number after "To".
F3- Clears data for data entry.
Return- Copy, Overwrite Selection Mode (P74. 17-2)



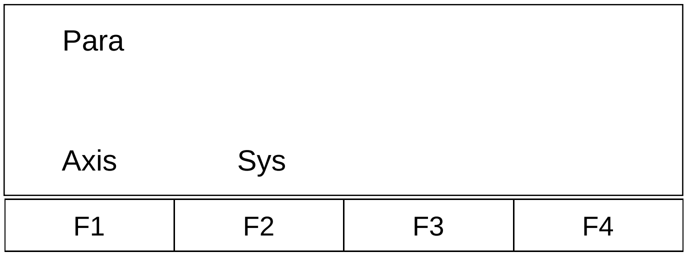
```
Prog Copy
From                      1      [  50]
To                        1      [  50]
Copy         OWrt
```

17-2   Copy, Overwrite Selection Mode (17-1. CR: return key)

F1- Copies the program.
F2- Overwrites the program.

3. Parameter Mode

Important Please contact our technical service department if parameters need to be changed for your system.



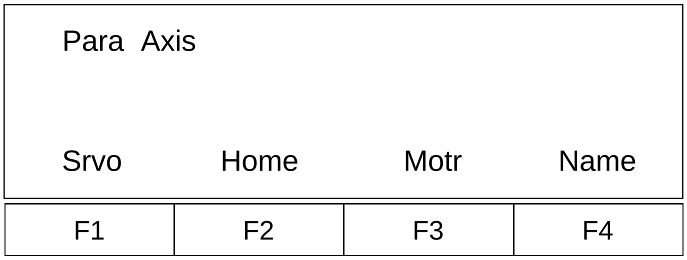
Para

Axis Sys

F1	F2	F3	F4

18. Parameter Mode (Step 3, F3)

F1- Parameter mode for each axis (P75. 18-1)
F2- System parameter mode (P75. 18-2) (P77.20)



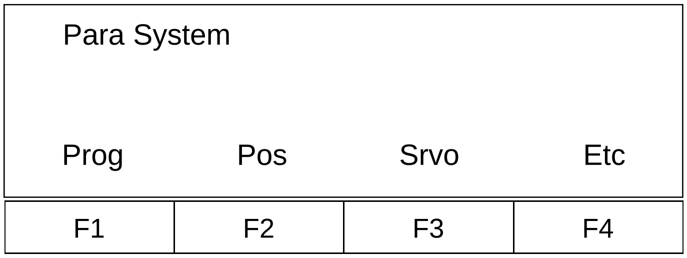
Para Axis

Srvo Home Motr Name

F1	F2	F3	F4

18-1 Parameter Mode for each axis (Step 18. F1)

F1- Servo parameter mode for each axis (P75.19-1)
F2- Home parameter mode for each axis (P76.19-2), axis activation mode
F3- Motor parameter mode for each axis (P76. 19-3)
F4- Axis name parameter mode for each axis (P77. 19-4)



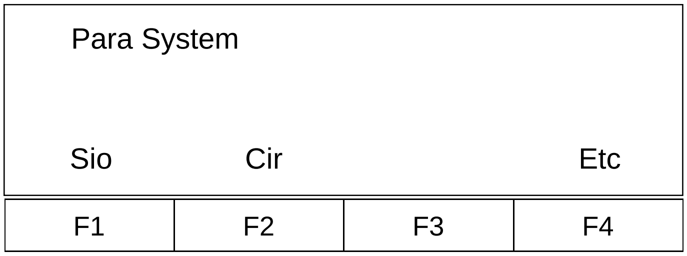
Para System

Prog Pos Srvo Etc

F1	F2	F3	F4

18-2 System Parameter Mode (Step 18. F2)

F1- System program parameter mode
F2- System point parameter mode
F3- System servo parameter mode
F4- Additional selection mode (P75. 18-3)



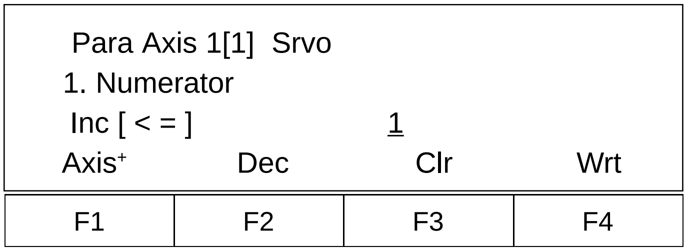
Para System

Sio Cir Etc

F1	F2	F3	F4

18-3 System parameter additional selections (Step 18-2. F4)

F1- (Sio: Serial I/O RS232C) System Sio parameter mode (P78. 20-4)
F2- (Cir: Circle) System path/circular parameter mode (P78. 20-5)
F4- (Etc) Returns to system parameter mode. (P75. 18-2)

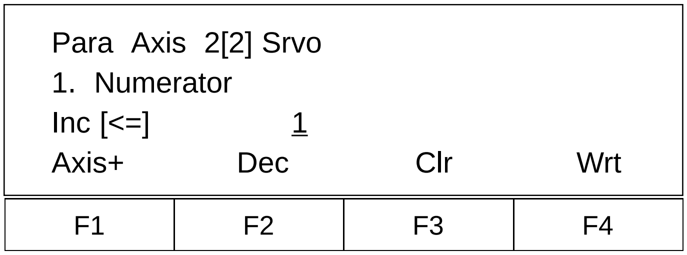


Para Axis 1[1] Srvo
1. Numerator
Inc [< =] 1
Axis^{+} Dec Clr Wrt

F1	F2	F3	F4

19-1 Servo Parameter Mode (18-1. F1)

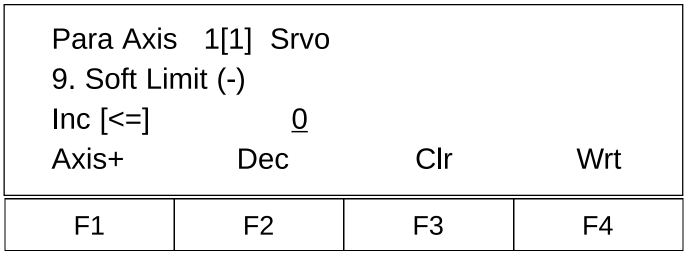
F1- Goes to next axis no. (+1). Axis name will also be changed. (P76. 19-1-1)
F2- Goes backward -1 (P76. 19-1-2)
F3- Clears data for data entry
F4- Direct number input
Return- (Inc[<=]) adds +1



```
Para  Axis  2[2] Srvo
1.  Numerator
Inc [<=]            1
Axis+         Dec          Clr          Wrt
```

F1	F2	F3	F4

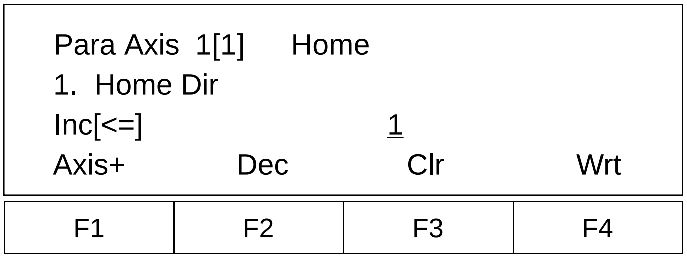
19-1-1 Press F1 to display axis number +1.



```
Para Axis   1[1]  Srvo
9. Soft Limit (-)
Inc [<=]            0
Axis+         Dec          Clr          Wrt
```

F1	F2	F3	F4

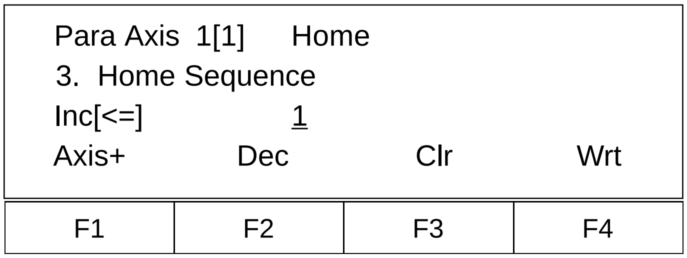
19-1-2 Press F2 to display item -1.



```
Para Axis  1[1]     Home
1.  Home Dir
Inc[<=]                     1
Axis+         Dec          Clr          Wrt
```

F1	F2	F3	F4

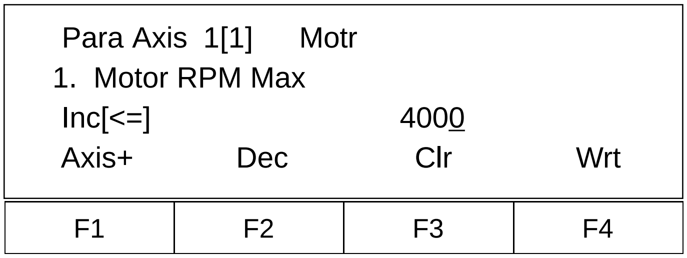
19-2 Home parameter mode for each axis, axis activation mode (Step 18-1, F2)



```
Para Axis  1[1]     Home
3.  Home Sequence
Inc[<=]             1
Axis+         Dec          Clr          Wrt
```

F1	F2	F3	F4

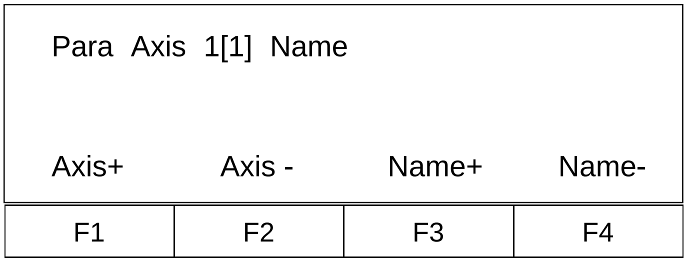
19-2-1 Home parameter mode for each axis, axis activation mode (Step 19-2, Press the RETURN key twice). Doing this brings you to the 3. Home Sequence screen. By inputting 0 here, the axis will be inactivated. Homing or Jog will not be performed. (Position indicator will display Δ).
By inputting a number (1 ~ 9), homing order for all activated axes can be designated.



```
 Para Axis  1[1]     Motr
1.  Motor RPM Max
 Inc[<=]                     4000
 Axis+        Dec          Clr          Wrt
```

F1	F2	F3	F4

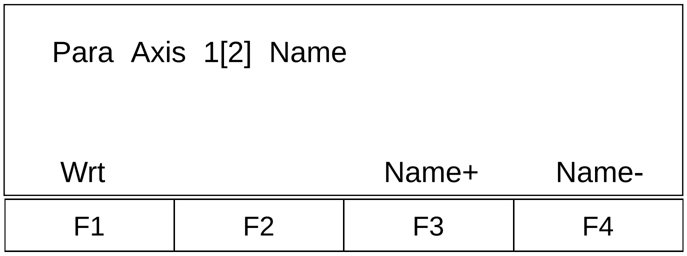
19-3 Motor Parameter Mode (Step 18-1. F3)



```
Para  Axis  1[1]  Name


Axis+        Axis -        Name+        Name-
F1           F2            F3           F4
```

19-4 Parameter Name Mode for Each Axis (Step 18-1. F4 Key)

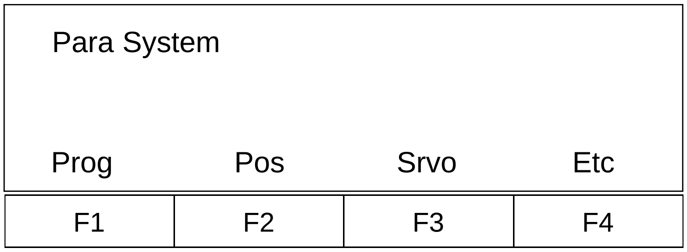
F1- Increments axis number by 1
F2- Decrements axis number by 1
F3- Increments axis name by 1 for range 1 ~ 9, A ~ Z (P77. 19-4-1)
F4- Decrements axis name by 1 for range 1 ~ 9, A ~ Z



```
Para  Axis  1[2]  Name


Wrt                        Name+        Name-
F1           F2            F3           F4
```

19-4-1 Axis Name Change (Step 19-4. F3)

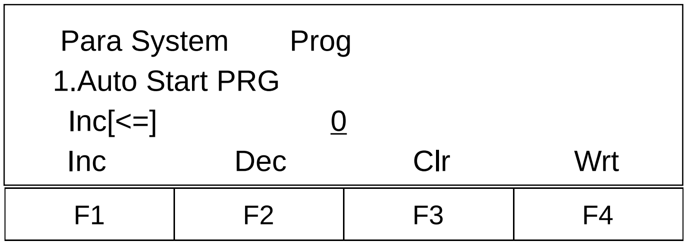
F1- (Wrt) Change axis name.



```
Para System


Prog         Pos           Srvo         Etc
F1           F2            F3           F4
```

20. System Parameter Mode (Step 18. F2)

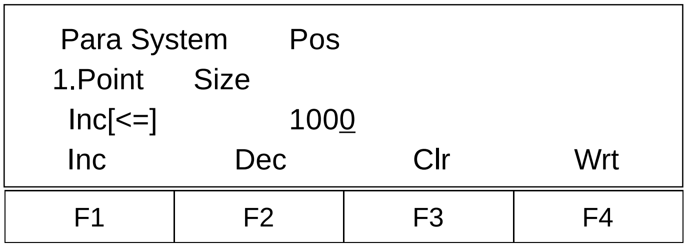
F1- System Program Parameter Mode (P77. 20-1)
F2- System Point Parameter Mode (P77. 20-2)
F3- System Servo Parameter Mode (P78. 20-3)



```
Para System      Prog
1.Auto Start PRG
 Inc[<=]                0
 Inc           Dec          Clr          Wrt
F1             F2           F3           F4
```

20-1 System Program Parameter Mode (Step 20. F1)

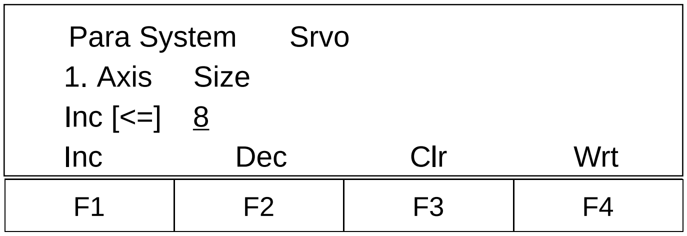
Press the RETURN key to increment the item. (Inc[<=])



```
Para System      Pos
1.Point     Size
 Inc[<=]               1000
 Inc           Dec          Clr          Wrt
F1             F2           F3           F4
```

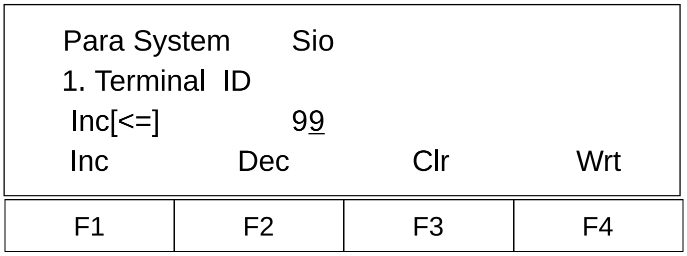
20-2 System Point Parameter Mode (Step 20. F2)

20-3 System Servo Parameter Mode (Step 20. F3)



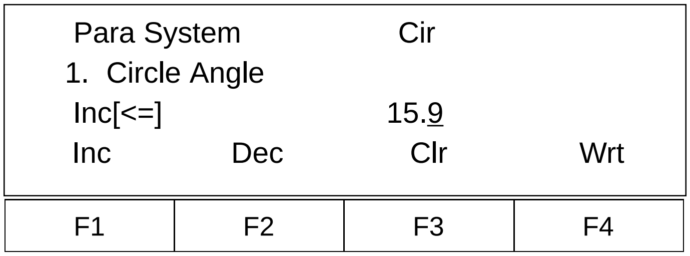
```
Para System      Srvo
1. Axis     Size
Inc [<=]    8
Inc           Dec           Clr           Wrt
F1            F2            F3            F4
```

20-4 System Sio Parameter Mode (18-3. F1 Key)



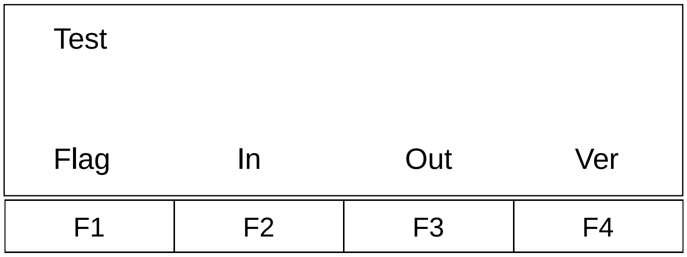
```
Para System       Sio
1. Terminal  ID
Inc[<=]           99
Inc           Dec           Clr           Wrt
F1            F2            F3            F4
```

20-5 System Circle, Arc, Parameter Mode (Step 18-3. F2)



```
Para System                 Cir
1.  Circle Angle
Inc[<=]                    15.9
Inc           Dec           Clr           Wrt
F1            F2            F3            F4
```

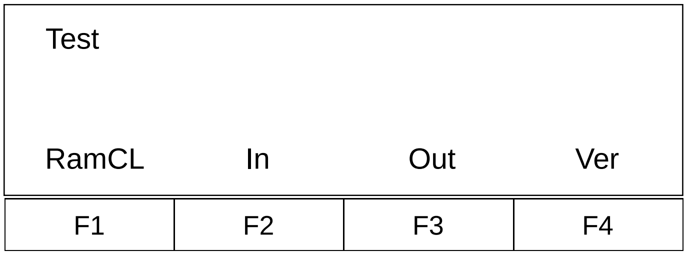
4. Test Mode



```
Test


Flag          In          Out          Ver
F1            F2          F3           F4
```

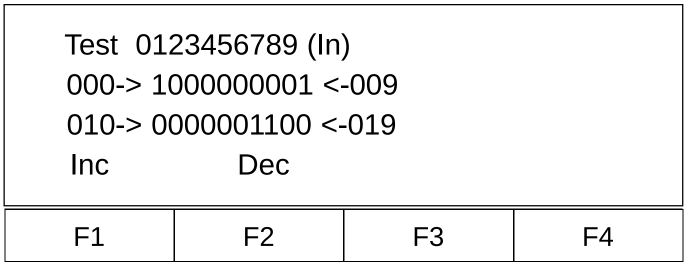
21. Test Mode (3. F4 Key)

F1- (Flag) Flag displayed. Press F1 and "1" key simultaneously. RamCL (Ram Clear) Mode will appear on the display.
F2- (In) Input Port Display (P79. 21-2)
F3- (Out) Output Port (P79. 21-3)
F4- (Ver: Version) Displays current software version of Servo, Main and Teaching Pendant (P80. 21-4)



```
Test


RamCL         In          Out          Ver
F1            F2          F3           F4
```

21-1 While holding down both the F1 and "1" keys, the display will remain the same as shown on the left. When the keys are released, Memory Clear Mode (RamCL) will appear. (P80. 21-5)

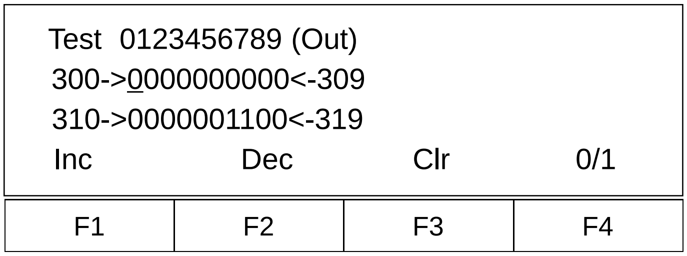


```
Test  0123456789 (In)
000-> 1000000001 <-009
010-> 0000001100 <-019
Inc           Dec
F1            F2          F3           F4
```

21-2 Test Input Port Display (Step 21. F2)

Displays input port status.

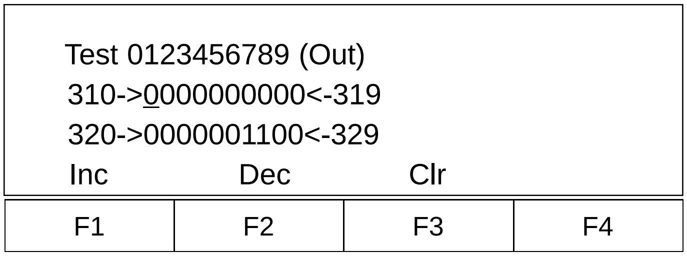
F1- Port displayed. +10
F2- Port displayed. -10



```
Test  0123456789 (Out)
300->0000000000<-309
310->0000001100<-319
Inc           Dec         Clr          0/1
F1            F2          F3           F4
```

21-3 Test Output Port Display (21. F3)

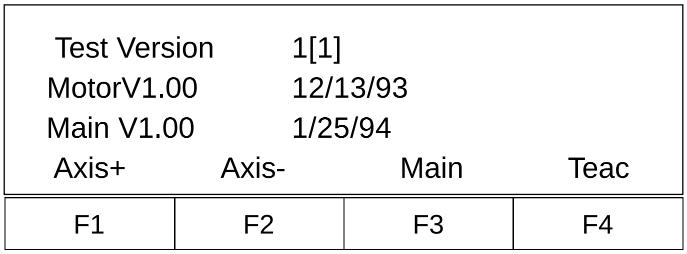
F1- Output port +10 (P79. 21-3-1)
F2- Output port -10
F3- All displayed ports will be 0 output
F4- Output 0 ⇒1, 1⇒0
. (decimal point), RETURN key moves cursor to the right
- (minus key) moves cursor to the left
"0" outputs cursor position port to 0
"1" outputs cursor position port to 1
With this display, you can force output ports ON/OFF, but you cannot monitor outputs as they change on the controller side in real time.



```
Test 0123456789 (Out)
310->0000000000<-319
320->0000001100<-329
Inc           Dec         Clr
F1            F2          F3           F4
```

21-3-1 F1 (output port +10)

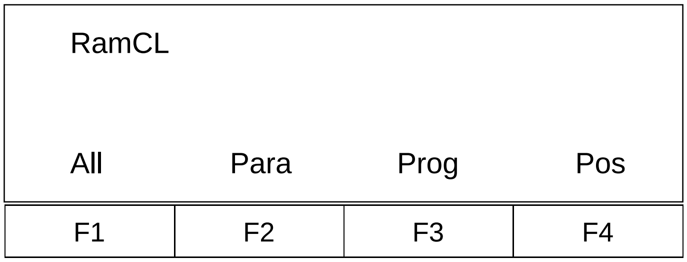
21-4 Test Version Mode (Step 21. F4)



```
Test Version      1[1]
MotorV1.00        12/13/93
Main V1.00        1/25/94
Axis+     Axis-      Main      Teac
F1        F2         F3        F4
```

21-5 Memory Clear (RamCL) Mode (Step 21. F1 + 1 key)
Caution: This will erase all data. Make sure to backup all data before doing this operation.

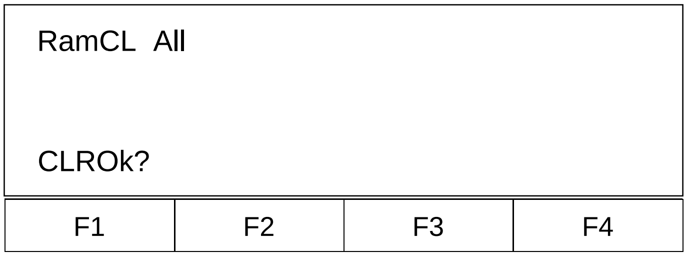
F1- Clears system parameter, program, and position area (P80. 21-5-1)
F2- Clears system parameter (P80. 21-5-2)
F3- Clears application program area (P80. 21-5-3)
F4- Clears position data area (P81. 21-5-4)



```
RamCL


All       Para       Prog      Pos
F1        F2         F3        F4
```

21-5-1 F1(All) is selected.

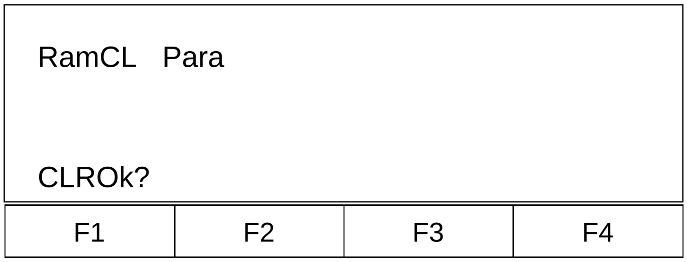
Press F1 (CLROk?: Clear OK?)
Note: This is the same as RESET.



```
RamCL  All


CLROk?
F1        F2         F3        F4
```

21-5-2 F2 (Para) is selected.

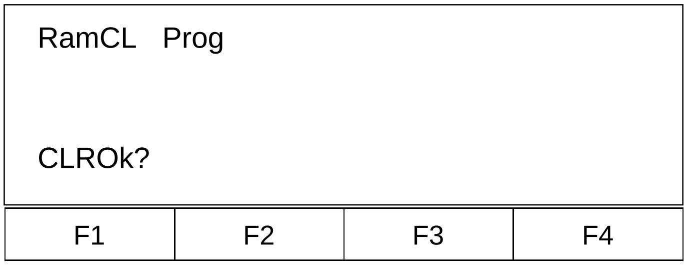
Press F1 (CLROk?) - same as Reset
Note: This is the same as RESET.



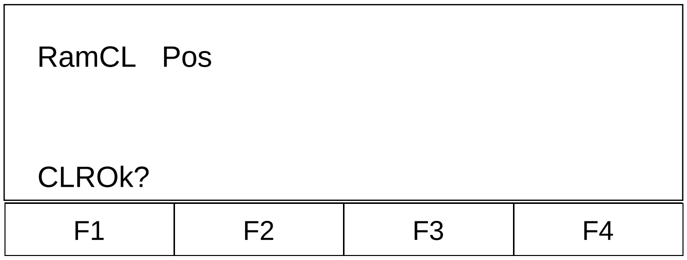
```
RamCL   Para


CLROk?
F1        F2         F3        F4
```

21-5-3 F3 (Prog) is selected.

Press F1
Clears all application programs.



```
RamCL   Prog


CLROk?
F1        F2         F3        F4
```

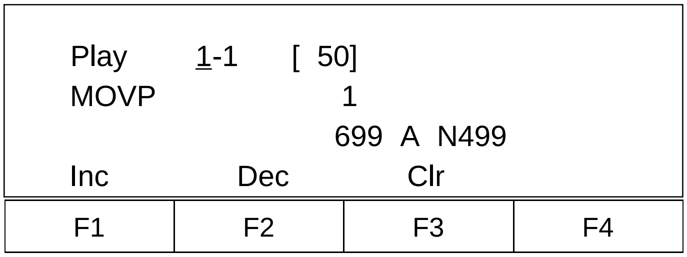


```
RamCL   Pos


CLROk?
   F1        F2        F3        F4
```

21-5-4   F4 (Pos) is selected.

Press F1
Clears all position data.

5.   Play Mode

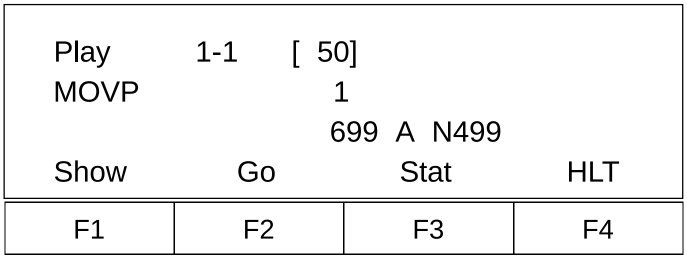


```
Play      1-1      [  50]
MOVP                  1
                    699  A  N499
Inc          Dec          Clr
   F1        F2        F3        F4
```

22.   Play (Play) Mode (3, F2 Key)

22-1   Play Program Input Mode

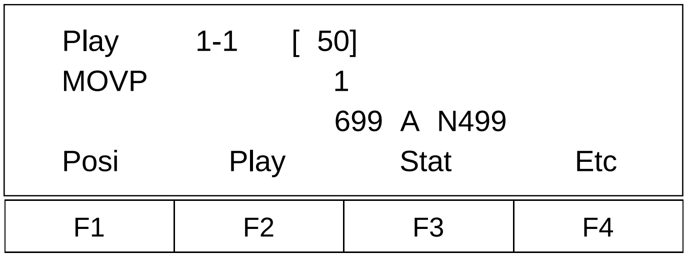
Input program number to execute (or stop). Program status can be seen by pressing START then PROG.



```
Play      1-1      [  50]
MOVP                  1
                    699  A  N499
Show         Go           Stat         HLT
   F1        F2        F3        F4
```

22-2 Program Execution/Stop Selection (22-1. CR: return key)

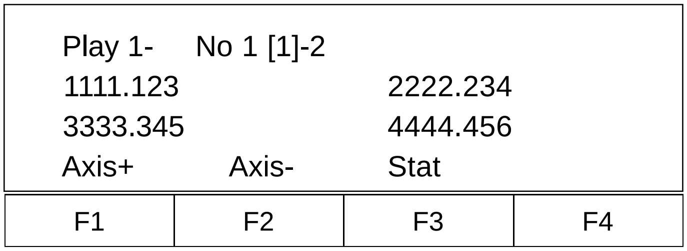
F1- (Show) Monitoring (when the designated program is executing.) (P84. 23)
F2- (Go) Program execution (P81. 22-3)
F3- (Status) show status (after checking the status, select execution or stop)
F4- (HLT) Stop (P81. 22-3)



```
Play      1-1      [  50]
MOVP                  1
                    699  A  N499
Posi         Play         Stat         Etc
   F1        F2        F3        F4
```

22-3   Execution Status Selection (22-2. F2) (22-2. F4)

F1- Axis status display (P81. 22-4)
F2- Other program execution, or stops currently executing program. (P82. 22-5)
F3- Shows program status (P82. 22-6)
F4- Input port, Output port, Flags (P82. 22-7)

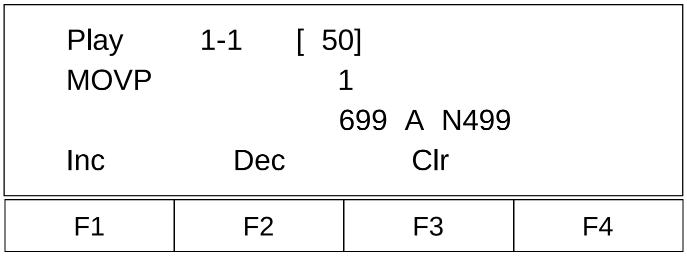


```
Play 1-     No 1 [1]-2
1111.123               2222.234
3333.345               4444.456
Axis+        Axis-        Stat
   F1        F2        F3        F4
```

22-4   Axis Status Display Mode (22-3. F1)

F1- Checks other axis status (+)
F2- Checks other axis status (-)

(Play   1-   No 1   [2]-8)
↑
Starting axis

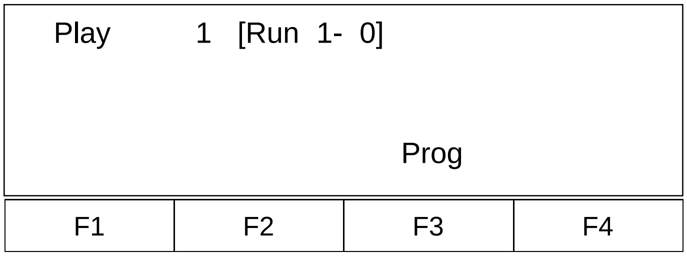


```
Play       1-1     [ 50]
MOVP                  1
                      699  A  N499
Inc          Dec          Clr
```

F1 F2 F3 F4

22-5 Execution, Stop Mode (22-3, F2)

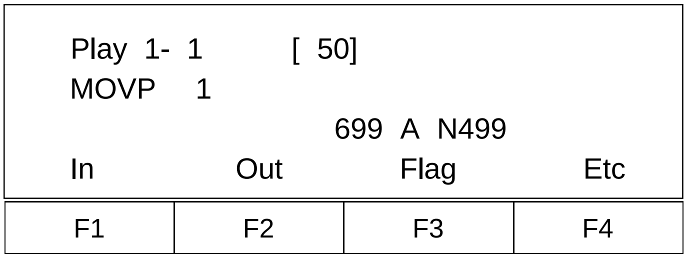
Input program number to be executed or stopped. Returns to 22-3.



```
Play         1   [Run  1-  0]



                          Prog
```

F1 F2 F3 F4

22-6 System movement and program number display (22-3, F3)

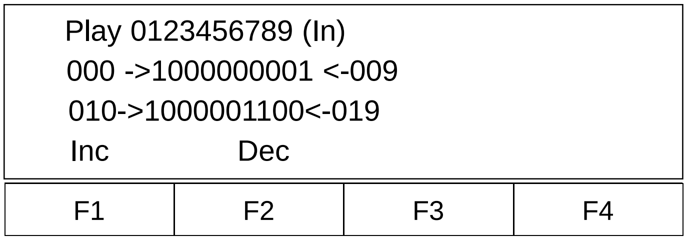


```
Play  1-  1        [  50]
MOVP    1
                     699  A  N499
In           Out          Flag          Etc
```

F1 F2 F3 F4

22-7 I/O port, Flag Status Selection Mode (22-3. F4)

F1- Input port change display, select (P82. 22-8)
F2- Ouput port change display, select (P82. 22-9)
F3- Flag change display, select (P83. 22-10)
F4- Execute, status selection screen (P83. 22-12)

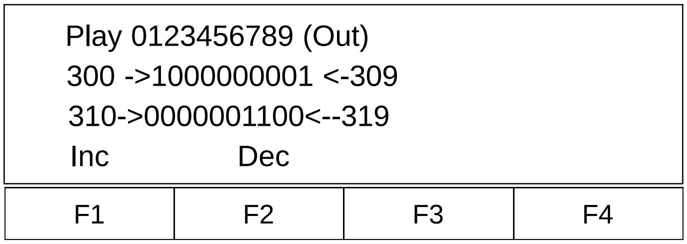


```
Play 0123456789 (In)
000 ->1000000001 <-009
010->1000001100<-019
Inc          Dec
```

F1 F2 F3 F4

22-8 Play Input Port Display (22-7. F1)

F1- Input Port +10
F2- Input Port -10

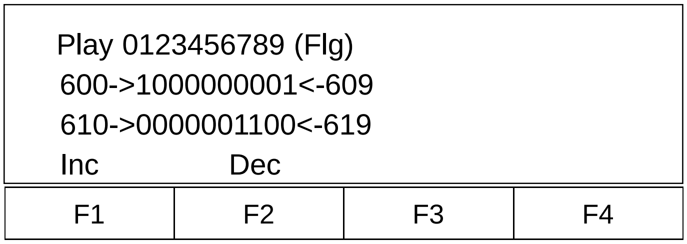


```
Play 0123456789 (Out)
300 ->1000000001 <-309
310->0000001100<--319
Inc          Dec
```

F1 F2 F3 F4

22-9 Play Output Display (22-7. F2)

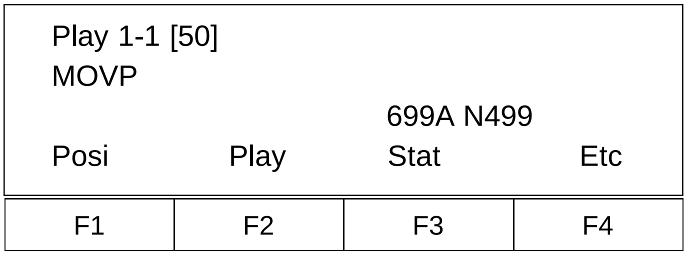
F1- Output +10
F2- Output -10



```
Play 0123456789 (Flg)
600->1000000001<-609
610->0000001100<-619
Inc          Dec
F1      F2      F3      F4
```

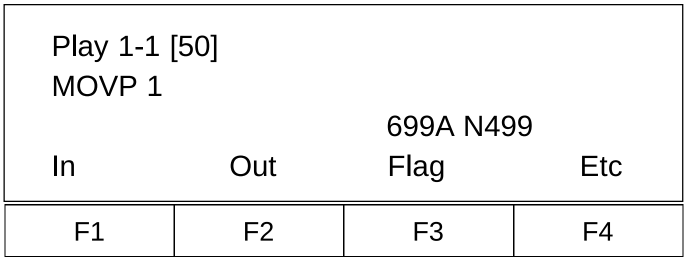
22-10 Play Flag Display (22-7. F3)

F1- Flag +10
F2- Flag -10



```
Play 1-1 [50]
MOVP
                          699A N499
Posi          Play        Stat          Etc
F1      F2      F3      F4
```

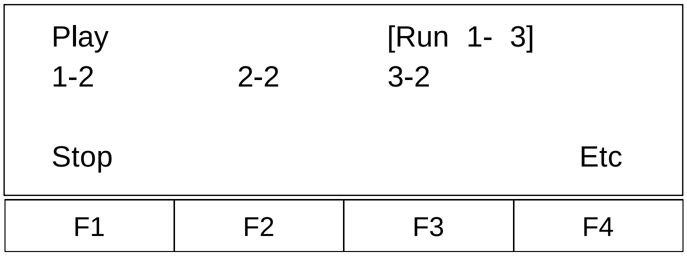
22-11-1 Select F1 (Show), F2 (GO), or F4 (HLT) from Program Execution/Stop Selection Display (P81. 22-2) to stop Play Mode (program stops). Execution Status Selection Display appears. Then press F4 (Etc). (P83. 22-11-2)



```
Play 1-1 [50]
MOVP 1
                          699A N499
In            Out         Flag          Etc
F1      F2      F3      F4
```

22-11-2 I/O Port, Flag Status Selection Mode (P82. 22-7)

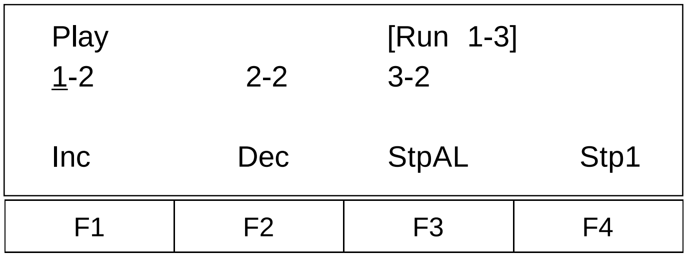
Press F4 (Etc). (P83. 22-12)



```
Play                      [Run  1-  3]
1-2           2-2         3-2

Stop                                    Etc
F1      F2      F3      F4
```

22-12. Execute, Status Selection Screen (22-7. F4)

F1 - Goes to Program Stop Selection Mode.
(P83. 22-13)

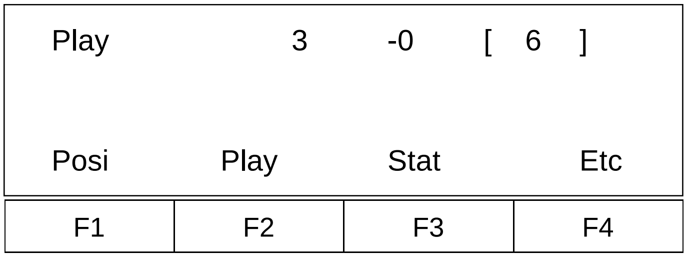


```
Play                      [Run  1-3]
1-2           2-2         3-2

Inc           Dec         StpAL         Stp1
F1      F2      F3      F4
```

22-13 Program Stop Selection Mode (22-12. F1)

F1- Changes program number to be stopped.
F2- Changes program number to be stopped.
F3- Stops all the programs being executed.
F4- Stops program number with the cursor underneath.

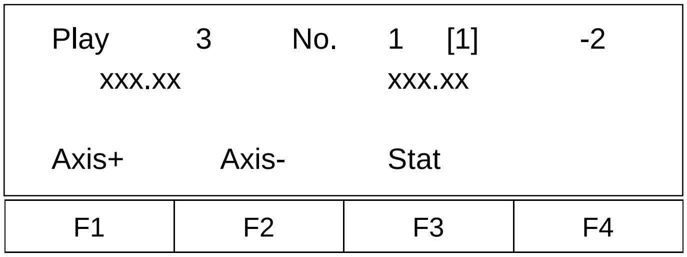
6. Show Mode



```
Play              3      -0      [  6  ]


Posi        Play        Stat        Etc
   F1          F2          F3          F4
```

23. Show Mode (Step 22-2. F1)

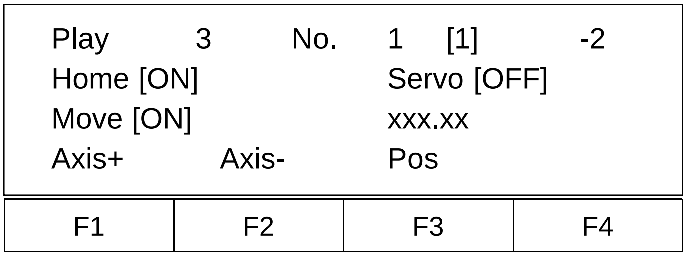
F1- Displays the position of each axis currently moving (P84. 23-1-1)
F2- Execute other program (P85.23-2)
F3- Shows program status (P85. 23-3-1)
F4- Current Input/Output Status, Flag Status Mode (P86. 23-4)



```
Play      3      No.    1   [1]       -2
   xxx.xx               xxx.xx

Axis+      Axis-       Stat
   F1          F2          F3          F4
```

23-1-1 Position Display Mode (Step 23. F1)

F1- Checks other axis status (+)
F2- Checks other axis status (-)
F3- Shows program status (P84. 23-1-2)

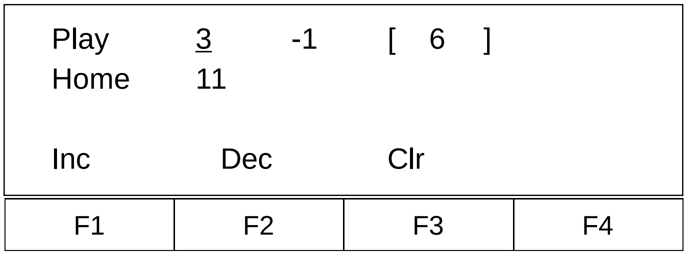


```
Play      3      No.    1   [1]       -2
Home [ON]               Servo [OFF]
Move [ON]               xxx.xx
Axis+      Axis-       Pos
   F1          F2          F3          F4
```

23-1-2 Status Display Mode (Step 23-1-1. F3)

Display homing incomplete (OFF)/complete (ON), servo ON/ OFF, moving incomplete (OFF)/complete (ON), and current value.

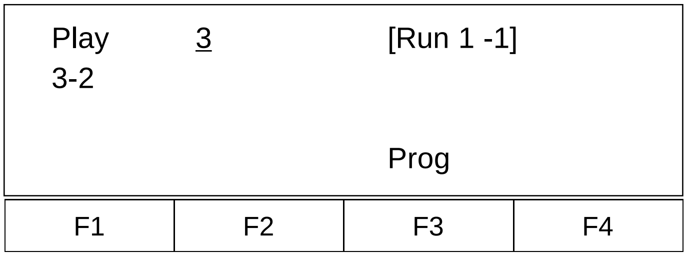
F3- Return to Position Display Mode (P84. 23-1-1)



```
Play    3     -1     [  6  ]
Home    11

Inc        Dec        Clr

F1        F2        F3        F4
```

23-2 Program Display Selection Mode (Step 23. F2)

F1- Increment Program No.
F2- Decrement Program No.
F3- Clear Program No.

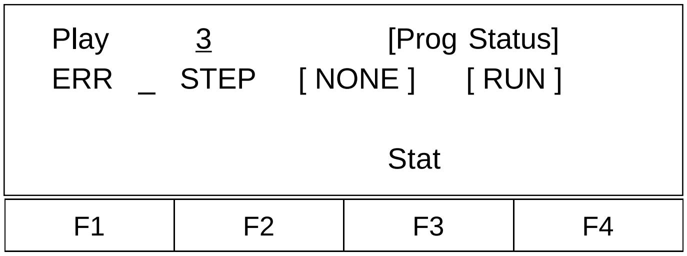


```
Play    3            [Run 1 -1]
3-2

                     Prog

F1        F2        F3        F4
```

23-3-1 Program Execution Status Mode (Step 23. F3)

3 -2
↓
↓ 2: Executable, 4: Wait for completion (Input)
Executing Program No.

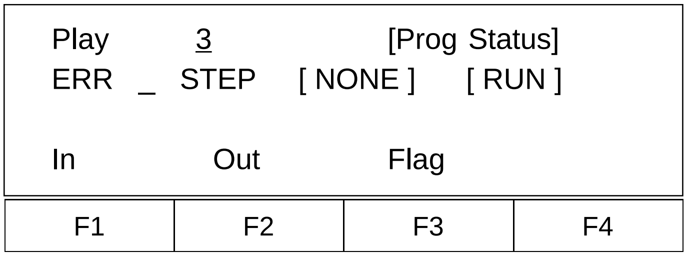
F3- Program Execution Status (P85. 23-3-2)



```
Play    3            [Prog Status]
ERR  _  STEP   [ NONE ]   [ RUN ]

                     Stat

F1        F2        F3        F4
```

23-3-2 Program Execution Status (Step 23-3-1. F3)

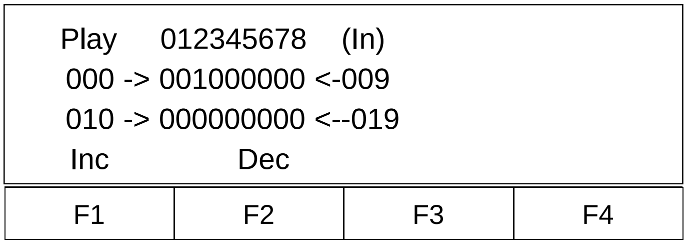
F3- Return to Program Execution Status Mode (P85. 23-3-1).



```
Play        3               [Prog Status]
ERR  _  STEP    [ NONE ]     [ RUN ]

In            Out           Flag
```

23-4 Input/Output, Flag Status Display Mode (Step 23. F4)

F1- Input Status Display Mode (P86. 23-4-1)
F2- Output Status Display Mode (P86. 23-4-2)
F3- Flag Status Display Mode (P86. 23-4-3)

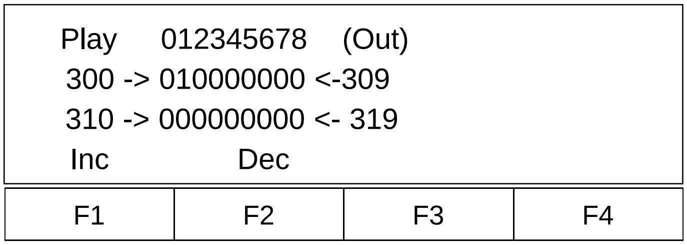


```
 Play    012345678   (In)
 000 -> 001000000 <-009
 010 -> 000000000 <--019
 Inc            Dec
```

23-4-1 Input Status Display Mode (Step 23-4. F1)

Display In 002 is ON.

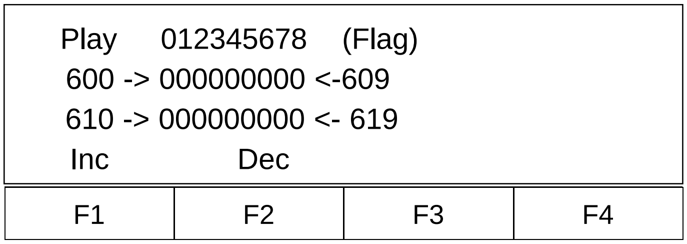
```
0   0   1   0  0  0  0  0  0  0
↓   ↓   ↓
000 001 002                  009   ←  Port No.
```



```
 Play    012345678   (Out)
 300 -> 010000000 <-309
 310 -> 000000000 <- 319
 Inc            Dec
```

23-4-2 Output Status Display Mode (Step 23-4. F2)

Display Out 301 is ON.



```
 Play    012345678   (Flag)
 600 -> 000000000 <-609
 610 -> 000000000 <- 619
 Inc            Dec
```

23-4-3 Flag Status Display Mode (Step 23-4. F3)

7. Simple Application Program Example

2-Axis actuator moves back and forth between point 1 and point 2.

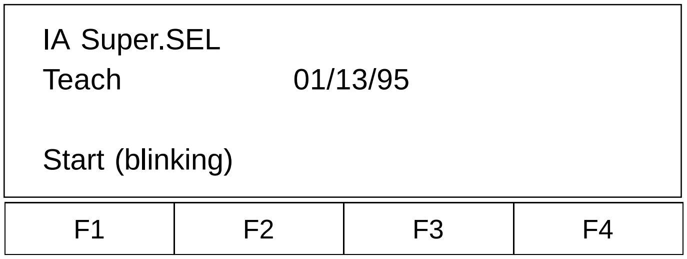
- Application Program

Command	Operand 1	Comment
HOME	11	X and Y axis homing (Servo ON)
VEL	100	Velocity setting 100mm/sec
ACC	0.3	Acceleration setting 0.3G
TAG	1	Assigns the target for GOTO command
MOVL	1	Moves to position number 1
MOVL	2	Moves to position number 2
GOTO	1	Jumps to TAG 1

- Position Data

Position Number	X Axis	Y Axis
1	100.000	100.000
2	10.000	10.000

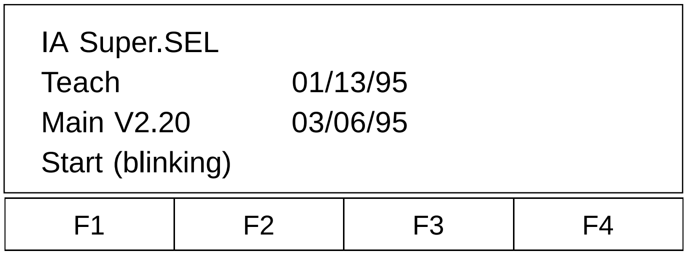
(LCD Display) *(Operation)*



```
IA Super.SEL
Teach              01/13/95

Start (blinking)
```

1. Initial Display

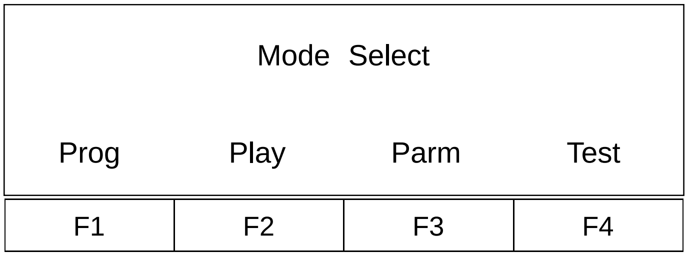
   Press F1 to start.



```
IA Super.SEL
Teach              01/13/95
Main V2.20         03/06/95
Start (blinking)
```

2. Controller ROM Version Display

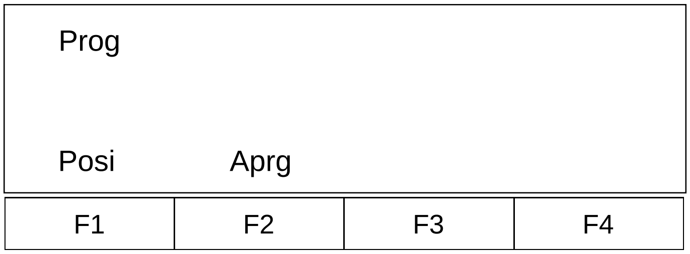
   Press F1 to start.



```
            Mode  Select

Prog       Play       Parm       Test
```

3. Mode Selection Display

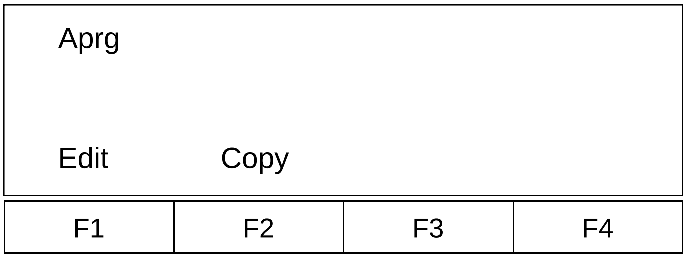
   Press F1 to select Prog Mode.



```
Prog


Posi        Aprg
F1   F2   F3   F4
```

4. Program Mode Display

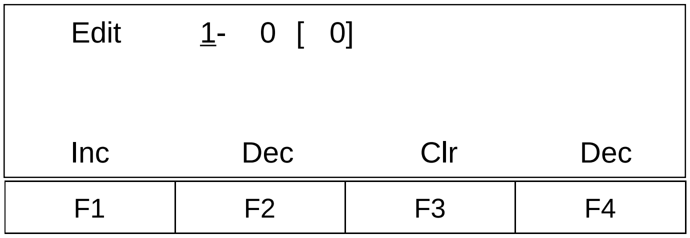
Press F2 to select Aprg (Application Program).



```
Aprg


Edit        Copy
F1   F2   F3   F4
```

5. Super SEL Programming Display

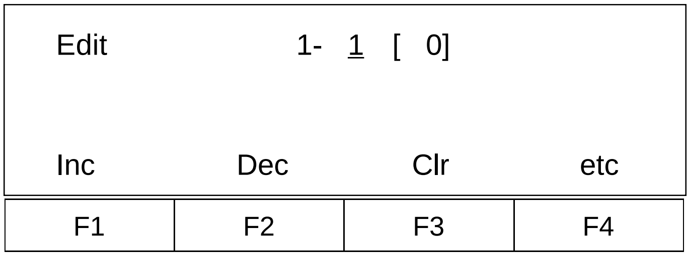
Press F1 to select Edit Mode to bring up Super SEL program, modify, add, insert, or erase.



```
Edit     1-   0  [   0]


Inc     Dec     Clr     Dec
F1      F2      F3      F4
```

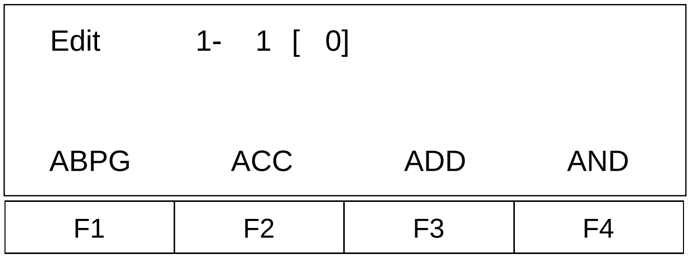
6. Super SEL Programming Edit Mode

Press 1 (Program number 1), then press return key. When the program number selected exists, the display shows the first step number. If the program number selected does not exist, a space will appear.



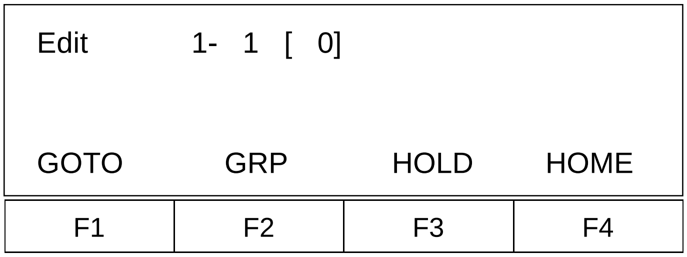
```
Edit            1-  1  [  0]


Inc     Dec     Clr     etc
F1      F2      F3      F4
```

7. Press return key.



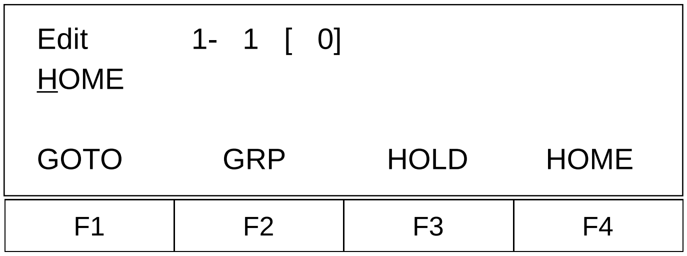
```
Edit     1-   1  [   0]


ABPG    ACC     ADD     AND
F1      F2      F3      F4
```

8. Press "." (decimal point) or "-" (minus) repeatedly until HOME appears.



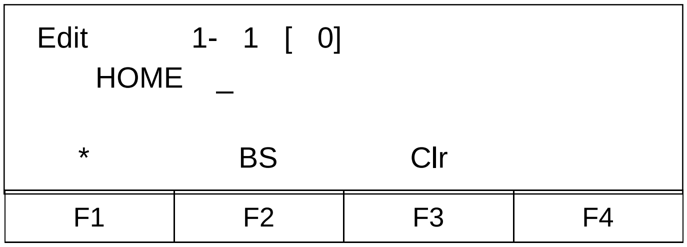
```
Edit        1-   1   [   0]


GOTO         GRP          HOLD       HOME
F1           F2           F3         F4
```

9. Press F4 to select HOME.



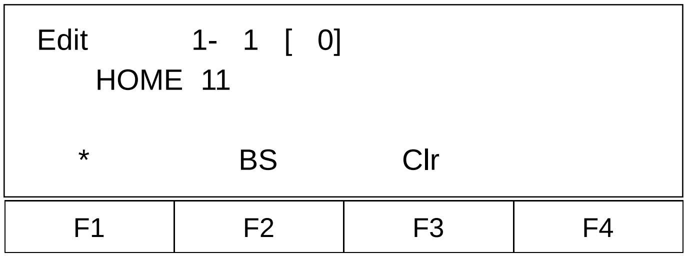
```
Edit        1-   1   [   0]
HOME

GOTO         GRP          HOLD       HOME
F1           F2           F3         F4
```

10. Press RETURN key.



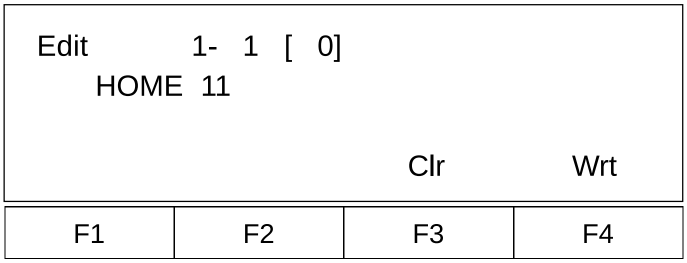
```
Edit        1-   1   [   0]
     HOME   _

   *           BS           Clr
F1           F2           F3         F4
```

11. Press "1" two times to home X and Y axes simultaneously.



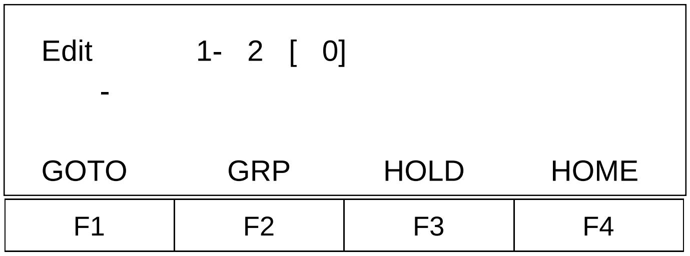
```
Edit        1-   1   [   0]
     HOME  11

   *           BS           Clr
F1           F2           F3         F4
```

12. Press return key 3 times.



```
Edit        1-   1   [   0]
     HOME  11

                            Clr        Wrt
F1           F2           F3         F4
```

13. Press F4 (Wrt: Writing).

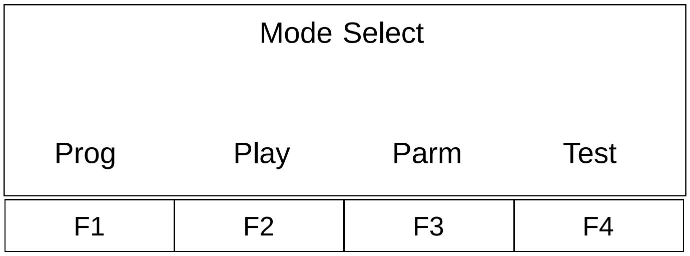


```
Edit        1-   2   [   0]
     -

GOTO         GRP          HOLD       HOME
F1           F2           F3         F4
```

14. Press "**.**" (decimal point) repeatedly until VEL appears on display.

Repeat the same process after step 9 to input commands.

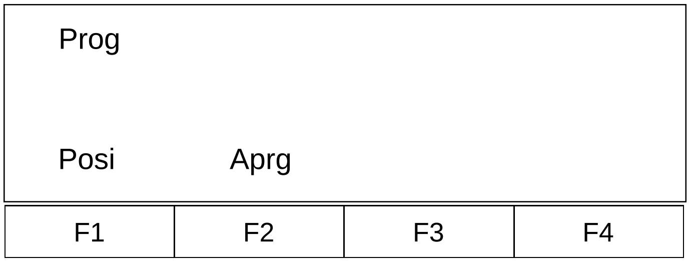
When programming is completed, press ESC key repeatedly until mode selection display appears.
Next, position data will be input.



```
              Mode Select


  Prog        Play        Parm        Test
  F1          F2          F3          F4
```

15. Mode Selection Display

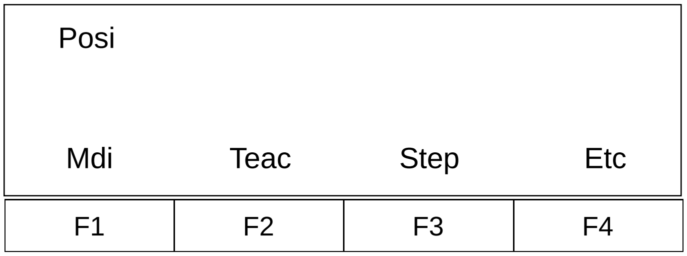
Press F1 to select Prog (Program) Mode.



```
  Prog


  Posi        Aprg
  F1          F2          F3          F4
```

16. Program Mode

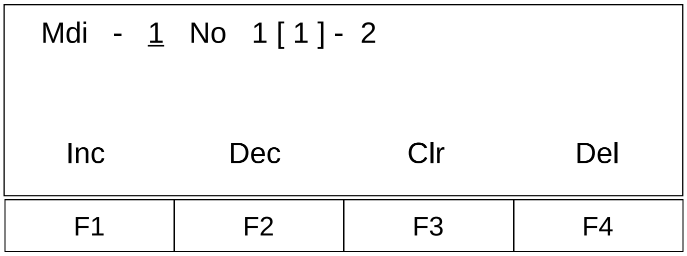
Press F1 to select Position data.



```
  Posi


  Mdi         Teac        Step        Etc
  F1          F2          F3          F4
```

17. Position Data Editing (Go back to the display in Step 4, then press F1).

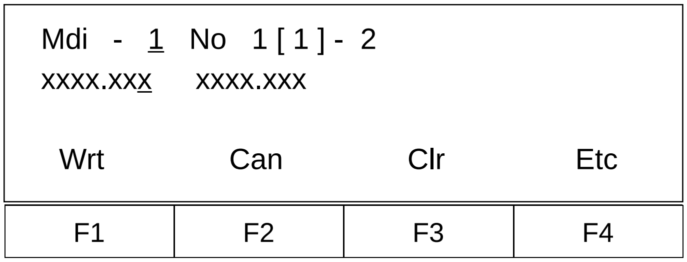
Press F1 to select Mdi Mode (position data direct input).



```
  Mdi  -  1  No  1 [ 1 ] -  2


  Inc         Dec         Clr         Del
  F1          F2          F3          F4
```

18. Mdi Mode

Press 1 for position data number 1, then press the RETURN key.



```
  Mdi  -  1  No  1 [ 1 ] -  2
  xxxx.xxx    xxxx.xxx

  Wrt         Can         Clr         Etc
  F1          F2          F3          F4
```

19. Press 100, then press RETURN key for X axis position data. Press 100, then press RETURN for Y axis position data.

Press F1 (Wrt: Writing) to save position data number 1 (100,100).

*Repeat the same process to input position data number 2.

Part 3 System Operation

Operating Mode Summary

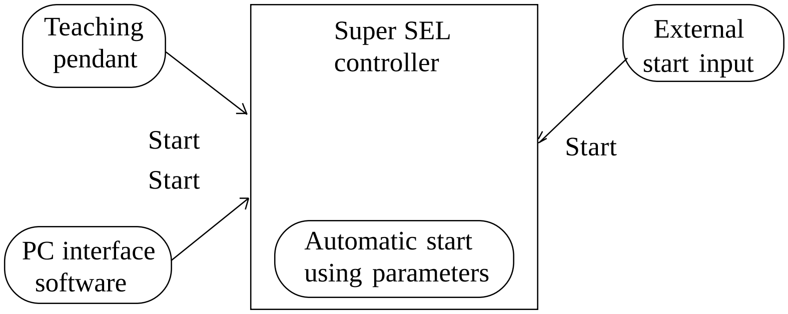
There are four operating modes in the Super SEL Controller System. Of these, two are primarily used for program debugging/trial operation and the remainng two are used in general applications at the factory site. The first two modes are: 1) operation from a teaching pendant and 2) operation from the PC interface software. These are used for simple operating checks. For the PC interface software mode, please read the operating manual that comes with the software. The latter two modes are: 3) automatic operation based on parameter settings and 4) operation based on selection of external signals. In the following sections, we will explain the operating modes except for 2, the PC software mode.

1. Automatic start using the parameters

Enter the number of the program you wish to start automatically in the controller parameter named "Automatic Start Program." After doing this, the program that was entered will start running automatically when the controller is reset or when it is powered up again after an emergency stop. This parameter can be set with the teaching pendant or the PC interface software.

! Things to remember when using the automatic start program

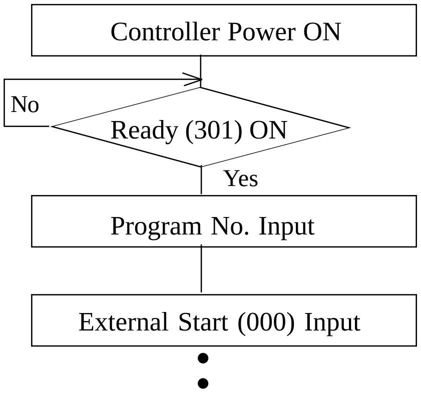
When you use an automatic start program, the sudden startup of the servo actuator can startle the operator. As a safety precaution, always use an interlock such as inserting a check signal at the beginning of the program so it will only start up after it receives the signal.
If you wish to run several programs at the same time, write the command to run the other programs using the [EXPG] command signal at the start of the primary automatic program.



2. External Start Operating Mode

Select Program No. from external unit, then input start signal.

① Flow Chart

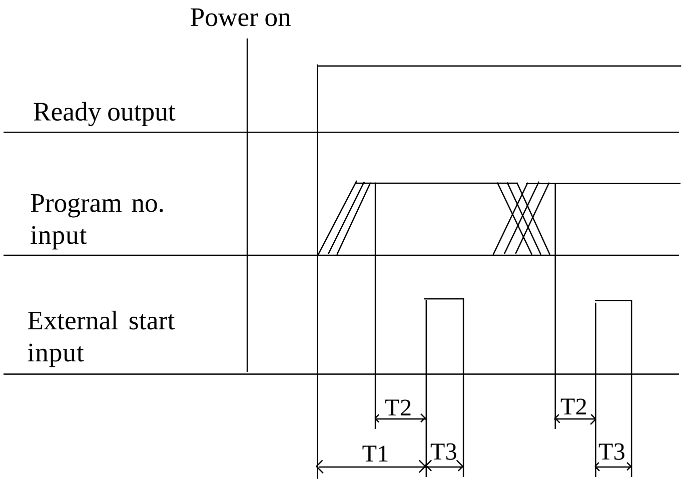


Program No. can be input before the Ready prompt.
The BCD code is input at input signal numbers 8~14.

Selected Program No. will start.

Program No. will be displayed on the front panel of the controller and the [AUTO] mode LED will light up.

② Timing Chart



T1: Time from ready output ON to start input
Min. 30msec

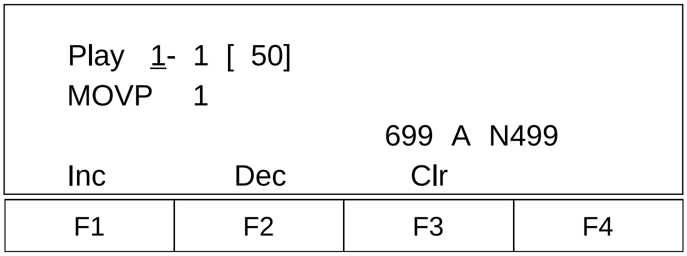
T2: Time from program No. input to external start input
Min. 30msec

T3: External start input
Min. 30msec

3. Teaching Pendant Operating Mode

Program is executed using the Teaching Pendant Play Mode.

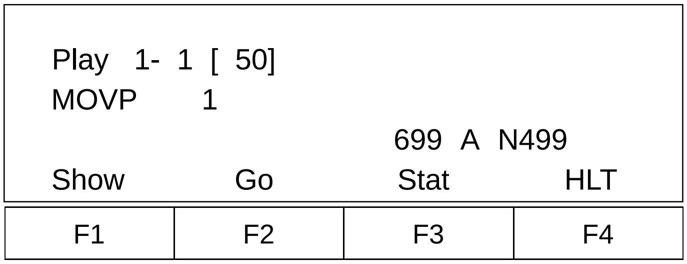
Play Mode



```
Play   1-  1  [  50]
MOVP    1
                    699  A  N499
Inc          Dec          Clr

F1           F2           F3           F4
```

Play Mode (Operation)
1. Play Program Input Mode

Input the program to be executed (stopped).



```
Play   1-  1  [  50]
MOVP       1
                    699  A  N499
Show         Go          Stat          HLT

F1           F2           F3           F4
```

2. Program Execution or Program Stop Selection

Designate program execution or stop.
F1 - (Show) Supervise (when the designated program is already executing.)
F2 - (Go) Execution
F3 - (Status) Display status. (Program can be executed or stopped after checking the status.)
F4 - (Hlt) Stop the program.
(Press F2 to execute a designated program)

4. Dedicated Input/Output Signal

(1) Dedicated Inputs

① External Start — This is the signal for external start. The port for this signal is (❍❍❍).
Reads the program number and starts the program.

② Program Selection PRG 1~ — Selects the program number for external start using BCD code input.
External start signal is input and can be used as a user input after the program starts.
Ports 8~14 are used for this.

③ Emergency Stop — Emergency Stop Signal
Servos and all user outputs turn OFF and alarm output turns ON.
This signal is Port 002.

④ System Reserve — This should **never** be used as a User Input.

(2) Dedicated Outputs

① Alarm Output — Turns ON in case of Emergency Stop input or when an alarm occurs (Error Code A?). This signal is Port 300.
↑—— another letter appears here.

② Ready Output — Turns ON after a few seconds when power is turned ON.
Turns OFF when alarm output is ON.
This signal is Port 301.

Note: These 2-dedicated outputs can be turned ON and OFF in the program.

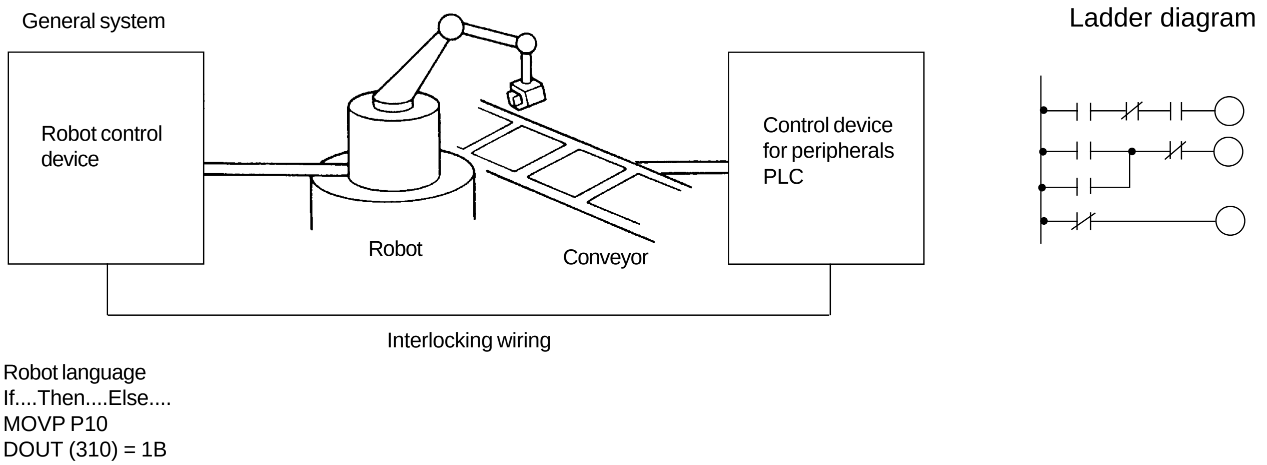
Chapter 3. Multi-tasking

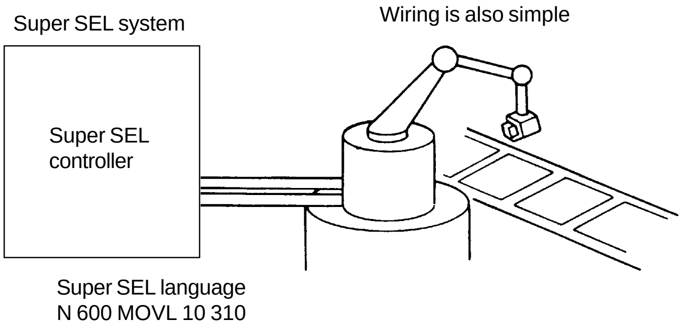
Part 1 Real-time Multi-tasking

1. Super SEL Programming Language

The Super SEL employs a 32 bit RISC CPU operated by a high speed operating system (OS). The Super SEL can control an entire system, including not only the actuators but also peripheral devices. The Super SEL enables the user to design an efficient automation system without learning various types of programming languages.

The original SEL language has been improved to what is now Super SEL language. In particular, the "Multi Tasking" feature allows for high speed control of multi programs and I/O. Programming is so simple that even non-technical users can write a parallel processing program.

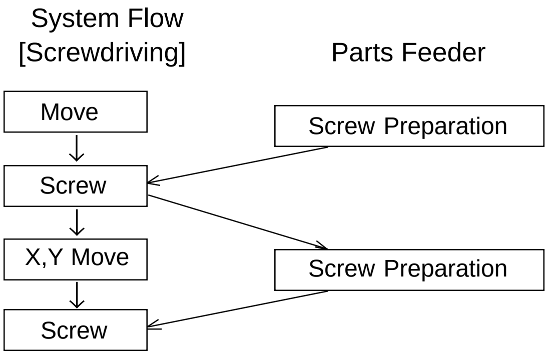




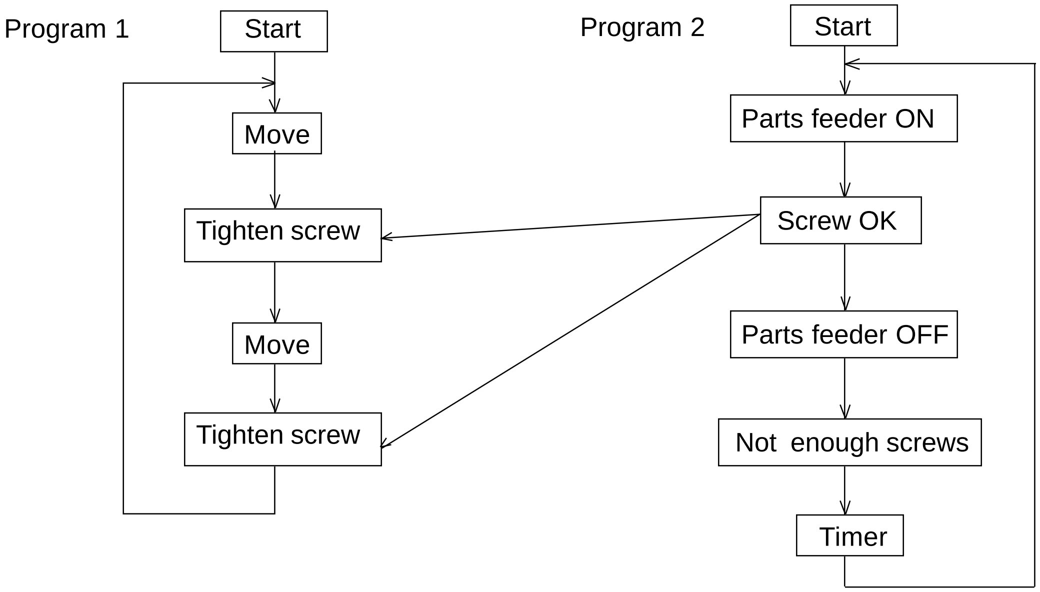
2. Multi-tasking

"Multi Tasking" simply means running multiple programs concurrently.

Let us consider a screwdriving robot system as an example. The following screwdriving system is composed of two actuators (X axis and Y axis) and a screwdriver with the part feeder.

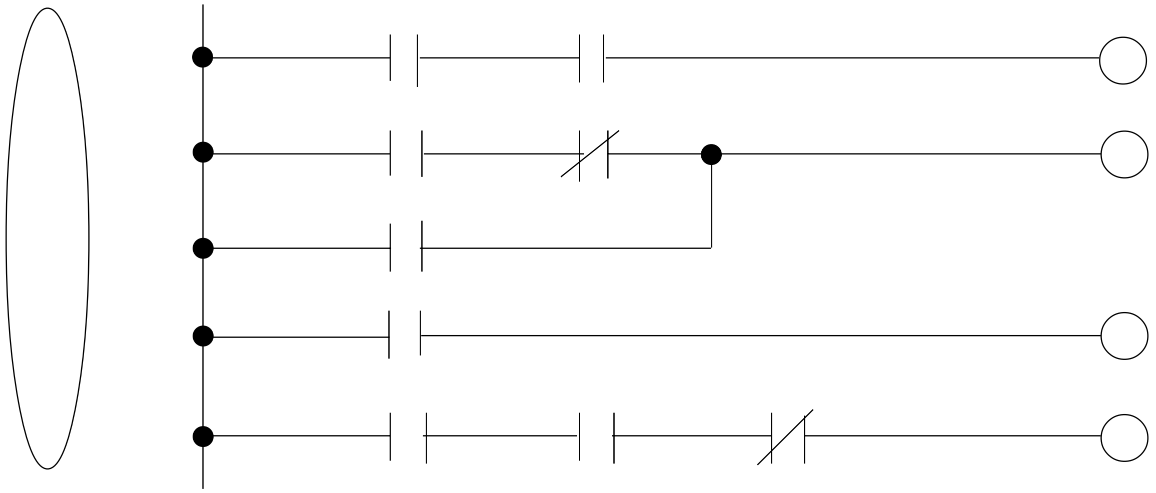


The following is a simple flowchart that shows the multi-tasking operation used when it is necessary to move a part feeder at the same time the XY actuator is operating.



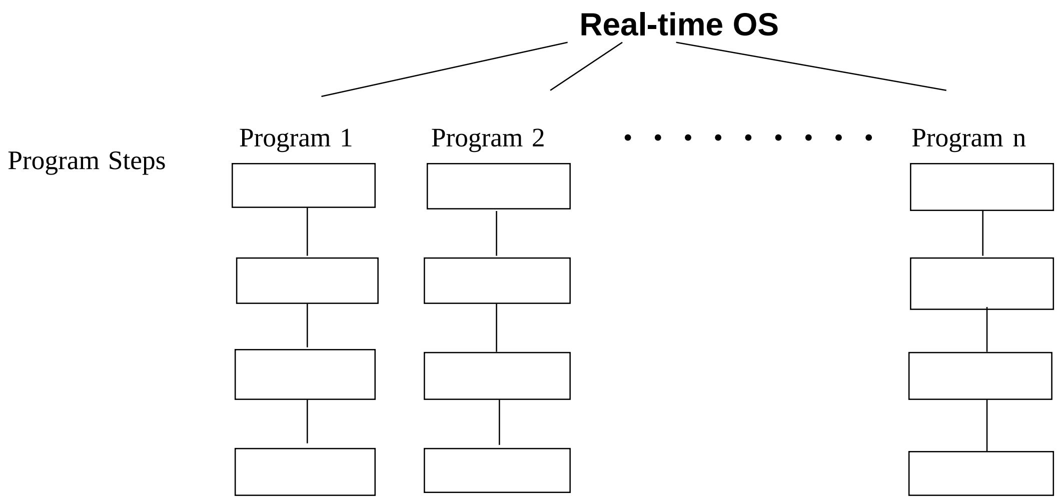
3. Difference Between the Super SEL and a PLC

In the past, parallel processing was accomplished by relay ladder circuitry. It was subsequently replaced by a PLC which is equipped with micro processors. There is considerable scanning time involved in PLCs. The following shows that a PLC scans the entire program, then sets the post if certain conditions are met.



The Super SEL operating system (OS) handles programs at a high speed and executes programs only when a certain event occurs (event driven), without scanning. Also, each parallel processing program is composed of simple one-line steps. This makes programming and debugging easy.

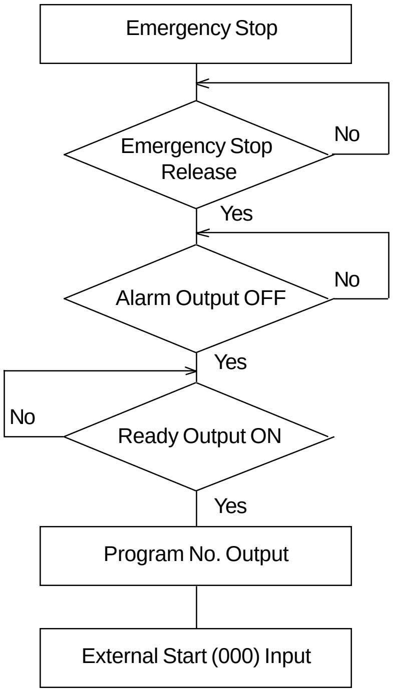
With the Super SEL controller, the user will be able to develop a multi tasking program without thinking about parallel processing, which is handled by the SEL operating system (OS).



4. Emergency Stop Release

Emergency stop is turned ON when emergency contact b is OFF, and emergency stop is released when ON.

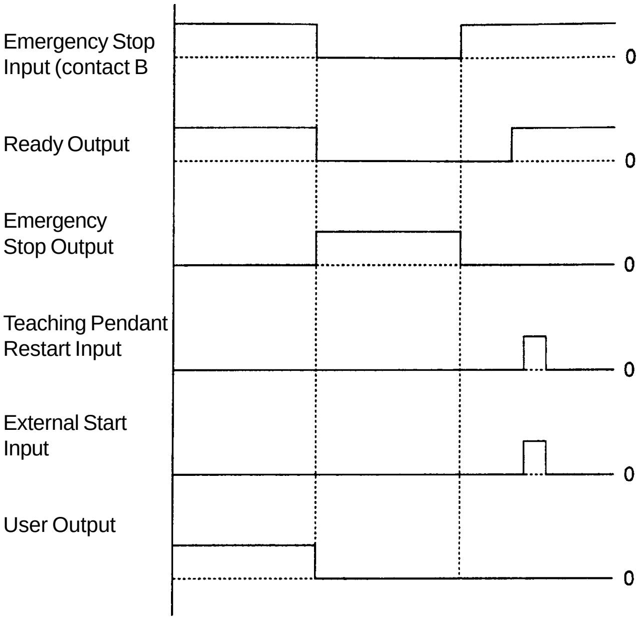
① Flow chart



Execute from step 1 in selected program

② Timing chart

Super SEL Controller Emergency Stop Release



◎When there is an emergency stop, the status inside of the controller is:

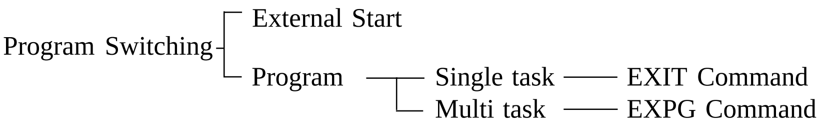
- •Program · · · · · · · · · · · · All stop
- •Output port, local flag local variable, home position } · · Clear
- •Global flag, global variable · · · Maintain

When peripheral devices are controlled by a program, the control program must be created beforehand. Control the peripheral devices using this program or perform other startup programs (EXPG) and aborting programs (ABPG) after checking the user input status.

*Emergency Stop output (Alarm Output)
Pin No.13B (Port No.300) outputs when there is an emergency stop output or when a servo related alarm A• occurs. (It does not output for every alarm).

5. Program Switching

There are several ways to switch programs depending on the program operating application but the typical methods are described below.



(1) External Start

Refer to Chapter 2 Part 3.
After the power is turned on, input a program No. and execute the designated program by external start input.
After the first program, another designated program can be executed by inputting the program No. and external start input.

(2) Program

- Single task
  Executing the EXIT command (program end) after each program, completes programs and returns the controller to the original status (the time power was turned ON). However, home position is maintained. Another program No. can be designated, and the program can be switched by external start input.

- Multi-task
  Execute the EXPG command (start up other program) within the management program and programs are executed one after another in parallel.

Chapter 3. Multi-tasking

Part 2 Screwdriving Robot System

This chapter explains the screwdriving robot system operated by the Super SEL Controller.

1. Components Used:

Automatic screwdriver with air cylinder (Z axis)
Screw feeder

X axis actuator:	Intelligent Actuator 60W, 300mm model
Y axis actuator:	Intelligent Actuator 60W, 300mm model
Controller:	Super SEL (Type G)

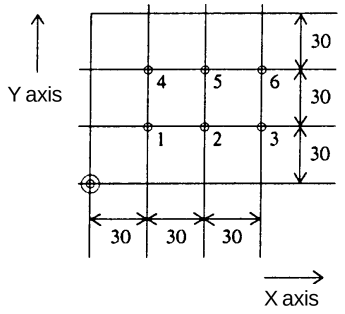
2. Operation:

(1) Screwdriving six screws each 30mm apart on both X and Y axes

① X and Y axes move to a screwdriving position.
② Z axis (air cylinder) goes down.
③ Begins screwdriving.
④ Completes screwdriving, then Z axis goes up.
⑤ X and Y axes move to the next position.

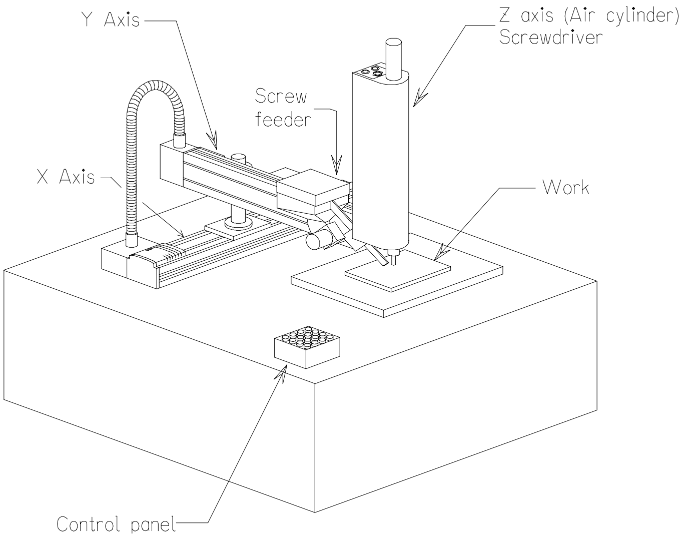
(2) The parts feeder moves simultaneously with the above operation.
① Screw feeder starts feeding when screws are depleted.
② Screw feeder stops feeding when screws are full.

Position Coordinates



3. Screwdriving System Illustration

The system consists of the X and Y actuators, automatic screwdriver and screw feeder. The screwdriver is positioned by the actuators, and screws are supplied by the feeder.



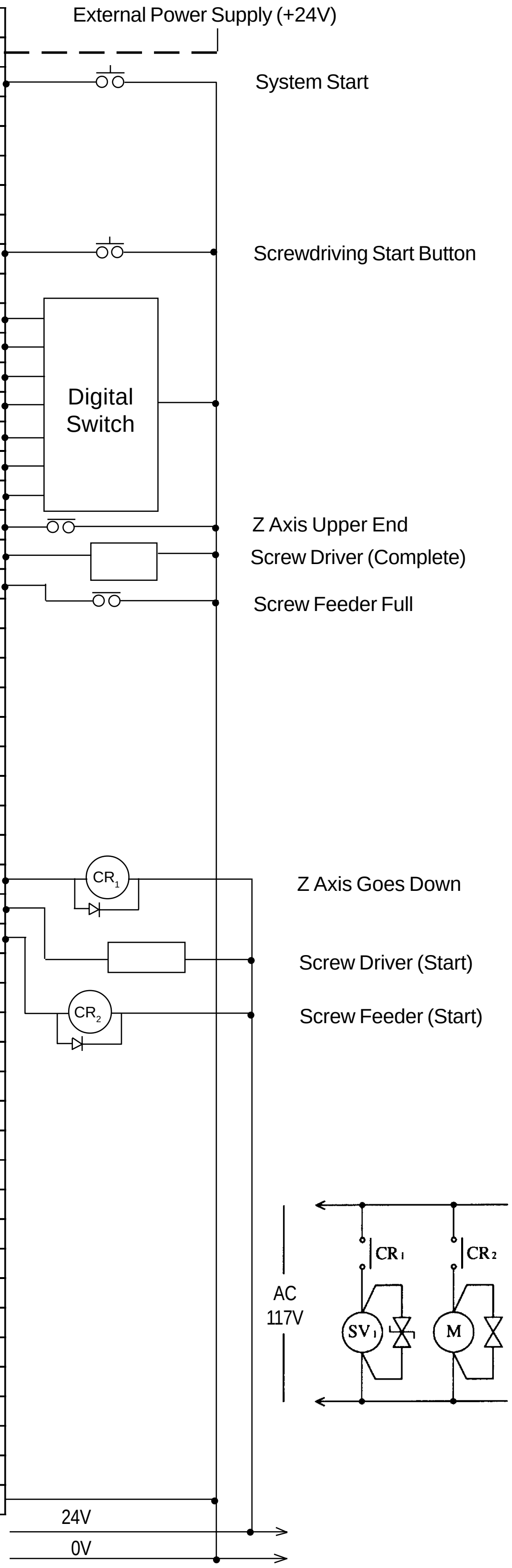
4. Hardware

(1) I/O Assignment

Pin No.	Category	Port No.	Function	Cable
1A	P24		---	1-Brown
1B	Input	000	External Start Input	1-Red
2A		001	User Input	1-Orange
2B		002	Emergency Stop b Contact Input	1-Yellow
3A		003	SystemReserve	1-Green
3B		004	SystemReserve	1-Blue
4A		005	User Input	1-Purple
4B		006	Screwdriving start	1-Gray
5A		007	Screwdriving end	1-White
5B		008	PRG No. 1 (User Input)	1-Black
6A		009	PRG No. 2 (User Input)	2-Brown
6B		010	PRG No. 4 (User Input)	2-Red
7A		011	PRG No. 8 (User Input)	2-Orange
7B		012	PRG No. 10 (User Input)	2-Yellow
8A		013	PRG No. 20 (User Input)	2-Green
8B		014	PRG No. 40 (User Input)	2-Blue
9A		015	Z-axis air cylinder up	2-Purple
9B		016	Screwdriving complete	2-Gray
10A		017	Parts feeder full	2-White
10B		018	User Input	2-Black
11A		019	User Input	3-Brown
11B		030	User Input	3-Red
12A		021	User Input	3-Orange
12B		022	User Input	3-Yellow
13A		023	User Input	3-Green
13B	Output	300	Emergency Stop/Alarm Output	3-Blue
14A		301	Ready Output	3-Purple
14B		302	User Output	3-Gray
15A		303	Z-axis air cylinder down	3-White
15B		304	Screwdriving start	3-Black
16A		305	Parts feeder start	4-Brown
16B		306	User Output	4-Red
17A		307	User Output	4-Orange
17B		308	User Output	4-Yellow
18A		309	User Output	4-Green
18B		310	User Output	4-Blue
19A		311	User Output	4-Purple
19B		312	User Output	4-Gray
20A		313	User Output	4-White
20B		314	User Output	4-Black
21A		315	User Output	5-Brown
21B		316	User Output	5-Red
22A		317	User Output	5-Orange
22B		318	User Output	5-Yellow
23A		319	User Output	5-Green
23B		320	User Output	5-Blue
24A		321	User Output	5-Purple
24B		322	User Output	5-Gray
25A		323	User Output	5-White
25B	N24		---	5-Black

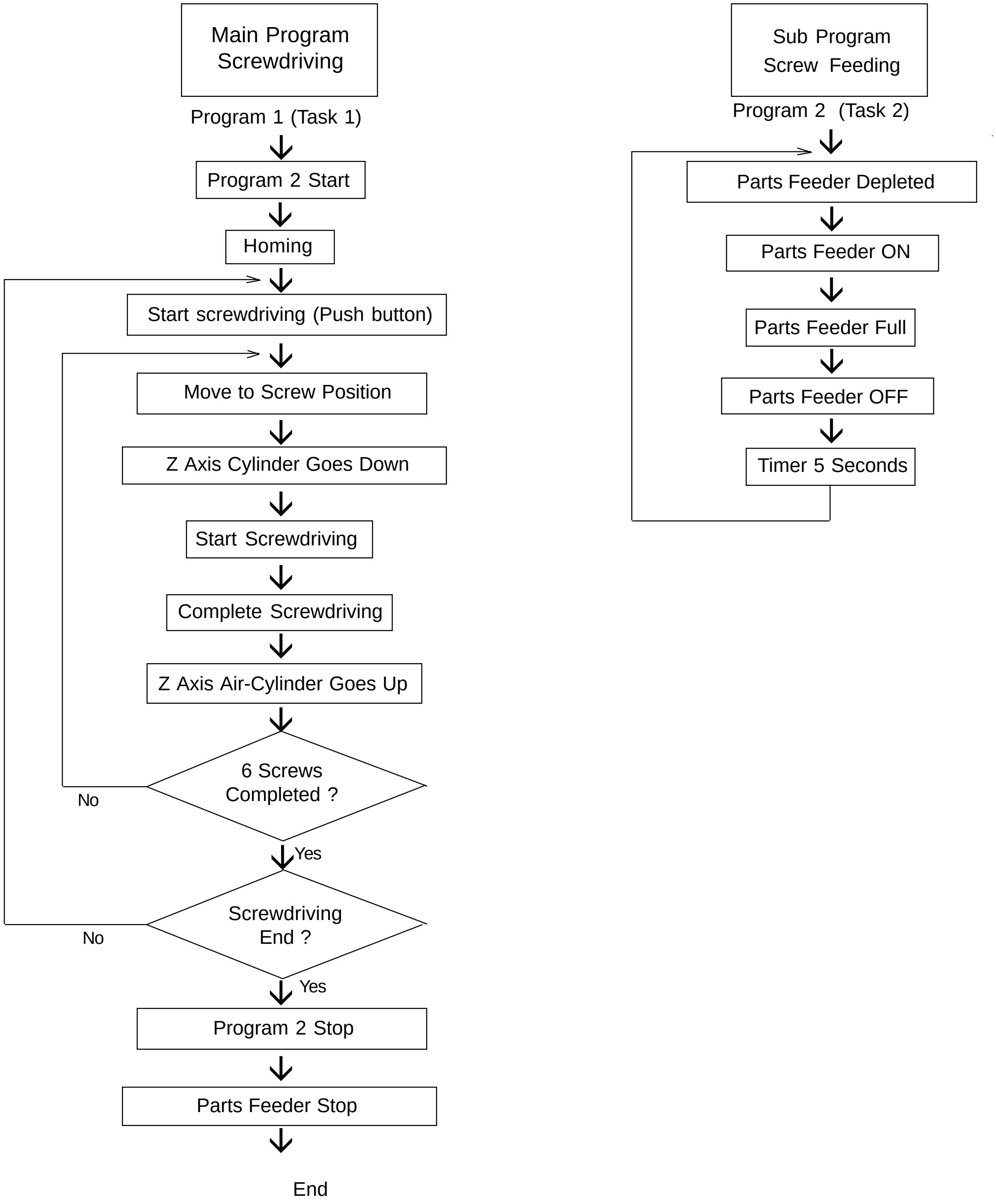
(2) Wiring

Pin No.	Category	Port No.	Function
1A	P24		---
1B	Input	000	External Start Input
2A		001	User Input
2B		002	Emergency Stop b Contact Input
3A		003	SystemReserve
3B		004	SystemReserve
4A		005	User Input
4B		006	User Input
5A		007	User Input
5B		008	PRG No. 1 (User Input)
6A		009	PRG No. 2 (User Input)
6B		010	PRG No. 4 (User Input)
7A		011	PRG No. 8 (User Input)
7B		012	PRG No. 10 (User Input)
8A		013	PRG No. 20 (User Input)
8B		014	PRG No. 40 (User Input)
9A		015	User Input
9B		016	User Input
10A		017	User Input
10B		018	User Input
11A		019	User Input
11B		030	User Input
12A		021	User Input
12B		022	User Input
13A		023	User Input
13B	Output	300	Emergency Stop/Alarm Output
14A		301	Ready Output
14B		302	User Output
15A		303	User Output
15B		304	User Output
16A		305	User Output
16B		306	User Output
17A		307	User Output
17B		308	User Output
18A		309	User Output
18B		310	User Output
19A		311	User Output
19B		312	User Output
20A		313	User Output
20B		314	User Output
21A		315	User Output
21B		316	User Output
22A		317	User Output
22B		318	User Output
23A		319	User Output
23B		320	User Output
24A		321	User Output
24B		322	User Output
25A		323	User Output
25B	N24		---



5. Software

(1) Flowcharts



(2) Main Program

Screwdriving • Program No. 1

Application Program

Line	Expansion	Input	Commands			Output	Comment
	AND/OR	I/O·Flag	Command	Operand 1	Operand 2	Output port·Flag	
1			EXPG	2			Program 2 start
2			HOME	11			Homing
3			VEL	100			Velocity 100mm/sec
4			ACC	0.3			Acceleration 0.3G
5			TAG	1			Restart Jump target
6			WTON	6			Screwdriving start Push Button
7			LET	1	1		Screw counter set
8			TAG	2			1 screw complete Jump target
9			MOVL	*1			Move
10			BTON	303			Z axis air cylinder down
11			BTON	304			Screwdriving start
12			WTON	16			Screwdriving complete
13			BTOF	303	304		Air cylinder up, screwdriving stop
14			WTON	15			Z axis air cylinder upper edge check
15			ADD	1	1		Screw counter +1
16			CPEQ	1	7	900	6 screws complete Compare
17		N900	GOTO	2			1 screw complete To the next screwdriving
18		N007	GOTO	1			Screwdriving To restart
19			ABPG	2			Program 2 stop
20			BTOF	305			Parts feeder stop
21			EXIT				Program 1 end

Position Program

No.	X	Y
1	30	30
2	60	30
3	90	30
4	30	60
5	60	60
6	90	60

(3) Sub Program

Screw Feeding • Program No. 2

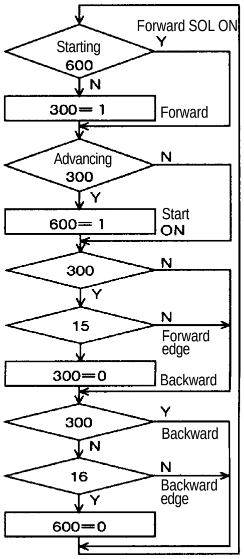
Application Program

Line	Expansion	Input	Commands			Output	Comment
	AND/OR	I/O·Flag	Command	Operand 1	Operand 2	Output port·Flag	
1			TAG	1			Repeated jump target
2			WTOF	17			Parts feeder depleted
3			BTON	305			Parts feeder start
4			WTON	17			Parts feeder full
5			BTOF	305			Parts feeder stop
6			TIMW	5			Restart Timer 5 seconds
7			GOTO	1			Repeat

Part 3 Multi-Tasking Programming Tips

Although multi-tasking methods are generally expressed as "the simultaneous execution of multiple programs (multi-tasks)," the programs are not actually carried out simultaneously. Rather, several programs are performed in sequence in a very short space of time using the free time available in each program. Avoid any program combination that uses up the controller CPU for one program. The most efficient configuration, making the multi-tasking function fully useful, will be shown later.

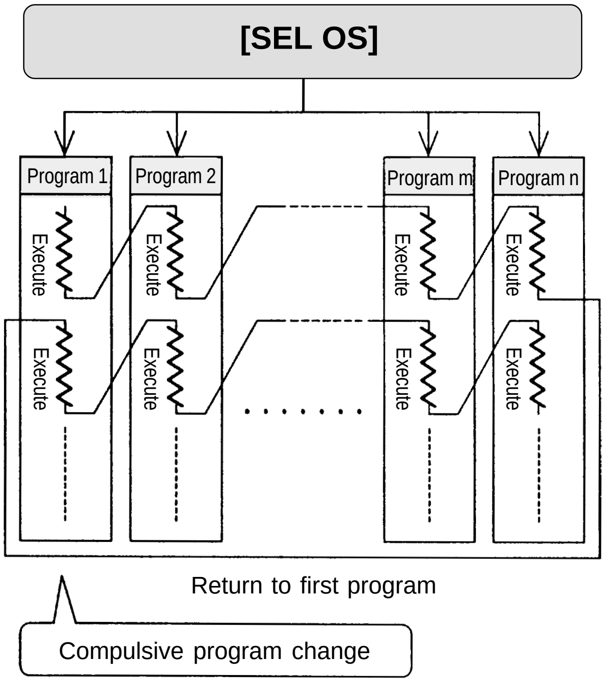
1. Inefficient Configuration



Program

```
                TAG     1
        N600
AND     N300    BTON    300
        N600
AND      300
AND       15    BTOF    300
        N300
AND       16    BTOF    600
                GOTO    1
```

Execution by constant scanning of the program.



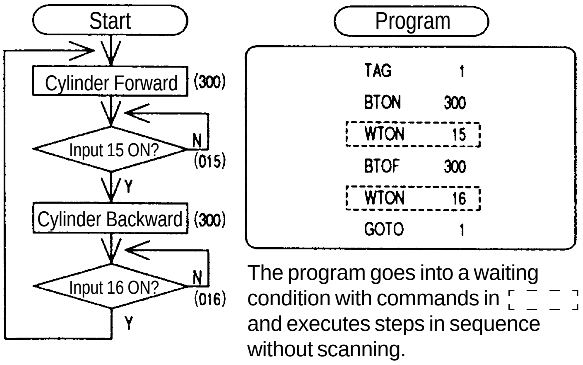
- Basically, in the multi-tasking operation, the controller system program [SEL OS] monitors each of the programs and when the program being executed goes into a waiting condition such as waiting for an input signal, the next program is executed.

- As shown at the left, all the programs are created to advance through the steps without waiting based on conditions alone so that they are continually scanning.

- In this case the program being executed does not have a waiting condition and uses up the CPU. Because of this, the controller system program [SEL OS] forces the programs that are running to switch in sequence.

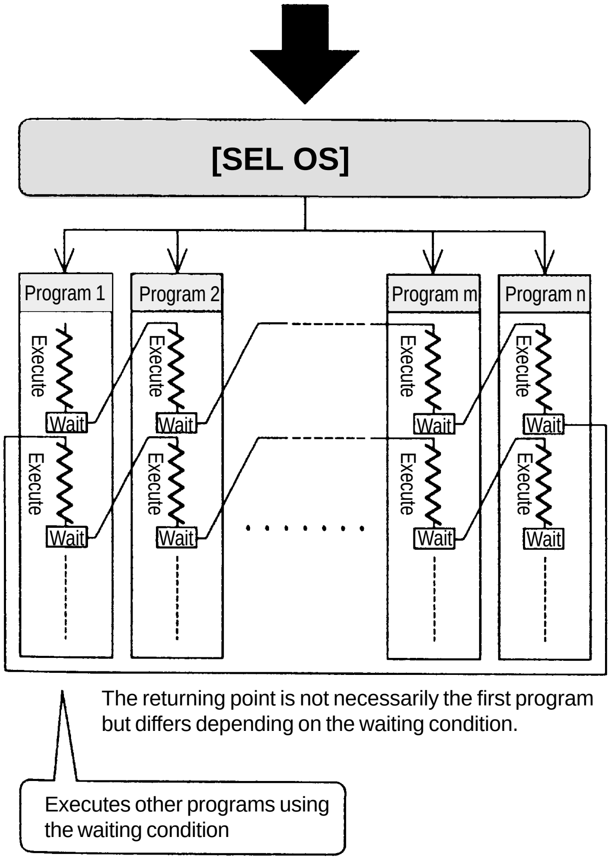
- Programs that are always scanning and using up the CPU do work but they are inefficient and this is not a true multi-tasking system. Try to avoid this kind of programming as much as possible.

2. Most Efficient Configuration



```
TAG     1
BTON    300
WTON    15
BTOF    300
WTON    16
GOTO    1
```

The program goes into a waiting condition with commands in ⌜ ⌟ and executes steps in sequence without scanning.



The returning point is not necessarily the first program but differs depending on the waiting condition.

Executes other programs using the waiting condition

- The most efficient configuration is to execute the next program during a waiting period such as waiting for an input signal.

- Use the WTON command (instead of conditional judgment) and have the steps executed in sequence rather than scanning the programs all the time.

- The WTON command keeps the program in a waiting condition until the designated condition turns ON.
  Commands to keep programs in waiting conditions are:
  WTON WTOF
  MOVP MOVL CIR ARC PATH
  TIMW

Chapter 4. Programming

Unlike other complicated robot languages, the Super SEL language is simple and easy to learn but can be used to develop highly sophisticated programs. Other languages that use BASIC are in "symbolic notation" and the "interpretation" of the language can be very time consuming. As programs become more and more complicated, interpreters are no longer capable of translating the languages within the time limit allowed. A compiler is added to solve this problem which causes MS-DOS to become extremely complex.* The new Super SEL interpreter can be used to develop highly complex programs without compilation. (*When executing a command, interpreters translate the command into computer language and execute it at the same time. Compilers translate the command into computer language before executing it.)

Part 1 Super SEL Language

1. Numerals and Symbols

The following numerals and symbols are used for super SEL programming.

Function	Global	Local	Note
Input Port	000 ~ 287 (288)		Varies according to actuator type
Output Port	300 ~ 587 (288)		Varies according to actuator type
Flag	600 ~ 887 (288)	900 ~ 999 (100)	
Variable (Integer)	200 ~ 299 (100)	1 ~ 99 (99)	Use INB, OUTB for 99
Variable (Real)	300 ~ 399 (100)	100 ~ 199 (100)	Use PPUT, PGET for 199
Column	300 ~ 399	1 ~ 299	
Tag No.		1 ~ 64 (64)	
Subroutine No.		1 ~ 64 (64)	
Axis Pattern	8 7 6 5 4 3 2 1		Varies according to actuator type
Axis No.	1 ~ 8		Varies according to actuator type
Position No.	1 ~ 2000		
Program No.	1 ~ 64		
Program Step No.	1 ~ 3000		
Channel	1 ~ 2		Varies according to actuator type
Timer	One in each program		
Position	±9999.999		
	Visible to all programs	Visible only within a program. (Local range clears at program startup)	Variables 99 and 199 are special variables used in computation. Avoid using these for general use.

- Battery Back-up Range

When the power is turned back ON, everything will be cleared except the area backed up by the battery.
(Same as emergency stop)

Program .... Stop
Output Port .... Clear
Local Flag .... Clear
Local Variable .... Clear
Home Position .... Clear
Global Flag .... Maintained
Global Variable .... Maintained

- Range of numerical values in the Super SEL

Super SEL uses two types of numbers, integers and real numbers but are subject to the following limitations.

1. Inside the controller
   The range of whole numbers that can be accommodated is ±2,147,483,648 and for real numbers the theoretical range is ±3.4 x $10^{38,}$ as a single precision floating point.
2. Limitations in actual use
   The programming tool developed initially was an LCD teaching pendant which resulted in certain constraints with respect to input and output from the program. The numerical values that can be handled from the program are -9,999,999~99,999,999 for integers and -999,999~9999,999 or -.999999~.999999, in other words an eight digit value including the decimal point sign for real numbers. Also, when doing floating point calculations, the significant figure can only be guaranteed up to 7 digits and it will include errors that are particular to floating points.
3. Position data
   Internally, position data is handled as whole number data but during the calculation process, these are incorporated into real numbers and treated as real numbers. There are no problems when dealing with numbers ±9999.999 but when these are internally calculated as general data and not position data (repeated multipliying and dividing), a problem arises with the accuracy of the last digit.
   When using the Super SEL, please pay close attention to these points. In particular, if you use the CPEQ command in a comparative calculation using real numbers, you will see almost no correlation. In this case, you will need to use the CPLE/OPGE command which can view the large and small relations in parallel.

2. Input/Output Port

(1) Input Port

Input ports are numbered from 001 ~ 95 and used for limit switches, sensor switches, etc. There are 24 inputs and 24 outputs for each I/O board.

Board No.	Input No. Assignment	Type E·G
1	001~023	Standard
2	024~047	Option *
3	048~071	Option *
4	072~095	Option *

(2) Output Port

Output ports are numbered from 300 ~ 587. There are 24 inputs and 24 outputs for each I/O board.

Board No.	Input No. Assignment	Type E·G
1	300~323	Standard
2	324~347	Option *
3	348~371	Option *
4	372~395	Option *

*Varies depending on the number of axes and motor wattage. Please refer to Part 6 Specifications in Chapter 1 Setting Up for details.

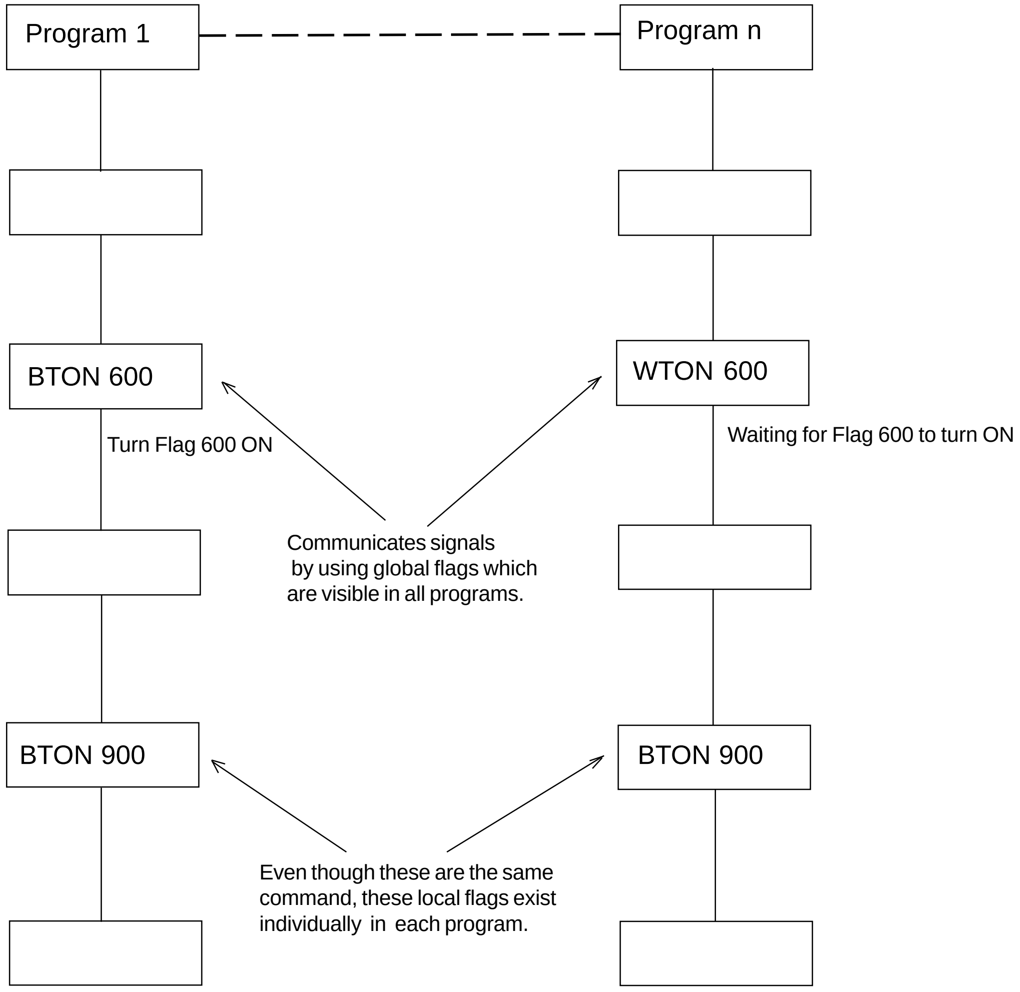
3. Flags

The function of flags is to set and reset data within "Memory." This is analagous to "internal relays" or "coils" in a PLC.

In general, there are two (2) types of flags: Global flags 600 ~ 887 which can be used in all programs and local flags 900 ~ 999 which can be used *only* in individual programs.

Global flags can be saved when the power is turned OFF (battery backup). Local flags are erased when the power is turned OFF.

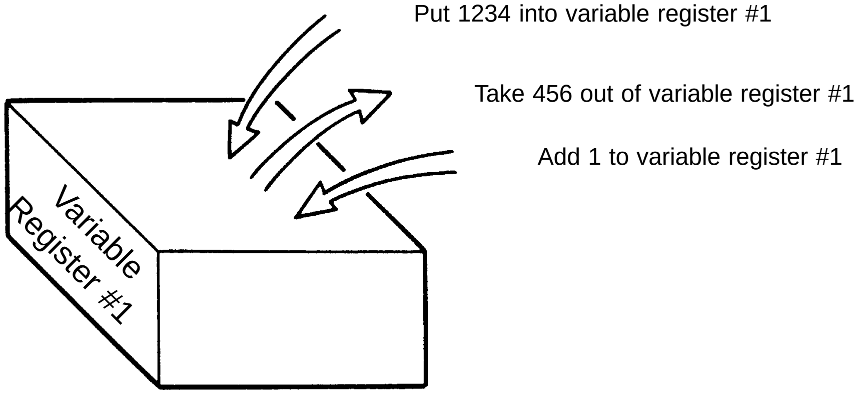
Flag Number	600~887	Global flag: Can be used in all programs
Flag Number	900~999	Local flag: Can be used only within an individual program


Program 1
Program n
BTON 600
WTON 600
Turn Flag 600 ON
Waiting for Flag 600 to turn ON
Communicates signals by using global flags which are visible in all programs.
BTON 900
BTON 900
Even though these are the same command, these local flags exist individually in each program.

4. Variable Register

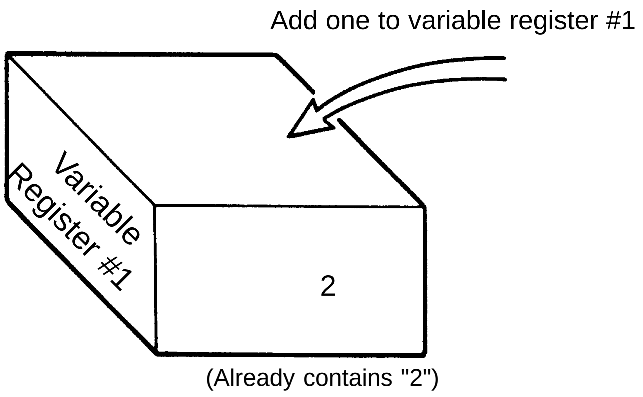
(1) What are Variables?

The term "variable register" is a software term. Imagine a box that holds numbers. Numbers can be put in and taken out, added, subtracted, and so on.


Put 1234 into variable register #1
Take 456 out of variable register #1
Add 1 to variable register #1
Variable Register #1

Command	Operand 1	Operand 2
Add	1	1

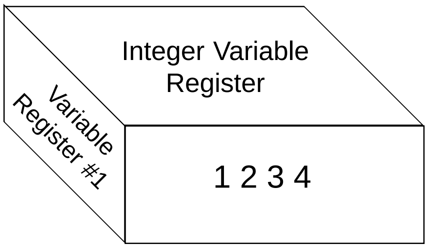
This command adds 1 to variable register #1. If the register contains 2, then the variable becomes 3.



(2) Types of Variables

There are two types of variables.

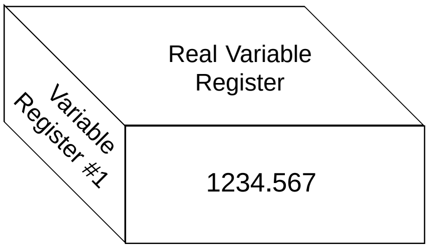
① Integer variable
These are whole number variables which cannot take decimal points. For example: [-2, -1, 0, 1, 2, 3]



Integer Variable No.	200~299	Global integer variable: Can be used in all programs
Integer Variable No.	1~99	Local integer variable: Can be used only within an individual program

Note: The variable 99 is a special register for whole integer calculation.
The numbers that can be input in the program are -9,999,999 to 99,999,999.

② Real variable
These are variables that can accommodate the actual value exactly as it appears, including digits following the decimal point. [Example: 1234.567]



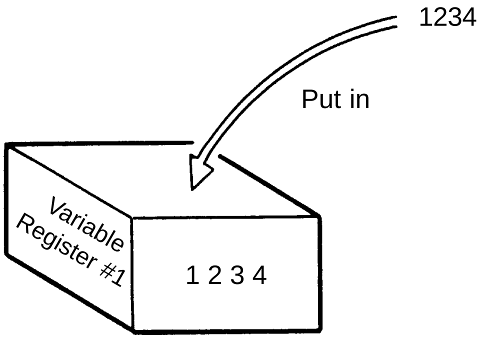
Real Variable No.	300~399	Global real variable: Can be used in all programs
Real Variable No.	100~199	Local real variable: Can be used only within an individual program

Note: The variable 199 is a special register for real number calculation.
The numbers that can be input in the program are -99,999.9 to 999,999.99 (8 digits which includes the decimal point sign).

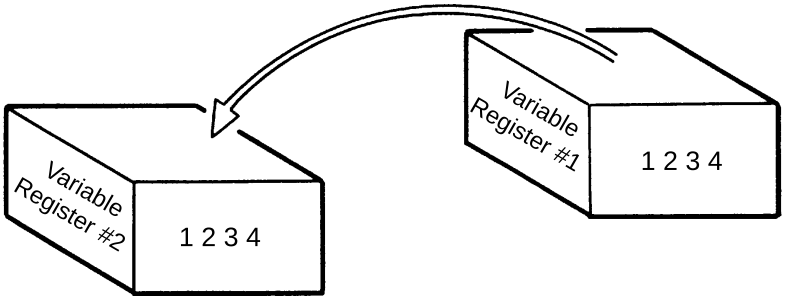
③ Variables with an asterisk (*)

The asterisk symbol (*) is used to designate contents of the variable register. In the example given below, the contents in variable register 1 are placed in variable register 2. If "1234" is in variable register 1, then "1234" is what goes in variable register 2.

Command	Operand 1	Operand 2
LET	1	1234

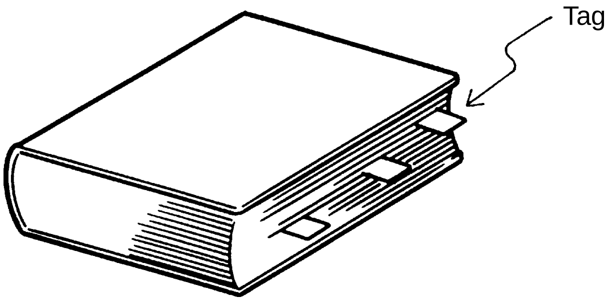


Command	Operand 1	Operand 2
LET	2	*1



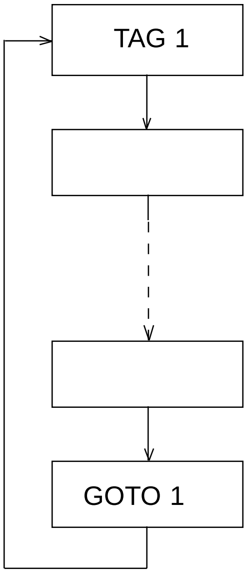
5. Tags

"Tag" means heading. A TAG can be thought of as the same as placing labels on important pages. The TAG as it is used in the Super SEL programming language is the "return to" area and is used in conjunction with the GOTO command to provide programming loops.



Command	Operand 1
TAG	Tag No. (Integers 1~64)

Can be used individually in each program.



6. Subroutine

Frequently repeated steps in a program can be expressed as subroutines in order to simplify the entire application program. These subroutines are individually usable in each program. (Up to a maximum of 15 subroutines can be nested)

Command	Operand 1
EXSR	Subroutine No. (1 ~ 64 Integers, or Variables)

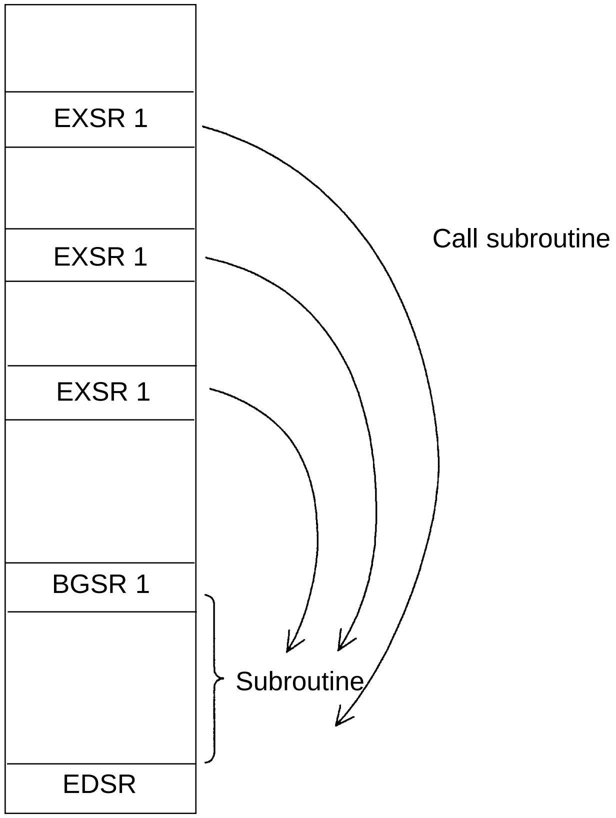
Execute subroutine command

Command	Operand 1
BGSR	Subroutine No. (1 ~ 64 Integers)

Begin subroutine command

Command	Operand 1
EDSR	--------

End subroutine command



7. Axis Designation

There are two ways to designate the axes to be used: axis number and axis pattern.

(1) Axis number and notation
With the Super SEL controller, multiple axes are indicated as shown in the table, but it is possible to change the figures using the parameters.

Axis No.	Default Notation
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8

The axis no. is used when designating one axis out of many axes.

Commands to designate Axis No. are:
BASE, PPUT, PGET

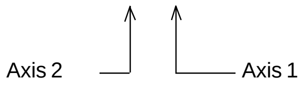
(2) Axis Pattern
Selection of an axis is specified by either "1" or "0"

Axis No.	8	7	6	5	4	3	2	1
Used	1	1	1	1	1	1	1	1
Not Used	0	0	0	0	0	0	0	0

Example

If Axis 1 and Axis 2 are in use, then this is signified by...

0 0 0 0 0 0 1 1

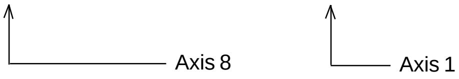


The zeroes before the 1 are unnecessary. The simplified form is 11, without leading zeroes.

Example

If Axis 1 and Axis 8 are in use, then this is signified by...

1 0 0 0 0 0 0 1



In this example, the zeroes *are* necessary in order to indicate the position of Axis 8.

Axis pattern is used when designating more than one axis at the same time.

Axis pattern designation command
OFST, GRP, SVON, SVOF, HOME, JFWN, JFWF, JBWN, JBWF, STOP, PTST, PRED

Part 2 Super SEL Language Structure

The Super SEL programming consists of a position and application program (command) section.

1. Position Program

In the position section, we have coordinates, velocity, acceleration, and variables.

1~1500 mm/sec

Standard
0.3G

±9999.999mm

Position No.	Velocity	Acceleration	Axis 1	Axis 2	Axis 3	Axis 4	Axis 5	Axis 6	Axis 7	Axis 8
1										
2										
3										
4										
1997										
1998										
1999										
2000										

2. Application Program (Commands)

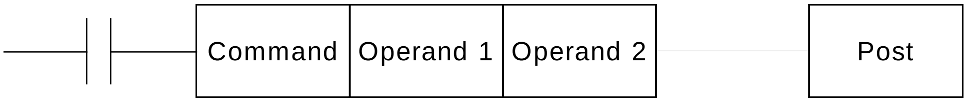
The outstanding feature of Super SEL language is the simplicity of its command structure which eliminates the need for a compiler and allows high speed operation with just an interpreter.

2-1 Structure of Super SEL language

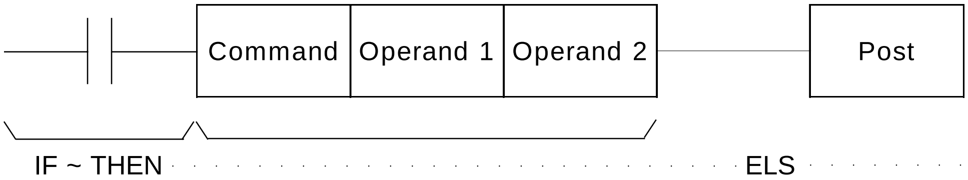
One step of the command has the following structure.

Expansion (AND · OR)	Input Condition (I/O · Flag)	Command			Post (Output port · Flag)
		Command	Operand 1	Operand 2	

Putting this in a ladder diagram, ⇩



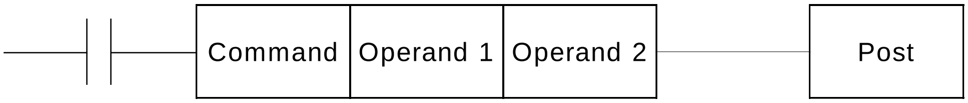
(1) The conditions before the commands are equivalent to "if ~ then" statements in BASIC language.



To the next step, ⇩

①Carry out a command when an input condition is established, and turn the output port ON, if output is designated. When not established, go on to the next step regardless of the next command (ex. WTON, WTOF). The designated output port remains the same , however it needs to be monitored carefully.
②If there is no conditioning set up, carry out command unconditionally.
③If condition is used as "negative condition", then place an "N" (NOT).
④Input/output port & flag can be used for condition.

(2) Output is set based on the result of the command execution.

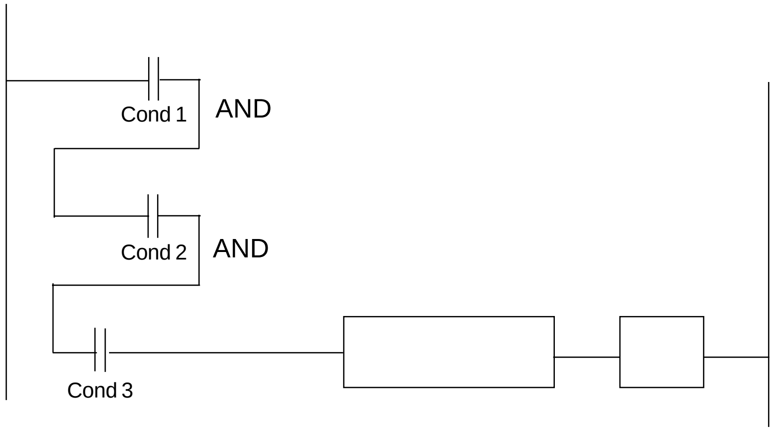


①Actuators motion control commands: becomes OFF immediately after the command starts to be executed, and becomes ON when the command is completed.
Computation commands: when the result becomes a certain value, it turns ON, and it stays OFF otherwise.
②Output port and flag can be used for output section.

2-2 Expansion Condition

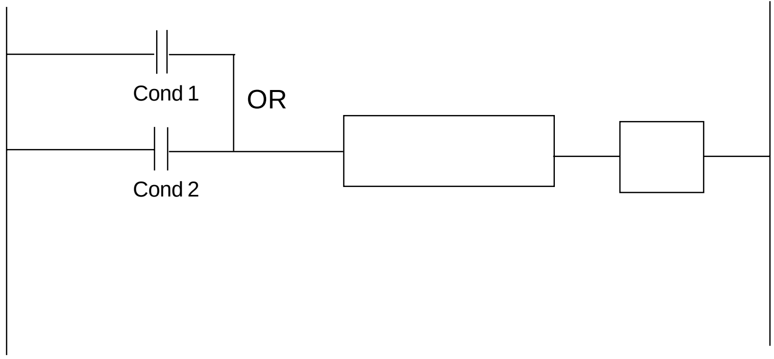
It is possible to combine conditions to make more complicated conditions as follows:

AND Expansion (Ladder Diagram display) (Super SEL Language)



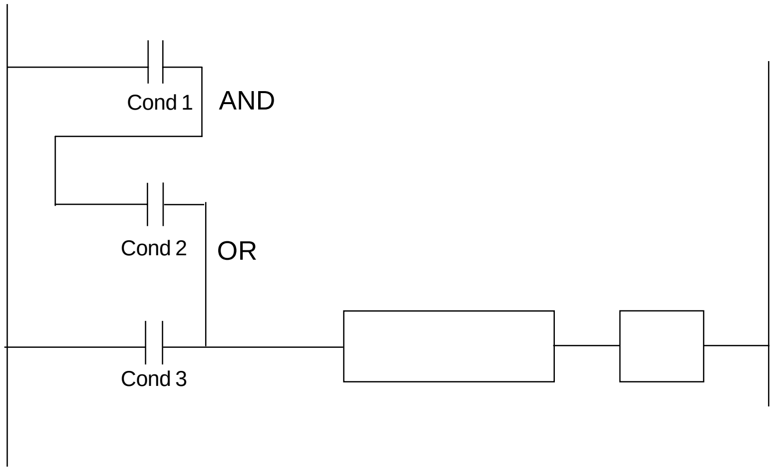
Expansion	Input	Commands			Output
		Command	Operand1	Operand2	
	Condition1				
AND	Condition 2				
AND	Condition 3	Command	Operand1	Operand2	

OR Expansion



Expansion	Input	Commands			Ouptput
		Command	Operand1	Operand2	
	Condition 1				
OR	Condition 2	Command	Operand 1	Operand 2	

AND/OR Expansion



Expansion	Input	Commands			Output
		Command	Operand1	Operand2	
	Condition1				
AND	Condition 2				
OR	Condition 3	Command	Operand1	Operand2	

Part 3 Standard Commands

1. Command Table

Category	Function	Command	Page
Actuator Control Designation	Set velocity	VEL	121
	Set acceleration	ACC	121
	Velocity scale factor	OVRD	122
	Designate offset	OFST	122
	Designate axis assignment	GRP	123
	Set arc degrees	DEG	123
	Designate axis temporary stop port	HOLD	124
	Designate axis stop port	CANC	125
	Designate axis base	BASE	125
Actuator Control Commands	Designated axis servo ON	SVON	126
	Designated axis servo OFF	SVOF	126
	Designated axis execute homing	HOME	127
	Designated group axes point move	MOVP	127
	Designated group axes point move	MOVL	128
	Circular movement	CIR	128
	PATH movement	PATH	129
	Designated axis jog forward while ON	JFWN	130
	Designated axis jog forward while OFF	JFWF	130
	Designated axis jog backward while ON	JBWN	131
	Designated axis jog backward while OFF	JBWF	131
	Stop movement of designated axis	STOP	132
	Arch motion trigger	ATRG	133
Input/Output Flag Operation Commands	Designated output port flag ON	BTON	134
	Designated output port flag OFF	BTOF	134
	Designated output port flag reverse	BTNT	135
	Designated output port flag ON waiting	WTON	135
	Designated output port flag OFF waiting	WTOF	136
	Reading binary value	IN	136
	Read BCD value	INB	137
	Binary output	OUT	137
	BCD output	OUTB	138
Timer Commands	Wait timer	TIMW	139
	Reset task timer	TIMC	139

Category	Function	Command	Page
Program Control Commands	Program end	EXIT	140
	Start task	EXPG	140
	Abort task	ABPG	141
Tag Command Designation	Jump	GOTO	142
	Designate tag for jump	TAG	142
Subroutine Control Designation Commands	Start designated subroutine	BGSR	143
	End subroutine	EDSR	143
	Call (execute) subroutine	EXSR	144
Calculation Commands	Assign	LET	145
	Add	ADD	145
	Subtract	SUB	146
	Multiply	MULT	146
	Divide	DIV	147
	Modulate	MOD	147
	Clear designated variable range	CLR	148
Function Computation Commands	Sine	SIN	149
	Cosine	COS	149
	Tangent	TAN	150
	Arctangent	ATN	150
	Square root	SQR	151
Logic Commands	Bitwise AND	AND	152
	Bitwise OR	OR	152
	Bitwise EXOR	EOR	153
Comparison Commands	Equal	CPEQ	154
	Unequal	CPNE	154
	Greater than	CPGT	155
	Equal to or greater than	CPGE	155
	Equal to or less than	CPLE	156
	Less than	CPLT	156
Point Data Operation Commands	Assign point data to designated axis	PPUT	157
	Read point data of designated axis	PGET	157
	Check for data in designated axis	PTST	158
	Copy point data	PCPY	158
	Clear point data	PCLR	159
	Read designated point data	PRED	159
	Point data range	PSIZ	160
	Assign velocity data	PVEL	160
	Assign acceleration data	PACC	161

2. Commands

2-1 Actuator Control Commands

VEL (Velocity)

Expansion condition (AND · OR)	Input condition (I/O · Flag)	Command			Post (Output port · Flag)
		Command	Operand 1	Operand 2	
Optional	Optional	VEL	Velocity		

[Function] Sets velocity of actuator movement in mm/sec. Maximum velocity varies according to the model of actuator so please set below that value.
*Decimal places cannot be used. Entering a decimal value will cause an error.
*The minimum velocity setting is 1mm/sec.

[Example] VEL 1000
1000mm/sec (Velocity Setting)

```
VEL1000
MOVP   1  ┐
MOVP   2  ┘  The velocity between these two points is 1000mm/sec.
VEL    500
MOVP   3  ┐
MOVP   4  ┘  The velocity between these two points is 500mm/sec.
```

ACC (Acceleration)

Expansion condition (AND · OR)	Input condition (I/O · Flag)	Command			Post (Output port · Flag)
		Command	Operand 1	Operand 2	
Optional	Optional	ACC	Acceleration		

[Function] Sets the acceleration of the actuator movement which is expressed in G (Gravity). Maximum acceleration varies and depends on the actuator model and payload. The rated acceleration is 0.3 G. The actuator moves at the rated acceleration 0.3 G when acceleration is not set by the ACC command.

[Example] ACC 0.3
0.3G (Acceleration setting at 0.3G)

OVRD (Override)

Expansion condition (AND · OR)	Input condition (I/O · Flag)	Command			Post (Output port · Flag)
		Command	Operand 1	Operand 2	
Optional	Optional	OVRD	Velocity ratio value		

[Function] This command decreases the velocity according to the designated ratio. (Velocity coefficient setting). The range of the ratio settings is from 1 ~ 100%.

*When you use the override function, any value below 1 will be clamped at 1. Any decimal value in the speed setting will be rounded off.

[Example]

VEL	100	100mm/sec setting
OVRD	50	100mm/sec is reduced by 50% and the actual velocity becomes 50mm/sec.

OFST (Offset)

Expansion condition (AND · OR)	Input condition (I/O · Flag)	Command			Post (Output port · Flag)
		Command	Operand 1	Operand 2	
Optional	Optional	OFST	Axis pattern	Offset value	

[Function] This command adds an offset value to the target value when the actuator moves. The offset amount is given in mm and the resolution is 0.001mm. Offset values can be negative numbers within the range of movement.

* The OFST command can only be used for the axes in that program. To set an offset value for axes in multiple programs, the OFST command must be executed for each program.

[Example] OFST 10000011 50.000

50mm is added to the movement amount of Axis 1, Axis 2, and Axis 8.

GRP (Grouping of Axes)

Expansion condition (AND · OR)	Input condition (I/O · Flag)	Command			Post (Output port · Flag)
		Command	Operand 1	Operand 2	
Optional	Optional	GRP	Axis pattern		

[Function] This command moves the actuator through the position data of the designated axis pattern. (Even if there is data in axes other than those designated, the actuator will not move to these positions).

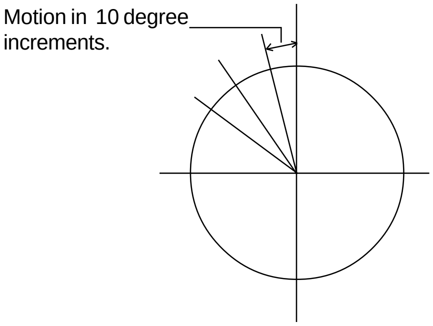
[Example] GRP 00000011
From position data for 8 axes, data from axis 1 and 2 is taken out and executed.

DEG (Degree Setting)

Expansion condition (AND · OR)	Input condition (I/O · Flag)	Command			Post (Output port · Flag)
		Command	Operand 1	Operand 2	
Optional	Optional	DEG	Angle		

[Function] This command sets up the motion increments for use with CIR (Circular Movement) and ARC (Arc Movement) commands. When performing CIR and ARC commands, passing points will be calculated by dividing a circle into the degrees as set. When increments are set small, the circular movement is accurate, however, when they are too small, the speed becomes too slow. When CIR and ARC commands are performed without setting increments, the motion increments of the actuator will be 15 degrees.

[Example] DEG 10

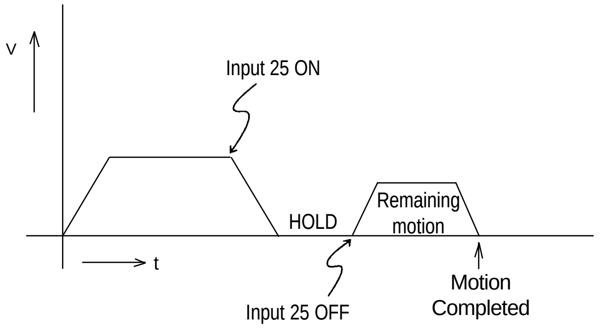


HOLD (Hold : Axis Temporary Stop)

Expansion condition (AND · OR)	Input condition (I/O · Flag)	Command			Post (Output port · Flag)
		Command	Operand 1	Operand 2	
Optional	Optional	HOLD	Input port		

[Function] Designates an input port for sending a command to decelerate and stop while a move command is being executed. If the designated input port turns ON, then velocity decreases until all motion stops. When the input port turns OFF, then motion begins again. The HOLD command applies only to the axes in the designated task (program), and does not affect axes running in other programs.

[Example] HOLD 25
When input port 25 turns ON, velocity decreases until all motion stops.



* When the HOLD function is used during a PATH motion command, the actuator stops at the next position. During the execution of straight line motion commands such as MOVL, MOVP, and circular/arc motion commands such as CIR, ARC, it stops immediately.

* The IA system uses a unique homing sequence which locks the servo and detects the stroke edge during homing. If the HOLD is activated at the end of homing, this might cause a "servo run-away = alarm" after the HOLD is released. Therefore, HOLD should be designated after the HOME command. If you need to designate HOLD from the beginning, a home area detection switch (an area limit switch) must be installed so that the HOLD designation will not be carried out in this area.

* HOLD and CANC cannot be used in the same program. (If both are written in the same program, the command that is designated later is the one that becomes effective).

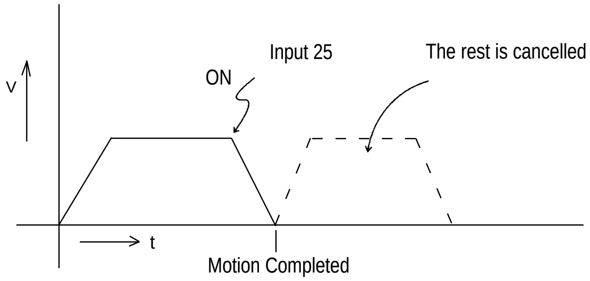
CANC (Cancel : Cancelling the next steps after axis stop motion)

Expansion condition (AND · OR)	Input condition (I/O · Flag)	Command			Post (Output port · Flag)
		Command	Operand 1	Operand 2	
Optional	Optional	CANC	Input port		

[Function] Designates an input port for sending a command to decelerate and stop while a move command is being executed. If the actuator is moving and the designated input port turns ON, then velocity decreases until all motion stops. Any other programmed motion thereafter is cancelled and not executed.

* HOLD and CANC cannot be used in the same program. (If both are written in the same program, the command that is designated later is the one that becomes effective).

[Example] CANC 25
When input port 25 turns ON, velocity decreases until all motion stops. All motion after this is cancelled.



* During PATH motion designation, the actuator moves to the next position. Any other programmed motion thereafter is cancelled and not executed.

BASE (Axis Base Designation) (This command is not supported by the tabletop type TT-300.)

Expansion condition (AND · OR)	Input condition (I/O · Flag)	Command			Post (Output port · Flag)
		Command	Operand 1	Operand 2	
Optional	Optional	BASE	Axis no.		

[Function] Count axes starting with the designated axis as the first axis.

[Example]	HOME 11	Axis No.1 and Axis No.2 perform homing.
	BASE 3	Axis No.3 is counted as the first axis.
	HOME 11	Axis No.3 and Axis No.4 perform homing. After homing, Axis No.3~8 move by designating Axis No.1~6 (axis pattern and position data).

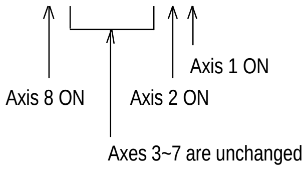
2-2 Actuator Control Commands

SVON (Servo ON)

Expansion condition (AND · OR)	Input condition (I/O · Flag)	Command			Post (Output port · Flag)
		Command	Operand 1	Operand 2	
Optional	Optional	SVON	Axis pattern		

[Function] This commands turns the servo of the designated axes ON.

[Example] SVON 1 0 0 0 0 0 1 1



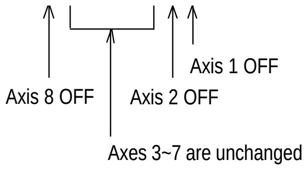
(Designating 0 does not turn the servo OFF but rather, leaves the servo already turned ON unchanged so that it stays ON.)

SVOF (Servo OFF)

Expansion condition (AND · OR)	Input condition (I/O · Flag)	Command			Post (Output port · Flag)
		Command	Operand 1	Operand 2	
Optional	Optional	SVOF	Axis pattern		

[Function] This commands turns the servos of the designated axes OFF.

[Example] SVON 1 0 0 0 0 0 1 1



(Designating 0 does not turn the servo ON but rather, leaves the servo already turned OFF unchanged so that it stays OFF.)

HOME (Return Home)

Expansion condition (AND · OR)	Input condition (I/O · Flag)	Command			Post (Output port · Flag)
		Command	Operand 1	Operand 2	
Optional	Optional	HOME	Axis pattern		Optional

[Function] This command executes homing of the designated axes. Servos turn ON automatically.

[Example] HOME 10000011
Axis 1, 2, and 8 axes execute homing.

MOVP (Point-to-point Position Data)

Expansion condition (AND · OR)	Input condition (I/O · Flag)	Command			Post (Output port · Flag)
		Command	Operand 1	Operand 2	
Optional	Optional	MOVP	Position no.		Optional

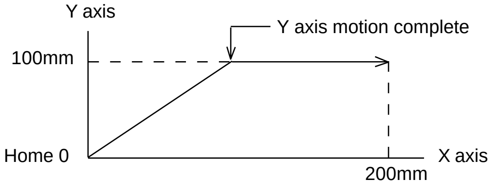
[Function] This command moves the actuator to the designated position number from point to point without interpolation.

[Example] MOVP 100
Moves to Position No. 100 (point to point).

MOVP *1
If variable 1 is 150, then the actuator moves to position number 150 (point to point).

Position No.	1	2	3
1			
2			
3			
· · ·			
100	100.00	100.00	xxx.xx
· · ·			
150	200.00	200.00	xxx.xx
· ·			

Move path when X-axis moves to 200mm point and Y-axis moves to 100mm point from home.



Each axis moves at its own designated speed.

MOVL (Position Data with Interpolation)

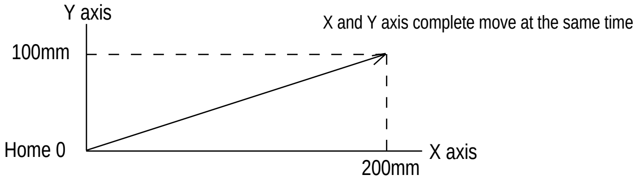
Expansion condition (AND · OR)	Input condition (I/O · Flag)	Command			Post (Output port · Flag)
		Command	Operand 1	Operand 2	
Optional	Optional	MOVL	Position No. Variable No.		Optional

[Function] Moves the actuator to the designated point while using interpolation (not point to point).

[Example] MOVL 100
Move to position No. 100 using interpolation.

MOVL *1
If variable 1 is 150, then the actuator moves to position 150 using interpolation.

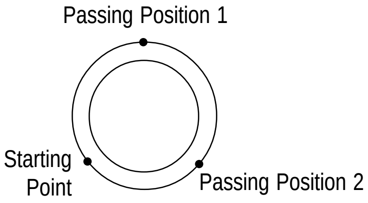
Move path when X-axis moves to 200mm point and
Y-axis moves to 100mm point from home.



The tip of the combined motion for each of the axes moves at the designated speed. The path from the starting point to the end point makes a straight line.

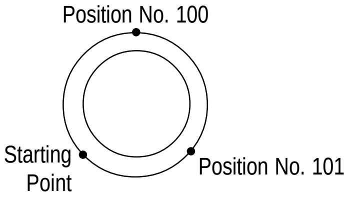
CIR (Circular Movement)

Expansion condition (AND · OR)	Input condition (I/O · Flag)	Command			Post (Output port · Flag)
		Command	Operand 1	Operand 2	
Optional	Optional	CIR	Passing position 1	Passing position 2	Optional



[Function] Executes circular motion using the current position as the starting point and passing points 1 and 2. The rotation direction is determined by the position data. The following diagram shows CW (clockwise) motion but this can be changed to CCW (counter-clockwise) by exchanging positions 1 and 2)

*This command is available for the specified orthogonal plane (Automatically selected by position data. Generally the XY plane is selected.) Also, care is needed if using this with the OFST command (check movement).



[Example] CIR 100 101
Executes a circular motion passing through position numbers 100 and 101.

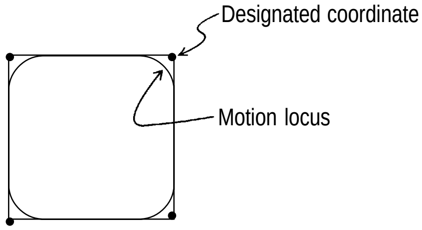
PATH (Path Movement)

Expansion condition (AND · OR)	Input condition (I/O · Flag)	Command			Post (Output port · Flag)
		Command	Operand 1	Operand 2	
Optional	Optional	PATH	Starting position no.	Ending position no.	Optional

[Function] Actuator moves continuously between the designated starting point and the finishing point. The locus is a B-spline-type, free-form curve which passes through the inside of the designated coordinate. It is possible for the actuator to move close to the designated coordinate by increasing the acceleration. However, when it exceeds the maximum acceleration, an error will occur.

* If HOLD is designated during the PATH command, the actuator stops at the next position, and later continues to perform the PATH motion without any problem except that the motion is less smooth at the joint section. When the distance between the position data is too long, the actuator will not stop immediately. If you need to stop the actuator immediately using HOLD, the distance between the position data must be short. If CANC is designated during PATH execution, the actuator moves to the next position (same as HOLD), ignores the rest of the PATH positions, and moves on to the next step.

* When the moving distance designated by the PATH command is too short, the actuator may not be able to reach the specified velocity. The velocity also decreases when the distance between the position data (PATH points) is too short. Therefore, we cannot guarantee the accuracy or quality of the linear speed (speed of the nozzle or other end-effectors) by the continuous locus control using the PATH command. (In general, the practical velocity of the continuous locus control is 100 ~ 200mm/sec.)



* It is very difficult to control an adhesive dispenser due to the viscosity of the adhesive and effects of temperature changes which will cause a delay in the actual dispensing time. It is necessary to move the dispenser, watch how the adhesive is applied and then adjust the dispensing start and end points. (In general, turn the dispenser ON slightly before the point where the adhesive is applied and turn it OFF slightly before the point where dispensing ends.)

* Three dimensional motion can be performed by this command.

[Example] PATH 100 120
Moves continuously to position 100 ~ 120.

JFWN (Jog Forward ON)

Expansion condition (AND · OR)	Input condition (I/O · Flag)	Command			Post (Output port · Flag)
		Command	Operand 1	Operand 2	
Optional	Optional	JFWN	Axis pattern	Input port · Flag	Optional

[Function] While the designated input port or flag (global flag) is ON, the axis moves forward.

*HOLD is not available for this command.

[Example] JFWN 10000011 25
When input port 25 is ON, then axis 1, axis 2, and axis 8 move forward.

JFWF (Jog Forward OFF)

Expansion condition (AND · OR)	Input condition (I/O · Flag)	Command			Post (Output port · Flag)
		Command	Operand 1	Operand 2	
Optional	Optional	JFWF	Axis pattern	Input port · Flag	Optional

[Function] While the designated input port or flag (global flag) is OFF, the axis moves forward.

* HOLD is not available for this command.

[Example] JFWF 10000011 25
When input 25 is OFF, then axis 1, axis 2, and axis 8 move forward.

JBWN (Jog Backward ON)

Expansion condition (AND · OR)	Input condition (I/O · Flag)	Command			Post (Output port · Flag)
		Command	Operand 1	Operand 2	
Optional	Optional	JBWN	Axis pattern	Input port · Flag	Optional

[Function] While the designated input port or flag (global flag) is ON, the axis moves backward.

* HOLD is not available for this command.

[Example] JBWN 10000011 25
When input port 25 is ON, then axis 1, axis 2 and axis 8 move backward.

JBWF (Jog Backward OFF)

Expansion condition (AND · OR)	Input condition (I/O · Flag)	Command			Post (Output port · Flag)
		Command	Operand 1	Operand 2	
Optional	Optional	JBWF	Axis pattern	Input port · Flag	Optional

[Function] While the designated input port or flag (global flag) is OFF, the axis moves backward.

* HOLD is not available for this command.

[Example] JBWF 10000011 25
When input port 25 is OFF, then axis 1, axis 2, and axis 8 move backward.

STOP (Stop Motion)

Expansion condition (AND · OR)	Input condition (I/O · Flag)	Command			Post (Output port · Flag)
		Command	Operand 1	Operand 2	
Optional	Optional	STOP	Axis pattern		

[Function] Stops the movement of axes in the main program and the other programs running in parallel, then proceeds to the next step. The STOP command is like the CANC command which can designate the axis but is executable from other programs.

[Example] STOP 10000011
Stops movement of axis 1, axis 2, and axis 8.

*This stops only the command that is currently being performed after which the program goes on to execute the next step. If you want the program [Example 1] to end after stopping the axis, rewrite the program as shown in [Example 2]. In this case, consider input signal 15 as the trigger signal for STOP.

```
[Example 1]                  [Example 2]
        MOVL 1                       MOVL 1
        MOVL 2           ⇨   15      EXIT
        MOVL 3                       MOVL 2
                             15      EXIT
                                     MOVL 3
                             15      EXIT
```

ATRG (Arch Motion Trigger)

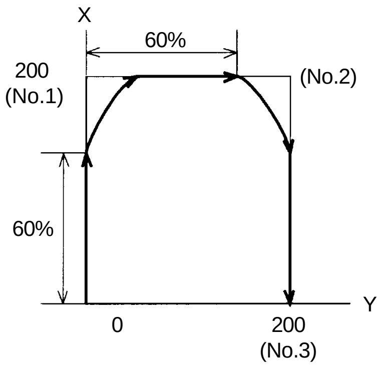
Expansion condition (AND · OR)	Input condition (I/O · Flag)	Command			Post (Output port · Flag)
		Command	Operand 1	Operand 2	
Optional	Optional	ATRG	Position ratio (%)		

[Function] Sets the axis movement position ratio to execute the ARCH command.

*The position ratio depends on the distance of the movement but it should be set at 50~60% or higher. When the ratio is set too low, a "C2" alarm may occur.

[Example] Application program

```
HOME   11
VEL    100
ATRG   60
ACC    0.3
ARCH   1   3
```



Position Data	X Axis	Y Axis
No. 1	200.000	xxx.xxx
No. 2	xxx.xxx	200.000
No. 3	0.00	xxx.xxx

2-3 I/O • Flag Operation Commands

BTON (Output Port · Flag ON)

Expansion condition (AND · OR)	Input condition (I/O · Flag)	Command			Post (Output port · Flag)
		Command	Operand 1	Operand 2	
Optional	Optional	BTON	Output port·Flag	Output port·Flag	

[Function] Turns the designated output port or flag ON.
Operand 1 and 2 can be used to designate a range of ports or flags.

[Example]	BTON	300		Output port 300 turns ON.
	BTON	300	310	Ouput port 300 ~ 310 turns ON.
	BTON	600		Flag 600 turns ON.
	BTON	600	610	Flag 600 ~ 610 turns ON.

BTOF (Output Port · Flag OFF)

Expansion condition (AND · OR)	Input condition (I/O · Flag)	Command			Post (Output port · Flag)
		Command	Operand 1	Operand 2	
Optional	Optional	BTOF	Output port·Flag	Output port·Flag	

[Function] Turns the designated output port or flag OFF.
Operand 1 and 2 can be used to designate a range of ports or flags.

[Example]	BTOF	300		Output port 300 turns OFF.
	BTOF	300	310	Output port 300 ~ 310 turns OFF.
	BTOF	600		Flag 600 turns OFF.
	BTOF	600	610	Flag 600 ~ 610 turns OFF.

BTNT (Output Port · Flag Invert)

Expansion condition (AND · OR)	Input condition (I/O · Flag)	Command			Post (Output port · Flag)
		Command	Operand 1	Operand 2	
Optional	Optional	BTNT	Output port· Flag	Output port· Flag	

[Function] Inverts the designated output port or flag. Operand 1 and 2 can be used to designate a range of ports or flags. When this command is executed, items that were ON change to OFF and items that were OFF change to ON.

[Example]	BTNT	300		Reverse output port 300.
	BTNT	300	310	Reverse output port 300 ~ 310.
	BTNT	600		Reverse flag 600.
	BTNT	600	610	Reverse flag 600 ~ 610.

WTON (Waiting Input/Output Port · Flag ON)

Expansion condition (AND · OR)	Input condition (I/O · Flag)	Command			Post (Output port · Flag)
		Command	Operand 1	Operand 2	
Optional	Optional	WTON	Input port · Flag	Time out	

[Function] Waits until the designated I/O port or flag turns ON and does not proceed to the next step. You can use the time out setting to set a time interval after which I/O waiting will be aborted. The time out setting is in units of seconds (0.01 ~ 99 sec). When there is a time out setting, the post section turns ON. When there is no time out setting, the post section is not available.

*Local flags cannot be used in Operand 1.

[Example]	WTON	25		Wait until input port 25 turns ON.
	WTON	25	10	Wait until input port 25 turns ON. Time out occurs after a 10 second interval and the program proceeds to the next step.

WTOF (Waiting for Input/Output Port · Flag OFF)

Expansion condition (AND · OR)	Input condition (I/O · Flag)	Command			Post (Output port · Flag)
		Command	Operand 1	Operand 2	
Optional	Optional	WTOF	Input port·Flag	Time out	

[Function] Waits until the designated I/O port or flag turns OFF and does not proceed to the next step. You can use the time out setting to set a time interval after which I/O waiting will be aborted. The time out setting is in units of seconds (0.01 ~ 99 sec). When there is a time out setting, the post section turns ON. When there is no time out setting, the post section is not available.

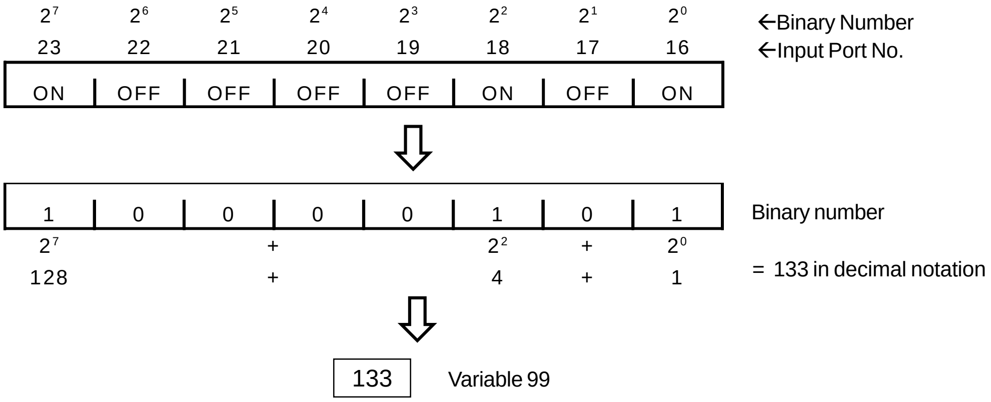
* Local flags cannot be used in Operand 1.

[Example]	WTOF	25		Wait until input port 25 turns OFF.
	WTOF	25	10	Wait until input port 25 turns OFF. Time out occurs after a 10 second interval and the program proceeds to the next step.

IN (Binary Numbers Read Input/Output)

Expansion condition (AND · OR)	Input condition (I/O · Flag)	Command			Post (Output port · Flag)
		Command	Operand 1	Operand 2	
Optional	Optional	IN	Start I/O port·Flag	Finish I/O port·Flag	

[Function] Reads the value from the designated I/O port or flag as a binary number, then stores this value in variable register 99.



2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	←Binary Number
23	22	21	20	19	18	17	16	←Input Port No.
ON	OFF	OFF	OFF	OFF	ON	OFF	ON	
1	0	0	0	0	1	0	1	Binary number
2^7			+		2^2	+	2^0	
128			+		4	+	1	= 133 in decimal notation

133 Variable 99

* The maximum input limit for the port is 31 consecutive bits.

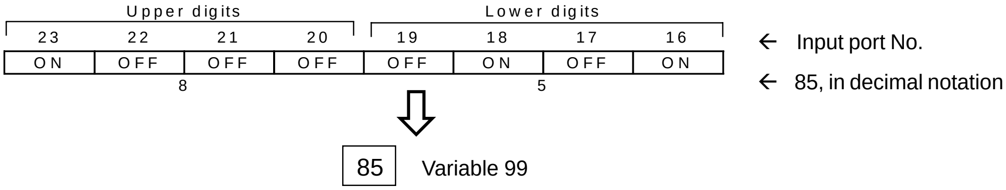
[Example] IN 16 23

Read the values from input ports 16 ~ 23 as a binary number and store in variable register 99.

INB (BCD Read Input/Output)

Expansion condition (AND · OR)	Input condition (I/O · Flag)	Command			Post (Output port · Flag)
		Command	Operand 1	Operand 2	
Optional	Optional	INB	Input port	No. of BCD digits	

[Function] Reads the BCD value from the designated input port, then stores this value in variable register 99.



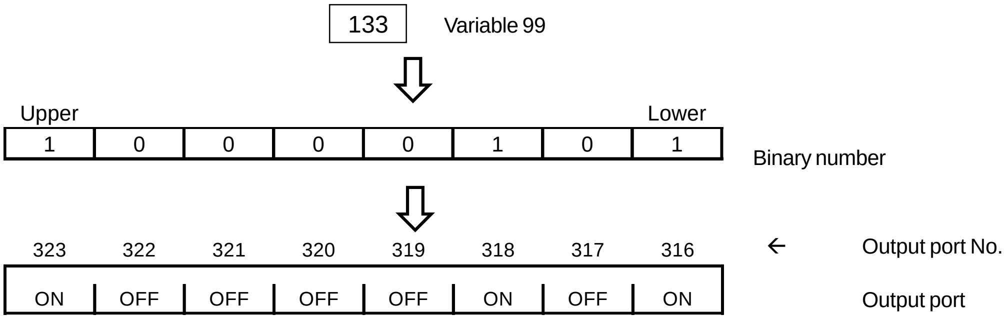
* The BCD value is read as 4 x n. (Every four input ports are read as one set) (n: no. of digits in Operand 2)
* The maximum number of digits that can be read is 8 but this requires a consecutive 32 point input port.

[Example] INB 16 2
From input port 16, read 4 x 2.

OUT (Binary Number Output)

Expansion condition (AND · OR)	Input condition (I/O · Flag)	Command			Post (Output port · Flag)
		Command	Operand 1	Operand 2	
Optional	Optional	OUT	Start output port· Flag	Finish output port· Flag	

[Function] Output the value in variable register 99 as a binary number to the designated output port or flag (range).



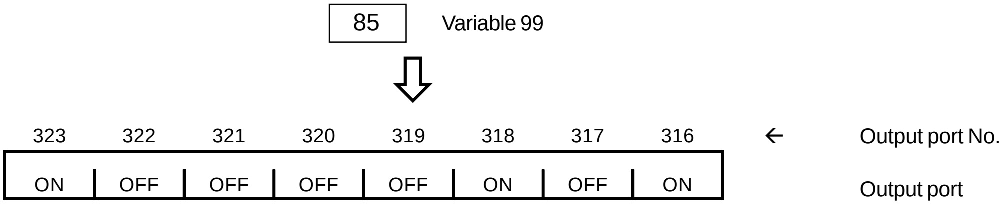
* The maximum output limit for the port is 31 consecutive bits.

[Example] OUT 316 323
Output the value of variable register 99 as a binary number to output ports 316 ~ 323.

OUTB (BCD Output)

Expansion condition (AND · OR)	Input condition (I/O · Flag)	Command			Post (Output port · Flag)
		Command	Operand 1	Operand 2	
Optional	Optional	OUTB	Start output port·Flag	No. of BCD digits	

[Function] Outputs the value in variable register 99 as a BCD value to the designated output port or flag.



323	322	321	320	319	318	317	316
ON	OFF	OFF	OFF	OFF	ON	OFF	ON

* The BCD value is read as 4 x n. (Every four input ports are read as one set) (n: no. of digits in Operand 2)
* The maximum number of digits that can be read is 8 but this requires a consecutive 32 point output port.

[Example] OUTB 316 2
From output port 316, output 4 x 2.

2-4 Timer Command

TIMW (Timer)

Expansion condition (AND · OR)	Input condition (I/O · Flag)	Command			Post (Output port · Flag)
		Command	Operand 1	Operand 2	
Optional	Optional	TIMW	Time		Optional

[Function] Sets a time interval during which the program waits to advance.
The unit for the time setting is in seconds (0.01 ~ 99 sec).

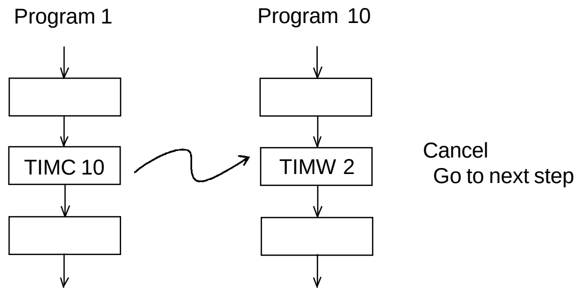
[Example] TIMW 1.5
Wait for 1.5 seconds.

TIMC (Timer Cancel)

Expansion condition (AND · OR)	Input condition (I/O · Flag)	Command			Post (Output port · Flag)
		Command	Operand 1	Operand 2	
Optional	Optional	TIMC	Program no.		

[Function] Cancels the timer of other concurrently running programs.

[Example] TIMC 10
Cancels the timer operating in program 10 and advances to the next step.



2-5 Program Control Commands

EXIT (Exit Program)

Expansion condition (AND · OR)	Input condition (I/O · Flag)	Command			Post (Output port · Flag)
		Command	Operand 1	Operand 2	
Optional	Optional	EXIT			

[Function] Finishes the program.

* The status when the program is complete
- Output Port .... Valid
- Local Flag .... Invalid
- Local Variable.............. Invalid
- Current Value .... Valid
- Global Flag .... Valid
- Global Variable .... Valid

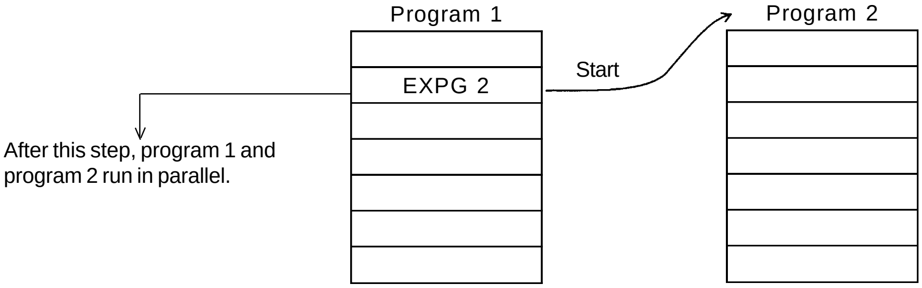
[Example] EXIT
Finish the program.

EXPG (Start another program)

Expansion condition (AND · OR)	Input condition (I/O · Flag)	Command			Post (Output port · Flag)
		Command	Operand 1	Operand 2	
Optional	Optional	EXPG	Program no.		Optional

[Function] Starts another program and processes it in parallel.
When that program (task) has been started, the port and flag in the post section is output.

[Example] EXPG 2
Start program No. 2 and process it in parallel.



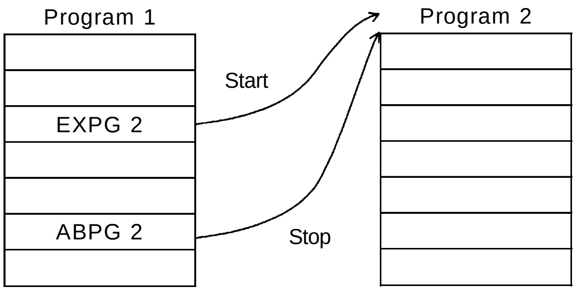
ABPG (Abandon Other Program)

Expansion condition (AND · OR)	Input condition (I/O · Flag)	Command			Post (Output port · Flag)
		Command	Operand 1	Operand 2	
Optional	Optional	ABPG	Program no.		Optional

[Function] Forces the other program being executed to end.
When that program (task) is forced to end, the port and flag in the post section is output.

* When ABPG is initiated while a motion command or timer command is being executed, the program ends after the motion or timer command is completed.

[Example] ABPG 2
Forces Program No.2, which is being executed in parallel to Program 1, to end.



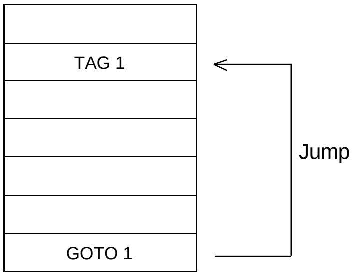
2-6 Turnout Commands

GOTO (Jump)

Expansion condition (AND · OR)	Input condition (I/O · Flag)	Command			Post (Output port · Flag)
		Command	Operand 1	Operand 2	
Optional	Optional	GOTO	Tag No.		

[Function] Jumps to the step designated by the tag number.
(Valid only within the same program)

[Example] GOTO 1
Jump to the step in Tag No. 1.

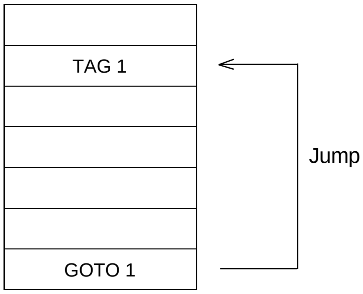


TAG (Tag Designation)

Expansion condition (AND · OR)	Input condition (I/O · Flag)	Command			Post (Output port · Flag)
		Command	Operand 1	Operand 2	
		TAG	Tag No.		

[Function] Uses the tag number to designate the place to jump to in the GOTO command.
(Valid only within the same program)

[Example] TAG 1
Prepares Tag 1 as the jump destination for GOTO 1.



2-7 Subroutine Control Commands

BGSR (Begin Subroutine)

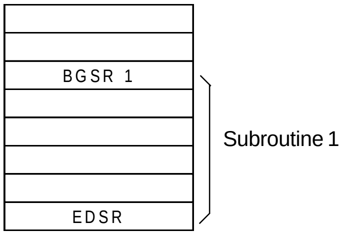
Expansion condition (AND · OR)	Input condition (I/O · Flag)	Command			Post (Output port · Flag)
		Command	Operand 1	Operand 2	
		BGSR	Subroutine No.		

[Function] Declares start of subroutine.

*Subroutines are generally used at the end of a program.

[Example] BGSR 1

Declare start of subroutine.



EDSR (End Subroutine)

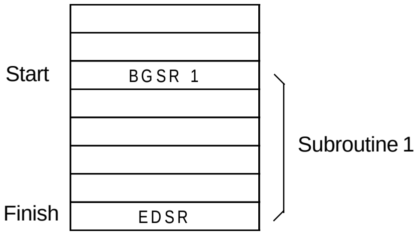
Expansion condition (AND · OR)	Input condition (I/O · Flag)	Command			Post (Output port · Flag)
		Command	Operand 1	Operand 2	
		EDSR			

[Function] Declares completion of subroutine.

*Subroutines are generally used at the end of a program.

[Example] EDSR

Declare completion of subroutine.
EDSR is always added at the end of a subroutine.

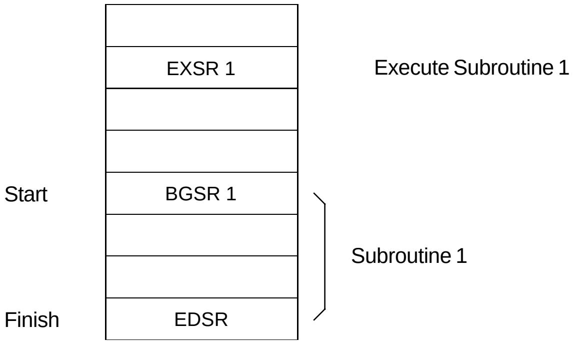


EXSR (Execute Subroutine)

Expansion condition (AND · OR)	Input condition (I/O · Flag)	Command			Post (Output port · Flag)
		Command	Operand 1	Operand 2	
Optional	Optional	EXSR	Subroutine No.		

[Function] Performs the designated subroutine number.
(Valid only within the same program)

[Example] EXSR 1
Perform subroutine 1



2-8 Calculation Commands

LET (Assignment)

Expansion condition (AND · OR)	Input condition (I/O · Flag)	Command			Post (Output port · Flag)
		Command	Operand 1	Operand 2	
Optional	Optional	LET	Variable No.	Data·Variable No.	Optional

[Function] Assigns data to the variable. When Operand 1 is 0, the post section turns ON.

[Example] LET 1 10

Assign a value of 10 to variable register 1.

LET 1 *2

Substitute variable 1 with contents of variable 2.

ADD (Addition)

Expansion condition (AND · OR)	Input condition (I/O · Flag)	Command			Post (Output port · Flag)
		Command	Operand 1	Operand 2	
Optional	Optional	ADD	Variable No.	Data·Variable No.	Optional

[Function] Adds contents of the variable in Operand 1 and the data in Operand 2, then stores this in the variable in Operand 1. Adds the contents of the variable in Operand 1 and the contents of the variable in Operand 2, then stores this in the variable in Operand 1 variable. If there is a post instruction (flag, output port), it turns ON when the computation result is zero.

[Example] ADD 1 10

If the content of variable 1 in Operand 1 is 10, then 10 (content of variable 1 in Operand 1) + 10 (data in Operand 2) = 20, and after the command is executed, the content of variable 1 becomes 20.

ADD 1 *2

If variable 1 in Operand 1 and variable *2 in Operand 2 both contain 10, then 10 (content of variable 1 in Operand 1) + 10 (contents of variable *2 in Operand 2) = 20, and after the command is executed, the content of variable 1 becomes 20.

SUB (Subtraction)

Expansion condition (AND · OR)	Input condition (I/O · Flag)	Command			Post (Output port · Flag)
		Command	Operand 1	Operand 2	
Optional	Optional	SUB	Variable no.	Data·Variable no.	Optional

[Function] Subtracts the data in Operand 2 from the variable in Operand 1, then stores this in the variable in Operand 1. Subtracts the variable in Operand 2 from the contents of the variable in Operand 1 and stores this in the variable in Operand 1. If there is a post instruction (flag, output port), it turns ON when the computation result is zero.

[Example] SUB 1 10

If the content of variable 1 in Operand 1 is 20, then 20 (content of variable 1 in Operand 1) - 10 (data in Operand 2) = 10, and after the command is executed, the content of variable 1 becomes 10.

SUB 1 *2

If variable 1 in Operand 1 is 20 and variable *2 in Operand 2 is 10, then 20 (content of variable 1 in Operand 1) - 10 (contents of variable *2 in Operand 2) = 10, and after the command is executed, the content of variable 1 becomes 10.

MULT (Multiplication)

Expansion condition (AND · OR)	Input condition (I/O · Flag)	Command			Post (Output port · Flag)
		Command	Operand 1	Operand 2	
Optional	Optional	MULT	Variable no.	Data·Variable no.	Optional

[Function] Multiplies the data in Operand 2 by the variable in Operand 1, then stores this in the variable in Operand 1. Multiplies the variable in Operand 2 by the contents of the variable in Operand 1 and stores this in the variable in Operand 1. If there is a post instruction (flag, output port), it turns ON when the computation result is zero.

[Example] MULT 1 10

If the content of variable 1 in Operand 1 is 10, then 10 (content of variable 1 in Operand 1) x 10 (data in Operand 2) = 100, and after the command is executed, the content of variable 1 becomes 100.

MULT 1 *2

If variable 1 in Operand 1 and variable *2 in Operand 2 both contain 10, then 10 (content of variable 1 in Operand 1) x 10 (contents of variable *2 in Operand 2) = 100, and after the command is executed, the content of variable 1 becomes 100.

DIV (Division)

Expansion condition (AND · OR)	Input condition (I/O · Flag)	Command			Post (Output port · Flag)
		Command	Operand 1	Operand 2	
Optional	Optional	DIV	Variable no.	Data·Variable no.	Optional

[Function] Divides the content of the variable in Operand 1 by the data in Operand 2, then stores this in the variable in Operand 1. Divides the content of the variable in Operand 1 by the contents of the variable in Operand 2 and stores this in the variable in Operand 1. If there is a post instruction (flag, output port), it turns ON when the computation result is zero.

[Example] DIV 1 5

If the content of variable 1 in Operand 1 is 10, then 10 (content of variable 1 in Operand 1) ÷ 5 (data in Operand 2) = 2, and after the command is executed, the content of variable 1 becomes 2.

DIV 1 *2

If the content of variable 1 in Operand 1 is 10 and the content of variable *2 in Operand 2 is 5, then 10 (content of variable 1 in Operand 1) ÷ 5 (contents of variable *2 in Operand 2) = 2, and after the command is executed, the content of variable 1 becomes 2.

MOD (Remainder)

Expansion condition (AND · OR)	Input condition (I/O · Flag)	Command			Post (Output port · Flag)
		Command	Operand 1	Operand 2	
Optional	Optional	MOD	Variable no.	Data·Variable no.	Optional

[Function] Divides the content of the variable in Operand 1 by the data in Operand 2, and stores the remainder in the variable in Operand 1. Divides the contents of the variable in Operand 1 by the contents of the variable in Operand 2 and stores the remainder in the variable in Operand 1. If there is a post instruction (flag, output port), it turns ON when the computation result (remainder) is zero.

* This command is executed with respect to integers.

[Example] MOD 1 3

If the content of variable 1 in Operand 1 is 10, then 10 (content of variable 1 in Operand 1) ÷ 3 (data in Operand 2) = 3 with a remainder of 1, and after the command is executed, the content of variable 1 becomes 1.

MOD 1 *2

If the content of variable 1 in Operand 1 is 10 and the content of variable *2 in Operand 2 is 3, then 10 (content of variable 1 in Operand 1) ÷ 3 (contents of variable *2 in Operand 2) = 3 with a remainder of 1, and after the command is executed, the content of variable 1 becomes 1.

CLR (Clear Variables)

Expansion condition (AND · OR)	Input condition (I/O · Flag)	Command			Post (Output port · Flag)
		Command	Operand 1	Operand 2	
Optional	Optional	CLR	Variable no.	Variable no.	

[Function] Clears the designated range of variables to zero.
(Be sure to input a variable number in Operand 2. When designating variable 1, be sure to input 1 in Operand 1 and Operand 2.

[Example] CLR 1 1
Clears the variable 1 to zero.

CLR 1 10
Clears variables 1 ~ 10 to zero.

2-9 Functional Commands

SIN (Sine)

Expansion condition (AND · OR)	Input condition (I/O · Flag)	Command			Post (Output port · Flag)
		Command	Operand 1	Operand 2	
Optional	Optional	SIN	Variable no.	Data·Variable no.	Optional

[Function] Stores the sine (SIN) of the data in Operand 2 in the variable in Operand 1. Stores the sine of the contents in the variable in Operand 2 in the Operand 1 variable. If there is a post instruction (flag, output port), it turns ON when the computation result is zero. (Be sure to input a real number variable 100~199 in Operand 1. This is the same when placing a variable no. in Operand 2. The data in Operand 2 must be expressed in radians instead of as an angle (degrees). A maximum of 7 digits can be input for the floating point.)

[Example] SIN 100 0.988031

SIN 0.988031 (radian) is 0.5, and after performing the command, variable 100 becomes 0.5.

SIN 100 *101

When the content of variable*101 in Operand 2 is 0.988031 (radian), then SIN 0.988031 (radian) becomes 0.5, and after performing the command, variable 100 becomes 0.5.

*180° = π (rad), 90° = 1/2 π (rad)
*Angle x π ÷ 180 = radian

COS (Cosine)

Expansion condition (AND · OR)	Input condition (I/O · Flag)	Command			Post (Output port · Flag)
		Command	Operand 1	Operand 2	
Optional	Optional	COS	Variable no.	Data·Variable no.	Optional

[Function] Stores the cosine (COS) of the data in Operand 2 in the variable in Operand 1. Stores the cosine of the contents in the variable in Operand 2 in the Operand 1 variable. If there is a post instruction (flag, output port), it turns ON when the computation result is zero. (Be sure to input a real number variable 100~199 in Operand 1. This is the same when placing a variable no. in Operand 2. The data in Operand 2 must be expressed in radians instead of as an angle (degrees). A maximum of 7 digits can be input for the floating point.)

[Example] COS 100 1.047197

COS 1.047197 (radian) is 0.5, and after performing the command, variable 100 becomes 0.5.

COS 100 *101

When the content of variable*101 in Operand 2 is 1.047197 (radian), then COS 1.047197 (radian) becomes 0.5, and after performing the command, variable 100 becomes 0.5.

TAN (Tangent)

Expansion condition (AND · OR)	Input condition (I/O · Flag)	Command			Post (Output port · Flag)
		Command	Operand 1	Operand 2	
Optional	Optional	TAN	Variable no.	Data·Variable no.	Optional

[Function] Stores the tangent (TAN) of the data in Operand 2 in the variable in Operand 1. Stores the tangent of the contents in the variable in Operand 2 in the Operand 1 variable. If there is a post instruction (flag, output port), it turns ON when the computation result is zero. (Be sure to input a real number variable 100~199 in Operand 1. This is the same when placing a variable no. in Operand 2. The data in Operand 2 must be expressed in radians instead of as an angle (degrees). A maximum of 7 digits can be input for the floating point.)

[Example] TAN 100 0.785398

TAN 0.785398 (radian) is 1, and after performing the command, variable 100 becomes 1.

TAN 100 *101

When the content of variable*101 in Operand 2 is 0.785398 (radian), then TAN 0.785398 (radian) becomes 1, and after performing the command, variable 100 becomes 1.

ATN (Arctangent)

Expansion condition (AND · OR)	Input condition (I/O · Flag)	Command			Post (Output port · Flag)
		Command	Operand 1	Operand 2	
Optional	Optional	ATN	Variable no.	Data·Variable no.	Optional

[Function] Stores the arc tangent (ATN) of the data in Operand 2 in the variable in Operand 1. Stores the arc tangent of the contents in the variable in Operand 2 in the Operand 1 variable. If there is a post instruction (flag, output port), it turns ON when the computation result is zero. (Be sure to input a real number variable 100~199 in Operand 1. This is the same when placing a variable no. in Operand 2. The data in Operand 2 must be expressed in radians instead of as an angle (degrees). A maximum of 7 digits can be input for the floating point.)

[Example] ATN 100 1

ATN 1 is 0.785398, and after performing the command, variable 100 becomes 0.785398.

ATN 100 *101

When the content of variable*101 in Operand 2 is 1, then ATN1 becomes 0.785398, and after performing the command, variable 100 becomes 0.785398.

SQR (Square Root)

Expansion condition (AND · OR)	Input condition (I/O · Flag)	Command			Post (Output port · Flag)
		Command	Operand 1	Operand 2	
Optional	Optional	SQR	Variable no.	Data·Variable no.	Optional

[Function] Stores the square root (SQR) of the data in Operand 2 in the variable in Operand 1. Stores the square root of the contents in the variable in Operand 2 in the Operand 1 variable. If there is a post instruction (flag, output port), it turns ON when the computation result is zero.

[Example] SQR 1 4

The square root of 4 is 2, and after performing the command, variable 1 becomes 2.

SQR 1 *2

When the content of variable*2 in Operand 2 is 4, then the square root of 4 is 2, and after performing the command variable 1 becomes 2.

2-10 Logical Operation Commands

AND (Logical AND)

Expansion condition (AND · OR)	Input condition (I/O · Flag)	Command			Post (Output port · Flag)
		Command	Operand 1	Operand 2	
Optional	Optional	AND	Variable no.	Data·Variable no.	Optional

[Function] Stores the results of the AND operation on the contents of the variable in Operand 1 and the data in Operand 2, in the variable in Operand 1. Stores the results of the AND operation on the variable in Operand 1 and the contents of the variable in Operand 2, in the variable in Operand 1. If there is a post instruction (flag, output port), it turns ON when the computation result is zero.

[Example] AND 1 3

If the content of variable 1 in Operand 1 is 131 in decimal notation (10000011 in binary notation) and the data in Operand 2 is 3 in decimal notation (00000011 in binary notation), the result of the AND operation is 3 in decimal notation (00000011 in binary notation), and variable 1 becomes 3.

Variable 1 (Operand 1)	10000011	AND
Data (Operand 2)	00000011	
Result (Store in Variable 1)	00000011 (3 in decimal notation, variable 1 becomes 3)	

OR (Logical OR)

Expansion condition (AND · OR)	Input condition (I/O · Flag)	Command			Post (Output port · Flag)
		Command	Operand 1	Operand 2	
Optional	Optional	OR	Variable no.	Data·Variable no.	Optional

[Function] Stores the results of the OR operation on the contents of the variable in Operand 1 and the data in Operand 2, in the variable in Operand 1. Stores the results of the OR operation on the variable in Operand 1 and the contents of the variable in Operand 2, in the variable in Operand 1. If there is a post instruction (flag, output port), it turns ON when the computation result is zero.

[Example] OR 1 3

If the content of variable 1 in Operand 1 is 128 in decimal notation (10000000 in binary notation) and the data in Operand 2 is 3 in decimal notation (00000011 in binary notation), the result of the OR operation is 131 in decimal notation (10000011 in binary notation), and variable 1 becomes 131.

Variable 1 (Operand 1)	10000000	OR
Data (Operand 2)	00000011	
Result (Store in Variable 1)	10000011 (131 in decimal notation, variable 1 becomes 131)	

EOR (Exclusive OR)

Expansion condition (AND · OR)	Input condition (I/O · Flag)	Command			Post (Output port · Flag)
		Command	Operand 1	Operand 2	
Optional	Optional	EOR	Variable no.	Data·Variable no.	Optional

[Function] Stores the results of the exclusive OR (EOR) operation on the contents of the variable in Operand 1 and the data in Operand 2, in the variable in Operand 1. Stores the results of the EOR operation on the variable in Operand 1 and the contents of the variable in Operand 2, in the variable in Operand 1. If there is a post instruction (flag, output port), it turns ON when the computation result is zero.

[Example] EOR 1 3

If the content of variable 1 in Operand 1 is 128 in decimal notation (10000000 in binary notation) and the data in Operand 2 is 3 in decimal notation (00000011 in binary notation), the result of the OR operation is 131 in decimal notation (10000011 in binary notation), and variable 1 becomes 131.

Variable 1 (Operand 1) 10000000 ┐ EOR
Data (Operand 2) 00000011 ┘
Result (Store in Variable 1) 10000011 (131 in decimal notation, variable 1 becomes 131)

2-11 Calculation Comparison

CPEQ (Compare if equal)

Expansion condition (AND · OR)	Input condition (I/O · Flag)	Command			Post (Output port · Flag)
		Command	Operand 1	Operand 2	
Optional	Optional	CPEQ	Variable no.	Data·Variable no.	Required

[Function] Turns ON the post flag or the output port if the values in Operand 1 and 2 are equal. (Variable values remain the same) When not equal (either greater or smaller), post section turns OFF.

* When real variables (100 ~ 199, 300 ~ 399) are used, it is almost impossible for the values to be equal after floating point calculations. To compare position data, a calculation technique must be applied prior to making the comparison, such as dividing by 100, then multiplying by 1000 to convert the number up to the first decimal place into an integer.

[Example] CPEQ 1 5 600

When the contents of variable 1 in Operand 1 equal the contents of Operand 2 (in this case, if it becomes 5), then output 600 turns ON. If the value is other than 5, output 600 turns OFF.

CPEQ 1 *2 600

When the contents of variable 1 in Operand 1 equal the contents of the variable in Operand 2, output 600 turns ON. If the content is not equal to that of Operand 2 (variable 2), 600 turns OFF.

CPNE (Compare if not equal)

Expansion condition (AND · OR)	Input condition (I/O · Flag)	Command			Post (Output port · Flag)
		Command	Operand 1	Operand 2	
Optional	Optional	CPNE	Variable no.	Data·Variable no.	Required

[Function] Turns ON the post flag or the output port if the values in Operand 1 and 2 are not equal (equivalent to when either the CPGT or CPLT condition is fulfilled). (Variable values remain the same) When the contents of Operand 1 and Operand 2 are equal, post section turns OFF.

[Example] CPNE 1 5 600

When the contents of variable 1 in Operand 1 does not equal the contents of Operand 2 (in this case, if it is greater or smaller than 5), then output 600 turns ON. If variable 1 reaches a value of 5, output 600 turns OFF.

CPNE 1 *2 600

When the contents of variable 1 in Operand 1 does not equal the contents of the variable in Operand 2, output 600 turns ON. If the content equals the value in Operand 2, 600 turns OFF.

CPGT (Compare if greater than)

Expansion condition (AND · OR)	Input condition (I/O · Flag)	Command			Post (Output port · Flag)
		Command	Operand 1	Operand 2	
Optional	Optional	CPGT	Variable no.	Data·Variable no.	Required

[Function] Turns ON the post flag or the output port when the value in Operand 1 is greater than the value in Operand 2. (Variable values do not change.) When the condition is not met (contents of Operand 1 are equal to or smaller than Operand 2), the output section turns OFF.

[Example] CPGT 1 5 600

When the content of variable 1 in Operand 1 is greater than the content of Operand 2 (in this case, when it becomes greater than 5), output 600 turns ON. When the condition is not met, 600 turns OFF.

CPGT 1 *2 600

When the content of variable 1 in Operand 1 is greater than the content of the variable in Operand 2, output 600 turns ON. When the condition is not met, it turns OFF.

CPGE (Compare if Equal or Greater)

Expansion condition (AND · OR)	Input condition (I/O · Flag)	Command			Post (Output port · Flag)
		Command	Operand 1	Operand 2	
Optional	Optional	CPGE	Variable no.	Data·Variable no.	Required

[Function] Turns ON the post flag or the output port when the value in Operand 1 is equal to or greater than the value in Operand 2. (Variable values do not change.) When the value in Operand 1 is smaller than the value in Operand 2, the output section turns OFF.

[Example] CPGE 1 5 600

When the content of variable 1 in Operand 1 is equal to or greater than the content of Operand 2 (in this case, equal to or greater than 5), output 600 turns ON. When smaller than 5, 600 turns OFF.

CPGE 1 *2 600

If the content of variable 1 in Operand 1 is equal to or greater than the content of the variable in Operand 2, output 600 turns ON. When the condition is not met, 600 turns OFF.

CPLE (Compare if equal or less than)

Expansion condition (AND · OR)	Input condition (I/O · Flag)	Command			Post (Output port · Flag)
		Command	Operand 1	Operand 2	
Optional	Optional	CPLE	Variable no.	Data·Variable no.	Required

[Function] Turns ON the post flag or the output port when the value in Operand 1 is equal to or less than the value in Operand 2. (Variable values do not change.) When the condition is not met, the output section turns OFF.

* It is almost impossible for the values to be equal as explained in the section on the CPEQ command.

[Example] CPLE 1 5 600

When the content of variable 1 in Operand 1 is equal to or less than the content of Operand 2 (in this case, equal to or less than 5), output 600 turns ON. If the content of variable 1 is 6 or greater, output 600 turns OFF.

CPLE 1 *2 600

If the content of variable 1 in Operand 1 is equal to or less than the content of the variable in Operand 2, output 600 turns ON. If it is greater than the content of variable 2, 600 turns OFF.

CPLT (Compare if less than)

Expansion condition (AND · OR)	Input condition (I/O · Flag)	Command			Post (Output port · Flag)
		Command	Operand 1	Operand 2	
Optional	Optional	CPLT	Variable no.	Data·Variable no.	Required

[Function] Turns ON the post flag or the output port when the value in Operand 1 is less than the value in Operand 2. (Variable values do not change.) If it is equal to or greater than the value in Operand 2, the output section turns OFF.

[Example] CPLT 1 5 600

When the content of variable 1 in Operand 1 is less than the content of Operand 2 (in this case, when it becomes equal to or less than 4), output 600 turns ON. When the condition is not met, output 600 turns OFF.

CPLT 1 *2 600

When the content of variable 1 in Operand 1 is less than the content of the variable in Operand 2, output 600 turns ON. When the condition is not met, 600 turns OFF.

2-12 Position Data Operation Commands

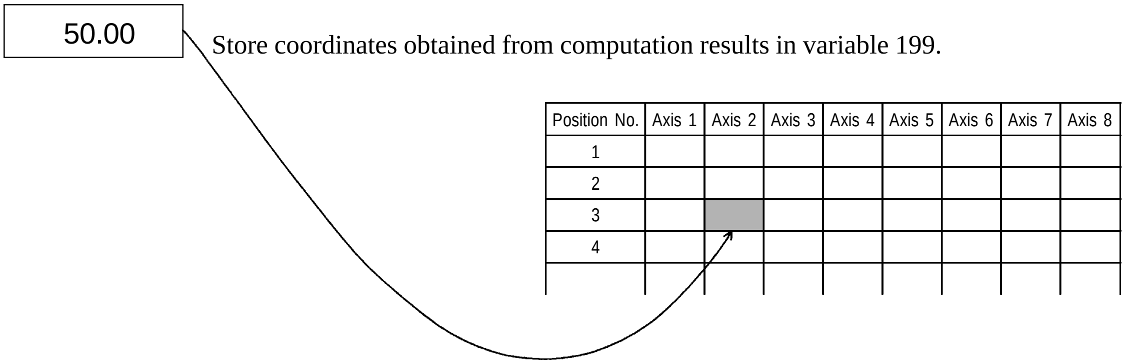
PPUT (Assign Axis Data)

Expansion condition (AND · OR)	Input condition (I/O · Flag)	Command			Post (Output port · Flag)
		Command	Operand 1	Operand 2	
Optional	Optional	PPUT	Axis no.	Position no. Variable no.	

[Function] Assigns the coordinates in variable 199 to the designated axis position data.

[Example] PPUT 2 3
Assign the coordinates in variable 199 to position no. 3 for axis no. 2.

Variable 199 (Exclusive Variable)



Store coordinates obtained from computation results in variable 199.

Position No.	Axis 1	Axis 2	Axis 3	Axis 4	Axis 5	Axis 6	Axis 7	Axis 8
1								
2								
3		▨						
4								

PGET (Read Axis Data)

Expansion condition (AND · OR)	Input condition (I/O · Flag)	Command			Post (Output port · Flag)
		Command	Operand 1	Operand 2	
Optional	Optional	PGET	Axis no.	Position no. Variable no.	

[Function] Reads the position data for the designated axis to variable 199. (Opposite of PPUT) When executing this command, if the data being read is xxx.xx, the data will not be entered in variable 199 (command is not executed).

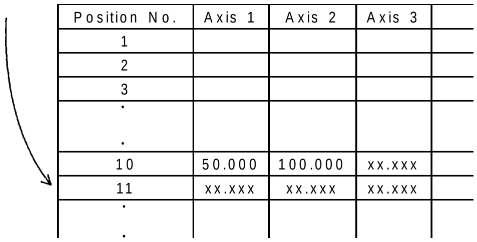
[Example] PGET 2 3
Read data at position no. 3 for axis no. 2 to variable 199.

PTST (Check Position Data)

Expansion condition (AND · OR)	Input condition (I/O · Flag)	Command			Post (Output port · Flag)
		Command	Operand 1	Operand 2	
Optional	Optional	PTST	Axis pattern	Position no.	Required

[Function] Checks to see whether there is valid data in the designated axis pattern and position number. If there is no data, the post flag or output port turns ON. Post section turns ON only when all the axes specified by the axis pattern are XX.XXX. ("0" is considered as data.)

[Example] PTST 11 11 600
If there is no data in position 11 of axis 1 and 2, flag 600 turns ON.



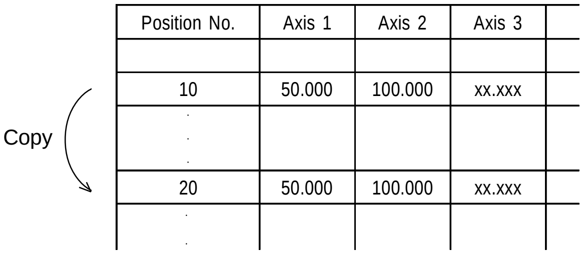
Position No.	Axis 1	Axis 2	Axis 3	
1				
2				
3				
· ·				
10	50.000	100.000	xx.xxx	
11	xx.xxx	xx.xxx	xx.xxx	
· ·				

PCPY (Position Data Copy)

Expansion condition (AND · OR)	Input condition (I/O · Flag)	Command			Post (Output port · Flag)
		Command	Operand 1	Operand 2	
Optional	Optional	PCPY	Position no.	Position no.	

[Function] Copies data in the designated position No. (copy data in Operand 2 to Operand 1).

[Example] PCPY 20 10
Copy data from position 10 in Operand 2 to position 20 in Operand 1.



Position No.	Axis 1	Axis 2	Axis 3	
10	50.000	100.000	xx.xxx	
· · ·				
20	50.000	100.000	xx.xxx	
· ·				

PCLR (Clear Position Data)

Expansion condition (AND · OR)	Input condition (I/O · Flag)	Command			Post (Output port · Flag)
		Command	Operand 1	Operand 2	
Optional	Optional	PCLR	Position no.	Position no.	

[Function] Clears the data in the range of positions designated by Operand 1 and Operand 2 (becomes XX.XXX, not 0.00).

[Example] PCLR 10 20
Clears data from position 10 in Operand 1 through position 20 in Operand 2.

PRED (Read Coordinates)

Expansion condition (AND · OR)	Input condition (I/O · Flag)	Command			Post (Output port · Flag)
		Command	Operand 1	Operand 2	
Optional	Optional	PRED	Axis pattern	Position no.	

[Function] Reads the current coordinates of the axis designated in Operand 1 and writes it to the position designated in Operand 2.

* This command can take in up to three decimal places.
* Please be sure that there is no error in the last digit after multiplication and division.

[Example] PRED 11 10
Write the current coordinates of axis 1 and axis 2 designated in Operand 1 into position 10.

PSIZ (Check Position Data Size)

Expansion condition (AND · OR)	Input condition (I/O · Flag)	Command			Post (Output port · Flag)
		Command	Operand 1	Operand 2	
Optional	Optional	PSIZ	Variable (to be assigned)		

[Function] Checks the maximum size of the position data that can be used.

[Example] PSIZ 1
The maximum value of the position data goes into variable 1 (variable to be assigned) in Operand 1.

PVEL (Assign Velocity Data)

Expansion condition (AND · OR)	Input condition (I/O · Flag)	Command			Post (Output port · Flag)
		Command	Operand 1	Operand 2	
Optional	Optional	PTST	Axis pattern	Position no.	

[Function] Assigns the value in Operand 1 as the velocity for the designated position data. Variables can also be used. This command is used to change the actual velocity.

* When a value is assigned that will result in a negative number after calculation, there is no warning at the time you execute this command but an alarm will occur when you try to use the data.

[Example] PVEL 100 3
Assign a value of 100mm/sec to the velocity data for position number 3.

PACC (Assign Acceleration Data)

Expansion condition (AND · OR)	Input condition (I/O · Flag)	Command			Post (Output port · Flag)
		Command	Operand 1	Operand 2	
Optional	Optional	PACC	Acceleration speed	Position no.	

[Function] Assigns the acceleration data in Operand 1 for the acceleration speed of the position data. As with the PVEL command, you can assign a value using the variable but there is no function that checks the value range when executing this command so please be careful not to assign a value that exceeds the actuator's limits.

[Example] PACC 0.3 3
Assign a value of 0.3 to the acceleration speed data for position number 3.

Part 4 Expansion Commands

1. Command Table

Note: These expansion commands cannot be used with the teaching pendant. Please use the PC Interface Software (DOS Ver. 2.0 or higher or Windows Ver. 1.0 or higher) for these commands.

Category	Function	Command	Page No.
Actuator Control Designation	S motion ratio setting	SCRV	163
Actuator Control Command	Incremental PTP movement	MVPI	164
	Incremental interpolation movement	MVLI	164
	Axis status acquisition	AXST	165
Timer Command	Time acquisition	GTTM	166
Calculation Command	Move	TRAN	167
Point Data Command	Read out axis pattern	PAXS	168
Structured IF Command	Equal	IFEQ	169
	Unequal	IFNE	169
	Greater than	IFGT	169
	Equal to or greater than	IFGE	169
	Less than	IFLT	169
	Equal to or less than	IFLE	169
	State execution destination when no IFconditions are established	ELSE	170
	Declare IF ending	EDIF	170
	Character string comparison Equal	ISEQ	171
	Character string comparison Unequal	ISNE	171
Structured DO Command	While equal	DWEQ	172
	While unequal	DWNE	172
	While greater than	DWGT	172
	While equal to or greater than	DWGE	172
	While less than	DWLT	172
	While equal to or less than	DWLE	172
	Leave DWXX	LEAV	173
	Repeat DWCC	ITER	173
	Declare DO ending	EDDO	174
External Input Output Command	Open channel	OPEN	175
	Close channel	CLOS	175
	Read in from channel	READ	176
	Write out to channel	WRIT	177
	Set ending character	SCHA	177
String Command	Copy character string	SCPY	178
	Compare character string	SCMP	179
	Get string	SGET	180
	Write in string	SPUT	180
	Change character string (decimal)	STR	181
	Change character string (hexadecimal)	STRH	182
	Change value (decimal)	VAL	183
	Change value (hexadecimal)	VALH	184
	Set length	SLEN	184
Branch Command	Declare beginning of branch	SLCT	185
	Declare branch destination when no conditions are established	OTHE	185
	Declare end of SLCT	EDSL	185
	Selected when true (variable)	WH*XX*	186
	Selected when true (character)	WS*XX*	187

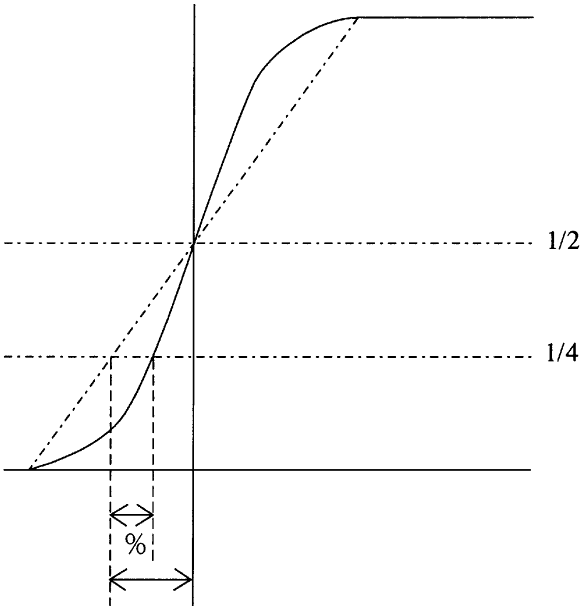
2. Commands

2-1 Actuator Control Designation

SCRV (S Motion Ratio Setting)

Expansion condition (AND · OR)	Input condition (I/O · Flag)	Command			Post (Output port · Flag)
		Command	Operand 1	Operand 2	
Optional	Optional	SCRV	Ratio		

[Function] Sets the ratio to control the S motion of the actuator.
The setting range is integers from 0 ~ 50 (%).
If this command is not used to set the ratio or when it is set to 0 (%), the actuator makes a trapezoid motion.



[Example]

```
SCRV    30
```

S motion ratio is set to 30%.

```
LET     1       50
SCRV    *1
```

Assign 50 to variable 1.
S motion ratio is set to 50 (%) which is the content of variable 1.

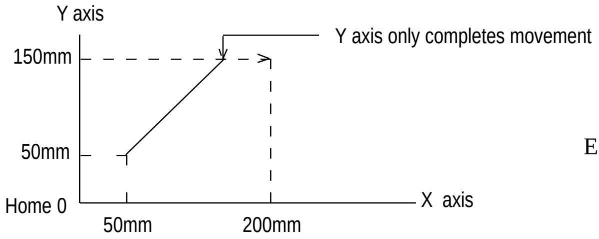
2-2 Actuator Control Command

MVPI (Incremental PTP Movement)

Expansion condition (AND · OR)	Input condition (I/O · Flag)	Command			Post (Output port · Flag)
		Command	Operand 1	Operand 2	
Optional	Optional	MVPI	Position no.		Optional

[Function] Moves the actuator to the designated position number in reference to the current position from point to point without interpolation.

[Example] MVPI 1
When the current position is (50, 50) and position 1 data is (150, 100), the actuators move 150 in the X direction and 100 in the Y direction to the position (200, 150).



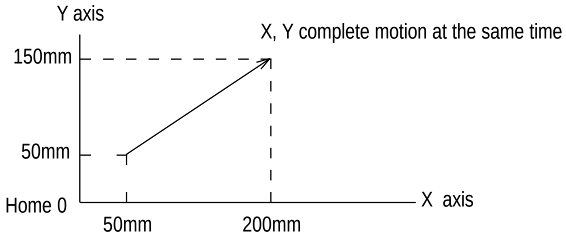
Each axis moves at its own designated speed.

MVLI (Incremental Interpolation Movement)

Expansion condition (AND · OR)	Input condition (I/O · Flag)	Command			Post (Output port · Flag)
		Command	Operand 1	Operand 2	
Optional	Optional	SCRV	Position no.		Optional

[Function] Moves the actuator to the designated point in Operand 1 from the current position while interpolating (not point to point).

[Example] MVLI 1
When the current position is (50, 50) and the position 1 data is (150, 100), the actuators move to the position (200, 150) which is 150 in X direction and 100 in Y direction from the current position.



The tip of the combined motion for each of the axes moves at the designated speed. The path from the start to the finish point makes a straight line.

AXST (Axis Status Acquisition)

Expansion condition (AND · OR)	Input condition (I/O · Flag)	Command			Post (Output port · Flag)
		Command	Operand 1	Operand 2	
Optional	Optional	AXST	Variable no.	Axis no.	

[Function] Stores the status (error code) of the axis in Operand 2 in the variable in Operand 1. Only error codes that begin with the letter "A" will be stored in the register in Operand 1. These error codes are the same ones that are displayed on the front panel of the controller. (The error codes in the table are written in hexadecimal numbers. The hexadecimal value in the variable in Operand 1 must be converted to a decimal number to identify its error code.)

[Example] AXST 1 2
Read the status for Axis 2 to variable 1.
If 161 was in variable 1,
161 ÷ 16 = 10 (= A) with a remainder of 1
This means that error code A1 (External Interrupt Error) occurred on Axis 2.

2-3 Timer Command

GTTM (Time Acquisition)

Expansion condition (AND · OR)	Input condition (I/O · Flag)	Command			Post (Output port · Flag)
		Command	Operand 1	Operand 2	
Optional	Optional	GTTM	Variable no.		

[Function] Writes the system time to the variable in Operand 1. The time unit is 10msec.
The time obtained with this command is a value that has no base. Therefore, call this command twice, and the difference gives the time that has elapsed.

[Example]

```
GTTM   1             Read the reference time to variable 1.
ADD    1      500    Set the ending time for 5 seconds later.
GTTM   2             Read the current time to variable 2.
DWLE   1      *2     After 5 seconds, proceeds to the next step after EDDO.
   .
   .
   .                 The transaction is repeated for 5 seconds.
   .
   .
GTTM   2             Read the current time to variable 2.
EDDO
```

2-4 Computation Command

TRAN (Transfer)

Expansion condition (AND · OR)	Input condition (I/O · Flag)	Command			Post (Output port · Flag)
		Command	Operand 1	Operand 2	
Optional	Optional	TRAN	Variable no.	Variable no.	

[Function] Assigns the contents of the variable in Operand 2 to the variable in Operand 1. This function is also known as "indirect addressing" or "pointing to a pointer."

[Example]

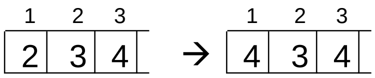
```
TRAN    1       2
```

Assign the content of variable 2 to variable 1. This has the same effect as,

```
LET     1       #2
```

```
LET     1       2
LET     2       3
LET     3       4
TRAN    1       *2
```

Assign 4 (4 is the content of the variable 3 which is the content of the variable 2) to variable 1.
The variable will change as below.



```
 1  2  3        1  2  3
 2  3  4   →    4  3  4
```

2-5 Position Data Operation Command

PAXS (Read Axis Pattern)

Expansion condition (AND · OR)	Input condition (I/O · Flag)	Command			Post (Output port · Flag)
		Command	Operand 1	Operand 2	
Optional	Optional	PAXS	Variable no.	Position no.	

[Function] Stores the axis pattern of the position in Operand 2 into the variable in Operand 1.

[Example]

```
PAXS     100     200
```

Store the axis pattern of position 200 into variable 100. When the points are set as in the position table below, 2 (10 in binary notation) is stored in variable 100.

[Example 2] When the points are set as below, 2 (10 in binary notation) is stored in variable 100.

```
LET      1       3
LET      2       301
PAXS     *1      *2
```

Assign 3 to variable 1.
Assign 101 to variable 2.
Store the axis pattern of the position for 101 which is the value contained in variable 2, to variable 3 which is contained in variable 1.
When the points are set as below, 3 (11 in binary notation) is stored in variable 3.

When the positions are set as shown in the table, the values to be stored in the variable will be as follows.

Position No.	Axis 2	Axis 1
98	xx.xxx	xx.xxx
99	xx.xxx	100.000
100	150.000	xx.xxx
101	100.000	50.000

· · · 00 = 0 + 0 = 0
· · · 01 = 0 + 1 = 1
· · · 10 = 2 + 0 = 2
· · · 11 = 2 + 1 = 3

2-6 Structured IF Command

IF*XX* (Structured IF)

Expansion condition (AND · OR)	Input condition (I/O · Flag)	Command			Post (Output port · Flag)
		Command	Operand 1	Operand 2	
Optional	Optional	IF*XX*	Variable no.	Data · Variable no.	

[Function] Compares the contents of the variable in Operand 1 and the value in Operand 2. When the condition is established, the program proceeds to the next step.
When the condition is not established, if there is a corresponding ELSE command, the program proceeds to the next step after that. If not, it proceeds to the next step after the corresponding EDIF command.
When the input condition is not established and there is no IFXX command executed, the program proceeds to the step following the corresponding EDIF.
Up to 15 levels of nesting are available when ISXX and DWXX are combined.

IFXX

EQ	· · ·	Operand 1 = Operand 2
NE	· · ·	Operand 1≠ Operand 2
GT	· · ·	Operand 1 > Operand 2
GE	· · ·	Operand 1≥ Operand 2
LT	· · ·	Operand 1 < Operand 2
LE	· · ·	Operand 1≤ Operand 2

[Example]

```
600   IFEQ   1     1     Select axis
      IFGE   2     0     Select moving direction
      JFWN   01    5     Move Axis 1 forward
      ELSE
      JBWN   01    5     Move Axis 1 backward
      EDIF
      ELSE
      IFNE   2     1     Select moving direction
      JFWN   10    5     Move Axis 2 forward
      ELSE
      JBWN   10    5     Move Axis 2 backward
      EDIF
      EDIF
```

Variable 1 selects Axis 1 or Axis 2. Variable 2 selects forward or backward to jog.
When flag 600 is OFF, nothing is done and the program proceeds to the step after the last EDIF.

*Do not use GOTO (TAG) in between IFXX and EDIF.

ELSE

Expansion condition (AND · OR)	Input condition (I/O · Flag)	Command			Post (Output port · Flag)
		Command	Operand 1	Operand 2	
		ELSE			

[Function] The ELSE command is used in conjunction with the IFXX command and ISXX command. When the condition is not established, the command following the ELSE statement will be executed.

[Example] Refer to IFXX.

EDIF (IF*XX* End)

Expansion condition (AND · OR)	Input condition (I/O · Flag)	Command			Post (Output port · Flag)
		Command	Operand 1	Operand 2	
		EDIF			

[Function] Declares the end of an IFXX command.

[Example] Refer to IFXX.

IS*XX* (String Comparison)

Expansion condition (AND · OR)	Input condition (I/O · Flag)	Command			Post (Output port · Flag)
		Command	Operand 1	Operand 2	
Optional	Optional	IS*XX*	Column no.	Column no. Literal character	

[Function] Compares the character string in the column numbers in Operand 1 and Operand 2. When the condition is established, the program proceeds to the next step.
When the condition is not established, if there is a corresponding ELSE command, the program proceeds to the next step after that. If not, it proceeds to the next step after the corresponding EDIF command.
The length of the string to be compared is set by the SLEN command. If there is a literal character in either Operand 1 or Operand 2, the length to be compared is that of the literal character.
When the input condition is not established and there is no IFXX command executed, the program proceeds to the step following the corresponding EDIF.
Up to 15 levels of nesting are available when ISXX and DWXX are combined.

```
ISXX
  ├─ EQ    · · ·   Operand 1 = Operand 2
  └─ NE    · · ·   Operand 1≠ Operand 2
```

```
            SCPY   10     'GOFD'    Go forward
            SCPY   14     'GOBK'    Go backward
            LET    1      5
            LET    2      14
600         ISEQ   1      '1AXS'    Select Axis
            SLEN   4
            ISEQ   5      10        Select moving direction
            JFWN   01     5         Move Axis 1 forward
            ELSE
            JBWN   01     5         Move Axis 1 backward
            EDIF
            ELSE
            ISNE   *1     *2        Select moving direction
            JFWN   10     5         Move Axis 2 forward
            ELSE
            JBWN   10     5         Move Axis 2 backward
            EDIF
            EDIF
```

Column 1 ~ 4 is to select Axis 1, Axis 2 and column 5 ~ 8 is to select the jog direction.
When flag 600 is OFF, nothing is done and the program proceeds to the step after the last EDIF.
When column 1 ~ 8 contains the data shown below, Axis 1 moves forward.

1	2	3	4	5	6	7	8
1	A	X	S	G	O	F	D

2-7 Structured DO Command

DW*XX* (DO WHILE)

Expansion condition (AND · OR)	Input condition (I/O · Flag)	Command			Post (Output port · Flag)
		Command	Operand 1	Operand 2	
Optional	Optional	DW*XX*	Variable no.	Data · Variable no.	

[Function] Compares the contents of the variable in Operand 1 and the value in Operand 2. While the condition is established, the commands are executed up to EDDO.
When the condition is not established, the program proceeds to the step after the corresponding EDDO command.
The LEAV command can be used to force the end of the loop.
When the input condition is not established, the DWXX command is not executed and the program proceeds to the next step after the corresponding EDDO.
Up to 15 levels of nesting are available when ISXX and DWXX are combined.

DWXX

EQ	· · ·	Operand 1 = Operand 2
NE	· · ·	Operand 1≠ Operand 2
GT	· · ·	Operand 1 > Operand 2
GE	· · ·	Operand 1≥ Operand 2
LT	· · ·	Operand 1 < Operand 2
LE	· · ·	Operand 1≤ Operand 2

[Example]

```
DWEQ  1      0
   .
   .
600   LEAV
   .
   .
EDDO
```

While variable 1 is 0, the commands up to the EDDO command are repeated.
If flag 600 turns ON during this time, the loop is forced to end and the program proceeds to the next step after the EDDO command.

LEAV (Escape from DO WHILE)

Expansion condition (AND · OR)	Input condition (I/O · Flag)	Command			Post (Output port · Flag)
		Command	Operand 1	Operand 2	
Optional	Optional	LEAV			

[Function] Escapes the DOXX loop, then the program proceeds to the next step after EDDO.

[Example]

```
          DWEQ  1      0
             .
             .
     600  LEAV
             .
             .
          EDDO
```

While variable 1 is 0, the commands up to the EDDO command are repeated.
If flag 600 turns ON during this time, the loop is forced to end and the program proceeds to the next step after the EDDO command.

ITER (Repeat)

Expansion condition (AND · OR)	Input condition (I/O · Flag)	Command			Post (Output port · Flag)
		Command	Operand 1	Operand 2	
Optional	Optional	ITER			

[Function] Forces the control to move to EDDO during the DOXX loop.

[Example]

```
          DWEQ  1      0
             .
             .
     600  ITER
             .
             .
          EDDO
```

While variable 1 is 0, the commands up to the EDDO command are repeated.
If flag 600 turns ON during this time, the loop is forced to end and control is forced to move to the EDDO command.

EDDO (End DO WHILE)

Expansion condition (AND · OR)	Input condition (I/O · Flag)	Command			Post (Output port · Flag)
		Command	Operand 1	Operand 2	
Optional	Optional	EDDO			

[Function] Declares the end of the loop which started with DWXX.
When a DWXX condition is not established, the program proceeds to next step after this command.

[Example] Refer to DWXX.

2-8 External Input Output Command

OPEN (Open Channel)

Expansion condition (AND · OR)	Input condition (I/O · Flag)	Command			Post (Output port · Flag)
		Command	Operand 1	Operand 2	
Optional	Optional	OPEN	Channel no.		

[Function] Opens the channel specified in Operand 1. Channels specified after this will be able to transmit and receive signals. An ending character must be set by the SCHA command before executing this command.

[Example]

```
SCHA    10
OPEN    1
        Designate 10 (=LF) as the ending character.
        Open channel 1.

SCHA    13
LET     1        2
OPEN    *1
        Designate 13 (=CR) as the ending character.
        Assign 2 to variable 1.  Open channel 2, the value contained in variable 1.
```

CLOS (Close Channel)

Expansion condition (AND · OR)	Input condition (I/O · Flag)	Command			Post (Output port · Flag)
		Command	Operand 1	Operand 2	
Optional	Optional	CLOS	Channel no.		

[Function] Closes the channel specified in Operand 1. Channels specified after this will be unable to transmit and receive signals.

[Example]

```
CLOS    1
        Close the channel.
```

[Example]

```
LET     1        2
CLOS    *1
        Assign 2 to variable 1.
        Close channel 2, the value contained in variable 1.
```

READ

Expansion condition (AND · OR)	Input condition (I/O · Flag)	Command			Post (Output port · Flag)
		Command	Operand 1	Operand 2	
Optional	Optional	READ	Channel no.	Column no.	

[Function] Reads the character string from the channel in Operand 1 to the column in Operand 2.
Stops reading when the character designated in the SCHA command appears.
The column can be either local or global.

[Example]

```
SCHA    10
OPEN    1
READ    1     2
CLOS    1
```

Set LF (= 10) for the ending character.
Open channel 1.
Read the character string from channel 1 to column 2 until LF appears.
Close channel.

```
LET     1     2
LET     2     3
SCHA    13
READ    *1    *2
```

Assign 2 to variable 1.
Assign 3 to variable 2.
Set CR (= 13) for the ending character.
Read the character string from channel 2 (content of variable 1) to column 3 (content of variable 2) until CR appears.

WRIT (Write)

Expansion condition (AND · OR)	Input condition (I/O · Flag)	Command			Post (Output port · Flag)
		Command	Operand 1	Operand 2	
Optional	Optional	WRIT	Channel no.	Column no.	

[Function] Writes the character string from the channel in Operand 1 to the column in Operand 2.
Stops writing after the character designated in the SCHA command is written.
The column can be either local or global.

[Example]

```
SCHA    10
OPEN    1
READ    1       2
CLOS    1
```

Set LF (= 10) for the ending character.
Open channel 1.
Write the character string up to LF from channel 1 to column 2.
Close channel.

```
LET     1       2
LET     2       3
SCHA    13
READ    *1      *2
```

Assign 2 to variable 1.
Assign 3 to variable 2.
Set CR (= 13) for the ending character.
Write the character string up to LF from channel 2 (content of variable 1) to column 3 (content of variable 2).

SCHA (Set Ending Letter)

Expansion condition (AND · OR)	Input condition (I/O · Flag)	Command			Post (Output port · Flag)
		Command	Operand 1	Operand 2	
Optional	Optional	SCHA	Character code		Optional

[Function] Sets the ending letter to be used in the READ command and WRIT command.
A value from 0 ~ 255 (character code used in BASIC) can be designated for the character.

[Example] Refer to the READ command and WRIT command.

2-9 String Management Command

SCPY (Copy Character String)

Expansion condition (AND · OR)	Input condition (I/O · Flag)	Command			Post (Output port · Flag)
		Command	Operand 1	Operand 2	
Optional	Optional	SCPY	Column no.	Column no. Literal character	

[Function] Copies the character string from the column in Operand 2 to the column in Operand 1.
Copies only the length set by the SLEN command.
When Operand 2 is a literal character, that is the length copied.

[Example]

```
SCPY    1       'ABC'
        Copy 'ABC' to column 1.

SLEN    10
SCPY    100     200
        Set the length of the operation to 10 bytes.
        Copy 10 bytes from column 200 to column 100.

LET     1       300
LET     2       400
SLEN    5
SCPY    *1      *2
        Assign 300 to variable 1.
        Assign 400 to variable 2
        Set the length of the operation to 5 bytes.
        Copy 5 bytes from column 400 (the content of variable 2) to column 300 (the content of variable 1).
```

SCMP (Compare Character String)

Expansion condition (AND · OR)	Input condition (I/O · Flag)	Command			Post (Output port · Flag)
		Command	Operand 1	Operand 2	
Optional	Optional	SCMP	Column no.	Column no. Literal character	Required

[Function] Compares the column in Operand 1 and the column in Operand 2.
Compares only the length set by the SLEN command.
When Operand 2 is a literal character, that is the length compared.

[Example]

```
SCMP    1       'ABC'   600
```

When column 1 ~ 3 is 'ABC', flag 600 turns ON.

```
SLEN    5
SCMP    10      30      999
```

Set the length to be compared to 5 bytes.
When the 5 bytes from column 10 and column 30 are equal, flag 999 turns ON.

```
LET     1       10
LET     2       20
SLEN    3
SCMP    *1      *2      310
```

Assign 10 to variable 1.
Assign 20 to variable 2
Set the length to be compared to 3 bytes.
When the 3 bytes in column 10 (the content of variable 1) and the 3 bytes in column 20 (the content of variable 2) are equal, then output 310 turns ON.

SGET (Acquire Character String)

Expansion condition (AND · OR)	Input condition (I/O · Flag)	Command			Post (Output port · Flag)
		Command	Operand 1	Operand 2	
Optional	Optional	SGET	Variable no.	Column no.	

[Function] Assigns 1 character from the column in Operand 2 to the variable in Operand 1.

[Example

```
SGET    1       100
```

Assign 1 byte of column 100 to variable 1.

```
LET     1       3
LET     2       1
SCPY    1       'A'
SGET    *1      *2
```

Assign 3 to variable 1.
Assign 1 to variable 2.
Copy 'A' to column 1.
Assign 'A' in column 1 (content of variable 2) to variable 3 (content of variable 1).

SPUT (Set Character)

Expansion condition (AND · OR)	Input condition (I/O · Flag)	Command			Post (Output port · Flag)
		Command	Operand 1	Operand 2	
Optional	Optional	SPUT	Column no.	Data	

[Function] Sets the data in Operand 2 to the column in Operand 1.

[Example]

```
SPUT    5       10
```

Set 10 (LF) to column 5.

```
LET     1       100
LET     2       50
SPUT    *1      *2
```

Assign 100 to variable 1.
Assign 50 to variable 2.
Set 50 (' 2') which is the content of variable 2 to column 100 (content of variable 1).

STR (Change Character String Decimal)

Expansion condition (AND · OR)	Input condition (I/O · Flag)	Command			Post (Output port · Flag)
		Command	Operand 1	Operand 2	
Optional	Optional	STR	Column no.	Data	

[Function] Copies the data in Operand 2 which has been converted to a decimal character string to the column in Operand 1. Uses zero-suppress to match this to the length set by the SLEN command.
Even if the data is longer than the length, the length set by the SLEN command takes precedence.

[Example]

```
SLEN    5. 3
STR     1        123
```

Set the length to a 5 digit integer with 3 decimals.
The following will be set in column 1~9,

1	2	3	4	5	6	7	8	9
		1	2	3	.	0	0	0

```
LET     1        10
LET     2        987. 6543
SLEN    2. 3
STR     *1       *2
```

Assign 10 to variable 1.
Assign 987. 6543 to variable 2.
Set the length to a 2 digit integer with 3 decimals.
The following will be set in column 10~15,

10	11	12	13	14	15
8	7	.	6	5	4

. Since the data was longer than the set length, 9 in the 100s place and 3 in the 4th decimal place are cut off.

STRH (Change Character String Hexadecimal)

Expansion condition (AND · OR)	Input condition (I/O · Flag)	Command			Post (Output port · Flag)
		Command	Operand 1	Operand 2	
Optional	Optional	STR	Column no.	Data	

[Function] Copies the data in Operand 2 which has been converted to a hexadecimal character string to the column in Operand 1. Uses zero-suppress to match only the integers to the length set by the SLEN command.
Even if the data is longer than the set length, the setting by the SLEN command will take precedence.

[Example]

```
SLEN    5
STRH    1         255
```

Set format for a 5 digit integer.
The following will be set in column 1~5,

1	2	3	4	5
			F	F

```
LET     1         10
LET     2         987. 6543
SLEN    2. 3
STRH    *1        *2
```

Assign 10 to variable 1.
Assign 987. 6543 to variable 2.
Set format for a 2 digit integer with 3 decimals.
The following will be set in column 10~11,

10	11
D	B

.3, the decimal segment of the SLEN command, and .6543 in variable 2 will be ignored.
The integer expressed in hexadecimal notation is ' 3DB'. However, 3 in the third digit will be cut off since the length is set to 2 digits.

VAL (Character String   Change Data   Decimal)

Expansion condition (AND · OR)	Input condition (I/O · Flag)	Command			Post (Output port · Flag)
		Command	Operand 1	Operand 2	
Optional	Optional	VAL	Variable no.	Column no.	

[Function] Converts the decimal data in the column in Operand 2 to a binary number and assigns this to the variable in Operand 1.
The length set by the SLEN command will be converted.

[Example]

```
SCPY    10      ' 1234'
SLEN    4
VAL     1       10
```

Set ' 1234' in column 10.
Set the length to 4 bytes.
' 1234' in column 10 is converted to the binary number 1234 and assigned to variable 1.

```
LET     1       100
LET     2       20
SCPY    20      ' 1234'
SCPY    24      '. 567'
SLEN    8
VAL     *1      *2
```

Assign 100 to variable 1.
Assign 20 to variable 2
Copy ' 1234' to column 20.
Copy '. 567' to column 24.
Set the length to 8 bytes.
' 1234. 567' in column 20 (content of variable 2) will be converted to the binary number 1234. 567 and assigned to variable 100 (content of variable 1).

VALH (Character String Data Change Hexadecimal)

Expansion condition (AND · OR)	Input condition (I/O · Flag)	Command			Post (Output port · Flag)
		Command	Operand 1	Operand 2	
Optional	Optional	VALH	Variable no.	Column no.	

[Function] Converts the hexadecimal data in the column in Operand 2 to a binary number and assigns this to the variable in Operand 1.
The length set by the SLEN command will be converted. Only the integers will be converted and the decimal places will be disregarded.

[Example]

```
SCPY    10      ' 1234'
SLEN    4
VAL     1       10
```

Set ' 1234' in column 10.
Set the length to 4 bytes.
The hexadecimal number ' 1234' in column 10 is converted to the binary number 4660 and assigned to variable 1.

```
LET     1       100
LET     2       20
SCPY    20      '  ABCD'
SLEN    4
VALH    *1      *2
```

Assign 100 to variable 1.
Assign 20 to variable 2
Copy ' ABCD' to column 20.
Set the length to 4 bytes.
The hexadecimal ' ABCD' in column 20 (content of variable 2) will be converted to the binary number 43982 and assigned to variable 100 (content of variable 1).

SLEN (Set Length)

Expansion condition (AND · OR)	Input condition (I/O · Flag)	Command			Post (Output port · Flag)
		Command	Operand 1	Operand 2	
Optional	Optional	SLEN	Length		

[Function] Sets the length for the string command.
The length must be set prior to using any of the following commands.

SCMP	· · ·	Decimals Invalid
SCPY	· · ·	Decimals Invalid
ISXX	· · ·	Decimals Invalid
STRH	· · ·	Decimals Invalid
VAL, VALH	· · ·	Decimals Invalid
STR	· · ·	Decimals Valid

[Example] Refer to each of the commands above.

2-10 Branch Command

SLCT (Beginning of selected group)

Expansion (AND·OR)	Input (Input·Output·Flag)	Command · Statement			Post (Output · Flag
		Command	Operand 1	Operand 2	
Optional	Optional	SLCT			

Function: Branches to the next step in the OTHE command if none of the conditions set up by the WH*XX*, WS*XX* commands or any commands up to the EDSL command are met.

Example:

```
        SCPY   1      'right'        Assign 'right' to columns 1 through 5.
          :
600     SLCT
        WSEQ   1      'right'        Since the string in columns 1~5 are equal to 'right', the commands
          :                          that follow this WSEQ will be executed.
        WSEQ   1      'left'         Since the string in columns 1~5 are not equal to 'left', the com-
          :                          mands that follow this WSEQ will not be executed.
        OTHE                         If it is neither, then the commands that follow OTHE are executed.
          :
        EDSL                         When flag 600 is OFF, or if any one of the conditions is executed,
                                     then end the select.
```

OTHE (Selected in the case of other)

Expansion (AND·OR)	Input (Input·Output·Flag)	Command · Statement			Post (Output · Flag
		Command	Operand 1	Operand 2	
		OTHE			

Function: This is used between the SLCT ~ EDSL commands. This declares the command to be executed when no other conditions are met.

Example: Please refer to SLCT, WH*XX* and WS*XX*

EDSL (End of selected group)

Expansion (AND·OR)	Input (Input·Output·Flag)	Command · Statement			Post (Output · Flag
		Command	Operand 1	Operand 2	
		EDSL			

Function: This declares the end of the SLCT command.

Example: Please refer to SLCT, WH*XX* and WS*XX*

WH*XX* (Selected when true Variable)

Expansion (AND·OR)	Input (Input·Output·Flag)	Command · Statement			Post (Output · Flag
		Command	Operand 1	Operand 2	
		WH*XX*	Variable No.	Data	

Function: This is used between the SLCT ~ EDSL commands. Compares contents of the variable in Operand 1 to the value in Operand 2. If the conditions are met, then the code following the WH*XX* will be executed up to the next WH*XX*. If the conditions are not met, the program will go to the next WH*XX* command or OTHE command or EDSL.

WH*XX*

EQ	.... Operand 1 = Operand 2
NE	.... Operand 1 ≠ Operand 2
GT	.... Operand 1 > Operand 2
GE	.... Operand 1 ≥ Operand 2
LT	.... Operand 1 < Operand 2
LE	.... Operand 1 ≤ Operand 2

Example:

LET	1	20	Assigns 20 to variable 1.
LET	2	10	Assigns 10 to variable 2.
:			
SLCT			Branches.
WHEQ	1	10	If the content of variable is 10, ① is executed but since the content is 20, program refers to the next condition.
:			
①			
:			
WHGT	1	*2	Executed if the content of variable 1 is greater than the content of variable 2. Variable 1 (=20) > variable 2 (=10), so ② is executed.
:			
②			
:			
OTHE			If no conditions are fulfilled, this is executed. Since ② was executed, ③ will not be executed.
:			
③			
:			
EDSL			When one of the conditions is met and that command is performed, processing moves to EDSL. In this example, ② and ④ are executed.
:			
④			
:			

* When there is a possibility of several conditions being met, the W*XXX* command that appears first goes into effect and the commands that follow are not executed.
When conditions are demanding, list the ones with the highest priority first.

WS*XX* (Selected when true Character)

Expansion (AND·OR)	Input (Input·Output·Flag)	Command · Statement			Post (Output · Flag
		Command	Operand 1	Operand 2	
		WS*XX*	Column No.	Column No · Literal character	

Function: This is used between the SLCT ~ EDSL commands. Compares the character string in the columns in Operand 1 and Operand 2. If the conditions are met, then the code following the WS*XX* will be executed up to the next WS*XX*. If the conditions are not met, the program will go to the next WS*XX* command or OTHE command or EDSL. Comparison is made based on the length designated in the SLEN command. When Operand 2 is a literal character, that is the length that is executed.

```
WSXX
   └─────────┬ EQ  ........... Operand 1  =  Operand 2
             └ NE  ........... Operand 1  ≠  Operand 2
```

Example:

```
SLEN    3
SCPY    1        'ABC'
LET     1        3
   :
SLCT
WSEQ 1'XYZ'
   :
   ①
   :
WSGT 2           *1
   :
   ②
   :
OTHE
   :
   ③
   :
EDSL
   :
   ④
   :
```

Sets the number of characters to be compared to 3.
Assigns 'ABC' to column 1.
Assigns 3 to variable 1.

Branches.
If columns 1~3 are 'XYZ', ① is executed but since columns 1~3 are 'ABC', this is not executed.

If the number of characters designated by SLEN from column 2 are the same as the contents in the column indicated by variable 1, then ② is executed.

If no conditions are fulfilled, this is executed. Since ② was executed, ③ will not be executed.

When one of the conditions is met and that command is performed, processing moves to EDSL. In this example, ② and ④ are executed.

* When there is a possibility of several conditions being met, the W*XXX* command that appears first goes into effect and the commands that follow are not executed.
When conditions are demanding, list the ones with the highest priority first.

Part 5 Parameter List

All system parameters are appropriately set at the time of shipment. Basically, the user does not need to change the parameters but if you need to change them for a special system or requirements, please contact our technical service department. Also, please save the parameters when you change them. Doing a reset after rewriting the parameters or after applying an emergency stop, validates the new parameters.
The following tables are the initial values displayed by the Teaching Pendant. The actual parameters set at the time of shipment will differ depending on the actuators in your particular system.)

1. Axis Parameters

(a) Servo control related (Srvo)

No.	Parameter Name	Default	Contents	Remarks
1	Numerator	1	Numerator	Usable
2	Denominator	1	Denominator	Usable
3*	Over ride (%)	100	Over ride	Unused, set with the servo device
4	Acceler (G)	0.30	Acceleration	
5	Jog Vel	30	Jog velocity	Velocity during Teach mode
6	Pend Band	10	Position end band (Pulse)	
7	Soft Limit Off	2.00	Software limit offset	
8	Soft Limit (+)	9999	Software limit (+)	
9	Soft Limit (-)	0	Software limit (-)	

*Currently, a common parameter is used for Over Ride so this is not available.

(b) Homing

No.	Parameter Name	Default	Contents
1	Home Dir	0	Home direction
2	Home Type	0	Homing method
3 *1	Home Sequence	0~9	Sequence (axis used · unused)
4	Home Sw Pol	1	Limit input polarity
5	Home Z Edge	1	Z-phase sensing edge
6 *2	Home Creep Vel	0	Creep velocity
7	Home Back Vel	10	Run-in velocity
8	Home Z Vel	5	Z-phase search velocity
9	Home Offset	0	Offset move amount (length)
10	Home Deviation	667	Hard stop deviation (pulse)
11	Home Current	60	Current limit

*1 If you set any number between 1~9, you can designate the homing sequence of the axes being used. If you set the number 0, the designated axis becomes an unused axis (unconnected).

*2 The creep function is only available for units with the limit switch option. Always set this value to 0. If it is set to a value other than 0 in actuators without the limit switch, homing will not be executed properly.

(c) Motor*

No.	Parameter Name	Default	Content
1	Motor RPM Max	4000	Motor RPM Maximum
2	Encoder Pulse	400	Encoder Pulse Per Rev.
3	Screw Lead	8	Screw Lead (mm)
4	Multiple	4	Encoder Pulse Multiplier
5	Brake Time	0.1	Brake Time (sec)
6	Position Gain	60	Position Gain
7	Speed Gain	80	Speed Gain
8	F/F Gain	0	Feed Forward Gain
9	Integral Gain	30	Integral Gain
10	Total Gain	150	Total Gain
11	Int. Volt. Lmt.	60	Integral Voltage Limit
12	Over Speed	410	Over Speed Constant
13	Error Range	2666	Error Range
14	Motor Max Cur	90	Motor Maximum Current
15	Motor Over Load	16300	Motor Overload Lower Limit

*Motor related parameters differ depending on the actuator. The table above is only an example. Please contact our technical service department when motor related parameters must be used for your system.

(d) Axis Name

No.	Parameter Name	Default	Valid Name
1	Axis 1	1	Axis Name 0 ~ 9, A - Z Setting
2	Axis 2	2	Axis Name 0 ~ 9, A - Z Setting
3	Axis 3	3	Axis Name 0 ~ 9, A - Z Setting
4	Axis 4	4	Axis Name 0 ~ 9, A - Z Setting
5	Axis 5	5	Axis Name 0 ~ 9, A - Z Setting
6	Axis 6	6	Axis Name 0 ~ 9, A - Z Setting
7	Axis 7	7	Axis Name 0 ~ 9, A - Z Setting
8	Axis 8	8	Axis Name 0 ~ 9, A - Z Setting

2. System Parameters

(a) Application Program

No.	Parameter Name	Default	Content	Standard
1	Auto Start PRG	0	Auto start program number	---
2	Emergency PRG	0	Emergency stop program number	---
3*	Program Size	64	Number of programs	64
4*	Task Size	16	Number of tasks	16
5*	Step Size	3000	Number of program steps	3000
6	Time Slice	0.01	Time slice check value	---

*For reference only. Cannot be changed.

(b) Position Data

No.	Parameter Name	Default	Content	Standard
1*	Point Size	2000	Point data quantity	2000

*For reference only. Cannot be changed.

(c) Servo Device

No.	Parameter Name	Default	Content	Remarks
1	Axis Size	8	Number of axes	
2	Numerator	1	Numerator	Unusable Set by axis
3	Denominator	1	Denominator	
4	Over Ride (%)	100	Over ride	Usable
5	Acceler (0.01G)	0.30	Acceleration factor	Usable
6	Acc Max (0.01G)	1.00	Maximum acceleration factor	
7	Drive Vel	100	Drive velocity mm/sec	Velocity during step positioning
8	Drive Vel Max	1000	Maximum drive velocity mm/sec	

(d) Communication

No.	Parameter Name	Default	Content
1 *	Terminal ID	99	Multi-drop address code
2 *	Time Out (sec)	0	Time out
3 * ⌖	Baud Rate	3	Baud rate
4 * ⌖	Char Length	0	Character length
5 * ⌖	Parity	1	Parity
6 * ⌖	Stop Bit	0	Stop bit

*For reference only. Cannot be changed. ⌖Actual setting is fixed: 9600 Baud, 8 Bit, N Parity, 1 Stop.

(e) Circular/Path

No.	Parameter Name	Default	Content
1 *	Circle Angle	15.0	Slice angle (15 degrees)
2 *	Circle Delt	0	Velocity increment (mm/sec)

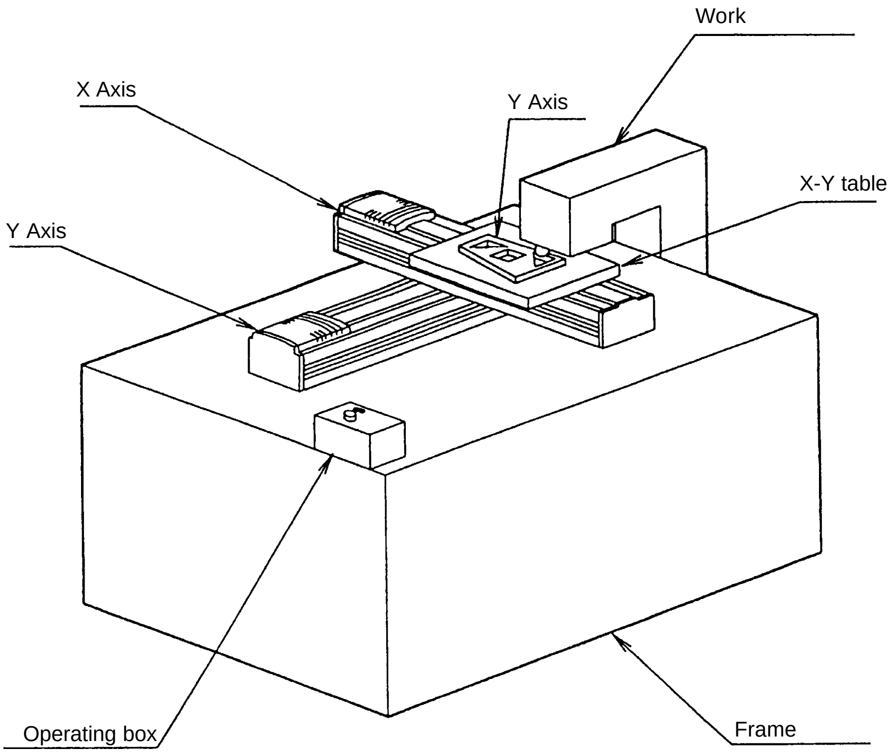
*Fixed parameters

Part 6 Application Program Examples

1. Movement Using the Point Move Command [Riveting Device]

(1) System Description

The riveting device is composed of an X-Y table (X and Y actuators) and a riveting unit. Work is placed on the X-Y table. The system homes, then a start signal is given. The device shown below attaches rivets to 3 points on the work.

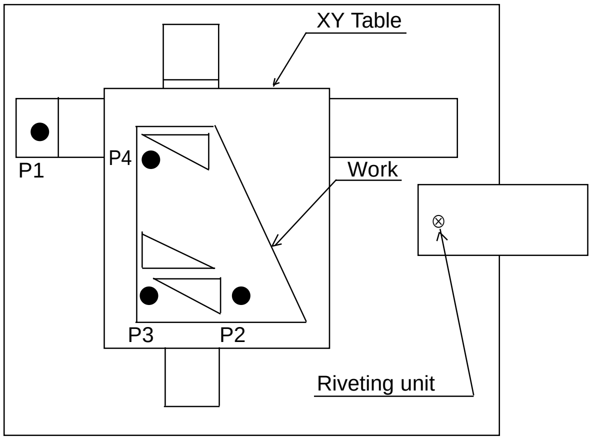


(2) Movement Description

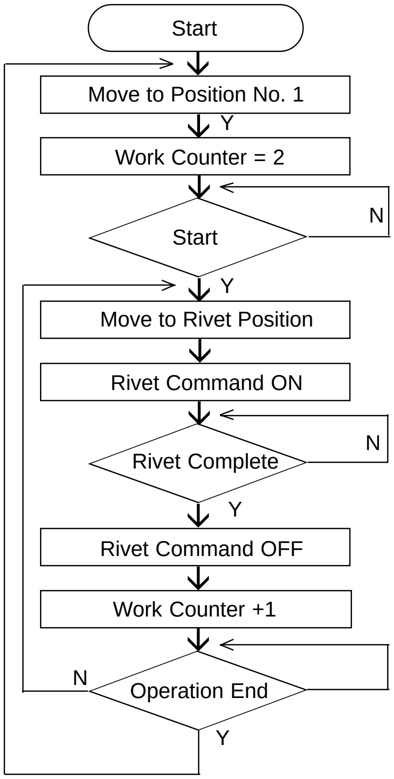
1. X and Y-axis return to home and wait.
2. Worker sets work on X-Y-table, then turns start switch (SW) ON.
3. X-Y-table moves to position No.1 then outputs riveting command to the riveting unit.
4. After riveting is completed, wait for completion signal input. Repeat the same procedure for position No.2 and position No.3.
5. After riveting at all three positions, axis returns to home.

Repeat the same procedure as above. The operating position, external I/O assignment and flowchart are as follows.

Operating Position



Flowchart



I/O Assignment

Section		I/O #	Signals	Specifications
SuperSEL	I	020	Start Command	Push Button SW
		021	Rivet Complete	Contact Signal
	O	310	Rivet Command	DC 24V
	Flag In Use From 600			

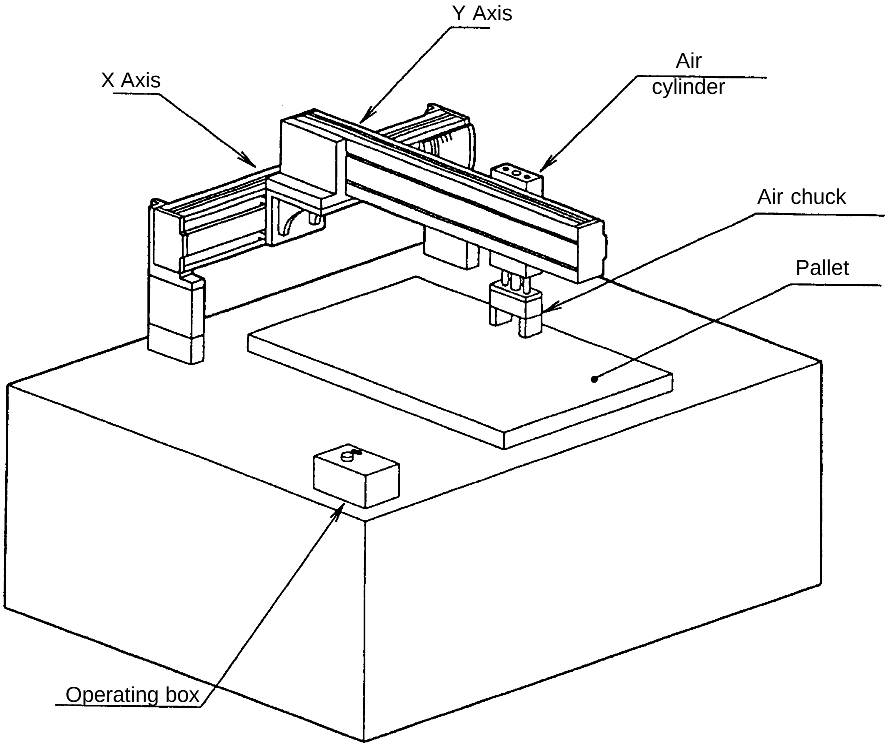
(3) Super SEL Controller Application Program

Step	A/O	N	OP-Code	Operand 1	Operand 2	Post	Comments
1			HOME	11			X-Y table homing (Servo ON)
2			VEL	400			Velocity 400mm/s setting
3			TAG	1			
4			MOVL	1			Move to position no. 1
5			LET	1	2		Set 2 on part counter
6			BTOF	600			Complete flag, clear
7			WTON	20			Waiting for start command
8			TAG	2			
9			MOVL	*1			Part counter position move
10			BTON	310			Rivet command ON
11			WTON	21			Waiting for rivet completion
12			BTOF	310			Rivet command OFF
13			ADD	1	1		Part counter +1
14			CPEQ	1	5	600	When part complete, Flag ON
15		N600	GOTO	2			When not complete, jump to TAG 2
16			GOTO	1			When complete, jump to TAG 1
17							
18							
19							
20							
21							
22							
23							
24							
25							
26							
27							
28							
29							
30							
31							
32							

2. Palletizing Operation [Palletizing Device]

(1) System Description

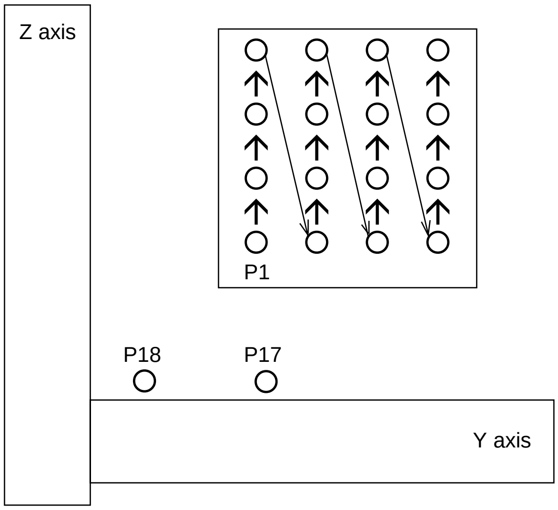
This system consists of an X-Y configuration with a pneumatic Z-axis. The system is used to pick up parts from a supply point, go to another point, and place the part on a pallet in a certain sequence.



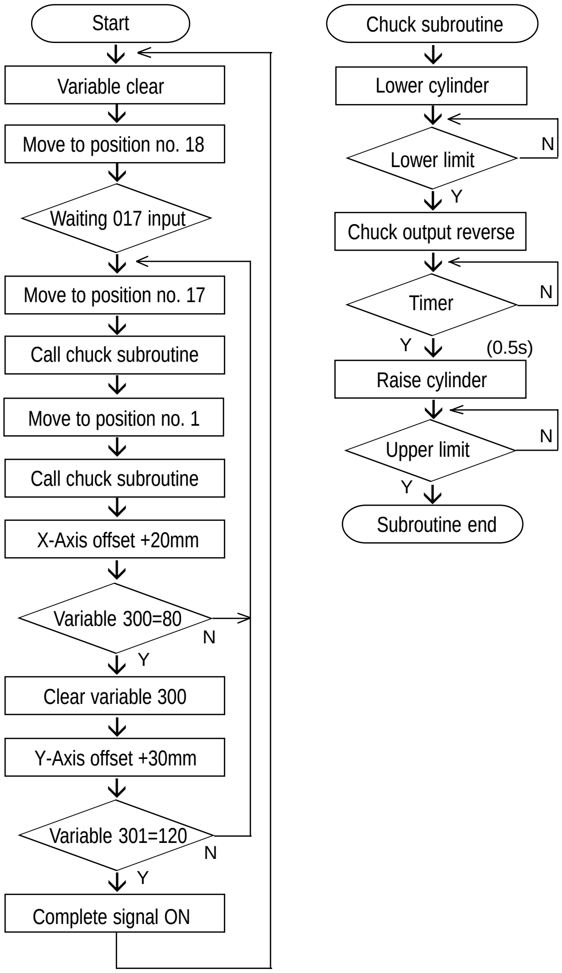
(2) Movement Description

1. Move to the waiting position, and wait for start input.
2. Move to the part supply point, after start input.
3. Z-axis moves downward, and the air chuck picks up a part.
4. Z-axis moves upward and moves to another point above the pallet.
5. Z-axis moves downward, then the air chuck releases the part.
6. Z-axis moves upward and moves back to the part supply point.
7. When the pallet has been completed, moves to P18. After the pallet completion output, waits for re-start input. Repeat the same procedure above. Operating position, external I/O assignment and flowchart are as follows.

Operating Position



Flowchart



I/O Assignment

Section	I/O #	Signals	Specifications
Input	015	Z-Axis Cylinder Up	Adjacent SW
	016	Z-Axis Cylinder Down	Adjacent SW
	017	Start	Push Button SW
Output	310	Z-Axis Cylinder SV	DC24V
	311	Z-Axis Chuck SV	DC24V
	312	Pallet Complete	DC24V
*Flag In Use From 600			

Pallet Specifications

X Direction	20mm pitch
Y Direction	30mm pitch

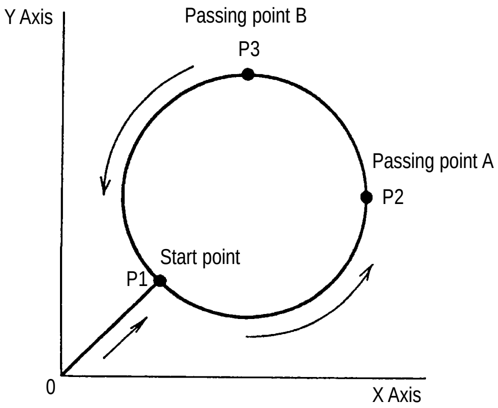
(3) Super SEL Controller Application Program

Step	A/O	N	OP-Code	Operand 1	Operand 2	Post	Comments
1			HOME	11			X and Y axis homing
2			VEL	100			Velocity 100mm/s setting
3			ACC	0.2			Acceleration 0.2G
4			TAG	1			
5			LET	300	0		Variable clear
6			LET	301	0		Variable clear
7			OFST	11	0		Offset value clear
8			MOVL	18			Move to position no. 18
9			WTON	17			Wait for start input
10			BTOF	312			Output 312 OFF
11			TAG	2			
12			OFST	11	0		Offset value clear
13			MOVL	17			Move to position no. 17
14			EXSR	1			Chuck subroutine call (Chuck)
15			OFST	1	*300		X-Axis, Variable 300 offset
16			OFST	10	*301		Y-Axis, Variable 301 offset
17			MOVL	1			Move to position no. 1 + offset value
18			EXSR	1			Chuck subroutine call (unchuck)
19			ADD	300	20		Add 20 to variable 300
20			CPEQ	300	80	600	When variable 300=80, flag 600 ON
21		N600	GOTO	2			If flag 600 is OFF, jump to tag 2
22			LET	300	0		Variable 300 clear
23			ADD	301	30		Add 30 to variable 301
24			CPEQ	301	120	601	When variable 301=120, flag 601 ON
25		N601	GOTO	2			If flag 601 is OFF, jump to tag 2
26			BTON	312			Output 312 ON
27			GOTO	1			Jumpt to tag 1
28			BGSR	1			Chuck subroutine start
29			BTON	310			Z-Axis cylinder down
30			WTON	16			Lower limit input waiting
31			BTNT	311			Air chuck output revert
32			TIMW	0.5			Timer 0.5 seconds
33			BTOF	310			Z-Axis cylinder up
34			WTON	15			Upper limit input waiting
35			EDSR				Chuck subroutine end
36							
37							
38							
39							

3. Circular Movement Command

In less sophisticated controllers it was generally understood that the changes in speed would result in changes in the motion profile. In the new Super SEL Controller, however, accurate circular motion profiles can be achieved by passing through the exact points regardless of changes in speed settings.

Locus



Programming

Circular movement only applies to two dimensional movement. A circular motion program is easily constructed by selecting a starting point and two passing points.

1. Set Position No.1 (P1) coordinate data (Starting Point).

2. Set Position No.2 (P2) and Position No.3 (P3) coordinate data (Passing Points).

3. Use the MOVL command to move to Position No.1, and make this the starting point (P1) of the circular move.

4. Use the CIR command to set the first passing point (P2) in OPRND1 and the second passing point (P3) in OPRND2.
   * The direction of rotation can be reversed by exchanging P2 and P3.

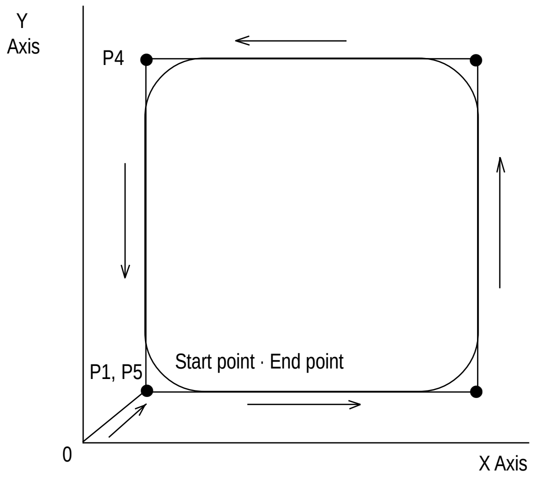
Program of Locus above

Step	A/O	N	OP-Code	Operand1	Operand2	Post	Comment
1			HOME	11			X and Y-Axis Homing
2			VEL	200			Velocity 200mm/s Setting
3			ACC	0.3			Acceleration 0.3G
4			MOVL	1			Move to Position No.1 (Starting Point)
5			TAG	1			
6			CIR	2	3		Circular Motion Passing through P2 and P3
7			GOTO	1			Jump TAG 1

4. Path Movement Command

In the Super SEL Controller, accurate path motion profiles can be achieved by passing through the exact points regardless of changes in speed settings.

Locus



Programming

A path motion program can be constructed easily with only a starting point and end point. Therefore, a program takes only one line regardless of the number of passing points between the starting point and end point.

1. Set Position No.1 (P1) coordinate data (Starting Point).

2. Set coordinate date for Position No. 2 (P2) through Position No. 4 (P4) (passing points), and Position No. 5 (P5) (end point). For continuous path movement, set the passing points in the sequence you wish them to move.

3. Use the MOVL command to move to Position No.1, and make this the starting point of the path movement.

4. Use the PATH command to set the starting point (P1) in OPRND1 and the end point (P5) in OPRND2.
   * The direction of the path movement can be reversed by exchanging the start and end points.

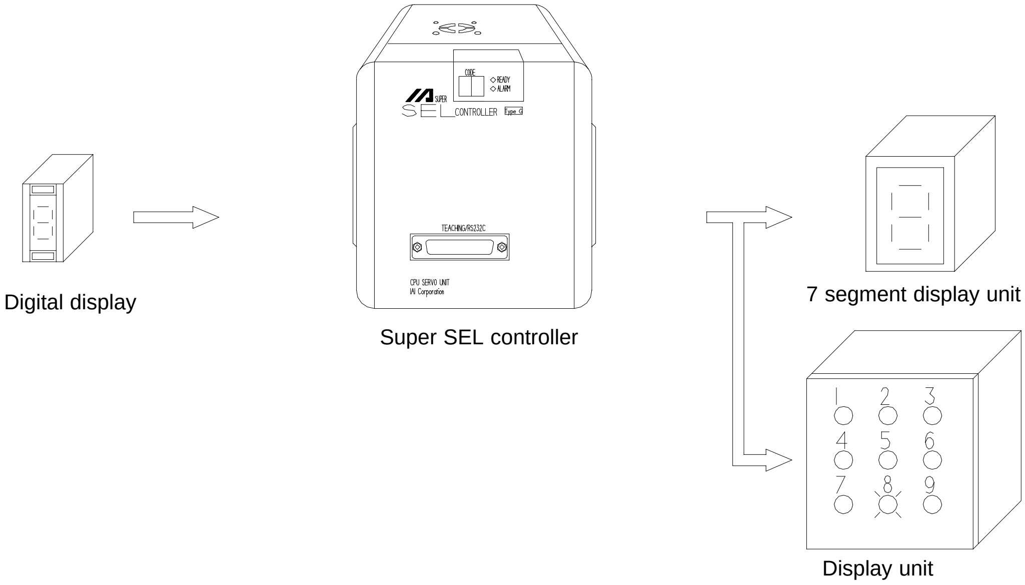
Program of Locus above

Step	A/O	N	OP-Code	Operand 1	Operand 2	Post	Comment
1			HOME	11			X and Y axes homing (Servo ON)
2			VEL	200			Velocity 200mm/s setting
3			ACC	0.3			Acceleration 0.3G
4			MOVL	1			Move to position no. 1 (starting point)
5			TAG	1			
6			PATH	1	5		Path motion from position no. 1 to no. 5
7			GOTO	1			Jump to TAG 1

5. BCD Code Signals Input and Output

(1) Circuit

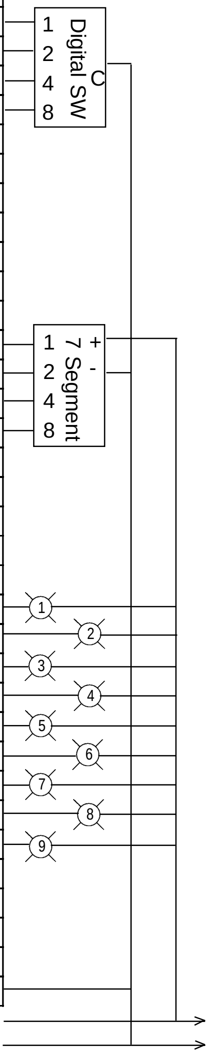
This circuit allows the controller to verify the BCD values input by the external digital switch, and lights the number on the display unit corresponding to the number displayed on the 7 segment display unit provided separately.



(2) I/O Connections

The following I/O connections are for external digital switches and the external display unit.

Pin No.	Category	Port No.	Function
1A	P24	---	---
1B	Input	000	External Start Input
2A		001	User Input
2B		002	Emergency Stop b Contact Input
3A		003	SystemReserve
3B		004	SystemReserve
4A		005	User Input
4B		006	User Input
5A		007	User Input
5B		008	PRG No. 1 (User Input)
6A		009	PRG No. 2 (User Input)
6B		010	PRG No. 4 (User Input)
7A		011	PRG No. 8 (User Input)
7B		012	PRG No. 10 (User Input)
8A		013	PRG No. 20 (User Input)
8B		014	PRG No. 40 (User Input)
9A		015	User Input
9B		016	User Input
10A		017	User Input
10B		018	User Input
11A		019	User Input
11B		030	User Input
12A		021	User Input
12B		022	User Input
13A		023	User Input
13B	Output	300	Emergency Stop/Alarm Output
14A		301	Ready Output
14B		302	User Output
15A		303	User Output
15B		304	User Output
16A		305	User Output
16B		306	User Output
17A		307	User Output
17B		308	User Output
18A		309	User Output
18B		310	User Output
19A		311	User Output
19B		312	User Output
20A		313	User Output
20B		314	User Output
21A		315	User Output
21B		316	User Output
22A		317	User Output
22B		318	User Output
23A		319	User Output
23B		320	User Output
24A		321	User Output
24B		322	User Output
25A		323	User Output
25B	N24	---	---

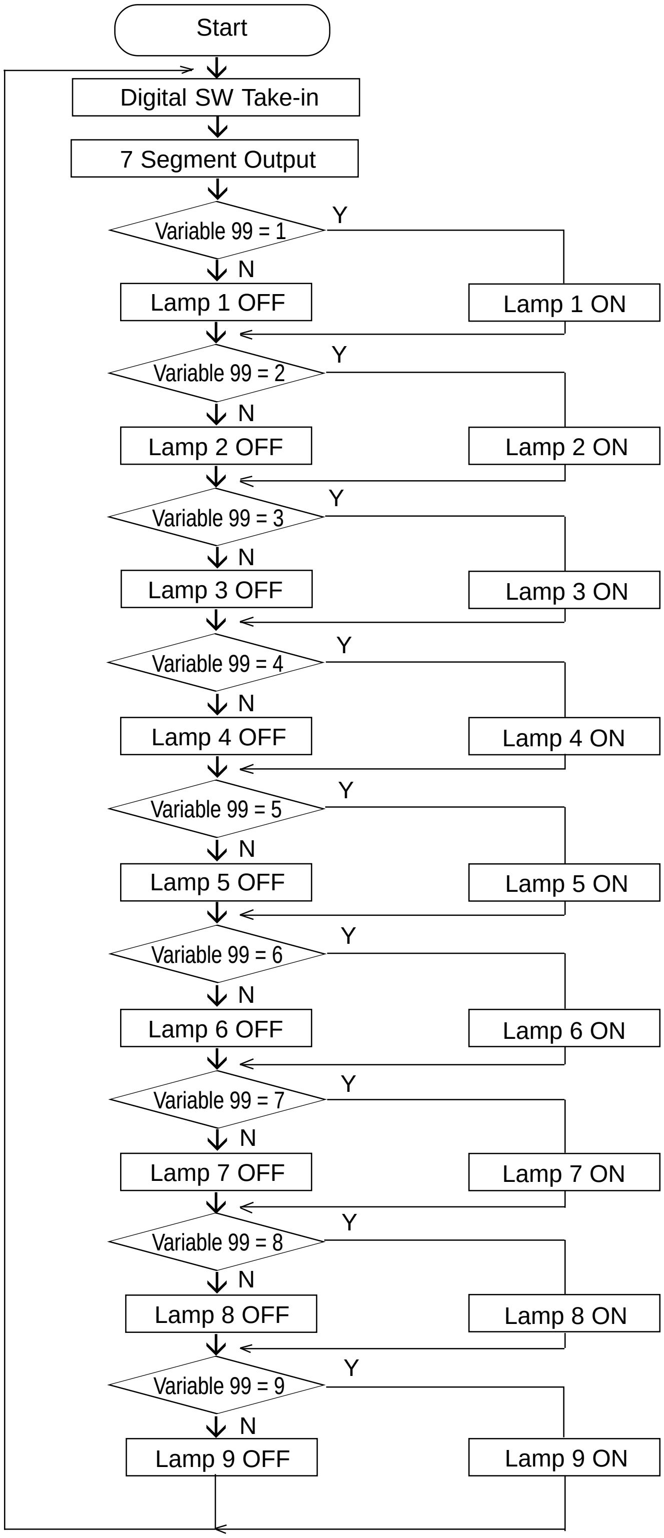


External 24V Power

I/O Assignment

Section	I/O#	Sgnals
Input	015	Digital Switch 1
	016	Digital Switch 2
	017	Digital Switch 4
	018	Digital Switch 8
Output	302	7 Segment 1
	303	7 Segment 2
	304	7 Segment 4
	305	7 Segment 8
	311	External Display Light 1
	312	External Display Light 2
	313	External Display Light 3
	314	External Display Light 4
	315	External Display Light 5
	316	External Display Light 6
	317	External Display Light 7
	318	External Display Light 8
	319	External Display Light 9

(3) Movement Description



(4) Application Program

Step	A/O	N	OP-Code	Operand1	Operand2	Post	Comment
1			TAG	1			
2			INB	15	1		Digital Switch Value into Variable 99
3			OUTB	302	1		Up-Take Value 7 Segment Output
4			CPEQ	99	1	601	When Up-Take Value = 1, Flag 601 ON
5		601	BTON	311			When Flag 601 ON, Lamp 1 ON
6		N601	BTOF	311			When Flag 601 OFF, Lamp 1 OFF
7			CPEQ	99	2	602	When Up-Take Value = 2, Flag 602 ON
8		602	BTON	312			When Flag 602 ON, Lamp 2 ON
9		N602	BTOF	312			When Flag 602 OFF, Lamp 2 OFF
10			CPEQ	99	3	603	When Up-Take Value = 3, Flag 603 ON
11		603	BTON	313			When Flag 603 ON, Lamp 3 ON
12		N603	BTOF	313			When Flag 603 OFF, Lamp 3 OFF
13			CPEQ	99	4	604	When Up-Take Value = 4, Flag 604 ON
14		604	BTON	314			When Flag 604 ON, Lamp 4 ON
15		N604	BTOF	314			When Flag 604 OFF, Lamp 4 OFF
16			CPEQ	99	5	605	When Up-Take Value = 5, Flag 605 ON
17		605	BTON	315			When Flag 605 ON, Lamp 5 ON
18		N605	BTOF	315			When Flag 605 OFF, Lamp 5 OFF
19			CPEQ	99	6	606	When Up-Take Value = 6, Flag 606 ON
20		606	BTON	316			When Flag 606 ON, Lamp 6 ON
21		N606	BTOF	316			When Flag 606 OFF, Lamp 6 OFF
22			CPEQ	99	7	607	When Up-Take Value = 7, Flag 607 ON
23		607	BTON	317			When Flag 607 ON, Lamp 7 ON
24		N607	BTOF	317			When Flag 607 OFF, Lamp 7 OFF
25			CPEQ	99	8	608	When Up-Take Value = 8, Flag 608 ON
26		608	BTON	318			When Flag 608 ON, Lamp 8 ON
27		N608	BTOF	318			When Flag 608 OFF, Lamp 8 OFF
28			CPEQ	99	9	609	When Up-Take Value = 9, Flag 609 ON
29		609	BTON	319			When Flag 609 ON, Lamp 9 ON
30		N609	BTOF	319			When Flag 609 OFF, Lamp 9 OFF
31			GOTO	1			Jump TAG 1
32							

Chapter 5. Option

1. I/O Card Unit (Model H-103)

Super SEL has as standard equipment, 24 inputs and 24 outputs. No PLC is necessary to control peripherals. The Super SEL can be expanded with up to 11 Expansion I/O units for a maximum of 288 inputs and 288 outputs.

Structure: photocoupler insulation. Output supports DC 24V 100mA (recommended value 20-50mA) . For larger loads and AC loads, a relay may be used. External DC-24V is necessary.

There is no problem adding additional I/O cards. To add additional I/O cards to a previously purchased SEL controller, please let IAI know the address of the expansion slot to be used.

External power supply for I/O cards:

Input Signal:	10mA/point
Output Signal:	30mA/point

For Example: 10 input signals/8 output signals.

10 x 10 + 30 x 8 = 340mA. Provide a DC 24V 0.5A power unit.

⌖High Speed Input Unit:
Each high speed input unit adds 32 input points but these are included when considering the maximum number of input points that can be used.

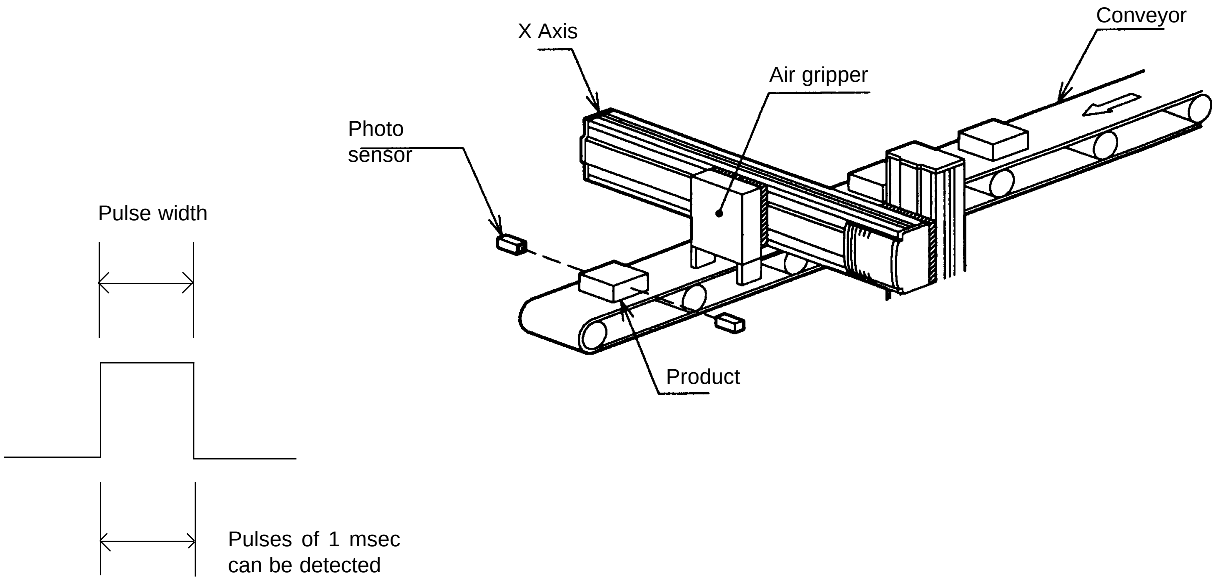
Note: The matrix to the right shows the second card's port numbers. For the nth card, the input port number begins from (N -1) x 24 + 24 and the output port number begins at (N -1) x 24 + 324.

Pin No.	Category	Port No.	Function	Wire Color
1A	P24		External Power +24V Input	1-Brown
1B	Input	24	User Input	1-Red
2A		25	User Input	1-Orange
2B		26	User Input	1-Yellow
3A		27	User Input	1-Green
3B		28	User Input	1-Blue
4A		29	User Input	1-Purple
4B		30	User Input	1-Gray
5A		31	User Input	1-White
5B		32	User Input	1-Black
6A		33	User Input	2-Brown
6B		34	User Input	2-Red
7A		35	User Input	2-Orange
7B		36	User Input	2-Yellow
8A		37	User Input	2-Green
8B		38	User Input	2-Blue
9A		39	User Input	2-Purple
9B		40	User Input	2-Gray
10A		41	User Input	2-White
10B		42	User Input	2-Black
11A		43	User Input	3-Brown
11B		44	User Input	3-Red
12A		45	User Input	3-Orange
12B		46	User Input	3-Yellow
13A		47	User Input	3-Green
13B	Output	324	User Output	3-Blue
14A		325	User Output	3-Purple
14B		326	User Output	3-Gray
15A		327	User Output	3-White
15B		328	User Output	3-Black
16A		329	User Output	4-Brown
16B		330	User Output	4-Red
17A		331	User Output	4-Orange
17B		332	User Output	4-Yellow
18A		333	User Output	4-Green
18B		334	User Output	4-Blue
19A		335	User Output	4-Purple
19B		336	User Output	4-Gray
20A		337	User Output	4-White
20B		338	User Output	4-Black
21A		339	User Output	5-Brown
21B		340	User Output	5-Red
22A		341	User Output	5-Orange
22B		342	User Output	5-Yellow
23A		343	User Output	5-Green
23B		344	User Output	5-Blue
24A		345	User Output	5-Purple
24B		346	User Output	5-Gray
25A		347	User Output	5-White
25B	N24		External Power Supply 0V	5-Black

2. High Speed Input Unit (Model H-104)

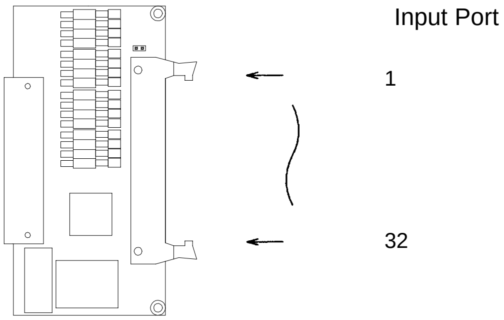
(1) What is the High Speed Input Unit?

In FA (Factory Automation), a high speed pulse may be required for some parts. For example, a photo sensor detects the presence of a part being carried down a conveyor at a very high speed. In these situations, the High Speed Input Unit can be used to detect these high speed signals and accurately transmit the signals to the Super SEL controller.

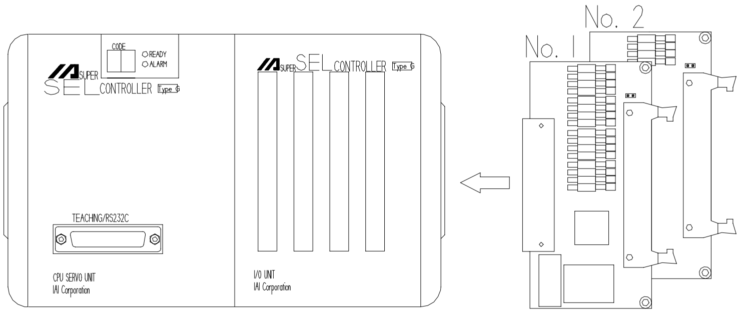


(2) Functions

1) The High Speed Input Unit offers up to 32 ports per unit.

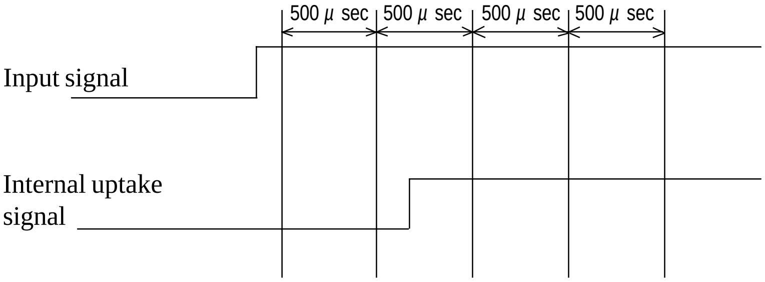


2) The "E" and "G" type Super SEL controllers can be expanded up to 2 units for a total of 64 ports.



3) The scan time (1 ~ 9 msec) within the detection pulse is also adjustable and can be set every 8 port groupings.

Movement when scan time is set to 1msec



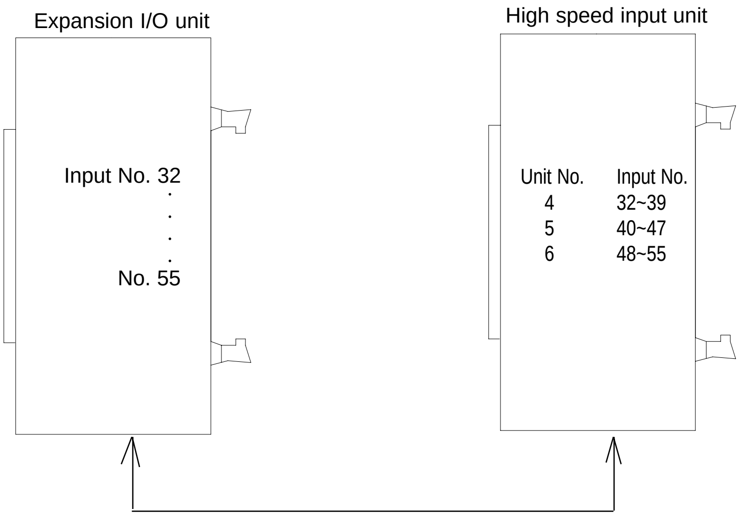
When the scan time is set at 1msec, it reads the input status for every 500 µ sec. If an input signal exists at the second 500 µ sec, it is recognized as a signal and transmitted to the inside.

4) The Super SEL controller can be expanded up to 287 input ports. They can be grouped into numbered units, one for every 8 ports. The High Speed Input ports can be assigned to any one of these.

Unit No.	Input Port No.
1	8 ~ 15
2	16 ~ 23
3	24 ~ 31
4	32 ~ 39
5	40 ~ 47
6	48 ~ 55
7	56 ~ 63
8	64 ~ 71
9	72 ~ 79
10	80 ~ 87
11	88 ~ 95
12	96 ~ 103
13	104 ~ 111
14	112 ~ 119
15	120 ~ 127
16	128 ~ 135
17	136 ~ 143
18	144 ~ 151
19	152 ~ 159
20	160 ~ 167
21	168 ~ 175
22	176 ~ 183
23	184 ~ 191
24	192 ~ 199
25	200 ~ 207
26	208 ~ 215
27	216 ~ 223
28	224 ~ 231
29	232 ~ 239
30	240 ~ 247
31	248 ~ 255
32	256 ~ 263
33	264 ~ 271
34	272 ~ 279
35	280 ~ 287

When input units are overlapping

Example



Unit No.	Input No.
4	32~39
5	40~47
6	48~55

Input No. 32 ~ 55 are overlapping. In this case, the high speed input unit goes into effect and the expansion I/O unit become invalid.

5) Program Example

WTXX Command

WTON 10 Responds within the parameter scan time.

6) Option

Please specify the following specifications at the time of order for this option so that the correct settings can be pre-set prior to shipment:

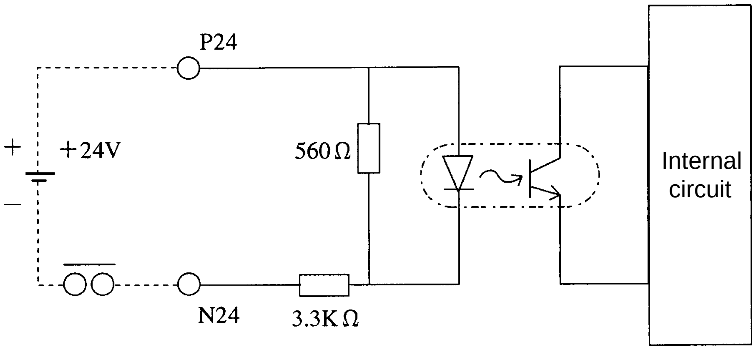
1. Number of H-104 units required
2. Input port number assignments
3. Scan time (1 ~ 9 msec) designations

Note: The above settings must be done through the PC interface software (supported by Ver.2.0 or later). They cannot be accomplished using the teaching pendant.

(3) Specifications

Input	Specifications
Point	Maximum Input....32 Points
Power Voltage	DC24V +/-20%
Current	7mA/DC24V
ON/OFF Power Voltage	ON....Min DC16.0V OFF....Max DC5.0V
ON/OFF Response Time	Varies (1msec ~ 9msec) depending on parameter
Isolation Method	Photocoupler

Internal Circuit



1) The power (DC24V) must be supplied externally to P24.
2) For the external circuit connection (no contact point), the leakage per 1 point must be kept lower than 1mA when the switch is OFF.

(4) Interface List

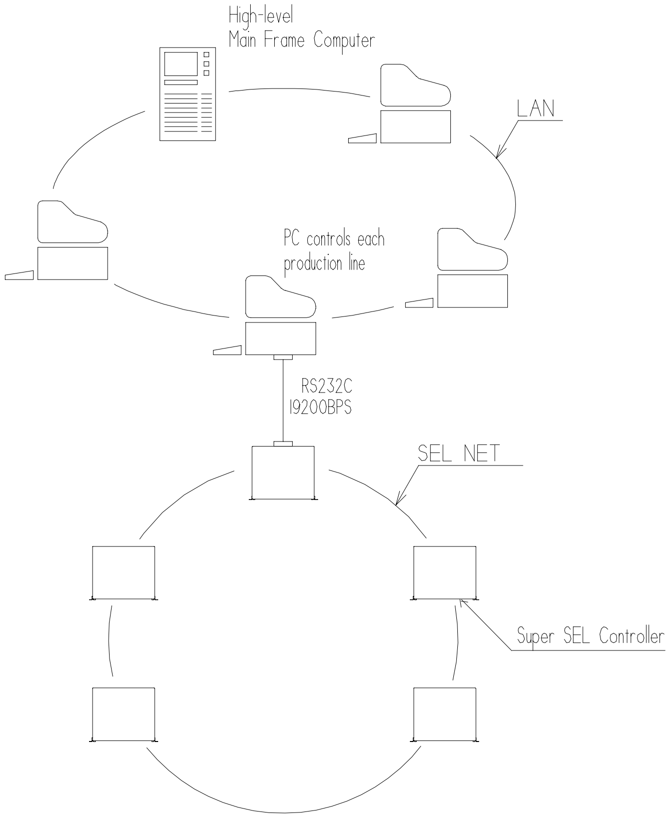
Pin No.	Category	Port No.	Function	Cable
1A	P24		---	1-Brown
1B	Input	000	User Input	1-Red
2A		001	User Input	1-Orange
2B		002	User Input	1-Yellow
3A		003	User Input	1-Green
3B		004	User Input	1-Blue
4A		005	User Input	1-Purple
4B		006	User Input	1-Gray
5A		007	User Input	1-White
5B		008	User Input	1-Black
6A		009	User Input	2-Brown
6B		010	User Input	2-Red
7A		011	User Input	2-Orange
7B		012	User Input	2-Yellow
8A		013	User Input	2-Green
8B		014	User Input	2-Blue
9A		015	User Input	2-Purple
9B		016	User Input	2-Gray
10A		017	User Input	2-White
10B		018	User Input	2-Black
11A		019	User Input	3-Brown
11B		020	User Input	3-Red
12A		021	User Input	3-Orange
12B		022	User Input	3-Yellow
13A		023	User Input	3-Green
13B		300	User Input	3-Blue
14A		301	User Input	3-Purple
14B		302	User Input	3-Gray
15A		303	User Input	3-White
15B		304	User Input	3-Black
16A		305	User Input	4-Brown
16B		306	User Input	4-Red
17A		307	User Input	4-Orange
17B	NC	308	---	4-Yellow
18A		309	---	4-Green
18B		310	---	4-Blue
19A		311	---	4-Purple
19B		312	---	4-Gray
20A		313	---	4-White
20B		314	---	4-Black
21A		315	---	5-Brown
21B		316	---	5-Red
22A		317	---	5-Orange
22B		318	---	5-Yellow
23A		319	---	5-Green
23B		320	---	5-Blue
24A		321	---	5-Purple
24B		322	---	5-Gray
25A		323	---	5-White
25B	N24		---	5-Black

3. SEL NET 2-Channel RS232C Unit (Model H-105)

The Super SEL Controller is equipped with a 1-channel RS232C port as a standard feature. This port is designed for communication with the Teaching Pendant or with a personal computer through our PC interface software. This port is not designed for general communication purposes. The H-105 option was developed for users who intend to develop their own programs and require communication with other devices via an RS232C port.

(1) What is SEL NET?
More and more manufacturers are turning to factory automation to improve manufacturing flexibility, increase efficiency and streamline their manufacturing operations. Today's factories abound with PLCs, robot controllers, sensors and other devices related to factory automation. These devices are typically connected by miles of cables and I/O wires. If all of these factory automation systems could be combined via network, a control system with a high level of efficiency could be built. Realizing this, the engineers at IAI developed what we call, SEL NET.

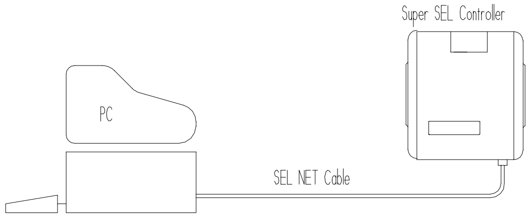
SEL NET Production Planning System
The following diagram shows a high-level main frame computer and several personal computers combined via a LAN. A single PC is used to communicate with a Super SEL Controller via RS232C. In addition, several Super SELs are combined via SEL NET and exchange synchronized or interlocked information among themselves. Although the main function of this sample system is an exchange of control information, other information such as production planning information could be shared.



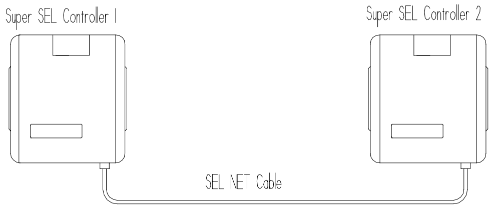
(2) SEL NET Functions

To achieve distributed control of a single production line using multiple controllers, usually I/O (input/output) signals are relayed to synchronize the individual controllers. But the SEL NET network does not use I/O signals. Instead, data is relayed between the controllers simply by connecting the controllers with the SEL NET cable. Data transmission is easily managed with SEL language programming. SEL NET allows you to build a simple system, a multiple controller network or a large scale network with high-level computers linked in a LAN network as described on the previous page.
The SEL NET option can be specified in two ways. It can be part of the main CPU module of the controller (as shown below and on page 209). This version uses a Nippon Molex 6 pin connector (not available for the Super SEL EU controller).

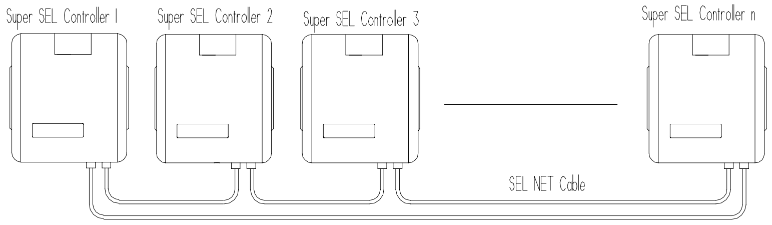
① Controller connected to a PC



② Controller connected to another controller



③ Multiple controllers connected

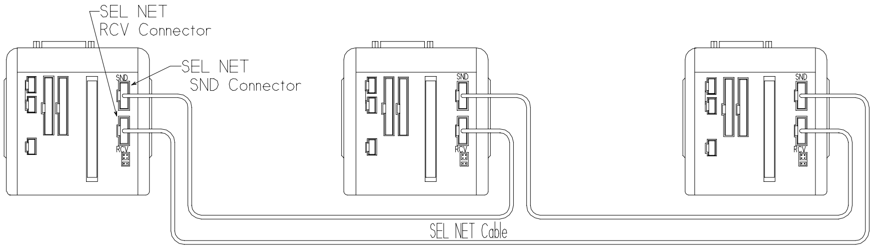


The second version is installed in the I/O expansion module. (See figures 1, 2, 3, 4 and 5 for various possible configurations) This version uses a standard 9 pin D-sub connector and is easily installed in the field. The cover to the I/O expansion module must be modified or replaced to allow access to the 9 pin connector.

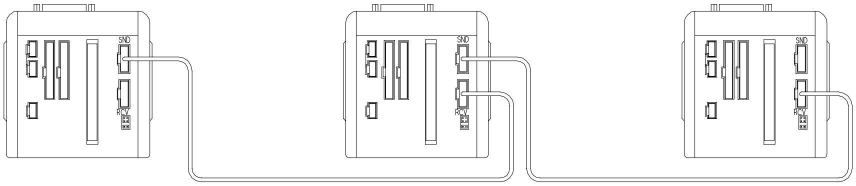
(3) SEL NET Structure · Connection

There is an [RCV] and an [SND] connector on the front panel of the Super SEL controller for connecting the SEL NET. The [RCV] and [SND] connectors for each controller are connected with the optional SEL NET cable as shown in the illustration below.

① Loop Connection



② Two-Way Connection



There are two methods of connecting controllers with SEL NET. The first diagram shows a loop connection while the second shows a two-way connection. You can program the network using either method but the two-way connection does not distinguish between the [SND] and [RCV] connectors as the data is both sent and received from one connector. In SEL NET, hardware settings are not required since the program manages all controller codes and data transmission formats. The only hardware preparation required is the connection of the SEL NET cables

(4) SEL NET Specifications

Item	Specification
Maximum no. of network nodes	No limitations (see note)
Maximum distance between stations	15m
Method of connecting the network	Loop connection · Two-way connection
Transmission speed	1200·2400·4800·9600·19200·38400BPS
Network Interface	SEL NET (Based on RS232C)

Note: There are no hardware limitations but when setting the number of nodes, take into consideration that the more nodes there are, the longer the transmission time.

(5) SEL NET cable (optional)

Please be sure to use the SEL NET cable which is offered as an optional accessory. The package includes cable for connecting the individual controllers and cable for connecting the controller to the PC.

Note: The SEL NET network uses the extended command set of Super SEL language which requires the PC interface software (DOS 2.0 or higher, Windows 1.0 or higher). (The extended commands cannot be used with the teaching pendant). For details on the extended commands, please refer to the instruction manual that comes with the Super SEL controller.

(6) The RS232C communication circuit has 2 channels and achieves high speed communications by using a dedicated CPU rather than the main CPU.

Figure 1 Between PC and Super SEL

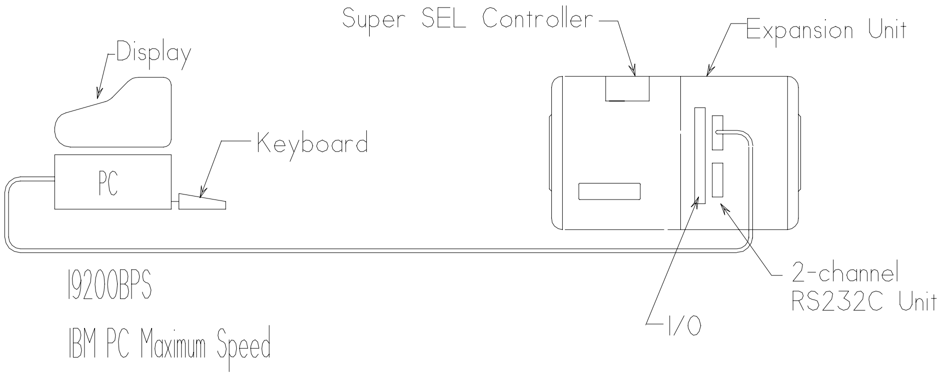


Figure 2 Between Super SEL and Super SEL

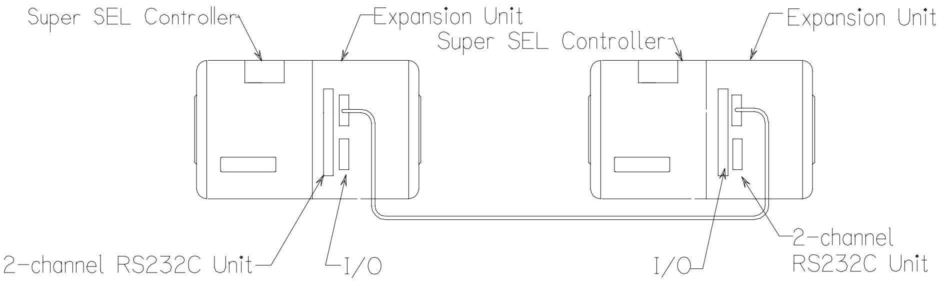
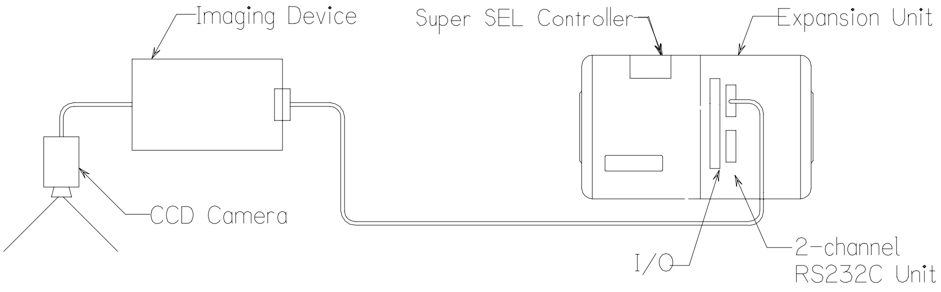


Figure 3 Between Other Devices and Super SEL



(7) System Using RS232C Communication Circuit

Figure 4 Connecting to a Vision Device

The vision device is used to check the position of the base board, and the mounter for odd-sized parts makes the adjustment of the position.

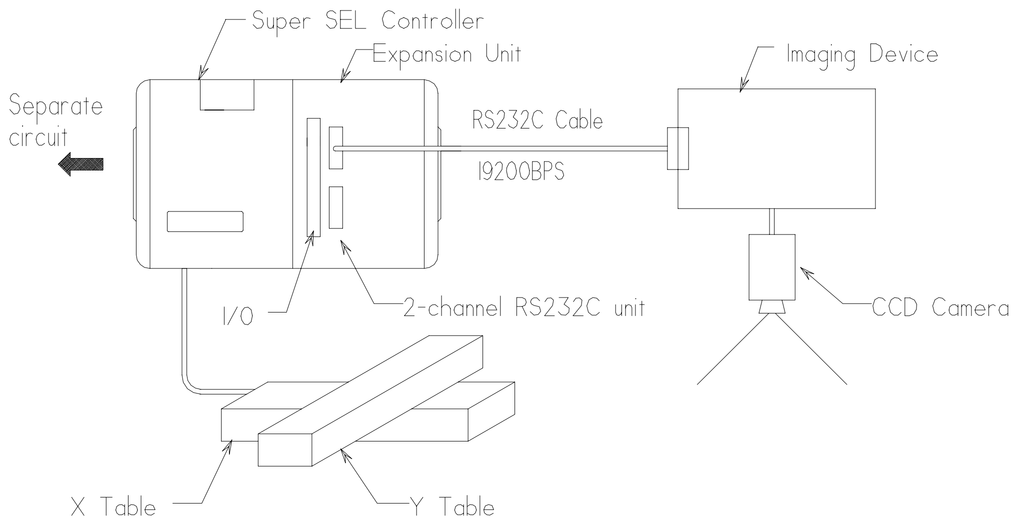
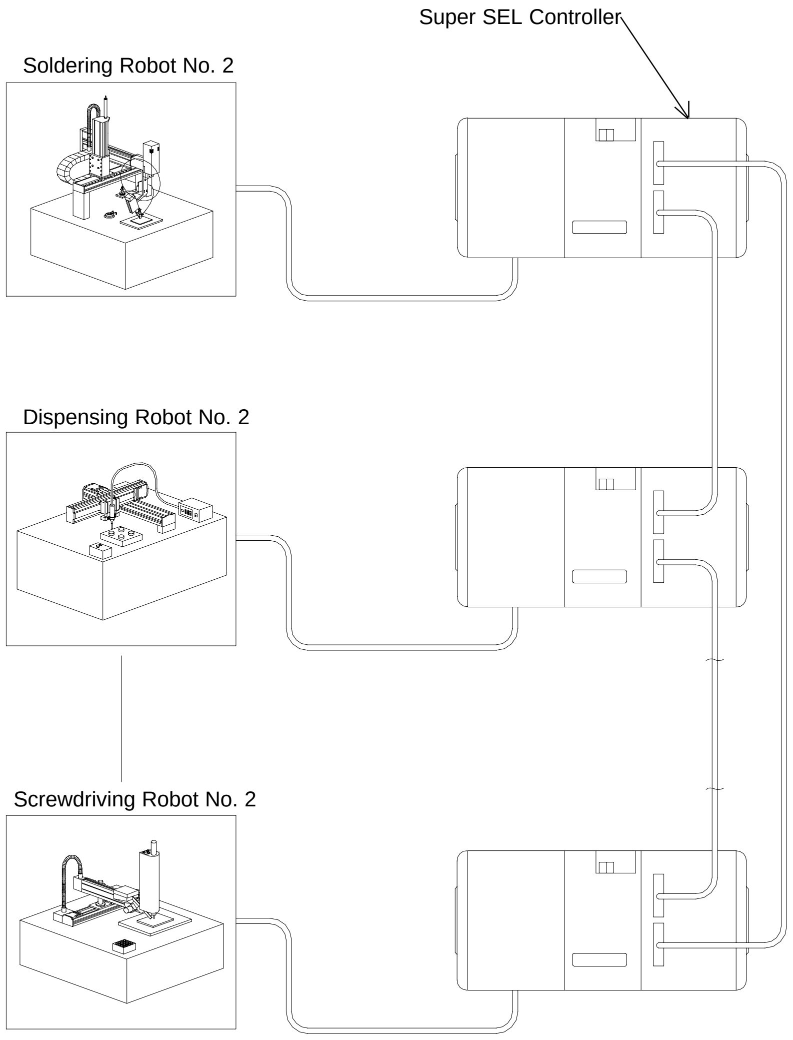


Figure 5 Multiple Super SEL Control System

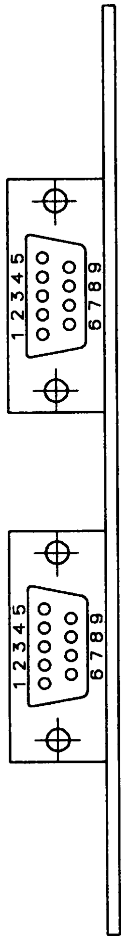


Control data is sent and received through a 1-to-1 loop connection (left) using the 2-channel RS232C.

(8) Specifications

		2-Channel RS232C Unit
Interface Specifications		Basing on RS-232C
Transmission Method		Full Duplex Method (No procedure)
Synchronous Method		Asynchronous Method
Transmission Velocity		1200, 2400, 4800, 9600, 19200, 38400bps
Data Form	Start Bit	1
	Data Bit	7/8
	Parity Bit	1/None
	Strip Bit	1/2
Error Detection		Parity (odd/even) / None
Transmission Distance		15m
Receive Buffer		512Byte
Connector		DELC-J9PAF-13L9 (JAE) X2

(9) Connector Pin Assignment



	No.	Function	
1 Channel	1	No connection	
	2	RD	
	3	TD	
	4	No connection	
	5	SG	
	6	No connection	
	7		(Connect)
	8		
	9	RI	
2 Channel	1	No connection	
	2	RD	
	3	TD	
	4	No connection	
	5	SG	
	6	No connection	
	7		(Connect)
	8		
	9	RI	

(10) Commands

Commands related to RS232C communication

Category	Function	Commands
External Input Output Command	Open Channel	OPEN
	Close Channel	CLOS
	Input from Channel	READ
	Output to Channel	WRIT
	Set Character of Sending-Receiving Signal Ending	SCHA
String Management Command	Copy Character String	SCPY
	Compare Character String	SCMP
	Character Acquisition	SGET
	Write in Character	SPUT
	Convert Character String (Decimal)	STR
	Convert Character String (Hexadecimal)	STRH
	Convert Number Value (Decimal)	VAL
	Convert Number Value (Hexadecimal)	VALH
	Set Length	SLEN

Note: These commands are expansion commands. Therefore, they must be used through the PC interface software (supported by DOS Ver.2.0 or later and Windows Ver 1.0 or later). They cannot be used with the Teaching Pendant.

① External Input Output Command

OPEN (Open Channel)

Expansion condition (AND · OR)	Input condition (I/O · Flag)	Command			Post (Output port · Flag)
		Command	Operand 1	Operand 2	
Optional	Optional	OPEN	Channel no.		

[Function] Opens the channel specified in Operand 1. Channels specified after this will be able to transmit and receive signals. An ending character must be set by the SCHA command before executing this command.

[Example]

```
SCHA   10
OPEN   1
       Designate 10 (=LF) as the ending character.
       Open channel 1.

SCHA   13
LET    1       2
OPEN   *1
       Designate 13 (=CR) as the ending character.
       Assign 2 to variable 1.  Open channel 2, the value contained in variable 1.
```

CLOS (Close Channel)

Expansion condition (AND · OR)	Input condition (I/O · Flag)	Command			Post (Output port · Flag)
		Command	Operand 1	Operand 2	
Optional	Optional	CLOS	Channel no.		

[Function] Closes the channel specified in Operand 1. Channels specified after this will be unable to transmit and receive signals.

[Example]

```
CLOS   1
       Close the channel.
```

[Example]

```
LET    1       2
CLOS   *1
       Assign 2 to variable 1.
       Close channel 2, the value contained in variable 1.
```

READ

Expansion condition (AND · OR)	Input condition (I/O · Flag)	Command			Post (Output port · Flag)
		Command	Operand 1	Operand 2	
Optional	Optional	READ	Channel no.	Column no.	

[Function] Reads the character string from the channel in Operand 1 to the column in Operand 2.
Stops reading when the character designated in the SCHA command appears.
The column can be either local or global.

[Example]

```
SCHA    10
OPEN    1
READ    1       2
CLOS    1
```

Set LF (= 10) for the ending character.
Open channel 1.
Read the character string from channel 1 to column 2 until LF appears.
Close the channel.

```
LET     1       2
LET     2       3
SCHA    13
READ    *1      *2
```

Assign 2 to variable 1.
Assign 3 to variable 2.
Set CR (= 13) for the ending character.
Read the character string from channel 2 (content of variable 1) to column 3 (content of variable 2) until CR appears.

WRIT (Write)

Expansion condition (AND · OR)	Input condition (I/O · Flag)	Command			Post (Output port · Flag)
		Command	Operand 1	Operand 2	
Optional	Optional	WRIT	Channel no.	Column no.	

[Function] Writes the character string from the channel in Operand 1 to the column in Operand 2.
Stops writing after the character designated in the SCHA command is written.
The column can be either local or global.

[Example]

```
SCHA    10
OPEN    1
READ    1       2
CLOS    1
```

Set LF (= 10) for the ending character.
Open channel 1.
Write the character string up to LF from channel 1 to column 2.
Close the channel.

```
LET     1       2
LET     2       3
SCHA    13
READ    *1      *2
```

Assign 2 to variable 1.
Assign 3 to variable 2.
Set CR (= 13) for the ending character.
Write the character string up to LF from channel 2 (content of variable 1) to column 3 (content of variable 2).

② String Management Commands

String refers to a character string and the Super SEL controller has global and local strings. The global strings can be read or written in all programs. Local strings can be used *only* within that particular program and not in any other program. Global and local strings are distinguished by their numbers: global strings range from 300~399 and local strings range from 1~299.

One of the requirements of string commands is that communications with general external devices (PC, controller) must be done through serial communications. The serial communication data must be processed as a string which can be done by comparing, moving, or exchanging the strings which contain the serial communication data. The string commands are available for the Super SEL Type E & G controllers.

SCPY (Copy Character String)

Expansion condition (AND · OR)	Input condition (I/O · Flag)	Command			Post (Output port · Flag)
		Command	Operand 1	Operand 2	
Optional	Optional	SCPY	Column no.	Column no. Literal character	

[Function] Copies the character string from the column in Operand 2 to the column in Operand 1.
Copies only the length set by the SLEN command.
When Operand 2 is a literal character, that is the length copied.

[Example]

```
SCPY    1       'ABC'
        Copy 'ABC' to column 1.

SLEN    10
SCPY    100     200
        Set the length of the operation to 10 bytes.
        Copy 10 bytes from column 200 to column 100.

LET     1       300
LET     2       400
SLEN    5
SCPY    *1      *2
        Assign 300 to variable 1.
        Assign 400 to variable 2
        Set the length of the operation to 5 bytes.
        Copy 5 bytes from column 400 (the content of variable 2) to column 300 (the content of variable 1).
```

SCMP (Compare Character String)

Expansion condition (AND · OR)	Input condition (I/O · Flag)	Command			Post (Output port · Flag)
		Command	Operand 1	Operand 2	
Optional	Optional	SCMP	Column no.	Column no. Literal character	Required

[Function] Compares the column in Operand 1 and the column in Operand 2.
Compares only the length set by the SLEN command.
When Operand 2 is a literal character, that is the length compared.

[Example]

```
SCMP    1       'ABC'   600
```

When column 1 ~ 3 are 'ABC', flag 600 turns ON.

```
SLEN    5
SCMP    10      30      999
```

Set the length to be compared to 5 bytes.
When the 5 bytes from column 10 and column 30 are equal, flag 999 turns ON.

```
LET     1       10
LET     2       20
SLEN    3
SCMP    *1      *2      310
```

Assign 10 to variable 1.
Assign 20 to variable 2
Set the length to be compared to 3 bytes.
When the 3 bytes in column 10 (the content of variable 1) and the 3 bytes in column 20 (the content of variable 2) are equal, then output 310 turns ON.

SGET (Acquire Character String)

Expansion condition (AND · OR)	Input condition (I/O · Flag)	Command			Post (Output port · Flag)
		Command	Operand 1	Operand 2	
Optional	Optional	SGET	Variable no.	Column no.	

[Function] Assigns 1 character from the column in Operand 2 to the variable in Operand 1.

[Example

```
SGET    1       100
        Assign 1 byte of column 100 to variable 1.

LET     1       3
LET     2       1
SCPY    1       'A'
SGET    *1      *2
        Assign 3 to variable 1.
        Assign 1 to variable 2.
        Copy 'A' to column 1.
        Assign 'A' in column 1 (content of variable 2) to variable 3 (content of variable 1).
```

SPUT (Set Character)

Expansion condition (AND · OR)	Input condition (I/O · Flag)	Command			Post (Output port · Flag)
		Command	Operand 1	Operand 2	
Optional	Optional	SPUT	Column no.	Data	

[Function] Sets the data in Operand 2 to the column in Operand 1.

[Example]

```
SPUT    5       10
        Set 10 (LF) to column 5.

LET     1       100
LET     2       50
SPUT    *1      *2
        Assign 100 to variable 1.
        Assign 50 to variable 2.
        Set 50 (' 2') which is the content of variable 2 to column 100 (content of variable 1).
```

STR (Change Character String Decimal)

Expansion condition (AND · OR)	Input condition (I/O · Flag)	Command			Post (Output port · Flag)
		Command	Operand 1	Operand 2	
Optional	Optional	STR	Column no.	Data	

[Function] Copies the data in Operand 2 which has been converted to a decimal character string to the column in Operand 1. Uses zero-suppress to match this to the length set by the SLEN command.
Even if the data is longer than the length, the length set by the SLEN command takes precedence.

[Example]
```
SLEN    5. 3
        STR     1       123
```
Set the length to a 5 digit integer with 3 decimals.
The following will be set in column 1~9,

1	2	3	4	5	6	7	8	9
		1	2	3	.	0	0	0

```
LET     1       10
LET     2       987. 6543
SLEN    2. 3
STR     *1      *2
```
Assign 10 to variable 1.
Assign 987. 6543 to variable 2.
Set the length to a 2 digit integer with 3 decimals.
The following will be set in column 10~15,

10	11	12	13	14	15
8	7	.	6	5	4

. Since the data was longer than the set length, 9 in the 100s place and 3 in the 4th decimal place are cut off.

STRH (Change Character String Hexadecimal)

Expansion condition (AND · OR)	Input condition (I/O · Flag)	Command			Post (Output port · Flag)
		Command	Operand 1	Operand 2	
Optional	Optional	STR	Column no.	Data	

[Function] Copies the data in Operand 2 which has been converted to a hexadecimal character string to the column in Operand 1. Uses zero-suppress to match only the integers to the length set by the SLEN command. Even if the data is longer than the set length, the setting by the SLEN command will take precedence.

[Example]

```
SLEN   5
STRH   1      255
```

Set format for a 5 digit integer.
The following will be set in column 1~5,

1	2	3	4	5
			F	F

```
LET    1      10
LET    2      987. 6543
SLEN   2. 3
STRH   *1     *2
```

Assign 10 to variable 1.
Assign 987. 6543 to variable 2.
Set format for a 2 digit integer with 3 decimals.
The following will be set in column 10~11,

10	11
D	B

.3, the decimal segment of the SLEN command, and .6543 in variable 2 will be ignored. The integer expressed in hexadecimal notation is ' 3DB'. However, 3 in the third digit will be cut off since the length is set to 2 digits.

VAL (Character String Change Data Decimal)

Expansion condition (AND · OR)	Input condition (I/O · Flag)	Command			Post (Output port · Flag)
		Command	Operand 1	Operand 2	
Optional	Optional	VAL	Variable no.	Column no.	

[Function] Converts the decimal data in the column in Operand 2 to a binary number and assigns this to the variable in Operand 1.
The length set by the SLEN command will be converted.

[Example]

```
SCPY    10      ' 1234'
SLEN    4
VAL     1       10
```

Set ' 1234' in column 10.
Set the length to 4 bytes.
' 1234' in column 10 is converted to the binary number 1234 and assigned to variable 1.

```
LET     1       100
LET     2       20
SCPY    20      ' 1234'
SCPY    24      '. 567'
SLEN    8
VAL     *1      *2
```

Assign 100 to variable 1.
Assign 20 to variable 2
Copy ' 1234' to column 20.
Copy '. 567' to column 24.
Set the length to 8 bytes.
' 1234. 567' in column 20 (content of variable 2) will be converted to the binary number 1234. 567 and assigned to variable 100 (content of variable 1).

VALH (Character String   Change Data   Hexadecimal)

Expansion condition (AND · OR)	Input condition (I/O · Flag)	Command			Post (Output port · Flag)
		Command	Operand 1	Operand 2	
Optional	Optional	VALH	Variable no.	Column no.	

[Function] Converts the hexadecimal data in the column in Operand 2 to a binary number and assigns this to the variable in Operand 1.
The length set by the SLEN command will be converted. Only the integers will be converted and the decimal places will be disregarded.

[Example]

```
SCPY    10        ' 1234'
SLEN    4
VAL     1         10
```

Set ' 1234' in column 10.
Set the length to 4 bytes.
The hexadecimal number ' 1234' in column 10 is converted to the binary number 4660 and assigned to variable 1.

```
LET     1         100
LET     2         20
SCPY    20        '  ABCD'
SLEN    4
VALH    *1        *2
```

Assign 100 to variable 1.
Assign 20 to variable 2
Copy ' ABCD' to column 20.
Set the length to 4 bytes.
The hexadecimal ' ABCD' in column 20 (content of variable 2) will be converted to the binary number 43982 and assigned to variable 100 (content of variable 1).

SLEN (Set Length)

Expansion condition (AND · OR)	Input condition (I/O · Flag)	Command			Post (Output port · Flag)
		Command	Operand 1	Operand 2	
Optional	Optional	SLEN	Length		

[Function] Sets the length for the string command.
The length must be set prior to using any of the following commands.

SCMP	· · ·	Decimals Invalid
SCPY	· · ·	Decimals Invalid
ISXX	· · ·	Decimals Invalid
STRH	· · ·	Decimals Invalid
VAL, VALH	· · ·	Decimals Invalid
STR	· · ·	Decimals Valid

[Example] Refer to each of the commands above.

SCHA (Set Ending Letter)

Expansion condition (AND · OR)	Input condition (I/O · Flag)	Command			Post (Output port · Flag)
		Command	Operand 1	Operand 2	
Optional	Optional	SCHA	Character code		Optional

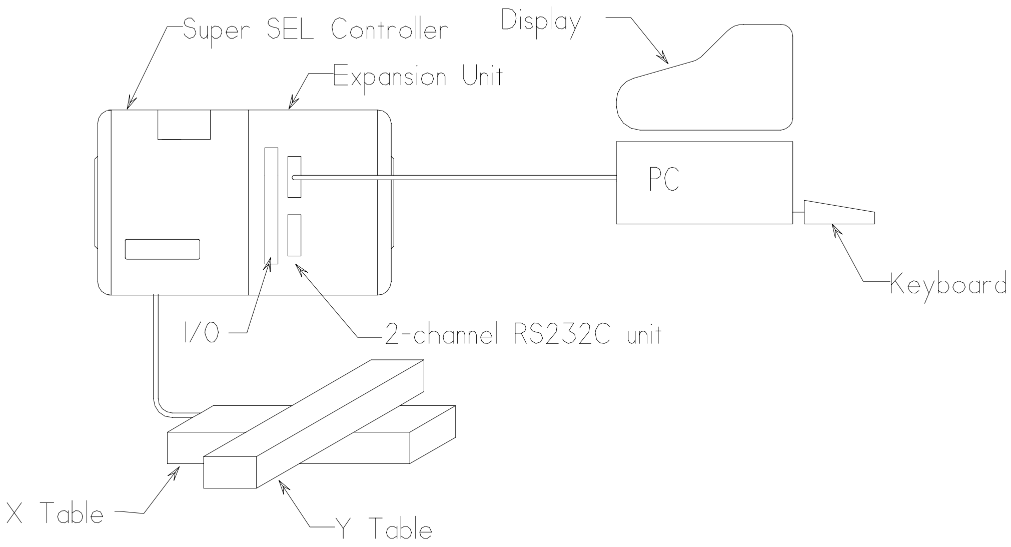
[Function] Sets the ending letter to be used in the READ command and WRIT command.
A value from 0 ~ 255 (character code used in BASIC) can be designated for the character.

[Example] Refer to the READ command and WRIT command.

(11) Application Program

Example of an application program using an RS232C unit:

The controller and the PC are connected to the RS232C communication circuit. Homing is performed and the motion is controlled by the MOVE command via PC. When the movement is complete, a signal is sent to the PC.

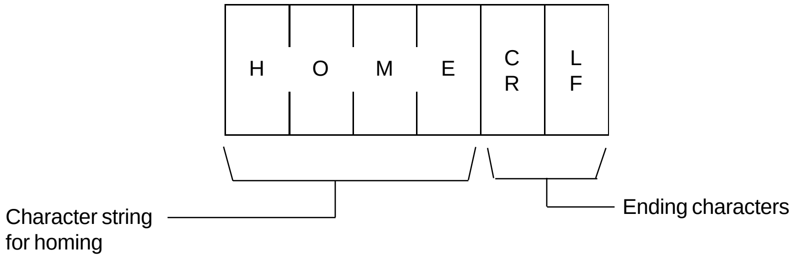


① Transmission Format

Communication in this system is carried out by exchanging character strings. Every character string is determined toperform a certain motion. When the receiver receives a character string, it recognizes it and performs the corresponding motion. This character string and the ending characters are combined to make up a transmission format, which can be determined by the user.

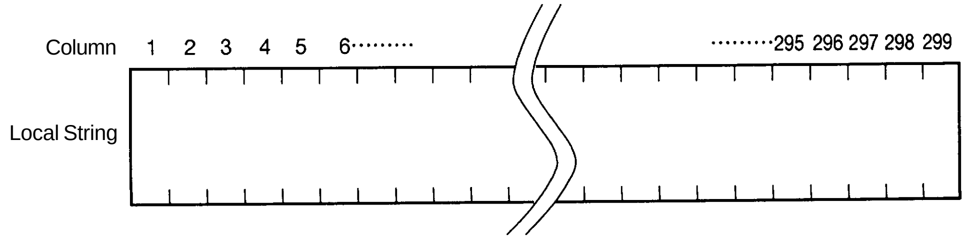
Example: For the Home command, make up a character string with 4 characters, "HOME". Then, determine the ending characters to indicate the end of the command. When N88 BASIC is used for the PC side, "CR" and "LF" must be used for the BASIC side.

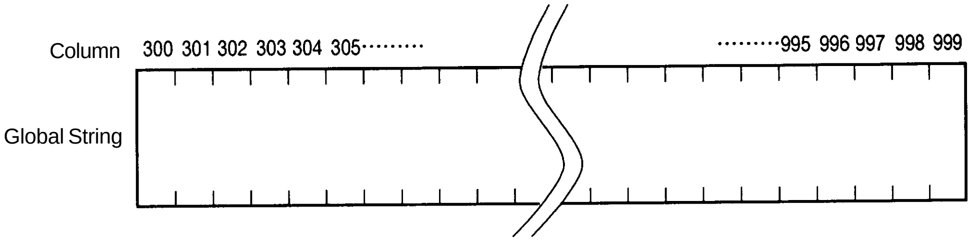
Transmission format



② String

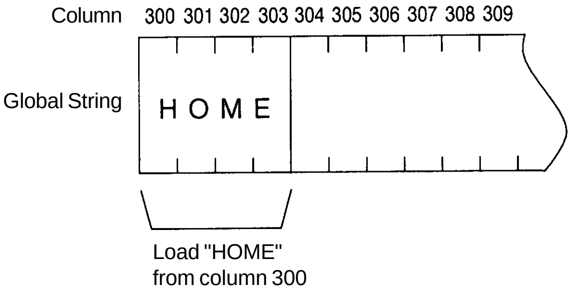
A character string in a transmission format is placed in a column box called "String" in order to be used in programs. There are two types of strings: global strings can be used in all programs and local strings can be used *only* in individual programs. They are distinguished by the column numbers.





Column numbers are used to specify the positions of the cells so that the columns can be easily set by commands.

Example: When a character string for "HOME" is received from PC, it is stored in the columns starting from Column 300 to be used in several programs.



③ Determination of Transmission Format

There are three types of transmission formats required for the application program. (These are only examples and they can be set differently by the user.)

The PC side is programmed with N88 BASIC.

- **Home Command Format**
  This is a format to designate homing to the controller from PC.

H	O	M	E	CR	LF

- **Move Command Format**
  This is a format to designate axis movement to the controller from the PC.

M	O	V	E	SPEED			AXIS NO. 1 POSITION								AXIS NO. 2 POSITION							CR	LF
				9	9	9	9	9	9	.	9	9	9	9	9	9	9	.	9	9	9		

- **Motion Complete Format**
  This is a format to send a signal from the controller to the PC after completion of homing and movement.

O	K	CR	LF

④ Procedure

This procedure explains the process for programming the application examples.

1) Set characters (terminating characters) "LF" to indicate the ending of the character string.
2) Open channel 1to use channel 1 in the RS232C unit.
3) If channel 1 receives a signal, the first column of the local string takes the signal in.
4) When the received data is "HOME", Axis-2 performs homing. After homing is completed, an "OK" signal will be sent back.
5) When the received data is "MOVE", the velocity data is converted into binary notation and set in variable 10. The position data is also converted into binary notation and set in position no.1. When the motion is completed, an "OK" signal will be sent back.

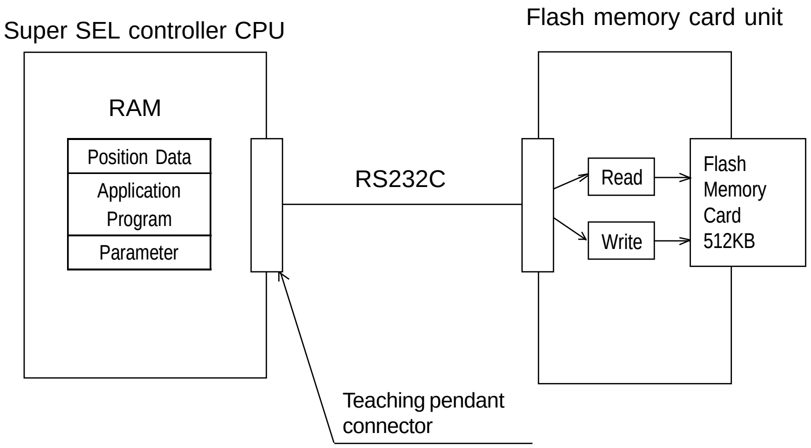
⑤ Application Program

Step	No.	N	Op-Code	Operand 1	Operand 2	Post	Comment
1			SCHA	10			Set LF as terminating characters
2			OPEN	1			Open SIO Channel 1
3			TAG	1			
4			READ	1	1		Read in one line from SIO
5							
6			ISEQ	1	'HOME'		HOME command
7			HOME	11			Homing
8			EXSR	1			Send OK signal
9			EDIF				
10							
11			ISEQ	1	'MOVE'		If a MOVE command
12			SLEN	3			3 lines for length
13			VAL	10	5		Velocity → Variable 10
14			VEL	*10			Set velocity
15							
16			PCLR	1	1		Clear position 1
17			SLEN	3.3			
18			VAL	199	8		Axis 1 position → Variable 199
19			PPUT	1	1		Set 1 data
20							
21			VAL	199	15		Axis 2 position → Variable 199
22			PPUT	2	1		Set axis 2 data
23			MOVL	1			Move
24			EXSR	1			Send OK signal
25			EDIF				
26							
27			GOTO	1			
28							— — — — — — — — — — —
29			BGSR	1			Send OK signal subroutine
30			SCPY	1	'OK'		Set OK
31			SPUT	3	13		Set CR
32			SPUT	4	10		Set LF
33			WRIT	1	1		Send signal
34			EDSR				

4. Flash Memory Card Unit (Model H-106)

The Flash Memory Card Unit itself does not need a backup battery and keeps data almost indefinitely. The Flash Memory Card Unit loads and saves all data, including application programs, position data, and parameters

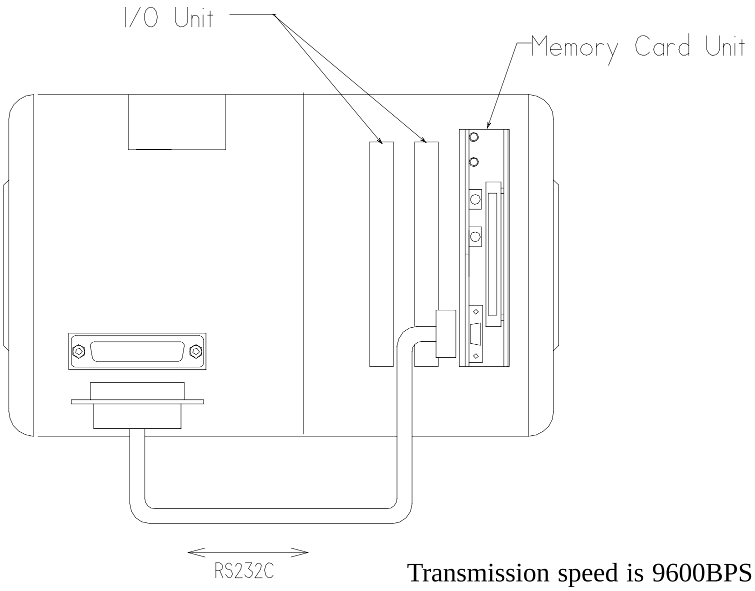
Diagram of complete system



(1) Function

1) Communications between the Super SEL CPU and the Flash Memory Card

The Super SEL Controller CPU communicates with the Flash Memory Card via the RS232C port (standard teaching pendant connector).



Transmission speed is 9600BPS

2) Save/Load Time

Although it takes time to load and save information via RS232C, there is an advantage of loading all data as a batch.

Time Required

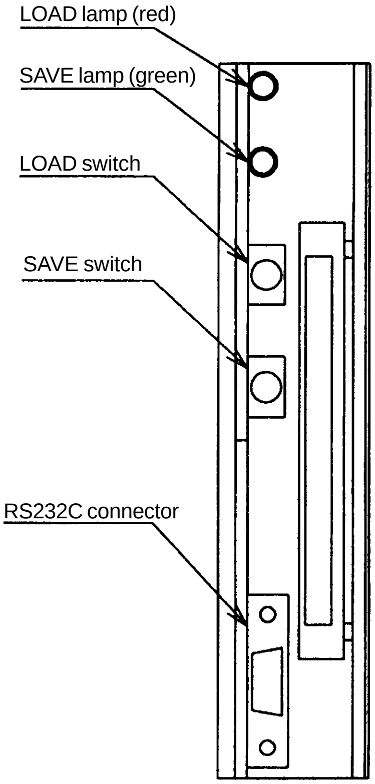
- Save

	Data Capacity	Time (Second)	
Position	6 Axis 2000 Points	220	
Program	3000 Steps	290	
Parameter	All Parameters	4	
Delete		72	
		Total	586

- Load
  Requires 10% longer than the saving time.

(2) Operation

Flash Memory Card Unit

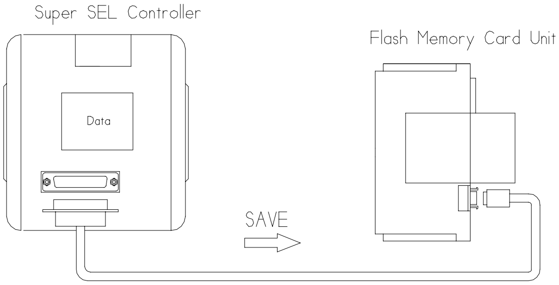


1) When the power is turned ON,

The SAVE LOAD lamp blinks 5 times to indicate the power has been turned ON. During this time, any button operations cannot be performed.

2) SAVE

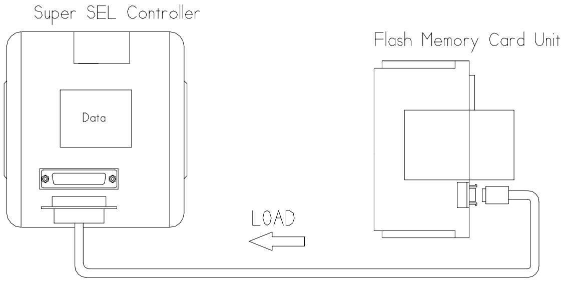
"SAVE" takes up data from the Super SEL Controller to the Flash Memory Card.



Start by pressing the SAVE button. When the data is being saved, normally the SAVE lamp blinks. When data is already written in the Memory Card, the existing data will be deleted and the data sent from the controller will be written in.Data can be saved while an application program in the controller is running.

3) LOAD

"LOAD" sends data to the Super SEL Controller from the Flash Memory Card.



Start loading by pressing the LOAD Button. When the data is being loaded, normally the LOAD lamp blinks. When there is no data in the Flash Memory Card, nothing will be loaded (If data with nothing written is sent to the controller, all data in the controller will be deleted). Data cannot be loaded while an application program in the controller is running.

(3) Error

Errors will be checked after the SAVE LOAD buttons are pressed.

1) Communication Error
Indicates an abnormality in the communication circuit. Both lamps (SAVE LOAD) blink approximately every second. The cable should be inspected.

2) Write-in Error
The SAVE lamp blinks approximately every second. The Flash Memory Card itself may be the problem. The Flash Memory Card should be replaced.

3) No Card Error
This error occurs when the SAVE or LOAD operation is tried without a Flash Memory Card. The SAVE or LOAD lamp blinks approximately every 0.5 second. A Flash Memory Card must be inserted.

4) Releasing Error
To reset back to the initial status, press either the SAVE or the LOAD Button.

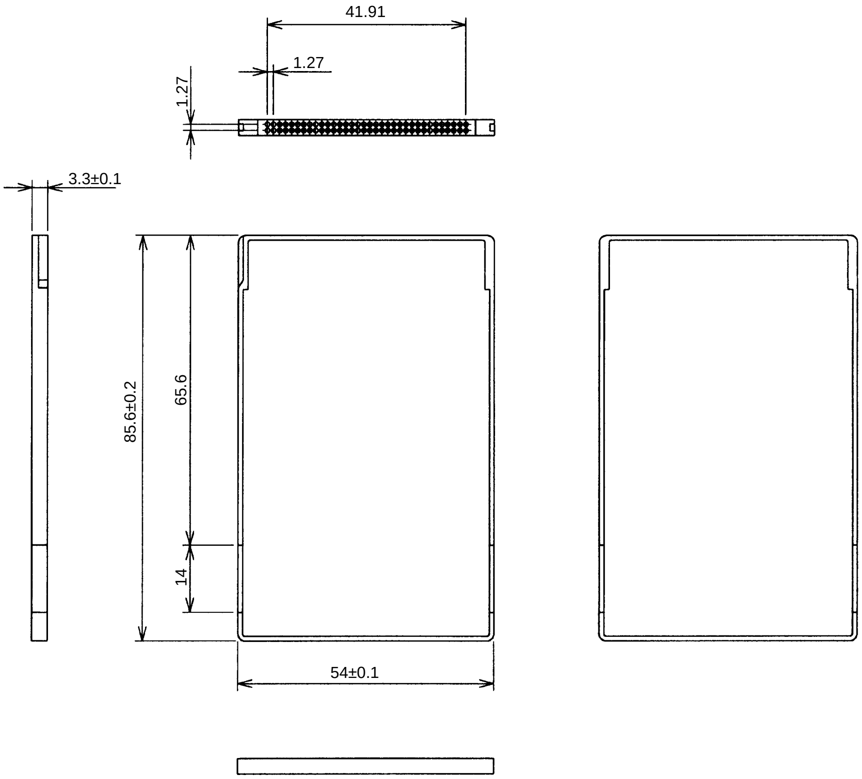
(4) Specifications

1) Flash Memory Card

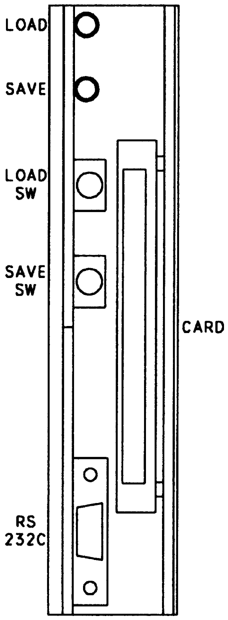
	Description
Memory Type	FLASH-EPROM
Memory Capacity	FLASH512 512Bytes
External Dimensions	85.6±0.2 x 54.0±0.1 x 33±0.1(mm)
Connector	Two piece (JEIDA Specification) Pitch 1.27mm 68 pins (2 rows)
Applicable Specification	JEIDA/PCMCIA (Direct bus method)
Data Bit Length	8 bit/16 bit
Power Supply Voltage	+5V±5% (MAX 6V)
Weight	50g(MAX)

2) Flash Memory Card Unit

① Card External Dimensions



② Unit Outline Drawing



5. PC Interface Software (Model H-101C to 101MW)

The Super SEL Controller is the culmination of many years of experience in motion control. The Super SEL Controller is used not only to control servo actuators, but also to control peripheral equipment. The PC interface software undergoes continuous improvement. At the time of this publishing, the Super SEL Controller offers over 120 commands.

In the past, it was enough to use a hand-held teaching pendant for programming. This was possible because of the fewer commands available at that time. Since then, a greater number of commands are now available. For this reason, the teaching pendant, in some instances, may not be the best choice. The PC interface software was developed as a more convenient method of programming. The Super SEL PC interface software offers the following features:

- Available for use with all available Super SEL commands.
- Ability to load and unload controller parameters.
- Ability to write application programs and position data and saving this data.
- Ability to debug multi-tasking programs (multi-tasking monitor).
- Ability to view I/O conditions with monitoring functions.
- Ability to install software to most personal computers:

DOS Version (for IBM compatible)	Model H-101-M
Windows Version (FD 1.4M)*	Model H-101-MW

It is easier to use the multi-tasking functions of the Super SEL Controller through the PC interface software.

Note: Certain Super SEL Controller functions may not be available in older versions of the PC interface software. PC interface software DOS version 2.0 and Windows version 1.0 and later can utilize all available functions.

For further details, please refer to the separate manual that comes with the PC interface software.

6. I/O Expansion Module - 2 Models

The number of expansion slots on the Super SEL Controller varies according to the controller model. There are 2 types of optional expansion units available. Model H-107-4 has four (4) expansion slots while Model H-107-12 has twelve (12) expansion slots.

1 **E-Type** Controller (1-Axis): Up to 100W
G-Type Controller (2-Axis): 50W 2 axes · 100W 2 axes

Example

The following example shows an E-Type or G-Type Controller with a 4-slot I/O expansion unit. Note that in this example, one of the expansion slots may be used for the 2-channel RS232C option (H-105) as follows:

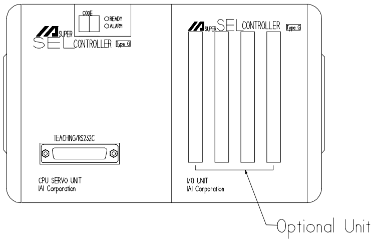
Expansion I/O Card Unit -	3
2-Channel RS232C Unit -	1
	4 slots

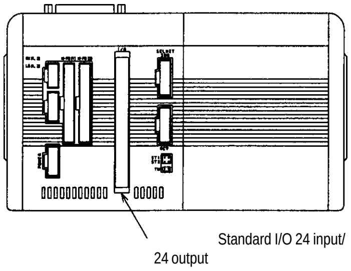
(I/O expansion unit with the 4 slot specification is connected to the CPU servo unit).

Note: On this model of the E-Type and G-Type Super SEL Controller, the standard 24/24 I/O are available on the CPU servo unit itself. The optional I/O expansion unit need only be purchased if additional I/O are required.

Note: It is not possible to add more than one 4-channel expansion module to a single CPU servo unit. In addition, only one 2-channel RS232C option can be added to each SEL Controller.

Note: As illustrated in the drawing below, the E-Type and G-Type Controller for the smaller 60W or 100W actuators differs from the model designed to handle the larger 100W, 200W, and 400W actuators (See illustration in next example).

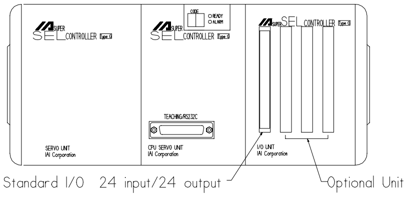




2 **E-Type** Controller (1-Axis) for 200W or 400W;
G-Type Controller (up to 4 Axes) for any combination of 100W, 200W or 400W.

Example:

The illustration below shows the larger E-Type and G-Type Controller designed to handle the larger 100W, 200W and 400W actuators. This model has a separate servo unit and CPU unit.
In this model, the I/O expansion unit itself (which is an option in the smaller 60W and 100W model) is standard with one of the 4 slots used to house the standard 24/24 I/O board. As shown in the drawing below, the other 3 slots on the I/O expansion unit are available to accept optional expansion boards. Maximum number of I/O: 24/24.

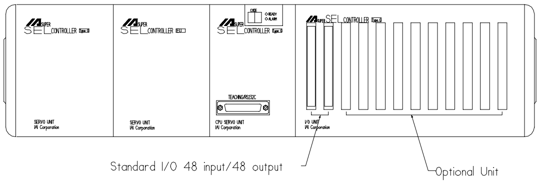


3 **G-Type** Controller

Example:

The illustration below shows the larger G-Type Controller. Although outwardly, the illustration below could represent several different G-Type controller configurations, this specific example shows a 2-Axis, G-Type Controller with 2 servo units: 400W and 200W. Again, although not indicated in the illustration, the 400W servo unit would be located to the extreme left of the CPU unit to reduce noise interference.

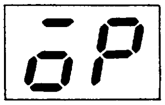
Note: The 12-slot I/O expansion unit offers 48/48 standard I/O (2 slots) with the remaining 10 slots used for optional expansion boards. Maximum number of I/O: 288/288



*Supplement

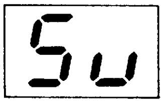
1. Super SEL Controller 7 Segment Display

(1)



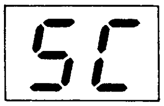
Open (Always appears at the beginning). Goes to the next step (2) after 500msec.

(2)



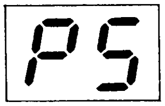
Opens SIO (Serial IO). If under EMG condition, waits till EMG is released. Goes to the next step (3) if no problem.

(3)



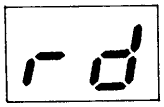
Check servo. Initialize the motor processor. Waits 500msec after initializing, then goes to the next step (4).

(4)



Sets up the position data and program data. Waits 500msec after setting, then goes to the next step (5).

(5)



Indicates normal status during power up under stand-by condition.

Each display indicates the "progress status of the initial setting", and it is not an alarm. However, switching the controller power supply for a very short period of time could cause this process to stop. This is a hardware related problem and nothing unusual. This problem should be corrected by turning the power back on after 15 seconds or longer.

*Supplement

2. **Power Required by the Super SEL Controller** (Manual display & method of calculation)

Power values written in the manual indicate rated power (effective power).

The required power is calculated based on the power consumption below.

• AC Specifications

			Power Consumption	
Controller			50W	at maximum standard (without I/O expansion)
			100W	at maximum with I/O expansion
AC Motor	Axis output	60W type	75W	at rated output
	Axis output	100W type	125W	at rated output
	Axis output	200W type	250W	at rated output
	Axis output	400W type	500W	at rated output

[Example] : For a controller with 60w + 100w specifications, (standard)

$$\underbrace{50}_{A} + \underbrace{(75 + 125)}_{B} = 250W$$

The calculated value is the required effective power when each axis is used at rated power.

The required power is calculated based on the power consumption below.

• DC Specifications

			Power Consumption	
Controller			50W	at maximum standard (without I/O expansion)
			100W	at maximum with I/O Expansion
DC Motor	Axis output	20W type	60W	at rated output
	Axis output	30W type	60W	at rated output
	Axis output	35W type	60W	at rated output
	Axis output	60W type	110W	at rated output
	Axis output	100W type	160W	at rated output
	Axis output	200W type	270W	at rated output

[Example] : For a controller with 60w + 100w specifications, (standard)

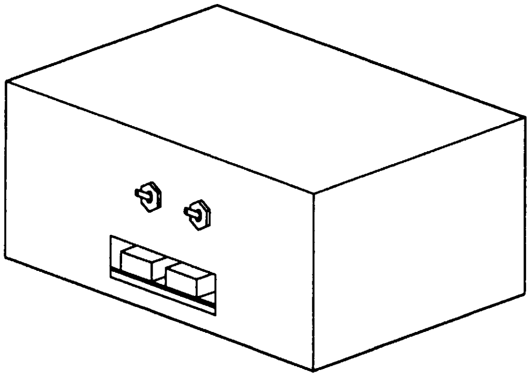
$$\underbrace{50}_{A} + \underbrace{(110 + 160)}_{B} = 320W$$

The calculated value is the required effective power when each axis is used at rated power.

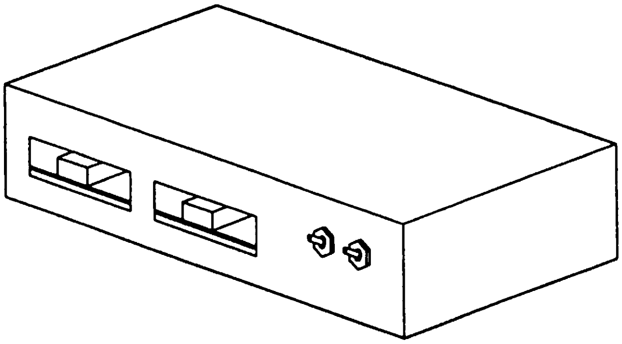
3. Brake Specification (Option)

(1) Summary

When an actuator with brake specification is controlled by the Super SEL Controller, connect the optional brake box to the controller.



AC Brake Box (2 Axis)

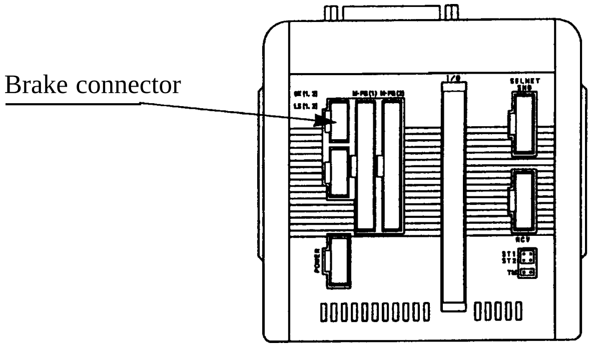


DC Brake Box (2 Axis)

(2) How to connect

① Brake Box and Controller

The brake connector numbers on the controller side are indicated on the cables. Connect the cables according to the numbers. The placement of the brake connectors on the controller differs depending on the actuator. (See below)

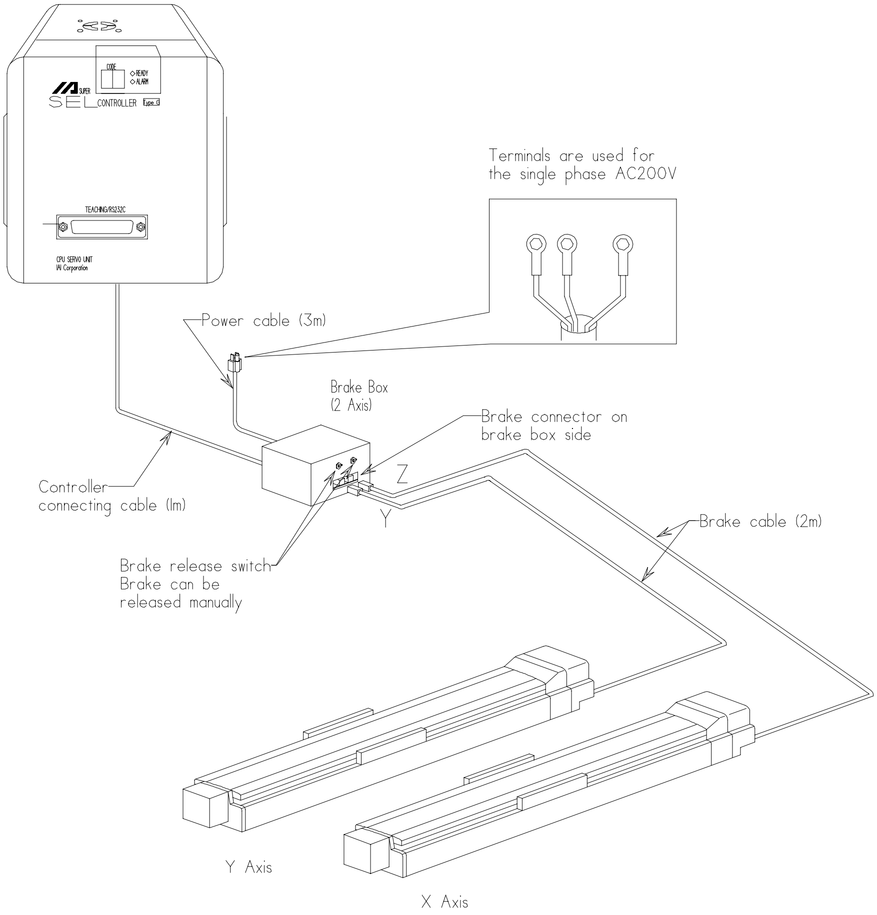


② Brake Box and Actuator

The type of actuator to be connected is indicated on the brake connector of the brake box. Connect brake cables accordingly.

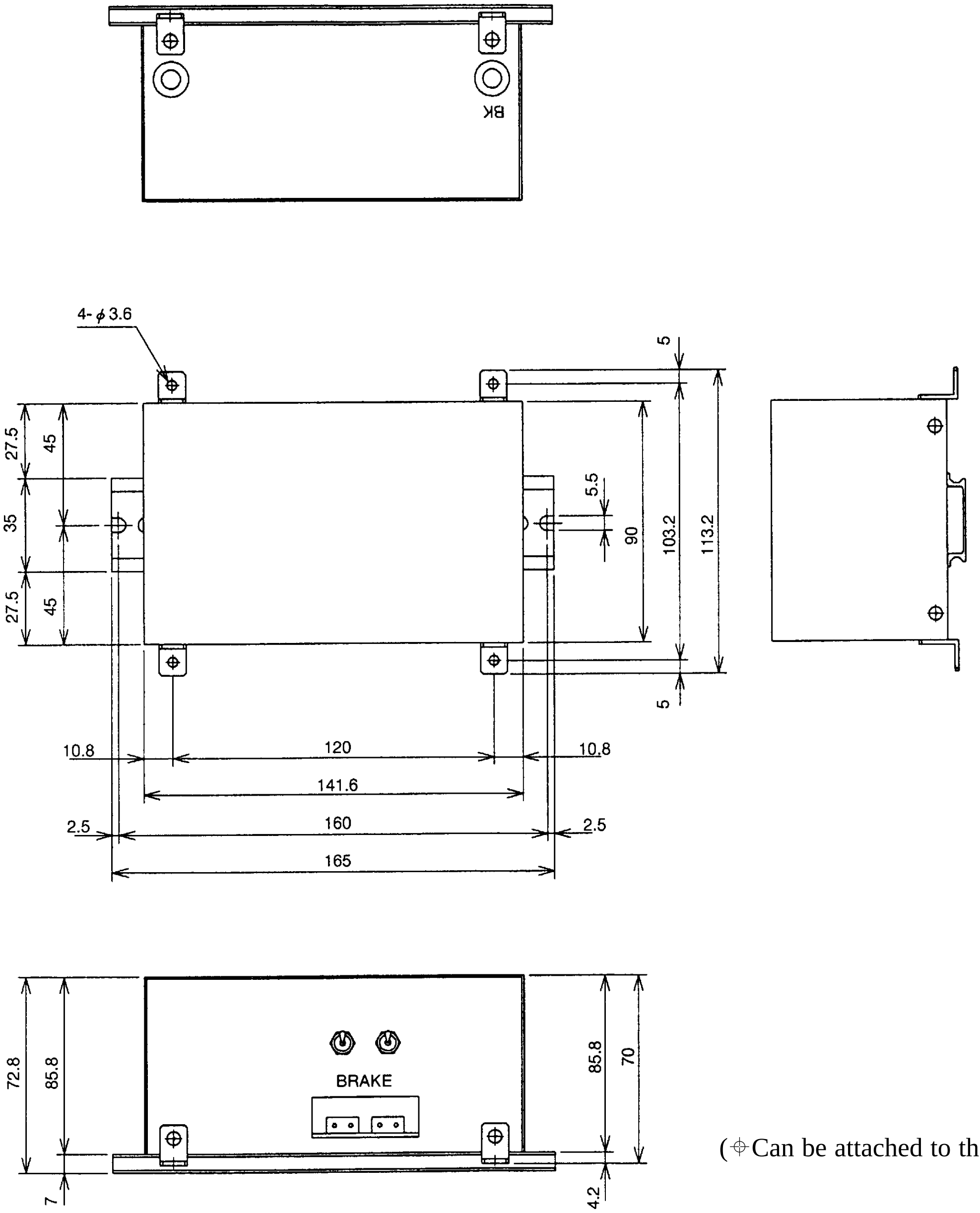
③ Brake 2 axis specification

- Brake Box (2 axis) use



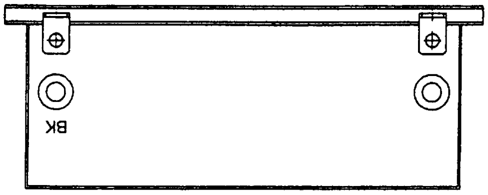
(3) External Dimensions

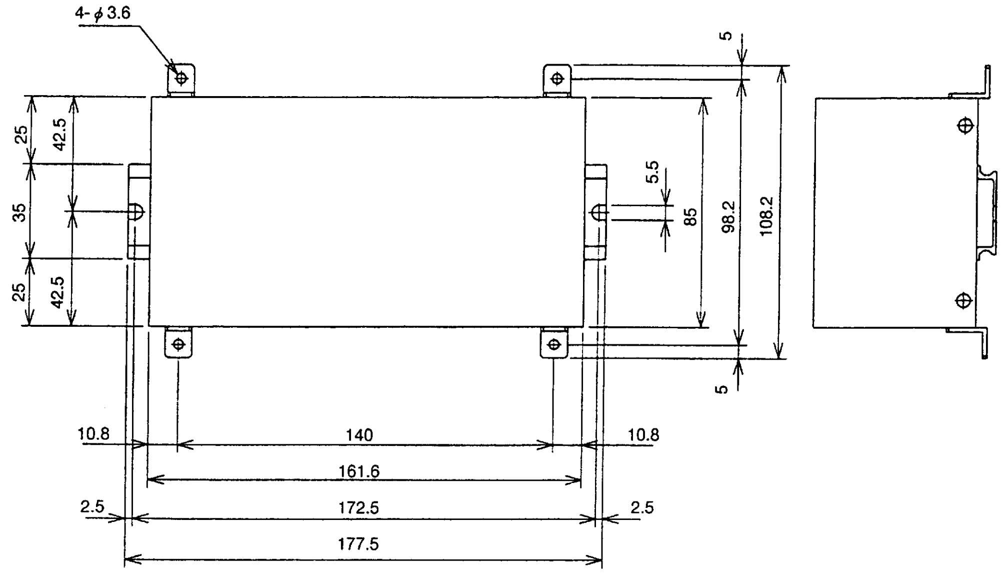
AC Brake Box

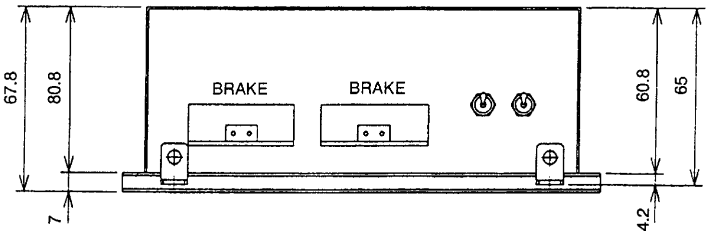


(⌖Can be attached to the DIN rail)

DC Brake Box



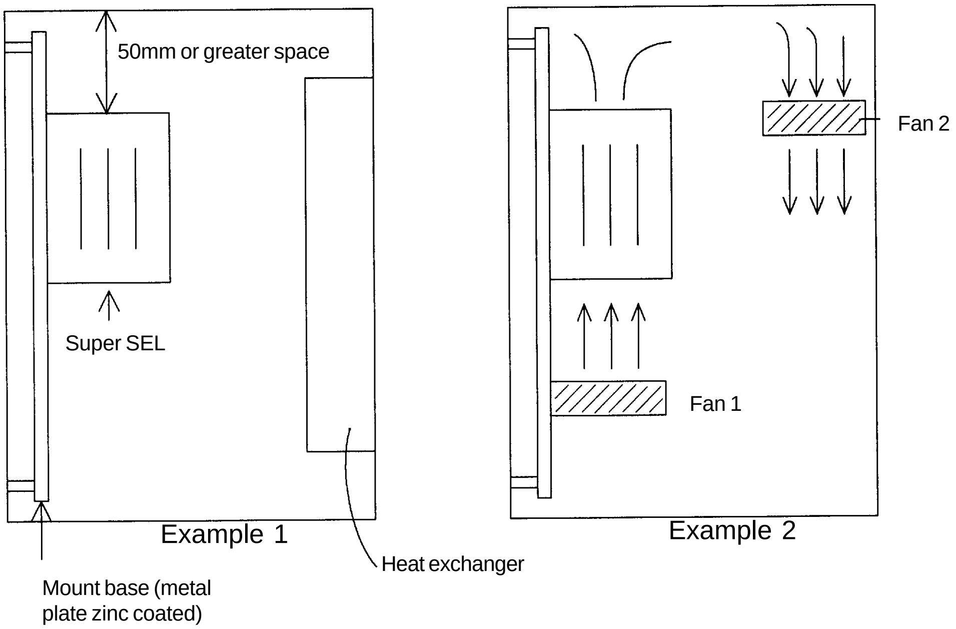




(⌖Can be attached to the DIN rail)

4. Heat Dissipation

① The type E and G Controllers are designed to be mounted inside of a control panel.
The air inside the panel must be cooled without external air exchange (Forced air current method or heat sink method).



② Heat consumption and heat dissipation (for DC Type)

		Heat Consumption	Maximum Heat Dissipation	60% Duty
35W	2 Axis	170W (rated)	30W (total load)	20W
60W	2 Axis	270W (rated)	50W (total load)	30W
100W	2 Axis	370W (rated)	70W (total load)	40W
200W	2 Axis	590W (rated)	100W (total load)	60W

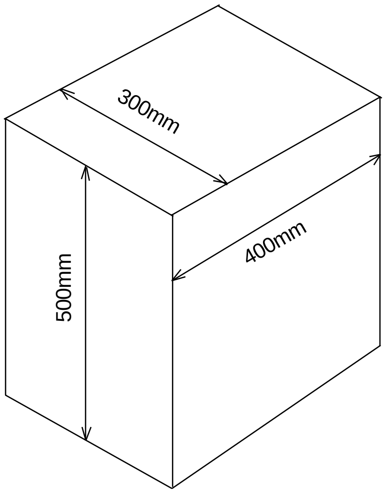
For general use, the load duty is approximately 60%. However, for vertical axes which are constantly affected by the weight of a load, greater heat dissipation must be taken into consideration.

3) The room temperature must be kept under 10°C.

The temperature rises 1°C / 6W / 1m^2 in a metal plate cabinet with a fan provided inside.

This means that the temperature inside of the cabinet rises 1°C when a 6W heating element exists in the cabinet which has a 1m^2 dissipating surface.

[Example]



When a 200W, 2-axis controller is placed in a cabinet as shown in the drawing.
When the attachment surface is the back of the cabinet, the effective dissipating surface is...

$$\underset{0.2}{\underline{0.4 \text{ x } 0.5}} + \underset{0.3}{\underline{0.3 \text{ x } 0.5 \text{ x } 2}} + \underset{0.24}{\underline{0.3 \text{ x } 0.4 \text{ x } 2}} = 0.74 \text{ m}^2$$

In order to keep the temperature rise within 10°C with this dissipating surface area, the heat dissipation must be kept under 50.4W as shown below.

$$60\text{W x } 0.74 \text{ x } 10 = 50.4\text{W}$$

Therefore, 10W must be released for the 200W 2-axis. In this case, a heat exchanger with a minimum capacity will be able to release this small amount of heat.

In reality, the Super SEL Controller is not the only source of heat. Other components inside of the control panel must be taken into consideration.

Note: Using external air for cooling seems to be the easiest method. However, when environmental conditions are not optimal, small amounts of dust, oil, and metal powder may be sucked into the controller, and can result in electrical damage to the controller.

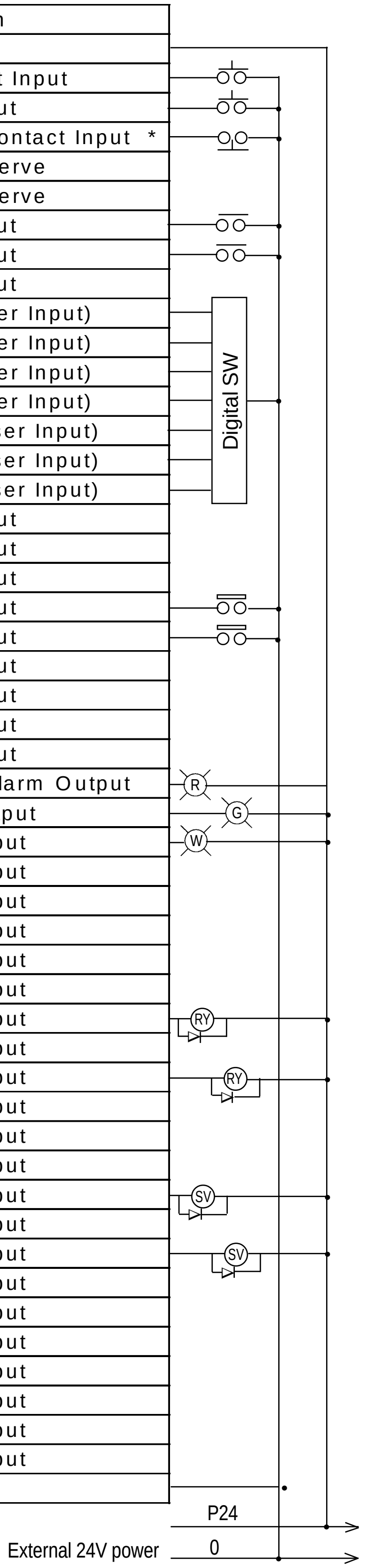
*Supplement

5. I/O DC24V Power Supply

For the Type E/G, there is no DC 24V power supply built in for the I/O. The DC24V power must be supplied externally. Connect +24V to the I/O connector Pin 1A, and 0V to Pin 25B.

A. Standard I/O

Pin No.	Category	Port No.	Function
1A	P24		---
1B	Input	000	External Start Input
2A		001	User Input
2B		002	Emergency Stop b Contact Input *
3A		003	SystemReserve
3B		004	SystemReserve
4A		005	User Input
4B		006	User Input
5A		007	User Input
5B		008	PRG No. 1 (User Input)
6A		009	PRG No. 2 (User Input)
6B		010	PRG No. 4 (User Input)
7A		011	PRG No. 8 (User Input)
7B		012	PRG No. 10 (User Input)
8A		013	PRG No. 20 (User Input)
8B		014	PRG No. 40 (User Input)
9A		015	User Input
9B		016	User Input
10A		017	User Input
10B		018	User Input
11A		019	User Input
11B		030	User Input
12A		021	User Input
12B		022	User Input
13A		023	User Input
13B	Output	300	Emergency Stop/Alarm Output
14A		301	Ready Output
14B		302	User Output
15A		303	User Output
15B		304	User Output
16A		305	User Output
16B		306	User Output
17A		307	User Output
17B		308	User Output
18A		309	User Output
18B		310	User Output
19A		311	User Output
19B		312	User Output
20A		313	User Output
20B		314	User Output
21A		315	User Output
21B		316	User Output
22A		317	User Output
22B		318	User Output
23A		319	User Output
23B		320	User Output
24A		321	User Output
24B		322	User Output
25A		323	User Output
25B	N24		---

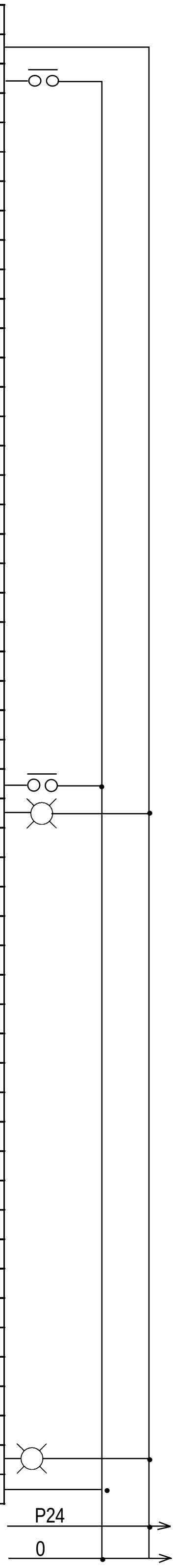


*Supplement

B. I/O Expansion

ForType E/G, there is no DC 24Vpower supply built in for the I/O. The DC24V power must be supplied externally. Connect +24V to the I/O connector Pin 1A, and 0V to Pin 25B.

Pin No.	Category	Port No.	Function
1A	P24		External Power +24V Input
1B	Input	024	User Input
2A		025	User Input
2B		026	User Input
3A		027	User Input
3B		028	User Input
4A		029	User Input
4B		030	User Input
5A		031	User Input
5B		032	User Input
6A		033	User Input
6B		034	User Input
7A		035	User Input
7B		036	User Input
8A		037	User Input
8B		038	User Input
9A		039	User Input
9B		040	User Input
10A		041	User Input
10B		042	User Input
11A		043	User Input
11B		044	User Input
12A		045	User Input
12B		046	User Input
13A		047	User Input
13B	Output	324	User Output
14A		325	User Output
14B		326	User Output
15A		327	User Output
15B		328	User Output
16A		329	User Output
16B		330	User Output
17A		331	User Output
17B		332	User Output
18A		333	User Output
18B		334	User Output
19A		335	User Output
19B		336	User Output
20A		337	User Output
20B		338	User Output
21A		339	User Output
21B		340	User Output
22A		341	User Output
22B		342	User Output
23A		343	User Output
23B		344	User Output
24A		345	User Output
24B		346	User Output
25A		347	User Output
25B			External Power 0V



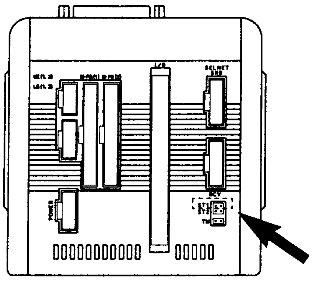
6. Emergency Stop

For the Super SEL Controller Type E·G, the Pin 2B and 0V must be short-circuited (because of the B contact point), otherwise, an Emergency Stop will occur. To release the Emergency Stop for testing first, remove the front cover and short-circuit the ST1 jumper post with a jumper pin at the bottom of CPU UNIT or CPU SERVO UNIT.

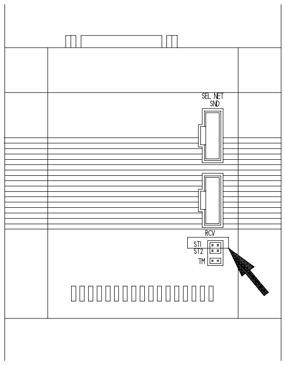
The Emergency Stop operation can be controlled via the teaching pendant while it is being tested.
Note: Please make sure to take off the jumper pin after testing so that the Emergency Stop can operate.

1) A jumper post (ST1) is placed at the bottom of CPU UNIT or CPU SERVO UNIT.

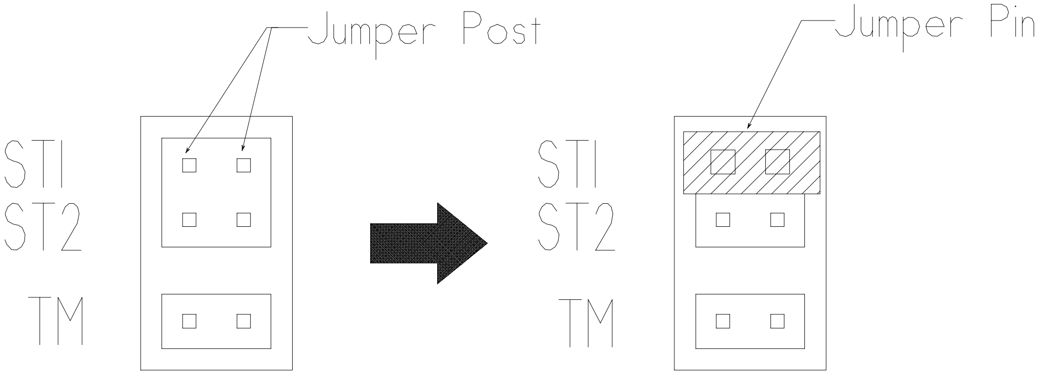
CPU UNIT
Bottom



CPU SERVO UNIT
Bottom



2) To release the Emergency Stop, short-circuit the jumper post (ST1) by inserting a jumper pin. After the test, please make sure to take the jumper pin out again.



7. Error Code List

Error Code	Error Name	Explanation
A1	External Interrupt Error	1. Motor over current 2. Over regenerative current (over negative load) 3. Driver overheat
A2	Motor Overload Error	Mechanical overload of motor
A3	Deviation Error	Motor is unable to perform properly due to mechanical overload
A4	Software Limit Error	Exceeded software limit
A5	Pole Sense Error	Unable to sense pole
B0	No Program Error	Program does not exist
B1	Program Execution Error	Execution of a currently executing program
B2	Program Over Error	Number of tasks exceeds those set as parameters
B3	Double Subroutine Number Error	Two or more of the same subroutine number are used
B4	Double Tag Number Error	Two or more of the same tag number are used
B5	Undefined Subroutine Number	Subroutine number is not defined
B6	Undefined Tag Number	Tag number is not defined
B7	Subroutine Pair Error	BGSR and EDSR are not the same quantity
B8	Step 1 BGSR Error	Step 1 is a BGSR Error
B9	DO, EDDO Pair Error	DO and EDDO are not the same quantity
BA	DO Nest Over Error	DO was used more than 15 times
BB	IF Pair Error	IF and ELSE are not the same quantity
BC	ELSE Error	ELSE was used in a place which was not between IF and EDIF
C0	No Homing Error	Homing was not performed before running actuators
C1	Point Data Error	Attempt has been made to executed unregistered point data
C2	Axis Double Execution Error	Move command given to axis currently moving
C3	Software Limit Error	Software limit exceeded in program
CA	Column Error	Column number was set outside the range of 1 ~ 999
CB	Channel No. Error	Device was set outside the range of 1 ~ 2
CC	Terminator Error	Ending letter was not set
CD	Source No. Error	Source number was set outside the range of 1 ~ 9
CE	S Motion Percent Error	S motion percent ws set outside the range of 0 ~ 50%
CF	Arch Trigger Error	Trigger was set outside the range of 50 ~ 100%
D0	Acceleration Error	Acceleration exceeds limits
D1	No Velocity Error	Velocity has not been set
D2	Override Error	Override was set outside the range of 1 ~ 100%
D3	Angle Error	Angle was set outside the range of 0.1 ~ 120 degrees
D4	Axis Pattern Error	Axis pattern was not set correctly. Displays D4 also for C1 (point data error)
D5	Axis Number Error	Axis number was set outside the range of 1 ~ 8
D6	Axis Error	More than 3 axes are designated in circular/arc motion
D7	Program Number Error	Program number exceeds the limit
D8	Position Number Error	Position number exceeds the limit
D9	Point Number Error	Negative number was input in the point number
DA	Flag Number Error	Flag is not assigned correctly
DB	Variable Error	Variable is not assigned correctly
DC	Digits Over Error	Assigned number exceeds 8 digits (binary 32 bits)
DD	Division (0) Error	Result of the division is "0"
DE	Circular Motion Computation Error	Position data that cannot perform circular motion was input
DF	Task Level Error	Task level was set outside of the range of 1 ~ 5
E0	Undefined Command Error	Attempted to execute undefined command
E1	Subroutine Over Nesting Error	Nesting of more than 15 subroutines
E2	Subroutine Under Nesting Error	EXSR and EDSR are not making a pair
E3	Controlling Column Error	Use of condition is not correct
EG	EMG Error	Emergency (Emergency Stop) was asserted
F0	Interrupt Error	Motor CPU and Interrupt management do not match

8. What to do When an Error Code Occurs

Below we indicate what to do in case any of the error codes described on the preceding page appear in the 7-segment display on the face of the controller.

(1) A1 ~ A5 alarms related to the servo

When one of these alarms related to the axis appears, determining which axis is the cause of the error makes it easier to solve the problem. One of the ways to do this is to judge by the axis status or movement at the time the error was generated. After the error is generated, you can try moving the axis manually if it is a small system. If the axis (when there is no brake) moves without resistance, there is a good possibility that this is the axis causing the problem. When these errors occur, you should ascertain the status of the actuator such as whether it was in the middle of homing.

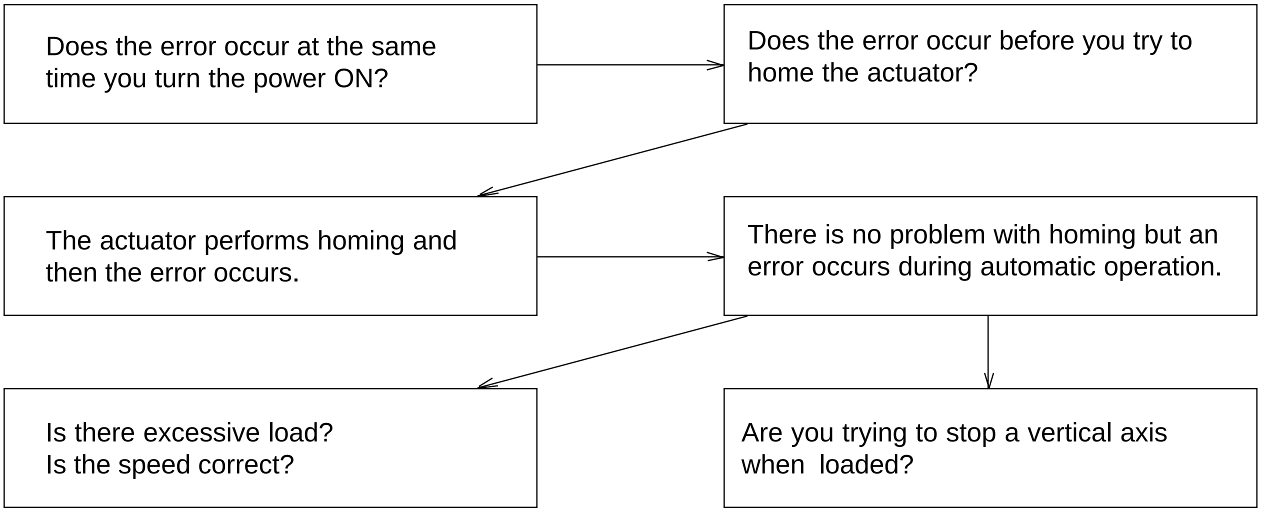
In the case of an A2 alarm where there is excessive load, the cause of the error must be corrected. If you are unsure of the cause, turn on the emergency stop or turn off the power and then after about 10 seconds, turn the power back on to see how the actuator runs. If you cannot find what the trouble is, please contact IAI or one of its agents.

When there is an A3 deviation error, it is possible that something is wrong with the connector cable.

When there is an A4 error, it is almost always caused by a programming error. Recheck the program to make sure that you are not trying to move the actuator beyond the stroke length.

When there is an A5 error, check to see how the axis is moving and then contact IAI. A5 errors can be caused by encoder breakdown, cable problems or driver problems.

Checklist


Does the error occur at the same time you turn the power ON?
Does the error occur before you try to home the actuator?
The actuator performs homing and then the error occurs.
There is no problem with homing but an error occurs during automatic operation.
Is there excessive load?
Is the speed correct?
Are you trying to stop a vertical axis when loaded?

(2) B0 - BC Programming Errors Group 1

An error will be displayed when there is a problem with the written program itself or the program that was started up. In this case, alarm output 300 will not be asserted.

Code	Error	What To Do
B0	No program	The program that was run from external startup has no defined content. Run the program with the proper number.
B1	Program run error	The program that was running restarted. This is not necessarily a problem - the code display is just a warning to the operator.
B2	Program over error	This occurs if you try to run 17 or more programs. Multi-tasking handles only up to 16 programs.
B3	Subroutine no. multiple definitions	A duplicate subroutine number was used. Revise the number.
B4	Tag no. multiple definitions	A duplicate tag number was used. Assign a different number.
B5	Subroutine no. undefined	The subroutine number being called up is undefined. Created the designated subroutine or check the number being input.
B6	Tag no. undefined	The GOTO destination tag is undefined. Check for tag number error or create tag definition.
B7	Subroutine pair error	BGSR is not paired with EDSR. Another BGSR was started before the EDSR was executed which is not allowed.
B8	Step 1 has a BGSR error	Defining a BGSR at the head of a program is not allowed. Define the subroutine at the end of the program.
B9	Too many DO, EDDO nested layers	DO is not paired with EDDO. The number of EDDO is greater or lesser than DO and needs correction. To perform homing after this error occurs, you must turn on the emergency stop once. Currently, this display may appear in the case of a [BB] error.
BA	Too many DO nested layers	This occurs when more than 15 DO nesting layers are set or there are more than a total of 15 nesting layers for the expansion commands. Take note of these nesting layers when using the expansion commands.
BB	IF pair error	IF is not paired with EDIF. The number of EDIF is greater or lesser than IF. Make sure to pair these correctly.
BC	ELSE error	ELSE was used at some place other than between IF and EDIF. Correct the syntax.

(3) C0 - CF programming errors Group 2/Command Error - 1

This group of errors is also related to programming, but primarily arises from the way the commands are used.

C0	Homing incomplete error	Tried to execute move command without performing homing. After the power is turned ON, or after an emergency stop, homing must always be performed.
C1	Position designation error	Tried to move to a position not specified by the position data. Set position data.
C2	Axis-in-motion error	Commanded axis to move again while already in motion. Be careful when doing multi-tasking.
C3	Soft limit error	Commanded axis to move beyond soft limit during the program. Or, the soft limit went into effect when mistakenly changing the parameter settings. Check conditions and make necessary corrections.
CA	Column error	Specified a column number outside the 1 ~ 999 range with respect to communication. Column number should be within 1 ~ 999.
CB	Channel number error	A channel device other than 1 - 2 was specified. Currently, only 1 - 2 can be used.
CC	Terminator error	The terminating character was not specified. Set the terminating character using the SCHA command.
CD	Resource No. error (Reserve error - currently not used)	Resource no. outside of 1 ~ 9 was specified. (Currently, commands that would generate this error are not supported).
CE	S motion percent error	An S motion percent other than 0 ~ 50 was specified. Reset using the range 0 ~ 50.
CF	Arch trigger error	Trigger setting outside of 50 ~ 100 was specified. Reset using the range 50 ~ 100.

(4) D0 - DF programming errors Group 3/Command Error - 2

Like Group 2 above, this group of errors primarily arises from the way the commands are used.

D0	Acceleration error	Commanded axis to accelerate beyond the parameter upper limit. Although acceleration speed can be set at a fairly high value, 0.3G is the basic speed that can be guaranteed. If this error occurs, there is a problem with the speed.
D1	No speed error	There is no speed setting written in this program. It is necessary to specifiy a speed in the program using the VEL command or using the position data.

D2	Override error	The override was specified outside the range of 1 ~ 100%. Specify value within this range.
D3	Angle error	The angle parameter for the circular move command was specified outside the range of 0.1 ~ 120°. Specify angle within this range.
D4	Axis pattern error	The axis pattern designation is incorrect. Or, the problem is the same as for a C1 position setting error. Correct the data setting.
D5	Axis number error	An axis number outside the range 1 ~ 8 was specified or an axis not supported by the controller was specified. Set the correct axis number.
D6	Circular axis designation error	There are data settings for more than 2 axes. The ARC/CIR commands can only be executed in two dimensions. Correct the data setting.
D7	Program number error	Operator attempted to run a program number higher than 64. Only program numbers 1 ~ 64 can be run.
D8	Point number error	A point number higher than 2000 was specified. For point numbers, only the numbers 1 ~ 2000 can be used.
D9	Point data error	Point data was specified as a negative number. Position data must be a positive number although other data can be stored as a negative number.
DA	Flag number error	The flag number assigned was incorrect. Flags can only use numbers 600 ~ 999.
DB	Variable error	The variable number assigned was incorrect. Variables and variables with * indicator must be within 1 ~ 399.
DC	Digit over error	The value input in operand 1, 2 exceeds 8 digits. Or, a value exceeding the range of 32 bits was assigned in the IN command. Assign a value within 8 digits and 32 bits.
DD	Divide by 0 error	Result of the division calculation is 0. To perform a division calculation, the denominator must be a number other than 0. Recheck the algorithm.
DE	Circular move calculation error	Position data was assigned that does not allow a circular move. Set position data that allows circular move.
DF	Task level error (reserve error)	A task level number other than 1 ~ 5 was assigned. (Currently, the command that would cause this error is not supported).

(5) E0 - E3 programming errors Group 4/Command error - 3

These errors, like those in sections 3 and 4 above, primarily arise from the way the commands are used.

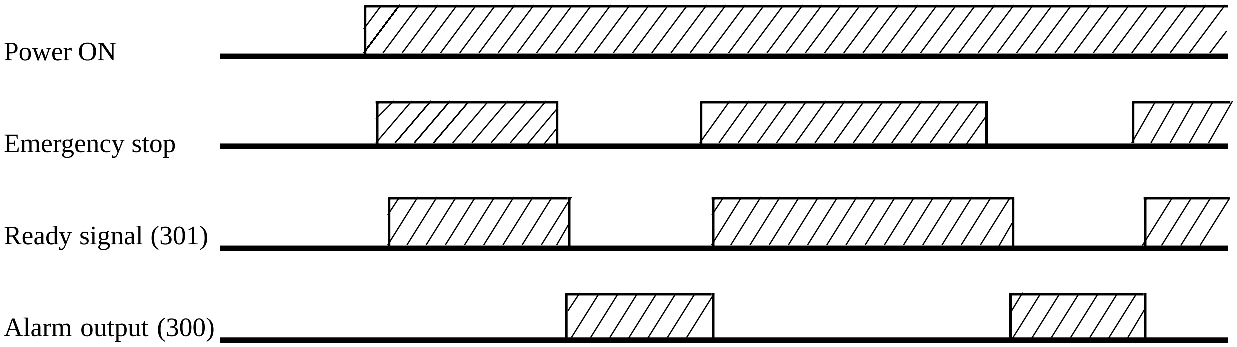
E0	Undefined command error	Attempted to execute an undefined command. If you use the PC interface software, the check function will prevent this.
E1	Subroutine overnesting error	There are more than 15 subroutines nested. This alarm occurs after trying to run the program. Write the program so that the number of nested subroutines is less than 15. Also note that when there is complex use of the IF command, this is likely to cause a [BA] error.
E2	Subroutine under nesting error	BGSR is not paired with EDSR. Another BGSR is found before the EDSR. Correct the syntax.
E3	Controlling column error	The expansion condition is used incorrectly. If you are using the PC software, it will sometimes indicate a syntax error at the time of input.

(6) EG error - Emergency Stop

If an EG alarm occurs, consider the following.

① Emergency stop signal asserted

Determine what triggered the emergency stop and then release the emergency stop (press the button). During an emergency stop, the ready signal or output 301 is OFF and the alarm or output 300 is ON. However, if the emergency stop is asserted when the power is turned ON, alarm output 300 will not go ON. The 300 output functions after the ready signal is given.



② Another thing to consider with an emergency stop

Usually, the emergency stop input is tied to a ground. In the case where you are using an external power supply, the power supply voltage can drop, causing an emergency stop to occur. The way the circuitry is designed, the 24V DC power supply must be turned on before the controller, and the power supply must not be turned OFF while the controller is in operation.

In addition, if there is a malfunction in the controller causing a part of the unit to break down, the EG condition will remain in effect and homing cannot be performed. If this happens, please contact IAI.

7 Other Errors

The following errors occur only rarely under normal operating conditions.

F0	Interrupt error	Motor CPU and the interrupt number do not match. This error could occur when noise interference causes faulty controller operation or there is a breakdown in the hardware. Homing is possible after turning the power OFF and then ON again. If this error occurs several times, contact IAI.
FF	CPU fault error	This indicates a fatal error occurred in the main CPU processing. In this case, the controller will stop working. You must perform homing after turning the power OFF and then ON. This alarm will occur if too many digits were used in a floating point calculation. Make sure that calculations using a real variable will produce a value within ±3.4x

Precautions when handling errors/alarms

When you need to turn the power OFF and then ON again, please make sure to wait approximately 15 seconds after turning the controller power OFF before turning it back ON.
If an error occurs in which you cannot perform homing, please contact a service representative after you have checked out the condition at the time the error occurred as thoroughly as possible. In some cases, the problem might be in the program itself and the representative may request a program list from you.

Index

Index

Index

Intelligent Actuator Inc.

2690 West 237th Street
Torrance, CA 90505
310-891-6015 / 310-891-0815 (Fax)

Publication No.	IAI-038C.v 7
Publication Date:	March1998
Price: US$20.00	