User Guide
Version:
V2.0

AUCTECH

PLC Programming Manual of M6 Series CNC controller

Applicable Product
M6200i Five-axis CNC controller

■ **Please read this Manual carefully before use**

This Manual provides a comprehensive introduction to the features, composition, functional commands of each part and their use, operating procedures, user programming methods and examples of the AUCTECH M618i/M620i/M6200i series CNC controller PLC, and it is a basic manual for users to quickly learn and use this PLC controller. The updating and upgrading of this Manual is authorized and organized by Guangzhou Auctech Automation Technology Ltd. Without the authorization or written permission of the Company, no unit or individual has the right to modify or correct the contents of this Manual, and the Company is not responsible for the customer losses caused thereby.

In this Manual, we have endeavored to describe the various events that are relevant to the operation of the controller. For reasons of space and product development orientation, it is not possible or feasible to describe in detail all events that are not necessary or not possible in the controller. Therefore, events not specifically described in this Manual are considered "impossible" or "not allowed".

* **All controller specifications and designs are subject to change without notice**

1 Introduction to sequential programs

1.1 Execution process of sequential programs

In a normal relay control circuit, the relays can be fully synchronized in time. In the example shown below, relays D and E can be activated at the same time as relay A (when contacts B and C are both closed). In PLC sequential control, each relay operates in turn. When relay A operates, relay D operates first and then relay E (as shown in Figure 1-1), i.e. the relays operate in the order shown in the ladder diagram.

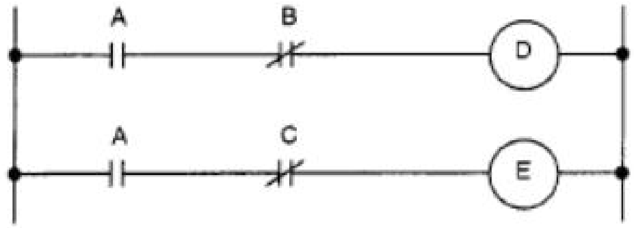


Figure1-1

Figures 1-2 and 1-3 show the difference between the actions of a relay circuit and a PLC program. 1) For a relay circuit, the actions in both figures are the same. When A (push-button switch) is switched on, current passes through coils B and C, and B and C are switched on. C is switched on and B is disconnected. 2) For the PC sequential program, Figure 1-2 is the same as the relay circuit. After A (push-button switch) is switched on, B and C are switched on, and after a cycle of PLC program, B is disconnected; However, in Figure 1-3, after switching on A (push-button switch), C is switched on, but B is not switched on.

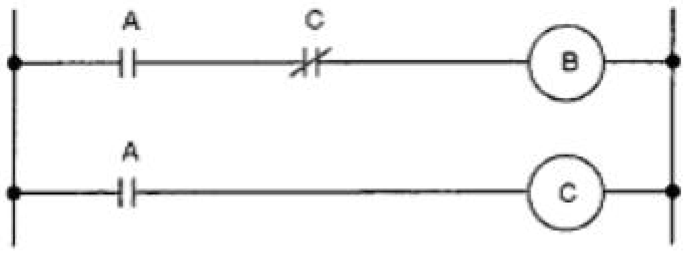


Figure1-2

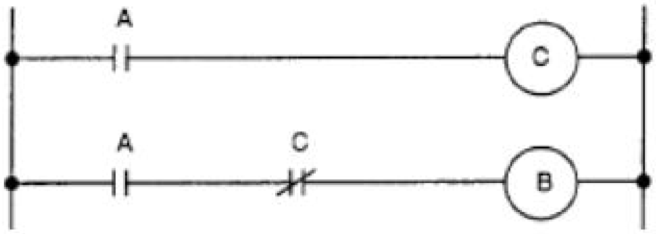


Figure1-3

1.2 Loop execution

Sequential programs are executed from the beginning of the ladder diagram until the end of the ladder diagram. After the program has been executed, it is executed again from the beginning of the ladder diagram, which is called cyclic execution. The execution time from the beginning to the end of the ladder diagram is called the cycle processing period and it depends on the size of the control (number of steps) and the size of the first level program. The shorter the processing cycle, the more responsive the signal will be (the faster the speeder).

1.3 Components of a sequential program

In a ladder diagram language that allows structured programming, the following advantages are offered: 1) good program readability, easy to understand, easy to prepare and maintain; 2) easy debugging of program errors, and in the event of errors, it is easy to find out reasons for the errors.

The three main structured programming methods are described as follows:

1. Subprogram The subprogram uses the ladder as the processing unit. This is shown in Figure 1-4.

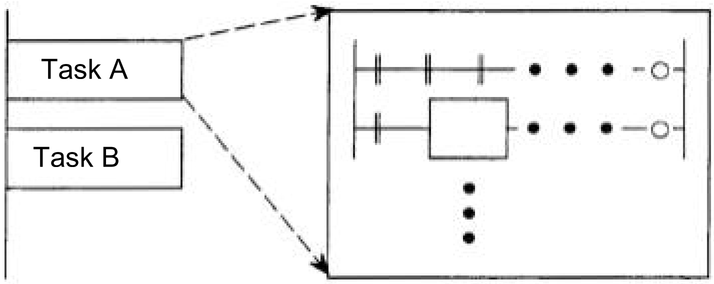


Figure 1-4 Subprogram structure

2. Nesting, i.e. a group of subprogram lines are nested into a structured program. This is illustrated in Figure 1-5.

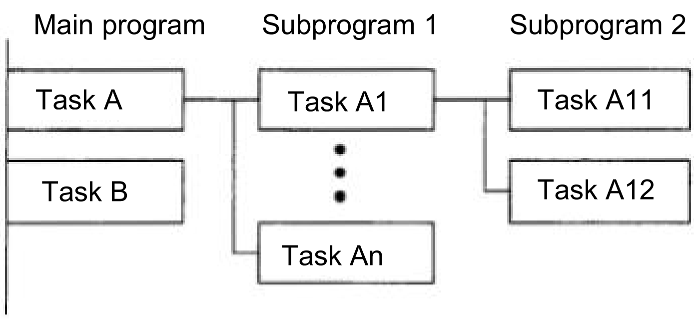


Figure 1-5 Nested structure

3. The conditional branch based on the given jump condition, jumps to the execution of the specified module when the condition is satisfied. If the condition is not satisfied, the specified module is not executed. The specified module can be either a subprogram or a collection of commands specified by a jump command. This is shown in Figure 1-6.

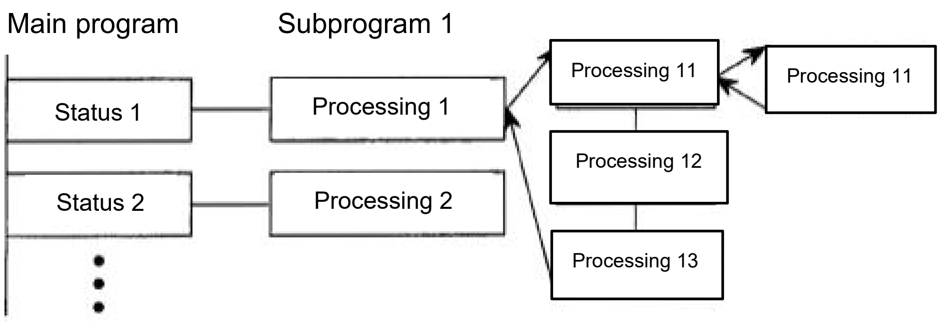


Figure 1-6 Branch structure

2 Ladder language programming

2.1 Ladder program structure and execution process

The ladder program is divided into three parts: the Initialization, the first-level program and the second-level program. The Initialization is executed only once at power-up, the first-level program is executed once per scan cycle, and the second-level program is executed in blocks, one block per scan cycle. The second-level program is therefore executed in cycles of:

Second-level program execution cycle = PLC scan cycle x number of second-level program chunks

If the first-level program is too long, it will directly affect the efficiency of the ladder execution. Therefore, when programming the first-level program, it is important to keep it as short as possible, and usually only programs with high real-time requirements are programmed in the first level.

The hierarchical structure of the ladder diagram is shown in Figure 2-1.

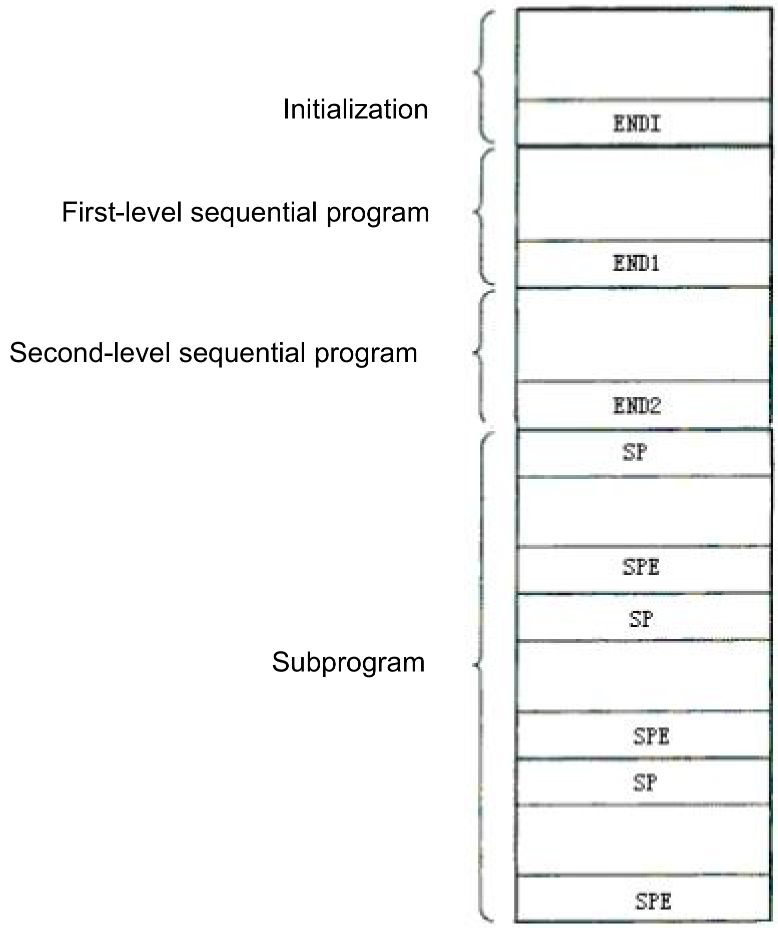


Figure 2-1 Hierarchical structure of the ladder program

2.2 Ladder signal classification and address format

The signal sources for ladder processing consist of five components: machine tool side I/O signals, CNC controller side signals, control panel key signals, internal registers and non-volatile memory (counters, timers and data tables). This is shown in Figure 2-2.

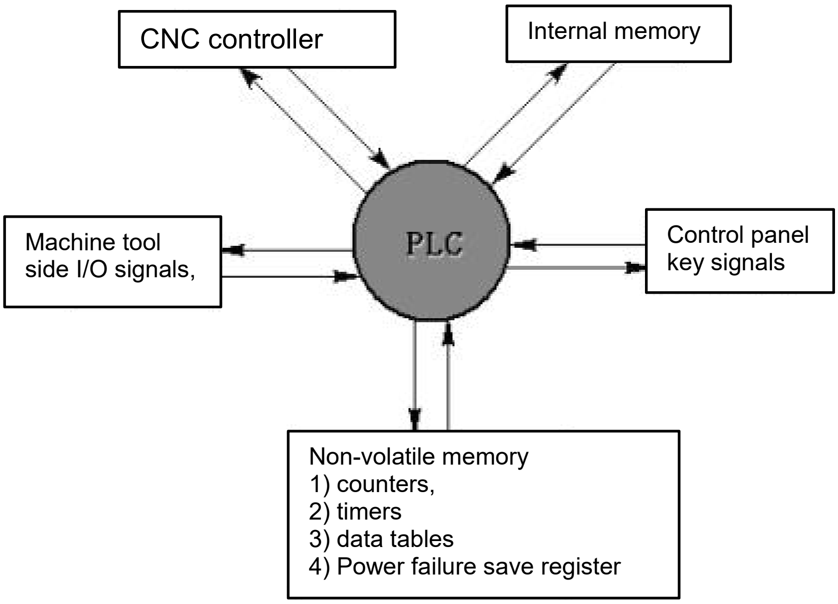


Figure 2-2 Classification of ladder signals

Each signal processed in a ladder diagram has a unique address. The address is made up of two parts: the type code and the address number. The type code is used to distinguish between the different signal types (i.e. the signal types shown in Figure 2-2), the address number is used to determine the location of the signal.

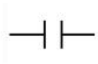
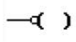
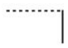
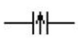
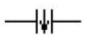
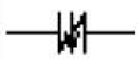
The address representation of each signal is shown in the table below.

Signal type		**Type Code**	**Address range**	**Description specify**	**Operation al properties**
Machine tool side I/O Signal	Input	X	0.x to 9.x	I/O input from machine tool side to PLC	Read only
	Output	Y	0.x to 9.x	Output from PLC to machine tool I/O	Read/write
CNC controller Side Letter No.	Input	F	0.x to 739.x	Input from CNC controller to PLC	Read only
	Output	G	0.x to 739.x	Output from PLC to CNC controller	Read/write
Control panel Keystrokes	Input	K	0.x to 29.x	Input from the control panel to the PLC	Read only
	Output	L	0.x to 29.x	Output from PLC to control panel	Read/write
Internal Hosting utensil	Input/o utput	R	0.x to 99.x	Input from internal registers to PLC Output from PLC to internal registers	Read/write
Power failure save Register	Input/o utput	B	0.x~19.x	Input from B register to PLC Output from PLC to B register	Read/write
Alarm Hosting utensil	Input/o utput	A	0.x~69.x	Input from data table to PLC Output from PLC to data table	Read/write

Note (1): Read-only signals cannot be output as coils. **Note (2)**: The address range indicated in the table above is the maximum range defined by the CNC controller, where the actual address range of X and Y depends on the number of IO points in the controller

2.3 Basic command programming

The basic ladder commands of the PLC are as follows.

	Normally open contacts, reads the status of the normally open contacts of the I/O points.
	Normally closed contact, reads the status of the normally closed contact of the I/O point.
—( )	Coil output.
	Take the inverse output. If the previous calculation results in 1, then output 0; if the calcula tion resu Its in 0, t hen output 1.
———	Horizontal lead-in line.
	Vertical lead-in line.
	Rising edge of normally open contact.
	Normally open contact falling edge.
	Rising edge of normally closed contact.
	Normally closed contact falling edge.
—(1)	Displacement output. If the previous calculation results in 1, the output is 1; if the calculation results in 0, the output remains unchanged.
—(0)	Reset output. If the previous calculation results in 1, output 0; if the calculation results in 0, keep the output unchanged.

2.4 Functional command programming

2.4. A: Process control commands

2.4. A.1 ENDI (End of initialization)

2.4. A.1.1 Function

End of initialization program flag. In PLC sequential programs, the Initialization is executed only once at power-up. The ENDI command is placed at the end of the Initialization.

2.4. A.1.2 Command Format

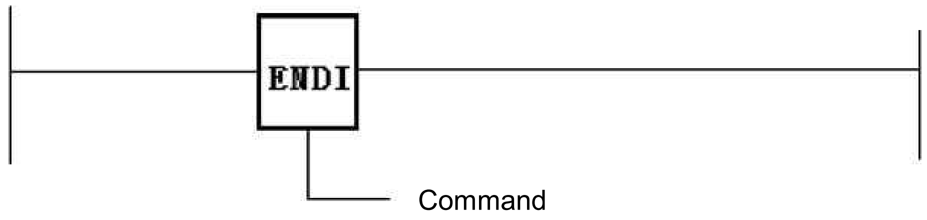


Figure 2-3 ENDI command ladder format

2.4. A.1.3 Control conditions

The ENDI command has no control conditions.

2.4. A.1.4 Command parameters

The ENDI command has no command parameters.

2.4. A.1.5 Result output

The ENDI command has no result output.

2.4. A.2 END1 (end of first level sequential program)

2.4. A.2.1 Function

The first level program end flag. The END1 command must be given once, and only once, in a PLC sequential program. When there is a first-level program, the END1 command is placed at the end of the first-level program; if there is no first-level program, it is placed at the beginning of the second-level program.

2.4. A.2.2 Command Format

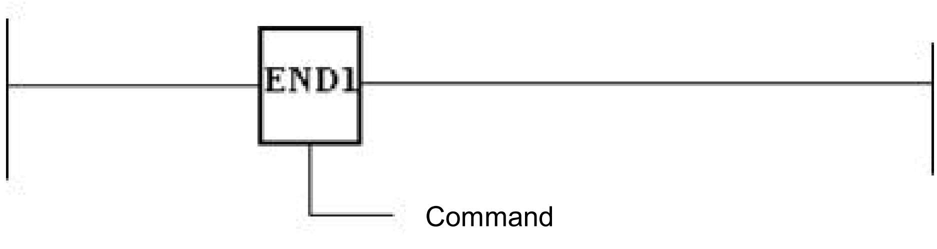


Figure 2-4 END1 command ladder format

2.4. A.2.3 Control conditions

The END1 command has no control conditions.

2.4. A.2.4 Command parameters

The ENDI command has no command parameters.

2.4. A.2.5 Result output

The END1 command has no result output.

2.4. A.3 END2 (end of second level sequential program)

2.4. A.3.1 Function

The second level program end flag. The END2 command must be given once, and only once, in a PLC sequential program. The second- level program should follow the first-level program (or the END1 command if there is no first-level program).

2.4. A.3.2 Command Format

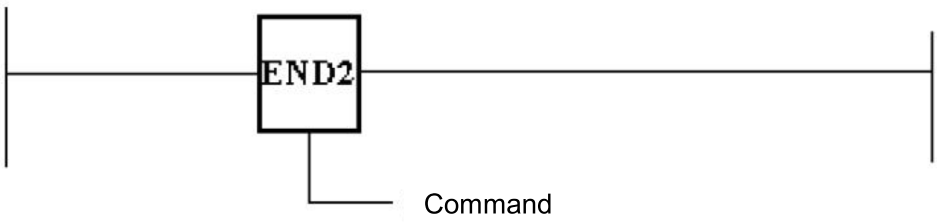


Figure 2-5 END2 command ladder format

2.4. A.3.3 Control conditions

The END2 command has no control conditions.

2.4. A.3.4 Command parameters

The END2 command has no command parameters.

2.4. A.3.5 Result output

The END2 command has no result output.

2.4. A.4 CALL (conditional subprogram call)

2.4. A.4.1 Function

The CALL command is used to call a subprogram. When the call conditions are met, the execution of the CALL command jumps the execution address of the PLC program and moves to the execution of the subprogram. After the execution of the subprogram is complete, then return to the execution jump to continue down the execution.

2.4. A.4.2 Command Format

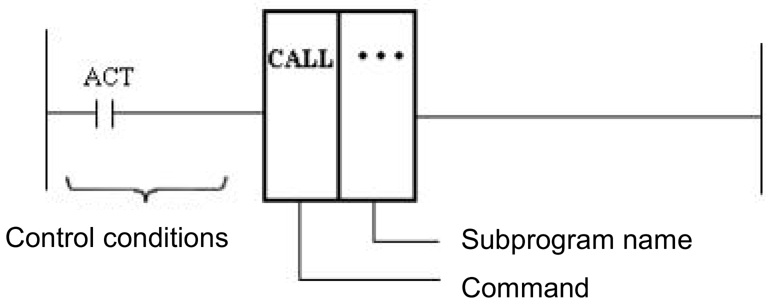


Figure 2-6 CALL command ladder format

2.4. A.4.3 Control conditions

ACT=0: No CALL command is executed, the PLC program is executed sequentially.

ACT=1: The CALL command is executed and the PLC program is transferred to a subprogram for execution.

2.4. A.4.4 Command parameters

Subprogram name: The program name of the subprogram, which identifies the target address of the PLC program jump. The program name is given by ASCII characters or numbers up to a maximum of five characters.

2.4. A.4.5 Result output

The CALL command has no result output.

2.4. A.5 SP (subprogram)

2.4. A.5.1 Function

The SP command is used to generate a subprogram that ends with an SPE command (i.e. the area between a pair of SPs and SPEs is the valid range of the subprogram). SPs should be used in pairs with SPE commands.

2.4. A.5.2 Command Format

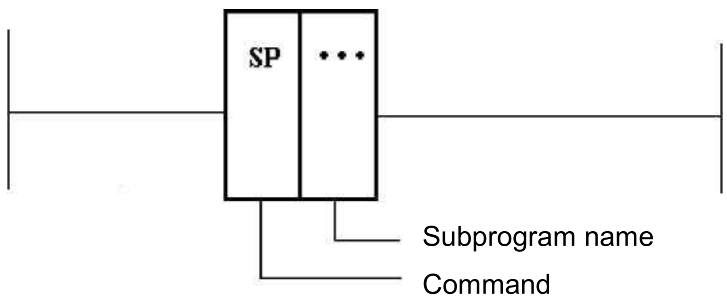


Figure 2-7 SP command ladder format

2.4. A.5.3 Control conditions

The SP command has no control conditions

2.4. A.5.4 Command parameters

Subprogram name: identifies the starting address of the subprogram. The program name consists of an ASCII character and a number and should be unique and not repeated in the PLC program.

2.4. A.5.5 Result output

The SP command has no result output.

2.4. A.6 SPE (end of subprogram)

2.4. A.6.1 Function

The SPE command is used to generate a subprogram. After the SPE command has been executed, the PLC program returns from the subprogram to the subprogram call point for further execution.

2.4. A.6.2 Command Format

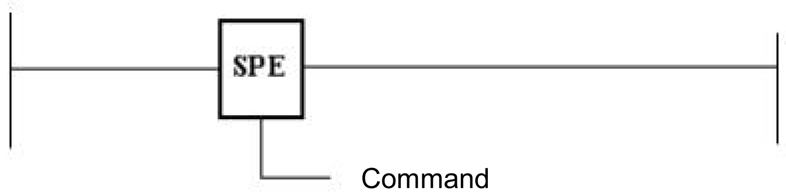


Figure 2-8 SPE command ladder format

2.4. A.6.3 Control conditions

The SPE command has no control conditions.

2.4. A.6.4 Command parameters

The SPE command has no command parameters.

2.4. A.6.5 Result output

The SPE command has no result output.

2.4. A.7 RETN (conditional subprogram returns)

2.4. A.7.1 Function

The RETN command is used in a subprogram to exit from the subprogram and return to the sub-program's call point when the control condition is met. The RETN command acts in the same way as the SPE command when the control condition is met.

2.4. A.7.2 Command Format

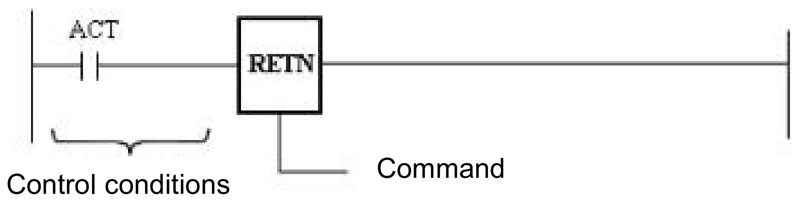


Figure 2-9 RETN command ladder format

2.4. A.7.3 Control conditions

ACT=0: No RETN command is executed, the subprograms are executed sequentially until the end of SPE.

ACT=1: The RETN command is executed and the subprogram ends and returns to the call point. At this time, RETN and SPE commands have the same effect.

2.4. A.7.4 Command parameters

The RETN command has no command parameters.

2.4. A.7.5 Result output

The RETN command has no result output.

2.4. A.8 JUMP (conditional jump)

2.4. A.8.1 Function

The JUMP command causes the PLC to shift the location of the commands it executes (jump to the specified location for execution, rather than executing them sequentially.)

2.4. A.8.2 Command Format

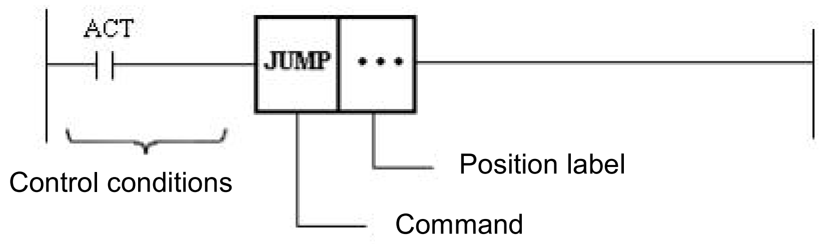


Figure 2-10 JUMP command ladder format

2.4. A.8.3 Control conditions

ACT=0: No JUMP command is executed.

ACT=1: Execute the JUMP command.

2.4. A.8.4 Command parameters

Position marker: This specifies the target position for the PLC to execute the transfer when the JUMP command is executed, which is given by the LABL command.

2.4. A.8.5 Result output

No result is output for this command.

2.4. A.9 LABL (location labeling)

2.4. A.9.1 Function

The LABL command is used to mark a location in a ladder program. This command is used in conjunction with the JUMP command to provide a target location for the JUMP command to be transferred to.

2.4. A.9.2 Command Format

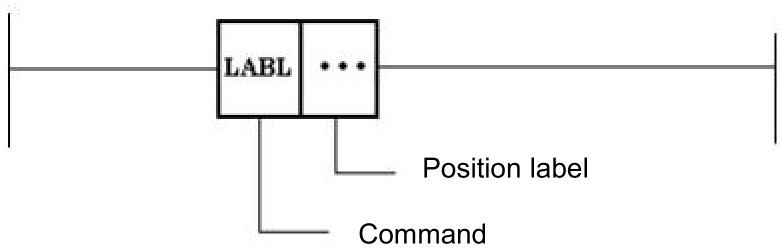


Figure 2-11 LABL command ladder format

2.4. A.9.3 Control conditions

There are no control conditions for this command.

2.4. A.9.4 Command parameters

Location labeling: used to mark the position of the ladder program.

2.4. B: Counter commands

2.4. B.1 CTR (counter)

2.4. B.1.1 Function

CTR is used as a counter and its functions include

1) Ring counter

When the count value reaches the preset value, CTR outputs a high level and the count value returns to its initial value.

2) Add/subtraction counter

CTR can be used as an addition counter or a subtraction counter.

3) Selection of initial values

The initial value can be selected as 0 or 1.

2.4. B.1.2 Command Format

This controller has 20 counters in the PLC with counter numbers 1 to 20.

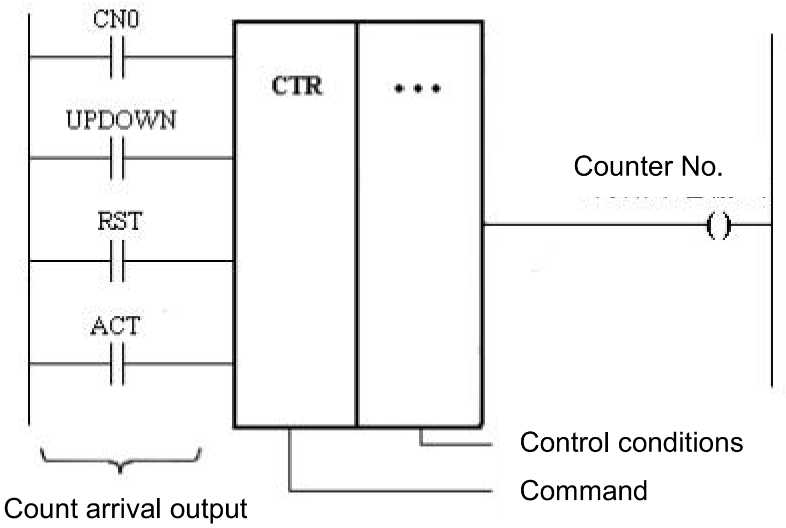


Figure 2-12 Ladder diagram format for CTR

2.4. B.1.3 Control conditions

1) Specify initial value (CN0)

   CN0=0: count value starts at 0, i.e. 0,1,2,3,4,5,

   CN0=1: count value starts at 1 (0 is not used), i.e. 1,2,3,4,5,

2) Specify the rising or falling counter (UPDOWN)

   UPDOWN=0: Subtracting counter. The counter is to subtract from the preset value, reaches the value specified by CN0, and then returns to the preset value to continue subtracting the count.

   UPDOWN=1: Add counter. The counter starts to add from the value specified by CN0, reaches the preset value and then returns to the value specified by CN0 to continue adding

3) Reset (RST)

   RST=0. Reset is disabled. At this point CTR is in the normal count state.

   RST=1: Reset is active. The counter outputs low level and the count value is reset to the initial value. The initial value is determined jointly by CN0 and UPDOWN.

4) Counting signal (ACT)

   The rising edge of the count signal is active, i.e. the ACT counts once when it jumps from low to high. This is shown in Figure 2-3

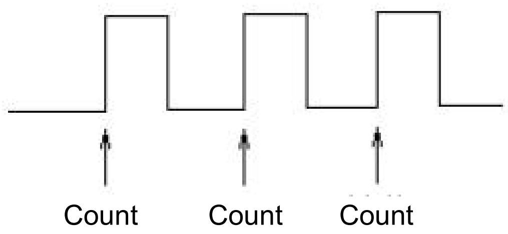


Figure 2-13 CTR count signal

2.4. B.1.4 Command parameters

Counter number: This is used to specify the ID number of the counter being used. There are 20 counters in the controller PLC and the counter numbers range from 1 to 20.

Note

It shall ensure that the counter number is not reused during PLC programming. If a repeated counter is used in the ladder diagram, the correctness and safety of its execution cannot be guaranteed.

2.4.B.1.5 Result output

RST=1, CTR is in the reset state, and the output is always kept low.

RST =0, CTR is in the counting state. At this time, the output can be divided into two cases: 1) if the counter is adding (UPDOWN=1), a high level is output when the count value=the preset value, otherwise a low level is output. 2) If the counter is subtracting (UPDOWN=0), a high level is output when the count value reaches the minimum value, otherwise a low level is output (the minimum value is specified as 0 or 1 by CN0).

2.4. C: Timer command

2.4. C.1 TMR (timer)

2.4. C.1.1 Function

Timer with delayed on-state timer. When the timer is off, there is no timed output function and the output is always low; when the timer is started, the output goes high when the set time is reached, enabling the subsequent timing relay to be on (timing relay is given by the designer).

The TMR (Timer) acts in the same way as the TMRC (Timer,) the difference being that the timing times for the two timers are specified differently. the timing time for the TMRC (Timer) is specified by the timer itself; the timing time for the TMR (Timer) is specified by the value stored in the data table corresponding to the timer number of the TMR (Timer).

2.4. C.1.2 Command Format

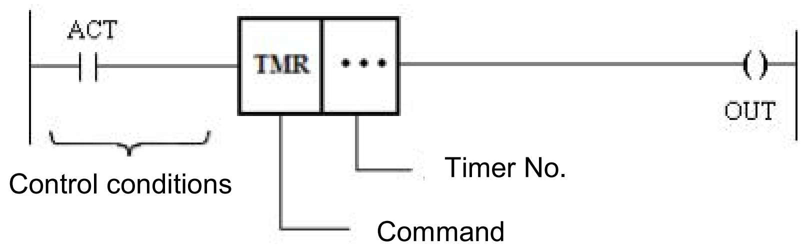


Figure 2-14 TMR command ladder format

2.4. C.1.3 Control conditions

ACT=0: Turn off the timer. At this point the timer has no delayed output function and the output remains low.

ACT=1: Start the timer. When the set time is reached, the output goes high to turn the timer relay on.

2.4. C.1.4 Command parameters

Timer number: This is used to specify the ID number of the TMR (timer) being used. There are 50 TMR timers in the PLC, with TMR counter numbers from 1 to 50. The corresponding data sheet addresses are T0, T2, T4, T6, T8...T98 (even serial numbers) and the data table specifies the time in milliseconds for the corresponding TMR (timer). Example: TMR (timer) No. 2 is timed by T2.

2.4. C.1.5 Result output

When ACT=0, the timer is off and the output is always low.

When ACT jumps from low to high (rising edge triggered,) the timer starts and the output goes low until the timing time is reached, after which the output remains high until another rising edge of ACT restarts the timer.

2.4. C.2 TMRC (timer)

2.4. C.2.1 Function

Delay to turn on the timer. When the timer is off, there is no timed output function and the output is always low; when the timer is started, the output goes high when the set time is reached, enabling the subsequent timing relay to be on (timing relay is given by the designer.)

2.4. C.2.2 Command Format

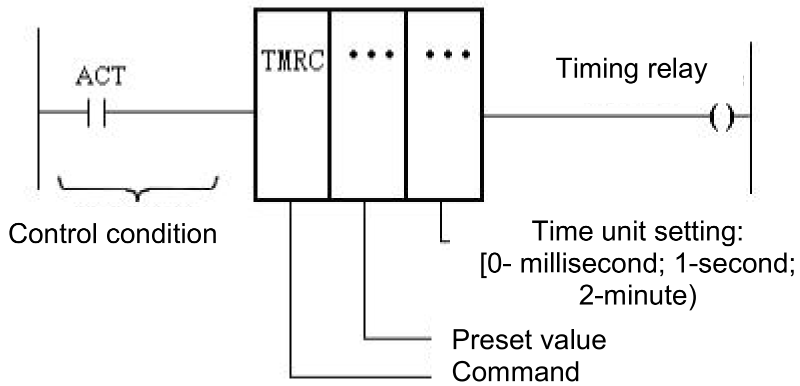


Figure 2-15 TMRC control format

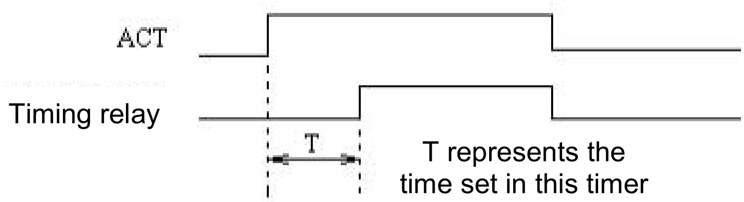


Figure 2-16 TMRC Control Timing Diagram

2.4. C.2.3 Control conditions

ACT=0: Turn off the timer. At this point the timer has no delayed output function and the output remains low.

ACT=1: Start the timer. When the set time is reached, the output goes high and the timer relay is turned on. See TMRC control timing diagram.

2.4. C.2.4 Command parameters

Preset value: Used to set the delay time for the timer action. Either a constant or a data table address can be entered. If a constant is entered, the value entered is the delay time; if a data table address is entered, the value corresponding to the data table address is the delay time.

Unit setting: used to set the unit for the delay time, 0-millisecond, 1-second, 2-minute.

2.4. C.2.5 Result output

When ACT=0, the timer is off and the output is always low.

When ACT jumps from low to high (rising edge triggered,) the timer starts and the output goes low until the timing time is reached, after which the output remains high until another rising edge of ACT restarts the timer (see TMRC control timing diagram).

2.4. D: Register command

2.4. D.1 WRTRB(write register byte), WRTRW(write register word) , WRTRDW (write register double words)

2.4. D.1.1 Function

The WRTRB / WRTRW / WRTRDW commands are used to write data to the specified data table address cell.

2.4. D.1.2 Command Format

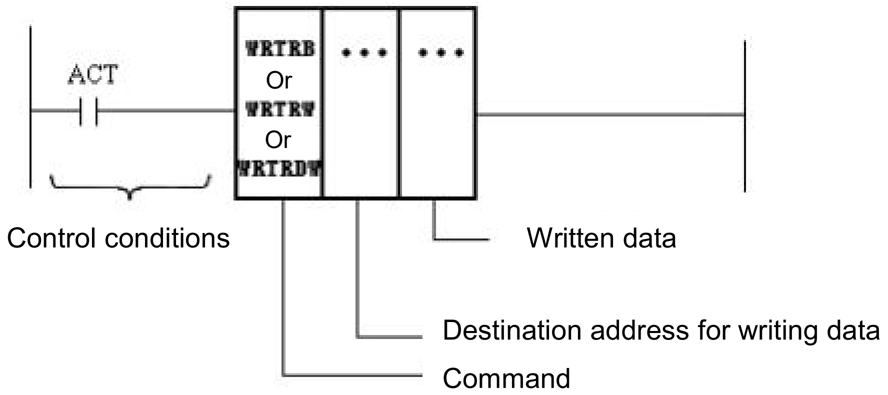


Figure 2-17 WRTRB / WRTRW / WRTRDW command format

2.4. D.1.3 Control conditions

ACT=0: WRTRB / WRTRW / WRTRDW commands are not executed.
ACT=1: Execute WRTRB / WRTRW / WRTRDW commands.

2.4. D.1.4 Command parameters

1) Write data destination address
This parameter specifies the address unit (in the data table)to which the data is written. The
WRTRB command writes to the target address of a register byte; the;
WRTRW command writes to the target address of a register word; the;
WRTRDW command writes to the target address of a register double word;
2) Data written in
This parameter specifies the data (constant or data address)to be written into the address unit.

2.4. D.1.5 Result output

No result is output for this command.

2.4. D.2 WRTRS (write register bit segment)

2.4. D.2.1 Function

The WRTRS command is used to write data to the specified bit-segment data table address cell.

2.4. D.2.2 Command Format

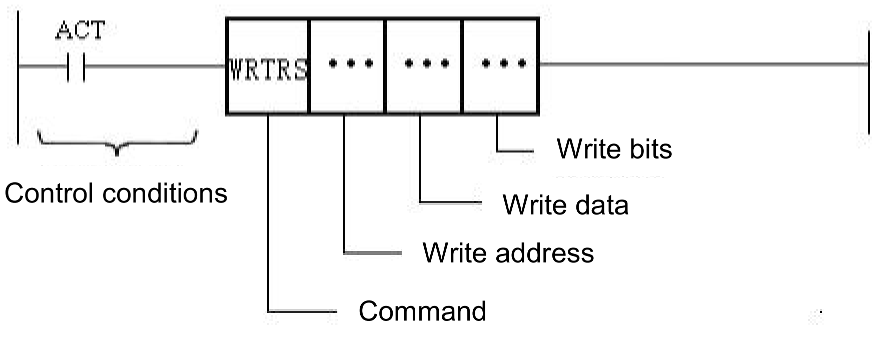


Figure 2-18 WRTRS command format

2.4. D.2.3 Control conditions

ACT=0: The WRTRS command is not executed.
ACT=1: Execute the WRTRS command.

2.4. D.2.4 Command parameters

1) Write to address

   This parameter specifies the data table address cell to which the register bits are written.

2) Write to data

   This parameter specifies the data (constant or data table address)tobe written to the address cell of the register bit segment.

3) Number of bits written in

   This parameter specifies the number of register bits to be written to the data register bit segment.

2.4. D.2.5 Result output

No result is output for this command.

2.4. D.3 READV (read variable value)

2.4. D.3.1 Function

The READV command is used to read the value of a variable in the address cell of the specified data table.

2.4. D.3.2 Command Format

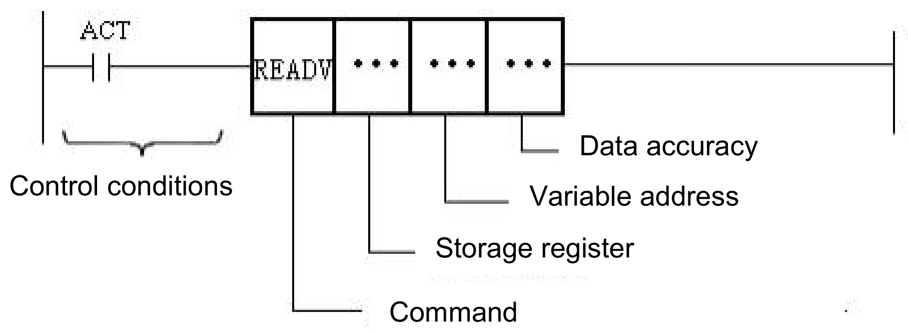


Figure 2-19 READV command format

2.4. D.3.3 Control conditions

ACT=0: The READV command is not executed.
ACT=1: Execute the READV command.

2.4. D.3.4 Command parameters

1) Storage Registers

   This parameter specifies the address unit (in the data table) to which the variable data is written.

2) Variable address

   This parameter specifies the address of the variable (a three- or four-digit variable address starting with V).

3) Data accuracy

   This parameter specifies the precision of the data

2.4. D.3.5 Result output

No result is output for this command.

2.4. D.4 DIFU (rising edge detection)

2.4. D.4.1 Function

The DIFU command detects the state of the input signal and when the rising edge of the signal occurs, the DIFU command outputs a pulse signal of the width of

1) If the DIFU command is in the first level of the program, its width is the scan period of the PLC.
2) If the DIFU command is in the second level program, its width is

Pulse width = PLC scan period x number of program chunks in second level

2.4. D.4.2 Command Format

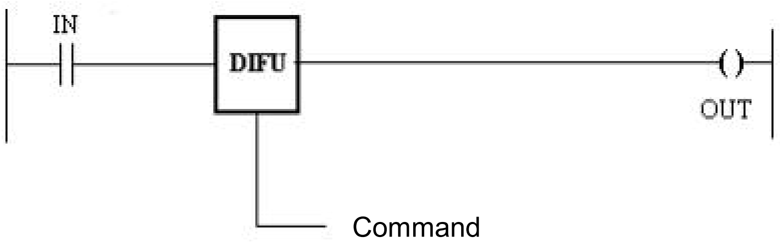


Figure 2-4 DIFU command format

2.4. D.4.3 Control conditions

At the rising edge of the input signal IN (0—1), the DIFU instruction is executed, setting the output signal OUT to 1 and setting OUT to 0 after one scan cycle.

When there is no rising edge of the input signal IN, the output signal OUT is held at 0.

The operation timing when the DIFU command is executed is shown in Figure 2-5.

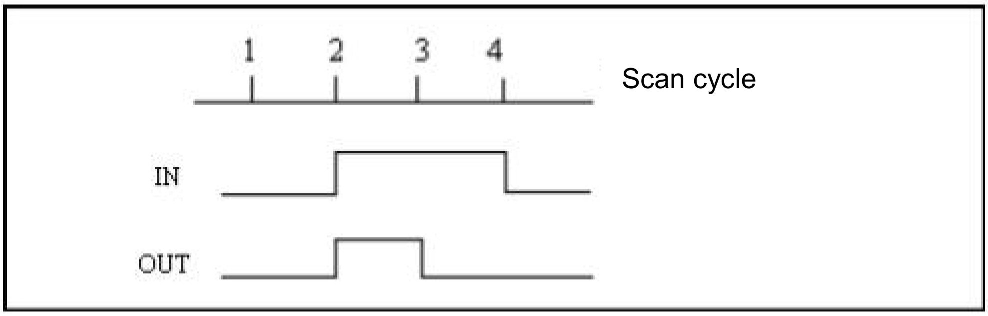


Figure 2-5 Timing of DIFU command operations

2.4. D.4.4 Command parameters

The DIFU command has no command parameters.

2.4. D.5 DIFD (falling edge detection)

2.4. D.5.1 Function

The DIFD command detects the state of the input signal and outputs a pulse signal when the falling edge of the signal occurs.

1) (a) If the DIFD command is in the first level of the program, its width is the scan period of the PLC.

2) If the DIFD command is in the second level program, its width is

Pulse width = PLC scan period x number of program chunks in second level

2.4. D.5.2 Command Format

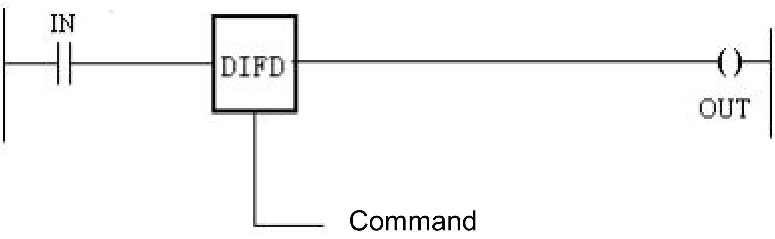


Figure 2-6 DIFD command format

2.4. D.5.3 Control conditions

At the falling edge of the input signal IN (1—0), the DIFD instruction is executed, setting the output signal OUT to 1 and setting OUT to 0 after one scan cycle.

When the falling edge of the input signal IN does not occur, the output signal OUT is held at 0.

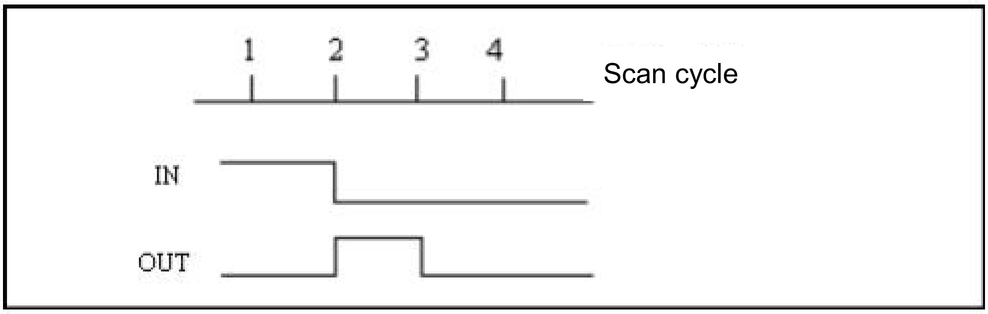


Figure 2-7 Timing of DIFD command operations

2.4. D.5.4 Command parameters

The DIFD command has no command parameters.

2.4. D.6 TURN (signal reversal)

2.4. D.6.1 Function

The TURN command is used to switch the target result state and hold it, with each trigger changing the state once and so on.

2.4. D.6.2 Command Format

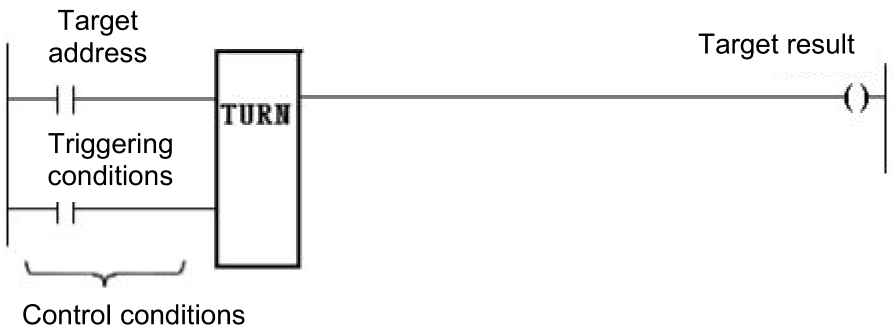


Figure 2-24 TURN command ladder format

2.4. D.6.3 Command parameters

1) Target address

   The target address to be controlled.

2) Trigger conditions

   When the trigger condition is on, a switch to the target address state is performed and held, with each trigger state changing and holding the output, and so on.

3) Target results

   The result output of the destination address.

2.4. E: Data comparison command

2.4. E.1 COMP (compare values)

2.4. E.1.1 Function

COMP is used to compare the magnitude of two given values. When data 1 > data 2, the output is low; when data 1 W data 2, the output is high.

2.4. E.1.2 Command Format

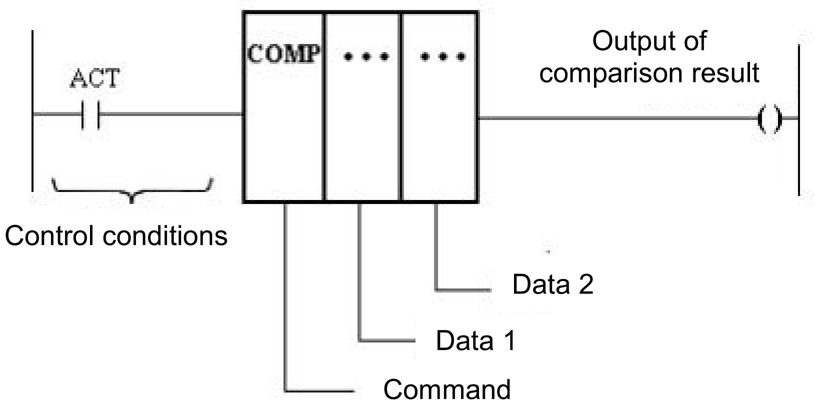


Figure 2-25 COMP command format

2.4. E.1.3 Control conditions

ACT=0: No COMP comparison command is executed, the COMP output remains unchanged.

ACT=1: Execute the COMP comparison command and output the result of the comparison to a relay (the relay is specified by the designer.)

2.4. E.1.4 Command parameters

2) Data 1, Data 2

Specifies the two data to be used for comparison. This can be set to either a constant or a data table address. When set to a constant, the constant is the data to be compared; when set to a data table address (which can only be a two-word data table address), the value corresponding to the data table address is the data to be compared.

2.4. E.1.5 Result output

If data 1 > data 2, the output is low;

If data 1 ≤ data 2, the output is high.

2.4. E.2 COINB (Byte Consistency Comparison)

2.4. E.2.1 Function

The COINB command is used to compare the contents of two byte address cells in a data table to see if they are equal.

2.4. E.2.2 Command Format

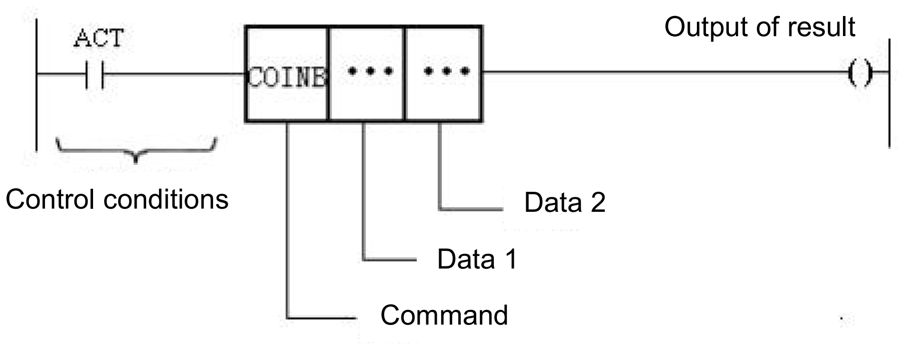


Figure 2-26 COINB command ladder format

2.4. E.2.3 Control conditions

ACT=0: No COINB command is executed and no output of the COINB command (no change in the state of the output relay.)

ATC=1: The COINB command is executed and a comparison result is output.

2.4. E.2.4 Command parameters

1) Data 1, Data 2

Specifies the data to be used for byte consistency comparisons. This can be set to either a constant or a byte data table address. When set to a constant, the constant set is the data used for comparison; when set to a byte data table address, the value corresponding to the data table address is the data used for comparison.

2.4. E.2.5 Result output

If ACT=0, output low. Otherwise:

Data 1 = Data 2: output high.

Data 1 ≠ Data 2: output low.

2.4. E.3 COIW (Consistency of Word Comparison)

2.4. E.3.1 Function

The COINW command is used to compare the contents of two word address cells in a data table to see if they are equal.

2.4. E.3.2 Command Format

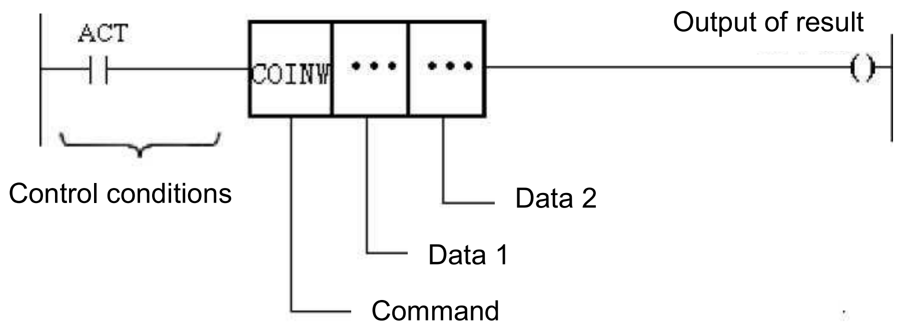


Figure 2-27 COINW command ladder format

2.4. E.3.3 Control conditions

ACT=0: No COINW command is executed and no output of the COINW command (no change in the state of the output relay).

ATC=1: COINW command is executed and a comparison result is output.

2.4. E.3.4 Command parameters

1) Data 1, Data 2

Specifies the data to be used for word consistency comparisons. This can be set to either a constant or a word data table address. When set to a constant, the constant set is the data used for comparison; when set to a word data table address, the value corresponding to the data table address is the data used for comparison.

2.4. E.3.5 Result output

If ACT=0, output low. Otherwise:

Data 1 = Data 2: output high.

Data 1 ≠ Data 2: output low.

2.4. E.4 COINDW (double word consistency comparison)

2.4. E.4.1 Function

The COINDW command is used to compare the contents of two double--word address cells in a data table to see if they are equal.

2.4. E.4.2 Command Format

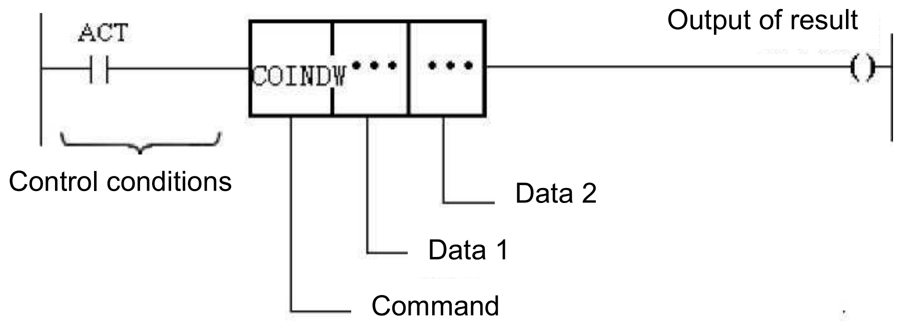


Figure 2-28 COINDW command ladder format

2.4. E.4.3 Control conditions

ACT=0: COINDW command is not executed and no output of the COINDW command (no change in the state of the output relay).

ATC=1: COINDW command is executed and a comparison result is output.

2.4. E.4.4 Command parameters

1) Data 1, Data 2

Specifies the data to be used for a two-word consistency comparison. This can be set to either a constant or a double-word data table address. When set to a constant, the constant set is the data used for comparison; when set to a double-word data table address, the value corresponding to the data table address is the data used for comparison.

2.4. E.4.5 Result output

If ACT=0, output low. Otherwise:

Data 1 = Data 2: output high.

Data 1≠ Data 2: output low.

2.4. E.5 COINS (bit-wise consistency comparison)

2.4. E.5.1 Function

The COINS command is used to compare the contents of a bit address cell in a data table with the equality of the comparison data.

2.4. E.5.2 Command Format

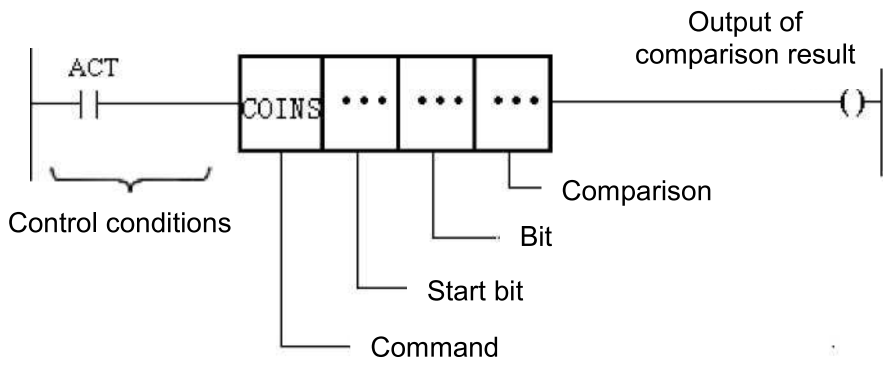


Figure 2-29 COINS command ladder format

2.4. E.5.3 Control conditions

ACT=0: No COINS command is executed and there is no output from the COINS command (the state of the output relay does not change.)

ATC=1: The COINS command is executed and a comparison result is output.

2.4. E.5.4 Command parameters

1) Starting position

Used to specify the start bit of a bit segment (register bit address.)

2) Number of bits

Used to specify the bit segment digits (constant).

3) Compare numbers

Used to specify the comparison data. Can be set to a constant or to a data table address. When set to a constant, the constant is the data to be compared; when set to a data table address, the value corresponding to the data table address is the data to be compared.

2.4. E.5.5 Result output

If ACT=0, output low. Otherwise:

Bit segment data = comparison data, output high.

Bit-segment data≠ comparison data: output low.

2.4. F: Code command

2.4. F.1 SCOD (seven-segment code)

2.4. F.1.1 Function

The seven-segment encoding command encodes the numbers 0 to 9 into binary data that can drive a seven-segment digital tube display.

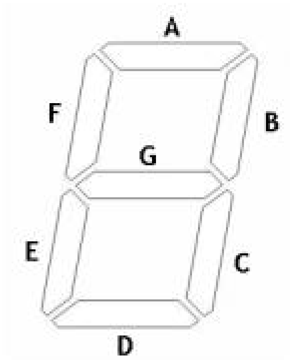


Figure 2-30 Seven-segment digital tube display

2.4. F.1.2 Command Format

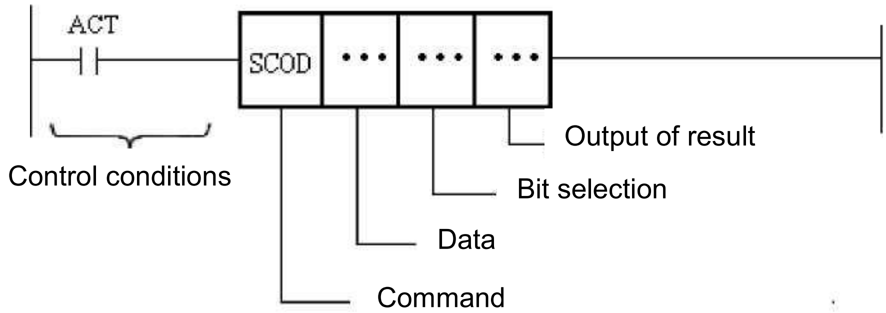


Figure 2-31 SCOD command format

2.4. F.1.3 Command parameters

1) Data

This parameter specifies the address in the data table where the data to be encoded is to be stored.

2) Bit selection:

0 - Encoding of ones place of data

1- Encoding of the tens of the data

2- Encoding the hundreds of the data

3- Encoding of data in thousands of bits

3) Resulting output.

Stores the output of the seven-segment A-G digital tube display.

2.4. F.1.4 Control conditions

ACT=0: The SCOD command is not executed.

ACT=1: SCOD command is executed.

2.4. F.1.5 Examples

Assuming D015=27, program the two seven-segment digital tubes as follows:

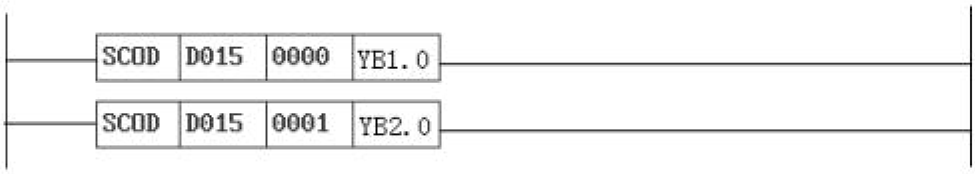


Figure 2-32 Example of a SCOD command

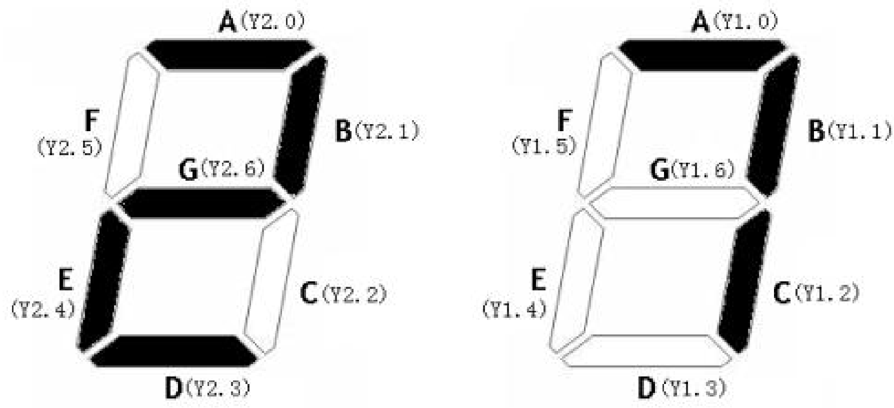


Figure 2-33 Example of a seven-segment digital tube display

2.4. F.2 DEGRAY (Gray Code decoding)

2.4. F.2.1 Function

The Gray Code decoding command specifies the encoding start bit and the number of bits to be encoded to calculate the offset value of the current Gray Code data.

2.4. F.2.2 Command Format

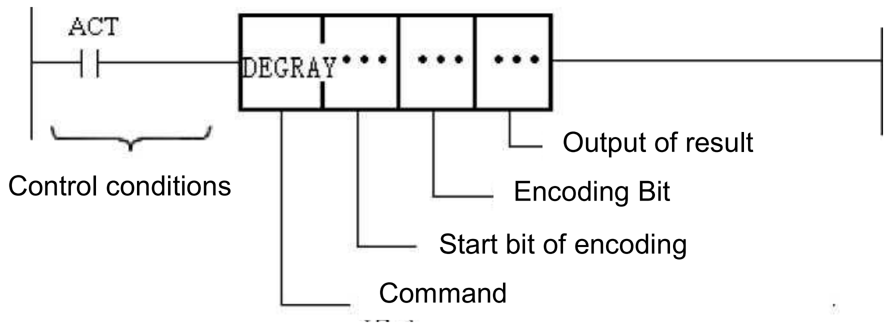


Figure 2-34 DEGRAY command format

2.4. F.2.3 Control conditions

ACT=0: The DEGRAY command is not executed.
ACT=1: The DEGRAY command is executed.

2.4. F.2.4 Command parameters

1) Coding start bit

Used to specify the start bit for reading the current Gray Code data.

2) Number of coding digits

Used to specify the number of bits of the current Gray Code data

3) Result output

Used to specify the address of the data table to be output as a result.

2.4. G: Data search commands

2.4. G.1 DSCH (Data Search)

2.4. G.1.1 Function

The DSCH command performs a search for a given target within the range specified in the data table and returns the target address.

2.4. G.1.2 Command Format

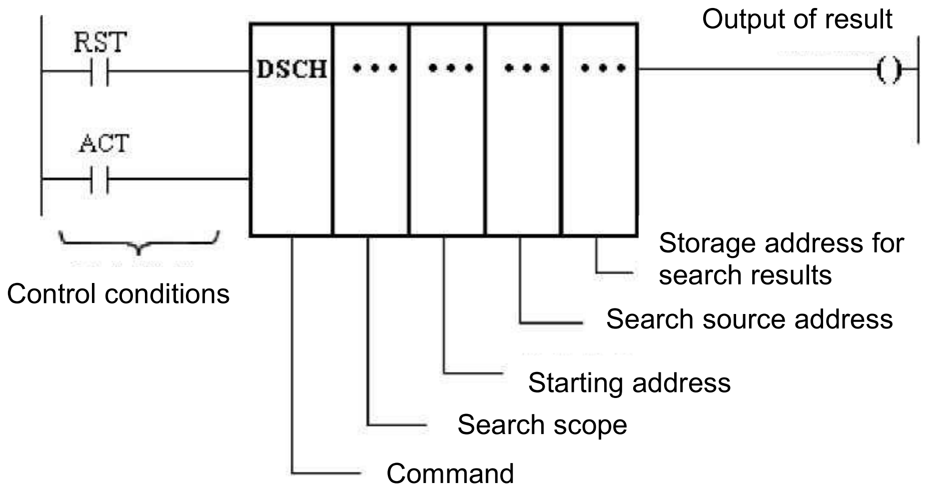


Figure 2-35 DSCH command ladder format

2.4. G.1.3 Control conditions

1) Reset input

RST=0: reset is invalid. the DSCH command is executed normally.

RST=1: reset is active, the DSCH command is not executed and the command output remains low.

2) Permissible conditions for command execution

ACT=0: The DSCH command is not executed and no output is available (the output relay remains in its original state).

ACT=1: The DSCH command is executed normally.

2.4. G.1.4 Command parameters

1) Search scope
Specifies the range of the DSCH command to be retrieved in the data table. Either a constant or the address of that constant in the data table can be entered.
2) Starting address
Specifies the starting address of the DSCH command to be retrieved in the data table. (Parameter 1) and (Parameter 2) together determine the search interval of the DSCH command in the data table as: Start address ～Start address + Search range-1 (i.e. Parameter 2 ～Parameter 2 + Parameter 1-1)
3) Retrieve the source address
Specify the address of the search source in the data table.
4) Search result storage address
If the same data as the search source is found in the search interval, the address of that data in the data table is stored in the address cell specified by this parameter. If there is more than one data in the search interval that is identical to the search source, only the address of the most preceding data is stored.

2.4. G.1.5 Result output

If the same data as the retrieved source is found in the search interval, a high level is output. Otherwise, a low level is output.

2.4. G.2 DSELA (Data Selection A)

2.4. G.2.1 Function

The Data Select A command selects one of a given set of source data to be stored in the specified address unit. The address of the selected data is
Data Address = Data Start Address + Offset Value

2.4. G.2.2 Command Format

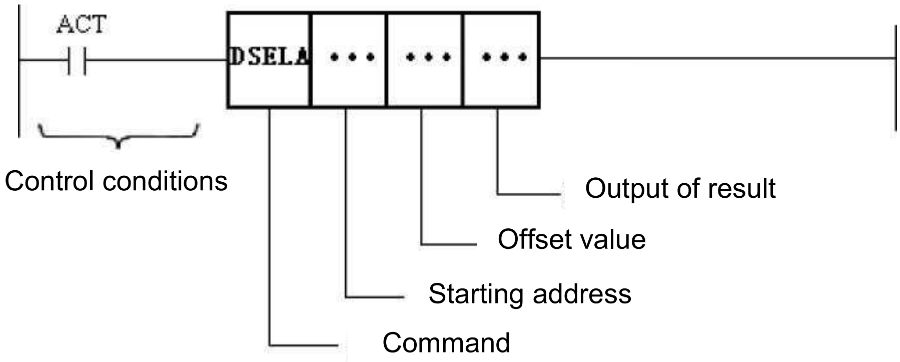


Figure 2-36 DSELA command format

2.4. G.2.3 Control conditions

ACT=0: DSELA command is not executed.
ACT=1: DSELA command is executed.

2.4. G.2.4 Command parameters

1) Data start address: Specifies the start address of the source data in the data table.
2) Offset value: Specifies the offset value. Can be a constant or a data table address.
3) Result output: Specify the address of the data table where the result output will be stored.

2.4. G.2.5 Examples

Assumptions. 1) D010=10; D011=20; D012=30; D013=40; D014=50;
2) R005=3

After the next command is executed, the value 40 of D013 is stored into D005, i.e. D005=40.

Figure 2-37 Example of DSELA

2.4. G.3 DSELB (Data Selection B)

2.4. G.3.1 Function

The Data Select B command selects one of a given set of source data to be stored in the specified address unit. The address of the selected data is

Data Address = Data Start Address + Offset Value

The offset address of the Data Select B command is calculated based on the bits in the offset address register that are 1, i.e. if bit 0 is 1, the offset address is 0, if bit 1 is 1, the offset address is 1, and so on. If more than one bit in the offset address register is 1, then only the lowest bit is valid.

2.4. G.3.2 Command Format

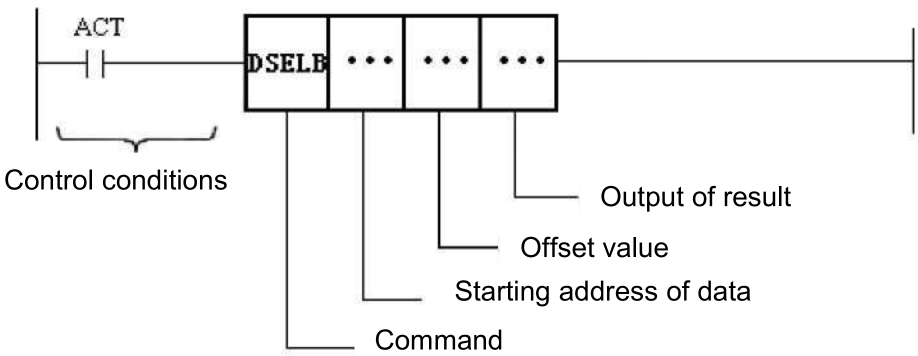


Figure 2-38 DSELB command format

2.4. G.3.3 Control conditions

ACT=0: DSELB command is not executed.
ACT=1: DSELB command is executed.

2.4. G.3.4 Command parameters

1) Data start address: Specifies the start address of the source data in the data table.
2) Offset value: Specifies the offset value. Can be a constant or a data table address.
3) Selection result storage address: Specify the address where the selection data is stored (data table).

2.4. G.3.5 Examples

Assumptions. 1) D010 = 10; D011 = 20; D012 = 30; D013 = 40; D014 = 50.
2) R5.4 = 1, other bits are 0.

R5.4=1, i.e. offset address 4. The next command is executed and the value 50 of D014 is stored in D005, i.e. D005=50.

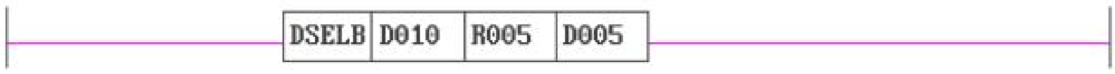


Figure 2-39 Example of a DSELB command

2.4. G.4 DSELC (Data Selection C)

2.4. G.4.1 Function

The Data Select C command selects one of a given set of source data to be stored in the specified address unit. The address of the selected data is

Data Address = Data Start Address + Offset Value

Unlike the Data Select A command which reads data in binary, the Data Select C command reads data in Gray Code when executed.

2.4. G.4.2 Command Format

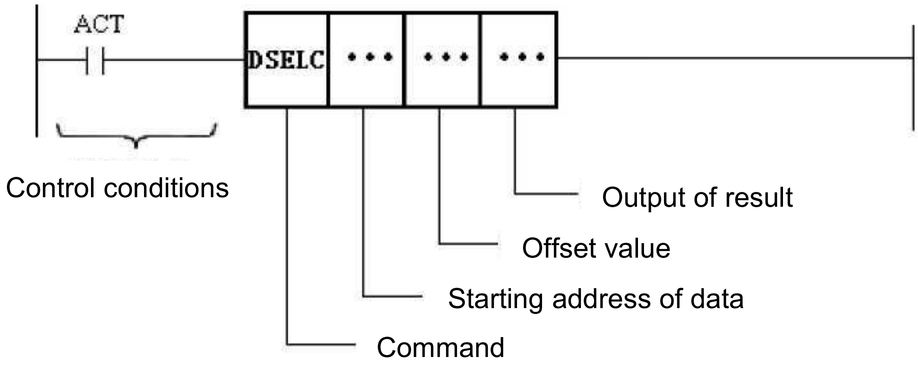


Figure 2-40 DSELC command format

2.4. G.4.3 Control conditions

ACT=0: The DSELC command is not executed.
ACT=1: DSELC command is executed.

2.4. G.4.4 Command parameters

1) Data start address: Specifies the start address of the source data in the data table.
2) Offset value: Specifies the offset value. Can be a constant or a data table address.
3) Selection result storage address: Specify the address where the selection data is stored (data table.)

2.4. G.4.5 Examples

Assuming that. 1) D200 = 10; D201 = 20; D202 = 30; D203 = 40; D204 = 50.
2) K08.1 = 1, K08.2 = 1, the rest of the bits are 0

K08.1=1, K08.2=1, the rest of the bits are 0. After reading in Gray Code mode, the offset address is 4. After the next command is executed, the value 50 of D204 is stored into D400, i.e. D400=50.

Figure 2-41 Example of DSELC

2.4. H: Mathematical operations commands

2.4. H.1 ADD (addition operation)

2.4. H.1.1 Function

The ADD command adds two values and stores the result of the operation in the specified data table address space.

2.4. H.1.2 Command Format

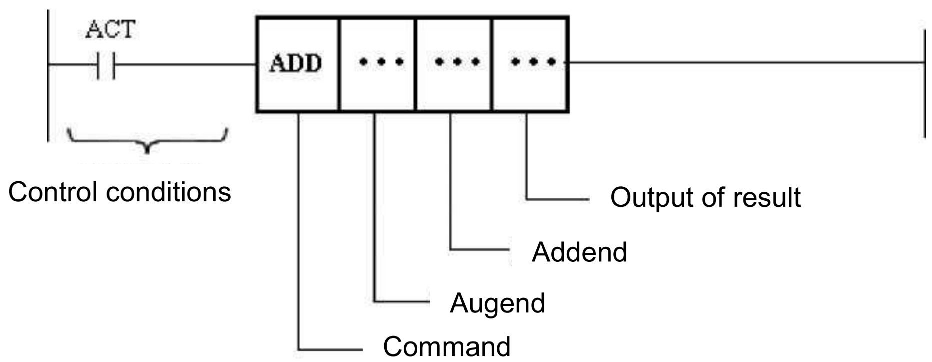


Figure 2-42 ADD command format

2.4. H.1.3 Control conditions

ACT=0: The ADD command is not executed.
ACT=1: Execute the ADD command.

2.4. H.1.4 Command parameters

1) Augends

Specifying the augends. There are two ways of specifying the augends: either by giving the constant (the value of the augend) directly, or by giving the address in the data table where the augend is stored.

2) Addends

Specifying addends. There are two ways to specify an addends: either by giving the constant (the value of the addend) directly, or by giving the address in the data table where the additive is stored.

3) Result output

Specify the storage address in the data table for storing the result of the operation.

2.4. H.1.5 Result output

No result is output for this command.

2.4. H.2 SUB (subtraction operations)

2.4. H.2.1 Function

The SUB command performs the subtraction of two values and stores the result of the operation in the specified data table address space.

2.4. H.2.2 Command Format

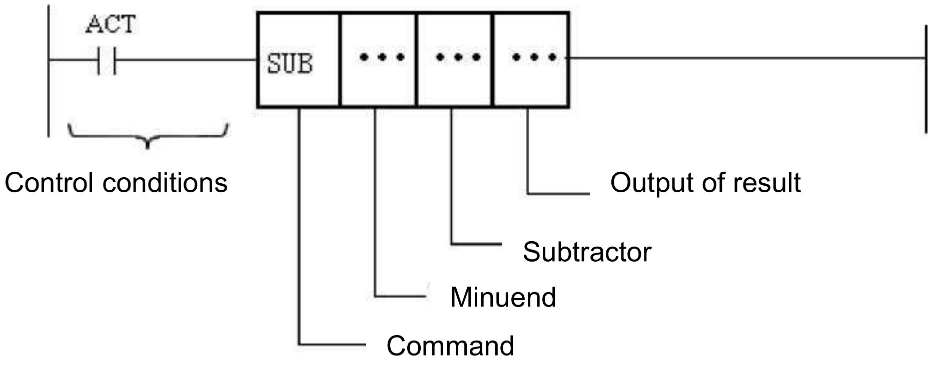


Figure 2-43 SUB command format

2.4. H.2.3 Control conditions

ACT=0: The SUB command is not executed.
ACT=1: Execute the SUB command.

2.4. H.2.4 Command parameters

1) Subtracted number

Specifying the minuend. There are two ways of specifying the minuend: either by giving the constant (the value of the minuend) directly, or by giving the address in the data table where the number to be subtracted is stored.

2) Subtractor

Specifying the subtractor. There are two ways of specifying the subtractor: either by giving the constant (the value of the subtractor) directly, or by giving the address in the data table where the decrement is stored.

3) Result output

Specify the storage address in the data table for storing the result of the operation

2.4. H.2.5 Result output

No result is output for this command.

2.4. H.3 MUL (Multiplication)

2.4. H.3.1 Function

The MUL command is used to multiply two values and store the result of the operation in the specified data table address space.

2.4. H.3.2 Command Format

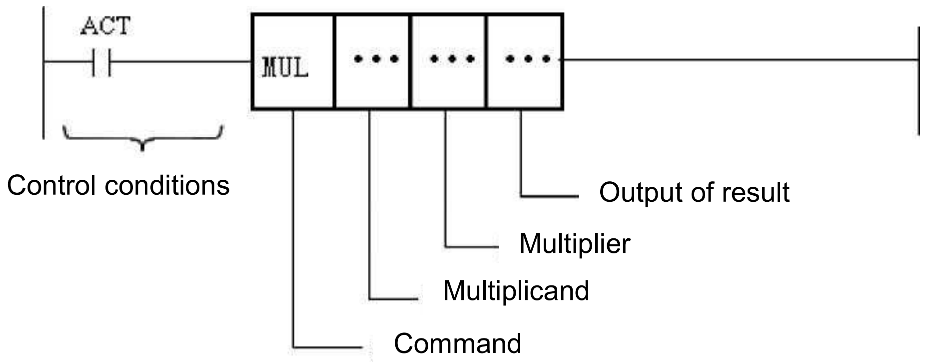


Figure 2-44 MUL command format

2.4. H.3.3 Control conditions

ACT=0: The MUL command is not executed.
ACT=1: Execute the MUL command.

2.4. H.3.4 Command parameters

1) Multiplicand

Specifying the multiplicand. There are two ways of specifying the multiplicand: either by giving the constant (the value of the multiplicand) directly, or by giving the address in the data table where the multiplier is stored.

2) Multiplier

Specifying the multiplier. There are two ways to specify the multiplier: one is to give the constant (the value of the multiplier) directly, the second is to give the address in the data table where the multiplier is stored.

3) Result output Specify the storage address in the data table for storing the result of the operation.

2.4. H.4 DIV (division operations)

2.4. H.4.1 Function

The DIV command performs the division of two values and stores the result in the specified data table address space.

2.4. H.4.2 Command Format

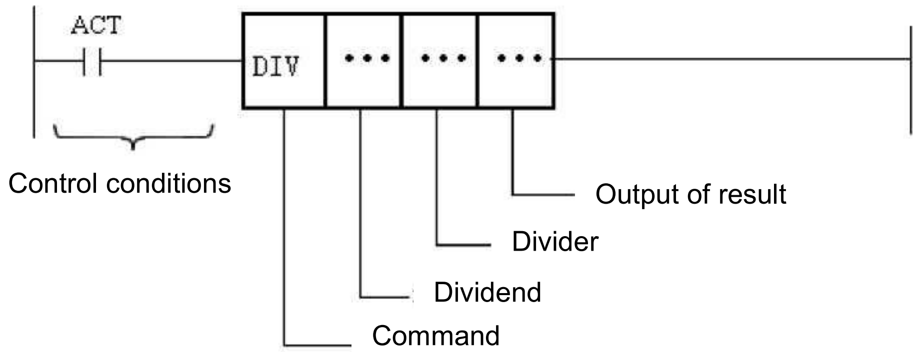


Figure 2-45 DIV command format

2.4. H.4.3 Control conditions

ACT=0: No DIV command is executed.
ACT=1: Execute the DIV command.

2.4. H.4.4 Command parameters

1) Dividend

Specifying the dividend. There are two ways of specifying the dividend: either by giving the constant (the value of the dividend) directly, or by giving the address in the data table where the divisor is stored.

2) Divider

Specify the divider (the divider cannot be 0.) There are two ways to specify the divider: either by giving the constant (the value of the divider) directly, or by giving the address in the data table where the divisor is stored.

3) Result output

Specify the storage address in the data table for storing the result of the operation.

2.4. H.4.5 Result output

No result is output for this command.

2.4. H.5 AND (and operation)

2.4. H.5.1 Function

The AND command implements the "AND" operation on two pieces of data. The two pieces of data participating in the operation perform the "AND" operation by binary bits (true only if both are true.) The result of the operation is stored in the address space of the specified data table.

2.4. H.5.2 Command Format

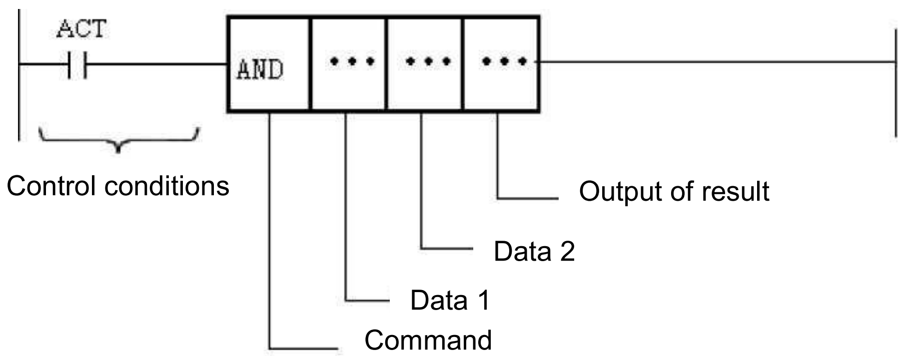


Figure 2-46 AND command format

2.4. H.5.3 Control conditions

ACT=0: The AND command is not executed.
ACT=1: The AND command is executed.

2.4. H.5.4 Command parameters

1) Data 1, Data 2
   Specify the two data involved in the operation. There are two ways to specify the data: firstly, by directly giving the constant (the value of the data,) and secondly, to give the address of the data stored in the data table.
2) Result output
   Specify the storage address in the data table for storing the result of the operation.

2.4. H.5.5 Result output

No result is output for this command.

2.4. H.6 OR (or operation)

2.4. H.6.1 Function

The OR command implements an "or" operation on two pieces of data, where the two pieces of data participating in the operation are "or" by binary bits (false only if both are false.) The result of the operation is stored in the address space of the specified data table.

2.4. H.6.2 Command Format

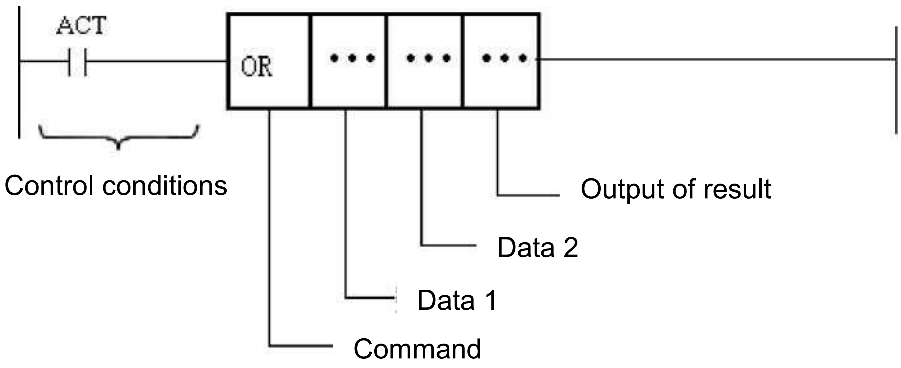


Figure 2-47 OR command format

2.4. H.6.3 Control conditions

ACT=0: The OR command is not executed.
ACT=1: Execute the OR command.

2.4. H.6.4 Command parameters

1) Data 1, Data 2

Specify the two data involved in the operation. There are two ways to specify the data: firstly, by directly giving the constant (the value of the data,) and secondly, to give the address of the data stored in the data table.

2) Result output

Specify the storage address in the data table for storing the result of the operation.

2.4. H.7 XOR (Xor operation)

2.4. H.7.1 Function

The XOR command implements an "iso-or" operation on two pieces of data, where the two pieces of data participating in the operation perform an "iso-or" operation by binary bits (same is false, different is true.) The result of the operation is stored in the specified data table address Space.

2.4. H.7.2 Command Format

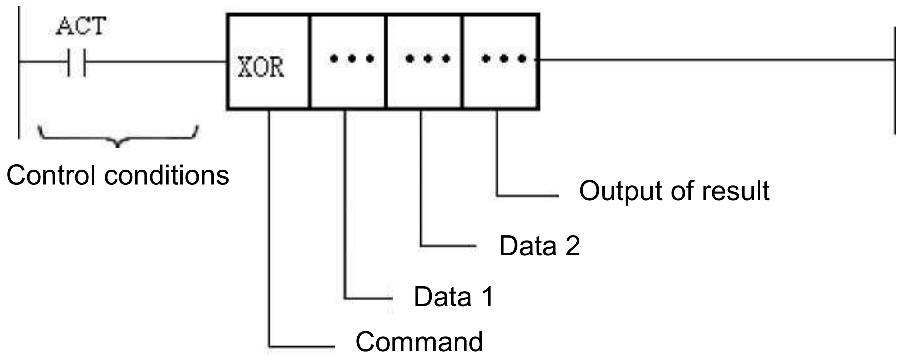


Figure 2-48 XOR command format

2.4. H.7.3 Control conditions

ACT=0: No XOR command is executed.
ACT=1: Execute the XOR command.

2.4. H.7.4 Command parameters

1) Data 1, Data 2

Specify the two data involved in the operation. There are two ways to specify the data: firstly, by directly giving the constant (the value of the data,) and secondly, to give the address of the data stored in the data table.

2) Result output

Specify the storage address in the data table for storing the result of the operation.

2.4. H.7.5 Result output

No result is output for this command.

2.4. H.8 MOD (remainder operation)

2.4. H.8.1 Function

The MOD command performs a 'remainder' operation on two pieces of data and stores the result of the operation in the specified data table address space.

2.4. H.8.2 Command Format

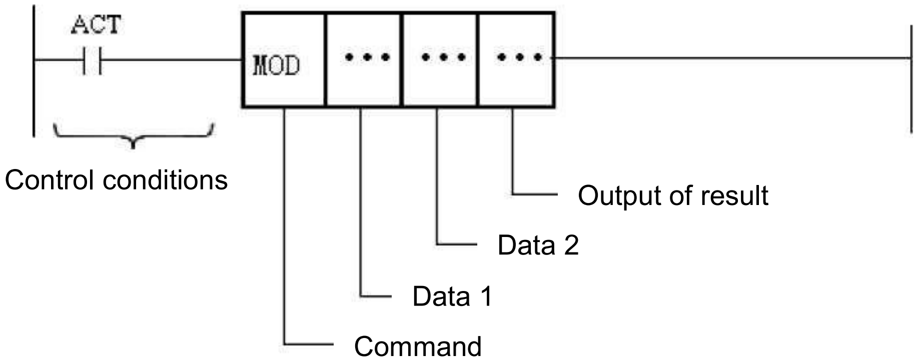


Figure 2-49 MOD command format

2.4. H.8.3 Control conditions

ACT=0: MOD command is not executed.
ACT=1: Execute the MOD command.

2.4. H.8.4 Command parameters

1) Data 1,data 2

Specify the two data involved in the operation. There are two ways to specify the data: firstly, by directly giving the constant (the value of the data,) and secondly, to give the address of the data stored in the data table.

2) Result output

Specify the storage address in the data table for storing the result of the operation.

2.4. I: Channel control command

2.4.I.1 CHMOD (channel operating mode)

2.4.1.1.1 Function

The CHMOD command sets the operating mode for the specified channel.

2.4.1.1.2 Command Format

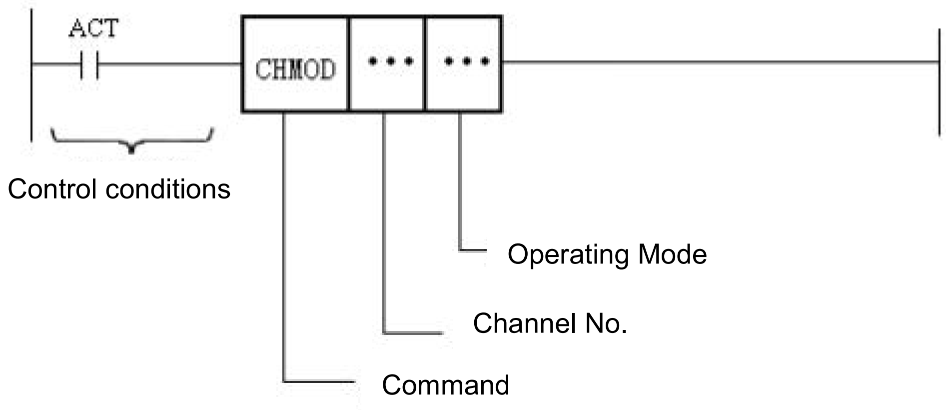


Figure 2-50 CHMOD command format

2.4.I.1.3 Control conditions

ACT=0: CHMOD command is not executed.
ACT=1: CHMOD command is executed.

2.4.I.1.4 Command parameters

1) Channel number

The channel number of the channel to be set.

2) Working mode

Setting the operating mode of the channel.

0: back to reference point; 1: manual; 2: hand wheel; 3: automatic

2.4.l.2 CHAXMOD (Channel Axis Control Mode)

2.4.1.2.1 Function

The CHAXMOD command specifies the control mode for the channel axes within the channel.

2.4.1.2.2 Command Format

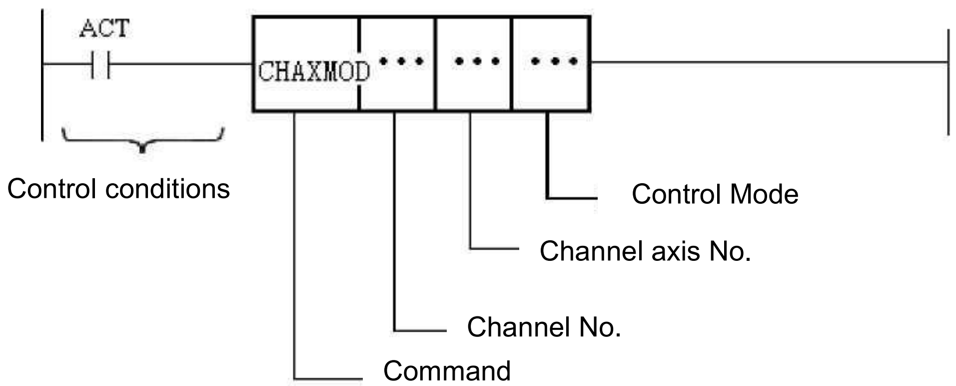


Figure 2-51 CHAXMOD command format

2.4.1.2.3 Control conditions

ACT=0: the CHAXMOD command is not executed.
ACT=1: CHAXMOD command is executed.

2.4.1.2.4 Command parameters

1) Channel number

The channel number of the channel to be set.

2) Channel axis number

Setting the channel axis number within the channel.

3) Control mode

Setting the control mode for this channel axis.

0-channel control; 1-speed control; 2-position control; 3-zero return control

2.4.I.3 CHAXROT (Channel axis rotation control)

2.4.1.3.1 Function

The CHAXROT command performs direction and speed control for the specified channel axis.

2.4.1.3.2 Command Format

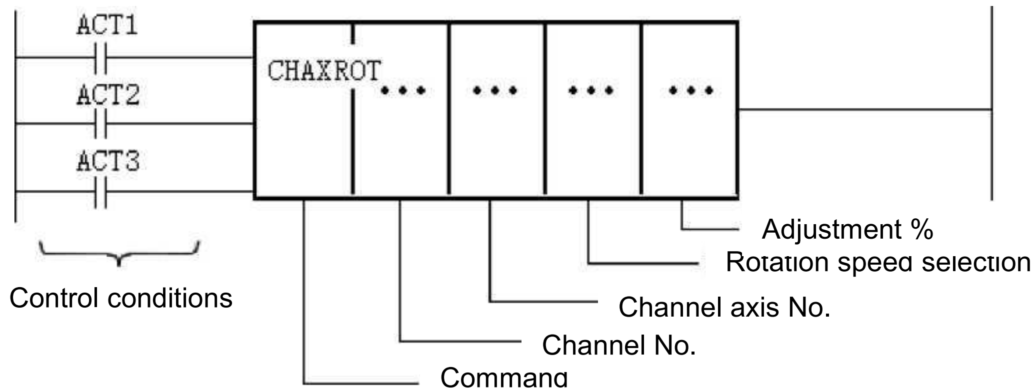


Figure 2-52 CHAXROT command format

2.4.I.3.3 Control conditions

(1) Specify direction of travel

ACT1 = 1: the CHAXROT command performs a forward movement.

ACT2=1: the CHAXROT command performs a negative move.

(2) Command execution permitted conditions

ACT3=0: no CHAXROT command is executed.

ACT3=1: execution of the CHAXROT command.

2.4.1.3.4 Command parameters

1) Channel number
   The channel number of the channel to be set.
2) Channel axis number
   Setting the channel axis number within the channel.
3) Rotational speed selection
   Specify the source of the speed (data table address) that controls this channel axis.
4) Adjustment %

Specifies the adjustment multiplier for this channel axis speed. This can be set to either a constant or a data table address. When set to a constant, the constant set is the adjustment multiplier for the channel axis; when set to a data table address, the value corresponding to the data table address is the adjustment multiplier for the channel axis.

2.4.1.4 CHAXSPD (Channel Axis Speed Control)

2.4.1.4.1 Function

The CHAXSPD command performs direction and speed control for the specified channel axis.

2.4.1.4.2 Command Format

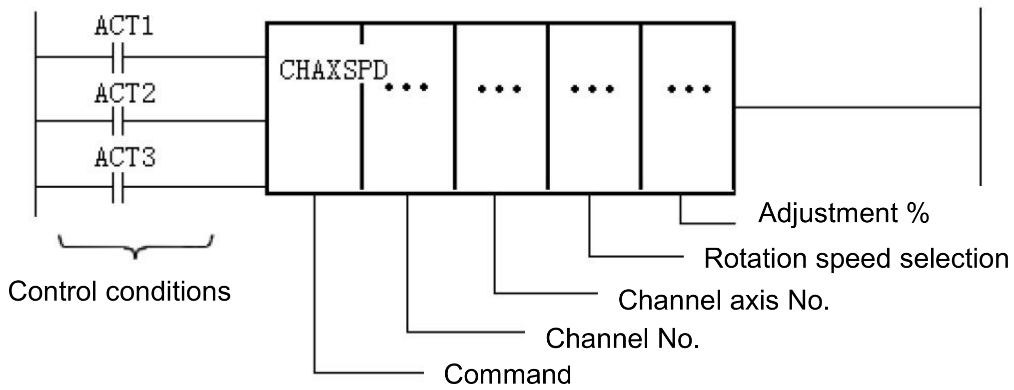


Figure 2-52 CHAXSPD command format

2.4.1.4.3 Control conditions

(1) specify direction of travel

ACT1 = 1: the CHAXSPD command performs a forward move.

ACT2 = 1: the CHAXSPD command performs a negative shift.

(2) command execution permitted conditions

ACT3=0: the CHAXSPD command is not executed.

ACT3=1: execution of the CHAXSPD command.

2.4.1.4.4 Command parameters

1) Channel number
   The channel number of the channel to be set.
2) Channel axis number
   Setting the channel axis number within the channel.
3) Speed selection
   Specify the speed source (data table address)that controls this channel axis.
4) Adjust %

Specifies the adjustment multiplier for this channel axis speed. This can be set to either a constant or a data table address. When set to a constant, the constant is the multiplier for the channel axis; when set to a data table address, the value corresponding to the data table address is the multiplier for the channel axis.

2.4.I.5 CHAXPOS (Channel Axis Position Control)

2.4.I.5.1 Function

The CHAXPOS command performs position control for the specified channel axis.

2.4.I.5.2 Command Format

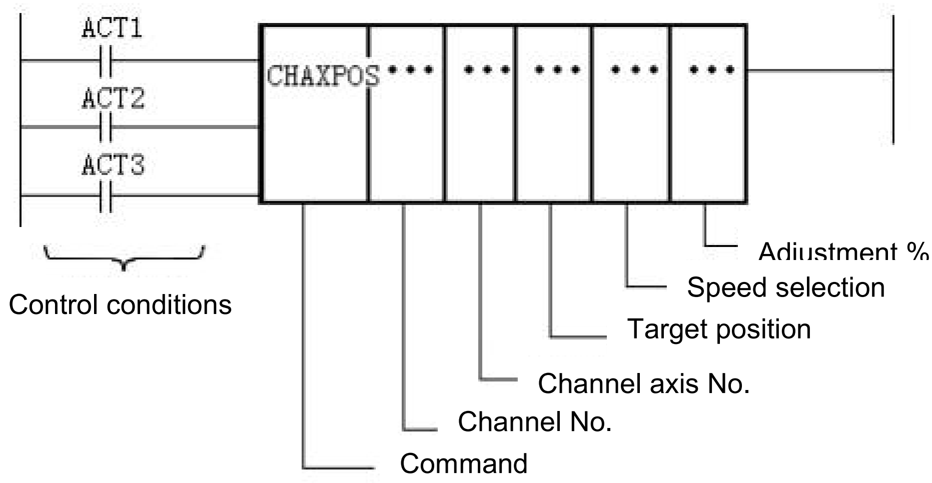


Figure 2-53 CHAXPOS command format

2.4.I.5.3 Control conditions

(1) Command start signal

ACT1=0: CHAXPOS command does not start;

ACT1=1: CHAXPOS command starts.

(2) Command run control signal

ACT2 = 0: the CHAXPOS command stops running;

ACT2 = 1: the CHAXPOS command runs.

(3) Command execution permitted conditions

ACT3=0: the CHAXPOS command is not executed;

ACT3=1: the CHAXPOS command is executed.

2.4.I.5.4 Command parameters

1) Channel number
   The channel number of the channel to be set.

2) Channel axis number

Setting the channel axis number within the channel.

3) Target location

The target position of the controlled channel axis. This can be set to a constant or to a data table address. When set to a constant, the set constant is the target position of the channel axis; when set to a data table address, the value corresponding to the data table address is the target position of the channel axis.

4) Speed selection

Specifies the speed source that controls this channel axis. This can be set to the speed step corresponding to the channel parameter or to the data table address. When set to a speed step, the corresponding parameter is the speed source for that channel axis; when set to a data table address, the value corresponding to the data table address is the speed source for that channel axis.

5) Adjust %

Specifies the adjustment multiplier for this channel axis speed. This can be set to either a constant or a data table address. When set to a constant, the constant is the multiplier for the channel axis; when set to a data table address, the value corresponding to the data table address is the multiplier for the channel axis.

2.4.I.6 CHAXINC (Channel Axis Incremental Control)

2.4.I.6.1 Function

The CHAXINC command performs incremental control for the specified channel axis

2.4.I.6.2 Command Format

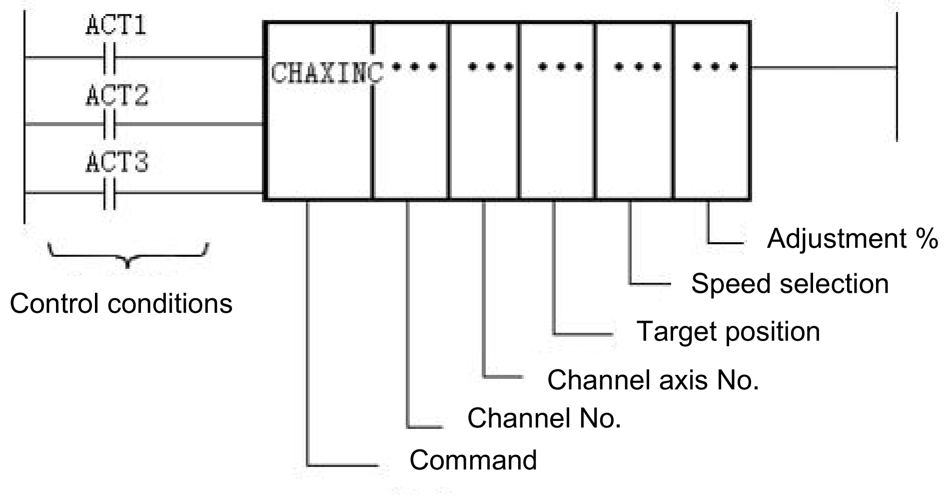


Figure 2-54 CHAXINC command format

2.4.I.6.3 Control conditions

(1) Command start signal

ACT1=0: CHAXINC command does not start;

ACT1=1: CHAXINC command starts.

(2) Command run control signal

ACT2=0: CHAXINC command stops running;

ACT2=1: CHAXINC command runs.

(3) Command execution permitted conditions

ACT3=0: the CHAXINC command is not executed;

ACT3=1: the CHAXINC command is executed.

2.4.I.6.4 Command parameters

1) Channel number

The channel number of the channel to be set.

2) Channel axis number

Setting the channel axis number within the channel.

3) Incremental position

The incremental position of the controlled channel axis. This can be set to a constant or to a data table address. When set to a constant, the set constant is the incremental position of the channel axis; when set to a data table address, the value corresponding to the data table address is the incremental position of the channel axis.

4) Speed selection

Specifies the speed source that controls this channel axis. This can be set to the speed step corresponding to the channel parameter or to the data table address. When set to a speed step, the corresponding parameter is the speed source for that channel axis; when set to a data table address, the value corresponding to the data table address is the speed source for that channel axis.

5) Adjust %

Specifies the adjustment multiplier for this channel axis speed. This can be set to either a constant or a data table address. When set to a constant, the constant is the multiplier for the channel axis; when set to a data table address, the value corresponding to the data table address is the multiplier for the channel axis.

2.4.I.7 CHAXHOME (Channel axis back to reference point)

2.4.I.7.1 Function

The CHAXHOME command performs a return to reference point operation for the specified channel axis

2.4.I.7.2 Command Format

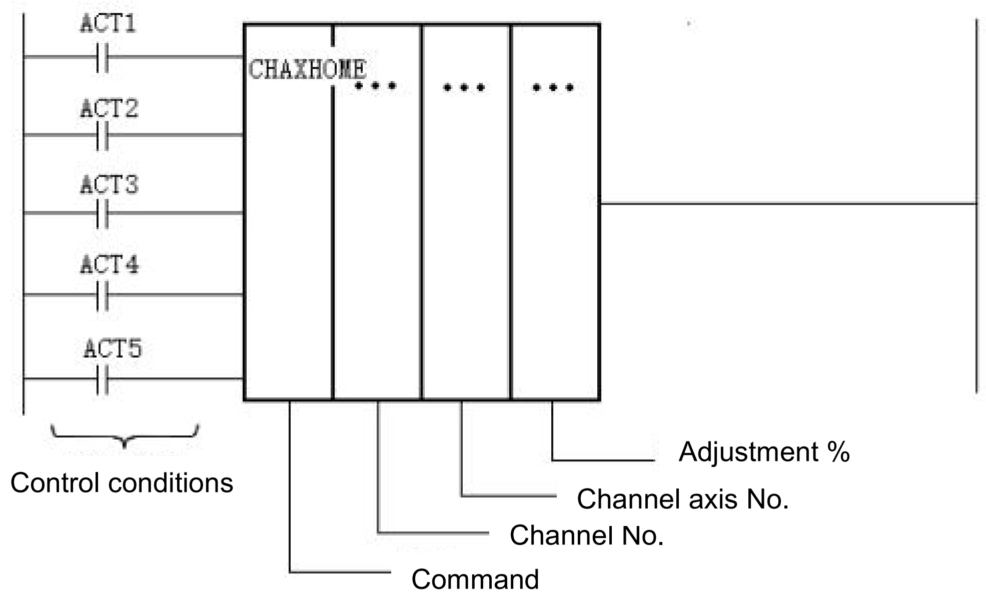


Figure 2-55 CHAXHOME command format

2.4.I.7.3 Control conditions

(1) Conditions for triggering channel axis back to the reference point

ACT1=1: the CHAXHOME command executes a positive return to zero;

ACT2=1: the CHAXHOME command executes a negative return to zero.

(2) ATC3: condition for triggering the zero return switch or zero return block signal for the channel axis;

(3) Condition for triggering channel axis back to reference point termination

ACT4=1: CHAXHOME command execution is terminated.

(4) Condition permitted for command execution

ACT5=0: the CHAXHOME command is not executed;

ACT5=1: the CHAXHOME command is executed

2.4.I.7.4 Command parameters

1) Channel number

Setting the channel number of the channel back to the reference point.

2) Channel axis number

Setting of the channel axis number within the channel.

3) Adjust %

Sets the adjustment multiplier for the speed of this channel axis back to the reference point. This can be set to a constant or to a data table address. When set to a constant, the constant is the multiplier for the channel axis; when set to a data table address, the value corresponding to the data table address is the multiplier for the channel axis.

2.4.I.8 CHPOSCHK (Channel axis position detection)

2.4.I.8.1 Function

The CHPOSCHK command performs position detection for the specified channel axis.

2.4.I.8.2 Command Format

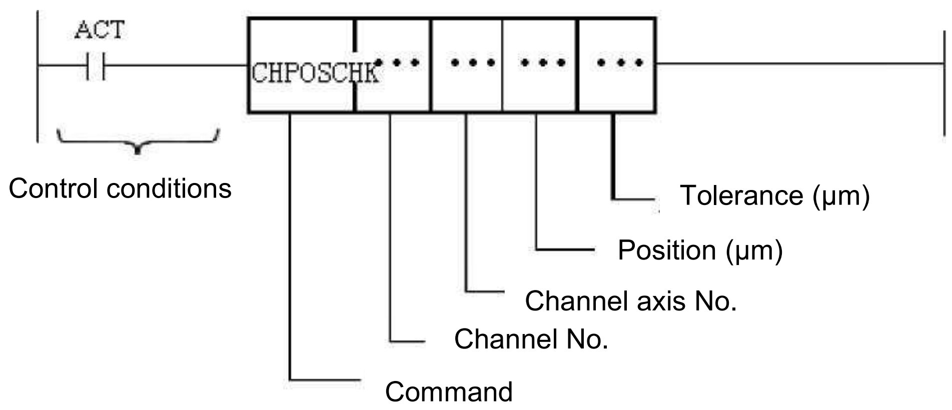


Figure 2-56 CHPOSCHK command format

2.4.1.8.3 Control conditions

ACT=0: The CHPOSCHK command is not executed.
ACT=1: The CHPOSCHK command is executed.

2.4.1.8.4 Command parameters

1) Channel number
   Setting the channel number of the position detection channel.
2) Channel axis number
   Setting of the channel axis number within the channel.
3) Location

   Sets the position value for the channel axis position detection. This can be set to a constant or to a data table address. When set to a constant, the set constant is the position value for the channel axis position detection; when set to a data table address, the value corresponding to the data table address is the position value for the channel axis position detection.
4) Allowance for difference
   Sets the tolerance range for channel axis position detection. This can be set to a constant or to a data table address. When set to a constant, the set constant is the tolerance range for the channel axis position detection; when set to a data table address, the value corresponding to the data table address is the tolerance range for the channel axis position detection.

2.4.I.9 BREAKCHK (channel breakpoint position detection)

2.4.I.9.1 Function

The CHPOSCHK command performs breakpoint position detection for the specified channel axis.

2.4.I.9.2 Command Format

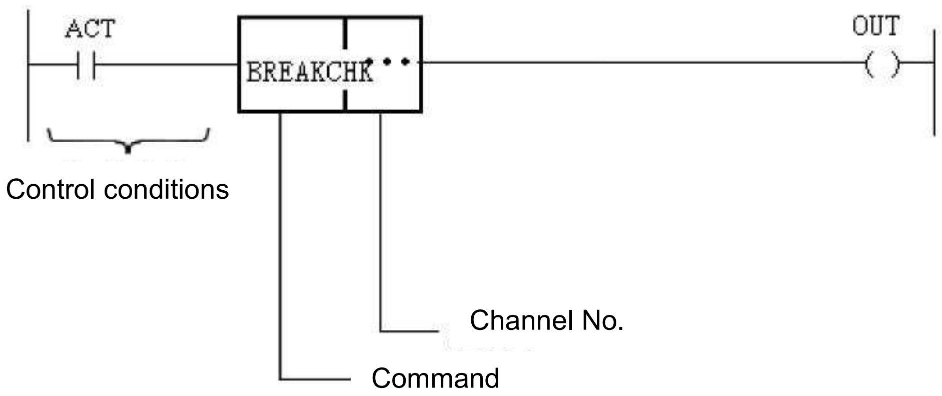


Figure 2-57 BREAKCHK command format

2.4.1.9.3 Control conditions

ACT=0: the BREAKCHK command is not executed.

ACT=1: BREAKCHK command execution.

2.4.1.9.4 Command parameters

1) Channel number

Setting the channel number of the breakpoint position detection channel.

2.4.I.10 CYCLE (channel cycling start)

2.4.I.10.1 Function

The CYCLE command performs program cycling start control for the specified channel.

2.4.I.10.2 Command Format

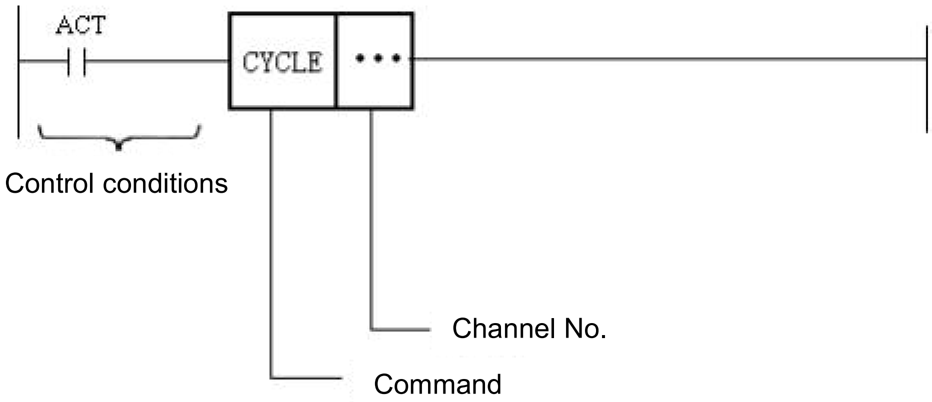


Figure 2-58 CYCLE command format

2.4.I.10.3 Control conditions

ACT=0: the CYCLE command is not executed.
ACT=1: CYCLE command is executed.

2.4.I.10.4 Command parameters

1) Channel number
   Setting the channel number of the position detection channel.

2.4.I.11 HOLD (channel feed hold)

2.4.I.11.1 Function

The HOLD command performs programmed feed hold control for the specified channel.

2.4.I.11.2 Command Format

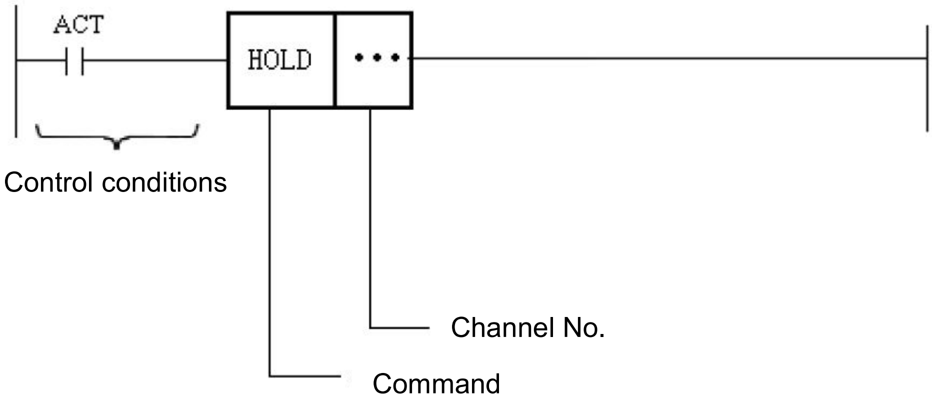


Figure 2-59 HOLD command format

2.4.I.11.3 Control conditions

ACT=0: the HOLD command is not executed.
ACT=1: the HOLD command is executed.

2.4.I.11.4 Command parameters

1) Channel number
Setting the channel number of the position detection channel.

2.4. J: Axis control command

2.4. J.1 AXMOD (axis control mode)

2.4. J.1.1 Function

The AXMOD command sets the control mode for the specified physical axis.

2.4. J.1.2 Command Format

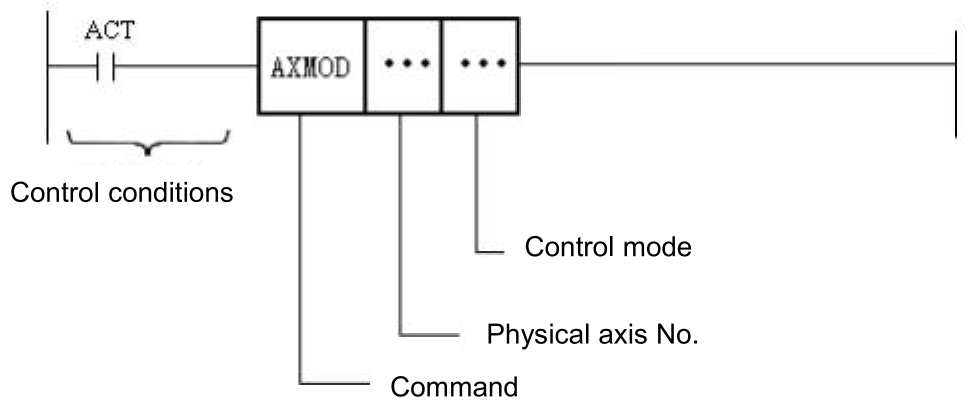


Figure 2-60 AXMOD command format

2.4. J.1.3 Control conditions

ACT=0: the AXMOD command is not executed.
ACT=1: AXMOD command is executed.

2.4. J.1.4 Command parameters

1) Physical axis number
   The axis number of the physical axis to be set.
2) Control mode
   Setting the control mode for that physical axis.
   0-channel control; 1-speed control; 2-position control; 3-zero return control

2.4. J.2 AXROT (axis speed control)

2.4. J.2.1 Function

The AXROT command performs speed control for the specified physical axis.

2.4. J.2.2 Command Format

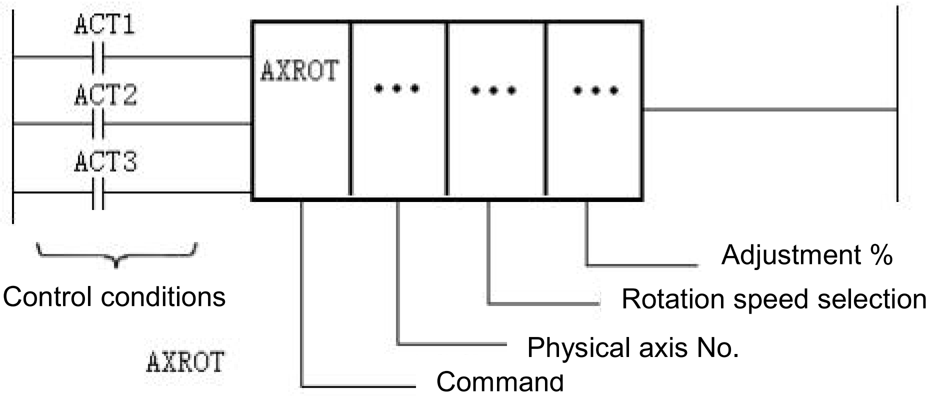


Figure 2-61 AXROT command format

2.4. J.2.3 Control conditions

(1) specify direction of travel

ACT1 = 1: the AXROT command executes a forward movement.

ACT2 = 1: the AXROT command performs a negative move.

(2) Specify the conditions permitted for execution

ACT3=1: AXROT command is executed;

ACT3=0: AXROT command is not executed.

2.4. J.2.4 Command parameters

1) Physical axis number

The axis number of the physical axis to be controlled.

2) Rotational speed selection

Specify the source of the speed (data table address) that controls this physical axis.

3) Adjust %

Specifies the adjustment multiplier for this physical axis speed. This can be set to either a constant or a data table address. When set to a constant, the constant is the speed adjustment multiplier for that physical axis; when set to a data table address, the value corresponding to the data table address is the speed adjustment multiplier for that physical axis.

2.4. J.3 AXSPD (axis speed control)

2.4. J.3.1 Function

The AXSPD command performs speed control for the specified physical axis.

2.4. J.3.2 Command Format

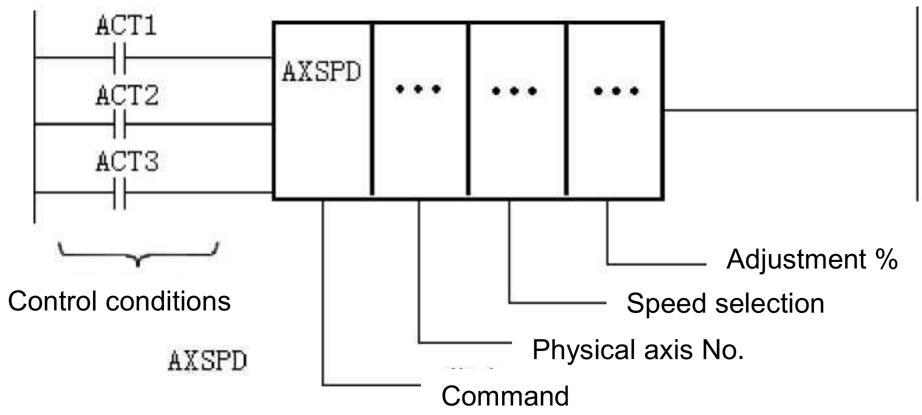


Figure 2-61 AXSPD command format

2.4. J.3.3 Control conditions

(1) specify direction of travel

ACT1 = 1: the AXSPD command executes a forward move.

ACT2 = 1: the AXSPD command performs a negative shift.

(2) Specify the conditions permitted for execution

ACT3=1: AXSPD command is executed;

ACT3=0: AXSPD command is not executed.

2.4. J.3.4 Command parameters

1) Physical axis number
   The axis number of the physical axis to be controlled.
2) Speed selection
   Specify the source (data table address)thatcontrols the speed of this physical axis.
3) Adjust %

Specifies the adjustment multiplier for the speed of this physical axis. This can be set to either a constant or a data table address. When set to a constant, the constant set is the speed adjustment multiplier for that physical axis; when set to a data table address, the value corresponding to the data table address is the speed adjustment multiplier for that physical axis.

2.4. J.4 AXPOS (axis position control)

2.4. J.4.1 Function

The AXPOS command performs position control for the specified physical axis.

2.4. J.4.2 Command Format

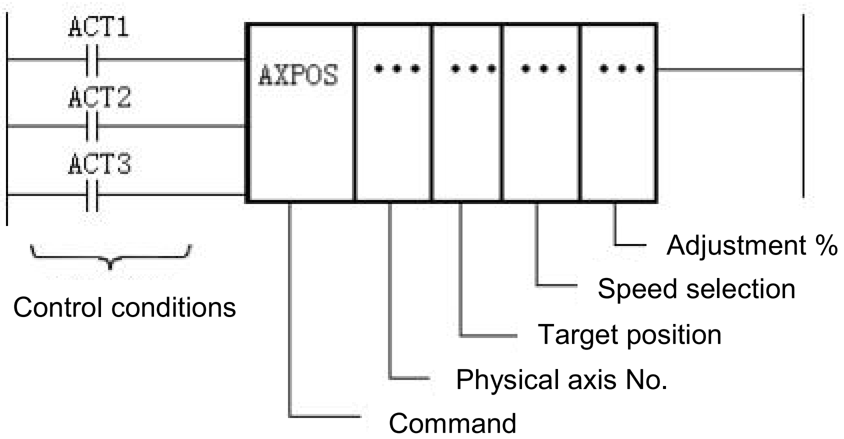


Figure 2-62 AXPOS command format

2.4. J.4.3 Control conditions

(1) Command start signal

ACT1=0: AXPOS command does not start;

ACT1=1: AXPOS command starts.

(2) Specify run control signal

ACT2=0: AXPOS command stops running;

ACT2=1: AXPOS command runs.

(3) Specify the conditions permitted for execution

ACT3=0: AXPOS command is not executed;

ACT3=1: AXPOS command is executed.

2.4. J.4.4 Command parameters

1) Physical axis number

The axis number of the physical axis to be controlled.

2) Target location

The target position of the physical axis being controlled. This can be set to a constant or to a data table address. When set to a constant, the set constant is the target position of the physical axis; when set to a data table address, the value corresponding to the data table address is the target position of the physical axis.

3) Speed selection

Specify the source of speed (constant or data table address)that controls this physical axis.

4) Adjust %

Specifies the adjustment multiplier for the speed of this physical axis. This can be set to either a constant or a data table address. When set to a constant, the constant set is the speed adjustment multiplier for that physical axis; when set to a data table address, the value corresponding to the data table address is the speed adjustment multiplier for that physical axis.

2.4. J.5 AXINC (Axis Incremental Control)

2.4. J.5.1 Function

The AXINC command performs incremental control for the specified physical axis.

2.4. J.5.2 Command format

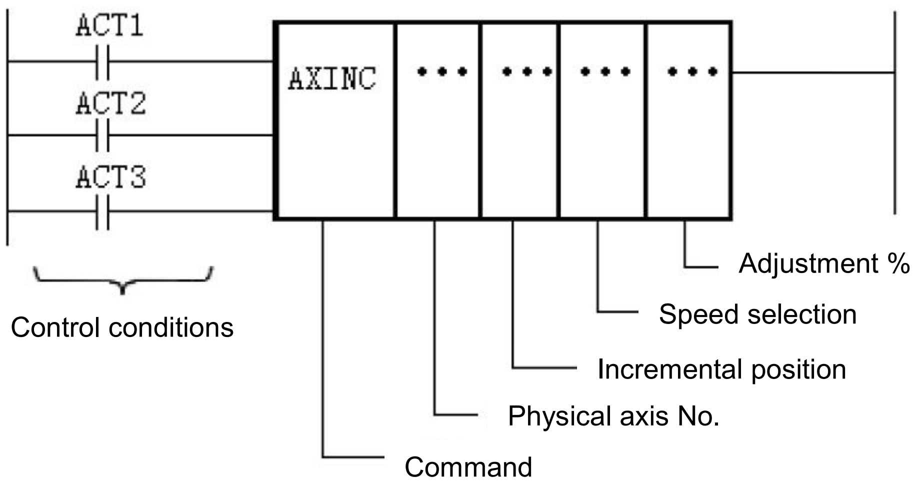


Figure 2-62 AXINC command format

2.4. J.5.3 Control conditions

(1) Specify the start signal

ACT1=0: AXINC command does not start;

ACT1=1: AXINC command starts.

(2) Command run control signal

ACT2=0: AXINC command stops runnning;

ACT2=1: AXINC command runs.

(3) Specify the conditions permitted for execution

ACT3=0: AXINC command is not executed.

ACT3 = 1: AXINC command is executed.

2.4. J.5.4 Command parameters

1) Physical axis number

The axis number of the physical axis to be controlled.

2) Incremental position

The incremental position of the physical axis being controlled. This can be set to a constant or to a data table address. When set to a constant, the set constant is the incremental position of the physical axis; when set to a data table address, the value corresponding to the data table address is the incremental position of the physical axis.

3) Speed selection

Specify the source of speed (constant or data table address)that controls this physical axis.

4) Adjust %

Specifies the adjustment multiplier for the speed of this physical axis. This can be set to either a constant or a data table address. When set to a constant, the constant set is the speed adjustment multiplier for that physical axis; when set to a data table address, the value corresponding to the data table address is the speed adjustment multiplier for that physical axis.

2.4. J.6 AXHOME (axis back to reference point)

2.4. J.6.1 Function

The AXHOME command performs a return to reference point operation for the specified physical axis.

2.4. J.6.2 Command Format

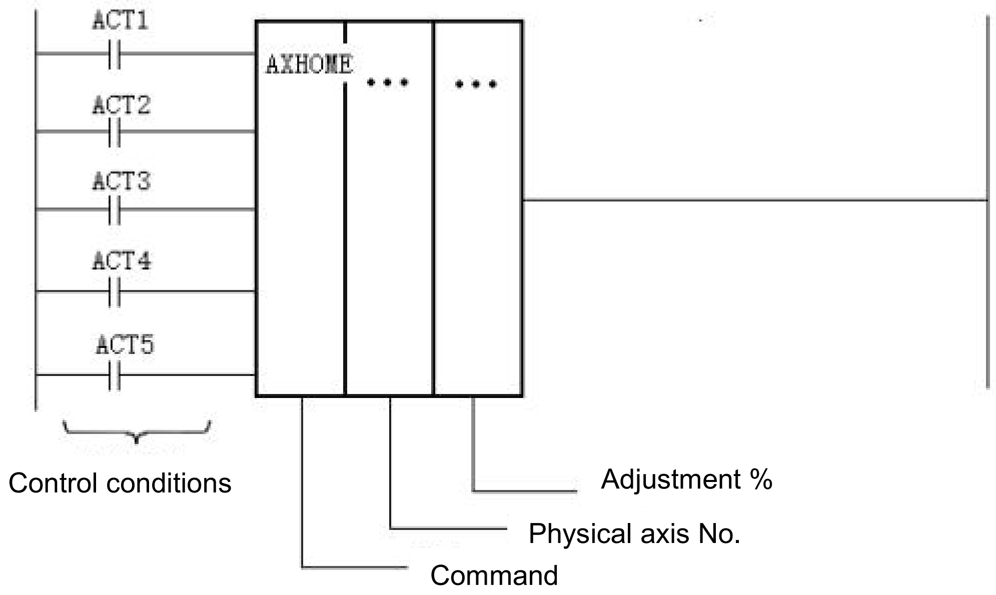


Figure 2-64 AXHOME command format

2.4. J.6.3 Control conditions

(1) Conditions for triggering the axis return to the reference point

ACT1=1: AXHOME command executes a positive return to zero;

ACT2=1: AXHOME command executes a negative return to zero.

(2) ATC3: trigger condition for axis return to zero switch or return to zero stop signal.

(3) Conditions for triggering axis back to reference point termination

ACT4=1: AXHOME command execution is terminated.

(4) Specify the conditions permitted for execution

ACT5=0: the AXHOME command is not executed;

ACT5=1: the AXHOME command is executed.

2.4. J.6.4 Command parameters

1) Physical axis number

The axis number of the physical axis to be controlled.

2) Modification %

Specifies the adjustment multiplier for the speed of this physical axis. This can be set to either a constant or a data table address. When set to a constant, the constant set is the rate at which the physical axis is to be adjusted back to the reference point; when set to a data table address, the value corresponding to the data table address is the rate at which the physical axis is to be adjusted back to the reference point.

2.4. J.7 POSCHK (axis position detection)

2.4. J.7.1 Function

The POSCHK command performs position detection for the specified physical axis.

2.4. J.7.2 Command Format

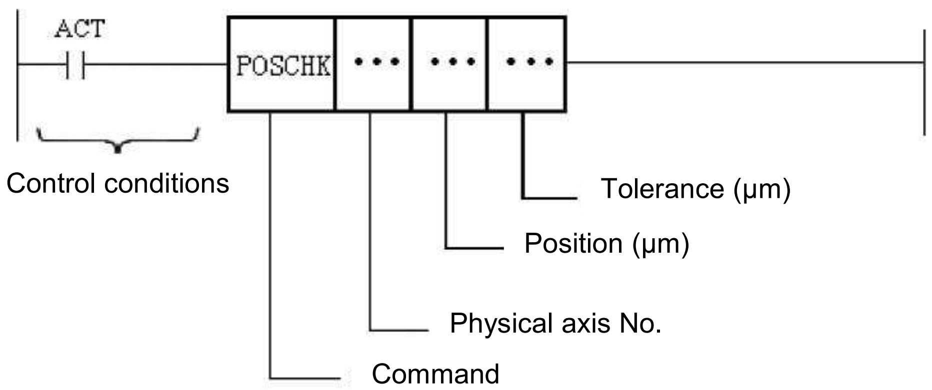


Figure 2-65 POSCHK command format

2.4. J.7.3 Control conditions

ACT=0: the POSCHK command is not executed.

ACT=1: POSCHK command is executed

2.4. J.7.4 Command parameters

1) Physical axis number

The axis number of the physical axis to be tested.

2) Location

Sets the position value for axis position detection. This can be set to a constant or to a data table address. When set to a constant, the set constant is the position value for the axis position detection; when set to a data table address, the value corresponding to the data table address is the position value for the axis position detection.

4) Tolerance

Sets the tolerance range for axis position detection. This can be set to a constant or to a data table address. When set to a constant, the constant is the tolerance range for the axis position detection; when set to a data table address, the value corresponding to the data table address is the tolerance range for the axis position detection.

2.4. J.8 SPCTRL (analogue spindle)

2.4. J.8.1 Function

The direction and speed of the spindle is controlled by a PLC in an analogue way.

2.4. J.8.2 Command Format

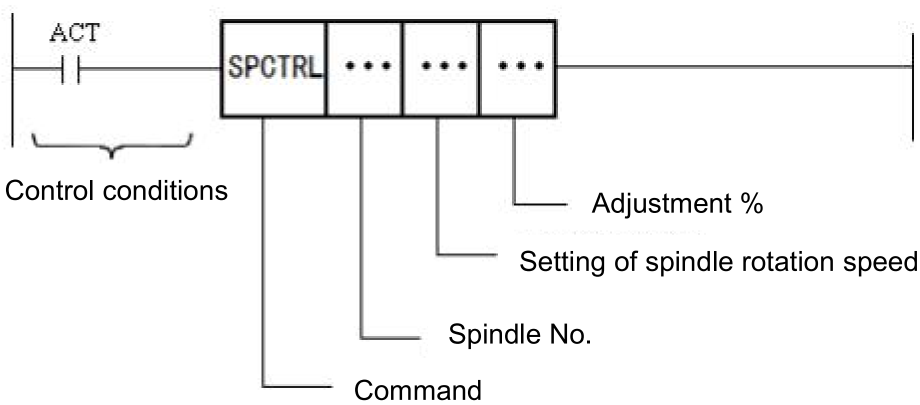


Figure 2-66 SPCTRL command format

2.4. J.8.3 Control conditions

ACT = 0: stopping the rotation of the spindle.
ACT=1; the state of the spindle is controlled by the SPCTRL command.

2.4. J.8.4 Command parameters

Spindle number: sets the port number of the PLC spindle.
Speed setting: sets the rotation speed of the spindle, either as a constant or as a data table address. If set to a constant, the constant is the speed; if set to the data table address, the value corresponding to the data table address is the speed; unit: rpm
Adjustment % : Specifies the adjustment multiplier for the spindle speed. Can be set to a constant or to a data table address. When set to a constant, the constant set is the spindle speed adjustment multiplier; when set to a data table address, the value corresponding to the data table address is the spindle speed adjustment multiplier.

2.4. J.8.5 Result output

No result is output for this command.

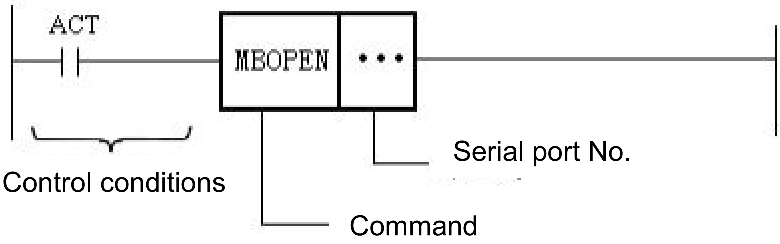
2.4. K: Communication commands

2.4. K.1 MBOPEN (MODBUS open)

2.4. K.1.1 Function

The MBOPEN command is used to open the MODBUS communication.

2.4. K.1.2 Command Format



2.4. K.1.3 Control conditions

ACT=0: the MBOPEN command is not executed.
ACT=1: execution of the MBOPEN command.

2.4. K.1.4 Command parameters

2.4. K.1.5 Result output

2.4. K.2 MBCLOSE (MODBUS close)

2.4. K.2.1 Function

2.4. K.2.2 Command Format

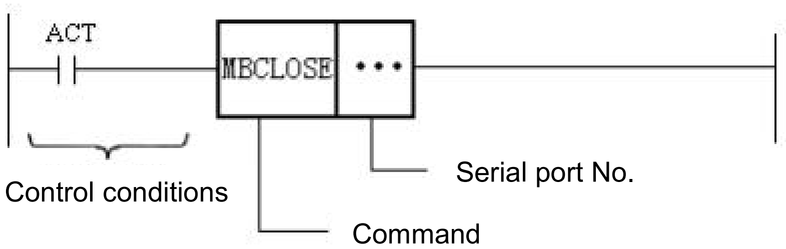


Figure 2-68 MBCLOSE command format

2.4. K.2.3 Control conditions

ACT=0: the MBCLOSE command is not executed.

ACT=1: execution of the MBCLOSE command.

2.4. K.2.4 Command parameters

2.4. K.3 MBCOMMU (MODBUS communication)

2.4. K.3.1 Function

2.4. K.3.2 Command Format

Figure 2-69 MSG command ladder format

2.4. K.3.3 Control conditions

ACT=0: no MSG command is executed.

ACT goes from low to high, the window message command is activated and the PLC sends the content of the message corresponding to the PLC message number to the prompt window bar.

2.4. K.3.4 Command parameters

2.4. K.3.5 Result output

2.4. L: Other commands

2.4. L.1 ROT (rotation command)

2.4. L.1.1 Function

The ROT command is used for rotary control, e.g. for the rotation of tool magazines. Its functions include:

3) Selection of the direction of rotation of the shortest path.
4) Calculation of the position of the target position or the number of steps from the current position to the target position.
5) Calculates the position of the previous position of the target or the number of steps to the previous position of the target.

2.4. L.1.2 Command Format

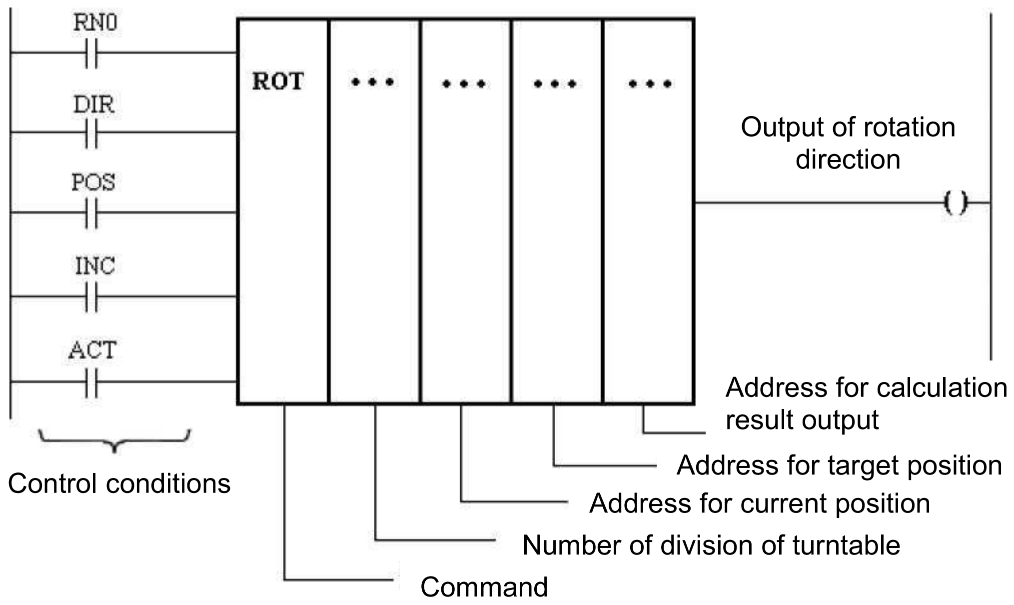


Figure 2-70 ROT command format

2.4. L.1.3 Control conditions

1) Specify the start position number of the rotary table
   RN0 = 0: the position number of the rotary table starts with 0.
   RN0 = 1: the position number of the rotary table starts with 1.
2) Specify whether to select the shortest path rotation direction
   DIR=0: no short path is selected and the rotation direction is always positive;
   DIR=1: the shortest path rotation direction is selected.
3) Specify the operating conditions
   POS=0: calculation of the target position;
   POS=1: calculation of the position preceding the target position.
4) Specify the calculation position or number of steps
   INC=0: number of positions calculated;
   INC=1: number of steps calculated.
5) Permissible conditions for command execution
   ACT=0: no ROT command is executed and the ROT output is unchanged.
   ACT=1: execution of the ROT command.

2.4. L.1.4 Command parameters

1) Indexing of rotary tables
   For specifying the number of indexes of the rotary table, e.g. the number of tool positions in the tool magazine, either by entering a constant or the address where the constant is stored in the data table.
2) Current location address
   The storage address of a given current position number, e.g. the current tool position number of a tool magazine.
3) Target location address
   Giving the storage address of the target position number, e.g. the tool position number of the tool to be selected for the tool magazine.
4) Output address for calculation results
   Gives the address where the result of the calculation (target position or number of steps from the target position) is stored.

2.4. L.1.5 Result output

The output of the command results in a forward and reverse signal: a low level is output when the calculation results in a positive rotation, and a high level is output when the calculation results in a reverse rotation

2.4. L.2 DAOUT (analogue output)

2.4. L.2.1 Function

The speed of the spindle is controlled via the PLC as an analogue voltage output.

2.4. L.2.2 Command Format

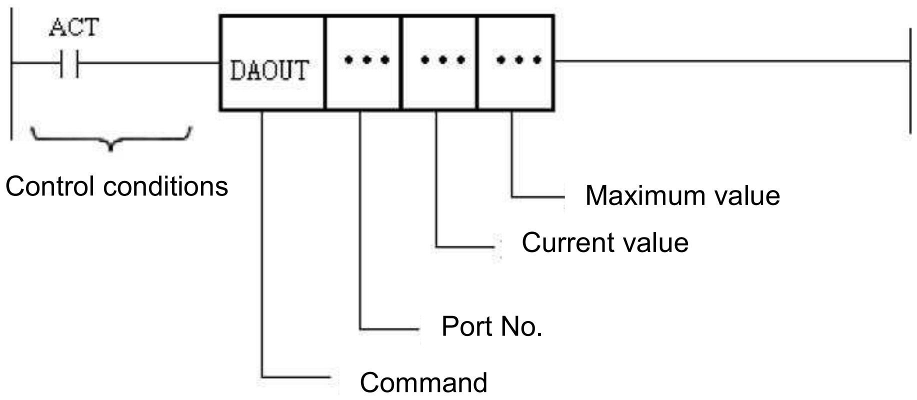


Figure 2-71 DAOUT command format

2.4. L.2.3 Control conditions

ACT=0: the DAOUT command is not executed.
ACT=1: the speed of the spindle is controlled by the DAOUT command.

2.4. L.2.4 Command parameters

Port number: sets the port number of the PLC spindle.

Current value: Sets the rotation speed of the spindle, either as a constant or as a data table address. If set to a constant, the set constant is the speed; if set to the data table address, the value corresponding to the data table address is the speed; unit: rpm

Max. value: Sets the maximum spindle speed corresponding to a voltage of 10V. Can be set to a constant or to the data table address. If set to a constant, the constant set is the maximum speed corresponding to the 10V voltage; if set to the data table address, the value corresponding to the data table address is the maximum speed at 10V voltage.

2.4. L.3 MSG (Window Message)

2.4. L.3.1 Function

Send a prompt message to the controller's prompt window bar via the PLC.

2.4. L.3.2 Command Format

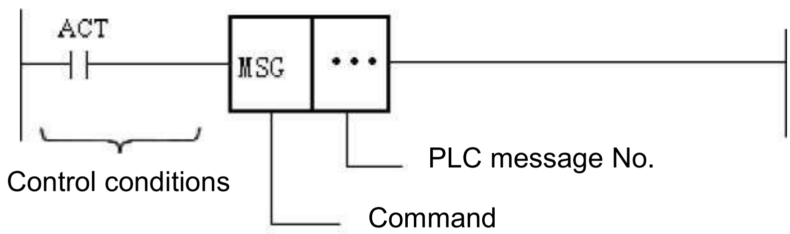


Figure 2-72 MSG command ladder format

2.4. L.3.3 Control conditions

ACT=0: no MSG command is executed.

ACT goes from low to high, the window message command is activated and the PLC sends the content of the message corresponding to the PLC message number to the prompt window bar.

2.4. L.3.4 Command parameters

PLC message number: This is used to specify the ID number of the MSG (window message) to be used. There are a total of 200 PLC messages assigned to the controller PLC and the PLC message number can be set from 1 to 200. The content of the output message is set in the table in [PLC Message] window.

2.4. L.3.5 Result output

No result is output for this command.

3 Appendix: PLC signals table

3.1 CNC controller side input signal (F)

The CNC controller-side input signals are fed from the CNC controller to the PLC control module. The following table shows the input signals that have been defined and used by the CNC controller.

<table>
<tr><th colspan="2">Signal</th><th>Definition</th></tr>
<tr><td rowspan="3">F0
controller Status Register</td><td>F0.0</td><td>controller reset</td></tr>
<tr><td>F0.5</td><td>controller alarms</td></tr>
<tr><td>F0.16~F0.19</td><td>Channel 0~Channel 3 Enable</td></tr>
<tr><td colspan="2">F5</td><td>controller timer (milliseconds)</td></tr>
<tr><td colspan="2">F6</td><td>Hand crank counter</td></tr>
<tr><td rowspan="4">F7
Hand crank status register</td><td>F7.0~F7.4</td><td>Hand crank axis selection X/Y/Z/4/5</td></tr>
<tr><td>F7.5~F7.8</td><td>Hand crank magnification X1/X10/X100/X1000</td></tr>
<tr><td>F7.9</td><td>Hand-cranked emergency stop</td></tr>
<tr><td>F7.10~F7.11</td><td>Hand crank shaft option 6/7</td></tr>
<tr><td>F10</td><td>F10.0</td><td>Kaiyuan-M5</td></tr>
</table>

controller type mark	F10.1	Kaiyuan-M5Plus-V
	F10.2	Kaiyuan-M5Plus-H
	F10.3	Kaiyuan-M5-II
	F10.4	Master-X6
	F10.5	Master-X6Plus-V
	F10.6	Master-X6Plus-H
	FB20.0	Current channel number
	F20.8~F20.11	Channel 0 to 3 Status bits
	FB21.0~FB21.3	Channel Sync Register (0-3)
F[50+50*X] Channel 0~3 Status word	F50.0	Operating mode: Back to reference point
	F50.1	Operating mode: Manual
	F50.2	Operating mode: Hand wheel
	F50.3	Operating mode: Automatic
	F50.5	Working mode: Demonstration
	F50.6	Working mode: Reproduction
	F50.7	Operating mode: Remote
	F50.17	Channel reset
	F50.20	Access brakes

F[51+50*X] Channel 0~3 shaft mounting sign word	F51.0	Shaft 0 Mounting marks
	F51.1	Shaft 1 Mounting marks
	F51.2	Shaft 2 Mounting marks
	F51.3	Shaft 3 Mounting marks
	F51.4	Shaft 4 Mounting marks
	F51.5	Shaft 5 Mounting marks
	F51.6	Shaft 6 Mounting marks
	F51.7	Shaft 7 Mounting marks
	F51.8	Shaft 8 Mounting marks
F[52+50*X] Channel 0~3 program status word	F52.0	Automatic program operation
	F52.1	MDI program running
	F52.2	Feed hold
	F52.3	Program stop
	F52.4	Automatic program readiness
	F52.5	MDI mode
	F52.6	MDI command ready
	F52.7	RTCP On flag
	F52.10	Single stage stop

	F52.14	Program verification
	F52.17	MST lock
	F52.24	Program interrupt flag
	F52.29	Clockwise arc
	F52.30	Counterclockwise arc
	F52.31	Rigid tapping
F[80+50*X]		Channel 0~3 Axis Status Register
F[80+X] Axis 0 to Axis 8 status register	F80.0	Axis 0 Control mode (bit 0)
	F80.1	Axis 0 Control mode (bit 1)
	F80.2	Axis 0 Control mode (bit 2)
	F80.4	Axis 0 Zero speed
	F80.5	Axis 0 Position lock
	F80.8	Axis 0 Back to zero
	F80.9	Axis 0 Return to zero complete
	F80.10	Axis 0 Servo ready
	F80.11	Axis 0 Servo enable ON
	F80.12	Axis 0 Servo speed control mode
	F80.15	Axis 0 Servo online

	F80.16	Axis 0 Positive movement state
	F80.17	Axis 0 Negative movement state
	F80.18	Axis 0 position control complete (reach target position)
	F80.19	Axis 0 Position control termination (does not necessarily reach the target location)
	F80.24	Axis 0 1st reference point in place
	F80.25	Axis 0 2nd reference point in place
	F80.26	Axis 0 3rd reference point in place
	F80.27	Axis 0 4th reference point in place
	F80.28	Axis 0 5th reference point in place
	F80.29	Axis 0 6th reference point in place
F[500+X*10] Physical axis 0~15 Status register	F500.0	Axis 0 Control mode (bit 0)
	F500.1	Axis 0 Control mode (bit 1)
	F500.2	Axis 0 Control mode (bit 2)
	F500.4	Axis 0 Zero speed
	F500.5	Axis 0 Position lock
	F500.8	Axis 0 return-to-zero status
	F500.9	Axis 0 Return to zero complete
	F500.10	Axis 0 Servo ready

	F500.11	Axis 0 Servo enable
	F500.12	Axis 0 Servo speed control mode
	F500.15	Axis 0 Servo online
	F500.16	Axis 0 Positive movement state
	F500.17	Axis 0 Negative movement state
	F500.18	Axis 0 position control complete (reach target position)
	F500.19	Axis 0 Position control termination (does not necessarily reach the target location)
	F500.24	Axis 0 1st reference point in place
	F500.25	Axis 0 2nd reference point in place
	F500.26	Axis 0 3rd reference point in place
	F500.27	Axis 0 4th reference point in place
	F500.28	Axis 0 5th reference point in place
	F500.29	Axis 0 6th reference point in place
Spindle 0 register	F660.0	Positive spin
	F600.1	Reversal
	F600.4	Location mode
	F660.16	Spindle ready
	F660.17	Targeted completion

	F660.18	Zero Speed
	F660.19	Speed to
	F660.20	Give an alarm

3.2 CNC controller side output signal (G)

CNC controller-side output signals are signals that are output from the PLC control module to the CNC controller and are processed by the CNC controller. The following table shows the output signals that have been defined and used by the CNC controller.

Signal		Definition
G0 controller Control Register	G0.0	controller emergency stop
	G0.1	controller reset
	G0.5	PLC operation enable
	G0.9	User operator rights
	G0.10	User administrator rights
G[50+50*X] channel 0~3 control words	G50.0	Operating mode: Back to reference point
	G50.1	Working mode: Manual
	G50.2	Working mode: Handwheel
	G50.3	Working mode: Automatic
	G50.5	Working mode: Demonstration teaching
	G50.6	Working mode: Reproduction

	G50.7	Operating mode: Remote
	G50.16	Passage emergency stop
	G50.17	Channel reset
	G50.20	Emergency braking
	G50.24	Servo reset
	G50.25	Servo enable
	G50.31	Servo speed mode switching
G[52+50*X] Channel 0~3 Program control word	G52.0	Program start
	G52.1	Feed hold
	G52.2	Program stop
	G52.3	Program reset
	G52.4	MDI mode
	G52.10	Single segment
	G52.11	Jump Paragraphs
	G52.12	Empty run
	G52.13	M01 Enable
	G52.14	Program verification
	G52.17	Auxiliary (MST) lock

	G52.20	Single week mode
	G52.21	Repeat mode
	G52.25	Run interrupt service program 1
	G52.26	Running the interrupt service program 2
	G52.27	Running the interrupt service program 3
	G52.28	Running the interrupt service program 4
	G52.29	Running the interrupt service program 5
G[53+50*X] Channel 0~3 manual control word	G53.0	Manual coordinate controller selection (real axis coordinate controller)
	G53.1	Manual coordinate controller selection (world coordinate controller)
	G53.2	Manual coordinate controller selection (Local coordinate controller)
	G53.3	Manual coordinate controller selection (Tool coordinate controller)
	G53.4	Manual coordinate controller selection (cylindrical coordinate controller)
	G53.8	Manual speed selection (1st gear)
	G53.9	Manual speed selection (2 speeds)
	G53.10	Manual speed selection (3 speeds)
	G53.11	Manual speed selection (4 speeds)
	G53.15	Manual RTCP Enable
	G54.0	Hand crank enablement

G[54+50*X] Channel 0~3 hand crank control word	G54.1	Hand crank interruption
	G54.2	Clear hand crank interruption volume
	G54.3	Save hand crank interruptions
	G54.8	Hand crank magnification X1
	G54.9	Hand crank magnification X10
	G54.10	Hand crank magnification X100
	G54.11	Hand crank magnification X1000
	G54.16	Hand crank shaft option X
	G54.17	Hand crank shaft option Y
	G54.18	Hand crank shaft option Z
	G54.19	Hand crank shaft option 4
	G54.20	Hand crank shaft option 5
	G54.21	Hand crank shaft option 6
G[58+50*X] Channel 0~3 adjustment control word	G58.0	Feed adjustment
	G58.1	Quick Move Repair
	G58.2	Spindle adjustmentming
G[60+50*X] Channel 0~3	G60.0	Side head input signal No. 1
	G60.1	Side head input signal No. 2

side head signal	G60.2	Side head input signal No. 3
	G60.3	Side head input signal No. 4
	G60.4	Side head input signal No. 5
	G60.5	Side head input signal No. 6
G[63+50*X] Channel 0~3 Z-axis synchronous control	G63.0~G63.31	Channel 0: Z0~Z31 synchronisation
G[80+50*X]		Channel 0~3 Axis control register
G[80+X] Axis 0 to Axis 8 status register of channel 0 G[130+X] axis 0 to axis 8 status register of channel 1 G[180+X] Axis 0 to Axis 8 status register of channel 1 G[230+X] Axis 0 to Axis 8 status register of channel 1	G80.0	Axis 0 Control mode (bit 0)
	G80.1	Axis 0 Control mode (bit 1)
	G80.2	Axis 0 Control mode (bit 2)
	G80.4	Emergency braking
	G80.5	Position lock
	G80.6	Positive limit switches
	G80.7	Negative limit switches
	G80.8	Positive return to zero
	G80.9	Negative return to zero
	G80.10	Termination back to zero
	G80.11	Zero return switch
	G80.12	External Z pulse

	G80.14	Forward rotation
	G80.15	Negative rotation
	G80.16	Forward movement
	G80.17	Negative movement
	G80.18	Position controlled start
	G80.20	Command Reset
	G80.21	Prohibition of detection of following errors
	G80.22	Second soft limit enable
	G80.24	Servo reset
	G80.25	Servo enable
	G80.31	Servo speed mode switching
G[500+X*10] Physical axis 0~15 Status register	G500.0	Axis 0 Control mode (bit 0)
	G500.1	Axis 0 Control mode (bit 1)
	G500.2	Axis 0 control mode (bit 2)
	G500.4	Emergency braking
	G500.5	Position lock
	G500.6	Positive limit switches
	G500.7	Negative limit switches

	G500.8	Positive return to zero
	G500.9	Negative return to zero
	G500.10	Termination back to zero
	G500.11	Zero return switch
	G500.12	External Z pulse
	G500.14	Forward rotation
	G500.15	Negative rotation
	G500.16	Forward movement
	G500.17	Negative movement
	G500.18	Position controlled start
	G500.20	Command reset
	G500.21	Prohibition of detection of following errors
	G500.22	Second soft limit enable
	G500.31	Servo speed mode switching
	G501.0	Desynchronisation
G[502+10*X]		Target position for physical axis 0~15
G[503+10*X]		Speed selection for physical axis 0 to 15
G[504+10*X]		Speed adjustment for physical axis 0~15

<table>
<tr><td colspan="2">G[508+10*X]</td><td>Servo Extension Control Register - Command Speed</td></tr>
<tr><td rowspan="8">Control word for spindle 0</td><td>G660.4/G660.16</td><td>Switching to position mode</td></tr>
<tr><td>G660.17</td><td>Enable</td></tr>
<tr><td>G660.18</td><td>Oriented</td></tr>
<tr><td>G660.19</td><td>Reversal rotation</td></tr>
<tr><td>G660.20</td><td>Positive rotation</td></tr>
<tr><td>G660.21</td><td>Reset</td></tr>
<tr><td>G662</td><td>Spindle 0 Command speed</td></tr>
<tr><td>G663.0</td><td>Spindle 0 multiplier</td></tr>
</table>

3.3 Internal registers (R)

Internal register address range: R00.x~R99.x.

3.4 Control panel key signal (K/L)

The output signal of the key is used to control whether the key indicator turns on or off. The key input signal is denoted by K and the output signal by L.

3.5 Machine tool side input/output signals (X/Y)

See machine tool manufacturer's manual

3.6 Data table (D)

The data table is a data storage area that can be read and written by the PLC program, with addresses ranging from 0 to 4999. Wherein, D0 ~D1999 is a non-volatile memory area where the data stored is not lost even after a power failure. The rest of the data table addresses are volatile storage areas where data is lost after a power failure and is reset to 0 when power is reapplied.

The following table shows the data table storage units defined and used by CNC controller.

Address	Uses	Remarks
D1000	Tool magazine capacity	
D1001	Cutter table offset address	
D1002	Tool set type table offset address	
D1003	Tool Type Table Offset Address	
D1010	Spindle tool number	Stores the tool number of the tool used on the spindle, a value of 0 in this cell means that there is no tool on the spindle.
D1011	Spindle tool types	
D1012	Current tool set number	Tool position number of the tool magazine in the tool change position.
D1015	S command memory unit	Stores the S-command sent by the CNC controller.
D1016	T command memory unit	Stores the T command from the CNC controller for tool change.
D1016.16	T command signal	This data table is set to 1 when the T command is executed
D1020	Max. spindle speed	
D1100~D1139	Workpiece count, 1~20 occupied	The units for setting the statistics for workpieces 1 to 10 in sequence. Each workpiece occupies two consecutive cells, the first cell stores the preset value of the workpiece statistic and the second cell stores the value of the workpiece statistic.
D1700～D1723	Tool number on tool cases	Tool numbers on the 24 tool cases
D1800～D1823	Type of tool case	Types of 24 tool cases
D1900～D1923	Type of tool	Types of 24 tools

3.7 Power failure save register (B)

The PLC Breakpoint Save Register, a total of 20 double words (640 points), is available as a coil output.

The breakpoint save register is denoted by B in the PLC program.

The breakpoint save register can also be used as a PLC parameter and the user can customise the use of each parameter.

3.8 Timer register (T)

The timer register is a data storage area that can be read and written by the PLC program, with an address range of 0 to 99.

Address	Uses	Remarks
T0000~T0099	Timer, 1~50 occupied	The memory units of timer 1~50 are in turn. Each timer occupies two consecutive cells, the first cell stores the timer preset value and the second cell stores the timer count value. 50 timers occupy a total of 50 x 2 = 100 cells.

3.9 Counter register (C)

The counter register is a data storage area that can be read and written by the PLC program and has an address range of 0 to 39.

Address	Uses	Remarks
C0000~C0039	Counters 1~20 Occupancy	The memory cells for counters 1 to 20 are in order. Each counter occupies two consecutive cells, the first cell stores the preset value of the counter and the second cell stores the count value of the counter. 20 counters occupy a total of 50 x 2 = 40 cells.

3.10 Alarm register (A)

The alarm register is a data storage area that can be read and written by the PLC program in the address range 0.x to 69.x. The alarm register can be used as PLC coil output.

The alarm register is represented by A in the PLC program.
Alarm registers can also be used as PLC parameters and the user can customise the use of each parameter.
The following table shows the alarm registers defined and used by the CNC controller.

Address	Signal definition
A0.0	Serial number error
A0.1	controller is not registered
A0.2	controller usage expires
A0.3	Exceeding the maximum number of control axes
A0.5	Inconsistent controller versions
A0.10	controller reset
A0.11	controller emergency stop
A0.15	PLC has stopped running
A1.0	Passage emergency stop
A1.2	Channel reset
A1.3	Program error
A1.5	Machine tool not in breakpoint position
A1.10	Probe not triggered
A5.0	Not back to zero
A5.1	Failure to return to zero
A5.2	Absolute position initialization failed
A5.3	Servo parameter upload failed
A5.4	Failed to download servo parameters
A5.5	Forward overtravel
A5.6	Negative overtravel
A5.7	Positive soft limi t
A5.8	Soft limit in the negative direction

A5.10	Too large following error
A5.11	Over-speed
A5.16	Loss of stored data

4 Revision history of this Manual

Version No.	Modification	Applicable software
V1.0		V1.0