
1. System Features

MSC-M30 is single/double axis moving and logic controller that our company new designed. The controller include moving and logic control.

This moving controller can implement the control of single axis or double axes(Asynchronous moving).The scale of output pulse frequency is 1-99999Hz,The function of control instructions is strong, it will meet any control request of user. It is a universal moving controller.

The logic control function can implement logical operation (And, Or, Not) to 8 input inside variable, the result that produced or medium can combine with moving control to incorporate moving control and logic control.

Function features:

- Control axis: One or two(X and Y axis), but linkage;
- Pulse frequency:1Hz -100KHz;
- Several rising/falling curve:0-4;
- Pulse output way: Single/Double;
- Speed adjust: Anytime adjust running speed by input port;
- Input point: At most 12(Optical coupler isolated);
- Output point: 2-6 (Optical coupler isolated);
- Display: 16 LED double display, Hint more working state, operation convenience;
- Display mode: You can make the second line display coordinate, angle and any count by Display adjust of Parameter set;
- Instruction feature: Flexible, Practicality, Simple and Strong function;
- Most programming scale:9999999 to 99999999(Pulse number);
- Most programming capacity: 130 lines;
- More operation function: Parameter set, Program manage, logic operation, Auto, Manual, Jog, Clear, Program receive and check-self;
- Logic operation function: 10 logic instructions;
- Program control function: 16 program control instruction, meet any control request;
- Realize cycle operation ,at most three layers;
- Multi-interrupt: At most 8 interrupt program;
- Multi-interrupt source: Different port input, Logic operation result, and inside and outside counter, and give priority to the first set;
- Inside counter: Two, you can add or subtract random number, at most -99999 to 999999;
- Outside counter: Two, add 1 count, the most input frequency less than 10k,the most count range is 0 to 999999;
- Install convenience: The controller can embed in panel;
- Serial communication: Receive program by RS232 from computer, baud rate:9600;
- Single power supply: DC24V(Power error less \pm % 15); inside DC-DC switch.

2. System function

This system have 10 main function, including Parameter set(PS), Program manage(PE),Logic program(LE),Auto(AU),Manual(HM),Jog operation, Coordinate clear(CC), System check-self(SL),Receive program(RP) and Zero position, corresponding digital/function key 1,2,3,4,5,6,7,8,9,0.

State	Hint	Digital function key	Declaration
Para Set	PS	1	System parameter set, set the operation environment that system running, about set means please refer Parameter set part 2.1.
Prog manage	PE	2	Control the edit and manage of program, it can input, amend, insert and delete etc by program edit. Please refer Program manage part 2.2
Logic prog	LE	3	Edit and manage of logic program, it can input, amend ,insert and delete etc by logic edit. Please refer Logic program part 2.3.
Auto	AU	4	Control program auto running. System will running automatic the program that input, at the same time startup logic program(when ST1 is ON),it can control pause, start, and quit. Please refer Auto2.4.
Manual	HM	5	Manual control state, you can operate motor by Up, Down ,Left and Right key. Correspond key press, motor running, loosing motor stop. Please refer Manual operation part 2.5.
Jog operation	JM	6	Jog control state, It can operate motor by up, down, left, right key. Each press, The motor move one "Jog increment". Please refer Jog operation part 2.6.
Coordi clear	CC	7	Clear the current coordinate, set the coordinate is zero. Please refer Coordinate clear part 2.7.
Check-self	SL	8	Check the input and output of system whether normal or not. Please refer System check-self part 2.8.
Receive program	RP	9	Receive the program that computer send and save to program area. Please refer Program receive part 2.9.
Zero position		0	System return to zero position.

Enter function: Press correspond digital/function key.

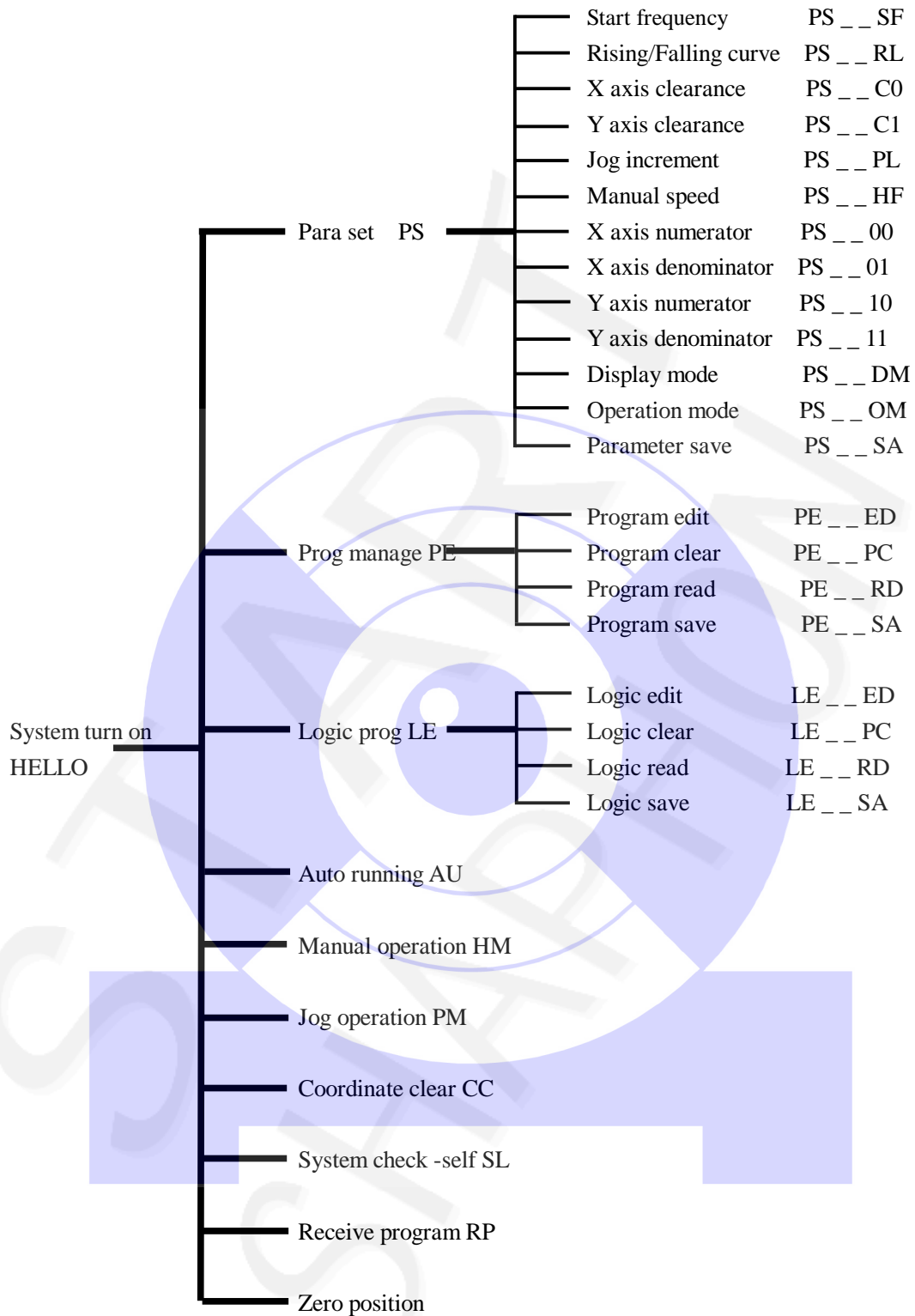
Quit function: Press "Quit" key.

The controller display "HELLO" after turning on. If enter operation function, it will display correspond hint, after quit it will enter main menu, and display "HELLO".

Note: For the tube have only 7 segments, the corresponding character is :

A(A) b(b) C(C) d(d) E(E) F(F) G(G) H(H) i(i) j(j)
 L(L) n(N) o(o) P(P) r(r) S(S) t(t) U(U) v(v)
 y(y) _(_) =(=)

Menu structure:



2.1 Parameter set

Enter “Parameter set”: Under main interface *HELLO*, press “1” to enter, it will hint: *PS __ SF*, press up and down key to change every parameter, press “Quit” to quit.

There are many parameter item in Parameter set, these parameters will infect moving, display operation etc, it need set every parameter according to actual application (frequency, clearance and rising/falling curve have relation with subdivision of driver; the display coordinate have relation

with numerator/denominator), parameter type as follows:

Name	Symbol	Data scope	Declaration
Start frequency	SF	400-50000	If the set value less 400, system will default 400 to deal with, user can set different start frequency according to actual circs.
Rising/falling curve	FL	0-4	There are 5 optimal rising/falling curve in controller, from 0 to 4 gathered speed, you can select different curve according to actual load.
X axis clearance	CO	0-9999	It be used for compensate the error that result from X axis clearance, the displacement don't reckon in coordinate, moving speed execute according to start frequency.
Y axis clearance	CI	0-9999	It be used for compensate the error that result from Y axis clearance, the displacement don't reckon in coordinate, moving speed execute according to start frequency.
Jog increment	PL	1-9999	Under Jog state, every click the button that motor moving control, step motor will move displacement that set.
Manual speed	HF	1-99999	Under manual state , motor's moving speed.
X axis numerator	00	1-32767	Specify the numerator that X axis display scale.
X axis denominator	01	1-32767	Specify the denominator that X axis display scale.
Y axis numerator	10	1-32767	Specify the numerator that Y axis display scale.
Y axis denominator	11	1-32767	Specify the denominator that Y axis display scale.
Display mode	d0	0-4	Select display mode: 0 display the current moving axis's coordinate; 1 display inside count 0 gate's number; 2 display inside count 1 gate's number; 3 display outside count 0 gate's number; 4 display outside count 1 gate's number.
Operation mode	o0	0-1	Select outside operation input's effect state (when ST3 is ON), Four input operation's effect state (earthed effect/off effect)
Parameter save	SA	NO	Parameter amend finish, if need save long time, please execute this function, or else all amend will lost.

Note: Coordinate display scale, it can display length and angle etc by change ratio of numerator (1-65535) and denominator(1-65535).

2.1.1 Amend means

After enter parameter amend, the symbol will first stay on SF, not twinkle. Using up and down key to select different parameter, the symbol changed in turn, but twinkle. After select the parameter that need amend, Press "Enter" and enter ,this moment the symbol and the value twinkle at one time. After input new number, the amend bit move right automatically and keep on twinkle until the last bit. If the twinkle cursor stay the last bit of parameter, cursor will not move to right, you can move cursor by left and right key, press "Enter" to confirm.

After one parameter amend finish, repeat the above state course, amend the other parameter until all parameters are right .Press "Quit" to return to main menu.

If the amend need save long time, please use the “Parameter save SA” that last item of Parameter set, When resume turn on ,all amend are effect.

2.1.2 Display mode instruction

Display mode *dn* define five display mode in running course(include Auto, Manual, Jog etc), The first line one and two bit is function hint(include Auto, Manual, Jog etc.); The four to six bit is sub-function hint; When display moving coordinate, the eight bit is X/Y display hint(\square express X axis, \square express Y axis). The second line display as follows:

DM	Display content	Declaration
0	Coordinate display	Positive number is eight ,for example: \square 12345678, Negative is seven, for example: \square -1234567 The current display value=count \times the axis’s display scale(N/D)
1	In counter 0	Positive number is six, for example : \square 1 123456, Negative is five, for example: \square 1 -12345
2	In counter 1	Positive number is sic, for example: \square 2 123456, Negative is five, for example: \square 2 -12345
3	Out counter0	Count ≥ 0 is six positive number. For example: \square 3 123456
4	Out counter 1	Count ≥ 0 is six positive number, For example: \square 4 123456

2.1.3 Switch set

This system have 4 state(ST) switch (open back door), the definition as follow:

No.	Name	Declaration
1	ST1	ON: Execute logic program, OFF: Forbid executing logic program
2	ST2	ON: Limit effect (I4:X+,I5:X-,I6:Y+,I7:Y-), OFF: Limit void (I4-I7 is common input)
3	ST3	ON: Permit outside operation(I0:Start,I1: Pause, I2: increase speed, I3: decrease speed, OFF: Forbid outside operation (I0-I3 is common input).
4	ST4	ON: Set pulse and direction output mode. OFF: Set double pulse output mode.

Note: 1.Outside operation’s effect signal lie on the set (0/1) of “Operation mode”.

2. When I0-I4 is outside manual or I4-I7 is limit , it can’t join logic operation. If it set is outside manual and limit simultaneously, you should avoid using logic operation.

2.2 Program manage

Enter “Program manage”: Under main interface *HELLO*, Press “2” to enter, the hint change is: *PE...Ed*, Press up and down key to switch every function of program manage, Press “Quit” to quit.

Sub-function instruction:

Name	Symbol	Declaration
Program edit	<i>Ed</i>	Write down new program, it need clear old program or edit the current control program.
Program clear	<i>PC</i>	Clear program area, if it need come back old program(saved control program), you should resume turn on or use program to read sub-function.
Program read	<i>rd</i>	Read the program from EEPROM to control program area, When turn on the system will execute this function automatically.
Program save	<i>SA</i>	Save the program to EEPROM long. The old program lost(system only can save one program).

Note: Please execute “Program clear” sub-function before write down new program.

2.3 Logic program

Enter “Logic program”: Under main interface *HELLO*, Press “3” key to enter, the hint change is: *LE__Ed*, Press up and down key to switch every function of Logic program ,Press “Cancel” to quit. Sub-function list as follows:

Name	Symbol	Declaration
Logic edit	<i>Ed</i>	Write down new program, it need clear old program or edit the current control program.
Logic clear	<i>PC</i>	Clear logic program area, if it need come back old program(saved control program), you should resume turn on or use program to read sub-function.
Logic read	<i>rd</i>	Read the program from EEPROM to logic program area, When turn on the system will execute this function automatically.
Logic save	<i>SA</i>	Save the program to EEPROM long. The old program lost(system only can save one program).

Note: Please execute “Logic clear” sub-function before write down new program.

2.4 Auto

Under main interface *HELLO*, press “4” to enter “Auto waiting for run “ state, waiting for start signal. After start (Press “Start” key or outside start effect), controller will execute program from first line, at one time start logic operation program until the last control instruction END. This moment, Auto running finished, logic operation stop, controller return to “Auto run” state(hint AU).

You should insure there are right control program and logic program before enter auto state, If no logic program, you can set ST1 is OFF. When there are no limit switch, you should set the ST2 is OFF; When limit effective (ST2 is ON), you should connect 4 limit switches (request often closed state, that is when ineffective, switch close). The unused limit can connect with 24V ground, but can not use another.

During Auto running, it will display the current program’s line number and some display information. During auto running ,you can use “Pause” key to stop running, press “startup” key to continue, state instruction as follow:

State	Display	Declaration
Pause	<i>AU SEP</i>	<i>SEP</i> Twinkle (Pause), express the current state is “Pause running”, waiting for “Start” signal. The second line display correspond value (decided by Display mode).
Auto	<i>AU 001</i>	<i>AU</i> (Auto) express “Auto”, <i>001</i> express the current program’s line number, the second line display correspond value (decided by Display mode).

2.5 Manual operation

Under main interface *HELLO*, press “5” to enter “Manual waiting for run”, the hint change is: *HA__AU*, Left and right key control X axis motor; Up and down key control Y axis motor. Press correspond key motor moving, loose motor stop. When running ,the display decide by Display mode.

When limit effective, encounter the direction of limit, it can’t go on moving, but reverse or another axis can move.

2.6 Jog operation

Under main interface *HELLO*, press “6” to enter “Jog moving”, the hint changed is: *PL_ _*, This moment the state of display and control is same to “Manual operation”. But the control mode is different: Each press the control key(up, down, left, right) of motor ,the correspond motor will move the gave value of “Jog increment” (PL). When running, the display decided by the parameter set of Display mode. Refer 3.1.2.

2.7 Coordinate clear

Under main interface *HELLO*, Press “7”, set the current coordinate is Zero point(include X and Y axis).

2.8 System check-self

Enter “System check-self”: Under main interface *HELLO*, press “8” to enter, the hint changed is *SL_ _*, The second row tube display: *1111111178*.

The every segment of tube express different meanings.

The upright of front six tubes is input state display, the number order from left to right: 0,1,2,3,4,5,6,7,8,9,10,11;

The bar of front six tubes is output state display, the number order from left to right: 0,1,2,3,4,5.

Upright No.	0	1	2	3	4	5	6	7	8	9	10	11
Corresponding input	I0	I1	I2	I3	I4	I5	I6	I7	I8	I9	I10	I11
Bar No.	0		1		2		3		4		5	
Corresponding output	O0		O1		O2		O3		O4		O5	

The back two tube express the state of interior variable B0-B9. As right figure shown:

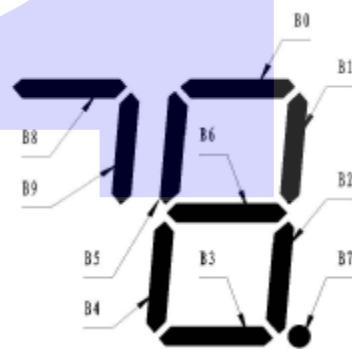
Operation means:

Input state, Connect the input pins (DB25) to ground, then the corresponding upright tube went out, after break, tube bright ;

Output state, Click once output check key(The digital key that bar No corresponding), the output display overturn once (from bright to went out or from went out to bright). Bright is output 0(effective), Went out is output 1(break). If the operation not in accord with display, please check the connect whether right or not. If there are problem again,

it is system trouble, need checking.

Interior variable: The state can changed by logic program, interior variable is 1,then the corresponding tube bright; Value is 0, then the corresponding tube went out.



2.9 Accept program

From computer accept control program. Once operation, band rate is 9600, It need change to

system acceptable program according to the ruled format, then carry to controller(refer Appendix I).

Connect the serial port of computer to RS232 interface, after electricity, Select this function(RP), then press “Enter” to enter accept state, this moment you can execute the sending program of computer.



3. Program edit and instructions explain

3.1 Summarize

MSC-M30 adopt program control mode, have 16 control instructions. It can implement many control requests of user, Strong and flexible instruction can make your control project easy to implement.

3.1.1 Enter and quit

The program edit function can be used to read, amend ,browse operation etc. Enter “Program edit state”: Under *PE--Ed* state, press “Enter” key to enter. After program edit finish, press “Quit” key to quit (before edit new program, use “program clear to clear program in program area).

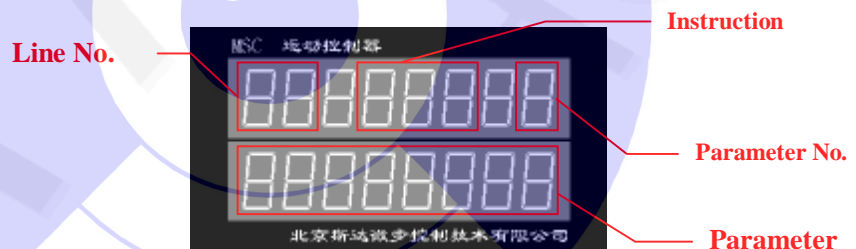
3.1.2 Program line

The controller can edit most 130 lines, don't need input line number, automatic sequence read in. When enter a new program, the first instruction is *Go 1* that system default, twinkle, can rolling to select instruction. According to hint, input all data that this instruction need.

You can insert or delete some line during editing, and updating program order automatically. Select some program line (instruction twinkle), this moment press “Delete” to delete this line, the instruction will change to next line automatically; Press “Enter” key (this moment the “Enter” key be used to insert key) will insert line before this line.

3.1.3 Instruction format

Two lines display:
The first line is instruction information, the second is data input line. The current twinkle position is the position of cursor.



1.Line No.: Two tube display in the front of up row. The scope of line No is 00 to 129(total 130), display 00 to C9; the first tube except for display 0 to 9 and display A, b, C (in order express 100, 110, 120).

2.Instruction area: Three tube display in up row 4-6. (concrete instruction display refer 3.2).

3.Parameter No.: The last tube display in up row. Identify parameter.

4.Parameter area: The all down row display the current data(1 to 8), Concrete parameter please refer 3.3.

The last instruction of program is “End” (If no this line ,execute program possible have error). When it have many exits, should have many “End” instruction. But, when program running to “End”, Control program execute finish.

3.1.4 Program inquire explain

When cursor in up row, you can press left and right key to switch among Line No, Instruction area and Parameter No. When cursor stay Parameter No, you can use up and down key to enter Parameter area(parameter No is not 0) or Hint area(parameter No is 0).

1. Cursor lie in Line No area: (The tube in Line No twinkle continuously)

Press up ,down key: Program forward , back one line;

- Press 9, 6 key: Program forward ,back ten lines;
- 2.Cursor lie in instruction area: (The tube in Instruction area twinkle continuously)
Press up, down key: Forward ,back select instruction circularly;
- 3.Cursor lie in parameter No area: (The tube in Parameter No twinkle continuously)
Press up, down key: Select parameter of display instruction, When forward to 0, up again then enter the before line program; When back to 7, down again then enter the after line program.
- 4.Cursor lie in Parameter area: (The tube in Parameter area twinkle continuously)
Press up, down key: Select the before or after parameter of Display instruction.
Press left, right key: In parameter area, cursor move one bit to left or right.
- 5.Enter Parameter area: When cursor lie in any position of the above state, Press “Enter” key, then enter next parameter area.

3.1.5 Program edit

- 1.When cursor lie in Parameter area, press digital key or negative key to input data.
 - 2.Insert line: Move the cursor to Line No area, press up or down key to find insert position, press insert key (-) to insert .
 - 3.Delete line: Move the cursor to Line No area, press up or down key to find the line to delete.
- Note: A new program line , instruction is **Go 1** that system default and the all parameters are 0. The last instruction of program must is “**End**” (If no this line ,execute program possible have error). When it have many exits, should have many “**End**” instruction. when program running to “**End**”, Control program can execute finish.

3.2 Instruction sort and format explain

3.2.1 Instruction sort

Control instructions have 16 and divided into four kinds:

Moving instruction:	Including Go1, Go2, Go3, Crd, SPd	Five;
Count instruction:	Including LoP, SCn, Cnt	Three;
Jump instruction:	Including JcN, Jbt, JMP, int	Four;
Others:	Including out, Dly, rEt, End	Four.

3.2.2 Instruction parameter

For the instruction include Coordinate, Data, Axis select, Direction select, I/O select ,etc, there are a sort of parameter hint (use digital express) for every instruction, At most have seven parameters, At least one parameter. According to the different of instruction, the number of parameter that with is different and the same number’s meanings have difference also, please you programming after read carefully. The base meanings is :

Symbol	Meanings
0	Instruction code, use left or right cursor select to produce
1	Input “Coordinate value” or “Counter changed value” (Positive or negative), 8 tubes bright. When input “Data” ,including Delay, Speed, Loop number, Count initial value etc less 8 tubes bright.
2	Axis select (0 is X, 1 is Y axis)
3	Direction select (0: Positive ,1: Negative), or Pause return select(0:breakpoint, 1: The next

	line after breakpoint)
4	The tab of destination line
5	Input select (00 to 07 is outside input, 10 to 19 is inside variable) . Output select (00 to 05) Counter select(00 is Cnt0, 01 is Cnt1, 10 is Int0, 11 is Int1)
6	Effective level (Input 0: Connect effective, 1: Break effective), Output level(0: Output low, 1: Output high level) The value compare of counter (0: Over jump, 1: Less jump)
7	This line tab (00-15, Except 00, the other all not allowed to repeat appear in same program to jump, 00 is no tab).

3.3 Program save

Refer 2.2 *SA* function, Press “Enter” key to start saving, it need waiting for about 3 seconds, during this moment press any key ineffective.

Note: For the save area is EEPROM, it have use life (about 1 ten thousand) , so you should try to use little, When program debugging, it need not to save if don't turn off power. When program is right ,you can use save function.

3.4 Instruction explain

No	Name	Display Form	Declaration	Parameter hint(√ Yes, × No)							
				0	1	2	3	4	5	6	7
1	Absolute value moving	G01	Control the selected axis move to the gave coordinate. 1:Select moving coordinate of move axis, 2: Axis select, 7: This line tab If gave move and current coordinate is same, not moving.	√	√	√	×	×	×	×	√
2	Increment moving	G02	Control the selected axis move to the gave value. 1:Select moving axis 's move value2:Axis selectm3:Moving direction 7: This line tab When the move value is 0, according to the gave direction moving and don't stop forever; When it isn't 0, according to the positive or negative of value to confirm direction, no relation with direction parameter.	√	√	√	√	×	×	×	√
3	Condition Stop	G03	Control selected axis move to the gave value or select input effective stop. 1:Selected axis's move value,2:Axis select,3: Moving direction,5:Input path select,6:Input effective state(0/1),7:Tab When the selected input is effective during moving, finish moving in advance, or else until move value to stop. When value is 0, moving according to gave direction, input effective ,stopping.	√	√	√	√	×	√	√	√
4	Set coordinate	GtD	Set selected axis as current coordinate. 1:For set coordinate of set axis2: Axis select 7: This line tab	√	√	√	×	×	×	×	√
5	Delay	dLY	Set delay waiting time. 1:Delay time(ms) 7: This line tab If delay time is 0, it is pause , waiting for start.	√	√	×	×	×	×	×	√
6	Set speed	SPd	Set moving speed (effective later) 1:For set speed(frequency),7: This line tab	√	√	×	×	×	×	×	√
7	Loop	LoP	Set loop start and times of loop 1:loop times,4: Aim tab 7:This line tab	√	√	×	×	√	×	×	√
8	Counter initial	SCn	Set the selected counter initial value. 1:Counter initial5:Counter select7: This line tab	√	√	×	×	×	√	×	√
9	Count amend	Cnt	Amend counter value 1:Amend value (can is negative)5:Counter select7:Tab	√	√	×	×	×	√	×	√
10	Count Jump	JCN	Select counter count jump 1:Compare value4:Aim tab5:Counter select6:Jump condition(0:Over,1:Less),7:Tab	√	√	×	×	√	√	√	√
11	Input Jump	Jbt	Select input bit effective jump. 4:Aim tab,5: input path select,6:Jump condition(0: ineffective jump,1: effective jump 7:This line tab	√	×	×	×	√	√	√	√
12	Unconditional jump	JNP	Unconditional jump. 4:Aim tab7:This line tab	√	×	×	×	√	×	×	√
13	Set interrupt	int	Set interrupt source(input, interior variable, counter and entrance address. 4:Aim tab5:input path select6:jump condition(0:ineffective,1:effective) 7: This line tab	√	×	×	×	√	√	√	√
14	Set output	oUt	Set selected output state 5:Output path select 6:output state(0/1), 7:This line tab	√	×	×	×	×	√	√	√
15	Interrupt return	rEt	Interrupt return 3:Return mode(0: to breakpoint,1:to next), 7:This line tab	√	×	×	√	×	×	×	√
16	Program end	End	Program end (return to main menu). 7:This line tab	√	×	×	×	×	×	×	√

3.5 Application example

【Example One】 Request: Start frequency 2.5KHz, increase or decrease speed is faster, Clearance compensate is 0; X axis running 90000 with the speed 2.9KHz, Y axis running reverse 5000 with the speed 15KHz; Repeat 10 times, then X axis move back to 2000, Y axis move back to 1000 and stop.

Error! Not found source. Parameter list: (Enter to Parameter set state to amend) SF=25000,RL=3,C0=0000,C1=0000.

Error! Not found source. Program list:

No.	Instruction	1	2	3	4	5	6	7	Description
0	SPd	002900						01	Set follow moving speed is 2.9KHz,Program line 01 as loop entrance
1	G02	00090000	0	0				00	X axis positive moving 90000
2	SPd	015000						00	Set follow moving speed is 15KHz
3	G02	-0005000	1	0				00	Y axis moving reverse 5000
4	LoP	000009			01			00	Loop entrance is this line 01, repeat 10 times
5	G01	00002000	0					00	X axis move to coordinate 00002000
6	G01	00001000	1					00	Y axis move to coordinate 00001000
7	End							00	Program end

【Example two】 Request: Start frequency 2.5KHz, increase or decrease speed is slower, X clearance compensate is 12; X axis move to 1234567 with speed 39KHz, then open output 01, after delay 55.9s, X axis move reverse to 234567 with the mode Condition input IO effective, then program pause until after restart motor return to position -888 as the same speed, close output 01,finish.

Error! Not found source. Parameter list: (Enter to Parameter set state to amend) SF=25000, RL=1, CO=00012.

Error! Not found source. Program list:

No	Instruction	1	2	3	4	5	6	7	Description
0	SPd	039000						00	Set follow moving speed is 39KHz
1	G02	01234567	0	0				00	X axis positive moving 1234567
2	oUt					01	1	00	Open output O1
3	dLY	055900						00	Delay 55900ms execute the next instruction

4	G03	-0234567	0	0		00	0	00	X axis move reverse. When input IO low level effective, finish this move advance.
5	dLY	000000						00	Pause, waiting for start
6	G01	-0000888	0					00	X axis move to coordinate -0000888
7	oUt					01	0	00	Close output O1
8	End							00	Program end

【Example three】 Request: There are a object, Move forward 100(as reference point) from zero position with 2.9KHz speed; Then check input bit I4, if input effective, motor return to zero position with same speed, if I4 input ineffective, motor move forward 1000 with 15KHz speed, then return to reference point with 35KHz speed; If this moment I4 effective, return to zero position, or else go on loop according to the first mode, analogy in turn.

Program list:

No.	Instruction	1	2	3	4	5	6	7	Description
0	SPd	002900						00	Set the follow moving speed is 2.9KHz
1	G02	00000100	0	0				00	X axis positive move to 00000100, this point as reference point
2	Jb It				02	04	0	01	When input I4 low effective, program jump to this line 02
3	SPd	015000						00	Set the follow move speed is 15KHz
4	G02	00010000	0	0				00	X axis positive move 10000
5	SPd	035000						00	Set follow move speed is 35KHz
6	G01	00000100	0					00	X axis return to reference point 00000100
7	JNP					01		00	Program jump to this line 01
8	SPd	002900						02	Set the follow speed is 2.9KHz, This line 02 as input jump entrance
9	G01	00000000	0					00	X axis return to zero position
10	End							00	Program end

【Example four】 As more advanced auto cutter controller.

System configure: MSC-M30 controller, two-phase step motor 130BYG250A,driver SH-2H130MH,AC220V isolated transformer(can select). Wheel girth 200mm. Operate

panel except for MSC-M30, also include: effective/ineffective key(locked-self key, it can defined as input IO): When this key pressed you can start motor ; if this key loose, motor can't move even it have photoelectric signal.

Running request: Example for length of paper, Every start ,high speed moving 500mm. In addition ask cutting paper 5 ten thousand, then open output signal o4 ten times(outside connect alarming equipment), this moment the counter need clear and restart.

Design analysis: Make driver working under 20 subdivision ,step angle is 0.09° , pulse equivalent:20 CP pulse/mm

Parameter set:(Enter Parameter set state) SF=1000, RL=4, C0=0, DM=1 (after controller electrify, display inside counter 0 path mode). These parameters can adjust according to concrete machine.

Program list:

No.	Instruction	1	2	3	4	5	6	7	Description
0	Jbt				02	00	1	00	When input IO high effective, program jump to this line 02, and quit
1	JLn	050000			01	00	0	00	When the value of interior counter arrive at 5 ten thousand, program jump to this line 01
2	SPd	028000						00	Set follow moving speed is 28KHz
3	G02	00010000	0	0				00	X axis move positive 10000, motor moving 500mm
4	Cnt	000001				00		00	Make the inside counter 0 path add 1
5	End							02	Program finish
6	oUt					04	1	01	Open output O4
7	dLY	000200						00	Delay 0.2 s
8	oUt					04	0	00	Close output O4
9	LoP	000009			01			00	The loop entrance is this line 01, repeat 10 times
10	SCn	000000				00		00	Make inside counter 0 path clear
11	End							00	Program end

【Example five】 As more advanced auto bag machine controller

System configure: MSC-M30 controller, two-phase step motor 130BYG250A,driver SH-2H130MH,AC220V isolated transformer(can select). Wheel girth 200mm. Operate panel except for MSC-M30, also include: **Error! Not found source.** effective/ineffective key(locked-self key, it can defined as input IO): When this key pressed you can start motor ; if this key loose, motor can't move even it have photoelectric signal. **Error! Not found source.** Print/Length select

key(locked-self key, it can defined as input I1): Pressed is Print mode, loose is Length mode.

Running request: Example for bag length 500mm, Every start, high speed moving 500mm. Under print mode, every start, first high speed moving 480mm, then change to low speed to find color code, if find color code it will stop at once. If running 510mm ,don't find color code, then consider it have trouble, need stop at once to open output signal 05 (outside connect short sound alarming equipment 100 times). In addition ask make bag 5 ten thousand, open output signal 04(outside connect long sound alarming equipment ten times),this moment the counter need clear and restart.

Design analysis: Make driver working under 20 subdivision ,step angle is 0.09° , pulse equivalent:20 CP pulse/mm.

Parameter set:(Enter Parameter set state) SF=1000, RL=4, C0=0, DM=1 (after controller electrify, display inside counter 0 path mode). These parameters can adjust according to concrete machine.

Program list:

No.	Instruction	1	2	3	4	5	6	7	Description
0	Jbt				02	00	1	00	When input I0high effective, program jump to this line 02, quit
1	Jcn	050000			01	00	0	00	When inside counter value reach to 5 ten thousand, program jump to this line 01
2	SPd	028000						00	Set the follow move speed is 28KHz
3	Jbt				03	01	0	00	When input I1 low effective, program jump to this line 03, printing.
4	G02	00010000	0	0				00	X axis positive move 10000 steps, motor move 500mm
5	Cnt	000001				00		04	Make inside counter 0 path add 1
6	End							02	Program end
7	out					04	1	01	Open output O4
8	dLY	000200						00	Delay 0.2s
9	out					04	0	00	Close output O4
10	LoP	000009			01			00	Loop entrance is this line 01,loop ten (9+1)times
11	Scn	000000				00		00	Make inside counter 0 path clear
12	End							00	Program end
13	G02	00009600	0	0				03	X axis positive move 9600 steps, motor move 480mm

14	SPd	001000						00	Set the follow move speed is 1KHz
15	Jbt				04	02	0	00	When input I2 low effective, program jump to this line 04, quit.
16	G02	00000600	0	0				00	X axis positive move 600 steps, motor running 30mm
17	oUt					05	1	05	Open output O5, and set loop entrance is 05.
18	dLY	000200						00	Delay 0.2s
19	oUt					05	0	00	Close output O5
20	LoP	000099			05			00	Loop entrance is this line 05, loop 100(99+1) times.
21	End							00	Program end

4. Logic program edit and instruction explain

4.1 Summarize

Logic operation and control function is one of main functions of MSC-M30. It can implement logic operation for 8 inputs and inside variable. Operation function include: Multi-input And, multi-input Or, single input Not, State output, Variable set etc. It can implement Input/Output logic (PLC) control by specifically control instruction.

4.1.1 Enter and quit

Program edit function can be used to read, amend, browse the saved control program etc. Enter Program edit state: Under *LE--Ed*, press “Enter” key to enter. After program edit finish, press “Quit” to quit(Before edit new program, use “Program clear” to clear program).

4.1.2 Program line

This controller can edit at most 20 logic program, don't need input tab, read according to auto sequence. When enter a new program, logic program instruction is *End* that system in default, Twinkle, you can roll and select instruction. According to hint , you can input all data that need.

During editing, you can insert or delete some line, it will automatically renew the order of program. Select some line program (the aim twinkling), this moment press “Delete” to delete, instruction will automatically change to next ; Press “Enter” key (this moment ,it is insert key), it will insert a new program line, “Left”, “Right” switch between variable and program, “Up” , “Down” key to roll select logic instruction. According to hint, input all data that need ,then insert this line.

4.1.3 Program format

Divide two line display: The first tube display aim variable(0 to 9, inside variable) or output path no(0 to 5,Move instruction), The third bit is hint symbol “=”, 5-8 is logic instruction; The second line display data that need editing: For “And” , “Or” instruction, the eight tubes all display 0 or 1, corresponding 8 inputs or 8 inside variable, hint input, When input 1, join operation, if input 0,don't join operation; For other instruction, First tube display 0 to hint input source variable(inside

from 0 to 7 or outside input from 0 to 7) or set instruction's install value(set variable). The last instruction is "End", if no this program, it can't execute control program rightly.

4.2 Instruction sort and explain

According to operation object is different, this system logic instruction have 10, it can divide into three sorts:

Outside input instruction, include Andi, ori, noti Three;
 Inside variable instruction, include And, or, not, Mov, SET Five
 Other, include out and End Two

No	Name	Instruction Form	Aim	Source	Description
1	Operate end	End	0	0	Logic program end. When logic program run to this instruction end, and restart running logic program.
2	Input "And"	And,	inside	Outside input	Make the all input that selected operate "And", and save the result to inside variable. Only one select input, the result same to input state.
3	Variable "And"	And	inside	inside	Make the all variable that selected operate "And" (corresponding code set 1), and save the result to inside variable. Only one select input, the result same to this variable.
4	Input "Or"	or,	inside	Outside input	Make the all input that selected operate "Or"(corresponding code set 1), and save the result to inside variable. Only one select input, the result same to this variable.
5	Variable "Or"	or	inside	inside	Make the all variable that selected operate "Or" (corresponding code set 1), and save the result to inside variable. Only one select input, the result same to this variable.
6	Input "Not"	not,	inside	Outside input	Select input bit state reverse and save the result to the gave inside variable.
7	Variable "Not"	not	inside	inside	Reverse the selected variable state and save the result to the gave inside variable.
8	Move Variable	Mov	inside	inside	Move the state of source variable to aim variable, source variable don't change.
9	Set variable	SET	inside	Digital 0-9	Set inside variable state is 1 or 0. Only is the first 8 variable(0-7).
10	Variable output	out	Output path	inside	Output the state of selected inside variable to output path.

4.3 Program operation

4.3.1 Input/Amend program

After clear program area, enter "Logic program edit". Display instruction is End, The twinkle cursor stay aim variable no(display 0). According program, input the current instruction's "Aim variable No". After input finish, The twinkle cursor move automatically to instruction display, Use "Up" and "Down" key to select right instruction, Use "Left" and "Right" move twinkle cursor, input right data or amend aim variable. After instruction input finish, use "Down" key to enter next instruction input, "Up" key to browse previous instruction.

Description: The No 0-7 of outside input variable can join the logic operation, from right to left in turn display; Inside variable's No is 0-9, The No 0-7 of inside variable can join logic operation, from right to left in turn display, the No 8 and 9 can save the operation result to

output ,but can not join operation directly.

4.3.2 View program

After enter “Logic program edit”, the display show the first instruction (include all data of this instruction), press “Down” key to enter next until *End* instruction, Press “Up” key to browse the previous.

4.3.3 Program line insert

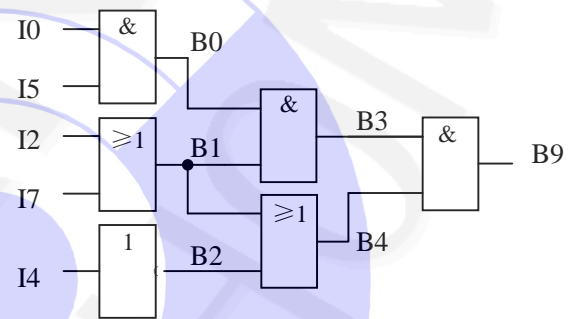
When the cursor stay in Instruction code area, press “Enter” key, the twinkle instruction change to *And* , this moment , left and right key to move cursor to select amend data, then finish new program line’s insert.

4.3.4 Program line delete

When cursor lie in Instruction code area, press “Delete” key, this instruction be deleted, this moment display the next instruction.

4.4 Application example

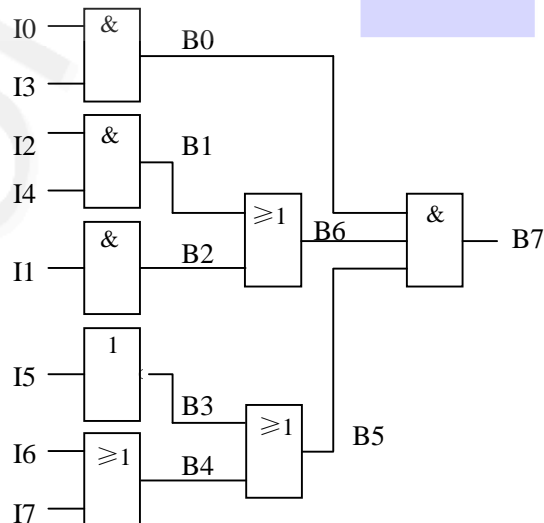
【Example one】 : As shown in figure: I0, I2, I4, I5,I7 all outside input, in order to program, select inside variable B0, B1, B2, B3 B4, B9. Save the logic result between I0 and I5 to B0; I2 and I7’s result save to B1; Reverse I4, result save to B2; B0 and B1’s logic And result save to B3; B1 and B2’s result that logic Or save to B4; Logic Or of B3 and B4 get the last result ,save to B9 and output to O5.



Logic program edit as follow:

0	=	ANDI	00100001	The result of input port I0 and I5 move to B0
1	=	OR I	10000100	The result of input port I2 and I7 move to B1
2	=	NOT I	4	The result of input port I5 move to B2
3	=	AND	00000011	The result of logic inside variable B0 and B1 move to B3
4	=	OR	00000110	The result of logic inside variable B1 and B2 move to B4
9	=	AND	00011000	The result of logic inside variable B3 and B4 move to B9
5	=	OUT	9	The state of B9 move to output O5
0	=	END	0	Program end

【Example two】:As shown in figure: I0~I7 all are outside input, in order to program, select inside variable B0 ~ B7. Save the result of I0 and I3 to inside variable B0; Save the result of I2 and I4 to B1; I1 transfer to inside variable B2; Reverse I5, result save to B3; I6 and I7’s logic result save to B4; B1 and B2’s logic Or result save to B6; B3 and B4’s logic Or result save to B5; B0, B5 and B6’s logic Or get the last result, save to B7, and output O4.



Logic program edit as follow:

0	=	ANDI	00001001	The result of input I0 and I3 move to B0
1	=	ANDI	00010100	The result of input I2 and I4 move to B1
2	=	ANDI	00000010	The result of input I1 move to logic variable B2
3	=	NOTI	5	The result of input I5 move to logic variable B3
4	=	ORI	11000000	The result of input I6 and I7 move to logic variable B4
5	=	OR	00011000	The result of logic inside variable B3 and B4 move to B5
6	=	OR	00000110	The result of logic inside variable B1 and B2 move to B6
7	=	AND	01100001	The result of logic inside variable B0, B5 and B6 move to B7
4	=	OUT	7	The state of logic inside variable B7 move to O4
0	=	END	0	Program end

5. Error code description

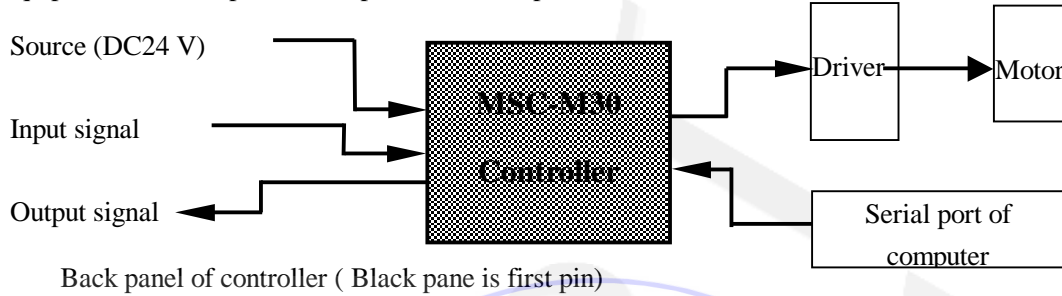
During program running, system will check it. If there are problem, it will stop running, and gave error code in order to find cause and repair. The code and deal with means as follow:

Display form	Content	Instruction	Deal with means
Ettot_1	Code error	Nothing	Check program line-by-line (or PC clear program, then input right program).
Ettot_2	Path select error	Go3, SCn, Cnt, JcN, Jbt, int, oUt	Check corresponding instruction, Change path select.
Ettot_3	Jump and not find tab	LoP, JcN, Jbt, JMp, int	View and right set the tab of jump entrance
Ettot_4	Jump and don't give tab	LoP, JcN, Jbt, JMp, int	View entrance tab of these instructions
Ettot_5	No finish symbol END	End	View program, Add finish instruction in program exit.
Ettot_6	Speed scope error	SPd	Gave moving speed again, it should in the speed scope(1-99999Hz).

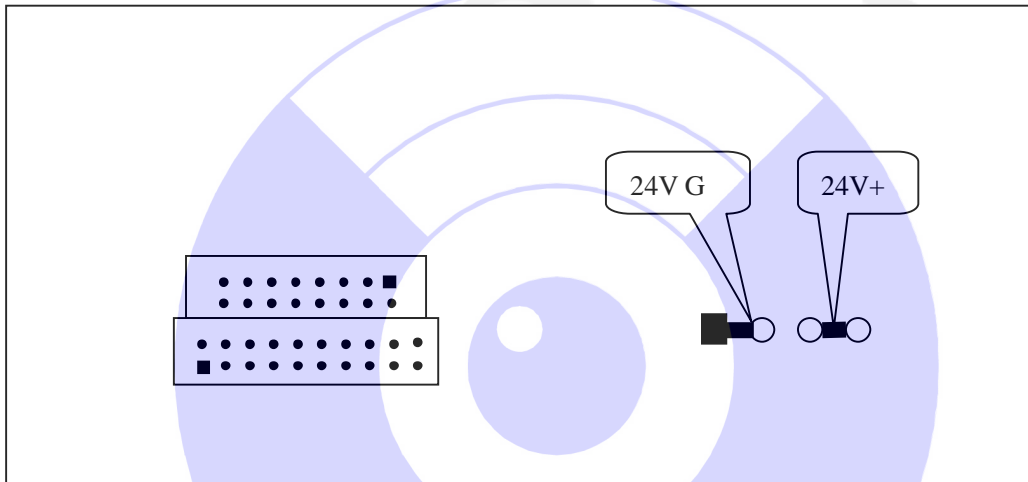
6. System connection and installation

6.1 System connection

System connect include the connection of between system and driver, between input/output equipments and computer serial port. Sketch map as shown below:



Back panel of controller (Black pane is first pin)



6.1.1 System pins list

Input socket define (the top socket 16 pins)

Define	I0	I2	I4	I6	I8	I10	Null	24V ground
Pin	1	3	5	7	9	11	13	15
Pin	2	4	6	8	10	12	14	16
Define	I1	I3	I5	I7	I9	I11	Null	24V ground

Output and serial port define (the under parts socket 20 pins).

Define	Xdir-	Xcp-	Ydir-	Ycp-	Vcc	Gnd	232Rxd	Out0	Out2	Out4
Pin	1	3	5	7	9	11	13	15	17	19
Pin	2	4	6	8	10	12	14	16	18	20
Define	Xdir+	Xcp+	Ydir+	Ycp+	Vcc	Gnd	232Txd	Out1	Out3	Out5

Source define (Green socket)

Pin	1	2	3	4
Define	24V ground	24V ground	+24V	+24V

MSC-M30 Transfer board define

J3 define:

Pin	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Define	IN0	IN1	IN2	IN3	IN4	IN5	IN6	IN7	IN8	IN9	IN10	IN11	TXD	RXD	GND

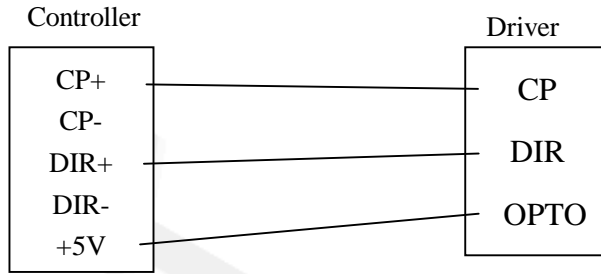
J4 define:

Pin	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Define	Xdir-	Xdir+	Xcp-	Xcp+	Ydir-	Ydir+	Ycp-	Ycp+	5V	OUT0	OUT1	OUT2	OUT3	OUT4	OUT5

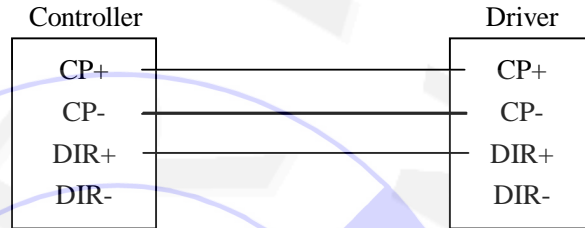
6.1.2 Motor connect with driver

There are two instances:

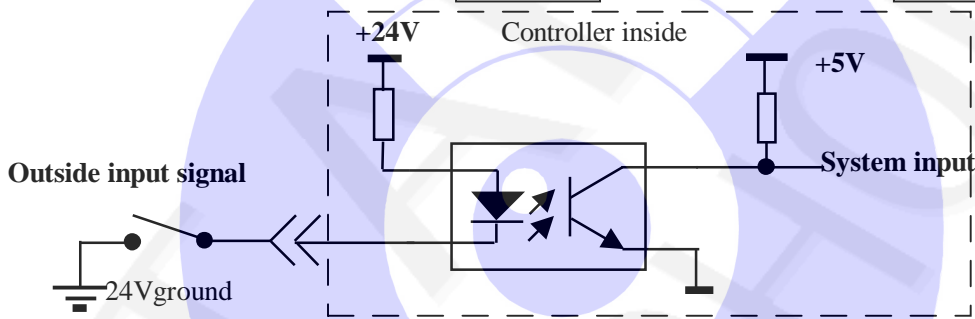
When the driver connect is not differential, according to right picture connect.



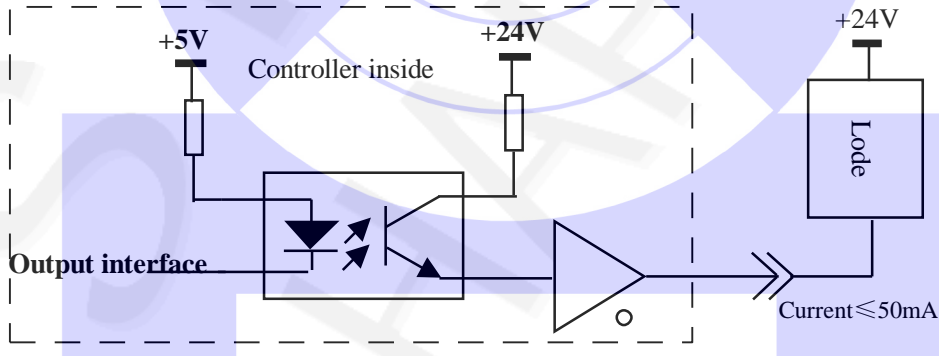
When the driver connect is differential According to right picture connect.



6.1.3 Input theory

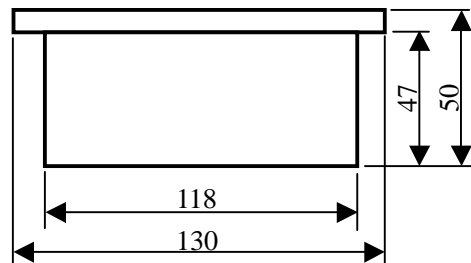
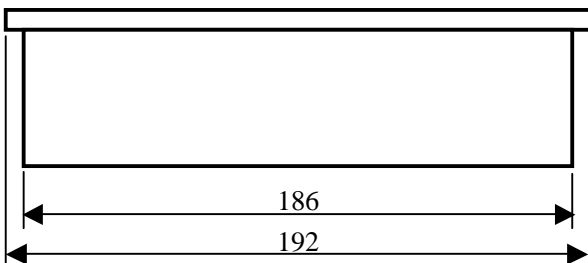


6.1.4 Output theory



6.2 Install size

The shell of this controller is embedded, if you need embed in control box, only need processing a hole (186×118), after embedded controller, fixed it inside. Notice that you should left space (over 30mm) in back of controller.



APPENDIX ONE Instruction format

The instructions are composed of 7 byte(C0-C6), list explain as shown below:

Instruction	C0	C1	C2	C3	C4	C5			C6	
						D7	D6	D5-D0	D7	D6-D0
G01	00H	Coordinate (Hex, low byte before) , Symbol				Axis select(0/1)	0	000000	0	Tab
G02	01H	Coordinate (Hex, low byte before) , Symbol				Axis select(0/1)	Direction select(0/1)	000000	0	Tab
G03	02H	Coordinate (Hex, low byte before) , Symbol				Axis select(0/1)	Direction select(0/1)	Path select	Input level	Tab
G04	03H	Coordinate (Hex, low byte before) , Symbol				Axis select(0/1)	0	000000	0	Tab
dLY	10H	Delay(unit: ms)				0	0	000000	0	Tab
SPd	11H	Speed value, No symbol		00H		0	0	000000	0	Tab
LoP	12H	Loop times		Aim tab		0	0	000000	0	Tab
SCn	20H	Count		00H		0	0	Count path select	0	Tab
Cnt	21H	Count		00H		0	0	Count path select	0	Tab
JCn	30H	Count		Aim tab		0	0	Count path select	Jump mode	Tab
Jbt	31H	00H	00H	00H	Aim tab	0	0	Input path select	Input level	Tab
JNP	32H	00H	00H	00H	Aim tab	0	0	000000	0	Tab
int	33H	00H	00H	00H	Aim tab	0	0	Path select	Input level	Tab
out	40H	00H	00H	00H	00H	0	0	Path select	Output level	Tab
RET	41H	00H	00H	00H	00H	0	Return mode	000000	0	Tab
End	42H	00H	00H	00H	00H	0	0	000000	0	Tab
Instruction parameter hint	0	1			4	2	3	5	6	7

Content

1. System Features	1
2. System function.....	2
2.1 Parameter set.....	3
2.1.1 Amend means	4
2.1.2 Display mode instruction.....	5
2.1.3 Switch set.....	5
2.2 Program manage	5
2.3 Logic program	6
2.4 Auto	6
2.5 Manual operation.....	6
2.6 Jog operation.....	7
2.7 Coordinate clear.....	7
2.8 System check-self.....	7
2.9 Accept program.....	7
3. Program edit and instructions explain.....	9
3.1 Summarize	9
3.1.1 Enter and quit	9
3.1.2 Program line	9
3.1.3 Instruction format	9
3.1.4 Program inquire explain.....	9
3.1.5 Program edit.....	10
3.2 Instruction sort and format explain.....	10
3.2.1 Instruction sort.....	10
3.2.2 Instruction parameter	10
3.3 Program save	11
3.4 Instruction explain	11
3.5 Application example	13
4. Logic program edit and instruction explain	17
4.1 Summarize	17
4.1.1 Enter and quit	17
4.1.2 Program line	17
4.1.3 Program format.....	17
4.2 Instruction sort and explain.....	18
4.3 Program operation	18
4.3.1 Input/Amend program.....	18
4.3.2 View program.....	19
4.3.3 Program line insert.....	19
4.3.4 Program line delete	19
4.4 Application example	19
5. Error code description.....	20
6. System connection and installation	21
6.1 System connection.....	21
6.1.1 System pins list.....	21
6.1.2 Motor connect with driver	22
6.1.3 Input theory	22
6.1.4 Output theory.....	22
6.2 Install size	22
APPENDIX ONE Instruction format	23