# Chapter 13 The NC Positioning Control of FBs-PLC

People use ordinary motor to exercise positioning control in early stage; since the speed and precision demand was not so high then, it was enough to fulfill the demand. As the increasing of mechanical operation speed for the efficiency purpose, finished product quality standard, and precision demands are getting higher, the stopping position control of motor is no more what the ordinary motor is capable to do. The best solution for this problem is to adopt NC positioning controller which incorporate with stepping or servo motor to do the position control. In the past, the extremely high cost limited the prevailing of its usage; however, the technology advance and cost decreasing, which made the pricing affordable, had helped to increase the prevailing of usage gradually. To cope with this trend, the FBs-PLC integrated into its internal SoC chip the special NC positioning controller that is available on the market, therefore makes it free from the bothersome data transaction and linking procedure between PLC and special NC positioning controller. Furthermore, it greatly lowered the entire gadget cost hence provides the user the solution for a good bargain, high quality, simple, and convenient integrated NC positioning control with PLC.

## 13.1 The methods of NC positioning

The methods for controlling interface of PLC and stepping or servo driver are as follows:

- Giving command by way of digital I/O: Easy to use but less dexterity in application.
- Giving command by way of analogue output: Better dexterity in controlling reaction but it is with a higher cost and easy to be interfered by noise.
- Giving command by way of communication: There is no standard for communication protocol and it is confined in communication reaction thus constitutes a bottleneck for application.
- Giving command by way of high speed pulse: The cost is low and is easy to precisely controlled.

Of these methods, controlling stepping or servo driver with high speed pulse is more frequently used method. The main unit of PLC contains multi-axis high speed pulse output and hardware high speed counter, and it can provide easy using, designing for positioning program editing. Therefore it makes the related application even more convenient and comfortable.

Following two kinds are frequently used NC server system that constituted by PLC associates with servo drivers:

- Semi closed loop control

  The PLC is responsible for sending high speed pulse command to servo driver. The motion detector installed on servo motor will forward directly to server driver, closed loop reaches only to server driver and servo motor. The superior point is that the control is simple and the precision is satisfactory (which is suitable for most of the applications). The defect is that it can't fully reflect the actual shift amount after the transmission element; furthermore, the element being consumed, become aging, or has defect will not be able to be compensated nor checked to verify.

- Closed loop control

  The PLC is responsible for sending high speed pulse command to servo driver. In addition to that the shift detection signal installed on servo motor which will be forwarded directly to servo driver, the attached shifting detector installed after the transmission element can fully reflect the actual shift amount and forward it to the high speed counter that PLC contains. So as to make the control becomes more delicate, and help to avoid the defect of above mentioned semi closed loop.

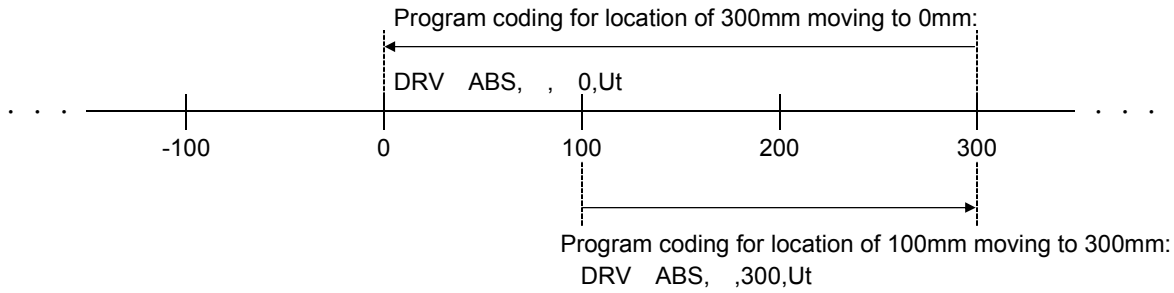
## 13.2 Absolute coordinate and relative coordinate

The designation of moving distance can be assigned by absolute location (absolute coordinate positioning), or assigned by relative distance (relative coordinate positioning). And the DRV instruction is used to drive motor.

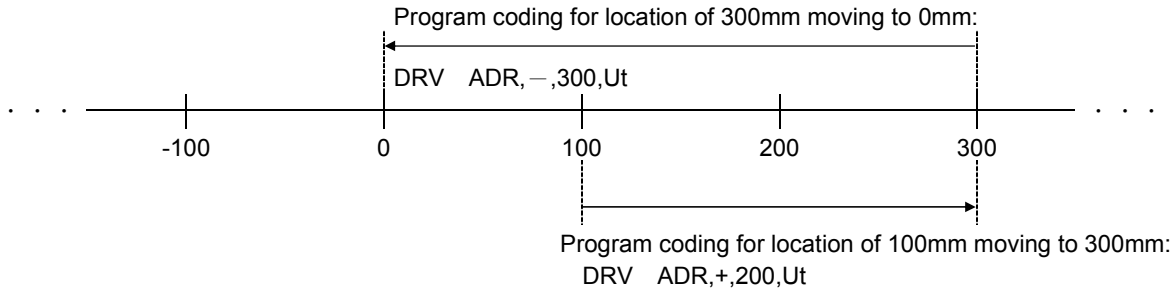
While marking the moving distance with absolute coordinate,
if it is located at 100mm at the present, for moving to 300 mm, the positioning instruction is : DRV ABS, ,300, Ut
if it is located at 300mm at the present, for moving to 0mm, the positioning instruction is : DRV ABS, , 0, Ut.

While marking the moving distance with relative coordinate,
if it is located at 100mm at the present, for moving to 300 mm, the positioning instruction is : DRV ADR, +, 200, Ut.
if it is located at 300mm at the present, for moving to 0mm, the positioning instruction is : DRV ADR, –, 300, Ut.

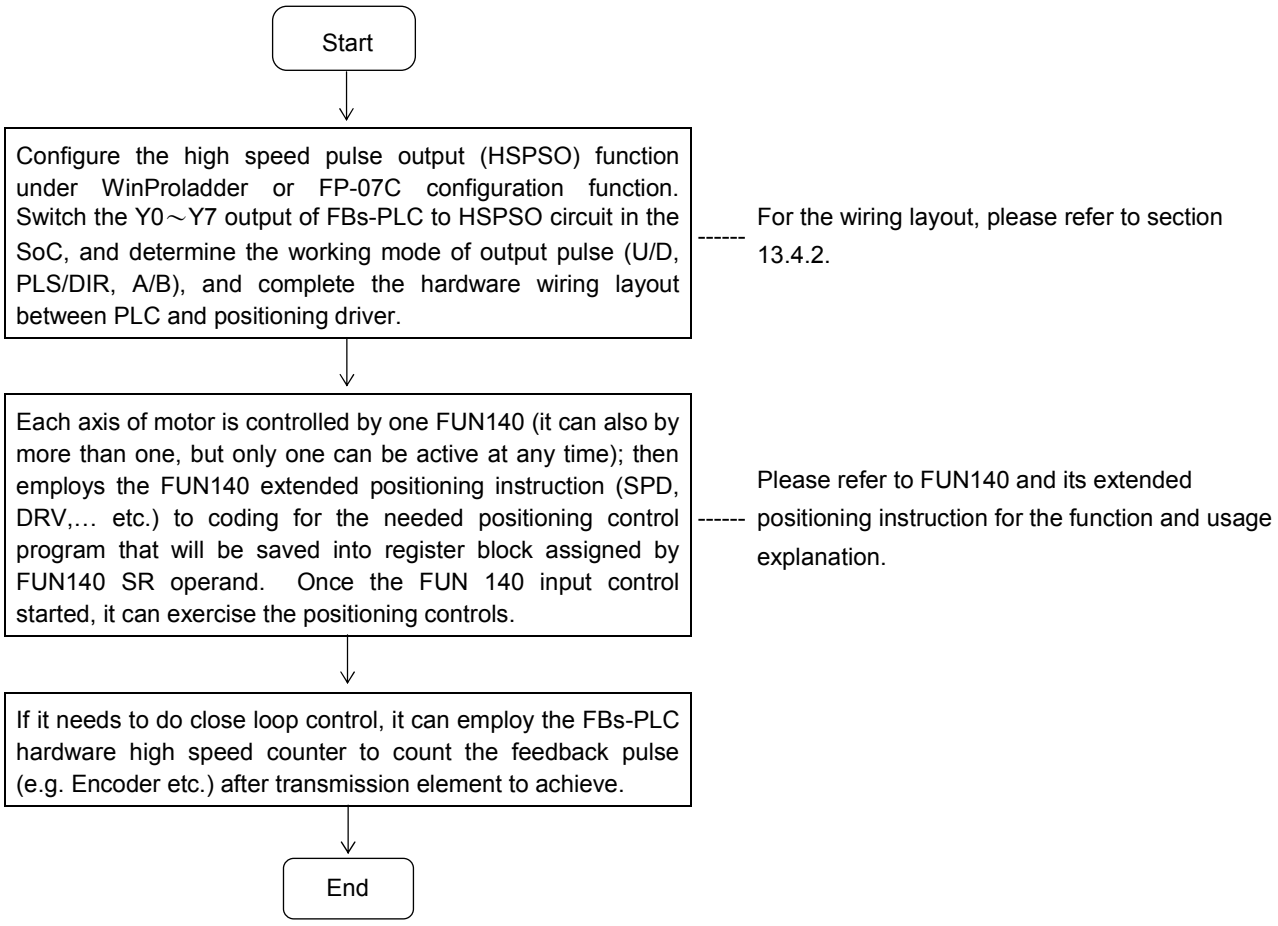
- Absolute coordinate labeling




- Relative coordinate labeling




## 13.3 Procedures of using FBs-PLC positioning control



## 13.4 Explanation for the positioning control hardware of FBs-PLC

### 13.4.1 Structure of output circuit of HSPSO

According to different main unit, it provides different frequency of output pulse, it includes 120KHz (High speed ) /20KHz (Medium speed) of single ended transistor output model (FBs-xxMCT), and high speed differential output model (FBs-xxMN) which can reach 920KHz (for single phase), two series of models.

High speed pulse output circuit share to use the Y0～Y7 exterior output of FBs-PLC. While it is not yet using the HSPSO function (haven't configured the PSO function under configuration function), the Y0～Y7 exterior output of FBs-PLC is corresponding to the Y0～Y7 status of internal output relay. When the HSPSO has been configured, the Y0 ～Y7 exterior output will switch directly to HSPSO output circuit within SoC, which has no relation with Y0～Y7 relay inside PLC.

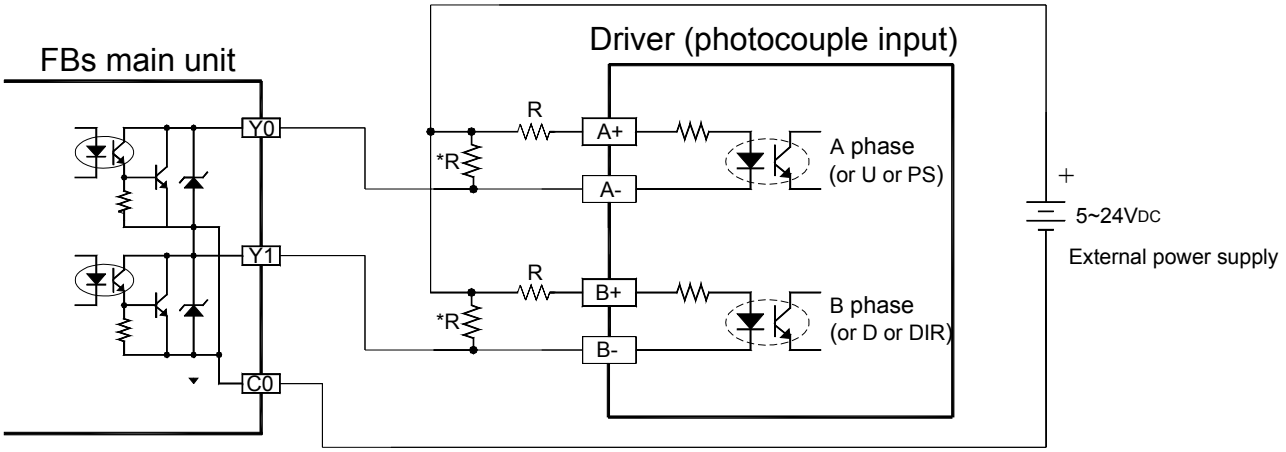
The following is the detailed signals list for respective axis output of main unit and the selectable output modes:

| Axis No. | Exterior output | Output modes | | | |
|---|---|---|---|---|---|
| | | U/D output | P/R output | A/B output | Single PLS output |
| PSO0 | Y0 , Y1 | Y0=U , Y1=D | Y0=P , Y1=R | Y0=A , Y1=B | Y0=PLS |
| PSO1 | Y2 , Y3 | Y2=U , Y3=D | Y2=P , Y3=R | Y2=A , Y3=B | Y2=PLS |
| PSO2 | Y4 , Y5 | Y4=U , Y5=D | Y4=P , Y5=R | Y4=A , Y5=B | Y4=PLS |
| PSO3 | Y6 , Y7 | Y6=U , Y7=D | Y6=P , Y7=R | Y6=A , Y7=B | Y6=PLS |

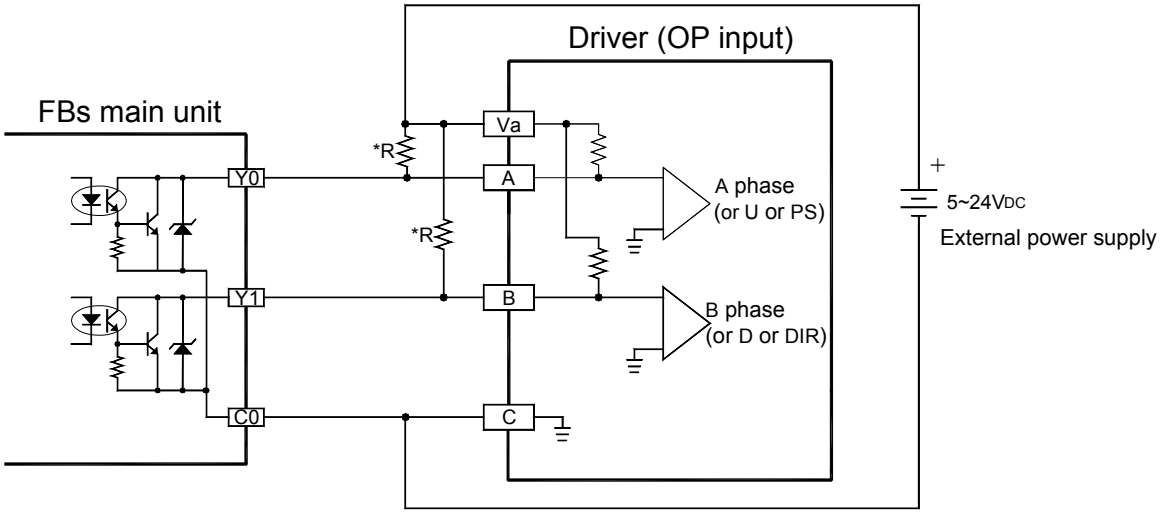
### 13.4.2 Hardware wiring layout for FBs-PLC positioning control

Take the 0th axis (PSO0) of FBs-XXMCT and FBs-XXMN main unit for example, it is illustrated with diagrams as follows; the others are the same.

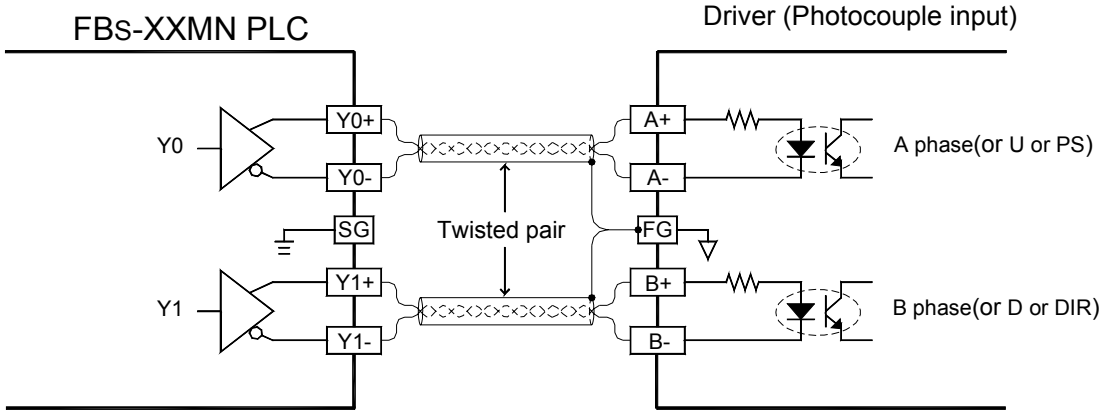
#### A, FBs-XXMCT single ended output wiring

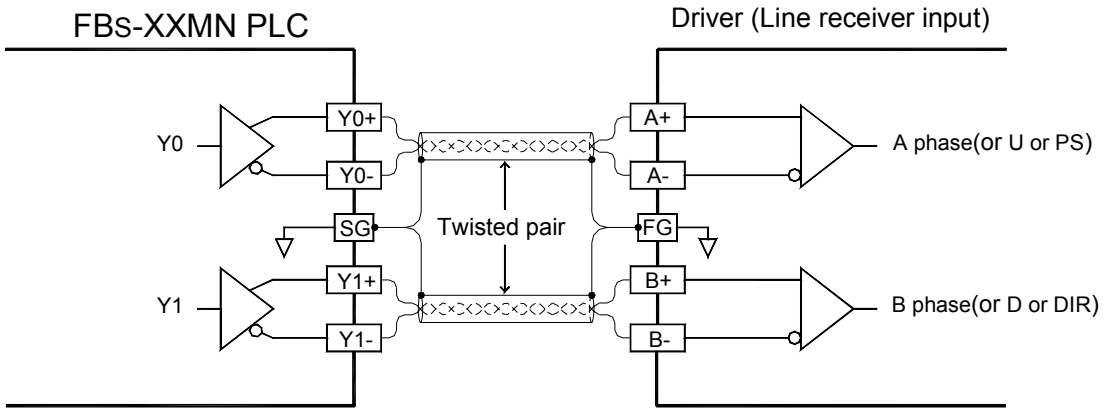



＊ Please refer to Hardware manual H7-6 for the usage of speed-up resistor "R".




B、FBS-XXMN differential output wiring




（For line receiver input, it must make PLC connect to FG of driver to eliminate common mode voltage）

Configuration of HSPSO with WinProladder

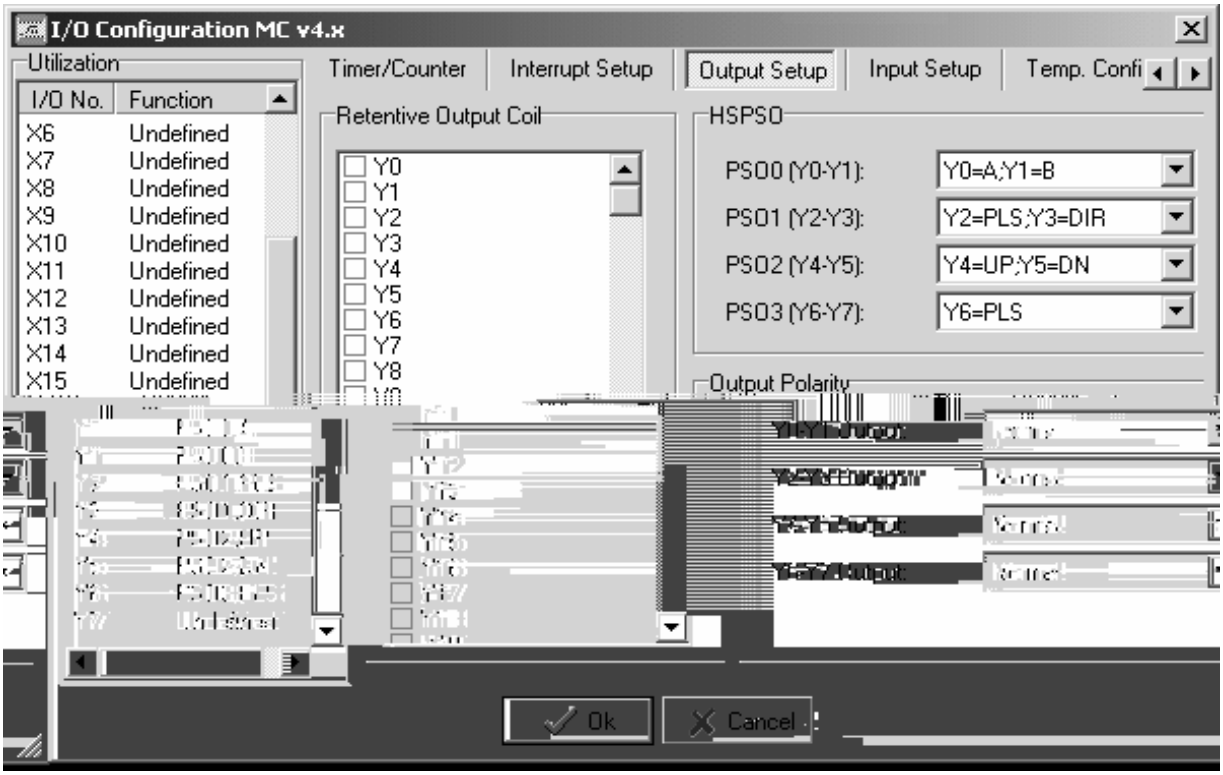
Click the “I/O Configuration” Item which in project windows：

Project name

System Configuration

I/O Configuration → select “Output Setup”

When “Output Setup” windows appear, then you can configure the Output type：




## 13.5 The explanation for the position control function of FBs-PLC

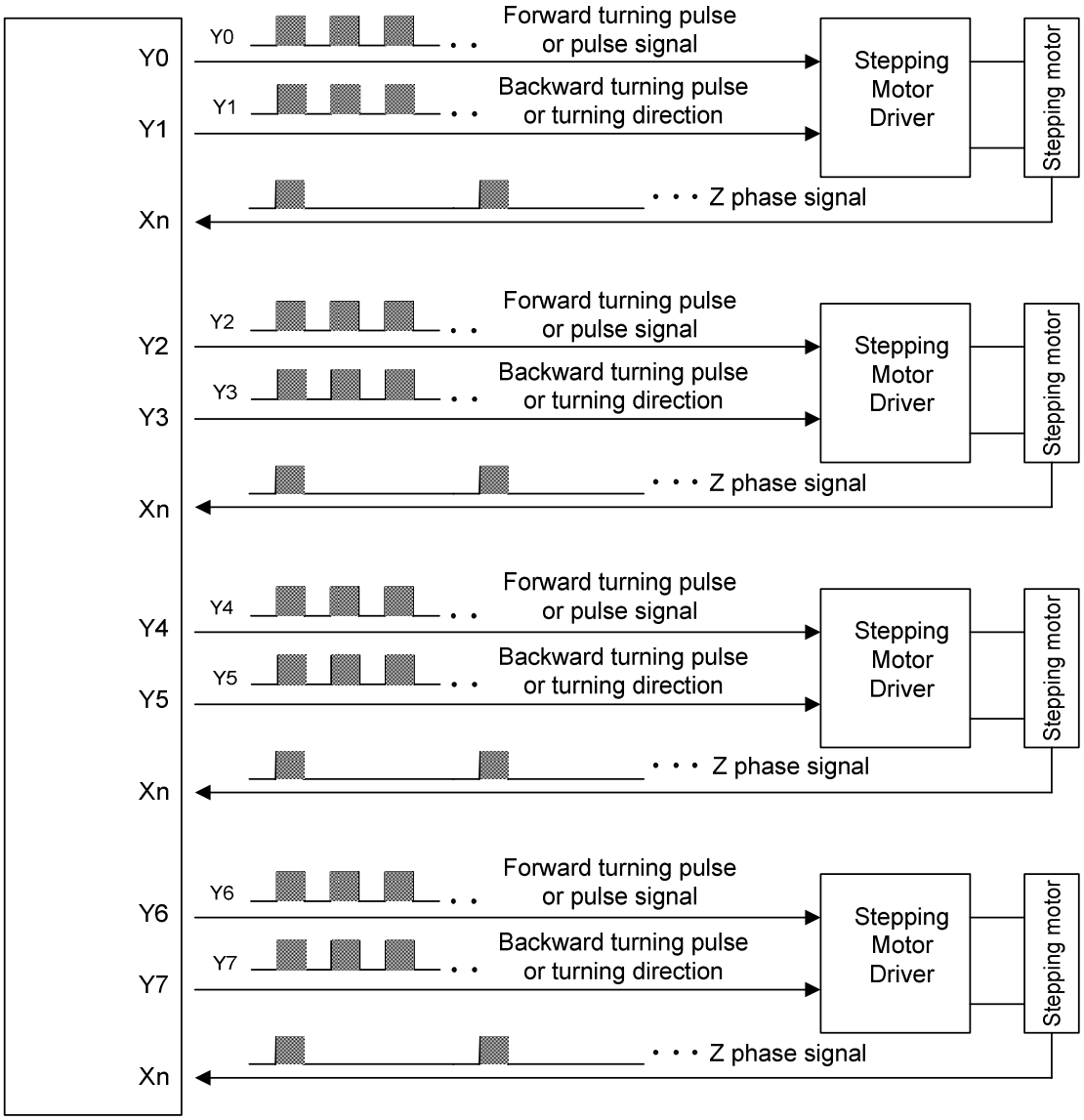
The position control function of FBs-PLC incorporates the dedicated NC position controller, which is available in the market, into the PLC. This makes the PLC and NC controller be able to share the same data block without the demand of complicated works like data exchange and synchronized controlling between these two systems. And it can still use the usual NC positioning control instruction (e.g. SPD , DRV,… etc.).

One main unit can control up to 4 axis of their position control, and can drive multi axis simultaneously. However, it provides point to point positioning and speed control, but also it provides the linear interpolation function. When the system is applying for more than 4 axis, it can also employ CPU LINK function of FBs-PLC to attain control over more positioning actions.

The NC position control instruction for FBs-XXMCT、FBs-XXMN main units are identical to each other. The difference is only on the different circuit output, as previously revealed. Hereby we assume that FBs-XXMCT main unit is used in the control of stepping motor with lower speed, and FBs-XXMN main unit is used in high speed servo motor control. Consequently, we illustrate only with the connecting diagram of FBs-XXMCT main unit that driving stepping motor and the diagram of FBs-XXMN main unit that driving servo motor. Of course we can also use FBs-XXMCT main unit to drive servo motor or use FBs-XXMN main unit to drive stepping motor instead, they can still work perfectly, as long as its circuit structure (single ended or differential) and frequency can match.

## 13.5.1 Interface of stepping motor

FBs-XXMCT main unit




- Stepping motor is designed to receive input pulse to attain to the control of desired angle or distance, therefore the turning angle and the input pulse count has a positive correlation ship, and the turning speed also depends on the input pulse frequency.

N (RPM) = 60 × f / n

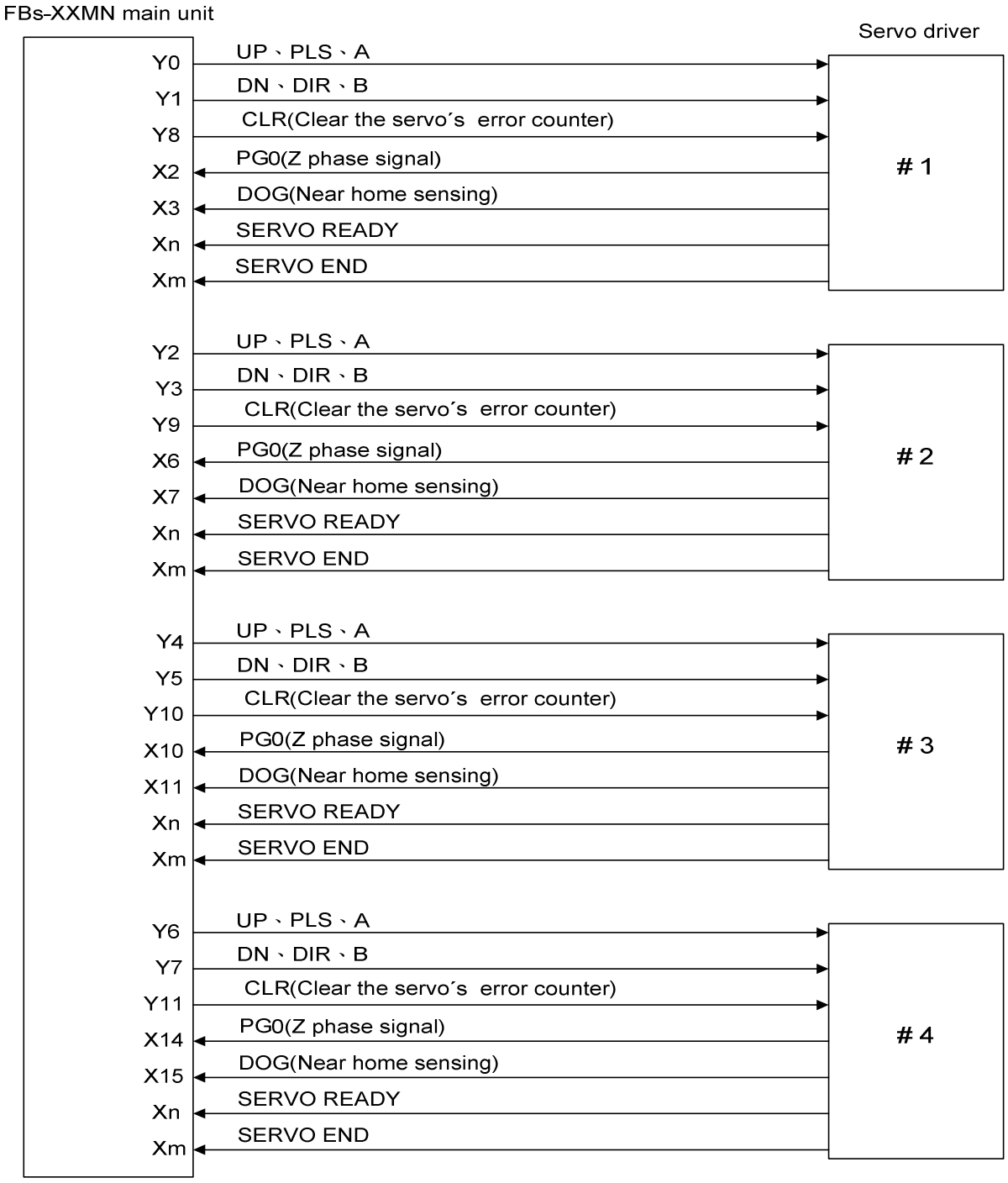
N : Revolving speed of motor (RPM)
f : Pulse frequency (Ps/Sec)
n : Pulse counts for motor to turn for a revolution (Ps/ Rev).

n = 360 / θs

θs : Angle (Deg)

| Phase | Basic pulse angle | FULL | | HALF | |
|---|---|---|---|---|---|
| | | Pulse angle | Pulse counts for turning one revolution | Pulse angle | Pulse counts for turning one revolution |
| 5 phase | 0.36° | 0.36° | 1000 | 0.18° | 2000 |
| | 0.72° | 0.72° | 500 | 0.36° | 1000 |
| 4 phase | 0.90° | 0.90° | 400 | 0.45° | 800 |
| 2 phase | 1.80° | 1.80° | 200 | 0.90° | 400 |

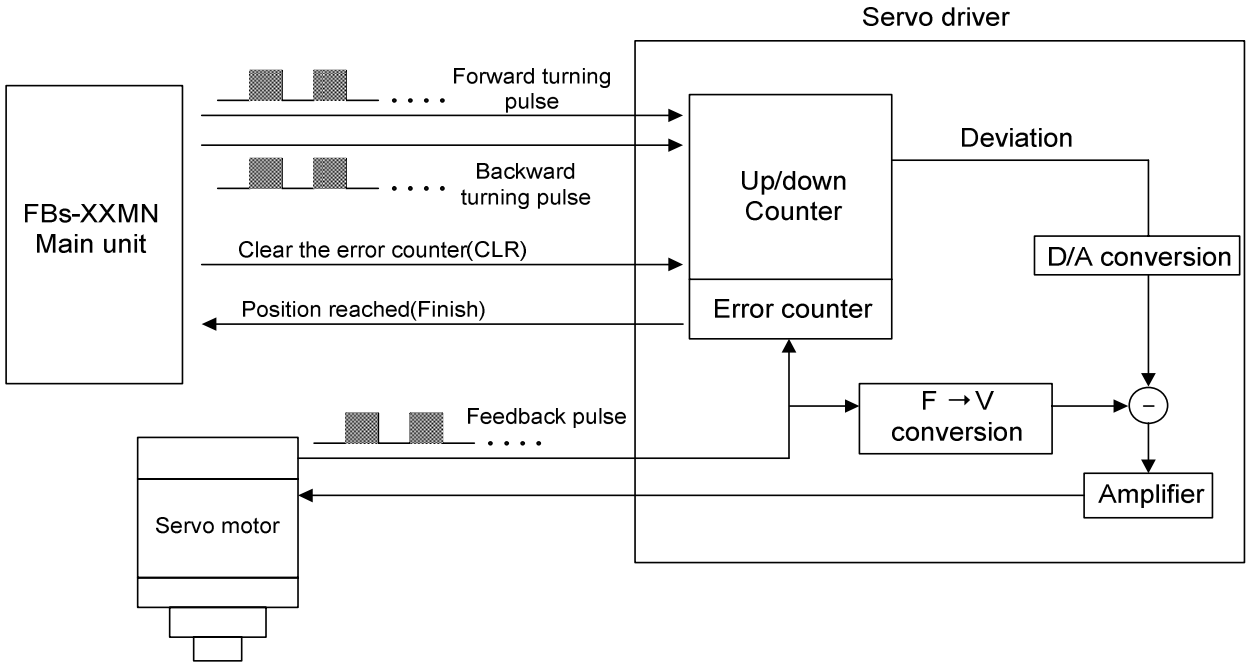
## 13.5.2 Interface of servo motor




※ Except that the Y0～Y7 of above diagram are for dedicated purpose, Y8～Y11 and respective inputs can be adjusted for using according to demand.

※ The left over travel, right over travel limit switchs for safity detection also need to be connected to PLC to assure proper operation.

### 13.5.3 Working diagram illustration for servo motor




- The Encoder of servo motor feedback the shifting detection signal to servo driver. The driver gets the pulse frequency, and pulse count of input signal (pulse command), as well as the frequency and pulse count of feedback signal processed with internal error counter and frequency to voltage conversion circuit, and acquired the pulse and turning speed deviations. Using these operations to control the servo motor, so as to obtain a high speed, precise speed and positional closed-loop processing system.

- The revolving speed of servo motor depends on the pulse frequency of input signal; the turning stroke of motor is determined by pulse count.

- Generally speaking, the final control error deviation of servo motor is ±1 pulse.

## 13.6 Explanation of function for NC position control instruction

The NC position control of FBs-PLC has following four related instructions:

- FUN140 (HSPSO) high speed pulse output instruction, which includes following 8 extension positioning instructions:

| | | |
|---|---|---|
| 1. SPD | 5. ACT | Used for positioning program coding and stored to SR operand area of FUN140 |
| 2. DRV | 6. EXT | |
| 3. DRVC | 7. GOTO | |
| 4. WAIT | 8. MEND | |

- FUN141 (MPARA) positioning parameter setting instruction

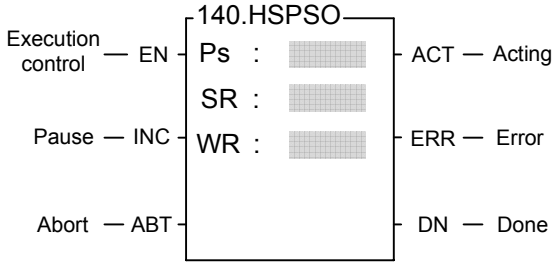
- FUN142 (PSOFF) enforcing pulse output stop instruction.

- FUN143 (PSCNV) converting the current pulse value to displaying value instruction.

The following function explanations are for the above mentioned 4 instructions:

| FUN 140<br>HSPSO | High Speed Pulse Output<br>(Including the extended positioning instruction) | FUN 140<br>HSPSO |
|---|---|---|

Ladder symbol




Ps: The set number of Pulse Output (0～3)
0:Y0 & Y1
1:Y2 & Y3
2:Y4 & Y5
3:Y6 & Y7

SR: Starting register for positioning program (example explanation)

WR: Starting register for instruction operation (example explanation). It controls 7 registers, which the other program cannot repeat in using.

| Operand \ Range | HR | DR | ROR | K |
|---|---|---|---|---|
| | R0 \| R3839 | D0 \| D3999 | R5000 \| R8071 | |
| Ps | | | | 0～3 |
| SR | ○ | ○ | ○ | |
| WR | ○ | ○ | ○* | |

Instruction Explanation

1. The NC positioning program of FUN140 (HSPSO) instruction is a program written and edited with text programming. We named every position point as a step (which includes output frequency, traveling distance, and transfer conditions). For one FUN140, it can be arranged with 250 steps of positioning points at the most, with every step of positioning point controlled by 9 registers.
2. The best benefit to store the positioning program into the registers is that in the case of association with MMI (Man Machine Interface) to operate settings, it may save and reload the positioning program via MMI when replacing the molds.
3. When execution control “EN”=1, if the other FUN140 instructions to control Ps0～3 are not active (corresponding status of Ps0=M1992, Ps1=M1993, Ps2=M1994, and Ps3=M1995 will be ON), it will start to execute from the next step of positioning point (when goes to the last step, it will be restarted from the first step to perform); if Ps0～3 is controlled by other FUN140 instruction (corresponding status of Ps0=M1992, Ps1=M1993, Ps2=M1994, and Ps3=M1995 would be OFF), this instruction will acquire the pulse output right of positioning control once the controlling FUN140 has released the control right.
4. When execution control input “EN” =0, it stops the pulse output immediately.
5. When output pause “PAU” =1 and execution control “EN” was 1 beforehand, it will pause the pulse output. When output pause “PAU” =0 and execution control is still 1, it will continue the unfinished pulse output.
6. When output abort “ABT”=1, it stops pulse output immediately. (When the execution control input “EN” becomes 1 next time, it will restart from the first step of positioning point to execute.)
7. While the pulse is in output transmitting, the output indication “ACT” is ON.
8. When there is execution error, the output indication “ERR” will be ON.
   (The error code is stored in the error code register.)
9. When each step of positioning point is complete, the output indication “DN” will be ON.

| FUN 140<br>HSPSO | High Speed Pulse Output<br>(Including the extended positioning instruction) | FUN 140<br>HSPSO |
|---|---|---|

*** The working mode of Pulse Output must be set (without setting, Y0～Y7 will be treated as general output) to be one of U/D, P/R, or A/B mode, thus the Pulse Output may have a regular output.

U/D Mode : Y0 (Y2, Y4, Y6), it sends out upward counting pulse.
Y1 (Y3, Y5, Y7), it sends out downward counting pulse.

P/R Mode : Y0 (Y2, Y4, Y6), it sends the pulse out.
Y1 (Y3, Y5, Y7), it sends out the directional signal;
ON=upward counting, OFF= downward counting.

A/B Mode : Y0 (Y2, Y4, Y6), it sends out the phase A pulse.
Y1 (Y3, Y5, Y7), it sends out the phase B pulse.

- The output polarity for Pulse Output can select to be Normal ON or Normal OFF.

〔The interfaces for positioning control〕

| | |
|---|---|
| M1991 | ON : stop or pause FUN140, slow down and stop pulse output. |
| | OFF : stop or pause FUN140, stop pulse output immediately. |
| M1992 | ON : Ps0 Ready |
| | OFF : Ps0 is in action |
| M1993 | ON : Ps1 Ready |
| | OFF : Ps1 is in action |
| M1994 | ON : Ps2 Ready |
| | OFF : Ps2 is in action |
| M1995 | ON : Ps3 Ready |
| | OFF : Ps3 is in action |
| M1996 | ON : Ps0 has finished the last step |
| M1997 | ON : Ps1 has finished the last step |
| M1998 | ON : Ps2 has finished the last step |
| M1999 | ON : Ps3 has finished the last step |

M2000 : ON, multi axes acting simultaneously (At the same scan, when execution control "EN"= 1of FUN140 instructions which control Ps0～3, their pulses output will be sent at the same time without any time lag).
: OFF, as the FUN140 for Ps0～3 starts, corresponding axis pulse output will be sent immediately; since the ladder program is executed in sequence, therefore even the FUN140 for Ps0～3 started at the same scan, there must be some time lag between them.

| Ps No. | Current output frequency | Current pulse position | The remaining pulse counts to be transmitted | Error code |
|---|---|---|---|---|
| Ps0 | DR4080 | DR4088 | DR4072 | R4060 |
| Ps1 | DR4082 | DR4090 | DR4074 | R4061 |
| Ps2 | DR4084 | DR4092 | DR4076 | R4062 |
| Ps3 | DR4086 | DR4094 | DR4078 | R4063 |

※ R4056 : When the value of low byte=5AH, it can be dynamically changed for its output frequency during the high speed pulse output transmitting at any time.

When the value of low byte is not 5AH, it can not be dynamically changed for its output frequency during the high speed pulse output transmitting.

The default value of R4056 is 0

| FUN 140<br>HSPSO | High Speed Pulse Output<br>(Including the extended positioning instruction) | FUN 140<br>HSPSO |
|---|---|---|

R4064 **:** The step number (positioning point) which has been completed of Ps0.

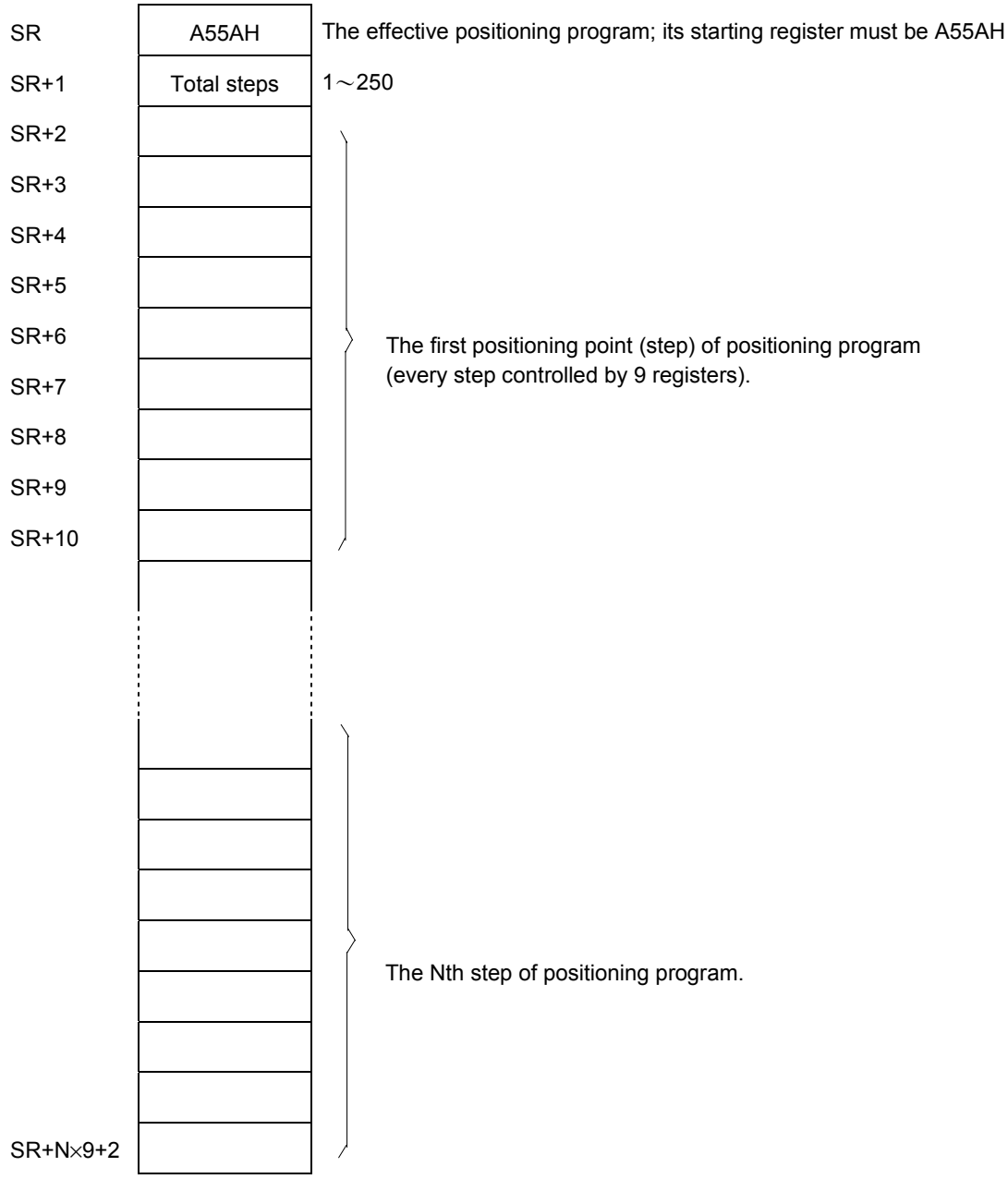
R4065 **:** The step number (positioning point) which has been completed of Ps1.

R4066 **:** The step number (positioning point) which has been completed of Ps2.

R4067 **:** The step number (positioning point) which has been completed of Ps3.

- Format of positioning program:

SR: Starting register of registers block which reserved to store positioning program, explained as follows:



| FUN 140<br>HSPSO | High Speed Pulse Output<br>(Including the extended positioning instruction) | FUN 140<br>HSPSO |
|---|---|---|

● Explanation for working register of instruction operation:

WR is the starting register.

| | |
|---|---|
| WR+0 | Being executed or stopped step |
| WR+1 | Working flag |
| WR+2 | Controlled by system |
| WR+3 | Controlled by system |
| WR+4 | Controlled by system |
| WR+5 | Controlled by system |
| WR+6 | Controlled by system |

WR+0 **:** If this instruction is in execution, the content of this register represents the step (1～N) being performed.
if this instruction is not in execution, the content of this register represents the step where it stopped at present
When execution control “EN” =1, it will perform the next step, i.e. the current step plus 1 (if the current step is at the last step, it will restart to perform from the first step).
Before starting the execution control “EN” =1, the user can renew the content of WR+0 to determine starting from which step to perform (when the content of WR+0 =0, and execution control “EN” =1, it represents that the execution starts from the first step).

WR+1 **:** B0～B7, total steps

B8 = ON, output paused

B9 = ON, waiting for transfer condition

B10 = ON, endless output (the stroke operand of DRV command is set to be 0 )

B12 = ON, pulse output transmitting (the status of output indicator “ACT”)

B13 = ON, instruction execution error (the status of output indicator “ERR”)

B14 = ON, finished being executed step (the status of output indicator “DN”)

*** Once the FUN140 instruction has been started (the B12 of WR+1=ON) and if suspended by a shut down for emergency or switchover from auto to manual mode while the pulse output has not yet completed, this instruction will be negated at next execution. It must clear the WR+1 register to be 0 before restarts the instruction next time, so as to make the instruction be initiated again; otherwise, the pulse output will not appear!

*** Whether execution control “EN” =0 or 1, executing the FUN140 instruction every scan , there won’t have the situation mentioned above.

*** When step which has been completed, the output indication “DN” will turn ON and keep such status if suspending ; the user may turn OFF the status of “DN” by using the rising edge of output coil controlled by "DN" to clear the content of WR+1 register to be 0, and it can be attained.

| FUN 140 HSPSO | High Speed Pulse Output (Including the extended positioning instruction) | FUN 140 HSPSO |
|---|---|---|

| Error indication | Error code | |
|---|---|---|
| R4060（Ps0） | 0 : Error free | |
| R4061（Ps1） | 1 : Parameter 0 error | |
| R4062（Ps2） | 2 : Parameter 1 error | |
| R4063（Ps3） | 3 : Parameter 2 error | |
| | 4 : Parameter 3 error | |
| | 5 : Parameter 4 error | |
| | 6 Parameter 5 error | The possible error codes for FUN141 execution |
| | 7 : Parameter 6 error | |
| | 8 : Parameter 7 error | |
| | 9 : Parameter 8 error | |
| | 10 : Parameter 9 error | |
| | 13 Parameter 12 error | |
| | 15 Parameter 14 error | |
| | 30 : Error of variable address for speed setting | |
| | 31 : Error of setting value for speed setting | |
| | 32 : Error of variable address for stroke setting | |
| | 33 : Error of setting value for stroke setting | |
| | 34 : Illegal positioning program | |
| | 35 : Length error of total step | The possible error codes for FUN140 execution |
| | 36 : Over the maximum step | |
| | 37 : Limited frequency error | |
| | 38 : Initiate/stop frequency error | |
| | 39 : Over range of compensation value for movement | |
| | 40 : Over range of moving stroke | |
| | 41 : ABS positioning is not allowed within DRVC commands | |

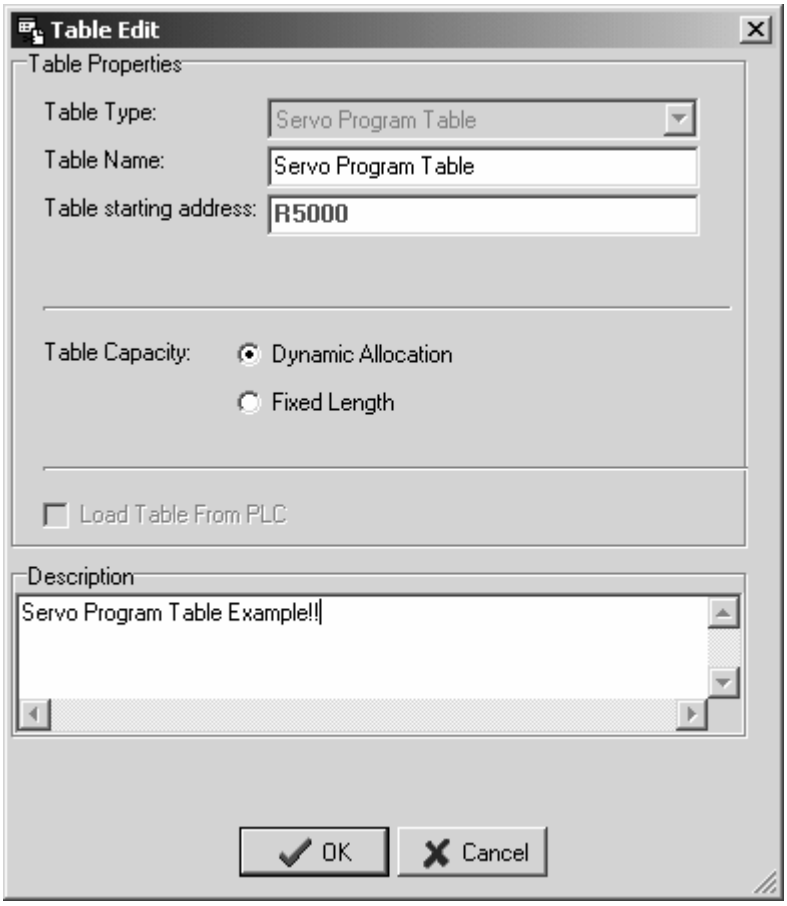
Note：The content of error indication register will keep the latest error code. Making sure that no more error to happen, you can clear the content of error indication register to be 0; as long as the content maintains at 0, it represents that there's no error happened.

## Editing Servo Program Table with WinProladder

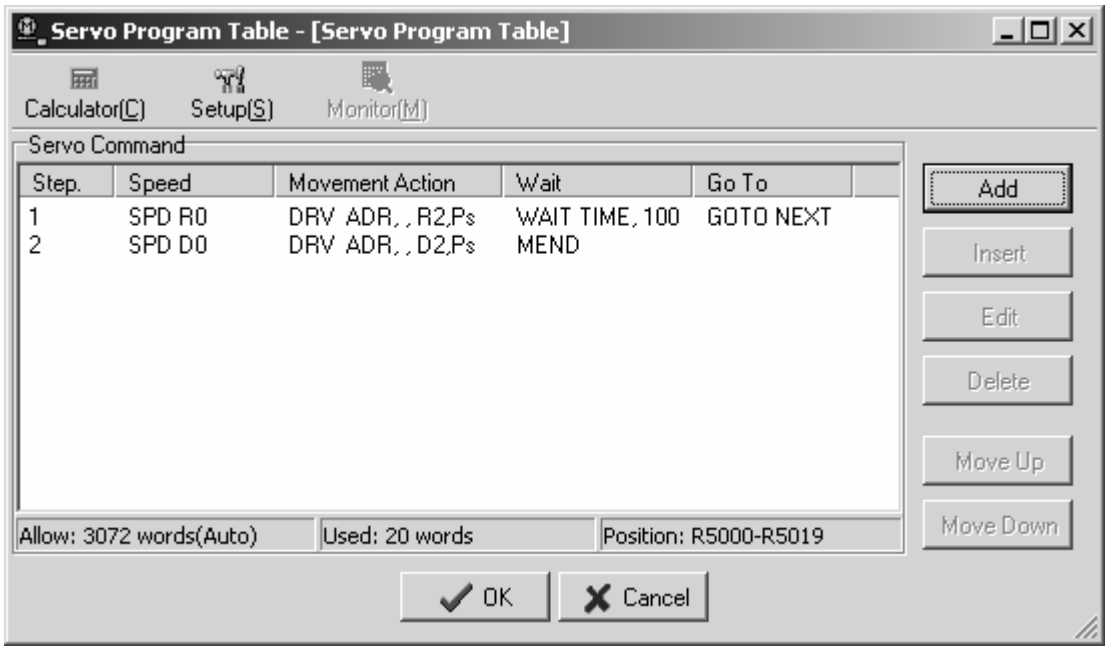
Click the "Servo Program Table" Item which in project windows：

Project name
└ Table Edit
└ Servo Program Table → Click right button and select "New Table"

| FUN 140<br>HSPSO | High Speed Pulse Output<br>(Including the extended positioning instruction) | FUN 140<br>HSPSO |
|---|---|---|




- Table Type **:** It will be fixed to ” Servo Program Table ”.
- Table Name **:** For modify or debug, you can give a convenient name.
- Table Starting address **:** Enter the address which Starting register of Servo Program Table.

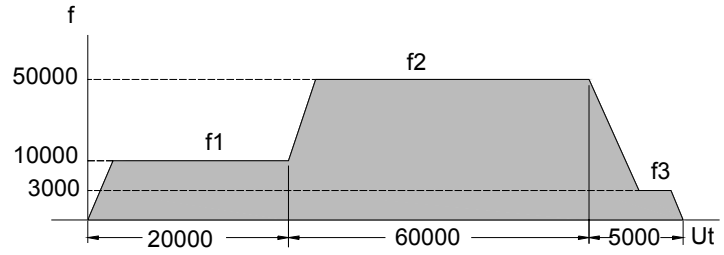


| FUN 140<br>HSPSO | High Speed Pulse Output<br>(Including the extended positioning instruction) | FUN 140<br>HSPSO |
|---|---|---|

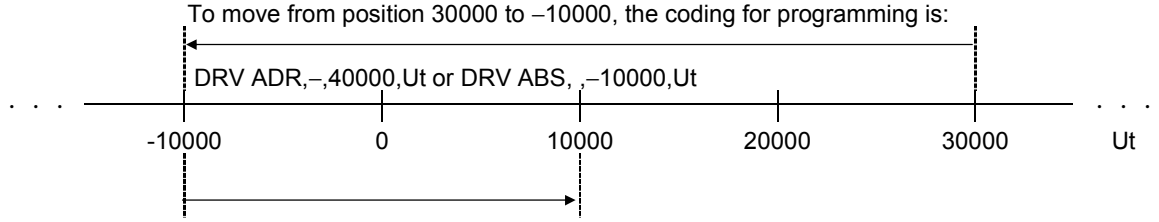
- For easy programming and trouble shooting, the WinProladder provides the text editing environment to edit the motion program(servo program table) for FUN140 execution; Key in the complete FUN140 instruction first and then move the cursor to the position of it, pressing the hot key ”Z”, then comes the text editing environment. The user can create the new motion program or display the existed program under this friendly user interface operation.
- Extended positioning instructions are listed as follows:

| Instruction | Operand | Explanation |
|---|---|---|
| SPD | XXXXXX or<br>Rxxxx or<br>Dxxxx | • Moving speed in frequency or velocity (FUN141 Parameter_0=0 represents velocity; Parameter_0=1 or 2 for frequency; the system default is frequency). The operand can be input directly with constant or variable (Rxxxx, Dxxxx); when the operand is variable, it needs 2 registers, e.g. D10 represents D10 (Low Word) and D11 (High Word), which is the setting of frequency or velocity.<br>• When selecting to use the velocity setting, the system will automatically convert the velocity setting to corresponding output frequency.<br>• Output frequency range: 1≦output frequency≦921600 Hz.<br>*** When the output frequency is 0, this instruction will wait until the setting value isn’t 0 to execute the positioning pulse output. |
| DRV | ADR，+，XXXXXXXX，Ut<br>ADR，+，XXXXXXXX，Ps<br>ADR，–，XXXXXXXX，Ut<br>ADR，–，XXXXXXXX，Ps<br>ADR，　，XXXXXXXX，Ut<br>ADR，　，–XXXXXXXX，Ut<br>ADR，　，XXXXXXXX，Ps<br>ADR，　，–XXXXXXXX，Ps<br>ADR，+，Rxxxx，Ut<br>ADR，+，Rxxxx，Ps<br>ADR，–，Rxxxx，Ut<br>ADR，–，Rxxxx，Ps<br>ADR，　，Rxxxx，Ut<br>ADR，　，Rxxxx，Ps<br>ADR，+，Dxxxx，Ut<br>ADR，+，Dxxxx，Ps<br>ADR，–，Dxxxx，Ut<br>ADR，–，Dxxxx，Ps<br>ADR，　，Dxxxx，Ut<br>ADR，　，Dxxxx，Ps<br>ABS，　，XXXXXXXX，Ut<br>ABS，　，–XXXXXXXX，Ut<br>ABS，　，XXXXXXXX，Ps<br>ABS，　，–XXXXXXXX，Ps<br>ABS，　，Rxxxx，Ut<br>ABS，　，Rxxxx，Ps<br>ABS，　，Dxxxx，Ut<br>ABS，　，Dxxxx，Ps | • Moving stroke setting in Ps or mm,Deg,Inch<br>(When FUN141 Parameter_0=1, the setting stroke in Ut is Ps; Parameter_0=0 or 2, the setting stroke in Ut is mm, Deg, Inch; the system default for Ut is Ps).<br>• When 4_th operand of DRV is Ut (not Ps) , according to parameter setting of 1, 2, 3 of FUN141, the system will convert the corresponding pulse count to output.<br>•There are 4 operands to construct DRV instruction as follows:<br>1_st operand: coordinate selection.<br>ADR or ABS: ADR, relative distance movement<br>ABS, absolute position movement<br>2_nd operand:<br>revolving direction selection (Valid for ADR only).<br>'+' , forward or clockwise<br>'–' , backward or counterclockwise<br>' ' ,direction is determined by the setting value<br>(positive value: forward; negative value: backward)<br>3_rd operand: moving stroke setting<br>XXXXXXXX:<br>or<br>–XXXXXXXX<br>or Rxxxx<br>or Dxxxx<br>It can directly input with constant or variable (Rxxxx, Dxxxx); it needs 2 registers when adopting the variable, e.g. R0 represents R0 (Low Word) and R1 (High Word) as the setting of moving stroke.<br>*** When the setting of moving stroke is 0 and 1_st operand is ADR, it represents to revolve endless.<br>Stroke setting range: –99999999 ≤ stroke setting ≤ 99999999<br>4_th operand: resolution of stroke setting<br>Ut or Ps:for Ut, the resolution is one unit;<br>(it is determined by parameter 0, 3 of FUN141); for Ps, the enforced resolution is one pulse. |

| FUN 140<br>HSPSO | High Speed Pulse Output<br>(Including the extended positioning instruction) | FUN 140<br>HSPSO |
|---|---|---|

| Instruction | Operand | Explanation |
|---|---|---|
| DRVC | ADR，+，XXXXXXXX，Ut<br>or or or or<br>ABS，–，Rxxxx，Ps<br>or<br>Dxxxx | The usage of DRVC and the operand explanation is the same as DRV's instruction.<br>*** DRVC is used to do successive speed changing control (8 speeds at the most).<br>*** Of the successive speed changing control, only the first DRVC instruction can use the absolute value coordinate for positioning.<br>*** The revolution direction of DRVC can only be decided by '+' or '–'.<br>*** The revolution direction only determined by the first DRVC of successive DRVC instructions; i.e. the successive speed changing control can only be the same direction.<br><br>For example: successive 3 speed changing control<br>001 SPD 10000 * Pulse frequency = 10KHz.<br>DRVC ADR，+，20000，Ut * Forward 20000 units.<br>GOTO NEXT<br>002 SPD 50000 * Pulse frequency =50 KHz<br>DRVC ADR，+，60000，Ut * Forward 60000 units.<br>GOTO NEXT<br>003 SPD 3000 * Pulse frequency = 3KHz.<br>DRV ADR，+，5000，Ut * Forward 5000 units.<br>WAIT X0 * Wait until X0 ON to restart from<br>GOTO 1 the first step to execute.<br><br>Note: The number of DRVC instructions must be the number of successive speeds deducted by 1, i.e. the successive speed changing control must be ended with the DRV instruction.<br>• The above mentioned example is for successive 3 speeds changing control, which used 2 DRVC instructions and the third must use DRV instruction.<br>• Diagram illustration for the above mentioned example:<br>  <br> |

Note: Comparison explanation between the relative coordinate positioning (ADR) and the absolute coordinate positioning (ABS)



| FUN 140<br>HSPSO | High Speed Pulse Output<br>(Including the extended positioning instruction) | FUN 140<br>HSPSO |
|---|---|---|

| Instruction | Operand | Explanation |
|---|---|---|
| WAIT | Time, XXXXX<br>or Rxxxx<br>or Dxxxx<br>or X0～X255<br>or Y0～Y255<br>or M0～M1911<br>or S0～S999 | • When pulse output is complete, performing the wait instruction for going to the next step. There are 5 kind of operands that explained as follows:<br>Time: The waiting time (the unit is 0.01 second), it can be directly input with constant or variable (Rxxxx or Dxxxx); when it is time up, performs the step that assigned by GOTO.<br>X0～X255: Waiting until the input status is ON, it performs the step that assigned by GOTO.<br>Y0～Y255:Waiting until the output status is ON, it performs the step that assigned by GOTO.<br>M0～M1911: Waiting until the internal relay is ON, it performs the step that assigned by GOTO.<br>S0～S999: Waiting until the step relay is ON, it performs the step that assigned by GOTO. |
| ACT | Time，XXXXX<br>or Rxxxx<br>or Dxxxx | • After the time to output pulses described by operand of ACT, it performs immediately the step that assigned by GOTO, i.e. after the pulse output for a certain time, it performs the next step immediately. The action time (the unit is 0.01 second) can be directly input with constant or variable (Rxxxx or Dxxxx); when the action time is up, it performs the step assigned by GOTO. |
| EXT | X0～X255<br>or Y0～Y255<br>or M0～M1911<br>or S0～S999 | • External trigger instruction; when it is in pulse output (the number of pulses sending is not complete yet), if the status of external trigger is ON, it will perform the step assigned by GOTO immediately. If the status of external trigger is still OFF when the pulse output has been complete, it is the same as WAIT instruction; waiting the trigger signal ON, then perform the step assigned by GOTO. |
| GOTO | NEXT<br>or 1～N<br>or Rxxxx<br>or Dxxxx | • When matching the transfer condition of WAIT, ACT, EXT instruction, by using GOTO instruction to describe the step to be executed.<br>NEXT: It represents to perform the next step.<br>1～N: To perform the described number of step.<br>Rxxxx: The step to be performed is stored in register Rxxxx.<br>Dxxxx: The step to be performed is stored in register Dxxxx. |
| MEND | | The end of the positioning program. |

| FUN 140 HSPSO | High Speed Pulse Output (Including the extended positioning instruction) | FUN 140 HSPSO |
|---|---|---|

- The coding for positioning programming **:**

  First,it must complete the FUN140 instruction before the editing of positioning program, and assigned in FUN140 instruction the starting register of registers block to store positioning program. While editing the positioning program, it will store the newly edited positioning program to the assigned registers block; for every one positioning point (called as one step) edited, it is controlled by 9 registers. If there are N positioning points, it will be controlled by N × 9 + 2 registers in total.

  Note: The registers storing the positioning program can not be repeated in using!

- Format and example for the positioning program 1:

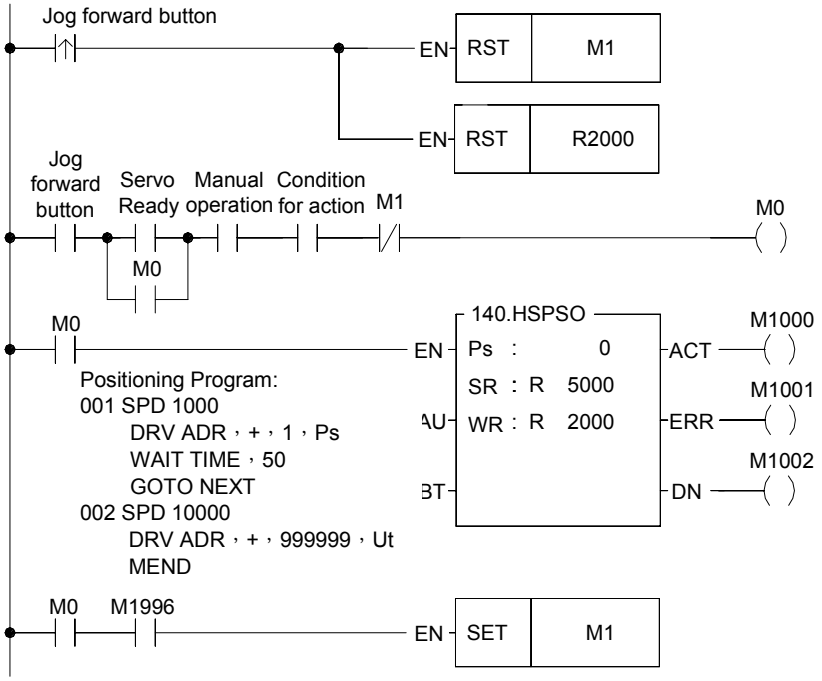
```
001 SPD   5000            ; Pulse frequency = 5KHz.
    DRV   ADR,+,10000,Ut  ; Moving forward 10000 units.
    WAIT  Time,100        ; Wait for 1 second.
    GOTO  NEXT            ; Perform the next step.
002 SPD   R1000           ; Pulse frequency is stored in DR1000 (R1001 and R1000).
    DRV   ADR,+,D100,Ut   ; Moving forward,the stroke is stored in DD100 (D101 and D100).
    WAIT  Time,R500       ; The waiting time is stored in R500.
    GOTO  NEXT            ; To perform the next step.
003 SPD   R1002           ; Pulse frequency is stored in DR1002 (R1003 and R1002).
    DRV   ADR,–,D102,Ut   ; Moving backward,the stroke is stored in DD102 (D103 and D102).
    EXT   X0              ; When external trigger X0 (slow down point) ON, it performs the next
    GOTO  NEXT            ; step immediately.
004 SPD   2000            ; Pulse frequency = 2KHz.
    DRV   ADR,–,R4072,Ps  ; Keep outputing the remain (stored in DR4072).
    WAIT  X1              : Wait until X1 ON,
    GOTO  1               : Perform the first step.
```

- Format and example for the positioning program 2:

```
001 SPD   R0              ; Pulse frequency is stored in DR0 (R1 & R0).
    DRV   ABS, ,D0,Ut     ; Move to the position stored in DD0 (D1 & D0).
    WAIT  M0              ; Wait until M0 ON,
    GOTO  NEXT            ; Perform the next step.
002 SPD   R2              ; Pulse frequency is stored in DR2 (R3 & R2).
    DRV   ADR, ,D2,Ut     ; Moving stroke is stored in DD2 (D3 & D2);working direction determined
                          ; by the sign of setting value
    MEND                  ; End of positioning program
```

## Program example: Jog forward

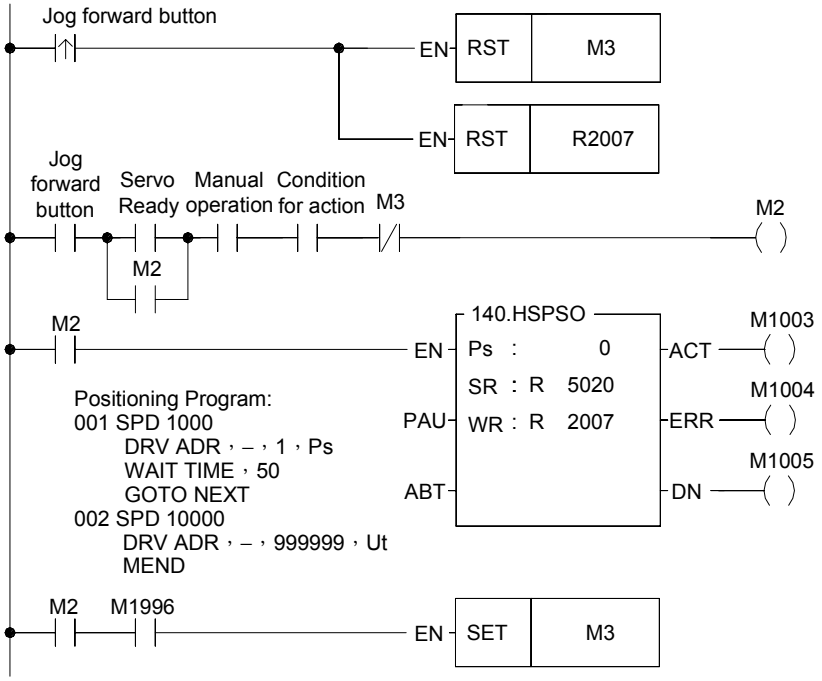
As the jog forward button has been pressed for less than 0.5 second (changeable), it sends out only one (changeable) pulse;
As the jog forward button has been pressed for more than 0.5 second (changeable), it continuously sends pulses out (the frequency is 10KHz, changeable), until the release of the jog forward button to stop the pulse transmitting; or it may be designed to send N pulses out at the most.




- Clear finish signal.
- Perform from the first step every time.
- When the last step been complete, set finish signal.

## Program example: Jog Backward

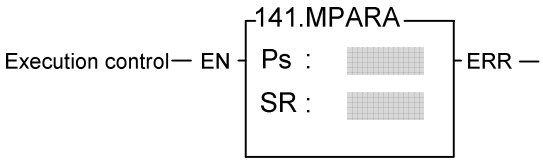
As the jog backward button has been pressed for less than 0.5 second (changeable) it sends out only one (changeable) pulse;
As the jog backward button has been pressed for more than 0.5 second (changeable), it continuously sends pulses out (the frequency is 10KHz, changeable), until the release of the jog backward button to stop the pulse transmitting; or it may be designed to send N pulses out at the most.




- Clear finish signal.
- Perform from the first step every time.
- When the last step been complete, set finish signal.

| FUN 141 MPARA | Instruction of Parameter Setting for Positioning Program | FUN 141 MPARA |
|---|---|---|

Ladder symbol




Ps: The set number of Pulse Output (0～3).

SR: Starting register for parameter table, it has totally 18 parameters which controlled by 24 registers.

| Range / Operand | HR | DR | ROR | K |
|---|---|---|---|---|
| | R0 \| R3839 | D0 \| D3999 | R5000 \| R8071 | |
| Ps | | | | 0～3 |
| SR | ○ | ○ | ○ | |

Instruction explanation

1.This instruction is not necessary if the system default for parameter value is matching what users need. However, if it needs to open the parameter value to do dynamic modification, this instruction is required.

2.This instruction incorporates with FUN140 for positioning control purpose, each axis can have one FUN141 instruction only.

3. Whether the execution control input “EN” = 0 or 1, anyway, this instruction will be performed.

4. When there is error in parameter value, the output indication “ERR” will be ON, and the error code is appeared in the error code register.

Explanation for the parameter table:

SR =Starting register of parameter table, suppose it is R2000.

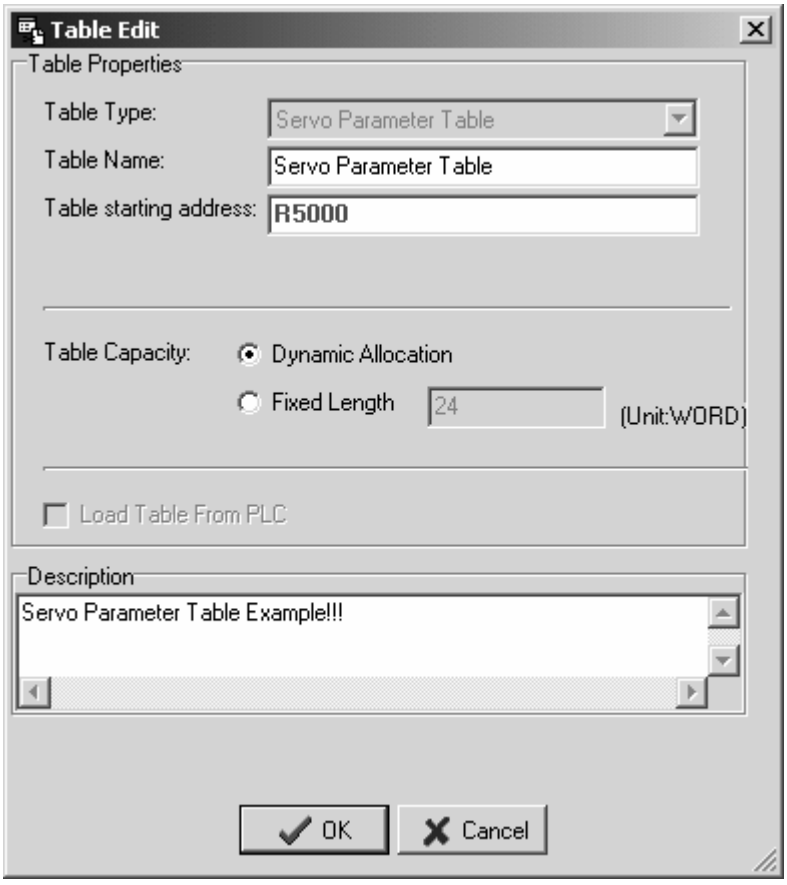
| | | | |
|---|---|---|---|
| R2000 (SR+0) | 0～2 | Parameter 0 | System default =1 |
| R2001 (SR+1) | 1～65535 Ps/Rev | Parameter 1 | System default =2000 |
| DR2002 (SR+2) | 1～999999 μM/Rev<br>1～999999 mDeg/Rev<br>1～999999×0.1 mInch/Rev | Parameter 2 | System default =2000 |
| R2004 (SR+4) | 0～3 | Parameter 3 | System default =2 |
| DR2005 (SR+5) | 1～921600 Ps/Sec<br>1～153000 | Parameter 4 | System default =512000 |
| DR2007 (SR+7) | 0～921600 Ps/Sec | Parameter 5 | System default =141 |
| R2009 (SR+9) | Reserved | Parameter 6 | System default =0 |
| R2010 (SR+10) | 0～32767 | Parameter 7 | System default =0 |
| R2011 (SR+11) | 0～30000 | Parameter 8 | System default =5000 |
| R2012 (SR+12) | 0～1 | Parameter 9 | System default =0 |
| R2013 (SR+13) | -32768～32767 | Parameter 10 | System default =0 |
| R2014 (SR+14) | -32768～32767 | Parameter 11 | System default =0 |
| R2015 (SR+15) | 0～30000 | Parameter 12 | System default =0 |
| R2016 (SR+16) | Reserved | Parameter 13 | System default =1 |
| DR2017 (SR+17) | 0～4294967295 | Parameter 14 | System default =0 |
| DR2019 (SR+19) | Reserved | Parameter 15 | System default =20000 |
| DR2021 (SR+21) | Reserved | Parameter 16 | System default =1000 |
| R2023 (SR+23) | Reserved | Parameter 17 | System default =10 |

| FUN 141 MPARA | Instruction of Parameter Setting for Positioning Program | FUN 141 MPARA |
|---|---|---|

## Editing Servo Parameter Table with WinProladder

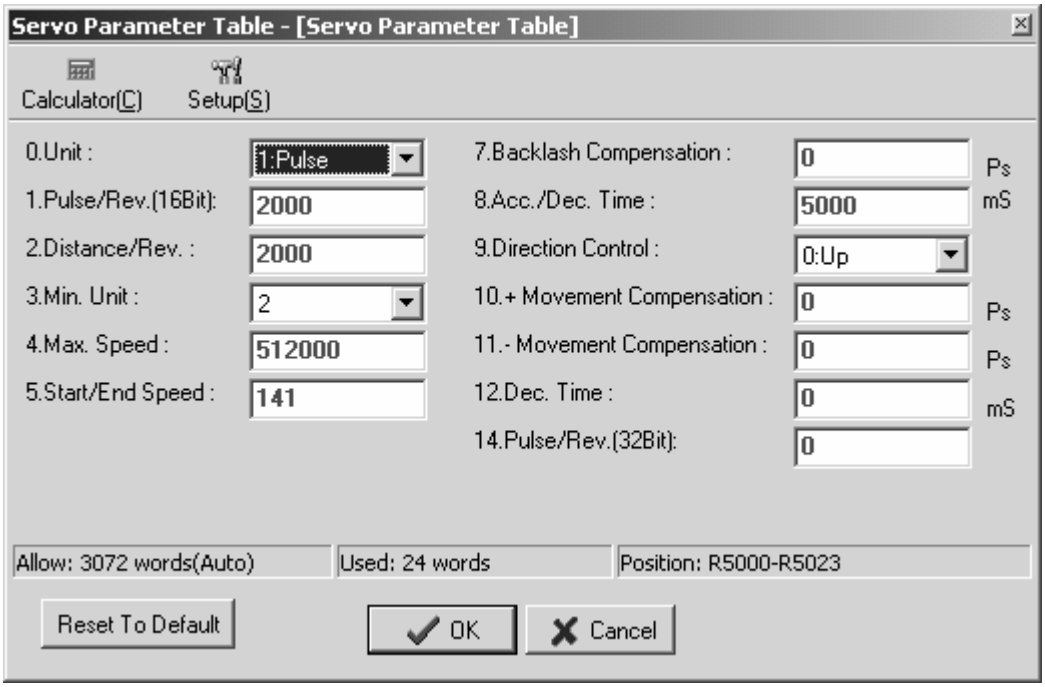
Click the “Servo Parameter Table” Item which in project windows **:**

Project name
└ Table Edit
└ Servo Parameter Table → Click right button and select “New Table”




- Table Type **:** It will be fixed to ” Servo Parameter Table ”.
- Table Name **:** For modify or debug, you can give a convenient name.
- Table Starting address **:** Enter the address which Starting register of Servo Parameter Table.

| FUN 141 MPARA | Instruction of Parameter Setting for Positioning Program | FUN 141 MPARA |
|---|---|---|




Explanation for the parameter:

- Parameter 0: The setting of unit, its default is 1.
  - When the setting value is 0, the moving stroke and speed setting in the positioning program will all be assigned with the unit of mm, Deg, Inch, so called machine unit.
  - When the setting value is 1, the moving stroke and speed setting in the positioning program will all be assigned with the unit of Pulse, so called motor unit.
  - When the setting value is 2, the moving stroke setting in the positioning program will all be assigned with the unit of mm, Deg, Inch, and the speed setting will all be assigned with the unit of Pulse/Sec, which is called as compound unit.

| Parameter 0, unit setting | "0" machine unit | "1" motor unit | "2" compound unit |
|---|---|---|---|
| Parameter 1, 2 | Must be set | No need to set | Must be set |
| Parameter 3, 7, 10, 11 | mm，Deg，Inch | Ps | mm，Deg，Inch |
| Parameter 4,5,6,15,16 | Cm/Min，Deg/Min，Inch/Min | Ps/Sec | Ps/Sec |

- Parameter 1: Pulse count/1-revolution, its default is 2000, i.e. 2000 Ps/Rev.
  - The pulse counts needed to turn the motor for one revolution
    A= 1～65535 (for value greater than 32767, it is set with hexadecimal) Ps/Rev
  - When Parameter 14 = 0, Parameter 1 is the setting for Pulse /Rev
  - When Parameter 14 ≠ 0, Parameter 14 is the setting for Pulse/Rev

- Parameter 2: Movement/1 revolution, its default is 2000, i.e. 2000 Ps/Rev.
  - The movement while motor turning for one revolution.
    B=1～999999 μM/Rev
    1～999999 mDeg/Rev
    1～999999×0.1 mInch/Rev

| FUN 141 MPARA | Instruction of Parameter Setting for Positioning Program | FUN 141 MPARA |
|---|---|---|

- Parameter 3: The resolution of moving stroke setting, its default is 2.

| Parameter 0 / Parameter 3 | Set value=0, machine unit; Set value=2, compound unit; | | | Set value=1 motor unit (Ps) |
|---|---|---|---|---|
| | mm | Deg | Inch | |
| Set value =0 | ×1 | ×1 | ×0.1 | ×1000 |
| Set value =1 | ×0.1 | ×0.1 | ×0.01 | ×100 |
| Set value =2 | ×0.01 | ×0.01 | ×0.001 | ×10 |
| Set value =3 | ×0.001 | ×0.001 | ×0.0001 | ×1 |

- Parameter 4: The limited speed setting, its default is 460000, i.e. 460000 Ps/Sec.
  - Motor and compound unit: 1～921600 Ps/Sec.
  - Machine unit: 1～153000 (cm/Min, ×10 Deg/Min, Inch/Min).
    However, the limited frequency can't be greater than921600 Ps/Sec.
    f_max = （V_max ×1000×A）/（6×B）≤ 921600 Ps/Sec
    f_min ≥ 1 Ps/Sec

  Note: A = Parameter 1, B =Parameter 2.

- Parameter 5: Initiate/Stop speed, the default = 141.
  - Motor and compound unit: 1～921600 Ps/Sec.
  - Machine unit: 1～15300 (cm/Min, ×10 Deg/Min, Inch/Min).
    However, the limited frequency can't be greater than 921600 Ps/Sec.

- Parameter 6: Reserved, the default = 0.

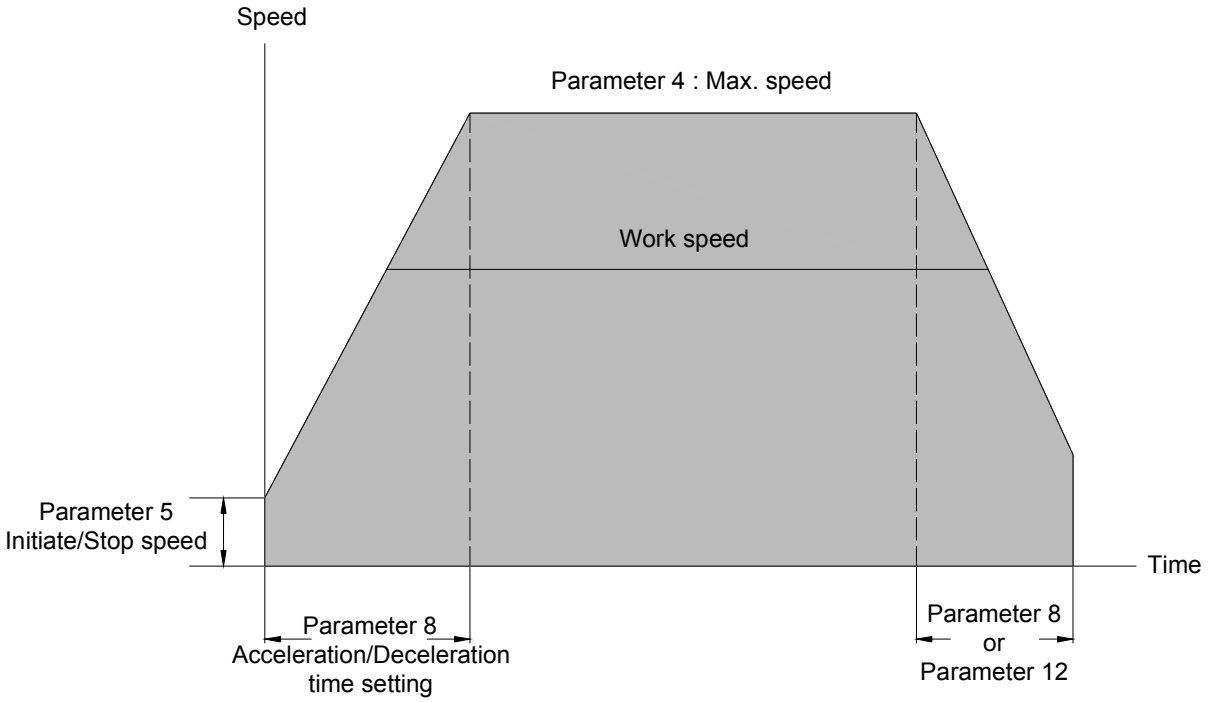
- Parameter 7: Backlash compensation, the default =0.
  - Setting range: 0～32767 Ps.
  - While backward traveling, the traveling distance will be added with this value automatically.

- Parameter 8: Acceleration/Deceleration time setting, the default = 5000, and the unit is mS.
  - Setting range: 0～30000 mS.
  - The setting value represents the time required to accelerate from idle state upto limited speed state or declerate from the limited speed state down to the idle state.
  - The acceleration/deceleration is constant slope depending on Parameter 4 / Parameter 8
  - When Parameter 12 = 0, Parameter 8 is the deceleration time
  - There will have the auto deceleration function for short stroke movement.

- Parameter 9: Coordinnate direction setting, the default =0.
  - Setting value =0, while in forward pulse output, the current Ps value is adding up.
    While in backward pulse output, the current Ps value is deducting down.
  - Setting value =1, while in forward pulse output, the current Ps value is deducting down.
    While in backward pulse output, the current Ps value is adding up.

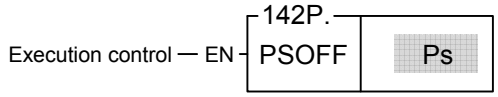
| FUN 141 MPARA | Instruction of Parameter Setting for Positioning Program | FUN 141 MPARA |
|---|---|---|

- Parameter 10: Forward movement compensation, the default = 0.
  - Setting range: –32768～32767 Ps.
  - When it is in forward pulse output, it will automatically add with this value as the moving distance.
- Parameter 11: Backward movement compensation, the default =0.
  - Setting range: –32768～32767 Ps.
  - When it is in backward pulse output, it will automatically add with this value as the moving distance.
- Parameter 12: Deceleration time setting, the default =0, and the unit is mS.
  - Setting range: 0～30000 mS.
  - When Parameter 12 = 0, Parameter 8 is the deceleration time
  - When Parameter 12 ≠ 0, Parameter 12 is the deceleration time
- Parameter 13: Reserved.
- Parameter 14: Pulse count/1-revolution, the default = 0.
  - The pulse counts needed to turn the motor for one revolution
  - When Parameter 14 = 0, Parameter 1 is the setting for Pulse /Rev
  - When Parameter 14 ≠ 0, Parameter 14 is the setting for Pulse/Rev
- Parameter 15: Reserved, it is recommended to be used as return home speed, the default = 20000 Ps/Sec.
- Parameter 16: Reserved, it is recommended to be used as slow down speed while returning home , the default = 1000 Ps/Sec.
- Parameter 17: Reserved.



| FUN 142 P PSOFF | Enforcing to Stop Pulse Output | FUN 142 P PSOFF |
|---|---|---|

Ladder symbol




N: 0～3, enforces the asigned set number of Pulse Output to stop its output.

## Instruction Explanation

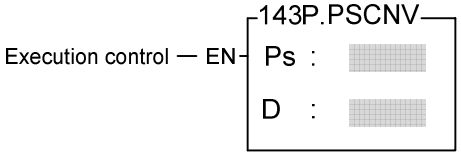
1. When stop control “EN” =1, or changes from 0→1( P instruction), this instruction will enforce the assigned set number of Pulse Output to stop its output.
2. When applying in the process of return home , as the home has returned, it can immediately stop the pulse output by using this instruction, so as to make it stop at the same position every time when performing machine homing.

## Program example


; When M0 changes from 0→1, it enforces the Ps0 to stop the pulse output.

| FUN 143 P PSCNV | Converting the Current Pulse Value to the Displaying Value (mm, Deg, Inch, PS) | FUN 143 P PSCNV |
|---|---|---|

Ladder symbol




Ps: 0～3; converting the asigned pulse position to mm (Deg, Inch, PS) which has the same unit as the set point, so as to make the current position displayed.

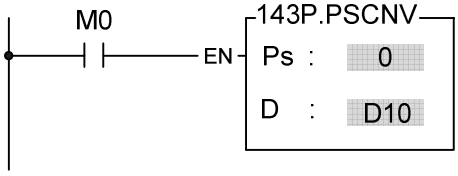
D: Registers that store the current position after conversion. It uses 2 registers, e.g. D10 represents D10 (Low Word) and D11 (High Word) two registers.

| Range / Operand | HR | DR | ROR | K |
|---|---|---|---|---|
| | R0 \| R3839 | D0 \| D3999 | R5000 \| R8071 | |
| Ps | | | | 0～3 |
| D | ○ | ○ | ○* | |

## Instruction Explanation

1. When execution control “EN” =1 or changes from 0→1( P instruction), this instruction will convert the assigned current pulse position (PS) to be the mm (or Deg, Inch, or PS) that has the same unit as the set value, so as to make current position displaying.
2. After the FUN140 instruction has been performed, it will then be able to get the correct conversion value by executing this instruction.

## Program Example



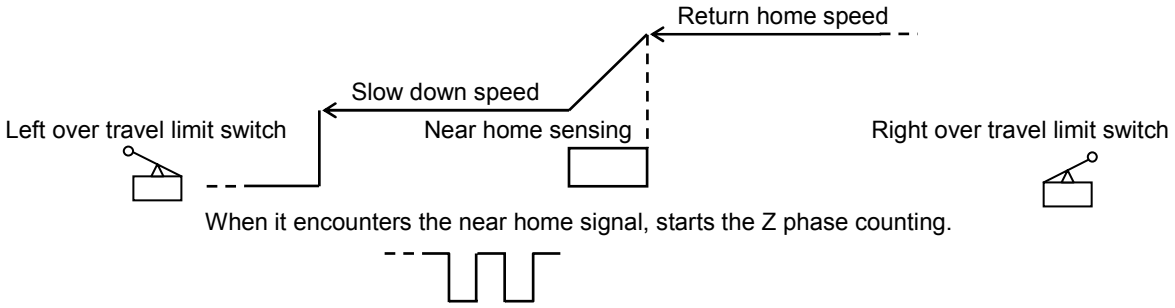

; When M0=1, it converts the current pulse position of Ps0 (DR4088) to the mm (or Deg or Inch or PS) that has the same unit as the set value, and store it into the DD10 to make the current position displaying.

## 13.7 Machine homing

The machine set which undertakes relative model Encoder as shifting detector usually need the reset action for the reference of positioning coordinate; we called this action as machine homing (seeking for zero reference).

The machine homing diagram for NC servo unit is as follows:

Method 1:



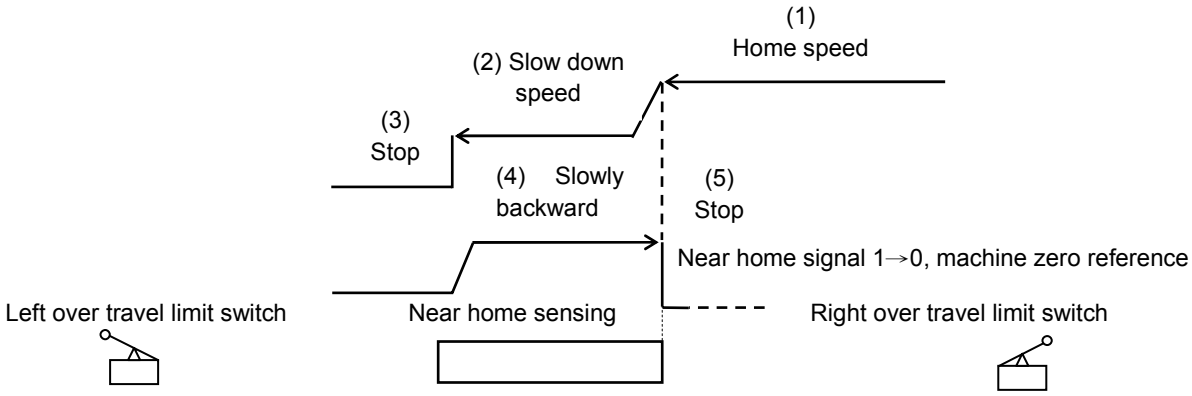

Z phase counting is up, the pulse output stops, then send out the CLR signal to clear the error counter of servo driver.

e.g.:

X3: Near home sensing input is configured as interrupt input; in the case of machine homing, it starts HSC4 to begin counting in X3+ interrupt service subroutine.

X2: Z phase counting input, it is configured as UP input of HSC4; the X2+ is prohibited to interrupt in regular time, when executing machine homing and X3 near home interrupt occurred, it starts HSC4 to begin Z phase counting. When HSC4 counting is up, it stops the pulse output, prohibit the X2+ interrupt, set home position to signal, and sends out the CLR signal to clear the error counter of servo driver. Please consult program example.

Method 2: According to application demand, it may slow down when encountering the near home sensor, while over the sensor a little far away, stop the pulse output, and then traveling slowly with backward direction; the very moment when it get out of near home sensor (the sensing signal changes from 1→0), it is treated as machine home. This program is simpler!



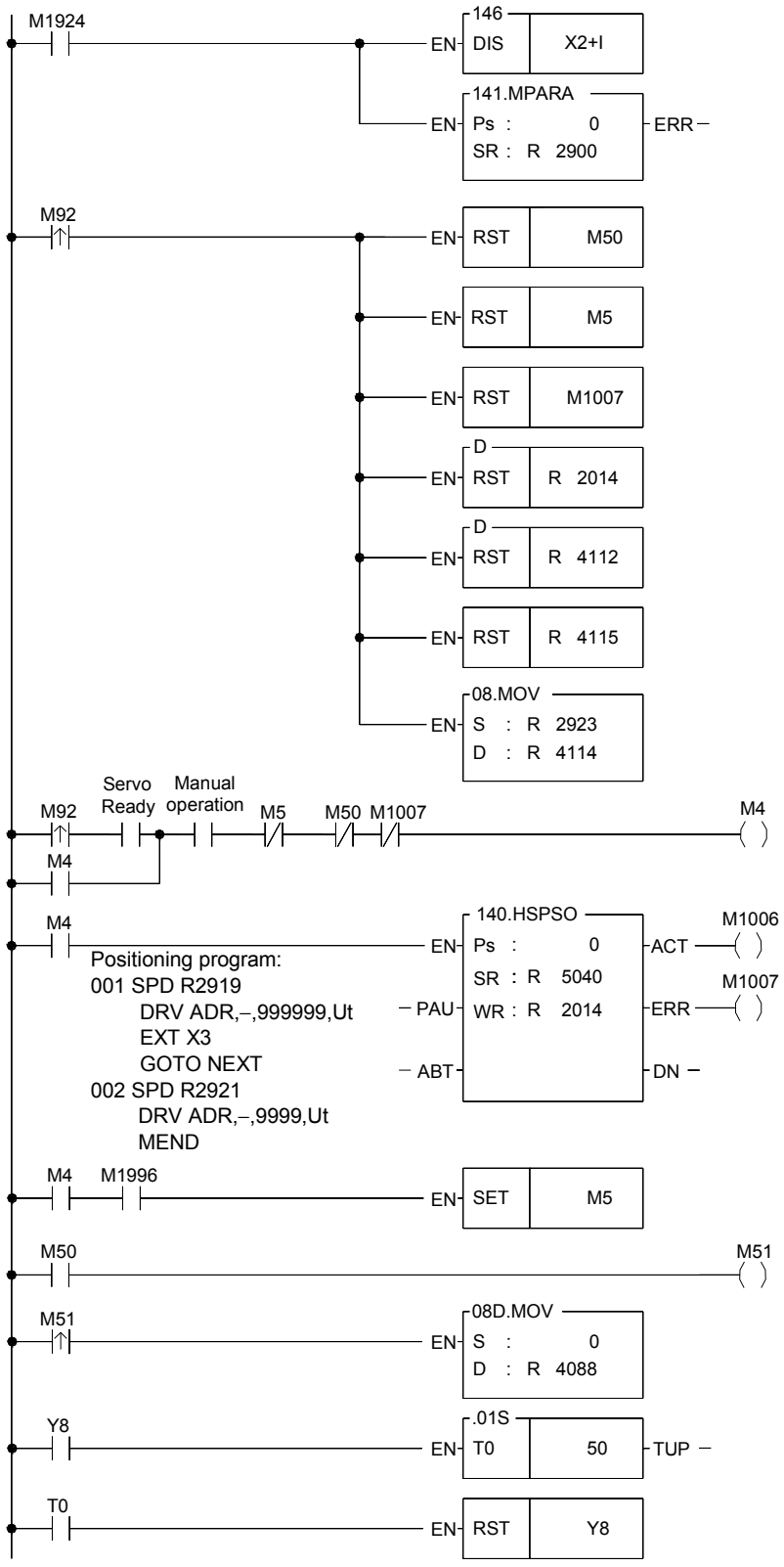

X3: Near home sensing input; it is configured as falling edge interrupt input.

- Once encountering the near home sensor, it will enable X3 falling edge interrupt, and slow down to stop within the near home sensing range.
- Slowly backward traveling until the near home sensing signal changes from 1→0.
- When the near home sensing signal changes from 1→0, it performs the X3- interrupt service subroutine immediately.
- The X3- interrupt service subroutine: Stops the pulse output immediately, prohibits the X3- interrupt, sets home position to signal, and sends out CLR signal to clear the error counter of servo driver. (Please consult the example program.)

## Program Example 1: Machine homing (method 1)

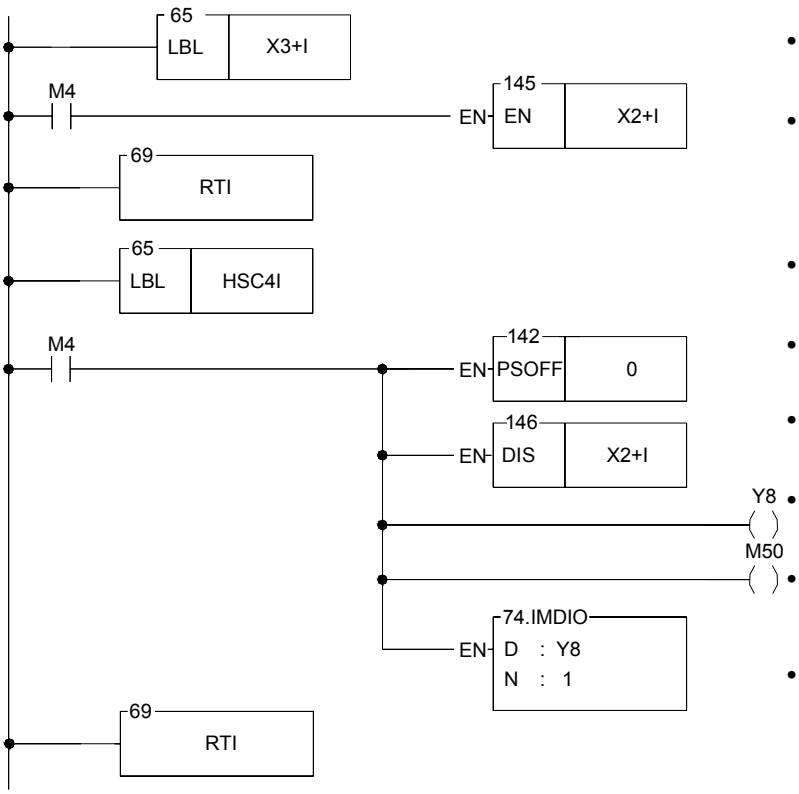
X2: Configured as the UP input of HSC4, and connected to Z phase input.
X3: Configured as the rising edge interrupt input, and connected to near home sensing input.

【Main Program】




- Prohibits X2+ interruption (HSC4 does not count)
- Parameter table R2900→R2923.
- Clears the homing completion signal.
- Clears the instruction completion signal for homing
- Clears the error signal.
- Clears the step pointer, it starts from the first step to execute.
- Clears the current value of HSC4.
- Clears the High Word of preset value for HSC4.
- Fill the prset value of HSC4 with the content of Parameter 17 of FUN141.
- Configure R5000～R5199 as the read only register (ROR) before programming, after then, when storing program, the Ladder program will automatically contains the positioning program.
- Homing instruction completed
- Signal for homing completion
- Fill the current PS registers with 0, while homing completed.
- Signal to clear error counter of servo driver -- Y8 is ON for 0.5 second.

【Sub Program】

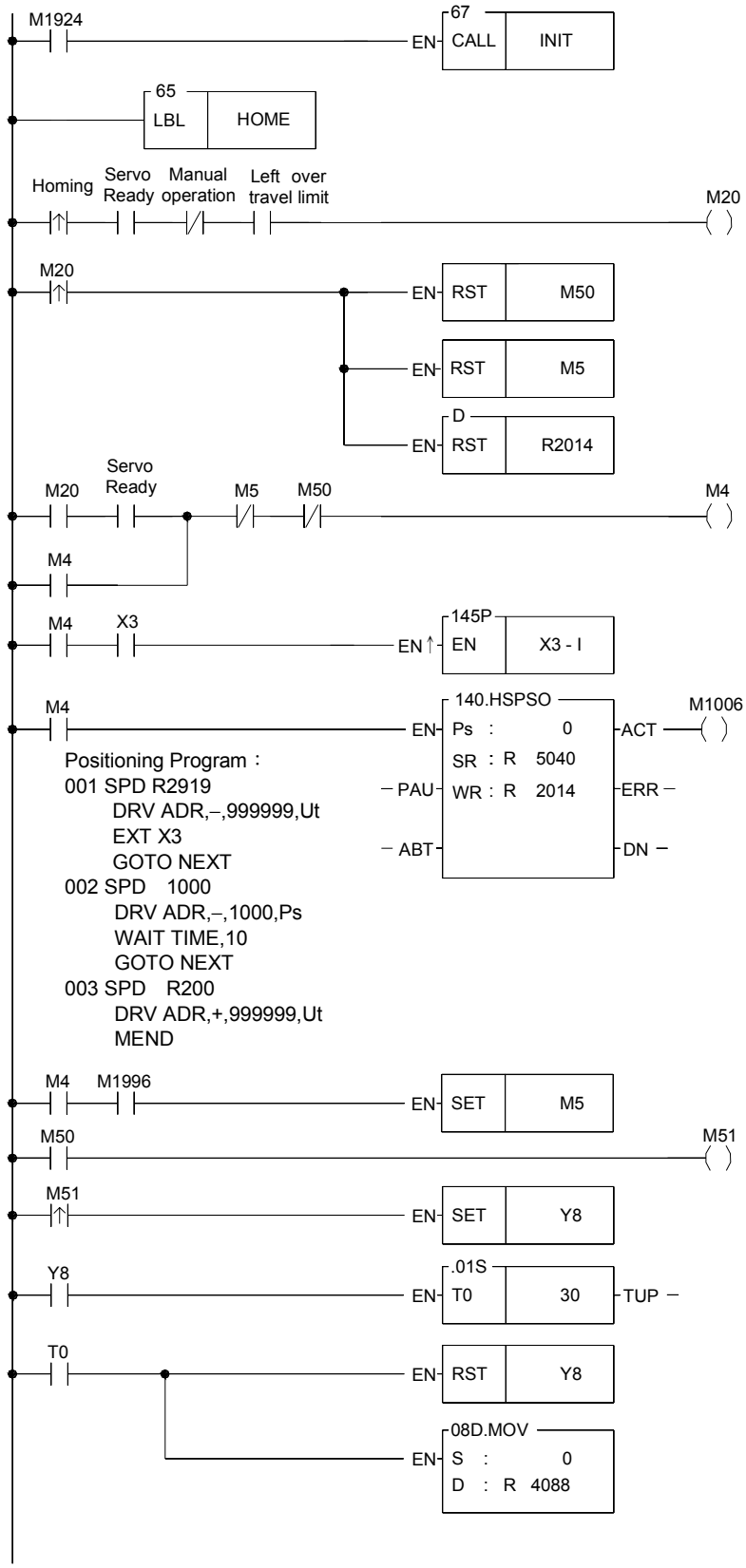



- X3 rising edge interrupt service subroutine.
- Enables HSC4 counting if homing.
- Interrupt service subroutine of HSC4 (Z phase counting is up)
- Stops pulse output immediately.
- Prohibits rising edge interrupt of X2.
- Output to clear error counter of servo driver.
- Sets the homing completion signal.
- Sends output immediately.

## Program Example 2: Machine homing (method 2)

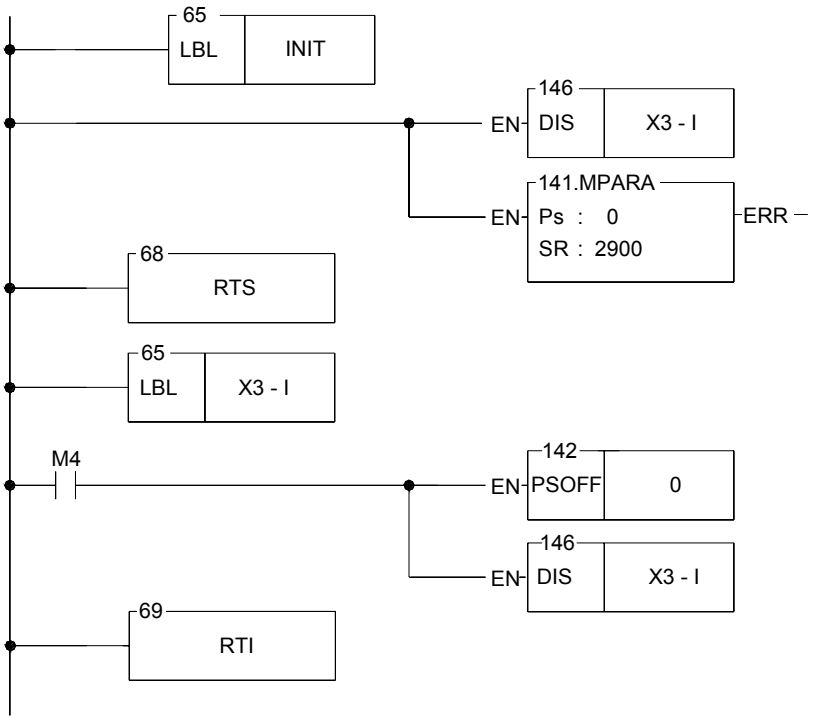
X3: Connected to near home sensing input, and configured as falling edge interrupt input.

【Main Program】



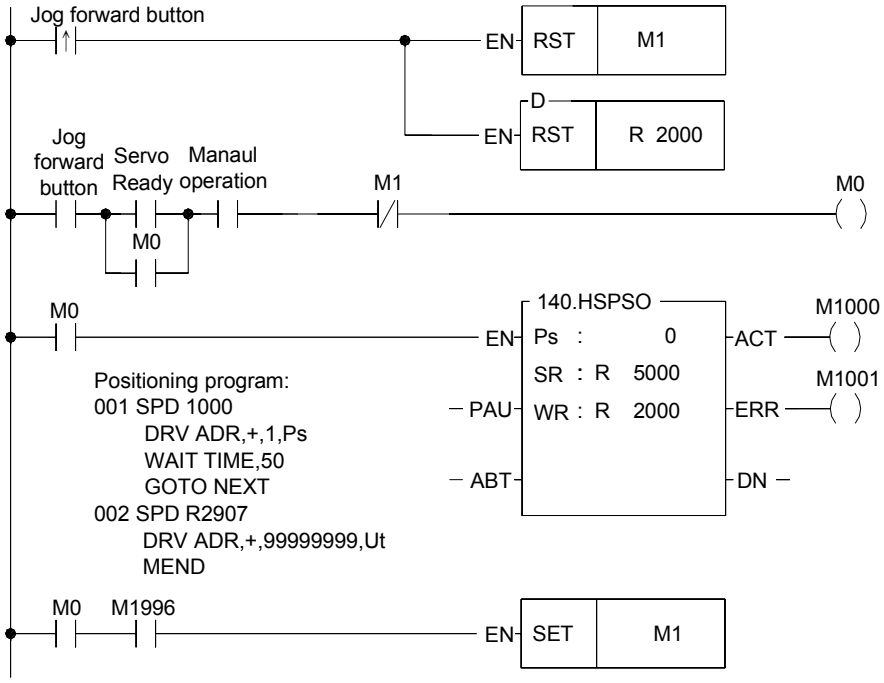

- Clears the homing completion signal.
- Clears the instruction completion signal for homing.
- Clears the step pointer, it starts from the first step to execute.
- Enable X3– (falling edge) interrupt.
- Configure R5000～R5199 to be the read only register (ROR) before programming, after then, when storing program, the Ladder program will automatically contains the positioning program.
- Homing instruction completed.
- Signal for homing completion.
- Output to clear error counter of servo driver -- Y8 is ON for 0.3 second.
- Fill the current PS registers with 0.

【Sub Program】



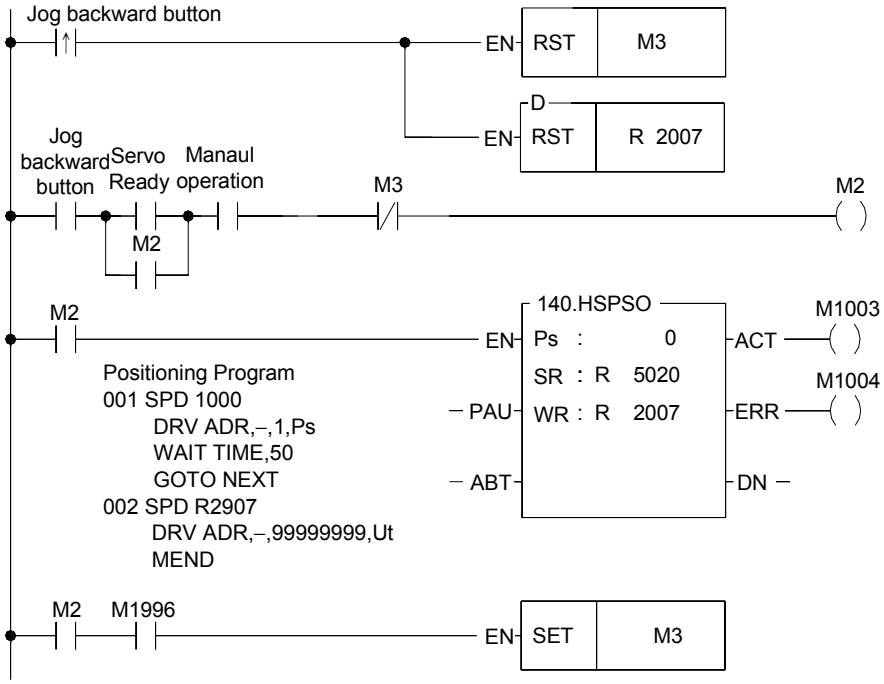

- Prohibits X3– interrupt
- Parameter table R2900～R2923
- X3 falling edge interrupt service subroutine.
- Stops pulse output immediately
- Prohibits X3– interrupt
- Sets the homing completion signal.

## Program Example 3: JOG Forward




- Clears the completion signal
- Starts from the first step every jog execution.
- As the execution of last step completed, it sets up the completion signal.

## Program Example 4: JOG Backward




- Clears the completion signal.
- Starts from the first step every jog execution.
- As the execution of the last step completed, it sets up the completion signal.

## Program Example 5: Step by step, One cycle, Continuous positioning control.

M93 : Start

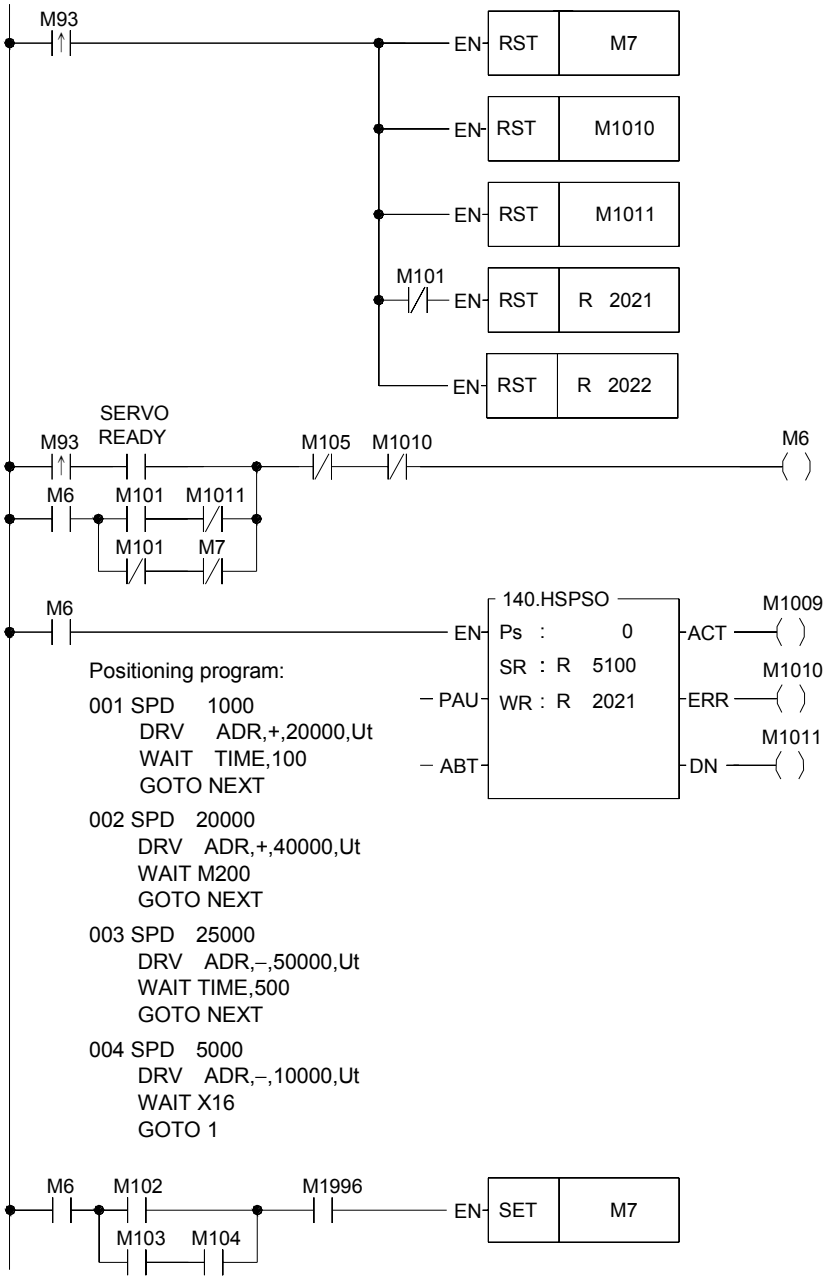
M101 : Step by step operation mode

M102 : One cycle operation mode

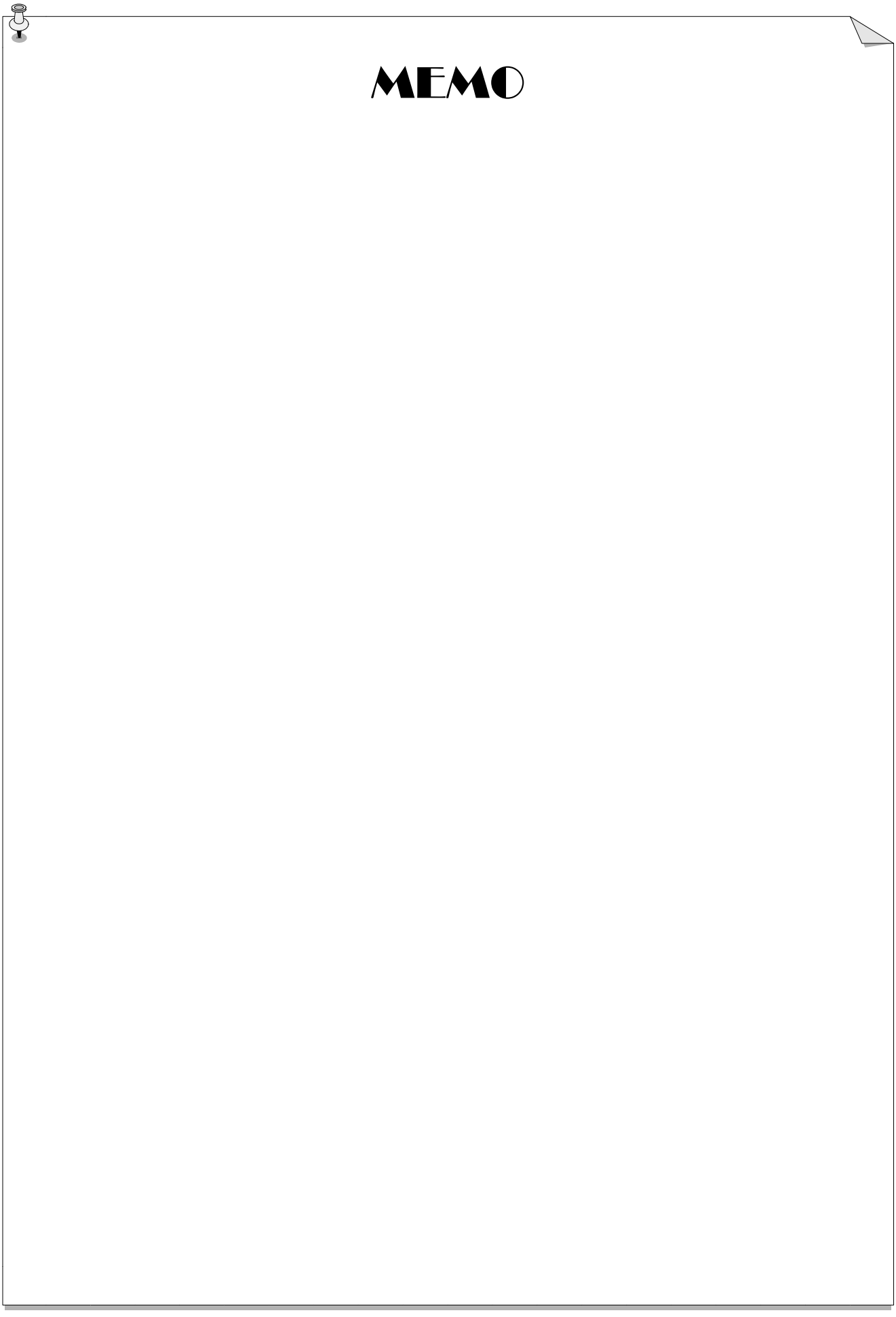
M103 : Continuous operation mode

M104 : Regular shut down.

M105 : Emergency stop.




- Clears shut down signal.
- Clears the error signal.
- Clears the step completion signal.
- Except step by step mode, the step pointer is cleared to be 0; it starts from the first step to execute.
- Clears being active bit of FUN140
- Set up the shut down signal.


MEMO