FBs-30GM

FBs-30GM Motion Controller
User Manual

V1.0.1

2014/4/25

FATEK Automation Corporation

Contents

Table

Figure

FBs-30GM Motion Controller User's Manual

1. Overview of FBs-30GM

   FBs-30GM is the 3-Axis Motion Control Module designed for FBs PLC series. With FBs-30GM, FBs PLC series can achieve circular interpolation, helical interpolation and other advanced motion control. Besides, FBs-30GM supports incremental rotary encoders and optical incremental linear encoders to implement precise close loop control. FBs-30GM adopts widely used G-code from standard RS274D to describe motion behavior. Pairing up with CAM software, FBs-30GM can help users in much more complicated motion control and dealing with applications in many aspects.

   1.1 Dimensions

   The dimensions of FBs-30GM as shown in 錯誤! 找不到參照來源。 below:

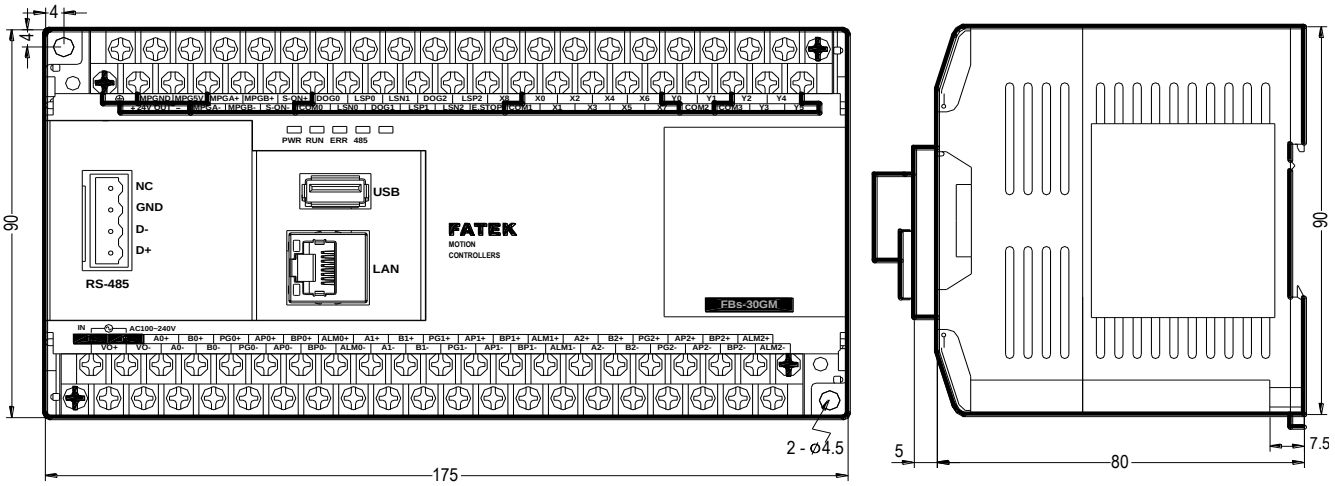


Figure 1: The dimensions of FBs-30GM

1.2 Composition and part names

錯誤! 找不到參照來源。 shows FBs-30GM's composition:

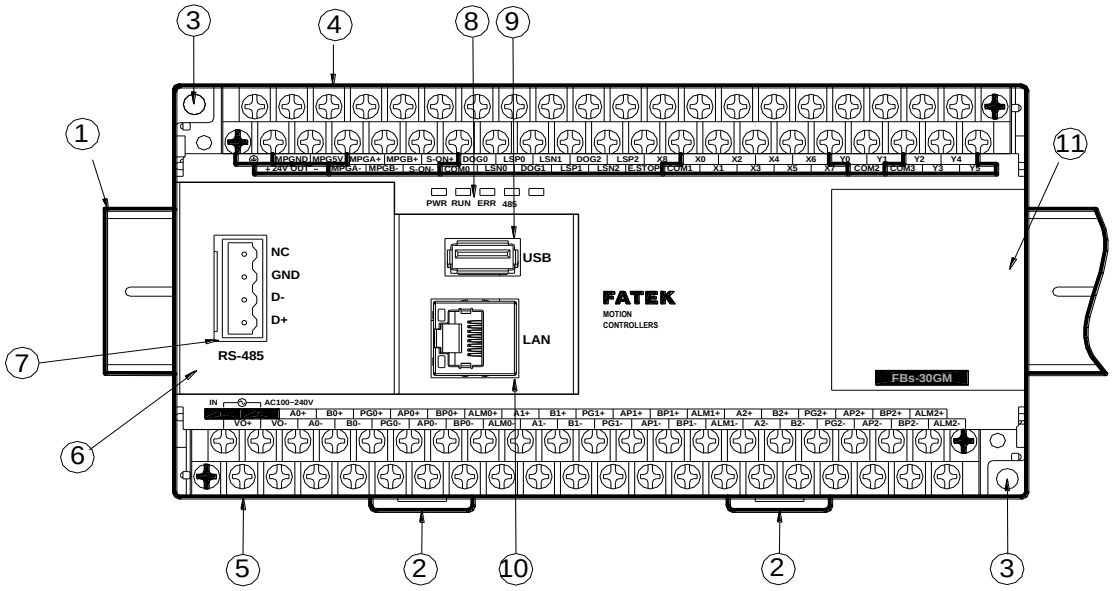


Figure 2: Front view of FBs-30GM

① 35mm-width DIN RAIL
② DIN RAIL tab
③ Hole for screw fixation (size: 4.5X2)
④ Terminals of 24VDC output and digital I/O terminals (Pitch 7.62mm)
⑤ Terminals of main power input and servo signals (Pitch 7.62mm)
⑥ Communication interface cover plate
⑦ RS-485 COM port
⑧ Status indicators
⑨ USB Host port
⑩ Ethernet RJ45 port
⑪ Right side cover plate

1.3 Status indicators

Table 1 shows the meaning of each status indicators.

Table 1: Status indicators

Name	Description
PWR	Green: FBs-30GM is connected to the ac power supply.
RUN	Yellow: System is ready. Blinking yellow: Motion program is processing.
ERR	Blinking red: Motion control kernel sends alarm message and has to suspend processing.
485	Yellow: RS485 communication success.
LAN	Green, LAN communication success. ◦

1.4 Terminals

Terminals and its descriptions are described as below.

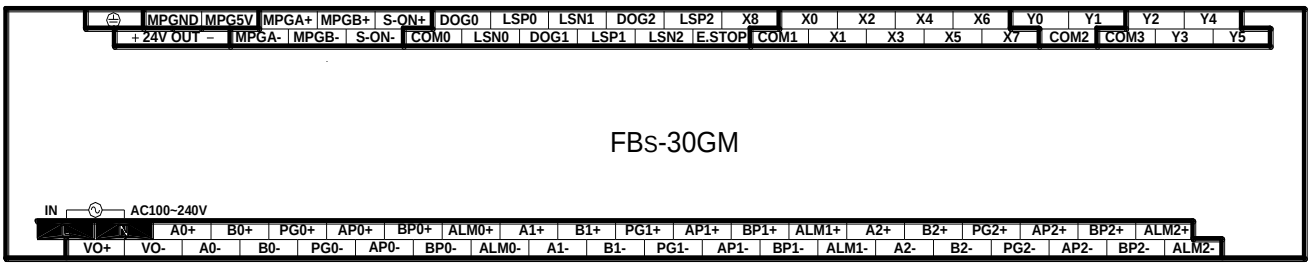


Figure 3: FBs-30GM terminals

Table 2: Upper terminal signals

Terminal	Description
⏚	Connect to PE (Protective Earth)
MPGND	The ground of MPG5V
MPG5V	5V DC output
+24V OUT-	24V DC output
MPGA(+/-)	Input of MPG hand wheel A-phase pulse
MPGB(+/-)	Input of MPG hand wheel B-phase pulse
S-ON(+/-)	System is all set and these two terminals become short-circuited (refer to FBs PLC's relay M1467)
DOG0 ~ 2	Near point signal input
LSP0 ~ 2	Limit Stroke of positive limit
LSN0 ~ 2	Limit Stroke of negative limit
E.STOP	Emergency stop, system will cease process and get into not-ready state when this signal is ON. Relay S-ON will be open (M1467 OFF) at the same time.
COM0	Common of DOG、LSP、LSN、E.STOP and X8 signals
X0 ~ X8	Digital input signals (refer to FBs PLC's relay M1480 ~ M1488)
COM1	Common of X0 ~ X7 signals
Y0 ~ Y5	Digital output signals (refer to FBs PLC's relay M1425 ~ M1430)
COM2	Common of Y0 ~ Y1 signals
COM3	Common of Y2 ~ Y5 signals

Table 3: Lower terminal signals

Terminal	Description
L, N	Main power input, 100 ~ 240 VAC, 50/60 Hz
VO(+/-)	Analog voltage output (controlled by D3435), range from -10V to +10V
A0(+,-) ~ A2(+,-)	A-phase feedback signals from encoder
B0(+,-) ~ B2(+,-)	B-phase feedback signals from encoder
PG0(+,-) ~ PG2(+,-)	Index signals from encoder
AP0(+,-) ~ AP2(+,-)	A-phase pulse signal outputs
BP0(+,-) ~ BP2(+,-)	B-phase pulse signal outputs
ALM0(+,-) ~ ALM2(+,-)	Axial alarm signals

2. Specification

Table 4: Power input/output specification

Power supply voltage	Main power voltage input 100 ~ 240 VAC, 50/60 Hz
Fuse capacity	2A/250 VAC
24VDC output current	24VDC output current up to 500mA
MPG5V output current	5VDC output current up to 250mA
Grounding	The diameter of grounding wire connected to PE shall not be less than that of L, N terminal of the power supply.

Table 5: Input signals

Terminal	Description	Max. input	
		Current	Voltage
MPGA+,MPGA-	Input of MPG hand wheel A-phase pulse (differential inputs)	15mA	5V
MPGB+,MPGB-	Input of MPG hand wheel B-phase pulse (differential inputs)	15mA	5V
DOG	Near point signal input	10mA	24V
LSP,LSN	Limit Stroke of positive and negative limit	10mA	24V
E.STOP	Emergency stop signal	10mA	24V
X0 ~ X8	Digital input signals, single-end sourcing input	10mA	24V
COM0	Common of DOG、LSP、LSN、E.STOP and X8 signals	110mA	0V
COM1	Common of X0 ~ X7 signals	80mA	0V

Table 6: Feedback signals

Item / Terminal	Description	Max. input	
		Current	Voltage
A+, A-	Axial feedback signal (500 kHz high speed digital signal input)	15mA	5V
B+, B-	Axial feedback signal (500 kHz high speed digital signal input)		
PG+, PG-	Encoder index signal (500 kHz high speed digital signal input)	15mA	5V
ALM+ , ALM-	Axial alarm feedback signal (low speed input)	10mA	24V

Table 7: Output signals

Item / Terminal	Description	Max. input	
		Current	Voltage
S-ON+,S-ON-	Relay output (after system start up, it switches to short-circuited)	1A	250 VAC 30VDC
AP+,AP-	Axial position control pulse signal	20mA	5V
BP+,BP-	Axial position control pulse signal	20mA	5V
Y0 ~ Y5	Digital output signal (photo coupler isolated output). Do not connect to any ac power source.	500mA	-
COM2/COM3	Common of Y0 ~ Y5 signals. Do not connect to any ac power source and connect a 2A fuse in series to ensure electrical circuit's safety.	1000mA	5 ~ 30V
VO+	Analog voltage output	10mA	+/-10V
VO-	Analog voltage output ground	10mA	0V

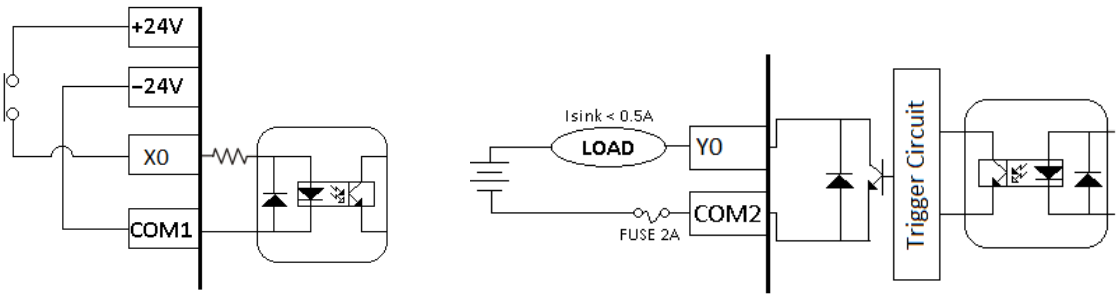


Figure 4: Input and output points wiring

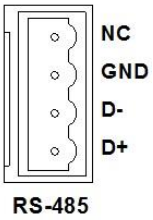


Figure 5: RS-485 COM port

Table 8: RS485 pin description

Pin	Description
NC	Not connected
GND	Ground
D-	Data-
D+	Data+

PLC connects to FBs-30GM with a specific port Port2 because it guarantees a 921600 high baud rate. Figure 6 takes FBs PLC-CB55 as example to illustrate how FBs PLC connects to FBs-30GM.

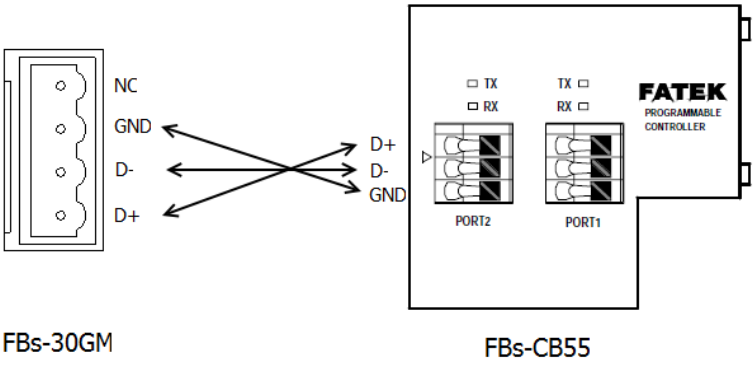


Figure 6: Connection between FBs PLC and FBs-30GM (with CB55)

⚠ **Warning! Please do not connect 24VDC ground and MPGND together. Otherwise it may cause internal hardware broken.**

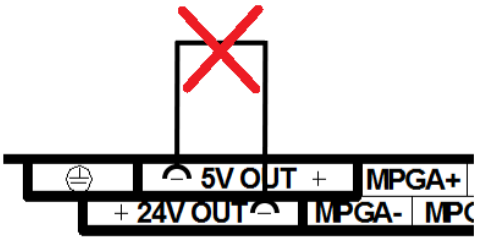


Figure 7: Improper wiring

Please use wires of 1.6mm and above for the grounding.

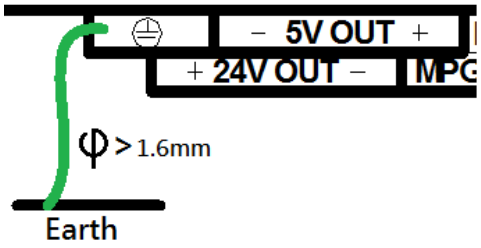


Figure 8: Selecting the grounding wire

⚠ **Never connect the AC main circuit power supply to any of the input/output terminals, as it will damage FBs-30GM. Check all the wiring prior to power up. To prevent any electromagnetic noise, make sure FBs-30GM is properly grounded. Do not touch the terminals when power on.**

3. Wiring

3.1 Wiring example with Yaskawa servo amplifier

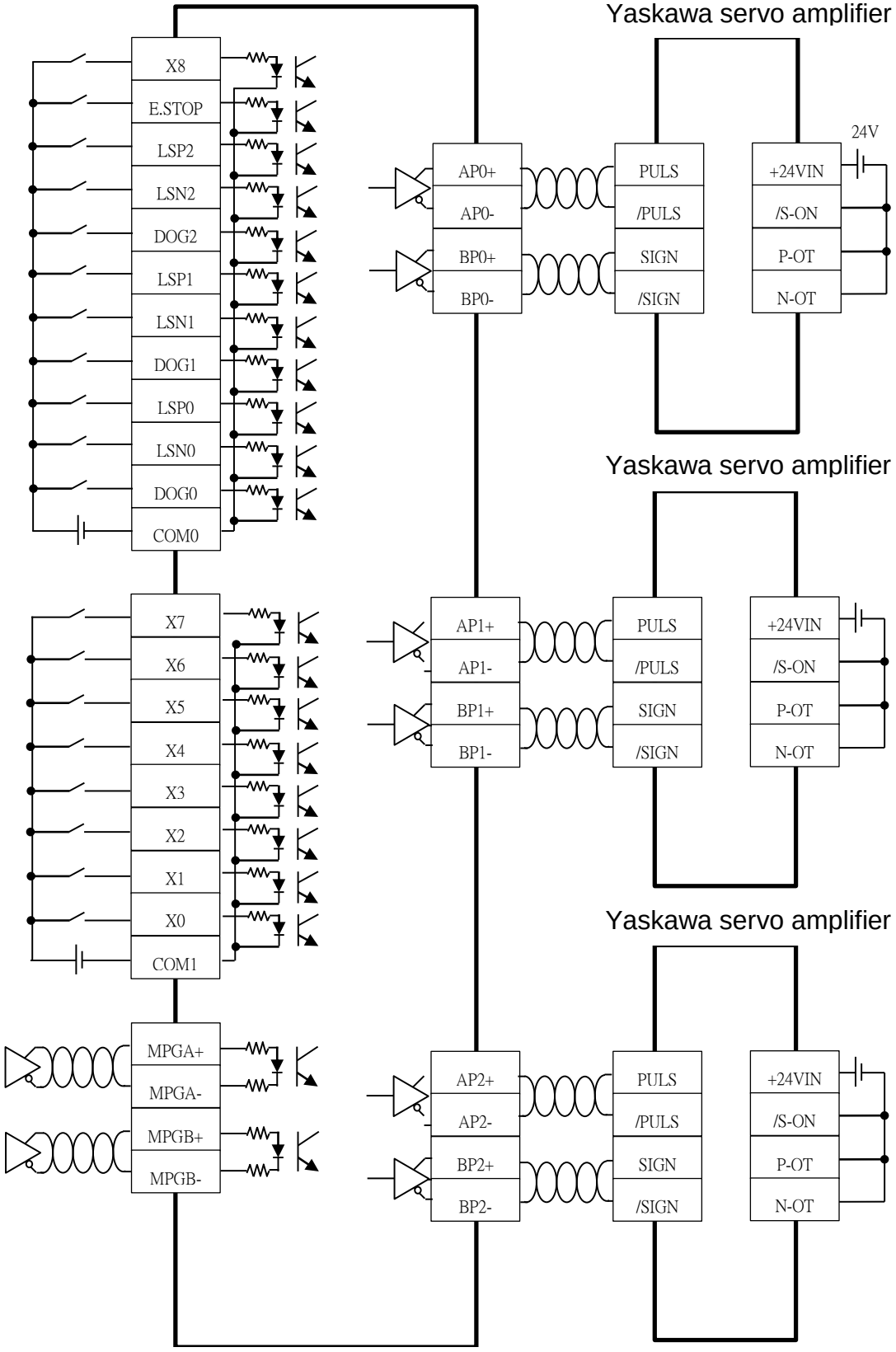


Figure 9: Wiring example with Yaskawa servo amplifier

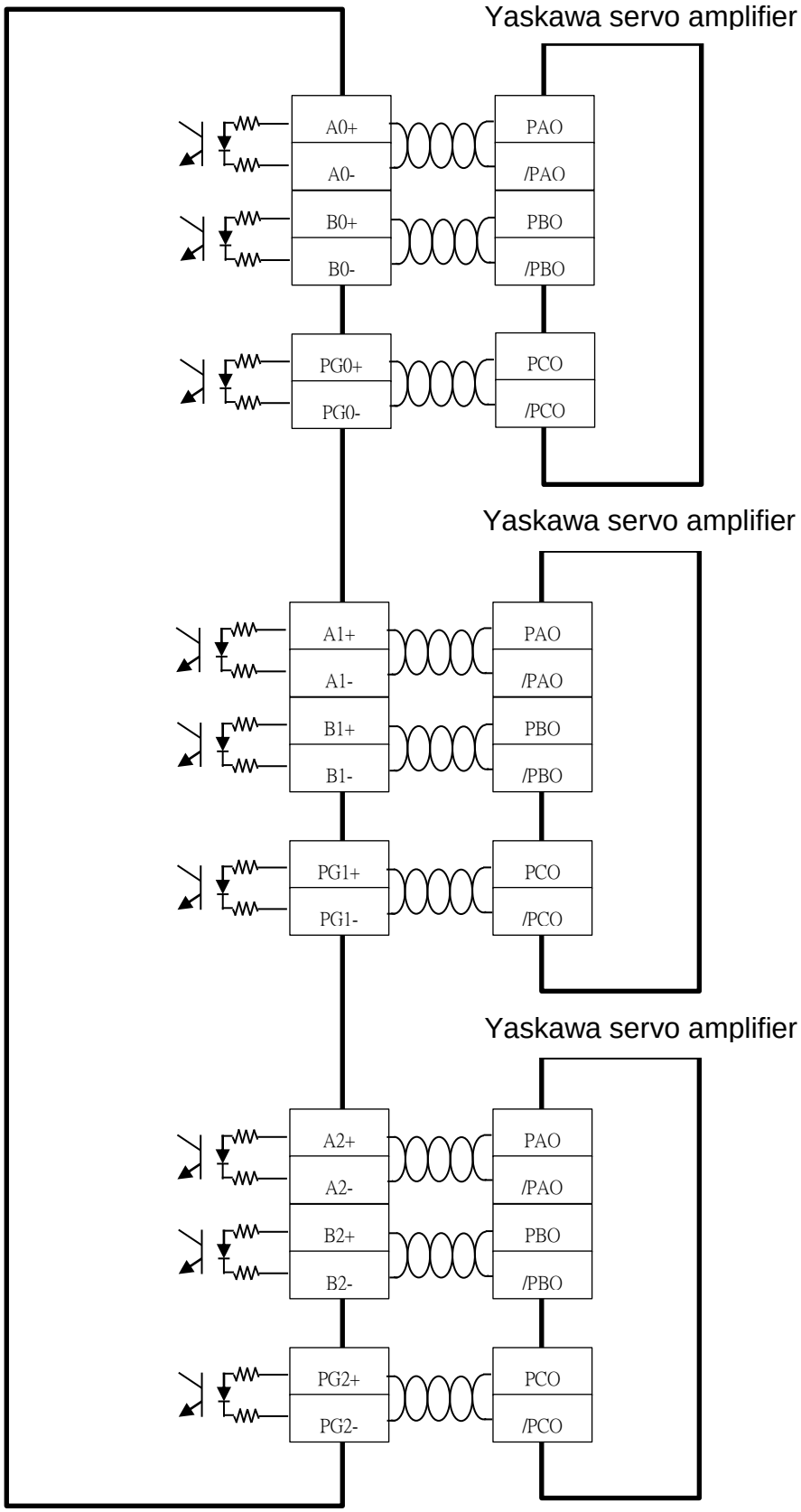


Figure 10: Connecting feedback signals from Yaskawa servo amplifier

3.2 Wiring example with Mitsubishi servo amplifier

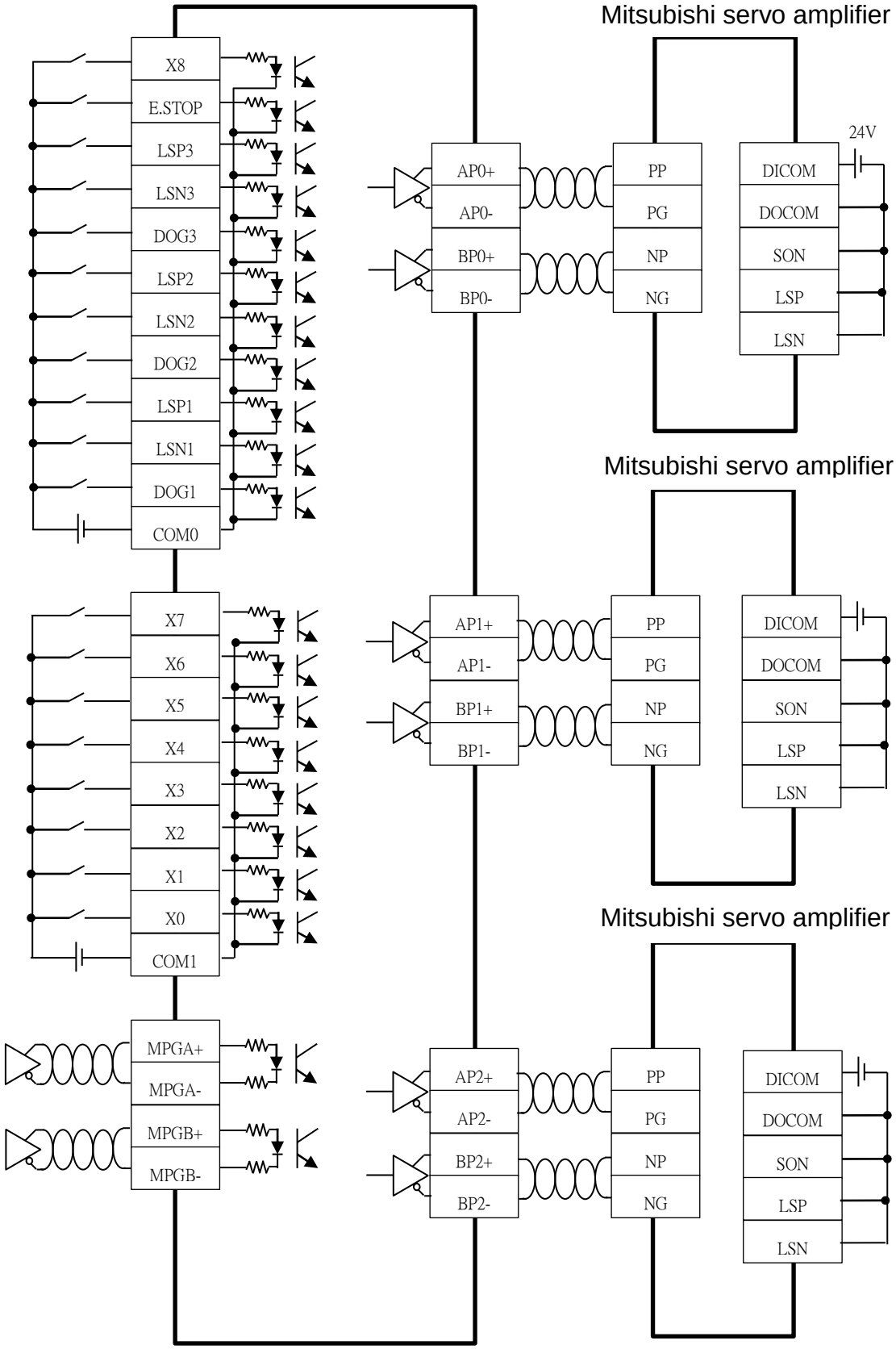


Figure 11: Wiring example with Mitsubishi servo amplifier

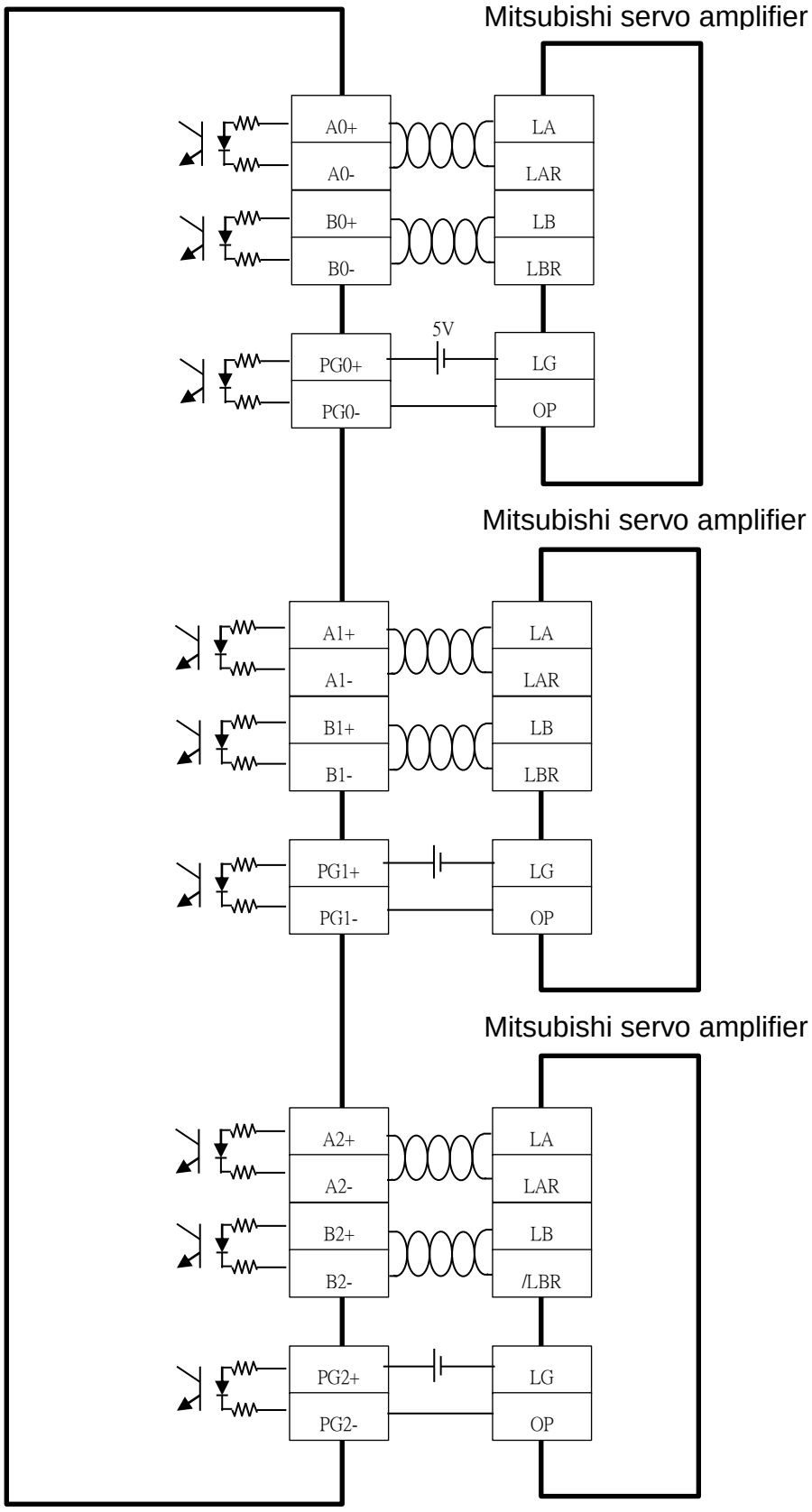


Figure 12: Connecting feedback signals from Mitsubishi servo amplifier

4. GMMon – monitor software

GMMon is the computer monitoring software for FBs-30GM. User can monitor the operating status of FBs-30GM by using GMMon. Installation is described in section 4.1. Section 4.2 is about setting up a connection. Section 4.3 is the introduction of GMMon.

4.1 GMMon Installation

Please follow the steps below to install GMMon.

1. Install Windows Update
2. Execute “MacroDev10.116.6C.msi” to install MacroDev
3. After the installation of MacroDev, execute “Fatek GMMon Setup.exe” to install GMMon

Please follow the steps below to install Windows Update:

i. VBSYS\setup.exe (necessary for 32-bit PC)
ii. vbrun60sp6.exe
iii. dotnetfx.exe
iv. msxml3.msi
v. VS2003LIB.msi
vi. dotnet2fx.exe (necessary for 32-bit PC)

Please restart the computer after the installation of Windows Update. If you encounter any error during the installation, please ignore it. Your computer might have higher version of software installed on it, so there is no need for you to install it,

Installation of MacroDev

Step1. Double-click the MacroDev setup file and then click “Next”.

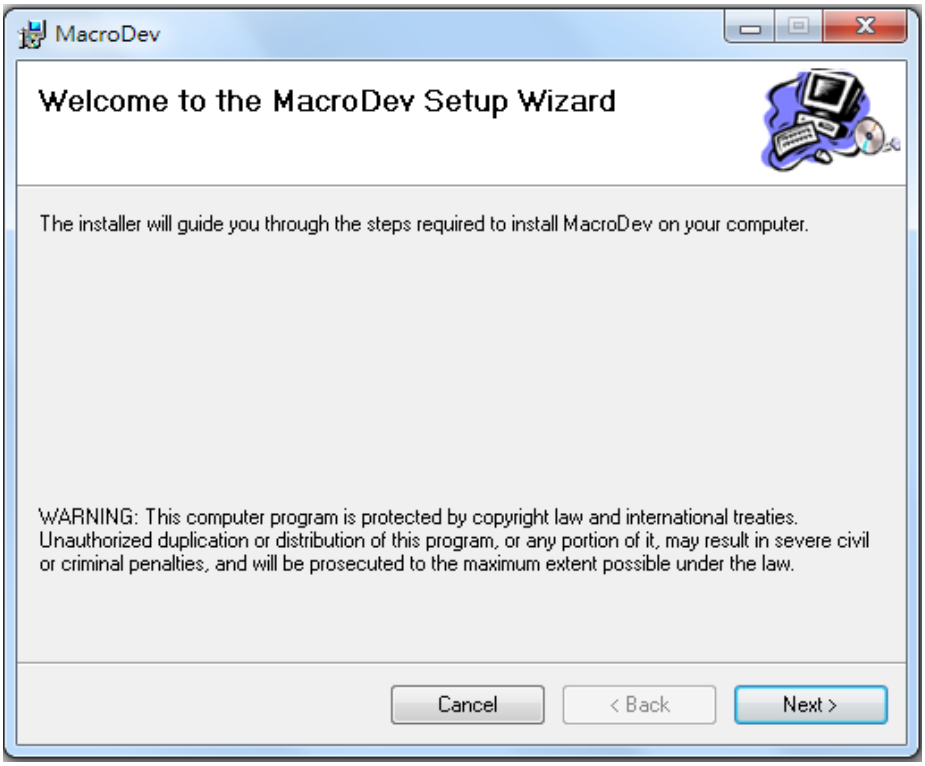


Figure 13: Step1 of MacroDev installation procedure

Step2. Select installation folder.

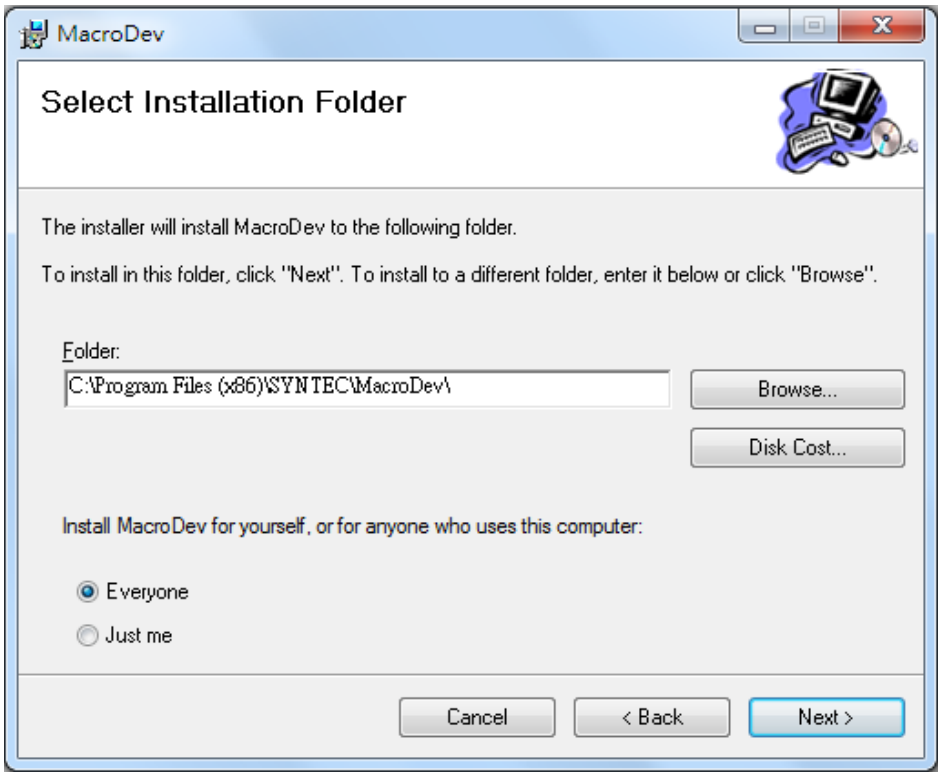


Figure 14: Step2 of MacroDev installation procedure

Step3. Confirm installation and then click “Next”.

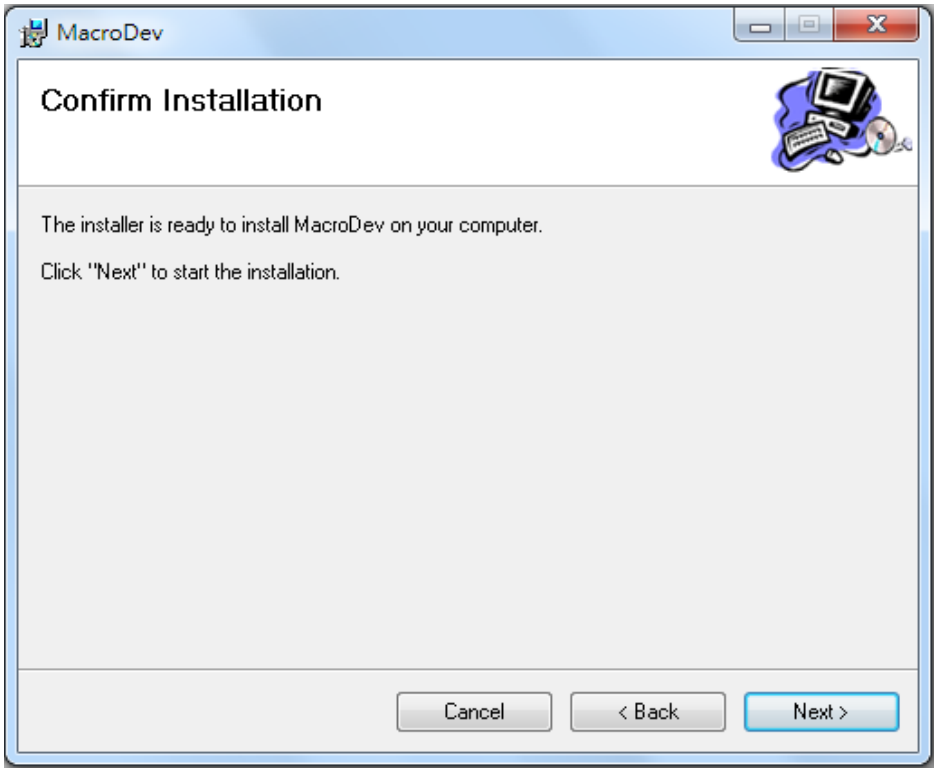


Figure 15: Step3 of MacroDev installation procedure

Step4. Installing MacroDev

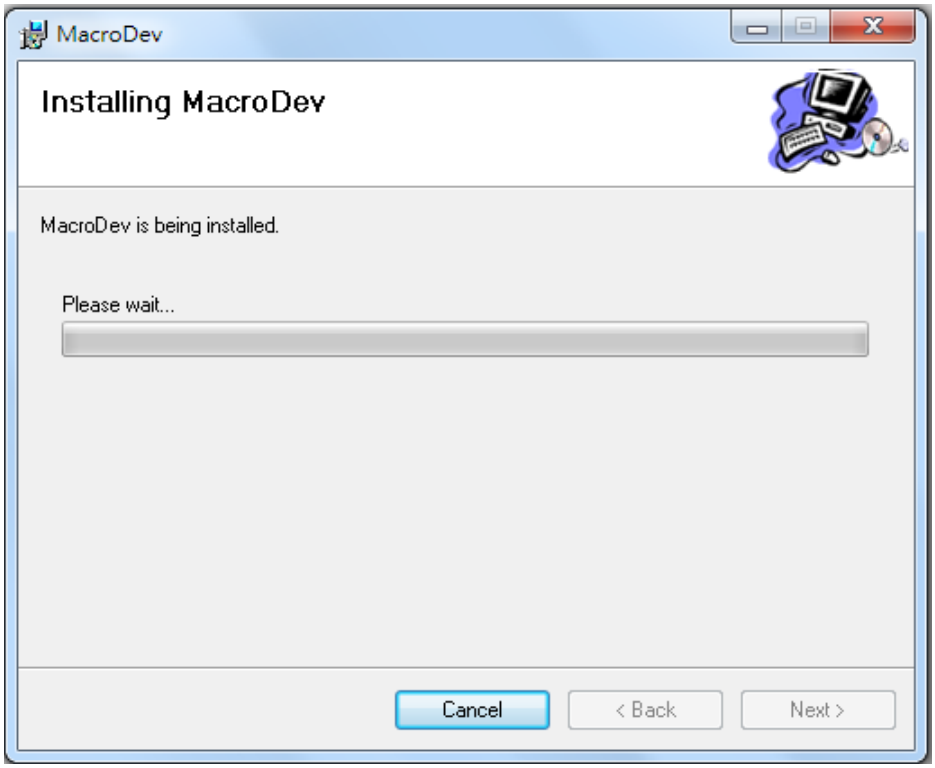


Figure 16: Step4 of MacroDev installation procedure

Step5. Installation is completed. Click “Close” to exit.

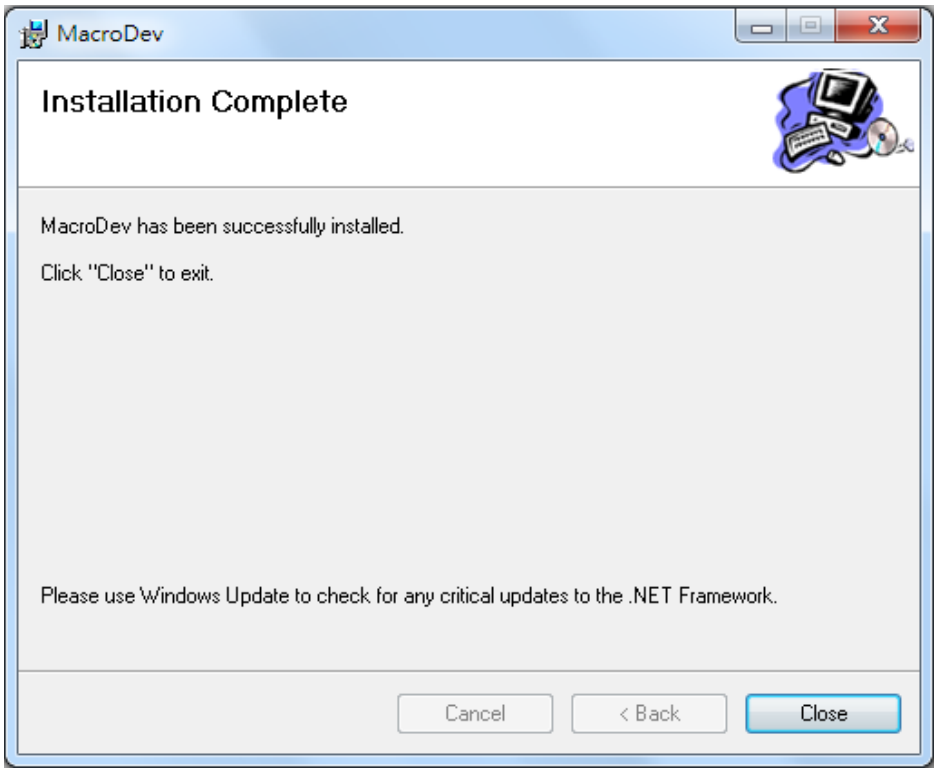


Figure 17: Step5 of MacroDev installation procedure

Installation of GMMon

Step1. Run “Fatek GMMon Setup.exe” and then click “Next”.

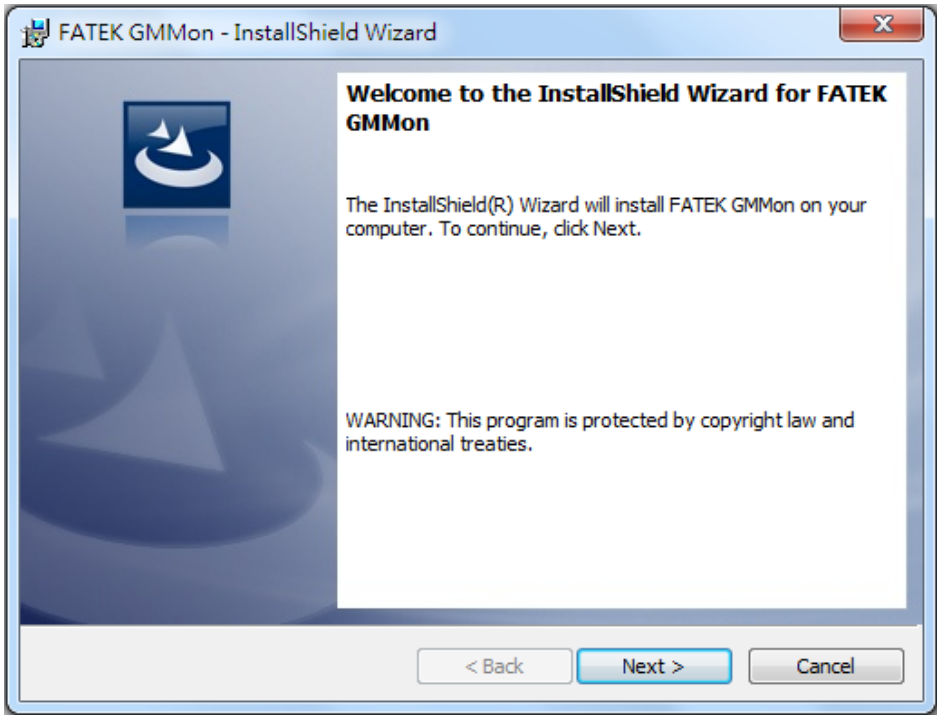


Figure 18: Step1 of FATEK GMMon installation procedure

Step2. Enter customer information.

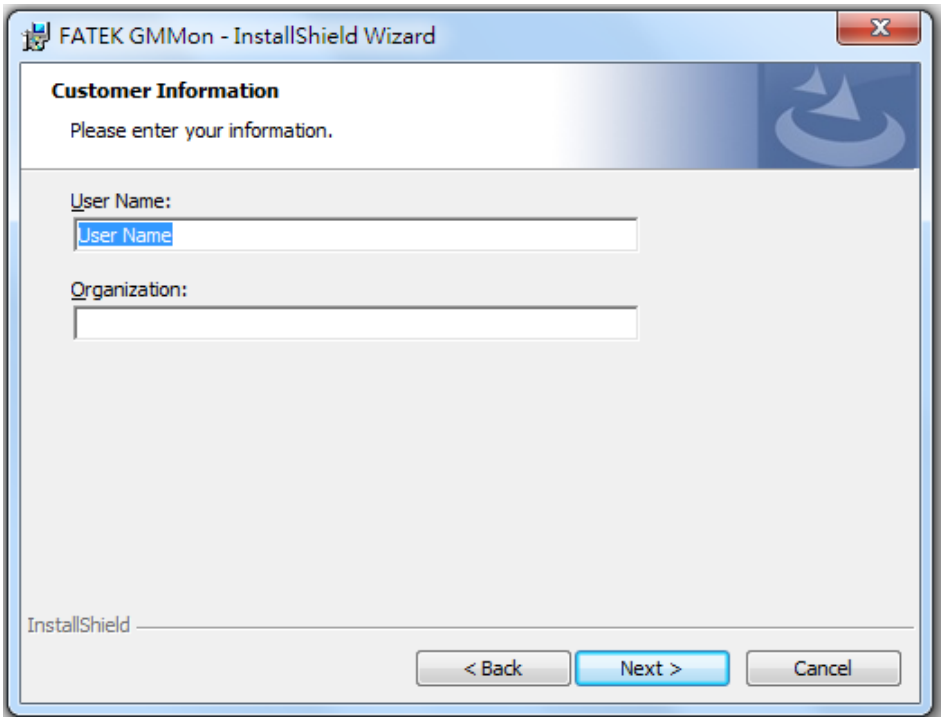


Figure 19: Step2 of FATEK GMMon installation procedure

Step3. Click “Install” to start Installation.

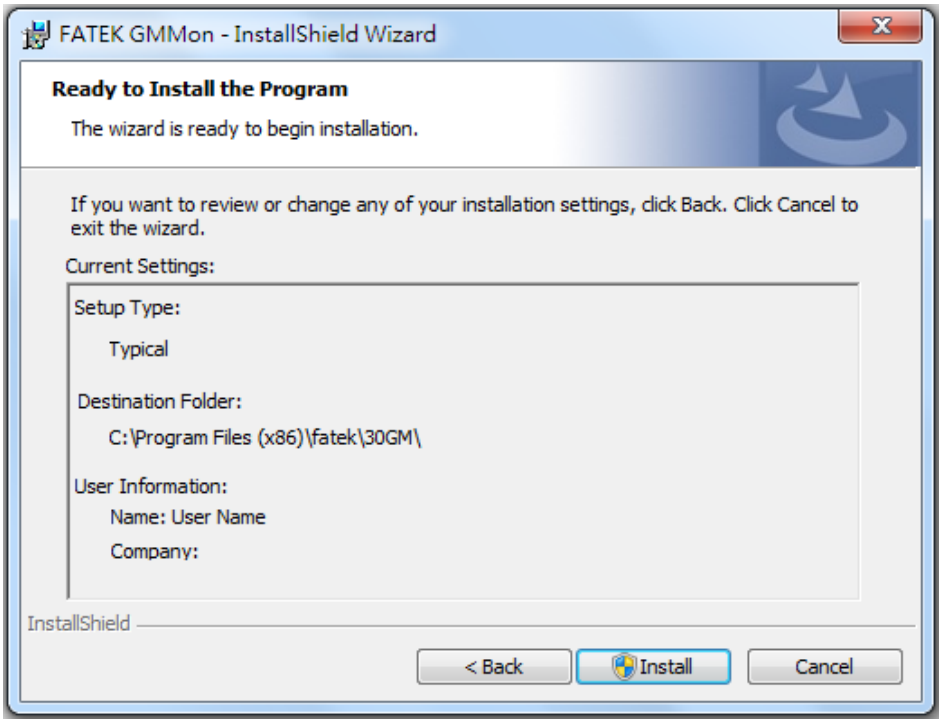


Figure 20: Step3 of FATEK GMMon installation procedure

Step4. Installing FATEK GMMon and waiting for the process bar to be completed.

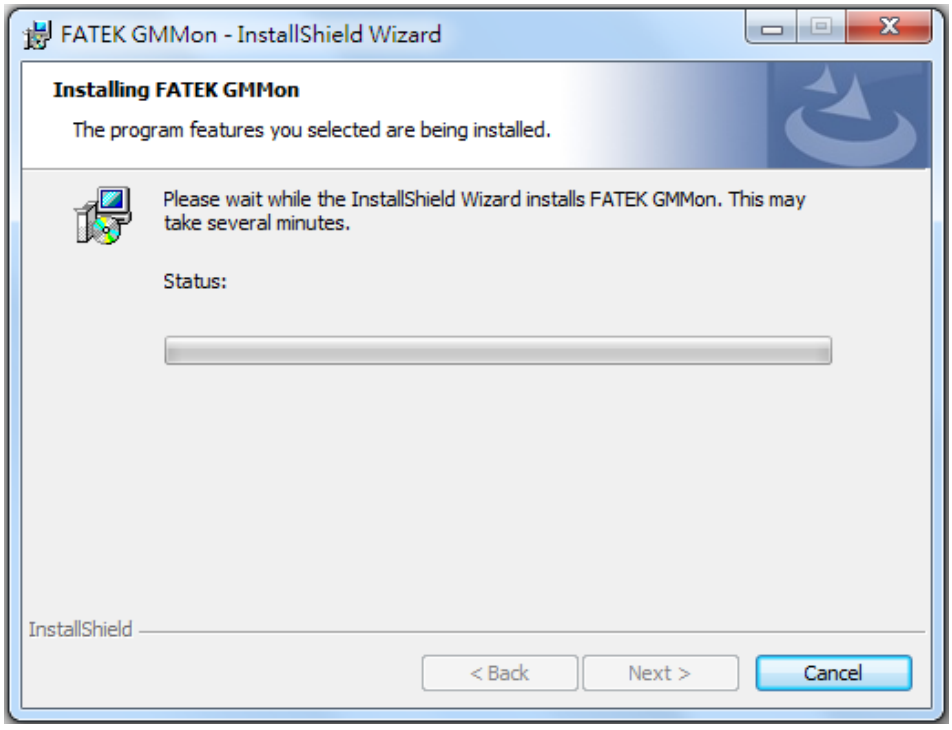


Figure 21: Step4 of FATEK GMMon installation procedure

Step5. Installation has been completed. Click “Finish” to exit.

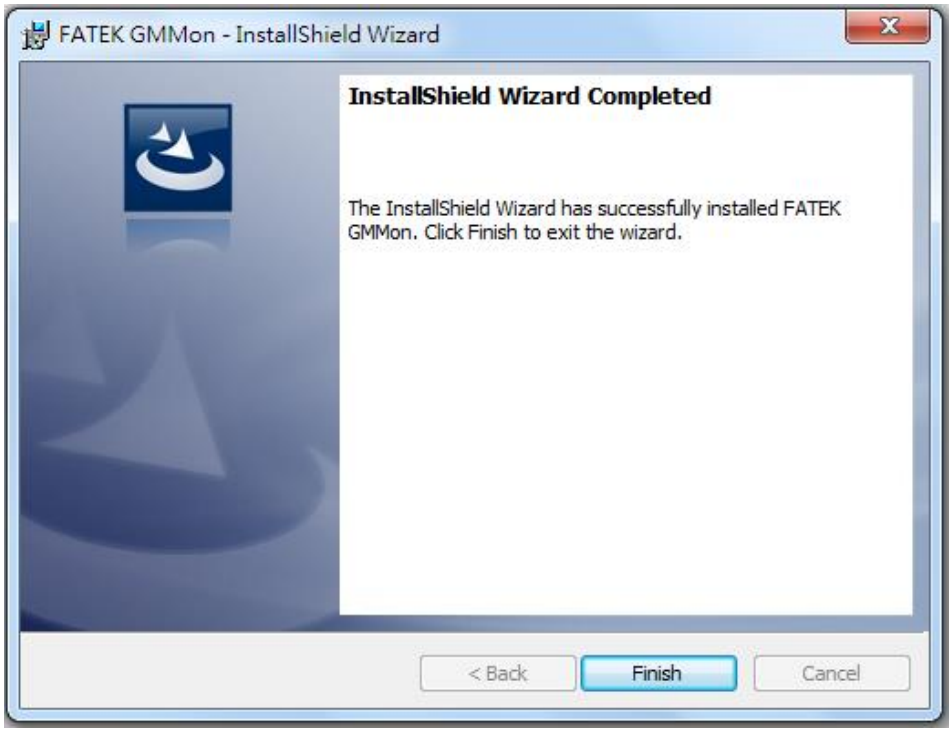


Figure 22: Step5 of FATEK GMMon installation procedure

4.2 Setting up a connection

4.2.1 Configure IP address

The default IP address in FBs-30GM is 192.168.10.10. The computer connected to FBs-30GM should have an IP address such as 192.168.10.XXX. If only one network interface card exist and the IP address is not 192.168.10.XXX, you can do the following steps to add a new IP address to your computer.

(PS: The computer and FBs-30GM should be in the same subnet, or your computer can connect to the network port of Fbs-30GM directly)

1. Go to Internet Protocol Version 4 (TCP/IPv4) Properties page and click “Advanced”.

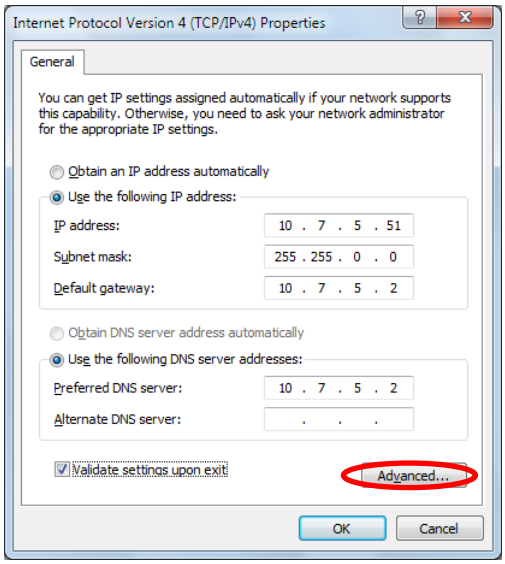


Figure 23: Internet Protocol Version 4 (TCP/IPv4) Properties

2. Click “Add” to add a new IP address as 192.168.10.XXX.

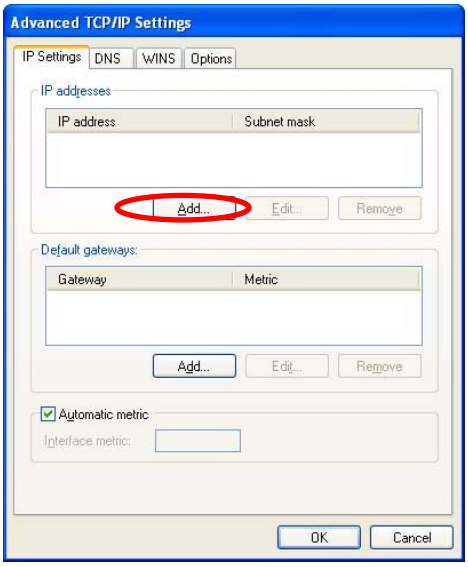


Figure 24: Add a new IP address

4.2.2 Change FBs-30GM's IP address

The default IP of FBs-30GM is 192.168.10.10. User can change its IP address with a USB flash drive by following the procedure below.

1. Prepare a USB flash drive
2. Create a file named "Setting0.ini" with the content below (take IP address "192.168.10.11" as example) and put this file in your USB root directory.

```
ACTION=SET_IP
PARAMETER=0,192.168.10.11,255.255.255.0,0,0,0
```

3. Insert the USB flash drive containing "Setting0.ini" to FBs-30GM.
4. Turn off FBs-30GM and on again, wait until RUN led is yellow: it means the system has finished restarting.
5. Pull out the USB and check its root directory. If a file named "Setting0.out" exists, it means that the IP address has been changed successfully.

⚠ **Note: When there exists "Setting0.out" file in the USB root directory before inserting the USB, FBs-30GM's IP address would not be modified. You have to delete "Setting0.out"**

4.3 Functions of GMMon

There are five main functions in GMMon, the System function, the Monitor function, the Simulate function, the Files function and the Debug function.

A. System: fill in the IP address of FBs-30GM to connect or disconnect. You can set the parameter or change the language.

B. Monitor: monitor the content and the graph illustrated by the motion program which is in process.

C. Simulate: Simulate a motion program on local PC without connection to FBs-30GM.

D. Files: manage motion program files.

E. Debug: you can use it for debugging parameters.

Monitor and Debug functions can only be operated when connecting to FBs-30GM, while Simulate and Files functions can only be operated when disconnecting to FBs-30GM.

4.3.1 System function page

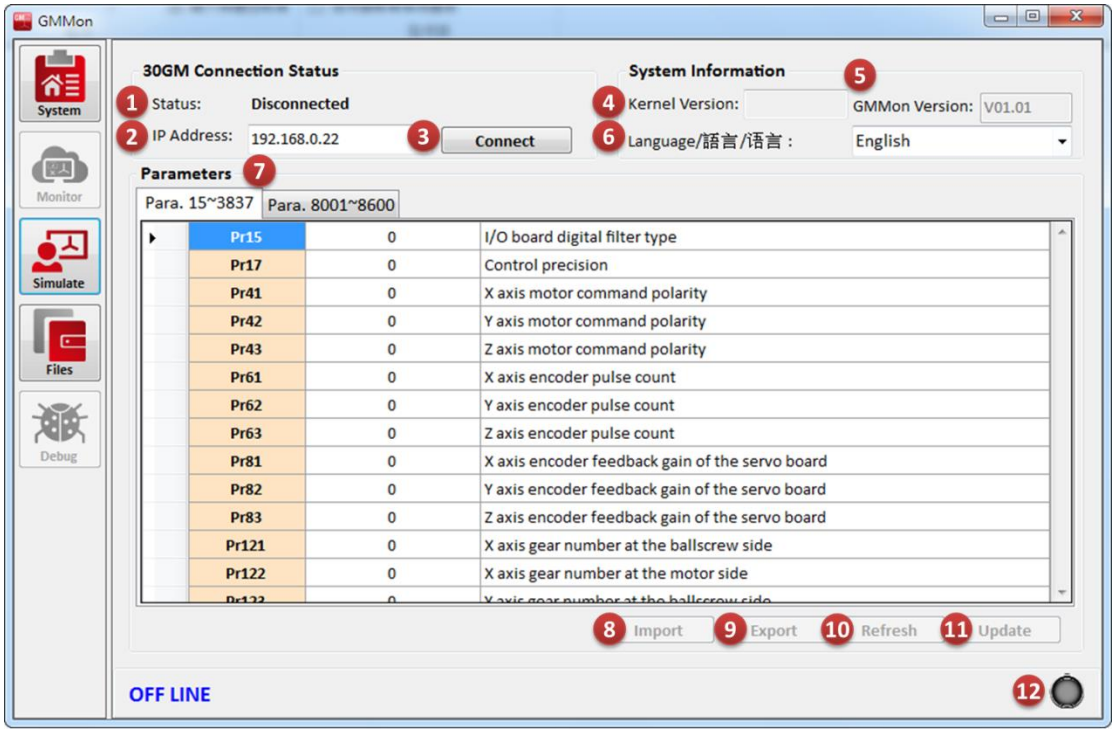


Figure 25: System function page

1. **Status:** ON LINE／OFF LINE status
2. **IP Address:** Input IP address of the FBs-30GM to connect
3. **Connect／Disconnect:** get connected／disconnected

4. **Kernel Version:** kernel version number of FBs-30GM
5. **GMMon Version:** GMMon software version number
6. **Language/語言/语言 :** Change the language of GMMon
7. **Parameters:** list of FBs-30GM's operating parameters
8. **Import:** import the parameter configuration file
9. **Export:** Export the parameters configuration to a file
10. **Refresh:** refresh the page to see the current value of FBs-30GM parameters
11. **Update:** update FBs-30GM parameters
12. **Connection indicator:** green light blinks when FBs-30GM is connected or red light blinks when alarm happens.

4.3.2 Monitor function page

After connecting to FBs-30GM, use can use Monitor function.

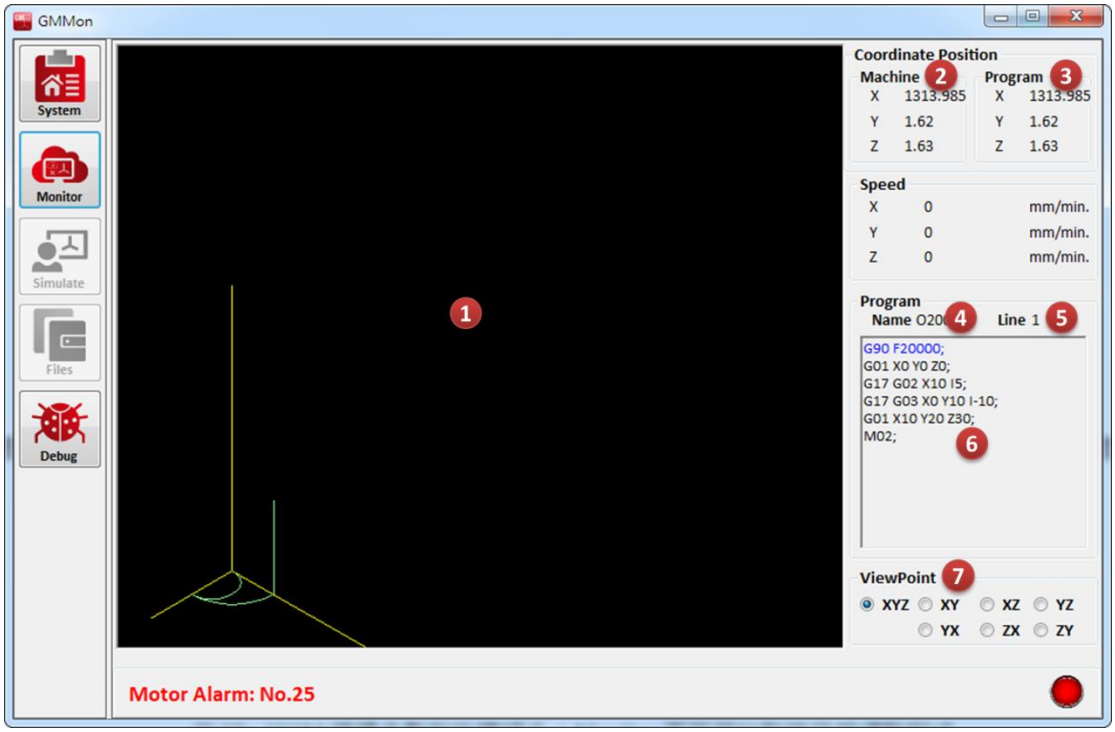


Figure 26: Monitor function page

1. **Monitoring screen:** According to the motion program file, the locus will be drawn on this screen and user also can foresee the future locus.
2. **Machine:** current coordinate values of machine
3. **Program:** current coordinate values of program
4. **Program Name:** motion program name

5. **Line:** the motion program line number which is in process
6. **Program content:** display the content of the motion program, and the line in blue means it is in progress
7. **ViewPoint:** select one of the seven coordinate systems such as XYZ space, XY plane, XZ plane, YZ plane, YX plane, ZX plane and ZY plane

4.3.3 Simulate function page

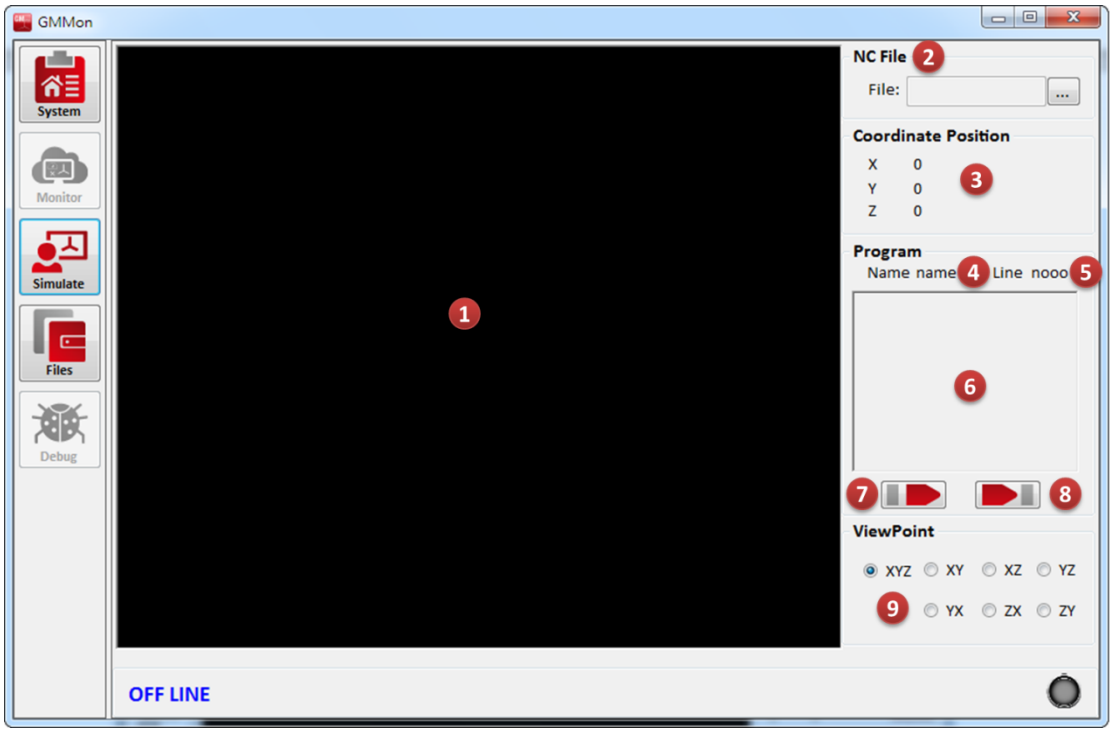


Figure 27: Simulate function page

1. **Simulation Result:** For user to check if the program is correct, it draws the trace according to the selected motion program.
2. **NC Files:** select the program which is going to be simulated
3. **Coordinate Position:** display the current simulation coordinates
4. **Program Name:** the program name of the selected program
5. **Line:** the motion program line number which is in simulation
6. **Program Content:** display the content of the simulated motion program, and the blue line has just being simulated
7. **Play:** simulate all the content of the motion program
8. **Step:** simulate one line of the motion program at a time

9. **ViewPoint:** select one of the seven coordinate systems such as XYZ space, XY plane, XZ plane, YZ plane, YX plane, ZX plane and ZY plane

4.3.4 Files function page

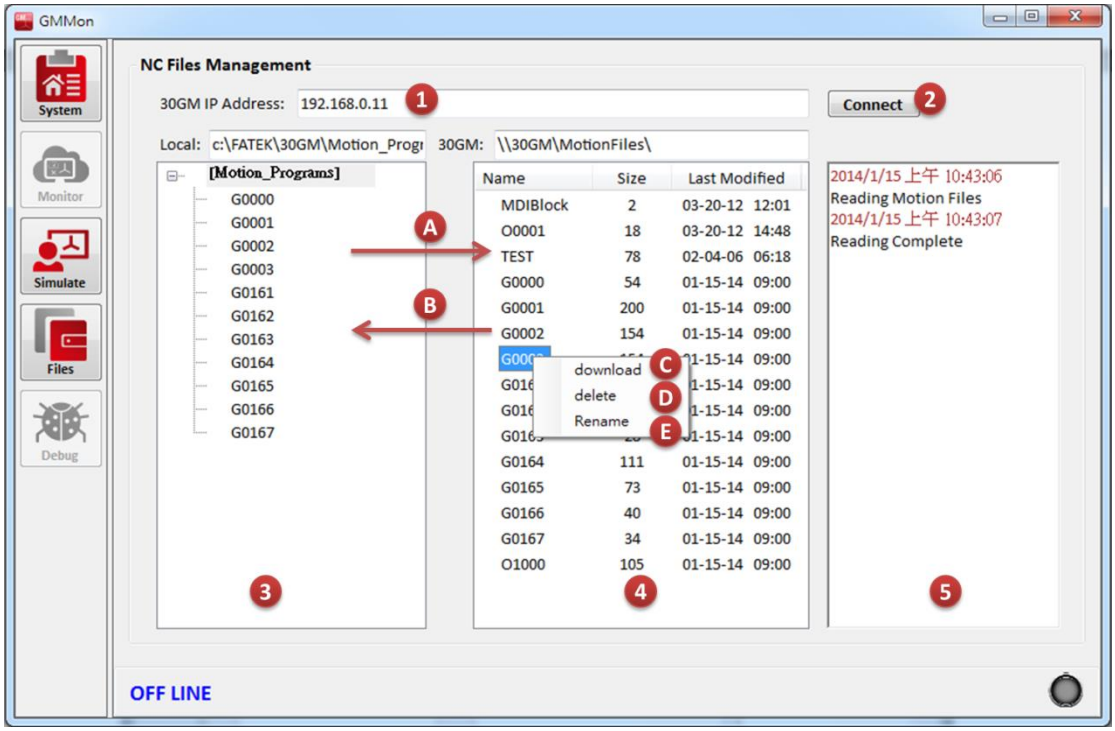


Figure 28: Files function page

1. **FBs-30GM IP Address:** enter IP address of the FBs-30GM to connect
2. **Connect:** get connected
3. **Local:** the motion program will be put in the local path C:\FATEK\30GM\Motion_Programs
4. **30GM:** the path of motion program on FBs-30GM
5. **Log message:** this displays log message of file management

A. Upload:

Drag and drop the file from Local to 30GM.

B. Download:

Drag and drop the file from 30GM to Local.

C. Download: Right click the mouse button to the file and select download.

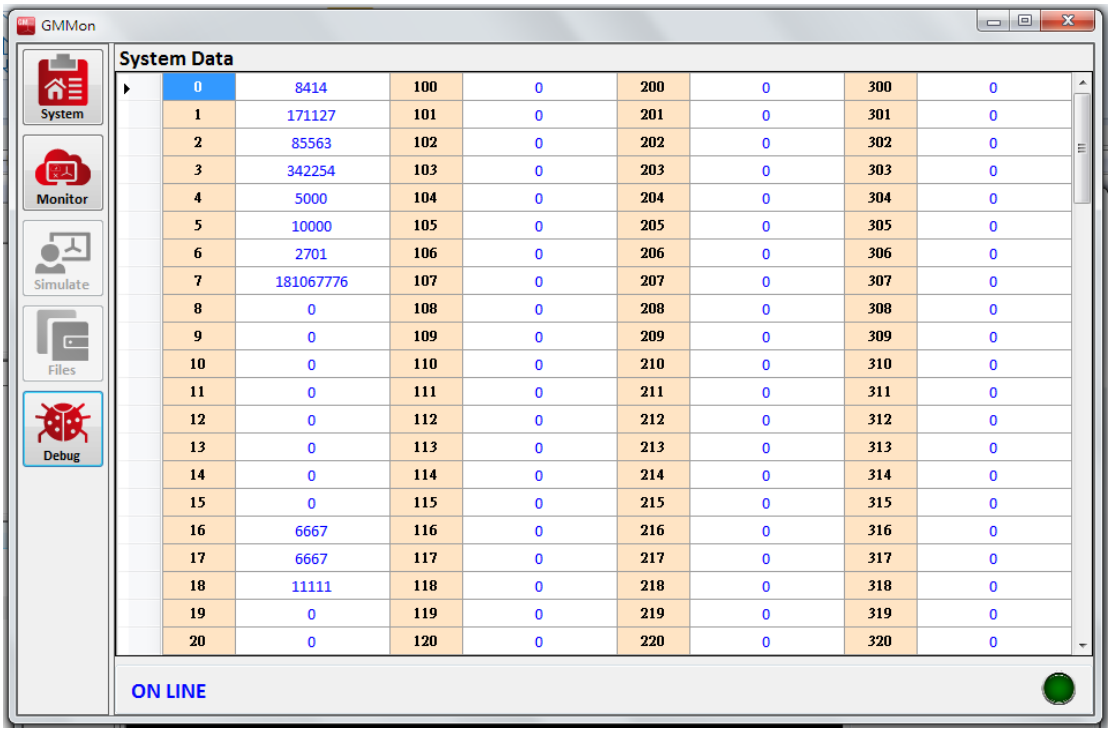
D. Delete:

Right click the mouse button to the file and select delete.

E. Rename:

Right click the mouse button to the file and select rename.

4.3.5 Debug function page



0	8414	100	0	200	0	300	0
1	171127	101	0	201	0	301	0
2	85563	102	0	202	0	302	0
3	342254	103	0	203	0	303	0
4	5000	104	0	204	0	304	0
5	10000	105	0	205	0	305	0
6	2701	106	0	206	0	306	0
7	181067776	107	0	207	0	307	0
8	0	108	0	208	0	308	0
9	0	109	0	209	0	309	0
10	0	110	0	210	0	310	0
11	0	111	0	211	0	311	0
12	0	112	0	212	0	312	0
13	0	113	0	213	0	313	0
14	0	114	0	214	0	314	0
15	0	115	0	215	0	315	0
16	6667	116	0	216	0	316	0
17	6667	117	0	217	0	317	0
18	11111	118	0	218	0	318	0
19	0	119	0	219	0	319	0
20	0	120	0	220	0	320	0

Figure 29: Debug function page

[8 ~ 10]: X/Y/Z axis following error value

[Definition]: The error amounts between axial position command values and feedback values, and is calculated as below.

X/Y/Z axis following error value =

Absolute position command value - Absolute position feedback value

Unit: BLU

[Description]:

1. These variables are the current amounts of axial tracking errors, used to check the amounts of errors between axial position command values and feedback values.
2. When the axis is stationary, the error amount at this time is called static error and in theory is almost equal to 0. If it is greater than Pr561 ~ Pr563 for X, Y and Z-axis, FBs-30GM will send alarm MOT-008.
3. When axes are moving, the error amounts at this time are called dynamic errors and in theory should be less than the maximum allowable amount of following

error values 16 ~ 18. Otherwise, FBs-30GM will send alarm MOT-019 or MOT-023.

4. When feedrate override is uniform, these variables should be almost equal to debug variables 32 ~ 34. Otherwise, please check the position control loop gain of the servo driver is the same as Pr181 ~. It may also be caused by enabled feed-forward or command filter function of servo driver. Of course, abnormal wire connection may cause the inconsistencies between debug variables 8 ~ 10 and 32 ~ 34.

[24 ~ 26]: X/Y/Z axis absolute position feedback value
[Definition]: The axial position control feedback of the motors
Unit: BLU
[Description]:

1. For non-absolute encoder, these variables will be set to zero after the first reference searching is completed.

[40 ~ 42]: X/Y/Z axis absolute position command value
[Definition]: Cumulative command pulses sent by FBs-30GM
Unit: BLU
[Description]:

1. These variables are the amounts of position commands sent by FBs-30GM and is not necessary exactly equal to debug variables 72 ~ 74 (machine coordinates) because these variables also include mechanical compensations (such as backlash, sharp, pitch and temperature).
2. For non-absolute encoder, this variable will be set to zero after the first reference searching is completed.

[48 ~ 50]: X/Y/Z axis motor index counter
[Definition]: The number of pulses is recorded when the motor index feedback signal of each axis is generated.
[Description]:

1. Theoretically updated increments of these variables each time have to be equal to Pr61 ~ Pr63, and if not, which means that the hardware may lose pulses. Please check the feedback signal (A +, A-, B +, B-, C +, C-) wiring are off or if it is affected by noise.
2. For non-absolute encoder, this variable will be set to zero after the first reference searching is completed.

Table 9: Debug variables

Debug variables			
8	X axis following error value	40	X axis absolute position command value
9	Y axis following error value	41	Y axis absolute position command value
10	Z axis following error value	42	Z axis absolute position command value
24	X axis absolute position feedback value	48	X axis motor index counter
25	Y axis absolute position feedback value	49	Y axis motor index counter
26	Z axis absolute position feedback value	50	Z axis motor index counter

Other diagnostic variables are for internal use only.

5. Operate and execute motion programs

In addition to operating FBs-30GM, FBs-PLC can monitor the input states and control the output states of 30GM. Please refer to Appendix I Special relays and interface registers of FBs-PLC.

5.1 Relation between FBs PLC and FBs-30GM

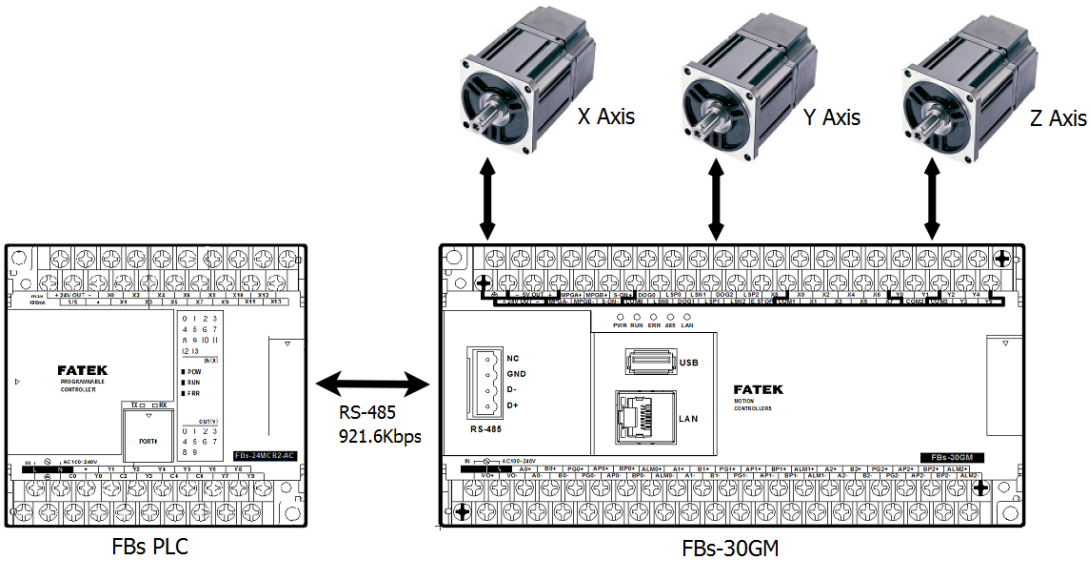


Figure 30: Relation between FBs PLC and FBs-30GM

FBs-30GM cannot run independently and must work with FBs PLC. After FBs PLC sends commands through RS-485 to 30GM, 30GM acts correspondingly.

5.2 Procedure to execute a motion program

5.2.1 Upload the motion program to FBs-30GM

Use Notepad or other text editors to edit a motion program. Upload the motion program to FBs-30GM.

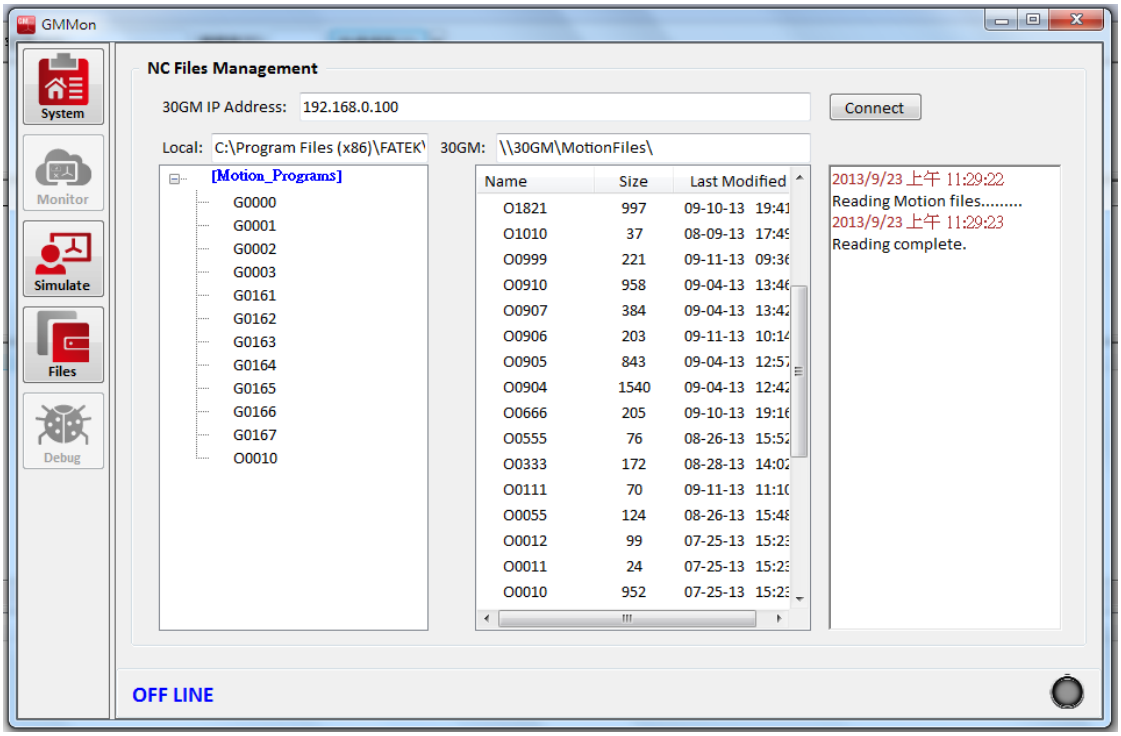


Figure 31: GMMon Files function

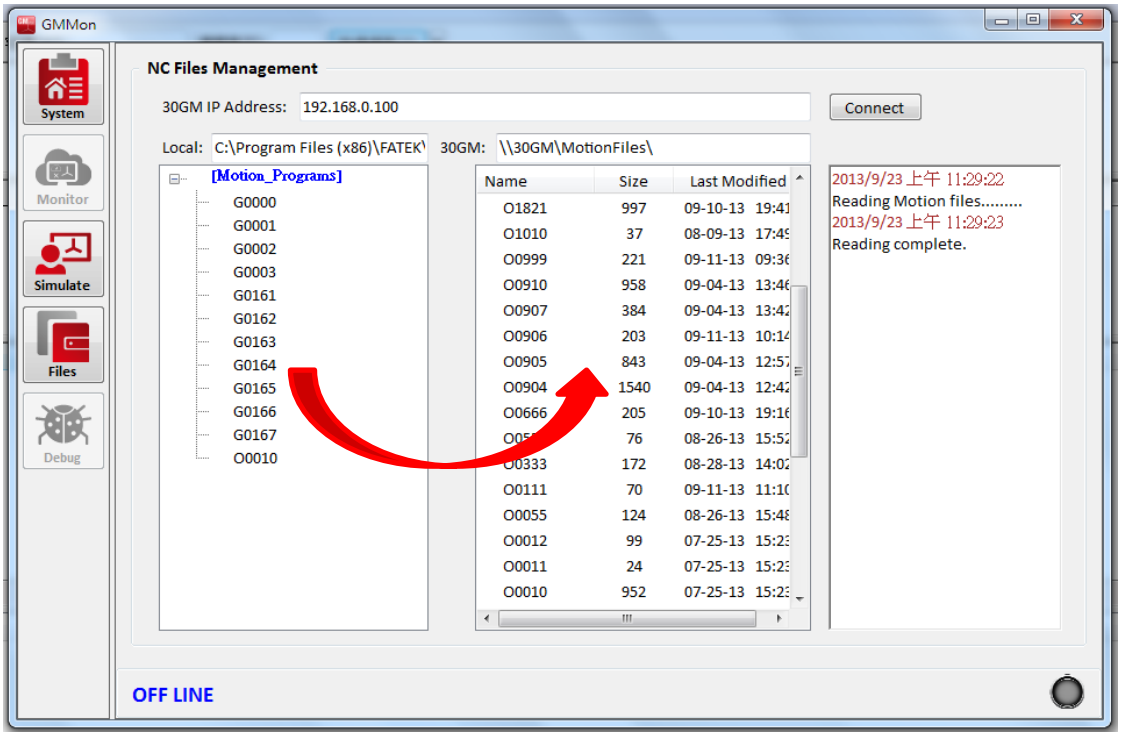


Figure 32: Drag and drop the file to upload

Motion program naming rule:

FBs PLC assigns the motion program to 30GM by setting the register D3431. Therefore, the file name of the motion program must follow the naming format below, so FBs-30GM is able to identify the designated motion program.

Motion program naming format:

A. Four digits come after an uppercase O.

B. If the digits are less than four, left pad zeroes to four digits.

C. The four-digit number ranges from 1 to 9999.
(Out of this range may cause unpredictable results)

Examples:

- Number 1 : O0001
- Number 456 : O0456
- Number 7156 : O7156
- Unqualified file name : O-1234、O83412、O0000、Oabcd

5.2.2 Configure FBs-30GM's operating parameters

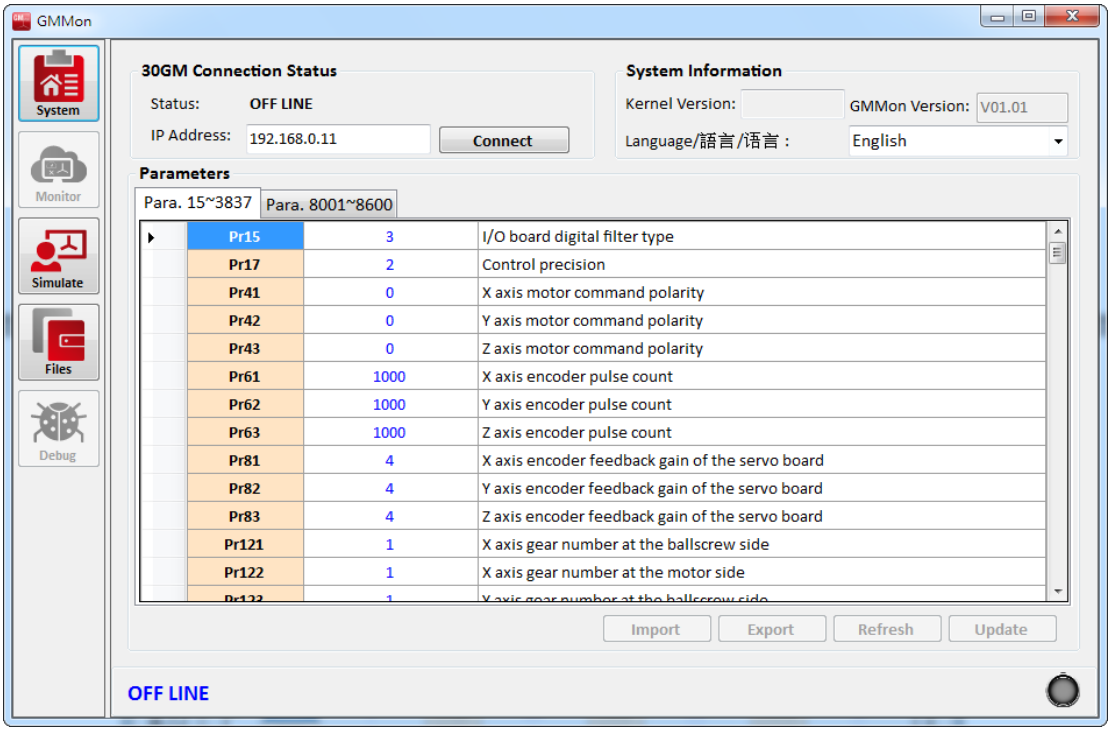


Figure 33: Use GMMon to set up operating parameters

Switch GMMon to System function page. Adjust parameters in the table to fulfill user's requirements.

Users can depend on their requirements to adjust the parameters. About parameter definitions and usage please see Appendix II.

⚠ **Limitations of FBs PLC**

Since FBs-30GM needs to use RS485 (port 2) of FBs PLC as a communication port, any other PLC's communication module or application need to use RS485 (port 2) or it will be impossible to use.

⚠ **When using FBs-30GM, FBs PLC specific registers (D3401 ~ D3467) and relays (M1400 ~ M1499) will be occupied for control purposes, users should avoid using this block registers and relays for other purposes, in order to avoid unexpected results.**

5.2.3 Use the JOG mode to test and adjust machine

Before using PLC to control FBs-30GM's JOG mode, you must first complete the connection between FBs PLC and FBs-30GM. FBs-30GM can execute Jog mode according to the following settings.

1. Go to http://www.fatek.com/ to download FBs-30GM PROGRAM BLOCK which establishes the communication with FBs-30GM (FATEK - Support - Software Download). Before using FBs-30GM PROGRAM BLOCK please update your PLC's OS to version V4.72.
2. Open FBs-30GM PROGRAM BLOCK and then continue to edit PLC's ladder
3. Set FBs-30GM to Jog mode (mode selection please refer to Table 10).

Table 10: Mode selection description

D3426	Description
0	Default value, same as Auto mode
2	Auto mode
4	JOG mode
6	MPG mode
7	HOME mode

4. The axes move by triggering the corresponding special relays (M1403 ~ M1408).

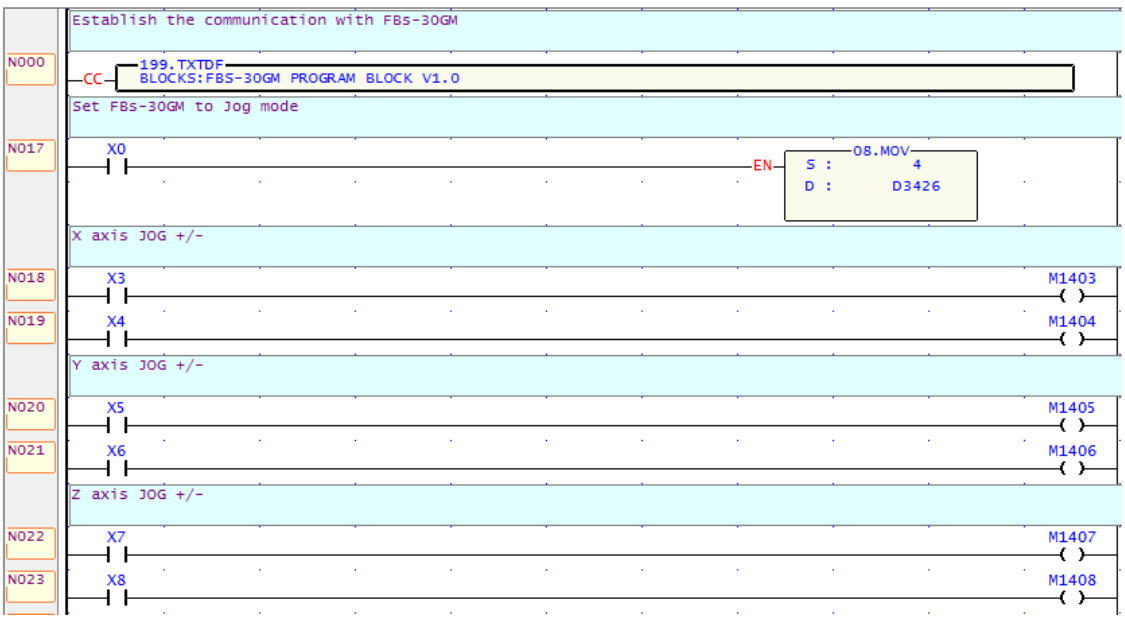


Figure 34: Example of JOG mode ladder diagram

About JOG mode please refer to section 0.

5.2.4 Procedure to execute a motion program

Before using 30GM to execute a motion program, you must first complete the connection between FBs PLC and FBs-30GM. FBs-30GM can run a motion program in Auto mode according to the following settings.

1. Go to http://www.fatek.com/ to download FBs-30GM PROGRAM BLOCK which establishes the communication with FBs-30GM (FATEK - Support - Software Download). Before using FBs-30GM PROGRAM BLOCK please update your PLC's OS to version V4.72.
2. Open FBs-30GM PROGRAM BLOCK and then continue to edit PLC's ladder
3. Set FBs-30GM to Auto mode (mode selection please refer to Table 10).
4. Specify the motion program number (D3431).
5. Set M1400 to start the program specified by D3431. If the value of D3431 is changed when the program is running, the changed setting of specified program would become effective at next start.
6. Motion program can be paused by setting M1401.
7. Set M1402 to stop and reset the motion program and FBs-30GM into standby state.

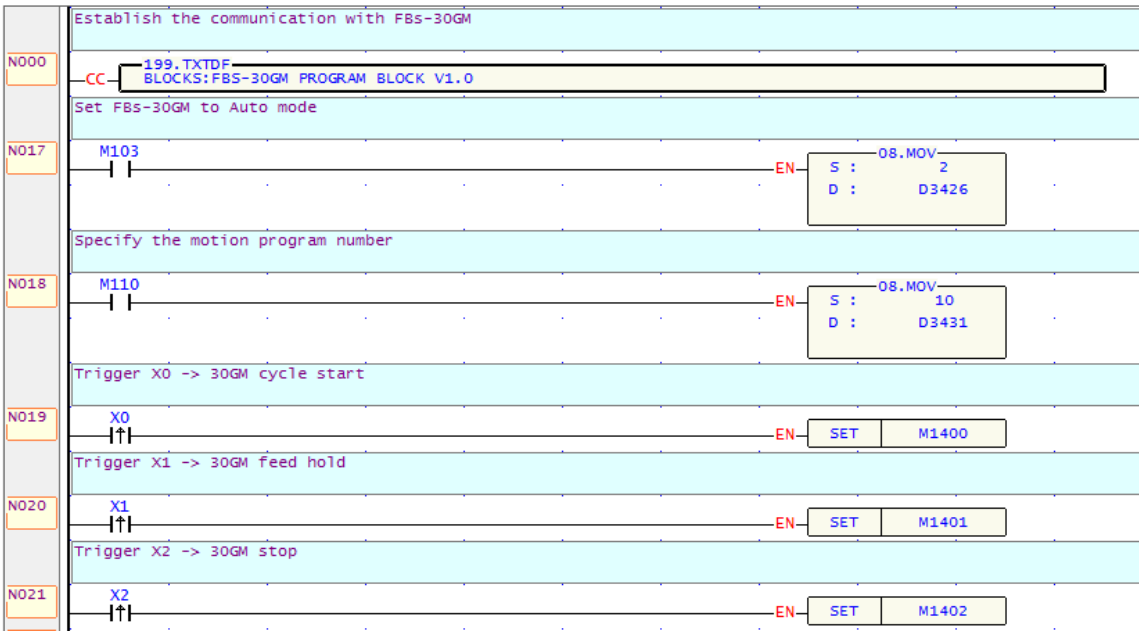


Figure 35: Example of Auto mode ladder diagram

About Auto mode please refer to section 6.1.

5.2.5 Example of FBs PLC ladder diagram

N000: Establishes the communication with FBs-30GM
N017: Set FBs-30GM to JOG mode
N018: Under JOG mode, the X axis moves in the positive direction
N019: Under JOG mode, the X axis moves in the negative direction
N020: Under JOG mode, the Y axis moves in the positive direction
N021: Under JOG mode, the Y axis moves in the negative direction
N022: Under JOG mode, the Z axis moves in the positive direction
N023: Under JOG mode, the Z axis moves in the negative direction
N024: Reset X axis machine position (set current position as the origin of X axis)
N025: Reset Y axis machine position (set current position as the origin of Y axis)
N026: Reset Z axis machine position (set current position as the origin of Z axis)
N027: Set FBs-30GM to Auto mode and specify the motion program No. 10 which is going to be execute
N028: Set M1400 to start the program
N029: Set M1401 to pause the program
N030: Set M1402 to stop the program

FBs-30GM PROGRAMBLK can be downloaded from http://www.fatek.com/ .
(FATEK - Support - Software Download)
Before using FBs-30GM PROGRAM BLOCK please update your PLC's OS to version V4.72.

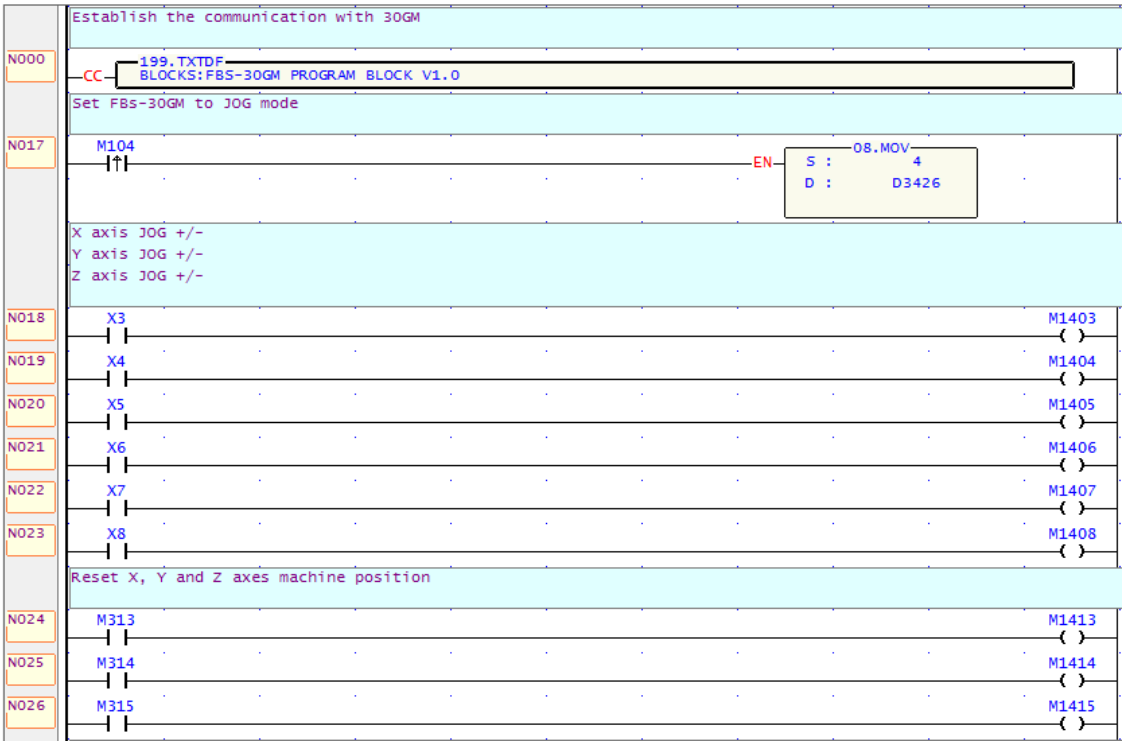


Figure 36: Example of FBs PLC ladder diagram

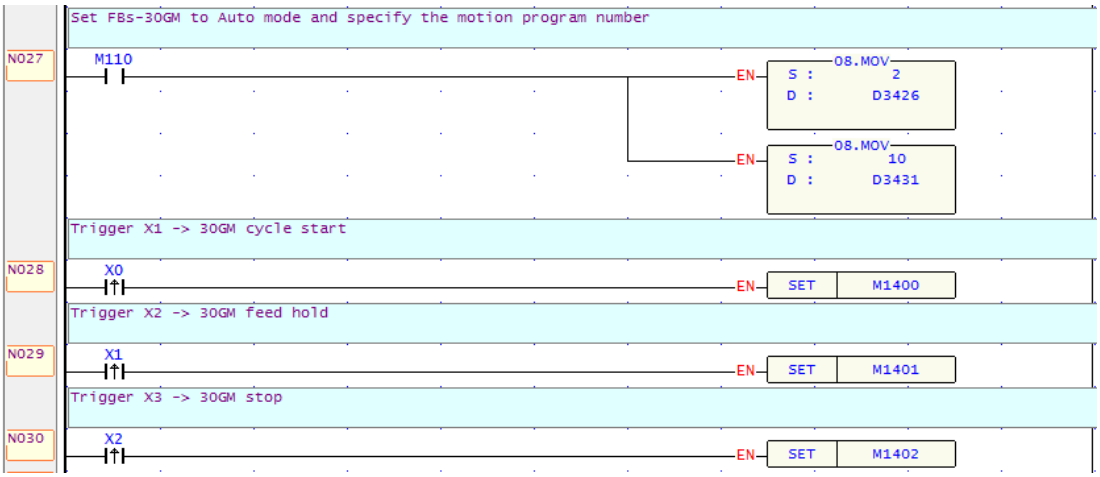


Figure 37: Example of FBs PLC ladder diagram (cont.)

5.3 Control and supervise the operating status

1. In addition to performing motion program, FBs-30GM's has a variety of functions by connecting to FBs-PLC to arrange FBs PLC's special relays (M1400 ~ M1430), special registers (D3426 ~ D3435) or use GMMon to modify the parameters.
2. In the process of motion program. Users can check the special relays (M1464 ~ M1474 and M1480 ~ M1488) and registers (D3440 ~ D3443) to monitor the operating status of FBs-30GM.
3. D3432 ~ D3434 and D3440 ~ D3443, the special registers of FBs PLC, are used to pass MACRO program's user-defined data in one way direction.
   - FBs PLC uses D3432 ~ D3434 to deliver user-defined data to FBs-30GM.
   - FBs PLC uses D3440 ~ D3443 to receive user-defined data from FBs-30GM.
4. FBs-30GM has an analog output terminal, which can be adjusted by setting D3435 to control its output voltage value. D3435 ranges from 0 to 20000 corresponding to the output voltage -10V ~ +10 V linearly. (D34305 = 0, VO =-10V; D3435 = 20000, VO = +10 V)

The user-defined data in FBs-30GM can be accessed in MACRO programs. Information such as X and Y axis coordinates can be delivered with the user-defined data.

About MACRO structure motion language please refer to section 8.

5.4 Troubleshooting

Whenever the system or the program stops due to an alarm, the alarm can be found by the two ways below.

1. Special relay M1474 of FBs PLC is ON.
2. The monitor screen of GMMon displays the alarm code.

⚠ **General alarms can be cleared by triggering STOP after solving the causes of the alarms. Some alarms have to be cleared by shutting down and then restarting FBs-30GM.**

About alarm messages please refer to Appendix III.

5.5 Trigger input terminals to execute motion programs

This function is a special application of FBs-30GM. When FBs-30GM is on standby or during the process of running, FBs-30GM can be assigned to a motion program directly and execute the program immediately by triggering one of the input terminals (X0 ~ X8) without the need to using FBs PLC to set STOP, START or change specified program.

How to use this function:

1. Set FBs PLC's M1424 ON.
2. Set FBs-30GM to Auto mode (mode selection please refer to Table 10).
3. Configure the parameters of FBs-30GM according to your requirement.
4. Trigger one of the input terminals (X0 ~ X8) of FBs-30GM.

After one of the input terminals (X0 ~ X8) of FBs-30GM is triggered, FBs-30GM will do the following actions in sequence.

A. Stop executing program. (No action is taken if FBS-30GM is already on standby).
B. Switch motion program to O1001 ~ O1009 corresponding to X0 ~ X8.
C. Execute once the motion program O1001 ~ O1009.
D. Switch to the previous motion program and return to standby state after the triggered program is finished.

⚠ **Note: Use this method to execute motion program, program name must be named as O1001 ~ O1009. Therefore, pay attention to having the corresponding motion programs in FBs-30GM, otherwise the alarm message will occur.**

6. Operation mode of FBs-30GM

The operation mode of FBs-30GM can be categorized into Auto, JOG, MPG and HOME mode. About instructions of each mode please see the following sections.

6.1 Auto mode

This mode is generally used when executing motion programs. When you want to perform exercise program, you must set the operation mode to" Auto".

In this mode, commands such as start, pause or stop motion programs can be issued by setting special relays. In addition, the applications and operations described in this manual are all based on Auto mode, unless otherwise specified mode.

Operation:

1. Set FBs-30GM to Auto mode (mode selection please refer to Table 10).
2. Specify the motion program number (D3431).
3. Set M1400 to start the program specified by D3431. If the value of D3431 was changed when the program is running, the changed setting of specified program would become effective at next start.
4. Motion program can be paused by setting M1401.
5. Set M1402 to stop and reset the motion program and FBs-30GM into standby state.

6.2 JOG mode

JOG function is suitable for user to test and adjust machine.
In JOG mode user can move the machine toward different directions by triggering the special relays (M1403 ~ M1408) accordingly.

Operation:

1. Set FBs-30GM to JOG mode (set D3426 to 4, mode selection please refer to Table 10).
2. Set FBs-30GM JOG speed percentage (D3429) and JOG feedrate (Pr521 ~ Pr523).
3. Trigger the special relays (M1403 ~ M1408) according to the direction you want the machine to travel toward.

Table 11: Axis JOG feedrate

FBs-30GM motion parameter	Descriptions
Pr521	X-axis JOG feedrate
Pr522	Y-axis JOG feedrate
Pr523	Z-axis JOG feedrate

Table 12: Special relays for JOG

Special relays for JOG	Axis and direction
M1403	X axis＋
M1404	X axis－
M1405	Y axis＋
M1406	Y axis－
M1407	Z axis＋
M1408	Z axis－

6.3 MPG mode

Manual Pulse Generator (MPG) mode is for the purpose of manual or semi-automatic machine control with an external electric hand wheel. Generally MPG mode can adjust machine or vary the execution speed of motion program. FBs-30GM can be used in two ways with electric hand wheel depending on user requirement.

MPG JOG

Description:

User can use MPG (Manual Pulse Generator) mode to move the machine

Operation:

1. Select MPG mode (set D3426 to 6)
2. Select corresponding axis X, Y, Z (set M1409 ~ M1411)
3. Select incremental rate (set D3427)
4. Rotate MPG, machine will move with velocity according to rotation speed of MPG device.

MPG simulation

Description:

Users can use this function to check the speed of motion program file. This function will use the rotation speed of hand wheel to decide the feedrate of G00, G01, G02 and G03. If the hand wheel speeds up, the program moves fast. If the hand wheel stops, then the program also stops. If the hand wheel moves reversely, the program moves reversely too.

Operation:

1. Select AUTO mode (set D3426 to 0 or 2)
2. Set M1412 to on.
3. Set M1400 to start running motion program file.
4. Operator can rotate MPG to run motion program file

The faster MPG rotates, the faster machining speed is. If MPG stops, machine stops too. This function can be “Enable” or “Disable” immediately.
P.S. This function is easy to use for testing machine.

Motion parameter Pr661 ~ 663: axis MPG feedrate upper bound.

6.4 HOME mode

Because of the tool setting, motion program coordinate is based on Machine zero point. So it is necessary to make sure where Machine zero point (HOME) is. When FBs-30GM boots up, the execution of reference searching (home search) is important. User should complete home return before starting AUTO motion program files.

The following describes three approaches of home return for users to select according to their machines. If users do not know which approach to choose or machines lack HOME DOG / motor index signals, users can adopt the instructions of “using absolute encoder” to do Home mode.

Using motor feedback

Step 1: Switch FBs-30GM to HOME mode (set D3426 to 7)
Step 2: Press JOG＋/－ of desired home return axis
Step 3: Motor moves to HOME DOG according to homing direction (Pr861 ~ 863), and 1st homing speed (Pr821 ~ 823)
Step 4: When FBs-30GM receives home DOG signal, it begins to stop
Step 5: After the motor stops at point A, it will move backwards with axis homing 2nd part speed (Pr841 ~ 843)

Step 6: When the machine leaves home DOG, FBs-30GM will search the nearest motor index signal

Step 7: After FBs-30GM receives the motor index signal, FBs-30GM will plan the stop action according to the home search method (Pr961 ~ 963) and homing offset (Pr881 ~ 883), and finally the motor will stop at point B

Step 8: After completing the 1st time HOME return, FBs-30GM will initialize the system data below according to home search method (Pr961 ~ 963) and home offset (Pr881 ~ 883).

Table 13: Parameters of home search method and axis home offset

	No961=0 No881=0	**No961=0/1 No881=L**	**No961=2 No881=L**
The absolute position command	0	0	-L
The absolute position feedback	0	0	-L
Machine coordinate	0	0	-L

P.S.

After the 2nd time HOME return, FBs-30GM will only execute step 8.

V-X diagram (speed vs position) for each type of HOME return is shown as below:

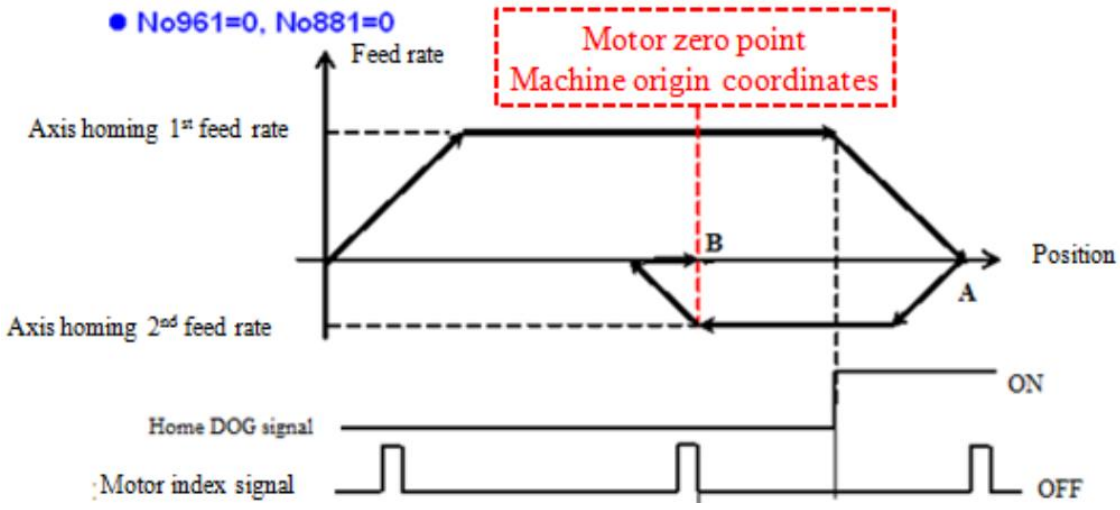


Figure 38: V-X diagram of using motor feedback, Pr961=0 and Pr881=0

● No961=0 or 1, No881=L

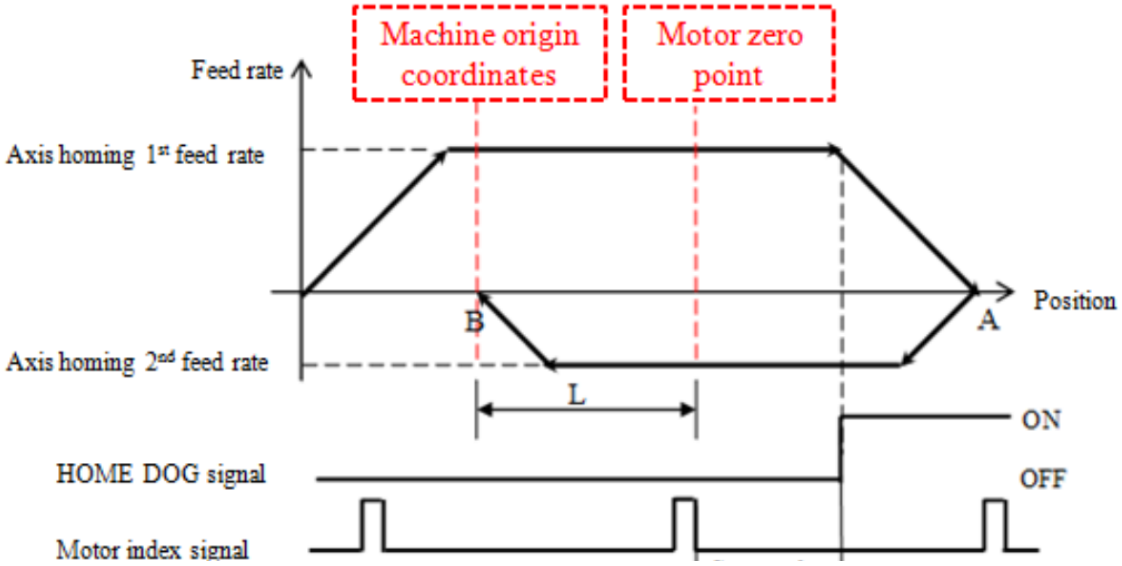


Figure 39: V-X diagram of using motor feedback, Pr961=0 or 1 and Pr881=L

● No961=2, No881=L

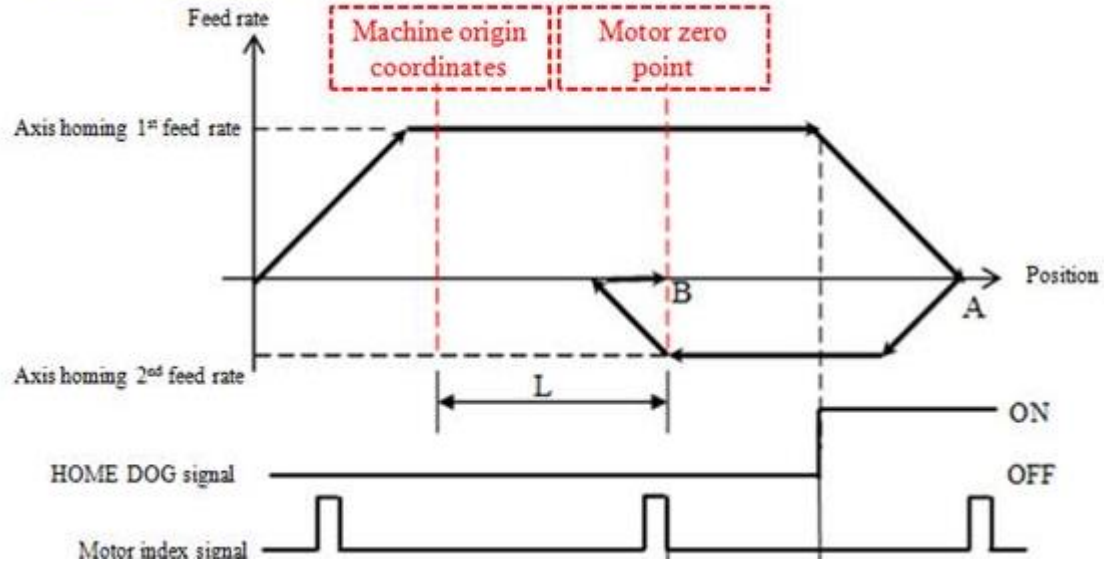


Figure 40: V-X diagram of using motor feedback, Pr961=2 and Pr881=L

Using linear encoder – dual feedback

Step 1: Switch FBs-30GM to home mode (set D3426 = 7)

Step 2: Press JOG＋/－ of desired home search axis

Step 3: Motor moves to HOME DOG according to homing direction (Pr861 ~ 863), and 1st homing speed (Pr821 ~ 823)

Step 4: When FBs-30GM receives the home DOG signal, it will plan the stop action

Step 5: After the motor stops at point A, it will move backwards with axis homing 2nd part speed (Pr841 ~ 843)

Step 6: When the machine leaves the home DOG, FBs-30GM waits for the nearest zero point on linear encoder

Step 7: After FBs-30GM receives the zero point on linear encoder, FBs-30GM will plan the stop action according to the home search method (Pr961 ~ 963) and homing offset (Pr881 ~ 883), and finally the motor will stop at point B

Step 8: At the 1st HOME return, linear encoder – dual feedback does not work, and due to the effect of mechanical error, machine cannot stop exactly on desired position (zero point of linear encoder or HOME offset), so after motor really stops on B point, FBs-30GM will instantly calculate this error Δ

Step 9: FBs-30GM will initialize the system data below according to home search method (Pr961 ~ 963) and home offset (Pr881 ~ 883).

P.S.

- After booting, linear encoder – dual feedback is always enabled when the 1st time HOME return is finish.
- After booting, from the 2nd time return HOME, FBs-30GM will only execute step 9.
- After executing the 1st HOME return successfully, the error Δ between real machine position and target position will be compensated in the next interpolation.

V-X diagram (speed vs position) for each type of HOME return is shown as below:

Table 14: Home mode and home offset settings

	No961=0 No881=0	No961=0/1 No881=L	No961=2 No881=L
The absolute position command	0	0	-L
The absolute position feedback	0	0	-L
The dual feedback position	Δ	Δ	Δ
Mechanical coordinate	0	0	-L

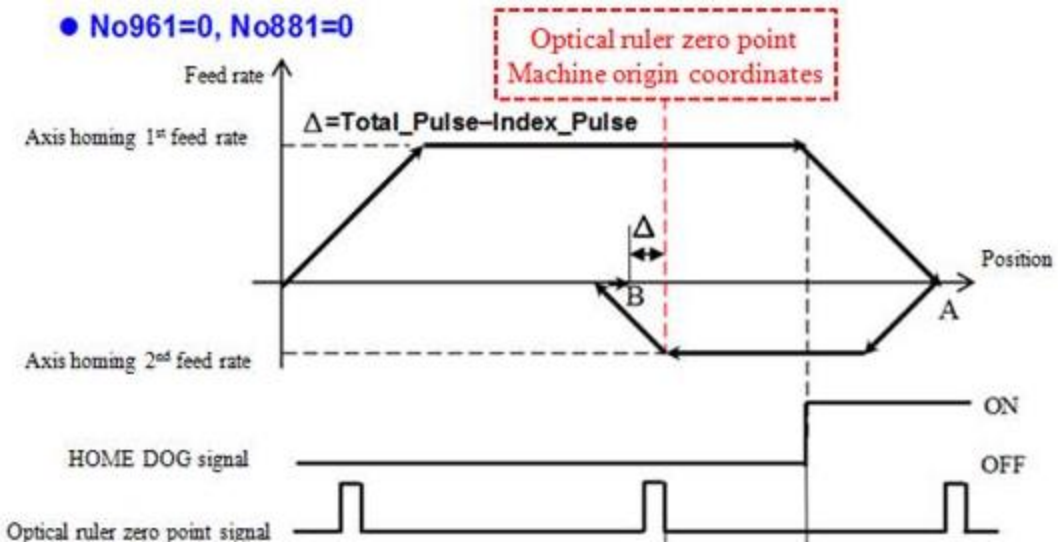


Figure 41: V-X diagram of dual feedback, Pr961=0 and Pr881=0

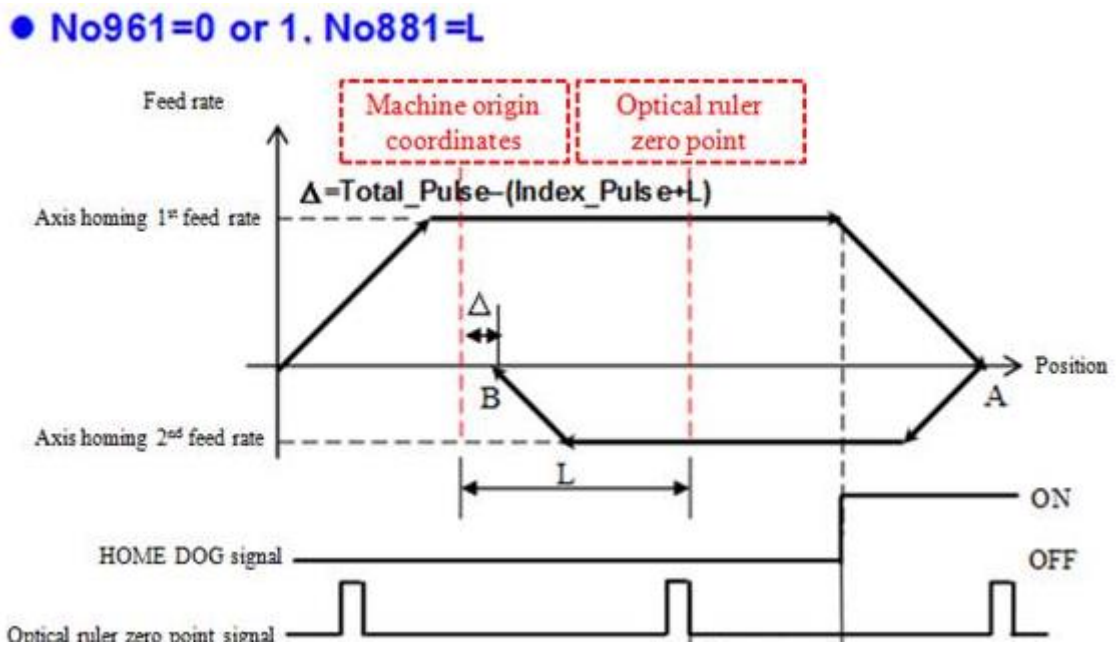


Figure 42: V-X diagram of dual feedback, Pr961=0 or 1 and Pr881=L

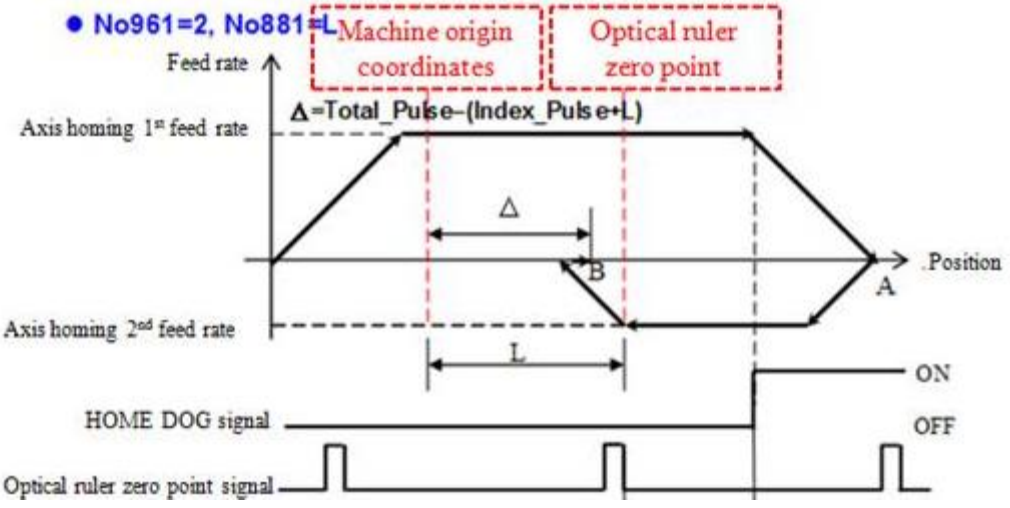


Figure 43: V-X diagram of dual feedback, Pr961=2 and Pr881=L

Using absolute encoder

Step 1: Move axis to the appointed point for machine origin during tuning process of servo driver

Step 2: After triggering M1413 ~ M1415, FBs-30GM automatically records the initial value A from encoder

Step 3: Next time when FBs-30GM is rebooted and communicates successfully with driver, regardless of positions of axis, FBs-30GM will compare present motor encoder position with value A to calculate the correct motor position

Step 4: Updating machine coordinate, servo command and motor feedback. (If dual feedback control is used, linear encoder feedback will be updated at the same time).

P.S.

This is the easiest approach of reference searching, as long as you trigger M1413 ~ M1415 to complete the steps and take current location as the origin of coordinates.

Home return disorders diagnostic steps

1. Axis moves in the opposite direction and stops until it meets hardware stroke limit when executing HOME return.

   Possible reasons:

   a. HOME DOG signal is always ON.

   Diagnostic method:

   Check if input HOME DOG signal of FBs-30GM is always ON.

   b. Servo motor index signal does not enter FBs-30GM.

   Diagnostic method:

   Move the axis manually, check whether the value of system debug variables 48 (X-axis), 49 (Y-axis) and 50 (Z-axis) change once or not when the motor turns one revolution, and the difference must equal to encoder resolution (parameters Pr61 ~ 63 and Pr81 ~ 83).

   c. FBs-30GM parameters are wrong

   Checking following parameters:

   - ✓ Pr201 ~ 203(encoder type) are set 0 or 1
   - ✓ Pr41 ~ 43(axis motor polarity) are the same as default setting of manufacturer
   - ✓ Pr861 ~ 863(axis homing direction) are the same as default setting of manufacturer

2. Related system alarms below, for detailed descriptions please refer to Appendix III.

   MOT-021: Must re-homing
   MOT-022: Home position inaccurate
   MOT-029: Miss index in homing
   MOT-030: Zero speed timeout in homing
   MOT-036: Can't leave home dog

7. G-code and M-code of motion program

7.1 G-code instructions

Table 15: G-code instructions listing

G-Code	Description	G-Code	Description
G00	Positioning	G66	Marco call
G01	Linear interpolation	G67	Marco call cancel
G02	Circular interpolation / Helical interpolation (CW)	G70	Unit setting of inch system
G03	Circular interpolation / Helical interpolation (CCW)	G71	Unit setting of metric system
G04	Dwell	G90	Absolute command
G09	Exact stop	G91	Incremental command
G17	X-Y plane selection	G92	Program coordinate system setting
G18	Z-X plane selection	G92.1	Rotating program coordinate system setting
G19	Y-Z plane selection	G161	Compensation setting of linear interpolation
G28	Return to reference position	G162	Vector compensation setting of circular interpolation
G28.1	Incremental distance triggered by sensor	G163	Radius compensation setting of circular interpolation
G30	2nd, 3rd and 4th reference position return	G164	Interpolation compensation cancellation
G53	Machine coordinate system setting	G165	Electrical zero point setting
G65	Simple calling	G166	Return to electrical zero point

G00	**POSITIONING**	G00

Command form:

G00 X__Y__Z__;

X、Y、Z: Specified point

Description:

Each axles move to appointed point in no interpolation status, X、Y、Z is the final position, use G90/G91 to design absolute or increment value.

<Notice>:

The movement mode can decide by motion parameter Pr411

(0: linear, 1: each axle move in max speed independently)

Example

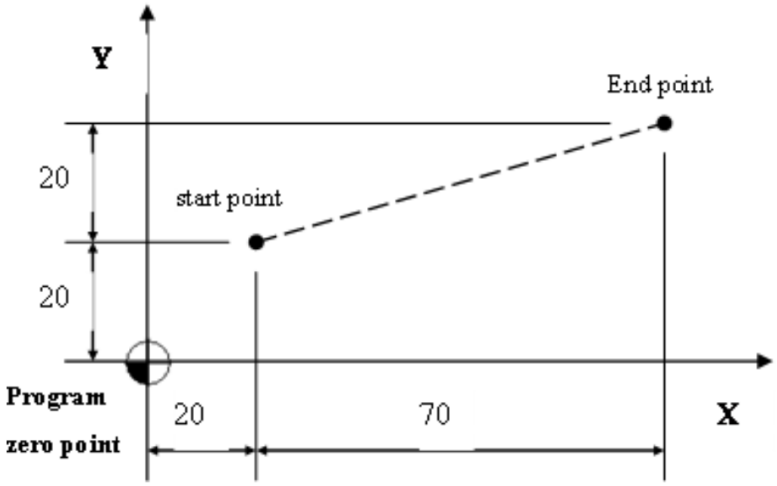


Figure 44: G00 positioning example

Program description:

1. First way (absolute): G90 G00 X90.0 Y40.0;
   //use difference value between appointed point and zero point to do straight interpolation to appointed point

2. Second way (increment): G91 G00 X70.0 Y20.0;
   //use difference value between appointed point and initial point to do straight interpolation to appointed point

G01	LINEAR INTERPOLATION	G01

Command form:

G01 X__ Y__ Z__ F__;

X、Y、Z: Specified point

F: Feed rate (mm/min)

Description:

G01 executes linear interpolation, it can be used with G90/G91 to decide absolute or increment mode, use feed rate provided by **F** to go to the specified position.

Example1:

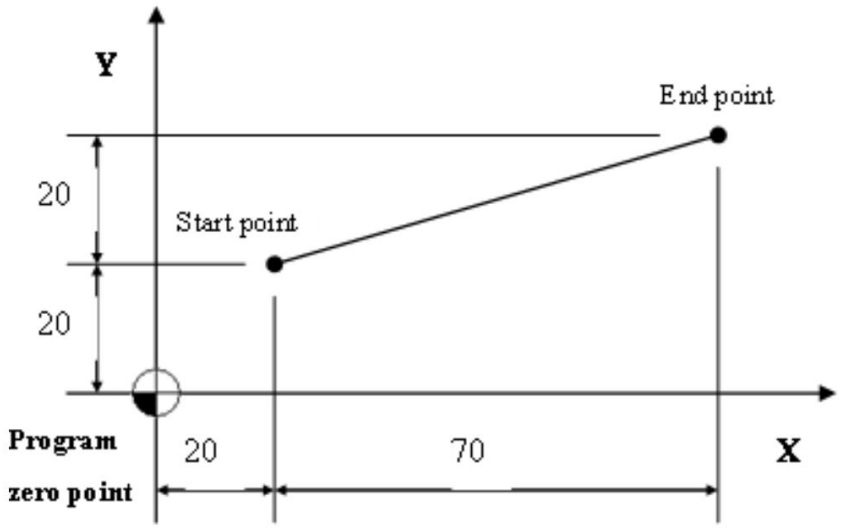


Figure 45: G01 linear interpolation example 1

1. Absolute command: G90 G01 X90.0 Y40.0;
   //do linear interpolation from zero point to the specified point(90,40)

2. Increment command: G91 G01 X70.0 Y20.0;
   //the tool does linear interpolation X + 70 and Y + 20 to the specified point

Example 2: processing example

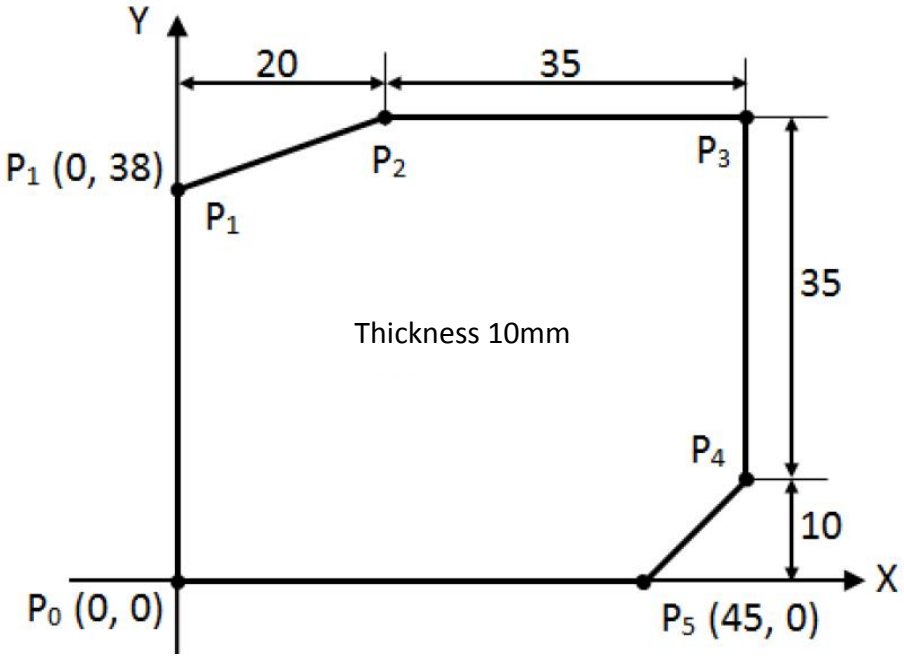


Figure 46: G01 linear interpolation example 2

Program description:

1. Absolute way:

N001 G00 X0.0 Y0.0 Z10.0; //positioning to above of P_0

N002 G90 G01 Z-10.0 F1000 ; //straight interpolation to bottom of workpiece, speed 1000mm/min

N003 Y38.0; //$P_0 \rightarrow P_1$

N004 X20.0 Y45.0; //$P_1 \rightarrow P_2$

N005 X55.0; //$P_2 \rightarrow P_3$

N006 Y10.0; //$P_3 \rightarrow P_4$

N007 X45.0 Y0.0; //$P_4 \rightarrow P_5$

N008 X0.0; //$P_5 \rightarrow P_0$

N009 G00 Z10.0; //positioning back to above of P_0

N010 M30; //program end

2. Increment way

N001 G00 X0.0 Y0.0 Z10.0;//positioning to above of P_0

N002 G91 G01 Z-20.0 F1000;//straight interpolation to bottom of workpiece, speed 1000mm/min

N003 Y38.0; //$P_0 \rightarrow P_1$

```
N004  X20.0 Y7.0;          //P1→P2
N005  X35.0;               //P2→P3
N006  Y-35.0;              //P3→P4
N007  X-10.0 Y-10.0;       //P4→P5
N008  X-45.0;              //P5→P0
N009  G00 Z20.0;           //positioning back to above of P0
N010  M30;                 //program end
```

G02 G03	**CIRCULAR INTERPOLATION**	G02 G03

Command form:

1. X-Y plane circular interpolation:

$$\text{G17}\ \begin{Bmatrix} G02 \\ G03 \end{Bmatrix}\ X\_\ Y\_\ \begin{Bmatrix} R\_ \\ I\_\ J\_ \end{Bmatrix}\ F\_$$

2. Z-X plane circular interpolation:

$$\text{G18}\ \begin{Bmatrix} G02 \\ G03 \end{Bmatrix}\ X\_\ Z\_\ \begin{Bmatrix} R\_ \\ I\_\ J\_ \end{Bmatrix}\ F\_$$

3. Z-X plane circular interpolation:

$$\text{G19}\ \begin{Bmatrix} G02 \\ G03 \end{Bmatrix}\ Y\_\ Z\_\ \begin{Bmatrix} R\_ \\ I\_\ J\_ \end{Bmatrix}\ F\_$$

X, Y, Z: Specified point

I, J, K: the vector value that starting point of arc to the center of a circle (center of a circle－starting point)

R: Radius of arc

F: Feed rate

G90/G91 decide absolute or increment

Description:

G02, G03 do circular interpolation according to appointed plane, coordinate system, size of arc and speed of interpolation, and the rotate direction decide by G02 (CW), G03 (CCW). Description of the command format as below:

Table 16: G02/G03 circular interpolation

Setting Data			Command	Definition
1	Plane selection		G17	X-Y plane setting
			G18	X-Z plane setting
			G19	Y-Z plane setting
2	Direction		G02	Clockwise direction (CW)
			G03	Counterclockwise direction (CCW)
3	End position	G90	Two axes of X, Y, Z	End coordinate of arc
		G91	Two axes of X, Y, Z	Vector value from start point to end point
4	Distance from start point to center of circle		Two axes of I, J, K	Vector value from start of arc to center of circle
	Radius of arc		R	Radius of arc
5	Speed of feed (feedrate)		F	Feedrate along the arc

Example:

1. G02, G03direction:

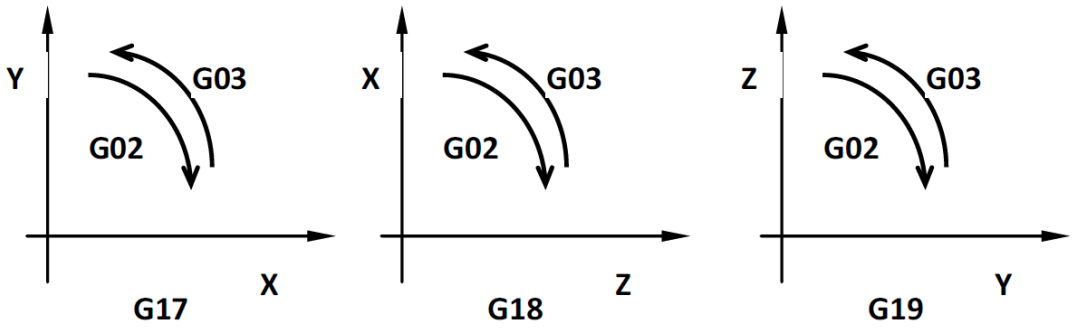


Figure 47: G02, G03 direction

2. I, J, K definition:

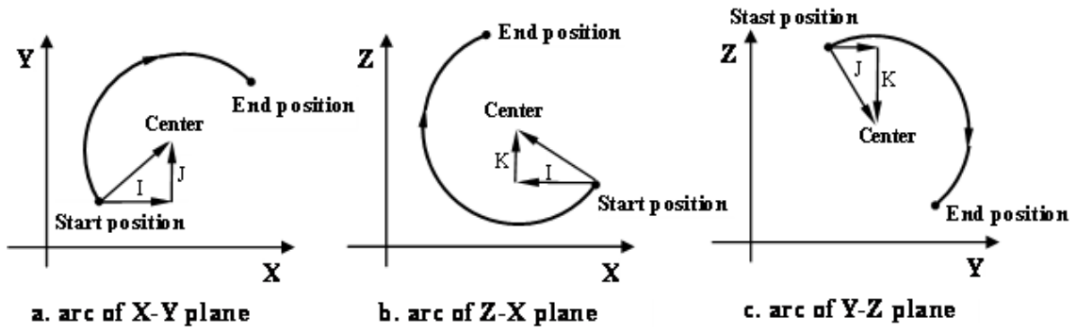


Figure 48: G02, G03 vector of I,J and K

3. How to use R

- When θ≦180 degree, R is positive.

$$\begin{Bmatrix} G02 \\ G03 \end{Bmatrix} \; X\_\_ \; Y\_\_ \quad R25.0$$

- When 180 degree＜θ＜360 degree, R is negative.

$$\begin{Bmatrix} G02 \\ G03 \end{Bmatrix} \; X\_\_ \; Y\_\_ \quad R-25.0$$

- When θ=360 degree, use **I, J and K**.

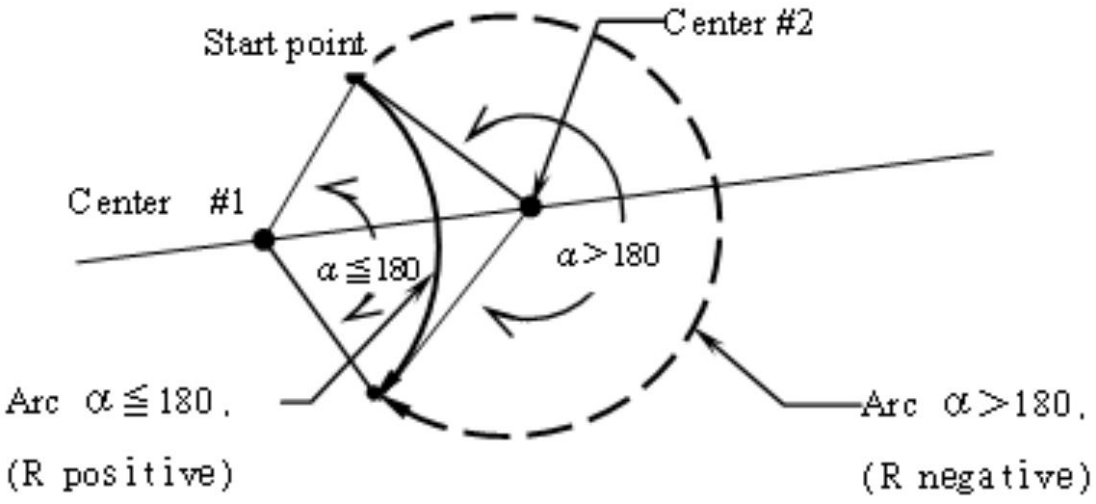


Figure 49: Circular interpolation of different θ

Example 1:

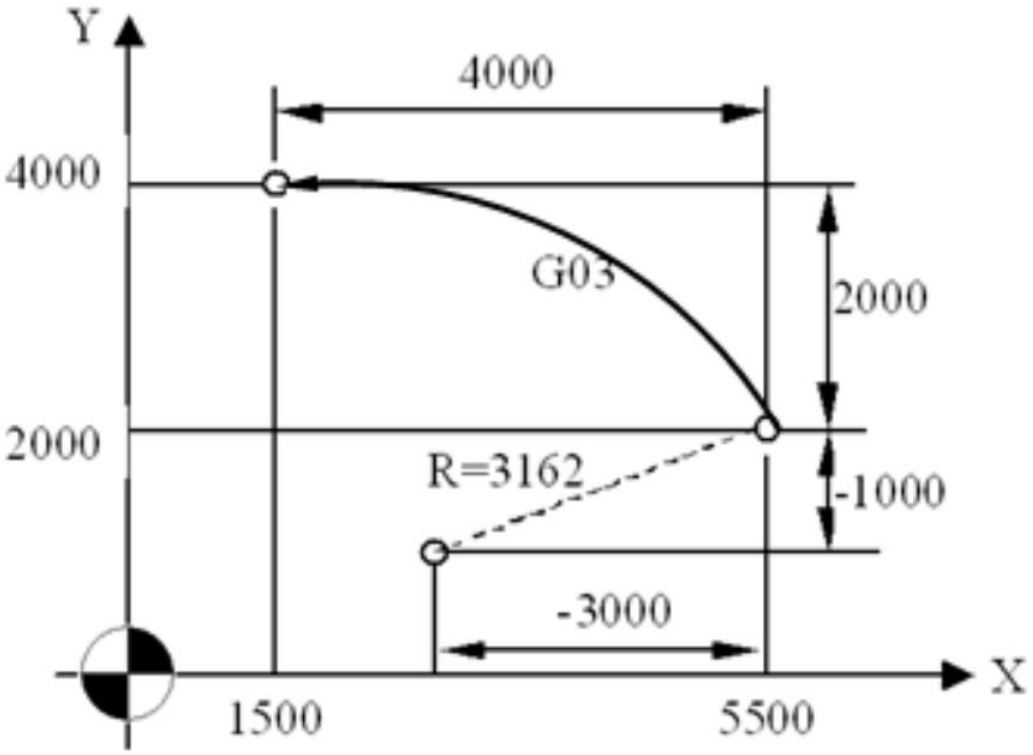


Figure 50: Circular interpolation example 1

G90 G00 X5500 Y4000; //positioning to start point of arc
G17 G90 G03 X1500 Y4000 I-3000 J-1000 F200; //absolute command
(G17 G91 G03 X-4000 Y2000 I-3000 J-1000 F200; //increment command)

Example 2: (interpolate a full circle)

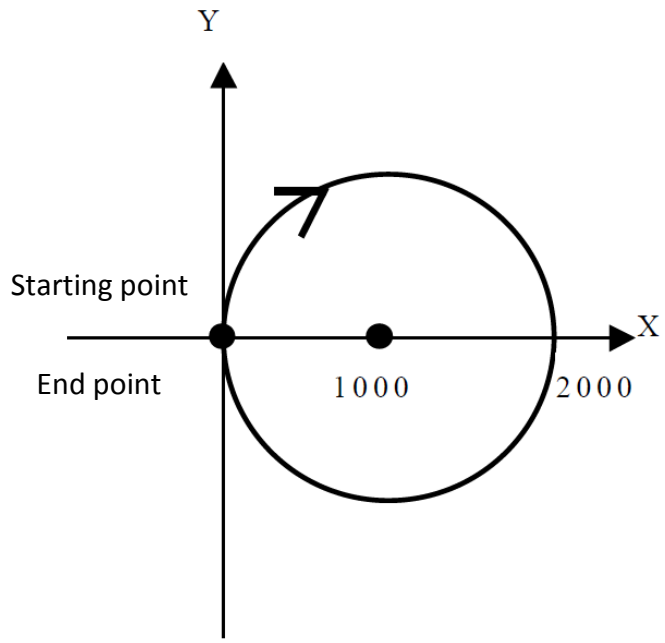


Figure 51: Circular interpolation example 2

G90 G00 X0 Y0;
G02 I1000 F100; //interpolate a full circle

G02 G03	**HELICAL INTERPOLATION**	G02 G03

Command form:

1.

$$\text{G17}\ \begin{Bmatrix} G02 \\ G03 \end{Bmatrix}\ X\_\ Y\_\ \begin{Bmatrix} R\_ \\ I\_\ J\_ \end{Bmatrix}\ Z\_\ F\_$$

X, Y: end position of arc;

Z: end position of straight line;

R: radius of arc;

I, J: center position of arc;

F: speed of tool feed(feed rate);

2.

$$\text{G18}\ \begin{Bmatrix} G02 \\ G03 \end{Bmatrix}\ X\_\ Z\_\ \begin{Bmatrix} R\_ \\ I\_\ J\_ \end{Bmatrix}\ Y\_\ F\_$$

X, Z: end position of arc;

Y: end position of straight line;

R: radius of arc;

I, K: center position of arc;

F: speed of tool feed(feed rate);

3.

$$\text{G19}\ \begin{Bmatrix} G02 \\ G03 \end{Bmatrix}\ Y\_\ Z\_\ \begin{Bmatrix} R\_ \\ I\_\ J\_ \end{Bmatrix}\ X\_\ F\_$$

Y, Z: end position of arc;

X: end position of straight line;

R: radius of arc;

J, K: center position of arc;

F: speed of tool feed(feed rate);

Description:

When the 3rd axis which is vertical to arc plane moves, G02/G03 is to be helical interpolation. The choice of helical interpolation is the same as circular interpolation. Helical interpolation uses G code (G17/G18/G19) to decide which plane to do circular interpolation.

G17 form: synchronously with arc of X-Y plane.
G18 form: synchronously with arc of Z-X plane.
G19 form: synchronously with arc of Y-Z plane

Example:

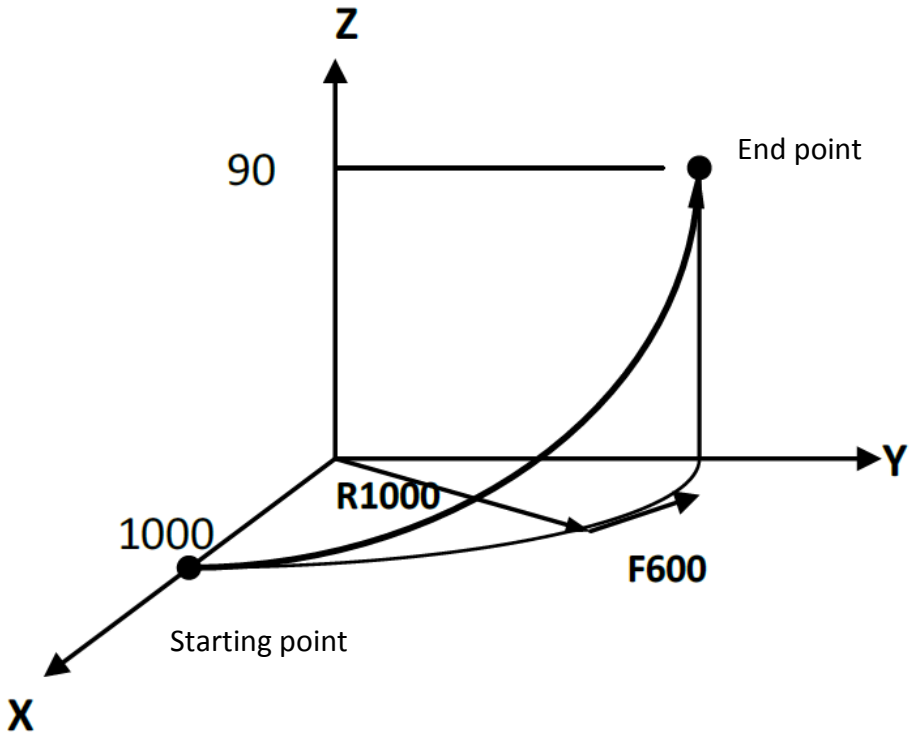


Figure 52: Helical interpolation

Program description:
G17 G03 X0.0 Y1000.0 R1000.0 Z900.0 F600;
// synchronously with arc of X-Y plane (CCW), do helical interpolation with feedrate 600mm/min

G04	**DWELL**	G04

Command form:

$$\mathbf{G04}\ \begin{Bmatrix} \boldsymbol{X}\_ \\ \boldsymbol{P}\_ \end{Bmatrix}$$

X: specific time (decimal point permitted 0.001～9999.999s)

P: specific time (decimal point not permitted)

Description:

By specifying a dwell, the execution of the next block is delayed by the specified time. In addition, a dwell can be specified to make an exact check.

Example:

G04 X2500; //delay 2.5 sec

G04 X2.5; //delay 2.5 sec

G04 P2500; //delay 2.5 sec

G04 P2.5; //delay **2** sec (decimal point not permitted)

G09	**EXACT STOP**	G09

Command form:

$$\mathbf{G09}\ \begin{Bmatrix} \boldsymbol{G00} \\ \boldsymbol{G01} \end{Bmatrix}\ \mathbf{X}\_\ \mathbf{Y}\_\ \boldsymbol{Z}\_..$$

X, Y, Z: position of exact stop

Description:

When pass through the corner, because tool moves too fast or servo system delays, tool cannot cut the exact shape of corner, but when you need to cut high precision rectangular, you can use G09 or G61 to make it, it slow down the tool when approach to corner, when reach to the specified position (in motion parameter range), it will run the next block. G09 exact stop only be effective in one block which has G09.

Notice:

G01 check window: parameter Pr421-423

G00 check window: parameter Pr461-463

Example:

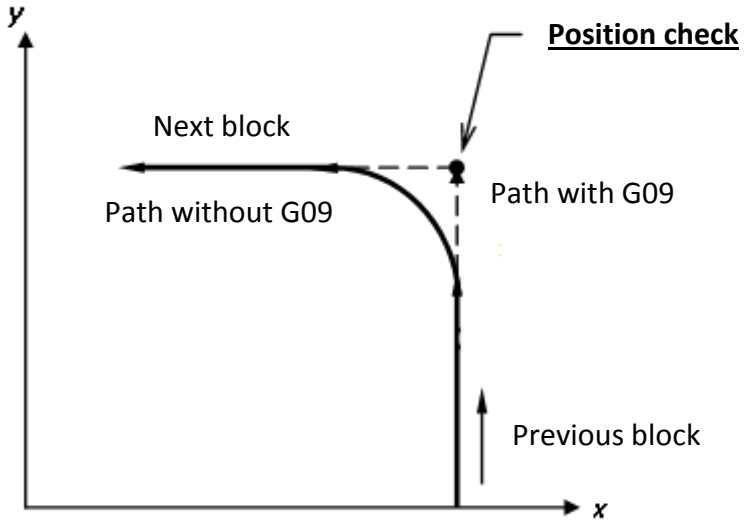


Figure 53: Exact stop example

G17	**X-Y PLANE SELECTION**	G17
G18	**Z-X PLANE SELECTION**	G18
G19	**Y-Z PLANE SELECTION**	G19

Command form:

G17; // X-Y plane selection

G18; // Z-X plane selection

G19; // Y-Z plane selection

Description:

When use circular interpolation, tool radius compensation or polar coordinate command, need to use G17, G18, or G19 to set moving plane and tell FBs-30GM the working plane (default G17).

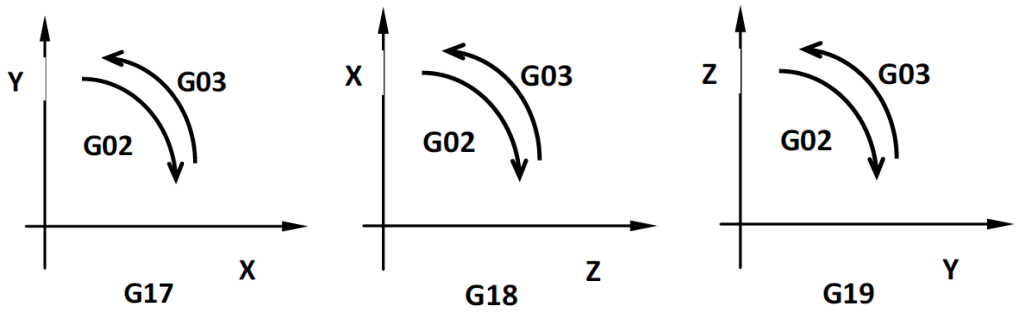


Figure 54: G17, G18, G19 setting interpolation plane

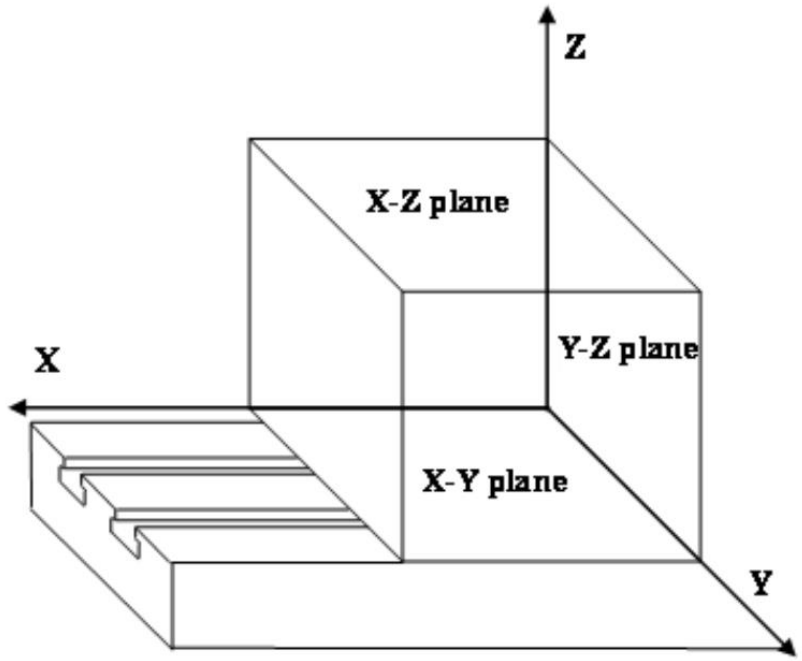


Figure 55: X-Y-Z space

G28	**RETURN TO REFERENCE POSITION**	G28

Command form:

G28 X__Y__Z__;

X, Y, Z: mid-point position (absolute value in G90 mode, increment value in G91 mode)

Description:

It can return to reference position or return to origin point, in order not to let the tool crush, it will use G00 mode to move from present position, it will move to the specified safety mid-point first and then return to origin point or reference point. Only the axes which are given values when using G28 will perform the reference position return.

Example 1:

G90 **G28** X50.0 Y30.0; //A→B→C, mid-point (50,30)

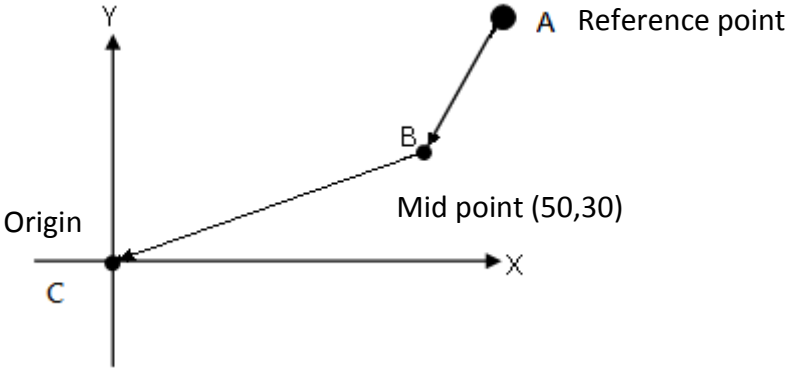


Figure 56: G28 return to reference position example1

Example 2:

G28 X0; //X axis return to zero point, Y axis and Z axis stay the same.

G28 Y0; //Y axis return to zero point, X axis and Z axis stay the same.

G28 Z0; //Z axis return to zero point, X axis and Y axis stay the same.

G28.1	**INCREMENTAL DISTANCE TRIGGERED BY SENSOR**	G28.1

Command form:

G28.1 X_ Q_ R_ F1 = _ F2 = _;

X: Specified point of the first part (X can be replaced with Y or Z).

Q: Second part distance, if there is no this argument, the second part distance will be the same with the first part (incremental distance).

R: The distance to the sensor

F1: The speed of the first part

F2: The speed of the second part

F: If F1 and F2 are not specified, the speed will be the same as the value of F_.

Description:

Move to X with the specified speed F1.

After reaching X, move to Q with the specified speed F2.

If FBs-30GM meets the optical sensor signal during the second part, FBs-30GM will immediately move R away from the sensor. Otherwise after the machine moves to

Q, the execution of the block is completed

Notice:
Please connect the optical sensor to the terminal of index signal.

G30	**2nd, 3rd and 4th REFERENCE POSTION RETURN**	G30

Command form:
G30 Pn X__ Y__ Z__ ;

X、Y、Z: mid-point coordinates; (absolute value under G90, increment value under G91)
Pn: Specified reference point (parameter #2801 ~ #2860)
P1: mechanical origin point;
P2: second reference point;
P_: default is P2;

Description:
For the convenience that change tool and check, we use parameter to set a reference point to suitable position, it can let tool need not return to mechanical zero point, increase efficiency in changing the tool, the usage of this command is the same as G28 only expect returned point. Floating reference position return command, usually use in the position of automatically change the tool differ from the origin point. Movement is G00 mode.

Example:

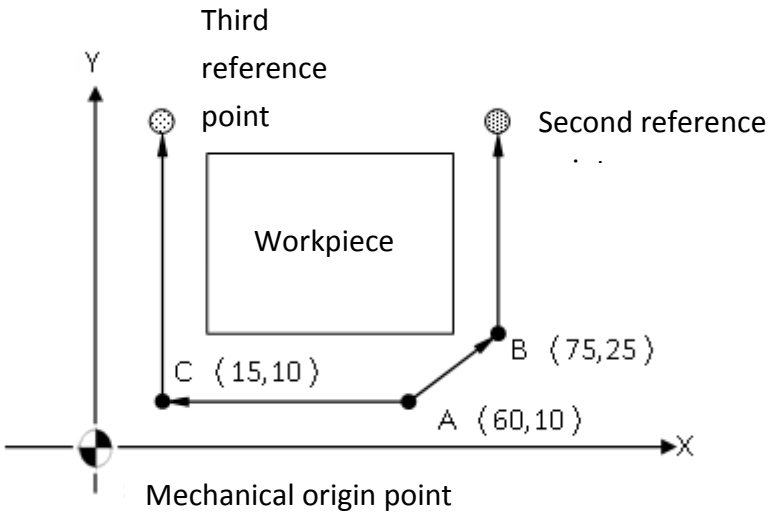


Figure 57: G30 reference position return example

Program description: presume tool is in A (60,10)

1. to second reference point

G30 P2 X75.0 Y25.0;//A→B→2nd reference point

2. to third reference point

G30 P3 X15.0 Y10.0;//A→C→3rd reference point

G53	**Machine coordinate system setting**	G53

Command form:

G53 X___ Y___ Z___;

X: move to specify machine coordinate of X position.

Y: move to specify machine coordinate of Y position.

Z: move to specify machine coordinate of Z position.

Description:

Machine origin point is the fixed origin point when factory build the machine, this coordinate system is fixed; when G53 is specified tool will move to the specified position on machine coordinate, when tool returns to machine zero point (0, 0, 0), this point is the origin point of machine coordinate system.

<Notes>:

1. G53 only effective in specified block;
2. G53 only effective absolute mode(G90), not effective in increment mode(G91);

3. Before use G53 to set coordinate system, must set coordinate system on the basement of reference return position by manual.

G65	**SIMPLE CALL**	G65

Command form:

G65 P__ L__ ;

P: number of the program to call;

L: repetition count;

Description:

After calling MACRO, P__ is called to execute and L__ indicates repeating times. But it is enabled only in the block with G65.

Example:

G65 P10 L20 X10.0 Y10.0

//Call sub-program O0010 continuously 20 times, and set X=10.0 and Y=10.0 into sub-program.

G66	**MACRO CALL**	G66
G67	**MACRO CALL CANCEL**	G67

Command form:

G66 P__ L__ ;macro call

G67 ;macro call cancel

P: number of the program to call;

L: repetition count;

Description:

After G66 is called, P__ is called to execute and L__ indicates repeating times. If there is a moving block, G66 block will be executed again after moving block ends until using G67 to cancel it.

Example:

N001 G91
N002 G66 P10 L2 X10.0 Y10.0
// Repeat twice calling sub-program O0010 and set X=10.0 and Y=10.0 into sub-program.
// Move to position X=20.0. After moving, call G66 P10 L2 X10.0 Y10.0.
N004 Y20.0
// Move to position Y=20.0. After moving, call G66 P10 L2 X10.0 Y10.0.
N005 G67 // Cancel macro call mode.

G70 G71	**UNIT SETTING OF INCH SYSTEM** **UNIT SETTING OF METRIC SYSTEM**	G70 G71

Command form:

G70;

G71;

Description:

G70: inch system

G71: metric system

After change inch/metric system, origin offset value of workpiece coordinate, tool data, system parameter, and reference point, all of that is still correct. System will deal the change of unit automatically. After change inch/metric system, item below will change as follow:

- Coordinate, unit of speed
- Increment JOG unit
- MPG JOG unit

Decimal Point Input

When parameter is inputted by decimal point input, will to be the common measurement unit, mm, inch, sec…etc., if input by whole number, it will to be the Min unit that system default, mm, ms, …etc.

Precision (BLU:)

Set motion parameter Pr17 to Control precision (BLU):

1: 0.001inch / 0.01mm / 0.01deg;

2: 0.0001inch / 0.001mm / 0.001deg;

3: 0.00001inch / 0.0001mm / 0.0001deg.

G90 G91	**ABSOLUTE COMMEND** **INCREMENT COMMEND**	G90 G91

Command form:

G90;

G91;

Description:

G90: absolute command.

G91: incremental command.

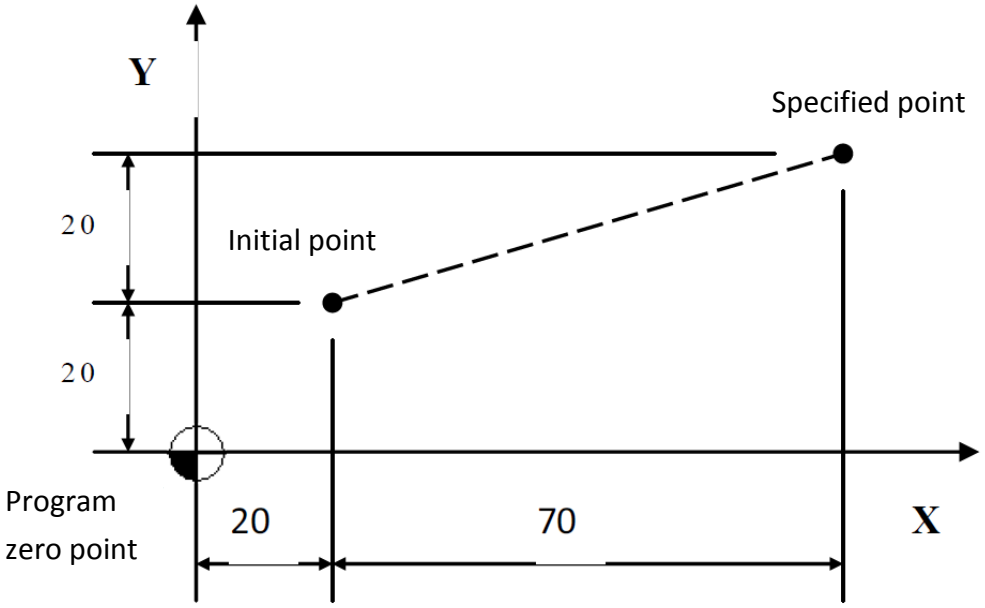


Figure 58: G90/G91 (absolute/increment) commend example

Program description:

1. First way(absolute): G90 G00 X90.0 Y40.0;

//use the different distance from specified point to program zero point, to linear interpolation to specified point

2. Second way(increment): G91 G00 X70.0 Y20.0;

//use the different distance from specified point to starting point, to linear interpolation to specified point

G92	**PROGRAM COORDINATE SYSTEM SETTING**	G92

Command form:

G92 X__ Y__ Z__;

X, Y, Z: set the position that work coordinate system(G92) in programmable coordinate system

Description:

When we design the program, we must set another program coordinate zero point, we can use G92 to set a new coordinate system at this time, this command is set a new zero point of coordinate system when the tool is in any position, after setting tool will start to perform at this point, absolute command is computed by this new coordinate system.

Example:

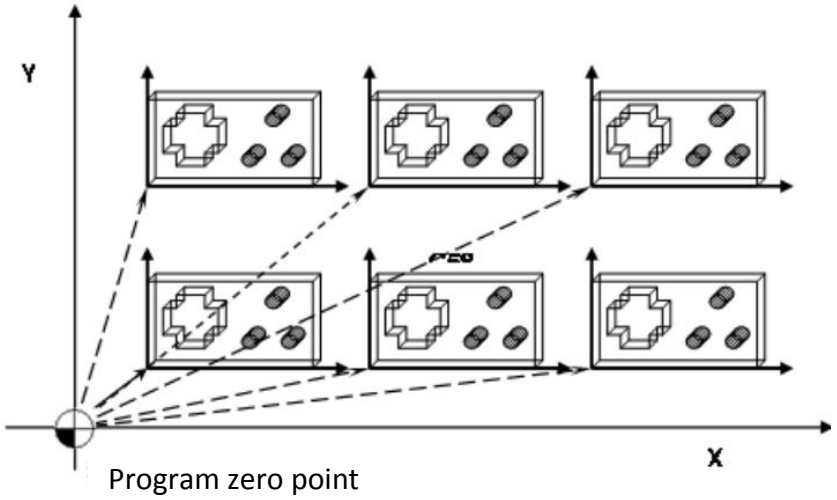


Figure 59: Program coordinate system setting example

Do the specified MACRO program and set program coordinate to zero before execution MACRO program with different machine coordinate.

G92.1	ROTATING PROGRAM COORDINATE SYSTEM SETTING	G92.1

Command form:

G92.1 X__ Y__ Z__ I__ J__ K__ R__;

X、Y 、Z: Set the position that work coordinate system (G92) in programmable coordinate system.

I、 J、K: Direction vector of an axis of rotation.

R: Angle of rotation.

Description:

This command will take the X, Y, Z filled value as new offset and rotate an angel R about the direction vector as a new coordinate system.

Example:

N1 G90 G00 X20. Y20.

// Machine coordinate X20. Y20.

// Program coordinate X20. Y20.

// Default of MACRO system variable #1901 #1902 coordinate offset is X0. Y0.

N2 G92.1 X10. Y10. K1. R45.

// Machine coordinate X20. Y20.

// Program coordinate X14.142 Y0.

// Set MACROsystem variable #1901 #1902 coordinate offset to X10. Y10.

// program coordinate X-Y plane rotate 45° about Z-axis

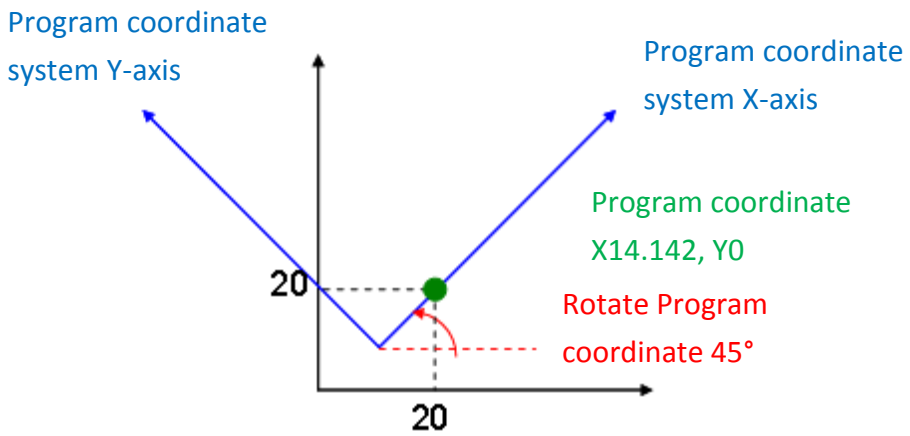


Figure 60: G92.1 rotating program coordinate system setting example

N3 G01 X100.

// Machine coordinate X80.711 Y80.711

// Program coordinate X100.0 Y0.0
// Coordinate offset X10. Y10.

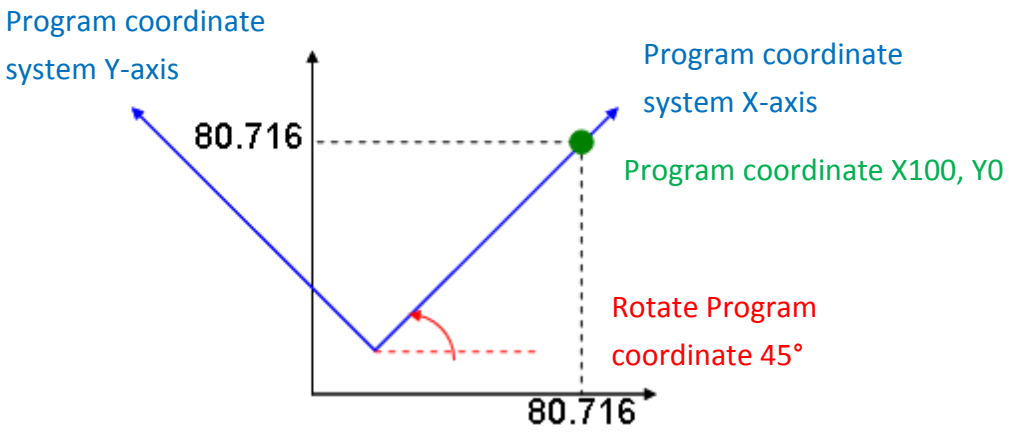


Figure 61: G92.1 rotating program coordinate system setting example (cont.)

N4 M30

G161	**COMPENSATION SETTING OF LINEAR INTERPOLATION**	G161

Command form:

G161 X__ Y__ Z__;

X: Compensation of linear interpolation X position.
Y: Compensation of linear interpolation Y position.
Z: Compensation of linear interpolation Z position.

Description:

After setting this linear compensation, when FBs-30GM performs G-code command (G01), tool will move with extra compensation value.
Compensation will be effective when the corresponding axis is specified.

Example:

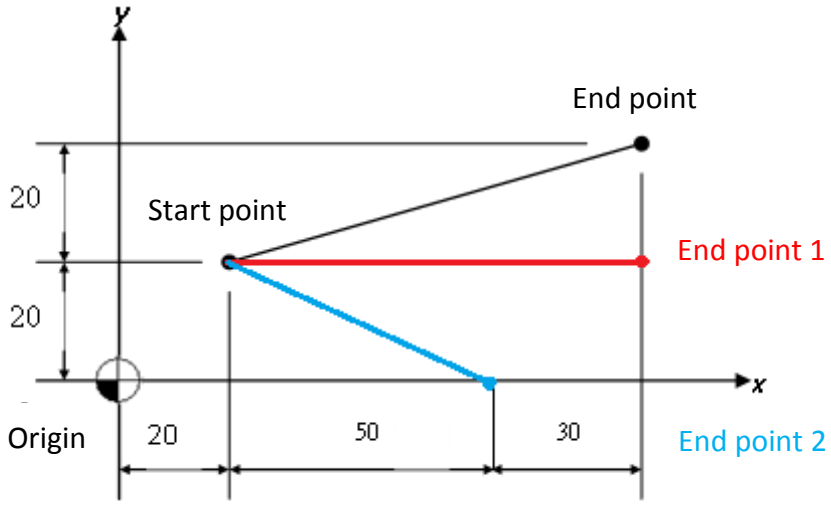


Figure 62: G161 linear interpolation compensation example

Uncompensated:

G90 G01 X100.0 Y40.0; //End point at X100.0 Y40.0

Set compensation:

case	G Code	Result
1	G90 G161 X-30.0 Y-20.0; G01 X130.0 Y40.0;	Move to end point 1.
2	G90 G161 X-30.0 Y-20.0; G01 X130.0;	Move to end point 1. Only X position compensation is effective.
3	G90 G161 X-30.0 Y-20.0; G01 X100.0 Y20.0;	Move to end point 2.

G162	VECTOR COMPENSATION SETTING OF CIRCULAR INTERPOLATION	G162

Command form:

G161 I__ J__ K__;

I, J, K: The vector compensation value that starting point of arc to the center of a circle (center of a circle－starting point)

Description:

After setting this vector compensation, when FBs-30GM performs G-code command (G02/G03), the compensation value will be added to the vector value.

Compensation will be effective when the corresponding component is specified.

Example:

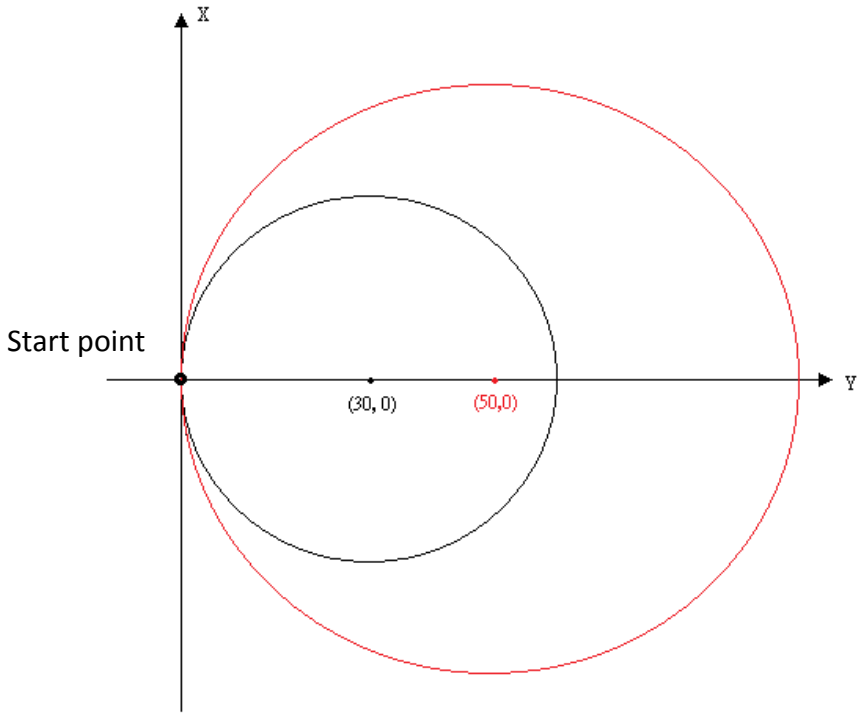


Figure 63: G162 vector compensation example

Uncompensated:

G17 G02 I30.0;

Set compensation:

G162 I20.0;

G17 G02 I30.0;

G163	**RADIUS COMPENSATION SETTING OF CIRCULAR INTERPOLATION**	G163

Command form:

G163 R;

R: Radius compensation value of arc

Description:

After setting this radius compensation, when FBs-30GM performs G-code command (G02/G03), the compensation value will be added to the radius of arc.

G164	**INTERPOLATION COMPENSATION CANCELLATION**	G164

Command form:

G164;

Cancel linear and circular compensation

Description:

Compensations about G01, G02 and G03 will be cleared.

G165	**ELECTRICAL ZERO POINT SETTING**	G165

Command form:

G165;

Record current X, Y, Z position as the electrical zero point.

Description:

Users can use G166 command to rapidly return to this point.

G166	**RETURN TO ELECTRICAL ZERO POINT**	G166

Command form:

G166;

Rapidly return to the electrical zero point

Description:

Move in the way of command G53.

Using this command requires setting the electrical zero point with command G165.

7.2 M code instructions

M code ancillary function is used to control machine function ON or OFF. The description is as below:

Table 17: M function table

M Code	Function
M01	Selectivity program dwell
M02	End program
M30	Program end, return to starting point
M98	Call the sub-program
M99	From sub-program return to main program

1. **M01: Selective program dwell**

   M01 is controlled by "optional stop”; when M1421 is ON, M01 is effective, program dwell; when the switch is OFF, then M01 is not effective.

2. **M02: End program**

   When there is M02 command in the end of main program. When FBs-30GM executes this command, machine will stop, if we need to execute the program again, we must perform "RESET", and then perform "program start".

3. **M30: Program end, return to starting point**

   M30 command is for end of program. When program execute M30 command, the program will stop all actions, and the memory will return to the initial of the program.

4. **M98/M99: sub-program control**

   A sub-program which has fixed performing method is executed usually, we prepare first and put it into memory, when we need to use, we can call by main program. We use M98 to call the sub-program and use M99 to end that.

   Command form:

M98 P__ H__ L__; //Sub-program called
P is specified number of program (ex. P1234 to motion program O1234)
H is the number of ranking in specified program.
L is the number of repeats that sub-program executes.

M99 P__; //Sub-program end
P is the line number that returns to main program after sub-program ends.

8. MACRO structure motion language

8.1 Introduction

To increase FBs-30GM application flexibility, FBs-30GM provide MACRO programmable function. After the machining program is declared as MACRO format, specific arithmetic operators can be used this way. The program will not only has simple motion control functions but logical and arithmetic operations.

8.2 File format

'%' is the head character and the first line is also called head line. If head line without keyword '@MACRO', statement at this file will process with standard ISO file. That means that file will not be able to use MACRO Syntax. Keyword '@MACRO' is all capitals characters. A semicolon ";" is required at the end of each line.

Example 1: MACRO file format

```
% @MACRO
IF @1 = 1 THEN
G00 X100.;
ELSE
G00 Y100.;
END_IF;
M99;
```

Example 2: ISO file format

```
%    //head line
G00 X100.;
G00 Y100.;
G00 X0;
G00 Y0;
M99;
```

8.3 Block format

Table 18: Block format list

/	N	G	X	Y	Z	I	J	K	F	M

/	Optional skip function (be effective when M1421 is ON)
N	If you use a sequence number, it must be the first in the block.
G	The preparatory function(s) G must follow N.
X	The linear dimension words follow G. Specify the X axis first.
Y	The linear dimension words follow G. Specify the Y axis second.
Z	The linear dimension words follow G. Specify the Z axis third.
I	The interpolation words follow the dimension words. Specify the X axis first.
J	The interpolation words follow the dimension words. Specify the Y axis second.
K	The interpolation words follow the dimension words. Specify the Z axis third.
F	It must follow the last dimension (and interpolation) to which it applies.
M	Any miscellaneous function(s) that you specify must last in the block, just ahead of the end of block character.

8.4 Operators

Table 19: Operator list

Operator	Symbol	Precedence
Parenthesis	( ) [ ]	1
Function Evaluation	Identifier (argument list)	2
Negative	-	3
Complement	NOT	3
Multiply	*	4
Divide	/	4
Modulus	MOD	4
Add	+	5
Subtract	-	5
Comparison	<,>,<=,>=	6
Equality	=	7

Inequality	<>	8
Boolean/Bitwise AND	&,AND	9
Boolean/Bitwise Exclusive OR	XOR	10
Boolean/Bitwise OR	OR	11

Note:

For operator "/", if the dividend and divisor are both integers, the result will be an integer

EX:

1.0 / 2 = 0.5

1/ 2.0 = 0.5

1/2 = 0

(1/2)*1.0 = 0

8.5 Statements

8.5.1 Assignment

Syntax: <Variable>: = <expression>;

Description: Assign a value to variable.

Example:

@1 := 123;

#1 := #3;

8.5.2 GOTO

Syntax: GOTO n;

Description: Jump to line numbers N

Example:

```
% @MACRO
#1 := 1;
#2 := 10;
G01 G90 X0. Y0. F1000;
IF( #1 = 1 ) THEN
   GOTO #2;
END_IF;
IF( #1 = 2 ) THEN
   GOTO 100;
```

```
END_IF;
N10 G01 G90 X50. Y0. F1000;
M30;
N100 G01 G90 X0. Y50. F1000;
M30;
```

8.5.3 CASE

Syntax:

```
CASE <INT expression> OF
<INT>:
   <Statement list>
<INT>, <INT>, <INT>:
   <Statement list>
<INT>,…<INT>:
   <Statement list>
ELSE
   <Statement list>
END_CASE;
```

Description: Conditional execution by cases. According to the result of INT expression in the CASE, FBs-30GM executes corresponding program block.

Example:

```
% @MACRO
#1 := 1;
G01 G90 X0. Y0. F1000;
CASE #1 OF
1:
   X(1.0*#1) Y(1.0*#1);
2:
   X(2.0*#1) Y(2.0*#1);
3, 4, 5:
   X(3.0*#1) Y(3.0*#1);
ELSE
   X(4.0*#1) Y(4.0*#1);
END_CASE;
M30;
```

8.5.4 IF

Syntax:

IF <Condition> **THEN**
  <Statement list>
ELSEIF <Condition> **THEN**
  <Statement list>
ELSE
<Statement list>
END_IF;

Description: conditional execution

Example:

```
% @MACRO
#1 := 3.0;
G01 G90 X0. Y0. F1000;
IF #1 = 1 THEN
   X(1.0*#1) Y(1.0*#1);
ELSEIF #1 = 2 THEN
   X(2.0*#1) Y(2.0*#1);
ELSEIF #1 = 3 THEN
   X(3.0*#1) Y(3.0*#1);
ELSE
   X(4.0*#1) Y(4.0*#1);
END_IF;
M30;
```

8.5.5 REPEAT

Syntax:

REPEAT
<Statement list>
UNTIL <Condition> **END_REPEAT**;

Description: REPEAT loop control

Example:

```
% @MACRO
#10 := 30.;
#11 := 22.5.;
#12 := #10/2;
#13 := #11/2;
#14 := 2.0;
#15 := 1.5;
G01 G90 X#12 Y#13 F1000;
REPEAT
   G00 X(#12+#14) Y(#13+#15);
   G01 X(#12+#14) Y(#13-#15);
   G01 X(#12-#14) Y(#13-#15);
   G01 X(#12-#14) Y(#13+#15);
   G01 X(#12+#14) Y(#13+#15);
   #14 := #14 + 2.0;
   #15 := #15 + 1.5;
UNTIL (#14 > #12) OR (#15 > #13) END_REPEAT;
M30;
```

8.5.6 WHILE

Syntax:

WHILE <Condition> **DO**
<Statement list>
END_WHILE;

Description: WHILE loop control

Example:

```
% @MACRO
#10 := 30.;
#11 := 22.5.;
#12 := #10/2;
#13 := #11/2;
#14 := 2.0;
#15 := 1.5;
G01 G90 X#12 Y#13 F1000;
WHILE (#14 <= #12) AND (#15 <= #13) DO
   G00 X(#12+#14) Y(#13+#15);
   G01 X(#12+#14) Y(#13-#15);
   G01 X(#12-#14) Y(#13-#15);
   G01 X(#12-#14) Y(#13+#15);
   G01 X(#12+#14) Y(#13+#15);
   #14 := #14 + 2.0;
   #15 := #15 + 1.5;
END_WHILE;
M30;
```

8.5.7 FOR

Syntax:

FOR <INT variable1> := <expression1> **TO** <expression2>
[BY <expression3>] DO <Statement list>
END_FOR;

Description: FOR loop control

variable1: loop control variable
expression1: loop start number, long or double
expression2: loop end number, long or double
expression3: loop increase(decrease)number, long or double
Statement list: execute statement

Example:

```
% @MACRO
#10 := 30.;
#11 := 22.5.;
#12 := #10/2;
#13 := #11/2;
#14 := 2.0;
#15 := 1.5;
G01 G90 X#12 Y#13 F1000;
FOR #6 := 0 TO 3 BY 1.0 DO
   G00 X(#12+#14) Y(#13+#15);
   G01 X(#12+#14) Y(#13-#15);
   G01 X(#12-#14) Y(#13-#15);
   G01 X(#12-#14) Y(#13+#15);
   G01 X(#12+#14) Y(#13+#15);
   #14 := #14 + 2.0;
   #15 := #15 + 1.5;
END_FOR;
M30;
```

8.5.8 EXIT

Syntax: EXIT;

Description: Break loop or exit jump control

Example:

```
% @MACRO
#10 := 30.;
#11 := 22.5.;
#12 := #10/2;
#13 := #11/2;
#14 := 2.0;
#15 := 1.5;
#16 := 1.0;
G01 G90 X#12 Y#13 F1000;
FOR #6 := 0 TO 3 BY 1.0 DO
   IF((#14 = 4) & (#16 = 1)) THEN
      EXIT;
   END_IF;
   G00 X(#12+#14) Y(#13+#15);
   G01 X(#12+#14) Y(#13-#15);
   G01 X(#12-#14) Y(#13-#15);
   G01 X(#12-#14) Y(#13+#15);
   G01 X(#12+#14) Y(#13+#15);
   #14 := #14 + 2.0;
   #15 := #15 + 1.5;
END_FOR;
M30;
```

8.5.9 Comment

Syntax:

(* < Statement list > ***)**

// <Statement list>

Description: Remark or explanation

Example1: Single line comment

```
% @MACRO
G00 G90 X0. Y0.; // Return to the origin
M30;
```

Example2: Block comment

```
% @MACRO
(*
This block is a comment.
The contents do not affect following program execution.
*)
G00 G90 X0. Y0.;
G00 G90 X10. Y0.;
G00 G90 X10. Y10.;
G00 G90 X0. Y10.;
G00 G90 X0. Y0.;
M30;
```

8.6 Functions listing

Table 20: Functions listing table

Function	Description

ABS	Calculates the absolute value of a number.
	Ex:
	#10 := -1.1;
	#1 := ABS(#10); // #1 = 1.1
	#2 := ABS(-1.2); // #2 = 1.2

ACOS	Calculates the arc cosine of a number.
	Ex:
	#10 := 1;
	#1 := ACOS(#10); // #1 = 0
	#2 := ACOS(-1); // #2 = 180

ASIN	Calculates the arc sine of a number.
	Ex:
	#10 := 1;
	#1 := ASIN(#10); // #1 = 90
	#2 := ASIN(-1); // #2 = -90

ATAN	Calculates the arc tangent of a number.
	Ex:
	#10 := 1;
	#1 := ATAN(#10); // #1 = 45
	#2 := ATAN(-1); // #2 = -45

CEIL	Return the smallest integer that is greater than or equal to a number.
	Ex:
	#10 := 1.4;
	#1 := CEIL(#10); // #1 = 2
	#2 := CEIL(1.5); // #2 = 2

COS	Calculates the cosine of a number.
	Ex:
	#10 := 180;
	#1 := COS(#10); // #1 = 1
	#2 := COS(-180); // #2 = -1

FLOOR	Return the largest integer that is less than or equal to a number.
	Ex:
	#10 := 1.4;
	#1 := FLOOR(#10); // #1 = 1
	#2 := FLOOR(1.5); // #2 = 1

GETARG	Read caller argument in subroutine.
	Ex:
	O0001 main program
	G101 X30. Y40. Z1=40. Z2=50.;
	
	G0101 extension G code macro
	#1 = GETARG(X); // the value of X argument will store in #1
	#2 = GETARG(Z1); // the value of Z1 argument will put in #2
	#3 = GETARG(W); // without W argument, #3 will be "VACANT"

GETTRAPARG	For G66/G66.1 modal macro call handler to get the block's information.
	Ex:
	O0001 main program
	G66 P100 X100. Y100.
	G01 X20.
	O0100 subroutine
	#1 := GETARG(X); // Get X argument 100. to #1
	#2 := GETTRAPARG(X); // Get the block X argument 20. to #2

MAX	Determines the maximum of two inputs.
	Ex:
	#10 := 1.2;
	#20 := 4.5;
	#1 := MAX(#10, #20); // #1 = 4.5
	#2 := MAX(-1.2, -4.5); // #2 = -1.2

MIN	Determines the minimum of two inputs.
	Ex:
	#10 := 1.2;
	#20 := 4.5;
	#1 := MIN(#10, #20); // #1 = 1.2
	#2 := MIN(-1.2, -4.5); // #2 = -4.5

PARAMETER	To read specified system parameter number. Ex: #1 := PARAM(3203); // To access interpolation time interval

POP	Pop value from Macro stack.
	Ex:
	PUSH(5); // push “5” into stack
	#1 := POP(); // popup a value to #1 (#1 = 5)

PUSH	Push value into Macro stack.
	Ex:
	PUSH(#1); // push #1 variable into stack
	PUSH(#3); // push #3 variable into stack

RANDOM	Generates a pseudorandom number. Ex: #1 := RANDOM();

ROUND	Return the value of the argument rounded to the nearest long value.
	Ex:
	#10 := 1.4;
	#1 := ROUND(#10); // #1 = 1
	#2 := ROUND(1.5); // #2 = 2

SCANTEXT	To scan text string from global variable. Notes: Because string is local, so only can stores in local variable, and cannot save to global variable. That is, following will get wrong result.

	Ex: % @MACRO @1:="12"; #1:=SCANTEXT(1); OPEN("NC"); PRINT("@1"); PRINT("#1"); CLOSE(); M30; (*The results: @1 = 12849 #1 = 12*)

SIGN	Return sign of a number, –1 for negative number, 1 for positive number, 0 for zero number.
	Ex:
	#10 := 4;
	#1 := SIGN(#10); // #1 = 1
	#2 := SIGN(-4); // #2 = -1
	#3 := SIGN(0); // #3 = 0

SIN	Calculate the sine of a number.
	Ex:
	#10 := 90;
	#1 := SIN(#10); // #1 = 1
	#2 := SIN(-90); // #2 = -1

SLEEP	Temporarily give up this cycle execution. Ex: SLEEP();

SQRT	Calculates the square root of a number.
	Ex:
	#10 := 4;
	#1 := SQRT(#10); // #1 = 2
	#2 := SQRT(9); // #2 = 3

STD	Standardize arguments, read a number, in argument one, by least increment method, in argument two, when necessary for decimal point programming.

	Ex: #9 := STD(#9,#1600); // normalize by distance axis (BLU)

STDAX	Standardize arguments, read a number, in argument one, by least increment method, in argument two is axis address.
	Ex:
	#24 := STDAX(#24,X); // normalize by X dimension
	#3 := STDAX(#3,A); // normalize by A dimension

STKTOP	Peek the stack value by index from top one.
	Ex:
	PUSH(5); // push 5 variable into stack
	PUSH(6); // push 6 variable into stack
	PUSH(7); // push 7 variable into stack
	#1 := STKTOP[0]; // #1 = 7
	#2 := STKTOP[1]; // #2 = 6
	#3 := STKTOP[2]; // #3 = 5

TAN	Calculates the tangent of a number.
	Ex:
	#10 := 45;
	#1 := TAN(#10); // #1 = 1
	#2 := TAN(-45); // #2 = -1

WAIT	Wait until all previous motion/logic commands are finished.
	Ex:
	% @MACRO // MACRO program
	G00 X0.; // G00 position to X0.0
	G01 X80.; // G01 linear interpolation to X80.0
	WAIT();
	// Wait until all previous motion/logic commands are finished.
	G01 X80.+@471;
	// G01 linear interpolation to X(80.0+@471)
	// Assign @471=20.0 before this single block is executed
	// After this block is executed, machine move to X100.0
	M30; // Program end
	
	Generally before executing a motion program, commands within the program will be pre-decoded in advance. Locus and endpoint of each single block are decided at this moment. By

	using WAIT() function to stop pre-decoding, after the start of the motion program, user can change the value of @471 before execution “G01 X80 + @ 471” block. The machine move to X(80.0 + @471) in the end.

8.7 Sub-program control

8.7.1 Call methods

Table 21: Call methods listing table

Syntax	Description	Examples
M98 P_ H_ L_	Subprogram call, P_ subroutine name H_ start N number L_ repeat times	M98 P10 L2;
G65 P_ L_	Macro call P_ subroutine name L_ repeat times	G65 P10 X10.0 Y10.0;
G66 P_ L_	Modal macro call, for every move block P_ subroutine name L_ repeat times	Example: G66 P10 X10.0 Y10.0; X20. Y20. Description: X20 and Y20. move command block will call O0010
G66.1 P_ L_	Modal macro call, for every block P_ subroutine name L_ repeat times	Example: G66.1 P10 X10.0 X20. G04 X2.; M31; Description: X20、G04 X2 and M31.every block will call O0010

8.7.2 Return methods

Table 22: Return methods listing table

Syntax	Description	Examples
M99	Return	M99;
M99 P_	Return and go to specified label P_ sequence number	M99 P100; Return to main program N100
M99 Q_	Return and go to specified line number Q_ line number	M99 Q100; Return to main program line100
G67	Modal macro call cancel	G67;

8.8 Variable specifications

MACRO variables can be divided into three types, local variables (Local variable, # 1 ~ # 400), system variables (System variable, # 1000 ~ # 31986), and public variables (Global variable, @ 1 ~ @ 165535). Different types of variables will have their different life cycles, as well as reading and writing rules. The following sections will have more detailed descriptions.

8.8.1 MACRO notices

1. Try to use local variables (#1 ~ #400) instead of global variables (@1 ~ @10495). Because of MACRO execution, the user's data are passed through the arguments (A_, B_, ..., Z_, X1 = _, Y1 = _, ...), but passed by global variables does not comply with user's usage.

2. Since the modal variables, #2001 ~ #2100, #3001 ~ #3080 will be reverted to VACANT state when the system is reset. Modal variables can be applied across multiple MACROs to exchange data and save shared resources.

3. When you execute MACRO, if you need to change mode G code (G91/G90, G17/G18/G19 ..., etc.) states, please backup its current states in the beginning and restore them to its original states before leaving MACRO.

4. After leaving the MACRO, if you still want to keep this MACRO interpolation mode (#1000), it is recommended to designate the interpolation mode to the MACRO program number before leaving MACRO program. Thereafter as long as encountering the axial displacement of the command block, the system will automatically call this MACRO program without specifying again. Of course, this MACRO interpolation mode will be automatically removed after encountering G00/G01 / G02/G03, or the content of # 1000 changes.

5. When performing motion program, system will predecode MACRO program, therefore MACRO execution speed is ahead of G/M-code instructions. So if specifying variables or reading data need to be synchronized with issuing G/M-code instructions, please add WAIT() instruction before specifying variables or reading data to ensure correct operation.

6. Being a sub-program, the MACRO program need to add "M99;" at the last line to return to the main program.

7. Please try to add more comment in the program to develop good habits, and this can help to increase the readability of the program and deal with follow-up maintenance or troubleshooting.

8.8.2 Global variable

Table 23: Global variable table

Variables	Description	Rule
@0	VACANT	R
@1 ~ @400	Normally arithmetic variables	R/W
@656 ~ @1999	Memorable variables(still exist when power off)	R/W
@120000~@165535	Corresponding to PLC register Registry R20000~R65535	R/W
Remark	All global variable lifetime will end when FBs-30GM is power off. If user wants to memorize @1 ~ @400 values, after shut down FBs-30GM, set Pr3811 for this function.	

Users please do not use other global variables that are not mentioned and have been used within the system to avoid system being abnormal.

8.8.3 Local variables

Table 24: Local variables listing

Variables.	Description	**Rule**
#0	VACANT	R
#1 ~ #400	Local variable for macro program	R/W
Remark	The local variables use in MACRO, the effective life time is only useful in MACRO executive process. When the execution is finish and escape from the program, the local variables will automatically become vacant. Sub-Program and main program can use the same local variable at the same time, the life time of variable ends along with the end of the main program.	

It is suitable to use local variables if operations need to be done in a MACRO program. When calling a MACRO program, FBs-30GM has its default addresses that can be used to store incoming arguments.

Table 25: Default argument specification

Address	Variable Number	Address	Variable Number	Address	Variable Number
A	#1	J	#5	U	#21
B	#2	K	#6	V	#22
C	#3	M	#13	W	#23
D	#7	P	#16	X	#24
E	#8	Q	#17	Y	#25
F	#9	R	#18	Z	#26
H	#11	S	#18		
I	#4	T	#20	X1	GETARG(X1)

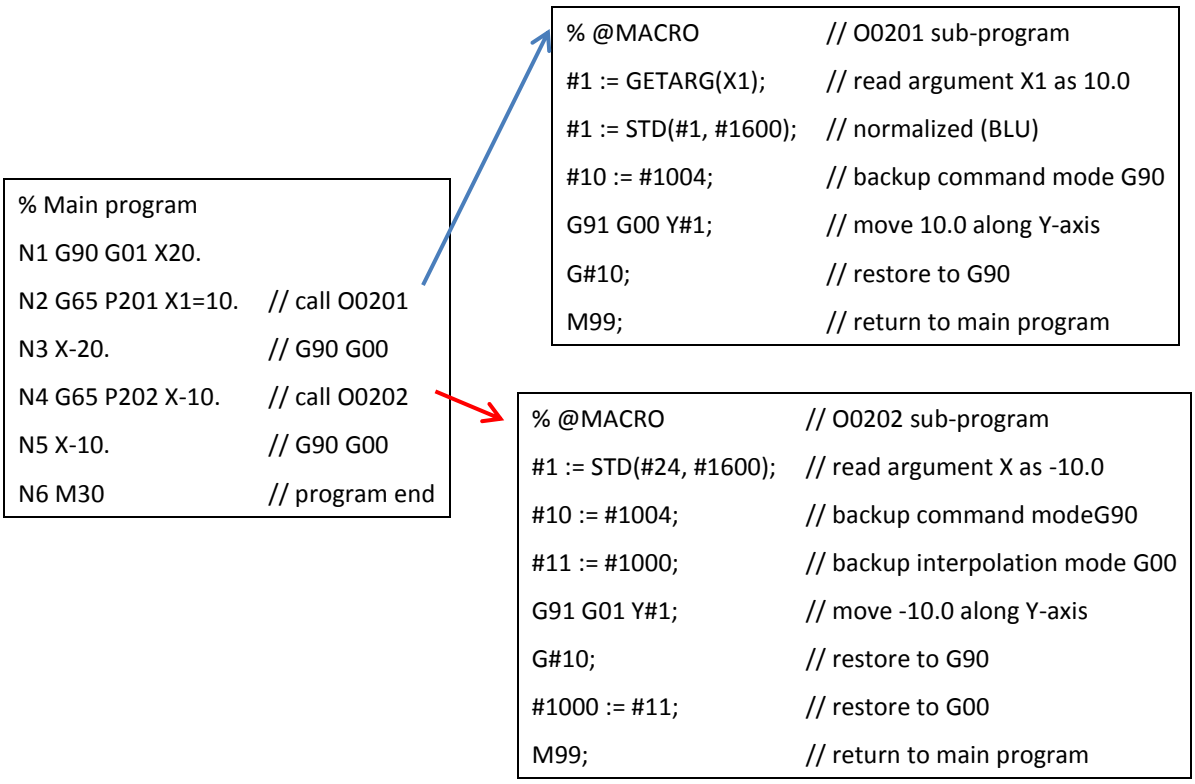
8.8.4 System variables

Table 26: System variables

No	Description	Rule
#1000	Interpolation mode, 00/01/02/03	R/W
#1002	Contouring plane selection mode, 17/18/19	R
#1004	Absolute/Incremental command mode, 90/91	R
#1010	Inch/Metric mode, 70/71	R
#1046	Feedrate command, F Code	R
#1048	Caller's current line number	R
#1050	Program start sequence number	R
#1301 ~ #1303	Block end position in program coordinate	R
#1321 ~ #1323	Current position of X, Y or Z-axis in machine coordinate, this value can't be read during movement.	R
#1341 ~ #1343	Current position of X, Y or Z-axis in program coordinate	R
#1600	Distance least input increment, refer to Pr17	R
#1602	Time/Rotation angle least input increment, refer to Pr17	R

8.8.5 MACRO example

- N1: Do linear interpolation with absolute command G90 and move to X20.0.
- N2: Call MACRO program O0201 and read caller argument X1 in subroutine.
  - After entering O0201, X1 is stored in the local variable #1.
  - Use #10 to backup absolute command mode G90.
  - Do positioning G00 with incremental command G91 and move 10.0 along Y-axis.
  - Restore to absolute command mode G90.
  - Return to main program.
- N3: Due to absolute command mode G90 and the last interpolation mode before leaving O0201 is G00 (#1000 = 0), this block shows the machine will move to X-20.0 with G00.
- N4: Call MACRO program O0202 and read argument X through #24.
  - After entering O0202, X is stored in the local variable #1.
  - Use #10 to backup absolute command mode G90.
  - Use #11 to backup interpolation mode G00.
  - Do linear interpolation G01 with incremental command G91 and move 10.0 along Y-axis.
  - Restore to absolute command mode G90.
  - Restore to interpolation mode G00.
  - Return to main program.
- N5: Do positioning G00 with absolute command G90 and move to X-10.0.
- N6: Program end



```
% Main program
N1 G90 G01 X20.
N2 G65 P201 X1=10.    // call O0201
N3 X-20.              // G90 G00
N4 G65 P202 X-10.     // call O0202
N5 X-10.              // G90 G00
N6 M30                // program end
```

```
% @MACRO                  // O0201 sub-program
#1 := GETARG(X1);         // read argument X1 as 10.0
#1 := STD(#1, #1600);     // normalized (BLU)
#10 := #1004;             // backup command mode G90
G91 G00 Y#1;              // move 10.0 along Y-axis
G#10;                     // restore to G90
M99;                      // return to main program
```

```
% @MACRO                  // O0202 sub-program
#1 := STD(#24, #1600);    // read argument X as -10.0
#10 := #1004;             // backup command modeG90
#11 := #1000;             // backup interpolation mode G00
G91 G01 Y#1;              // move -10.0 along Y-axis
G#10;                     // restore to G90
#1000 := #11;             // restore to G00
M99;                      // return to main program
```

9. Examples of motion program

9.1 S-curve

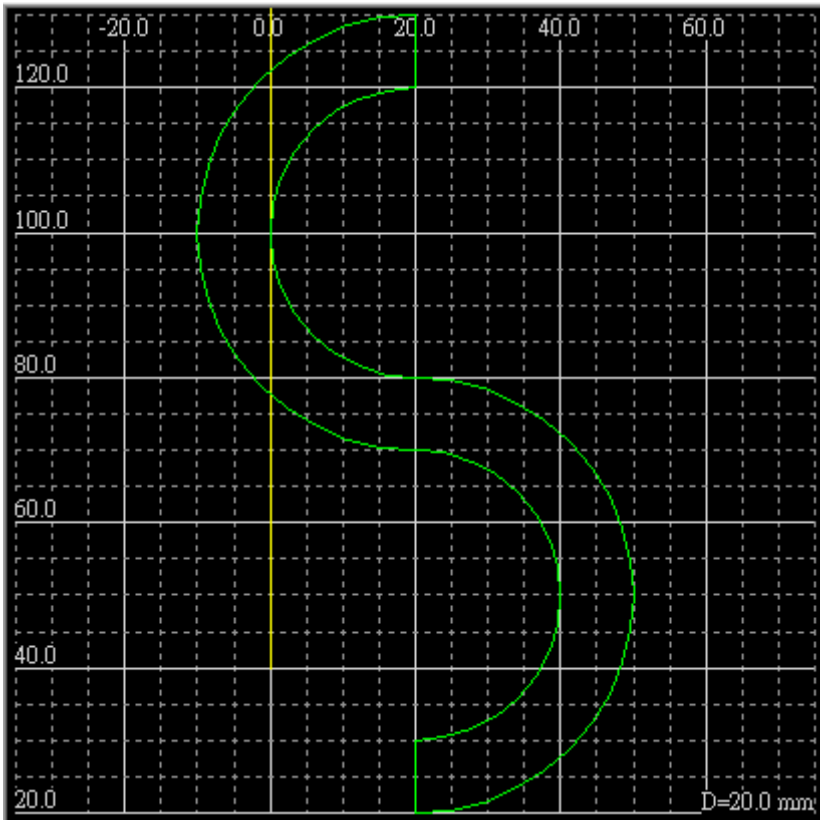


Figure 64: S-curve

Program description:

```
G90   G17;                          // set to absolute command and X-Y
                                    plane
G00   X20.0   Y20.0;                // positioning to (20,20)
G03   X20.0   Y80.0   R30.0   F500; // CCW circular interpolation to (20,80)
G02   X20.0   Y120.0   R20.0;       // CW circular interpolation to (20,120)
G01   Y130.0;                       // linear interpolation to (20, 130)
G03   X20.0   Y70.0   R30.0;        // CCW circular interpolation to (20,70)
G02   X20.0   Y30.0   R20.0;        // CW circular interpolation to (20,30)
G01   Y20.0;                        // linear interpolation to (20, 20)
M02;                                // Program end
```

9.2 Multi-speed control

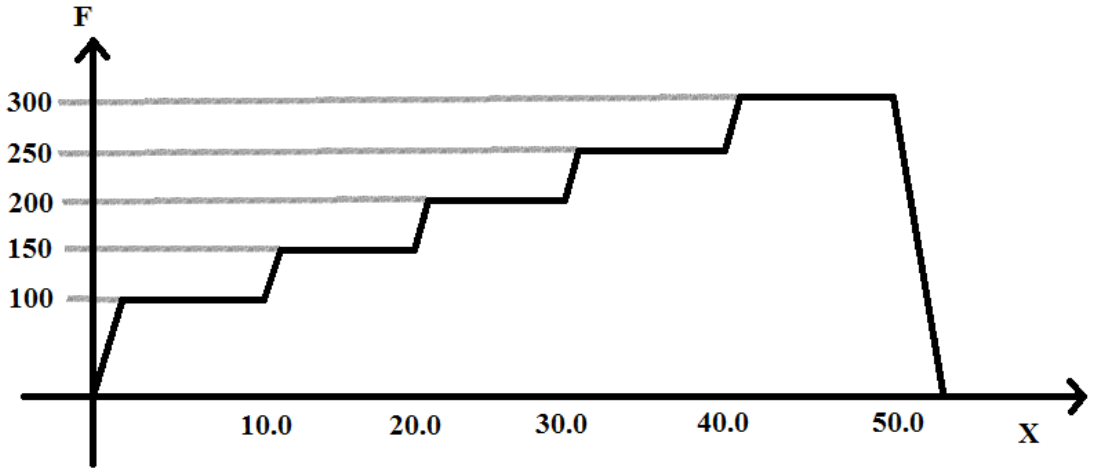


Figure 65: Multi-speed control

Program description:

```
G90;
G00   X0.0   Y0.0   Z0.0;
G01   X10.0   Y15.0   F100;
G01   X20.0   Y30.0   F150;
G01   X30.0   Y45.0   F200;
G01   X40.0   Y60.0   F250;
G01   X50.0   Y75.0   F300;
M02;
```

9.3 Coupling

Set Pr3825 to select coupling type.

0: Cancel coupling

1: Machine coupling, coupling starts from power on and can't be canceled.

2: PeerSynchronization coupling;

Coupling starts from power on and M1422 on. When M1422 is off, coupling is canceled.

FBs-30GM receives commands from the master axis or the slave axis and then sends to two axes at the same time.

3: Superimposition coupling

Coupling starts from power on and M1422 on. When M1422 is off, coupling is canceled.

Superimposition coupling is the slave axis superimpose on the master axis. When FBs-30GM receives commands from the master axis, both of the axes will move. When FBs-30GM receives commands from the slave axis, the slave axis will move relatively to the position of the master axis.

4: MasterSlaveSynchronization coupling

Coupling starts from power on and M1422 on. When M1422 is off, coupling is canceled.

MasterSlaveSynchronization coupling is FBs-30GM gets commands from the master axis and then sends to two axes to execute.

5: One to many coupling

Coupling starts from power on and M1422 on. When M1422 is off, coupling is canceled.

Similar to PeerSynchronization coupling, FBs-30GM receives commands from the master axis or the slave axes and sends to these axes to execute.

When Bit on, the axis is coupling.

Bit 1: X axis to carry 2

Bit 2: Y axis to carry 4

Bit 3: Z axis to carry 8

When Pr3822 is 12(12=4+8), the slave axes are Y-axis and Z-axis.

Note: When use one to many coupling, master axis ratio and slave axis ratio become 1:1. Settings of Pr3823 and Pr3824 are not useful.

9.4 Trigger input terminals to execute motion program

1. Prepare motion programs for external trigger function
2. The program files can be named from O1001 to O1009.
   (O1001 ~ O1009 correspond to the input terminal of FBs-30GM X0 ~ X8.)
3. Upload the motion program to FBs-30GM.
4. Set M1424 ON.
5. Trigger input terminals X0 ~ X8 to begin the corresponding motion programs O1001 ~ O1009.

(If you are currently running a motion program, FBs-30GM will directly switch to the corresponding motion program and start. After the program is finished, FBs30GM will switch back to the previous motion program and return to standby state.)

9.5 Dynamically change endpoint

Program description:

```
% @MACRO          // MACRO program
G00 X0.;          // G00 position to X0.0
G01 X80.;         // G01 linear interpolation to X80.0
WAIT();           // Wait until all previous motion/logic commands are finished.
G01 X80.+@471;    // G01 linear interpolation to X(80.0+@471)
                  // Assign @471=20.0 before this single block is executed
                  // After this block is executed, machine move to X100.0
M30;              // Program end
```

Generally before executing a motion program, commands within the program will be pre-decoded in advance. Locus and endpoint of each single block are decided at this moment. By using WAIT() function to stop pre-decoding, after the start of the motion program, user can change the value of @471 before execution “G01 X80 + @ 471” block. The machine move to X(80.0 + @471) in the end.

9.6 Sensor-triggered incremental displacement

Program description:

```
G00 X0.0;
G28.1 X10.0 Q30.0 R20.0 F1=1000 F2=200;
M02;
```

Move to X10.0 with the specified speed F1.
After reaching X, machine move to Q with the specified speed F2.

If FBs-30GM meets the optical sensor signal during the second part, FBs-30GM will immediately move 20.0 away from the sensor. Otherwise after the machine moves to Q, the execution of the block is completed

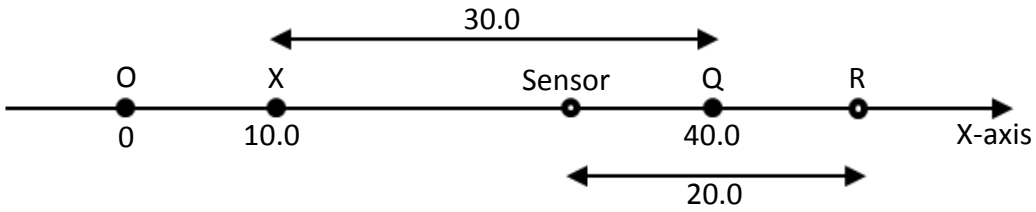


Figure 66: Sensor-triggered incremental displacement

Notice:

Please connect the optical sensor to the terminal of index signal.

Appendix I (Special relays and registers of FBs PLC)

FBs PLC series have special relays and registers to control or monitor the operation state of FBs-30GM. The detailed descriptions are listed in the tables below.

The special relays of FBs PLC can be divided into two types.

A. Control relays M1400 ~ M1430: These relays are for FBs PLC to control FBs-30GM.
B. State relays M1464 ~ M1474 and M1480 ~ M1488: These relays are for FBs PLC to monitor the operation state of FBs-30GM. Hence users can confirm the operation state of FBs-30GM by checking these state relays.

Special registers D3426 ~ D3431 store part of the operating parameters of FBs-30GMand their values can be modified through FBs PLC. Users can write data to specific user-defined global variables of MACRO by registers D3432 ~ D3434. Register D3435 can determine the output voltage of VO terminal. Registers D3440 ~ D3443 can read out data from specific user-defined global variables of MACRO.

Notice:
Relays M1400 ~ M1499 and registers D3401 ~ D3467 of FBs PLC are designed for the system of FBs-30GM. Users please do not use these registers for other purposes to avoid unpredictable behavior.

Table 27: Control relays of FBs PLC for FBs-30GM

Relay	Function	Description
M1400	Start	In AUTO mode, turn ON this relay can be used to start the motion program.
M1401	Feed Hold	In the process, turn ON this relay can be used to suspend the motion program.
M1402	Reset	Turn ON this relay to reset and stop the motion program.
M1403	X Axis JOG+	In JOG mode, turn ON the relay and the machine will move along the corresponding direction of axis. In HOME mode, turn ON the relay to trigger reference point searching of the corresponding axis.
M1404	X Axis JOG-	
M1405	Y Axis JOG+	
M1406	Y Axis JOG-	
M1407	Z Axis JOG+	
M1408	Z Axis JOG-	

M1409	X Axis MPG Selection	In MPG mode, if the corresponding axial relay is ON, the machine will go relative displacement according to the hand wheel input.
M1410	Y Axis MPG Selection	
M1411	Z Axis MPG Selection	
M1412	MPG Simulation	In Auto mode, when this relay is ON, after starting the motion program, G00, G01, G02 and G03's FEEDRATE OVERRIDE MPG determined by the rotational speed. The faster the rotation, the faster the machine movement. MPG stops, the machine stops. It is suitable for processing test of machine.
M1413	RESET X Axis Machine Position	Set current position to zero as the corresponding axial machine coordinate origin. Suited for test processing and adjust the machine coordinate. If used during processing, it may cause the machine coordinates incorrect.
M1414	RESET Y Axis Machine Position	
M1415	RESET Z Axis Machine Position	
M1416	Single Block	When this relay is ON, FBs-30GM stops after a BLOCK of G-CODE is finished. Users have to set Start to start doing next BLOCK。
M1417	Optional Skip	When this relay is ON, if there is a skip sign " \ " in process program, it will skip this line and do next BLOCK.
M1418	X axis Machine Lock	When this relay is ON, the program will run, but the X-axis does not move. It is usually used for program checking.
M1419	Y axis Machine Lock	When this relay is ON, the program will run, but the Y-axis does not move. It is usually used for program checking.
M1420	Z axis Machine Lock	When this relay is ON, the program will run, but the Z-axis does not move. It is usually used for program checking.
M1421	Optional Stop	When this relay is ON, the program will pause if it

		encounters “M01” during processing. When this relay is OFF, it will skip this line.
M1422	Axis Coupling Request	This relay enables or disables coupling. When Pr3825 is 2, 3, 4 or 5, and if M1422 is ON, coupling is enabled. If M1422 is OFF, coupling is disabled.
M1423	Stroke Limit Two Switch	The second software travel limit switch. 0: Without second software travel limit 1: With second software travel limit Please refer to parameters 2441 - 2446 for further instructions.
M1424	FBs-30GM launch	FBs-30GM triggers the execution of motion programs. 0: Disable 1: Enable to trigger the execution of motion programs directly from FBs-30GM.
M1425	Drive FBs-30GM DO (Y0)	Control Y0 of FBs-30GM. 0: output transistor OFF. 1: output transistor ON.
M1426	Drive FBs-30GM DO (Y1)	Control Y1 of FBs-30GM. 0: output transistor OFF. 1: output transistor ON.
M1427	Drive FBs-30GM DO (Y2)	Control Y2 of FBs-30GM. 0: output transistor OFF. 1: output transistor ON.
M1428	Drive FBs-30GM DO (Y3)	Control Y3 of FBs-30GM. 0: output transistor OFF. 1: output transistor ON.
M1429	Drive FBs-30GM DO (Y4)	Control Y4 of FBs-30GM. 0: output transistor OFF. 1: output transistor ON.
M1430	Drive FBs-30GM DO (Y5)	Control Y5 of FBs-30GM. 0: output transistor OFF. 1: output transistor ON.

Table 28: State relays of FBs PLC for FBs-30GM

Relay	Function	Description
M1464	Start Light	This relay is ON when the motion program is processing.

M1465	Feed Hold Light	This relay is ON when the motion program is paused.
M1466	Block Stop	This relay is ON when the motion program is in block stop.
M1467	Ready	This relay will be ON after FBs-30GM boots up completely.
M1468	X Axis Busy	When the corresponding axial relay is ON indicates that the axis manual functions (hand wheel / JOG / Home) are running, FBs-30GM cannot accept new manual commands. When the corresponding relay is OFF indicates that the axial axis in the Idle state, allowing accepted new manual commands.
M1469	Y Axis Busy	
M1470	Z Axis Busy	
M1471	X Axis Home OK	After returning HOME, the corresponding axial relay will be ON, stroke limit of each axis will be activated from then. Users should notice that if these relays are not ON, you should not start motion program.
M1472	Y Axis Home OK	
M1473	Z Axis Home OK	
M1474	Alarm	When ALARM occurs, FBs-30GM will stop and this relay will be ON.
M1480	FBs-30GM DI Status (X0)	The state of input terminal X0. 0: Input transistor OFF; 1: ON.
M1481	FBs-30GM DI Status (X1)	The state of input terminal X1. 0: Input transistor OFF; 1: ON.
M1482	FBs-30GM DI Status (X2)	The state of input terminal X2. 0: Input transistor OFF; 1: ON.
M1483	FBs-30GM DI Status (X3)	The state of input terminal X3. 0: Input transistor OFF; 1: ON.
M1484	FBs-30GM DI Status (X4)	The state of input terminal X4. 0: Input transistor OFF; 1: ON.
M1485	FBs-30GM DI Status (X5)	The state of input terminal X5. 0: Input transistor OFF; 1: ON.
M1486	FBs-30GM DI Status (X6)	The state of input terminal X6. 0: Input transistor OFF; 1: ON.
M1487	FBs-30GM DI Status (X7)	The state of input terminal X7. 0: Input transistor OFF; 1: ON.
M1488	FBs-30GM DI Status (X8)	The state of input terminal X8. 0: Input transistor OFF; 1: ON.

Table 29: Special registers of FBs PLC for FBs-30GM

Register No.	Function	Description	Remark
D3426	Mode selection	This register can be used to select the operation mode of FBs-30GM. 0: default(Auto) 2: Auto 4: JOG 6: MPG 7: HOME	
D3427	MPG Override	MPG step percentage speed % 0: x100(default) 1: x1 2: x10 3: x100 4: Set to the value of Pr2001	
D3428	Feedrate Override	G01, G02 and G03 feedrate override percentage % 0: default(=10) 1: 10% 2: 20% …… 20: 200% When Pr3207 = 2, the percentage is set as the above specifications. Example: D3428 = 5 means 50%. When Pr3207 = 1, the percentage is equal to the value of this Register. Example: D3428 = 5 means 5%.	
D3429	JOG Override	JOG override percentage % 0: default(=10) 1: 10% 2: 20% ……	

		20: 200% When Pr3207 = 2, the percentage is set as the above specifications. Example: D3428 = 5 means 50%. When Pr3207 = 1, the percentage is equal to the value of this Register. Example: D3428 = 5 means 5%.	
D3430	Rapid Traverse Override	G00 rapid traverse override percentage 0: 100% 1: 0% (equal to Pr501 ~ Pr503) 2: 25% 3: 50% 4: 100% When Pr3207 = 2, the percentage is set as the above specifications. Example: D3430 = 1 means that is equal to the setting of Pr501 ~ Pr503. When Pr3207 = 1, the percentage is equal to the value of this Register. Example: D3428 = 10 means 10%. (If the percentage is less than 10, the rapid traverse override percentage is 10%).	

D3431	Motion program Number	Motion program number specified. This Register is used to specify the number of motion programs to be executed. Range: 1 to 9999 Activate method: reset	
D3432	User define input	Corresponds to FBs-30GM MACRO global variable @471.	Write only
D3433	User define input	Corresponds to FBs-30GM MACRO global variable @472.	Write only
D3434	User define input	Corresponds to FBs-30GM MACRO global variable @473.	Write only
D3435	Control VO value. Range: 0 ~ 20000 VO range: -10V ~ +10 V	VO output voltage adjustment. Range from 0 ~ 20000 correspond to -10V ~ +10 V.	Write only
D3440	User define output	Corresponds to FBs-30GM MACRO global variable @476.	Read only
D3441	User define output	Corresponds to FBs-30GM MACRO global variable @477.	Read only
D3442	User define output	Corresponds to FBs-30GM MACRO global variable @478.	Read only
D3443	User define output	Corresponds to FBs-30GM MACRO global variable @479.	Read only

Appendix II (FBs-30GM Motion parameters)

I. Motion parameters listing

Table 30: Motion parameters listing table

Index	No	Description
1	Pr15	I/O board digital filter type
2	Pr17	Control precision
3	Pr41	X axis motor command polarity
4	Pr42	Y axis motor command polarity
5	Pr43	Z axis motor command polarity
6	Pr61	X axis encoder resolution
7	Pr62	Y axis encoder resolution
8	Pr63	Z axis encoder resolution
9	Pr81	X axis encoder feedback scaling factor
10	Pr82	Y axis encoder feedback scaling factor
11	Pr83	Z axis encoder feedback scaling factor
12	Pr121	X axis gear number at the ballscrew side
13	Pr122	X axis gear number at the motor side
14	Pr123	Y axis gear number at the ballscrew side
15	Pr124	Y axis gear number at the motor side
16	Pr125	Z axis gear number at the ballscrew side
17	Pr126	Z axis gear number at the motor side
18	Pr161	X axis pitch of the ballscrew
19	Pr162	Y axis pitch of the ballscrew
20	Pr163	Z axis pitch of the ballscrew
21	Pr181	X axis loop gain of the position loop (1/sec)
22	Pr182	Y axis loop gain of the position loop (1/sec)
23	Pr183	Z axis loop gain of the position loop (1/sec)
24	Pr201	X axis sensor type
25	Pr202	Y axis sensor type
26	Pr203	Z axis sensor type
27	Pr221	X servo axis type
28	Pr222	Y servo axis type
29	Pr223	Z servo axis type
30	Pr241	X axis dual feedback related to port no.
31	Pr242	Y axis dual feedback related to port no.

32	Pr243	Z axis dual feedback related to port no.
33	Pr261	X axis dual feedback resolution
34	Pr262	Y axis dual feedback resolution
35	Pr263	Z axis dual feedback resolution
36	Pr301	X axis dual feedback scaling factor
37	Pr302	Y axis dual feedback scaling factor
38	Pr303	Z axis dual feedback scaling factor
39	Pr401	Cutting acceleration time
40	Pr402	Acceleration accelerated to 1G time
41	Pr404	Post cutting bell-shaped acceleration time
42	Pr405	Maximum cutting feedrate
43	Pr406	Maximum corner reference feedrate
44	Pr408	Arc cutting reference feedrate at radius 5 mm
45	Pr410	MPG acceleration time
46	Pr411	Rapid Travel G00
47	Pr413	Reserve local coordinate G92(G92.1) after reset
48	Pr414	Reserve Workpiece Coordinate System after reset
49	Pr421	X axis cutting in-position window
50	Pr422	Y axis cutting in-position window
51	Pr423	Z axis cutting in-position window
52	Pr441	X axis rapid travel (G00) acceleration time
53	Pr442	Y axis rapid travel (G00) acceleration time
54	Pr443	Z axis rapid travel (G00) acceleration time
55	Pr461	X axis max. rapid travel (G00) feedrate
56	Pr462	Y axis max. rapid travel (G00) feedrate
57	Pr463	Z axis max. rapid travel (G00) feedrate
58	Pr481	X axis rapid travel in-position window (G09)
59	Pr482	Y axis rapid travel in-position window (G09)
60	Pr483	Z axis rapid travel in-position window (G09)
61	Pr501	X axis rapid travel (G00) F0 feedrate
62	Pr502	Y axis rapid travel (G00) F0 feedrate
63	Pr503	Z axis rapid travel (G00) F0 feedrate
64	Pr521	X axis JOG feedrate
65	Pr522	Y axis JOG feedrate
66	Pr523	Z axis JOG feedrate
67	Pr541	X axis cutting acceleration time

68	Pr542	Y axis cutting acceleration time
69	Pr543	Z axis cutting acceleration time
70	Pr561	X axis loss pulse check window
71	Pr562	Y axis loss pulse check window
72	Pr563	Z axis loss pulse check window
73	Pr581	X axis velocity feed forward percentage
74	Pr582	Y axis velocity feed forward percentage
75	Pr583	Z axis velocity feed forward percentage
76	Pr601	X axis corner reference feedrate (mm/min)
77	Pr602	Y axis corner reference feedrate (mm/min)
78	Pr603	Z axis corner reference feedrate (mm/min)
79	Pr621	X axis maximum cutting feedrate (G01)
80	Pr622	Y axis maximum cutting feedrate (G01)
81	Pr623	Z axis maximum cutting feedrate (G01)
82	Pr641	X axis cutting bell-shaped acceleration time
83	Pr642	Y axis cutting bell-shaped acceleration time
84	Pr643	Z axis cutting bell-shaped acceleration time
85	Pr661	X axis MPG feedrate
86	Pr662	Y axis MPG feedrate
87	Pr663	Z axis MPG feedrate
88	Pr821	X axis speed of first part homing
89	Pr822	Y axis speed of first part homing
90	Pr823	Z axis speed of first part homing
91	Pr841	X axis speed of second part homing
92	Pr842	Y axis speed of second part homing
93	Pr843	Z axis speed of second part homing
94	Pr861	X axis negative homing direction
95	Pr862	Y axis negative homing direction
96	Pr863	Z axis negative homing direction
97	Pr881	X axis home offset
98	Pr882	Y axis home offset
99	Pr883	Z axis home offset
100	Pr901	X axis zero speed check window
101	Pr902	Y axis zero speed check window
102	Pr903	Z axis zero speed check window
103	Pr921	X axis home dog polarity

104	Pr922	Y axis home dog polarity
105	Pr923	Z axis home dog polarity
106	Pr941	Enable X axis home grid function
107	Pr942	Enable Y axis home grid function
108	Pr943	Enable Z axis home grid function
109	Pr961	Home mode of X axis
110	Pr962	Home mode of Y axis
111	Pr963	Home mode of Z axis
112	Pr981	X axis homing 2nd protect revolution (encoder type)
113	Pr982	Y axis homing 2nd protect revolution (encoder type)
114	Pr983	Z axis homing 2nd protect revolution (encoder type)
115	Pr1001	X axis fast home return function
116	Pr1002	Y axis fast home return function
117	Pr1003	Z axis fast home return function
118	Pr1221	X axis backlash compensation start
119	Pr1222	Y axis backlash compensation start
120	Pr1223	Z axis backlash compensation start
121	Pr1241	X axis G00 backlash compensation value (BLU)
122	Pr1242	Y axis G00 backlash compensation value (BLU)
123	Pr1243	Z axis G00 backlash compensation value (BLU)
124	Pr1261	X axis G01 backlash compensation value (BLU)
125	Pr1262	Y axis G01 backlash compensation value (BLU)
126	Pr1263	Z axis G01 backlash compensation value (BLU)
127	Pr1281	X axis backlash critical speed (mm/min)
128	Pr1282	Y axis backlash critical speed (mm/min)
129	Pr1283	Z axis backlash critical speed (mm/min)
130	Pr1301	X axis pitch error compensation type
131	Pr1302	Y axis pitch error compensation type
132	Pr1303	Z axis pitch error compensation type
133	Pr1321	X axis pitch error compensation Interval (BLU)
134	Pr1322	Y axis pitch error compensation Interval (BLU)
135	Pr1323	Z axis pitch error compensation Interval (BLU)
136	Pr1341	X axis table index for reference (home)
137	Pr1342	Y axis table index for reference (home)
138	Pr1343	Z axis table index for reference (home)
139	Pr1401	X axis mechanical compensation time constant (ms)

140	Pr1402	Y axis mechanical compensation time constant (ms)
141	Pr1403	Z axis mechanical compensation time constant (ms)
142	Pr1421	X axis max. static dual feedback error (BLU)
143	Pr1422	Y axis max. static dual feedback error (BLU)
144	Pr1423	Z axis max. static dual feedback error (BLU)
145	Pr2001	MPG 4th scaling factor
146	Pr2041	MPG resolution (Pulse/rev)
147	Pr2051	MPG scaling factor
148	Pr2401	X axis 1st Software travel limit (positive direction)
149	Pr2402	X axis 1st Software travel limit (negative direction)
150	Pr2403	Y axis 1st Software travel limit (positive direction)
151	Pr2404	Y axis 1st Software travel limit (negative direction)
152	Pr2405	Z axis 1st Software travel limit (positive direction)
153	Pr2406	Z axis 1st Software travel limit (negative direction)
154	Pr2441	X axis 2nd Software travel limit (positive direction)
155	Pr2442	X axis 2nd Software travel limit (negative direction)
156	Pr2443	Y axis 2nd Software travel limit (positive direction)
157	Pr2444	Y axis 2nd Software travel limit (negative direction)
158	Pr2445	Z axis 2nd Software travel limit (positive direction)
159	Pr2446	Z axis 2nd Software travel limit (negative direction)
160	Pr2481	2nd software limit persistency
161	Pr2801	X axis 2nd reference point
162	Pr2802	Y axis 2nd reference point
163	Pr2803	Z axis 2nd reference point
164	Pr2821	X axis 3rd reference point
165	Pr2822	Y axis 3rd reference point
166	Pr2823	Z axis 3rd reference point
167	Pr2841	X axis 4th reference point
168	Pr2842	Y axis 4th reference point
169	Pr2843	Z axis 4th reference point
170	Pr3202	I/O scan time
171	Pr3203	Interpolation time interval
172	Pr3207	Feedrate override selection
173	Pr3221	Debug level
174	Pr3241	Decimal point type
175	Pr3805	Static dual feedback error timeout

176	Pr3807	Destination not on arc check window (BLU)
177	Pr3811	Start address of persist working global variable
178	Pr3817	Fatal dual feedback error
179	Pr3818	Dual feedback self-detect error (pulse)
180	Pr3821	Coupling master axis number
181	Pr3822	Coupling slave axis number
182	Pr3823	Coupling master axis ratio factor
183	Pr3824	Coupling slave axis ratio factor
184	Pr3825	Coupling type
185	Pr3826	Coupling couple time (ms)
186	Pr3827	Coupling decouple time (ms)
187	Pr3837	Initial Command Mode
188	Pr8001 ~ 8100	X axis positive direction pitch error compensate, compensation table 1 ~ 100
189	Pr8101 ~ 8200	X axis negative direction pitch error compensate, compensation table 1 ~ 100
190	Pr8201 ~ 8300	Y axis positive direction pitch error compensate, compensation table 1 ~ 100
191	Pr8301 ~ 8400	Y axis negative direction pitch error compensate, compensation table 1 ~ 100
192	Pr8401 ~ 8500	Z axis positive direction pitch error compensate, compensation table 1 ~ 100
193	Pr8501 ~ 8600	Z axis negative direction pitch error compensate, compensation table 1 ~ 100

II. Descriptions of motion parameters

No	Descriptions	Range	Unit	Initial	Activate method
15	I/O board digital filter type	[0 ~ 3]	-	3	reset

■ I/O board digital filter type, the larger value is better to filter the noise, but also reduce the sensitivity of the I/O Signal.

■ 0:

The system input state is on➔ If the off signal get in, checking the next two signals. If either signal is off, the system input state is changed to off.

The system input state is off ➔ If the off signal gets in, checking the two signals behind it. If either signal is on, the system input state is changed to on.

■ 1:

The system input state is on➔ If the off signal gets in, checking the next signal. If signal is off, the system input state is changed to off.

The system input state is off➔ If the on signal gets in, checking the next signal. If signal is on, the system input state is changed to on.

■ 2:

The system input state is on➔ If the off signal gets in, checking the next two signals. If both of signals are off, the system input state is changed to off.

The system input state is off➔ If the on signal gets in, checking the next two signals. If both of signals are on, the system input state is changed to on.

■ 3:

The system input state is on➔ If the off signal gets in, checking the next four signals. If all of signals are off, the system input state is changed to off.

The system input state is off➔ If the on signal gets in, checking the next four signals. If all of signals are on, the system input state is changed to on.

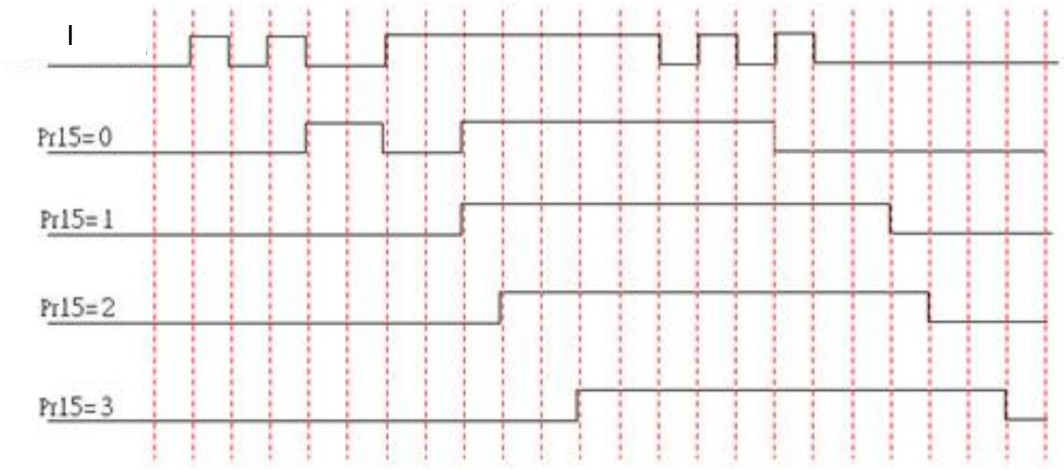


Figure 67: I/O board digital filter

No	Descriptions	Range	Unit	Initial	Activate method
17	Control precision	[1 ~ 3]	-	2	restart

- Set the parameter to Control precision (BLU):
  1: 0.001 inch / 0.01 mm / 0.01 deg;
  2: 0.0001 inch / 0.001 mm / 0.001 deg;
  3: 0.00001 inch / 0.0001 mm / 0.0001 deg.
- It would not be affected by imperial system.
- When the parameter is changed, all of the parameters that relate BLU have to change.

No	Descriptions	Range	Unit	initial	Activate method
41 ~ 43	Axis motor command polarity	[0 ~ 1]	-	0	reset

- The definition of motor rotation direction to the machine movement:
  0: Same;
  1: Reverse the direction.
- If the direction of machine movement is reverse the direction of command, set the parameter to revise the command.

No	Descriptions	Range	Unit	initial	Activate method
61 ~ 63	Axis encoder resolution	[10 ~ 2500000]	-	1250	reset

- If encoder is used, setting unit is pulse/rev; if ruler is used, setting unit is pulse/mm. Note that this setting value is resolution for single phase (A or B phase) before frequency multiplication.
- Assume that the ruler resolution is 1um/pulse (i.e., 1mm/1000pulse), with encoder scaling factor of 4 (Pr8x=4). Thus, this parameter shall set to (1000/4) =250.
- Assume that the ruler resolution is 10um/pulse (i.e., 1mm/100pulse), with encoder scaling factor of 4 (Pr8x=4). Thus, this parameters shall set to (100/4) =25.

No	Descriptions	Range	Unit	initial	Activate method
81 ~ 83	Axis encoder scaling factor	[1 ~ 4]	-	4	reset

- Encoder feedback gain of the servo board can set to 1, 2, or 4.

No	Descriptions	Range	Unit	initial	Activate method
121 ~ 126	Gear number at the ballscrew side. Gear number at the motor	[1 ~ 999999999]	-	1	reset

	side.				

- Gear number at the ballscrew side, Gear number at the motor side:
- System can decide the speed rate by the parameters.
- Ex: Gear number at the ballscrew side: Gear number at the motor side = 2:1è Motor speed: ballscrew = 2:1

No	Descriptions	Range	Unit	initial	Activate method
161 ~ 163	Pitch of the ballscrew	[1 ~ 1000000]	BLU	5000	reset

- Pitch of the ballscrew:
- Ballscrew rotate a revolution that move value of linear. (When change the Pr17, this parameter have to change.)

No	Descriptions	Range	Unit	initial	Activate method
181 ~ 183	Loop Gain of the position loop	[1 ~ 1000000]	1/sec	30	reset

- Loop Gain of the position loop for servo system:
  1. For each corresponding axis direction, the parameter setting value should be the same as loop gain of the position loop for driver. (Suggest every feed axis should be the same)
  2. System can compute reasonable servo following error by the parameter setting value. When output signal is pulse (driver is position control), the parameter setting value is only for system monitoring motor motion is OK or not.

  When output signal is voltage (driver is velocity control), the parameter setting value is loop gain of the position loop for servo system.

  When the parameter setting value bigger than 1000, system will input original parameter value divided by 1000. Otherwise, system will input original parameter value. (EX: 78500 divided by 100 becomes 78.5). Please refer to debug variables No.352 ~ No.354. When stable state, No.352 ~ No.354 are real loop gain of the position loop.

- When System sends pulse commands (Pr381 ~ ≠1), the parameter means:

According to the formula, $F_e = \frac{V_{cmd}}{K_p(Pr181 \sim)}$, calculate ideal following error (System debug variable No.32 ~ No.34) and real following error (System debug variable No.8

~ No.10).If the difference is too big, FBs-30GM will alarm “Fatal following error exceed”.

If the feed forward turn on, FBs-30GM will calculate by the parameter then send compensation to decrease the following error.

■ System sends voltage command(Pr381 ~ Pr383＝1). If tool rigidity is better, the value of the parameter could set bigger and make higher precision. If the value is too big, the tool will shake.

No	Descriptions	Range	Unit	initial	Activate method
201 ~ 203	Axis sensor type	[0 ~ 2]	-	0	restart

■ This parameter is used to define the encoder feedback type

0: Incremental encoder

1: Optical linear encoder

2: No feedback

No	Descriptions	Range	Unit	initial	Activate method
221 ~ 223	Type of servo axis	[0 ~ 5]	-	0	reset

■ Set the parameter is 0 : (linear axis)

1. Machine coordinate and absolute coordinate are linear axes.
2. Metric coordinate and inch coordinate transform.
3. G28 and G30 (reference coordinate instruct) will go back the machine origin.
4. It is useful in backlash compensation and quad-peak error compensation and home grid function.

■ Set the parameter is 1: (Rotary axis A)

Machine coordinate and absolute coordinate are rotary axes.

Coordinate value is between 0 ~ 360 degree.

The sign +/- is the direction of absolute coordinate (G90) moving instruct.

The unit in Metric coordinate system and inch coordinate system both are degree.

G28 and G30 (reference coordinate instruct) will go back to the machine origin that rotates in a revolution.

It’s useful in backlash compensation and quad-peak error compensation and home grid function

Absolute coordinate (G90) moving instruction is automatic to choose the shortest path.

■ Set the parameter is 2: (Rotary axis B)

Machine coordinate and absolute coordinate are rotary axes.
Coordinate value is between 0 ~ 360 degree.
The sign +/- is the direction of absolute coordinate (G90) moving instruct. + rotate positive direction and – rotate negative direction.
The unit in Metric coordinate system and inch coordinate system both are degree.
G28 and G30 (reference coordinate instruct) will go back the machine origin that rotates in a revolution.
It's useful in backlash compensation and quad-peak error compensation and home grid function

■ Set the parameter is 3: (Rotary axis C)

Machine coordinate and absolute coordinate are rotary axes.
Coordinate value is between -360 ~ 360 degree.
The unit in Metric coordinate system and inch coordinate system both are degree.
G28 and G30 (reference coordinate instruct) will go back the machine origin that rotates in a revolution.
It's useful in backlash compensation and quad-peak error compensation and home grid function

■ Set the parameter is 4: (Rotary axis D)

Machine coordinate is rotary axis and absolute coordinate is linear axis.
Coordinate value is between 0 ~ 360 degree.
The unit in Metric coordinate system and inch coordinate system both are degree.
G28 and G30 (reference coordinate instruct) will go back the machine origin.
It's useful in backlash compensation and quad-peak error compensation and home grid function

■ Set the parameter is 5: (Rotary axis E)

Machine coordinate and absolute coordinate are linear axes.
The unit in Metric coordinate system and inch coordinate system both are degree.
G28 and G30 (reference coordinate instruct) will go back the machine origin.
It's useful in backlash compensation and quad-peak error compensation and home grid function

Table 31: Type of servo axis setting

Setting value	1	2	4	5	3（Note 1）
Workpiece coordinate display	0~+360°		0~±360000°		0~±360°，over ±360° back to 0°
Machine coordinate display	0~+360°			0~±360000°	0~±360°，over ±360° back to 0°
Absolute instruction	The shortest distance（within half circle）	Use command signal (+) or (-) as moving direction, moving to the close command corresponding angle position（within one circle）	The same as linear axis behavior, move to command position（maybe over 1 circle）		Direct move to goal position（within 2 circle）
Increment instruction	Use command signal (+) or (-) as moving direction. Do increment movement.				
Reference position return	Move to middle point by increment or absolute type command, from middle point back to origin. (EX: Machine coordinate positioning)				
Machine coordinate positioning	The shortest distance（within half circle）			The same as linear axis behavior（maybe over 1 circle）	Direct move to goal position（within 2 circle）

Note1: Type C (Setting value is 3) is the specification for special purpose machine.

No	Descriptions	Range	Unit	Initial	Activate method
241 ~ 243	Axis dual feedback servo channel no.	[0 ~ 3]	-	0	restart

■ This parameter is used to define the actual axis number that is used to receive dual feedback signal from ruler. X-axis corresponds to 1, Y-axis corresponds to 2, and Z-axis corresponds to 3.

■ NOTE: With each servo axis that wants to set up a dual feedback, it needs two hardware ports on the servo card. In which, the first port is applied to send command from FBs-30GM and receive the encoder feedback of encoder. The second port is applied to receive the ruler's (optical encoder) feedback. Therefore, please check whether the hardware ports are enough to set up a dual feedback control system.

No	Descriptions	Range	Unit	initial	Activate method
261 ~ 263	Axis dual feedback resolution	[10 ~ 2500000]	Pulse/mm	250	reset

■ This parameter is used to set the resolution of ruler feedback of each servo axis. Note that this setting value is resolution for single phase (A or B phase)

■ Setting unit is pulse/mm for linear axis and is pulse/rev for rotation axis

■ Example:

1. Assume that the ruler resolution is 1um/pulse (1mm/1000pulse), with scaling factor of 4 (Pr30x=4). Thus, parameters Pr26x is set to (1000/4) =250.
2. Assume that the ruler resolution is 10um/pulse (1mm/100pulse), with scaling factor of 4 (Pr30x=4). Thus, parameters Pr26x is set to (1000/4) =25.
3. Assume that the rotary optical encoder resolution is 10mdeg/pulse (1rev/3600000pulse), with scaling factor of 4 (Pr30x=4). Thus, parameters Pr26x is set to (3600000/4) =90000.

No	Descriptions	Range	Unit	initial	Activate method
301 ~ 303	Axis dual feedback scaling factor	[1 ~ 4]	-	4	reset

■ This parameter is used to define the dual feedback encoder scaling factor and it can be set to 1, 2 or 4.

No	Descriptions	Range	Unit	initial	Activate method
401	Cutting acceleration time	[0 ~ 60000]	ms	300	reset

■ Set each axis under G01/G02/G03/G31 mode, this parameter is the spending time on compound feedrate accelerates to Pr405. In other words, this parameter and Pr405 will determine maximum compound acceleration.

$$A_{max} = \frac{Pr405/60}{Pr401/1000}\left(mm/sec^2\right)$$

No	Descriptions	Range	Unit	initial	Activate method
402	Acceleration accelerated to 1G time	[1 ~ 60000]	ms	150	reset

■ Set each axis under G01/G02/G03 mode, this parameter is the spending time on compound acceleration accelerates to 1G. In other words, this parameter will determine maximum compound jerk.

$$J_{max} = \frac{9.8}{Pr402/1000}\left(m/sec^3\right)$$

No	Descriptions	Range	Unit	initial	Activate method
404	Post cutting bell-shaped acceleration time	[0 ~ 60000]	ms	20	reset

■ The parameter can smooth the path of speed that plan before interpolation. The shake will be restrained. Suggest value is 20msec ~ 30msec.

■ EX:

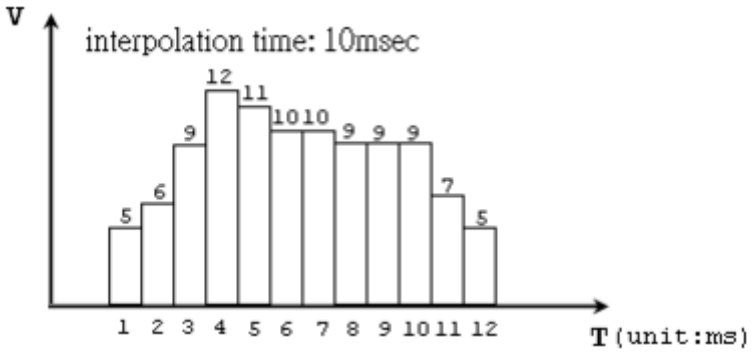


Figure 68: Speed-time before interpolation

The figure is speed-time before interpolation. If the post cutting bell-shaped acceleration time is 0, the option is disabled. If the parameter is existed, the command will be smoothed. EX: Pr404→ 5ms

Table 32: Interpolation time and command

Interpolation time (ms)	Command before interpolation (pulse)	Command after interpolation (pulse)
0	0	0
0	0	0
0	0	0
0	0	0
1	5	(0+0+0+0+5)/5=1
2	6	(0+0+0+5+6)/5=2.2
3	9	(0+0+5+6+9)/5=4
4	12	(0+5+6+9+12)/5=6.4
5	11	(5+6+9+12+11)/5=8.6
6	10	(6+9+12+11+10)/5=9.6
7	10	(9+12+11+10+10)/5=10.4
8	9	(12+11+10+10+9)/5=10.4
9	9	(11+10+10+9+9)/5=9.8
10	9	(10+10+9+9+9)/5=9.4
11	7	(10+9+9+9+7)/5=8.8
12	5	(9+9+9+7+5)/5=7.8
13	0	(9+9+7+5+0)/5=6
14	0	(9+7+5+0+0)/5=4.2
15	0	(7+5+0+0+0)/5=2.4
16	0	(5+0+0+0+0)/5=1

The command of speed is smoothed. The post cutting bell-shaped acceleration time can smooth the command and restrain the speed change.

No	Descriptions	Range	Unit	initial	Activate method
405	Maximum cutting feedrate	[6 ~ 3600000]	mm/min	5000	reset

■ Set the maximum cutting feedrate for compound speed.

No	Descriptions	Range	Unit	initial	Activate method
406	Maximum corner reference feedrate	[6 ~ 3600000]	mm/min	500	reset

■ Set the maximum corner feedrate. FBs-30GM will check the length of corner and decrease the speed before into the corner.
■ The parameter is the max speed at corner that the angle is 120 degree. Suggest value is 200mm/min.
■ The parameter is bigger and the speed is faster but the precise is worse. The parameter is smaller and the speed is slower but the precise is better.
■ Note:
If the program has G09 in position check, control will cancel decrease speed plan.
If you don't need corner decrease speed, Parameter 406 and 408 could set a huge value and the system will turn a corner with a high speed. Please Pr404 set bigger to protect tool and avoid the huge shake.

No	Descriptions	Range	Unit	initial	Activate method
408	Arc cutting reference feedrate at radius 5 mm	[0 ~ 3600000]	mm/ min	500	reset

■ Servo lag will make the arc path shrink during the arc cutting. The shrink error is:

$$E = \frac{T^2 V^2}{2R}$$

(T: servo system time constant. V: tangent velocity. R: radius)
■ We can calculate the speed with the radius by the function when shrink error and servo character is the same.

$$\frac{V}{V_{ref}} = \sqrt{\frac{R}{R_{ref}}}$$

(Circular velocity is direct proportion to square of circular radius)
■ Reference radius Rref=5mm. Using the Rref to set the circular velocity Vref. Normal tool suggest setting Vref=500mm/min.

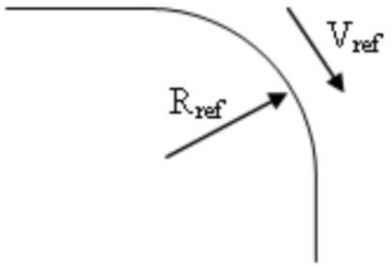


Figure 69: Reference radius and velocity

■ Note:

Huge curvature path and short block path both are clamped by Pr408. The same curvature path will clamp to the same velocity because of the Pr408. The following error will become small because of the velocity become small. The precise will become higher. If the following is still too big, please turn on the feed forward percentage (Pr581 ~ Pr583). It will send compensation for servo lag, but it makes bigger acceleration and shake. To solve the problem, cutting acceleration time (Pr401) can set longer.

If the high speed make centrifugal force is too bigger, the tool may shake. Before set Pr408, please check the machine rigidity to avoid shake.

No	Descriptions	Range	Unit	initial	Activate method
410	MPG acceleration time	[10 ~ 60000]	ms	200	reset

No	Descriptions	Range	Unit	initial	Activate method
411	Rapid Travel G00	[0 ~ 1]	-	0	reset

■ Rapid Travel G00:

0: Linear;

1: Independent.

No	Descriptions	Range	Unit	Initial	Activate method
413	Reserve local coordinate G92(G92.1) after reset	[0 ~ 2]	-	0	reset

■ Set reserve local coordinate G92(G92.1) after reset:

0: After reset, it will not reserve local coordinate;

1: After reset, it will reserve local coordinate, but restart is not;

2: After reset or restart, it will not reserve local coordinate.

No	Descriptions	Range	Unit	Initial	Activate method
414	Reserve Workpiece Coordinate System after reset	[0 ~ 2]	-	0	reset

■ Reserve Workpiece Coordinate System after reset:

0: After Reset reserve to default;

1: After Reset no reserve to default;

2: After Reset or Turn-OFF no reserve to default.

No	Descriptions	Range	Unit	initial	Activate method
421 ~ 423	Axis cutting in-position window	[0 ~ 300000]	BLU	30	reset

■ When program include G09, the system will check the position of block.
■ After system stop sending command below 2second, system will check motor feedback of position in the window. If it is in the range, systems send command for next block. If it spend time over 2sec, system alarm 『Exact Stop wait too long』

No	Descriptions	Range	Unit	initial	Activate method
441 ~ 443	Axis rapid travel (G00) acceleration time	[0 ~ 60000]	ms	200	reset

■ Set each axis under G00 mode, Pr441 ~ Pr443 are the spending time on each axis velocity accelerate to Pr461 ~ Pr463 respectively. In other words, Pr441 ~ Pr443 and Pr461 ~ Pr463 will determine maximum compound acceleration.

$$A_{max} = \frac{Pr461 \sim /_{60}}{Pr441 \sim /_{1000}} \left(mm/_{sec^2} \right)$$

No	Descriptions	Range	Unit	initial	Activate method
461 ~ 463	Axis max. rapid travel (G00) feedrate	[6 ~ 360000]	mm/ min	10000	reset

■ Set each axis under G00 mode, this parameter represent the max allowable feedrate when G00 override is not F0.

No	Descriptions	Range	Unit	initial	Activate method
481 ~ 483	Rapid travel in-position window	[0 ~ 300000]	BLU	30	reset

■ When program include G09, the system will check the position of block.
■ After system stop sending command below 2second, system will check motor feedback of position in the window. If it is in the range, system sends command for next block. If it spend time over 2sec, system alarm 『Exact Stop wait too long』

No	Descriptions	Range	Unit	initial	Activate method
501 ~ 503	Axis rapid travel (G00) F0 feedrate	[0 ~ 15000]	mm/ min	0	reset

■ Set each axis under G00 mode, this parameter represent the max allowable feedrate when G00 override is F0.

No	Descriptions	Range	Unit	initial	Activate method
521 ~ 523	Axis JOG feedrate	[6 ~ 360000]	mm/ min	6000	reset

■ Set each axis under JOG mode, this parameter represent each axis maximum feedrate.

■ On MPG mode, if Pr661~Pr663 are zero, then MPG movement maximum feedrate also dominated by Pr521~Pr523.

No	Descriptions	Range	Unit	initial	Activate method
541 ~ 543	Axis cutting acceleration time	[0 ~ 60000]	ms	50	reset

■ Set each axis under G01 mode, Pr541~Pr543 are the spending time on compound feedrate accelerate to Pr621~Pr623 respectively. In other words, Pr541~Pr543 and Pr621~Pr623 will determine each axis maximum jerk.

$$A_{max} = \frac{Pr621 \sim /_{60}}{Pr541 \sim /_{1000}} \left(mm/_{sec^2} \right)$$

No	Descriptions	Range	Unit	initial	Activate method
561 ~ 563	Axis loss pulse check window	[50 ~ 300000]	BLU	100	reset

■ After system stop sending command over 1second, system will check the difference between command and motor feedback. If it is over the range, system alarm 『Lost position』.

No	Descriptions	Range	Unit	initial	Activate method
581 ~ 583	Axis velocity feed forward percentage	[-10000 ~ 1000]	%	0	reset

■ FBs-30GM use the following formula to adjust command. Then this method will change Kp and improve servo lag phenomenon. When bigger Pr581~Pr583, servo lag amounts are smaller, but user need to notice that it will cause machine vibration.

$$K_p{}' = \frac{Pr181}{1 - Pr581/_{100}}$$

No	Descriptions	Range	Unit	initial	Activate method
601 ~ 603	Axis corner reference feedrate	[6 ~ 3600000]	mm/min	360000	reset

■ The parameters are set for corner feedrate. FBs-30GM will check the length of corner and decrease the speed before into the corner.
■ The parameters are the max speed at corner that the angle is 120 degree. Suggest value is 60mm/min.
■ The parameters are bigger and the speed is faster but the precise is worse. The parameter is smaller and the speed is slower but the precise is better.
■ Note:
If the program has G61 or G09 in position check, control will cancel decrease speed plan.
If you don't need corner decrease speed, Parameter 406 and 408 could set a huge value and the system will turn a corner with a high speed. Please Pr404 set bigger to protect tool and avoid the huge shake.
If the program has auxiliary axis or rotation axis, please set Pr601~Pr623 to avoid machine vibration. Suggest value is 500.

No	Descriptions	Range	Unit	initial	Activate method
621 ~ 623	Axis maximum cutting feedrate	[6 ~ 3600000]	mm/min	5000	reset

■ Set each axis under G01 mode, Pr621~Pr623 are the each axis maximum cutting feedrate.

No	Descriptions	Range	Unit	initial	Activate method
641 ~ 643	Axis cutting bell-shaped acceleration time	[1 ~ 60000]	ms	10	reset

■ Set each axis under G00/G01 mode, Pr621~Pr623 are the spending time on each axis acceleration accelerates to 1G. In other words, this parameter will determine each axis maximum jerk.

$$J_{max} = \frac{9.8}{Pr641 \sim /_{1000}} \left({}^{m}/_{sec^3} \right)$$

No	Descriptions	Range	Unit	initial	Activate method
661 ~ 663	Axis MPG feedrate	[0 ~ 3600000]	mm/min	6000	reset

- Pr661~Pr663: axis MPG feedrate upper bound.
- When parameter is set to 0, it means using JOG feedrate as MPG feedrate.

No	Descriptions	Range	Unit	initial	Activate method
821 ~ 823	Speed of first part homing	[0 ~ 240000]	mm/ min	10000	reset

- On Home search process, this parameter will determine the maximum moving velocity before touching Home DOG switch.

No	Descriptions	Range	Unit	initial	Activate method
841 ~ 843	Speed of second part homing	[0 ~ 240000]	mm/ min	2000	reset

- On Home search process, this parameter will determine the maximum moving velocity after leaving Home DOG switch.

No	Descriptions	Range	Unit	initial	Activate method
861 ~ 863	Negative homing direction	[0,1]		0	reset

- On Home search process, this parameter will determine the direction of Home DOG switch.

No	Descriptions	Range	Unit	initial	Activate method
881 ~ 883	Axis home offset	[-99999999 ~ 99999999]	BLU	0	reset

- The parameter have to fit Pr961~Pr980(Home search method)。
- Pr961~Pr963 is 0 or 1: When FBs-30GM find the motor index, tool will move to specialize point that is the offset position. After arriving the point, machine coordinate will be zero.
- Pr961~Pr963 is 2: When FBs-30GM find the motor index, tool will move to point that is the index. After arriving the point, machine coordinate will be offset value.
- Pr961~Pr963 is 3: When FBs-30GM leave DOG sensor, tool will move to specialize point that is the offset position. After arriving the point, machine coordinate will be zero.
- Home Offset Action

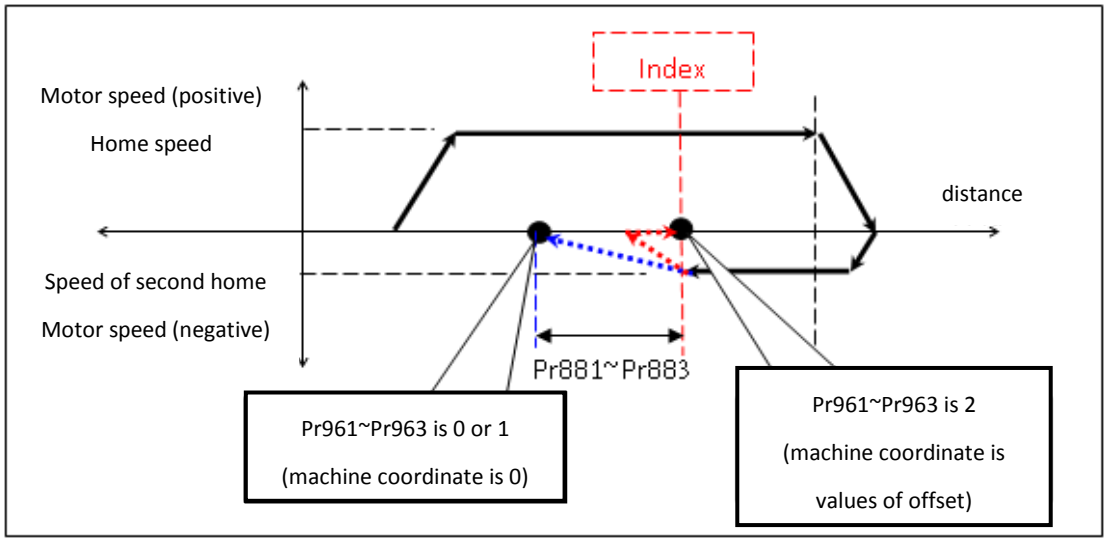


Figure 70: Home Offset Action

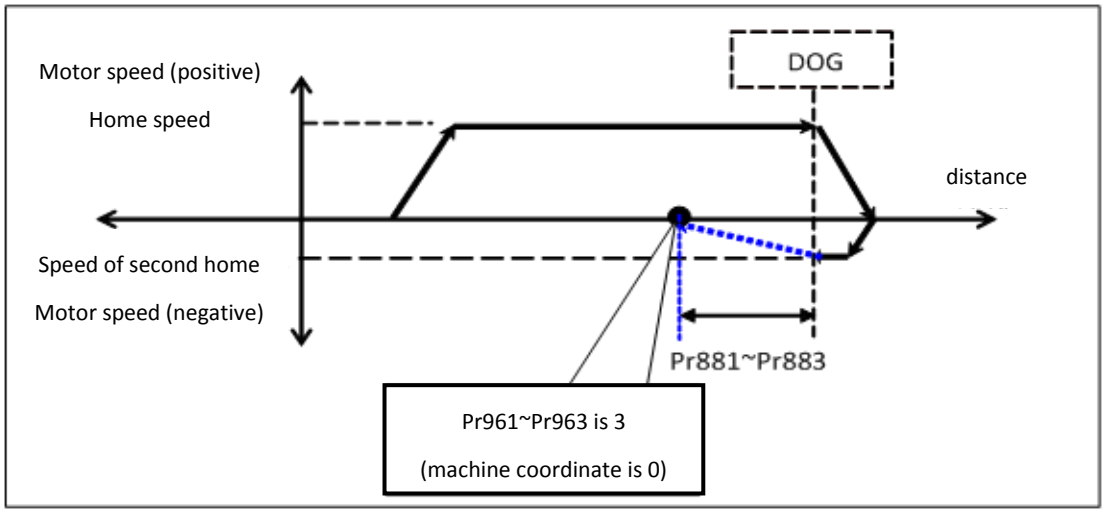


Figure 71: Home Offset Action (cont.)

No	Descriptions	Range	Unit	initial	Activate method
901 ~ 903	Axis zero speed check window(count)	[3 ~ 10000]	Pulse	3	reset

■ When FBs-30GM doing home search, touch the HomeDog, the second moving and Servo-On, motor will check the zero speed stop of state. The parameter is the value of range. If encoder feedback is in the range, FBs-30GM deems the motor is stop, or alarm and stop.

No	Descriptions	Range	Unit	initial	Activate method
921 ~ 940	Home dog polarity (0:positive;1:negative)	[0 ~ 1]	-	0	reset

■ Set HOME DOG polarity, the normal write is NORMAL CLOSE, but in the advance switch case is NORMAL OPEN.

No	Descriptions	Range	Unit	initial	Activate method
941 ~ 943	Enable axis home grid function	[0-1]	-	0	reset

■ Enable axis home grid function
0: disable
1: enable
■ Enable axis home grid function. If the grid value is smaller than 50% (motor half-revolve). FBs-30GM will ignore this index signal and find the next index to be original signal.
■ Home grid:
When motor leave home dog and move to the first index of motor, motor rotate the revolution. It show on the system variable 56~59. The unit is percent. 25 is mean 1/4 rev. 50 is mean 1/2 rev.
■ When HOME search method is 3, this function will disable.

No	Descriptions	Range	Unit	initial	Activate method
961 ~ 963	Home mode of each axis	[0 ~ 3]	-	0	reset

■ These parameters are used to decide the HOME search method of each axis:
0: By HomeDog sensor, suitable for linear axis or rotary axis witch the proportion of motor and pitch is not 1. After HOME, table moved on the machine position which offset had added;

1: By reference index of motor, suitable for linear axis or rotary axis witch the proportion of motor and pitch is 1;

2: By HomeDog sensor, suitable for linear axis or rotary axis witch the proportion of motor and pitch is not 1. After HOME, motor laid on index;

3: By HomeDog sensor, but no encoder index signal. Suitable for linear axis or screw and motor gear ratio is not integer for rotary axis. When axis direction finds DOG sensor for Home shift processing, direct move to machine coordinate position. After arriving position, clear machine coordinate position to 0, then it is called finish Home search action;

No	Descriptions	Range	Unit	initial	Activate method
981 ~ 983	Axis homing 2nd protect revolution(encoder type)	[1 ~ 999999]	Rev	5	reset

■ These parameters are used to determine the numbers of pitches when searching home, if motor can't leave Home Dog after moving over the number of pitches, FBs-30GM will send alarm message.

■ These parameters are effective when Pr201 ~ Pr203 are set to 0 and Pr961 ~ Pr963 are set to 0, 2 or 3.

No	Descriptions	Range	Unit	initial	Activate method
1001 ~ 1003	Axis fast home return function	[0 ~ 1]	-	0	restart

■ These parameters are used to determine whether to enable fast home return function of each axis and are off by default in order to be compatible with HOME mode. Enable the axis fast home return function (Pr100x = 1) and the specifications are as follows:

1. When the machine has not yet executed the first reference searching, the mechanical origin has not been established (M1471 ~ M1473 Off). If carrying out reference searching, FBs-30GM will follow Pr96x's setting to decide the reference searching method. During reference searching, the first and the second homing speed will be determined by Pr82x, Pr84x.

2. After the first reference searching, the mechanical origin has been established (M1471 ~ M1473 On). If FBs-30GM carries out reference searching again, the machine will not go back to the mechanical origin with the previous reference searching method, but do rapid positioning (G00) to the origin directly.

No	Descriptions	Range	Unit	initial	Activate method
1221 ~ 1223	Backlash compensation start	[0 ~ 2]	-	0	reset

■ Set Backlash compensation start or not.
0: OFF;
1: Linear Guideway ON;
2: Box Guideway ON.

No	Descriptions	Range	Unit	initial	Activate method
1241 ~ 1260	G00 backlash compensation value	[-999999 ~ 999999]	BLU	0	reset

■ The parameter is machine tool on the high speed (G00) and move to a point with negative and positive direction. The backlash is the error of stop.

No	Descriptions	Range	Unit	initial	Activate method
1261 ~ 1263	G01 backlash compensation value	[-999999 ~ 999999]	BLU	0	reset

■ The parameter is machine tool on the low speed (F10) and move to a point with negative and positive direction. The backlash is the error of stop.

No	Descriptions	Range	Unit	initial	Activate method
1281 ~ 1283	Backlash critical speed	[0 ~ 3000]	mm/min	800	reset

■ The backlash and the speed is a relation of exponent. The parameter set for backlash coverage speed. If the value is bigger, the coverage speed is faster.
■ When Pr1281 ~ Pr1283 are equal to zero, FBs-30GM will still follow default value 800 to process compensation amount estimation.

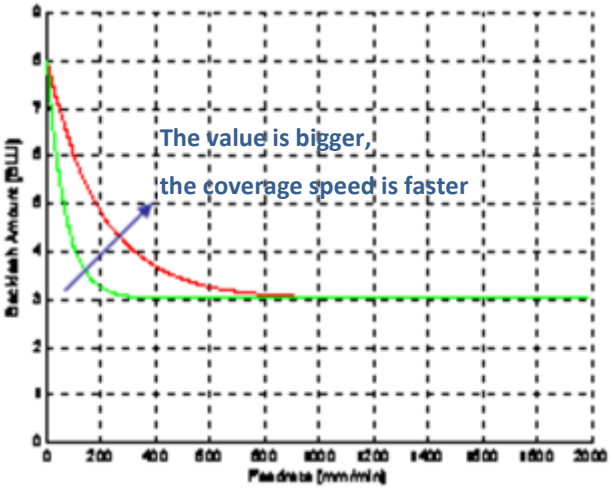


Figure 72: Backslash amount vs feedrate

No	Descriptions	Range	Unit	initial	Activate method
1301 ~ 1303	Pitch error compensation type	[0 ~ 2]	-	0	reset

■ Set the parameter to decide to start compensation or not
0: No compensation;
1: Unidirection;
2: Bidirection.

No	Descriptions	Range	Unit	initial	Activate method
1321 ~ 1323	Pitch error compensation Interval	[1000 ~ 99999999]	BLU	50000	reset

■ After interval compensation start, according to this setup, set the pitch of compensation.

No	Descriptions	Range	Unit	initial	Activate method
1341 ~ 1343	Table index for reference (home)	[1 ~ 100]	-	50	reset

■ After interval compensation start, what number is mechanical origin in table for compensation, suggest 50.

No	Descriptions	Range	Unit	initial	Activate method
1401 ~ 1403	Axis mechanical compensation time constant	[0 ~ 60000]	ms	0	reset

■ Mechanical compensation (backlash, pitch error) is described as an exponential curve. This parameter is used to determine the time constant (ms) of exponential curve. The lower the setting value is, the lesser time needed to complete the compensation. However, it may find the machine vibrates during operation if the time constant is too low. The suggested setting value is 100ms.

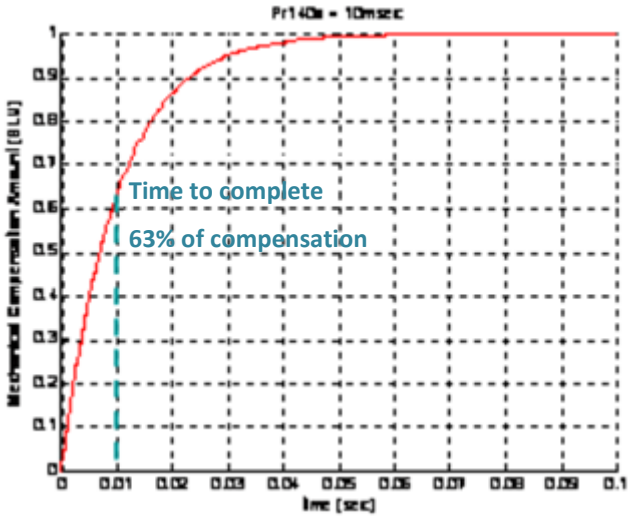


Figure 73: Mechanical compensation amount vs time

No	Descriptions	Range	Unit	initial	Activate method
1421 ~ 1423	Axis max. static dual error	[0 ~ 100000]	BLU	1000	reset

■ This parameter is used to define the maximum allowed error between motor encoder and ruler's (optical encoder) feedback signal in static state.

No	Descriptions	Range	Unit	initial	Activate method
2001	MPG 4th scaling factor	[10 ~ 1000]	LIU	100	reset

■ Set the MPG 4th of pulse to the LIU.
■ The min unit of LIU, the unit will be controlled by mode of metric or inch.

No	Descriptions	Range	Unit	initial	Activate method
2041	MPG resolution (Pulse/rev)	[100 ~ 2500000]	-	100	reset

No	Descriptions	Range	Unit	initial	Activate method
2051	MPG scaling factor	[1 ~ 4]	-	4	reset

No	Descriptions	Range	Unit	initial	Activate method
2401 ~ 2406	1st Software travel limit	[-999999999 ~ 999999999]	BLU	-999999999 999999999	reset

■ After homing, control use axis positive software limit.

No	Descriptions	Range	Unit	initial	Activate method
2441 ~ 2446	2nd Software travel limit	[-999999999 ~ 999999999]	BLU	-999999999 999999999	reset

■ The second software travel limit is turned on or off by M1423.

No	Descriptions	Range	Unit	initial	Activate method
2481	2nd software limit persistency	[0 ~ 2]	-	0	reset

■ This parameter is used to set the second software limit persistency:
0: Stop FBs-30GM to restore the limit to the settings in Pr2441 ~ 2446
1: Stop FBs-30GM to retain the limit set by MACRO variables #1941 ~ #1943 (2nd software positive limit), #1961 ~ #1963 (2nd software negative limit).
2: Stop or turn on/off FBs-30GM to retain the limit set by MACRO variables #1941 ~ #1943 (2nd software positive limit), #1961 ~ #1963 (2nd software negative limit).

No	Descriptions	Range	Unit	initial	Activate method
2801 ~ 2803	2nd reference point	[-999999999 ~ 999999999]	BLU	0	reset
2821 ~ 2823	3rd reference point	[-999999999 ~ 999999999]	BLU	0	reset
2841 ~ 2843	4th reference point	[-999999999 ~ 999999999]	BLU	0	reset

No	Descriptions	Range	Unit	Initial	Activate method
3202	I/O scan time	[100 ~ 5000]	0.001ms	5000	restart

■ After system start, the scan time of I/O card.

No	Descriptions	Range	Unit	Initial	Activate method
3203	Interpolation time interval	[500 ~	0.001ms	5000	restart

		2000000]			

■ After system start, when each axis direction movement, command time interval.

No	Descriptions	Range	Unit	Initial	Activate method
3207	Feedrate override selection	[1 ~ 2]	-	2	restart

■ Set the override type:
1: override is reality percentage,
range: -200% ~ +200 % (industrial mechanical setup);
2: override default steps,
range: 1 ~ 20.

No	Descriptions	Range	Unit	initial	Activate method
3221	Debug level	[0 ~ 2]	-	0	reset

■ When MACRO program execute, single step block execute or not.
0: disable;
1: enable;
(M1416 have to be ON before program start)

No	Descriptions	Range	Unit	initial	Activate method
3241	Decimal point type	[0 ~ 1]	-	0	restart

■ Set the parameter for decimal point type:
0: standard, 1=0.001mm;
1: pocket, 1= 1mm.

No	Descriptions	Range	Unit	initial	Activate method
3805	Static dual feedback error timeout	[0 ~ 60000]	ms	1000	reset

■ This parameter is used to define the waiting time before FBs-30GM switches to static state when it stops sending command.

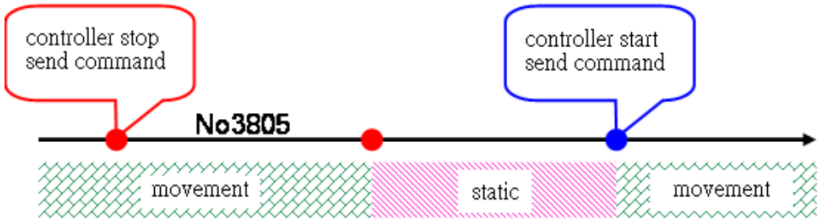


Figure 74: Static dual feedback error timeout

No	Descriptions	Range	Unit	initial	Activate method
3807	Destination not on arc check window	[0 ~ 1000]	BLU	5	reset

■ Set the error of radius from start-point to end-point. If the error is larger than this parameter, FBs-30GM alarms.

No	Descriptions	Range	Unit	initial	Activate method
3811	Start address of persist working global variable	[0 ~ 400]	-	0	restart

■ 0: @1 ~ @400 data all reset after power off;
1 ~ 400: Start address of persist working global variable.
EX: setting 100, @100 ~ @400 data will persist after power off.

No	Descriptions	Range	Unit	initial	Activate method
3817	Fatal dual feedback error	[0 ~ 100000]	BLU	10000	reset

■ This parameter is used to define the maximum allowed dual error between motor encoder and ruler's (optical encoder) feedback signal in dynamic state.
■ If setting value is 0, this checking function is inactive.

No.	Description	Range	Unit	Default	Activate method
3818	Dual feedback self-detect error (pulse)	[0 ~ 50]	Pulse	0	reset

■ After activating dual feedback, the A/B pulse number between two indexes are recorded and self-checking every time FBs-30GM encounters an index from ruler (optical encoder), if the difference exceeds the value set by this parameter, FBs-30GM shall pop-up MOT-40 "Dual feedback self-detect error exceed".
■ If the setting value is 0, the self-checking function shall be disabled
■ Generally, it is applied to all types of optical encoder including both equal distance Optical encoder and distance code Optical encoder.
■ Limitation
- ➢ This function is only enabled after the axis completes returning reference point (search HOME)
- ➢ When a problem occurs, the system shall not pop-up alarm immediately, but hold until the 5th index is received, then only the alarm pop-up. In other words, if the movement range is within 4 indexes, such detection function is inactive
- ➢ Default index's width set by the system is 5 Pulses

No	Descriptions	Range	Unit	initial	Activate method
3821	Coupling master axis number	[0 ~ 3]	-	0	restart
3822	Coupling slave axis number	[0 ~ 3]		0	restart

■ Pr3821 and Pr3822 are set to coupling axis number.
■ EX: When Pr3821 = 1 (it means X axis) and Pr3822 = 2 (it means Y axis), then Y axis movement will follow X axis, and the moving ratio according to Pr3823 and Pr3824.

No	Descriptions	Range	Unit	initial	Activate method
3823	Coupling master axis ratio factor	[1 ~ 999999]		0	restart
3824	Coupling slave axis ratio factor	[-999999999 ~ 999999999]		0	restart

■ Pr3823 and Pr3824 are set to the moving ratio for synchronous moving axis direction.
■ EX: When Pr3823 = 1 and Pr3824 = 2, it implies “if master axis moves 1mm, then slave axis moves 2mm”.

No	Descriptions	Range	Unit	initial	Activate method
3825	Coupling type	[0 ~ 5]		0	restart

■ Pr3825 set the enable timing of the two couple axes.

0: cancel couple

1: Machine coupling, coupling starts from power on and can’t cancel.

2: PeerSynchronization coupling:
■ Coupling starts from power on and M1422 on. When M1422 is off, coupling is canceled.
■ FBs-30GM adds command from master axis and slave axis and sends to two axes at the same time.

3: Superimposition coupling
■ Coupling starts from power on and M1422 on. When M1422 is off, coupling is canceled.

■ Superimposition coupling is slave axis superimpose on the master axis. When the command makes for master axis, both of the axis will move. When commands make for slave axis, the slave axis will move and relative to the position of the master axis.

4: MasterSlaveSynchronization coupling

■ Coupling starts from power on and M1422 on. When M1422 is off, coupling is canceled.

■ MasterSlaveSynchronization coupling is FBs-30GM will get the command from master axis then send two axes to execute.

5: One to many coupling

■ Coupling starts from power on and M1422 on. When M1422 is off, coupling is canceled.

■ Similar to PeerSynchronization coupling, FBs-30GM adds command from master axis and slave axis and sends to all axes to execute.

■ Bit on, the axis is coupling.
Bit 1: X axis to carry 2
Bit 2: Y axis to carry 4
Bit 3: Z axis to carry 8
When Pr3822 is 12(12=4+8), the slave axes are Y axis and Z axis.

■ Note: When use one to many coupling, master axis ratio and slave axis ratio become 1:1. Settings of Pr3823 and Pr3824 are not useful.

No	Descriptions	Range	Unit	initial	Activate method
3826	Coupling couple time(ms)	[0 ~ 60000]	ms	0	reset
3827	Coupling decouple time(ms)	[0 ~ 60000]	ms	0	reset

■ Pr3826: Coupling couple time

■ Pr3827: Coupling decouple time

No	Descriptions	Range	Unit	initial	Activate method
3837	Initial Command Mode (0:default;1:G90;2:G91)	[0 ~ 2]	-	0	restart

■ Default is G90.

No	Descriptions	Range	Unit	initial	Activate method
8001 ~ 8600	Pitch error compensate , compensation table	[-999999 ~ 999999]	BLU	0	reset

■ The parameter set for the compensation of the pitch error. The value is modulus.

Compensation＝Command – reality

Pr8001 ~ 8100 are X axis positive direction pitch error compensation table 1 ~ 100.

Pr8101 ~ 8200 are X axis negative direction pitch error compensation table 1 ~ 100.

Pr8201 ~ 8300 are Y axis positive direction pitch error compensation table 1 ~ 100.

Pr8301 ~ 8400 are Y axis negative direction pitch error compensation table 1 ~ 100.

Pr8401 ~ 8500 are Z axis positive direction pitch error compensation table 1 ~ 100.

Pr8501 ~ 8600 are Z axis negative direction pitch error compensation table 1 ~ 100.

■ Ex:

Command value is 20000 BLU, machine value is 20002 BLU then the compensation value is -2

Command value is 40000 BLU, machine value is 39999 BLU then the compensation value is 1

Command value is -20000 BLU, machine value is -20002 BLU then the compensation value is 2

Command value is -40000 BLU, machine value is -39999 BLU then the compensation value is -1

Instruction of pitch error compensation

Manufacturing error of screw leads to the inconsistence between command and actual motion of working table. However, because this error is a constant value, it can be measured by the equipment and setting parameters into FBs-30GM to compensate this error in the machining process.

Pr1301 ~ 1303 determine whether Pitch error compensation function is enabled.

Pr1321 ~ 1323 determine the value of basic pitch error compensation.

Pr1341 ~ 1343 determines the starting compensation no. of original point in pitch compensation table. For every axis FBs-30GM provides totally 100 compensation points, the default and recommended value is 50.

Steps for measurement of pitch compensation parameter

Step 1: Close all mechanical compensation (pitch—Pr130x; backlash—Pr122x, Pr124x, Pr126x, Pr128x; sharp corner—Pr136x, Pr144x), and do the home search action

Step 2: Load the attachment example program, and then with the measuring instruments measures the pitch error of every single pitch.

Step 3: According to pitch compensation type (one-way / two-way), and stroke direction of axis (home direction positive / negative), select the corresponding fill in format.

One-way pitch compensation (just fill in positive table)

Regardless of moving direction of axes, FBs-30GM will send all positive direction values in the reference table as the compensation values at the same point of the stroke.

Axial stroke is in the positive direction of home:

Moves the machine away from home and progress to the positive direction of machine coordinate, measures the pitch error and enters the error into Pr800x “Positive absolute compensation pitch error table”. Note that the fill in serial no. of pitch error compensation is to the higher direction.

Move the machine away from home and progress in the positive direction of machine coordinate and Pr134x=50 fill in Pos.table 50, 51….59, 60																			
40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60
										0	→									

Axial stroke is in the negative direction of home:
Moves the machine away from home and progress to the negative direction of machine coordinate, measures the pitch error and enters the error into Pr800x "Negative absolute compensation pitch error table". Note that the fill in serial no. of pitch error compensation is to the lower direction.

Move the machine away from home and progress in the negative direction of machine coordinate and Pr134x=50 fill in Pos.table 50,49….41, 40																				
40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60
←										0										

Two-way pitch compensation (fill in positive & negative table)
According to the moving direction of machine, FBs-30GM will determine to use positive or negative table value at the same point of stroke.
Axial stroke is in the positive direction of home: Moves the machine away from home and progress to the positive direction of machine coordinate, measures the pitch error and enters the error into Pr800x "Pos. abs. comp. pitch err. table". Revert the machine progress direction and move back to home, measures the pitch error and enters the error into Pr810x "Neg. abs. comp. pitch err. table".

Move the machine away from home and progress in the positive direction of machine coordinate and Pr134x=50 fill in **Pos. table** 50, 51…, 59, 60																				
40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60
										0	→									
When the machine progress direction is revert and move back to home fill in **Neg. table** 60, 59…, 51, 50																				
40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60
										0	←									

Axial stroke is in the negative direction of home:
Moves the machine away from home and progress to the negative direction of machine coordinate, measures the pitch error and enters the error into Pr810x "Neg. abs. comp. pitch err. table". Revert the machine progress direction and move back to home, measures the pitch error and enters the error into Pr800x "Pos. abs. comp. pitch err. table".

Move the machine away from home and progress to the negative direction of machine coordinate, and Pr134x=50 fill in **Neg. table** 50, 49…, 41, 40

40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60
←										0										

When the machine progress direction is revert and move back to home fill in **Pos. table** 40, 41…, 49, 50

40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60
→										0										

At last do the experiment again to measure pitch compensation parameter and to verify the effectiveness of compensation.

4. Q & A

Q1: Pitch error compensation function is ineffectiveness

Ans: Pitch error compensation function is only enabled when the home search action is finished.

Q2: Machine is still at inaccurate position after being pitch error compensation.

Ans: The effectiveness of optimize mechanism compensation is depends on the reproducible of mechanism action. Thus, when this phenomenon occurs, please check whether the assembly of mechanism is appropriate.

Appendix III (Alarm ID.)

Operation alarm:

Alarm ID	OP-023	Alarm title	Power break in machining, re-calibrate before machining
Description	As start machining, FBs-30GM will set up machining flag in registry.dat and it will be removed when machining comebacks to ready status. When rebooting, if machining flag is not removed, this alarm will appear.		
Possible cause	Discontinue power in machining process.		
Solution	1. Check whether machining data setting is correct. 2. Reboot.		

Motor alarm:

Alarm ID	MOT-005	Alarm Title	DDA command overflow
Description	FBs-30GM sends too many commands. In the one interpolation time interval, if software calculates that the number of commands to be sent is out of 2047 pulses, this alarm will appear		
Possible Cause	1. DDA software time setting value (interpolation time interval, parameter Pr3203) is too long 2. Motion velocity is too fast 3. Servo resolution is set too high 4. Backlash compensation or pitch compensation is too large 5. Compensation is enabled before booting		
Solution	1. Recommend that low interpolation time interval setting (parameter 3203) is not less than 2000 2. Reduce the velocity to do the test if max rapid travel feedrate is to high (Pr461-Pr463) 3. Reduce the servo resolution setting to do test (encoder and FBs-30GM Pr61-Pr63) 4. If mechanical compensation time constant is set (parameter 1401 ~ 1420), cancel the mechanical compensation setting to do test		

	and find the best setting. 5. If system had set feed forward (parameter 581 ~ 600), cancel feed forward setting to do test and find the best setting. 6. Please contact staff of machinery manufacturer to solve problem
More description	In order to achieve the multi-axis coordinated control, FBs-30GM uses DDA (Digital Differential Analyzer), Cycle Time of DDA is set by parameter Pr3203. In one Cycle time of DDA, every axial is allowed to send maximum 2047 pulses. Once exceeding this value, FBs-30GM will send alarm

Alarm ID	**MOT-008**	**Alarm Title**	**Loss Pulse**
Description	One second after sending command, FBs-30GM will check whether the error of feedback command and sending command is in predetermined error range. If no, FBs-30GM will send alarm.		
Possible Cause	1. Kinematic occurs obstruction phenomenon 2. Servo drive occurs unexpected Servo ON / OFF 3. CPU board send the data to axis card unsuccessfully (CPU board or axis card has problem, the contact between CPU and axis card is not good) 4. The cable that sends command from FBs-30GM to servo driver has poor quality or is disconnected. 5. FBs-30GM doesn't set servo drive alarm check, FBs-30GM continues to send motion command although the drive is abnormal 6. Local interference		

Solution	1. Do not shut down FBs-30GM when alarm occurs. Please check whether the value of No 8, 9, 10 in debug function page is zero 2. Check whether the mechanical lubrication system is good. 3. Open the cover of axial to check whether foreign matter blocks the motion of axial. 4. Rotate screw to check whether machine is stuck (loading of driver) 5. Check the drive servo-on and the servo-off of power or cable signal 6. If the setting value of No 8, 9, 10 in debug function page do not change, please take home search action (don't need to reboot), after that check whether parameters 24, 25, 26, 40, 41, 42 are equal to zero, if the parameters 24, 25, 26 are not equal to zero, the feedback loop has problems 7. If the parameters 40, 41, 42 are not equal to zero, command transmission from FBs-30GM to the motor has been lost pulse. 8. If all parameters 24, 25, 40, 41, 42 are not zero, then the interference signal is relatively large, specifically in the machining process, the setting value of parameters 8, 9, 10 gradually become large. The reason is the contact point between CPU board and axis card is not good. Try to replace CPU board and axis card
More description	Set parameters 561 ~ 580 to check the range of loss pulse 8[X axis following error value] 9[Y axis following error value] 10[Z axis following error value] 24[X axis absolute position feedback value] 25[Y axis absolute position feedback value] 26[Z axis absolute position feedback value] 40[X axis absolute position command value] 41[Y axis absolute position command value] 42[Z axis absolute position command value]

Alarm ID	MOT-009	Alarm Title	Servo Driver Alarm
Description	Drive sends out warning signal		
Possible Cause	Drive alarm mostly is because of external causes. Ex: High temperature, connecting wire error, internal parameters is set wrong, servo motor is unsuitable, driver is error, etc.		
Solution	Follow the steps in driver's application manual to solve alarm		

Alarm ID	MOT-017	Alarm Title	First Positive software limit exceed
Description	The end point in movement of servo motor exceeds positive software limit		
Possible Cause	Stroke movement of machine table exceeds the setting value		
Solution	Remove alarm, and let axis moves to negative movement out of the stroke protection software		

Alarm ID	MOT-018	Alarm Title	First Negative software limit exceed
Description	The end point in movement of servo motor exceeds negative software limit		
Possible Cause	Stroke movement of machine table exceeds the setting value		
Solution	Remove alarm, and let axis move to positive movement out of the stroke protection software		

Alarm ID	MOT-019	Alarm Title	Following error exceed
Description	Because of the characteristics of servo, servo motor location, there is no way to respond the command of FBs-30GM immediately, so a slow phenomenon appears, when this latency is not in allowed range, FBs-30GM will send out the alarm.		
Possible Cause	1. Movement mechanism is not smooth 2. Contact wire has poor quality 3. Setting values of acceleration and deceleration time are too small 4. Servo on off Relay is interfered		

	5. Inner loop gain of driver is set too small 6. Encoder solution and electric gear ratio is set wrong 7. Drive or motor is damaged 8. Encoder or line between encoder and FBs-30GM is abnormal 9. On debug function page, variable number 23 is not equal to100
Solution	1. Add lubricating oil to machine 2. Use electric meter to check whether wire connecting is correct. 3. When FBs-30GM runs dry run mode, open case to check whether servo on off of relay pulses abnormally. 4. Increase acceleration and deceleration time (parameter 401) 5. Inner loop gain of driver is set too small. For Mitsubishi driver, check Pr37 6. Contact to machinery manufacturers for helping
More description	Maximum velocity setting value of G00 and home search is equal to setting parameter divided by Kp. This value multiplied by 2 is setting range of FBs-30GM. Reasonable following error: Ferr = speech in command/ setting value of loop gain Alarm allowed values= {max[(velocity of first stage in home search process), velocity G00 of each axis]/Kp}*2 For example: Speed 1000mm/min, loop gain 30, precision, 1um, Ferr = 1000*1000÷60÷30=555 32[X axis reasonable following error] 33[Y axis reasonable following error] 34[Z axis reasonable following error]

Alarm ID	**MOT-020**	**Alarm Title**	**Cannot back control mode when move**
Description	When emergency stop or monitor mode (C31 ~) is canceled, in one Interpolation time interval (No 3203) if the motor movement exceeds zero speed check window (901), FBs-30GM will send alarm.		
Possible	1. Cancel instantly movement of machine by hand		

Cause	2. Drive gain is set badly. Therefore, when cancelling instantly, motor will be trembled
Solution	1. Avoid man-made movement 2. Check the drive's position loop gain and speed loop gain setting

Alarm ID	**MOT-021**	**Alarm Title**	**Must re-homing**
Description	When MOT-0020 and MOT-0022 appear, FBs-30GM will send alarm		
Possible Cause	MOT -0020[Cannot back control mode when move] or MOT -0022[Home position inaccurate] is triggered		
Solution	See MOT -0020 or MOT -0022-alarm		

Alarm ID	**MOT-022**	**Alarm Title**	**Home position inaccurate**
Description	After booting, at the N(N>1) times of searching home, home grid will be compared to the result of the first time searching home, if the error is over 0.1 turn of motor, FBs-30GM will send alarm.		
Possible Cause	6. Homing signal of motor is abnormal 7. Stopper, coupling or bearings is not locked tightly		
Solution	1. Move motor in the same direction and observe to check whether position counter index changes normally. 2. Check whether the mechanism components are fixed properly		

Alarm ID	**MOT-023**	**Alarm Title**	**Fatal following error exceed**
Description	Because of the characteristics of servo, servo motor location, and FBs-30GM cannot respond immediately command, a delay phenomenon will appear, when this delay phenomenon is not in allowed limit, FBs-30GM will send alarm.		
Possible Cause	1. Servo motor doesn’t receive control due to external force 2. Parameter of drive - inner loop gain is too small 3. Parameters of acceleration and deceleration time is set too short 4. Encoder is abnormal or connecting encoder to FBs-30GM is abnormal		

Solution	1. Check the external motion of machine table 2. Check the setting parameter of drive 3. Check the acceleration and deceleration setting of each axis, parameters 401, 541-560 4. Maintain the connection between encoder and servo drives.
More description	Maximum velocity value of G00 and home search is equal to setting parameter divided by Kp. This value multiplied by 4 is setting range of FBs-30GM. Reasonable following error: Ferr = speech in command/ loop gain Alarm allowed values= {max[(velocity of first stage in home search process), velocity G00 of each axis]/Kp}*4 32[X axis reasonable following error] 33[Y axis reasonable following error] 34[Z axis reasonable following error]

Alarm ID	**MOT-024**	**Alarm Title**	**Fatal dual feedback error exceed**
Description	If FBs-30GM discovers that the command and the second command of encoder feedback exceed allowable limit set in Pr3817, FBs-30GM will send this alarm.		
Possible Cause	1. Servo motor doesn't receive control due to movement caused by external force 2. External encoder signal is unusual 3. External encoder parameters are set wrong		
Solution	1. Check external motion mechanism 2. Check whether external encoder wire is normal 3. Check whether external encoder corresponding to mechanical axis (Pr241 ~ 260), resolution (parameter 261 ~ 280) and feedback scaling factor (301 ~ 320) are set correctly. 4. Contact machinery manufactures in case no solution is found.		

Alarm ID	MOT-025	Alarm Title	Positive hardware limit exceed
Description	Servo motor touches the positive hardware limit in moving process		
Possible Cause	1. Machine table exceeds protection point 2. Hardware stroke switches are damaged or broken 3. Input signal has error		
Solution	1. Use MPG mode to move machine table to opposite direction once discovering that machine table stops on the switch 2. If machine table is not on the switch, check IO terminal blocks, 24V power supply terminal blocks, connecting wire and components of switch. 3. Check whether IO card is abnormal		

Alarm ID	MOT-026	Alarm Title	Negative hardware limit exceed
Description	Servo motor touches the negative hardware stroke limit in moving process		
Possible Cause	1. Machine table exceeds protection point 2. Hardware stroke switches are damaged or broken 3. Input signal has errors		
Solution	1. Use MPG mode to move machine table in opposite direction once discovering that machine table stops on the switch 2. If machine table is not on the switch, check IO terminal blocks, 24V power supply terminal blocks, connecting wire and components of switch. 3. Check whether IO card is abnormal		

Alarm ID	MOT-029	Alarm Title	Miss index in homing
Description	When searching home, if motor does not find out motor index signal after leaving home DOG more than 5 pitches, FBs-30GM will send this alarm.		
Possible Cause	1. Can't read the index signal. 2. The setting of homing 2nd travel feedrate is too fast. 3. The setting of motor reduction ratio is too big		

	4. The distance between index signal and HomeDog is more than 5 pitches
Solution	1. Check motor index wire connecting; observe debug variables 48(X), 49(Y), 50(Z) to check whether index signal is read. If no, please check whether connecting wire is correct. 2. Reduce setting value of the homing 2nd travel feedrate (Parameter 841 ~ 843)
More description	When searching home, machine will use the velocity setting value of the first stage to move to home DOG, and stop. After that machine moves backward with velocity of the second stage. After leaving home DOG to move backward, it start to search the nearest motor index signal. In the second stage, FBs-30GM will calculate according to resolution of encoder. If FBs-30GM leaves home DOG more than 5 pitches and cannot find out the index signal. FBs-30GM will send alarm.

Alarm ID	**MOT-030**	**Alarm Title**	**Zero speed timeout in homing**
Description	When motor touches HomeDog, if motor cannot stop, FBs-30GM will send this alarm.		
Possible Cause	1. Setting drive gain is not good, so it makes motor vibrating 2. Motor running causes resonance phenomenon.		
Solution	1. Check the position loop gain and velocity loop gain setting of driver 2. Start the resonance frequency inhibition ability of driver 3. Contact machinery manufacturers for help.		
More description	When searching home, machine will use the velocity setting value of the first stage to move to home DOG, and stop once it meets home DOG. After that machine moves backward with velocity of the second stage. After leaving home DOG to move backward, it start to search the nearest motor index signal. At the first stage to find the home DOG, motor will decrease velocity to stop. After 0.1 second command stops, if system data 8(X), 9(Y), 10(Z)-error register receives values bigger than zero speed check window(Pr901 ~ Pr920), FBs-30GM will send alarm.		

Alarm ID	MOT-036	Alarm Title	Can't leave home dog
Description	When searching home, if motor can’t leave HomeDog after moving over 5 pitches, FBs-30GM will send this alarm message.		
Possible Cause	HomeDog is damaged		
Solution	Use the electrical multimeter to check whether the sensor of HomeDog is damaged or wiring connection is missing.		
More description	When searching home, machine will use the velocity setting value of the first stage to move to home DOG, and stop. After that machine moves backward with velocity of the second stage. After leaving home DOG to move backward, it start to search the nearest motor index signal. In the second stage, FBs-30GM will calculate according to resolution of encoder. If FBs-30GM leaves home DOG more than 5 pitches and cannot find out the index signal, FBs-30GM will send alarm.		

Alarm ID	MOT–041	Alarm Title	Second Positive software limit exceed
Description	Position value of end point of servo motor exceeds setting value in FBs-30GM- Second Positive software limit		
Possible Cause	The motion of machine table exceeds setting value		
Solution	Remove alarm. Move axis in negative direction out of stroke protection software.		

Alarm ID	MOT–042	Alarm Title	Second Negative software limit exceed
Description	Position value of end point of servo motor exceeds setting value in FBs-30GM- Second negative software limit		
Possible Cause	The motion of machine table exceeds setting value		
Solution	Remove alarm. Move axis in positive direction out of stroke protection software.		

Alarm ID	MOT–051	Alarm Title	Inhibit cycle start in moving
Description	Before all manual commands are sent, prohibit starting machining to prevent operation error.		
Possible Cause	Manual command (JOG, INJOG, and MPGJOG) cannot be sent successfully.		
Solution	Remove alarm. Wait until machine stops, then start machining		

Compiler alarm:

Alarm ID	COM–001	Alarm Title	EOF in comment
Description	The symbol "(*" and "*)" must be used in pairs, if the program uses "(*" as the beginning of the comment, but doesn't use "*)" at the end of the comment. System will send alarm		
Possible Cause	Programming error		
Solution	Using symbol "(*" before command and symbol "*)" after command		

Alarm ID	COM–003	Alarm Title	Syntax error
Description	MACRO program has syntax error when FBs-30GM interprets it		
Possible Cause	Programming error		
Solution	Check program syntax according to symbol appears on the screen		

Alarm ID	COM–004	Alarm Title	Illegal variable
Description	System cannot access variable, this alarm will appear.		
Possible Cause	Change error variable		
Solution	Check program variable and confirm whether system uses that variable		

Alarm ID	COM–005	Alarm Title	expression too complex
Description	MACRO is too complicated,		
Possible Cause	Programming error		

Solution	Check whether logic is clear and correct

Alarm ID	COM–006	Alarm Title	EXIT statement outside loop statement
Description	The purpose of EXIT command is to jump out loop. If EXIT command cannot go to next loop, system will send alarm		
Possible Cause	Programming error		
Solution	Check whether EXIT command in program is used correctly		

Alarm ID	COM–007	Alarm Title	Repeat loop too deep
Description	IF Loop command in MACRO such as REPEAT loop, REPEAT loop, WHILE loop, FOR loop repeats more than 10 times, system will send this alarm.		
Possible Cause	Programming error		
Solution	Change MACRO program to avoid too many loop commands.		

Alarm ID	COM–008	Alarm Title	absent end of statement character ';'
Description	Program doesn’t have terminal symbol when MACRO command finishes.		
Possible Cause	Programming error		
Solution	Check MACRO program to confirm whether it has the terminal symbol		

Alarm ID	COM–009	Alarm Title	wrong assignment character ':='
Description	In program, if Assigning value to symbolic variable does not use the correct notation“: ＝”, system will send alarm		
Possible Cause	Programming error		
Solution	Check MACRO program to see whether assigning value to symbolic variable is correct		

Alarm ID	COM–010	Alarm Title	absent right ')'
Description	In program, notation “(” and “)” must be used in pairs, if “(” lacks “)”, system will send alarm		
Possible Cause	Programming error		
Solution	Check MACRO program to confirm whether using “(” and “)” is correct		

Alarm ID	COM–011	Alarm Title	absent right ']'
Description	In program, notation “[” and “]” must be used in pairs, if “[” lacks “]”, system will send alarm		
Possible Cause	Programming error		
Solution	Check MACRO program to confirm whether using “[” and “]” is correct		

Alarm ID	COM–012	Alarm Title	absent 'FOR' keyword in FOR statement
Description	If FOR loop in MACRO uses TO to define loop condition incorrectly, this alarm will appear.		
Possible Cause	Programming error		
Solution	Check MACRO program to confirm whether FOR loop uses TO correctly		

Alarm ID	COM–013	Alarm Title	absent 'DO' keyword in FOR statement
Description	If FOR loop in MACRO uses DO to define Implement task in loop incorrectly, this alarm will appear.		
Possible Cause	Programming error		
Solution	Check MACRO program to confirm whether FOR loop uses DO correctly		

Alarm ID	COM–014	Alarm Title	absent 'END_FOR' keyword in FOR statement
Description	If FOR loop in MACRO doesn’t use END_FOR to finish loop, this alarm will appear.		
Possible	Programming error		

Cause	
Solution	Check MACRO program to confirm whether FOR loop uses END_FOR

Alarm ID	**COM–015**	**Alarm Title**	**absent 'UNTIL' keyword in REPEAT statement**
Description	If REPEAT loop in MACRO uses UNTIL to define loop condition incorrectly, this alarm will appear.		
Possible Cause	Programming error		
Solution	Check MACRO program to confirm whether using UNTIL in REPEAT loop is correct		

Alarm ID	**COM–016**	**Alarm Title**	**absent 'END_REPEAT' keyword in REPEAT statement**
Description	If REPEAT loop doesn’t have END_REPEAT to finish loop, this alarm will be sent		
Possible Cause	Programming error		
Solution	Check MACRO program to confirm whether REPEAT loop has END_REPEAT		

Alarm ID	**COM–017**	**Alarm Title**	**absent 'DO' keyword in WHILE statement**
Description	If WHILE loop uses DO to define implement task incorrectly, this alarm will appear		
Possible Cause	Programming error		
Solution	Check MACRO program to confirm whether WHILE loop uses DO correctly		

Alarm ID	**COM–018**	**Alarm Title**	**absent 'END_WHILE' keyword in WHILE statement**
Description	If WHILE loop doesn’t have END_WHILE to finish loop		
Possible Cause	Programming error		

Solution	check MACRO program to confirm whether WHILE loop has END_WHILE to end

Alarm ID	**COM–019**	**Alarm Title**	**absent 'THEN' keyword in IF statement**
Description	If IF uses THEN to define implement task incorrectly, system will send this alarm		
Possible Cause	Programming error		
Solution	Check MACRO program to confirm whether IF loop use END correctly		

Alarm ID	**COM–020**	**Alarm Title**	**absent 'END_IF' or 'ELSE' keyword in IF statement**
Description	If IF loop doesn’t have ELSE or END_IF, this alarm will appear		
Possible Cause	Programming error		
Solution	check whether IF loop uses ELSE or END_IF		

Alarm ID	**COM–021**	**Alarm Title**	**absent 'END_IF' keyword in IF statement**
Description	If IF loop uses END_IF to finish loop incorrectly, this alarm will appear		
Possible Cause	Programming error		
Solution	Check whether IF loop uses END_IF correctly		

Alarm ID	**COM–022**	**Alarm Title**	**absent 'OF' keyword in CASE statement**
Description	If CASE command uses OF incorrectly, this alarm will appear		
Possible Cause	Programming error		
Solution	Check whether CASE command uses OF correctly		

Alarm ID	**COM–023**	**Alarm Title**	**absent 'END_CASE' or 'ELSE' keyword in CASE statement**
Description	If CASE command doesn’t use ELSE or END_CASE		
Possible	Programming error		

Cause	
Solution	Check whether CASE loop uses ELSE or END_CASE correctly

Alarm ID	COM–024	Alarm Title	absent 'END_CASE' keyword in CASE statement
Description	If CASE command doesn’t have END_CASE keyword		
Possible Cause	Programming error		
Solution	Ensure that END_CASE keyword is used before finishing CASE command		

Alarm ID	COM–025	Alarm Title	absent ':' or ',' delimiter in CASE statement
Description	If CASE command in MACRO uses ‘;’or ‘, ’, this alarm will appear.		
Possible Cause	Programming error		
Solution	Check MACRO program. In CASE statement, ‘;’or ‘, ’ is correct. However, you should use ‘;’ when finishing CASE command.		

Coordinate alarm:

Alarm ID	COR–001	Alarm title	Array Index must be Integer
Description	When indirect variable is not an integer, the system will send this alarm Ex: if #1 in @[#1+1] command is not positive integral, this alarm will appear		
Reason	Programming error.		
Solution	Please check the machining program, the index in MACRO command has to be rounded Ex: @[ROUND(#1)+1]		

Alarm ID	COR–002	Alarm title	File not found
Description	If the file that the system wants to read does not exist EX: Use M98 (or G65.G66…etc.) to call a no existence file.		
Reason	Programming error.		
Solution	Check the machining program to make sure the existence of the file.		

Alarm ID	COR–003	Alarm title	Divide by zero
Description	If denominator in division of MACRO is equal to 0 Ex: If #3 in #1 :=(#2 / #3) command is equal to 0.		
Reason	Programming error		
Solution	Check the machining program to ensure that the denominator is not equal to 0.		

Alarm ID	COR–004	Alarm title	Operand domain error
Description			
Reason	Programming error		
Solution	Please check the machining program.		

Alarm ID	COR–005	Alarm title	Program loading failure
Description	MACRO syntax error.		
Reason	Programming error		
Solution	Please check the machining program.		

Alarm ID	COR–006	Alarm title	Arc not on work plane
Description	In G02 and G03 syntax, if vector from center to starting point is not on the arc of working plane, this alarm will appear. Ex: G17 G02 I50. K10.; if it implements the left program, this alarm will appear.		
Reason	Programming error		
Solution	Check the machining program to ensure that G02 and G03 are used correctly.		

Alarm ID	COR–007	Alarm title	Arc radius too short
Description	In G02 and G03 syntax, if Arc radius is smaller than 10 to the power of minus 10 (10^-10), system will send this alarm		
Reason	Programming error		
Solution	Check the machining program to ensure that the Arc radius of G02 and G03 are used correctly		

Alarm ID	COR–008	Alarm title	Arc destination not on arc
Description	In G02 and G03 syntax, if the Arc end point coordinate is not on the circle, system will send this alarm. From V8.31 version, parameter 3807- destination not on arc check window is added. It allows error set in parameter 3807. When error of Arc end point coordinate is smaller than setting value in Pr3807, system will automatically correct center coordinate, so the end point can be on arc correctly. If error of Arc end point coordinate is bigger than setting value in Pr3807, system will send alarm.		
Reason	Programming error		
Solution	Check the machining program to ensure that the Arc radius of G02 and G03 are used correctly		

Alarm ID	COR–009	Alarm title	Macro call too deep
Description	Use G65 to call MACRO subprogram that has more than 12 layers		
Reason	Programming error		
Solution	Check machining program to ensure that G65 calls MACRO subprogram that has less than 12 layers		

Alarm ID	COR–010	Alarm title	Modal macro call too deep
Description	Use G66 to call MACRO subprogram that has more than 4 layers		
Reason	Programming error		
Solution	Check machining program to ensure that G66 calls MACRO subprogram that has less than 4 layers		
Alarm ID	COR–011	Alarm title	Subprogram call too deep
Description	Use M98 to call subprogram that has more than 16 layers		
Reason	Programming error		
Solution	Check machining program to ensure that M98 calls subprogram that has less than 16 layers		

Alarm ID	COR–012	Alarm title	Too many modal macro canel,G67
Description	G66 and G67 need to be used in pairs. When number of G67 is larger than G66 in one machining program, this alarm will appear.		
Reason	Programming error		
Solution	Check program to ensure that G66 and G67 are used in pairs		

Alarm ID	COR–013	Alarm title	G65,G66 must be the last one in G code list
Description	G65 and G66 are MACRO, so in single block the right hand side of G65 and G66 will have processing arguments. So in single block, please put other G code in the left hand side of G65 and G66. If the right hand side of G65 and G66 has G code or M code, system will send this alarm		
Reason	Programming error		
Solution	Please check the machining program.		

Alarm ID	COR–014	Alarm title	Absent program number
Description	The right hand side of G65 and G66 doesn't have parameter P to specify program number, system will send this alarm.		
Reason	Programming error		
Solution	Please check the machining program to ensure the use of G65 and G66.		

Alarm ID	COR–015	Alarm title	Too many M code
Description	There are more than 3 M codes in a single block.		
Reason	Programming error		
Solution	Please check the machining program to ensure that there are equal or less than 3 M codes in a single block		
Alarm ID	COR–016	Alarm title	Illegal variable access
Description	Accessing variables do not exist.		
Reason	Programming error		
Solution			

Alarm ID	COR–017	Alarm title	Label not found
Description	Cannot find out corresponding line number N in GOTO command		
Reason	Programming error		
Solution	Please check the machining program.		

Alarm ID	COR–019	Alarm title	sub program no M99
Description	Subprogram has no M99 to return main program		
Reason	Programming error		
Solution	Write M99 at the end of subprogram		

Alarm ID	COR–020	Alarm title	Too many G code
Description	There are more than 10 G codes in a single block.		
Reason	Programming error		
Solution	Dividing that single block into others single block that has less than 10G codes		

Alarm ID	COR–021	Alarm title	Too many (I,J,K) triples
Description	Repeat too much IJK command in the same single block.		
Reason	Programming error		
Solution	Please check the machining program.		

Alarm ID	COR–022	Alarm title	Use undefined workpiece coordinate
Description	Do not input G17, G18, G19		
Reason	Programming error		
Solution	Decide the working plane, and input G17, G18, or G19		

Alarm ID	COR–024	Alarm title	Invalid arc radius value
Description	When implementing G02, G03, appointing Arc end point and given radius is contradicted, given radius cannot meet appointing Arc end point. Ex: G03X1500Y4000R2000		
Reason	Programming error		

Solution	Check the program and recalculate.

Alarm ID	**COR–026**	**Alarm title**	**macro stack is empty**
Description	Empty stack still has value pop()		
Reason	The numbers of Push commands and Pop commands are not the same.		
Solution	Check the program to ensure that the number of Push commands is the same with that of Pop commands.		

Alarm ID	**COR–027**	**Alarm title**	**Invalid macro arguments**
Description	Macro Alarm.		
Reason	Once Macro finds out the unreasonable situation, machining program will be stopped and alarm will appear		
Solution	According to display content of alarm to find out where error is		

Alarm ID	**COR–040**	**Alarm title**	**Block end point exceed software limit**
Description	The coordinate in the program exceeds machine limit.		
Reason	Program error		
Solution	Check the machining program, and correct coordinate position		

Alarm ID	**COR–041**	**Alarm title**	**GOTO label must be integer**
Description	The input GOTO label is not an integer. Ex: GOTO 1 Correct GOTO 1. Wrong N1; Correct N1.; Wrong		
Reason	Program error		
Solution	Check the machining program, and input integer in GOTO label.		

Alarm ID	**COR–043**	**Alarm title**	**ASIN()/ACOS() operand must between -1.0 and 1.0**
Description	ASIN()/ACOS() Operand is not between -1.0 and 1.0.		
Reason	Programming error		

Solution	Check the machining program.

Alarm ID	COR–044	Alarm title	SQRT() operand should not be negative
Description	The square root of a negative value will be imaginary, but FBs-30GM does not provide this function.		
Reason	Programming error		
Solution	Check the program; enter a positive value in SQRT operand.		

Alarm ID	COR–047	Alarm title	M address should be integer
Description	M address is not an integer.		
Reason	Programming error		
Solution	Check the program, and use M address in integer.		

Alarm ID	COR–052	Alarm title	Sub-program number, P, should be integer
Description	If the sub-program number P is not an integer, FBs-30GM will send this alarm.		
Reason	Programming error		
Solution	Please check the program, and use the sub-program number P in integer.		

Alarm ID	COR–053	Alarm title	Repeat count, L, should be integer
Description	If the repeat times L is not an integer, this alarm will appear.		
Reason	Programming error		
Solution	Please check the program, and use the repetitive times L in integer.		

Alarm ID	COR–054	Alarm title	Incompatible data type
Description	When the data format doesn’t meet the requirements set by FBs-30GM, FBs-30GM will send this alarm.		
Reason	Machining program is not compatible with FBs-30GM.		
Solution	Make sure that the data format is suitable for FBs-30GM.		

Alarm ID	COR–059	Alarm title	Subprogram call sequence num., H, must integer
Description	Number H called in subprogram is not an integer		
Reason	Program error		
Solution	Change the number H of subprogram into an integer.		

Alarm ID	COR–060	Alarm title	M99 return sequence number, P, must integer
Description	The return sequence number P of M99 is not an integer.		
Reason	Program error		
Solution	Change the return sequence number P of M99 into an integer.		

Alarm ID	COR–064	Alarm title	P address must be integer
Description	If P address is not an integer, this alarm will be sent.		
Reason	Programming error		
Solution	Change P address into an integer.		

Alarm ID	COR–066	Alarm title	Inc. axis command and abs. axis command conflict
Description	Both G91 and G90 are in the same line.		
Reason	Programming error		
Solution	Decide to use incremental or absolute command, and enter the correct command.		

Alarm ID	COR–067	Alarm title	Arc center vector and radius conflict
Description	The arc end point is not on the arc created by the arc starting point and the specify center.		
Reason	Programming error		
Solution	Please check the machining program.		

Alarm ID	COR–070	Alarm title	Invalid G Code
Description	Enter incorrect G code to FBs-30GM.		
Reason	Program error		
Solution	Enter the valid G-code.		

Alarm ID	COR–071	Alarm title	No main program assignment
Description	The name of main program is not specified.		
Reason	The program is not loaded.		
Solution	Specify the name of main program.		

Alarm ID	COR–075	Alarm title	Exact stop wait timeout
Description	After 1 second sending Exact stop (G09/G61) command, If the difference between feedback and command exceeds allowable value, this alarm will be sent.		
Reason	Servo vibration		
Solution	1. Servo tuning 2. Change parameters		

Alarm ID	COR–076	Alarm title	G04 dwell time cannot be negative
Description	When input value of dwell time G04 is negative, this alarm will appear.		
Reason	Program error		
Solution	Check the machining program, and enter a positive value to G04		

Alarm ID	COR–201	Alarm title	Part program file not exist
Description	When specified program does not exist, this alarm will appear.		
Reason			
Solution	Ensure that program file exists		

Alarm ID	COR–202	Alarm title	Communication link failure
Description	When communication link is dropped, FBs-30GM will send this alarm.		
Reason			
Solution	Reconnect a good communication link		

Alarm ID	COR–204	Alarm title	File size too large
Description	When program file is too large, FBs-30GM will send this alarm		
Reason	Program error		
Solution	Reduce the program size, or split program into two subprograms.		

Alarm ID	COR–205	Alarm title	File content is empty
Description	After FBs-30GM loads the program, it finds out that the file content is null.		
Reason	Loading program error or CF card damaged		
Solution	Reload program or replace CF card		

Alarm ID	COR–207	Alarm title	Sequence number not found
Description	When sequence number is not found, FBs-30GM will send this alarm.		
Reason	Program error		
Solution	Use sequence number in the program range.		