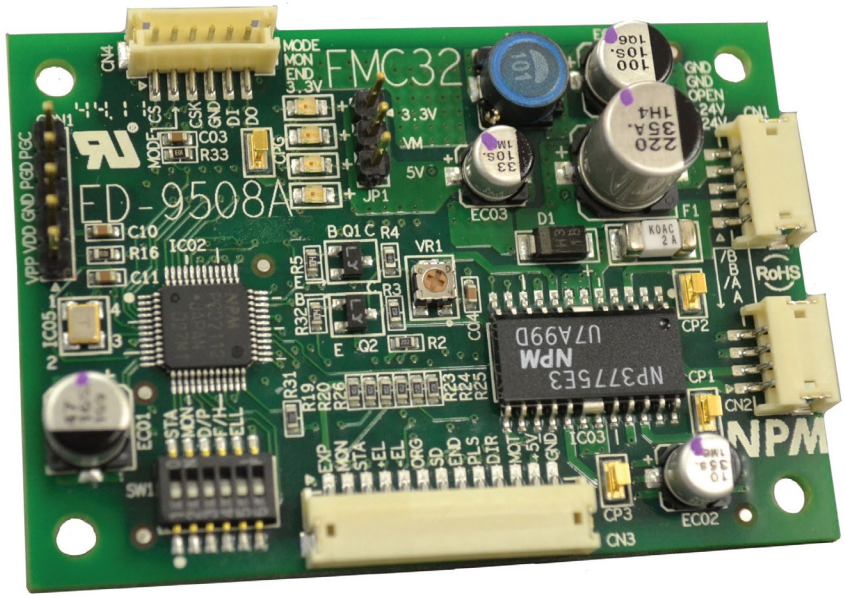
Start-Up Manual

FMC32

Compact Controller with Integrated Driver



NPM

— Blank Page —

Index

1. Preface

This manual shows explanations for basic operation and control of motors using the FMC32. Prior installation of hardware and software is required.

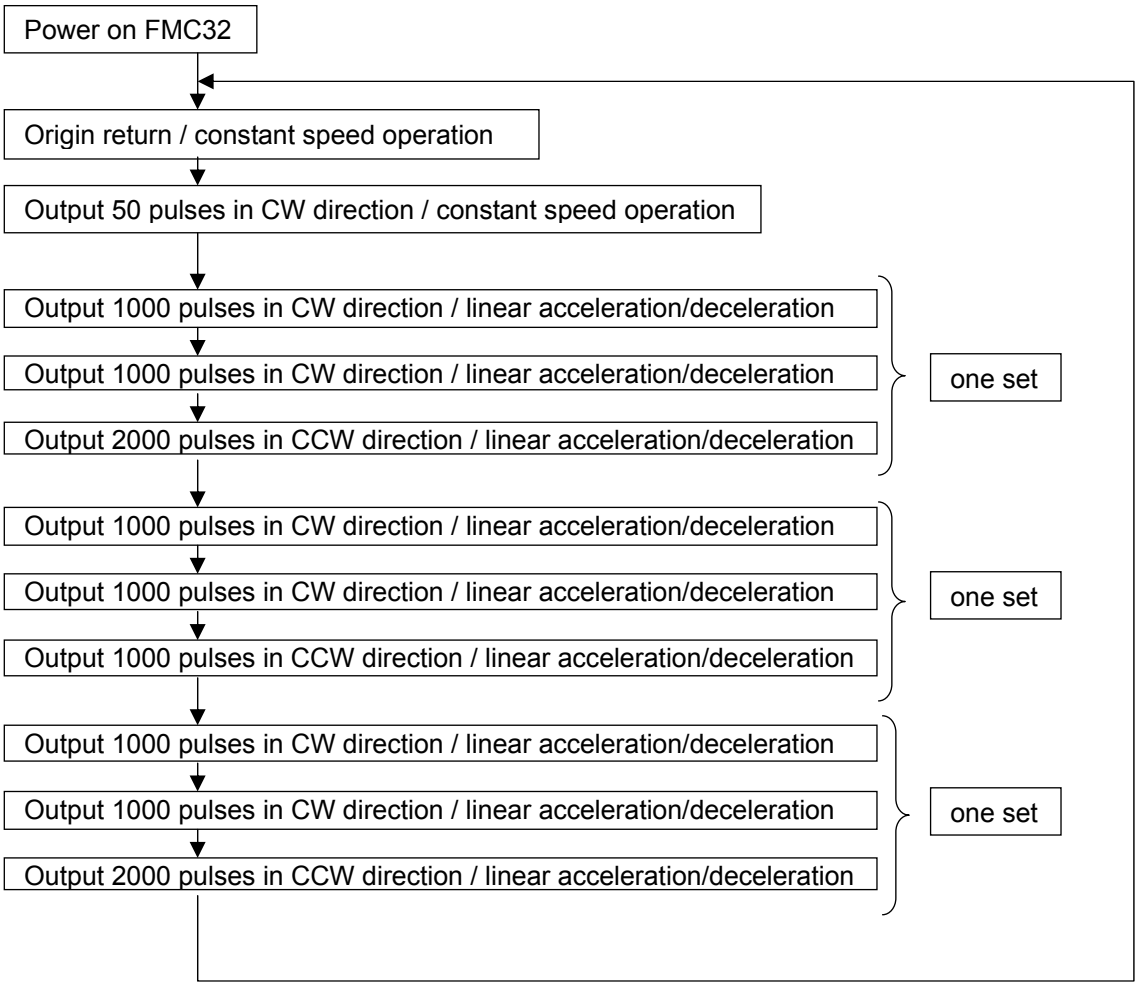
For how to use the FMC32, please see the following user's manual (described below) together.

-Compact controller with integrated driver, FMC32 Hardware, User's manual (Document No. YA 7162)
-FMC32 Control software, compact controller with built-in driver, User's manual (Document No. YA7163)
- USB to 4-wire serial conversion unit PUSB-3503, User's manual (Document No. YA7164)
- Pulse Control LSI PCD2112 for serial bus control, User's manual (Document No. DA70115)

2. Outline

2-1. Operation sequence to be designed this time

This manual describes the procedure to simply control the FMC32. The series of operation for the FMC32 is called an operation procedure (or operation sequence). The work to prepare an operation procedure is called "programming" and operation procedure data is called "program".



Outline of operation:

After power on, the FMC32 searches the origin point and the machine moves to the start point that is 50 pulses far from the origin point.

Then, after it repeats three sets of operation from positive direction to negative direction, it repeats operation to search the origin point again.

Other conditions:

The initial speed is 10pps and the maximum speed is 500pps.

The acceleration time and deceleration time is both 500ms.

The constant speed is 200pps.

Between each operation within one set operation, 200ms waiting time is inserted.

2-2. Environment requirements

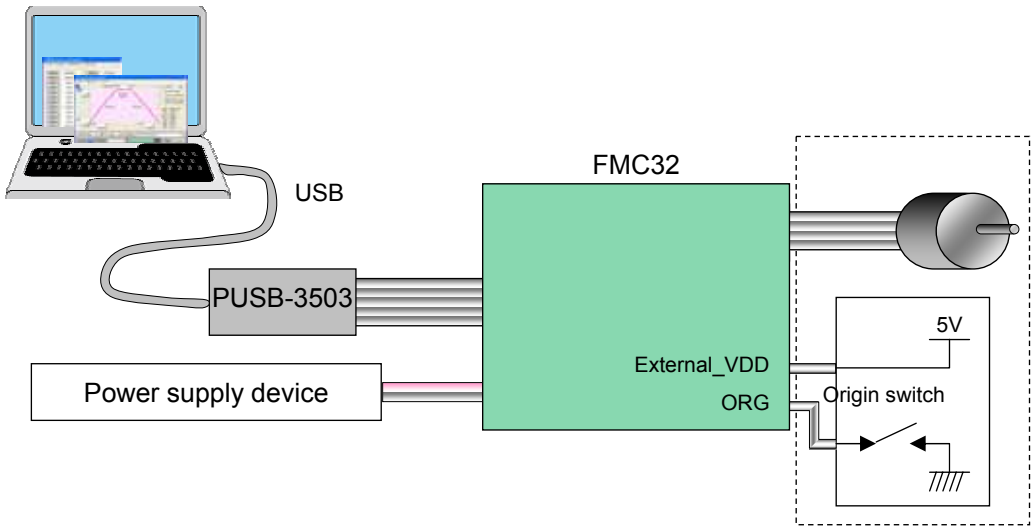
The following environment is needed to operate the operation sequence created according to this manual.

Item	Contents	Notes
PC	It is used for creating an operation pattern, running a program simulation and writing operation data to FMC32. WindowsXP SP3 or Windows 7 is necessary as OS.	Indispensable
PUSB-3503	It is our product used to control FMC32 from a PC .	Indispensable
FMC32	It is a product to be operated according to this manual.	Indispensable
Power supply device for FMC32	It is a power supply device to provide power to FMC32. It can apply 6.5V to 40V. It is used to drive the connected motor. Prepare one that is compatible with the motor to be used.	Indispensable
Stepper motor	A motor to be driven by FMC32. PFCU25-24C1G (our product) , etc.	
Origin switch	It is used to notice an origin point and you can use a toggle switch as a substitute for this. Ideal one is a slider switch with an origin point sensor.	
Power supply for origin switch	It is a 5V power supply to drive an origin return switch and photo-coupler of FMC32.	

Items that are not marked as "Indispensable" in the Notes field, are not necessary for basic operation of the device. Even if you do not prepare these items, you can confirm the programmed operation by simulation.

2-3. Connection examples

Connect the devices as follows.



The products surrounded by dots in the above figure do not have to be prepared if you cannot prepare them.

Make the origin switch OFF. The FMC32 recognizes that a point where the origin switch turns on is an origin point.

2-4. Order to power on

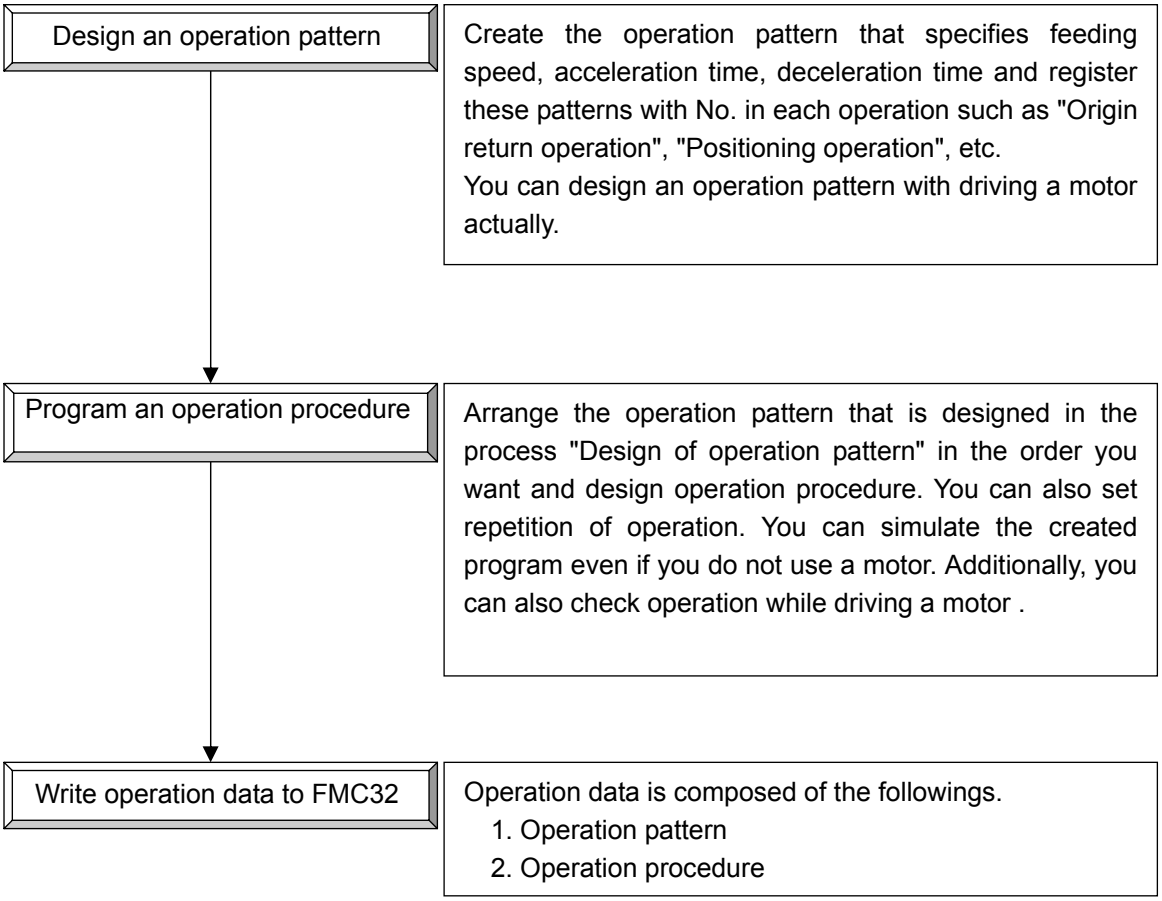
Please power the products on in the following order.

Connect FMC32 with PUSB-3503 and FMC32 with an origin switch and a motor.
Power on FMC32 and the origin switch (5V). You can power on whichever first.
Connect PUSB-3503 with PC through an USB cable (PUSB-3503 is applied power from USB bus-power).

Please power them off in the reverse order.

2-5. Outline of operation procedure

The design for operation procedure to FMC32 is classified to three steps as follows.

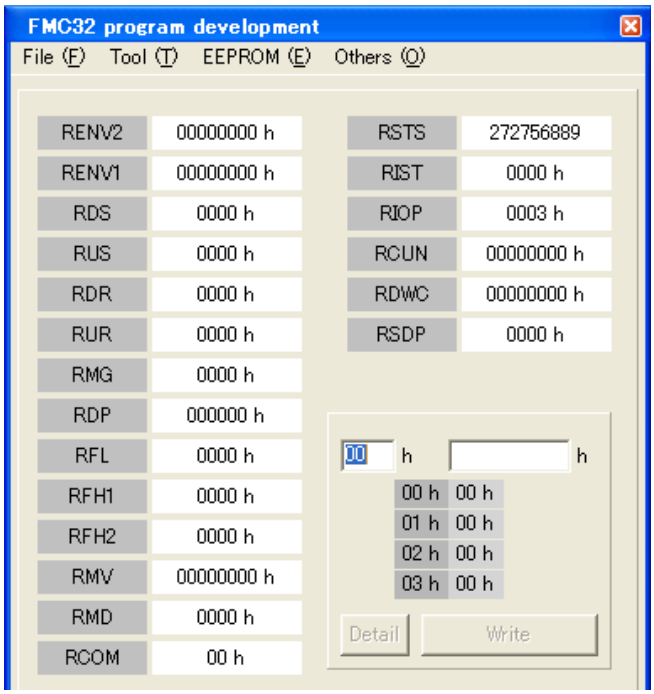


3. Prepare operation

3-1. Start a program

Double-click the following icon.

The software starts and the following main window appears.

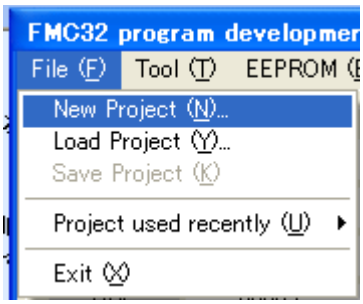


[Main window]

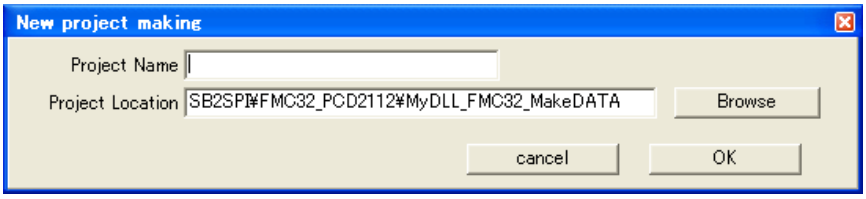
3-2. Prepare a project

First, prepare a project to save data.

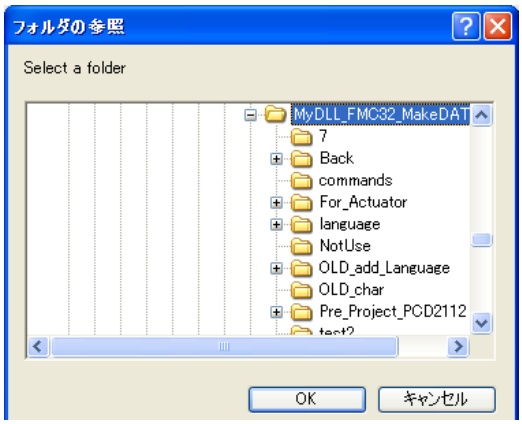
1. Select "New Project" from the "File" menu.



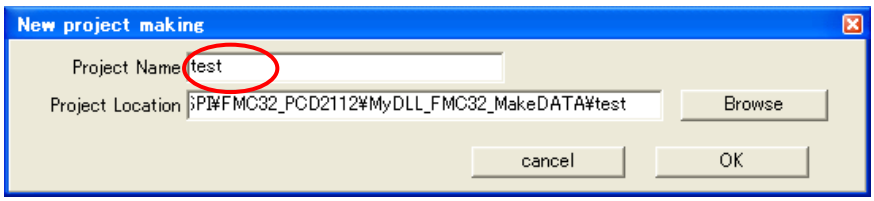
The following window appears.



2. Click the "Browse" button and select the folder where you want to save the project.
   A new folder that has the same name as the “Project Name” specified, is made under the folder you select. The operation data is saved in this folder.



3. Input a project name.
   After you input a project name, click the "OK" button.

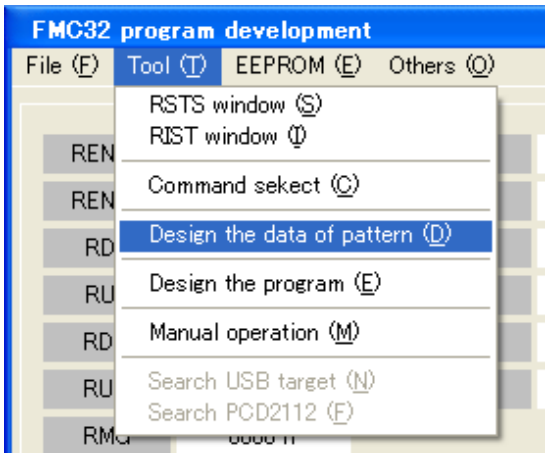


3-3. Design operation patterns

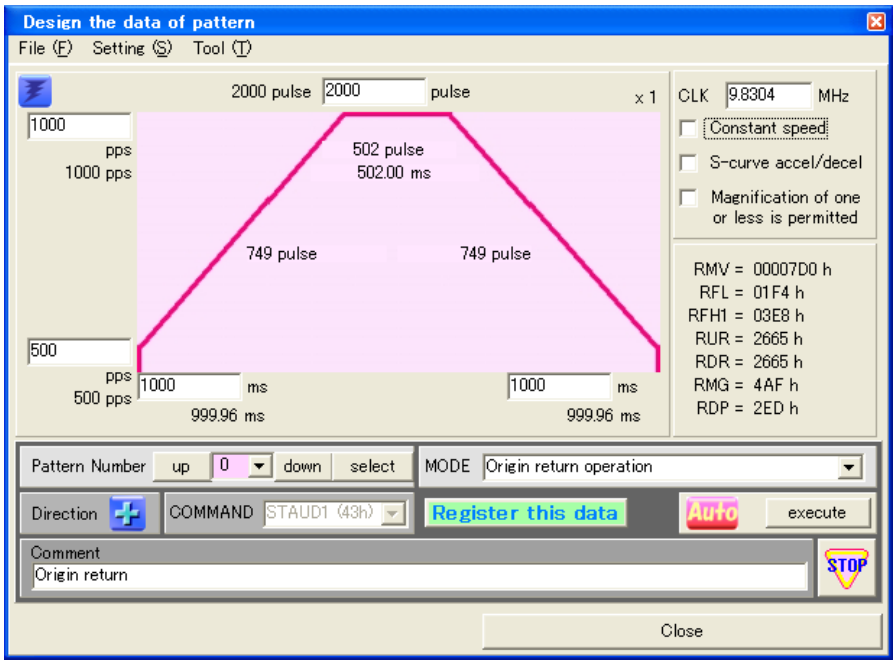
"Operation pattern" means a operation such as positioning operation and origin return operation, etc. You can set the factors such as "initial speed", "maximum speed", "acceleration time", "deceleration data" and "feeding amount" that are necessary for this positioning operation while you drive a motor actually.

3-3-1. "Design the data of pattern" window

Select "Design the data of pattern " from the "Tool" menu.

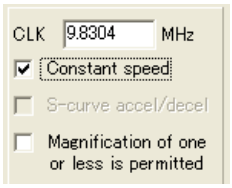


The following window appears.

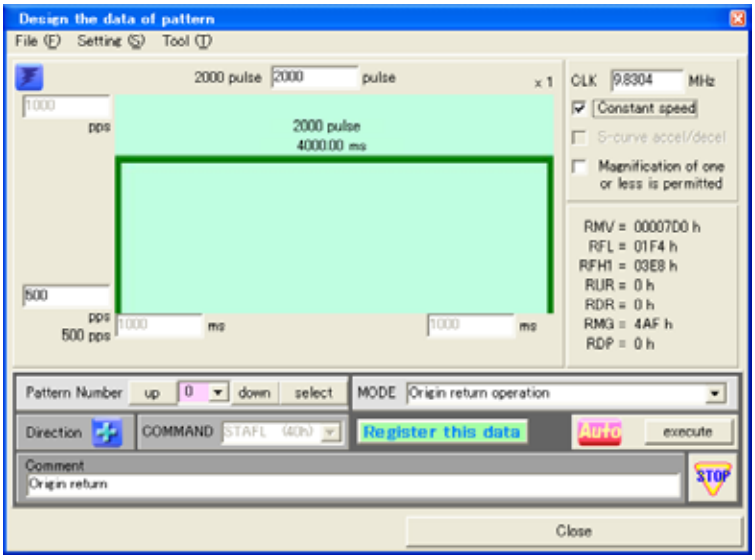


3-3-2. Design and registration of Origin return operation

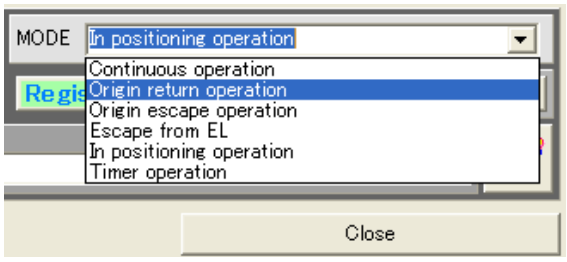
Check the "Constant speed" checkbox in the programming window.



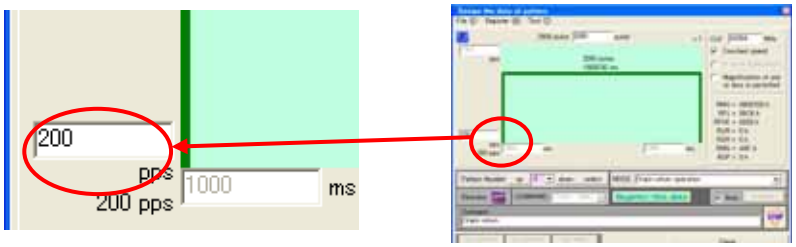
The programming window will display as follows.



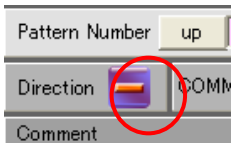
Select "Origin return operation" from the pull-down menu of the mode selection.



Then, change the value in the following column to "200".

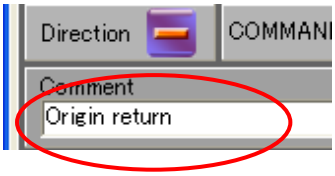


Next, click the "Direction" icon to change it to "-".



With that, operation pattern for origin return operation is prepared.

Please input your comment if possible. It helps you to make subsequent programs.



[Check operation]

If a motor is actually connected with the FMC32, you can operate the motor. This allows you to check operation pattern.

Click the "Execute" button.

After the motor starts to operate, the motor will stop when the ORG switch turns on.

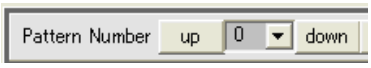
Note 1: Do not uncheck the "Auto" checkbox.

Note 2: You cannot check an origin return operation when there is not an ORG switch.

Note 3: If a motor is not connected with FMC32, you can check operation condition by RCUN register value's increasing and decreasing.

[Check pattern number]

Please confirm that the pattern number is "0".

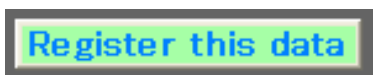


[Register operation pattern]

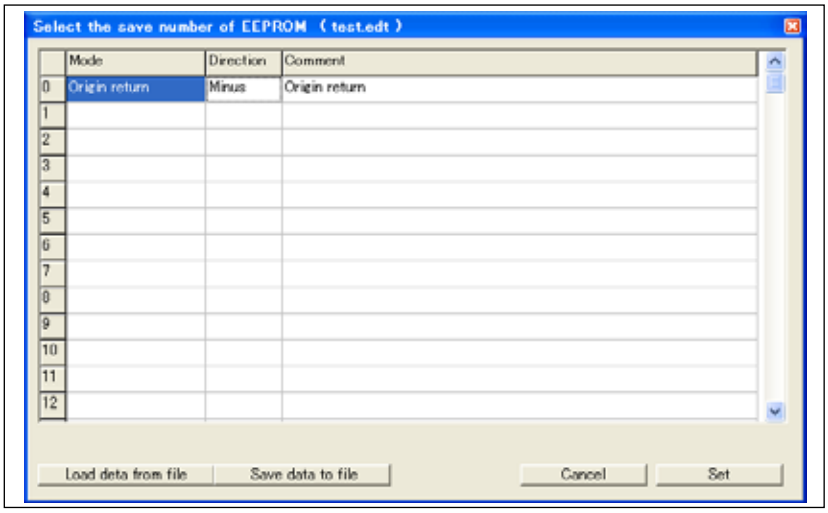
Register the prepared operation pattern. If you do not register it, you cannot use it with the program you will use. Please register the data after you confirm that there are no problems.

Up to 32 operation patterns can be registered.

Please click the "Register this data" button.



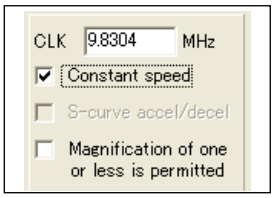
The following window appears.



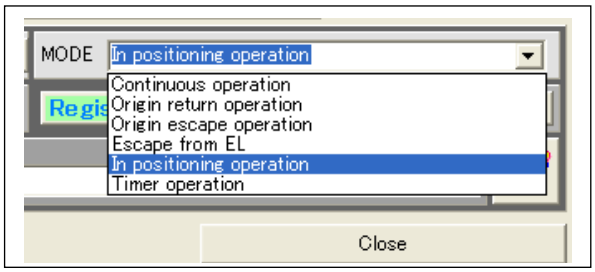
"Operation mode", "Direction" and "Comment" columns appears.
Other information is not shown. Therefore, we recommend that you input a name that you can identify the operation in the "comment" field.
Click the "Set" button" to register origin return operation as patter No.0.

3-3-3. Design and register operation to output 50 pulses in CW direction

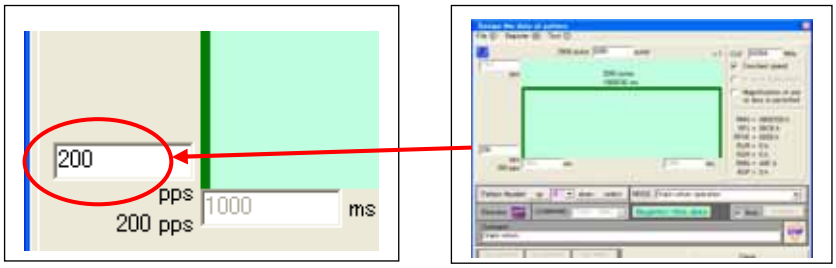
Keep the "constant speed" check box checked in "Design of the data of pattern" window.



Next, select "In positioning operation" from the pull-down menu next to "Mode".



The value in the following does not change from "200".

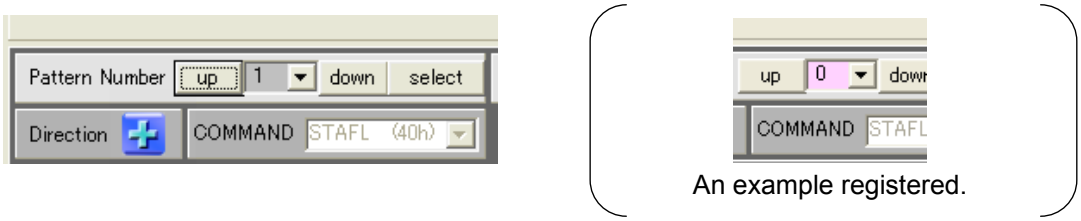


Change the following value to 50. This is the number of pulses to be output.

Then, click the icon next to "Direction" to make it from "-" to "+".
Input comments as necessary.

With that, you designed an operation to output 50 pulses in CW direction.

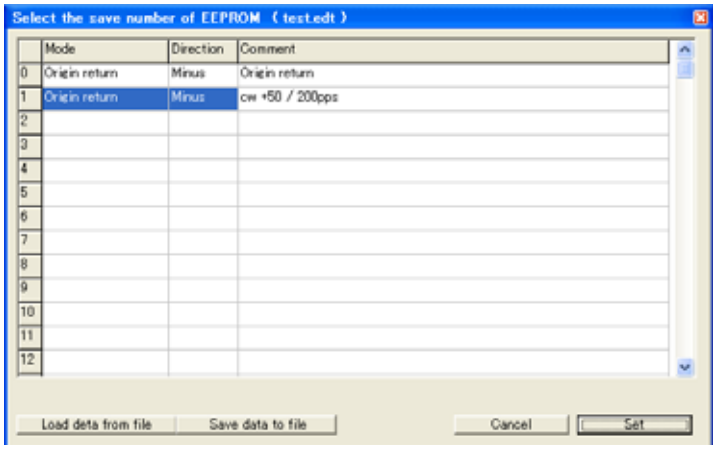
Next, you will register this designed pattern. Before you register it, click the "up" button next to "Pattern Number" to change register number to "1". As the following figure shows, the gray background color of the field means that operation is not registered in this pattern number.



An example registered.

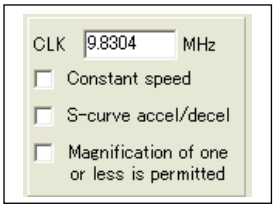
Click the following "Register this data" button.

Click the "Set" button to register this operation to pattern No.1.

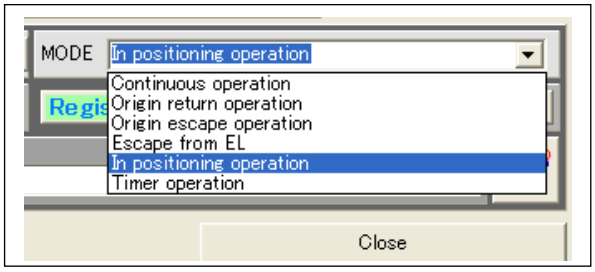


3-3-4. Design and register operation to output 100 pulses in CW direction

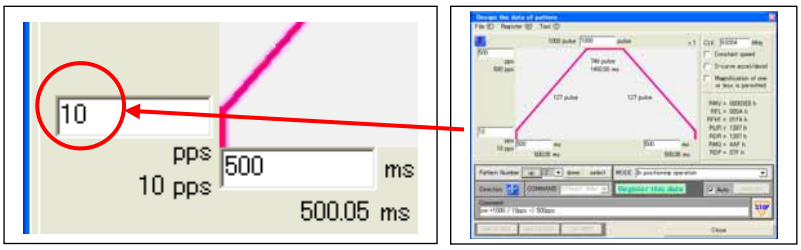
Remove the check in the "Constant speed" check box.



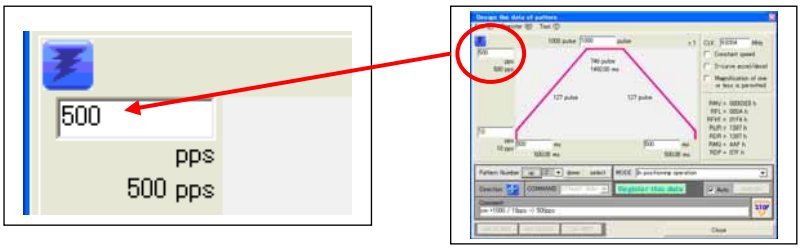
Next, select "In positioning operation" from the pull-down menu next to "Mode".



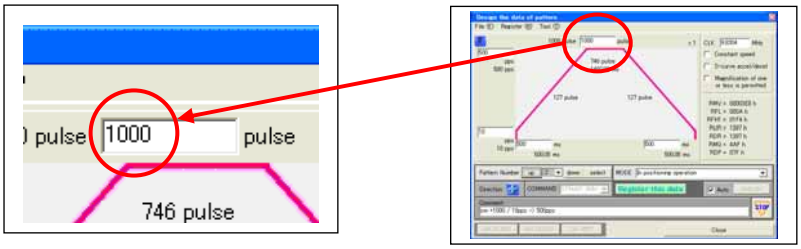
Set an initial speed as follows.



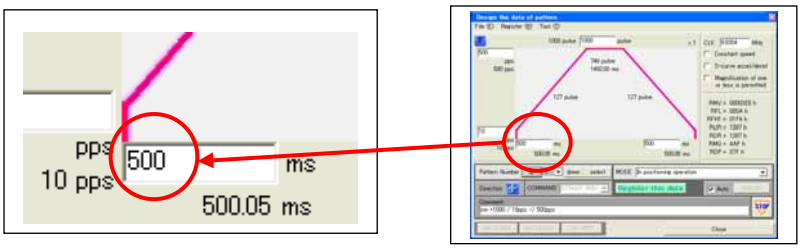
Put the following value in the maximum speed field.



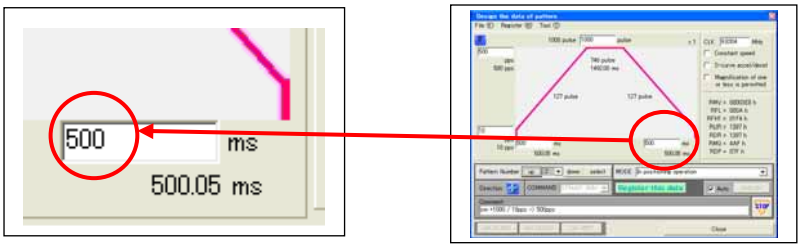
Set the number of pulses to be output.



Set an acceleration time.



Set a deceleration time.

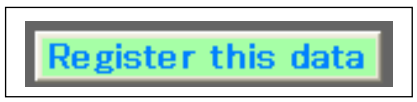


Next, click the "Direction" icon to change it to "+". Input comments.

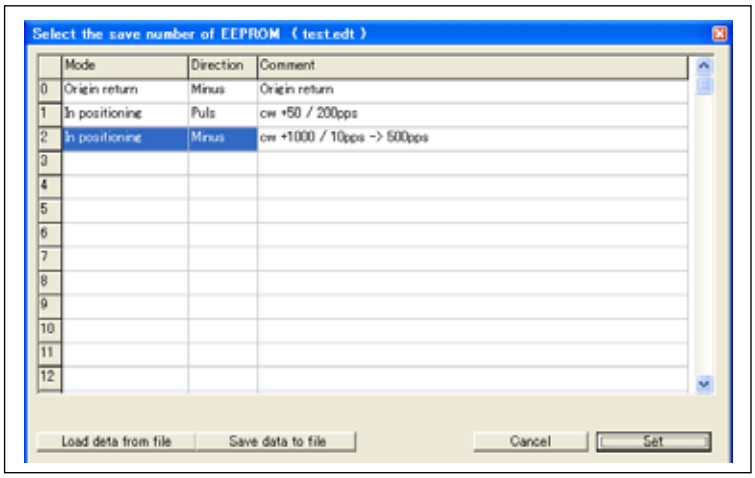
With that, you designed the operation to output 1000 pulses in CW direction.

Then, you will register this created pattern. Before register it, click the "up" button next to "Pattern Number" to change register number to "2".

The following "Register this data" button.



Click the "Set" button to register this operation to pattern No.2.

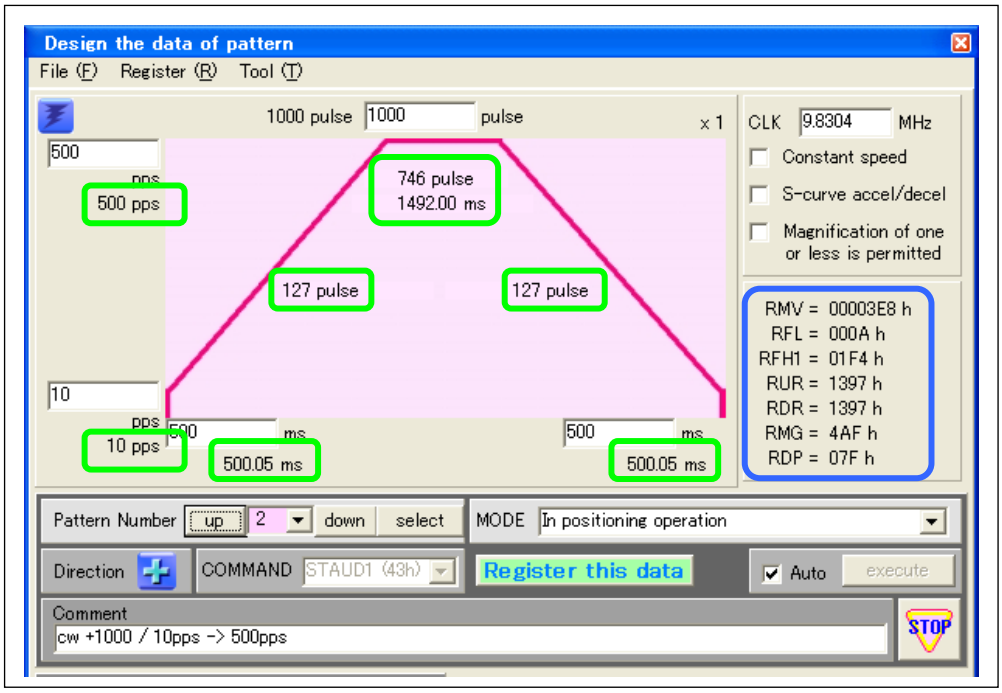


[Confirmation value to be set]

If the work has gone correct, the "design the data of pattern" window should show as follows.

FMC uses PCD2112 and values are set to each register automatically so as that this PCD2112 operates using the specified values.

The field bounded by the blue line in the following figure shows the actual value that is set to PCD2112 registers.

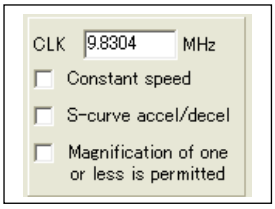


The value bounded by green lines in the upper line shows how PCD 2112 operates using the value set in registers actually.

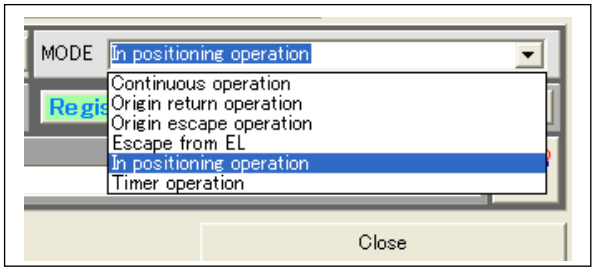
PCD2112 does not necessarily operate in accordance with the value you set. The reason is because some errors are observed between the result calculated in PCD2112 and the setting value under PCD2112 specifications. For example, if you input 500ms for acceleration time, acceleration time will be 500.05ms in calculation.

3-3-5. Design and register operation to output 2000 pulses in CCW direction

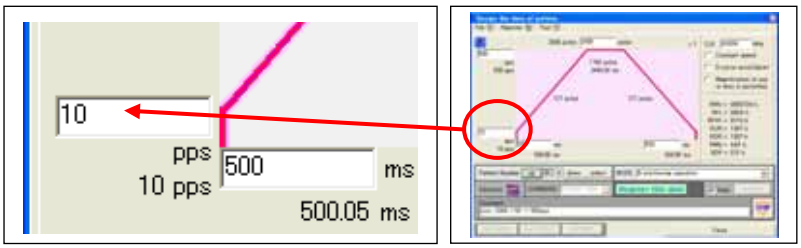
Keep the "Constant speed" check box unchecked in the "Design the data of pattern" window.



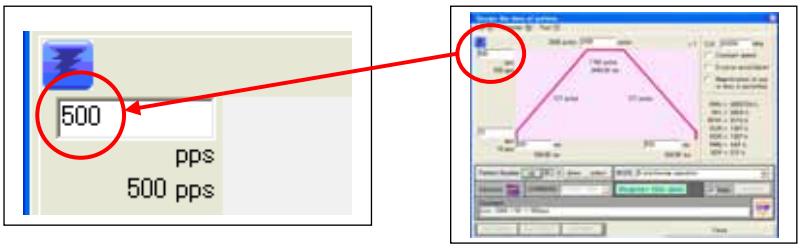
Next, select "In positioning operation" from the pull-down menu next to "Mode".



Initial speed is not changed.

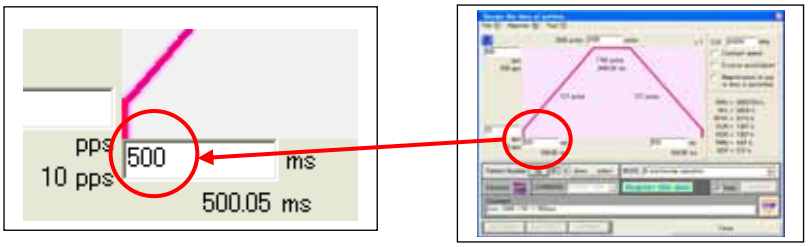


Maximum speed is not changed.

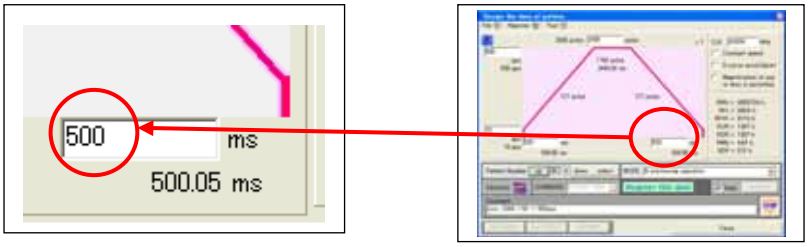


Set the number of pulses to be output.

Acceleration time is not changed.



Deceleration time is not changed.

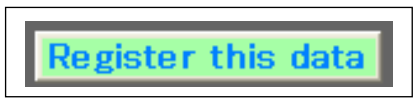


Then, click the icon next to "Direction" to make it from "+" to "-".
Change comments.

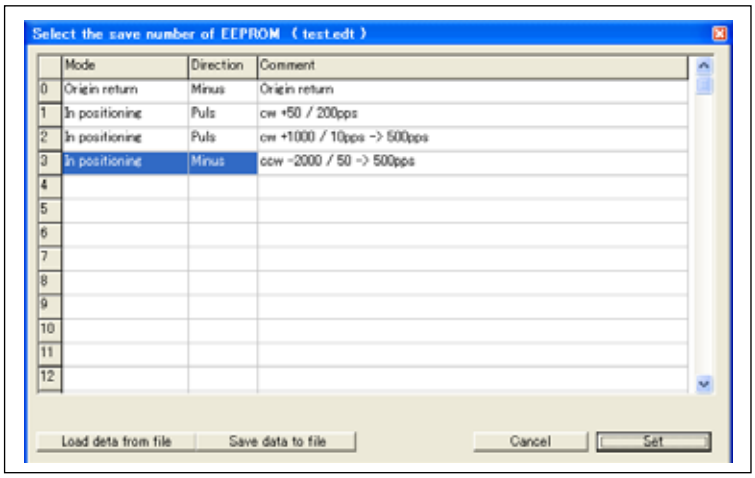
With that, you designed an operation to output 2000 pulses in CCW direction.

Then, you will register this created pattern. Before you register it, click the "up" button next to "Pattern Number" to change register number to "3".

Click the following "Register this data" button.



Click the "Set" button to register this operation to pattern No.3.



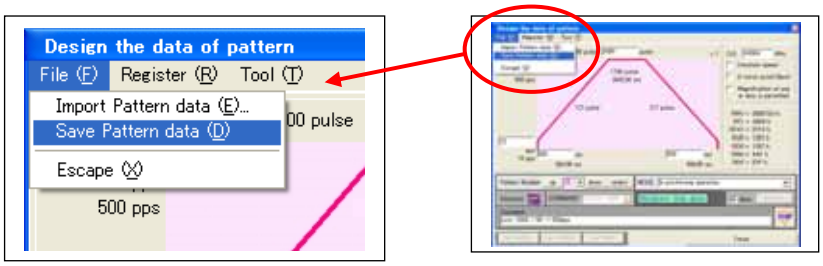
3-3-6. Save the registered operation pattern data

Save registered data as a file. If you exit the program without saving, all the registered operation patterns are lost.

There are two ways to save data. Select one to proceeded.

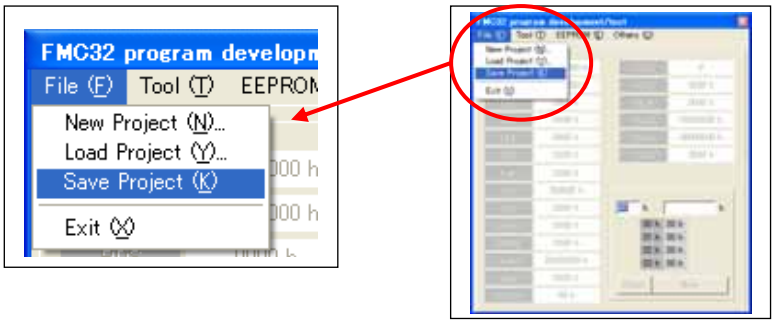
[Operation pattern save procedure 1]

Select "Save Pattern data" from the "File" menu of the menu bar in "Design the data of pattern" window.



[Operation pattern save procedure 2]

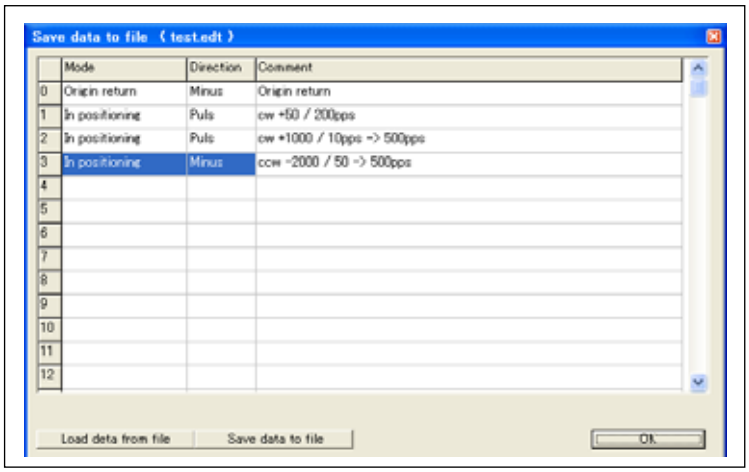
Select "Save Project" from the "File" menu of the menu bar in the main window.



3-3-7. Check registered data

You can check what operation is registered in which pattern No.
Click the "select" button.

Registration window appears and the outlines of the operation pattern appear in the pattern number field that operation has already registered.



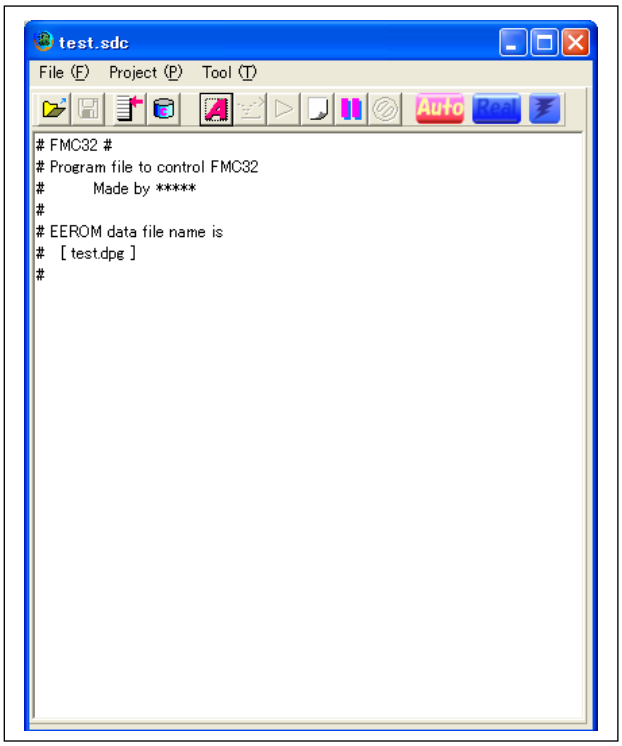
All information is not shown. Therefore, we recommend you put descriptive comments to help you easily identify operations.

3-4. Operation procedure programming

Next, the operation procedure will be designed by arranging the designed operation patterns.
You can check designed operation procedure to be processed in the order you want.

3-4-1. Show editor window

Select "Design the program" from the "Tool" menu.
The following editor window appears.



Comment information is already described by default. Do not delete them.
Do not change the content of the first line. (You can change the contents of lines below the first line.)

All lines with "#" head are handled as comments. Though "#" is in the middle of lines, character strings that follows "#" are handled as comments if there is a "#" that is separated with more than one space.

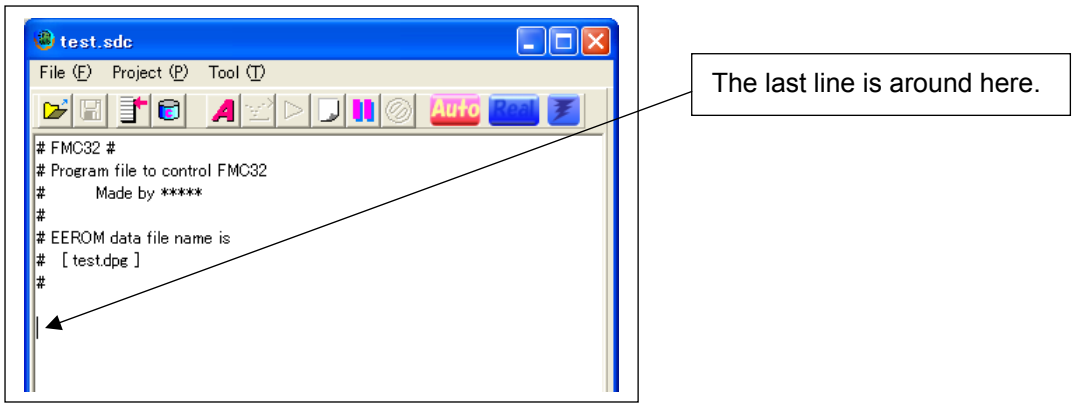
The character strings other than comments is handled as program.
FMC32 processes the specified operation in order from the first line of the program in the stand alone mode after it is powered on

3-4-2. Programming flow

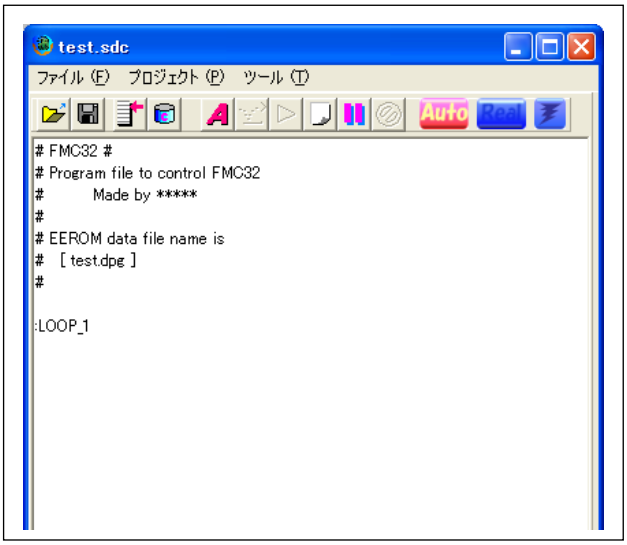
The following shows the operation procedure described in "2-1. Operation sequence to be designed this time".

[Label setting 1]

This operation procedure repeats the same operation continuously.
The jump point for repeated operation is specified as "Label"(character string).

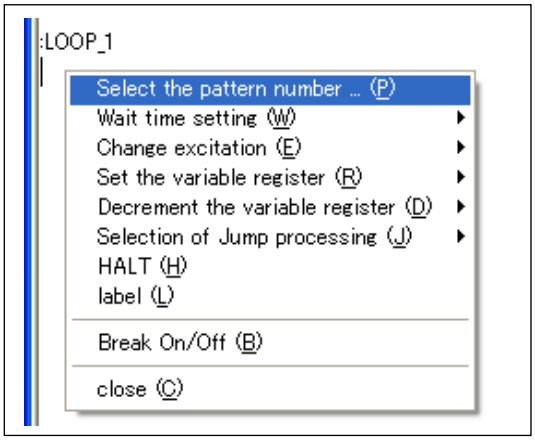


Click the last line of the editor and put the cursor in the last line.
Input "Loop_1" using keyboard. Do not forget to attach a head colon (:).
Attaching a (:) colon at the head of line shows that the character string is a label.
Label characters are composed of only alphanumeric characters and underscores.

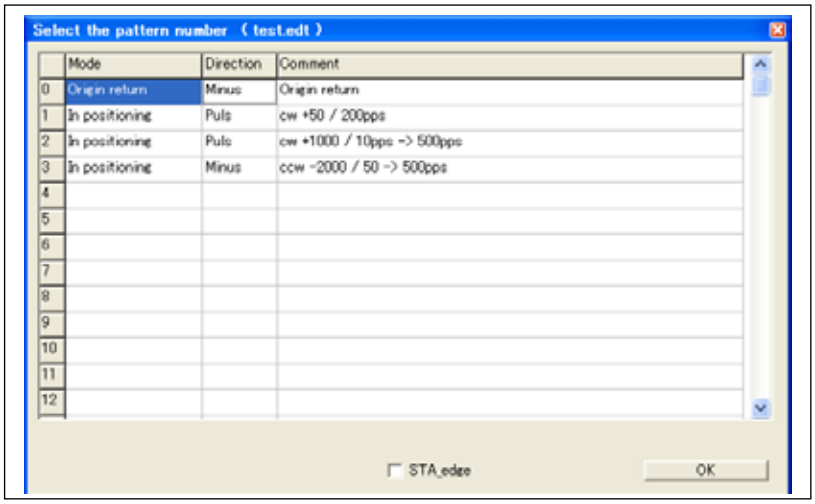


[Origin return]

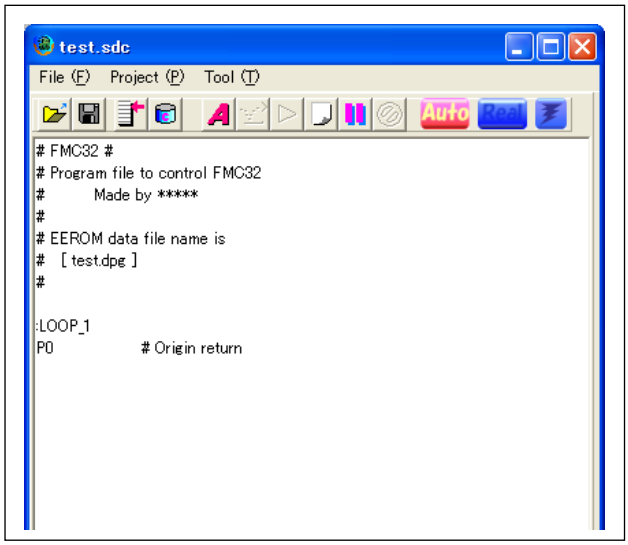
Put the cursor under the label and right-click there to show the following pop-up menu.
Select "Select the pattern number..." from the pop-up menu.



The list of the operation patterns that have been designed and registered appears. In this list, select 0 (Origin return) and click the "OK" button.



The program for selected operation pattern (pattern No. 0) is added as follows.
The line shows that "P0" operates pattern No. 0.
The character string that follows "#" is a comment to be input when operation pattern is designed.

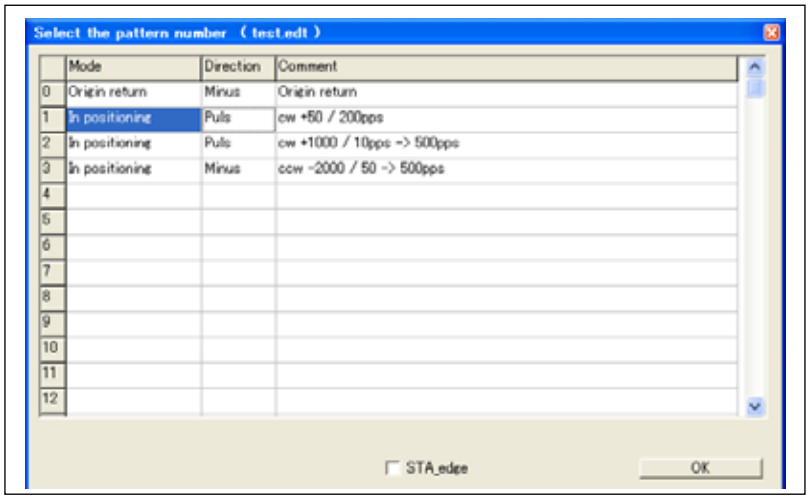


[Operation to output 50 pulses in CW direction]

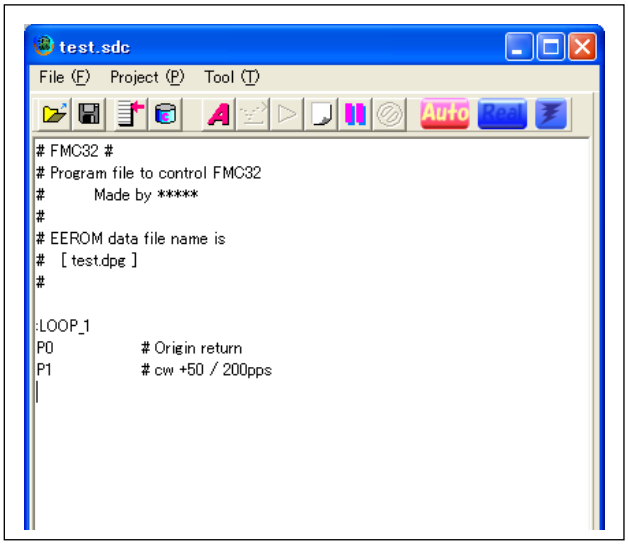
Put the cursor under the line of "P0" and right-click there to show the following pop-up menu.

Select "Select the pattern number..." from the pop-up menu.

The list of the operation patterns appears. In this list, select the first "In positioning" and click the "OK" button.

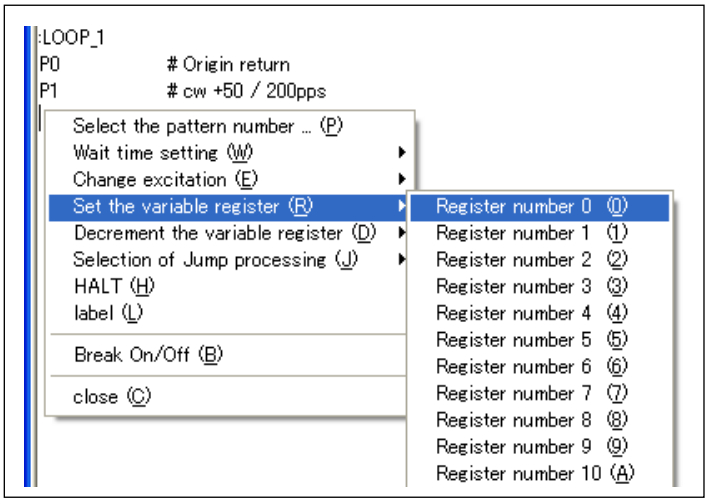


"P1" and the comment is added as follows.

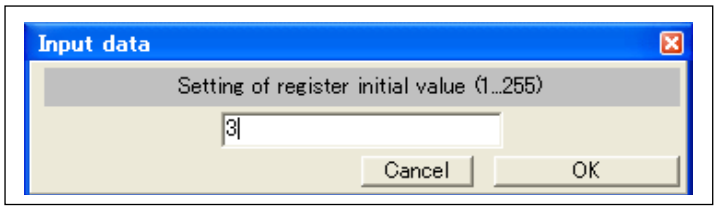


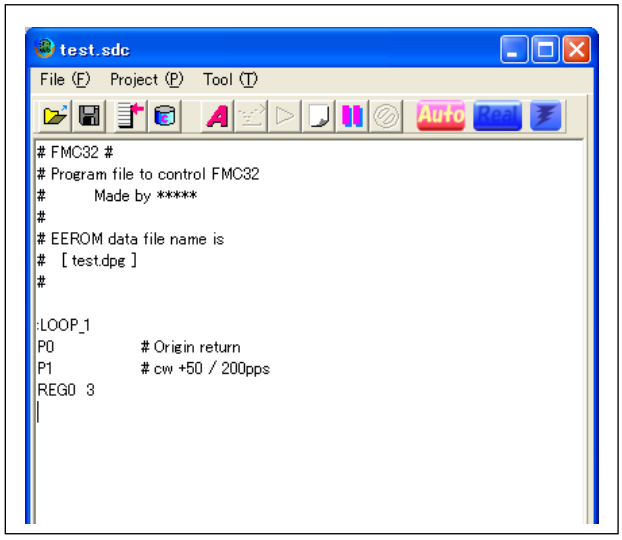
[Set repeating time]

Set the repeating time so as to repeat one set operation from positive direction to negative direction three times. Put the cursor under "P1" and right-click it. Select "Set the variable register" from the pop-up menu and select "Register number 0".



The following window to set register initial value appears. Input "3" in this field and click the "OK" button. The program that assigns "3" to REG0 register is added.



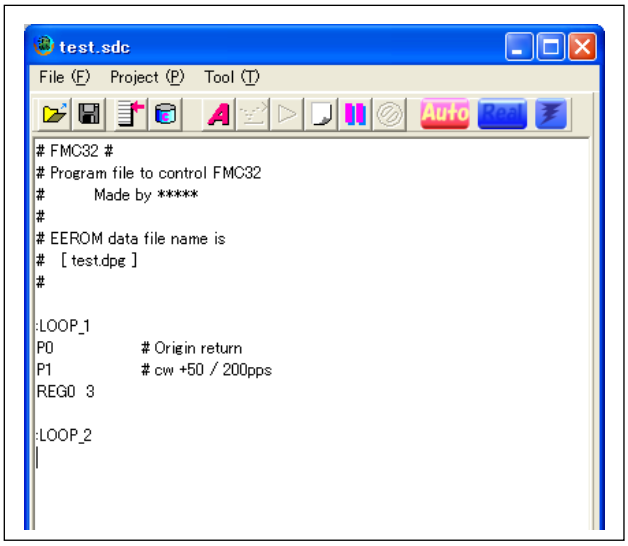


[Label setting 2]

Set the repeating time so as to repeat one set operation from positive direction to negative direction three time. The jump point for reported operation is specified as "Label".

Put the cursor under "REGO 3" line and begin a new line. (This operation is to improve appearance and is not indispensable.)

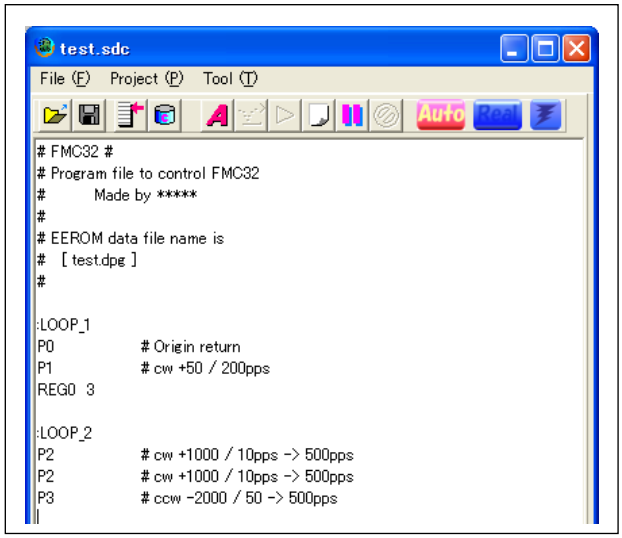
Next input ":LOOP_2". Input the head ":" without fail.



[Set one operation set from positive direction to negative direction]

Set the repeating time so as to repeat one operation set from positive direction to negative direction three time. The setting method is the same as "origin return" or "operation to output 50 pulses" described in the above.

The window should be as follows after programming.

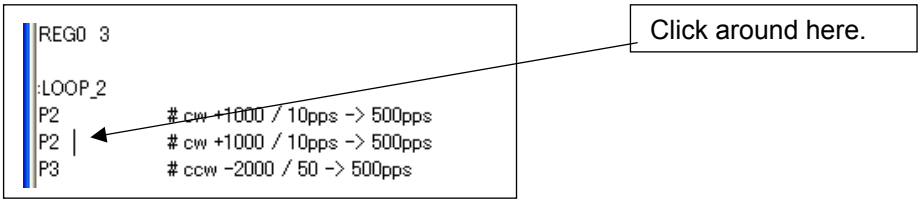


There are two "P2" lines and one "P3" line.

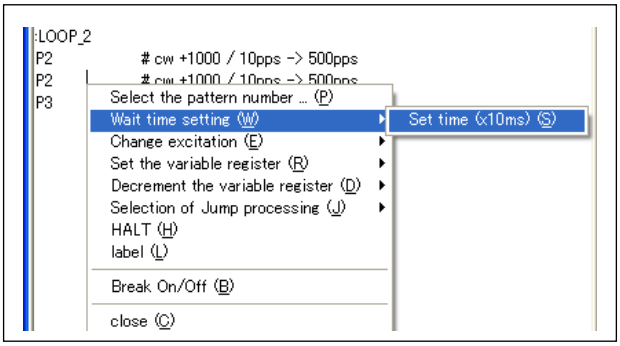
[Put the waiting time]

This operation is to execute the next operation after each operation of one operation set from positive direction to negative direction is completed and 200ms waiting time is passed over.

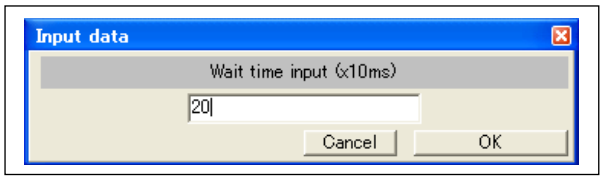
Click the cursor the second "P2" line to be put.



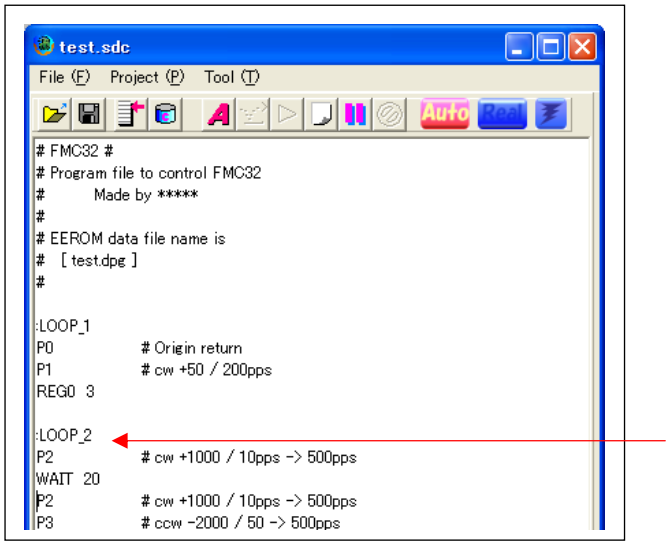
Right-click here and select "wait time setting" from the pop-up menu and select "Set time "x10ms).



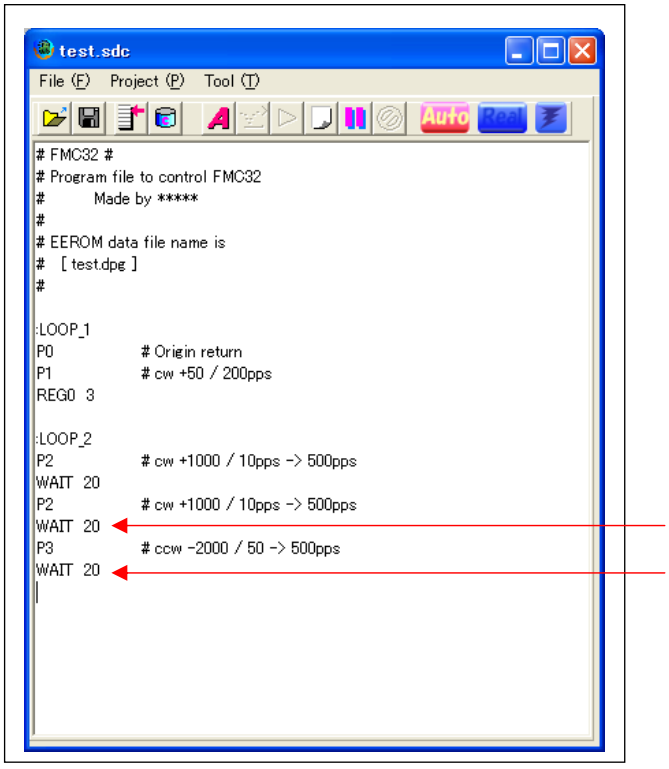
The "wait time input" field appears. Input "20" in this field and click the " OK" button. Waiting time (that is 10 times of the input value in this field) is taken.



The line specified by the cursor comes down and the program "WAIT 20" is inserted.



Insert the waiting time by the similar operation and edit the program as follows.



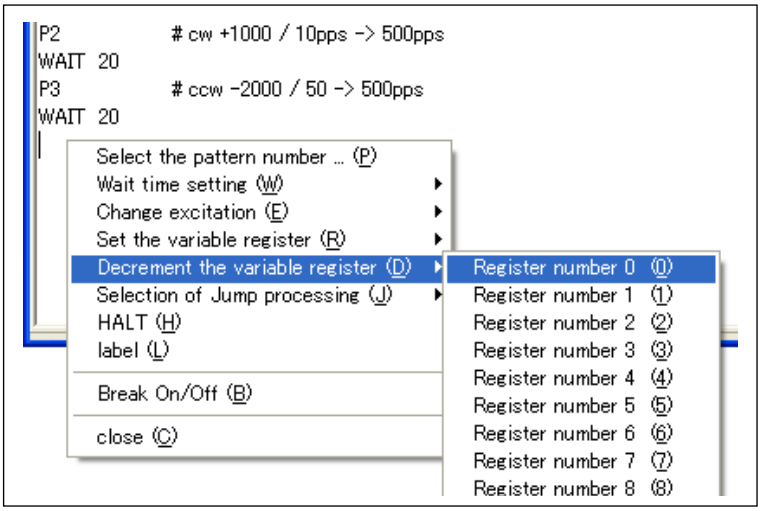
[Control to repeat operation three time]

Decrement the REG0 value and operate repetition control by confirming whether or not this result is 0.

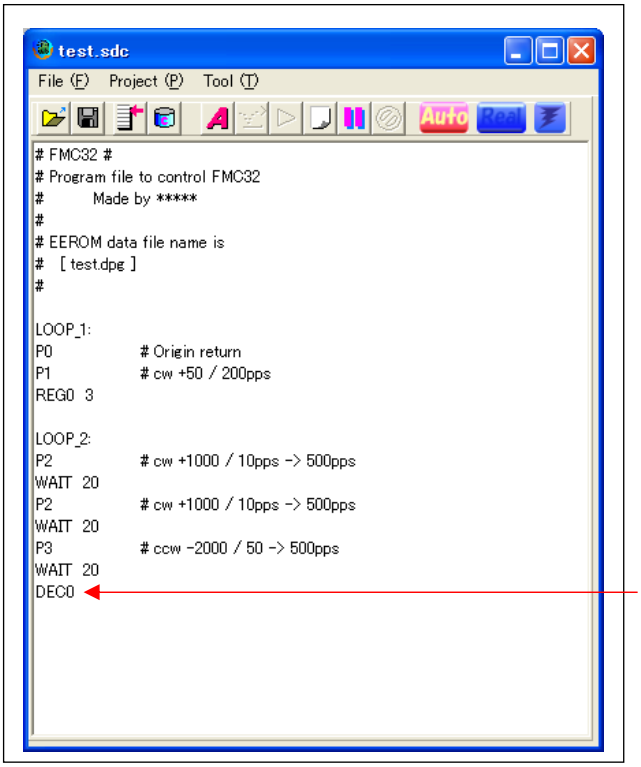
First decrement REG0.

Click the line under the third "WAIT 20" to move the cursor.

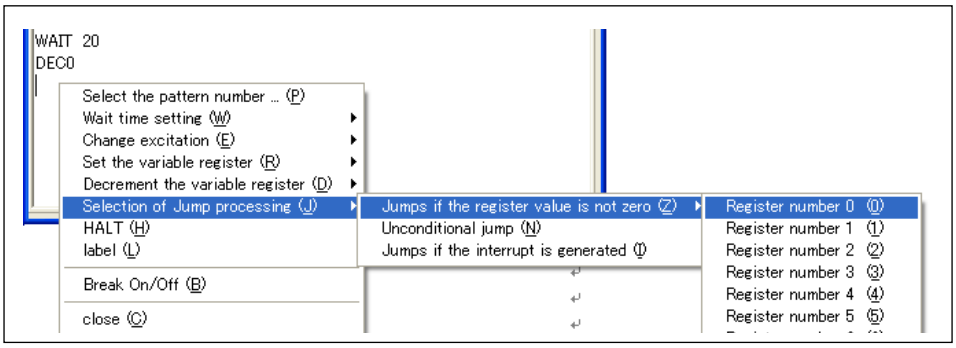
Right-click here and select "Decrement the variable register" from the pop-up menu and select "Register number 0".



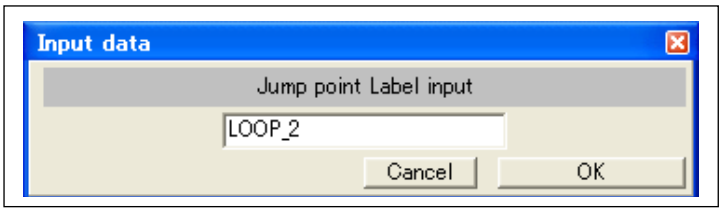
Decrement command of REG0 is inserted.



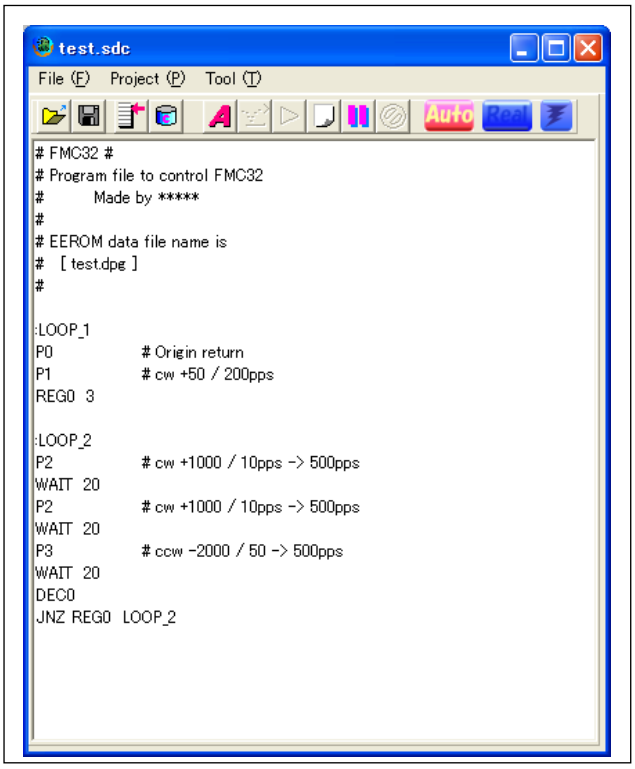
Next, jump command is described.
Make the cursor put under the inserted "DEC0" line and right-click. Select "Selection of Jump processing" from the pop-up menu and select "Jumps if the register value is not zero" and "Register number 0".



"Jump point Label input window" appears. Input "LOOP_2" in this field. The head (:) is unnecessary to be input.



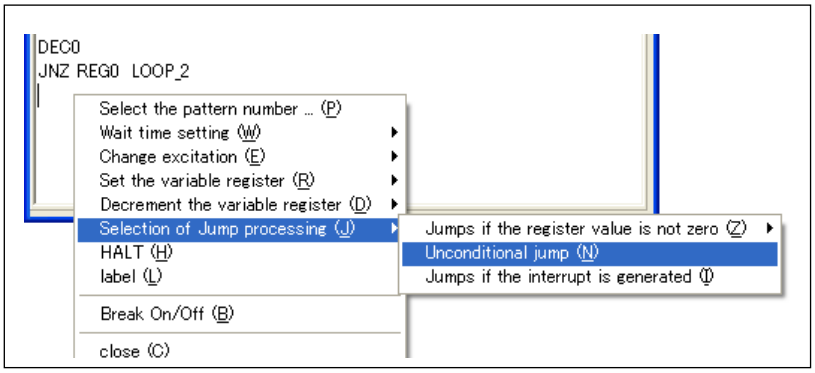
A jump command is added.



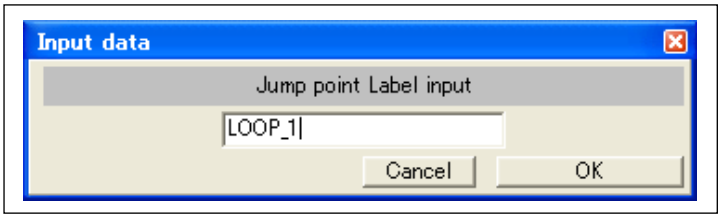
[Process to jump to the head of the program]

After the repetition operation is completed three times, jump to the head of the program to operate origin return operation again.

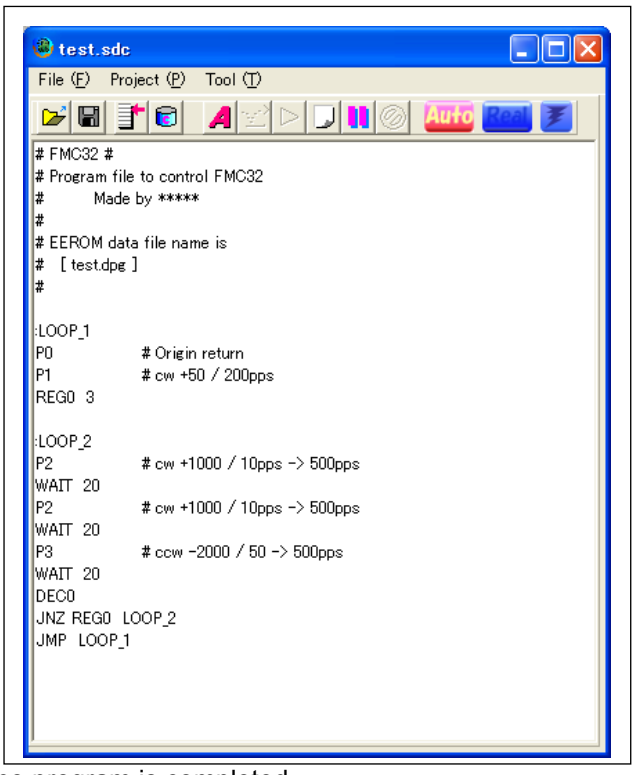
Place the cursor under "JNZ REG0 Loop_2" and right-click, and select "Selection of Jump processing" from the pop-up menu and select "Jumps if the register value is not zero."



"Jump point Label input" window appears. Input "LOOP_2" in this field. The head (:) is unnecessary to be input.



A jump command is added.

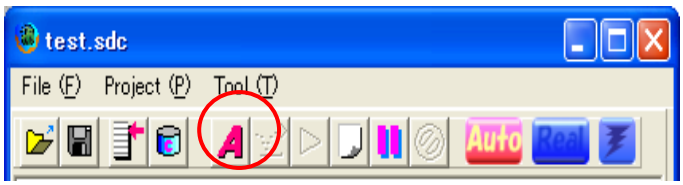


With this, the program is completed.

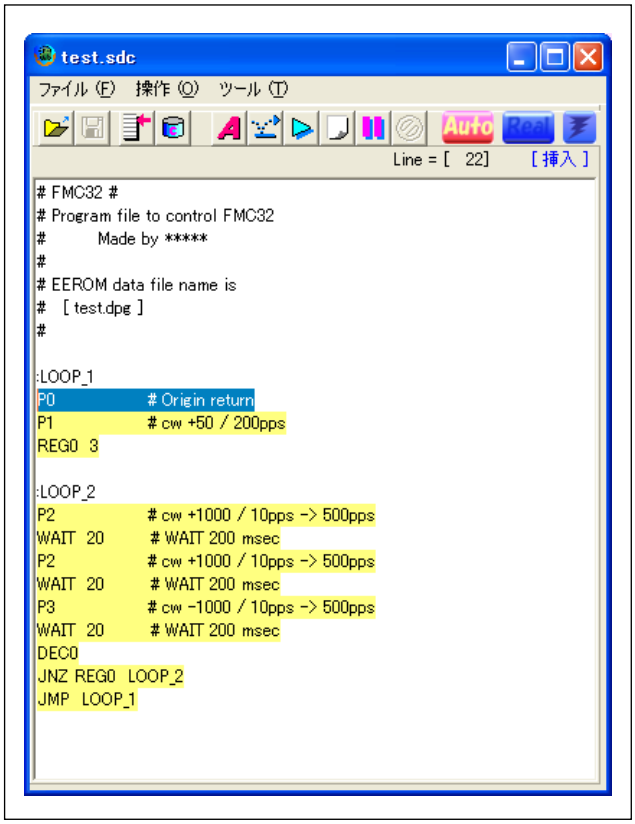
3-4-3. Confirm operation of program （Simulation）

Confirm whether or not the created program is operated in the order you want.
First, confirm the operation without using a motor.
In the simulation of operation confirmation, commands are not sent to the motor driver. The purpose of simulation is to confirm whether the designed program is executed in the order you want.

[Assemble program]
Click the following "Assemble" button.

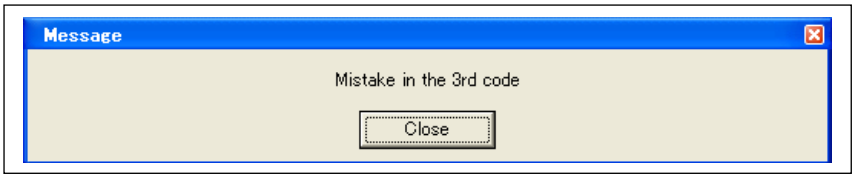


If there are no errors in the designed program, the part of the program (the character string but comment and labels) is highlighted in yellow and the first line of the program is displayed inverted.

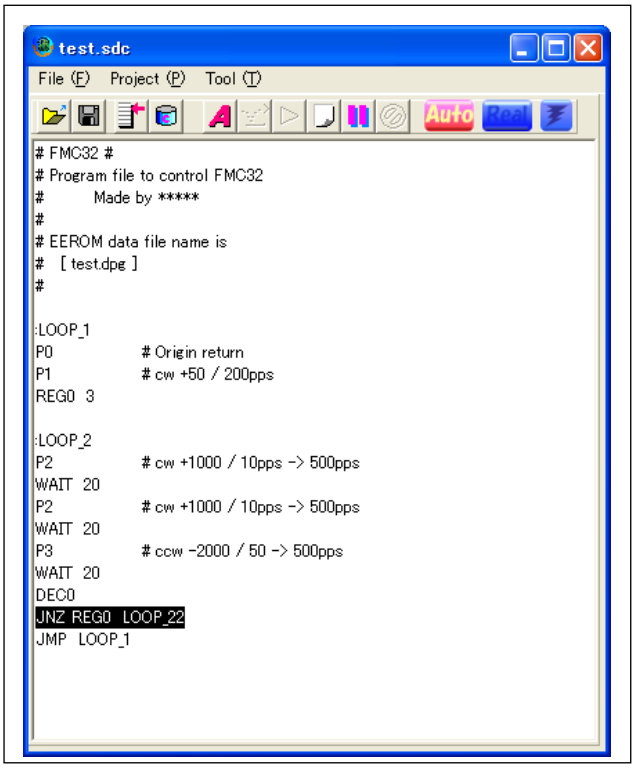


[In the case that there is an error to assemble]

If there is an error in the program, the following message appears



This message shows that the third character string has an error.
Click the "Close" button. The line with error is highlighted as follows.

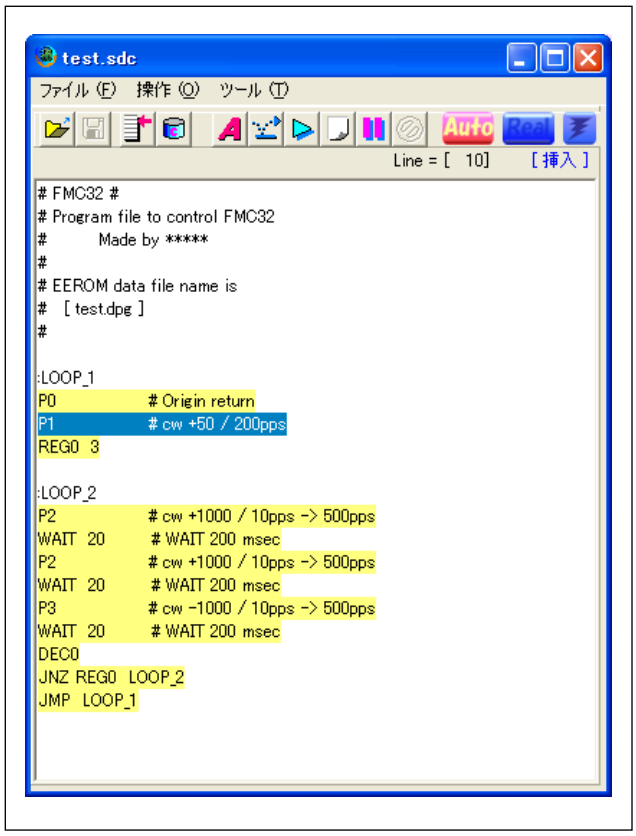


The third character string " JNZ REG0 LOOP_22" is wrong. The line should be "LOOP_22".
Like the above example, if the program has any errors, correct the errors by taking a cue from the part of highlighted display.

[Operation confirmation by step execution]

Click the following "Step execution" button.

The line that is highlighted inverted is processed and the highlighted line moves to the next program line to be processed.



Continue to click the "step execution" button and check whether program is arranged in the order you want.

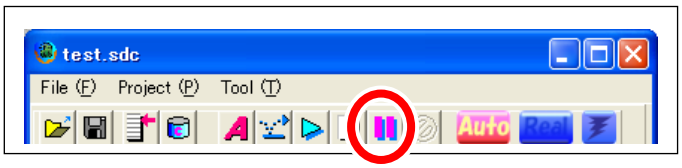
[Confirmation operation by continuous exaction]

Click the following "Run" button.

The program to be executed is highlighted in order.
The simulation is processed at the speed you can follow.
Waiting time of WAIT command is ignored at this time. You do not wait actually.

[Stop continuous execution]

Click the "Stop" button.



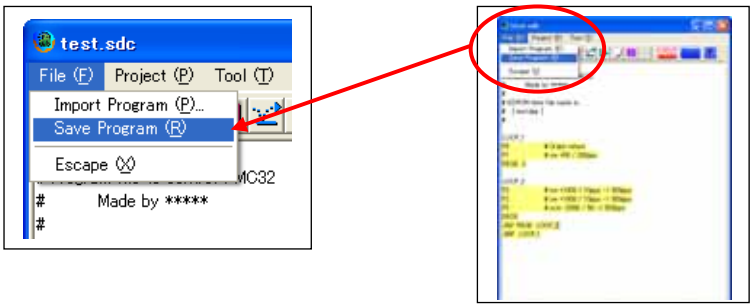
If you click "Run" button again, program starts to be processed from the location that the process stopped.
If you want to restart from the beginning, click the "Assemble" button again.

3-4-4. Save a program

Save the created program as a file.
If you exit program without saving data, the designed program is lost.
There are two kinds of data save procedures. Select one to be processed.

[Program save procedure 1]

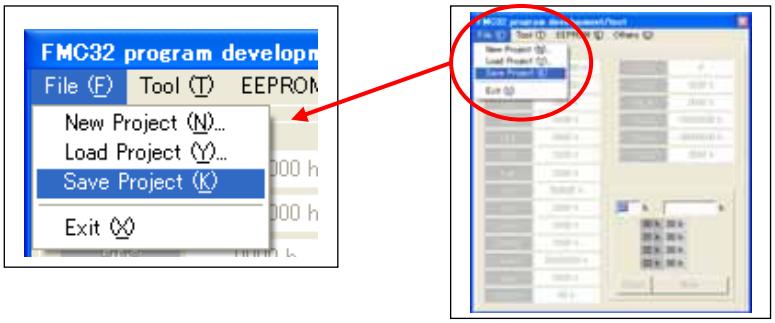
Select "Save Pattern data " from the "File" menu of the menu bar in the editor window.



Or you can save data by clicking the following save button.

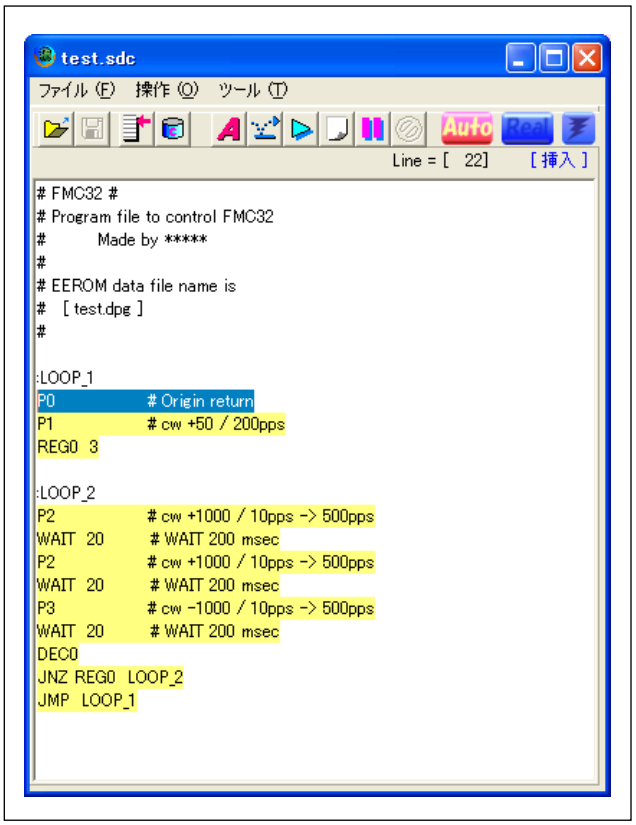
[Program save procedure 2]

Select "Save Project" from the "File" menu of the menu bar in the main window.



3-4-5. Confirm operation of program (Actual operation)

You can check the designed program operation while connecting a motor actually.
First, assemble the program and make the window as follows.



Click the following excitation switch button.

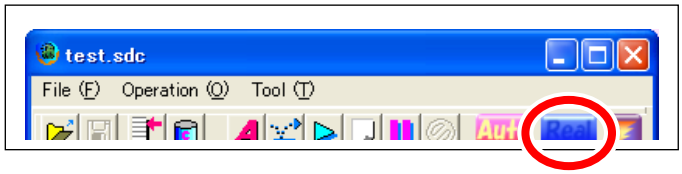
After switching the excitation condition, the button changed as follows.

← Shows that excitation is on.

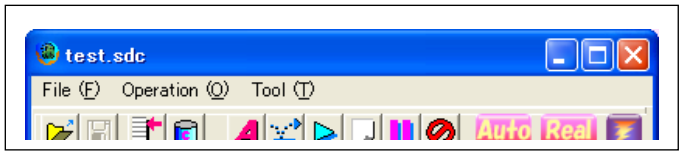
If the excitation switch button is not changed, the message: “Excitation is not changed" appears.
In this case, FMC32 is not detected as connected. Check the condition of the connection to the FMC32 and the condition of the power supply.

Caution: In this start-up guide, a button is used for switching excitation. Actually, you can use an excitation switch command and change the status automatically. The "Ex_ON" command can make excitation on and the "Ex_OFF" command can make it off.

After the excitation switch button is changed, click the following "REAL" (Switch to a real operation mode) button.



If the REAL button change as follows, the mode is the actual operation mode.

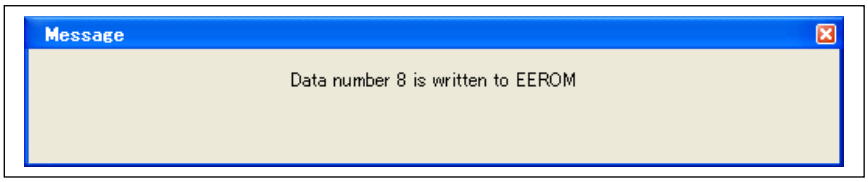


Subsequent procedure is the same as operation confirmation by simulation.

3-4-6. Write data to EEPROM

Write designed operation pattern and the program using the operation pattern to EEPROM on FMC32.
Select "Write to EEPROM" from the EEPROM menu on the Main window.

If an operation pattern is being written, the following message appears.
It is not written to the operation number that operation pattern is not registered. Comments are also not written.



If writing operation pattern is completed normally, writing program starts.
All writing is completed, the following message appears.

Caution: To Write created program, assemble it before writing program.
Check that the program does not have any errors by assembling it.
If you write unassembled program, the following error message appears and the program is not written. (Operation pattern is written.)
Unsaved program is not written. Please save it before completing this step.

Message example
Please assemble the program.
The program has been changed. Please assemble it.
Please save the program.
Please save the data of pattern.

3-4-7. Operation in stand-alone mode

First, unplug a USB cable and power off the FMC32 and origin switch. Separate the four-wire serial communication cable from the PUSB-3503 to the FMC32.
Finally, turn the origin switch on and power on the FMC32.
After powering on, the FMC32 operates in stand-alone mode according to the written data.
According to each dip switch of "STA external input" and "MON external input", there are cases that FMC32 does not operate.
FMC32 should operate if "STA external input" is on and "MON external input" is on.

CAUTION	The descriptions in this manual may be changed without prior notice to improve performance or quality.

NPM Nippon Pulse Motor Co., Ltd.

Head Office: No.16-13, 2-chome, Hongo, Bunkyo-ku, Tokyo, 113-0033, Japan
TEL: 81-3-3813-8841 FAX: 81-3-3813-8665
Web: http://www.pulsemotor.com E-mail: int-l@npm.co.jp

Issued in September 2011