YASKAWA

FS100 OPTIONS INSTRUCTIONS

User's Manual for New Language Environment MotoPlus

Upon receipt of the product and prior to initial operation, read these instructions thoroughly, and retain for future reference.

MOTOMAN INSTRUCTIONS

MOTOMAN-□□□ INSTRUCTIONS
FS100 INSTRUCTIONS
FS100 OPERATOR'S MANUAL
FS100 MAINTENANCE MANUAL

Please have the following information available when contacting Yaskawa Customer Support:
- System
- Primary Application
- Software Version *(Located on Programming Pendant by selecting: {Main Menu} - {System Info} - {Version})*
  Robot Serial Number *(Located on robot data plate)*
  Robot Sales Order Number *(Located on controller data plate)*

Part Number: 162002-1CD
Revision: 5

MANUAL NO.
HW1480837 ⑦

Yaskawa Customer Support

- 24-hour Telephone Number: **(937) 847-3200**
  Use for urgent or emergency needs for technical support, service and/or replacement parts
- Routine Technical Inquiries: *techsupport@motoman.com*
  Allow up to 36 hours for response

MANDATORY

- This manual explains MotoPlus of the FS100 system. Read this manual carefully and be sure to understand its contents before handling the FS100.
- General items related to safety are listed in the Chapter 1: Safety of the FS100 Instructions. To ensure correct and safe operation, carefully read the FS100 Instructions before reading this manual.

CAUTION

- Some drawings in this manual are shown with the protective covers or shields removed for clarity. Be sure all covers and shields are replaced before operating this product.
- The drawings and photos in this manual are representative examples and differences may exist between them and the delivered product.
- YASKAWA may modify this model without notice when necessary due to product improvements, modifications, or changes in specifications. If such modification is made, the manual number will also be revised.
- If your copy of the manual is damaged or lost, contact a YASKAWA representative to order a new copy. The representatives are listed on the back cover. Be sure to tell the representative the manual number listed on the front cover.
- YASKAWA is not responsible for incidents arising from unauthorized modification of its products. Unauthorized modification voids your product's warranty.

NOTES FOR SAFE OPERATION

Read this manual carefully before installation, operation, maintenance, or inspection of the FS100.

In this manual, the Notes for Safe Operation are classified as “DANGER”, “WARNING”, “CAUTION”, “MANDATORY”, or “PROHIBITED”.

 DANGER	Indicates an imminent hazardous situation which, if not avoided, could result in death or serious injury to personnel.
 WARNING	Indicates a potentially hazardous situation which, if not avoided, could result in death or serious injury to personnel.
 CAUTION	Indicates a potentially hazardous situation which, if not avoided, could result in minor or moderate injury to personnel and damage to equipment. It may also be used to alert against unsafe practices.
 MANDATORY	Always be sure to follow explicitly the items listed under this heading.
 PROHIBITED	Must never be performed.

Even items described as “CAUTION” may result in a serious accident in some situations. At any rate, be sure to follow these important items.



To ensure safe and efficient operation at all times, be sure to follow all instructions, even if not designated as “DANGER”, “WARNING” and “CAUTION”.

WARNING

- Before operating the manipulator, check that servo power is turned off when the emergency stop button on the programing pendant is pressed.
  When the servo power is turned off, the SERVO ON LED on the programing pendant is turned off.

Injury or damage to machinery may result if the emergency stop circuit cannot stop the manipulator during an emergency. The manipulator should not be used if the emergency stop buttons do not function.

Fig. : Emergency Stop Button



- In the case of not using the programming pendant, be sure to supply the emergency stop button on the equipment. Then before operating the manipulator, check to be sure that the servo power is turned OFF by pressing the emergency stop button.
  Connect the external emergency stop button to the 5-6 pin and 16-17 pin of the robot system signal connector (CN2).

- Upon shipment of the FS100, this signal is connected by a jumper cable in the dummy connector. To use the signal, make sure to prepare a new connector, and then input it.

If the signal is input with the jumper cable connected, it does not function, which may result in personal injury or equipment damage.

- Once the emergency stop button is released, clear the cell of all items which could interfere with the operation of the manipulator. Then turn the servo power ON.

Injury may result from unintentional or unexpected manipulator motion.

Fig. : Release of EM



- Observe the following precautions when performing teaching operations within the P-point maximum envelope of the manipulator:
  - Be sure to use a lockout device to the safeguarding when going inside. Also, display the sign that the operation is being performed inside the safeguarding and make sure no one closes the safeguarding.
  - View the manipulator from the front whenever possible.
  - Always follow the predetermined operating procedure.
  - Ensure that you have a safe place to retreat in case of emergency.

Improper or unintended manipulator operation may result in injury.

WARNING

- Confirm that no person is present in the P-point maximum envelope of the manipulator and that you are in a safe location before:
  - Turning on the power for the FS100.
  - Moving the manipulator with the programming pendant.
  - Running the system in the check mode.
  - Performing automatic operations.

Injury may result if anyone enters the working envelope of the manipulator during operation. Always press an emergency stop button immediately if there are problems.
The emergency stop button is located on the programing pendant.

CAUTION

- Perform the following inspection procedures prior to conducting manipulator teaching. If problems are found, repair them immediately, and be sure that all other necessary processing has been performed.
  - Check for problems in manipulator movement.
  - Check for damage to insulation and sheathing of external wires.
- Always return the programming pendant to the hook on the FS100 cabinet after use.

The programming pendant can be damaged if it is left in the manipulator's work area, on the floor, or near fixtures.

- Read and understand the Explanation of the Warning Labels in the FS100 Instructions before operating the manipulator.

Definition of Terms Used Often in This Manual

The MOTOMAN is the YASKAWA industrial robot product.

The MOTOMAN usually consists of the manipulator, the FS100 controller, manipulator cables, the FS100 programming pendant (optional), and the FS100 programming pendant dummy connector (optional).

In this manual, the equipment is designated as follows:

Equipment	Manual Designation
FS100 controller	FS100
FS100 programming pendant	Programming pendant
Cable between the manipulator and the controller	Manipulator Cable
FS100 programming pendant dummy connector	Programming pendant dummy connector

Descriptions of the programming pendant keys, buttons, displays and keyboard of the PC are shown as follows:

Equipment		Manual Designation
Programming Pendant	Character Keys	The keys which have characters printed on them are denoted with [ ]. e.g. [ENTER]
	Symbol Keys	The keys which have a symbol printed on them are not denoted with [ ] but depicted with a small picture. e.g. PAGE key  The cursor key is an exception, and a picture is not shown.
	Axis Keys Numeric Keys	“Axis keys” and “Numeric keys” are generic names for the keys for axis operation and number input.
	Keys Pressed Simultaneously	When two keys are to be pressed simultaneously, the keys are shown with a “+” sign between them. e.g. SHIFT key  + COORD key 
	Mode Key	Three kinds of modes that can be selected by the mode key are denoted as follows: REMOTE, PLAY, or TEACH
	Button	Three buttons on the upper side of the programming pendant are denoted as follows: HOLD button START button EMERGENCY STOP button
	Displays	The menu displayed in the programming pendant is denoted with { }. e.g. {JOB}
PC Keyboard		The name of the key is denoted. e.g. Ctrl key on the keyboard

Description of the Operation Procedure

In the explanation of the operation procedure, the expression “Select • • • ” means that the cursor is moved to the object item and the SELECT key is pressed.

Registered Trademark

In this manual, names of companies, corporations, or products are trademarks, registered trademarks, or brand names for each company or corporation. The indications of (R) and TM are omitted.

1 Outline 1-1

1.1 Introduction 1-1

1.2 Features 1-1

1.3 What MotoPlus Can Do 1-2

1.4 Supported Services 1-3

1.5 Application Program Development Environment 1-5

1.5.1 Development Environment Configuration 1-5

1.5.2 Development Environment Component 1-6

1.5.3 Application Program Execution Environment 1-7

1.5.4 Development Flow 1-8

1.6 Notes 1-9

2 Development Environment Setup 2-1

2.1 DVD Contents 2-1

2.2 Installation of Development Environment Software 2-2

2.3 Installation of TELNET Connection Utility 2-5

3 Creating Application Software 3-1

3.1 Rules for Creating Program Source Code 3-1

3.2 Creating MotoPlus Project 3-2

3.3 Outline of MotoPlus IDE 3-3

3.4 Text Search 3-4

3.5 MotoPlus Library Files 3-4

3.6 Program Build (Compile/Link) 3-4

3.7 Changing Build Settings 3-5

4 Valid/Invalid Setting of MotoPlus Function 4-1

5 Installation and Start-up of Application Program 5-1

5.1 Installation of Application Program in Maintenance Mode 5-1

5.2 Start-up of Application Program 5-4

5.3 To Temporarily Prevent MotoPlus Application from Starting 5-5

6 User Defined File 6-1

6.1 Saving in and Reading from External Memory 6-1

6.1.1 Procedure to Save or Read User Defined File 6-1

6.2 Initialization of User Defined File 6-3

6.3 Setting of File Control Task Priority Limit 6-5

6.3.1 Changing File Control Task Priority Limit 6-5

7 Debugging 7-1

7.1 Debugging Environment 7-1

7.2 Connecting FS100 with PC 7-2

7.3 Debugging via Telnet Communication 7-3

7.3.1 Operation Procedure 7-3

7.3.2 Example Usage of printf() 7-6

7.4 Debugging by User Variables of FS100 7-7

8 Alarm List 8-1

1 Outline

1.1 Introduction

This function is used to develop customer specific applications that operate on the main CPU board (CPU201R board) of the FS100. The function name MotoPlus is an abbreviation of **Moto**man Professional **P**rogramming **L**ang**u**age for **S**uperior Use. MotoPlus is a programming IDE (integrated development environment) for professionals, with which the application program developed on the PC using C language runs as a task of the robot controller. Users can develop a variety of application programs on their own without modifying the source codes of the robot controller.

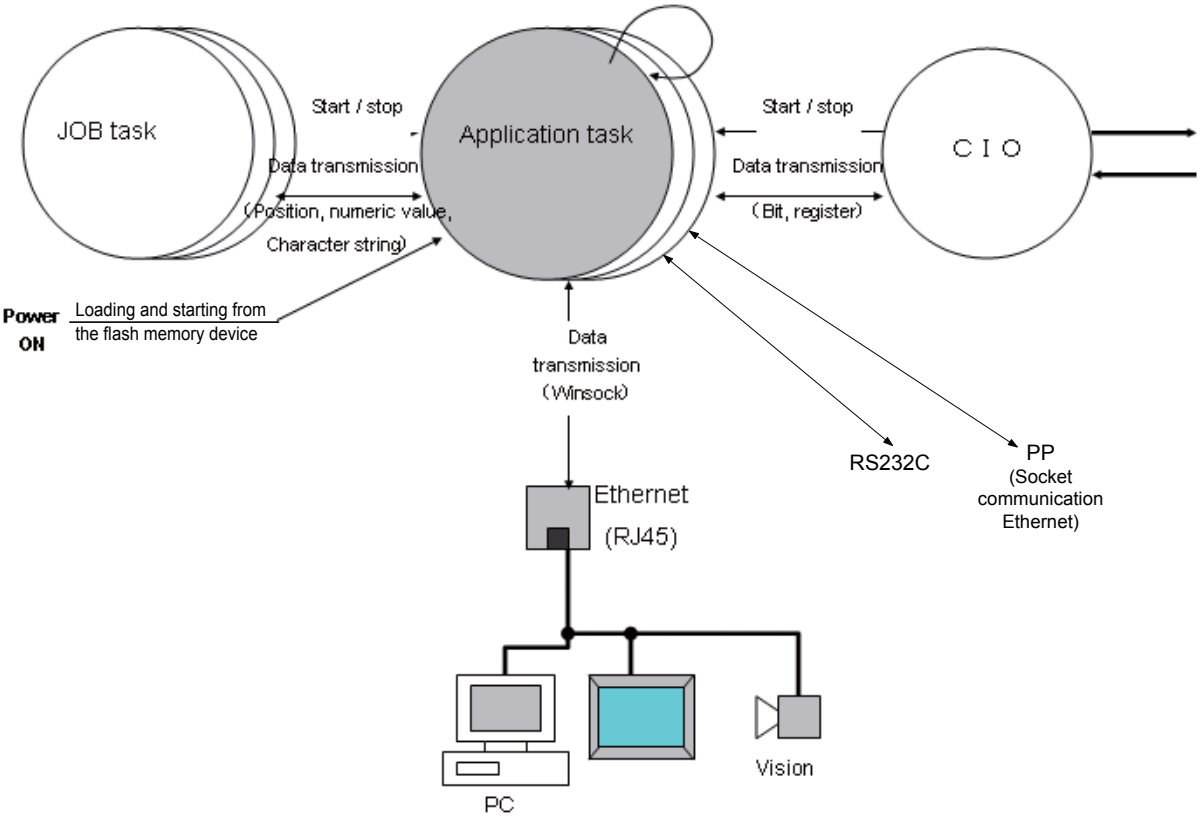
1.2 Features

1. With C language as the system development language and a wealth of standard libraries, users can create customer specific application programs.
2. With the text editor of MotoPlus IDE, users can program off-line.
3. The created application program runs as a task on the main CPU of the robot controller. Thus, no additional hardware is required.
4. The created application program runs as the native code of the CPU. Thus, the execution speed becomes faster.
5. With a variety of included APIs (application program interface) to transmit data from/to the job, the Ethernet communication port, and the RS232C serial communication port, etc., users can easily correct the manipulator position and connect the robot to the external PC or sensors.

1.3 What MotoPlus Can Do

Users can use the C language, a general programming language, and the C language and MOTOMAN libraries to develop a customer specific application program in which various types of resources are used, e.g., the Ethernet communication port, the RS232C serial communication port, and the programming pendant of the robot controller. Refer to *fig. 1-1 "Available Resources for MotoPlus"* .

Fig. 1-1: Available Resources for MotoPlus



Application program examples:

1. Operation to correct the manipulator position by a communication program with a vision sensor or other sensors, and by data from the sensor
2. Data transmission with an external PC via the Ethernet (TCP/IP) communication
3. JOB execution sequence control

1.4 Supported Services

MotoPlus provides users with a variety of services as follows:

1. Application task control
   Application task start/stop, data transmission between application programs by the mailbox and semaphore, exclusive control and execution synchronization
2. Robot control
   Robot control from the application program
3. JOB control
   Start-stop control of the JOB by the application, data transmission between Jobs by variables (byte, integer, double-precision, floating-point, character string, and position variable), and execution synchronization
4. CIO control
   I/O between the application program and the CIO, and read/ write of the register
5. Ethernet communication control
   The Ethernet (TCP/IP) communication from the application, especially using the socket function library included as an API
6. Programming pendant communication
   Data transmission between the application and the programming pendant application
7. EVENT
   Event notification to the application at every I/O control cycle and interpolation control cycle of the system
8. RS232C serial communication control
   RS232C serial communication from the application
9. Sensor control
   The APIs which transfer data to and from a job and change operating conditions such as path correction and speed change, and the instructions which transfer data between a job and MotoPlus application
10. Memory management
    The protected memory management
    Specific instructions, malloc and mfree, support the data area which the system manages for MotoPlus.
11. General-purpose file control
    The function to access multiple general-purpose files by using the fixed area on CMOS as a drive
12. Existing file control
    The function to access existing files (jobs and condition files, etc. which can be loaded and saved by external memory)
13. Servo control
    Servo control from the application program
14. User watchdog
    Watchdog to monitor whether the application operates normally

15. Coordinate conversion
    Calculation of the manipulator order or inverse kinematics, coordinate conversion, pulse conversion from the feedback pulse to the arithmetic pulse, or linear algebra calculation.
16. Force control
    Retrieves information about the force control from the applications and operates the force feedback control.
17. Development environment
    (1) Integrated development environment (MotoPlus IDE) for source file project management, program editor, and compiler and linker.
    (2) Installation of the MotoPlus application from the CF/USB memory in the programming pendant
    (3) Debugging by printf() via Telnet

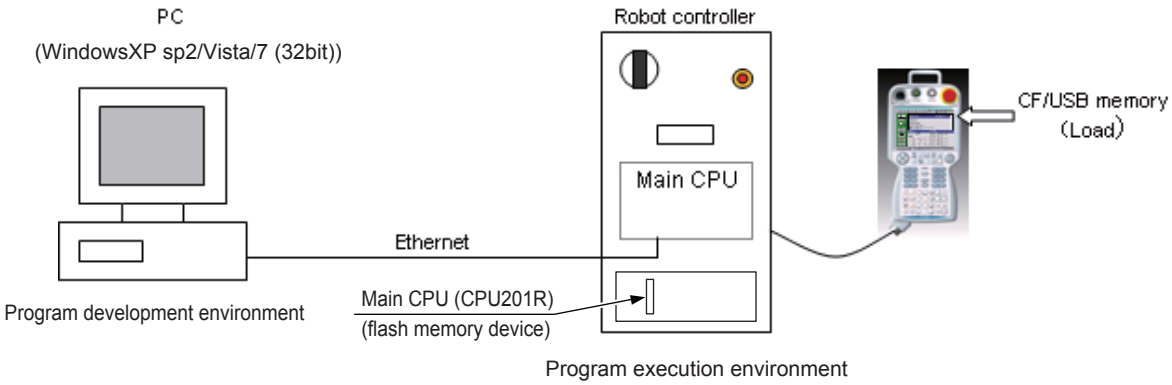
1.5 Application Program Development Environment

A MotoPlus application program is written in C language on a PC and debugged on the FS100 controller. Use the text editor of MotoPlus IDE to write the application program, then compile and link it to create the execution object. This is installed and executed on the FS100 to debug.

1.5.1 Development Environment Configuration

Use a PC to develop an application program. The system is configured as shown in *fig. 1-2 "Development Environment Configuration"* to debug using Telnet. Install the developed application by loading it to the CF of the main CPU (CPU201R) using the flash memory device in the programming pendant.

Fig. 1-2: Development Environment Configuration



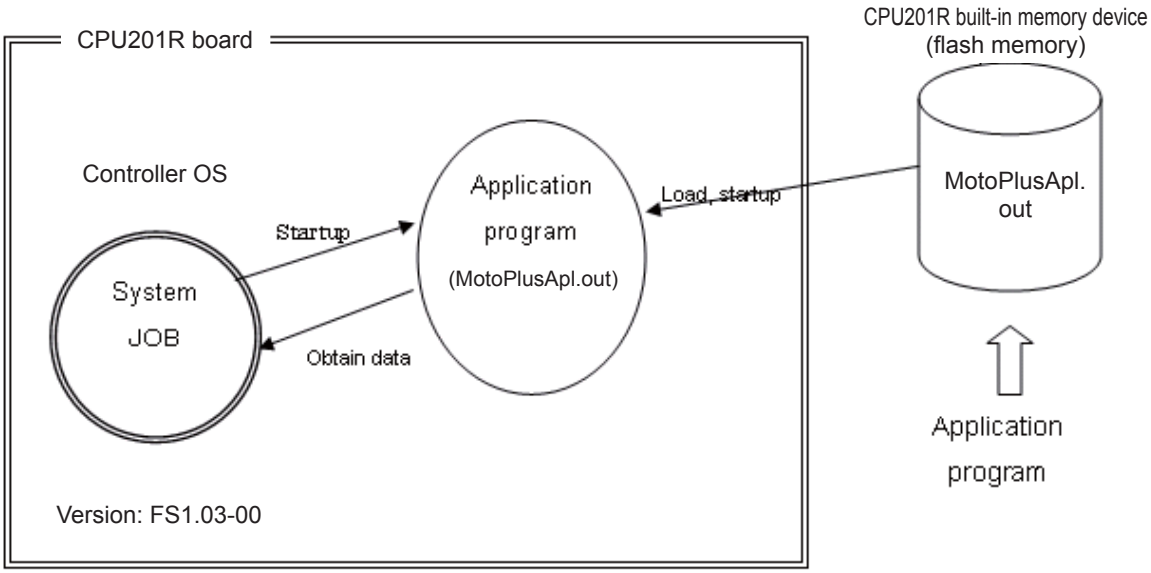
1.5.2 Development Environment Component

No.	Device	Details	Remarks
1	Windows PC	CPU: 1GHz or more Memory: 1 Gbyte or more Hard disk space: 40 Gbyte or more OS: Windows XP Service Pack 2, Windows Vista, Windows 7 (32bit)	Prepared by the user
2	Text editor	Creates the application source program in C language. MotoPlus IDE editor is recommended. (A commercially-available text editor and Windows Notepad can also be used.)	Prepared by the user
3	Compiler	Converts the source program into the machine language of the target CPU. Integrated in MotoPlus IDE, and can be executed from the menu.	Provided by Yaskawa
4	Linker/library	Links the C language library provided by GNU with the developed application program to create an executable load module. Integrated in MotoPlus IDE, and can be executed from the menu.	Provided by Yaskawa
5	MotoPlus library	API library which provides application program services of the robot controller such as data communication with JOB. Integrated in MotoPlus IDE.	Provided by Yaskawa
6	CF/USB memory	Compact Flash memory or USB memory to load the developed load module into the controller.	Prepared by the user
7	Ethernet cable	Connects the PC with the controller via Ethernet to debug the developed program. CAT5 or greater (cross/straight) cable	Prepared by the user
8	Ethernet hub	Connects the PC with the controller.	Prepared by the user
9	Install DVD	Installs the above-mentioned software on the Windows PC.	Provided by Yaskawa
10	Force sensor	It is necessary in order to use the force control. Cannot use the force sensor other than the provided by YASKAWA.	Provided by Yaskawa
11	High speed serial communication board	It is necessary in order to use the force control. Cannot use the communication board other than the provided by YASKAWA.	Provided by Yaskawa

1.5.3 Application Program Execution Environment

The application program developed in the above-mentioned program development environment is stored in the flash memory device on the main CPU (CPU201R) of the robot controller, as shown in *fig. 1-3 "Program Execution Environment"* . No special hardware or software is required for installation other than using the force control, but the system software version must be FS1.03-00 or later.

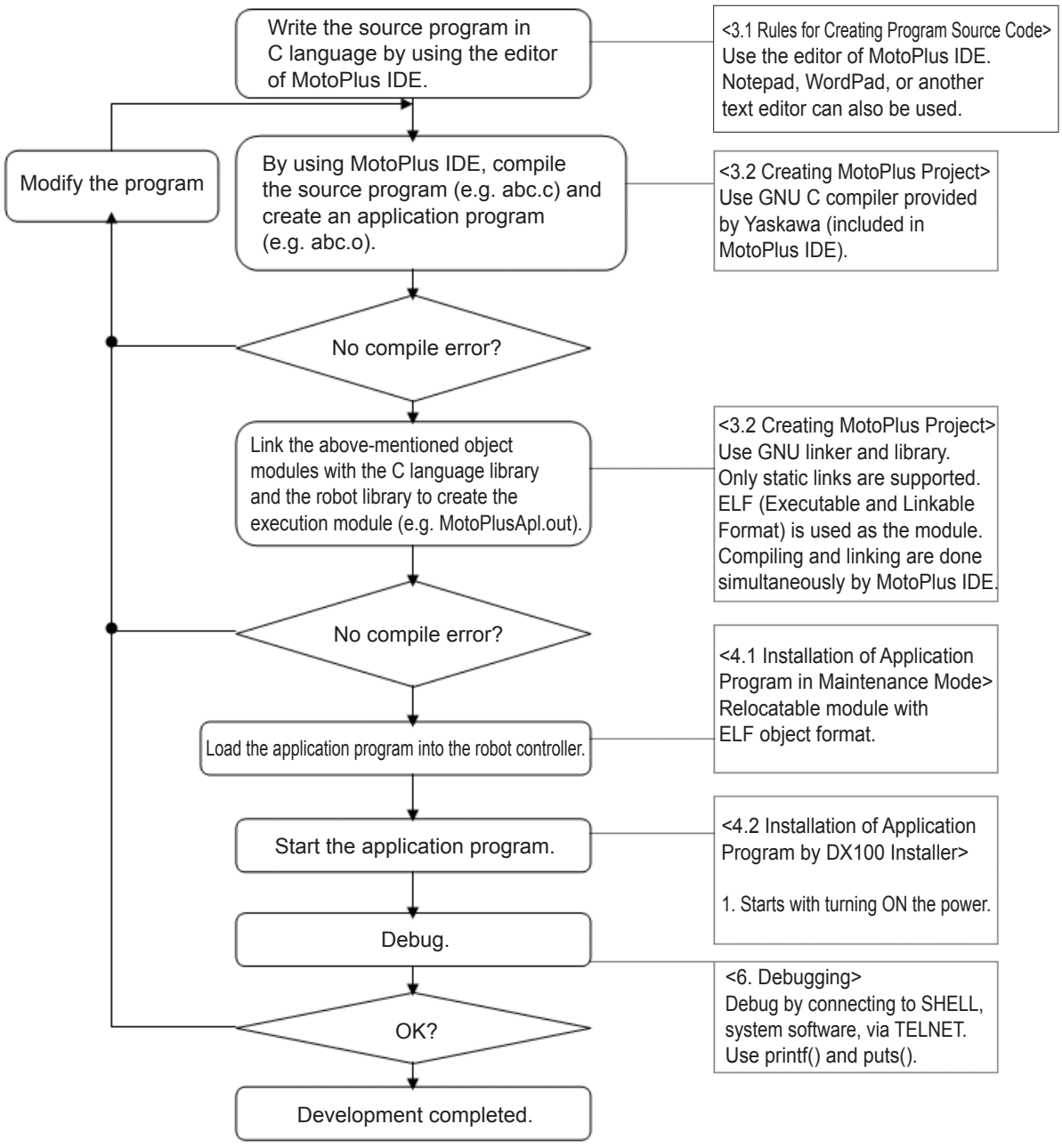
Fig. 1-3: Program Execution Environment



When the controller power is turned ON, the application program MotoPlusApl stored in the flash memory device on the main CPU (CPU201R), is loaded into the memory of the CPU201R board. Then, the application program is started as a task on the real-time operating system controller OS, and executed synchronously with the system program. The application task can start multiple tasks, and can perform data transmission with JOB or I/O, synchronous execution control, and exclusive control, by using the mailbox or semaphore.

1.5.4 Development Flow

Fig. 1-4: Program Development Flow



1.6 Notes

NOTE

- C language memory operation
  When programming in C language, the pointer variable which stores the memory address can be used. If the pointer variable is set incorrectly, the system memory area may be rewritten.
  If the system memory area is rewritten, critical problems occur, e.g., the software hangs up (the hang-up status is detected by a CPU exception or the watchdog check function, then the servo power turns OFF and the system shuts down), the robot stops its operation due to an alarm, or the programming pendant becomes inoperable.
  If the system memory operation is incorrect, the application itself may not operate as it is designed. Thus, by testing adequately, the error location can be found. Check the functions of the application software well enough, and make sure that all functions operate as designed.
- Task “mpUsrRoot” which starts when the power is turned ON
  The task “mpUsrRoot” has a high priority to start other application tasks and initialize the entire application quickly. Due to its high task priority, if a time-consuming process (100 microseconds or longer) is done in this task, the processing time for the robot control becomes insufficient. Then the system alarm may occur or the programming pendant or the HOLD button may freeze. Thus, as the sample program (refer to “13.3 Task Control Sample Program” in “Programmer’s Manual for New Language Environment MotoPlus (HW1480839)”, make sure to complete the task “mpUsrRoot” after starting up another application or creating a semaphore.
- Debugging via Telnet

1. With the Telnet function, debug only the tasks with normal priority. For debugging the tasks with high priority (the tasks started with the task priority: MP_PRI_IO_CLK_TAKE, MP_PRI_IP_CLK_TAKE, or MP_TIME_CRITICAL, or the task “mpUsrRoot”), use the variables of the FS100. This is because, if printf() or puts() is done with a high priority task, the processing time for the robot operation control is consumed, and a system alarm may occur due to insufficient processing time.
2. After completing debugging, make sure to clear the debugging code using printf() or puts() which is described in the source code. This is because the processing time of printf() and puts() influences the system, and an error may occur such as insufficient processing time during robot operation.
3. When using the SHELL debugging function of the controller OS which becomes available by connecting with the FS100 via Telnet, a system alarm may occur due to insufficient processing time for robot operation. Do NOT use the SHELL debugging function.

2 Development Environment Setup

2.1 DVD Contents

The DVD contains the following items.

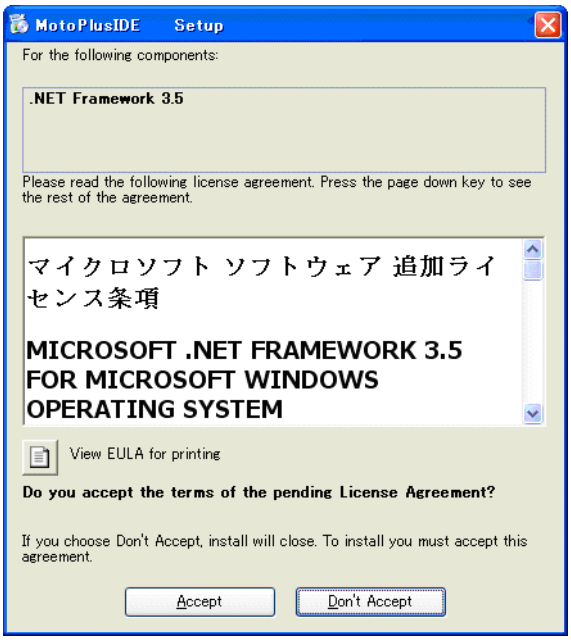
Folder		Contents	Remarks
Root		MotoPlus IDE installer	
	Manual (Japanese/ English)	MotoPlus user's manual	MotoPlus outline, installation procedure of the development environment, and development procedure
		MotoPlus programmer's manual	Programming procedure
		MotoPlus reference manual	API specifications
		GCC manual	
	Sample	Sample source code	
	.NET Framework	Installer of .NET Framework version3.5 Service Pack 1	
	Sentinel Driver	Installer of the system driver for the hardware key	

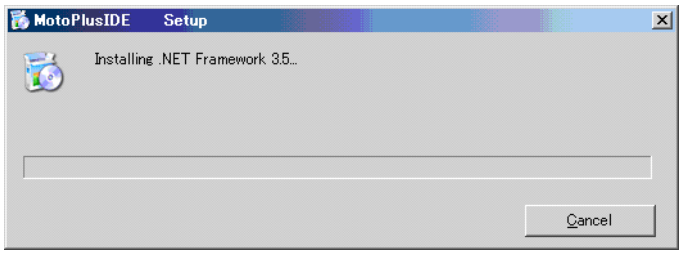
2.2 Installation of Development Environment Software

Install MotoPlusIDE as indicated in the following procedure.

1. Insert the MotoPlusIDE installation DVD in the DVD slot on the PC.
   – If AutoRun is enabled, the installation starts.
   If not, run “setup.exe” in the DVD.
2. Confirm .NET Framework 3.5 Service Pack1 (SP1)
   – If the .NET Framework 3.5 SP1 is not installed on the PC, the following window appears and the installation starts.

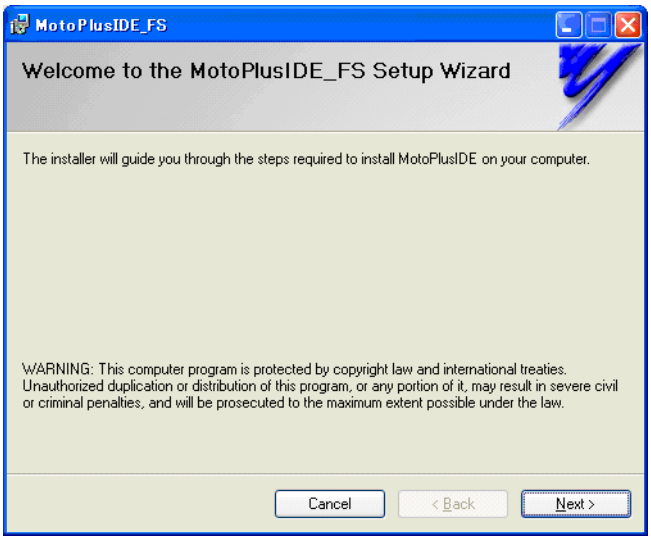
NOTE To install .NET Framework 3.5 SP1, PC must be connected to the Internet.





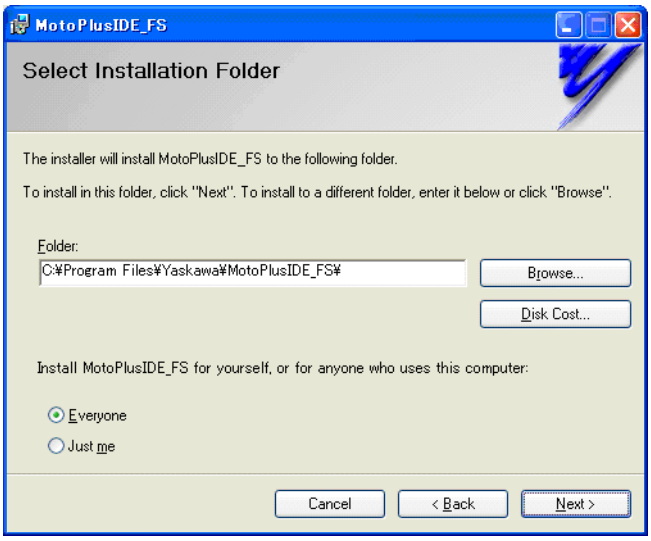
3. Install MotoPlusIDE

– After the installation of .NET Framework 3.5 SP1 is completed (or if it is already installed), the following MotoPlusIDE installation window appears.

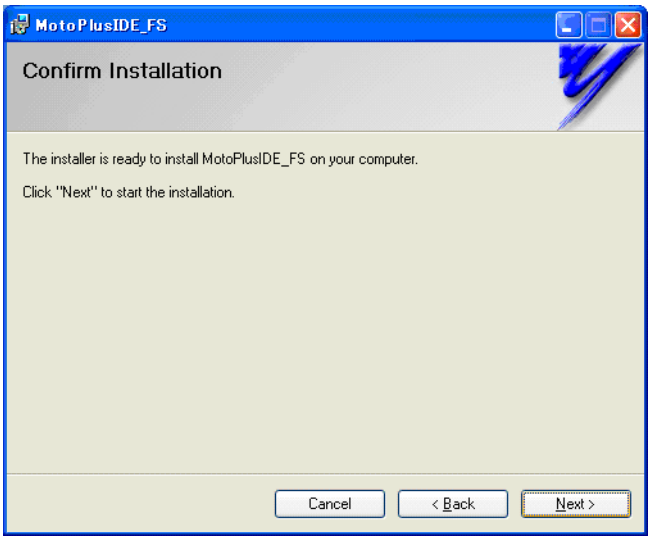


4. Click [Next>]

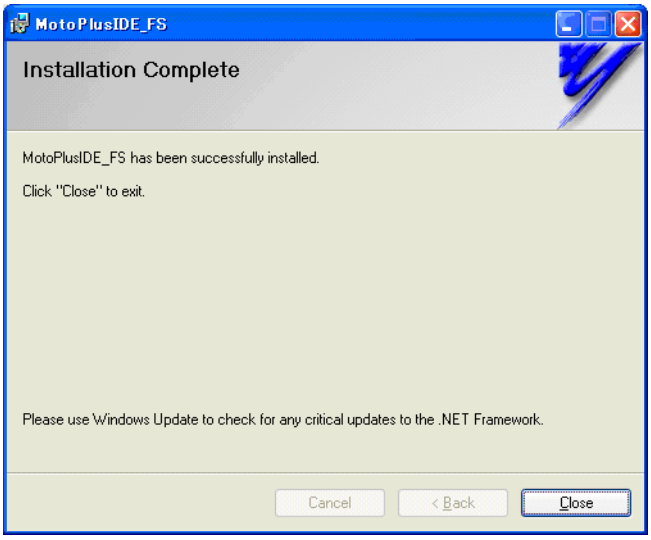
– The following window appears to specify the installation destination folder.
The default installation folder:
“C:\Program Files\Yaskawa\MotoPlusIDE_FS”



5. Click [Next>]
   – The following confirmation window for installation appears.



6. Click[Next>]
   – The progress of the installation appears.
   When the installation is completed properly, the following window appears.
   Upon completing the installation, shortcuts of MotoPlusIDE are added on the desktop and in the Programs folder of the Start menu.

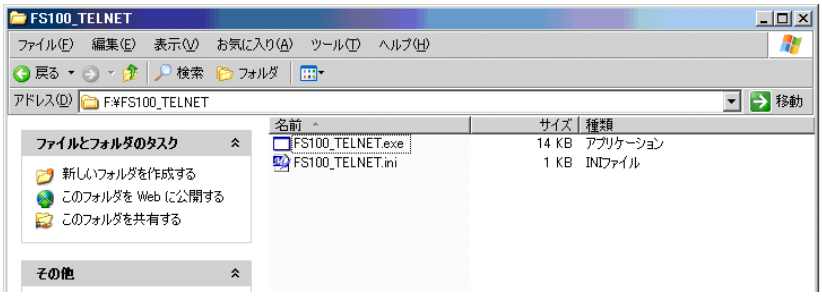


7. After completing the installation, restart the PC
   – Shutdown the PC and restart it.
8. Install the system driver for the hardware key
   – After restarting the PC, start “SentinelSystemDriverInstaller7.5.7.exe” in the “SentinelDriver” folder in the installation DVD.
9. Install the hardware key
   – Insert the provided hardware key into any available USB port.
   The message “Your new hardware is installed and ready to use.” appears in the taskbar.

2.3 Installation of TELNET Connection Utility

To debug the MotoPlus application, connect the FS100 and the PC via TELNET and describe using printf() in the application program. Thus, the TELNET connection utility is prepared. Install it by following the procedure below.

1. Copy the entire “FS100_TELNET” folder in the MotoPlus installation DVD into an appropriate hard disk of the PC. (“FS100_TELNET.exe” and “FS100_TELNET.ini” must be in the same folder.) Run “FS100_TELNET.EXE” to connect to the FS100 via TELNET.



2. Also, configure the settings of TELNET connection by “FS100_TELNET.ini”. Set the IP address of the FS100 here.
   ipaddress: IP address of the FS100
   (Set the same IP address as the one set in the HOST SETUP window.)
   sendtime: Waiting time from the startup of “FS100_TELNET.EXE” until starting the TELNET connection (default value: 8, unit: sec)
   inputTime: Waiting time from starting the TELNET connection until starting to login (default value: 4, unit: sec)

NOTE When the setup utility completes, you must also install the system driver for the hardware key. Start “Sentinel System Driver Installer 7.5.7.exe” to install the driver.

3. Insert the provided hardware key into any available USB port. The message “Your new hardware is installed and ready to use.” will appear in the taskbar.

3 Creating Application Software

3.1 Rules for Creating Program Source Code

Be sure to observe the following rules when creating the source code.

Refer to “Programmer’s Manual for New Language Environment MotoPlus (HW1480839)” for detailed information on programming configuration, files to be included, and sample programs.

Please remember that the source code should follow ANSI-C syntax. The only exception is that users should not define a “main (…)” function. This is replaced by “mpUsrRoot(…)”.

1. Up to 32 one-byte characters can be used for a file name. Up to 16 two-byte characters can also be used. File names are case-sensitive. Set “.c” as the extension.
2. Put “#include<MotoPlus.h>” at the top of each source code file. When “MotoPlus.h” is included, the information of all services provided by MotoPlus is included.
3. The application entry point is the function which is called from the system software when the FS100 is started. This function must be described as:
   void mpUsrRoot (int arg1, int arg2, int arg3, int arg4, int arg5, int arg6, int arg7, int arg8, int arg9, int arg10)
   Be sure to describe mpExitUserRoot at the end of the mpUserRoot function.

```
void mpUsrRoot (int arg1, int arg2, int arg3, int arg4, int arg5, int arg6,
int arg7, int arg8, int arg9, int arg10)
   {
      /* 10 parameters from arg1 to arg10*/
      .
      mpExitUsrRoot;// End of mpUsrRoot program (End of the task)
   }
```

e.g. filename: abc.c

* The original argument of arg1 to arg10 is mpUsrRoot. arg1 to arg10 are prepared to receive arguments in the future.
  At present, arg1 is the function name of the user entry, and arg2 to arg10 are reserved for the manufacturer (if referenced, all of them are “0”).



The task “mpUsrRoot” has a high priority to start other application tasks and initialize the entire application quickly. Due to its high task priority, if a time-consuming process (100 microseconds or longer) is done in this task, the processing time for the robot control becomes insufficient. Then the system alarm may occur or the programming pendant or the HOLD button may freeze. Thus, as the sample program (refer to “13.3 Task Control Sample Program” in “Programmer’s Manual for New Language Environment MotoPlus (HW1480839)”, make sure to complete the task “mpUsrRoot” after starting up another application or creating a semaphore.

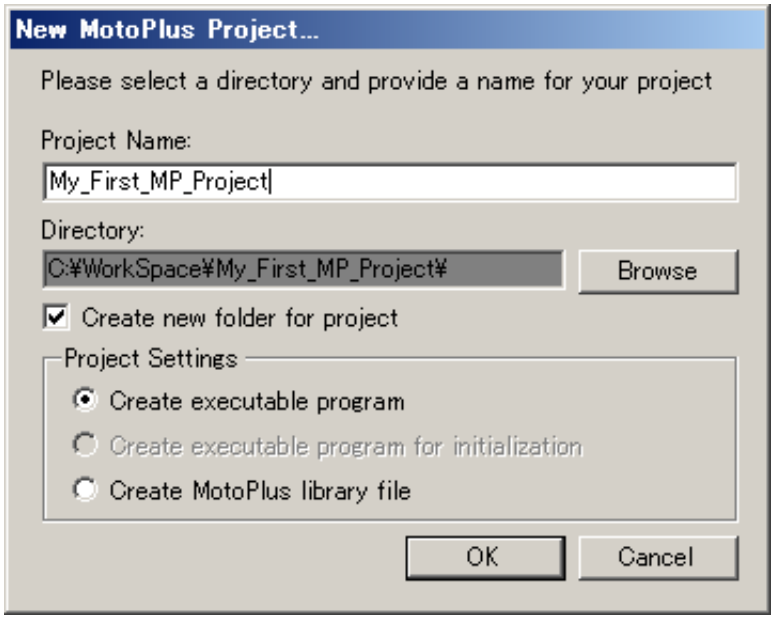
3.2 Creating MotoPlus Project

Start MotoPlus IDE from the Windows Start menu. The window of MotoPlus IDE appears.

To create a MotoPlus application, it is necessary to create a project. The project is the unit to manage the source files for each application program.

All of the source files required for the application are included in the project. Double-click an existing project in Windows Explorer, then MotoPlus IDE automatically starts and the project is opened.

To create a new project, go to the top menu bar, then click File > New > Project. Then, the following window appears.



Type a project name (up to 32 single-byte characters which can be used as a file name), then a folder to store the source file will be automatically created (the folder can be located anywhere in the PC). Select the type of the MotoPlus file to be created from “Project Settings”.

There are following options for “Project Settings”:

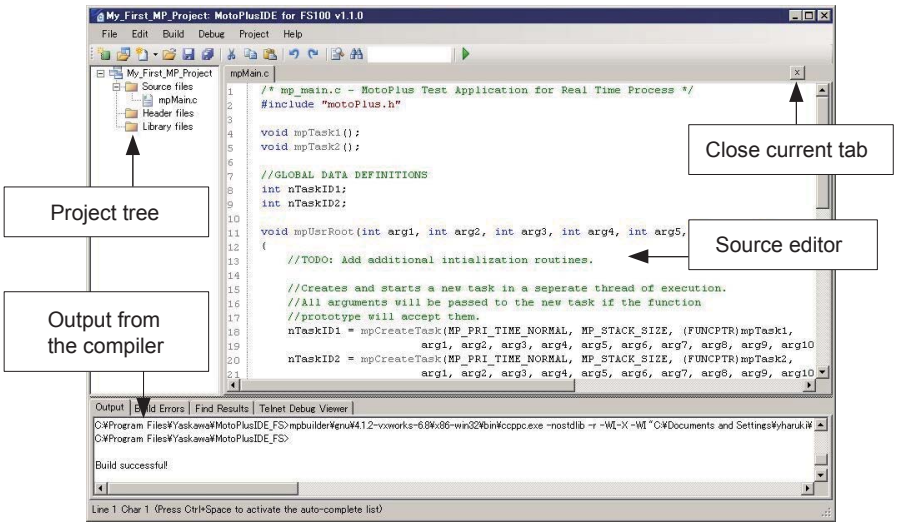
1. Executable program (default): Creates an *.out file. This is a normal MotoPlus application which can be executed as a task of the FS100.
2. MotoPlus library file: Creates a static library file (*.fsLib). This type of library cannot be executed on the FS100 but can be called from MotoPlus application. Thus, import it from another project file to use it.

When a new project is created, the source code template “mpMain.c” opens. Use it if necessary. Delete it if not necessary.

3.3 Outline of MotoPlus IDE

The left pane of MotoPlus IDE contains the project tree. This lists all source files, header files, and library files that will be compiled when the project is built. To edit a file that is in the project tree, double-click the filename, then the contents of the file appears as a new tab in the source editor window. See *fig. 3-1 "Main interface for MotoPlus IDE"* .

Fig. 3-1: Main interface for MotoPlus IDE



Right-click on the project tree to show the following options:

1. Remove a file from the project
2. Create sub-folders to organize files (click and drag files to move them to a new folder)
3. Add files to the project (new or existing)
4. Add MotoPlus libraries to the project
5. Compile the code and build your program

The source editor has the following features:

- Automatic syntax highlighting of keywords
- Automatic indenting
- Undo/redo up to 50 actions
- Tab for each open document for easy navigation between files
- Right-click menu to instantly open the header file that has been included
- Search for text in the entire project

3.4 Text Search

To search for text, go to the top menu bar and click Edit > Find and Replace (shortcut is Ctrl + f). Then, the “Find” dialog appears. When “Find All” is selected, the cursor will not immediately go to the next searched text. Instead, the bottom pane will change to the “Find Results” tab to show the search results. Double-click a result in this pane to go to the corresponding source code tab and move the cursor to the searched text.

3.5 MotoPlus Library Files

A MotoPlus library file (*.fsLib) is a file containing the source code which is already compiled. After the source code is compiled to .fsLib format, it cannot be edited or viewed. (The library can be distributed and used without distributing the source code.)

To create a library file, select “Create MotoPlus Library file” when creating a new project. This library can be used by other projects, but in this case, prepare an appropriate header file (*.h) in advance. Also, DO NOT use mpUsrRoot() function for the library.

To use the library file in an executable project, go to the top menu bar and click Project > Add MotoPlus library file. Then you can locate an .mpLib file on your computer. When you select the file, it will be copied to the current project folder along with the corresponding header file. If the header is not present (or is named differently), it will not be copied. In this case, you must refer to the library documentation provided by the creator of the library. When you build your project, the compiled code from the library will be added to the final executable object (.out).

3.6 Program Build (Compile/Link)

To compile the created source code and build either an executable object (*.out) or a library file (*.fsLib), go to the top menu bar and click Build > Build Project. This will save any changes made to the project.

The bottom pane will automatically switch to the “Output” tab during a build attempt. The text box in this window will log the build process. The “Build Errors” tab will display any errors or warnings given by the compiler. If there is a fault in the source code, it will indicate the line number where it encountered the error. To quickly go to the appropriate file and line number, just double-click the error message. Please note that due to the structure of C language, an error on any line could be caused by a fault in any line above the error. If there is a fault in an include file, the error message may not even indicate the correct file.

If the Output window indicates the build was successful, you will be able to locate your .out file in a folder labeled “output”. This folder will be in the directory of your project file. If your project is set to build a library file, the output folder will contain the .fsLib file and the .h file.

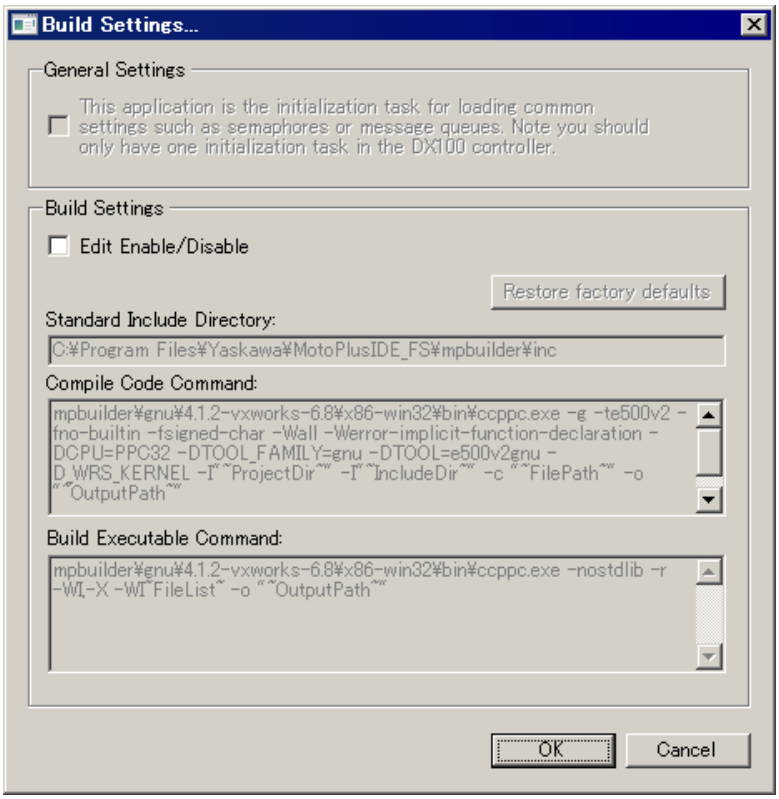
3.7 Changing Build Settings

To change build settings, select “Build”, then “Build settings...” under the menu.

The sample program in the DVD is created in the default installation folder (C:\Program Files\Yaskawa\MotoPlusIDE_FS). Thus, when it is installed in another folder, the build settings must be initialized.

To initialize the build settings, check the “Edit Enable/Disable” and click on the button “Restore factory defaults” .

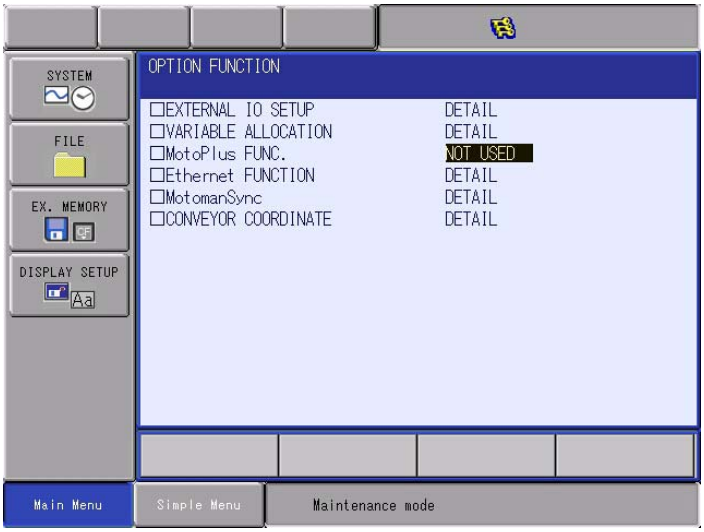
Also, compile options, build options, etc. can be changed by using “Compile Code Command” or “Build Executable Command”. But basically, use the default settings.



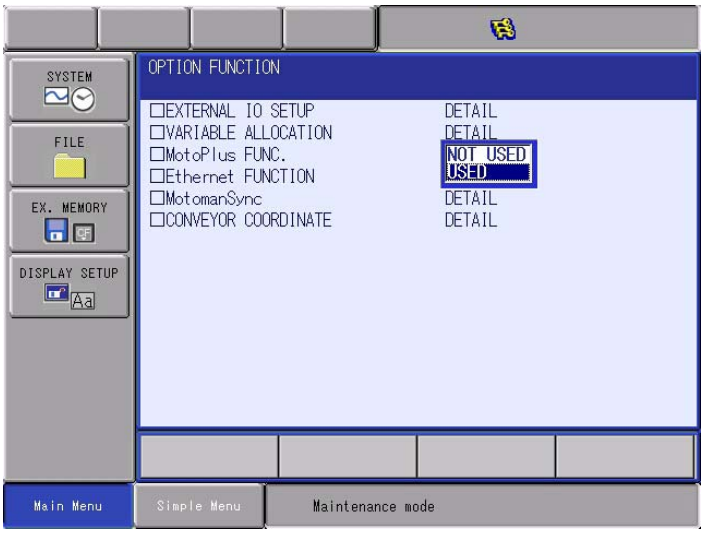
4 Valid/Invalid Setting of MotoPlus Function

The following is the procedure to set the MotoPlus function to valid.

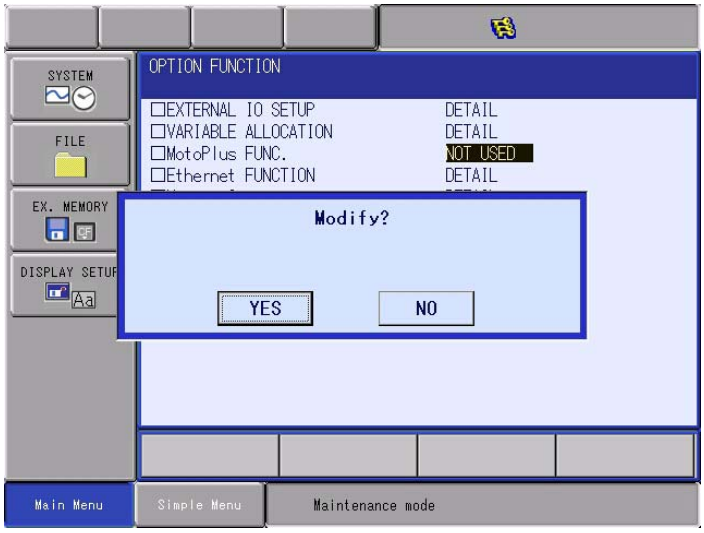
1. Start the FS100 in MAINTENANCE MODE, and change to the “MANAGEMENT MODE” in the security mode.
2. Select {SYSTEM} under the main menu, then select {SETTING}.
   – The option function window appears.



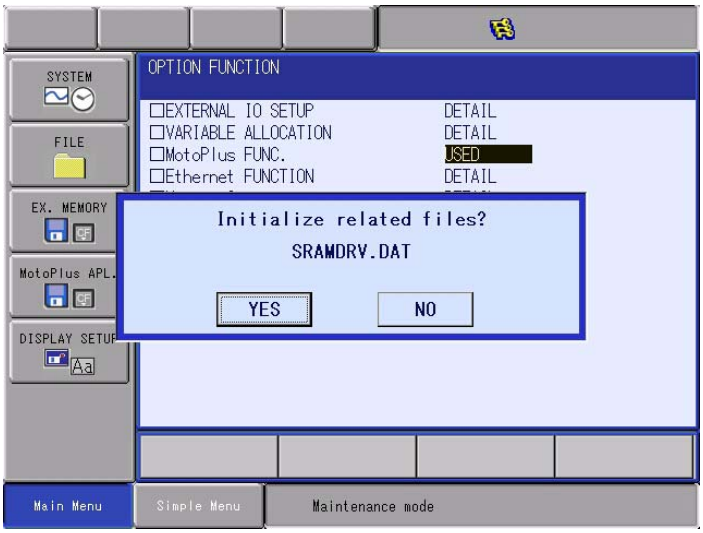
3. Select “MotoPlus FUNC.”
   – The selection list “NOT USED”/“USED” appears.



4. Select “USED”.
   – The setting change confirmation dialog appears.



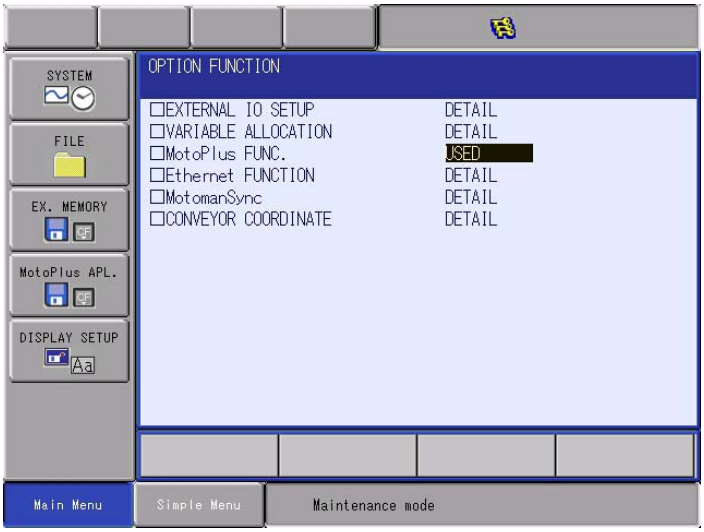
5. Select “YES” on the confirmation dialog.
   – The SRAMDRV.DAT initialization confirmation dialog appears.



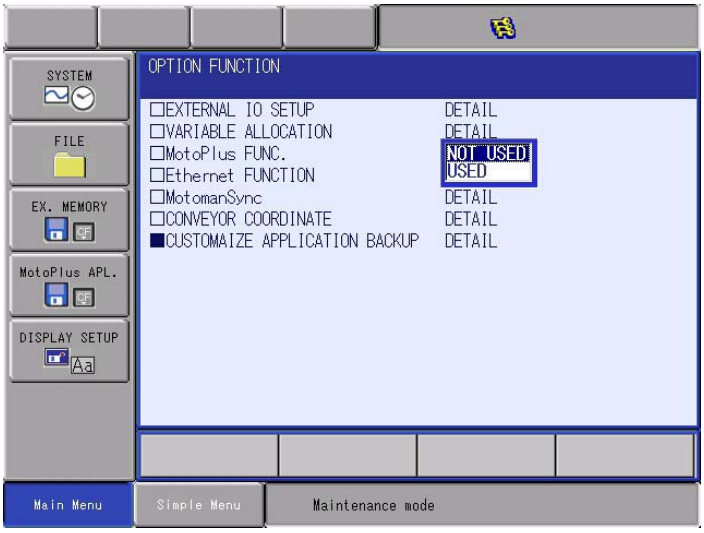
6. Select “YES” on the confirmation dialog.
   – The MotoPlus function becomes valid.

The following is the procedure to set the MotoPlus function to invalid.

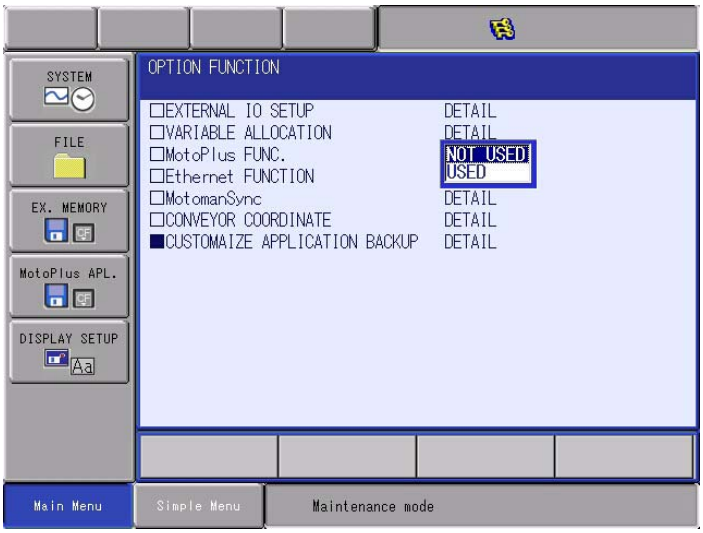
1. Start the FS100 in MAINTENANCE MODE, and change to the “MANAGEMENT MODE” in the security mode.
2. Select {SYSTEM} under the main menu, then select {SETUP}.
   – The option function window appears.



3. Select “MotoPlus FUNC.”
   – The selection list “NOT USED”/“USED” appears.



4. Select “NOT USED”.
   – The setting change confirmation dialog appears.



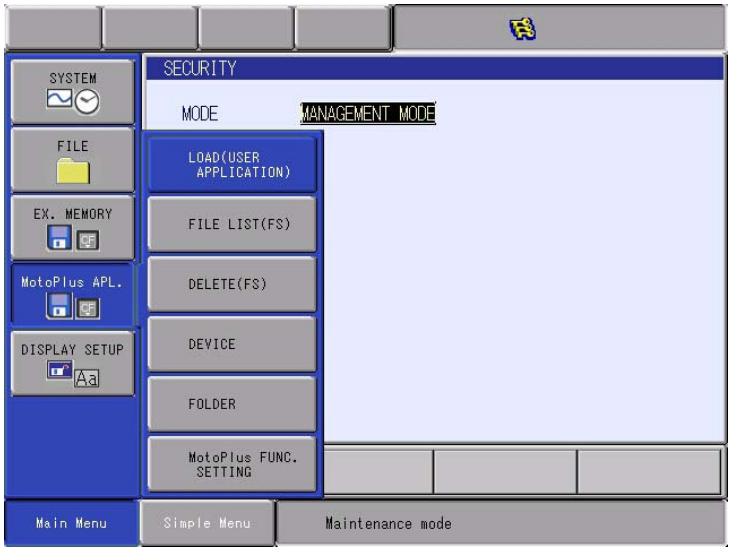
5. Select “YES” on the confirmation dialog.
   – The MotoPlus function becomes invalid.

5 Installation and Start-up of Application Program

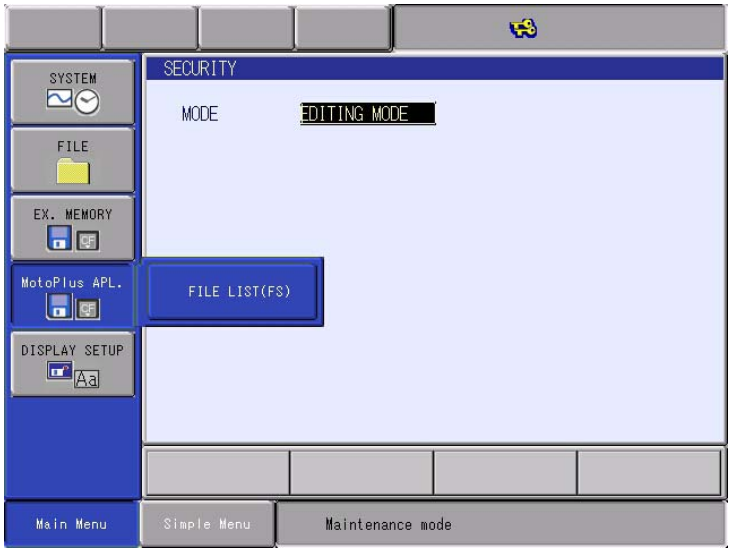
5.1 Installation of Application Program in Maintenance Mode

1. Load function
   Load the application program and install it by following the instructions below.

   (1) Set the security mode to the MANAGEMENT MODE. Select the main menu {MotoPlus APL.}. Then {LOAD (USER APPLICATION)}, {LOAD (SYSTEM LIBRARY)}, {FILE LIST (FS)}, {DELETE (FS)}, {DEVICE}, and {FOLDER} appear as submenus.



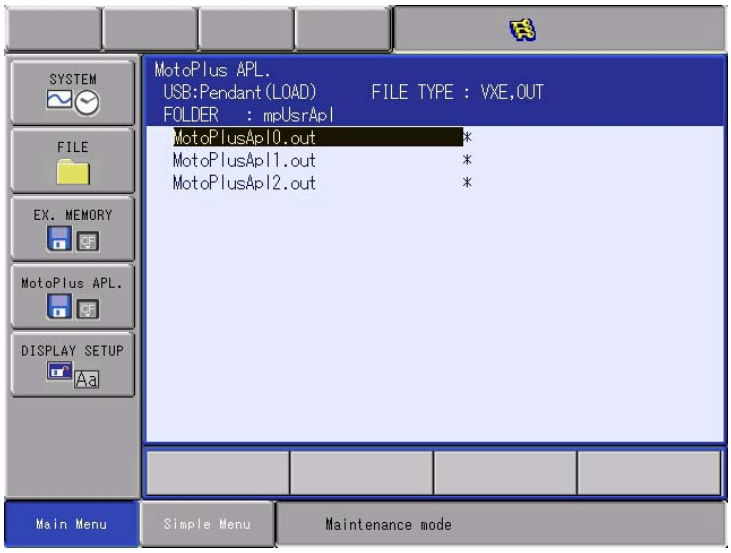
   Note that in the EDITING MODE, only {FILE LIST (FS)} appears.



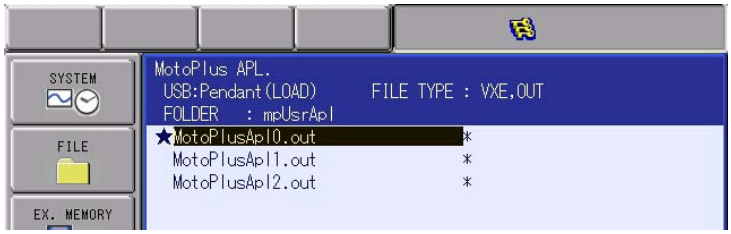
   (2) Specify the file location.
   Select an appropriate file location with the submenus {DEVICE (to select CF/USB)} and {FOLDER (to select the folder which has the application program)}.

(3) Select and load a file.

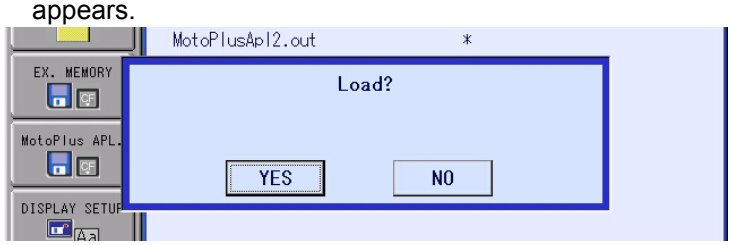
• Select the submenu {LOAD (USER APPLICATION)}. Then, "*.out" files in the specified folder appear as shown below.



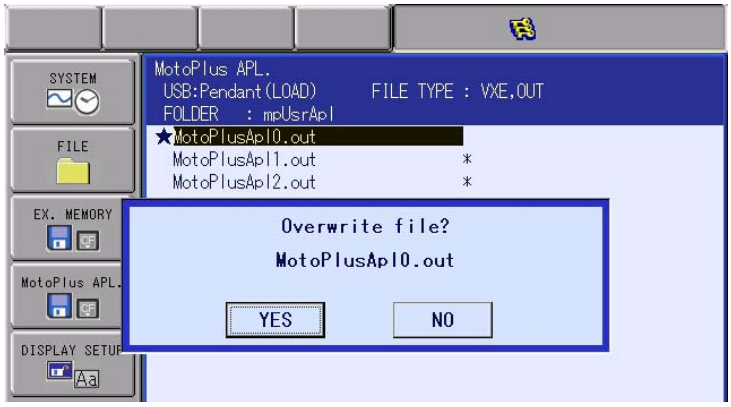
• Move the cursor and press [SELECT]. Then, the selection indicator "★" appears on the left of the application file name. Press [SELECT] again to clear "★".



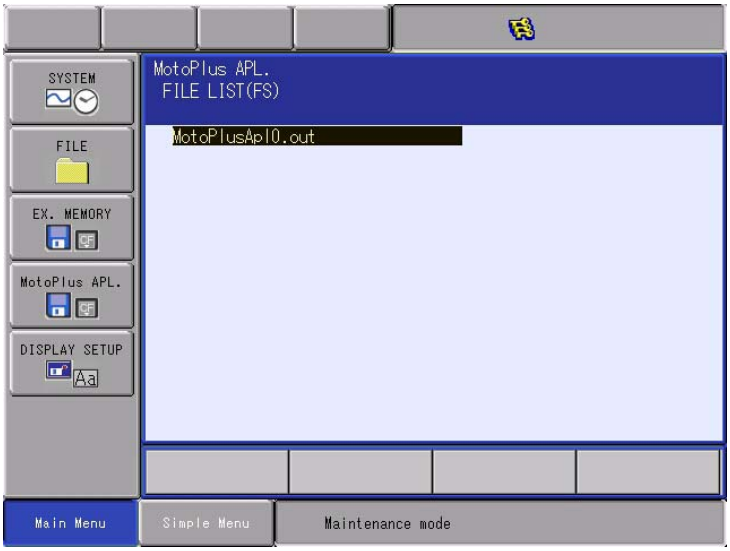
• Press [ENTER], then the following confirmation dialog box appears.



• When {YES} is selected and the selected application file contains the file which already exists in the FS100, the following confirmation dialog box appears. Select {YES} to load.

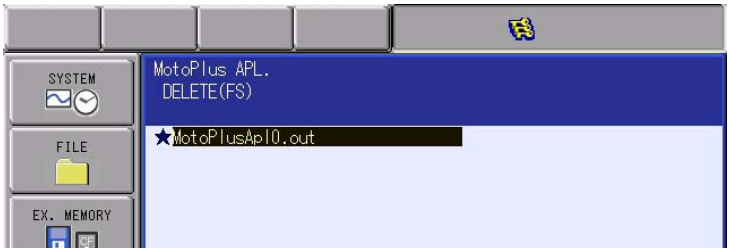


2. List
   Select the submenu {FILE LIST (FS)}. Then, the list of the application files which already exist in the FS100 appears.

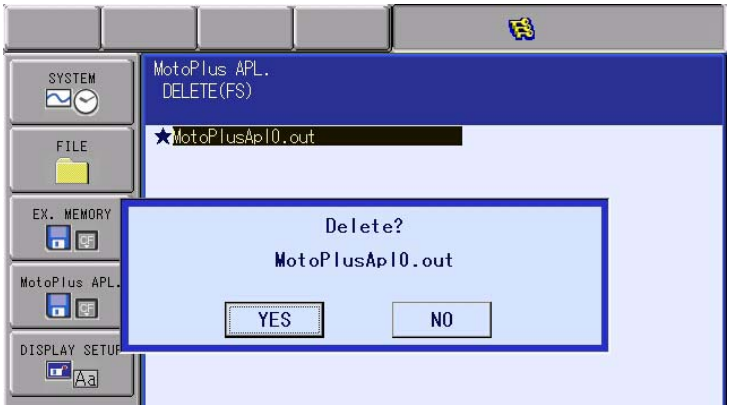


3. Deletion of application program
   By the submenu {DELETE (FS)}, the application file which already exists in the FS100 can be deleted.

   (1) Move the cursor and press [SELECT]. Then, the selection indicator "★" appears on the left of the application file name. Press [SELECT] again to clear "★".



   (2) Press [ENTER], then the following confirmation dialog box appears.



5.2 Start-up of Application Program

When the FS100 is turned ON, the application program is loaded automatically and started.

<Limitations>

- Only one “*.out” file can be loaded. Thus, install only one application. If two or more applications are installed, the alarm 1020: “MOTOPLUS APPLICATION LOAD ERROR [1]” occurs on loading. The error status can be checked by reading $B051 from the job.
- The loadable memory size is up to 2 Mbyte as the total of the code area and the static memory area. If it is more than 2 Mbyte, the start-up fails and the alarm 1020: “MOTOPLUS APPLICATION LOAD ERROR [4]” occurs on loading. The error status can be checked by reading $B051 from the job.

  $B051
  0: The application is loaded successfully on start-up.
  1: The number of files exceeds the limit.
  2: The memory is insufficient. (Available memory area is less than 3 Mbyte.)
  3: APPLICATION folder cannot be found in the CF.
  4: The size of the MotoPlus application exceeds the limit.

5.3 To Temporarily Prevent MotoPlus Application from Starting

Due to a problem of the application program, the system may hang up during robot operation, and then the FS100 may not start normally. For recovery, it is necessary to start the FS100 normally while preventing the application program from operating.

The following is the procedure not to start MotoPlus application temporarily.

1. Start the FS100 in the MAINTENANCE mode, set the security mode to the MANAGEMENT MODE. Select {MotoPlus APL.}, then {MotoPlus FUNC. SETTING} to show the setting window. Then set “APPLI. AUTOSTART AT POWER ON” to “DISABLE”.
2. Press [ENTER]. The confirmation dialog box “Modify?” appears. Select {YES}.



When the FS100 and the programming pendant cannot be connected because of a problem of the application program, open the door of the controller and change the RSW1of the main CPU circuit board from “0” to “7” by using a flat tip screwdriver(2mm), then turn ON the FS100. This process can start the FS100 in the MAINTENANCE mode forcibly. After the work, restore the RSW1 to “0”.

6 User Defined File

With MotoPlus, user defined files can be created, read, written, deleted, etc. from an application by using the file control.

These files can be saved in or read from external memory (CF or USB) by using the programming pendant.

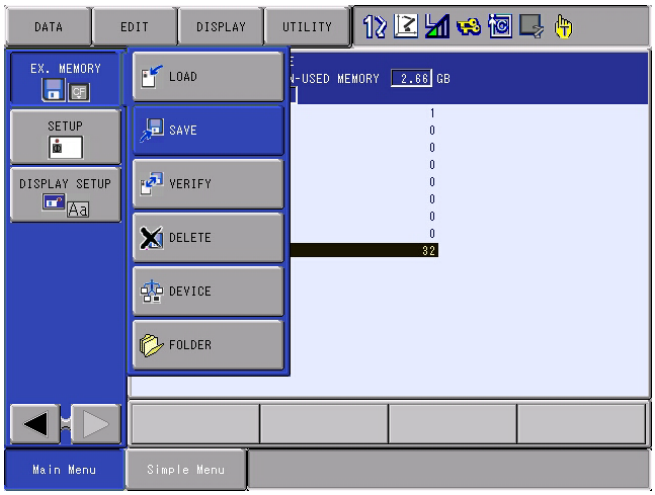
6.1 Saving in and Reading from External Memory

Saved files can be checked on the display of the programming pendant as

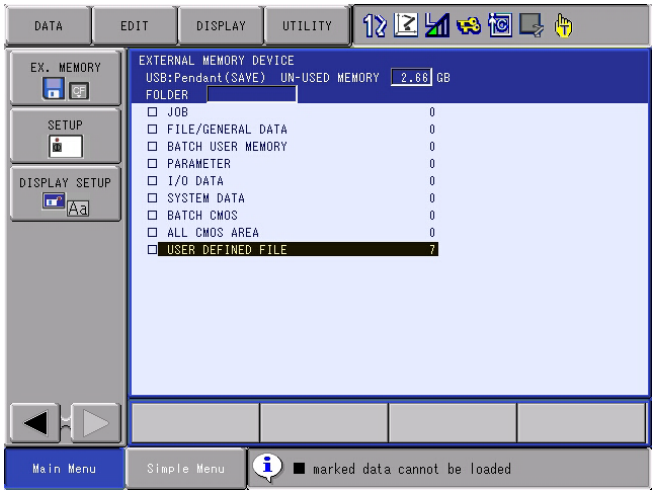
“USER DEFINED FILE”. The files can be saved in or read from external memory such as CF or USB on this display.

6.1.1 Procedure to Save or Read User Defined File

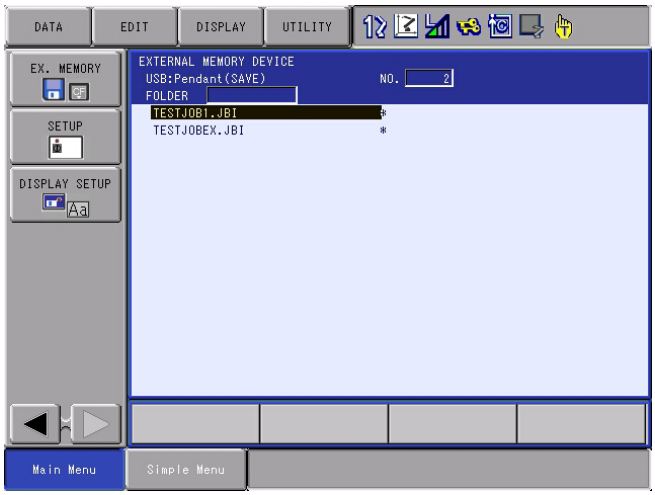
1. Start in the normal mode, and select {EX. MEMORY}, then {SAVE} or {LOAD}.



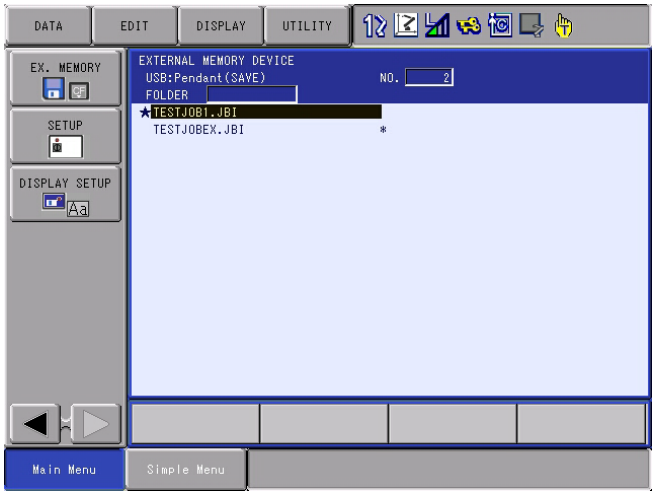
2. Select {USER DEFINED FILE}.



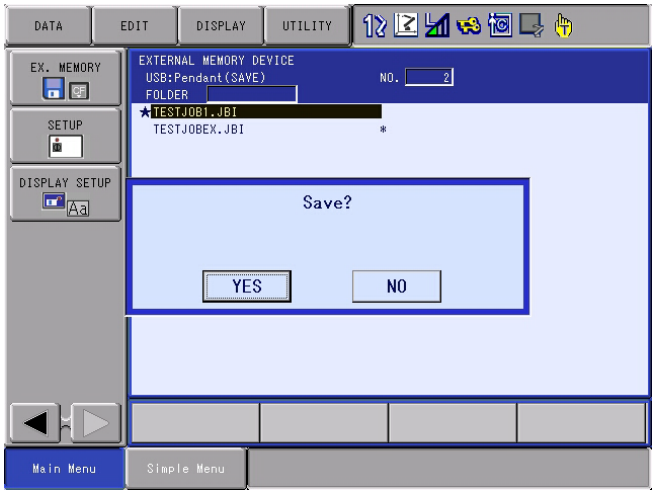
3. Select a file to be saved in external memory or to be loaded from external memory to the controller.



4. ★ appears on the left of the selected file. Multiple files can also be selected.

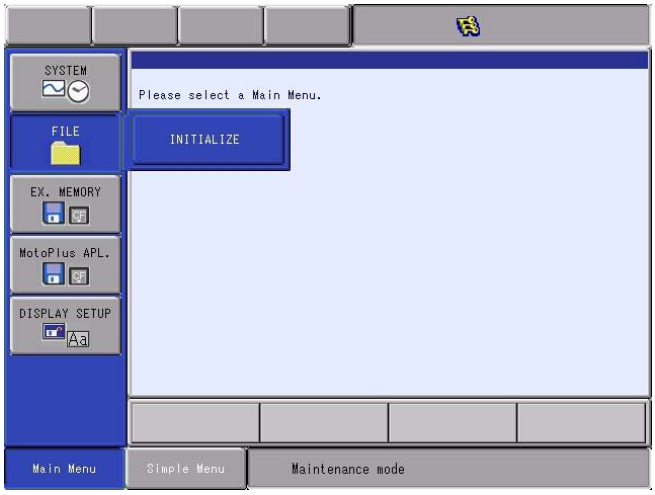


5. Select a file and press [ENTER], then the following window appears. Select “YES”.

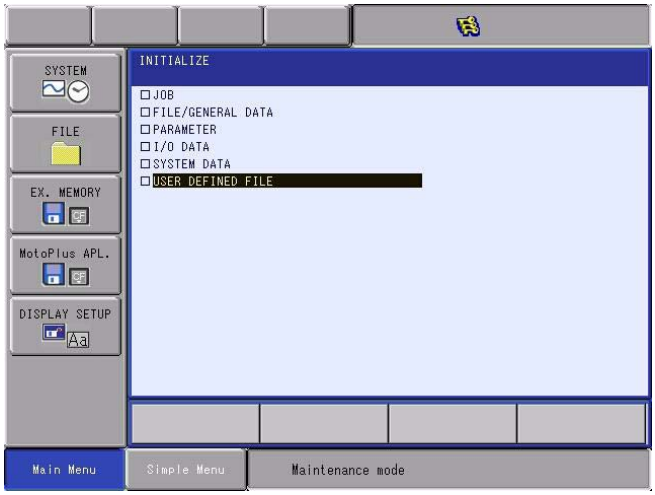


6.2 Initialization of User Defined File

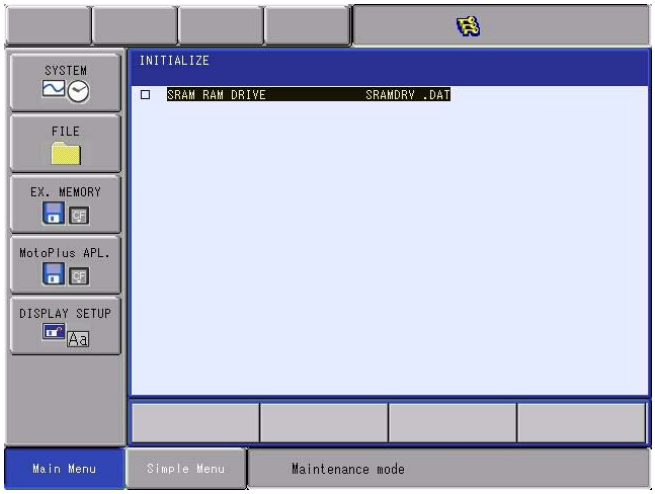
1. Start the controller in the maintenance mode. Then, select {Main Menu}, {FILE}, then {INITIALIZE}.



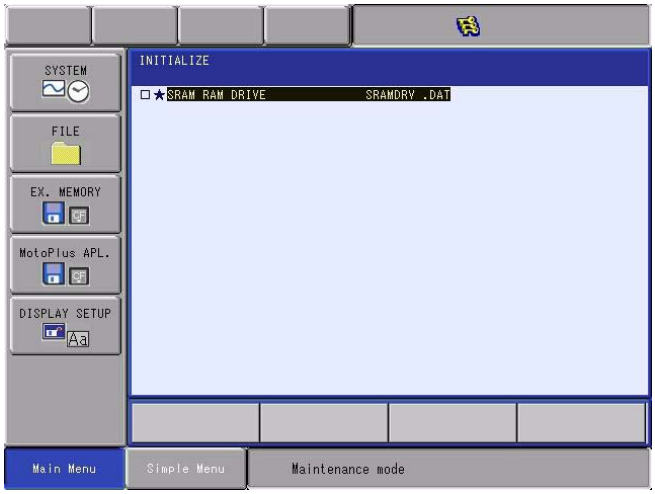
2. Select “USER DEFINED FILE”.



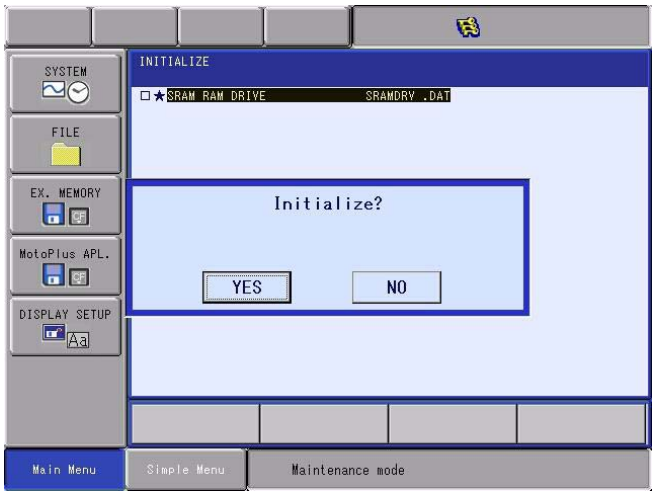
3. Select “SRAM RAM DRIVE SRAMDRV .DAT”.



4. ★ appears as shown below.



5. Press [ENTER], then the following window appears. Select {YES}.



6.3 Setting of File Control Task Priority Limit

Only the normal priority (MP_PRI_TIME_NORMAL) is used as the task priority for the file control API.

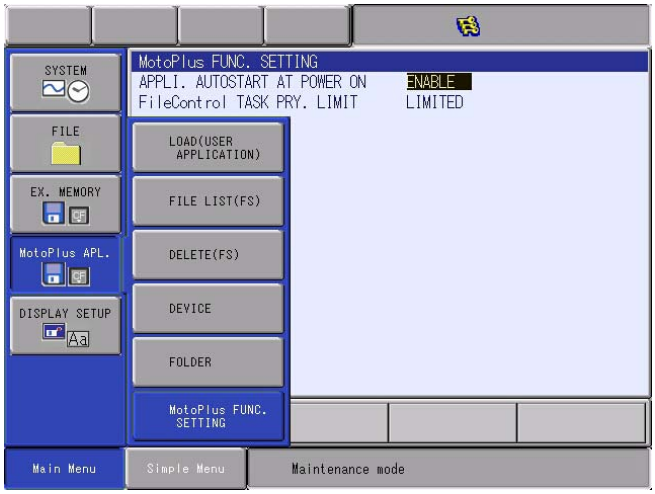
When the file control API is used from the task with higher priority, the operation differs depending on the setting of the parameter S2C1101 as shown in the following table.

This setting can also be changed by using “FileControl TASK PRY. LIMIT” in the maintenance mode.

S2C1101	FileControl TASK PRY. LIMIT	Operation
0	LIMITED	The API returns an error.
1 (default value)	AUTO	Temporarily changes the task priority to MP_PRI_TIME_NORMAL and executes the API.

6.3.1 Changing File Control Task Priority Limit

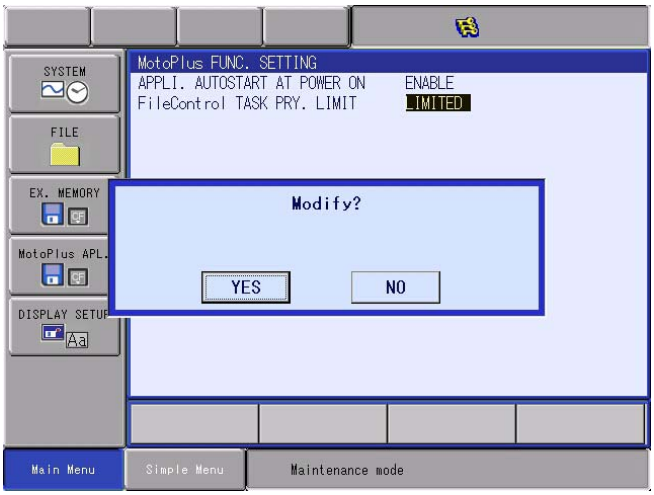
1. Start in the maintenance mode. Select {Main Menu}, {MotoPlus APL.}, then {MotoPlus FUNC. SETTING}.



2. Press [SELECT] at “FileControl TASK PRY. LIMIT” to select “LIMITED” or “AUTO”.



3. Press [ENTER], then select {YES} in the “Modify?” window to complete the setting.



7 Debugging

7.1 Debugging Environment

For the debugging environment, connect the PC for development with the RUN-TIME debugger of the controller OS via Telnet. Then you can insert printf() or puts() in the application source code to display the messages about variable and program execution states when debugging.

NOTE

Delete all the debugging codes described in the source code after completing debugging. Otherwise, the processing time of printf() and puts() may affect the system and may cause insufficient processing time, etc. during robot operation.

NOTE

With the Telnet function, debug only the tasks with normal priority. For debugging the tasks with high priority (the tasks started with the task priority: MP_PRI_IO_CLK_TAKE, MP_PRI_IP_CLK_TAKE, or MP_TIME_CRITICAL, or the task “mpUsrRoot”), use the variables of the FS100. This is because, if printf() or puts() is done with a high priority task, the processing time for the robot operation control is consumed, and a system alarm may occur due to insufficient processing time.

NOTE

When programming in C language, the pointer variable which stores the memory address can be used. If the pointer variable is set incorrectly, the system memory area may be rewritten.

If the system memory area is rewritten, critical problems occur, e.g., the software hangs up (the hang-up status is detected by a CPU exception or the watchdog check function, then the servo power turns OFF and the system shuts down), the robot stops its operation due to an alarm, or the programming pendant becomes inoperable.

If the system memory operation is incorrect, the application itself may not operate as it is designed. Thus, by testing adequately, the error location can be found. Check the functions of the application software well enough, and make sure that all functions operate as designed.

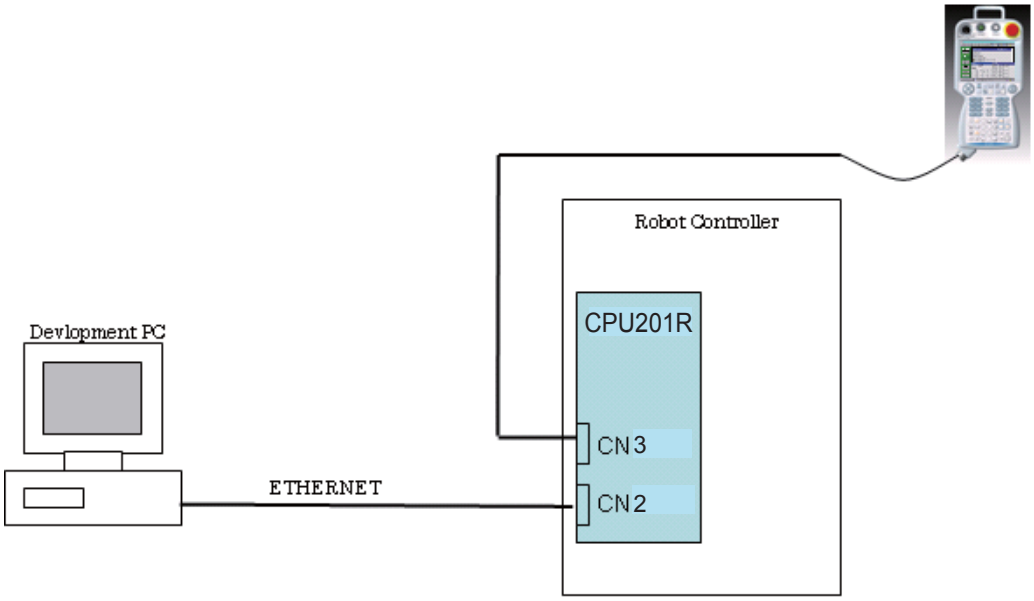
NOTE

When using the SHELL debugging function of the controller OS which becomes available by connecting with the FS100 via Telnet, a system alarm may occur due to insufficient processing time for robot operation. Do NOT use the SHELL debugging function.

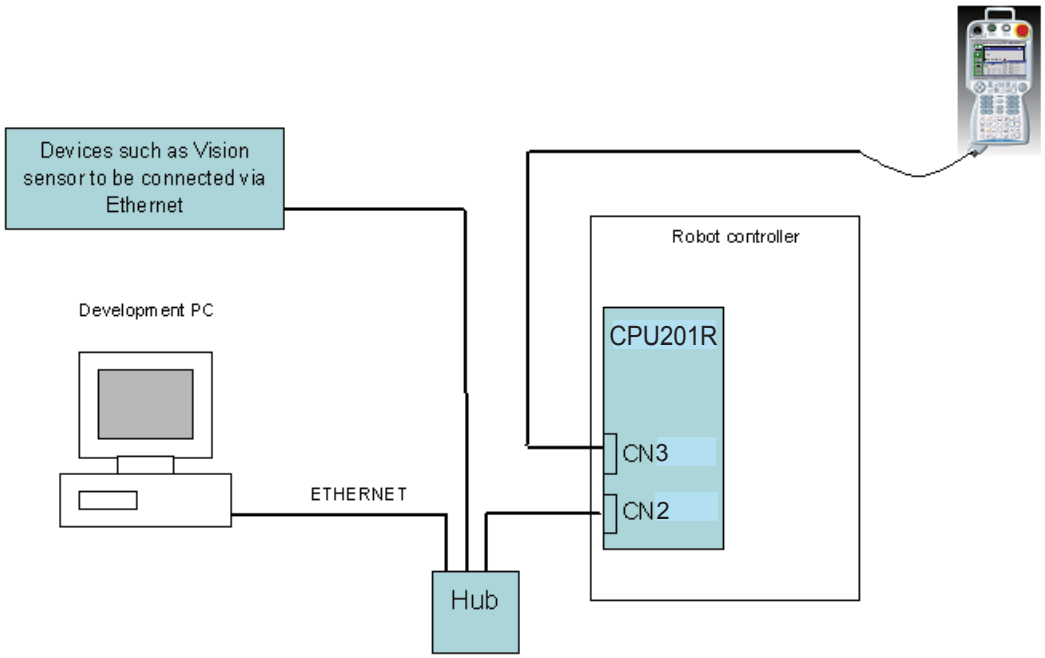
7.2 Connecting FS100 with PC

Connect the PC for development with the FS100. Then the PC can be used as the terminal to debug programs via Telnet.

As shown below, connect an Ethernet cable (shielded twisted-pair, CAT 5 or greater) to the LAN connector CN2 that is mounted on the front of the CPU201R board in the CPU rack of the FS100 controller.



To use the Ethernet service provided by MotoPlus, use a commercially available hub and set up the connection as shown below.



7.3 Debugging via Telnet Communication

The RUN-TIME debugger for the FS100 system software, can be connected with a debug PC via Telnet to provide the user with various types of debugging function. Thus, printf() and puts() functions can be inserted in the application source code, and the desired memory content or message can be shown on the Telnet terminal.

With this procedure, debug only the tasks with normal priority. For debugging the tasks with high priority (the tasks started with the task priority: MP_PRI_IO_CLK_TAKE, MP_PRI_IP_CLK_TAKE, or MP_TIME_CRITICAL, or the task “mpUsrRoot”), refer to *chapter 7.4 “Debugging by User Variables of FS100”* .

NOTE

If the Telnet function is enabled, an alarm such as “0500: SEGMENT PROC NOT READY” may occur when the FS100 is started.
This is because the Telnet task of the FS100 side is performed with a high priority, so processing of other system tasks are not performed normally.
If such an alarm occurs, restart the FS100.

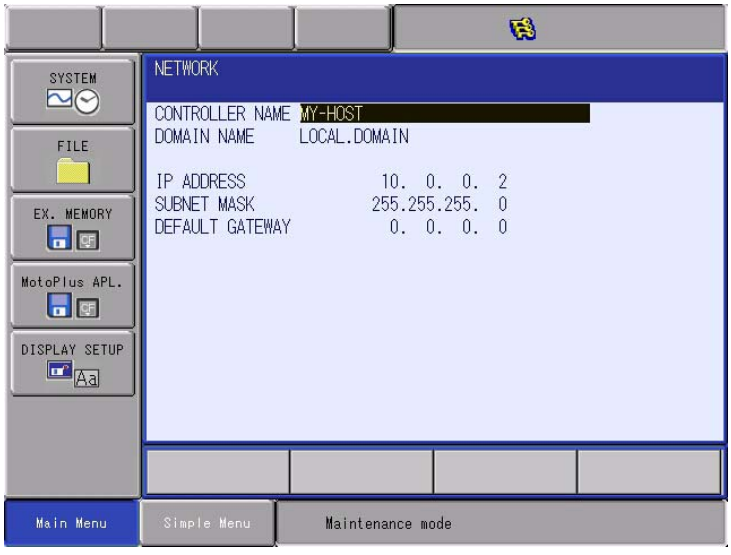
7.3.1 Operation Procedure

1. Set the IP address, etc.
   The Ethernet communication settings corresponding to your environment (such as the IP address setting at the FS100 side) must be performed in the maintenance mode.
   The procedure is as follows:

   (1) Turn OFF the FS100 then back ON while pressing the [MAIN MENU] to start the maintenance mode.

   (2) Select {SYSTEM} under the main menu, then select {SECURITY}. Then, change the security mode to the MANAGEMENT MODE.

   (3) Select {SYSTEM} under the main menu, then select {SETUP}. In the setup window, select {NETWORK}. Then, the NETWORK window appears.



(4) Set the IP address, etc. in HOST SETUP window.
Normally, set “10.0.0.2” as the IP address. Ensure that the FS100 and the WindowsPC have the same subnet address.
In the example of the FS100 setting shown below, set the IP address: 10.0.0.xx (xx is decimal number from 0 to 255, different from the FS100) and the subnet mask: 255.255.255.0 for the PC.



2. Restrictions on the IP address setting:
Ensure that all the devices that connect to the FS100 via Ethernet including the programming pendant and the PC have the same subnet address.

3. Restrictions on communication ports:
For the Ethernet function of the FS100, the system occupies the specified ports for UDP and TCP. Thus, the port numbers less than 10040 cannot be used for MotoPlus applications. Also, the ports to be used by the system may be added or changed depending on the FS100 software version. If your robot is using any optional functions related to the FS100 Ethernet function, be sure not to use the port numbers that are described in the instruction manual for each optional function.

4. Set the fixed IP address (10.0.0.9) as the network setting of the PC. (“10.0.0.” is the same as the FS100 setting.)
   (1) Select Control Panel and then double-click “Network connections”.
   (2) Double-click “Local Area Connection”.
   (3) Open the “General” window and select Internet protocol (TCP/IP), then click “Properties”.
   (4) Check “Use the following IP address”.
   (5) Enter “10.0.0.9” for the IP address.
   (6) Enter “255.255.255.0” for the subnet mask.

5. Turn ON the FS100.

6. Run the automatic login tool “FS100_TELNET.EXE” after the MOTOMAN logo is displayed on the initial screen.
   When the TELNET connection is made successfully, the prompt “->” shows and the RUN-TIME debugger of the controller OS starts.

NOTE

Run the automatic login tool after the MOTOMAN logo is displayed on the initial screen.
If the automatic login tool is run after the screen of starting connection to the controller is displayed, the memory contents and the messages cannot be shown.

7.3.2 Example Usage of printf()

The following is an example of using the printf() function. This sample displays the counter value of execution while looping.

<Operation>

1. Create the following “sample.c” with MotoPlus IDE to create “sample.out”.

```
/**** sample.c ****/
/* Copyright 2009 YASKAWA ELECTRIC All Rights reserved. */

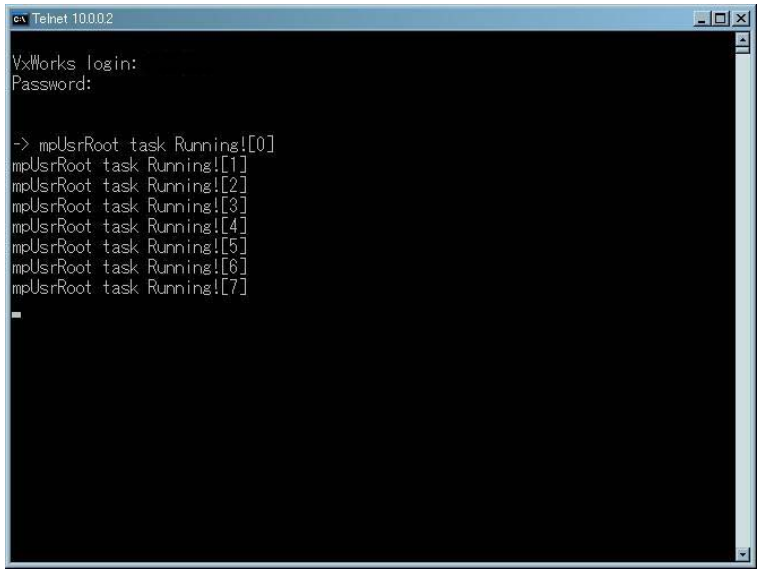
#include "motoPlus.h"

// for GLOBAL DATA DEFINITIONS

void mpUsrRoot(int arg1, int arg2, int arg3, int arg4, int arg5,
               int arg6, int arg7, int arg8, int arg9, int arg10)
{
        unsigned int run_cnt;
        run_cnt = 0;
        while (1)
        {
                mpTaskDelay(2500);
                printf("mpUsrRoot task Running![%d]¥n", run_cnt);
                run_cnt++;
        }

        mpExitUsrRoot; //(or) mpSuspendSelf
}
```

2. Install “sample.out” to the FS100. (Refer to *chapter 5 “Installation and Start-up of Application Program”* .)
3. Turn ON the FS100.
4. Connect the FS100 to the PC via Telnet.
5. The following window appears.



7.4 Debugging by User Variables of FS100

Debugging with printf() or put() for the tasks with high priority (the tasks started with the task priority: MP_PRI_IO_CLK_TAKE, MP_PRI_IP_CLK_TAKE, or MP_PRI_TIME_CRITICAL, or the task “mpUsrRoot”) may cause insufficient processing time for the robot operation control task. Thus, use the user variables of the FS100 (B variable, I variable, D variable, and R variable) to debug.

In this case, use the API for writing to the variable “mpPutVarData()” or “mpPutUserVars()” in the MotoPlus application to describe debug information, then debug while checking the data in the VARIABLE window on the programming pendant.

8 Alarm List

Alarm Number	Alarm Name	Subcode	Meaning
1020	MOTOPLUS APPLICATION LOAD ERROR	1	The number of loaded files exceeds the limit.
		2	The memory is insufficient. (Available memory area is less than 3 Mbyte.)
		3	Failed to open the directory.
		4	Load failure (Failed to open the file.)
		5	Load failure (Undefined symbol)
		6	Load failure (Others: application overloaded)
		7	Failed to initialize the API library.
		8	No user root task
		9	Failed to create the user root task.
		10	Failed to create RAM-Disk.
4478	MM TASK NO RESPONSE (MotoPlus)		The process requested from the MotoPlus application to the MM task was not completed within the specified time.
4479	MOTOPLUS MM TASK WATCHDOG ERROR		The man-machine task does not run for 3 seconds or more.

FS100 OPTIONS
INSTRUCTIONS

User's Manual for New Language Environment MotoPlus

HEAD OFFICE
2-1 Kurosakishiroishi, Yahatanishi-ku, Kitakyushu 806-0004, Japan
Phone +81-93-645-7703 Fax +81-93-645-7802

YASKAWA America Inc. (Motoman Robotics Division)
100 Automation Way, Miamisburg, OH 45342, U.S.A.
Phone +1-937-847-6200 Fax +1-937-847-6277

YASKAWA Europe GmbH (Robotics Divsion)
Yaskawastrasse 1, 85391 Allershausen, Germany
Phone +49-8166-90-100 Fax +49-8166-90-103

YASKAWA Nordic AB
Verkstadsgatan 2, Box 504 ,SE-385 25 Torsas, Sweden
Phone +46-480-417-800 Fax +46-486-414-10

YASKAWA Electric (China) Co., Ltd.
22F, One Corporate Avenue, No.222, Hubin Road, Huangpu District, Shanghai 200021, China
Phone +86-21-5385-2200 Fax +86-21-5385-3299

YASKAWA SHOUGANG ROBOT Co. Ltd.
No7 Yongchang North Road, Beijing E&T Development Area, China 100176
Phone +86-10-6788-2858 Fax +86-10-6788-2878

YASKAWA India Private Ltd. (Robotics Division)
#426, Udyog Vihar, Phase- IV, Gurgaon, Haryana, India
Phone +91-124-475-8500 Fax +91-124-475-8542

YASKAWA Electric Korea Corporation
35F, Three IFC, 10 Gukjegeumyung-ro, Yeongdeungpo-gu, Seoul, Korea 07326
Phone +82-2-784-7844 Fax +82-2-784-8495

YASKAWA Electric Taiwan Corporation
12F, No.207, Sec. 3, Beishin Rd., Shindian District, New Taipei City 23143, Taiwan
Phone +886-2-8913-1333 Fax +886-2-8913-1513

YASKAWA Electric (Singapore) PTE Ltd.
151 Lorong Chuan, #04-02A, New Tech Park, Singapore 556741
Phone +65-6282-3003 Fax +65-6289-3003

YASKAWA Electric (Thailand) Co., Ltd.
59,1st-5th Floor, Flourish Building, Soi Ratchadapisek 18,Ratchadapisek Road,
Huaykwang, Bangkok 10310, THAILAND
Phone +66-2-017-0099 Fax +66-2-017-0199

PT. YASKAWA Electric Indonesia
Secure Building-Gedung B Lantai Dasar & Lantai 1 JI. Raya Protokol Halim Perdanakusuma,
Jakarta 13610, Indonesia
Phone +62-21-2982-6470 Fax +62-21-2982-6741

©2011 YASKAWA ELECTRIC CORPORATION

Published by YASKAWA

YASKAWA

YASKAWA ELECTRIC CORPORATION

November 2018 11-12

MANUAL NO.
HW1480837 ⑦