User's Manual

EB-850/JG2+TFT

TESSERA TECHNOLOGY INC.

Date published: June 2008

V 1.01

Windows and Windows XP are registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

CAUTION:

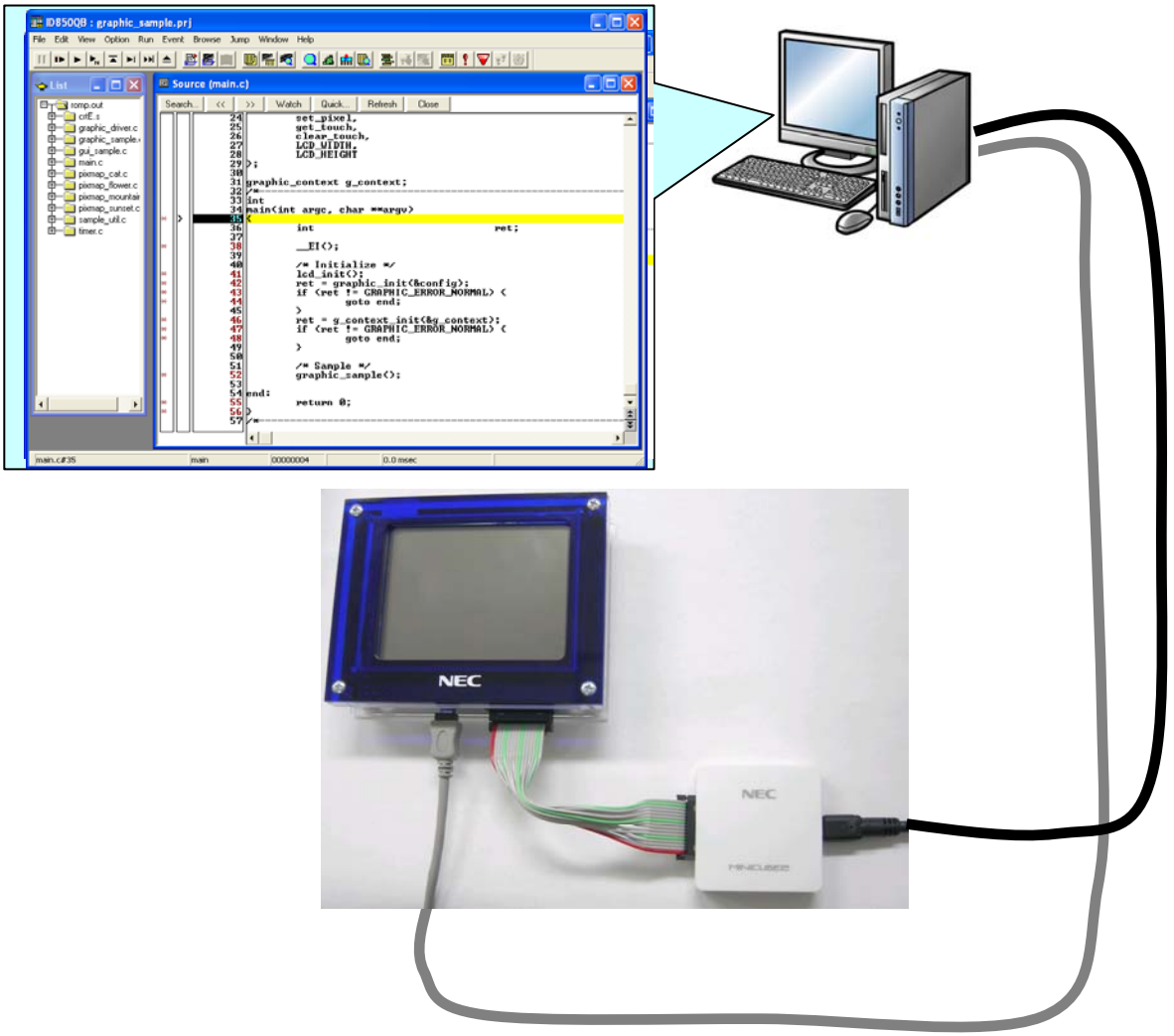
- Do not give any physical damage to this equipment such as dropping.
- Do not superimpose voltage to this equipment.
- Do not use this equipment with the temperature below 0℃ or over 40℃.
- Make sure the USB cables are properly connected.
- Do not bend or stretch the USB cables.
- Keep this equipment away from water.
- Take extra care to electric shock.
- This equipment should be handled like a CMOS semiconductor device. The user must take all precautions to avoid build-up of static electricity while working with this equipment.
- Measurement tools including the workbench must be grounded when testing.
- The user/operator must be grounded using the wrist strap.
- The connectors and/or device pins should not be touched with bare hands.

TABLE OF CONTENTS

Introduction

EB-850/JG2+TFT is the evaluation kit for developing an application system using "V850ES/JG2", NEC Electronics 32bit All Flash microcontroller.
The user only needs to install the development tools and USB driver, and connect MINICUBE2 to this kit with USB, to experience the development processes such as coding, building, debugging, and monitoring the output.

Configuration for Debugging



As the sample programs are preinstalled, the user can check the TFT displays and the touch panel operations by supplying power with bundled USB cable.

Overview

This manual consists of the following contents.
Read chapter 1 and 2 first for installing the development tools and using the sample programs.
Read chapter 3-6 for customizing the sample programs and the hardware.

Chapter 1: Preparations
Install the tools
Chapter 2: Experiences
Experience the basic operations of integrated development environment (PM+) and integrated debugger (ID850QB) with using sample programs.
Chapter 3: Hardware Specifications
Explain the hardware specifications of EB-850/JG2+TFT
Chapter 4: Troubleshooting
Describe how to solve troubles you may face, such as errors when starting the integrated debugger (ID850-QB)
Chapter 5: Other Information
Introduce other information, such as how to create a new workspace (project) on integrated development environment (PM+), how to register additional source files, and some useful tips of the integrated debugger.
The circuit diagrams of demonstration kit are included in this chapter.

Reader

This manual is intended for development engineers who are new to those development tools for development using V850.
It is assumed that the readers have been familiar with basics of microcontrollers, C and assembler languages, and the Windows operating system.

Purpose

This manual is intended to give users an understanding of the features, hardware configurations, development tools for V850.
Users can understand more by reading this manual and using the development tools together.

Chapter 1 Preparation

This chapter describes following topics:

- Overview of development tools
- Installation of development tools
- Preparation and overview of sample programs

Using the development tools bundled with EB-850/JG2+TFT, the user can experience the development processes such as coding, building, debugging, and monitoring the output.

1.1 Development Tools / Software

- Device file DF703724 V1.00
  A device file contains device specific information. So, users need a device file to use the development tools.

- Integrated Development Environment (IDE) PM+ V6.30
  The IDE works on Windows operation system.
  Users can develop a system efficiently by using the editor with idea processor function, compiler, and debugger.

- C Compiler Package CA850 W3.10 (code size limited version)
  C compiler for the V850 series. The object code size is limited to 128 Kbyte.
  This compiles C source program and assembler source program into executable code for V850 series.

- Integrated Debugger ID850QB V3.41
  This is the tool for debugging the object program generated from C compiler and assembler. The debugger enables to do C source level debugging. With the debugger, you can debug the code easily and efficiently by referring and changing variables, using step-in debugging function, and so on.

- Built-in Flash Memory Writing Program QB-Programmer
  This is the Windows software to write programs on microcontroller built-in flash memory.

- Sample Programs, "Graphic display program" and "Washing machine menu program"
  Sample programs using graphic libraries.

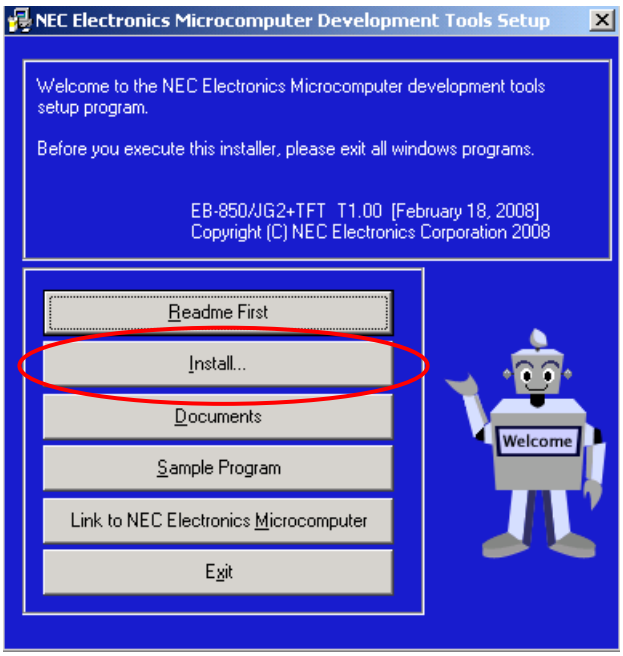
1.2 Installation of Development Tools

1.2.1 Installation Package

The installation package includes the development tools and documentations. Users can use the installer to install those development tools and documentations.

1.2.2 Installation of Development Tools

1) Please insert the CD-ROM in the drive. The installer will show up automatically.
If it does not start automatically, please start it by double clicking the SETUP.EXE from Windows Explorer.

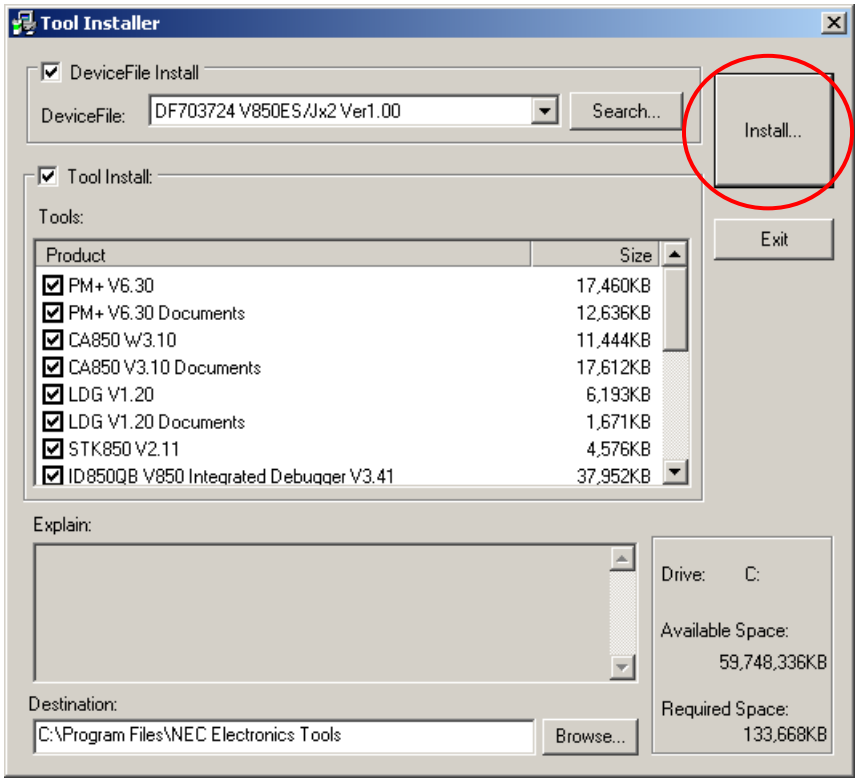


2) Click "Install…"

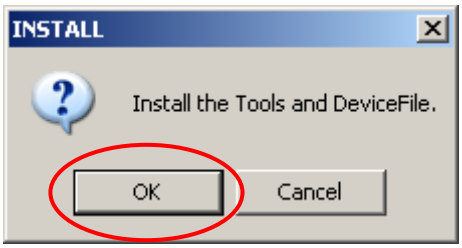
3) "Installer" dialog box shown below is opened.
Select tools that you need to install.
(As default, all the tools that you need to use the demonstration kit are selected.)
"Explain" area displays an explanation of the selected tool.
To change the installation destination, click "Browse…".

When all the settings are completed, click "Install…".

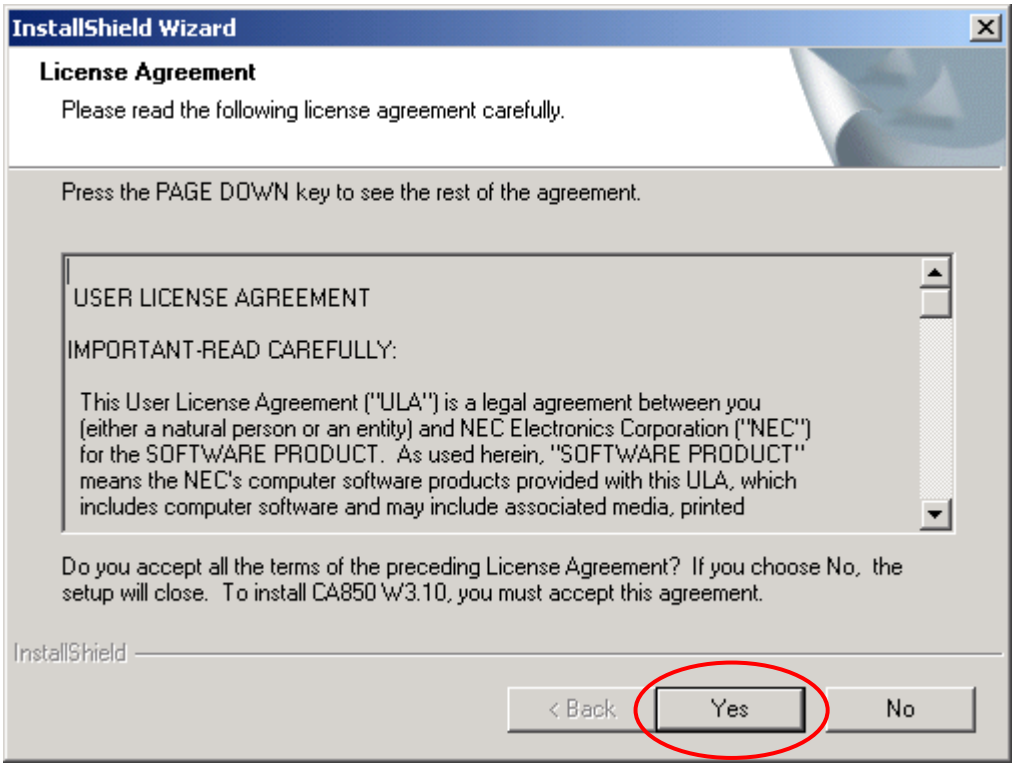
* In this document, it is assumed that users install the tools under "NEC Electronics Tools" directory (default installation directory). Users can find the tools by selecting "Start Menu" -> "Programs" -> "NEC Electronics Tools".



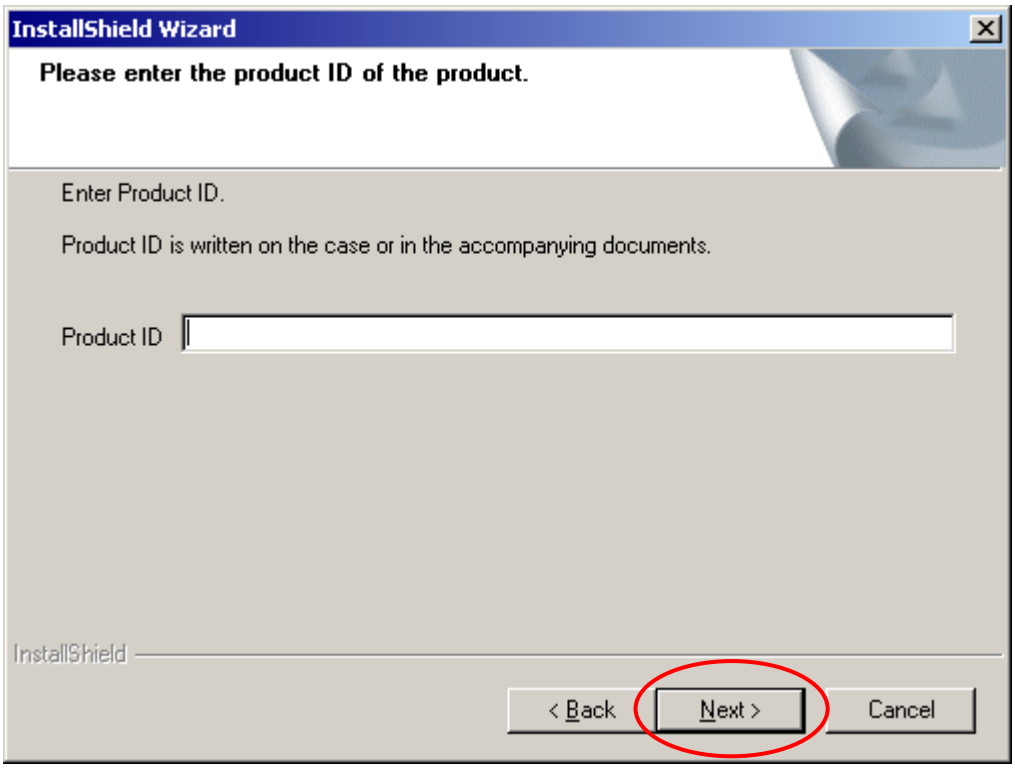
4) Click "OK", when "Install" confirmation dialog box is displayed.



5) Agree with the license agreement and click "Yes" for continuing the installation.
To stop the installation, click "No".



6) Enter the product ID, and click "Next".
* The product ID is available on the attached sheet and the document "README_E.HTML".



7) It starts copying the files.

8) When the installation is completed, the following dialog opens. Click "OK".
Installation of the development tools is completed.

[Notes on the installation]

- Support Windows versions are Windows XP and Windows 2000
- Administrator authority is required for this installation.
- Installation directory name must use only ASCII characters (1-byte alphanumeric characters). Do not use 2-byte and /*:<>?|"¥;, characters. It may cause errors.

[Limitation]

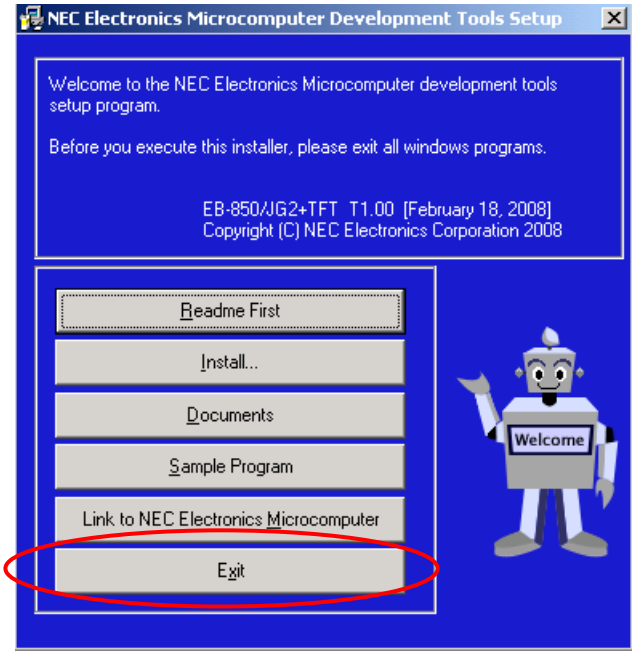
- CA850 limits the object size to 128 Kbyte.

1.3 Sample Programs

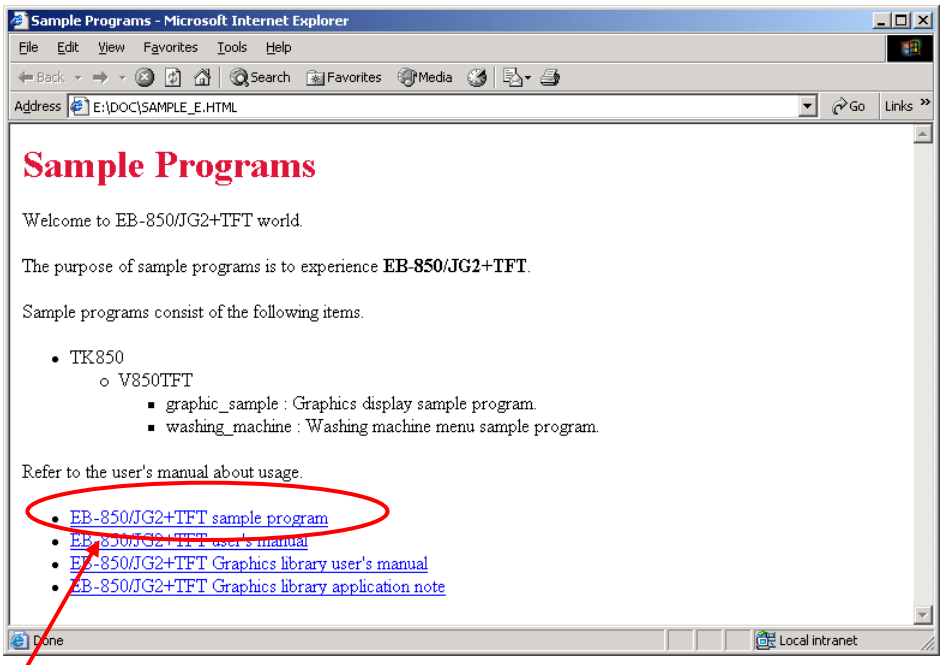
This section explains the preparation and overview of sample programs.

1.3.1 Preparation of Sample Programs

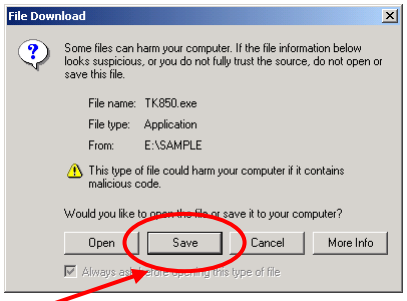
1) Please insert the CD-ROM in the drive. The installer will show up automatically.
If it does not start automatically, please start it by double clicking the SETUP.EXE from Windows Explorer.



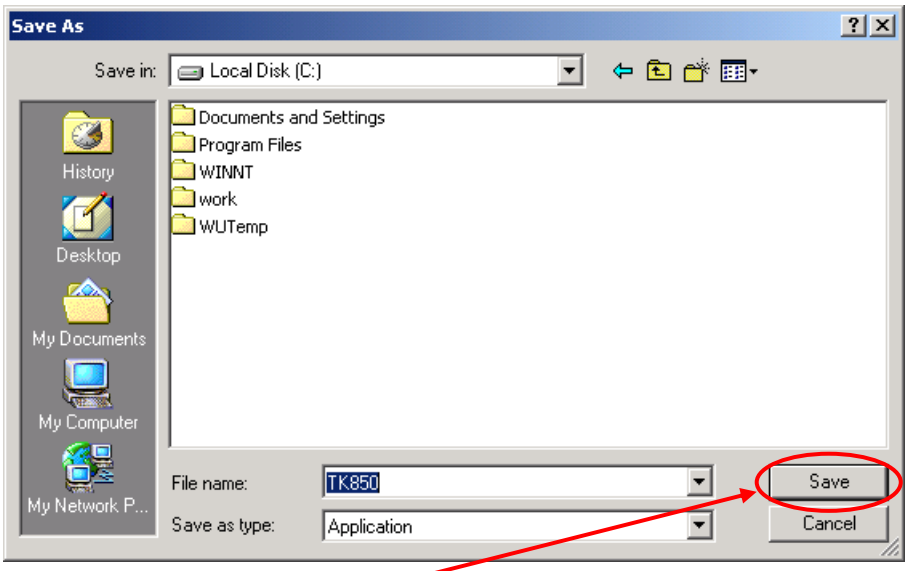
2) Click "Sample program"



3) Click "EB-850/JG2+TFT sample program"



4) Click "Save"



5) Click "Save"

6) The self-extraction sample program set (TK850.exe) is copied to the specified directory. When this file is executed, "TK850" directory is created, and then another directory is created under it for storing the sample programs.

1.3.2 Overview of Sample Programs

The sample programs consist of following directories.

Directory	Content
TK850¥V850TFT¥include	Sample program common header files
TK850¥V850TFT¥src	Sample program common source files
TK850¥V850TFT¥graphic_sample	Graphic display sample project
TK850¥V850TFT¥washing_machine	Washing machine menu sample project
TK850¥V850TFT¥bin	Bitmap conversion tool
TK850¥V850TFT¥lib	Graphic library

Chapter 2 Experiences

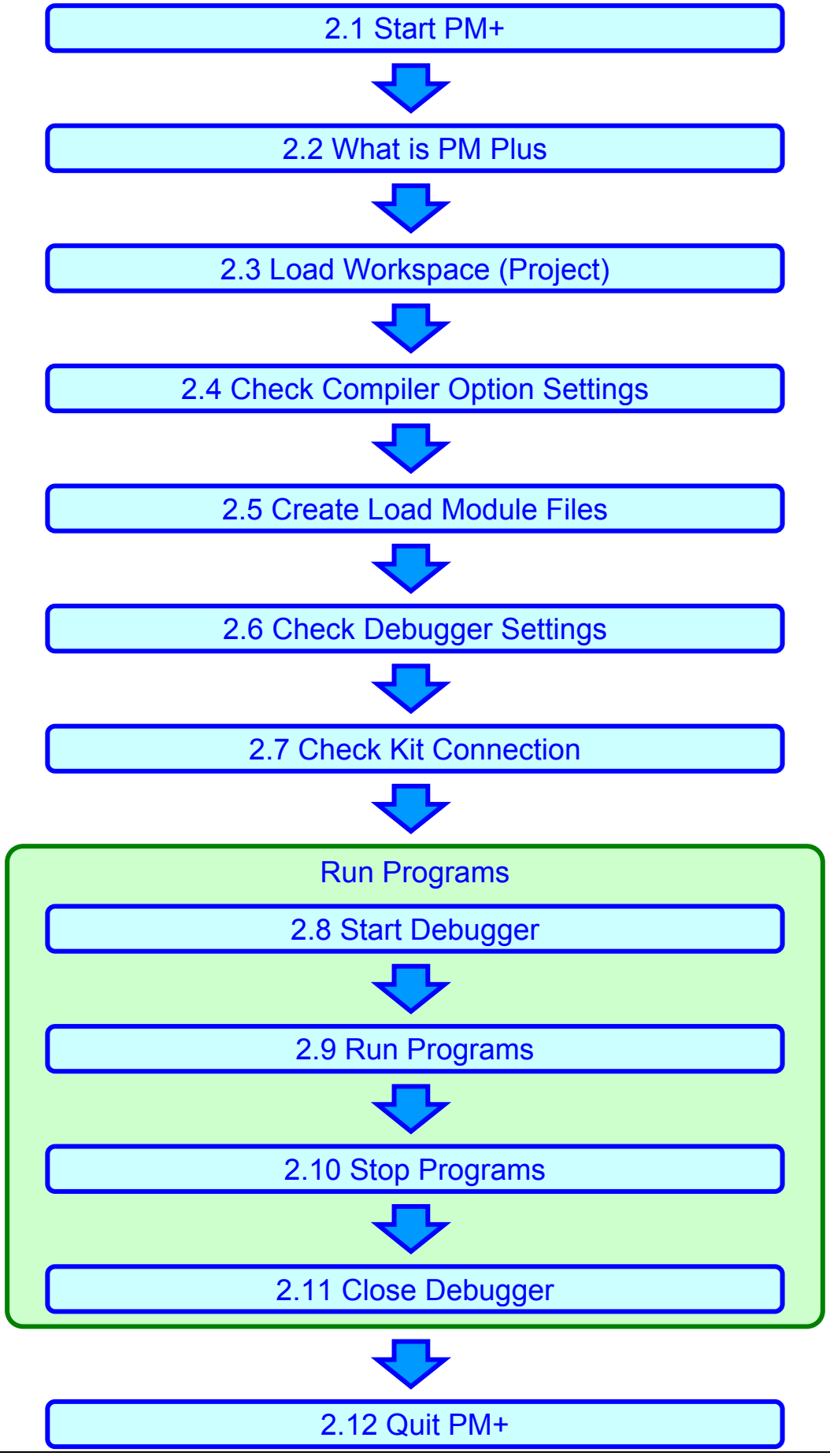
In this chapter, you will experience following development tools with using sample programs.

- Build with Integrated Development Environment (IDE), PM+
- Run the program with Integrated Debugger, ID850QB

You will use the programs that you prepared in "1.3 Sample Programs", as the sample programs for EB-850/JG2+TFT.

By building the sample programs and running the program with ID850-QB, you will be able to understand how to use the development tools (PM+, ID850QB) and the concept of project files which you need to be aware for producing application programs.

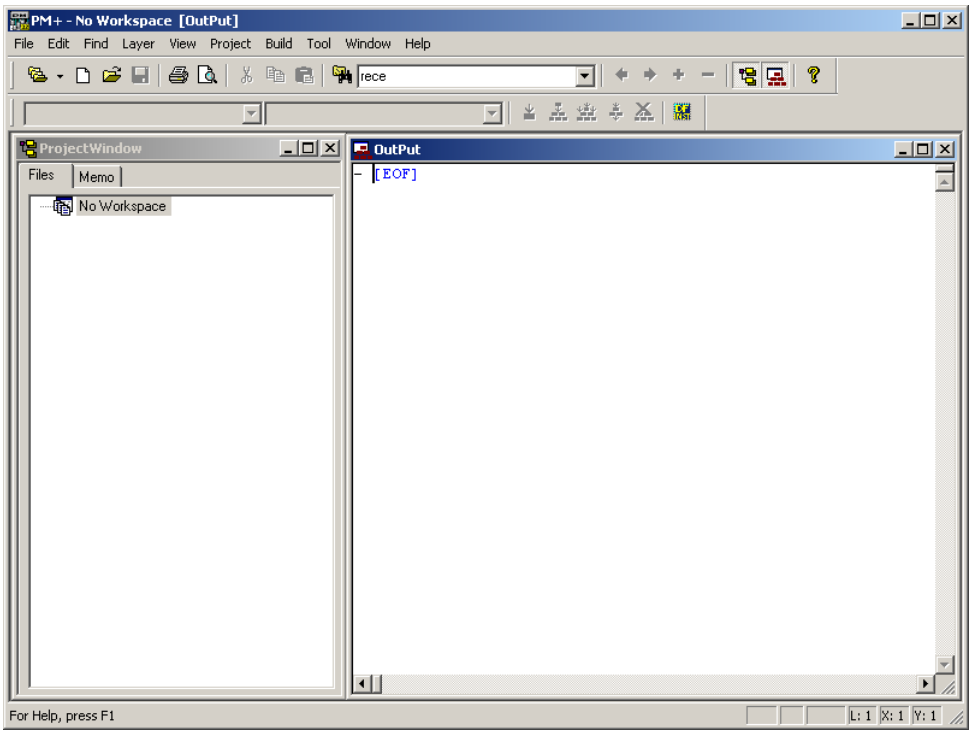
The overall steps are as follows:



2.1 Start PM+

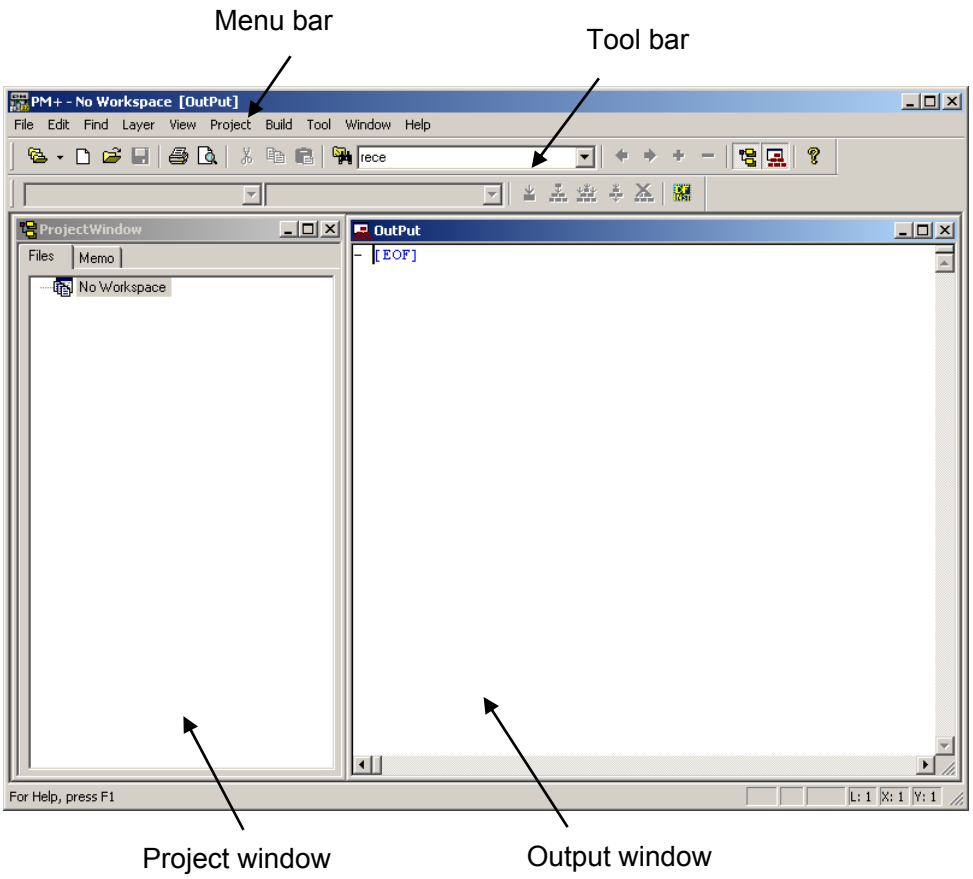
Let's start using the development tools.
First, start the PM+.
Select "Windows Start Menu" -> "Programs" -> "NEC Electronics Tools " -> " PM+ V6.30".

PM + starts up.



2.2 What is PM Plus

In PM+, application programs and environment settings are handled as a single project. It manages series of actions such as programming with the editor, source management, build, and starting debugger.
Also, one or more project files are managed as a workspace.



Project window:
A window in which the project names, source files, and include files are displayed with a tree structure.

Output window:
A window in which the build execution status is displayed.

➪ For details about the menu bar and tool bar, refer to "Help" in PM+.
"Help" can be found by selecting "Help" on menu bar, then "PM+ Help".

What is a project?

A project is the unit that PM+ manages. A project refers to application systems developed on PM+ and environment on PM+.
PM+ saves and loads project information in a "project file".

What is a project file?

A project file contains project information that includes the source files, device name, tool options for compiling, editor, and debugger information.
The file name format is "xxxxx.prj".
Project files are created in the directory you specifies when you create a new workspace.

What is a project group?

A project group is a group related projects in an application system.
The target device of each project must be the same within a project group.

What is a workspace?

A workspace is the unit used to manage all the projects and project groups required for one application system.
A workspace file contains one or more project files.
The file name format is "xxxxx.prw".

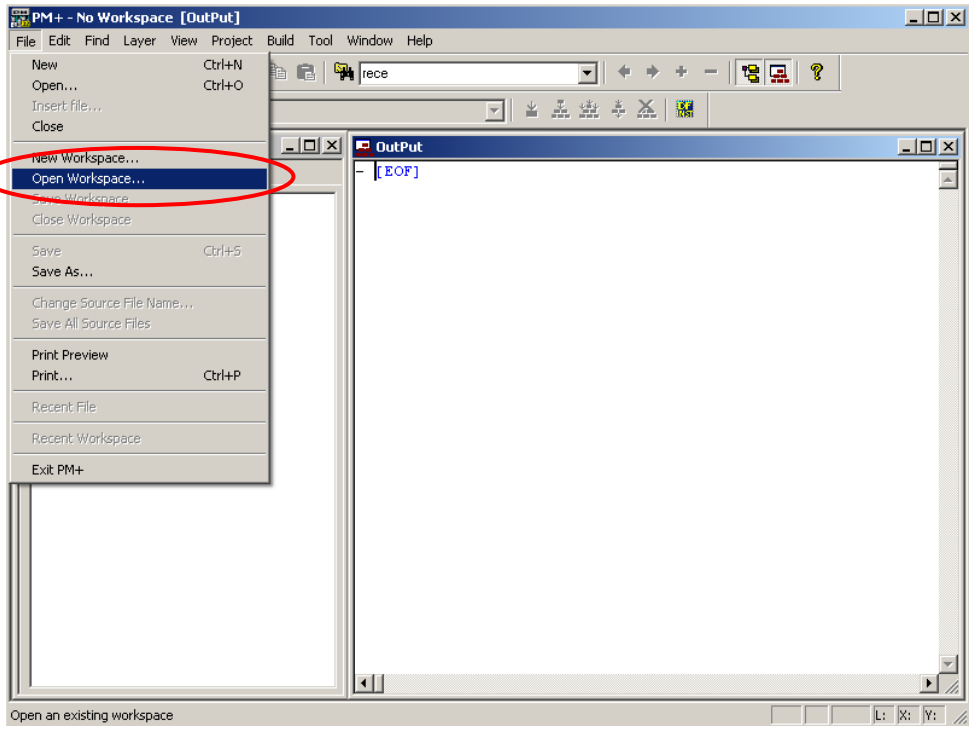
2.3 Load Workspace (Project)

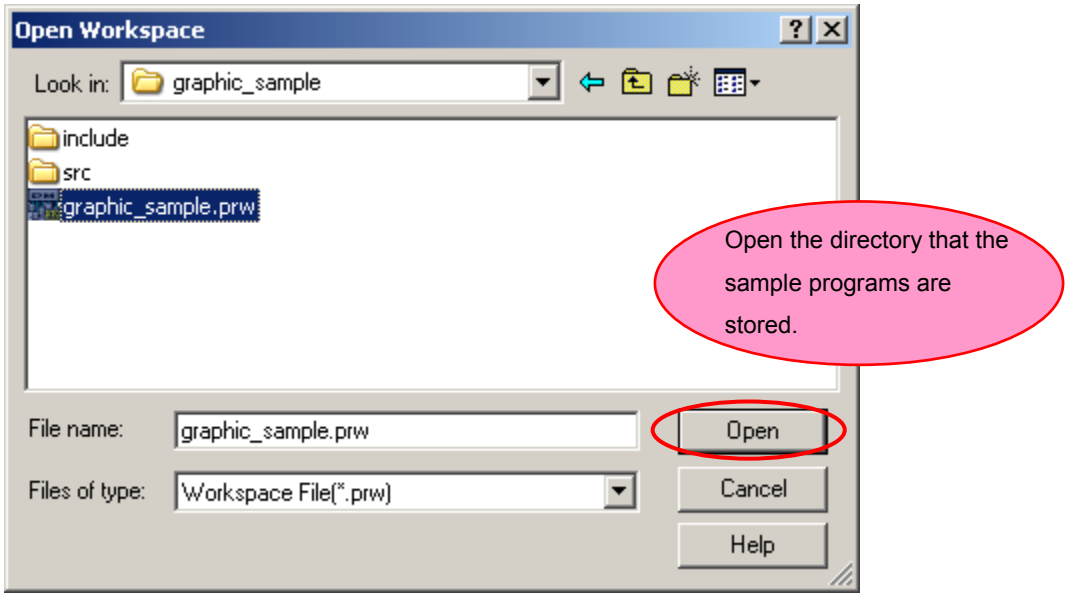
In this section, you will be using the workspace that you created in "1.3 Sample Programs".

⇨ For creating a new workspace, refer to "Chapter 5 Other Information".

The workspace has information about the build environment for the sample programs.

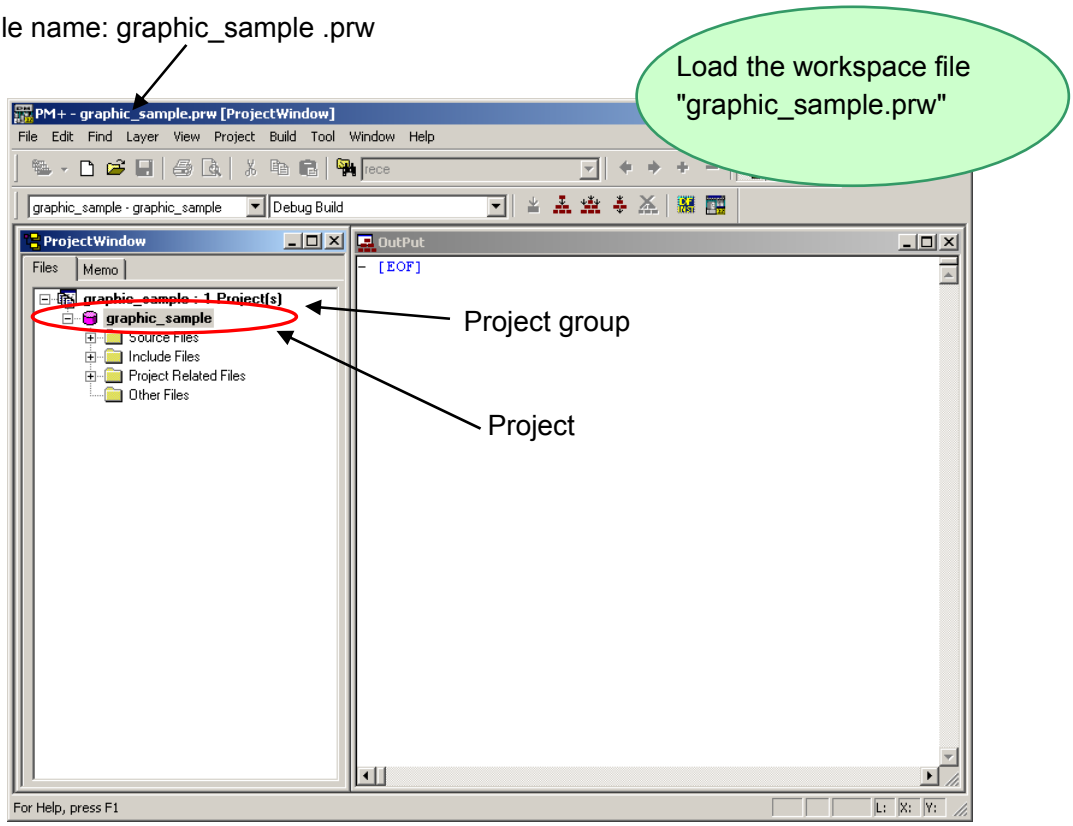
Select "File" on menu bar and "Open Workspace…".
Then, select "graphic_sample.prw" under the directory "TK850¥V850TFT¥graphic_sample".





Select "graphic_sample.prw" and click "Open".

Workspace file name: graphic_sample .prw



The workspace file "graphic_sample.prw" contains one project called "graphic_sample". You will use this project "graphic_sample".

CAUTION:
Please ignore when you get a prompt saying that files could not be found. This may be occurred when the installation directory is not the default.

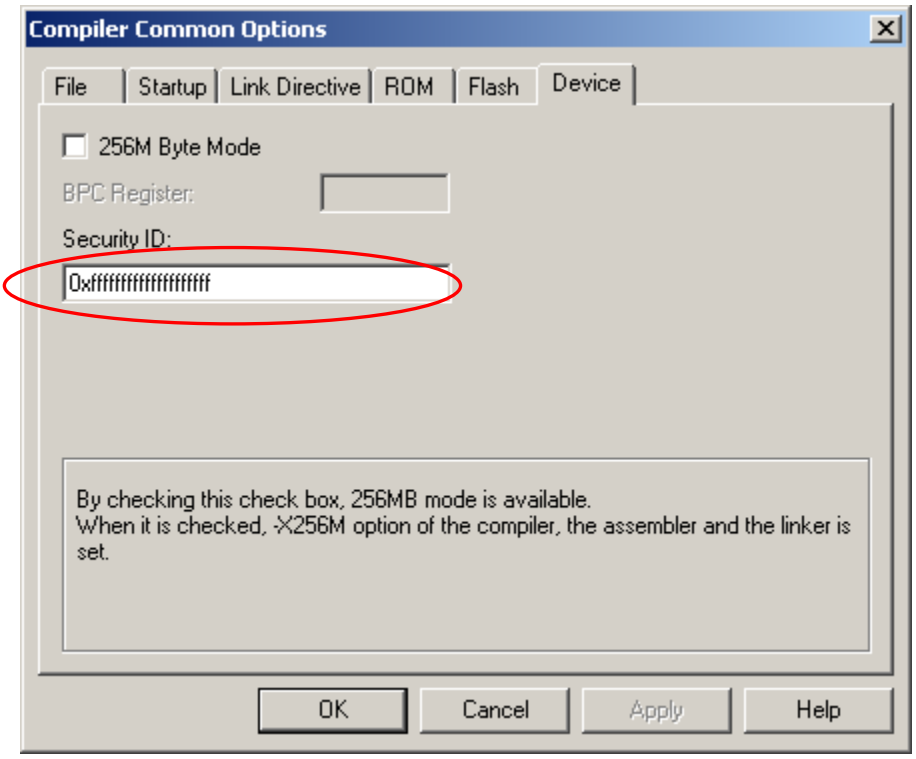
2.4 Check Compiler Option Settings

The options have been set by project file. However, because some options are important and useful, following two settings are covered in this section.

- To set security ID
- To enable C++ comments

2.4.1 Compiler Common Options

Select "Tool" on menu bar, then "Compiler Common Options...".
Select "Device" tab on "Compiler Common Options" window, and check following settings.

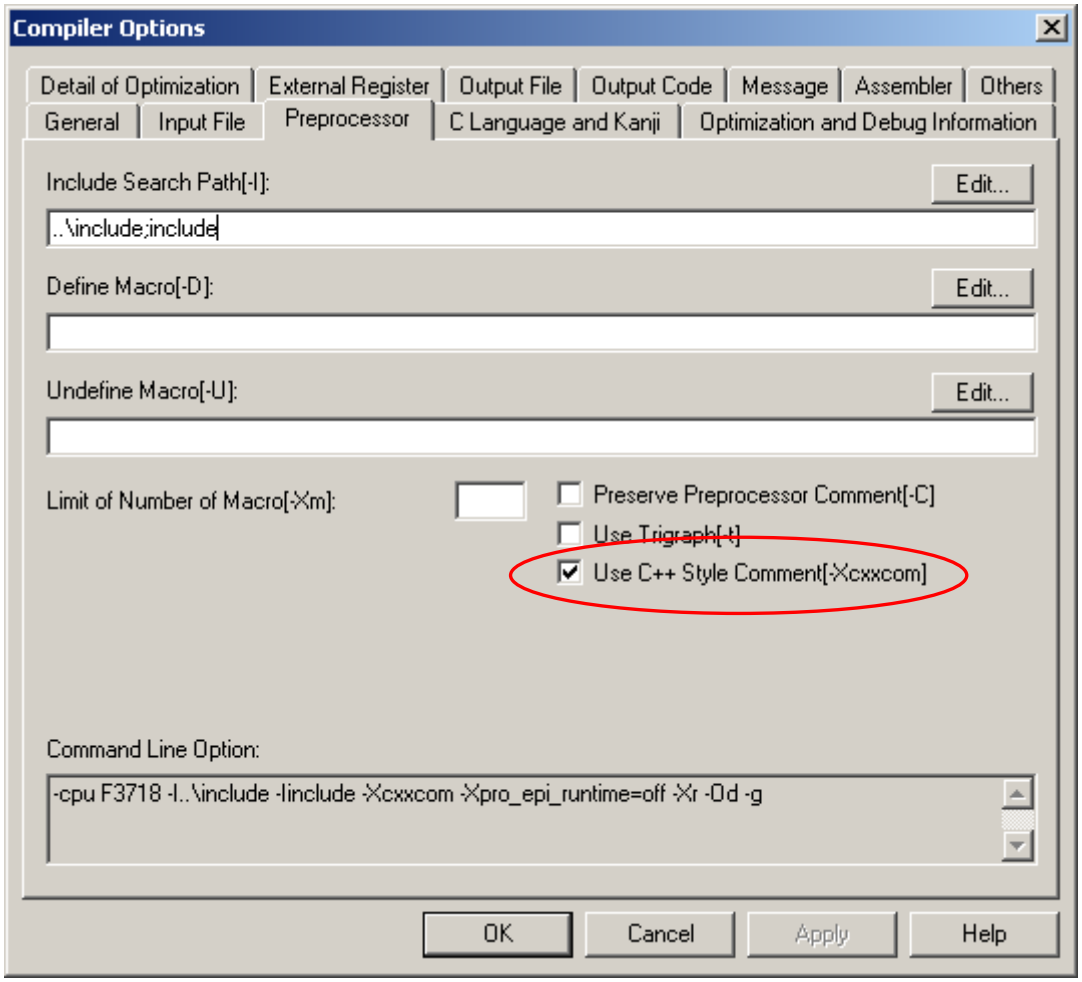


Set the security ID. The security ID is a specific ID code (10 bytes) used for the authentication when the debugger is starting. The security ID is stored at the address 70H-79H in the microcontroller built-in flash memory. The security ID in the flash memory and the ID entered in the configuration dialog are compared when the debugger is starting. When the ID does not match, the debugger cannot be started. Therefore, it is useful to protect the program data in the memory from others.
If you do not need to care about the security, it is recommended to enter "FFFFFFFFFFFFFFFFFFFF". (When you erase the data in the flash memory, the ID is set to this value.)

If you forgot the security ID code stored in the address 70H-79H, the debugger (ID850-QB) will not be able to start.
In this case, you need to use "QB-Programmer" to erase the built-in flash memory. By erasing the flash memory, the security ID is set to "FFFFFFFFFFFFFFFFFFFF".
For details, refer to "5.4 QB-Programmer".

2.4.2 Preprocessor Tab

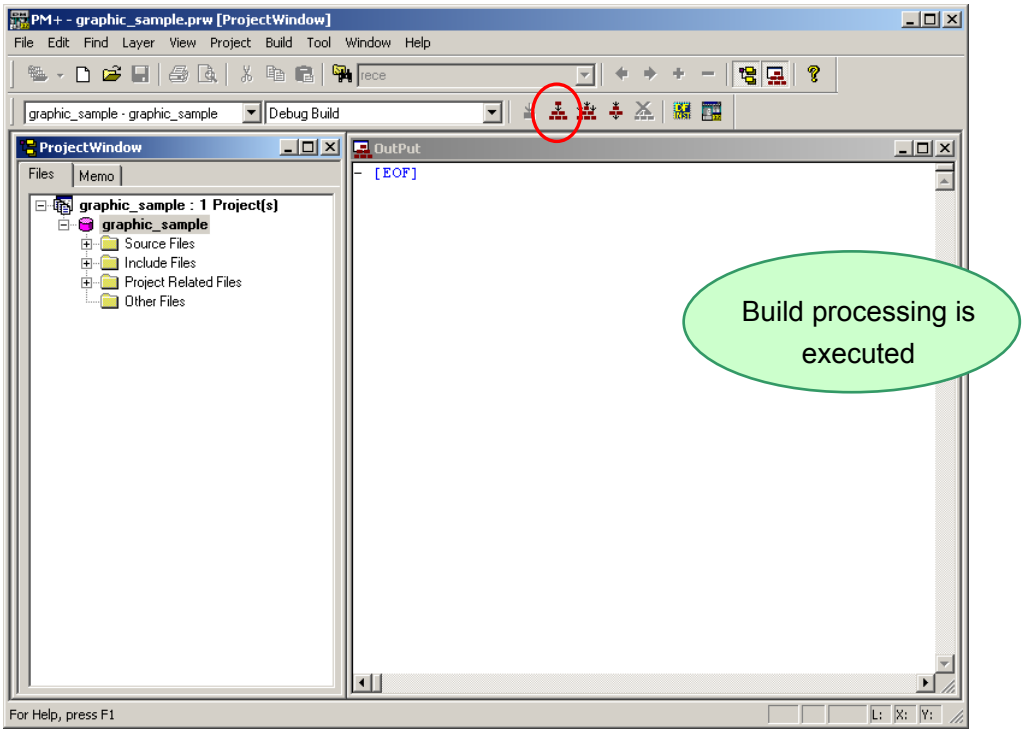
Select "Tool" on menu bar in PM+, then "Compiler Options...".
Select "Preprocessor" tab, and check "Use C++ Style Comment".
This setting allow you to use the C++ comment using "//".
It is useful feature when editing programs.

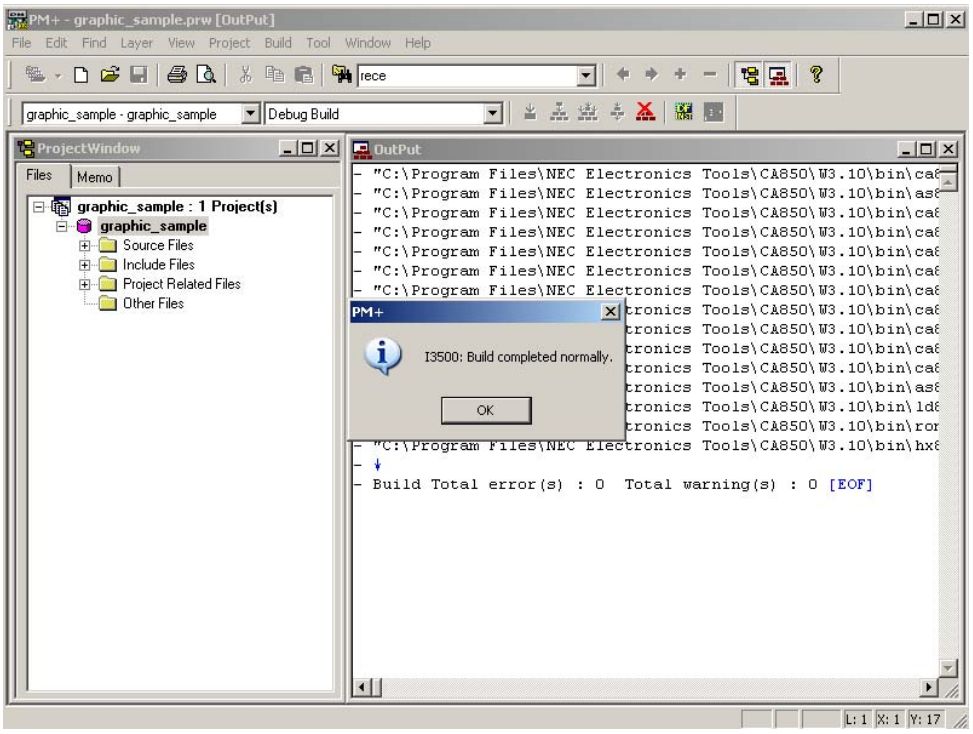


2.5 Create Load Module Files

Now, you are going to compile, assemble, and link the program you developed to create load module files. This process is called build.

Click the build button  in PM+, or select "Build" on menu bar, then "Build".





Build has been completed successfully.

What is build?

Build is a function that creates an executable file from source files in a project.
PM+ automatically performs compiling, assembling, linking, and other processing actions.
To reduce the time for the build, PM+ detects and compiles/assembles only the files that have been updated from the previous build process.

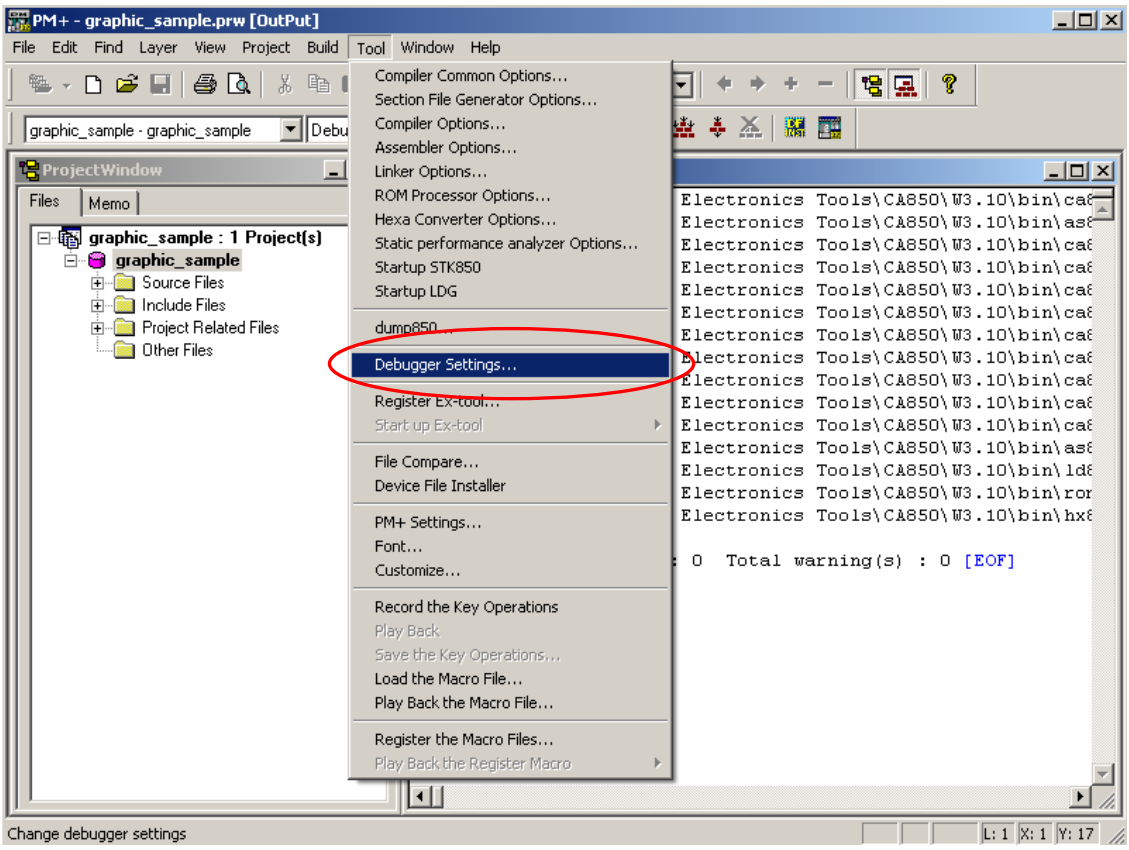
What is rebuild?

Build compiles and assembles only the source files that have been updated from the previous time, whereas rebuild compiles and assembles all the source files.
When the settings, such as compiler options, have been changed, you must rebuild instead of build.

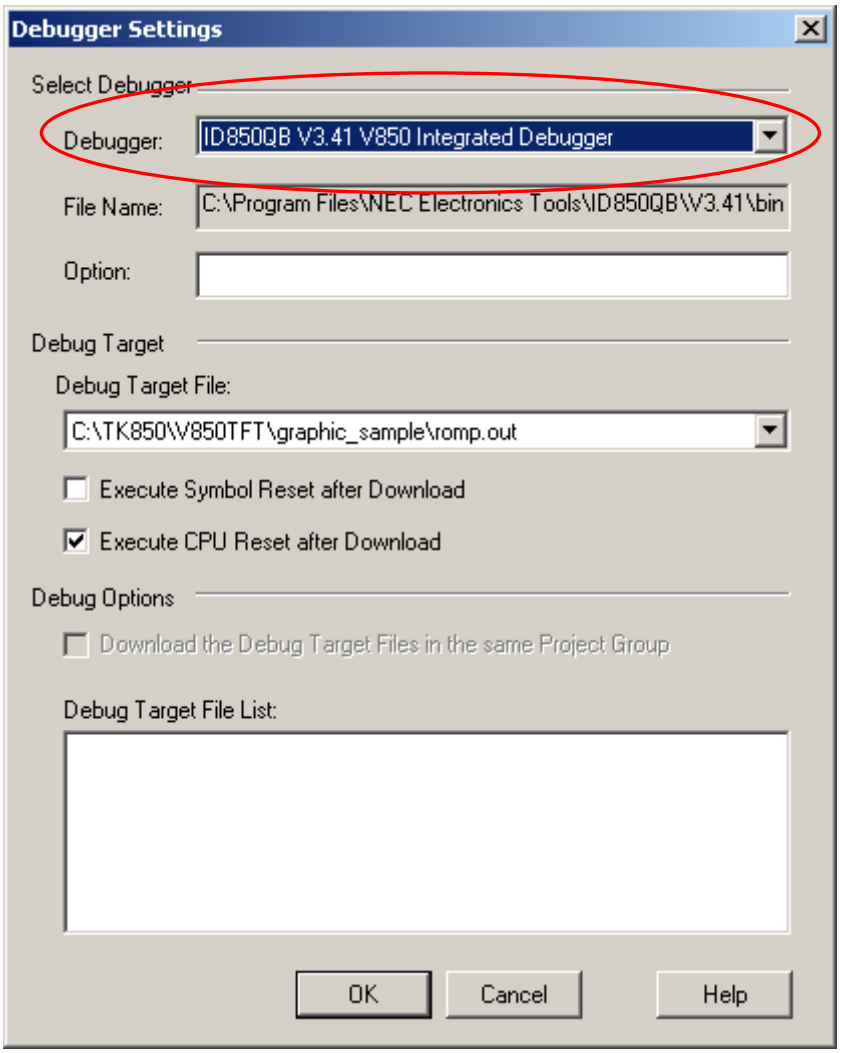
2.6 Check Debugger Settings

As the build process is completed, now you are going configure the debugger settings for debugging. The debugger settings have been set by the project file as well. However, because those settings are important for debugging, some settings are covered in this section.

Select "Tool" on menu bar in PM+, then "Debugger Settings...".



Check if "ID850QB V3.41 V850 Integrated Debugger" is selected on "Debugger".



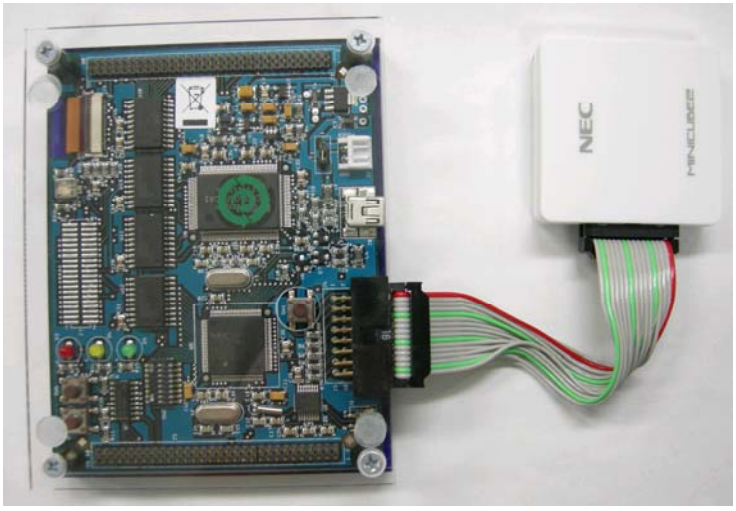
If you cannot select "ID850QB V3.41 V850 Integrated Debugger", use menu bar to select "Project", "Project settings" -> "Tool version settings" -> "Detail settings" -> then select "ID850QB" and "V3.41".

2.7 Check Kit Connection

To debug, connect the kit, MINICUBE2, and PC.

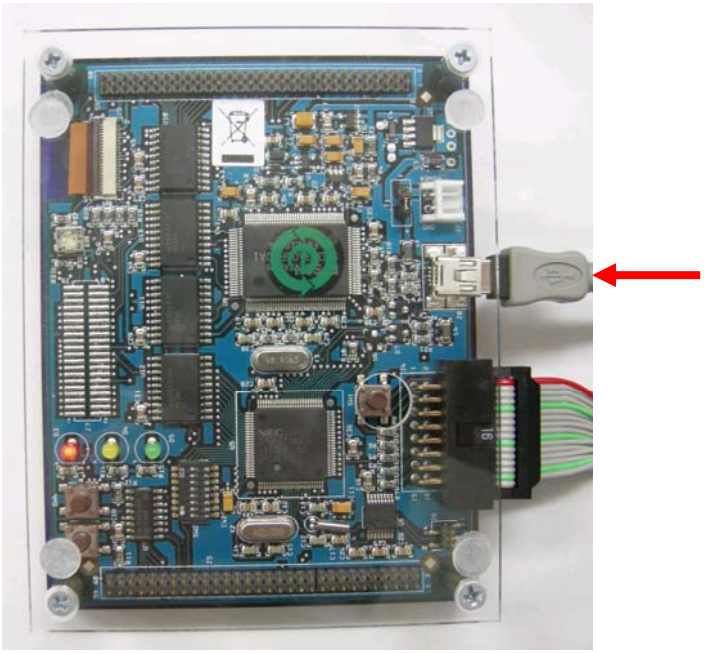
1) Set the switches on MINICUBE2 to "M2" and "T" as shown below.

2) Connect the MINICUBE2 cable to J4 on EB-850/JG2+TFT as shown below.



3) Connect MINICUBE2 to PC with USB cable.

4) Connect EB-850/JG2+TFT to PC with USB cable.



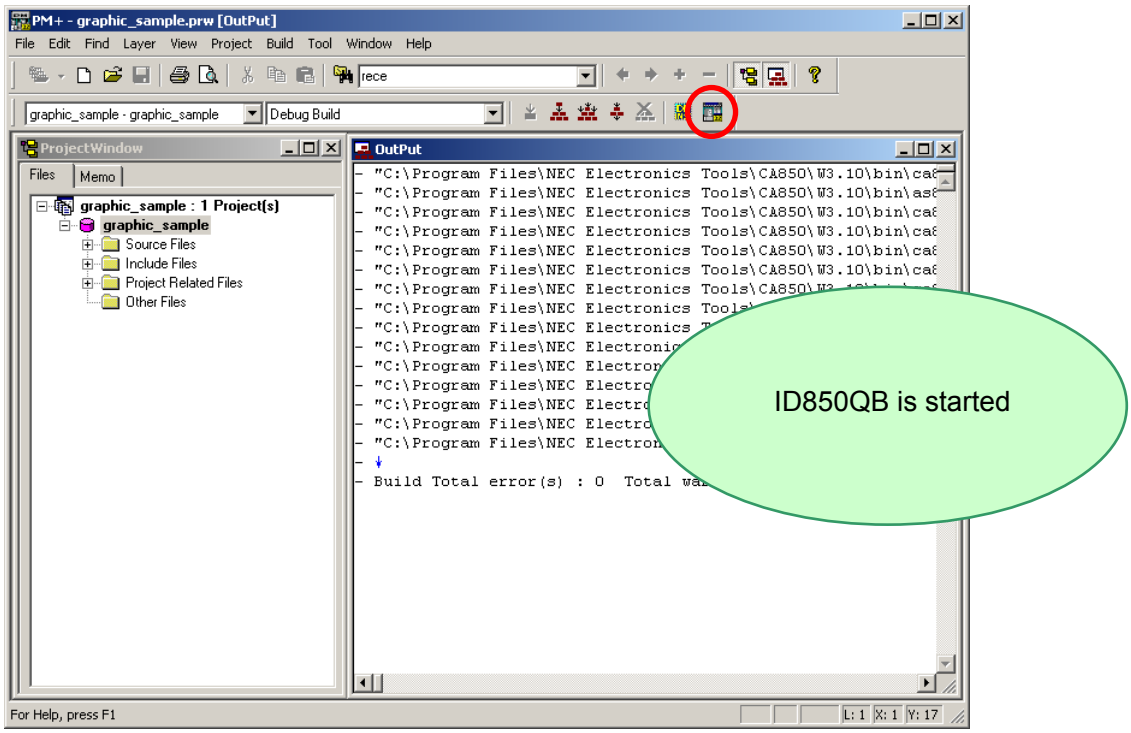
* For USB connections, MINICUBE2 must be connected first. If you connect the others first, it may break the system.
* When you disconnect the USB connections, the order must be (1) EB-850/JG2+TFT (2) MINICUBE2.
* Make sure to turn off the systems before the J4 connection.

2.8 Start Debugger (ID850QB)

Click the debug button  in PM+, or select "Build" on menu bar, then "Debug".

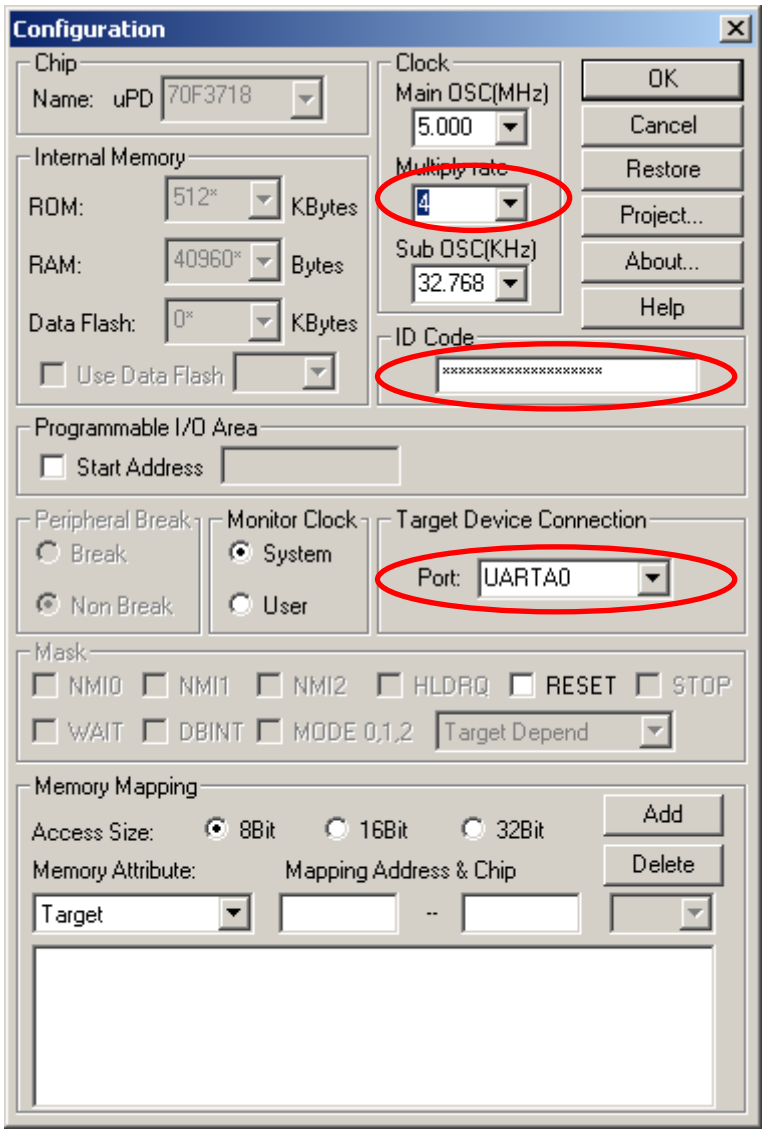
If you do not see the debug button, refer to "2.6 Check Debugger Settings" to configure the settings.

The steps to start the debugger will be explained below.

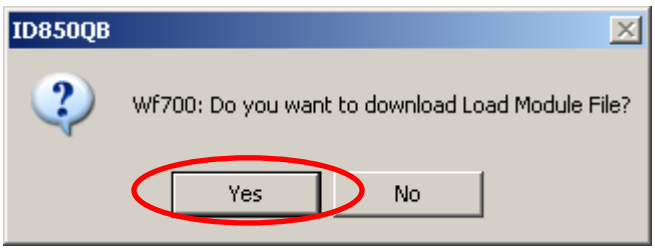


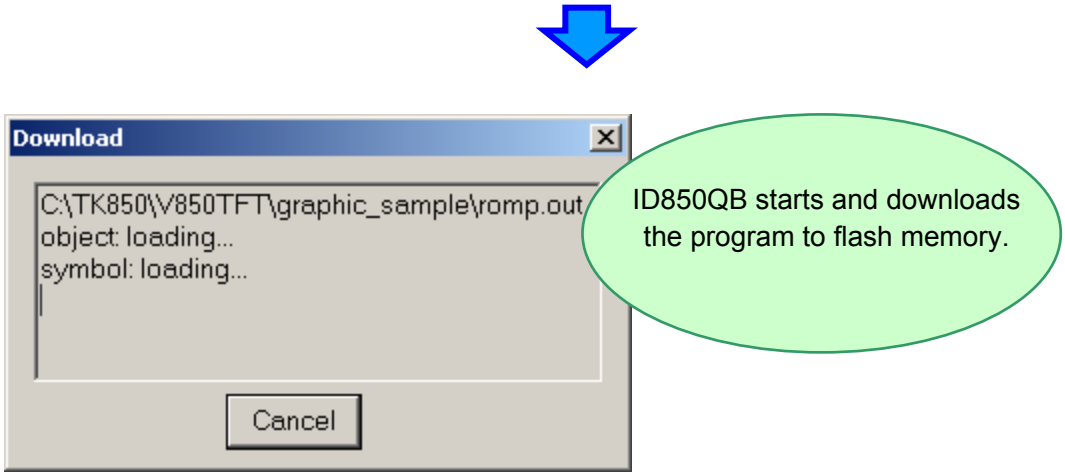
"Configuration" dialog is opened. Follow the settings below and click "OK".

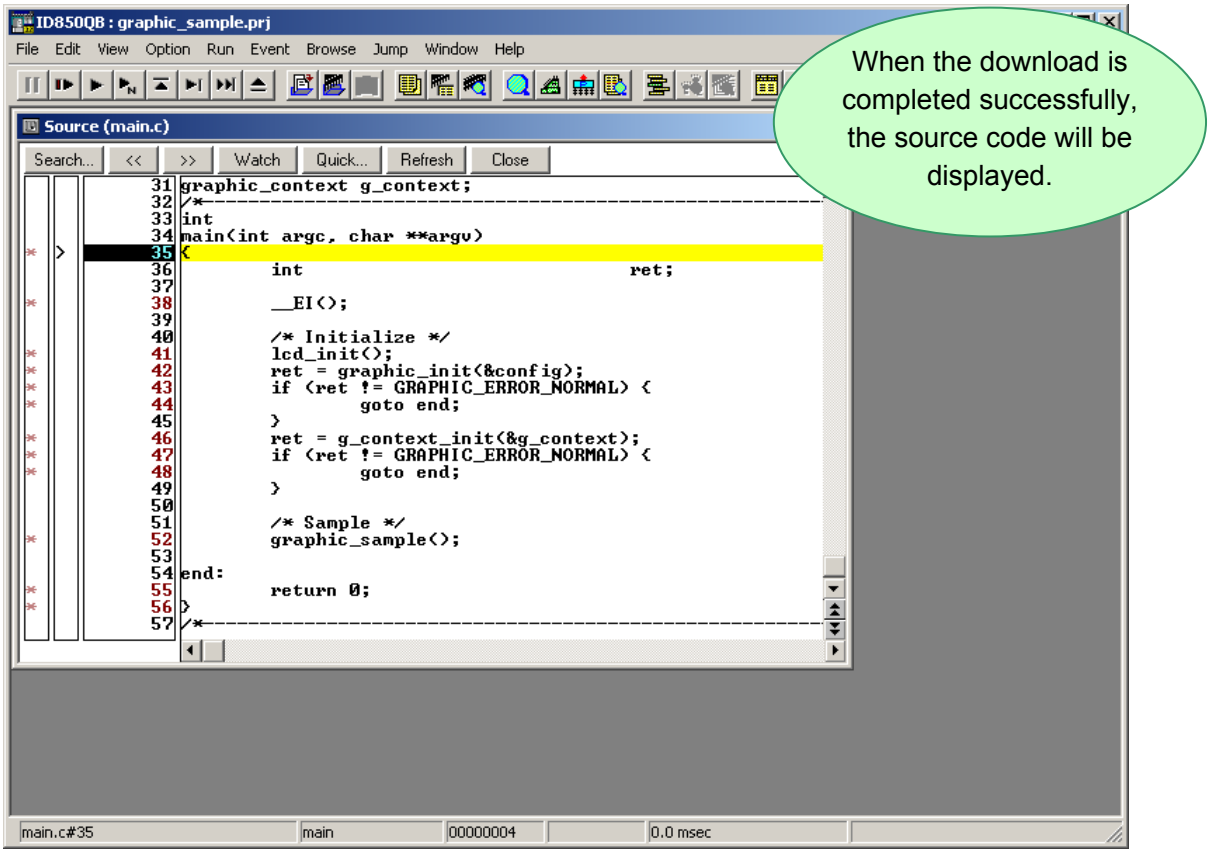
- Select "4" in "Multiply rate"
- Enter "FFFFFFFFFFFFFFFFFFFF" (F x 20) in "ID Code"
- Select "UARTA0" in "Port" at "Target Device Connection" area



Click "Yes" when the confirmation dialog for downloading load module file is opened.







NOTE:

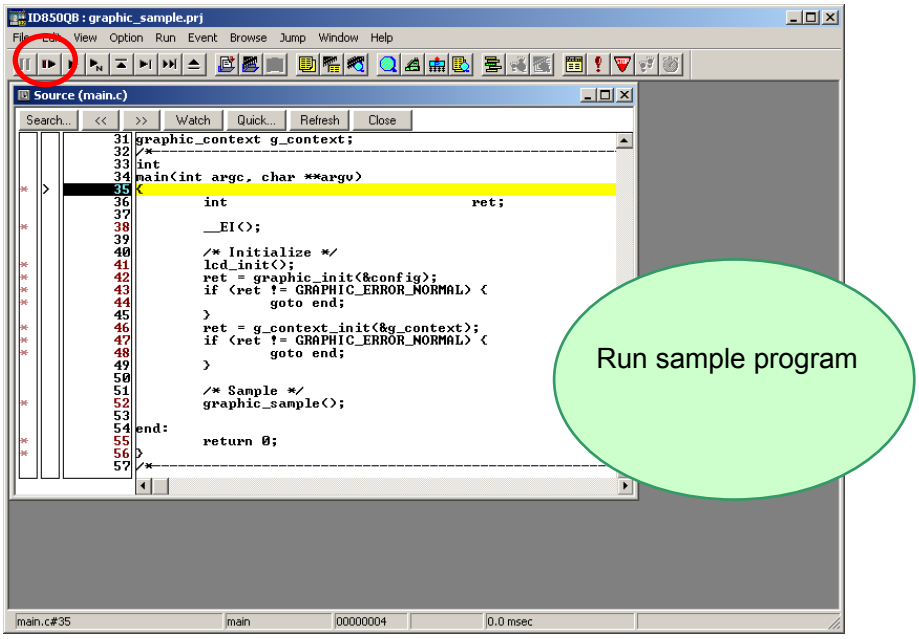
It just completes the download of programs (writing the programs on built-in flash memory). It does not mean that the program is running. Therefore, the LCD on the kit does not display anything. To run the demonstration, move on to the next section "2.9 Run Programs".

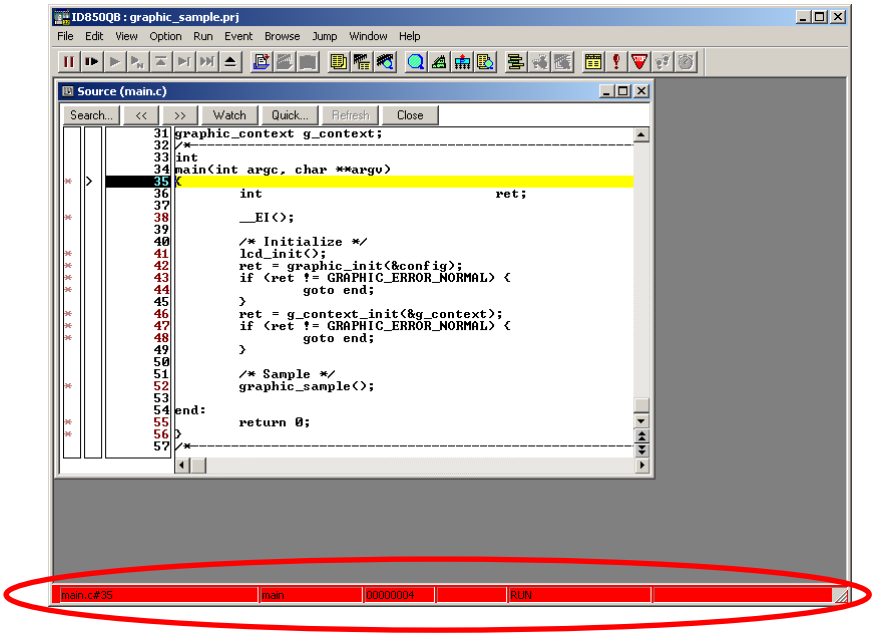
2.9 Run Programs

Now, you are ready to run the program.

Click the restart button  in ID850QB, or select "Run" on menu bar, then "Restart".

The sample program runs.





When programs are running, the status bar will be red.

Check if graphics are displayed on the LCD panel of EB-850/JG2+TFT.

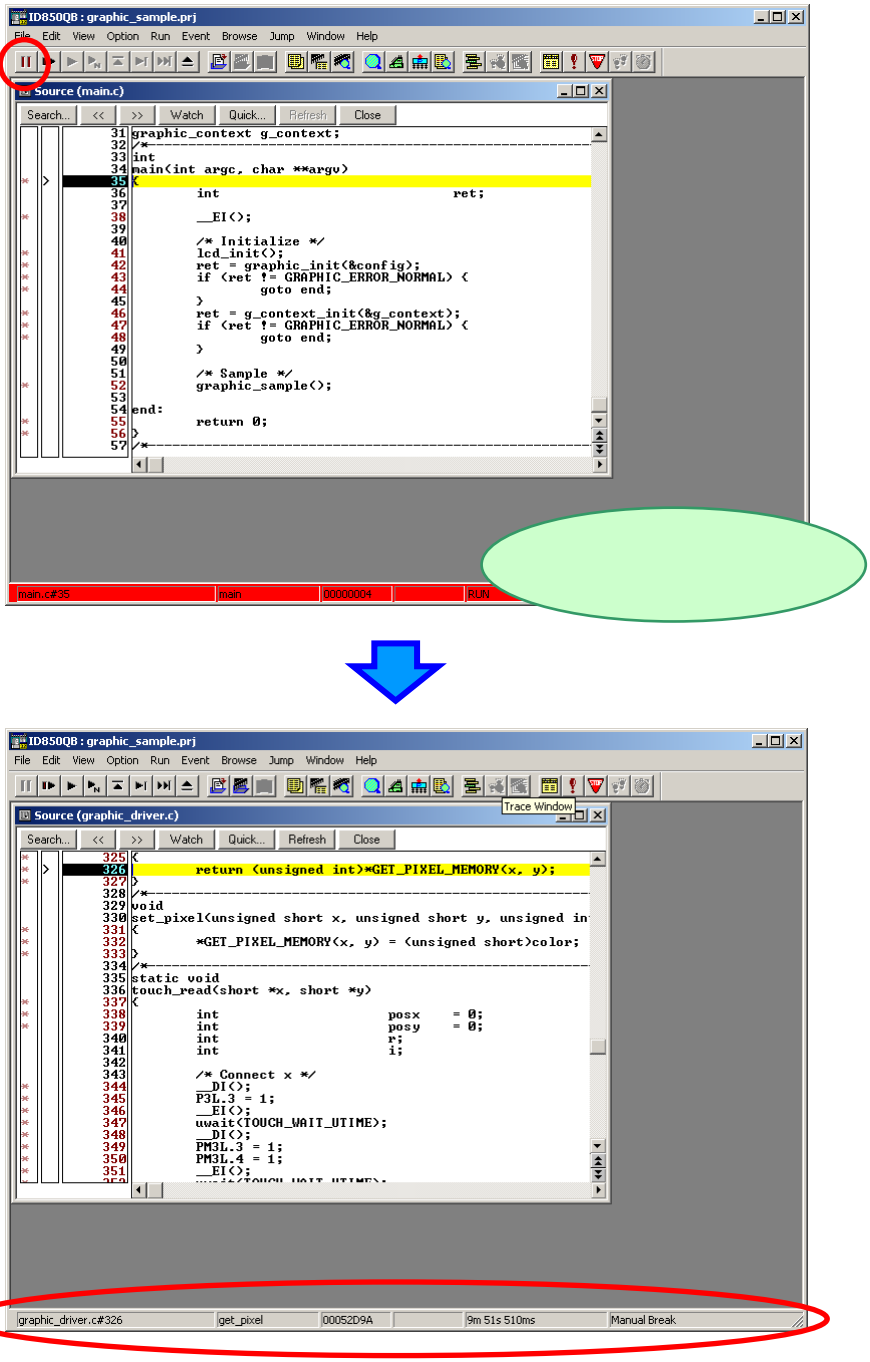
You could confirm the sample program is working.

- Menu can be opened by touching the title bar.
- For more information about the sample programs, refer to "Application Note" document.
- The programs downloaded by ID850QB cannot be used without MINICUBE2 connection. For stand-alone operation, write the HEX file with QB-Programmer.

2.10 Stop Programs

Now, you are going to stop the program.

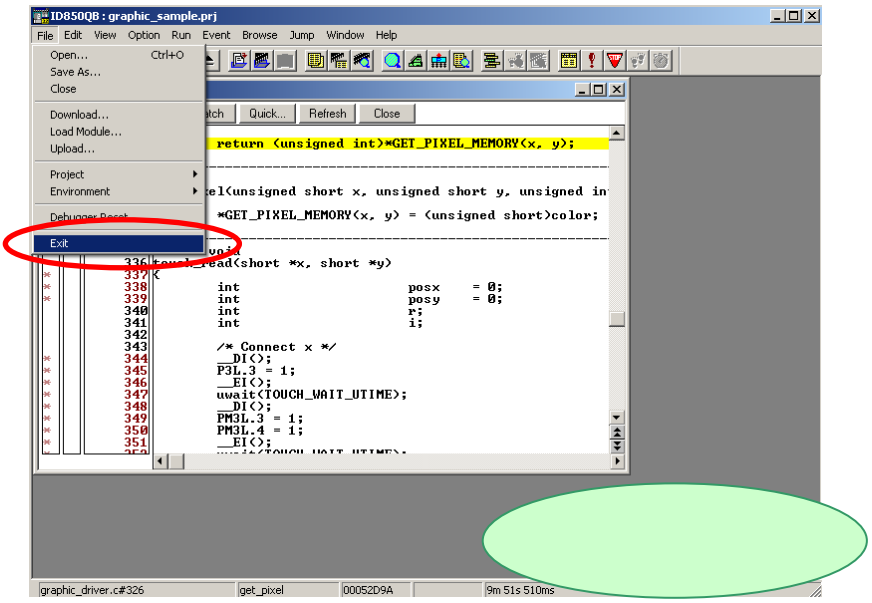
Click the stop button  in ID850QB, or select "Run" on menu bar, then "Stop".

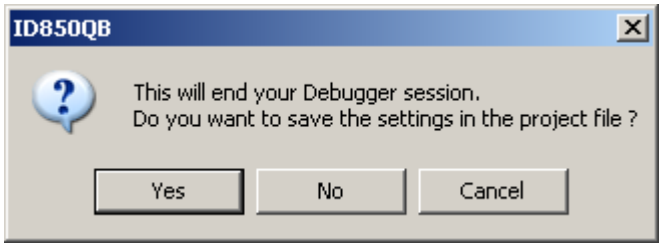


When the program stops, the status bar changes back to the original color.

2.11Close Debugger (ID850QB)

Select "File" on ID850QB menu bar, then "Exit".

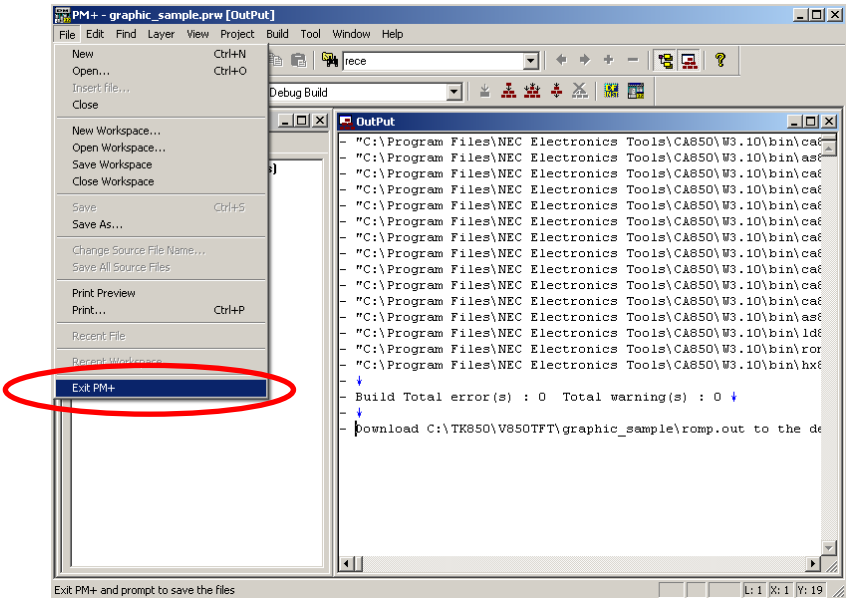




If you click "Yes", it saves the current settings in the project file, and then closes the ID850QB. It is recommended to save the settings as it saves the window you used, window size, layout, etc.
If you click "No", it does not save the current settings and closes the ID850QB.

2.12 Quit PM+

Select "File" on PM+ menu bar, then "Exit PM+".



PM+ is closed.

The experiences section ends here.
You can find more information about basic operations and other useful features in "Chapter 5 Other Information".

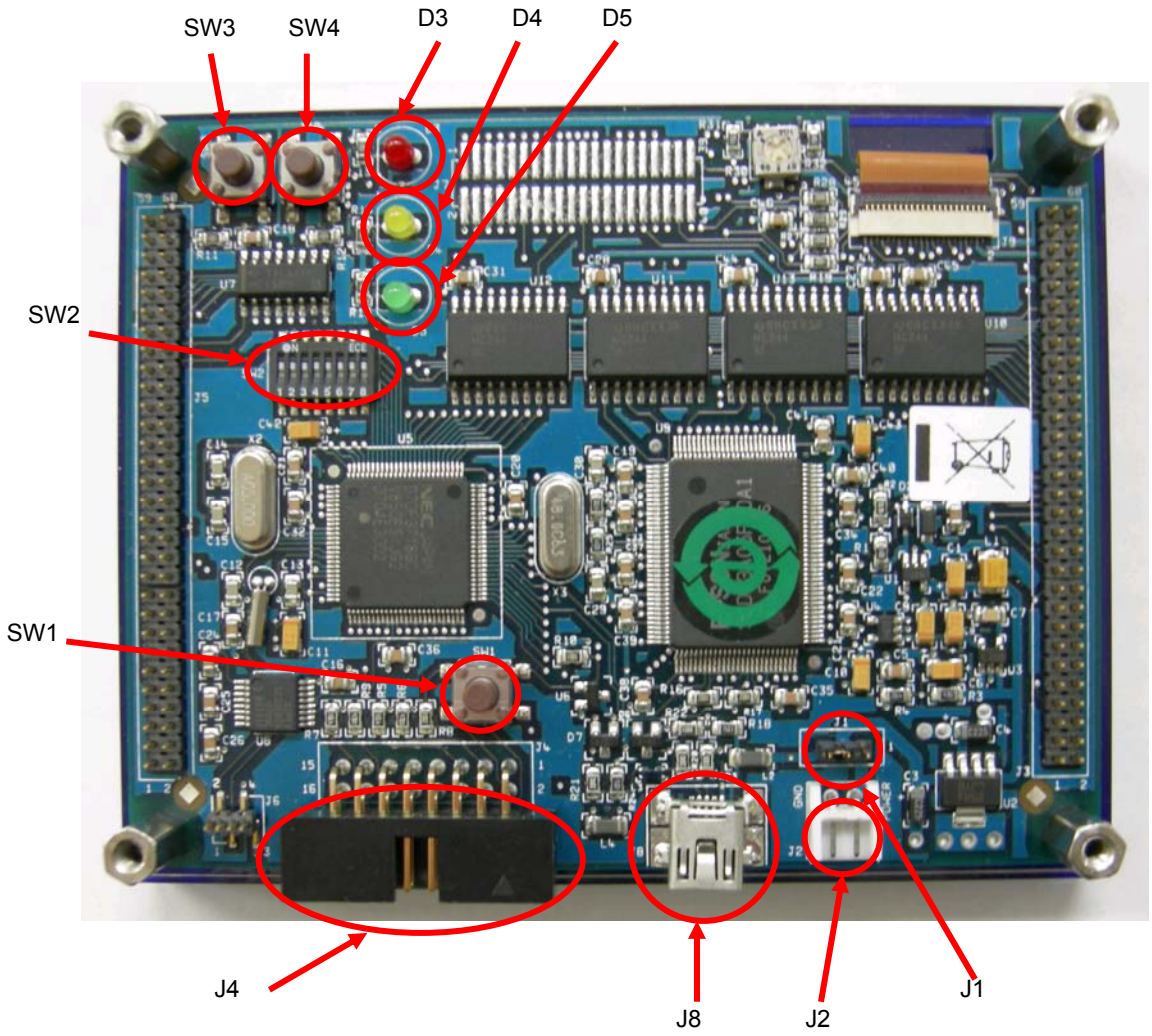
Chapter 3 Hardware Specifications

In this chapter, the hardware of EB-850/JG2+TFTwill be explained.

Microcontroller	µPD70F3718 *V850ES/JG2
Clock	Main system clock: 20MHz (5MHz x 4 multiplying) Sub system clock: 32.768KHz
LCD controller	S1D13A05 (EPSON) External memory bus connection 256Kbyte display built-in memory
LCD panel	NL2432HC22-41K (NEC LC technology) 3.5 inches TFT Touch panel 240x320 dots, 250,000 color
Interface	USB (J8) For power supply (no driver bundled) Connector for MINICUBE2 connection (J4)
Power supply voltage	5V (J8)

* The name with bracket is the name printed on the board.

3.1 Layout of hardware functions



3.2 Hardware Functions

3.2.1 SW1

SW1 is the reset switch. You can reset the micro controller by pressing this switch.

3.2.2 SW2

SW2 is the DIP switches connected to "P75/ANI5"-"P711/ANI11" in CPU. By setting this to ON, it can connect to GND. If you do not use as A/D, set this to ON. As pull-up resistor is not connected, it will not be "High" by setting the switch OFF.
Normally, set it ON.

3.2.3 SW3 (INTP0)

SW3 is the Push switch connected to "P03/INTP0" pin in microcontroller. When you push the switch, it becomes "Low", and when you release the switch, it becomes "High" by pull-up resistor.

3.2.4 SW4 (INTP1)

SW4 is the Push switch connected to "P04/INTP1" pin in microcontroller. When you push the switch, it becomes "Low", and when you release the switch, it becomes "High" by pull-up resistor.

3.2.5 D3

D3 is the power LED. When the power is ON, it lights.

3.2.6 D4

D4 is the LED connected to "P50" pin in microcontroller. When you drive the "P50" to "Low", it lights.

3.2.7 D5

D5 is the LED connected to "P51" pin in microcontroller. When you drive the "P51" to "Low", it lights.

3.2.8 J1

J1 is the jumper switch pin to select power supply.

1-2 Short	Use power supply from J2 connector
2-3 Short	Use USB power supply from J8 (USB) connector

3.2.9 J2

J2 is the connector for external power supply. Supply power with 5V.

3.2.10 J4

J4 is the connector for MINICUBE2 connection.

3.2.11 J8

J8 is the connector for USB connection. It is connected to USB pin from LCD driver (S1D13A05), but as this kit does not bundle USB driver, use it for power supply purpose.

Chapter 4 Troubleshooting

This chapter describes how to solve troubles you may face.

4.1 If you cannot find USB driver when you connect MINICUBE2 and PC

Check Point 1

If you use USB hub, do not use it.

Check Point 2

Check if you installed "MINICUBE2 USB Driver" in "1.2 Installation of Development Tools". If not, install the driver.

4.2 Error when you start the debugger

There could be several reasons to make this error happened.
The solving processes differ depending on errors. Please check the error message first.
The solving processes for each error are described below.

4.2.1 "Can not communicate with ICE..." (F0100 or A0109)

Check Point 1

If you use USB hub, do not use it. (USB hub is not supported)

Check Point 2

If above check point is not the problem, close the debugger and disconnect the USB cable from PC. Re-connect USB cable properly to both the PC and MINICUBE2, and then re-start the debugger.

4.2.2 "Incorrect ID Code." (Ff603)

This error occurs when the security ID stored in microcontroller built-in flash memory is different from the ID code you entered at the start of debugger.

Security ID entry area at the start of debugger

Check Point 1

Enter correct security ID and click OK on the configuration window.

Check Point 2

If you forgot the security ID, you need to erase the microcontroller built-in flash memory. Before erasing it, check if you actually set the security ID. Also remember the code you set for the security ID.
After this, erase the flash memory with referring to "5.4 QB-Programmer".

4.2.3 "The on-chip debug function had been disabled in the device." (F0c79)

This error occurs when the value at address C3H (On-chip debug option byte) in microcontroller built-in flash memory is incorrect. You need to erase the flash memory.

Check Point 1

Check if you actually set the correct on-chip debug option byte. If it is not correct, set it correctly.

Check Point 2

Erase the flash memory with referring to "5.4 QB-Programmer".

4.2.4 "Disabling the on-chip debug function is prohibited." (F0c33)

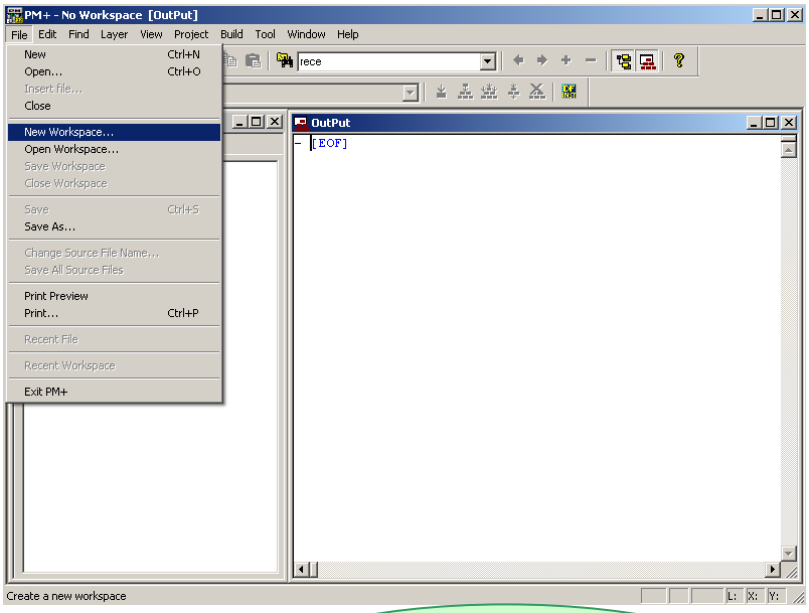
Basically, this error occurs when you start (download) the debugger without doing the settings correctly. Do the same checking processes as "4.2.3 The on-chip debug function had been disabled in the device. (F0c79)".

Chapter 5 Other Information

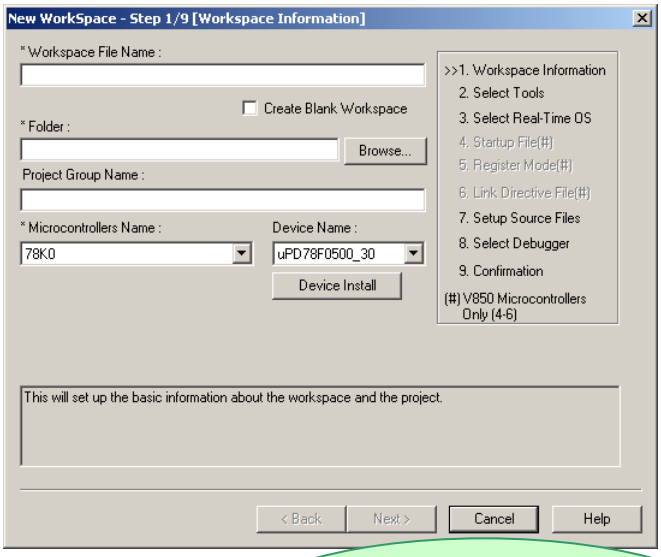
This chapter explains about some important and useful operation techniques of development tools, and circuit diagram of the kit for developing of user programs.

5.1 Create a New Workspace

Now, create a new workspace and project.
PM+ allows you to create a new project including all the information needed for build by following the step-by-step dialog.
Select "File" → "New Workspace..." in the menu bar of PM+.



The dialog box for creating new workspace is displayed



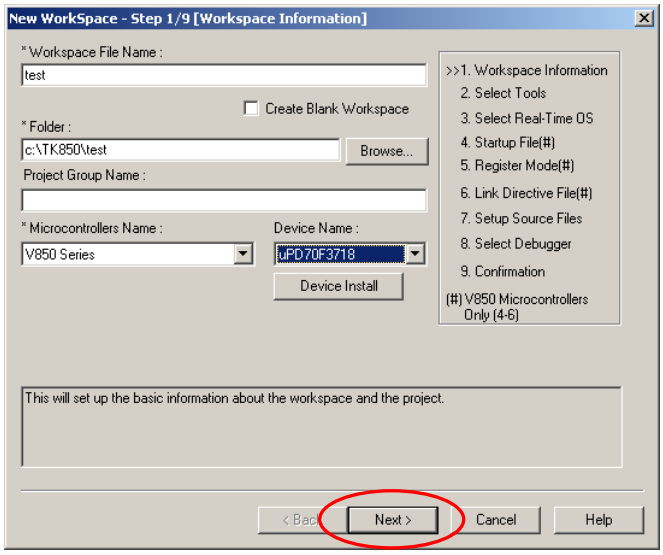
The concrete information set here is described on the following pages

<Description of items>

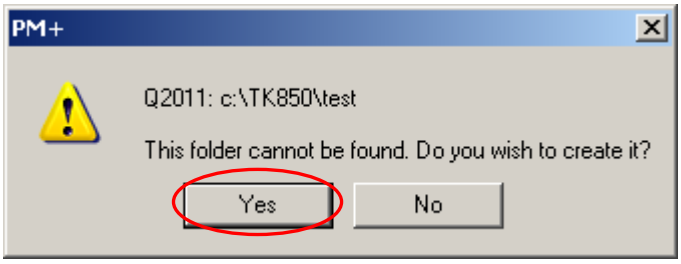
- Workspace File Name:
  Specify the name of the workspace file that manages the project files.
  ".prw" is automatically suffixed as the file type.
  A project file (.prj) of the same name is simultaneously created.
- Folder:
  Specify the folder for saving the workspace file by entering its absolute path.
  This item can be selected from a reference dialog box by clicking the "Browse…" button.
- Project Group Name:
  Specify this item if wishing to manage multiple projects together in function units.
  If nothing is specified, this item is the same as the workspace file name.
- Series Name:
  Specify the series name of the device to be used.
- Device Name:
  Specify the name of the device to be used.

Input the workspace information setting as follows.

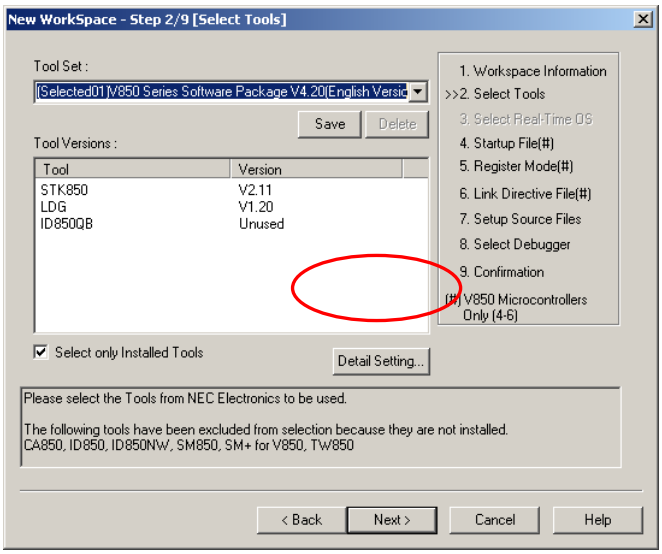
- Workspace file name
  → test
- Folder
  → C:¥TK850¥test
- Project Group Name
  → Don't input this item.
- Series Name
  → V850 Series
- Device Name
  → uPD70F3718



Click "Next" button.



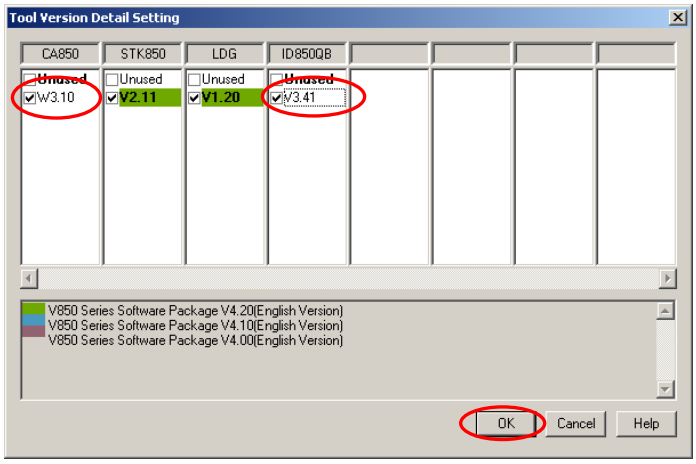
Click "Yes".



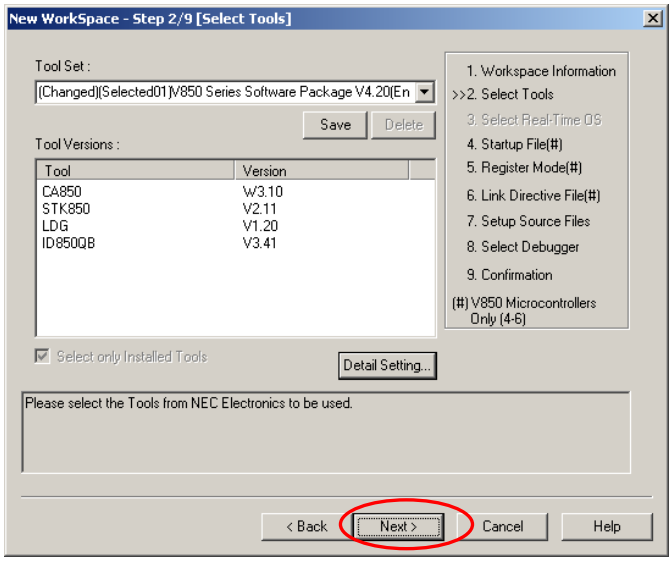
Click "Detail Settings" button.

Set the version of tools as follows.

CA850: W3.10
ID850QB: V3.41

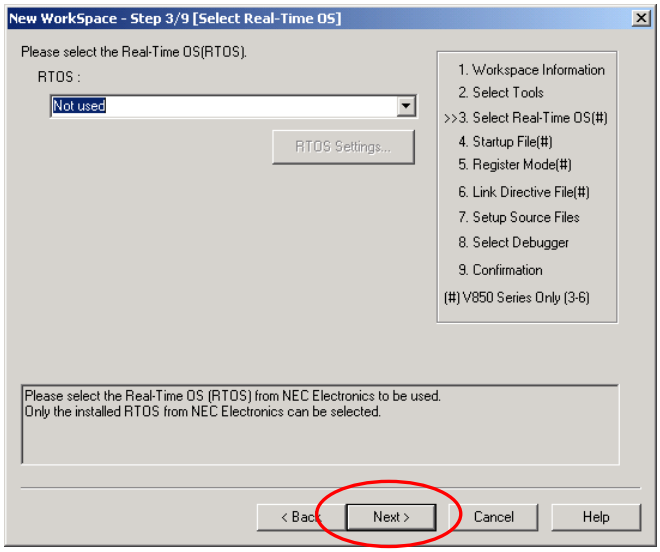


Click "OK".

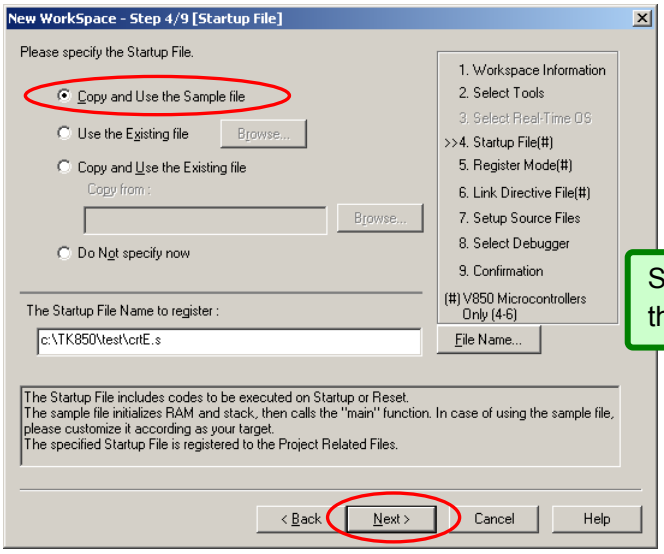


Click "Next".

We do not use RTOS.

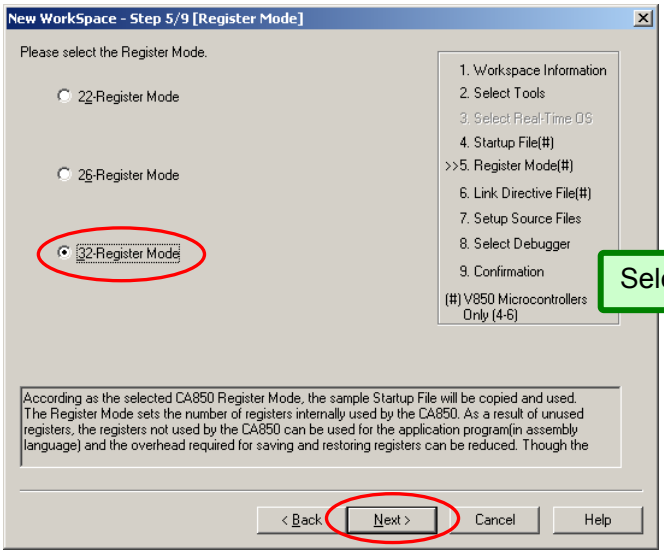


Click "Next".



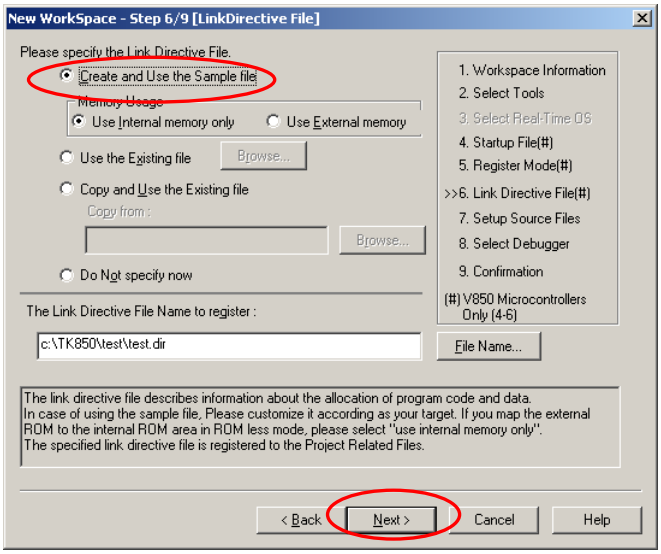
Select "Copy and use the sample"

Click "Next".



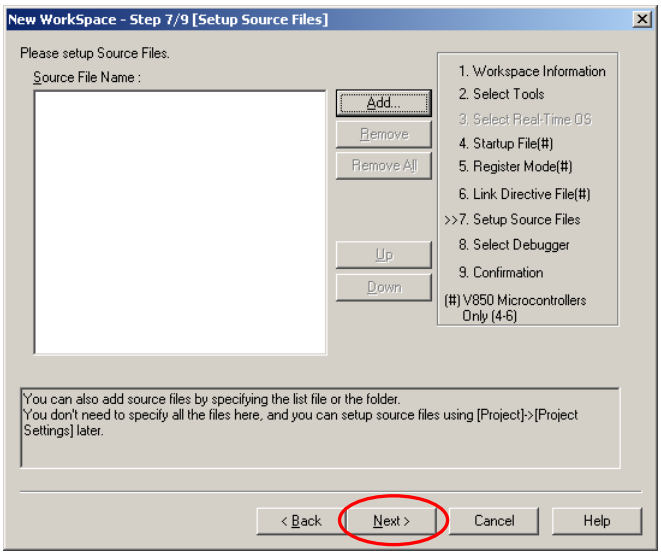
Select "32 register mode"

Click "Next".

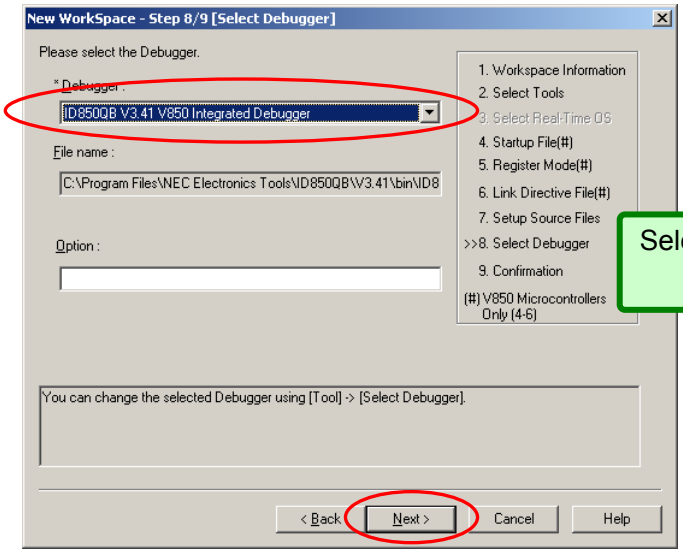


Select "Copy and use the sample"

Click "Next".

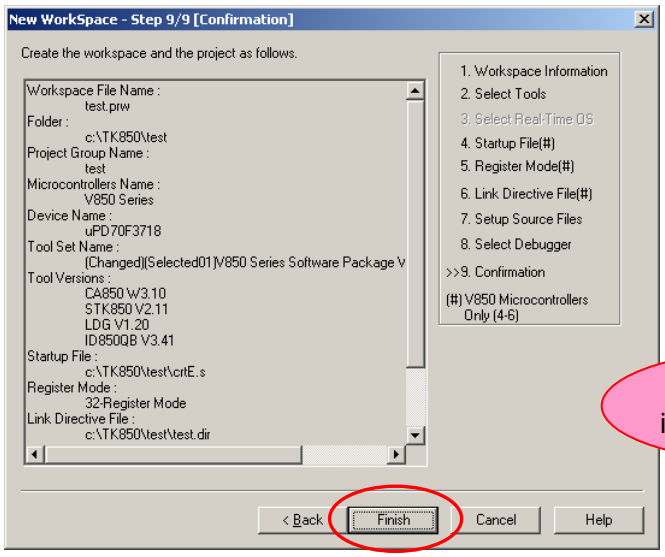


Click "Next".



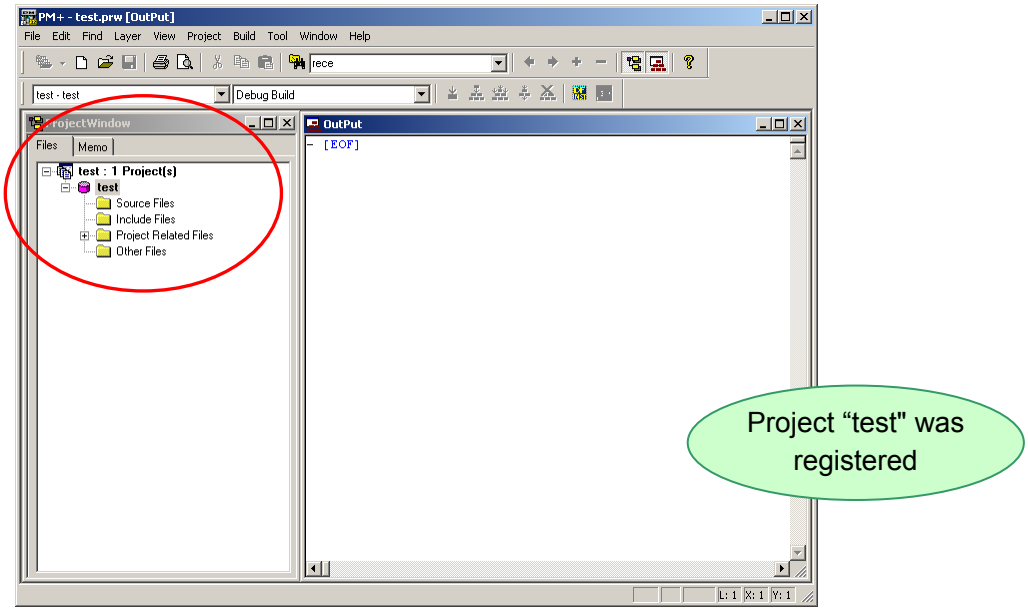
Select "ID850QB V3.41"

Click "Next".



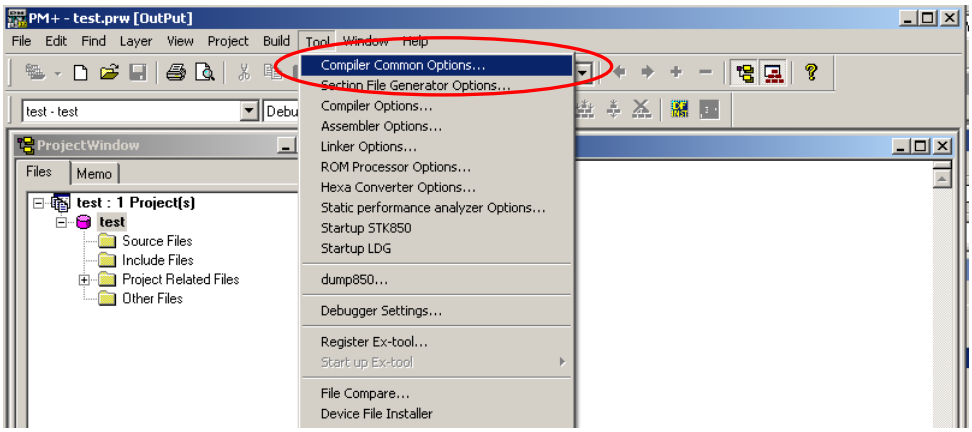
Check the project information settings

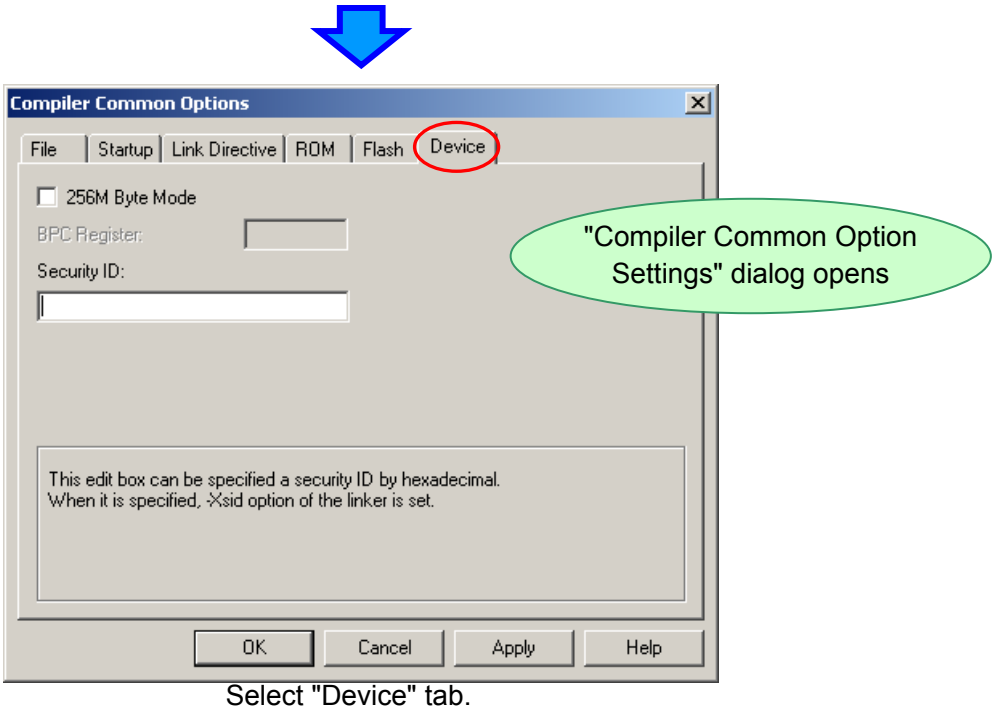
Click "Finish".



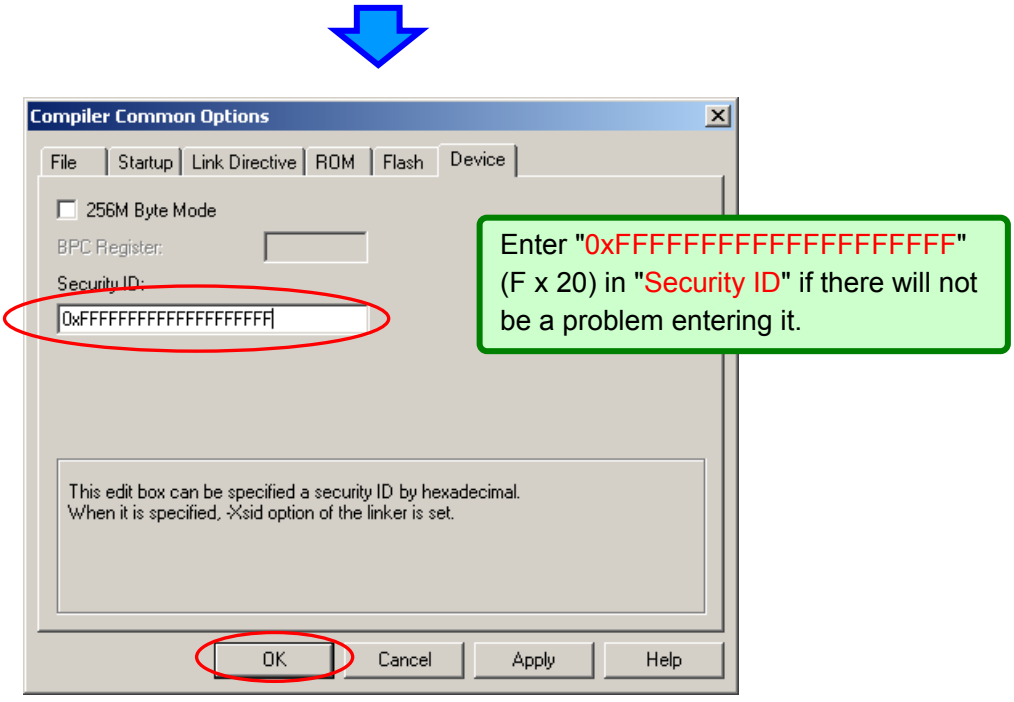
This completes workspace and project creation.
Additional source files can be registered at any time thereafter.
➪ For details, refer to "5.2 Register Additional Source File".

Next, you are going to set the security ID.
Select "Tool" on menu bar, then "Compiler Common Options...".





Select "Device" tab.

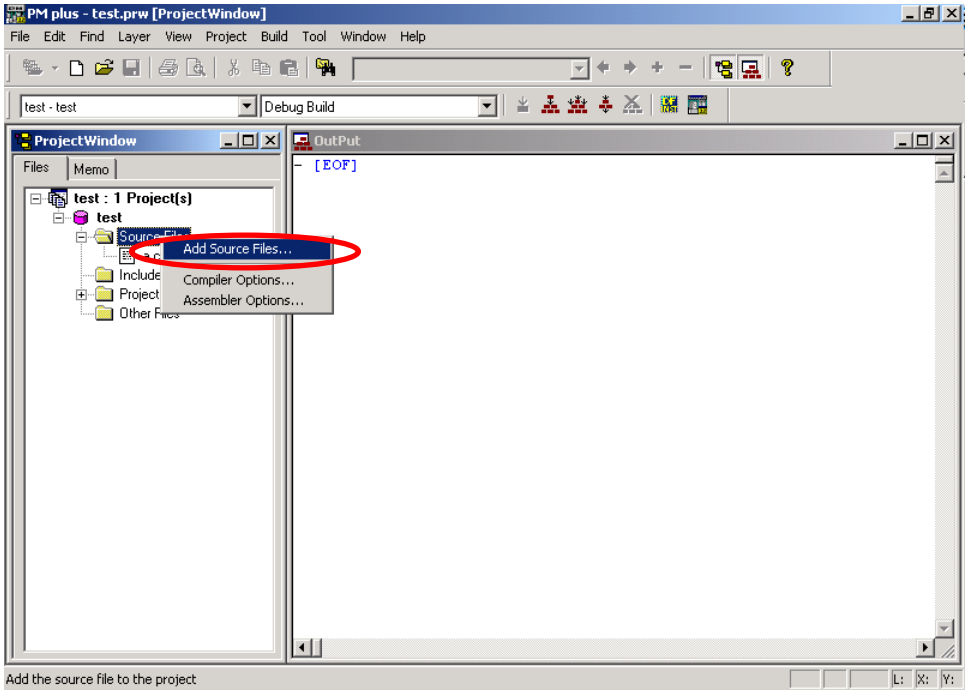


Click "OK".

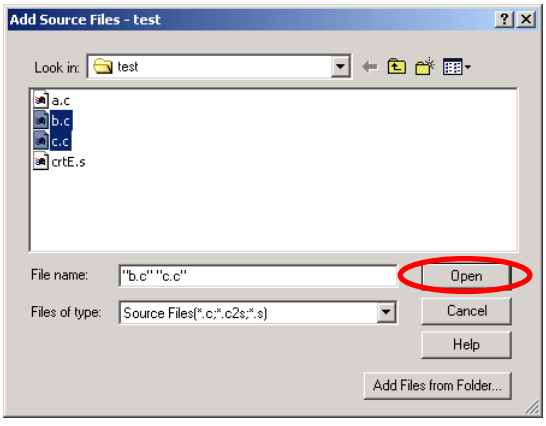
This completes the setting of the security ID.

5.2 Register Additional Source File

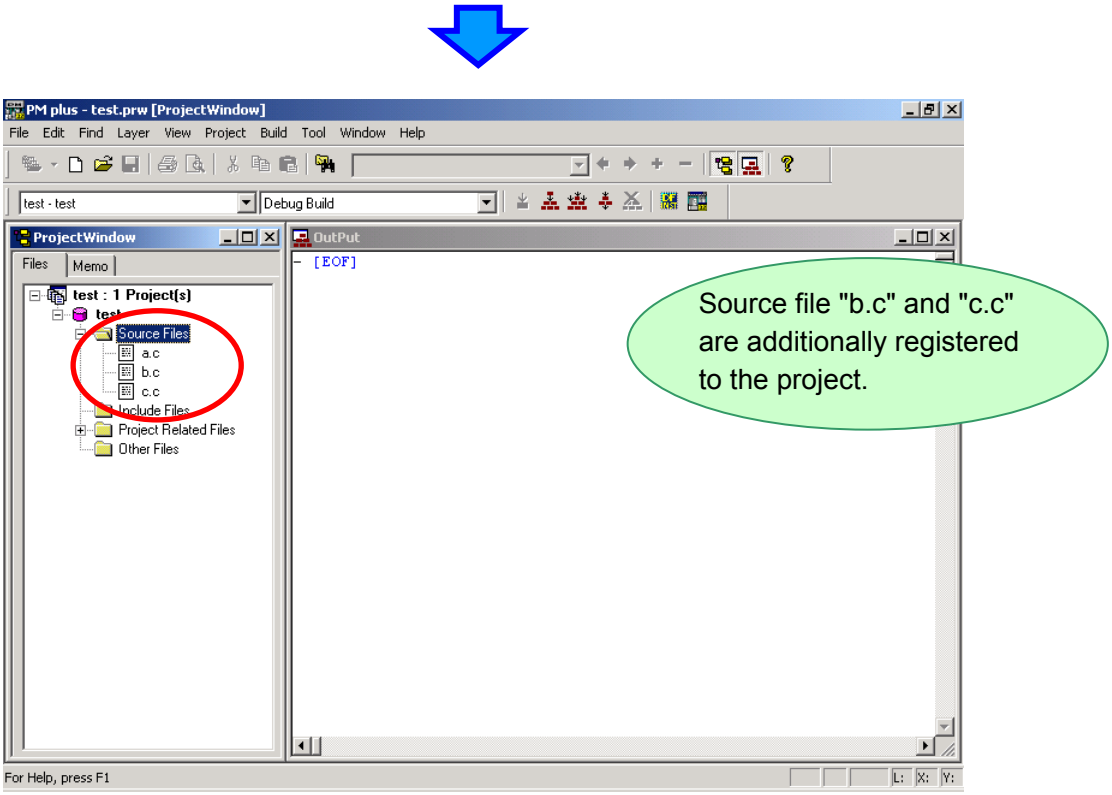
This section explains how to register additional source files to a project.
The following example shows that the additional source files "b.c" and "c.c" are registered to the project which has the source file "a.c".
Place the cursor on the source file in the PM+ project window, and select "Add Source Files…" displayed in the right-click menu.



Select source files "b.c" and "c.c" and then press the Open button.



Multiple source files can be selected by clicking the desired source files while pressing Ctrl key.


PM plus - test.prw [ProjectWindow]
File Edit Find Layer View Project Build Tool Window Help
test - test
Debug Build
ProjectWindow
Files
Memo
test : 1 Project(s)
test
Source Files
a.c
b.c
c.c
Include Files
Project Related Files
Other Files
OutPut
[EOF]
For Help, press F1
Source file "b.c" and "c.c" are additionally registered to the project.

5.3 Debugger Useful Tips

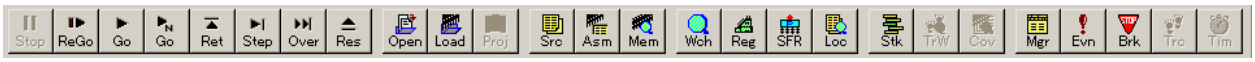
This section describes some useful techniques for the debugger (ID850QB).

5.3.1 Change Display of Buttons

Execution controls (run, stop, step-in debugging, reset, etc) and opening functional window can be made by below buttons. However, it could be difficult to know which button does what.

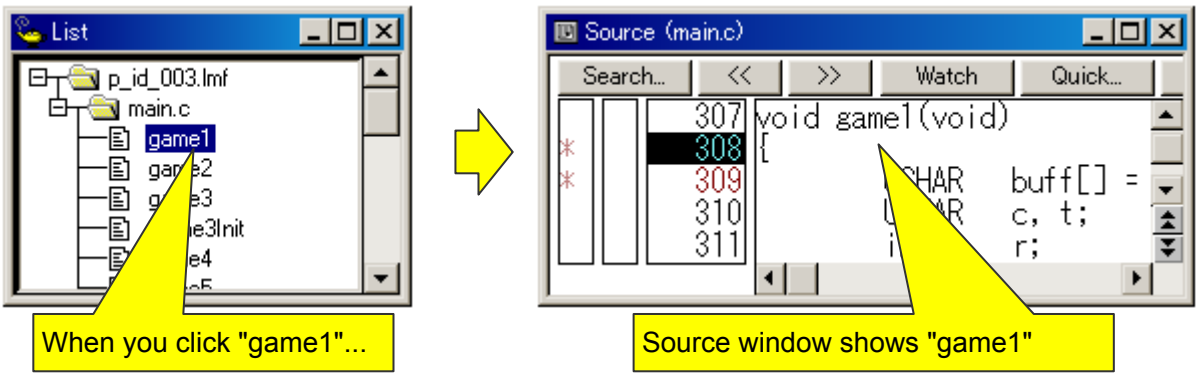
In this case, select "Option" on menu bar, then "Debugger Option". Check "Pictures and Text" on "Tool Bar Picture" setting area.

With this setting, the buttons display the text as well, so that it is easier to know what they are.



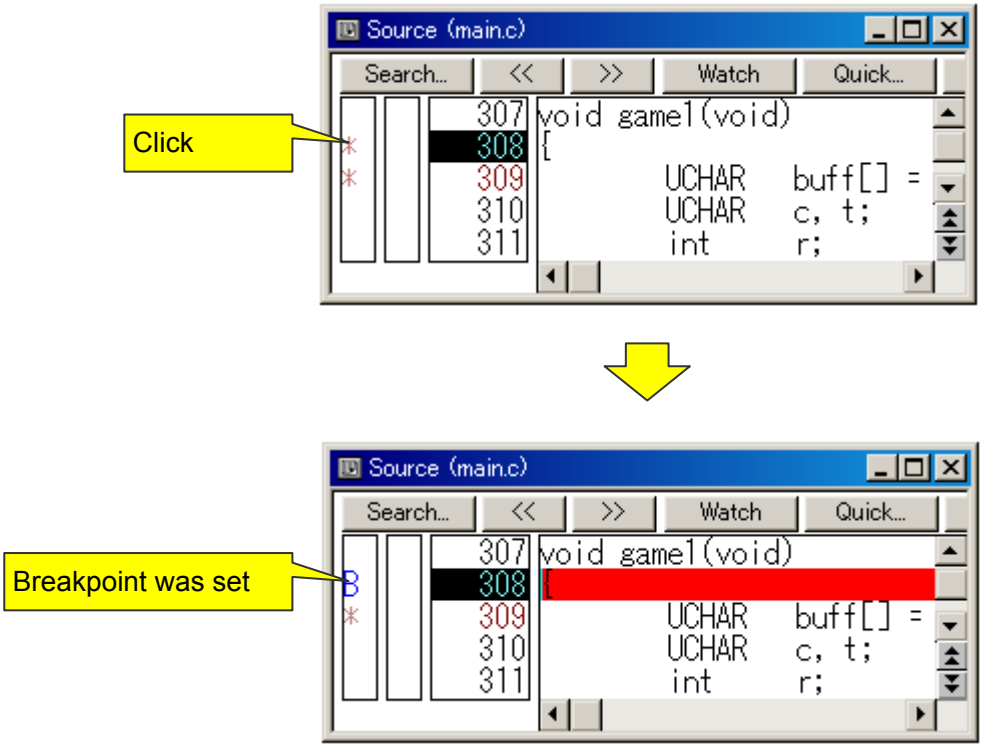
5.3.2 Display Source List and Function List

When you wish to see source file list or function list, select "Browse" on menu bar, then "Other" -> "List" to open the list window. The information in the windows is synchronized with the source window. Therefore, it is not just for referring to the list, but it is useful when you wish to update files or functions.



5.3.3 Set/Remove Breakpoints

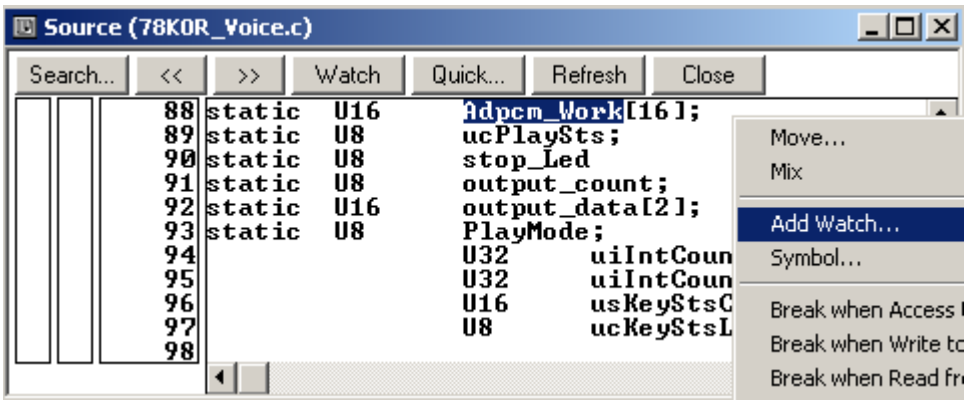
Breakpoints are set or removed by clicking lines in which " * " is displayed
It changes to "B" when you click "*" (the breakpoint is set).
It changes to "*" when you click "B" (the breakpoint is canceled).



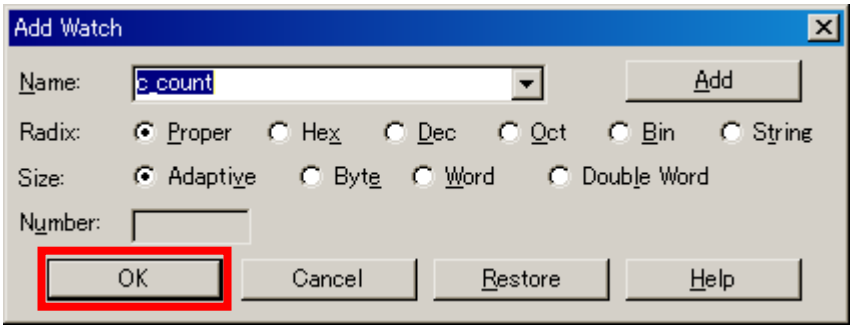
5.3.4 Display Global Variables

With using Watch window, you can display global variables. There are several ways to register global variables to Watch window. In this section, how to register global variables from source window is described.

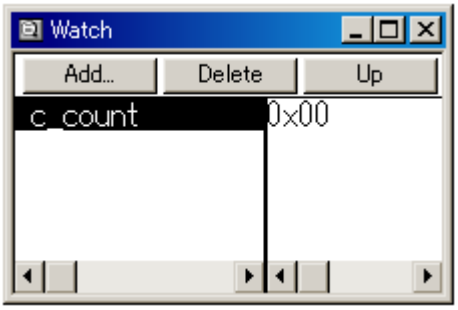
1) Right-click the variable on Source window, then select "Add Watch..."



2) Add Watch dialog opens. Click "OK".



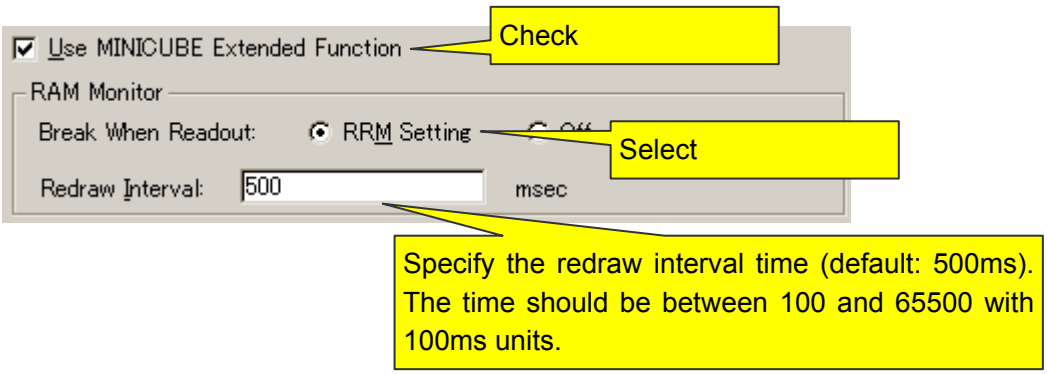
3) Adding a variable to watch window is completed



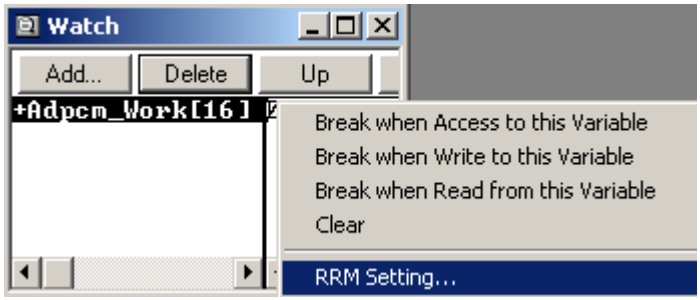
5.3.5 Display Global Variables While Programs Are Running

Global variable can be referred even when the programs are running.

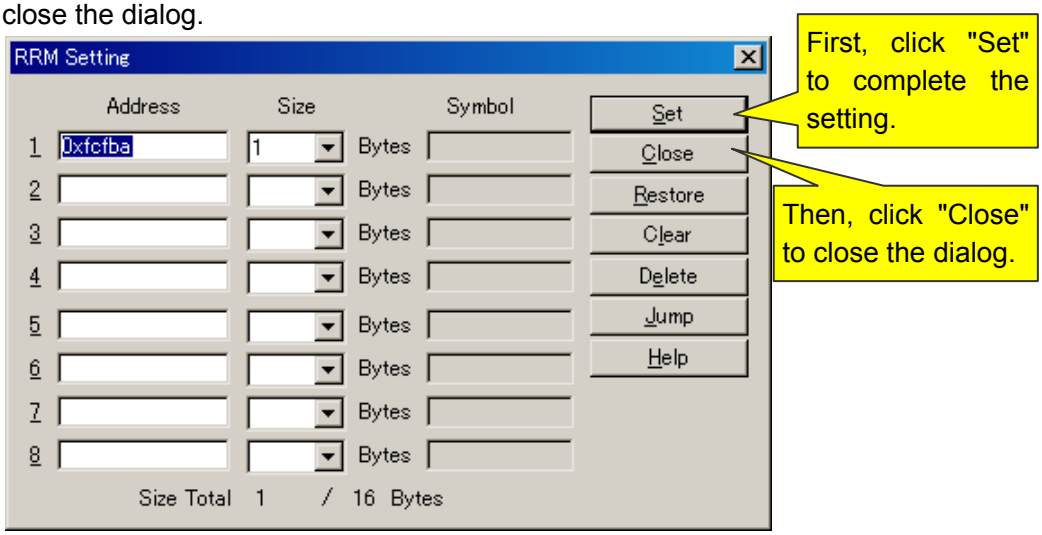
1) Select "Option" menu -> "Extended Option...". Follow below settings in the "RAM Monitor" area.



2) Right-click the variable in Watch window while programs are running, then select "RRM Setting...".



3) RRM Setting dialog opens. Click "Set" button to complete the setting, then click "Close" button to close the dialog.

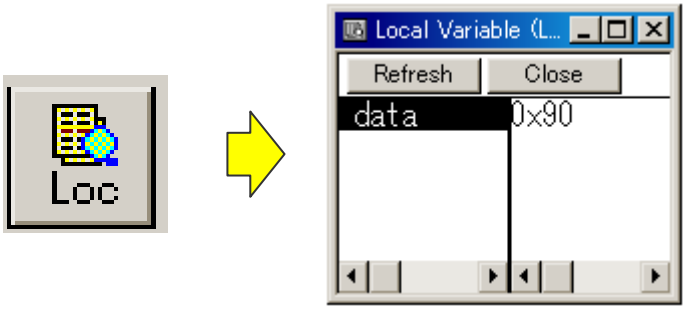


This completes the settings. This is a useful function, but there are some notes for this function.

- The maximum size of variable area is 16byte in total when programs are running.
- The maximum number of variable areas is 8 areas when programs are running.
- Program momentarily breaks upon a read of variables.

5.3.6 Display Local Variables

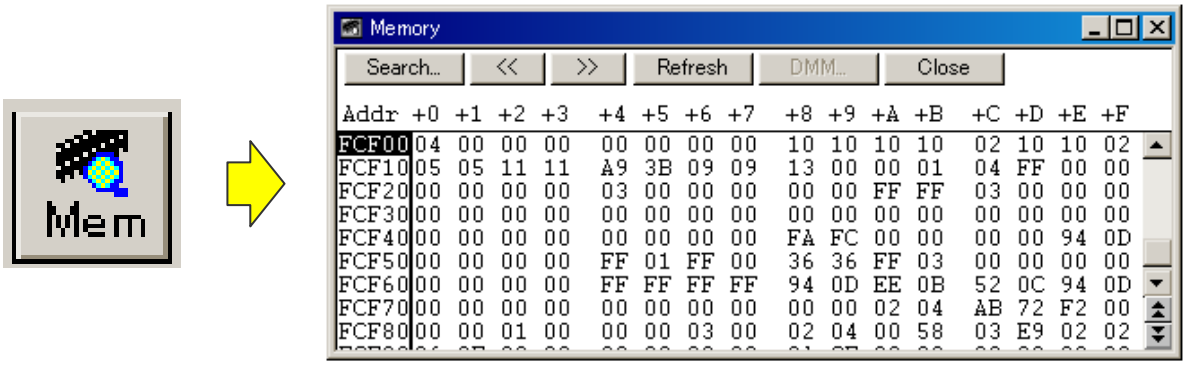
Local variable window is used to display local variables.
By clicking the button below, you can open the local variable window.
Unlike global variables, local variables cannot be displayed when programs are running.



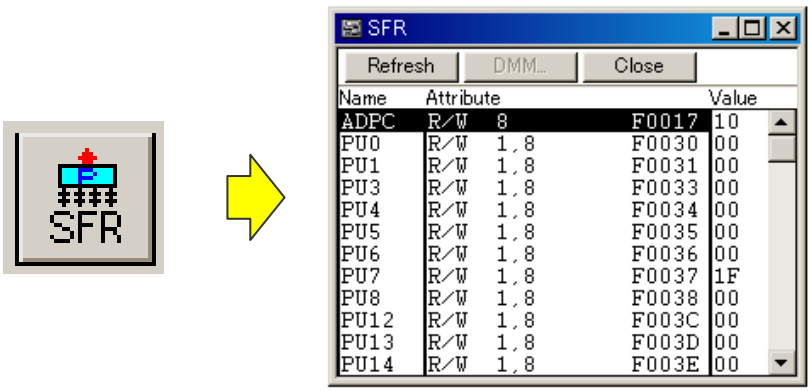
5.3.7 Display Memory and SFR Contents

Similar to the local variable window, you can display memory and SFR contents.

- By clicking the button below, you can open the memory window.



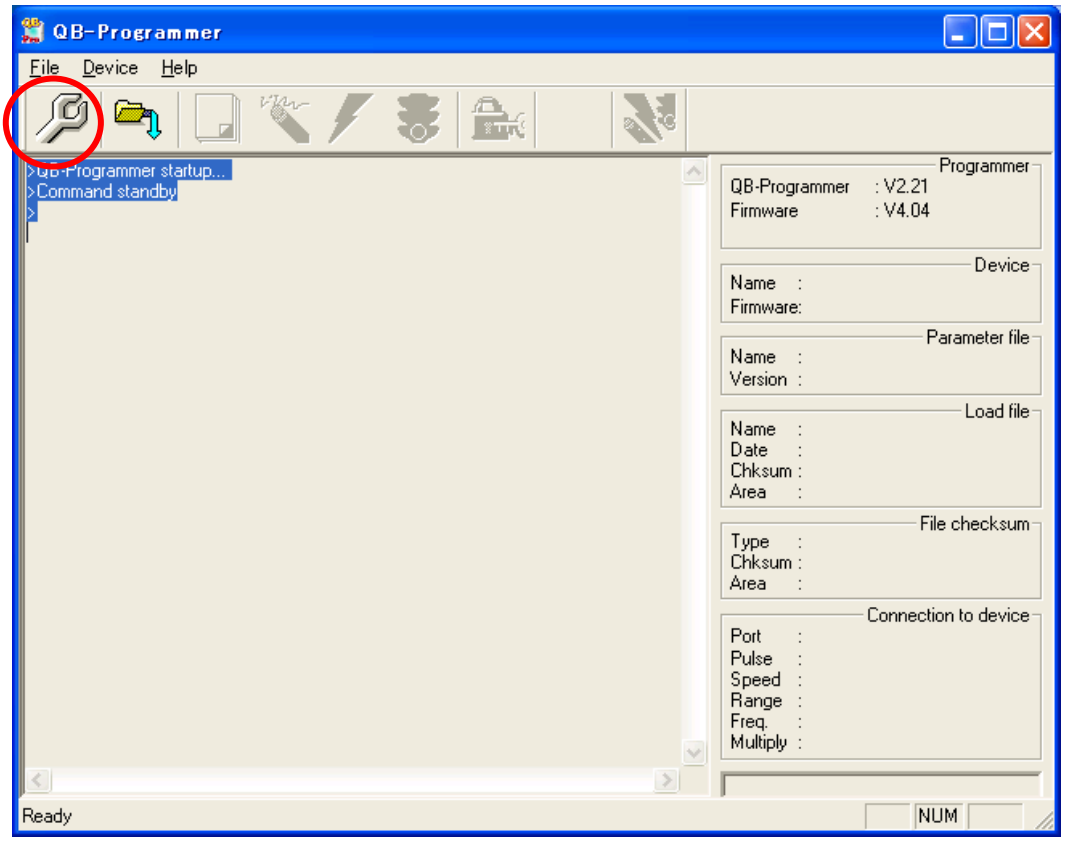
- By clicking the button below, you can open the SFR window.



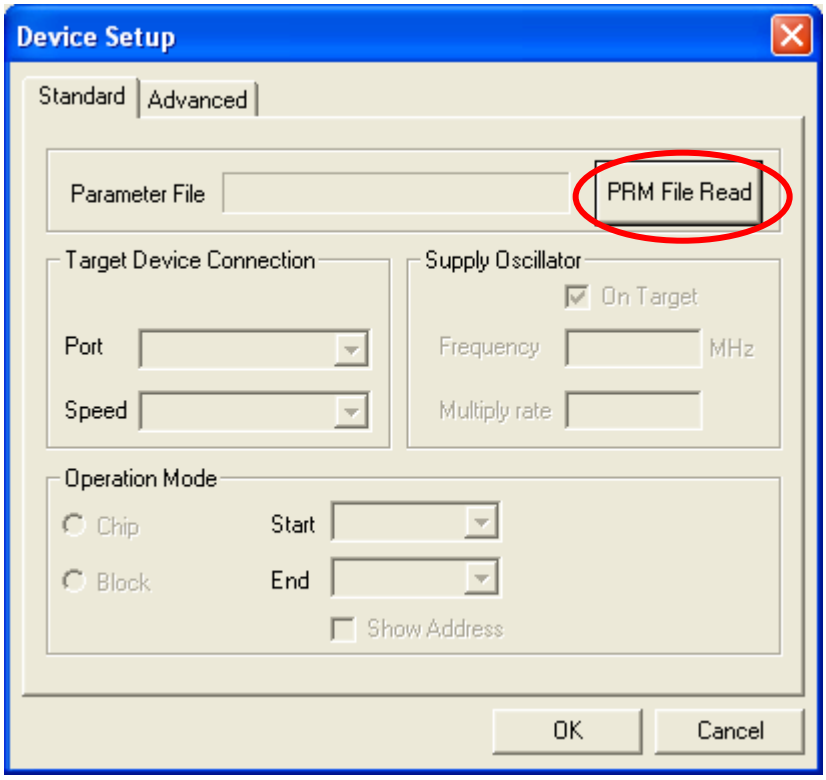
5.4 QB-Programmer

When you forget the security ID or overwrite unintended value, you can erase the flash memory by using QB-Programmer. By erasing the flash memory, you can set the security ID to "FFFFFFFFFFFFFFFFFFFF".
Also, you can use QB-Programmer when you wish to run programs without the MINICUBE2 connection (stand-alone) after completing debugging.
As ID850QB downloads monitoring files for debugging, the programs cannot be run with stand-alone. QB-Programmer does not write monitoring files, so that the programs can be run with stand-alone.

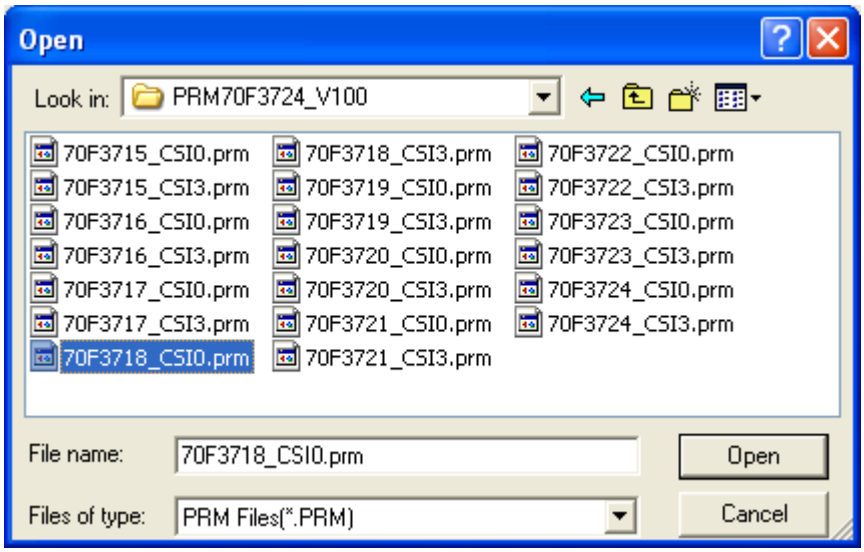
1) Start QB-Programmer.



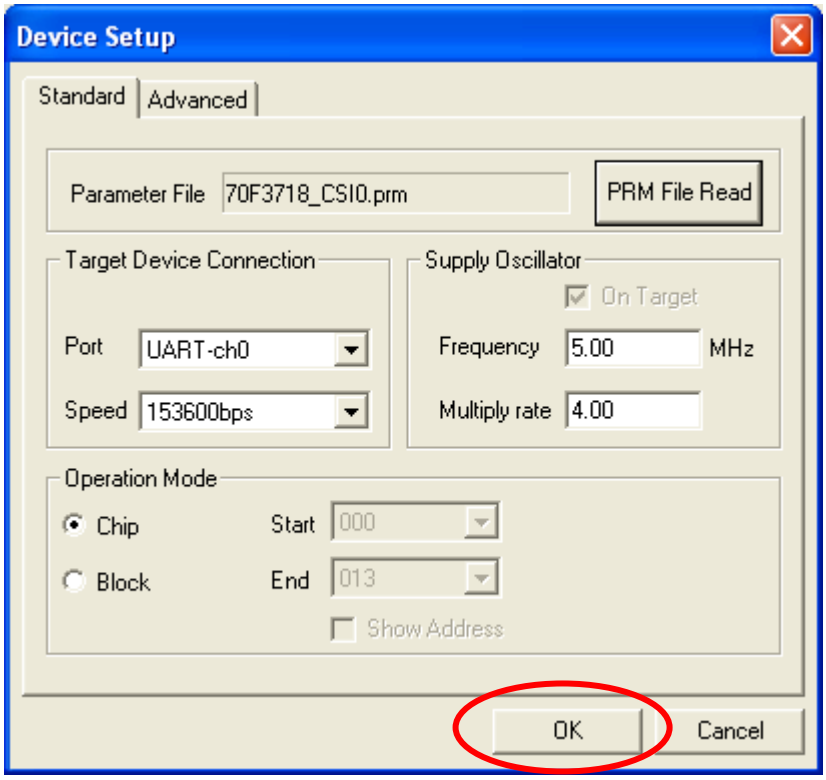
2) Click "Setup" button.



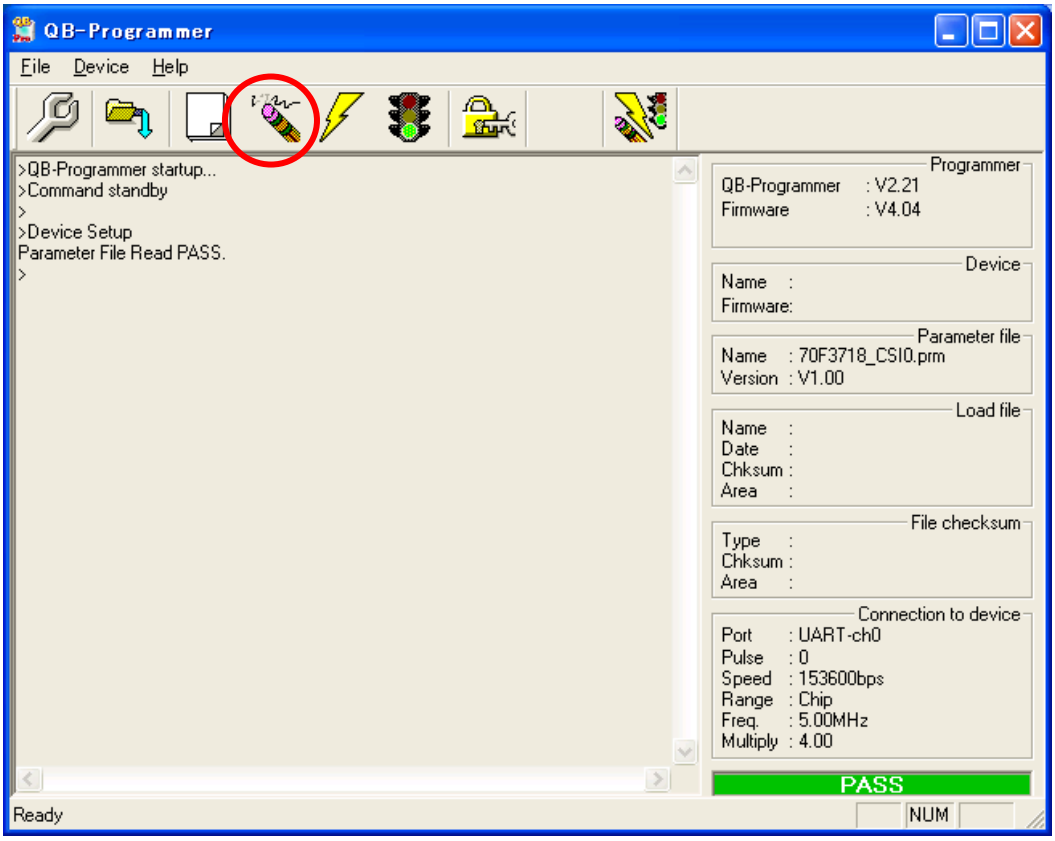
3) Click "PRM File Read" button.



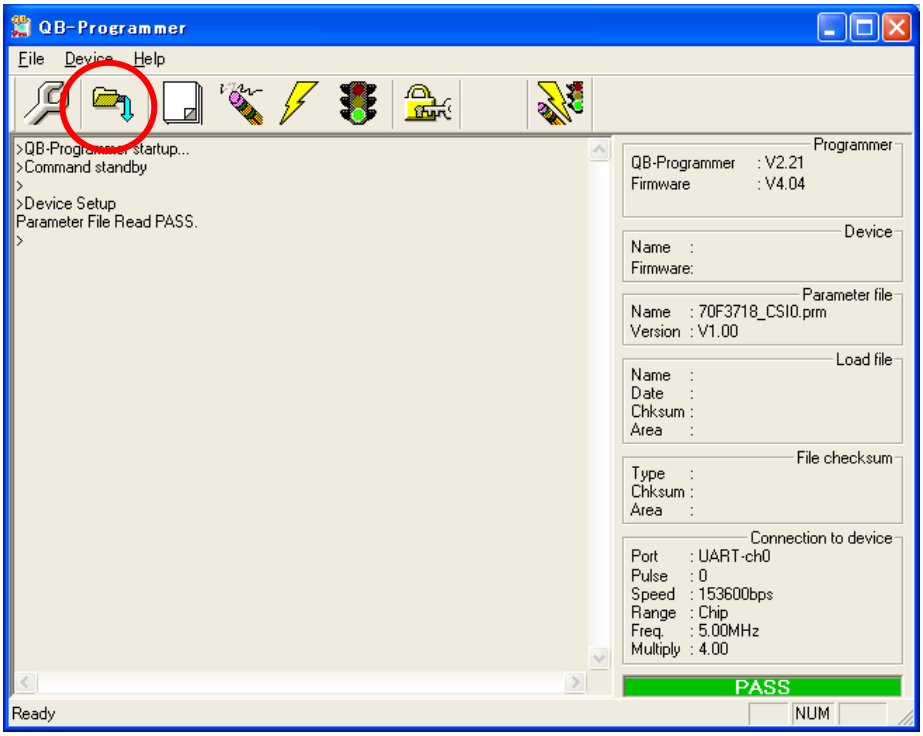
4) Select the file "PRM¥PRM70F3724_V100¥70F3718_CSI0.prm" from CD-ROM.



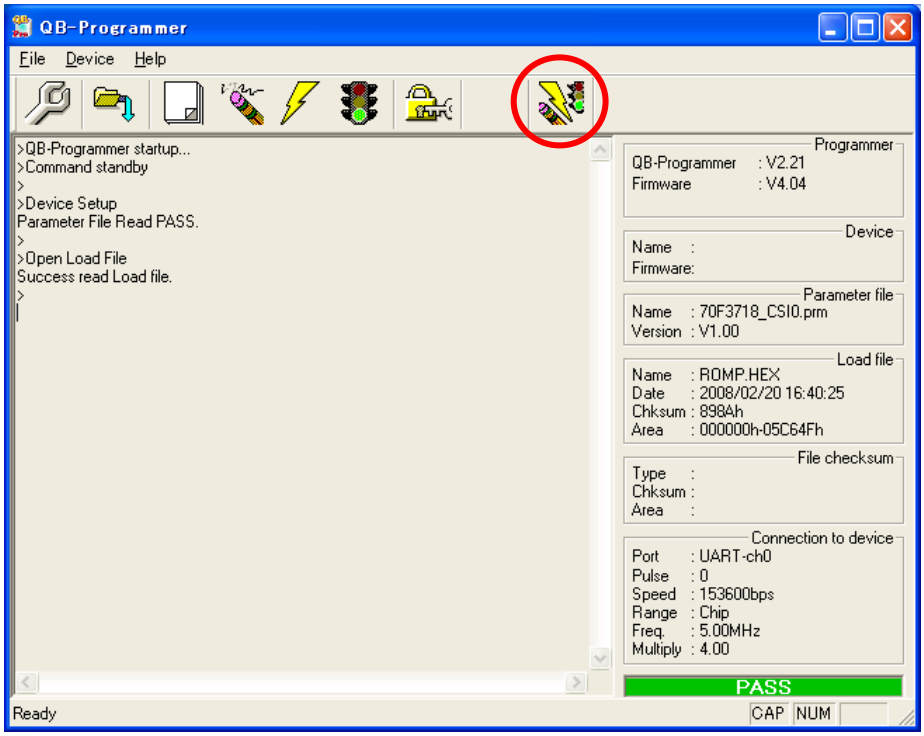
5) Check all the settings are shown above.



6) Click "Erase" button, when you need to erase the flash memory.



7) Click "Load" button when you need to write programs. Select a HEX file.



8) Click "Auto Procedure" button to start the writing process.

5.5 Circuit Diagram

From following page, it shows the circuit diagram of the kit.