Applicable Product

- **Range-free Multi-controller FA-M3**
  - Model Name: SF560-MCW
  - BASIC Programming Tool M3 for Windows

The document number and document model code for this manual are given below. Refer to the document number in all communications; also refer to the document number or the document model code when purchasing additional copies of this manual.

- Document No. : IM 34M06Q22-02E
- Document Model Code : DOCIM
Important

■ About This Manual

- This Manual should be passed on to the end user.
- Before using this product, read this manual thoroughly to have a clear understanding of the product.
- This manual explains the functions of this product, but there is no guarantee that they will suit the particular purpose of a user.
- Under absolutely no circumstances may the contents of this manual be transcribed or copied, in part or in whole, without permission.
- The contents of this manual are subject to change without prior notice.
- Every effort has been made to ensure accuracy in the preparation of this manual. However, should any errors or omissions come to the attention of the user, please contact the nearest Yokogawa Electric representative or sales office.

■ Symbols Related to Safety

⚠️ **Danger.** This symbol on the product indicates that the operator must follow the instructions laid out in this user's manual to avoid the risk of personnel injuries, fatalities, or damage to the instrument. Where indicated by this symbol, the manual describes what special care the operator must exercise to prevent electrical shock or other dangers that may result in injury or the loss of life.

соединיה

**Protective Ground Terminal.** Before using the instrument, be sure to ground this terminal.

соединיה

**Function Ground Terminal.** Before using the instrument, be sure to ground this terminal.

**Alternating current.** Indicates alternating current.

**Direct current.** Indicates direct current.
The following symbols are used only in the user's manual.

**WARNING**

Indicates a “Warning”.

Draws attention to information essential to prevent hardware damage, software
damage or system failure.

**CAUTION**

Indicates a “Caution”

Draws attention to information essential to the understanding of operation and
functions.

**TIP**

Indicates a “TIP”

Gives information that complements the present topic.

**SEE ALSO**

Indicates a “SEE ALSO” reference.

Identifies a source to which to refer.

## Safety Precautions when Using/Maintaining the Product

- For the protection and safe use of the product and the system controlled by it, be
sure to follow the instructions and precautions on safety stated in this manual
whenever handling the product. Take special note that if you handle the product in a
manner other than prescribed in these instructions, the protection feature of the
product may be damaged or impaired. In such cases, Yokogawa cannot guarantee
the quality, performance, function and safety of the product.

- When installing protection and/or safety circuits such as lightning protection devices
and equipment for the product and control system as well as designing or installing
separate protection and/or safety circuits for fool-proof design and fail-safe design of
processes and lines using the product and the system controlled by it, the user
should implement it using devices and equipment, additional to this product.

- If component parts or consumable are to be replaced, be sure to use parts specified
by the company.

- This product is not designed or manufactured to be used in critical applications
which directly affect or threaten human lives and safety — such as nuclear power
equipment, devices using radioactivity, railway facilities, aviation equipment,
shipboard equipment, aviation facilities or medical equipment. If so used, it is the
user’s responsibility to include in the system additional equipment and devices that
ensure personnel safety.

- Do not attempt to modify the product.

- In order to prevent electrical shock, turn off all the power sources before connecting
wires, etc.

- This product is classified as Class A for use in industrial environments. If used in a
residential environment, it may cause electromagnetic interference (EMI). In such
situations, it is the user’s responsibility to adopt the necessary measures against
EMI.
Exemption from Responsibility

- Yokogawa Electric Corporation (hereinafter simply referred to as Yokogawa Electric) makes no warranties regarding the product except those stated in the WARRANTY that is provided separately.
- Yokogawa Electric assumes no liability to any party for any loss or damage, direct or indirect, caused by the use or any unpredictable defect of the product.

Software Supplied by the Company

- Yokogawa Electric makes no other warranties expressed or implied except as provided in its warranty clause for software supplied by the company.
- Use the software with one computer only.
- You must purchase another copy of the software for use with each additional computer.
- Copying the software for any purposes other than backup is strictly prohibited.
- Store the original media that contain the software in a safe place.
- Reverse engineering, such as decompiling of the software, is strictly prohibited.
- Under absolutely no circumstances may the software supplied by Yokogawa Electric be transferred, exchanged, or sublet or leased, in part or as a whole, for use by any third party without prior permission by Yokogawa Electric.
General Requirements for Using the FA-M3 Controller

Set the product in a location that fulfills the following requirements:
- Where the product will not be exposed to direct sunlight, and where the operating surrounding air temperature is from 0°C to 55°C (32°F to 131°F).
  - There are modules that must be used in an environment where the operating surrounding air temperature is in a range smaller than 0°C to 55°C (32°F to 131°F). Refer to hardware user’s manual or the applicable user’s manual. In case of attaching such a module, the entire system's operating surrounding air temperature is limited to the module’s individual operating surrounding air temperature.
- Where the relative humidity is from 10 to 90%.
  - In places where there is a chance of condensation, use a space heater or the like to constantly keep the product warm and prevent condensation.
- For use in Pollution Degree 2 Environment.
- Where there are no corrosive or flammable gases.
- Where the product will not be exposed to mechanical vibration or shock that exceed specifications.
- Where there is no chance the product may be exposed to radioactivity.

Use the correct types of wire for external wiring:
- USE COPPER CONDUCTORS ONLY.
- Use conductors with temperature ratings greater than 75°C.

Securely tighten screws:
- Securely tighten module mounting screws and terminal screws to avoid problems such as faulty operation.
- Tighten terminal block screws with the correct tightening torque. Refer to the hardware user’s manual or the applicable user’s manual for the appropriate tightening torque.

Securely lock connecting cables:
- Securely lock the connectors of cables, and check them thoroughly before turning on the power.

Interlock with emergency-stop circuitry using external relays:
- Equipment incorporating the FA-M3 controller must be furnished with emergency-stop circuitry that uses external relays. This circuitry should be set up to interlock correctly with controller status (stop/run).

Ground for low impedance:
- For safety reasons, connect the [FG] grounding terminal to a Japanese Industrial Standards (JIS) Class D (earlier called Class 3) Ground*1. For compliance to CE Marking, use braided or other wires that can ensure low impedance even at high frequencies for grounding.

*1 Japanese Industrial Standard (JIS) Class D Ground means grounding resistance of 100 Ω max.
Configure and route cables with noise control considerations:
- Perform installation and wiring that segregates system parts that may likely become noise sources and system parts that are susceptible to noise. Segregation can be achieved by measures such as segregating by distance, installing a filter or segregating the grounding system.

Configure for CE Marking Conformance:
- For compliance with CE Marking, perform installation and cable routing according to the description on compliance to CE Marking in the “Hardware Manual”.

We recommend that you stock up on maintenance parts:
- We recommend that you stock up on maintenance parts, including spare modules, in advance.
- Preventive maintenance (replacement of the module or its battery) is required for using the module beyond 10 years. For enquiries on battery replacement service (for purchase), contact your nearest Yokogawa Electric representative or sales office. (The module has a built-in lithium battery. Lithium batteries may exhibit decreased voltage, and in rare cases, leakage problems after 10 years.)

Discharge static electricity before touching the system:
- Because static charge can accumulate in dry conditions, first touch grounded metal to discharge any static electricity before touching the system.

Wipe off dirt with a soft cloth:
- Gently wipe off dirt on the product’s surfaces with a soft cloth.
- If you soak the cloth in water or a neutral detergent, tightly wring it out before wiping the product.
  Letting water enter the module interior can cause malfunctions.
- Do not use volatile solvents such as benzine or paint thinner or chemicals for cleaning, as they may cause deformity, discoloration, or malfunctioning.

Avoid storing the FA-M3 controller in places with high temperature or humidity:
- Since the CPU module has a built-in battery, avoid storage in places with high temperature or humidity.
- Since the service life of the battery is drastically reduced by exposure to high temperatures, take special care (storage surrounding air temperature should be from –20°C to 75°C).
- There is a built-in lithium battery in a CPU module and temperature control module which serves as backup power supply for programs, device information and configuration information. The service life of this battery is more than 10 years in standby mode at room temperature. Take note that the service life of the battery may be shortened when installed or stored at locations of extreme low or high temperatures. Therefore, we recommend that modules with built-in batteries be stored at room temperature.

Always turn off the power before installing or removing modules:
- Failing to turn off the power supply when installing or removing modules, may result in damage.
Do not touch components in the module:
- In some modules you can remove the right-side cover and install ROM packs or change switch settings. While doing this, do not touch any components on the printed-circuit board, otherwise components may be damaged and modules may fail to work.

Do not use unused terminals:
- Do not connect wires to unused terminals on a terminal block or in a connector. Doing so may adversely affect the functions of the module.

Use the following power source:
- Use only power supply module F3PU□□□□□□ in FA-M3 Controller for supplying power input for control circuit connection.
- If using this product as a UL-approved product, for the external power supply, use a limited voltage / current circuit power source or a Class 2 power source.

Refer to the user's manual before connecting wires:
- Refer to the hardware user's manual or the applicable user's manual for the external wiring drawing.
- Refer to "A3.6.5 Connecting Output Devices" in the hardware user's manual before connecting the wiring for the output signal.
- Refer to “A3.5.4 Grounding Procedure” in the hardware user's manual for attaching the grounding wiring.
Waste Electrical and Electronic Equipment

Waste Electrical and Electronic Equipment (WEEE), Directive 2002/96/EC
(This directive is only valid in the EU.)

This product complies with the WEEE Directive (2002/96/EC) marking requirement. The following marking indicates that you must not discard this electrical/electronic product in domestic household waste.

Product Category
With reference to the equipment types in the WEEE directive Annex 1, this product is classified as a “Monitoring and Control instrumentation” product.
Do not dispose in domestic household waste.
When disposing products in the EU, contact your local Yokogawa Europe B. V. office.

How to Discard Batteries

The following description on DIRECTIVE 2006/66/EC (hereinafter referred to as the EU new directive on batteries) is valid only in the European Union.

Some models of this product contain batteries that cannot be removed by the user. Make sure to dispose of the batteries along with the product.

Do not dispose in domestic household waste.
When disposing products in the EU, contact your local Yokogawa Europe B. V. office.

Battery type: Lithium battery

Note: The symbol above means that the battery must be collected separately as specified in Annex II of the EU new directive on batteries.
Introduction

■ Overview of This Manual

This manual is for BASIC Programming Tool M3 for Windows.

BASIC Programming Tool M3 for Windows is a developmental environment for the FA-M3 BASIC CPU module.

BASIC programming, debugging and CPU setting and diagnosis can be carried out using BASIC Programming Tool M3 for Windows.

This manual describes the hardware and software environments that are necessary for using the BASIC Programming Tool M3 for Windows, and the functions and specifications prepared as the BASIC developmental environment.

The module models that support BASIC Programming Tool M3 for Windows are shown below.

<table>
<thead>
<tr>
<th>Module</th>
<th>Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>BASIC CPU module</td>
<td>F3BP20-0N,F3BP30-0N</td>
</tr>
</tbody>
</table>

SEE ALSO

For BASIC syntax, refer to the Instruction Manual for BASIC CPU Modules and YM-BASIC/FA Programming Language (IM 34M06Q22-01E).

■ Configuration of This Manual

This configuration of this manual is as follows.

Chapter 1 Introduction of Software

Describes the operating environment and setup of BASIC Programming Tool M3 for Windows.

Chapter 2 Software Overview

Describes the configuration of BASIC Programming Tool M3 for Windows and summarizes BASIC development using BASIC Programming Tool M3 for Windows.

Chapter 3 Startup and Termination of the Tool

Describes startup and termination of BASIC Programming Tool M3 for Windows.

Chapter 4 File Management Function

Describes the program file management operation.

Chapter 5 Editing Function

Describes the BASIC program editing function.

Chapter 6 Debugging Function

Describes the online debugging function by connecting the BASIC CPU module.

Chapter 7 Maintenance Function

Describes the operation management and setting functions for the BASIC CPU module.
### Appendix

<table>
<thead>
<tr>
<th>Appendix</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Appendix 1</td>
<td>Lists error codes and meaning.</td>
</tr>
<tr>
<td>Appendix 2</td>
<td>Lists and explains reserved words.</td>
</tr>
</tbody>
</table>
How to Read This Manual

Use BASIC Programming Tool M3 for Windows after thoroughly reading Chapters 1 and 2 of this manual. These two chapters include a basic general description of the setup methods and BASIC application development.

If you find any unclear sections while using BASIC Programming Tool M3 for Windows, refer to the explanation for each function in Chapter 3 and later.

The contents of each item are independent, respectively, and the manual has the configuration and contents only in reference to a function that will help you understand the specifications for BASIC Programming Tool M3 for Windows.

The basic portions of the operation and editing functions of BASIC Programming Tool M3 for Windows are realized with specifications that are as equivalent to the commercially available Windows software as possible. Accordingly, editing and operation screens are not specifically shown except for those screens specific to BASIC Programming Tool M3 for Windows.

Notation in the Text

Notation for Windows Screens and Operation

Character strings and symbols enclosed in brackets [ ]

These include menus in the menu bar, commands, text box names, and buttons.

(Example): Click [Debug]-[Run] in the menu bar.

Click [Debug] in the menu bar with the mouse and click [Run] in the pulldown menu.

Character strings and symbols enclosed in double quotation marks “ ”

These include icons, windows, dialog boxes, and key-input character strings.

(Example): The “Set date” dialog is displayed.

Notation of drawings related to BASIC Programming Tool M3 for Windows and display screens

Examples of display screens in the text are described considering a Windows 2000 environment.

In a Windows XP, Windows Vista and Windows 7 environment, icons or application names may differ.

Drawings shown in this manual may be highlighted, simplified, or partially omitted for reason of description. In addition, in the environment of a machine in actual operation, the image of the screen may not agree with that shown in this manual.

Function keys and shortcuts

In BASIC Programming Tool M3 for Windows, operations with the function keys or shortcuts are possible in addition to mouse operation from the menus.

Since key arrangements on keyboards and the presence or lack of keys vary with the type of computer, this manual describes mouse operation as a basic operation but does not describe operation using the function keys or shortcuts for duplicating functions.
Other Instruction Manuals

- When BASIC Programming Tool M3 for Windows is to be used, be sure to read the following instruction manuals before reading this manual.
  
  First read instruction manual for BASIC Programming Tool M3 for Windows Read Me First (IM 34M06Q22-11E).
  
  Refer to the following instruction manuals for the FA-M3 as necessary in addition to this manual.

- For BASIC CPU modules and YM-BASIC/FA syntax, see:
  
  Instruction Manual for the BASIC CPU Modules and YM-BASIC/FA Programming Language.

- For FA-M3 specifications and configuration, mounting and wiring, commissioning, maintenance and inspection, and limitations in mounting the entire system, see the following instruction manual:

  Hardware instruction manual (IM 34M06C11-01E)
  
  Note: For specifications of products other than the power module, base module, I/O module, cables, and terminal board unit, see the instruction manual for each product.

- If you use sequence CPU F3SP71, 76, see the following instruction manuals:

  Sequence CPU Instruction Manual — Functions (for F3SP71-4N/4S, F3SP76-7N/7S) (IM 34M06P15-01E)

- If you use sequence CPU F3SP66, 67, see the following instruction manuals:

  Sequence CPU Instruction Manual — Functions (for F3SP66-4S, F3SP67-6S) (IM 34M06P14-01E)

- If you use sequence CPU F3SP22, 28, 38, 53, 58, 59, see the following instruction manuals:

  Sequence CPU Instruction Manual — Functions (for F3SP22-0S, F3SP28-3N/3S, F3SP38-6N/6S, F3SP53-4H/4S, F3SP58-6H/6S, F3SP59-7S) (IM 34M06P13-01E)

- If you use sequence CPU F3SP08, 21, 25, 35, see the following instruction manuals:

  Sequence CPU Instruction Manual — Functions (for F3SP21, F3SP25 and F3SP35) (IM 34M06P12-02E)

- For information on the instructions used with sequence CPUs, see the following instruction manuals:

  Sequence CPU Instruction Manual — Instructions (IM 34M06P12-03E)

- When programming using ladder, see the following instruction manual.

  FA-M3 Programming Tool WideField3 Manual (Instruction and Trouble shooting) (IM 34M06Q16-01E)

  FA-M3 Programming Tool WideField3 Manual (Offline) (IM 34M06Q16-02E)

  FA-M3 Programming Tool WideField3 Manual (Online) (IM 34M06Q16-03E)

  FA-M3 Programming Tool WideField3 Manual (Script) (IM 34M06Q16-04E)
Copyrights and Trademarks

- **Copyrights**

  The copyrights for programs contained on the CD-ROM and in the online manual belong to Yokogawa Electric Corporation.

  For the online manual, PDF security is such that its contents cannot be altered. It can be output to a printer.

  If the online manual is output to a printer, use it only for utilizing this product. If the manual is printed out with a printer, exercise care so that no confusion with the latest version occurs. To use the contents, check that the version number agrees with that of the most recent CD-ROM.

  Copying the online manual or assignment for, or selling or distributing it (including providing it via a network of personal computers) to a third party is prohibited. In addition, registering or recording the manual on video tape cassette or other medium is also prohibited.

- **Trademarks**

  - “Microsoft Windows 2000”, “Microsoft Windows XP”, “Microsoft Windows Vista” and “Microsoft Windows 7” are registered trademarks of Microsoft Corporation in the US.
  
  - Pentium is a trademark of Intel Corporation.
  
  - In addition, the product and company names used in the manual are the trademarks or registered trademarks of their respective companies.
CONTENTS

Applicable Product: ................................................................. i
Important .................................................................................. ii
Introduction ............................................................................... vii
Copyrights and Trademarks ....................................................... xi

1. Introduction of Software .................................................. 1-1
   1.1 Operating Environment ..................................................... 1-1
   1.2 Install .............................................................................. 1-2
   1.3 Uninstall .......................................................................... 1-3
   1.4 Personal Computer Environment Setup ............................ 1-4
   1.5 Connection with BASIC CPU Module ............................... 1-5

2. Software Overview ............................................................. 2-1
   2.1 Software Environment Configuration .............................. 2-2
      2.1.1 Screen Configuration ................................................. 2-3
      2.1.2 Menu Configuration .................................................. 2-8
      2.1.3 Tool Buttons ............................................................ 2-11
      2.1.4 Basic Operation ......................................................... 2-12
      2.1.5 Message Box ............................................................ 2-14
      2.1.6 Setups ...................................................................... 2-17
   2.2 Programming Flow .......................................................... 2-18
   2.3 Overview of BASIC Programming Tool M3 for Windows Files 2-20
   2.4 Interpretation of Terms for BASIC Programming Tool for Windows 2-22

3. Startup and Termination of the Tool ................................. 3-1

4. File Management Functions .............................................. 4-1
   4.1 Creation of New Program ............................................... 4-2
   4.2 Opening of Existing Program ........................................... 4-3
   4.3 File Saving by Overwriting ............................................. 4-4
   4.4 Saving with New Name ................................................... 4-5
   4.5 Subprogram Saving ......................................................... 4-6
   4.6 Program File Printing ..................................................... 4-7
   4.7 Printer Setup ................................................................. 4-8
   4.8 Error List Saving .......................................................... 4-9

5. Editing Functions ............................................................... 5-1
### 5. Limitation to Editing Functions
- **5.1** Limitation to Editing Functions ............................................................. 5-1
- **5.2** Programming in Edit Window ............................................................... 5-2
- **5.3** Cut ............................................................................................................. 5-5
- **5.4** Copy .......................................................................................................... 5-6
- **5.5** Paste ......................................................................................................... 5-7
- **5.6** Search ....................................................................................................... 5-8
- **5.7** Replace ..................................................................................................... 5-9
- **5.8** Jump ......................................................................................................... 5-10
- **5.9** Automatic Line Number Assignment Setup ....................................... 5-11
- **5.10** Line Number Reassignment ................................................................. 5-12
- **5.11** Program Display Switching .................................................................. 5-13
- **5.12** Inclusion of Subprogram ...................................................................... 5-14
- **5.13** Deletion of Subprogram ....................................................................... 5-15
- **5.14** Modification of Subprogram ................................................................ 5-16
- **5.15** Library Link ............................................................................................ 5-17
- **5.16** Error List Display .................................................................................. 5-18

### 6. Debugging Function ................................................................................. 6-1
- **6.1** Debug Starting and Ending .................................................................... 6-2
- **6.2** Program Forwarding ............................................................................... 6-4
  - **6.2.1** Initialization of Program Area ..................................................... 6-4
  - **6.2.2** Downloading of BASIC Program................................................ 6-5
  - **6.2.3** Uploading of BASIC Program .................................................... 6-6
- **6.3** Debug Functions ..................................................................................... 6-7
  - **6.3.1** Debugging.................................................................................. 6-7
  - **6.3.2** Step Execution........................................................................... 6-8
  - **6.3.3** Breakpoint .................................................................................. 6-9
  - **6.3.4** Branch Tracing ........................................................................... 6-12
  - **6.3.5** Variable Tracing ....................................................................... 6-14
- **6.4** Program Residence ............................................................................... 6-16
- **6.5** Operation Setup during Debug Completion ....................................... 6-17
- **6.6** Debug List .............................................................................................. 6-18

### 7. Maintenance Function .............................................................................. 7-1
- **7.1** Restart ...................................................................................................... 7-2
- **7.2** ROM Ctrl................................................................................................ 7-3
  - **7.2.1** Switching Modes ........................................................................ 7-4
  - **7.2.2** Common Area Write ................................................................... 7-5
  - **7.2.3** Program Area Writing ................................................................ 7-6
  - **7.2.4** Erase ROM ................................................................................ 7-7
- **7.3** Communication Speed Setup ................................................................ 7-8
- **7.4** Operation Monitor ................................................................................... 7-9
- **7.5** Configuration .......................................................................................... 7-10
  - **7.5.1** CPU Configuration ................................................................. 7-11
  - **7.5.2** Shared Device Setting ............................................................. 7-12
- **7.6** Error Log ................................................................................................ 7-13
  - **7.6.1** Log Display ................................................................................ 7-14
7.6.2 Log Clear ................................................................. 7-15
7.6.3 File Saving the Log ................................................... 7-16
7.6.4 Error Messages and Their Correspondence .......... 7-17

7.7 I/O Configuration Display ................................................ 7-19
7.8 Data Set .............................................................................. 7-21

Appendix 1. Error Codes .................................................. A1-1
  Appendix 1.1 YM-BASIC/FA Error Codes .................. A1-1
  Appendix 1.2 Detailed Error Codes ......................... A1-6

Appendix 2. Reserves Words ........................................... A2-1

Revision History ................................................................. i
1. Introduction of Software

This chapter describes the personal computer environment, the methods of installation and “uninstallation,” and connection to the BASIC CPU module for introduction to BASIC Programming Tool M3 for Windows.

**CAUTION**

Log in with Administrator privileges in order to set up, perform maintenance on, or remove the BASICM3 for Windows software. Users without Administrator privileges cannot set up, perform maintenance on, or remove the BASICM3 for Windows software.

In Windows Vista or Windows 7, select **Run as Administrator** in the installer program. Users without Administrator privileges will not be able to install this software.

**CAUTION**

When User Account Control (UAC) is enabled in Windows Vista or Windows 7, the installer might not automatically run from the CD-ROM.

If this occurs, use Explorer to select **Setup.exe** on the CD-ROM, and then select **Run as Administrator** to start the installer.

**CAUTION**

When performing setup in Windows, it is recommended to install the software in a folder for which restricted users have access rights. If the software is installed in a folder that cannot be accessed by restricted users, such users will be unable to use the BASIC Programming Tool M3 for Windows software.
1.1 Operating Environment

The BASIC Programming Tool M3 for Windows operates in the personal computer environment shown below.

<table>
<thead>
<tr>
<th>Item</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>OS</td>
<td>Microsoft Windows 7 (32bit/64bit)</td>
</tr>
<tr>
<td></td>
<td>Microsoft Windows Vista (32bit/64bit)</td>
</tr>
<tr>
<td></td>
<td>Microsoft Windows XP Service Pack 3</td>
</tr>
<tr>
<td></td>
<td>Microsoft Windows 2000 Service Pack 4</td>
</tr>
<tr>
<td>CPU</td>
<td>Pentium 133 MHz or faster, adequate for the operating system to run properly.</td>
</tr>
<tr>
<td>Memory</td>
<td>32 MB or more, adequate for the operating system to run properly.</td>
</tr>
<tr>
<td>Space on hard disk</td>
<td>30 MB minimum.</td>
</tr>
<tr>
<td>CD-ROM drive</td>
<td>Supported by the above operating system</td>
</tr>
<tr>
<td>Printer</td>
<td>Any printer compatible with the operating systems listed above and supports A4 printing</td>
</tr>
<tr>
<td>Communication</td>
<td>RS-232-C start-stop synchronization; 9600 and 19200 bps</td>
</tr>
<tr>
<td>requirements</td>
<td></td>
</tr>
<tr>
<td>Supported CPU modules</td>
<td>F3BP20-0N and F3BP30-0N</td>
</tr>
</tbody>
</table>

**TIP**

166 MHz or faster CPU is recommended. If the CPU is slower, the software may run very slowly.

**CAUTION**

The online help might not be displayed in Windows Vista/7 which does not support the Windows help function by default.

In this case, install a Windows Help program applicable for each OS from the Microsoft Support site.
1.2 Install

The installation procedure for BASIC Programming Tool M3 for Windows is described.

1. Insert the software CD-ROM into the CD-ROM drive.
2. Click [Start]-[Run by specifying file name] from the Windows task bar.
3. Select or enter “setup.exe” located in the root directory of the CD-ROM drive.
   
   ![Run dialog box](F010201.EPS)

4. Click [OK].
5. The setup tool starts up. Check the display contents and click [Next] in turn.
6. After the setup tool starts, the setup destination can be changed in the setup display of the directory of the installation destination drive.
7. After checking all settings, click [Next].
1.3 **Uninstall**

A procedure to delete all BASIC Programming Tool M3 for Windows environments from the personal computer is described.

1. Click [Start]-[Set]-[Control Panel]-[Add and delete applications] from the Windows task bar.

2. Select “BASICM3 for Windows” and click [Add and delete].

![Confirm File Deletion](image)

3. Click [Yes].

---

**CAUTION**

If a system file required for the removal procedure has been deleted, “uninstallation” cannot be executed. In addition, if there are folders and files not related to the system in the system folder, the removal procedure may not terminate normally.

---

**TIP**

For inquiries on the deletion of shared files, select the [All not] button and continue “Uninstall.”
1.4 Personal Computer Environment Setup

- **Changing the COM Port Number**
  The COM port number must be changed depending on the type of personal computer and setup.
  Set up the personal computer for the BASIC Programming Tool M3 for Windows port.

  **Setting on the BASIC Programming Tool M3 for Windows side**
  Open the BASICM3.ini file in the folder where BASIC Programming Tool M3 for Windows is installed using the text editor and change the following setting:
  
  `[COM]`  
  port=COM2: in the case of using a serial port as COM2

  **TIP**
  The “BASICM3.ini” file can be opened also by selecting [CPU Setting] - [Port Setting] from the main menu for BASIC Programming Tool M3 for Windows.

  **TIP**
  In Windows Vista/7, the folder security prevents restricted users from accessing the folder where BASIC Programming Tool M3 for Windows is installed. In this case, edit the "BASICM3.ini" file in "\ProgramData\Yokogawa\BASICM3" in the installation drive.

  **Setting on the personal computer side**
  Check the usable port numbers by selecting “Control panel”-“System”-“Device manager”-“Port”-“Communication port.”
  If not set correctly, communication with the BASIC CPU module cannot be made.

  **TIP**
  To establish an online connection using KM13-1S, it is required to set up the dedicated driver software on the PC in advance.
  USB cable instruction manual (IM34M06C91-01E), which describes the installation procedure and COM port number setting procedure, and the driver software are supplied with the cable. Install the driver software from the CD-ROM supplied with the cable or from our website.
### Shortcut registration

1. Start “Explorer.”
2. Specify the folder where BASIC Programming Tool M3 for Windows is installed and drag out BASM3WIN.
3. Move the above file onto the Windows Desktop form and drop it there.
4. A shortcut can also be carried out.

### Program folder setup

Prepare a folder in advance for saving the BASIC program.

1. Start “Explorer.”
2. Enter the folder name selecting [File]-[New]-[Folder] from the menu bar.

---

**SEE ALSO**

To change the personal computer or Windows environment, see applicable instruction manual.
1.5 Connection with BASIC CPU Module

This section describes connection with the BASIC CPU module.

### Preparation

- **Personal computer cable**

  Prepare the cable dedicated to the FA-M3 CPU shown below. Select one depending on the specifications for the connector of the personal computer serial port.

<table>
<thead>
<tr>
<th>Model</th>
<th>Suffix Code</th>
<th>Style Code</th>
<th>Option Code</th>
<th>Specifications</th>
</tr>
</thead>
<tbody>
<tr>
<td>KM11</td>
<td>-2T</td>
<td>&quot;A&quot;</td>
<td>---</td>
<td>D-sub 9-pin female</td>
</tr>
<tr>
<td></td>
<td>-3T</td>
<td>&quot;A&quot;</td>
<td>---</td>
<td>3m long</td>
</tr>
<tr>
<td></td>
<td>-4T</td>
<td>&quot;A&quot;</td>
<td>---</td>
<td>5m long</td>
</tr>
<tr>
<td>KM13</td>
<td>-1S</td>
<td>---</td>
<td>---</td>
<td>USB1.1-compliant cable for use with USB port.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3m long</td>
</tr>
</tbody>
</table>
Setting a serial port number

Please set the COM number of a serial port as “BasicM3.ini” which exists in the installation place of “BASICM3 for Windows”.

Open and change the contents of “BasicM3.ini” by the text editor.
It can set up from COM1 to COM9. (Default : COM1)

(example) When the COM number of a serial port is COM5.

```
[com]
port=COM5
```

TIP
The "BASICM3.ini" file can be opened also by selecting [CPU Setting] - [Port Setting] from the main menu for BASIC Programming Tool M3 for Windows.

TIP
In Windows Vista/7, the folder security prevents restricted users from accessing the folder where BASIC Programming Tool M3 for Windows is installed. In this case, edit the "BASICM3.ini" file in "\ProgramData\Yokogawa\BASICM3" in the installation drive.

Connection

1. Connection to serial port.
   Connect the cable to the serial port in the rear of the personal computer.
2. Connect the cable to the PROGRAMMER port of the BASIC CPU module. Securely connect the cable removing the protection cover on the CPU side.
3. Confirmation of communications.
   Operate the Debug function or Maintenance function to confirm communication with the CPU.

TIP
For a personal computer having more than one serial port or using a COM port number for infrared communications, it may be necessary to change the COM port number used by the serial port.

TIP
To establish an online connection using KM13-1S, it is required to set up the dedicated driver software on the PC in advance.

USB cable instruction manual (IM34M06C91-01E), which describes the installation procedure and COM port number setting procedure, and the driver software are supplied with the cable. Install the driver software from the CD-ROM supplied with the cable or from our website.
2. Software Overview

This chapter gives a basic overview of BASIC Programming Tool M3 for Windows, the programming flow, and application files.
2.1 Software Environment Configuration

BASIC Programming Tool M3 for Windows is a program development environment for the FA-M3 BASIC CPU module under a Windows environment. It incorporates the following functions:

- **Editing Functions**
  - Program configuration and display function
    - Subprogram addition
    - Subprogram deletion
  - BASIC program editing function
    - Cut, copy, and pasting functions
    - Search, jump, and substitute functions
    - Line number assignment and setting functions
    - Line number re-assignment function
  - Line number management function
  - Error list display function

- **Debug Functions**
  - Debug run/stop
  - Downloading and uploading
  - Trace functions
    - Branch tracing
    - Variable tracing
  - Break point setting, step run
  - Program residence setting
  - Run setting when debugging is completed
  - Free area displaying

- **Maintenance Functions**
  - CPU reset start
  - ROM management function
    - Writing to common area
    - Writing to program area
    - ROM erasing
  - Baud setting
  - Program run monitor
  - Configuration
    - CPU configuration
    - Shared-device setting
  - Error log
  - I/O configuration display
  - Date setting
2.1.1 Screen Configuration

The main frame and each window of BASIC Programming Tool M3 for Windows are described.

Main Frame

The windows and bars that configure BASIC Programming Tool M3 for Windows are:

1. Menu bar
2. Toolbar
3. Program configuration window
4. Edit window
5. Debug window
6. Status bar
Program Configuration Window

This window appears on the left side of the main frame and displays the program configuration. In this window, programs to be displayed in the edit window are selected.

- Game.n.sa: Name of a file being edited
- MAIN: Main program
- Other displayed names: Subprograms
## Edit Window

This window is the program editing and debug area that appears from the center to the upper right side of the main frame. It displays a program selected in the program configuration window. For the operating method, see Chapter 5 of this manual, “Editing Functions.”

```plaintext
320 !%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
330 !
340   WHILE 1
350     IF ISEC = 60 THEN
360       IF IMIN = 59 THEN
370         GOSUB JIHOU@
380         IMIN = 0
390       IHOUR = IHOUR + 1
400     ELSE
410       IMIN = IMIN + 1
420     ENDIF
430   ISEC = ISEC - 0
440 ELSE
450   ISEC = ISEC + 1
460 ENDIF
470 END WHILE
480 IF IMIN = 59 THEN
490 GOSUB JIHOU@
500 IMIN = 0
510 IHOUR = IHOUR + 1
520 ELSE
530 IMIN = IMIN + 1
540 ENDIF
```
- **Debug Window**

  This window displays and switches between one of three types of windows: debug, error list, and error log windows.

  ![Debug Window Example](image)

  - **Debug Window**
    In this window, debug data sent from the BASIC CPU module are displayed when debugging.

  - **Error List**
    This list displays the result of checking the BASIC program syntax.

  - **Error Log**
    Information on the CPU error log is displayed. For the operating method, see Chapter 7 of this manual, "Error Log Function in Maintenance Functions."

- **CAUTION**

  The vertical scroll bar of the debug window may disappear or the horizontal scroll bar may not return to the left side, but there is no problem in displaying data. If the vertical scroll bar disappears, click the vertical scroll bar display position.
Menu Bar, Toolbar, and Status Bar

- **Menu Bar**
  The menu is for processing the functions of BASIC Programming Tool M3 for Windows. Basic operation is carried out from this menu.

- **Toolbar**
  There are three types of frequently used tool bars: standard toolbar, search toolbar, and debugging toolbar. Each function is the same as that of the menu bar. For a description of the functions, see the section for each function.

- **Status Bar**
  The CPU communication status, resident mode indication, and the progress status when downloading or uploading, are displayed.
### 2.1.2 Menu Configuration

The basic items in the menu bar and described.

<table>
<thead>
<tr>
<th>File</th>
<th>Edit</th>
<th>View</th>
<th>Debug</th>
<th>Operation</th>
<th>Setup</th>
<th>Diagnostics</th>
<th>Help</th>
</tr>
</thead>
</table>

#### File

- **Basic program file access, program printing, and termination of BASIC Programming Tool M3 for Windows can be performed.**

<table>
<thead>
<tr>
<th>File</th>
<th>Edit</th>
<th>View</th>
<th>Debug</th>
<th>Operation</th>
<th>Setup</th>
<th>Diagnostics</th>
<th>Help</th>
</tr>
</thead>
</table>

- **File**

  - **New(N) Ctrl+N**
  - **Open(O)... Ctrl+O**
  - **Save(S) Ctrl+S**
  - **Save As(Å)...**
  - **Print(P)... Ctrl+P**
  - **Printer Setup(PR)...**
  - **1 C:sers\basic\Game.bas**
  - **Exit(X)**

#### Edit

- **The editing function common to the off-line status and debugging status. Each function can be used in editing programs.**

<table>
<thead>
<tr>
<th>File</th>
<th>Edit</th>
<th>View</th>
<th>Debug</th>
<th>Operation</th>
<th>Setup</th>
<th>Diagnostics</th>
<th>Help</th>
</tr>
</thead>
</table>

- **Edit**

  - **Cut(C) Ctrl+X**
  - **Copy(C) Ctrl+C**
  - **Paste(P) Ctrl+V**
  - **Find(F)... Ctrl+F**
  - **Find Next(I)... F3**
  - **Replace(RE)... Ctrl+H**
  - **Jump(J)...**
  - **Automatic line number assignment(Å)... Ctrl+F**
  - **Line number reassignment(PR)...**
  - **Subprogram(S)...**
  - **Library(L)...**
**View**
Display for error lists, display and setting for the menu bar and status bar, and font setting for the edit window can be performed.

**Debug**
This is the debug function in the on-line status. Each function becomes effective after connection to the BASIC CPU module.

**Operation Setup**
Reset start of the CPU, the ROM management function, and the baud rate are set.
- **Diagnosis**
  This is the diagnostic function for the BASIC CPU module. CPU operation monitor, configuration, error log saving and other operations are performed.

- **Help**
  On-line help is displayed. On-line help includes function help for BASIC Programming Tool M3 for Windows, functions for the BASIC CPU module, and statement help.

---

### TIP

Operation setting and the diagnostic function do not operate during debugging. To execute operation setting and the diagnostic function, terminate debugging.

---

### CAUTION

The online help might not be displayed in Windows Vista/7 which does not support the Windows help function by default.

In this case, install a Windows Help program applicable for each OS from the Microsoft Support site.
2.1.3 Tool Buttons

An overview of the tool buttons is described. For a description of the functions corresponding to each tool button, see the section for each respective function.

### Standard Toolbar

Tool buttons for file access, editing, and print functions

- New
- Copy
- Open
- Paste
- Save
- Print
- Cut
- Help

### Debug Toolbar

Toolbar used for debug function

- Run
- Set/release break point
- Step
- Start debugging
- Stop
- Terminate debugging

### Search Toolbar

Toolbar used for Find, Move, and Replace in the editor window for editing and debugging functions.

- Find
- Move to top line
- Replace
- Move to bottom line
- Move to specified line

⚠️ CAUTION

A description of the tool buttons is not displayed during debugging.
2.1.4 Basic Operation

This section describes the basic operating specifications for BASIC Programming Tool M3 for Windows by giving examples. Read the section for each respective function after reading this section.

- **Selection from the menu bar**
  Operate the function by selecting it from the menu bar.
  (Example) Search
  1. Move the mouse pointer onto [Edit] on the toolbar and then click.
  2. Move the mouse pointer to the position for [Search].
  3. Highlight [Search] and click.
  4. The “Search” menu dialog is displayed.

- **Selection using the tool button**
  Operate the function from the tool button.
  (Example) Search
  1. Bring the mouse pointer onto the [ ] tool button and then click.
  2. The “Search” menu dialog is displayed.

- **Entering characters**
  Enter them from the keyboard.
  (Example) Statement input
  1. Click an edit line in the edit window.
  2. Since the cursor is displayed, enter characters from the keyboard.
  3. Press the Enter key.
### Shortcut

The shortcut keys that are usable in BASIC Programming Tool M3 for Windows are shown below. The functions are the same as the operations from the menu bar. For the operation methods after startup, see each section.

<table>
<thead>
<tr>
<th>Pull-down Menu Function</th>
<th>Shortcut Key(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>[File] - [New]</td>
<td>Ctrl + N</td>
</tr>
<tr>
<td>[File] - [Open]</td>
<td>Ctrl + O</td>
</tr>
<tr>
<td>[File] - [Save]</td>
<td>Ctrl + S</td>
</tr>
<tr>
<td>[File] - [Print]</td>
<td>Ctrl + P</td>
</tr>
<tr>
<td>[Edit] - [Cut]</td>
<td>Ctrl + X</td>
</tr>
<tr>
<td>[Edit] - [Copy]</td>
<td>Ctrl + C</td>
</tr>
<tr>
<td>[Edit] - [Paste]</td>
<td>Ctrl + V</td>
</tr>
<tr>
<td>[Edit] - [Find]</td>
<td>Ctrl + F</td>
</tr>
<tr>
<td>[Edit] - [Find Next]</td>
<td>F3</td>
</tr>
<tr>
<td>[Edit] - [Replace]</td>
<td>Ctrl + H</td>
</tr>
<tr>
<td>[Edit] - [Line number reassignment]</td>
<td>Ctrl + R</td>
</tr>
<tr>
<td>[Debug] - [Run]</td>
<td>F5</td>
</tr>
<tr>
<td>[Debug] - [Stop]</td>
<td>Pause</td>
</tr>
<tr>
<td>[Debug] - [Continue]</td>
<td>Ctrl + [F5]</td>
</tr>
<tr>
<td>[Debug] - [Step]</td>
<td>F6</td>
</tr>
<tr>
<td>[Debug] - [Set/release a break point]</td>
<td>F9</td>
</tr>
<tr>
<td>[Help] - [Help Topics]</td>
<td>F1</td>
</tr>
</tbody>
</table>

Shortcuts by clicking the right mouse button vary with the type of window that is open. These are set for functions frequently used or functions that do not exist in the menu bar.
2.1.5 Message Box

This section describes the message boxes displayed in BASIC Programming Tool M3 for Windows and their button input processing. They are omitted in the descriptions of the functions in each chapter.

### Confirmation and Selection Message

When operation is performed from BASIC Programming Tool M3 for Windows, these messages are displayed when confirmation of the operation is required or some preprocessing is necessary for that operation.

<table>
<thead>
<tr>
<th>Message Box</th>
<th>Options Available</th>
<th>Time of Confirmation Display</th>
<th>Type of Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Save changes to xxxxxxx?</td>
<td>Yes No Cancel</td>
<td>When currently displayed program is changed but is not yet saved</td>
<td>File</td>
</tr>
<tr>
<td>File xxxxxxx already exists. Overwrite?</td>
<td>Yes No</td>
<td>When the name of a file to be saved already exists</td>
<td>File</td>
</tr>
<tr>
<td>OK to delete xxxxxxx?</td>
<td>Yes No</td>
<td>At deletion of subprogram</td>
<td>Subprogram</td>
</tr>
<tr>
<td>OK to release xxxxxxx link?</td>
<td>Yes No</td>
<td>At release of library link</td>
<td>Library link</td>
</tr>
<tr>
<td>Connects to CPU module. Select process after connection.</td>
<td>Upload Download Cancel</td>
<td>At start of debugging</td>
<td>Debugging</td>
</tr>
<tr>
<td>OK to disconnect from CPU module?</td>
<td>Yes No</td>
<td>At termination of debugging</td>
<td>Debugging</td>
</tr>
<tr>
<td>OK to initialize program area?</td>
<td>Yes No</td>
<td>At initialization of program area</td>
<td>Debugging</td>
</tr>
<tr>
<td>&quot; xxxxxxx already exists. OK to abort xxxxxxx program?</td>
<td>Yes No</td>
<td>When another program exists during downloading</td>
<td>Debugging</td>
</tr>
<tr>
<td>OK to make xxxxxxx resident?</td>
<td>Yes No</td>
<td>At setting of program residence</td>
<td>Debugging</td>
</tr>
<tr>
<td>OK to release resident xxxxxxx?</td>
<td>Yes No</td>
<td>At release of program residence</td>
<td>Debugging</td>
</tr>
<tr>
<td>After debugging, OK to run program automatically?</td>
<td>Yes No</td>
<td>At completion of debugging or at selection of run set</td>
<td>Debugging</td>
</tr>
<tr>
<td>After debugging, OK to abort program?</td>
<td>Yes No</td>
<td>At completion of debugging or at selection of stop set</td>
<td>Debugging</td>
</tr>
</tbody>
</table>
### Setting Error

A setting error is displayed if operation from BASIC Programming Tool M3 for Windows results in an error and cannot be executed.

<table>
<thead>
<tr>
<th>Message Box</th>
<th>Options Available</th>
<th>Time of Confirmation Display</th>
<th>Type of Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Communication error</td>
<td>OK</td>
<td>When an error occurs in communication with the CPU</td>
<td>Diagnosis</td>
</tr>
<tr>
<td>Program does not exist.</td>
<td>OK</td>
<td>When running is started or stopped without a program in the CPU</td>
<td>Diagnosis</td>
</tr>
<tr>
<td>Program is already running.</td>
<td>OK</td>
<td>Running is started while program is running.</td>
<td>Diagnosis</td>
</tr>
<tr>
<td>Program is already stopping.</td>
<td>OK</td>
<td>Stopping is selected while program is stopping.</td>
<td>Diagnosis</td>
</tr>
<tr>
<td>Illegal area size; invalid setpoint value</td>
<td>OK</td>
<td>Specified size is out of allowable range in configuration.</td>
<td>Diagnosis</td>
</tr>
<tr>
<td>Cannot register.</td>
<td>OK</td>
<td>When configuration is registered while BASIC is running.</td>
<td>Diagnosis</td>
</tr>
<tr>
<td>System cannot be reset.</td>
<td>OK</td>
<td>When system reset is selected for CPU that is not the main CPU</td>
<td>Diagnosis</td>
</tr>
<tr>
<td>Invalid CPU</td>
<td>OK</td>
<td>Selected function does not exist in CPU being connected.</td>
<td>Diagnosis</td>
</tr>
<tr>
<td>Not in ROM writer mode. Data cannot be written or deleted.</td>
<td>OK</td>
<td>When CPU is not in ROM writer mode.</td>
<td>Diagnosis</td>
</tr>
<tr>
<td>Not permitted because BASIC is running.</td>
<td>OK</td>
<td>When set to ROM writer mode while BASIC is running.</td>
<td>Diagnosis</td>
</tr>
<tr>
<td>OK to restart system?</td>
<td>Yes / No</td>
<td>At start of system reset</td>
<td>Operation setting</td>
</tr>
<tr>
<td>OK to restart CPU?</td>
<td>Yes / No</td>
<td>At start of CPU reset</td>
<td>Operation setting</td>
</tr>
<tr>
<td>OK to write common area to ROM?</td>
<td>Yes / No</td>
<td>When writing common area</td>
<td>ROM</td>
</tr>
<tr>
<td>OK to write program area to ROM?</td>
<td>Yes / No</td>
<td>When writing program area</td>
<td>ROM</td>
</tr>
<tr>
<td>OK to delete ROM contents?</td>
<td>Yes / No</td>
<td>When deleting contents of ROM</td>
<td>ROM</td>
</tr>
<tr>
<td>OK to set ROM writer mode</td>
<td>Yes / No</td>
<td>At setting of ROM writer mode</td>
<td>ROM</td>
</tr>
<tr>
<td>OK to reset ROM writer mode?</td>
<td>Yes / No</td>
<td>At release of ROM writer mode</td>
<td>ROM</td>
</tr>
<tr>
<td>OK to operate xxxxxx?</td>
<td>Yes / No</td>
<td>When running with program operating status display dialog</td>
<td>Diagnosis</td>
</tr>
<tr>
<td>OK to stop xxxxxx?</td>
<td>Yes / No</td>
<td>When stopping with program operating status display dialog</td>
<td>Diagnosis</td>
</tr>
<tr>
<td>OK to update contents?</td>
<td>Yes / No</td>
<td>At registration of each diagnostic (set) function</td>
<td>Diagnosis</td>
</tr>
<tr>
<td>OK to clear error log?</td>
<td>Yes / No</td>
<td>When selecting error log clear</td>
<td>Diagnosis</td>
</tr>
</tbody>
</table>
2.1.6 Setups

This subsection describes the changes of the display setup on the screens of BASIC Programming Tool M3 for Windows.

• Setup of Toolbar Display Items

Display or no-display can be set for each item for the three types of toolbars.
1. Click [View] - [Toolbar] from the menu bar.
2. Check the tool to be displayed.
3. Click the [OK] button.
4. Checked toolbar is displayed.

• Selection of Status Bar Display

1. Click [View] - [Status bar] from the menu bar.
2. When the status bar is displayed, a check mark appears to the left of the item name.
2.2 Programming Flow

This section interprets the flow from FA-M3 BASIC application programming to debugging.

The section describes the functions of the BASIC Programming Tool M3 for Windows that become effective in each process.

● Off-line Programming Functions

Start up BASICM3 for Windows.

Create a program. [New], [Open]

Edit the program. [Edit]

[Save] the program.

Download?

Y

[Exit]

Online

N

Programming is completed.

Chapter 4 File Management Functions
Chapter 5 Editing Functions
Debug and Maintenance Functions

Chapter 7 Maintenance Functions

Chapter 5 Editing Functions
Chapter 6 Debug Functions

Chapter 7 Maintenance Functions
2.3 Overview of BASIC Programming Tool M3 for Windows Files

### File Configuration

This section describes the files created by BASIC Programming Tool M3 for Windows. It is recommended that user files be saved with the following identifying extensions:

<table>
<thead>
<tr>
<th>Extension</th>
<th>Title</th>
<th>Contents</th>
</tr>
</thead>
</table>
| .SA       | BASIC application | BASIC program edited and created with BASIC Programming Tool M3 for Windows or commercially available editor.  
Text source file. |
| .UN       | BASIC application | BASIC program edited and created with BASIC Programming Tool M3 for Windows.  
Intermediate file. |
| .ELG      | Error log     | Text file in which CPU error information is stored on a line basis.       |
| .ERR      | Error log     | Text file for displays of syntax errors and the like in editing. Stores up to 1024 items. |
| .LST      | Debug list    | Text file for data tracing information and error information lists when CPU is undergoing debugging. |
BASIC Application File Specifications

A BASIC application file (extension: .SA) can be edited with a commercially available editor and be used. Create it with the following file specifications:

- **File Name**
  Normal operation can be obtained with a character string of up to eight standard-width (single-byte) alphanumeric characters starting with an alphabetical letter.

- **File Specifications**
  Program block: Compose the entire program block with one file.
  Line number: Initialize the file for every program block and describe it in ascending order.
  Line number delimiter: Insert one or more standard width spaces between each line number and corresponding program.
  Subprogram name: Name it with a character string of up to eight standard-width alphanumeric characters starting with an alphabetical letter.

- **Folder Name**
  A character string not containing a double-width space.

**CAUTION**

- A program name displayed by the selection of “Diagnosis” - “Program run monitoring” - “Program name” is composed of eight standard-width alphanumeric characters. If a program having file names exceeding eight characters or file names using doublewidth or special characters is downloaded, the displayed characters may vary irregularly. In addition, the part of the program name exceeding the eight characters is not displayed.
- If a file that is not correctly recognized by BASIC Programming Tool M3 for Windows is opened or downloaded, operation after that may be not normal.

**TIP**

For program syntax, refer to the Instruction Manual for BASIC CPU Modules and YM-BASIC/FA Programming Language (IM 34M06Q22-01E).
2.4 Interpretation of Terms for BASIC Programming Tool M3 for Windows

This section interprets nouns specific to BASIC programming tool for M3 for Windows used in this manual.

- **Main Program**
  A program block that executes initial operation when a BASIC program is operated, and becomes the main part. It is necessary for an application to have a main program.

- **Subprogram**
  A function block called from a program. Variables and line numbers are all managed independently.

- **Program area**
  An area to store BASIC programs within the BASIC CPU module. The size of the area varies with the module. In addition, the size of the program area can be set using Configuration.

- **Common area**
  An area to store variable data within the BASIC CPU module. This is used when the same data are used between programs. In addition, the size of the common area can be set using Configuration.

- **Download**
  The operation to bring a BASIC program from a personal computer to the BASIC CPU module.

- **Upload**
  The operation to store a BASIC program residing in the BASIC CPU module on a personal computer.

- **Debugging**
  An operation mode in which BASIC program downloading, program changes, program additions, operation checks, etc. are executed by connecting a personal computer to the BASIC CPU module. The BASIC CPU module is connected at the start of debugging to enter debugging.

- **Break point**
  A function to debug the source code of a BASIC program. A running program can be suspended at a line number at which a break point is set and a step run can be made from that line.

- **Step run**
  A function to run a BASIC program step by step.
Branch tracing
Traces branching programs, such as subprograms or subroutines. The transition status of functions in the program can be seen.

Variable tracing
The status of variable data used in the program can be seen. Information is displayed every time variable data are changed.

Error log
Error information of the BASIC CPU module system.

Library
An application program package using machine language. Its configuration and usage are the same as that of a subroutine.
3. Startup and Termination of the Tool

■ Start up

1. Start BASIC programming tool M3 for Windows.
2. "Mainframe" is displayed.

⚠️ CAUTION

BASIC programming tool M3 for Windows cannot be started more than one time. [Already running] is displayed.

■ Terminate

From the menu bar, click [File] and then click [Exit].

SEE ALSO

If file saving is required or debugging is not terminated, the appropriate message is displayed. For the operating method, refer to the menu sections for each function.
4. File Management Functions

This chapter on File Management Functions describes management of a BASIC application file and program printing.

- **BASIC Application File Management**
  - File is opened.
  - File is saved.
  - Subprogram is saved.

- **Printing Function**
  - Entire program printing
  - Partial printing

---

**CAUTION**

The names of folders used for saving BASIC application files cannot have character strings containing double-width spaces.
4.1 Creation of New Program

Newly creates a BASIC program.

Terminates a BASIC program that has been being edited till now.

If new programming is executed, the program configuration and edit windows are cleared and new programming becomes possible.

1. Select new programming.

   From the menu bar, click [File] and then click [New].

   If program editing is executed in the edit window;

   ![Image of Save dialog box]

   [Yes]: If there is no file name, the Save dialog box is displayed. If the file name already exists, the file is saved with that name.

   [No]: The program file being edited is not updated and new programming is ready.

   [Cancel]: New programming is canceled and the screen returns to the window where the current editing is taking place.

   If program editing is not executed in the edit window;

   The program file is not updated and terminated.

2. Open a new programming screen.

   The program configuration and edit windows are cleared and a new programming screen appears.
4.2 Opening of Existing Program

Opens a BASIC program saved on a personal computer.
Terminates a BASIC program that has been being edited till now.

- Opening from [Open]
  1. Open
     From the menu bar, click [File] and then click [Open].
     “Open” dialog is displayed.
  2. Select program file
     Select the program file to be edited.
     ![Open dialog]
     Click [Open].
  3. Program display
     The program configuration is analyzed, and the configuration is displayed in the program configuration window and the main program in the edit window, respectively.

- Opening by selecting a recently used file
  1. A list of recently used files is displayed in the [File] menu.
  2. Click the file name.

**TIP**
A file currently being edited cannot be re-displayed even if it is selected from the recently used files. Select it from [File] - [Open].
4.3 **File Saving by Overwriting**

Saves a file being edited with the same name as that prior to editing.

1. **Save by overwriting.**
   - From the menu bar, click [File] and then click [Save].

2. **Saved**
   - It is saved with the name of the file being edited.
   - Overwriting to the existing file is not confirmed.

**TIP**

BASIC application linked to a library can be saved in intermediate language format (.UN).
4.4 Saving with New Name

The BASIC application being edited is given a new name and saved as a file.

1. Save
   From the menu bar, click [File] and then click [Save As].

2. Select type of file
   Set “File type.”
   - [Text format (*.SA)]: The file becomes a text file that can be edited in another editor.
   - [Intermediate format (*.UN)]: Saves a file in the intermediate language.

**CAUTION**

Save a BASIC application linked with a library in intermediate language format. If it is saved in text format, the library part becomes invalid.

3. Choose name
   Enter a file name.
   - **If there is no file with the same name,**
     The file is saved with the entered name.
   - **If a file with the same name already exists,**
     [Yes]: Saves by overwriting the file.
     [No]: Does not overwrite and the display returns to the Save dialog.

**CAUTION**

The names of folders used for saving BASIC application files cannot have character strings containing double-width spaces.
4.5 Subprogram Saving

A subprogram being edited in the edit window can be saved as a file.
A saved subprogram can be used in another application.

1. Select program
   Select a program to be saved in the “Program configuration window” and display it in
   the edit window.

2. Save subprogram
   From the menu bar, click [Edit] and then [Subprogram].
   The Save dialog box is displayed.

3. Choose file name
   Enter a file name.

   **If there is no file with the same name,**
   The file is saved with the entered name.

   **If a file with the same name already exists,**

   ![Save a Subprogram]

   [Yes]: Saves by overwriting the file.
   [No]: Does not overwrite and the display returns to the Save dialog.
4.6 Program File Printing

Prints a program being edited.
The function includes printing an entire program being edited or printing a specified program.

**Printing Entire Program**

Prints an entire program being edited.

1. Start of printing
   From the menu bar, click [File] and then click [Print].
2. Select [All]
   Click the [All] option button of [Printing range].
3. Print
   Confirm the printer and other items and then click [OK].

**Printing Specified Program**

Prints a program being displayed in the edit window.

1. Display program
   Display the program to be printed.
2. Start printing
   From the menu bar, click [File] and then click [Print].
3. Select [Selected area]
   Click the [Selected area] option button of [Printing range].
4. Print
   Confirm the printer and other items and then click [OK].
4.7 Printer Setup

Set the output destination of BASIC program printing.

1. Open the dialog box.
   From the menu bar, click [File] and then click [Printer Setup].
   The setup dialog box appears.
2. Change the setting.
3. Close the dialog box.
   [OK]: The change is made.
   [Cancel]: Setting reverts to the setting before the change.
4.8 Error List Saving

The contents displayed for syntax errors or the like during editing are saved in a file. The saved file can be read in a text editor.

1. Display the error list.
   From the menu bar, click [View] and then click [Error list] and [Display].
   The debug window displays the error list.

2. Save
   From the menu bar, click [View] and then click [Error list] and [Save].

3. Choose a file name.
   Enter the file name and click [Save].
5. Editing Functions

This chapter describes the editing functions of a BASIC application program. The descriptions center around operation from the menu bar.

- **Program Editing Functions**
  General editing functions, such as cutting, deletion, pasting, search, and jump, are described.

- **Program Configuration Editing Functions**
  Functions, such as subprogram addition and deletion, are described.

- **Error List**
  The display, saving and clearing of data for an error list are described.

5.1 Limitations to Editing Functions

The editing functions have the following limitations.

- If data are to be exchanged with other editors using the clipboard, use correct data. No data can be pasted if they cannot be handled in the clipboard.
- BASIC programming tool M3 for Windows does not have an operation-redo (restoring) function. Carefully execute, for example, program deletion.
- Although line numbers are automatically assigned when lines are inserted using the paste function, if the number of pasted lines is greater than the number of available lines, the line numbers cannot be acquired and thus copying is not executed.
- The length of a line can be no longer than the line number (6 characters including spaces) plus one single-byte space plus the statement (up to 246 single-byte characters). A double-byte character in a statement is converted to two single-byte characters.
5.2 Programming in Edit Window

This section describes program input in the edit window.

Text Input

The specifications are equivalent to that of a commercially available screen editor. Display the cursor in the "edit window" and enter text from the keyboard.
Enter the line number, insert one or more spaces, and then enter the description.
If the automatic line number assignment function is set, it is not necessary to enter a line number.

1. Place the cursor to the right of the line number.

```
10 | Let's go!
20 |
```

2. Enter “OPTION BASE 1” from the keyboard.

```
10 | Let's go!
20 | OPTION BASE 1
```

3. Press the [Enter] key.

```
10 | Let's go!
20 | OPTION BASE 1
30 |
```

If the automatic line number assignment function is set, a new line is added.

**TIP**

If the automatic line number assignment function is not set, enter the line number. Separate the line number from program data with one or more spaces.
Line Feed

Feed a line using the [Enter] key in the editor.
When feeding a line, the entered line is checked for syntax error.

Line Number Management

If the [Enter] key is pressed at the end of a line, a new line is inserted. In this case, if the automatic line number assignment function is set, a line number is generated on the new line. The generated line number is described.

Example 1  If the line number has sufficient margins or the last line:

```
10 PRINT "A" |  
50 END  

10 PRINT "A"  
20 |  
50 END
```

A new line number is generated by adding the designated increment to the preceding line number.

Example 2  If the line number has sufficient margins but the margin is equal to or less than the "increment":

```
10 PRINT "A" |  
20 END  

10 PRINT "A"  
11 |  
20 END
```

A new line number is generated by adding 1 to the preceding line number.

Example 3  If the line number has no margin:

```
10 PRINT "A" |  
11 END
```

An error message is displayed.

Blank Line

Don’t make a blank line between program lines.
If there is a blank line between program lines, an error occurs.
Error Check

A line for which an error occurs due to a syntax error or the like is displayed in the “error list” in the “debug window” and the character string on that line is displayed in red.

If the contents of the “error list” is double-clicked, a jump is made to the relevant error position.

Debug Window Error Display

If an error is detected, the error code and message are displayed in the “error list” of the “debug window.”

Edit Window Error Display

The character string in the error line is displayed in red.

Line 280 is displayed in red.
5.3 Cut

Cuts the program list partially.
Cut data are saved in the clipboard.

1. Select the area to be cut.
   Select the area to be cut with the mouse.

2. Cut
   From the menu bar, click [Edit] and then click [Cut].
   The selected area is cut.
5.4 Copy

Copies the program list partially.
Copied data are saved in the clipboard.

1. Select the copying area.
   Select the copying area with the mouse.

2. Copy
   From the menu bar, click [Edit] and then click [Copy].
   The selected area is saved in the clipboard.
5.5 Paste

Pastes the data from the clipboard.
If correct data are not saved to the clipboard, the data are not pasted.
If the number of lines to be pasted is more than the number of empty lines, pasting is not performed and an error is displayed.
The line number for the inserted data is automatically assigned.

1. Specify the position to be inserted.
   Specify the position where data are to be inserted with the cursor.

2. Paste
   From the menu bar, click [Edit] and then click [Paste].

3. The data are pasted.

CAUTION

If a program is to be copied from another Windows application via the clipboard, use data with the same specifications as the BASIC application file (.sa) If the data are not correct, copying is not executed.

TIP

If pasted on a line basis, a line number is assigned regardless of the original program line number.
5.6 Search

Searches for character strings.

Searches are performed only within a program being edited. Other program blocks cannot be searched.

1. Open the Search dialog box.
   From the menu bar, click [Edit] and then click [Find].

2. Enter the target of the search.
   Place the cursor in the [target character string] input area and enter the target of the search.

```
Find what: DATA
Find Next
Match whole word only
Match case
```

3. Search
   Click [Find Next].
   The cursor moves to the target of the search.

```
Find what: DATA
Find Next
Match whole word only
Match case
```

4. Search for the next target.
   Click [Find Next] again.
5.7 Replace

Replaces program data.

Replacement is carried out only within a program being edited. Other program blocks are not replaced. The replacement methods include a method to execute each replacement after confirming it and one to make all replacements at one time.

1. Open the Replace dialog box.
   From the menu bar, click [Edit] and then click [Replace].
2. Enter the contents of the replacement.
   Place the cursor in the input area for the character string before replacement and enter it; then, place the cursor in the input area for the character string after replacement and enter it.

3. Replace.

**Replacement after confirming**
Click [Find Next].
A search is started and the cursor moves to the target of replacement.

**Replacement of all targets**
Click [Replace all].
The character string in the cursor position and all subsequent occurrences of the character string are replaced.
5.8 Jump

Jumps by specifying a line number.
A jump can be carried out only within a program being edited.
There are the following three modes for jump.

- **[Move to Top Line]**
  If selected, a jump is made to the beginning of the program being edited.

- **[Move to Bottom Line]**
  If selected, a jump is made to the end of the program being edited.

- **[Move to specified line]**
  A jump is made by specifying the line number of the destination.

### Jump by specifying line number

1. Open the Jump dialog box.
   From the menu bar, click [Edit] and then click [Jump].
2. Enter the line number of the destination.
3. Execute.
   Click [OK].
5.9 Automatic Line Number Assignment Setup

Automatic line number assignment is a function to automatically generate line numbers when beginning the editing of a new line.

**Example)** Automatic assignment at the setting of starting number 12 in increments of 10

12 !////////////////////////////////////////////////////////////////////////////////////////////////////////////////
22 !/ <Program NAME> SequenceCPU common /
32 !///////////////////////////////////////////////////////////////////////
42 OPTION BASE 1
52 DEFSGN A
62 DEFDDBL C
72

When a new line is generated, a numeric value obtained by adding 10 to the preceding line number is given to that line.

1. Open the dialog box for setting automatic line number assignment.
   - From the menu bar, click [Edit] and then click [Automatic line number assignment].

   ![Set line number automatically](F050901.EPS)

   - **Set the automatic line number assignment.**
     - Click the [Yes] option button.
     - Enter [Start line number] and [Increment] respectively.

   ![Set line number automatically](F050902.EPS)

   - **Release the automatic line number assignment.**
     - Click the [No] option button.

2. **Set**
   - **[OK]:** Sets the entered contents.
   - **[Cancel]:** Terminates the setting without change.
5.10 Line Number Reassignment

Reassigns line numbers.

Example) Automatic assignment at the setting of old line number 12 and new line number 10 in increments of 10

```
12 !______________________________________________________________________________
13 !/ <Program NAME> SequenceCPU common /
14 !______________________________________________________________________________
15 OPTION BASE 1
16 DEFSNG A
17 DEFDBL C
   (from)
   |  (to)
10 !______________________________________________________________________________
20 !/ <Program NAME> SequenceCPU common /
30 !______________________________________________________________________________
40 OPTION BASE 1
50 DEFSNG A
60 DEFDBL C
```

1. Open the dialog box for reassigning line numbers.
From the menu bar, click [Edit] and then click [Line number reassignment].

![Line number reassignment dialog box]

- Reassign with automatic setup.
  Line numbers are reassigned at the setting in increments of 10 from the top line.
  Click the [Auto set] option button.

- Reassign with manual setup.
  Reassign line numbers by setting the old line number, new line number and increment.
  Click the [Manual set] option button.
  Enter [Old line number], [New line number], and [Increment] respectively.

![Manual set dialog box]

2. Set.
   [OK]: Sets the entered contents.
   [Cancel]: Terminates the setting without change.
5.11 Program Display Switching

Switch the programs displayed in the edit window.

1. Click a program to be displayed.

2. The selected program is displayed in the edit window.
5.12 Inclusion of Subprogram

Adds a subprogram to the program being opened.

 adicion of a subprogram

Name: ADDSUB(AS,BS) Starting from a file

OK Cancel

Reading Subprogram File

1. Open the subprogram inclusion dialog box.
   From the menu bar, click [Edit] and then click [Subprogram] and [Add].
2. Select a file name.
   Click [Starting from a file] to display the file list.
3. Select a file name and click [Open].

Inclusion of New Subprogram

1. Open the subprogram adding dialog box.
   From the menu bar, click [Edit] and then click [Subprogram] and [Add].
2. Enter the name of a subprogram.
   Enter a subprogram name and argument.
3. Click [OK].

CAUTION

If including a subprogram from a file, add a file configured with a subprogram only. If codes other than a subprogram exist, the inclusion may be executed incorrectly.

TIP

The name of a file that is the target of editing becomes the main program name. When adding a subprogram, a subprogram with a name identical to the file name cannot be added.
5.13 **Deletion of Subprogram**

Deletes a selected subprogram from the whole program.

1. Select a subprogram to be deleted.
   - In the program configuration window, select and highlight the subprogram to be deleted with the mouse.

2. Delete.
   - From the menu bar, click [Edit] and then click [Subprogram] and [Delete].

3. Confirmation message is displayed.
   - [Yes]: Deletes.
   - [No]: Does not delete.
5.14 Deletion of Subprogram

Modifies the program name and argument of a subprogram.

1. Selection of subprogram
   Display a subprogram to be modified in the "edit window."

2. Subprogram modification.
   From the menu bar, click [Edit] and then click [Subprogram] and [Modify].

3. Entry of subprogram
   Modify subprogram name and argument.

4. Click [OK].
   The program name in the “Program configuration window” and the display of the subprogram name and argument of the subprogram are modified.

**TIP**

- Modification of a subprogram name and argument cannot be executed in the “Edit window.” The name of the file that is the target of editing becomes the main program name.
- Modification to a subprogram with a name identical to the file name cannot be done.
5.15 **Library Link**
Sets and releases a library link.

**Link Setup**

1. Start the link setup.  
   From the menu bar, click [Edit] and then click [Library] and [Add].
2. Select library.  
   Select a library file and click [Open].
3. The library is added.

**Link Release**

1. Select library.  
   Select a library to be deleted in the program configuration window.
2. Start link release.  
   From the menu bar, click [Edit] and then click [Library] and [Delete].
3. Confirmation message is displayed.  
   [Yes]: Releases.  
   [No]: Does not release.
5.16 **Error List Display**

The error list checks the syntax of the program being edited and displays the lines in which there are errors and the error codes in its own list in the debug window. From the menu bar, click [View] and then click [Error list] to start each process.

- **[Display]**
  The debug window section switches to the error list display.

- **[Clear]**
  Clears the error list page.

- **[Save]**
  Saves the error list.
  Enter the file name and click [Save].
  The saved file can be read in a text editor.

- **Jump to the error part**
  If the error contents displayed in the “Error list” in the debug window are double-clicked, a jump is made to the relevant error part.

**SEE ALSO**

For error contents and countermeasures, refer to the Instruction manual for BASIC CPU Modules YMBASIC/FA Programming Language (IM 34M06Q22-01E), or Online Help.
6. Debugging Functions

This chapter mainly explains the operating methods regarding the debug functions of the BASIC CPU module application programs.

- **Program forwarding and initialization**
  Explains how to download and upload an application program and how to initialize the program area.

- **Debug function**
  Explains the operation of each type of tracing function.

- **CPU control function**
  Explains the residence setup of the application program and the operation method at the completion of debugging.

- **Debug list**
  Explains the clearing of the debug list area and file saving.
6.1 Debug Starting and Ending

**Debug Starting**

Connect with the BASIC CPU Module. After the connection, “Online” will appear in the status bar.

1. Confirm the connection of the hardware.
   - Connect the cable connector of the computer to the PROGRAMMER port on the front of the BASIC CPU.
   - Connect the cable connector of the computer to the COM port in the back of the computer.

2. Start Debugging
   - Click [Debug] - [Start debugging] from menu bar.

   ![Connection Window](F060101.EPS)

   - [Upload]: Uploads the program that is in the CPU.
   - [Download]: Downloads the program being edited in the CPU.
   - [Cancel]: Stops the connection with the CPU

   Selecting [Upload] or [Download] will place the module in the debug mode, enabling the BASIC CPU Module to be debugged.

3. Online Display
   - “Online” appears in the status bar.

---

**CAUTION**

When downloading after editing the file, follow the instructions given below:

- To download the file before editing: Download it without saving.
- To download the edited contents: Download them after saving.
- When downloading a new file, be sure to save the file, and then download the file. Otherwise, the edited contents will be lost.
## Debug Termination

Terminates debug communications between the BASIC CPU Module and ML-bus CPU Module.

1. **Terminate Debugging**
   - Click [Debug] - [Terminate debug] from the menu bar to end the debug mode. After termination, the BASIC CPU Module and ML-bus CPU Module cannot be debugged.

2. **Offline display**
   - “Offline” appears in the status bar.

---

### CAUTION

If the BASIC CPU Module is not in the residence mode at the time of debug termination, the program will be erased. In the case of saving a program in the BASIC CPU module, specify program residence. Refer to Section 6.4, “Program Residence” for the residence function.
6.2 Program Forwarding
This section describes the process of forwarding program information to the BASIC program area.
- Initialization of program area
- Downloading of BASIC program
- Uploading of BASIC program

6.2.1 Initialization of Program Area
Erase and initialize the contents of the program area. With the initialization function of the program area, the common area will not be initialized. To initialize the common area, use the INIT COM statement. For the INIT COM statement, refer to the Instruction Manual for BASIC CPU Modules and YM-BASIC/FA Programming Language (IM 34M06Q22-01E)

1. Initializing the program area
   Click [Debug] - [Initialize the program area] from the menu.
2. Confirmation
   Confirm the initialization
   [Yes]: Program area is initialized.
   [No]: No program area is initialized.
3. The program area is initialized
### 6.2.2 Downloading of BASIC Program

The download function downloads the BASIC programs into the BASIC CPU module after performing a syntax check of the program that is undergoing editing.

If a syntax error occurs, the results are displayed in the “Debug window - Error list”, and no downloading is done.

1. **Opening the program**
   
   Open the program file before downloading. If it is already opened, the program that is the object of editing is downloaded.

2. **Downloading**
   
   Click [Debug] - [Download] from the menu.
   
   The following operation is performed according to the BASIC program and CPU status.

   **If there are no errors in the program:**
   
   “OK to download the program?” is displayed in the message box.
   
   - [Yes]: Downloading is started.
   - [No]: No downloading is done.

   “Downloading is completed” appears in the message box.

   **If there are errors in the program:**
   
   “Error occurred. Downloading cannot be done” is displayed in the message box.

   Confirm the program.

   **If the program already exists in the CPU:**
   
   A confirmation message box is displayed.
   
   - [Yes]: Downloading is started.
   - [No]: No downloading is done.

---

**TIP**

To avoid an insufficient domain error during downloading, before downloading, save the program in BASIC intermediate files. After confirming the program size with the help of the explorer, etc., carry out the downloading.
6.2.3 Uploading of BASIC Program

While connecting with the BASIC CPU module, if a program resides at the BASIC CPU module side, upload that program.

1. Uploading
   - Click [Debug] - [Upload] from the menu bar.
   - The program that resides in the CPU is uploaded. The upload status is displayed in the “Status Bar.”

2. Display
   - The uploaded program is displayed.
6.3 Debug Functions
This section describes the online debug function of the BASIC application program. A detailed explanation of the function and operations from the menu bar is given.

- **Debugging**
  Describes the change functions of CPU running operations like running, disconnecting, step running, etc.

- **Breakpoint setup function**
  Describes the breakpoint setup function.

- **Trace Function**
  Describes the branch tracing and variable tracing setup functions.

### 6.3.1 Debugging
To start debugging, stop debugging and run programs continuously, follow the instructions given below:

- **Start Debugging**
  Click `[Debug] - [Run]` from the menu bar.
  Debugging is started.

- **Stop Debugging**
  Click `[Debug] - [Pause]` from the menu bar.
  The program under execution is temporarily stopped.
  The character string that is stopped changes to green.

- **Continue Debugging**
  Click `[Debug] - [Continue]` from the menu bar.
  The stopped program will restart.

---

**CAUTION**

- An explanation of the tool button will not be displayed during debugging.
- If the program is edited after setting the breakpoint setup function and tracing function, that program will not operate correctly. In that case, make individual settings again after editing.
6.3.2 **Step Execution**

Step execution is a debug function that runs the program step by step after it has been stopped, or paused due to a breakpoint.

1. **Stopping running the CPU.**
   - Stops the CPU using the stop function or at a breakpoint.
   - When it is stopped, the character string of the line following the one that was last executed changes to green.

2. **Running the program step by step**
   - Click [Debug] - [Step] from the menu bar.

3. **Step movement**
   - Move to the next executed line.

---

```plaintext
320 !://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/://!/:<
6.3.3 Breakpoint

This sets the place in a program where the program execution halts. The breakpoint can be set up one by one for each subprogram.

Setting the breakpoint

1. Selecting the break line
   Display the program in the [Edit Window] and specify the break position with the cursor.

   ```
   370   WHILE 1
   390     IF ISEC = 60 THEN
   390     IF IMIN = 59 THEN
   410       IMIN = 0
   420       IHOUR = IHOUR + 1
   430       ELSE
   440       IMIN = IMIN + 1
   450   ENDIF
   460     ISEC = 0
   470     ELSE
   480       ISEC = ISEC + 1
   490   ENDIF
   500   END WHILE
   510   STOP
   ```

2. Setting the breakpoint
   Click the [Debug] - [Set/release a breakpoint] from the menu bar.

3. Entering the number of transit times
   Enter the number of transit times.

4. To set the number of transit times, click [OK].

TIP

Setting the number of transit times is a function that makes a break only when the break setup lines pass over the setup cycles.
Set 1 when the break is set for the first time.
After stopping at the breakpoint one time, the setting of the number of break times returns to 1.
5. The number of transit times is set. The breakpoint is displayed in blue characters.

<table>
<thead>
<tr>
<th>Line</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>420</td>
<td>I HOUR = I HOUR + 1</td>
</tr>
<tr>
<td>430</td>
<td>ELSE</td>
</tr>
<tr>
<td>440</td>
<td>I MIN = I MIN + 1</td>
</tr>
<tr>
<td>450</td>
<td>END IF</td>
</tr>
<tr>
<td>460</td>
<td>I SEC = 0</td>
</tr>
</tbody>
</table>

**CAUTION**

There may be some cases where the program is not stopped at the breakpoint when a breakpoint is set to the added lines while editing the program. In this case, reset a breakpoint after executing once.

**TIP**

Only one breakpoint can be set in one subprogram. If a multiple number of breakpoints is set in one subprogram, only the last setting will be valid and the others will be cancelled. A breakpoint set in a subprogram breaks only once.
Releasing a breakpoint

Release a breakpoint that has been set.

1. Select the line.

   Display the program in [Edit Window] and specify with the cursor the position at which the breakpoint has to be released.

   ```
   420   IHPUR = IHPUR + 1
   430     ELSE
   | 440   IMPN = IMPN + 1
   450     ENDIF
   460     ISC = 0
   ```

2. Release the breakpoint.

   Click [Debug]-[Set/release a breakpoint] from the menu bar.

3. The breakpoint is then released.

Releasing a breakpoint

Release all the breakpoints in the entire program.

1. To release all the breakpoints:

   Click [Debug]-[Release all breakpoints].

2. All breakpoints are released.
6.3.4 Branch Tracing

Branch tracing is a function that traces the branch status in a subprogram. In a subprogram in which branch tracing is set, when program branching is done, the branch information will be displayed under [Debug] in [Debug Window].

CAUTION

If editing is done after setting branch tracing, branch tracing will not be displayed correctly. After editing, reset branch tracing once again.

Setting of branch tracing

1. Start the Branch Tracing dialog box.
   Click [Debug]-[Branch tracing] from the menu bar.
2. Select the subprogram that traces the branch status.
   Select the subprogram displayed in [Not Set] and highlight it in reverse video.
3. Set the branch tracing.
   Click the [Set >] button.
   The selected subprogram is moved to “To be set.”
Branch tracing setup of all programs

1. Start the branch tracing dialog box.
   Click [Debug] - [Branch tracing] from the menu bar.
2. Set the branch tracing of the whole program
   Click the [All >>] button.
   All the programs are moved to “To be set.”

Release of branch tracing

1. Start the branch tracing dialog box.
   Click [Debug] - [Branch tracing] from the menu bar.
2. Select the subprogram that releases tracing of the branch status.
   Select the subprogram that is displayed in “to be set,” and highlight it in reverse video.
3. Release
   Click the [<Release] button.
   The selected subprogram moves to “not set.”

Branch tracing release of all subprograms

1. Start the branch trace dialog box.
   Click [Debug] - [Branch tracing] from the menu bar.
2. Release
   Click the [<<All] button.
   Move all the subprograms to “not set.”

Display of the branch status

The branch status is displayed under [Debug] in [Debug window].

The above example shows the item that has been branched at line 1230 of the program “SUBPRG” in MASTER.
6.3.5 Variable Tracing

Variable tracing is the function that traces a variable change during debugging. For a variable for which variable tracing is set, whenever the data undergoes a change, the change information will be displayed under [Debug] in [Debug Window].

![Variable tracing dialog box]

**CAUTION**

After adding a variable in online editing, setup for tracing of the variable cannot be done before executing the program. If it is set, it will be displayed in the debug window as a "Variable name not defined" error. In the case of tracing an additional variable, trace it after executing the program. If editing is done after setting variable tracing, variable tracing will not be displayed correctly. After editing, reset variable tracing once again.

### Setting of variable tracing

1. Start the variable tracing dialog box.
   - Click [Debug] - [Variable tracing] from the menu bar.
2. Select [Yes].
   - Click the [Yes] option button.
3. Enter the subprogram name and variable that is to be traced.
   - Select the program and input the variable.

![Variable tracing dialog box]

4. Set the variable tracing.
   - [OK]: Variable tracing is set.
   - [Cancel]: Deletes and ends the task before it is completed.
Release of variable tracing

1. Start the variable tracing dialog box.
   Click [Debug] - [Variable tracing] from the menu bar.
2. Select [No].
   Click the [No] option button.
   [OK]: Variable tracing is released.
   [Cancel]: Deletes and ends the task before it is completed.

Display of the variable data change

A change in variable data is displayed under [Debug] in [Debug Window].

The above example shows that the variable data in the "PROGR" program is changed to 128 on line 325; after that, it changes to 24 on line 410.
6.4 Program Residence

The program residence function is the function that makes the program resident in the CPU during program downloading or at the end of debugging.

1. Specify the Reside.
   Click [Debug] - [Specifying program residence] from the menu bar.

2. Set up.
   Click [Reside] or [Release].
   - [Reside]: Does the residence specification.
   - [Release]: Deletes the residence specification.

CAUTION

If there is no residence setup, the application program will not be saved in the BASIC CPU module. In this case, the program will be erased at the end of debugging.
6.5 Operation Setup during Debug Completion

The operation setup function at the completion of debugging runs or stops the program after program downloading or at the completion of debugging.

1. Change the operation mode at the completion of debugging.
   - Click [Debug] - [Operation mode during debugging completed] from the menu bar.

2. Setup.
   - Click [Run] or [Stop].
     - [Run]: Run starts at the completion of debugging.
     - [Stop]: Stops at the completion of debugging.

3. Confirmation of setup
   - Display the confirmation message.
     - [Yes]: Makes the setup valid.
     - [No]: Makes the setup invalid.

---

CAUTION

“Operation setup at completion of debugging” operates only when the program is resident.
6.6 Debug List

The debug list is a list of each type of trace data and error and is displayed while the CPU program is undergoing debugging.

- **Displaying the program list.**
  Click the [Debug] tab in [Debug Window].

- **Clearing the debug list.**
  Click [Debug] - [Debug list] - [Clear] from the menu bar to clear the debug list.

- **Saving the Debug list.**
  Click [Debug] - [Debug list] - [Save in File] from the menu bar.

  Input the file name in the Save dialog box, and click [Save].
  A saved file can be read in a text editor.
7. Maintenance Function

The maintenance functions are detailed set-up functions of the CPU.
The following are the maintenance functions

- Restart
- ROM Control
- Communication Speed Setup
- Operation Monitor
- Configuration
- Shared Device Setup
- Error Log
- I/O Configuration Display
- Date and Time Setup

⚠️ CAUTION

The maintenance functions cannot be executed during online debugging.
Before using the maintenance functions, perform [Terminate debugging].
7.1 Restart

The restart function has two types of resets: system restart and CPU restart.

- **System restart**
  
  Restarts the CPU for the entire unit. System restart is valid only for the slot 1 CPU.

- **CPU restart**
  
  The targeted CPU alone will restart.

### System restart

1. System restart
   
   From the menu bar, click [Operation setup] and then click [Restart] and [System].

2. Confirmation
   
   Displays the confirmation dialog box
   
   [Yes]: System restarts.
   
   [No]: Ends.

3. Restart
   
   Displays Dialog box “System restarted.”

### CPU restart

1. CPU restart
   
   From the menu bar, click [Operation setup] and then click [Restart] and [CPU].

2. Confirmation
   
   Displays the confirmation dialog box
   
   [Yes]: CPU restarts.
   
   [No]: Ends.

3. Restart
   
   Displays Dialog box “CPU restarted.”

---

**CAUTION**

- Restart enables the resetting of the hardware default values and the checking of the self-diagnosis and I/O configuration.

- When restarted, programs which are not resident get erased. The residing programs are saved and started.

- Restarting can be done for an individual CPU. But, if individual CPU resetting (restarting) is done in multiple CPUs, an inter-CPU communications error or an I/O error occurs in other CPUs which are not reset. This will have an adverse effect on program running. To avoid this, restart a multiple CPU system by restarting the system.
7.2 **ROM Control**

The ROM Control function writes an application program in ROM. There are two types of ROM writing.

- **Common area write**
  Write Program, Common in ROM

- **Program area write**
  Writes only the program in ROM

- **ROM Erase**
  Erase the contents of ROM

**TIP**

ROM control can use only the BASIC CPU module (F3BP20-0N, F3BP30-0N) to which the ROM pack can be attached.

**SEE ALSO**

Refer to the Instruction Manual for BASIC CPU Modules and YM-BASIC/FA Programming Language (IM 34M06Q22-01E) for details on the ROM Control function.
7.2.1 Switching Modes

In the case of ROM operations, it is necessary to switch to the ROM writer mode.

■ Switching to ROM Writer Mode

1. Set ROM writer mode
   From the menu bar, click [Operation setup] and then click [ROM] - [ROM mode] - [Set].

2. Confirmation
   Displays confirmation dialog box
   [Yes]: Switches to ROM mode.
   [No]: Ends.

■ End ROM Mode

1. End ROM mode
   From the menu bar, click [Operation setup] and then click [ROM] - [ROM mode] - [Reset].

2. Confirmation
   Displays confirmation dialog box.
   [Yes]: Cancels ROM mode.
   [No]: Ends.

TIP

When a BASIC program is in operation, the ROM mode cannot be set. Switch to this mode after stopping the BASIC program.
7.2.2 Common Area Write

Write Program and Common area in ROM.

1. Downloading of program
   Downloads the program written in ROM to the CPU.

2. Switch to residence mode
   Performs the residence specification of the program.

3. Switch to ROM mode
   Switches to ROM mode.

4. Fix the ROM
   Switch off the power to the FA-M3 and fix the ROM and then switch the power back on.

5. Write Common area
   From the menu bar, click [Operation setup] and then click [ROM] and [Write to Common area].

6. Confirmation
   Confirmation dialog box is displayed.
   [Yes]: Write in ROM
   [No]: Ends without writing to ROM.

TIP

Download the program before switching to the ROM write mode.

Downloading cannot be done in the ROM mode.
7.2.3 Program Area Writing

Only the program is written in ROM.

1. Downloading the program
   The program written in ROM is downloaded in the CPU.

2. Switch to the Reside mode.
   Specify the program residence.

3. Switch to ROM mode.
   Switch to ROM mode.

4. Setting up ROM
   Switch off the power to the FA-M3, set up the ROM and switch the power back on again.

5. Writing to Program area.
   Click [Operation setup] - [ROM ] - [Write to Program area].

6. Confirmation
   Confirmation dialog box is displayed.
   [Yes]: Write to ROM.
   [No]: End without writing to ROM.

TIP
Download the program before switching to the ROM mode.
Downloading cannot be done in the ROM mode.
7.2.4 Erase ROM

Erasing the contents of ROM

1. To erase ROM:
   From the menu bar, click [Operation setup] and then click [ROM] and [Erase].

2. Confirmation
   The confirmation dialog box is displayed.
   [Yes]: Erase ROM.
   [No]: End.

3. Erased.
   The confirmation dialog box is displayed.
7.3 Communication Speed Setup

Set up the speed of communication with the CPU. After changing the communication speed, carry out [Start Debugging].

1. Changing the communication speed (Baud set)
   From the menu bar, click [Operation setup] and then click [Baud set].

2. Selecting the Baud rate
   Select from 2 modes:
   - [9600 bps]: This is low speed.
   - [19200 bps]: This is high speed.

⚠️ CAUTION

After starting debugging, the baud rate cannot be changed.
After the baud setting, debugging is done by performing [Start debugging]
## 7.4 Operation Monitor

Display running status of the BASIC CPU module.

- **Start**
  
  From the menu bar, click [Diagnostics] and then click [Program run monitoring].

- **View**
  
  ![Program run monitoring](F070401.EPS)

  **Status**: Displays the run status.

  **Name**: The name given in the BASIC program. ‘********’ denotes that the program does not exist. ‘$$$$$$$$’ denotes programs that are not saved.

- **[Run] Button**
  
  Starts a paused program.

- **[Stop] Button**
  
  Stops a running program.
7.5 Configuration
Sets up the default settings of the BASIC CPU module.

- CPU Configuration
- Shared device setup

⚠️ WARNING

If CPU Configuration is executed, the program and data saved in the CPU before the execution will be lost. It is recommended that the program be saved before executing CPU Configuration.
### 7.5.1 CPU Configuration

In CPU Configuration, the 'immediate mode' (momentary power failure detection mode), 'user area size setting' and 'common area setting' are performed.

#### CPU configuration

<table>
<thead>
<tr>
<th>Power failure mode</th>
<th>Standard mode</th>
<th>Immediate mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>User area size</td>
<td>400 KB</td>
<td></td>
</tr>
<tr>
<td>Common area size</td>
<td>112 KB</td>
<td></td>
</tr>
<tr>
<td>Total area size</td>
<td>512 KB</td>
<td></td>
</tr>
</tbody>
</table>

1. **Start the dialog box**
   - From the menu bar, click [Diagnostics] and then click [Configuration] and [CPU Configuration].

2. **Switch to immediate mode**
   - Click either of the option buttons, [Standard Mode]*1 or [Immediate Detection Mode]*1.
   
   *1: For details, refer to the Instruction Manual for BASIC CPU Modules and YM-BASIC/FA Programming Language (IM34M06Q22-01E).

3. **Change or editing of user area size**
   - Click the input area.
   - Input to the Setting Display (view) Area becomes possible. Input numeric values.

4. **Change or editing of common area size**
   - Click the input Area.
   - Input to the setting display (view) area becomes possible. Input numeric values.

5. **Registration of change or edit**
   - Click the [Register] button after having set it in the BASIC CPU Module, clear the memory and automatically start.

#### CAUTION

The setting of contents to the BASIC CPU Module is performed by [Register]. If it is not performed, the BASIC CPU Module returns to the original setting.

Click [Cancel] in the case where the contents are not set in the BASIC CPU Module.

#### SEE ALSO

For details on the immediate mode, user area and common area, refer to the Instruction Manual for BASIC CPU Modules and YM-BASIC/FA Programming Language (IM 34M06Q22-01E).
7.5.2 Shared Device Setting

Do the setting for the number of shared devices.

1. Start the shared relay and register configuration dialog box.
   From the menu bar, click [Diagnostics] and then click [Configuration] and [Shared relay and register configuration].

2. Change data (on number of devices).
   Click ‘no. of devices display area.’
   Input the number of devices. For shared relays, also do only the setting by matching with the other CPUs. Data sharing with the sequence CPU depending on shared relays cannot be done.

3. Registration of change
   Click the [Register] button.
   The BASIC CPU Module setting is transmitted and registered.

CAUTION

The setting of contents to the BASIC CPU Module is performed by [Register]. In the case where [Register] is not done, the BASIC CPU Module returns to its original setting.
Click [Cancel] in the case where the contents are not set in the BASIC CPU Module.

SEE ALSO

With regard to an overview of the shared devices and operation specifications, refer to the Instruction Manual for BASIC CPU Modules and YM-BASIC/FA Programming Language (IM 34M06Q22-01E).
7.6 Error Log

The log data for each type of error that occurs during operation are displayed in the BASIC CPU Module.

- **Log Display**
  Display the Log information.

- **Log Clear**
  Clear all the Log contents.

- **Save in the File**
  Save the displayed Log contents in a file.

**TIP**

- Power on/off, etc. of the FA-M3, changes in system status, abnormalities, error information, etc. are displayed with latest data first.

- A maximum of around 100 to 150 pieces of information can be stored, but this number will differ depending upon the type of information.

- If the storage capacity becomes full, the information will be erased in order starting with the oldest items, and new information will be stored.
7.6.1 Log Display

Log will be displayed in [Error log] page of “Debug Window.”

1. “Debug Window” is switched to “Error Log Display.”
   Click the [Error log] tab of the “Debug Window.”

   Switched over to “Error Log Display” window.

2. Display error log.
   From the menu bar, click [Diagnostics] and then click [Error log] and [Display]
   This is uploaded and displayed in “Debug Window.”

<table>
<thead>
<tr>
<th>Date</th>
<th>Time</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1999/08/20</td>
<td>14:25:53</td>
<td>Power down</td>
</tr>
<tr>
<td>1999/08/20</td>
<td>14:25:37</td>
<td>E82-A1 L=00081 Incorrect Slot number</td>
</tr>
<tr>
<td>1999/08/20</td>
<td>14:25:37</td>
<td>Start-up completed</td>
</tr>
<tr>
<td>1999/08/20</td>
<td>14:24:05</td>
<td>E54-92 L=00350 Communication error/CPU type error</td>
</tr>
</tbody>
</table>
7.6.2 Log Clear

Clear the log information of the BASIC CPU Module.
The error log display of “Debug Window” is also cleared.

1. “Debug Window” is switched to “Error Log Display.”
   Click the [Error log] tab of “Debug Window.”

2. Clear the log
   From the menu bar, click [Diagnostics] and then click [Error log] and [Clear].
   The error log is cleared.
7.6.3 **File Saving the Log**

Save the error log, while it is displayed in “Debug Window”, in a file. The saved file can be read in a text editor.

1. “Debug window” is switched to “Error Log Display.”
2. Save File.
   
   From the menu bar, click [Diagnostics] and then click [Error log] and [Save].

3. Enter the file name in the file save dialog box, and click [Save].
7.6.4 Error Messages and Their Correspondence

This section describes the method of correspondence when an error message is displayed in the error log display.

There are the following two types of messages that are displayed in the error log display.

- **System message:** Error message corresponding to the operation status of the power on/off system.
- **BASIC Error Message:** Error message corresponding to the operation status of the BASIC program.

Two types of message information are recorded in the error log file. The recording method to the file is as follows:

- **System message:** Record in the normal time error log file.
- **BASIC Error Message:** Recorded in the error log file while BASIC is operating in the real mode.
Correspondence of System Messages

The format of a system message is displayed, and it explains the contents of the message and its correspondence.

- **Display Format**

<table>
<thead>
<tr>
<th>Date</th>
<th>Time</th>
<th>Character string of the message</th>
</tr>
</thead>
<tbody>
<tr>
<td>1999/12/01</td>
<td>15:12:15</td>
<td>F070605.EPS</td>
</tr>
<tr>
<td>(Year/Month/Day)</td>
<td>(Hour/Minute/Second)</td>
<td></td>
</tr>
</tbody>
</table>

- **System Message List**

<table>
<thead>
<tr>
<th>Message Character String</th>
<th>Meaning</th>
<th>Correspondence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start Up End</td>
<td>The CPU is set up normally and is in the READY status.</td>
<td>Process not necessary since it is normal.</td>
</tr>
<tr>
<td>Occurrence of momentary power failure.</td>
<td>Recovered after detecting a momentary stop/failure.</td>
<td>Check the power source system.</td>
</tr>
<tr>
<td>CPU Down</td>
<td>Forced down due to an abnormality in the CPU.</td>
<td>Replace the CPU module.</td>
</tr>
<tr>
<td>Program Abnormality End Program = x x x x</td>
<td>The system program xxxx has abnormally ended.</td>
<td>Replace the CPU module.</td>
</tr>
<tr>
<td>Back-up battery Abnormality</td>
<td>Back-up battery defective or life span affected, or there has been damage to memory.</td>
<td>Replace the CPU module.</td>
</tr>
<tr>
<td>Power down</td>
<td>The power has been disconnected.</td>
<td>Not necessary for normal process.</td>
</tr>
<tr>
<td>SLOTn I/O Module abnormality:*1 Error code = k</td>
<td>When the CPU accesses the I/O module, an abnormality in the I/O module is detected.</td>
<td>Replace I/O module. There may be a case where the base module and CPU module also must be replaced.</td>
</tr>
</tbody>
</table>

*1: n: Slot number of the I/O module where the abnormality occurred.
K: Error Code (Contents Information)

Correspondence of BASIC error messages

Describes the contents and their correspondence BASIC error messages.

- **Display Format**

<table>
<thead>
<tr>
<th>Date</th>
<th>Time</th>
<th>Error code</th>
<th>Line Number</th>
<th>Message character string</th>
</tr>
</thead>
<tbody>
<tr>
<td>1999/12/01</td>
<td>15:12:15</td>
<td>E82-E6</td>
<td>L=00170</td>
<td>f.RSCOMTST p.RSCOMPT1 F070605.EPS</td>
</tr>
<tr>
<td>(Year/Month/Day)</td>
<td>(Hour/Minute/Second)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **BASIC Error Message**

  See Appendix 1, Error Codes later in this manual.
### 7.7 I/O Configuration Display

Displays the module configuration information. A module accessed by the BASIC CPU module is displayed.

#### I/O Module

<table>
<thead>
<tr>
<th>I/O Module/ I/O Configuration Display</th>
</tr>
</thead>
<tbody>
<tr>
<td>From the menu bar, click [Diagnosis] and then click [I/O Module].</td>
</tr>
</tbody>
</table>

#### Usage I/O Display

The I/O that is used by the CPU is displayed in yellow.

#### Alarm Display

A module that displays red denotes an alarm.

![I/O Module Configuration Display](image-url)
CAUTION

- If the common area size is changed, “registration” must be carried out. Take care as the size will be changed if “Registration” is not done.
- If the setup value of the common area size is not correct, the message “Area size specification error” will be displayed.
7.8 Date Set

Change the date on the clock, which is operated in the BASIC CPU module. The corrected value should be set in a numeric value (unit: seconds) every 24 hours. In the case where the clock runs fast, enter the changed value by placing a minus sign (-) before the numeric value. On the other hand, in the case where the clock is slow, enter the changed value without placing any sign.

![Date Set Dialog]

- **Display of Date Setting**
  1. To display the date setting dialog box:
     - From the menu bar, click [Diagnostics] and then clock [Set date].
     - The “Date Set Dialog” is displayed.

- **Change of Date**
  1. Enter the corrected date value.
     - Click the Input Area with the mouse, and the cursor is displayed.
     - Input from the cursor position.
  2. Check the change.
     - Check the left side check box of the data that changes.
  3. Setup
     - Click [Setup].
     - The items that are checked in the check box undergo a setting change.
CAUTION

- The contents are set in the BASIC CPU module by [Set]. In the case where [Set] is not carried out, it returns to the original setup.
  When the contents are not set in the BASIC CPU module, click [Cancel].
- The correction of the clock as per the corrected value that has been entered is executed the next time the CPU is restarted.
- The restart of the CPU can be executed either by using the CPU restart function or turning on the FA-M3 power.
CONTENTS

Appendix 1.  Error Codes ................................................................. A1-1
  Appendix 1.1  YM-BASIC/FA Error Codes ................................................. A1-1
  Appendix 1.2  Detailed Error Codes ...................................................... A1-5
Appendix 2.  Reserved Words ............................................................. A2-1
## Appendix 1. Error Codes

### Appendix 1.1 YM-BASIC/FA Error Codes

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Error Message</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>System errors</strong></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Intermediate code not interpreted</td>
</tr>
<tr>
<td>2</td>
<td>Parameter error</td>
</tr>
<tr>
<td>3</td>
<td>Timer error</td>
</tr>
<tr>
<td><strong>Syntax error</strong></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Incorrect syntax (invalid operand specified)</td>
</tr>
<tr>
<td>5</td>
<td>RESERVED</td>
</tr>
<tr>
<td>6</td>
<td>Executing TRANSFER statement (attempted to access module while executing TRANSFER statement)</td>
</tr>
<tr>
<td><strong>Computation error</strong></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Computation overflow</td>
</tr>
<tr>
<td>8</td>
<td>Division by zero</td>
</tr>
<tr>
<td>9</td>
<td>Underflow error</td>
</tr>
<tr>
<td>10</td>
<td>Integer overflow</td>
</tr>
<tr>
<td>11</td>
<td>Unmatched data type (incorrect data results)</td>
</tr>
<tr>
<td>12</td>
<td>Numeric value abnormal or argument (function) error</td>
</tr>
<tr>
<td>13</td>
<td>Unallocated variable</td>
</tr>
<tr>
<td>14</td>
<td>Undefined variable name</td>
</tr>
<tr>
<td>15</td>
<td>String concatenate overflow</td>
</tr>
<tr>
<td>16</td>
<td>Cannot assign</td>
</tr>
<tr>
<td><strong>Array variable declaration error</strong></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>Invalid subscript value (less than 0 or above 32768) in ALLOCATE statement</td>
</tr>
<tr>
<td>18</td>
<td>Array size exceeded</td>
</tr>
<tr>
<td>19</td>
<td>Common area size exceeded</td>
</tr>
<tr>
<td><strong>Used-defined function error</strong></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>Undefined function</td>
</tr>
<tr>
<td>21</td>
<td>RESERVED</td>
</tr>
<tr>
<td>22</td>
<td>Wrong number of arguments</td>
</tr>
<tr>
<td>23</td>
<td>RESERVED</td>
</tr>
<tr>
<td>24</td>
<td>Attempt to redefine a previously defined function</td>
</tr>
<tr>
<td><strong>Reference to array variable</strong></td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>Array subscript out of range</td>
</tr>
<tr>
<td>26</td>
<td>Array subscript dimension error</td>
</tr>
<tr>
<td><strong>FOR - NEXT error</strong></td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>Improper FOR ~ NEXT, WHILE ~ END WHILE, IF ~ ENDIF matching</td>
</tr>
<tr>
<td>28</td>
<td>Needs simple numeric variable in FOR statement</td>
</tr>
<tr>
<td>29</td>
<td>RESERVED</td>
</tr>
<tr>
<td>Error Code</td>
<td>Error Message</td>
</tr>
<tr>
<td>------------</td>
<td>---------------------------------------------</td>
</tr>
<tr>
<td>30</td>
<td>Too many entries (excessive input data)</td>
</tr>
<tr>
<td>31</td>
<td>Too few entries (insufficient data)</td>
</tr>
<tr>
<td>32</td>
<td>No data to be read by READ statement</td>
</tr>
<tr>
<td>33</td>
<td>Unmatched</td>
</tr>
<tr>
<td>34</td>
<td>RESERVED</td>
</tr>
<tr>
<td>35</td>
<td>Improper match between IMAGE and data item</td>
</tr>
<tr>
<td>36</td>
<td>RESERVED</td>
</tr>
<tr>
<td>37</td>
<td>RESERVED</td>
</tr>
<tr>
<td>38</td>
<td>RESERVED</td>
</tr>
<tr>
<td>39</td>
<td>Output data overflow</td>
</tr>
<tr>
<td>40</td>
<td>Value overflow</td>
</tr>
<tr>
<td>41</td>
<td>RESERVED</td>
</tr>
<tr>
<td>42</td>
<td>IMAGE statement missing in a line referenced</td>
</tr>
<tr>
<td>43</td>
<td>Branch destination for GOTO/GOSUB statement not found</td>
</tr>
<tr>
<td>44</td>
<td>Program statement name not found</td>
</tr>
<tr>
<td>45</td>
<td>Invalid position of OPTION BASE statement</td>
</tr>
<tr>
<td>46</td>
<td>Invalid position of DEFINT, DEFLNG, DEFSNG, DEFDBL statements</td>
</tr>
<tr>
<td>47</td>
<td>Invalid operand in DEFINT, DEFLNG, DEFSNG, DEFDBL statements</td>
</tr>
<tr>
<td>48</td>
<td>RESERVED</td>
</tr>
<tr>
<td>49</td>
<td>RESERVED</td>
</tr>
<tr>
<td>50</td>
<td>RESERVED</td>
</tr>
<tr>
<td>51</td>
<td>RESERVED</td>
</tr>
<tr>
<td>52</td>
<td>Communications error/CPU type error</td>
</tr>
<tr>
<td>53</td>
<td>Communications error/CPU type error</td>
</tr>
<tr>
<td>54</td>
<td>Communications error/CPU type error (refer to detail error codes)</td>
</tr>
<tr>
<td>55</td>
<td>SIGNAL transmission error</td>
</tr>
<tr>
<td>56</td>
<td>RESERVED</td>
</tr>
<tr>
<td>57</td>
<td>RESERVED</td>
</tr>
<tr>
<td>58</td>
<td>RESERVED</td>
</tr>
<tr>
<td>59</td>
<td>RESERVED</td>
</tr>
<tr>
<td>60</td>
<td>RESERVE, RELEASE statement error</td>
</tr>
<tr>
<td>61</td>
<td>Invalid numeric value in ON statement</td>
</tr>
<tr>
<td>62</td>
<td>Existing variable or array redeclared</td>
</tr>
<tr>
<td>63</td>
<td>RESERVED</td>
</tr>
<tr>
<td>64</td>
<td>Incorrect RETURN statement (RETURN with no GOSUB)</td>
</tr>
<tr>
<td>65</td>
<td>RESERVED</td>
</tr>
<tr>
<td>66</td>
<td>RESERVED</td>
</tr>
<tr>
<td>67</td>
<td>Numeric conversion error</td>
</tr>
<tr>
<td>68</td>
<td>Error detected in ON ERROR processing</td>
</tr>
<tr>
<td>69</td>
<td>RESERVED</td>
</tr>
<tr>
<td>70</td>
<td>Attempt to execute a statement not executable.</td>
</tr>
<tr>
<td>71</td>
<td>RESERVED</td>
</tr>
<tr>
<td>72</td>
<td>RESERVED</td>
</tr>
<tr>
<td>73</td>
<td>RESERVED</td>
</tr>
<tr>
<td>74</td>
<td>RESERVED</td>
</tr>
<tr>
<td>75</td>
<td>Prerun error, conflicting program</td>
</tr>
<tr>
<td>76</td>
<td>Conflicting error in main program</td>
</tr>
<tr>
<td>77</td>
<td>RESERVED</td>
</tr>
<tr>
<td>78</td>
<td>Syntax error in subprogram</td>
</tr>
<tr>
<td>Error Code</td>
<td>Error Message</td>
</tr>
<tr>
<td>------------</td>
<td>-------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Memory overflow</td>
<td></td>
</tr>
<tr>
<td>80</td>
<td>Stack area is now being used (insufficient area).</td>
</tr>
<tr>
<td>81</td>
<td>Insufficient area for reserving variable areas</td>
</tr>
<tr>
<td>82</td>
<td>I/O error (see detailed error codes)</td>
</tr>
<tr>
<td>File or library error</td>
<td></td>
</tr>
<tr>
<td>83</td>
<td>File name (program name) already exists.</td>
</tr>
<tr>
<td>84</td>
<td>File name (program name) not found. File name contains double-width spaces.</td>
</tr>
<tr>
<td>85</td>
<td>END OF FILE</td>
</tr>
<tr>
<td>86</td>
<td>RESERVED</td>
</tr>
<tr>
<td>87</td>
<td>RESERVED</td>
</tr>
<tr>
<td>88</td>
<td>Error generated in library 88-7n n: n-th parameter error</td>
</tr>
<tr>
<td>92</td>
<td>Resident mode error</td>
</tr>
<tr>
<td>93</td>
<td>Resident program size too large</td>
</tr>
<tr>
<td>94</td>
<td>Resident program upload error</td>
</tr>
<tr>
<td>95</td>
<td>CHAIN statement execution error</td>
</tr>
<tr>
<td>System error</td>
<td></td>
</tr>
<tr>
<td>101</td>
<td>Insufficient dynamic free area</td>
</tr>
<tr>
<td>102</td>
<td>Variable name not found in the symbol table</td>
</tr>
<tr>
<td>103</td>
<td>Numeric data value abnormal</td>
</tr>
<tr>
<td>104</td>
<td>Numeric data value abnormal</td>
</tr>
<tr>
<td>105</td>
<td>Source line length exceeds 306 bytes.</td>
</tr>
<tr>
<td>106</td>
<td>Non-reserved word in program code</td>
</tr>
<tr>
<td>107</td>
<td>Error in analysis of a computational expression</td>
</tr>
<tr>
<td>108</td>
<td>Reserved word code not found</td>
</tr>
<tr>
<td>109</td>
<td>Coding error in coded text</td>
</tr>
<tr>
<td>110</td>
<td>Parameter error in subroutine call</td>
</tr>
<tr>
<td>111</td>
<td>Editor error</td>
</tr>
<tr>
<td>112</td>
<td>Editor error</td>
</tr>
<tr>
<td>113</td>
<td>Editor error</td>
</tr>
<tr>
<td>114</td>
<td>Editor error</td>
</tr>
<tr>
<td>Command error</td>
<td></td>
</tr>
<tr>
<td>115</td>
<td>Command entry validity error, command-disabled state due to Type SS, US, or SB program</td>
</tr>
<tr>
<td>116</td>
<td>Syntax error</td>
</tr>
<tr>
<td>117</td>
<td>Statement not found</td>
</tr>
<tr>
<td>Invalid line number</td>
<td></td>
</tr>
<tr>
<td>118</td>
<td>Line number not found</td>
</tr>
<tr>
<td>119</td>
<td>Invalid line number</td>
</tr>
<tr>
<td>120</td>
<td>Line number exceeds 65535.</td>
</tr>
<tr>
<td>Error Code</td>
<td>Error Message</td>
</tr>
<tr>
<td>------------</td>
<td>---------------</td>
</tr>
<tr>
<td>121</td>
<td>Character string not found</td>
</tr>
<tr>
<td>122</td>
<td>Free area not more than 400 bytes</td>
</tr>
<tr>
<td>Subprogram error</td>
<td></td>
</tr>
<tr>
<td>123</td>
<td>Subprogram not found</td>
</tr>
<tr>
<td>124</td>
<td>Unable to assign new subblock. Illegal replacement of SUB statement</td>
</tr>
<tr>
<td>Syntax error</td>
<td></td>
</tr>
<tr>
<td>130</td>
<td>Unable to assign stack area</td>
</tr>
<tr>
<td>131</td>
<td>Invalid array variable in a statement</td>
</tr>
<tr>
<td>132</td>
<td>GO, GOTO, GOSUB not described in ON statement.</td>
</tr>
<tr>
<td>133</td>
<td>Invalid timer number</td>
</tr>
<tr>
<td>134</td>
<td>RESERVED</td>
</tr>
<tr>
<td>135</td>
<td>Variable name other than that described as operand was used.</td>
</tr>
<tr>
<td>136</td>
<td>Incorrect line number or comma described as operand</td>
</tr>
<tr>
<td>137</td>
<td>Invalid significant character string</td>
</tr>
<tr>
<td>138</td>
<td>Invalid program name</td>
</tr>
<tr>
<td>139</td>
<td>FROM or INTO operand in TRANSFER statement.</td>
</tr>
<tr>
<td>140</td>
<td>TO operand missing (FOR statement, etc.)</td>
</tr>
<tr>
<td>141</td>
<td>RESERVED</td>
</tr>
<tr>
<td>142</td>
<td>RESERVED</td>
</tr>
<tr>
<td>143</td>
<td>Invalid operand description</td>
</tr>
<tr>
<td>144</td>
<td>Computational expression error</td>
</tr>
<tr>
<td>145</td>
<td>RESERVED</td>
</tr>
<tr>
<td>146</td>
<td>RESERVED</td>
</tr>
<tr>
<td>147</td>
<td>Invalid data list</td>
</tr>
<tr>
<td>148</td>
<td>Invalid #Tn or #Un description</td>
</tr>
<tr>
<td>149</td>
<td>Invalid binary constant</td>
</tr>
<tr>
<td>150</td>
<td>RESERVED</td>
</tr>
<tr>
<td>151</td>
<td>Invalid line number or label</td>
</tr>
<tr>
<td>152</td>
<td>Too many variable names or labels</td>
</tr>
<tr>
<td>153</td>
<td>Invalid variable type declaration</td>
</tr>
<tr>
<td>154</td>
<td>Nesting of IF statement exceeds 16 levels.</td>
</tr>
<tr>
<td>155</td>
<td>THEN without processing statement</td>
</tr>
<tr>
<td>156</td>
<td>ELSE without corresponding IF</td>
</tr>
<tr>
<td>157</td>
<td>ENDIF without corresponding IF</td>
</tr>
<tr>
<td>158</td>
<td>Statement not terminated in correct format</td>
</tr>
<tr>
<td>159</td>
<td>Left part or “=” does not appear in computational expression</td>
</tr>
<tr>
<td>160</td>
<td>Invalid DEF statement</td>
</tr>
<tr>
<td>161</td>
<td>Permissible numeric size exceeded</td>
</tr>
<tr>
<td>162</td>
<td>Invalid hexadecimal constant</td>
</tr>
<tr>
<td>163</td>
<td>Invalid FIND command operand</td>
</tr>
<tr>
<td>164</td>
<td>Undefined statement type</td>
</tr>
<tr>
<td>165</td>
<td>Command headed by a line number</td>
</tr>
<tr>
<td>166</td>
<td>Not executed with immediately executable statement</td>
</tr>
<tr>
<td>167</td>
<td>Six or more LF codes found between significant characters other than blank characters</td>
</tr>
<tr>
<td>168</td>
<td>Invalid statement number</td>
</tr>
<tr>
<td>169</td>
<td>Coded statement area (514 bytes) not assigned</td>
</tr>
<tr>
<td>170</td>
<td>Statement not allowed in multiple statement line</td>
</tr>
<tr>
<td>171</td>
<td>Data type error</td>
</tr>
<tr>
<td>172</td>
<td>Invalid subscript</td>
</tr>
<tr>
<td>173</td>
<td>Operand error (incorrect operator described in character expression)</td>
</tr>
<tr>
<td>174</td>
<td>Variable, label or line number not defined in an immediately executable statement</td>
</tr>
</tbody>
</table>
Appendix 1.2 Detailed Error Codes

The following detailed error codes are output in cases where the BASIC error code 082 (I/O error), 054 (shared access error) or 055 (SIGNAL transmission error) appears.

- **I/O errors**

  82 - xx

  The following describes the meaning of error codes and probable causes where errors are output when accessing the Sequence CPU Module and Contact Input/Output Module.

  Where errors appear when accessing another module, see applicable instruction manual.

<table>
<thead>
<tr>
<th>Detailed Error Code (expressed in hexadecimal)</th>
<th>Error Message</th>
<th>Probable Cause</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>No driver exists.</td>
<td></td>
</tr>
<tr>
<td>0C</td>
<td>Insufficient system area</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>Driver internal error</td>
<td></td>
</tr>
<tr>
<td>81</td>
<td>No driver exists.</td>
<td></td>
</tr>
<tr>
<td>82</td>
<td>Invalid function</td>
<td>• Attempted to execute statement which module does not support (OUTPUT statement for input module, etc.). • Incorrect I/O module slot number</td>
</tr>
<tr>
<td>83</td>
<td>Invalid logical file number</td>
<td></td>
</tr>
<tr>
<td>84</td>
<td>Invalid buffer length</td>
<td>• Invalid number of devices</td>
</tr>
<tr>
<td>85</td>
<td>Invalid parameter</td>
<td>• Invalid parameter</td>
</tr>
<tr>
<td>86</td>
<td>Invalid parameter address</td>
<td></td>
</tr>
<tr>
<td>91</td>
<td>Invalid parameter</td>
<td>• Device number out of range • Invalid parameter</td>
</tr>
<tr>
<td>92</td>
<td>Invalid data setting</td>
<td>• Data setting incorrect (data type, etc.) • Specified terminal number other than 0 or 1</td>
</tr>
<tr>
<td>93</td>
<td>Invalid command description</td>
<td>• Invalid format specified</td>
</tr>
<tr>
<td>94</td>
<td>Invalid module specified</td>
<td>• Invalid module name assigned • Module name not assigned</td>
</tr>
<tr>
<td>95</td>
<td>Invalid bit pattern</td>
<td></td>
</tr>
<tr>
<td>9A</td>
<td>Number of specified processing requests exceeded</td>
<td></td>
</tr>
<tr>
<td>9B</td>
<td>Number of specified processing requests exceeded</td>
<td></td>
</tr>
<tr>
<td>9C</td>
<td>Internal error in ASSIGN statement</td>
<td></td>
</tr>
<tr>
<td>9D</td>
<td>ASSIGN statement not executed, I/O not installed</td>
<td>• Incorrect module name assigned • Module name not assigned • Sequence CPU module not found</td>
</tr>
<tr>
<td>Detailed Error Code (expressed in hexadecimal)</td>
<td>Error Message</td>
<td>Probable Cause</td>
</tr>
<tr>
<td>-----------------------------------------------</td>
<td>--------------</td>
<td>----------------</td>
</tr>
<tr>
<td>A1</td>
<td>Incorrect slot number</td>
<td></td>
</tr>
<tr>
<td>B0</td>
<td>Invalid access procedure</td>
<td></td>
</tr>
<tr>
<td>B1</td>
<td>Invalid module designation or data number</td>
<td>Incorrect device name</td>
</tr>
<tr>
<td>B2</td>
<td>Data high/low limit overflow</td>
<td></td>
</tr>
<tr>
<td>B3</td>
<td>Invalid device specification</td>
<td>Incorrect read/write units</td>
</tr>
<tr>
<td>B4</td>
<td>Improper number of devices accessed</td>
<td>Incorrect device name</td>
</tr>
<tr>
<td>B5</td>
<td>Invalid data or code</td>
<td></td>
</tr>
<tr>
<td>B6</td>
<td>Invalid module designation</td>
<td></td>
</tr>
<tr>
<td>BB</td>
<td>Invalid interruption code or number</td>
<td></td>
</tr>
<tr>
<td>BC</td>
<td>Interruption code already requested.</td>
<td></td>
</tr>
<tr>
<td>C1</td>
<td>Buffer overflow</td>
<td></td>
</tr>
<tr>
<td>C7</td>
<td>I/O reset detected</td>
<td></td>
</tr>
<tr>
<td>CD</td>
<td>Insufficient area (driver work area)</td>
<td></td>
</tr>
<tr>
<td>D1</td>
<td>Device error</td>
<td>Faulty module</td>
</tr>
<tr>
<td>D2</td>
<td>Data verification error</td>
<td></td>
</tr>
<tr>
<td>D3</td>
<td>Receiving data error</td>
<td></td>
</tr>
<tr>
<td>D4</td>
<td>Communication error</td>
<td></td>
</tr>
<tr>
<td>D5</td>
<td>No terminator in received text</td>
<td></td>
</tr>
<tr>
<td>D6</td>
<td>Communication error</td>
<td></td>
</tr>
<tr>
<td>D7</td>
<td>Hardware error during data transmission</td>
<td></td>
</tr>
<tr>
<td>D8</td>
<td>Received text header information invalid</td>
<td></td>
</tr>
<tr>
<td>D9</td>
<td>Received text header information invalid</td>
<td></td>
</tr>
<tr>
<td>DA</td>
<td>Program initiation text received</td>
<td></td>
</tr>
<tr>
<td>DB</td>
<td>Invalid transmission mode</td>
<td></td>
</tr>
<tr>
<td>DC</td>
<td>Buffer overflow</td>
<td></td>
</tr>
<tr>
<td>E1</td>
<td>DEVICE NOT READY</td>
<td>Module not installed, Incorrect slot number, Faulty module</td>
</tr>
<tr>
<td>E2</td>
<td>DEVICE BUSY</td>
<td>Sequence CPU module cannot receive a BASIC statement.</td>
</tr>
<tr>
<td>E3</td>
<td>Data error</td>
<td>Received statement not processed within the given time.</td>
</tr>
<tr>
<td>E6</td>
<td>Timeout</td>
<td></td>
</tr>
<tr>
<td>E8</td>
<td>Checksum error</td>
<td></td>
</tr>
<tr>
<td>EA</td>
<td>Data overrun</td>
<td></td>
</tr>
<tr>
<td>F1</td>
<td>Statement-execution check error</td>
<td>Sequence CPU module not in statement execution status</td>
</tr>
<tr>
<td>F2</td>
<td>Improper internal status</td>
<td>Statement executed, but sequence CPU module is not in normal status</td>
</tr>
<tr>
<td>F3</td>
<td>Internal error</td>
<td></td>
</tr>
<tr>
<td>F4</td>
<td>Internal error</td>
<td></td>
</tr>
<tr>
<td>F6</td>
<td>Internal error</td>
<td></td>
</tr>
<tr>
<td>FE</td>
<td>Device not configured</td>
<td></td>
</tr>
</tbody>
</table>
• Shared access and SIGNAL transmission errors

54 - xx or 55 - xx

The following describes the meaning of error codes and probable causes where errors appear at the time of shared accessing.

If errors are output at the time of SIGNAL transmission, the following failures may be considered. You should also check parameters, cable connections and the destination status.

• Parameter errors involving unit numbers, or the like
• Cables disconnected
• Terminators not connected (terminators missing)
• Destination module or card defective

<table>
<thead>
<tr>
<th>Detailed Error Code (expressed in hexadecimal)</th>
<th>Error Message</th>
<th>Probable Cause</th>
</tr>
</thead>
<tbody>
<tr>
<td>03</td>
<td>Destination BASIC task not found</td>
<td></td>
</tr>
<tr>
<td>05</td>
<td>Event-receive intermediate buffer overflow in destination BASIC</td>
<td></td>
</tr>
<tr>
<td>82</td>
<td>Internal error</td>
<td></td>
</tr>
<tr>
<td>84</td>
<td>Incorrect buffer length specification</td>
<td>Common variables are not of integer type or long integer type.</td>
</tr>
<tr>
<td>88</td>
<td>Incorrect destination UNIT type (incorrect configuration)</td>
<td></td>
</tr>
<tr>
<td>89</td>
<td>Incorrect own UNIT type</td>
<td></td>
</tr>
<tr>
<td>91</td>
<td>Invalid parameter</td>
<td>Invalid parameter</td>
</tr>
<tr>
<td>92</td>
<td>Invalid data setting</td>
<td>Set data do not match with configuration.</td>
</tr>
</tbody>
</table>
| 9D                                            | Sequence CPU not installed | Incorrect module name assigned
|                                               |                | Sequence CPU not assigned
<p>|                                               |                | Sequence CPU module not found |
| A7                                            | Unconnected UNIT no. specified |                |
| B1                                            | Internal error |                |
| B2                                            | Internal error |                |
| B3                                            | Internal error |                |
| B4                                            | Internal error |                |
| B5                                            | Internal error |                |
| B6                                            | Internal error |                |
| B7                                            | Internal error |                |
| BA                                            | Internal error |                |</p>
<table>
<thead>
<tr>
<th>Detailed Error Code (expressed in hexadecimal)</th>
<th>Error Message</th>
<th>Probable Cause</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1</td>
<td>Destination memory access error</td>
<td></td>
</tr>
<tr>
<td>C3</td>
<td>Internal error</td>
<td></td>
</tr>
<tr>
<td>C4</td>
<td>Internal error</td>
<td></td>
</tr>
<tr>
<td>D1</td>
<td>Error found in a communication line</td>
<td></td>
</tr>
<tr>
<td>D2</td>
<td>Error found in a communication line</td>
<td></td>
</tr>
<tr>
<td>D3</td>
<td>Error found in a communication line</td>
<td></td>
</tr>
<tr>
<td>D4</td>
<td>Error found in a communication line</td>
<td></td>
</tr>
<tr>
<td>D5</td>
<td>Error found in a communication line</td>
<td></td>
</tr>
<tr>
<td>D6</td>
<td>Error found in a communication line</td>
<td></td>
</tr>
<tr>
<td>D7</td>
<td>Error found in a communication line</td>
<td></td>
</tr>
<tr>
<td>D8</td>
<td>Error found in a communication line</td>
<td></td>
</tr>
<tr>
<td>D9</td>
<td>Error found in a communication line</td>
<td></td>
</tr>
<tr>
<td>DA</td>
<td>Error found in a communication line</td>
<td></td>
</tr>
<tr>
<td>DB</td>
<td>Error found in a communication line</td>
<td></td>
</tr>
<tr>
<td>DC</td>
<td>Error found in a communication line</td>
<td></td>
</tr>
<tr>
<td>E1</td>
<td>DEVICE NOT READY</td>
<td>• Module not installed</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Invalid slot number</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Faulty module</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Subunit turned OFF/ON at the time of Shared accessing</td>
</tr>
<tr>
<td>E6</td>
<td>Timeout</td>
<td></td>
</tr>
<tr>
<td>FE</td>
<td>Device not configured</td>
<td>Invalid slot number</td>
</tr>
</tbody>
</table>
# Appendix 2. Reserved Words

<table>
<thead>
<tr>
<th>ABORTIO</th>
<th>COPY</th>
<th>DP</th>
<th>GLOAD</th>
<th>LCOPY</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABS</td>
<td>COS</td>
<td>DU</td>
<td>GMODE</td>
<td>LEFTS</td>
</tr>
<tr>
<td>ALL</td>
<td>GO</td>
<td>GOSUB</td>
<td>LEN</td>
<td>LET</td>
</tr>
<tr>
<td>ALLOCATE</td>
<td>DATA</td>
<td>EDIT</td>
<td>GOTO</td>
<td>LHEX$</td>
</tr>
<tr>
<td>AND</td>
<td>APPEND</td>
<td>DATE$</td>
<td>ELLIPSE</td>
<td>GPOSITION</td>
</tr>
<tr>
<td>ARNAM</td>
<td>DBADD</td>
<td>ELSE</td>
<td>IF</td>
<td>GPRINT</td>
</tr>
<tr>
<td>ASC</td>
<td>DBCLOSE</td>
<td>ELSE</td>
<td>GRAPHIC</td>
<td>IS</td>
</tr>
<tr>
<td>ASGN</td>
<td>DBCOND</td>
<td>ENABLE</td>
<td>GSAVE</td>
<td>LINPUT</td>
</tr>
<tr>
<td>AT</td>
<td>DBDEF</td>
<td>ENABLE INTR</td>
<td>GSAVE</td>
<td>LINPUT#</td>
</tr>
<tr>
<td>ATN</td>
<td>DBDEL</td>
<td>END</td>
<td>GSLECT</td>
<td>NT</td>
</tr>
<tr>
<td>ATTR$</td>
<td>DBDELM</td>
<td>ENDIF</td>
<td>LISTDEV</td>
<td>LOAD</td>
</tr>
<tr>
<td>AUTO</td>
<td>DBDROP</td>
<td>END WHILE</td>
<td>LISTDEV</td>
<td>LOAD</td>
</tr>
<tr>
<td>DBEND</td>
<td>ENTER</td>
<td>HALT</td>
<td>LOCAL</td>
<td>LISTDEV</td>
</tr>
<tr>
<td>DBFIND</td>
<td>EOL</td>
<td>HEX$</td>
<td>LOCAL LOCKOUT</td>
<td>LOAD</td>
</tr>
<tr>
<td>BATCH</td>
<td>DBFLD</td>
<td>ERASE</td>
<td>HINSTR</td>
<td>LOF</td>
</tr>
<tr>
<td>BATCHOFF</td>
<td>DBFLUSH</td>
<td>ERLIST</td>
<td>HLEFT$</td>
<td>LOG</td>
</tr>
<tr>
<td>BCD</td>
<td>DBGET</td>
<td>ERRC</td>
<td>HLEN</td>
<td>LROTATE</td>
</tr>
<tr>
<td>BEEP</td>
<td>DBGETM</td>
<td>ERRCE</td>
<td>HMID$</td>
<td>LSHIFT</td>
</tr>
<tr>
<td>BFORMAT</td>
<td>DBINIT</td>
<td>ERRCS</td>
<td>HRIGHT$</td>
<td>LSHIFT</td>
</tr>
<tr>
<td>BINEND</td>
<td>DBOPEN</td>
<td>ERRL</td>
<td>MAINTENANCE</td>
<td>LOAD</td>
</tr>
<tr>
<td>BINNUT</td>
<td>DBORDER</td>
<td>ERRMS</td>
<td>MAINTENANCE</td>
<td>LOAD</td>
</tr>
<tr>
<td>BINOR</td>
<td>DBPUT</td>
<td>EXOR</td>
<td>IF</td>
<td>MASTER</td>
</tr>
<tr>
<td>BINXOR</td>
<td>DBPUTM</td>
<td>EXP</td>
<td>IMAGE</td>
<td>Mappable</td>
</tr>
<tr>
<td>BIT</td>
<td>DBRLS</td>
<td>IF</td>
<td>IMAGE</td>
<td>MERGE</td>
</tr>
<tr>
<td>BLEN</td>
<td>DBUPD</td>
<td>INPUT</td>
<td>IMAGE</td>
<td>MERGE</td>
</tr>
<tr>
<td>BOOT</td>
<td>DEF</td>
<td>FIELD#</td>
<td>INPUT#</td>
<td>MOVE</td>
</tr>
<tr>
<td>BOX</td>
<td>DEFAULT OFF</td>
<td>FILES</td>
<td>INPUT$</td>
<td>MOVE</td>
</tr>
<tr>
<td>BOX FILL</td>
<td>DEFAULT ON</td>
<td>FIND</td>
<td>INSTR</td>
<td>MOVE</td>
</tr>
<tr>
<td>BYE</td>
<td>DEFDBL</td>
<td>FOR</td>
<td>INT</td>
<td>NAME</td>
</tr>
<tr>
<td>CALL</td>
<td>DEFINT</td>
<td>FREE</td>
<td>IOSAVE</td>
<td>NEW</td>
</tr>
<tr>
<td>CALLLIB</td>
<td>DEFLNG</td>
<td>FROM</td>
<td>IOSAVE</td>
<td>NEWL</td>
</tr>
<tr>
<td>CHAIN</td>
<td>DEFSHORT</td>
<td>FUSING</td>
<td>IOLOAD</td>
<td>NL</td>
</tr>
<tr>
<td>CHAR</td>
<td>DEFSNG</td>
<td>NOT</td>
<td>IOLOAD</td>
<td>NL</td>
</tr>
<tr>
<td>CHG</td>
<td>DEFDAT</td>
<td>NOT</td>
<td>IOLOAD</td>
<td>NL</td>
</tr>
<tr>
<td>CHR$</td>
<td>DEL</td>
<td>GCOLOR</td>
<td>KEY IS</td>
<td>LABEL</td>
</tr>
<tr>
<td>CIRCLE</td>
<td>DEL#</td>
<td>GCREADX</td>
<td>KEY IS</td>
<td>LABEL</td>
</tr>
<tr>
<td>CIRCLE FILL</td>
<td>DELF</td>
<td>GREADY</td>
<td>KEY IS</td>
<td>LABEL</td>
</tr>
<tr>
<td>CLOSE</td>
<td>DELP</td>
<td>GCURSOR OFF</td>
<td>LASTBIT</td>
<td>OFF EOT</td>
</tr>
<tr>
<td>COL</td>
<td>DIM</td>
<td>GCURSOR ON</td>
<td>LBCD</td>
<td>OFF ERROR</td>
</tr>
<tr>
<td>COLOR</td>
<td>DISABLE</td>
<td>GDDISPLAY OFF</td>
<td>LBINAND</td>
<td>OFF EVENT</td>
</tr>
<tr>
<td>COM</td>
<td>DISP</td>
<td>GDDISPLAY ON</td>
<td>LBINAND</td>
<td>OFF EVENT</td>
</tr>
<tr>
<td>COM IS</td>
<td>DISP USING</td>
<td>GERASE</td>
<td>LBINOR</td>
<td>OFF GRAPHIC</td>
</tr>
<tr>
<td>CONT</td>
<td>DIV</td>
<td>GET#</td>
<td>LBINXOR</td>
<td>OFF INPUT</td>
</tr>
<tr>
<td>CONTROL</td>
<td>DJLINE</td>
<td>GINIT</td>
<td>LBIT</td>
<td>OFF INT</td>
</tr>
<tr>
<td>Command</td>
<td>Usage</td>
<td>Arguments</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>---------------------</td>
<td>-----------------</td>
<td>-----------</td>
<td>----------------------</td>
<td></td>
</tr>
<tr>
<td>OFF KEY</td>
<td>PI</td>
<td>READ IO</td>
<td>SCREEN TAN</td>
<td></td>
</tr>
<tr>
<td>OFF SEGEV</td>
<td>POINT</td>
<td>RECEIVE</td>
<td>SET MARKER THEN</td>
<td></td>
</tr>
<tr>
<td>OFF SYSEV</td>
<td>POKE</td>
<td>RECOM</td>
<td>SET PAINT TIMES$</td>
<td></td>
</tr>
<tr>
<td>OFF TIME</td>
<td>POLYGON</td>
<td>RELEASE</td>
<td>SET PEN TIMEMS</td>
<td></td>
</tr>
<tr>
<td>OFF TIMEOUT</td>
<td>POLYGON FILL</td>
<td>REM</td>
<td>SET STATUS TO</td>
<td></td>
</tr>
<tr>
<td>OFF TIMER</td>
<td>POSITION</td>
<td>REMOTE</td>
<td>SET TEXT TRACE</td>
<td></td>
</tr>
<tr>
<td>ON</td>
<td>POSITION#</td>
<td>RENAME</td>
<td>SET TIMEOUT TRACEP</td>
<td></td>
</tr>
<tr>
<td>ON COMINT</td>
<td>POSX</td>
<td>RENUM</td>
<td>SETDAY TRACEV</td>
<td></td>
</tr>
<tr>
<td>ON EOF</td>
<td>POSY</td>
<td>REPEAT</td>
<td>SETMD RES TRANSFER</td>
<td></td>
</tr>
<tr>
<td>ON EOT</td>
<td>PR</td>
<td>RESERVE</td>
<td>SETTIME TRIGGER</td>
<td></td>
</tr>
<tr>
<td>ON ERROR</td>
<td>PRCSR/D</td>
<td>RESET</td>
<td>SGN</td>
<td></td>
</tr>
<tr>
<td>ON EVENT</td>
<td>PRCSWT</td>
<td>RESET STATUS SHIFT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ON EEVENT</td>
<td>PRELEASE</td>
<td>RESTORE SIGNAL UN TIL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ON INPUT</td>
<td>PRESERVE</td>
<td>RETRY SIN USING</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ON INT</td>
<td>PRINT</td>
<td>RETURN SLOAD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ON KEY</td>
<td>PRINT BFORMAT</td>
<td>RIGHT$ SPC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ON SEGEVT</td>
<td>PRINT NOFORMAT</td>
<td>RND SQR VAL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ON SYSEV</td>
<td>PRINT USING</td>
<td>RNPAR SSAVE VIEWPORT W</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ON TIME</td>
<td>PRINT#</td>
<td>ROTATE START VOL$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ON TIMEOUT</td>
<td>PRINTER IS</td>
<td>RUN STATUS VOLUMES</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ON TIMER</td>
<td>PROG</td>
<td>STEP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ON UNIT</td>
<td>PSET</td>
<td>STOP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OPEN</td>
<td>PU</td>
<td>SAVE STRS WAIT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OPTION BASE</td>
<td>PUT#</td>
<td>SCRATCH SUB WHILE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OR</td>
<td>SCRATCHP</td>
<td>SUBCOM WINDOW</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OUTPUT</td>
<td>SCRATCHV</td>
<td>SUBEND WRITE IO</td>
<td></td>
<td></td>
</tr>
<tr>
<td>QUIT</td>
<td>SEND</td>
<td>SUBEXIT</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>SEQACTV SWAP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PAINT</td>
<td></td>
<td>SET BLINK</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PAUSE</td>
<td>RANDOMIZE</td>
<td>SET CHAR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PEEK</td>
<td>READ</td>
<td>SET LINE TAB</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
# Revision History

Document Name: BASIC Programming Tool M3 for Windows  
Document No.: IM 34M06Q22-02E

<table>
<thead>
<tr>
<th>Edition</th>
<th>Date</th>
<th>Revised Item</th>
</tr>
</thead>
<tbody>
<tr>
<td>1st</td>
<td>Oct. 1999</td>
<td>New publication</td>
</tr>
<tr>
<td>2nd</td>
<td>Sep. 2012</td>
<td>Updated for R1.04 Version</td>
</tr>
</tbody>
</table>

Written by PLC Solution Department  
IA Systems Business Division  
IA Platform Business Headquarters  
Yokogawa Electric Corporation

Published by Yokogawa Electric Corporation  
2-9-32 Nakacho, Musashino-shi, Tokyo, 180-8750, JAPAN

Printed by Kohoku Publishing & Printing Inc.