

FLASH USB DIRECT Programmer
User Manual

June 6, 2013

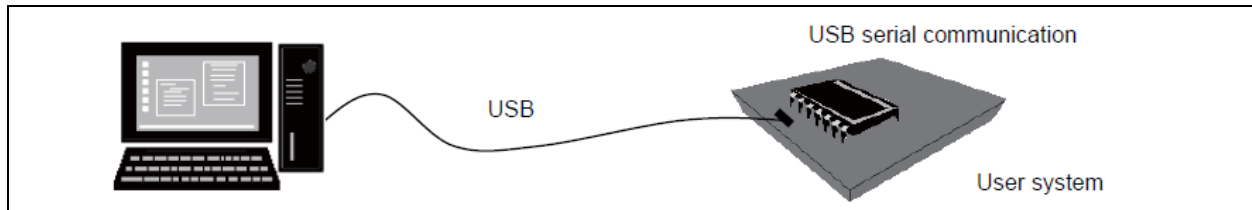
Software version :V01L08

FUJITSU SEMICONDUCTOR LIMITED

Contents

1. Configuration Diagram	1
2. Compatible Microcontrollers.....	1
3. Example of Connection for On-board Programming	2
4. Pins Used	4
5. Installation and Execution of Software.....	5
6. Programmer Functions.....	6
6-1. Downloading	7
6-2. Erasing and Programming	8
6-3. Motorola-S format decoder specification	11
6-4. Intel-Hex format decoder specification	12
7. Operating environment.....	13
8. Others.....	14
9. Cautions	15

1. Configuration Diagram



Using USB cable connected to the personal computer (Windows PC), flash memory data in the microcontroller mounted in the user system can be reprogrammed.

2. Compatible Microcontrollers

FR

MB91F662, MB91F668, MB91F669

FM3/FM4

MB9AF311K, MB9AF312K

MB9AF311L/M/N, MB9AF312L/M/N, MB9AF314L/M/N, MB9AF315M/N, MB9AF316M/N

MB9AF341L/M/N, MB9AF342L/M/N, MB9AF344M/N

MB9AFB41L/M/N, MB9AFB42L/M/N, MB9AFB44L/M/N

MB9BF216S/T, MB9BF217S/T, MB9BF218S/T

MB9BF304N/R, MB9BF305N/R, MB9BF306N/R

MB9BF312N/R, MB9BF314N/R, MB9BF315N/R, MB9BF316N/R

MB9BF316S/T, MB9BF317S/T, MB9BF318S/T

MB9BF321K/L/M, MB9BF322K/L/M, MB9BF324K/L/M

MB9BF328S/T, MB9BF329S/T(*)

MB9BF366M/N/R, MB9BF367M/N/R, MB9BF368M/N/R(*)

MB9BF504N/R, MB9BF505N/R, MB9BF506N/R

MB9BF512N/R, MB9BF514N/R, MB9BF515N/R, MB9BF516N/R

MB9BF516S/T, MB9BF517S/T, MB9BF518S/T

MB9BF521K/L/M, MB9BF522K/L/M, MB9BF524K/L/M

MB9BF528S/T, MB9BF529S/T(*)

MB9BF566M/N/R, MB9BF567M/N/R, MB9BF568M/N/R(*)

MB9BF616S/T, MB9BF617S/T, MB9BF618S/T

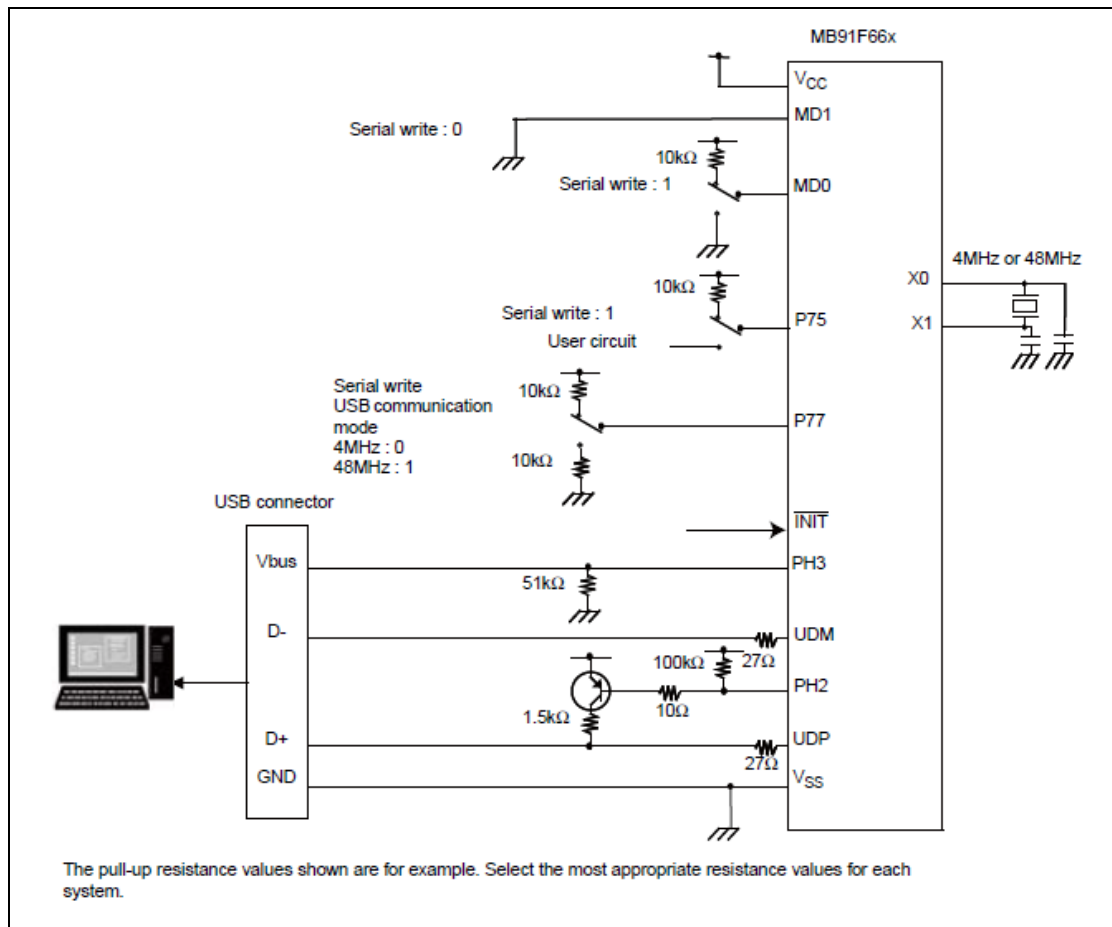
MB9BFD16S/T, MB9BFD17S/T, MB9BFD18S/T

(*: New)

Note: The MCU which has suffix (A, B, etc.) as end of Product Name, and when it's equivalent of the MCU which has no suffix, they are not in list.

3. Example of Connection for On-board Programming

For FR(MB91F66x)

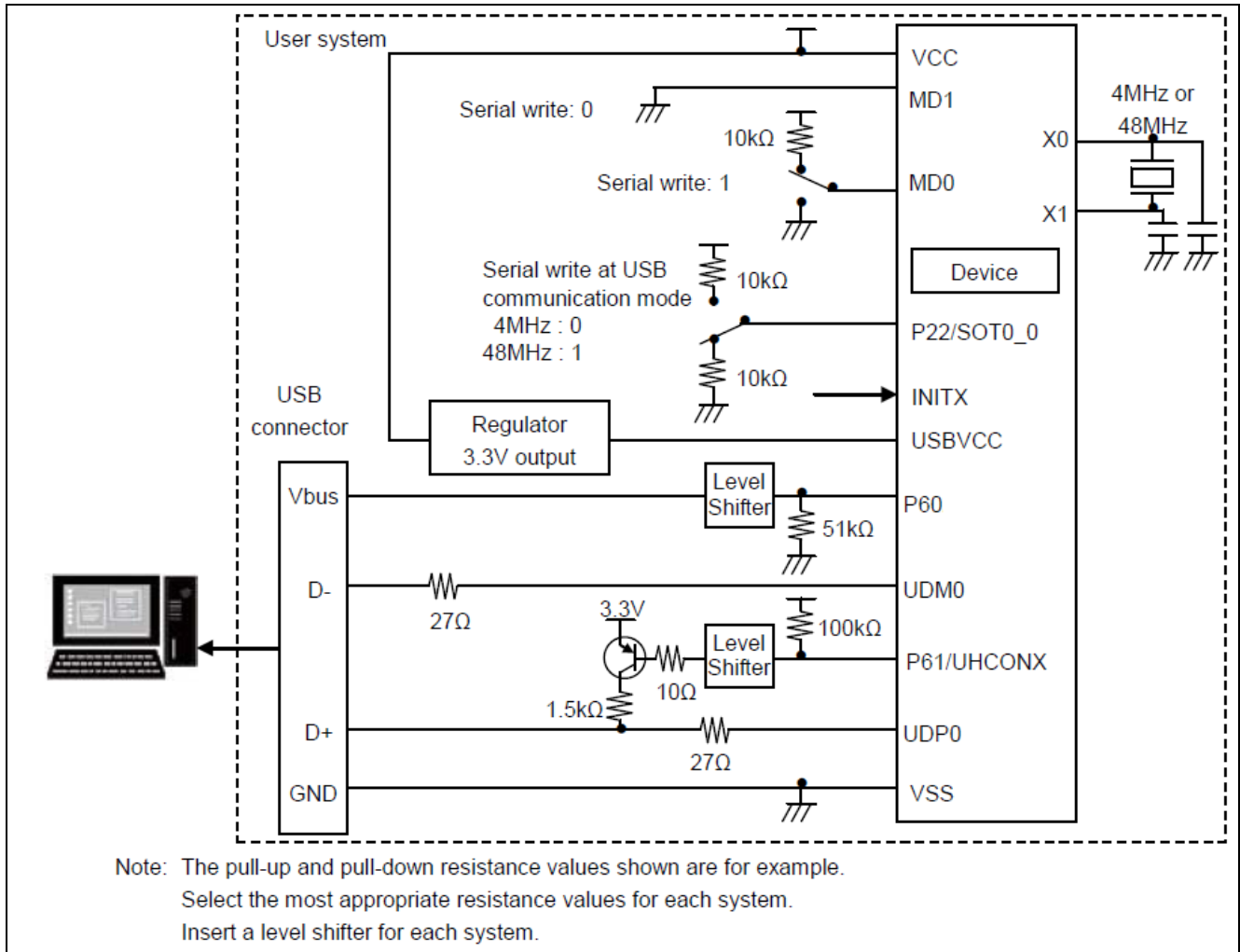


The MD1 and MD0 pins, and Pxx pins cannot be controlled by the PC and should be set in the user system.

During serial reprogramming, when the RSTX pin is set from "Low" to "High" level after setting the MD1 pin and MD0 pin, and Pxx pin, the microcontroller enters the USB reprogramming mode, enabling serial reprogramming from the PC.

After the reprogramming, control is shifted to the normally-used mode as for MD1 and MD0 pins and to the user circuit side as for Pxx pin. Then RSTX pin set from "Low" to "High" level executes user program.

For FM3/FM4



The MD1 and MD0 pins, and Pxx pins cannot be controlled by the PC and should be set in the user system.

During serial reprogramming, when the RSTX pin is set from “Low” to “High” level after setting the MD1 pin and MD0 pin, and Pxx pin, the microcontroller enters the USB reprogramming mode, enabling serial reprogramming from the PC.

After the reprogramming, control is shifted to the normally-used mode as for MD1 and MD0 pins and to the user circuit side as for Pxx pin. Then RSTX pin set from “Low” to “High” level executes user program.

4. Pins Used

(1) Pins used for USB programming (FR(MB91F66x))

Function	Pin name	Supplement
Mode pin	MD1, MD0	Performing an external reset or turning on the power after setting MD0=H and MD1=L enters the serial programming mode.
Communication mode select pin	P75, PH3	Setting it to "H" enables the USB programming mode.
Pull-up control pin for UDP	PH2	This pin controls the pull-up of USB side (+) when the communication mode is USB. PH2 = L : Connect the pull-up resistor PH2 = H : Disconnect the pull-up resistor
USB source oscillating frequency select pin	P77	When the communication mode is set to USB, this pin controls the frequency for source oscillation clock. P77=L : source oscillation frequency : 4MHz P77=H : source oscillation frequency: 48MHz
Reset pin	INITX	-
USB I/O pin	UDP	-
USB I/O pin	UDM	-

(2) Pins used for USB programming (FM3/FM4)

Function	Pin name	Supplement
Mode pin	MD1, MD0	Performing an external reset or turning on the power after setting MD0=H and MD1=L enters the serial programming mode.
Communication mode select pin	P60	Setting it to "H" enables the USB programming mode.
Pull-up control pin for UDP0	P61/HCONX	This pin controls the pull-up of USB side (+) when the communication mode is USB. UHCONX=L : Connect the pull-up resistor UHCONX=H : Disconnect the pull-up resistor
USB source oscillating frequency select pin	P22/SOT0_0	When the communication mode is set to USB, this pin controls the frequency for source oscillation clock. P22=L : source oscillation frequency : 4MHz P22=H : source oscillation frequency: 48MHz
Reset pin	INITX	-
USB I/O pin	UDP0	-
USB I/O pin	UDM0	-

5. Installation and Execution of Software

If the old software version is installed, uninstall it first before installation.

Starting the installer to operate as instructed will complete the installation. Note that the install might not be performed when a directory in a deep nest is specified as the install directory.

After installation, click the Windows *Start* button => *Program* => *FUJITSU USB DIRECT Programmer* => *USBDirect* to start the programmer software.

The device driver for Windows is stored in "Driver" folder of the FLASH USB DIRECT Programmer installation folder. Please use the driver software stored in "Driver" folder.

[for Windows XP]

C:\Program Files\Fujitsu\FUJITSU USB DIRECT Programmer\driver\WinXP

[for Windows 7]

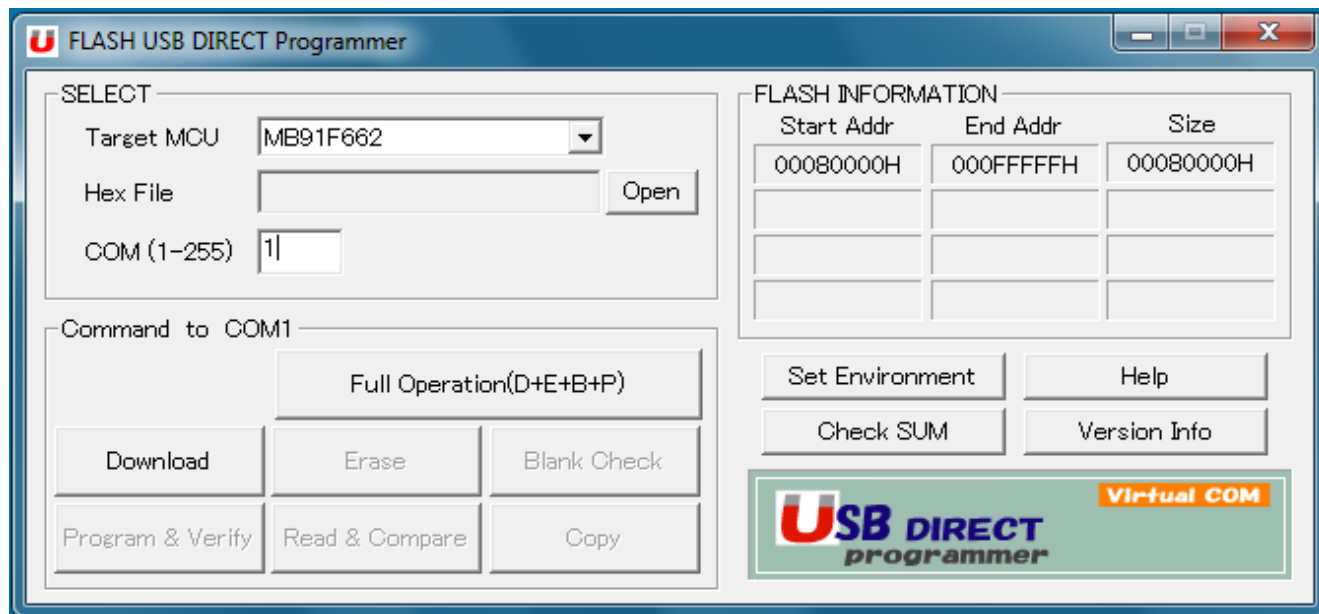
C:\Program Files\Fujitsu\FUJITSU USB DIRECT Programmer\driver\Win7

6. Programmer Functions

Erase, Blank Check, Program & Verify, Read & Compare, and Copy can be executed for flash memory integrated into the microcontroller.

Main dialog box

Programmer software is started to open the dialog box as shown below.



Overview of operating procedure

First, complete setting of the user system (microcontroller board) that data is programmed to (see *Chapter 3*). In starting or when setting has been changed, it is necessary to perform downloading (described later).

After downloading terminates normally, perform procedures such as Erase and Programming.

6-1. Downloading

This section describes the operating procedure for downloading and the operating state of the program.

(a) Set information for target system

(1) Target MCU : Select number of target MCU

(2) COM <1-255> : Input number of virtual COM

[How to confirm of number of virtual COM]

"Control panel" -> "System" --> "Hardware" -> "Device Manager"

In the device manager window, expand the "Ports (COM and LPT)" to see its contents.

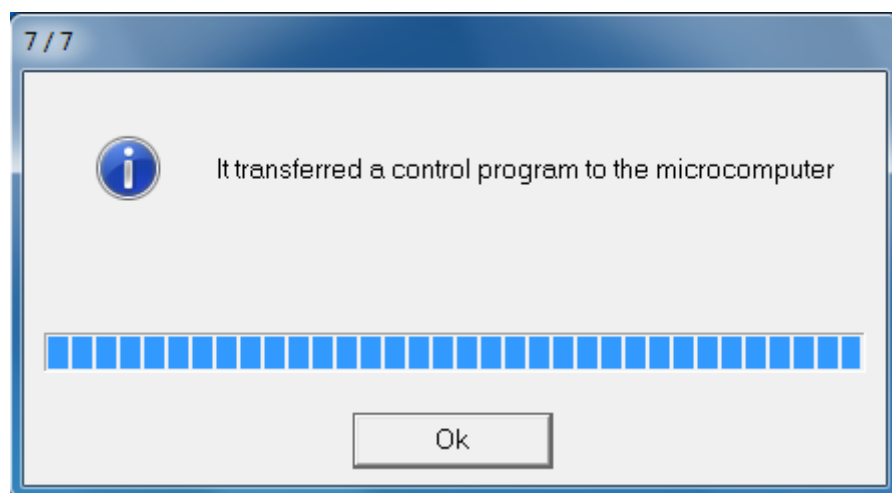
Verify the number denoted by the ** in "USB Serial Port (COM**)"

(d) Execution of downloading

Click the *[Download]* button.

If the following dialog window is opened, Input a reset signal to the microcontroller to start the program in the flash programming mode and then click the *[OK]* button

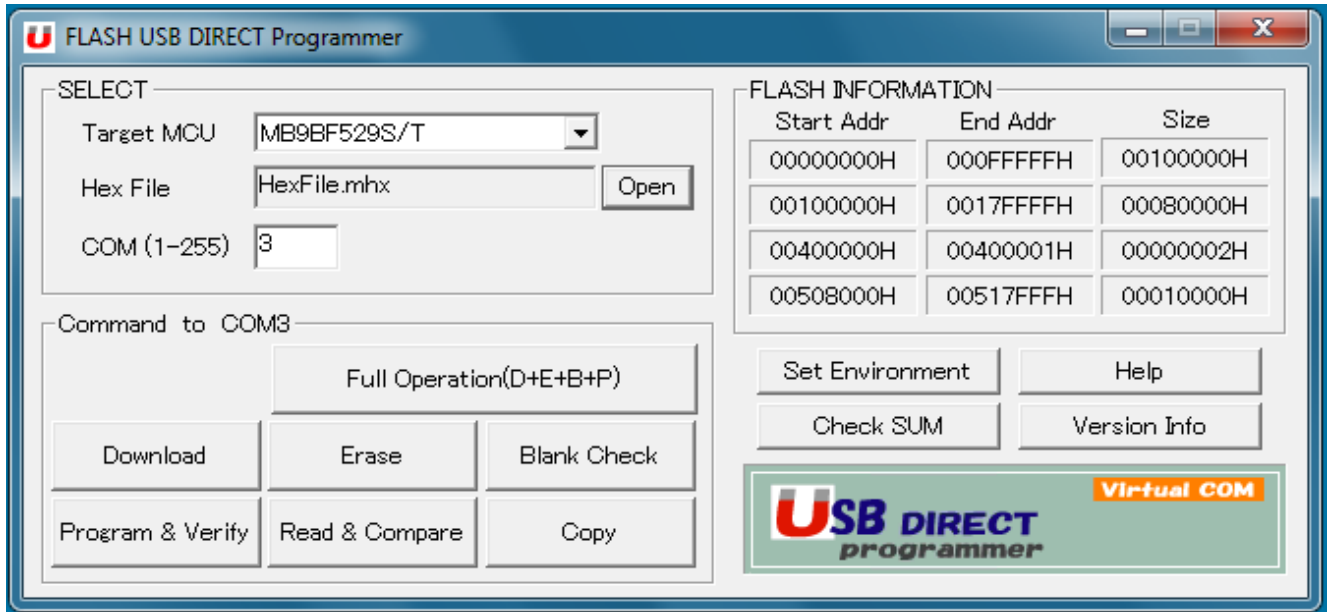
Downloading is performed to open the "Download" window. When downloading is completed normally, the following dialog window opens.



When the *[OK]* button is clicked to close the dialog window, the *[Erase]*, *[Blank Check]*, *[Program & Verify]*, *[Read & Compare]* and *[Copy]* buttons are enabled.

6-2. Erasing and Programming

This section explains how to specify *Hex File* and the processing and operation performed when the *[Erase]*, *[Blank Check]*, *[Program & Verify]*, *[Read & Compare]*, *[Copy]* and *[Full Operation (D+E+B+P)]* buttons are clicked.



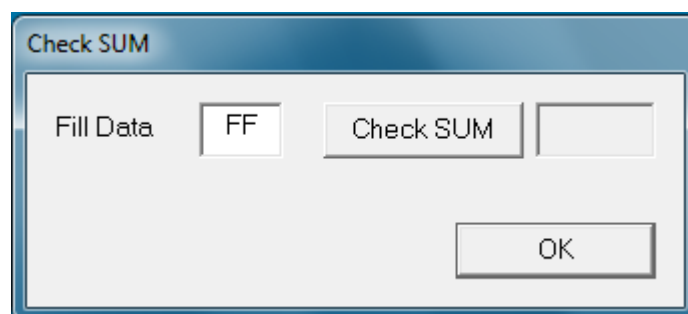
(a) *Hex File*: Select the file to be programmed to flash memory

Specify the Motorola-S or Intel-HEX format file to be programmed to flash memory in the microcontroller. Although the specification method by drags and drops a direct file from Explorer etc. is recommended, it can specify also by the file appointed window displayed by pushing the *[Open]* button.

Hex File must be specified to execute *[Program & Verify]*, *[Read & Compare]* and *[Full Operation (D+E+B+P)]*. Since it is decoded at the head of these processings each time, even if the specified Motorola-S or Intel-HEX format file changes specification of a file just before processing, it is OK.

After Hex File is specified, checksum to ROM image after Motorola-S format file or Intel-HEX format file shown in Hex File is deciphering done can be calculated.

The dialog box to calculate checksum when a lower right *[Check SUM]* button is pushed opens.



The range of the calculation of checksum is limited to the Flash area shown in the upper right of the main dialog. When the area has divided into plural block, the empty area between blocks is not added, and the total of each block is calculated.

The calculation method is simple addition of every one byte, and the result shows the last 4 digits (It is not a complement representation) by the hexadecimal number.

ROM value in the Flash area not shown in Hex File is calculated assuming that it is a value indicated by Fill Data at the left of the dialog. When starting, FF is set here. Please specify it by two hexadecimal number digits when changing.

[Notes concerning checksum]

This function doesn't calculate the checksum of ROM image written in the FLASH memory in the microcontroller chip. When Hex File is not specified, and the error is detected at the decipherment of Hex File, nothing is displayed.

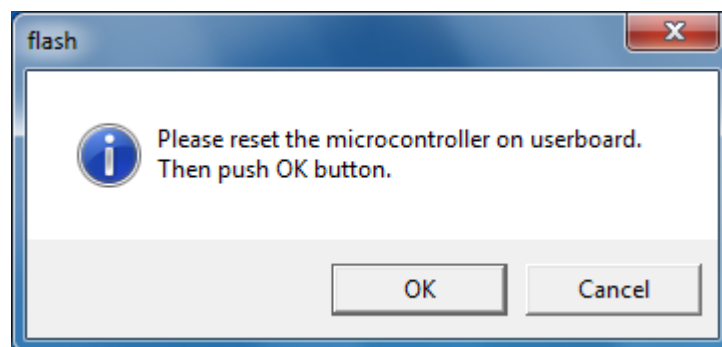
The SUM value calculated here is not peculiar against Hex File. When another microcontroller is selected, same Hex File might reach another value.

The value specified with Fill Data is not written at the time of writing. This value is used only for the calculation of checksum.

(b) **Erase:** Erase all flash memory areas

All flash memory must be in the blank state (0xff) when programming a new program to it. By pushing this button, a chip erase command is published to FLASH and elimination is performed. When the flash memory is protected, the following windows might be displayed while executing the Erase command. If the following dialog window is opened, Input a reset signal to the microcontroller to start the program in the flash programming mode and then click the [OK] button.

In addition, a blank check does not perform this command.

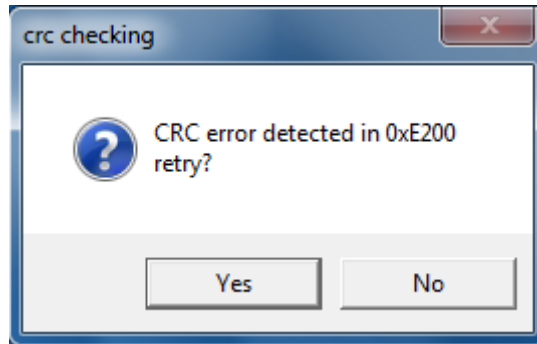


(c) **Blank Check:** Check that all flash memory areas are blank

This button is clicked to check that all flash memory is in the blank state (0xff).

(d) **Program & Verify:** Program data to flash memory

This button is clicked to program the Motorola-S or Intel-HEX format file specified in **Hex File** to flash memory in the microcontroller concurrently with verification. An error dialog is displayed, when writing is performed for 512 bytes of every block and a CRC error is detected by the block.



This dialog If YES is pushed, the block of an error will be resent and it will continue writing. A push on NO interrupts write-in processing.

(e) **Read & Compare:** Compare *Hex File* with data in flash memory in microcontroller

This button is clicked to compare data in the Motorola-S or Intel-HEX format file specified in *Hex File* with data in flash memory in the microcontroller. Like the *[Program & Verify]* processing, The data of FLASH is transmitted for 512 bytes of every block, a CRC error check is performed, and comparison processing is performed.

(f) **Copy:** Save data in flash memory in microcontroller to file

This button is clicked to read data from flash memory integrated into the microcontroller and save it as an Motorola-S or Intel-HEX format file. Like *[Read & Compare]* processing, FLASH memory reading is performed for 512 bytes of every block, and a CRC error check is performed similarly.

The output file format can be changed by right-clicking in the Copy button. The button name changes by *[Copy]* and *[Copy_i]* whenever right-clicking. The state of *[Copy]* shows Motorola-S format, and the state of *[Copy_i]* shows Intel-HEX format.

Processing begins when the button is left-clicked, the folder is specified preservation ahead, the file name is input, and the *[Save]* button is pushed when the form is selected.

(g) **Full Operation (D+E+B+P):** Automatic programming

Operation to *[Download]* to *[Program & Verify]* is performed by package.

In the case of a blank chip, processing is performed in order of *[Download]*, *[Blankcheck]*, and *[Program & Verify]*. When it is not a blank chip, processing is performed in order of *[Download]*, *[Blankcheck]*, *[Erase]*, *[Blankcheck]*, and *[Program & Verify]*.

6-3. Motorola-S format decoder specification

Before programming, Motorola-S format Decoder of programmer changes Motorola S format data into binary data, according to the following specification.

(a) The decoder does not error when overlap of addresses occurs.

The decoder does not error about overlap of address. If user writes a data on an address which was already written another data before, former data is overwritten by new data.

(b) Available address

If user writes a data beyond an address range of FLASH memory, programming results in an error.

(c) About the error detected by the decoder.

The error detected by the decoder is the following (1)-(4). When these errors are detected, processing is interrupted by the decoder. Then the line number and the cause of the error are displayed in the dialog window.

(1) file error

The start of the line is not "S".

(2) S-format error

The start of the line is not "S0", "S1", "S2", "S3", "S5", "S7", "S8" and "S9".

(3) decode error

- There are character except "0123456789ABCDEF". ("S" is excluded. See (1) and (2).) And, the small letter "abcdef" cannot be used.
- The LENGTH data is different from the length of an actual data row.
- The SUM data is different.

(4) address error

There is data besides the FLASH area. (See b.Available address)

(d) Other detail

The decoder skips a line. even if the line is contained only new-line code NL and programming goes on.

A line beginning with "S0", "S5", "S7", "S8" and "S9" is ignored and decoder skips such lines in S format file without error.

6-4. Intel-Hex format decoder specification

Before programming, Intel-Hex format Decoder of programmer changes Intel-Hex format data into binary data, according to the following specification.

(a) The decoder does not error when overlap of addresses occurs.

The decoder does not error about overlap of address. If user writes a data on an address which was already written another data before, former data is overwritten by new data.

(b) Available address

If user writes a data beyond an address range of FLASH memory, programming results in an error.

(c) About the error detected by the decoder.

The error detected by the decoder is the following (1)-(4). When these errors are detected, processing is interrupted by the decoder. Then the line number and the cause of the error are displayed in the dialog window.

(1) file error

The start of the line is not ":".

(2) decode error

There are character except "0123456789ABCDEF". ":" is excluded. See (1).) And, the small letter "abcdef" cannot be used.

The record type is not 00, 01, 02, 03, 04, and 05.

The length of the data string that turns out by the number of data is different from the length of an actual data string.

The SUM data is different.

(3) record error

When not is in number 0 of data at 01 records.

When not is in number 2 of data at 02 records.

When not is in number 4 of data at 03 records.

When not is in number 2 of data at 04 records.

When not is in number 4 of data at 05 records.

(4) address error

There is data besides the FLASH area. (See b.Available address)

7. Operating environment

Required equipment :

- Windows PC with the USB port
- USB cable (commercial item)

OS :

- Windows XP / Windows 7

Memory size :

- More than the memory quantity that OS recommends

Hard disk :

- (Availability) Not less than 10 MB

NOTICE : The operations of all models that meet the above conditions may not be ensured.

8. Others

(A) Error messages

Many error messages are displayed owing to the setting mistake of hardware and software.

the case where an error is outputted in addition even if it checks these in detail, please tell the person in charge of software acquisition origin a detailed condition.

No.	Item	Description
No.001	Message	Download NG!
	Cause	The response of download processing is unusual.
	Action	Please check connection and a setup of hardware.
No.003	Message	Timeout error
	Cause	The response of a command does not come on the contrary.
	Action	Please check connection and a setup of hardware.
No.006	Message	Unable to open COM port
	Cause	Another application is using COM.
	Action	Please check the use situation and port number of a COM port.
No.009	Message	Unable to gain COM port info.
	Cause	It will be in the state where the target COM port can be used.
	Action	Please check the number of a COM port and setup to be used.
No.010	Message	Unable to change COM port setting.
	Cause	A communication setup cannot be set as the target COM port.
	Action	Please inform support of condition.
No.011	Message	Communication error
	Cause	The unusual command response was received.
	Action	Please re-perform by improving connection and a setup of hardware.
No.012	Message	Read error
	Cause	The response at the time of read&compare or copy processing is unusual.
	Action	Please re-perform by improving connection and a setup of hardware.
No.013	Message	Program error
	Cause	The response at the time of Program&Verify processing is unusual.
	Action	Please re-perform by checking whether a chip is blank.
No.015	Message	COM port write error
	Cause	There is the possibility of the abnormalities of a COM port driver or the port itself.
	Action	Please inform support of condition.

9. Cautions

The PC programming software has the possibility of receiving the influence by the communications cable, the outside environment, and the PC.

Therefore, please evaluate it enough when you use the software.

Please use programming systems of programmer venders when you write two or more devices at the same time.

Please don't use the USB HUB between PC and MCU.

The specifications of the product are subject to change without notice.