

# EF1SRP-01U Supplement (M16C/60 Series Edition)

Third edition issued April, 1999

## 1. General Description

This supplement contains information required for reading, writing and clearing data to/from Mitsubishi Electric M16C/60 Series MCUs with built-in flash memory. The supplement also contains a description of command operation for the various functions of the M16C/60 Series.

## 2. Operating Environment

Use the MCUs mentioned in this supplement in an environment as follows.

[EFP-I]

Monitor Version: Ver.2.08.04 or later

[Control Software]

WinEFP Version: Ver.1.02.00 or later

EFP MCU.TBL Version: Ver.2.00.00 or later

SRP MCU.TBL Version: Ver.1.02.00 or later

## 3. Individual Writing Specifications

- 4-wire serial writing
- ID collation function
- Data protection function (block lock)
- Block erase
- Boot area output function
- Page program (256 bytes)

## 4. Pin Connection

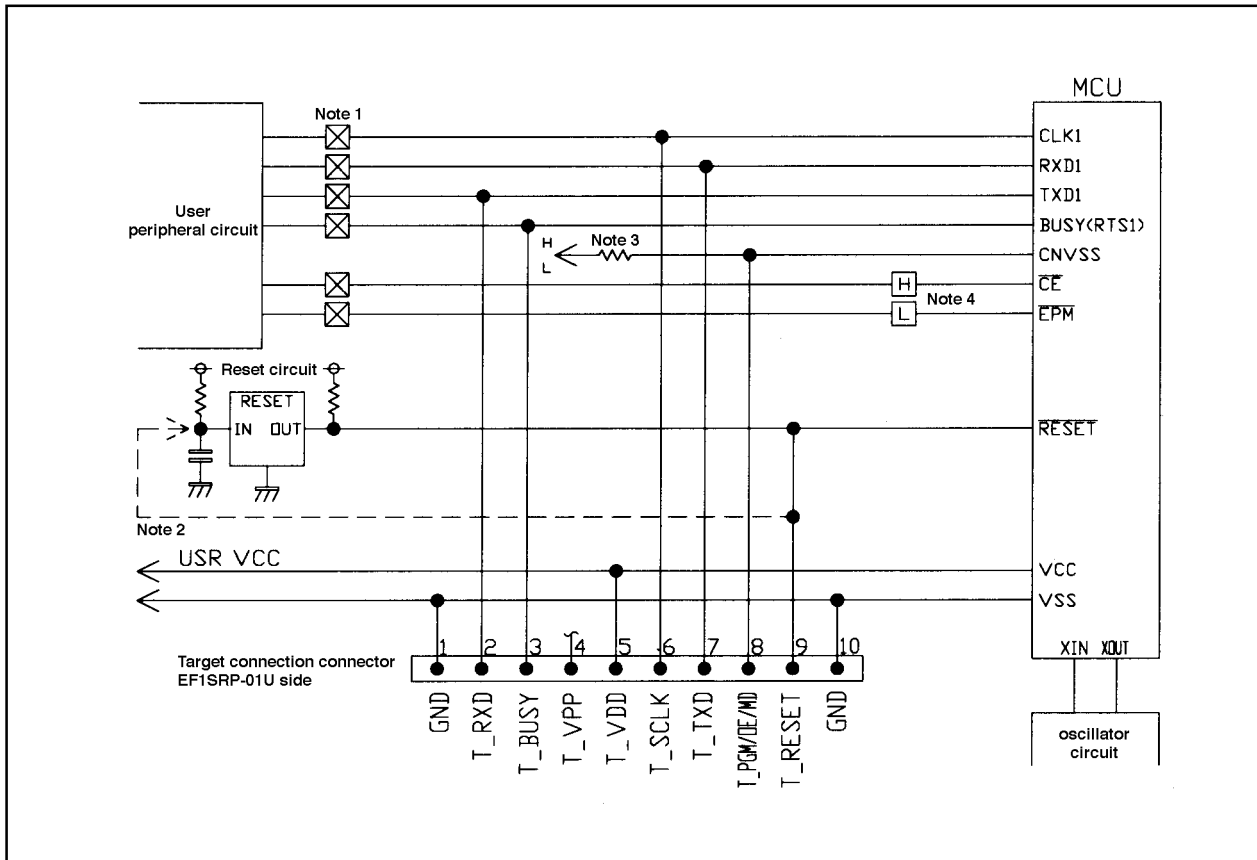
Table 4.1 lists the connection of target connection cable pin of the M16C/60 Series.

**Table 4.1: Connection of the Target Connection Cable Pin**

Pin No. (EF1SRP-01U side)	Target End Wire Color	Signal	4-Wire Cable Pin No.	MCU Connection Pin for Serial Writing
1	Orange/red dotted 1	GND	1	Connects to VSS pin *3
2	Orange/black dotted 1	GND	1	Connects to VSS pin *3
3	Gray/red dotted 1	T_VPP	4	Unconnected
4	Gray/black dotted 1	T_VDD	5	Connects to VCC pin (VCC is used for user power source) *1
8	White/black dotted 1	T_PGM/OE/MD	8	Connects to CNVSS pin
9	Yellow/red dotted 1	T_SCLK	6	Connects to CLK1 pin
10	Yellow/black dotted 1	T_TXD	7	Connects to RXD1 pin
11	Pink/red dotted 1	T_RXD	2	Connects to TXD1 pin
12	Pink/black dotted 1	T_BUSY	3	Connects to BUSY (RTS1) pin
14	Orange/black dotted 2	T_RESET	9	Connects to RESET pin *2
15	Gray/red dotted 2	GND	10	Connects to VSS pin *3
16	Gray/black dotted 2	GND	10	Connects to VSS pin *3

- \*1. Supply VCC from user side to match source voltage of output buffer used on EFP-I side with user side source voltage (VCC).
- \*2. Reset cancel is not carried out following write verify. To execute MCU, you should therefore unplug the target connection cable.
- \*3. The signal GND has 4 pins (No. 1, 2, 15 and 16) of EF1SRP-01U side connector. When connecting to the target board, there is no problem for connecting only one pin, but it is recommended to connect more than 2 pins.
- \*4. When serial writing, connect the  $\overline{CE}$  pin and the  $\overline{EPM}$  pin of the MCU to the VCC pin and the VSS pin respectively.
- \*5. Connect the MCU's Xin and Xout terminals to the oscillator circuit.

(1) An example of target MCU peripheral circuit when using the M16C/60 Series is shown in Fig. 4.1.

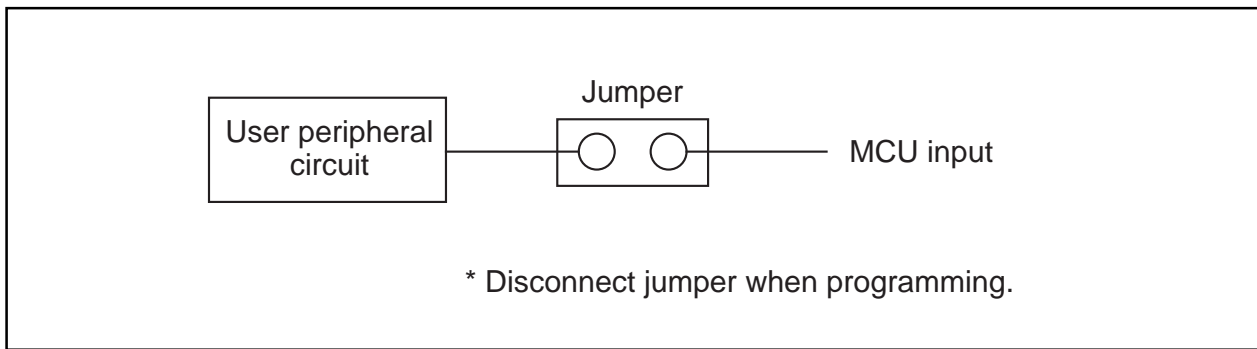


**Fig. 4.1: Target MCU Peripheral Circuit Example**

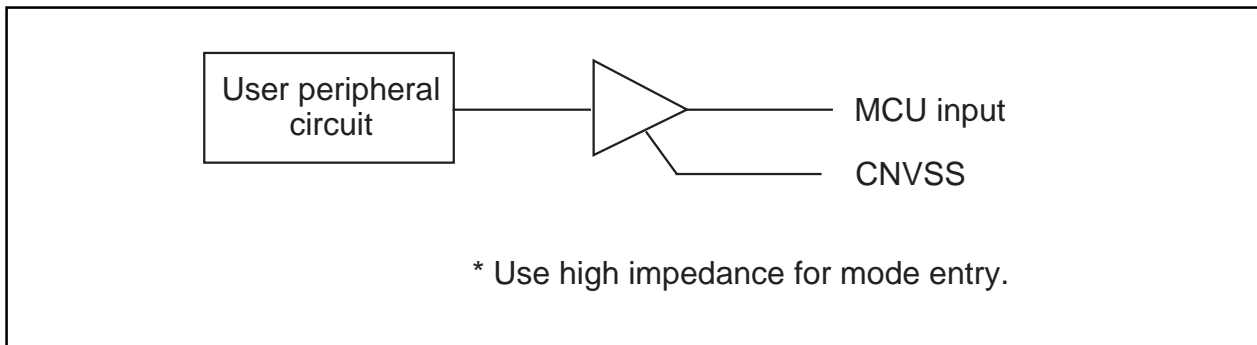
#### Notes

1. If the user peripheral circuit is an output circuit, you should disconnect by jumper to avoid output collision when serial writing.
2. EFP-I side reset output is an open collector, therefore connect directly to the  $\overline{\text{RESET}}$  pin for open collector output. A pull-up resistor however must be connected. If the reset circuit is CMOS output, do as described in note 1, or connect the EFP-I side T\_RESET signal to reset circuit input. Make reset delay within 30ms.
3. Pull-up or pull-down the CNVSS by the resistor at 5.1K $\Omega$  before connecting.
4. When serial writing, fix the  $\overline{\text{CE}}$  and  $\overline{\text{EPM}}$  to "H" and "L" respectively, and in other case, connect to the user peripheral circuit or pull-up.

- (2) An example of a collision prevention circuit when user peripheral circuit outputs is shown in Fig. 4.2 and 4.3.



**Fig. 4.2: Collision Prevention Circuit Using Jumper**



**Fig. 4.3: Collision Prevention Circuit Using Three State Buffer**

## 5. ID Code Area

ID code area is provided for FFFDFH, FFFE3H, FFFE3H, FFFE3H, FFFE3H, FFFF3H, FFFF7H and FFFFBH of the internal flash memory of M16C/60 Series MCUs.

ID data size is fixed at 7 bytes. Set the ID code in each ID area one byte at a time.

MCUs for which an ID code has been written in the ID code area carry out ID code collation and the MCU's internal flash memory cannot be read, written in or cleared unless the ID code matches.

\* Does not include when the ID code area is blank.

ID code area configuration is shown in Fig. 5.1.

FFDFH	ID code (1st byte)
•	•
FFFE3H	ID code (2nd byte)
•	•
FFFE3H	ID code (3rd byte)
•	•
FFFE3H	ID code (4th byte)
•	•
FFFF3H	ID code (5th byte)
•	•
FFFF7H	ID code (6th byte)
•	•
FFFBH	ID code (7th byte)

**Fig. 5.1: ID Code Area Configuration**

## 6. ID Collation Function

The ID code collation is carried out by inputting the ID code etc. for the ID collation parameter of the WinEFP environment setting dialog. Be sure to carry out the ID code collation if an ID has been written in the ID code area of the target MCU.

The commands of [Device] of the WinEFP window menu cannot be used if the ID code does not match.

\* If the ID code area is blank, the ID collation is not carried out even if you input the ID code.

The ID collation parameter configuration is shown in Fig. 6.1.

The image shows a dialog box titled "ID Collation". It contains three input fields: "Input Format(I):" with radio buttons for "ASCII" (selected) and "HEX"; "Start Address(A):" with a text box containing "FFDF"; and "ID Code(C):" with an empty text box. At the bottom right, there are two buttons: "Save(S)..." and "Refer(B)...".

**Fig. 6.1: ID Collation Parameter Configuration**

### (1) Input format

The ID code input format is specified as ASCII or HEX.

### (2) Top address

Specifies the top address of ID code area. The top address is fixed at FFFDFH for the M16C/60 Series.

### (3) ID code

Input the ID code. The number of bytes for ID code input is fixed at 7 bytes.

### (4) Save button

Saves input ID collation parameters in a file. When the save button is clicked, the file section dialog appears, so input the file name of your choice.

### (5) Refer button

Reads the file in which ID collation parameters are saved and sets parameters according to the contents of the file. When the refer button is clicked, the file section dialog appears, so select the file of your choice.

\* An example of the operating procedure for ID collation is shown below. The ID code area of the target MCU's internal flash memory is set as listed in Table 6.1.

**Table 6.1: ID Code Setting**

ID Code Area Address	ID Data
FFDFH	53H
FFFE3H	55H
FFFE3H	49H
FFFE3H	53H
FFFF3H	45H
FFFF7H	49H
FFFFBH	2EH

- (1) Check if FFFDFH is set for the top address parameter. If the device type parameter is set to "M3062xFx", FFFDFH is automatically set as the top address of the code collation parameter. The ID code will not match if an address other than FFFDFH is set.
- (2) Set ASCII or HEX for the input format parameter, and input a 7-byte ID code for the ID code parameter. An example of ID code parameter setting is given below.

For ASCII:

ID Code: SUISEI.

For HEX:

ID Code: 53 55 49 53 45 49 2E  
 S U I S E I .

## 7. Block Set Command

The block set command handles the lock bit of each block.

Setting a block's lock bit to "Lock" allows you to protect the block from being written to or erased (i.e. the block is locked).

### 7.1 Screen Layout

Fig. 7.1 shows the screen layout of the block set command.

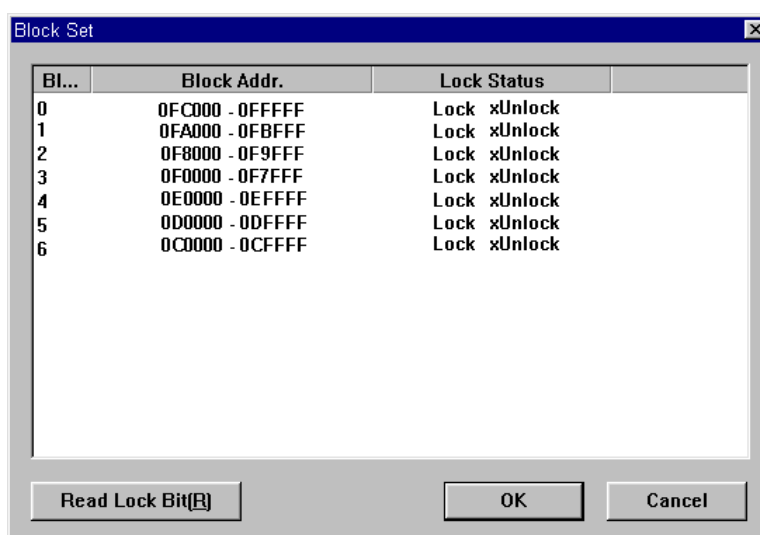


Fig. 7.1: Screen Layout of Block Set Command

(1) Block number

Indicates the block number assigned to each block.

(2) Block address

Indicates the start address and the end address of each block.

(3) Lock status

Indicates the lock bit status of each block.

xLock      Unlock      <-- The lock bit is locked.

Lock      xUnlock      <-- The lock bit is unlocked.

(4) Read Lock Bit button

Reads the content of the lock bit from a target MCU, and indicates the lock bit status according to the content.

(5) OK button

Clicking the OK button writes to the target MCU the contents of the lock bits of the blocks set to "Lock".

(6) Cancel button

Cancels the command.

## 7.2 Setting the lock bit

Here follow the steps to set a lock bit to "Lock".

(1) Position the mouse cursor on an arbitrary line and double-click. Then the lock bit status indicated in the lock status switches. Set it to "Lock".

(2) Click the OK button. Then the content of the lock bit of the block set to "Lock" will be written to the target MCU.

\* Once the content of the lock bit is written to the MCU by use of the OK button, the block set command cannot return a locked block back to the unlocked status.

\* For unlocking a locked block, see 8. Erase Command.

## 8. Erase Command

The erase type parameter contained in the erase command enables you to erase by block or erase all blocks. The erase command parameter input dialog is shown in Fig. 8.1.

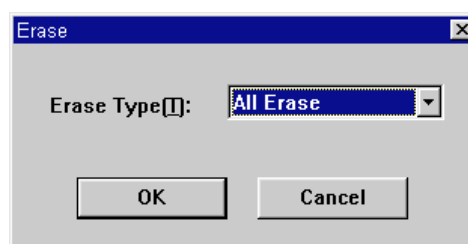


Fig. 8.1: Erase Command Parameter Input Dialog



### (1) Erase type

All Erase and block address area (xxxxxxH - xxxxxxH) are displayed in the drop-down list to the right of the erase type parameter display field (displayed by clicking the arrow pointing downward with the mouse). Select the block erase method.

### (2) OK button

Executes the block erase command.

### (3) Cancel button

Cancels the command.

- \* To erase a locked block, follow the steps given below. These steps also allow you to unlock a locked block.

1) Choose [Environment Settings] from the [Option] menu in the WinEFP window to open the environment settings dialog box.

Set the lock type parameter to "Lock bit ineffective", then click the OK button.

2) Choose [Erase] from the [Device] menu in the WinEFP window to open the erase command parameter input dialog box.

Set the erase type parameter to the locked block, then click the OK button.

## 9. Boot Read Command

Reads the contents of the boot area, and then writes in the EFP-I's internal RAM. Specify FE000H to FFFFFH for the start and end address parameter.

Unfixed data will be read if address outside the start and end address area is specified.

## 10. Parameter Input by Device Command

M16C/60 Series serial writing MCUs read and write data by the page. Because one page of data is 256 bytes, the input format for the start and end addresses is set as follows.

A parameter error results if addresses outside the page unit are input for the start and end addresses.

Input format:

Start address      xxxx00H

End address        xxxxFFH

## 11. Auto Offset Address of EFP-I's Internal RAM

Capacity of the EFP-I's internal RAM is 512K bytes. Thus the area from 0H to 7FFFFH can be used under ordinary circumstances. The address area of the M30624FG internal flash memory is C0000H to FFFFFH, so the HEX offset address of 80000H is automatically set for the EFP-I's internal RAM.

When 80000H is set for the HEX offset, the 0h at the top of the EFP-I's internal RAM changes to 80000H and the end address changes to FFFFFH.

Specify 80000H to FFFFFH for the start and end address parameter in the [Edit] command of the WinEFP window menu.