This manual shows how to program MB91F36x/MB91F46x-derivatives (e.g. MB91F46D) using the Software “FME FR Flash Programmer”.
# Table of Contents

1 **PREPARATIONS** .................................................................................................................................3
  1.1 Software installation ...........................................................................................................................3
  1.2 Hardware Setup : ..............................................................................................................................4

2 **USING THE SOFTWARE** ..................................................................................................................5
  2.1 Getting started .....................................................................................................................................5
  2.2 Using the “Automatic mode” ............................................................................................................6
  2.3 Sub-Menu “External Flash” ............................................................................................................7
  2.4 Sub-Menu “Boot-ROM Functions” ...................................................................................................8
  2.5 Sub-Menu “Flash Functions” ..........................................................................................................9
  2.6 Sub-Menu “Options” .....................................................................................................................11
  2.7 Sub-Menu “Signals” ......................................................................................................................12

3 **APPENDIX** .......................................................................................................................................13
  3.1 Information on Files to be programmed (MHX-Format) ................................................................13
  3.2 Required Linker-Settings ..............................................................................................................13
  3.3 Using “K-line” ..............................................................................................................................13
  3.4 Command line options ..................................................................................................................14

4 **NOTE ON EXTERNAL FLASH PROGRAMMING** .........................................................................18

5 **DISCLAIMER** ..................................................................................................................................19
What you’ll need

- MB9146x Starterkit (or similar hardware) with MB91F46x MCU
- Host (PC with \geq Win95 and free COM port)
- Software FLASHPRG.EXE (MB91360, MB91460 Flash Programmer)
- RS232 Serial cable (“extension type” 1:1)
- Power supply (9-12V DC ; + on shield) for Starterkit

1 Preparations

1.1 Software installation

Extract the provided “FME_FR_Flashprogrammer_v4_xx.zip” to to C:\Flashprg (or any other directory). The following files will be installed:

FLASHPRG.EXE : Stand-alone Flash-programmer software (details see below)
BRLxxxxxx.BIN : Binary files for programming specific devices
README.TXT : Usage and version Information
FLASHPRG.PDF : Flashprogrammer documentation (this file)
FLASHBAT.BAT : Example batch-file for automated flash programming
1.2 Hardware Setup:

Set jumper settings of your starter kit to asynchronous flash programming (please refer to the User Guide of your starter kit for further information).

Connect a PC-COM-port via a RS232-cable to the UART which is used for flash programming (see User Guide) of your board.

Before the Flashprogrammer can connect to the device a reset is necessary. If you don’t want to reset your device manually see your User Guide for DTR-reset jumper settings.

If you use your own hardware, make sure you have external access to the UART. In addition, you should be able to initiate a power-on reset.
2 Using the Software

2.1 Getting started

Run the installed Software-tool “FME-FR Programmer”. The following main screen will appear:

Enter the correct COM-port number (1-x) and your device from the selections menu. Then press the connect button and - shortly afterwards - switch on the Starterkit (or hit “Reset”). See 2.7 Sub-Menu “Signals” how to make an automatic reset.

✓ If a connection could be established to the internal Boot-RO, the message “Connection OK” and the Chip ID will be shown in the log-window.

✗ If you fail to get a connection, the message “there was no reply” will appear in the log-window after a certain amount of time. Double-check the COM-port number on your PC and the hardware-settings. Be sure the starterkit is ready and you are using the correct RS232-cable.
To increase the time until “no reply”, change the value in “no. of attempts”. This value is the total number of attempts to receive the acknowledge-character from the device after sending a start-code.

### 2.2 Using the “Automatic mode”

A very convenient way of programming your application to flash-ROM is using the automatic mode. All you have to do is select the appropriate file to program and press the “Automatic”-button. The entire programming procedure should work on its own.

Specify the file to be programmed to flash-ROM in the “file to program”-field (use the browse-button to select). This file must be a converted linker output file from Softune in the Motorola™-Format “*.MHX”. (See Appendix for more information).

![Automatic Mode](image)

Then press the automatic button and switch on the starterkit (or press Monitor Reset). The following sequence will be processed:

- **Connect**: Establishes a serial connection to the device
- **Dump Flashloader**: The loader will be written to RAM using the Boot-ROM protocol
- **Start Flashloader**: The baudrate will be set and the flashloader will be executed
- **Initialize Flashmode**: The controller will be initialized for flash-programming
- **Blank Check**: The flash-ROM is checked for programmed words
- **Erase flash, if necessary**: If the flash-ROM is not blank, it will be erased now
- **Program flash and verify**: The file will be programmed to flash-ROM and verified
- **Call application at address**: As an option, the application will be started automatically

- ✓ If the automatic sequence was successfully terminated, the message “Automode finished” will be shown in the log-window.
- ✗ If any of the above operations will fail, the sequence stops with an error-message and has to be started again. If errors occur always during flash-operations, try to reduce the baudrate before starting again.

Note that some operations may take longer – esp. erasing the flash. This depends on the number of previous write/erase-cycles.

If you wish to adjust the programming baudrate (only for the flash-programming itself, not for any operations using the Boot-ROM protocol!), use the “programming baudrate” selection. Supported baudrates are 9600…115200 Baud. High-speed baudrates such as 115200 require low-noise transmission-lines and must be supported by your PC.
2.3 **Sub-Menu “External Flash”**

The external Flash functions are only supported by the MB91460 series MCU’s and this sub-menu is only visible if one of this MCU’s is selected as Device Type.

![External Flash Options](image)

**Fig. 2. : External Flash Options**

This new feature potentiates the programming of memory devices connect to the external bus of the MCU. If this feature is selected, all memory addresses within your MHX project file, that fit into the configured external memory area, are programmed into the external Flash!

The following configuration settings are available:

- **Enable ext. Flash:** Enables the support of the Flashloader to program external memory.

  - **CSx:** Selects the hardware connection of the external Flash device. Please select the corresponding Chip Select to which you Flash memory is connected.

  - **Chip Erase:** If this checkbox is enabled, the chip erase command also executes a chip erase on the external Flash memory.

  - **ASRx:** Area Select Register -> select the base memory mapping address of the external memory. For more details please refer to the MB91460 series Hardware Manual.
**ACRx:** Area Configuration Register -> Configuration of the selected memory I/F. For more details please refer to the MB91460 series Hardware Manual.

**AWCRx:** Area WaitConfiguration Register -> Wait setting configuration of the selected memory I/F. For more details please refer to the MB91460 series Hardware Manual.

### 2.4 Sub-Menu “Boot-ROM Functions”

All available functions in sub-menus are individual operations which will be used by “automatic mode” in a defined sequence. Use these functions if customized operation is required or to find errors. Background knowledge of the Boot-ROM and flash operation is highly recommended in order to prevent misoperation, damage or permanent loss of program code.

The following functions represent the basic operations of the built-in Boot-ROM:

- **Write File**: Dumps a binary file into the internal RAM (FastRAM) beginning at the specified address. The transmission will be verified by a checksum.

- **Write Byte**: Writes one single byte to the given location (RAM)

- **Call**: Calls the program or function at the given address. If there is a “RET”-statement at the end, the operation will continue (return value is displayed).

- **Read**: Reads out a number of bytes from a given address and creates a binary file.

- **Checksum**: Displays the checksum from the last operation.

**Note**: Further details about the Boot-ROM functions and the protocol are included in the hardware manual of the device.
2.5 Sub-Menu “Flash Functions”

The flash functions work together with the various flashloaders (the “brloadxx.bin”-files), that have to be downloaded to FastRAM first. Without any of the provided flashloaders operating from FastRAM, these functions cannot be used.

Initialize: Checks whether the flashloader which was dumped to FastRAM is available, initialized at the correct baudrate and ready to operate. This function checks for a prompt character (“>”) and displays the flashloader version. If this operation fails (e.g. no flashloader present), none of the following functions will be enabled.

Blank Check: Reads out every byte of each flash-sector to check for non-blank (≠ “FF”) cells. The result is displayed in the flash-diagram shown (red = not blank sectors; green = blank sectors). If any of the sectors is not blank, the message “Flash NOT blank” will appear.  
Note: The flash configuration depends on the device and total flash-size. This operation is performed automatically after each programming/erasing operation. All not blank sectors are also listed in the Sector Erase-field.

Sector Erase: Erases the sectors indicated by the numbers in the Sector Erase field (separated by “,” eg: “1,2,3” will erase sectors 1,2,3). The result is displayed in the log-window.

ChipErase: Erases the entire flash-ROM (all sectors). The result is displayed in the log-window.

ProgWord: Programs one word to flash at the specified address. The result is displayed in the log-window. Note that “word” can mean different sizes (e.g. MB91F361 can only program 16-bit words). If the external memory feature is selected, programming of external memory addresses is possible too.

ProgFile: Handles the transfer of the specified MHX-file to the flashloader. Each line will be processed (programmed) sequentially. If any error occurs, the function will be aborted with an error-message (e.g. “Loading Error”). During programming, a progress-bar is displayed.

BlockTransfer: This option enables the block transfer of the MHX-file (instead of transmitting only single lines they are compressed to bigger chunks). This option will increase the programming speed and also enables the option to use an offset for the target address.

Read: Reads out a number of bytes beginning at the specified address and displays the results in hexadecimal format on the log-window. This function can be used to verify certain areas (e.g. security-vector).
If the external memory feature is selected, reading of external memory addresses is possible too.

**Call**: This function will divert operation of the device to the specified location. This terminates the flashloader.

**Read to File**: This function will read and save the complete flash memory to a binary file. This function can be used to verify the flash.
2.6 Sub-Menu “Options”

This page contains other settings:

**Create Trace-File**: If this box is checked, a file “flashprg.trc” will be created which contains all in- and out-going characters from this application. This file can be used to check for errors (knowledge of the protocol is required). Note: this file will be available after FLASHPRG has terminated.

**Working Directory**: Whenever this program is started, the actual location of the FLASHPRG.EXE will be treated as “Working Directory” where all necessary files (e.g. brloadxx.bin) must be placed. You may change this setting manually if binary files are placed in a different directory.

**User Directory**: The user directory is the location for the files to be programmed (mhx-files). This directory can be chosen independently from the working directory and can also be specified as command parameter (instead of programming file).
2.7 Sub-Menu “Signals”

This page contains all settings for the RS232-Signals:

**Online:** Defines if either DTR or RTS should be active when a device is connected. It is also selectable if the signal should be high or low active.

**Reset:** Defines if DTR or RTS is used to trigger a reset on the target board during startup (normally a DTR high-level is used).

**Reset for:** This option defines how long the reset signal should be active (values below 100ms are inaccurate).

**Reset Delay** This option defines how long communication is delayed after RESET during connect. The accuracy of the internal timer is about 15ms!

If you want to reset your device automatically before each connection attempt you'll have to jumper your board for DTR-reset and select DTR high-level as reset-signal. If you are using an USB-to-serial converter you might have to increase the time the reset-signal is generated.
3 Appendix

3.1 Information on Files to be programmed (MHX-Format)

The hex-file to be programmed to flash-ROM must be specified in the “file to program”-field (use the browse-button to select). This file must be a converted linker output file from Softune Workbench in the Motorola™ -Format (.MHX). If you can’t find this file in your project directory, you probably haven’t included the converter in your tool-chain. To do so, select “Project – Setup – Make/Build” from the Softune main menu and tick “Absolute module converter is started”. Be sure you have selected “Motorola S-Format” in your Converter settings (tool options). After a successful “Make” or “Build”, you should be able to find the “<projectname>.MHX”-file in the ABS-directory.

3.2 Required Linker-Settings

When linking your project, you should make sure that code and constant areas are located inside the flash area (MB91F361 = 180000\text{hex}..1FFFFF\text{hex}). Also make sure that your application starts at 1F4000\text{hex} !

3.3 Using “K-line”

This software as well as the Boot-ROM are capable of using a “K-line”-type connection. This is a special 1-wire communication standard which multiplexes serial input and output to one line. K-line is mainly used in automotive applications for diagnostic purposes and therefore ideal for re-programming a microcontroller inside a control unit.

If you are using a “K-line”-connection in your application you can use the Flash Programmer as described in this manual without any further actions. The software will automatically detect the presence of a K-line connection because of the transmission echos. If such a connection was detected, a message “K-line detected !” is displayed in the log-window and on the status bar.

Note that some K-line adapters use transistors instead of integrated circuits for multiplexing the signals. This can lead to a limited maximum baudrate. If you experience problems while flash-programming using a K-line connection, try to reduce the baud-rate.
### 3.4 Command line options

The Flash Programmer can process various command line parameters during start. This method allows to set certain parameters and start the automode:


- **(1st) com:** COM-Port : 1-x
- **-Dx** device selection :
  - D0 = MCM
  - D1 = MB91F361
  - D2 = MB91F365
  - D3 = MB91F362 (default)
  - D4 = MB91F365,6,7,8,9Gx
  - D5, D6, D7 = ext. Flash (Cremson Starterkit)
  - D8 = MB91F364
  - D9 = MB91F376G
  - D10 = MB91302
  - D11 = Flash16MB_CS2456
  - D12 = ADA_91V460_91F467D
  - D13 = MB91F467D
  - D14 = MB91F464A
  - D15 = MB91F465K
  - D16 = MB91F469G
  - D17 = EMA91V460
  - D18 = MB91F463N
  - D19 = MB91F467B
  - D20 = MB91F465X
  - D21 = MB91F467R
  - D22 = MB91F467C
  - D23 = MB91F467M

- **-C(address)** call-address after writing the file

- **-Q** quit FLASHPRG after successful programming

- **-B(Baudrate)** selects programming baudrate
  - (9600,19200,38400,57600,115200)

- **-S5** ignore S5 record

- **-E(errorlogfile)** creates an error-log file for batch-processing (see below)

  **Errorcodes :**
  - 0 = success
  - 1 = wrong command shell parameters
  - 2 = flash erase error
  - 3 = download error
  - 4 = user abort
  - 5 = file open error
  - 6 = no connect
  - 7 = command error
  - 8 = wrong echo (protocol) received

- **-ONLINE=[NONE/RTS/DTR]** select signal for online state

- **-RESET=[NONE/RTS/DTR]** select signal for reset insertion

- **-ICE** chip erase will not be executed in automatic session
-ATP=n (n)umber of attempts
-PSV security verctors are programmed (MB91F46xx)
-XEN enable external flash feature
-XCS(chipselect) enable chip select CHIPSELECT for ext. flash

ChipSelect:
1 = CS1
2 = CS2
3 = CS3
4 = CS4
5 = CS5
6 = CS6
-XCER execute chip erase on external flash

-XAWR=[WaitRegister] wait register setting for external bus chip select area
WaitRegister: 3378 (default)
-XASR=[AreaSelect] base memory address for chip select
AreaSelect: 0100
-XACR=[Configuration] configuration for external bus chip select area chip select
Configuration: 8822 (default: MB91F467D)
-RT=[t] Reset time t (in ms): defines how long the reset signal is send.
-RD=[t] Delay time t (in ms) after RESET at ‘connect’.
-ASE=[sectors] Will erase the given sectors in automatic mode instead of a chip erase (only if they are not blank), eg: -ASE=0,5,6
-TB transfers mhx-file as block (if supported by the flashloader)
-TBDB creates a new mhx-file (same name as the old +’_n’) containing the compressed S-Record strings used by the block-transfer function.
-OFFSET=[offset] defines an offset for the flash-programming (only if using -TB)
-TRACE enables trace mode
-UCLK=[freq] selects the frequency of the used oscillator (default 4 [MHz])

file1[,file2,..] hexfiles (must include full path+name+.mhx)
or path sets path for user directory only (last) file:

Examples:
FLASHPRG 2 c:\flashprg\Demo360.mhx

Opens COM2 and automatically programs the "Demo360"-example to the connected device at default baudrate (38400).

FLASHPRG 1  -B115200  -Q  -Ec:\temp\error.log c:\flashprg\LEDdemo1.mhx

Opens COM1 and automatically programs the "LEDdemo1"-example to the connected device using 115200 baud.
FLASHPRG will quit (whether successful or not) and create the file "error.log" which indicated the result "code=.." as explained above.

See the provided examples FLASHBAT.BAT and FLASHBAT_NT.BAT as examples for flash programming using batch files. Use these batch files to create your own automated flash programming environment.

**Important**: Please note that Fujitsu does not allow to use the flashprogrammer as a tool for mass production! See the disclaimer on page 13 for more details.
Examples for use in Softune Workbench:

FLASHPRG 2 -D1 %D%A.mhx (for use in Softune Workbench)

Opens COM2 and automatically programs the actual Softune project (converted file) into the flashROM of an MB91F361 device.

FLASHPRG 1 -D0 -C1F4000 %D%A.mhx (for use in Softune Workbench)

Opens COM1 and automatically programs the actual Softune project into the flashROM of an MCM and executes from 1F4000.

FLASHPRG 2 -C1F4000 C:\Softune\Sample\Test\Abs\Test.Mhx

This would be an example for programming and starting a fixed mhx-file through COM2 (Device is MB91F361 by default).

NOTE:

Be sure to enable the option "Absolute Load Module Converter is started" in the "Project Setup"-Menu. This creates the MHX-file.

To register the flash programmer as a “tool” in Softune, use the “setup-tool” dialog:

![Setup Tool](image)

Fig. 3. : Registering the flash programmer in Softune Workbench
4 Note on external flash programming

The selection “external flash” is used for the “Cremson Starterkit” in order to program any mhx-file to the 1MB, 8MB or 16MB external flash devices connected to the MB91F36x. This mode can be used for any external flash connection which has similar characteristics:

1MB external flash (2 devices MBM29LV400T) must be connected to external bus on CS2 in memory area : 0030:0000 ... 003F:FFFF. See the “Cremson Starterkit” documentation for schematics and further details.
5 Disclaimer

Warranty and Disclaimer

To the maximum extent permitted by applicable law, Fujitsu Mikroelektronik GmbH restricts its warranties and its liability for all products delivered free of charge (eg. software include or header files, application examples, target boards, evaluation boards, engineering samples of IC’s etc.), its performance and any consequential damages, on the use of the Product in accordance with (i) the terms of the License Agreement and the Sale and Purchase Agreement under which agreements the Product has been delivered, (ii) the technical descriptions and (iii) all accompanying written materials. In addition, to the maximum extent permitted by applicable law, Fujitsu Mikroelektronik GmbH disclaims all warranties and liabilities for the performance of the Product and any consequential damages in cases of unauthorised decompiling and/or reverse engineering and/or disassembling. Note, all these products are intended and must only be used in an evaluation laboratory environment.

1. Fujitsu Mikroelektronik GmbH warrants that the Product will perform substantially in accordance with the accompanying written materials for a period of 90 days form the date of receipt by the customer. Concerning the hardware components of the Product, Fujitsu Mikroelektronik GmbH warrants that the Product will be free from defects in material and workmanship under use and service as specified in the accompanying written materials for a duration of 1 year from the date of receipt by the customer.

2. Should a Product turn out to be defect, Fujitsu Mikroelektronik GmbH’s entire liability and the customer’s exclusive remedy shall be, at Fujitsu Mikroelektronik GmbH’s sole discretion, either return of the purchase price and the license fee, or replacement of the Product or parts thereof, if the Product is returned to Fujitsu Mikroelektronik GmbH in original packing and without further defects resulting from the customer’s use or the transport. However, this warranty is excluded if the defect has resulted from an accident not attributable to Fujitsu Mikroelektronik GmbH, or abuse or misapplication attributable to the customer or any other third party not relating to Fujitsu Mikroelektronik GmbH.

3. To the maximum extent permitted by applicable law Fujitsu Mikroelektronik GmbH disclaims all other warranties, whether expressed or implied, in particular, but not limited to, warranties of merchantability and fitness for a particular purpose for which the Product is not designated.

4. To the maximum extent permitted by applicable law, Fujitsu Mikroelektronik GmbH’s and its suppliers’ liability is restricted to intention and gross negligence.

NO LIABILITY FOR CONSEQUENTIAL DAMAGES

To the maximum extent permitted by applicable law, in no event shall Fujitsu Mikroelektronik GmbH and its suppliers be liable for any damages whatsoever (including but without limitation, consequential and/or indirect damages for personal injury, assets of substantial value, loss of profits, interruption of business operation, loss of information, or any other monetary or pecuniary loss) arising from the use of the Product.

Should one of the above stipulations be or become invalid and/or unenforceable, the remaining stipulations shall stay in full effect.