Fujitsu Microelectronics Europe
Application Note

MCU-AN-3900031-25

F²MC-16LX FAMILY

16-BIT MICROCONTROLLER

ALL 16LX SERIES WITH FLASH

PROGRAMMING FLASH MCU'S

APPLICATION NOTE

FUJITSU

Revision History

Date	Issue
01.08.1999	New Format, new updated version, Tka
17.12.1999	3.5 Program Code Download, MDx mode pin setting description changed, Tka
01.03.2000	MB90F497 added, MST
16.03.2000	Updated for new Flash500 programming software, Tka
17.01.2001	Schematic for serial programming changed, recommendation added to connect HST directly to RST, Tka
17.04.2002	new series and Software Version included, MST
12.08.2002	Version. 2.0, New series added, typos corrected, MST
22.01.2004	Version 2.1 New series added: MB90340, MB90350, MB90455, MB90480, MB90860, MSt Programming Times added, HWe
03.02.2004	Version 2.2 New series added: MB90890, MSt
07.01.2005	Version 2.3 New series added: MB90330, MB90335,MB90800, MB90820, MB90945, Boot loader versions updated MSt
16.02.2005	Version 2.4 Typo in Flash Series description corrected: MB90F038S, MB90F34x Chapter 2.4 table corrected (MB90F39x, MB90F59x, MB90F94x)
11.04.2005	Version 2.5 MB90360 Series added, HWe

This document contains 37 pages.

Warranty and Disclaimer

To the maximum extent permitted by applicable law, Fujitsu Microelectronics Europe GmbH restricts its warranties and its liability for **all products delivered free of charge** (e.g. 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 Microelectronics Europe 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 Microelectronics Europe 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 Microelectronics Europe 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 Microelectronics Europe GmbH´s entire liability and the customer's exclusive remedy shall be, at Fujitsu Microelectronics Europe 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 Microelectronics Europe 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 Microelectronics Europe GmbH, or abuse or misapplication attributable to the customer or any other third party not relating to Fujitsu Microelectronics Europe GmbH.

3. To the maximum extent permitted by applicable law Fujitsu Microelectronics Europe 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 Microelectronics Europe 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 Microelectronics Europe 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

Contents

REVISION HISTORY **2**

WARRANTY AND DISCLAIMER **3**

CONTENTS **4**

0 INTRODUCTION **6**

1 FUJITSU FLASH MICROCONTROLLER OVERVIEW **7**

Main Flash Features 9

1.1 Evaluation boards 9

2 PROGRAMMING FLASH MICROCONTROLLERS **10**

2.1 Use an ordinary EPROM Programmer 10

2.2 Fujitsu embedded Burn-IN ROM serial programming mode 10

2.3 User application software (also called bootloader) 11

3 FUJITSU EMBEDDED BURN-IN ROM SERIAL PROGRAMMING **15**

3.1 Serial Interface of the Microcontroller 15

3.2 Serial interface of the PC 15

3.3 Mode Setting for the Microcontroller 15

3.4 Installing the PC Software 17

3.5 Program Code Download 18

3.6 QFP120 evaluation board Preparing 19

3.7 Preparing the Flash-CAN-100P evaluation board 19

4 WORKING WITH A USER APPLICATION SOFTWARE (BOOTLOADER) **22**

4.1 General Description 22

4.2 Different methods to reprogram the flash memory 23

4.3 Application Call 23

4.4 Special Note for MB90F553 V2.3 Bootloader 24

4.5 Bootloader Commands 24

4.5.1 Command overview 24

4.6 Modifying the Bootloader 25

4.7 Updating the Bootloader 26

5 COMMAND LINE OPTIONS **27**

5.1 HEXLOADW 27

5.2 SKWIZARD 27

5.3 BINHEX 28

6 BOOTLOADER SELECTOR GUIDE 29

7 TROUBLESHOOTING 32

7.1 Fujitsu FlashMCU Programmer 16LX 32

7.1.1 Error Messages 32

7.1.2 Using Laptops 34

7.1.3 Wrong file is programmed 34

7.1.3.1 Using version V01L03, V01L05 or V01L07 34

7.1.3.2 Using Flash version 3xx, 4xx, 5xx or 6xx with Win NT 34

7.1.4 Unable to program file 34

8 PROGRAMMING TIME 35

8.1 Serial asynchronous mode 35

8.1.1 Tool: Fujitsu MCU Flash Programmer (freeware) 35

8.1.2 Tool: FlashKit 35

8.1.3 Tool: GALEP-4 35

8.2 Serial synchronous mode 36

8.2.1 Tool: FlashKit 36

8.2.2 Tool: GALEP-4 36

8.2.3 Tool: Yokogawa AF221 36

8.3 Parallel mode 37

8.3.1 Tool: DATA-IO 37

8.3.2 Tool: ANDO 37

0 Introduction

Fujitsu offers a wide range of 16-bit Flash Microcontroller.

This Application Note explains the different possibilities to program the Flash MCU's.

Using parallel Programmer, in-circuit programming via serial asynchronous/synchronous connection, or via User-application (e.g. bootloader).

1 Fujitsu Flash Microcontroller Overview

Fujitsu offers at this time the following Flash Microcontroller of the 16LX Family

MB90F038S, 512KB Flash Memory, single CAN, 4x UART-LIN, 16 ch. ADC
MB90F334A, 384KB Flash Memory, USB with mini host function, 16 ch. ADC
MB90F337, 64KB Dual Operating Flash Memory, USB with mini host function
MB90F342A(S)/CA(S), 256KB Flash Memory, dual CAN, 4x UART-LIN, 16/24 ch. ADC
MB90F345A(S)/CA(S), 512KB Flash Memory, dual CAN, 4x UART-LIN, 16/24 ch. ADC
MB90F347A(S)/CA(S), 128KB Flash Memory, single CAN, 4x UART-LIN, 16/24 ch. ADC
MB90F349A(S)/CA(S), 256KB Flash Memory, single CAN, 4x UART-LIN, 16/24 ch. ADC
MB90F351/S, 64KB Flash Memory, single CAN, 2x UART-LIN, 15 ch. ADC
MB90F352/S, 128KB Flash Memory, single CAN, 2x UART-LIN, 15 ch. ADC
MB90F362(T)(S), 64KB Flash Memory, single CAN, 2x UART-LIN, 16ch. ADC
MB90F367(T)(S), 64KB Flash Memory, single CAN, 2x UART-LIN, 16ch. ADC, Clock-Supervisor
MB90F387/S, 64KB Flash Memory, single CAN, small package (48-pin)
MB90F394H, 384KB Flash Memory, dual CAN interface
MB90F423GA/GB/GC, 128KB Flash Memory, dual CAN, LCD
MB90F428GA/GB/GC, 128KB Flash Memory, single CAN, LCD
MB90F438L/LS, 128KB Flash Memory, external bus interface
MB90F439/S, 256KB Flash Memory, external bus interface
MB90F443G, 128KB Flash Memory, triple CAN, external bus interface
MB90F455/S, 24KB Flash Memory, small package (48-pin)
MB90F456/S, 32KB Flash Memory, small package (48-pin)
MB90F457/S, 64KB Flash Memory, small package (48-pin)
MB90F462, 64KB Flash Memory, Low Cost
MB90F474H/L, 256K Flash Memory, 3V device, external bus interface
MB90F481, 192K Flash Memory, 3V device, external bus interface
MB90F482, 256K Flash Memory, 3V device, external bus interface
MB90F497G, 64KB Flash Memory, single CAN, external bus interface
MB90F498G, 128KB Flash Memory, single CAN, external bus interface
MB90F523B, 128KB Flash Memory, LCD
MB90F543/G/GS, 128KB Flash Memory, double CAN, external bus interface
MB90F546G/GS, 256KB Flash Memory, single CAN, external bus interface
MB90F548G/GS, 128KB Flash Memory, single CAN, external bus interface
MB90F549, 256KB Flash Memory, single CAN, external bus interface
MB90F553A, 128KB Flash Memory, external bus interface
MB90F562/B, 64KB Flash Memory, Low Cost
MB90F568, 128KB Flash Memory, Low Cost, 3V device

MB90F574/A, 256KB Flash Memory, Large Memory, external bus interface

MB90F583B, 128KB Flash Memory, 5 UART's, external bus interface

MB90F591G, 384KB Flash Memory, dual CAN interface

MB90F594G, 256KB Flash Memory, dual CAN interface

MB90F598/G, 128KB Flash Memory, single CAN interface

MB90F804-101/-201, 256K Flash Memory, 4x48 LCD Interface, 12 ch. ADC, 3V device

MB90F822, 64KB Flash Memory, Multi Function timer for 3-phase PWM, 16 ch. ADC

MB90F823, 128KB Flash Memory, Multi Function timer for 3-phase PWM, 16 ch. ADC

MB90F867/S, 128KB Flash Memory, UART-LIN, 24 ch. ADC, I²C, external Bus-interface

MB90F897/S, 64KB Dual Operation Flash Memory, single CAN, small package (48-pin)

MB90F947, 128KB Flash Memory, single CAN, 15 ch. ADC

MB90F949, 256KB Flash Memory, single CAN, 15 ch. ADC

Main Flash Features

- Single 5V power supply
- 10000 erase cycles
- 10 years data retention
- Different sector sizes available
- Sectors can erased individually
- In system programmability

1.1 Evaluation boards

For the ***MB90F523B and MB90F574*** series the **FLASH-EVA2-120P-M13** evaluation board is available as a low cost multifunctional evaluation board. This Flash-Test-Board can be used as a simple target board for the emulator and as an evaluation board to download and program the software via RS232 connection and test the user application. This board is shipped with an MB90F523, programmed with a boot-loader to download software via RS232. The board does not offer a special ROM monitor debugger.

To download the user application the SKWizard, which is a terminal program with integrated software download function, or the HexloadW utility program from Fujitsu can be used.

Note: For Windows NT it could happen that the current version of the SKWizard does not work correctly.

For the ***MB90F428GA/GB/GC, MB90F438L/LS, MB90F439/S MB90F443G, MB90F474H/L, MB90F481, MB90F482, MB90F553A, MB90F543/G/GS, MB90F546G/GS, MB90F548G/GS, MB90F549, MB90F583, MB90F591G, MB90F594/A/G, and the MB90F598/G*** the ***Flash-CAN-100P-M06*** evaluation board can be used. In the main it is used as an emulator target board. But it is also possible to use this board as a quite simple Flash evaluation board to test some functions of the user application.

For the ***MB90F462, MB90F497G, MB90F498G, MB90F562/B, MB90F568*** the ***Flash-CAN-64P-M09-V2*** evaluation board can be used.

For the ***MB90F394H***, ***MB90F395H*** the ***Flash-CAN-120P-390*** evaluation board can be used.

For the ***MB90F038S, MB90F342A(S)/CA(S), MB90F345A(S)/CA(S), MB90F347A(S)/CA(S), MB90F349A(S)/CA(S), MB90F804-101/201, MB90F867/S, MB90F947, MB90F949*** the ***Flash-CAN-100P-340*** evaluation board can be used.

For the ***MB90F387/S, MB90F455/S, MB90F456/S, MB90F457/S, MB90F897/S*** and ***MB90F36x*** the ***Flash-CAN-48P-M26*** evaluation board can be used.

For the ***MB90F351/S, MB90F352/S*** the ***Flash-CAN-64P-350*** evaluation board can be used.

For the ***MB90F822, MB90F823*** the ***SK-90820-80PFM-562*** evaluation board can be used.

2 Programming Flash Microcontrollers

To program Flash microcontrollers there exist three possibilities:

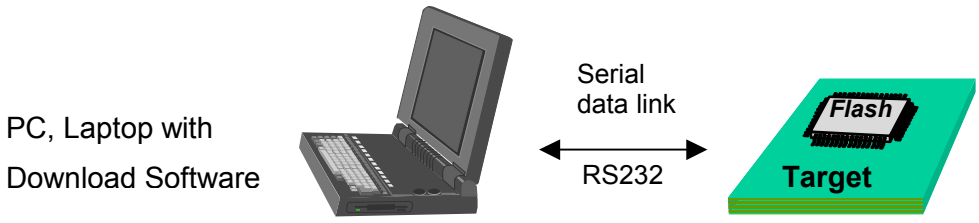
2.1 Use an ordinary EPROM Programmer

- **Minato 1890A,** with Base OU910 unit, with programming adapter
- **Data I/O, SMS Sprint Programmer,** with programming adapter
- **Conitec Datensysteme (GalepIII, IV),** with programming adapter
- **BP Microsystems,** with programming adapter
- **Stag,** with programming adapter
- **RK System,** with programming adapter

2.2 Fujitsu embedded Burn-IN ROM serial programming mode

In this mode a standard asynchronous RS232 connection to a PC can be used to download the software and program the Flash Microcontroller (even blank devices) directly in the system. Therefore an IN-ROM bootstrap loader is used which is enabled by a special mode pin setting of the microcontroller. On the PC side a special Software is used. On the target system a RS232 interface must be assigned for the corresponding UART of the microcontroller. Additionally the Mode pins and two port pins must be used to set-up the microcontroller into this programming mode after Reset or power on. The download rate is about 9600Bit/s if a 4MHz crystal is used for the Microcontroller.

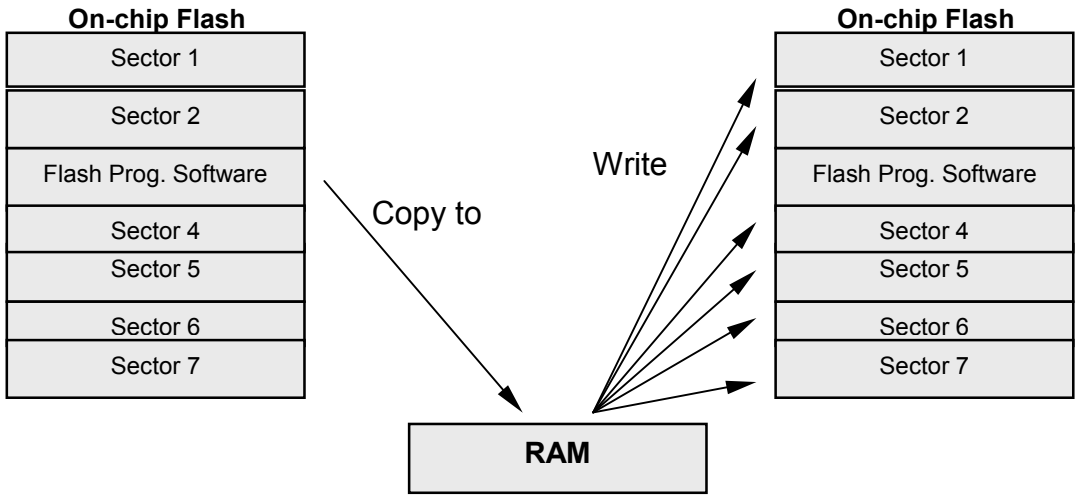
Note: Be aware that not all Flash microcontroller support this special serial-programming mode, therefore please see the table below.



2.3 User application software (also called bootloader)

With this method a user application software is started to download the program data and program the Flash memory. Therefore such a user application (bootloader) must be located in the Flash memory (programmed into the Flash via method 1) or 2)). This bootloader could be located in a small sector and is used by a special command to download the data (e.g. via RS232 or CAN interface) and to program the data into the Flash memory. During Flash programming the bootloader programming software itself must be running in the RAM area.

This method is available as an example application on the QFP120 Flash-Test-Board using the RS232 interface for data download. Furthermore bootloaders for the MB90F523, MB90F543, MB90F574, MB90F583, MB90F598, MB90F553A and MB90F594/A are available. Of course it is possible to modify the bootloaders corresponding to the needs of the application.



The user Flash programming software is copied to the internal RAM. After that, the program is started in the RAM and the Flash programming is executed.

2.4 Overview

The following table gives an overview about the Flash Microcontrollers and the Flash programming possibilities for these devices.

Device	Data I/O *1	Minato 1830 + OU910	Conitec Galepp III	Burn-In ROM Serial Programming Mode (UART used by µC)	Program example for a User Bootloader available (current Version)
MB90F038SPF	--	--	--	UART0	Same as for MB90F347
MB90F334A	--	--	--	UART0	--
MB90F337	--	--	--	UART0	--
MB90F342PF MB90F342SPF MB90F342CPF MB90F342CSPF	--	--	210871	UART0	Same as for MB90F347
MB90F345PF MB90F345SPF MB90F345CPF MB90F345CSPF	--	--	210871	UART0	Same as for MB90F347
MB90F347PF MB90F347SPF MB90F347CPF MB90F347CSPF	PAA-FP-Q183 QFP100	--	210871	UART0	V1.0
MB90F349PF MB90F349SPF MB90F349CPF MB90F349CSPF	PAA-FP-Q183 QFP100	--	210871	UART0	Same as for MB90F347
MB90F351PF MB90F351SPF	--	--	210897	UART3	Same as for MB90F352
MB90F352PF MB90F352SPF	--	--	210897	UART3	V1.0
MB90F36x	--	--	--	UART1	No (compact Flash)
MB90F387PMT MB90F387SPMT	--	--	210880	UART	--
MB90F394HPMT	PS-S5173H QFP120	--	210854	UART0 (async) SIO4 (sync)	--
MB90F395HPMT	--	--	--	UART0 (async) SIO4 (sync)	--
MB90F423GAPF MB90F423GBPF MB90F423GCPF	S5023	--	--	UART1	---

Device	Data I/O *1	Minato 1830 + OU910	Conitec Galepp III	Burn-In ROM Serial Programming Mode (UART used by µC)	Program example for a User Bootloader available (current Version)
MB90F428GAPF MB90F428GBPF MB90F428GCPF	S5023	MF00-989	--	UART1	---
MB90F438LPF MB90F438LSPF	S5023	MF00-796	210871	UART1	Same as for MB90F543
MB90F439PF MB90F439SPF	S5023	MF00-796	210871	UART1	Same as for MB90F543
MB90F443GPF	S5023	MF00-989	--	UART1	Same as for MB90F543
MB90F455PMT MB90F455SPMT	--	--	210880	UART	--
MB90F456PMT MB90F456SPMT	--	--	210880	UART	--
MB90F457PMT MB90F457SPMT	--	--	210880	UART	--
MB90F462PFM	--	--	--	UART0	--
MB90F474LPF	--	MF00-989	--	UART0	--
MB90F474HPF	--	MF00-989	--	UART0	--
MB90F481PF	--	--	--	UART0	--
MB90F482PF	--	--	--	UART0	--
MB90F497GPFM	--	MF13-786	210873	UART1	V1.4
MB90F523PFV	S5024	MF00-23	210872	Not supported	V3.0
MB90F523BPFV	S5024	MF00-23	--	UART0	V3.0
MB90F543GPF MB90F543GSPF	S5023	MF00-989	210871	UART1	V1.1
MB90F546GPF MB90F546GSPF	S5023	MF00-989	210871	UART1	Same as for MB90F543
MB90F548GPF MB90F548GSPF	S5023	MF00-989	210871	UART1	Same as for MB90F543
MB90F549PF	S5023	MF00-989	210871	UART1	Same as for MB90F543
MB90F553APF	S5023	MF00-989	210871	UART0	V4.0
MB90F562/B	--	MF13-786	210873	UART1 (P60/61)	--
MB90F568	--	MF13-786+ML01-781	210873	UART1 (P60/61)	--
MB90F574APFV	S5024	MF00-729	210872	UART0	V1.1
MB90F583BPF	S5023	MF00-989	210871	UART0	V1.0
MB90F591GPF	S5023	MF00-796	210871	UART0 (async) SIO3 (sync)	Same as for MB90F594

Device	Data I/O *1	Minato 1830 + OU910	Conitec Galepp III	Burn-In ROM Serial Programming Mode (UART used by µC)	Program example for a User Bootloader available (current Version)
MB90F594/GPF	S5023	MF00-989	210871	Not supported	V2.0
MB90F594A/GPF	S5023	MF00-989	210871	UART0 (async) SIO3 (sync)	V2.0
MB90F598/GPF	S5023	MF00-989	210871	UART1	Same as for MB90F594
MB90F804-101PF MB90F804-201PF	--	--	--	UART0	--
MB90F822PFM	--	--	--	UART0	--
MB90F823 PFM	--	--	--	UART0	--
MB90F867PF MB90F867SPF	--	--	210871	UART1	Same as for MB90F347
MB90F897PMT MB90F897SPMT	--	--	210880	UART1	--
MB90F947PF	--	--	210877	UART0 (async) SIO4 (sync)	--
MB90F949PF	--	--	210877	UART0 (async) SIO4 (sync)	--

Table 1: Overview of Flash MCU's and available Programming adapter

***1: Note:**

There are several different Data I/O Programmer available.

Check if adapter supports your Data I/O programmer.

3 Fujitsu embedded Burn-In ROM serial programming

Fujitsu Flash microcontrollers offer a quite simple way to program the Flash memory directly in the system, without the need of a special programmer. The Serial Asynchronous Programming Mode allows the user to program the microcontroller directly on the target system via a standard RS232 connection. So Software updates can be done very easily.

All you need to program the Flash Memory inside the microcontroller is a PC and a RS232 cable. For the microcontroller some additional circuits have to be assigned on the target board. The Block diagram (Figure1) shows all necessary connections, which has to be done to program the Flash memory.

3.1 Serial Interface of the Microcontroller

To connect the microcontroller to the serial interface only the Transmit (SOT) and the Receive line (SIN) of the UART are necessary. For voltage level conversion a RS232 driver has to be used (e.g. MAX232). The RS232 driver can be assigned directly on the target board or it can be mounted on an adapter, which is connected to the target to program the device. This solution saves some space on the target and minimises the costs.

Note:
The MB90F334A, MB90F337, MB90F34x(C)AS, MB90F39x, MB90F474H/L, MB90F481, MB90F482, MB90F462, MB90F553A, MB90F574, MB90F583, MB90F591G, MB90F594A, MB9082x, MB90F867/S, MB90F94x use the UART0 interface.
The MB90F36x, MB90F428GA/GB/GC, MB90F443G, MB90F497, MB90F562/B, MB90F543/G/GS, MB90F546G/GS, MB90F548G/GS, MB90F549, MB90F598, MB90F804-101/-201 and MB90F897/S use the UART1 interface.
The MB90F35x/S series use the UART3 interface.
The MB90F387/S, MB90F455/S, MB90F456/S, MB90F457/S series includes one UART interface. This interface must be used for Flash Programming.

At this time only these devices can be programmed using the Fujitsu Burn-In ROM Serial Asynchronous Programming Mode.

3.2 Serial interface of the PC

To establish a connection to the microcontroller the RTS (Request To Send) and the CTS (Clear To Send) lines of the PC interface must be connected together. Also the DTR (Data Terminal Ready) must be connected to the DSR (Data Set Ready) signal. Otherwise the program cannot send any data to the microcontroller and a "Communication Error" occurs. The Transmit line (TXD) of the PC interface has to be connected to SIN, the Receive line (RXD) has to be connected to SOT via the RS232 driver. The GND line of the RS232 interface must also be connected.

3.3 Mode Setting for the Microcontroller

To program the Flash memory of the microcontroller it is necessary to change the operating mode of the controller to the Asynchronous Serial Programming Mode. Therefore the mode pins MD0, MD1, MD2 and the port pins P00 and P01 are used. Check below tables for different port pins usage of some series.

Note: MB90385, MB90455, MB90890 series is using P30 and P31 instead of P00 and P02. MB90390 can be programmed also with 5MHz/10MHZ/20MHz external clock.

Table1/2/3/4 show the required settings.

Pin	MD2	MD1	MD0
Level	HIGH	HIGH	LOW

Table 2: Mode pin settings

External clock	(4/8/16) MHz		RS232Mode
Pin	P01	P00	
Level	LOW	LOW	Asynchronous
Level	HIGH	LOW	Synchronous

Table 3: Port settings for 16LX series, exceptions see below tables

External clock	(4/8/16) MHz		(5/10/20) MHz		RS232Mode
Pin	P01	P00	P01	P00	
Level	LOW	LOW	LOW	HIGH	Asynchronous
Level	HIGH	LOW	HIGH	LOW	Synchronous

Table 4: Port settings for MB90390 series

External clock	(4/8/16) MHz		RS232Mode
Pin	P31	P30	
Level	LOW	LOW	Asynchronous
Level	HIGH	LOW	Synchronous

Table 5: Port settings for MB90385, MB90455, MB90890 series

External clock	(4/8/16) MHz		(5/10/20) MHz		RS232Mode
Pin	P81	P80	P81	P80	
Level	LOW	LOW	LOW	HIGH	Asynchronous
Level	HIGH	LOW	HIGH	LOW	Synchronous

Table 6: Port settings for MB90470 series

External clock	(4/8/16) MHz		(3/6/12/24) MHz		RS232Mode
Pin	P81	P80	P81	P80	
Level	LOW	LOW	LOW	HIGH	Asynchronous
Level	HIGH	LOW	HIGH	LOW	Synchronous

Table 7 Port settings for MB90F481 series

External clock	(3/6/12/24) MHz		(5/10/20) MHz		RS232Mode
Pin	P81	P80	P81	P80	
Level	LOW	LOW	LOW	HIGH	Asynchronous
Level	HIGH	LOW	HIGH	LOW	Synchronous

Table 8: Port settings for MB90F482 series

External clock	4 MHz		6 MHz		RS232Mode
Pin	P66	P65	P66	P65	
Level	LOW	LOW	LOW	HIGH	Asynchronous
Level	HIGH	LOW	HIGH	LOW	Synchronous

Table 9: Port settings for MB90F804 series

External clock	6 MHz		RS232Mode
Pin	P61	P60	
Level	LOW	LOW	Asynchronous
Level	HIGH	LOW	Synchronous

Table 10: Port settings for MB90F334A, MB90F337 series

External clock	(4/8) MHz		RS232Mode
Pin	P84	P83	
Level	LOW	LOW	Asynchronous
Level	HIGH	LOW	Synchronous

Table 11: Port settings for MB90360 series

All these settings are also available in the flash Programmer for 16LX help file.

With these settings the microcontroller enters the Asynchronous/synchronous Serial Programming Mode after power-on or after generating a Reset using /RST.
It is recommended to connect /HST directly to /RST. It might be useful to connect the reset line /RST to a button to generate a Reset. Otherwise Power-on must be used to change the operating mode.
Note: Using only the /HST instead of /RST to generate a microcontroller Reset will not set the microcontroller into the Asynchronous Serial Programming Mode. The usage of /RST is mandatory to change the operating mode! Nevertheless the /HST has its own functionality which must be considered in the design.

3.4 Installing the PC Software

To install the PC Software just run the self-extracting file FlashV01L03.. This will unzip the corresponding Flash programming software by default in c:\Softune\Utility\FlashV01L03. But any other directory can also be used. Afterwards start the install.exe to install the FlashMCU Programmer 16LX. (Please, always check the Internet page for available updates).
For troubleshooting refer to chapter 7.

3.5 Program Code Download

To download data to the microcontroller, the Flash programming Software must be started. After that, first the corresponding device must be selected from the device list and the clock frequency must be set corresponding to the external clock frequency used for the controller. Only the following external clock frequencies are allowed: 4, 8, and 16MHz. Other frequencies will not work because the baud rate is fixed in the Burn-In ROM. After that select the Com port which is used for the download.
Now the <Download> command can be executed. Now, first the Flash programming routines itself will be downloaded into the RAM of the microcontroller. A "Download ok" message confirms that the connection to the controller could be established and the software is now ready to start. After the download, the baud rate is increased to download the application at a higher transfer speed. The following table gives an overview.

External Clock Frequency	Baudrate used to download Flash Programming routines	Baudrate used to download the application itself (used with erase -, write & verify -, read -, blank check -, auto command)
4MHz	4800	9600
8MHz	9600	19200
16MHz	19200	38400

Table 12: Possible download baudrates

For the next step the file, which has to be downloaded, must be selected. This is done via the <search> command. The file itself must be a Motorola S2-Record and the S-Record must contain a start record (S0) and an end record (S8). Otherwise a temporary generated binary file (_w_o_r_k.5, which is used for the download) cannot be generated and the error message "unable to open _w_o_r_k.5" occurs. The S0 and S8 record is only used to recognise the start and the end of the data section during the transfer.
Note: Check the Motorola S-Rec for S0 and S8 Record!
With the <Auto> command a complete programming sequence is started: Download, Erase complete Flash memory, Blank check, write file to Flash memory and verify data.
Of course single commands like <Erase>, <Blank Check>, <Write&Verify>, <Read and Compare> and <copy> can be executed as well.
After the programming sequence the application can be started. Therefore the operating mode of the microcontroller must be changed to the corresponding setting which is necessary for the application (e.g. single chip mode: MD2=0, MD1=1, MD0=1). After that, a Power-on Reset or pressing the Reset button will start the application in the Flash memory.

3.6 QFP120 evaluation board Preparing

If the QFP120 Flash evaluation board is used to program the MB90F574 device make sure that the Reset LED is off after power-on. Otherwise change the DTR polarity with jumper J20 or just remove the jumper to avoid any influence of the DTR line during the programming sequence. To generate a correct Reset, the jumper J21 must be set to /RST. To enable the Serial Programming Mode, the Mode pins must be set correctly. Therefore the switch S1 must be set for MD2, MD1 and MD0 to position ON. The connection of the port pins P01 and P00 to GND must be done with two additional wires. The UART0 of the MB90F574 is connected to the DB-9 connector by plugging the RN4 resistor network in the position MB90570 and setting the jumpers J12, J13, J14 to Burn-In position. The RTS / CTS and the DTR / DSR signals can be wired together on the evaluation board as well (RTS pin 7, CTS pin 8 of the DB-9 connector). The serial cable must be a straight 1:1 connection.
After the download the Mode pin setting must be changed to the corresponding setting of the application (e.g. single chip mode: MD2 Off, MD1 ON, MD0 Off – 0 1 1). A Power-on Reset or pressing the Reset button will start the application in the Flash memory.

3.7 Preparing the Flash-CAN-100P evaluation board

To program a microcontroller on the Flash-CAN evaluation board it is necessary to connect the RS 232 interface to the PC. Take care that the correct serial channel of the microcontroller is connected to the RS232 interface connector. Therefore the jumpers JP7, JP8, JP9, JP10 must be used. The following Table shows the settings for the corresponding devices (Flash-CAN board Rev. 1.2 or later):

Device	JP7	JP8	JP9	JP10
MB90F428GA/GB/GC	JP7 pin 2 – Pin 89	JP 8 pin 2 – Pin 88		
MB90F438L/LS	--	--	1-2 (P24)	1-2 (P21)
MB90F439/S	--	--	1-2 (P24)	1-2 (P21)
MB90F443G	--	--	1-2 (P24)	1-2 (P21)
MB90F474H/L	JP7 pin 2 – Pin 28	JP 8 pin 2 – Pin 27		
MB90F481	JP7 pin 2 – Pin 28	JP 8 pin 2 – Pin 27		
MB90F482	JP7 pin 2 – Pin 28	JP 8 pin 2 – Pin 27		
MB90F543/G/GS	--	--	1-2 (P24)	1-2 (P21)
MB90F546G/GS	--	--	1-2 (P24)	1-2 (P21)
MB90F548G/GS	--	--	1-2 (P24)	1-2 (P21)
MB90F549	--	--	1-2 (P24)	1-2 (P21)
MB90F553A	1-2 (P19)	1-2 (P20)	--	--
MB90F583B	1-2 (P19)	--	JP9 P18 - JP10 pin2	--
MB90F591G	2-3 (P14)	2-3 (P16)	--	--
MB90F594A	2-3 (P14)	2-3 (P16)	--	--
MB90F598	--	--	1-2 (P24)	1-2 (P21)

Table 13: FLASH-CAN-100P-M06, UART Jumper settings

For the serial PC interface it is also necessary to connect RTS with CTS. DTR is already connected to DSR on the board.

After that the Mode pins and the port pins P00 and P01 must be switched to the right position.

To set the mode pins and the port pins for serial async. programming the DIL switch S3 must be set as follows:

SW1 ON, SW2 OFF, SW3 OFF, SW4 OFF, SW5 ON, SW6 OFF, SW 7 ON, SW8 ON. SW1 to 3 are responsible for the Mode pins, SW5 to connect RTS to CTS, SW7 and SW8 set the port pins P00 and P01to GND. Additionally jumper JP12 must be removed to avoid that the DTR signal resets the board.

After these settings the Flash programming software can be started. After programming the application can be started. Therefore the switches SW7 and SW8 should be set to OFF and the Mode pins must be set corresponding to the application. For single chip Mode the setting is: SW1 OFF, SW2 OFF, SW3 ON. After Power-on or Reset the application will start now.

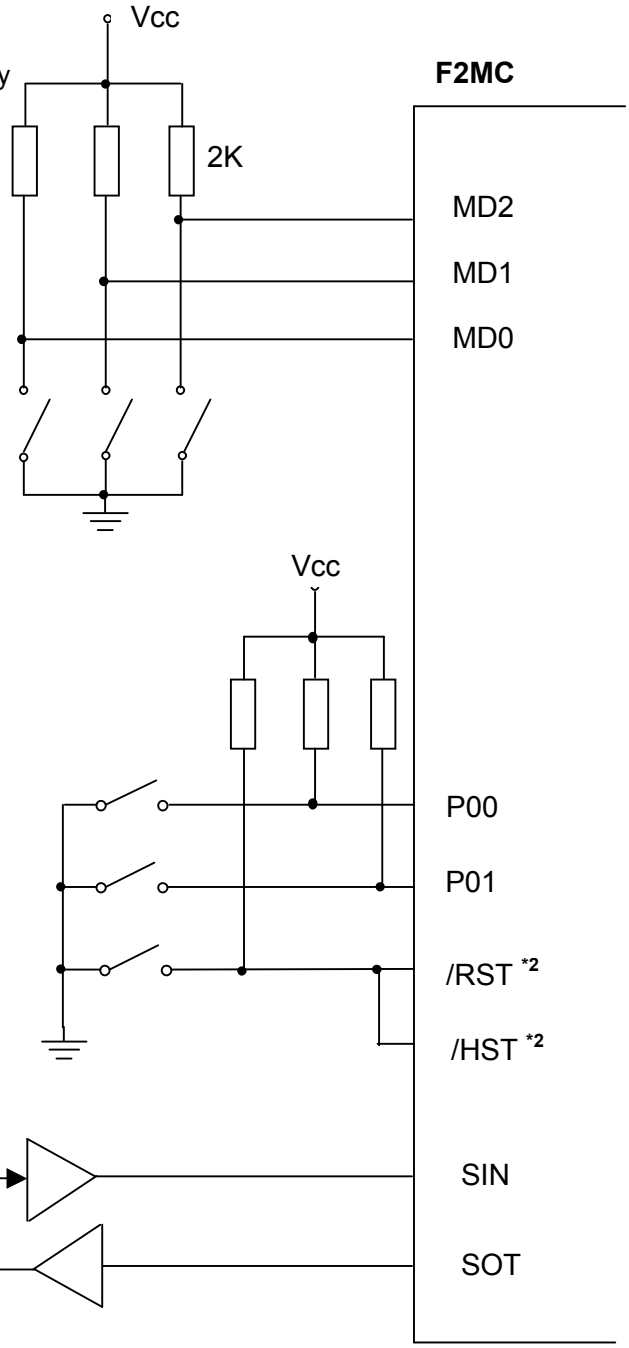
NOTE: MB90F347/438/439/443G/462/543/546/548/549/562B/598G/867 programming

The MB90F543 has a Flash security feature, which is enabled by setting bit 0 of address $FE0001. This enables a read protection for the internal flash memory, so nobody can read out the flash memory anymore (For detailed description please see Hardware Manual of the series).

If the flash security bit is set, it is not possible to program or erase the Flash with the Flash361/400/500/612/V01L03 program anymore!

The reason for this is that only a chip erase command is excepted when the security is set. But the Flash362/400/500/612/V01L03 software uses subsequent single sector erase commands to erase the chip.

Figure 1
Hardware configuration for Serial Asynchronous Programming



NOTE: Even 2K pull up resistors are used here, it is recommended to connect the MDx pins directly to Vcc or GND.

Pin	MD2	MD1	MD0
Level	HIGH	HIGH	LOW

Table 14: Mode Pin setting

Pin (*1)	P01	P00
Level	LOW	LOW

Table 15: Port pin setting

1: MB90385 series is using P30 and P31 instead P00 and P01. MB90470 series is using P80 and P81 instead of P00 and P01. When using MB90390 with (5/10/20) MHz external clock use setting: P00: HIGH, P01: LOW. Refer to chapter: 3.3 Mode Setting

2: RST = Reset, HST = VCC when using MB 9054xGHDS series

4 Working with a User Application Software (Bootloader)

4.1 General Description

As already mentioned there exist an Application Note for an examples of a user bootloader for the MB90F523, MB90F543, MB90F553, MB90F574, MB90F583, MB90F594/A and the MB90F598. The functionality of such a bootloader will be described now, with this bootloader for the MB90F523 as a reference. Take care that the bootloader examples supplied for other devices may slightly be different. There are some small differences regarding baud rate and handling of the application reset vector. Therefore please see the table „Bootloader Selector Guide" afterwards.

NOTE:

The bootloaders have been carefully checked and are believed to be reliable. However, no responsibility is assumed for any inaccuracies. Fujitsu does not accept any liability arising out of the use of this application example.

All bootloaders use the UART of the microcontroller for communication and download. Only the transmit and receive line of the corresponding UART interface are used. After Power-on the user has 1sec to send the character <ESC> via a terminal to the microcontroller. Therefore e.g. the SKWizard terminal emulation program can be used. If no <ESC> character is sent, the application is called after a timeout of approximately 1-second.

The SKWizard offers some more features, especially to download software. After sending the character, the bootloader enters a monitor mode and a prompt (>) appears on the terminal. Now some commands can be entered (commands are listed below) or the download of the application software can be started. With the SKWizard the download can be started by the load button or with the following command line:

SKWizard 1 -sk16 -i19200 -r -c %x\%A.cnv

Instead of using the SKWizard it is also possible to use the HEXLOADW download-utility to transfer the linker-output file. The settings (Option-SetUtility-menu) should be

HEXLOADW.EXE 1 -Flash -w -c -i19200 %x\%A.cnv

HEXLOADW automatically erases the flash memory sectors respectively and downloads the application program code.

4.2 Different methods to reprogram the flash memory

There are two methods to download programs into Flash memory,

Method A and Method B.

In the previous example, the application has been programmed into the flash-memory using Method A.

Method A: The entire sector will be erased before a byte will be written to it. Method A does not check, whether the content of the sector is different from the data that should be written to it or not. Hexloadw supports only this method. SKWizard also works with Method B.

Method B: In this case the code will be transferred twice to the Flash-Board. During the first transmission, the flash-monitor checks for differences between the actual contents and the received data. During the second transmission, it will only erase those sectors, to which new data should be written. This method also takes into account, that a flash memory cell only has to be erased, if their data-bit has to be reprogrammed from logical "0" to "1".

To choose Method B with the SKWizard add an the –flash option in the command line:

SKWIZARD.EXE 1 –flash -sk16 -i19200 –d

4.3 Application Call

After Power-on of the MCU, the bootloader always takes control, at least for the first steps to check the cause of reset.

After Power On the version string of the bootloader "MB90523 Flash loader - V2.0" is output and you have the chance to respond with ESC in order to enter the monitor mode of the bootloader.

If „ESC" is pressed during the timeout period, the program execution branches to the Flash Monitor.

Note that the loader reads the WDTC (WatchDog Timer-Control) register in order to find out the cause for reset. While reading WDTC, the content is destroyed, but through the R0 register, a copy of the content is passed to the application software for further evaluation.

If you download an application, which normally has its own reset vector at H'FFFFDC, this "application vector" will be stored in INT#7 location at address H'FFFFE0. Therefore a vector call for INT#7 is no longer available in conjunction with this boot loader. The boot loader itself takes care of this vector displacement. This way of handling with the reset vector is very flexible because the "application vector" can be defined in the software project inside Softune as the normal reset vector and is not fixed to a certain address.

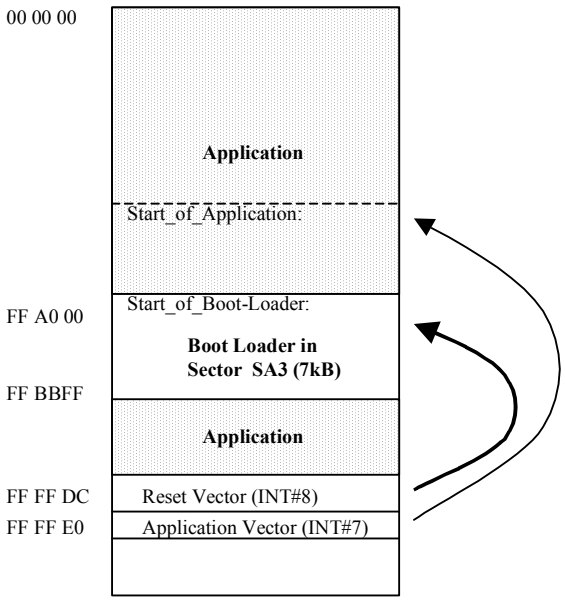


Figure 13: Location of reset vector and application vector

4.4 Special Note for MB90F553 V2.3 Bootloader

The bootloader for the MB90F553 V2.3 does not support this INT#7 feature. For this bootloader the application start address is fixed to H'FF0000. So if the bootloader V2.3 of the MB90F553A is used, the application must start at address H'FF0000 and no Reset Vector may be generated by the compiler for the application. Otherwise the Reset Vector of the bootloader would be overwritten, which is protected by the bootloader software. But if a reset vector is included the download will be stopped.

4.5 Bootloader Commands

After Power-on the message "MB90523 Flash loader - V2.0" appears as the first prompt at the terminal. Respond with "ESC" within 1 second enters the monitor mode and a prompt ">" appears. In the monitor mode of the bootloader the following commands can be entered:

4.5.1 Command overview

- RM AAAAAA NN : "Read Memory": NN bytes from location AAAAAA*)
- ES NN : "Erase Sector" : Erases sector NN
- G : "Go" : Call user application
- CALL AAAAAA : "Call" : Call address AAAAAA
- PROTECT_RESET_VECTOR OFF : Disables write-protect function for reset-vector
- PROTECT_RESET_VECTOR ON : Enables write-protect function for reset-vector
- PROTECT_SECTORS OFF : Disables erase-protect function
- PROTECT_SECTORS ON : Enables erase-protect function

- CS OFF : Switch checksum-algorithm off
- RST : Software reset of MCU
- HEXDWL : Switch to Download-Mode (Method A)
- HEXDWL1 : Switch to Download-Mode (Method B – first step)
- HEXDWL2 : Switch to Download-Mode (Method B – second step)
- (Escape) : any ESC aborts the current function

*) AAAAAA: 24-bit upper case hex address, NN: two-digit upper case hex number
Defaults:
PROTECT_SECTORS ON
PROTECT_RESET_VECTOR ON
to protect the monitor from erasing the reset vector and itself.

4.6 Modifying the Bootloader

The provided bootloader is just an example how to generate such kind of "bootloader". It can be adapted to the needs of the application. For that purpose the source code of the current monitor must be modified. Therefore the following instructions must be considered:

a) During erasing and reprogramming of the on-chip flash, the code must be executed from the RAM! Therefore, the monitor copies part of the code from ROM into the RAM area. For the Softune Workbench, the linker can provide a dedicated mechanism for that purpose: RAMCode. This feature allows to link code to ROM, which will be executed in the RAM later on.

   (Note: When you are still working with the old Softune V01 development environment the following issues must be considered: For the linker, the flash routines of the code must be placed in a RAM-location during compilation-time. So this part of code has to be copied back to ROM before programming a new monitor into the Flash memory. To do so, use the BINHEX-tool (Version 2.1 or later):

   BINHEX /z=FFA000 NAME.cnv /y /a

   This will copy all code linked to RAM-locations (below 1000hex) to FFAxxxhex. The output file with the corrected load-module will be in Motorola S-format with the extension *.mhx by default.)

b) If the device supports the serial Burn-In ROM programming, it is possible to download the bootloader via RS232 to a blank device using the special serial programming software (described in chapter 2).

c) A blank MB90F523 or MB90F594 devices can only be programmed using an EPROM-programmer like the MINATO 1890A or the Data I/O Plus48. A special programming adapter is necessary for each programmer.

4.7 Updating the Bootloader

If a bootloader is already programmed in the flash memory it is possible to update the bootloader by using the bootloader itself. Therefore the following procedure must be done:

Before downloading the new bootloader to the flash device , also the reset-vector protection and sector-protection of the current bootloader must be switched off. Otherwise, any attempt to overwrite the current version of the bootloader will result in an error. Use the commands:

PROTECT_RESET_VECTOR **OFF**

and

PROTECT_**SECTORS OFF**

from the SKWizard or any terminal program to allow overwriting of the bootloader area and download the new bootloader file to the flash memory. After reset (/HST or Power On) the new version of the bootloader should appear. Take care that during the programming sequence the power supply is not switched off. Otherwise the programming could fail and the bootloader will not work anymore.

5 Command Line Options

5.1 HEXLOADW

HEXLOADW {PORT} [-R|-C] [-W] [-Flash] [-Ibaud] [file]

PORT This is the communication port that the board is connected to.

-W Wait for target (Power on)

-C Close HexloadW after download

-R Run program, do not use for Flash boot loader

-Ibaud Initialise serial port to baud (300, 1200, 2400, 4800, 7200, 9600, 19200, 38400).

-Flash Flash-programming using Method A

Examples:

HEXLOADW 1 –i38400 -flash –w myprog.cnv

Waits for the target to be reset/switched on, programs myprog to flash via COM-port 1 using method A

HEXLOADW 2 –i19200 –flash –c myprog.cnv

Programs myprog to flash via COM-port 2 using method A, and shuts down

5.2 SKWIZARD

SKWIZARD [P] [-IBaud] [-SKType] [-D] [-R [-C]] [File]

P COM-Port Number (1,2,3,4)

Baud Baudrate (300,1200,2400,4800,9600,19200,38400)

Type Starterkit-Type (8,16,32)

-D Detect Baudrate (synchronises Baudrate with board)

-R Run after load (only useful if file is specified)

-C Close program after a successful load and execution (only if -R is specified)

-F Flash-programming using Method B

File (last parameter given) should contain full path and filename+ext of hex-file

Examples:

SKWIZARD 1 –SK16 –i19200

Simply opens the application for use with the Flash-Board via COM-port 1

SKWIZARD 1 -SK16 -i19200 myprog.cnv

As above, but will program myprog to flash using method A

SKWIZARD 2 -SK16 -i19200 –f –r -c myprog.cnv

Now uses COM 2 and will program myprog to flash using method B, then execute it and shuts down

5.3 BINHEX

BINHEX /o=-FF0000 /z=FFA000 /m test.cnv /o=FF0000 /a

/o is the read offset (in this case negative)

/z is new offset for records below 1000 hex

/m is for Motorola format

test.cnv is the input file

/o is the write offset (to put the record back where it started)

/a is to change addresses only

6 Bootloader Selector Guide

Device Loader name	Loader address	Appl. start address / reset vector	Start condition	Comm	Loading
MB90F347 V1.0	boot sector linked to: FFA000 – FFBBFF	Reset vector **must** be used, Vector stored at FFFFE0 (int#7) by loader Dummy application included, which toggles port P10	- Boot loader checks WDTC, if POWER-ON or HST reset → start main loader, if other reset cause → jump to application pointed by FFFFE0 - Main loader displays WELCOME message - waits 1 second for any character, if not → PLL off, generate software reset, if something → command line mode - WDTC stored in R0	UART0 at 38400 Baud (PLL x4)	loading by SK-Wizard: - power-on target - press ESC after message - proceed as usual loading with Hexloadw: -call "hexloadw 1 -flash -w -c path\name.cnv" - power-on target
MB90F352 V1.0	boot sector linked to: FFA000 – FFBBFF	Reset vector **must** be used, Vector stored at FFFFE0 (int#7) by loader Dummy application included, which toggles port P10	- Boot loader checks WDTC, if POWER-ON or HST reset → start main loader, if other reset cause → jump to application pointed by FFFFE0 - Main loader displays WELCOME message - waits 1 second for any character, if not → PLL off, generate software reset, if something → command line mode - WDTC stored in R0	UART3 at 38400 Baud (PLL x4)	loading by SK-Wizard: - power-on target - press ESC after message - proceed as usual loading with Hexloadw: -call "hexloadw 1 -flash -w -c path\name.cnv" - power-on target
MB90F497G V1.4	boot sector linked to: FFA000 – FFBBFF	Reset vector **must** be used, Vector stored at FFFFE0 (int#7) by loader Dummy application included, which toggles port P10	- Boot loader checks WDTC, if POWER-ON or HST reset → start main loader, if other reset cause → jump to application pointed by FFFFE0 - Main loader displays WELCOME message - waits 1 second for any character, if not → PLL off, generate software reset, if something → command line mode - WDTC stored in R0	UART0 at 38400 Baud (PLL x4)	loading by SK-Wizard: - power-on target - press ESC after message - proceed as usual loading with Hexloadw: -call "hexloadw 1 -flash -w -c path\name.cnv" - power-on target

Device Loader name	Loader address	Appl. start address / reset vector	Start condition	Comm	Loading
MB90F523 V3.0	boot sector linked to: FFA000 – FFBBFF	Reset vector **must** be used, Vector stored at FFFFE0 (int#7) by loader Dummy application included, which toggles port P10	- Boot loader checks WDTC, if POWER-ON or HST reset → start main loader, if other reset cause → jump to application pointed by FFFFE0 - Main loader displays WELCOME message - waits 1 second for any character, if not → PLL off, generate software reset, if something → command line mode - WDTC stored in R0	UART0 at 19200 Baud (PLL x4)	loading by SK-Wizard: - power-on target - press ESC after message - proceed as usual loading with Hexloadw: -call "hexloadw 1 -flash -w -c path\name.cnv" - power-on target
MB90F543 V1.1	Fixed boot sector: FFA000 – FFBBFF ———	Reset vector **must** be used, Vector stored at FFFFE0 (int#7) by loader Dummy application included, which toggles port40	- Boot loader checks WDTC, if POWER-ON only → start main loader, if other reset cause → jump to application pointed by FFFFE0 - Main loader displays WELCOME message - waits 1 second for any character, if not → PLL off, generate software reset, if something → command line mode - WDTC stored in R0	UART0 at 38400 Baud (PLL x4)	loading by SK-Wizard: - power-on target - press ESC after message - proceed as usual loading with Hexloadw: - call "hexloadw 1 -flash -w -c path\name.cnv" - power-on target
MB90F553 V4.0	Sector: FFA000...FFBBFF	Reset vector **must** be used, vector stored at FFFFE0 (int#7) by loader (vector displacement) dummy application included, that does a endless loop	- Boot loader checks WDTC, if POWER-ON or HST only → start main loader, if other reset cause → generate software reset and start again - Main loader displays WELCOME message - waits 1 second for ESC, if not → PLL off, generate software reset, if ESC → command line mode - WDTC stored in R0	UART0 at 19200 Baud (PLL x4)	loading by SK-Wizard: - power-on target - press ESC after message - proceed as usual loading with Hexloadw: - call "hexloadw 1 -i19200 -flash -w -c path\name.cnv" - power-on target
MB90F574 V1.1	Sector: FF8000 – FF9FFF	Reset vector **must** be used, vector stored at FFFFE0 (int#7) by loader (vector displacement) dummy application included, that does a endless loop	- Boot loader checks WDTC, if POWER-ON or HST only → start main loader, if other reset cause → generate software reset and start again - Main loader displays WELCOME message - waits 1 second for ESC, if not → PLL off, generate software reset, if ESC → command line mode - WDTC stored in R0	UART0 at 38400 Baud (PLL x4)	loading by SK-Wizard: - power-on target - press ESC after message - proceed as usual loading with Hexloadw: - call "hexloadw 1 -i19200 -flash -w -c path\name.cnv" - power-on target

Device Loader name	Loader address	Appl. start address / reset vector	Start condition	Comm	Loading
MB90F583 V1.0	Fixed boot sector: FFA000 – FFBBFF	Reset vector **must** be used, Vector stored at FFFFE0 (int#7) by loader Dummy application included, which toggles port40	- Boot loader checks WDTC, if POWER-ON only → start main loader, if other reset cause → jump to application pointed by FFFFE0 - Main loader displays WELCOME message - waits 1 second for any character, if not → PLL off, generate software reset, if something → command line mode - WDTC stored in R0	UART0 at 19200 Baud (PLL x4)	loading by SK-Wizard: - power-on target - press ESC after message - proceed as usual loading with Hexloadw: - call "hexloadw 1 -flash -w -c path\name.cnv" - power-on target
MB90F594 MB90F598 V2.0	Fixed boot sector: FFA000 – FFBBFF The binary of this loader fits for both MB90F590 and 595 Be aware that the e 590 header files do not completely fit for 595.	Reset vector **must** be used, Vector stored at FFFFE0 (int#7) by loader Dummy application included, which toggles port40	- Boot loader checks WDTC, if POWER-ON only → start main loader, if other reset cause → jump to application pointed by FFFFE0 - Main loader displays WELCOME message - waits 1 second for any character, if not → PLL off, generate software reset, if something → command line mode - WDTC stored in R0	UART0 at 38400 Baud (PLL x4)	loading by SK-Wizard: - power-on target - press ESC after message - proceed as usual loading with Hexloadw: - call "hexloadw 1 -flash -w -c path\name.cnv" - power-on target

Table 16: Bootloader selector guide

NOTE: The bootloaders have been carefully checked and are believed to be reliable. However, no responsibility is assumed for any inaccuracies. Fujitsu does not accept any liability arising out of the use of this application example.

7 Troubleshooting

7.1 Fujitsu FlashMCU Programmer 16LX

7.1.1 Error Messages

Refer also to Fujistu Flash MCU Programmer for F²MC-16Lx Specification Manual

No.001 Download error*1

Cause: Downloading failed

Action: Return the folder and file configurations to the installation defaults.

No.003 Timeout error

Cause: The microcontroller does not respond.

(Not changed to flash memory reprogramming mode)

Action: Recheck the setting of pins used for reprogramming flash memory.

No.006COM port open error

Cause: The COM port is disabled.

Action: Enable the COM port.

No.007 Download file open error

Cause: m_flash.xxx not found

Action: Return the folder and file configurations to the installation defaults.

No.008 File size get error

Cause: File access failed

Action: Check whether the PC is unstable.

No.009 COM port setting information get error

Cause: The COM port is disabled.

Action: Enable the COM port.

No.010 COM port setting information change error

Cause: The COM port is disabled.

Action: Enable the COM port.

No.011 Communication error

Cause: The microcontroller returned a communication error.

Action: Re-execute the command or replace the chip.

No.012 Read error

Cause: Data cannot be read from flash memory in the microcontroller.

Action: Re-execute the command or replace the chip.

No.013 Write error

Cause: Data cannot be programmed to flash memory in the microcontroller.

Action: Re-execute the command again or replace the chip.

No.015 COM port write error

Cause: The COM port is disabled.

Action: Check the RS-232C cable connected to the COM port.

No.016 COM port read error

Cause: The COM port is disabled.

Action: Check the RS-232C cable connected to the COM port.

No.017 File access error

Cause: m_flash.xxxnot read

Action: Return the folder and file configurations to the installation defaults.

No.018 Erase error *1

Cause: Erasing failed

Action: Return the folder and file configurations to the installation defaults.

No.101 Set "hex file."

Cause: "Hex file"not set

Action: Set "hex file"in the dialog box.

No.102 Batch command error

Cause: An error occurred at batch command execution

Action: Return the folder and file configurations to the installation defaults.

No.103 Invalid "hex file"

Cause:The selected "hex file"is invalid.

Action:Select the S2 file as the "hex file."

No.207 Memory allocation error

Cause:Unable to allocate memory for execution

Action:Quit any running application and retry.

Please redo from download operation *2

*1:"MCU xxH"is displayed if the error cause is returned from the microcontroller at a download error.

"MCU xxH" means:

MCU 02H SUM error at downloading

MCU 04H Abnormal termination at downloading

*2:This is an additional message. It is displayed as necessary after other messages are displayed.

7.1.2 Using Laptops

Failure: 'No.003 Timeout error' is displayed

Description: FlashMCU Programmer 16LX is unable to start a correct communication, because of the default COM port Level (L-Level) of some Laptops.

Background: Laptop'S are using Power saving modes especially when powered by battery. To avoid problems, it is recommended to switch off the Power saving modes. (check BIOS of the Laptop)

Workaround: use V01L07 version or higher

7.1.3 Wrong file is programmed

7.1.3.1 Using version V01L03, V01L05 or V01L07

Description:

Instead of the actual file an older version is programmed into the Flash.
The FlashMCU Programmer 16LX is converting the Motorola-hex file into a binary file. This file is stored as *.bin inside the *.mhx folder.
If no *.mhx files are used, the *.bin file is not updated. The FlashMCU Programmer 16LX is directly programming the existing *.bin file.
Workaround:

- Use *.mhx file.
- Delete existing *.bin file before start programming a new file. (when using other extentions)

7.1.3.2 Using Flash version 3xx, 4xx, 5xx or 6xx with Win NT

Description:

In some cases it can happen, that this software version of the software does not fully support Windows NT. This is because some procedures are done using batch files. So it could happen that an error occurs using Windows NT (Error 103, wrong write file).
Workarround:
In case of error, please look at the c:\softune\flashxxx directory and make a copy of the temporary created batch file. Start the batchfile separately first. After that the Flash programming software itself can be started. The batchfile creates a work.bin file, which is used by the Flash programming software.

7.1.4 Unable to program file

Failure: "error 103, wrong write file"
Description:
In some cases rare "error 103, wrong write file" can occur. Check if large folder trees are used for the FlashMCU Programmer 16LX or the *.mhx file. When programming via Network check if Network connecting is working properly, try to program from local disk.
Workaround:
Copy the motorola S-Record which should be programmed to the internal Flash memory (myname.mhx) to the installation directory of the FlashMCU Programmer 16LX software or try to install the FlashMCU Programmer 16LX directly in the root directory.

8 Programming Time

The programming time depends on the communication mode (serial-asynchronous / -synchronous, parallel) and the tool that is used for programming. Further, the age of the flash may influence the programming time, too. The flash characteristics are shown in the datasheet of the related microcontroller.

The times given below should give a notion, but exact program-times have to be requested by the tool-provider for each microcontroller case by case.

8.1 Serial asynchronous mode

8.1.1 Tool: Fujitsu MCU Flash Programmer (freeware)

Device: MB90F543G, 128K, 4MHz, 38400Baud

Chip-Erase:	4sec	Blank-Check:	1sec
Program:	N/A	Program incl. verify:	2min 30sec
Verify (CRC):	N/A	Verify (Full):	1min 20sec

Device: MB90F347, 128K, 4MHz, 115kBaud

Chip-Erase:	4sec	Blank-Check:	1sec
Program:	N/A	Program incl. verify:	18sec
Verify (CRC):	N/A	Verify (Full):	15sec

8.1.2 Tool: FlashKit

Device: MB90F543G, 128K, 4MHz, 38400Baud

Chip-Erase:	4sec	Blank-Check:	1sec
Program:	45sec	Program incl. verify:	N/A
Verify (CRC):	N/A	Verify (Full):	55sec

8.1.3 Tool: GALEP-4

Device: MB90F543G, 128K, 4MHz, 38400Baud

Chip-Erase:	11sec	Blank-Check:	9sec
Program:	1min 10sec	Program incl. verify:	1min 55sec
Verify (CRC):	8sec	Verify (Full):	56sec

Device: MB90F347, 128K, 4MHz, 38400Baud

Chip-Erase:	11sec	Blank-Check:	9sec
Program:	1min 10sec	Program incl. verify:	1min 55sec
Verify (CRC):	8sec	Verify (Full):	56sec

8.2 Serial synchronous mode

8.2.1 Tool: FlashKit

Device: MB90F543G, 128K, 4MHz

Chip-Erase:	4sec	Blank-Check:	1sec
Program:	8sec	Program incl. verify:	N/A
Verify (CRC):	4sec	Verify (Full):	25sec

Device: MB90F347, 128K, 4MHz

Chip-Erase:	4sec	Blank-Check:	1sec
Program:	8sec	Program incl. verify:	N/A
Verify (CRC):	4sec	Verify (Full):	25sec

8.2.2 Tool: GALEP-4

Device: MB90F543G, 128K, 4MHz

Chip-Erase:	5sec	Blank-Check:	3sec
Program:	13sec	Program incl. verify:	36sec / 19sec (CRC)
Verify (CRC):	6sec	Verify (Full):	23sec

Device: MB90F347, 128K, 4MHz

Chip-Erase:	5sec	Blank-Check:	3sec
Program:	13sec	Program incl. verify:	36sec / 19sec (CRC)
Verify (CRC):	6sec	Verify (Full):	23sec

8.2.3 Tool: Yokogawa AF221

Device: MB90F543G, 128K, 4MHz

Chip-Erase:	9sec	Blank-Check:	4sec
Program:	5sec	Program incl. verify:	7sec (CRC)
Verify (CRC):	N/A	Verify (Full):	N/A

8.3 Parallel mode

8.3.1 Tool: DATA-IO

Sprint-Family: Plus48, Optima, Dual, Quad, Octal, PP100, PS200

128K-devices: MB90F543G, MB90F347, etc.:

Chip-Erase:	4sec	Blank-Check:	1sec
Program:	38sec	Program incl. verify:	N/A
Program incl. CRC-Verify:	N/A	Verify:	5sec
Readout:	18sec		

FlashCore - Family: FlashPAK - Proline RoadRunner - PS300FC

128K-devices: MB90F543G, MB90F347, etc.

Chip-Erase:	4sec	Blank-Check:	0.1sec
Program:	2.1sec	Program incl. verify:	N/A
Program incl. CRC-Verify:	N/A	Verify:	0.2sec
Readout:	10sec		

8.3.2 Tool: ANDO

AF9723 (gang type, adapter TE100-553F01A)

128K-devices: MB90F553A

Chip-Erase incl. Blank-Check:	14sec
Program incl. Verify:	14sec
Blank-Check, Program, Verify:	17sec

AF9708 (single programming type)

128K-devices: MB90F553A

Chip-Erase incl. Blank-Check:	13sec
Program incl. Verify:	15sec