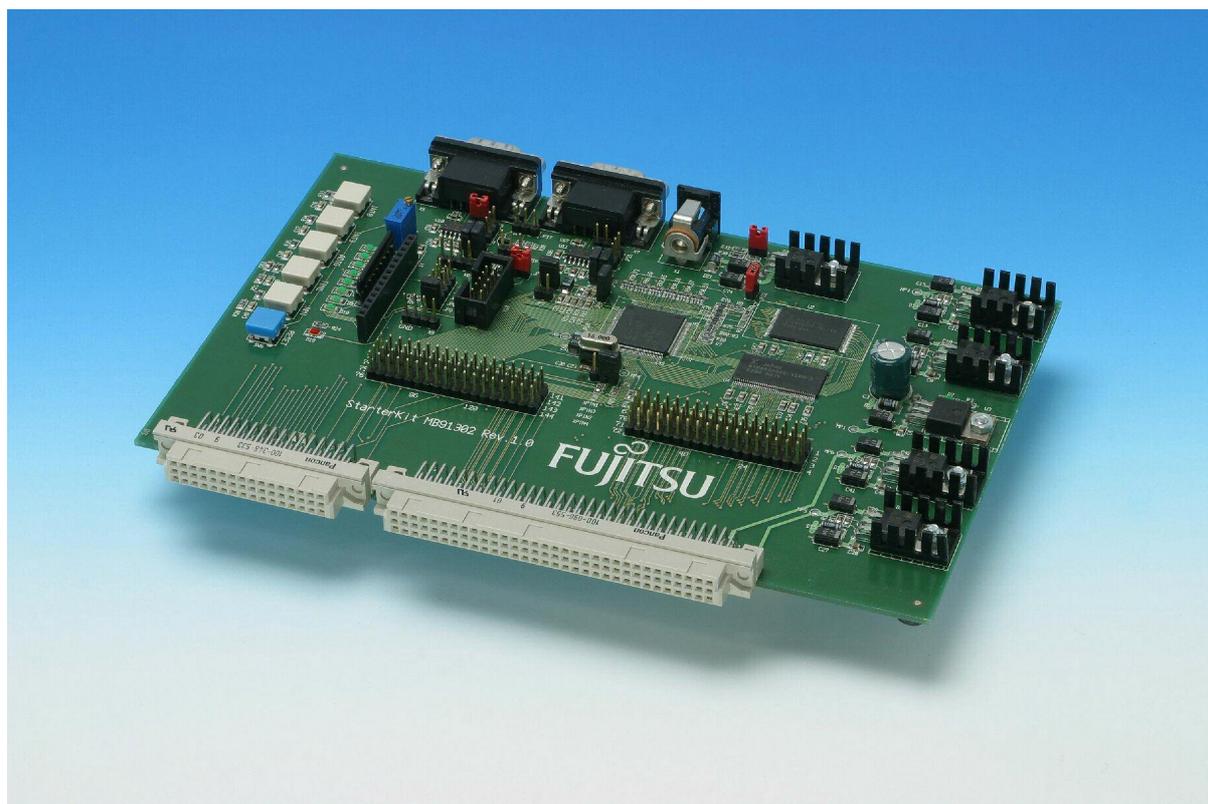


FR FAMILY EVALUATION BOARD STARTERKIT MB91302

USER GUIDE



Revision History

Date	Issue
2004-07-09	V1.0,TKa/HW, First Release
2004-07-26	V1.1, Tka, chapter 7.5, 7.7.3, 8 updated
2004-11-17	V1.2, Tka, Chapter 9 Programming corrected – COM2 removed
2005-01-14	V1.3, Tka, programming area corrected, page 45 Table with pin description for connector X2, X3 added
2005-01-18	V1.4, Tka, table in chapter 6.3 corrected
2005-01-31	V1.4, Tka, version number in bookmarks corrected

This document contains 54 pages.

Warranty and Disclaimer

To the maximum extent permitted by applicable law, Fujitsu Microelectronics Europe GmbH restricts its warranties and its liability for the “**STARTERKITMB91302**” Board and all its deliverables (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 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: The “STARTERKITMB91302” Board and all its deliverables 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 from 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 supplier’s liability are 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

1 Contents

REVISION HISTORY	2
WARRANTY AND DISCLAIMER	3
1 CONTENTS.....	4
2 OVERVIEW.....	6
2.1 Abstract.....	6
2.2 Features.....	6
2.3 General Description.....	7
3 INSTALLATION	8
3.1 Hardware Installation.....	8
3.1.1 Power-Supply configuration	8
3.1.2 STARTERKITMB91302 Configuration	9
3.1.3 Use as Emulation Target Board for Emulator MB2198-01.....	10
3.2 Software Installation.....	11
3.2.1 Installation of Softune Workbench	11
4 QUICK PROJECT START-UP / TEST “STARTERKITMB91302”	12
5 JUMPERS AND SWITCHES.....	14
5.1 Jumper Overview	14
5.2 Operating-Mode (JP1).....	16
5.3 Power Supply Voltage (JP: 22).....	16
5.4 Analogue Power Supply Voltage (JP: 2, 4, 7).....	17
5.5 UART”A” (JP: 12, 13, 14)	18
5.6 UART”B” (JP: 18, 19, 20)	18
5.7 Reset-Generation (JP: 11, 15, 16, 17, 21).....	19
5.8 Buttons INT0, INT3, ADTG, TIN1, NMI(JP: 3, 5, 6, 8, 9).....	20
6 CONNECTORS.....	21
6.1 Power connector (X1).....	21
6.2 Edge connector (XPIN1, XPIN2, XPIN3, XPIN4)	21
6.3 UART”A”, UART”B” connector (X5, X7).....	21
6.4 USER-LEDs & LC-Display (optional).....	22
6.5 Graphic Board Interface	23
7 GETTING STARTED.....	25
7.1 Introduction to Softune Workbench	25
7.2 Project Start-up	27

7.3	Softune Workbench Monitor Debugger.....	32
7.3.1	General Description.....	32
7.3.2	Basic Debugger Features.....	35
7.4	Advanced Monitor Debugger Features.....	36
7.5	Memory Configuration for User Applications.....	39
7.6	Provided Examples.....	41
7.7	Advanced Monitor Debugger Information.....	42
7.7.1	Status Indication of Monitor Debugger after Reset.....	42
7.7.2	Memory Map of Softune Workbench Monitor Debugger.....	43
7.7.3	User Configuration of the Softune Workbench Monitor Debugger....	44
8	STARTERKIT LIMITATIONS.....	46
9	PROGRAMMING THE EXTERNAL FLASH MEMORY.....	47
10	TROUBLE SHOOTING.....	50
11	SILK-PLOT OF THE BOARD.....	51
12	RELATED DOCUMENTS.....	52
12.1	Application notes.....	52
12.2	Manuals.....	52
12.3	Website Links.....	52
13	RELATED PRODUCTS.....	53
14	INFORMATION AND CONTACTS.....	54

2 Overview

2.1 Abstract

The “STARTERKITMB91302” is a multifunctional starterkit for the Fujitsu 32-Bit Flash microcontroller MB91302series. It can be used stand alone for software development and testing or as a target board to work with the emulator system. The board enables the designer to start immediately with the software development before his own final target system is available.

The “STARTERKITMB91302” comes with the following parts

- “STARTERKITMB91302” and Softune Workbench monitor debugger
- Serial cable D-Sub 9 pin male-female (1:1)
- Power-Supply
- Printed version of the user guide for “STARTERKITMB91302”
- “STARTERKITMB91302” CD-ROM
- Microcontroller CD-ROM V3.5 or higher
- Addendum sheet with latest important instructions

2.2 Features

The starterkit is being delivered with the MB91302APFF-G-020-BND-E1 microcontroller and a monitor debugger installed already in the external flash.

- ▶ Supports MB91302series in FPT-144P-M12 package
- ▶ On-board 64Mbit Flash and 64Mbit SDRAM
- ▶ In-Circuit serial Flash programming
- ▶ Integrated Softune Workbench monitor debugger
- ▶ All resources available for evaluation on connectors
- ▶ All 144 pins routed to connectors
- ▶ 15 MHz main-crystal
- ▶ Two UART Interfaces (UART0, 1)
- ▶ 8 User LEDs, optional: alphanumeric standard LC-Display connectable instead of LEDs
- ▶ Reset-button, Reset-LED
- ▶ 5 User-buttons configurable for INT0, INT3, ADTG, TIN1 and NMI
- ▶ 6-9V (500mA) unregulated external DC power supply
- ▶ 3V on-board voltage regulator with power status-LEDs
- ▶ Graphic-Controller board support:
 - ▶ 96pin + 48pin VG connector
 - ▶ Power-supply (+5V, +3V3, +2V5)

**This board must only be used for test applications
in an evaluation laboratory environment.**

2.3 General Description

The “STARTERKITMB91302” supports microcontroller members of the MB91301series. It can be used as a stand-alone evaluation board or as a target board for emulator debugger.

The evaluation board supports the following package: FPT-144P-M12

The board is supplied with a socket 15MHz crystal as main oscillation clock. Using the internal PLL of the μ C, internal clock rates up to 60MHz can be achieved.

Two separate RS232 transceivers are available to connect the on-chip UART0 and UART1 to 9-pin D-Sub connectors (X5, X7). The transceivers generate the adequate RS232 levels for receive (RXD) and transmit (TXD) lines. Either the DTR line or the CTS line can be selected with jumpers (JP15, JP17, JP21) to generate a system reset. The RTS signal can be shortcut to CTS using the jumpers JP12 and JP18.

The external flash memory is programmed with a monitor-debugger, which uses UART0 (UART“A”; X5) for communication and to download the user-application.

If the board is used as an emulator target board, the microcontroller must be removed from the PCB and a socket (NQPACK144SE) must be soldered instead. The following configuration has to be used:

Emulation main unit: **MB2198-01**

DSU cable: **MB2198-10**

Probe Cable: **MB2198-101**

Socket/ Header:**NQPACK144SE + HQPACK144SE**

All pins of the microcontroller are connected to the edge connectors XPIN1 to XPIN4. Furthermore, on X2 and X3 Fujitsu graphic-controller starterkits (daughter boards) can be connected.

The on-board voltage regulator allows the user to connect an unregulated DC input voltage between +6V to +9V (500mA). Also the power-supply voltages for an optional graphic-controller starterkit are generated. In this case the input voltage should be +8V to +9V (1500mA). In case of any modifications of the board, care should be taken that the total power consumption will not damage the regulator.

There are six push buttons on the board, used for Reset, External Interrupts INT0 and INT3, Trigger for A/D-converter (ADTG) and Reload-Timer1 (TIN1) and a non-maskable interrupt (NMI) – used for Abort function of Softune Workbench monitor debugger.

Eight user-LEDs are connected via a 220 Ohm pull-down resistor network to Port PG0-PG7. Parallel to the LEDs the connector X4 can be used to connect a standard alphanumeric display. If neither LEDs nor LCD are required, the resistor network can be removed to free the port.

The jumper JP1 (MD0) can select the operating mode of the microcontroller.

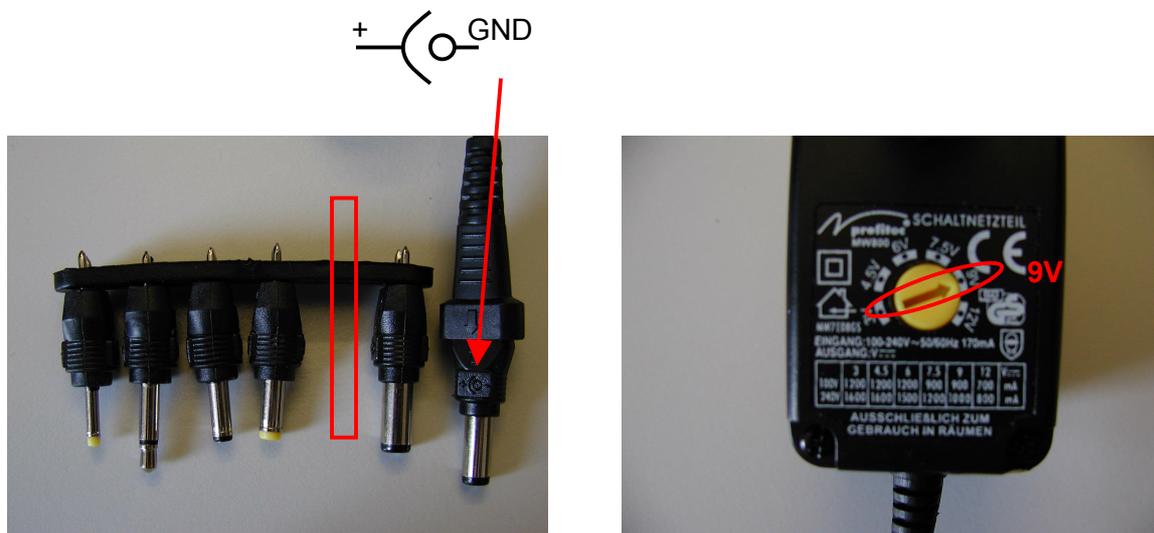
3 Installation

3.1 Hardware Installation

3.1.1 Power-Supply configuration

The STARTERKITMB91302 contains a universal AC/DC power-supply adapter. Before connecting the power-supply to the starterkit ensure the correct configuration of the power-supply:

1. Voltage **selection to 9V**
2. Appropriate **DC plug** (2nd largest plug is most suitable) is plugged into the socket.
3. Adjust correct **polarity**:



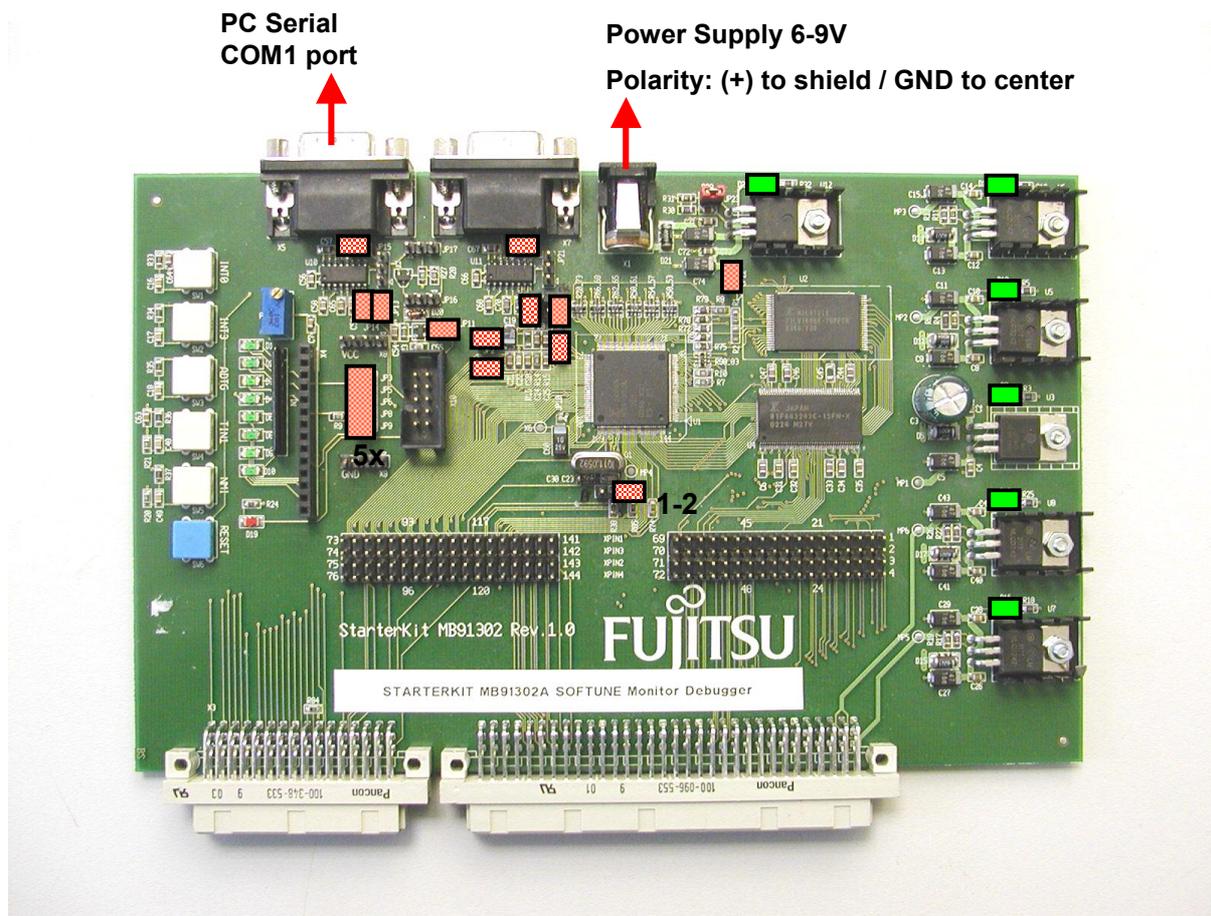
After these settings are checked the supply is ready to be used with the STARTERKITMB91302.

3.1.2 STARTERKITMB91302 Configuration

Carefully remove the board from the shipping carton. Check if there are any damages first before power on the starterkit.

For the power supply a DC input voltage of 6V-9V/500mA is recommended (8V-9V/1500mA if graphic sub-boards are used). The positive voltage (+) must be connected to the shield, and ground (GND) must be connected to the centre of the connector X1!

The following picture shows the default jumper settings as a short reference. For details please check the chapter 4.1 Jumpers and Switches.



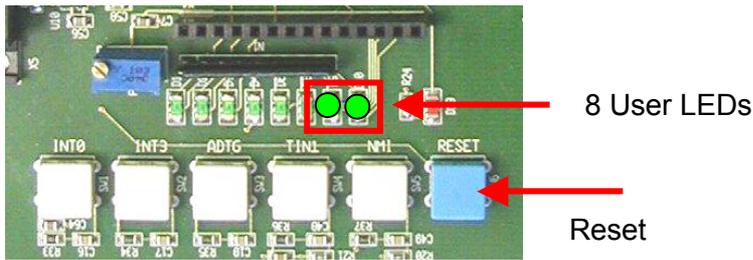
Basic set-up configuration with default jumper settings and power LEDs

After power-on push the blue reset button RESET (SW6) to ensure proper start-up. After power-on the green power status LEDs (D22, D14, D12, D7, D10, D16) should be on. If the LEDs do not light switch off the power supply and check the default jumper settings. The red reset LED (D19) must be off.

By default, the evaluation board is equipped with a MB91302A-020 and the external Flash is pre-programmed with the Softune Workbench monitor-debugger V1.02, 60MHz CPU clock (CLKB), 30MHz Peripheral clock (CLKP), 30MHz Bus clock (CLKT):

MonDebStarterkit91302_15MHz_CLKT30_V102

The monitor-debugger enables the user to debug his own application software located in the external SDRAM. After power-on the monitor debugger starts and finally D9, D10 are switched on to display that the debugger is working.



In case the monitor debugger detects any errors during start-up a status will be displayed on the 8 user LEDs. See chapter 6.2.3 “Advanced Debug Information” for more details.

3.1.3 Use as Emulation Target Board for Emulator MB2198-01

The “STARTERKITMB91302” can also be used as an emulator target board. In this case the user must unsolder the MCU MB91302A-020, U1 and solder a socket instead (NQPack144SE). The probe cable must be mounted to the socket, no MCU must be inserted in this case. Take care of pin 1 marking onto the socket and fix the probe cable with screws.

Do not use other probe cable than MB2198-101 only!

Connect the probe cable to the emulation pod. Check all DIP-switch-settings of the evaluation board and the emulation pod.

For the power on sequence the emulator system must be switched on first. Then switch on the evaluation board. Please look at the corresponding user manuals for the emulator how to set up the emulator system. After the power on the Reset-LED of the emulator must be off and the Vcc-LED (D22) must be on. If the Reset-LED is still on, check the Vcc voltage switch-settings of the emulator system and the power supply of the evaluation board.

For more details please check the following application notes for MB2198-01

MB2198-01 Installation Guide MB2198-01	AN-910026-11-MB2198INST.pdf
MB2198-01 Getting Start	AN-910027-10-MB2198Start.pdf
MB2198-01 Emulation and Debugging with Softune	AN-910028-10-MB2198emu.pdf

3.2 Software Installation

3.2.1 Installation of Softune Workbench

Fujitsu supplies a full working 32bit development environment with the “STARTERKITMB91302” called Softune Workbench V6. The Softune Workbench also supports a monitor debugger which is pre-programmed into the external Flash memory of the “STARTERKITMB91302”. To develop own software and to work with the Monitor Debugger of the “STARTERKITMB91302” the Softune Workbench development environment must be installed first. Follow the instructions for successful installation of the Softune Workbench.

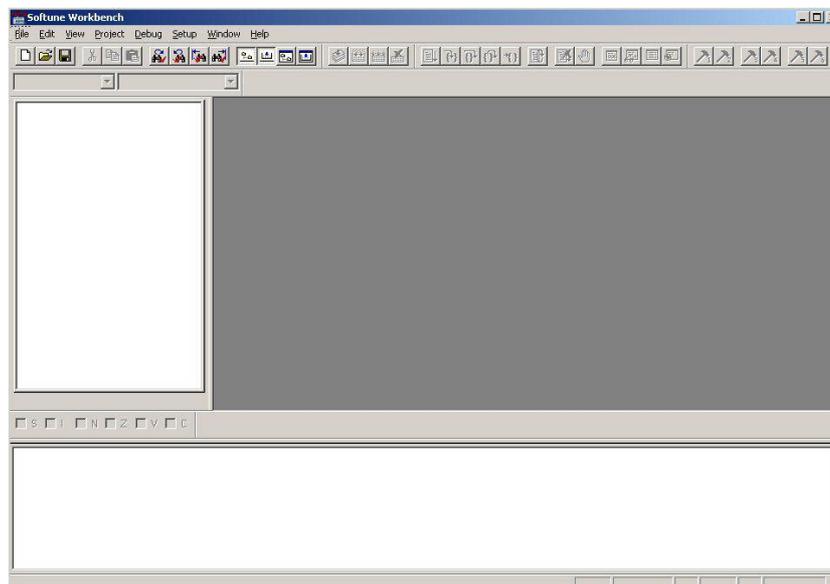
1. Before starting the installation setup ensure that you are logged in with **administrator or poweruser permissions**, otherwise the Softune installation will fail! Be aware that Softune does not support multi-user support. Therefore install- and user login must be the same.
2. Browse on the starterkit CD-ROM into the directory Software\SWBInstall and start the **setup.exe**

You can also start the installation using a browser by opening the index.html of the CD-ROM. Use the link “Software\Softune Workbench\” in the left side frame.

3. Follow the installation instructions
4. For the default installation path it is recommended to use c:\Softune32
5. After the installation is finished, Softune Workbench for FR can be started via the Windows “Start” menu



6. When Softune Workbench is started the following window will be shown

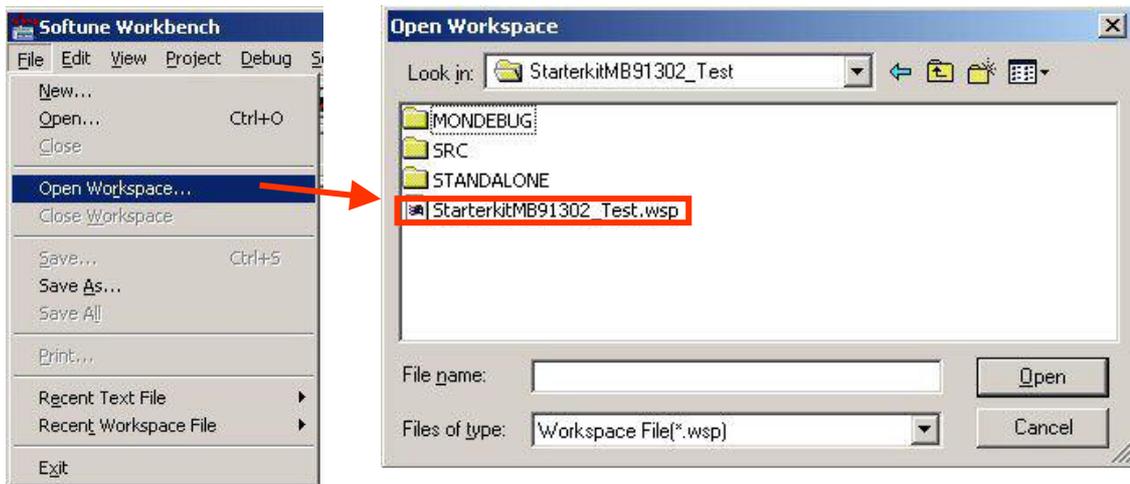


4 Quick Project Start-up / Test “STARTERKITMB91302”

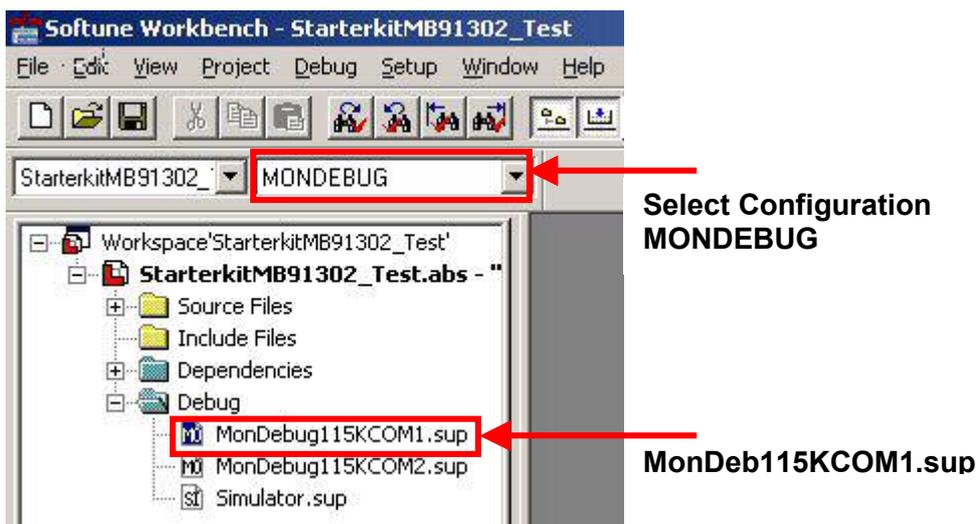
To open a project with Softune Workbench and to work with the monitor debugger, Softune Workbench must be started first. To open a project the following steps must be taken.

1. Ensure that no other windows application is using the COM1 port
2. Ensure correct jumper setting of **JP1 position 1-2** (MCU Operating Mode)
3. Ensure correct jumper setting of **JP15 position closed** (RTS/CTS shortcut)
4. Ensure that the “STARTERKITMB91302” is connected via **serial cable** (X5, UART0) with **COM1** of the PC.

Open the test-project using the <File>, <Open Workspace> dialog. Browse into the folder c:\..\software\examples\StarterkitMB91302_Test\ and select the file **StarterkitMB91302_Test.wsp**

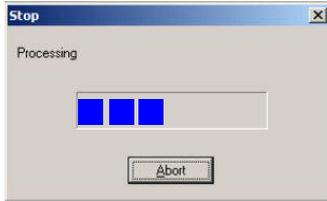


5. Ensure that the **configuration “MONDEBUGGER”** is selected! Start the Monitor Debugger by double click on MonDeb115KCOM1 in the following menu

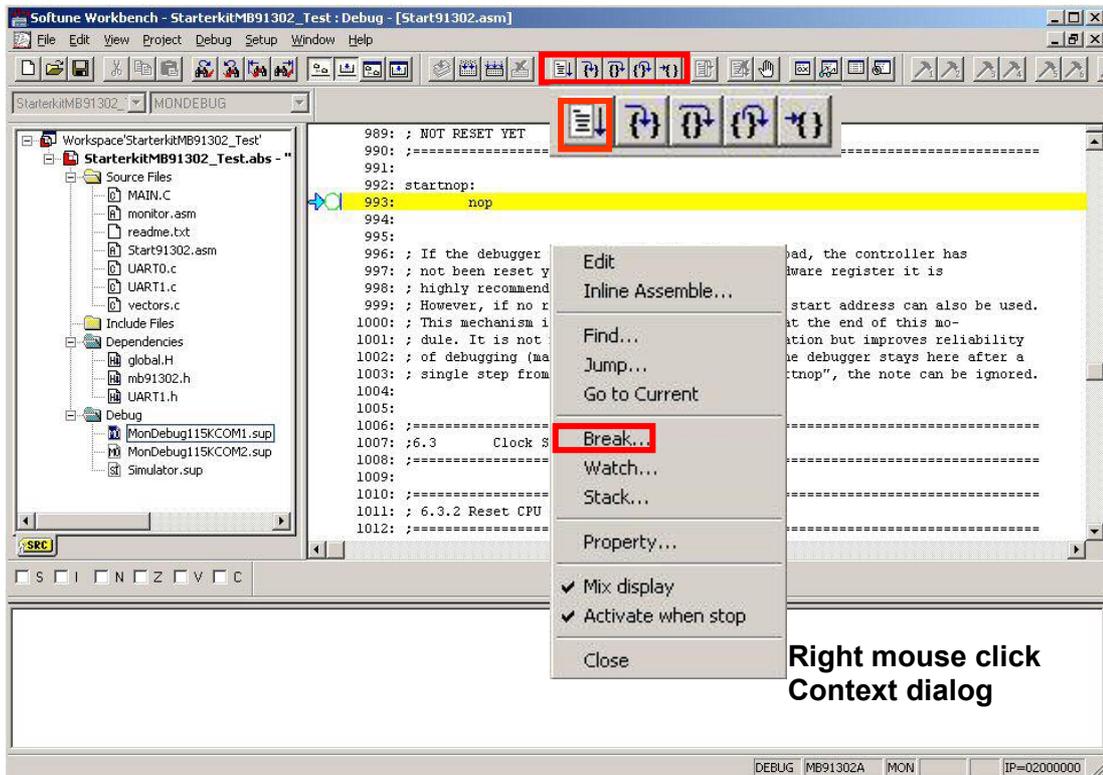


If a different COM port is used please use MonDebug115COM2.sup for COM2 or right-click on the “MonDeb115KCOM1” and select change. Follow the wizard to modify settings.

- Now the debugger starts and a download progress bar is shown indicating that the application program is downloaded to the "STARTERKITMB91302".



- After downloading the application successfully into the external SDRAM of the "STARTERKITMB91302" the following Softune Workbench screen is shown. The debugger offers now versatile features as "Run continuously, Step-In, Step-over, Step-out, .." which can be used via the Icon list. Using the right mouse button opens a context menu offering more settings as "Set Breakpoints, ...".



- Use the button "Run Continuously" to start the application and use the "NMI" button on the "STARTERKITMB91302" to Abort the Continuous program execution. Two times a running light should be shown now on the 8 user LEDs and if a terminal (115KB, 8N1) is connected to UART1 (X7) of the starterkit, a string is output corresponding to the pushed user button (INT0, INT3, ADTG, TIN1).

Note:

The command "Abort" to stop program execution is not supported by the Softune Workbench monitor debugger and might cause malfunction of the debugger if used. To "Abort" program execution use the button "NMI" on the "STARTERKITMB91302".

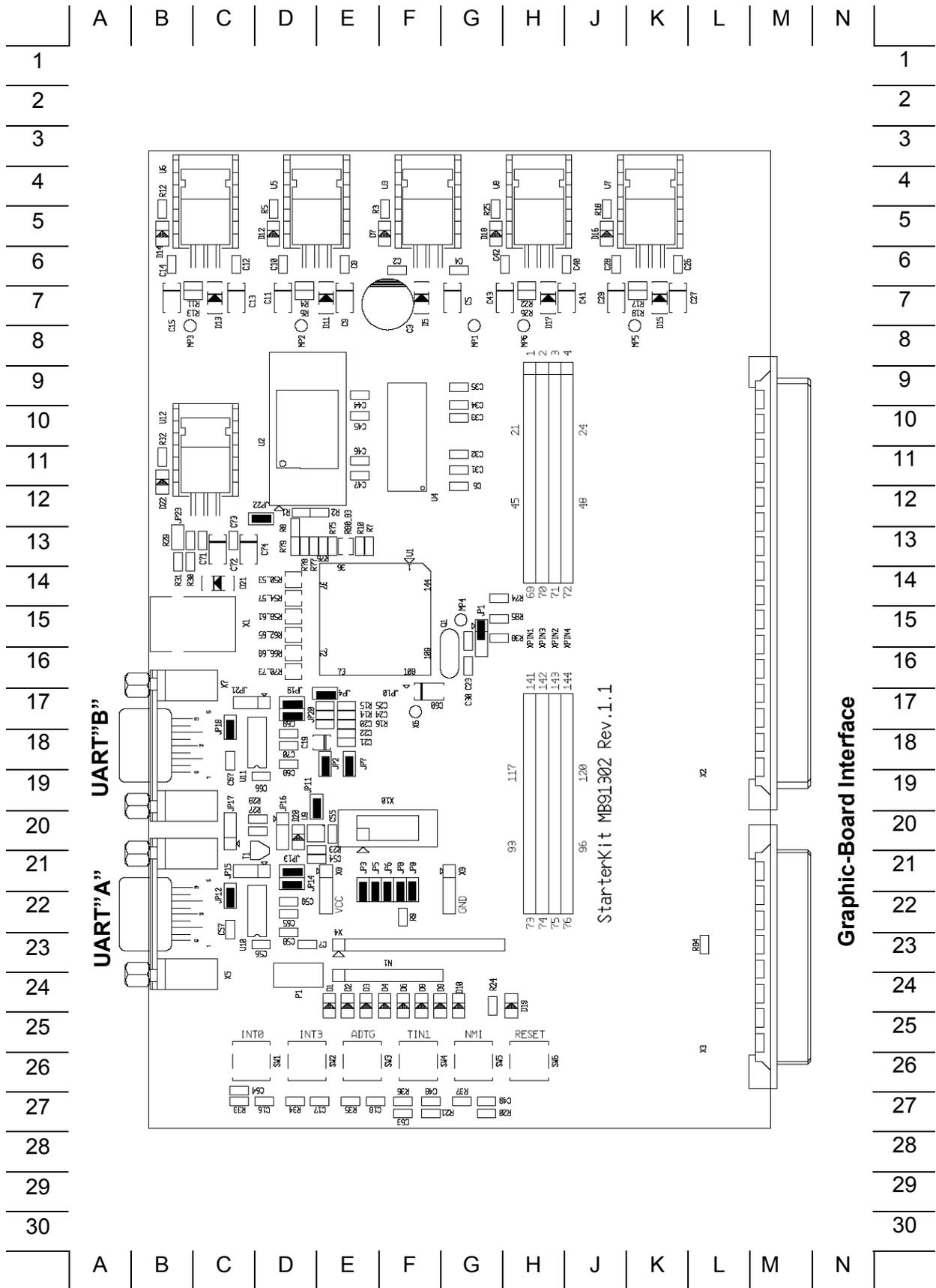
- For more details about Softune Workbench monitor debugger please check chapter 7.

5 Jumpers and Switches

This chapter describes all jumpers and switches, which can be modified, on the “STARTERKITMB91302”. The default setting is shown with a grey shaded area. All jumpers and switches are named directly on the board, so it is very easy to set the jumpers according to the features.

5.1 Jumper Overview

Jumper	Description / Function	Type	Default	Coordinates
JP1	Mode-selection, MD0	Jumper 3 pol	1-2	G 15
JP2	AVCC	Jumper 2 pol	Closed	D/E 18
JP3	SW INT0	Jumper 2 pol	Closed	E 21
JP4	AVRH	Jumper 3 pol	Closed	D/E 17
JP5	SW INT3	Jumper 2 pol	Closed	E/F 21
JP6	SW ADTG	Jumper 2 pol	Closed	F 21
JP7	AVSS	Jumper 2 pol	Closed	E 18
JP8	SW TIN1	Jumper 2 pol	Closed	F 21
JP9	SW NMI	Jumper 2 pol	Closed	F 21
JP11	Reset	Jumper 2 pol	Closed	D/E 19
JP12	UART A (RTS-CTS)	Jumper 2 pol	Closed	C 22
JP13	UARTA TxD (SOT0)	Jumper 2 pol	Closed	D 21
JP14	UARTA RxD (SIN0)	Jumper 2 pol	Closed	D 21
JP15	UARTA DTR/RTS	Jumper 2 pol	Open	C/D 21
JP16	DTR/DTR	Jumper 2 pol	Open	D 20
JP17	Reset by UARTA / UARTB	Jumper 2 pol	Open	C 20
JP18	UARTB (RTS-CTS)	Jumper 2 pol	Closed	C 17/18
JP19	UARTB TxD (SOT1)	Jumper 2 pol	Closed	D 17
JP20	UARTB RxD (SIN1)	Jumper 2 pol	Closed	D 17
JP21	UARTB DTR/RTS	Jumper 2 pol	Open	C/D 17
JP22	MCU_VCC	Jumper 2 pol	Closed	C/D 12



5.2 Operating-Mode (JP1)

The jumper JP1 is used to set the operating mode of the μC .

Ensure that the mode pin settings correspond to the operation-mode of the application.

For more detailed information please check the Hardware-Manual of the microcontroller.

Jumper	Setting	Description
JP1 (MD0)	1 - 2	MD0='1',
	2 - 3	MD0='0',

By default, the Single Chip Run-Mode is selected.

5.3 Power Supply Voltage (JP: 22)

JP22 This Jumper is used to connect the Vcc supply voltage to the μC .

Connecting an Ampere-meter allows measuring of the power-supply-current of the microcontroller (Icc).

Jumper	Setting	Description
JP22 (MCUVcc)	ON (Closed)	Power supply Vcc connected to μC
	OFF (Open)	Disconnected from Power supply Vcc

Default: JP22 = ON,

By default, the microcontroller is powered.

5.4 Analogue Power Supply Voltage (JP: 2, 4, 7)

The power supply as well as the positive reference voltage for the A/D-converter can be provided internally or externally.

JP2, JP7 connect power supply voltages (AVcc and AVss)

JP4 connect reference voltages (AVRH to AVcc)

Jumper	Setting	Description
JP2 (AVcc)	ON (Closed)	AVcc is connected to Vcc
	OFF (Open)	AVcc is disconnected from Vcc
JP4 (AVRH)	ON (Closed)	AVRH is connected to AVcc
	OFF (Open)	AVRH defined by resistor network ^{*1}
JP7 (AVss)	ON (Closed)	AVss is connected to GND
	OFF (Open)	AVss is disconnected from GND

^{*1}By default the resistor network (R15 and R16) is not mounted on the board

Default: JP2, JP4 and JP7 are closed

By default, the A/D-converter supply and reference voltage is +3V3.

Note:

If JP2 and JP7 are open, the user has to supply an adequate analogue voltage supply (AVcc and AVss) to the A/D-converter.

If JP4 is open, the resistors R15 and R16 define AVRH.

By default the resistor network (R15 and R16) is not mounted on the board.

5.5 UART"A" (JP: 12, 13, 14)

One RS232-transceiver can be connected to UART0.

JP13, JP14 connects UART0 to the RS232-transceiver (U10, X5)

JP12 Some terminal-tools may need a connection between CTS and RTS

Jumper	Setting	Description
JP13 (UART"A"TxD)	ON (Closed)	SOT0 is connected to RS232-Transceiver
	OFF (Open)	SOT0 is disconnected from RS232-Transceiver
JP14 (UART"A"RxD)	ON (Closed)	SIN0 is connected to RS232-Transceiver
	OFF (Open)	SIN0 is disconnected from RS232-Transceiver
JP12 (RTS-CTS)	ON (Closed)	RTS and CTS is shortcut on X5
	OFF (Open)	RTS and CTS is not shortcut on X5

Default: JP12=ON, JP13=ON, JP14 = ON

By default, UART0 is used as UART"A".

5.6 UART"B" (JP: 18, 19, 20)

One RS232-transceiver can be connected to UART1.

JP19, JP20 connects UART1 to the RS232-transceiver (U11, X7)

JP18 Some terminal-tools may need a connection between CTS and RTS

Jumper	Setting	Description
JP19 (UART"B"TxD)	ON (closed)	SOT1 is connected to RS232-Transceiver
	OFF (open)	SOT1 is disconnected from RS232-Transceiver
JP20 (UART"B"RxD)	ON (closed)	SIN1 is connected to RS232-Transceiver
	OFF (open)	SIN1 is disconnected from RS232-Transceiver
JP18 (RTS-CTS)	ON (closed)	RTS and CTS is shortcut on X7
	OFF (open)	RTS and CTS is not shortcut on X7

Default: JP18=ON, JP19=ON, JP20 = ON

By default, UART1 is used as UART"B".

5.7 Reset-Generation (JP: 11, 15, 16, 17, 21)

Additional to the internal Power-On-Reset the microcontroller can be reset by an external Reset-circuit (Voltage-Monitor) and by the UARTs, too.

- JP15, JP21** Either the DTR-line or the RTS-Line of UART" A" or UART" B" can be used to generate a system-reset.
- JP17** This jumper selects whether the DTR/RTS line from UART" A" or UART" B" will generate a system-reset.
- JP11** open this jumper if no external Reset shall be generated. In this case only the internal reset is active (e.g.: power-on)
- JP16** The polarity of the DTR/RTS line can be invert by this jumper. Remove the jumper in order to disable the reset logic.

Jumper	Setting	Description
JP15 (DTR / RTS "A")	1-2	DTR of UART" A" is selected
	2-3	RTS of UART" A" is selected
JP17 (UART" A"/" B")	1-2	UART" A" is used to generate Reset
	2-3	UART" B" is used to generate Reset
JP21 (DTR / RTS "B")	1-2	DTR of UART" B" is selected
	2-3	RTS of UART" B" is selected
JP11 (Main Reset)	Closed	External Reset generation is active
	Open	No external Reset generation
JP16 (Polarity)	1-2	No polarity inversion for the DTR/RTS signal
	2-3	Polarity inversion for the DTR/RTS signal

Default: JP11 = closed, JP15, JP16, JP17 and JP21 are not set

By default, the external Reset generation is active. The Reset by UART is disabled.

Note:

While a reset signal is asserted the red Reset-LED D19 is lit. During normal operation, this LED should be off! If JP16 (Polarity) is set, than JP15 and JP17 or JP21 have to be set, too. If the reset LED is steadily on, check the power supply input voltage and the settings for the reset-generation by UART.

5.8 Buttons INT0, INT3, ADTG, TIN1, NMI(JP: 3, 5, 6, 8, 9)

JP3, JP5: Two push buttons can be used to trigger the ext. interrupts INT0 and INT3

JP6: One button can be used as trigger-input for the A/D-converter (ADTG)

JP8: One button can be used as trigger-input for the Reload-Timer1 (TIN1)

JP9: One Button can be used for the non-maskable interrupt. (NMI)

Jumper	Setting	Description
JP3 (INT0)	Closed	INT0 is connected to Push-button "INT0"
	Open	No connection to INT0
JP5 (INT3)	Closed	INT3 is connected to Push-button "INT3"
	Open	No connection to INT3
JP6 (ADTG)	Closed	ADTG is connected to Push-button "ADTG"
	Open	No connection to ADTG
JP8 (TIN1)	Closed	TIN1 is connected to Push-button "TIN1"
	Open	No connection to TIN1
JP9 (NMI)	Closed	NMI is connected to Push-button "NMI"
	Open	No connection to NMI

Default: JP3, JP5, JP6, JP8, JP9 = Closed

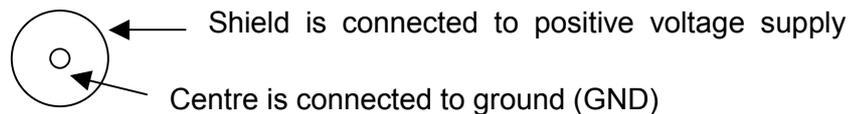
By default, INT0, INT3, ADTG, TIN1 and NMI of the microcontroller are connected to the Push buttons.

6 Connectors

6.1 Power connector (X1)

The following figure shows the power connection jack X1. This connector is used to connect an external unregulated DC power supply voltage (6V-9V DC) to the evaluation board.

Connector X1:



It is recommended to use 6V in order to keep the power dissipation to a minimum. Otherwise, an additional heat sink for the linear voltage regulator might be necessary.

NOTE:

If an optional graphic board will be connected, then the power supply has to be 8V-9V DC.

6.2 Edge connector (XPIN1, XPIN2, XPIN3, XPIN4)

All pins of the microcontroller are directly connected to XPIN1 to XPIN4 as follows:

Connector	MCU Pins
XPIN1	1, 5, 9, ..., 141
XPIN2	3, 7, 11, ..., 143
XPIN3	2, 6, 10, ..., 142
XPIN4	4, 8, 12, ..., 144

On the PCB, the corresponding pin numbers of the μ C are written next to the connector pins.

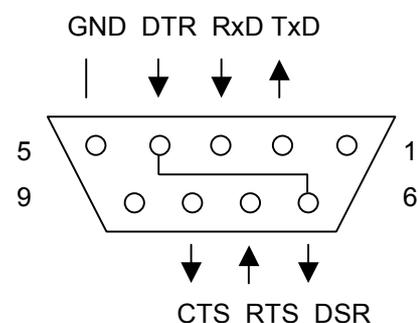
6.3 UART"A", UART"B" connector (X5, X7)

Two 9-pin D-Sub female connectors are used for the serial interface UART"A" and UART"B".

TXD is the transmit output, RXD is the receive input.

The DTR or RTS signal can be used to generate a reset.

Please use 1:1 cable for PC-connection.



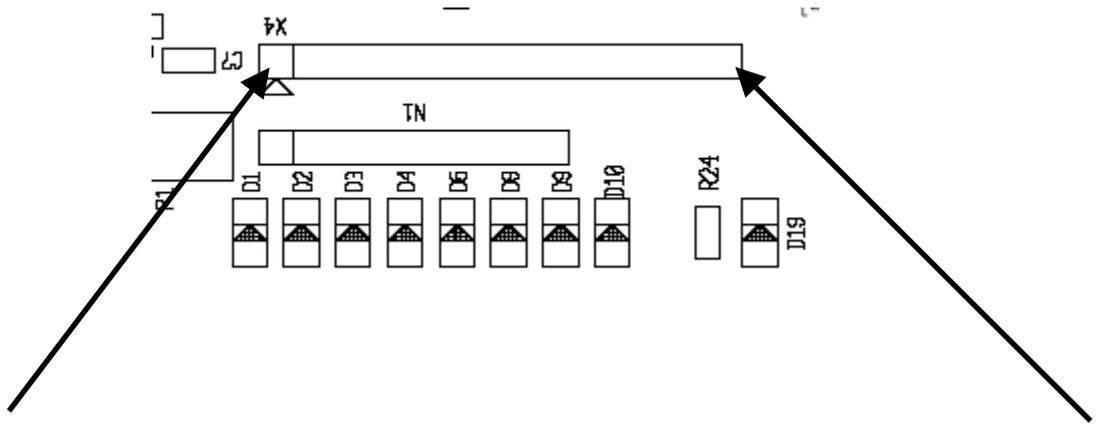
Connectors X5, X7:

6.4 USER-LEDs & LC-Display (optional)

Eight LEDs are reserved for user-application. In order to disconnect the LEDs from the related microcontroller port, the resistor network N1 can be removed.

Instead of the user-LEDs one alphanumeric LC-Display (optional) can be connected.

The following control-signals are reserved:



	1	2	3	4	5	6	7	8	9	10	11	12	13	14
LCD	GND	VCC	V0	RS	R/W	E	-	-	-	-	D4	D5	D6	D7
LED				D1	D2	D3	D4				D6	D8	D9	D10
MCU				81	82	83	84				85	86	87	88
Port				PG0	PG1	PG2	PG3				PG4	PG5	PG6	PG7

6.5 Graphic Board Interface

Pin	Name	91302	Pin	Name	91302	Pin	Name	91302
A1	2.5 V		B1	2.5 V		C1	2.5 V	
A2	3.3V		B2	3.3V		C2	3.3V	
A3	5.0 V		B3	5.0 V		C3	5.0 V	
A4	GND		B4	GND		C4	GND	
A5	D0	132	B5	D1	133	C5	D2	134
A6	D3	135	B6	D4	136	C6	D5	137
A7	D6	138	B7	D7	139	C7	D8	142
A8	D9	143	B8	D10	144	C8	D11	1
A9	D12	2	B9	D13	3	C9	D14	4
A10	D15	5	B10	D16	8	C10	D17	9
A11	D18	10	B11	D19	11	C11	D20	12
A12	D21	13	B12	D22	14	C12	D23	15
A13	D24	18	B13	D25	19	C13	D26	20
A14	D27	21	B14	D28	22	C14	D29	23
A15	D30	24	B15	D31	25	C15	GND	
A16	GND		B16	GND		C16	GND	
A17	A0	45	B17	A1	46	C17	A2	47
A18	A3	48	B18	A4	49	C18	A5	50
A19	A6	51	B19	A7	52	C19	A8	55
A20	A9	56	B20	A10	57	C20	A11	58
A21	A12	59	B21	A13	60	C21	A14	61
A22	A15	62	B22	A16	64	C22	A17	65
A23	A18	66	B23	A19	67	C23	A20	68
A24	A21		B24	A22		C24	A23	
A25	A24		B25	A25		C25	A26	
A26	A27		B26	A28		C26	A29	
A27	A30		B27	A31		C27	GND	
A28	S0 (PG0)	81	B28	S1 (PG1)	82	C28 (PG2)	S2	83
A29	S3 (PG3)	84	B29	S4 (PG4)	85	C29 (PG5)	S5	86
A30	S6 (PG6)	87	B30	S7 (PG7)	88	C30	GND	
A31	2.5V		B31	2.5V		C31	NC	
A32	GND		B32	GND		C32	GND	

Pinning of connector X2

Pin	Name	91302	Pin	Name	91302	Pin	Name	91302
A1	CS0		B1			C1	CS1	123
A2	CS2	124	B2			C2	CS3	125
A3	CS4	126	B3			C3	CS5	127
A4	CS6		B4			C4	RDY	28
A5	BGRNT	29	B5			C5	BRQ	30
A6	RDX	31	B6			C6	WR0	32
A7	WR1	33	B7			C7	WR2	34
A8	WR3	35	B8			C8	AS	40
A9	ALE		B9			C9	CLK	38
A10	DREQ0	103	B10			C10	DACK0	104
A11	DEOP0	105	B11			C11	NC	
A12	IRQ A	83	B12			C12	I ² C SDA	107
A13	IRQ B	84	B13			C13	I ² C SCL	106
A14	Uart RX		B14			C14	Uart TX	
A15	GND		B15			C15	Uart Clk	
A16	RESET	119	B16			C16	NC	

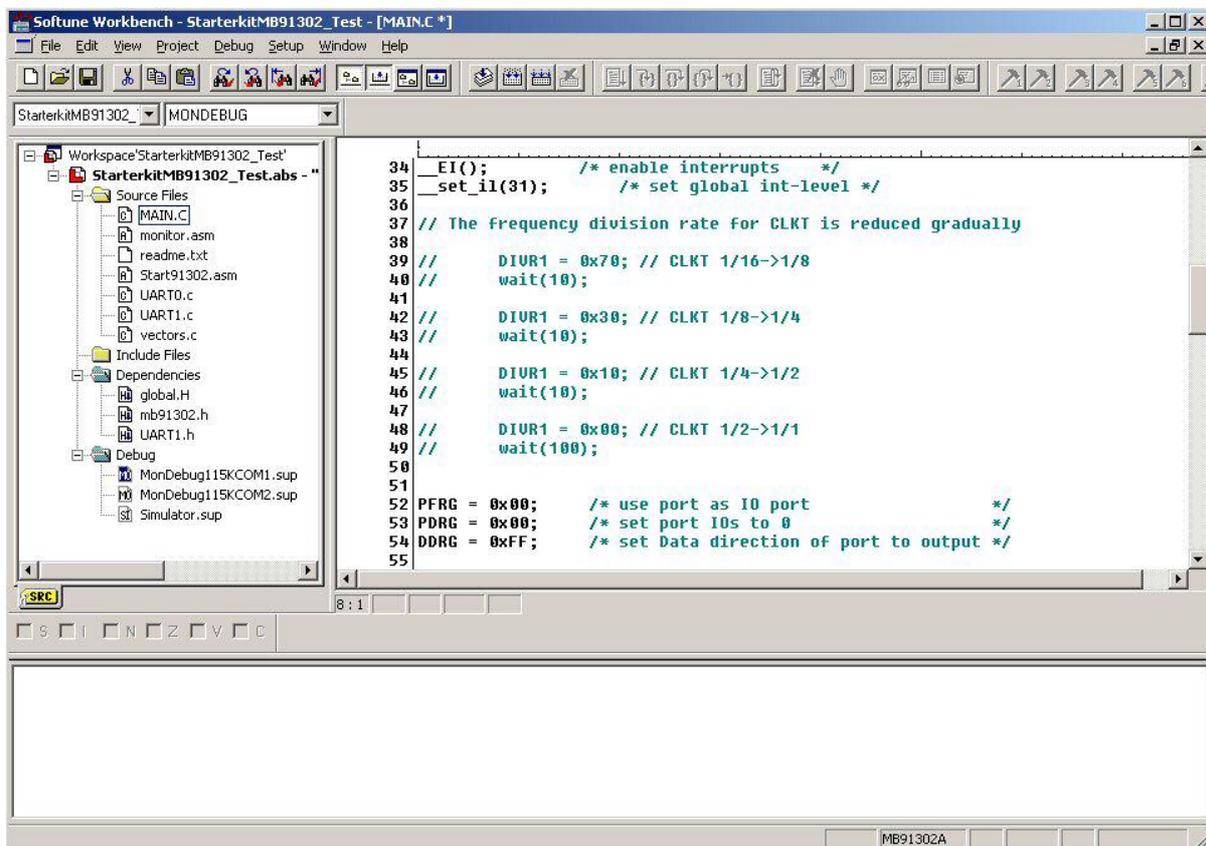
Pinning of connector X2

7 Getting Started

Make sure Softune Workbench for FR is installed on your PC and the “STARTERKITMB91302” is ready to use (power-on (X1), serial connection (X5) to COM1 of the PC). Refer to Chapter 3 for details on the hardware and software installation.

7.1 Introduction to Softune Workbench

Start Softune Workbench. Select <File> <Open Workspace> to load the first example “StarterkitMB91302_test.wsp”. All examples can be found in the folder \software\examples on the starterkit CD-ROM. It is recommended to copy this folder to the local hard-drive. In the project tree on the left side, you can open the “source”-folder which contains the source-files registered to this project. A double-click on one of the files will invoke the built-in editor, which supports syntax-highlighting, tags and various other functions.



Softune Workbench main window

Note:

You may customize the editor by a right-click on the editor window.

Whenever you modify the source-files, you have to re-compile and link the related files to produce a valid loadmodule (ABS-file). This is done using the MAKE-function. MAKE invokes the assembler, C-compiler and linker for FR whenever necessary (only modified files will be re-compiled).

If you wish to re-compile the entire project regardless of any changes, you can use the BUILD-function. To check for syntax-errors on a specific source-file, use the COMPILE/ASSEMBLE function.

These three functions are available on the button-bar or from the main menu (Project – Compile / Make / Build / Abort).



Click on MAKE or BUILD. Messages from the individual language tools will be fed into the output window at the bottom of the main screen. If the tool chain (C-compiler → Assembler → Linker) was completed successfully, the message “No Error” will appear.

If you get any errors during compilation, an appropriate message will be generated. Try this with a simple syntax-error (e.g. delete a semicolon “;” from the end of a C-line) and click on MAKE again. You will now see a message like this:

```

Now Making...
-----Configuration: Starterkit91302_Test.prj - Debug-----
main.c
*** C:\StarterkitMB91302\MonDebStarterkit91302_Test\Src\main.c(81) E4065B: expected a ";"
*** C:\StarterkitMB91302\MonDebStarterkit91302_Test\Src\main.c(104) W1012B: warning: parsing
restarts here after previous syntax error
1 error detected in the compilation of
"C:\StarterkitMB91302\MonDebStarterkit91302_Test\Src\main.c".
-----
Error detected.
-----

```

To locate the position in the source-file, where the error has occurred, double-click on the message. The editor will open the appropriate source-file, indicating the error highlighted in red, depending on the customise settings of the editor. Correct the error and re-compile the project as explained above.

If more errors occurred, you can go through the error list step by step using the menu “Edit – Top/Previous/Next/Bottom Error” or using the appropriate buttons which have the same functions:



Note:

To get on-line help about a specific error, select the error message and press F1. In many cases, you can get some useful hints how to solve the problem. Of course, you can also use the HELP-menu anytime during development or debugging.

7.2 Project Start-up

In order to allow a quick and smooth project start-up Fujitsu Microelectronics Europe supplies a template project as a reference. The easiest way to start a new project is to make a copy of the template project and use this copy as a start-up. The template includes the latest start91302.asm file, MCU headerfile, IRQ table, basic linker and C-Compiler settings.

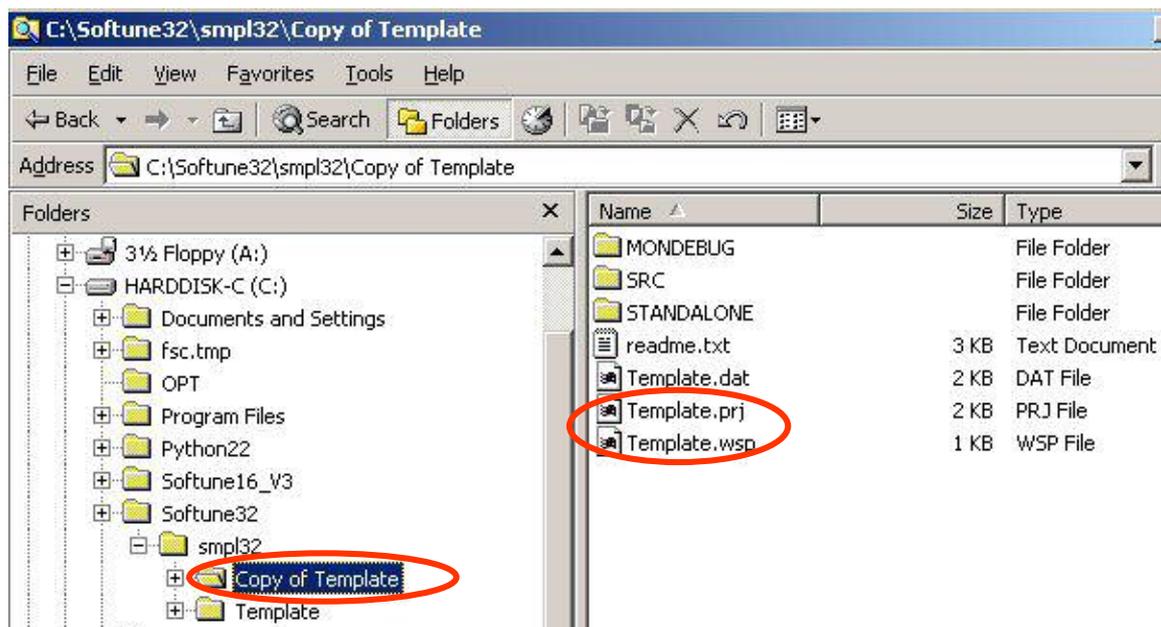
Note:

In any case the settings done in the template must be checked and have to be adjusted to the specific needs and settings of the final application. The template is providing an example for building up a new project.

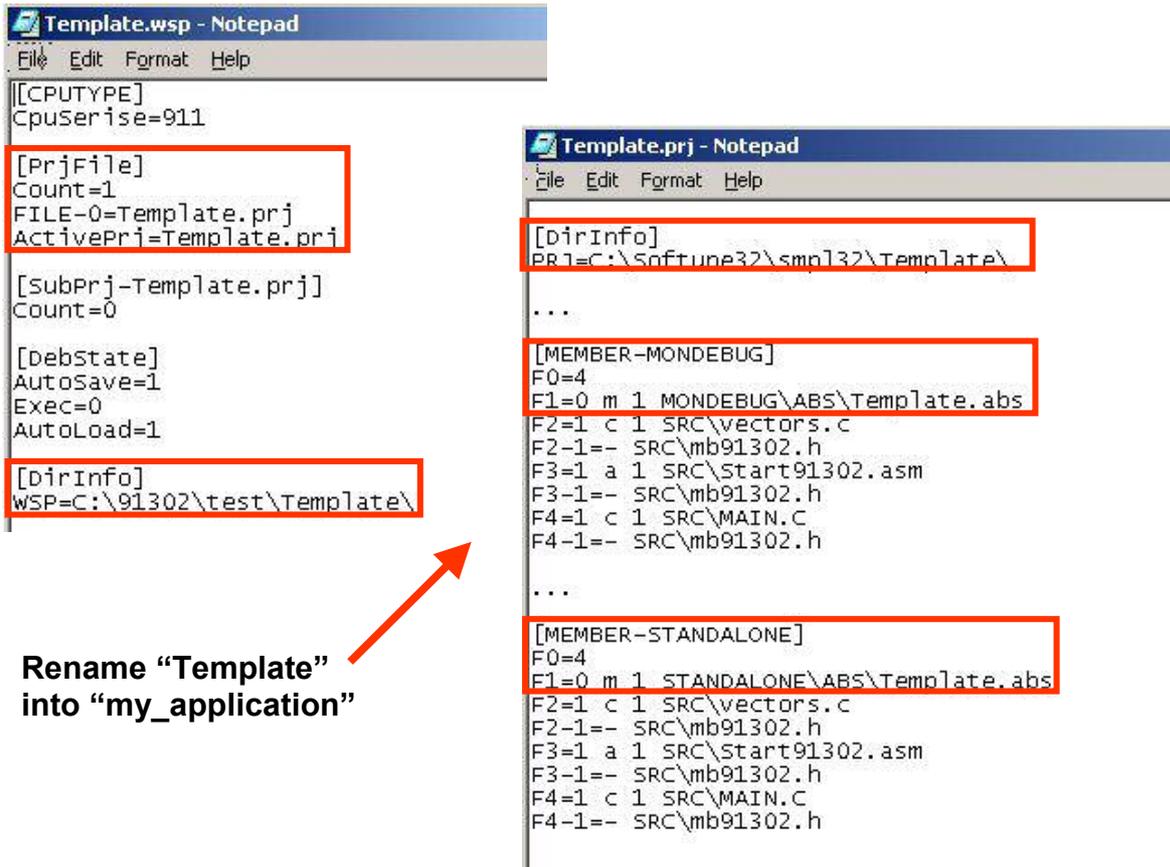
THIS SAMPLE CODE IS PROVIDED AS IS AND IS SUBJECT TO ALTERATIONS. FUJITSU MICROELECTRONICS ACCEPTS NO RESPONSIBILITY OR LIABILITY FOR ANY ERRORS OR/ELIGIBILITY FOR ANY PURPOSES. (C) Fujitsu Microelectronics Europe GmbH

Create a new project:

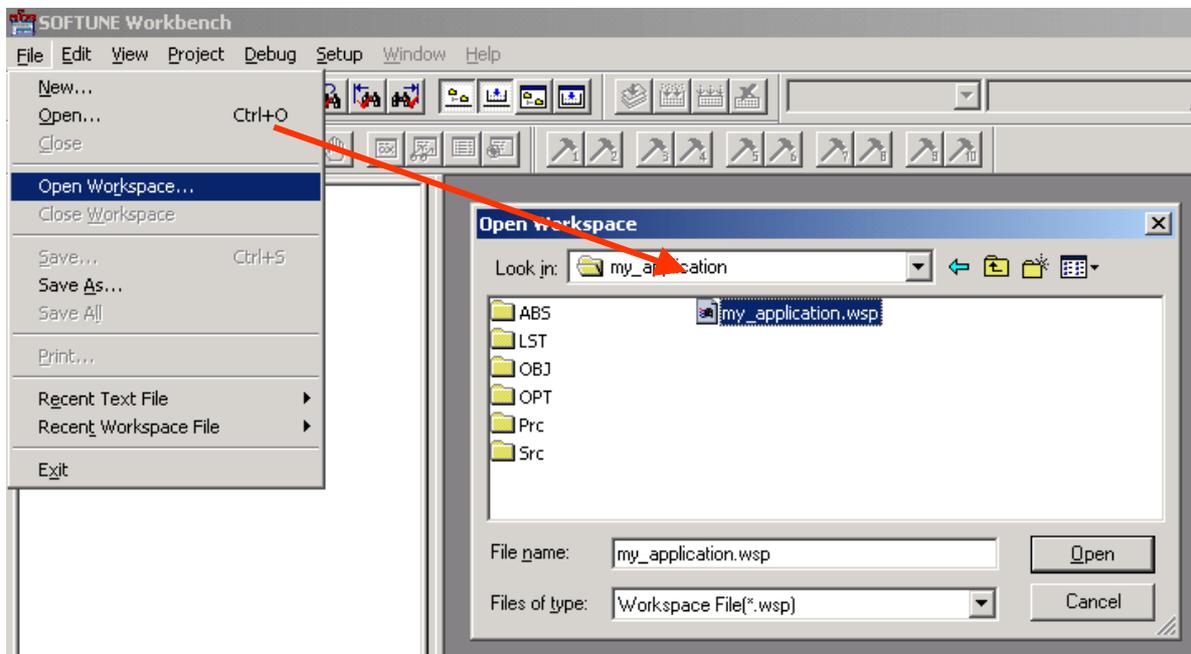
1. In order to start a new user-project use the “**Template**” project
2. **Copy** the folder **Template** and **rename** the folder to e.g. my_application



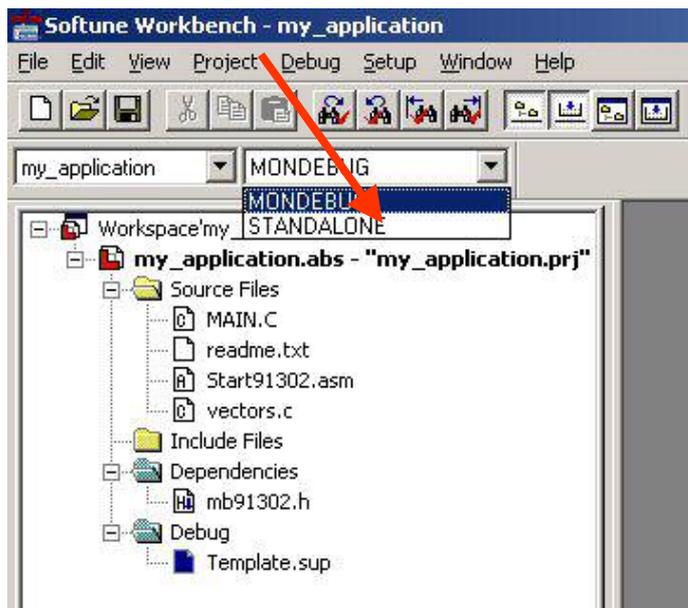
3. **Enter the folder “my_application”**
Rename “template.prj” into “my_application.prj”
Rename “template.wsp” into “my_application.wsp”
4. **Edit “my_application.prj”**
Rename “template” -> “my_application”
5. **Edit “my_application.wsp”**
Rename “template” -> “my_application”



6. Start Softune Workbench and open your project



7. Select Active Configuration "MONDEBUG"



Use Active Configuration MONDEBUG

With this selection the project is linked to be used with the Softune Workbench monitor debugger. Corresponding linker settings are done for the memory map in order to avoid overwriting of the sections which are used by the monitor debugger. The code is linked in this configuration for external SDRAM, otherwise it is not possible to debug code with the Softune Workbench monitor debugger. Additionally the MACRO SWBMonDeb is switched on in the assembler settings. This affects settings in start91302.asm which are mandatory to work with the Softune Workbench monitor debugger.

Note:

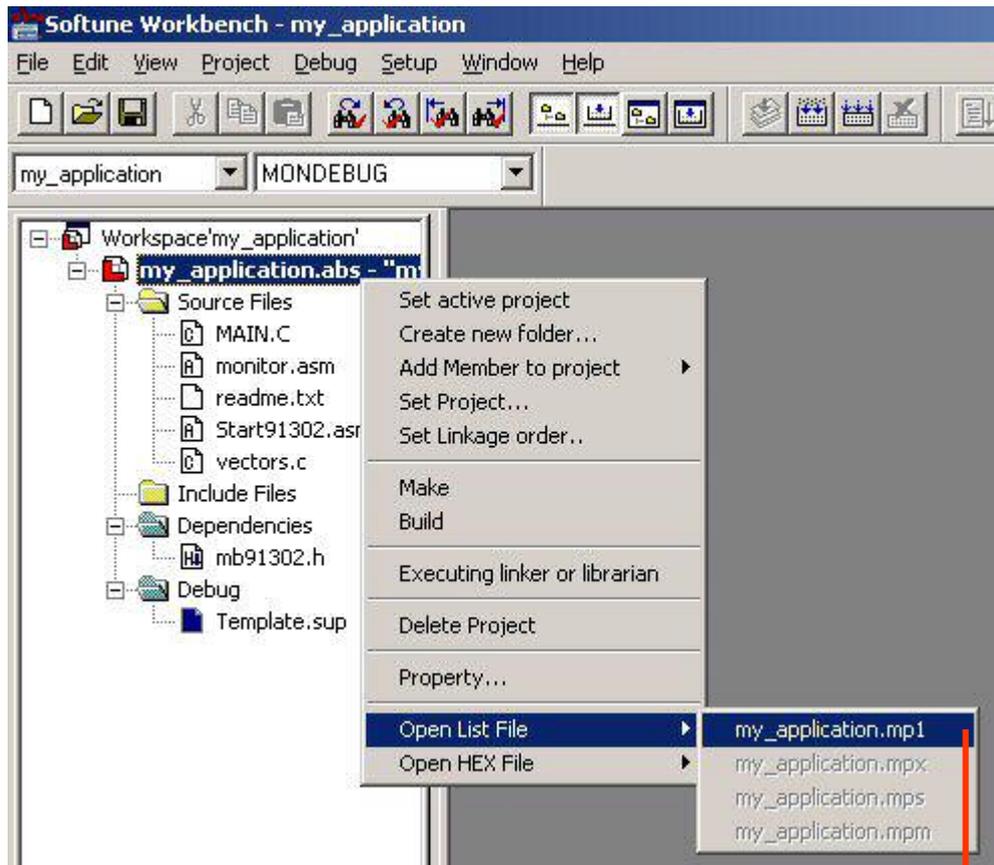
Always check the memory map of the linker settings and ensure that this memory map is suitable for the application and target system in use. Additionally the .mp1 file should be checked to ensure correct settings.

8. Check MCU settings in the module **start91302.asm**. Be careful regarding any modifications. If the peripheral clock speed is modified it can happen that the communication between Softune Workbench and monitor debugger is lost. In the module start91302 some basic MCU settings are done and some initial data copy transfers are performed for specific data sections e.g INIT. Check if the macro "SWBMonDeb" is enabled (default) in the Assembler options (<Project>, <Setup Project>, <Assembler>, category <Define Macro>).
9. Check module **vectors.c** if any interrupts are used for proper vector table set-up.
10. Write application code in the module **main.c** or add any other module to the project.
11. Write and modify source code and settings corresponding to the needs of the application. Finally use **"Make"**, or **"Build"** to compile and link the project. The generated output files can be found in the sub-folder MONDEBUGGER of the project folder. The *.mhx file (Motorola S-Record) is located in the ABS sub-folder.

Note:

Always check the memory map of the linker settings and ensure that this memory map is suitable for the application and target system in use. Additionally the *.mp1 file should be checked to ensure correct settings. Do

not remove the monitor.asm file for the configuration MONDEBUB because it reserves the address areas used by the monitor debugger.



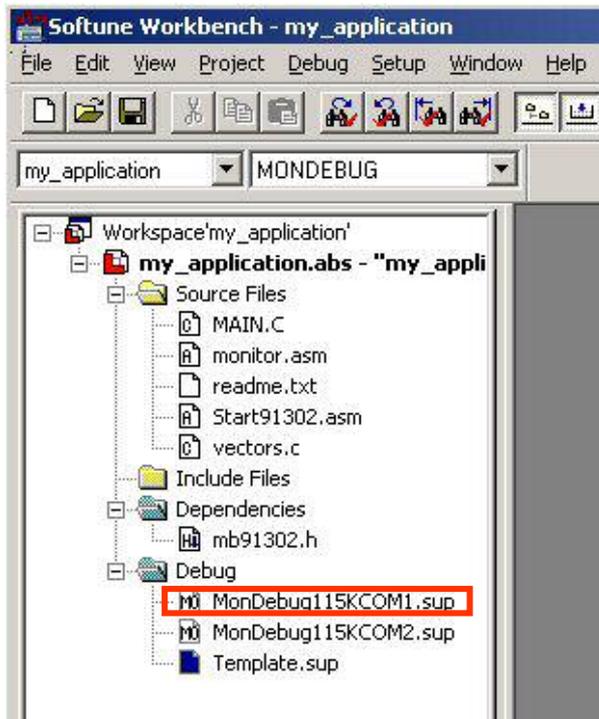
Check linker mapping list in the*.mp1 file by right-click on “my_application.abs”, Open List File, *.mp1.

```

45 ■FR Family Softune Linker      Mapping List
46
47 S_Addr. -E_Addr.   Size      Section Type A1      Sec.(Top 81)
48 0003F000-0003F003 00000004  CODE  N R-XI 00 ABS  MONDEBDATA
49 0003F800-0003FBFB 000003FC  STACK P RW-- 04 REL  SSTACK
50 0003F800-..... 00000000  DATA P RW-- 04 REL  DATA
51 0003F800-..... 00000000  DATA P RW-- 04 REL  INIT
52 0003FBFC-0003FBFD 00000002  STACK P RW-- 04 REL  USTACK
53 00080000-00080003 00000004  CODE  N R-XI 00 ABS  TESTPATTERN
54 000FC000-000FC003 00000004  CODE  N R-XI 00 ABS  MONDEBCODE
55 02000000-020000E9 000000EA  CODE  P R-XI 04 REL  START
56 020000EA-0200025F 00000176  CODE  P R-XI 02 REL  CODE
57 02000260-..... 00000000  CONST P R--I 04 REL  CONST
58 02000260-..... 00000000  CONST P R--I 04 REL  EXT_CTOR_DTOR
59 02000260-..... 00000000  DATA P R--- 04 REL  #INIT
60 020FFC00-020FFFFF 00000400  CONST P R--I 04 REL  INTVECT
61 ■FR Family Softune Linker      Symbol List
62

```

12. Start Softune Monitor Debugger via the “Debug” Command



13. After the project is built and debugged successfully switch to the active configuration **STANDALONE**

Use Active configuration **STANDALONE**

14. With this selection the project is linked to work in external flash memory. Check that the macro “SWBMonDeb” is switched off (default) in the Assembler options (<Project>, <Setup Project>, <Assembler>, category <Define Macro>).

This configuration will not work with the Softune Workbench monitor debugger In this configuration the code is linked to work in external Flash memory, only.

15. Use “**Make**”, or “**Build**” to compile and link the project for external Flash memory. The generated output files can be found in the sub-folder STANDALONE of the project folder. The *.mhx file (Motorola S-Record) is located in the ABS sub-folder.

Note:

Always check the memory map of the linker settings and ensure that this memory map is suitable for the application and target system in use. Additionally the *.mp1 file should be checked to ensure correct settings.

16. Program the generated *.mhx file (Motorola S-Record) with the Flash programming utility into the external Flash. See Chapter 9 “Programming the external Flash Memory” for more details.

Note:

It is not possible to use the Softune Workbench monitor debugger with this configuration because no code can be debugged which is located in external Flash. Also this configuration does not support special debugger settings, which are disabled for this configuration. The file monitor .asm can be removed from the link list to free address areas which are normally used by the monitor debugger.

7.3 Softune Workbench Monitor Debugger

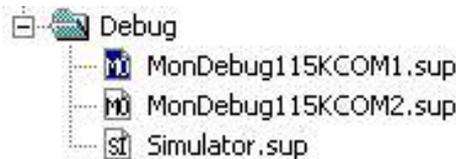
7.3.1 General Description

Whenever you have created a valid load module successfully, you can switch from the development mode to the debugging mode of Softune Workbench.

Basically, there are 3 types of debugging systems supported :

1. The software simulator: This type of debugger is always present and does not require any special hardware extensions. The simulator will cover the FR-core features, but no peripheral functions. Therefore, you can use the simulator to verify program flow, check for dynamic errors, look at the generated assembler code , etc. .
2. The monitor debugger: This debugger type requires an evaluation board like the “STARTERKITMB91302” connected to one of the COM-ports of your PC. Therefore, make sure the evaluation board connected and powered-up as described before. Explanations in this manual refer to the monitor debugger only.
3. The emulator debugger: The in-circuit-emulator (ICE) is a system, which allows a connection to any target system using a probe-cable. The appropriate system for the MB91302A series is the MB2198-01 system. More information about this system can be found on Fujitsus website.

Which debugger is used for the actual project can be configured in the project tree as shown below or in the “Project – Setup Project – Debug” menu. By default (blue marked icon), the monitor debugger is selected for MB91302A-020 project examples :



The default settings for the monitor debugger are : **COM-Port 1, 115Kbaud, 8N1**. If you wish to change the COM-port, please right-click on the “MonDeb115KCOM1” entry and select change.

Note:

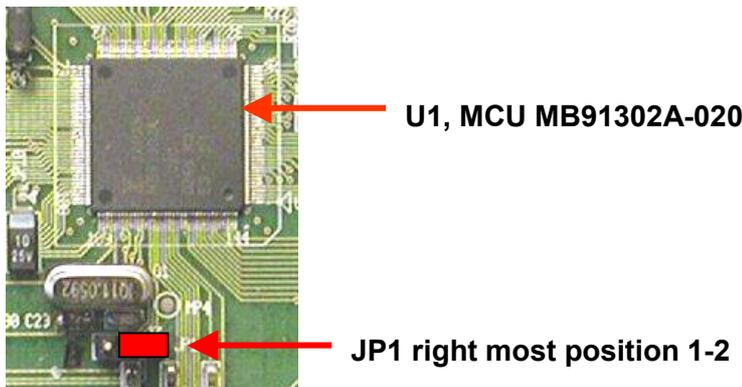
If you change the baudrate or use a different crystal frequency , you have to change the UART-settings in the debugging kernel source project, re-compile and flash the new kernel to your “STARTERKITMB91302” first. The project can be found in Software\SWBMonitorDebugger. All settings are done in the module u_io.asm (see chapter 6.6.3 for more information).

If these settings are correct, start your debugging session :

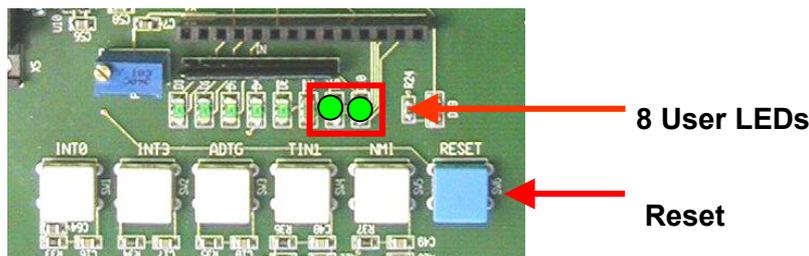
Make sure you have connected the “STARTERKITMB91302” and your monitor-kernel is stored in the external flash memory of the “STARTERKITMB91302”. Then power up and reset the starterkit. Check the following configurations:

1. Correct CPU Mode selection (JP1 1-2)

After Power on the Monitor debugger indicates via the 8 user-LEDs the current status. The following status indications exist. Ensure that the CPU is working in normal RUN Mode (Jumper **JP1** right most position 1-2)



2. Correct start-up of the monitor debugger



In case of any other status indication push the reset button or check chapter 6.6.1 for detailed information.

Start the debugger by double clicking the “MonDeb115KCOM1” entry in the project/debug tree - or by using “Debug – Start Debug” from the menu. You should see a progress bar indicating the download process.

If no process is indicated an error message will occur (“e.g. Invalid communication status”). Double-check the settings again as explained above. Also make sure the evaluation-board is powered on, ready to use and the RS232-cable is connected properly. Also see the troubleshooting section in the appendix.

After starting the debugger, the source window will be opened which contains locate-arrows for each (possible) source-line, the original source-code,

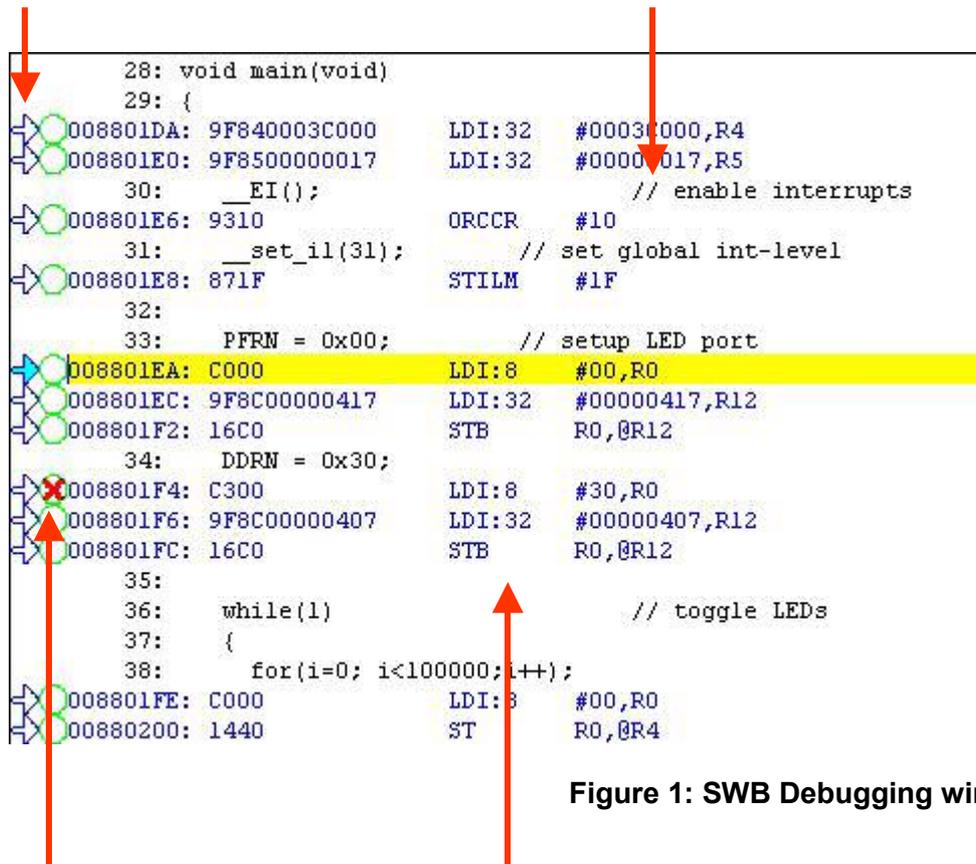


Figure 1: SWB Debugging window

Breakpoint-indicators and the assembler code (in “mixed view”). The current program position will be indicated by a yellow line. Use right-click in the source window to configure the view (add/remove control characters, View Mixed source code, Breakpoint ...)

Note:

You can choose “mixed view” from the context menu (right-click) to display source and assembly code at the same time. The command “go to current” will take you to the active code line!

7.3.2 Basic Debugger Features



GO: Executes the program continuously from the memory location pointed to by the current program counter (PC) until a termination factor occurs.



STEP IN: Executes the program stepwise according to the listing in the source window (steps in C- or ASM). Note that interrupts are disabled during step execution !



STEP OVER: Executes the program stepwise except call-instructions (which will be executed until return). Interrupts are not disabled during a continuous CALL...RETURN execution.



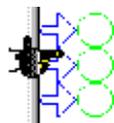
STEP OUT: Executes the program continuously to the parent function (until RETURN). Interrupts are enabled. Note, that debug information is required for this function.



EXECUTE UNTIL CURSOR: Automatically sets a breakpoint at the actual cursor position and executes continuously until this breakpoint, which will be deleted afterwards.

Each valid code line in a debugger window automatically has a locator (blue arrow) and a breakpoint-option (circle). Note that some C-lines may not be displayed with locators and breakpoints, because the compiler has created “optimized” assembler-code. Select “mixed view” in order to check the compiler output.

Instead of single-steps, you may also use the arrows to directly execute your program until a certain line of your source-code :





ABORT: Forcibly terminates execution. This button is not fully supported by the monitor debugger and may cause malfunction if used to abort “continuous operation” of the MCU.

This command button can only be used to abort single code line operations

Example: `for(k=DELAY_CONSTANT; k>0; k--);`

Use the **button NMI on the “STARTERKITMB91302”** for **ABORT** function.



To **ABORT** continuous execution on the “STARTERKITMB91302”, you have to use the **NMI button** on the “STARTERKITMB91302”.

7.4 Advanced Monitor Debugger Features



TOGGLE BREAKPOINT: Sets or deletes breakpoint at the current source line:

To set or delete a breakpoint, click the circles at the beginning of a source-line.

A  indicates an active breakpoint. Hit “Run Continuously”  to execute code until reaching this line. A list of all breakpoints can be found under the “Debug – Breakpoint” menu. 255 Software-Breakpoints (using TRAP replacement) are possible.

Note:

To set breakpoints at positions which are currently not visible (e.g. because the source window of that module is not open), you can also enter a symbolic label directly in the “Breakpoint” menu. Example: Enter “main” in the address-field and confirm. The new breakpoint will automatically be assigned to the address of the “main()”-function.



REGISTER WINDOW: Displays the CPU-register window. Updated registers appear in red. *Setup* in context menu defines which Registers should be displayed.



WATCH WINDOW: Displays the current variables to „watch“. Double-click on any variable in your code then specify *watch* in context menu to add to watch window. All listed variables in a watch window can be displayed in any number format. Use *Edit* to directly change the contents.



MEMORY WINDOW: Displays memory areas in various formats defined by *Setup* (context menu). Changing of address/data is possible when debugger is not executing.

Note:

If you modify the ROM-area (200.0000'H to 27F.FFFF'H) it can affect the currently loaded program !

The memory window can be very helpful to check the registers or the frame memory of the connected graphic controller !



DISASSEMBLER: Disassembles the content of the code memory beginning from actual PC position and displays the result in a separate window. Individual assembler-lines can be changed using the „inline assemble“-function.

DEBUG - STACK: Displays the current stack contents in terms of function calls. Parameters, if any, are displayed in brackets.

DEBUG - VECTOR: Reads out and displays the actual interrupt-vector table. Use the „jump“ function to display any code areas pointed to by an interrupt vector (e.g. interrupt service routines used in your program).

DEBUG - CALL: This function can be used to „call“ any routine defined in your code when the debugger is on hold.

DEBUG – LOAD TARGET FILE: Starts a new download of the current load module (ABS file). Usually this function will be executed automatically after starting the debugger (defined in debug settings ; see „Project-Setup-Debug“). However, this function can be useful to re-initialize the debugger for the current debug session.

VIEW - SYMBOL: This function displays all (global) symbols of the current project. Information about the type (char, integer etc.) and the location (address in RAM or register) can be shown using „detail“.

VIEW - LOCAL: Similar to View-Symbol, but only local variables of the current function are shown.

VIEW - COMMAND: From this window, the debugger can be controlled using a command line input. All GUI functions are available as individual commands.

Example:

Run continuously

Step

examine PDRG

set break main

Dump/Halfword 0x2000000..0x20000FF

Shortcuts to complex commands can be defined (“alias”) and parameter strings can be substituted by variables.

Example: mydump = Dump/Halfword 2000000..20000ff

A set of commands can be combined to a “macro”. Aliases and macros can be defined using the command window context menu.

Example: restart: set register PC=2000000
 set break main
 run continuously

In order to document or save debugging details such as memory contents, all outputs fed to the command console can be written to a file (“logging”).

A whole “program” of commands (incl. flow control) can be written to a PRC-file and executed. Procedure-files are simple text-files with the extension “.PRC” and can be created by the Softune Workbench editor or any other text editor. To execute a procedure-file, use the command window menu or the “file-open” menu during debugging.

Note:

Procedure-files can be very helpful to configure the debug-environment automatically, perform automated tests (e.g. when a new C-module has to be tested) or to enhance basic emulator functions such as breakpoints. Using a procedure file, breakpoints can be defined dynamically depending on program conditions. See the “Command Reference Manual” (on-line help) for detailed information about the available command and parameter syntax. Some procedure files exist already included in the example project directory.

7.5 Memory Configuration for User Applications

Valid load module files can only be achieved if the language tool configuration is appropriate. The most convenient way to create own projects is to copy the provided Template-Projects from the example-directory and use it as “blank sheet”.

If the Softune Monitor debugger is being used the following memory areas are reserved by the monitor debugger and must not be used by the application.

Memory Map of Softune Workbench Monitor debugger MonDebStarterkit91302_15MHz_CLKT30_V102			
Memory Type	Used for	Area(s)	Sections
Internal CPU Data* RAM	Stack, Data	0x3F000-* 0x3F707	STACK, DATA
External Flash 8MB* memory	Flash Testpattern	0x8.0000-* 0x8.0003	Flash_Test_Pattern (default 0xaa5555aa)
External Flash 8MB*	Debug Kernel	0x0F:C000-* 0x0F:DEB6	CODE, CONST
External Flash 8MB*	Vector Table	0x0F:FC00-* 0x0F:FFFF	INTVECT
Note: Be aware that the memory map may depend on the debugger version and any modifications of the debugger made by the user. The current version of Flash programming tool just supports programming the memory area 0x08.0000 – 0x0F.FFFF			

Memory map of Monitor debugger MonDebStarterkit91302_15MHz_CLKT30_V102

Note:

The following chip select areas are used by the Softune monitor debugger and must not be used or overlap with other chip select areas:

CS0 – external Flash memory 0x0000.0000 – 0x007F.FFFF

CS6 – SDRAM 0x0200.0000 – 0x027F.FFFF

Always check the Linker Settings (“Project – Setup Project – Linker – Disposition”) to make sure the following memory map is applied according to the project configuration:

Recommended linker settings for “STARTERKITMB91302” for debugging			
Memory Type	Used for	Area(s)	Sections
Data RAM,	Variables, Stack	0x3F800-0x3FFFF	DATA, INIT, STACK
External SDRAM 64MB	Code, Const	0x200:0000-0x20F:FBFF	CODE = ROM1
External SDRAM 64MB	Code	0x210:0000..0x27F:FFFF	CODE = ROM2
External SDRAM 64MB	IRQ Vector Table	0x020FFC00-0x020FFFFFF	INTVECT = INTVECT_AREA
External Flash 8MB*	Code, Const	0x08.0004-0x0F.FBFF 0x10.0000-0x7F.FFFF*	CODE, CONST
Note: The current version of Flash programming tool just supports programming the memory area 0x08.0000 – 0x0F.FFFF			

Linker Settings for MB91302A-020 Debug configuration

Recommended linker settings for “STARTERKITMB91302” without Monitor Debugger			
Memory Type	Used for	Area(s)	Sections
Data RAM	Variables, Stack	0x3F000..0x3FFFF	DATA, INIT, STACK,
External SDRAM 64MB	Code, Data	0x200:0000..0x27F:FFFF	CODE, DATA
External Flash 8MB*	Code, Const	0x08:0000..0x7F:FFFF*	CODE, CONST
Note: The current version of Flash programming tool just supports programming the memory area 0x08.0000 – 0x0F.FFFF			

Linker Settings for MB91302A-020 Standalone configuration

Note:

In any case all settings must be checked and corrected corresponding to specific application requirements. Also check settings in the start91302.asm initialisation file! Use the linker mapping list (e.g. *.mp1 file) to check the final memory allocation.

The following chip select areas are used by the Softune monitor debugger and must not be used or overlap with other chip select areas:

CS0 – external Flash 0x0000.0000 – 0x007F.FFFF
CS6 – SDRAM 0x0200.0000 – 0x027F.FFFF

Checklist for project configuration:

- **Make sure, the correct Target MCU (“Project – Setup Project - MCU” menu) MB91302A is selected.**
- **Use the correct linker settings as outlined above depending on your target MCU and the target area (embedded flash or external RAM).**
- **Use the provided templates to make sure the sections are located correctly.**
- **Always check the *.mp1 file (use right mouse click on top *.abs file in the project window of Softune Workbench and select “open list file”) to see the used sections and address areas!**
- **The following Chip select areas are used by the Softune monitor debugger and must not be used or overlap with other chip select areas:**
CS0 – external Flash 0x0000.0000 – 0x007F.FFFF
CS6 – SDRAM 0x0200.0000 – 0x027F.FFFF

Note:

Be aware that the Softune Workbench monitor debugger is only able to debug code which is located in RAM! It is not possible to debug code located in Flash memory!

7.6 Provided Examples

On the “STARTERKITMB91302” CD-ROM and on the MCU website some example projects are provided. The projects can be found in \Software\Examples and are regularly updated on the website. Please check <http://www.fme.gsdc.de/gsdcd.htm> for latest information and version.

Use the “Template” project as a reference to start your own application software. The “Template” project offers a quick start-up and includes basic MCU initialisation files and tools settings (e.g. linker memory map).

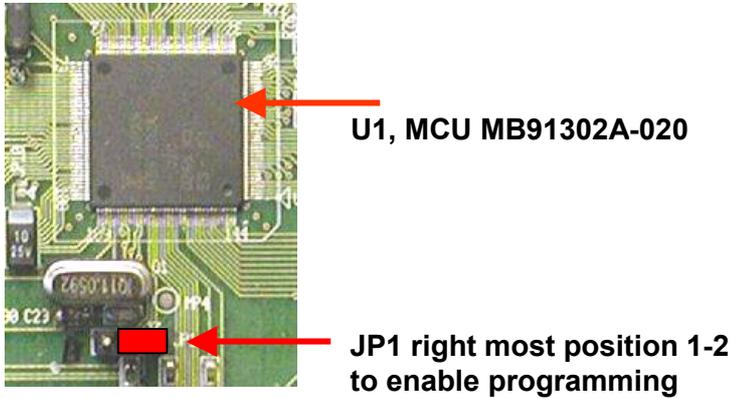
Note:

In any case all settings must ALWAYS be checked and corrected corresponding to specific application requirements. Be aware that the Softune Workbench monitor debugger is only able to debug code located in RAM! It is not possible to debug code located in Flash memory!

7.7 Advanced Monitor Debugger Information

7.7.1 Status Indication of Monitor Debugger after Reset

After power-on the monitor debugger indicates via the 8 user-LEDs the current status. The following status indications exist. Ensure that the CPU is working in normal RUN Mode (Jumper JP1 position 1-2)



- Push the **Reset** button (blue button “Reset”, SW6)
- If the debugger started correctly the 8 user-LEDs show the following pattern:

● ● ○ ○ ○ ○ ○ ○	
D10	D1

Monitor Debugger running, all tests passed
- In case the Monitor Debugger detected any error during start-up the 8 user-LEDs show a special initialisation status code:

- | | |
|-----------------|----|
| ○ ○ ○ ● ○ ○ ○ ○ | |
| D10 | D1 |

Monitor Debugger started, but all tests and settings failed
- | | |
|-----------------|----|
| ○ ○ ● ○ ○ ○ ○ ○ | |
| D10 | D1 |

Clock settings done (CLKB, CLKP, CLKT)
- | | |
|-----------------|----|
| ○ ○ ● ● ○ ○ ○ ○ | |
| D10 | D1 |

External Bus Interface for SDRAM and Flash initialised
- | | |
|-----------------|----|
| ○ ● ○ ○ ○ ○ ○ ○ | |
| D10 | D1 |

Cache initialised (if enabled in debugger source code, default is disabled)
- | | |
|-----------------|----|
| ○ ● ○ ● ○ ○ ○ ○ | |
| D10 | D1 |

Standby Control Register initialised

	UART initialised (115Kbaud, 1N8)
	UART initialisation started but no return from this function
	Start SDRAM Memory test
	SDRAM memory test, Error
	SDRAM Memory test finished, no Error
	Flash Memory test, Error
	Flash Memory test finished, all tests passed, no Errors

7.7.2 Memory Map of Softune Workbench Monitor Debugger

When using the monitor debugger take care that the application is not using the same memory area which is used by the monitor debugger. The following areas are used by the debugger version “MonDebStarterkit91302_15MHz_CLKT30_V102”

Memory Map of Softune Workbench Monitor debugger MonDebStarterkit91302_15MHz_CLKT30_V102			
Memory Type	Used for	Area(s)	Sections
Internal CPU Data RAM	Stack, Data	0x3F000-0x3F707	STACK, DATA
External Flash 8MB* memory	Flash Testpattern	0x8.0000-0x8.0003	Flash_Test_Pattern (default 0xaa5555aa)
External Flash 8MB*	Debug Kernel	0x0F:C000-0x0F:DEB6	CODE, CONST
External Flash 8MB*	Vector Table	0x0F:FC00-0x0F:FFFF	INTVECT
Note: Be aware that the memory map may depend on the debugger version and any modifications of the debugger made by the user.			

Memory map of Monitor debugger MonDebStarterkit91302_15MHz_CLKT30_V102

7.7.3 User Configuration of the Softune Workbench Monitor Debugger

In some cases it might be necessary to modify the monitor debugger (e.g. change crystal frequency, change UART baud rate or use UART1 instead of UART0). For this purpose the source of the monitor debugger can be found in the directory

```
\\Software\\SWB\\MonitorDebugger\\MonDebStarterkit91302_15MHz_CLKT30_V102
```

By default the following settings are exist (u_io.asm) in version V1.02:

- **CLKB Base clock** (CPU clock rate): PLLx4 = 60MHz in case of 15MHz crystal
- **CLKP Peripheral clock**: CLKB/2 = 30MHz in case of 15MHz crystal
- **CLKT Bus clock**: CLKB/2 = 30MHz in case of 15MHz crystal
- **Chip select CS0 external Flash memory 0x0000.0000 – 0x007F.FFFF**
- **Chip select CS6 external SDRAM 0x0200.0000 – 0x027F.FFFF**

- **Communication channel**: For debug interface the UART0 is used, 115kbaud, 8N1
- **Cache**: disabled
- **Special Pins**: output of MCKE, MCKEE, SYSCLK enabled

- **Flash memory (CS0 - 0x0000.0000 – 0x007F.FFFF):**
 - o Area Select Register ASR0 for CS0 (ASR0 = 0x0000.0000)
 - o Area Control Register ACR0 for CS0 (ACR0 = 0x7422)
 - 8Mbyte (0x0000.0000 - 0x007F.FFFF)
 - 16bit data bus
 - pre-fetch off
 - single access
 - write enable
 - big endian
 - normal access (asynchronous)
 - WR pin enabled for write
 - Wait by RDY pin disabled
 - o Area Wait Register AWR0 for CS0 (AWR0 = 0x3378)
 - First access wait cycle: Auto-wait cycle 3
 - Inpage access wait cycle: Auto-wait cycle 3
 - Read/write idle cycle: 1
 - Write recover Cycle: 3
 - Asynchronous write strobe output enabled
 - CS delay yes
 - CS Read/Write setup delay 0
 - RD/WR -> CS hold extension cycle 0

- **SDRAM (CS6 - 0x0200.0000 – 0x027F.FFFF):**
 - Area Select Register ASR6 for CS6 (ASR6 = 0x0200.0000)
 - Area Control Register ACR6 for CS6 (ACR6 = 0x7868)
 - 64Mbyte (0x0200.0000 - 0x027F.FFFF)
 - 32bit
 - single access
 - Enable Prefetch
 - Write Enable
 - Big endian
 - Type SDRAM/FCRAM
 - Area Wait Register AWR6 for CS6 (AWR0 = 0x1159)
 - RAS/CAS delay 1 cycle
 - CAS latency 1 cycle
 - Read to Write cycles 2 cycles
 - Write recover cycles before next read command 2 cycles
 - Write recover cycles 3 cycles
 - RAS active time 1 cycle
 - RAS precharge 2 cycles
 - Memory Control Register MCRA (MCRA = 0x07)
 - 8 columns
 - single write
 - 4 banks for burst write
 - 4 active banks
 - Refresh Control Register (RCR = 0xe247)
 - RCR=0xe247 ->0xe24F (to power-on SDRAM I/F)
 - Chip Select Enable Register CSER (CSER = 0x41)
 - CSER = 0x41 (enable CS0, CS6)
 - Port Function Registers PFRx (to enable control signals used by external bus)
 - PFR8 = 0xE0
 - PFR9 = 0x7E

Please check the corresponding readme.txt of the debugger project to get more details about the monitor debugger and latest information. For most modifications of the debugger it is sufficient to modify the file u_io.asm including all basic hardware related initialisations.

Note:

The monitor debugger requests shortcut between RTS/CTS. Otherwise a communication error will be displayed!

The current version of Flash programming tool only supports the programming of the memory area 0x08.0000 – 0x0F.FFFF! Check the Fujitsu website for any possible updates.

8 Starterkit Limitations

If creating new projects, be sure **not** to use the following functions, which are reserved for the Starterkit:

- ▶ Chip Selects:
CS0 is used for the external 8Mbyte Flash memory (0x0000.0000 – 0x7F.FFFF)
CS6 is used for the external 8Mbyte SDRAM (0x0200.0000 – 0x027F.FFFF)

This Chip select areas must not overlap with any other defined chip select areas!

- ▶ Port PG4-PG7 is used to display the debugger status information on the user LEDs
The ports PG4..7 are initialised to output.
- ▶ UART0 used for monitor debugger and flash programming
- ▶ NMI used for Abort function

- ▶ The following memory areas are reserved for the monitor debugger as minimum

Memory Map of Softune Workbench Monitor debugger MonDebStarterkit91302_15MHz_CLKT30_V102			
Memory Type	Used for	Area(s)	Sections
Internal CPU Data RAM	Stack, Data	0x3F000-0x3F707	STACK, DATA
External Flash 8MB* memory	Flash Testpattern	0x8.0000-0x8.0003	Flash_Test_Pattern (default 0xaa5555aa)
External Flash 8MB*	Debug Kernel	0x0F:C000-0x0F:DEB6	CODE, CONST
External Flash 8MB*	Vector Table	0x0F:FC00-0x0F:FFFF	INTVECT

Note:
Be aware that the memory map may depend on the debugger version and any modifications of the debugger made by the user.

Memory map of Monitor debugger MonDebStarterkit91302_15MHz_CLKT30_V102

Note:

Be aware that the Softune Workbench monitor debugger is only able to debug code which is located in RAM! It is not possible to debug code located in Flash memory!

Note:

The current version of Flash programming tool only supports the programming of the memory area 0x08.0000 – 0x0F.FFFF! Check the Fujitsu website for any possible updates.

9 Programming the external Flash Memory

To program the Monitor Debugger Kernel of the Softune Workbench monitor Debugger or any other application into the external flash following steps must be taken.

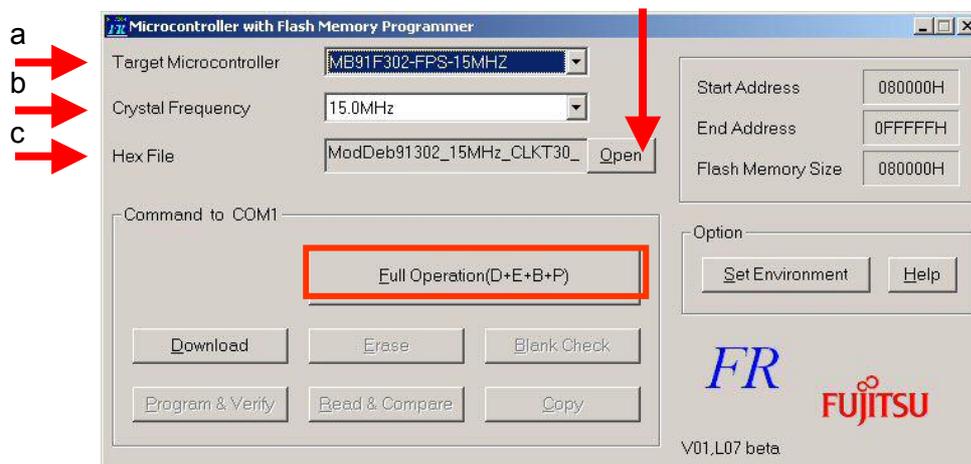
Note:

The current version of the flash programming utility only allows to program the external flash memory in the area 0x8.0000 to 0x0F.FFFF

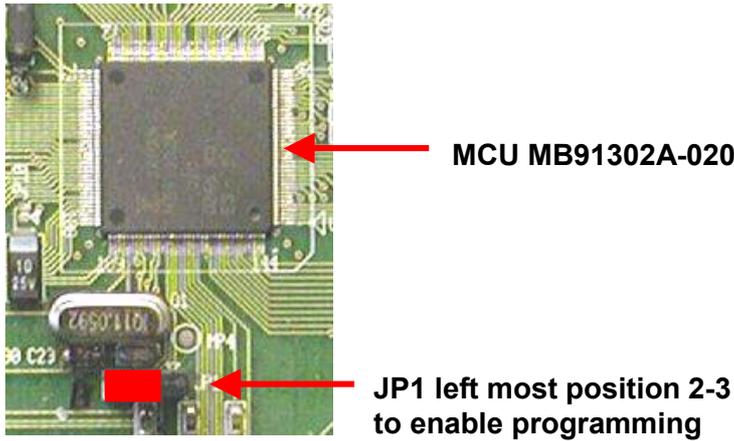
1. To install the Flash Programming utility on the PC the following steps must be taken
 - a. Copy folder “Software\FlashProgramming” from the starterkit CD-ROM to the hard drive of the PC, recommended c:\Softune32\FlashProgramming
 - b. Start Flash Programming tool by double click flash.exe within the copied folder: C:\.\FlashProgramming\FR65e\flash.exe
2. Open the Flash programming Utility by double click the **flash.exe** file
3. Select CPU type, Clock frequency, Hex File as shown in the following figure
 - a. Target Microcontroller: **MB91F302-FPS-15MHz**
 - b. Crystal Frequency: **15.0MHz**
 - c. Hex File: **ModDeb91302_15MHz_CLKT30_v12.mhx**

Use button “Open” and browse to

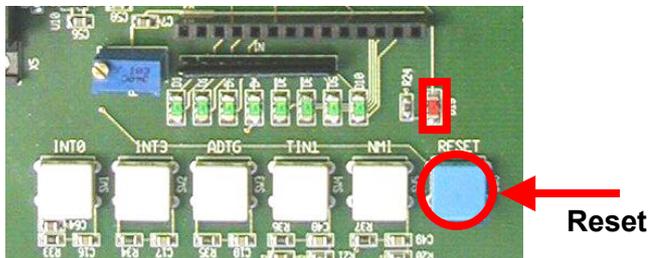
c:\.\Software\MonitorDebugger\ModDeb91302_15MHz_CLKT30_v12.mhx



4. Connect “STARTERKITMB91302” (X5) by **serial cable** with **PC COM1**
5. Set MCU mode to <BIROM> mode: **JP1 position 2-3**

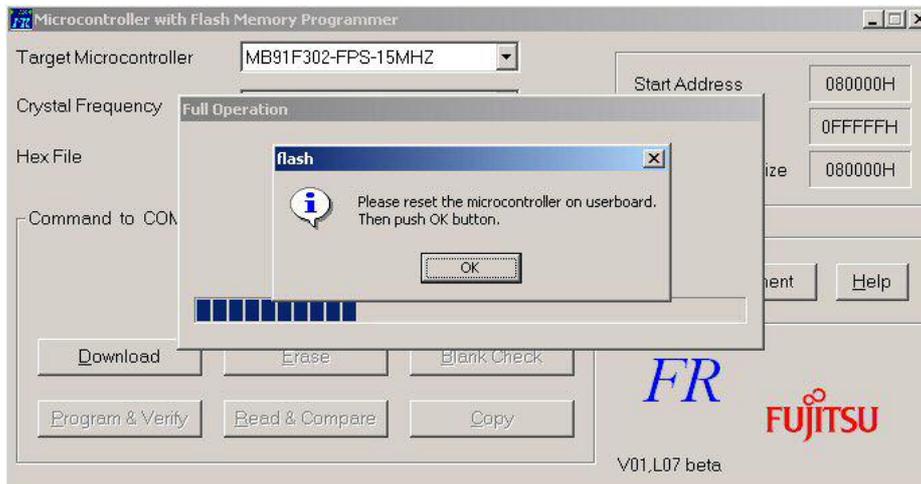


6. **Power-on** "STARTERKITMB91302" (Supply voltage at X1)
7. **Push Reset** button SW6 on "STARTERKITMB91302" and check that LED flashes

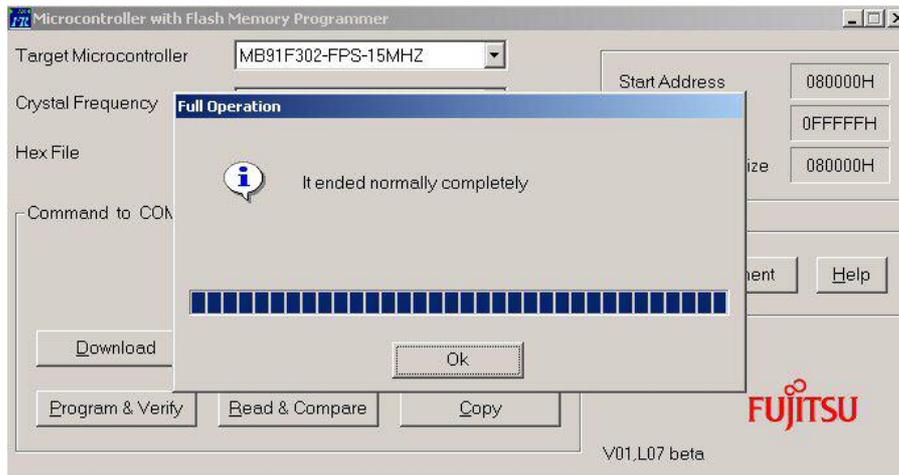


If reset button is pushed LED D19 must flash.

8. Use **<Full Operation>** button of flash.exe programming tool. The following dialog will be displayed:



9. Push **Reset** (blue button "Reset", SW6 on "STARTERKITMB91302") (**red LED D19** must **flash** if button is pressed)
10. **Click OK** button in dialog
11. After programming the Monitor Debugger successfully, the following dialog occurs and LED D9 on the Starterkit is flashing. **Click OK** button in dialog to finish programming.

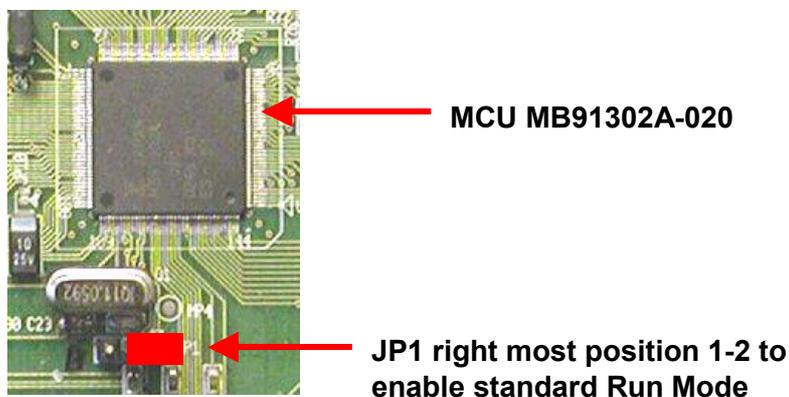


In case any error messages check:

- repeat steps 8-10
- serial cable connection “STARTERKITMB91302” – PC
- Correct “Target Microcontroller” selection in the software on PC
- Correct “Crystal Frequency” selection in the software on PC
- Correct setting JP1 2-3 to enable Boot Mode
- Correct soldering of X5, U10, all circuits related to U10, U1

12. Power-off “STARTERKITMB91302” (disconnect X5)

13. Set MCU mode to <Normal RUN> mode: **JP1 position 1-2**



15. Power-on “STARTERKITMB91302” (connect supply voltage at X1)

16. Push **Reset** (blue button “Reset”, SW6 on “STARTERKITMB91302”)

Now the LED **D10 and D9** on the “STARTERKITMB91302” should be **lit**, all others are off



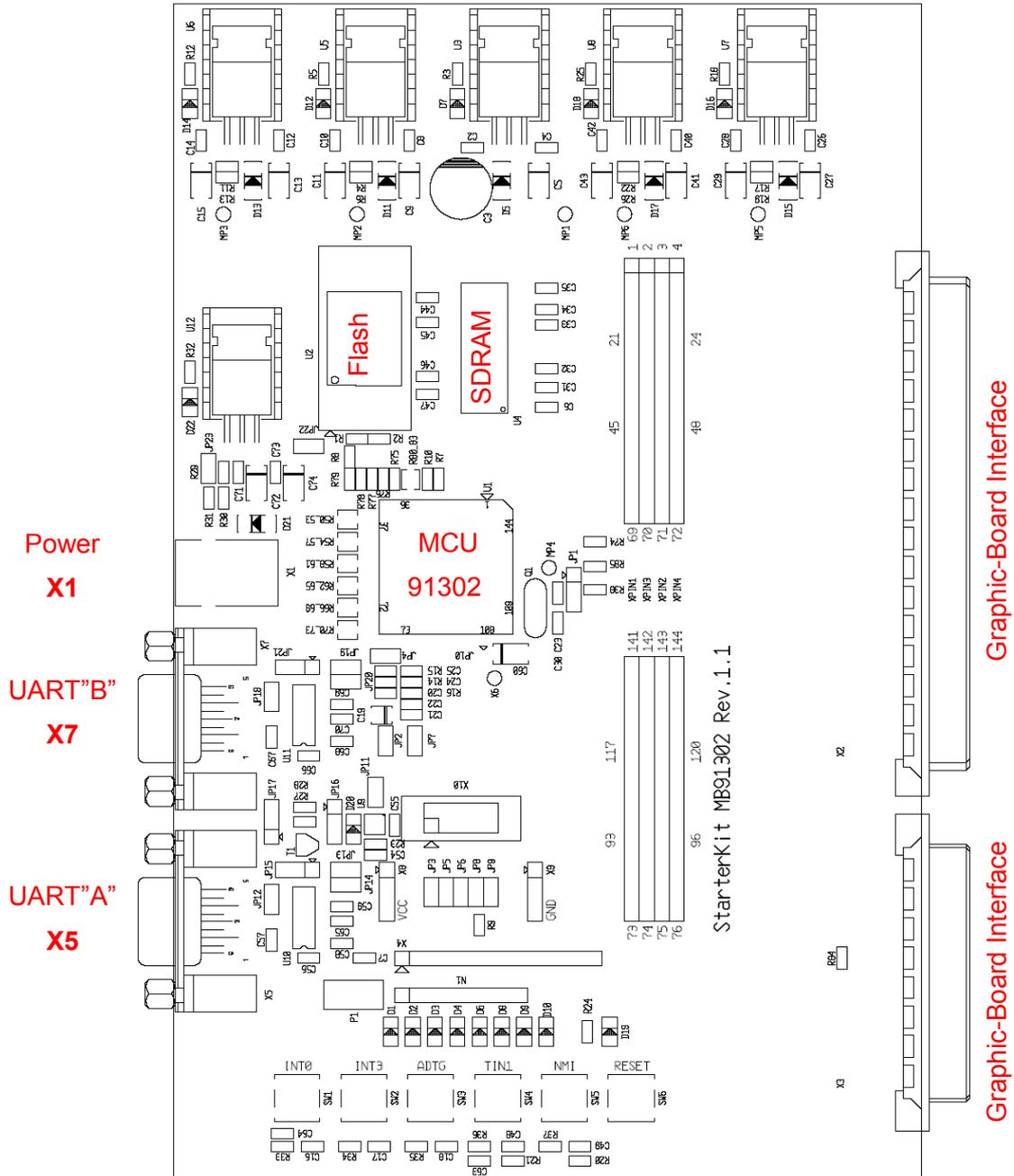
In case any other LED pattern shows up an error occurred, check Chapter 6.6 for further failure analysis.

10 Trouble Shooting

Problem	Solution
Starterkit does not power-on correctly. Some or all of the power LEDs (D7, D10, D12, D14, D16, D22) do not lit up.	<p>Check voltage supply and jumpers :</p> <ul style="list-style-type: none"> * The voltage on DC-Power plug should be in the range of 6-9V. The plug should have + on shield and – on the center. The power supply should be able to supply at least 500mA. To support both, the “STARTERKITMB91302” and a graphic sub-board the supply should be able to supply 1500mA. * The LEDs should light up after switching on the board. Use a multimeter to see whether 5V,3.3V and 2.5V are present on the Vcc/GND test measurement points MP1, 2, 3, 5, 6 and X8. * Check jumper positions according to chapter 4.1 (especially : JP1) ! * Remove all user extensions of the starterkit
Starterkit stays in reset. LED D19 (Reset indicator) ligths up permanently.	<ul style="list-style-type: none"> * LED D19 (Reset) should flash. If D19 lights up permanently, the voltage is too low (below 2.9V) - Increase the voltage supply and check available supply current to solve the problem. * Remove all user extensions on the board to avoid any shortcuts or leakage current!
The communication from Softune Workbench to the evaluation board fails (Communication errors)	<ul style="list-style-type: none"> * Make sure your COM-port number and the baudrate settings are correct (see debugger introduction). The default COM-port is 1 and the default baudrate is 115KBaud. * Make sure no other programs are using the same COM-port on your PC. Close all other applications (e.g. Flash programmer utilities, terminals etc.) * Check if RTS/CTS is closed on the CPU module (check jumper JP12, 13, 14 are closed). * Use only a RS232 extension 1:1 cable
The windows of the debugger are empty.	<ul style="list-style-type: none"> * This is due to a wrong (maybe old) file path information. Closing all windows in the debugger and re-open should fix the problem.
When trying to program the application to the embedded flash, the programmer stops at first address with “loading error” !	<ul style="list-style-type: none"> * You are probably trying to program a program to the flash which is linked for the external RAM area. Check for details the linker memory map and the linker output mapping file *.mp1 file.
Hardware units such as LEDs, user buttons, UART0, 1 are not working.	<ul style="list-style-type: none"> * Make sure you have enabled these units on the evaluation board using the appropriate jumpers (e.g. JP3, 6, 5, 8, 9 for LEDs). See the provided examples and the hardware manual for information on how to control the peripherals.
LCD at parallel LCD port does not work	<ul style="list-style-type: none"> * Adjust the potentiometer P1 to adjust contrast of LCD. Remove resistor network N1
Graphic sub-board does not work	<ul style="list-style-type: none"> Check power supply 8-9V/1500mA. Reduce external bus clock frequency

For more information see the Softune Workbench on-line help, the provided manuals, last minute information and additional or updated documentation on our web-site.

11 Silk-Plot of the Board



12 Related Documents

12.1 Application notes

- MB2198-01 Installation Guide MB2198-10 AN-910026-xx-MB2198INST.pdf
- MB2198-01 Getting Start AN-910027-xx-MB2198Start.pdf
- MB2198-01 Emulation and Debugging AN-910028-xx-MB2198emu.pdf
with Softune

12.2 Manuals

- ▶ MB91301 Hardware Manual
- ▶ MB91302A Datasheet
- ▶ FR Family C-Compiler Manual
- ▶ FR Family Instruction Manual
- ▶ FR Family Assembler Manual
- ▶ FR Family Simulator Manual
- ▶ FR Family Linkage Kit Manual
- ▶ FR Family Softune Workbench Manual

Note :

Updated information of application notes, manuals, datasheets, software tools example projects, headerfiles, ... can be found on the technical MCU web-site: <http://www.fme.gsdc.de/gsdsc.htm>

12.3 Website Links

Datasheets/Hardware Manuals: <http://www.fme.gsdc.de/macrofam/32bitdev.htm>

Application Notes: <http://www.fme.gsdc.de/macrofam/applica0.htm>

Softune Workbench Manuals: <http://www.fme.gsdc.de/products/softune0.htm>

Development Tools: <http://www.fme.gsdc.de/products/32bitdev.htm>

Software Utilities/Examples: <http://www.fme.gsdc.de/macrofam/software.htm>

Package Information: <http://www.fme.gsdc.de/macrofam/packageq.htm>

Third Party Support: <http://www.fme.gsdc.de/macrofam/thirdpar.htm>

European MCU Forum: www.accemic.com/Forums

European MCU Contact: <http://www.fme.fujitsu.com/products/micro/techform.html>

European Distributors: <http://www.fme.fujitsu.com/contacts/02.html>

13 Related Products

- ▶ “STARTERKITMB91302” Evaluation board for MB91302 series
- ▶ MB2198-01 Emulator debugger main unit
- ▶ MB2198-10 DSU cable (not included in main unit)
- ▶ MB2198-100 Evaluation chip adapter board
- ▶ MB2198-101 Emulator probe adapter for package FPT-144P-M12
- ▶ MB2198-90 optional emulation RAM
(only necessary, if no RAM is available on target)
- ▶ MB91V301ACR-ESE1 Evaluation chip for MB91301/302 series
- ▶ MB91302APFF-G-020-BNDE1 Flash-Microcontroller
- ▶ NQPack144SE Socket for package FPT-144P-M12
(Tokyo Eletech Corp. www.tetc.co.jp/e_tet.htm)
- ▶ HQPack144SE Header for FPT-144P-M12

14 Information and Contacts

Information about FUJITSU MICROELECTRONICS Products can be found on the following Internet pages:

Microcontrollers (8-, 16- and 32bit), Graphics Controllers
Datasheets and Hardware Manuals, Support Tools (Hard- and Software)

<http://www.fme.gsdc.de/gsdc.htm>

Linear Products: Power Management, A/D and D/A Converters

<http://www.fme.fujitsu.com/products/linear/start.html>

For more information about FUJITSU MICROELECTRONICS

<http://www.fme.fujitsu.com/products/start.html>

In case of any problems please contact the following email address:

<http://www.fme.fujitsu.com/products/micro/techform.html>

Third Party Support

<http://www.fme.gsdc.de/macrofam/thirdpar.htm>

Microcontroller Forum

www.accemic.com/Forums

European Distributors

<http://www.fme.fujitsu.com/contacts/02.html>