MB91460 SERIES

EVALUATION BOARD

SK-91460P-176PMC

USER GUIDE

FUJITSU
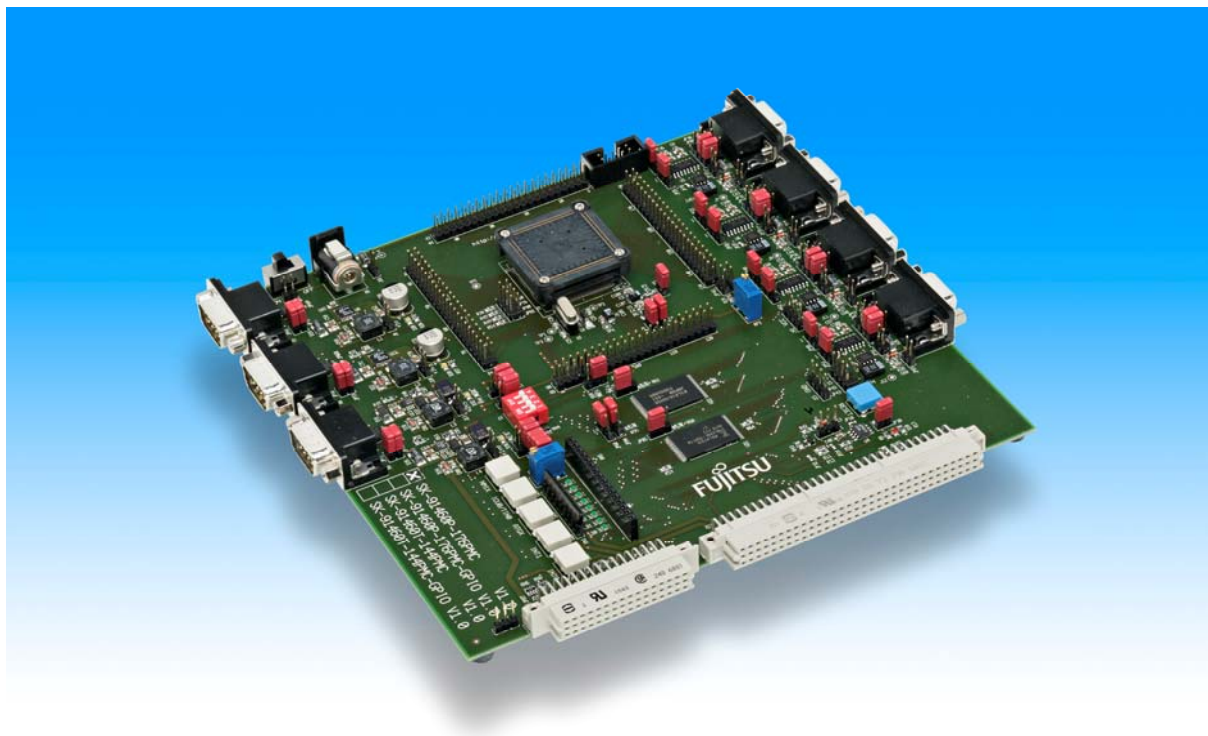
Revision History

Date	Issue
2008/03/31	MHe - Initial version V1.0
2008/05/15	MHe - V1.1 Figure 6.4 added
2008/05/29	MHe - V1.2 typos corrected
2008/06/10	MHe - V1.3 Table 3-11 JP32 added
2008/06/12	MHe - V1.4 Chapter 4.4 corrected
2008/06/16	MHe - V1.5 Chapter 5.4.1 & 5.5.1 memory configuration corrected
2008/09/26	MHe - V1.6 Chapter 0 corrected
2008/11/05	MHe - V1.7 Minor addendum DIP-Switch S1

This document contains 59 pages.

Warranty and Disclaimer

To the maximum extent permitted by applicable law, Fujitsu Microelectronics Europe GmbH restricts its warranties and its liability for **the SK-91460P-176PMC 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 SK-91460P-176PMC 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 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 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

0 Contents

1 Overview

1.1 Abstract

The SK-91460P-176PMC is a multifunctional evaluation board for the Fujitsu 32-bit FR60 Flash microcontroller series MB91460.

It can be used stand-alone for software development and testing or as a simple target board to work with the emulator system.

The board allows the designer immediately to start software development before his own final target system is available.

1.2 Features

▸ Supports Fujitsu's MB91F465PA MCU in FPT-176P-M07 package or the MB2198 Emulator System (ICE 2198-01 + EMA-MB91V460A-002B-80) with the Probe Cables EMA-MB91F465P-NLS-176M07.

▸ 9-15V unregulated external DC power supply

▸ 5V, 3.3V, 2.5V and 1.8V onboard switching mode voltage regulators

▸ Power-LEDs for all supply voltages

▸ Onboard voltage supervisor monitors three of four supply voltages

▸ In-Circuit serial Flash programming (UART4)

▸ All resources available for evaluation

▸ All MCU pins routed to connectors

▸ 4 MHz main crystal

▸ 32 kHz crystal for sub clock operation

▸ Four RS232- or LIN interfaces and three CAN interfaces are usable simultaneously

▸ 3V capable CAN, LIN and RS232 transceivers

▸ 8 User LEDs, optional: alphanumeric standard LC-Display connectable instead of LEDs

▸ Reset button, Reset LED

▸ 5 User buttons

▸ 10-turn potentiometer connectable to ADC channel.

▸ 96+48-pin VG connectors featuring external bus signals, e.g. for graphic sub-board

▸ 8 MB Flash ROM and 2 MB SRAM connected to external bus

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

1.3   General Description

The SK-91460P-176PMC supports the 32-bit Flash microcontroller MB91F465PA.

It can be used as a stand-alone evaluation board or as a target board for the emulator debugger.

The evaluation board supports the following package: FPT-176P-M07.

1.3.1   MCU Clocks

The board is supplied with a 4 MHz crystal as main oscillation source. Using the internal PLL of the µC, internal clock rates up to 100 MHz can be achieved.

1.3.2   RS-232 and LIN

Four separate RS232 transceivers and four single-wire LIN-transceivers (TLE7259) are available to connect four on-chip UARTs to 9-pin D-Sub connectors (X2-X5). The transceivers generate the adequate RS232 levels for the receive (RXD) and transmit (TXD) and LIN bus lines. In RS232 mode, either the DTR line or the RTS line can be selected with jumpers (JP24, 50, 51, 89) to generate a system reset. The RTS signal can be shortcut to CTS using the jumpers JP17, 20, 44 and 45. Each of the four D-Sub connectors can be configured as RS232 or LIN. The LIN Vs line can be powered by the unregulated supply input of the board, so no additional supply is needed (JP30, 31, 56 and 57).

All transceivers are fully 3.3V IO compatible to enable low voltage applications.

In-circuit programming (asynchronous) can be done via LIN-UART 4 (X2).

1.3.3   CAN Bus

Three high-speed CAN transceivers (TLE6250GV33) can be connected to the CAN interfaces of the MCU to allow easy connection to CAN networks.

All transceivers are fully 3.3V IO compatible to enable low voltage applications.

1.3.4   MCU Pins / External Bus Interface Connector

All pins of the microcontroller except the oscillator pins X0/1(A) are connected to edge connectors in functional groups and are directly available to the user. Furthermore, all signals of the external bus interface are connected to a 96+48-pin VG connector to interface e.g. external graphic sub-boards such as Fujitsu's 'Jasmine' sub-board.

1.3.5   Power Supply

The on-board switching mode voltage regulators allow the user to connect an unregulated DC input voltage of +9V to +15V, and supplies all voltages needed by the board and an optional graphic sub-board. The regulators are rated with 3A (5V and 3.3V) resp. 1.5A (2.5V, 1.8V) and are thermally protected against overload.

1.3.6   User Buttons

There are six push button switches on the board, which can be connected to input ports of the microcontroller. This allows the user to evaluate external Interrupts, external ADC trigger or Input Capture functions as well as simple input polling. One button is reserved as 'Reset'-button for the microcontroller, controlled by the supply monitor IC.

1.3.7 User LEDs and optional LCD

Eight user LEDs are connected to Port 15/16 and grounded by a 1k resistor network. If these LEDs are not required, the resistor network can be removed to disconnect the LEDs and to free the IO port.

1.3.8 Onboard Memories

8 MB Flash ROM and 2 MB SRAM are connected to external bus chip select areas 0 (Flash) and 1 (SRAM). SRAM can be accessed 8 bit or 16 bit wide (JP60, JP94).

1.3.9 I2C Bus

Additional pull-up resistors can be connected to the I²C bus lines by setting the according Jumpers (JP68, 76, 90, 92) for I²C 0/1.

1.3.10 Emulator System

If the board is used as an emulator target board, the microcontroller must be removed from the socket and the corresponding probe cable has to be mounted:

Series	V-Chip	Probe cable	Socket
MB91460	EMA-MB91V460A-002B-80	EMA-MB91F465P-NLS-176M07	NQPACK176SD-ND HQPACK176SD

Table 1-1: Emulation System

Attention:

Socket for Flash chip mounting and probe cable connection is very sensitive. Always fit probe cable or socket cover very carefully. If any unexpected problems occur, please check proper connection of MCU/probe cable pads to socket.

Note:

Some customers experience connectivity problems when connecting the MCU into the socket adapter. Only the small red screwdriver available in your box should be used to connect the cover (HQPACK) onto the socket (NQPACK).

If the four screws are not tightening equally, then it may cause a poor contact.

Do not screw the cover too tight (max 0.054 Nm). If you have connectivity problems then please loosen the screws and tighten again the screws equally.

Do not clean NQPACK, YQPACK, and YQSOCKET with steam. Cleaning material will contaminate inside of connector.

2 Installation

2.1 Connection/Power-On

Carefully remove the board from the shipping carton.

First, check if there are any damages before powering up the evaluation board.

For the power supply a DC input voltage of 9V – 15V is recommended. The positive voltage (+) must be connected to the center pin, and ground (GND) must be connected to the shield of the connector X6!

Special care must be taken to the max. input voltage, if the LIN Vs lines are supplied by the board (JP30, 31, 56, 57), since the input supply voltage is directly applied to pin 1 of X2-X5 without a voltage regulator.

After power-on (Switch S3 or JP66), the yellow power-on LEDs (D20, D28, D32, D36) should be lit. If the LEDs do not light up, switch off the power supply and check input polarity and current capability of the DC supply used.

The starter-kit is delivered with programmed Softune Workbench (SWB) Monitor Debugger in the external flash. The SWB Monitor Debugger checks also the external SRAM access at CS1. In case of successful access to external SRAM the LED LD1, LD3 and LD7 are on.

To use the SWB Monitor Debugger, the CPU mode selection must be set to the External Reset Vector mode by setting DIP switch S1 to (On, Off, Off, Off), which is the delivery state.

To use the MCU with the internal flash, which is normally required for the final application, the CPU mode selection must be set to the Internal Reset Vector mode by setting DIP switch S1 to (Off, Off, Off, Off)

The in-circuit programming allows the user to program own applications into the Flash memory. The procedures for Flash programming are described in chapter 6.

If the board shall be used as an emulator target board, switch off the power supply and remove the microcontroller from the socket. Now the probe cable can be mounted on the socket. Take care of the pin 1 marking on the socket and fasten the probe cable with the provided screws.

Do not use any other probe cable than EMA-MB91F465P-NLS-176M07

Connect the probe cable to the EMA-MB91V460A-002B-80 Adapter Board. Check all jumper settings of the evaluation board, the Probe Cable and the Adapter Board.

When turning on the system, be sure to use the following power-up sequence:

1. Power up the Emulator Main Unit (MB2198-01)

2. Power up the Adapter Board (EMA-MB91V460A-002B-80)

3. Power up the target Board (SK-91460P-176PMC)

To turn off the system, switch off the components in reverse order, beginning with the target board.

Please refer to the corresponding user manuals and application notes for the emulator how to set up the emulator system. After power on, the 'Reset'-LED of the emulator must be off and the 'Vcc'-LED must be on.

If the Reset LED is still lit, check the settings of the emulator system and the power supply of the evaluation board.

2.2 Default Jumper settings for SK-MB91460P-176PMC

The following table lists all jumpers including its default setting and location on the starterkit.

Jumper	Description / Function	Type	Default Setting	Coordinates
JP1	SW1 (INT0)	Jumper 3 pin	1-2	E 12
JP2	SW2 (INT1)	Jumper 3 pin	1-2	E 12
JP3	AVcc	Jumper 2 pin	Closed	H 9
JP4	SW3 (ATGX)	Jumper 2 pin	Closed	E 12
JP5	AVCC=AVRH	Jumper 2 pin	Closed	J 7
JP6	SW4 (ICU0/TIN0)	Jumper 2 pin	Closed	E 12
JP7	AVss	Jumper 2 pin	Closed	H 9
JP8	VDD35	Solder JP 3 pin	2-3	E 8
JP9	VDD5	Jumper 2 pin	Closed	E 9
JP10	GDC CS	Jumper 3 pin	Open	B17
JP11	VDD5	Jumper 3 pin	1-2	E 9
JP12	VDD5R	Jumper 3 pin	1-2	E 9
JP13	DTR / DTRx	Jumper 3 pin	Open	J 15
JP14	SW5 (NMIX)	Jumper 2 pin	Closed	E 13
JP15	Reset by UART5	Jumper 2 pin	Open	L 13
JP16	Reset by UART4	Jumper 2 pin	Open	L 13
JP17	RTS-CTS UART5	Jumper 2 pin	Closed	M 6
JP18	Reset: immediate / delayed	Solder JP 3 pin	1-2	K 15
JP20	UART0/4: RTS-CTS	Jumper 2 pin	Closed	M 2
JP21	SIN4: RS232 / LIN	Jumper 3 pin	1-2	L 2
JP22	SIN5: RS232 / LIN	Jumper 3 pin	1-2	L 5
JP24	UART5: DTR/RTS	Jumper 3 pin	Open	M 5
JP25	SOT4: RS232 / LIN	Jumper 3 pin	1-2	L 2
JP26	SOT5: RS232 / LIN	Jumper 3 pin	1-2	L 6
JP27	LIN-UART 4: RS232 / LIN	Jumper 3 pin	1-2	M 2
JP28	LIN-UART 5: RS232 / LIN	Jumper 3 pin	1-2	M 6
JP29	Monitor 2.5V/1.8V	Solder JP 3 pin	2-3	J 16
JP30	LIN4: VBat	Jumper 2 pin	Open	M 2
JP31	LIN5: VBat	Jumper 2 pin	Open	M 5
JP32	Monitor 2V5 / 3V3	Solder JP 3 pin	1-2	J16
JP33	LIN4: Enable	Jumper 2 pin	Open	M 3
JP35	LIN5: Enable	Jumper 2 pin	Open	M 6
JP36	Mounting option: TLE7259/TLE6259	Solder JP 3 pin	1-2	L 3
JP37	Mounting option: TLE7259/TLE6259	Solder JP 3 pin	1-2	L 7
JP38	Reset MCU	Jumper 2 pin	Closed	L 16
JP39	LIN 4: Master	Jumper 2 pin	Open	M 4
JP40	LIN 5: Master	Jumper 2 pin	Open	M 7
JP41	ADC	Jumper 2 pin	Open	K 8
JP42	Reset by UART6	Jumper 2 pin	Open	L 14
JP43	Reset by UART7	Jumper 2 pin	Open	L 14
JP44	UART6: RTS-CTS	Jumper 2 pin	Closed	M 9
JP45	UART7: RTS-CTS	Jumper 2 pin	Closed	M 12

Jumper	Description / Function	Type	Default Setting	Coordinates
JP48	SIN6: RS232 / LIN	Jumper 3 pin	1-2	L 8
JP49	SIN7: RS232 / LIN	Jumper 3 pin	1-2	L 12
JP50	UART6: DTR/RTS	Jumper 3 pin	Open	M 9
JP51	UART7: DTR/RTS	Jumper 3 pin	Open	M 12
JP52	SOT6: RS232 / LIN	Jumper 3 pin	1-2	L 9
JP53	SOT7: RS232 / LIN	Jumper 3 pin	1-2	L 12
JP54	LIN-UART 6: RS232 / LIN	Jumper 3 pin	1-2	M 9
JP55	LIN-UART 7: RS232 / LIN	Jumper 3 pin	1-2	M 12
JP56	LIN 6: VBat	Jumper 2 pin	Open	M 8
JP57	LIN 7: VBat	Jumper 2 pin	Open	M 12
JP59	LIN6: Enable	Jumper 2 pin	Open	M 10
JP60	SRAM: Word / Byte selection	Jumper 3 pin	1-2	F 12
JP61	LIN7: Enable	Jumper 2 pin	Open	M 13
JP62	Mounting option: TLE7259/TLE6259	Solder JP 3 pin	1-2	M 10
JP63	Mounting option: TLE7259/TLE6259	Solder JP 3 pin	1-2	M 13
JP64	LIN6: Master	Jumper 2 pin	Open	M 11
JP65	LIN7: Master	Jumper 2 pin	Open	M 14
JP66	MAINS	Jumper 2 pin	Open	C 2
JP66A	Mounting option: TLE6250GV33/PCA82C250E6250GV33/PCA82C250	Solder JP 3 pin	1-2	C 2
JP66B	Mounting option: TLE6250GV33/PCA82C250TLE6250GV33/PCA82C250	Solder JP 3 pin	1-2	C 5
JP66C	Mounting option: TLE6250GV33/PCA82C250E6250GV33/PCA82C250	Solder JP 3 pin	1-2	C 9
JP68	SDA0 Pullup	Jumper 2 pin	Open	F 5
JP69	CAN0 RX	Jumper 2 pin	Closed	B 3
JP70	CAN1 RX	Jumper 2 pin	Closed	B 7
JP71	CAN2 RX	Jumper 2 pin	Closed	B 10
JP72	CAN0 TX	Jumper 2 pin	Closed	B 3
JP73	CAN1 TX	Jumper 2 pin	Closed	B 7
JP74	CAN2 TX	Jumper 2 pin	Closed	B 10
JP76	SCL0 Pullup	Jumper 2 pin	Open	F 6
JP87	VCC1.8V Ext.	Solder JP 2 pin	Closed	C 17
JP88	Ext. Supply	Solder JP 2 pin	Open	E 17
JP89	UART4: DTR/RTS	Jumper 3 pin	Open	M 2
JP90	SDA1 Pullup	Jumper 2 pin	Open	F 6
JP92	SCL1 Pullup	Jumper 2 pin	Open	F 6
JP94	SRAM: 16bit / 8bit	Jumper 3 pin	1-2	F 12
JP95	Flash Enable	Jumper 3 pin	2-3	G 13
JP96	SRAM Enable	Jumper 3 pin	2-3	G 11
JP102	VG X13A/25A -> GND	Solder JP 2 pin	Closed	K17
JP103	VG X13B/25B -> GND	Solder JP 2 pin	Closed	K17
JP104	MD3, fast clock input	Jumper 3 pin	1-2	F10

Table 2-1: Jumper Settings

2.3 Jumper Location

The following picture shows the silk plot of the starter-kit with marked default jumper settings.
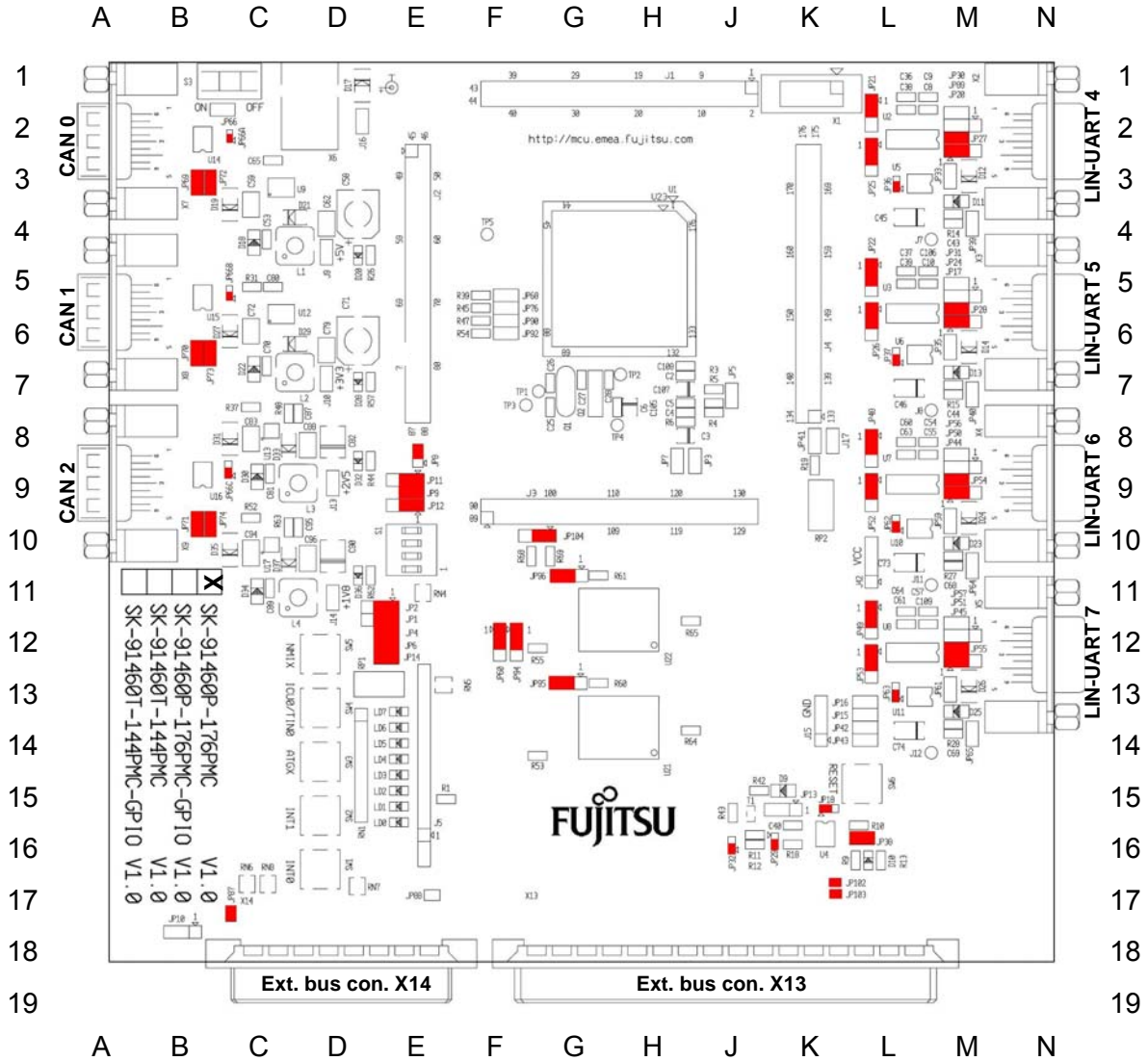

Figure 2-1: Default Jumper Settings

Regarding JP10: Using the same CS for memory device and graphic-controller simultaneously is not possible. E.g. if graphic-controller is controlled by CS1 (JP10 set 2-3), the SRAM has to be disabled (JP96 set 1-2).

2.4   Software Installation

2.4.1   Installation of Softune Workbench

With the "SK-91460P-176PMC" Fujitsu supplies a full working development environment called Softune Workbench V6. To develop own software 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 Workbench installation will fail! Be aware that Softune Workbench does not support multi-user support. Therefore install- and user login must be the same.

2) Browse on the Micros DVD-ROM to "Software Registration". After your free online registration you will receive your password by email.

3) Browse on the Micros DVD-ROM to "Software and Utilities" for installation Softune Workbench.

4) Follow the installation instructions.

5) For the default installation path it is recommended to use c:\Softune6.

6) After the installation is finished. The FR Family Softune Workbench can be started via the Windows "Start" menu.

Figure 2-2: Softune Workbench Start Menu Location

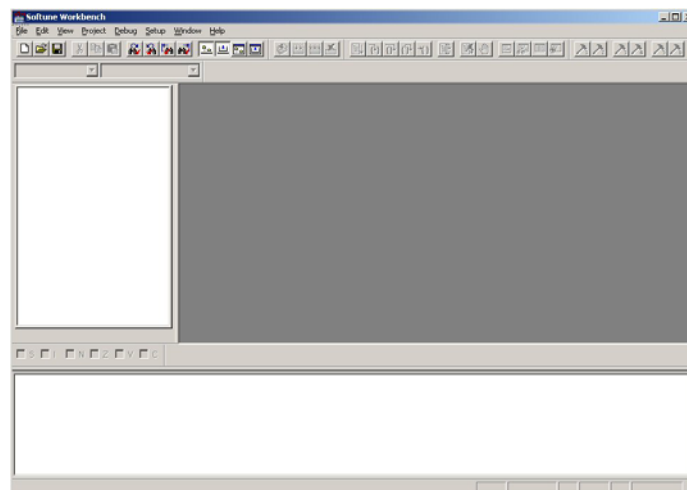7) When Softune Workbench was started the following window will be shown.

Figure 2-3: Softune Workbench V6 IDE

3 Jumpers and Switches

This chapter describes all jumpers and switches that can be modified on the evaluation board. The default setting is shown with a grey shaded area.

3.1 Operating Mode (S1)

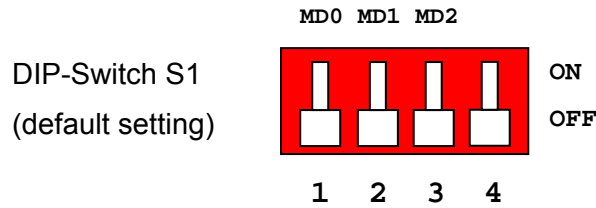| DIP switch	Setting	Logical value
S1/1 (MD0)	ON (closed)	1 (high)
	OFF (open)	0 (low)
S1/2 (MD1)	ON (closed)	1 (high)
	OFF (open)	0 (low)
S1/3 (MD2)	ON (closed)	1 (high)
	OFF (open)	0 (low)
S1/4	NC	

Table 3-1: MCU Operating Mode

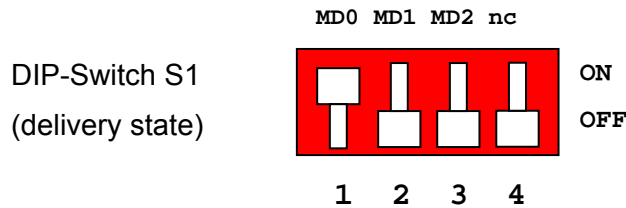**Figure 3-1: MCU mode switch: Internal ROM Mode Vector**

Figure 3-2: MCU mode switch: External ROM Mode Vector

3.2 Power Supply (S3, JP: 8, 9, 11, 12, 66)

The onboard voltage regulators provide stabilized 5V and 3.3V supplies to the MCU and peripherals. Even though they are thermally protected against overload, care must be taken when supplying current for additional circuitry.

The LIN Vs line can be connected directly to the input supply of the board by Jumpers. In this case, the input voltage to the board has to be suitable for the connected bus devices (mostly around 12V). Since there is a protection diode between Vin and Vs, it is not possible to power the board over the LIN bus.

S3   Power Switch

JP8   Selects the MCU External Bus voltage VDD35 (3.3V or 5V)

JP9   Connects the MCU IO voltage to Vcc. Can be used for current measurement.

JP11   Selects the MCU IO and peripherals Vcc voltage (3.3V or 5V)

JP12   Selects the MCU core voltage regulator input voltage (3.3V or 5V)

JP66   Mains jumper; overrides the power switch S3.

Switch	Setting	Description
S3	ON (1-2)	Power ON
	OFF (2-3)	Power OFF

Table 3-2: Power Switch

Jumper	Setting	Description
JP8 (VDD35 3V/5V)	1 - 2	External Bus @ 5V
	2 - 3	External Bus @ 3.3V
JP9 (VDD5)	Closed	Power supply Vcc connected to VDD5
	Open	Power supply Vcc not connected to VDD5
JP11 (VDD5 3V/5V)	1 - 2	MCU IO / Peripherals @ 5V
	2 - 3	MCU IO / Peripherals @ 3.3V
JP12 (VDD5R)	1 - 2	Supply to MCU Core voltage regulator is 5V
	2 - 3	Supply to MCU Core voltage regulator is 3.3V
JP30, 31, 56, 57 (VBAT)	Closed	Vs (Pin1) of LIN connector is powered by the board
	Open	No voltage supply to Vs of LIN A
JP66 (Mains)	Closed	Board is always on
	Open	S3 switches the Board on and off

Table 3-3: Power Supply Configuration

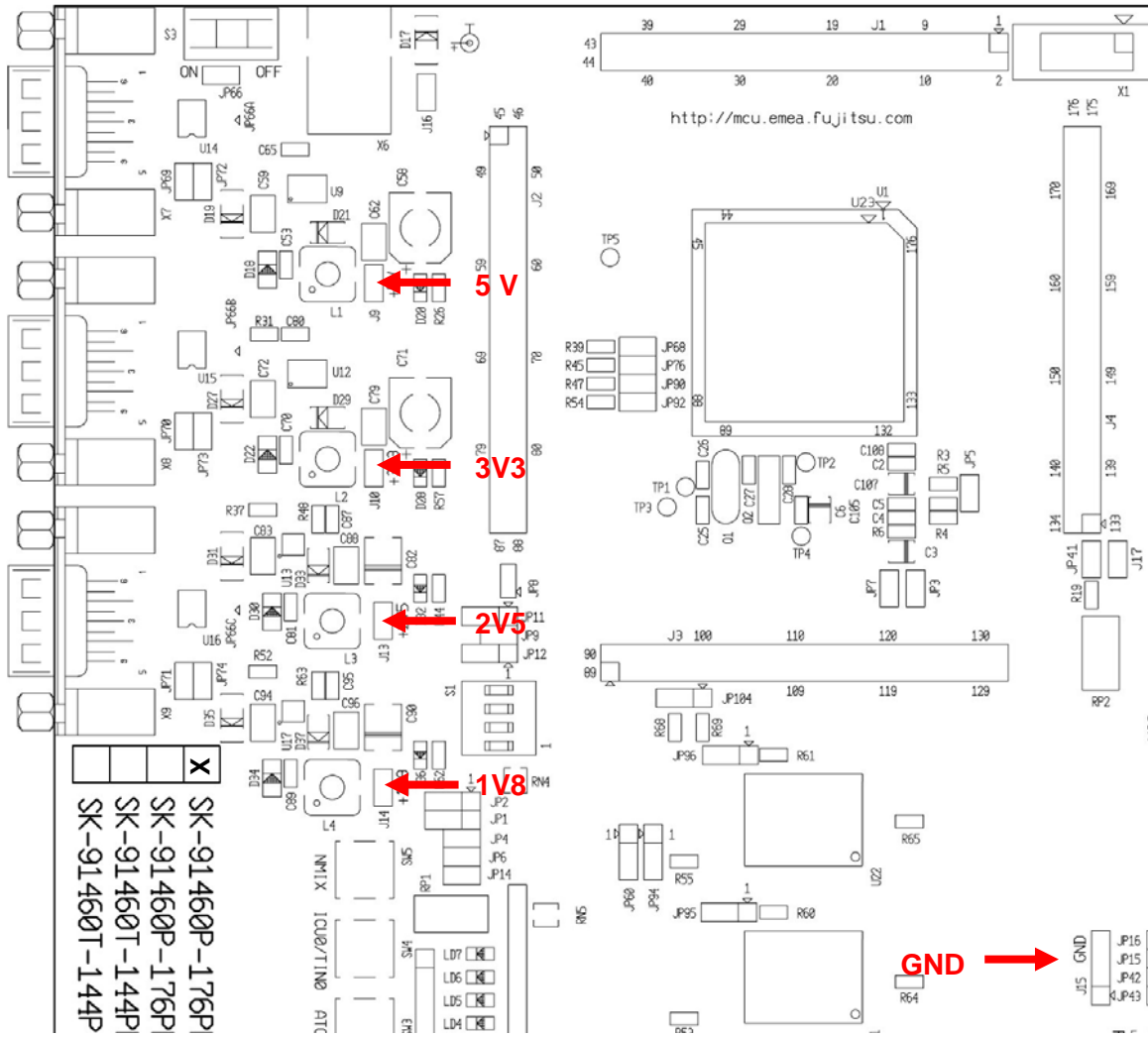By default, I/O supply and MCU Core are set to 5V and the external bus I/F is set to 3.3 V.

Figure 3-3: Voltage Test Pins

Regulator	Test Pin	Voltage
U9	J9	5.0 V
U12	J10	3.3 V
U13	J13	2.5 V
U17	J14	1.8 V
-	J15	GND

Table 3-4: Voltage Test Pins

3.3 Analog Power Supply Voltage (JP: 3, 5, 7)

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

JP3, JP7   connects analog power supply voltages (AVcc and AVss)

JP5   connects the analog reference voltage AVRH to AVcc

Jumper	Setting	Description
JP3 (AVcc)	Closed	AVcc is connected to Vcc
	Open	AVcc is disconnected from Vcc
JP5 (AVRH)	Closed	AVRH is connected to AVcc
	Open	AVRH defined by resistor network[1]
JP7 (AVss)	Closed	AVss is connected to GND
	Open	AVss is disconnected from GND

[1]By default the resistor network (R4, R6) is not mounted on the board

Table 3-5: ADC Supply

Default: JP3, JP5 and JP7 are closed

By default, the A/D-converter supply and reference voltage is same as the microcontroller supply voltage.

Note:

If JP3 and JP7 are open, the user has to supply an adequate analog voltage supply (AVcc and AVss) to the A/D-converter.

If JP5 is open, the resistors R4 and R6 define AVRH.

By default the resistor network (R4, R6) is not mounted on the board.

3.4 LIN / UART Connectors (X2 – X5)

There are four identical circuit blocks for LIN or RS232 connections. Each of the four D-Sub connectors can be configured as LIN or RS232 interface. DTS or RTS can be selected as reset source, and RTS and CTS can be connected by a jumper, since some terminals and Flash programming tools need this connection. Pin 1 (PIN Vs) of X2-X5 can be connected to the voltage input of the board by jumpers in order to supply the LIN bus. When using this feature, pay attention to the input voltage of the board, since there is only a protection diode between Vin and Vs.

LIN-UART 4:

Jumper	Setting	Description
JP21 (RXD)	1-2	SIN4 is connected to RS232 transceiver
	2-3	SIN4 is connected to LIN transceiver
JP25 (TXD)	1-2	SOT4 is connected to RS232 transceiver
	2-3	SOT4 is connected to LIN transceiver
JP27 (RS232/LIN)	1-2	X2 Pin2 is connected to RS232 transceiver (T1out)
	2-3	X2 Pin2 is connected to LIN transceiver (BUS)
JP20 (RTS-CTS)	Closed	RTS and CTS of X2 are connected
	Open	RTS and CTS of X2 are not connected
JP89 (DTR/RTS)	1-2	DTR signal (Pin 6 of X2) is used as reset source
	2-3	RTS signal (Pin 7 of X2) is used as reset source
JP33 (LIN enable)	Closed	LIN transceiver for X2 is enabled
	Open	LIN transceiver for X2 is disabled
JP39 (LIN master)	Closed	LIN-UART 4 is LIN Master
	Open	LIN-UART 4 is LIN Slave
JP30 (LIN Vbat)	Closed	Pin 1 (LIN Vs) of X2 is connected to Vin
	Open	Pin 1 (LIN Vs) of X2 is not connected to Vin

Table 3-6: UART 4 Settings

By default, UART4 is connected to X2 and configured as RS232 interface.

For Flash programming, UART4 has to be used (RS232).

LIN-UART 5:

Jumper	Setting	Description
JP22 (RXD)	1-2	SIN5 is connected to RS232 transceiver
	2-3	SIN5 is connected to LIN transceiver
JP26 (TXD)	1-2	SOT5 is connected to RS232 transceiver
	2-3	SOT5 is connected to LIN transceiver
JP28 (RS232/LIN)	1-2	X3 Pin2 is connected to RS232 transceiver (T1out)
	2-3	X3 Pin2 is connected to LIN transceiver (BUS)
JP17 (RTS-CTS)	Closed	RTS and CTS of X3 are connected
	Open	RTS and CTS of X3 are not connected
JP24 (DTR/RTS)	1-2	DTR signal (Pin 6 of X3) is used as reset source
	2-3	RTS signal (Pin 7 of X3) is used as reset source
JP35 (LIN enable)	Closed	LIN transceiver for X3 is enabled
	Open	LIN transceiver for X3 is disabled
JP40 (LIN master)	Closed	LIN-UART 5 is LIN Master
	Open	LIN-UART 5 is LIN Slave
JP31 (LIN Vbat)	Closed	Pin 1 (LIN Vs) of X3 is connected to Vin
	Open	Pin 1 (LIN Vs) of X3 is not connected to Vin

Table 3-7: UART 5 Settings

By default, UART5 is connected to X3 and configured as RS232 interface.

LIN-UART 6:

Jumper	Setting	Description
JP48 (RXD)	1-2	SIN6 is connected to RS232 transceiver
	2-3	SIN6 is connected to LIN transceiver
JP52 (TXD)	1-2	SOT6 is connected to RS232 transceiver
	2-3	SOT6 is connected to LIN transceiver
JP54 (RS232/LIN)	1-2	X4 Pin2 is connected to RS232 transceiver (T1out)
	2-3	X4 Pin2 is connected to LIN transceiver (BUS)
JP44 (RTS-CTS)	Closed	RTS and CTS of X4 are connected
	Open	RTS and CTS of X4 are not connected
JP50 (DTR/RTS)	1-2	DTR signal (Pin 6 of X4) is used as reset source
	2-3	RTS signal (Pin 7 of X4) is used as reset source
JP59 (LIN enable)	Closed	LIN transceiver for X4 is enabled
	Open	LIN transceiver for X4 is disabled
JP64 (LIN master)	Closed	LIN-UART 6 is LIN Master
	Open	LIN-UART 6 is LIN Slave
JP56 (LIN Vbat)	Closed	Pin 1 (LIN Vs) of X4 is connected to Vin
	Open	Pin 1 (LIN Vs) of X4 is not connected to Vin

Table 3-8: UART 6 Settings

By default, UART6 is connected to X4 and configured as RS232 interface.

LIN-UART 7:

Jumper	Setting	Description
JP49 (RXD)	1-2	SIN7 is connected to RS232 transceiver
	2-3	SIN7 is connected to LIN transceiver
JP53 (TXD)	1-2	SOT7 is connected to RS232 transceiver
	2-3	SOT7 is connected to LIN transceiver
JP55 (RS232/LIN)	1-2	X5 Pin2 is connected to RS232 transceiver (T1out)
	2-3	X5 Pin2 is connected to LIN transceiver (BUS)
JP45 (RTS-CTS)	Closed	RTS and CTS of X5 are connected
	Open	RTS and CTS of X5 are not connected
JP51 (DTR/RTS)	1-2	DTR signal (Pin 6 of X5) is used as reset source
	2-3	RTS signal (Pin 7 of X5) is used as reset source
JP61 (LIN enable)	Closed	LIN transceiver for X5 is enabled
	Open	LIN transceiver for X5 is disabled
JP65 (LIN master)	Closed	LIN-UART 7 is LIN Master
	Open	LIN-UART 7 is LIN Slave
JP57 (LIN Vbat)	Closed	Pin 1 (LIN Vs) of X5 is connected to Vin
	Open	Pin 1 (LIN Vs) of X5 is not connected to Vin

Table 3-9: UART 7 Settings

By default, UART7 is connected to X5 and configured as RS232 interface.

3.5 CAN interfaces (X7-X9)

Three high-speed CAN-transceivers can be connected to the microcontroller's CAN interfaces (CAN0-CAN2).

Jumper	Setting	Description
JP69 (CAN0 RX)	Closed	RX0 of MCU is connected to CAN0 (X7)
JP72 (CAN0 TX)	Closed	TX0 of MCU is connected to CAN0 (X7)
JP70 (CAN1 RX)	Closed	RX1 of MCU is connected to CAN1 (X8)
JP73 (CAN1 TX)	Closed	TX1 of MCU is connected to CAN1 (X8)
JP71 (CAN2 RX)	Closed	RX2 of MCU is connected to CAN2 (X9)
JP74 (CAN2 TX)	Closed	TX2 of MCU is connected to CAN2 (X9)

Table 3-10: CAN Settings

By default, all three CAN Interfaces are connected to X7-X9.

3.6 Reset Generation (JP: 13, 15, 16, 18, 29, 38, 42, 43)

In addition to the internal Power-On reset, the microcontroller can be reset by an external reset circuit (Voltage Monitor) and also by a RS232 interface. Refer to the chapter 'LIN / UART Connectors (X2 – X5)' for DTR / RTS selection.

JP15, 16, 42, 43: Selects X2, X3, X4 or X5 as reset source

JP13 The signal on the DTR/RTS line can be negated with this jumper.
Remove the jumper in order to disable the RS232 reset circuit.

JP18 This solder jumper selects between normal (immediate) reset and delayed reset. In delayed reset mode, the reset button has to be held down for 2 sec before a reset is generated to avoid accidental resets.

JP29 This solder jumper selects the third voltage monitored by the reset IC (5V and 3.3V are always monitored).

JP38 Open this jumper if no external Reset shall be generated. In this case only the internal reset is active (e.g.: power-on)

Jumper	Setting	Description
JP16 (RST 0/4)	Closed	LIN-UART 4 (X2) connected to reset circuit
	Open	LIN-UART 4 (X2) not connected to reset circuit
JP15 (RST 1/5)	Closed	LIN-UART 5 (X3) connected to reset circuit
	Open	LIN-UART 5 (X3) not connected to reset circuit
JP42 (RST 2/6)	Closed	LIN-UART 6 (X4) connected to reset circuit
	Open	LIN-UART 6 (X4) not connected to reset circuit
JP43 (RST 3/7)	Closed	LIN-UART 7 (X5) connected to reset circuit
	Open	LIN-UART 7 (X5) not connected to reset circuit
JP13 (DTR / DTRx)	1-2	No negation for the DTR/RTS signal
	2-3	DTR/RTS signal is negated
JP18 (Reset imm./delayed)	1-2	Reset is applied immediately when SW6 is pressed
	2-3	Reset is applied when SW6 is pressed >2sec
JP29	1-2	VCC1V8 is monitored by reset IC
	2-3	VCC3V3 is monitored by reset IC
JP32	1-2	VCC2V5 is monitored by reset IC
	2-3	VCC3V3 is monitored by reset IC
JP38 (RST MCU)	Closed	External reset generation is active
	Open	No external reset generation

Table 3-11: Reset Connection

By default, the external reset is enabled and set to immediate reset, the reset by UART is disabled.

Note:

While a reset signal is asserted the red Reset-LED D10 is lit.

During normal operation, this LED should be off!

If JP13 (DTR/DTRx) is set, one of the RST jumpers (JP15, 16, 42, 43) and the according DTR/RTS jumper 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.

3.7 User Buttons SW1, SW2, SW3, SW4, SW5 (JP: 1, 2, 4, 6, 14)

JP1, 2, 4, 6, 14   Five user push buttons (SW1-SW5) can be connected to the microcontroller.

Jumper	Setting	Description
JP1 (SW1)	1-2	Pin 146 (INT0) of the MCU is connected to "SW1"
	2-3	Pin 32 (INT0) of the MCU is connected to "SW1" (for PPMUX.PR0=1)
JP2 (SW2)	1-2	Pin 147 (INT1) of the MCU is connected to "SW2"
	2-3	Pin 33 (INT1) of the MCU is connected to "SW2" (for PPMUX.PR0=1)
JP4 (SW3)	Closed	Pin 65 (ATGX) of the MCU is connected to "SW3"
	Open	No connection to the microcontroller
JP6 (SW4)	Closed	Pin 76 (ICU0/TIN0) of the MCU is connected to "SW4"
	Open	No connection to the microcontroller
JP14 (SW5)	Closed	Pin 105 (NMIX) of the MCU is connected to "SW5"
	Open	No connection to the microcontroller

Table 3-12: User Push Buttons

By default, all push-buttons are connected to the microcontroller.

3.8 User potentiometer (RP2, JP41)

A 10kΩ, 10-turn precision potentiometer can be connected to an ADC channel, AN6, with the jumper JP41. The voltage to this pin can be adjusted with RP2 and measured at J17.

Jumper	Setting	Description
JP41	closed	RP2 is connected to AN6

Table 3-13: Potentiometer Settings

3.9   Memories (SRAM/Flash)

SRAM can be accessed 8bit or 16bit wide. This is selectable by jumpers JP60 and JP94 (SRAM). Flash ROM can be accessed only 16bit wide.

The memories can be disabled by jumpers JP95 (Flash) and JP96 (SRAM) to use the chip select lines CSX0 and CSX1 for the graphic interface.

Jumper	Setting	Description
JP95: (Flash Enable)	1-2	Flash is disabled
	2-3	Flash is enabled
JP60 (SRAM 8/16bit)	1-2	SRAM is used in 16bit mode
	2-3	SRAM is used in 8bit mode
JP94: (SRAM A0/D16)	1-2	SRAM is used in 16bit mode
	2-3	SRAM is used in 8bit mode
JP96: (SRAM Enable)	1-2	SRAM is disabled
	2-3	SRAM is enabled

Table 3-14: Flash/SRAM setting

4 Connectors

4.1 Power connector (X6)

The following figure shows the power connection jack X6. This connector is used to connect an external unregulated DC power supply voltage (9V-15V DC) to the evaluation board.

Connector X6:

Shield is connected to GND

+

Center is connected to positive voltage supply (+)
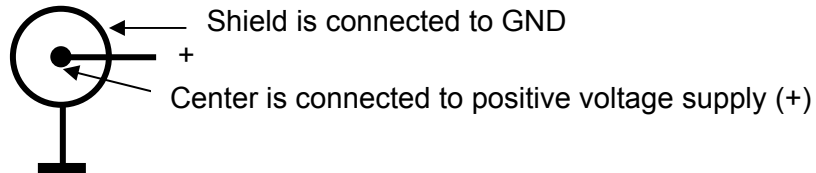
Figure 4-1: Power Connector

The switching mode voltage regulators allow a wide input range. This might be of interest when using Vin to power the LIN bus Vs line.

4.2 Edge connector (J1, J2, J3, J4)

All pins (except oscillator Pins) of the microcontroller are directly connected to J2 to J5 as follows:

Connector	MCU Pins
J1	1 – 44
J2	45 – 88
J3	89 – 132
J4	133 – 176

The odd pin numbers are located on the one side and the even pin numbers are located on the other side of the connector.

4.3   LIN-UART connectors (X2-X5)

Four 9-pin D-Sub female connectors are used for the serial interfaces. Note that X2 to X5 are shared between the RS232- and LIN transceivers and must be configured to the desired functionality (refer to chapter 3.4 for details).
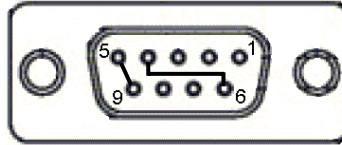
Figure 4-2: UART Connector

Pin Number	Pin Signal	Description
1	+VBat	Power from / to LIN bus
2	TXD	RS-232 transmit output
	LIN	Bi-directional LIN-interface bus
3	RXD	RS-232 receive input
4	DTR	Connected to DSR (pin 6)
5	GND	Ground normally used for RS232 connection
6	DSR	Connected to DTR (pin 4)
7	RTS	Can be connected with CTS by jumper
8	CTS	Can be connected with RTS by jumper
9	LGND	Ground normally used for LIN connection
Shield	GND	Ground

Table 4-1: UART Connector Signals

4.4   CAN Connector (X7-X19)

Six 9-pin D-Sub male connectors are used for the CAN interfaces CAN0-CAN2. All three CAN interfaces can be used simultaneously.
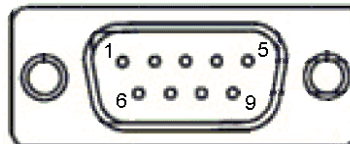
Figure 4-3: CAN Connector

Pin Number	Pin Signal	Description
1	NC	Not used
2	CANL	LOW-level CAN voltage input/output
3	GND	Ground
4	NC	Not used
5	NC	Not used
6	NC	Not used
7	CANH	HIGH-level CAN voltage input/output
8	NC	Not used
9	NC	Not used
Shield	GND	Ground

Table 4-2: CAN Connector Signals

4.5 USER-LEDs & optional LC-Display

Eight LEDs are supplied for user applications. In order to disconnect the LEDs from the related microcontroller port (Port 15, Pin 0-4 and Port 16, Pin 2-5), the resistor network RN1 can be removed.

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

The potentiometer RP1 can be used to adjust the contrast of the LC-Display. Pin 15 and 16 of J1 are normally not mounted. If the used LC-Display has pins for LED backlight at this position, they can be connected here (Pin15: Vcc via 39Ω/0.5W, Pin16: GND).

The following control signals are provided:

Pins for LCD-backlight
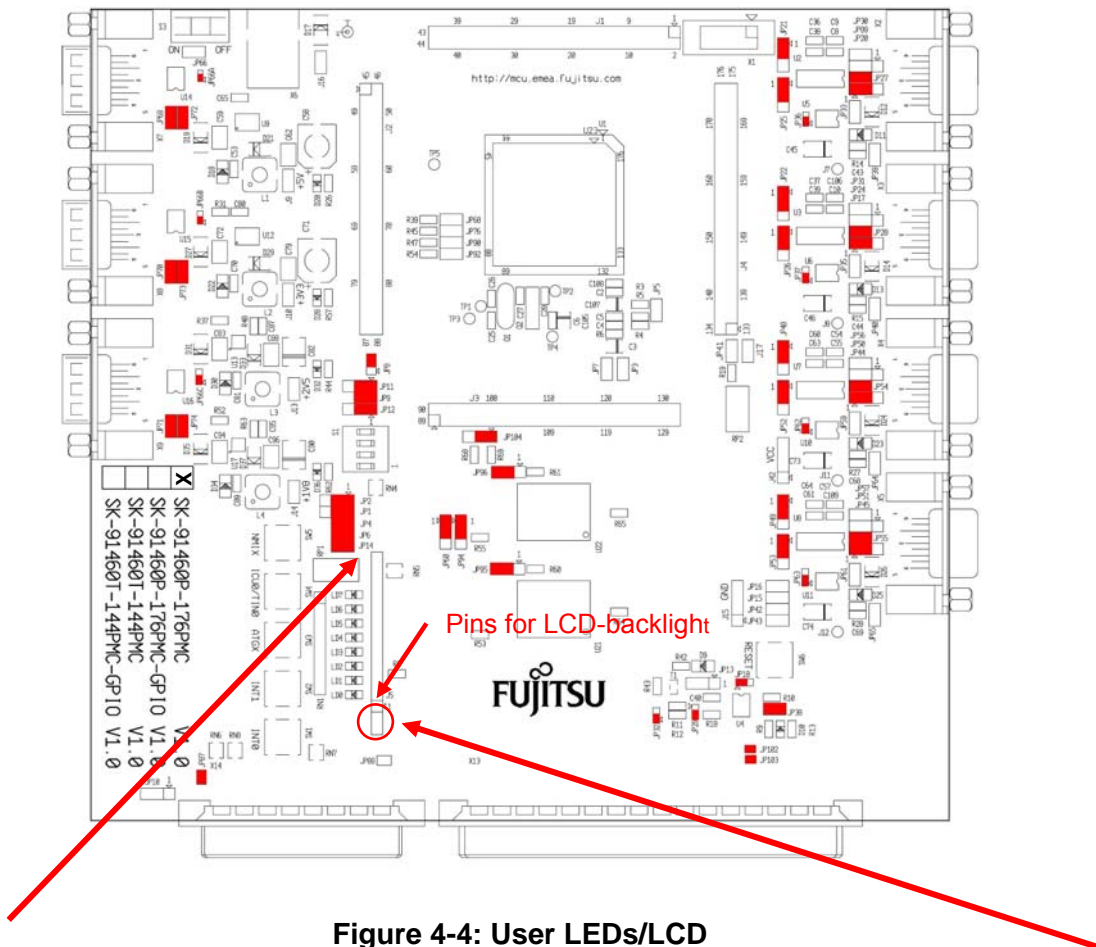
Figure 4-4: User LEDs/LCD

	14	13	12	11	10	9	8	7	6	5	4	3	2	1	16	15
LCD	D7	D6	D5	D4	-	-	-	-	E	R/W	RS	V0	VCC	GND	LED-	LED+
LED	LD7	LD6	LD5	LD4					LD2	LD1	LD0					
Port	16_5	16_4	16_3	16_2					15_2	15_1	15_0					

Table 4-3: User LEDs/LCD

(Optional Pins: Pin15: Vcc via 39Ω/0.5W, Pin16: GND)

4.6   In-Circuit-Programming Connector (X1)

There is a flash-programming socket on the starter-kit which makes it possible to program the flash MCU with a special programming adapter. The reset signal is also available at this connector.
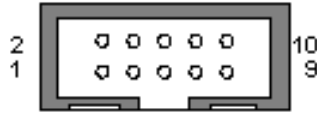
Figure 4-5: In-circuit programming connector

Pin Number	Pin Signal	Description
1	NC	Not used
2	NC	Not used
3	NC	Not used
4	NC	Not used
5	INITX	MCU reset signal
6	SIN4	UART4 receive data
7	SOT4	UART4 transmit data
8	SCK4	UART4 clock
9	VCC	Board supply voltage
10	GND	Ground

Table 4-4: In-circuit programming connector

4.7 External Bus Connectors (X13, X14)

4.7.1 VG96ABC DIN 41612 (X13)

Pin	Name	MCU	Pin	Name	MCU	Pin	Name	MCU
A1	2.5 V	-	B1	2.5 V	-	C1	2.5 V	-
A2	3.3 V	-	B2	3.3 V	-	C2	3.3 V	-
A3	5.0 V	-	B3	5.0 V	-	C3	5.0 V	-
A4	GND	-	B4	GND	-	C4	GND	-
A5	NC	-	B5	NC	-	C5	NC	-
A6	NC	-	B6	NC	-	C6	NC	-
A7	NC	-	B7	NC	-	C7	NC	-
A8	NC	-	B8	NC	-	C8	NC	-
A9	NC	-	B9	NC	-	C9	NC	-
A10	NC	-	B10	D16	P01_0	C10	D17	P01_1
A11	D18	P01_2	B11	D19	P01_3	C11	D20	P01_4
A12	D21	P01_5	B12	D22	P01_6	C12	D23	P01_7
A13	D24	P00_0	B13	D25	P00_1	C13	D26	P00_2
A14	D27	P00_3	B14	D28	P00_4	C14	D29	P00_5
A15	D30	P00_6	B15	D31	P00_7	C15	GND	VSS
A16	GND	VSS	B16	GND	VSS	C16	GND	VSS
A17	A0	P07_0	B17	A1	P07_1	C17	A2	P07_2
A18	A3	P07_3	B18	A4	P07_4	C18	A5	P07_5
A19	A6	P07_6	B19	A7	P07_7	C19	A8	P06_0
A20	A9	P06_1	B20	A10	P06_2	C20	A11	P06_3
A21	A12	P06_4	B21	A13	P06_5	C21	A14	P06_6
A22	A15	P06_7	B22	A16	P05_0	C22	A17	P05_1
A23	A18	P05_2	B23	A19	P05_3	C23	A20	P05_4
A24	A21	P05_5	B24	A22	P05_6	C24	A23	P05_7
A25	A24	-	B25	A25	-	C25	NC	-
A26	NC	-	B26	NC	-	C26	NC	-
A27	NC	-	B27	NC	-	C27	GND	VSS
A28	P14_1	P14_1	B28	P14_2	P14_2	C28	P14_3	P14_3
A29	P35_4	P35_4	B29	P35_5	P35_5	C29	P35_6	P35_6
A30			B30			C30	GND	VSS
A31	2.5V	-	B31	2.5V	-	C31	NC	-
A32	GND	VSS	B32	GND	VSS	C32	GND	VSS

Table 4-5: VG96 (X13) Signals

> **Note:**
> The pin numbers of the VG connectors refer to the male connectors, that plugs into the Starterkit.

4.7.2 VG48ABC DIN 41612 (X14)

Pin	Name	MCU-P.	Pin	Name	MCU-P.	Pin	Name	MCU-P.
A1	CSX0	P09_0	B1	GND	VSS	C1	CSX1	P09_1
A2	CSX2	P09_2	B2	1.8V	-	C2	CSX1/CSX2	-
A3	NC	-	B3	GND	VSS	C3	NC	-
A4	X14A_CSX6	-	B4	1.8V	-	C4	RDY	P08_7
A5	X14A_BGRTNX	-	B5	GND	VSS	C5	X14C_BRQ	-
A6	RDX	P08_4	B6	NC	-	C6	WRX0	P08_0
A7	WRX1	P08_1	B7	NC	-	C7	X14C_WRX2	-
A8	X14A_WRX3	-	B8	NC	-	C8	ASX	P10_1
A9	NC	-	B9	NC	-	C9	SYSCLK	P10_0
A10	X14A_DREQ0	-	B10	NC	-	C10	X14C_DACKX0	-
A11	X14A_DEOP0	-	B11	NC	-	C11	NC	-
A12	INT2	P24_2	B12	NC	-	C12	SDA0	P22_4
A13	INT3	P24_3	B13	NC	-	C13	SCL0	P22_5
A14	SIN5	P19_4	B14	Vin	-	C14	SOT5	P19_5
A15	GND	VSS	B15	Vin	-	C15	SCK5	P19_6
A16	INITX	INITX	B16	Vin	-	C16	GND	VSS

Table 4-6: VG48 (X14) Signals

> **Note:**
> The pin numbers of the VG connectors refer to the male connectors, that plugs into the Starterkit.

4.7.3 Distance between VG-Connectors

To connect own applications to the Starterkit the following figure shows the distance between the VG DIN 41612 connectors.

8,89mm/
350mil
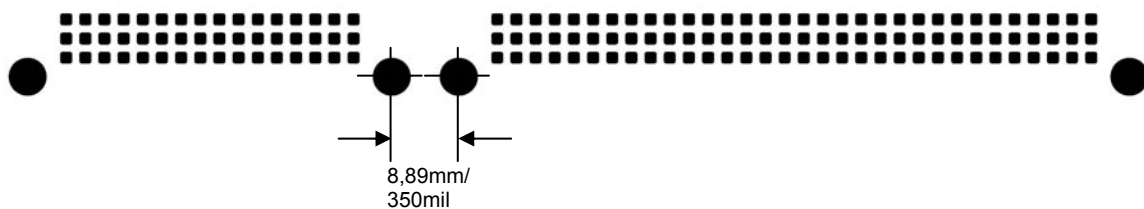
Figure 4-6: Layout of DIN VG Connectors:

5 Getting Started

Make sure that the FR Family Softune Workbench is installed on your PC and the "SK-91F460P-176PMC" is ready to use (power-on, serial connection (UART4, X2) to COM port of the PC).

5.1 Introduction to Softune Workbench

Start Softune Workbench. Select <File> <Open Workspace> to load the first example "91460_template_91465p.wsp". All examples can be found on our webpage at http://mcu.emea.fujitsu.com/mcu_product/mcu_all_software.htm.

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.
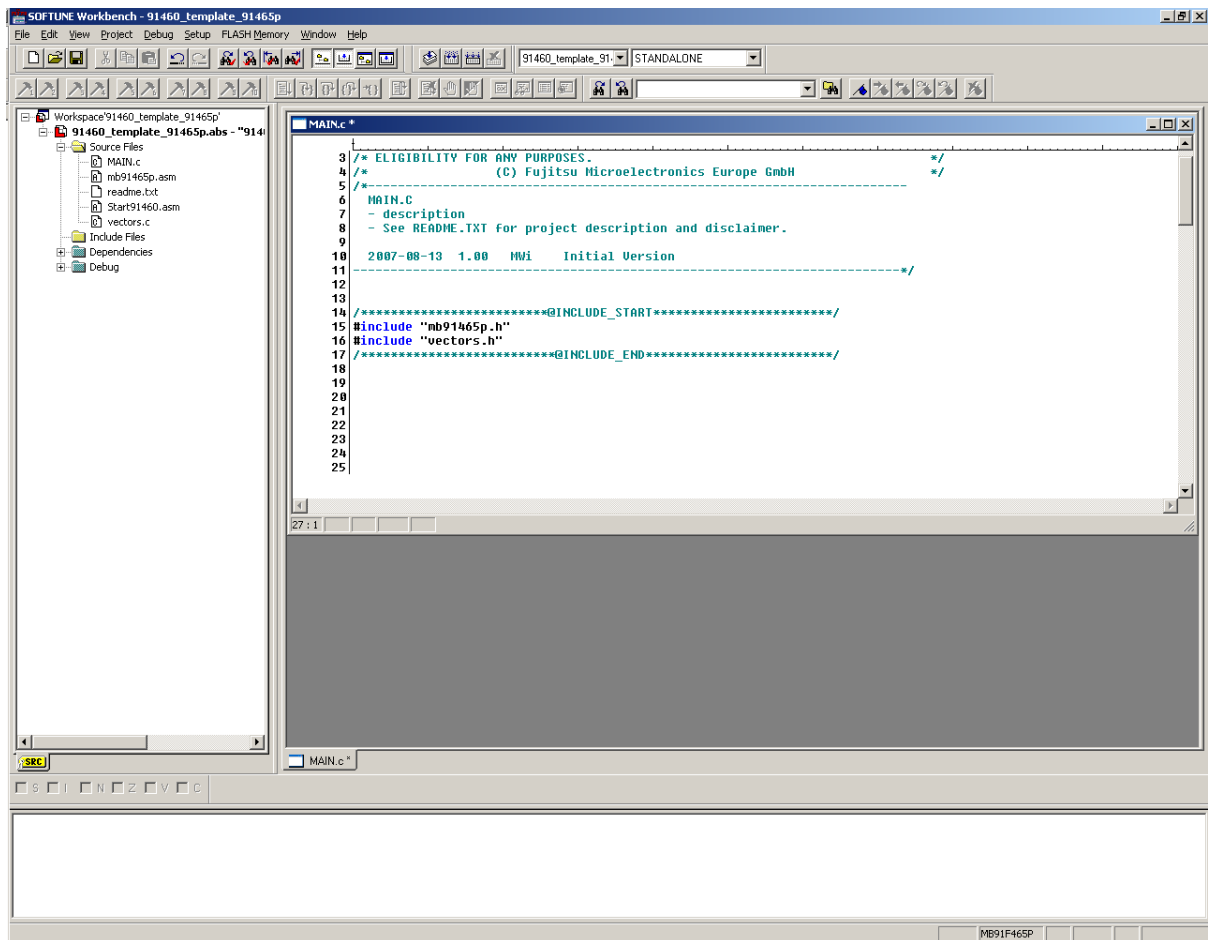
Figure 5-1: Softune Workbench main window

> **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 build, 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: 91460_template_91465p.prj - STANDALONE--------------------

MAIN.c

*** C:\Projects\91460serie\91460_template_91465p-v10\SRC\MAIN.c(46) E4065B: expected a ";"

*** C:\Projects\91460serie\91460_template_91465p-v10\SRC\MAIN.c(57) W1012B: warning: parsing
restarts here after previous syntax error

1 error detected in the compilation of "C:\Projects\91460serie\91460_template_91465p-
v10\SRC\MAIN.c".

-----------------------------

Error detected.

-----------------------------
```

To locate the position in the source-file, where the error 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.

5.2   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 start91460.asm file, MCU header file, 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**

5.2.1   Create a New Project:

1. In order to start a new user-project use the **"91460_template_91465p"** project
2. **Copy** the folder **91460_template_91465p** and **rename** the folder to e.g. my_application
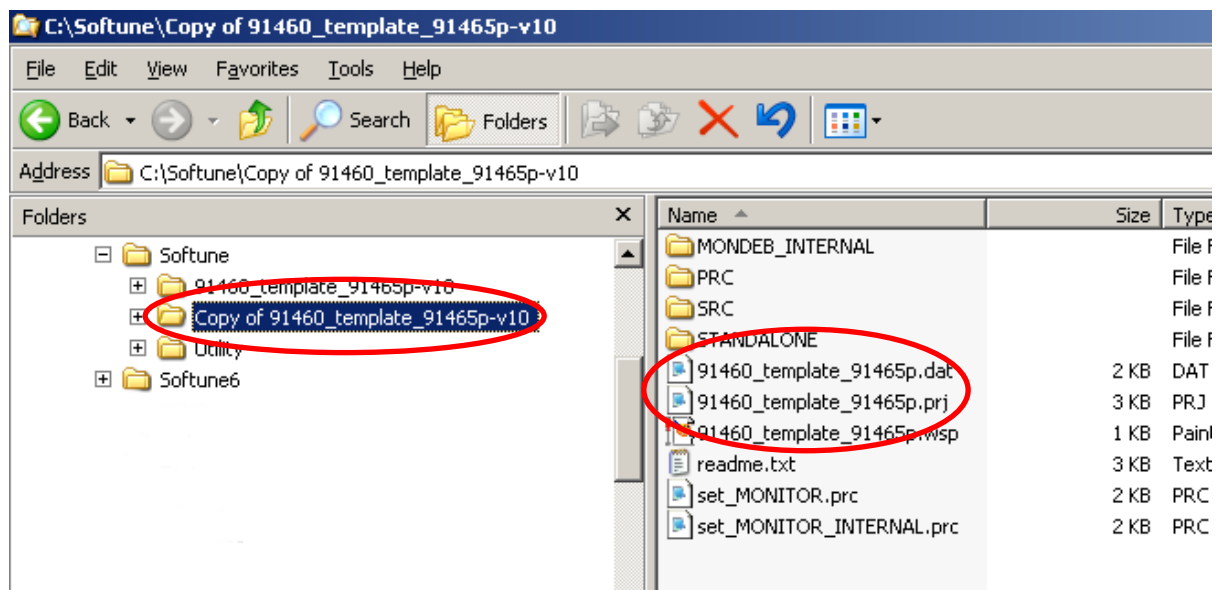
3. **Enter the folder "my_application"**
   Rename "91460_template_91465p.prj" into "my_application.prj"
   Rename "91460_template_91465p" into "my_application.wsp"

4. **Edit "my_application.prj"**
   Rename "91460_template_91465p" -> "my_application"

5. **Edit "my_application.wsp"**
   Rename "91460_template_91465p" -> "my_application"
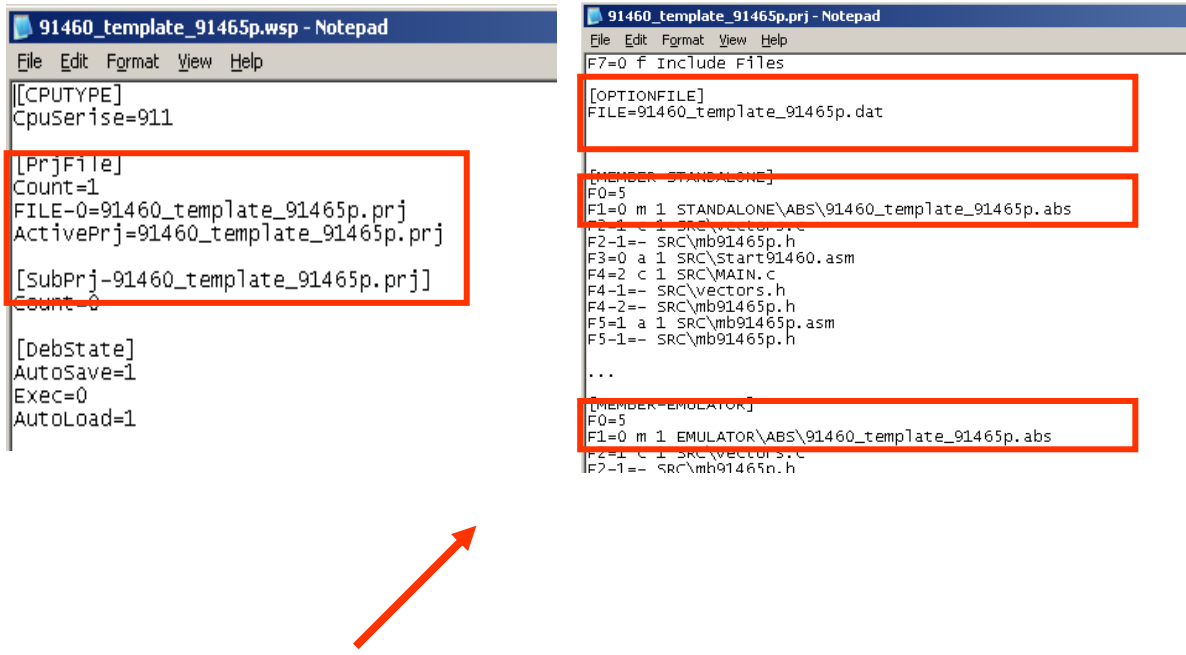
Rename "91460_template_91465p"
into "my_application"

6. Start Softune Workbench and open your project
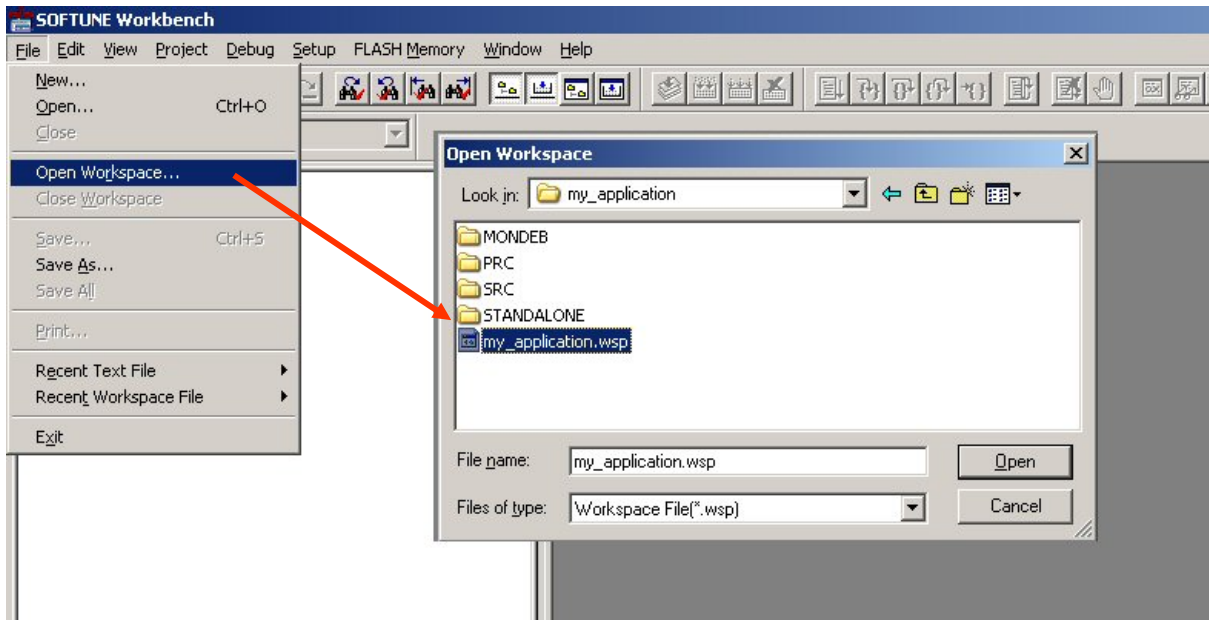
Figure 5-2: Open Softune Workbench workspace (*.wsp)

7. Select the active configuration STANDALONE. In this configuration the application is linked for the available memory on flash-device, e.g. MB91F465P.

8. Use "**Make**", or "**Build**" to compile and link the project for available memory on flash-device. 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.

9. Now, the application can be debugged, therefore, please have a look at the next chapter. Or the generated *.mhx file (Motorola S-Record) can be programmed with the flash programming utility into the internal flash. Please keep in mind, that therefore the mode pins have to be set to "Internal ROM Mode Vector".

> **Note:**
> The starter-kit is delivered with the mode pin settings for the "External ROM Mode Vector", which allows to debug the application with the pro-programmed SWB Monitor Debugger.
>
> If the application should be programmed into the internal flash, the mode pins have to be switched to the "Internal ROM Mode Vector".

5.3 Softune Workbench Monitor Debugger

5.3.1 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: The simulator is part of the Softune Workbench and does not require any target hardware. 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 SWB Monitor Debugger: This debugger type requires an evaluation board like the "SK-91F460P-176PMC" connected to one of the COM-ports of your PC. Explanations in this manual refer to the SWB Monitor Debugger only. For further information please see also the application note "swb_monitor_debugger"

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 MB91460 series is the MB2198-01 system. More information about this system can be found on the Fujitsu 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. The current selection is blue marked (icon):
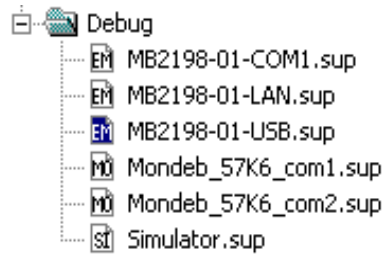
There are some little restrictions, when using the SWB Monitor Debugger:

1) Due to the fact that the application code is located in the external RAM, the execution speed of the code is slower, because of the slower external bus access.

2) Depending on the used linkage option of the SWB Monitor Debugger, CS 0 and CS 1 are required for the external memory. The affected CS has to be switched off in the start-up of the application. (#set CS0 OFF, #set CS1 OFF).

3) Since some settings for the SWB Monitor Debugger during start-up are required, the start-up file cannot be debugged. After starting the SWB Monitor Debugger, the program execution stops at main(). Furthermore, the clock settings of the application start-up are omitted. For the desired clock settings the SWB Monitor Debugger has to be adapted.

4) To indicate the current state of the SWB Monitor Debugger a certain port is used. This feature can also be disabled to give the port free for the application.

To stop the execution of the program the external interrupt INT0 is used. This feature can also be disabled to give the port free for the application. The "abort" button of the Softune Workbench is not fully supported. A detailed explanation of the necessary changes of the SWB Monitor Debugger can be found in the application note "mb91460_swb_monitor_debugger".

5.4 SWB Monitor Debugger with external Flash and external RAM

After software development the SWB Monitor Debugger can be used for debugging.

1) Please make sure, that SWB Monitor Debugger is programmed into the external Flash. To use the Monitor Debugger, the CPU Mode selection must be set to "External ROM Mode Vector" mode by setting DIP switch 1 to (On, Off, Off, Off).
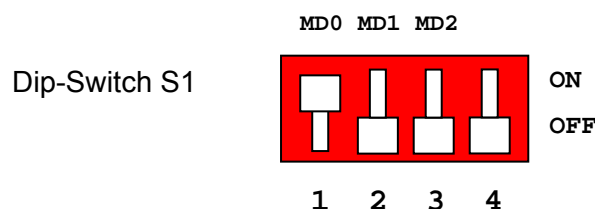
Figure 5-3: SW1 setting for using Monitor Debugger in external Flash

2) The correct start-up of the SWB Monitor Debugger is indicated by LD0, LD2 and LD3
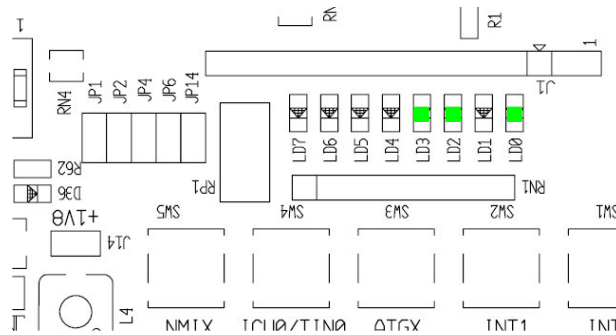
Figure 5-4: Correct start-up of the SWB Monitor Debugger

3) To start the SWB Monitor Debugger, double-click the entry "Mondeb_57K6_com1.sup" for communication via COM1 (57K6 baud, 8N1) or "Mondeb_57K6_com2.sup" for communication via COM2 (57K6 baud, 8N1).
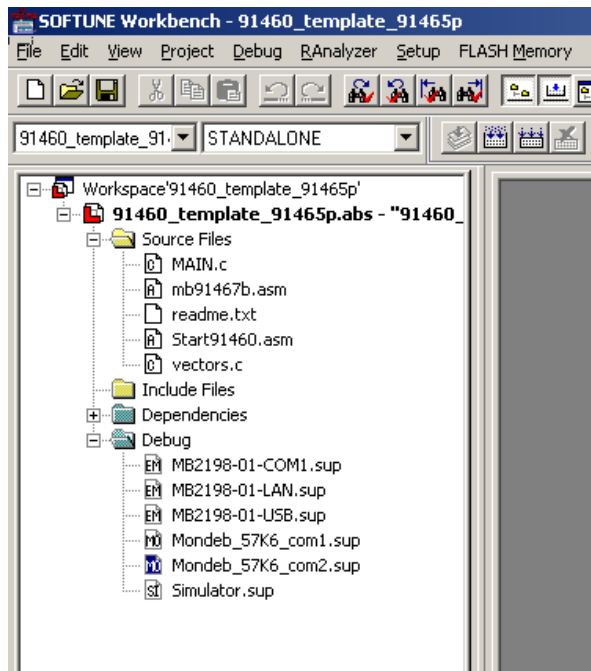
Figure 5-5: Start SWB Monitor Debugger with external flash and SRAM

4) Since some settings for the SWB Monitor Debugger during start-up are required, the start-up file can not be debugged. The settings are done with the help of procedure files. After starting the SWB Monitor Debugger, the program execution stops at main().

5.4.1 Memory configuration

The following picture shows the linkage map and the memory organisation, if the SWB Monitor Debugger is used with external flash and SRAM.
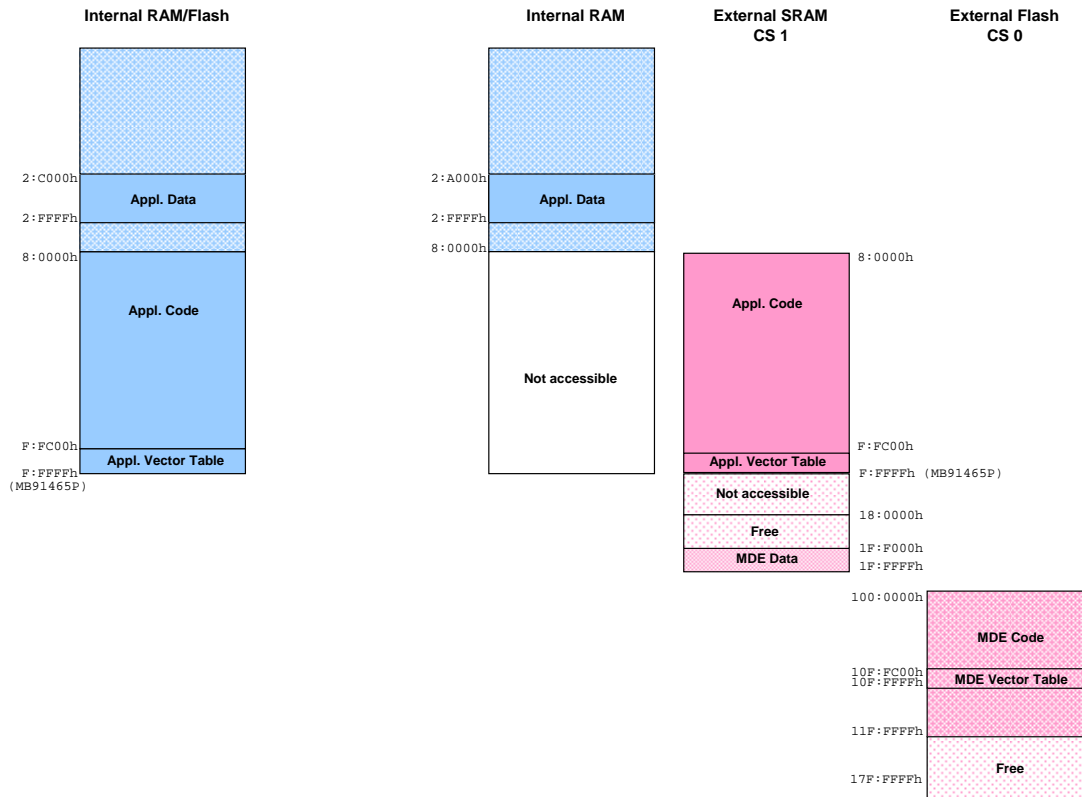
Figure 5-6: Default linkage map for the SWB Monitor Debugger of the MB91465P

On the left hand side the memory organisation of the normal operation mode of the microcontroller MB91F465P can be seen, the whole application is internal. For this operation mode the mode pins have to be set to 0-0-0-0, hence DIP switch S1 has to be set to off, off, off, off.

On the right hand side the microcontroller runs in the external reset-vector mode, which requires the mode pins setting 1-0-0-0, hence DIP switch S1 has to be set to on, off, off, off. In this mode the internal flash is not accessible any more. The external SRAM takes the place of the internal flash. The kernel of the SWB Monitor Debugger is located in the external flash.

5.5 SWB Monitor Debugger with external RAM

After software development the SWB Monitor Debugger can be used for debugging. For further information regarding "getting started with the Softune Workbench", please refer to the Application mode "mcu-an-391027-e-mb2198_getting_started.pdf".

1) Please make sure, that SWB Monitor Debugger is programmed into the internal Flash. To use the Monitor Debugger, the CPU Mode selection must be set to "Internal ROM Mode Vector" mode by setting DIP switch 1 to (Off, Off, Off, Off). The correct start-up of the SWB Monitor Debugger is indicated by LD2, LD3.
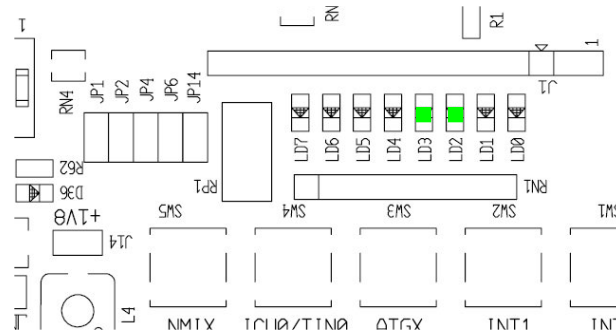
Figure 5-7: Correct start-up of the SWB Monitor Debugger

2) Select the active configuration MONDEB_INTERNAL. With this selection the project is linked to be used with the SWB Monitor Debugger. Corresponding linker settings are done for the memory map in order to avoid overwriting of the sections which are used by the SWB Monitor Debugger. The code is linked in this configuration for external SRAM; otherwise it is not possible to debug code with the Softune Workbench SWB Monitor Debugger. Since the linkage map is different from the linkage map of the configuration STANDALONE, a rebuild is required.
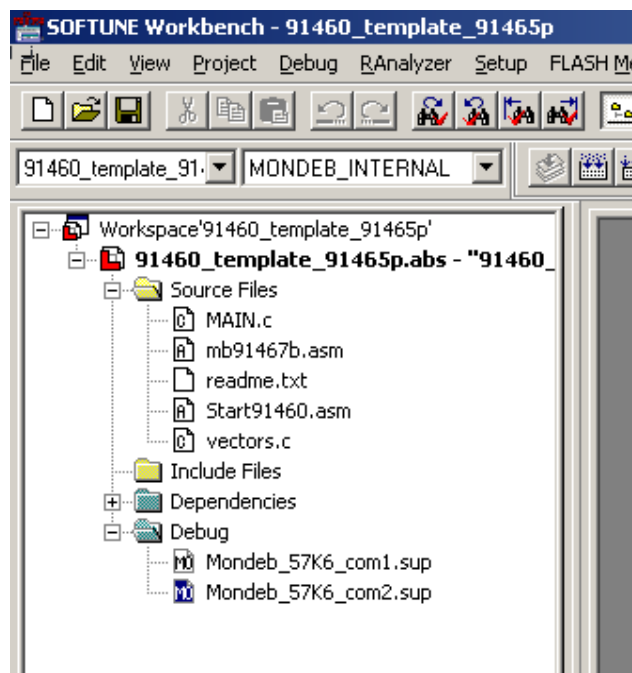
Figure 5-8: Start Softune Workbench SWB Monitor Debugger with external SRAM

3) To start the SWB Monitor Debugger, double-click the entry "Mondeb_57K6_com1.sup" for communication via COM1 (57K6 baud, 8N1) or "Mondeb_57K6_com2.sup" for communication via COM2 (57K6 baud, 8N1).

4) Since some settings for the SWB Monitor Debugger during start-up are required, the start-up file cannot be debugged. The settings are done with the help of procedure files. After starting the SWB Monitor Debugger, the program execution stops at main().

5)

If you wish to change the COM-port, please right-click on the entry and select change. After the project is built and debugged successfully the application has to be re-build in the active configuration STANDALONE to get the application, linked for the flash device.

5.5.1 Default Linkage Map

The following picture shows the linkage map and the memory organization if the SWB Monitor Debugger is used with external SRAM only.
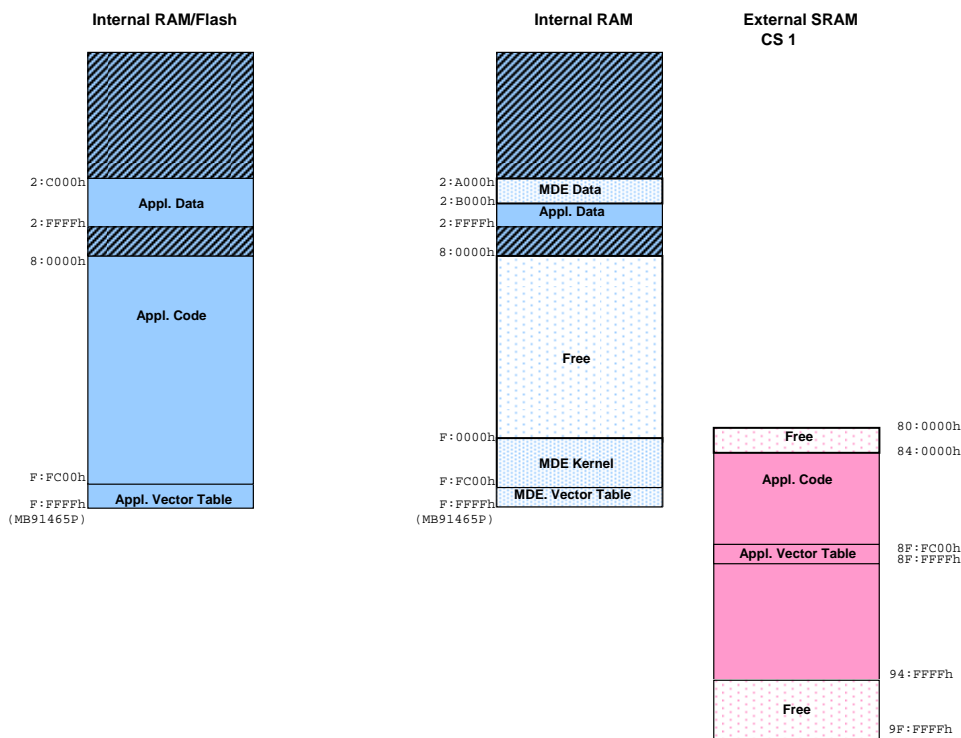
Figure 5-9: Default Linkage Map for the SWB Monitor Debugger of the MB91465P (II)

On the left hand side the memory organisation of the normal operation mode of the microcontroller MB91F465P can be seen, the whole application is internal. For this operation mode the mode pins[0:2] have to be set to V_{SS}-V_{SS}-V_{SS}, hence for the SK-91F460P-176PMC DIP switch S1 has to be set to off, off, off, off.

On the right hand side the microcontroller runs in the normal operation mode of the microcontroller, too. The code of the application is linked to the external SRAM. The kernel of the SWB Monitor Debugger is located in the internal FLASH.

The size of the internal flash and RAM depends on the specific device, in this case of the MB91465P. Hence, the required SRAM of the SWB Monitor Debugger for the application code depends on the specific device, too. Since the SWB Monitor Debugger uses the internal RAM, too, the linkage of the RAM section has to be adapted for different devices.

5.6 The SWB Monitor Debugger

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


```
          28: void main(void)
          29: {
   008801DA: 9F840003C000        LDI:32    #0003C000,R4
   008801E0: 9F8500000017        LDI:32    #00000017,R5
          30:      __EI();                       // enable interrupts
   008801E6: 9310                ORCCR     #10
          31:      __set_il(31);        // set global int-level
   008801E8: 871F                STILM     #1F
          32:
          33:      PFRN = 0x00;         // setup LED port
   008801EA: C000                LDI:8     #00,R0
   008801EC: 9F8C00000417        LDI:32    #00000417,R12
   008801F2: 16C0                STB       R0,@R12
          34:      DDRN = 0x30;
   008801F4: C300                LDI:8     #30,R0
   008801F6: 9F8C00000407        LDI:32    #00000407,R12
   008801FC: 16C0                STB       R0,@R12
          35:
          36:      while(1)                       // toggle LEDs
          37:      {
          38:         for(i=0; i<100000;i++);
   008801FE: C000                LDI:8     #00,R0
   00880200: 1440                ST        R0,@R4
```

Figure 5-10: 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!

5.6.1 Basic Debugger Features

In the following the basic features of the SWB Monitor Debugger are described.

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:
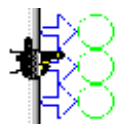
ABORT: Forcibly terminates execution. This button is not fully supported by the SWB Monitor Debugger and may cause malfunction if used to abort "continuous operation" of the MCU.

This command button can only be used to abort single code line operations
Example: for(k=DELAY_CONSTANT; k>0; k--);

Use the **button NMIX on the "SK-91F460P-176PMC"** for **ABORT** function.
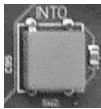
5.7 Advanced Softune Workbench SWB Monitor Debugger Features

In the following advanced features of the SWB Monitor Debugger are described.

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.

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.

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.

6 Programming the internal Flash

All Flash devices have an internal bootloader for asynchronous as well as synchronous Flash-programming:

▸ Asynchronous serial Flash-programming via X2 (UART 4)

▸ Synchronous serial Flash-programming via X1

6.1 Asynchronous Mode

This chapter describes the serial asynchronous programming of the internal flash with the help of the 'FME-FR Programmer' in automatic mode. For other programming possibilities, please see the application note mcu-an-300012-e-mb91460_flash_programming.pdf.

1) For serial asynchronous programming SUB-D9 connector X2 which is connected to UART4 has to be used. The following jumper setting is needed.

Jumper	Setting	Description
JP21 (RXD)	1-2	SIN4 is connected to RS232 transceiver
JP25 (TXD)	1-2	SOT4 is connected to RS232 transceiver
JP27 (RS232/LIN)	1-2	X2 Pin2 is connected to RS232 transceiver (T1out)

Table 6-1: Jumper Settings for Programming

2) If the programming software should generate a reset signal on RTS or DTR line, refer to the following jumper list.

Jumper	Setting	Description
JP20 (RTS-CTS)	Closed	RTS and CTS of X2 are connected
	Open	RTS and CTS of X2 are not connected
JP89 (DTR/RTS)	1-2	DTR signal (Pin 6 of X2) is used as reset source
	2-3	RTS signal (Pin 7 of X2) is used as reset source
JP16 (RST 0/4)	Closed	LIN-UART 4 (X2) connected to reset circuit
JP13 (DTR / DTRx)	1-2	No negation for the DTR/RTS signal
	2-3	DTR/RTS signal is negated
JP38 (RST MCU)	Closed	External reset generation is active

Table 6-2: Jumper Settings for reset signal

3) The CPU Mode selection must be set to internal reset vector mode by setting DIP switch S1 to (Off, Off, Off, Off) (only if DIP Switch mounted!)

4) Start the 'FME-FR Programmer' and connect the UART4 to your serial PC communication port. A straight cable connection has to be used.

5) Choose the microcontroller device ("MB91F465P") as device type and browse to the mhx-file you want to program to the embedded flash memory of the microcontroller. Press the button "Automatic Mode" to start programming.

6) After that, immediately reset the microcontroller.
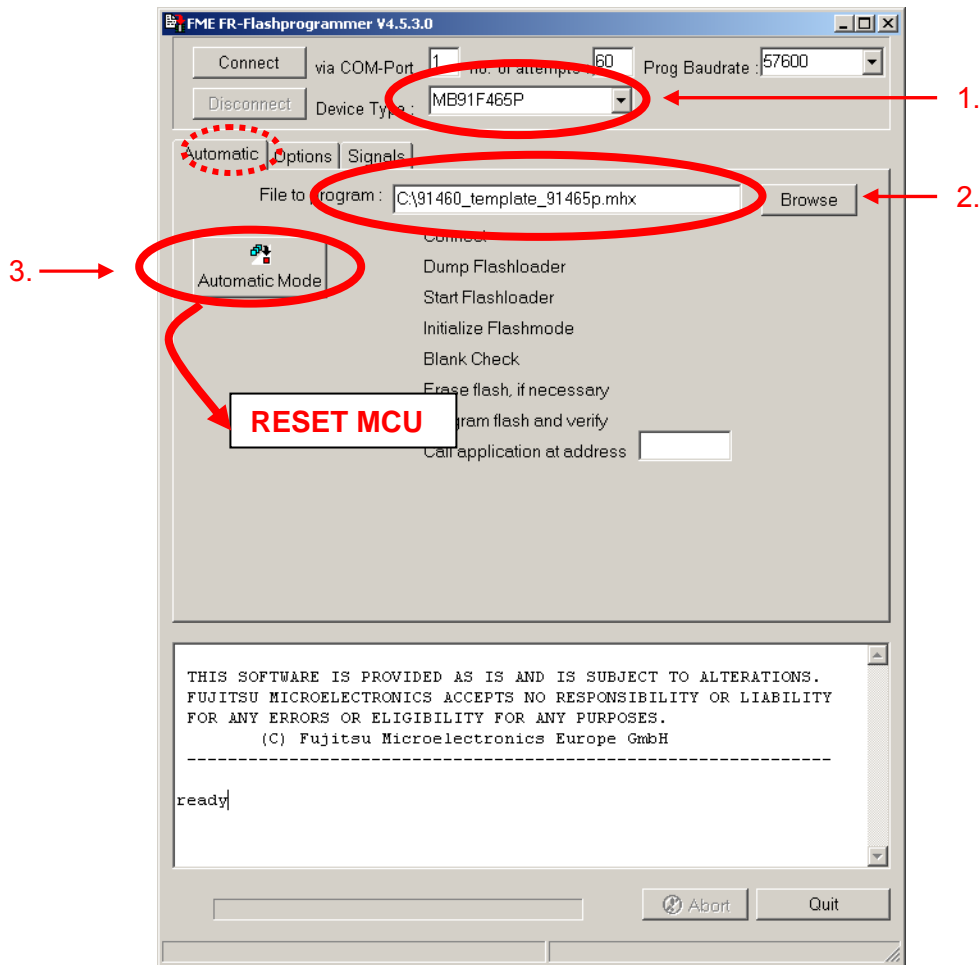
Figure 6-1: Settings for Automatic Mode

7) If your hardware supports MCU reset via DTR or RTS signal lines of the PC's COM port, go to sheet "Signals" and select the appropriate reset signal line. The programmer now resets the microcontroller automatically before starting the programming sequence and you do not have to reset the MCU manually immediately after pressing the button "Automatic Mode".
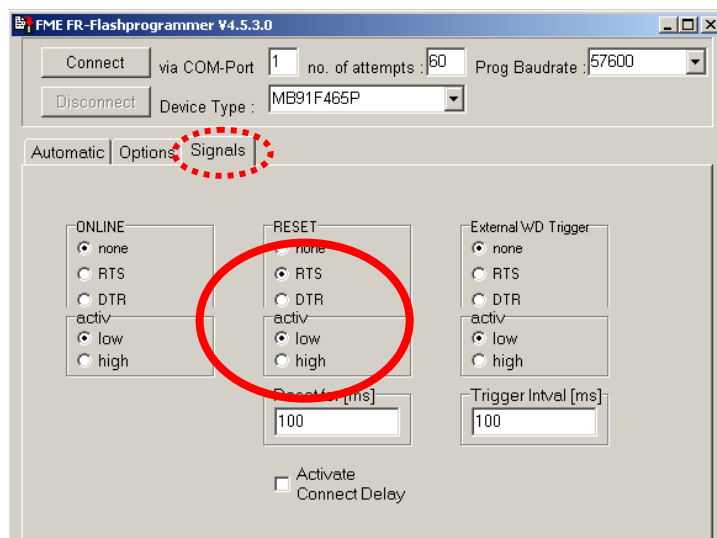
Figure 6-2: Reset Signal Settings

The software now automatically writes a flash loader to the microcontroller, does a blank check and erases the flash memory if necessary. After that the user application is programmed to the embedded flash memory.

During this sequence the actual progress status can be seen by checked off steps on the "Automatic" sheet and the messages below in the message box.

If the programming sequence is ended successfully, you will receive the output shown in figure 7.

The Addresses 0x14:8000 to 0x14:800F cannot be written in automatic mode. This is a built in security feature to prevent unintended writing to the Flash Security Vectors and Boot Security Vectors. Otherwise it would be possible to secure flash content and to lock the microcontroller for serial programming/reading/erasing by not being able to enter the internal loader anymore. If you want to use these features explicitly, you have to program these dedicated addresses by using the "Prog Word" Function of the step-by-step mode.
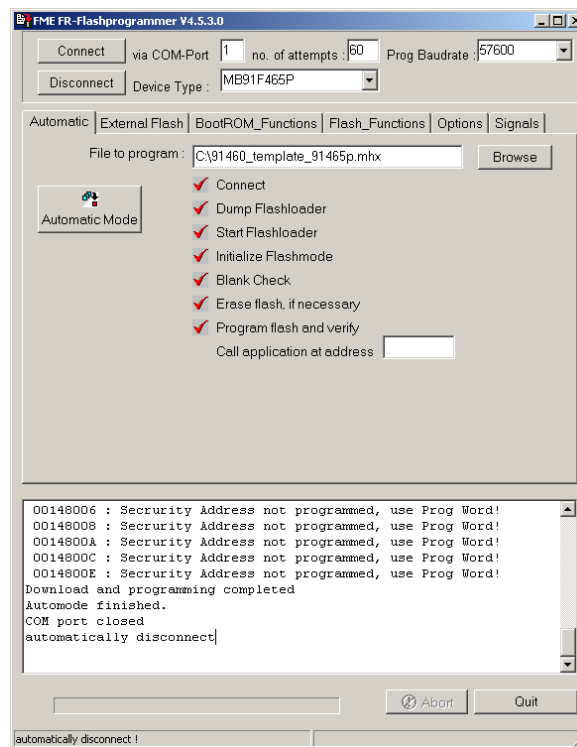
Figure 6-3: Successful Programming in Automatic Mode

6.2   Synchronous Mode

In order to program the Flash-ROM synchronously special third-party soft- and hardware has to be used, e.g. GALEP-4 from www.conitec.de. This tool is not available for free.

A dedicated Flash programming socket (X1) is provided on the evaluation-board for direct connection to this programmer.
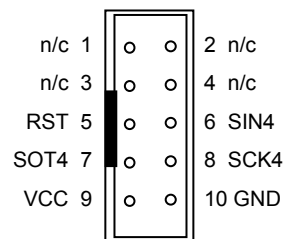
Figure 6-4: X1 Flash Programming Socket

7 Programming SWB Monitor Debugger

To program Softune Workbench Monitor Debbuger to external Flash, get the latest Monitor Debugger Project (91460_mondeb_uart4_mb91465p_extbus_flash_sram_16bit) from our website.

1) The CPU mode selection must be set to internal reset vector mode by setting DIP switch S1 to (Off, Off, Off, Off)

2) Follow steps 1 to 4 of chapter 6.1

3) Choose MB91F465P as device type and browse to the mhx-file you want to program to the external flash memory.

4) If available select tab 'External Flash', Make sure, that 'Enable ext. Flash' is NOT selected.
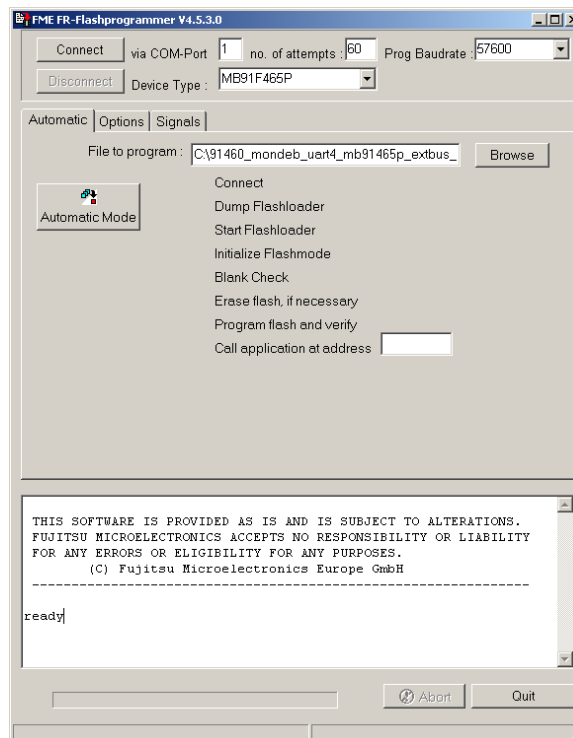
Figure 7-1: Settings for SWB Monitor Debugger programming

5) Select tab "Automatic" and press the button "Automatic Mode" to start programming.

6) After that, immediately reset the microcontroller.

7) After programming is finished - reset the microcontroller

8) LED D1 and D3 should be on after reset.

9) Press SW3 (ATGX) -> Now only LED D2 is on; DO NOT TURN OFF POWER NOW

10) After a short while (~ 60 sec.) LED D0…D3 start flashing

11) Now the Monitor Debugger programming to external Flash-Rom is finished

12) Set S1 to 'on – off – off – off' and reset the microcontroller

13) The correct start-up of the SWB Monitor Debugger is indicated by LD0, LD2 and LD3.

14) Now connection to Softune Workbench can be started.

8 Troubleshooting

Problem	Solution
Starter-kit does not power-on correctly. Some or all of the power LEDs (D20, D28, D32, D36) do not light.	Check voltage supply and jumpers : ∗ The voltage on DC-Power plug should be in the range of 9-12V. The plug should have '+' on shield and '–' on the centre. The power supply should be able to supply at least 500 mA. To support both, the "SK-91F460P-176PMC" 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 millimetre to see whether 5V,3.3V, 2.5V and 1.8V are present on the Vcc/GND test measurement points J9, J10, J13 and J14. ∗ Check jumper positions according to chapter 2.2. ∗ Remove all user extensions of the starter-kit
Starter-kit stays in reset. LED D10 (Reset indicator) lights up permanently.	∗ LED D10 (Reset) should flash. If D10 lights up permanently, the voltage is too low. 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).	∗ Make sure your COM-port number and the baud rate settings are correct (see debugger introduction). The default COM-port is 1 and the default baud rate is 57.6 kBaud. ∗ 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 JP21, 25, 20 are closed). * Use only a RS232 extension 1:1 cable.
The windows of the debugger are empty.	* This is due to 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" !	* 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, UART are not working.	∗ Make sure you have enabled these units on the evaluation board using the appropriate jumpers. See the provided examples and the hardware manual for information on how to control the peripherals.
Optional LC-Display at J1 does not work	∗ Adjust the potentiometer RP1 to adjust contrast of LCD. Remove resistor network RN1.
Graphic sub-board does not work	Check power supply 9-12V/1500mA. Reduce external bus clock frequency.

FUJITSU

9 Appendix

9.1 Related Products

▶ SK-91460P-176PMC   Evaluation board for the MB91F465PA MCU
in FPT-176P-M07 package

▶ MB2198-01   Emulator debugger main unit

▶ MB2198-10   DSU cable

     EMA-MB91V460A-002B-80   Adapter board
▶ NQPACK176SD-ND   Socket for package FPT-176P-M07
(Tokyo Eletech Corp)

     http://www.tetc.co.jp/e_products/e_nq_techdrawings.htm

▶ EMA-MB91F465P-NLS-176M07   Emulator probe cable for MB91F465PA MCU

▶ MB91460 Series

     ▶ MB91F465PA-176PMC   Flash MCU

9.2   Information in the WWW

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://mcu.emea.fujitsu.com/

http://www.fujitsu.com/emea/services/microelectronics/displaycontrollers/

9.3   Tables

9.4 Figures

9.5 Abbreviations

ADC   Analogue Digital Converter

CAN   Controller Area Network

CS   Chip Select

FME   Fujitsu Microelectronics Europe GmbH

LIN   Local Interconnect Network

MCU   Microcontroller Unit

PCB   Printed Circuit Board

SRAM   Static Random Access Memory

UART   Universal Asynchronous Receiver Transmitter

10 China-RoHS regulation

Evaluation Board评估板

Emulation Board 仿真板

根据SJ/T11364-2006

《电子信息产品污染控制标识要求》特提供如下有关污染控制方面的信息。

The following product pollution control information is provided according to SJ/T11364-2006 *Marking for Control of Pollution caused by Electronic Information Products.*

1．电子信息产品污染控制标志说明 Explanation of Pollution Control Label

该标志表明本产品含有超过中国标准SJ/T11363-2006

《电子信息产品中有毒有害物质的限量要求》中限量的有毒有害物质。标志中的数字为本产品的环保使用期，表明本产品在正常使用的条件下，有毒有害物质不会发生外泄或突变，用户使用本产品不会对环境造成严重污染或对其人身、财产造成严重损害的期限，单位为年。

为保证所申明的环保使用期限，应按产品手册中所规定的环境条件和方法进行正常使用，并严格遵守产品维修手册中规定的定期维修和保养要求。

产品中的消耗件和某些零部件可能有其单独的环保使用期限标志，并且其环保使用期限有可能比整个产品本身的环保使用期限短。应到期按产品维修程序更换那些消耗件和零部件，以保证所申明的整个产品的环保使用期限。

本产品在使用寿命结束时不可作为普通生活垃圾处理，应被单独收集妥善处理。

请注意：环保使用期限50年的指定不是与产品的耐久力，使用期限或任何担保要求等同的。

This symbol to be added to all EIO sold to China, indicates the product contains hazardous materials in excess of the limits established by the Chinese standard SJ/T11363-2006 *Requirements for Concentration Limits for Certain Hazardous Substances in Electronic Information Products.* The number in the symbol is the Environment-friendly Use Period (EFUP), which indicates the period, starting from the manufacturing date, during which the toxic or hazardous substances or elements contained in electronic information products will not leak or mutate under normal operating conditions so that the use of such electronic information products will not result in any severe environmental pollution, any bodily injury or damage to any assets, the unit of the period is "Year".

In order to maintain the declared EFUP, the product shall be operated normally according to the instructions and environmental conditions as defined in the product manual, and periodic maintenance schedules specified in Product Maintenance Procedures shall be followed strictly.

Consumables or certain parts may have their own label with an EFUP value less than the product. Periodic replacement of those consumables or parts to maintain the declared EFUP shall be done in accordance with the Product Maintenance Procedures.

This product must not be disposed of as unsorted municipal waste, and must be collected separately and handled properly after decommissioning.

Please note: The designation of 10 years EFUP is not to be equated with the durability, use-duration or any warranty-claims of the product.

产品中有毒有害物质或元素的名称及含量

Table of hazardous substances name and concentration

部件名称 Component Name	有毒有害物质或元素 Hazardous substances name					
	铅 (Pb)	汞 (Hg)	镉 (Cd)	六价铬 (Cr(VI))	多溴联苯 (PBB)	多溴二苯醚 (PBDE)
SK-91460P-176PMC	x	o	o	o	o	o

O: 表示该有毒有害物质在该部件所有均质材料中的含量均在SJ/T11363-2006 标准规定的限量要求以下

X: 表示该有毒有害物质至少在该部件的某一均质材料中的含量超出SJ/T11363-2006 标准规定的限量要求

- 此表所列数据为发布时所能获得的最佳信息
- 由于缺少经济上或技术上合理可行的替代物质或方案，此医疗设备运用以上一些有毒有害物质来实现设备的预期临床功能，或给人员或环境提供更好的保护效果。

O: Indicates that this toxic or hazardous substance contained in all of the homogeneous materials for this part is below the limit requirement in SJ/T11363-2006.

X: Indicates that this toxic or hazardous substance contained in at least one of the homogeneous materials used for this part is above the limit requirement in SJ/T11363-2006.

- Data listed in the table represents best information available at the time of publication

11 Recycling

Gültig für EU-Länder:

Gemäß der Europäischen WEEE-Richtlinie und deren Umsetzung in landesspezifische Gesetze nehmen wir dieses Gerät wieder zurück.

Zur Entsorgung schicken Sie das Gerät bitte an die folgende Adresse:

Fujitsu Microelectronics Europe GmbH

Warehouse/Disposal

Monzastraße 4a

63225 Langen

Valid for European Union Countries:

According to the European WEEE-Directive and its implementation into national laws we take this device back.

For disposal please send the device to the following address:

Fujitsu Microelectronics Europe GmbH

Warehouse/Disposal

Monzastraße 4a

63225 Langen