Fujitsu Microelectronics Europe
User Guide

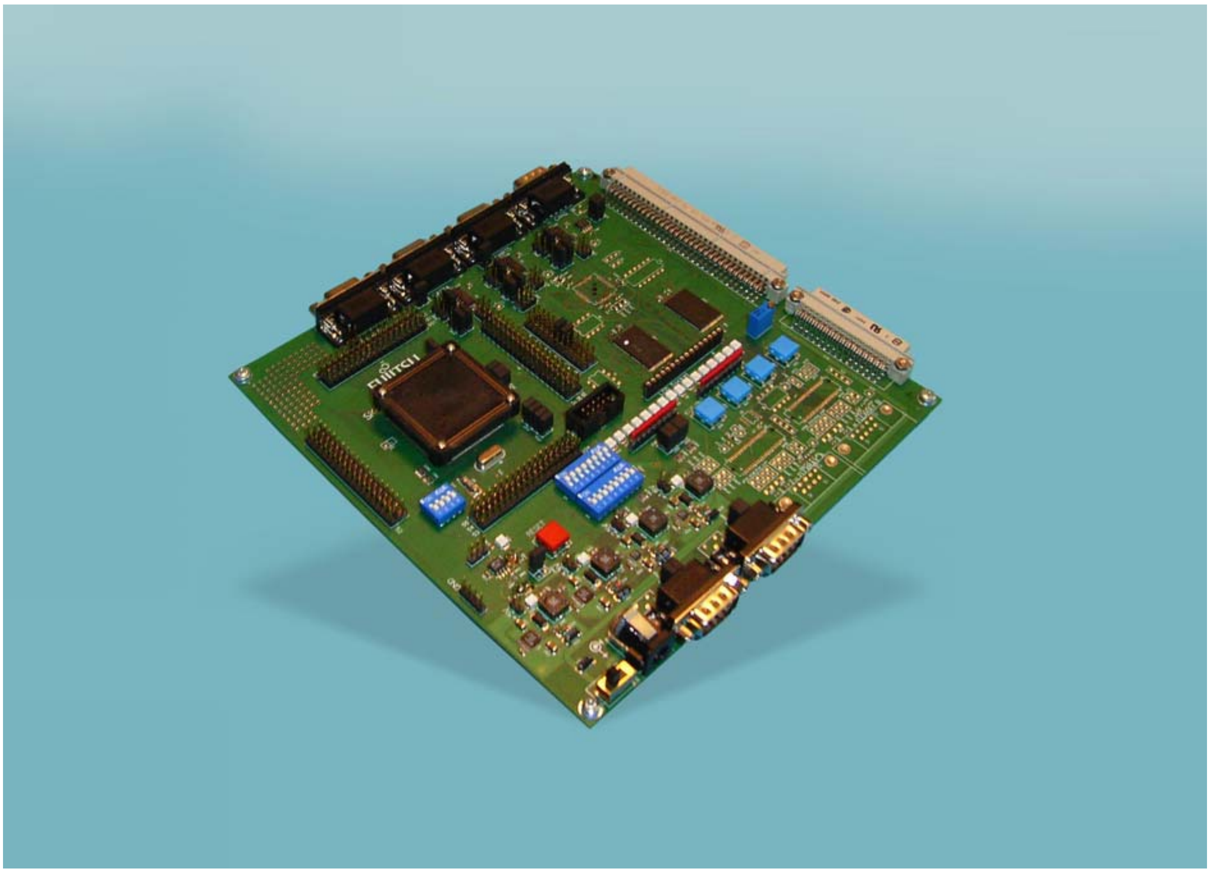
FMEMCU-UG-910014-13

MB91460 SERIES

EVALUATION BOARD

SK-91F467D-208PFV

USER GUIDE



FUJITSU

Revision History

Date	Issue
22.11.2005	V1.0, UMa, first official version
28.03.2006	V1.1, UMa Updated the description of the SWB Monitor Debugger, which is now located in the external flash.
13.04.2006	V1.2, UMa Added chapter “Internal Flash Programming”
04.06.2006	V1.3, UMa Changed chapter “Programming the Monitor Debugger”.

This document contains 60 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-91F467D-208PFV 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-91F467D-208PFV 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 suppliers´ liability is restricted to intention and gross negligence.

NO LIABILITY FOR CONSEQUENTIAL DAMAGES

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

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

Contents

1 Introduction

1.1 Abstract

The SK-91F467D-208PFV is a multifunctional development board for the FUJITSU 32-bit flash microcontroller MB91460 series.

It can be used stand-alone for software development and testing or together with the Softune Workbench Monitor Debugger.

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

1.2 Features

- Supports 32-bit flash microcontroller MB91F467D
- 9-12 V (2000 mA) unregulated external DC power supply
- 5 V, 3.3 V, 2.5 V and 1.8 V on-board switching regulators with power status LEDs
- Triple supply monitor to watch 5 V, 3.3 V and selectable 2.5 V or 1.8 V
- On-board Memory:
    - 32 Mbit (4 MByte) SRAM
    - 128 Mbit (16 MByte) flash
- All microcontroller resources available for evaluation
- All microcontroller pins routed to pin header
- In-circuit serial flash programming
- Three selectable RS-232 or LIN UART-interfaces
- Three high-speed CAN interfaces
- 16 User LEDs
    - optional: alphanumeric standard LC-Display connectable
- Reset button, reset LED
- 4 User buttons, one 8 bit MCU port can be connected to DIP switches
- External bus interface routed to 96pin and 48pin DIN 41612 (VG) connectors

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

1.3 General Description

The SK-91F467-208PFV supports the FUJITSU 32-bit flash microcontroller MB91F467D.

The starter-kit can be used as a stand-alone development platform, or instead of the flash MCU MB91F467D, with the emulation system MB2198-01 and the adapter board (MB2198-300) and the 208 pin probe cable board (PB-91467D-208PFV or PB-91467D-LS-208PFV).

This User Guide is describing PCB version V1.1. The PCB version is printed at the TOP side of the starter-kit close to MB91F467D MCU [U1].

1.3.1 On Board Memory

SK-91F467D-208PFV: The starter-kit SK-91F467D-208PFV includes 32 Mbit (4 MByte) SRAM memory and 128 Mbit (16 MByte) flash memory on board. The flash can be used with 3.3 V external bus supply voltage only (JP21: 2-3).

1.3.2 MCU Clocks

The board is supplied with a 4 MHz crystal as the main oscillation clock for the MCU. Using the internal PLL of the MCU, internal clock rates up to 96 MHz can be achieved. The MCU sub clock is connected to a 32.768 kHz crystal.

1.3.3 MCU Operating Mode

The operating mode of the microcontroller can be selected with the DIP-switch S5.

1.3.4 External Bus interface Connectors

Via DIN 41612 VG96ABC and VG48ABC connectors (external bus interface of the MCU), it is possible to connect other devices e.g. user applications or Fujitsu graphic device sub boards.

1.3.5 CAN Bus

Three high-speed CAN-transceivers (PCA82C250) are available to connect all available on-chip CAN-controllers to 9-pin D-Sub connectors (X9, X10 and X11). The transceivers provides differential transmit and receive capability between CAN-controller and CAN-bus.

1.3.6 RS-232 and LIN

Three separate RS-232 transceivers and three separate LIN transceivers are available to connect the on-chip USARTs to the 9-pin D-Sub connectors.

The RS-232 transceivers generate the adequate RS-232 levels for receive (RXD) and transmit (TXD) lines. The RTS signal can be shortcut to CTS using jumpers (some PC software needs this connection).

Either the DTR line or the RTS line of X1, X4 or X8 connectors can be selected to generate a system reset.

The LIN transceivers (TLE6259) generate the adequate levels to drive the bus line in LIN-systems for automotive and industrial applications.

1.3.7 MCU Pins

All 208 pins of the microcontroller are connected to the edge connectors J1, J2, J3 and J4 and are directly available to the user.

1.3.8 User Buttons

There are four push buttons on board, which can be connected to input-ports of the microcontroller. Some ports may support additional functions like external interrupts (INT0, INT1 and INT2) and trigger for the Reload Timer or for the Input-Capture Unit (TIN0 / ICU0). One additional button is reserved as system-reset-button to reset the MCU, but it is possible to disconnect the RST-port with JP54.

It is possible to connect with DIP-switch S3 eight port input switches at S4 to the MCU port P26. All eight ports can be switched separately to use e.g. only four ports as input.

1.3.9 User LEDs and Optional LCD

Sixteen user-LEDs are connected via two pull-down resistor networks to port P25 and to port P16. Parallel to the LEDs on port P25, the connector J6 can be used to connect a standard alphanumeric display. The resistor networks RN2 and RN3 can be removed to free the ports.

1.3.10 Power Supply

The four on-board step-down switching regulators allow the user to connect an unregulated DC input voltage between +9 V to +12 V (max. 2000 mA) to the starter-kit. The switching regulators provide the voltages of 5 V(3 A), 3.3 V(3 A), 2.5 V(1.5 A) and 1.8 V(1.5 A) on the starter-kit. These voltages give also the possibility to connect an optional graphic-controller starter-kit to the board. The switching regulators are short circuit protected and provide a thermal shutdown.

2 Installation

Remove the board carefully from the shipping carton.

First check whether there are any damages before power on the starter-kit.

To supply the starter-kit use the delivered plug-in AC-DC Adapter (UPA 2000). Select the AC plug adapter suitable for your country power sockets

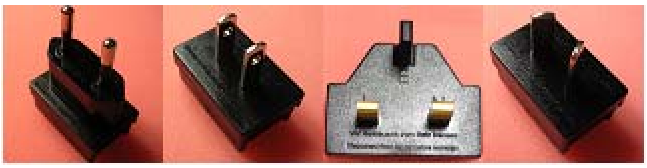


Figure 2-1 AC plug adapter

and insert this adapter in the connection terminal on the AC-DC adapter.

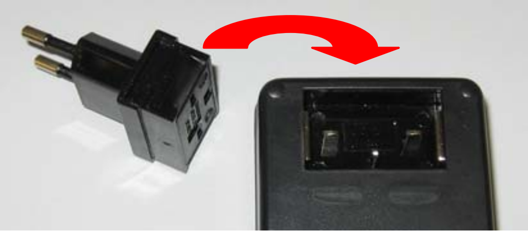
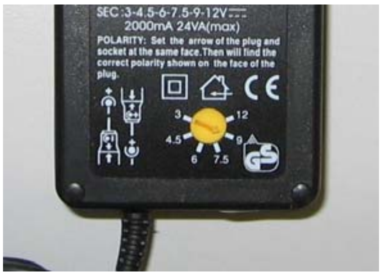


Figure 2-2: AC plug adapter selection

Check the selected output DC voltage of the AC-DC plug-in adapter at the voltage selection key. It should be 9 V! Change the output voltage only in a powered-down condition!



SEC:3-4.5-6-7.5-9-12V
2000mA 24VA(max)
POLARITY: Set the arrow of the plug and socket at the same face.Then will find the correct polarity shown on the face of the plug.
3
12
4.5
9
6
7.5

Figure 2-3: DC voltage selection

Select the low voltage adapter suitable to the power connector X5 at the SK-91F467D-208PFV and plug it to the connecting socket in the right orientation ('+' connected to shield and 'GND' connected to centre pin).

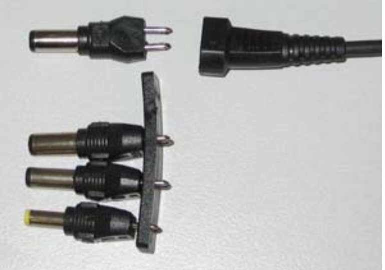


Figure 2-4: Low voltage adapter selection

Now connect the DC low voltage adapter to the starter-kit and plug in the plug-in power supply to a power socket. The SK-91F467D-208PFV can be turned on with the on board power switch S8 (or S9).

For the power supply of the SK-91F467D-208PFV a DC input voltage of 9V – 12V is recommended. The positive voltage (+) must be connected to the shield, and ground (GND) must be connected to the centre of the connector X5!

After power-on of the SK-91F467D-208PFV, the four red power-on LEDs D24 (5 V), D30 (3.3 V), D34 (2.5 V) and D40 (1.8 V) should be light. The reset LED D18 should be off.

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 D1, D4 and D8 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 S5 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 S5 to (Off, Off, Off, Off)

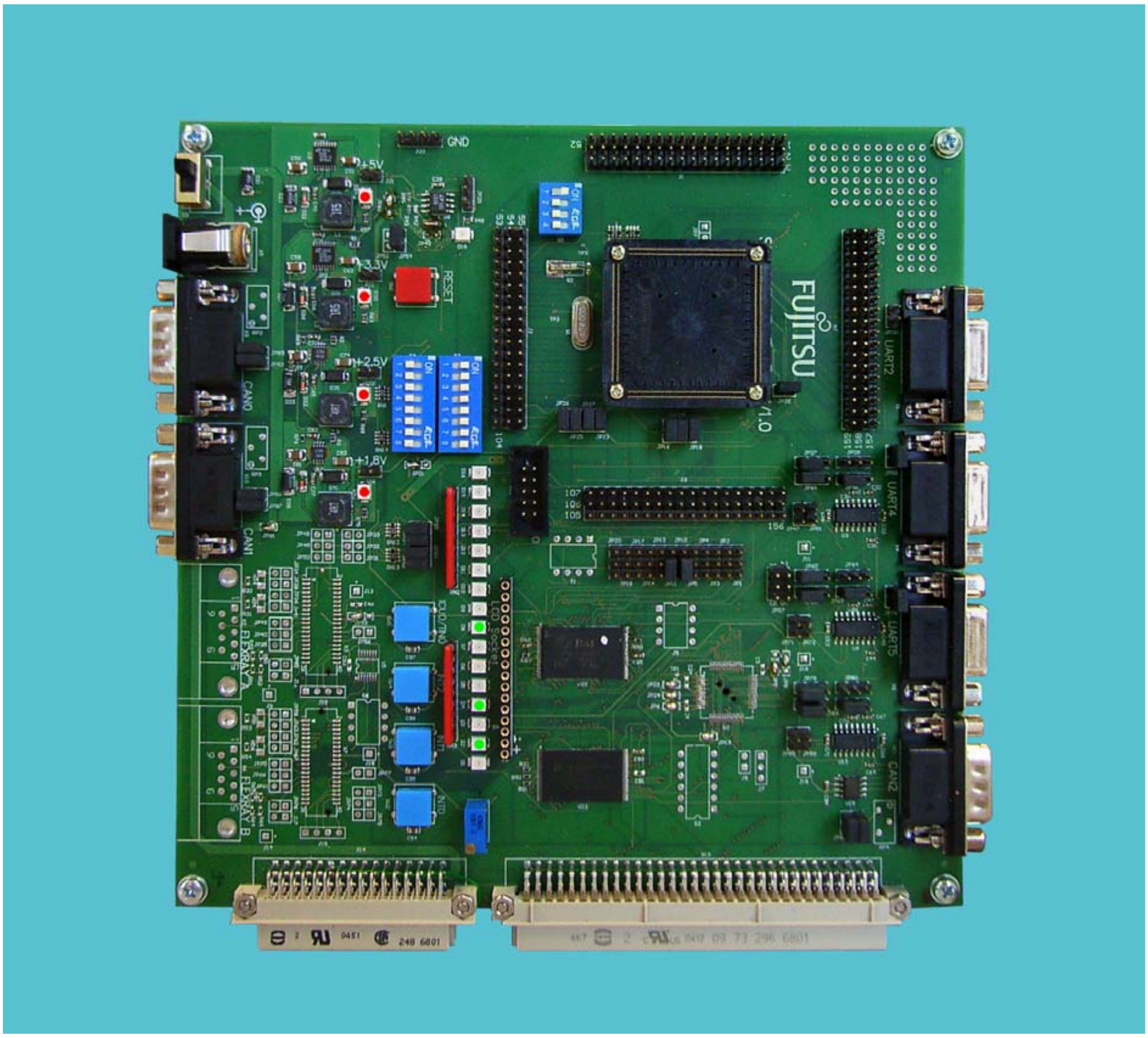


Figure 2-5: Starter-kit status after power on

2.1 Jumper Settings

The following table lists all jumpers including its default setting and location on the starter-kit.

Jumper	Description / Function	Type	Default	Layout Coordinates	Schematic Page
JP15	MCUVcc - AVcc	Jumper 2 pin	Closed	E 9	1 / 3
JP16	AVcc - AVRH	Jumper 2 pin	Closed	E 9	1 / 3
JP18	GND - AVss	Jumper 2 pin	Closed	E 8	1 / 3
JP21	VDD35 MCUVcc / 3.3V	Solder Jumper 3 pad	2-3	F 15	1 / 3
JP23	MCUVcc – HVDD5	Jumper 2 pin	Closed	E 10	1 / 3
JP25	MCUVcc – VDD5R	Jumper 2 pin	Closed	E 11	1 / 3
JP26	Vcc - MCUVcc	Jumper 2 pin	Closed	E 11	1 / 3
JP27	MCUVcc – VDD5	Jumper 2 pin	Closed	E 11	1 / 3
JP28	GND – HVSS5	Jumper 2 pin	Closed	E 6	1 / 3
JP29	DTR select DTR / /DTR	Jumper 3 pin	Open	B 14	2 / 3
JP30	UART2 RTS - CTS	Jumper 2 pin	Open	D 3	2 / 3
JP37	UART2 RxD RS232 / LIN	Jumper 3 pin	2-3	F 5	2 / 3
JP39	UART2 DTR / RTS	Jumper 3 pin	Open	F 4	2 / 3
JP42	Watch 1.8V	Solder Jumper 2 pad	Open	B 15	2 / 3
JP43	Watch 2.5V	Solder Jumper 2 pad	Closed	B 15	2 / 3
JP44	UART2 TxD RS232 / LIN	Jumper 3 pin	2-3	F 5	2 / 3
JP46	UART2 RS232 / LIN	Jumper 3 pin	2-3	F 4	2 / 3
JP47	UART2 LIN Enable	Jumper 2 pin	Closed	G 6	2 / 3
JP51	UART2 LIN Master Yes / No	Jumper 2 pin	Closed	G 5	2 / 3
JP54	RESET - MCU RST	Jumper 2 pin	Closed	B 15	2 / 3
JP57	UARTX RST Select	Jumper 6 pin	Open	H 6	2 / 3
JP58	UART4 RTS - CTS	Jumper 2 pin	Closed	F 3	2 / 3
JP60	UART4 RxD RS232 / LIN	Jumper 3 pin	1-2	H 5	2 / 3
JP64	UART4 DTR / RTS	Jumper 3 pin	Open	H 4	2 / 3
JP65	UART4 TxD RS232 / LIN	Jumper 3 pin	1-2	H 5	2 / 3
JP68	Vcc 5V / 3.3V	Solder Jumper 3 pad	1-2	B 16	2 / 3
JP69	UART4 RS232 / LIN	Jumper 3 pin	1-2	H 4	2 / 3
JP71	UART4 LIN Enable	Jumper 2 pin	Open	H 6	2 / 3
JP72	UART4 LIN Master Yes / No	Jumper 2 pin	Open	H 6	2 / 3
JP78	UART5 RTS - CTS	Jumper 2 pin	Closed	H 3	2 / 3
JP79	UART5 RxD RS232 / LIN	Jumper 3 pin	1-2	J 5	2 / 3
JP80	UART5 DTR / RTS	Jumper 3 pin	Open	J 4	2 / 3
JP81	UART5 TxD RS232 / LIN	Jumper 3 pin	1-2	K 5	2 / 3
JP82	CAN0 RxD	Jumper 2 pin	Closed	D 19	2 / 3
JP83	CAN0 TxD	Jumper 2 pin	Closed	D 19	2 / 3
JP84	UART5 RS232 / LIN	Jumper 3 pin	1-2	K 4	2 / 3
JP85	UART5 LIN Enable	Jumper 2 pin	Open	K 6	2 / 3
JP86	UART5 LIN Master Yes / No	Jumper 2 pin	Open	K 6	2 / 3
JP87	CAN1 RxD	Jumper 2 pin	Closed	G 19	2 / 3
JP88	CAN1 TxD	Jumper 2 pin	Closed	F 19	2 / 3
JP89	CAN2 RxD	Jumper 2 pin	Closed	M 4	2 / 3
JP90	CAN2 TxD	Jumper 2 pin	Closed	M 4	2 / 3
JP91	Vin – Ext Voltage	Solder Jumper 2 pad	Closed	G 18	3 / 3
JP92	INT0	Jumper 2 pin	Closed	G 15	3 / 3
JP93	INT1	Jumper 2 pin	Closed	G 15	3 / 3
JP94	INT2	Jumper 2 pin	Closed	G 15	3 / 3
JP95	ICU0/TIN0	Jumper 2 pin	Closed	G 15	3 / 3
JP96	Direct RESET	Solder Jumper 3 pad	1-2	A 14	2 / 3
JP97	VCC1V8 to X14B2	Solder Jumper 2 pad	Open	H 16	3 / 3

Table 2-1: Jumper settings

2.2 Jumper Location

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

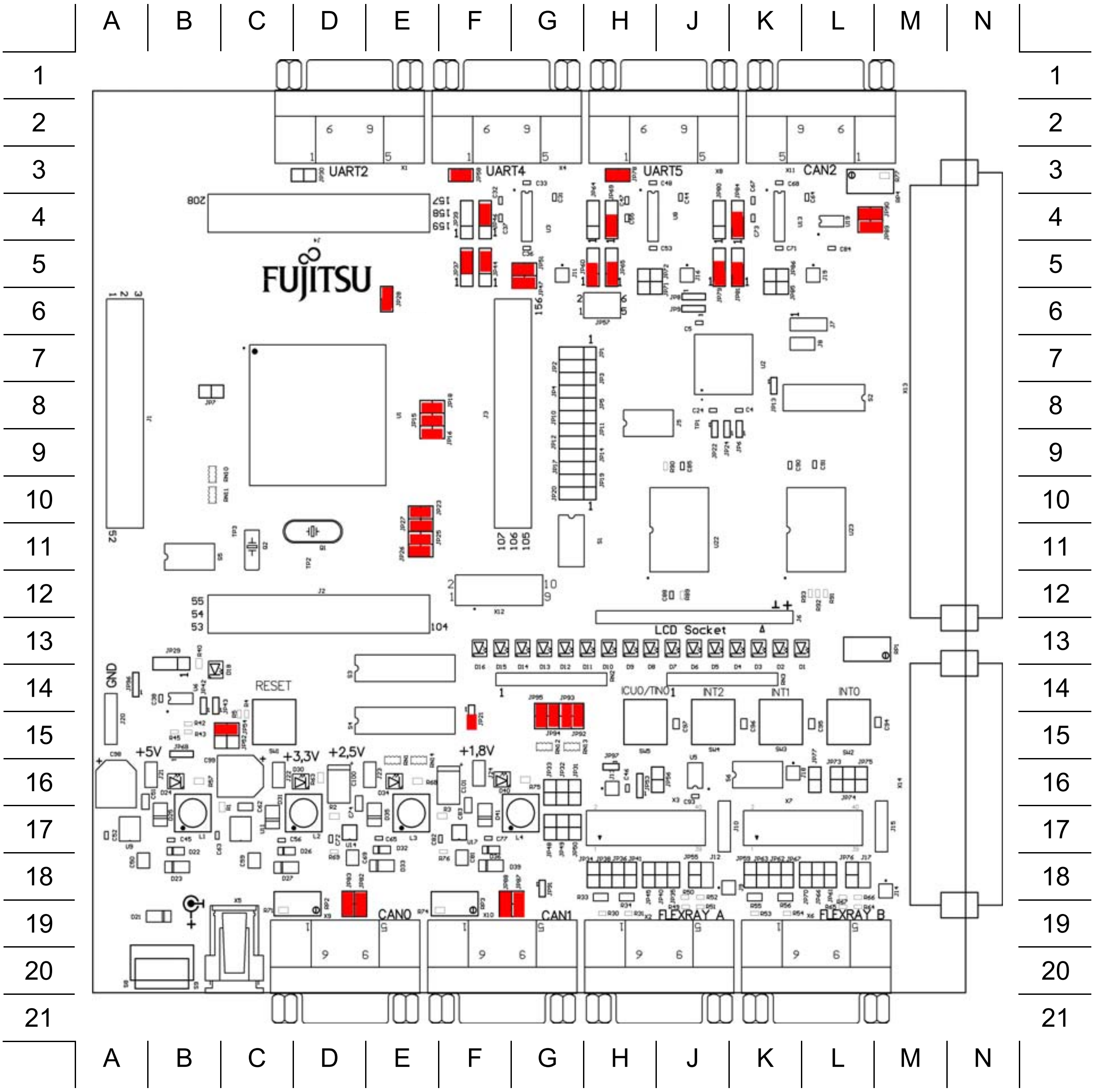


Figure 2-6: Default jumper settings

2.3 Software Installation

2.3.1 Installation of Softune Workbench

With the “SK-91F467D-208PFV” Fujitsu supplies a full working development environment called Softune Workbench V6. The Softune Workbench also supports the SWB Monitor Debugger which is pre-programmed into the external flash memory of the “SK-91F467D-208PFV”. To develop own software and to work with the SWB Monitor Debugger of the “SK-91F467D-208PFV” 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 starter-kit CD-ROM into the directory Software\SWB_Install and start the FRSTA*.exe.

3) Follow the installation instructions.

4) For the default installation path it is recommended to use c:\Softune32.

5) After the installation is finished. The FR Family Softune Workbench can be started via the Windows “Start” menu.

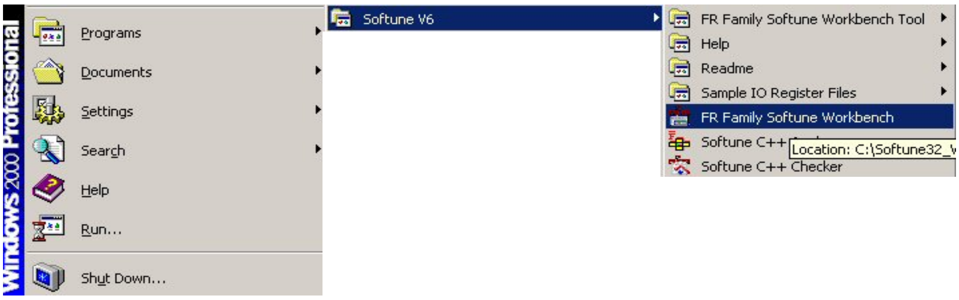


Figure 2-7: Softune Workbench start menu location

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

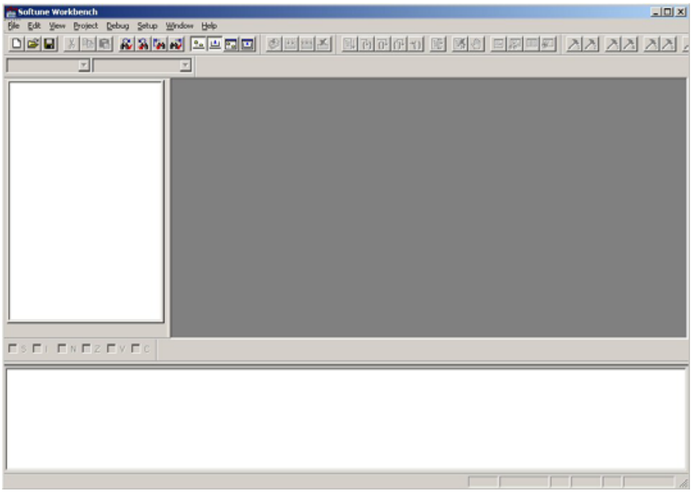


Figure 2-8: Softune Workbench V6 IDE

3 Jumpers and Switches

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

3.1 MCU Operating-Mode (S5)

The DIP-switch S5 is used to set the operating mode of the MCU. 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.

DIP switch	Setting	Logical value
S5/1 (MD0)	ON (closed)	1 (high)
	OFF (open)	0 (low)
S5/2 (MD1)	ON (closed)	1 (high)
	OFF (open)	0 (low)
S5/3 (MD2)	ON (closed)	1 (high)
	OFF (open)	0 (low)
S5/4 (Not used)	ON (closed)	Not connected
	OFF (open)	Not connected
Default: grey		

Table 3-1: MCU operating mode

By default, the “External ROM Mode Vector” is selected for the SWB Monitor Debugger, which is pre-programmed into the external flash. In this mode the internal flash is inaccessible.

DIP-Switch S5

(default setting)

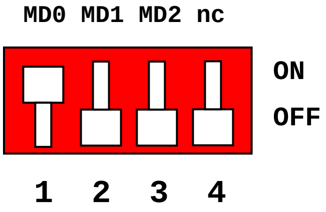


Figure 3-1: MCU mode switch: External ROM Mode Vector

The “Internal ROM Mode Vector” of MB91F467D is normally required for the final application, which should be stored in the internal flash. In this mode the internal flash is accessible.

DIP-Switch S5

(default setting)

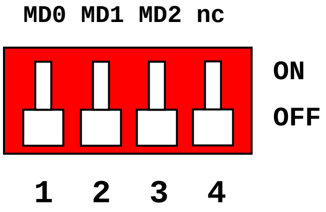


Figure 3-2: MCU mode switch: Internal ROM Mode Vector

3.2 Power Supply

There are four on-board switching regulators to provide the voltages 5 V, 3.3 V, 2.5 V and 1.8 V on the starter-kit. With the power ON/OFF-switch S8 or S9 (S9 is a soldering option to use a different switch), the main input voltage from DC-connector X5 will be connected to the Vin voltage of the board. The Vin voltage supplies the switching regulators.

Switch	Setting	Description
S8 or S9	ON (1-2)	Power ON
	OFF (2-3)	Power OFF
Default: grey		

Table 3-2: Power switch

With JP91 the pins B14, B15 and B16 of the external bus interface connector X14 can be connected to the Vin voltage, so that it is possible to supply the board from an external connected board.

There is a triple supply monitor on-board, which monitors three of the four DC output voltages and generates a system reset, in case with wrong levels of the on board voltages. 5 V and 3.3 V are always monitored and the third monitored voltage can be selected with the solder jumpers JP42 and JP43.

With JP68 it is possible to select the whole board supply voltage Vcc to 5 V or 3.3 V

Jumper	Setting	Description
JP91	ON (closed)	Vin is connected to X14 pins B14, B15 and B16
	OFF (open)	Vin is not connected to X14 pins B14, B15 and B16
JP42	ON (closed)	Vcc1V8 is connected to supply monitor
	OFF (open)	Vcc1V8 is not connected to supply monitor
JP43	ON (closed)	Vcc2V5 is connected to supply monitor
	OFF (open)	Vcc2V5 is not connected to supply monitor
JP68	1-2	Vcc is connected to 5 V
	2-3	Vcc is connected to 3.3 V
Default: grey		

Table 3-3: Power supply configurations

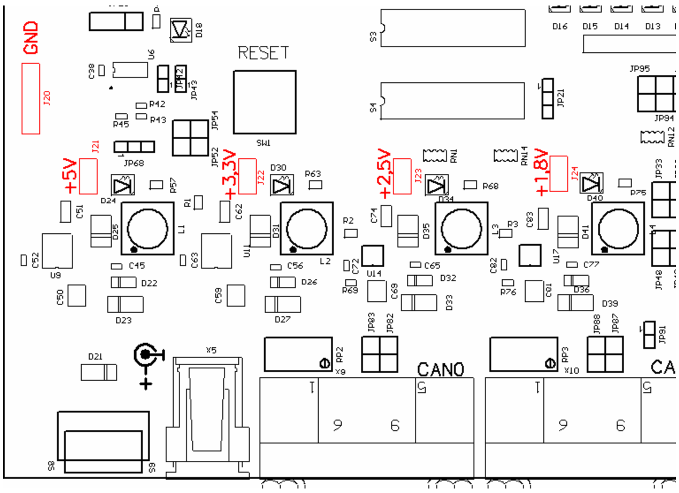


Figure 3-3: Voltage Test Points

Regulator	Test point	Voltage
U9	J21	+5V
U11	J22	+3.3V
U14	J23	+2.5V
U17	J24	+1.8V
-	J20	GND

Table 3-4: Voltage Test Points

3.2.1 MCU Power Supply Voltage

JP21 With this jumper it is possible to connect the MCU external bus interface supply VDD35 to 3.3 V. The MCU MB91460 series supports the function to supply the MCU with 5 V so that the normal I/O ports work with 5 V and supply the external bus interface with 3.3 V.

JP23 This jumper is used to connect the HVDD5 supply to MCUVcc. The HVDD5 is the MCU supply voltage for the high current ports.

JP25 This jumper is used to connect the VDD5R MCU supply pins to MCUVcc.

JP26 This jumper is used to connect the Vcc voltage to MCUVcc.

JP27 This jumper is used to connect the VDD5 MCU supply pins to MCUVcc.

JP28 This jumper is used to connect the HVSS5 MCU pins to GND. HVSS5 are the GND pins for the high current ports.

Note:
JP23, JP25, JP26, JP27 and JP28 can be used for measurements of power consumption of the MCU.

Jumper	Setting	Description
JP21	1-2	VDD35 is connected to MCUVcc
	2-3	VDD35 is connected to 3.3 V
JP23	ON (closed)	HVDD5 is connected to MCUVcc
	OFF (open)	HVDD5 is not connected to MCUVcc
JP25	ON (closed)	VDD5R is connected to MCUVcc
	OFF (open)	VDD5R is not connected to MCUVcc
JP26	ON (closed)	MCUVcc is connected to Vcc
	OFF (open)	MCUVcc is not connected to Vcc
JP27	ON (closed)	VDD5 is connected to MCUVcc
	OFF (open)	VDD5 is not connected to MCUVcc
JP28	ON (closed)	HVSS5 is connected to GND
	OFF (open)	HVSS5 is not connected to GND
Default: grey		

Table 3-5: MCU power supply SK-91F467D-208PFV

3.2.2 MCU Analogue Power Supply Voltage

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

JP15 Connects power supply AVcc of the A/D converter to MCUVcc

JP16 Connects high reference voltage AVRH of the A/D converter to AVcc

JP18 Connects AVss of the A/D converter to GND

Jumper	Setting	Description
JP15	ON (closed)	AVcc is connected to MCUVcc
	OFF (open)	AVcc is disconnected
JP16	ON (closed)	AVRH is connected to AVcc
	OFF (open)	AVRH defined by resistor network[*1]
JP18	ON (closed)	AVss is connected to GND
	OFF (open)	AVss is disconnected
Default: grey		

[*1] By default the resistor network (R10 and R13) is not assembled on the board

Table 3-6: MCU ADC Supply

> **Note:**
> If JP15 or JP18 are open, the user has to supply an adequate analogue voltage supply (AVcc and AVss) to the A/D-converter.
>
> If JP16 is open, the potential divider comprising resistors R10 and R13 define AVRH.

3.3 CAN0 – CAN2

Three high-speed CAN-transceivers (PCA82C250) are on-board to connect all available on-chip CANs to 9-pin D-Sub connectors (X9, X10 and X11).

JP82 Connects MCU RX0 to CAN transceiver U15

JP83 Connects MCU TX0 to CAN transceiver U15

JP87 Connects MCU RX1 to CAN transceiver U18

JP88 Connects MCU TX1 to CAN transceiver U18

JP89 Connects MCU RX2 to CAN transceiver U19

JP90 Connects MCU TX2 to CAN transceiver U19

Jumper	Setting	Description
JP82	ON (closed)	CAN0 RxD connected to MCU
	OFF (open)	No connection to the microcontroller
JP83	ON (closed)	CAN0 TxD connected to MCU
	OFF (open)	No connection to the microcontroller
JP87	ON (closed)	CAN1 RxD connected to MCU
	OFF (open)	No connection to the microcontroller
JP88	ON (closed)	CAN1 TxD connected to MCU
	OFF (open)	No connection to the microcontroller
JP89	ON (closed)	CAN2 RxD connected to MCU
	OFF (open)	No connection to the microcontroller
JP90	ON (closed)	CAN2 TxD connected to MCU
	OFF (open)	No connection to the microcontroller
Default: grey		

Table 3-7: CAN-MCU connection

3.4 LIN / RS-232 UART

There are three UART connectors X1, X4 and X8 (9-pin D-Sub) on-board, which can be used for RS-232 or LIN communication (to get information in detail about pin usage, please refer to chapter 4.2 UART / LIN Connector (X1, X4, X8)). The LIN transceiver needs to be configured as LIN-master or LIN-slave.

3.4.1 UART 2

UART2 (connector X1) is selected as LIN master at default setting.

JP37 Select, whether SIN2 is connected to RS-232- or LIN-transceiver

JP44 Select, whether SOT2 is connected to RS-232- or LIN-transceiver

JP46 Select, whether RS-232- or LIN-transceiver is connected to X1 pin2 (shared pin)

JP30 Some terminal programs needs a connection between CTS and RTS

JP39 Selects, whether DTR or RTS will be used for system reset generation

JP47 Enables the LIN transceiver

JP51 Selects, whether UART2 will be LIN master or LIN slave

Note:
UART2 has to be selected with JP57 to generates a system reset

Jumper	Setting	Description
JP37	1-2	MCU SIN2 connected to RS-232 transceiver
	2-3	MCU SIN2 connected to LIN transceiver
JP44	1-2	MCU SOT2 connected to RS-232 transceiver
	2-3	MCU SOT2 connected to LIN transceiver
JP46	1-2	X1 pin2 is connected to RS-232 transceiver
	2-3	X1 pin2 is connected to LIN transceiver
JP30	ON (closed)	RTS and CTS is shortcut at connector X1
	OFF (open)	RTS and CTS is not shortcut at connector X1
JP39	1-2	DTR can be used for system reset
	2-3	RTS can be used for system reset
JP47	ON (closed)	LIN transceiver is enabled
	OFF (open)	LIN transceiver is not enabled
JP51	ON (closed)	LIN master mode
	OFF (open)	LIN slave mode
Default: grey		

Table 3-8: UART2 settings

3.4.2 UART 4 (Flash Programming)

UART4 (connector X4) is selected as RS-232 at default setting. UART4 is also used to write user software into the flash of the MCU.

JP60 Select, whether SIN4 is connected to RS-232- or LIN-transceiver

JP65 Select, whether SOT4 is connected to RS-232- or LIN-transceiver

JP69 Select, whether RS-232- or LIN-transceiver is connected to X4 pin2 (shared pin)

JP58 Some terminal programs needs a connection between CTS and RTS

JP64 Selects, whether DTR or RTS will be used for system reset generation

JP71 Enables the LIN transceiver

JP72 Selects, whether UART4 will be LIN master or LIN slave

Note:
UART4 has to be selected with JP57 to generates a system reset

Jumper	Setting	Description
JP60	1-2	MCU SIN4 connected to RS-232 transceiver
	2-3	MCU SIN4 connected to LIN transceiver
JP65	1-2	MCU SOT4 connected to RS-232 transceiver
	2-3	MCU SOT4 connected to LIN transceiver
JP69	1-2	X4 pin2 is connected to RS-232 transceiver
	2-3	X4 pin2 is connected to LIN transceiver
JP58	ON (closed)	RTS and CTS is shortcut at connector X4
	OFF (open)	RTS and CTS is not shortcut at connector X4
JP64	1-2	DTR can be used for system reset
	2-3	RTS can be used for system reset
JP71	ON (closed)	LIN transceiver is enabled
	OFF (open)	LIN transceiver is not enabled
JP72	ON (closed)	LIN master mode
	OFF (open)	LIN slave mode
Default: grey		

Table 3-9: UART4 settings

Note:
UART4 is used as serial interface for Softune Workbench Monitor Debugger. Jumper J58 must be closed.

3.4.3 UART 5

UART5 (connector X8) is selected as RS-232 at default setting.

JP79 Select, whether SIN5 is connected to RS-232- or LIN-transceiver

JP81 Select, whether SOT5 is connected to RS-232- or LIN-transceiver

JP84 Select, whether RS-232- or LIN-transceiver is connected to X8 pin2 (shared pin)

JP78 Some terminal programs needs a connection between CTS and RTS

JP80 Selects, whether DTR or RTS will be used for system reset generation

JP85 Enables the LIN transceiver

JP86 Selects, whether UART5 will be LIN master or LIN slave

Note:
UART5 has to be selected with JP57 to generates a system reset

Jumper	Setting	Description
JP79	1-2	MCU SIN4 connected to RS-232 transceiver
	2-3	MCU SIN4 connected to LIN transceiver
JP81	1-2	MCU SOT4 connected to RS-232 transceiver
	2-3	MCU SOT4 connected to LIN transceiver
JP84	1-2	X4 pin2 is connected to RS-232 transceiver
	2-3	X4 pin2 is connected to LIN transceiver
JP78	ON (closed)	RTS and CTS is shortcut at connector X4
	OFF (open)	RTS and CTS is not shortcut at connector X4
JP80	1-2	DTR can be used for system reset
	2-3	RTS can be used for system reset
JP85	ON (closed)	LIN transceiver is enabled
	OFF (open)	LIN transceiver is not enabled
JP86	ON (closed)	LIN master mode
	OFF (open)	LIN slave mode
Default: grey		

Table 3-10: UART5 settings

3.5 User Push Buttons INT0, INT1, INT2 and ICU0/TIN0

There are four user push buttons on the starter-kit, which can be connected to the microcontroller.

JP92 – JP95 Connects the user push buttons (SW2 – SW5) to the microcontroller

Jumper	Setting	Description
JP92 (SW2)	ON (closed)	Button INT0 is connected to the microcontroller
	OFF (open)	No connection to the microcontroller
JP93 (SW3)	ON (closed)	Button INT1 is connected to the microcontroller
	OFF (open)	No connection to the microcontroller
JP94 (SW4)	ON (closed)	Button INT2 is connected to the microcontroller
	OFF (open)	No connection to the microcontroller
JP95 (SW5)	ON (closed)	Button ICU0/TIN0 is connected to the microcontroller
	OFF (open)	No connection to the microcontroller
Default: grey		

Table 3-11: User push buttons

3.6 User DIP Switch 8 Bit (S4)

There is an 8 bit user-DIP-switch on-board, which can be connected to one MCU port (P26) in order to set high- or lowlevel. S3, also an 8 bit DIP-switch, dis- / connects the eight channels of S4 to the MCU pins. In case of other usage of these pins, open S3 switch accordingly.

DIP switch	Setting	Logical value
S4/1 - 8	ON (closed)	0 (low)
	OFF (open)	1 (high)
Default: grey		

Table 3-12: S4 values

Note:
DIP-switch S3 is used to connect DIP-switch S4 to the MCU port pins at port P26.

DIP switch	Setting	Description
S3/1 - 8	ON (closed)	S4/1-8 connected to MCU port P26_0-7
	OFF (open)	No connection to the microcontroller
Default: grey		

Table 3-13: S3 settings

3.7 Reset Generation

A system reset can be generated in different ways. The first possibility is to press the user reset push button (SW1). Secondly, the supply monitor will generate a reset if an unusual voltage is detected and third, a reset is raised if the selected UART generates one.

An active system reset is indicated by the reset LED D18.

The system reset will do a simultaneous reset of the microcontroller. The MCU can be disconnected from the system reset signal.

JP96 The Reset push button can be used in two modes:

- Immediate reset
- Secured Reset (2s)

The selection is done by solder jumper JP96.

JP54 Connects the MCU to the system reset signal

Jumper	Setting	Description
JP54	ON (closed)	Reset is connected to MCU INITX
	OFF (open)	No connection to the MCU
JP96	1-2	Reset is generated immediately when SW1 is pressed
	2-3	Reset is generated after SW1 is pressed 2s
Default: grey		

Table 3-14: Reset connections

The triple supply monitor on-board has the possibility to watch 5V, 3.3V and the third voltage is user configurable 2.5V or 1.8V. If any voltage fails, a system reset will be done.

JP42 Monitoring of 1.8V supply voltage

JP43 Monitoring of 2.5V supply voltage

Jumper	Setting	Description
JP42	ON (closed)	Vcc1V8 connected to supply monitor
	OFF (open)	Vcc1V8 not connected to supply monitor
JP43	ON (closed)	Vcc2V5 connected to supply monitor
	OFF (open)	Vcc2V5 not connected to supply monitor
Default: grey		

Table 3-15: Supply monitor settings

By default 2.5V supply will be monitored. Do not close both jumpers simultaneously.

The system reset also can be done via any UART. Therefore the user has the possibility to select one of the three on-board UARTs.

JP57 Select which UART should generate the system reset

JP29 The polarity of the DTR/RTS signal can be inverted by this jumper. Remove the jumper to disable the UART reset function

Jumper	Setting	Description
JP57	1-2	Reset will be done via UART2
	3-4	Reset will be done via UART4
	5-6	Reset will be done via UART5
JP29	1-2	No polarity inversion for the DTR/RTS signal
	2-3	Polarity inversion for the DTR/RTS signal

Table 3-16: Reset UART selection

By default the UART reset is disabled.

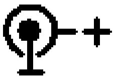
Note:
During normal operation, the reset LED should be off!

If the reset LED is steadily on, check the power supply voltages and the settings for the reset generation by UART.

4 Connectors

4.1 **Power Connector** (X5)

The following figure shows the power connection jack of the starter-kit. This connector is used to connect an external unregulated DC power supply voltage (9 V-12 V DC recommended 2000 mA).



Shield is connected to positive voltage supply (+)

Centre is connected to ground (GND)

Figure 4-1 Power connector

4.2 **UART / LIN Connector** (X1, X4, X8)

Three 9-pin D-Sub female connectors are used for the serial interfaces LIN/UART2, LIN/UART4 and LIN/UART5.

Take care, that the RS232 as well as the LIN signals are shared at the connectors and have to be selected by jumpers (see chapter 3.4).

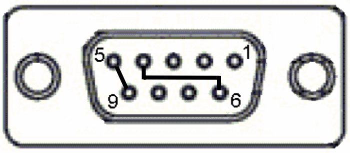


Figure 4-2 UART connector

Pin Number	Pin Signal	Description
1	+VBat	Power from LIN bus
2	TXD	RS-232 transmit output
	LIN	Bi-directional LIN-interface
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

Note:
Please use 1:1 cable for RS232 PC-connection.

4.3 CAN Connector (X9, X10, X11)

Three 9-pin D-Sub male connectors are used for the CAN interfaces CAN0, CAN1 and CAN2.

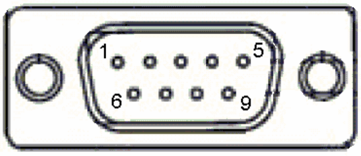


Figure 4-3: CAN connector

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

4.4 USER-LEDs and Optional LC-Display (D1-D16, J6)

There are sixteen user LEDs on-board, D1-D8 connected to MCU port P25 and D9-D16 connected to MCU port P16. To disconnect the LEDs from the MCU, it is possible to remove RN3 (D1-D8 at port P25) and RN2 (D9-D16 at port P16).

It is also possible to connect a standard LCD module with backlight to connector J6. With the potentiometer RP1 the contrast of the LCD can be adjusted.

Pin	14	13	12	11	10	9	8	7	6	5	4	3	2	1	16	15
LCD	DB7	DB6	DB5	DB4	-	-	-	-	E	R/W	RS	VO	Vcc	GND	LED-	LED+
LED	D8	D7	D6	D5				D4	D3	D2	D1					
Port	P25_7	P25_6	P25_5	P25_4				P25_3	P25_2	P25_1	P25_0					
MCU	187	186	185	184				181	180	179	178					

Table 4-3: LED/LCD Signals

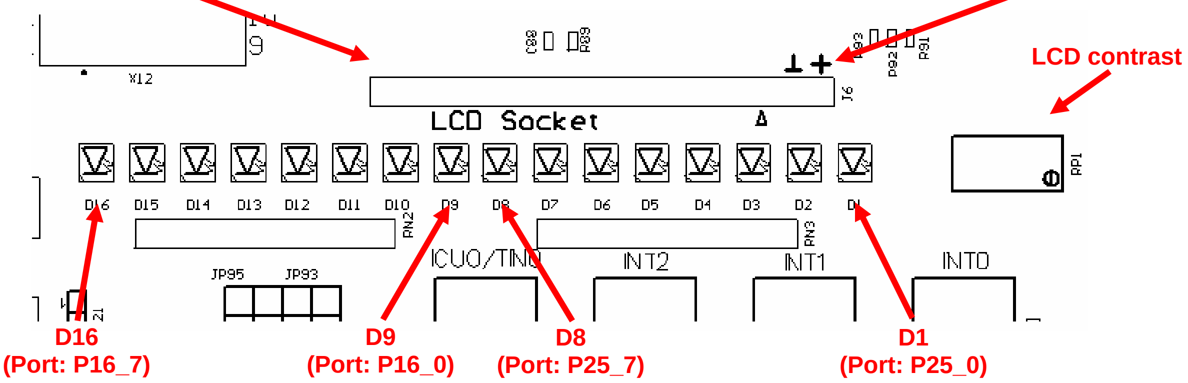


Figure 4-4: User LEDs / LCD

4.5 In-Circuit-Programming Connector (X12)

There is a flash-programming socket on the starter-kit which makes it possible to program the flash MCU with a special programming adapter. Mode pins and reset signal are also available at this connector.

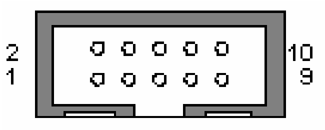


Figure 4-5: In-circuit programming connector

Pin Number	Pin Signal	Description
1	NC	Not used
2	NC	Not used
3	MD0	MCU mode-pin 0
4	MD2	MCU mode-pin 2
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

4.6 MCU Pin Connectors (J1-J4)

All MCU pins are directly connected to the four pin header around the MCU. The last two pins of the pin header J1, J2, J3 and J4 are unused (see PCB silk plot).

Connector	MCU Pins
J1	1-52
J2	53-104
J3	105-156
J4	157-208

Table 4-5: MCU pin connectors

The corresponding pin numbers of the MCU are written next to the connector pins on the PCB.

4.7 External Bus Connectors (X13, X14)

4.7.1 VG96ABC DIN 41612 (X13)

Pin	Name	MCU-P.	Pin	Name	MCU-P.	Pin	Name	MCU-P.
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	D0	192	B5	D1	193	C5	D2	194
A6	D3	195	B6	D4	196	C6	D5	197
A7	D6	198	B7	D7	199	C7	D8	200
A8	D9	201	B8	D10	202	C8	D11	203
A9	D12	204	B9	D13	205	C9	D14	206
A10	D15	207	B10	D16	2	C10	D17	3
A11	D18	4	B11	D19	5	C11	D20	6
A12	D21	7	B12	D22	8	C12	D23	9
A13	D24	10	B13	D25	11	C13	D26	12
A14	D27	13	B14	D28	14	C14	D29	15
A15	D30	16	B15	D31	17	C15	GND	-
A16	GND	-	B16	GND	-	C16	GND	-
A17	A0	18	B17	A1	19	C17	A2	20
A18	A3	21	B18	A4	22	C18	A5	23
A19	A6	24	B19	A7	25	C19	A8	28
A20	A9	29	B20	A10	30	C20	A11	31
A21	A12	32	B21	A13	33	C21	A14	34
A22	A15	35	B22	A16	36	C22	A17	37
A23	A18	38	B23	A19	39	C23	A20	40
A24	A21	41	B24	A22	42	C24	A23	43
A25	A24	44	B25	A25	45	C25	NC	-
A26	NC	-	B26	NC	-	C26	NC	-
A27	NC	-	B27	NC	-	C27	GND	-
A28	P27_0	158	B28	P27_1	159	C28	P27_2	160
A29	P27_3	161	B29	P27_4	164	C29	P27_5	165
A30	P27_6	166	B30	P27_7	167	C30	GND	-
A31	2.5V	-	B31	2.5V	-	C31	NC	
A32	GND	-	B32	GND	-	C32	GND	

Table 4-6: VG96 (X13) signals

Note:
The pin numbers of the VG connectors refer to the male connectors, that plugs into the starter-kit. Please see chapter 5 to find the right orientation.

4.7.2 VG48ABC DIN 41612 (X14)

Pin	Name	MCU-P.
A1	CSX0	56
A2	CSX2	58
A3	NC	-
A4	CSX6	60
A5	BGRNTX	51
A6	RDX	50
A7	WRX1	47
A8	WRX3	49
A9	NC	-
A10	DREQ0	189
A11	DEOP0	191
A12	INT12	97
A13	INT13	98
A14	SIN5	109
A15	GND	-
A16	INITX	73

Pin	Name	MCU-P.
B1	GND	-
B2	1.8V	-
B3	GND	-
B4	1.8V	-
B5	GND	-
B6	NC	-
B7	NC	-
B8	NC	-
B9	NC	-
B10	NC	-
B11	NC	-
B12	NC	-
B13	NC	-
B14	Vin	-
B15	Vin	-
B16	Vin	-

Pin	Name	MCU-P.
C1	CSX1	57
C2	CSX3	59
C3	NC	-
C4	RDY	55
C5	BRQ	54
C6	WRX0	46
C7	WRX2	48
C8	ASX	62
C9	MCLKO	65
C10	DACKX0	190
C11	NC	-
C12	SDA0	99
C13	SCL0	100
C14	SOT5	110
C15	SCK5	111
C16	GND	-

Table 4-7: VG48 (X14) signals

> **Note:**
> The pin numbers of the VG connectors refer to the male connectors, that plugs into the starter-kit. Please see chapter 5 to find the right orientation.

4.7.3 Distance between VG-Connectors

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

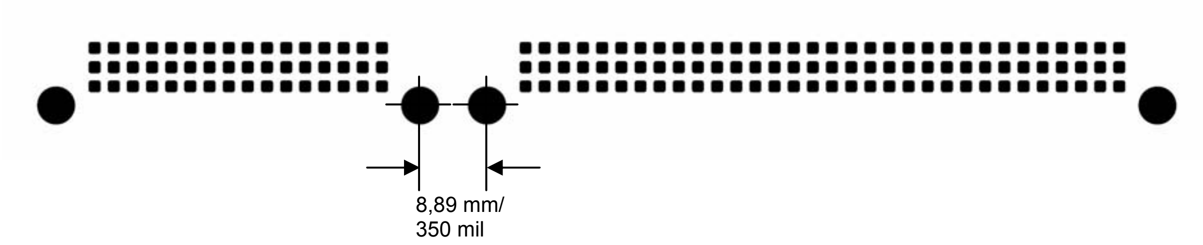


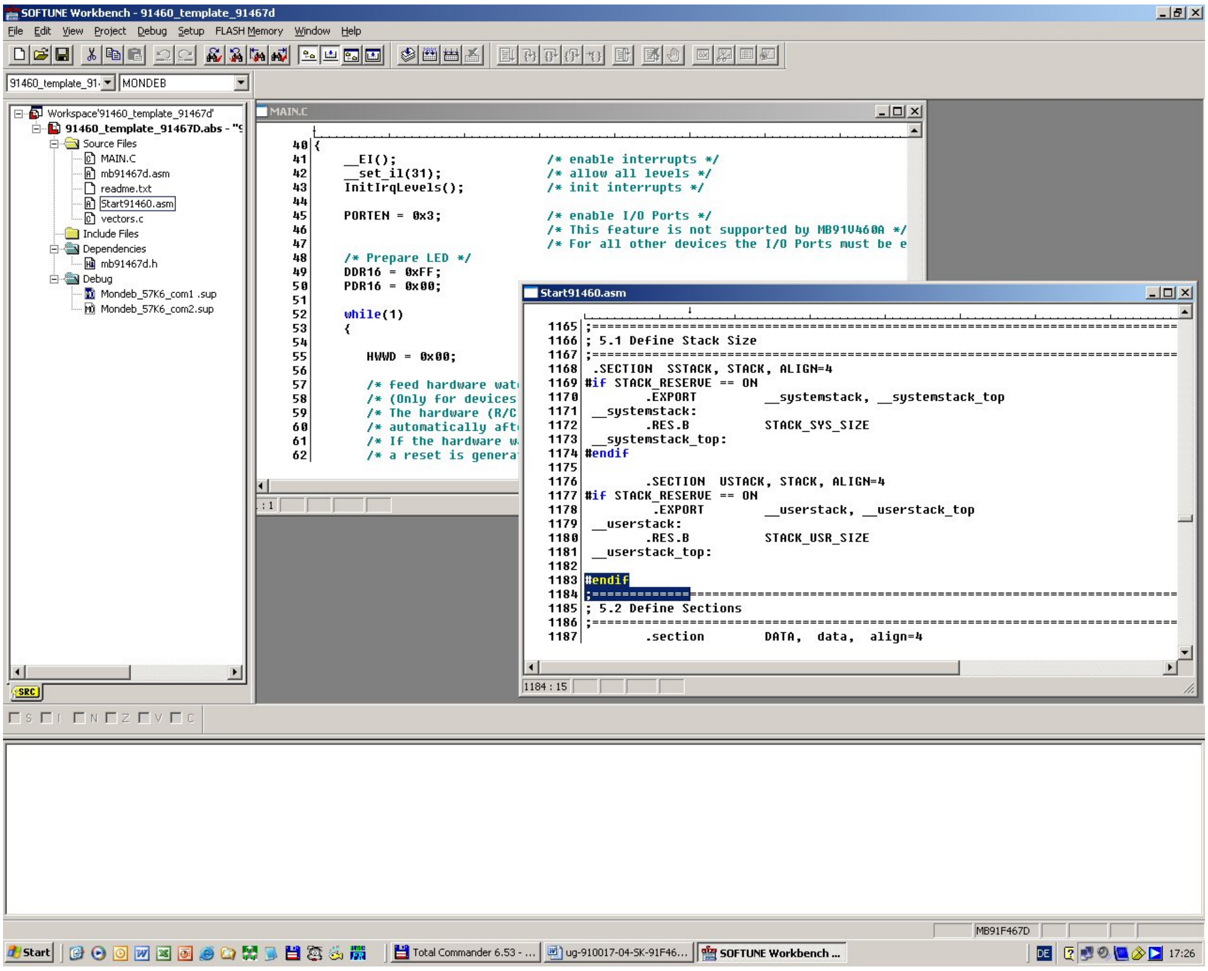
Figure 4-6: Layout of DIN VG connectors

5 Getting Started

Make sure that the FR Family Softune Workbench is installed on your PC and the “SK-91F467D-208PFV” is ready to use (power-on (X5), serial connection (UART4) to COM port of the PC).

5.1 Introduction to Softune Workbench

Start Softune Workbench. Select <File> <Open Workspace> to load the first example “mb91460_io.wsp”. All examples can be found in the folder \Software\SWB_Samples on the starter-kit 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.



```
{
    __EI();                         /* enable interrupts */
    __set_il(31);                   /* allow all levels */
    InitIrqLevels();                /* init interrupts */

    PORTEN = 0x3;                   /* enable I/O Ports */
                                    /* This feature is not supported by MB91V460A */
                                    /* For all other devices the I/O Ports must be e

    /* Prepare LED */
    DDR16 = 0xFF;
    PDR16 = 0x00;

    while(1)
    {

       HWWD = 0x00;

       /* feed hardware wat
       /* (Only for devices
       /* The hardware (R/C
       /* automatically aft
       /* If the hardware w
       /* a reset is genera
```

```
;===========================================================================
; 5.1 Define Stack Size
;===========================================================================
 .SECTION  SSTACK, STACK, ALIGN=4
#if STACK_RESERVE == ON
        .EXPORT         __systemstack, __systemstack_top
 __systemstack:
        .RES.B          STACK_SYS_SIZE
 __systemstack_top:
#endif

        .SECTION  USTACK, STACK, ALIGN=4
#if STACK_RESERVE == ON
         .EXPORT        __userstack, __userstack_top
 __userstack:
        .RES.B          STACK_USR_SIZE
 __userstack_top:

#endif
;===========================================================================
; 5.2 Define Sections
;===========================================================================
        .section        DATA,  data,  align=4
```

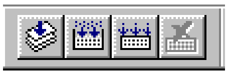
Figure 5-1: Softune Workbench main window

> **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).



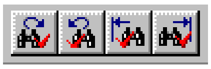
Click on MAKE or BUILD. Messages from the individual language tools will be fed into the output window at the bottom of the main screen. If the tool chain (C-compiler → Assembler → Linker) was completed successfully, the message “No Error” will appear.

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_io.prj - Debug--------------------
main.c
*** C:\SK-91F467-208PFV\91460_io\Src\main.c(81) E4065B: expected a ";"
*** C:\SK-91F467-208PFV\91460_io\Src\main.c(104) W1012B: warning: parsing restarts here after
previous syntax error
1 error detected in the compilation of
"C:\ SK-91F467-208PFV\91460_io\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.

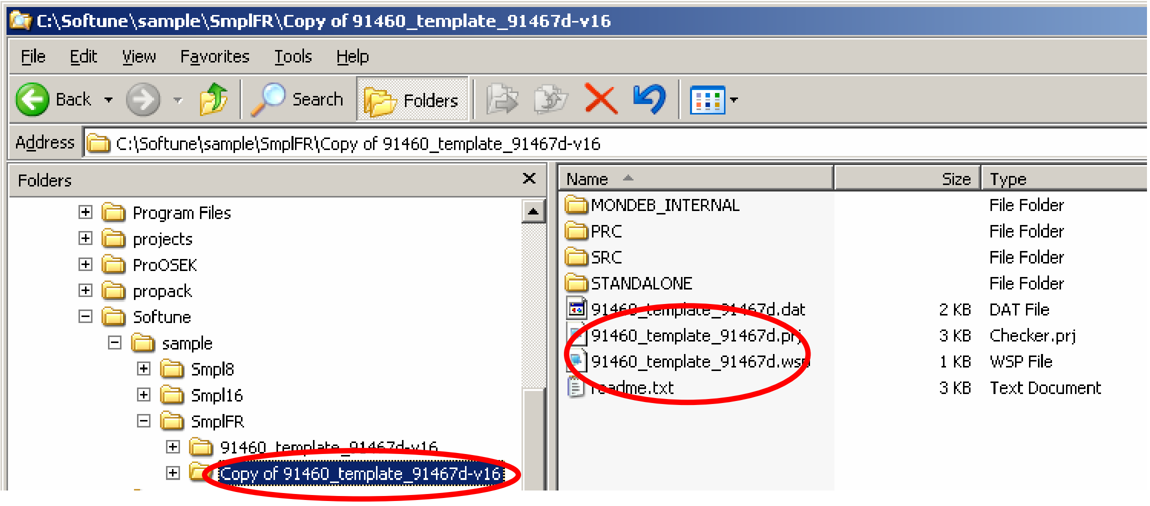
Note:

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_91467d”** project
2. **Copy** the folder **91460_template_91467d** and **rename** the folder to e.g. my_application



C:\Softune\sample\SmplFR\Copy of 91460_template_91467d-v16

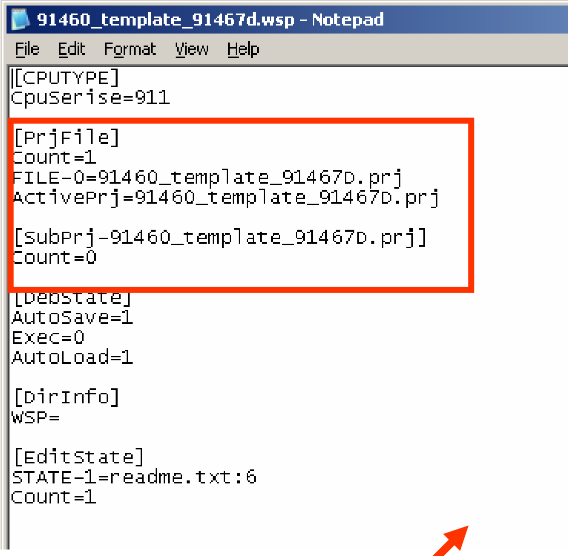
Address: C:\Softune\sample\SmplFR\Copy of 91460_template_91467d-v16

Name	Size	Type
MONDEB_INTERNAL		File Folder
PRC		File Folder
SRC		File Folder
STANDALONE		File Folder
91460_template_91467d.dat	2 KB	DAT File
91460_template_91467d.prj	3 KB	Checker.prj
91460_template_91467d.wsp	1 KB	WSP File
readme.txt	3 KB	Text Document

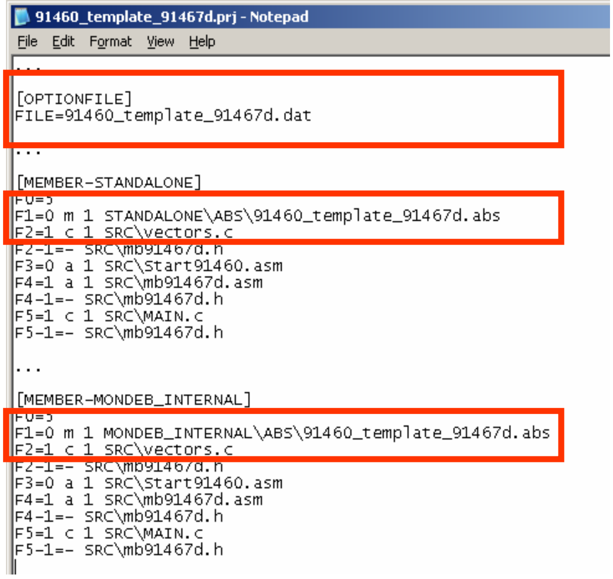
3. **Enter the folder “my_application”**
   Rename “91460_template_91467d.prj” into “my_application.prj”
   Rename “91460_template_91467d” into “my_application.wsp”

4. **Edit “my_application.prj”**
   Rename “91460_template_91467d” -> “my_application”

5. **Edit “my_application.wsp”**
   Rename “91460_template_91467d” -> “my_application”


91460_template_91467d.wsp - Notepad
File Edit Format View Help

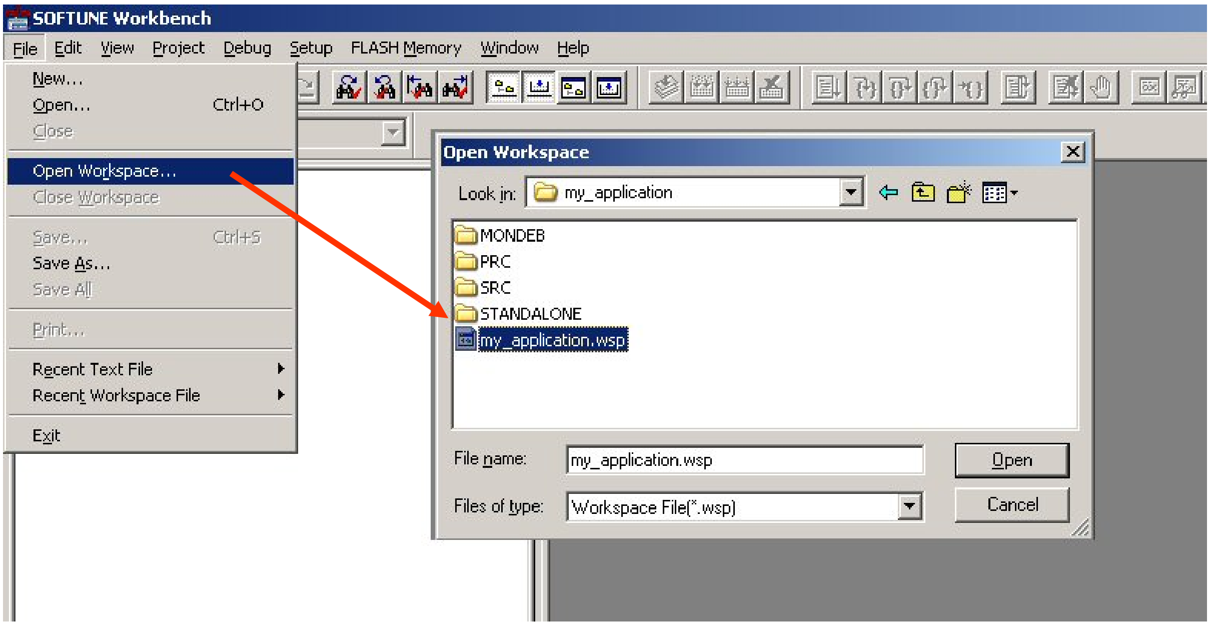
```
[CPUTYPE]
CpuSerise=911

[PrjFile]
Count=1
FILE-0=91460_template_91467D.prj
ActivePrj=91460_template_91467D.prj

[SubPrj-91460_template_91467D.prj]
Count=0
[DebState]
AutoSave=1
Exec=0
AutoLoad=1

[DirInfo]
WSP=

[EditState]
STATE-1=readme.txt:6
Count=1
```


91460_template_91467d.prj - Notepad
File Edit Format View Help

```
...
[OPTIONFILE]
FILE=91460_template_91467d.dat
...
[MEMBER-STANDALONE]
F0=5
F1=0 m 1 STANDALONE\ABS\91460_template_91467d.abs
F2=1 c 1 SRC\vectors.c
F2-1=- SRC\mb91467d.h
F3=0 a 1 SRC\Start91460.asm
F4=1 a 1 SRC\mb91467d.asm
F4-1=- SRC\mb91467d.h
F5=1 c 1 SRC\MAIN.c
F5-1=- SRC\mb91467d.h
...
[MEMBER-MONDEB_INTERNAL]
F0=5
F1=0 m 1 MONDEB_INTERNAL\ABS\91460_template_91467d.abs
F2=1 c 1 SRC\vectors.c
F2-1=- SRC\mb91467d.h
F3=0 a 1 SRC\Start91460.asm
F4=1 a 1 SRC\mb91467d.asm
F4-1=- SRC\mb91467d.h
F5=1 c 1 SRC\MAIN.c
F5-1=- SRC\mb91467d.h
```

Rename “91460_template_91467d**”**
into “my_application”

6. Start Softune Workbench and open your project



Look in: my_application

File name: my_application.wsp

Files of type: Workspace File(*.wsp)

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. MB91F467D.
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

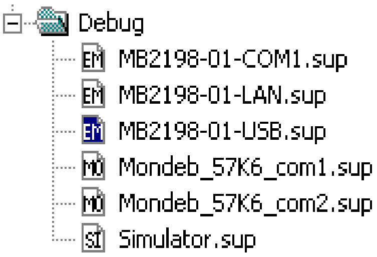
5.3.1 General Description

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-91F467-208PFV” 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):



```
Debug
    MB2198-01-COM1.sup
    MB2198-01-LAN.sup
    MB2198-01-USB.sup
    Mondeb_57K6_com1.sup
    Mondeb_57K6_com2.sup
    Simulator.sup
```

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”.

5.3.2 Starting the SWB Monitor Debugger

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 S5 to (On, Off, Off, Off). The correct start-up of the SWB Monitor Debugger is indicated by D2, D4 and D8

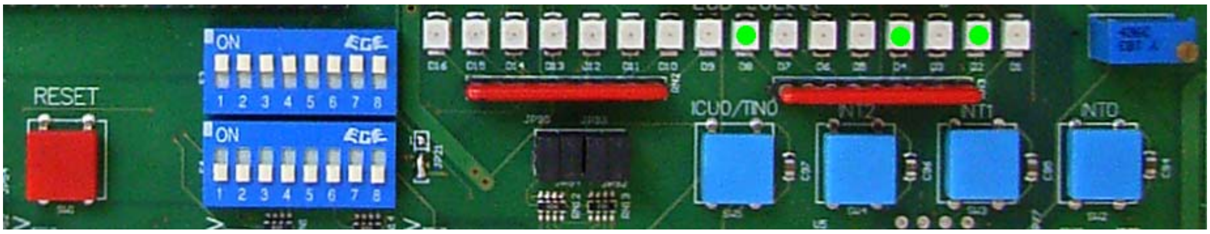
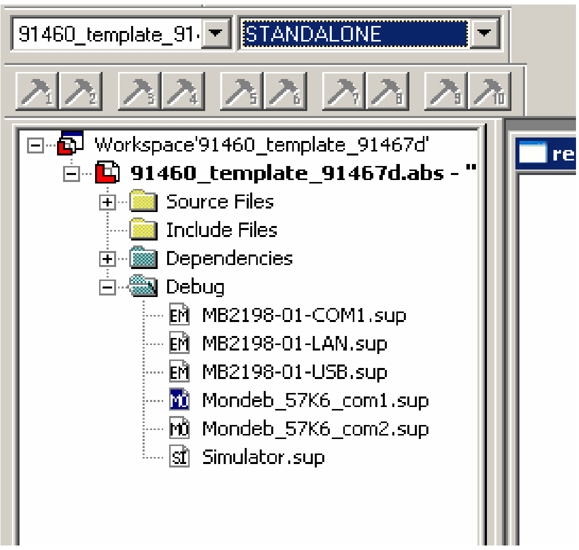


Figure 5-3: Correct start-up of the SWB Monitor Debugger

2) 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).



91460_template_91·
STANDALONE

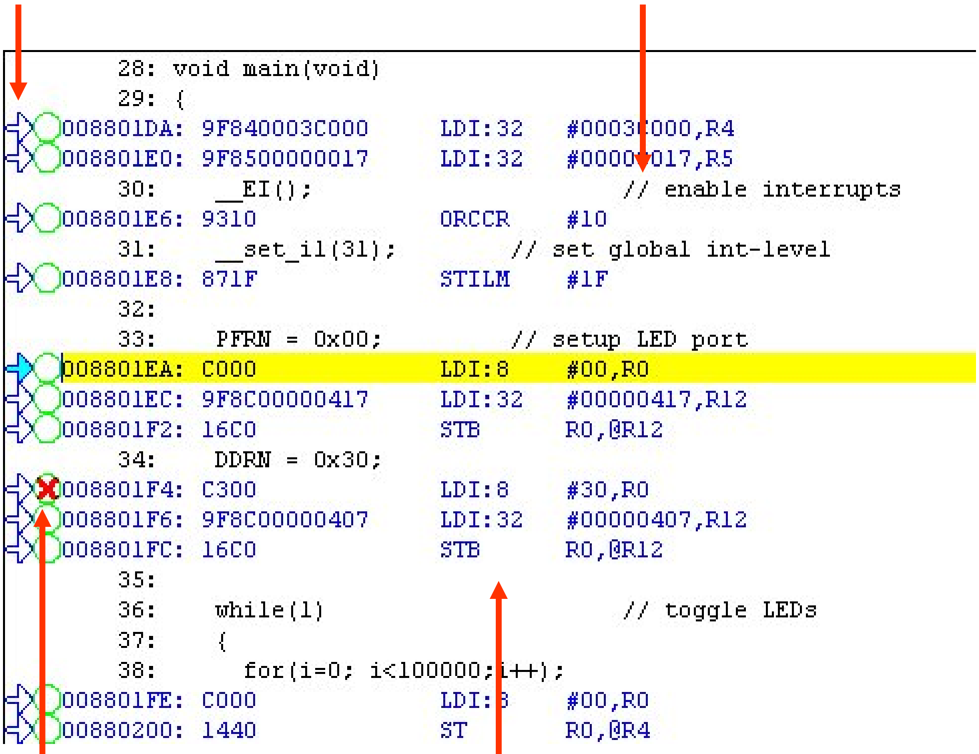
- Workspace'91460_template_91467d'
  - 91460_template_91467d.abs - "
    - Source Files
    - Include Files
    - Dependencies
    - Debug
      - MB2198-01-COM1.sup
      - MB2198-01-LAN.sup
      - MB2198-01-USB.sup
      - Mondeb_57K6_com1.sup
      - Mondeb_57K6_com2.sup
      - Simulator.sup

Figure 5-4: Start SWB Monitor Debugger with external flash and SRAM

3) 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().

5.3.3 Basic Debugger Features

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 5-5: 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!

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!

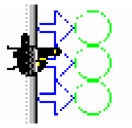
STEP OVER: Executes the program stepwise except call-instructions (which will be executed until return). Interrupts are not disabled during a continuous CALL...RETURN execution.

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 INT0 on the "SK-91F467-208PFV"** for **ABORT** function.

To **ABORT** continuous execution, you have to use the **INT0 button** on the starter-kit.

5.4 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.

5.5 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.

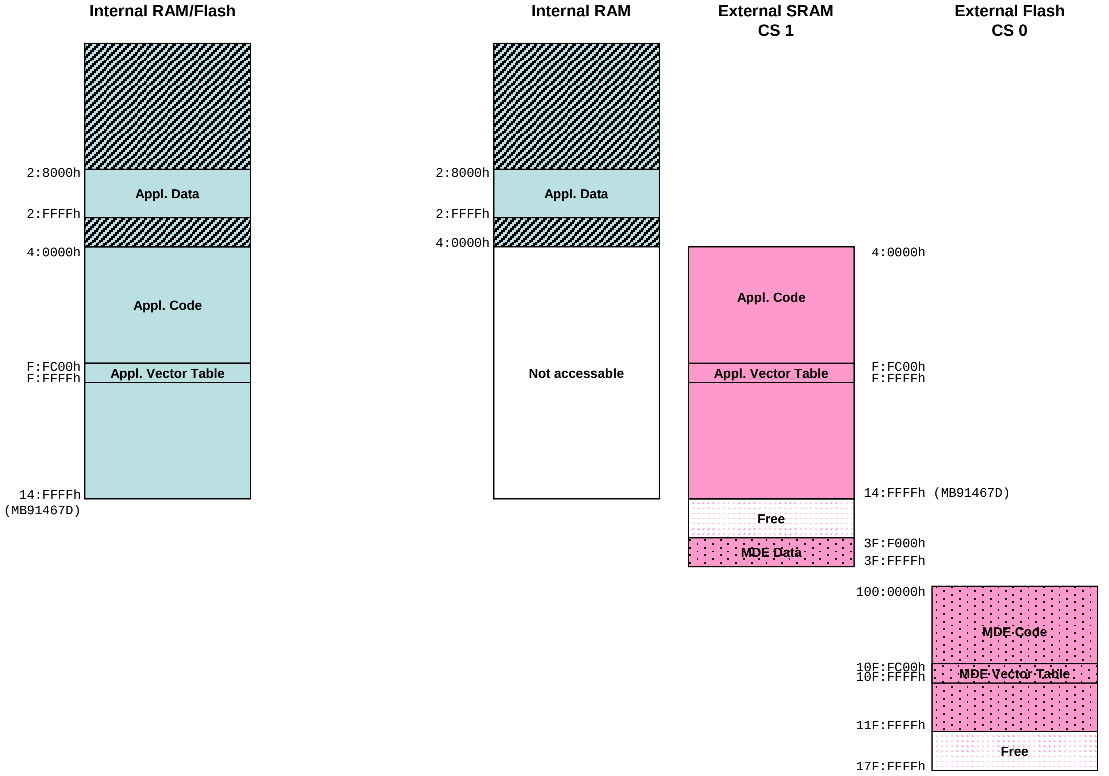


Figure 5-6: Default linkage map for the SWB Monitor Debugger of the MB91467D

On the left hand site the memory organisation of the normal operation mode of the microcontroller MB91F467D can be seen, the whole application is internal. For this operation mode the mode pins have to be set to 0-0-0, hence DIP switch S5 has to be set to off, off, off, off.

On the right hand site the microcontroller runs in the external reset-vector mode, which requires the mode pins setting 1-0-0, hence DIP switch S5 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.

6 Programming the internal Flash

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 X4 which is connected to UART4 has to be used. The following jumper setting is needed.

Jumper	Setting	Description
JP60	1-2 closed	MCU SIN4 connected to RS-232 transceiver
JP65	1-2 closed	MCU SOT4 connected to RS-232 transceiver
JP69	1-2 closed	X4 pin2 is connected to RS-232 transceiver

Table 6-1: Jumper setting on starterkit SK-91F467D-208PFV

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

Jumper	Setting	Description
JP58	closed	RTS and CTS is shortcut at connector X4
	open	RTS and CTS is not shortcut at connector X4
JP64	1-2 closed	DTR can be used for system reset
	2-3 closed	RTS can be used for system reset
JP57	3-4 closed	Reset will be done via UART4
JP29	1-2 closed	No polarity inversion for the DTR/RTS signal
	2-3 closed	Polarity inversion for the DTR/RTS signal

Table 6-2: Jumper settings for reset signal on starterkit SK-91F467D-208PFV

3) The CPU Mode selection must be set to internal reset vector mode by setting DIP switch S5 to (Off, Off, Off, Off)

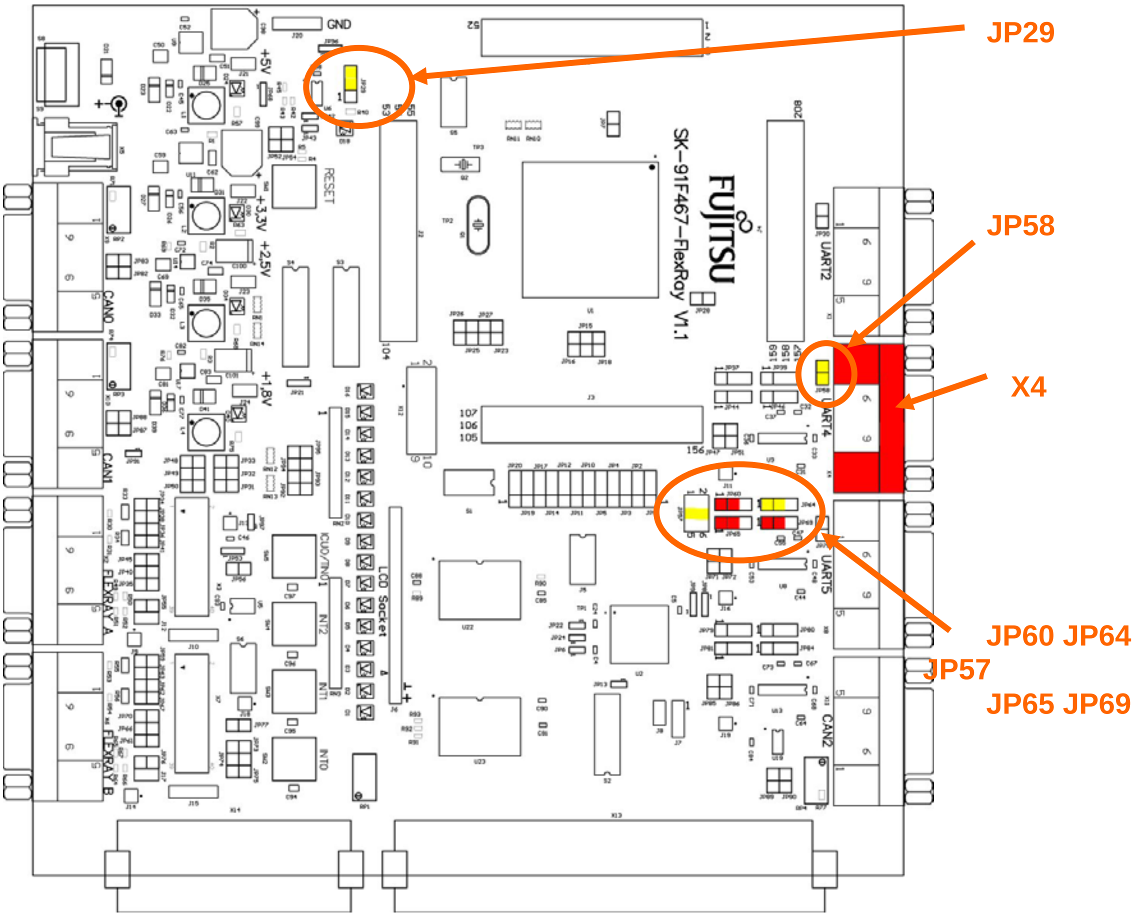
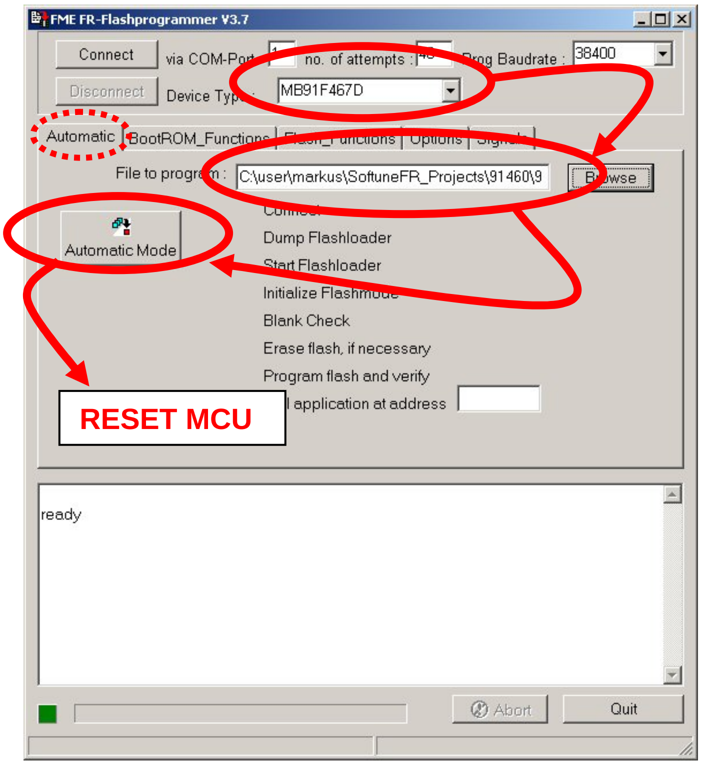


Figure 6-1: Jumper setting on starterkit SK-91F467D-208PFV

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 ("MB91F467D") 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.



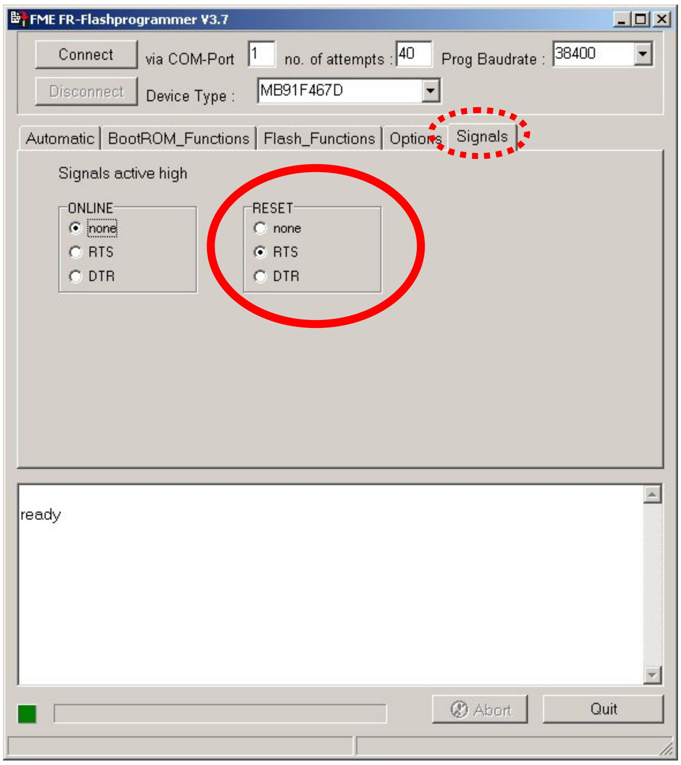
via COM-Port: 1
no. of attempts: 40
Prog Baudrate: 38400
Device Type: MB91F467D
File to program: C:\user\markus\SoftuneFR_Projects\91460\9

Connect
Dump Flashloader
Start Flashloader
Initialize Flashmode
Blank Check
Erase flash, if necessary
Program flash and verify
l application at address

ready

Figure 6-2: 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".



via COM-Port: 1
no. of attempts: 40
Prog Baudrate: 38400
Device Type: MB91F467D

Signals active high

ONLINE
● none
○ RTS
○ DTR

RESET
○ none
● RTS
○ DTR

ready

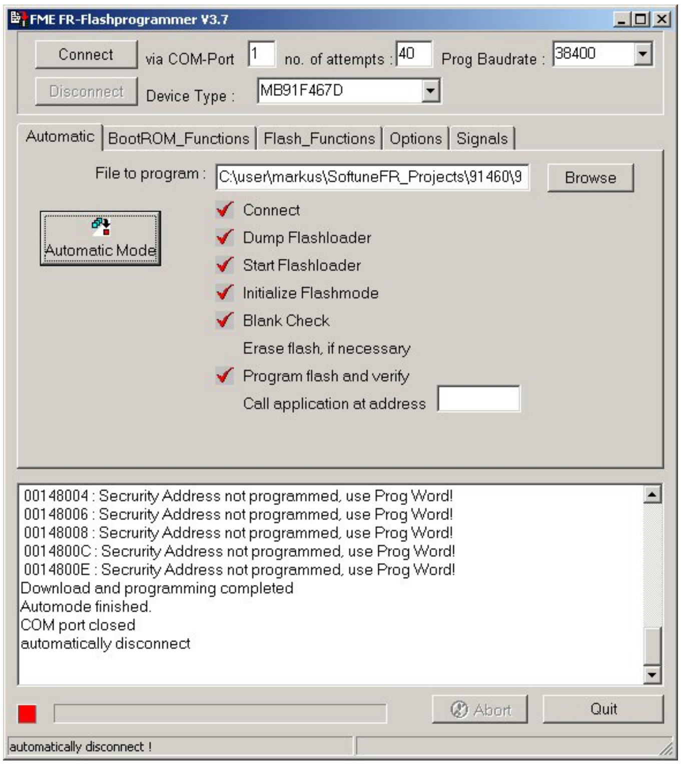
Figure 6-3: Reset Signal Settings

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.



via COM-Port 1 no. of attempts : 40 Prog Baudrate : 38400

Device Type : MB91F467D

File to program : C:\user\markus\SoftuneFR_Projects\91460\9

✓ Connect
✓ Dump Flashloader
✓ Start Flashloader
✓ Initialize Flashmode
✓ Blank Check
Erase flash, if necessary
✓ Program flash and verify
Call application at address

```
 00148004 : Secrurity Address not programmed, use Prog Word!
 00148006 : Secrurity Address not programmed, use Prog Word!
 00148008 : Secrurity Address not programmed, use Prog Word!
 0014800C : Secrurity Address not programmed, use Prog Word!
 0014800E : Secrurity Address not programmed, use Prog Word!
Download and programming completed
Automode finished.
COM port closed
automatically disconnect
```

automatically disconnect !

Figure 6-4: Successful Programming in Automatic Mode

7 Programming the SWB Monitor Debugger

The following procedure describes the installation process for the SK-91F467D-208PFV and the MB91F467D.

1) The CPU mode selection must be set to internal reset vector mode by setting DIP switch S5 to (Off, Off, Off, Off)

2) Program the monitor-programmer "91460_mondeb_uart4_extbus_flash_sram" into the external Flash, e.g. with the help of the FME Flash Programmer V3.11 or later. Please see the chapter above and the application note mcu-an-300012-e-mb91460_flash_programming.pdf, too. Please use the automatic mode. The settings for CS0 should be ASR0: 0100, ACR0: 8822, AWR0: 3378.

3) Please, power-off the SK-91F467D-208PFV.

4) The CPU Mode selection must be set to external reset vector mode by setting DIP switch S5 to (On, Off, Off, Off)

5) The correct start-up of the SWB Monitor Debugger is indicated by D2, D4 and D8.

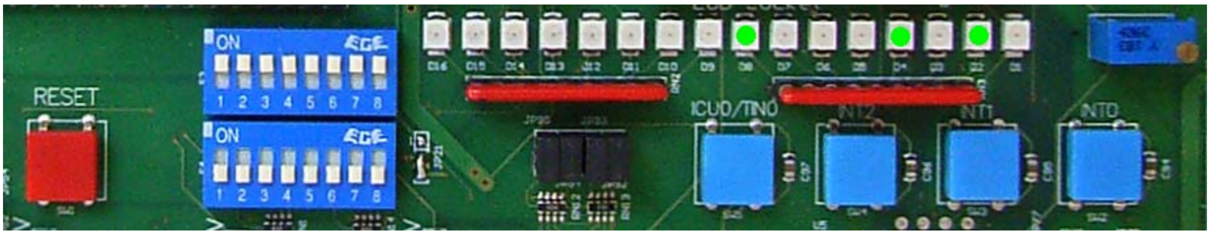


Figure 7-1: Correct Start-Up of the SWB Monitor Debugger

6) The SWB Monitor Debugger is now ready to use.

8 Trouble shooting

Problem	Solution
Starter-kit does not power-on correctly. Some or all of the power LEDs (D24, D30, D34, D40) 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-91F467D-208PFV" 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 J21, J22, J23 and J24. ∗ Check jumper positions according to chapter 2.1 ∗ Remove all user extensions of the starter-kit
Starter-kit stays in reset. LED D18 (Reset indicator) lights up permanently.	∗ LED D18 (Reset) should flash. If D18 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 JP58, 60, 65 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 J6 does not work	∗ Adjust the potentiometer RP1 to adjust contrast of LCD. Remove resistor network RN3.
Graphic sub-board does not work	Check power supply 9-12V/1500mA. Reduce external bus clock frequency.

Table 8-1: Trouble shooting

9 Silk-Plot of the Board

9.1 Top Side

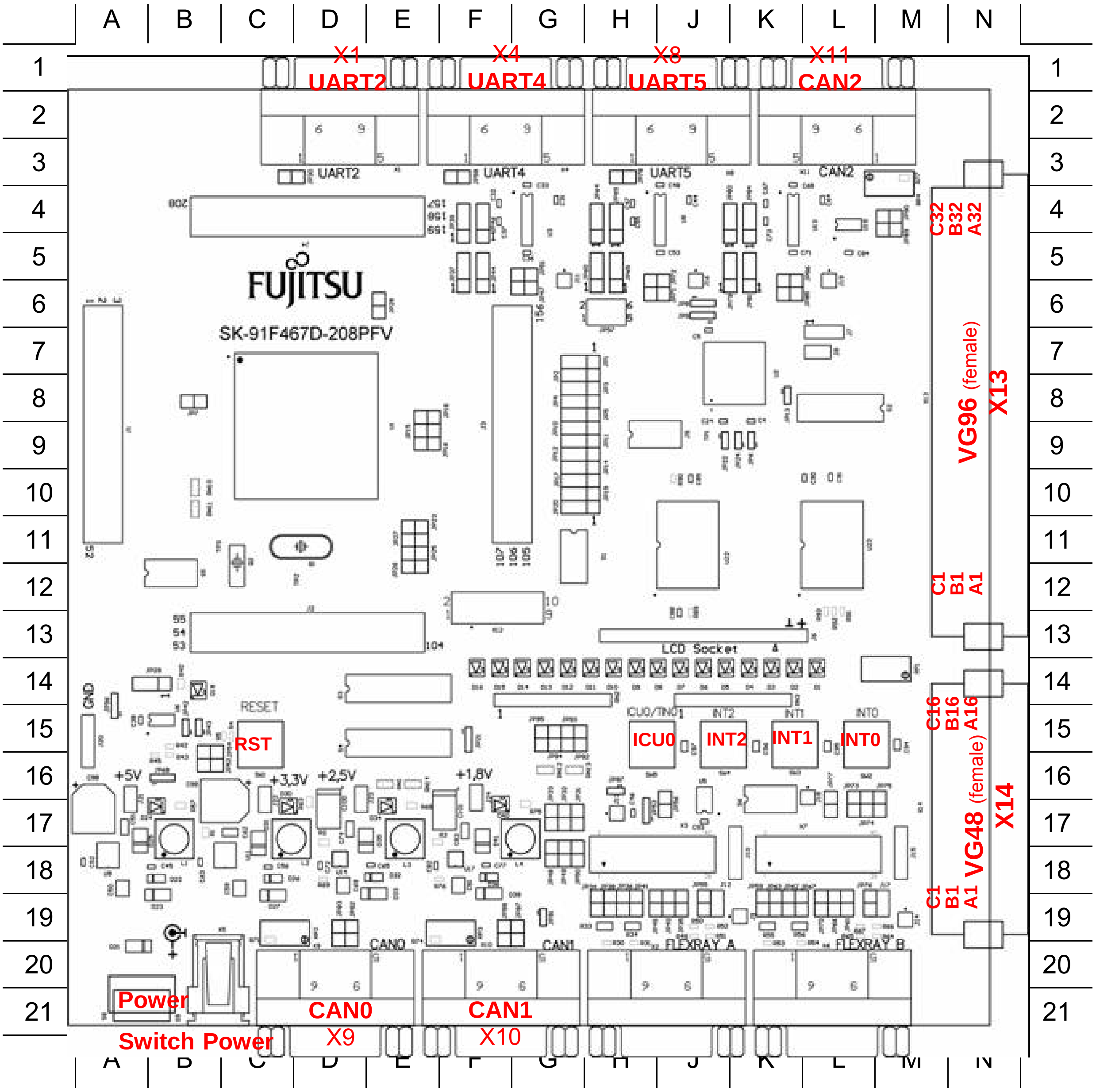


Figure 9-1: Board layout (top side)

9.2 Bottom Side

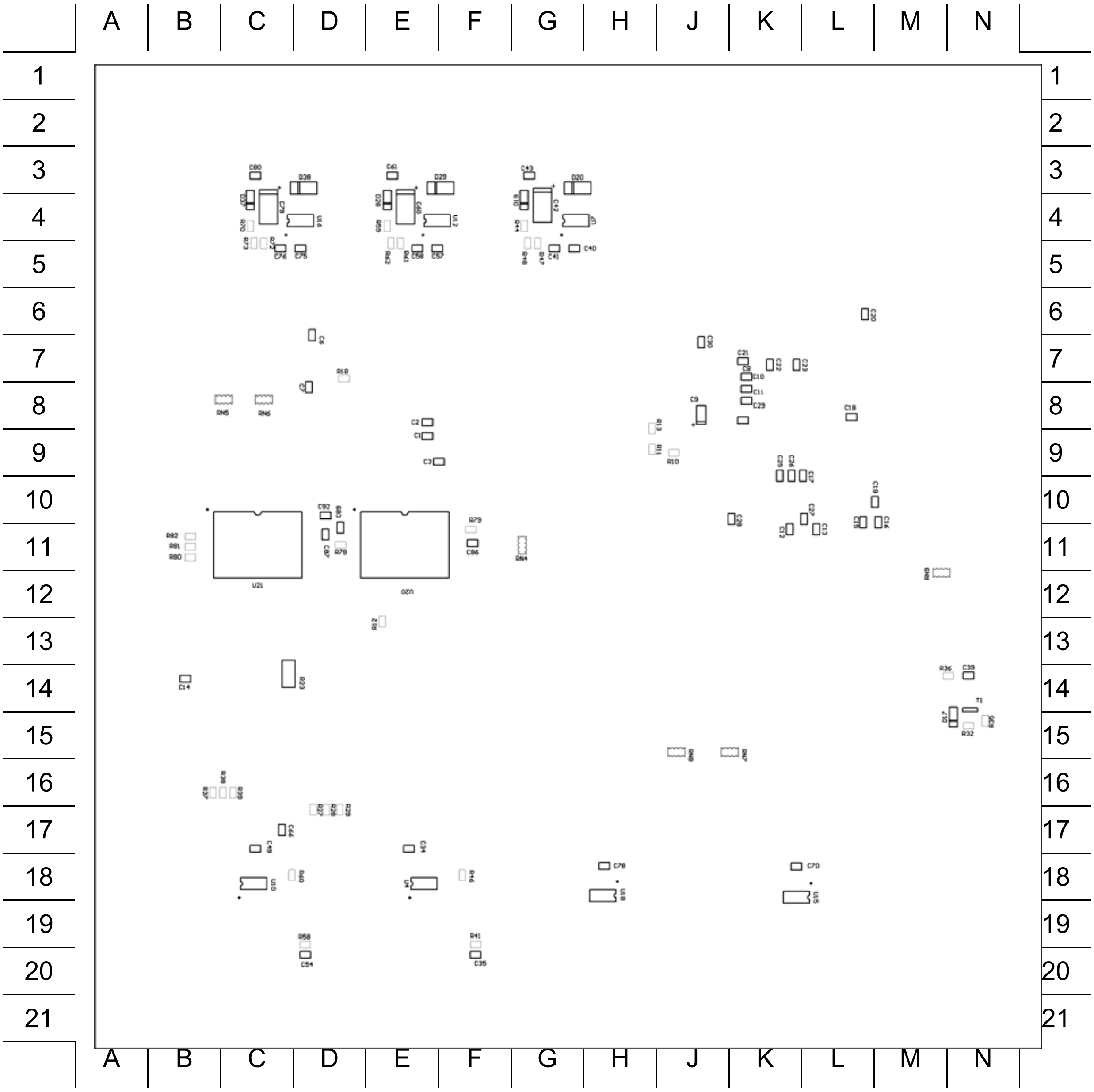


Figure 9-2: Board layout (bottom side)

10 PCB History

10.1 SK-91F467-208PFV V1.0

Part	Problem	Fixed
U20	Wrong UB/LB connection (WRX2 and WRX3 has to be connected)	V1.1
X14	Connect 1.8V supply not directly to X14 (add jumper)	V1.1
U6	Additionally pull-up resistors for MCU and ASSP reset line	V1.1
U6	Reset button connection (SW1), add further option to activate reset directly when pressing SW1. Actually, SW1 must be pressed 2s to activate reset. (add solder jumper to select one option)	V1.1
U9	Add C=220uF/16F between Vcc5V and GND	V1.1
U11	Add C=220uF/16F between Vcc3V3 and GND	V1.1
U14	Add C=100uF/16F between Vcc2V5 and GND	V1.1
U17	Add C=100uF/16F between Vcc1V8 and GND	V1.1

Table 10-1: History PCB V1.0

11 Appendix

11.1 Related Products

SK-91F467D-208PFV	Evaluation board with MB91F467D
SK-91F467-FLEXRAY	Evaluation board with MB91F467D and MB88121A
SK-91460-MAIN	Evaluation main board for MB91460 Series
SK-91460-91F467D-208PFV	MCU-board incl. MB91F467D for SK-91460-MAIN
ADA-91V460-91F467D	Conversion adapter MB91V460 to MB91F467D

MB2198-01	Emulator debugger main unit
MB2198-10	DSU cable
MB2198-300	Adapter board for MB91V460 emulation chip (BGA-660P)
MB91V460	Emulation chip of MB91460 Series
PB-91467D-208PFV	Probe cable for MB91F467D target system (Recommended for SK-91F467D-208PFV)
PB-91467D-NLS-208PFV	Probe cable for MB91F467D target system (Recommended for SK-91F467-FLEXRAY)
NQPACK208SD	Socket for package FPT-208P-M01 and FPT-208P-M04 Tokyo Eletech Corp. www.tetc.co.jp/e_tet.htm
HQPACK208SD306H	Header for FPT-208P-M01 e.g.: MB91F467D
YQPACK208SD	Joint connector extension for NQPACK208SD
YQSOCKET208SDN	Joint connector socket for NQPACK208SD

MB91F467D	Flash chip of MB91460 Series
MB88121A	FlexRay Communication Controller

11.2 Information on the WWW

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

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

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

Automotive products: MCU, Graphic display controller, MPEG en/decoder, gyro sensors

http://www.fujitsu.com/emea/services/industries/automotive/

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

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

Media Products: SAW filters, acoustic resonators and VCOs

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

For more information about FUJITSU MICROELECTRONICS

http://emea.fujitsu.com/microelectronics

11.3 Tables

11.4 Figures

11.5 Abbreviations

ADC	Analogue Digital Converter
ASSP	Application Specific Standard Product
CAN	Controller Area Network
CS	Chip Select
FME	Fujitsu Microelectronics Europe GmbH
LB	Lower Byte
LIN	Local Interconnect Network
MCU	Microcontroller Unit
PCB	Printed Circuit Board
SRAM	Static Random Access Memory
UART	Universal Asynchronous Receiver Transmitter
UB	Upper Byte
UG	User Guide
USART	Universal Synchronous Asynchronous Receiver Transmitter