The following document contains information on Cypress products. The document has the series name, product name, and ordering part numbering with the prefix “MB”. However, Cypress will offer these products to new and existing customers with the series name, product name, and ordering part number with the prefix “CY”.

How to Check the Ordering Part Number

1. Go to www.cypress.com/pcn.
2. Enter the keyword (for example, ordering part number) in the **SEARCH PCNS** field and click **Apply**.
3. Click the corresponding title from the search results.
4. Download the Affected Parts List file, which has details of all changes

For More Information

Please contact your local sales office for additional information about Cypress products and solutions.

About Cypress

Cypress is the leader in advanced embedded system solutions for the world's most innovative automotive, industrial, smart home appliances, consumer electronics and medical products. Cypress' microcontrollers, analog ICs, wireless and USB-based connectivity solutions and reliable, high-performance memories help engineers design differentiated products and get them to market first. Cypress is committed to providing customers with the best support and development resources on the planet enabling them to disrupt markets by creating new product categories in record time. To learn more, go to www.cypress.com.

AN205206

FR Family MB91460 Series MB91461R Emulation

This application note describes the different methods of debugging the MB91461R MCU.

Contents

1 Introduction....1
2 Evaluation system with MB91V460....1
3 Evaluation system with MB91461R....5
3.1 EMU_FLASH_CS0_USB.sup....8
3.2 EMU_SRAM_CS1_USB.sup....11
4 Softune Workbench Monitor Debugger....11
5 Accemic MDE Monitor Debugger....13
6 Softune Workbench Simulator....16
7 Comparison Table....17
8 Document History....18
Worldwide Sales and Design Support....19
Products....19
PSoC® Solutions....19
Cypress Developer Community....19
Technical Support....19

1 Introduction

The MB91461R is a ROMless derivate of the Cypress FR-MB91460 Series. It has a complete debug support unit (DSU4) integrated like the evaluation chip MB91V460. This is a major difference to other MB91460 series members. Because of this emulation is slightly different to the Flash devices of this family.

This application note describes the different methods of debugging the MB91461R MCU.

There are five possibilities:

1. Emulating the MB91461R by use of the evaluation chip MB91V460, adapter board and emulator (Chapter 2)
2. Connecting the emulator directly to DSU4 on MB91461R (Chapter 3)
3. Use of the Softune Workbench monitor debugger (Chapter 4)
4. Use of the Accemic MDE debugger (Third party tool, Chapter 5)
5. Softune Workbench Simulator (Chapter 6)

The last chapter gives a short overview of needed internal and peripheral resources as well as the different debug functionalities.

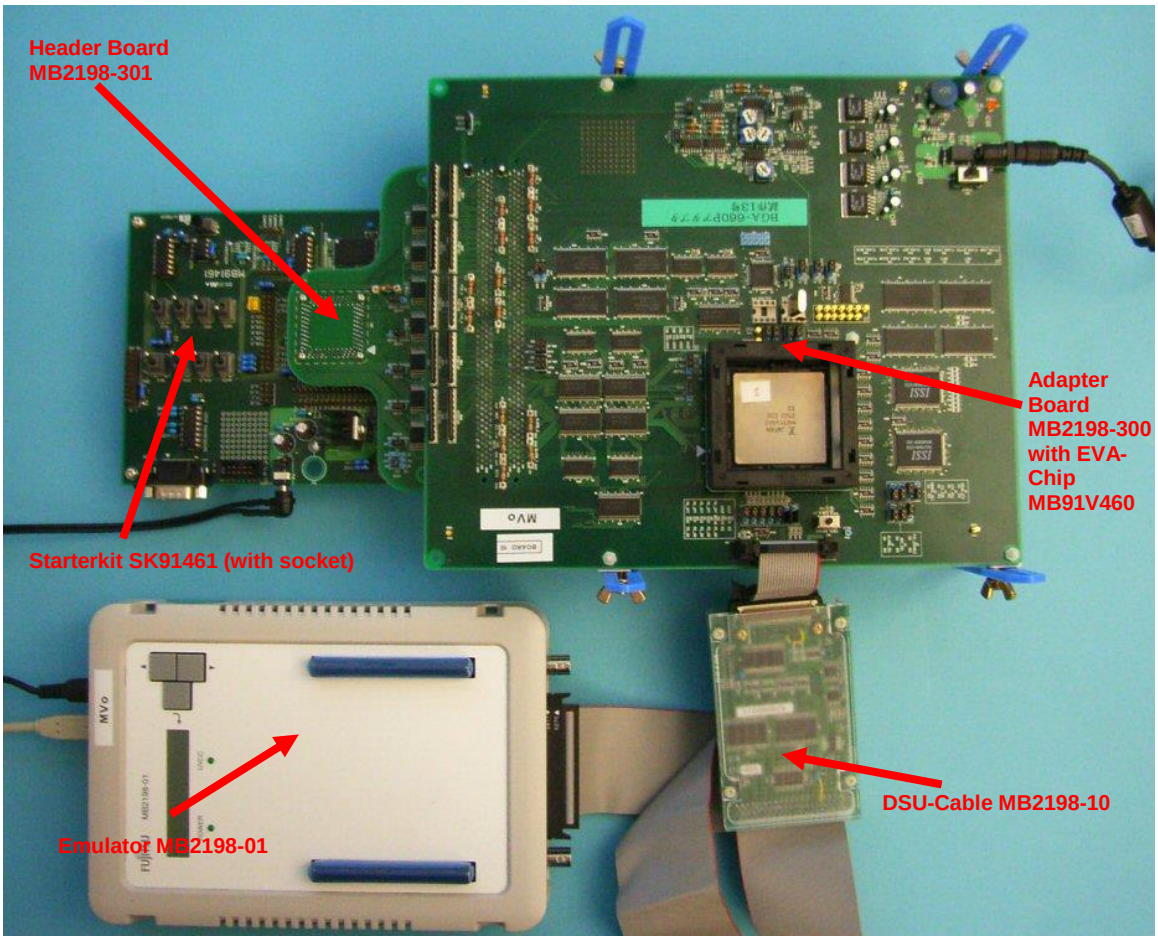
2 Evaluation system with MB91V460

This chapter describes the evaluation system containing the evaluation chip MB91V460.

The evaluation system using the evaluation chip MB91V460 consists of following devices:

- DSU-FR Emulator MB2198-01
- DSU-FR Cable MB2198-10
- Adapter board MB2198-300
- Evaluation Chip MB91V460 (BGA-660 pin package)
- Header Board MB2198-301 (for LQFP-176 pin package)
- Starterkit SK91461 or own target application with socket

Figure 1. Evaluation System with Eva-Chip



The following tables show the needed jumper settings on adapter and header board for use of the starterkit SK91461:

Table 1. Jumper settings on adapter board MB2198-300

No.	Name	Setting	No.	Name	Setting
1	ALARM_SEL	DIS	19	ICD0	PD_OFF
2	ADC_SEL	DIS	20	BREAK	PD_OFF
3	HWWDG_KILL	DIS	21	P10_7	PULLDOWN
4	FIX_ENX	DIS	22	X0	EML
5	CSV_KILL	EN	23	X1	EML
6	SRAM_SFX	DIS	24	X0A	EML
7	FSC_DISABLE	EN	25	X1A	EML
8	C	VCC3	26	SHDNSEL	UVCC3
9	-	-	27	VCC3SEL	3.5V
10	MDENS1	1	28	VCC5	UVCC5
11	MDENS0	1	29	VCC3H	UVCC3
12	MDENS2	1	30	MCLKI	EVA

No.	Name	Setting	No.	Name	Setting
13	D32_D16SL	D16	31	MCLKO	EVA
14	INITX	ICE	32	P224	Connection
15	TRSTX	ICE	33	P242	Connection
16	ICD3	PD_OFF	34	P225	Connection
17	ICD2	PD_OFF	35	P243	Connection
18	ICD1	PD_OFF	-		

Table 2. Jumper settings on header board MB2198-301

No.	Name	Setting
1	VCCB1	VCC5H
2	VCCB2	VCC5H
3	VCCB3	VCC5H
4	VCCB4	VCC5H
5	VCCB5	VCC3H
6	VCC5	Pin 133

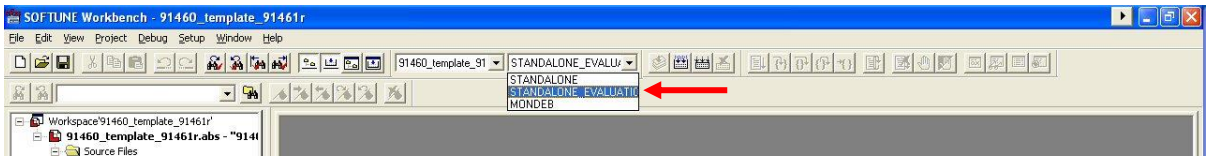
Set switches on starterkit SK91461 as given in table below:

Table 3. Jumper and switches on SK91461

Name (Switches)	Setting
RES	ICE
P15_3	*Don't care*
P15_2	*Don't care*
INT0	*Don't care*
MD0	H
MD1	L
MD2	L
MD3	L
Name (Jumpers)	**Setting**
P21_0	*Don't care*
P21_1	*Don't care*
P15_3	*Don't care*
P15_2	*Don't care*

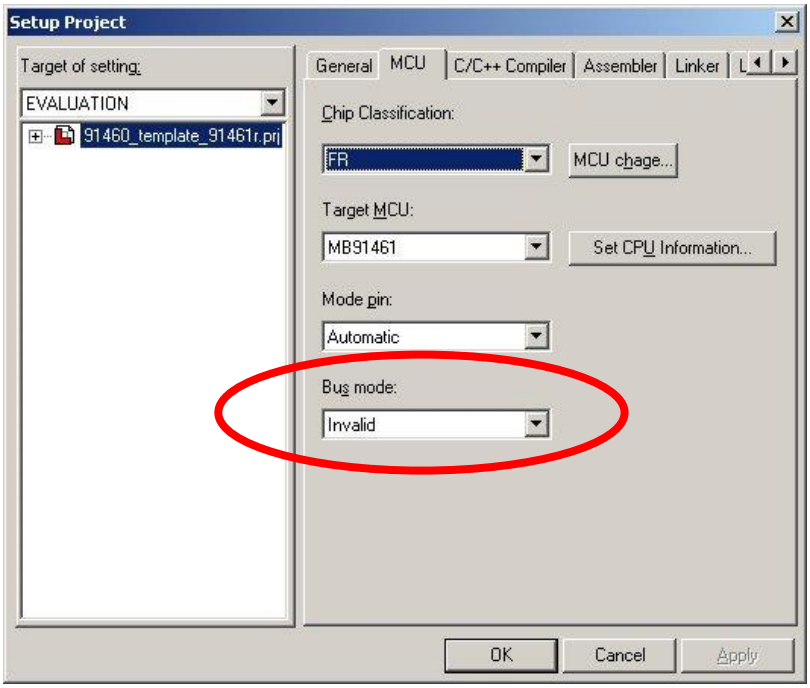
If your project is based on the MB91461R template project, please change the actual project configuration to STANDALONE_EVALUATION.

Figure 2. Select project configuration STANDALONE_EVALUATION



If you do not use the template project, please check following setting on 'MCU' tab of the 'Setup Project' dialog (*'Project' -> 'Setup Project…'*) and change bus mode to invalid.

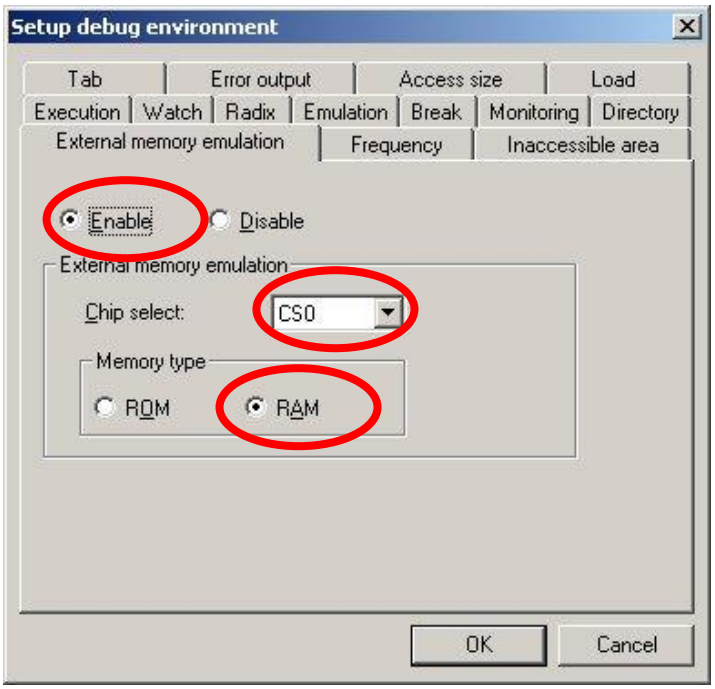
Figure 3. Configure Bus Mode Invalid



Double-click on MB2198-01_USB.sup (or change it for another connection type) to start the debug session.

With the predefined setting in template project you can directly download your code and start debugging. Otherwise you will have to check the following setting under *'Setup' -> 'Debug Environment' -> 'Debug Environment…'*

Figure 4. Enable external memory emulation



Enable external memory emulation for chip select 0 area as RAM type. This will force the evaluation MCU to use the external SRAM on MB2198-300, which is connected to the external bus, as memory for the debug session.

If you use the template project for project startup, the *.sup file calls two procedure files when starting debug session. Before the code is loaded to external emulation RAM, the prc-file 'Ext_Emu_RAM_before_load.prc' initializes the memory area for code download. After the download, prc-file 'Ext_Emu_RAM_after_load.prc' is called which initializes the MCU for run mode.

This type of emulation system has the advantage that you can debug directly on your target system without the need of a connector socket for the DSU cable on the target system itself. You only have to mount a socket instead of the MB91461R chip for connection of the header board. Because of the usage of the external emulation memory you can use full 8MB size, not caring about the amount of memory connected on your target system.

3 Evaluation system with MB91461R

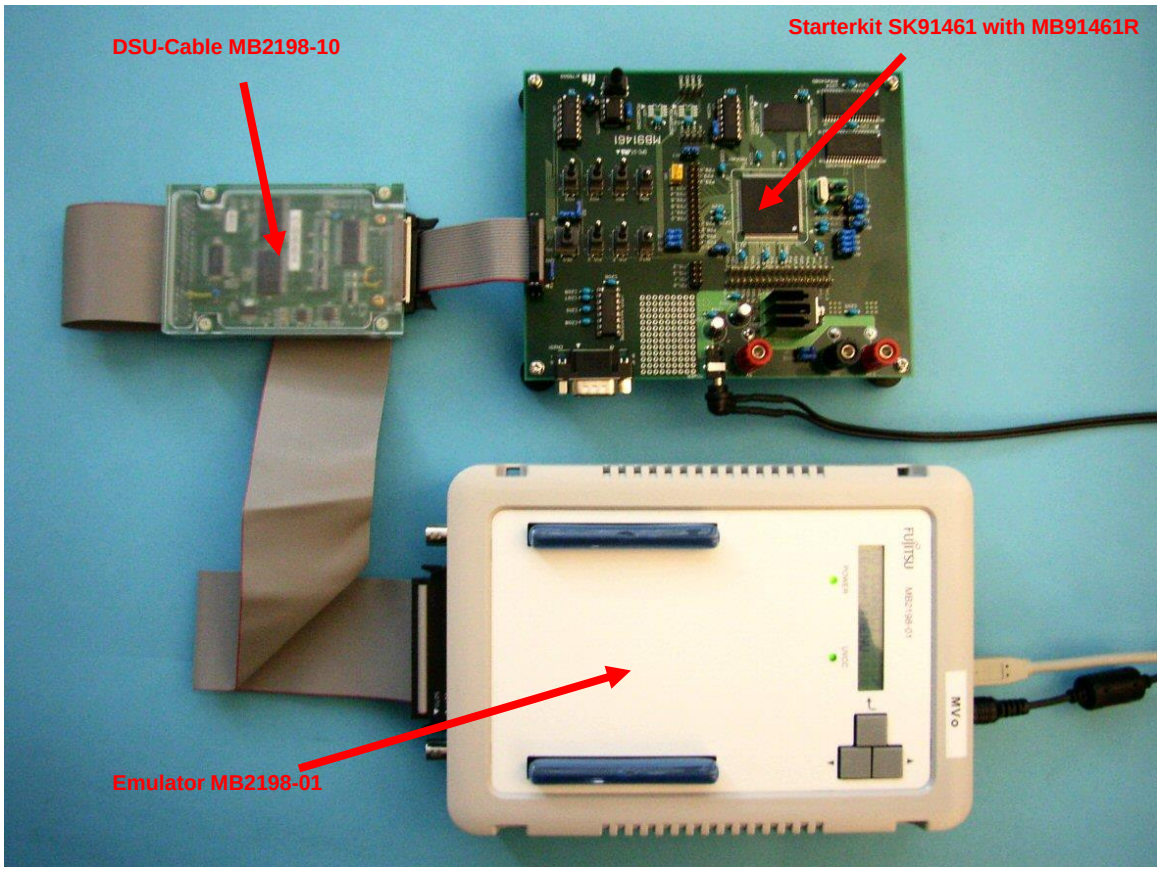
This chapter describes the evaluation system containing the target chip MB91461R.

The emulator can be directly connected to debug support unit (DSU4) on MB91461R. This is a main difference to other MB91460 series derivates.

The system consists of following devices:

- DSU-FR Emulator MB2198-01
- DSU-FR Cable MB2198-10
- Starterkit SK91461 or own target application with mounted MB91461R chip and connector for DSU cable

Figure 5. Evaluation system with MB91461R



Connection of the DSU pins of the microcontroller to the DSU connector is shown in MB91461R hardware manual, chapter 2.2.

Figure 6. DSU connector

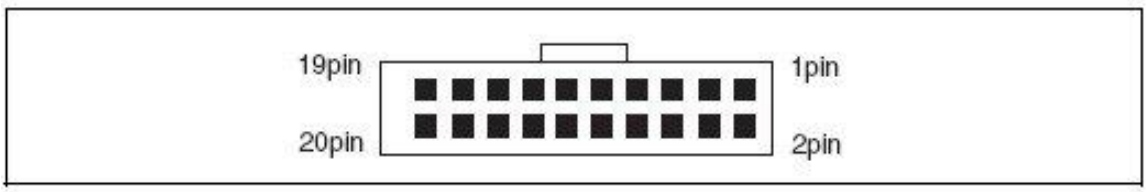
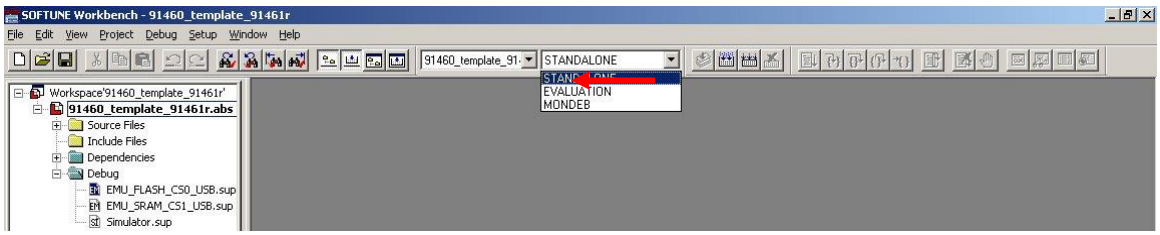


Table 4. Assignment of DSU Pins to Connector

Pin no.	Signal line name	Input/ output	Pin handling	
1	EVCC2	Input	OPEN	
2	EVCC3	Input	OPEN	
3	DSUIO	I/O	OPEN	
4	UVCC	Output	User Vcc output	
6	XRSTIN	Output	Connected to the INITX signal in the user circuit	
8	PLVL	Input	OPEN	
5	XTRST	Input	MB91460	Connected to TRSTX (Pin no. 130)
7	XINIT	Input		Connected to INITX (Pin no. 131)
9	GND	–		Connected to VSS
10	BREAK	Input		Connected to BREAK (Pin no. 98)
11	ICD[3]	I/O		Connected to ICD3 (Pin no. 93)
12	ICD[2]			Connected to ICD2 (Pin no. 92)
13	ICD[1]			Connected to ICD1 (Pin no. 91)
14	ICD[0]			Connected to ICD0 (Pin no. 90)
15	GND	–		Connected to VSS
16	ICS[2]	Output		Connected to ICS2 (Pin no. 96)
17	ICS[1]			Connected to ICS1 (Pin no. 95)
18	ICS[0]			Connected to ICS0 (Pin no. 94)
19	GND	–		Connected to VSS
20	ICLK	Output		Connected to ICLK (Pin no. 97)

Please select project configuration type STANDALONE if your project is based on the MB91461R template project.

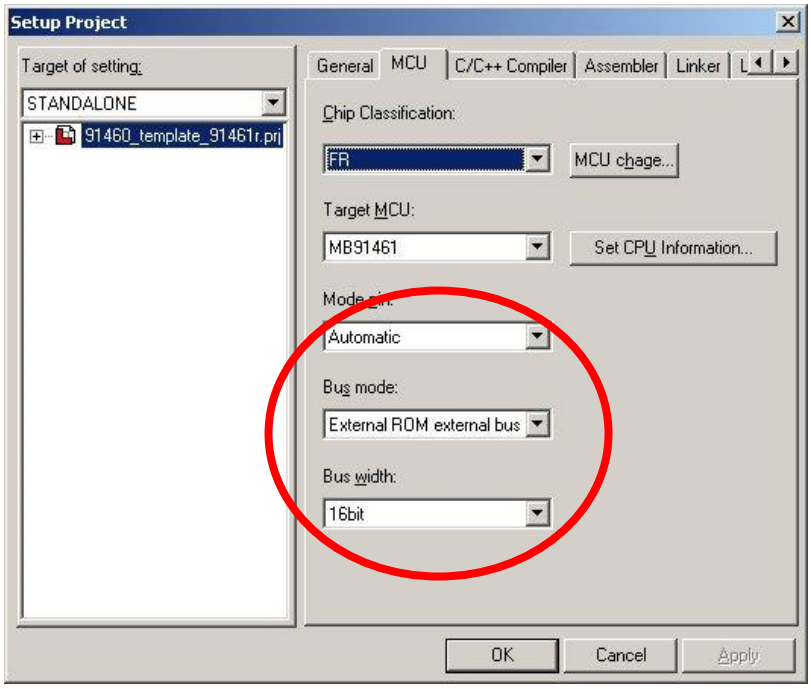
Figure 7. Select project configuration STANDALONE



Otherwise please check the following items under *'Project' ->'Setup Project…' -> 'MCU' tab:*

- Bus Mode: External ROM external Bus
- Bus width: 16bit

Figure 8. Configure Bus Mode External ROM



This kind of emulation system can be used in two different ways:

- The external Flash memory on SK91461 (4MB) connected to CS0 can be used
- The external SRAM memory on SK91461 (1MB) connected to CS1 can be used.

The template project therefore provides two debug configurations.

- EMU_FLASH_CS0_USB.sup
- EMU_SRAM_CS1_USB.sup

For proper startup of the debug sessions the template project includes some procedure file in folder PRC. Please refer to them if your project is not based on the template.

3.1 EMU_FLASH_CS0_USB.sup

This debug configuration uses the external Flash memory connected to CS0 area of the microcontroller.

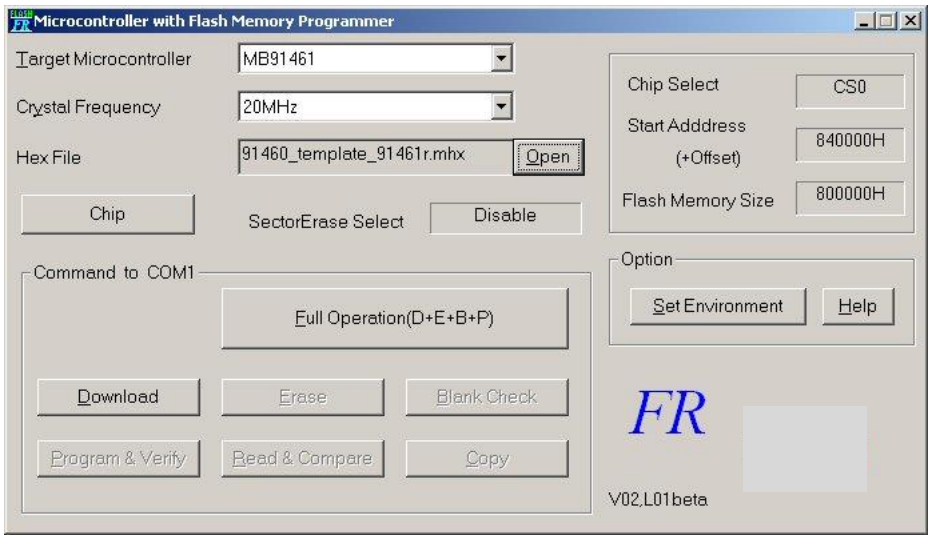
After generating your software project use the PC Writer software to program the adequate mhx-file (e.g. STANDALONE\ABS\91460_template_91461r.mhx) to the external Flash on SK91461. Therefore connect the starterkit via serial cable (crossed cable!) to the PC's COM port and set the onboard switches as following:

Table 5. Jumper and switches on SK91461 for Flash programming

Name (Switches)	Setting
RES	BRD
P15_3	L
P15_2	L
INT0	*Don't care*
MD0	L
MD1	L
MD2	L
MD3	L
Name (Jumpers)	**Setting**
P21_0	Connection
P21_1	Connection
P15_3	Connection
P15_2	Connection

Press the 'Full Operation' button and follow the instructions. Please refer to the programmer description for more details.

Figure 9. PC Writer software



After Flash programming you can disconnect the serial cable and change the settings back to:

Table 6. Jumper and switches on SK91461 Flash Emulation

Name (Switches)	Setting
RES	ICE
P15_3	*Don't care*
P15_2	*Don't care*
INT0	*Don't care*
MD0	H
MD1	L
MD2	L
MD3	L
Name (Jumpers)	**Setting**
P21_0	*Don't care*
P21_1	*Don't care*
P15_3	*Don't care*
P15_2	*Don't care*

Before starting debug session, you have to edit the procedure file PRC\Emu_FLASH_after_load.prc and set correct name of your generated abs-file.

```
# Emu_Flash_after_load.prc
# ========================
#
# 2006-04-24  V1.0 MVo

printf "\nprocedure file : Emu_Flash_after_load.prc\n"

printf "\n*********************************************************"
printf "\n*                                                       *"
printf "\n*   MB91461 Standalone Emulation with external FLASH    *"
printf "\n*                                                       *"
printf "\n*********************************************************\n"
printf "\nAttention:"
printf "\nPlease use PC-Writer software to program adequeate mhx file"
printf "\n(e.g. STANDALONE\\ABS\\91460_template_91461r.mhx) to external
Flash memory!"

RESET

printf "\n\n-------------------------"
printf "\nload debug information for project programmed to Flash\n"

LOAD/DEBUG STANDALONE\ABS\91460_template_91461r.abs
```

Attention:
You have to care that the programmed mhx-file as well as the debug information taken from the abs-file originate from the same project!

3.2 EMU_SRAM_CS1_USB.sup

This debug configuration uses the external SRAM memory connected to CS1 area of the microcontroller.

Please make sure the following setting on starterkit is done:

Table 7. Jumper and switches on SK91461 SRAM Emulation

Name (Switches)	Setting
RES	ICE
P15_3	*Don't care*
P15_2	*Don't care*
INT0	*Don't care*
MD0	H
MD1	L
MD2	L
MD3	L
Name (Jumpers)	**Setting**
P21_0	*Don't care*
P21_1	*Don't care*
P15_3	*Don't care*
P15_2	*Don't care*

Start the debug session by double-clicking on EMU_SRAM_CS1_USB.sup. CS1 area is initialized for program download to SRAM via procedure file. After that, CS1 is configured for program execution, MODR register and program counter are patched and the startup file is executed up to main()-function. This is also done by procedure file and is necessary to prevent any changes in startup file that could lead to malfunction of the debug session.

Attention:
Take care not to overwrite settings for CS1 area in the startup file! Otherwise program execution from external SRAM does not work anymore.

4 Softune Workbench Monitor Debugger

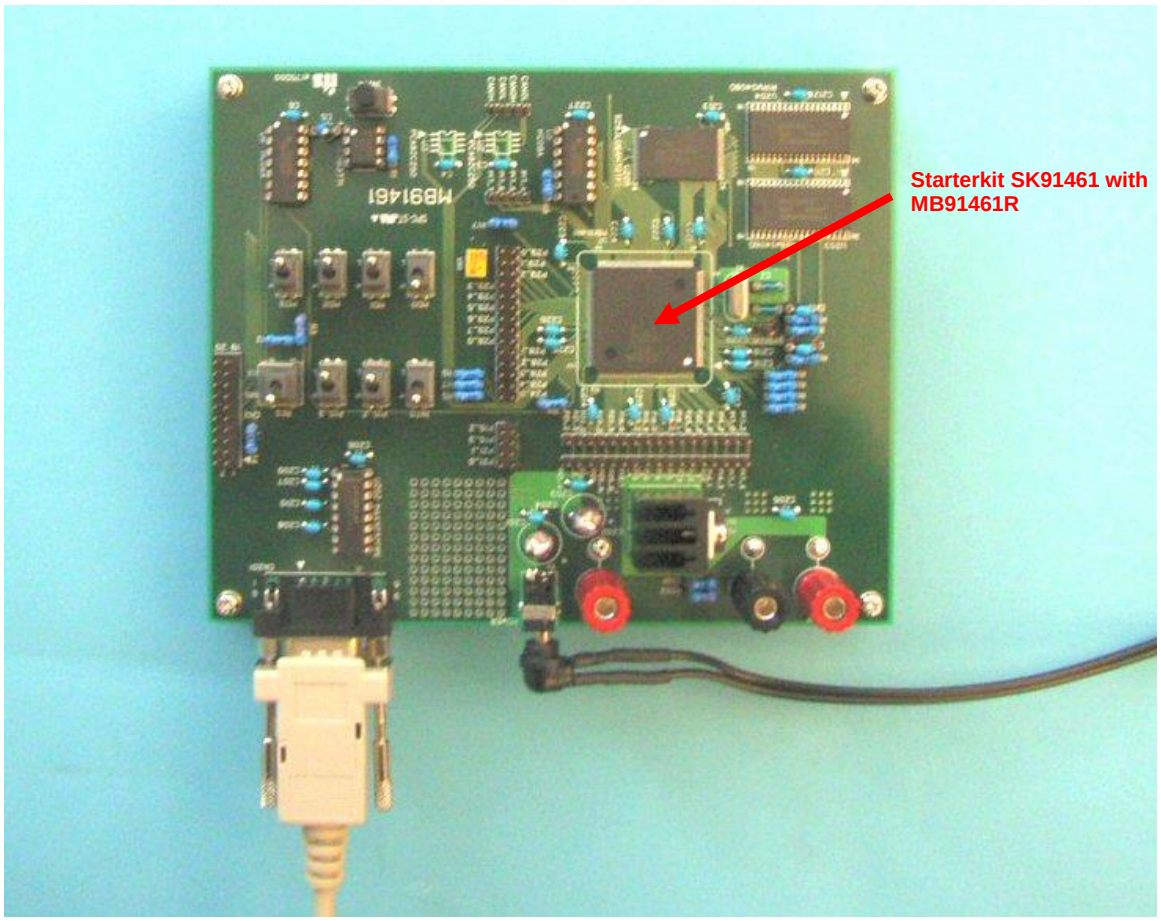
This chapter describes the debugging on MB91461R by use of the Softune Workbench Monitor Debugger.

The Softune Workbench monitor debugger is a control program, which has to be downloaded to the external Flash memory before starting the debug session.

Because the monitor debugger skips the clock initialisation part of the user project's startup file, you have to make all clock related settings in the file settings.h of the SWB monitor debugger project.

If you have made these settings, use the PC Writer Software to program monitor kernel to external Flash ROM via serial crossed cable.

Figure 10. Standalone usage with monitor debugger



The monitor debugger kernel initializes the CS1 area for the SRAM on starterkit to download the user project.

If your project is based on the MB91461R template project, choose project configuration MONDEB. Otherwise you will have to take care that your code and data sections are linked to the CS1 area configured by the monitor debugger kernel.

Before starting debug session, please check the settings on starterkit:

Table 8. Jumper and switches on SK91461 for SWB Monitor Debugger

Name (Switches)	Setting
RES	BRK
P15_3	*Don't care*
P15_2	*Don't care*
INT0	L
MD0	H
MD1	L
MD2	L
MD3	L
Name (Jumpers)	**Setting**
P21_0	Connection
P21_1	Connection
P15_3	*Don't care*
P15_2	*Don't care*

To start debug session connect UART0 via crossed serial cable to your PC's COM port and double-click the appropriate debug configuration (Mondeb_57K6_COM1/COM2.sup).

A procedure file is executed to run through the startup file of the user project without overwriting important settings like internal clocks or chip select settings. Procedure file stops at main()-function.

You can start debugging step-by-step or run up to a breakpoint. Do not use the 'Stop Execution' button in Softune Workbench! Set INT0 switch on starterkit to H to stop program execution.

See also application Note "AN205205 - FR Family MB91460, SWB Monitor Debugger" for more information on the monitor debugger features.

5 Accemic MDE Monitor Debugger

This chapter describes the debugging on MB91461 by use of the Accemic MDE Monitor Debugger.

The Accemic MDE Monitor Debugger is a third party tool for debugging the MB91461R. It provides an own IDE surface for debugging.

To use the Accemic MDE you have to add two files delivered with the MDE environment (monitorFR.asm, monitor.h) to your Softune Workbench project (STANDALONE configuration). Also take care not to locate your project sections overlapped with MDE sections.

Accemic MDE needs following address ranges:

```
S_Addr. -E_Addr.   Size      Section Type Al      Sec.(Top 81)
0003FF00-0003FFFF  00000100  DATA  N RW-- 00 ABS  ACC_MDATA
000FFB00-000FFBFF  00000100  CONST N R--I 00 ABS  ACC_UDATA
007E0000-007EFFFF  00010000  CONST N R--I 00 ABS  ACC_MCODE
```

Because normally the Accemic MDE handles clock startup, please remove clock setting section for startup. All clock settings will be made to monitor kernel.

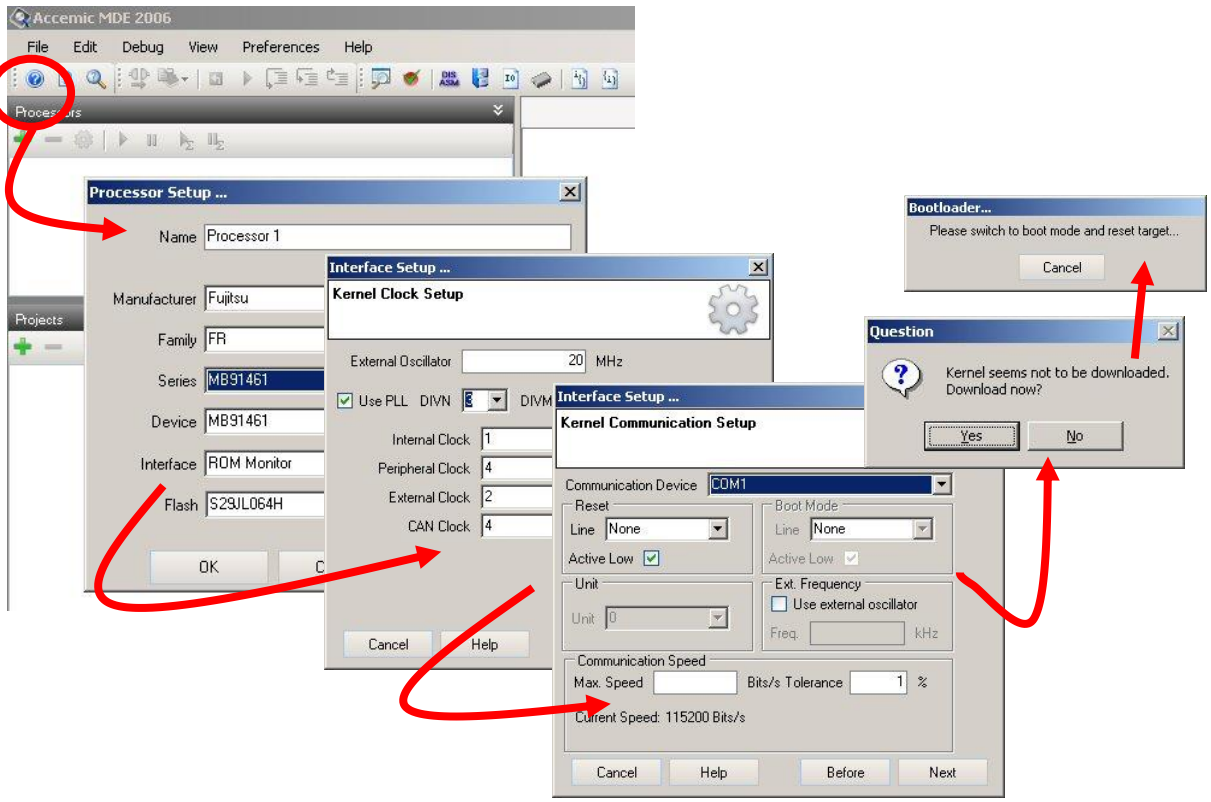
After you have build your project with the settings mentioned above, please check settings on starter kit shown below and start Accemic MDE software.

Table 9. Jumper and switches on SK91461 for Accemic MDE Monitor Debugger

Name (Switches)	Setting
RES	BRK
P15_3	*Don't care*
P15_2	*Don't care*
INT0	L
MD0	H
MD1	L
MD2	L
MD3	L
Name (Jumpers)	**Setting**
P21_0	Connection
P21_1	Connection
P15_3	*Don't care*
P15_2	*Don't care*

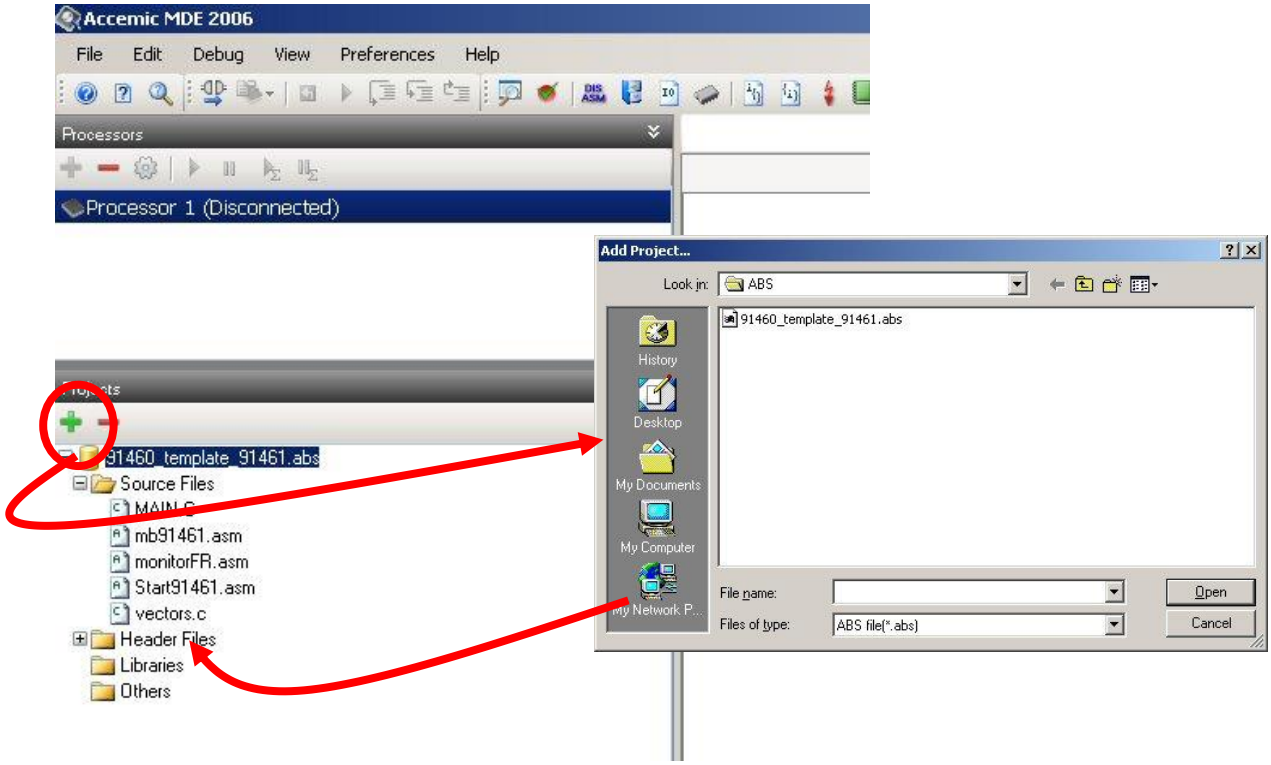
As first step you have to select processor type. Press '+' button in processor window to add MB91461. Processor Setup dialog opens to choose appropriate microcontroller. Make your selections and click 'OK' to get to the next menu. Here the clock settings for the monitor kernel are made. Press 'Next' and follow the menu to set for example communication COM port and access code. After all settings are finished, the monitor kernel has to be downloaded to MCU. Please set microcontroller to programming mode (MD[0:3] = L) and follow the instructions.

Figure 11. Download of monitor kernel



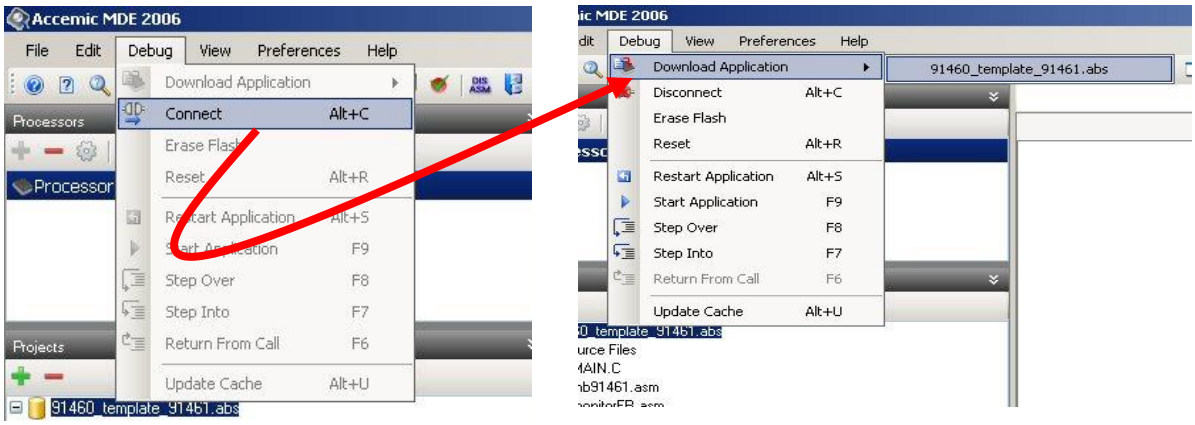
If the kernel download ended successfully, set microcontroller back to run mode (see table 9). Click '+' button in projects window and add your project.

Figure 12. Select project



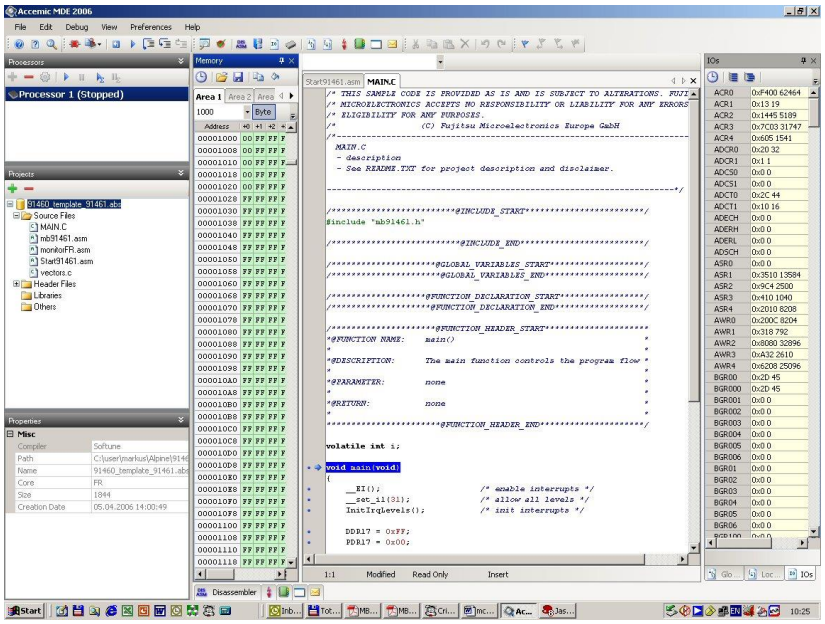
Start connection to the monitor kernel via 'Debug' -> 'Connect'. After connection is set up, download your application to the microcontroller ('Debug' -> 'Download Application' -> *your_project.abs*).

Figure 13. Connection and download of application



The source code window opens and current position of program counter is marked. Now you can step through the application (C-Code or assembler steps), run up to breakpoints, watch variables, view memory contents, modify IO-registers…

Figure 14. GUI of Accemic MDE Monitor Debugger



6 Softune Workbench Simulator

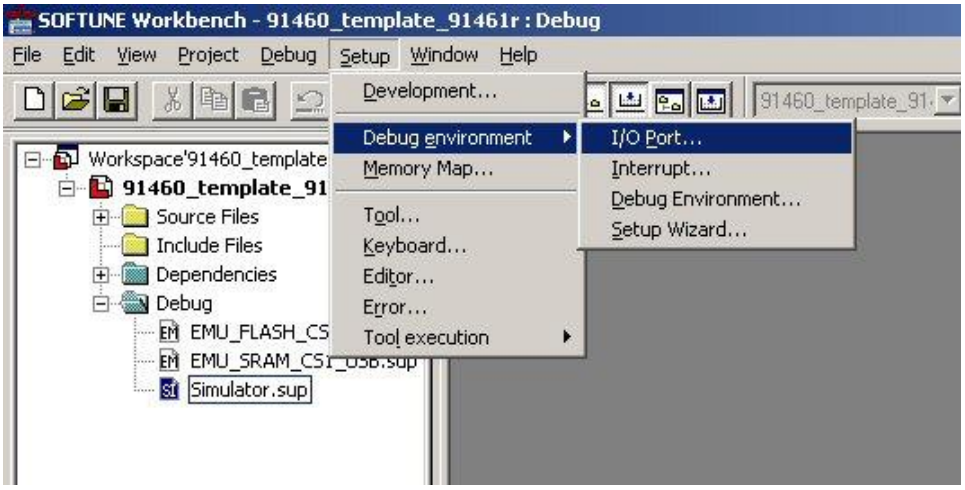
This chapter describes the debugging functionality of the Softune Workbench Simulator.

Softune Workbench includes a core simulator. This simulator can be used to debug all core related functions of your application. Resource functions are not supported.

Double click on Simulator.sup to start debugging session.

Beside step execution and use of breakpoints it is possible to simulate IO accesses and simulate resource interrupts. Therefore use menus ‘Setup’ -> ‘Debug Environment’ -> ‘I/O Port…’ or ‘Interrupt…”

Figure 15. SWB simulator



Please refer to Softune Workbench User Manual for detailed description of the functionality.

7 Comparison Table

This chapter gives a rough overview about needed internal or peripheral resources for debugging and debugging functionality of the different possibilities.

Debug Type / Resource/ Functionality	EVA-System with MB91V460	EVA-System with MB91461	SWB Monitor Debugger	Accemic MDE Debugger	SWB Simulator
Target Chip MB91461RA	-	X	X	X	-
Evaluation Chip MB91V460	X	-	-	-	-
Adapter Board MB2198-300	X	-	-	-	-
Header Board MB2198-301	X	-	-	-	-
Starterkit SK91461 (or own target)	with socket	X	X	X	-
External Flash connected to CS0 on target board	-	(X)*1	X	X	-
External RAM connected to CS1 on target board	-	(X)*1	X	-	-
UART 0 for communication to debugger	-	-	X	X	-
Code Breakpoints	X	X	X	X	X
Data Breakpoints	Use event breakpoint	Use event breakpoint	-	*2)	X
Data Match Breakpoint	Use event breakpoint	Use event breakpoint	-	*2)	-
Monitoring (memory/watch variables)	X	X	-	-	X
Trace	X	X	-	-	1000 instructions
Time measurement	X	X	-	-	Number of cycles

*1: At least one type (Flash or SRAM) is needed.

*2: Not supported with Accemic MDE version used for this application note, but feature under development.

8 Document History

Document Title: AN205206 - FR Family MB91460 Series MB91461R Emulation

Document Number: 002-05206

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	-	NOFL	04/27/2006	Initial release
			04/26/2007	updates based on template “91460_template_91461r-v14”
*A	5123278	NOFL	02/03/2016	Migrated Spansion Application Note from MCU-AN-300029-E-V11 to Cypress format
*B	5843233	AESATP12	08/03/2017	Updated logo and copyright.
*C	6054520	NOFL	02/05/2018	Updated links. Updated Sales page.

Worldwide Sales and Design Support

Cypress maintains a worldwide network of offices, solution centers, manufacturer's representatives, and distributors. To find the office closest to you, visit us at Cypress Locations.

Products

Arm® Cortex® Microcontrollers	cypress.com/arm
Automotive	cypress.com/automotive
Clocks & Buffers	cypress.com/clocks
Interface	cypress.com/interface
Internet of Things	cypress.com/iot
Memory	cypress.com/memory
Microcontrollers	cypress.com/mcu
PSoC	cypress.com/psoc
Power Management ICs	cypress.com/pmic
Touch Sensing	cypress.com/touch
USB Controllers	cypress.com/usb
Wireless Connectivity	cypress.com/wireless

PSoC® Solutions

PSoC 1 | PSoC 3 | PSoC 4 | PSoC 5LP | PSoC 6 MCU

Cypress Developer Community

Community | Projects | Videos | Blogs | Training | Components

Technical Support

cypress.com/support

All other trademarks or registered trademarks referenced herein are the property of their respective owners.

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709

© Cypress Semiconductor Corporation, 2006-2018. This document is the property of Cypress Semiconductor Corporation and its subsidiaries, including Spansion LLC ("Cypress"). This document, including any software or firmware included or referenced in this document ("Software"), is owned by Cypress under the intellectual property laws and treaties of the United States and other countries worldwide. Cypress reserves all rights under such laws and treaties and does not, except as specifically stated in this paragraph, grant any license under its patents, copyrights, trademarks, or other intellectual property rights. If the Software is not accompanied by a license agreement and you do not otherwise have a written agreement with Cypress governing the use of the Software, then Cypress hereby grants you a personal, non-exclusive, nontransferable license (without the right to sublicense) (1) under its copyright rights in the Software (a) for Software provided in source code form, to modify and reproduce the Software solely for use with Cypress hardware products, only internally within your organization, and (b) to distribute the Software in binary code form externally to end users (either directly or indirectly through resellers and distributors), solely for use on Cypress hardware product units, and (2) under those claims of Cypress's patents that are infringed by the Software (as provided by Cypress, unmodified) to make, use, distribute, and import the Software solely for use with Cypress hardware products. Any other use, reproduction, modification, translation, or compilation of the Software is prohibited.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS DOCUMENT OR ANY SOFTWARE OR ACCOMPANYING HARDWARE, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. No computing device can be absolutely secure. Therefore, despite security measures implemented in Cypress hardware or software products, Cypress does not assume any liability arising out of any security breach, such as unauthorized access to or use of a Cypress product. In addition, the products described in these materials may contain design defects or errors known as errata which may cause the product to deviate from published specifications. To the extent permitted by applicable law, Cypress reserves the right to make changes to this document without further notice. Cypress does not assume any liability arising out of the application or use of any product or circuit described in this document. Any information provided in this document, including any sample design information or programming code, is provided only for reference purposes. It is the responsibility of the user of this document to properly design, program, and test the functionality and safety of any application made of this information and any resulting product. Cypress products are not designed, intended, or authorized for use as critical components in systems designed or intended for the operation of weapons, weapons systems, nuclear installations, life-support devices or systems, other medical devices or systems (including resuscitation equipment and surgical implants), pollution control or hazardous substances management, or other uses where the failure of the device or system could cause personal injury, death, or property damage ("Unintended Uses"). A critical component is any component of a device or system whose failure to perform can be reasonably expected to cause the failure of the device or system, or to affect its safety or effectiveness. Cypress is not liable, in whole or in part, and you shall and hereby do release Cypress from any claim, damage, or other liability arising from or related to all Unintended Uses of Cypress products. You shall indemnify and hold Cypress harmless from and against all claims, costs, damages, and other liabilities, including claims for personal injury or death, arising from or related to any Unintended Uses of Cypress products.

Cypress, the Cypress logo, Spansion, the Spansion logo, and combinations thereof, WICED, PSoC, CapSense, EZ-USB, F-RAM, and Traveo are trademarks or registered trademarks of Cypress in the United States and other countries. For a more complete list of Cypress trademarks, visit cypress.com. Other names and brands may be claimed as property of their respective owners.