FR Family FR81S Series, Emulation System

Doc. No. 002-04887 Rev. *A

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709
http://www.cypress.com

Copyrights

Trademarks

Source Code

Disclaimer

CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Cypress reserves the right to make changes without further notice to the materials described herein. Cypress does not assume any liability arising out of the application or use of any product or circuit described herein. Cypress does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress' product in a life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges.

Contents

1. Introduction

The chapter gives short overview of this document.

This document describes the first steps how to use the FR81S emulation system. It describes also the software and hardware installation procedure.

Further information can be found in the following manuals:

- ‘SOFTUNE Workbench Installation Manual for V6’ for information about the installation procedure of the software
- ‘MB2100-01-E Hardware Manual’ (SS01-00026-1E.pdf) for information about the embedded emulator
- Specific user guides for Starter Kit, e.g. FMEMCU- UG-910092-10 for SK-91580-001-MC
- Short version explanation for getting started FR81_tools_ReadMe.pdf

The FR81S emulation system allows the designer immediately to start with the software development before his own final target system is available.

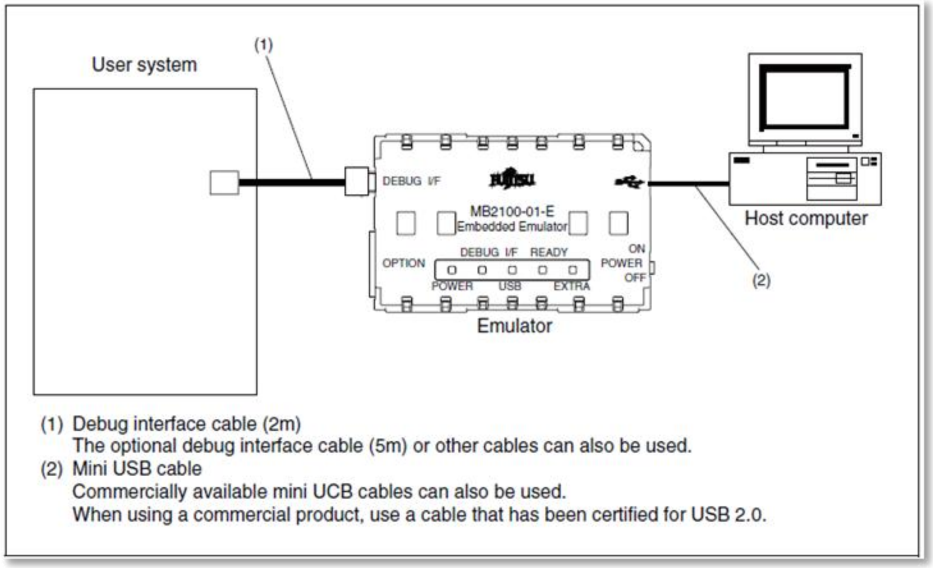
2. Hardware Requirements

The chapter lists the required hardware for the MB91460 emulation system.

The FR81S emulator system consists of

- The embedded emulator MB2100-01-E including mini USB cable for power supply
- Debug interface cable
- FR81S StarterKit, e.g. SK-91580-001-MC

Figure 2-1. General overview of MB2100-01-E connections



(1) Debug interface cable (2m)
The optional debug interface cable (5m) or other cables can also be used.
(2) Mini USB cable
Commercially available mini UCB cables can also be used.
When using a commercial product, use a cable that has been certified for USB 2.0.

3. Software Installation

This chapter describes the software installation procedure

This installation procedure is valid for Windows 7/Vista/XP/2000. For other operating systems and for further details, please refer to the SOFTUNE Workbench installation manual for V6 and to the hardware manual of the Emulator MB2100-01-E

3.1 Notes on Win7 operating system

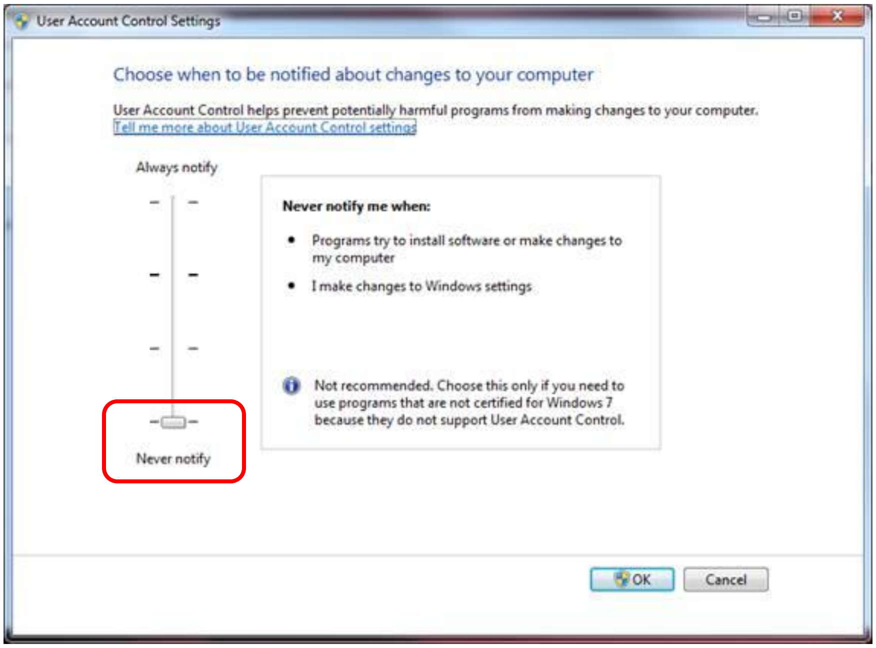
3.1.1 User Account Control (UAC) function of Windows 7

When the UAC function is enabled, the USB driver will not be installed automatically. Therefore, the UAC function must be disabled when installing SOFTUNE Workbench.

To disable the UAC function, click on [Control Panel] > [User Accounts and Family Safety] > [User Account] > [User Account Control settings].

Then move the Slider to Never Notify (see Figure 3-1)

Figure 3-1. UAC Settings Window



User Account Control Settings

Choose when to be notified about changes to your computer

User Account Control helps prevent potentially harmful programs from making changes to your computer.

Tell me more about User Account Control settings

Never notify me when:

- Programs try to install software or make changes to my computer
- I make changes to Windows settings

Not recommended. Choose this only if you need to use programs that are not certified for Windows 7 because they do not support User Account Control.

The UAC function prevents programs that require administrator privileges from being installed automatically into the system by a virus, illegal operation, operational error or other factors.

Therefore, it is recommended to turn the function back on after installing SOFTUNE.

3.1.2 MultiVersionManager

When MultiVersionManager (MVM) is used, it is necessary to turn off UAC. (Both installation, uninstallation, and use)

3.2 General Issues before Installation

Before installation of the Softune Workbench following issues must be done:

- If a former Softune Workbench version was installed, please uninstall this version separately and use remove the old registry files by using a registry cleaner tool (e.g. CCleaner).
- When several Softune Workbench propack versions ought to be installed, it is recommended to install the Softune MultiVersionManager tool (see chapter 3.3.1).
- For installation local administrator rights are required.
  - WinXP: after successful installation user can change the settings from “administrator mode” to “user mode”
  - Win7: Additionally turn of UAC feature (see chapter 3.1)
- When MultiVersionManager (MVM) is used, it is necessary to turn off UAC. (Both installation, uninstallation, and use)
- When using different MCU families (8/16/32bit MCUs) different installation paths for different propack versions for the corresponding MCU families must be applied.
  - 8bit MCU: SoftuneV3\896s-rev3xxxxx
  - 16bit MCU: SoftuneV3\907s-rev3xxxxx
  - 32bit MCU: SoftuneV6\911s-rev6xxxxx

Note: revision number is given in the installer name of the corresponding SOFTUNE installer version.

- The updated USB driver *wdapi1021.dll* file for Win7 is replaced by every new installation. After an installation please rename *wdapi1021.dll* to *wdapi1021_old.dll* and copy *wdapi1021_32.dll into “C:\Windows\sysWOW64” and change filename to wdapi1021.dll.*

3.3 SOFTUNE Workbench

There are two possibilities to install SOFTUNE Workbench.

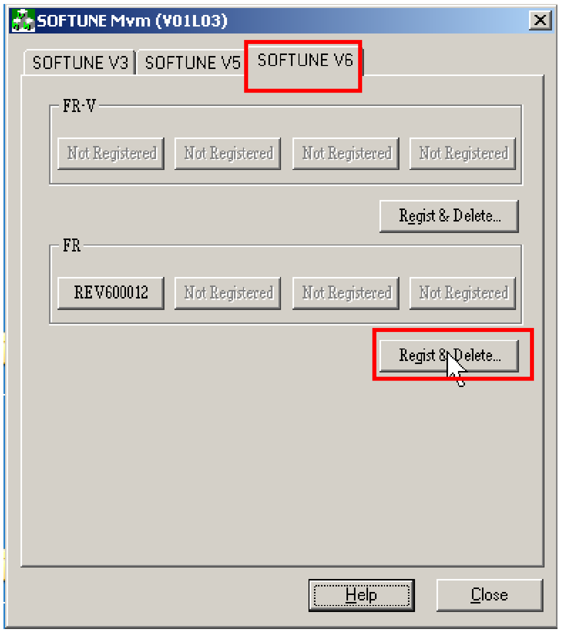
- The standard installation by execution of *setup.exe*.
- Installation of the MultiVersionManager.

Advantage of the MultiVersionManager: different propack versions of one MCU family can be installed in parallel. But take into account, that different installation paths must be used.

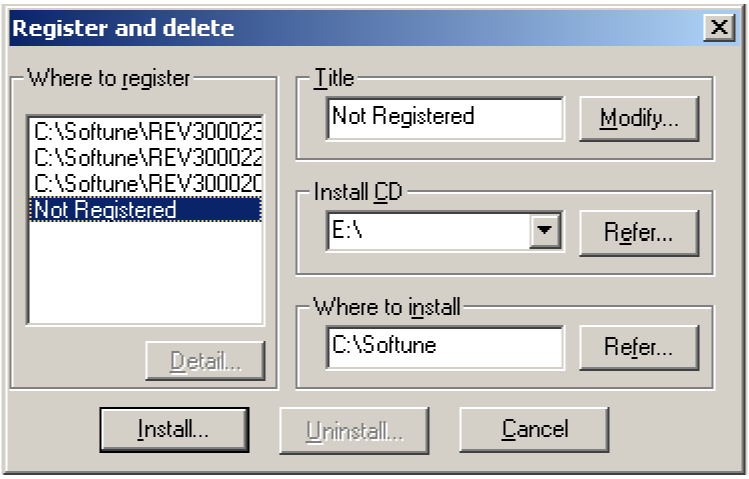
3.3.1 Installation with MultiVersionManager

To use different SOFTUNE propack versions of one MCU family the MultiVersionManager must be installed. In the following steps the installation is explained:

- If MultiVersionManager is uninstalled, execute its installation file: CD_Image\SOFTUNE MVM\SOFTUNE_MVM_V0xL0x.exe
- After successful installation, open the tool in Windows Start [Start] > [Programs] > [SOFTUNE MultiVersionManager]
- To install a SOFTUNE propack version, click on [SOFTUNE V6] tab > [Regist & Delete…] button.



- Select in the [Where to register] any “Not Registered” line for installation and enter the setup.exe path of the corresponding propack (e.g. “..\911s-rev600012\CD_Image”.

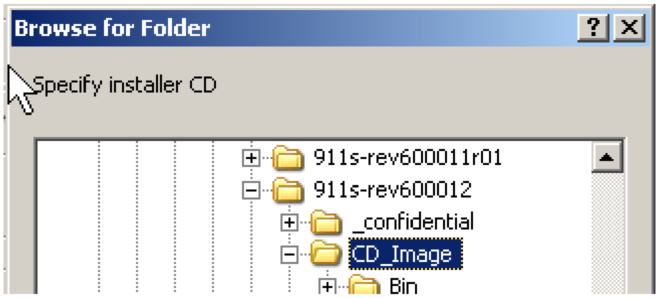


Where to register:
C:\Softune\REV30002З
C:\Softune\REV300022
C:\Softune\REV30002C
Not Registered

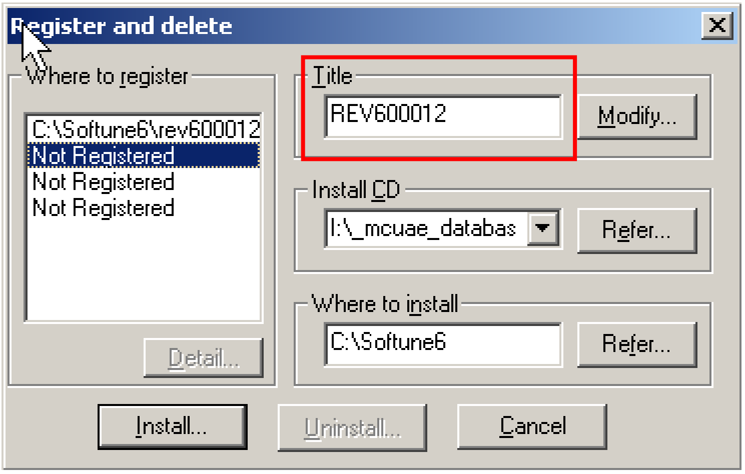
Title: Not Registered

Install CD: E:\

Where to install: C:\Softune



The [Title] text field is filled automatically by the SOFTUNE revision number.



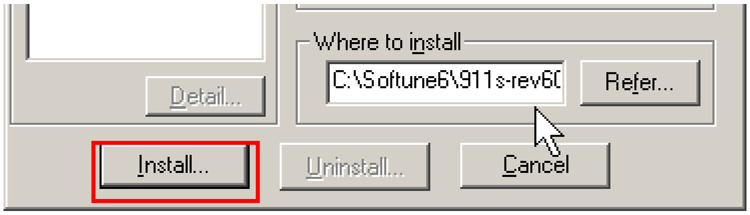
Where to register:
C:\Softune6\rev600012
Not Registered
Not Registered
Not Registered

Title: REV600012

Install CD: I:_mcuae_databas

Where to install: C:\Softune6

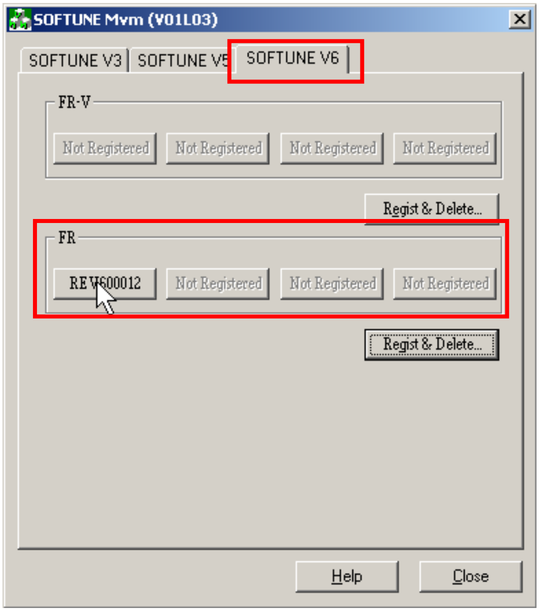
- Enter an installation path, which is only valid for this propack version C:\Softune6\911s-revXXXXXX.



Where to install: C:\Softune6\911s-rev6(

- Press on [Install..] button to execute installation.
- Installation of SOFTUNE Workbench is finished. For starting the tool it will be found under [Start] > [Programs] > [SOFTUNE MultiVersionManager] > [SOFTUNE MultiVersionManager].

For uninstallation of one version use also the MultiVersionManager



FR-V: Not Registered | Not Registered | Not Registered | Not Registered

FR: REV600012 | Not Registered | Not Registered | Not Registered

- After successful installation you can open an installed propack by selection of the [SOFTUNE V6] tab and press on enabled button in the [FR] line

The updated USB driver *wdapi1021.dll* file for Win7 is replaced by every new installation. After an installation please rename *wdapi1021.dll* to *wdapi1021_old.dll* and copy *wdapi1021_32.dll into "C:\Windows\sysWOW64" and change filename to wdapi1021.dll.*

3.3.2 Installation without MultiVersionManager

The development with Cypress MCU needs SOFTUNE Workbench as Integrated Development Environment (IDE).

The following instructions are necessary for the installation of SOFTUNE Workbench:

Note 1: To install the software under Windows 7/Vista/XP/2000, the user requires administration authority. Please refer to chapter 3.1 .

- On the CD image you will find a folder „SOFTUNE"
- Unpack the „CD_Image" from **minimum** Propack911s-rev600011.zip to a local device on your computer
- After the unpacking, execute „Setup.exe" to install SOFTUNE Workbench
- Follow the instructions of the setup program
- Select a specific path for the installation, e.g. C:Program Files/ SOFTUNE6
- With the installation a SOFTUNE6 folder will be created
- When finished the installation, check for following files:

  a. 911.csv

  b. FSDIO200.csv

  c. Folder „Flash"

from following link:

http://www.cypress.com/16lx

The updated USB driver *wdapi1021.dll* file for Win7 is replaced by every new installation. After an installation please rename *wdapi1021.dll* to *wdapi1021_old.dll* and copy *wdapi1021_32.dll into "C:\Windows\sysWOW64" and change filename to wdapi1021.dll.*

4. Installation of the Emulation System

This chapter describes, what should be considered and checked before power-up the emulation system. Also the start-up procedure of the emulation system is described.

In order to properly install the emulation system, follow the following steps.

4.1 Features MB2100-01-E

For emulation and debugging of the FR81S devices the embedded emulator MB2100-01-E is used.

In the next chapters the features of the MB2100-01-E embedded emulator are explained. For further explanation refer to ‘MB2100-01-E Hardware Manual’ (SS01-00026-1E.pdf)

4.1.1 Checking the delivered product

Before using the product, confirm that the following components are included in the box:

- 1x Emulator unit
- 1x USB cable (USB-ECOM520: Elecom Co. Ltd.)
- 1x Debug interface cable (2m) (MB2100-201-E)
- 1x Coaxial connector
- 1x (HRM-300-126B (40): HIROSE ELECTRIC Co., Ltd.)
- 1x Operation manual (English version, this manual)
- 1x Operation manual (Japanese version)

4.1.2 Appearance and part names

In Figure 2-1 and Table 4-1 the external appearance and component names of the embedded emulator MB2100-01-E are shown.

Figure 4-1. External Appearance of MB2100-01-E

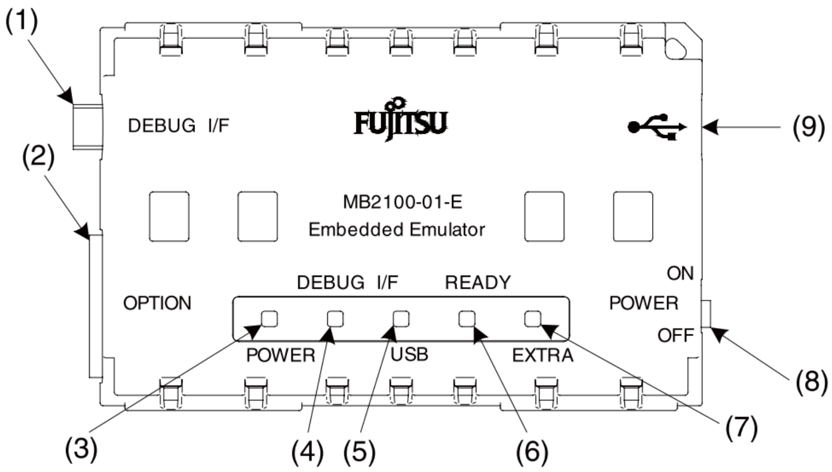


Table 4-1. External Appearance of Embedded Emulator MB2100-01-E

No.	Name	Description
(1)	Debug interface connector	Connects to the user system using the debug interface cable. Connector part number: HRM-200-2S-C(40) from HIROSE ELECTRIC Co., Ltd.
(2)	Option connector	Connector for connecting to optional components. This connector also doubles as the product maintenance port. Take care to avoid touching metal parts of this connector with your hand or other objects when the connector is not connected.
(3)	POWER LED	This light is on when the power is on and off when the power is off. The light is off when the USB suspended.
(4)	DEBUG I/F LED	Indicates the communication status of the debug interface bus. The light turns on during communication.
(5)	USB LED	Indicates the communication status of the USB bus. The light turns on during communication.
(6)	READY LED	This light turns on when the emulator and the host computer are able to communicate.
(7)	EXTRA LED	This light turns on when operating with various functions of the debugging software. See the SOFTUNE Workbench Operation Manual of the debugging software for details.
(8)	POWER switch	Turns the power to the emulator on and off.
(9)	USB connector	Connects to the host computer using a mini USB cable.

The debug interface cable is a cable with SMA coaxial connectors that uses a 1.5D-2V (characteristic impedance 50Ω) wire core. The maximum permissible communication speed from the MCU to the emulator is **50Mbps** regardless of the cable length. In Table 4-2. Cable Length and Communication Speed for Embedded Emulator MB210100-01-E

the cable length and maximum communication speed for emulator is shown.

Table 4-2. Cable Length and Communication Speed for Embedded Emulator MB210100-01-E

Cable length	Maximum permissible communication speed, Emulator side
2 m or less	25 Mbps
5 m or less	12.5Mbps
10 m or less	6.25 Mbps

4.1.3 General Specifications

In the following table the main specifications for the emulator are shown.

Table 4-3. Emulator specifications

Item	Specification
Name	Emulator
Part number	MB2100-01-E
Power supply	5V / 500mA (USB bus powered)
Debug interface bus communication scheme	Single wire UART (NRZ or phase-modulated encoding). The encoding scheme used is determined by the MCU.
Host computer communication scheme	USB 2.0 compliant
USB communication speed (theoretical value)	480 Mbps (Hi-Speed),12 Mbps (Full-Speed)
Operating temperature	+5 °C to + 40 °C
Operating humidity	20 % to 80 % (no condensation)
Storage temperature	0 °C to +70 °C
Storage temperature	20 % to 80 % (no condensation)
External dimensions	84.8 (D) × 53.6 (W) × 21.3 (H) mm (not including connectors, switches, or other protruding parts.)
Weight	70g

4.2 Connecting and installation of MB2100-01-E

The following steps have to be established for getting the embedded emulator MB2100-01-E.

Attention: The USB driver for Windows 7 operating system must be installed separately. The USB drivers must be downloaded from

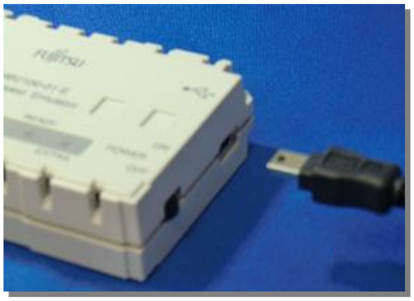
http://www.cypress.com/16lx

Please follow the instructions of the USB drive installation guide, before executing next steps

Installation Steps:

- Pick up the USB cable which is packed with the emulator.
- Plug in the mini-USB interface of the USB cable into emulator

Figure 4-2. Connection to MiniUSB Interface of Emulator

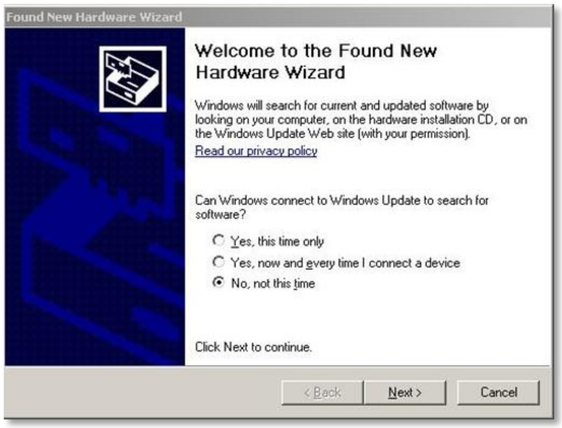


- Connect the counter side of the USB cable to your host computer
- Windows will “Found New Hardware: MB2100-01-E” and the windows hardware wizard should start automatically

Note2: The installation procedure may differ with different operating systems.

- Do not connect to Windows Updates to search for Software

Figure 4-3. Windows Hardware Wizard Found New Hardware (1)



Found New Hardware Wizard

Welcome to the Found New Hardware Wizard

Windows will search for current and updated software by looking on your computer, on the hardware installation CD, or on the Windows Update Web site (with your permission).

Read our privacy policy

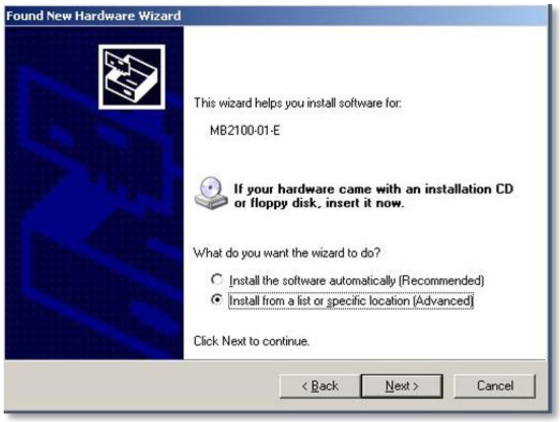
Can Windows connect to Windows Update to search for software?

○ Yes, this time only
○ Yes, now and every time I connect a device
● No, not this time

Click Next to continue.

- Select: install from a list or specific location

Figure 4-4. Windows Hardware Wizard Found New Hardware (2)



Found New Hardware Wizard

This wizard helps you install software for:

MB2100-01-E

If your hardware came with an installation CD or floppy disk, insert it now.

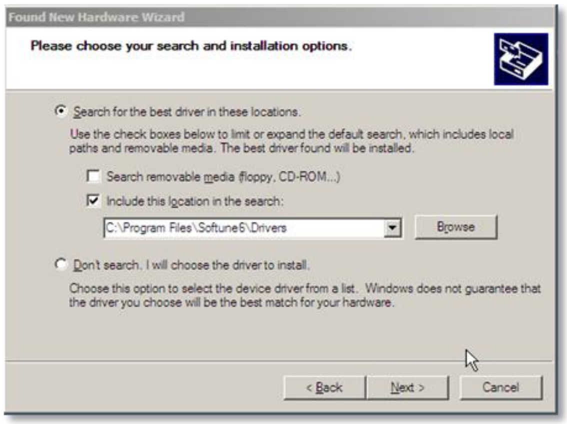
What do you want the wizard to do?

○ Install the software automatically (Recommended)
● Install from a list or specific location (Advanced)

Click Next to continue.

- Select „search for the best driver“
- The driver path can be found under ../Softune6/Drivers

Figure 4-5. Windows Hardware Wizard Found New Hardware (3)



Please choose your search and installation options.

● Search for the best driver in these locations.

Use the check boxes below to limit or expand the default search, which includes local paths and removable media. The best driver found will be installed.

☐ Search removable media (floppy, CD-ROM...)

☑ Include this location in the search:

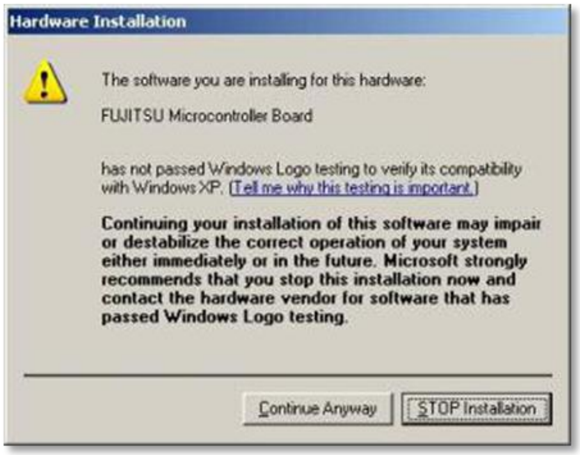
C:\Program Files\Softune6\Drivers

○ Don't search. I will choose the driver to install.

Choose this option to select the device driver from a list. Windows does not guarantee that the driver you choose will be the best match for your hardware.

- Click Next for installing driver
- Maybe a window occurs for installation software, please click on „Continue Anyway“

Figure 4-6. Windows Hardware Wizard Found New Hardware (4)



The software you are installing for this hardware:

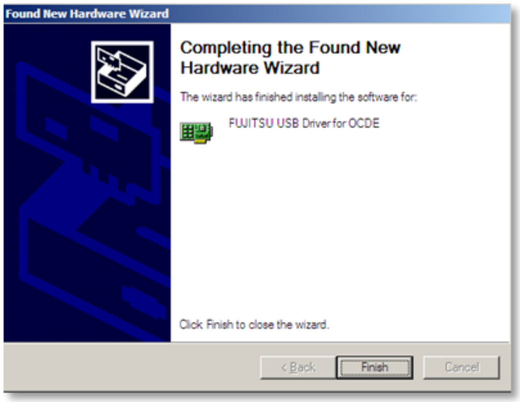
FUJITSU Microcontroller Board

has not passed Windows Logo testing to verify its compatibility with Windows XP. (Tell me why this testing is important.)

Continuing your installation of this software may impair or destabilize the correct operation of your system either immediately or in the future. Microsoft strongly recommends that you stop this installation now and contact the hardware vendor for software that has passed Windows Logo testing.

- Windows completing the installation by copying some files „Finish“ to close the window

Figure 4-7. Windows Hardware Wizard Found New Hardware (5)



Completing the Found New Hardware Wizard

The wizard has finished installing the software for:

FUJITSU USB Driver for OCDE

Click Finish to close the wizard.

- The hardware installed and ready to be used

Note3: The Software driver is necessary for using the MB2100-01-E embedded emulator. If the automatically installation of the driver was not finished successfully, you should install it manually.

Note4: There is an emulator update available since November 2010. If you have not yet updated the emulator Software (SW), follow these steps:

- Please contact Cypress to get the SW update files
- Unpack the Softune6_MB2100_20101105.zip to a local storage
- Copy the files Wv911e3.dll and SiDRVo.dll from folder "bin" into your local SOFTUNE folder ..\Softune6\Bin
- Copy the file Lib\911\EmlOcd_911.MHX and into your local..\Softune6\Lib.

With starting the Softune Workbench environment the next time an update is generated. This is an update of the MB2100 emulator SW. Please do not interrupt the update process!

- Pick up the co-axial cable which is packed with the emulator
- Connect the emulator and the StarterKit with the co-axial cable to the SMA connector

Figure 4-8. Debug Cable Connection to Emulator

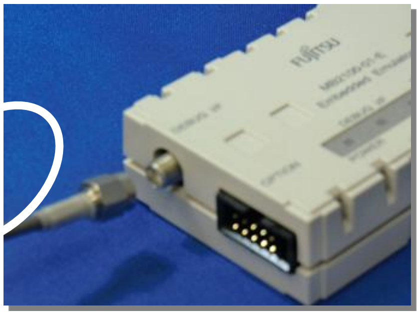
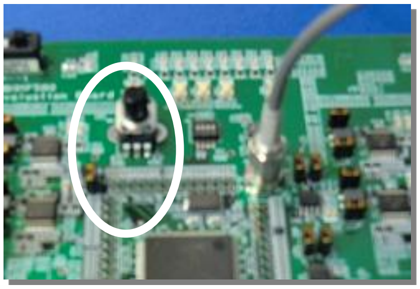


Figure 4-9. Debug Cable Connection to StarterKit



The hardware setup is finished. Keep Power SW off.

5. SOFTUNE Workbench Getting Started

This chapter gives an introduction how to get started with Cypress SOFTUNE Workbench.

5.1 New Project in SOFTUNE Workbench

PREPARING A NEW PROJECT

This chapter describes how to set up a new project in Cypress SOFTUNE Workbench.

5.1.1 Getting template project

In order to start a new user project use the template project for the specific devices. Sample Software is available from CD image or under following link:

http://www.cypress.com/16lx

A specific template project can be downloaded from web site. This project includes the startup code, header files and short descriptions of the usage.

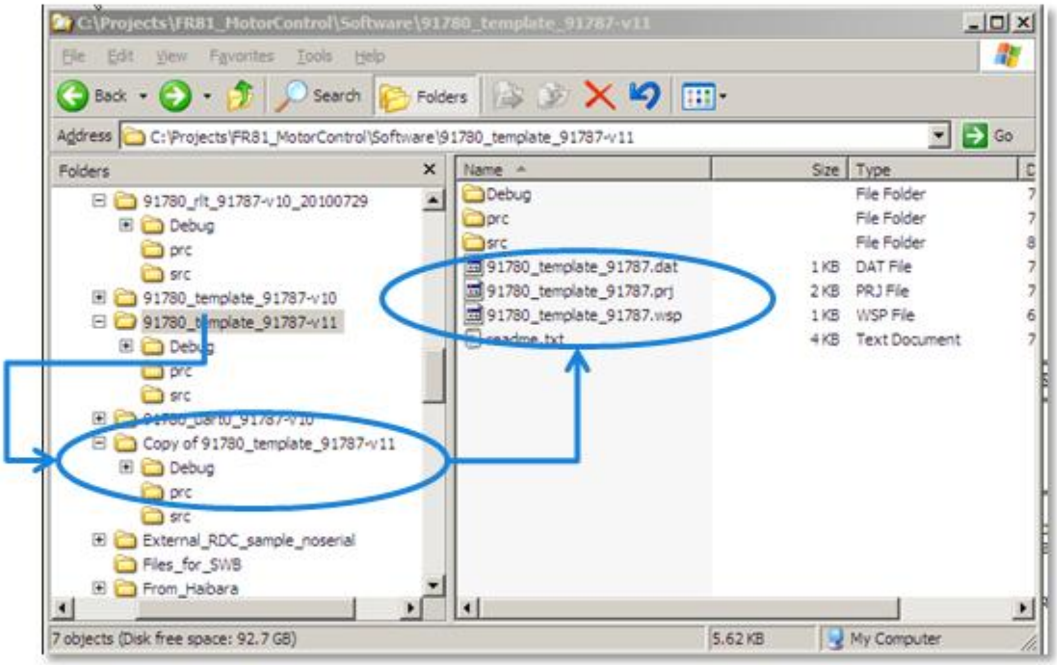
Note5: Please always read the readme.txt file for the explanation of the SW sample project.

5.1.2 Setup own project

Follow the next steps to create your own application project (this is an example for MB91F787 template project)

- Save the “91780_template_91787-v11” template project to a local storage.
- Copy the folder within in the local storage and rename ‘Copy of 91780_template_91787-v11’ into ‘91780_my_application_91787-v10’

Figure 5-1. Renaming of Project

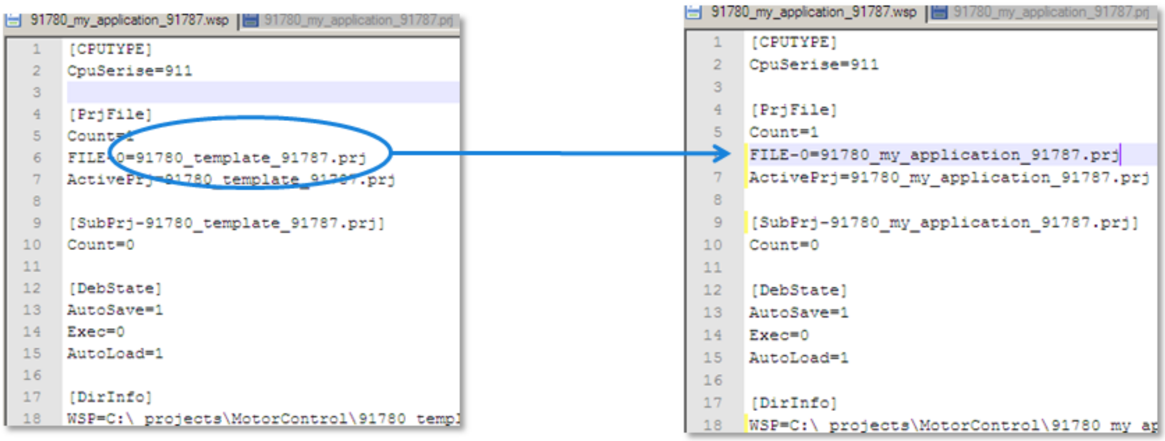


C:\Projects\FR81_MotorControl\Software\91780_template_91787-v11
File Edit View Favorites Tools Help
Back
Search
Folders
Address C:\Projects\FR81_MotorControl\Software\91780_template_91787-v11
Go
Folders
91780_rlt_91787-v10_20100729
Debug
prc
src
91780_template_91787-v10
91780_template_91787-v11
Debug
prc
src
91780_uart0_91787-v10
Copy of 91780_template_91787-v11
Debug
prc
src
External_RDC_sample_noserial
Files_for_SWB
From_Haibara
7 objects (Disk free space: 92.7 GB)
5.62 KB
My Computer

Name	Size	Type
Debug		File Folder
prc		File Folder
src		File Folder
91780_template_91787.dat	1 KB	DAT File
91780_template_91787.prj	2 KB	PRJ File
91780_template_91787.wsp	1 KB	WSP File
readme.txt	4 KB	Text Document

- Enter „91780_my_application_91787-v10“ folder
- Rename „91780_template_91787.prj“ into „91780_my_application_91787.prj“
- Rename „91780_template_91787.wsp“ into „91780_my_application_91787.wsp“
- Rename „91780_template_91787.dat“ into „91780_my_application_91787.dat“
- Edit „my_application.prj“ by using an editor program
  - ☐ Rename all „template“ into „my_application
- Edit „my_application.wsp“ by using an editor program
  - ☐ Rename all „template“ into „my_application

Figure 5-2. Edit .wsp and .prj file



91780_my_application_91787.wsp
91780_my_application_91787.prj
91780_my_application_91787.wsp
91780_my_application_91787.prj

```
[CPUTYPE]
CpuSerise=911

[PrjFile]
Count=1
FILE-0=91780_template_91787.prj
ActivePrj=91780_template_91787.prj

[SubPrj-91780_template_91787.prj]
Count=0

[DebState]
AutoSave=1
Exec=0
AutoLoad=1

[DirInfo]
WSP=C:\ projects\MotorControl\91780_templ
```

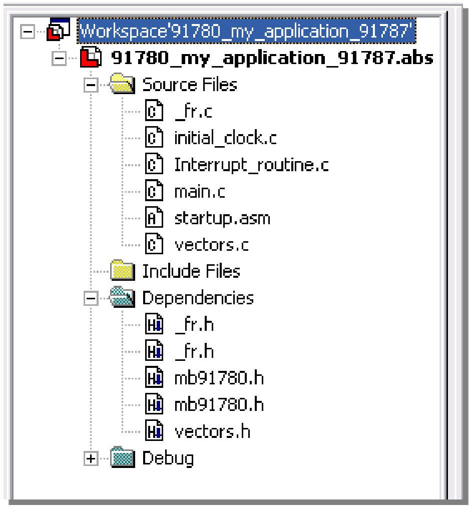
```
[CPUTYPE]
CpuSerise=911

[PrjFile]
Count=1
FILE-0=91780_my_application_91787.prj
ActivePrj=91780_my_application_91787.prj

[SubPrj-91780_my_application_91787.prj]
Count=0

[DebState]
AutoSave=1
Exec=0
AutoLoad=1

[DirInfo]
WSP=C:\ projects\MotorControl\91780_my_ap
```

- Start Softune Workbench and open the project via File -> Open Workspace…
- In the browse window please go to the local path off the project and select my_application.wsp

Note6: It is also possible to start the Softune Workbench and open the workspace via Drag&Drop into the Softune window

- The project opens and the project structure will be displayed on the left side (explorer view) of the SOFTUNE screen such as shown in Figure 5-3.

Figure 5-3. Project structure in SOFTUNE Workbench

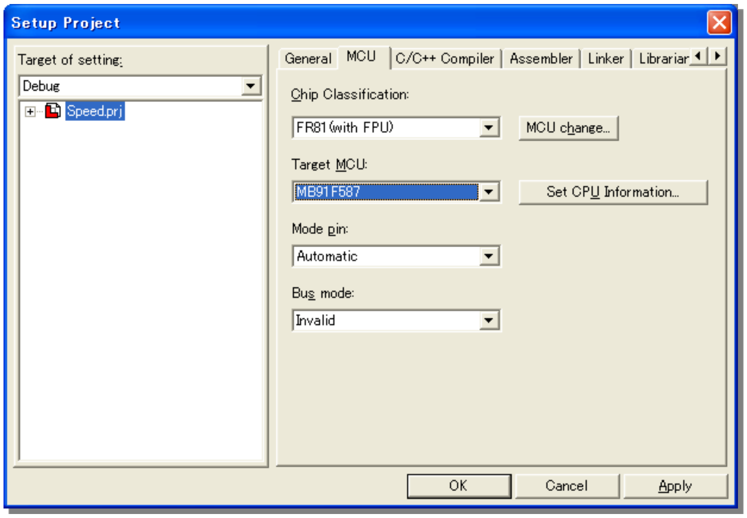


```
Workspace'91780_my_application_91787'
  91780_my_application_91787.abs
    Source Files
      _fr.c
      initial_clock.c
      Interrupt_routine.c
      main.c
      startup.asm
      vectors.c
    Include Files
    Dependencies
      _fr.h
      _fr.h
      mb91780.h
      mb91780.h
      vectors.h
    Debug
```

The top level is the Workspace itself: "my_application". The next level contains the ABS-File of the Project, which contains the compiled program data. The source files, their dependencies and the debugger settings follow.

- Do not miss the Target MCU selection at the project setting, when opening the project development, (the project setting can be checked by "MCU" tab of Setup Project Window which is opened by the selecting [Project] -> [Setup Project].)

Figure 5-4. Setup MCU



Target of setting: Debug

Speed.prj

Chip Classification: FR81 (with FPU)

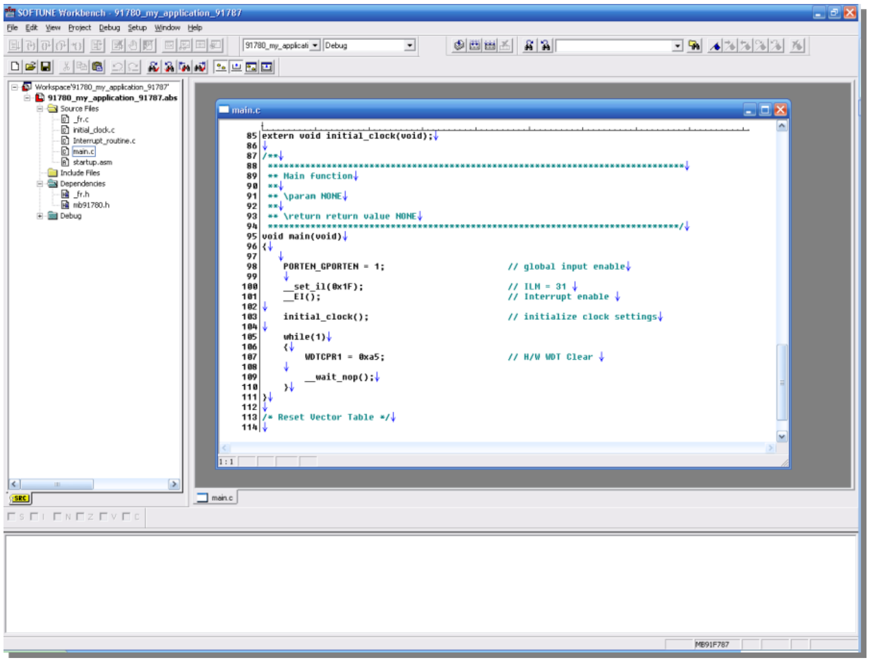
Target MCU: MB91F587

Mode pin: Automatic

Bus mode: Invalid

- For editing the source code files, open them by double-click on the file, e.g. main.c
- Now a text editor window will open, which shows the contents of "Main.c". It will look like this:

Figure 5-5. Main.c in Editor Mode


SOFTUNE Workbench - 91780_my_application_91787
File Edit View Project Debug Setup Window Help
91780_my_applicati
Debug
Workspace'91780_my_application_91787'
91780_my_application_91787.abs
Source Files
_fr.c
initial_clock.c
Interrupt_routine.c
main.c
startup.asm
Include Files
Dependencies
_fr.h
mb91780.h
Debug
main.c
1 : 1
MB91F787

```
extern void initial_clock(void);

/**
 ********************************************************************************
 ** Main function
 **
 ** \param NONE
 **
 ** \return return value NONE
 *******************************************************************************/
void main(void)
{

    PORTEN_GPORTEN = 1;                         // global input enable

    __set_il(0x1F);                             // ILM = 31
    __EI();                                     // Interrupt enable

    initial_clock();                            // initialize clock settings

    while(1)
    {
        WDTCPR1 = 0xa5;                         // H/W WDT Clear

        __wait_nop();
    }
}

/* Reset Vector Table */

```

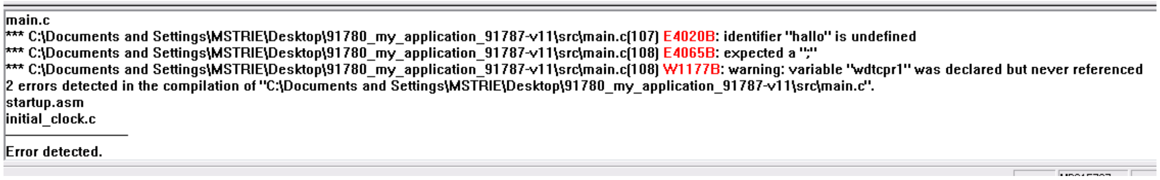
5.1.3 Entering code for own application

The development can begin by editing application code into the files. For modular coding style you can add own *.c files by right mouse click on folder “Source Files” in explorer view. Header file (*.h) must not be added to project, they will be included as dependencies. Interrupt definition is done in the file “vectors.c”.

5.1.4 Compiling the project

- When finished coding, the project can be build by the buttons or by selecting [Project] -> [Build] or [Make]
- In the information window below edit window the state of compiling is shown. In case of warning or errors, this is shown here.

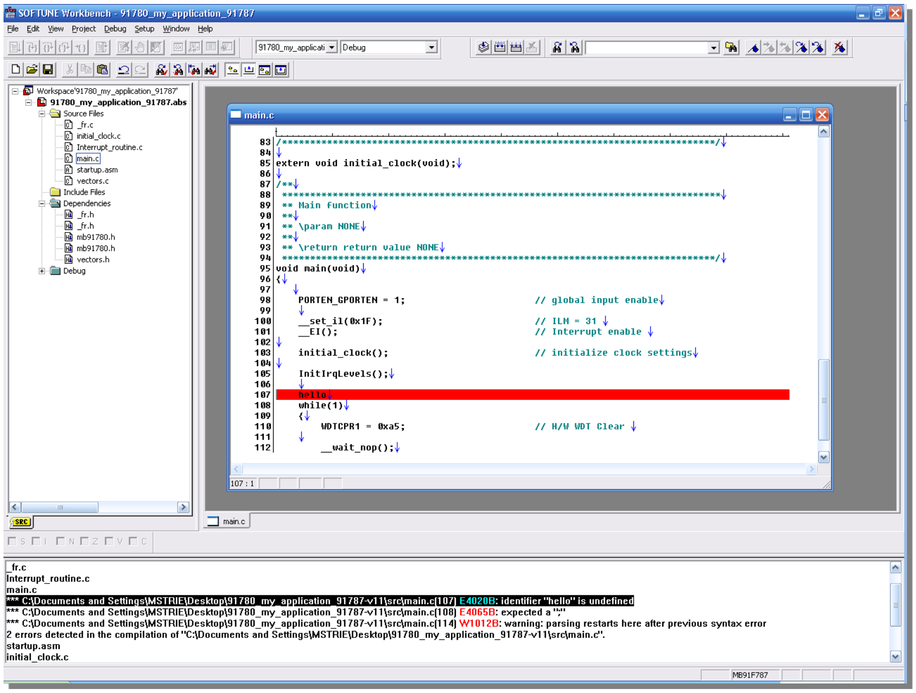
Figure 5-6. Error Window after Compiling



```
main.c
*** C:\Documents and Settings\MSTRIE\Desktop\91780_my_application_91787-v11\src\main.c(107) E4020B: identifier "hallo" is undefined
*** C:\Documents and Settings\MSTRIE\Desktop\91780_my_application_91787-v11\src\main.c(108) E4065B: expected a ";"
*** C:\Documents and Settings\MSTRIE\Desktop\91780_my_application_91787-v11\src\main.c(108) W1177B: warning: variable "wdtcpr1" was declared but never referenced
2 errors detected in the compilation of "C:\Documents and Settings\MSTRIE\Desktop\91780_my_application_91787-v11\src\main.c".
startup.asm
initial_clock.c

Error detected.
```

- With double-click on the error notification the SOFTUNE opens the file and indicates the line of error such as shown

Figure 5-7. Error Indication of SOFTUNE Workbench



```
 83 /*************************************************************************/
 84 
 85 extern void initial_clock(void);
 86 
 87 /**
 88  *************************************************************************
 89  ** Main function
 90  **
 91  ** \param NONE
 92  **
 93  ** \return return value NONE
 94  *************************************************************************/
 95 void main(void)
 96 {
 97 
 98     PORTEN_GPORTEN = 1;                         // global input enable
 99 
100     __set_il(0x1F);                             // ILM = 31
101     __EI();                                     // Interrupt enable
102 
103     initial_clock();                            // initialize clock settings
104 
105     InitIrqLevels();
106 
107     hello
108     while(1)
109     {
110         WDTCPR1 = 0xa5;                         // H/W WDT Clear
111 
112         __wait_nop();
```

```
_fr.c
Interrupt_routine.c
main.c
*** C:\Documents and Settings\MSTRIE\Desktop\91780_my_application_91787-v11\src\main.c(107) E4020B: identifier "hello" is undefined
*** C:\Documents and Settings\MSTRIE\Desktop\91780_my_application_91787-v11\src\main.c(108) E4065B: expected a ";"
*** C:\Documents and Settings\MSTRIE\Desktop\91780_my_application_91787-v11\src\main.c(114) W1012B: warning: parsing restarts here after previous syntax error
2 errors detected in the compilation of "C:\Documents and Settings\MSTRIE\Desktop\91780_my_application_91787-v11\src\main.c".
startup.asm
initial_clock.c
```

When finished the compiling process with “No Error” the debugging process can be started.

5.1.5 Linker Settings

5.1.5.1 Fill Option

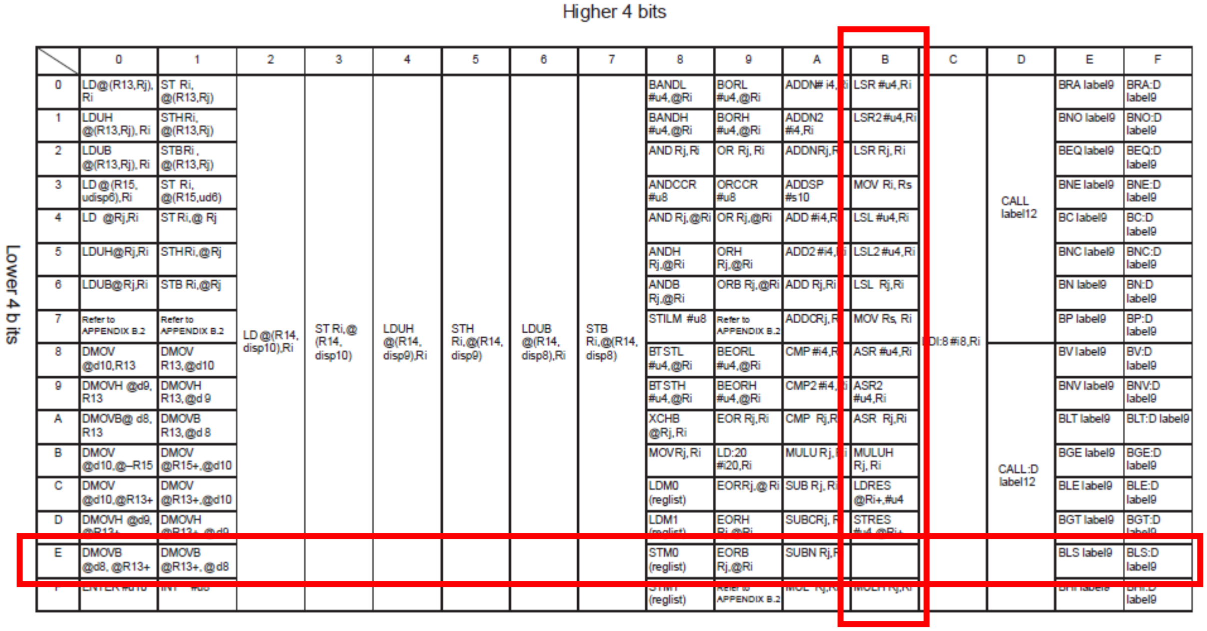
By employment of the –fill option unused area can be filled with a pattern.

There two use cases:

- Fill RAM area with pattern for stack overflow check
- Fill unused ROM area for illegal code instruction detection

For second case in general it is recommended to search for 1-byte instruction in “instruction maps” within the programming manual. In the Figure 5-8 one 1-byte instruction map is shown, which that the instruction “BE” is illegal.

Figure 5-8. Instruction Map from Programming Manual



Higher 4 bits

Lower 4 bits

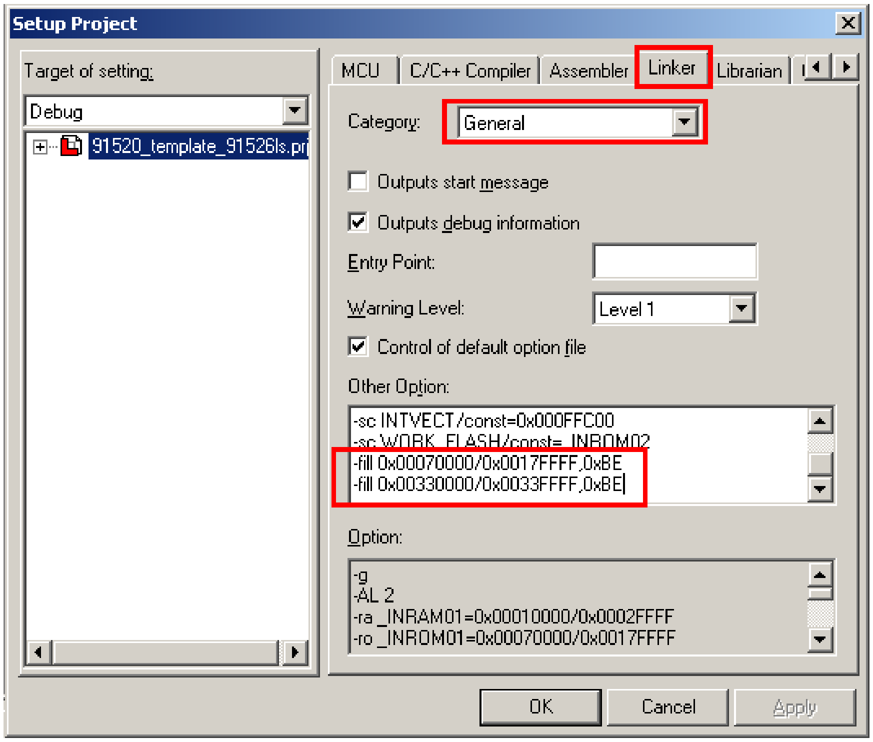
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	LD@(R13,Rj), Ri	ST Ri, @(R13,Rj)	LD @(R14, disp10),Ri	ST Ri,@ (R14, disp10)	LDUH @(R14, disp9),Ri	STH Ri,@(R14, disp9)	LDUB @(R14, disp8),Ri	STB Ri,@(R14, disp8)	BANDL #u4,@Ri	BORL #u4,@Ri	ADDN# i4,Ri	LSR #u4,Ri	LDI:8 #i8,Ri	CALL label12	BRA label9	BRA:D label9
1	LDUH @(R13,Rj), Ri	STHRi, @(R13,Rj)							BANDH #u4,@Ri	BORH #u4,@Ri	ADDN2 #i4,Ri	LSR2 #u4,Ri			BNO label9	BNO:D label9
2	LDUB @(R13,Rj), Ri	STBRi , @(R13,Rj)							AND Rj, Ri	OR Rj, Ri	ADDNRj,Ri	LSR Rj, Ri			BEQ label9	BEQ:D label9
3	LD @(R15, udisp6),Ri	ST Ri, @(R15,ud6)							ANDCCR #u8	ORCCR #u8	ADDSP #s10	MOV Ri, Rs			BNE label9	BNE:D label9
4	LD @Rj,Ri	ST Ri,@ Rj							AND Rj,@Ri	OR Rj,@Ri	ADD #i4,Ri	LSL #u4,Ri			BC label9	BC:D label9
5	LDUH@Rj,Ri	STHRi,@Rj							ANDH Rj,@Ri	ORH Rj,@Ri	ADD2 #i4,Ri	LSL2 #u4,Ri			BNC label9	BNC:D label9
6	LDUB@Rj,Ri	STB Ri,@Rj							ANDB Rj,@Ri	ORB Rj,@Ri	ADD Rj,Ri	LSL Rj,Ri			BN label9	BN:D label9
7	Refer to APPENDIX B.2	Refer to APPENDIX B.2							STILM #u8	Refer to APPENDIX B.2	ADDCRj,Ri	MOV Rs, Ri			BP label9	BP:D label9
8	DMOV @d10,R13	DMOV R13,@d10							BTSTL #u4,@Ri	BEORL #u4,@Ri	CMP #i4,Ri	ASR #u4,Ri		CALL:D label12	BV label9	BV:D label9
9	DMOVH @d9, R13	DMOVH R13,@d 9							BTSTH #u4,@Ri	BEORH #u4,@Ri	CMP2 #i4,Ri	ASR2 #u4,Ri			BNV label9	BNV:D label9
A	DMOVB@ d8, R13	DMOVB R13,@d 8							XCHB @Rj,Ri	EOR Rj,Ri	CMP Rj,Ri	ASR Rj,Ri			BLT label9	BLT:D label9
B	DMOV @d10,@–R15	DMOV @R15+,@d10							MOVRj,Ri	LD:20 #i20,Ri	MULU Rj,Ri	MULUH Rj, Ri			BGE label9	BGE:D label9
C	DMOV @d10,@R13+	DMOV @R13+,@d10							LDM0 (reglist)	EORRj,@ Ri	SUB Rj, Ri	LDRES @Ri+,#u4			BLE label9	BLE:D label9
D	DMOVH @d9, @R13+	DMOVH @R13+, @d9							LDM1 (reglist)	EORH Rj,@Ri	SUBCRj, Ri	STRES #u4,@Ri+			BGT label9	BGT:D label9
E	DMOVB @d8, @R13+	DMOVB @R13+, @d8							STM0 (reglist)	EORB Rj,@Ri	SUBN Rj,Ri				BLS label9	BLS:D label9
F	ENTER #u10	INT #u8							STM1 (reglist)	Refer to APPENDIX B.2	MUL Rj,Ri	MULH Rj,Ri			BHI label9	BHI:D label9

To fill unused Rom area in Softune Workbench following steps must be done:

Enter [Project] > [Setup Project] > tab [Linker], select category “General” and enter within “Other Option” text field the –fill option for 1-byte fill pattern as follows (see Figure 5-9):

-fill <start address>/<end address>,<pattern>

Figure 5-9. Fill Pattern Settings



Target of setting: Debug

91520_template_91526ls.pr

Category: General

☐ Outputs start message

☑ Outputs debug information

Entry Point:

Warning Level: Level 1

☑ Control of default option file

Other Option:

```
-sc INTVECT/const=0x000FFC00
-sc WORK_FLASH/const=_INROM02
-fill 0x00070000/0x0017FFFF,0xBE
-fill 0x00330000/0x0033FFFF,0xBE
```

Option:

```
-g
-AL 2
-ra _INRAM01=0x00010000/0x0002FFFF
-ro _INROM01=0x00070000/0x0017FFFF
```

After confirmation of new linker settings in *vectors.c* one more interrupt vector must be added for illegal instruction interrupt (see below) and the corresponding interrupt handler must be defined.

After finishing these steps the .abs file will be filled with the pattern in unused ROM areas and the illegal instruction handling can be checked even during debug session.

```
#pragma  intvect     Vectors_Isr_Illegal_InstructionHandler 14         ///<
Illegal Instruction

.
.

.

__interrupt void Vectors_Isr_Illegal_InstructionHandler  (void)

{

    // disable interrupts

    __DI();


    // halt system or wait for watchdog reset

    while(1)

    {

        __wait_nop();

    }

}
```

5.2 Debugging in SOFTUNE Workbench

Evaluation and Debugging with Softune Workbench

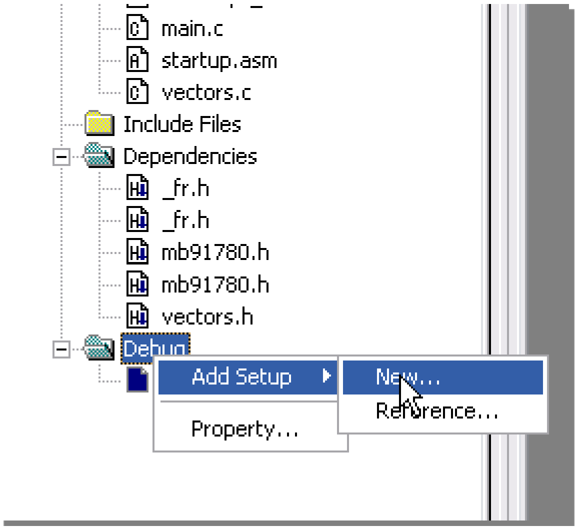
This chapter describes how to start debug a project in Cypress SOFTUNE Workbench.

5.2.1 Create and edit setup file

Before starting a debug session, some setups have to be done.

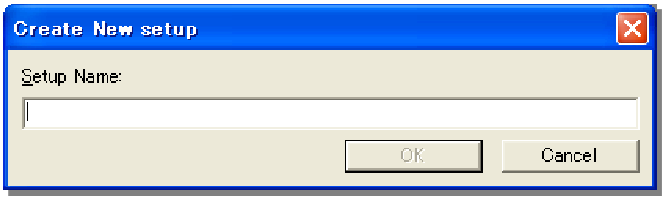
- Click the “Debug” of the explore window with right mouse click and select [Add Setup]->[New] with the right click

Figure 5-10. Setup New Debug Session



- Now it is possible to insert an own name for the setup file

Figure 5-11. Setup New Debug Session, Name



Note7: In case of existing setup file, you can also edit this by right mouse click and select [Edit]

- Follow the Setup Wizard and click on “Next” when following settings are done:

Table 5-1. Setup Items for Debugging

Item	Setup Value
Debug Type	Emulator Debugger
ICE Type	MB2100-01 (Default)
Monitor program Auto-loading	Checked
Device Type	USB
Oscillation Frequency	Depends on oscillation frequency (e.g. 4MHz)
Length DEBUG I/F cable	Depends on your cable length
Software Brake	Depend on your debugging
Specification batch file	Default: Not specified
Auto load	Checked
Batch file (before/after)	Default: Not specified e.g. insert a .prc file which should be loaded before or after loading file
Select setting item	Select All

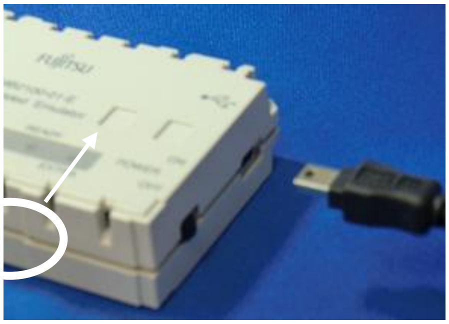
5.2.2 Start debug session

Before starting the debugging with On Chip Debugger, please check hardware connection which is described in the chapter “4.2 Connecting and installation of MB2100-01-E”

- Power on the emulator. The “Power” LED of Emulator turns on and also “READY” LED of Emulator turns on with blue colour.

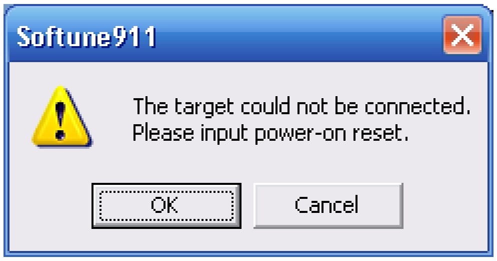
Note8: In this point, Board power keeps off

Figure 5-12. Power on of MB2100-01-E



- Start the debug session via [Debug] -> [Start Debug] from SOFTUNE GUI menu or by double-click on .sup file
- SOFTUNE asks for “Power on Reset”
- Power on the StarterKit before clicking “OK” on sub-window

Figure 5-13. "Power on Reset": Sub-window for



Softune911

The target could not be connected.
Please input power-on reset.

OK Cancel

- □ Debug mode is started

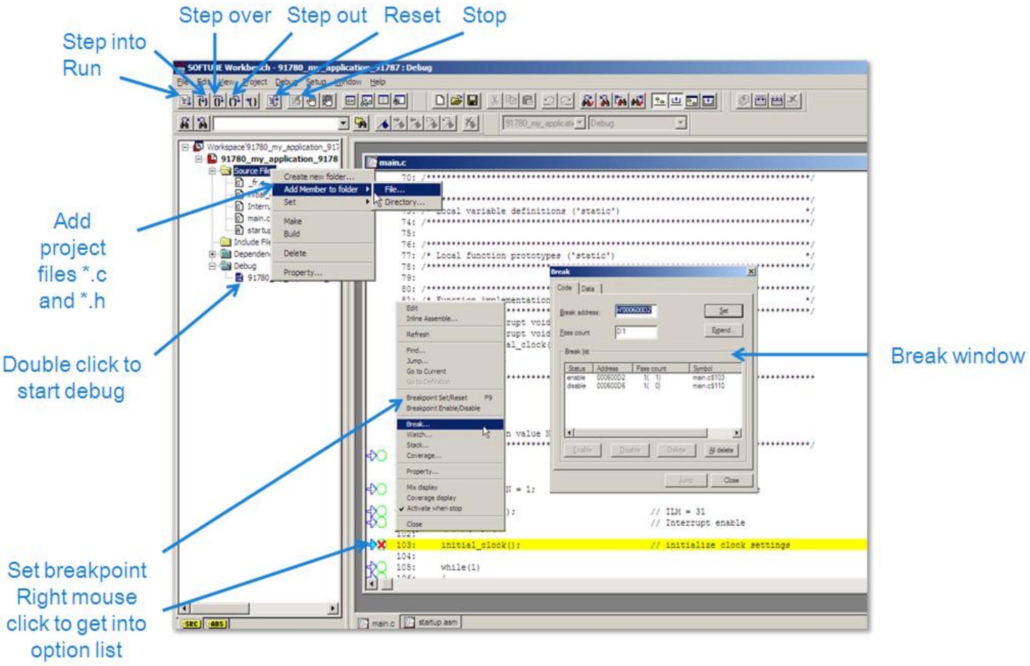
5.2.3 Executing and debugging the application

For executing and debugging your application some first introduction explanations follow.

- Select [Debug]->[Run]->[Go], press F5 or the button the execute/run your application
- For stopping [Debug]->[Abort], or the button. The emulator system now is being halted. The execution can be continued by entering the “Go”-Mode again.
- Set a breakpoint or sequence by pressing right mouse click into window (-> [Break]) or setting an red X such as shown in Figure 5-14.
- Select [View]->[Trace] for tracing the application
- Mark a variable and click right mouse button -> [Watch] to watch a variable value changed

In Figure 5-14 a short overview of the debug functionality is given. For further explanations refer to Cypress SOFTUNE Workbench feature description documents.

Figure 5-14. Short Overview for Debug Functionality

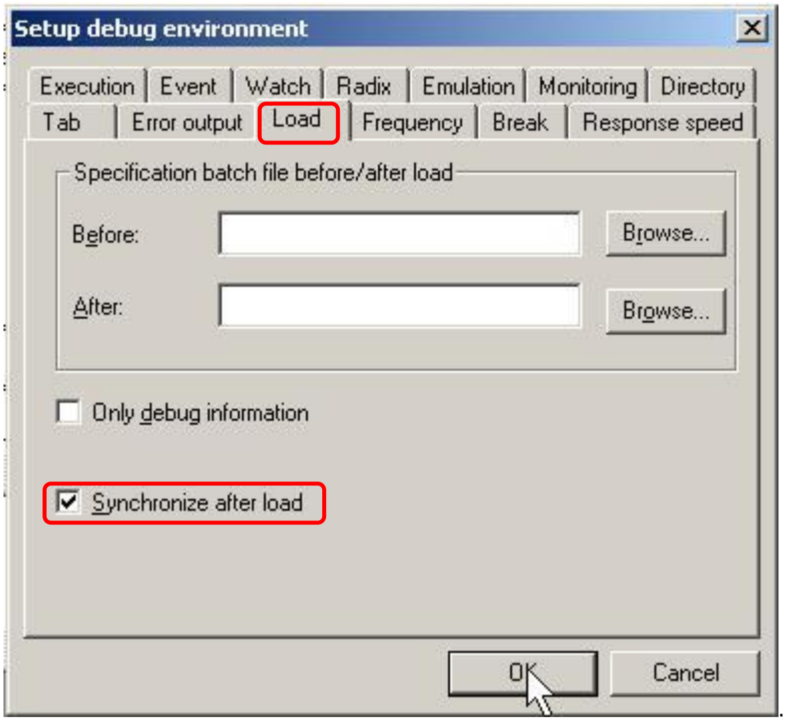


Step over
Step out
Reset
Stop
Step into
Run
Add project files *.c and *.h
Double click to start debug
Break window
Set breakpoint Right mouse click to get into option list

5.2.4 Automatic Flash Synchronisation

Usually during active debug session loading new compiled code is only possible, when the debug session is stopped and the “reset” button in the GUI is pressed. In order to synchronize the changed software code with the code inside the Flash automatically during stopped debug session, following setps must be done: Stop debug session, enter [Setup] > [Debug Environment] > [Debug Environment...], tab [Load] and enable the tick of the functionality [Synchronize after load] (see Figure 5-15). After that save the setting changes by close the window with the ‘OK’.

Figure 5-15. Flash Synchronisation Function in ‘Setup Debug Environment’ Window



Before:

After:

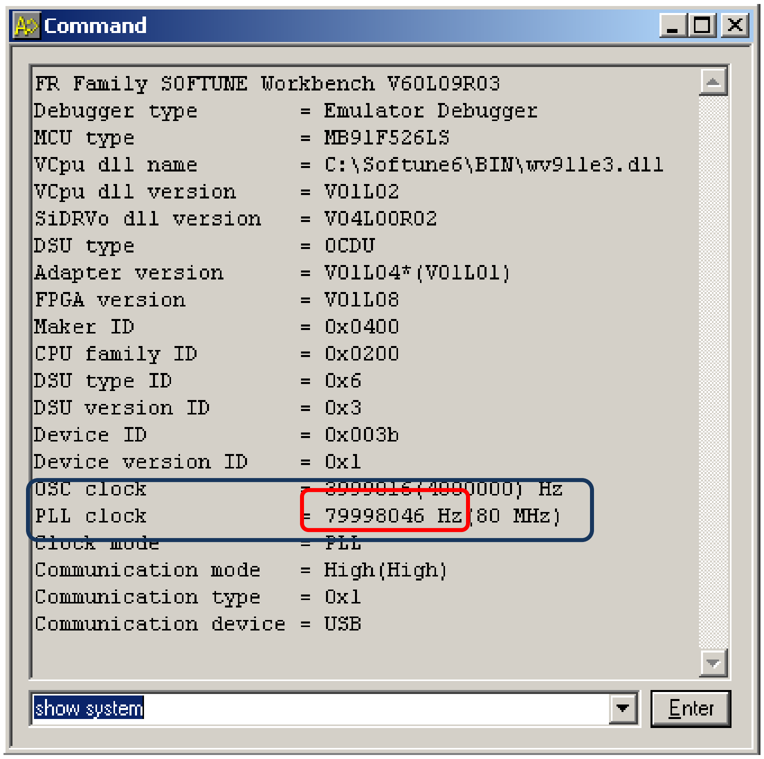
☐ Only debug information

☑ Synchronize after load

5.2.5 High Speed Mode

Usually the Standard clock is selected for the debug communication, which corresponds to the baud rate: Main clock frequency divided by 8 (500kBaud @ 4MHz Main clock). In addition to standard baud rate the high speed mode can be selected. The baud rate for communication direction from MCU to PC is rated to ‘PLLCK / 2’ (e.g. 40MHz baud rate @ 80MHz PLLCK clock).

Figure 5-16. Command Window with System Information



```
FR Family SOFTUNE Workbench V60L09R03
Debugger type          = Emulator Debugger
MCU type               = MB91F526LS
VCpu dll name          = C:\Softune6\BIN\wv911e3.dll
VCpu dll version       = V01L02
SiDRVo dll version     = V04L00R02
DSU type               = OCDU
Adapter version        = V01L04*(V01L01)
FPGA version           = V01L08
Maker ID               = 0x0400
CPU family ID          = 0x0200
DSU type ID            = 0x6
DSU version ID         = 0x3
Device ID              = 0x003b
Device version ID      = 0x1
OSC clock              = 3999016(4000000) Hz
PLL clock              = 79998046 Hz(80 MHz)
Clock mode             = PLL
Communication mode     = High(High)
Communication type     = 0x1
Communication device   = USB
```

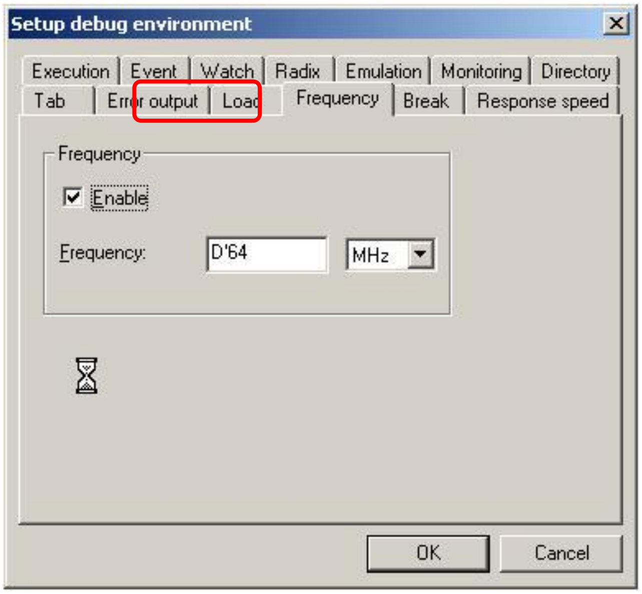
```
show system
```

To enter into high speed mode several steps must be executed:

Open the command windows as follows: [View] > [Command]. Enter “show system” into text field and some system information is prompted. In the line with text ‘PLL clock’ the left value marked in red shows the real VCO clock (e.g. ~80MHz in Figure 5-16). The right value corresponds to selected frequency of high speed mode.

After detection of the frequency the debug session must be stopped. Then the high speed mode can be enabled by clicking [Setup] > [Debug Environment] > [Debug Environment] > Tab [Frequency] > [Enable] (see Figure 5-17). Enter the integer value of the PLL clock command window line and press on ‘OK’ button.

Figure 5-17. High Speed Mode Selection in Setup Debug Evironment



Frequency

☑ Enable

Frequency: D'64 MHz

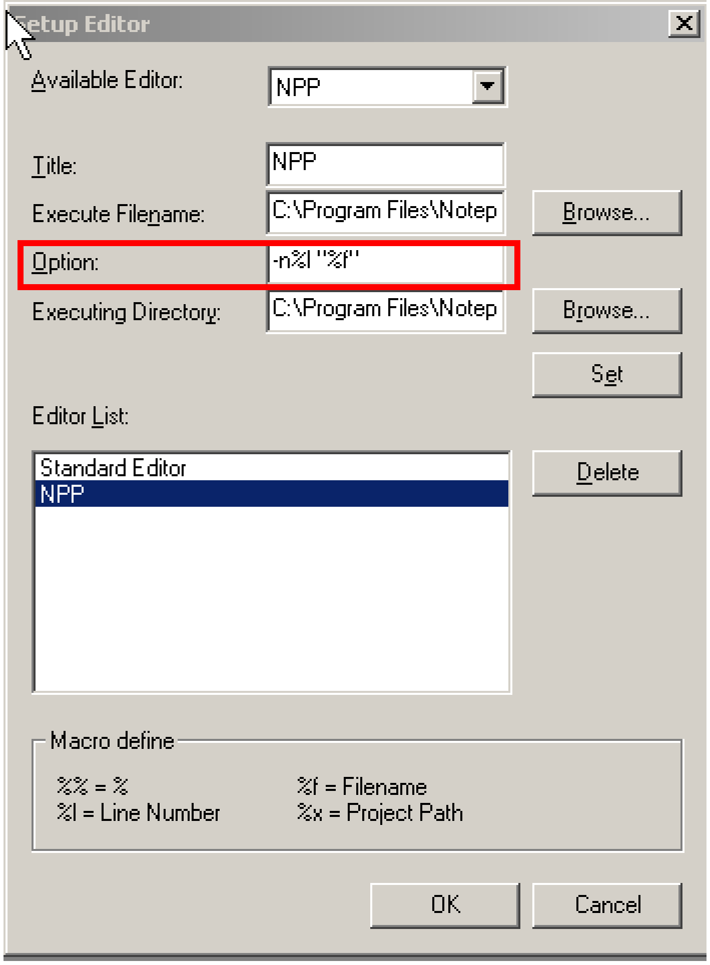
5.3 Integration of external text editor

Integration of external text editor is also supported, which can be used for programming source code and jumping to erroneous code line by double clicking error message in the output window.

Following settings must be done for selection of alternate editor:

1. Click on [Setup] > [Editor...]
2. Enter a title for text editor
3. Brows for the execution file (.exe) of editor
4. Enter option settings. -n%l "%f"
5. Select text editor folder as execution directory
6. Click on button [Set]
7. The select your new text editor in the selection box [Available Editor].
8. Click on [OK] for final confirmation

Figure 5-18. Text Editor Settings



Setup Editor

Available Editor: NPP

Title: NPP

Execute Filename: C:\Program Files\Notep

Option: -n%l "%f"

Executing Directory: C:\Program Files\Notep

Editor List:

Standard Editor
NPP

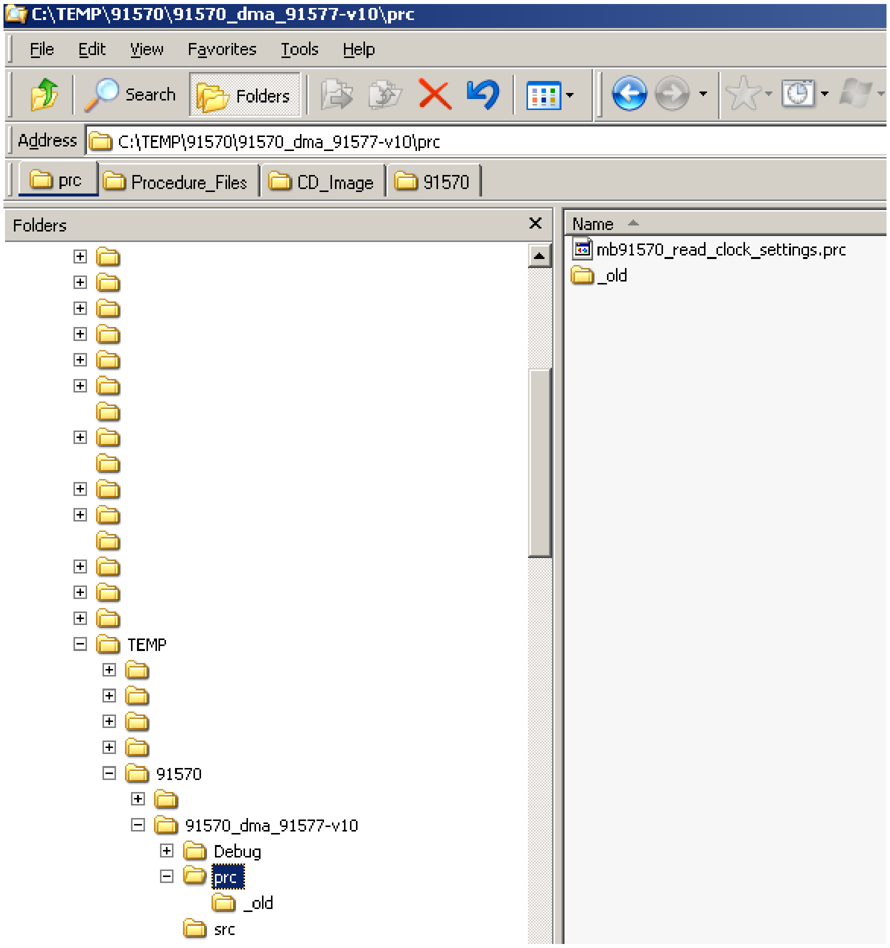
Macro define

%% = %
%l = Line Number
%f = Filename
%x = Project Path

5.4 Script Files (Procedure Files)

A procedure file is a batch file which can be executed at any time while the Softune Workbench is in Debug session, and can be used to control the debug session in order it follows the fixed list of commands previously programmed by the user.

We recommend that you store the procedure into the **/prc** folder of your Softune Workbench project.



Address C:\TEMP\91570\91570_dma_91577-v10\prc

Example of usage:

As an example let's suppose we want to execute a procedure file called **mb91570_read_clock_settings.prc** which needs two parameters (p1, p2).

In this case the parameters are the quartz xtals which are connected to the device:

Main xtal clock: 4 MHz, Sub-clock xtal: 32768 Hz

Then the procedure file input parameters are: P0 = 4000000, P1=32768

A maximum of 10 parameters are allowed (P0, …, P9)

This procedure file configures the system clocks and print-out the results in the command window. The clock settings which are configure with this procedure file are listed below:

CPU frequency to MCLK/2 = 2 MHz

External interface clock TCLK = 1 MHZ

Peripheral clocks PCLK PCLK2 = 500 KHz

To execute an existing procedure file follow these steps:

1. Open a debug session: **Debug-> Start Debug**
2. Open a Command Window: **View-> Command**
3. For executing the procedure files, two options are possible:

- If the procedure file has no input parameters, then just drag and drop the procedure file to the Softune Workbench environment while it is in Debug session (not the case of our example)
- If the procedure file has input parameters, in the command window type the following command sequence:

```
set radix/decimal
[Press/Click ENTER]
batch prc/mb91570_read_clock_settings.prc, 4000000, 32768
[Press/Click ENTER]
```

where:

batch: command to execute procedure files

prc: path inside the Softune Workbench project were the procedure file is stored

mb91570_read_clock_settings.prc: name of the procedure file

4000000: Parameter 0 (P0)

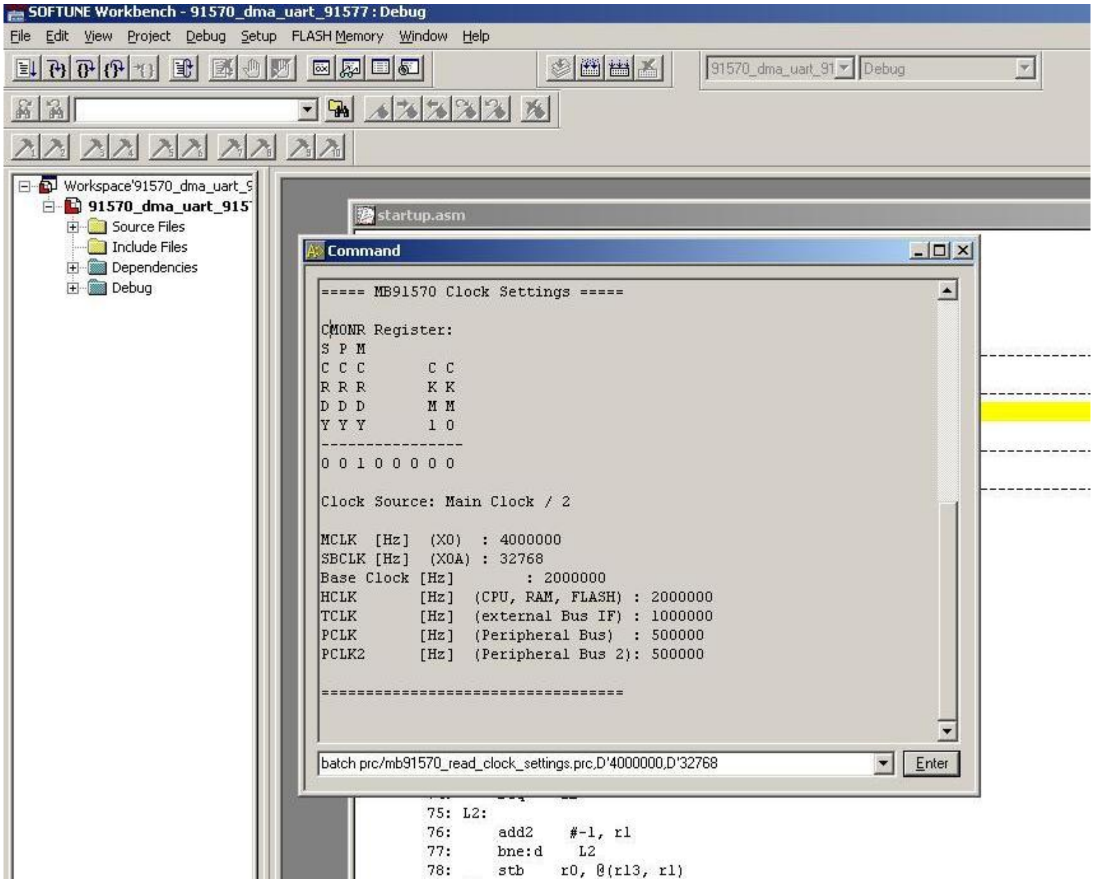
32768: Parameter 1 (P1)

If hexadecimal basis is preferred to enter the data, then enter the following command sequence:

```
set radix/hexadecimal
[Press/Click ENTER]
batch prc/mb91570_read_clock_settings.prc, 3D0900, 8000
[Press/Click ENTER]
```

Same results are expected. See the picture below for details:

Figure 5-19. Usage of Batch File in Command Window



SOFTUNE Workbench - 91570_dma_uart_91577 : Debug
File Edit View Project Debug Setup FLASH Memory Window Help
91570_dma_uart_91
Debug
Workspace'91570_dma_uart_9
91570_dma_uart_915
Source Files
Include Files
Dependencies
Debug
startup.asm
Command
Enter

```
===== MB91570 Clock Settings =====

CMONR Register:
S P M
C C C       C C
R R R       K K
D D D       M M
Y Y Y       1 0
----------------
0 0 1 0 0 0 0 0

Clock Source: Main Clock / 2

MCLK  [Hz]  (X0)  : 4000000
SBCLK [Hz]  (X0A) : 32768
Base Clock [Hz]        : 2000000
HCLK       [Hz]  (CPU, RAM, FLASH) : 2000000
TCLK       [Hz]  (external Bus IF) : 1000000
PCLK       [Hz]  (Peripheral Bus)  : 500000
PCLK2      [Hz]  (Peripheral Bus 2): 500000

==================================
```

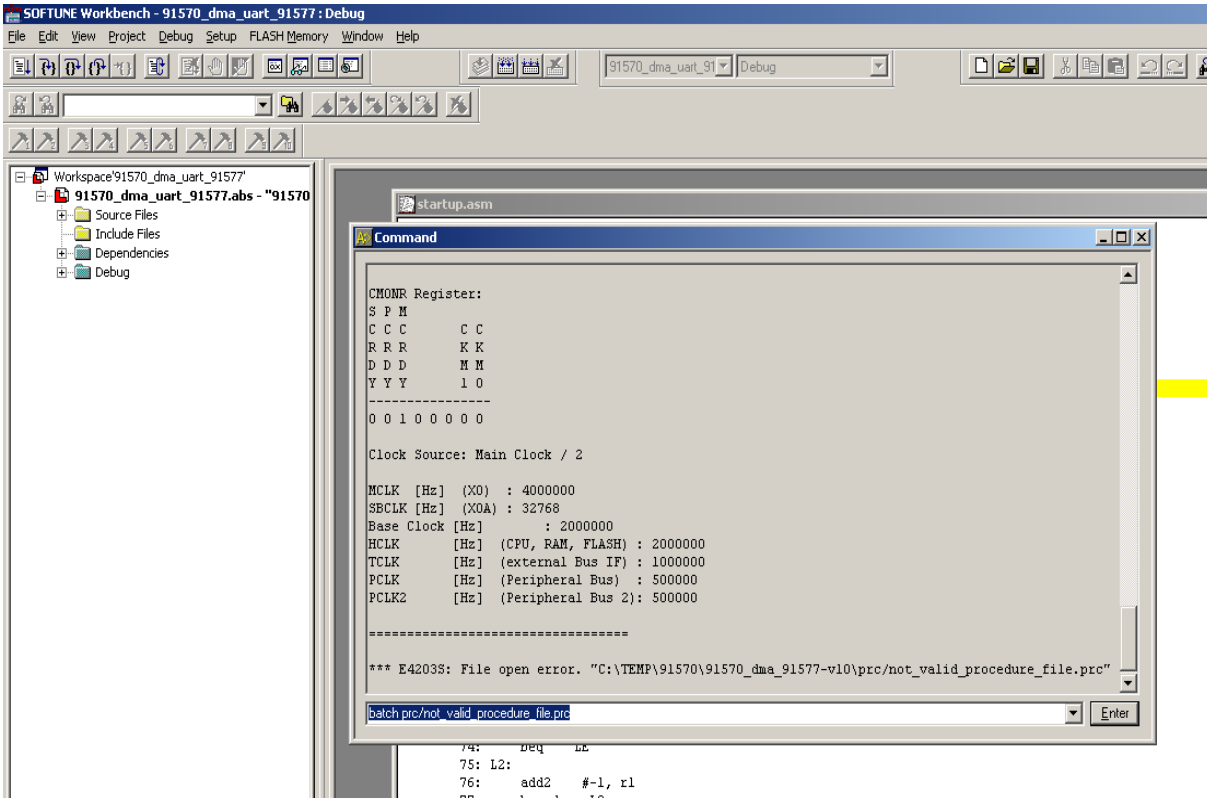
```
batch prc/mb91570_read_clock_settings.prc,D'4000000,D'32768
```

```
75: L2:
76:     add2    #-1, r1
77:     bne:d    L2
78:     stb    r0, @(r13, r1)
```

Batch commands are not case sensitive. Then **`set radix/hexadecimal`** and **`SET RADIX/HEXADECIMAL`** will get the same results.

If a file is not found or a command is not recognized, the command window will display an error message like following (see the picture below):

Figure 5-20. Usage of Batch File in Command Window (2)


SOFTUNE Workbench - 91570_dma_uart_91577 : Debug
File Edit View Project Debug Setup FLASH Memory Window Help
91570_dma_uart_91
Debug
Workspace'91570_dma_uart_91577'
91570_dma_uart_91577.abs - "91570
Source Files
Include Files
Dependencies
Debug
startup.asm
Command
Enter

```
CMONR Register:
S P M
C C C       C C
R R R       K K
D D D       M M
Y Y Y       1 0
----------------
0 0 1 0 0 0 0 0

Clock Source: Main Clock / 2

MCLK  [Hz]  (X0)  : 4000000
SBCLK [Hz]  (X0A) : 32768
Base Clock [Hz]        : 2000000
HCLK       [Hz]  (CPU, RAM, FLASH) : 2000000
TCLK       [Hz]  (external Bus IF) : 1000000
PCLK       [Hz]  (Peripheral Bus)  : 500000
PCLK2      [Hz]  (Peripheral Bus 2): 500000

==================================

*** E4203S: File open error. "C:\TEMP\91570\91570_dma_91577-v10\prc/not_valid_procedure_file.prc"
```

```
batch prc/not_valid_procedure_file.prc
```

```
75: L2:
76:     add2    #-1, r1
```

Further information might be found in the **FR Family Softune Workbench Command Reference Manual** (FscmdE.pdf) document which can be downloaded from the FSEU website.

5.5 External Text Editor Notepad++

If as alternate text editor Notepad++ (NPP) is applied, then several settings are be recommended.

Following NPP plugins are recommended for comfortable software development:

- Plugin Function list: All functions within a selected source file are listed within additional window
- Plugin Open CTags: Jump to function, variable definitions and #Defines by pressing “Alt+space” key and pressing one more time “space” bar for jumping to definition.
- Plugin QuickText: Automatic completion of brackets by pressing “Tab” key after entering keyword. E.g.“while”.is entered and “tab” key is pressed , then all brackets for while-loop are added.
- Replacement of tab by spaces: In [Settings] > [Options ...] > tab [Language Menu/Tabs] select replacement of tab width by ‘4’ spaces.
- Auto completion: Click on [Settings] > [Options ...] > tab [Backup/Auto Completion] and enable [Auto Completion] feature.

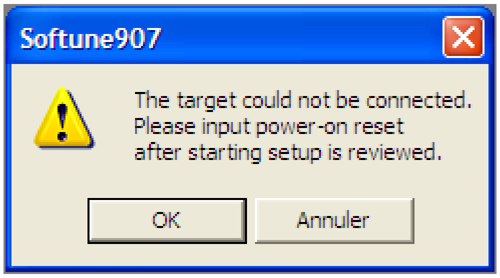
6. SOFTUNE Trouble Shooting

The chapter describes trouble shooting for different scenarios with SOFTUNE.

1. Prompt: “The target could not be connected. Please input power-on reset after starting setup is reviewed”

- Check, if external watchdog causes reset in MCU.

Figure 6-1. Trouble Shooting (1)



Softune907

The target could not be connected.
Please input power-on reset
after starting setup is reviewed.

2. When debug session is started, SOFTUNE crashes and application is closed

- Was the USB driver for MB2100-01-E updated (e.g. required for Win7)? If not, please download from following link and follow the driver installation guide:

  http://www.cypress.com/16lx

- If USB driver update installed and either SOFTUNE was reinstalled or a new version was installed, please do following steps:

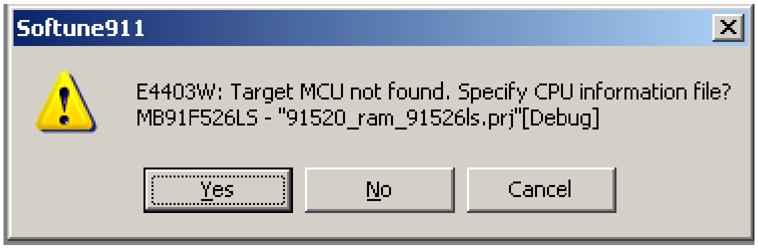
  □ Replace “SiUSB.dll” manually from the new USB driver update into the “Softune\bin” folder

  □ Enter folder “c:\windows\SysWOW64” and rename original “wdapixxxx.dll” (e.g. wdapi1021.dll) to “wdapixxxx_old.dll”. Then copy "wdapixxxx_32.dll" from unzipped “SiUSBdrvr_xxxx\bin” folder into “c:\windows\SysWOW64” folder and rename this file to the original filename “wdapixxxx.dll” (in this case: “wdapi1021.dll”).

- Do you have administrator rights on your user account? If not, then set user account to administrator.
- Is the User Account Control (UAC) feature (min. Win7) enabled? Please disable UAC (see 3.1.1).
- If a new SOFTUNE
- Final step: Reopen SOFTUNE and start debug session.

3. Prompt: “E4403W: Target MCU not found. Specify CPU information file?...”

Figure 6-2. Trouble Shooting (2)



Softune911

E4403W: Target MCU not found. Specify CPU information file?
MB91F526LS - "91520_ram_91526ls.prj"[Debug]

- Did you update the SOFTUNE “.csv” and “FLASH” files from web page link

http://www.cypress.com/16lx

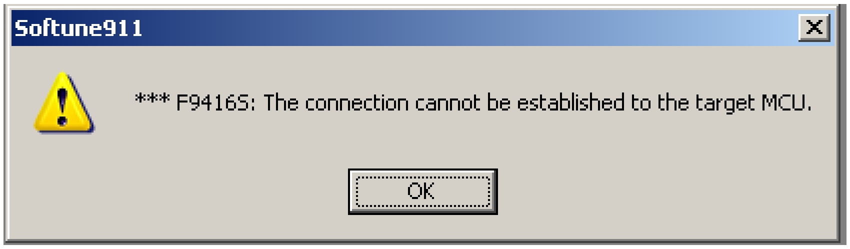
Please follow guidelines in “Readme.txt” of the csv-update.

4. In debug session application is running and when application runs without debugger, a reset is caused periodically.

- Check, if internal watchdog is handled resp. disabled (please check hardware manual of the corresponding MCU series).

5. Prompt: “The connection cannot be established to the target MCU.”

Figure 6-3. Trouble Shooting (3)



Softune911

*** F9416S: The connection cannot be established to the target MCU.

- Power off hardware system and debugger and power on again.
- Any ROM area might be not defined correctly. Please use Cypress Flash Programmer tool, if MCU can be programmed via UART interface. Switch MCU mode pins into “serial write mode” (see Table 6-1)Apply .mhx file from the MCU specific template project and start Flash programming by pressing button [Full Operation (D+E+B+P)] (see chapter 7.). Starting debug session with the template project resp. the original project should be possible again. In this case check resp. change ROM area settings.
- Check, if mode pins are in “user mode” (see Table 6-1). Change settings, if incorrect and make a Power-On reset on target hardware.

Table 6-1. Mode Pin Settings

Operation mode	MD1	MD0	P006
User mode	0	1	-
Serial writer mode	1	0	1

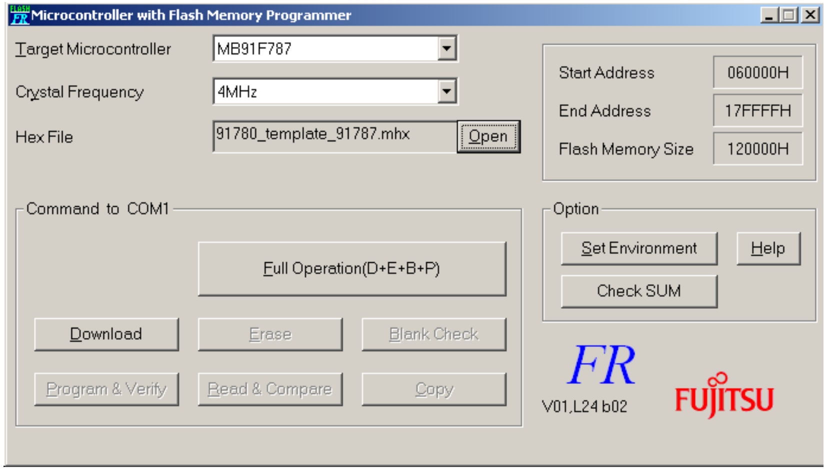
7. Flash Programming of MCU

For programming the MCU via serial interface following steps are listed:

Note9: To flash the MCU via serial interface may some jumper settings are necessary, therefore refer to specific user guide of the SarterKit.

- Install the newest FR Flash Programmer.
- Connect the UART0 of the board via UART'A' or UART'B' connector to the PC.
- Start the FR Flash Programmer.
- Select the Target, Frequency and the (HEX)-File (.mhx) as the following picture shows.

Figure 7-1. FR Flash Programmer



Microcontroller with Flash Memory Programmer

Target Microcontroller: MB91F787

Crystal Frequency: 4MHz

Hex File: 91780_template_91787.mhx

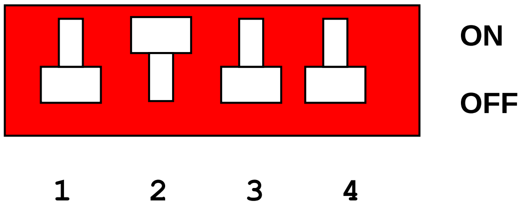
Start Address: 060000H

End Address: 17FFFFH

Flash Memory Size: 120000H

- The next step is to switch into programming mode. For the programming mode the MD0 pin has to be on low level and the MD1 pin on high level. In default all switches of dip switch from StarterKit are on OFF-setting. Put the switch 2 to the ON-setting to get the programming mode as the following picture shows.

Figure 7-2. Dip Switch for Programming Modes



- Press the “Full Operation” button to start programming, press the RESET button and click on ‘OK’.

  CAUTION: DO NOT INTERRUPT OR CUT OFF POWER DURING ERASE!!!

- When the download was not successful (message on PC) check the jumper settings of UART and the cable connections.
- After a successfully downloaded program, put all switches of dip switch back to the OFF-position and push the RESET button.

The application is flashed in the MCU and starts after the reset.

Revision History

Document Revision History

Document Title: FR Family FR81S Series, Emulation System Document Number: 002-04887			
Revision	**Issue Date**	**Origin of Change**	**Description of Change**
**	03/28/2011	MKEA	Initial Release
	06/09/2011		Changed contact mailing address
	04/20/2012		Chapter “high speed mode” added
	04/23/2012		Chapter “Automatic Flash Synchronisation” added
	11/01/2012		Chapter “linker settings” added, Win7 notes added installation chapter, new chapters added.
	11/29/2012		Update of “installation” chapters, new: “chapter “trouble shooting”
*A	12/28/2015	MKEA	Migrated Spansion document from MCU-AN-381000-E-V15 to Cypress format