Please note that Cypress is an Infineon Technologies Company.
The document following this cover page is marked as “Cypress” document as this is the company that originally developed the product. Please note that Infineon will continue to offer the product to new and existing customers as part of the Infineon product portfolio.

Continuity of document content
The fact that Infineon offers the following product as part of the Infineon product portfolio does not lead to any changes to this document. Future revisions will occur when appropriate, and any changes will be set out on the document history page.

Continuity of ordering part numbers
Infineon continues to support existing part numbers. Please continue to use the ordering part numbers listed in the datasheet for ordering.



THIS SPEC IS OBSOLETE

Spec No: 002-05235

Spec Title: AN205235 - FM3 Microcontroller MB9B500 Series, FSS MB9BF506R EV-BOARD

Replaced by: None

AN205235

FM3 Microcontroller MB9B500 Series, FSS MB9BF506R EV-BOARD

This application note describes an economical and simple means for study usage for MB9B506 series MCU. The board compatible with both 3.3 and 5V system contains abundant external resources (LCD, buttons, UART, Nand Flash, CAN…) to demonstrate MCU periphery function.

OBSOLETE

Contents

1 Introduction....1
1.1 Product Overview....1
1.2 MB9B506 Series MCU....2
1.3 Board Features....3
2 Hardware Setting....4
2.1 Main Features....4
2.2 Jumpers Overview....5
2.3 Setting for UART On-Board Programming....5
2.4 Setting for USB On-Board Programming....6
2.5 Setting for Debug Tool....6
3 Flash On-Board Programming....7
3.1 On-Board Programming via UART....7
3.2 On-Board Programming via USB....8
4 Sample Code....9
4.1 UART....10
4.2 Nand Flash....12
4.3 RTC....12
4.4 CAN....13
4.5 USB Function....14
4.6 USB Host....15
4.7 LCD & AD & Key....16
5 Debug Tool and IDE....17
5.1 Debug with J-Link in IAR EWARM Workbench....18
5.2 Debug with U-Link ME in Keil uVision4....24
6 Materials Download....30
A Appendix....31
A.1 Links to Microcontrollers....31
A.2 Links to third party sites....31
Document History....32

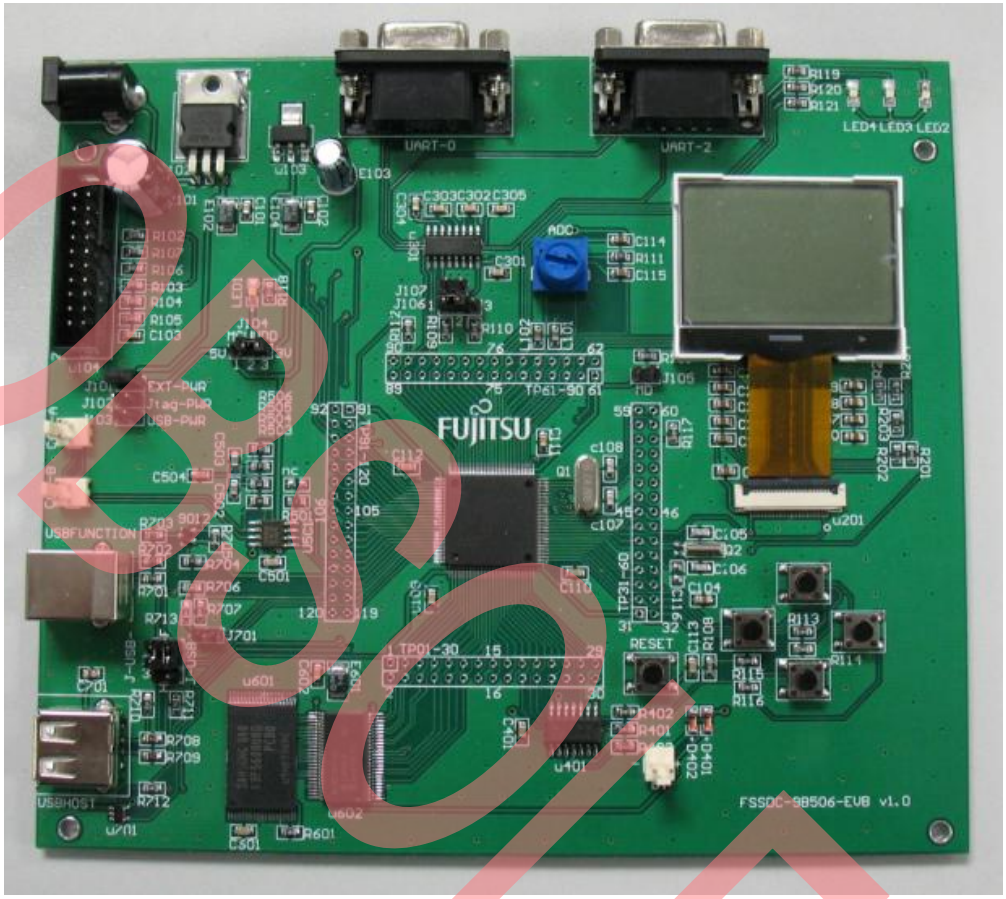
1 Introduction

1.1 Product Overview

FSS MB9BF506R EV-Board (PN: FSSDC-9B506-EVB) provides an economical and simple means for study usage for MB9B506 series MCU. The board compatible with both 3.3 and 5V system contains abundant external resources (LCD, buttons, UART, Nand Flash, CAN…) to demonstrate MCU periphery function.

It also provides standard 20 pin JTAG interface, which is both compatible with IAR and Keil debug tool. In addition, it allows On-board programming with both USB and UART mode.

Figure 1. FSSDC-9B506-EVB Overview



1.2 MB9B506 Series MCU

MB9B506 series MCU is 32-bit general purpose MCU of FM3 family that features the industry's leading-edge ARM Cortex-M3TM CPU and integrates Fujitsu's highly reliable and high-speed secure embedded flash technology. This MCU can operate at up to 80MHz CPU frequency and work at a wide voltage range (2.7V-5.5V), which can be both compatible with 3.3V and 5V system.

It includes a host of robust peripheral features, including motor control timers (MFT), base timer (can be configured to PWM, PPG, Reload, PWC timer), ADCs, on-chip memory (up to 512K Flash, up to 64K SRAM) and a wide range of communication interfaces (USB, I2C, SIO, LIN, CAN).

The size of on-chip memory can be configured according to different part number and the package is available in LQFP and BGA, shown in following table.

Table 1. Product List

Product	Flash	SRAM	Package
MB9BF500N/R	256kB	32kB	N: LQFP-100 R: LQFP-120
MB9BF504N/R	256kB	32kB	N: LQFP-100/BGA-112 R: LQFP-120
MB9BF505N/R	384kB	48kB	N: LQFP-100/BGA-112 R: LQFP-120
MB9BF506N/R	512kB	64kB	N: LQFP-100/BGA-112 R: LQFP-120

OBSOLETE

1.3 Board Features

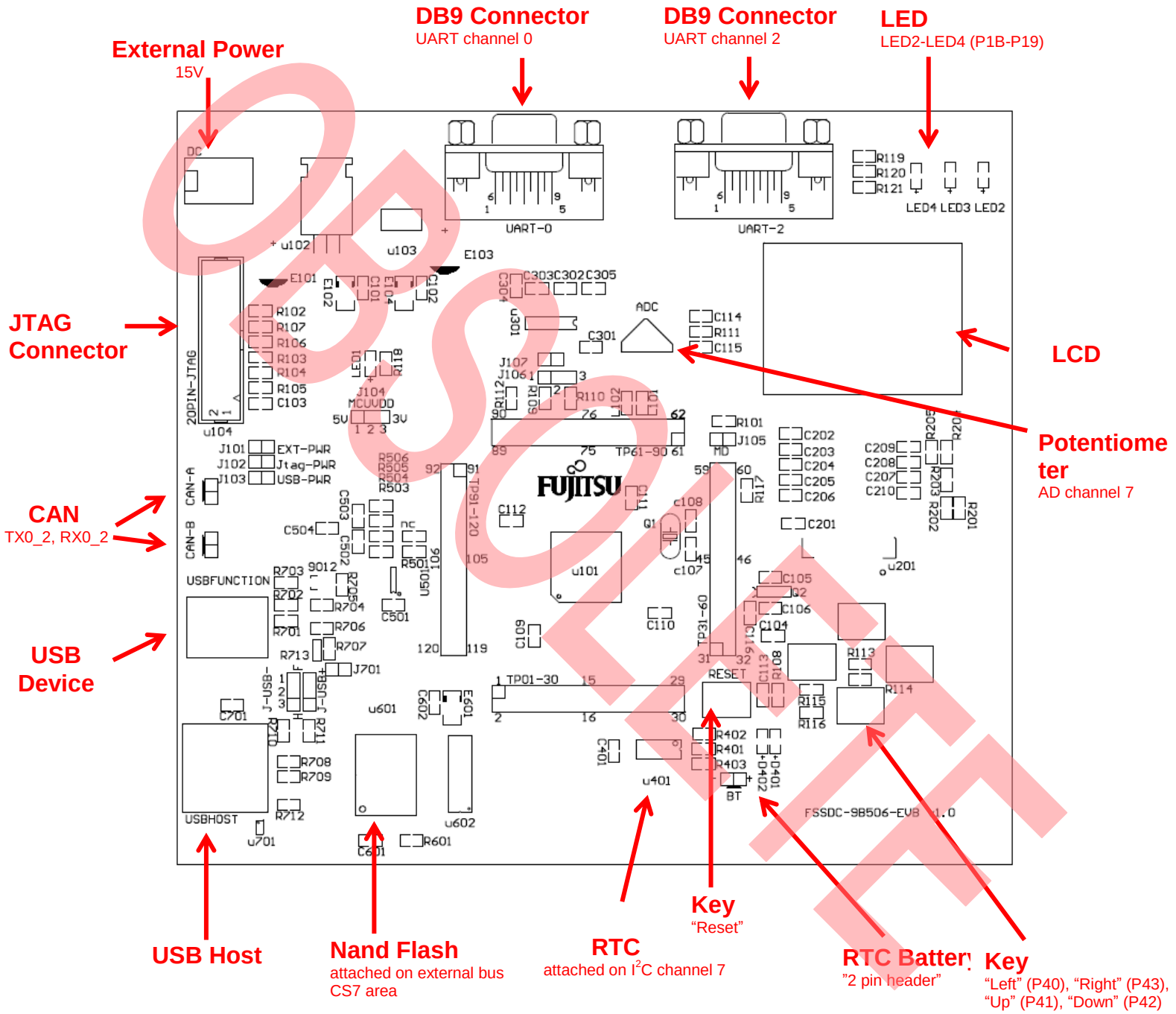
FSS MB9BF506R EV-Board includes following features:

- Microcontroller MB9BF506R
- 2x UART-Transceiver (Female DB9 connector)
- 2x High-speed CAN-Transceiver (2 pin header)
- 1x USB-Host (Type-A connector)
- 1x USB-Device (Type-B connector)
- 32M x 8bit Nand Flash (K9F5608U0D)
- High Stability I^2C-Bus Interface RTC Module (RX-8025T)
- Standard JTAG Interface on a 20 pin-header
- 128 x 64 matrix LCD
- 3x ‘User’ LED
- 4x ‘User’-button
- 1x ‘Reset’-button
- 1x potentiometer regulating input voltage to AD channel
- All 120 pins routed to test pads
- On-board 5V and 3V voltage regulators, ‘Power’-LED
- Power supply via USB-Device, JTAG or external with 15V power connector

OBSOLETE

2 Hardware Setting

2.1 Main Features



OBSOLETE

2.2 Jumpers Overview

Table 2. Jumper List

Jumper Name	Function	Setting
J101-J103[*1]	Power selection	Short J101: select external power Short J102: select JTAG power Short J103: select USB power
J104[*2]	Voltage selection	Short 1,2: 5V Short 2,3: 3.3V
J-USB- J-USB+	USB Host/Function selection	Short 1,2: USB Function Short 2,3: USB Host
J105	Mode setting	Short: Programming mode Open: Normal mode
J106	Oscillator Selection	Short 1,2: when 4M oscillator is used Short 2,3: when 48M oscillator is used
J107	P22 connection	Short: connect P22 (UART_TX0) with MAX3232 Open: disconnect P22 (UART_TX0) with MAX3232
J701	Voltage division	Short: when 5V power is used Open: when 3.3V power is used

Note: Attention:

- Never short J101-103 at the same time! When Keil U-Link ME is used, JTAG power cannot be used.
- When Keil U-link is used, only 3.3V can be selected, never select 5V power!

2.3 Setting for UART On-Board Programming

Table 3. Setting for UART On-Board Programming

Part Number	Function	Setting
UART-0	DB9 connector	Connect with PC via RS232 cable
J106	Oscillator Selection	Open
J105	Mode setting	Short
J107	P22 connection	Short

OBSOLETE

2.4 Setting for USB On-Board Programming

Table 4. Setting for USB On-Board Programming

Part Number	Function	Setting
USBFUNCTION	USB function connector	Connect with PC via USB cable
J-USB- J-USB+	Function/Host selection	Short 1,2
J105	Mode setting	Short
J106	Oscillator Selection	Short 1,2
J107	P22 connection	Open
J701	Voltage division	Short: when 5V power is used Open: when 3.3V power is used

2.5 Setting for Debug Tool

- Use IAR J-Link

Table 5. Setting for J-Link

Part Number	Function	Setting
20PIN-JTAG	JTAG connctor	Connect with J-Link
J101-J103	Power selection	Short J101: select external power Short J102: select JTAG power Short J103: select USB power
J104	Voltage selection	Short 1,2: 5V Short 2,3: 3.3V
J105	Mode setting	Open

- Use Keil U-Link ME

Table 6. Setting for U-Link

Part Number	Function	Setting
20PIN-JTAG	JTAG connctor	Connect with U-Link ME
J101-J103	Power selection	Short J101: select external power Short J103: select USB power
J104	Voltage selection	Short 2,3
J105	Mode setting	Open

OBSOLETE

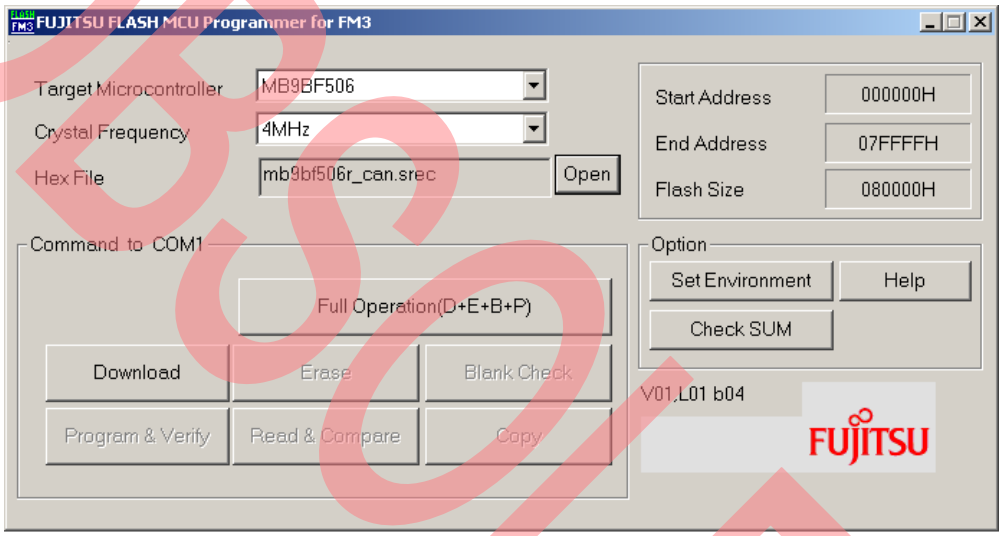
3 Flash On-Board Programming

There are two ways to program the on-chip Flash of MB9BF506 series MCU: UART On-Board Programming and USB On-Board Programming.

3.1 On-Board Programming via UART

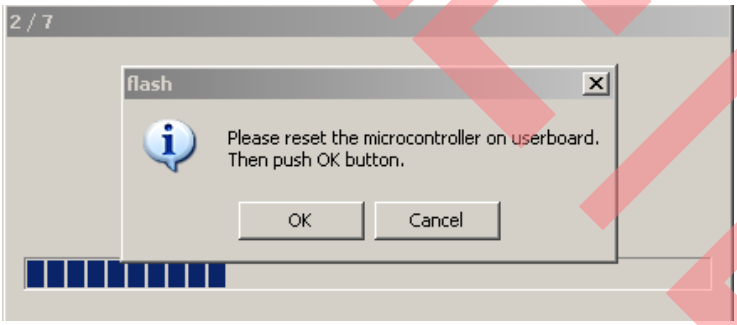
- First check the hardware setting as introduced by section 2.3.
- Install the UART programmer: PCWFM3_V01L01. (It can be downloaded on the web)
- Open it, set the parameter as shown in following figure, and select Hex file.

Figure 2. UART Programmer Overview



- Click Full Operation.

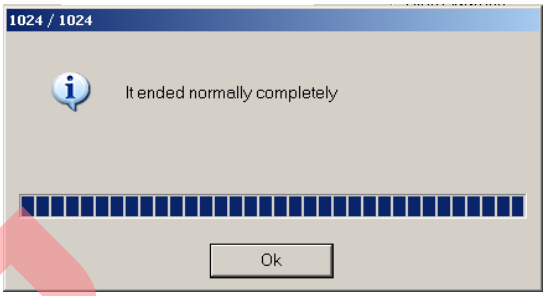
Figure 3. Press Reset Button



OBSOLETE

- Press reset key in the board, and the programming will start.

Figure 4. Start Programming



3.2 On-Board Programming via USB

- First check the hardware setting as introduced by section 2.4.
- After connect with PC via USB cable, the EVB can be identified as a USB device.

Figure 5. USB Device Sign



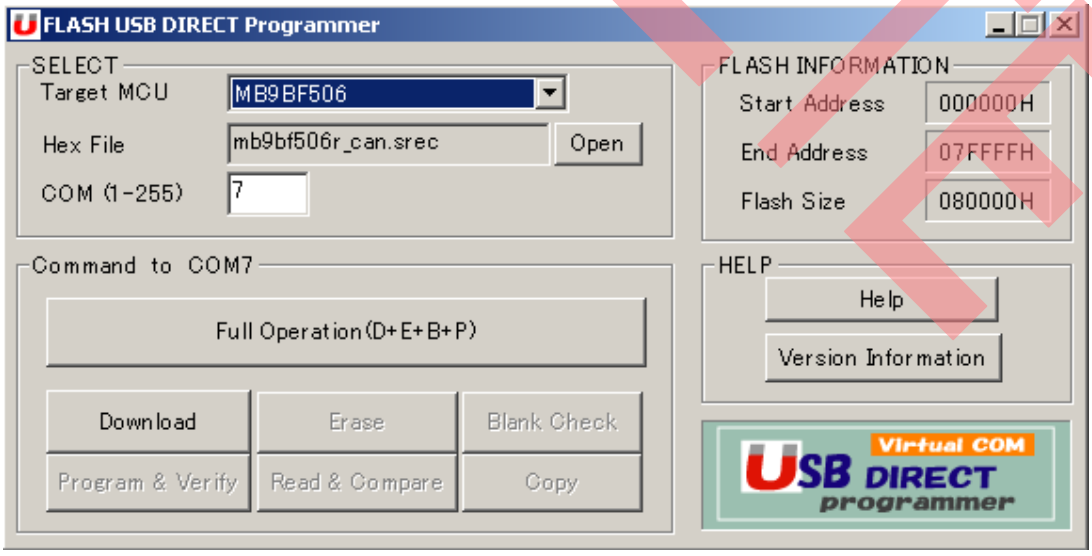
- Check the COM port for this USB port in the device manager.

Figure 6. COM Port in Device Manager



- Install the USB programmer: USBDW_V01L03. (It can be downloaded on the web)
- Open it, set the parameter as shown in following figure, and select Hex file.

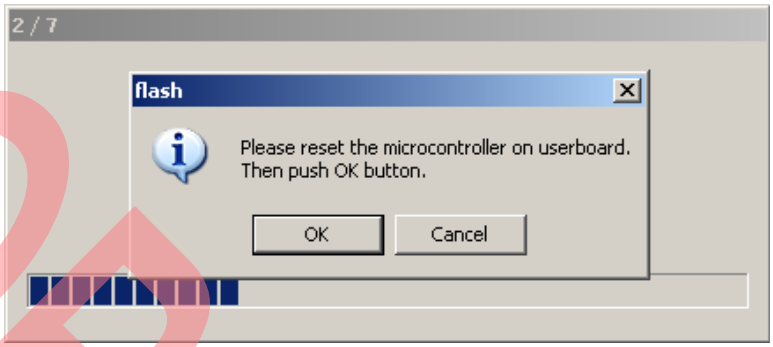
Figure 7. USB Programmer Overview



OBSOLETE

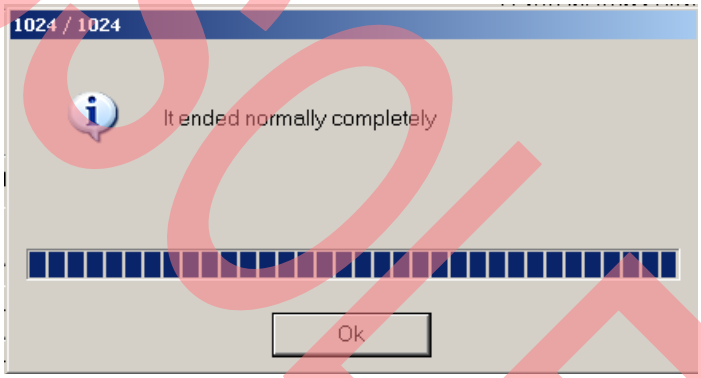
- Click Full Operation.

Figure 8. Press Reset Button



- Press reset key on the board. Programming will start.

Figure 9. Start Programming



4 Sample Code

The sample codes for FSSDC-9B596-EV board are listed as following table.

Table 7. Sample Code

Project Name	Description
ext_bus_nandflash	Use external bus to access Nand Flash
Uart	Use UART to communicate with PC
Can	Implement CAN communication between 2 CAN nodes
Rtc	Use IIC to access RTC module
usb_function_mose	Implement the USB mouse to demonstrate USB function
usb_host_catch_mouse	Identify a USB mouse and print the mouse position on debug window
lcd_ad_key	Display ad value and key status on LCD

OBSOLETE

Note:

1. It provides both IAR and Keil project for these sample code, IAR project is developed in EWARM Embedded Workbench V6102, and Keil project is developed in Keil uVision 4.20.
2. If user use other version to open these projects, compiling error may occur, in this case, please check following setting.

- IAR IDE
  - MCU type
  - Pre-included file
  - ICF file
  - Flash loader
- Keil IDE
  - MCU type
  - Pre-included file
  - ROM & RAM memory address

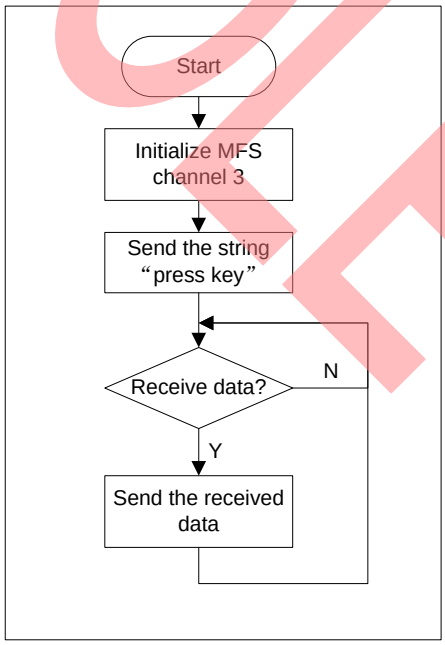
4.1 UART

- **Hardware Setting**

Connect UART-2 with PC via RS232 cable

- **Flowchart**

Figure 10. UART Sample Code Flowchart

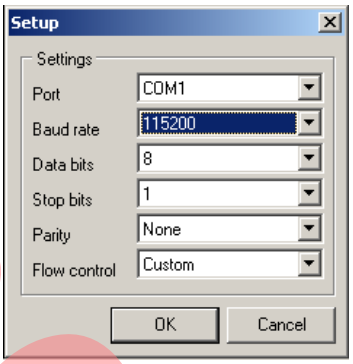


OBSOLETE

■ **Usage**

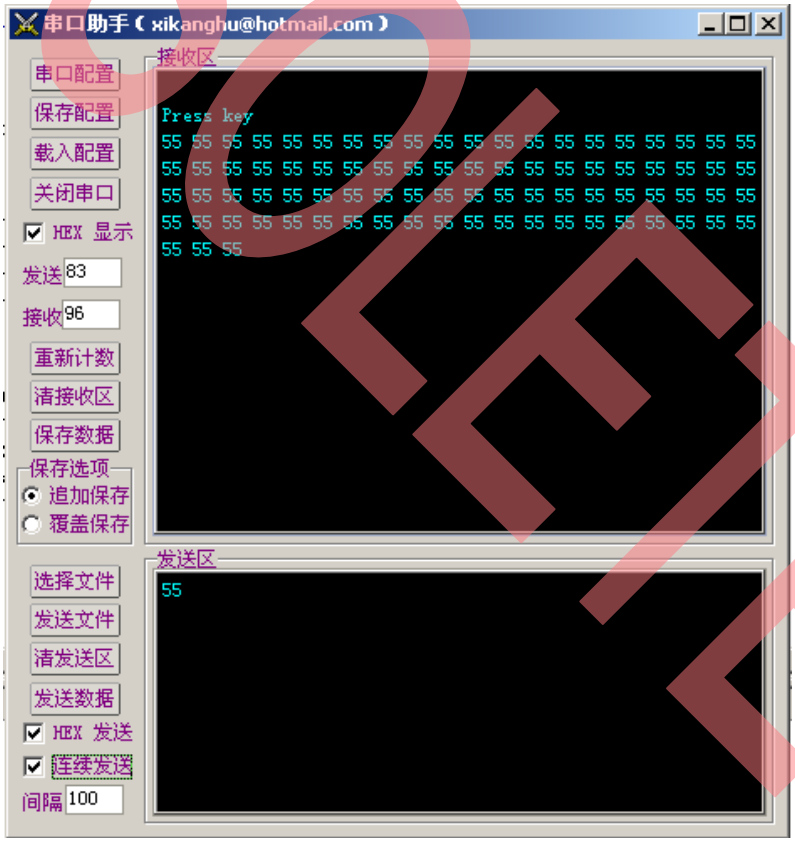
1. Open a COM assistant software, make following setting, and open COM port

Figure 11. COM Assistant Setting



2. Open UART sample code and run it, user can watch a string on the COM assistant window. Then send character to EV-board, the character can be received.

Figure 12. Send UART Data From PC Side

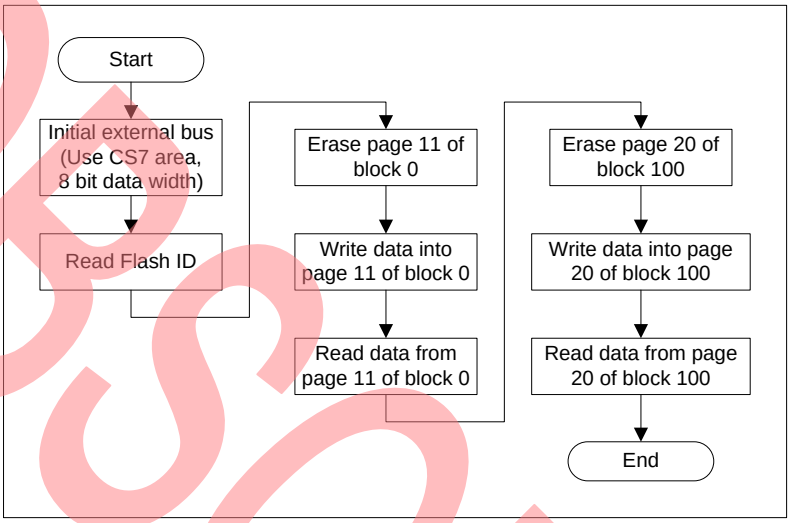


OBSOLETE

4.2 Nand Flash

- **Hardware Setting**
  None
- **Flowchart**

Figure 13. Nand Flash Sample Code Flowchart



- Usage

1. Open Nand Flash sample project, and start debug.
2. Use “step over” to debug, and watch local variable “flash_id”, “buf”.

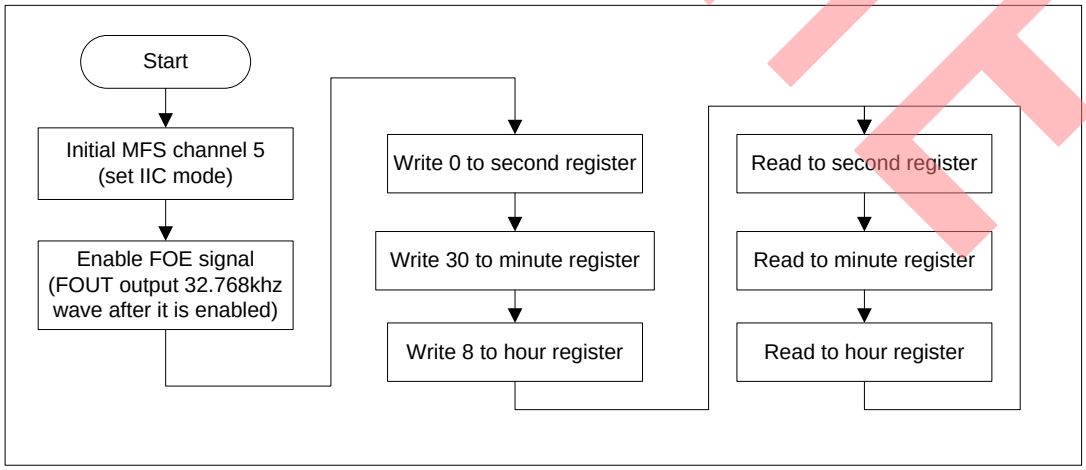
4.3 RTC

- Hardware Setting

It is an option that user can user external battery to supply power for RTC module (RX8025-T) so that the date and time can be stored after system power shut down.

- **Flowchart**

Figure 14. RTC Sample Code Flowchart



OBSOLETE

- Usage

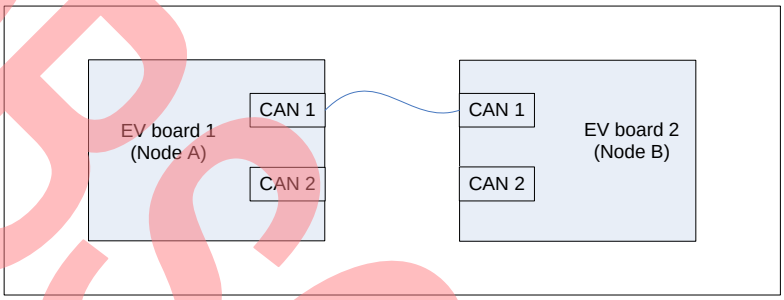
1. Open Nand Flash sample project, and start debug.
2. Use “step over” to debug, and watch local variable “sec”, “min”, “hour”.

4.4 CAN

- Hardware Setting

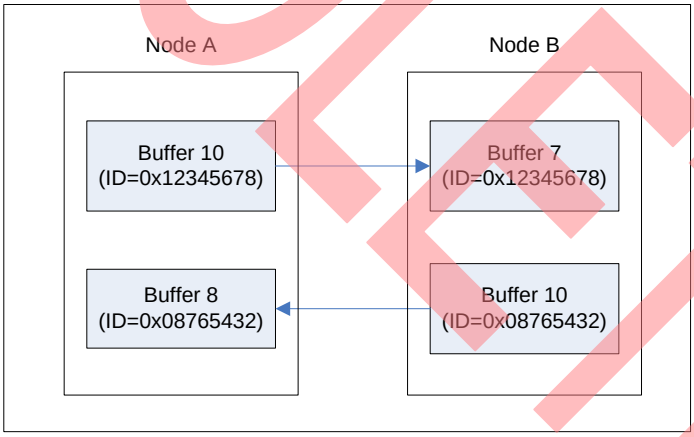
Connect 2 EV-board with CAN interface, as following figure.

Figure 15. CAN Hardware Connection



- Block Diagram
  - Node A: use buffer 10 (ID = 0x12345678) to transfer and buffer 8 (ID = 0x08765432) to receive
  - Node B: use buffer 7 (ID = 0x12345678) to receive and buffer 10 (ID = 0x08765432) to transfer.

Figure 16. CAN Demo Block Diagram



- Usage

1. Define “CAN_TX_RX” in the code, and compile, then program the code into Node A
2. Don’t define “CAN_TX_RX” in the code, and compile, then program the code into Node B.
3. Connect Node A with Node B

When in real usage, user can program one node, and use the other node to debug.

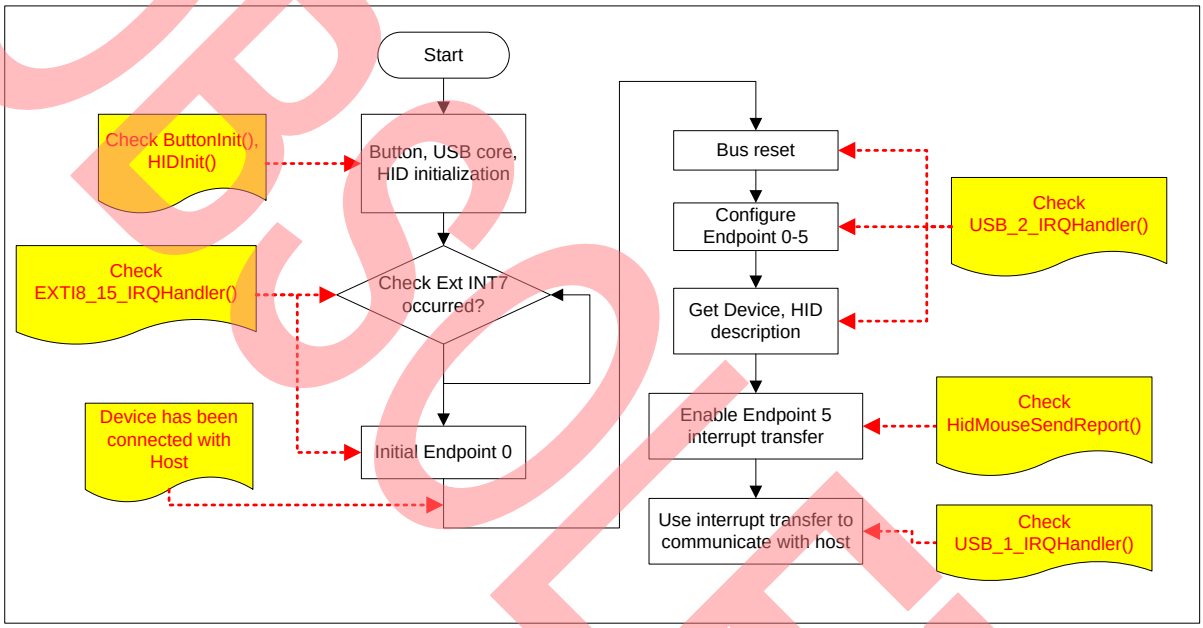
OBSOLETE

4.5 USB Function

- Hardware Setting
  - Check if 1,2 of J-USB- and J-USB+ short
  - Check J701 (Short: 5V, Open: 3.3V)
  - Connect with PC via USB cable
- Flowchart

The following flowchart illuminates the procedure to implement a USB mouse, it is not a certain flow of a function, but provides a clue to study and understand the sample code.

Figure 17. USB Function Sample Code Flowchart



- Usage

1. Make the hardware setting.
2. Watch the mouse moving when pressing “Up”, “Down”, “Left”, “Right” buttons.

OBSOLETE

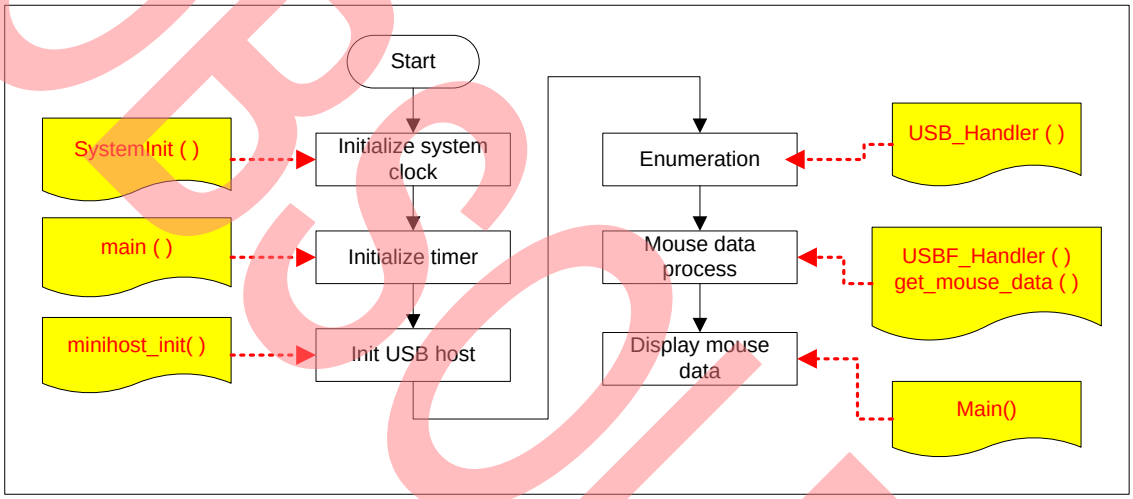
4.6 USB Host

- Hardware Setting
  - Check if 2,3 of J-USB- and J-USB+ short
  - Connect with a USB mouse

- Flowchart

The following flowchart illuminates the procedure to implement USB host function, which can catch the position of a USB mouse, it is not a certain flow of a function, but provides a clue to study and understand the sample code.

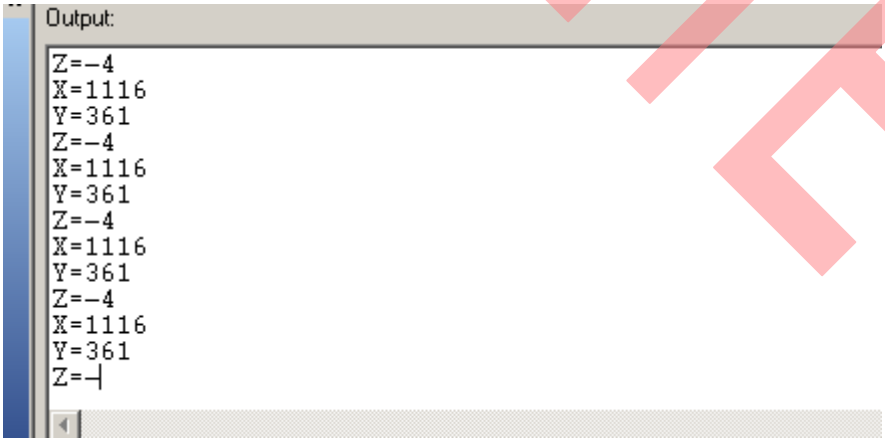
Figure 18. USB Host Sample Code Flowchart



- Usage

1. Make the hardware setting
2. Enable definition “Debug” in the code
3. Run the code.
4. Move the USB mouse, and watch the mouse position in the terminal IO window as following figure. (Debug viewer window in Keil IDE)

Figure 19. USB Mouse Position Display



OBSOLETE

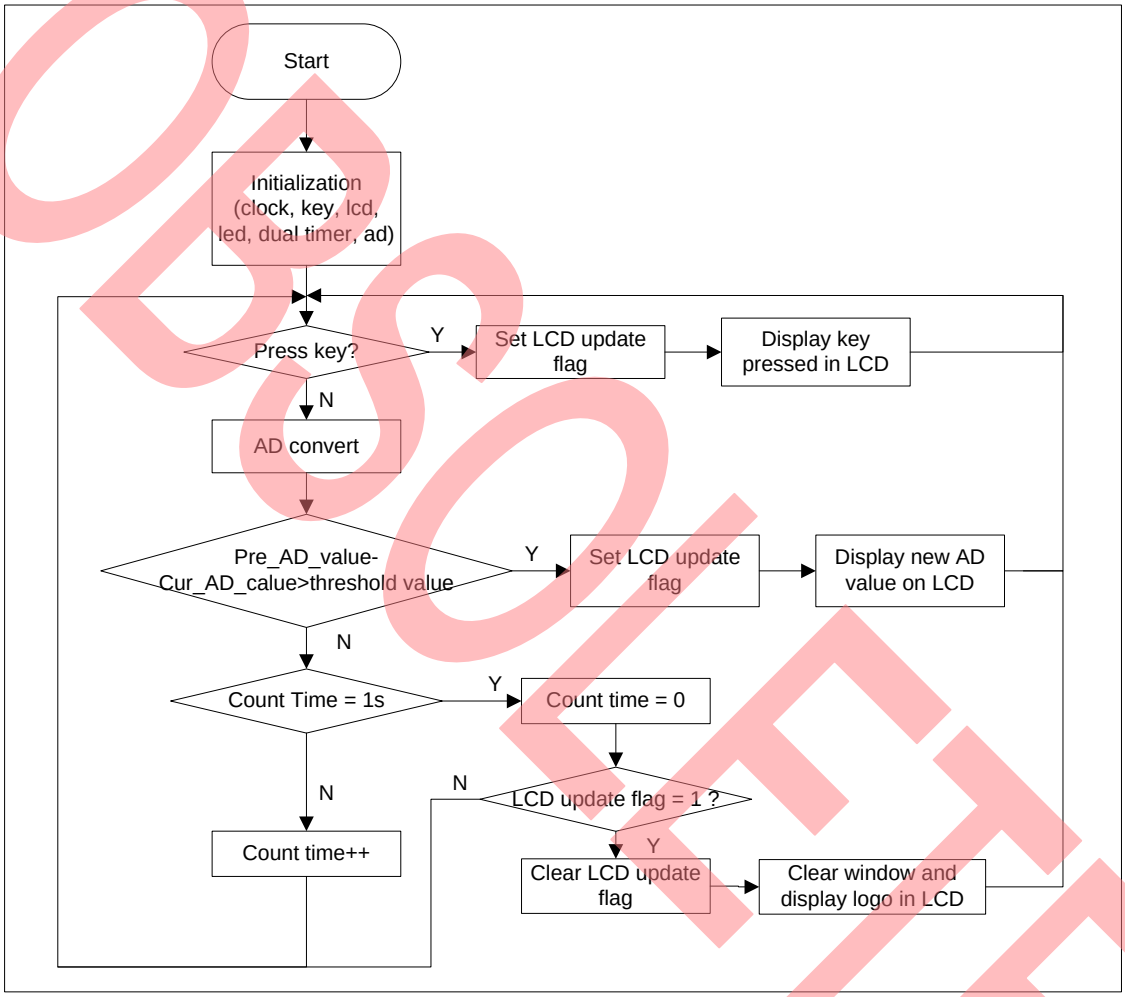
4.7 LCD & AD & Key

- Hardware Setting

None

- Flowchart

Figure 20. : LCD, key, AD Sample Code Flowchart



- Usage

1. Run the code
2. Press key, and the key pressed can display on the LCD
3. Regulate the potentiometer to change the input voltage of channel 7, and this voltage can be detected and displayed on LCD.



OBSOLETE

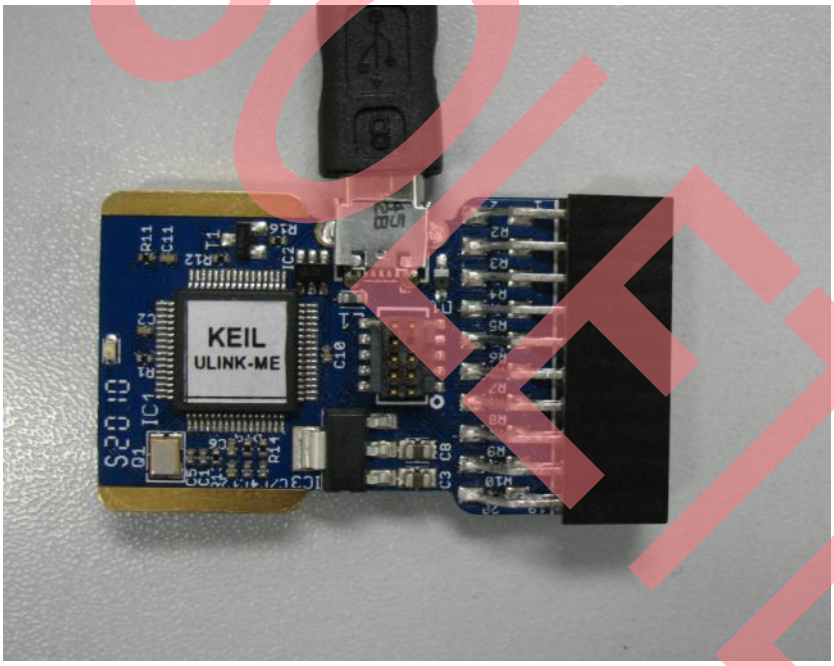
5 Debug Tool and IDE

FSS MB9BF506R EV-Board supports both Keil U-Link-ME and IAR J-Link for debug shown as following.

Figure 21. J-Link Overview



Figure 22. U-Link Overview



The U-Link-me should be used with Keil uVision 4 which can be downloaded freely from following web.

https://www.keil.com/update/sw/RVMDK/4.20

The J-Link should be used with IAR Embedded Workbench which can be downloaded freely from following web.

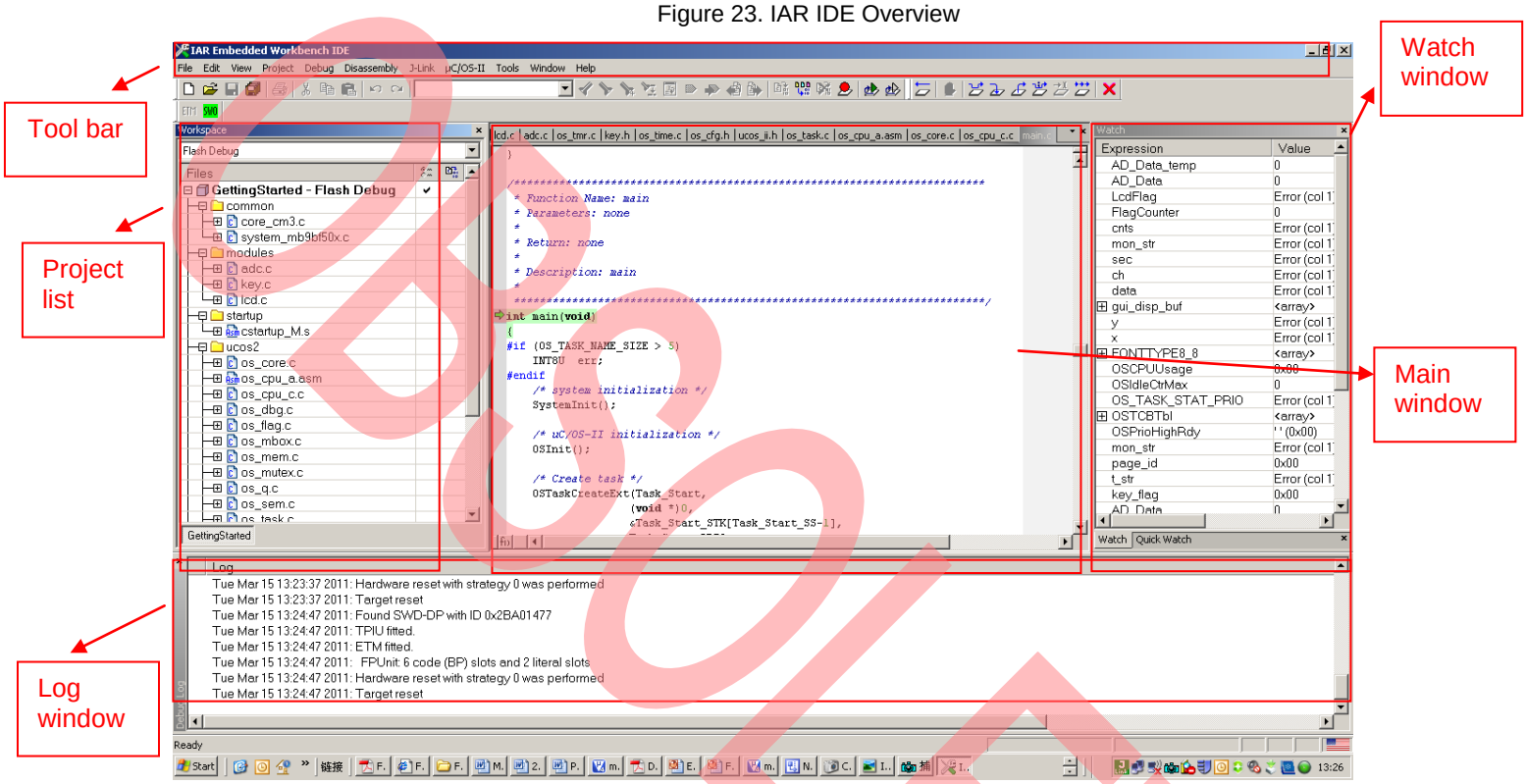
http://www.iar.com/website1/1.0.1.0/68/1/

OBSOLETE

5.1 Debug with J-Link in IAR EWARM Workbench

The sample code can be debugged in IAR EWARM Workbench with J-Link. The following figure shows basic debug window.

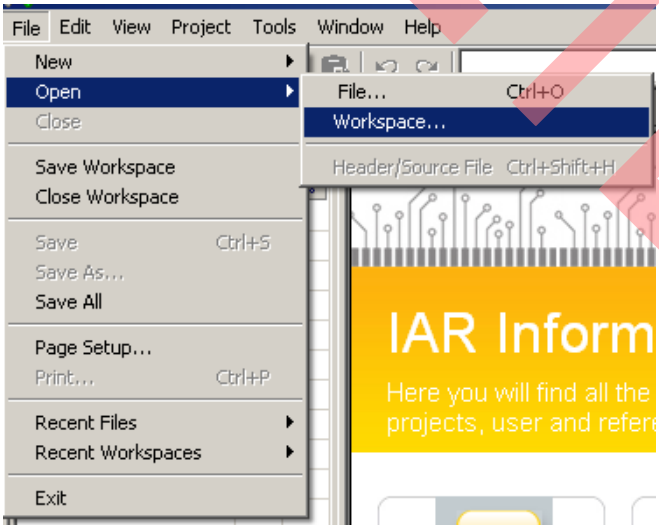
Figure 23. IAR IDE Overview



- Run an Existed Project

1. Open a project by clicking “File | Open | Workspace”

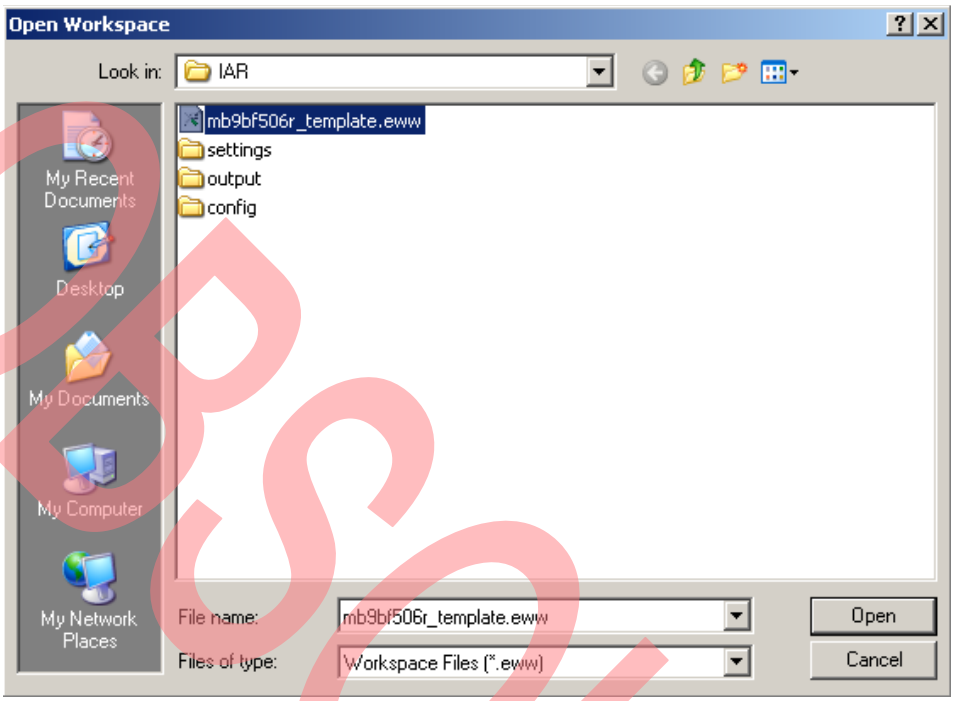
Figure 24. Open a Project



OBSOLETE

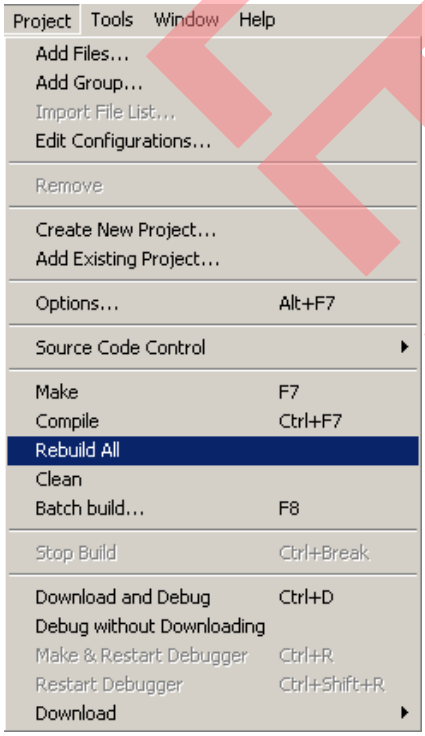
2. Select a project (eww file)

Figure 25. Select a Project



3. Click “Project | Rebuild All”

Figure 26. Rebuild All



OBSOLETE

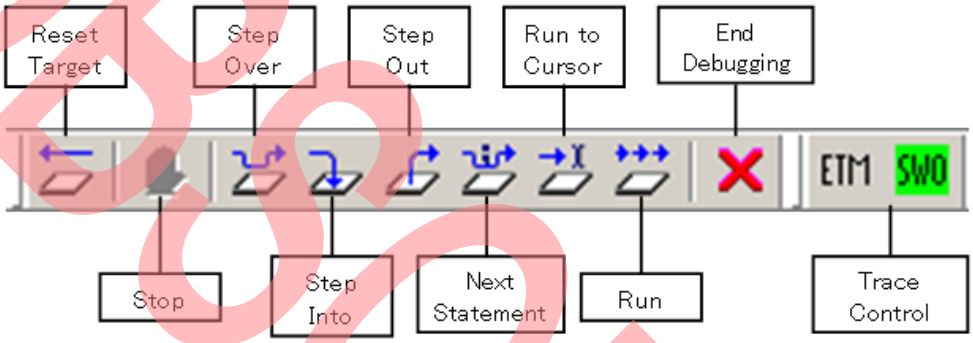
4. Click “Download and Debug”

Figure 27. Click Download and Debug



5. Use following tool bar to debug

Figure 28. Debug Tool Bar

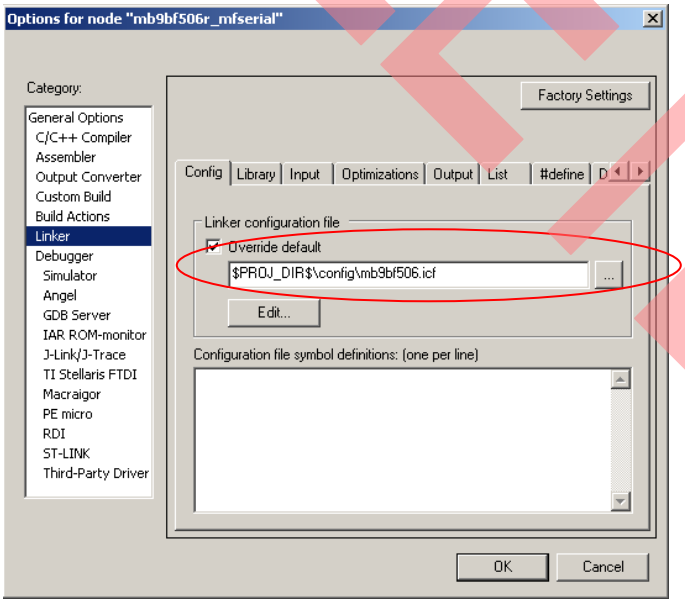


The sample codes support both Flash and RAM debug in IAR EWARM Workbench, if Flash debug is used, the code is programmed into MB9BF506 Flash. If RAM debug is used, the code only runs in RAM area, and after power off, the code will not be stored, but the RAM debug will be faster than Flash debug.

- Setting for Flash Debug

1. Check the configuration file path ($PROJ_DIR$\config\mb9bf506.icf) in Linker table.

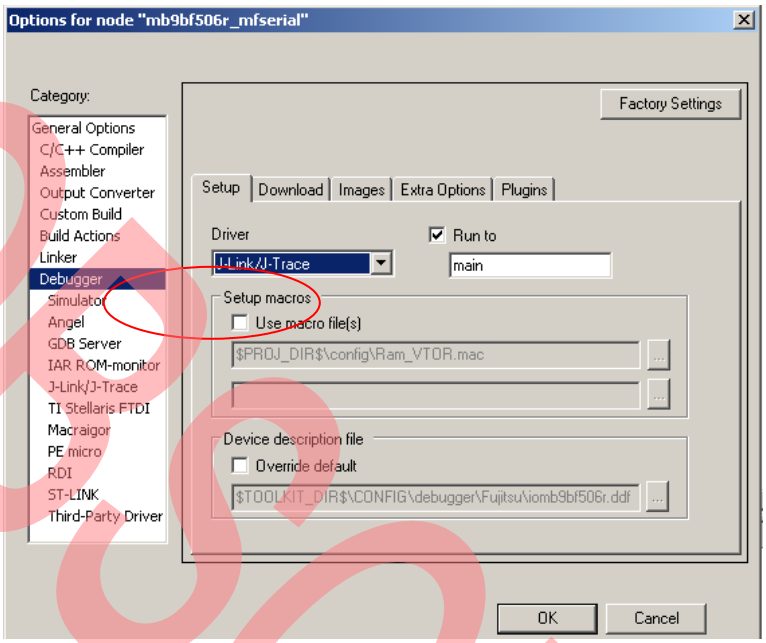
Figure 29. ICF File for Flash Debug



OBSOLETE

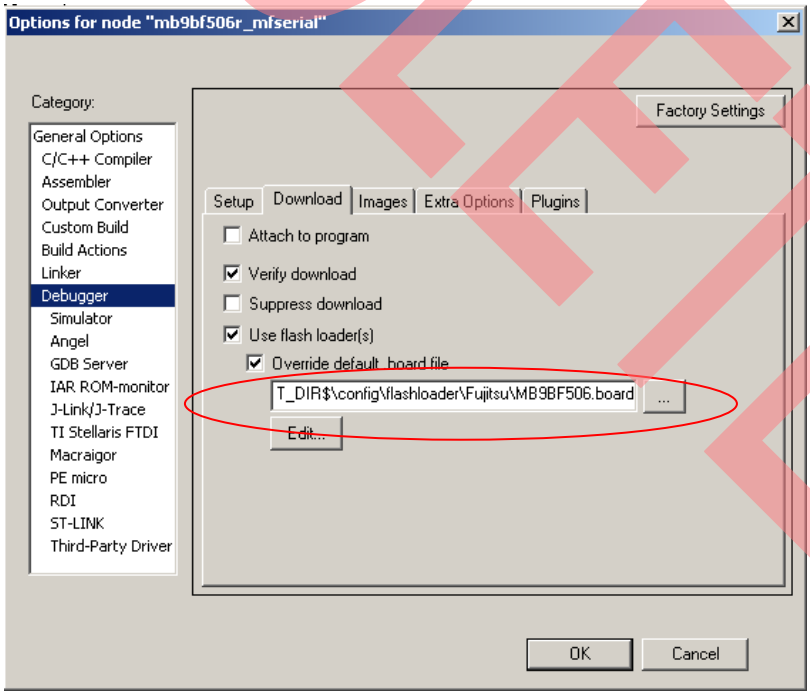
2. Don’t select “Use macro files” in “Debugger|Setup” table.

Figure 30. Macro File Disabled



3. Set Flash loader file path ($TOOLKIT_DIR$\config\flashloader\Fujitsu\MB9BF506.board) in “Debug|Download” table.

Figure 31. Flash Load File Path

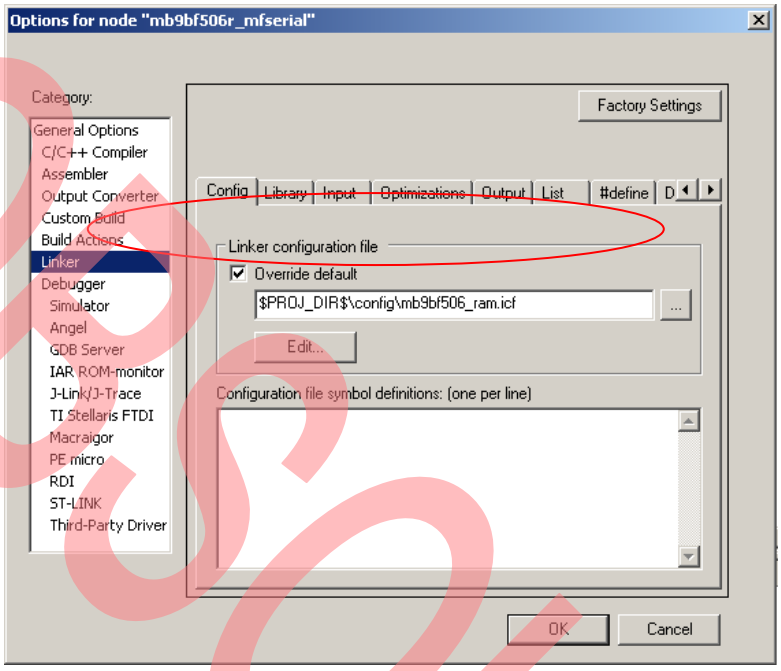


OBSOLETE

- Setting for RAM Debug

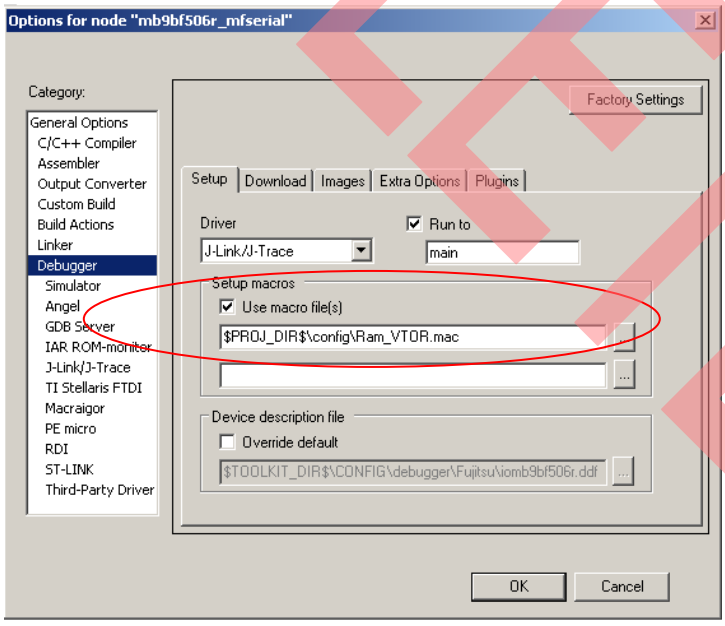
1. Check the configuration file path ($PROJ_DIR$\config\mb9bf506_ram.icf) in Linker table.

Figure 32. ICF File for RAM Debug



2. Select “Use macro files” in “Debugger|Setup” table.

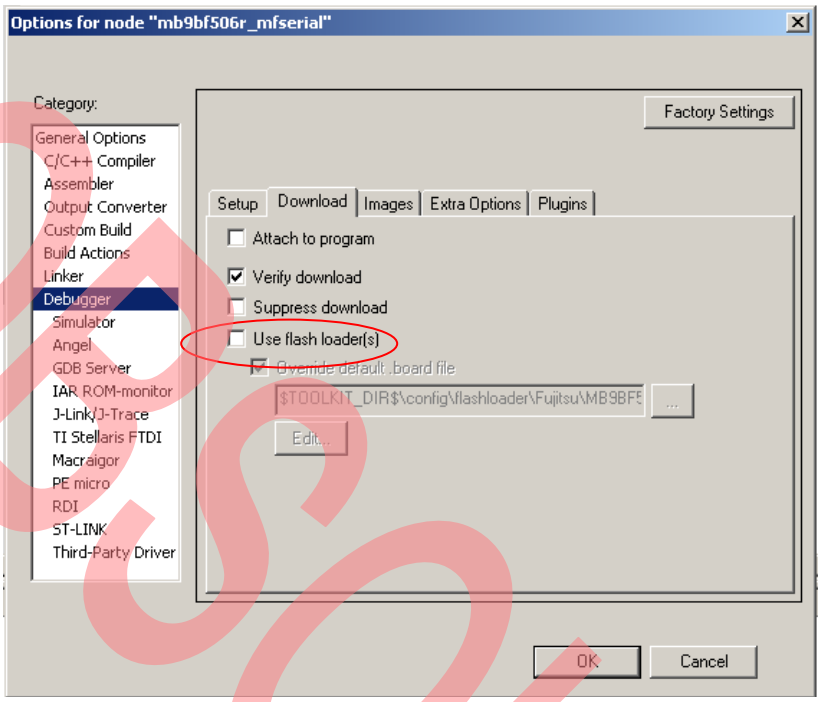
Figure 33. Macro File Enabled



OBSOLETE

3. Don’t use Flash loader file.

Figure 34. Flash Loader File Disabled

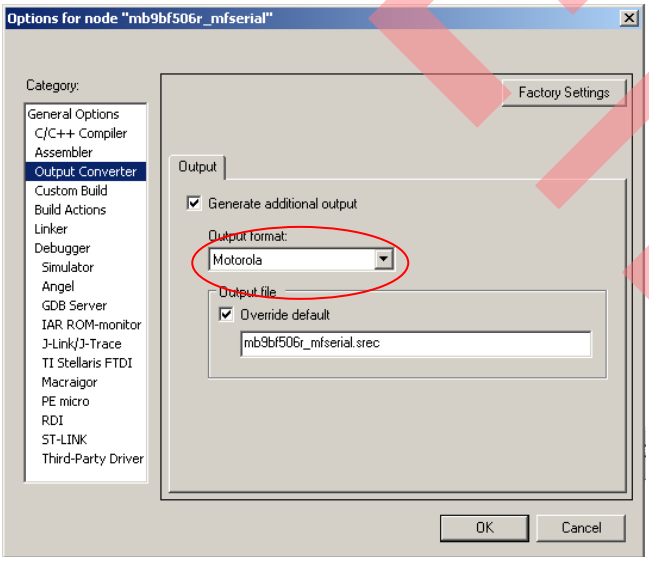


If user need to program the hex file into Flash via UART or USB programmer, a hex file need to be produced first.

- How to Make a HEX File

1. Use Flash debug
2. Select “Generate additional output” in “Output Converter” table.

Figure 35. Hex File Genration in IAR IDE



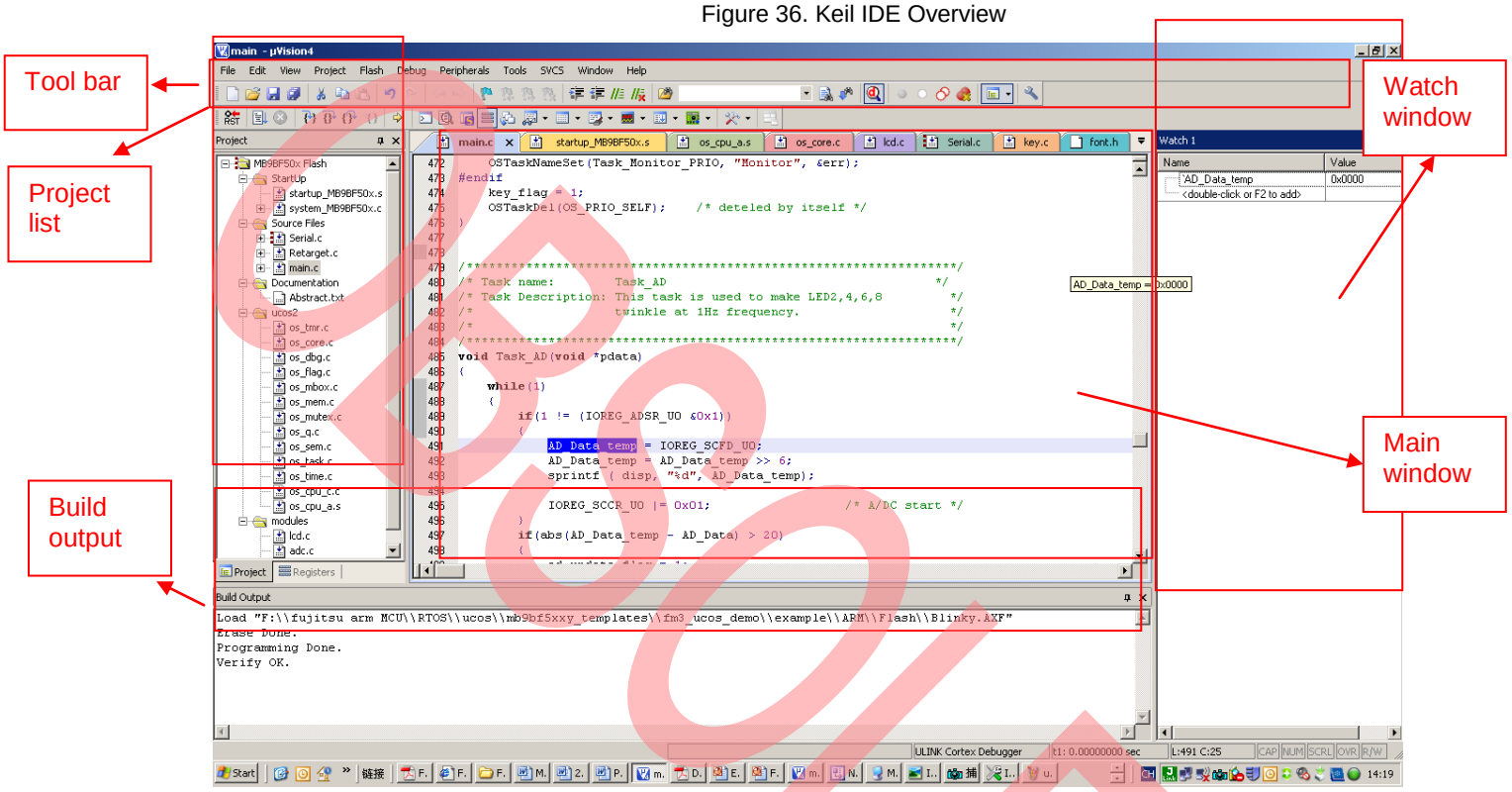
3. User can find the generated file in path (../Debug/Exe)

OBSOLETE

5.2 Debug with U-Link ME in Keil uVision4

The sample code can also be debugged in Keil µVersion4 with U-Link. The following figure shows basic debug window.

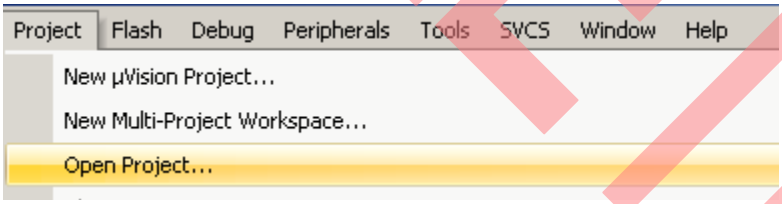
Figure 36. Keil IDE Overview



- Run an Existed Project

1. Open a project by clicking “Project | Open Project…”

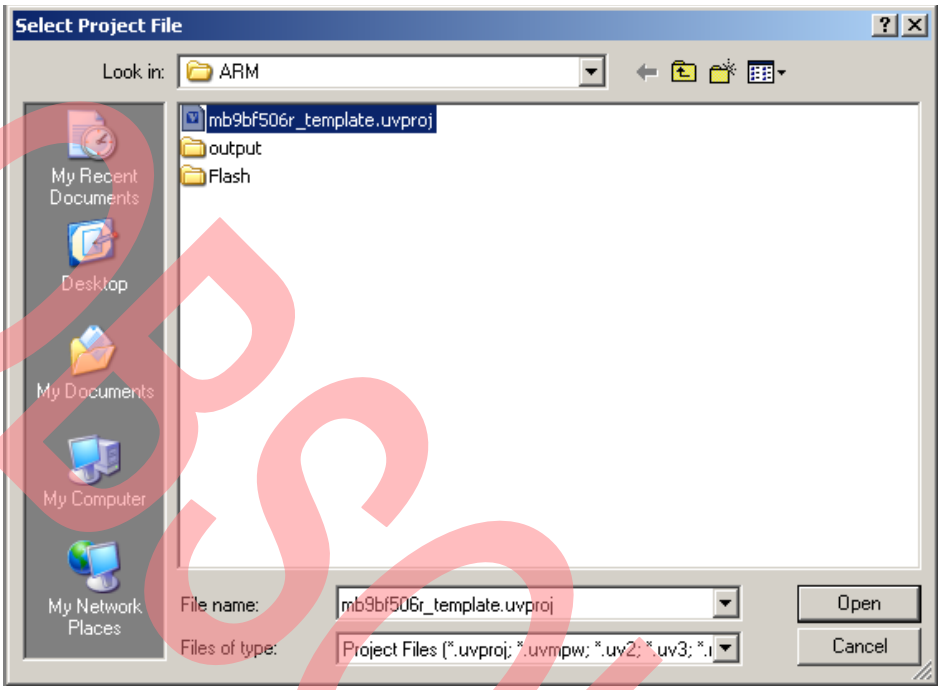
Figure 37. Open a Project



OBSOLETE

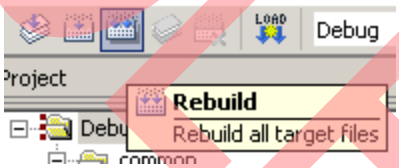
2. Select a project (uvproj file)

Figure 38. Select a Project



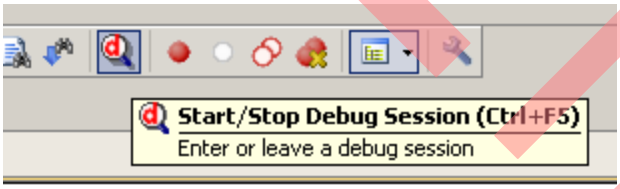
3. Rebuild all

Figure 39. Rebuild All Files



4. Start debug

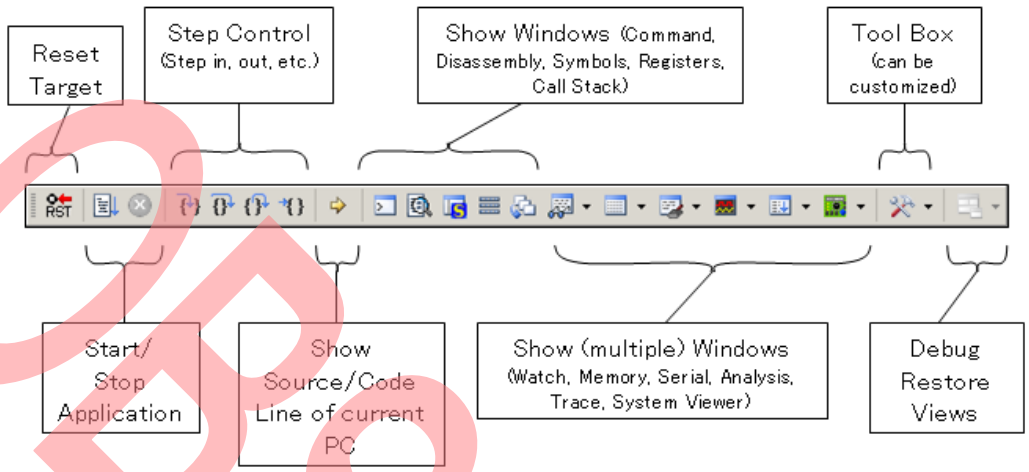
Figure 40. Start Debug



OBSOLETE

5. Use following tool bar to debug

Figure 41. Debug Tool Bar

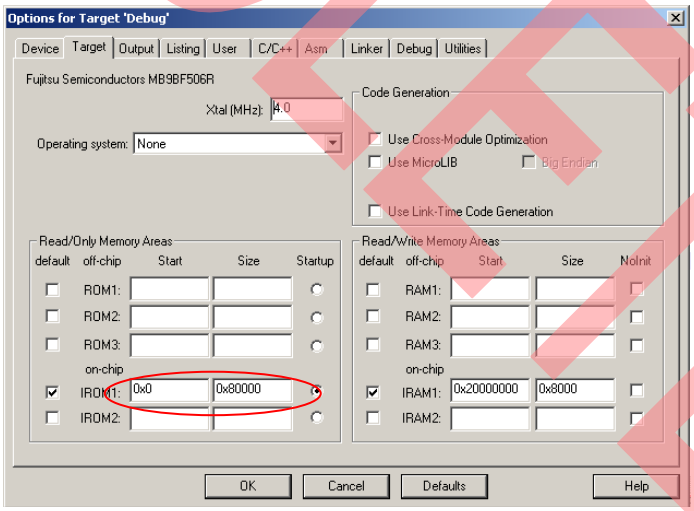


The sample codes support both Flash and RAM debug in Keil uVison 4, if Flash debug is used, the code is programmed into MB9BF506 Flash. If RAM debug is used, the code only runs in RAM area, and after power off, the code will not be stored, but the RAM debug will be faster than Flash debug.

- Setting for Flash Debug

1. Set ROM address in Flash area. (0x00000000-0x00080000)

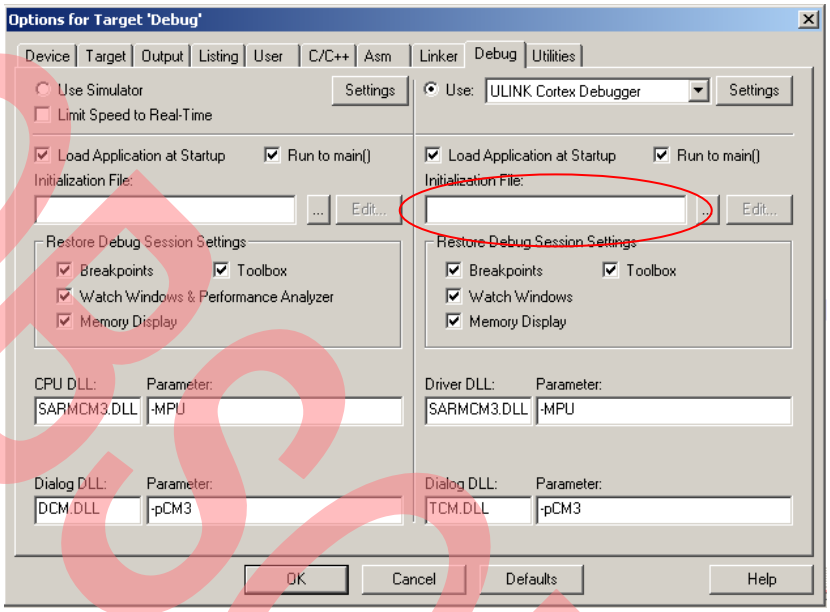
Figure 42. ROM Address Setting for Flash Debug



OBSOLETE

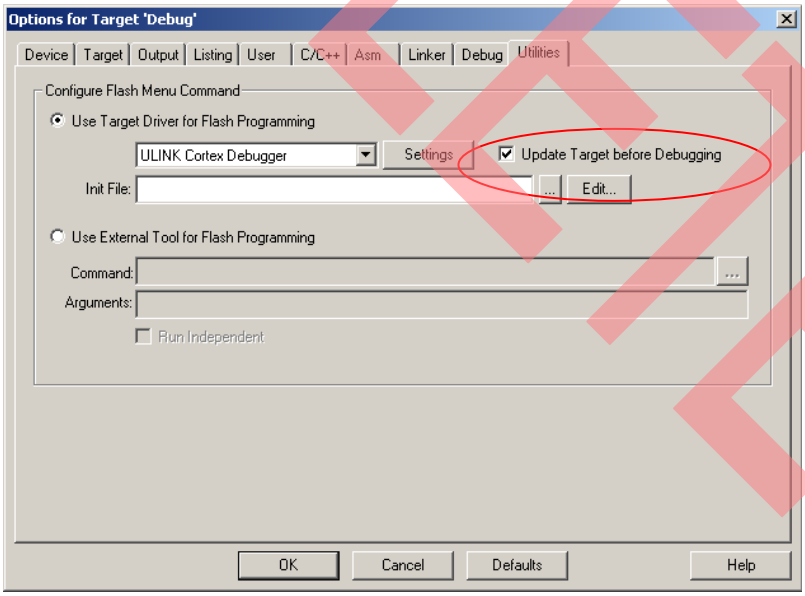
2. Don’t use initialization file.

Figure 43. Don’t use initialization File



3. Check “Update Target before Debugging” checkbox

Figure 44. Select Update Target Before Debugging

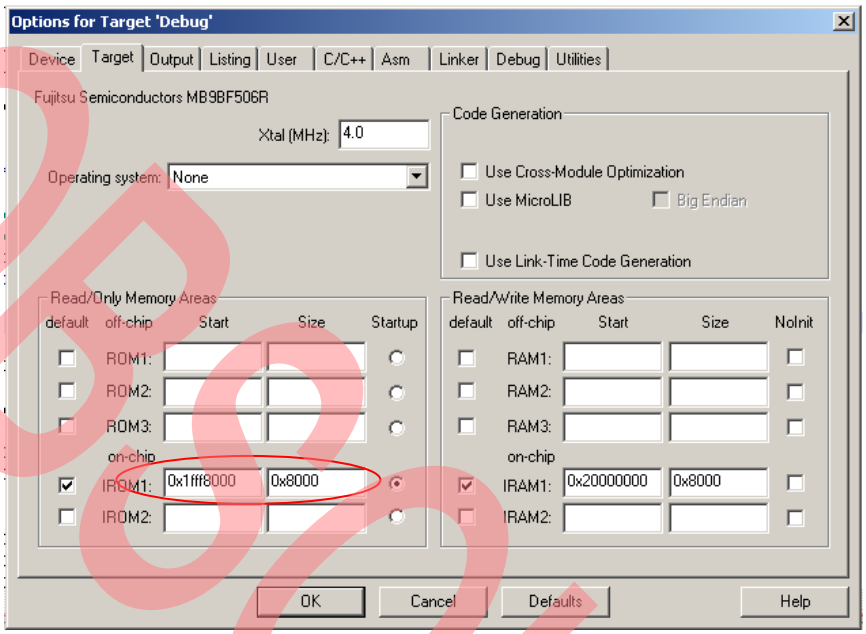


OBSOLETE

- Setting for RAM Debug

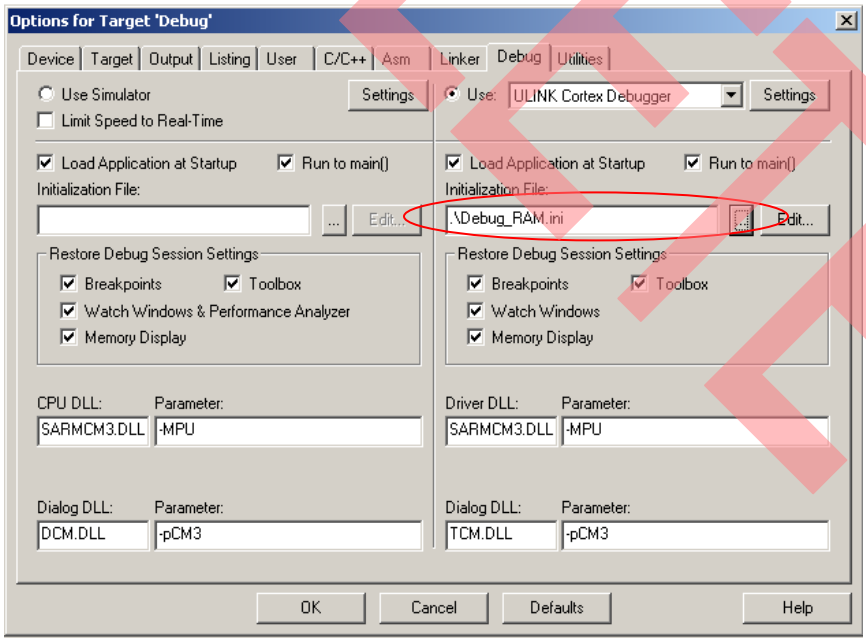
1. Set ROM address in Code SRAM area. (0x1fff8000-0x1fffffff)

Figure 45. Set RAM Address for RAM Debug



2. Set initialization file path. (..\Debug_RAM.ini)

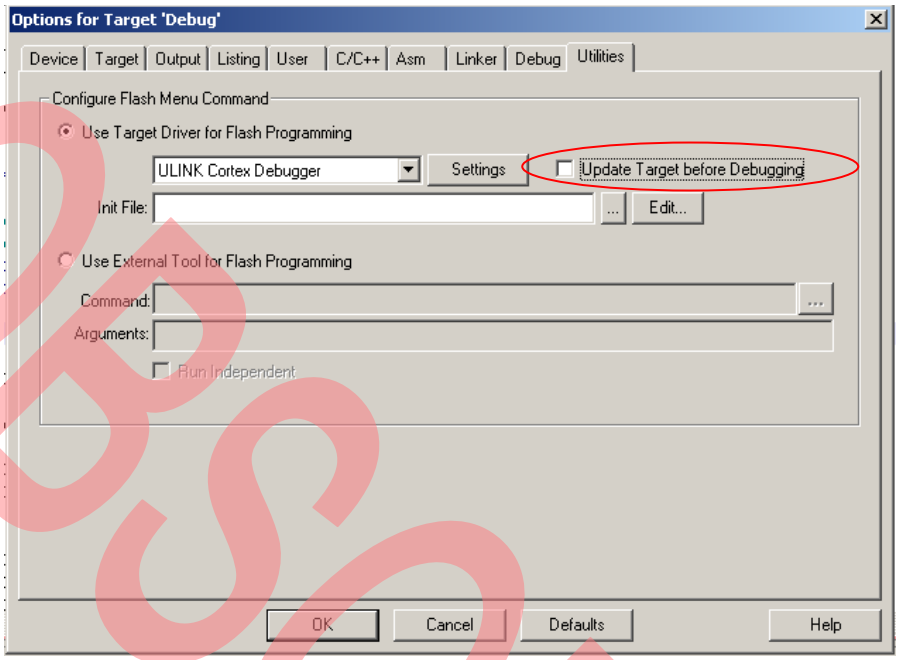
Figure 46. Select Initialization File



OBSOLETE

3. Don't Check "Update Target before Debugging" checkbox

Figure 47. Select Update Target Before Debugging

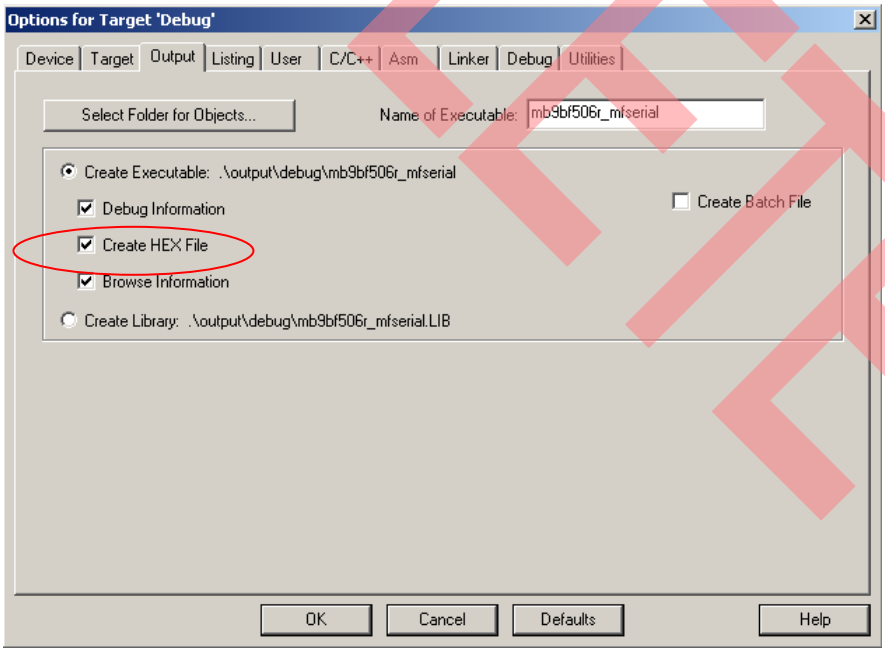


If user need to program the hex file into Flash via UART or USB programmer, a hex file need to be produced first.

- How to Make a HEX File

1. Check "Create HEX File" checkbox (This file is Intel Format HEX)

Figure 48. HEX File Generation in Keil IDE



2. User can find the generated file in path "..\output\debug\"

OBSOLETE

6 Materials Download

The following materials can be downloaded from below web site.

http://www.spansion.com/CN/Products/microcontrollers/32-bit-ARM-Core/Pages/fm3-fssdc-9b506-evb.aspx

- Software
  - Cypress Flash MCU Programmer
  - Cypress Flash USB DIRECT Programmer
- Document
  - FSSDC-9B506-EVB User Manual
  - FSSDC-9B506-EVB Schematic
  - MB9B506 Series Datasheet
  - MB9B506 Series Peripheral Manual
  - MB9B506 Series Flash Programming Manual
- Sample code
  - uart
  - iic_rtc
  - ext_bus_nandflash
  - can
  - usb_func_mouse
  - usb_host_catch_mouse
  - lcd_ad_key
- IDE Study Material
  - IAR IDE study material
  - Keil IDE study material

OBSOLETE

A Appendix

A.1 Links to Microcontrollers

A.2 Software and Learning materials

- http://www.spansion.com/CN/Products/microcontrollers/32-bit-ARM-Core/Pages/fm3-fssdc-9b506-evb.aspx
- FM3 32 Bit Microcontrollers www.cypress.com/cypress-microcontrollers

A.3 Links to third party sites

- Cypress plug-in update site: www.spansion.com/Products/microcontrollers/32-bit-ARM-Core/fm3/Pages/overview_32fm3.aspx

OBSOLETE

Document History

Document Title: AN205235 - FM3 Microcontroller MB9B500 Series, FSS MB9BF506R EV-BOARD

Document Number: 002-05235

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	—	HUAL	04/21/2011	Initial release.
*A	5268523	HUAL	07/01/2016	Migrated Spansion Application Note from MCU-AN-510014-E-10 to Cypress format. Link to the Hardware and Firmware doesn't exist and this AN to be Obsolete.

OBSOLETE

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
Lighting & Power Control	cypress.com/powerpsoc
Memory	cypress.com/memory
PSoC	cypress.com/psoc
Touch Sensing	cypress.com/touch
USB Controllers	cypress.com/usb
Wireless/RF	cypress.com/wireless

PSoC® Solutions

PSoC 1 | PSoC 3 | PSoC 4 | PSoC 5LP

Cypress Developer Community

Forums | Projects | Videos | Blogs | Training | Components

Technical Support

cypress.com/support

OBSOLETE

PSoC is a registered trademark and PSoC Creator is a trademark of Cypress Semiconductor Corporation. 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

Phone : 408-943-2600
Fax : 408-943-4730
Website : www.cypress.com

© Cypress Semiconductor Corporation, 20011-2016. 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. 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, 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.