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THIS SPEC IS OBSOLETE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: 002-04464

Spec Title: AN204464 - SK_FM4_U_Peripheral 32-Bit Microcontroller FM4 Family User Manual

Replaced by: None

AN204464

SK_FM4_U_Peripheral 32-Bit Microcontroller FM4 Family User Manual

This application notes describes how to use the demo board of SK_FM4_U_Peripheral.

Contents

1 Introduction

1.1 About Document

This application notes describes how to use the demo board of SK_FM4_U_Peripheral.

1.2 About Peripheral Board

The peripheral board is a demo board used on FM4 MCU (such as MB9B560 serial MCUs), including 11 modules (Power, LCD, JTAG, TF, USB slave, EEPROM, LIN, SPI, CAN, UART and AUDIO). The peripheral board is demo board that must be connected with a MCU board (such as SK_FM4_U120_9B560).

Note:

For more information please refer to www.cypress.com/documentation/development-kitsboards/sk-fm4-u-peripheral.

1.3 About MB9B560 SERIAL MCU

MB9B560 series MCU is 32-bit general purpose MCU of FM4 family that features the industry's leading-edge ARM Cortex-M4F CPU and integrates Cypress's highly reliable and high-speed secure embedded flash technology. This MCU can operate at up to 160MHz CPU frequency and work at a wide voltage range (2.7-5.5V), 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 , ADCs, on-chip memory (up to 1024K main Flash and 32K work Flash, up to 128K RAM) 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 Table 1.

Table 1. MB9BF560 Product List

Product	Main Flash	Work Flash	SRAM	Package
MB9BF564K/L	256Kb	32kB	32kB	L: LQFP: FPT-64P/QFN: LCC-64P K: LQFP: FPT-48P/ QFN: LCC-48P
MB9BF565K/L	384kB	32kB	48kB	L: LQFP: FPT-64P/QFN: LCC-64P K: LQFP: FPT-48P/ QFN: LCC-48P
MB9BF566K/L	512kB	32kB	64kB	L: LQFP: FPT-64P/QFN: LCC-64P K: LQFP: FPT-48P/ QFN: LCC-48P
MB9BF566M/N/R	512kB	32kB	64kB	M: LQFP: FPT-80P N: QFP: FPT-100P/ BGA: BGA-112P R: LQFP: FPT-120P/ BGA: BGA-144P
MB9BF567M/N/R	768kB	32kB	96kB	M: LQFP: FPT-80P N: QFP: FPT-100P/ BGA: BGA-112P R: LQFP: FPT-120P/ BGA: BGA-144P
MB9BF568M/N/R	1024kB	32kB	128kB	M: LQFP: FPT-80P N: QFP: FPT-100P/ BGA: BGA-112P R: LQFP: FPT-120P/ BGA: BGA-144P

1.4 About FM4 peripheral Demo Project

This is a sample project to demonstrate how to use FM4 peripheral board. It is developed in IAR EWARM Workbench V6.60, and works on SK_FM4_U120_9B560

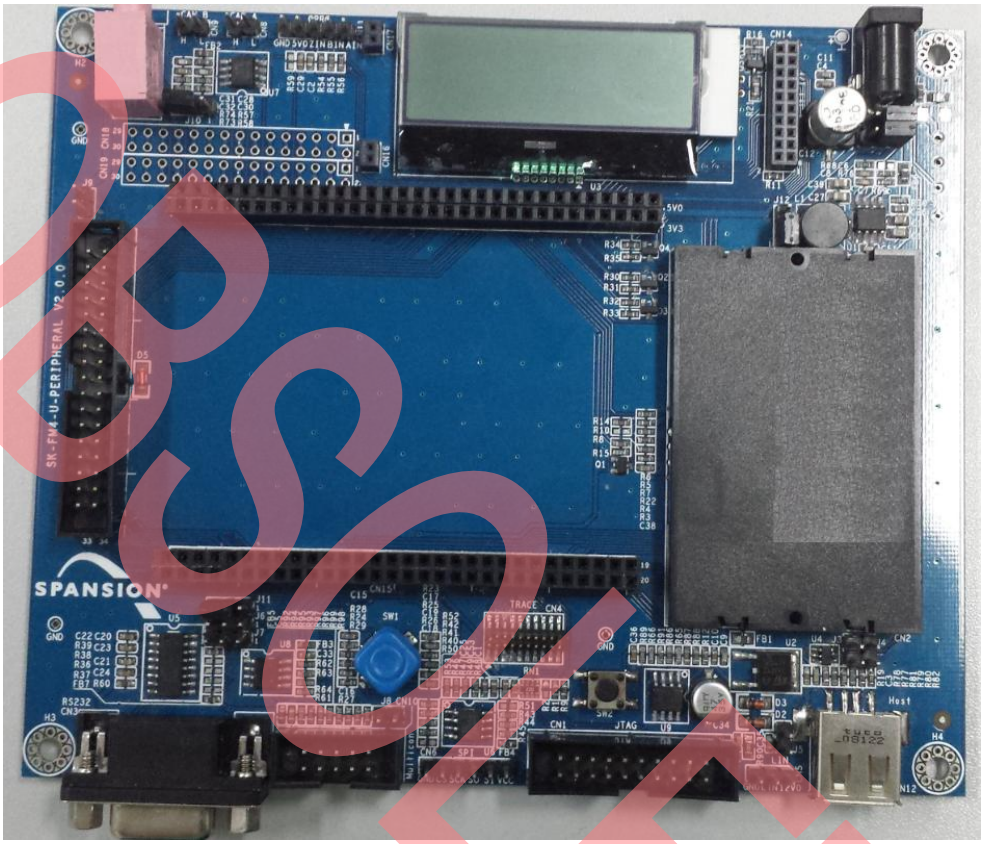
Note:

1. If the later version of IAR EWARM Workbench V6.21 is used to open this example project, MCU type information in project setting may lose, please check it.
2. If user select the “Use CMSIS” option (in Library Configuration table of General Options) with IAR EWARM Workbench V6.20 later, please erase the head files with prefix “core_” in common folder (path: ..¥..¥common).
3. If the former version of IAR EWARM Workbench V6.21 is used to open this example project, MCU type, pre-included files (in preprocess table of C/C++ compiler), icf file (in link table of debug option), flash loader file (down table of debugger option) may lose, please check these settings.
4. if the former version of Keil µVision V4.20 is used to open this example project, MCU type, pre-included files (in C/C++ table of project option), debug setting (in debug table of project setting) may lose, please check these information.
5. If user uses former version of IAR EWARM Workbench V6.20, it is a MUST to copy two files (core_cmFunc.h and core_cmInstr.h in cmsis_low_ver folder in current project directory) to common folder (path: ..¥..¥common).

2 System Overview

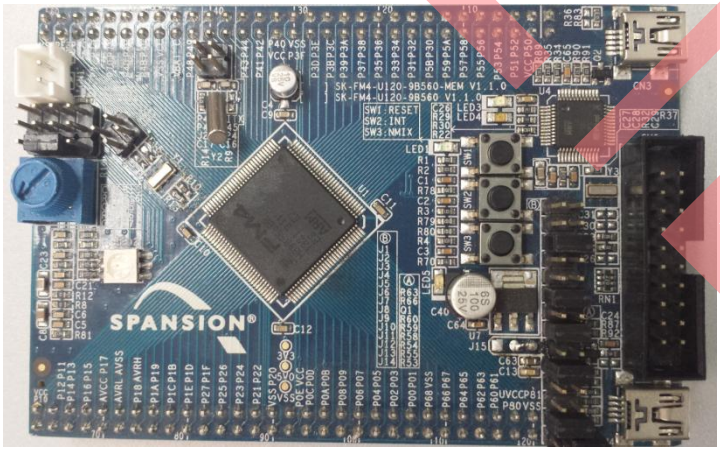
The peripheral board includes LCD, CAN, AUDIO, TF, USB, LIN, SPI, KEY, EEPROM and UART.

Figure 1. Main Board



The peripheral board does not solder with a MCU, so if user want to use it, a MCU is need

Figure 2. MCU Board



3 Modules Description

3.1 Power

3.2 Keys

There is an ADC key on the peripheral board, which contains 5 functions: left, right, up, down and enter

There are three Keys on the MCU board: reset, SW2 (module switch) and SW3 (module confirm)

Figure 3. Module Switch Keys

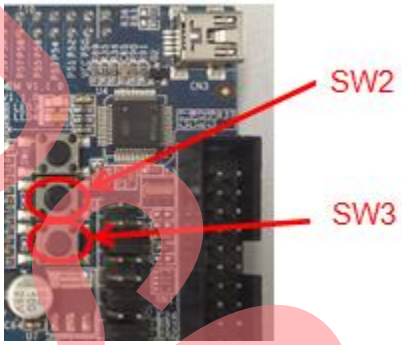


Figure 4. Operation Keys

3.3 TF card

There is a TF card socket, user can insert a card and modify system firmware to display the contents of card. In sample code the firmware just can detect the card and show a test successful on LCD.

Operation steps:

6. SW2 to switch the module to TF and SW3 to enter TF
7. Press enter key to start TF operation 

8. Press right key and enter key, system will detect card
9. After system detected a card, LCD will display 

Note:

When using CSK of smart card, the jumper J12 pin1 and pin2 are need to be connected,

When using lever shift DIR control, the jumper J12 pin2 and pin3 are need to be connected

3.4 USB

The peripheral board can detect USB device and read USB content. In sample project the firmware will read folder name in USB and display the last one on LCD.

Operation steps:

1. Press SW2 to switch module to USB and press SW3 to enter USB 

2. If there is a USB in USB port, the folder name will display on LCD

Note:

If the USB host is used, the jumper J3 and J4 need to be closed,

If the USB slave is used, the jumper J3 and J4 need to be open.

3.5 EEPROM

There is a 2K EEPROM soldered on peripheral, user can save 2K data in it.

Operation steps:

1. Press SW2 to switch module to EEPROM and press SW3 to enter
2. The right and left key can move cursor
3. The up and down key can change the address, data and read/write mode
4. The enter key used to enter the write/read operation

3.6 LIN

The LIN module can connect with another LIN board and communication with it.

The sample code of peripheral board execute receiving mode, when data transfer the LCD will display “test OK”.

Operation steps:

1. Connect “12V VCC”, “GND” and “LIN” to LIN board 

2. Press SW2 to switch module to LIN and press SW3 to enter 

3. If there is a LIN data send into peripheral board,
4. The LCD will display 

Note:

If the mode is master the jumper J5 need to be connected.

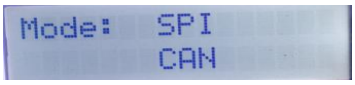
If the mode is slave, the jumper J5 is open

3.7 SPI

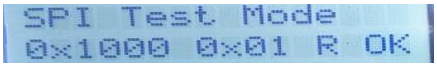
There is a Cypress MCU MB85RS64V in board. The MB85RS64V is a FRAM (Ferroelectric Random Access Memory) chip in a configuration of 8,192 words x 8 bits.

User can save data from 0x0000 to 0x1000 in FRAM.

Operation steps:

1. Press SW2 to switch the module to SPI, and press SW3 to enter 

2. The right and left key can move cursor 

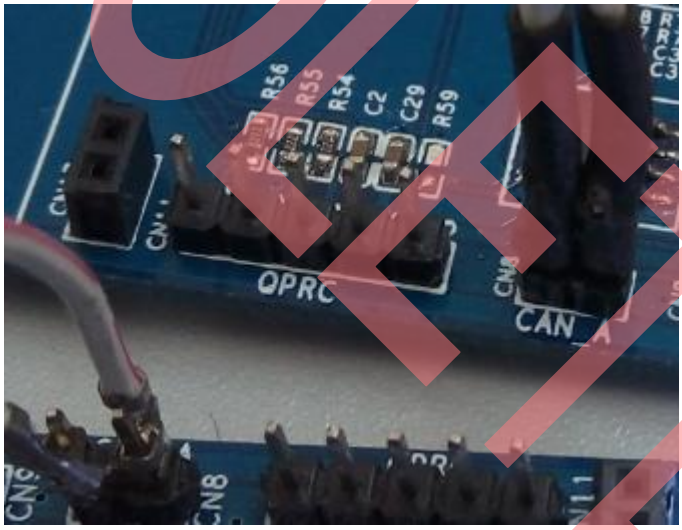
3. The up and down key can change the address, data and read/write mode
4. The ahead six data is the address of FRAM
5. The seventh to tenth data is the data that read from/write to FRAM
6. When “R” is display, press enter key the data can be read from FRAM
7. When “W” is display, press enter key the data will be wrote to FRAM
8. When operation is successful, the OK will display 

3.8 CAN

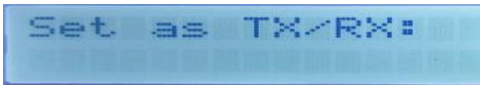
The CAN module can do send and receive when connected with another can board.

Operation steps:

1. The CAN-H connects to CAN-H, and CAN-L connects to CAN-L.



2. Press SW2 to switch the module to CAN, press SW3 to enter 

3. Press enter key to enter can operation 

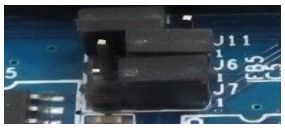
4. Press down key to “RX” and press enter to “receiving” module
5. If there is a CAN data in, the LCD will display “successful”

3.9 UART

The UART module includes RS232 and can communication with other UART module. If the UART tool is installed in pc, user can send/receive data to/from PC.

The UART baud rate is 115200 without odd-even check, data is 8bit with one bit stop.

Operation step:



1. Connect “1” and “2” in “J6” and “J7”
2. Press SW2 to switch module to UART, press SW3 to enter
3. Connect with another UART module
4. If the another module send a data to peripheral board, the LCD will display the data

Note:

Connect jumper J11 pin1 and pin2, when UART CTS.

Connect jumper J11 pin2 and pin3, when using LCD background.

3.10 DAC

There is a Phonejack on the board. User can send different pulse from DAC to generate different audio.

In sample project there is a sine wave tones will play from Phonejack.

Operation step:

1. Connect jumper J10 pin2 and pin3
2. Press SW2 to switch module to AUDIO, press SW3 to enter
3. Sine wave tones will play.

4 Sample Project

There are many communication modules in the sample project, such as IIC, LIN, SPI and CAN.

In following chapter, these communication samples will be described.

4.1 IIC

4.1.1 mi2c.c

Mi2c.c is the driver of IIC, the read/write and initial functions are in it, if user uses IIC, this file is need.

Table 2. mi2c functions list

Functions	Description	Remark
void InitI2c(void)	Initial IIC module	
en_result_t WriteEepByte(uint8_t Addr, uint8_t WrDate)	Write one byte data to appointed address	
en_result_t ReadEepByte(uint8_t Addr, uint8_t *pReadBuffer)	Read one byte data from appointed address	

If mi2c.c is used, the mfs_hl.c is need too.

4.1.2 mfs_hl.c

The functions in mfs_hl.c configurates registers of serial module include IIC.

4.1.3 EepromApp.c

The functions in EepromApp.c call IIC apps in mi2c.c to execute EEPROM operation according to user need.

Table 3. EepromApp functions list

Functions	Description	Remark
static void DisplayOperationRemind(uint8_t *pbuffer)	realize the EEPROM app write function	
static void EepromAppElemW(void)	realize the EEPROM app write function	
static void EepromAppElemR(void)	realize the EEPROM app read function	
void InitEeprom(void)	Initialize EEPROM global parameter	
static void EepromAppServce(void)	EEPROM app button&key function	
static void UpdateLcdBuffer(void)	update the LCD buffer content	
static void UpdataLcdContent(void)	LCD display	
void EepromApp(void)	realize the EEPROM app	

4.2 LIN

4.2.1 LIN_Slave.c

The LIN used functions in mfs_hl.c to realize all functions about LIN.

Table 4. LIN_Slave functions list

Functions	Description	Remark
void LIN_Init(void)	LIN Interface Initilazation as Slave	
static void SampleMfsLin(void)	Realize LIN operation	
void Lcd_LinSlave(void)	LIN Interface transmit data display on LCD	

4.3 SPI

4.3.1 mfs.c

The file includes MFS module drivers, which set module register and finish basic function. That is similar with file mfs_hl.c

4.3.2 Spi.c

Functions in spi.c realized all SPI functions, which call the derivers in mfs.c

Table 5. SPI functions List

Functions	Description	Remark
en_result_t MfsCsioWrite16(uint16_t pu8TxBuf, uint8_t* pu8RxBuf, uint16_t u16TransferSize, boolean_t bCsHolding,uint8_t Dat)	Write 16bits data	
en_result_t MfsCsioRead16(uint16_t pu8TxBuf, uint8_t* pu8RxBuf, uint16_t u16TransferSize, boolean_t bCsHolding)	Read 16bites data from FRAM	
en_result_t Spi_WrOne(uint8_t WDat, boolean_t bCsHolding)	Write one byte to FRAM without ACK	
void Fram_EnLatch(void)	Enable FRAM write	
void Fram_ResetLatch(void)	Reset FRAM latch	
void Fram_Write(uint16_t Addr,uint8_t Dat)		
void Fram_Init(void)	Initialize FRAM	
uint8_t CalculatVal(uint8_t Dat, uint8_t Rang,uint8_t Stat)		
void Key_Spi(void)	Key value calculate and status value define	
void Fram_Operation(void)	Do save and read operation	
void TaskcreateSpi(void)	Realize SPI and display status	

4.4 CAN

4.4.1 can.c

Drivers of CAN function, which detailed descripts available of CAN module.

4.4.2 Mcan.c

Use drivers in can.c to realize CAN send and CAN receive.

Functions	Description	Remark
static void CanConfigGPIO()	Setting GPIO of CAN	
static void CanConfigParameter()	Setting parameter of CAN	
boolean_t CanInitParameter()	Initialize CAN parameter	
void CanTransmitSample(void)	Do CAN send operation	
void CanReceiveSample(void)	Do CAN receive operation	
void CanKeyLCDApp(void)	CAN operation and status display	

5 Reference Documents

1. (SCH)SK-FM4-U120-9B560-v1.2.0.pdf
2. (SCH)SK-FM4-U-PERIPHERAL-V2.0.0.pdf
3. MB9B560R-DS709-00001.pdf
4. ReadMe_SK-FM4-U120-9B560_v11.pdf

6 Document History

Document Title: AN204464 - SK_FM4_U_Peripheral 32-Bit Microcontroller FM4 Family User Manual

Document Number: 002-04464

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	–	HUAL	10/24/2014	Initial Release.
*A	5028544	CHMA	11/30/2015	Migrated Spansion Application Note MBF568R_AN709-00007-1v0-E to Cypress format.
*B	5512183	CHMA	11/10/2016	Links are missing for the product and this AN to be 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

Automotive	cypress.com/go/automotive
Clocks & Buffers	cypress.com/go/clocks
Interface	cypress.com/go/interface
Internet of Things	cypress.com/iot
Lighting & Power Control	cypress.com/go/powerpsoc
Memory	cypress.com/go/memory
PSoC	cypress.com/go/psoc
Touch Sensing	cypress.com/go/touch
USB Controllers	cypress.com/go/usb
Wireless/RF	cypress.com/go/wireless
Spansion Products	spansion.com/products

PSoC® Solutions

psoc.cypress.com/solutions

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

Cypress Developer Community

Forums | WICED IOT Forums | Projects | Videos | Blogs | Training | Components

Technical Support

cypress.com/go/support

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

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