FUJITSU MICROELECTRONICS
SUPPORT SYSTEM

MCU-AN-500072-E-10

F²MC-8FX FAMILY
8-BIT MICROCONTROLLER
MB95F310 SERIES

MB2146 - 450 - E

SETUP GUIDE

FUJITSU

PREFACE

Thank you for purchasing the F2MC*1-8FX Family EV-Board: MB2146-450-E*2.
This product is an EV-Board for F2MC-8FX MB95F310 Series MCU, which comes with MB2146-08-E (F2MC-8FX BGM Adapter)*3, MB2146-450-E (F2MC-8FX Family MB95F310 Series Evaluation Board)*2 and F2MC-8L/8FX Family SOFTUNE Professional Pack Evaluation Version*4.
This manual explains how to use the EV-Board. Be sure to read this manual before using the product.
About mass production/evaluation MCUs for this product, please consult with sales representatives or support representatives.

*1: F2MC is the abbreviation of FUJITSU Flexible Microcontroller.
*2: Referring to the following as the “EV-Board”.
MB2146-450-E is the EV-Board for MB95F310 MCU;

*3: Referring to the following as the “BGMA”.
*4: Referring to the following as the “SOFTUNE”.

■ Handling and usage

Handling and usage of this product and notes regarding its safe use are described in the manuals for products bundled with the EV-Board.
Follow the instructions in the manuals to use this product.
Keep this manual at hand so that you can refer to it anytime during use of this product.

■ Notice on this document

All information included in this document is current as of the date it is issued.
The information is subject to change without any prior notice.
Please confirm the latest relevant information with the sales representatives.

■ Caution of the products described in this document

The following precautions will apply to the product described in this manual.

⚠ WARNING Indicate a potentially hazardous situation which, if not avoided appropriately, could result in death or serious injury and/or a fault in the user's system.

Electric shock, Damage	Before performing any operation described in this manual, turn off all the power supplies to the system. Performing such an operation with the power on may cause an electric shock or device fault.
Electric shock, Damage	Once the product has been turned on, do not touch any metal part of it. Doing so may cause an electric shock or device fault.

⚠ CAUTION Indicates the presence of a hazard that may cause a minor or moderate injury, damages to this product or devices connected to it, or may cause to loose software resources and other properties such as data, if the device is not used appropriately.

Cuts, Damage	Before moving the product, be sure to turn off all the power supplies and unplug the cables. Watch your step when carrying the product. Do not use the product in an unstable location such as a place exposed to strong vibration or a sloping surface. Doing so may cause the product to fall, resulting in an injury or fault.
Cuts	The product contains sharp edges that are left unavoidably exposed, such as jumper plugs. Handle the product with due care not to get injured with such pointed parts.
Damage	Do not place anything on the product or expose the product to physical shocks. Do not carry the product after the power has been turned on. Doing so may cause a malfunction due to overloading or shock.
Damage	Since the product contains many electronic components, keep it away from direct sunlight, high temperature, and high humidity to prevent condensation. Do not use or store the product where it is exposed to much dust or a strong magnetic or electric field for an extended period of time. Inappropriate operating or storage environments may cause a fault.
Damage	Use the product within the ranges given in the specifications. Operation over the specified ranges may cause a fault.
Damage	To prevent electrostatic breakdown, do not let your finger or other object come into contact with the metal parts of any of the connectors. Before handling the product, touch a metal object (such as a door knob) to discharge and any static electricity from your body.
Damage	Before turning the power on, in particular, be sure to finish making all the required connections. Furthermore, be sure to configure and use the product by following the instructions given in this document. Using the product incorrectly or inappropriately may cause a fault.
Damage	Always turn the power off before connecting or disconnecting any cables from the product. When unplugging a cable, unplug the cable by holding the connector part without pulling on the cable itself. Pulling the cable itself or bending it may expose or disconnect the cable core, resulting in a fault.
Damage	Because the structure of the MCU socket does not allow an evaluation MCU to be mounted in the incorrect orientation, be very careful of the orientation of the evaluation MCU when mounting it. Inserting the evaluation MCU in the wrong orientation may damage the MCU, causing the MCU to become faulty.
Damage	Because the product has no casing, it is recommended that it be stored in the original packaging. Transporting the product may cause a damage or fault. Therefore, keep the packaging materials and use them in case of for the re-shipment of the product.

- The contents of this document are subject to change without notice.
  Customers are advised to consult with sales representatives before ordering.
- The information, such as descriptions of function and application circuit examples, in this document are presented solely for the purpose of reference to show examples of operations and uses of FUJITSU MICROELECTRONICS semiconductor device; FUJITSU MICROELECTRONICS does not warrant proper operation of the device with respect to use based on such information. When you develop equipment incorporating the device based on such information, you must assume any responsibility arising out of such use of the information. FUJITSU MICROELECTRONICS assumes no liability for any damages whatsoever arising out of the use of the information.
- Any information in this document, including descriptions of function and schematic diagrams, shall not be construed as license of the use or exercise of any intellectual property right, such as patent right or copyright, or any other right of FUJITSU MICROELECTRONICS or any third party or does FUJITSU MICROELECTRONICS warrant non-infringement of any third-party's intellectual property right or other right by using such information. FUJITSU MICROELECTRONICS assumes no liability for any infringement of the intellectual property rights or other rights of third parties which would result from the use of information contained herein.
- The products described in this document are designed, developed and manufactured as contemplated for general use, including without limitation, ordinary industrial use, general office use, personal use, and household use, but are not designed, developed and manufactured as contemplated (1) for use accompanying fatal risks or dangers that, unless extremely high safety is secured, could have a serious effect to the public, and could lead directly to death, personal injury, severe physical damage or other loss (i.e., nuclear reaction control in nuclear facility, aircraft flight control, air traffic control, mass transport control, medical life support system, missile launch control in weapon system), or (2) for use requiring extremely high reliability (i.e., submersible repeater and artificial satellite). Please note that FUJITSU MICROELECTRONICS will not be liable against you and/or any third party for any claims or damages arising in connection with above-mentioned uses of the products.
- Any semiconductor devices have an inherent chance of failure. You must protect against injury, damage or loss from such failures by incorporating safety design measures into your facility and equipment such as redundancy, fire protection, and prevention of over-current levels and other abnormal operating conditions.
- Exportation/release of any products described in this document may require necessary procedures in accordance with the regulations of the Foreign Exchange and Foreign Trade Control Law of Japan and/or US export control laws.
- The company names and brand names herein are the trademarks or registered trademarks of their respective owners.

© 2009 FUJITSU MICROELECTRONICS LIMITED

Contents

PREFACE **1**
■ Handling and usage 1
■ Notice on this document 1
■ Caution of the products described in this document 2
CONTENTS **4**
1 PRODUCT OVERVIEW **6**
1.1 Objective and Deliverable 6
1.2 System Block 6
1.3 Handling Precautions 6
1.4 Feature 7
1.5 Hardware Setup 7
2 BGMA MANUAL **8**
2.1 BGMA Overview 8
2.2 Function List 8
2.3 IDC10 Interface Description 9
2.4 BGMA USB Configuration 9
2.5 LED Description 12
3 EV-BOARD MANUAL **13**
3.1 EV-board Overview 13
3.2 Function List 14
3.3 EV-board Schematic 15
3.4 HW Module Description and Jumper settings 16
3.4.1 Pin Assignment of MB95F310 16
3.4.2 Power Module 17
3.4.3 I2C Module 18
3.4.4 A/D Module 18
3.4.5 LCD Module 20
3.5 Operation Manual 21
3.5.1 Mode Switch 21
3.5.2 EEPROM Sub-address Switch and EDIT 22
3.5.3 Factory Mode 24
3.5.4 Temperature Operation 24
3.5.5 UART Send 25
3.5.6 RTC Operation 26
3.5.7 Wake Up Operation 28

3.5.8 Reset Operation 29
3.5.9 Test Pin 29
3.5.10 Battery Usage 29
4 SAMPLE CODE MANUAL 30
4.1 Project Structure 30
4.2 Source Code File Description 31
4.2.1 ADC .c 31
4.2.2 Beep .c 32
4.2.3 EEPROM .c 32
4.2.4 Ext-interrupt .c 32
4.2.5 I2C .c 33
4.2.6 LCD .c 33
4.2.7 RTC .c 34
4.2.8 UART .c 35
4.2.9 delay_try Function 35
4.3 Global Variable Description 36
4.4 How to Add these Library 37
4.5 Usage Demo 37
5 DEVELOPMENT PLATFORM QUICK START 38
5.1 Tools Setup Sequence 38
5.2 Open Project and Start Debug 38
5.3 Operation Precautions 38

1 Product Overview

This product is a set of EV-Board of MB95f310 series. It is composed of a BGMA (MB2146-08-E) and an EV-board (MB2146-450-E). Combining the SOFTUNE Workbench on PC, the EV-Board enables the quick start of development before the user system is ready.

1.1 Objective and Deliverable

The EV-Board provides users a complete development platform. Before starting using the EV-Board, please make sure that the following devices are placed in the package:

- ✓ EV-board (MB2146-450-E) 1PCS;
- ✓ Hardcopy (China RoHS report, quick start guide) 2PCS

1.2 System Block

To set up a debugging system, first connect a PC, a BGMA and an EV-board together shown as follows:

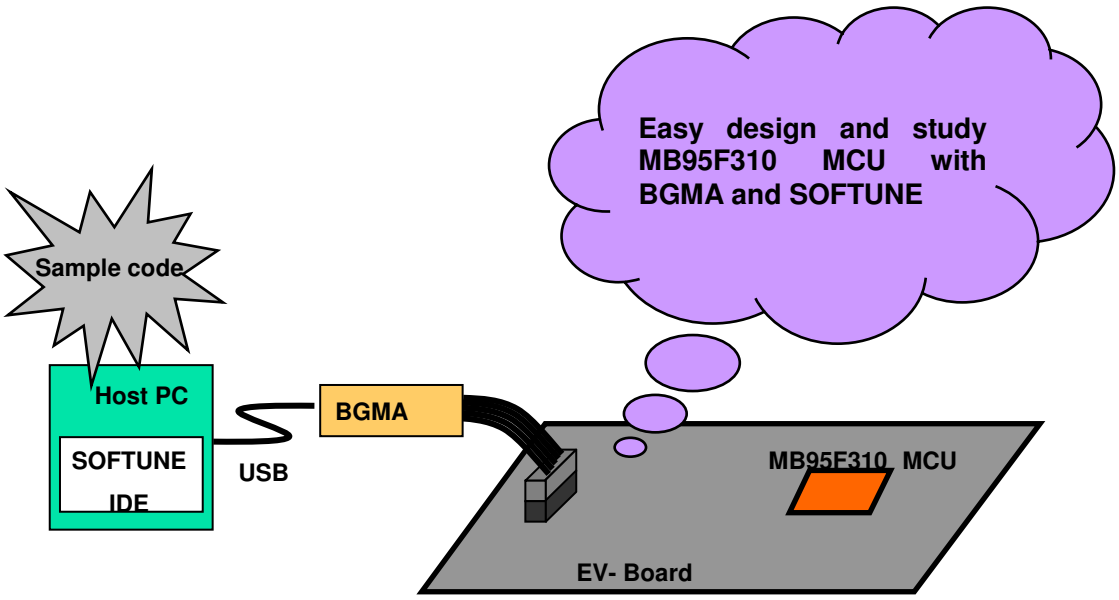


Figure 1.2-1 System Block

1.3 Handling Precautions

The EV-Board can be used in connection with its bundled products. To ensure use of this product correctly in a proper environment, please observe the following guideline:

- Follow the instructions described in each manual for the bundled product to use this product.

1.4 Feature

The MB95F310 Series EV-Board is the best for a performance and functional evaluation, and a checking of operation before including MB95f310 Series MCU in a user's system.

The feature of the BGM debugger for MB95f310 Series MCU is shown as follows.

- Microcomputer operation voltage. It corresponds to +1.8V to +3.3V.
- Compact development environment, a light and small BGM Adapter.
- Since a monitor program is performed in exclusive memory space, it does not consume user memory space.
- Continuation execution, step execution and break correspondence.
- It connects with a host computer by the USB interface.

1.5 Hardware Setup

In the hardware setup procedure, you should configure and connect the hardware products. This chapter includes the configuring and connecting procedure for each product in-order. Check the contents and complete the hardware setup.

- Configuration of each product
  - Configuring EV-Board
- Connection of each product
  - Connecting BGMA and EV-board
  - Connecting EV-Board power supply

2 BGMA Manual

This chapter gives introduction how to set up BGMA.

2.1 BGMA Overview

Following is the close shot of a MB95200 Series BGMA. The Part Number is MB2146-08-E. It provides a debug platform for the MB95F310 Series MCU in a small size (55.7mm (W) X127mm (D) X30mm (H)).

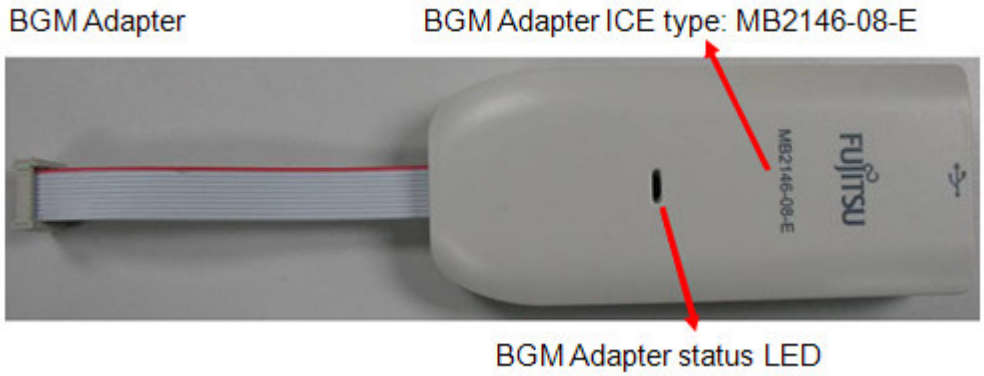


Figure 2.1-1 BGMA overview

2.2 Function List

ID	Function description	Remarks
1	Support MB95F310 Series MCU	MCU MAX machine clock: 16.25 MHz MCU power voltage: 1.8*1*2 V ~ 3.3*1 V
2	Break pointer	256 software breakpoints
3	USB interface to PC/SOFTUNE	Compatible to USB protocol version 1.1
4	1-Line UART interface to the MB95F310 Series MCU	The Baud rate is 62,500 bps
5	Support the MCU flash programming for engineering development	The program and read speed is about 800 B/S.

*1: The value varies is depending on the operating frequency, the machine clock or the analog guaranteed range.

*2: The value is 1.9V when the low-voltage detection resetting is used.

*3: The threshold voltage can be set to be 1.9, 2.35, 2.85V by software.

2.3 IDC10 Interface Description

Pin Number	Pin Name	Description
1	UVCC	Target MCU Vcc
2	GND	Target MCU Vss
3	RSTIN	Target MCU reset input
4	RSTOUT	Target MCU reset output
5	RSV	Reserved
6	RSV	Reserved
7	RSV	Reserved
8	DBG	Target MCU debug pin
9	RSV	Reserved
10	RSV	Reserved

2.4 BGMA USB Configuration

The BGMA is provided with a USB cable. Connect the BGMA to a PC with a USB cable. If the connection is right, the following window will pop up. Follow the instructions displayed, and then click “Next”,



Figure 2.4-1 Install BGMA in Windows (1)

Select “Install from a list or specific location (Advanced)”, then click “Next”,

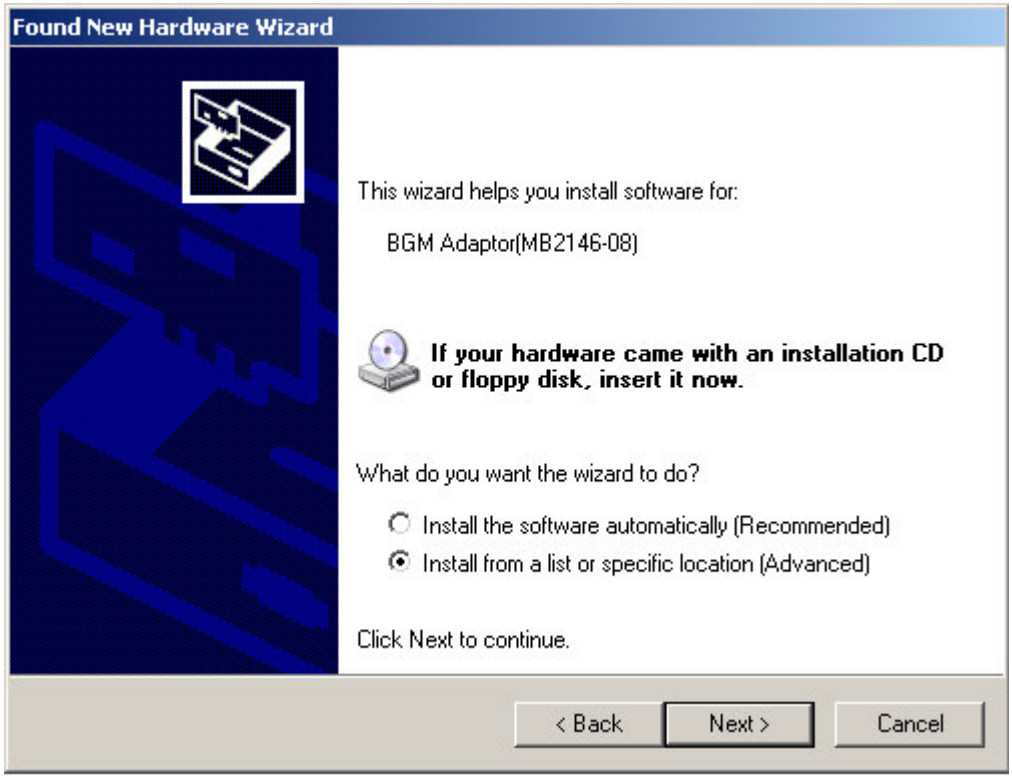


Figure 2.4-2 Install BGMA in Windows (2)

Select “…\Drivers” from the folder where SOFTUNE is installed, click “Next”,

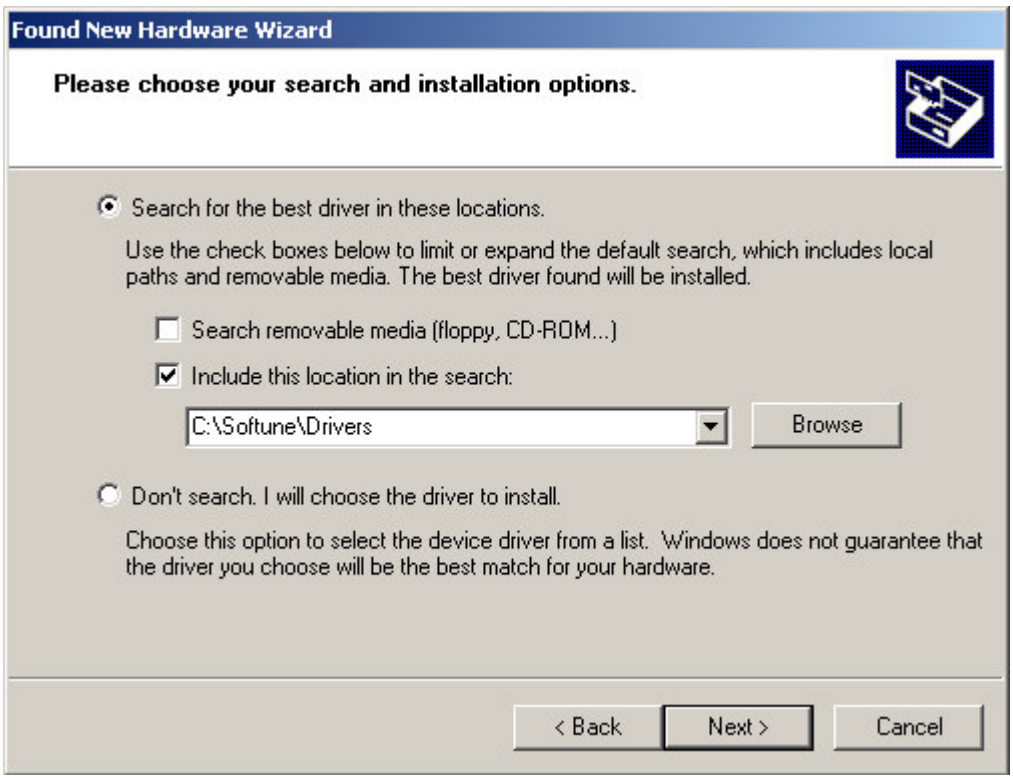


Figure 2.4-3 Install BGMA in Windows (3)

Select BGMA (MB2146-08) displayed as follows, and then click “Next”,

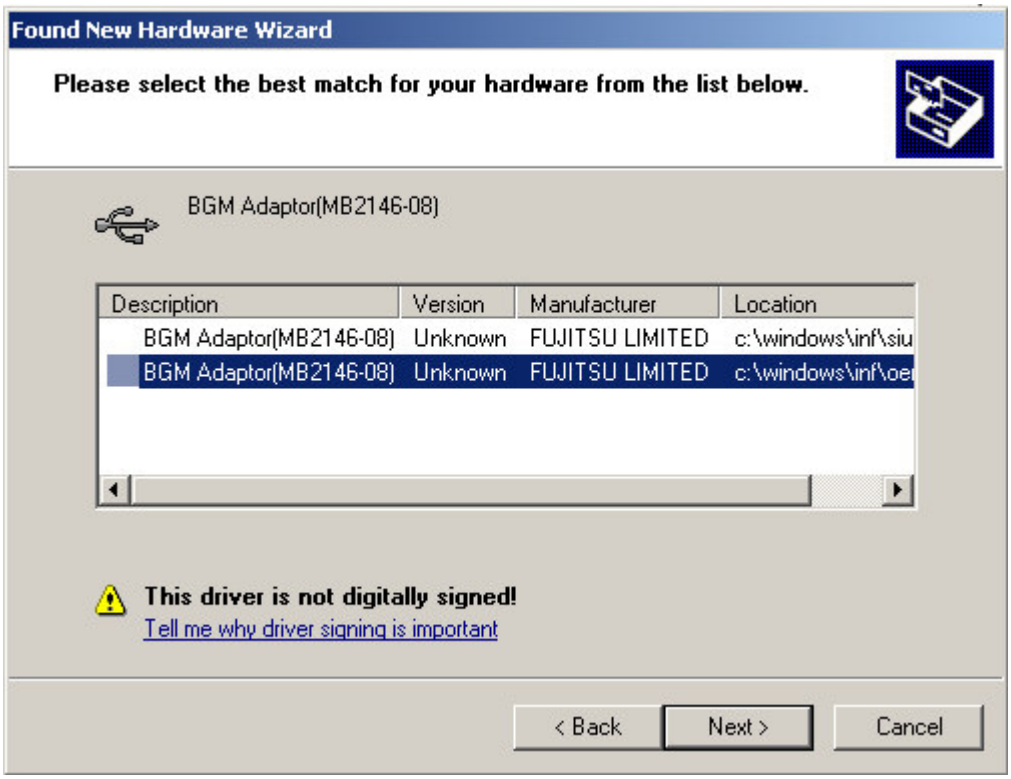


Figure 2.4-4 Install BGMA in Windows (4)

Windows will install the driver automatically. Click “Finish” after the driver has completed the installation normally. Then users can find the BGMA is recognized as MB2146-08 in Windows system.

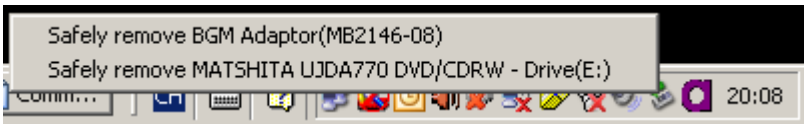


Figure 2.4-5 BGMA is installed in Windows

2.5 LED Description

First, when USB cable is plugged to PC, check the Power LED turning Green or not. Refer to Figure 2.5-1.



Figure 2.5-1 BGMA Power LED (1)

Second, plug IDC10 cable to the EV-board (target MCU board), then turn on EV-board. After that, check Power LED on the BGMA turning Orange or not. Refer to Figure 2.5-2.

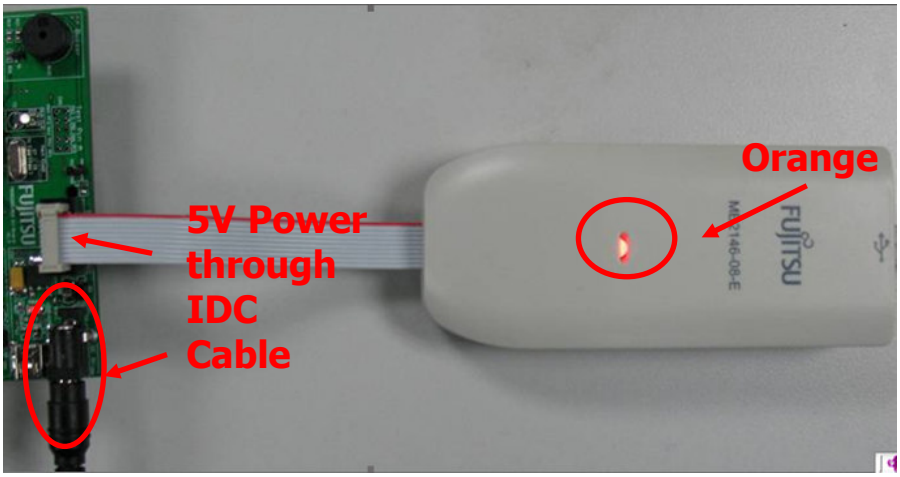


Figure 2.5-2 BGMA Power LED (2)

3 EV-board Manual

This chapter gives introduction how to setup EV-board and functions of EV-Board.

3.1 EV-board Overview

MB95F310 EV-board is an evaluation platform for the MB95F310 Family microcontroller.

Figure 3.1-1 below is a close shot of EV-board.

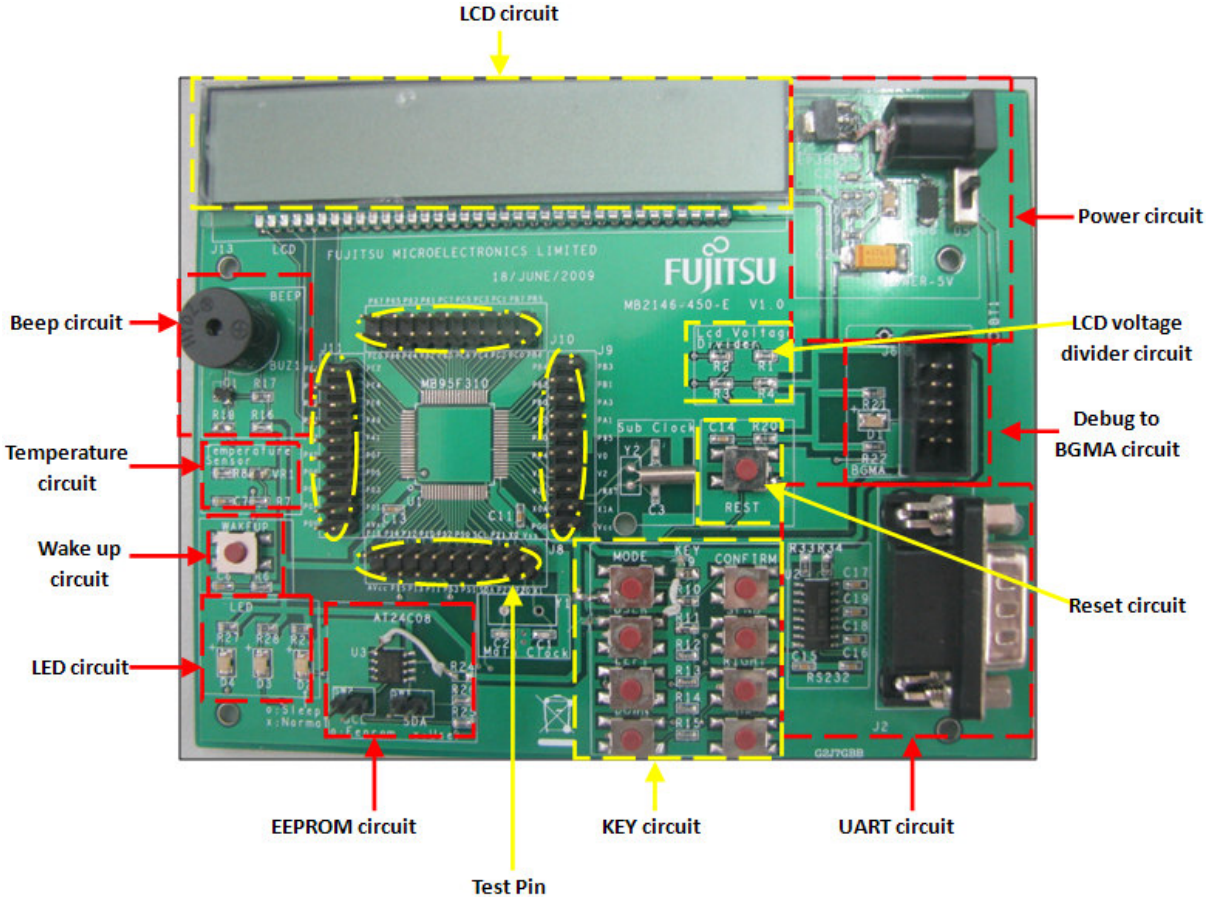


Figure 3.1-1 EV-board Overview

3.2 Function List

The EV-board is consisted of a board and a sample firmware. The board provides a useful platform for using the MCU and its peripherals. It is a useful development platform together with a BGMA (PN: MB2146-08-E) and a SOFTUNE. It displays the following functions,

- ✓ Support LCD, can display five letters and six numbers
- ✓ Support I2C, can write/read data to/from EEPROM
- ✓ Support UART, can send and receive data asynchronous
- ✓ Support temperature detect, can detect temperature from -40℃ to 85℃
- ✓ Support low power control, can set standby time
- ✓ Support mode switch, mode switch “EEPROM -> temperature -> UART -> RTC -> EEPROM….”
- ✓ Support wake up from standby mode
- ✓ Support beep, when key pressed beep sound out and then off

3.3 EV-board Schematic

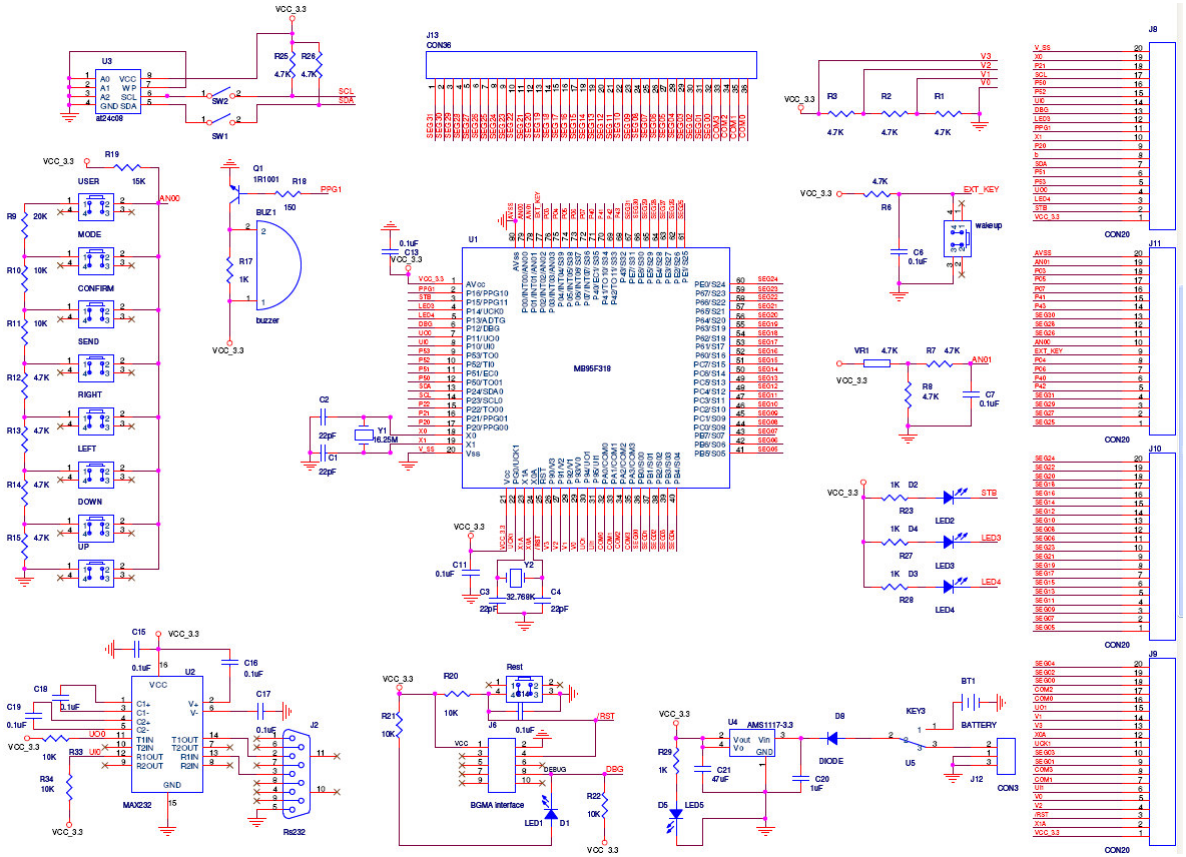


Figure 3.3-1 EV-board Schematic

3.4 HW Module Description and Jumper settings

3.4.1 Pin Assignment of MB95F310

Table 3.4.1 - 1 describes assignment of MCU Pin

Pin name	Pin Usage Description	Logical Description
P16	PPG1	Low drive beep
P15	STB	Low drive LED
P14	LED3	Low drive LED3
P13	LED4	Low drive LED4
P12	DBG	--
P11	UO0	--
P10	UI0	--
P24	SDA	--
P23	SCL	--
X0	external main clock	--
X1	external main clock	--
X1A	external Sub-clock	--
X0A	external Sub-clock	--
/RST	reset Pin	Low reset
V0	LCD voltage divider	Link to GND
V1	LCD voltage divider	Divider voltage 1.1V
V2	LCD voltage divider	Divider voltage 2.2V
V3	LCD voltage divider	Divider voltage 3.3V
PA0	COM0	--
PA1	COM1	--
PA2	COM2	--
PA3	COM3	--
PB0 to PE7	SEG00 to SEG31	--
P02	external interrupt	Low drive interrupt
P01	temperature AD	AD value from 0 to 0xff
P00	AD key	No key AD value is 0xff

Table 3.4.1 - 1 Assignment of Pin

3.4.2 Power Module

EV-board has 2 kinds of power supplies for user to choose. Please read the following instructions before using.

✓ DC Adaptor: 5V DC:

Output voltage: 3.3V

Connection: Connector (J12)

✓ Battery:

QTY: 4PCS;

Model: AA;

Nominal voltage: 1.5V.

Connection: Socket for Battery

If any of the power supplies is connected to the EV-board correctly, power LED (D5) on the EV-board will be on. Refer to Figure 3.4.1-1.

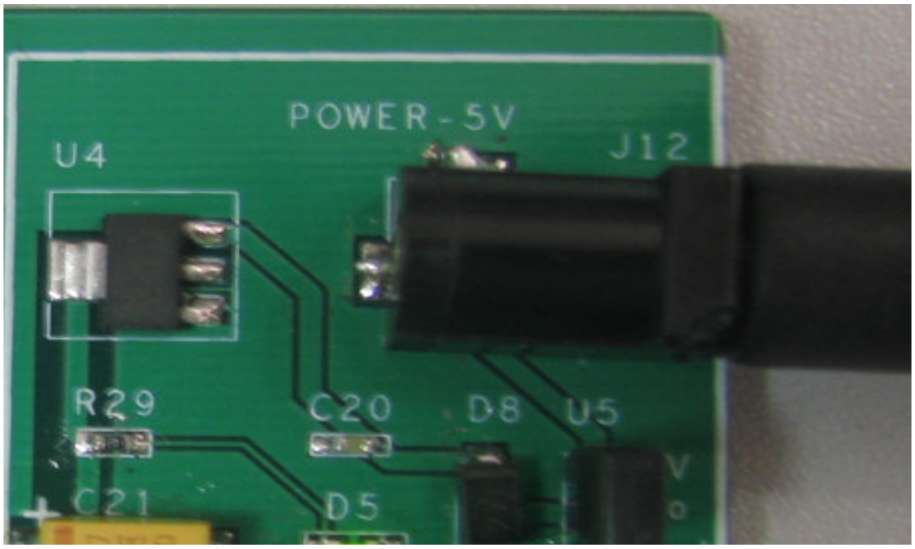


Figure 3.4.2 - 1 Power Module

The following two power supplies are recommended. Please follow the settings below,

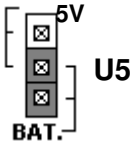
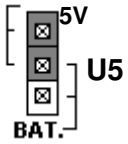
Power supply	Header name	Settings
4 AA batteries from BT1 on the back of the EV-board.	U5: BAT.	
5V DC from J12	U5: 5V.	

Table 3.4.2 - 1 Selection of Power Supply

3.4.3 I2C Module

There are two switches between at24c08 and EEPROM, when using i2c to visit EEPROM, it will switch off and when using i2c to visit other external chip, it will switch on. Following settings is recommended:

Modules	Header name	Settings
EEPROM	SW1: on. SW2: on	
External chip	SW1: off. SW2: off	

Table 3.4.3 - 1 I2C Switch

3.4.4 A/D Module

Key module and temperature module use the ADC to detect the real-time operation and environment temperature.

For key module please refer to the following figure 3.4.4 - 1

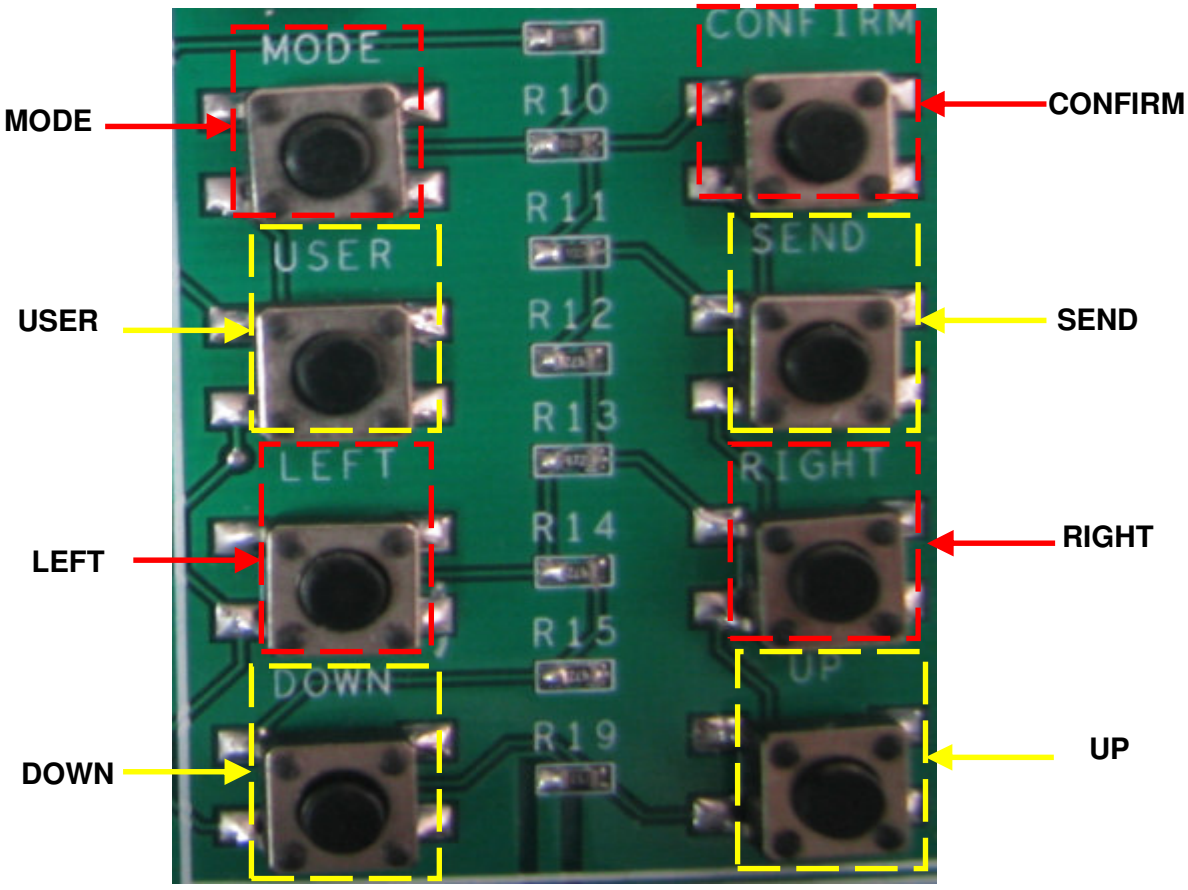


Figure 3.4.4 - 1 Key Module

Table 3.4.4 – 1 describes the function of keys

KEY	FUNCTION
MODE	Switch mode:"EEPROM -> Temperature -> UART -> RTC -> EEPROM…."
CONFIRM	Confirm some operation when want to enter into edit mode or confirm edit result
USER	Reserve for user
SEND	Send data by UART send function
LEFT	Move led position to left
RIGHT	Move led position to right
DOWN	Reduce value by 1 step or reduce EEPROM sub-address by 1 step
UP	Add value by 1 step or add EEPROM sub-address by 1 step

Table 3.4.4 - 1 KEY Description

3.4.5 LCD Module

Using COM0~COM3 and SEG00 to SEG31 to drive LCD, table 3.4.5 – 1 describes the relationship between SEG and LCD

SEG(00…..31)	LED(0…10)
SEG00	LED0
SEG01	
SEG02	LED1
SEG03	
SEG04	
SEG05	
SEG06	LED2
SEG07	
SEG08	
SEG09	
SEG10	LED3
SEG11	
SEG12	
SEG13	
SEG14	LED4
SEG15	
SEG16	
SEG17	
SEG18	LED5
SEG19	
SEG20	
SEG21	
SEG22	LED6
SEG23	
SEG24	LED7
SEG25	
SEG26	LED8
SEG27	
SEG28	LED9
SEG29	
SEG30	LED10
SEG31	

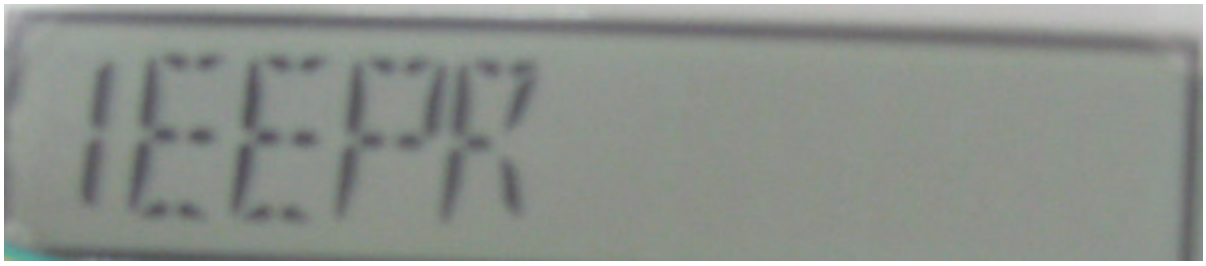
Table 3.4.5 - 1 LCD Assignment

3.5 Operation Manual

3.5.1 Mode Switch

Pressing mode key can switch EEPROM mode to RTC mode, following figure describes the operation.

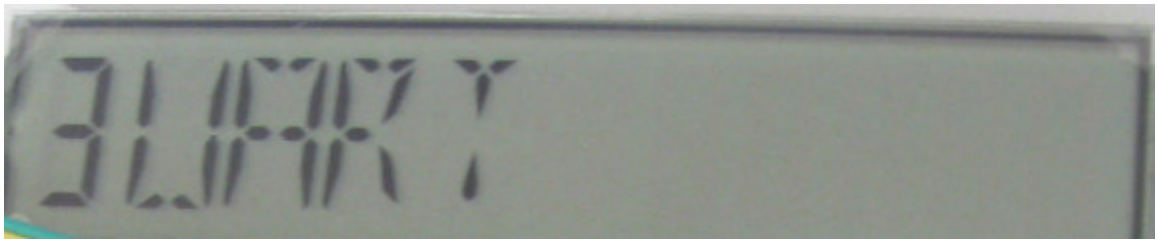
- ✓ Default is EEPROM mode



- ✓ Pressing **mode** key ---switch to temperature mode



- ✓ Pressing **mode** key ---switch to UART mode

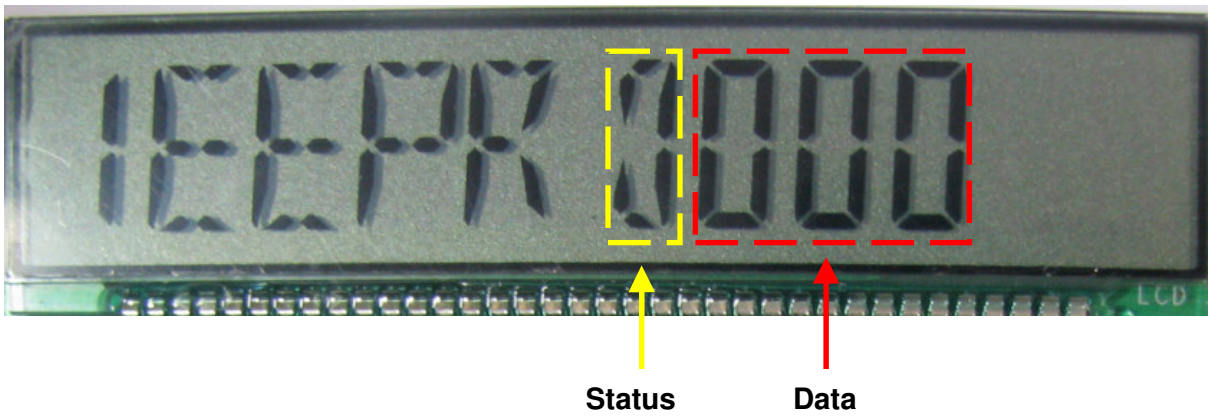


- ✓ Pressing **mode** key ---switch to RTC mod

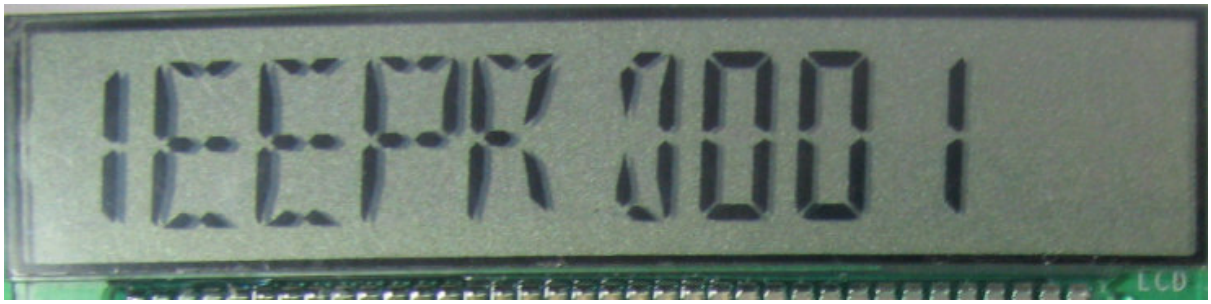


3.5.2 EEPROM Sub-address Switch and EDIT

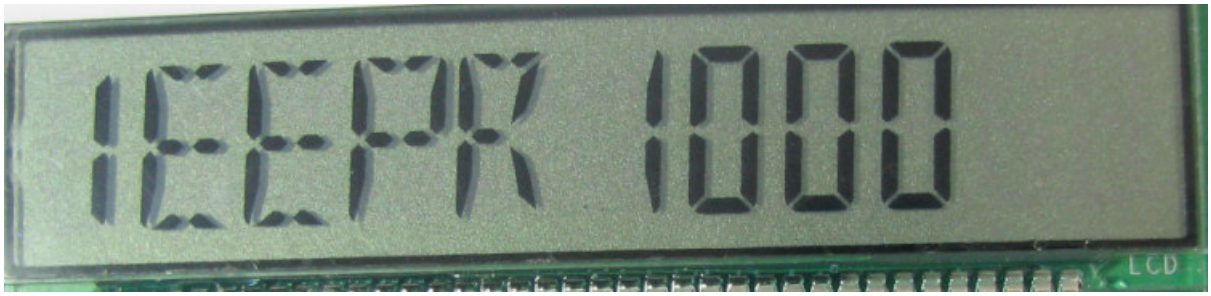
In EEPROM mode, the LCD displays the value of a certain sub-address



If you press up or down key, sub address and value will change.



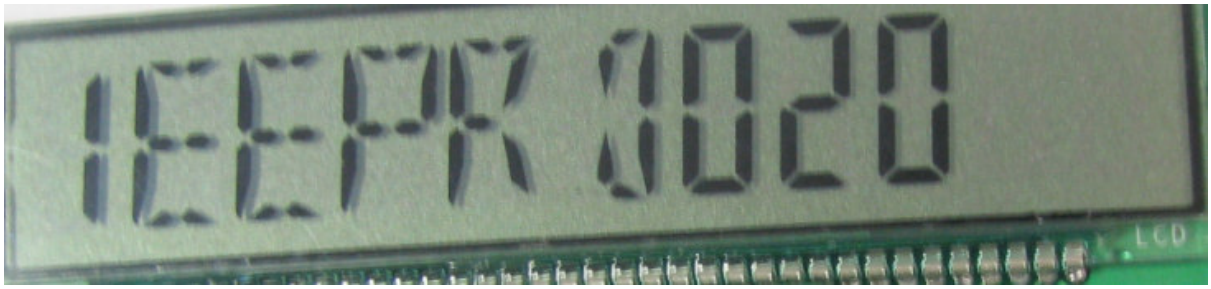
When you want to change the value of this sub address, please press **confirm** key



Pressing **left** or **right** (select LED to change) →**up** or **down** key to change the value.



Press **confirm** key to ensure this operation



Note:

Status “0” ------ normal status

Status “1” ------ setting status

3.5.3 Factory Mode

When in EEPROM normal status press “send” key and press “wake up” key again, system enter into factory mode. In factory mode the all EEPROM data are initialized to its’ sub address value and the LCD will display the value which be wrote into sub address.

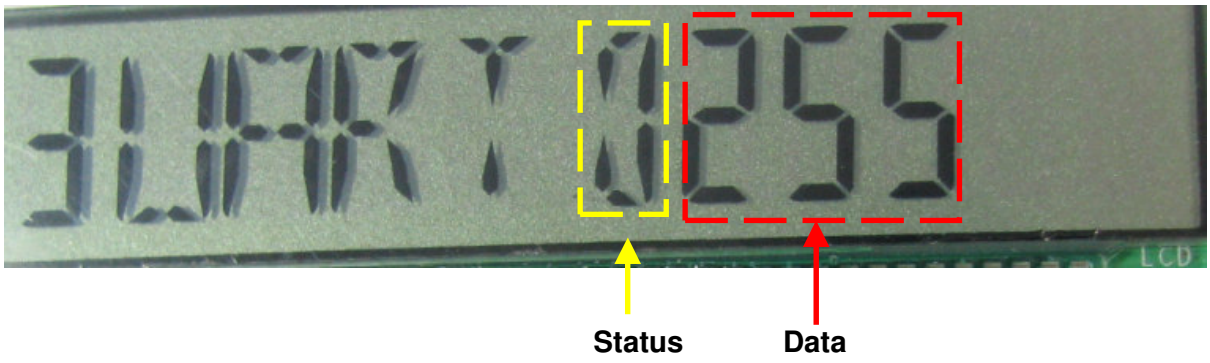
3.5.4 Temperature Operation

When it is changed to the environment temperature, the LCD will display the changed temperature value by 5℃ once.

3.5.5 UART Send

Switch the mode to UART, the LCD will display the value of received data; if you want to send data, the following operation will be an example.

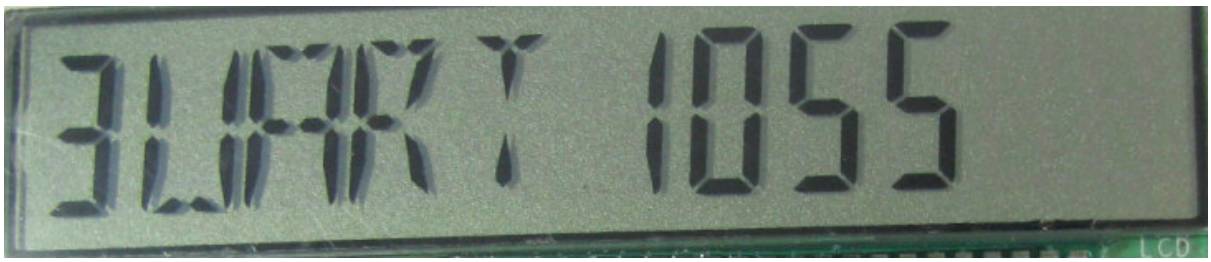
✓ Pressed **mode** key ---switch to UART mode



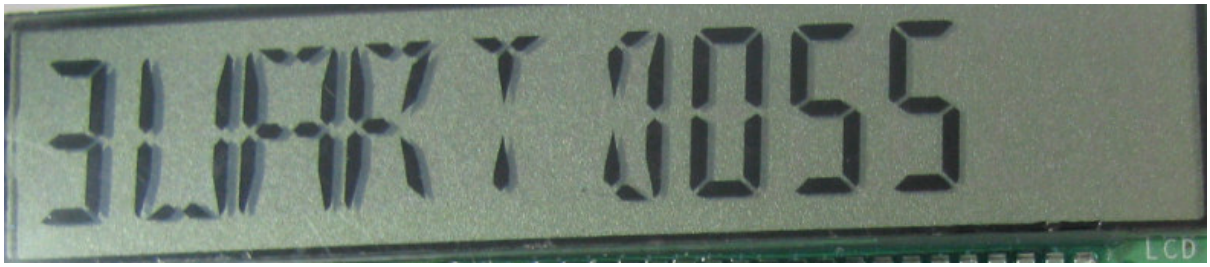
✓ press **confirm** key to enter EDIT



✓ Pressing **left** or **right** (select LED to change) →**up** or **down** key to change the value



✓ Press **send** key to send out and return receiving status



Note:

Status “0” ------ normal status

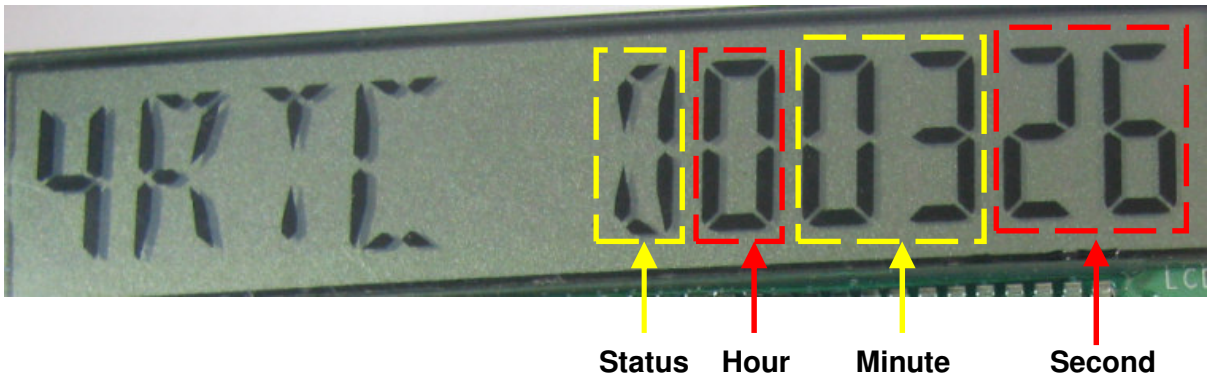
Status “1” ------ setting status

3.5.6 RTC Operation

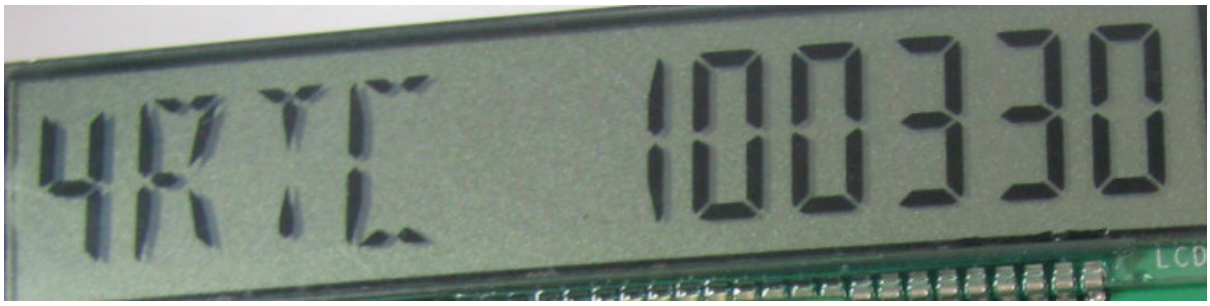
When system is powered on, the RTC timer begins to count. When user wants to enter standby mode, the sleep mode can be selected by setting the timer; after setting standby time, counter count continuously and enter sleep mode when the counter value is same with the set value.

Following operation describes the standby time setting.

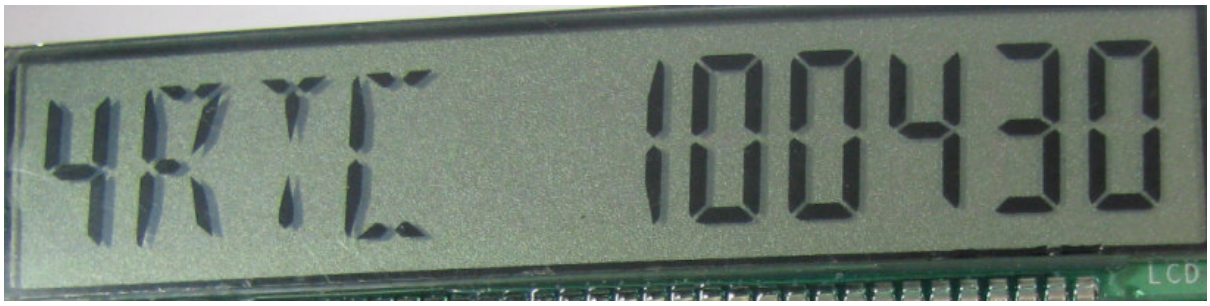
✓ Pressed **mode** key ---switch to RTC mode



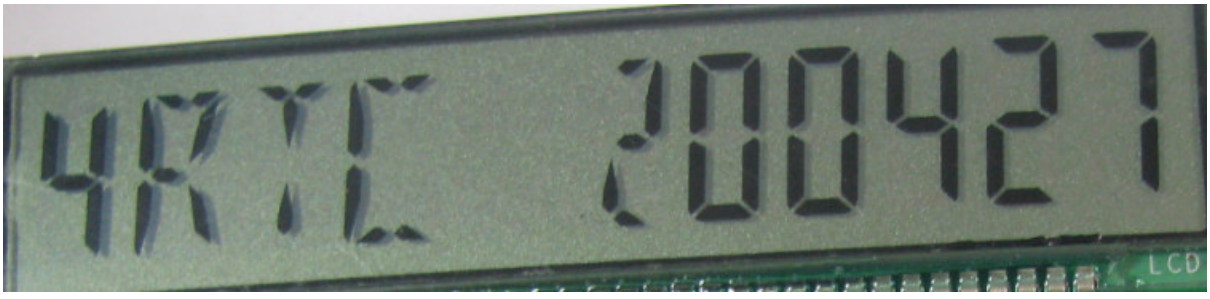
✓ press **confirm** key to enter EDIT status



✓ Pressing **left** or **right** (select LED to change) →**up** or **down** key to change the value, set standby time to 0hour 4minutes and 30seconds



✓ press **confirm** key to ensure this setting



Note:

1. When set time is 0hour 0minute and 0second that will release standby mode, system will not enter standby mode
2. The setting time must be bigger than the system running time.
3. The max time that enters standby mode is 9hours 59minutes and 59seconds.
4. Status “0” ------normal status

   Status “1”------ setting status

   Status “2”------ waiting to standby

3.5.7 Wake Up Operation

There is a wake up key in the LCD EV-Board, which uses external interrupt to wake up system from sleep mode.

When system enters sleep mode, no operation can be made except the wake-up key.

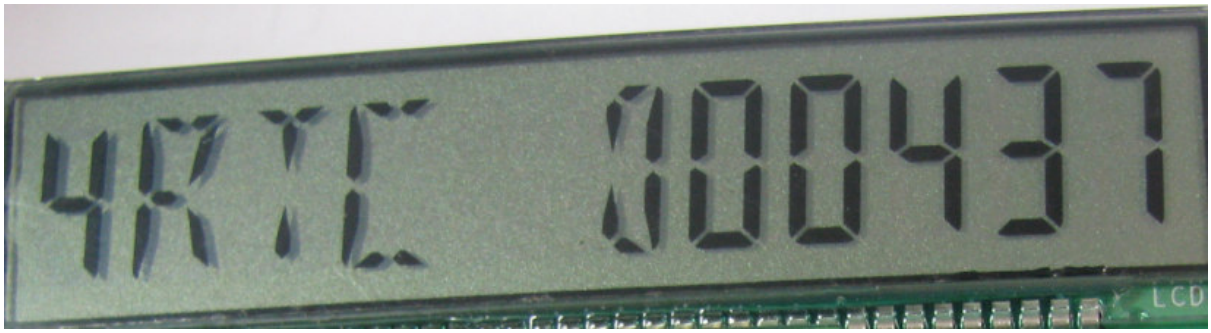
The following operation describes the wake-up process.

✓ When the timer is the same with standby setting time, system will enter sleep mode





✓ Press **wake up** key, system enters normal status; timer counts continuously.





3.5.8 Reset Operation

There is reset key to be used when user wants to reset system. In normal condition the reset pin is connected to V_{CC} when it is pressed to connect to GND by reset key.

3.5.9 Test Pin

For every pin of MB95F310 there is a test pin connecting to it.

3.5.10 Battery Usage

The supply power is 3.3V for MCU EEPROM and MAX232. The LCD EV-Board uses power chip AMS1117-3.3 to transfer input power from 5V to 3.3V, so the max battery supply is four cells (1.5V each cell).

4 Sample Code Manual

4.1 Project Structure

There is a project LCD EV-Board Project.prj in this MB95F310 EV-Board

The following structure describes the flow chart of this project, shown in Figure 4.1-1.

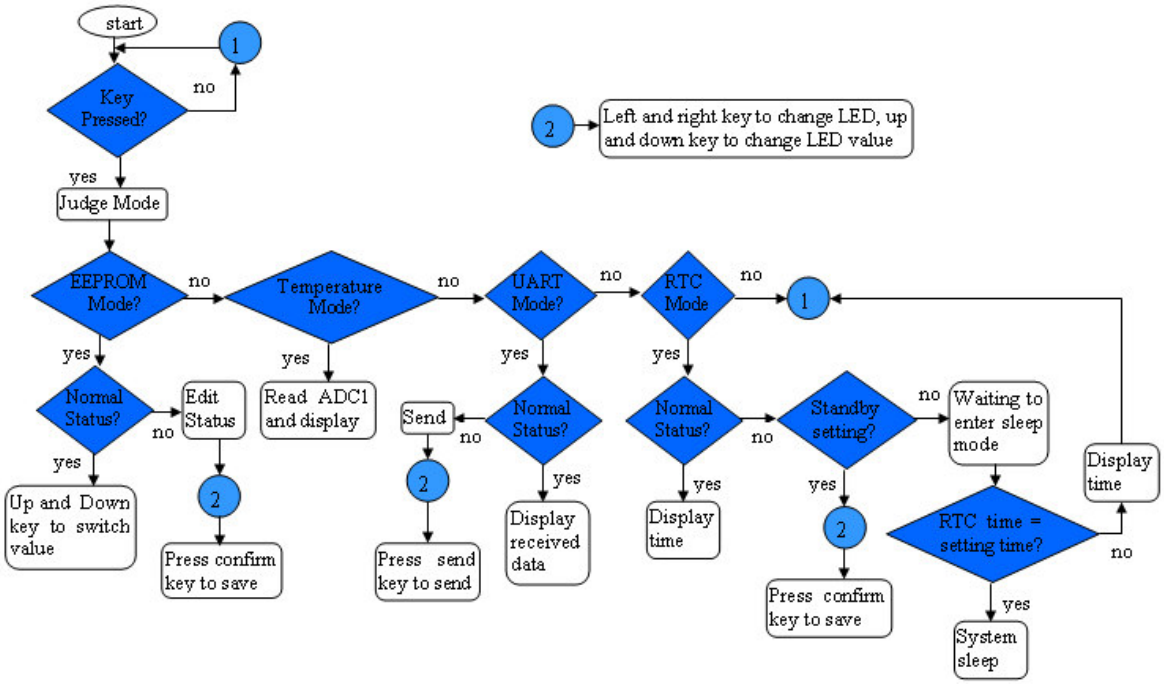


Figure 4.1 - 1 Project Structure

The following sample codes are provided with MB95F310 Series EV-Board,

4.2 Source Code File Description

Eight drive files are available in this project shown below,

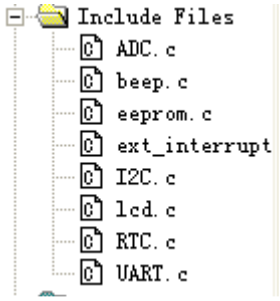


Figure 4.2 - 1 Source Code Files

4.2.1 ADC .c

In this function, environment temperature and key value will be calculated.

Following table describes the ADC library:

Function Name	Description
void ADC0_Init(void)	Initializes ADC function
void ADCTempSwitKey(void)	Detect keyboard value and temperature value

Following table describes how to use these functions.

Example		
Type	**Operation**	**Return**
Key Value Detect	ADC0_Init(); ADCTempSwitKey();	Global variable PraGlo.KEYFuncValue saved the key value
Temperature Detect		Global variable PraGlo.TempNum saved the temperature value

4.2.2 Beep .c

When AD key is pressed, it will sound beep and then it will be off. All the occasions are driven by setting different values for PIN PPG1.

Following table describes the Beep library:

Function Name	Description
void Beep_On(void)	Make beep to sound out
void Beep_Off(void)	Off beep sound

4.2.3 EEPROM .c

In this function, all values of EEPROM are initialled to 0x80 and global parameters are initialled to 0.

Following table describes the EEPROM library:

Function Name	Description
void Init_EEProm(void)	Initializes all global variable and EEPROM to its' value to sub-address number

4.2.4 Ext-interrupt .c

In this function external interrupt register are initialled.

Following table describes the external interrupt library:

Function Name	Description
void EXT_Init(void)	Initializes external interrupt
__interrupt void INTER_EXT (void)	Captures falling edge and generates interrupt

4.2.5 I2C .c

In this function, we should initialize I2C status and use SCL and SDA to send data to master device and receive data from slave device.

Following table describes the I2C library:

Function Name	Description
unsigned char RD_I2C(unsigned char Main_Addr, unsigned char Sub_Addr)	Read data from Sub_Addr
void Write_I2C_Proc(unsigned char MainAddr, unsigned char SubAddr, unsigned char I2Cdata)	Write data I2Cdata to SubAddr

Following table describes how to use these functions.

Example		
Type	**Operation**	**Return**
Read data	RD_I2C(0xa0,0x01);	Read data from EEPROM sub address 0x01
Write data to EEPROM	Write_I2C_Proc(0xa0,0x01,0x25);	Write 0x25 to EEPROM sub address 0x01

4.2.6 LCD .c

In this function, we should initialize LCD register and drive LCD by sending different data to SEG00…SEG31

Following table describes the LCD library:

Function Name	Description
void Init_LCD(void)	Initializes LCD module
void LCD_Clear(void)	Clear LCD display
void LCD_LigON_NUM(unsigned char Num, unsigned char Dat)	Drive LCD Num to display number Dat

Following table describes how to use these functions.

Example		
Type	**Operation**	**Return**
Display LCD	LCD_LigON_NUM(0x02,0x06);	Drive LED 2 to display number 6

4.2.7 RTC .c

In this function, we should initialize clock register, start time base timer, set display value for LCD and set condition to enter standby mode

Following table describes the LCD library:

Function Name	Description
void TBT_Init(void)	Initializes time base timer
__interrupt void TBT_Inter (void)	Time base timer interrupt for 0.35ms arrived
void STB_Mode_Set(void)	Set standby time

Following table describes how to use these functions.

Example		
Type	**Operation**	**Return**
Enable RTC	TBT_Init();	Sets interval time
Set standby time	TBT_Init(); PraBli.Bli_6 = 0x01; PraBli.Bli_7 = 0x00; PraBli.Bli_8 = 0x00; PraBli.Bli_9 = 0x02; PraBli.Bli_a = 0x06; STB_Mode_Set();	PraBli.Bli_6 is hour, PraBli.Bli_7 and PraBli.Bli_8 are minute, PraBli.Bli_9 and PraBli.Bli_a are second. In this example setting after 1hour 26seconds the system will standby
Standby and Wake up	EXT_Init(); TBT_Init(); PraBli.Bli_6 = 0x01; PraBli.Bli_7 = 0x00; PraBli.Bli_8 = 0x00; PraBli.Bli_9 = 0x02; PraBli.Bli_a = 0x06; STB_Mode_Set();	Press “wakeup” key system wake up

4.2.8 UART .c

In this function, the MCU works in an asynchronous mode, baud rate is generated by dedicated baud rate generator.

Following table describes the LCD library:

Function Name	Description
void UART_init (void)	Initializes UART module
void UART_sendbyte (char ch)	Send data ch by UART
__interrupt void UART_REC_INTER(void)	Received data by UART when data is arrived

Following table describes how to use these functions.

Example		
Type	**Operation**	**Return**
Send data	UART_init(); UART_sendbyte(0x55);	Send out data 0x55 by UART
Receive data	UART_init();	Global variable PraGlo.Uart_Rec saved the received data

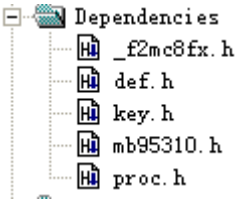
4.2.9 delay_try Function

In this project some instruction need to be delayed, so the delay function must be defined before using these C file. Following code is the delay_try function.

```
void delay_try(unsigned char Dat)
{
        unsigned char i,j;
        for(i=0;i<255;i++)
        {
                for(j=0;j<Dat;j++)
                {
                        __wait_nop();
                }
        }
}
```

4.3 Global Variable Description

Five drive files are available in this project shown below,



- ✓ _f2mc8fx.h

  _f2mc8fx.h is system define document. Ignore it

- ✓ mb95310.h

  Mb95310.h is system define document. Ignore it

- ✓ def.h

  This file includes roughly all global variable of FW

  Following table describes the main struct

Struct Name	Description
Global	includes global variable used between c files
RTCPar	includes global variable used in time base timer and standby
EepPra	Includes global variable used in I2C module
BlinkPra	Includes global variable used in six LEDs

- ✓ Key.h

  This file includes eight key names, which are “UP” “DOWN” “RIGHT” “LEFT” “SEND” “CONFIRM” “MODE” and “USER”.

- ✓ Proc.h

  This file includes all functions which will be used in other functions.

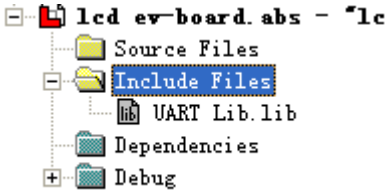
4.4 How to Add these Library

Before using chapter4.2 library, following step ought to be referenced.

- Create a new project
- Copy .lib file to project document



- Add .lib file to project



- Add all needed .lib files to project

4.5 Usage Demo

Following table is an example for how to use this C file. Process_Control() is a function user defined which transfer C file to come true user's idea when to use ADC when to use I2C when to use RTC.

```
ADC0_Init();                    // initialise ADC
UART_init();                    // initialise UART
EXT_Init();                     // initialise Interrupt
Init_LCD();                     // initialise LCD
TBT_Init();                     // initialise Timebase timer
Init_EEProm();                  // initialise EEPROM
while(1)
{
        ADCTempSwitKey();       // Judge key value
        Process_Control();      // transfer every library
        LCD_LigON_NUM(0,2);     // display LCD
}
```

5 Development Platform Quick Start

5.1 Tools Setup Sequence

Start the debugging system in the following sequence:

- ✓ Connect a BGMA to the PC with a USB cable, confirm the LED on the BGMA is Green;
- ✓ Connect an EV-board to BGMA IDC10 socket;
- ✓ Select the EV-board power supply and turn on the EV-board, confirm the LED on the BGMA is Orange and the Power LED on the EV-board is on.

5.2 Open Project and Start Debug

Users can start a debug from a PC software SOFTUNE workbench in the following sequence. Here take SIMULATE LCD EVBOARD project for example.

- ✓ Start the SOFTUNE from “Startup Menu>Programs> SOFTUNE V3> FFMC-8L Family SOFTUNE Workbench” in Windows;
- ✓ Click “Open workspace” from “File” Menu in SOFTUNE;
- ✓ Select “SIMULATE LCD EVBOARD.wsp” in “Open Space” window;
- ✓ Click “Start debug” from “Debug” Menu.

If the entire procedure goes right, a debug will start normally.

5.3 Operation Precautions

- ✓ All pins of MB95F310 Series MCU are connected to Testing Pin on the EV-board. If user wants to test separate pin performance, please connect to the test pin.
- ✓ Note: Before connecting with BGMA, do not power on system.
- ✓ It’s recommended that only one power supply is used as a power module input at a time.

MCU-UM-95F310-E-10

FUJITSU MICROELECTRONICS LIMITED • SUPPORT SYSTEM

F2MC-8FX Family MB95F310 Series
EV-Board
MB2146-450-E
SETUP GUIDE

Aug. 2009 the second edition

Published FUJITSU MICROELECTRONICS LIMITED

Edited Strategic Business Development Dept


FUJITSU