MS51 Series BSP User Guide

Based on IAR Embedded Workbench 8051 V10.10.1
For NuMicro® 8051 Family

Directory Information

Please extract the "MS51_Series_BSP_IAR_V1.00.003.zip" file firstly and confirm the following content of this BSP folder.

MS51DA9AE_MS51BA9AE_BSP_IAR	BSP for MS51DA9AE and MS51BA9AE. 8KB Flash APROM share with 4KB LDROM 256 Byte RAM, 1024 Byte XRAM In TSSOP14 and MSOP10 package
MS51FB9AE_MS51XB9AE_MS51XB9BE_BSP_IAR	BSP for MS51FB9AE, MS51XB9AE and MS51XB9BE. 16KB Flash APROM share with 4KB LDROM 256 Byte RAM, 1024 Byte XRAM In TSSOP20 and QFN20 package
MS51FC0AE_MS51XC0BE_MS51EC0AE_MS51TC0AE_MS51PC0AE	BSP for MS51FC0AE, MS51XC0BE, MS51EC0AE, MS51TC0AE and MS51PC0AE. 32KB Flash APROM share with 4KB LDROM 256 Byte RAM, 2048 Byte XRAM In TSSOP20, QFN20, TSSOP28, QFN33 and LQFP32 package

Each folder listed above with following content folders

Document\	Driver reference manual and revision history.
Library\	Device driver header and source files.
SampleCode\	Driver sample code.

The information described in this document is the exclusive intellectual property of Nuvoton Technology Corporation and shall not be reproduced without permission from Nuvoton.

Nuvoton is providing this document only for reference purposes of NuMicro microcontroller based system design. Nuvoton assumes no responsibility for errors or omissions.

All data and specifications are subject to change without notice.

For additional information or questions, please contact: Nuvoton Technology Corporation.

www.nuvoton.com

1 .\Document\

Nuvoton_MS51_8K_Series_BSP_IAR_Revision_History.pdf	This document shows the revision history of MS51 BSP for IAR Embedded Workbench for 8051.

2 .\Library\

Device\	MS51 Device header file
Startup\	A51 startup file and executable file
StdDriver\	All peripheral driver header and source files.

3 .\SampleCode\

StdDriver	Demonstrate the usage of MS51 series MCU peripheral driver.
Template	A project template for MS51 series MCU.

4 .\SampleCode\RegBased

ADC_Bandgap_VDD	Calculate the real V$_{DD}$ value of the device system based on the difference between the pre-stored ADC conversion result values when V$_{DD}$ is 3.072V and the system converted band-gap value.
ADC_GPIO _Trig	Demonstrate how to use GPIO to start ADC initial setting and show the conversion result in ADCRH and ADCRL register.
ADC_Multi_channel	Demonstrate how to regularly sample from different ADC input channel.
ADC_PWM_Trig	Demonstrate how to use each of PWM timer period timeout to trigger ADC conversion.
ADC_Simple	Start ADC conversion by triggering ADCS bit [ADCCON0.6] and check the flag register ADCF bit [ADCCON0.7] to confirm if a conversion is finished.
Fsys_ModifyHIRC	Call the library fie “sys.c” to modify system setting as 16 MHz or 24 MHz and check clock out pin to confirm if Fsys is modified. The MS51 HIRC can be selected within 16 MHz or 24 MHz.
Fsys_Select_ECLK	Change the MS51 system clock from HIRC to external clock initial setting. The MS51 external clock input ranges from 4 MHz to 24 MHz.
GPIO_ClockOut	Show the MS51 system clock and output from CLKO pin.
GPIO_Input_Output	Toggle each MS51 GPIO pin output from high to low after 200ms delay.
I2C_EEPROM	Show how to use MS51 as master to read external connect EEPROM by I^{2}C bus.
IAP_AP_program_AP_Dataflash	Demonstrate how MS51 APROM is used as Data Flash to implement erase / program / read verify function. All APROM memory can be used as Data Flash.
IAP_AP_program_LD	Demonstrate how MS51 IAP runs in APROM to program LDROM and implement erase / program / read verify function. User first needs to confirm if the LDROM is enabled through CONFIG setting.
IAP_Dataflash_EEPROM_	Simulate Data Flash as EEPROM mode by calling the library file “eeprom.c”. This process includes copy one page of

Mode	Data Flash values in RAM, modify data, erase Data Flash, then copy new values from RAM to Data Flash.
IAP_LD_Program_AP	Demonstrate how MS51 IAP runs in LDROM to program APROM and implement erase / program / read verify function.
IAP_program_Config	Demonstrate using MS51 IAP command to modify CONFIG bytes.
IAP_Read_UCID	Demonstrate using MS51 IAP command to read the unique customer ID (UCID). Only for customer special order MS51 MCU. One UCID is only for one customer.
IAP_Read_UID	Demonstrate using MS51 IAP command to read the Unique code of MS51. The UID value of each MS51 is different.
INT0_ExtInt	Perform MS51 external interrupt pin INT0 enabled initial setting.
INT1_ExtInt	Perform MS51 external interrupt pin INT1 enabled initial setting.
Interrupt_ISR_all	List all interrupt sector in the library file "isr.c". Call this file to implement interrupt subroutine.
Pin_Interrupt	Demonstrate how to wake up MS51 from Idle / Power-down mode through external interrupt input by enabling MS51 pin interrupt function.
PWM0_DeadTime	Configure PWM as Complementary mode. Control 3 pairs output, set each 2 channel PWM output as same duty and insert dead time.
PWM0_Independent	Configure PWM as independent mode. Each PWM channel outputs independently and each PWM channels output with different duty and interrupt enabled.
PWM0_Synchronous	Configure PWM as Synchronous mode. Each PWM0 channel 0/2/4 outputs different duty and PWM0 channel 1/3/5 duty following 0/2/4 setting.
PWM0123_GroupStart	Configure PWM0/1/2/3 as independent mode and start with same time.
PWM123_Independent	Configure PWM0/1/2/3 as independent mode each channel duty is different.
SPI_Flash	Connect MS51 with W25Q16BV SPI Flash and set it as master to read and write data sample code.

Timer0_mode_0_Interrupt	Configure Timer 0 as mode 0,13-bit timer with interrupt enabled initial setting.
Timer0_mode_1_Interrupt	Configure Timer 0 as mode 1, 16-bit timer with interrupt enabled initial setting.
Timer0_mode_2_Interrupt	Configure Timer 0 as mode 2, 8-bit timer with auto reload function and interrupt enabled initial setting.
Timer01_mode_3_Interrupt	Configure Timer 0 mode 3 as two separated timer initial settings with interrupt enabled. Timer0 counter value register high byte TH0 overflow will set the register flag bit TF1 as 1 (Timer1 overflow flag).
Timer1_mode_0_Interrupt	Configure Timer 1 as mode 0,13-bit timer with interrupt enabled initial setting.
Timer1_mode_1_Interrupt	Configure Timer 1 as mode 1, 16-bit timer with interrupt enabled initial setting.
Timer1_mode_2_Interrupt	Configure Timer 1 as mode 2, 8-bit timer with auto reload function and interrupt enabled initial setting.
Timer2_AutoReload_Capture	Configure Timer 2 as one channel input capture with interrupt enabled initial setting. Timer 2 capture interrupt vector is different to the Timer 2 overflow interrupt.
Timer2_AutoReload_Delay	Configure Timer 2 as auto reload delay setting with interrupt enabled initial setting.
Timer3	Configure Timer 3 as auto reload mode initial setting and interrupt enabled.
UART0	Loop transmit from UART0 TXD pin initial setting . To confirm if each byte transfer is completed by checking UART0_TXD transfer flag bit SI set as 1.
UART0_Interrupt_RW	Configure UART0 transfer including transmit and receive with interrupt enabled.
UART1	Configure UART1 transmit and receive initial setting and enable interrupt subroutine.
UART2	Configure SC0 as UART2 transfer including transmit and receive with interrupt enabled.
UART3	Configure SC1 as UART3 transfer including transmit and

	receive with interrupt enabled.
UART4	Configure SC2 as UART4 transfer including transmit and receive with interrupt enabled.
WakeupTimer_INT	Enable wake-up timer with interrupt function. Main loop enters Power-down mode after initial setting, and once WKT timeout, MS51 will wake up and then jump into interrupt subroutine to toggle GPIO output.
Watchdog_INT	Demonstrate Watchdog Timer (WDT) initial setting with interrupt enabled and Watchdog Timer reset function disabled. The WDT counter overflow will jump into WDT interrupt subroutine.
Watchdog_Reset	Demonstrate Watchdog Timer reset initial setting and include the library file “Watchdog.c” to call Enable / Disable WDT reset subroutine. Set P0.0 as high level to disable WDT reset function; otherwise, keep WDT reset enabled.

Based on the features of the different products these projects is not necessarily included in folder ..\SampleCode\RegBased .

1 REVISION HISTORY

Date	Revision	Description
2019.9.27	1.00	Initial release.
2019.6.27	1.00.002	1. Added MS51DA9AE_MS51BA9AE_BSP_Keil folder 2. Modified MS51_16K_BSP_Keil_V1.00 folder name to S51FB9AE_MS51XB9AE_MS51XB9BE
2019.11.27	1.00.003	Added MS51FC0AE_MS51XC0BE_MS51EC0AE_ MS51TC0AE_MS51PC0AE

Important Notice

Nuvoton Products are neither intended nor warranted for usage in systems or equipment, any malfunction or failure of which may cause loss of human life, bodily injury or severe property damage. Such applications are deemed, "Insecure Usage".

Insecure usage includes, but is not limited to: equipment for surgical implementation, atomic energy control instruments, airplane or spaceship instruments, the control or operation of dynamic, brake or safety systems designed for vehicular use, traffic signal instruments, all types of safety devices, and other applications intended to support or sustain life.

All Insecure Usage shall be made at customer's risk, and in the event that third parties lay claims to Nuvoton as a result of customer's Insecure Usage, customer shall indemnify the damages and liabilities thus incurred by Nuvoton.

Please note that all data and specifications are subject to change without notice.
All the trademarks of products and companies mentioned in this datasheet belong to their respective owners.