

## MS51 Series BSP User Guide

Based on Keil uVision4 and PK51 Development Kit V9.52

For NuMicro® 8051 Family

### Directory Information

Please extract the “MS51\_Series\_BSP\_Keil\_V1.00.003.zip” file firstly and confirm the following content of this BSP folder.

<b>MS51DA9AE_MS51BA9AE_MS51IA9AE</b>	BSP for MS51DA9AE and MS51BA9AE. 8KB Flash APROM share with 4KB LDROM 256 Byte RAM, 1024 Byte XRAM In TSSOP14 , MSOP10 and SOP8 package
<b>MS51FB9AE_MS51XB9AE_MS51XB9BE</b>	BSP for MS51FB9AE, MS51XB9AE and MS51XB9BE. 16KB Flash APROM share with 4KB LDROM 256 Byte RAM, 1024 Byte XRAM In TSSOP20 and QFN20 package
<b>MS51FC0AE_MS51XC0BE_MS51EC0AE_MS51TC0AE_MS51PC0AE</b>	BSP for MS51FC0AE, MS51XC0BE, MS51EC0AE, MS51TC0AE and MS51PC0AE. 32KB Flash APROM share with 4KB LDROM 256 Byte RAM, 2048 Byte XRAM In QFN33, LQFP32, TSSOP28, TSSOP20 and QFN20 package

Each folder listed above with following content folders

<b>Document\</b>	Driver reference manual and revision history.
<b>Library\</b>	Device driver header and source files.
<b>SampleCode\</b>	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](http://www.nuvoton.com)

## 1 .\Document\

**MS51DA9AE\_MS51BA9AE\_Series\_BSP\_Keil\_Revision\_History.pdf**

This document shows the revision history of MS51DA9AE\_MS51BA9AE\_BSP\_Keil.

OR

**MS51FB9AE\_MS51XB9AE\_MS51XB9BE\_BSP\_Keil\_Revision\_History.pdf**

This document shows the revision history of MS51FB9AE\_MS51XB9AE\_MS51XB9BE\_BSP\_Keil.

OR

**MS51FC0AE\_MS51XC0BE\_MS51EC0AE\_MS51TC0AE\_MS51PC0AE\_BSP\_Keil\_Revision.pdf**

This document shows the revision history of MS51FC0AE\_MS51XC0BE\_MS51EC0AE\_MS51TC0AE\_MS51PC0AE\_BSP\_Keil.

## 2 .\Library\

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

### 3 .\SampleCode\

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

## 4 .\SampleCode\RegBased

<b>ADC_Bandgap</b>	Configure band-gap as ADC input channel and show ADC conversion result in ADCRH and ADCRL register.
<b>ADC_Bandgap_VDD</b>	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.
<b>ADC_Continuous</b>	Configure ADC as continuous mode. Define convert result storage in XRAM area. When ADC interrupt means continuous sample finished print all the result from UART0 TXD.
<b>ADC_GPIO_Trig</b>	Demonstrate how to use GPIO to start ADC initial setting and show the conversion result in ADCRH and ADCRL register.
<b>ADC_Multi_channel</b>	Demonstrate how to regularly sample from different ADC input channel.
<b>ADC_PWM_Trig</b>	Demonstrate how to use each of PWM timer period timeout to trigger ADC conversion.
<b>ADC_Simple</b>	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.
<b>Fsys_ModifyHIRC</b>	Call the library file "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.
<b>Fsys_Select_ECLK</b>	Change the MS51 initial setting of system clock from HIRC to the external clock input. The MS51 external clock input ranges from 4 MHz to 24 MHz.
<b>GPIO_ClockOut</b>	Show the MS51 system clock and output from CLKO pin.
<b>GPIO_Input_Output</b>	Toggle each MS51 GPIO pin output from high to low after 200ms delay.
<b>GPIO_PowerDown_BODdi sable</b>	Show how to disable MS51 Burn-out detect function and into power down mode. This project is special for measure the

	power down current of MS51.
<b>I2C_Read_EEPROM</b>	Show how to use MS51 as master to read external connect EEPROM by I <sup>2</sup> C bus.
<b>IAP_AP_program_AP_Dataflash</b>	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.
<b>IAP_AP_program_LD</b>	Demonstrate how MS51 IAP runs in APROM to program LDROM and implements erase / program / read verify function. Firstly, user needs to confirm if the LDROM is enabled through CONFIG setting.
<b>IAP_Dataflash_EEPROM_Mode</b>	Simulate Data Flash as EEPROM mode by calling the library file "eeprom.c". This process includes read data and storage in RAM / modify value / erase Data Flash / copy new value from RAM to Data Flash.
<b>IAP_LD_Program_AP</b>	Demonstrate how MS51 IAP runs in LDROM to program APROM and implements erase / program / read verify function.
<b>IAP_program_Config</b>	Demonstrate using MS51 IAP command to modify CONFIG bytes.
<b>IAP_Read_UCID</b>	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.
<b>IAP_Read_UID</b>	Demonstrate using MS51 IAP command to read the Unique code of MS51. The UID value of each MS51 is different.
<b>INT0_Ext_Interrupt</b>	Perform MS51 external interrupt pin INT0 enabled initial setting.
<b>INT1_Ext_Interrupt</b>	Perform MS51 external interrupt pin INT1 enabled initial setting.
<b>ISP_UART0</b>	Standard ISP protocol process boot loader source code. Communication with UART0 for NuMicro ISP Programming Tool.
<b>ISP_UART1</b>	Standard ISP protocol process boot loader source code. Communication with UART1 for NuMicro ISP Programming Tool.
<b>Pin_Interrupt</b>	Demonstrate how to wake up MS51 from Idle / Power-down mode through external interrupt input by enabling MS51 pin interrupt function.
<b>PWM0_DeadTime</b>	Configure PWM as Complementary mode. Control 3 pairs output, set each 2 channel PWM output as same duty and insert dead time.

<b>PWM0_Independent</b>	Configure PWM as independent mode. Each PWM channel outputs independently and each PWM channels output with different duty and interrupt enabled.
<b>PWM0_Synchronous</b>	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.
<b>PWM0123_GroupStart</b>	Configure PWM0/1/2/3 as independent mode and start with same time.
<b>PWM123_Independent</b>	Configure PWM0/1/2/3 as independent mode each channel duty is different.
<b>PWM_INT</b>	Configure PWM one channel output with falling edge trig interrupt enable initial setting.
<b>PWM_Simple</b>	Configure PWM as Independent mode. Only set one channel output.
<b>PWM_Synchronous</b>	Configure PWM as Synchronous mode. Set PWM Channel 0/2/4 output with independent duty value. PWM1/3/5 outputs follow PMW 0/2/4 setting.
<b>SPI_Flash</b>	Connect MS51 with W25Q16BV SPI Flash and set it as master to read and write data sample code.
<b>Timer0_mode_0_Interrupt</b>	Configure Timer 0 as mode 0,13-bit timer with interrupt enabled initial setting.
<b>Timer0_mode_1_Interrupt</b>	Configure Timer 0 as mode 1, 16-bit timer with interrupt enabled initial define.
<b>Timer0_mode_2_Interrupt</b>	Configure Timer 0 as mode 2, 8-bit timer with auto reload function and interrupt enabled initial setting.
<b>Timer01_mode_3_Interrupt</b>	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).
<b>Timer1_mode_0_Interrupt</b>	Configure Timer 1 as mode 0,13-bit timer with interrupt enabled initial setting.
<b>Timer1_mode_1_Interrupt</b>	Configure Timer 1 as mode 1, 16-bit timer with interrupt enabled initial setting.
<b>Timer1_mode_2_Interrupt</b>	Configure Timer 1 as mode 2, 8-bit timer with auto reload



	function and interrupt enabled initial setting.
<b>Timer2_AutoReload_Capture</b>	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.
<b>Timer2_AutoReload_Delay</b>	Configure Timer 2 as auto reload delay setting with interrupt enabled initial setting.
<b>Timer3</b>	Configure Timer 3 as auto reload mode initial setting and interrupt enabled.
<b>UART0_Interrupt_RW</b>	Configure UART0 transfer including transmit and receive with interrupt enabled.
<b>UART0_Printf</b>	Demonstrate printf instruction based on transmit from UART0 TXD pin.
<b>UART1</b>	Configure UART1 transmit and receive initial setting and enable interrupt subroutine.
<b>UART1_Printf</b>	Demonstrate printf instruction based on transmit from UART1 TXD pin. Include how to modify putchar.c file.
<b>UART2</b>	Configure SC0 as UART2 transfer including transmit and receive with interrupt enabled.
<b>UART3</b>	Configure SC1 as UART3 transfer including transmit and receive with interrupt enabled.
<b>UART4</b>	Configure SC2 as UART4 transfer including transmit and receive with interrupt enabled.
<b>WakeupTimer_Interrupt</b>	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.
<b>Watchdog_Interrupt</b>	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.
<b>Watchdog_Reset</b>	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 product these project is not necessarily included in folder of .\SampleCode\RegBased.

## 5 REVISION HISTORY

Date	Revision	Description
2019.1.29	1.00.001	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 MS51FB9AE_MS51XB9AE_MS51XB9BE_BSP_Keil
2019.11.27	1.00.003	Added MS51FC0AE_MS51XC0BE_MS51EC0AE_MS51TC0AE_MS51PC0AE_BSP_Keil

### 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.*