MS51 16K Series BSP Guide

Directory Introduction for NuMicro™ 8051 Family

Directory Information

Please extract the "MS51_16K_BSP_Keil_C51_V1.00.zip" file firstly, and confirm the following folder all contain.

This BSP folder contents:

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

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

www.nuvoton.com

1 .\Document\

Nuvoton_MS51_16K_BSP_Revision_History.pdf	This document shows the revision history of MS51 BSP.

2 .\Library\

Device	Compliant device header file
Startup	A51 startup file and executable file
StdDriver	All peripheral driver header and source files.

3 .\ Device \

Function_Define.h	Special define file include the function peripheral initial setting define.
MS51.h	MS51 header file with keil C51format.
SFR_Macro.h	SFR bit register define with set/clr to make sure can bit to define SFR.

4 .\ Startup \

EXE	hex2bin.exe file included for project call hex to create bin file execute tool.
Keil	The Keil startup.A51file special for MS51 16K body

5 .\SampleCode\

RegBased	Register base the usage of MS51 series MCU peripheral code.
Template	A project template for MS51 series MCU

6 .\SampleCode\RegBased

ACMP_Bandgap	ADC input pin is bandgap demo.
ACMP_Bandgap_VDD	ADC converts demo code with band-gap value to calculate the VDD value.
ADC_GPIO _Trig	ADC trig start by GPIO demo. External GPIO level change will trig ADC convert start.
ADC_multi_channel	ADC input from ADC_CH4 pin and Bandgap input routine to convert demo.
ADC_PWM_Trig	ADC convert start trig by PWM period demo. Each PWM period over will trig ADC by hardware.
ADC_Simple	ADC convert start by trig SFR bit and convert finish by check flag.
Fsys_ModifyHIRC	Modify internal HIRC value to 16MHz or 24MHz demo. This modify value is from internal defined, after loading, HIRC deviation not over 1%.
Fsys_Select_ECLK	MS51 system clock select from HIRC to External clock demo.
GPIO_ClockOut	MS51 system clock output from GPIO demo. System clock output 100KHz after divider. Since HIRC maybe define to 16MHz or 24MHz, based on HIRC real status to define divider value.
GPIO_InputOutput	MS51 gpio simple toggle out demo. This demo is easy to confirm MS51 normal run status.
GPIO_PowerDown_BODdisable	MS51 power down mode demo. For confirm power down current of each MCU. Call BOD disable function first to confirm down to lowest power consumption.
I2C0_EEPROM	I^2C setting as master mode to connect W24LC64 EEPROM to write in and read out data demo.
IAP_APROM_program_DataFlash	MS51 Data flash demo code. all APROM memory can be used as Dataflash.
IAP_APROM_program_LD	IAP run in APROM to program LDROM. First need confirm

ROM	the LDROM is enabled.
IAP_Dataflash_EEPROM	Customer use this macro, each time call this subroutine, can use Data flash as EEPROM mode, the process include read old data / erase / modify new code/ write in.
IAP_LD-Program-AP	IAP run in LDROM to program APROM. This function is use in ISP function.
IAP_program_Config	Use code IAP function to modify CONFIG area.
IAP_Read_UCID	Use IAP command to read the UCID of each MS51. Only for customer special order MS51 MCU. One UCID is only for one customer.
IAP_Read_UID	Use IAP command to read the UID of each MS51. Each pieces of MS51 UID is different.
INT0_Ext_ Interrupt	External INT0 function demo. The basic 8501 EXT0 function.
INT1_Ext_ Interrupt	External INT1 function demo. The basic 8051 EXT1 function.
Pin_Interrupt	One group each GPIO of MS51 use as external interrupt pin. Trig IC wakeup from idle / power down mode.
PWM_DeadTime	PWM output insert with dead time setting demo, include picture to show after insert dead time waveform.
PWM_Interrupt	PWM with interrupt subroutine, after period PWM into interrupt process.
PWM_Simple	PWM simple define output from individual PWM channel.
ROM_Const_One	Combine data table into ROM define special address, picture show the defined item in options. After compiler the output hex or bin file with this table.
ROM_Const_One	Two table file combine special define item in options demo.
Timer0_mode_0_Interrupt	Timer0 mode 0 13-bit timer counter define with interrupt demo. One shot no auto reload function.
Timer0_mode_1_Interrupt	Timer0 mode 1 16-bit timer counter define with interrupt demo. One shot no auto reload function.

Timer0_mode_2_Interrupt	Timer0 mode 2 8-bit auto reload timer counter define with interrupt demo.
Timer01_mode_3_Interrupt	Mode 3 is two separate 8 bit, TR0 and TR1 to control begin the Timer counter.
Timer1_mode_0_Interrupt	Timer1 mode 0 13-bit timer counter define with interrupt demo. One shot no auto reload function.
Timer1_mode_1_Interrupt	Timer1 mode 1 16-bit timer counter define with interrupt demo. One shot no auto reload function.
Timer1_mode_2_Interrupt	Timer1 mode 2 8-bit auto reload timer counter define with interrupt demo.
Timer2_AutoReload_Capture	Timer 2 use as capture initial setting. With 7 channel 3 input capture module setting.
Timer2_AutoReload_Delay	Timer 2 delay counter define with auto reload function.
Timer3	Timer 3 delay counter define with auto reload function. When use timer 3 need confirm this time not used as UART baud rate generator.
UART0	UART0 demo code. Include transmit and receive demo, since default HIRC16MHz not fit for baud rate over 38400, modify HIRC to 24MHz to confirm the baud rate deviation not over 1%.
UART0_Interrupt_RW	UART0 with interrupt, received data from UART0_RXD transmit to UART0_TXD loop.
UART0_printf	Code use “printf” instruction from UART0 setting. Special with “putchar.c” define.
UART1	UART1 demo code Include transmit and receive demo and enable interrupt subroutine. To use UART1 function must remove ICE connect since ICE function will disturb UART1 transmit.
UART0_Interrupt_RW	UART1 with interrupt, received data from UART1_RXD transmit to UART1_TXD loop. To use UART1 function must remove ICE connect since ICE function will disturb UART1 transmit.
UART1_printf	Code use “printf” instruction transmit from UART1 setting. Special with “putchar.c” define. To use UART1 function must

	remove ICE connect since ICE function will disturb UART1 transmit.
WakeupTimer_Interrupt	Wakeup timer over into interrupt to toggle GPIO P1.2 to show result. Based on LIRC clock as source.
Watchdog_Interrupt	Watchdog timer over only jump into interrupt. This auto reload timer counter not with reset and not need special define in CONFIG.
Watchdog_Reset_Disable	After watchdog reset function is enabled, only modify CONFIG to disable it can disable WDT reset function. This demo is use IAP to disable CONFIG WDT function.
Watchdog_Reset_Enable	Watchdog timer over cause reset initial. Include “CONFIG” enable use IAP command and check with POF flag if reset is enabled before, not do IAP again.

7 REVISION HISTORY

Date	Revision	Description
2019.1.29	1.00	Initially issued.

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.