

ML51 Series BSP User Guide

Based on Keil uVision4 and PK51 Development Kit V9.52
For NuMicro® 8051 Family

Directory Information

Please extract the “ML51_BSP_Keil_C51_V2.00.001.zip” file firstly and confirm the following content of this BSP folder.

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.
TK-Utility	Touch Key Calibration Tool.

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\

ML51_Series_BSP_Keil_Revision_History.pdf

This document shows the revision history of ML51_Series_BSP_Keil.

2 .\Library\

Device\	ML51 series flash device header file.
Startup\	A51 startup file and executable file.
StdDriver\	All peripheral driver header and source files.

3 .\SampleCode\

NK_ML56SD\	Demonstrate the usage of NK-ML56SD function project.
RegBased\	Demonstrate the usage of ML51 series MCU peripheral driver.
StdDriver\	Project based on StdDriver\ library to setup MCU peripheral driver.
Template\	project template for ML51 series MCU

4 .\SampleCode\NK_ML56SD

NK_ML56SD_LCD_TK

Demonstrate how to calibration TK0 in the NuMaker ML56SD. After completion, LCD displays with the ML56SD real temperature, firmware version and will be changed information according to TK0 pressed.

5 .\SampleCode\RegBased

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.
ADC_VTEMP	Demonstrate how to use ADC convert to get temperature sensor value.
FMC_ROM_CONST_0x3800	Show how to locate two table data file in the APROM address start from 0x3800 by special define options of keil project.
FMC_ROM_CONST_simple	Show how to simply include const table data in the APROM address without special define.
Fsys_Select	Change the ML51 initial setting of system clock from HIRC to the external clock input. The ML51 external clock input ranges from 4 MHz to 24 MHz.
GPIO_ClockOut	Show the ML51 system clock and output from CLKO pin.
GPIO_Input_Output	Toggle each ML51 GPIO pin output from high to low after 200ms delay.
GPIO_INT0_ExtInt	Perform ML51 external interrupt pin INT0 enabled initial setting.
GPIO_INT1_ExtInt	Perform ML51 external interrupt pin INT1 enabled initial setting.
GPIO_Pin_Interrupt	Demonstrate how to wake up ML51 from Idle / Power-down mode through external interrupt input by enabling ML51 pin interrupt function.
I2C_EEPROM_Read_Write	Show how to use ML51 as master to read external connect EEPROM by I ² C bus.

I2C_Master	Show how a master accesses a slave. This sample code needs to work with I2C_Slave.
I2C_Slave	Demonstrate how to set I2C in Slave mode to receive 10 bytes data from a master. This sample code needs to work with I2C_Master.
Interrupt_ISR_all	List all ML51 interrupt vector.
ISP_UART0	Sample ISP firmware communicated with ISP tool through UART0 interface.
ISP_UART1	Sample ISP firmware communicated with ISP tool through UART1 interface.
ISP_UART2	Sample ISP firmware communicated with ISP tool through UART2 interface.
PowerDown_BODdisable	Show how to disable ML51 Burn-out detect function and into power down mode. This project is special for measure the power down current of ML51.
PWM0_DeadTime	Configure PWM as Complementary mode. Control 3 pairs output, set each 2 channel PWM output as same duty and insert dead time.
PWM_Interrupt	Configure PWM one channel output with falling edge trig interrupt enable initial setting.
PWM_Simple	Configure PWM as Independent mode. Only set one channel output.
PWM0123_GroupStart	Configure PWM0/1/2/3 as independent mode and start with same time.
SPROM	Demonstrate how ML51 runs in SPROM from startup from APROM and implements lock SPROM function.
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 define.
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.
WakeupTimer_Interrupt	Enable wake-up timer with interrupt function. Main loop enters Power-down mode after initial setting, and once WKT timeout, ML51 will wake up and then jump into interrupt subroutine to toggle GPIO output.
Watchdog_Interrupt	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.

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

6 .\SampleCode\StdDriver

ACMP_CRV	Demonstrate analog comparator (ACMP) comparison by comparing ACMP0_P0 input and internal voltage (CRV) and shows the result on UART console in ACMP interrupt.
ACMP_VBG	Demonstrate analog comparator (ACMP) comparison by comparing ACMP0_P0 input and band-gap and shows the result on UART console in ACMP interrupt
ACMP_Wakeup	Use ACMP to wake up system from Power-down mode while comparator output changes.
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_Continuous	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.
ADC_InternalVref	Configure ADC reference is internal voltage reference and shows the result on UART console.
ADC_VTEMP_VREF	Calculate the real temperature value of the device system based on the internal temperature sensor. Configure ADC reference is internal voltage reference and shows the result on UART console.
GPIO_Input_Output	Toggle each ML51 GPIO pin output from high to low after 200ms delay.
GPIO_Pin_Interrupt	Demonstrate how to wake up ML51 from Idle / Power-down mode through external interrupt input by enabling ML51 pin interrupt function.
IAP_APROM_CRC	Demonstrate how ML51 IAP runs in APROM to program with CRC check. shows the CRC result on UART console.
IAP_AP_program_AP_Dataflash	Demonstrate how ML51 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_LDROM	Demonstrate how ML51 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.

IAP_Dataflash_EEPROM_Array	Simulate APROM 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.
IAP_Dataflash_EEPROM_SPROM_Array	Simulate SPROM as EEPROM mode by calling the library file "eeprom_sprom.c".
IAP_LD_Program_AP	Demonstrate how ML51 IAP runs in LDROM to program APROM and implements erase / program / read verify function.
IAP_Read_UCID	Demonstrate using ML51 IAP command to read the unique customer ID (UCID). Only for customer special order ML51 MCU. One UCID is only for one customer.
IAP_Read_UID	Demonstrate using ML51 IAP command to read the Unique code of ML51. The UID value of each ML51 is different.
LCD_Blinking	Configure LCD driving 1/4 bias, 8COM, 32SEG, VLCD base on internal charge pump 3.6V, buffer power saving mode turn on 1/4 cycle and with blinking display demo.
LCD_Interrupt	Configure LCD driving with internal charge pump interrupt and show the result on UART console.
LPR_LPLVR	Demonstrate ML51 run in Low power run mode and enable low power LVR function to decrease power consumption demo.
LPR_PDMA_MemoryToMemory	Demonstrate ML51 run in low power run mode to implement data transfer by PDMA memory to memory.
LPR_PWM	Demonstrate ML51 run in low power run mode execute PWM output function.
LPR_UART2_PDMA	Demonstrate ML51 run in low power run mode and enable UART2 receive data storage in XRAM through configuration PDMA.
PDMA_MemoryToMemory	Demonstrate ML51 PDMA implement memory to memory transfer demo.
PDMA_SPI_RX	Configure SPI0 as Slave mode and demonstrate how to receive data by PDMA to continuous storage the result in XRAM.
PDMA_SPI_TX	Configure SPI0 as master mode and demonstrate how to implement PDMA to continuous transmit data
PDMA_UART2_Memory	Demonstrate how to implement UART2 continuous receive data and storage in XRAM base on PDMA demo.

PWM0_Complementary	Configure PWM as Complementary mode. Set PWM Channel 0/2/4 output with independent duty value. PWM1/3/5 outputs follow PMW 0/2/4 setting.
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.
PWM123_Independent	Configure PWM0/1/2/3 as independent mode each channel duty is different.
RTC_Alarm_Interrupt	Demonstrate the RTC alarm function. It sets an alarm 10 seconds after execution.
RTC_Tick_Detection	Demonstrate how to use RTC one second tick interrupt event to wake up system.
SPI_Flash_Read_Write	Demonstrate ML51 how to driving W25Q16BV SPI Flash and set it as master to read and write data sample code.
SPI_Master	Configure SPI as master transmit and receive data. Check flag as transmit one byte finish condition.
SPI_Slave_Interrupt	Configure SPI as slave transmit and receive data. Enable interrupt function when slave is receive data.
SPI_Slave_Polling	Configure SPI as slave transmit and receive data. Check flag as received one byte finish condition.
Timer2_AutoReload_Capture	Show how to use the timer2 capture function to capture timer2 counter value.
UART0_Interrupt_RW	Configure UART0 transfer including transmit and receive with interrupt enabled.
UART0_Printf	Demonstrate printf instruction based on transmit from UART0 TXD pin.
UART1	Configure UART1 transmit and receive initial setting and enable interrupt subroutine.
UART1_Printf	Demonstrate printf instruction based on transmit from UART1 TXD pin. Include how to modify putchar.c file.
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.
WakeupTimer_Interrupt	Implement WKT time-out interrupt event to wake up system and generate time-out to period wakeup demo
Watchdog_Interrupt	Implement WDT time-out interrupt event to wake up system and generate time-out reset system event while WDT time-out reset delay period expired.

7 .\SampleCode\Template

Project_temp	A project template for ML51
TouchKey	Demonstrate Touch Key demo system. Include TK calibration or normal run sample and whole Touch key library.

8 COMPATIBILITY LIST

Project Name	ML56	ML54	ML51		
			64KB	32KB	16KB
NK_ML56SD_LCD_TK	√				
ADC_GPIO_Trig	√	√	√	√	√
ADC_Multi_channel	√	√	√	√	√
ADC_PWM_Trig	√	√	√	√	√
ADC_Simple	√	√	√	√	√
ADC_VTEMP	√	√	√	√	√
FMC_ROM_CONST_0x3800	√	√	√	√	√
FMC_ROM_CONST_simple	√	√	√	√	√
Fsys_Select	√	√	√	√	√
GPIO_ClockOut	√	√	√	√	√
GPIO_Input_Output	√	√	√	√	√
GPIO_INT0_ExtInt	√	√	√	√	√
GPIO_INT1_ExtInt	√	√	√	√	√
GPIO_Pin_Interrupt	√	√	√	√	√
I2C_EEPROM_Read_Write	√	√	√	√	√
I2C_Master	√	√	√	√	√
I2C_Slave	√	√	√	√	√
Interrupt_ISR_all	√	√	√	√	√
ISP_UART0	√	√	√	√	√
ISP_UART1	√	√	√	√	√
ISP_UART2	√	√	√	√	√
PowerDown_BODdisable	√	√	√	√	√
PWM0_DeadTime	√	√	√	√	√
PWM_Interrupt	√	√	√	√	√
PWM_Simple	√	√	√	√	√
PWM0123_GroupStart	√	√	√		
SPROM	√	√	√	√	√
Timer0_mode_0_Interrupt	√	√	√	√	√

Timer0_mode_1_Interrupt	√	√	√	√	√
Timer0_mode_2_Interrupt	√	√	√	√	√
Timer01_mode_3_Interrupt	√	√	√	√	√
Timer1_mode_0_Interrupt	√	√	√	√	√
Timer1_mode_1_Interrupt	√	√	√	√	√
Timer1_mode_2_Interrupt	√	√	√	√	√
Timer2_AutoReload_Capture	√	√	√	√	√
Timer2_AutoReload_Delay	√	√	√	√	√
Timer3	√	√	√	√	√
WakeupTimer_Interrupt	√	√	√	√	√
Watchdog_Interrupt	√	√	√	√	√
ACMP_CRV	√	√	√		
ACMP_VBG	√	√	√		
ACMP_Wakeup	√	√	√		
ADC_Bandgap_VDD	√	√	√	√	√
ADC_Continuous	√	√	√	√	√
ADC_InternalVref	√	√	√	√	√
ADC_VTEMP_VREF	√	√	√		
GPIO_Input_Output	√	√	√	√	√
GPIO_Pin_Interrupt	√	√	√	√	√
IAP_APROM_CRC	√	√	√		
IAP_AP_program_AP_Dataflash	√	√	√	√	√
IAP_AP_program_LDROM	√	√	√	√	√
IAP_Dataflash_EEPROM_Array	√	√	√	√	√
IAP_Dataflash_EEPROM_SPROM_Array	√	√	√	√	√
IAP_LD_Program_AP	√	√	√	√	√
IAP_Read_UCID	√	√	√	√	√
IAP_Read_UID	√	√	√	√	√
LCD_Blinking	√	√			
LCD_Interrupt	√	√			
LPR_LPLVR	√	√	√	√	√
LPR_PDMA_MemoryToMemory	√	√	√	√	√

LPR_PWM	√	√	√	√	√
LPR_UART2_PDMA	√	√	√	√	√
PDMA_MemoryToMemory	√	√	√	√	√
PDMA_SPI_RX	√	√	√	√	√
PDMA_SPI_TX	√	√	√	√	√
PDMA_UART2_Memory	√	√	√	√	√
PWM0_Complementary	√	√	√	√	√
PWM0_Independent	√	√	√	√	√
PWM0_Synchronous	√	√	√	√	√
PWM123_Independent	√	√	√		
RTC_Alarm_Interrupt	√	√	√		
RTC_Tick_Detection	√	√	√		
SPI_Flash_Read_Write	√	√	√	√	√
SPI_Master	√	√	√	√	√
SPI_Slave_Interrupt	√	√	√	√	√
SPI_Slave_Polling	√	√	√	√	√
Timer2_AutoReload_Capture	√	√	√	√	√
UART0_Interrupt_RW	√	√	√	√	√
UART0_Printf	√	√	√	√	√
UART1	√	√	√	√	√
UART1_Printf	√	√	√	√	√
UART2	√	√	√	√	√
UART3	√	√	√		
WakeupTimer_Interrupt	√	√	√	√	√
Watchdog_Interrupt	√	√	√	√	√
Project_temp	√	√	√	√	√
TouchKey	√				

9 REVISION HISTORY

Date	Revision	Description
2019.1.29	1.00.001	Initial release.
2019.6.27	1.00.002	1. Added ML51DA9AE_ML51BA9AE_BSP_Keil folder 2. Modified ML51_16K_BSP_Keil_V1.00 folder name to ML51FB9AE_ML51XB9AE_ML51XB9BE_BSP_Keil
2019.11.27	1.00.003	Added ML51FC0AE_ML51XC0BE_ML51EC0AE_ML51TC0AE_ML51PC0AE_BSP_Keil
2020.01.06	1.00.004	Added SPI_Master, SPI_Slave_interrupt, SPI_Slave_polling, in all folder. Modified SPI_Flash_Read_Write in all folder. Added spi.c , spi_flash.c in all library.
2020.09.01	2.00.001	Added project to support ML56/ML54/ML64KB Flash series.

NK_ML56SD_LCD_TK

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.

