Mini57 Series CMSIS BSP Guide

Directory Introduction for 32-bit NuMicro® Family

Directory Information

Please extract the “Mini57 Series BSP_CMSIS_V3.01.000.zip” file firstly, and then put the “Mini57 Series BSP_CMSIS_V3.01.000” folder into the working folder (e.g. .\Nuvoton\BSP Library\).

This BSP folder contents:

Document	Device 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\

CMSIS.html	Introduction of CMSIS version 4.5.0. CMSIS components included CMSIS-CORE, CMSIS-Driver, CMSIS-DSP, etc. ● CMSIS-CORE: API for the Cortex-M0 processor core and peripherals. ● CMSIS-Driver: Defines generic peripheral driver interfaces for middleware making it reusable across supported devices. ● CMSIS-DSP: DSP Library Collection with over 60 functions for various data types: fix-point (fractional q7, q15, q31) and single precision floating-point (32-bit).
NuMicro Mini57 Series CMSIS BSP Revision History.pdf	The revision history of Mini57 BSP.
NuMicro Mini57 Series Driver Reference Guide.chm	The usage of drivers in Mini57 BSP.

2 .\Library\

CMSIS	Cortex® Microcontroller Software Interface Standard (CMSIS) V4.5.0 definitions by ARM® Corp.
Device	CMSIS compliant device header file.
StdDriver	All peripheral driver header and source files.

3 .\Sample Code\

Hard_Fault_Sample	Show hard fault information when hard fault happened. The hard fault handler shows some information included program counter, which is the address where the processor was executing when a hard fault occurred. The listing file (or map file) can show what function and instruction that was. It also shows the Link Register (LR), which contains the return address of the last function call. It can show the status where CPU comes from to get to this point.
Semihost	Show how to print and get character through IDE console window.
RegBase	The sample codes which access control registers directly.
StdDriver	Demonstrate the usage of Mini57 series MCU peripheral driver APIs.
Template	A project template for Mini57 series MCU.

4 .\SampleCode\RegBase

ACMP	Demonstrate analog comparator (ACMP) comparison by comparing ACMP0_P0 input and VBG voltage and show the result on UART console.
BPWM_DeadZone	Demonstrate the BPWM dead-zone feature.
BPWM_DoubleBuffer	Demonstrate the BPWM double buffer feature.
EADC_Ind2SH	Convert ADC0 channel 0 and ADC1 channel 0 in Independent 2SH mode and print conversion results.
EADC_IndSimple	Convert ADC0 channel 0 and ADC1 channel 0 in Independent Simple mode and print conversion results.
EADC_IndSimple_BandGap	Convert ADC0 channel 6 (Band-Gap) in Independent Simple mode and print conversion results.
EADC_IndSimple_TempSensor	Convert ADC1 channel 6 (Temperature Sensor) in Independent Simple mode and print conversion results.
EADC_PWMTrigger	Configure PWM0 to trigger ADC0 channel 0 periodically and print conversion results.
EADC_SimSeq3R	Convert ADC0 channel 0, channel 6, and ADC1 channel 0 in EADC Simultaneous Sequential 3R mode and print conversion results.
EADC_SimSeq4R	Convert ADC0 channel 0, channel 6, ADC1 channel 0, and channel 3 in EADC Simultaneous Sequential 4R mode and print conversion results.
EADC_SimSimple	Convert ADC0 channel 0 and ADC1 channel 0 in Simultaneous Simple mode and print conversion results.
EADC_TimerTrigger	Configure Timer0 to trigger ADC0 channel 0 periodically and print conversion results.

EADC_Valid_Overrun	Demonstrate how to check OVERRUN status, VALID status, and read data by correct order.
EADC_WCompare	Demonstrate EADC conversion and window comparison function by monitoring the conversion result of ADC0 channel 0.
ECAP_Capture	Configure ECAP channel 0 to capture input square wave and print capture results. The input square wave is generated by Timer0 and GPIO output pin.
ECAP_CmpMatch	Demonstrate ECAP capture and compare match function by monitoring the capture result of ECAP channel 0.
EPWM_DeadZone	Demonstrate the EPWM dead-zone feature.
EPWM_DoubleBuffer	Demonstrate the EPWM double buffer feature.
FMC_CRC32	Show FMC CRC32 calculation capability.
FMC_IAP	Include LDROM image (fmc_ld_iap) and APROM image (fmc_ap_main), which shows how to branch between APROM and LDROM. To run this sample code, the boot mode must be "Boot from APROM with IAP".
FMC_RW	Show FMC read Flash IDs, erase, read, and write functions.
GPIO_IOTest	Use GPIO driver to control the GPIO pin direction and the high/low state, and show how to use GPIO interrupts.
GPIO_PowerDown	Demonstrate how to wake up system form Power-down mode by GPIO interrupt.
HDIV	Demonstrate how to divide two signed integers by HDIV engine.
PGA_PGAO	Demonstrate how to amplify input signals with different gain levels and output to PGA_O output pin.
SYS_CLKO	Demonstrate how to output different clocks one after another to the same CLKO (PA0) pin.

SYS_Control	Demonstrate some system manager controller functions, including reading PDID, getting reset source, system write-protection, power-down wake up by Watchdog timer, and CPU reset.
TIMER_Delay	Demonstrate the usage of TIMER_Delay() API to generate a 1 second delay.
TIMER_EventCounter	Use pin PB.3 to demonstrate timer event counter function.
TIMER_FreeCountingMode	Use the ACMP0 positive input pin to demonstrate timer free counting mode function, and display the measured input frequency to console.
TIMER_Periodic	Use the timer periodic mode to generate timer interrupt every 1 second.
TIMER_ToggleOut	Demonstrate the timer 0 toggle out function on the pin PB.3.
TIMER_Wakeup	Use the timer to wake up system from Power-down mode periodically.
USCI_I2C_EEPROM	Show how to use USCI_I2C interface to access EEPROM.
USCI_I2C_Master	Show how to set USCI_I2C in Master mode and send data to Slave device. This sample code needs to work with USCI_I2C_Slave.
USCI_I2C_Slave	Show how to set USCI_I2C in Slave mode and receive the data from Master. This sample code needs to work with USCI_I2C_Master.
USCI_SPI_Loopback	Implement USCI_SPI1 Master loop back transfer. This sample code needs to connect USCI_SPI1_MISO pin and USCI_SPI1_MOSI pin together. It will compare the received data with transmitted data.
USCI_SPI_MasterMode	Configure USCI_SPI1 as Master mode and demonstrate how to communicate with an off-chip SPI Slave device. This sample code needs to work with USCI_SPI_SlaveMode.

USCI_SPI_SlaveMode	Configure USCI_SPI1 as Slave mode and demonstrate how to communicate with an off-chip SPI Master device. This sample code needs to work with USCI_SPI_MasterMode.
USCI_UART_TxRxFunction	Transmit and receive data from PC terminal through an RS232 interface.
WDT_Polling	Use polling mode to check WDT time-out state and reset WDT after time out occurs.
WDT_Wakeup	Use WDT to wake up system from Power-down mode periodically.

5 .\SampleCode\StdDriver

ACMP	Demonstrate analog comparator (ACMP) comparison by comparing ACMP0_P0 input and VBG voltage and show the result on UART console.
BPWM_DeadZone	Demonstrate the BPWM dead-zone feature.
BPWM_DoubleBuffer	Demonstrate the BPWM double buffer feature.
EADC_Ind2SH	Convert ADC0 channel 0 and ADC1 channel 0 in Independent 2SH mode and print conversion results.
EADC_IndSimple	Convert ADC0 channel 0 and ADC1 channel 0 in Independent Simple mode and print conversion results.
EADC_IndSimple_BandGap	Convert ADC0 channel 6 (Band-Gap) in Independent Simple mode and print conversion results.
EADC_IndSimple_TempSensor	Convert ADC1 channel 6 (Temperature Sensor) in Independent Simple mode and print conversion results.
EADC_PWMTrigger	Configure PWM0 to trigger ADC0 channel 0 periodically and print conversion results.
EADC_SimSeq3R	Convert ADC0 channel 0, channel 6, and ADC1 channel 0 in EADC Simultaneous Sequential 3R mode and print conversion results.
EADC_SimSeq4R	Convert ADC0 channel 0, channel 6, ADC1 channel 0, and channel 3 in EADC Simultaneous Sequential 4R mode and print conversion results.
EADC_SimSimple	Convert ADC0 channel 0 and ADC1 channel 0 in Simultaneous Simple mode and print conversion results.
EADC_TimerTrigger	Configure Timer0 to trigger ADC0 channel 0 periodically and print conversion results.

EADC_Valid_Overrun	Demonstrate how to check OVERRUN status, VALID status, and read data by correct order.
EADC_WCompare	Demonstrate EADC conversion and window comparison function by monitoring the conversion result of ADC0 channel 0.
ECAP_Capture	Configure ECAP channel 0 to capture input square wave and print capture results. The input square wave is generated by Timer0 and GPIO output pin.
ECAP_CmpMatch	Demonstrate ECAP capture and compare match function by monitoring the capture result of ECAP channel 0.
EPWM_DeadZone	Demonstrate the EPWM dead-zone feature.
EPWM_DoubleBuffer	Demonstrate the EPWM double buffer feature.
FMC_CRC32	Show FMC CRC32 calculation capability.
FMC_IAP	Include LDROM image (fmc_ld_iap) and APROM image (fmc_ap_main), which shows how to branch between APROM and LDROM. To run this sample code, the boot mode must be "Boot from APROM with IAP".
FMC_RW	Show FMC read Flash IDs, erase, read, and write functions.
GPIO_IOTest	Use GPIO driver to control the GPIO pin direction and the high/low state, and show how to use GPIO interrupts.
GPIO_PowerDown	Demonstrate how to wake up system form Power-down mode by GPIO interrupt.
HDIV	Demonstrate how to divide two signed integers by HDIV engine.
PGA_PGAO	Demonstrate how to amplify input signals with different gain levels and output to PGA_O output pin.
SYS_CLKO	Demonstrate how to output different clocks one after another to the same CLKO (PA0) pin.

SYS_Control	Demonstrate some system manager controller functions, including reading PDID, getting reset source, system write-protection, power-down wake up by Watchdog timer, and CPU reset.
TIMER_Delay	Demonstrate the usage of TIMER_Delay() API to generate a 1 second delay.
TIMER_EventCounter	Use pin PB.3 to demonstrate timer event counter function.
TIMER_FreeCountingMode	Use the ACMP0 positive input pin to demonstrate timer free counting mode function, and display the measured input frequency to console.
TIMER_Periodic	Use the timer periodic mode to generate timer interrupt every 1 second.
TIMER_ToggleOut	Demonstrate the timer 0 toggle out function on the pin PB.3.
TIMER_Wakeup	Use the timer to wake up system from Power-down mode periodically.
USCI_I2C_EEPROM	Show how to use USCI_I2C interface to access EEPROM.
USCI_I2C_Master	Show how to set USCI_I2C in Master mode and send data to Slave device. This sample code needs to work with USCI_I2C_Slave.
USCI_I2C_Slave	Show how to set USCI_I2C in Slave mode and receive the data from Master. This sample code needs to work with USCI_I2C_Master.
USCI_SPI_Loopback	Implement USCI_SPI1 Master loop back transfer. This sample code needs to connect USCI_SPI1_MISO pin and USCI_SPI1_MOSI pin together. It will compare the received data with transmitted data.
USCI_SPI_MasterMode	Configure USCI_SPI1 as Master mode and demonstrate how to communicate with an off-chip SPI Slave device. This sample code needs to work with USCI_SPI_SlaveMode.

USCI_SPI_SlaveMode	Configure USCI_SPI1 as Slave mode and demonstrate how to communicate with an off-chip SPI Master device. This sample code needs to work with USCI_SPI_MasterMode.
USCI_UART_TxRxFunction	Transmit and receive data from PC terminal through an RS232 interface.
WDT_Polling	Use polling mode to check WDT time-out state and reset WDT after time out occurs.
WDT_Wakeup	Use WDT to wake up system from Power-down mode periodically.

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.