M2A23 Series CMSIS BSP Guide

Directory Introduction for 32-bit NuMicro™ Family

Directory Information

Please extract the “M2A23_Series_BSP_CMSIS_V3.00.001.zip” file firstly, and then put the “M2A23_Series_BSP_CMSIS_V3.00.001” 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

TABLE OF CONTENTS

1 Document

CMSIS.html	Introduction of CMSIS version 5. 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).
Revision History.pdf	The revision history of M2A23 Series BSP.
NuMicro M2A23 Series Driver Reference Guide.chm	The usage of drivers in M2A23 Series BSP.

2 Library

CMSIS	Cortex® Microcontroller Software Interface Standard (CMSIS) V5 definitions by ARM® Corp.
Device	CMSIS compliant device header file.
LlsiYcableLib	Library for accessing LlsiYcable.
NuMaker	Specific libraries for M2A23 NuMaker board.
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 show some information included program counter, which is the address where the processor was executing when the hard fault occur. 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.
ISP	Sample codes for In-System-Programming.
StdDriver	Demonstrate the usage of M2A23 series MCU peripheral driver APIs.
Template	A project template for M2A23 series MCU.

4 SampleCode\ISP

ISP_CAN	In-System-Programming Sample code through CAN interface.
ISP_I2C	In-System-Programming Sample code through I^2C interface.
ISP_RS485	In-System-Programming Sample code through RS485 interface.
ISP_SPI	In-System-Programming Sample code through SPI interface.
ISP_UART	In-System-Programming Sample code through UART interface.

5 SampleCode\StdDriver

System Manager (SYS)

SYS_BODWakeup	Show how to wake up system form Power-down mode by brown-out detector interrupt.
SYS_PLLClockOutput	Change system clock to different PLL frequency and output system clock from CLKO pin.
SYS_PowerDown_MinCurrent	Demonstrate how to minimize power consumption when entering power down mode.
SYS_TrimIRC	Demonstrate how to use LXT to trim HIRC.

Clock Controller (CLK)

CLK_ClockDetector	Show the usage of clock fail detector and clock frequency monitor function.
CLK_ClockDetectorWakeup	Show how to wake up system from Power-down mode by clock fail detector interrupt.

Flash Memory Controller (FMC)

FMC_APPROT	Demonstrate how to use FMC APROM Protect function.
FMC_CRC32	Demonstrate how to use FMC CRC32 ISP command to calculate the CRC32 checksum of APROM, LDROM, and SPROM.
FMC_DualBank	Demonstrate how dual processes work in dual bank flash architecture.
FMC_DualBankFwUpgrade	Implement a firmware update mechanism based on dual bank flash architecture.
FMC_ExeInSRAM	Implement a code and execute in SRAM to program embedded Flash.
FMC_FwUpgradeApplication	Bank remap sample code.

FMC_IAP	Show how to set VECMAP to LDROM and reboot to LDROM from APROM.
FMC_MultiBoot	Implement a multi-boot system to boot from different applications in APROM/LDROM.
FMC_MultiWordProgram	This sample run on SRAM to show FMC multi word program function.
FMC_ReadAllOne	Demonstrate how to use FMC Read-All-One ISP command to verify APROM/LDROM pages are all 0xFFFFFFFF or not.
FMC_RW	Demonstrate how to read/program embedded Flash by ISP function.
FMC_SPROM	This sample shows how to make an application running on APROM but with a sub-routine on SPROM, which can be secured.

General Purpose I/O (GPIO)

GPIO_EINTAndDebounce	Show the usage of GPIO external interrupt function and de-bounce function.
GPIO_EINTTriggerPDMA	Show the usage of GPIO EINT trigger PDMA function.
GPIO_EINTTriggerPWM	Show the usage of GPIO EINT trigger PWM function.
GPIO_INT	Show the usage of GPIO interrupt function.
GPIO_OutputInput	Show how to set GPIO pin mode and use pin data input/output control.
GPIO_PowerDown	Show how to wake up system from Power-down mode by GPIO interrupt.

PDMA Controller (PDMA)

PDMA_BasicMode	Use PDMA Channel 2 to transfer data from memory to memory.
PDMA_ScatterGather	Use PDMA Channel 4 to transfer data from memory to memory by scatter-gather mode.

PDMA_ScatterGather_PingPongBuffer	Use PDMA to implement Ping-Pong buffer by scatter-gather mode (memory to memory).

Timer Controller (TIMER)

TIMER_ACMPTrigger	Use ACMP to trigger Timer0 counter reset mode.
TIMER_CaptureCounter	Show how to use the timer2 capture function to capture timer2 counter value.
TIMER_Delay	Show how to use timer0 to create various delay time.
TIMER_EventCounter	Implement timer1 event counter function to count the external input event.
TIMER_FreeCountingMode	Use the timer0 pin PA.11 to demonstrate timer free counting mode function. And displays the measured input frequency to UART console.
TIMER_InterTimerTriggerMode	Demonstrate how to use Inter-Timer trigger function.
TIMER_Periodic	Use the timer periodic mode to generate timer interrupt every 1 second.
TIMER_PeriodicINT	Implement timer counting in periodic mode.
TIMER_TimeoutWakeup	Use timer0 periodic time-out interrupt event to wake up system.
TIMER_ToggleOut	Implement timer counting in toggle-output mode.

Watchdog Timer (WDT)

WDT_TimeoutWakeupAndReset	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.

Window Watchdog Timer (WWDT)

WWDT_CompareINT	Show how to reload the WWDT counter value.

PWM Generator and Capture Timer (PWM)

PWM_AccumulatorINT_TriggerPDMA	Demonstrate PWM accumulator interrupt trigger PDMA.
PWM_AccumulatorStopMode	Demonstrate PWM accumulator stop mode.
PWM_Brake	Demonstrate how to use PWM brake function.
PWM_Capture	Capture the PWM0 Channel 0 waveform by PWM0 Channel 2.
PWM_DeadTime	Demonstrate how to use PWM Dead Zone function.
PWM_DoubleBuffer	Change duty cycle and period of output waveform by PWM Double Buffer function.
PWM_OutputWaveform	Demonstrate how to use PWM counter output waveform.
PWM_PDMA_Capture	Capture the PWM0 Channel 0 waveform by PWM0 Channel 2, and use PDMA to transfer captured data.
PWM_PDMA_Capture_1MHz Signal	Capture the PWM0 Channel 0 waveform by PWM0 Channel 2, and use PDMA to transfer captured data. Frequency of PWM Channel 0 is 1 MHz to test maximum input frequency for PWM Capture function.
PWM_SwitchDuty	Change duty cycle of output waveform by configured period.
PWM_SyncStart	Demonstrate how to use PWM counter synchronous start function.

Basic PWM Generator and Capture Timer (BPWM)

BPWM_DoubleBuffer	Change duty cycle and period of output waveform by BPWM Double Buffer function.
BPWM_OutputWaveform	Demonstrate how to use BPWM counter output waveform.
BPWM_SwitchDuty	Change duty cycle of output waveform by configured period.
BPWM_SyncStart	Demonstrate how to use BPWM counter synchronous start function.

UART Interface Controller (UART)

UART_AutoBaudRate	Show how to use auto baud rate detection function.
UART_Autoflow	Transmit and receive data with auto flow control.
UART_IrDA	Transmit and receive data in UART IrDA mode.
UART_LIN	Transmit LIN header and response.
UART_LIN_Wakeup	LIN Power-down/Wake-up and header and response.
UART_PDMA	Transmit and receive UART data with PDMA.
UART_RS485	Transmit and receive data in UART RS485 mode.
UART_SingleWire	Transmit and receive data by UART Single-Wire mode.
UART_TxRxFunction	Transmit and receive data from PC terminal through RS232 interface.
UART_Wakeup	Show how to wake up system from Power-down mode by UART interrupt.

Serial Peripheral Interface (SPI)

SPI_Loopback	Implement SPI Master loop back transfer. This sample code needs to connect MISO pin and MOSI pin together. It will compare the received data with transmitted data.
SPI_MasterFIFOMode	Configure SPI0 as Master mode and demonstrate how to communicate with an off-chip SPI Slave device with FIFO mode. This sample code needs to work with SPI_SlaveFIFOMode sample code.
SPI_SlaveFIFOMode	Configure SPI0 as Slave mode and demonstrate how to communicate with an off-chip SPI Master device with FIFO mode. This sample code needs to work with SPI_MasterFIFOMode sample code.

I²C Serial Interface Controller (I²C)

I2C_EEPROM	Demonstrate how to access EEPROM through a I²C interface

I2C_Master	Demonstrate how a Master accesses Slave. This sample code needs to work with I2C_Slave sample code.
I2C_MultiBytes_Master	Demonstrate how to use multi-bytes API to access slave. This sample code needs to work with I2C_Slave sample code.
I2C_SingleByte_Master	Demonstrate how to use single byte API to access slave. This sample code needs to work with I2C_Slave sample code.
I2C_Slave	Demonstrate how to set I^2C in slave mode to receive the data from a Master. This sample code needs to work with I2C_Master sample code.
I2C_Wakeup_Slave	Show how to wake up MCU from Power-down mode through I2C interface. This sample code needs to work with I2C_Master sample code.

Universal Serial Control Interface Controller - UART Mode (USCI-UART)

USCI_UART_AutoBaudRate	Show how to use auto baud rate detection function
USCI_UART_Autoflow	Transmit and receive data using auto flow control.
USCI_UART_PDMA	Transmit and receive UART data with PDMA.
USCI_UART_RS485	Transmit and receive data in UART RS485 mode.
USCI_UART_TxRxFunction	Transmit and receive data from PC terminal through RS232 interface.
USCI_UART_Wakeup	Show how to wake up system from Power-down mode by USCI interrupt in UART mode.

Universal Serial Control Interface Controller - SPI Mode (USCI-SPI)

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

	sample code.
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 sample code.
USCI_SPI_PDMA_LoopTest	Demonstrate USCI_SPI data transfer with PDMA. USCI_SPI0 will be configured as Master mode and USCI_SPI1 will be configured as Slave mode. Both TX PDMA function and RX PDMA function will be enabled.

Universal Serial Control Interface Controller - I2C Mode (USCI-I2C)

USCI_I2C_EEPROM	Demonstrate how to access EEPROM through a USCI_I2C interface.
USCI_I2C_Loopback	Show a Master how to access 7-bit address Slave.s
USCI_I2C_Loopback_10bit	Show a Master how to access 10-bit address Slave.
USCI_I2C_Master	Demonstrate how a Master access Slave. This sample code needs to work with USCI_I2C_Slave sample code.
USCI_I2C_Master_10bit	Demonstrate how a Master use 10-bit addressing access Slave. This sample code needs to work with USCI_I2C_Slave_10bit sample code.
USCI_I2C_MultiBytes_Master	Demonstrate how to use multi-bytes API to access slave. This sample code needs to work with USCI_I2C_Slave sample code.
USCI_I2C_SingleByte_Master	Demonstrate how to use single byte API to access slave. This sample code needs to work with USCI_I2C_Slave sample code.
USCI_I2C_Slave	Demonstrate how to set I^2C in slave mode to receive the data from a Master. This sample code needs to work with USCI_I2C_Master sample code.
USCI_I2C_Slave_10bit	Demonstrate how to set I^2C in 10-bit addressing slave mode to receive the data from a Master. This sample code needs to work withUSCI_I2C_Master_10bit sample code.
USCI_I2C_Wakeup_Slave	Demonstrate how to set I^2C to wake up MCU from Power-down mode. This sample code needs to work

	USCI_I2C_Master sample code.

CRC Controller (CRC)

CRC_CCITT	Implement CRC in CRC-CCITT mode and get the CRC checksum result.
CRC_CRC8	Implement CRC in CRC-8 mode and get the CRC checksum result.
CRC_CRC32	Implement CRC in CRC-32 mode with PDMA transfer.

Analog-to-Digital Converter (ADC)

ADC_BandGap	Convert Band-gap (channel 29) and print conversion result.
ADC_BurstMode	Perform A/D Conversion with ADC burst mode.
ADC_ContinuousScanMode	Perform A/D Conversion with ADC continuous scan mode.
ADC_PDMA_SingleCycleScanMode	Perform A/D Conversion with ADC single cycle scan mode and transfer result by PDMA.
ADC_PwmTrigger	Demonstrate how to trigger ADC by PWM.
ADC_ResultMonitor	Monitor the conversion result of Channel 2 by the digital compare function.
ADC_SingleCycleScanMode	Perform A/D Conversion with ADC single cycle scan mode.
ADC_SingleMode	Perform A/D Conversion with ADC single mode.
ADC_STADC_Trigger	Show how to trigger ADC by STADC pin.
ADC_TIMER_Trigger	Show how to trigger ADC by Timer.

Analog Comparator Controller (ACMP)

ACMP_CompareDAC	Demonstrate how ACMP compare DAC output with ACMP1_P1 value.
ACMP_CompareVBG	Demonstrate how ACMP compare VBG output with

	ACMP1_P1 value.
ACMP_Wakeup	Show how to wake up MCU from Power-down mode by ACMP wake-up function.
ACMP_WindowComapre	Demonstrate the usage of ACMP window compare function.

Controller Area Network (CAN)

CANFD_CAN_Loopback	Use CAN mode function to do internal loopback test.
CANFD_CAN_MonitorMode	Use CAN Monitor mode to listen to CAN bus communication test.
CANFD_CAN_TxRx	Transmit and receive CAN message through CAN interface.
CANFD_CAN_TxRxINT	An example of interrupt control using CAN bus communication.
CANFD_CANFD_Loopback	Use CAN FD mode function to do internal loopback test.
CANFD_CANFD_MonitorMode	Use CAN FD Monitor mode to listen to CAN bus communication test.
CANFD_CANFD_TxRx	Transmit and receive CAN FD message through CAN interface.
CANFD_CANFD_TxRxINT	An example of interrupt control using CAN FD bus communication.

LED Light Strip Interface (LLSI)

LLSI_Marquee	This is a LLSI demo for marquee display in software mode. It needs to be used with WS2812 LED strip.
LLSI_PDMA_Marquee	This is a LLSI demo for marquee display in PDMA mode. It needs to be used with WS2812 LED strip.
LLSI_Y_Cable_Control	his is a LLSI demo for Y-cable control. It needs to be used with AP6112Y LED strip.

6 REVISION HISTORY

Date	Revision	Description
2025.01.22	3.00.001	1. 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.