M253 Series CMSIS BSP Guide

Directory Introduction for 32-bit NuMicro® Family

Directory Information

Document	Driver reference guide and revision history.
Library	Driver header and source files.
SampleCode	Driver sample code.
ThirdParty	Libraries from third parties.

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

www.nuvoton.com

TABLE OF CONTENTS

1 Document

CMSIS.html	Document of CMSIS version 5.1.1.
NuMicro M253 Series CMSIS BSP Driver Reference Guide.chm	This document describes the usage of drivers in M253 BSP.
NuMicro M253 Series CMSIS BSP Revision History.pdf	This document shows the revision history of M253 BSP.

2 Library

CMSIS	Cortex® Microcontroller Software Interface Standard (CMSIS) V5.1.1 definitions by Arm® Corp.
Device	CMSIS compliant device header file.
StdDriver	All peripheral driver header and source files.

3 SampleCode

FreeRTOS	Simple FreeRTOS™ demo code.
Hard_Fault_Sample	Show hard fault information when hard fault happened. The hard fault handler shows some information including program counter, which is the address where the processor is executed when the hard fault occurs. The listing file (or map file) can show what function and instruction that is. 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.
PowerManagement	Power management sample code. For more information about M253 series power management, please refer to the application note.
Semihost	Show how to print and get character through IDE console window.
StdDriver	Sample code to demonstrate the usage of M253 series MCU peripheral driver APIs.
Template	A project template for M253 series MCU.
XOM	Demonstrate how to create XOM library and use it. For more information about M253 series XOM, please refer to the application note.

4 ThirdParty

FreeRTOS	A real time operating system available for free download. Its official website is: http://www.freertos.org/.

5 SampleCode\FreeRTOS

Blinky	This project provides two demo applications. A simple blinky style project, and a more comprehensive test and demo application.

6 SampleCode\ISP

ISP_CAN	In-System-Programming Sample code through CAN interface.
ISP_DFU	In-System-Programming Sample code through USB interface and following Device Firmware Upgrade Class Specification.
ISP_HID	In-System-Programming Sample code through USB HID interface.
ISP_I2C	In-System-Programming Sample code through I^2C interface.
ISP_MSC	In-System-Programming Sample code through USB interface and following Mass Storage Class Specification.
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.

7 SampleCode\PowerManagement

The M253 series MCU provides some power modes with different power consumption level and wake-up time. For more information, please refer to the application note.

SYS_PowerDownMode	Show how to enter a different Power-down mode and wake up by RTC.
SYS_PowerMode	Show how to set different core voltage and main voltage regulator type.

8 SampleCode\StdDriver

System Manager (SYS)

SYS_BODWakeup	Demonstrate how to wake up system from Power-down mode by brown-out detector interrupt.
SYS_TrimHIRC	Demonstrate how to use LXT to trim HIRC.
SYS_TrimMIRC	Demonstrate how to use Timer to trim MIRC.

Clock Controller (CLK)

CLK_ClockDetector	Demonstrate the usage of clock fail detector and clock frequency range detector function.

Flash Memory Controller (FMC)

FMC_CRC32	Demonstrate how to use FMC CRC32 ISP command to calculate the CRC32 checksum of APROM and LDROM.
FMC_ExeInSRAM	Implement a code and execute it in SRAM to program embedded Flash.
FMC_IAP	Demonstrate FMC IAP boot mode and show how to use vector remap function. LDROM image has been embedded in APROM image and programmed to LDROM Flash at run-time. This sample also shows how to branch between APROM and LDROM.
FMC_MultiBoot	Implement a multi-boot system to boot from different applications in APROM or LDROM by VECMAP.
FMC_MultiWordProgram	Show how to use FMC multi-word program ISP command to program APROM 0x18000~0x20000 area.
FMC_ReadAllOne	Demonstrate how to use FMC Read-All-One ISP command to verify APROM or LDROM pages are all 0xFFFFFFFF or not.
FMC_RW	Show FMC read Flash IDs, erase, read, and write functions.

FMC_XOM	Show how to configure and set up an XOM region then perform XOM function. For more information, please refer to the application note.

General Purpose I/O (GPIO)

GPIO_EINTAndDebounce	Show the usage of GPIO external interrupt function and de-bounce function.
GPIO_INT	Show the usage of GPIO interrupt function.
GPIO_OutputInput	Show how to set GPIO pin mode and use pin data input and output control.
GPIO_PowerDown	Show how to wake up system from Power-down mode by GPIO interrupt.
GPIO_SingleCycleIO	Show GPIO single cycle I/O bus performance.

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_CaptureCounter	Show how to use the Timer2 capture function to capture Timer2 counter value.
TIMER_Delay	Demonstrate the usage of TIMER_Delay API to generate a 1 second delay.

TIMER_EventCounter	Use pin PB.4 to demonstrate Timer event counter function.
TIMER_FreeCountingMode	Use the Timer pin to demonstrate Timer free counting mode function, and display the measured input frequency to UART console.
TIMER_InterTimerTriggerMode	Use the Timer pin to demonstrate inter-timer trigger mode function, and display the measured input frequency to UART console.
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	Demonstrate the Timer0 toggle out function on pin PB.5.

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.

Real Timer Clock (RTC)

RTC_Alarm_Test	Demonstrate the RTC alarm function. It sets an alarm 10 seconds after execution.
RTC_Alarm_Wakeup	Use RTC alarm interrupt event to wake up system.
RTC_Time_Display	Demonstrate the RTC function and display current time to the UART console.

Basic PWM Generator and Capture Timer (BPWM)

BPWM_Capture	Use BPWM0 channel 0 to capture the Timer0 waveform.
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.

UART Interface Controller (UART)

UART_AutoBaudRate	Show how to use auto baud rate detection function.
UART_AutoFlow	Transmit and receive data using auto flow control.
UART_IrDA	Transmit and receive UART data in UART IrDA mode.
UART_PDMA	Demonstrate UART transmit and receive function with PDMA.
UART_RS485	Transmit and receive data in UART RS485 mode.
UART_SingleWire	Transmit and receive data in 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_Flash	Access SPI Flash through SPI interface.
SPI_HalfDuplex	Demonstrate SPI half-duplex mode. Configure USPI0 as master mode and SPI0 as slave mode. Both USPI0 and SPI0 are half-duplex mode.

SPI_Loopback	A SPI read/write demo connecting SPI0 MISO and MOSI pins.
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 could work with SPI_SlaveFIFOMode sample code.
SPI_PDMA_LoopTest	Demonstrate SPI data transfer with PDMA. SPI0 will be configured as slave mode and USPI0 will be configured as master mode. Both Tx PDMA function and Rx PDMA function will be enabled.
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.
SPI_SlaveFIFOModeINT	Configure SPI0 as slave mode and demonstrate how to use FIFO mode to communicate with an off-chip SPI master device, transmit and receive data in the interrupt handler. This sample code needs to work with SPI_MasterFIFOMode sample code.
SPII2S_Master	Configure SPI0 as I²S master mode and demonstrate how I²S works in master mode. This sample code needs to work with SPII2S_Slave sample code.
SPII2S_PDMA_Codec	An I²S demo with PDMA function connected with audio codec.
SPII2S_PDMA_Play	An I²S demo for playing data and demonstrating how I²S works with PDMA.
SPII2S_PDMA_PlayRecord	An I²S demo for playing and recording data with PDMA function.
SPII2S_PDMA_Record	An I²S demo for recording data and demonstrating how I²S works with PDMA.
SPII2S_Slave	Configure SPI0 as I²S slave mode and demonstrate how I²S works in slave mode. This sample code needs to work with SPII2S_Master sample code.

I²C Serial Interface Controller (I²C)

I2C_EEPROM	Read and write EEPROM via I²C interface.
I2C_GCMode_Master	Demonstrate how a master uses I²C address 0x0 to write data to I²C slave. This sample code needs to work with I2C_GCMode_Slave sample code.
I2C_GCMode_Slave	Demonstrate how to receive master data in GC (General Call) mode. This sample code needs to work with I2C_GCMode_Master sample code.
I2C_Loopback	Show how an I²C master accesses 7-bit address slave via loopback of 2 I²C ports.
I2C_Loopback_10bit	Show how an I²C master accesses 10-bit address slave via loopback of 2 I²C ports.
I2C_Master	Show how an I²C master accesses 7-bit address slave. This sample code needs to work with I2C_Slave sample code.
I2C_Master_10bit	Show how an I²C master accesses 10-bit address slave. This sample code needs to work with I2C_Slave_10bit 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_PDMA_TRX	Demonstrate I²C PDMA mode, which need to connect I²C0 (master) and I²C1 (slave).
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	Show how an I²C 7-bit address slave receives data from master. This sample code needs to work with I2C_Master sample code.
I2C_Slave_10bit	Show how an I²C 10-bit address slave receives data from master. This sample code needs to work with I2C_Master_10bit sample code.

I2C_Wakeup_Slave	Demonstrate how to set I^{2}C to wake up MCU from Power-down mode. This sample code could 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_Master	Transmit and receive data with auto flow control. This sample code needs to work with USCI_UART_Autoflow_Slave sample code.
USCI_UART_Autoflow_Slave	Transmit and receive data with auto flow control. This sample code needs to work with USCI_UART_Autoflow_Master sample code.
USCI_UART_PDMA	This is a USCI_UART PDMA demo and needs to connect USCI_UART Tx and Rx.
USCI_UART_RS485_Master	Transmit and receive data in RS485 mode. This sample code needs to work with USCI_UART_RS485_Slave sample code.
USCI_UART_RS485_Slave	Transmit and receive data in RS485 mode. This sample code needs to work with USCI_UART_RS485_Master sample code.
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_SPI0 master loop back transfer. This sample code needs to connect USCI_SPI0_MISO pin and USCI_SPI0_MOSI pin together. It will compare the received data with transmitted data.
USCI_SPI_MasterMode	Configure USCI_SPI0 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_PDMA_LoopTest	Demonstrate 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.
USCI_SPI_SlaveMode	Configure USCI_SPI0 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_SlaveModeINT	Configure USCI_SPI0 as slave mode and demonstrate how to communicate with an off-chip SPI master device, transmit and receive data in the interrupt handler. This sample code needs to work with USCI_SPI_MasterMode sample code.

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

USCI_I2C_EEPROM	Show how to use USCI_I2C interface to access EEPROM.
USCI_I2C_Master	Show how an I²C master accesses 7-bit address slave. This sample code needs to work with USCI_I2C_Slave sample code.
USCI_I2C_Master_10bit	Show how an I²C master accesses 10-bit address slave. This sample code needs to work with USCI_I2C_Slave_10bit sample code.
USCI_I2C_Monitor	Use USCI_I2C to monitor and log I2C bus traffic.
USCI_I2C_MultiBytes_Master	Use UI2C multiple-byte functions to read and write data to slave. Need to work with the USCI_I2C_Slave sample code.
USCI_I2C_SingleByte_Master	Use UI2C single-byte functions to read and write data to slave. Need to work with the USCI_I2C_Slave sample code.
USCI_I2C_Slave	Show how an I²C 7-bit address slave receives data from

	master.
USCI_I2C_Slave_10bit	Show how an I²C 10-bit address slave receives data from master. This sample code needs to work with USCI_I2C_Master_10bit sample code.
USCI_I2C_Wakeup_Slave	Demonstrate how to set I²C to wake up MCU from Power-down mode. This sample code needs to work with USCI_I2C_Master sample code.

USB 2.0 Full-Speed Device Controller (USBD)

USBD_Audio_Codec	Demonstrate how to implement a USB audio class device.
USBD_HID_Keyboard	Demonstrate how to implement a USB keyboard device. This sample code supports to use GPIO to simulate key input.
USBD_HID_Mouse	Simulate a USB mouse and draws circle on the screen.
USBD_HID_MouseKeyboard	Simulate an USB HID mouse and HID keyboard. Mouse draws circle on the screen and Keyboard uses GPIO to simulate key input.
USBD_HID_RemoteWakeup	Simulate a HID mouse supporting USB suspend and remote wakeup.
USBD_HID_Transfer	Demonstrate how to transfer data between a USB device and PC through a USB HID interface. A windows tool is also included in this sample code to connect with a USB device.
USBD_HID_Transfer_And_ Keyboard	Demonstrate how to implement a composite device of HID transfer and keyboard. Transfer data between USB device and PC through USB HID interface. A windows tool is also included in this sample code to connect with a USB device.
USBD_HID_Transfer_And_ MSC	Demonstrate how to implement a composite device of HID transfer and mass storage. Transfer data between USB device and PC through USB HID interface. A windows tool is also included in this sample code to

	connect with a USB device.
USBD_HID_Transfer_CTRL	Use USB host core driver and HID driver. It shows how to submit HID class request and how to read data from control pipe. A windows tool is also included in this sample code to connect with a USB device.
USBD_Mass_Storage_CDROM	Demonstrate the emulation of USB mass storage device, CD-ROM.
USBD_Mass_Storage_Flash	Use internal Flash as backend storage media to simulate a USB pen drive.
USBD_Micro_Printer	Demonstrate how to implement a USB micro printer device.
USBD_Printer_And_HID_ Transfer	Demonstrate how to implement a composite device of USB micro printer and HID transfer. Transfer data between USB device and PC through USB HID interface. A windows tool is also included in this sample code to connect with a USB device.
USBD_VCOM_And_HID_ Keyboard	Demonstrate how to implement a composite device of VCOM and HID keyboard.
USBD_VCOM_And_HID_ Transfer	Demonstrate how to implement a composite device of VCOM and HID transfer. Transfer data between USB device and PC through USB HID interface. A windows tool is also included in this sample code to connect with a USB device.
USBD_VCOM_And_Mass_ Storage	Demonstrate how to implement a composite device of VCOM and mass storage.
USBD_VCOM_DualPort	Demonstrate how to implement a USB dual virtual COM port device.
USBD_VCOM_MultiPort	Demonstrate how to implement a USB multi virtual COM ports device.
USBD_VCOM_MultiPort_CMD	Demonstrate how to implement a USB multi virtual COM ports device with a terminal echo port.
USBD_VCOM_SerialEmulator	Demonstrate how to implement a USB virtual COM port

	device.

Controller Area Network with Flexible Data Rate (CAN FD)

CANFD_CAN_Loopback	Use CAN mode function to do internal loopback test.
CANFD_CAN_TxRx	Transmit and receive CAN messages 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_TxRx	Transmit and receive CAN FD messages through CAN interface.
CANFD_CANFD_TxRxINT	An example of interrupt control using CAN FD bus communication.

Enhance 12-bit Analog-to-Digital Converter (EADC)

EADC_Accumulate	Demonstrate how to get accumulated conversion result.
EADC_ADINT_Trigger	Use ADINT interrupt to trigger the EADC conversion.
EADC_Average	Demonstrate how to get average conversion result.
EADC_BandGap	Convert band-gap (Sample module 16) and print conversion result.
EADC_BandGapCalculateAVDD	Demonstrate how to calculate battery voltage (AV_{DD}) by using band-gap.
EADC_BPWM_Trigger	Demonstrate how to trigger EADC by BPWM.
EADC_PDMA_BPWM_Trigger	Demonstrate how to trigger EADC by BPWM and transfer conversion data by PDMA.
EADC_Pending_Priority	Demonstrate how to trigger multiple sample modules and got conversion results in order of priority.
EADC_ResultMonitor	Monitor the conversion result of channel 2 by the digital

	compare function.
EADC_SWTRG_Trigger	Trigger EADC by writing EADC_SWTRG register.
EADC_TempSensor	Convert temperature sensor (Sample module 17) and print conversion result.
EADC_Timer_Trigger	Show how to trigger EADC by Timer.

CRC Controller (CRC)

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

9 SampleCode\XOM

In the M253 series MCU, XOM (Execute-Only Memory) is a secure ROM region which forbids any data access. However, the code stored in the XOM region could still be executed by CPU since it is accessed by instruction fetch. For more information, please refer to the application note.

XOMLib	Demonstrate how to create XOM library.
XOMLibDemo	Demonstrate how to use XOMLib.

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.