M091 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.

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 M091 Driver Reference Guide.chm	This document describes the usage of drivers in M091 BSP.
NuMicro M091 Series CMSIS BSP Revision History.pdf	This document shows the revision history of M091 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

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 occurs. 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.
Semihost	Show how to print and get character through IDE console window.
StdDriver	Sample code to demonstrate the usage of M091 series MCU peripheral driver APIs.
Template	A project template for M091 series MCU.

4 SampleCode\ISP

ISP_I2C	In-System-Programming Sample code through I^{2}C 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	Demonstrate how to wake up system from 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.

Thermal Sensor (TS)

TS_TemperatureMeasure	Show how Thermal Sensor measure the current temperature by interrupt mechanism.

Flash Memory Controller (FMC)

FMC_CRC32	Demonstrate how to use FMC CRC32 ISP command to calculate the CRC32 checksum of APROM and LDROM.
FMC_IAP	Demonstrate FMC IAP boot mode and show how to use vector remap function. LDROM image was embedded in APROM image and be programmed to LDROM Flash at run-time. This sample also shows how to branch between APROM and LDROM.
FMC_RW	Show FMC read Flash IDs, erase, read, and write functions.

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.

PDMA Controller (PDMA)

PDMA_BasicMode	Use PDMA channel 1 to transfer data from memory to memory.
PDMA_ScatterGather	Use PDMA channel 1 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 Timer capture function to capture Timer counter value.
TIMER_Delay	Demonstrate the usage of TIMER_Delay API to generate a 1 second delay.
TIMER_EventCounter	Use TM0 pin to demonstrate Timer event counter function.
TIMER_FreeCountingMode	Use the timer TM0_EXT pin to demonstrate timer free counting mode function. And displays the measured input frequency to UART console.
TIMER_InterTimerTriggerMode	Use the timer TM0 pin to demonstrate inter timer trigger mode function. Also 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 timer to wake up system from Power-down mode periodically.
TIMER_ToggleOut	Demonstrate the Timer0 toggle out function on TM0 pin.

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_ReloadCounter	Show how to reload the WWDT counter value.

Basic PWM Generator and Capture Timer (BPWM)

BPWM_Capture	Capture the BPWM Channel 2 waveform by BPWM Channel 0.
BPWM_DoubleBuffer	Change duty cycle and period of output waveform by BPWM double buffer function.
BPWM_DutySwitch	Change duty cycle of output waveform by configured period.
BPWM_OutputWaveform	Demonstrate how to use BPWM counter output waveform.

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_Loopback	SPI read/write demo connecting SPI MISO and MOSI pins.
SPI_MasterFIFOMode	Configure SPI 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.
SPI_PDMA_LoopTest	SPI read/write demo in PDMA mode. Connecting SPI MISO and MOSI pins. Both TX PDMA function and RX PDMA function will be enabled.
SPI_SlaveFIFOMode	Configure SPI 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.

I²C Serial Interface Controller (I²C)

I2C_Double_Buffer_Slave	Demonstrate how to set I²C two-level buffer in Slave mode to receive 256 bytes data from a master. This sample code needs to work with I2C_MultiBytes_Master.
I2C_EEPROM	Show how to use I²C interface to access EEPROM.
I2C_GCMode_Master	Show how a master uses I²C address 0x0 to write data to a slave. This sample code needs to work with I2C_GCMode_Slave.
I2C_GCMode_Slave	Show how a slave receives data from a master in GC (General Call) mode. This sample code needs to work with I2C_GCMode_Master.
I2C_Loopback	Demonstrate how to set I²C Master mode and Slave Mode, and show how a master accesses a slave on a chip.

I2C_Master	Show how a master accesses a slave. This sample code needs to work with I2C_Slave.
I2C_MultiBytes_Master	Show how to set I^{2}C Multi bytes API Read and Write data to Slave. This sample code needs to work with I2C_Slave.
I2C_PDMA_TRX	Demonstrate I^{2}C PDMA mode and need to connect I2C0 (master) and I2C1 (slave).
I2C_SingleByte_Master	Show how to use I^{2}C Single byte API Read and Write data to Slave. This sample code needs to work with I2C_Slave.
I2C_Slave	Demonstrate how to set I^{2}C in Slave mode to receive 256 bytes data from a master. This sample code needs to work with I2C_Master.
I2C_Wakeup_Slave	Show how to wake up MCU from Power-down mode via the I^{2}C interface. This sample code needs to work with I2C_Master.

CRC Controller (CRC)

CRC_CCITT	Implement CRC in CRC-CCITT mode and get the CRC checksum result.
CRC_CRC32_PDMA	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.

Analog-to-Digital Converter (ADC)

ADC_1411ksps_ContinuousScan Mode	Demonstrate how to use HIRC as ADC clock source to achieve 1411 ksps ADC conversion rate.
ADC_ADINT_Trigger	Use ADINT interrupt to do the ADC Single-cycle scan conversion.
ADC_BandGap	Convert Band-gap (channel 29) and print conversion result.

ADC_BandGapCalculateAVDD	Demonstrate how to calculate battery voltage(AVdd) by using band-gap.
ADC_BurstMode	Perform A/D Conversion with ADC burst mode.
ADC_ContinuousScanMode	Perform A/D Conversion with ADC continuous scan mode.
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_SwTrg_Trigger	Trigger ADC by writing ADC software trigger register.
ADC_Timer_Trigger	Show how to trigger ADC by Timer.

Digital-to-Analog Converter (DAC)

DAC_PDMA_TimerTrigger	Show how Timer triggers DAC to fetch data with PDMA and convert sine wave outputs.
DAC_SoftwareTrigger	Demonstrate how software triggers DAC to convert sine wave outputs.
DAC_TimerTrigger	Demonstrate how Timer triggers DAC to convert sine wave outputs.

Manchester Codec Controller (MANCH)

MANCH_RawSRAM2MTX	Show how to encode SRAM raw data to MTX and then move MTX content to another SRAM area by PDMA.
MANCH_RX	Show how to receive data from RX pin by PDMA.
MANCH_TX	Show how to send data from TX pin by PDMA.
MANCH_TX2DACWithAmplitude	Demonstrate how to use MANCH, TIMER, PDMA and

Modulation	DAC to generate MANCH signal with amplitude modulation.
MANCH_TX2DACWithFrequency Modulation	Demonstrate how to use MANCH and DAC0 to generate MANCH signal with frequency modulation.
MANCH_TXRXLoopback	Show how to send and receive data from TX and RX pin, respectively.

Operational Amplifier (OPA)

OPA_Amplify	Show how to use OPA to amplify input signals, and output signals converted by ADC channels.

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.