M471M/R1/S 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	Library from third party.

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

TABLE OF CONTENTS

1 Document

CMSIS.html	Document of CMSIS version 4.5.0
NuMicro M471M_R1_S CMSIS BSP Revision History.chm	This document shows the revision history of M471M/R1/S BSP.
NuMicro M471M_R1_S Driver Reference Guide.html	This document describes the usage of drivers in M471M/R1/S BSP.

2   Library

CMSIS	Cortex® Microcontroller Software Interface Standard (CMSIS) V4.5.0 definitions by Arm® Corp.
Device	CMSIS compliant device header file.
SmartcardLib	Smart card library binary and header file.
StdDriver	All peripheral driver header and source files.

3 SampleCode

CortexM4	Cortex®-M4 sample code.
FreeRTOS	Simple FreeRTOS™ demo 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.
Semihost	Show how to print and get character through IDE console window.
StdDriver	Sample code to demonstrate the usage of M471M/R1/S series MCU peripheral driver APIs.
Template	A project template for M471M/R1/S series MCU.

4   SampleCode\CortexM4

BitBand	Demonstrate the usage of Cortex®-M4 Bit-band.
DSP_FFT	Demonstrate how to call ARM CMSIS DSP library to calculate FFT.
MPU	Demonstrate the usage of Cortex®-M4 MPU.

5 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.

6 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.

Clock Controller (CLK)

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

Flash Memory Controller (FMC)

FMC_ExeInSRAM	Implement a code and execute in SRAM to program embedded Flash.
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_MultiBoot	Implement a multi-boot system to boot from different applications in APROM or LDROM by VECMAP.
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 2 to transfer data from memory to memory.
PDMA_ScatterGather	Use PDMA channel 5 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	Show how to use Timer0 to create various delay time.
TIMER_EventCounter	Implement Timer1 event counter function to count the external input event.
TIMER_PeriodicINT	Implement Timer counting in periodic mode.
TIMER_TimeoutWakeup	Use Timer0 periodic time-out interrupt event to wake up system.

Watchdog Timer (WDT)

WDT_TimeoutWakeup AndReset	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_AlarmWakeup	Use RTC alarm interrupt event to wake up system.
RTC_SpareRegisterRW	Show how to access RTC spare registers in supported chip.
RTC_TimeAndTick	Demonstrate the RTC function and display current time to the UART console.

PWM Generator and Capture Timer (PWM)

PWM_Capture	Capture the PWM1 channel 0 waveform by PWM1 channel 2.
PWM_DeadZone	Demonstrate how to use PWM dead-time insertion function.
PWM_DoubleBuffer	Change duty cycle and period of output waveform by PWM double buffer function.
PWM_PDMA_Capture	Capture the PWM1 channel 0 waveform by PWM1 channel 2, and use PDMA to transfer captured data.

UART Interface Controller (UART)

UART_AutoBaudRate_ Master	Show how to use auto baud rate detection function. This sample code needs to work with UART_AutoBaudRate_Slave.
UART_AutoBaudRate_ Slave	Show how to use auto baud rate detection function. This sample code needs to work with UART_AutoBaudRate_Master.
UART_AutoFlow_Master	Transmit and receive data with auto flow control. This sample code needs to work with UART_Autoflow_Slave.
UART_AutoFlow_Slave	Transmit and receive data with auto flow control. This sample code needs to work with UART_Autoflow_Master.
UART_IrDA_Master	Transmit and receive data in UART IrDA mode. This sample code needs to work with UART_IrDA_Slave.
UART_IrDA_Slave	Transmit and receive data in UART IrDA mode. This sample code needs to work with UART_IrDA_Master.
UART_PDMA	Demonstrate UART transmit and receive function with PDMA.

UART_RS485_Master	Transmit and receive data in UART RS485 mode. This sample code needs to work with UART_RS485_Slave.
UART_RS485_Slave	Transmit and receive data in UART RS485 mode. This sample code needs to work with UART_RS485_Master.
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.

Smart Card Host Interface (SC)

SC_ReadATR	Read the smartcard ATR from Smartcard interface.
SCUART_TxRx	Demonstrate Smartcard UART mode by connecting PA.0 and PA.1 pins.

Serial Peripheral Interface (SPI)

SPI_Loopback	A SPI read/write demo connecting SPI0 MISO and MOSI pins.
SPI_MasterMode	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_SlaveMode sample code.
SPI_PDMA_LoopTest	Demonstrate SPI data transfer with PDMA. SPI0 will be configured as slave mode and QSPI0 will be configured as master mode. Both Tx PDMA function and Rx PDMA function will be enabled.
SPI_SlaveMode	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_MasterMode sample code.

I²C Serial Interface Controller (I²C)

I2C_EEPROM	Read and write EEPROM via I²C interface.

I2C_GCMode_Master	Demonstrate how a master uses I^{2}C address 0x0 to write data to I^{2}C 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_Master	An I²C master mode demo code. This sample code needs to work with I2C_Slave sample code.
I2C_Slave	An I²C slave mode demo code. This sample code needs to work with I2C_Master sample code.
I2C_SMBus	Show how to control SMBus interface and use SMBus protocol between host and slave.
I2C_Wakeup_Master	Show how to wake up MCU from Power-down. This sample code needs to work with I2C_Wakeup_Slave.
I2C_Wakeup_Slave	Show how to wake up MCU from Power-down mode through I^{2}C interface. This sample code needs to work with I2C_Wakeup_Master.

External Bus Interface (EBI)

EBI_NOR	Configure EBI interface to access NOR Flash connected on EBI interface.
EBI_SRAM	Configure EBI interface to access SRAM connected on EBI interface.

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.

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_PDMA_PWM_ Trigger	Demonstrate how to trigger EADC by PWM 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_PWM_Trigger	Demonstrate how to trigger EADC by PWM.
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_Timer_Trigger	Show how to trigger EADC by Timer.

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.