M071M 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

NuMicro M071M Series Driver Reference Guide.chm	Describe the definition, input and output of each API.
Revision History.pdf	Show all the revision history about specific BSP.

2 Library

CMSIS	CMSIS 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.
ISP	Sample codes for In-System-Programming.
RegBased	The sample code able to access control registers directly.
Semihost	Show how to debug with semi-host message print.
StdDriver	Driver Samples.
Template	Software Development Template.

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\RegBased

System Manager (SYS)

SYS_PLLClockOutput	Change system clock to different PLL frequency and output system clock from CLKO pin.

Flash Memory Controller (FMC)

FMC_IAP	Show how to call LDROM functions from APROM. The code in APROM will look up the table at 0x100E00 to get the address of function of LDROM and call the function.
FMC_MultiBoot	Implement a multi-boot system to boot from different applications in APROM or LDROM by VECMAP.
FMC_RW	Show how to read/program embedded flash by ISP function.

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.

Timer Controller (TIMER)

TIMER_Capture	Show how to use the timer0 capture function to capture timer0 counter value
TIMER_ Counter	Implement timer0 event counter function to count the external input event.
TIMER_Delay	Show how to use timer0 to create various delay time.

TIMER_PeriodicINT	Implement Timer counting in periodic mode.
TIMER_ PowerDown	Use timer0 periodic time-out interrupt event to wake up system.

Watchdog Timer (WDT)

WDT_PowerDown	Use WDT time-out interrupt event to wake-up system.
WDT_TimeoutINT	Implement periodic WDT time-out interrupt event.
WDT_TimeoutReset	Show how to 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_Capture	Capture the PWM1 Channel 0 waveform by PWM0 Channel 0.
PWM_DeadZone	Demonstrate how to use PWM Dead Zone function.
PWM_DoubleBuffer	Change duty cycle and period of output waveform by PWM Double Buffer function.

UART Interface Controller (UART)

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_LIN	Transmit LIN header and response.
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_TxRx_Function	Transmit and receive data from PC terminal through RS232 interface.
UART_Wakeup	Show how to wake up system form Power-down mode by UART interrupt.

Serial Peripheral Interface (SPI)

SPI_Loopback	Implement SPI Master loop back transfer. This sample code needs to connect SPI0_MISO0 pin and SPI0_MOSI0 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. 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. This sample code needs to work with SPI_MasterFifoMode sample code.

I²C Serial Interface Controller (I²C)

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 Slave. This sample code needs to work with I2C_GCMode_Slave.

I2C_GCMode_Slave	Show a Slave how to receive data from Master in GC (General Call) mode. This sample code needs to work with I2C_GCMode_Master.
I2C_Master	Show a Master how to access Slave. This sample code needs to work with I2C_Slave.
I2C_Slave	Show how to set I^2C in Slave mode and receive the data from Master. This sample code needs to work with I2C_Master.
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^2C interface. This sample code needs to work with I2C_Wakeup_Master.

Analog-to-Digital Converter (ADC)

ADC_ContinuousScanMode	Perform A/D Conversion with ADC continuous scan mode.
ADC_MeasureAVDD	Measure AVDD voltage by ADC.
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.

Controller Area Network (CAN)

CAN_Set_MaskFilter	Use MaskFilter to receive message in Normal mode. This sample code needs to work with CAN_Test_MaskFilter.
CAN_Test_MaskFilter	Use message object No.1 to send message objects (ID=0x700~0x70F). This sample code needs to work

	with CAN_Set_MaskFilter.

6 SampleCode\StdDriver

System Manager (SYS)

SYS_PLLClockOutput	Change system clock to different PLL frequency and output system clock from CLKO pin.

Flash Memory Controller (FMC)

FMC_IAP	Show how to call LDROM functions from APROM. The code in APROM will look up the table at 0x100E00 to get the address of function of LDROM and call the function.
FMC_RW	Show how to read/program embedded flash by ISP function.

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.

Timer Controller (TIMER)

TIMER_Capture	Show how to use the timer0 capture function to capture timer0 counter value
TIMER_ Counter	Implement timer0 event counter function to count the external input event.
TIMER_Delay	Show how to use timer0 to create various delay time.
TIMER_PeriodicINT	Implement Timer counting in periodic mode.

TIMER_ PowerDown	Use timer0 periodic time-out interrupt event to wake up system.

Watchdog Timer (WDT)

WDT_PowerDown	Use WDT time-out interrupt event to wake-up system.
WDT_TimeoutINT	Implement periodic WDT time-out interrupt event.
WDT_TimeoutReset	Show how to 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_Capture	Capture the PWM1 Channel 0 waveform by PWM0 Channel 0.
PWM_DeadZone	Demonstrate how to use PWM Dead Zone function.
PWM_DoubleBuffer	Change duty cycle and period of output waveform by PWM Double Buffer function.

UART Interface Controller (UART)

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_LIN	Transmit LIN header and response.
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_TxRx_Function	Transmit and receive data from PC terminal through RS232 interface.
UART_Wakeup	Show how to wake up system form Power-down mode by UART interrupt.

Serial Peripheral Interface (SPI)

SPI_Loopback	Implement SPI Master loop back transfer. This sample code needs to connect SPI0_MISO0 pin and SPI0_MOSI0 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. 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. This sample code needs to work with SPI_MasterFifoMode sample code.

I²C Serial Interface Controller (I²C)

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 Slave. This sample code needs to work with I2C_GCMode_Slave.
I2C_GCMode_Slave	Show a Slave how to receive data from Master in GC (General Call) mode. This sample code needs to work with I2C_GCMode_Master.

I2C_Master	Show a Master how to access Slave. This sample code needs to work with I2C_Slave.
I2C_Slave	Show how to set I^2C in Slave mode and receive the data from Master. This sample code needs to work with I2C_Master.
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^2C interface. This sample code needs to work with I2C_Wakeup_Master.

Analog-to-Digital Converter (ADC)

ADC_ContinuousScanMode	Perform A/D Conversion with ADC continuous scan mode.
ADC_MeasureAVDD	Measure AVDD voltage by ADC.
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.

Controller Area Network (CAN)

CAN_BasicMode_Receive	Implement receive message in Basic mode. This sample code needs to work with CAN_BasicMode_Transmit.
CAN_BasicMode_Transmit	Implement transmit message in Basic mode. This sample code needs to work with CAN_BasicMode_Receive.
CAN_NormalMode_Receive	Implement receive message in Normal mode. This sample code needs to work with

	CAN_NormalMode_Transmit.
CAN_NormalMode_Transmit	Implement transmit message in Normal mode. This sample code needs to work with CAN_NormalMode_Receive.
CAN_Wakeup	Show how to wake up system form Power-down mode by detecting a transition.

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.