

## 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.

*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](http://www.nuvoton.com)

# TABLE OF CONTENTS

<b>1</b>	<b>DOCUMENT.....</b>	<b>3</b>
<b>2</b>	<b>LIBRARY .....</b>	<b>4</b>
<b>3</b>	<b>SAMPLECODE .....</b>	<b>5</b>
<b>4</b>	<b>SAMPLECODE\ISP .....</b>	<b>6</b>
<b>5</b>	<b>SAMPLECODE\REGBASED .....</b>	<b>7</b>
	System Manager (SYS) .....	7
	Flash Memory Controller (FMC).....	7
	General Purpose I/O (GPIO).....	7
	Timer Controller (TIMER).....	7
	Watchdog Timer (WDT) .....	8
	Window Watchdog Timer (WWDT) .....	8
	PWM Generator and Capture Timer (PWM).....	8
	UART Interface Controller (UART).....	8
	Serial Peripheral Interface (SPI).....	9
	I <sup>2</sup> C Serial Interface Controller (I <sup>2</sup> C) .....	9
	Analog-to-Digital Converter (ADC) .....	10
<b>6</b>	<b>SAMPLECODE\STDDRIVER.....</b>	<b>11</b>
	System Manager (SYS) .....	11
	Flash Memory Controller (FMC).....	11
	General Purpose I/O (GPIO).....	11
	Timer Controller (TIMER).....	11
	Watchdog Timer (WDT) .....	12
	Window Watchdog Timer (WWDT) .....	12
	PWM Generator and Capture Timer (PWM).....	12
	UART Interface Controller (UART).....	12
	Serial Peripheral Interface (SPI).....	13
	I <sup>2</sup> C Serial Interface Controller (I <sup>2</sup> C) .....	13
	Analog-to-Digital Converter (ADC) .....	14

## 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

<b>Hard_Fault_Sample</b>	Show hard fault information when hard fault happened.
<b>ISP</b>	Sample codes for In-System-Programming.
<b>RegBased</b>	The sample code able to access control registers directly.
<b>Semihost</b>	Show how to debug with semi-host message print.
<b>StdDriver</b>	Driver Samples.
<b>Template</b>	Software Development Template.

## 4 SampleCode\ISP

ISP_I2C	In-System-Programming Sample code through I <sup>2</sup> 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\RegBased

### System Manager (SYS)

<b>SYS_PLLClockOutput</b>	Change system clock to different PLL frequency and output system clock from CLK0 pin.
<b>SYS_PowerDown_MinCurrent</b>	Demonstrate how to minimize power consumption when entering power down mode.

### Flash Memory Controller (FMC)

<b>FMC_IAP</b>	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.
<b>FMC_MultiBoot</b>	Implement a multi-boot system to boot from different applications in APROM or LDROM by VECMAP.
<b>FMC_RW</b>	Show how to read/program embedded flash by ISP function.

### General Purpose I/O (GPIO)

<b>GPIO_EINTAndDebounce</b>	Show the usage of GPIO external interrupt function and de-bounce function.
<b>GPIO_INT</b>	Show the usage of GPIO interrupt function.
<b>GPIO_OutputInput</b>	Show how to set GPIO pin mode and use pin data input and output control.
<b>GPIO_PowerDown</b>	Show how to wake up system from Power-down mode by GPIO interrupt.

### Timer Controller (TIMER)

<b>TIMER_Capture</b>	Show how to use the timer0 capture function to capture timer0 counter value
<b>TIMER_Counter</b>	Implement timer0 event counter function to count the

	external input event.
<b>TIMER_Delay</b>	Show how to use timer0 to create various delay time.
<b>TIMER_PeriodicINT</b>	Implement Timer counting in periodic mode.
<b>TIMER_PowerDown</b>	Use timer0 periodic time-out interrupt event to wake up system.

### Watchdog Timer (WDT)

<b>WDT_PowerDown</b>	Use WDT time-out interrupt event to wake-up system.
<b>WDT_TimeoutINT</b>	Implement periodic WDT time-out interrupt event.
<b>WDT_TimeoutReset</b>	Show how to generate time-out reset system event while WDT time-out reset delay period expired.

### Window Watchdog Timer (WWDT)

<b>WWDT_CompareINT</b>	Show how to reload the WWDT counter value.
------------------------	--

### PWM Generator and Capture Timer (PWM)

<b>PWM_Capture</b>	Capture the PWM1 Channel 0 waveform by PWM0 Channel 0.
<b>PWM_DeadZone</b>	Demonstrate how to use PWM Dead Zone function.
<b>PWM_DoubleBuffer</b>	Change duty cycle and period of output waveform by PWM Double Buffer function.

### UART Interface Controller (UART)

<b>UART_Autoflow_Master</b>	Transmit and receive data with auto flow control. This sample code needs to work with UART_Autoflow_Slave.
<b>UART_Autoflow_Slave</b>	Transmit and receive data with auto flow control. This sample code needs to work with UART_Autoflow_Master.



<b>UART_IrDA_Master</b>	Transmit and receive data in UART IrDA mode. This sample code needs to work with UART_IrDA_Slave.
<b>UART_IrDA_Slave</b>	Transmit and receive data in UART IrDA mode. This sample code needs to work with UART_IrDA_Master.
<b>UART_LIN</b>	Transmit LIN header and response.
<b>UART_RS485_Master</b>	Transmit and receive data in UART RS485 mode. This sample code needs to work with UART_RS485_Slave.
<b>UART_RS485_Slave</b>	Transmit and receive data in UART RS485 mode. This sample code needs to work with UART_RS485_Master.
<b>UART_TxRx_Function</b>	Transmit and receive data from PC terminal through RS232 interface.
<b>UART_Wakeup</b>	Show how to wake up system form Power-down mode by UART interrupt.

## Serial Peripheral Interface (SPI)

<b>SPI_Loopback</b>	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.
<b>SPI_MasterFifoMode</b>	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.
<b>SPI_SlaveFifoMode</b>	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.

## I<sup>2</sup>C Serial Interface Controller (I<sup>2</sup>C)

<b>I2C_EEPROM</b>	Show how to use I <sup>2</sup> C interface to access EEPROM.
<b>I2C_GCMode_Master</b>	Show how a Master uses I <sup>2</sup> C address 0x0 to write data to Slave. This sample code needs to work with

	I2C_GCMode_Slave.
<b>I2C_GCMode_Slave</b>	Show a Slave how to receive data from Master in GC (General Call) mode. This sample code needs to work with I2C_GCMode_Master.
<b>I2C_Master</b>	Show a Master how to access Slave. This sample code needs to work with I2C_Slave.
<b>I2C_Slave</b>	Show how to set I <sup>2</sup> C in Slave mode and receive the data from Master. This sample code needs to work with I2C_Master.
<b>I2C_Wakeup_Master</b>	Show how to wake up MCU from Power-down. This sample code needs to work with I2C_Wakeup_Slave.
<b>I2C_Wakeup_Slave</b>	Show how to wake up MCU from Power-down mode through I <sup>2</sup> C interface. This sample code needs to work with I2C_Wakeup_Master.

## Analog-to-Digital Converter (ADC)

<b>ADC_ContinuousScanMode</b>	Perform A/D Conversion with ADC continuous scan mode.
<b>ADC_MeasureAVDD</b>	Measure AVDD voltage by ADC.
<b>ADC_PwmTrigger</b>	Demonstrate how to trigger ADC by PWM.
<b>ADC_ResultMonitor</b>	Monitor the conversion result of channel 2 by the digital compare function.
<b>ADC_SingleCycleScanMode</b>	Perform A/D Conversion with ADC single cycle scan mode.
<b>ADC_SingleMode</b>	Perform A/D Conversion with ADC single mode.

## 6 SampleCode\StdDriver

### System Manager (SYS)

<b>SYS_PLLClockOutput</b>	Change system clock to different PLL frequency and output system clock from CLKO pin.
<b>SYS_PowerDown_MinCurrent</b>	Demonstrate how to minimize power consumption when entering power down mode.

### Flash Memory Controller (FMC)

<b>FMC_IAP</b>	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.
<b>FMC_RW</b>	Show how to read/program embedded flash by ISP function.

### General Purpose I/O (GPIO)

<b>GPIO_EINTAndDebounce</b>	Show the usage of GPIO external interrupt function and de-bounce function.
<b>GPIO_INT</b>	Show the usage of GPIO interrupt function.
<b>GPIO_OutputInput</b>	Show how to set GPIO pin mode and use pin data input and output control.
<b>GPIO_PowerDown</b>	Show how to wake up system from Power-down mode by GPIO interrupt.

### Timer Controller (TIMER)

<b>TIMER_Capture</b>	Show how to use the timer0 capture function to capture timer0 counter value
<b>TIMER_Counter</b>	Implement timer0 event counter function to count the external input event.
<b>TIMER_Delay</b>	Show how to use timer0 to create various delay time.

<b>TIMER_PeriodicINT</b>	Implement Timer counting in periodic mode.
<b>TIMER_PowerDown</b>	Use timer0 periodic time-out interrupt event to wake up system.

### Watchdog Timer (WDT)

<b>WDT_PowerDown</b>	Use WDT time-out interrupt event to wake-up system.
<b>WDT_TimeoutINT</b>	Implement periodic WDT time-out interrupt event.
<b>WDT_TimeoutReset</b>	Show how to generate time-out reset system event while WDT time-out reset delay period expired.

### Window Watchdog Timer (WWDT)

<b>WWDT_CompareINT</b>	Show how to reload the WWDT counter value.
------------------------	--

### PWM Generator and Capture Timer (PWM)

<b>PWM_Capture</b>	Capture the PWM1 Channel 0 waveform by PWM0 Channel 0.
<b>PWM_DeadZone</b>	Demonstrate how to use PWM Dead Zone function.
<b>PWM_DoubleBuffer</b>	Change duty cycle and period of output waveform by PWM Double Buffer function.

### UART Interface Controller (UART)

<b>UART_Autoflow_Master</b>	Transmit and receive data with auto flow control. This sample code needs to work with UART_Autoflow_Slave.
<b>UART_Autoflow_Slave</b>	Transmit and receive data with auto flow control. This sample code needs to work with UART_Autoflow_Master.
<b>UART_IrDA_Master</b>	Transmit and receive data in UART IrDA mode. This sample code needs to work with UART_IrDA_Slave.

<b>UART_IrDA_Slave</b>	Transmit and receive data in UART IrDA mode. This sample code needs to work with UART_IrDA_Master.
<b>UART_LIN</b>	Transmit LIN header and response.
<b>UART_RS485_Master</b>	Transmit and receive data in UART RS485 mode. This sample code needs to work with UART_RS485_Slave.
<b>UART_RS485_Slave</b>	Transmit and receive data in UART RS485 mode. This sample code needs to work with UART_RS485_Master.
<b>UART_TxRx_Function</b>	Transmit and receive data from PC terminal through RS232 interface.
<b>UART_Wakeup</b>	Show how to wake up system from Power-down mode by UART interrupt.

## Serial Peripheral Interface (SPI)

<b>SPI_Loopback</b>	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.
<b>SPI_MasterFifoMode</b>	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.
<b>SPI_SlaveFifoMode</b>	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.

## I<sup>2</sup>C Serial Interface Controller (I<sup>2</sup>C)

<b>I2C_EEPROM</b>	Show how to use I <sup>2</sup> C interface to access EEPROM.
<b>I2C_GCMode_Master</b>	Show how a Master uses I <sup>2</sup> C address 0x0 to write data to Slave. This sample code needs to work with I2C_GCMode_Slave.

<b>I2C_GCMode_Slave</b>	Show a Slave how to receive data from Master in GC (General Call) mode. This sample code needs to work with I2C_GCMode_Master.
<b>I2C_Master</b>	Show a Master how to access Slave. This sample code needs to work with I2C_Slave.
<b>I2C_Slave</b>	Show how to set I <sup>2</sup> C in Slave mode and receive the data from Master. This sample code needs to work with I2C_Master.
<b>I2C_Wakeup_Master</b>	Show how to wake up MCU from Power-down. This sample code needs to work with I2C_Wakeup_Slave.
<b>I2C_Wakeup_Slave</b>	Show how to wake up MCU from Power-down mode through I <sup>2</sup> C interface. This sample code needs to work with I2C_Wakeup_Master.

## Analog-to-Digital Converter (ADC)

<b>ADC_ContinuousScanMode</b>	Perform A/D Conversion with ADC continuous scan mode.
<b>ADC_MeasureAVDD</b>	Measure AVDD voltage by ADC.
<b>ADC_PwmTrigger</b>	Demonstrate how to trigger ADC by PWM.
<b>ADC_ResultMonitor</b>	Monitor the conversion result of channel 2 by the digital compare function.
<b>ADC_SingleCycleScanMode</b>	Perform A/D Conversion with ADC single cycle scan mode.
<b>ADC_SingleMode</b>	Perform A/D Conversion with ADC single mode.

### **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.

---

*Please note that all data and specifications are subject to change without notice.  
All the trademarks of products and companies mentioned in this datasheet belong to their respective owners.*