

# M030G/M031G Series CMSIS BSP Guide

Directory Introduction for 32-bit NuMicro® Family

## Directory Information

<b>Document</b>	Driver reference guide and revision history.
<b>Library</b>	Driver header and source files.
<b>SampleCode</b>	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\STDDRIVER.....</b>	<b>7</b>
	System Manager (SYS) .....	7
	Thermal Sensor (TS) .....	7
	Flash Memory Controller (FMC).....	7
	General Purpose I/O (GPIO).....	7
	PDMA Controller (PDMA) .....	8
	Timer Controller (TIMER).....	8
	Watchdog Timer (WDT) .....	9
	Window Watchdog Timer (WWDT) .....	9
	Basic PWM Generator and Capture Timer (BPWM).....	9
	UART Interface Controller (UART).....	9
	Serial Peripheral Interface (SPI) .....	10
	I <sup>2</sup> C Serial Interface Controller (I <sup>2</sup> C) .....	10
	CRC Controller (CRC) .....	11
	Analog-to-Digital Converter (ADC) .....	11
	Digital-to-Analog Converter (DAC) .....	12
	Manchester Codec Controller (MANCH).....	12
<b>6</b>	<b>SAMPLE CODE COMPATIBILITY LIST .....</b>	<b>14</b>

## 1 Document

CMSIS.html	Document of CMSIS version 5.1.1.
NuMicro M030G&M031G Driver Reference Guide.chm	This document describes the usage of drivers in M030G/M031G BSP.
NuMicro M030G&M031G Series CMSIS BSP Revision History.pdf	This document shows the revision history of M030G/M031G BSP.

## 2 Library

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

### 3 SampleCode

<b>Hard_Fault_Sample</b>	<p>Show hard fault information when hard fault happened.</p> <p>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.</p> <p>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.</p>
<b>ISP</b>	Sample codes for In-System-Programming.
<b>Semihost</b>	Show how to print and get character through IDE console window.
<b>StdDriver</b>	Sample code to demonstrate the usage of M030G/M031G series MCU peripheral driver APIs.
<b>Template</b>	A project template for M030G/M031G series MCU.

## 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\StdDriver

### System Manager (SYS)

<b>SYS_BODWakeup</b>	Demonstrate how to wake up system from Power-down mode by brown-out detector interrupt.
<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.

### Thermal Sensor (TS)

<b>TS_TemperatureMeasure</b>	Show how Thermal Sensor measure the current temperature by interrupt mechanism.
------------------------------	---

### Flash Memory Controller (FMC)

<b>FMC_CRC32</b>	Demonstrate how to use FMC CRC32 ISP command to calculate the CRC32 checksum of APROM and LDROM.
<b>FMC_IAP</b>	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.
<b>FMC_RW</b>	Show FMC read Flash IDs, erase, read, and write functions.

### 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.
-----------------------	--

## PDMA Controller (PDMA)

<b>PDMA_BasicMode</b>	Use PDMA channel 1 to transfer data from memory to memory.
<b>PDMA_ScatterGather</b>	Use PDMA channel 1 to transfer data from memory to memory by scatter-gather mode.
<b>PDMA_ScatterGather_PingPongBuffer</b>	Use PDMA to implement Ping-Pong buffer by scatter-gather mode (memory to memory).

## Timer Controller (TIMER)

<b>TIMER_CaptureCounter</b>	Show how to use the Timer capture function to capture Timer counter value.
<b>TIMER_Delay</b>	Demonstrate the usage of TIMER_Delay API to generate a 1 second delay.
<b>TIMER_EventCounter</b>	Use TM0 pin to demonstrate Timer event counter function.
<b>TIMER_FreeCountingMode</b>	Use the timer TM0_EXT pin to demonstrate timer free counting mode function. And displays the measured input frequency to UART console.
<b>TIMER_InterTimerTriggerMode</b>	Use the timer TM0 pin to demonstrate inter timer trigger mode function. Also display the measured input frequency to UART console.
<b>TIMER_Periodic</b>	Use the Timer periodic mode to generate Timer interrupt every 1 second.
<b>TIMER_PeriodicINT</b>	Implement Timer counting in periodic mode.
<b>TIMER_TimeoutWakeup</b>	Use timer to wake up system from Power-down mode periodically.
<b>TIMER_ToggleOut</b>	Demonstrate the Timer0 toggle out function on TM0 pin.



## Watchdog Timer (WDT)

<b>WDT_TimeoutWakeupAndReset</b>	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)

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

## Basic PWM Generator and Capture Timer (BPWM)

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

## UART Interface Controller (UART)

<b>UART_AutoBaudRate</b>	Show how to use auto baud rate detection function.
<b>UART_AutoFlow</b>	Transmit and receive data using auto flow control.
<b>UART_IrDA</b>	Transmit and receive UART data in UART IrDA mode.
<b>UART_PDMA</b>	Demonstrate UART transmit and receive function with PDMA.
<b>UART_RS485</b>	Transmit and receive data in UART RS485 mode.
<b>UART_SingleWire</b>	Transmit and receive data in UART single-wire mode.
<b>UART_TxRxFunction</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>	SPI read/write demo connecting SPI MISO and MOSI pins.
<b>SPI_MasterFIFOmode</b>	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.
<b>SPI_PDMA_LoopTest</b>	SPI read/write demo in PDMA mode. Connecting SPI MISO and MOSI pins. Both TX PDMA function and RX PDMA function will be enabled.
<b>SPI_SlaveFIFOmode</b>	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.

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

<b>I2C_Double_Buffer_Slave</b>	Demonstrate how to set I <sup>2</sup> C two-level buffer in Slave mode to receive 256 bytes data from a master. This sample code needs to work with I2C_MultiBytes_Master.
<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 a slave. This sample code needs to work with I2C_GCMode_Slave.
<b>I2C_GCMode_Slave</b>	Show how a slave receives data from a master in GC (General Call) mode. This sample code needs to work with I2C_GCMode_Master.
<b>I2C_Loopback</b>	Demonstrate how to set I <sup>2</sup> C Master mode and Slave Mode, and show how a master accesses a slave on a chip.

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

## CRC Controller (CRC)

<b>CRC_CCITT</b>	Implement CRC in CRC-CCITT mode and get the CRC checksum result.
<b>CRC_CRC32_PDMA</b>	Implement CRC in CRC-32 mode and get the CRC checksum result.
<b>CRC_CRC8</b>	Implement CRC in CRC-8 mode and get the CRC checksum result.

## Analog-to-Digital Converter (ADC)

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

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

## Digital-to-Analog Converter (DAC)

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

## Manchester Codec Controller (MANCH)

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

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

## 6 Sample Code Compatibility List

Category Type Sample Code	M030G	M031G
ADC_1411ksps_ContinuousScanMode	√	√
ADC_ADINT_Trigger	√	√
ADC_BandGap	√	√
ADC_BandGapCalculateAVDD	√	√
ADC_BurstMode	√	√
ADC_CountinuousScanMode	√	√
ADC_ResultMonitor	√	√
ADC_SingleCycleScanMode	√	√
ADC_SingleMode	√	√
ADC_STADC_Trigger	√	√
ADC_SwTrg_Trigger	√	√
ADC_Timer_Trigger	√	√
BPWM_Capture	√	√
BPWM_DoubleBuffer	√	√
BPWM_DutySwitch	√	√
BPWM_OutputWaveForm	√	√
CRC_CCIT	√	√
CRC_CRC8	√	√
CRC_CRC32_PDMA	√	√
CRC_POLYNOMIAL	-	√
DAC_PDMA_TimerTrigger	√	√
DAC_SoftwareTrigger	√	√
DAC_TimerTrigger	√	√
FMC_CRC32	√	√

FMC_IAP	√	√
FMC_RW	√	√
GPIO_EINTAndDebounce	√	√
GPIO_INT	√	√
GPIO_OutputInput	√	√
GPIO_PowerDown	√	√
I2C_Double_Buffer_Slave	√	√
I2C_EEPROM	√	√
I2C_GCMode_Master	√	√
I2C_GCMode_Slave	√	√
I2C_Loopback	√	√
I2C_Master	√	√
I2C_MultiBytes_Master	√	√
I2C_PDMA_TRX	√	√
I2C_SingleByte_Master	√	√
I2C_Slave	√	√
I2C_Wakeup_Slave	√	√
MANCH_RawSRAM2MTX	-	√
MANCH_RX	-	√
MANCH_TX	-	√
MANCH_TX2DACWithAmplitudeModulation	-	√
MANCH_TX2DACWithFrequencyModulation	-	√
MANCH_TXRXLoopback	-	√
PDMA_BasicMode	√	√
PDMA_ScatterGather	√	√
PDMA_ScatterGather_PingPongBuffer	√	√
SPI_Loopback	√	√
SPI_MasterFIFOmode	√	√

<b>SPI_PDMA_LoopTest</b>	√	√
<b>SPI_SlaveFIFOmode</b>	√	√
<b>SYS_BODWakeup</b>	√	√
<b>SYS_PLLClockOutput</b>	-	√
<b>SYS_PowerDown_MinCurrent</b>	√	√
<b>TIMER_CaptureCounter</b>	√	√
<b>TIMER_Delay</b>	√	√
<b>TIMER_EventCount</b>	√	√
<b>TIMER_FreeCountingMode</b>	√	√
<b>TIMER_InterTimerTriggerMode</b>	√	√
<b>TIMER_Periodic</b>	√	√
<b>TIMER_PeriodicINT</b>	√	√
<b>TIMER_TimeoutWakeup</b>	√	√
<b>TIMER_ToggleOut</b>	√	√
<b>TS_TemperatureMeasure</b>	√	√
<b>UART_AutoBaudRate</b>	√	√
<b>UART_AutoFlow</b>	√	√
<b>UART_IrDA</b>	√	√
<b>UART_PDMA</b>	√	√
<b>UART_RS485</b>	√	√
<b>UART_SingleWire</b>	√	√
<b>UART_TxRxFunction</b>	√	√
<b>UART_Wakeup</b>	√	√
<b>WDT_TimeoutWakeupAndReset</b>	√	√
<b>WWDT_ReloadCounter</b>	√	√



### **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.*