

M031 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.
ThirdParty	Library from FreeRTOS™ 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.....	4
2	LIBRARY	5
3	SAMPLECODE	6
4	SAMPLECODE\ISP	7
5	SAMPLECODE\LEVEL1_TRAINING	8
6	SAMPLECODE\REGBASED	9
	System Manager (SYS)	9
	Clock Controller (CLK).....	9
	Flash Memory Controller (FMC)	9
	General Purpose I/O (GPIO).....	9
	PDMA Controller (PDMA)	10
	Timer Controller (TIMER).....	10
	Watchdog Timer (WDT)	11
	Window Watchdog Timer (WWDT)	11
	Real Timer Clock (RTC)	11
	Basic PWM Generator and Capture Timer (BPWM).....	11
	PWM Generator and Capture Timer (PWM).....	12
	UART Interface Controller (UART).....	13
	Serial Peripheral Interface (SPI)	13
	Quad Serial Peripheral Interface (QSPI)	14
	I ² C Serial Interface Controller (I ² C)	14
	Universal Serial Control Interface Controller - UART Mode (USCI-UART)	15
	Universal Serial Control Interface Controller - SPI Mode (USCI-SPI).....	16
	Universal Serial Control Interface Controller - I ² C Mode (USCI-I2C)	16
	External Bus Interface (EBI)	17
	CRC Controller (CRC)	17
	Analog-to-Digital Converter (ADC)	18
	Analog Comparator Controller (ACMP)	19

	Hardware Divider (HDIV).....	19
7	SAMPLECODE\STDDRIVER.....	20
	System Manager (SYS)	20
	Clock Controller (CLK).....	20
	Flash Memory Controller (FMC)	20
	General Purpose I/O (GPIO).....	21
	PDMA Controller (PDMA)	21
	Timer Controller (TIMER).....	22
	Watchdog Timer (WDT)	22
	Window Watchdog Timer (WWDT)	22
	Real Timer Clock (RTC)	23
	Basic PWM Generator and Capture Timer (BPWM).....	23
	PWM Generator and Capture Timer (PWM).....	23
	UART Interface Controller (UART).....	24
	Serial Peripheral Interface (SPI).....	25
	Quad Serial Peripheral Interface (QSPI)	26
	I ² C Serial Interface Controller (I ² C)	26
	Universal Serial Control Interface Controller - UART Mode (USCI-UART)	27
	Universal Serial Control Interface Controller - SPI Mode (USCI-SPI)	27
	Universal Serial Control Interface Controller - I ² C Mode (USCI-I2C)	28
	External Bus Interface (EBI)	29
	USB 1.1 Device Controller (USBD)	29
	CRC Controller (CRC)	31
	Analog-to-Digital Converter (ADC)	31
	Analog Comparator Controller (ACMP)	32
	Hardware Divider (HDIV).....	33
8	THIRDPARTY.....	34
9	SAMPLE CODE COMPATIBILITY LIST	35

1 Document

CMSIS.html	Document of CMSIS version 4.5.0.
NuMicro M031 Driver Reference Guide.chm	This document describes the usage of drivers in M031 BSP.
NuMicro M031 Series CMSIS BSP Revision History.pdf	This document shows the revision history of M031 BSP.

2 Library

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

3 SampleCode

FreeRTOS	Simple FreeRTOS™ demo code.
Hard_Fault_Sample	<p>Show hard fault information when hard fault happened.</p> <p>The hard fault handler shows some information included program counter, which is the address where the processor is executing when the hard fault occurred. The listing file (or map file) can show what function and instruction that is.</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>
ISP	Sample codes for In-System-Programming.
Level1_Training	Sample codes for Level1 training course.
NuMaker-M03xBT	Sample codes, tools and documents for M031BT/M032BT.
RegBased	The sample codes that access control registers directly.
Semihost	Show how to print and get character through IDE console window.
StdDriver	Sample code to demonstrate the usage of M031 series MCU peripheral driver APIs.
Template	A project template for M031 series MCU.

4 SampleCode\ISP

ISP_DFU	Demonstrate how to use specific requests in DFU class to read data from Flash or write a firmware image file to Flash.
ISP_HID	Sample ISP firmware communicated with the ISP tool through a USBD HID interface.
ISP_I2C	Sample ISP firmware communicated with the ISP tool through an I ² C interface.
ISP_MSC	The Mass Storage (MSC) ISP supports to download firmware to APROM through mass storage interface. There is no tool or driver installation required before downloading firmware.
ISP_RS485	Sample ISP firmware communicated with the ISP tool through a RS485 interface.
ISP_SPI	Sample ISP firmware communicated with the ISP tool through a SPI interface.
ISP_UART	Sample ISP firmware communicated with the ISP tool through a UART interface.

5 SampleCode\Level1_Training

Level1_ADC_BandGapCalculateAVDD	Demonstrate how to calculate battery voltage (AV_{DD}) by using band-gap.
Level1_GPIO_Ctrl	Demonstrate how to use GPIO to control LEDs.
Level1_I2C_EEPROM	Demonstrate how to use I ² C to access EEPROM.
Level1_PWM_LED_Breathing	Demonstrate how to use PWM to control LED.
Level1_PWM_LED_Duty	Demonstrate how to use PWM to control LED.
Level1_SPI_Flash	Demonstrate how to access SPI Flash.
Level1_Timer_Ctrl	Demonstrate how to use the Timer0 periodic mode to generate Timer interrupt every 1 second and Timer1 toggle out function on pin PB.14.
Level1_UART_ReceiveUnknownDataLengthPkg	Demonstrate how to use UART timeout function to realize getting data package within unknown data length.
Level1_WDT	Demonstrate how to wake up MCU by using WDT.

6 SampleCode\RegBased

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 CLK0 pin.
SYS_TrimHIRC	Demonstrate how to use LXT to trim HIRC.

Clock Controller (CLK)

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

Flash Memory Controller (FMC)

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_ADC_1882ksps_ContinuousScanMode	Demonstrate how to transfer 1.822 Msps conversion data from ADC to SRAM by PDMA when ADC clock source is selected to HXT that is connected to 32 MHz crystal.
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).
PDMA_TimerCapture_CalculatePlusePeriod	Demonstrate timer capture function to measure the frequency of external signal through PDMA transfer.
PDMA_TimerTrigger_SRAM2GPIO	Demonstrate how to use PDMA channel 1 to transfer data from SRAM to GPIO based on 1 kHz timer trigger.

Timer Controller (TIMER)

TIMER_ACMPTrigger	Use ACMP to trigger Timer reset mode.
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 display the measured input frequency to UART console.
TIMER_InterTimerTriggerMode	Use the timer TM0 pin to demonstrate inter timer trigger mode function, and 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_SW_RTC	Implement software RTC function by Timer0.
TIMER_TimeOutWakeup	Use timer to wake up system from Power-down mode periodically.
TIMER_ToggleOut	Demonstrate the Timer0 toggle out function on pin PB.5.

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.
---------------------------	--

Real Timer Clock (RTC)

RTC_Alarm_Test	Demonstrate the RTC alarm function that sets an alarm 10 seconds after execution.
RTC_Alarm_Wakeup	Demonstrate how to wake up system periodically with RTC interrupt.
RTC_Time_Display	Demonstrate the RTC function and display current time to the UART console.

Basic PWM Generator and Capture Timer (BPWM)

BPWM_Capture	Capture the BPWM1 Channel 0 waveform by BPWM0 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 a configured period.
BPWM_OutputWaveform	Demonstrate how to use BPWM counter output waveform.
BPWM_SyncStart	Demonstrate how to use BPWM counter synchronous start function.

PWM Generator and Capture Timer (PWM)

PWM_240KHz_DutySwitch	Demonstrate how to set PWM0 channel 0 output 240 kHz waveform and switch duty in each 0.5%.
PWM_Brake	Demonstrate how to use PWM brake function.
PWM_Capture	Capture the PWM0 Channel 0 waveform by PWM0 Channel 2.
PWM_DeadZone	Demonstrate how to use PWM Dead Time function.
PWM_DoubleBuffer_PeriodLoadingMode	Change duty cycle and period of output waveform by PWM Double Buffer function.
PWM_DutySwitch	Change duty cycle of output waveform by a configured period.
PWM_OutputWaveform	Demonstrate how to use PWM counter output waveform.
PWM_PDMA_Capture	Capture the PWM0 Channel 0 waveform by PWM0 Channel 2, and use PDMA to transfer captured data.
PWM_PDMA_Capture_1MHzSignal	Capture the PWM0 Channel 0 waveform by PWM0 Channel 2, and use PDMA to transfer captured data. The frequency of PWM Channel 0 is 1 MHz, which is used to test the maximum input frequency for PWM Capture function.
PWM_SyncStart	Demonstrate how to use PWM counter synchronous start function.

UART Interface Controller (UART)

UART_6Mbps_SingleWire	Demonstrate how to transfer 6 Mbps UART data through Single Wire.
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	A SPI read/write demo connecting SPI0 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.

I2S_Master	Configure SPI in I ² S Master mode and demonstrate how I ² S works in Master mode.
I2S_PDMA_NAU8822	An I ² S demo with PDMA function connected to audio codec NAU8822.
I2S_PDMA_Play	An I ² S demo for playing data and demonstrating how I ² S works with PDMA.
I2S_PDMA_PlayRecord	An I ² S demo for playing and recording data with PDMA function.
I2S_PDMA_Record	An I ² S demo for recording data and demonstrating how I ² S works with PDMA.
I2S_Slave	Configure SPI as I ² S Slave mode and demonstrate how I ² S works in Slave mode. This sample code needs to work with I2S_Master.

Quad Serial Peripheral Interface (QSPI)

QSPI_DualMode_Flash	Access SPI Flash using QSPI dual mode.
QSPI_QuadMode_Flash	Access SPI Flash using QSPI quad mode.

I²C Serial Interface Controller (I²C)

I2C_EEPROM	Show how to use I ² C interface to access EEPROM.
I2C_GCMode_Master	Show how a master uses I ² C 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 I ² C 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. Need to connect I2C0 (master) and I2C1 (slave).
I2C_SMBus_Master	Show how to control SMBus interface and use SMBus protocol between Host and Slave. This sample code needs to work with I2C_SMBus_Slave.
I2C_SMBus_Slave	Show how to control SMBus interface and use SMBus protocol between Host and Slave. This sample code needs to work with I2C_SMBus_Master.
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.

Universal Serial Control Interface Controller - UART Mode (USCI-UART)

USCI_UART_AutoBaudRate	Show how to use auto baud rate detection function.
USCI_UART_Autoflow_Master	Transmit and receive data with auto flow control. This sample code needs to work with USCI_UART_Autoflow_Slave.
USCI_UART_Autoflow_Slave	Transmit and receive data with auto flow control. This sample code needs to work with USCI_UART_Autoflow_Master.
USCI_UART_PDMA	This is a USCI_UART PDMA demo and need to connect USCI_UART Tx and Rx.

USCI_UART_RS485_Master	Transmit and receive data in RS485 mode. This sample code needs to work with USCI_UART_RS485_Slave.
USCI_UART_RS485_Slave	Transmit and receive data in RS485 mode. This sample code needs to work with USCI_UART_RS485_Master.
USCI_UART_TxRxFunction	Transmit and receive data from PC terminal through RS232 interface.
USCI_UART_Wakeup	Show how to wake up system from Power-down mode by USCI interrupt in UART mode.

Universal Serial Control Interface Controller - SPI Mode (USCI-SPI)

USCI_SPI_Loopback	Implement USCI_SPI0 Master loop back transfer. This sample code needs to connect USCI_SPI0_MISO pin and USCI_SPI0_MOSI pin together. It will compare the received data with transmitted data.
USCI_SPI_MasterMode	Configure USCI_SPI0 as Master mode and demonstrate how to communicate with an off-chip SPI slave device. This sample code needs to work with USCI_SPI_SlaveMode.
USCI_SPI_PDMA_LoopTest	Demonstrate USCI_SPI data transfer with PDMA. USCI_SPI0 will be configured as Master mode and USCI_SPI1 will be configured as Slave mode. Both TX PDMA function and RX PDMA function will be enabled.
USCI_SPI_SlaveMode	Configure USCI_SPI0 as Slave mode and demonstrate how to communicate with an off-chip SPI master device. This sample code needs to work with USCI_SPI_MasterMode.

Universal Serial Control Interface Controller - I²C Mode (USCI-I2C)

USCI_I2C_EEPROM	Show how to use USCI_I2C interface to access EEPROM.
USCI_I2C_Lookback	Show how a master accesses 7-bit address Slave (loopback).

USCI_I2C_Loopback_10bit	Show how a master accesses 10-bit address Slave (loopback).
USCI_I2C_Master	Show how a master accesses a slave. This sample code needs to work with USCI_I2C_Slave.
USCI_I2C_Master_10bit	Show how a master accesses a 10-bit address slave. This sample code need works with USCI_I2C_Slave_10bit.
USCI_I2C_Monitor	Demonstrate USCI_I2C Monitor mode.
USCI_I2C_MultiBytes_Master	Show how to set USCI_I2C Multi bytes API Read and Write data to Slave. This sample code needs to work with USCI_I2C_Slave.
USCI_I2C_SingleByte_Master	Show how to use USCI_I2C Single byte API Read and Write data to Slave. This sample code needs to work with USCI_I2C_Slave.
USCI_I2C_Slave	Show how a slave receives data from a master. This sample code needs to work with USCI_I2C_Master.
USCI_I2C_Slave_10bit	Show how a 10-bit address slave receives data from a master. This sample code need works with USCI_I2C_Master_10bit.
USCI_I2C_Wakeup_Slave	Show how to wake up USCI_I2C from Deep Sleep mode. This sample code needs to work with USCI_I2C_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.
CRC_CRC32_PDMA	Implement CRC in CRC-32 mode and get the CRC checksum result.

Analog-to-Digital Converter (ADC)

ADC_2MSPs_ContinuousScanMode	Demonstrate how to use PLL as ADC clock source to achieve 2 MSPs ADC conversion rate.
ADC_1411KSPs_ContinuousScanMode	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 a 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_PDMA_PWM_Trigger	Demonstrate how to trigger EADC by PWM and transfer conversion data by PDMA.
ADC_PWM_Trigger	Demonstrate how to trigger EADC 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_SingleCycleScanMode2	Perform A/D Conversion with ADC single mode to simulate 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 EADC by Timer.

Analog Comparator Controller (ACMP)

ACMP_ComapreVBG	Demonstrate analog comparator (ACMP) comparison by comparing ACMP1_P1 input and VBG voltage and shows the result on UART console.
ACMP_Wakeup	Use ACMP to wake up system from Power-down mode while comparator output changes.
ACMP_WindowCompare	Show how to monitor ACMP input with window compare function.
ACMP_WindowLatch	Demonstrate how to use ACMP window latch mode.

Hardware Divider (HDIV)

HDIV	Demonstrate how to divide two signed integers by HDIV engine.
-------------	---

7 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 CLK0 pin.
SYS_TrimHIRC	Demonstrate how to use LXT to trim HIRC.

Clock Controller (CLK)

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

Flash Memory Controller (FMC)

FMC_CRC32	Demonstrate how to use FMC CRC32 ISP command to calculate the CRC32 checksum of APROM, LDROM, and SPROM.
FMC_ExecInSRAM	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. A LDROM code and 4 APROM code are implemented in this sample code.
FMC_MultiWordProgram	Show FMC multi-word program ISP command to program APROM 0x10000~0x20000 area.
FMC_ReadAllOne	Demonstrate how to use FMC Read-All-One ISP command to verify if APROM/LDROM pages are all 0xFFFFFFFF.

FMC_RemapBank	Demonstrate how to use FMC Bank Remap ISP command to remap different bank to address 0.
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_ADC_1882ksps_ContinuousScanMode	Demonstrate how to transfer 1.822 Msps conversion data from ADC to SRAM by PDMA when ADC clock source is selected to HXT that is connected to 32 MHz crystal.
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).
PDMA_TimerCapture_CalculatePlusePeriod	Demonstrate timer capture function to measure the frequency of external signal through PDMA transfer.
PDMA_TimerTrigger_SRAM2GPIO	Demonstrate how to use PDMA channel 1 to transfer data from SRAM to GPIO based on 1 kHz timer trigger.

Timer Controller (TIMER)

TIMER_ACMPTrigger	Use ACMP to trigger Timer reset mode.
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 display the measured input frequency to UART console.
TIMER_InterTimerTriggerMode	Use the timer TM0 pin to demonstrate inter timer trigger mode function, and 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_SW_RTC	Implement software RTC function by Timer0.
TIMER_TimeOutWakeup	Use timer to wake up system from Power-down mode periodically.
TIMER_ToggleOut	Demonstrate the Timer0 toggle out function on pin PB.5.

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.
---------------------------	--

Real Timer Clock (RTC)

RTC_Alarm_Test	Demonstrate the RTC alarm function that sets an alarm 10 seconds after execution.
RTC_Alarm_Wakeup	Demonstrate how to wake up system periodically with RTC interrupt.
RTC_Time_Display	Demonstrate the RTC function and display current time to the UART console.

Basic PWM Generator and Capture Timer (BPWM)

BPWM_Capture	Capture the BPWM1 Channel 0 waveform by BPWM0 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 a configured period.
BPWM_OutputWaveform	Demonstrate how to use BPWM counter output waveform.
BPWM_SyncStart	Demonstrate how to use BPWM counter synchronous start function.

PWM Generator and Capture Timer (PWM)

PWM_240KHz_DutySwitch	Demonstrate how to set PWM0 channel 0 output 240 kHz waveform and switch duty in each 0.5%.
PWM_Brake	Demonstrate how to use PWM brake function.
PWM_Capture	Capture the PWM0 Channel 0 waveform by PWM0 Channel 2.
PWM_DeadZone	Demonstrate how to use PWM Dead Time function.
PWM_DoubleBuffer_PeriodLoadingMode	Change duty cycle and period of output waveform by PWM Double Buffer function.

PWM_DutySwitch	Change duty cycle of output waveform by a configured period.
PWM_OutputWaveform	Demonstrate how to use PWM counter output waveform.
PWM_PDMA_Capture	Capture the PWM0 Channel 0 waveform by PWM0 Channel 2, and use PDMA to transfer captured data.
PWM_PDMA_Capture_1MHzSignal	Capture the PWM0 Channel 0 waveform by PWM0 Channel 2, and use PDMA to transfer captured data. The frequency of PWM Channel 0 is 1 MHz, which is used to test the maximum input frequency for PWM Capture function.
PWM_SyncStart	Demonstrate how to use PWM counter synchronous start function.

UART Interface Controller (UART)

UART_6Mbps_SingleWire	Demonstrate how to transfer 6 Mbps UART data through Single Wire.
UART_115200bps_SingleWire_ISP	The UART ISP LDROM firmware used to update APROM through Single Wire (PB.12). The UART must be set to "baud-rate 115200 bps" and "8-N-1". Boot option must be set to "boot in LDROM".
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	A SPI read/write demo connecting SPI0 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.
I2S_Master	Configure SPI in I ² S Master mode and demonstrate how I ² S works in Master mode.
I2S_PDMA_NAU8822	An I ² S demo with PDMA function connected to audio codec NAU8822.
I2S_PDMA_Play	An I ² S demo for playing data and demonstrating how I ² S works with PDMA.
I2S_PDMA_PlayRecord	An I ² S demo for playing and recording data with PDMA function.
I2S_PDMA_Record	An I ² S demo for recording data and demonstrating how I ² S works with PDMA.
I2S_Slave	Configure SPI as I ² S Slave mode and demonstrate how I ² S works in Slave mode. This sample code needs to work with I2S_Master.

Quad Serial Peripheral Interface (QSPI)

QSPI_DualMode_Flash	Access SPI Flash using QSPI dual mode.
QSPI_QuadMode_Flash	Access SPI Flash using QSPI quad mode.

I²C Serial Interface Controller (I²C)

I2C_EEPROM	Show how to use I ² C interface to access EEPROM.
I2C_GCMode_Master	Show how a master uses I ² C 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 I ² C 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. Need to connect I2C0 (master) and I2C1 (slave).
I2C_SMBus_Master	Show how to control SMBus interface and use SMBus protocol between Host and Slave. This sample code needs to work with I2C_SMBus_Slave.
I2C_SMBus_Slave	Show how to control SMBus interface and use SMBus protocol between Host and Slave. This sample code needs to work with I2C_SMBus_Master.
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.

Universal Serial Control Interface Controller - UART Mode (USCI-UART)

USCI_UART_AutoBaudRate	Show how to use auto baud rate detection function.
USCI_UART_Autoflow_Master	Transmit and receive data with auto flow control. This sample code needs to work with USCI_UART_Autoflow_Slave.
USCI_UART_Autoflow_Slave	Transmit and receive data with auto flow control. This sample code needs to work with USCI_UART_Autoflow_Master.
USCI_UART_PDMA	This is a USCI_UART PDMA demo and need to connect USCI_UART Tx and Rx.
USCI_UART_RS485_Master	Transmit and receive data in RS485 mode. This sample code needs to work with USCI_UART_RS485_Slave.
USCI_UART_RS485_Slave	Transmit and receive data in RS485 mode. This sample code needs to work with USCI_UART_RS485_Master.
USCI_UART_TxRxFunction	Transmit and receive data from PC terminal through RS232 interface.
USCI_UART_Wakeup	Show how to wake up system from Power-down mode by USCI interrupt in UART mode.

Universal Serial Control Interface Controller - SPI Mode (USCI-SPI)

USCI_SPI_Loopback	Implement USCI_SPI0 Master loop back transfer. This sample code needs to connect USCI_SPI0_MISO pin and USCI_SPI0_MOSI pin together. It will compare the received data with transmitted data.
--------------------------	---

USCI_SPI_MasterMode	Configure USCI_SPI0 as Master mode and demonstrate how to communicate with an off-chip SPI slave device. This sample code needs to work with USCI_SPI_SlaveMode.
USCI_SPI_PDMA_LoopTest	Demonstrate USCI_SPI data transfer with PDMA. USCI_SPI0 will be configured as Master mode and USCI_SPI1 will be configured as Slave mode. Both TX PDMA function and RX PDMA function will be enabled.
USCI_SPI_SlaveMode	Configure USCI_SPI0 as Slave mode and demonstrate how to communicate with an off-chip SPI master device. This sample code needs to work with USCI_SPI_MasterMode.

Universal Serial Control Interface Controller - I²C Mode (USCI-I2C)

USCI_I2C_EEPROM	Show how to use USCI_I2C interface to access EEPROM.
USCI_I2C_Lookback	Show how a master accesses 7-bit address Slave (loopback).
USCI_I2C_Loopback_10bit	Show how a master accesses 10-bit address Slave (loopback).
USCI_I2C_Master	Show how a master accesses a slave. This sample code needs to work with USCI_I2C_Slave.
USCI_I2C_Master_10bit	Show how a master accesses a 10-bit address slave. This sample code need works with USCI_I2C_Slave_10bit.
USCI_I2C_Monitor	Demonstrate USCI_I2C Monitor mode.
USCI_I2C_MultiBytes_Master	Show how to set USCI_I2C Multi bytes API Read and Write data to Slave. This sample code needs to work with USCI_I2C_Slave.
USCI_I2C_SingleByte_Master	Show how to use USCI_I2C Single byte API Read and Write data to Slave. This sample code needs to work with USCI_I2C_Slave.

USCI_I2C_Slave	Show how a slave receives data from a master. This sample code needs to work with USCI_I2C_Master.
USCI_I2C_Slave_10bit	Show how a 10-bit address slave receives data from a master. This sample code need works with USCI_I2C_Master_10bit.
USCI_I2C_Wakeup_Slave	Show how to wake up USCI_I2C from Deep Sleep mode. This sample code needs to work with USCI_I2C_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.

USB 1.1 Device Controller (USBD)

USBD_Audio_NAU8822_Microphone	Demonstrate how to implement a USB audio class device (Microphone) with HID key (MediaKey/JoyStick). NAU8822 is used in this sample code to record data from NAU8822 to Host.
USBD_Audio_NAU8822_Speaker	Demonstrate how to implement a USB audio class device (Speaker) with HID key (MediaKey/JoyStick). NAU8822 is used in this sample code to play the audio data from Host.
USBD_HID_Keyboard	Demonstrate how to implement a USB keyboard device. This sample code supports to use GPIO to simulate key input.
USBD_HID_Mouse	Simulate a USB mouse and draws circle on the screen.
USBD_HID_MouseKeyboard	Simulate an USB HID mouse and HID keyboard. Mouse draws circle on the screen and Keyboard uses GPIO to simulate key input.
USBD_HID_RemoteWakeup	Simulate a HID mouse supports USB suspend and

	remote wakeup.
USBD_HID_Touch	Demonstrate how to implement a USB touch digitizer device. Two lines demo in Paint.
USBD_HID_Transfer	Demonstrate how to transfer data between a USB device and PC through a USB HID interface. A windows tool is also included in this sample code to connect with a USB device.
USBD_HID_Transfer_And_Keyboard	Demonstrate how to implement a composite device of HID transfer and keyboard. Transfer data between USB device and PC through USB HID interface. A windows tool is also included in this sample code to connect with a USB device.
USBD_HID_Transfer_And_MSC	Demonstrate how to implement a composite device of HID transfer and mass storage. Transfer data between USB device and PC through USB HID interface. A windows tool is also included in this sample code to connect with a USB device.
USBD_HID_Transfer_CTRL	Use USB host core driver and HID driver. It shows how to submit HID class request and how to read data from control pipe. A windows tool is also included in this sample code to connect with a USB device.
USBD_Mass_Storage_CDROM	Demonstrate the emulation of USB Mass Storage Device CD-ROM.
USBD_Mass_Storage_Flash	Use internal flash as backend storage media to simulate a USB pen drive.
USBD_Micro_Printer	Demonstrate how to implement a USB micro printer device.
USBD_Printer_And_HID_Transfer	Demonstrate how to implement a composite device of USB micro printer and HID transfer. Transfer data between USB device and PC through USB HID interface. A windows tool is also included in this sample code to connect with a USB device.
USBD_VCOM_And_HID_Keyboard	Demonstrate how to implement a composite device of VCOM and HID keyboard.

USBD_VCOM_And_HID_Transfer	Demonstrate how to implement a composite device. (VCOM and HID Transfer) Transfer data between USB device and PC through USB HID interface. A windows tool is also included in this sample code to connect with a USB device.
USBD_VCOM_And_Mass_Storage	Demonstrate how to implement a composite device.(Virtual COM port and Mass storage device)
USBD_VCOM_DualPort	Demonstrate how to implement a USB dual virtual COM port device.
USBD_VCOM_SerialEmulator	Demonstrate how to implement a USB virtual COM port device.

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.
CRC_CRC32_PDMA	Implement CRC in CRC-32 mode and get the CRC checksum result.

Analog-to-Digital Converter (ADC)

ADC_2Msps_ContinuousScanMode	Demonstrate how to use PLL as ADC clock source to achieve 2 Msps ADC conversion rate.
ADC_1411ksps_ContinuousScanMode	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 a conversion result.
ADC_BandGapCalculateAVDD	Demonstrate how to calculate battery voltage (AV _{DD}) 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_PDMA_PWM_Trigger	Demonstrate how to trigger EADC by PWM and transfer conversion data by PDMA.
ADC_PWM_Trigger	Demonstrate how to trigger EADC 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_SingleCycleScanMode2	Perform A/D Conversion with ADC single mode to simulate 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 EADC by Timer.

Analog Comparator Controller (ACMP)

ACMP_ComapreVBG	Demonstrate analog comparator (ACMP) comparison by comparing ACMP1_P1 input and VBG voltage and shows the result on UART console.
ACMP_Wakeup	Use ACMP to wake up system from Power-down mode while comparator output changes.
ACMP_WindowCompare	Show how to monitor ACMP input with window compare function.
ACMP_WindowLatch	Demonstrate how to use ACMP window latch mode.

Hardware Divider (HDIV)

HDIV	Demonstrate how to divide two signed integers by HDIV engine.
------	---

8 ThirdParty

FreeRTOS

A real time operating system available for free download. Its official website is: <http://www.freertos.org/>.

9 Sample Code Compatibility List

Category Type	Part Number	Note
Type-B	M031EB0AE, M031FB0AE, M031TB0AE	
Type-C	M031EC1AE, M031FC1AE, M031TC1AE	
Type-D	M031LD2AE, M031SD2AE, M031TD2AE, M031LC2AE, M031SC2AE	
Type-E	M031LE3AE, M031SE3AE, M032LE3AE, M032SE3AE	
Type-I	M031SIAAE, M031KIAAE, M032SIAAE, M032KIAAE	
Type-G	M031LG6AE, M031SG6AE, M031KG6AE, M031LG8AE, M031SG8AE, M031KG8AE, M032LG6AE, M032SG6AE, M032KG6AE, M032LG8AE, M032SG8AE, M032KG8AE	
Type-J	M032FC1AE, M032EC1AE, M032TC1AE	
Type-L	M032LC2AE, M032TD2AE, M032LD2AE	

Category Type Sample Code	Type-B	Type-C	Type-D	Type-E	Type-I	Type-G	Type-J	Type-L
ACMP_ComapreVBG	-	-	√	√	√	√	-	-
ACMP_Wakeup	-	-	√	√	√	√	-	-
ACMP_WindowCompare	-	-	√	√	√	√	-	-
ACMP_WindowLatch	-	-	√	√	√	√	-	-
ADC_2Msps_ ContinuousScanMode	-	-	√	√	√	√	-	-
ADC_1411ksps_ ContinuousScanMode	√	√	√	√	√	√	√	√
ADC_ADINT_Trigger	√	√	√	√	√	√	√	√
ADC_BandGap	√	√	√	√	√	√	√	√
ADC_BandGapCalculateA VDD	√	√	√	√	√	√	√	√
ADC_BurstMode	√	√	√	√	√	√	√	√

ADC_ContinuousScanMode	√	√	√	√	√	√	√	√
ADC_PDMA_PWM_Trigger	-	√	√	√	√	√	-	-
ADC_PWM_Trigger	√	√	√	√	√	√	-	-
ADC_ResultMonitor	√	√	√	√	√	√	√	√
ADC_SingleCycleScanMode	√	√	√	√	√	√	√	√
ADC_SingleCycleScanMode2	√	√	√	√	√	√	√	√
ADC_SingleMode	√	√	√	√	√	√	√	√
ADC_STADC_Trigger	√	√	√	√	√	√	√	√
ADC_SwTrg_Trigger	√	√	√	√	√	√	√	√
ADC_Timer_Trigger	√	√	√	√	√	√	√	√
BPWM_Capture	-	-	-	-	√	√	-	-
BPWM_DoubleBuffer	-	-	-	-	√	√	√	√
BPWM_DutySwitch	-	-	-	-	√	√	√	√
BPWM_OutputWaveform	-	-	-	-	√	√	√	√
BPWM_SyncStart	-	-	-	-	√	√	-	√
CLK_ClockDetector	√	√	√	√	√	√	-	-
CRC_CCITT	√	√	√	√	√	√	-	-
CRC_CRC8	√	√	√	√	√	√	-	-
CRC_CRC32_PDMA	-	√	√	√	√	√	-	-
EBI_NOR	-	-	-	√	√	√		
EBI_SRAM	-	-	-	√	√	√		
FMC_CRC32	√	√	√	√	√	√	√	√
FMC_ExecInSRAM	√	√	√	√	√	√	√	√
FMC_IAP	√	√	√	√	√	√	√	√
FMC_MultiBoot	√	√	√	√	√	√	-	-
FMC_MultiWordProgram	-	-	-	-	√	√	-	-
FMC_ReadAllOne	-	-	-	-	√	√	-	-

FMC_RemapBank	-	-	-	-	√	-	-	-
FMC_RW	√	√	√	√	√	√	√	√
GPIO_EINTAndDebounce	-	-	√	√	√	√	-	√
GPIO_INT	√	√	√	√	√	√	√	√
GPIO_OutputInput	√	√	√	√	√	√	√	√
GPIO_PowerDown	√	√	√	√	√	√	√	√
HDIV	√	√	√	√	√	√	-	√
I2C_EEPROM	√	√	√	√	√	√	-	-
I2C_GCMode_Master	√	√	√	√	√	√	-	-
I2C_GCMode_Slave	√	√	√	√	√	√	-	-
I2C_Loopback	√	√	√	√	√	√	-	-
I2C_Master	√	√	√	√	√	√	-	-
I2C_MultiBytes_Master	√	√	√	√	√	√	-	-
I2C_PDMA_TRX	-	-	√	√	√	√	-	-
I2C_SMBus_Master	-	-	-	-	√	√	-	-
I2C_SMBus_Slave	-	-	-	-	√	√	-	-
I2C_SingleByte_Master	√	√	√	√	√	√	-	-
I2C_Slave	√	√	√	√	√	√	-	-
I2C_Wakeup_Slave	√	√	√	√	√	√	-	-
I2S_Master	√	√	√	√	√	√	√	√
I2S_PDMA_NAU8822	-	√	√	√	√	√	-	-
I2S_PDMA_Play	-	√	√	√	√	√	-	√
I2S_PDMA_PlayRecord	-	√	√	√	√	√	-	√
I2S_PDMA_Record	-	√	√	√	√	√	-	√
I2S_Slave	√	√	√	√	√	√	√	√
PDMA_ADC_1882ksps_ContinousScanMode	-	√	√	√	√	√	-	-
PDMA_BasicMode	-	√	√	√	√	√	-	√
PDMA_ScatterGather	-	√	√	√	√	√	-	√
PDMA_ScatterGather_	-	√	√	√	√	√	-	√

PingPongBuffer								
PDMA_TimerCapture_ CalculatePlusePeriod	-	√	√	√	√	√	-	√
PDMA_TimerTrigger_SRA M2GPIO	-	√	√	√	√	√	-	√
PWM_240KHz_ DutySwitch	√	√	√	√	√	√	-	-
PWM_Brake	√	√	√	√	√	√	-	-
PWM_Capture	√	√	√	√	√	√	-	-
PWM_DeadZone	√	√	√	√	√	√	-	-
PWM_DoubleBuffer_ PeriodLoadingMode	√	√	√	√	√	√	-	-
PWM_DutySwitch	√	√	√	√	√	√	-	-
PWM_OutputWaveForm	√	√	√	√	√	√	-	-
PWM_PDMA_Capture	-	√	√	√	√	√	-	-
PWM_PDMA_Capture_ 1MHzSignal	-	√	√	√	√	√	-	-
PWM_SyncStart	√	√	√	√	√	√	-	-
QSPI_DualMode_Flash	-	-	-	-	√	√	-	√
QSPI_QuadMode_Flash	-	-	-	-	√	√	-	√
RTC_Alarm_Test	-	-	-	-	√	√	-	-
RTC_Alarm_Wakeup	-	-	-	-	√	√	-	-
RTC_Time_Display	-	-	-	-	√	√	-	-
SPI_LoopBack	√	√	√	√	√	√	√	√
SPI_MasterFIFOmode	√	√	√	√	√	√	√	√
SPI_PDMA_LoopTest	-	√	√	√	√	√	-	√
SPI_SlaveFIFOmode	√	√	√	√	√	√	√	√
SYS_BODWakeup	√	√	√	√	√	√	√	√
SYS_PLLClockOutput	-	-	√	√	√	√	-	-
SYS_PowerDon_ MinCurrent	√	√	√	√	√	√	√	√

SYS_TrimHIRC	-	√	√	√	√	√	√	√
TIMER_ACMPTTrigger	-	-	√	√	√	√	-	-
TIMER_CaptureCounter	-	√	√	√	√	√	-	√
TIMER_Delay	√	√	√	√	√	√	√	√
TIMER_EventCounter	√	√	√	√	√	√	√	√
TIMER_FreeCounting Mode	√	√	√	√	√	√	√	√
TIMER_InterTimerTrigger Mode	√	√	√	√	√	√	√	√
TIMER_Periodic	√	√	√	√	√	√	√	√
TIMER_PeriodicINT	-	√	√	√	√	√	-	√
TIMER_SW_RTC	-	√	√	√	√	√	-	-
TIMER_TimeOutWakeup	√	√	√	√	√	√	√	√
TIMER_ToggleOut	√	√	√	√	√	√	√	√
UART_6Mbps_SingleWire	√	√	√	√	√	√	-	-
UART_115200bps_ SingleWire_ISP[*1]	√	√	√	√	√	√	√	√
UART_AutoBaudRate	√	√	√	√	√	√	-	-
UART_AutoFlow	√	√	√	√	√	√	-	-
UART_IrDA	√	√	√	√	√	√	-	-
UART_PDMA	-	√	√	√	√	√	-	-
UART_RS485	√	√	√	√	√	√	-	-
UART_SingleWire	√	√	√	√	√	√	-	-
UART_TxRxFunction	√	√	√	√	√	√	√	√
UART_Wakeup	√	√	√	√	√	√	-	-
USBDAudio_NAU8822_ Headset	-	-	-	√	√	√	-	-
USBDAudio_NAU8822_ Microphone	-	-	-	√	√	√	-	-
USBDAudio_NAU8822_ Speaker	-	-	-	√	√	√	-	-

USBD_HID_Keyboard	-	-	-	√	√	√	√	√
USBD_HID_Mouse	-	-	-	√	√	√	√	√
USBD_HID_Mouse Keyboard	-	-	-	√	√	√	√	√
USBD_HID_RemoteWake up	-	-	-	√	√	√	√	√
USBD_HID_Touch	-	-	-	√	√	√	√	√
USBD_HID_Transfer	-	-	-	√	√	√	-	-
USBD_HID_Transfer_And _Keyboard	-	-	-	√	√	√	-	-
USBD_HID_Transfer_And _MSC	-	-	-	√	√	√	-	-
USBD_HID_Transfer_ CTRL	-	-	-	√	√	√	√	√
USBD_Mass_Storage_ CDROM	-	-	-	√	√	√	√	√
USBD_Mass_Storage_ Flash	-	-	-	√	√	√	-	-
USBD_Micro_Printer	-	-	-	√	√	√	√	√
USBD_Printer_And_HID_ Transfer	-	-	-	√	√	√	-	-
USBD_VCOM_And_HID_ Keyboard	-	-	-	√	√	√	√	√
USBD_VCOM_And_HID_ Transfer	-	-	-	√	√	√	-	-
USBD_VCOM_And_Mass_ Storage	-	-	-	√	√	√	-	-
USBD_VCOM_DualPort	-	-	-	√	√	√	√	√
USBD_VCOM_Serial Emulator	-	-	-	√	√	√	√	√
USCI_I2C_EEPROM	-	-	√	√	√	√	-	√
USCI_I2C_Loopback	-	-	-	-	√	√	-	√

USCI_I2C_Loopback_10bit	-	-	-	-	√	√	-	√
USCI_I2C_Master	-	-	√	√	√	√	-	√
USCI_I2C_Master_10bit	-	-	√	√	√	√	-	√
USCI_I2C_Monitor	-	-	√	√	√	√	-	-
USCI_I2C_MultiBytes_Master	-	-	√	√	√	√	-	√
USCI_I2C_SingleByte_Master	-	-	√	√	√	√	-	√
USCI_I2C_Slave	-	-	√	√	√	√	-	√
USCI_I2C_Slave_10bit	-	-	√	√	√	√	-	√
USCI_I2C_Wakeup_Slave	-	-	√	√	√	√	-	√
USCI_SPI_Loopback	-	-	√	√	√	√	-	√
USCI_SPI_MasterMode	-	-	√	√	√	√	-	√
USCI_SPI_PDMA_LoopTest	-	-	-	-	√	√	-	√
USCI_SPI_SlaveMode	-	-	√	√	√	√	-	√
USCI_UART_AutoBaudRate	-	-	√	√	√	√	-	√
USCI_UART_AutoFlow_Master	-	-	√	√	√	√	-	√
USCI_UART_AutoFlow_Slave	-	-	√	√	√	√	-	√
USCI_UART_PDMA	-	-	√	√	√	√	-	√
USCI_UART_RS485_Master	-	-	√	√	√	√	-	√
USCI_UART_RS485_Slave	-	-	√	√	√	√	-	√
USCI_UART_TxRxFunction	-	-	√	√	√	√	-	√
USCI_UART_Wakeup	-	-	√	√	√	√	-	√
WDT_TimeoutWakeup	√	√	√	√	√	√	√	√

AndReset								
WWDT_ReloadCounter	√	√	√	√	√	√	√	√

Note 1: This GCC sample code size exceeds 2KB.

Note 2: USB samples are only supported in M032LE3AE, M032SE3AE, M032LG6AE, M032SG6AE, M032KG6AE, M032LG8AE, M032SG8AE, M032KG8AE, M032SIAAE, M032KIAAE, M032FC1AE, M032EC1AE, M032TC1AE, M032LC2AE, M032TD2AE and M032LD2AE.

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.*