

## M3351 Series CMSIS BSP Guide

Directory Introduction for 32-bit NuMicro® Family

Please extract the “M3351\_Series\_BSP\_CMSIS\_V3.00.000.zip” file firstly and then put the “M3351\_Series\_BSP\_CMSIS\_V3.00.000” folder into the working folder.

To experience the powerful features of M3351 series in few minutes, please select the sample code to download and execute on the NuMaker board. Open the project files to build them with Keil® MDK, IAR or VSCode, and then download and trace them on the NuMaker board to see how it works.

### Directory Information

Document	Driver reference guide and revision history.
Library	Driver header and source files.
SampleCode	Driver sample code.
ThirdParty	Libraries from third parties.
Tool	Utility programs.

*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 and microprocessor 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 DOCUMENT .....</b>	<b>4</b>
<b>2 LIBRARY.....</b>	<b>5</b>
<b>3 SAMPLECODE.....</b>	<b>6</b>
<b>4 THIRDPARTY .....</b>	<b>7</b>
<b>5 TOOL .....</b>	<b>8</b>
<b>6 SAMPLECODE\CRYPTO .....</b>	<b>9</b>
<b>7 SAMPLECODE\FREERTOS.....</b>	<b>10</b>
<b>8 SAMPLECODE\ISP .....</b>	<b>11</b>
<b>9 SAMPLECODE\NUMAKER-M3351KI.....</b>	<b>12</b>
<b>10 SAMPLECODE\POWERMANAGEMENT .....</b>	<b>13</b>
<b>11 SAMPLECODE\SECUREAPPLICATION.....</b>	<b>14</b>
<b>12 SAMPLECODE\STDDRIVER .....</b>	<b>15</b>
Analog Comparator Controller (ACMP) .....	15
Basic PWM Generator and Capture Timer (BPWM).....	15
Cache Controller (CACHE).....	15
Controller Area Network with Flexible Data-Rate (CAN FD) .....	16
Clock Controller (CLK).....	16
CRC Controller (CRC) .....	16
Cryptographic Accelerator (CRYPTO) .....	17
Digital to Analog Converter (DAC).....	17
Data Flash Memory Controller (DFMC) .....	17
Enhanced 12-bit Analog-to-Digital Converter (EADC) .....	18
External Bus Interface (EBI) .....	18

---

Enhanced Input Capture Timer (ECAP) .....	19
Enhanced LED Light Strip Interface (ELLSI).....	19
Enhanced Quadrature Encoder Interface (EQEI) .....	19
Flash Memory Controller (FMC).....	19
General Purpose I/O (GPIO).....	20
I <sup>2</sup> C Serial Interface Controller (I <sup>2</sup> C) .....	21
I3C Serial Interface Controller (I3C).....	21
LED Light Strip Interface (LLSI) .....	22
PDMA Controller (PDMA) .....	22
PWM Generator and Capture Timer (PWM).....	22
Quad Serial Peripheral Interface (QSPI) .....	23
Random Number Generator (RNG).....	23
Real Timer Clock (RTC) .....	23
Serial Peripheral Interface (SPI).....	24
System Manager (SYS) .....	25
Timer Controller (TIMER).....	25
UART Interface Controller (UART).....	26
USB 1.1 Device Controller (USBD) .....	27
USB Host Controller (USBH) .....	29
Universal Serial Control Interface Controller - I <sup>2</sup> C Mode (USCI-I2C) .....	30
Universal Serial Control Interface Controller - SPI Mode (USCI-SPI) .....	31
Universal Serial Control Interface Controller - UART Mode (USCI-UART) .....	31
Watchdog Timer (WDT) .....	32
Window Watchdog Timer (WWDT) .....	32
<b>13 SAMPLECODE\TRUSTZONE .....</b>	<b>33</b>
<b>14 SAMPLECODE\XOM .....</b>	<b>34</b>

## 1 Document

CMSIS.html	Document of CMSIS.
<b>NuMicro M3351 Series CMSIS BSP Driver Reference Guide.chm</b>	This document describes the usage of drivers in M3351 Series CMSIS BSP.
<b>NuMicro M3351 Series CMSIS BSP Revision History.pdf</b>	This document shows the revision history of M3351 Series CMSIS BSP.

## 2 Library

<b>CMSIS</b>	Cortex® Microcontroller Software Interface Standard (CMSIS) V6.1 definitions by Arm® Corp.
<b>Commu</b>	Helper functions of communication protocols, e.g., xmodem.
<b>CryptoAccelerator</b>	Crypto accelerator source code for MbedTLS library.
<b>Device</b>	CMSIS compliant device header files.
<b>LlSiYcableLib</b>	LLSI library source code.
<b>StdDriver</b>	All peripheral driver header and source files.
<b>Storage</b>	Disk I/O modules for FatFs.
<b>UsbHostLib</b>	USB host library source code.

### 3 SampleCode

Crypto	Crypto sample codes using MbedTLS library.
FreeRTOS	Simple FreeRTOS™ demo codes.
Hard_Fault_Sample	Show hard fault information when hard fault happened. The hard fault handler shows some information including program counter, which is the address where the processor is executed when the hard fault occurs. The listing file (or map file) can show what function and instruction that is. 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.
NuMaker-M3351KI	Sample codes for NuMaker-M3351KI board.
PowerManagement	Sample codes for power management.
SecureApplication	Sample codes for secure application. VSCode projects require Python 3.12 at least for post-build.
StdDriver	Sample codes to demonstrate the usage of M3351 series MCU peripheral driver APIs.
Template	Project template for M3351 series MCU.
TrustZone	Demo of secure codes and non-secure codes.
XOM	Demonstrate how to create XOM library and use it.

## 4 ThirdParty

FatFs	An open-source FAT/exFAT file system library. A generic FAT file system module for small embedded systems. Its official website is <a href="http://elm-chan.org/fsw/ff/00index_e.html">http://elm-chan.org/fsw/ff/00index_e.html</a> .
FreeRTOS	Real-time operating system for microcontrollers. Its official website is <a href="http://www.freertos.org/">http://www.freertos.org/</a> .
mbedtls	Mbed TLS offers an SSL library with an intuitive API and readable source code, so you can understand what the code does. Its official website is <a href="https://tls.mbed.org/">https://tls.mbed.org/</a> .

## 5 Tool

### imgtool.exe

Used to perform the operations that are necessary to manage keys and sign images.

It is converted from Python script, <https://github.com/mcu-tools/mcuboot/tree/main/scripts/imgtool>.

## 6 SampleCode\Crypto

<b>mbedTLS_AES</b>	Show how mbedTLS AES function works.
<b>mbedTLS_SHA256</b>	Show how mbedTLS SHA256 function works.

## 7 SampleCode\FreeRTOS

Blinky	This project provides two demo applications. A simple blinky style project, and a more comprehensive test and demo application.
TicklessIdle	Show how to enable FreeRTOS™ tickless idle mode, and enable RTC for long duration task idle.
TrustZone	Show how to enable FreeRTOS™ on TrustZone application.

## 8 SampleCode\ISP

<b>common</b>	Simplified system files, drivers and ISP command handle function due to LDROM code size limitation.
<b>ISP_CAN</b>	In-System-Programming Sample code through CAN interface.
<b>ISP_DFU</b>	In-System-Programming Sample code through USB interface and following Device Firmware Upgrade Class Specification.
<b>ISP_HID</b>	In-System-Programming Sample code through USB HID interface.
<b>ISP_I2C</b>	In-System-Programming Sample code through I <sup>2</sup> C interface.
<b>ISP_MSC</b>	In-System-Programming Sample code through USB interface and following Mass Storage Class Specification.
<b>ISP_RS485</b>	In-System-Programming Sample code through RS485 interface.
<b>ISP_SPI</b>	In-System-Programming Sample code through SPI interface.
<b>ISP_UART</b>	In-System-Programming Sample code through UART interface.

## 9 SampleCode\NuMaker-M3351KI

Xmodem

Demonstrate how to transfer data with UART Xmodem.

## 10 SampleCode\PowerManagement

The M3351 series MCU provides some power modes with different power consumption level and wake-up time. For more information, please refer to the application note.

SYS_PowerDown_MinCurrent	Demonstrate how to minimize power consumption when entering Power-down mode.
SYS_PowerDownMode	Show how to enter to different Power-down mode and wake-up by Wake-up Timer.
SYS_SPDMode_Wakeup	Show how to wake up system from SPD Power-down mode by different wakeup sources.
SYS_SPDMode_WakeupAndReturn	Show how to continue executing code after wake-up from SPD Power-down mode by SRAM data retention function.
SYS_SPDMode_WakeupVTOR	Show how to execute code after wake-up from SPD Power-down mode by VTOR function.

## 11 SampleCode\SecureApplication

<b>SecureBootDemo</b>	Demonstrate how to generate the first booting image, NuBL2. After NuBL2 runs, NuBL2 will authenticate NuBL32 and NuBL33 then jump to execute in NuBL32.
<b>SecureISPDemo<sup>2</sup></b>	Demonstrate how to initialize a SecureISP client mode. This sample code needs to work with <a href="#">USBH_SecureISP</a> sample code.
<b>USBH_SecureISP</b>	Demonstrate how to initialize a SecureISP server mode via USB host connected to SecureISPDemo. This sample code needs to work with <a href="#">SecureISPDemo</a> sample code.

<sup>2</sup> SecureISPDemo and USBH\_SecureISP samples disable CACHE.

## 12 SampleCode\StdDriver

### Analog Comparator Controller (ACMP)

ACMP_ComapreDAC	Demonstrate ACMP comparison by comparing ACMP1_P0 input and DAC voltage and show the result on UART console.
ACMP_ComapreVBG	Demonstrate ACMP comparison by comparing ACMP1_P0 input and VBG voltage and show 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.

### Basic PWM Generator and Capture Timer (BPWM)

BPWM_Capture	Use BPWM0 channel 0 to capture the BPWM1 channel 2 waveform.
BPWM_DoubleBuffer	Change duty cycle and period of output waveform by BPWM double buffer function.
BPWM_OutputWaveform	Demonstrate how to use BPWM counter output waveform.
BPWM_SwitchDuty	Change duty cycle of output waveform by configured period.
BPWM_SyncStart	Demonstrate how to use BPWM counter synchronous start function.

### Cache Controller (CACHE)

CACHE_CheckParityError	Demonstrate Cache parity error detection.
CACHE_Performance	Demonstrate the performance of CACHE.

## Controller Area Network with Flexible Data-Rate (CAN FD)

CANFD_CAN_Loopback	Use CAN mode function to do internal loopback test.
CANFD_CAN_MonitorMode	Use CAN Monitor mode to monitor the CAN bus communication test.
CANFD_CAN_TxRx	Transmit and receive CAN messages through CAN interface.
CANFD_CAN_TxRxINT	An example of interrupt control using CAN bus communication.
CANFD_CANFD_Loopback	Use CAN FD mode function to do internal loopback test.
CANFD_CANFD_MonitorMode	Use CAN FD Monitor mode to monitor the CAN bus communication test.
CANFD_CANFD_TxRx	Transmit and receive CAN FD messages through CAN interface.
CANFD_CANFD_TxRxINT	An example of interrupt control using CAN FD bus communication.

## Clock Controller (CLK)

CLK_ClockDetector	Show the usage of clock fail detector and clock frequency monitor function.
-------------------	---

## 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_DMA	Implement CRC in CRC-32 mode and get the CRC checksum result.
CRC_Polynomial	Demonstrate how to use polynomial mode and get the CRC checksum result.

## Cryptographic Accelerator (CRYPTO)

CRYPTO_AES	Show Crypto IP AES-128 ECB mode encrypt/decrypt function.
CRYPTO_AES_CCM	Demonstrate how to encrypt/decrypt data by AES CCM.
CRYPTO_AES_GCM	Demonstrate how to encrypt/decrypt data by AES GCM.
CRYPTO_HMAC	Show Crypto IP HMAC function.
CRYPTO_SHA	Use Crypto IP SHA engine to run through known answer SHA1 test vectors.

## Digital to Analog Converter (DAC)

DAC_ExtPinTrigger	Demonstrate how to trigger DAC conversion by external pin.
DAC_PDMA_PWMTrigger	Demonstrate how to use PDMA and trigger DAC0 by PWM.
DAC_PDMA_TimerTrigger	Demonstrate how to PDMA and trigger DAC by Timer.
DAC_PWMTrigger	Demonstrate how to trigger DAC by PWM.
DAC_SoftwareTrigger	Demonstrate how to trigger DAC conversion by software.
DAC_TimerTrigger	Demonstrate how to trigger DAC by Timer.

## Data Flash Memory Controller (DFMC)

DFMC_CRC32	Demonstrate how to use DFMC CRC32 ISP command to calculate the CRC32 checksum of DFMC data flash.
DFMC_EEPROM	Demonstrate how to configure, initialize, read and write DFMC EEPROM.
DFMC_NonBlocking	Show DFMC erase, read, and write functions in non-blocking mode.
DFMC_ReadAllOne	Demonstrate how to use DFMC Read-All-One ISP command to verify data flash pages are all 0xFFFFFFFF or not.

<b>DFMC_RW</b>	Show DFMC read flash IDs, erase, read, and write functions.
----------------	---

## Enhanced 12-bit Analog-to-Digital Converter (EADC)

<b>EADC_Accumulate</b>	Demonstrate how to get accumulated conversion result.
<b>EADC_ADINT_Trigger</b>	Use ADINT interrupt to do the EADC continuous scan conversion.
<b>EADC_Average</b>	Demonstrate how to get average conversion result.
<b>EADC_AverageCMP</b>	Demonstrate how to compare average conversion result.
<b>EADC_BandGap</b>	Convert Band-gap and print conversion result.
<b>EADC_BPWM_Trigger</b>	Demonstrate how to trigger EADC by BPWM.
<b>EADC_PDMA_BPWM_Trigger</b>	Demonstrate how to trigger EADC by BPWM and transfer conversion data by PDMA.
<b>EADC_PDMA_PWM_Trigger</b>	Demonstrate how to trigger EADC by PWM and transfer conversion data by PDMA.
<b>EADC_Pending_Priority</b>	Demonstrate how to trigger multiple sample modules and got conversion results in order of priority.
<b>EADC_PWM_Trigger</b>	Demonstrate how to trigger EADC by PWM.
<b>EADC_ResultMonitor</b>	Monitor the conversion result of channel 2 by the digital compare function.
<b>EADC_SWTRG_Trigger</b>	Trigger EADC by writing EADC software trigger register.
<b>EADC_TempSensor</b>	Convert temperature sensor and print conversion result.
<b>EADC_Timer_Trigger</b>	Show how to trigger EADC by Timer.

## External Bus Interface (EBI)

<b>EBI_NOR</b>	Configure EBI interface to access NOR Flash connected on EBI interface.
----------------	---

<b>EBI_SRAM</b>	Configure EBI interface to access SRAM connected on EBI interface.
-----------------	--

## Enhanced Input Capture Timer (ECAP)

<b>ECAP_GetEQEIFreq</b>	Show how to use ECAP interface to get EQEI frequency.
<b>ECAP_GetInputFreq</b>	Show how to use ECAP interface to get input frequency

## Enhanced LED Light Strip Interface (ELLSI)

<b>ELLSI_Marquee</b>	This is an ELLSI demo for marquee display in software mode. It needs to be used with WS2812 LED strip.
<b>ELLSI_Marquee_Y_Cable</b>	This is an ELLSI demo for marquee display in software mode. It needs to be used with AP6112Y LED strip. The AP6112Y LED strip will be initialized through manual software settings.
<b>ELLSI_PDMA_Marquee</b>	This is an ELLSI demo for marquee display in PDMA mode. It needs to be used with WS2812 LED strip.
<b>ELLSI_PDMA_Marquee_Y_Cable</b>	This is an ELLSI demo for marquee display in PDMA mode. It needs to be used with AP6112Y LED strip. The AP6112Y LED strip will be initialized through automatic hardware settings.

## Enhanced Quadrature Encoder Interface (EQEI)

<b>EQEI_CompareMatch</b>	Show the usage of EQEI compare function.
<b>EQEI_PeriodicINT</b>	Show the usage of EQEI Unit Timer function.

## Flash Memory Controller (FMC)

<b>FMC_APWPROT</b>	Demonstrate how to use FMC APROM Protect function.
<b>FMC_CRC32</b>	Demonstrate how to use FMC CRC32 ISP command to calculate the CRC32 checksum of APROM and LDROM.

<b>FMC_DualBank</b>	Demonstrate how dual processes work in dual bank flash architecture.
<b>FMC_DualBankFwUpdate</b>	Implement a firmware update mechanism based on dual bank flash architecture.
<b>FMC_ECC</b>	Demonstrated FMC ECC status and ECC error fault address.
<b>FMC_ExeInSRAM</b>	Implement a code and execute it in SRAM to program embedded Flash.
<b>FMC_FwUpdateApplication</b>	Firmware update sample code.
<b>FMC_IAP</b>	Show how to call LDROM function from APROM.
<b>FMC_MultiBoot</b>	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.
<b>FMC_ReadAllOne</b>	Demonstrate how to use FMC Read-All-One ISP command to check whether the specified APROM or LDROM region are all 0xFFFFFFFF or not.
<b>FMC_RW</b>	Show FMC read Flash IDs, erase, read, and write functions.
<b>FMC_XOM</b>	Show how to configure and set up an XOM region then perform XOM function.

## General Purpose I/O (GPIO)

<b>GPIO_EINTAndDebounce</b>	Show the usage of GPIO external interrupt function and debounce function.
<b>GPIO_EINTTriggerPDMA</b>	Show the usage of GPIO EINT trigger PDMA function.
<b>GPIO_EINTTriggerPWM</b>	Show the usage of GPIO EINT trigger PWM 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.
-----------------------	--

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

<b>I2C_DoubleBuffer_Slave</b>	Demonstrate how to set I <sup>2</sup> C two-level buffer in Slave mode to receive data from a master. This sample code needs to work with <a href="#">I2C_MultiBytes_Master</a> sample code.
<b>I2C_EEPROM</b>	Show how to use I <sup>2</sup> C interface to access EEPROM.
<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 <a href="#">I2C_Slave</a> sample code.
<b>I2C_MultiBytes_Master</b>	Demonstrate how to use multi-byte API to access slave. This sample code needs to work with <a href="#">I2C_Slave</a> sample code.
<b>I2C_PDMA_TRx</b>	Demonstrate I <sup>2</sup> C PDMA mode and need to connect I2C0 (master) and I2C1 (slave).
<b>I2C_Slave</b>	Demonstrate how to set I <sup>2</sup> C in Slave mode to receive data from a master. This sample code needs to work with <a href="#">I2C_Master</a> sample code.
<b>I2C_SMBus</b>	Demonstrate how to control SMBus interface and use SMBus protocol between Host and Slave.
<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 could work with <a href="#">I2C_Master</a> sample code.

## I3C Serial Interface Controller (I3C)

<b>I3C_Controller</b>	Demonstrates how to initialize the I3C Controller and perform operations to the I3C Target. This sample code needs to use two boards.
<b>I3C_Target</b>	Demonstrates how to initialize the I3C Target and response command from the I3C Controller. This sample code needs

to use two boards.

## LED Light Strip Interface (LLSI)

LLSI_Marquee	This is a LLSI demo for marquee display in software mode. It needs to be used with WS2812 LED strip.
LLSI_PDMA_Marquee	This is a LLSI demo for marquee display in PDMA mode. It needs to be used with WS2812 LED strip.
LLSI_Y_Cable_Control	This is a LLSI demo for Y-cable control. It needs to be used with AP6112Y LED strip.

## PDMA Controller (PDMA)

PDMA_BasicMode	Use PDMA0 channel 2 to transfer data from memory to memory.
PDMA_ScatterGather	Use PDMA0 channel 4 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).

## PWM Generator and Capture Timer (PWM)

PWM_AccumulatorINT_TriggerPDMA	Demonstrate how to use PWM accumulator interrupt trigger PDMA.
PWM_AccumulatorStopMode	Demonstrate PWM accumulator stop mode.
PWM_Brake	Demonstrate how to use PWM brake function.
PWM_Capture	Capture the PWM1 Channel 0 waveform by PWM1 Channel 2.
PWM_DeadTime	Demonstrate how to use PWM Dead Zone function.
PWM_DoubleBuffer	Change duty cycle and period of output waveform by PWM Double Buffer function.

PWM_OutputWaveform	Demonstrate how to use PWM output waveform.
PWM_PDMA_Capture	Capture the PWM1 Channel 0 waveform by PWM1 Channel 2 and use PDMA to transfer captured data.
PWM_SwitchDuty	Change duty cycle of output waveform by configured period.
PWM_SyncStart	Demonstrate how to use PWM counter synchronous start function.

## Quad Serial Peripheral Interface (QSPI)

QSPI_DualMode_Flash	Access SPI Flash using QSPI dual mode.
QSPI_PDMA_Master	Configure QSPI0 as Master mode and demonstrate how to communicate with an off-chip QSPI Slave device with FIFO mode. This sample code needs to work with <a href="#">QSPI PDMA Slave</a> sample code.
QSPI_PDMA_Slave	Configure QSPI as Slave mode and demonstrate how to communicate with an off-chip QSPI Master device with FIFO mode. This sample code needs to work with <a href="#">QSPI PDMA Master</a> sample code.
QSPI_QuadMode_Flash	Access SPI Flash using QSPI quad mode.
QSPI_Slave3Wire	Configure QSPI as Slave 3-wire mode and demonstrate how to communicate with an off-chip SPI Master device with FIFO mode. This sample code needs to work with <a href="#">SPI MasterFIFOMode</a> sample code.

## Random Number Generator (RNG)

RNG_EntropyPoll	Generate entropy from hardware entropy source.
RNG_Random	Generate random numbers.

## Real Timer Clock (RTC)

RTC_Alarm_Test	Demonstrate the RTC alarm function. It sets an alarm 10
----------------	---

	seconds after execution.
<b>RTC_Alarm_Wakeup</b>	Use RTC alarm interrupt event to wake up system.
<b>RTC_Time_Display</b>	Demonstrate the RTC function and display current time to the UART console.

## Serial Peripheral Interface (SPI)

<b>SPI_Flash</b>	Access SPI Flash through SPI interface.
<b>SPI_HalfDuplex</b>	Demonstrate SPI half-duplex mode. Configure SPI0 as master mode and SPI1 as slave mode. Both SPI0 and SPI1 are half-duplex mode.
<b>SPI_Loopback</b>	Implement SPI Master loop back transfer. This sample code needs to connect MISO pin and MOSI pin together. It will compare the received data with transmitted data.
<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 could work with <a href="#">SPI_SlaveFIFOMode</a> sample code.
<b>SPI_PDMA_LoopTest</b>	SPI read/write demo in PDMA mode. Connect 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 <a href="#">SPI_MasterFIFOMode</a> sample code.
<b>SPII2S_Master</b>	Configure SPI as I <sup>2</sup> S master mode and demonstrate how I <sup>2</sup> S works in master mode. This sample code needs to work with <a href="#">SPII2S_Slave</a> sample code.
<b>SPII2S_PDMA_Codec</b>	An I <sup>2</sup> S demo with PDMA function connected with audio codec.
<b>SPII2S_PDMA_Play</b>	An I <sup>2</sup> S demo for playing data and demonstrating how I <sup>2</sup> S works with PDMA.
<b>SPII2S_PDMA_PlayRecord</b>	An I <sup>2</sup> S demo for playing and recording data with PDMA

	function.
<b>SPII2S_PDMA_Record</b>	An I <sup>2</sup> S demo for recording data and demonstrating how I <sup>2</sup> S works with PDMA.
<b>SPII2S_Slave</b>	Configure SPI as I <sup>2</sup> S slave mode and demonstrate how I <sup>2</sup> S works in slave mode. This sample code needs to work with <a href="#">SPII2S_Master</a> sample code.

## System Manager (SYS)

<b>SYS_BODWakeup</b>	Show 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 CLKO pin.
<b>SYS_TrimHIRC</b>	Demonstrate how to use LXT to trim HIRC.

## Timer Controller (TIMER)

<b>TIMER_ACMPTrigger</b>	Use ACMP to trigger Timer counter reset mode.
<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 TM0_EXT pin to demonstrate Timer free counting mode function. And display the measured input frequency to UART console.
<b>TIMER_InterTimerTriggerMode</b>	Use TM0 pin to demonstrate inter-timer trigger mode function and 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_PWM_AccumulatorINT StopMode</b>	Demonstrate TIMER PWM accumulator interrupt to stop counting.
<b>TIMER_PWM_AccumulatorINT TriggerPDMA</b>	Demonstrate TIMER PWM accumulator interrupt to trigger PDMA transfer.
<b>TIMER_PWM_ChangeDuty</b>	Change duty cycle and period of output waveform in PWM down count type.
<b>TIMER_PWM_OutputWavefor m</b>	Demonstrate output different duty waveform in Timer0~Timer3 PWM.
<b>TIMER_TimeoutWakeup</b>	Use Timer to wake up system from Power-down mode periodically.
<b>TIMER_ToggleOut</b>	Demonstrate the Timer toggle out function.

## 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_LIN</b>	Transmit LIN frame including header and response in UART LIN 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.

## USB 1.1 Device Controller (USBD)

<b>USBD_Audio_Codec</b>	Demonstrate how to implement a USB audio class device.
<b>USBD_Audio_Headset</b>	Demonstrate how to implement a USB audio class device. Codec is used in this sample code to play the audio data from Host. It also supports to record data from codec to Host.
<b>USBD_HID_Keyboard</b>	Demonstrate how to implement a USB keyboard device. It supports to use GPIO to simulate key input.
<b>USBD_HID_Mouse</b>	Show how to implement a USB mouse device. The mouse cursor will move automatically when this mouse device connecting to PC by USB.
<b>USBD_HID_MouseKeyboard</b>	Simulate an USB HID mouse and HID keyboard. Mouse draws circle on the screen and Keyboard uses GPIO to simulate key input.
<b>USBD_HID_RemoteWakeups</b>	Demonstrate how to implement a USB mouse device. It uses PA0 ~ PA5 to control mouse direction and mouse key. It also supports USB suspend and remote wakeup.
<b>USBD_HID_Touch</b>	Demonstrate how to implement a USB touch digitizer device. Two lines demo in Paint.
<b>USBD_HID_Transfer</b>	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.
<b>USBD_HID_Transfer_And_Keyboard</b>	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.
<b>USBD_HID_Transfer_And_MSC</b>	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.
<b>USBD_HID_Transfer_CTRL</b>	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.
<b>USBD_MassStorage_CDROM</b>	Demonstrate the emulation of USB Mass-Storage device, CD-ROM.
<b>USBD_MassStorage_Flash</b>	Use Flash as storage to implement a USB Mass-Storage device.
<b>USBD_MassStorage_SRAM</b>	Use internal SRAM as backend storage media to simulate a USB pen drive.
<b>USBD_MicroPrinter</b>	Demonstrate how to implement a USB micro printer device.
<b>USBD_Printer_And_HID_Transfer</b>	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.
<b>USBD_VCOM_And_HID_Keyboard</b>	Demonstrate how to implement a composite device of VCOM and HID keyboard.
<b>USBD_VCOM_And_HID_Transfer</b>	Demonstrate how to implement a composite device of 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.
<b>USBD_VCOM_And_MassStorage</b>	Demonstrate how to implement a composite device of VCOM and mass storage.
<b>USBD_VCOM_MultiPort</b>	Demonstrate how to implement a USB multiple virtual COM port device.
<b>USBD_VCOM_SerialEmulator</b>	Demonstrate how to implement a USB virtual COM port device.
<b>USBD_VCOM_SerialEmulator_DoubleBuffer</b>	Demonstrate how to implement a USB virtual COM port device using double buffer mode.
<b>USBD_VENDOR_LBK</b>	This sample code works as a proprietary Vendor LBK device. It's created for <a href="#">USBH_VENDOR_LBK</a> of this BSP. Vendor LBK device includes Control, Bulk, Interrupt, and

Isochronous in/out endpoint pairs. Each endpoint pair receive data from host via the out-endpoint and send data back to host via the in-endpoint.

## USB Host Controller (USBH)

<b>USBH_DEV_CONN<sup>4</sup></b>	Use connect/disconnect callback functions to handle of device connect and disconnect events.
<b>USBH_FirmwareUpdate</b>	Automatically search and read new firmware from USB drive, if found, update APROM Flash with it.
<b>USBH_HID</b>	Use USB Host core driver and HID driver. Demonstrate how to submit HID class request and read data from interrupt pipe. This sample code supports dynamic device plug/un-plug and multiple HID devices.
<b>USBH_HID_Keyboard</b>	Demonstrate reading key inputs from USB keyboards. This sample code includes an USB keyboard driver which is based on the HID driver.
<b>USBH_HID_MouseKeyboard</b>	Demonstrate how to support USB mouse and keyboard input.
<b>USBH_MassStorage</b>	Use a command-shell-like interface to demonstrate how to use USBH mass storage driver and make it work as a disk driver under the FATFS file system.
<b>USBH_UAC</b>	Demonstrate how to use USBH Audio Class driver. It shows the mute, volume, auto-gain, channel, and sampling rate control.
<b>USBH_UAC_HID</b>	Show how to use USBH Audio Class driver and HID driver at the same time. The target device is a Game Audio (UAC+HID composite device).
<b>USBH_UAC_Loopback</b>	Receive audio data from an UAC device, and immediately send back to the UAC device.

<sup>4</sup> USBH engine clock source of USBH samples uses HIRC instead of PLL.

<b>USBH_VCOM</b>	Demonstrate how to use the USB Host core driver and CDC driver to connect a CDC class VCOM device.
<b>USBH_VCOM_MassStorage</b>	This sample uses a command-shell-like interface to demonstrate how to use USBH mass storage driver and make it working as a disk driver under FATFS file system. It also demonstrates how to use CDC driver to connect a CDC class VCOM device.
<b>USBH_VENDOR_LBK</b>	Show how to do transfer on a known device with a vendor driver. This sample code requires a USB device running sample <a href="#">USBD_VENDOR_LBK</a> to be connected.

## Universal Serial Control Interface Controller - I<sup>2</sup>C Mode (USCI-I2C)

<b>USCI_I2C_EEPROM</b>	Demonstrate how to access EEPROM through a USCI_I2C interface.
<b>USCI_I2C_Master</b>	Demonstrate how a Master accesses Slave. This sample code needs to work with <a href="#">USCI_I2C_Slave</a> sample code.
<b>USCI_I2C_Master_10bit</b>	Demonstrate how a Master uses 10-bit addressing access Slave. This sample code needs to work with <a href="#">USCI_I2C_Slave_10bit</a> sample code.
<b>USCI_I2C_Monitor</b>	Demonstrate how USCI_I2C monitors transmission between I <sup>2</sup> C Master and I <sup>2</sup> C Slave.
<b>USCI_I2C_MultiBytes_Master</b>	Demonstrate how to use multi-byte API to access slave. This sample code needs to work with the <a href="#">USCI_I2C_Slave</a> sample code.
<b>USCI_I2C_Slave</b>	Demonstrate how to set USCI_I2C in slave mode to receive the data from a Master. This sample code needs to work with <a href="#">USCI_I2C_Master</a> sample code.
<b>USCI_I2C_Slave_10bit</b>	Demonstrate how to set USCI_I2C in 10-bit addressing slave mode to receive the data from a Master. This sample code needs to work with <a href="#">USCI_I2C_Master_10bit</a> sample code.
<b>USCI_I2C_Wakeup_Slave</b>	Demonstrate how to set USCI_I2C to wake up MCU from Power-down mode. This sample code needs to work with

---

[USCI\\_I2C\\_Master](#) sample code.

---

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

<b>USCI_SPI_Loopback</b>	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.
<b>USCI_SPI_MasterMode</b>	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 <a href="#">USCI_SPI_SlaveMode</a> sample code.
<b>USCI_SPI_PDMA_LoopTest</b>	Demonstrate 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.
<b>USCI_SPI_SlaveMode</b>	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 <a href="#">USCI_SPI_MasterMode</a> sample code.

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

<b>USCI_UART_AutoBaudRate</b>	Show how to use auto baud rate detection function.
<b>USCI_UART_Autoflow</b>	Transmit and receive data with auto flow control.
<b>USCI_UART_PDMA</b>	This is a USCI_UART PDMA demo and needs to connect USCI_UART Tx and Rx.
<b>USCI_UART_RS485</b>	Transmit and receive data in RS485 mode.
<b>USCI_UART_TxRxFunction</b>	Transmit and receive data from PC terminal through RS232 interface.
<b>USCI_UART_Wakeup</b>	Show how to wake up system from Power-down mode by USCI interrupt in UART mode.

## 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\_CompareINT**

Show how to reload the WWDT counter value.

**WWDT\_ReloadCounter**

Demonstrate how to reload the WWDT counter value without resetting the MCU.

## 13 SampleCode\TrustZone

<b>HardFault</b>	Show the hard fault usages in both secure and non-secure code.
<b>Template</b>	Demonstrate how to implement code for secure and non-secure.

## 14 SampleCode\XOM

XOMLib	Demonstrate how to create XOM library.
--------	--

XOMLibDemo	Demonstrate how to use <a href="#"><u>XOMLib</u></a> .
------------	--

### **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.