

## M460 Series CMSIS BSP Guide

Directory Introduction for 32-bit NuMicro™ Family

### Directory Information

Please extract the “M460\_Series\_BSP\_CMSIS\_V3.00.001.zip” file firstly, and then put the “M460\_Series\_BSP\_CMSIS\_V3.00.001” folder into the working folder (e.g. .Nuvoton\BSP Library\).

To experience the powerful features of M460 in few minutes, please select the sample code to download and execute on the M460 board. Open the project files to build them with Keil® MDK, IAR or Eclipse, and then download and trace them on the M460 board to see how it works.

This BSP folder contents:

<b>Document\</b>	Device driver reference manual and reversion history.
<b>Library\</b>	Device driver header and source files.
<b>SampleCode\</b>	Device driver sample code.
<b>ThirdParty\</b>	Libraries of third parties.

*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>4</b>
<b>2</b>	<b>LIBRARY .....</b>	<b>5</b>
<b>3</b>	<b>SAMPLE CODE .....</b>	<b>6</b>
<b>4</b>	<b>THIRDPARTY .....</b>	<b>7</b>
<b>5</b>	<b>SAMPLECODE\CORTEXM4.....</b>	<b>9</b>
<b>6</b>	<b>SAMPLECODE\CRYPTO.....</b>	<b>10</b>
<b>7</b>	<b>SAMPLECODE\ISP .....</b>	<b>11</b>
<b>8</b>	<b>SAMPLECODE\NUMAKER_M467HJ .....</b>	<b>12</b>
<b>9</b>	<b>SAMPLECODE\POWERMANAGEMENT .....</b>	<b>14</b>
<b>10</b>	<b>SAMPLECODE\STDDRIVER.....</b>	<b>15</b>
	System Manager (SYS) .....	15
	Clock Controller (CLK) .....	15
	Flash Memory Controller (FMC).....	15
	General Purpose I/O (GPIO).....	16
	HyperBus Interface Controller (HBI) .....	16
	PDMA Controller (PDMA) .....	16
	Timer Controller (TIMER) .....	17
	Watchdog Timer (WDT) .....	18
	Window Watchdog Timer (WWDT) .....	18
	Real Timer Clock (RTC) .....	18
	Basic PWM Generator and Capture Timer (BPWM).....	18
	Enhance PWM Generator and Capture Timer (EPWM) .....	19
	Enhanced Quadrature Encoder Interface (EQEI) .....	19
	UART Interface Controller (UART).....	19
	Smart Card Host Interface (SC).....	20
	I <sup>2</sup> S Controller (I <sup>2</sup> S).....	20
	Serial Peripheral Interface (SPI) .....	21

SPI Synchronous Serial Interface Controller (SPIM).....	22
Programmable Serial I/O (PSIO).....	22
Quad Serial Peripheral Interface (QSPI) .....	22
I <sup>2</sup> C Serial Interface Controller (I <sup>2</sup> C).....	23
Universal Serial Control Interface Controller - UART Mode (USCI-UART) .....	23
Universal Serial Control Interface Controller - SPI Mode (USCI-SPI).....	24
Universal Serial Control Interface Controller - I <sup>2</sup> C Mode (USCI-I2C).....	24
External Bus Interface (EBI) .....	25
USB 1.1 Device Controller (USBD) .....	25
High Speed USB 2.0 Device Controller (HSUSBD) .....	27
USB 1.1/2.0 Host Controller (HSUSBH) .....	28
USB On-The-Go (OTG).....	29
High Speed USB On-The-Go (HSOTG) .....	29
Ethernet MAC Controller (EMAC) .....	29
CRC Controller (CRC) .....	29
Enhance 12-bit Analog-to-Digital Converter (EADC) .....	30
Digital-to-Analog Converter (DAC) .....	30
Analog Comparator Controller (ACMP) .....	31
Biphase Mark Coding Controller (BMC) .....	31
Controller Area Network (CAN).....	31
Camera Capture Interface Controller (CCAP).....	31
Key Store (KS).....	32
Cryptographic Accelerator (CRYPTO) .....	32
Enhanced Input Capture Timer (ECAP) .....	33
Random Number Generator (RNG).....	33
SCUART .....	33
Secure Digital Host Controller (SDH).....	33
<b>11 SAMPLECODE\TEMPLATE .....</b>	<b>34</b>
<b>12 SAMPLECODE\XOM.....</b>	<b>35</b>

## 1 Document

<b>CMSIS.html</b>	<p>Introduction of CMSIS version 5. CMSIS components included CMSIS-CORE, CMSIS-Driver, CMSIS-DSP, etc.</p> <ul style="list-style-type: none"> <li>● CMSIS-CORE: API for the Cortex-M0 processor core and peripherals.</li> <li>● CMSIS-Driver: Defines generic peripheral driver interfaces for middleware making it reusable across supported devices.</li> <li>● CMSIS-DSP: DSP Library Collection with over 60 Functions for various data types: fix-point (fractional q7, q15, q31) and single precision floating-point (32-bit).</li> </ul>
<b>Revision History.pdf</b>	The revision history of M460 Series BSP.
<b>NuMicro M460 Series Driver Reference Guide.chm</b>	The usage of drivers in M460 Series BSP.

## 2 Library

<b>CMSIS</b>	Cortex® Microcontroller Software Interface Standard (CMSIS) V5 definitions by ARM® Corp.
<b>CryptoAccelerator</b>	Library for mbed TLS crypto.
<b>Device</b>	CMSIS compliant device header file.
<b>NuMaker</b>	Specific libraries for M460 NuMaker board.
<b>SmartcardLib</b>	Library for accessing a smartcard.
<b>StdDriver</b>	All peripheral driver header and source files.
<b>UsbHostLib</b>	USB host library source code.

### 3 Sample Code

<b>CortexM4</b>	Cortex®-M4 sample code.
<b>Crypto</b>	Crypto sample code using MbedTLS library.
<b>FreeRTOS</b>	Simple FreeRTOS™ demo code.
<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 occur. 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>NuMaker_M467HJ</b>	Sample codes for NuMaker-PFM-M460 board.
<b>PowerManagement</b>	Power management sample code.
<b>StdDriver</b>	Demonstrate the usage of M460 series MCU peripheral driver APIs.
<b>Template</b>	A project template for M460.
<b>XOM</b>	Demonstrate how to create XOM library and use it.

## 4 ThirdParty

<b>FatFs</b>	<p>An open source FAT/exFAT file system library.</p> <p>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>.</p>
<b>FreeRTOS</b>	<p>A real time operating system available for free download. Its official website is: <a href="http://www.freertos.org/">http://www.freertos.org/</a>.</p>
<b>libjpeg</b>	<p>A software implements JPEG baseline, extended-sequential, and progressive compression processes maintained and published by the Independent JPEG Group (IJG). Its official website is: <a href="http://ijg.org/">http://ijg.org/</a>.</p>
<b>LibMAD</b>	<p>A MPEG audio decoder library that currently supports MPEG-1 and the MPEG-2 extension to lower sampling frequencies, as well as the de facto MPEG 2.5 format. All three audio layers — Layer I, Layer II, and Layer III (i.e. MP3) are fully implemented. This library is distributed under GPL license. Please contact Underbit Technologies (<a href="http://www.underbit.com/">http://www.underbit.com/</a>) for the commercial license.</p>
<b>lwIP</b>	<p>A widely used open source TCP/IP stack designed for embedded systems. Its official website is: <a href="http://savannah.nongnu.org/projects/lwip/">http://savannah.nongnu.org/projects/lwip/</a>.</p>
<b>mbedtls-2.13.0</b>	<p>A portable, easy to use, readable and flexible SSL library. Unless specifically indicated otherwise files are licensed under the Apache 2.0 license. The official website: <a href="https://tls.mbed.org/">https://tls.mbed.org/</a>.</p>
<b>mbedtls-3.1.0</b>	<p>mbed TLS offers an SSL library with an intuitive API and readable source code, so you can actually understand what the code does. Also the mbed TLS modules are as loosely coupled as possible and written in the portable C language. This allows you to use the parts you need, without having to include the total library. The official website: <a href="https://tls.mbed.org/">https://tls.mbed.org/</a>.</p>
<b>paho.mqtt.embedded-c</b>	<p>Eclipse Paho MQTT C/C++ client for Embedded platforms. Its official website is: <a href="https://www.eclipse.org/paho/clients/c/embedded/">https://www.eclipse.org/paho/clients/c/embedded/</a>.</p>

**shine**

A blazing fast MP3 encoding library implemented in fixed-point arithmetic. Its official website is:  
<https://github.com/toots/shine>.



## 5 SampleCode\CortexM4

BitBand	Demonstrate the usage of Cortex®-M4 Bit-band.
DSP_FFT	Demonstrate how to call ARM CMSIS DSP library to calculate FFT.
MPU	Demonstrate the usage of Cortex®-M4 MPU.

## 6 SampleCode\Crypto

<b>mbedTLS_AES</b>	Show how mbedTLS AES function works.
<b>mbedTLS_ECDH</b>	Show how mbedTLS ECDH function works.
<b>mbedTLS_ECDSA</b>	Show how mbedTLS ECDSA function works.
<b>mbedTLS_RSA</b>	Show how mbedTLS RSA function works.
<b>mbedTLS_SHA256</b>	Show how mbedTLS SHA256 function works.

## 7 SampleCode\ISP

<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_DFU_20</b>	In-System-Programming Sample code through HSUSBD interface and following Device Firmware Upgrade Class Specification.
<b>ISP_HID</b>	In-System-Programming Sample code through USB HID interface.
<b>ISP_HID_20</b>	In-System-Programming Sample code through HSUSBD HID interface.
<b>ISP_I2C</b>	In-System-Programming Sample code through I <sup>2</sup> C interface.
<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.

## 8 SampleCode\NuMaker\_M467HJ

<b>Blinky</b>	A simple LED toggle sample code. It could be used as the startup of M460 NuMaker board.
<b>lwIP</b>	Common drivers for LwIP samples.
<b>LwIP_httpd_netconn</b>	A simple HTTP server that demonstrates LwIP netconn API under FreeRTOS™. This HTTP server's IP address can be configured statically to 192.168.1.2, or assigned by DHCP server.
<b>LwIP_httpd_socket</b>	A simple HTTP server that demonstrates LwIP socket API under FreeRTOS™. This HTTP server's IP address can be configured statically to 192.168.1.2, or assigned by DHCP server.
<b>LwIP_MQTT</b>	A MQTT client sample. The lower level MQTT client functions are from eclipse paho.
<b>LwIP_SSL_Client</b>	A simple HTTPS client that sends a fixed request and displays the response.
<b>LwIP_SSL_Server</b>	A simple HTTPS server that sends a fixed response. It serves a single client at a time.
<b>LwIP_TCP_EchoClient</b>	A TCP echo client which is implemented with LwIP under FreeRTOS™. This client sends "nuvoton" string to server.
<b>LwIP_TCP_EchoServer</b>	A TCP echo server which is implemented with LwIP under FreeRTOS™. The server listens to port 80, and its IP address can be configured statically to 192.168.1.2 or assigned by DHCP server. This server replies "Hello World!!" if the received string is "nuvoton", otherwise replies "Wrong Password!!" to its client.
<b>LwIP_tftp_client</b>	A TFTP client sample that can receive a file from TFTP server or send a file to TFTP server.
<b>LwIP_tftp_server</b>	A TFTP server sample that communicates with TFTP client.
<b>LwIP_UDP_EchoClient</b>	A UDP echo client which is implemented with LwIP under FreeRTOS™. This client sends "Hi there..." string to the server.

<b>LwIP_UDP_EchoServer</b>	A UDP echo server which is implemented with LwIP under FreeRTOS™. The echo server listens to port 80, and its IP address can be configured statically to 192.168.1.2 or assigned by DHCP server. After receiving any string from its peer, this server echoes that string back.
<b>MP3_Player</b>	MP3 player sample plays MP3 files stored on SD memory card.
<b>MP3_Recorder</b>	MP3 recorder sample records sound to MP3 files stored on SD memory card and press button to play it.

## 9 SampleCode\PowerManagement

<b>SYS_DPDMode_Wakeup</b>	Show how to wake up system form DPD Power-down mode by Wake-up pin(PC.0), Wake-up Timer, RTC Tick, RTC Alarm or RTC Tamper 0.
<b>SYS_PowerDown_MinCurrent</b>	Demonstrate how to minimize power consumption when entering power down mode.
<b>SYS_PowerDownMode</b>	Show how to enter to different Power-down mode and wake-up by RTC.
<b>SYS_PowerMode</b>	Show how to set different core voltage and main voltage regulator type.
<b>SYS_SPDMode_Wakeup</b>	Show how to wake up system form SPD Power-down mode by GPIO pin(PC.0), Wake-up Timer, Wake-up ACMP, RTC Tick, RTC Alarm, RTC Tamper 0, BOD or LVR.

## 10 SampleCode\StdDriver

### System Manager (SYS)

<b>SYS_BODWakeup</b>	Show how to wake up system form 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_TrimIRC</b>	Demonstrate how to use LXT to trim HIRC.

### Clock Controller (CLK)

<b>CLK_ClockDetector</b>	Show the usage of clock fail detector and clock frequency monitor function.
--------------------------	---

### Flash Memory Controller (FMC)

<b>FMC_APPROT</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_DualBankFwUpgrade</b>	Implement a firmware update mechanism based on dual bank flash architecture.
<b>FMC_ExecInSRAM</b>	Implement a code and execute in SRAM to program embedded Flash (support KEIL MDK only).
<b>FMC_FwUpgradeApplication</b>	Bank remap 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_MultiWordProgram</b>	Show how to read/program embedded flash by ISP function.

<b>FMC_OTP</b>	Demonstrate how to program, read and lock OTP.
<b>FMC_ReadAllOne</b>	Demonstrate how to use FMC Read-All-One ISP command to verify APROM/LDROM pages are all 0xFFFFFFFF or not.
<b>FMC_RW</b>	Demonstrate how to read/program embedded Flash by ISP function.
<b>FMC_XOM</b>	This sample code shows how to configure and setup an XOM region the perform XOM function.

### General Purpose I/O (GPIO)

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

### HyperBus Interface Controller (HBI)

<b>HBI_ExeInHRAM</b>	Implement a code to execute in HyperRAM.
<b>HBI_RW</b>	Show HyperRAM read/write control via HyperBus Interface.
<b>HBI_RW_MemMap</b>	Show HyperRAM read/write through HyperBus Interface.

### PDMA Controller (PDMA)

<b>PDMA_BasicMode</b>	Use PDMA0 Channel 2 to transfer data from memory to memory.
<b>PDMA_ScatterGather</b>	Use PDMA0 channel 4 to transfer data from memory to memory by scatter-gather mode.
<b>PDMA_ScatterGather_PingPongBuffer</b>	Use PDMA0 to implement Ping-Pong buffer by scatter-gather mode(memory to memory).



<b>PDMA_StrideRepeat</b>	Use PDMA0 channel 0 to transfer data from memory to memory with stride and repeat.
<b>PDMA_TimeOut</b>	Demonstrate PDMA0 channel 1 get/clear timeout flag with UART1.

## Timer Controller (TIMER)

<b>TIMER_ACMPTrigger</b>	Use ACMP to trigger Timer0 counter reset mode.
<b>TIMER_CaptureCounter</b>	Show how to use the timer2 capture function to capture timer2 counter value.
<b>TIMER_Delay</b>	Demonstrate the usage of TIMER_Delay() API to generate a 1 second delay.
<b>TIMER_EventCounter</b>	Demonstrates the timer event counter function.
<b>TIMER_FreeCountingMode</b>	Use the timer0 pin PA.11 to demonstrate timer free counting mode function. And displays the measured input frequency to UART console.
<b>TIMER_InterTimerTriggerMode</b>	Use the timer pin PB.5 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_PWM_AccumulatorInterruptStopMode</b>	Demonstrate TIMER PWM accumulator interrupt to stop counting.
<b>TIMER_PWM_AccumulatorInterruptTriggerPDMA</b>	Demonstrate TIMER PWM accumulator interrupt to trigger PDMA transfer.
<b>TIMER_PWM_Brake</b>	Demonstrate how to use Timer0 PWM brake function..
<b>TIMER_PWM_ChangeDuty</b>	Change duty cycle and period of output waveform in PWM down count type.
<b>TIMER_PWM_DeadTime</b>	Demonstrate Timer PWM Complementary mode and Dead-Time function.
<b>TIMER_PWM_OutputWaveform</b>	Demonstrate output different duty waveform in

rm	Timer0~Timer5 PWM.
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_CompareINT	Show how to reload the WWDT counter value.
-----------------	--

### Real Timer Clock (RTC)

RTC_Alarm_Test	Demonstrate the RTC alarm function. It sets an alarm 10 seconds after execution.
RTC_Alarm_Wakeup	Use RTC alarm interrupt event to wake up system.
RTC_Dynamic_Tamper	Demonstrate the RTC dynamic tamper function.
RTC_Spare_Access	Demonstrate the RTC spare register read/write function and displays test result to the UART console.
RTC_Static_Tamper	Demonstrate the RTC static tamper function.
RTC_Time_Display	Demonstrate the RTC function and displays current time to the UART console.

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

BPWM_Capture	Use BPWM0 Channel 0 to capture the BPWM1 Channel 0 Waveform.
BPWM_DoubleBuffer	Change duty cycle and period of output waveform by BPWM Double Buffer function.

<b>BPWM_OutputWaveform</b>	Demonstrate how to use BPWM counter output waveform.
<b>BPWM_SwitchDuty</b>	Change duty cycle of output waveform by configured period.
<b>BPWM_SyncStart</b>	Demonstrate how to use BPWM counter synchronous start function.

## Enhance PWM Generator and Capture Timer (EPWM)

<b>EPWM_AccumulatorINT_Tri ggerPDMA</b>	Demonstrate EPWM accumulator interrupt trigger PDMA.
<b>EPWM_AccumulatorStopMo de</b>	Demonstrate EPWM accumulator stop mode.
<b>EPWM_Brake</b>	Demonstrate how to use EPWM brake function.
<b>EPWM_Capture</b>	Capture the EPWM1 Channel 0 waveform by EPWM1 Channel 2.
<b>EPWM_DeadTime</b>	Demonstrate how to use EPWM Dead Zone function.
<b>EPWM_DoubleBuffer</b>	Change duty cycle and period of output waveform by EPWM Double Buffer function (Period loading mode).
<b>EPWM_OutputWaveform</b>	Demonstrate how to use EPWM output waveform.
<b>EPWM_PDMA_Capture</b>	Capture the EPWM1 Channel 0 waveform by EPWM1 Channel 2, and use PDMA to transfer captured data.
<b>EPWM_SwitchDuty</b>	Change duty cycle of output waveform by configured period.
<b>EPWM_SyncStart</b>	Demonstrate how to use EPWM counter synchronous start function.

## Enhanced Quadrature Encoder Interface (EQEI)

<b>EQEI_CompareMatch</b>	Show the usage of EQEI compare 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 data in UART IrDA mode.
<b>UART_LIN</b>	Transmit LIN frame including header and response in UART LIN mode.
<b>UART_PDMA</b>	Transmit and receive UART data with PDMA.
<b>UART_RS485</b>	Transmit and receive data in UART RS485 mode.
<b>UART_SingleWire</b>	Transmit and receive data by 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.

## Smart Card Host Interface (SC)

<b>SC_ReadATR</b>	Read the smartcard ATR from smartcard 0 interface.
<b>SC_ReadSIM_PhoneBook</b>	Demonstrate how to read phone book information in the SIM card.
<b>SC_Timer</b>	Demonstrate how to use SC embedded timer.

## I<sup>2</sup>S Controller (I<sup>2</sup>S)

<b>I2S_Codec</b>	This is an I <sup>2</sup> S demo using NAU8822/88L25 audio codec, and used to play back the input from line-in.
<b>I2S_Codec_PDMA</b>	This is an I <sup>2</sup> S demo with PDMA function connected with codec.
<b>I2S_MP3PLAYER</b>	MP3 player sample plays MP3 files stored on SD memory card.
<b>I2S_WAVPLAYER</b>	This is a WAV file player which plays back WAV file stored in SD memory card.

## Serial Peripheral Interface (SPI)

<b>SPI_Flash</b>	Access SPI flash through SPI interface.
<b>SPI_HalfDuplex</b>	Demonstrate SPI half-duplex mode. SPI0 will be configured as Master mode and SPI1 will be configured as Slave mode. Both SPI0 and SPI1 will be configured as 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 SPI0 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 <a href="#">SPI_SlaveFIFOmode</a> sample code.
<b>SPI_PDMA_LoopTest</b>	Demonstrate SPI data transfer with PDMA. QSPI0 will be configured as Master mode and SPI1 will be configured as Slave mode. Both TX PDMA function and RX PDMA function will be enabled.
<b>SPI_SlaveFIFOmode</b>	Configure SPI0 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 SPI0 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>	This is an I <sup>2</sup> S demo with PDMA function connected with audio codec.
<b>SPII2S_PDMA_Play</b>	This is an I <sup>2</sup> S demo for playing data and demonstrate how I <sup>2</sup> S works with PDMA.
<b>SPII2S_PDMA_PlayRecord</b>	This is an I <sup>2</sup> S demo for playing and recording data with PDMA function.
<b>SPII2S_PDMA_Record</b>	This is an I <sup>2</sup> S demo for recording data and demonstrate how I <sup>2</sup> S works with PDMA.
<b>SPII2S_Slave</b>	Configure SPI0 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.

## SPI Synchronous Serial Interface Controller (SPIM)

<b>SPIM_CIPHER</b>	Demonstrate SPIM DMA read/write with cipher enabled. This sample also dumps SPI flash content via I/O mode read to prove it is encrypted cipher context.
<b>SPIM_DMA_RW</b>	Show SPIM DMA mode read/write function.
<b>SPIM_DMA_RW</b>	Demonstrate SPIM DMM mode read function. This sample programs SPI flash with DMA write and verify flash with DMA read and DMM mode CPU read respectively.
<b>SPIM_DMM_RUN_CODE</b>	This sample shows how to make an application booting from APROM with a sub-routine resided on SPIM flash.
<b>SPIM_IO_RW</b>	This sample demonstrates how to issue SPI flash erase, program, and read commands under SPIM I/O mode.

## Programmable Serial I/O (PSIO)

<b>PSIO_1Wire</b>	Demonstrate how to implement 1-Wire protocol by PSIO.
<b>PSIO_DM512</b>	Demonstrate how to implement DM512 protocol by PSIO.
<b>PSIO_IR</b>	Demonstrate how to implement NEC IR protocol by PSIO.
<b>PSIO_LED</b>	Demonstrate how to light up the WS1812B LED array.
<b>PSIO_PS2_Device</b>	Demonstrate how to implement PS/2 slave protocol by PSIO.
<b>PSIO_PS2_Host</b>	Demonstrate how to implement PS/2 host protocol by PSIO.
<b>PSIO_Wiegand</b>	Demonstrate how to implement Wiegand26 protocol by PSIO.

## Quad Serial Peripheral Interface (QSPI)

<b>QSPI_DualMode_Flash</b>	Access SPI flash using QSPI dual mode.
<b>QSPI_QuadMode_Flash</b>	Access SPI flash using QSPI quad mode.
<b>QSPI_Slave3Wire</b>	Configure QSPI0 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.

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

<b>I2C_EEPROM</b>	Demonstrate how to access EEPROM through a I <sup>2</sup> C interface
<b>I2C_Loopback</b>	Demonstrate how a Master accesses Slave.
<b>I2C_Master</b>	Demonstrate how a Master accesses 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-bytes API to access slave. This sample code needs to work with <a href="#">I2C_Slave</a> sample code.
<b>I2C_PDMA_TRX</b>	Demonstrate I2C PDMA mode and need to connect I2C0(Master) and I2C1(Slave).
<b>I2C_SingleByte_Master</b>	Demonstrate how to use single byte API to access slave. This sample code needs to work with <a href="#">I2C_Slave</a> sample code.
<b>I2C_Slave</b>	Demonstrate how to set I <sup>2</sup> C in slave mode to receive the 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 through I2C interface. This sample code needs to work with <a href="#">I2C_Master</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 using auto flow control.
<b>USCI_UART_PDMA</b>	Transmit and receive UART data with PDMA.
<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 a RS232 interface.
<b>USCI_UART_Wakeup</b>	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)

<b>USCI_SPI_Loopback</b>	Implement USCI_SPI1 Master loop back transfer. This sample code needs to connect USCI_SPI1_MISO pin and USCI_SPI1_MOSI pin together. It will compare the received data with transmitted data.
<b>USCI_SPI_MasterMode</b>	Configure USCI_SPI1 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_SlaveMode</b>	Configure USCI_SPI1 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 - 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 access 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 use 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 I2C Master and I2C Slave.
<b>USCI_I2C_MultiBytes_Master</b>	Demonstrate how to use multi-bytes API to access slave. This sample code needs to work with <a href="#">USCI_I2C_Slave</a> sample code.
<b>USCI_I2C_SingleByte_Master</b>	Demonstrate how to use single byte API to access slave. This sample code needs to work with <a href="#">USCI_I2C_Slave</a> sample code.
<b>USCI_I2C_Slave</b>	Demonstrate how to set I <sup>2</sup> C 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 I <sup>2</sup> C 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 I <sup>2</sup> C to wake up MCU from Power-down mode. This sample code needs to work <a href="#">USCI_I2C_Master</a> sample code.

## External Bus Interface (EBI)

<b>EBI_NOR</b>	Configure EBI interface to access MX29LV320T (NOR Flash) on EBI interface.
<b>EBI_SRAM</b>	Configure EBI interface to access BS616LV4017(SRAM) on EBI interface.

## 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 use GPIO to simulate key input.
<b>USBD_HID_RemoteWakeup</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 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_Keyboard</b>	Demonstrate how to implement a composite device (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 (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_Mass_Storage_CDROM</b>	Demonstrate the emulation of USB Mass Storage Device CD-ROM.
<b>USBD_Mass_Storage_Flash</b>	Use Flash as storage to implement a USB Mass-Storage device.
<b>USBD_Mass_Storage_SD</b>	Use SD card as storage to implement a USB Mass-Storage device.
<b>USBD_Mass_Storage_SRAM</b>	Use internal SRAM as back end storage media to simulate a 44 KB USB pen drive.
<b>USBD_Micro_Printer</b>	Demonstrate how to implement a USB micro printer device.
<b>USBD_Printer_And_HID_Transfer</b>	Demonstrate how to implement a composite device(USB micro printer device and HID Transfer). 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_VCOM_And_HID_Keyboard</b>	Demonstrate how to implement a composite device(VCOM and HID Keyboard).
<b>USBD_VCOM_And_HID_Transfer</b>	Demonstrate how to implement a composite device(VCOM and HID Transfer). It supports one virtual COM port and transfers 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_VCOM_And_Mass_S</b>	Demonstrate how to implement a composite device (Virtual COM port and Mass storage device).

storage	
<b>USBD_VCOM_DualPort</b>	Demonstrate how to implement a USB dual 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 works as a proprietary Vendor LBK device. It's created for sample USBH_VENDOR_LBK 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.

## High Speed USB 2.0 Device Controller (HSUSBD)

<b>HSUSBD_Audio10_Codec</b>	An UAC1.0 sample used to record and play the sound sent from PC through the USB interface.
<b>HSUSBD_Audio10_Headset</b>	An UAC1.0 sample and used to plays the sound send from PC through the USB interface.
<b>HSUSBD_Audio20_Codec</b>	An UAC2.0 sample used to record and play the sound sent from PC through the USB interface.
<b>HSUSBD_Audio20_Headset</b>	An UAC2.0 sample used to play the sound sent from PC through the USB interface.
<b>HSUSBD_HID_Mouse</b>	Simulate a USB mouse and draws circle on the screen.
<b>HSUSBD_HID_MouseKeyboard</b>	Simulate a USB mouse and a USB keyboard.
<b>HSUSBD_HID_Transfer</b>	Demonstrate how to 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>HSUSBD_HID_Transfer_And_MSC</b>	Demonstrate how to implement a composite device (HID Transfer and Mass storage). Transfer data between USB device and PC through the USB HID interface. A windows tool is also included in this sample code to connect with a USB device.
<b>HSUSBD_Mass_Storage_CDROM</b>	USB Mass Storage Device CD-ROM Emulation.
<b>HSUSBD_Mass_Storage_</b>	Use embedded Data Flash as storage to implement a USB

<b>DataFlash</b>	Mass-Storage device.
<b>HSUSBD_Mass_Storage_SactterGather</b>	Demonstrate the usage of USB DMA scatter gather function.
<b>HSUSBD_Mass_Storage_S D</b>	Implement a SD card reader.
<b>HSUSBD_Mass_Storage_ShortPacket</b>	Implement a mass storage class sample to demonstrate how to receive a USB short packet.
<b>HSUSBD_Mass_Storage_S RAM</b>	Use internal SRAM as back end storage media to simulate a 30 KB USB pen drive.
<b>HSUSBD_VCOM_SerialEmulator</b>	Demonstrate how to implement a USB virtual com port device.
<b>HSUSBD_VENDOR_LBK</b>	Implement a proprietary Vendor LBK device. This sample requires a M460 USB host running sample HSUSBD_USBH_VENDOR_LBK to be connected.

## USB 1.1/2.0 Host Controller (HSUSBH)

<b>HSUSBH_USBH_AudioClasses</b>	Demonstrate how to use USBH Audio Class driver. It shows the mute, volume, auto-gain, channel, and sampling rate control.
<b>HSUSBH_USBH_DEV_CON N</b>	Use connect/disconnect callback functions to handle of device connect and disconnect events.
<b>HSUSBH_USBH_Firmware_Update</b>	Automatically search and read new firmware from USB drive, if found, update APROM Flash with it.
<b>HSUSBH_USBH_HID</b>	Use USB Host core driver and HID driver. This sample demonstrates how to submit HID class request and read data from interrupt pipe. This sample supports dynamic device plug/un-plug and multiple HID devices
<b>HSUSBH_USBH_HID_Keyboard</b>	Demonstrate reading key inputs from USB keyboards. This sample includes an USB keyboard driver which is based on the HID driver.
<b>HSUSBH_USBH_HIDMouse_Keyboard</b>	Demonstrates how to support USB mouse and keyboard input.
<b>HSUSBH_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>HSUSBH_USBH_SPIM_Write</b>	Provide a command line interface for reading files from USB disk and writing to SPIM Flash. This sample code also provides

er	functions of dump SPIM Flash, compares USB disk file with SPIM Flash, and branches to run code on SPIM.
HSUSBH_USBH_UAC_HID	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).
HSUSBH_USBH_UAC_LoopBack	Receive audio data from an UAC device, and immediately send back to the UAC device.
HSUSBH_USBH_VCOM	Demonstrate how to use the USB Host core driver and CDC driver to connect a CDC class VCOM device.
HSUSBH_USBH_VENDOR_LBK	Show how to do transfer on a known device with a vendor driver. This sample requires a M460 USB device running sample <a href="#">HSUSBD_VENDOR_LBK</a> or <a href="#">USBD_VENDOR_LBK</a> to be connected.

## USB On-The-Go (OTG)

OTG_Dual_Role_UMAS	An OTG sample code that will become a USB host when connected with a Micro-A cable, and can access the pen drive when plugged in. It will become a removable disk when connected with a Micro-B cable, and then plug into PC.
OTG_HNP	Show HID mouse with OTG HNP protocol.

## High Speed USB On-The-Go (HSOTG)

HSOTG_Dual_Role_UMAS	An OTG sample code that will become a USB host when connected with a Micro-A cable, and can access the pen drive when plugged in. It will become a removable disk when connected with a Micro-B cable, and then plug into PC.
HSOTG_HNP	Show HID mouse with OTG HNP protocol.

## Ethernet MAC Controller (EMAC)

EMAC_Iwiperf	A LwIP iperf sample on M460.
EMAC_TxRx	This Ethernet sample tends to get a DHCP lease from DHCP server. After IP address configured, this sample can reply to PING packets.

## 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	Implement CRC in CRC-32 mode with PDMA transfer.

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

EADC_Accumulate	Demonstrate how to get accumulate conversion result.
EADC_ADINT_Trigger	Use ADINT interrupt to do the EADC continuous scan conversion.
EADC_Average	Demonstrate how to get average conversion result.
EADC_BandGap	Convert Band-gap (Sample module 16) and print conversion result.
EADC_PDMA_EPWMTrigger	Demonstrate how to trigger EADC by EPWM and transfer conversion data by PDMA.
EADC_EPWMTrigger	Demonstrate how to trigger EADC by EPWM.
EADC_Pending_Priority	Demonstrate how to trigger multiple sample modules and got conversion results in order of priority.
EADC_ResultMonitor	Monitor the conversion result of channel 2 by the digital compare function.
EADC_SWTRG_Trigger	Trigger EADC by writing EADC_SWTRG register.
EADC_TempSensor	Convert temperature sensor (Sample module 17) and print conversion result.
EADC_TimerTrigger	Show how to trigger EADC by timer.
EADC_VBat	Convert VBAT/4 (Sample module 18) and print conversion result.

## Digital-to-Analog Converter (DAC)

DAC_EPWMTrigger	Demonstrate how to trigger DAC by EPWM.
-----------------	---

<b>DAC_ExtPinTrigger</b>	Demonstrate how to trigger DAC conversion by external pin.
<b>DAC_GroupMode</b>	Demonstrate DAC0 and DAC1 work in group mode
<b>DAC_PDMA_EPWMTrigger</b>	Demonstrate how to use PDMA and trigger DAC0 by EPWM.
<b>DAC_PDMA_TimerTrigger</b>	Demonstrate how to PDMA and trigger DAC by Timer.
<b>DAC_SoftwareTrigger</b>	Demonstrate how to trigger DAC conversion by software.
<b>DAC_TimerTrigger</b>	Demonstrate how to trigger DAC by timer.

### Analog Comparator Controller (ACMP)

<b>ACMP_CompareDAC</b>	Demonstrate how ACMP compare DAC output with ACMP1_P1 value.
<b>ACMP_CompareVBG</b>	Demonstrate how ACMP compare VBG output with ACMP1_P1 value.
<b>ACMP_Wakeup</b>	Show how to wake up MCU from Power-down mode by ACMP wake-up function.
<b>ACMP_WindowComapre</b>	Demonstrate the usage of ACMP window compare function.

### Biphase Mark Coding Controller (BMC)

<b>BMC_Output</b>	BMC output demo.
<b>BMC_PDMA_Output</b>	BMC with PDMA output demo.

### Controller Area Network (CAN)

<b>CANFD_CANFD_Loopback</b>	Use CAN FD mode function to do internal loopback test.
<b>CANFD_CANFD_TxRx</b>	Transmit and receive CAN FD message through CAN interface.
<b>CANFD_CANFD_TxRxINT</b>	An example of interrupt control using CAN FD bus communication.

### Camera Capture Interface Controller (CCAP)



<b>CCAP_Mono_1Bit_Luma</b>	Use luminance 8-bit to 1-bit conversion to store captured image from HM01B0 sensor to SRAM.
<b>CCAP_Packet_DownScale</b>	Use packet format (all the luma and chroma data interleaved) to store captured image from NT99141 sensor to SRAM.
<b>CCAP_Packet_JpegEncode</b>	Use packet format (all the luma and chroma data interleaved) to store captured image from NT99141 sensor to SRAM and encode the image to jpeg.

## Key Store (KS)

<b>KS_AESKey</b>	Demo to use the AES in Key Store.
<b>KS_ECDH</b>	Demo to use ECC ECDH with Key Store.
<b>KS_ECDSA</b>	Demo to use the ECC ECDSA with Key Store.

## Cryptographic Accelerator (CRYPTO)

<b>CRYPTO_AES</b>	Show Crypto IP AES-128 ECB mode encrypt/decrypt function.
<b>CRYPTO_AES_CCM</b>	Demonstrate how to encrypt/decrypt data by AES CCM.
<b>CRYPTO_AES_GCM</b>	Demonstrate how to encrypt/decrypt data by AES GCM.
<b>CRYPTO_ECC_Demo</b>	ECDSA signature and verification demo
<b>CRYPTO_ECC_ECDH</b>	ECDH demonstrate how to calculate share key by A private key and B private key.
<b>CRYPTO_HMAC</b>	Show Crypto IP HMAC function.
<b>CRYPTO_RSA</b>	Show how to use Crypto RSA engine to sign and verifysignatures.
<b>CRYPTO_RSA_AccessKeyStore</b>	Use Crypto RSA engine accesses key from key store to sign and verify signatures.
<b>CRYPTO_RSA_CRTBypass</b>	Show how to use Crypto RSA engine CRT/CRT bypass mode to sign two signatures.
<b>CRYPTO_RSA_CRTBypass</b>	Use Crypto RSA engine CRT/CRT bypass mode accesses



<b>AccessKeyStore</b>	key from key store to sign and verify signatures.
<b>CRYPTO_SHA</b>	Use Crypto IP SHA engine to run through known answer SHA1 test vectors.

### Enhanced Input Capture Timer (ECAP)

<b>ECAP_GetInputFreq</b>	Show how to use ECAP interface to get input frequency
<b>ECAP_GetQEIFreq</b>	Show how to use ECAP interface to get QEI frequency.

### Random Number Generator (RNG)

<b>RNG_EntropyPoll</b>	Demo to use ECC ECDH with Key Store.
<b>RNG_Random</b>	Demo to use ECC ECDH with Key Store.

### SCUART

<b>SCUART_TxRx</b>	Demonstrate smartcard UART mode by connecting PB.4 and PB.5 pins.
--------------------	---

### Secure Digital Host Controller (SDH)

<b>SDH_FATFS</b>	Access a SD card formatted in FAT file system.
------------------	--

## 11 SampleCode\Template

Template	A project template for M460 MCU.
----------	----------------------------------

## 12 SampleCode\XOM

XOMLib	Demonstrate how to create XOM library.
XOMLibDemo	Demonstrate how to use <a href="#">XOMLib</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.*