

## M480 Series BSP Guide

Directory Introduction for 32-bit NuMicro® Family

### Directory Information

<b>Document</b>	Driver reference guide and revision history.
<b>Library</b>	Driver header and source files.
<b>SampleCode</b>	Driver sample code.
<b>ThirdParty</b>	Library from third party, including FatFs, LibMAD, lwIP, uIP, FreeRTOS™, libjpeg, and Mbed TLS.

*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

1	DOCUMENT.....	4
2	LIBRARY .....	5
3	SAMPLECODE .....	6
4	THIRDPARTY.....	7
5	SAMPLECODE\CORTEXM4.....	8
6	SAMPLECODE\ISP .....	9
7	SAMPLECODE\NUMAKER-M487MKCAN .....	10
8	SAMPLECODE\NUMAKER-PFM-M487 .....	11
9	SAMPLECODE\ POWERMANAGEMENT.....	13
10	SAMPLECODE\SECUREBOOT .....	14
11	SAMPLECODE\SECUREBOOT_LD .....	15
12	SAMPLECODE\STDDRIVER.....	16
	System Manager (SYS) .....	16
	Clock Controller (CLK).....	16
	Flash Memory Controller (FMC) .....	16
	External Bus Interface (EBI) .....	17
	General Purpose I/O (GPIO).....	17
	PDMA Controller (PDMA) .....	18
	Timer Controller (TIMER).....	18
	Watchdog Timer (WDT) .....	19
	Window Watchdog Timer (WWDT) .....	19
	Real Timer Clock (RTC) .....	19
	Basic PWM Generator and Capture Timer (BPWM).....	20

Enhanced PWM Generator and Capture Timer (EPWM) .....	20
Enhanced Input Capture Timer (ECAP) .....	21
Quadrature Encoder Interface(QEI).....	21
UART Interface Controller (UART).....	21
Smartcard Host Interface (SC) .....	21
Quad Serial Peripheral Interface (QSPI) .....	22
Serial Peripheral Interface (SPI).....	22
Serial Peripheral Interface Master Mode (SPIM).....	23
I <sup>2</sup> C Serial Interface Controller (I <sup>2</sup> C) .....	23
Universal Serial Control Interface Controller - UART Mode (USCI-UART) .....	24
Universal Serial Control Interface Controller - I2C Mode (USCI-I2C) .....	25
Universal Serial Control Interface Controller - SPI Mode (USCI-SPI).....	25
I <sup>2</sup> S Controller (I <sup>2</sup> S).....	26
Controller Area Network (CAN).....	26
USB 1.1 Device Controller (USBD) .....	27
High Speed USB 2.0 Device Controller (HSUSBD) .....	29
USB 1.1/2.0 Host Controller (HSUSBH).....	30
USB On-The-Go (OTG).....	31
High Speed USB On-The-Go (HSOTG).....	31
Ethernet MAC Controller (EMAC) .....	32
SD Host Controller (SDH).....	32
Crypto Accelerator (CRYPTO).....	32
True Random Number Generator (TRNG) .....	33
Camera Capture Interface Controller (CCAP).....	33
CRC Controller (CRC) .....	33
Enhanced 12-bit Analog-to-Digital Converter (EADC) .....	34
Digital-to-Analog Converter (DAC) .....	34
Analog Comparator Controller (ACMP) .....	35
OP Amplifier (OPA).....	35
<b>13 SAMPLE CODE COMPATIBILITY LIST .....</b>	<b>36</b>

## 1 Document

<b>CMSIS.html</b>	Document of CMSIS version 4.5.0.
<b>NuMicro M480 CMSIS BSP Revision History.pdf</b>	This document shows the revision history of M480 BSP.
<b>NuMicro M480 Driver Reference Guide.html</b>	This document describes the usage of drivers in M480 BSP.

## 2 Library

<b>CMSIS</b>	Cortex® Microcontroller Software Interface Standard (CMSIS) V4.5.0 definitions by Arm® Corp.
<b>Device</b>	CMSIS compliant device header file.
<b>FWUpdate</b>	Firmware update library binary and header files.
<b>SmartcardLib</b>	Smartcard library binary and header file.
<b>StdDriver</b>	All peripheral driver header and source files.
<b>UsbHostLib</b>	USB host library source code.

### 3 SampleCode

<b>CortexM4</b>	Cortex®-M4 sample code.
<b>Crypto_MbedTLS</b>	Mbed TLS test suites using Crypto accelerator.
<b>FreeRTOS</b>	Simple FreeRTOS™ demo code.
<b>Hard_Fault_Sample</b>	Show hard fault information when hard fault happened.
<b>ISP</b>	ISP firmware samples.
<b>NuMaker-ETM-M487</b>	Sample codes for NuMaker-ETM-M487 board.
<b>NuMaker-M487KMCAN</b>	Sample codes for NuMaker-M487KMCAN board.
<b>NuMaker-PFM-M487</b>	Sample codes for NuMaker-PFM-M487 board.
<b>NuMaker-PFM-M487D</b>	Sample codes for NuMaker-PFM-M487D board.
<b>SecureBoot</b>	Firmware update samples in secure boot mode.
<b>SecureBootLD</b>	Sample codes for M48xGC/M48xG8 secure boot.
<b>Semihost</b>	Show how to debug with semi-host message print.
<b>StdDriver</b>	Sample code to demonstrate the usage of M480 MCU peripheral driver APIs.
<b>Template</b>	A project template for M480.

## 4 ThirdParty

<b>BLE_AB1602</b>	AB1602 BLE (Bluetooth Low Energy) module driver.
<b>FatFs</b>	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> .
<b>FreeRTOS</b>	A real time operating system available for free download. Its official website is: <a href="http://www.freertos.org/">http://www.freertos.org/</a> .
<b>libjpeg</b>	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> .
<b>LibMAD</b>	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.
<b>lwIP</b>	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> .
<b>mbedtls-2.13.0</b>	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>
<b>paho.mqtt.embedded-c</b>	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>
<b>uip-0.9</b>	uIP is a very small implementation of the TCP/IP stack that is written by Adam Dunkels <adam@sics.se>. More information can be obtained from the uIP homepage at <a href="http://www.sics.se/~adam/uip/">http://www.sics.se/~adam/uip/</a> .

## 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\ISP

<b>ISP_CAN</b>	In-System-Programming sample code through CAN interface.
<b>ISP_DFU</b>	In-System-Programming sample code through USB DDFU( Device Firmware Upgrade) class.
<b>ISP_DFU_20</b>	In-System-Programming sample code through HSUSB DDFU( Device Firmware Upgrade) class.
<b>ISP_HID</b>	In-System-Programming sample code through a USB HID interface.
<b>ISP_HID_20</b>	In-System-Programming sample code through a HSUSB 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.
<b>ISP_UART_SPIFLASH_M487KM</b>	In-System-Programming sample code that supports SPI Flash programming through UART interface.

## **7 SampleCode\NuMaker-M487MKCAN**

<b>APROM_Loader</b>	Show how to execute application stored in SPI Flash for M487MKCAN.
<b>SPIM_DMM_RUN_CODE</b>	Show how to make an application booting from APROM with a sub-routine resided on SPIM flash for M487MKCAN.
<b>SYS_DPDMode_Wakeup</b>	Demonstrate how to wake up system from Deep Power-down mode by Wake-up pin(PC.0) or Wake-up Timer for M487MKCAN.

## 8 SampleCode\NuMaker-PFM-M487

<b>BLE_AB1602</b>	BLE (Bluetooth Low Energy) samples demonstrate HOGP (HID over GATT Profile) and data transfer.
<b>Heart_beating</b>	Measure the heartbeat rate by amplifying and filtering electrocardiogram signals through OPA, and converting them into digit values through ADC. The calculated heartbeat rate will be sent and printed on screen by UART.
<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 192.168.0.2, or assign by DHCP server.
<b>LwIP_httpd_scket</b>	A simple HTTP server that demonstrates LwIP socket API under FreeRTOS™. This HTTP server's IP address can be configured statically to 192.168.0.2, 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™. This echo 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 server 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™. This 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.

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

## 10 SampleCode\SecureBoot

<b>HSUSBD_FWUpdate</b>	Use MKROM API to update firmware. This sample code is executed in APROM, and uses MKROM API to update LDROM. After system reset, the program will boot from LDROM.
<b>HSUSBD_IAP</b>	Use MKROM API to do IAP. This sample code is executed in APROM, and loads an image to LDROM to do Secure boot ISP to update APROM. Remember to reset the system after firmware update is complete.
<b>OTA_FWUpdate</b>	Use MKROM API to update firmware. This sample code is executed in LDROM, and uses it to boot from APROM or update APROM.
<b>UART_FWUpdate</b>	Use MKROM API to update firmware. This sample code is executed in LDROM, and uses MKROM API to update APROM. After system reset, the program will boot from APROM.

## 11 SampleCode\SecureBoot\_LD

SecureBootAP	Demonstrate how to generate a secure boot application. A post-build tool SecureBootMaker.exe is used to translate a bin file into a hex file for secure boot.
SecureISP	An example implementation of secure ISP firmware. Secure ISP firmware connects to the Host(PC) SecureISP tool via an UART or USB connection. It receives commands from Host to update the user application in APROM/LDROM. InM48xGC/M48xG8secure boot scenario, Secure Bootloader (MKROM) verifies and loads the ISP firmware, then in turn ISP firmware verifies and loads user applications.

## 12 SampleCode\StdDriver

### System Manager (SYS)

<b>SYS_BODWakeup</b>	Demonstrate how to wake up system from Power-down mode by brown-out detector interrupt.
<b>SYS_PLLClockOutput</b>	Change system clock to different PLL frequency and output system clock from CLKO pin.
<b>SYS_TrimIRC</b>	Demonstrate how to use LXT to trim HIRC.

### Clock Controller (CLK)

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

### Flash Memory Controller (FMC)

<b>FMC_CRC32</b>	Demonstrate how to use FMC CRC32 ISP command to calculate the CRC32 checksum of APROM, LDROM, and SPROM.
<b>FMC_Dual_Bank</b>	Show FMC dual bank capability. Non-blocking program APROM bank1 while program is running on APROM bank0 is running on bank0 without being blocked
<b>FMC_ExecInSRAM</b>	Implement a code and execute in SRAM to program embedded Flash.
<b>FMC_IAP</b>	Demonstrate FMC IAP boot mode and show how to use vector remap function. LDROM image was embedded in APROM image and be programmed to LDROM Flash at run-time. This sample also shows how to branch between APROM and LDROM.
<b>FMC_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 FMC multi-word program ISP command to program APROM 0x00000~0x20000 area.



<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>	Show FMC read Flash IDs, erase, read, and write functions.
<b>FMC_SecureKey</b>	Show how to setup the KPROM and how to perform KPROM comparison.
<b>FMC_SPROM</b>	Show how to make an application running on APROM but with a sub-routine on SPROM, which can be secured
<b>FMC_XOM</b>	An example of using FMC driver to set up and erase XOM regions.
<b>FMC_XOM_LibDemo</b>	Show a solution of calling the library resided in an XOM region.

## External Bus Interface (EBI)

<b>EBI_NOR</b>	Configure EBI interface to access NOR Flash connected to EBI interface.
<b>EBI_SRAM</b>	Configure EBI interface to access SRAM connected to EBI interface.

## 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.

## PDMA Controller (PDMA)

<b>PDMA_BasicMode</b>	Use PDMA channel 2 to demonstrate memory to memory transfer.
<b>PDMA_ScatterGather</b>	Use PDMA channel 5 to demonstrate memory to memory transfer by scatter-gather mode.
<b>PDMA_ScatterGather_PingPongBuffer</b>	Use PDMA to implement Ping-Pong buffer by scatter-gather mode (memory to memory).
<b>PDMA_Stride</b>	Use PDMA channel 2 to transfer data from memory to memory with stride.
<b>PDMA_Stride_Repeat</b>	Use PDMA channel 0 to transfer data from memory to memory with stride and repeat.
<b>PDMA_TimeOut</b>	Demonstrate PDMA timeout feature.

## Timer Controller (TIMER)

<b>TIMER_ACMPTTrigger</b>	Use ACMP to trigger timer 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>	Use pin PD.4 to demonstrate timer event counter function.
<b>TIMER_FreeCountingMode</b>	Use the timer pin PA.7 to demonstrate timer free counting mode function. And displays the measured input frequency to UART console.
<b>TIMER_InterTimerTriggerMode</b>	Use the timer pin PD.4 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_Brake</b>	Demonstrate how to use Timer PWM brake function.
<b>TIMER_PWM_ChangeDuty</b>	Change duty cycle and period of output waveform by Timer PWM Double Buffer function.
<b>TIMER_PWM_DeadTime</b>	Demonstrate how to use Timer PWM Dead Time function.
<b>TIMER_PWM_OutputWaveform</b>	Enable 4 Timer PWM output channels with different frequency and duty ratio.
<b>TIMER_TimeoutWakeup</b>	Use Timer to wake up system from Power-down mode periodically.
<b>TIMER_ToggleOut</b>	Demonstrate the timer 0 toggle out function on pin PD.4.

### Watchdog Timer (WDT)

<b>WDT_TimeoutWakeupAndReset</b>	Implement WDT time-out interrupt event to wake up system and generate time-out reset system event while WDT time-out reset delay period expired.
----------------------------------	--

### Window Watchdog Timer (WWDT)

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

### Real Timer Clock (RTC)

<b>RTC_Alarm_Test</b>	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_Dynamic_Tamper</b>	Show how to use RTC dynamic tamper function.
<b>RTC_Spare_Access</b>	Show how to access RTC spare registers.
<b>RTC_Static_Tamper</b>	Show how to use RTC static tamper function.
<b>RTC_Time_Display</b>	Demonstrate the RTC function and displays current time to the UART console.

## Basic PWM Generator and Capture Timer (BPWM)

<b>BPWM_Capture</b>	Use BPWM0 Channel 0 (PA.0) to capture the BPWM1 Channel 0(PE.13) Waveform
<b>BPWM_DoubleBuffer</b>	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.

## Enhanced PWM Generator and Capture Timer (EPWM)

<b>EPWM_AccumulatorINT_TriggerPDMA</b>	Demonstrate EPWM accumulator interrupt trigger PDMA.
<b>EPWM_AccumulatorStopMode</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 Time function.
<b>EPWM_DoubleBuffer</b>	Change duty cycle and period of output waveform by EPWM Double Buffer function.
<b>EPWM_OutputWaveform</b>	Demonstrate how to use PWM 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 PWM counter synchronous start function.

## Enhanced Input Capture Timer (ECAP)

ECAP_GetInputFreq	Show how to use ECAP to measure clock frequency.
ECAP_GetQEIFreq	Show how to use ECAP interface to get QEIA frequency.

## Quadrature Encoder Interface(QEI)

QEI_CompareMatch	Show the usage of QEI compare function.
------------------	---

## UART Interface Controller (UART)

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_LIN	Demonstrate how to send data to LIN bus.
UART_PDMA	Demonstrate UART transmit and receive function with PDMA.
UART_RS485	Transmit and receive data in UART RS485 mode.
UART_TxRxFunction	Transmit and receive data from PC terminal through a RS232 interface.
UART_Wakeup	Show how to wake up system from Power-down mode by UART interrupt.

## Smartcard Host Interface (SC)

SC_ReadATR	Read the smartcard ATR from smartcard 1 interface.
SC_ReadSIM_PhoneBook	Demonstrate how to read phone book information in the SIM card.
SC_Timer	Demonstrate how to use SC embedded timer
SCUART_TxRx	Demonstrate smartcard UART mode by connecting PA.0 and PA.1 pins.

## 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>	Demonstrate QSPI0 3-wire mode.

## Serial Peripheral Interface (SPI)

<b>SPI_Flash</b>	Access SPI Flash through a SPI interface.
<b>SPI_HalfDuplex</b>	Demonstrate SPI half-duplex mode.
<b>SPI_LoopBack</b>	A SPI read/write demo connecting SPI0 MISO and MOSI pins.
<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. SPI0 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 SPI1 as I <sup>2</sup> S Master mode and demonstrate how I <sup>2</sup> S works in Master mode.
<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 SPI1 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> .

## Serial Peripheral Interface Master Mode (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>	Demonstrate SPIM DMA mode read/write function.
<b>SPIM_DMM</b>	Demonstrate SPIM DMM mode read function. This sample programs SPI Flash with DMA write and verify Flash with DMA read and CPU read respectively.
<b>SPIM_DMM_RUN_CODE</b>	Show how to make an application booting from APROM with a sub-routine resided on SPIM flash.
<b>SPIM_IO_RW</b>	Demonstrate how to issue SPI Flash erase, program, and read commands under SPIM I/O mode.

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

<b>I2C_EEPROM</b>	Read/write EEPROM via I <sup>2</sup> C interface.
<b>I2C_Loopback</b>	Demonstrate how a Master accesses Slave.
<b>I2C_Master</b>	An I <sup>2</sup> C master mode demo code.
<b>I2C_Master_PDMA</b>	Demonstrate how a Master accesses Slave using PDMA TX mode and PDMA RX mode.
<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> .
<b>I2C_PDMA_TRX</b>	Demonstrate I <sup>2</sup> C PDMA mode that needs 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> .
<b>I2C_Slave</b>	An I <sup>2</sup> C slave mode demo code.
<b>I2C_Slave_PDMA</b>	Demonstrate how a Slave uses PDMA Rx mode receive data from a Master.
<b>I2C_SMBus</b>	Show how to control SMBus interface and use SMBus protocol between host and slave.
<b>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 with <a href="#">I2C Master</a> .

## 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_Master</b>	Transmit and receive data with auto flow control. This sample code needs to work with <a href="#">USCI_UART_Autoflow_Master</a> .
<b>USCI_UART_Autoflow_Slave</b>	Transmit and receive data with auto flow control. This sample code needs to work with <a href="#">USCI_UART_Autoflow_Slave</a> .
<b>USCI_UART_PDMA</b>	Demonstrate USCI_UART data transfer with PDMA.
<b>USCI_UART_RS485_Master</b>	Transmit and receive data in RS485 mode. This sample code needs to work with <a href="#">USCI_UART_RS485_Master</a> .
<b>USCI_UART_RS485_Slave</b>	Transmit and receive data in RS485 mode. This sample code needs to work with <a href="#">USCI_UART_RS485_Slave</a> .
<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 - I2C Mode (USCI-I2C)

<b>USCI_I2C_EEPROM</b>	Show how to use USCI_I2C interface to access EEPROM.
<b>USCI_I2C_Lookback</b>	Show a Master how to access 7-bit address Slave (loopback).
<b>USCI_I2C_Loopback_10bit</b>	Show a Master how to access 10-bit address Slave (loopback).
<b>USCI_I2C_Master</b>	Show a Master how to access Slave.
<b>USCI_I2C_Master_10bit</b>	Show a Master how to access 10-bit address Slave.
<b>USCI_I2C_Monitor</b>	Use USCI_I2C to monitor and log I <sup>2</sup> C bus traffic.
<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_MultiBytes_Master</a> .
<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_SingleByte_Master</a> .
<b>USCI_I2C_Slave</b>	Show a Slave how to receive data from Master.
<b>USCI_I2C_Slave_10bit</b>	Show a 10-bit address Slave how to receive data from Master.
<b>USCI_I2C_Wakeup_Slave</b>	Show how to wake-up USCI_I2C from deep sleep mode.

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

<b>USCI_SPI_Loopback</b>	A USCI_SPI read/write demo connecting USCI_SPI0 and USCI_SPI1 interface.
<b>USCI_SPI_MasterMode</b>	Configure USCI_SPI1 as Master mode and demonstrate how to communicate with an off-chip SPI Slave device. Needs to work with <a href="#">USCI_SPI_MasterMode</a> sample code.
<b>USCI_SPI_PDMA_LoopTest</b>	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.
<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_SlaveMode</a> sample code.

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

<b>I2S_Codec</b>	An I <sup>2</sup> S demo used to play back the input from line-in or MIC interface.
<b>I2S_Codec_PDMA</b>	An I <sup>2</sup> S with PDMA demo used to play back the input from line-in or MIC interface.
<b>I2S_MP3PLAYER</b>	A MP3 player sample that plays MP3 files stored on SD memory card.
<b>I2S_WAVPLAYER</b>	A WAV file player that plays back WAV file stored in a USB pen drive.

## Controller Area Network (CAN)

<b>CAN_BasicMode_Rx</b>	Demonstrate CAN bus receive a message with basic mode. This sample code could work with <a href="#">CAN_BasicMode_Tx</a> sample code.
<b>CAN_BasicMode_Tx</b>	Demonstrate CAN bus transmit a message with basic mode. This sample code could work with <a href="#">CAN_BasicMode_Rx</a> sample code.
<b>CAN_BasicMode_Tx_Rx</b>	Demonstrate CAN bus transmit and receive a message with basic mode by connecting CAN0 and CAN1 to the same CAN bus.
<b>CAN_NormalMode_Rx</b>	Demonstrate CAN bus receive a message with normal mode. This sample code could work with <a href="#">CAN_NormalMode_Tx</a> sample code.
<b>CAN_NormalMode_Tx</b>	Demonstrate CAN bus transmit a message with normal mode. This sample code could work with

	<a href="#">CAN_NormalMode_Rx</a> sample code.
<b>CAN_NormalMode_Tx_Rx</b>	Demonstrate CAN bus transmit and receive a message with normal mode by connecting CAN 0 and CAN1 to the same CAN bus.

## 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>	An UAC1.0 sample used to play the sound sent from PC through the USB interface.
<b>USBD_HID_Keyboard</b>	Demonstrate how to implement a USB keyboard device. This sample code supports to use GPIO to simulate key input.
<b>USBD_HID_Mouse</b>	Simulate a USB mouse and draws circle on the screen.
<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_RemoteWakeup</b>	Simulate a HID mouse 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 (HID Transfer and keyboard). Transfer data between 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_MSC</b>	Demonstrate how to implement a composite device (HID Transfer and Mass storage). Transfer data between 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_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 internal Flash as back end storage media to simulate a USB pen drive.
<b>USBD_Mass_Storage_SD</b>	Implement a SD card reader.
<b>USBD_Mass_Storage_SRAM</b>	Use internal SRAM as back end storage media to simulate a 30 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 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). Transfer data between 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_Storage</b>	Demonstrate how to implement a composite device (Virtual COM port and Mass storage device).
<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_VENDOR_LBK</b>	Implement a proprietary Vendor LBK device. This sample requires a M480 USB host running sample <a href="#">HSUSBD_USBH_VENDOR_LBK</a> to be connected.
------------------------	--

## 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_DataFlash</b>	Use embedded Data Flash as storage to implement a USB Mass-Storage device.
<b>HSUSBD_Mass_Storage_SactterGather</b>	Demonstrate the usage of USBD DMA scatter gather function.
<b>HSUSBD_Mass_Storage_SD</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_SRAM</b>	Use internal SRAM as back end storage media to simulate a 30 KB USB pen drive.
<b>HSUSBD_RNDIS</b>	Demonstrate how to implement a USB Ethernet (Remote Network Driver Interface Specification).
<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 M480 USB host running sample <a href="#">HSUSBD_USBH_VENDOR_LBK</a> to be connected.

## USB 1.1/2.0 Host Controller (HSUSBH)

<b>HSUSBH_USBH_AudioClass</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_CONN</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_Writer</b>	Provide a command line interface for reading files from USB disk and writing to SPIM Flash. This sample code also provides functions of dump SPIM Flash, compares USB disk file with SPIM Flash, and branches to run code on SPIM.
<b>HSUSBH_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>HSUSBH_USBH_UAC_LoopBack</b>	Receive audio data from an UAC device, and immediately send back to the UAC device.
<b>HSUSBH_USBH_VCOM</b>	Demonstrate how to use the USB Host core driver and CDC driver to connect a CDC class VCOM device.
<b>HSUSBH_USBH_VENDOR_LBK</b>	Show how to do transfer on a known device with a vendor driver. This sample requires a M480 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)

<b>OTG_Dual_Role_UMAS</b>	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.
<b>OTG_HNP</b>	Show HID mouse with OTG HNP protocol.

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

<b>HSOTG_Dual_Role_UMAS</b>	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.
<b>HSOTG_HNP</b>	Show HID mouse with OTG HNP protocol.



## Ethernet MAC Controller (EMAC)

EMAC_httpd	A LwIP httpd sample that supports CGI and SSI.
EMAC_lwiperf	A LwIP iperf speed test sample in M480.
EMAC_TimeStamp	Demonstrate the usage of Ethernet time stamp function. It sets current time to 1000 second and prints out current time every second. It also sets an alarm at 1010 second. And rewind current time by 5 seconds after the alarm.
EMAC_TxRx	This Ethernet sample tends to get a DHCP lease from DHCP server, and use 192.168.10.10 as IP address if failed to get a lease. After IP address configured, this sample can reply to PING packets.
EMAC_uIP_httpd	Implement a HTTP server using uIP.
EMAC_uIP_telnetd	Implement a Telnet server using uIP.

## SD Host Controller (SDH)

SDH_FATFS	Access a SD card formatted in FAT file system.
SDH_FATFS_Dual	Access two SD cards formatted in FAT file system.
SDH_Firmware_Update	Automatically search and read new firmware from SD card, if found, update APROM Flash with it.

## Crypto Accelerator (CRYPTO)

CRYPTO_AES	Show Crypto IP AES-128 ECB mode encrypt/decrypt function.
CRYPTO_ECC_GenerateSecretZ	Show Crypto IP ECC CDH secret Z generation.
CRYPTO_ECC_KeyGeneration	Show Crypto IP ECC P-192 key generation function.
CRYPTO_ECC_Signature Generation	Show Crypto IP ECC P-192 ECDSA signature generation function.
CRYPTO_ECC_Signature	Show Crypto IP ECC P-192 ECDSA signature



<b>Verification</b>	verification function.
<b>CRYPTO_HMAC</b>	Show Crypto IP HMAC function.
<b>CRYPTO_PRNG</b>	Generate random numbers using Crypto IP PRNG.
<b>CRYPTO_SHA</b>	Use Crypto IP SHA engine to run through known answer SHA1 test vectors.
<b>CRYPTO_TDES</b>	Show Crypto IP Triple DES CBC mode encrypt/decrypt function.

### True Random Number Generator (TRNG)

<b>TRNG_GenRndNum</b>	Generate random numbers using TRNG.
-----------------------	-------------------------------------

### 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 to SRAM.
<b>CCAP_Packet_JpegEncode</b>	Capture an image and convert to JPEG format.

### CRC Controller (CRC)

<b>CRC_CCITT</b>	Implement CRC in CRC-CCITT mode and get the CRC checksum result.
<b>CRC_CRC8</b>	Implement CRC in CRC-8 mode and get the CRC checksum result.
<b>CRC_CRC32</b>	Implement CRC in CRC-32 mode and get the CRC checksum result.

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

<b>EADC_ADINT_Trigger</b>	Use ADINT interrupt to do the EADC continuous scan conversion.
<b>EADC_BandGap</b>	Convert Band-gap (Sample module 16) and print conversion result.
<b>EADC_EPWM_Trigger</b>	Demonstrate how to trigger EADC by EPWM.
<b>EADC_PDMA_EPWM_Trigger</b>	Demonstrate how to trigger EADC by EPWM and transfer conversion data by PDMA.
<b>EADC_Pending_Priority</b>	Demonstrate how to trigger multiple sample modules and get conversion results in order of priority.
<b>EADC_ResultMonitor</b>	Monitor the conversion result of channel 2 by the digital compare function.
<b>EADC_SimultaneousMode</b>	Demonstrate EADC0 and EADC1 are triggered in simultaneous mode.
<b>EADC_SWTRG_Trigger</b>	Trigger EADC by writing EADC_SWTRG register.
<b>EADC_TempSensor</b>	Convert temperature sensor (Sample module 17) and print conversion result.
<b>EADC_Timer_Trigger</b>	Show how to trigger EADC by timer.

## Digital-to-Analog Converter (DAC)

<b>DAC_EPWMTrigger</b>	Demonstrate EPWM trigger DAC to convert sine wave outputs.
<b>DAC_ExtPinTrigger</b>	Demonstrate external pin trigger DAC convert sine wave outputs.
<b>DAC_GroupMode</b>	Show how to make 2 DAC output channels work in group mode.
<b>DAC_PDMA_EPWMTrigger</b>	Show EPWM trigger DAC to fetch data with PDMA and convert sine wave outputs.
<b>DAC_PDMA_TimerTrigger</b>	Show timer trigger DAC to fetch data with PDMA and

	convert sine wave outputs.
<b>DAC_SoftwareTrigger</b>	Demonstrate software trigger DAC to convert sine wave outputs.
<b>DAC_TimerTrigger</b>	Demonstrate timer trigger DAC to convert sine wave outputs.

## Analog Comparator Controller (ACMP)

<b>ACMP_ComapreDAC</b>	Demonstrate analog comparator (ACMP) comparison by comparing ACMP0_P0 input and DAC voltage and shows the result on UART console.
<b>ACMP_ComapreVBG</b>	Demonstrate analog comparator (ACMP) comparison by comparing ACMP0_P0 input and VBG voltage and shows the result on UART console.
<b>ACMP_Wakeup</b>	Use ACMP to wake up system from Power-down mode while comparator output changes.
<b>ACMP_WindowCompare</b>	Show how to monitor ACMP input with window compare function.
<b>ACMP_WindowLatch</b>	Demonstrate how to use ACMP window latch mode.

## OP Amplifier (OPA)

<b>OPA_Control</b>	Show how to control OPA.
--------------------	--------------------------

## 13 Sample Code Compatibility List

Part Number Sample Code	M48xxGAAE / M487KMCAN IDEA,	M48xxG8AE / GCAE
CortexM4	√	√
Crypto_MbedTLS	√	√
FreeRTOS	√	√
Hard_Fault_Sample	√	√
Semihost	√	√
Template	√	√
ISP_CAN	√	√
ISP_DFU	√	√
ISP_DFU_20	√	-
ISP_HID	√	√
ISP_HID_20	√	-
ISP_I2C	√	√
ISP_RS485	√	√
ISP_SPI	√	√
ISP_UART	√	√
ISP_UART_SPIFLASH_M487KM	√	-
NuMaker-ETM-M487	√	√
NuMaker-M487KMCAN	√	-
BLE_AB1602	√	-
Heart_beating	√	√
lwIP	√	-
LwIP_httpd_netconn	√	-
LwIP_httpd_socket	√	-
LwIP_MQTT	√	-
LwIP_SSL_Client	√	-
LwIP_SSL_Server	√	-
LwIP_TCP_EchoClient	√	-

LwIP_TCP_EchoServer	√	-
LwIP_tftp_client	√	-
LwIP_tftp_server	√	-
LwIP_UDP_EchoClient	√	-
LwIP_UDP_EchoServer	√	-
SecureBoot	√	-
SecureBoot_LD	-	√
ACMP_ComapreDAC	√	√
ACMP_ComapreVBG	√	√
ACMP_Wakeup	√	√
ACMP_WindowCompare	√	√
ACMP_WindowLatch	√	√
BPWM_Capture	√	√
BPWM_DoubleBuffer	√	√
BPWM_OutputWaveform	√	√
BPWM_SwitchDuty	√	√
BPWM_SyncStart	√	√
CAN_BasicMode_Rx	√	√
CAN_BasicMode_Tx	√	√
CAN_BasicMode_Tx_Rx	√	√
CAN_NormalMode_Rx	√	√
CAN_NormalMode_Tx	√	√
CAN_NormalMode_Tx_Rx	√	√
CCAP_Mono_1Bit_Luma	-	√
CCAP_Packet_DownScale	-	√
CCAP_Packet_JpegEncode	-	√
CLK_ClockDetector	√	√
CRC_CCITT	√	√
CRC_CRC8	√	√
CRC_CRC32	√	√

CRYPTO_AES	√	√
CRYPTO_ECC_GenerateSecretZ	√	√
CRYPTO_ECC_KeyGeneration	√	√
CRYPTO_ECC_SignatureGeneration	√	√
CRYPTO_ECC_SignatureVerification	√	√
CRYPTO_HMAC	√	√
CRYPTO_PRNG	√	√
CRYPTO_SHA	√	√
CRYPTO_TDES	√	-
DAC_EPWMTrigger	√	√
DAC_ExtPinTrigger	√	√
DAC_GroupMode	√	-
DAC_PDMA_EPWMTrigger	√	√
DAC_PDMA_TimerTrigger	√	√
DAC_SoftwareTrigger	√	√
DAC_TimerTrigger	√	√
EADC_ADINT_Trigger	√	√
EADC_BandGap	√	√
EADC_EPWM_Trigger	√	√
EADC_PDMA_EPWM_Trigger	√	√
EADC_Pending_Priority	√	√
EADC_ResultMonitor	√	√
EADC_SimultaneousMode	-	√
EADC_SWTRG_Trigger	√	√
EADC_TempSensor	√	√
EADC_Timer_Trigger	√	√
EBI_NOR	√	√
EBI_SRAM	√	√
ECAP_GetInputFreq	√	√
ECAP_GetQEIFreq	√	√

EMAC_httpd	√	-
EMAC_lwiperf	√	-
EMAC_TimeStamp	√	-
EMAC_TxRx	√	-
EMAC_uIP_httpd	√	-
EMAC_uIP_telnetd	√	-
EPWM_AccumulatorINT_TriggerPDMA	√	√
EPWM_AccumulatorStopMode	-	√
EPWM_Brake	√	√
EPWM_Capture	√	√
EPWM_DeadTime	√	√
EPWM_DoubleBuffer	√	√
EPWM_OutputWaveform	√	√
EPWM_PDMA_Capture	√	√
EPWM_SwitchDuty	√	√
EPWM_SyncStart	√	√
FMC_CRC32	√	√
FMC_Dual_Bank	√	-
FMC_ExecInSRAM	√	√
FMC_IAP	√	√
FMC_MultiBoot	√	√
FMC_MultiWordProgram	√	√
FMC_OTP	√	√
FMC_ReadAllOne	√	√
FMC_RW	√	√
FMC_SecureKey	√	√
FMC_SPROM	√	-
FMC_XOM	-	√
FMC_XOM_LibDemo	-	√
GPIO_EINTAndDebounce	√	√

GPIO_INT	√	√
GPIO_OutputInput	√	√
GPIO_PowerDown	√	√
HSOTG_Dual_Role_UMAS	√	-
HSOTG_HNP	√	-
HSUSBD_Audio10_Codec	√	-
HSUSBD_Audio10_Headset	√	-
HSUSBD_Audio20_Codec	√	-
HSUSBD_Audio20_Headset	√	-
HSUSBD_HID_Mouse	√	-
HSUSBD_HID_MouseKeyboard	√	-
HSUSBD_HID_Transfer	√	-
HSUSBD_HID_Transfer_And_MSC	√	-
HSUSBD_Mass_Storage_CDROM	√	-
HSUSBD_Mass_Storage_DataFlash	√	-
HSUSBD_Mass_Storage_SactterGather	√	-
HSUSBD_Mass_Storage_SD	√	-
HSUSBD_Mass_Storage_ShortPacket	√	-
HSUSBD_Mass_Storage_SRAM	√	-
HSUSBD_RNDIS	√	-
HSUSBD_VCOM_SerialEmulator	√	-
HSUSBD_VENDOR_LBK	√	-
HSUSBH_USBH_AudioClass	√	√
HSUSBH_USBH_DEV_CONN	√	√
HSUSBH_USBH_Firmware_Update	√	√
HSUSBH_USBH_HID	√	√
HSUSBH_USBH_HID_Keyboard	√	√
HSUSBH_USBH_HID_Mouse_Keyboard	√	√
HSUSBH_USBH_MassStorage	√	√



HSUSBH_USBH_SPIM_Writer	√	√
HSUSBH_USBH_UAC_HID	√	√
HSUSBH_USBH_UAC_LoopBack	√	√
HSUSBH_USBH_VCOM	√	√
HSUSBH_USBH_VENDOR_LBK	√	√
I2C_EEPROM	√	√
I2C_Loopback	√	√
I2C_Master	√	√
I2C_Master_PDMA	√	√
I2C_MultiBytes_Master	√	√
I2C_PDMA_TRX	√	√
I2C_SingleByte_Master	√	√
I2C_Slave	√	√
I2C_Slave_PDMA	√	√
I2C_SMBus	√	√
I2C_Wakeup_Slave	√	√
I2S_Codec	√	√
I2S_Codec_PDMA	√	√
I2S_MP3PLAYER	√	√
I2S_WAVPLAYER	√	√
OPA_Control	√	-
OTG_Dual_Role_UMAS	√	-
OTG_HNP	√	-
PDMA_BasicMode	√	√
PDMA_ScatterGather	√	√
PDMA_ScatterGather_PingPongBuffer	√	√
PDMA_Stride	√	√
PDMA_Stride_Repeat	-	√
PDMA_TimeOut	√	√

QEI_CompareMatch	√	√
QSPI_DualMode_Flash	√	√
QSPI_QuadMode_Flash	√	√
QSPI_Slave3Wire	√	√
RTC_Alarm_Test	√	√
RTC_Alarm_Wakeup	√	√
RTC_Dynamic_Tamper	√	√
RTC_Spare_Access	√	√
RTC_Static_Tamper	√	√
RTC_Time_Display	√	√
SC_ReadATR	√	√
SC_ReadSIM_PhoneBook	√	√
SC_Timer	√	√
SCUART_TxRx	√	√
SDH_FATFS	√	√
SDH_FATFS_Dual	√	√
SDH_Firmware_Update	√	√
SPI_Flash	√	√
SPI_HalfDuplex	√	√
SPI_LoopBack	√	√
SPI_MasterFIFOmode	√	√
SPI_PDMA_LoopTest	√	√
SPI_SlaveFIFOmode	√	√
SPII2S_Master	√	√
SPII2S_PDMA_Codec	√	√
SPII2S_PDMA_Play	√	√
SPII2S_PDMA_PlayRecord	√	√
SPII2S_PDMA_Record	√	√
SPII2S_Slave	√	√

SPIM_CIPHER	√	-
SPIM_DMA_RW	√	-
SPIM_DMM	√	-
SPIM_DMM_RUN_CODE	√	-
SPIM_IO_RW	√	-
SYS_BODWakeup	√	√
SYS_PLLClockOutput	√	√
SYS_TrimIRC	√	√
TIMER_ACMPTTrigger	√	√
TIMER_CaptureCounter	√	√
TIMER_Delay	√	√
TIMER_EventCounter	√	√
TIMER_FreeCountingMode	√	√
TIMER_InterTimerTriggerMode	√	√
TIMER_Periodic	√	√
TIMER_PeriodicINT	√	√
TIMER_PWM_Brake	√	√
TIMER_PWM_ChangeDuty	√	√
TIMER_PWM_DeadTime	√	√
TIMER_PWM_OutputWaveform	√	√
TIMER_TimeoutWakeup	√	√
TIMER_ToggleOut	√	√
TRNG_GenRndNum	√	-
UART_AutoBaudRate	√	√
UART_AutoFlow	√	√
UART_IrDA	√	√
UART_LIN	√	√
UART_PDMA	√	√
UART_RS485	√	√
UART_TxRxFunction	√	√

UART_Wakeup	√	√
USBD_Audio_Codec	√	√
USBD_Audio_Headset	√	√
USBD_HID_Keyboard	√	√
USBD_HID_Mouse	√	√
USBD_HID_MouseKeyboard	√	√
USBD_HID_RemoteWakeup	√	√
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_Mass_Storage_SD	√	√
USBD_Mass_Storage_SRAM	√	√
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_SerialEmulator	√	√
USBD_VENDOR_LBK	√	√
USCI_I2C_EEPROM	√	-
USCI_I2C_Lookback	√	-
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_TimeoutWakeupAndReset	√	√
WWDT_CompareINT	√	√

### **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.*