

## M433 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\ POWERMANAGEMENT.....	10
8	SAMPLECODE\STDDRIVER.....	11
	Analog Comparator Controller (ACMP) .....	11
	Basic PWM Generator and Capture Timer (BPWM).....	11
	Controller Area Network (CAN).....	11
	Clock Controller (CLK).....	12
	CRC Controller (CRC) .....	12
	Enhanced 12-bit Analog-to-Digital Converter (EADC) .....	12
	External Bus Interface (EBI) .....	13
	Enhanced Input Capture Timer (ECAP) .....	13
	Enhanced PWM Generator and Capture Timer (EPWM) .....	13
	Flash Memory Controller (FMC) .....	14
	General Purpose I/O (GPIO).....	15
	I <sup>2</sup> C Serial Interface Controller (I <sup>2</sup> C) .....	15
	USB On-The-Go (OTG).....	16
	PDMA Controller (PDMA) .....	16
	Quadrature Encoder Interface(QEI).....	16
	Quad Serial Peripheral Interface (QSPI) .....	17
	Real Timer Clock (RTC) .....	17
	Smartcard Host Interface (SC) .....	17

---

SD Host Controller (SDH) .....	17
Serial Peripheral Interface (SPI) .....	17
System Manager (SYS) .....	18
Timer Controller (TIMER).....	19
UART Interface Controller (UART).....	20
USB 1.1 Device Controller (USBD) .....	20
USB 1.1 Host Controller (USBH) .....	22
Watchdog Timer (WDT) .....	23
Window Watchdog Timer (WWDT) .....	23

## 1 Document

<b>CMSIS.html</b>	Document of CMSIS version 4.5.0.
<b>NuMicro M433 CMSIS BSP Revision History.pdf</b>	This document shows the revision history of M433 BSP.
<b>NuMicro M433 Driver Reference Guide.html</b>	This document describes the usage of drivers in M433 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>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>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>PowerManagement</b>	Power management sample code.
<b>Semihost</b>	Show how to debug with semi-host message print.
<b>StdDriver</b>	Sample code to demonstrate the usage of M433 MCU peripheral driver APIs.
<b>Template</b>	A project template for M433.

## 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> .

## 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_HID</b>	In-System-Programming sample code through a USB DDFU 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.

## 7 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.
<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 or LVR or BOD.

## 8 SampleCode\StdDriver

### Analog Comparator Controller (ACMP)

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

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

BPWM_Capture	Use BPWM0 Channel 0 (PA.0) to capture the BPWM1 Channel 0(PE.13) Waveform
BPWM_DoubleBuffer	Change duty cycle and period of output waveform by BPWM Double Buffer function.
BPWM_OutputWaveform	Demonstrate how to use BPWM counter output waveform.
BPWM_SwitchDuty	Change duty cycle of output waveform by configured period.
BPWM_SyncStart	Demonstrate how to use BPWM counter synchronous start function.

### Controller Area Network (CAN)

CAN_BasicMode_Rx	Demonstrate CAN bus receive a message with basic mode. This sample code could work with <a href="#">CAN_BasicMode_Tx</a> sample code.
CAN_BasicMode_Tx	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.

## Clock Controller (CLK)

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

## 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 got conversion results in order of priority.
<b>EADC_ResultMonitor</b>	Monitor the conversion result of channel 2 by the digital compare function.
<b>EADC_SWTRG_Trigger</b>	Trigger EADC by writing EADC_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.

### 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.

### Enhanced Input Capture Timer (ECAP)

<b>ECAP_GetInputFreq</b>	Show how to use ECAP to measure clock frequency.
<b>ECAP_GetQEIFreq</b>	Show how to use ECAP interface to get QEIA frequency.

### 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.

## 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_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_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.

## 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.

## 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_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> .

I2C_Slave	An I <sup>2</sup> C slave mode demo code.
I2C_SMBus	Show how to control SMBus interface and use SMBus protocol between host and slave.
I2C_Wakeup_Slave	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> .

## 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.

## PDMA Controller (PDMA)

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

## Quadrature Encoder Interface(QEI)

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



## Quad Serial Peripheral Interface (QSPI)

QSPI_DualMode_Flash	Access SPI Flash using QSPI dual mode.
QSPI_QuadMode_Flash	Access SPI Flash using QSPI quad mode.
QSPI_Slave3Wire	Demonstrate QSPI0 3-wire mode.

## 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_Time_Display	Demonstrate the RTC function and displays current time to the UART console.

## 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.

## SD Host Controller (SDH)

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

## Serial Peripheral Interface (SPI)

SPI_Flash	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 SPI_SlaveFIFOmode 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 SPI_MasterFIFOmode 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 SPII2S_Master.

## 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.

## Timer Controller (TIMER)

<b>TIMER_ACMPTrigger</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.

## UART Interface Controller (UART)

<b>UART_AutoBaudRate</b>	Show how to use auto baud rate detection function.
<b>UART_AutoFlow</b>	Transmit and receive data using auto flow control.
<b>UART_IrDA</b>	Transmit and receive UART data in UART IrDA mode.
<b>UART_LIN</b>	Demonstrate how to send data to LIN bus.
<b>UART_PDMA</b>	Demonstrate UART transmit and receive function with PDMA.
<b>UART_RS485</b>	Transmit and receive data in UART RS485 mode.
<b>UART_TxRxFunction</b>	Transmit and receive data from PC terminal through a RS232 interface.
<b>UART_Wakeup</b>	Show how to wake up system from Power-down mode by UART interrupt.

## USB 1.1 Device Controller (USBD)

<b>USBD_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 M433 USB host running sample USBH_VENDOR_LBK to be connected.

## USB 1.1 Host Controller (USBH)

<b>USBH_AudioClass</b>	Demonstrate how to use USBH Audio Class driver. It shows the mute, volume, auto-gain, channel, and sampling rate control.
<b>USBH_DEV_CONN</b>	Use connect/disconnect callback functions to handle of device connect and disconnect events.
<b>USBH_Firmware_Update</b>	Automatically search and read new firmware from USB drive, if found, update APROM Flash with it.
<b>USBH_HID</b>	Use USB Host core driver and HID driver. 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>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>USBH_HID_Mouse_Keyboard</b>	Demonstrates how to support USB mouse and keyboard input.

<b>USBH_MassStorage</b>	Use a command-shell-like interface to demonstrate how to use USBH mass storage driver and make it work as a disk driver under the FATFS file system.
<b>USBH_UAC_HID</b>	Show how to use USBH Audio Class driver and HID driver at the same time. The target device is a Game Audio (UAC+HID composite device).
<b>USBH_UAC_LoopBack</b>	Receive audio data from an UAC device, and immediately send back to the UAC device.
<b>USBH_VCOM</b>	Demonstrate how to use the USB Host core driver and CDC driver to connect a CDC class VCOM device.
<b>USBH_VENDOR_LBK</b>	Show how to do transfer on a known device with a vendor driver. This sample requires a M433 USB device running sample USBD_VENDOR_LBK to be connected.

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

### **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.*