M480 Series BSP Directory

Directory Introduction for 32-bit NuMicro® Family

Directory Information

Document	Driver reference manual and revision history.
Library	Driver header and source files.
SampleCode	Driver sample code.
ThirdParty	Library from third party, including FatFs, LibMAD, lwIP, uIP, and FreeRTOS™.

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

1 Document Information

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

2 Library Information

CMSIS	Cortex® Microcontroller Software Interface Standard (CMSIS) V4.5.0 definitions by ARM® Corp.
Device	CMSIS compliant device header file.
SmartcardLib	Smartcard library binary and header file.
StdDriver	All peripheral driver header and source files.
UsbHostLib	USB host library source code.

3 Sample Code Information

CortexM4	Cortex®-M4 sample code.
FreeRTOS	Simple FreeRTOS™ demo code.
FreeRTOS_lwIP_httpd	A simple HTTP server demonstrates LwIP under FreeRTOS™. This HTTP server's IP address could be configured statically to 192.168.0.2, or assign by DHCP server.
FreeRTOS_lwIP_TCP_ EchoServer	A TCP echo server which is implemented with LwIP under FreeRTOS™. This echo server listens to port 80, and its IP address could 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.
FreeRTOS_lwIP_UDP_ EchoServer	A UDP echo server which is implemented with LwIP under FreeRTOS™. This echo server listens to port 80, and its IP address could 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.
Hard_Fault_Sample	Show hard fault information when hard fault happened.
Heart_beating	Measure 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.
Semihost	Show how to debug with semi-host message print.
StdDriver	Sample code to demonstrate the usage of M480 MCU peripheral driver APIs.
Template	A project template for M480.

4 Third Party Information

FATFS	A generic FAT file system module for small embedded systems. Its official website is: http://elm-chan.org/fsw/ff/00index_e.html.
FreeRTOSV9.0.0	A real time operating system available for free download. Its official website is: http://www.freertos.org/.
LibMAD	A MPEG audio decoder library which 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 (http://www.underbit.com/) for the commercial license.
lwip-2.0.2	A widely used open source TCP/IP stack designed for embedded systems. Its official website is: http://savannah.nongnu.org/projects/lwip/.
uip-0.9	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 http://www.sics.se/~adam/uip/.

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\StdDriver

ACMP_ComapreDAC	Demonstrate analog comparator (ACMP) comparison by comparing ACMP0_P0 input and DAC voltage and shows the result on UART console.
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.
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.
CAN_BasicMode_Rx	Demonstrate CAN bus receive a message with basic mode.
CAN_BasicMode_Tx	Demonstrate CAN bus transmit a message with basic mode.
CAN_BasicMode_Tx_Rx	Demonstrate CAN bus transmit and receive a message with basic mode by connecting CAN0 and CAN1 to the same CAN bus.
CAN_NormalMode_Rx	Demonstrate CAN bus receive a message with normal mode.
CAN_NormalMode_Tx	Demonstrate CAN bus transmit a message with normal mode.

CAN_NormalMode_Tx_Rx	Demonstrate CAN bus transmit and receive a message with normal mode by connecting CAN 0 and CAN1 to the same CAN bus.
CLK_ClockDetector	Demonstrate the usage of clock fail detector and clock frequency range detector function.
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 and get the CRC checksum result.
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_SignatureGeneration	Show Crypto IP ECC P-192 ECDSA signature generation function.
CRYPTO_ECC_SignatureVerification	Show Crypto IP ECC P-192 ECDSA signature verification function.
CRYPTO_HMAC	Show Crypto IP HMAC function.
CRYPTO_PRNG	Generate random numbers using Crypto IP PRNG.
CRYPTO_SHA	Use Crypto IP SHA engine to run through known answer SHA1 test vectors.
CRYPTO_TDES	Show Crypto IP Triple DES CBC mode encrypt/decrypt function.
DAC_EPWMTrigger	Demonstrate EPWM trigger DAC to convert sine wave outputs.

DAC_ExtPinTrigger	Demonstrate external pin trigger DAC convert sine wave outputs.
DAC_GroupMode	Show how to make 2 DAC output channels work in group mode.
DAC_PDMA_EPWMTrigger	Show EPWM trigger DAC to fetch data with PDMA and convert sine wave outputs.
DAC_PDMA_TimerTrigger	Show timer trigger DAC to fetch data with PDMA and convert sine wave outputs.
DAC_SoftwareTrigger	Demonstrate software trigger DAC to convert sine wave outputs.
DAC_TimerTrigger	Demonstrate timer trigger DAC to convert sine wave outputs.
EADC_ADINT_Trigger	Use ADINT interrupt to do the EADC continuous scan conversion.
EADC_BandGap	Convert Band-gap (Sample module 16) and print conversion result.
EADC_EPWM_Trigger	Demonstrate how to trigger EADC by EPWM.
EADC_PDMA_EPWM_Trigger	Demonstrate how to trigger EADC by EPWM and transfer conversion data by PDMA.
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_Timer_Trigger	Show how to trigger EADC by timer.
EADC_VBat	Convert VBAT/4 (Sample module 18) and print conversion result.

EBI_NOR	Configure EBI interface to access NOR Flash connects on EBI interface.
EBI_SRAM	Configure EBI interface to access SRAM connects on EBI interface.
ECAP_GetInputFreq	Show how to use ECAP to measure clock frequency.
ECAP_GetQEIFreq	Show how to use ECAP interface to get QEIA frequency.
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.
EPWM_AccumulatorINT_TriggerPDMA	Demonstrate EPWM accumulator interrupt trigger PDMA.
EPWM_Brake	Demonstrate how to use EPWM brake function.
EPWM_Capture	Capture the EPWM1 Channel 0 waveform by EPWM1 Channel 2.
EPWM_DeadTime	Demonstrate how to use EPWM Dead Time function.
EPWM_DoubleBuffer	Change duty cycle and period of output waveform by EPWM Double Buffer function.
EPWM_OutputWaveform	Demonstrate how to use PWM output waveform.
EPWM_PDMA_Capture	Capture the EPWM1 Channel 0 waveform by EPWM1 Channel 2, and use PDMA to transfer captured data.

EPWM_SwitchDuty	Change duty cycle of output waveform by configured period.
EPWM_SyncStart	Demonstrate how to use PWM counter synchronous start function.
FMC_CRC32	Demonstrate how to use FMC CRC32 ISP command to calculate the CRC32 checksum of APROM, LDROM, and SPROM.
FMC_Dual_Bank	Show FMC dual bank capability. Non-blocking program APROM bank1 while program is running on APROM bank0 is running on bank0 without being blocked
FMC_ExeInSRAM	Implement a code and execute in SRAM to program embedded Flash.
FMC_IAP	Demonstrate FMC IAP boot mode and show how to use vector remap function. LDROM image was embedded in APROM image and be programmed to LDROM Flash at run-time. This sample also shows how to branch between APROM and LDROM.
FMC_MultiBoot	Implement a multi-boot system to boot from different applications in APROM. A LDROM code and 4 APROM code are implemented in this sample code.
FMC_MultiWordProgram	Show FMC multi-word program ISP command to program APROM 0x00000~0x20000 area..
FMC_OTP	Demonstrate how to program, read, and lock OTP.
FMC_ReadAllOne	Demonstrate how to use FMC Read-All-One ISP command to verify APROM/LDROM pages are all 0xFFFFFFFF or not.
FMC_RW	Show FMC read Flash IDs, erase, read, and write functions.
FMC_SecureKey	Show how to setup the KPROM and how to perform KPROM comparison.
FMC_SPROM	Show how to make an application running on APROM but with a sub-routine on SPROM, which can be secured
GPIO_EINTAndDebounce	Show the usage of GPIO external interrupt function and de-bounce function.

GPIO_INT	Show the usage of GPIO interrupt function.
GPIO_OutputInput	Show how to set GPIO pin mode and use pin data input/output control.
GPIO_PowerDown	Show how to wake up system from Power-down mode by GPIO interrupt.
HSOTG_Dual_Role_UMAS	An OTG sample code which 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.
HSUSBD_Audio10_Codec	An UAC1.0 sample used to record and play the sound sent from PC through the USB interface
HSUSBD_Audio20_Codec	An UAC2.0 sample used to record and play the sound sent from PC through the USB interface.
HSUSBD_HID_Mouse	Simulate a USB mouse and draws circle on the screen.
HSUSBD_HID_MouseKeyboard	Simulate a USB mouse and a USB keyboard.
HSUSBD_HID_Transfer	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.
HSUSBD_Mass_Storage_DataFlash	Use embedded Data Flash as storage to implement a USB Mass-Storage device.
HSUSBD_Mass_Storage_ShortPacket	Implement a mass storage class sample to demonstrate how to receive a USB short packet.
HSUSBD_Mass_Storage_SRAM	Use internal SRAM as back end storage media to simulate a 30 KB USB pen drive.
HSUSBD_Mass_Storage SactterGather	Demonstrate the usage of USBD DMA scatter gather function.
HSUSBD_VCOM_	Demonstrate how to implement a USB virtual com port

SerialEmulator	device.
HSUSBH_USBH_AudioClass	Demonstrate how to use USBH Audio Class driver. It shows the mute, volume, auto-gain, channel, and sampling rate control.
HSUSBH_USBH_DEV_CONN	Use connect/disconnect callback functions to handle of device connect and disconnect events.
HSUSBH_USBH_ Firmware_Update	Automatically search and read new firmware from USB drive, if found, update APROM Flash with it.
HSUSBH_USBH_HID	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.
HSUSBH_USBH_HID_Keyboard	Demonstrate reading key inputs from USB keyboards. This sample includes an USB keyboard driver which is based on the HID driver.
HSUSBH_USBH_MassStorage	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.
HSUSBH_USBH_SPIM_Writer	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.
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	Receives audio data from UAC device, and immediately send back to that UAC device
I2C_EEPROM	Read/write EEPROM via I^{2}C interface.
I2C_Loopback	Demonstrate how a Master accesses Slave.
I2C_Master	An I^{2}C master mode demo code.
I2C_Master_PDMA	Demonstrate how a Master accesses Slave using PDMA TX

	mode and PDMA RX mode.
I2C_MultiBytes_Master	Demonstrate how to use multi-bytes API to access slave. This sample code needs to work with I2C_Slave.
I2C_PDMA	Demonstrate I^{2}C PDMA mode which needs to connect I2C0 (Master) and I2C1 (Slave).
I2C_SingleByte_Master	Demonstrate how to use single byte API to access slave. This sample code needs to work with I2C_Slave.
I2C_Slave	An I^{2}C slave mode demo code.
I2C_Slave_PDMA	Demonstrate how a Slave uses PDMA Rx mode receive data from a Master.
I2C_Wakeup_Slave	Demonstrate how to set I2C to wake up MCU from Power-down mode. This sample code needs to work with I2C_Master.
I2S_Codec	An I^{2}S demo used to play back the input from line-in or MIC interface.
I2S_Codec_PDMA	An I^{2}S with PDMA demo used to play back the input from line-in or MIC interface.
I2S_MP3PLAYER	A MP3 player sample which plays MP3 files stored on SD memory card.
I2S_WAVPLAYER	A WAV file player which plays back WAV file stored in a USB pen drive.
OPA_Control	Show how to control OPA.
OTG_Dual_Role_UMAS	An OTG sample code which 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_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).
QEI_CompareMatch	Show the usage of QEI compare function.
SPI_DualMode_Flash	Access SPI Flash using QSPI dual mode.
SPI_QuadMode_Flash	Access SPI Flash using QSPI quad mode.
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	Show how to use RTC dynamic tamper function.
RTC_Spare_Access	Show how to access RTC spare registers.
RTC_Static_Tamper	Show how to use RTC static tamper function.
RTC_Time_Display	Demonstrate the RTC function and displays current time to the UART console.
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.
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.
SPI_Flash	Access SPI Flash through SPI interface.
SPI_LoopBack	A SPI read/write demo connecting QSPI0 MISO and

	MOSI pins.
SPI_MasterFIFOMode	Configure QSPI0 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.
SPI_PDMA_LoopTest	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.
SPI_SlaveFIFOMode	Configure QSPI0 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.
SPII2S_Master	Configure SPI1 as I^{2}S Master mode and demonstrate how I^{2}S works in Master mode.
SPII2S_PDMA_Codec	An I^{2}S demo with PDMA function connected with audio codec.
SPII2S_PDMA_Play	An I^{2}S demo for playing data and demonstrating how I^{2}S works with PDMA.
SPII2S_PDMA_PlayRecord	An I^{2}S demo for playing and recording data with PDMA function.
SPII2S_PDMA_Record	An I^{2}S demo for recording data and demonstrating how I^{2}S works with PDMA.
SPII2S_Slave	Configure SPI1 as I^{2}S Slave mode and demonstrate how I^{2}S works in Slave mode. This sample code needs to work with SPII2S_Master.
SPIM_CIPHER	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.
SPIM_DMA_RW	Demonstrate SPIM DMA mode read/write function.

SPIM_DMM	Demonstrate SPIM DMM mode read function. This sample programs SPI Flash with DMA write and verify Flash with DMA read and CPU read respectively.
SPIM_DMM_RUN_CODE	Show how to make an application booting from APROM with a sub-routine resided on SPIM flash.
SPIM_IO_RW	Demonstrate how to issue SPI Flash erase, program, and read commands under SPIM I/O mode.
SYS_BODWakeup	Demonstrate how to wake up system from Power-down mode by brown-out detector interrupt.
SYS_DPDMode_Wakeup	Demonstrate how to wake up system from Deep Power-down mode by Wake-up pin(PC.0) or Wake-up Timer or RTC Tick or RTC Alarm or RTC Tamper 0.
SYS_PLLClockOutput	Change system clock to different PLL frequency and output system clock from CLKO pin.
SYS_SPDMode_Wakeup	Demonstrate how to wake up system from Standby Power-down mode by GPIO pin(PA.0), Wake-up Timer, Wake-up ACMP, RTC Tick, RTC Alarm, RTC Tamper 0 , BOD, or LVR.
SYS_TrimIRC	Demonstrate how to use LXT to trim HIRC.
TIMER_ACMPTrigger	Use ACMP to trigger timer reset mode.
TIMER_CaptureCounter	Show how to use the timer2 capture function to capture timer2 counter value.
TIMER_Delay	Demonstrate the usage of TIMER_Delay() API to generate a 1 second delay.
TIMER_EventCounter	Use pin PD.4 to demonstrates timer event counter function.
TIMER_FreeCountingMode	Use the timer pin PA.7 to demonstrate timer free counting mode function. And displays the measured input frequency to UART console.
TIMER_InterTimerTriggerMode	Use the timer pin PD.4 to demonstrate inter-timer trigger mode function. Also display the measured input frequency to UART console.

TIMER_Periodic	Use the timer periodic mode to generate timer interrupt every 1 second.
TIMER_PeriodicINT	Implement timer counting in periodic mode.
TIMER_PWM_Brake	Demonstrate how to use Timer PWM brake function.
TIMER_PWM_ChangeDuty	Change duty cycle and period of output waveform by Timer PWM Double Buffer function.
TIMER_PWM_DeadTime	Demonstrate how to use Timer PWM Dead Time function.
TIMER_PWM_OutputWaveform	Enable 4 Timer PWM output channels with different frequency and duty ratio.
TIMER_ToggleOut	Demonstrate the timer 0 toggle out function on pin PD.4.
TIMER_Wakeup	Use Timer to wake up system from Power-down mode periodically.
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 RS232 interface.
UART_Wakeup	Show how to wake up system from Power-down mode by UART interrupt.
USBD_Audio_Codec	Demonstrate how to implement a USB audio class device.
USBD_HID_Keyboard	Demonstrate how to implement a USB keyboard device. This sample code supports to use GPIO to simulate key input.

USBD_HID_Mouse	Simulate a USB mouse and draws circle on the screen.
USBD_HID_MouseKeyboard	Simulate an USB HID mouse and HID keyboard. Mouse draws circle on the screen and Keyboard uses GPIO to simulate key input.
USBD_HID_RemoteWakeup	Simulate a HID mouse supports USB suspend and remote wakeup.
USBD_HID_Touch	Demonstrate how to implement a USB touch digitizer device. Two lines demo in Paint.
USBD_HID_Transfer	Demonstrate how to transfer data between a USB device and PC through a USB HID interface. A windows tool is also included in this sample code to connect with a USB device.
USBD_HID_Transfer_ And_Keyboard	Demonstrate how to implement a composite device (HID Transfer and keyboard). Transfer data between USB device and PC through USB HID interface. A windows tool is also included in this sample code to connect with a USB device.
USBD_HID_Transfer_ And_MSC	Demonstrate how to implement a composite device (HID Transfer and Mass storage). Transfer data between USB device and PC through USB HID interface. A windows tool is also included in this sample code to connect with a USB device.
USBD_HID_Transfer_CTRL	Use USB Host core driver and HID driver. It shows how to submit HID class request and how to read data from control pipe. A windows tool is also included in this sample code to connect with a USB device.
USBD_Mass_Storage_ CDROM	Demonstrate the emulation of USB Mass Storage Device CD-ROM.
USBD_Mass_Storage_ Flash	Use internal Flash as back end storage media to simulate a USB pen drive.
USBD_Mass_Storage_ SRAM	Use internal SRAM as back end storage media to simulate a 30 KB USB pen drive.
USBD_Micro_Printer	Demonstrate how to implement a USB micro printer

	device.
USBD_Printer_And_HID_Transfer	Demonstrate how to implement a composite device. (USB micro printer device and HID Transfer). Transfer data between USB device and PC through USB HID interface. A windows tool is also included in this sample code to connect with a USB device.
USBD_VCOM_And_HID_Keyboard	Demonstrate how to implement a composite device.(VCOM and HID keyboard)
USBD_VCOM_And_HID_Transfer	Demonstrate how to implement a composite device.(VCOM and HID Transfer) Transfer data between USB device and PC through USB HID interface. A windows tool is also included in this sample code to connect with a USB device.
USBD_VCOM_And_Mass_Storage	Demonstrate how to implement a composite device.(Virtual COM port and Mass storage device)
USBD_VCOM_DualPort	Demonstrate how to implement a USB dual virtual COM port device.
USBD_VCOM_SerialEmulator	Demonstrate how to implement a USB virtual COM port device.
USCI_I2C_EEPROM	Show how to use USCI_I2C interface to access EEPROM.
USCI_I2C_Lookback	Show a Master how to access 7-bit address Slave (loopback).
USCI_I2C_Loopback_10bit	Show a Master how to access 10-bit address Slave (loopback).
USCI_I2C_Master	Show a Master how to access Slave.
USCI_I2C_Master_10bit	Show a Master how to access 10-bit address Slave.
USCI_I2C_MultiBytes_Master	Demonstrate how to use multi-bytes API to access slave. This sample code needs to work with USCI_I2C_Slave.
USCI_I2C_SingleByte_Master	Demonstrate how to use single byte API to access slave. This sample code needs to work with USCI_I2C_Slave.
USCI_I2C_Slave	Show a Slave how to receive data from Master.

USCI_I2C_Slave_10bit	Show a 10-bit address Slave how to receive data from Master.
USCI_I2C_Wakeup_Slave	Show how to wake-up USCI_I2C from deep sleep mode.
USCI_SPI_Loopback	A USCI_SPI read/write demo connecting SPI0 and SPI1 interface.
USCI_SPI_MasterMode	Configure USCI_SPI1 as Master mode and demonstrate how to communicate with an off-chip SPI Slave device. Needs to work with USCI_SPI_SlaveMode sample code.
USCI_SPI_PDMA_LoopTest	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.
USCI_SPI_SlaveMode	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 USCI_SPI_MasterMode sample code.
USCI_UART_AutoBaudRate	Show how to use auto baud rate detection function.
USCI_UART_Autoflow_Master	Transmit and receive data with auto flow control. This sample code needs to work with USCI_UART_Autoflow_Slave.
USCI_UART_Autoflow_Slave	Transmit and receive data with auto flow control. This sample code needs to work with USCI_UART_Autoflow_Master.
USCI_UART_PDMA	Demonstrate UART data transfer with PDMA.
USCI_UART_RS485_Master	Transmit and receive data in RS485 mode. This sample code needs to work with USCI_UART_RS485_Slave.
USCI_UART_RS485_Slave	Transmit and receive data in RS485 mode. This sample code needs to work with USCI_UART_RS485_Master.
USCI_UART_TxRxFunction	Transmit and receive data from PC terminal through RS232 interface.
USCI_UART_Wakeup	Show how to wake up system from Power-down mode by

	USCI interrupt in UART mode.
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.
WWDT_CompareINT	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.